

Bakalářská práce



České  
vysoké  
učení technické  
v Praze

**F3**

Fakulta elektrotechnická  
Katedra počítačové grafiky a interakce

# Rozšíření systému XML Check

**Martin Šubrt**

Softwarové technologie a management, Web & multimedia

Květen 2017

Vedoucí práce: Doc. RNDr. Irena Holubová, Ph.D.



České vysoké učení technické v Praze  
Fakulta elektrotechnická

katedra počítačové grafiky a interakce

## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Martin Šubrt**

Studijní program: Softwarové technologie a management

Obor: Web a multimedia

Název tématu: **Rozšíření systému XML Check**

Pokyny pro vypracování:

Cílem práce je modifikace a rozšíření systému XML Check, který byl implementován v rámci Bakalářské práce J. Konopáska a rozšířen v rámci Bakalářské práce P. Hudečka a je intenzivně využíván v rámci předmětu Technologie XML na MFF UK i FEL ČVUT. Úkolem autora je seznámit se podrobně se systémem a rozšířit ho pro využití v předmětu Databázové systémy i obecně. Modifikace a rozšíření budou zahrnovat:

- Plug-iny pro odevzdávání a kontrolu domácích úkolů předmětu Databázové systémy.
- Podporu možnosti zadávat příklady k procvičení a jejich správná i špatná řešení.
- Opravy chyb v poslední používané verzi systému.

Systém bude otestován v rámci odpovídajících cvičení a/nebo s využitím historických domácích úkolů.


Seznam odborné literatury:

Konopásek, J.: Systém pro správu úkolů a testů. MFF UK 2011.  
<http://www.ksi.mff.cuni.cz/~holubova/bp/Konopasek.pdf>

Hudeček, P.: Rozšíření systému XML Check. MFF UK 2015.  
<http://www.ksi.mff.cuni.cz/~holubova/bp/Hudecek.pdf>

Vedoucí: doc. Irena Holubová RNDr., Ph.D.

Platnost zadání: do konce zimního semestru 2017/2018

  
prof. Ing. Jiří Žára, CSc.  
vedoucí katedry



  
prof. Ing. Pavel Ripka, CSc.  
děkan

V Praze dne 29. 2. 2016



## Poděkování / Prohlášení

Děkuji všem, kteří mě podporovali při psaní této práce. Zejména vedoucí práce Doc. RNDr. Ireně Holubové, Ph.D. za odbornou pomoc, kterou této práci věnovala, a za její báječný přístup, se kterým ke mně a celé práci přistupovala. Děkuji také rodině a přátelům za podporu psychickou, gramatickou a stylistickou.

Prohlašuji, že jsem tuto bakalářskou práci vypracoval sám s přispěním vedoucí práce a konzultanta a že jsem v práci uvedl veškeré použité informační zdroje.

Dále prohlašuji, že nemám námitek proti půjčování nebo zveřejňování mé bakalářské práce nebo její části se souhlasem katedry Počítačové grafiky a interakce na FEL.

V Praze dne 26. 5. 2017

.....

## Abstrakt / Abstract

Tato práce se zabývá rozšířením systému na odevzdávání domácích úkolů XML Check vytvořeným v roce 2011 a rozšířeným v roce 2015. Systém je používán na MFF UK a FEL ČVUT v předmětu Technologie XML. V roce 2015 vznikl nápad systém rozšířit pro předmět Databázové systémy. To je jedno z hlavních zaměření práce. Druhé hlavní zaměření je přidání nové funkcionality a to možnosti přidání příkladů k procvičení pro cvičící a jejich zobrazení pro studenty. Je opravena chyba s jazykovým nastavením systému. Pokud uživatel neměnil jazykovou verzi, tak webový prohlížeč s preferencí českého jazyka zobrazí českou verzi, jinak anglickou.

**Klíčová slova:** odevzdávání domácích úkolů, kontrola domácích úkolů, příklady k procvičení

This thesis is about improvement of the 2011 homework manager improved in 2015 called XML Check. The homework manager is used at MFF CU and FEE CTU in XML Technologies course. An idea of improving the system was born in 2015. Extension of the system homeworks support for Database systems course is one of the main goals of this thesis. The other main goal is adding samples for practise to this system. Tutors are able to manage these samples and their solutions, correct or incorrect. Language settings are fixed now and the system shows correct language version. If the web browser preference is set to the czech language the czech version is shown, english otherwise.

**Keywords:** homeworks submission, homeworks check, samples for practise

**Title translation:** Bachelor thesis – XML Check extension

# Obsah /

<b>1 Úvod</b> .....	1
<b>2 Základní informace</b> .....	3
2.1 Používané pojmy .....	3
2.2 Předmět DBS .....	3
2.3 Domácí úkoly v předmětu DBS .....	4
2.3.1 Úkol 1: konceptuální model v ER a UML.....	4
2.3.2 Úkol 2: transformace ER modelu do relační- ho modelu .....	5
2.3.3 Úkol 3: jazyk SQL – vytváření a aktualizace dat .....	5
2.3.4 Úkol 4: jazyk SQL – dotazování a pohledy .....	5
2.3.5 Úkol 5: relační algebra a relační kalkuly .....	5
2.3.6 Úkol 6: implementace jednoduché databázové aplikace.....	5
<b>3 Analýza</b> .....	6
3.1 Systémy pro odevzdávání domácích úkolů .....	6
3.1.1 Zaměření analýzy .....	6
3.1.2 Course Ware .....	6
3.1.3 Moodle .....	7
3.1.4 Squeezer .....	9
3.1.5 Baryk.....	10
3.1.6 ProgTest .....	11
3.1.7 Závěr analýzy .....	12
3.2 Systém XML Check .....	12
3.3 Analýza rozšiřované části systému XML Check.....	12
3.3.1 Účel aplikace .....	12
3.3.2 Zainteresoované osoby a instituce (stakeholders) ..	13
3.3.3 Uživatelé .....	13
3.3.4 Požadavky na funkcion- alitu a klíčové potřeby uživatelů.....	13
3.3.5 Funkční požadavky .....	13
3.3.6 Případy užití .....	14
3.3.7 Relační model .....	16
<b>4 Vlastní práce</b> .....	18
4.1 Rozšíření pro předmět DBS ...	18
4.1.1 Rozšířit systém XML Check pro další předmět obecně ....	18
4.1.2 Rozšířit systém XML Check pro předmět DBS .....	19
4.2 Přidání příkladů k procvičení .	20
4.2.1 Rozšíření databáze.....	20
4.2.2 Rozšíření uživatelského rozhraní .....	22
4.3 Oprava chyb.....	25
4.4 Testování .....	26
<b>5 Závěr</b> .....	27
<b>Literatura</b> .....	28
<b>A Zkratky a pojmy</b> .....	29
<b>B Obsah příloženého CD</b> .....	30

## / Obrázky

<b>3.1.</b>	Systém Course Ware .....	7
<b>3.2.</b>	Systém Moodle .....	8
<b>3.3.</b>	Systém Squeezer .....	9
<b>3.4.</b>	Systém Baryk .....	10
<b>3.5.</b>	Systém ProgTest .....	11
<b>3.6.</b>	Uživatelé .....	14
<b>3.7.</b>	Odevzdávání domácích úkolů ..	15
<b>3.8.</b>	Správa příkladů k procvičení ..	16
<b>3.9.</b>	Část realčního modelu po rozšíření .....	16
<b>4.1.</b>	Tabulka Subjects .....	18
<b>4.2.</b>	Tabulka přednášek v uživa- telském rozhraní .....	19
<b>4.3.</b>	Formulář pro úpravu před- nášky v uživatelském rozhraní .	19
<b>4.4.</b>	Tabulka Samples .....	21
<b>4.5.</b>	Tabulka Solutions .....	22
<b>4.6.</b>	Tabulka příkladů k procvi- čení v uživatelském rozhraní z pohledu cvičícího .....	23
<b>4.7.</b>	Tabulka řešení příkladů k procvičení v uživatelském rozhraní z pohledu cvičícího ...	23
<b>4.8.</b>	Tabulka příkladů k procvi- čení v uživatelském rozhraní z pohledu studenta .....	24
<b>4.9.</b>	Tabulka řešení příklad k pro- cvičení v uživatelském roz- hraní z pohledu studenta .....	24
<b>4.10.</b>	Detail příkladu k procviče- ní v uživatelském rozhraní z pohledu studenta .....	25
<b>4.11.</b>	Detail řešení příkladu k pro- cvičení v uživatelském roz- hraní z pohledu studenta .....	25



# Kapitola 1

## Úvod

Podobně jako na ostatních vysokých se i na Fakultě elektrotechnické Českého vysokého učení technického používá několik systémů pro odevzdávání domácích úkolů. Nejčastěji fungují na principu webového rozhraní. Studentům umožňují odesílat řešení těchto domácích úkolů. Učitelům pak umožňují tato řešení buďto přímo ve webovém rozhraní nebo po stažení do počítače kontrolovat. Existují systémy, které nabízejí nejen automatickou kontrolu studentova řešení, ale i přidělování bodů studentovi na základě této kontroly. Avšak mnoho těchto systémů přidělování bodů studentovi neumožňuje. Často umožňuje studentům odevzdání řešení domácího úkolu a následně poskytnutí zpětné vazby. Například seznam chyb v odevzdaném řešení.

