



Pedagogická
fakulta
Faculty
of Education

Jihočeská univerzita
v Českých Budějovicích
University of South Bohemia
in České Budějovice

Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky

BAKALÁŘSKÁ PRÁCE

Interaktivní webové stránky pro testování žáků z matematiky na ZŠ

Vypracoval: Václav Kandus
Vedoucí práce: PaedDr. Petr Pexa, Ph.D.

České Budějovice 2014

Prohlášení

Prohlašuji, že svoji bakalářskou práci na téma Interaktivní webové stránky pro testování žáků z matematiky na ZŠ jsem vypracoval samostatně pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě, elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách, a to se zachováním mého autorského práva k odevzdanému textu této kvalifikační práce. Souhlasím dále s tím, aby toutéž elektronickou cestou byly v souladu s uvedeným ustanovením zákona č. 111/1998 Sb. zveřejněny posudky školitele a oponentů práce i záznam o průběhu a výsledku obhajoby kvalifikační práce. Rovněž souhlasím s porovnáním textu mé kvalifikační práce s databází kvalifikačních prací Theses.cz provozovanou Národním registrem vysokoškolských kvalifikačních prací a systémem na odhalování plagiátů.

V Českých Budějovicích 27. 7. 2014

.....

Děkuji tímto panu PaedDr. Petru Pexovi, Ph.D. za odborné vedení mé práce, cenné připomínky a čas, který mi při zpracování bakalářské práce věnoval.

Abstrakt

Cílem této bakalářské práce je vytvoření interaktivních webových stránek, jež budou zaměřeny na procvičování a testování žáků základních škol z vybraných partií matematiky. Webová aplikace jim dává prostor nejen k procvičování svých dovedností v patnácti kategoriích o třech obtížnostech a přihlašování se do pracovních skupin, ale dále i k vypracovávání a sledování úspěšnosti v testech, které pro ně vytvořil jejich vyučující. Učitelům aplikace umožňuje od již zmíněného vytváření testů a skupin i vkládání vlastních příkladů a monitorování některých statistických ukazatelů, jako například toho, jak si žáci při jimi zadaných úkolech vedli.

První část práce se zabývá analýzou obdobně zaměřených stránek a jejich porovnáváním. Druhá část je poté zaměřena na analýzu a návrh samotné aplikace a popisuje stěžejní části jejího vývoje. Následující část se věnuje možným bezpečnostním hrozbám a způsobům, jak jim předejít. Ve čtvrté části práce je popsána implementace systému, jež stručně charakterizuje použité technologie a nástroje při vytváření webové aplikace. Závěrečná část práce popisuje ověření vytvořené aplikace na vybraných základních školách a víceletých gymnáziích.

Klíčová slova

Interaktivní webová aplikace, Online procvičování a testování matematiky, Analýza, návrh a implementace systému

Abstract

The aim of this bachelor thesis is to create particular interactive web pages that would be focused on both the practice and the testing of elementary school pupils in selected areas of mathematics. This web application enables the pupils not only to practise their skills in 15 different categories comprising three levels of difficulty and to access their work groups, but also to complete assigned tests created by their teacher and to follow their progress in them as well. Apart from the test and the group creation, the application also enables the teachers to put in their own tasks and exercises and to monitor certain statistic indexes, for instance how well the pupils performed the assigned tasks.

The first part of the thesis deals with the analysis of similar web pages and their comparison. The second part is then focused on the analysis and the design of the application itself and further describes the crucial parts of its development. The following, i.e. the third, part is devoted to possible security threats and how to prevent them. The system implementation is described in the fourth part which also briefly characterizes all of the used technologies and tools when creating the web application. The final part discusses the use of this application at selected elementary and grammar schools.

Klíčová slova

Interactive web pages, Online practising and testing exercises from mathematics, Analysis, design and implementation of web application

Zde vložit zadání celkem 3 strany.

Obsah

1 Úvod	12
1.1 Cíle práce	13
1.2 Metody práce	13
2 Analýza webových stránek	14
2.1 Online cvičení	16
2.2 Matematika hrou	19
2.3 Pocitejme.cz	21
2.4 Procvičuj.cz	23
2.5 Testy a kvízy on-line	25
2.6 Shrnutí	28
3 Analýza a návrh systému	29
3.1 Specifikace systémových požadavků	29
3.1.1 Popis systému	29
3.1.2 Funkční požadavky na systém	30
3.1.3 Nefunkční požadavky na systém	32
3.2 Případy užití	33
3.2.1 Popis prvků diagramu	33
3.2.2 Diagramy případu užití systému	35
3.3 Diagram aktivit	41
3.3.1 Diagramy aktivit aplikace	43
3.4 Diagram tříd	44
3.4.1 Diagram tříd aplikace	46
3.5 Datový model	47
3.5.1 Entitně – relační model aplikace	47
3.6 Drátový model	53
3.6.1 Návrh webu	54
4 Zabezpečení	55
4.1 Hlášení chyb	55

4.2	Zakázání nebezpečných funkcí	56
4.3	Ukládání hesel	56
4.4	SQL Injection	57
4.5	XSS	58
4.6	Direktiva registr_globals	58
4.7	Soubor .htaccess	59
5	Implementace	60
5.1	Použité technologie	60
5.1.1	HTML	60
5.1.2	CSS	61
5.1.3	PHP	62
5.1.4	Smarty	62
5.1.5	MySQL	64
5.1.6	JavaScript	64
5.1.7	MathJax	65
5.2	Jednotkové testování	66
5.2.1	PHPUnit	67
5.3	Verzování	67
5.3.1	Git	69
6	Ověření v praxi	70
6.1	Gymnázium Strakonice	70
6.2	Gymnázium J. V. Jirsíka v Českých Budějovicích	70
6.3	Základní škola 9.května v Sezimově Ústí	71
6.4	Základní škola a mateřská škola Domažlice	71
6.5	Shrnutí	72
7	Závěr	73
	Použitá literatura a zdroje	74
	Seznam obrázků	75

Seznam tabulek	77
A Příloha	78

1 Úvod

V posledních letech se zvyšuje počet výukových programů a webových aplikací, které lze využít přímo ve vyučovacích hodinách různých předmětů, jenž mohou přispívat ke zvýšení efektivnosti výkladu učitele či k získání větší pozornosti žáků. Praktickou částí této práce je právě vytvoření takovéto aplikace ve formě interaktivních webových stránek pro procvičování a zkoušení matematiky na úrovni 2. stupně základní školy. Webová aplikace je umístěna na adrese <http://www.vypoctito.cz>.

První část práce se věnuje analýze a porovnání webových stránek, jež umožňují online procvičování matematiky. Pro zařazení do tohoto porovnávání musely nalezené webové stránky splnit několik základních kritérií - zaměření obsahu stránek na žáky 2. stupně základní školy, více variant zadání jednoho cvičení, obsah stránek v českém jazyce a dostupnost zdarma. U každé aplikace je popsáno její uživatelské rozhraní, nabídka funkcí a technické zpracování.

Druhá část práce se zabývá analýzou a návrhem systému prostřednictvím objektově orientovaného jazyka UML. Při návrhu systému bylo využito následujících diagramů jazyka UML – diagram případu užití a diagram tříd, jenž jsou dále doplněny specifikací systémových požadavků a diagramy aktivit. Popis vývoje aplikace je navíc doplněn o drátové modely jednotlivých webových stránek a popis databázové struktury systému.

Následující část práce popisuje bezpečnostní rizika webové aplikace, se kterými bylo zapotřebí se vypořádat, a stručně představuje technologie, jež byly použity při tvorbě a testování funkčnosti aplikace. V samotném závěru práce je popsán průběh ověřování webové aplikace v hodinách matematiky na základních školách, respektive víceletých gymnáziích.

Toto téma jsem si zvolil z důvodu osobního zájmu o matematiku, a také z důvodu možnosti využít moderní webové technologie k vytvoření aplikace, jenž by mohla být užitečná v rámci výuky či samostatného procvičování.

1.1 Cíle práce

Cílem bakalářské práce bylo vyhledání, analyzování a porovnání webových stránek umožňujících procvičování příkladů nebo testování žáků z matematiky na úrovni 2. stupně základní školy.

Dále navrhnout a implementovat webovou aplikaci sloužící k procvičování příkladů a testování žáků z vybraných partií matematiky na úrovni základní školy, která bude prospěšná i učitelům a naplnit ji samotným obsahem. Webové stránky rozlišují mezi rolemi žáka, učitele a administrátora. Učitelé mohou vytvářet skupiny, testy a vkládat vlastní příklady do systému. Žáci si budou moci procvičovat jednotlivé příklady výběrem z nabízených kategorií, vstupovat do různých skupin, v rámci těchto skupin absolvovat testy a prohlížet si své výsledky z procvičování a testování.

Mezi cíle též patřilo otestování aplikace na několika základních školách, respektive víceletých gymnáziích.

1.2 Metody práce

Nejprve byl proveden průzkum zaměřený na webové stránky umožňující testování žáků z matematiky. Z množiny nalezených stránek byly vybrány takové, které splňovaly stanovená kritéria, mezi něž patřila např. dostupnost zdarma či zaměření na 2. stupeň základní školy, a tyto stránky byly podrobněji analyzovány a porovnány.

Před samotnou implementací webové aplikace byla provedena potřebná analýza, návrh a otestování jednotlivých částí. K tomu byly využity nástroje jazyka UML a tyto informace byly zobrazeny pomocí jednotlivých diagramů. Součástí návrhu je i databázová struktura aplikace a prvotní zobrazení grafického rozhraní jednotlivých stránek prostřednictvím wireframe. Po této části byly zvoleny vhodné technologie a naprogramována daná aplikace.

Než byla samotná aplikace odzkoušena v praxi, byly potřebné kategorie naplněny dostatečným množstvím příkladů a řádně odcitovány. Na závěr byla webová aplikace ověřena v hodinách matematiky na několika základních školách a víceletých gymnáziích a získána zpětná vazba od učitelů, kteří také zprostředkovali dojmy svých žáků.

2 Analýza webových stránek

Tato část práce je zaměřena na analýzu a porovnání českých webů, které se věnují online procvičování a testování znalostí z matematiky na úrovni 2. stupně základní školy.

Níže je analyzováno pět webových stránek, jež se zobrazily¹ na předních pozicích ve vyhledávači Google po zadání kombinací klíčových slov matematika, procvičování, testy, příklady, online. Pro vybrání webů k analýze byly stanoveny minimální požadavky, které měla webová stránka splňovat:

- zaměření na 2. stupeň základní školy,
- více variant zadání jednoho cvičení,
- obsah stránek v českém jazyce,
- dostupnost zdarma.

Bohužel těmito kritérii prošlo pouze několik webových portálů. Ve většině případů nesplnily druhý požadavek. Testy nebo cvičení obsahovaly pouze jednu variantu zadání, se kterou si žák své znalosti moc nezdokonalí. Pro představu můžeme uvést příklad webových stránek *O škole*². Některé podobné weby byly vytvořeny přímo učiteli a jsou k dispozici žákům na domovských stránkách školy. Jelikož není potřeba se přihlašovat, mohou toto procvičování využít i žáci mimo danou školu. Pro představu si uvedeme webové stránky základní školy Dobřichovice³.

Vyhledávač zobrazoval ve výsledcích vyhledávání velmi často weby zaměřené na jinou věkovou skupinu žáků. Převládaly webové stránky se zaměřením na žáky prvního stupně základní školy. V tomto případě si můžeme uvést příklad stránek *Matematika online: Výuka a doučování zdarma*⁴.

Často se mezi výsledky zobrazovaly také webové stránky, které se nevěnovaly online procvičování a testování matematiky, ale obsahovaly výklad látky a ukázkově

¹Březen 2014.

²http://www.oskole.sk/index.php?id_cat=34

³<http://www.zsdoobrichovice.cz/ukoly/matika/testy/testy.php?go=blanke.cz/ukoly/matika/testy/testy.php?go=blank>

⁴<http://www.celysvet.cz/matematika>

řešené příklady nebo pouze ukázkově řešené příklady. I zde si pro názornost uvedeme například webové stránky *E-matematika.cz*⁵.

U každého výukového webu si uvedeme nabídku funkcí, možnosti uživatele, technické a grafické zpracování. Pro hodnocení byla zvolena následující kritéria:

- přehlednost webu,
- množství reklam,
- poslední aktualizace obsahu webu,
- množství variant příkladů k jednomu tématu,
- zpětná vazba,
- statistika výsledků – zobrazení počtu správných, resp. chybných odpovědí ve cvičení,
- možnost přihlášení a vytvoření svého účtu,
- možnost vytvářet skupiny uživatelů,
- grafické a technické zpracování – dostatečné rozlišení obrázků,
- validita zdrojového kódu,
- přístupnost webu.

Při kontrole validnosti a přístupnosti stránek byly použity kromě osobní kontroly také webové stránky *SEO Servis*⁶ zabývající se optimalizací webu a *W3C validátor*⁷. Pro testování funkcí analyzovaných webů bylo dle potřeby vytvořeno několik fiktivních uživatelských účtů. Datum poslední aktualizace webových stránek je uváděno z doby 6. – 11. 3. 2014.

⁵<http://www.e-matematika.cz/zakladni-skoly/>

⁶<http://seo-servis.cz/source-zdrojovy-kod/>

⁷<http://validator.w3.org/>

2.1 Online cvičení

Provozovatel/správce/autor: VEREA, s.r.o.

Zaměření: 1. a 2. stupeň ZŠ

Poslední aktualizace: březen 2014

URL adresa: http://www.onlinecviceni.cz/exc/list_sel_topics.php

Webová stránka www.onlinecviceni.cz je určena pro žáky prvního a druhého stupně základní školy. Kromě matematiky nabízí také procvičování českého jazyka.

Žáci mají na výběr z příkladů na procvičení, které jsou členěny nejprve podle ročníku a poté podle tématu. Tato nabídka je dostupná jak přihlášeným, tak nepřihlášeným uživatelům. U některých témat je možné vybírat z více variant testů. Řešiteli se při počítání ukazuje uběhlý čas. Pokud by tato časomíra mohla žáka rozptylovat nebo stresovat, lze ji při vyplňování testu skrýt. Po ukončení počítání uživatel vidí všechny příklady se svými odpověďmi, u kterých je napsáno, zda byl příklad správně nebo chybně vyřešen. Ve druhém případě jsou zobrazena i správná řešení. Dále uživatel vidí svůj celkový čas, za který cvičení vyřešil, počet správně a chybně řešených příkladů a procentuální úspěšnost.

Stránky umožňují vytvoření uživatelských skupin. Zakladatel skupiny se stává jejím administrátorem a rozhoduje o tom, kdo se stane jejím členem. Těmto uživatelům může následně zadávat a kontrolovat privátní úkoly, resp. testy v rámci této skupiny podobné veřejným soutěžím.

Další sekce – úkoly, soutěže, skupiny a výsledky – jsou dostupné pouze přihlášenému uživateli. Všichni přihlášení uživatelé mají stejná práva a vidí stejnou nabídku funkcí. Nerozlišuje se tedy, zda jste v roli žáka, učitele nebo rodiče.

Pod volbou *Skupiny* najdeme přehled zde vytvořených skupin a můžeme požádat o přiřazení do některé z nich. Dále jsou tu sekce *Založit skupinu*, *Moje skupiny* a *Zadat test*. Testy můžeme zadávat pouze v rámci našich skupin. V sekci *Úkoly* je zobrazen seznam testů vytvořených v rámci skupin, do kterých se uživatel přihlásil. V této části si můžeme zároveň prohlédnout aktuálně běžící soutěže a zapojit se do nich.

Uživatel se může zúčastnit vypsanych veřejných soutěží, která jsou na různá témata a probíhají obvykle několik dní. Soutěžícímu se přičítají body za umístění

na prvních třech místech a za bezchybné vyřešení. Během i po ukončení soutěže je k dispozici žebříček s umístěním v jednotlivých soutěžích.



Obrázek 1: Úvodní strana webových stránek www.onlinecviceni.cz.

V části *Výsledky* si může uživatel prohlédnout svou osobní statistiku ze cvičení, z úkolů a ze soutěží. U výsledků soutěže si může zobrazit i celkové pořadí všech účastníků v soutěžích. Na úvodní stránce jsme ihned informováni o nejúspěšnějších řešitelích v rámci soutěží a v počtu vyřešených různých cvičení bez chyb.

Zpracování příkladů je jednoduché, srozumitelné a výstižné. Obtížnost příkladů je převážně na základní úrovni a obdobná u všech deseti zobrazených příkladů jednoho cvičení.

Na první pohled stránka zaujme svým výrazným pozadím, jež může na někoho působit až příliš dětinsky. Grafika se mění podle toho, v jaké části se nacházíme – úvodní strana, matematika, český jazyk.

Webové stránky je možné přepnout do anglického jazyka. Překlad je v současné době jen částečný. Titulní strana je přeložena, ale například témata lekcí v rozcestníku už ne.

Jako velké plus bych hodnotil absenci různých reklam. Stránka působí přehledně a není třeba přemýšlet nad tím, zda dané okno je ještě součástí webu nebo jen další reklamní upoutávka. Dále velkou předností těchto stránek je nabídka různých soutěží, úkolů a členění do skupin, o kterých jsme se již zmínili.

Naopak jako poměrně velký nedostatek bych označil to, že nevíme, kdo je autorem nebo provozovatelem webových stránek. Nikde na úvodní stránce není k nalezení odkaz *O nás* či uveden autor. K této informaci se dostaneme netradiční cestou – přes odkaz na novinky. Na této stránce je kromě očekávaných novinek i kontakt na brněnskou společnost VEREA, s.r.o. Dále nikde na úvodní stránce není napsáno úvodní slovo, které by uvádělo, co je cílem a předmětem těchto stránek a co od nich uživatelé mohou očekávat. Tuto informaci opět nalezneme až po kliknutí na novinky, kde se můžeme seznámit také se způsobem registrace, jak soutěžit, jak vytvářet skupiny a jak získat emailové zprávy s výsledky a bonusová cvičení.

Emailové zprávy, ve kterých se uživatel dozví své denní souhrnné výsledky z absolvovaných cvičení, patří již mezi placené služby. Tento doplněk je možné zakoupit za sto korun za jednoho uživatele. Také bonusová cvičení jsou zpoplatněna a nabízejí cvičení navíc nebo anglicky namluvená cvičení. Balíček bonusových cvičení stojí sto korun.

Další výtku bych uvedl k ovládní webu (pohybu po stránkách). Po zvolení některého z témat není zřejmé, jak se dostat zpátky na předchozí výběr. Jasně viditelné a označené tlačítko je zde pouze Domů, jež nás odkáže na úvodní stránku webu. Existuje tu sice tlačítko pro zobrazení témat zvoleného ročníku, to ale není dostatečně označeno a snadno se přehlédne. Kvůli lepší orientaci by bylo vhodné, kdyby stránky obsahovaly drobečkovou nápovědu.

Z hlediska kvality zdrojového kódu a přístupnosti webu jsou na tom webové stránky Online cvičení velmi špatně. Z nejvýznamnějších chyb bych uvedl:

- absence mapy webu,
- absence některých alternativních textů u netextových prvků,
- absence atributu title u odkazů,
- definice CSS stylů a JavaScriptových funkcí nejsou v externím souboru,

- část textu je pouze ve formě obrázku,
- nízká kvalita obrázků – viditelná „zubatost“ čar,
- absence standardních klávesových zkratk,
- zdrojový kód není validní.

2.2 Matematika hrou

Provozovatel/správce/autor: Jiří Dobrý

Spolupracovník autora: Petra Janošková

Zaměření: 1. a 2. stupeň ZŠ

Poslední aktualizace: září 2013

URL adresa: <http://matematika.hrou.cz/>

Webové stránky Matematika hrou jsou určeny pro žáky základních škol 1. až 9. ročníku. Již název domény napovídá, že se stránky zaměřují pouze na matematiku.

