

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

## **Diplomová práce**

# **E-learningový systém s podporou mobilních zařízení**

**Podklad pro zadání DIPLOMOVÉ práce studenta**

<b>PŘEDKLÁDÁ:</b>	<b>ADRESA</b>	<b>OSOBNÍ ČÍSLO</b>
Bc. PRAUM Kamil	Strašice 544/II, Strašice	A12N0082P

**TÉMA ČESKY:**

E-learningový systém s podporou mobilních zařízení

**NÁZEV ANGLICKY:**

E-learning System with Support of Mobile Devices

**VEDOUcí PRÁCE:**

Ing. Ladislav Pešička - KIV

**ZÁSADY PRO VYPRACOVÁNÍ:**

1. Prozkoumejte vybrané dostupné systémy pro tvorbu e-learningového obsahu a jejich použitelnost na mobilních platformách.
2. Analyzujte vhodný formát vytváření výukového obsahu s důrazem na snadnou použitelnost a multiplatformnost.
3. Navrhněte systém pro authoring (tvorbu e-learningového obsahu), který bude v těsné integraci se stávajícím systémem pro výuku. Tento obsah půjde pohodlně prohlížet na mobilních platformách (mobilní web, nativní aplikace pro operační systém Android).
4. Dále navrhněte rozšíření výukových testů a statistik stávajícího systému pro online testování.
5. Navržený systém realizujte a ověřte jeho funkcionalitu v prostředí katedry, zhodnoťte možnosti dalšího rozšíření systému.

**SEZNAM DOPORUČENÉ LITERATURY:**

Dodá vedoucí práce.

Podpis studenta: .....

Datum: .....

Podpis vedoucího práce: .....

Datum: .....



# Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 28. dubna 2015

Kamil Praum.....

# Poděkování

Na tomto místě bych rád poděkoval rodině, přítelkyni a všem, kteří mě podporovali v mé práci.

# **Abstract**

## **E-learning System with Support of Mobile Devices**

This work deals with the creation of system for e-learning and student's knowledge testing with supporting mobile devices.

After the analysis of available systems was designed and successfully realized a system suitable for authoring and e-learning in the environment of the faculty. The system was implemented as the web portal. The native application for mobile devices with Android operating system were also created.

*Keywords: e-learning, m-learning, LMS, mobile devices, tablet, authoring*

# Abstrakt

## E-learningový systém s podporou mobilních zařízení

Tato práce se zabývá tvorbou systému pro elektronickou výuku a testování znalostí studentů s podporou mobilních zařízení.

Po analýze dostupných systémů byl navržen a úspěšně realizován systém pro výuku a tvorbu obsahu v prostředí fakulty. Systém je implementován jako webový portál. Byla také vytvořena nativní mobilní aplikace pro operační systém Android.

*Klíčová slova: e-learning, m-learning, LMS, mobilní zařízení, tablet, authoring*

# Obsah

<b>1</b>	<b>Úvod</b>	<b>2</b>
<b>2</b>	<b>Průzkum dostupných systémů</b>	<b>3</b>
2.1	Typy vzdělávání . . . . .	3
2.2	Příklady systémů . . . . .	7
<b>3</b>	<b>Analýza</b>	<b>12</b>
3.1	Výukový obsah . . . . .	12
3.2	Podpora mobilních zařízení . . . . .	13
3.3	Aplikační rozhraní . . . . .	16
<b>4</b>	<b>Seznam realizovaných funkcí</b>	<b>17</b>
<b>5</b>	<b>Webová verze systému</b>	<b>19</b>
5.1	Authoring modul . . . . .	19
5.2	Webová služba . . . . .	23
5.3	Rozšíření statistik . . . . .	29
5.4	Rozšíření výukových testů . . . . .	32
5.5	Rozšíření ostrých testů . . . . .	35
5.6	Kontextová nápověda a notifikace . . . . .	38
5.7	Drobečková navigace . . . . .	39
5.8	Hromadný import studentů . . . . .	40
5.9	Skrytí předmětu . . . . .	41
<b>6</b>	<b>Podpora mobilních zařízení</b>	<b>42</b>
6.1	Redesign webu . . . . .	42
6.2	Implementace aplikace pro operační systém Android . . . . .	49
6.3	Publikace mobilní aplikace . . . . .	54
<b>7</b>	<b>Automatické testování</b>	<b>58</b>
7.1	Jednotkové testy . . . . .	58
7.2	Integrační testy . . . . .	60
7.3	Funkční testy . . . . .	60
7.4	Výhody automatického testování aplikací . . . . .	61
7.5	Psaní automatických testů . . . . .	61
7.6	Vývoj řízený testy . . . . .	62
7.7	Implementace testů . . . . .	63
<b>8</b>	<b>Funkčnost řešení a statistiky</b>	<b>66</b>
<b>9</b>	<b>Možnosti dalšího rozšíření systému</b>	<b>73</b>
<b>10</b>	<b>Závěr</b>	<b>74</b>