Mezi systémy, které poskytují kontrolu řešení domácích úkolů a podání zpětné vazby studentovi, patří i systém XML Check. Tento systém například kontroluje splnění domácího úkolu, tedy správnost syntaxe jazyka, v jakém je dané řešení psáno, a přítomnost povinných částí domácího úkolu. Studentům nejsou přidělovány body systémem, ale řešení musí zkontrolovat a ohodnotit cvičící. Systém byl doposud využíván pro účely předmětu *Technologie XML*, který je vyučován na Matematicko-fyzikální fakultě Univerzity Karlovy<sup>1</sup> a na Fakultě elektrotechnické Českého vysokého učení technického<sup>2</sup>.

XML Check byl vytvořen v roce 2011 Janem Konopáskem[1] a nepřetržitě používán do roku 2015. Během této doby bylo nalezeno několik chyb, které způsobovaly problémy s ukládáním řešení úkolů do databáze, studenti tak někdy museli řešení cvičícímu poslat e-mailem. Systém také občas nepřijímal správná řešení. Tyto a další chyby byly v roce 2015 opraveny a systém byl upraven a rozšířen Petrem Hudečkem[2]. Avšak i přes tuto úpravu vznikly další nápady na rozšíření systému jak ze strany učitelů tak ze strany studentů.

Jedním z nich je přidání příkladů k procvičení. Učitel může příklad přidat, upravit nebo smazat. Nyní se studenti měli možnost učit z příkladů v přednáškách a ukázaných na přednáškách. Nad takovými příklady je těžké udržovat kontrolu a přehled, jaká témata jsou v příkladech rozebírána a na jaká je třeba ještě studenty upozornit, aby si dávali pozor.

Tato možnost bude v budoucnu využita při výuce předmětu *Technologie XML*.

Dalším požadavkem bylo rozšířit systém pro další předmět, a to *Databázové systémy*. V tomto předmětu se studenti seznamují s problematikou databází. U relačních databází se využívá standardizovaný strukturovaný dotazovací jazyk SQL.

Kontroly řešení domácích úkolů budou probíhat podobně jako v předmětu *Technologie XML*. Bude tedy probíhat kontrola syntaxe jazyka SQL a dále kontrola přítomnosti všech požadavků daného úkolu. Například má-li v domácím úkolu student vytvořit tabulku databáze, proběhne kontrola přítomnosti klíčového slova `CREATE TABLE`. V systému XML Check nyní není přítomna kontrola syntaxe jazyka SQL, bude přidána v průběhu této práce.

<sup>1</sup> <https://www.mff.cuni.cz/>

<sup>2</sup> <http://www.fel.cvut.cz/>

V první kapitole této práce vysvětlujeme často používané pojmy, v krátkosti uvádíme historii předmětu *Databázové systémy*, zkoumáme znění domácích úkolů předmětu.

V druhé kapitole analyzujeme podobné odevzdávací systémy na domácí úkoly využívané na Fakultě elektrotechnické Českého vysokého učení technického, popisujeme systém XML Check, vypisujeme body jeho rozšíření. Definujeme účel aplikace, zainteresované osoby a instituce, uživatele systému, požadavky na funkcionalitu, funkční požadavky, případy užití a část relačního modelu rozšiřované části systému.

Ve třetí kapitole popisujeme samotné rozšíření systému, pro další předmět obecně a konkrétně, přidání příkladů k procvičení a testování jednotlivých částí rozšíření.

V poslední kapitole zpětně hodnotíme práci a její výsledek, také navrhujeme možné další změny a vylepšení systému.

# Kapitola 2

## Základní informace

Nejprve definujeme nejdůležitější pojmy, které jsou v následujícím textu použity.

V této kapitole též popisujeme předmět DBS, který se učí na Fakultě elektrotechnické Českého vysokého učení technického a podobu všech domácích úkolů v něm zadávaných.

### 2.1 Používané pojmy

Následující pojmy jsou použity v textu této práce<sup>1</sup>.

- **Matematicko-fyzikální fakulta Univerzity Karlovy v Praze (MFF)** je fakulta, na které se vyučuje předmět *Technologie XML*.
- **Předmět *Technologie XML (Technologie XML)*** je předmět vyučovaný na MFF a FEL. Pro správu domácích úkolů v *Technologiích XML* je využíván systém XML Check.
- **Předmět *Databázové systémy (DBS)*** je předmět vyučovaný na FEL.

### 2.2 Předmět DBS

V této sekci popisujeme předmět DBS vyučovaný na FEL. Popisujeme jeho podobu v době sepisování této práce (tj. v akademickém roce 2016/2017).

Předmět DBS byl na FEL do roku 2015 vedený a samostatně vyučovaný vedoucí práce Doc. RNDr. Irenou Holubovou, Ph.D. Od roku 2015 forma předmětu nebyla zásadně změněna.

Předmět je dotován 2+2 vyučovacími hodinami, tzn. že každý týden je odpřednášena 90 minut dlouhá přednáška a odevčeno 90 minutové cvičení. Předmět je ukončen zápočtem a zkouškou. K udělení zápočtu musí student získat určitý počet bodů během semestru, poté je připuštěn ke zkoušce.

Na přednáškách jsou probírány principy a postupy týkající se tvorby a správy DBS. Vyučovány jsou SQL databáze. V přednáškách je také prezentována správná syntaxe tohoto jazyka a forma SQL dotazů.

V průběhu cvičení jsou rekapitulovány nejdůležitější problémy probrané na poslední přednášce a problémy týkající se obsahu cvičení. Studenti následně vypracovávají sérii malých úkolů ohledně dané problematiky. Řešení úkolů mohou studenti odesílat cvičící na e-mail, pokud tak stihnou v časovém limitu. Za správné řešení student získává jeden bod.

Každé druhé cvičení je zadán úkol domácí, na jehož vypracování studenti mají dva týdny. Těchto domácích úkolů je zadáno šest během celého semestru. Zadání domácích úkolů se nijak zásadně neměnilo napříč historií předmětu DBS. Kontroluje se originalita těchto domácích úkolů.

---

<sup>1</sup> Další použité zkratky a pojmy jsou vysvětleny zde A

Za každý domácí úkol může student získat až 20 bodů. V semestru student může celkem získat 120 bodů za domácí úkoly a přibližně 30 bodů za malé úkoly vypracované na cvičeních a aktivitě během cvičení. Při dosažení alespoň 80 bodů, je studentovi přiznán zápočet. Body získané nad hranici zápočtu 80 bodů se přičítají k výsledku písemného testu u zkoušky.

Na konci semestru se koná písemná zkouška, na které je účast povinná. Z písemného testu je možné získat 100 bodů. Pokud student získá méně než 30 bodů z písemného testu zkoušky, body převedené ze semestru se mu nulují a k dalším pokusům zkoušky je již nemá k dispozici.

Bodové hranice jednotlivých známek:

- alespoň 90 bodů ... A
- alespoň 80 bodů ... B
- alespoň 70 bodů ... C
- alespoň 60 bodů ... D
- alespoň 50 bodů ... E
- méně než 50 bodů ... F

Anotace předmětu<sup>1</sup> zní:

„Základní principy DBS, architektura SŘBD, funkce jednotlivých komponent. Konceptuální, databázová a fyzická úroveň pohledu na data. Přehled databázových modelů - síťový, relační, objektově relační a objektový. Konceptuální datový model. Základní konstrukty, vyjádření integritních omezení. Schéma. Relační model dat, relace, atributy, domény, schéma relační databáze. Kvalita schématu. Vyjádření integritních omezení formou funkčních závislostí. Normální formy relací. Databázové dotazové jazyky. Relační algebra, relační kalkul, QBE, SQL. Návrh relačního schématu. Normalizace schématu formou dekompozice. Kritéria kvality dekompozice. Návrh schématu relační databáze přímou transformací z konceptuálního schématu. Transakce, zotavení z chyb, koordinace paralelního přístupu, ochrana dat. Fyzický model dat, hromada, index, cluster indexovaný, cluster hashovaný.“

## 2.3 Domácí úkoly v předmětu DBS

Domácí úkoly na sebe navazují. Student vypracovává všechny úkoly na téma, které si zvolil na začátku semestru. Pokud v pozdějších úkolech studentovi chybí v konceptu nějaké požadavky, musí je student doplnit ve všech předchazejících úkolech.

### 2.3.1 Úkol 1: konceptuální model v ER a UML

V prvním úkolu má student za úkol vybrat si téma, na které vytvoří konceptuální model buď v ER (Entity relationship) nebo v UML (Unified modeling language)<sup>2</sup>. Může si vybrat jakékoliv téma z běžného života, například autoopravnu, kde bude mít entity jako zaměstnanec (, směna), automobil, klient, ... Může zvolit téma letiště, restaurace. Toto téma pak student má po celý semestr a úkoly další navazují na tento první.

Tento model se nejvíce podobá realitě.

Odevzdávaný úkol je ve formě PDF nebo obrázku PNG, JPG, atp.

<sup>1</sup> Syllabus na FEL <https://www.fel.cvut.cz/cz/education/bk/predmety/13/92/p1392506.html/>

<sup>2</sup> <http://www.uml.org/what-is-uml.htm/>

### ■ 2.3.2 Úkol 2: transformace ER modelu do relačního modelu

Při navrhování databáze, je nutné konceptuální model převést do relačního modelu. Zde se přidávají další tabulky, když je to potřeba. Pokud máme vztah M:N, musíme vytvořit další tabulku, která nám zajistí spojení těchto dvou entit.