Hru můžeme zvolit buď podle tématu, nebo dle třídy, kde každý ročník obsahuje několik témat. Autor uvádí, že příklady jsou řazeny podle doporučení osnov matematiky, ale že může docházet k různým odchylkám u jednotlivých škol. Při každém počítání se nám zobrazí vždy deset příkladů. Před zahájením každé hry jsme informováni, za jaký čas musíme hru splnit, abychom dosáhli na známku výborně. Bohužel během počítání nejsme informováni o základních ukazatelích (čas, číslo příkladu).

Kontrola řešení jednotlivých příkladů probíhá automaticky po každém zadání výsledku, aniž by bylo potřeba tuto kontrolu potvrdit. Pokud zadáme správný výsledek, načte se nám automaticky nový příklad. Pokud zadáme špatný výsledek, tak se nic nestane, nezobrazí se žádné oznámení o chybě, pouze se nemůžeme posunout na nový příklad. Nezobrazení hlášení o chybě bych označil jako nedostatek těchto stránek. Žák si nemusí uvědomit, že udělal chybu, a proto se mu neukáže další příklad. Může si myslet, že se například pouze nenačítá stránka či stránky nefungují.

U některých témat je k dispozici více variant příkladů, které se náhodně generují. Obtížnost příkladů bychom mohli označit jako základní, což může souviset s tím,

Nejhranější matematické hry

1. stupeň ZŠ

1. [Malá násobilka](#) - základ matematiky pro všechny.
2. [Procvičování násobilky](#) - vyber násobilku, procvičuj.
3. [Pexeso na sčítání](#) - odkrojěj obrázek sčítáním malých čísel.
4. [Sčítání jednociferných čísel](#) bez přechodu přes 10.
5. [Sčítání a odčítání do 20](#) - sčítání a odčítání se střídá.

2. stupeň ZŠ

1. [Malá násobilka](#) - základ matematiky, bez debat!
2. [Umocňování na druhou, snadné](#) - výsledek je do 50.
3. [Převod na římské číslice](#) - včetně taháku.
4. [Krácení zlomků](#) - v pexesu vyber rozšířený a zkrácený zlomek.
5. [Doplňování desetinných čísel](#) - základy desetinných čísel.

Nejnovější hry

[Absolutní hodnota](#)

9. září 2013

Vypočít absolutní hodnotu příkladu na sčítání a odčítání.

[Násobilka s nápovědou](#)

4. dubna 2013

Násobilka do 22 s nápovědou.

[Obvod rovinných útvarů](#)

16. prosince 2012

Vhodné jako základy obvodu pro 3. a 4. třídu.

[Obsah rovinných útvarů](#)

12. prosince 2012

Vhodné jako základy obsahu pro 3. a 4. třídu.

[Doplň desetinná čísla](#)

9. prosince 2012

Graficky sčítáme desetinná čísla. Vynikající jako základ desetinných čísel.

[Dělení desetinným číslem](#)

7. prosince 2012

Výsledek je celé číslo, např. $3 : 1,5 = 2$.

Příklady podle tříd

[Sčítání a odčítání](#)

[Násobení a dělení](#)

[Mocniny a odmocniny](#)

[Zlomky](#)

[Desetinná čísla](#)

[Dělitelé a násobky](#)

[Římské číslice](#)

[Obvod a obsah](#)

[Geometrie](#)

[O webu a kontakt](#)

Obrázek 2: Úvodní strana webových stránek www.matematika.hrou.cz.

že autor těchto stránek není ani učitel, ani matematik. Cílem těchto stránek je pravděpodobně spíše zaujmout žáky pro matematiku než procvičit všechny oblasti probírané matematiky na základní škole.

Stránky obsahují minimum reklam, úvodní strana dokonce žádné, což přispívá k přehlednosti webu.

Stránky nám umožňují zaregistrovat se jako žák či jako učitel. Učitel může zakládat třídy a posílat vzkazy žákům, jenž se zobrazí na jejich nástěnce po přihlášení do portálu. Učiteli se zobrazují aktivity všech žáků, kteří se zařadili do jeho tříd. Žák si může vybrat, do jaké třídy se zařadí. Jako nedostatek bych zde uvedl to, že učitel nemá žádná práva pro omezení vstupu nějakého žáka do své třídy. Dále žák vidí svoje nejlépe hodnocená cvičení, ale bohužel pouze název tématu, datum a čas, za který cvičení splnil. Nejsou zde uvedeny žádné informace o počtu chyb.

Z hlediska kvality zdrojového kódu a přístupnosti webu jsou na tom webové stránky Matematika hrou poměrně dobře. Text a nadpisy jsou správně strukturo-

vané, netextové prvky obsahují alternativní text, titulek stránek je vyplněn. Přesto i zde se vyskytují některé zásadní nedostatky ve zdrojovém kódu a přístupnosti, uvedeme si několik z nich:

- absence mapy webu,
- absence klíčových slov,
- definice CSS stylů a JavaScriptových funkcí nejsou v externím souboru,
- absence atributu title u odkazů,
- absence standardních klávesových zkratk,
- zdrojový kód není validní.

2.3 Pocitejme.cz

Provozovatel/správce/autor: Michal N.

Zaměření: 1. a 2. stupeň ZŠ, SŠ

Poslední aktualizace: 2014

URL adresa: <http://www.pocitejme.cz/>

Webové stránky Pocitejme.cz se zaměřují pouze na příklady z matematiky a jsou určeny pro žáky základních i středních škol.

Příklady jsou členěny nejprve podle věku žáka – první stupeň, druhý stupeň, střední škola a pokročilí. Sekci *Pokročilí* bychom mohli chápat jako bonusové úlohy – stejné typy příkladů jako v ostatních částech, jen s větší obtížností. V každé úrovni si můžeme vybrat z několika témat. Existuje více variant testů, příklady jsou náhodně generovány a před spuštěním každého cvičení si žák může zvolit počet příkladů v testu. Po vyhodnocení testu se nám zobrazí výsledná známka, náhodně generovaný obrázek a seznam správných a špatných výsledků. Také je nám nabídnuta možnost napsat si test znovu ze stejného tématu, ale s jiným zadáním příkladů.

Užitečnou funkcí těchto stránek je virtuální kalkulačka, kterou si můžeme při počítání otevřít a pomoci si při výpočtech.

Jako nevýhodu těchto stránek bych uvedl to, že zde není možnost vytvoření uživatelského účtu a tím uchovávání výsledků z absolvovaných cvičení. Za to tvůrce nabízí možnost přepnout tyto stránky do devíti jazyků, například slovenština, polština, ruština a angličtina, a stále hledá nové kolegy pro překlad stránek do dalších jazyků, aby se používání stránek mohlo rozšířit do co největšího počtu zemí.



Obrázek 3: Úvodní strana webových stránek www.pocitejme.cz.

Web působí přehledně a jednotným dojmem, i když obsahuje několik reklam. Zadání příkladů je sice velmi stručné, ale i tak je na první pohled zřejmé, co je naším úkolem.

Z hlediska kvality zdrojového kódu jsou na tom webové stránky *Pocitejme.cz* velmi dobře. Nadpisy jsou správně strukturované, netextové prvky obsahují alternativní text, je vyplněn titulek stránek. Přesto i zde se vyskytuje několik nedostatků ve zdrojovém kódu a v přístupnosti webu, uvedeme si několik z nich:

- absence mapy webu,
- definice JavaScriptových funkcí nejsou v externím souboru,
- absence standardních klávesových zkratk,
- text není strukturovaný do odstavců,
- zdrojový kód není validní.

2.4 Procvičuj.cz

Administrátor: Petr Ehler

Matematika: Eva Ehlerová

Zaměření: 1. a 2. stupeň ZŠ (není přesně specifikováno)

Poslední aktualizace: listopad 2011

URL adresa: <http://nove.procvicuj.cz/www/>

Portál *Procvičuj.cz* se kromě matematiky zaměřuje i na procvičení chemie a češtiny na úrovni základní školy. Stupeň nebo ročníky základní školy nejsou přesně specifikovány.

Příklady jsou tedy členěny pouze podle tématu, která obsahují několik podkategorií. Existuje více variant testů a příklady jsou náhodně generovány. U každé kategorie můžeme vždy určit počet příkladů v testu, který se pohybuje mezi 10 – 50 příklady. Dále můžeme navolit ještě několik parametrů testu, které se liší podle tématu. Například u kategorie *Opakování desetinných čísel* můžeme navíc nastavit počet členů v příkladu, zda se budou používat pouze kladná čísla a horní hranici používaných čísel. Bohužel nabídka témat není nijak rozsáhlá a svou náročností jsou určena spíše mladším žákům 5. – 7. ročníků.

Po spuštění testu je nám nabídnuta volba *Vytisknout zadání*, která nám aktuálně vygenerované zadání příkladů zobrazí na další webové stránce jen jako obyčejný text bez použití formulářů. Pokud nemáme vyplněny všechny odpovědi a i přesto chceme test vyhodnotit, jsme dotázáni, zda tak skutečně chceme učinit a zvýrazní se prázdné položky v testu. Po vyhodnocení testu se nám zobrazí naše odpovědi, správné odpovědi, počet správných a chybných odpovědí a známka.



Obrázek 4: Úvodní strana webových stránek www.procvicuj.cz

Příklady jsou dostupné zdarma, bez nutnosti přihlášení. Při registraci kromě běžných údajů vyplňujeme, zda jsme žák nebo učitel a ID školy, ke které patříme. Příslušnou školu vybíráme ze seznamu již zaregistrovaných škol nebo současně při vlastní registraci novou školu zaevidujeme zadáním ID školy, adresy, města a URL adresy webu školy. Pokud se zaregistrujeme jako žák, můžeme si prohlížet výsledky vyplněných testů. Dále si můžeme prohlédnout souhrnnou statistiku testů, ve které se dozvíme, jaké a kolik známek jsme za jednotlivé testy získali, známkový průměr, kolik testů jsme vyřešili z matematiky, chemie a češtiny. Získané známky máme zobrazeny také pomocí výšečového grafu.

V roli učitele máme k dispozici stejnou nabídku funkcí jako žák, ale navíc vidíme všechny třídy s žáky, které jsou na portále vedeny pod stejnou školou jako učitel. Vyučující si může prohlédnout souhrnnou statistiku všech testů žáků i odpovědi

u jejich jednotlivých testů. Učitel má také možnost smazat záznamy o žákových vypočítaných testech.

Jako nevýhodu tohoto systému bych uvedl to, že nikdo neschvaluje, kdo se přidal do jaké školy, a to jak z pohledu žáka, tak z pohledu učitele. Není potřeba povolení administrátora či nějaké jiné odpovědné osoby za danou školu.

Webové stránky působí přehledně a rychle se v nich orientuje, i když obsahují v dolní části reklamy. Na úvodní stránce se v krátkém povídání ihned seznámíme s obsahem webu a s nabízenými funkcemi.

Z hlediska kvality zdrojového kódu a přístupnosti webu jsou webové stránky *Procvičuj.cz* na vysoké úrovni. Stránky mají vyplněný titulek, strukturovaný text, jsou uvedena klíčová slova, mapa webu. Přesto i zde se vyskytují některé nedostatky a několik z nich si nyní uvedeme:

- absence atributu title u odkazů,
- stránka obsahuje dva nadpisy první úrovně,
- definice CSS stylů a JavaScriptových funkcí nejsou v externím souboru,
- absence standardních klávesových zkratk,
- zdrojový kód není validní.

2.5 Testy a kvízy on-line

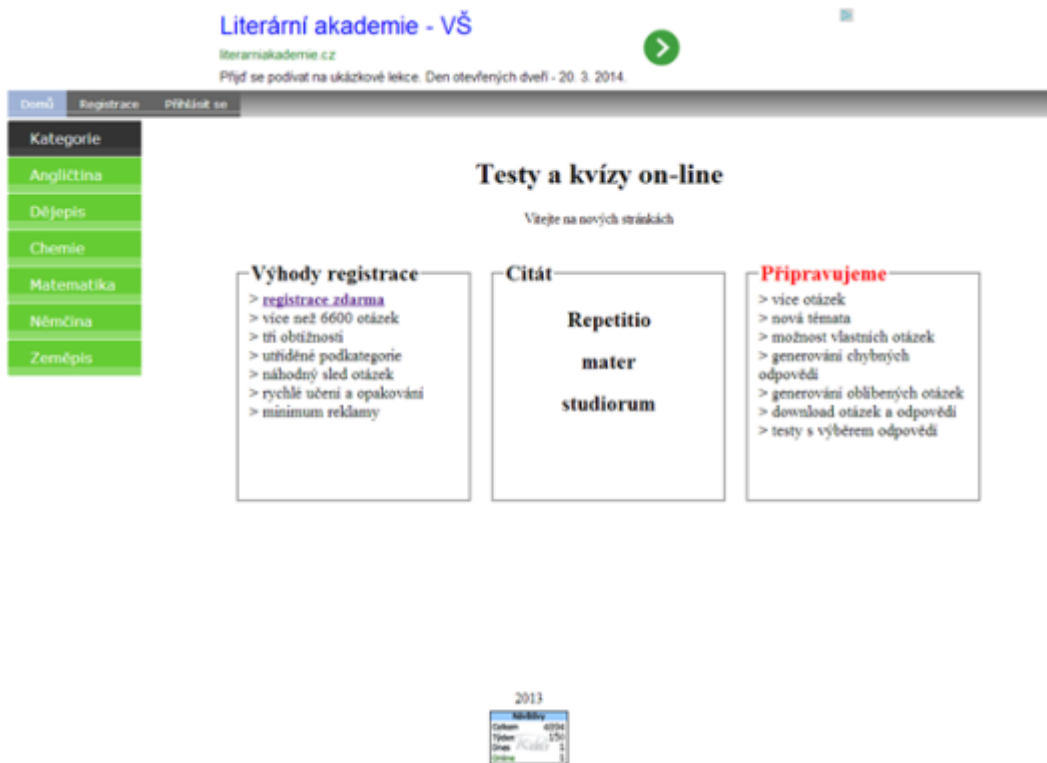
Provozovatel/správce/autor: nevedeno

Zaměření: 1. a 2. stupeň ZŠ (není přesně specifikováno)

Poslední aktualizace: 2013

URL adresa: <http://1quiz.jirpa.cz/>

Webové stránky *Testy a kvízy on-line* obsahují testy kromě matematiky i z jiných předmětů, například angličtina, dějepis, chemie. Bohužel nikde není uvedeno, pro jaký stupeň školy jsou stránky určeny. Ze zaměření nabízených témat lze odhadnout, že stránky mají sloužit žákům základní školy. U kategorie matematika bychom mohli říci, že jsou určeny pro žáky prvního stupně a pro žáky 5. – 7. ročníku druhého stupně.



Obrázek 5: Úvodní strana webové stránky www.1quiz.jirpa.cz/index.php.

Testy jsou rozděleny podle témat a ty se dále dělí na jednotlivá podtémata. Těchto podkategorií není v současné době mnoho, ale autoři podle informací na titulní stránce chystají rozšíření.

Stránky jsou zdarma a lze je používat i bez registrace. Nepřihlášený uživatel má ale velmi omezené možnosti – vidí pouze jednu variantu testu k danému tématu. Každý test obsahuje deset příkladů. Zadání je velmi stručné, ale postačující k vyřešení příkladů. Po vyhodnocení testu se nám zobrazí správné odpovědi, počet našich správných odpovědí a procentuální úspěšnost.

Přihlášený uživatel má k dispozici více variant testů k jednomu tématu, navíc se mu zpřístupní další příklady na procvičení. V této nové části je na výběr ze stejných témat jako u testů, ale příklady se zobrazují po jednom a můžeme zvolit obtížnost následujícího příkladu (1 – 3). Příklady jsou obdobné jako máme k dispozici v testu. Dle číslování jednotlivých příkladů, které se pohybují řádově ve stovkách, můžeme říci, že velikost databáze zadání příkladů je poměrně rozsáhlá. Zadání se generuje v náhodném pořadí. Po vyhodnocení každého příkladu vidíme, jaká měla

být správná odpověď a zda jsme odpověděli správně. Současně při počítání neustále vidíme průběžnou statistiku - procentuální úspěšnost, počet dobrých a špatných odpovědí a celkový počet zatím řešených příkladů. Do této statistiky se započítávají i řešené příklady z testů, což nemusí být na první pohled zřejmé, jelikož se tato statistika u testů nezobrazuje. Toto bych označil jako nedostatek stránek. Vhodnější by bylo pro tyto dvě části zobrazovat oddělenou statistiku.

V rámci svého účtu si přihlášený uživatel může prohlédnout svoji statistiku. V té je uveden celkový počet zodpovězených příkladů, kolik z nich bylo správně, kolik chybně, kolik z nich nebylo vyplněno a procentuální úspěšnost. Tato statistika je vedena dohromady za příklady z procvičování a z testů. Nevýhodou je, že si uživatel nemůže prohlédnout svou úspěšnost v jednotlivých testech.

Při registraci se nerozlišuje, zda uživatel je žák či učitel. Všichni uživatelé mají stejná práva. Portál neumožňuje, aby uživatelé vytvářeli různé skupiny.

Další rozdíl mezi přihlášeným a nepřihlášeným uživatelem je v množství reklam. Přihlášenému uživateli se žádné nezobrazují. I tak stránky obsahují jen minimum reklamy, jež nezasahují do zadání příkladů.

Web působí poměrně skromným dojmem, neobsahuje téměř žádné grafické prvky. Na titulní straně jsme informováni o výhodách registrace a jaká vylepšení stránek jsou v plánu, nikde však nevidíme poznámku o autorovi, tvůrci stránek, ani pár slov o tomto projektu.

Z hlediska kvality zdrojového kódu a přístupnosti webu jsou webové stránky *Testy a kvízy on-line* na poměrně dobré úrovni. Stránky mají vyplněný titulek, nadpis první úrovně a správně strukturované další úrovně, jsou uvedena klíčová slova. Přesto i zde se vyskytují některé zásadní nedostatky, které by bylo radno odstranit. Nyní si uvedeme několik z nich:

- absence mapy webu,
- absence atributu title u odkazů,
- text není strukturovan do odstavců,
- některé definice CSS stylů nejsou v externím souboru,

- absence standardních klávesových zkratk,
- zdrojový kód není validní.

2.6 Shrnutí

Z výše uvedeného přehledu je zřejmé, že českých portálů, věnovaných online testování a procvičování znalostí z matematiky, zatím příliš není. Stále převládají webové stránky, které se věnují pouze výkladu matematiky, řešením ukázkových příkladů nebo kombinaci těchto dvou zaměření.

Pozitivním zjištěním ale je, že některé analyzované stránky jsou ve velkém počtu prakticky využívány žáky a učiteli různých škol, což lze usuzovat z počtu zaregistrovaných škol na některých stránkách, z počtu uživatelů účastnících se soutěží apod.

Některé výše analyzované webové stránky představují pěknou ukázkou toho, jakým směrem by se weby, věnující se online procvičování, mohly ubírat. Přesto ani jeden portál neobsahoval všechny funkce, jež by dle mého názoru měly obsahovat. Mezi ně můžeme zahrnout vytvoření uživatelského účtu, rozlišování rolí přihlášených uživatelů, vytváření vlastních skupin, soutěže, možnost vkládat vlastní příklady, zobrazení statistiky úspěšnosti. Nejblíže ke splnění všech těchto požadavků byly weby *Online cvičení* a *Matematika hrou*. Přehled funkcí všech analyzovaných webových stránek je zobrazen v příloze v tabulce 1.

