

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Webová aplikace umožňující širokou škálu možností z hlediska posilování a zdraví

Bakalářská práce

Vedoucí práce:
Ing. Pavel Turčínek, Ph.D.

Jakub Kubišta

Brno 2016

Děkuji svému vedoucímu práce Ing. Pavlu Turčínkovi, Ph.D. a pracovníkům Ústavu informatiky za věnovaný čas, trpělivost, vstřícnost a důležité rady, které mi byly užitečné při zpracování této bakalářské práce.

Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Webová aplikace umožňující širokou škálu možností z hlediska posilování a zdraví**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 15. května 2016

.....

Abstract

Kubišta, J. Web application that provides a wide range of options about fitness and health. Bachelor thesis. Brno: Mendel University, 2016.

The bachelor thesis deals with the design and implementation of web application which aims to reduce the lack of public information about healthy lifestyle and bodybuilding. UML diagrams were used for the application design. The purpose of the web application is to prepare environment for free distribution of materials within the community of people interested in weightlifting and healthy lifestyle. Authenticated users can upload documents in correct format into the application and these documents will be available to others. The application is primarily implemented in a way that is user-friendly and easy to control.

Key words

Web application, Nette, AJAX, bodybuilding and healthy eating.

Abstrakt

Kubišta, J. Webová aplikace umožňující širokou škálu možností z hlediska posilování a zdraví. Bakalářská práce. Brno: Mendelova univerzita v Brně, 2016.

Bakalářská práce se zabývá návrhem a implementací webové aplikace, která má za cíl snížit nedostatek veřejných informací ke zdravému způsobu života a posilování. Pro návrh aplikace byly použity diagramy jazyka UML. Účelem vytvářené webové aplikace je připravit prostředí pro volné šíření materiálů, a to v rámci komunity lidí, kteří se zabývají posilováním a zdravým životním stylem. Ověření uživatelé mohou do aplikace v požadovaném formátu sami nahrávat dokumenty, které budou k dispozici ostatním. Aplikace je vytvořena a zpracována tak, aby byla především uživatelsky příjemná a její ovladatelnost byla jednoduchá.

Klíčová slova

Webová aplikace, Nette, AJAX, posilování a zdravé stravování.

Obsah

1	Úvod	6
2	Cíl práce	7
3	Metodika	8
3.1	Webová aplikace	8
3.1.1	Účel webových aplikací	9
3.1.2	Výhody a nevýhody webových aplikací	9
3.2	Nástroje pro tvorbu webových aplikací	9
3.2.1	HTML5 a CSS3	10
3.2.2	PHP 7	10
3.2.3	Architektura MVC	10
3.2.4	Nette	11
3.2.5	MySQL	11
3.2.6	XAMPP	11
3.2.7	AJAX	12
3.2.8	Use Case	12
4	Návrh řešení	13
4.1	Use Case Model	14
4.1.1	Aktoři	14
4.1.2	Případy užití	15
4.1.3	Scénář pro hodnocení jídelníčků	17
4.1.4	Sekvenční diagram	18
4.1.5	Komunikační diagram	19
4.1.6	Stavový diagram	20
4.1.7	Diagram aktivit	21
4.2	ERD diagram	22
5	Implementace aplikace	25
5.1	Grafický návrh	25
5.1.1	Domovská stránka	26
5.1.2	Podstránka se seznamem položek	28
5.1.3	Podstránka konkrétní položky	29
5.1.4	Ostatní podstránky	29
5.1.5	Test navigace	30
5.1.6	Logo	30
5.1.7	Menu	31
5.1.8	Slider	32
5.1.9	Další grafické prvky	32
5.1.10	Responzivní a multiplatformní web	33
5.2	Aplikační logika	35

5.2.1	Přihlašování a registrace	35
5.2.2	Hodnocení	36
5.2.3	Komentování	37
5.2.4	Vyhledávání a seřazení	38
5.2.5	Stránkování	38
5.2.6	Zobrazení obsahu konkrétní položky	39
5.2.7	Nahrávání a mazání materiálu	39
6	Marketing aplikace	41
6.1	Sociální sítě	41
6.2	Adwords a Sklik	41
7	Diskuze	42
7.1	Podstata a způsob navrhovaného řešení	42
7.2	Porovnání výsledků s podobnými aplikacemi	42
7.3	Další možné rozšíření aplikace	43
8	Závěr	44
9	Seznam zdrojů	45
	Přílohy	47
A	Hodnocení	48
B	Komentování	50

1 Úvod

V dnešní době je řada lidí zdravím a cvičením doslova posedlá. Důvodem je snaha o kompenzaci nedostatku pohybu, například při sedavém zaměstnání či studiu. Lidé si dnes sestavují jídelníčky a zajímají se víc do hloubky o stravu oprávněně, protože díky znečištění a horší úrodnosti půdy dnes nemůžeme vypěstovat potraviny s takovými nutričními hodnotami, jako například před půl stoletím (Scheer, 2011).

Primárně se tato bakalářská práce zaměřuje především na nabídku produktů ve formě jídelníčků a tréninkových plánů pro sportovce. Podobných webových aplikací je velké množství. Většina z nich se zabývá vysvětlováním cvičení jako takového, a když už se zabývají jídelníčky, tak nabízejí velice úzkou škálu produktů. Mezi konkurenčními stránkami se často vyskytuje taková forma, kde uživatel zadá své parametry, následně zaplatí a čeká na vyjádření správců. Po určité době obdrží jídelníček, ale pokud uživatel nemá dostatečné vědomosti, tak nedokáže vyvodit objektivní srovnání. Celý tento proces by tato aplikace ulehčila. Velice populární jsou dnes především na toto téma kanály na YouTube, kde se sportovci reprezentují v určitém vlogu a pojednávají o posilování v širokém pohledu za účelem sledovanosti. Dále jsou, zejména v mladších generacích, často k vidění stránky na sociálních sítích s enormním počtem uživatelů, kde správci radí začátečníkům ve cvičení.

Na základě přístupu, který zájemci o tyto sportovní aktivity jeví, je jasné, že se nejedná pouze o správné cvičení a zdravé stravování. Nemá smysl, aby k sobě byl člověk přísný v těchto ohledech, a přitom nedbal na kvalitní spánek a například pravidelně konzumoval alkohol. Jedná se tedy o určitý životní styl podporující životasprávu. K tomuto tématu je ještě důležité podotknout, že svaly lze posilovat podobně jako mozek. V obou případech je třeba k rozvinutí určitá námaha. U mozku je v určitých případech při tomto stavu možné, aby vytvářel zcela nové neurony. Podobně je to u svalů, ale svaly se odlišují tím, že se stará vlákna nejprve musí přetrhat, aby narostla nová a silnější. Tělu je tedy třeba dávat k růstu neustálé šoky, jelikož si lidské tělo lehce zvykne na většinu činností. Naprosto špatným přístupem by tedy byla kupříkladu stejná každodenní snídaně nebo neustále opakovaný trénink. Což vede znovu k tomu, že čím víc bude dostupných materiálů pro uživatele, tím lépe (Durkáčová, 2015).

Sestavení kvalitního jídelníčku nebo tréninkového plánu je dnes nákladné, jak časově, tak finančně. Vytvoření jídelníčku na míru trvá v průměru, za předpokladu dobrých znalostí, přibližně hodinu čistého času. U člověka s menšími znalostmi o striktním stravování to může trvat až 3 hodiny. Ve chvíli, kdy by byl nashromážděn dostatek dokumentů o stravování, by se těmto časovým ztrátám drtivá většina lidí vyhnula. Podobně je tomu u tréninkových plánů, ale tato sekce je určena spíše pro pokročilé sportovce, než pro naprosté začátečníky. Důvodem je, že naprostému začátečníkovi nelze poskytnout pouze informace a spoléhat na to, že vždy provede správnou techniku. V praxi je třeba cviky vizuálně ukázat a říct k nim na jaké chyby je třeba se zaměřit. Materiálů ohledně jídelníčků a tréninkových plánů je k nalezení nespočetně mnoho, ale v malých počtech a na různých stránkách.

2 Cíl práce

Vytvoření webové aplikace, která se bude zaměřovat především na určitou skupinu trenérů. Tato cílová skupina si sdílí nejen tréninkové plány a jídelníčky, ale také nabízí své služby veřejným návštěvníkům. Tím lektoři zvyšují svoji klientelu. Je nutné, aby se trenéři prokázali svojí licenci, za účelem vkládání souborů na webové stránce. Přihlášení je umožněno i ostatním uživatelům. Aby aplikace byla využívána pokud možno co nejvíce uživateli, je třeba ji ze začátku více zpřístupnit veřejnosti. Tím je myšleno, že dokud aplikace nebude všeobecně známá, tak bude moci každý přihlášený uživatel nahrávat soubory, které se týkají jídelníčků a tréninků. Cílem je také zvýšení klientely posiloven. Jedná se pouze o posilovny, jejichž vedení souhlasilo se zapojením do tohoto projektu. Prostřednictvím aplikace je usilováno o vznik takového sběru dat, který by umožňoval shromažďování uživatelských aktivit.

Východiska zkoumaných problémů, které byly uvedeny v předchozí kapitole, představuje tato aplikace. Řešením je seskupení velkého množství materiálů ohledně posilování a zdraví na jednom místě, možné porovnání a vysvětlení těchto dokumentů. V budoucnu se očekává pro aplikaci taková databáze, která bude disponovat s jídelníčkem či tréninkovým plánem pro každého. Aby bylo zobrazení nabídky aplikace opravdu specifikováno pro určitého člověka, bude u jídelníčků možno zadat do vyhledávacího filtru váhu, věk a účel jídelníčku. Pro vyhledání specifikovaného tréninkového plánu bude rozdíl v tom, že uživatel zadá typ tréninku, počet tréninkových dní v týdnu a délku tréninkového plánu. Díky těmto faktorům budou pro uživatele náklady na sestavení jídelníčku nebo tréninkového plánu minimalizovány. Trenéry lze vyhledávat zadáním parametrů, jako jsou vzdělání, poskytované služby a ve které posilovně pracují. Při hledání konkrétní posilovny uživatel může zadat údaje, jako je adresa a rozmezí cen vstupného. Za pomoci těchto specifikací uživatel snadno vyhledá zkušeného trenéra a cenově dostupnou posilovnu v blízké lokalitě. Celý web bude založen na hodnocení, komentování a počtu zobrazení hlavních objektů v databázi. Jedná se o trenéry, posilovny, tréninkové plány a jídelníčky. Díky těmto informacím bude pro zákazníka klíčové například to, který jídelníček nejlépe odpovídá specifikované kategorii pro určitého zákazníka.

3 Metodika

Tvorba aplikace bude probíhat následnými kroky, které vyplývají ze zadání práce:

- Pro nastudování problematiky ohledně vývoje webových aplikací, které se zaměřují na architekturu MVC, budou informace čerpány z odborné literatury a internetových zdrojů.
- Průzkum aktuálního stavu webových aplikací, které se týkají zdravé výživy a posilování, bude proveden na moderních aplikacích od českých i zahraničních autorů. Z tohoto průzkumu budou vyvozeny nedostatky zkoumaných aplikací a možné inovace.
- Sestavení požadavků na návrh a následné tvorby aplikace bude provedeno na základě určení cílových skupin a jejich potřeb. Následně bude vymezeno, jakým způsobem budou tyto potřeby uspokojeny. Pomocí použití UML diagramů pro vytvoření návrhu budou definovány možnosti aplikace a rozdělení uživatelů do kategorií dle oprávnění.
- Reprezentace pomocí ERD pro návrh datového schématu bude obsahovat veškeré potřebné tabulky a prvky pro chod aplikace. Zároveň se do databáze vloží data pro vývoj a testování.
- Implementace webové aplikace bude vyplývat z návrhu práce. Bude provedena tak, že se nejprve vytvoří grafická šablona. Následně budou naprogramovány jednotlivé třídy a jejich funkce.
- Za účelem propagace bude zakoupena doména a webhosting. Následně bude proveden průzkum ohledně marketingových kampaní.
- Výsledky budou otestovány na náhodných sportovcích se zájmem o posilování a zdraví. Poté bude nabídnuta spolupráce posilovněm a trenérům v Brně. Následně bude vyhodnocena úspěšnost celé práce.

3.1 Webová aplikace

Jedná se o program, který probíhá na straně webového serveru a obvykle je spojen s databází. Konečná podoba webové aplikace je obsažena na webové stránce, která je poskytována za pomoci internetu nebo intranetu a zobrazí se díky zvolenému prohlížeči. Toto je zásadní rozdíl od aplikací typu klient/server, které běží na serverech ve vnitřní síti. Webová stránka nemusí vždy obsahovat webovou aplikaci, i když se to tak převážně jeví. Webové služby jsou definovány rovněž jako webové aplikace (Rouse, 2011).

3.1.1 Účel webových aplikací

Podle Web.AppStorm vývojář Jarel Remick jednoduše a výstižně popsal smysl webové aplikace, jakožto webové stránky provádějící určité funkce. „Vždy jsem se ptal sám sebe: ‚Dělá to něco?‘ I kdyby to třeba dělalo něco malého a specifického, stále se jedná o webovou aplikaci. Google vyhledávač je také webová aplikace, ale jeho původní koncepce je stěží odlišná od telefonního seznamu, který umožňuje vyhledat jména nebo čísla“ (Remick, 2011).

3.1.2 Výhody a nevýhody webových aplikací

Uvedené pozitiva a negativa pro webové aplikace jsou řečeny podle Youngwertha (2015). Mezi hlavní výhody webových aplikací je možno začlenit:

1. Multiplatformní software, což znamená, že naše aplikace mohou fungovat na jakékoli platformě s webovým prohlížečem.
2. Obrovská komunita, díky které je možno použít velké množství volně dostupných kódů a balíčků.
3. Rychlý vývoj nástrojů, jako jsou Javascript, HTML, CSS, PHP apod.
4. Založení a vývoj na standardech, aneb není předpokladem použití proprietárního softwaru.

Naopak za nevýhody můžeme brát:

1. Rychlost, která ukazuje jak může být webová aplikace pomalejší než desktopová.
2. Jen jeden jediný jazyk, který pro vývoj webových aplikací na straně klienta (Javascript).
3. Menší kontrola nad počítačovými zdroji.

3.2 Nástroje pro tvorbu webových aplikací

Uvedené nástroje pro vývoj jmenované webové aplikace byly zvoleny s úmyslem ušetřit práci a zvýšit její produktivitu. Jsou použity aktuální verze softwaru. Vysvětlení se vztahuje spíše na méně povědomé pojmy. Do nástrojů pro tvorbu webových aplikací patří také JavaScript, jQuery, SQL, Apache, phpMyAdmin, webhosting Wedos, Photoshop CS5. Tyto nástroje nejsou dále specifikované, protože by se takováto specifikace mohla jevit příliš obecnou a známou. Naproti tomu dále vypsané nástroje jsou uvedeny velice stručně a zjednodušeně.