# 1 Úvod

Teri je systém pro elektronickou výuku dlouhodobě používaný v prostředí katedry. Původní verzi systému jsem vytvořil v rámci svojí bakalářské práce a byla zaměřena čistě na testování znalostí studentů. Cílem práce je rozšířit stávající e-learningový systém Teri o authoring modul pro tvorbu a publikaci studijních materiálů. Učitelé a studenti by měli dostat komplexní nástroj vhodný pro elektronickou výuku. Dalším cílem je rozšířit výukové a ostré testy, které byly implementovány již v původní verzi systému.

Při tvorbě bude kladen důraz na multiplatformnost systému. Celý návrh a implementace má za cíl zpřístupnit výuku na počítačích a mobilních zařízeních. Klíčová je tedy snadná použitelnost na většině zařízení, které mají studenti k dispozici.

Výsledná aplikace bude nasazena v reálném provozu při výuce a testování studentů. Tím bude dosaženo praktické ověření funkčnosti a použitelnosti.

Obecně je potenciál e-learningu s využitím mobilních technologií ve výuce málo využit. To je hlavní důvod pro tvorbu této práce. Původní systém Teri se během několika let osvědčil ve výuce v prostředí katedry. Byl však zaměřen na úzkou oblast elektronického vzdělávání pokrývající pouze testování znalostí studentů. Proto je vhodné ho dále rozšiřovat, aby se stal více komplexním a použitelným pro širší využití.

## 2 Průzkum dostupných systémů

V této kapitole je obecně popsána problematika elektronické výuky a jsou zde také uvedeny konkrétní příklady implementací řešení pro e-learning.

### 2.1 Typy vzdělávání

Elektronická výuka je zde téměř od počátku rozvoje osobních počítačů. První výukové aplikace se distribuovaly na disketách a později na optických médiích. Teprve až s rozvojem internetu oblast elektronického vzdělávání zažila vzestup. V současné době se jedná o velmi diskutované téma v oblasti vzdělání. E-learning lze rozdělit na dvě oblasti. Na e-learning **synchronní** a **asynchronní** [1].

Synchronní varianta vyžaduje přímou interakci s lektorem. Výuka tedy často probíhá klasicky ve třídě jen za pomoci e-learningového softwaru, nebo na dálku a studenti komunikují online. Výuka ale vždy probíhá v konkrétní čas. Forma synchronního e-learningu je často využívána ve školách. S rozvojem mobilních technologií jako jsou tablety se začíná na progresivních školách po celém světě stávat majoritní formou výuky.

Asynchronní e-learning má filosofii nastavenou směrem k samostudiu. Student se učí, kdy uzná za vhodné. Dále si sám přizpůsobí dobu trvání výuky a obtížnost probírané látky. Asynchronní e-learning je už delší dobu hojně využíván ve firmách ke vzdělávání zaměstnanců. V takovém prostředí totiž bývá synchronní varianta ekonomicky nákladnější a u některých firem nemožná. Je potřeba, aby zaměstnanci, kterých se školení týká ukončili svojí pracovní činnost v době konání výuky. Z tohoto důvodu se ve firmách častěji pro vzdělávání zaměstnanců využívá asynchronního e-learningu. Dobu a čas studia lze přizpůsobit efektivněji každému zaměstnanci dle jeho pracovního vytížení. Ve školách se nejčastěji používá kombinace obou těchto přístupů k elektronické výuce.

Tento kombinovaný přístup se nazývá **Blended learning** [2]. Jedná se o přístup výuky, který je efektivní například při výuce cizích jazyků. Studenti si ve svém volném čase procvičují v aplikaci slovní zásobu a gramatiku. Poté v různých intervalech absolvují klasické synchronní kurzy s lektory a rodinnými mluvčími.