Stejně jako první úkol se tento odevzdává ve formách PDF, PNG, JPG, atp.

### ■ 2.3.3 Úkol 3: jazyk SQL – vytváření a aktualizace dat

Na základě relačního modelu z druhého úkolu lze vytvořit databázi. Nyní ji má student za úkol vytvořit pomocí SQL a vytvořit 3 integritní omezení (z nich alespoň jedno typu CHECK). Dále má student naplnit daty, každou tabulku alespoň o 5 hodnotách.

Odevzdávaný soubor je ve formě SQL.

### ■ 2.3.4 Úkol 4: jazyk SQL – dotazování a pohledy

Ve čtvrtém úkolu student pomocí SQL dotazů tvoří alespoň 5 netriviálních dotazů nad databází ze čtvrtého úkolu. Alespoň v jednom dotazu musí být alespoň jedna množinová operace UNION/INTERSECT/EXCEPT, třídění ORDER BY, agregaci GROUP BY, HAVING, vnořený poddotaz, přirozené, vnitřní a vnější spojení. Nakonec alespoň dva pohledy CREATE VIEW.

Stejně jako v úkolu 3 se odevzdává forma SQL.

### ■ 2.3.5 Úkol 5: relační algebra a relační kalkuly

Nad relačním schématem z druhého úkolu má student za úkol vytvořit alespoň 3 dotazy v relační algebře, 3 dotazy v doménovém kalkulu a 3 dotazy v n-ticovém kalkulu. Nesmí se jednat o stejné dotazy, ani o jejich přepisy (například z relační algebry do doménového kalkulu).

Odevzdávaná forma je zpravidla textový soubor.

### ■ 2.3.6 Úkol 6: implementace jednoduché databázové aplikace

Poslední šestý úkol se týká vytvoření jednoduché aplikace, která se pomocí JDBC, ODBC apod. připojí k databázi studenta. Aplikace umožňuje zobrazovat (vyhodnotit alespoň jeden dotaz, ke kterému uživatel zadá alespoň dva parametry) a editovat (přidat/smazat/modifikovat) data v alespoň jedné z tabulek databáze.

Odevzdává se forma programu, tedy všech zdrojových kódů programu. Například v archivu ZIP

# Kapitola 3

## Analýza

### 3.1 Systémy pro odevzdávání domácích úkolů

Systém XML Check<sup>1</sup> se využívá již od roku 2011. Doposud byl využíván v předmětu emTechnologie XML na MFF a FEL. V roce 2015 vznikl nápad rozšířit systém pro předmět DBS na FEL. V tomto předmětu se pro odevzdávání domácích úkolů používalo pouze odesílání řešení e-mailem cvičícímu. Jelikož se na FEL využívá několik systémů pro odevzdávání domácích úkolů, provedli jsme analýzu těchto, abychom zjistili, jestli už vhodný systém pro předmět DBS neexistuje.

#### 3.1.1 Zaměření analýzy

V rámci ČVUT funguje několik odevzdávacích systémů. Při jejich analýze jsme se zaměřili na následující témata:

- **Účel:** K jakému účelu je nyní systém používán?
- **Popis:** Jak systém funguje z uživatelského hlediska pro studenta? Jak systém funguje z uživatelského hlediska pro cvičícího nebo pro přednášejícího?
- **Kontrola:** Slouží systém pouze jako filtr odevzdávaných úkolů (např. kontrolou přípony právě odevzdávaného souboru)? Přiděluje systém studentům body plně nebo semi automaticky? Pokud ne, dokáže systém dát studentovi zpětnou vazbu?
- **Výhody a nevýhody:** Jaké výhody systém pro uživatele poskytuje s přihlédnutím k použitým technologiím a účelu k jakému se systém nyní používá? Plynou z použitých technologií či účelu systému nějaké nevýhody pro používání systému v předmětu DBS?
- **Dostupnost:** Je k dispozici zdrojový kód systému?
- **Upravitelnost:** Můžeme systém upravit? Jak náročné je upravit zdrojový kód, aby vyhovoval potřebám studentů, cvičících i přednášejících v předmětu DBS?
- **Technologie:** Jakých technologií systém využívá?
- **Vývoj a historie:** Kdy vznikla první verze systému? Jak dlouho je systém aktivně používán? Probíhá na systému stále aktivní vývoj?

#### 3.1.2 Course Ware

Systém Course Ware<sup>2</sup> je jedním z hlavních systémů pro podporu výuky na FEL. Slouží k podpoře výuky jako takové, ne pouze k odevzdávání domácích úkolů.

UploadSystem, část Course Ware, ve které se odevzdávají domácí úkoly, funguje od roku 2005.

<sup>1</sup> <http://xmlcheck.projekty.ms.mff.cuni.cz/>

<sup>2</sup> <http://cw.fel.cvut.cz/wiki/>

The screenshot shows the Course Ware interface for the course 'a7b36omo -- Objektové modelování'. On the left is a sidebar with navigation links: Hlavní stránka, Domácí úkoly, Infrastruktura, Kombinované studium, Kodex práce v kyberprostoru a plagiátorství, Přednášky, Cvičení, Test v semestru, and Zkouška. The main content area includes a breadcrumb trail 'courses:a7b36omo:start', a 'Hide Sidebar' button, and a 'Login' button. Below this is the course title 'a7b36omo -- Objektové modelování' and links for 'Diskusní fórum', 'Popis předmětu na FEL', and 'Rozvrh na FEL'. A photograph of a boat on a lake is shown with the caption 'Reka Ormo, Etiopie.'. The section 'Organizace předmětu' provides details for the winter semester 2016, listing lecturers and a list of requirements for passing the course, including a minimum score of 6 points and a deadline for assignments.

Obrázek 3.1. Systém Course Ware

- **Účel:** UploadSystem<sup>1</sup> je podsystém celku Course Ware. Je používán v desítkách předmětů každý semestr na FEL.
- **Popis:** Course Ware neslouží pouze k odevzdávání úkolů, je to nástroj pro podporu výuky obecně. Umožňuje jednoduchým způsobem vytvářet stránky předmětů. Přednášející a cvičící zveřejňují např. prezentace přednášek, cvičení, zadání domácích úkolů, kontakty apod. na těchto stránkách.  
Existují zde diskuzní fóra, kde studenti i cvičící diskutují na téma odevzdávaných úkolů.
- **Kontrola:** Je na vyučujícím, jestli bude nahrán kontrolovací systém nebo domácí úkol bude zkontrolován vyučujícím.
- **Výhody a nevýhody:** **Výhodou** je samotné odevzdávání, u kterého neprobíhá automatická kontrola. **Nevýhodou** pak nepřítomnost automatické kontroly dotazů SQL.
- **Dostupnost:** Systém je oficiální aplikace pro podporu výuky na FEL, je tedy dostupný pro všechny předměty vyučované na FEL.
- **Upravitelnost:** Systém je používán v několika předmětech na FEL. Je upravitelný pro potřeby předmětu DBS.
- **Technologie:** Systém je naprogramován v programovacím jazyce PHP<sup>2</sup>.
- **Vývoj a historie:** UploadSystem byl nasazen v roce 2005, stále je udržován.

### 3.1.3 Moodle

V roce 1999 Martin Dougiamas vytvořil systém pro podporu výuky Moodle. Na FEL je systém Moodle<sup>3</sup> další z hlavních systémů pro podporu výuky. Stejně jako Course Ware slouží k podpoře výuky jako takové, ne pouze k odevzdávání domácích úkolů.

<sup>1</sup> <http://cw.felk.cvut.cz/upload/>

<sup>2</sup> <http://php.net/>

<sup>3</sup> <https://moodle.fel.cvut.cz/>



The screenshot displays the Moodle LMS interface for FEL ČVUT. At the top, the user 'Martin Šubrt' is logged in. The main content area is titled 'Kurzy' and features a search bar with the text 'Vyhledat kurzy:' and a 'Proveď' button. Below the search bar, there is a list of course categories: 'Příprava následujícího semestru', 'Bakalářské studium', 'Magisterské studium', 'Doktorské studium', and 'Archiv'. To the left, there are three sidebars: 'MOJE KURZY' with a list of courses, 'NAVIGACE' with a 'Titulní stránka' menu, and 'SPRÁVA' with a 'Nastavení mého profilu' link. To the right, there is a 'UŽIVATELSKÁ PODPORA' section with contact information and a 'KALENDÁŘ' section showing a calendar for January 2017.

