

Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informatiky a kvantitativních metod

Aplikace pro sportovní trenéry

Bakalářská práce

Autor: Netolický Štěpán
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Daniel Vondra
KIKM

Hradec Králové

duben 2024

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 17.4.2024

.....
Netolický Štěpán

Poděkování:

Děkuji vedoucímu bakalářské práce, Ing. Danielu Vondrovi, za metodické vedení práce a své rodině, přátelům a celému akademickému sboru, kteří mi při bakalářské práci velice pomohli a podporovali mě.

Anotace

Cílem bakalářské práce je vytvoření aplikace, která umožní evidovat úroveň zdatnosti hráčů ve sportovních klubech. Tato aplikace poskytne trenérům lepší orientaci a přehlednost o aktivitě jejich svěřenců. Jak trenér, tak i daný sportovec bude mít umožněno, díky uloženým datům, pracovat v rámci svých týmů a vytvářet si přehledné statistiky. V práci je také popsán vývoj a postupy při vytváření daného softwaru. Kromě jiného bakalářská práce obsahuje i popis technologií využitých ke zpracování a naprogramování této aplikace. Můžeme zde nalézt jiné dostupné možnosti programů, které slouží podobnému účelu.

Klíčová slova: webová aplikace, framework, java, aplikace, trénování, frontend

Annotation

Title: Application for a sport coaches

The aim of the bachelor's thesis is to create an application that allows to record the fitness level of players in sports clubs. This application will provide coaches with better orientation and clarity about the activity of their wards. Both the coach and the athlete in question will be able, thanks to the stored data, to work within their teams and create clear statistics. The thesis also describes the development and creation procedures of the software. Among other things, the thesis includes a description of the technologies used to develop and program the application. We can also find other available program options that serve a similar purpose.

Keywords: web application, framework, java, application, coaching, frontend

Obsah

1	Úvod.....	1
2	Cíl práce.....	5
3	Dostupná řešení.....	6
3.1	SidelineSports.....	6
3.2	SportLyzer.....	7
3.3	iTreneo.....	8
3.4	EOS club zone.....	9
4	Příprava vývoje aplikace.....	11
4.1	Volba druhu aplikace.....	13
4.1.1	Nativní aplikace.....	14
4.1.2	Hybridní aplikace.....	14
4.1.3	Webové aplikace.....	14
4.2	Návrh aplikace.....	16
4.2.1	Vaadin framework.....	16
4.2.2	Java.....	17
4.2.3	Intellij IDEA.....	18
4.2.4	PostgreSQL.....	19
4.2.5	Spring a SpringBoot.....	20
5	Vývoj aplikace.....	21
5.1	Programování aplikace.....	21
5.1.1	Frontend.....	22
5.1.2	Backend.....	30
5.1.3	Databáze aplikace.....	33
5.1.4	Uživatelský přístup.....	35
5.2	Testování aplikace.....	37

6	Shrnutí výsledků.....	39
7	Závěry a doporučení	41
8	Seznam použité literatury.....	42
9	Přílohy	45

Seznam obrázků

Obr. 1 Domovská stránka aplikace (zdroj vlastní).....	22
Obr. 2 Přihlašovací stránka aplikace (zdroj vlastní)	23
Obr. 3 Stránka s registrací nového uživatele (zdroj vlastní).....	24
Obr. 4 Stránka s přehledem týmů uživatele (zdroj vlastní)	25
Obr. 5 Detailní pohled na stránku týmu (zdroj vlastní).....	26
Obr. 6 Modální okno vytváření události (zdroj vlastní).....	27
Obr. 7 Detailní pohled konkrétní události (zdroj vlastní).....	28
Obr. 8 Stránka s kalendářem (zdroj vlastní)	29
Obr. 9 Stránka výkonů uživatele s časovou osou (zdroj vlastní).....	30
Obr. 10 Třída Scheduler (zdroj vlastní)	31
Obr. 11 Funkcionalita ukládání PDF souboru (zdroj vlastní).....	32
Obr. 12 Funkcionalita generování Excel souboru (zdroj vlastní)	33
Obr. 13 Struktura databáze (zdroj vlastní).....	34
Obr. 14 Detailní pohled události s nahraným PDF souborem (zdroj vlastní)	37
Obr. 15 Detailní pohled na statistiky hráčů (zdroj vlastní)	40
Obr. 16 Vygenerovaný .xlsx souboru s docházkou (zdroj vlastní)	40

1 Úvod

Webová aplikace pro trenéry byla zvolena jako bakalářská práce především z důvodu, že na internetu neexistuje žádné neplacené řešení pro správu a přehled trénovanosti hráčů. V praxi se můžete setkat s obvyklými případy, kdy trenéři florbalových klubů nevyužívají žádné nástroje pro evidenci a vedení jejich týmů, a to se poté projevuje i na výkonnosti jejich svěřenců. Často se pak organizace přiklání k řešení pomocí sociálních sítí nebo jiných platform. Těmito platformami jsou nejčastěji komunikační kanály jako je Discord nebo Teams. Tato řešení mohou poskytovat do určité míry přehled o docházce a možnost komunikace s hráči. Nikdy však týmům neumožní tak dobře evidovat poplatky klubu, zlepšování hráčů nebo uchování konceptů tréninků pro další používání. Dalším nesporným problémem, který se v klubech řeší, je sjednocení komunikace s hráči, trenéry, vedením a rodiči pod jednu platformu. Zpravidla se stává, že s jednou skupinou komunikujete přes email, s další přes WhatsApp a s jinou zase přes Discord. Z důvodu uvedených překážek a výzev vznikl záměr vytvořit aplikaci, která by mohla efektivněji reagovat na potřeby klubů. Tato iniciativa by měla směřovat k zajištění komplexnějšího řešení několika konkrétních problémů, se kterými se sportovní kluby běžně potýkají. Předpokládaná aplikace by tak měla poskytnout nástroje umožňující klubům účinnější správu a organizaci svých aktivit. V rámci řešení by se aplikace zaměřila na zdokonalení sledování sportovních výkonů, efektivní komunikaci mezi členy týmu a usnadnění organizace událostí. Současně by měla být schopná umožňovat snadnější sledování a správu údajů o hráčích, což by klubům pomohlo lépe porozumět výkonnosti a potenciálu svých členů. V neposlední řadě by byl kladen důraz na optimalizaci procesu plánování a koordinace tréninků a utkání. Tímto způsobem by se aplikace snažila minimalizovat možné komplikace spojené s nedostatečnou organizací a komunikací v rámci sportovního týmu.

Nejprve bylo nutné vybrat vhodný programovací jazyk pro tento projekt. Svět informačních technologií nabízí nespočet technologických prostředků pro tvorbu webových aplikací. Mezi ně nepochybně patří programovací jazyk Java, který je velmi oblíbenou volbou mnoha programátorů, a díky tomu bylo jasné, že v případě

potřeby bude dostupná dobře zpracovaná dokumentace nebo jiné návody. Jako každý kvalitní program bude i tato aplikace potřebovat nástroj pro tvorbu přívětivého a hezkého uživatelského prostředí a pro tento úkol by nám samostatný programovací jazyk nepostačoval. Proto se musel najít určitý nástroj, který nám ulehčí tvorbu vzhledu stránky a je přímo programovaný pro tento úkol. V dnešní době máme k dispozici mnoha takzvaných frontendových prostředků, které slouží pro tvorbu uživatelských vzhledů aplikací. Nicméně bylo zapotřebí zajistit, aby tento framework správně fungoval se zvoleným programovacím jazykem a v nejlepším případě byl přímo doporučený pro tento jazyk. Mezi vhodné možnosti se zařadil i framework s názvem Vaadin. Je velmi jednoduché ho používat, jelikož transformuje základní HTML prvky na třídy v programovacím jazyku a umožňuje tak programátorům přímo přistupovat k frontendovým prvkům uvnitř zdrojového kódu pro backend. Programátor tak nepotřebuje mít oddělené soubory s backendem a frontendem, ale může vše tvořit dohromady uvnitř jedné třídy pohledu. To sebou samozřejmě nese značné výhody i nevýhody. Mezi nevýhody může patřit to, že programátor by měl dbát na přehlednost kódu a pravidelně okomentovávat jednotlivé metody, aby se v kódu vyznali i ostatní. Naprostá většina základních HTML prvků je zdarma k použití. Mezi placené, pak patří nadstavbové prvky jako je například cookies bar nebo grafy. Obecně se s Vaadinem pracuje velice intuitivně a je ideální volbou pro začátečníky. [3][4][5]

Dalším nutným úkolem k zajištění funkční aplikace, která bude schopna vykonávat dané cíle bylo třeba vytvořit ideální databázi, která bude mít na starost uchování dat zadávaná v aplikaci uživateli. Naprostá většina programovacích jazyků podporuje většinu databázových systémů. Pro výběr optimálního řešení pro tuto aplikaci bylo zapotřebí zhodnotit finanční a výkonnostní složky některých databázových systémů. Mezi vhodnými zástupci splňujícími tyto požadavky se nachází i relační databázový systém PostgreSQL. Ten je volně dostupný, výkonný a zároveň poskytuje dobrý uživatelský zážitek. Nabízí moderní uživatelské rozhraní nazývané PgAdmin, ale dá se používat i ve spojení s jiným programem. Stejně jako jiné databázové systémy umožňuje vytváření všeho, co jazyk SQL nabízí. Od základních selektů po složité triggerly nebo funkce. [8][9]

Pokud jde o výběr frameworku pro propojení a práci s databází, je potřeba dbát na jeho integrovatelnost s již už použitými technologiemi. Jednou z možností bylo zvolení frameworku Hibernate. Ten však nezapadal ideálně do konceptu vybraných prostředků pro tento projekt a představoval problémy pro budoucí funkčnost aplikace. V důsledku těchto obtíží se hledala adekvátní možnost. SpringBoot framework se ukazoval jako vhodná volba, zejména pro plynulé fungování s ostatními technologiemi určenými pro tvorbu této aplikace. Volba tohoto frameworku by nám měla zajistit plynulé a bezproblémové nasazení aplikace a zároveň ušetřit problémy s integrací při vývoji daného projektu.

Jednou z posledních věcí, kterou bylo zapotřebí zajistit, bylo vyřešení, na jaký hostingový server bude taková aplikace nahrána, aby mohla pracovat online. Je klíčové brát v úvahu několik kritérií, aby byl zajištěn optimální výkon, spolehlivost a bezpečnost. Jednou z nejdůležitějších věcí je, aby takový server umožňoval nasazení naší aplikace a podporoval technologie, které jsou k vývoji aplikace zapotřebí. Dále je podstatné zkoumat výkonnostní parametry serveru, včetně rychlosti odezvy, propustnosti a jeho schopnosti zvládat vysoký počet požadavků za sekundu (RPS). Dalším důležitým kritériem, které je potřeba vzít v úvahu při výběru hostingového serveru je jeho škálovatelnost. Neboli zajištění efektivního zpracování přírůstku uživatelů a zátěže. Aby byla data vložená uživateli chráněna před ukradením, je potřebné vzít do podvědomí i bezpečnostní hlediska. To může být například podpora pro HTTPS a ochrana před různými kybernetickými útoky. [17]

U aplikací je nezbytné, aby fungovaly co nejlépe pro uživatele a zajišťovaly jim skvělý uživatelský zážitek. Proto je při vývoji nutné brát v úvahu, na jakých platformách bude využívána primárně. Tato aplikace by měla být dělána přednostně pro rozhraní tabletů a laptopů, jelikož nepůjde o aplikaci, kterou by museli mít uživatelé dostupnou kdykoliv a odkudkoliv. Její hlavní využití pro ně bude, si v klidu naplánovat rozvrh následujících dní a zobrazit ho členům svého týmu. Úkolem aplikace je mít všechno přehledně pohromadě, od soupisky hráčů po seznam událostí v kalendáři, či exportování docházky na tréninky do excelu. Důležitým

úkolem bylo, aby tyto funkcionality dobře pracovaly na počítačích a tabletech, jelikož chceme mít přehledy k dispozici, ale nepotřebujeme je mít vždy po ruce. Hlavním prvkem aplikace bude, že si trenér bude moci zobrazit docházku jednou až dvakrát týdně nebo klidně i měsíčně. A díky tomu, že aplikace nebude poskytovat funkcionality, které by byly potřeba využívat v daný čas na daném místě, není tedy nezbytně nutné mít i mobilní verzi.

2 Cíl práce

Úkolem zvolené bakalářské práce je vytvořit aplikaci, která bude ulehčovat práci a dávat větší přehled sportovním trenérům v trénovanosti a připravenosti jejich svěřenců. Trenér bude mít přehled o hráčích v týmech a budou mu poskytnuta data, která ukazují, jak je na tom daný hráč fyzicky a jak vyniká v dalších florbalových vlastnostech.

Projekt bude primárně určen pro trenéry kolektivních sportů, podrobněji pak více zaměřen na florbal. Pro evidenci hráčovi výkonnosti bude mít k dispozici 6 atributů, charakterizujících jeho zlepšení. Střela, nahrávky, ovládání balónku, fyzicka, herní myšlení a mentalita. Po přihlášení se mu hodnoty u všech konkrétních atributů nastaví automaticky na nulu. Hráč by měl být schopen se přihlašovat na tréninky vypsané do kalendáře, které mají vždy určitý fokus. Těch může být přiřazeno k události i více, nicméně aplikace požaduje vybrat minimálně jeden. Postupem času, jak bude hráč procházet tréninky, se mu budou připisovat body do těchto atributů a trenér by tak měl mít přehled o jeho potencionálním zlepšení. Hráčovi bude vždy po absolvovaném tréninku připsán číselný koeficient do jeho statistik.

Výhodou této aplikace bude jednoduchost a přehlednost. Hráči budou rozděleni do týmů podle trenérova uvážení a ten o nich bude mít absolutní přehled. Zároveň každý uživatel bude mít svůj kalendář, kam se budou propisovat události z jejich týmů, tím je budou mít všechny přehledně zapsané v jednom kalendáři. Pokaždé, když se hráč nepřihlásí na trénink jim bude zpřístupněn chat u dané události, kam zapíšou svoji omluvenku. Tímto se zajistí, že se hráči nebudou zapomínat omlouvat a trenér bude přesně vědět s kým počítat a s kým ne.