Obecně základem e-learningu je server na kterém běží **LMS (Learning Management system)** což je centrální aplikace e-learningového systému. Typickým příkladem LMS řešení jsou **IBM Lotus Learning Space, WebCT, iTutor, uLern** a v neposlední řadě opensource řešení **Moodle**. LMS umožňuje distribuovat studijní materiály pro výuku studentů a dále testovat jejich znalosti. Testování znalostí

probíhá nejčastěji prostřednictvím interaktivních testů kombinujících různé druhy otázek. Nedílnou součástí takových systémů jsou odpovídající statistiky výuky a výukových testů. Někdy bývá do LMS také začleněn systém pro komunikaci. Samotný LMS spolu se serverem na kterém běží jsou nejdražší položky při implementaci e-learningu. To může být limitující pro menší firmy či školy. Pro takové typy zákazníků se v poslední době objevují řešení běžící v cloudu, které si lze pronajímat jako službu. Lze tak za relativně malé náklady zprovoznit e-learning i pro menší počet studentů.

Dnešní generace dětí jsou již od útlého věku konfrontovány s moderními technologiemi, které každý den používají a jsou nuceny zpracovávat velké množství informací. Způsob jejich myšlení je tedy odlišný od dětí v minulosti. Škola jako taková by měla reflektovat dění v reálném světě a děti do života v takovém světě připravit. Proto se zdá evoluce ve školství směrem k moderním technologiím a e-learningu na všech úrovních vzdělávacího systému nevyhnutelná. Jako ideální prostředek pro výuku se začínají jevit tablety. Navíc již dnes levné tablety dosahují dostatečné výdrže a výkonu potřebného pro výuku.

V dnešní době se výukové materiály resp. výuka samotná pomalu přesouvají do elektronické podoby. Například Jižní Korea jako první země na světě v roce 2015 plánuje zrušení veškerých papírových učebnic a výukové materiály budou mít studenti už pouze v elektronické podobě na tabletech a jiných mobilních zařízeních [3]. Tato země dlouhodobě patří mezi technologické lídry udávající směr, ale je více než pravděpodobné, že vyspělé země budou v blízké době takové kroky následovat. Využití elektroniky ve výuce přináší velké množství výhod a inovací. Například u elektronických knih jsou již standardem multimediální prvky, které mají z hlediska názornosti vykládané látky velké edukační účinky. Objevují se ale také negativní názory na tento trend, který ve školství nastává. Odpůrci nejčastěji argumentují pořizovací cenou a nepříznivým zdravotním vlivem. Praxe ale ukazuje, že klady jednoznačně převažují nad zápory elektronické výuky.



Zajímavý výzkum v této oblasti provedla britská agentura **Family Kids and Youth** na vybraných středních školách ve Velké Británii. [4]

Závěry tohoto výzkumu:

- **90%** učitelů se domnívá, že moderní technologie zvyšují zájem studentů o výuku.
- **91% studentů, 88% rodičů a 74% učitelů** chce, aby se tablety staly běžnou součástí výuky na středních školách.
- **71%** učitelů tvrdí, že tablety do jejich výuky patří, ale je vyžadováno speciálních školení, aby byli schopni výuku udělat více interaktivní a zajímavou.
- **64%** učitelů si myslí, že při použití moderních technologií ve výuce budou mít jejich studenti lepší prospěch.
- **72%** učitelů uvedlo, že by měli zájem o prohloubení svých znalostí moderních technologií. Domnívají se, že je nedostačující.
- **53%** studentů tvrdí, že mají v oblasti internetu a technologií více znalostí než jejich rodiče.
- **44%** studentů se domnívá, že jsou z hlediska IT více vzdělaní než jejich učitelé.
- **51%** učitelů potvrdilo, že s tabletem umí lépe zacházet jejich studenti.

Z tohoto průzkumu tedy vyplynulo, že již v dnešní době mají tablety ve výuce svůj přínos a místo. Na druhou stranu je patrné, že zejména učitelé nemají potřebné znalosti pro takovou výuku. Zdá se to tedy jako jedno z úzkých hrdel pro hromadné rozšíření. Moderní trendy ve výuce jsou prospěšné, motivují studenty k učení a obohacují učební procesy. Mají velký prospěch jak studentům, tak i učitelům. Kopírují současný životní styl.

Podle odborníků jsou tablety vhodné pro výuku na všech úrovních vzdělávání. Například i v mateřských školách s aplikacemi pro děti předškolního věku.