3.2.1 HTML5 a CSS3

HTML5 a CSS3 nám dávají nové možnosti základu pro budoucí generace webových aplikací. Dokážou umožnit takovou tvorbu webů, pomocí které je správa webu jednoduchá a navíc jsou webové stránky k uživatelům přívětivější. Výhodou HTML5 jsou nové elementy, které formulují vkládání obsahu a strukturu. Tudíž není nutné používat zdrojové kódy navíc, zásuvné moduly a jiné doplňky.

Oproti předchozím verzím umožňuje CSS3 práci s pokročilými selektory a poskytuje grafická vylepšení (Hogan, 2011, s. 16).

3.2.2 PHP 7

PHP je skriptovací jazyk vykonávaný na straně serveru vkládaný do běžného HTML kódu. Každou stránku, která obsahuje PHP skripty, server nejprve vezme a vykoná všechny příkazy v PHP, které jsou na stránce uvedené, poté pošle klientovi již čistý HTML kód, který je výsledkem běhu skriptu.

Nová verze jazyka PHP 7 s sebou přináší vyšší výkonnost, redukcí využití paměti a řadu zajímavých funkcí. Poslední dostupnou verzí je PHP 7.0.4, která byla vydána 3. března a je v tomto projektu obsažena. K dispozici je nyní možné využít Microsoft Azure (Holec, 2016).

3.2.3 Architektura MVC

Aktualizované informace, které se týkají této architektury, jsou řečeny podle nejaktivnějšího vývojáře frameworku Nette Grudla (2015). Velice efektivní softwarová architektura Model-View-Controller umožňuje zpřehlednění kódu, a tím i usnadní nadcházející implementaci. Důvodem je oddělení kódu do tří částí. Jmenovitě se jedná o kód aplikační logiky (Model), zobrazovacích dat (View) a obsluhy (Controller):

- Model znamená pro aplikaci funkční a datový základ. Veškeré akce v této aplikační logice reprezentují všechny akce uživatele. Mezi tyto akce řadíme například registrace, přihlášení, rezervace apod. Z toho vyplývá, že se v modelu nachází aplikační logika. Pokud zavoláme funkce jednotlivých rozhraní, pak je díky modelu snadné zjistit a měnit jeho stav.
- View je vrstva aplikace, známá také pod názvem pohled, díky které můžeme vidět výsledek našeho požadavku. Zobrazení výsledků požadavků, případně jejich komponentů, je dáno šablonovacím systémem.
- Controller má za úkol zpracovat všechny požadavky uživatele. Tento řadič spolupracuje s Modelem a s View. Nejprve se zavolá Model za účelem sestavení aplikační logiky. Poté proběhne vykreslení dat díky zavolání View. V aplikacích je možno Controller vidět jako presenter.

3.2.4 Nette

Framework Nette je velice vstřícný šablonovací systém. Zaměřuje se především na přehledný a úsporný kód, zabezpečení a objektový návrh. Tento moderní framework podporuje HTML5, PHP 5, AJAX, SEO a mnoho dalších koncepcí a technologií. Výhodou je také dostupnost, jelikož je zdarma a jeho licence BSD, která je otevřený zdrojový kód, se řadí mezi nejvolnější. Zabezpečení zajišťuje CSRF, XSS, session hijacking a mnoho dalších bezpečnostních nástrojů. Poslední dostupná verze Nette 2.2.12 vyšla v květnu 2016 (Grudl, 2008).

Tento framework je založen na architektuře MVC. Je nutno zmínit hlavní nejčastěji používané funkce v kontroléru.

V kontroléru se nejčastěji vyskytují hlavně formuláře. Nette rozděluje formulář na dvě části. V první části se formulář vytvoří a zadají se vstupy. Tyto vstupy se uloží do proměnné, kterou si zavolá funkce na dokončení formuláře. V této funkci se už obvykle pracuje s databází a na závěr se obnoví stránka.

Další hojně používanou funkcí v kontroléru je handler, který funguje podobně jako všeobecně známá funkce onClick za použití JavaScriptu. Hlavní rozdíl je v tom, že se implementuje prostřednictvím kontroléru, a to za použití jazyka PHP. Díky tomu může rovnou pracovat s databází. Kontrolér si sám volá funkce z knihovny jQuery, takže se do nich při implementaci nezasahuje. onClick je na rozdíl od této funkce záležitostí čistě JavaScriptu.

3.2.5 MySQL

MySQL je velmi populární databáze. Podle mnoha zdrojů je to rovněž velmi rychlý software. Nemá však tolik funkcí a možností jako některé konkurenční databázové systémy. Vybrat si vhodnou databázi je tedy klasický kompromis mezi rychlostí softwaru a jeho schopnostmi. MySQL má samozřejmě své zastánce a odpůrce.

Odpůrci například často tvrdí, že MySQL je tak rozšířená proto, že webhostingové společnosti ji často nabízejí pro hostované weby jako součást portfolia svých služeb. Kdyby se webhostingové společnosti rozhodly upřednostnit jiný software, popularita MySQL by podle jejich odpůrců podstatně klesla.

Naproti tomu zastánci MySQL tvrdí, že webhostingové společnosti nabízejí MySQL proto, že je dobrá, a kdyby existovala lepší databáze než MySQL, byla by nabízena. Zastánci MySQL prostě tvrdí, že trh si MySQL vybral – a to podle nich jasně poukazuje na její kvality (Zajíc, 2005).

3.2.6 XAMPP

Webový server obsahující základní části Apache, MySQL, PHP a Perl. Výhodou této distribuce je její jednoduchost a de facto minimální nároky na nastavení. V podstatě ho stačí jen nainstalovat (popřípadě extrahovat) a spustit. XAMPP obsahuje distribuce nejen pro Linux, ale i Solaris, Windows, MAC OS X (Pařík, 2012).

3.2.7 AJAX

AJAX (Asynchronous JavaScript and XML) je metoda k vytvoření interaktivní aplikace pro web, která provede ihned uživatelský požadavek bez obnovení stránky. AJAX kombinuje řadu programovacích nástrojů, jako je například JavaScript, DHTML, XML, XMLHttpRequest, CSS a DOM. Je součástí RIA, nového směru programování vedoucímu k vyššímu uživatelskému komfortu a funkčnosti aplikací. AJAX tedy umožňuje změnu obsahu webové stránky, aniž by uživatel použil HTTP požadavek, po kterém uživatel musí čekat než se stránka obnoví. AJAX však má i určité nevýhody, především při nevhodném užití snižuje významně použitelnost stránek. Proto je třeba, stejně jako u jiných technologií, AJAX aplikaci dobře promyslet a před nasazením i důkladně otestovat na uživatelích (Rouse, 2007).

Uvedeným uživatelským požadavkem může být také zejména použití a změna dat v databázi, což je hlavní důvod použití AJAX v této práci. Framework Nette má funkcionalitu AJAX vestavenou v jedné ze svých knihoven, a to pod názvem `nette.ajax.js`. Z předchozího odstavce vyplývá, že tato metoda značně zrychluje stránky a zpříjemňuje jejich vzhled.

Propojení Nette a AJAX využívá možnosti, které usnadňují práci a šetří kód. Konkrétně se jedná o makra, která jsou označena jako `snippet`. Umožňují za pomoci AJAX překreslit v šabloně jen tu část kódu, která je makry vymezena, což podstatně proces překreslení urychluje. Obsažený kód ve `snippetu` je překreslen pomocí formuláře nebo funkce `handler`.

3.2.8 Use Case

Use Case diagram zobrazuje chování systému tak, jak ho vidí uživatel. Účelem diagramu je popsat funkcionalitu systému, tedy to, co od něj klient očekává. Diagram vypovídá o tom, co má systém umět, ale neříká jak je to implementováno. Díky těmto faktům je u tohoto úvodního diagramu důležité si uvědomit, co má systém umět a následně provádět implementaci. Use Case diagram se skládá z případů užití (Use Case), dále aktérů (aktorů) a vztahů mezi nimi. Případ užití zjednodušeně odpovídá na otázku: "Co se děje?" Aktora je možné brát v úvahu například jako čas, systémy, lidi nebo uživatele. Případy užití byly pro přehlednost rozděleny co nejvíce do dílčích částí (Čápka, 2015).

Z tohoto Use Case diagramu vyplývá několik dalších diagramů. Tyto schémata slouží pro reprezentaci pouze jednoho případu užití. Důvodem je, že každý případ užití má odlišnou funkcionalitu od ostatních případů užití a model Use Case je tudíž vzhledem k systému velice obecný. Na jakýkoli případ užití lze shlédnout z mnoha pohledů, které spolu vzájemně souvisí. Tento pohled je reprezentován diagramem, ale jak již bylo řečeno, žádný diagram nemůže zachytit komplexní aplikaci v celku. Tedy tyto diagramy se zaměřují vždy na jeden hlavní aspekt.

4 Návrh řešení

Je pro programátora naprostou nezbytností pro úspěšné řešení implementace. Valná většina profesionálů radí těm, kdo aplikaci navrhují a zároveň implementují, aby věnovali více času pro návrh aplikace. Nejprve je třeba se primárně zaměřit na to, pro které cílové skupiny je aplikace vytvořena. Jedná se nejen o trenéry, ale také o každého člověka, který vyhledává dostupné informace o posilování a zdraví prostřednictvím internetu. Obecně to platí zejména pro kulturisty a sportovce všech odvětví. Dále je možné zaměřit se na specifické cílové skupiny. Například aktivně sportující studenti, jedinci se sedavým zaměstnáním a rodiče se zájmem o zdraví svých dětí.

Další otázkou při navrhování programu je: "Jakým způsobem budou materiály poskytovány?" Někdo by řekl, že to je vždy na první pohled jasné. Nicméně opak je pravdou. Představme si nyní následující situaci za účelem jasného vyjádření problematiky. Pro zjednodušení berme nyní v úvahu pouze sportovní jídelníčky. Vyhledávání jídelníčků bude zadáno ženou, která má váhu 51 kilogramů a 35 let. Přeje si, aby byl dokument ověřený správcem aplikace, aby byl bez suplementace a mohla si ho stáhnout v upravitelném formátu. Vstupy jsou tedy velice konkrétní a návrhář má nyní dva hlavní záměry na aplikaci, aby tato žena našla to, co hledá.

Návrhářův záměr je takový, aby aplikace nabízela dostatečné množství materiálů, ve kterých opravdu každý nalezne řadu dokumentů přesně dle svých zadaných parametrů. Lze toho docílit tak, že jídelníčky bude vkládat hodně uživatelů. V takovém případě by se po vyhledávání tato žena dočkala opravdu dlouhé řady jídelníčků, ale v tomto případě musí počítat s tím, že jich bude část hodnocena záporně. Dále se většina jídelníčků ani nezobrazí, protože nebudou ověřené. V naprosto nejhorším případě se bude jednat o dokument, který ani nemá se cvičením a zdravím nic společného a správci aplikace se k němu ještě nevyjádřili.

Návrhář se může nad aplikací zamyslet i tak, že bude hlavním požadavkem především kvalita a důvěryhodnost materiálů. V tom případě by se vkládání jídelníčků umožnilo pouze ověřeným trenérům. Vyhledat však dostatek trenérů, kteří by zdarma přispívali materiály, za které mají ve skutečném světě peníze, je velice obtížné. Především na stránkách s malou klientelou, která by byla doprovázena nízkou nabídkou vložených materiálů. V tomto případě se žena může těšit na kvalitní materiály, ale s největší pravděpodobností by se musela rozhodovat mezi malým počtem vybraných příspěvků a všem je dobře známo, že si zákazníci rádi vybírají ve velkém.

Tyto záměry jsou tedy očividně protichůdné a je třeba se soustředit buď na kvantitu nebo kvalitu sdílených dokumentů. Výsledkem této úvahy je, že se návrh provede nejprve se záměrem kvantitativním. S velkým množstvím nabízených informací přichází velké množství zákazníků. Po tomto procesu, který se zaměřuje na počet uživatelů, se bude postupně zpříšňovat vkládání materiálů až bude omezeno opravdu pouze na trenéry. Tedy z kvantitativního záměru se přejde na kvalitativní a začne hromadné třídění materiálů, jestli na stránku patří či nikoli. To, že má nějaký dokument o posilování a zdraví špatné hodnocení, neznamená, že je nutné

jej odstranit. Ba naopak slouží jako vzorový příklad, která cesta není ta správná. Uživatelé se však jistě budou zaměřovat na materiály s kladným hodnocením a na ty se záporným by přihlíželi spíše informativně.

Pojednání o návrhu aplikace vzhledem k shromažďování materiálů by mohlo být chápáno nebo definováno mnoha jinými způsoby, ale tato aplikace navazuje především na výše zmíněné myšlenky. Kvalitu aplikace nelze formulovat exaktními postupy ani empiricky měřit. Je však důležité nalézt takové řešení, které uspokojí jak uživatele, tak splní podmínky, jež náleží návrháři.

4.1 Use Case Model

Po uvedení do problematiky při navrhování aplikace je třeba zvolit metodu, která bude návrh nejlépe vystihovat. Pro tuto reprezentaci návrhu byl zvolen Use Case Model. Zobrazení modelu lze vidět na obr. 1. Každý případ užití má svůj vlastní scénář. V tomto scénáři nalezneme posloupnost po sobě jdoucích vstupů ze strany uživatele a výstupů ze strany systému. Scénář lze definovat různě. Jedna z variant je, že se uvede souhrn událostí v dokonalé podobě, tedy bez komplikací. V jiných případech se navrhuje scénář tak, aby vystihl jak kladné, tak i záporné odezvy systému vzhledem k uživatelským úkonům. V této práci je uvedena druhá varianta. V rámci této práce se uvádí diagram sekvenční, komunikační, stavový a diagram aktivit, které vyplývají z Use Case diagramu. Zmíněné diagramy jsou uvedeny pro demonstraci případu užití s názvem Hodnocení jídelníčků. Příčinou výběru tohoto Use Case je, že se jedná o jeden z hlavních aspektů aplikace a zároveň je jednoduchý na vyjádření scénáře i diagramů.

4.1.1 Aktoři

Aktor Uživatel může obecně komentovat a používat nápovědu neomezeně. Jeho další možnosti je nutné rozdělit a přiřadit aktorům, kteří od aktora uživatel dědí. Konkrétně se jedná o aktory Nezaregistrovaný uživatel a Zaregistrovaný uživatel.

Aktor Nezaregistrovaný uživatel má v aplikaci velmi omezené možnosti. Aktor Zaregistrovaný uživatel dědí aktory Nový uživatel a Stálý uživatel. V tomto pořadí má uživatel stále více možností, co se týče používání aplikace.

Aktor Admin a jeho specifikace je vymezena následně. Správu aplikace z pozice administrátora bude moci provádět nejen tvůrce aplikace, ale i pověřený trenér, jehož znalosti budou na takové úrovni, aby dokázal ověřit přínos nových materiálů. Aktor Admin je povinen kontrolovat činnost uživatelů a evidovat informace o posilovnách a trenérech.

4.1.2 Případy užití

Případy užití (Use Cases), jenž se vztahují na aktora Uživatel jsou funkce aplikace, které platí pro všechny uživatele. Konkrétně se jedná o Nápovědu a Komentování, jejichž funkce jasně vyplývá z názvu.