V neposlední řadě trenér bude moci využít funkce pro export dat do excel souboru. Zde se mu zobrazí docházka či statistiky hráčů za daný časový termín, který si předvolí. V excelu si pak dále bude moci vytvářet grafy, statistiky nebo cokoliv, co uzná za vhodné.

3 Dostupná řešení

V této části si ukážeme existující aplikace, popis jejich fungování a ohodnocení jejich poskytovaných služeb.

3.1 *SidelineSports*

SidelineSports nabízí komplexní program pro evidenci a přehled výkonnosti hráčů nazývaný **XPS Network**. Toto je efektivní program pro přehled o trénovanosti hráčů, analytiku zápasů, ukládání tréninků a komunikaci s hráči.

XPS Network nabízí výběr z bohaté palety sportů od ledního hokeje po silový trénink. Na každý sport se zaměřuje z hlediska toho, co je u tohoto sportu podstatné. Je možné zadávat tělesné hodnoty hráčů a z toho následně dokáže vyčíst a zpracovat statistiky zdraví, jako např. BMI, tělesný tuk, zavodnění atd. Navíc umožňuje trenérům či fyzioterapeutům týmu mít dostatečně podrobný přehled o zraněních. Do programu lze zadávat různé druhy poranění a trenér má pak k dispozici výstupy o tom, zda je hráč připraven trénovat, nebo délce jejich rekonvalescence. [1][2]

Trenér může plánovat tréninky a připravovat si, jakou budou mít formu. Je umožněno tyto taktiky vkládat do tréninků. Z toho pak mohou čerpat potřebné informace o průběhu tréninkové jednotky i hráči. Lze evidovat zápasy, tréninky a různé další události do kalendáře. Zároveň s tím se mu generuje i docházka hráčů, kteří získávají také určité body trénovanosti. Trenér získává přehled o docházce, zaměření tréninků, výkonnosti hráčů díky grafům, které si může zobrazit. K tomu může navíc s jednotlivými hráči komunikovat skrz program. [1][2]

Velkou výhodou XPS Network je možnost zakoupit si extra licenci na program pro editaci videí a vytváření videorozborů pro hráče či pro sebe samotného. Vytvořené video, které si trenér natočí během tréninku nebo zápasu, si nahraje do programu a následně s ním může dále pracovat a editovat. XPS umožňuje využívat elektronické tužky, takže trenér může kreslit do videa a zvýrazňovat hráče nebo jejich pohyby

jednoduše jako na trenérské tabulce. Má k dispozici si video i sestříhat a upravit jeho vzhled. [1][2]

XPS Network je pravděpodobně nejvýkonnějším programem pro evidenci tréninků, rozborů, analýzu a komunikaci s hráči dostupný na internetu. Jeho nevýhoda ovšem spočívá v tom, že pokud chceme všechny tyto funkce využívat, musíme si zaplatit několik licencí. Pro spoustu florbalových klubů v České republice jsou tyto licence velký zásah do rozpočtu, který si nemohou dovolit. Velké kluby, hrající nejvyšší soutěže si určité licence dovolit mohou a tento program používají. Pro menší kluby, které chtějí růst, by taková aplikace byla nápomocnou rukou a mnoho práce by zjednodušila či ušetřila. Tím by se více mohly zaměřovat na výkonnostní stránku. [1][2]

3.2 SportLyzer

Další aplikaci, kterou si tady představíme je řešení s názvem SportLyzer. Funkčnostmi, které nabízí, se asi nejvíce bude podobat programu XPS Network a tím pádem se podobá předchozímu programu i v možnostech dostupnosti pro kluby. Toto řešení je dostupné ve verzi webu i jako mobilní aplikace. [15]

Trenérům nabízí možnost evidovat docházku velmi jednoduše podle toho, jak se jednotliví hráči zapisují u událostí. Trenér si tak nemusí po každém tréninku či zápasu pamatovat, jaký hráč přišel či nikoliv, ale všechno záleží na hráčích a jejich zodpovědnosti. Velkou výhodou této aplikace je, že se hráči do ní nemusí registrovat. Stačí, když si trenér nechá vygenerovat jedinečné odkazy, které hráčům rozešle. Pokaždé, když hráč tento odkaz rozklikne, uvidí své nadcházející události a bude se u nich moci zaregistrovat, čímž se zapíše trenérovi do docházky. Další výhodou je možnost sdílení docházky s jinými trenéry. V momentě, kdy jeden trenér nemůže udělat docházku týmu, tak kdokoliv jiný, kdo má přístup, toto může udělat za něj. Trenér má k dispozici i různé filtrace docházky, podle typů událostí, nebo podle týmů, které má na starost. Důležité k tomuto kroku ale je, aby si to dobře prvotně nastavil. [15]

Co se týče hlediska hráčů, umožňuje jim mít přehled o důležitých událostech na dosah ruky. Hráči navíc mohou dávat zpětnou vazbu k tréninkům, jak se jim líbil, či jak byl náročný. Toto jsou velmi cenné informace pro trenéry, kteří díky tomu mohou optimalizovat připravenost a výkonnost hráčů a věnovat tomu zvláštní pozornost. Pro všechny uživatele v týmu je implementovaný jeden sdílený kalendář, který jim umožňuje mít přehled o všech událostech na jednom místě a přehledně. [15]

V neposlední řadě, trenér může podobně jako u aplikace XPS Network vytvářet analýzy a plány tréninků s hodnocením od hráčů. Evidovat si fyzickou připravenost, zranění či nemoci hráčů a udržovat si přehled o jejich testování a hodnotit je vlastními poznámkami a čísly. [15]

Jak bylo zmíněné na začátku, tato aplikace se funkcemi velmi podobá XPS Network. Nenabízí možnost video-analyzeru, ovšem přehledností, umožněním evidence docházky a dalšími funkcemi je dobrým konkurentem dříve zmíněné aplikace XPS. Přínosem této aplikace je, že se hráči nemusí do ní přímo registrovat do, ale mohou ji využívat. Jediným, kdo u této aplikace potřebuje mít registraci je trenér, který pak zasílá odkazy na své týmy v aplikaci hráčům. Co se týče cenové dostupnosti, nabízí verzi zdarma, ovšem v té chybí základní kameny aplikace, které jsme si tady popsali. Plný balíček pak vychází okolo 1300 Kč/měsíc nebo 11 500 Kč/rok. Pro větší kluby cena této licence nejspíše nebude problém, ovšem pro kluby s malým rozpočtem je to částka, kterou si častokrát nemohou dovolit. [15]

3.3 iTreneo

ITreneo je další aplikace, která umožňuje online pracovat se sportovci, zapisovat o nich podrobné informace a vytvářet jim individuální plány. Je primárně připravena pro osobní trenéry, kteří pracují s jednotlivci, ale dá se jednoduše využít i na větší skupinu sportovců. Pomáhá vidět či odbourávat chyby sportovců a zlepšovat jejich výkony. [6]

Jako trenér máte k dispozici možnosti si vytvořit knihovnu určitých cvičení či tréninků a vytvářet z nich jednotlivé tréninkové programy. Ty i další jednotlivá cvičení, pak můžete jednoduše nasdílet svým klientům do kalendáře. Díky sdílenému kalendáři budete mít stále přehled o výkonech klientů, a dokonce i o jejich jiné aktivitě, kterou dělali mimo plán. Výstupem ze zadaných údajů je evidence o statistikách klientů, jako je například počet odcvičených aktivit, celkový čas strávený cvičeními nebo celková odběhnutá či odcvičená vzdálenost. Všechny tyto záznamy jsou dohledatelné v rámci týdne, měsíce či roku. [6]

Velkou výhodou této aplikace je možnost využití interního chatu se svým trenérem přímo v aplikaci. Není nutné si proto zahrnovat svůj vlastní komunikační kanál skupinami či zprávami se svými trenéry. Díky tomu můžete tuto kategorii oddělit přímo do aplikace zaměřené na váš sportovní výkon. [6]

Jako každá jiná má tato aplikace své přednosti i nedostatky. Celkově nabízí mnoho možností, jak pracovat s hráči. Nicméně, jak bylo uvedeno výše, je primárně zaměřena na individuální tréninky, tudíž pro kolektivní sporty jako je florbal postrádá určité funkce. Přednost určitě spočívá v tom, že aplikaci můžete využívat do velké části zdarma. Pokud opravdu potřebujete tuto aplikaci využít na práci s týmem, vychází relativně dostupně a po finanční stránce vychází levněji než např. XPS Network. [6]

3.4 EOS club zone

V pořadí čtvrtou aplikací, kterou si tu představíme je aplikace vyvinutá společností EOS. Ta se přímo zaměřuje na hledání ideálního řešení pro kluby. Ať už potřebují vytvořit webové stránky nebo právě aplikaci v rámci klubové sekce, která má pomáhat s docházkou a ulehčovat trenérům a hráčům práci. Pro spoustu sportovních klubů je toto ideální řešení, a proto tento program využívají. Klubová sekce může být buď ve formě webu či jako mobilní aplikace. [7]

Co se týče funkcí, je oproti předchozím aplikacím více zaměřena na management a organizaci uvnitř klubu než na sportovní výkony hráčů. Umožňuje tvořit týmy

a přidávat do nich hráče. Samozřejmě nechybí možnost vytváření událostí uvnitř týmů, jako jsou například zápasy či tréninky. Hlavní výhodou EOS club zone je, že můžete nahrávat a mít kontrolu o různých dokumentech, které obdržíte od hráčů. Běžnými dokumenty, které lze nahrát jsou například zdravotní prohlídky, smlouvy o přestupech a spoustu dalšího. Zároveň máte implementovanou funkci vnitřního emailu, což vám dost ulehčuje práci v komunikaci s rodiči, vedením klubu či hráči samotnými. Jedna z předností aplikace spočívá v tom, že si zde můžete nahrát i bankovní konta rodičů, či jiné důležité bankovní účty, což vám usnadní práci při vybírání a převodu peněz. [7]

Tato aplikace řeší trochu jiný problém než předchozí tři. XPS Network, SportLyter a iTreneo jsou více orientovány na sportovní a výkonnostní stránku členů. Zaměřují se na statistiky hráčů a jejich výkonnost za určité období, zatímco EOS club zone se tímto směrem neubírá a žádná výkonnost hráčů se v ní neměří. Na druhou stranu EOS lépe řeší problémy s administrací a fungováním členů uvnitř klubu. Všechny čtyři aplikace nabízejí možnost evidence tréninků a jiných událostí uvnitř klubu. [7]

Pokud jde o koncept a vývoj aplikace TrWolf, bylo rozhodnuto postupovat tak, že se využijí pozitivní aspekty každého výše uvedeného softwaru. Bylo mířeno na takové řešení, které je pro kluby nejen výhodné, ale také cenově přijatelné. Tato kombinace byla sloučena do jedné aplikace s důrazem na jednoduchost a přehlednost, což bylo stanoveno jako prioritní. V následující kapitole budou popsány přípravy na vývoj aplikace a její funkce.

4 Příprava vývoje aplikace

Ještě před samotným vytvářením aplikace bylo nezbytné si navrhnout plán, jak bude aplikace fungovat. S tím je samozřejmě také spojené vybírání vhodných vývojových nástrojů a jazyka, ve kterém bude aplikace programována.

Jako první bylo třeba rozhodnout, zda bude lepší řešení mobilní, počítačová či univerzální aplikace. Na výběr bylo mezi třemi možnostmi. Hybridní, nativní nebo webovou aplikaci. Vzhledem k tomu, pro koho a za jakým účelem byla aplikace tvořena, nakonec po zvážení kladů a záporů byla zvolena poslední varianta čili webová aplikace. Podle posudku prvních dvou možností bylo usouzeno, že by nepřinášely pro kluby přidanou hodnotu z jejich hlavních kladů. Tím by nebyly tak dobře použitelné pro účely tohoto programu jako právě webová aplikace. V tom případě by byl výsledek pro splnění funkcí aplikace spíše horší. Navíc by tato řešení byla pouze pro mobilní verzi. Vybrána byla tedy webová aplikace splňující očekávání do vývoje a hlavní cíl projektu. Více podrobné porovnání těchto možností je popsáno v následující kapitole s názvem Volba druhu aplikace.

Základem celé aplikace je stabilní databáze. Ta je určená k uchovávání dat, které uživatelé aplikace buď obdrží či posílají. Bylo zapotřebí najít vhodný nástroj pro práci s databázovým systémem a pro tuto úlohu byl zvolen open-source databázový systém jménem PostgreSQL. Návrh databáze se vytvářel v programu Enterprise Architect, který slouží pro tvoření UML diagramů systémů. Je možné v něm vytvářet věci od modelování databáze, aplikace či procesů týkajících se běžného života. Tento software umožnil namodelovat pomocí svých nástrojů přehledný diagram o tom, jak budou databázové tabulky propojeny. Po dokončení celého diagramu dokáže vygenerovat SQL script, který po spuštění v databázovém systému vytvoří odpovídající databázi. Pro naše účely byl diagram namodelován pro systém PostgreSQL. Jelikož se aplikace později rozšiřovala o nové funkcionality, bylo za potřebí taktéž rozvést odpovídající databázi.