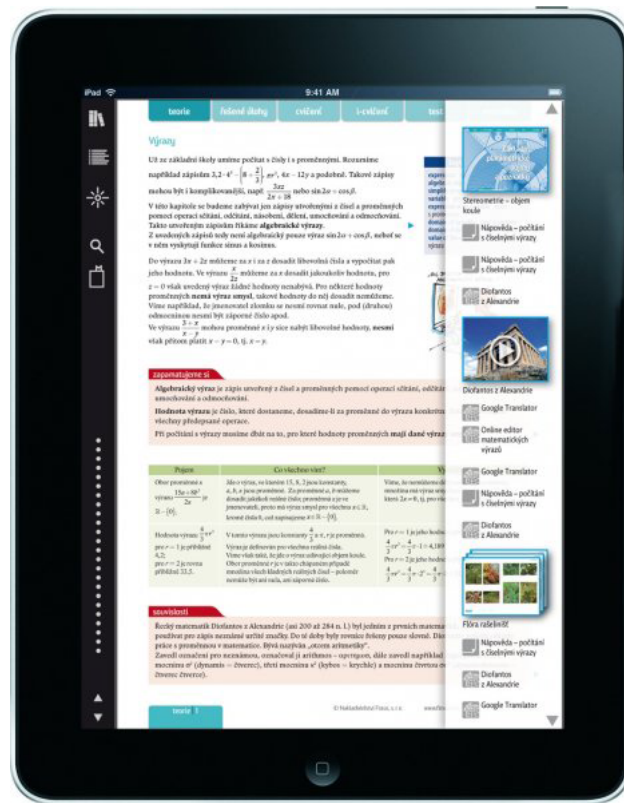
Shrnutí výhod využití tabletů ve výuce:

- **Motivace** - díky moderním technologiím mají studenti více možností projevit svoje individuální schopnosti. K tomu do velké míry přispívá, že každý student může pracovat vlastním tempem a mít individuální zadání. S učitelem může komunikovat online a vypracovávat zadané úlohy.
- **Objektivita** - opravování prací studentů a jejich známkování je díky technologiím rychlejší a více objektivní. Odpadá subjektivní zaujatost, která někdy může nastat.

- **Zpětná vazba** - velmi důležitý aspekt je zpětná vazba, která dovoluje zkvalitnit a zrychlit proces výuky. Učitel může rychleji a přesněji reagovat na aktuální situaci a přizpůsobit tak svůj výklad.
- **Úspora času učitele** - jednodušší příprava podkladů k výuce a rychlejší vyhodnocování testů.
- **Úspora času studentů** - při psaní poznámek. Mohou se tak více zaměřit na samotné učivo.
- **Okamžité sdílení obsahu** - zrychluje výuku a dělá jí více efektivní.

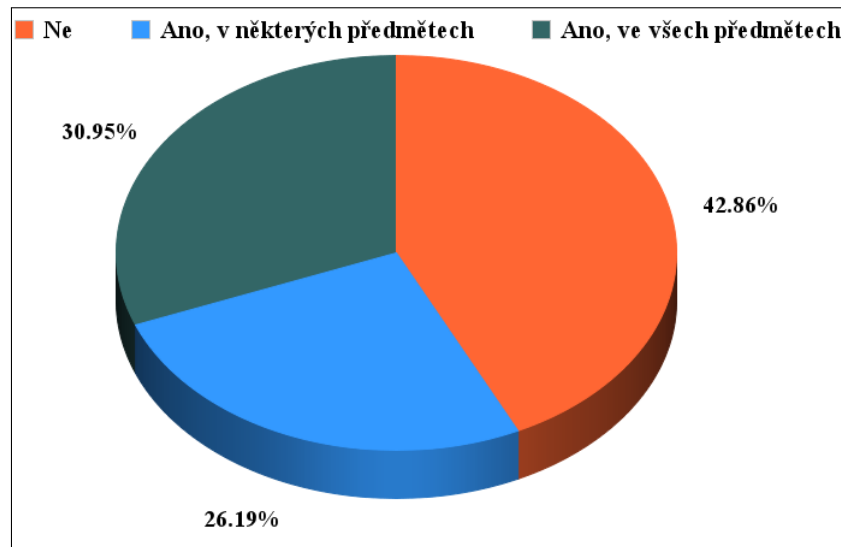
## 2.2 Příklady systémů

V České Republice se už objevují první pilotní projekty a průzkumy. Jedním z nich je například projekt Flexibook 1:1 který byl proveden pod záštitou nakladatelství **Fraus**. [2] Tento projekt byl proveden ve školním roce 2012/2013 a jeho cílem bylo, aby v několika vybraných základních a středních školách probíhala výuka elektronickou formou (viz obr. 2.1). Papírové učebnice nahradily tablety, do kterých si žáci mohli stahovat elektronické učebnice s podporou interaktivních prvků a videí obohacujících výuku.