Při návrhu webových stránek *Vypoctito.cz* jsem se proto zaměřil na všechny výše zmíněné funkce a na kvalitu zpracování zdrojového kódu, který je u analyzovaných portálů velmi často odbyt, a na přístupnost webu, jež je u všech zcela opomíjena. Dále jsem se zaměřil na větší rozsah nabízených témat, jenž u některých porovnávaných webů byl poměrně nedostačující.

3 Analýza a návrh systému

V této kapitole se budeme zabývat tvorbou analytického modelu, který nám popisuje, co vše by měl systém umět, avšak už se nezabývá otázkou, jak toho dosáhne. Do této fáze náleží stanovení požadavků na systém, případy užití a diagramy aktivit. Zbylá část kapitoly se věnuje návrhu systému, který se již zabývá otázkou, jakým způsobem dosáhnout požadované funkčnosti. Cílem je vytvořit model systému, který již následně bude možné implementovat. Zde se budeme zabývat návrhovým modelem tříd, databázovým návrhem a drátovým modelem aplikace.

3.1 Specifikace systémových požadavků

V průběhu specifikace požadavků sestavujeme prvotní požadavky na realizaci systému. Pojmeme požadavky chápeme vše, co by systém měl obsahovat a umožňovat uživateli. V této chvíli se tedy nezaměřujeme na to, jakým způsobem této funkčnosti chceme dosáhnout, ale pouze na to, jaké funkce od systému požadujeme. Rozlišujeme dva typy požadavků:

- funkční požadavky, tj. jaké funkce bude systém obsahovat,
- nefunkční požadavky, tj. vlastnosti, omezení a nároky na systém apod. [3]

3.1.1 Popis systému

Webové stránky *VypočtiTo* budou zaměřeny na online procvičování a testování žáků základní školy. Z obsahu předmětu matematika pro 2. stupeň základní školy jsou vybrána některá témata, jež jsou rozdělena do 15 základních okruhů a 3 obtížností. Do těchto kategorií jsou zařazeny jednotlivé příklady pro procvičování, vytváření testů a soutěží. Systém mohou využívat uživatelé i bez přihlášení, mají však omezené možnosti. Nepřihlášený uživatel může pouze procvičovat a vybírá si z daných 15 témat.

Uživatel se může zaregistrovat v roli učitele nebo v roli žáka. Žák může spravovat svůj účet, procvičovat příklady, zúčastnit se vypsaných soutěží, zapsat se do skupin a odepsat se z ní, vyplnit testy (zadané v rámci žákových skupin) či si prohlížet

svou statistiku úspěšnosti. Učitel může měnit své osobní údaje, spravovat skupiny (vytvářet, rušit), sestavovat a spravovat testy (editovat, smazat) a rozšiřovat databázi systému o vlastní příklady.

Správu systému zajišťuje administrátor, který spravuje příklady (vkládá, edituje, ruší a schvaluje příklady vložené učitelem), spravuje uživatelské účty (může smazat uživatele, skupinu nebo test) a vytváří soutěže (včetně editace).

Pro tento systém definujeme následující role:

- nepřihlášený uživatel,
- žák,
- učitel,
- administrátor.

3.1.2 Funkční požadavky na systém

Funkční požadavky představují funkci systému, zaměřujeme se na to, co systém dělá. Pro náš systém si definujeme následující typy požadavků:

1. Registrace, přihlášení a evidence uživatelů.

Systém bude evidovat žáky a učitele, kteří se do něho zaregistrují. Každý uživatel bude mít pouze jednu roli, se kterou jsou spojena určitá práva k používání systému. Každý uživatel bude mít ve svém účtu uvedeno uživatelské jméno, heslo, jméno, pohlaví, školu a email.

2. Vkládání nových příkladů.

Nové příklady bude moci zadávat do systému pouze administrátor nebo učitel. U každého příkladu bude uvedeno zadání, řešení, obtížnost příkladu, autor, zdroj (odkud byl příklad převzat) a okruh, do kterého tematicky zapadá. Podle potřeby lze přiložit k zadání obrázek. V případě učitele lze ještě nastavit, zda nový příklad bude veřejný (dostupný všem) či nikoli (tj. bude ho využívat pouze daný učitel). Pokud příklad vkládá učitel a nastaví jej jako veřejný, musí čekat, zda ho schválí administrátor, a teprve poté se zobrazí i ve veřejné části systému.

3. Procvičování.

Procvičovat příklady může nepřihlášený uživatel a žák. Uživatel nejprve vybere téma, které chce procvičit, následně zvolí počet a obtížnost příkladů. Po vyhodnocení příkladů se uživateli zobrazí počet správných, chybných a nezodpovězených odpovědí a jeho procentuální úspěšnost. Obě role mají v tomto případě stejné podmínky v nastavení funkcí, liší se pouze v tom, že přihlášený žák si může zobrazené statistiky prohlédnout i později ve svém profilu.

4. Vytvoření soutěže.

Soutěže bude do systému přidávat pouze administrátor. U zadání každé soutěže bude uveden datum spuštění a ukončení soutěže a název (označení soutěže).

5. Řešení soutěže.

Řešitelem soutěže může být pouze uživatel v roli zaregistrovaného žáka. Žák má u každé soutěže informace o datu spuštění a ukončení soutěže, o jejím názvu, a pokud je umožněno, může vidět i průběžné, respektive závěrečné výsledky. Ve výsledcích je uvedeno pořadí žáků od prvního místa, jména řešitelů a jejich procentuální úspěšnost.

6. Vytvoření skupiny.

Skupinu vytváří učitel a při jejím zakládání uvádí název skupiny a nechá automaticky vygenerovat ID skupiny, které slouží k zajištění jednoznačné identifikace dané skupiny. Administrátor má právo libovolnou skupinu smazat.

7. Vstup a přehled skupin.

Žáci se mohou začleňovat do skupin (může jich být více). Danou skupinu vyhledají pomocí jejího ID. O začlenění do skupiny musí žáci zažádat a čekat na potvrzení od učitele. V přehledu žákových skupin je u každé z nich uveden datum vzniku, název, zakladatel a informace, zda má žák již potvrzenou registraci či nikoli. Žák může kdykoli libovolnou skupinu opustit nebo může být vyloučen ze strany učitele.

8. Vytvoření testu.

Testy v systému vytváří učitelé pro žáky v rámci svých skupin. Učitel zadává

název (označení) testu, skupinu, resp. skupiny, pro který je test určen, téma, zda bude možné test opakovat, datum spuštění a ukončení testu a následně vybírá jednotlivé příklady. Administrátor má právo libovolný test smazat.

9. Prohlížení a editace testu.

Učitel si vytvořené testy může prohlédnout, smazat je nebo editovat. U každého testu si může nechat zobrazit výsledky žáků, kteří ho již absolvovali. Při editaci může učitel změnit všechny parametry popsané v bodě 8 kromě výběru zaměření příkladů – tématu. Pokud test již absolvoval nějaký žák, nelze pozměnit žádné parametry testu.

10. Řešení testu.

Test může řešit žák a to pouze pokud je přihlášen a schválen ve skupině, do které byl daný test zařazen.

11. Zobrazení statistik.

Žák si může ve svém profilu nechat zobrazit statistiku úspěšnosti z procvičování, testů, nebo soutěží. U každého cvičení, resp. testu je uveden název tématu, resp. název testu, počet příkladů, počet správných, chybných a nezodpovězených odpovědí a procentuální úspěšnost. U testu se navíc zobrazuje datum, kdy ho žák absolvoval. Ve výsledcích soutěže je uveden název soutěže a sestupně seřazení řešitelé soutěže podle jejich úspěšnosti.

3.1.3 Nefunkční požadavky na systém

Nefunkční požadavky charakterizují různá omezení a nastavení systému. Pro systém online procvičování matematiky *VypočtiTo* jsou následující:

- dostupnost přes webový prohlížeč,
- implementace pomocí PHP a MySQL,
- postavení na validních a přístupných webových stránkách,
- volně dostupný,
- přehledný a jednoduchý na ovládání.

3.2 Případy užití

Případy užití (z anglického originálu Use Cases) se zabývají stanovením funkčních požadavků kladených na informační systém. Případy užití musí pokrýt celkový rozsah možností a nástrojů informačního systému. V této fázi tvorby se nezabýváme jakým způsobem budou jednotlivé funkce realizovány. Zajímá nás pouze, co bude systém umožňovat.

Tyto vlastnosti systému graficky znázorňujeme do diagramu případu užití (Use Case Diagram), který zobrazuje systém z pohledu uživatele. Při tvorbě diagramu je potřeba určit několik prvků – hranice systému, aktéry, případy užití a relace. [1]

3.2.1 Popis prvků diagramu

Při návrhu diagramu musíme nejprve určit hranice systému (System boundary), které určují, co patří, respektive nepatří do našeho systému. Od verze UML 2 se hranice systému nazývají také subjekt. Vně tyto hranice se nacházejí aktéři a uvnitř systému jsou jednotlivé případy užití.

V diagramu se hranice systému značí rámečkem s popisem názvu systému. Symboly aktérů se zakreslují vně obdélníku a případy užití uvnitř hranic. Příkladem názvu hranic systému může být *Nápojový automat*. [3]

Označení aktér (Actor) představuje roli, která zastupuje skupinu uživatelů, kteří mají v informačním systému stejná práva a přístup ke stejným funkcím. Označení aktéra může být například *Prodavač*. Je to obecná role, která v sobě zahrnuje více konkrétních jedinců. Jeden uživatel může představovat více rolí, například může být současně *Prodavač* a *Vedoucí prodejny*.

Všechny akce v systému jsou vyvolány aktéry. Jeden aktér může provádět několik případů užití a jeden případ užití může být prováděn více aktéry v jednom modelu. V pozici aktéra se kromě lidí může objevit i externí systém, který přímo komunikuje s naším systémem, nebo i čas, spouštějící naplánované úlohy, například záloha systému. [1][2]

Pro znázornění aktéra v diagramu případu užití se používá symbol postavičky, pod kterou se zapisuje název role, kterou představuje. Druhý způsob zobrazení aktéra je obdélník. [3]

Případy užití (Use Cases) si můžeme přestavit jako sled jednotlivých událostí (tzv. scénáře), které systém postupně vykoná na základě podnětu od jednoho nebo více aktérů. Tato posloupnost akcí jednoho případu užití vede vždy na společný cíl. Případy užití chápeme jako součást systému a vytváříme je z pohledu aktéra, který danou sekvenci inicializoval. Případ užití vyjadřuje, co budoucí systém nabídne uživateli, tedy výsledkem případu užití musí být něco, co používá některý z aktérů. Tvorba případů užití probíhá v několika opakování, dokud nejsou zahrnuty všechny vztahy mezi aktéry a systémem. [2]

Jak konkrétně bychom měli pojmenovávat případy užití, nestanovuje žádný standard jazyka UML. Přesto se v praxi osvědčilo, aby název případu užití tvořila slovesná vazba, jelikož představuje nějakou akci v systému, například *Založit novou objednávku*, *Editovat osobní údaje*. Název by měl být dále krátký, výstižný a jedinečný v rámci popisu jednoho informačního systému.

V diagramu se pro případ užití používá symbol elipsy, ve které je název případu užití. Vztah mezi aktérem a případem užití značíme jednoduchou spojnicí. [1]

Kromě základních vztahů mezi aktéry a případy užití můžeme použít i pokročilé typy relací (Relationships). Tyto nové vztahy slouží k definování dalších funkcionalit systému nebo k větší přehlednosti diagramu případu užití díky použití menšího množství symbolů.

Mezi tyto relace patří [1][3]:

- relace include
 - používá se u případů užití, které jsou společné nebo podobné pro více případů užití,
 - vyčlení se tím společné vlastnosti ze základních případů užití do nového samostatného případu užití,
 - základní případ užití již po tomto vyčlenění není bez rozšiřujícího případu kompletní,
 - značí se orientovanou čárkovanou čarou, která vede od závislého případu užití k případu užití, na níž závisí, a je popsána názvem include uzavřeným do dvojitých lomených závorek,

- relace extend
 - rozšiřuje základní případ užití o novou funkčnost, chování,
 - primární případ užití nezávislý na doplňujících informacích,
 - lze použít jen na určitých místech, tzv. bodech rozšíření (extension points), které poukazují na možné rozšíření funkčnosti systému,
 - značí se orientovanou čárkovanou čarou, která vede směrem od doplňujícího případu užití k základnímu a je popsána názvem extend uzavřeným do dvojitých lomených závorek,
- generalizace případu užití
 - představuje zobecnění vztahu mezi obecnějším a konkrétním případem užití,
 - seskupuje související případy užití do jednoho předka.

3.2.2 Diagramy případu užití systému

Pomocí výše zmíněných symbolů diagramu případu užití znázorníme tento diagram pro vyvíjený systém určený pro online cvičení a testování matematiky. Diagramy případu užití byly vytvořeny v programu Enterprise Architect.

Modely případů užití si zobrazíme v několika pohledech. Na obrázku 6 je nejprve zobrazen abstraktní pohled na případy užití, ve kterém jsou znázorněni všichni aktéři systému současně. Předvedeme si, jak jsme při modelování tohoto diagramu postupovali. Nejprve je potřeba stanovit hranice systému, jimiž jsou webové stránky *VypočtiTo* pro online procvičování a testování z matematiky žáků základních škol. Hranice systému zobrazíme jako prázdný obdélník s názvem systému.

V dalším kroku určíme aktéry ze specifikace systému, kterými jsou: nepřihlášený uživatel, žák, učitel a administrátor. Symboly pro tyto aktéry umístíme vně hranice systému. Abychom v další kroku správně identifikovali všechny případy užití, popíšeme si jednotlivá práva aktérů v systému.

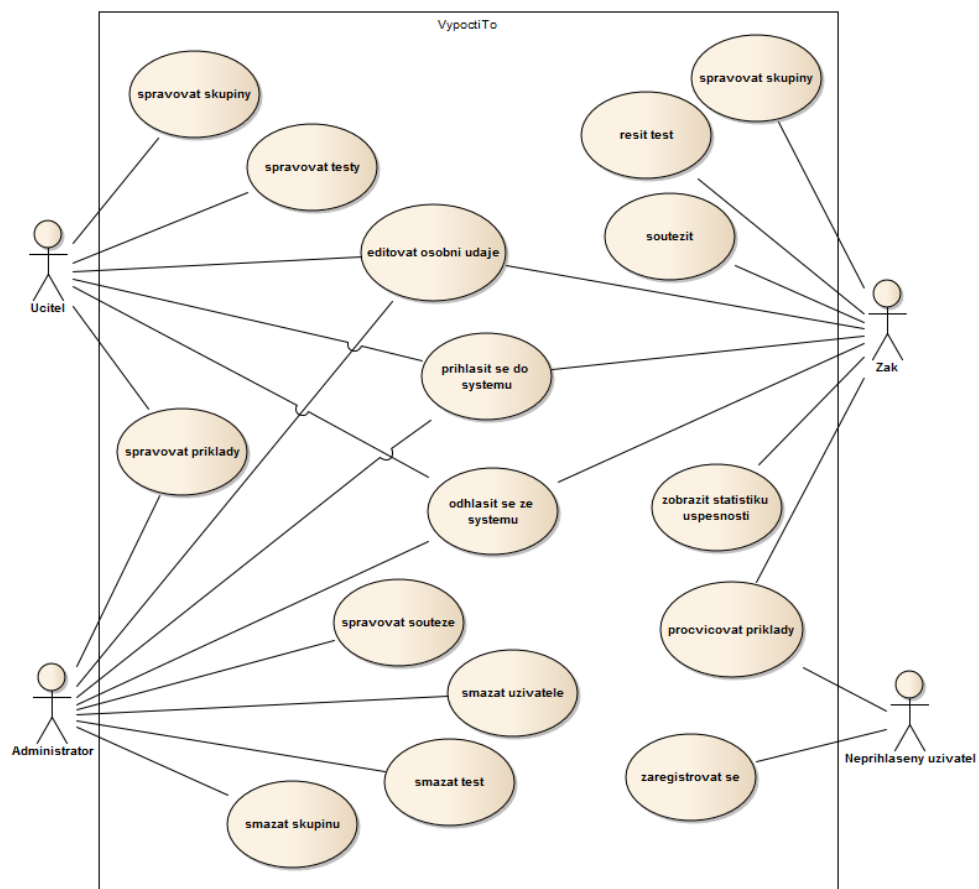
Popis aktérů:

- nepřihlášený uživatel
 - procvičuje příklady rozdělené podle témat,
 - volí si počet a obtížnost příkladů v procvičování,
 - zobrazí se mu statistika úspěšnosti po vyhodnocení cvičení,

- žák
 - přihlašuje se do systému a odhlašuje se z něj,
 - edituje své osobní údaje,
 - procvičuje příklady rozdělené podle témat,
 - volí si počet a obtížnost příkladů v procvičování,
 - zařazuje se do některé skupiny, resp. skupin,
 - skládá testy, které jsou dostupné v jeho skupinách,
 - účastní se soutěží,
 - zobrazí se mu jeho statistika úspěšnosti za dříve splněné cvičení, resp. testy,

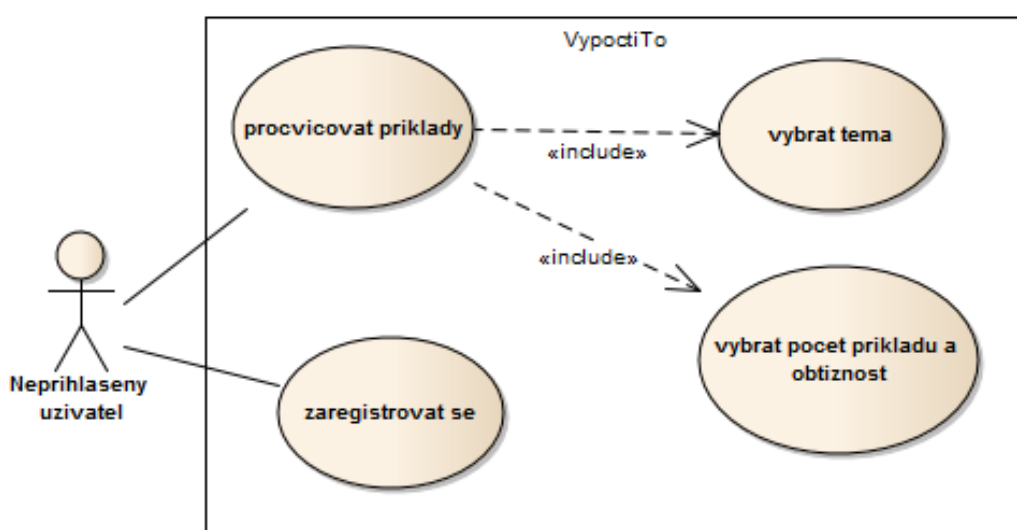
- učitel
 - přihlašuje se do systému a odhlašuje se z něj,
 - edituje své osobní údaje,
 - vytváří nové skupiny,
 - spravuje skupiny, tj. potvrzuje zařazení žáků do skupin, maže žáky ze svých skupin, edituje skupiny (včetně mazání),
 - spravuje své testy, tj. vytváří nové testy z příkladů v systému, edituje (včetně mazání) testy,
 - prohlíží si výsledky jednotlivých žáků z těchto testů,
 - přidává nový příklad do systému,

- edituje (včetně mazání) své příklady,
- administrátor
 - přihlašuje se do systému a odhlašuje se z něj,
 - přidává nové příklady do témat,
 - schvaluje nově vložené příklady od učitele, které učitel povolil zveřejnit,
 - edituje (včetně mazání) již vložené příklady v systému,
 - vytváří a edituje (včetně mazání) soutěže,
 - maže účet registrovaného uživatele – žáka, učitele,
 - maže vytvořené skupiny,
 - maže vytvořené testy.



Obrázek 6: Abstraktní pohled na případy užití systému.