Dalším krokem ve vývoji bylo vytvoření projektu ve vybraném programovacím jazyku a přidání dalších nástrojů, které byly nápomocny v programování celkové aplikace. Z hlediska zhodnocení programovacích jazyků, které se nejčastěji používají pro vývoj webových aplikací byla ve finální fázi vybrána Java, jelikož je to jazyk oblíbený mezi programátory a taktéž má dobré reference na vývoj webových aplikací. Pro vývoj frontendu byl vybrán framework Vaadin, který je velmi dobře integrovatelný s jazykem Java. Po jeho přidání do projektu vám zpřístupní použití veškerých svých volně dostupných komponent pro vytváření struktury HTML vaší webové aplikace. Jeho připojení do projektu můžete zajistit pomocí některých nástrojů pro přidávání externích balíčků do projektu. Jedním z těchto nástrojů je i Maven, díky kterému byl v našem případě připojen Vaadin do aplikace. Aby ji bylo možno vůbec začít vytvářet, muselo se zajistit propojení samotné aplikace a databáze. O to se stará framework SpringBoot ve spojení s JPA (Java Persistence API), což je nástroj, který umožňuje objektově relační mapování entit s tabulkami v databázi. Díky těmto nástrojům je umožněno vygenerování odpovídajících entit do Java aplikace ve formě tříd. V tomto momentě bylo vše nastavené tak, aby se mohlo pokračovat do samotného programování funkcí v aplikaci.

První z věcí, která byla řešena bylo přihlašování a registrace do aplikace. Bylo nutné zajistit bezpečné ukládání hesel v databázi. K tomu je využita třída *BCryptPasswordEncoder* z knihovny Springboot Security. Ta je zodpovědná za proces bezpečného zahashování a ukládání uživatelova hesla do databáze. Je použita při vytváření nového uživatele a chrání tak jeho heslo před odhalením od potencionálních útočníků. Uživatelé se registrují pomocí svých osobních údajů, jako je jméno a email a taktéž při registraci vybírají, z jakého přístupu budou aplikaci využívat (*hráč/trenér*). Této informace je dále v aplikaci využíváno na implementaci toho, co může, jaký uživatel dělat a k jakým informacím se dostat.

Díky tomu, že byla splněna úloha přihlašování, jsme mohli dále získávat informace o aktuálně přihlášeném uživateli a pracovat s ním v aplikaci. V ní je vytvořeno několik pohledů za pomoci Vaadin frameworku a jeho komponent, které zobrazují

různé informace splňující účely této aplikace. Více o tom, co se v jednotlivých pohledech nachází, si řekneme v pozdější kapitole Frontend.

Jelikož je požadováno, aby trenéři měli možnost dále s daty z aplikace pracovat podle vlastního uvážení, bylo implementováno stahování excel souborů. Tuto funkcionalitu, díky balíčku od Apache, zajišťuje vytvořená třída *ExcelExporter* se třemi předpřipravenými šablonami. Díky této knihovně vzniká možnost si vygenerovat .xlsx soubor s docházkou za období, které si lze zvolit. Toto ovšem není jediná věc, která jde v aplikaci generovat do excel souboru. Dalšími možnostmi využití je generování docházky hráčů za určitou událost. Třetí možností, je pak stažení excel souboru s body za vaši docházku na různé události neboli celkový souhrn toho, jak a kdy jste dané body získali.

Po naprogramování aplikace a finálních úpravách bylo potřeba najít vhodného webového hostitele, který dokáže poskytnout služby k nahrání aplikace na webový server. Pro tuto úlohu bylo zvoleno služby Heroku, což je cloudová platforma, která umožňuje vaši aplikaci spravovat na webu. Jako první bylo třeba si založit účet na *heroku.com*. Poté už následovalo nasazení aplikace na jejich servery. Nejprve byl vygenerován spustitelný .jar soubor aplikace, který se poté přes příkazy v terminálu nahrál na webový server. Taktéž bylo důležité přidat doplněk pro správu databáze PostgreSQL, které je v programu využíváno a Heroku ho nabízí. První pokus o nahrání aplikace se nepovedl z důvodu kompatibility webového serveru s JDK. Přestože byla aplikace původně programována v Javě 19, později bylo objeveno že Heroku server nepodporuje právě tuto verzi. Proto musel být projekt poupraven na verzi Javy 17. Posledním úkolem pro úspěšné online fungování aplikace bylo nutné naplnit databázi na serveru potřebnými daty. Po tomto kroku byla aplikace připravena na používání. [18]

4.1 Volba druhu aplikace

Nyní si více popíšeme, jaké jsou rozdíly, přednosti a nevýhody třech hlavních druhů aplikací, které byly zvažovány při návrhu tohoto projektu.

4.1.1 Nativní aplikace

Nativní a hybridní aplikace jsou primárně určeny pouze na mobilní verzi. Nativní aplikace je taková, která je vyvíjena pouze pro určitou mobilní platformu. To přináší svoje benefity, ale také negativa, jako je např. cena nebo čas strávený vývojem. V praxi to většinou chodí tak, že pokud chcete aplikaci, která bude schopná pracovat na obou nejznámějších mobilní platformách (iOS a Android), budete potřebovat 2 vývojové týmy pro každou z nich. To zákazníka samozřejmě stojí více peněz, jelikož je potřeba zaplatit více lidí. Díky tomu, že je vyvíjena pro specifickou verzi, přináší jednu hlavní výhodu a tou je přístup k hardwaru zařízení. To přináší možnost konat složité grafické či početní operace s rychlou odezvou. Pro vývoj aplikace na platformu Android se používá Java a pro iOS jazyk Objective-C. [10][11]

4.1.2 Hybridní aplikace

Hybridní verze je stejně jako nativní určena pro mobilní zařízení. Od předchozí verze je hlavní rozdíl v tom, že pro obě platformy se vyvíjí aplikace ve stejném prostředí. Díky tomu jsou tyto aplikace na trhu levnější a obecně rychlejší na vývoj. Tato verze využívá znalostí dvou druhých možností a je svým způsobem jejich kombinací. Z toho, že není přímo určena pro jistou platformu vyplývá, že bude náročné úkony na procesor řešit pomaleji než aplikace nativní. Navzdory tomu je v poslední době velmi populární a postupně se ustupuje od vývoje nativních aplikací k hybridním. [10][11]

4.1.3 Webové aplikace

Webová aplikace je svým způsobem upravená webová stránka. Při jejím vývoji se používají stejné technologie jako při tvorbě webu, jako je např. HTML, CSS nebo JavaScript. Hlavním rozdílem od webu je skutečnost, že se většinou potřebuje více pracovat s backendem. Přednost těchto aplikací spočívá v tom, že se mohou jednoduše zakomponovat do existujícího webu, takže je to velmi oblíbené řešení pro firmy, které v dnešní době hledají vývoj aplikací pro vlastní účely. Další nespornou výhodou je, že jsou dostupné z jakéhokoliv zařízení, i když se musí počítat s responzivitou. [11][12]

Pro tento projekt byla vybrána webová aplikace především ze dvou důvodů. Prvním důvodem bylo, že aplikace musí být spustitelná na všech typech zařízení. Především pak na počítačové platformě, která byla pro tuto aplikaci stěžejní. Tím, že aplikace měla být taktéž přizpůsobená napříč všemi věkovými kategoriemi se usoudilo pro formu webové aplikace, která nám poskytuje snadnou ovladatelnost a dobrou škálovatelnost.

4.2 Návrh aplikace

Fungování programu bylo navrženo na jednoduché databázi, kde spolu komunikuje několik tabulek. Aplikace je navržena tak, aby byla jednoduchá na používání, přitom ale pokryla většinu problémů, týkajících se problémovosti docházky a přehlednosti událostí. Uživatelské rozhraní je navrženo tak, aby bylo snadno ovladatelné a bylo jasné, co má uživatel dělat. Toto zajišťuje, že je aplikace velice intuitivní a přímočará. Grafické rozhraní aplikace bylo uděláno pomocí frameworku do Javy nazývaný Vaadin, který používá prvky jazyka CSS a HTML. Pomocí tohoto frameworku se dali vytvořit základní HTML prvky se kterými se pracuje v aplikaci. Pomocí Javy společně s frameworkem SpringBoot a Hibernate se ukládají data do databáze PostgreSQL. Zároveň se z databáze načítají data zpět do aplikace a pracuje se s nimi pomocí již zmíněných HTML prvků.

4.2.1 Vaadin framework

Vaadin je frontendový nástroj neboli framework, který je přizpůsobený k vývoji webových aplikací založených na Java backendu. Byl vyvíjen od roku dva tisíce skupinou developerů s hlavním sídlem ve Finsku. Vaadin je finské slovo znamenající sob a toto zvíře je hlavním logem organizace. Jejich cílem je vytvářet framework, který bude intuitivní a pomůže programátorům tvořit webové aplikace s příjemným uživatelským rozhraním pro jejich uživatele. [3][4]

Vaadin pracuje společně se SpringBootem a má na starosti frontend. SpringBoot pak společně s Javou zajišťuje backend. Více si o SpringBootu a Javě řekneme v dalších kapitolách Java, Spring a Springboot. [3][4]

Přidat Vaadin do svého Java projektu lze několika způsoby. Nejjednodušší je si na jejich stránkách nastavit počáteční projekt a ten si pak stáhnout. Balíček frameworku se automaticky přidá pomocí nástroje pro správu projektů – Maven. Druhou možností je přidat balíček pomocí nástroje Gradle. Obě možnosti fungují podobně a velmi spolehlivě. Je jen na vývojáři, co mu vyhovuje více. V tomto projektu byl vybrán nástroj Maven a pomocí souboru pom.xml byly přidány potřebné dependencies. [3][4]

Po tomto kroku se přidá do projektu balíček Vaadin, který obsahuje mnoho HTML komponentů. S těmito prvky, pak umožňuje programátorovi pracovat stejně jako s třídami v Javě. Díky tomuto přístupu se stal velmi oblíbeným nástrojem pro vytváření frontendu aplikací. Každá tato komponenta je velmi jednoduše dostupná ve zdrojovém kódu a díky tomu vám umožňuje velmi rychle a jednoduše tvořit uživatelské rozhraní. Vaadin také umožňuje upravovat CSS charakteristiky těchto komponentů přímo pomocí funkcí v Javě. Může se tak upravovat jejich velikost či nastavovat předdefinovaný styl pro komponenty. Pokud potřebujete přímo upravovat složitější věci v CSS, označíte si komponentu classou, kterou pak můžete v odděleném souboru .css upravovat dále. [3][4]

Díky těmto funkcionalitám se Vaadin stal nepostradatelným nástrojem pro tvorbu moderních webových aplikací a šetří tvůrcům webových aplikací spoustu času i energie.

4.2.2 Java

Java je programovací jazyk, který je často využíván vývojáři softwaru a poskytovateli internetových služeb k vytváření webových či desktopových aplikací.

Tento programovací jazyk je založený na objektovém programování. To znamená, že se programuje pomocí objektů neboli tříd, které v sobě uchovávají svoje atributy a metody a zároveň je každá instance tohoto objektu může volat a používat. Jeho výhodou je, že byl vytvořen s co nejmenšími implementačními požadavky pro maximální přenositelnost. Ku příkladu integer je vždy 32bitové celé číslo se znaménkem dvojkového doplňku, bez ohledu na architekturu procesoru použitého ke spuštění aplikace. [5]

Java byla vytvořena v roce devatenáct set devadesát pět firmou Sun Microsystems. Tento programovací jazyk se velice rozšířil a je jeden z nejvíce používaných programovacích jazyků pro vytváření webových aplikací. Vývojáři, kteří programují

webové aplikace v Javě, využívají možnosti, že kód vytváří pouze jednou a jsou schopni ho reprodukovat. Tento jazyk ovšem není používán pouze pro webové aplikace, ale můžete s ním tvořit od jednoduchých desktopových aplikací až po složité grafické hry. [5]

Java sama o sobě je pouze programovacím jazykem. K tomu, abychom mohli zapisovat příkazy Javy, budeme potřebovat JDK, což je z anglického Java Development Kit. JDK je soubor nástrojů zprostředkujících kompilaci a spouštění Java aplikací. Obsahuje knihovny, kompilátor a další nástroje potřebné ke psaní a spouštění zdrojového kódu. Ten se zpravidla zapisuje do textových editorů uzpůsobených pro programování aplikací. Nejznámější editory, které se využívají při práci s Javou jsou pak např. NetBeans, Eclipse, Visual Studio Code či IntelliJ IDEA, o které si povíme trochu více v další kapitole. [5]

4.2.3 IntelliJ IDEA

IntelliJ IDEA z anglického IDE (Integrated Development Environment) je integrované vývojové prostředí vyvinuté firmou JetBrains. Je velmi oblíbené mezi programátory a využívá se pro práci s programovacími jazyky Java, Kotlin nebo Groovy. [16]

Díky své výkonnosti a chytrosti napomáhá zdrojový kód držet přehledný a dobře zformátovaný. Dokáže odhalovat chyby, upozorňovat na možná nebezpečí či špatné přístupy nebo napovídat programátorům s metodami, které by mohly být použity v kódu. [16]

Každý správný programátor jistě ví, jak potřebné je mít zálohovaný kód a určitě se mu hodí, když může sledovat pravidelné změny v kódu, či se vracet ke starým funkčním verzím programu. K tomu všemu slouží Version Control System neboli verzovací systém (VCS). Samotná platforma IntelliJ IDEA umožňuje integrovat váš kód s repositářem na některém z verzovacích systémů (GitLab, GitHub, SVN atd.) a pravidelně provádět commity, což zahrnuje komentování a ukládání změn ve vašem programu. [16]

Tento editor vám taktéž umožňuje měnit vzhled prostředí. Ať už potřebujete větší kontrast mezi barvami, temné či světlé prostředí nebo barevně odlišit jednotlivé projekty, toto všechno vám IntelliJ poskytne a umožní. [16]

4.2.4 PostgreSQL

PostgreSQL je volně dostupný relační databázový systém pro práci s databázovými tabulkami. Relační databázový systém znamená, že na rozdíl od jiných typů databázových systémů nevyužívá rodičovských vztahů mezi jednotlivými položkami databáze, ale jeho data jsou organizována do uspořádaných n-tic. Jednotlivé n-tice pak představují řádky tabulky a tím vzniká, že každý jednotlivý záznam tvoří řádek tabulky a položka pak daný sloupec. Je založený na databázovém jazyku SQL a díky tomu umožňuje tvořit SQL dotazy, procedury či triggerly a vytvářet příkazy neboli queries pro tabulky. Byl vyvinut na univerzitě v Berkeley v Kalifornii v devadesátých letech dvacátého století a je jedním z nejoblíbenějších systémů pro práci s databázemi. Svou popularitu si získal především výhodami, které nabízí. Ať už je řeč z pohledu vývojáře, pro kterého zprostředkovává stabilní systém pro práci s databázemi. Nebo pro začátečníka, který se učí SQL a hodí se mu mít volně dostupný výkonný program na dosah ruky. [8]