Obrázek 2.1: Elektronická učebnice nakladatelství Fraus.

Pilotní projekt měl zejména ukázat výhody a nevýhody užití tabletů ve výuce. V jeho první fázi se ho zúčastnilo 16 škol a studenti využívali tuto formu výuky ve třech vyučovacích předmětech. Po ukončení pilotního projektu Flexibook 1:1 byla úspěšnost zjišťována dotazníkovou formou. Výstupem bylo, že pro více jak 3/4 vyučujících splnila výuka s pomocí elektronických materiálů očekávání (viz obr. 2.2). Tento systém není klasickou e-learningovou aplikací, jedná se pouze o authoring aplikaci pro výukové materiály a knihy.



Obrázek 2.2: Jak z pohledu učitelů splnila elektronická učebnice jejich očekávání?

Dalším hlavním ukazatelem tohoto projektu bylo zjištění, že lze během jednoho školního roku přejít z klasických tištěných učebnic na učebnice elektronické při zachování kvality a smysluplnosti výuky. Naopak při očekávaném zvýšení interaktivity těchto učebnic se kvalita výuky zlepšila.

Nutno dodat, že většina učitelů, kteří byli s elektronickou výukou spokojeni a doporučili by jí i dalším kolegům, má kladný vztah k technologiím. Projekt tedy ve svém závěru poukazuje, že největšími překážkami jsou finanční aspekt na pořízení tabletů a poté negativní postoj některých učitelů majících špatný vztah k moderním technologiím.

Dá se tedy očekávat pouze pozvolný nástup elektronické výuky v českém školství.

Začínají se u nás ale také prosazovat velcí a renomovaní výrobci přinášející svá vlastní řešení. Například jihokorejská společnost **Samsung** nabízí své řešení elektronické výuky s názvem **Samsung Smart School** [5]. Tento projekt interaktivní třídy je koncipován tak, že každý student má vlastní tablet, na něm textové i multimediální výukové materiály a speciální aplikaci na výuku. Součástí je interaktivní tabule na které učitel prezentuje obsah a píše poznámky. Tyto poznámky jsou přenášeny i do tabletů studentů. Zároveň má učitel možnost kontrolovat vzdáleně tablety všech studentů ve třídě. Má tak kontrolu, zdali student dává pozor, nebo v případě problémů může studentovi pomoci. Součástí takové aplikace je kromě authoring obsahu samozřejmě také testovací e-learningová část, ve které učitel může dávat studentům testovací otázky. Může tak zefektivnit celou výuku.

Například během výkladu po určitých úsecích otestuje na několika otázkách studenty a dostane se mu okamžité zpětné vazby, zdali studenti látku pochopili. Díky

takové zpětné vazbě může interaktivně reagovat a přizpůsobit tempo výkladu dané látky. Aby studenty nelákalo na tabletu provádět jiné úkony nesouvisející s výukou, má učitel také možnost tablety studentům uzamknout, na uzamčeném tabletu pak student vidí pouze aplikaci pro výuku. Pro názornost výuky lze také na interaktivní tabuli zobrazit obsah libovolného tabletu ve třídě a pokusit se vysvětlit, v čem daný student chyboval. Další zajímavostí je funkce **group activity**, kde učitel může studenty rozdělit do skupinek, které navzájem spolupracují na zadaném úkolu.

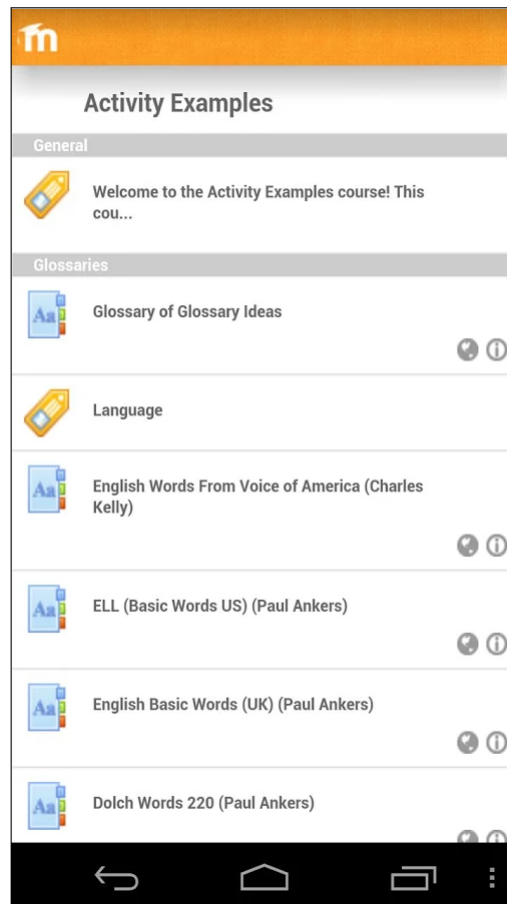


Obrázek 2.3: Samsung smart school.[5]