Podle výše uvedeného popisu aktérů byly identifikovány jednotlivé případy užití a zakresleny uvnitř hranic systému. Nyní zbývá pouze vytvořit relace mezi aktéry a případy užití. V diagramu si můžeme všimnout, že některé činnosti se opakují u více aktérů, a proto je zakreslíme do návrhu pouze jednou a spojíme je s danými aktéry (např. případ užití „přihlásit se do systému“ je společný pro aktéry žák, učitel a administrátor). V abstraktním pohledu jsou využity pouze jednoduché vazby a specializované relace (*include*, *extend*, generalizace případů užití) jsou kvůli přehlednosti použity až v podrobnějších modelech.



Obrázek 7: Diagram případu užití z pohledu nepřihlášeného uživatele.

Na následujících obrázcích jsou postupně znázorněny rozpracované pohledy na případy užití jednotlivých aktérů systému – *nepřihlášený uživatel*, *žák*, *učitel* a *administrátor*. Je zřejmé, že nepřihlášený uživatel bude moci využívat pouze minimum z nabídky aplikace. Tento diagram obsahuje pouze dva hlavní případy užití. Navíc před procvičováním příkladů je třeba vždy vybrat kategorii, která se bude procvičovat a zvolit počet a obtížnost příkladů. V tuto chvíli je vhodné použít relaci *include*.

Na obrázku 8 je zobrazen rozpracovaný pohled na případy užití z pozice žáka. Tento aktér má určeny stejné případy užití pro procvičování příkladů jako nepřihlášený uživatel. Tentokrát ale nezůstává pouze u této činnosti, může se přihlásit do systému, odhlásit se z něho, změnit své osobní údaje atd. Kromě základních relací

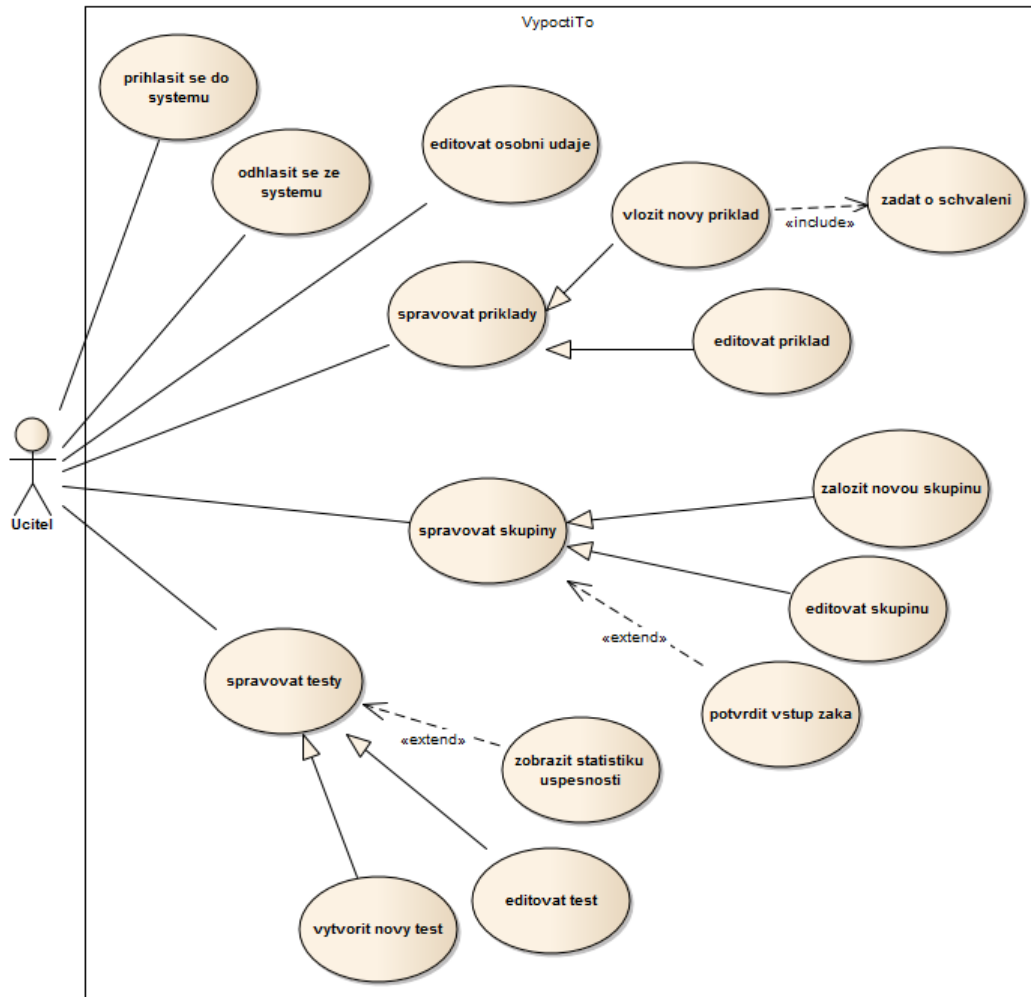
jsou zde použity relace *include*, *extend* a generalizace případu užití. Posledně jmenovaný vztah je použit u zobrazení statistiky úspěšnosti, která představuje zobecnění dvou typů statistik, které jsou si velmi podobné, ale v několika prvcích se odlišují (např. rozlišování stupně obtížnosti příkladů). Jedná se o statistiku úspěšnosti z procvičování a statistiku úspěšnosti z testů.



Obrázek 8: Diagram případu užití z pohledu žáka.

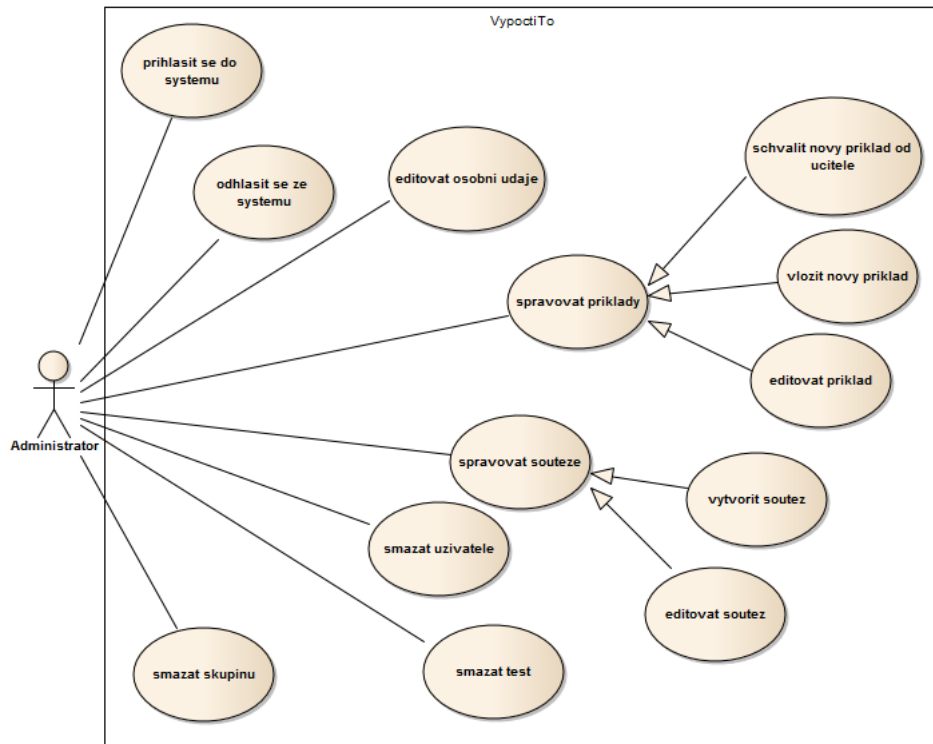
Na obrázku 9 vidíme náhled diagramu z pozice učitele. Tento aktér má několik shodných případů užití jako aktér žák (přihlásit se, odhlásit se ze systému a změna osobních údajů). Dalšími obecnými případy užití jsou *spravovat příklady*, *spravovat skupiny* a *spravovat testy*, které můžeme dále upřesnit na editaci a vytvoření nového

příkladu, respektive skupiny, testu. Tento vztah je vyjádřen pomocí generalizace případů užití. V případě vložení nového příkladu do systému musí učitel čekat na jeho schválení od administrátora. Tato vazba je znázorněna prostřednictvím relace include, jež musí být vždy vykonána.



Obrázek 9: Diagram případů užití z pohledu učitele.

Posledním aktérem systému, který je podrobněji zobrazen na obrázku 10, je *administrátor*. Je zřejmé, že tento aktér má přiděleny nejvyšší práva ke správě celého systému. Z toho vyplývají případy užití *smazat uživatele*, *smazat skupinu* a *smazat test*. K podrobnějšímu popisu případů užití *spravovat příklady* a *spravovat soutěže* je opět použita generalizace případů užití.



Obrázek 10: Diagram případu užití z pohledu administrátora.

3.3 Diagram aktivit

J. Arlow a I. Neustadt ([3], str. 285) definují diagram aktivity následovně:

Diagramy aktivit (z anglického originálu Activity Diagram) jsou „objektově orientovanými vývojovými diagramy“. Umožňují modelování procesu jako aktivity, která se skládá z kolekce uzlů spojených hranami.

Podle UML 2 se diagramy aktivit zakreslují zcela podle nové sémantiky, jež je založena na sémantice Petriho sítí. Diagram aktivit slouží k modelování chování systému a lze ho použít v různých fázích vývoje. Nejčastěji se však používá ve fázi analýzy, kde je využíván k modelování cest mezi případy užití. Diagram případu užití znázorňuje, co lze se systémem vykonávat a diagram aktivit specifikuje postupy, kterými toho dosáhneme. [3]

Chování v diagramu aktivit modelujeme pomocí imaginárního prvku – tokenu. Token „cestuje“ v síti uzlů a hran, které se řídí určitými pravidly. Token může reprezentovat postup řízení, objekt či určitá data. [3]

V diagramu lze rozlišovat několik rolí, které se účastní dané aktivity. Jednotlivé role rozdělí diagram na oddíly – tzv. paralelní dráhy. Někdy se oddíly aktivit nazývají také zónami odpovědností nebo plaveckými dráhami. Každá dráha je nadepsaná jménem role a obsahuje aktivity vykonávané danou rolí. Postup od jedné aktivity ke druhé může nastávat i mezi různými dráhami. Oddíly jsou odděleny svislými nebo vodorovnými čarami, případně křivkami. [2][3]

Aktivity se skládají ze dvou prvků – uzlů a hran. Tyto prvky se dále člení podle pravidel použití. Uzly mohou být akční, řídicí nebo objektové. Hrany členíme na řídicí a objektové.

Akční uzel iniciuje nějakou činnost (aktivita, chování nebo operace), u níž není potřeba dále ji členit. Akční uzel se aktivuje pouze v případě, že token projde zároveň všemi vstupními hranami uzlu a pokud splňuje všechny vstupní podmínky daného uzlu. Akční uzel se obvykle pojmenovává slovesem nebo slovesným spojením, i když UML 2 neuvádí žádný pevně stanovený standard. [3]

Řídicí uzel řídí cestu v rámci aktivity. Cestu mezi uzly znázorňujeme pomocí řídicího toku. Rozlišujeme několik základní typů řídicích uzlů:

- počáteční uzel
 - představuje počátek aktivity, který aktivuje prvotní postup tokenu,
- koncový uzel aktivity
 - ukončuje celou aktivitu,
- koncový uzel cesty
 - ukončuje jednu specifickou cestu v rámci aktivit,
 - zbylé cesty v aktivitě jsou stále aktivní,
- uzel rozhodnutí
 - uzel obsahuje více výstupních hran,
 - token pokračuje přes tu hranu, jejíž kontrolní podmínku splnil,

- jednotlivé podmínky hran musí mít prázdný průnik, tj. existuje pouze jedna cesta, kterou token může pokračovat,
- uzel sloučení
 - uzel obsahuje více vstupních hran,
 - vstupní tokeny se zkopírují do jedné výstupní hrany,
- uzel rozvětvení
 - rozvětví vstupní token do několika paralelních výstupních cest (hran),
- uzel spojení
 - synchronizuje několik paralelních cest. [3]

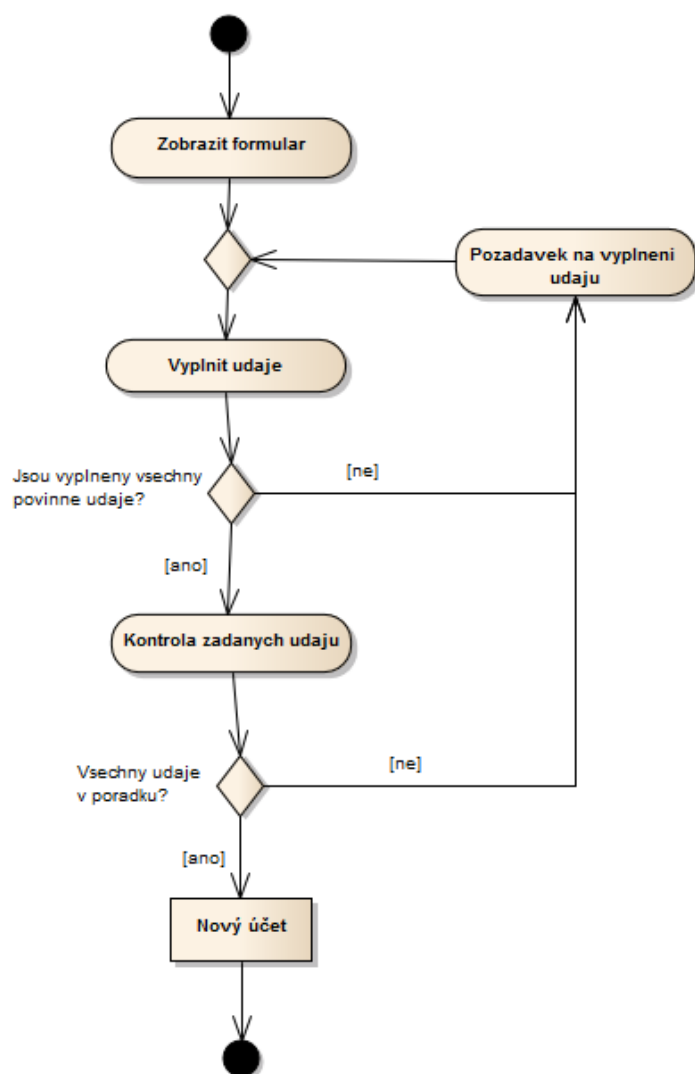
Objektový uzel zastupuje objekt v rámci aktivity. Cestu objektu, tj. vstupní a výstupní hrany tohoto uzlu, označujeme jako objektový tok (cesty objektu).

3.3.1 Diagramy aktivit aplikace

Na obrázku 11 je zobrazen diagram aktivit týkající se registrace uživatele. Skládá se z několika kroků – vyplnění všech požadovaných údajů, kontrola těchto údajů (např. tvar emailové adresy) a v případě nějakých nesrovnalostí je uživatel vyzván k opravě či doplnění chybných údajů.

Obrázek 19 v příloze zobrazuje založení nové skupiny z pohledu učitele. Ve *Správě skupin* zvolí volbu založit novou skupinu, zadá potřebné údaje a jestliže jsou vyplněny všechny povinné prvky formuláře, vytvoří se nová skupina.

Diagram na obrázku 20 v příloze popisuje procvičování příkladů z pohledu žáka a skládá se z logické posloupnosti akcí, jež bude stejná pro přihlášeného žáka i nepřihlášeného uživatele, a začíná vybráním jedné z nabízených kategorií. Následně musí uživatel zadat počet příkladů a obtížnost, které bude chtít počítat. Pokud zadal vyšší počet příkladů, než je možné zobrazit, je vyzván, aby zadal nový údaj. Poté systém zobrazí první zadání a uživatel se může pustit do počítání. Po potvrzení odpovědi systém zobrazí, zda žák počítal správně či nikoli. Následně se žákovi zobrazí



Obrázek 11: Diagram aktivit - registrace uživatele.

další příklad a až do vyčerpání zadaného počtu úloh. Na závěr se zobrazí statistika úspěšnosti.

3.4 Diagram tříd

Diagram tříd modeluje strukturu systému pomocí objektů, jež se v systému vyskytují, a jejich vzájemných vztahů. Kromě tříd může diagram obsahovat také rozhraní či výčtové třídy.

Diagram tříd lze vytvářet na několika úrovních – doménový, analytický, návrhový a implementační model, které se liší v zobrazení podrobností o jednotlivých třídách.

Doménový model představuje základní logickou strukturu systému a je sestaven podle požadavků na systém. Z něho vychází analytický model tříd, který rozšiřuje diagram o další kandidátní třídy a upřesňuje vztahy mezi jednotlivými entitami. Třídy mohou obsahovat názvy stěžejních atributů a operace, jež vyjadřují hlavní účel třídy. [3]

Další úrovní je návrhový model, který vzniká z doménového modelu doplněním implementačních detailů. Specifikujeme podrobněji atributy tím, že doplníme informaci o datovém typu, případně viditelnost a implicitní hodnoty. Dále sestavíme všechny metody specifikováním operací z doménového modelu. Při doplňování těchto detailů může nastat situace, že bude potřeba jednu třídu dekomponovat na několik podrobných návrhových tříd. Dále v této části zpřesňujeme vztahy mezi třídami.

Tvorbu diagramu tříd završuje implementační model, který již musí obsahovat veškeré podrobnosti struktury systému. S touto úrovní pracují většinou vývojáři systému. Při použití objektově orientovaného jazyka je možné převést diagram přímo na kód. Návrhový model rozšiřujeme o operace, které nastavují a vrací hodnoty atributů, tzv. gettery a settery. Dále můžeme třídy slučovat do balíků. [3]

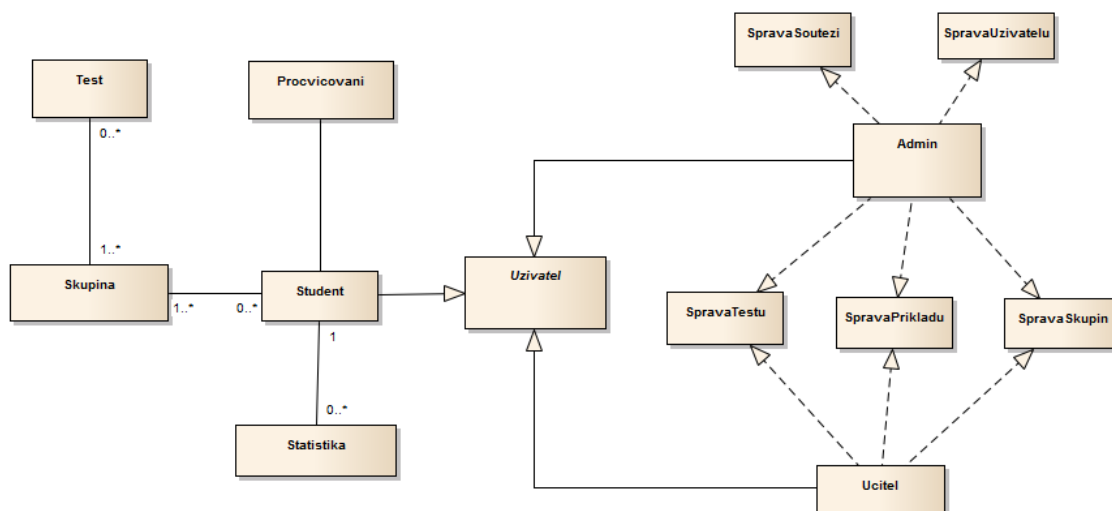
Při tvorbě diagramu tříd používáme následující vazby:

- dědičnost
 - hierarchický vztah rodič (předek) – potomek,
 - podřízená třída obsahuje své atributy a metody a zároveň dědí všechny atributy a operace své nadřízené třídy,
- asociace
 - modeluje vztah mezi dvěma třídami,
- agregace (seskupení)
 - speciální druh asociace,
 - modeluje vztah celek – část,
 - složený objekt (celek) může existovat bez svých komponent (částí),

- kompozice (složení)
 - speciální druh asociace,
 - modeluje vztah celek – část,
 - složený objekt (celek) nemůže existovat bez svých komponent (částí),
- násobnost (kardinalita)
 - představuje počet vzájemných výskytů objektů provázaných tříd v daném okamžiku. [4]

3.4.1 Diagram tříd aplikace

Z analýzy požadavků, která je popsána v kapitole 3.1, byl vytvořen prvotní doménový model ilustrující fungování aplikace. Tento model je zobrazen na níže zobrazeném obrázku.