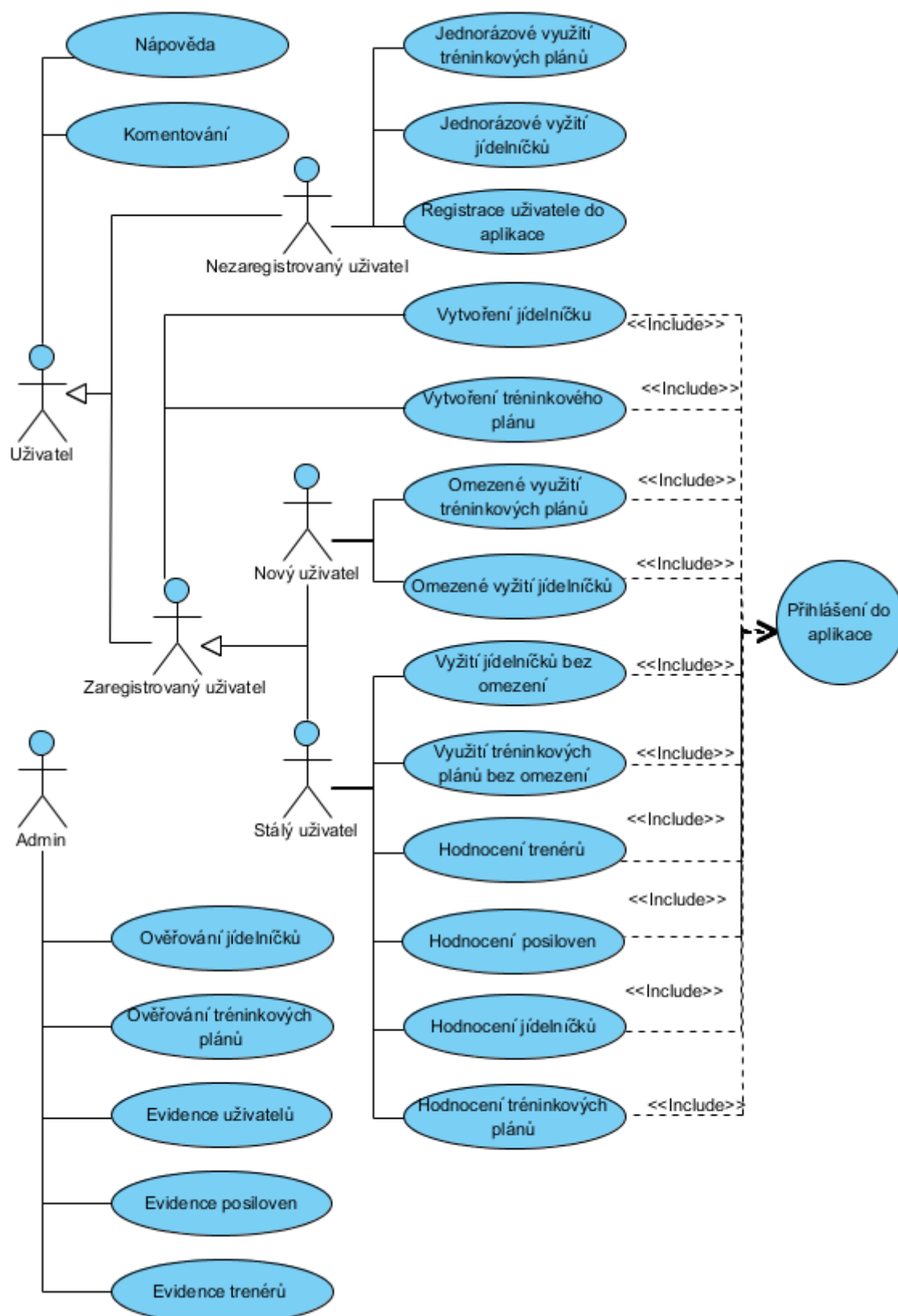
K případům užití Nezaregistrovaného uživatele platí, že při druhém zobrazení detailu kterékoli položky z možností je uživatel vyzván, aby se přihlásil, pokud chce zobrazit obsah uvedené položky. Při práci s aplikací má uživatel neustále možnost se zaregistrovat. Při komentování je uživatel povinen zadat jméno a email, čehož je v případě přihlášení ušetřen.

Následující odstavec pojednává o případech užití pro Zaregistrovaného uživatele. V případě, že je po registraci uživatel přihlášen, může libovolně vkládat nové příspěvky, co se týče jídelníčků a tréninkových plánů. Toto vkládání materiálů uživateli velice mírně zvyšuje jeho body za aktivitu v aplikaci.

Případy užití aktora Nový uživatel probíhají tak, že poprvé přihlášený uživatel disponuje pouze s jedním bodem za aktivitu na webových stránkách. Což vede k tomu, že může zobrazit celý obsah pouze u tří položek. Následně bude vyzván o zlepšení svého uživatelského hodnocení prostřednictvím vkládání materiálů. Je rovněž nutné, aby byl tento Nový uživatel přihlášen, pokud chce využít svých možností.

Případy užití aktora Stálý uživatel je definován v následujícím textu. Ve chvíli kdy uživatel dosáhne osobního hodnocení tři body a více, je mu umožněno nejen plně využívat celého obsahu souborů, ale může příspěvky i hodnotit. Po určité době veřejného využívání aplikace bude nutno tento způsob hodnocení uživatele zpřísnit na minimálních pět bodů, aby se zamezilo vkládání materiálů, jejichž obsah nemá s účelem aplikace nic společného. V první fázi publikování aplikace bude uživatelů nízký počet a tudíž budou jejich příspěvky jednoduše kontrolovatelné. Hodnotit uživatel může pouze jednou a tento krok bude nevratný. Při zmíněné veškeré manipulaci s aplikací je nutné, aby byl uživatel přihlášen. Při každém čtvrtém zobrazení materiálu přijde aktor Stálý uživatel o 0,1 bodů. Je to z toho důvodu, aby byl tento aktor nucen jednou za čas přispět novými materiály.

Doposud nezmíněné případy užití se týkají aktora Admin. Případy užití ohledně ověření dokumentů mají zajistit to, že se jedná opravdu o materiály, které odpovídají tématu aplikace. Při vkládání materiálů, které tomu neodpovídají má aktor Admin právo uživatele vymazat. Posilovny a trenéry aktor Admin vloží při implementaci aplikace a dále je bude moci evidovat pouze prostřednictvím kódu. Ve chvíli, kdy aktor Admin ověří materiály od aktora Zaregistrovaný uživatel, tak tento zákazník obdrží 1,5 bodů za aktivitu a bude moci odebírat další materiály. Pokud takto aktor Admin učiní, tak se z aktora Nový uživatel stane aktor Stálý uživatel, protože už bude disponovat se třemi uživatelskými body.



Obrázek 1: Use Case diagram

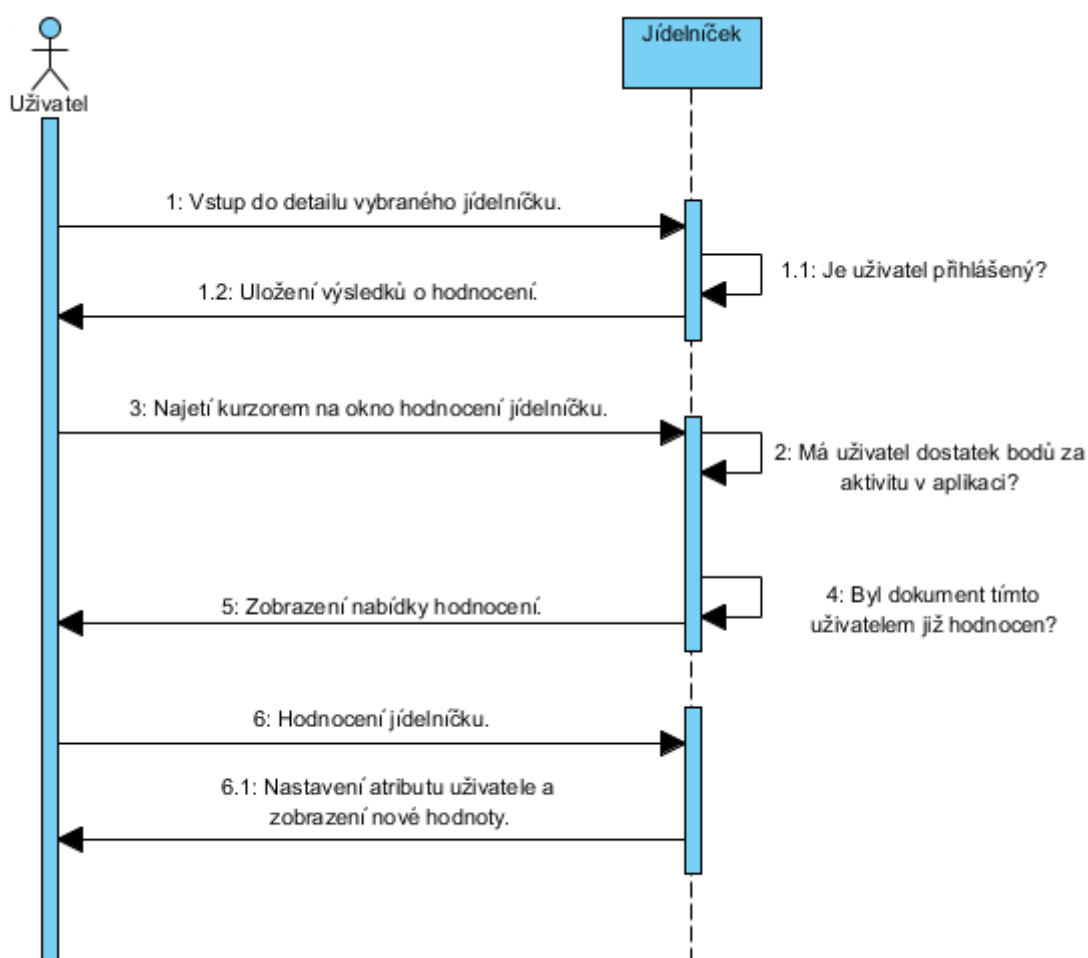
4.1.3 Scénář pro hodnocení jídelníčků

Je třeba znovu podotknout, že vše uvádíme z pohledu uživatele a zaměřujeme se zejména na to, jaký má pro tohoto uživatele případ užití význam. Scénář by tedy neměl poukazovat na to, jak systém vypadá, ale znázorňovat jeho funkcionalitu. V této charakteristice návrhu se při postupu střídá aktér se systémem. Zde konkrétně vystihují lichá čísla aktéra a sudá čísla systém. Při pohledu na scénář je patrné, že scénář nemusí projít zcela hladce, ale mohou naopak nastat komplikace díky nesplněným požadavkům systému. Odborně se pojednává o hlavních a alternativních tocích scénáře. Alternativní scénáře, jak už název napovídá, se odchyľují od hlavního scénáře, který na rozdíl od alternativního neřeší možné chyby. Uživatelem je tedy požadováno dosáhnout posledního bodu, aby systém provedl cílenou operaci. Nyní už je možné přejít k samotnému scénáři, který je v tomto případě vystižen těmito sedmi body:

1. Uživatel vstoupí do detailu vybraného jídelníčku ze škály poskytnutých jídelníčků.
2. Systém ověří, zda je uživatel přihlášený.
 - a) Pokud ano, systém je připraven k další kontrole a čeká na uživatelskou interakci.
 - b) Pokud ne, systém bude moci, po interakci s oknem hodnocení, uživateli zobrazit pouze průměrné hodnocení jídelníčku a počet hodnocení, ale neumožní uživateli hodnotit.
3. Uživatel najede kurzorem na okno hodnocení jídelníčku, které zobrazuje hodnocení v procentech.
4. Systém zkontroluje, jestli má uživatel dostatek bodů za aktivitu v aplikaci.
 - a) Pokud ano, systém přejde k poslední kontrole.
 - b) Pokud ne, systém zobrazí uživateli průměrné hodnocení a celkový počet hodnocení. Hvězdy k ohodnocení dokumentu nezobrazí.
5. Systém zjistí, jestli uživatel dokument nehodnotil.
 - a) Pokud ano, systém postupuje jako u bodu 4. b).
 - b) Pokud ne, systém uživateli zobrazí pět hvězd pro jednoduché ohodnocení materiálu.
6. Uživatel ohodnotí jídelníček.
7. Systém nastaví uživateli hodnotu, že jídelníček již hodnotil a hodnocení nebude opětovně povoleno. Následně vypočítá nový průměr hodnocení a zvýší počet hodnocení o jedničku.

4.1.4 Sekvenční diagram

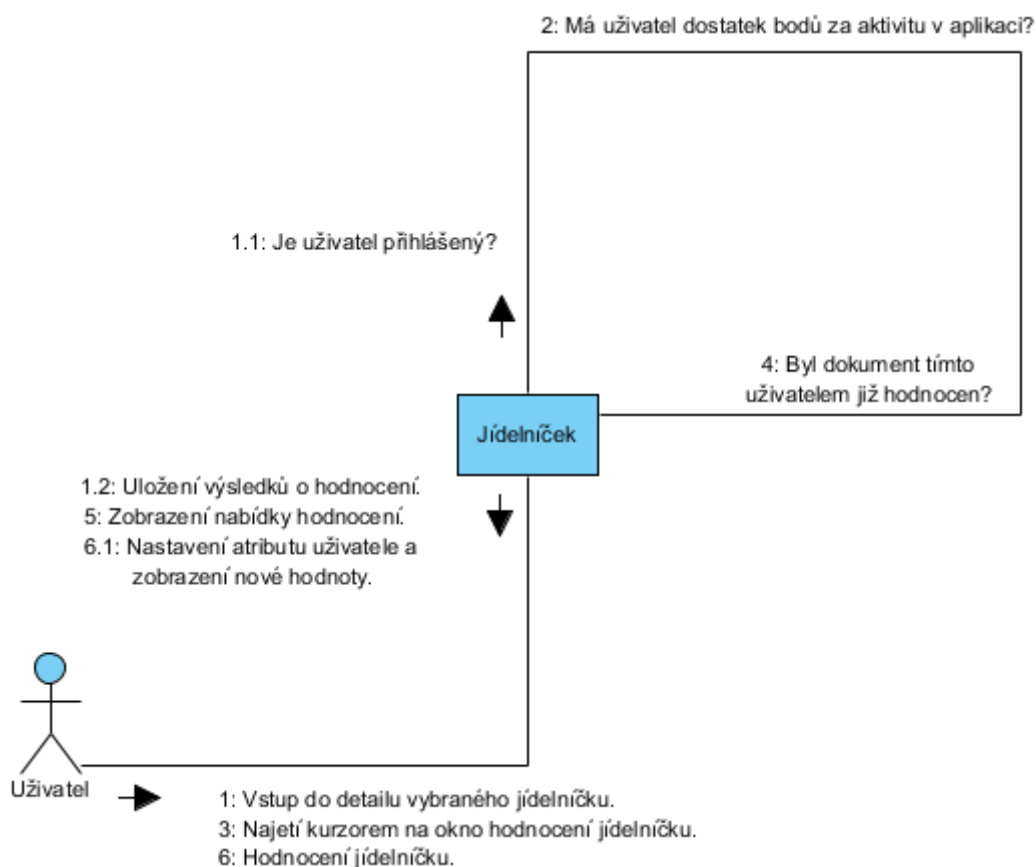
Skládá se z objektů a mezi nimi zasílaných zpráv. Diagram tedy vystihuje objekt Uživatel a jeho možnost hodnocení jídelníčku. Tento proces může provést aktor Stálý uživatel s dostatečným kreditem uživatelských bodů, a to až po přihlášení do aplikace. Zobrazení diagramu je možno shledat na obr. 2. Šipky, které vedou od aktora Uživatel k případu užití Jídelníček, jsou vstupní zprávy aktora do systému. Ukazatele v opačném směru jsou výstupy ze systému. V případě že případ užití posílá zprávu sám sobě, pak se jedná o ověření k danému vstupu.