Mezi jeho hlavní výhodu samozřejmě patří jeho volná dostupnost. Program si můžete zadarmo stáhnout, nainstalovat a pracovat na svých projektech. Díky tomu, že používá jazyk SQL je velmi jednoduché se s tímto programem naučit a pokud již máte nějaké zkušenosti z jiného databázového systému, o to je to jednodušší. Je taktéž velmi stabilním programem, který málokdy spadne a je určený pro práci s mnoha programovacími jazyky. Proto nemusíte mít strach, že když si zvolíte právě PostgreSQL, budete mít problémy s integritou programovacího jazyka a přecházet na některý jiný. Na webových stránkách můžete najít velmi dobře zpracovanou dokumentaci programu k použití, proto bude pro vás velmi jednoduché vyřešit problémy, se kterými si nebudete vědět rady. A v neposlední řadě vývojáři tento systém neustále aktualizují, proto se drží stále při nových IT trendech a je kompatibilní s většinou nových věcí. [8]

Stejně jako Java, tak i PostgreSQL potřebuje svůj textový editor. Je několik možností, které můžete použít. Např. Psql, DBeaver, Navicat nebo PgAdmin a právě tento nástroj si teď představíme. [8][9]

PgAdmin je volně dostupný grafický nástroj určený pro správu PostgreSQL databází. Je součástí instalačního balíčku PostgreSQL a je kompatibilní se všemi verzemi operačních systémů. Může být spuštěn buď formou desktopové aplikace nebo webu. Jeho velkou výhodou je, že má velmi přehledné a příjemné grafické rozhraní a díky tomu pak můžete spravovat svou databázi velmi jednoduše pomocí pár kliknutí. Svou jednoduchostí vás nenutí si pamatovat či složitě vyhledávat query. Spoustu těchto věcí lze provést pomocí myši. Například přidávání nových záznamů do tabulek můžete nastavit tím, že kliknete na tabulku, do její kolonky vyplníte údaje a jednoduše uložíte tento záznam namísto psaní query. Díky hezky zpracovanému grafickému rozhraní se vám bude dobře pracovat a budete mít přehled o všech důležitých věcech uložených ve vaší databázi. [9]

4.2.5 Spring a SpringBoot

Spring je rodina frameworků, které podporují vývoj webových aplikací psaných v jazyce Java. V projektu je pak využíváno jeho rozšíření, které se nazývá SpringBoot. Rodina Spring nabízí mnoho funkcionalit, jako je perzistence dat, zabezpečení či integrace s jinými systémy. V této aplikaci se stará o zabezpečení a je nápomocný s propojením dat z aplikace do databáze. [13][14]

SpringBoot je velmi populární nástroj pro vývoj, jelikož poskytuje většinu řešení pro problémy spojené s vývojem webové aplikace a je velmi jednoduchý na používání. Kdybyste chtěli sami implementovat logiku, která je zapotřebí k fungování webové aplikace jako jsou např. HTTP Requesty, museli byste naprogramovat komplexní program, který se o toto bude starat. Pokud použijete SpringBoot či jiný framework, řeší nejen tuto otázku za vás a vy ho můžete použít ve vašem projektu jednoduchým stažením závislostí přes Maven či Gradle. Po stažení těchto závislostí můžeme

pracovat s funkcemi, které SpringBoot nabízí. Poté už jen stačí používat anotace, které vašemu programu řeknou, že se má používat určitý framework. [13][14]

5 Vývoj aplikace

V této části bude přesně popsáno, jak byla aplikace vyvíjena, chyby při tvorbě a jaké nastaly problémy. Taktéž se probere komunikace s databází, jak se vytvářelo rozhraní a v poslední řadě výstupy, které se od aplikace očekávaly.

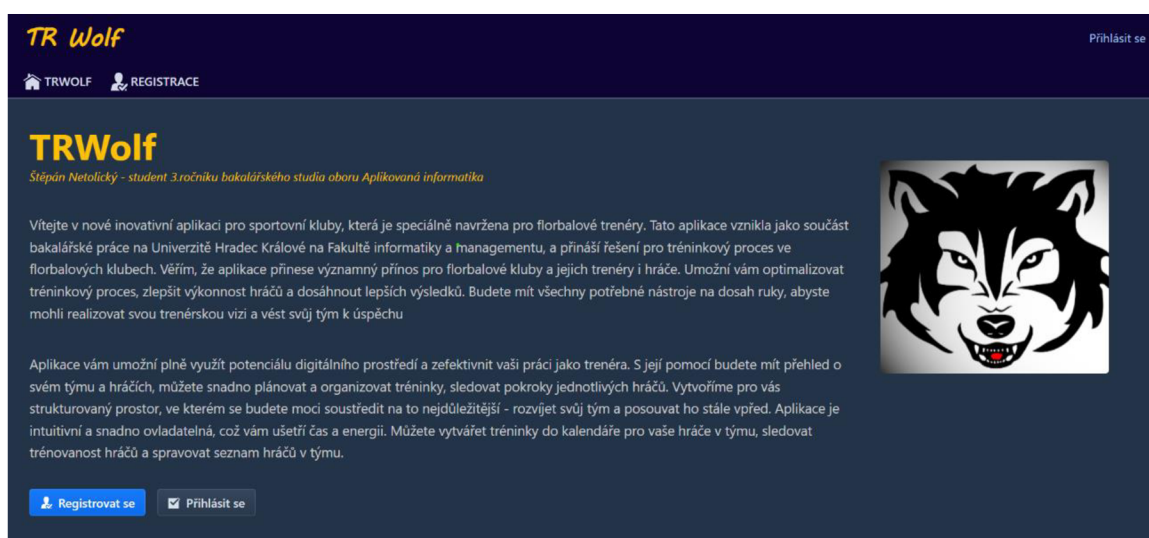
5.1 Programování aplikace

Jako u většiny aplikací, tak obzvlášť webové aplikace potřebují vyvinout svůj frontend, tak i backend. Frontend v tomto projektu nejenže zajišťuje vzhled aplikace, ale také má na starosti informování uživatele o potvrzených akcích, které se úspěšně či neúspěšně provedly. Je úzce spjat s backendem a využívá jeho informací k předání grafické informace uživateli. Byl kladen důraz na uživatelskou přívětivost a rychlost odezvy při komunikaci s backendem, tak aby aplikace uživatele neodváděla od hlavních funkcionalit, které potřebuje k dosažení účelů. Proto je vzhled aplikace poněkud jednoduchý, ale zároveň intuitivní pro uživatele.

Hlavní prací backendu je v tomto případě komunikace s databází, ať už jde o ukládání dat do databáze, předání dat frontendu, aby mohl zobrazit informace uživateli, či aktualizace různých datových struktur, které se v aplikaci používají. Další úlohou backendu je zpracování excel souboru podle specifik zadaných uživatelem a postarání se o jeho správné vyexportování, aby si ho uživatel mohl stáhnout z webu. Nahrání a uložení cesty nahrávaného PDF souboru k aplikaci je jednou z jeho dalších funkcí. Významnou úlohou backendu je pak práce s uživateli, ať už se jedná o správu jejich údajů nebo kontrolu dat při přihlašování a registrování a bezpečné zakódování hesel zadávaných uživateli. Za stěžejní funkci backendu můžeme označit funkci, která pravidelně provádí kontrolu ukončení událostí a přidává body přihlášeným uživatelům k dané události. Celkově bylo cílem během vývoje aplikace dosažení vysoké úrovně funkčnosti, bezpečnosti a uživatelské spokojenosti.

5.1.1 Frontend

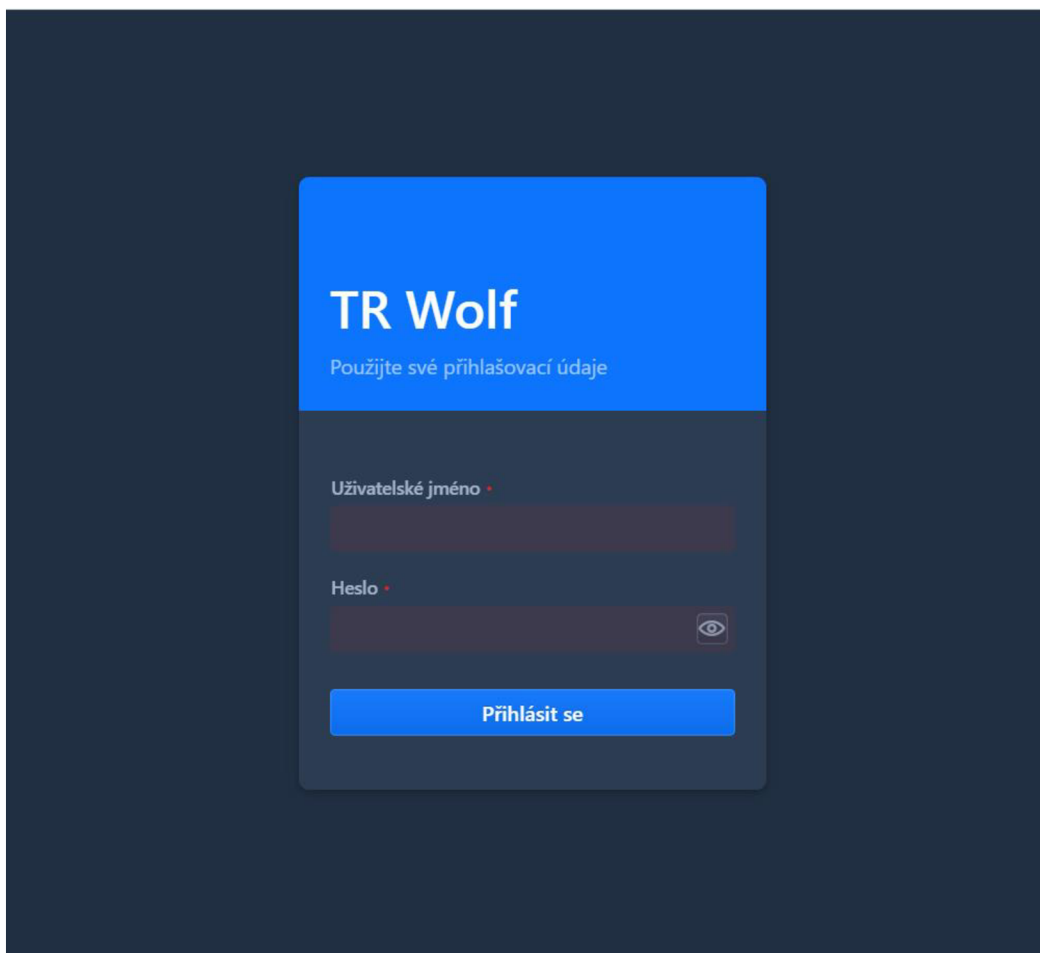
Celkový vzhled aplikace, jak už bylo zmíněno výše, byl orientován na to, aby byl intuitivní, přívětivý pro uživatele a zároveň poskytoval rychlou odezvu. Toto nám zajišťuje framework Vaadin, který pracuje s HTML prvky jako vytvořenými třídami v Javě. Díky tomu bylo možno jednoduše přistupovat k těmto prvkům přímo ve zdrojovém kódu a upravovat jim základní CSS charakteristiky bez potřeby toho, aby se používal další oddělený HTML soubor. HTML a CSS za pomoci Vaadinu tvoří celkovou vzhledovou strukturu aplikace. Jelikož Vaadin nabízí upravování HTML prvků pouze v základu, byl vygenerován i oddělený .css soubor, kde byly upravovány další atributy HTML prvků. K těm se přistupuje jednoduše, označením instance HTML prvku ve zdrojovém kódu pomocí funkce „`addClassName(String s)`“. Po označení daného prvku touto funkcí a pojmenování prvku Stringem pro třídu se pak jednoduše odkážete na tento prvek přes CSS soubor. Tohoto principu bylo využito v celém procesu tvorby frontendu.



Obr. 1 Domovská stránka aplikace (zdroj vlastní)

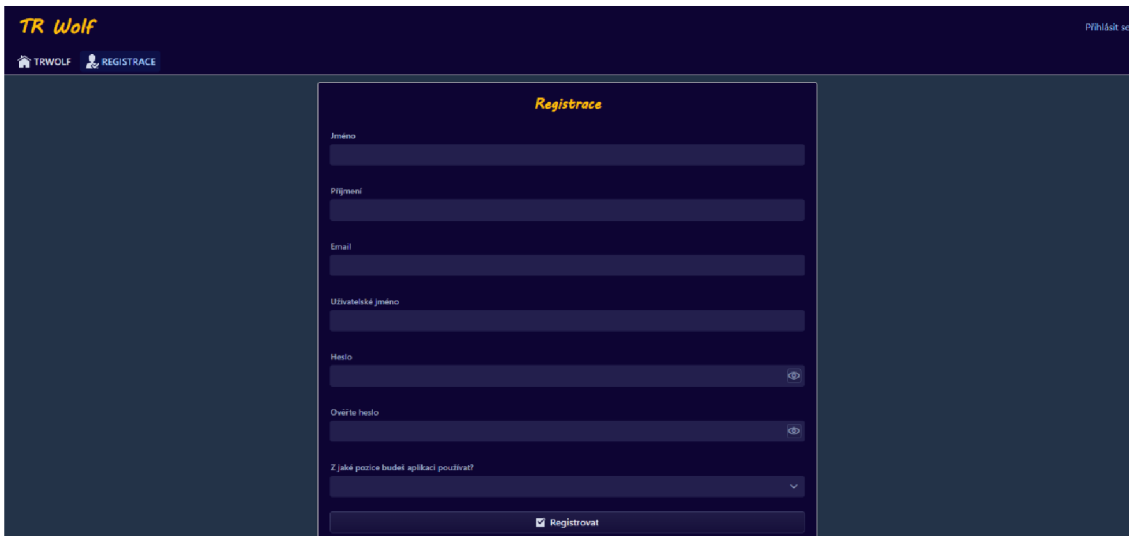
První stránku, kterou uživatel uvidí při načtení aplikace, je domovská stránka. Ta je zobrazena na obrázku číslo 1. V našem případě se jedná o obyčejnou stránku s texty o tom, pro koho je tato aplikace určena a v čem by měla splňovat uživatelské očekávání. Přes tuto stránku se uživatel může díky tlačítkům prokliknout na přihlášení nebo v případě nového uživatele přejít na registraci. Na každou variantu

je možnost se dostat dvěma způsoby, pomocí definovaných tlačítek pro tyto dvě varianty.