Obrázek 3.2. Systém Moodle

- **Účel:** Moodle je druhá z nejvíce využívaných aplikací na FEL. Slouží k podpoře výuky v mnoha předmětech.
- **Popis:** Stejně jako aplikace Course Ware slouží Moodle k podpoře výuky jako takové, ne pouze k odevzdávání domácích úkolů. Pro přednášející a cvičící je zde možnost zadávat testy pro studenty.
- **Kontrola:** Na Moodle je možné odevzdávat úkoly ve formě souboru. Automatické kontroly nejsou vynucené.
- **Výhody a nevýhody:** **Výhodou** je robustnost systému a jeho široké použití v mnoha předmětech na FEL i na mnoha dalších fakultách a dalších školách, v důsledku tohoto je systém velmi kvalitně rozvíjen a udržován. **Nevýhodou** pak nepřítomnost automatické kontroly dotazů SQL.
- **Dostupnost:** Systém je již používán podobně jako Course Ware v několika předmětech na FEL. Systém je ke stažení na webové adrese<sup>1</sup>. Jelikož systém je pod licencí GNU General Public License, je možné kód používat i upravovat pod pravidly této licence.
- **Upravitelnost:** Je možné vytvořit stránku předmětu s přednáškami a cvičeními. V rámci licence.
- **Autorská práva a licenční podmínky Moodle** Název Moodle™ je registrovanou obchodní známkou společnosti The Moodle Trust. Zamýšlíte-li využít tento název k podpoře komerčních aktivit, musíte si vyžádat povolení na adrese<sup>2</sup> v souladu s předpisy o obchodních známkách. Při jiném využití tohoto názvu neplatí žádná omezení. Držitel autorských práv k softwaru Moodle: Copyright © 1999 a dále, Martin Dougiamas. Tento program je volný software; můžete jej šířit a/nebo modifikovat podle ustanovení Obecné veřejné licence GNU, vydávané Free Software Foundation; a to buď verze 2 této licence anebo (podle vašeho uvážení) kterékoli pozdější verze. Tento program je rozšiřován v naději, že bude užitečný, avšak **bez jakékoli záruky**; neposkytují se ani odvozené záruky **prodejnosti** anebo **vhodnosti pro určitý účel**. Další podrobnosti

<sup>1</sup> <https://moodle.org/>

<sup>2</sup> [moodle.com/helpdesk](https://moodle.com/helpdesk)



hledejte v Obecné veřejné licenci GNU (její české znění přikládáme, je dostupné též na<sup>1</sup>, anglický originál na<sup>2</sup>).[3]

- **Technologie:** Systém je naprogramován v jazyce PHP.
- **Vývoj a historie:** Moodle vznikl v roce 1999, vytvořen byl Martinem Dougiamasem.

### ■ 3.1.4 Squeezer

Systém Squeezer je na FEL využíván ve dvou předmětech společných pro všechny obory studijního programu Softwarové technologie a management. Využívá se ke kontrole zdrojových kódů napsaných v programovacím jazyce Java. Autory projektu jsou Mgr. Michal Píše a Ing. Jan Drchal, Ph.D.

## Aktuální stav testovacího systému

- [Zadání domácích úkolů](#)
- [Seznam repozitářů čekajících na stažení do testovacího systému](#)
- [Seznam repozitářů čekajících na otestování](#)
- [Výstupy příkazu git log](#)
- [Logy posledních testů](#)



Evropský sociální fond. Praha & EU: Investujeme do vaší budoucnosti

Obrázek 3.3. Systém Squeezer

- **Účel:** Systém Squeezer je využívám v předmětech OMO<sup>3</sup> na webu<sup>4</sup> a DSA<sup>5</sup> na webu<sup>6</sup>.
- **Popis:** U systému se využívá principu GitLab repozitáře A na straně studenta a principu Hook A na straně kontrolujícího serveru. Systém kontroluje úkoly, které dávají jednoznačný a jediný možný výsledek k zadaným vstupním hodnotám. Na FEL jsou to kódy napsané v jazyce Java.

Student do svého GitLab repozitáře A nahraje řešení, které je ze strany serveru staženo systémem Hook A. Kontroly jsou prováděny 3–4 krát denně. Na straně serveru je použit CRON (příkaz operačního systému linux), ten vyvolá kontrolu všech stažených repozitářů. Systém porovnává výstup studentova řešení s výstupem referenčního řešení. Na základě rozdílu zaznamená, zda studentovo řešení prošlo či nikoliv. Studentovi též poskytuje zpětnou vazbu ve formě textového dokument dostupného z webu.

<sup>1</sup> <http://www.gnu.cz>

<sup>2</sup> <http://www.gnu.org/copyleft/gpl.html>

<sup>3</sup> Objektové modelování

<sup>4</sup> <http://omo.squeezer.felk.cvut.cz/>

<sup>5</sup> Databázové struktury a algoritmy

<sup>6</sup> <http://dsa.squeezer.felk.cvut.cz/>

- **Kontrola:** V obou výše zmiňovaných předmětech se systém předmětů se využívá k plně automatické kontrole odevzdaných řešení. Cvičící může po nahlédnutí do zdrojového kódu řešení studenta upravit rozhodnutí systému, děje se tak ale jen velmi vyjíměčně.
- **Výhody a nevýhody:** Automatická kontrola a přidělování bodů studentům je **výhodou** tohoto systému. Cvičící se nemusí zabývat každým odevzdaným řešením. Pouze uděluje různé výjimky. A při odhalení podobnosti dvou studentů systém varuje vyučující.  
Systém Squeezer plně kontroluje odevzdaná řešení studentů. U obrázkových souborů je to **nevýhodou**, nelze efektivně zkontrolovat obsah obrázku.
- **Dostupnost:** Systém je využíván na fakultě, jeho kód dostupný pro ostatní vyučující a předměty.
- **Upravitelnost:** Jelikož systém kontroluje výstup studentova řešení s výstupem řešení referenčního, musel by se upravovat celý systém kontroly.
- **Technologie:** Kontrolovány jsou kódy napsané v programovacím jazyce Java i referenční řešení úkolů je napsané v tomto jazyce. Pro stažení studentova repozitáře je vyvoláván příkaz systému Linux.
- **Vývoj a historie:** Systém je stále vyvíjen.

### 3.1.5 Baryk

Pro dva předměty funguje systém pro podporu výuky na FEL Baryk<sup>1</sup>.

**Testování úloh z předmětu Počítačové sítě**

**Úloha č.1 - TCP**



Robot Zemi verze 1.0, patří v baročce (přátelskému zrudení)



Evropský sociální fond  
Praha & EU: Investujeme do vaší budoucnosti

*Pokud chcete zaarchivovat svůj rozpracovaný zdrojový kód a tím prokázat aktivitu (povinné při každé větší změně kódu), označte na disku zdrojový kód a zadejte svoje uživatelské jméno:*

Soubor nevybrán

*Také můžete zkusit přeložit a otestovat kvalitu serverové části Vaší první úlohy. Zdrojový kód je zároveň zaarchivován. Požadavky na zdrojový kód a proces překladu jsou popsány v zadání.*

*Označte na disku zdrojový kód a zadejte svoje uživatelské jméno:*

Soubor nevybrán

*Pokud chcete otestovat kvalitu svého programu přes Internet, zadejte, prosím, Vaši IP adresu a port, na kterém Váš server běží:*

*Poznámka: Používejte raději první způsob (upload zdrojového kódu). Tím prokázujete svoji aktivitu během semestru. Testování přes internet neposkytuje vyučujícím žádnou informaci o historii Vašeho programu a úloha Vám (v rámci boje proti plagiátorství) nemusí být uznána.*

*Nevíte číslo portu, který máte použít? Zadejte uživatelské jméno a informujte se:*

*Také můžete otestovat samotný server Baryk.  
Můžete též zjistit, zda nějaké procesy přilís nevytáhní server Baryk.*

Obrázek 3.4. Systém Baryk

- **Účel:** Systém Baryk je využíván v předmětech PSI<sup>2</sup> a KBE<sup>3</sup>.
- **Popis:** Student odevzdává zdrojový kód napsaný v jazyce java nebo C/C++. Po nahrání je kód studenta archivován, přeložen a zkontrolován jeho výstup je porovnán s požadovanými výstupy. Na základě rozdílu vstupu studenta a požadovaných hodnot, systém ukáže, zda řešení prošlo (zobrazením zeleného rámečku) nebo neprošlo

<sup>1</sup> <http://baryk-ng.felk.cvut.cz/>

<sup>2</sup> Počítačové sítě

<sup>3</sup> Kódy a bezpečnost

(zobrazením červeného rámečku). Systém zobrazuje výstup do konzole. Na základě tohoto výstupu student může zjistit chyby v kódu.

- **Kontrola:** Řešení úkolu je přijatelné pouze, pokud projde kontrolou systému. Zadání úkolů může být rozděleno na podúkoly. Podle splněných podúkolů a po rozpravě jsou studentovi přiděleny body. Student může dostat 0 bodů, i kdyby měl splněny všechny podúkoly, pokud je prokázáno, že odevzdané řešení není originál.
- **Výhody a nevýhody:** Baryk porovnává výstup s jasně definovanými hodnotami. V předmětu DBS nejsou výstupy jasně definovány. Nutnost mít předem definované výstupy je pro nás **nevýhoda**.
- **Dostupnost a Upravitelnost:** Jelikož systém funguje v rámci ČVUT, systém je v rámci školy dostupný a upravitelný.
- **Technologie:** Kódy odevzdávané do systému jsou psané v programovacích jazycích Java nebo C/C++, i referenční řešení jsou napáány v těchto.
- **Vývoj a historie:** Autorem systému je Ing. Jiří Smítka. Aktivní vývoj stále trvá.

### 3.1.6 ProgTest

Systém ProgTest<sup>1</sup> je využíván na Fakultě Informačních technologií ČVUT.