Obrázek 12: Doménový model aplikace.

Uživatel systému může nabývat tří rolí – žák, učitel a administrátor. Administrátor spravuje celý systém, proto je mu přidělen největší množství funkcí. Učitelé mohou spravovat testy, vkládat nové příklady a skupiny a spravovat je. Žák může procvičovat příklady, prohlížet si statistiku z počítání, vstoupit do různých skupin nebo se počítat testy (soutěž je chápána jako druh testu).

Detailnější pohled diagramu tříd zobrazený prostřednictvím analytického modelu je kvůli své rozsáhlosti k dispozici v příloze na obrázku 21.

3.5 Datový model

Pojem databáze můžeme stručně popsat jako kolekci souvisejících dat shromážděných za určitým účelem. V některé literatuře se databáze definují v širším smyslu i se softwarovým systémem, jenž umožňuje definovat, vytvářet a udržovat databázi a poskytuje řízený přístup k uloženým datům. Tento systém se nazývá systém řízení báze dat (SŘBD).

K návrhu databázové struktury se nejčastěji používá entitně – relační model (ER model). Tento model se skládá ze dvou základních prvků – *entit* a *relací*. Entita představuje určitou množinu reálných objektů, které mají shodné vlastnosti (např. vozidlo, zboží, osoba) a je určena svým jménem a seznamem vlastností neboli *atributů*. Skupina atributů, jež jednoznačně identifikují příslušnou entitu, se nazývá *primární klíč*. Entity se zobrazují jako obdélník se jménem entity uvnitř. Relace (vztahy) si lze představit jako množinu spojení mezi zúčastněnými entitami. Každá vazba obsahuje popisek, jenž vystihuje její funkci, a znázorňuje se pomocí čáry, která spojuje potřebné entity. Také relace mohou být doplněny o atributy. [22]

3.5.1 Entitně – relační model aplikace

Při návrhu aplikace *VypočtiTo* byl využit výše popsaný entitně – relační model. Při návrhu databáze bylo dbáno na to, aby byla splněna třetí normální forma (3NF). To znamená, že musí být zajištěna atomicita hodnot, plná funkční závislost atributů na primárním klíči a neexistence tranzitivního vztahu mezi nepřimárními klíčovými atributy. Databáze byla navržena programem Navicat for MySQL⁸. Na obrázku 13 je zobrazen přehled tabulek databáze aplikace.

⁸<http://www.navicat.com/products/navicat-for-mysql>



Obrázek 13: Přehled tabulek v databázi.

Popis entitních množin:

- kategorie

Objekty entity jsou všechna témata, do kterých jsou rozděleny jednotlivé příklady v aplikaci. Tato entita obsahuje pouze dva atributy – id a název kategorie, jež jsou zároveň i primárním klíčem.

- příklady

Tabulka obsahuje zadání a další potřebné informace o jednotlivých příkladech.

Primárním klíčem je zde atribut `id`. Entita obsahuje atribut `obtiznost`, kde máme na výběr ze tří možností, `kategorie`, do které je příklad zařazen, `zdroj`, ze kterého je příklad převzat, a `obrazek`, pokud je potřeba k pochopení zadání. Atribut `autor` obsahuje jméno uživatele, který daný příklad do databáze vložil, a ten může prostřednictvím `verejne` nastavit, zda tento příklad bude viditelný ve veřejné části systému. Atribut `schvalen` obsahuje informaci o tom, zda prošel příklad kontrolou administrátora. Atribut `typ_ulohy` uchovává druh úlohy pro zobrazení odpovědi ve správném stylu. Entita obsahuje cizí klíč `id_odpoved` odkazující se na správnou odpověď k příkladu.

- `prikladyVKategorie`

Jedná se o asociační tabulku, jež slouží ke spojení tabulky `kategorie` a v ní umístěné `priklady`. Primární klíč je tvořený atributy `kategorie` a `priklad`.

- `obtiznost`

Tabulka obsahuje druhy obtížnosti, kterou si můžeme při počítání zvolit. Primární klíč je zde opět `id`.

- `odpovedi`

Prvky množiny jsou správné odpovědi, které jsou svázány s `priklady`. Primárním klíčem je `id_priklad`.

- `uzivatel`

Objekty entity jsou všichni uživatelé, kteří jsou evidováni v systému. Primárním klíčem je atribut `login`. Heslo je uloženo zabezpečeno pomocí hashovací funkce se „solí“. Atribut `prava` obsahuje informaci, zda je uživatel veden jako žák, učitel či admin. Význam ostatních atributů je zřejmý z pojmenování.

- `prava`

Tabulka obsahuje typy rolí, ze kterých si může uživatel vybrat, a je propojena s tabulkou `uzivatel`. Primárním klíčem je atribut `id`.

- `pohlavi`

Tabulka obsahuje doplňující informaci o pohlaví uživatele, je spojena s tabulkou `uzivatel` a je určena primárním klíčem `id`.

- statistikaProcvicovani

Zde nalezneme souhrnné informace o výsledcích uživatele z procvičování příkladů. Tabulka obsahuje složený primární klíč, který je tvořen z atributů `login`, `id_kategorie` a `obtiznost`, a je propojena s `uzivatelem`. Vlastnost entity `pocet_procvicovani` nám říká, kolikrát jsme počítali příklady z dané kategorie a s určitou obtížností.

- skupina

Objekty jsou skupiny, které vytvořili uživatelé v roli učitele. Jméno učitele obsahuje atribut `zakladatel` a odkazuje se na tabulku `uzivatel`. Primárním klíčem tabulky je atribut `id`.

- uzivateleSkupiny

Objekty entity jsou jednotliví uživatelé, kteří se zařadili do určité skupiny. Představuje asociační tabulku mezi tabulkami `uzivatel` a `skupina`. Primárním klíčem jsou atributy `id_skupiny` a `uzivatel`.

- test

Podle hodnoty atributu `typ` jsou prvky množiny buď testy vytvořené učitelem, nebo soutěže zadané administrátorem. Primárním klíčem je atribut `id`. Dále obsahuje již pouze atribut `nazev`.

- testyVeSkupine

Představuje asociační tabulku, jež propojuje tabulky `test` a `skupina`. Obsahuje atribut `opakovani`, který určuje, zda bude možné test spustit vícekrát. Atribut `zverejneni`, respektive `uzavreni` obsahuje údaje o čase a datu, kdy bude test žákům k dispozici (kdy se jim zpřístupní), respektive kdy se uzavře.

- prikladyVTestu

Představuje asociační tabulku, jež propojuje tabulky `test` a `priklady`.

- odpovediTest

Prvky množiny představují odpovědi na příklady v daném testu. Primárním klíčem je zde `login`, `id_test` a `id_priklad`. Jedná se o asociační tabulku mezi entitou `test` a `odpovedi`.

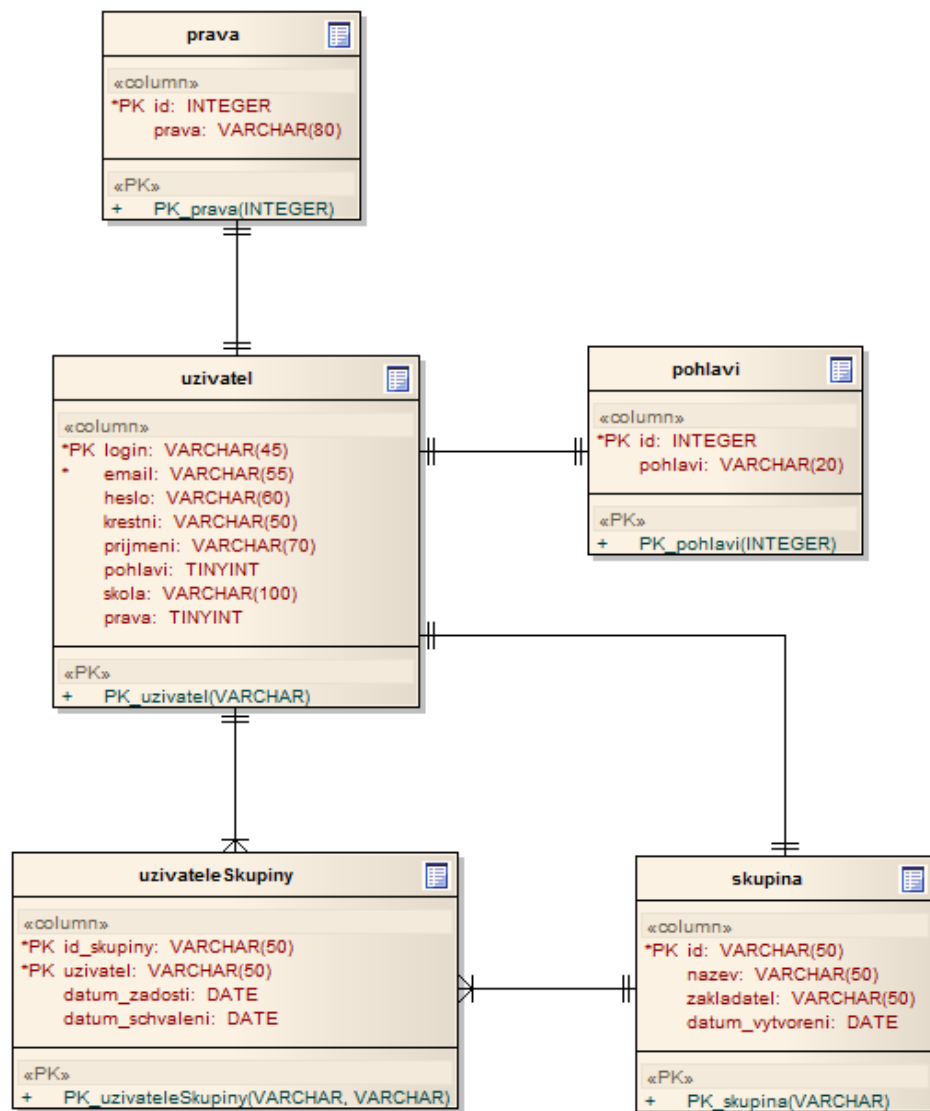
- statistikaTest

Zde nalezneme souhrnné informace o výsledcích uživatele z testů. Tabulka obsahuje složený primární klíč, který tvoří `login` a `id_test`, a je propojena s tabulkou `uzivatel` a `test`. Hodnota `pokus` nám říká, kolikrát již uživatel test spustil. Údaje o spuštění a dokončení testu obsahují atributy `zapocato` a `ukonceno`. Zbýlé atributy obsahují informace o počtu správných, chybných a nevyplněných odpovědí.

- zprava

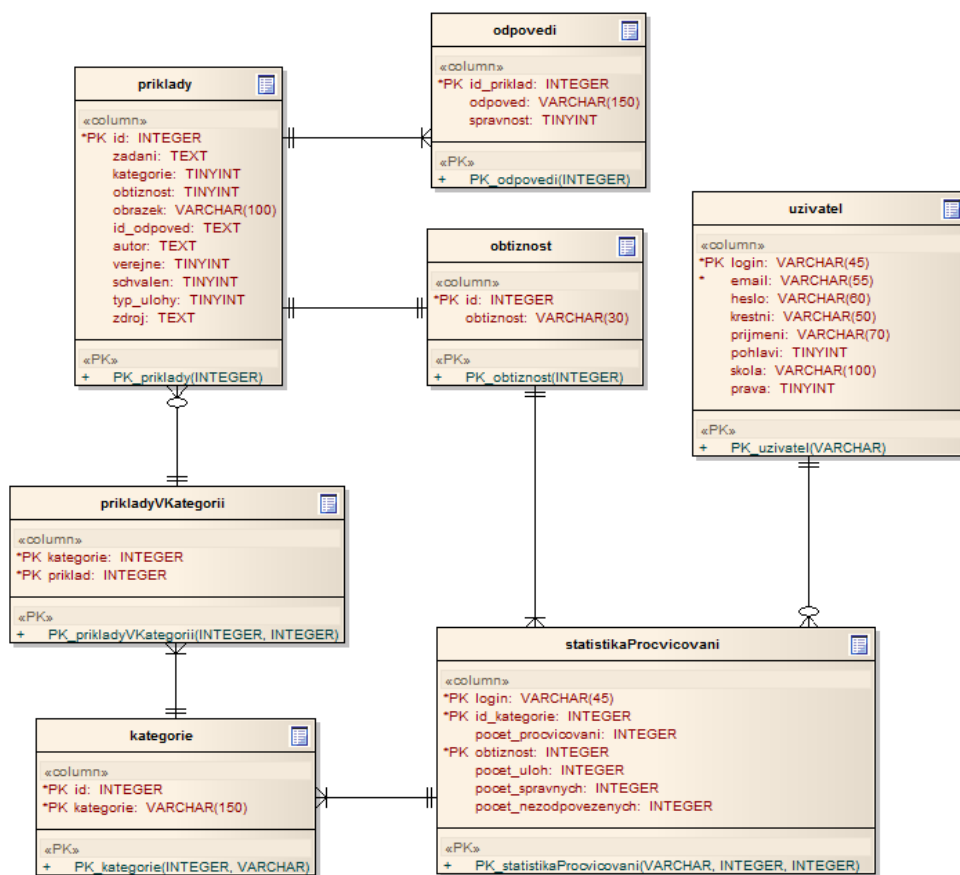
Zde jsou uloženy emailové zprávy, jež byly zaslány prostřednictvím kontaktního formuláře na webových stránkách. Primárním klíčem je atribut `id`, význam ostatních atributů je zřejmý z jejich názvů.

Na obrázku 14 je zobrazena základní tabulka `uzivatel`, která uchovává informace o uživateli. Dalšími důležitými tabulkami jsou ty, které se starají o nastavení oprávnění a přiřazení uživatele do skupin. Tabulka `uzivateleSkupiny` slouží jako asociační tabulka pro spojení tabulek `uzivatel` a `skupina`, kdy byl vztah N:M nahrazen vztahem 1:N. Zároveň jsou tabulky `uzivatel` a `skupina` spojeny ve vztahu 1:1, kdy jeden uživatel představuje zakladatele jedné skupiny. Doplňující informaci o pohlaví uživatele obsahuje tabulka `pohlavi`.



Obrázek 14: Práva uživatele a jeho přiřazení do skupiny.

Důležitými tabulkami jsou zde ty, které se starají o správu příkladů v systému. Ty jsou zobrazeny na obrázku 15. Ústřední tabulkou je zde tabulka `priklady`, která je dále propojena vztahem 1: N s tabulkami `odpovedi` a `prikladyVKategorie` a vztahem 1:1 je nastaven atribut `obtiznost` z tabulky `obtiznost`. Tabulka s názvem `statistikaProcvicovani` uchovává informace o úspěších či neúspěších žáků při počítání a je jednoznačně určena atributy `login`, `id_kategorie` a `obtiznost`, proto je také spojena vztahy 1:N, respektive N:1 s tabulkami, které jsou pojmenovány `uzivatel`, `kategorie` a `obtiznost`.



Obrázek 15: Procvičování příkladů.

3.6 Drátový model

Drátový model, tzv. wireframe, se využívá při návrhu struktury webových stránek. Tyto modely nastíní, jak by měl web přesně vypadat a fungovat, aniž by bylo potřeba pracně vytvářet grafiku či dokonce programovat. Pomocí drátových modelů si můžeme udělat jasnou představu o výsledném projektu a v případě nespokojenosti některé části rychle a bez dalších nákladů pozměnit.

Před vytvářením samotného drátového modelu je vhodné zakreslit jednoduchou mapu stránek, ze které bude zřejmé, které stránky jsou spolu propojeny. K tomu postačí jednoduché symboly obdélníků popsané názvy stránek, které představují, a spojit je jednoduchými vazbami. [5]

Při vytváření drátového modelu můžeme postupovat do různých úrovní detailů. Nejprve webovou stránku rozdělíme pouze na logické bloky (záhlaví, zápatí, menu, hlavní obsah) a poté doplňujeme další elementy (např. nadpisy, tlačítka, obrázky,

text apod.). Stále ale musí platit, že výsledkem bude pouze kostra stránek bez designu. Jednotlivé elementy, např. tlačítka, nadpisy, mohou obsahovat již finální popisky nebo mohou zůstat prázdné, resp. obecné a sloužit pouze pro představu rozložení prvků na stránce. [6]

Drátové modely je možné vytvářet pomocí tužky a papíru. Existuje však řada sofistikovanějších nástrojů, např. Axure RP, Visio atd. Modely webové stránky Vypoctito.cz byly vytvořeny v programu GUI Design Studio⁹.

3.6.1 Návrh webu

Titulní strana webových stránek obsahuje jednotlivé kategorie, ze kterých může uživatel vybírat téma pro procvičování. Dále se stránka skládá z hlavičky, jež obsahuje název webu, logo a komponenty potřebné pro přihlášení či registraci uživatele, z menu a patičky, jež obsahuje základní navigační prvky a formulář pro zasílání zpráv. Hlavní obsah stránky se po přihlášení zcela změní podle role uživatele, hlavička se změní částečně, menu a patička zůstávají stále stejné. V příloze je vložen drátový model titulní strany webu, náhled obrazovky po přihlášení žáka a po přihlášení učitele.

⁹<http://www.carettasoftware.com/guidesignstudio/>

4 Zabezpečení

Při tvorbě webové aplikace nesmíme zapomenout na zajištění její bezpečnosti. Tato otázka je v posledních letech čím dál aktuálnější a neměla by se podceňovat. Proto je důležité při vytváření webových stránek dodržovat určité zásady, které snižují rizika napadení stránek.

Mezi takové zásady patří používání apostrofů místo uvozovek, protože apostrofy zabraňují interpretaci proměnných a řídicích znaků. Pro vkládání souborů do aplikace by se měl používat příkaz `require()` místo `include()`, jelikož se tím vyhneme nepředvídatelným situacím. Když příkaz `require()` selže, vykonávání skriptu se zastaví (při vypnutí chybových hlášení se zobrazí např. pouze bílá stránka), zatímco v případě příkazu `include()` se může pokračovat dál. Pokud na vstupu od uživatele přijímáme číslo, je potřeba ho explicitně přetypovávat např. příkazem

```
$cislo = (int) $GET['cislo'];
```

nebo kontrolovat, zda se opravdu jedná o číslo. K tomu slouží funkce `is_numeric`, jež je použita i ve webové aplikaci. [7]

V následujících kapitolách jsou uvedeny další zásady, jež byly dodrženy v aplikaci *VypočtiTo*, a závažné hrozby, které by mohly nastat při používání aplikace napsané v jazyce PHP a využívající databázi.

4.1 Hlášení chyb

Nezbytnou součástí vývoje aplikace a jejího ladění je používání chybových hlášení. Jazyk PHP vypisuje informace o chybě do okna webového prohlížeče. Proto v případě selhání skriptu může tyto informace o aplikaci či konfiguraci serveru vidět i běžný návštěvník stránek a může je poté snadno zneužít k nepovolenému přístupu do systému. Je tedy potřeba, aby při ostrém spuštění aplikace na webu byla tato direktiva vypnuta. Toho docílíme tím, že v souboru `php.ini` nastavíme direktivu `error_reporting = 0`. Druhou možností, jak vypnout zobrazování chyb, je použití direktivy `display_errors = Off`.