Obrázek 2: Sekvenční diagram

4.1.5 Komunikační diagram

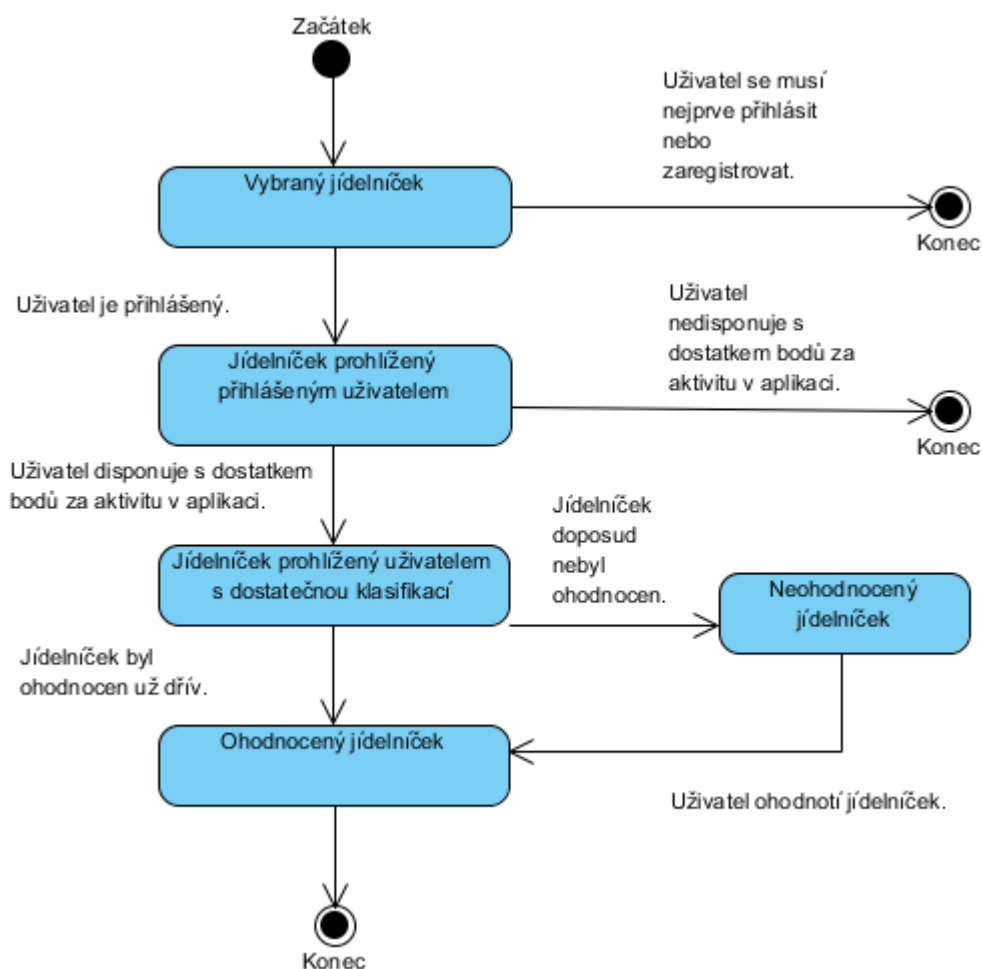
Jako u sekvenčního diagramu se zde zachycují objekty, tedy instance tříd, avšak místo čar života jsou navzájem propojeny. Díky diagramu na obr. 3 je tedy možné analyzovat komunikaci aktora Uživatel s příslušným objektem Jídelníček. Interakce je pro znázornění pořadí jako u sekvenčního diagramu číslována, ale nyní je jasně viditelné, že méně zdůrazňují časovou souvislost. Díky těmto faktům komunikační schéma jasně vyplývá ze sekvenčního diagramu. Tím, že je Use Case Model rozdělen na naprosto dílčí části, je náhled na komunikaci velice jednoduchý a pochopitelný.



Obrázek 3: Komunikační diagram

4.1.6 Stavový diagram

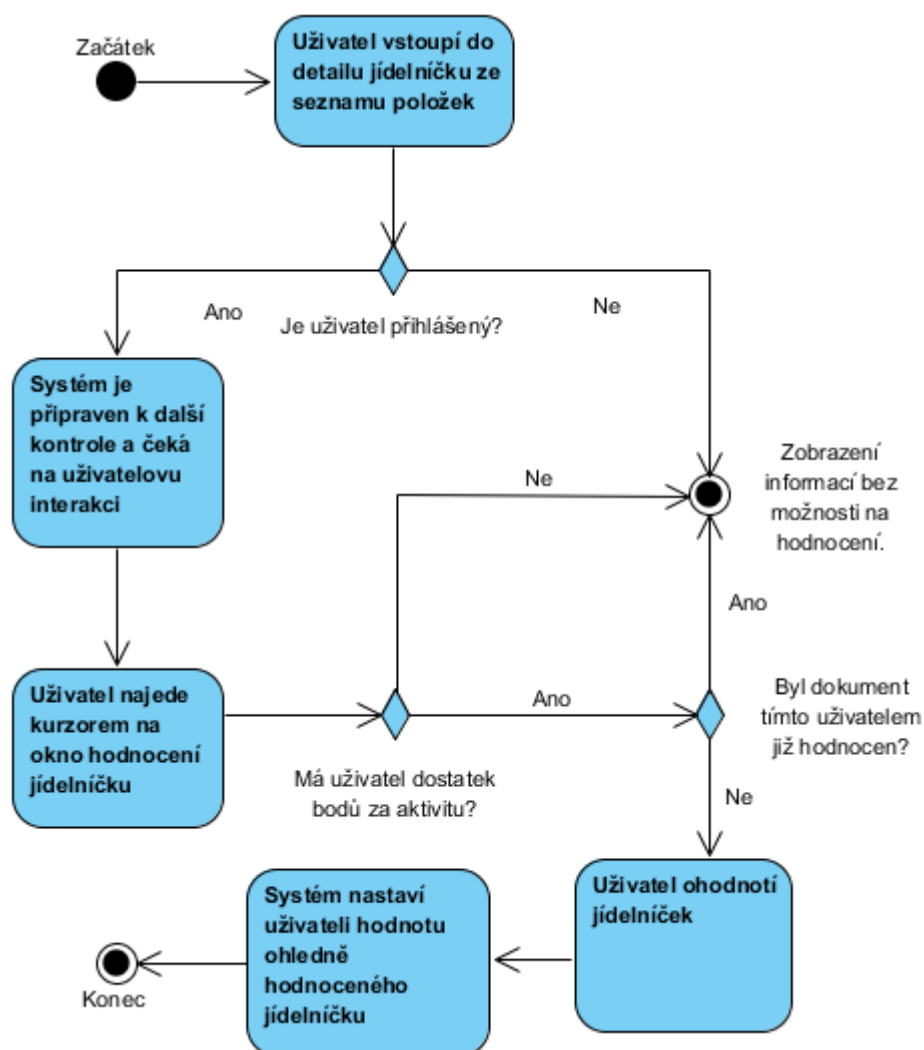
Zmíněný stavový diagram, který lze vidět na obr. 4, nám ukazuje vývoj jídelníčku při hodnocení uživatelem. Stavový diagram musí být konečný, acyklický a jeho stavy musí být jednoznačné. Schéma se skládá pouze z aktivit, nikoli z událostí, protože každý ze stavů může trvat delší dobu, kdežto událost je dána pouze určitým časem bez prodlevy. Z diagramu je patrné, že mohou nastat až tři různé konce. V prvním případě nastává konečný stav Jídelníčku ve chvíli, kdy uživatel není přihlášen nebo zaregistrován. Dále v případě, že uživatel nemá dostatečný uživatelský kredit za aktivitu v aplikaci. Požadovaný stav Jídelníčku nastává ve chvíli, kdy jsou splněny všechny předpoklady a neohodnocený jídelníček uživatel transformuje na ohodnocený.



Obrázek 4: Stavový diagram

4.1.7 Diagram aktivit

Tento diagram znázorňuje celkový průběh ohodnocení jídelníčku. Je možné jej nalézt na obr. 5. Způsob zobrazení průběhu se nejvíce podobná algoritmickému řešení programu. Diagram se větví pomocí tří různých uzlů, jejichž logika vyplývá ze scénáře případu užití. Při prozkoumání návaznosti jednotlivých aktivit je možné se dobrat ke dvěma různým výsledkům. První závěr diagramu říká, že se grafický prvek v aplikaci zobrazí bez možnosti na hodnocení. V opačném případě postupuje až do konce a změní hodnotu hodnocení o uživatelem zadaný vstup.



Obrázek 5: Diagram aktivit

4.2 ERD diagram

Tento diagram byl vytvořen za pomoci nástroje phpMyAdmin. Toto konceptuální a abstraktní znázornění dat vyjadřuje databázi, která umožňuje správu s větším množstvím dat. Databáze obsahuje jedenáct vzájemně propojených tabulek, které se skládají z řady prvků. Tyto prvky jsou specifické svými atributy.

Při vytváření diagramu bylo postupováno následujícím způsobem. Byly navrženy entity a vazby mezi nimi, tak aby pokrývaly potřeby aplikace pro uchování dat. Na základě entit a jejich vazeb byly v databázi vytvořeny tabulky. Jejich struktura je reprezentována na obr. 6. Avšak díky velikosti databáze není snadné rozpoznat jednotlivé vazby, což vede k nutnosti je uvést slovně.

Při výběru velikosti datových typů je nutné brát v úvahu raději větší velikost, kvůli plánovanému počtu prvků v tabulkách.

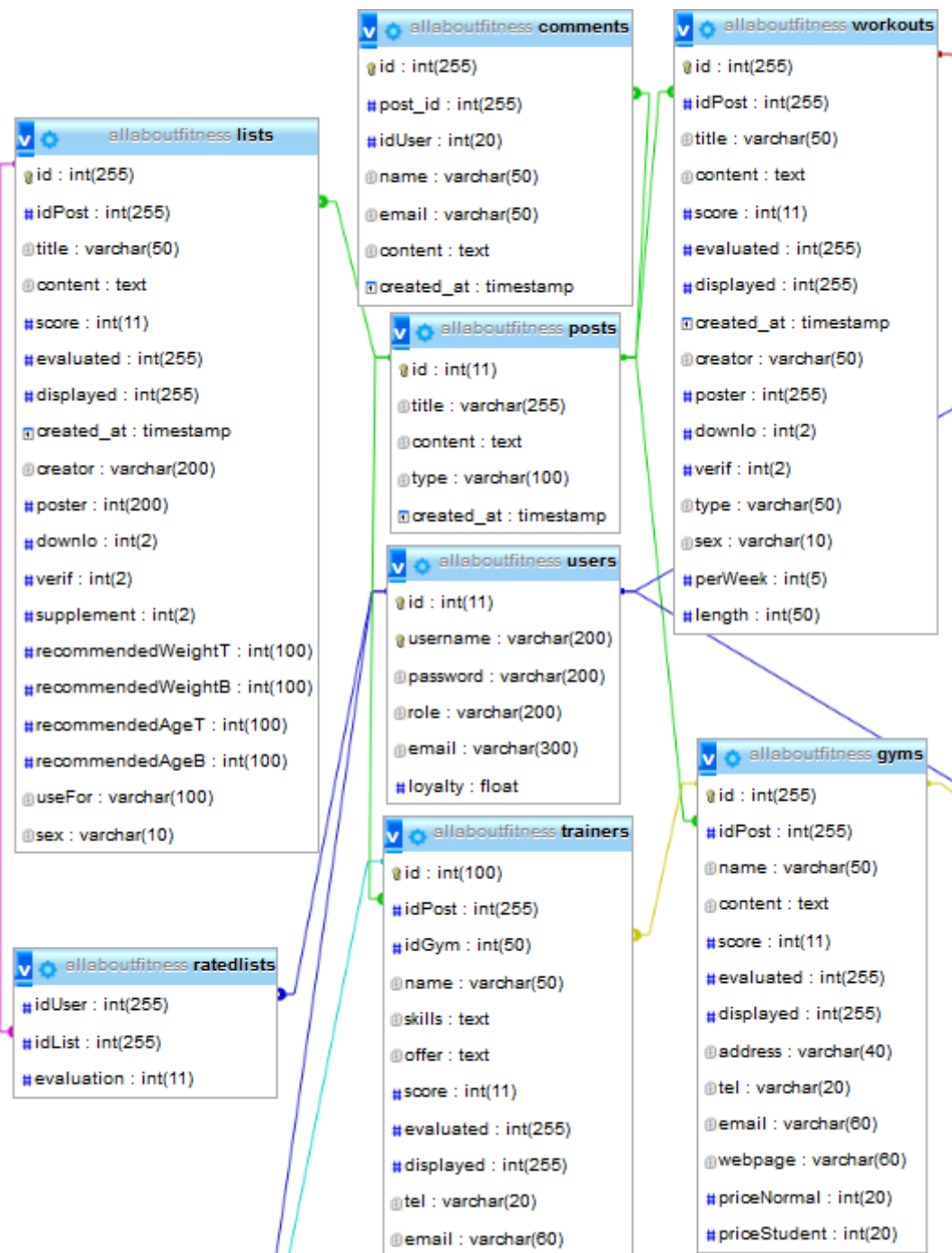
Tabulka users je v aplikaci nezbytná, jelikož reprezentuje uživatele a jejich parametry. Hlavními atributy této tabulky jsou identifikace uživatele, jeho heslo a jeho loajalita k aplikaci. Tato tabulka je propojena s tabulkami, které jsou určeny k zachování hodnot. Tuto údaje vystihují co bylo hodnoceno, kdo hodnotil a jak hodnotil. Identifikovat tento typ tabulek lze díky předponě rated. Smysl těchto tabulek je takový, že pokud již uživatel nějaký soubor hodnotil, tak se tato vazba uloží a nelze tento dokument hodnotit opakovaně. Aktivní je zde kaskádové mazání. Důvodem je, že pokud bude uživatel jednat neoprávněně, tak aby po jeho vymazání z databáze byly smazány i veškeré jeho záznamy. Výjimkou jsou komentáře, jelikož tyto zprávy v aplikaci mohou být užitečné pro další uživatele i po vymazání jejich účtu z databáze. Při pohledu na již zmíněný obr. 6 lze nalézt pouze jednu tabulku s předponou rated. Konkrétně se jedná o ratedlists. Na ostatní tabulky tohoto typu vedou jejich vazby mimo diagram, avšak jejich logika je založena na naprosto stejném principu. Důvodem je velikost ERD diagramu v poměru s velikostí stránky.