Jméno	Hodnocení	Termin odevzdání	
Domácí úloha 01	5.00	23.10.2011 23:59:59	Zobrazit
Domácí úloha 02	5.00	30.10.2011 23:59:59	Zobrazit
Domácí úloha 03	6.00	06.11.2011 23:59:59	Zobrazit
Domácí úloha 04	0.00	13.11.2011 23:59:59	Zobrazit
Domácí úloha 05	5.00	20.11.2011 23:59:59	Zobrazit
Domácí úloha 06	0.00	27.11.2011 23:59:59	Zobrazit
Domácí úloha 07	6.00	05.12.2011 23:59:59	Zobrazit
Extra	0.00	14.12.2011 23:59:59	Zobrazit
Domácí úloha 08	0.00	18.12.2011 23:59:59	Zobrazit
Soutěžní domácí úloha	0.00	31.12.2011 23:59:59	Zobrazit
Test 01	8.00	10.01.2012 23:59:59	Zobrazit
Znalostní test - 02 demo	3.00	30.10.2011 23:59:59	Zobrazit
Znalostní test - 03 demo	3.00	13.11.2011 23:59:59	Zobrazit
Znalostní test - 04 demo	2.00	11.12.2011 23:59:59	Zobrazit
Znalostní test - 01	4.00	10.01.2012 23:59:59	Zobrazit
Znalostní test - 01 demo	2.00	16.10.2011 23:59:59	Zobrazit
Znalostní test - 02	4.00	10.01.2012 23:59:59	Zobrazit
Znalostní test - 03	3.00	10.01.2012 23:59:59	Zobrazit
Znalostní test - 04	2.00	10.01.2012 23:59:59	Zobrazit
Výsledky			Zobrazit

Obrázek 3.5. Systém ProgTest

- **Účel:** Systém je využíván k odevzdávání úkolů v předmětech BI-AG1<sup>2</sup>, BI-AAG<sup>3</sup>, BI-OSY<sup>4</sup>, BI-PA1<sup>5</sup>, BI-PA2<sup>6</sup> a BI-TED<sup>7</sup>.
- **Popis:** Do ProgTestu se dají odevzdávat zdrojové kódy programovacích jazyků C/C++ a  $\text{\TeX}$ . Přeložené programy dávají na základě vstupu jasně definované a očekávané výstupy. Student má omezené množství odevzdání
- **Kontrola:** Po vzniku zadání je vytvořeno (často jedním z cvičících) referenční řešení úkolu. Při kontrole studentova řešení jsou porovnávány výstupy referenčního a studentova řešení. Kontrolovány jsou také další faktory, jako např. neuvolněná paměť

<sup>1</sup> <https://progtest.fit.cvut.cz/index.php/>

<sup>2</sup> Algoritmy a grafy 1

<sup>3</sup> Automaty a gramatiky

<sup>4</sup> Operační systémy

<sup>5</sup> Programování a algoritmizace 1

<sup>6</sup> Programování a algoritmizace 2

<sup>7</sup> Technická dokumentace

programu. Na základě kontroly jsou studentovy přiděleny body. Cvičící ani přednášející nemohou přidělené body systémem upravovat. Při prokázání, že odevzdané řešení není studentovo vlastní, mohou mu být odebrány body.

- **Výhody a nevýhody:** **Nevýhoda** je automatická oprava úkolů a přidělování bodů studentovi.
- **Dostupnost a Upravitelnost:** Podobně jako systém Baryk, systém ProgTest funguje v rámci ČVUT, je tedy v rámci školy dostupný a upravitelný.
- **Technologie:** Systém je naprogramován v jazyce PHP.
- **Vývoj a historie:** Systém vznikl v roce 2008. Autorem je Ing. Ladislav Vagner, Ph.D. V roce 2012 vznikly další komponenty, na kterých spolupracovali Ing. Ondřej Guth, Ph.D. a Ing. Jan Žďárek, Ph.D.

### ■ 3.1.7 Závěr analýzy

Po analýze těchto odborných systémů pro odevzdávání domácích úkolů fungujících na FEL zamítáme Course Ware a Moodle pro jejich přílišnou robustnost. Pro potřeby předmětu DBS by oba systémy použít šly, avšak zavrhlí jsme je, toho je popsáno více dále v této kapitole.

Systémy Squeezer, Baryk a ProgTest nepoužijeme pro jejich způsob kontroly domácích úkolů. Tyto systémy porovnávají výstup domácího úkolu studenta s výstupem referenčního řešení tohoto úkolu. U předmětu DBS nelze referenční řešení vytvořit, protože výstup řešení úkolu každého studenta je jiné.

V systému XML Check doposud také nebyla možná automatická kontrola syntaxe SQL dotazů, avšak přidání této kontroly podle analýzy je stejně složité jako u některých výše popsaných systémů. Systém XML Check, je využíván v předmětu *Technologie XML*, který byl v době vzniku nápadu vytvořit (upravit) odevzdávací systém na domácí úkoly v předmětu DBS na FEL i na MFF vyučován Doc. RNDr. Irenou Holubovou, Ph.D.

## ■ 3.2 Systém XML Check

Nejdříve se podíváme, co systém XML Check umí a umožňuje a co je potřeba v systému vytvořit.

Systém umí přijmout domácí úkol v souboru od studenta a spustit nad tímto úkolem kontrolující plugin. Kontrolujících pluginů v systému nyní existuje řada, avšak kontrolují syntax jazyka XML a jemu podobných jazyků. Systém umí přijmout od studenta soubory typu obrázku.

V systému není plugin na kontrolu syntaxe jazyka SQL, ten je nutné vytvořit.

Systém nenabízí možnost přidat nebo zobrazit příklady k procvičení, ani jejich řešení.

Jak konkrétně bude systém rozšířen si popíšeme v následující sekci.

## ■ 3.3 Analýza rozšiřované části systému XML Check

### ■ 3.3.1 Účel aplikace

Jednoduché odevzdávání úkolů s jejich automatickou kontrolou. Přidávání příkladů k procvičení, jejich správná a špatná řešení.

### ■ 3.3.2 Zainterесované osoby a instituce (stakeholders)

- **FEL:** Systém XML Check je na fakultě využíván.
- **MFF:** Systém XML Check je na fakultě využíván.
- **Vyučující na FEL a MFF:**
- **Studenti na FEL a MFF:**

### ■ 3.3.3 Uživatelé

- Typickým uživatelem je **přednášející**, který zadává problémy do jím přednášených předmětů. K problémům přidává, upravuje nebo maže příklady k procvičení pro studenty.
- Dalším uživatelem je **cvičící**. Odevzdané úkoly studenty hodnotí právě jejich cvičící. Cvičící také může přidávat, upravovat i mazat příklady k procvičení.
- **Student** odevzdává úkoly. Má možnost procházet příklady k procvičení.

### ■ 3.3.4 Požadavky na funkcionalitu a klíčové potřeby uživatelů

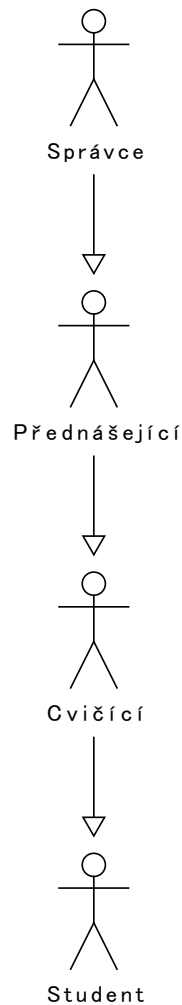
- Možnost odevzdávat úkoly v předmětu DBS
- Automatická kontrola úkolů v předmětu DBS
- Podporu pro zadávání příkladů k procvičení
- Podporu pro přidávání správných a špatných řešení u příkladů k procvičení

### ■ 3.3.5 Funkční požadavky

- **FR01: Nahrání domácího úkolu**  
Systém umožní studentovi nahrát domácí úkol ve formátech podporovaných v předmětu DBS.
- **FR02: Podání zpětné vazby**  
Systém podá studentovi zpětnou vazbu na jím odevzdaný úkol.
- **FR03: Hodnocení domácího úkolu**  
Systém umožní cvičícímu ohodnotit domácí úkol.
- **FR04: Vytvoření správného řešení příkladu k procvičení**  
Systém umožní cvičícímu vytvořit správné řešení daného příkladu k procvičení.
- **FR05: Smazání správného řešení příkladu k procvičení**  
Systém umožní cvičícímu smazat správné řešení daného příkladu k procvičení.
- **FR06: Vytvoření špatného řešení příkladu k procvičení**  
Systém umožní cvičícímu vytvořit správné řešení příkladu k procvičení.
- **FR07: Smazání špatného řešení příkladu k procvičení**  
Systém umožní cvičícímu smazat správné řešení příkladu k procvičení.
- **FR08: Úprava řešení příkladu k procvičení**  
Systém umožní cvičícímu upravit řešení příkladu k procvičení.
- **FR09: Vytvoření příkladu k procvičení**  
Systém umožní cvičícímu vytvořit nový příklad k procvičení k danému problému. K nově přidanému příkladu k procvičení musí být přidáno minimálně jedno správné a jedno nesprávné řešení.
- **FR10: Upravení příkladu k procvičení**  
Systém umožní cvičícímu upravit příklad k procvičení. Přidat, odebrat nebo upravit jeho správná i špatná řešení.

**FR11: Smazání příkladu k procvičení**

Systém umožní cvičícímu smazat příklad k procvičení a všechna jeho řešení.