O chybové zprávy jsme se tímto nastavením ale úplně nepřípravili. Můžeme je nechat protokolovat do logovacího souboru, který specifikujeme v direktivě `error_log`. Tuto funkci aktivujeme zapnutím direktivy `log_errors`. [7]

4.2 Zakázání nebezpečných funkcí

Jazyk PHP nabízí programátorům celou řadu funkcí, některé z nich se mohou stát bezpečnostním rizikem. Proto je vhodné tyto funkce z preventivních důvodů uživatelům znepřístupnit. Toho dosáhneme tím, že v souboru `php.ini` napíšeme direktivu `disable_functions` a za ní seznam daných funkcí oddělených čárkou. Mezi takové funkce patří (a které byly v aplikaci vypnuty) například:

- `phpinfo()`
 - zobrazí komplexní informace o konfiguraci PHP – používaná verze PHP, operační systém, verze jádra atd.
- `fopen()`
 - slouží k otevírání souborů a URL adres. [17]

4.3 Ukládání hesel

Jedním z hlavních prvků sloužících k zajištění bezpečnosti webové aplikace, ve které se mohou uživatelé zaregistrovat, je ochrana přihlašovacího hesla. Ukládání hesla ve stejné podobě, v jaké jej zadal uživatel, tj. bez jakékoli ochrany, je v současné době považováno za hrubou chybu. Proto se k ukládání hesla využívají hashovací funkce, které zadaný řetězec převedou na hash pevné délky. Délka hashe je ovlivněna použitým typem hashovací funkce.

Heslo uživatele by se nemělo dozvědět ani správce aplikace, který má přístup k algoritmu hashovací funkce. Proto se do databáze ukládá otisk (hash) hesla a použitá funkce je pouze jednosměrná, tj. z hashe nelze v rozumné době složit původní heslo. Když se poté uživatel chce přihlásit do systému, zadá své heslo a aplikace ho stejnou funkcí změní na hash, který porovnává s hodnotou otisku uloženou v databázi. [13]

Pro zvýšení bezpečnosti systému se kromě samotné hashovací funkce používá „sůl“. Sůl je jedinečný řetězec přiložený k heslu uživatele. Teprve takto rozšířené heslo se zahashovává. Je to z toho důvodu, aby v případě výskytu více uživatelů se stejným heslem, se zabránilo situaci, že útočník prolomí jedno heslo a tím získá hesla i všech ostatních uživatelů. Heslo můžeme „osolit“ například jménem uživatele, datem registrace uživatele nebo nějakým speciálním řetězcem, který však nebude uložen v databázi aplikace.

Mezi nejčastěji používané hashování funkce patří MD5 a SHA1. Výsledkem funkce MD5 je 32 znakové číslo v hexadecimálním tvaru a výsledkem SHA1 je 40 znakové číslo také v hexadecimálním tvaru. Velkou nevýhodou hashovací funkce MD5 je, že u ní byly objeveny bezpečnostní trhliny. Dochází k tomu, že k různým heslům je vytvořen stejný hash. Tyto kolize jsou např. v digitálních podpisech nepřípustné. Pro vývojáře je tedy vhodné, aby používali dokonalejší algoritmy např. SHA2, SHA3 nebo výše zmíněný SHA1. [13][14]

V aplikaci *VypočtiTo* je použita hashovací funkce rozšířená o „sůl“ k utajení hesla uživatele. Z důvodu bezpečnosti si zde nebudeme uvádět přesný název použitého algoritmu. Dále aplikace obsahuje dvojí kontrolu hesla, to znamená, že při změně hesla musí uživatel zadat nové i původní heslo. Tím zabráníme tomu, aby přihlášenému uživateli mohl někdo cizí změnit heslo (když si např. na chvíli odběhne od počítače a neodhlásí se) a mohl tak získat snadný přístup k danému účtu. Také při registraci musí uživatel vyplňovat heslo dvakrát a kontroluje se, zda si hesla odpovídají. V tomto případě je tak zajištěno, že se uživatel při zadávání nedopustí nějakého překlepu, kterého si není vědom, což by způsobilo, že se na nově vytvořený účet nebude moci do systému přihlásit.

4.4 SQL Injection

SQL Injection (narušení SQL dotazu) je označení bezpečnostní chyby, díky které má útočník možnost vkládat vlastní SQL dotazy a manipulovat tak s daty v databázi. Ukažme si jednoduchý příklad dotazu:

```
$dotaz="SELECT * FROM uživatel WHERE login='admin' OR 1=1";
```

Tento dotaz se vykoná pokaždé, protože kvůli logickému operátoru OR je podmínka vždy vyhodnocena jako pravdivá (true). Proto uživatel bude vždy do systému přihlášen a navíc se přihlásí na první účet v tabulce, jenž obvykle bývá používán přímo administrátorem nebo nějakým testovacím účtem se stejnými právy jako má administrátor.

Bránit se proti takovému zásahu do kódu můžeme validací vstupních dat nebo například ošetříme-li vstupní data funkcí `mysql_real_escape_string`. [19]

4.5 XSS

XSS (Cross-Site Scripting) neboli skriptování přes weby představuje závažné pochybení ve filtraci dat od uživatele. Při tomto útoku nedochází k úpravě kódu uloženého na serveru, ale způsobuje při návštěvě uživatele napadené stránky odesílání dat třetí straně. To může vést k tomu, že útočník odcizí nic netušícímu uživateli jeho identitu a může se za něho vydávat. Zabránit těmto útokům lze v jazyce PHP hned několika funkcemi: `escapeshellarg()`, `escapeshellcmd()`, `htmlentities()` a `strip_tags()`. Od verze PHP 5.2 má programátor přístup k nativnímu rozšíření `Filter`, jenž nabízí širokou škálu různých druhů filtrů. V aplikaci *VypočtiTo* je použita funkce `filter_var`, která rozpozná v zadaném řetězci speciální symboly a nahradí je jejich znakovými entitami. [7]

Jedním z nejčastějších problémů na vstupu je výskyt uvozovek a apostrofů. Tento problém je možné řešit (včetně již výše zmíněných funkcí) například tím, že znaky nahradíme tzv. escape sekvencí (k tomu slouží funkce `addslashes()`). [7]

4.6 Direktiva `register_globals`

K přenosu dat od uživatele jsou určeny tři metody – GET, POST a COOKIE. Pokud je zapnuta direktiva `register_globals`, zobrazí se proměnné předávané výše uvedenými metodami jako globální proměnné v daném skriptu. To může vést k tomu, že případný útočník vloží do skriptu své proměnné s potřebnými hodnotami a prolomí tak zabezpečení celé aplikace. Aby tato situace nenastala, je nutné používat superglobální proměnné `$_POST`, `$_GET` a `$_COOKIE`. Dále je zapotřebí mít vy-

pnutou direktivu `register_globals`, kterou mívají poskytovatelé webhostingu již implicitně vypnutu. Navíc od verze PHP 5.4 tato direktiva již ani neexistuje a nelze ji tedy povolit. [18]

4.7 Soubor `.htaccess`

Jedním z dalších vážných bezpečnostních rizik je nekontrolované prohlížení adresářové struktury aplikace prostřednictvím webového prohlížeče. Uživateli stačí bližší prozkoumání adresného řádku a již si může dle libosti prohlížet obsah jednotlivých složek. Této situaci zamezíme použitím souboru `.htaccess`. Postačí, když do souboru zapíšeme následující řádek:

`Options -Indexes`

`.htaccess` je textový soubor, který můžeme vytvořit v obyčejném textovém editoru (např. v Poznámkovém bloku). V operačním systému Windows mohou nastat menší problémy při vytváření tohoto souboru, protože obsahuje pouze příponu bez názvu. Kromě výše uvedené funkce, která je použita i ve webové aplikaci *VypočtiTo*, můžeme v souboru `.htaccess` nastavit:

- výpis chybového oznámení,
- blokování přístupu ke stránkám pro určité IP adresy,
- heslo pro ochranu části nebo celých stránek,
- uživatelské skupiny,
- přesměrování na jinou adresu,
- apod. [15][16]

5 Implementace

Následující kapitoly stručně popisují nástroje, postupy a technologie, které byly použity při tvorbě interaktivních webových stránek *VypočtiTo*, jež jsou dostupné na adrese <http://www.vypoctito.cz>. Veškeré zdrojové kódy byly napsány v programu PhpStorm¹⁰ a pro usnadnění práce s databází byl použit program Navicat for MySQL¹¹.

5.1 Použité technologie

Pro implementaci webové aplikace byl zvolen jazyk PHP, šablonovací systém Smarty, databázový systém MySQL, značkovací jazyk HTML5 a kaskádové styly CSS3. Tuto kombinaci jsem zvolil kvůli široké nabídce funkcí, které výše zmíněné technologie nabízejí a také vysoké podpoře těchto nástrojů ze strany poskytovatelů hostingu. Další technologie, která byla použita při tvorbě webu, avšak v menší míře, je JavaScript.

Webové stránky, jak je již ze zadání patrné, obsahují řadu matematických zápisů, se kterými si jazyk HTML, resp. XHTML nedokáže poradit. Pro zobrazení těchto prvků proto byla použita javascriptová knihovna MathJax.

Veškeré výše zmíněné technologie budou v následujících řádkách stručně charakterizovány.

5.1.1 HTML

HTML je zkratka z anglického názvu HyperText Markup Language, jež znamená hypertextový značkovací jazyk. HTML je jedním z předních jazyků používaných pro vytváření webových stránek.

Zdrojový kód je vytvářen pomocí HTML značek, neboli tagu, podle kterých prohlížeč rozpozná, jak má dané části stránky zobrazit. Tagy mohou být párové a obsah je mezi nimi uzavřen (např. pro zobrazení odstavce) nebo nepárové (např. pro zobrazení obrázku). Některé elementy mohou obsahovat atributy, které nastavují

¹⁰<http://www.jetbrains.com/phpstorm/>

¹¹<http://www.navicat.com/products/navicat-for-mysql>

vlastnosti daného prvku. [12]

Dokument obsahující HTML kód má příponu `.html` nebo `.htm`. Každý dokument musí obsahovat informace o použitém typu dokumentu (specifikace DTD). Základní strukturu dokumentu tvoří následující elementy:

- `html` – uvozuje a zakončuje celou webovou stránku,
- `head` – vymezuje hlavičku dokumentu, jež obsahuje informace o stránce (tzv. metadata),
- `title` – titulek stránky, zobrazuje se v horním pruhu prohlížeče,
- `body` – vymezuje tělo stránky, do kterého se zapisuje veškerý obsah HTML stránky. [12]

5.1.2 CSS

CSS je akronym ze slov *Cascading Style Sheets* a do češtiny je překládán jako kaskádové styly. Tento jazyk byl vytvořen organizací W3C a je určen pro grafickou úpravu webových stránek. Využívá se zejména ve spojení s technologiemi HTML, XHTML a XML. Poslední verzí jazyka je CSS3.

Díky tomuto jazyku je možné oddělit nastavení vzhledu od obsahu samotného dokumentu, což umožňuje jednodušší údržbu webové prezentace. Vlastnosti tagů v HTML dokumentu lze nastavit buď globálně pro všechny tagy stejného typu, nebo pouze jednomu konkrétnímu prostřednictvím návštějí. [10]

Pomocí pouze jediného souboru, který obsahuje pravidla pro vzhled stránky, můžeme nastylovat celý web. Existuje několik způsobů, jak připojit externí CSS soubor (s příponou `.css`) do HTML dokumentu. Ukážeme si dva nejčastější způsoby:

- použití elementu `link`

```
<link rel="stylesheet" type="text/css" href="soubor.css"/>
```
- použití direktivy `import`

```
@import url(style.css) screen;
```

CSS styly lze v HTML dokumentu použít i přímo vložením atributu style. Tím se však poruší hlavní výhoda kaskádových stylů – oddělenost vzhledu od obsahu. Pomocí všech výše uvedených vložení CSS můžeme nastavit různé způsoby zobrazení webu pro prohlížeč, tisk, mobily apod. [10]

5.1.3 PHP

PHP je rekurzivní akronym pro Hypertext Preprocessor. Původně ale akronym PHP vznikl z anglického spojení Personal Home Page. PHP je skriptovací jazyk, který je určen především pro programování dynamických webových stránek. Zdrojové kódy jsou zpracovávány na straně serveru a uživateli (např. webovému prohlížeči) je vrácena výsledná odpověď v určitém formátu (např. HTML, JSON, XML).

Kód jazyka PHP se vkládá přímo do HTML dokumentu, kde musí být oddělen od ostatních HTML prvků pomocí značek `<? a ?>` nebo `<?php a ?>`.

Jazyk PHP podporuje řadu knihoven – pro přístup do databáze, pro zpracování textu, grafiky apod. Podporuje také řadu internetových protokolů – HTTP, SMTP, POP3, FTP apod., protokol http využívá ke komunikaci mezi webovým serverem a prohlížečem.

Mezi výhody jazyka PHP bychom mohli zařadit nezávislost na platformě, existenci velkého množství funkcí a podporu mnoha databázových systémů (např. MySQL, Oracle, ODBC, MSSQL), je téměř standardem v hostingových službách apod. [7][8]

5.1.4 Smarty

Smarty je šablonovací systém pro jazyk PHP. Šablonovací systém (emplating engine) slouží k oddělení aplikační (samotný kód PHP) a prezentační (klasické HTML) logiky aplikace. Systém vytvořili Andrei Zmievski a Monte Orte a byl uvolněn pod licencí GNU Lesser General Public License (LGPL).

Hlavní výhody použití šablon jsou:

- Grafik může pracovat na aplikaci téměř nezávisle na vývojáři. Není nutné, aby byl zároveň zdatným programátorem, protože syntaxe používaná v šablonovacích systémech, bývá obvykle jednodušší než syntaxe programovacího jazyka, který používá vývojář.

- Snižuje redundanci kódu a komplikovanost údržby, která by mohla vzniknout, pokud bychom chtěli část kódu modifikovat pro více si podobných, ale zároveň různých použití (např. generování dat pro tisk, web, zprávy elektronické pošty apod.). [7]

Mezi přednosti systému Smarty patří možnost použití řídicích struktur, cyklů, průchod vícerozměrnými poli apod. Další výhodou tohoto systému je jeho rychlost, které dosahuje díky tomu, že jednotlivé šablony kompiluje a vytváří z nich srovnatelné skripty PHP, jež se poté ukládají do mezipaměti a odtud se následně načítají, pokud nedošlo k žádné změně šablony. [8]

```
<html>
<head>
<title> {$title} </title>
</head>
<body>
<p> Tady je {$name}. Vítejte v báječném světě Smarty. </p>
</body>
</html>
```

Obrázek 16: Návrhová šablona Smarty (welcome.tpl). [7, str. 390]

```
<?php
require(\Smarty.class.php");
$smarty = new Smarty;
//přiřadí dvě proměnné Smarty
$smarty->assign(\name", \Jason Gilmore");
$smarty->assign(\title", \Šablonovací system Smarty!");
//získá a vypíše šablonu
$smarty->display(\welcome.tpl");
?>
```

Obrázek 17: Aplikační logika šablony index.tpl (index.php). [7, str. 390]

Výše je vidět jednoduchý příklad šablony, která obsahuje dvě proměnné - `$title` a `$name`. Proměnné jsou uzavřeny do složených závorek, které fungují v systému Smarty jako oddělovače a upozorňují systém, že má s obsahem uvnitř dále pracovat. V zobrazeném příkladu se tyto proměnné nahradí skutečnými hodnotami, což je pouze jedna ze základních možností využití systému Smarty. Pod návrhovou šablonou je zobrazena aplikační logika šablony, kde jsou uvedeny hodnoty, které se budou dosazovat za výše zmíněné proměnné. [7]

5.1.5 MySQL

MySQL (My Structured Query Language) je relační databázový systém. MySQL představuje jednu z nejrozšířenějších multiplatformních databází, jež povoluje víceuživatelský přístup a přístup k datům je realizován prostřednictvím jazyka SQL (Structured Query Language).

Systém je vyvíjen společností Oracle a je nabízen pod dvěma licencemi – GNU General Public License a pod komerční licencí. Jako open source nese název MySQL Community Server a komerční verze nese označení MySQL Enterprise Server. Placená verze obsahuje navíc přidané některé aplikace a služby.

Mezi výhody tohoto systému můžeme uvést: cenu, stabilitu, jednoduchost, funkční výbavu a modulárnost. [7]

5.1.6 JavaScript

JavaScript je skriptovací jazyk, který je určen pro zpracování dat na straně klienta a interakci s webovými stránkami. Jazyk vyvinula v roce 1995 společnost Netscape. JavaScript je nejčastěji používán pro tvorbu dynamického webu.

Tento jazyk se skládá ze tří hlavních částí:

- jádra (ECMAScript)
 - poskytuje základní funkční prvky,
 - jazyk ECMAScript je definován ve standardu ECMA – 262,
- objektový model dokumentu (DOM – Document Object Model)

- je určen pro práci s obsahem webové stránky,
- umožňuje měnit obsah stránky, aniž by bylo potřeba ji nechat znovu celou načíst,
- model byl standardizován asociací W3C,
- objektový model prohlížeče (BOM – Browser Object Model)
 - slouží k interakci s prohlížečem,
 - jediná část jazyka, která není standardizovaná. [9]

Kód napsaný v JavaScriptu se nazývá skript. V HTML kódu vytvoříme odkaz na daný skript a stránka si samotný JavaScript sama stáhne. Soubory obsahující JavaScript musí mít příponu .js.

V dnešní době je podporován snad všemi předními webovými prohlížeči (Internet Explorer, Firefox, Chrome, Opera a Safari), i když v různém rozsahu. [9]

5.1.7 MathJax

MathJax je javascriptová knihovna pro zobrazování matematických symbolů na internetových stránkách využívající notaci LaTeXu, MathML nebo AsciiMath. Jedná se o multiplatformní a otevřený software. Vznikl z dřívější javascriptového systému jsMath, který vyvinuly společnosti American Mathematical Society (AMS), Design Science a Society for Industrial and Applied Mathematics (SIAM). V roce 2009 vytvořily tyto společnosti MathJax Consortium, jenž se skládá z řady odborníků, společností a je podporováno řadou sponzorů.

Pro využívání MathJaxu není nutno instalovat žádné pluginy. Jediné co je vyžadováno je podpora JavaScriptu v prohlížeči. V současné době MathJax podporují všechny předních webové prohlížeče (Internet Explorer, Firefox, Chrome, Opera a Safari), mobilní zařízení a tablety. [11]

K zobrazení matematického textu a symbolů používá MathJax webová písma. V případě, že tato služba není prohlížečem podporována, hledá MathJax použitelná písma v uživatelově systému. Pokud se nepodaří najít vhodné písmo ani tímto způsobem, zobrazí MathJax matematické symboly jako obrázek.

Při používání MathJaxu se tvůrce webu nemusí obávat rozbití vzhledu stránky, neboť MathJax využívá moderní CSS a webové fonty, které umožňují zobrazit matematický text ve stejném měřítku jako okolní text, a to i při zvětšování stránky.

Velkou předností této knihovny je, že takto zobrazené písmo rozpoznají i čtecí zařízení zrakově postižených a lze ho zvětšovat pomocí běžných nástrojů prohlížečů bez újmy na kvalitě zobrazení. Další výhodou je možnost běžně kopírovat matematický text do textových dokumentů (např. MS Word), LaTeX dokumentů, matematických programů (např. Maple, Mathematica) apod. Díky tomu, že jsou matematické prvky s MathJax založeny více na principu textu než na obrázku, nebrání nic vyhledávačům k jejich nalezení. [11]

Ukažme si jednoduchý příklad - zápis vztahu pro binomické rozdělení pomocí jazyka LaTeX. Pro správné zobrazení MathJaxem je potřeba, aby byl příkaz uzavřen mezi symboly $\[a \]$ nebo $\(a \)$. Na obrázku níže vidíme, jak si s následujícím vzorcem poradil MathJax.