Většina kvalitních e-learningových systémů je placená. Najdou se však i výjimky, jednou z nich je **LMS Moodle** [6]. Jedná se o open source e-learningový systém, jehož hlavním tvůrcem je Australan Martin Dougiamas. Zejména ve školství je nasazení volně šiřitelného LMS systému jedinou možností k provozování e-learningu ve výuce. I proto si ho oblíbilo velké množství středních i vysokých škol v České republice (VUT, ČVUT, Univerzita v Hradci Králové).

Systém podporuje také českou lokalizaci. V Moodle lze provádět úkony jako v podobných placených systémech, tedy vytvářet online testy, úkoly, tvorba studijních materiálů, monitorovat výstupy a výsledky, nahrávat dokumenty a další výukový obsah. Z pohledu tvorby výukových článků lze vytvářet v plnohodnotném editoru text, který lze dále dělit na logické celky. V základní verzi je ale minimální podpora pro export těchto materiálů.

Moodle je navíc plně modulární, lze tedy sestavit systém podle potřeb dané školy, nebo doinstalovat další rozšiřující moduly. Podle statistik je na světě více než sto tisíc instalací LMS Moodle, což zaručuje velkou komunitu pro případ nutnosti řešení nějakých problémů.



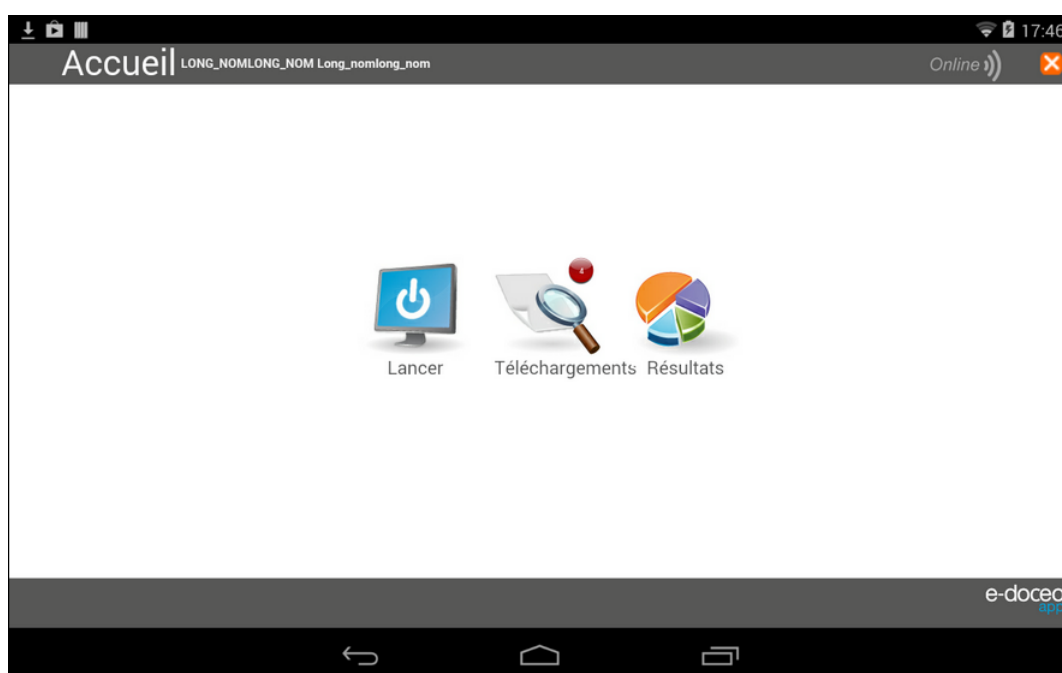
Obrázek 2.4: Android aplikace Moodle.

Z hlediska podpory mobilních zařízení nabízí poslední verze Moodle zlepšení. Web je plně responzivní a přizpůsobený pro použití na tabletech nebo mobilních telefonech. Existují také mobilní aplikace pro Android (viz obr. 2.4), iOS a Windows Phone. Ty jsou však v rané fázi a dle hodnocení uživatelů vykazují vysokou chybovost a neodladěnost.