Na zmíněné tabulky s předponou rated navazují přes cizí klíče tabulky vystihující aplikaci. Jmenovitě se jedná o lists, workouts, trainers a gyms. V takto daném pořadí charakterizují jídelníčky, tréninkové plány, trenéry a posilovny. Tabulka lists obsahuje veškeré potřebné atributy k identifikaci, nalezení, hodnocení, určení a jiné specifikace jídelníčku. Obdobně je tomu s podobnými parametry u tabulky workouts. Opět je mazání řešeno kaskádově. Jako příklad lze uvést, že při smazání jídelníčku je nutné vymazat i s ním spojené hodnocení a šetřit tak paměť serveru.

Tabulky gyms a trainers se od těchto dvou tabulek liší více, jelikož vystihují reálné subjekty. Gyms obsahuje hlavně specifika k nalezení a zhodnocení posilovny. Trainers znázorňuje trenérové údaje, reference, kontakt a hodnocení. Tyto tabulky jsou vzájemně propojené, jelikož je častým úkazem, že trenéři pracují v některé z posiloven.

Každá z těchto čtyř tabulek disponuje s cizím klíčem, který odkazuje na tabulku posts. Tato tabulka znázorňuje debatu, jenž je vytvořena automaticky s vytvořením některé ze čtyř tabulek uvedených jako tabulky vystihující aplikaci. Stejně tak jako automaticky vzniká i zaniká, a to pomocí kaskádového mazání. Smazání nastává

ve chvíli, kdy je odebrán objekt, ke kterému se konverzace vztahuje. Na primární klíč této tabulky se odkazuje cizím klíčem tabulka se jménem comments. Jak už název napovídá, jedná se o tabulku vyjadřující komentáře v každém rozhovoru. Tyto komentáře zanikají kaskádovým mazáním při smazání jejich příslušné tabulky posts. Na základě této hierarchie je očividné, že tabulka s komentáři bude disponovat s nejvíce prvky.



Obrázek 6: ERD diagram

5 Implementace aplikace

V této kapitole je pojednáváno v první části o grafickém návrhu práce, který byl vytvořen jako první. V dalších sekcích jsou jednotlivě uvedeny primární části aplikace, které jsou seřazeny podle času vytvoření. Z toho vyplývá, že při čtení jednotlivých sekcí je čtenář obeznámen s postupem, který byl aplikován při implementaci aplikace.

5.1 Grafický návrh

Prvotní úkol při implementaci je navrhnout informační architekturu webové stránky. Aby byly objekty správně zařazeny do kategorií, tak je pro návrh zvoleno nejednoznačné organizační schéma. Důvod je takový, že aplikace nabízí řadu možností a to pro více cílových skupin. Při volbě jiného schématu by mohl vzniknout nepřehledný nebo chaotický web. Uvedené schéma se skládá z domovské stránky, která je ostatním částem nadřazená, a dále z jídelníčků, tréninků, trenérů, posiloven, diskuze a v neposlední řadě je možno odkázat se na návod jak stránku použít.

Je zde třeba apelovat zejména na to, jak má web na člověka působit. Proto je nedílnou součástí při navrhování vzhledu výběr barevného schématu. Předpoklad byl, aby na sebe aplikace upozornila. Ke zvýšené pozornosti se využívá odstínu žluté (#FFCC33) a černé (#FFFFFF) kombinace barev. V přírodě tato kombinace signalizuje nebezpečí, ale v případě webových stránek se jeví opravdu spíše jako signál ke zbystření.

Dominantní barvou aplikace je šedá (#0B0C0E), která vznikla díky nastavené průhlednosti černé. Jedná se o neutrální barvu. V kombinaci s moderní technologií vypadá esteticky vkusně a doplňkové barvy jí dodají kontrast. Díky těmto faktorům se jedná o schéma monochromatické.

Jelikož byly převážně zvoleny tmavé odstíny barev, tak má text bílou barvu. V grafickém návrhu se často vyskytuje průhlednost různých prvků, aby stránka nepůsobila příliš smutně. Tato průhlednost byla zvolena tak, aby bylo písmo dobře čitelné. Pozadí stránky má znázorňovat pevnost a odolnost, což jsou charakteristické rysy pro sporty jako je posilování.

5.1.1 Domovská stránka

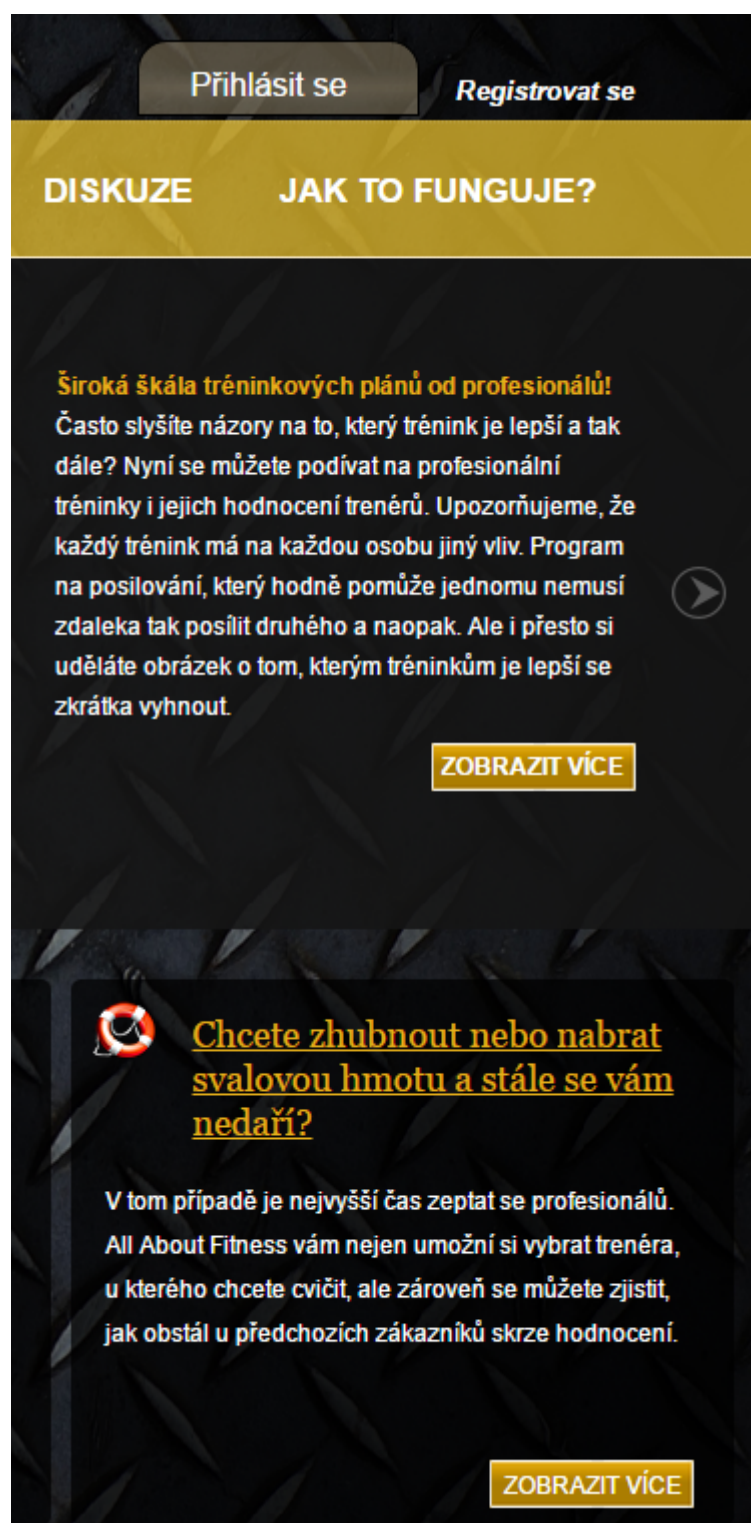
Ostatním stránkám je nadřazená. Je možno se z ní odkázat na dalších pět primárních částí aplikace. Webová stránka je rozdělena obecným způsobem na hlavičku, tělo a patičku. Pro všechny její části je toto rozdělení stejné a je definováno pomocí MVC architektury. Stránka je zobrazená na obr. 7.

Hlavička a patička je pro každou podstránku stejná, protože jsou obsažené v souboru layout, ze kterého je ostatní soubory dědí. V hlavičce se nachází logo, název aplikace, citát, tlačítka přihlášení a registrace. Citát je vybrán náhodným generátorem pomocí jazyka PHP. Dále k hlavičce patří, i když tomu tak vždy nebývá, navigace webové stránky.

Tělo webové stránky je pro každou část aplikace jiné, jelikož je obsaženo v souboru layout pouze pomocí metody content, která se předá do ostatních souborů reprezentujících jejich zobrazení. Kontroléry těchto zobrazení budou uvedeny v sekci Aplikační logika. V těle úvodní stránky je obsažen slider a šest prvků, které by měly uživatele zaujmout při vstupu do aplikace, a ze kterých se lze odkazovat do jednotlivých částí aplikace. Nejzajímavější z nich je rubrika zvaná Nejnovější příspěvky, která vypíše poslední přidaný ověřený příspěvek z tréninkových plánů a jídelníčků.

Patička stránky obsahuje odkazy na multimédia, autora a jeho autorské právo. Zároveň se k patičce připojuje tlačítko, které uživatele odkáže na formulář pro zaslání zprávy správci aplikace.

Na domovskou stránku je možné se odkázat z ikony domů, názvu stránky a motivačního citátu. Důvod je takový, aby měl uživatel jednoduchou navigaci, která zamezí pocitu, že se na webové stránce ztratil. Orientace na stránce je blíže specifikována v sekci Test navigace. Kontejnery na domovské stránce obsahují ikony, které jsou zdarma pro komerční účely, od grafika jmenem Zhouyx (2011).



Obrázek 7: Domovská stránka

5.1.2 Podstránka se seznamem položek

Toto uspořádání prvků platí pro hlavní části v navigaci, jako jsou jídelníčky, tréninky, trenéři a posilovny. U každé z těchto podstránek platí stejná pravidla, jen se pracuje s jinými daty z databáze. Obsahuje tabulku s příslušnými záznamy dle tématu podstránky, formulář pro vyhledávání, funkce pro seřazení dat v tabulce a stránkování tabulky.

Formulář pro vyhledávání se nachází v těle vlevo. Vpravo od něj je zobrazena zmíněná tabulka. Seřazení představují symboly šipek, které se nachází v prvním řádku tabulky. Jednotlivé čísla stránek seznamu jsou uvedeny pod tabulkou. Po najetí myši na některou z položek v tabulce změní celý řádek barvu, a to za efektu postupného prolnutí. U každého formuláře v aplikaci je po najetí myši textové pole zesvětleno.

Část HTML kódu, která se nachází v šabloně a slouží k vypsání tabulky, je uvedena pod tímto odstavcem. Podrobnější popis každého prvku, které jsou uspořádány ve zmíněných podstránkách, lze dohledat v kapitole Aplikační logika.

```
1 <div n:foreach="$lists as $list" class="list">
2   <tr {$count++}>
3     <td>
4       <a~n:href="Menu:listDetail $list->id" id="underno">
5         <div id="specialTitle">{$count}</div>
6       </a>
7     </td>
8     <td><a~n:href="Menu:listDetail $list->id" id="underno">
9       <div id="specialTitle">{$list->title}</div></a>
10    </td>
11    <td id="mobileTable">
12      <a~n:href="Menu:listDetail $list->id" id="underno">
13        <div id="specialTitle">
14          {$list->constructed|date: '%d.%m.%Y'}
15        </div>
16      </a>
17    </td>
18    <td class="text_right">
19      <a~n:href="Menu:listDetail $list->id" id="underno">
20        <div id="specialTitle">{$list->score}%</div>
21      </a>
22    </td>
```

5.1.3 Podstránka konkrétní položky

Z jednotlivých seznamů položek si uživatel vybírá konkrétní položky. Pro konkrétní jídelníček, tréninkový plán, trenéra nebo posilovnu platí, že mají stejné uspořádání grafických prvků. Hlavní rozdíl je v tom, že u zobrazení konkrétního jídelníčku nebo tréninku je možné stáhnout materiál ve formátu docx a dokument je zobrazen ve formátu pdf. U konkrétního trenéra nebo posilovny tyto možnosti nejsou, ale tyto záznamy jsou doprovázeny obrázky, které jsou upraveny tak, že se po najetí myši zvětší. Toto CSS zpracování je vystiženo v následujícím kódu.

```
1 .gymTrainerPic:hover{
2   width: 65%;
3   padding: 0;
4   -webkit-transition: all 0.3s;
5   -moz-transition: all 0.3s;
6   -o-transition: all 0.3s;
7   -ms-transition: all 0.3s;
8 }
```

Pro všechny tyto konkrétní položky platí, že před jejich ilustrací je tabulka, která obsahuje údaje o této položce a pod ní se nachází podrobný popis. Pod ilustrací jsou uvedeny komentáře a formulář s možností přidat nový komentář. Napravo vedle obsahu těla stránky je hodnocení právě prohlížené položky. Po najetí myši se toto okno s hodnocením rozšíří, zobrazí počet hodnocení dané položky a možnost hodnotit položku. Grafická úprava této možnosti je založena na symbolu hvězdy a patřičně upravena pomocí kaskádových stylů. Hodnotit je možné od jedné hvězdy do pěti hvězd.

5.1.4 Ostatní podstránky

Podstránky určené pro přihlášení, registrace, reference, nahrávání a mazání materiálu nejsou v grafickém návrhu blíže specifikovány, jelikož se jedná o zobrazení, které mají ve svém těle pouze formulář z kontroléru. Podobě je tomu i s podstránkami, jako je diskuze a návod. Obsahují stejné grafické prvky jako úvodní stránka, ale bez slideru.

5.1.5 Test navigace

Pro zjištění jednoduchosti orientace na stránce byl proveden kufrový test, který je ve formě 6 následujících otázek:

1. O jaký server se jedná?
 - Vystihuje logo aplikace.
2. Na jaké jsme stránce?
 - Definuje název stránky.
3. Jaké jsou hlavní sekce tohoto serveru?
 - Celkový souhrn kategorií v hlavní nabídce.
4. Jaké máme na této úrovni možnosti?
 - Aplikace nemá sekundární navigaci, ale v každé podstránce, která slouží k hlavním účelům aplikace, lze tyto možnosti definovat jako jednotlivé prvky v tabulce podstránky.
5. Kde se nacházíme v hierarchii serveru?
 - Zobrazeno v adrese URL, ale v budoucnu je vhodnějším řešením drobečková navigace.
6. Jakým způsobem můžeme vyhledávat?
 - V každé podstránce, která slouží k hlavním účelům aplikace, je textové pole, díky kterému je možno vyhledávat položky v podstránce.