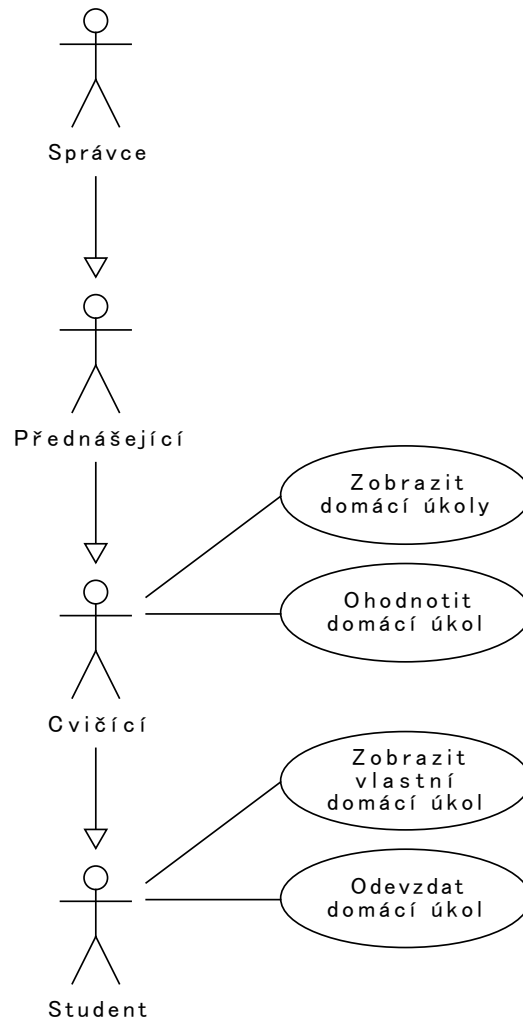
**3.3.6 Případy užití****Uživatelé**

**Obrázek 3.6.** Uživatelé

- Správce má absolutní práva.
- Přednášející spravuje předmět jako celek, tedy přednášky, testy, i příklady k procvičení a jejich řešení
- Cvičící má možnost přidávat, upravovat nebo mazat problémy, úkoly. Procházet řešení studentů svých skupin. Přidávat hodnocení k odevzdaným úkolům. Má i možnost upravovat příklady k procvičení daného předmětu a jejich správná a nesprávná řešení.
- Student má možnost odevzdávat úkoly. Procházet své úkoly. Procházet příklady k procvičení, jejich správná a nesprávná řešení.

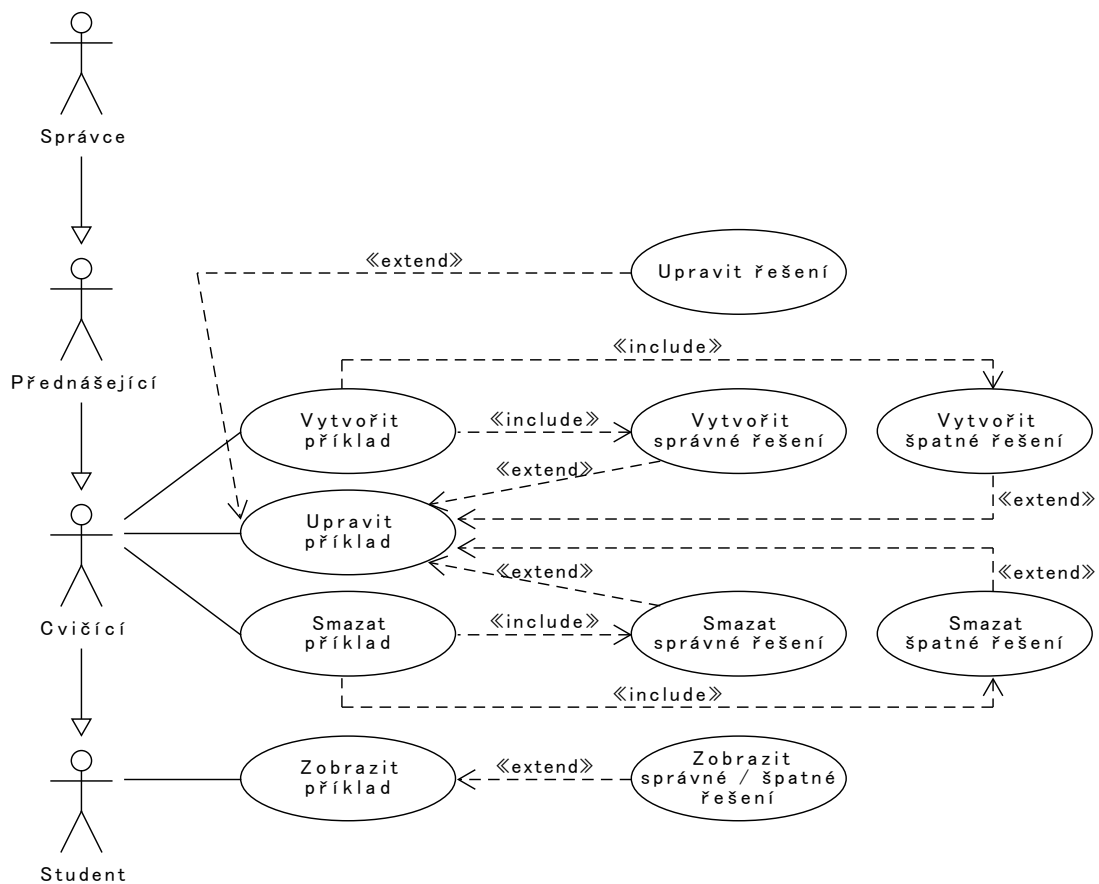
## ■ Případy užití

- Odevzdávání domácích úkolů



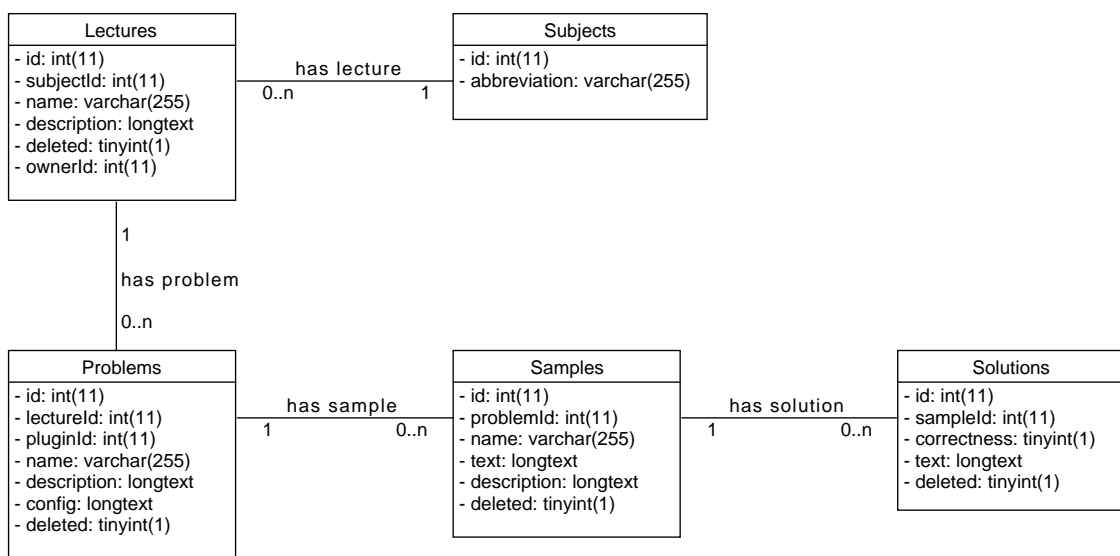
**Obrázek 3.7.** Odevzdávání domácích úkolů

- Správa příkladů k procvičení



Obrázek 3.8. Správa příkladů k procvičení

### 3.3.7 Relační model



Obrázek 3.9. Část realčního modelu po rozšíření



■ **Problem**

Problem (problém) v systému XML Check je objekt spojený s nějakou přednáškou, který popisuje, co student musí učinit, aby daný problém vyřešil. Problém je často asociovaný s nějakým kontrolujícím pluginem.[2]

■ **Sample**

Sample (příklad k procvičení) v systému XML Check je právě k jednomu problému. Má alespoň jedno správné a alespoň jedno nesprávné řešení. Řešení může mít neomezeně mnoho.

■ **Solution**

Solution (řešení) v systému XML Check je řešení příkladu k procvičení. Je právě k jednomu příkladu k procvičení. Řešení může být správné nebo nesprávné. Příklad k procvičení může mít neomezeně mnoho řešení.

# Kapitola 4

## Vlastní práce

### 4.1 Rozšíření pro předmět DBS

Jako první úkol této práce je rozšíření systému XML Check. A to pro další předmět, konkrétně pak pro předmět Databázové systémy. Doposud systém fungoval jako odevzdávací systém pouze pro předmět *Technologie XML*. Předmět DBS je stále vyučován podobnou formou jako předmět *Technologie XML* a do roku 2015 byl vyučován i stejnou vyučující Doc. RNDr. Irenou Holubovou, Ph.D., proto v roce 2015 vznikl nápad rozšířit systém XML Check.

Nyní se pojdme podívat na to, co bylo potřeba udělat, aby systém mohl přijímat a kontrolovat domácí úkoly z dalšího předmětu.

#### 4.1.1 Rozšířit systém XML Check pro další předmět obecně

- V databázi byla přidána nová tabulka `Subjects` (Předměty).

Subjects
- id: int(11) - abbreviation: varchar(255)

Obrázek 4.1. Tabulka Subjects

Viz relační model 3.9.

Tabulka má 2 atributy:

- **id** Jednoznačný celočíselný identifikátor záznamu v tabulce, je primárním klíčem této tabulky.
- **abbreviation** Jedinečná zkratka předmětu o 3 znacích, tuto zkratku mají všechny předměty na FEL.