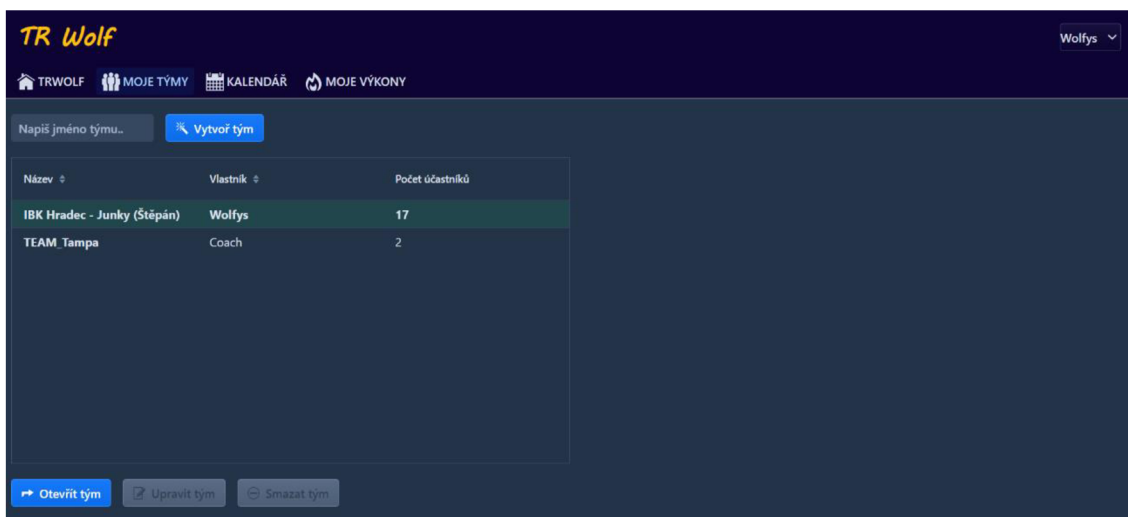
Obr. 2 Přihlašovací stránka aplikace (zdroj vlastní)

V případě toho, že je uživatel už zaregistrovaný do aplikace se může přihlásit. To provede kliknutím na jedno z tlačítek „Přihlásit se“, které jsou viditelné na obrázku 1. Tyto tlačítka ho odkáží na stránku s přihlašováním do aplikace, viz. obrázek 2. Ta pracuje pouze s jedním oknem, kde se nachází dva inputy pro zadání uživatelského jména a zvoleného hesla při registrování. Pokud uživatel zadá správné údaje, které zadával při registrování, tlačítko „Log in“ ho odkáže zpátky na úvodní stránku. Už ale jako přihlášeného uživatele. To znamená, že mu zmizí tlačítka „Přihlásit se“ a „Registrovat se“ a v pravém horním rohu bude mít svoje uživatelské jméno, pod kterým se skrývají funkce pro správu uživatelského účtu. Pokud ovšem zadá špatné přihlašovací údaje, objeví se mu chybová hláška a bude vyzván k opravě přihlášení.



Obr. 3 Stránka s registrací nového uživatele (zdroj vlastní)

Pokud je ovšem novým uživatelem aplikace, musí se nejprve zaregistrovat. To provede kliknutím na tlačítko „*Registrovat se*“ nebo kliknutím na záložku „*Registrace*“, kterou můžeme vidět na obrázku číslo 1. Tato funkce nás odkáže na stránku s formulářem pro registraci. Registrační formulář je ukázán na obrázku 3. V tomto formuláři je několik input polí, kam uživatel vyplní svoje údaje. Za zmínku stojí pole s ověřením hesla. To je nastaveno tak, že pokud se neshoduje s heslem zadaným v poli heslo, nedovolí uživateli se zaregistrovat a upozorní ho na vzniklý problém chybovou hláškou. Druhé pole, které stojí za zmínku je selectbox, kde uživatel zadává, z jaké pozice bude aplikaci používat. Toto pole mu nabídne ze dvou možností, hráč nebo trenér. Po zvolení jedné z možností, je pak při úspěšné registraci přiřazena do databáze uživateli hodnota. Ta je poté velice důležitá k implementaci uživatelských přístupů. O tom si ovšem více povíme v části backendu. Po úspěšné registraci je uživatel zpátky přesměrován na hlavní stránku, stále ale jako nepřihlášený uživatel. Z tohoto důvodu se musí ještě nejdříve přihlásit.



Obr. 4 Stránka s přehledem týmů uživatele (zdroj vlastní)

Po přihlášení může uživatel používat aplikaci podle možností, které jsou definovány pro hráče a trenéry. Všichni uživatelé vidí záložky „*Moje Týmy*“, „*Kalendář*“ a „*Moje Výkony*“. Nejprve si popíšeme stránku „*Moje Týmy*“, ta je zachycena na obrázku 4. Tato stránka je velice jednoduchá a obsahuje tabulku všech týmů, které uživatel buď vytvořil (označené zeleně) anebo byl do nich přidán jako člen. Na této stránce se dají vytvořit jednoduše další týmy tím, že do pole v horní části týmu napíšete název skupiny a kliknete na „*Vytvoř tým*“. Toto tlačítko zajistí přidání týmu do tabulky a jelikož jste označen jako vlastník týmu, budete ho mít označený zeleně. Družstva může vytvářet jakýkoliv uživatel, ať už hráč nebo trenér. Do detailního týmu se můžete dostat buď dvojklikem na příslušný řádek skupiny v tabulce anebo označením daného řádku a kliknutím na „*Otevřít tým*“. Tlačítka „*Upravit tým*“ a „*Smazat tým*“ se zpřístupní pouze tehdy, pokud kliknete na tým, jehož jste vlastníkem, aby bylo zabráněno úpravě cizích týmů.

TR Wolf

IBK HK - juniorky

Členové

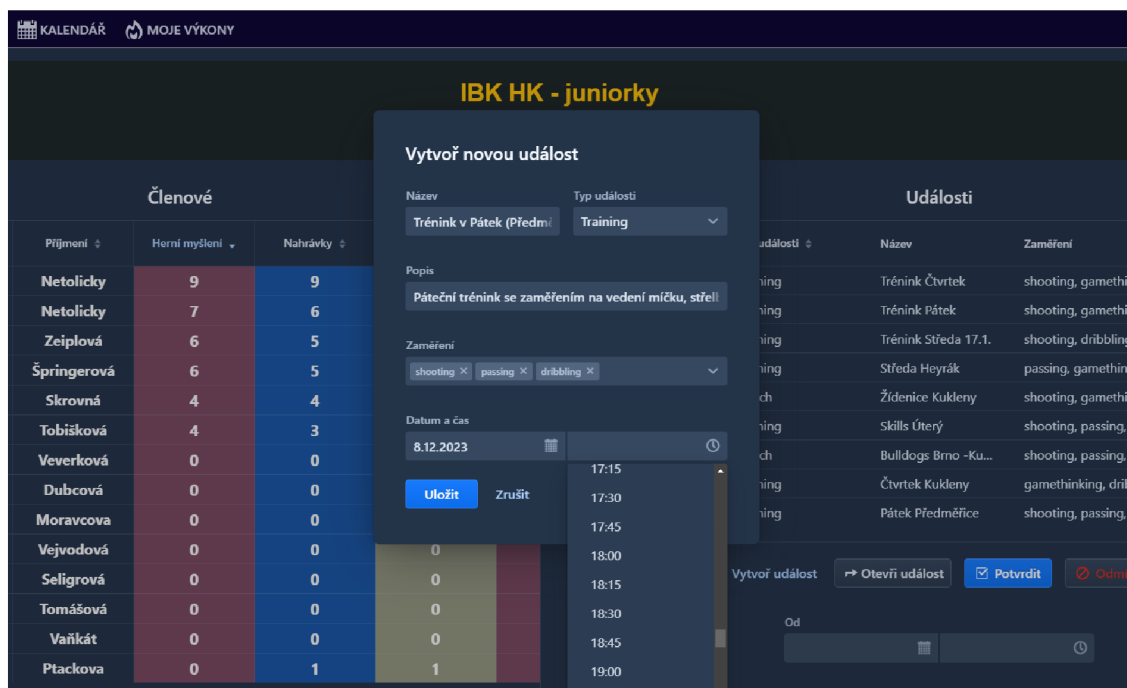
Uřazení	Jméno	Příjmení	První týden	Druhý týden	Třetí týden	Čtvrtý týden	Veškeré body	Měsíc
Trenér	Štěpán	Nerolícky	9	9	6	9	10	5
Wahllys	Števan	Nerolícky	7	6	6	7	7	2
Zip	Aneta	Zajícová	6	5	6	7	6	2
Havran	Verniška	Špringerová	6	5	6	7	6	2
adé	Adriana	Števaná	4	4	3	3	4	1
Kata.tob...	Kateřina	Tobillová	4	3	3	4	4	1
Vida	Valérie	Nováková	0	0	0	0	0	0
Duběnač	Verniška	Dubcová	0	0	0	0	0	0
VikyMa...	Viktorie	Moravcová	0	0	0	0	0	0
vikky	Viktorie	Vajřvodová	0	0	0	0	0	0
Pája	Prošna	Selgrová	0	0	0	0	0	0
Margj	Margj	Tomášová	0	0	0	0	0	0
Olga	Olga	Vančát	0	0	0	0	0	0
zpřes.kova	Zuzka	Plánková	0	1	1	1	1	0

Události

Datum a čas	Typ události	Název	Zaměřeno	Událost
18.01.2024 20:00	training	Táňa Čvrtak	shooting, gametheking, pas...	Otevřít událost
19.01.2024 18:30	training	Táňa Čvrtak	shooting, gametheking, ar...	Otevřít událost
17.01.2024 17:30	Training	Táňa Čvrtak 17.1.	shooting, dribbling	Přijmout
28.11.2023 16:30	training	Štěpa Haytík	passing, gametheking, dri...	Přijmout
02.12.2023 14:30	Match	Števan Kulkový	shooting, gametheking, ar...	Přijmout
16.01.2024 14:30	Training	Silvi Olavj	shooting, passing, gamethe...	Přijmout
05.12.2023 17:00	Match	Burkha Ben - Galány	shooting, passing, gamethe...	Přijmout
28.11.2023 20:00	training	Čárta Káťa	gametheking, dribbling, ph...	Otevřít událost
01.12.2023 18:30	training	Páňa Piončíková	shooting, passing, dribbling...	Přijmout

Obr. 5 Detailní pohled na stránku týmu (zdroj vlastní)

Nyní se podíváme na příslušné detailní zobrazení vybraného týmu, které je umožněno k prohlédnutí na obrázku č. 5. Na tomto obrázku je zobrazení družstva z pozice vlastníka. Ten v levé tabulce vidí všechny členy týmu, může je zároveň přidávat, odstraňovat či prohlížet samostatně detailní informace o jednotlivých pomoci tlačítek pod tabulkou. Tlačítko „Přidat“ otevře modální okno se všemi členy v databázi. Je implementováno vyhledávání členů podle jména, příjmení nebo uživatelského jména a správce týmu může klidně přidat několik členů do týmu najednou. Tlačítko „Zobrazit detaily“ zobrazí jiné modální okno s kontaktními informacemi a statistikami výkonnosti o vybraném členovi. V pravé části obrazovky jsou v tabulce vypisovány jednotlivé události související s aktuálním týmem. Členové družstva tyto události potvrzují či odmítají pomocí tlačítek pod tabulkou. Pokud přijmou událost, zároveň se jim přidá do kalendáře. Navíc budou po skončení události odměněni body podle toho, na co byla daná událost zaměřena. Tlačítko „Otevřít událost“ přeměruje uživatele na stránku s detailním popisem události, které můžeme vidět na obrázku 7.

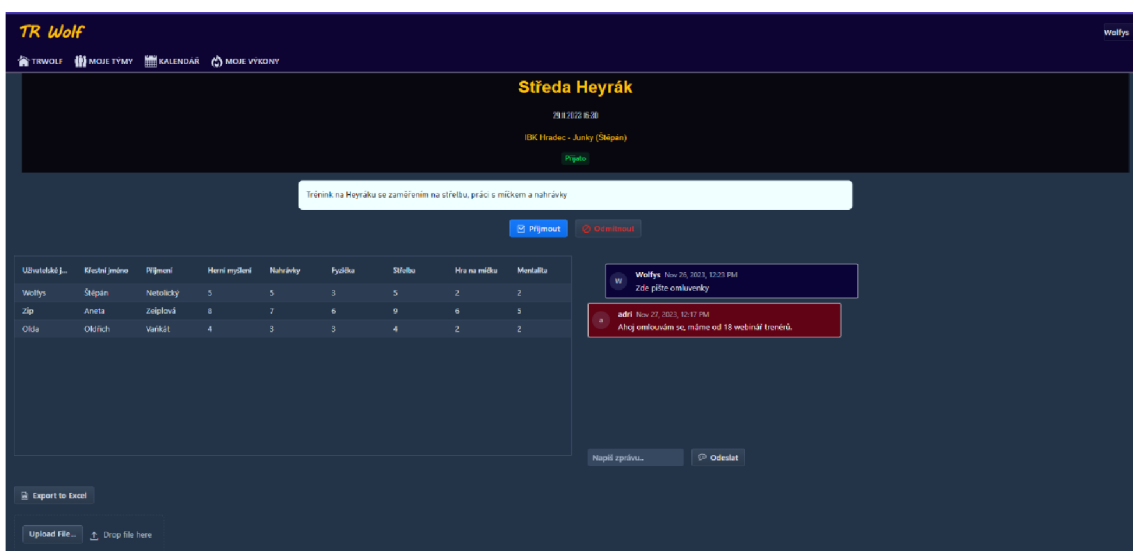


Obr. 6 Modální okno vytváření události (zdroj vlastní)

Naopak tlačítko „Vytvoř událost“ otevře modální okno, kam uživatel zadá informace ohledně vytvářené události. Detailní ukázka této funkcionality je zachycena na obrázku 6. Uživatel zadá povinný název, může přidat volitelný popis a také typ události, kdy má na výběr ze čtyř možností (Trénink, Zápas, Týmové soustředění a Jiné). Tento argument je povinný. Dále pak vybere datum a čas události a zároveň zvolí jeden až šest typů, na co je událost zaměřena. Toto zaměření je spojeno se statistikami hráčů, které se jim zvyšují na základě toho, zda událost na dané zaměření absolvovali či nikoliv. Na výběr je zaměření na střelbu, nahrávky, fyzickou, herní myšlení, ovládání míčku a mentalitu hráče. Po vyplnění těchto polí může událost uložit a ta se přidá do tabulky. Pokud nevyplnil některé z povinných polí, tak je uložení odmítnuto a frontend mu zobrazí chybovou hlášku.