5.1.6 Logo

V kapitole Nástroje pro tvorbu webových aplikací bylo zmíněno, že při tvorbě aplikace bylo použito obecně známého programu Photoshop CS5, díky kterému bylo možné jednoduše a rychle upravovat řadu ikon, pozadí nebo pracovat s barvami. Nejvíce se však uplatnil při tvorbě loga, které je možné spatřit na obr. 8. Jedná se o piktogram vystihující sportovní aktivitu posilování.



Obrázek 8: Logo

5.1.7 Menu

Reprezentuje primární navigaci. Jedná se o jednoúrovňové menu, které je na každé stránce stejné. U zařízení o velikosti 700px a více je možné se navigovat pomocí bublinkového menu, které je upraveno pomocí kaskádových stylů a JavaScriptu. Tvorba hlavní nabídky pro toto rozlišení byla inspirována navigací, která je otevřený zdrojový kód, od známého vývojáře jménem Wang (2015).

Princip v JavaScriptu je takový, že po najetí na položku v menu, která je jiná než aktuální položka, se propočítá zpoždění za ukazatelem. Toto přepočítání je umožněno díky vestavěné funkci timer. Toto zpoždění se postupně při přibližování k najetému políčku zpomaluje. Následně konečný stav mírně překročí a vrátí se do požadované polohy. Vyřešena je i změna velikosti pole, aby při vložení užšího prvku byla stejně odsazená bublina. Jedna z nejdůležitějších implementací v menu je funkce, která řeší změnu polohy. Tato funkce zobrazena zde:

```
1 | move: function (target) {
2 |     clearInterval(slip.timer);
3 |     var direction =
4 |     (slip.offsetLeft < target.offsetLeft) ? 1 : -1;
5 |     slip.timer = setInterval(function () {
6 |         sse2.mv(target, direction); }, 15);
7 | },
```

Jedná se pouze o výňatek z kódu, takže ukázka uvedené funkce není této dokumentaci ukončena. Průběh je takový, že nejprve zastaví v daný okamžik čas pomocí `clearInterval`. Dále uloží do proměnné `direction` výsledek získaný na základě ternárního operátoru, který porovná odsazení od proměnné `slip` spolu s odsazením od proměnné `target`. Kde je `slip` elementem v menu, ze kterého se chceme posunout k proměnné `target`. Tato proměnná je ve uvedená v hlavičce funkce. Následně se nastaví časovač, který je založen na opakovanému intervalu posunu. Tento interval proběhne každých 15 milisekund. Rovněž je nastaven na samotnou funkci posunu s názvem `mv`. Tato funkce je zavolána na CSS třídě s názvem `sse2`, která reprezentuje samotné menu, a následně jí udá cíl a směr.

5.1.8 Slider

Je obsažen v těle domovské stránky. Při tvorbě obrázkové prezentace byly porovnávány různé slidery prostřednictvím internetu. Nejvíce inspirativní nápad, který pojednává o způsobu tvorby slideru, byl nalezen díky grafikovi jménem Doherty (2009).

Slider funguje prostřednictvím knihovny jQuery. Jeho původní forma velmi jednoduchá a došlo k velkým grafickým změnám, díky kterým má slider jeho nynější podobu. Původně se jednalo o dvě textová tlačítka na bocích obrázku, který byl automaticky posouván pomocí intervalu. Tlačítka byla nahrazena šipkami. Dále změna v obrázkové prezentaci byla doplněna o zrychlení přechodu. Vedle obrázku uvnitř slideru byl vložen text, který vystihuje úvodní seznámení s aplikací a aktuality.

Zjednodušeně lze princip uvést následujícím způsobem. Nejprve se v JavaScriptu nastaví proměnná s názvem sliderCount na jedničku. Tato proměnná představuje číslo aktuálního snímku. Z toho vyplývá, že se při změně obrázku v prezentaci mění také, a to podle toho, ke kterému snímku se její hodnota nastaví. Každý snímek je reprezentován třídou. Tyto třídy jsou zabaleny v selektoru, který pomocí funkce siblings nastavuje aktuální snímek. Při přechodu funkce vyhledá novou třídu a následně smaže tu předchozí. Po stranách slideru jsou nastaveny šipky na přepnutí, které reprezentují funkce before a after. Při zavolání before se proměnná sliderCount sníží o jedničku a při vyvolání funkce after se naopak o tuto hodnotu zvýší. Automatický přechod je dán časovačem, díky kterému je po uplynutí časové lhůty zavolána funkce s názvem autoSlide.

Tento grafický prvek obsahuje čtyři fotografie, které byly získány na stránkách firmy Pixabay. Tato společnost poskytuje zdarma obrázky a videa, které jsou zproštěny autorských práv. Obrázky byly poskytnuty fotografem jménem Skeeze (2015).

5.1.9 Další grafické prvky

Většina prvků v aplikaci, ze kterých se uživatel může odkazovat, je uskutečněna pomocí plynulého posunutí, prolnutí v jinou barvu a nebo obojí zároveň. Tyto grafické úpravy prvků, které probíhají po najetí myši, mají delší časový interval, aby se aplikace chovala elegantně. Je to zejména právě díky tomuto nastavení odkazů, že se aplikace jeví graficky moderně a propracovaně.

Při zařízení s širší zobrazovací šířkou než 700px je hodnocení jednotlivých položek z libovolného seznamu zobrazeno CSS třídou, která vypadá jako bublina na levé straně obrazovky. Po najetí se tato bublina zvětší a zobrazí doposud skryté informace o položce. Aplikační logika je realizovaná pouze za pomoci kaskádových stylů. Stejně nastavení platí pro tlačítka, u kterých se plynule po najetí změní jejich šířka.

Grafické zpracování textového pole ve formulářích bylo vytvořeno tak, aby se vzhledově jevílo co nejvíce, jako textové pole za použití CSS frameworku Bootstrap. Jediný prvek, na který byl použit krátký výňatek tohoto framework jsou tabulky. Účelem je zajištění responzivního a příjemného vzhledu.

5.1.10 Responzivní a multiplatformní web

Framework Nette, který je v práci použit, podporuje řadu CSS frameworků, jako je například Bootstrap. To vede k závěrům, že vypracování responzivního designu by nemuselo být příliš pracné pro aktuální ani pro budoucí vývoj aplikace. Nicméně již byla implementována valná část kódu, a to ve chvíli, kdy bylo skrze vědomosti možné tyto technologie na zpracování responzivního designu použít. To vedlo k závěrům, že je lepší vzhledem k času i k délce kódu provést implementaci designu na mobilní zařízení bez CSS frameworků.

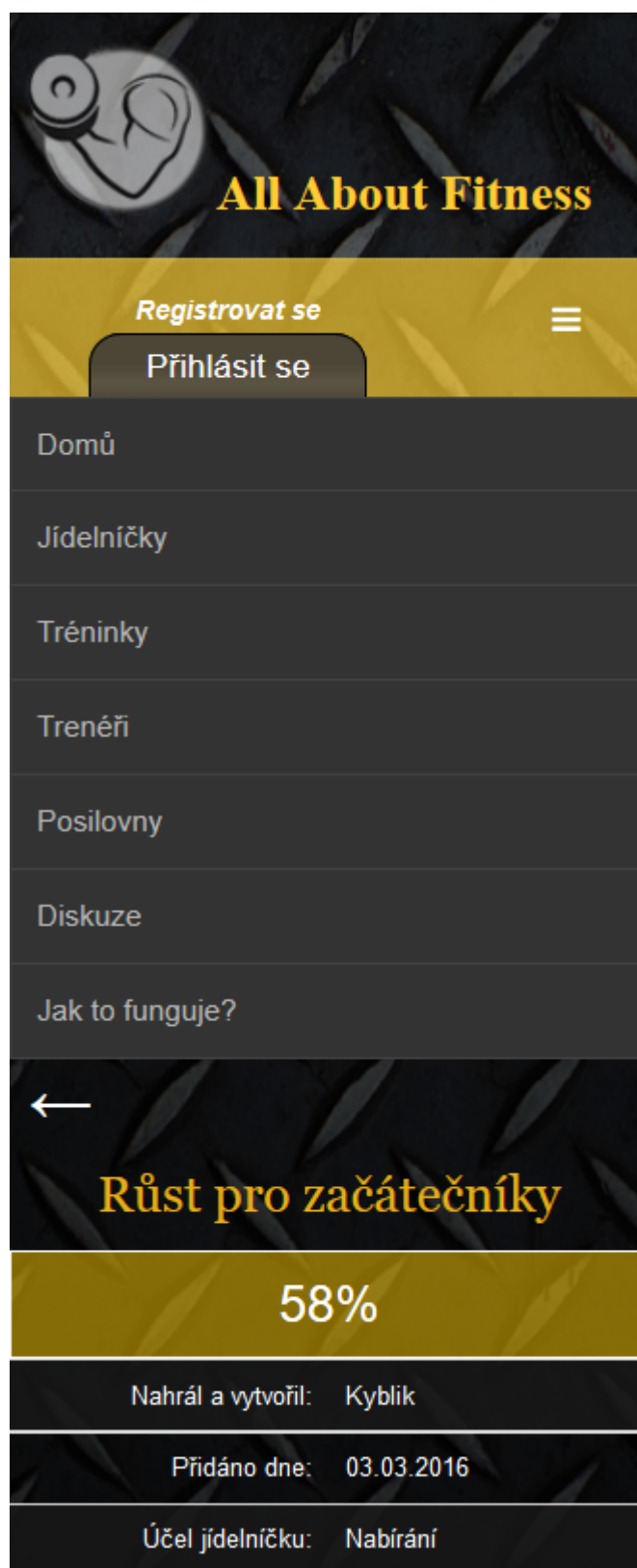
U zařízení se zobrazovací plochou 700px a méně, jako jsou zejména chytré mobilní telefony a tablety, je šířka těla stránky nastavena na sto procent, aby bylo využito co nejvíce místa. Úprava probíhá podobně u těch CSS tříd a elementů, u kterých to bylo zapotřebí. Například u hodnocení byl konkrétně použit níže zmíněný příkaz, který jasně udává, že z bubliny, kterou hodnocení doposud reprezentovalo, vznikne položka o šířce přes celou obrazovku.

```
1 @media all and (max-width: 700px){
2   .ratings {
3     position: relative;
4     margin:0;
5     float:left;
6     width:100%;
7     text-align: center;
8   }
9 }
```

Stejný princip je použit i u ostatních CSS tříd a elementů. Toto zobrazení obsahuje všechny prvky, včetně těch které jsou náročnější na vygenerování, jako je například slider. Dále obsahuje prvky, které jsou při mobilním zobrazení vzhledově upraveny na dotykové ovládání. Do této sekce spadá navigace stránky, která je zde ve formě tzn. „Hamburgerového“ menu. Úprava je dále taková, že se pomocí JavaScriptu při kliknutí na menu upraví CSS třída. Díky tomu se dočasně schová slider a zobrazí menu, aby uživatel nebyl na malém zařízení zahlcen příliš mnoho prvky. Grafický návrh pro uvedené rozlišení je na obr. 9.

Zobrazení jídelníčků a tréninkových plánů ve formátu pdf je nastaveno jinak pro zařízení s širší a s užší zobrazovací šířkou. V prvním případě je soubor zobrazen jako objekt s pluzinou (embed) a ve druhém případě jako objekt (object) s podporou docs od společnosti Google. Důvodem je, že u zařízení s užší zobrazovací šířkou nelze bez této podpory soubor zobrazit a v opačném případě není podpora na všech prohlížečích.

To, že je aplikace multiplatformní, lze tvrdit na základě výroku, že je použitelná nejen na veškerých aktuálně nejčastěji používaných operačních systémech, ale i na prohlížečích, kterými lze aplikaci zobrazit. Realizace nemusí být vždy snadná, jelikož některé prohlížeče nemusí podporovat příkazy, které jiné prohlížeče podporují.



Obrázek 9: Multiplatformní a responzivní web

5.2 Aplikační logika

Tato sekce pojednává o operacích, které byly jmenovány v prezentační logice, ale nebyla popsána jejich funkcionalita. Operace, které jsou v této sekci uvedeny, vždy pracují s databází. Jejich syntaxe je dána frameworkem Nette. V architektuře MVC spadají do částí kontrolér a model. Pokud jsou v metodách použity formuláře, tak se v šabloně nachází pouze ve formě jednořádkového zavolání formuláře z kontroléru, takže je šetřen HTML kód. Na všech následujících funkcích je použita technologie AJAX. Při manipulaci se šablonou pomocí těchto funkcí jsou v vždy použity snippets, které ohraničují HTML kód pro obnovení.

Základní funkce, jako je výběr, nahrávání a změna prvků v databázi, práce s formuláři a mnoho dalších, nejsou blíže specifikovány, jelikož jsou všeobecně známé. Tyto metody jsou nejlépe vystiženy v sekcích hodnocení a komentování, které jsou pro úplnou představu uvedeny v přílohách.

Podsekce vystihující nahrávání materiálu a podsekce určená k zobrazení obsahu konkrétní položky platí pouze pro jídelníčky a tréninkové plány. Všechny ostatní následně uvedené podsekce mají shodnou funkcionalitu pro jídelníčky, tréninkové plány, trenéry a posilovny. Z tohoto důvodu budou tyto podsekce uvedeny konkrétně pro jídelníčky.

5.2.1 Přihlašování a registrace

V případě přihlášení je v kontroléru formulář, ve kterém je nutno zadat jméno a heslo. Heslo musí mít minimálně šest znaků a uživatel může potvrdit, aby zůstal přihlášen ještě 20 minut po ukončení aplikace. Formulář pro registraci a všechny ostatní formuláře v aplikaci obsahují funkci `addProtection`, která stanoví při nečinnosti pro každý formulář časovou lhůtu. Pokud tato lhůta vyprší, formulář se stane neplatným a musí se vyplnit znovu. Po odeslání správně vyplněného formuláře se vyvolá autentizační metoda z modelu, která je zabalena v `try catch`. Toto zapouzdření, které chrání blok a ošetřuje výjimky, je možné vidět v následujícím kódu. Tato funkce ověří, jestli je zadané jméno v databázi. Pokud ano, tak na zadané heslo použije funkci `sha1`, která heslo zašifruje a takto zašifrované heslo se porovná s heslem u vyhledaného jména. Jestli je vše v pořádku, tak do sdílené proměnné uloží uživatelskou identitu, aby se parametry uživatele mohly v kterémkoli kontroléru vyvolat. Při odhlášení se zavolá funkce `logout`, která je vestavěná v knihovně Nette.

```
1 | try {
2 |     $this->user->login($values->username, $values->password);
3 | } catch (Nette\Security\AuthenticationException $e) {
4 |     $form->addError($e->getMessage());
5 | }
```

Formulář pro registraci má také vlastní kontrolér. Do formuláře je nutné zadat jméno, heslo a místo potvrzovacího boxu formulář obsahuje textové pole pro zadání emailu. Registrace má pouze tyto tři textové pole, aby se uživatel mohl zaregistrovat

rychle a nebyl registrací odrazen. Pokud je formulář správně vyplněn, tak vloží do databáze nového uživatele s následujícími údaji. Jméno a email jsou vloženy beze změny. Dále je nastavena role uživatele na návštěvníka, zašifruje se heslo pomocí funkce sha1 a přiřadí se počáteční bodové hodnocení uživatele na jedničku. Po odeslání dat do databáze se nastaví cookie, aby se uživatel nemohl následující hodinu znovu zaregistrovat a nezneužíval takovýmto způsobem možnosti aplikace. Následující kód definuje funkci, která nastane po odeslání správně vyplněného formuláře.

```
1 $this->database->table('users')->insert(array(  
2     'username' => $values->username,  
3     'password' => sha1($values->password),  
4     'role' => "guest",  
5     'email' => $values->email,  
6     'loyalty' => 1,  
7 ));  
8 $response = $this->getHttpResponse();  
9 $response->setCookie('registrovaný', 't', time() + 3600);
```

5.2.2 Hodnocení

Následující podsekcce byla podrobně vystižena v kapitole Návrh řešení. Při načtení konkrétní stránky ze seznamu proběhne metoda, která uživateli umožňuje, aby jídelníček ohodnotil. Nejprve se zavolá funkce s názvem isLoggedIn, která ověří, zdali je uživatel přihlášen. Následně, po přepočítání uživatelských bodů, projde v databázi tabulku s názvem ratedlists a zjistí, jestli uživatel zobrazený jídelníček již neohodnotil. V případě, že ho uživatel doposud neohodnotil, tak zobrazí pět hvězd v okně hodnocení, díky kterým uživatel může jídelníček ohodnotit.