$\[P(E) = \{n \ \text{choose} \ k\} \ p^k \ (1-p)^{\{n-k\}}\]$

$$P(E) = \binom{n}{k} p^k (1-p)^{n-k}$$

Obrázek 18: Zobrazení vzorce pro binomické rozdělení pomocí MathJaxu. (zdroj: <http://www.mathjax.org/demos/tex-samples/>)

5.2 Jednotkové testování

Jednotkové testy slouží k ověření, zda jednotlivé komponenty (jednotky) programu dělají přesně to, co se od nich očekává. Jednotky představují nejmenší testovatelnou část aplikace. V tomto případě máme na mysli třídy a metody, které je potřeba testovat izolovaně od dalších částí kódu či systému, tj. neměly bychom přistupovat k Internetu, používat databázi, komunikovat s dalšími třídami apod.

Jednotkové testy lze využívat již v počátcích vývoje aplikace a průběžně se k nim vracet. Je zřejmé, že ani pomocí opakovaného testování nelze odhalit veškeré chyby, obzvláště některé integrační chyby nebo širší systémové chyby. Pro snadnější práci s těmito testy se používá tzv. xUnit Framework. V aplikacích vyvíjených v jazyce

PHP se používá framework PHPUnit, který si podrobněji popíšeme v následující podkapitole. [20]

5.2.1 PHPUnit

PHPUnit je framework pro jednotkové testování, který je dostupný zdarma z repozitáře PEAR. Framework pochází z dílny Sebastiana Bergmanna. Každý test představuje samostatnou třídu, jež dědí od `PHPUnit_Framework_TestCase`. Každá z těchto tříd obsahuje metody (testovací případy) určené k testování daných jednotek. Pokud budeme chtít daný test spustit, stačí zavolat příkaz `phpunit` a jako parametr uvést cestu ke složce s testy.

Je dobrým zvykem v názvu metody uvádět prefix „test“, což umožní, aby si tyto metody PHPUnit sám zavolal. Další způsob, jak můžeme metodu označit, je použitím anotace `@test`. Dále je vhodné za slovo „test“ uvést název odpovídající metody, jež je testována. Naopak u testovací třídy je dobrým zvykem slovo „test“ psát v názvu jako postfix. Tento tvar pojmenování opět zajistí, že si je PHPUnit sám vyhledá. Testovací třída se označuje jako sada testů a testovací případ (metoda) se označuje pojmem „test case“. [8]

Aplikace *VypočtiTo* byla otestována z hlediska funkčnosti automatickými testy na úrovni kódu pomocí frameworku PHPUnit ve vývojovém prostředí PhpStorm.

5.3 Verzování

Systém správy verzí (zkráceně VCS z anglického originálu Version Control System) je využíván vývojáři ke správě změn, respektive obsahu. Při vývoji určitého projektu je vhodné mít k dispozici předešlé verze, jestliže by například současná verze projektu nevedla ke zdárnému výsledku. Scott Chacon ([21], str. 17) definuje správu verzí následovně:

„Správa verzí je systém, který zaznamenává změny souboru nebo sady souborů v průběhu času, a uživatel tak může kdykoli obnovit jeho/jejich konkrétní verzi (tzv. verzování).“

Rozlišujeme následující tři typy systémů správy verzí: [21]

- lokální
 - nejjednodušší způsob je vytvořit kopii souborů v jiném adresáři,
 - pokročilejší způsob je použití některého lokálního systému VCS s jednoduchou databází, ve které jsou uloženy jednotlivé verze souborů,
 - např. nástroj rcs,

- centralizované
 - zkráceně CVCS z anglického originálu Centralized Version Control System,
 - systém je vhodný pro projekty, na nichž se podílí více osob (každý má přehled o práci ostatních v týmu),
 - data (verzované soubory) jsou uložena pouze na centrálním serveru, jež sdílí vývojáři (nelze tedy pracovat jinak než online),
 - uživatel si stáhne pouze nejnovější verzi souboru (tzv. snímku),
 - dojde-li k výpadku serveru, jdou všichni uživatelé odříznuti od centrální databáze, v nejhorším případě může dojít ke ztrátě dat,
 - vzhledem ke sdílení dat mezi více osob dochází ke zpomalení celého systému,
 - je potřeba přidělit práva zápisu do centrálního repozitáře,
 - např. nástroje CVS, Subversion, Perforce,

- distribuované
 - zkráceně DVCS z anglického originálu Distributed Version Control System,
 - i tento systém je vhodný pro projekty řešené v týmech,
 - oproti centralizovanému systému správy verzí si uživatelé stahují kompletní kopii databáze a systém je takto chráněn před nechtěnou ztrátou dat např. kvůli výpadku,

- spolupracují i s jinými vzdálenými repozitáři,
- např. nástroje Git, Mercurial, Bazaar, Darcs.

V následující kapitole si blíže popíšeme zřejmě nejznámější distribuovaný systém správy verzí – Git, jenž byl použit při tvorbě aplikace *VypočtiTo*, s grafickou nadstavbou SmartGit¹².

5.3.1 Git

Nástroj Git vytvořila komunita vývojářů Linuxu (převážně Linus Torvalds) v roce 2005 kvůli licenčním neshodám u jiného verzovacího systému BitKeeper. Vývojáři používali BitKeeper ke správě vývoje jádra Linuxu. Proto musela komunita Linuxu vytvořit takový DVCS systém, který by byl velmi rychlý, měl jednoduché ovládání, uměl spravovat i rozsáhlé projekty apod.

Git pracuje na jiném principu než ostatní systémy VCS. VCS systémy obvykle uchovávají základní verzi souborů a seznam změn, jež byly provedeny za daný čas. Oproti tomu Git uchovává kromě základní verze souborů také i nově změněné soubory, z nichž vytvoří tzv. snímek. Pokud se soubor oproti předchozí verzi nijak nezměnil, uloží Git kvůli úspoře místa pouze referenci na originální soubor. Soubory se uchovávají v komprimované podobě.

Díky tomu, že uživatel má na lokálním disku všechny potřebné soubory, může pracovat kdykoli a kdekoli, aniž by bylo potřeba připojení k síti. Uživatel pracuje offline a jakmile se připojí k síti, aktualizují se soubory v repozitáři. Z tohoto důvodu je systém Git i velmi rychlý. Zobrazuje historii projektu nebo provádí lokální výpočty téměř neprodleně. [21]

¹²<http://www.syntevo.com/smartgithg/>

6 Ověření v praxi

Vytvořená aplikace *VypočtiTo* byla vyzkoušena na dvou základních školách a dvou osmiletých gymnáziích v hodinách matematiky. Cílem testování bylo zjistit, zda jsou webové stránky plně funkční, zda je jejich ovládání pro žáky i učitele srozumitelné a intuitivní.

6.1 Gymnázium Strakonice

Ověření webových stránek probíhalo na osmiletém gymnáziu ve Strakonici prostřednictvím učitelky matematiky Mgr. Petry Babkové ve dvou vyučovacích hodinách matematiky, konkrétně v sekundě. Paní učitelka chtěla aplikaci využít k celkovému opakování učiva sedmého ročníku, tj. procenta, zlomky, poměr, přímá a nepřímá úměrnost, obsah a obvod obrazců, objem a povrch hranolů. Vzhledem k časovému rozmezí, které mohlo být na vyzkoušení uvolněno, paní učitelka nakonec vybrala dvě témata – procenta, obvody a obsahy – každé na jednu vyučovací hodinu.

Paní učitelka zajistila na obě potřebné hodiny počítačovou učebnu. Kvůli vyššímu počtu žáků než je počítačů v učebně, však seděli žáci po dvojicích. V úvodu hodiny byli žáci nejprve seznámeni s cílem hodiny a tématem, které si na webových stránkách mají zvolit. Žákům bylo umožněno, aby si zvolili, zda si před zahájením práce chtějí vytvořit svůj uživatelský účet nebo přímo počítat. Z celkového počtu třinácti dvojic si svůj účet vytvořili čtyři. Pro první vyučovací hodinu byla vybrána kategorie *Procenta* a pro druhou hodinu kategorie *Obsahy a obvody obrazců*. Paní učitelka při hodinách žáky obcházela a pozorovala, jak se jim daří správně počítat zadané úlohy různé obtížnosti. Žáci měli určen počet příkladů a obtížnost (*střední*), kterou mají počítat.

6.2 Gymnázium J. V. Jirsíka v Českých Budějovicích

Nasazení aplikace na Gymnáziu J. V. Jirsíka v Českých Budějovicích proběhlo ve spolupráci s učitelem matematiky Mgr. Petrem Janáčkem ve výuce matematiky třídy tercie. Testování probíhalo v rámci jedné vyučovací hodiny a přítomno bylo 27 žáků. Pan učitel umožnil žákům, aby si sami vybrali téma, které probírali v letošním roce

a je mezi nabízenými patnácti kategoriemi. Žáci zvolili kategorii *Povrchy a objemy těles*.

Vyučovací hodina neprobíhala v počítačové učebně, ale v klasické učebně a pro zobrazení aplikace byla využita interaktivní tabule. Žáci si k počítání sami zvolili nejtěžší obtížnost. Všichni žáci pracovali samostatně v lavicích na jednom stejném zadání příkladu, poté pan učitel vybral jednoho žáka, aby šel k tabuli příklad vyřešit. Před potvrzením odpovědi měli žáci říkat, zda se spolužákovým výsledkem souhlasí či nikoli, a až poté teprve vybraný žák dal svou odpověď zkontrolovat. V případě neúspěchu měl žák popsat, jakým způsobem postupoval, a učitel ho upozornil na chybu, které se ve výpočtu dopustil.

6.3 Základní škola 9.května v Sezimově Ústí

Testování aplikace probíhalo na Základní škole 9. května v Sezimově Ústí v hodinách matematiky třídy 6. A za vedení paní učitelky Mgr. Andreji Smetanové. K vyzkoušení aplikace byla vyhrazena jedna vyučovací hodina, které se účastnilo 18 žáků. Paní učitelka z nabízených kategorií zvolila *Dělitelnost přirozených čísel*, kterou probírali v nedávné době a chtěla ji takto zopakovat.

Hodina probíhala za vedení paní učitelky, která stránky promítala žákům přes dataprojektor. Při navolené obtížnosti *střední* počítali žáci zobrazené příklady samostatně, paní učitelce nahlašovali závěry, ke kterým došli, a paní učitelka zadala do systému nejčastěji se opakující řešení. Příklady z obtížnosti *těžká* již počítali všichni dohromady za pomoci paní učitelky, která je usměrňovala ke správnému řešení.

6.4 Základní škola a mateřská škola Domažlice

Paní učitelka Mgr. Hana Bradáčová umožnila ověření aplikace na Základní škole a mateřské škole v Domažlicích, jež proběhlo ve dvou třídách – 6. B a 7. A. V obou třídách proběhlo testování v rámci jedné vyučovací hodiny. V šesté třídě se ho účastnilo 20 žáků a v sedmé 24 žáků. V šesté třídě počítali příklady z kategorie *Dělitelnost přirozených čísel* a v sedmé třídě vybrali kategorii *Poměr, přímá a nepřímá úměrnost*.

Ve třídě 6. B paní učitelka zorganizovala hodinu tak, že žákům zobrazovala aplikace přes dataprojektor. Paní učitelka vždy vybrala jednoho žáka, který měl nahlas říkat, jak by daný příklad řešil. Poté paní učitelka zadala výsledek do systému a odeslala ke kontrole. Třída počítala většinu příkladů z obtížnosti *lehká* a v závěru hodiny několik příkladů z obtížnosti *střední*.

V sedmém ročníku použila paní učitelka pro zobrazení webové aplikace opět dataprojektor. Žáci pracovali samostatně v lavicích a do systému zapisoval výsledek ten žák, který daný příklad vypočítal jako první. Paní učitelka nejprve v úvodu hodiny zvolila několik příkladů z lehké obtížnosti, aby si žáci učivo připomenuli, poté pokračovali v počítání příkladů z obtížnosti *střední*.

6.5 Shrnutí

Při testování nebyly odhaleny žádné závažné nedostatky, reakce systému byla vždy odpovídající a téměř okamžitá.

Všichni účastníci hodnotili testování aplikace vesměs stejně – použití stránek v hodině bylo zajímavé a netradiční, někteří žáci se snažili i více počítat, v systému se rychle zorientovali a nenastal žádný závažný problém. Učitelé měli několik podnětných připomínek, které byly přínosem v mé další práci. Například vyučující kladně hodnotili rozdělení příkladů do kategorií a následně ještě do různých úrovní obtížnosti, jenž umožňuje využití webových stránek v různých fázích výkladu. Naopak někteří z nich jako nevýhodu uvedli nutnost použití počítačů, což není ve výuce vždy možné, a proto vidí větší potenciál této aplikace ve využívání pro domácí přípravu žáků či pro zadávání domácích úkolů. Někteří žáci uvedli, že se na stránky chtějí doma znovu podívat a vyzkoušet si i jiné kategorie.

7 Závěr

Cílem této práce bylo vytvořit interaktivní webové stránky pro procvičování a testování žáků z matematiky na úrovni základní školy z vybraných partií matematiky. Záměrem bylo vytvořit stránky, které žáky zaujmou, budou jednoduché na ovládání a budou obsahovat veškeré informace potřebné k jejich používání. Domnívám se, že tohoto cíle bylo zdárně dosaženo. Webové stránky jsou dostupné na internetové adrese www.vypoctito.cz.

Součástí práce bylo také pilotní ověření na několika základních školách, respektive víceletých gymnáziích. Z tohoto důvodu obsahovaly stránky již v této fázi stovky příkladů rozdělených do příslušných kategorií a obtížností. Příklady, které jsou převzaty z literatury, jsou řádně citovány kvůli zachování autorských práv. Obrázky vložené u některých zadání byly vytvořeny v programu GeoGebra či v Cabri 3D v2.

Aplikace je určena pro žáky a jejich učitele, lze však předpokládat, že jejími primárními uživateli budou žáci, kteří ji mohou používat například pro přípravu na přijímací zkoušky na střední školu. Relevantnost a správnost vkládaných dat učiteli bude i nadále kontrolována administrátorem.

Při tvorbě aplikace nebylo opomenuto ani zajištění jejího zabezpečení proti různým útokům (ať jednotlivých účtů uživatelů, tak celé struktury webu). Dále struktura webu byla navržena tak, aby bylo možné její další rozšíření. Námětem pro další rozšíření webových stránek *VypočtiTo* by mohlo být například navýšení z patnácti kategorií o další oblasti matematiky (např. různé matematické hlavolamy), přidáním podkategorií do jednotlivých témat či propojení s komplementárními projekty.

Použitá literatura a zdroje

- [1] KANISOVÁ, Hana a Miroslav MÜLLER. *UML srozumitelně*. 2. aktualiz. vyd. Brno, 2006, 176 s. ISBN 80-251-1083-4.
- [2] SCHMULLER, Joseph. *Myslíme v jazyku UML*. Vyd. 1. Grada, 2001, 360 s. ISBN 80-247-0029-8.
- [3] ARLOW, Jim a Ila NEUSTADT. *UML 2 a unifikovaný proces vývoje aplikací: objektově orientovaná analýza a návrh prakticky*. Vyd. 1. Překlad Bogdan Kiszka. Brno: Computer Press, 2011, 567 s. ISBN 978-80-251-1503-9.
- [4] JONES, Meilir. *Základy objektově orientovaného návrhu v UML*. Vyd. 1. Praha: Grada, 2001, 367 s. ISBN 80-247-0210-X.
- [5] BEACH, Matt. Wire Frame Your Site Article. *Sitepoint* [online]. © 2000 – 2014 [cit. 2014-04-22]. Dostupné z: <http://www.sitepoint.com/wire-frame-your-site/>
- [6] NOVOTNÝ, Pavel. Jak vytvořit správný wireframe. *Web consultants* [online]. © 2008 - 2013 [cit. 2014-04-22]. Dostupné z: <http://www.web-consultants.cz/jak-vytvorit-spravny-wireframe/>
- [7] GILMORE, W. *Velká kniha PHP 5 a MySQL: kompendium znalostí pro začátečníky i profesionály*. Nové, 3. vyd. Překlad Jan Pokorný. Brno: Zoner Press, 2011, 736 s. Encyklopedie Zoner Press. ISBN 978-80-7413-163-9.
- [8] LECKY-THOMPSON, Ed a Steven D NOWICKI. *PHP 6: programujeme profesionálně*. Vyd. 1. Překlad Ondřej Gibl. Brno: Computer Press, 2010, 718 s. Programujeme profesionálně. ISBN 978-80-251-3127-5.
- [9] ZAKAS, Nicholas C. *JavaScript pro webové vývojáře: programujeme profesionálně*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2009, 832 s. ISBN 978-80-251-2509-0.
- [10] MEYER, Eric A. *Eric Meyer o CSS: kompletní průvodce*. Vyd. 1. Překlad Jan Pokorný. Brno: Zoner Press, 2007, 560 s. ISBN 978-80-86815-64-0.