Zajímavou funkcí je také exportování excelu, viditelné znovu na obrázku 5. Funguje tak, že uživatel, který má oprávnění, nejprve vybere datумы od a do kdy chce docházku v excelu vygenerovat. Následně klikne na tlačítko exportu. Toto mu připraví zpracovaný excel soubor s daty o uživateli a událostech a o jejich docházce podle toho, jestli tuto událost přijmou, či nikoliv.

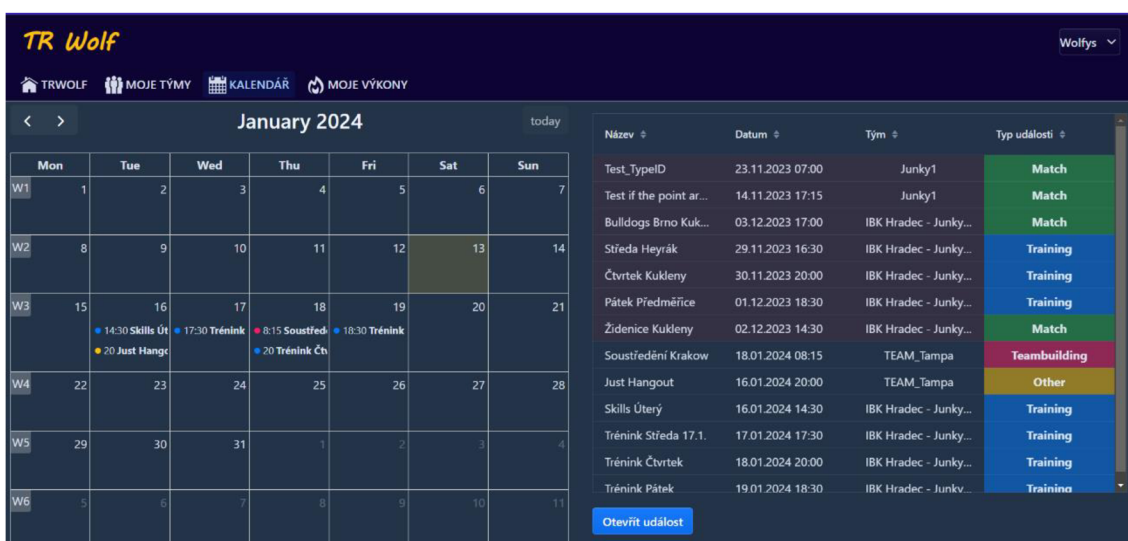
Důležité je poznamenat, že se liší zobrazení tlačítek pro běžné členy neboli hráče, správce týmu a člena týmu s příznakem trenéra. Člen týmu s příznakem trenéra má stejné oprávnění jako správce týmu, ikdyž doopravdy není vlastníkem. Na této stránce má k dispozici plnou verzi funkcí. Členové týmu s příznakem hráče, kteří nejsou jeho vlastníkem, nemají možnost exportovat excel, spravovat členy týmu a přidávat události do týmu. Více si o uživatelských přístupech povíme v kapitole Uživatelský přístup.



Obr. 7 Detailní pohled konkrétní události (zdroj vlastní)

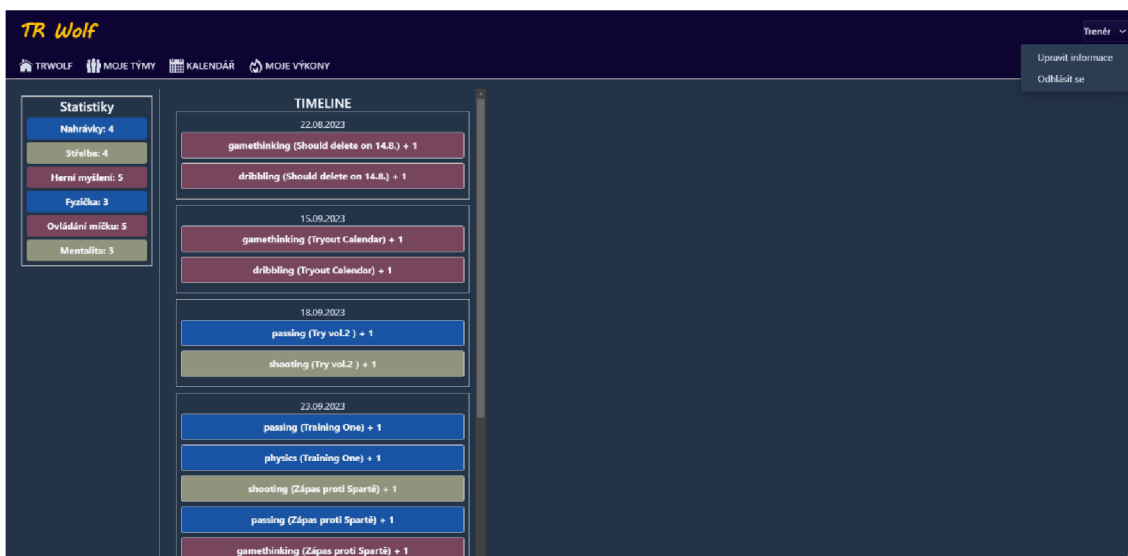
Dalším pohledem je stránka s detailním popisem události. Tuto stránku ukazuje obrázek 7. Na konkrétní událost se dá dostat dvěma způsoby. Prvním je, že pokud jsme událost už přijmuli, zobrazí se nám na stránce s kalendářem. Přes tuto stránku se dá poté dostat do konkrétní události. Stránka s kalendářem je ukázána na obrázku 8. Více to bude vysvětleno níže. Druhý způsob je přes stránku týmu a pomocí dvojkliku či tlačítka „Otevřít událost“ zachyceného na obrázku 5 s detailním pohledem. Z tohoto pohledu se můžete dostat jak do potvrzené, tak odmítnuté události. Na této stránce uvidíte tabulku se členy, kteří tuto událost už potvrdili. Lze to vidět na obrázku 7. Z tohoto pohledu můžete také danou událost potvrdit či přijmou a změny se projeví do jiných pohledů. Zároveň je implementován jednoduchý chat, pro omluvenky z tréninků. Zprávy od jiných uživatelů jsou označeny červeně, vaše zprávy modře. Zároveň je implementována další možnost

stáhnutí docházky u konkrétní události do excelu. V tomto excelu trenér uvidí seznam členů přihlášených k aktuální události. V neposlední řadě je na této stránce přidána funkcionality uploadu PDF souborů. Toto může být užitečné pro trenéry, pokud si chtějí uchovávat PDF soubory, které budou obsahovat například tréninkovou jednotku či jiné poznámky. Velikost daného souboru, ale nesmí přesáhnout 10 MB.



Obr. 8 Stránka s kalendářem (zdroj vlastní)

Jako další si popíšeme, jak funguje obrazovka kalendáře, viz. obrázek 8. Tento kalendář je jako další součást Vaadinu, který z něj vytváří samostatnou třídu v Javě. Přeprocovává však originální FullCalendar, který byl napsán za pomoci JavaScriptu. V levé části obrazovky můžeme vidět právě kalendář, kam jsou automaticky zapsány akce, které uživatel přijmul. Zároveň po kliknutí na kterýkoliv den můžete událost vytvořit i z tohoto pohledu, ale navíc je zapotřebí vybrat tým, pro který je daná akce vytvářena. Různé typy událostí jsou odlišeny barevně. V pravé části obrazovky pak uživatel může vidět tabulky s přijatými událostmi. Ty, které jsou označeny jako červené se pro uživatele s příznakem hráče mažou automaticky po skončení datumu události. Pro uživatele s příznakem trenéra se pouze označí červeně, aby se do nich trenér mohl zpětně vracet a spravovat PDF soubor. Z tohoto pohledu se zároveň můžete dostat na stránku s detailními informacemi události, jak bylo zmíněno výše.



Obr. 9 Stránka výkonů uživatele s časovou osou (zdroj vlastní)

Poslední stránkou je pohled na výkony hráče, ukázaný na obrázku číslo 9. Na této stránce se zobrazuje tabulka s celkovým přehledem statistik a bodů za jednotlivá zaměření, která uživatel do této doby získal. Vedle je pak zobrazena časová osa, která vystihuje, kdy uživatel tyto body získával a za jaké události. Toto všechno vykonává funkce scheduleru, který je v aplikaci implementován a pravidelně přidává body členům za přijaté události. V pravém horním rohu je také zobrazena funkčnost správy svého účtu. Uživatel se buď může odhlásit anebo upravit svoje informace. To mu zobrazí modální okno, kde si může změnit svůj email nebo uživatelské jméno.

Nyní přejdeme na kapitolu backendu, kde si podrobně popíšeme, jak určité funkce pracují a co zajišťují v aplikaci.

5.1.2 Backend

Backendová část aplikace nám zajišťuje propojení aplikace s databází, ukládání a sbírání dat z databáze a další funkcionality, jako je automatické přidělování bodů hráčům či ukládání a stahování PDF souborů. Data jsou uložena, jak už bylo zmíněno výše, v databázovém systému PostgreSQL. Pomocí Springu jsou vytvářeny odpovídající entity pro databázové tabulky. Za pomoci takzvaných repositories a services se může přistupovat k databázovým tabulkám skrz kód. V daném

repository se vždy spojí entita s danou tabulkou a ve třídě service můžeme vytvářet metody pro CRUD operace, které zajišťují vytváření, čtení, aktualizování a mazání daných záznamů do tabulek. Za pomoci tohoto přístupu, pak máme přímý přístup k databázi skrze programovaný kód. V aplikaci bylo třeba vyřešit automatizované přidělování bodů hráčům. K tomu byla implementována třída ScheduleService a Scheduler. Daný kód implementovaný v této třídě můžeme vidět na obrázku číslo 10. Třída ScheduleService obsahuje metodu „checkAndGiveEventBonuses()“, která řeší logiku problematiky přidělování bodů hráčům po skončení události. Nejprve se zkontrolují události, jejichž datum je před časem aktuálním, tudíž v minulosti. Pak se pro tyto události kontrolují vazby s danými hráči, kteří tuto událost potvrdili. Těmto hráčům se přičtou body do takových statistik, které se shodují se zaměřením události. A vytvoří se záznamy o tomto úkonu do odpovídající tabulky. Ve třídě Scheduler se pak tato metoda volá automaticky každých dvanáct hodin.

```
package com.example.application.components;

import com.example.application.data.service.services.ScheduleService;
import org.springframework.scheduling.annotation.EnableScheduling;
import org.springframework.scheduling.annotation.Scheduled;
import org.springframework.stereotype.Component;

@EnableScheduling
@Component
public class Scheduler {
    private final ScheduleService scheduleService;

    public Scheduler(ScheduleService scheduleService) { this.scheduleService = scheduleService; }

    @Scheduled(fixedDelay = 43200000) //checks after half a day
    public void schedule() {
        System.out.println("Přidávám bonusy a odstraňuji hráčům eventy");
        scheduleService.checkAndGiveEventBonuses();
    }
}
```

Obr. 10 Třída Scheduler (zdroj vlastní)

Druhou podstatnou úlohou, kterou zajišťuje backend je ukládání a zajišťování stahování PDF souborů k daným událostem. Vaadin poskytuje komponentu uploadu, která umožňuje otevřít souborový prohlížeč a nahrát PDF soubor. Základní

úlohou backendu je pak uložení daného souboru k události. Kód, který se stará o tuto funkčnost, je viditelný na obrázku číslo 11. Informace o daném souboru nejprve uloží do záznamu databáze. Hlavním atributem je atribut „*path*“, který ukládá cestu k souboru v aplikaci. Samotný soubor se pak nahraje pomocí *FileInputStreamu* do adresáře, který je vytvořený v samotné aplikaci. Tato funkčnost zaručí, že databáze bude ukládat informace o souboru včetně cesty k němu. Samotný soubor je pak nahrán v adresáři aplikace. Pokud chce uživatel daný soubor připojený k události stáhnout znovu, tak se nejprve najde odpovídající záznam o souboru v databázi. Poté se podle jeho informací lze dostat do správného adresáře, odkud jde soubor znovu stáhnout.

```
1  /**Stores the record of the file into the database*/
2  private void storeFileToDB(String fileName){
3      String filePath = "/files/" + fileName;
4
5      FileDB uploadedFile = new FileDB();
6      uploadedFile.setEvent(mainEvent);
7      uploadedFile.setSuffix(".pdf");
8      uploadedFile.setFileName(fileName);
9      uploadedFile.setPath(filePath);
10     fileService.save(uploadedFile);
11 }
12
13 /**Stores the file into the directory in the app*/
14 private void storeFileToDirectory(InputStream fileData, String fileName){
15     File fileDir = new File( pathname: "files");
16     try {
17         File outputFile = new File(fileDir, fileName);
18         OutputStream outputStream = new FileOutputStream(outputFile);
19         byte[] buffer = new byte[1024];
20         int bytesRead;
21         while ((bytesRead = fileData.read(buffer)) != -1) {
22             outputStream.write(buffer, off: 0, bytesRead);
23         }
24         outputStream.close();
25         fileData.close(); // Close the input stream
26     } catch (FileNotFoundException e) {
27         e.printStackTrace();
28     } catch (IOException e) {
29         e.printStackTrace();
30     }
31 }
```