Po dokončení grafické úpravy byl vytvořen handler na samotné hodnocení, který je zavolán díky odkazu ze zvolené hvězdy. Je možné udělit procentuální hodnocení 100, 75, 50, 25 a 0. Odkaz předá metodě kromě tohoto hodnocení ještě číslo aktuálně prohlíženého jídelníčku, jeho hodnocení a počet hodnocení.

Funkce na ohodnocení v kontroléru nejprve ze získaných parametrů spočítá nový průměr hodnocení dokumentu a zvýší celkový počet hodnocení o jedničku. Následně se nové údaje zapíše do tabulky jídelníčků k příslušnému záznamu. Do tabulky ratedlists se uvede, že aktivní uživatel tento jídelníček již hodnotil. Díky AJAX se nyní v šabloně překreslí vymezená část obsahu stránky s novými hodnotami.

5.2.3 Komentování

Odeslat svůj názor lze u reference, u jednotlivých položek, které náleží příslušnému seznamu a v diskuzních místnostech. Zprávu lze odeslat jako nepřihlášený uživatel, který kromě komentáře musí zadat ještě jméno a email. U vstupního pole email se za pomoci vestavěné funkce Nette ověří, zdali se jedná skutečně o email. Druhou možností je odeslat vyjádření jako přihlášený uživatel, u kterého stačí vyplnit pouze zprávu. V přílohách je uvedena druhá varianta, na které je v následujících odstavcích vysvětlen princip komentování.

Po zadání údajů se testuje, jestli byly skutečně všechny údaje vyplněny. Pokud je toto kritérium splněno, dojde k ověření, že se nejedná o spam. Aplikace zavolá metodu z modelu, která po zadání zprávy ověří, jestli se stejný komentář v dané konverzaci již nevyskytuje.

V případě, že se jedná o přihlášeného uživatele, tak může odeslat zprávu, která se již v dané diskuzi vyskytuje, ale pouze za předpokladu, že se nejedná o výskyt jeho vlastní zprávy. Důvodem jsou zprávy jako je souhlas, pozdrav a jiné vyjádření, které se často opakuje. Nepřihlášený uživatel tuto možnost nemá, a tudíž musí svoji zprávu v dané konverzaci odlišit od ostatních. Příčinou je nedůvěra v nepřihlášeného uživatele a možnost zahlcení databáze opakovaným odesláním stejné zprávy. Druhým důvodem tohoto opatření je, že se v prohlížeči Chrome po zavedení AJAX občas odeslaná zpráva odeslala dvakrát.

Při splnění těchto kritérií se, po stisknutí tlačítka s názvem Odeslat, zavolá metoda o úspěšném odeslání formuláře. Tato funkce nejprve zjistí, kdo je přihlášený uživatel a v jaké diskuzní místnosti se nachází. Tyto identifikační údaje jsou, dle pořadí, uloženy do proměnných `userId` a `postId`. Za pomoci proměnné `userId` si funkce zjistí jméno uživatele. Pokud se jedná o nepřihlášeného uživatele, nastaví mu identifikační číslo nula a jméno neznámý. Toto je preventivní opatření pro případ, že je uživatel delší dobu nečinný. Aplikace ho v takovém případě následně automaticky odhlásí, stránka se díky snippetům neobnoví celá a v tomto stavu aplikace uživatel odešle zprávu.

V posledním kroku metody se zmíněné údaje nahrají do tabulky `comments` v databázi, formulář se vyprázdní a ta část šablony, která obsahuje komentáře s formulářem, bude obnovena pomocí AJAX.

5.2.4 Vyhledávání a seřazení

V případě vyhledávání se do formuláře zadají nepovinné specifikace pro hledanou položku. Po odeslání formuláře se tyto hodnoty uloží do řetězce, ve kterém jsou jednotlivě odděleny středníkem. Tento řetězec je předán do proměnné HTTP POST, která se ověřuje při každém zobrazení stránky. Následně se obnoví uvedené zobrazení se změněnou proměnnou a řetězec se převede do pole. Pomocí jednoduchých podmínek se kontrolují hodnoty každého prvku v poli. Pokud prvek obsahuje nenulovou hodnotu, upraví se funkce pro zobrazení seznamu položek o tuto hodnotu. Příkladem je následující příkaz, který v databázi vyhledá zadaný název nebo jeho část. Pokud v tabulce, prostřednictvím vyhledávání, není zobrazena žádná položka, tak aplikace zobrazí zprávu, že nebyl nalezen žádný výsledek.

```
1 | $this->template->lists->where('title LIKE ?', "%".  
   | $filterSplit[0][0]."%");
```

Funkce seřazení je založená na podobném principu jako vyhledávání. Rozdíl je v tom, že proměnná HTTP POST s názvem sort představuje číslo, podle kterého jsou položky v seznamu seřazeny. Následující kód definuje jedno z devíti možných seřazení v tabulce jídelníčků.

```
1 | if($sort==2){  
2 |     $this->template->lists->order('title ASC');  
3 | }
```

5.2.5 Stránkování

Rozdělení seznamu položek na jednotlivé stránky umožňuje níže uvedená funkce. Nejprve se zjistí počet položek v zobrazené tabulce. Na základě toho je spočítán počet stránek. Následně se určí, na které stránce se nacházíme, která stránka je ve stránkování první, a která je poslední. Poté je zjištěno, které položky ze seznamu budou v aktuální stránce zobrazeny. Veškeré potřebné údaje se předají do zobrazení. Přejít do jiné stránky v seznamu se provádí zavoláním metody handleTurnPage, která změní proměnnou thisPage.

```
1 | $paginator = new Nette\Utils\Paginator;  
2 | $paginator->setItemCount(count($this->template->lists));  
3 | $paginator->setItemsPerPage($itemsPagi);  
4 | $paginator->setPage($currentPage);  
5 | $this->template->maximus = $paginator->getPageCount();  
6 | $this->template->thisPage = $paginator->getPage();  
7 | $maxCount = $this->template->thisPage*$itemsPagi;  
8 | $this->template->count = $maxCount-$itemsPagi;  
9 | $this->template->lists->  
10 | limit($paginator->getLength(), $paginator->getOffset());
```

5.2.6 Zobrazení obsahu konkrétní položky

Tato sekce se týká ověření uživatele. V první řadě se zjistí, jestli uživatel může daný dokument zobrazit a stáhnout. Princip je takový, že pokud je uživatel nepřihlášený, tak se po prvním zobrazení dokumentu nastaví proměnná cookie na hodnotu, která v šabloně zavolá pouze zprávu, že se uživatel musí přihlásit.

Podobně je tomu u nového uživatele. Po prvním zobrazení se nastaví hodnota této proměnné na číslo jedna, po dalším zobrazení na číslo dvě a nakonec na číslo tři. Pokud hodnota proměnné dosáhne čísla tři, tak se místo dokumentu zobrazí zpráva, že uživatel nemá dostatek bodů k zobrazení dokumentu. Uživateli je v této zprávě doporučeno, aby nahrál nový dokument nebo bude moci další materiály zobrazit za hodinu. Část tohoto zásahu do proměnné cookie je zobrazena za tímto odstavcem.

```
1 | if($this->database->table('users')->get($userId)->loyalty<3){
2 |     if (!$request->getCookie('zobrazeniObsahuPrihlaseny'.
3 |         $userId))
4 |     {
5 |         $response->setCookie('zobrazeniObsahuPrihlaseny'.
6 |             $userId, 1, time() + 3600);
7 |     }
```

Pokud bylo načtení obsahu stránky úspěšné, tak se zvýší celkový počet zobrazení u této položky o jedničku. Opět bude nastavena nová proměnná cookie, ale tentokrát tato proměnná definuje, že uvedený materiál byl tímto uživatelem nyní zobrazen. Pokud se jedná o nepřihlášeného uživatele, tak se proměnná vztahuje pouze na číslo dokumentu.

5.2.7 Nahrávání a mazání materiálu

U přihlášeného uživatele je možno při zobrazení seznamu záznamů přidat novou položku do seznamu. Formulář obsahuje povinné a nepovinné vstupní údaje o novém dokumentu. Mezi povinné údaje se řadí název materiálu, popis, účel a pohlaví. Nepovinné údaje jsou jméno tvůrce, rozsah váhy, rozmezí věku a suplementace. Za těmito údaji je tlačítko umožňující výběr dokumentu z disku. Výběr tohoto dokumentu je rovněž povinný a obsahuje dvě pravidla. První pravidlo zkontroluje, že je soubor menší než 1 MB a druhé testuje formát dokumentu. Pomocí modelu je kontrolováno, zdali je název nového záznamu unikátní. Tento model je založen na stejném principu jako antispamová kontrola pro nepřihlášené uživatele u komentování.

Jakmile je formulář správně vyplněn a odeslán, tak se údaje naformátují a odešlou do příslušné tabulky. Tímto formátováním je myšleno například to, že když uživatel nezadal jméno tvůrce dokumentu, tak se jako tvůrce nastaví přihlášený uživatel. U nevyplněných nepovinných údajů jsou nastaveny výchozí hodnoty. Následně se uživateli přidělí 0,5 bodů, za nahrání dokumentu. Na konci této funkce je nahrání souboru na disk, které je uvedeno v kódu za odstavcem. Nejprve se ověří, jestli soubor není poškozen pomocí vestavěné funkce `isOk`, která je obsažena v knihovně `Nette`. A poté je soubor vložen na disk pomocí funkce `move`.


```
1 | if($values->DOCX->isOk()){
2 |     $fileDOCX = $values->DOCX;
3 |     $fileDOCX->move('files/listsDoc/'.$lastListId.'.docx');
```

U uživatele, který má roli administrátora, lze mazat jednotlivé záznamy ze seznamu položek. Po kliknutí na odkaz s názvem smazat, který je u vybrané položky v seznamu materiálů, se zavolá nové zobrazení. Toto zobrazení je z vytvořeno z bezpečnostních důvodů, aby administrátor nesmazal něco omylem. Obsahuje název souboru a dvě tlačítka. Prvním se smazání potvrdí a druhým se může administrátor vrátit zpět.

Po potvrzení se nejprve zjistí číslo diskuze. Pomocí tohoto čísla aplikace vyhledá navazující tabulky. Následně se díky funkci delete smažou kaskádově komentáře, hodnocení, záznam a diskuze samotná. Poté jsou na serveru smazány příslušné soubory, které se záznamem souvisely, a to pomocí funkce unlink.

```
1 | $postId = $this->database->table('lists')
2 |     ->get($thisListId)->idPost;
3 | $this->database->table('posts')->where('id', $postId)->delete
   | ();
4 | unlink('files/listsDoc/'.$thisListId.'.docx');
```

6 Marketing aplikace

Pro aplikaci byla koupena doména a zřízen webhosting za pomoci webhostingových služeb firmy Wedos. Následně proběhlo nastavení serverů, importování dat do databáze a nahrání souborů na FTP server. Po osobním střetnutí s provozními různými posiloven bylo rozhodnuto, že aplikace má určitý potenciál a nebude na škodu, když se aplikace začne zveřejňovat způsobem, který osloví více zákazníků.

6.1 Sociální sítě

Levný a do jisté míry účinný způsob zveřejnění je prostřednictvím sociálních sítí. Například na stránce Facebook lze aplikaci sdílet na známých sportovních stránkách a událostech. Dále lze na sociálních sítích vytvořit vlastní novou stránku se stejným názvem. Aplikace se tímto způsobem může stát více používanou, aniž by byla nová doména dohledatelná ve vyhledávačích.

6.2 Adwords a Sklik

Další varianta je finančně nákladnější, ale u aplikací tohoto typu často účinnější. Jedná se o vytvoření marketingové kampaně, které poskytují PPC systémy od vyhledávačů. Vhodným nástrojem, který je určen pro správu marketingových kampaní, je například AdWords od společnosti Google. Další možnou variantou je Sklik od společnosti Seznam.

Aby aplikace zasáhla co nejvíce zákazníků, tak byla vytvořena marketingová kampaň pomocí Google AdWords. Typ této kampaně je vyhledávací síť s optimalizovanou obsahovou sítí a reklama pokrývá celou ČR. Při tvorbě této kampaně bylo nejdůležitější zvolit správná klíčová slova pro vyhledání aplikace, stanovit rozpočet a nastavit časy, ve kterých má reklama přes den působit. Pokud bude tato reklama prospěšná, tak bude vytvořena i u PPC systému Sklik.

7 Diskuze

7.1 Podstata a způsob navrhovaného řešení

Způsob vývoje aplikace by se dal přirovnat k agilním metodikám, které jsou založeny na iterativním a inkrementálním vývoji. Důvodem je zejména ušetřený čas při opravování kódu a při dodatečných úpravách vzhledu. To umožňuje vyšší rychlost vývoje při vytváření aplikace. Při úvaze o vývoji aplikace způsobem vodopádového modelu je důležité, aby si tvůrce uvědomil, že není snadné uvést některou z fází tvorby do dokonalosti. Zejména v případech, že se jedná o aplikaci, která je určena veřejnosti a ne soukromému sektoru. To je hlavní důvod, proč se vývoj aplikace odchyloval od vodopádového modelu.

Tvorba aplikace proběhla na základě konzultací s budoucími uživateli, vedoucím práce a s pracovníky Ústavu informatiky. Tyto diskuze byly příčinou mnoha změn a zásahů jak do implementace, tak i do samotného návrhu. Průběžná konzultace s různými budoucími zákazníky, jako jsou trenéři, provozní posiloven nebo nadšenci do posilování, umožňuje po jednotlivých krocích zlepšovat přehlednost a ovladatelnost programu. Vývojář tedy ví, jaká je představa zákazníků. Může se stát, že se tvrzení zákazníků bude lišit. Například jednomu zákazníkovi bude na první pohled více vyhovovat hodnocení dokumentu ve stupnici od jedné do deseti, ale druhému zase v procentech od jedné do sta. Na základě tohoto poznatku bylo důležité se uvědomit, které požadavky zákazníků se shodují. Řada doporučení od vedoucího práce či pracovníků Ústavu informatiky vedla při implementaci k šetření paměti, lepšímu uspořádání prvků a kratšímu kódu. Rozdíl byl tedy v tom, že diskuze s uživatelem vedla spíše ke zlepšení vzhledu webové stránky a doporučení odborníků spíše ke zlepšení aplikační logiky.