Posledním zkoumaným řešením byl komerční LMS **eDoceo** [7]. Jedná se o klasický e-learningový systém mířící zejména na firemní klientelu. Umožňuje správu výukových testů a materiálů. V rámci celého systému je propracována funkcionality zpětné vazby na výukový obsah a komunikace mezi studenty a učiteli.

Poskytuje také moduly pro integraci s provozovanými ERP systémy ve firmách. Z pohledu výuky lze využívat synchronní i asynchronní přístup. Vše je plně lokalizované do češtiny a tento komerční systém používá v České republice velké množství známých firem a státních institucí. Pro platformy Android a iOS existuje pro tento systém také mobilní aplikace viz obr. 2.5.

Jedná se o jednoduchou aplikaci, která je stále ve fázi vývoje.



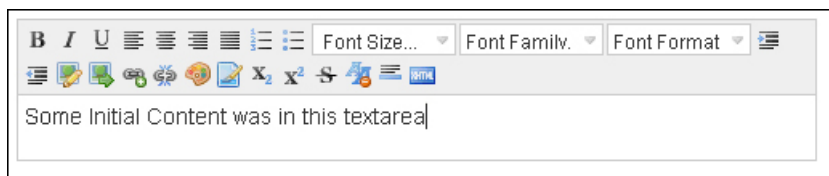
Obrázek 2.5: Android aplikace eDoceo.

## 3 Analýza

V předchozí kapitole byly prozkoumány dostupné vybrané systémy, které slouží k elektronické výuce a tvorbě výukového obsahu. Na základě tohoto průzkumu byly určeny funkce, které by výsledný systém měl obsahovat. V této kapitole jsou pak analyzovány technologie pro implementaci požadovaných funkcí systému.

### 3.1 Výukový obsah

Z průzkumu e-learningových systémů vyplynulo, že všechny aplikace, které podporují tvorbu výukového obsahu poskytují uživatelům poměrně vysoký komfort při tvorbě a editaci. Ten je zajištěn pokročilými textovými procesory implementovanými v aplikaci. Tyto procesory se obecně nazývají **WYSIWYG** editory. Pro použití ve webové aplikaci existuje velké množství hotových knihoven. Až na výjimky se jedná o knihovny napsané v jazyce **JavaScript**, které se namapují na formulářový prvek typu **textarea**. Takto rozšířené pole pro zadávání textu pak dostane funkcionalitu pro pohodlnou tvorbu formátovaného dokumentu viz obr. 3.1. Výstupem je zpravidla text strukturovaný a formátovaný v HTML a CSS. Jedná se tedy o textovou reprezentaci dat, kterou je snadné ukládat v databázi a na rozdíl od dat binárních je lze pohodlně zobrazovat či transformovat.



Obrázek 3.1: Ukázka WYSIWYG editoru NicEdit.

Mezi nejrozšířenější editory patří **TinyMCE** [8], **NicEdit** [9] a **CKEditor** [10]. Všechny zmíněné editory jsou volně šiřitelné JavaScript knihovny. Mezi hlavní výhody patří jejich velká rozšířenost a uživatelská komunita. Všechny také podporují rozšiřující moduly v případě, že si uživatel nevystačí se základní funkcionalitou editoru.

Kromě samotné tvorby článků by měl mít uživatel možnost nahrávat již hotové výukové materiály (nejčastěji dokumenty). V případě nahrávání dokumentů na server lze využít dvou přístupů uložení dat. Prvním z nich je ukládat dokumenty na souborový systém serveru, nebo ukládat data do databáze jako **BLOB** objekty. Oba přístupy mají své výhody a nevýhody. Vhodnost použitého přístupu je závislá na konkrétní aplikaci a povaze dat.



Ukládání do databáze má nespornou výhodu v tom, že je možné jednoduše zálohovat a obnovovat data k určitému datu. Nicméně u systémů s velkým objemem dokumentů dojde k výraznému zatížení databáze a případná obnova je zdlouhavý a náročný proces.