Celá jmána předmětů jsou získávána až na úrovni PHP skriptů na základě jejich jedinečných značek.

Do této tabulky jsou vloženy 2 předměty s těmito záznamy.

```
Technologie XML {id: 1, abbreviation: "xml"}  
Databázové systémy {id: 2, abbreviation: "dbs"}
```

Takovéto řešení získávání celých jmen předmětů bylo zvoleno kvůli jednoduchosti a rozšiřitelnosti. Jméno předmětu by bylo možné vložit přímo jako sloupec tabulky v databázi. Avšak pro každou jazykovou mutaci by musel v tabulce existovat vlastní sloupec se jménem v daném jazyce, při potenciálním rozšíření systému přidáním další jazykové mutace by se musel přidat nový sloupec. Z tohoto důvodu nebylo rozšiřováno uživatelské rozhraní pro správu předmětů.

Na nutnost rozšíření databáze (přidání záznamu do tabulky Subject) i rozšíření PHP skriptů **je potřeba si dát pozor!**

- Při zobrazování přednášek bylo doposud zobrazováno pouze jméno přednášky. Nyní je zobrazeno i jméno předmětu, jakéhož je tato přednáška součástí.
- **Ukázka uživatelského rozhraní**

Předmět	Jméno	Popis
Technologie XML	lecture1	test lecture 1
Technologie XML		test lecture 1

Obrázek 4.2. Tabulka přednášek v uživatelském rozhraní

**Přidat nebo upravit přednášku**

Předmět: Technologie XML

Jméno přednášky: Databázové systémy

Popis:

Potvrdit

Obrázek 4.3. Formulář pro úpravu přednášky v uživatelském rozhraní

#### 4.1.2 Rozšířit systém XML Check pro předmět DBS

Pro kontrolu domácích úkolů musí být v systému přítomny kontrolující pluginy. I pro předmět DBS musí být nějaké pluginy, pokud chceme domácí úkoly kontrolovat a dávat studentům zpětnou vazbu. Jak jsou tedy jednotlivé domácí úkoly předmětu DBS systémem XML Check kontrolovány?

- Řešení 1. a 2. domácího úkolu je odevzdáváno ve formě PDF či obrázku. Obsah těchto řešení může být velmi různorodý. V této formě nemůže být obsah řešení efektivně kontrolován. Je zvolena kontrola, že odevzdaný soubor je ve formátu PDF či obrázku.

Tato funkcionalita je již v systému přítomna.

Studentovi není poskytnuta žádná nápověda ohledně odevzdaného řešení. Vyučujícímu je ulehčeno od množství e-mailů.

- Řešení 3. a 4. domácího je odevzdáváno ve formě SQL. Zde jsou vytvořeny pluginy pro kontrolu těchto domácích úkolů.

Pro kontrolu syntaxe SQL dotazů je použit SQL parser, jež je využíván v aplikaci phpMyAdmin<sup>1</sup>. Kód je napsán v jazyce PHP, což je jazyk, ve kterém je napsán i systém XML Check. Ke stažení je zde<sup>2</sup>. Vlastní parser není nijak upraven. Je využitelný pod veřejnou licencí GNU<sup>3</sup>. Nevzniká zde tedy problém s autorskými právy.

Pro 3. domácí úkol je přidána kontrola vytvoření samotné databáze, tedy musí být alespoň pětkrát přítomný dotaz typu `CREATE`, následně `INSERT INTO` a 3 integritních omezení (jedno z nich musí být typu `CHECK`).

Ve čtvrtém úkolu má student více povinností ohledně psaných dotazů na jeho databázi, s tím je spojena i složitější kontrolující plugin. Stále jde o vyhledávání klíčových slov v dotazech, která musí být v nich přítomna. Student by mohl vložit všechny požadované části dotazů do jednoho nebo dvou (množinovou operaci `UNION/INTERSECT/EXCEPT`, třídění `ORDER BY`, agregaci `GROUP BY`, `HAVING`, vnořený poddotaz, přirozené, vnitřní a vnější spojení `JOIN` a dva pohledy `CREATE VIEW`). Avšak v požadavku úkolu je 5 netriviálních dotazů, tedy jedna z výše zmíněných operací musí být v každém z pěti dotazů.

## 4.2 Přidání příkladů k procvičení

Dalším úkolem této práce bylo přidání nové funkcionality, a to příkladů k procvičování pro studenty.

V minulých letech neexistovala tato možnost skrze systém XML Check. Studenti se učili z přednášek a příkladů předvedených na cvičeních. Ani vyučující neměli možnost vidět všechny předváděné příklady najednou a tak mohly vznikat nejasnosti na straně cvičících.

Ze strany vyučujících bylo vyjádřeno přání na přidání této funkcionality do systému. Systém byl rozšířen na řadě míst, aby tato možnost byla přidána.

### 4.2.1 Rozšíření databáze

- Nejdříve byla přidána tabulka do databáze `Samples` (Příklady k procvičení).

<sup>1</sup> <https://www.phpmyadmin.net/>

<sup>2</sup> <https://github.com/phpmyadmin/sql-parser>

<sup>3</sup> <http://www.gnu.cz>

Samples
<ul style="list-style-type: none"> <li>- id: int(11)</li> <li>- problemId: int(11)</li> <li>- name: varchar(255)</li> <li>- text: longtext</li> <li>- description: longtext</li> <li>- deleted: tinyint(1)</li> </ul>

Obrázek 4.4. Tabulka Samples

Viz relační model 3.9.

Tabulka má atributy:

- **id** Jednoznačný celočíselný identifikátor záznamu v tabulce, je primárním klíčem této tabulky.
- **problemId** Cizí klíč, identifikátor problému, na který je tento příklad k procvičení navázán.
- **name** Jméno příkladu k procvičení pro usnadnění vyhledávání.
- **text** Znění samotného příkladu.
- **description** Vysvětlivky k danému příkladu. Může zde být uvedeno, kde se s takovýmto příkladem student setká v praxi, k čemu se v praxi používá a jakou problematiku demonstruje.
- **deleted** Příznak, zda byl příklad ze systému odstraněn. Tento argument má většina tabulek v systému.

Možnost přidávání a spravování příkladů k procvičení byla přidána nejen přednášejícímu, ale i cvičícímu. Ve většině případů právě cvičící vědí, jaké příklady studentům dělají problémy.

Přednášející má samozřejmě tuto možnost také, z hlediska zkušeností ohledně přednášených problematik je na tom nejlépe právě přednášející.

Další, kdo má možnost spravovat příklady k procvičení, je Admin. Kvůli celkové správě systému.

- U SQL dotazů ani u jazyka XML nemusí být vždy jen jedno správné řešení problému a ke kýženému výsledku se dá dostat více různými způsoby. Proto již při analýze této problematiky bylo zřejmé, že systém musí nabízet možnost vytvoření více řešení příkladu k procvičení, ať už správných nebo nesprávných.

Příklad k procvičení tedy může mít N řešení. Pro řešení příkladů byla do databáze přidána tabulka Solutions.

Solutions
- id: int(11) - sampleId: int(11) - correctness: tinyint(1) - text: longtext - deleted: tinyint(1)

Obrázek 4.5. Tabulka Solutions

Viz relační model 3.9.

Tabulka má atributy:

- **id** Jednoznačný celočíselný identifikátor záznamu v tabulce, je primárním klíčem této tabulky.
- **sampleId** Cizí klíč, identifikátor příkladu, který toto řešení řeší.
- **correctness** Správnost řešení, řešení může být správné či nesprávné.
- **text** Znění samotného řešení příkladu, zpravidla SQL dotaz / XML.
- **description** Vysvětlivky k danému řešení příkladu. Proč dané řešení je / není správné, k jakému výsledku vede apod.
- **deleted** Příznak, zda byl příklad ze systému odstraněn.

## ■ 4.2.2 Rozšíření uživatelského rozhraní

- V uživatelském rozhraní byla u tabulky problémů přidána možnost zobrazit příklady k procvičení vázané na daný problém.
- V postranním menu byla u cvičícího přidána možnost Příklady k procvičení.
- V postranním menu byla u cvičícího přidána možnost Řešení příkladů k procvičení.
- V postranním menu byla u studenta přidána možnost Příklady k procvičení.
- **Ukázka uživatelského rozhraní**

Z pohledu cvičícího

The screenshot shows the 'XML Check 1.0.g' application interface. On the left is a navigation menu with categories like 'Student', 'Cvičení', 'Přednášející', 'System', and 'Nastavení'. The main area displays a table titled 'Příklady k procvičení' with columns: Problém, Jméno, Text, and Popis. The table contains three rows of data.

	Problém	Jméno	Text	Popis
	problem1	sample3	testim testim testim	
	problem1	sample2	testuji tez více radku	
	problem1	sample1	testuji sampl	

At the bottom of the table, it says 'Zobrazit 15 řádků na stránce'.

**Obrázek 4.6.** Tabulka příkladů k procvičení v uživatelském rozhraní z pohledu cvičícího  
Cvičící může:

- přidat nový příklad k procvičení
- upravit existující příklad k procvičení
- odstranit existující příklad k procvičení
- zobrazit tabulku řešení daného příkladu k procvičení

Při otevření tabulky řešení konkrétního příkladu k procvičení se zobrazí tabulka s nastaveným filtrem právě pro daný příklad k procvičení.