Obr. 11 Funkcionalita ukládání PDF souboru (zdroj vlastní)

Další potřebnou funkcí bylo vyřešení problému generování a stahování Excel souborů. Toho bylo docíleno za pomoci knihovny Apache POI, stažené za pomoci Mavenu, a která umožňuje generování a upravování excel souborů přímo ve zdrojovém kódu. V naší aplikaci je pak toto implementováno ve třídě ExcelExporter, která je ukázána na obrázku 12. Nejprve se připraví a nadefinuje, jak bude vypadat daný excelový list. Poté se pomocí takzvaných for-cyklů určí, jak se mají daná data zapisovat do políček v listu. Nakonec se vytvořila metoda „generateExcelFile()“, jenž bere jako argumenty dané Listy s obsahem a ExportType typu Enum, který byl v aplikaci vytvořen pro odlišení generovaných Excel souborů. Vytvoří se daný excelový list a „ByteArrayOutputStream“ pro ukládání bytů. Následně se zapíše obsah listu do outputstreamu a tento stream se vrátí jako bytové pole. Na daném pohledu stránky se následně vytvoří link a soubor tak může být přes tento odkaz stažen.

```
public byte[] generateExcelFile(List<?> data, List<?> data2, List<?> data3, ExportType exportType) {
    try (Workbook workbook = new XSSFWorkbook()) {
        sheet = workbook.createSheet("Data Sheet");
        if(exportType == ExportType.ATTENDANCE){
            prepareAttendance((List<User>) data, (List<Event>) data2, (List<Actualevent>) data3);
        }
        else{
            Exception exception = new Exception();
            exception.printStackTrace();
        }

        ByteArrayOutputStream outputStream = new ByteArrayOutputStream();
        workbook.write(outputStream);
        return outputStream.toByteArray();
    } catch (Exception e) {
        e.printStackTrace();
        return null;
    }
}
```

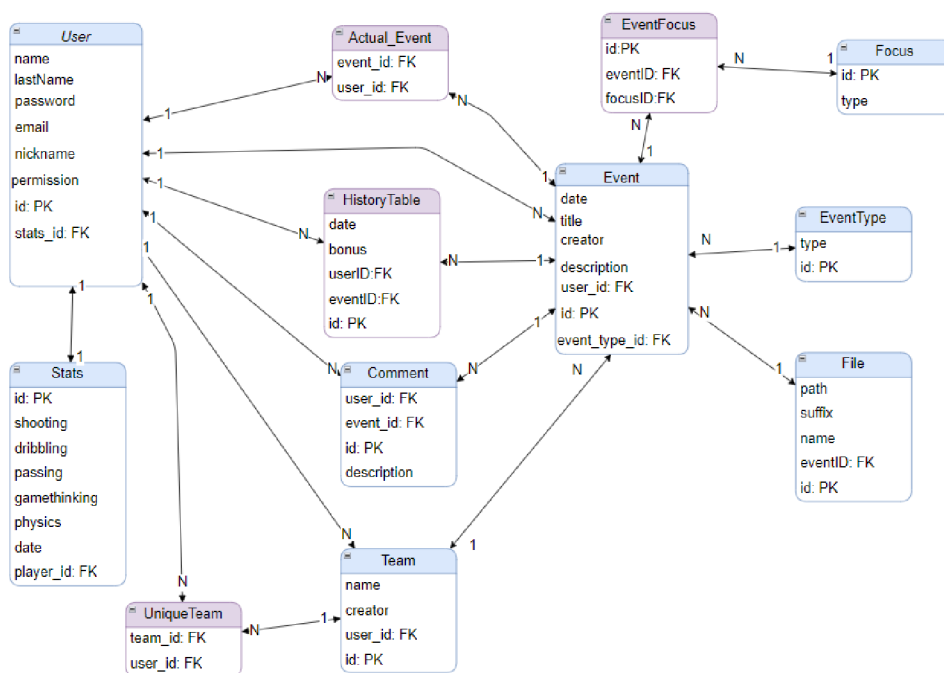
Obr. 12 Funkcionalita generování Excel souboru (zdroj vlastní)

5.1.3 Databáze aplikace

Databáze je u většiny aplikací, nejen webových, nedílnou součástí zajišťující správnou funkčnost, uchování a zabezpečení dat v aplikaci. V našem případě tomu není jinak. Naše databáze se dohromady skládá z dvanácti tabulek. Z toho osm je hlavních a čtyři slouží na oddělení informací v aplikaci a správné propojení tabulek.

Dá se proto říci, že patří mezi vedlejší tabulky. Kompletní struktura databáze je pak ukázána na obrázku číslo 13.

Základní tři tabulky, bez kterých by se tato aplikace neobešla, jsou tabulky uživatele, týmu a události. Tyto struktury tvoří základní kostru celého programu i databáze a díky nim bylo možno vytvořit základní logiku a realizovat cíl aplikace. Jelikož logika programu říká, že jeden uživatel může být jak ve více událostech, tak zároveň členem více týmů, bylo třeba zajistit propojení těchto tabulek skrze další pomocné. Těmito tabulkami jsou „*Actual_Event*“ a „*UniqueTeam*“ a díky nim bylo možné realizovat, že jeden tým či událost bude mít více členů a zároveň jeden člen může být součástí více týmů či událostí. Tyto tabulky jsou zároveň spojeny s uživatelem ještě další vazbou pro zajištění, že budeme vědět o zakladateli. Tato podobná logika platí i v případě propojení události se zaměřením.



Obr. 13 Struktura databáze (zdroj vlastní)

Uživatel má dále přístup k tabulce statistik, která odděleně uchovává seznam jednotlivých dovedností pro daného uživatele. Proto je zde vazba jedna ku jedné. Důležitou tabulkou je bezpochyby „*HistoryTable*“. Ta má za úkol nést informace

o změnách ve statistikách určitého uživatele. V aplikaci je pak její hlavní využití v pohledu výkonů hráčů. Ti tak mají k dispozici časovou osu s přehledem získaných bodů z daných událostí. A v neposlední řadě zde máme tabulku, která uchovává informace o nahrávaných souborech do aplikace, jak už bylo zmíněno výše. Důležité je ale zmínit, že ona samotná vůbec nedrží obsah daného souboru. Pouze uchovává jméno a příponu a dále pak proměnou „*path*“, která vždy obsahuje pevnou adresářovou cestu k uloženému souboru v aplikaci.

5.1.4 Uživatelský přístup

Už od začátku bylo jasné, že budeme potřebovat oddělit informace, které uvidí samotný hráč a trenér. K tomu, abychom těchto přístupů mohli využívat, slouží uživatelský přístup. Díky němu můžeme v aplikacích využívat následujících logik. Skrývat citlivé informace před určitými uživateli, omezovat jejich pohyb po stránce, či znemožňovat jim používat určité funkce. Každý uživatel většinou při registraci dostane daný stupeň oprávnění, který určuje to, co může v aplikaci vykonávat.

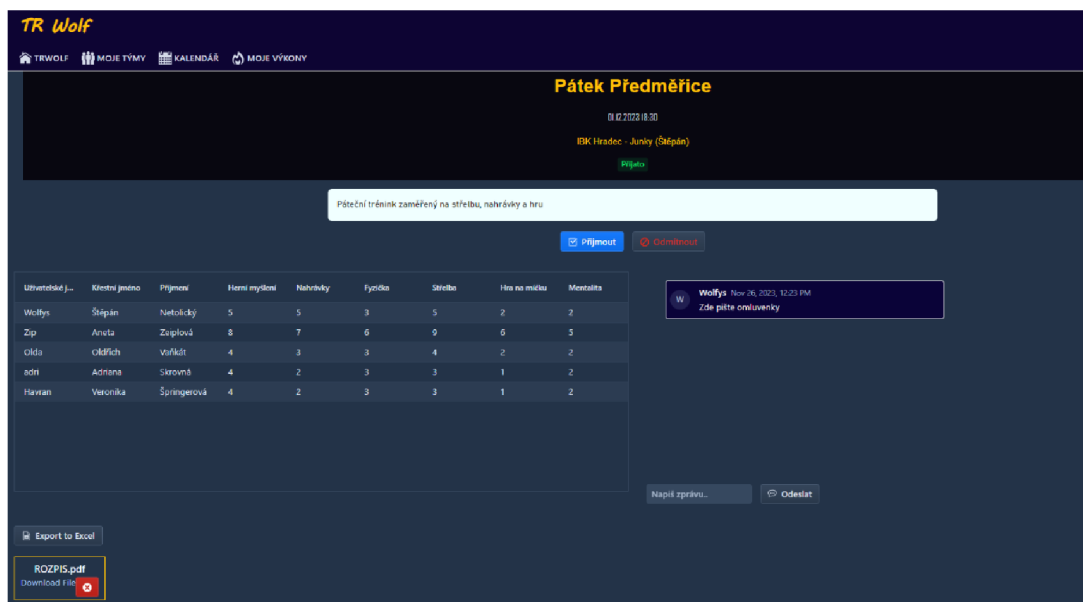
Jinak tomu není ani v naší aplikaci. Původně bylo zamýšleno, že k oprávněním hráče a trenéra přibude také administrátor. To se však po vyhodnocení cílů aplikace ukázalo jako nadbytečné. V momentální fázi aplikace, podle funkcí, pro které je naprogramována, oprávnění administrátora ztrácí význam. V aplikaci jsou tak implementovány pouze dvě hlavní úrovně uživatelských oprávnění a těmi jsou hráč a trenér. Budeme-li brát v potaz i nově přichozícího uživatele, který se zatím neregistroval, můžeme ho označit za další stupeň oprávnění. Tito uživatelé se bez registrace dostanou pouze na hlavní stránku a stránku s registračním formulářem. Popřípadě na stránku s přihlašovacím formulářem, ale do toho zatím nemají potřebná data. Hlavní oprávnění je určováno podle toho, co uživatel zvolí při registraci do aplikace a tato práva mu zůstávají už napořád, nemůžou se tedy už měnit později. Počítáme s tím, že uživatelé budou čestní a zvolí správné oprávnění podle toho, jak chtějí aplikaci používat. Každý člen může mít pouze jedno oprávnění. Pokud tedy najdeme takového hráče, který je zároveň i trenér, bude si buď muset vytvořit dva účty anebo zvolit vyšší oprávnění, což je v našem případě oprávnění trenéra a aplikaci používat skrz tento přístup i jako hráč. O své body do statistik

nepřijde, protože právě kvůli tomuto důvodu v aplikaci nebyla implementována logika přidělování bodů pouze hráčům. To ve výsledku znamená, že nezáleží, jaké oprávnění uživatel má, ale záleží na tom, zda danou událost přijme či nikoliv.

Jak už bylo řečeno, v této aplikaci jsou implementovány dva hlavní přístupy. Přístup pro hráče a druhý pro přístup trenéra. První z jmenovaných má menší oprávnění ovšem dělí se dále podle toho, zda je daný hráč vlastníkem týmu či nikoliv. Hlavní rozdíl v oprávněních najdeme především ve funkcích jako je nahrávání PDF souboru či exportování excelu. Tyto funkcionality najdeme na pohledech detailů týmu a událostí. Na detailním pohledu týmu má hráč, který není vlastníkem velmi omezené možnosti. Za prvé nevidí statistiky ostatních hráčů, vidí pouze jejich kontaktní informace. Dále samozřejmě nemůže přidávat či odebírat hráče z daného týmu a zároveň vytvářet nové události do týmu. Nemá také možnost exportovat docházku týmu do excelu. Tyto věci se upravují, pokud je uživatel s příznakem hráče vlastníkem týmu. Může exportovat docházku, vytvářet nové události či spravovat obsazení týmu. Jediná věc, která zůstává stejná je, že stejně jako ostatní uživatelé s úrovní hráč nevidí statistiky ostatních uživatelů. Toto je implementováno kvůli tomu, abychom zajistili diskrétnost hráčů a v týmech nevznikaly zbytečné náznaky šikany či posměchu. V detailním pohledu události jsou práva oddělena podobně. Stejně jako v předchozím případě platí, že uživatelé, kteří jsou registrováni jako hráč, nevidí statistiky ostatních, ani když jsou tvůrcem události. Zároveň ale takoví uživatelé mají k dispozici nahrávání PDF souboru k události, stejně jako uživatelé s příznakem trenér. Tito uživatelé mají oprávnění nahlížet a vytvářet věci i když nejsou zakladatelem daného týmu či události. Jedinou věc, kterou mají zakázanou, je mazání týmů, které nejsou jejich. Pohled události s nahraným a připojeným PDF souborem je poté zobrazen na obrázku číslo 14.

Tyto funkce jsou v kódu řešeny pomocí různých podmínek, které se ptají na atribut „*permission*“ aktuálně přihlášeného uživatele. Tento atribut uchovává jeho tabulka a podle toho se různě naplní datové struktury či zobrazí různé komponenty na stránce. Tento přístup zajišťuje jednoduché rozdělení možností pro přihlášeného

uživatelé a napomáhá nám zajišťovat diskrétnost informací, které nechceme zobrazovat pro daný typ uživatelů.