V obou případech je důležité podotknout, že tímto způsobem lze inovovat aplikaci i v budoucnu. Zároveň díky tomuto postupu stránky disponují s mnoha vlastnostmi, ale i přes tuto širokou škálu možností je třeba neustále produkt zlepšovat a modernizovat, za účelem spokojenosti uživatelů.

7.2 Porovnání výsledků s podobnými aplikacemi

Pro porovnání aplikace s webovými stránkami, které jsou založeny na podobném principu, lze uvést zejména stránky Bodybuilding, Kalorické tabulky a Česko-Slovenskou filmovou databázi.

První web, z výše uvedených stránek, je spojením široké nabídky tréninkových plánů s internetovým obchodem. Graficky se jedná o velmi propracovaný web, ale nabídka tréninkových plánů je velmi omezená. Principiálně nabídne tréninkový plán toho trenéra, který má nejlépe zaplacenou reklamu, takže i když zadáte své parametry na vygenerování vhodného tréninkového plánu, tak vám s nejvyšší pravděpodobností předloží nevhodný trénink.

Další výše uvedená webová aplikace ušetří čas na vytvoření jídelníčku. Tato aplikace je velmi úspěšná a hodně používaná. I když může být takto nápomocná, tak vytvoření propracovaného jídelníčku zabere každý týden přibližně půl hodiny času. Tento čas je dostatečně klíčový k tomu, aby dal důvod k inovaci a vytvoření nové aplikace, která bude časově úspornější.

Poslední ze zmíněných webů se zabývá popisem, hodnocením a komentováním filmů. I když se nejedná o aplikaci zabývající se posilováním, tak je této práci nejpodobnější. Důvodem je, že tyto parametry zadávají sami uživatelé a díky tomu stránka disponuje s jistou důvěryhodností, kterou uživatelé vytvořili svými subjektivními názory. Tato práce se zakládá na stejném principu, ale s tím rozdílem, že se zabývá posilováním. Aplikace překonává tento web tím, že kromě možnosti zadávat popis, hodnocení a komentování jídelníčků a tréninkových plánů nabízí i jejich dokumentaci, což umožňuje jejich další úpravu a rozšiřování.

7.3 Další možné rozšíření aplikace

V hlavní navigaci aplikace lze přidat zcela novou sekci s názvem Uživatelé, která zobrazí seznam uživatelů a umožní vyhledání konkrétního uživatele. U jednotlivých profilů uživatelů je vhodné přidat možnost, aby nahráli porovnání své fyzické stavby těla před posilováním se svými dosavadními sportovními výsledky, a to ve formě dvou obrázků.

U každého trenéra je vhodné, za pomoci propojení tabulek v databázi, aby mohli zobrazit seznam uživatelů, které trénují.

Z důvodů bezpečnosti bude v budoucnu žádoucí změnit ukládání proměnných, které jsou na serveru uloženy prostřednictvím HTTP cookie. Jedním z trendů, které se používají je standard pro bezpečný přenos JWT (JSON Web Token). Dále je příhodné před zobrazením nebo stahováním souboru ověřit uživatele prostřednictvím testu Captcha, jestli se nejedná o program na dolování dat z databáze.

Pro jednodušší přihlašování do aplikace je v budoucnu vhodné umožnit přihlášení prostřednictvím sociálních sítí.

8 Závěr

Cílem práce bylo vytvoření webové aplikace umožňující širokou škálu možností z hlediska posilování a zdraví. Webová stránka je nyní veřejně dostupná. Vytvořená aplikace umožňuje zobrazit seznam jednotlivých položek z databáze a jejich obsah. V době odevzdání tato práce disponuje s dostatečným množstvím jídelníčků a tréninkových plánů pro zavedení aplikace, přičemž jejich množství neustále přibývá. V databázi aplikace jsou uloženy jen trenéři a posilovny, se kterými proběhla domluva o jejich zveřejnění v aplikaci, avšak v následujících měsících je plánováno zapojit většinu takovýchto subjektů z celého Brna.

Aplikace, prostřednictvím omezeného množství trenérů v databázi, umožňuje zvyšovat klientelu těchto instruktorů, protože jsou v aplikaci uveřejněni spolu s jejich údaji. Toto tvrzení platí i pro posilovny. V budoucnu bude, dle původního záměru, správa tréninkových plánů a jídelníčků čistě v režii trenérů, ale to až ve chvíli, kdy bude aplikace disponovat větším množstvím všech položek v databázi.

Při zobrazení seznamu lze rychle vyhledávat konkrétní položky, a to dle parametrů, které byly uvedeny v kapitole Cíl práce. V aplikaci je možnost, dle daného oprávnění uživatele, konkrétní položky ze seznamu zobrazit, hodnotit a komentovat. Tyto operace se provádí bez obnovení stránek za pomoci AJAX. Následně byla pro uživatele vytvořena možnost, aby mohl vkládat nové dokumenty. Dále byl pro uživatele vytvořen profil o statistikách, které představují počty jednotlivých úkonů na webové stránce. Pro nové uživatele byla vytvořena příručka, jak v aplikaci postupovat a případně mohou kontaktovat správce.

Aplikace je responzivní, což znamená, že ji lze bez jakýchkoliv komplikací používat i na mobilních telefonech a tabletech. Forma vzhledu uvedených zařízení se zobrazí při rozlišení s šířkou 700px a méně, což je standardní rozmezí u CSS frameworků a zahrnuje i obvyklou šířku obrazovek mobilních telefonů 320px. U většího rozlišení než 700px je použito desktopové zobrazení aplikace.

Byla vytvořena marketingová kampaň na Google AdWords, ale jelikož se jedná o aplikaci, která je zveřejněná na zcela nové doméně, lze očekávat požadovaný účinek z marketingové kampaně až za určitou dobu.

Tato práce byla otestována řadou studentů Provozně ekonomické fakulty v Brně Mendelovy univerzity, kteří se zajímají o posilování a zdraví. Tito studenti se v aplikaci rychle zorientovali a začali neprodleně dostupných materiálů využívat.

Díky této práci došlo k prohloubení znalostí tvůrce, které může ihned uplatnit v praxi. Jedná se zejména o schopnost ovládat do jisté míry framework Nette, který je na trhu práce, zejména v ČR, častým požadavkem při programování webových aplikací. Při tvorbě této práce bylo významné, že tvůrce získal důležité kontakty v odvětví posilování a zdraví.

Praktická část bakalářské práce dle předchozích odstavců splňuje veškeré body, které byly plánovány a disponuje řadou možností, které byly implementovány mimo zadání.

9 Seznam zdrojů

- SCHEER, RODDY. *Have Fruits and Vegetables Become Less Nutritious?* [online]. 2011 [cit. 2015-12-17]. Dostupné z: <http://www.scientificamerican.com/article/soil-depletion-and-nutrition-loss>.
- DURČÁKOVÁ, KATARÍNA. *Mozek se obnovuje. A ne, že ne.* [online]. 2015 [cit. 2015-05-11]. Dostupné z: <https://www.online.muni.cz/vite/6098-mozek-se-obnovuje>.
- ROUSE, MARGARET. *Web application definition* [online]. 2011 [cit. 2015-12-16]. Dostupné z: <http://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>.
- REMICK, JAREL. *What Is a Web App? Here's Our Definition* [online]. 2011 [cit. 2015-12-16]. Dostupné z: <http://web.appstorm.net/general/opinion/what-is-a-web-app-heres-our-definition>.
- YOUNGWERTH, ANDREW. *What are the advantages and disadvantages of web based application development vs. desktop application development?* [online]. 2015 [cit. 2015-12-16]. Dostupné z: <https://www.quora.com/What-are-the-advantages-and-disadvantages-of-web-based-application-development-vs-desktop-application-development>.
- HOGAN, BRIAN P. *HTML5 a CSS3 : výukový kurz webového vývojáře*. Vyd. 1. Brno: Computer Press, 2011 [cit. 2015-12-16], 272 s. ISBN 978-80-251-3576-1.
- HOLEC, MIROSLAV. *Vyzkoušejte si nejnovější release PHP 7.0.4* [online]. 2016 [cit. 2016-04-10]. Dostupné z: <https://www.miroslavholec.cz/blog/vyzkoustejte-si-nejnovejsi-release-php-7-0-4>.
- GRUDL, DAVID. *Seznámení s Nette Frameworkem* [online]. 2008 [cit. 2015-12-15]. Dostupné z: <https://doc.nette.org/cs/2.3/getting-started>.
- GRUDL, DAVID. *MVC aplikace a presentery* [online]. 2015 [cit. 2015-12-15]. Dostupné z: <https://doc.nette.org/cs/2.3/presenters>.
- ZAJÍC, PETR. *MySQL (1) - pestrý svět databází* [online]. 2005 [cit. 2016-03-10]. Dostupné z: http://www.linuxsoft.cz/article.php?id_article=731.
- PAŘÍK, TADEÁŠ. *XAMPP* [online]. 2012 [cit. 2016-04-17]. Dostupné z: <https://wiki.ubuntu.cz/xampp>.
- ROUSE, MARGARET. *Ajax (Asynchronous JavaScript and XML)* [online]. 2007 [cit. 2016-05-02]. Dostupné z: <http://searchwindevelopment.techtarget.com/definition/Ajax>.

- ČÁPKA, DAVID. *2. díl - UML - Use Case Diagram* [online]. 2015 [cit. 2016-04-17]. Dostupné z: <http://www.itnetwork.cz/navrhove-vzory/uml/uml-use-case-diagram/?all-comments#komentare>.
- ZHOUYX, BEN. *Free Website icons* [online]. 2011 [cit. 2016-05-15]. Dostupné z: <https://www.iconfinder.com/iconsets/free-website-icons-by-benbackman>.
- WANG, JOHN. *Menu Code: Horizontal menu codes* [online]. 2015 [cit. 2016-05-14]. Dostupné z: <http://www.menucool.com/horizontal/menu-code?menu=2&skin=2&p=1>.
- DOHERTY, NIALL. *Coda-Slider 2: Customizable jQuery Slider* [online]. 2009 [cit. 2016-05-14]. Dostupné z: <http://webresourcesdepot.com/coda-slider-2-customizable-jquery-slider>.
- SKEEZE, WILLIAM. *Skeeze pictures* [online]. 2015 [cit. 2016-05-15]. Dostupné z: <https://pixabay.com/cs/users/skeeze-272447>.
- BODYBUILDING. *Training* [online]. 2016 [cit. 2016-05-15]. Dostupné z: <http://www.bodybuilding.com/fun/training.html>.
- KALORICKÉ TABULKY. *Jídelníček* [online]. 2016 [cit. 2016-05-15]. Dostupné z: <http://www.kaloricketabulky.cz/jidelnicek-bez-registrace.php>.
- ČESKO-SLOVENSKÁ FILMOVÁ DATABÁZE. *Žebříčky* [online]. 2016 [cit. 2016-05-15]. Dostupné z: <http://www.csfd.cz/zebricky/nejlepsi-filmy>.
- ALL ABOUT FITNESS. *Vítejte na našich stránkách* [online]. 2016 [cit. 2016-05-15]. Dostupné z: <http://www.allaboutfitness.cz>.

Přílohy

A Hodnocení

```
public function handleRate($thisListId,$rate, $actualRating,
$actualEvaluated){
    if ($this->getUser()->isLoggedIn()){
        $userId = $this->user->getIdentity()->getId();
    }
    $posterId = $this->database->table('lists')->get(
        $thisListId)->poster;
    $thisUserLoyalty = $this->database->table('users')->get(
        $posterId)->loyalty;
    $verif = $this->database->table('lists')->get($thisListId
    )->verif;
    $actualRating = (($actualRating*$actualEvaluated)+$rate)
        /($actualEvaluated+1);
    $actualEvaluated += 1;

    if ($this->getUser()->isLoggedIn()){
        if(($userId==3)&&($verif!=1)){
            $this->database->table('users')
                ->where('id', $posterId)
                ->update(array(
                    'loyalty' => ($thisUserLoyalty+1),
                ));
            $this->database->table('lists')
                ->where('id', $thisListId)
                ->update(array(
                    'verif' => 1,
                ));}
    }

    $thisUserLoyalty = $this->database->table('users')->get(
        $posterId)->loyalty;
    if($thisUserLoyalty>5){
        $this->database->table('users')->where('id=?',
        $posterId)
            ->update(array(
                'loyalty' => 5,
            ));}

    $this->database->table('lists')
        ->where('id', $thisListId)
        ->update(array(
            'score' => $actualRating,
            'evaluated' => $actualEvaluated,
```

```
    ));

    $this->database->table('ratedlists')
        ->insert(array(
            'idUser' => $userId,
            'idList' => $thisListId,
            'evaluation' => $rate,
        ));

    if ($this->isAjax()) {
        $this->redrawControl("rateSnip");
    }else{
        $this->redirect("Menu:listDetail", $thisListId);
    }
}
```

B Komentování

```
protected function createComponentCommentLogForm()
{
    $form = new Form;
    $form->getElementPrototype()->class("ajax");

    $postId = $this->database->table('posts')->get(
        $this->database->table('lists')->get(
            $this->getParameter('listId'))->idPost);
    if ($this->getUser()->isLoggedIn()){
        $userId = $this->user->getIdentity()->getId();
    }else{
        $userId = 0;
    }
    $messages = $this->database->table('comments')->where('
        post_id=? AND idUser=?', $postId, $userId);

    $form->addTextArea('content')
        ->setRequired("Zadejte text do prázdného pole")
        ->addRule(callback('App\Model\UniqueModel', '
            messageChecker'), '', $messages)
        ->setAttribute("placeholder", "Komentovat");

    $form->addSubmit('send')
        ->setAttribute("placeholder", "Odeslat")
        ->setAttribute("class", "ajax");

    if ($this->isAjax()) {
        $this->redrawControl("commentSnip");
    }

    $form->addProtection();
    $form->onSuccess[] = array($this, '
        commentLogFormSucceeded');
    return $form;
}
```

```
public function commentLogFormSucceeded($form, $values)
{
    $postId = $this->database->table('posts')->get(
        $this->database->table('lists')->get(
            $this->getParameter('listId')->idPost);

    if ($this->getUser()->isLoggedIn()){
        $userId = $this->user->getIdentity()->getId();
        $thisUser = $this->database->table('users')->get(
            $userId);
        $thisUserPoster = $thisUser->username;
    }else{
        $thisUserPoster = "neznamy";
    }

    $this->database->table('comments')->insert(array(
        'post_id' => $postId,
        'idUser' => $userId,
        'name' => $thisUserPoster,
        'email' => $thisUser->email,
        'content' => $values->content,
    ));

    $form->setValues(array(), true);
    if ($this->isAjax()) {
        $this->redrawControl("commentSnip");
    }else{
        $this->redirect('this');
    }
}
```