Všechny výše zmíněné akce může provádět i přednášející a ADMIN systému.

The screenshot shows the 'XML Check 1.0.g' application interface. The main area displays a table titled 'Řešení příkladu' with columns: Příklad, Správnost, Text, and Popis. The table contains three rows of data.

	Příklad	Správnost	Text	Popis
	sample1	Správně	test	
	sample2	Správně	resim sample test	
	sample2	Nesprávně	řešení test	
	sample3	Správně	test	test correct create

At the bottom of the table, it says 'Zobrazit 15 řádků na stránce'.

**Obrázek 4.7.** Tabulka řešení příkladů k procvičení v uživatelském rozhraní z pohledu cvičícího

Cvičící může:

- přidat nové řešení příkladu k procvičení
- upravit existující řešení příkladu k procvičení
- odstranit existující řešení příkladu k procvičení

Z pohledu studenta

The screenshot shows the 'XML Check 1.0.9' application interface. The top navigation bar includes the application name, a user profile icon for 'Administrator', and a 'Logout' button. A left sidebar menu is expanded to 'Student', showing options like 'Subjects', 'Assignments', 'Submissions', 'Subscriptions', 'Samples for Practise', 'Tutor', 'Lecturer', 'System', and 'Settings'. The main content area displays a table titled 'Samples for practise' with the following data:

Assignment	Name	Text	Description
problem1	sample3	testim testim testim	
problem1	sample2	testuji tez vice radku	
problem1	sample1	testuji sampl	

At the bottom of the table, it says 'Show 15 rows per page' and '0 / 0'.

**Obrázek 4.8.** Tabulka příkladů k procvičení v uživatelském rozhraní z pohledu studenta

Student může:

- zobrazit detail příkladu k procvičení
- zobrazit řešení daného příkladu

The screenshot shows the 'XML Check 1.0.9' application interface. The top navigation bar includes the application name, a user profile icon for 'Administrator', and an 'Odhlásit se' button. A left sidebar menu is expanded to 'Student', showing options like 'Předměty', 'Úkoly', 'Řešení', 'Členství', 'Příklady k procvičení', 'Cvičení', 'Přednášející', 'Systém', and 'Nastavení'. The main content area displays a table titled 'Řešení příkladu' with the following data:

Příklad	Správnost	Text	Popis
sample2	Správné	resim sample test	
sample2	Nesprávné	řešení test	

At the bottom of the table, it says 'Zobrazit 15 řádků na stránce' and '1 / 1'.

**Obrázek 4.9.** Tabulka řešení příklad k procvičení v uživatelském rozhraní z pohledu studenta

Student může zobrazit detail řešení příkladu k procvičení.

Při zobrazení detailu příkladu k procvičení, nebo jeho řešení, je zobrazen speciální formulář, který má needitovatelné pole.



The screenshot shows the 'XML Check 1.0.9' application interface. On the left is a navigation menu under the 'Student' header, with options: Předměty, Úkoly, Řešení, Členství, Příklady k procvičení (highlighted), Cvičíci, Přednášející, Systém, and Nastavení. The main content area is titled 'Podrobnosti o příkladu k procvičení' and displays the following details:

Úkol:	problem1
Jméno:	sample1
Text:	testuji sampl
Popis:	

**Obrázek 4.10.** Detail příkladu k procvičení v uživatelském rozhraní z pohledu studenta

The screenshot shows the 'XML Check 1.0.9' application interface. On the left is the same navigation menu as in the previous screenshot. The main content area is titled 'Řešení příkladu' and displays the following details:

Příklad:	sample1
Správnost:	Správné
Text:	test
Popis:	

**Obrázek 4.11.** Detail řešení příkladu k procvičení v uživatelském rozhraní z pohledu studenta

## 4.3 Oprava chyb

Oprava chyb byla posledním úkolem této práce.

Hlavní chybou, která byla opravena, je preference jazyka uživatelského rozhraní. Po instalaci systému byl zobrazovaný jazyk anglický, na český musel uživatel systém přepnout. Chyba byla ve čtení cookie `language`, která ještě nebyla nastavena. Nyní, když cookie `language` ještě není nastavena, zobrazí se jazyková verze podle jazykového nastavení prohlížeče. Pokud má uživatel zobrazuje stránku v prohlížeči s preferencí českého jazyka, zobrazí se česká jazyková mutace webu, jinak se zobrazí anglická jazyková mutace.

Většina kódu byla refaktorována, nyní je lépe čitelný.

## 4.4 Testování

V této fázi byly u systému rozšířeny databáze a uživatelské rozhraní. Systém dříve podporoval pouze jeden předmět, dosud nebyly vytvořeny automatické testy pro testování uživatelského rozhraní systému XML Check. Stejně tak pro příklady k procvičení a jejich řešení bylo testováno uživatelské rozhraní.

Testování uživatelského rozhraní probíhalo progresně regresními testy.

### Progresní a regresní testy

Progresní testy využíváme při kontrole nových funkcí nebo vlastností aplikace. K jejich správnému provedení je nutná dokumentace, která přesně popisuje nově implementované oblasti. Používáme je ve všech etapách testování.

Regresní testy se využívají při opětovném testování funkcí a vlastností aplikace. Jejich smyslem je ověření, že provedené změny či implementace nových vlastností v aplikaci nemělo žádný vliv na stávající funkce a vlastnosti. Tedy především na oblasti, které zůstaly v programovém kódu nezměněny. Oblasti, které byly předmětem úprav, by správně již měly být otestovány funkčními testy. Takovéto situace z pravidla nastávají po opravení chyb či po novém release. Regresní testy je velmi vhodné automatizovat.

V praxi jsou regresní testy velmi rozšířené. V různých formách se používají prakticky u většiny projektů. Využívají se hlavně při retestech opravených chyb. Oproti tomu progresní testy nejsou příliš rozšířeny a často je tato kategorie testů rozložena do jiných kategorií.[4]

# Kapitola 5

## Závěr

Cílem této práce bylo rozšířit systém pro předmět *DBS*, přidat možnost zadávat příklady k procvičení a opravit nejzávažnější chyby systému.

O provedené práci je psáno v kapitole 4

Rozšířit systém pro další předmět se podařilo obecně, nyní je jednoduché přidat další předmět do systému, i konkrétně pro předmět *DBS*. U přednášky je zobrazováno i jméno předmětu, jehož je tato přednáška součástí. Při vytváření nové přednášky musí přednášející zvolit jeden z předmětů, které se v databázi nacházejí.

Podpora možnosti přidání příkladů k procvičení je nyní funkční a otestovaná regresními testy. Cvičící mohou přidávat příklady k procvičení, editovat je, i mazat (schovávat). Mohou také zobrazovat, vytvářet, editovat či mazat (schovávat) správná i nesprávná řešení těchto příkladů k procvičení. Studenti mohou tyto příklady zobrazovat, řadit jejich pořadí podle názvu, textu nebo i názvu domácího úkolu, ke kterému se vztahují. Cvičící i studenti mají větší přehled v nabízených příkladech.

Většina opravených chyb se týkala pouze refaktoringu kódu PHP skriptů. Opravena byla chyba jazykového nastavení systému, systém dříve předpokládal existenci souboru, který ale nebyl před tím získáván, proto nefungovala česká jazyková mutace jako výchozí. Nyní je česká jazyková mutace nastavena jako výchozí pro webové prohlížeče, které mají nastavený český jazyk jako primární, pro ostatní anglická jazyková mutace.

Před nasazením do produkce je nutné doimplementovat kontrolující pluginy pro předmět *DBS*, jinak je systém zcela připraven k vydání.



## Literatura

- [1] Jan Konopásek. *Systém pro správu úkolů a testů*. 2011.  
<http://www.ksi.mff.cuni.cz/~holubova/bp/Konopasek.pdf>.
- [2] Petr Hudeček. *Rozšíření systému XML Check*. 2015.  
<http://www.ksi.mff.cuni.cz/~holubova/bp/Hudecek.pdf>.
- [3] *Autorská práva a licenční podmínky Moodle*. 2016.  
<https://docs.moodle.org/archive/cs/Licence>.
- [4] *Progresní a regresní testy*. 2017.  
<http://testovanisoftwaru.cz/tag/regresni-testy/>.

# Příloha A

## Zkratky a pojmy

- GitLab repozitář Gitlab repozitář je vzdálené uložení, které nabízí „zrcadlit“ zpravidla kód na tomto uložení. Nadstavbou této služby je nástroj Git<sup>1</sup>, verzovací systém. Verzovací znamená, že umožňuje uživateli mít více větví programu, kde každý commit (aktualizace) je zaznamenáván. Tím pádem má programátor data chráněna a ještě se může vracet zpět, pokud danou větev sledá nehodnou.
- Hook Neboli Git Hook<sup>2</sup> je podslužba služby Git. Hook běžně vyvolává nějaké akce, když se objeví nějaká událost. Například může vyvolat stažení repozitáře do jiného.

---

<sup>1</sup> <https://git-scm.com/>

<sup>2</sup> <https://git-scm.com/book/gr/v2/Customizing-Git-Git-Hooks>

## Příloha B

### Obsah přiloženého CD

- Text bakalářské práce (soubor BP-Subrt.pdf)
- Zdrojový kód, project IntelliJ IDEA (adresář project)
  - V adresáři www je webová aplikace
  - Jsou přítomny další adresáře a soubory nutné pro programátorskou instalaci