Obr. 14 Detailní pohled události s nahraným PDF souborem (zdroj vlastní)

5.2 Testování aplikace

Po dokončení projektu a jeho nasazení na server Heroku se mohlo přejít k testování v reálném prostředí. Jelikož bylo za potřebí zapojit několik uživatelů, aby výsledky aplikace odpovídaly realitě, byly osloveny juniorky z týmu IBK Hradec Králové, které autor tohoto projektu trénoval. Byl vytvořen tým, který spravoval on osobně ve spolupráci s dalším trenérem. Všichni uživatelé byli nejdříve seznámeni se základním fungováním aplikace a co bude jejich úkolem. Hráčkám byly pravidelně po týdnu vypisovány události do společného týmu podle reálného rozpisu tréninků a zápasů. Ony měly za úkol na tyto události zareagovat přijmutím nebo odmítnutím události a v případě nepřítomnosti se omluvit do chatu u dané události. Testovalo se po dobu tří měsíců od poloviny října až do konce ledna s tím, že byla přestávka o Vánocích. Bylo cíleno na to, aby události co nejlépe zachycovaly měsíční tréninkový cyklus ve florbale. To znamená, že tréninky byly vždy cíleny na specifickou dovednost, kterou bylo potřeba v tomto daném měsíci trénovat. Co se

týče jiných událostí, byly vypisovány i zápasy. Daný trenér udělal a poskytl nominaci. Podle toho se měly dané hráčky přihlásit, aby o nich byl přehled i v systému. To, jak se tyto plány dařilo plnit, si popíšeme v další kapitole.

6 Shrnutí výsledků

Při samotném vytváření aplikace se narazilo na problém s kalendářem. Během zvažování různých možností, jakým způsobem implementovat kalendář do aplikace se narazilo na zásadní problém. Většina kalendářů jako je Outlook nebo Google jsou placené, a to v naší situaci nebylo vhodné řešení. Po větším vyhledávání se naskytl řešení s Vaadin kalendářem, který se nakonec dobře implementoval a je využit v naší aplikaci. Poskytuje jednoduché přidávání událostí a jeho přehlednost a funkčnost je srovnatelná s jinými řešeními. Nespornou výhodou bylo, že v aplikaci byl Vaadin už využíván, tudíž nebylo zapotřebí instalovat do projektu další externí knihovny.

Při testování byly největší problémy především v komunikaci s hráčkami a trenérem. Od začátku testování byla snaha zapojit více hráček do reakcí na události, ale bohužel si těchto výzev všimla jen menšina a tím pádem nereagovaly všechny. Kvůli tomu byly výsledky z aplikace zkreslené malým počtem dat od jednotlivců.

Během celého procesu testování aplikace online se nenarazilo na problém s funkčností jakýchkoliv věcí. Prvotně bylo zapotřebí naplnit databázi, která byla taktéž nahrána na heroku serveru. Jakmile aplikace měla k dispozici veškerá základní data do číselníků, mohli se uživatelé registrovat a zkusit aplikaci. Slovní zpětná vazba na tento program byla v průměru dobrá. Byla oceněna její jednoduchost a přehlednost. Jak už bylo řečeno, na žádný funkční problém se nenarazilo, a tak testování probíhalo ve výsledku v klidu. Postupně se mohly přidávat nové události a hráčky na ně mohly reagovat. Po ukončení testovacího procesu v lednu bylo umožněno vygenerování celkového přehledu docházky na události podle hráček a také bylo vidět, kdo by měl vynikat v jakých dovednostech. Tyto přehledy jsou zobrazeny na obrázcích číslo 15 a 16.

Členové								
Uživatel	Jméno	Příjmení	Herní myšlenka	Nahrávky	Fyzická	Síťovka	Vedení míčku	Mentalita
Trenér	Štěpán	Netolický	9	9	6	9	10	5
Wolfys	Števen	Netolický	7	6	6	7	7	2
Zip	Aneta	Zeiplová	6	5	6	7	6	2
Havran	Veronika	Špringerová	6	5	6	7	6	2
adri	Adriana	Skrovná	4	4	3	3	4	1
kata.tobi...	Kateřina	Tobišková	4	3	3	4	4	1
Váňa	Valérie	Veverková	0	0	0	0	0	0
Dubínek	Veronika	Dubcová	0	0	0	0	0	0
VikyMor...	Viktorie	Moravcova	0	0	0	0	0	0
viking	Viktorie	Vejvodová	0	0	0	0	0	0
Pája	Pavlina	Seligrová	0	0	0	0	0	0
Margi	Margi	Tomášová	0	0	0	0	0	0
Olda	Oldřich	Vaňkát	0	0	0	0	0	0
zptackova	Zuzka	Ptackova	0	1	1	1	1	0

Obr. 15 Detailní pohled na statistiky hráčů (zdroj vlastní)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1	Event/User	Wofys	Victorie Moravcová	Váňa	Hanyš69	Pavlna Seligrová	Havran	Zip	adri	Pája	Olda	Margi	viking	verca.dubcova	kata.tobiskova	Kačka	Viki Moravcová	zptackova	
2	(1.Test) Trenink Pátek	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	Sředa Heyrák	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
4	Čtvrtek Kuklenny	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0
5	Pátek Předměřice	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0
6	Ždemece Kuklenny	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0
7	Bulldogs Brno Kuklenny	1	0	0	0	0	0	1	1	1	0	1	0	0	0	1	0	0	0
8	Trenink Sředa	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
9	Trenink Čtvrtek	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
10	Trenink Pátek	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
11	Trenink Heyrovského	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	Trenink Čtvrtek	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	Trenink Pátek - Předměřice	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
14	Zápas proti Sportě	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
15	Zápas Blatci	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
16	Skibi Úterý	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	Trenink Sředa 17.1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	Trenink Čtvrtek	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
19	Trenink Pátek	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
20	Skibi trenink	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
21	Heyrák trenink	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
22	Kuklenny Arena trenink	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
23	Předměřice Trenink	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
24	Láberec Match	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

Obr. 16 Vygenerovaný .xlsx souboru s docházkou (zdroj vlastní)

7 Závěry a doporučení

Aplikace celkově splňuje veškerá očekávání, která byla stanovena. V aplikaci je možné vytvářet a rozdělovat uživatele do různých týmů. Zároveň je možné vytvářet události, což je vlastně její základní prvek a cíl. Do aplikace nakonec byl místo Outlook kalendáře implementován Vaadin kalendář, který je součástí nástrojů, kterých bylo využito pro tvorbu frontendu. Bylo tak učiněno z důvodu dostupnosti jednotlivých řešení a z hlediska náročnosti na implementaci. Jak už bylo specifikováno v základních požadavcích, aplikace má umožňovat následující. Uživatelům zobrazovat různé dashboardy s přehledy o hráčích a jejich trénovanosti, podle toho, jak se jednotliví hráči přihlašují na události. Tyto přehledy si trenéři mají možnost vyexportovat do excelu a z obdržných dat dělat grafy či různé jiné přehledy, které uznají za vhodné.

Co se týče vzhledu aplikace, nabízí základní, ale efektivní vzhled pro přehlednost v aplikaci, který je responzivní pro počítače a tablety. Frontend aplikace je věc, která by se dala posunout na další úroveň. Aplikaci chybí mobilní verze, která by určitě byla vhodná vytvořit v dalších fázích a umožnit tak uživatelům používat ji z pohodlí svého telefonu. Při výzkumu ostatních aplikací se zjistilo, že umožňují kompletnější správu pro celý klub a jsou i více zaměřeny na jeho organizaci. Této věci by se dalo v budoucnu dále věnovat a implementovat do aplikace takové funkce, které by umožňovaly spravovat nejen sportovní, ale i organizační stránku fungování klubu. Mezi tyto funkce by bylo určitě dobré zařadit přehled členských poplatků od hráčů, jelikož správa financí bývá v menších klubech často chaotická. Taktéž by byla možnost vytvořit z aplikace větší komunikační kanál, který by sloužil pro komunikaci napříč klubem mezi členy, tak i mezi rodiči. Toto by obnášelo propojení s emailovým systémem pro lepší přehlednost a účinnost. V poslední řadě by se dala aplikace dále rozvézt do formy pro mobilní telefony, ať už přístupem výše zmíněných hybridních či nativních aplikací.

Celkově, ale aplikace splňuje daná očekávání a je schopná být využívána v reálném prostředí sportovního klubu.

8 Seznam použité literatury

[1] SIDELINESPORTS. *XPS Network* [online]. 2023 [cit. 2023-10-03]. Dostupné z: <https://www.sidelinesports.com/cs/features/>

[2] *XPS Network: Komplexní softwarové řešení pro kluby, trenéry a sportovce* [online]. In: . s. 19 [cit. 2023-10-03]. Dostupné z: <https://www.sidelinesports.com/overview/cs/general.pdf>

[3] *Book of Vaadin*. 4.vydání. Finsko: Marko Grönroos, 2012. ISBN 978-952-92-6753-8. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=piFYAwAAQBAJ&oi=fnd&pg=PP2&dq=vaadin+framework&ots=v8sUlwdGLO&sig=cksl7AbBkt2bZ4gKQE YSvEhVr8&redir_esc=y#v=onepage&q&f=false

[4] VAADIN LTD. *Build Java web applications faster with Vaadin* [online]. 2023 [cit. 2023-11-11]. Dostupné z: <https://vaadin.com/>

[5] ARNOLD, Ken, James GOSLING a David HOLMES. *THE Java™ Programming Language*. 4.vydání. Addison Wesley Professional, 2005. ISBN 0-321-34980-6.

[6] ITRENEO. *ITreneo* [online]. 2023 [cit. 2023-10-03]. Dostupné z: <https://www.itreneo.cz/>

[7] EOS. *EOS club zone* [online]. 2023 [cit. 2023-10-03]. Dostupné z: <https://www.eos.cz/klubova-platforma>

[8] JUBA, Salahaldin, Achim VANNAHME a Andrey VOLKOV. *Learning PostgreSQL* [online]. 1. Velká Británie: Packt Publishing, 2015 [cit. 2023-10-10]. ISBN 978-1-78398-918-8. Dostupné z: <https://books.google.cz/books?id=jfKoCwAAQBAJ&printsec=frontcover&hl=cs#v=onepage&q&f=false>

[9] ADSERVIO. What is pgAdmin? *Adservio.fr* [online]. 2022, 13.5.2022 [cit. 2023-11-18]. Dostupné z: <https://www.adservio.fr/post/what-is-pgadmin>

[10] PAVLÍČEK, Jaroslav. Nativní vs. hybridní aplikace. Která se vám víc vyplatí? *Inited solutions* [online]. 2 [cit. 2023-11-04]. Dostupné z: <https://inited.cz/blog/nativni-vs-hybridni-aplikace-ktera-se-vam-vic-vyplati/>

[11] IQUEST S.R.O. *Hybridní, nativní nebo webové aplikace?* [online]. [cit. 2023-11-04]. Dostupné z: <https://www.hybridniaplikace.cz/srovnani.html>

[12] ADOBE. *Informace o webových aplikacích* [online]. ADOBE. [cit. 2023-11-04]. Dostupné z: <https://helpx.adobe.com/cz/dreamweaver/using/web-applications.html>

[13] WALLS, Craig. *Spring in action* [online]. 6. Shelter Island, New York: Simon and Schuster, 2022 [cit. 2023-11-18]. ISBN 9781617297571. Dostupné z: https://books.google.cz/books?hl=cs&lr=&id=2zVbEAAAQBAJ&oi=fnd&pg=PA1&q=spring+&ots=PhPAwA1uVj&sig=-w9v8esluDdNQd4cXKfzYqeKUn0&redir_esc=y#v=onepage&q=spring&f=false

[14] BAELDUNG. A Comparison Between Spring and Spring Boot. *Baeldung.com* [online]. 23.11.2022 [cit. 2023-11-18]. Dostupné z: <https://www.baeldung.com/spring-vs-spring-boot>

[15] SPORTLYZER LLC. *Všestranné řešení pro správu sportovního klubu* [online]. 2018, 2023 [cit. 2023-11-11]. Dostupné z: <https://www.sportlyzer.com/cs/>

[16] JETBRAINS. Features overview. JETBRAINS. *Jetbrains.com* [online]. c2000-2023 [cit. 2023-11-18]. Dostupné z: <https://www.jetbrains.com/idea/features/>

[17] ACTIVE24. *Servery.cz*. Online. C2024, 2024. Dostupné z: <https://servery.cz/>. [cit. 2024-03-26].

[18] HEROKU. *Heroku Dev Center*. Online. 2023. Dostupné z: <https://devcenter.heroku.com/>. [cit. 2024-04-01].

9 Přílohy

- 1) Zdrojový kód aplikace připojený na portálu eVŠKP a v repozitáři – https://gitlab.com/StepanNetol/trwolf-vskp-prace/-/tree/master?ref_type=heads



Zadání bakalářské práce

Autor:	Štěpán Netolický
Studium:	I2100253
Studijní program:	B1802 Aplikovaná informatika
Studijní obor:	Aplikovaná informatika
Název bakalářské práce:	Aplikace pro trenéry
Název bakalářské práce AJ:	Application for sport coaches

Cíl, metody, literatura, předpoklady:

Jedná se o aplikaci pro trenéry/hráče sportovních klubů. Tato aplikace by měla být aplikovatelná do reálného provozu a využití. V aplikaci se budou evidovat tréninky, které budou specifikovány, na co jsou zaměřeny a podle toho se pak hráčům přidávají určité "zkušenosti trénovanosti". Možno také přidávat zápasy. Oboje se bude propisovat do kalendáře pro přehled. Jednalo by se, pravděpodobně, o outlook kalendář, pro větší použitelnost a přístupnost. Trenér bude mít také seznam hráčů ve svém týmu a podle kritérií může sledovat jejich trénovanost. Hráč bude za trénink, na kterém bude přítomen, dostávat body do svých atributů a trenér pak bude mít náhled, jak je hráč připraven. Finálním výstupem bude pak dashboard s jasným přehledem statistik o jednotlivých trénincích a účastnících s možností exportu do excelu pro další použití.

Zadávací pracoviště:	Katedra informatiky a kvantitativních metod, Fakulta informatiky a managementu
Vedoucí práce:	Ing. Daniel Vondra
Datum zadání závěrečné práce:	26.1.2021