- [11] *MathJax: Beautiful math in all browsers* [online]. © 2011 [cit. 2014-04-23]. Dostupné z: <http://www.mathjax.org/>
- [12] HTML4 and HTML5 Tutorial. *W3Schools* [online]. 1999-2014 [cit. 2014-05-03]. Dostupné z: <http://www.w3schools.com/html/default.asp>
- [13] Ukládání hesel. VRÁNA, Jakub. *PHP triky* [online]. © 2005-2014 [cit. 2014-04-30]. Dostupné z: <http://php.vrana.cz/ukladani-hesel.php>
- [14] *Crypto-World: Informační sešit GCUCMP* [online]. 2013 [cit. 2014-04-30]. Dostupné z: http://crypto-world.info/casop15/crypto0910_13.pdf
- [15] .htaccess Tutorial: Part 2 - .htaccess Commands. *Free Webmaster Help* [online]. © 1999 - 2001 [cit. 2014-05-02]. Dostupné z: <http://www.freewebmasterhelp.com/tutorials/htaccess/2>
- [16] .htaccess Tutorial: Part 1 - Introduction. *Free Webmaster Help* [online]. © 1999 - 2001 [cit. 2014-05-02]. Dostupné z: <http://www.freewebmasterhelp.com/tutorials/htaccess>
- [17] Zabezpečení serveru Apache a PHP. *Security-Portal.cz* [online]. [cit. 2014-05-03]. Dostupné z: <http://www.security-portal.cz/clanky/zabezpe%C4%8Den%C3%AD-serveru-apache-php>
- [18] PHP - register_globals. *Wedos* [online]. 2014 [cit. 2014-05-03]. Dostupné z: <http://kb.wedos.com/php/register-globals.html>
- [19] Bezpečnost webových stránek. JAHODA, Bohumil. *Je čas* [online]. 2013 [cit. 2014-05-03]. Dostupné z: <http://jecas.cz/bezpecnost>
- [20] Testování a tvorba testovatelného kódu v PHP. *Zdroják.cz* [online]. [cit. 2014-05-03]. Dostupné z: <http://www.zdrojak.cz/clanky/testovani-a-tvorba-testovatelného-kódu-v-php/>
- [21] CHACON, Scott. *Pro Git* [online]. Praha: CZ.NIC, c2009, 263 s. [cit. 2014-05-13]. Edice CZ.NIC. ISBN 978-80-904248-1-4. Dostupné z: http://knihy.nic.cz/files/nic/edice/scott_chacon_pro_git.pdf

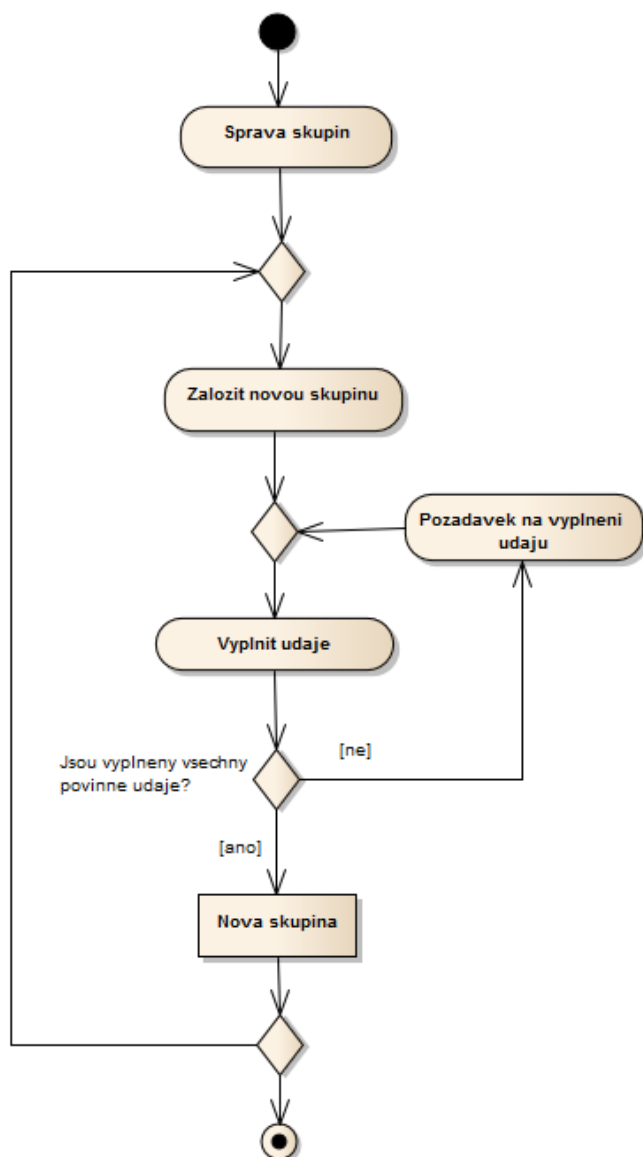
Seznam obrázků

1	Úvodní strana webových stránek www.onlinecviceni.cz	17
2	Úvodní strana webových stránek www.matematika.hrou.cz	20
3	Úvodní strana webových stránek www.pocitejme.cz	22
4	Úvodní strana webových stránek www.procvicuj.cz	24
5	Úvodní strana webové stránky www.1quiz.jirpa.cz/index.php	26
6	Abstraktní pohled na případy užití systému.	37
7	Diagram případu užití z pohledu nepřihlášeného uživatele.	38
8	Diagram případu užití z pohledu žáka.	39
9	Diagram případů užití z pohledu učitele.	40
10	Diagram případu užití z pohledu administrátora.	41
11	Diagram aktivit - registrace uživatele.	44
12	Doménový model aplikace.	46
13	Přehled tabulek v databázi.	48
14	Práva uživatele a jeho přiřazení do skupiny.	52
15	Procvičování příkladů.	53
16	Návrhová šablona Smarty (<code>welcome.tpl</code>). [7, str. 390]	63
17	Aplikační logika šablony <code>index.tpl</code> (<code>index.php</code>). [7, str. 390]	63
18	Zobrazení vzorce pro binomické rozdělení pomocí MathJaxu. (zdroj: http://www.mathjax.org/demos/tex-samples/)	66
19	Diagram aktivit - vytvoření nové skupiny z pohledu učitele.	78
20	Diagram aktivit - procvičování příkladů z pohledu žáka.	80
21	Analytický model diagramu tříd aplikace.	81
22	Drátový model titulní strany webových stránek.	82
23	Drátový model webu po přihlášení žáka.	83
24	Drátový model webu po přihlášení učitele.	84

Seznam tabulek

1	Porovnání základních funkcí analyzovaných webových stránek pro on-line procvičování matematiky.	79
---	---	----

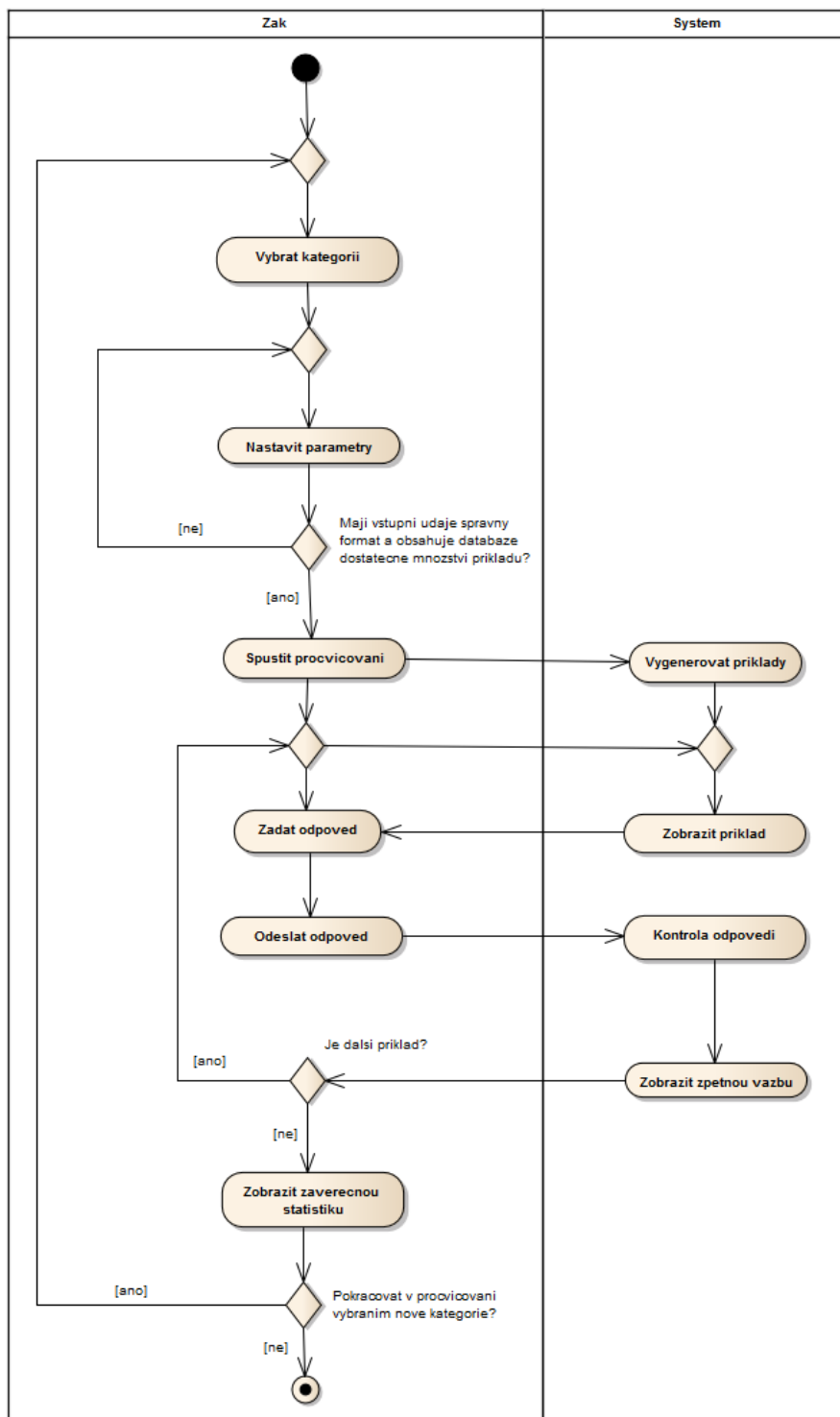
A Příloha



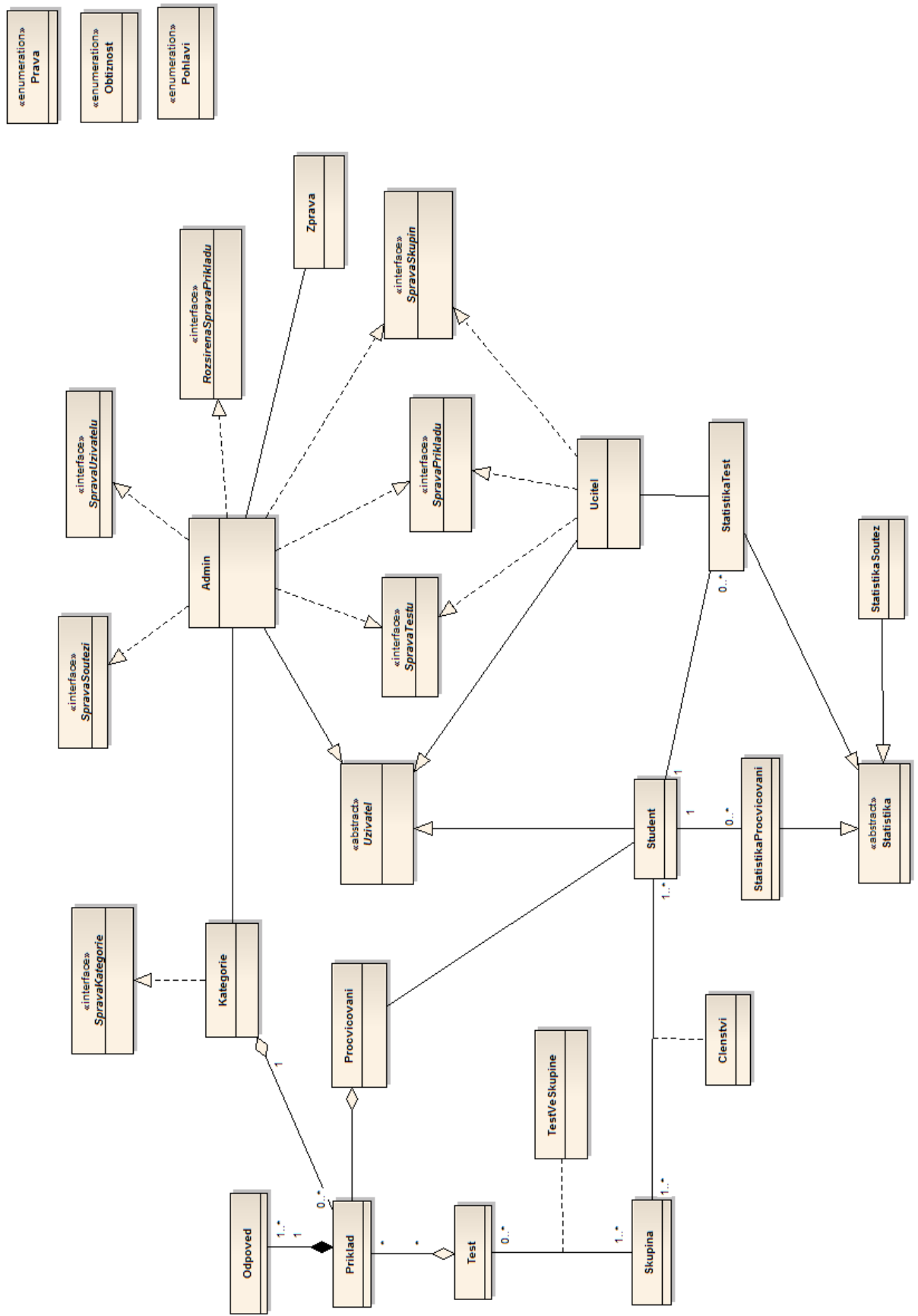
Obrázek 19: Diagram aktivit - vytvoření nové skupiny z pohledu učitele.

	Online cvičení	Matematika hrou	Pocitejme.cz	Testy a kvízy on-line	Procvičuj.cz
Registrace	ano	ano	ne	ano	ano
Rozlišování rolí	ne	ano	ne	ne	ano
Skupiny	ano	ano	ne	ne	ne
Soutěže	ano	ne	ne	ne	ne
Souhrnná statistika	ano	ano	ne	ano	ano
Statistika jednotlivých testů	ano	ano	ano	ano	ano
Vkládání příkladů	ne	ne	ne	ne	ne

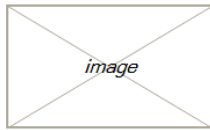
Tabulka 1: Porovnání základních funkcí analyzovaných webových stránek pro online procvičování matematiky.



Obrázek 20: Diagram aktivit - procvičování příkladů z pohledu žáka.



Obrázek 21: Analytický model diagramu tříd aplikace.

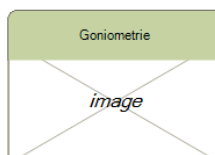
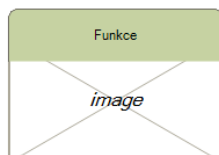
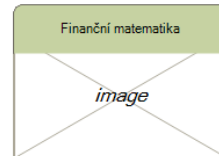
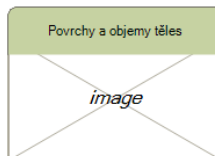
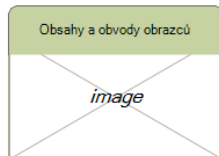
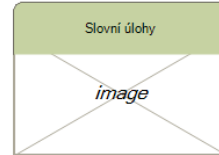
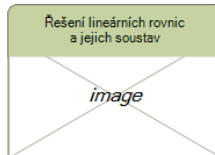
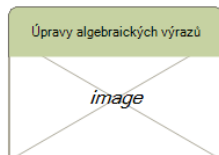
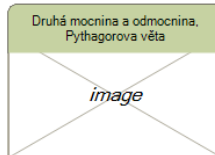
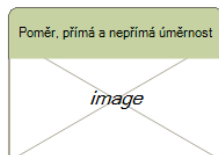
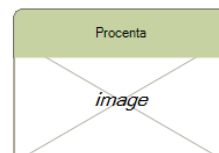


VypočtiTo

Jméno: Heslo:

[Registrace](#)

[Domů](#) | [O projektu](#) | [FAQ](#)



O nás

Lorem ipsum dolor sit amet, consectetur adipiscing elit.

Morbi vitae nunc eget orci hendrerit tempor. Vivamus erat massa, euismod sit amet, aliquet eget, dictum a, felis.

Curabitur diam massa.

Servisní odkazy

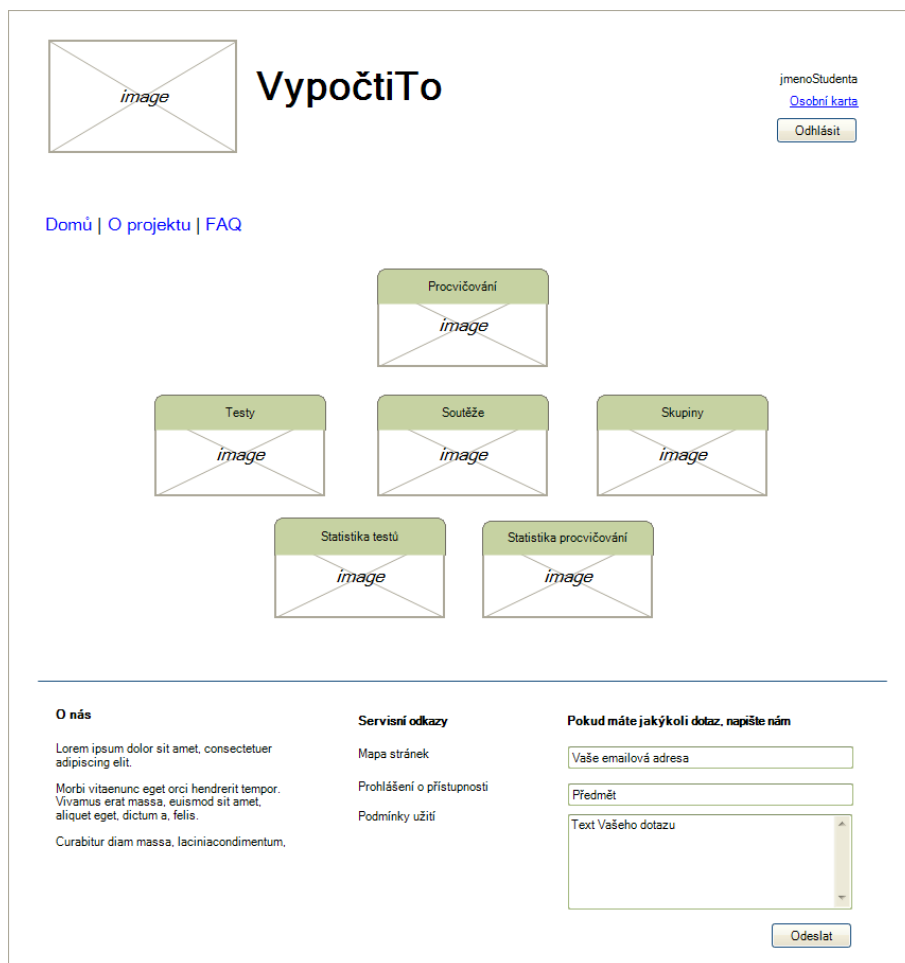
[Mapa stránek](#)

[Prohlášení o přístupnosti](#)

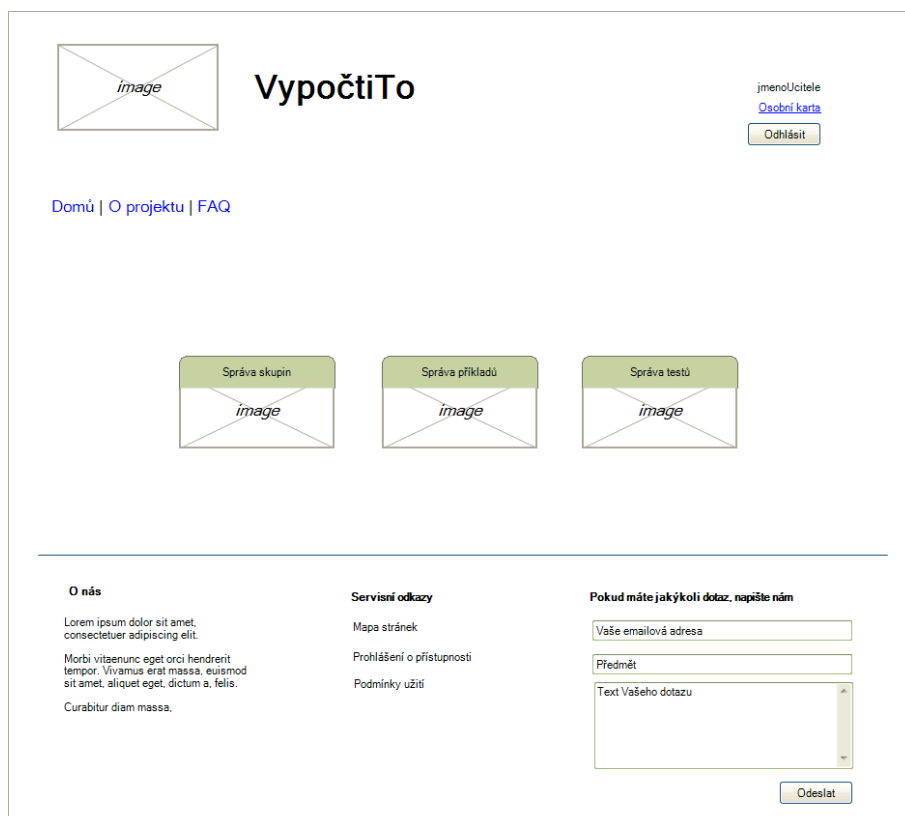
[Podmínky užití](#)

Pokud máte jakýkoli dotaz, napište nám

Obrázek 22: Drátový model titulní strany webových stránek.



Obrázek 23: Drátový model webu po přihlášení žáka.



Obrázek 24: Drátový model webu po přihlášení učitele.