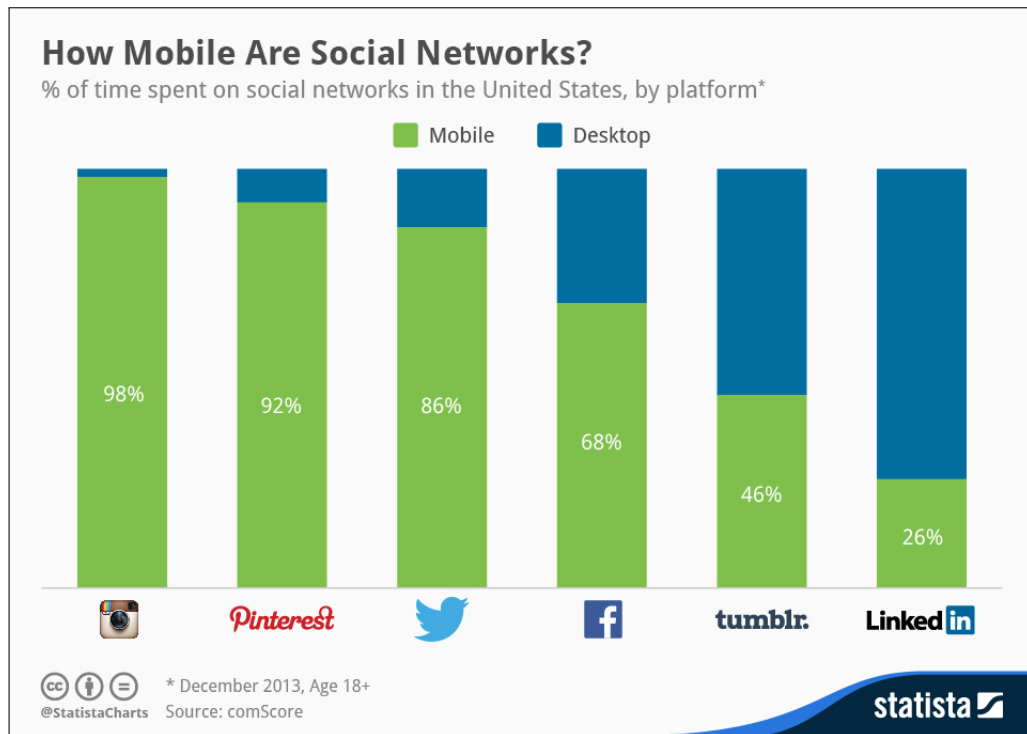
Ukládání na souborový systém má problémy s transakcí, špatně se zálohuje a obnovuje. Nicméně v dnešní době jsou u dobře konfigurovaných serverů zálohy souborového systému řešeny. Zůstává tedy pouze problém s konzistencí ukládaných souborů. To je třeba řešit zejména v případě, pokud se nahrávají nějaká unikátní, nebo kritická data.

## 3.2 Podpora mobilních zařízení

Mezi základní požadavky na výsledný systém je jeho multiplatformnost s ohledem na použití v mobilních zařízeních. Zkoumané systémy většinou nabízejí web s podporou mobilních zařízení, nebo aplikace pro nejrozšířenější mobilní platformy.

Dříve se u webových stránek předpokládalo, že si je uživatelé budou prohlížet prostřednictvím obrazovky počítače. Při tvorbě stránky se tedy vývojáři zaměřovali hlavně na to, aby web vypadal vždy stejně. Optimalizovalo se zejména pro všechny používané internetové prohlížeče.

V posledních letech je velký rozmach nejrůznějších zařízení, prostřednictvím kterých uživatelé surfují na internetu. Patří mezi ně mobilní telefony, tablety, chytré televizory a čtečky elektronických knih. Dochází tak k velké fragmentaci velikostí obrazovek a jejich rozlišení, se kterými je třeba při vývoji moderního webu počítat. Některé velké weby a sociální sítě ukazují, že počet návštěvníků jejich webu z mobilních zařízení již předčil klasické počítače. Například na sociální síť **Twitter** přistupuje více jak 80% uživatelů přes mobilní zařízení [11]. Neméně zajímavé statistiky přináší také sociální síť Facebook. Ta měla za druhé čtvrtletí roku 2014 více jak 60% svých příjmů z mobilní reklamy.[12]. **Facebook** také na začátku roku 2014 překonal 1 miliardu aktivních uživatelů na mobilních zařízeních[13]. Přes 30% uživatelů dokonce na tuto sociální síť přistupuje pouze přes mobilní telefony. Detailní graf poměru mobilních zařízení vůči počítačům viz obr. 3.2.



Obrázek 3.2: Využití mobilních zařízení na sociálních sítích.[14]

Je tedy třeba se na mobilní zařízení při vývoji zaměřit. Z tohoto důvodu se začíná prosazovat filosofie vývoje **mobile first**[15], kdy se řeší design webu s ohledem na mobilní zařízení a pak až teprve pro počítačové prohlížeče. Dostáváme tak úplně opačný pohled na webdesign, než jsme byli doposud zvyklí.

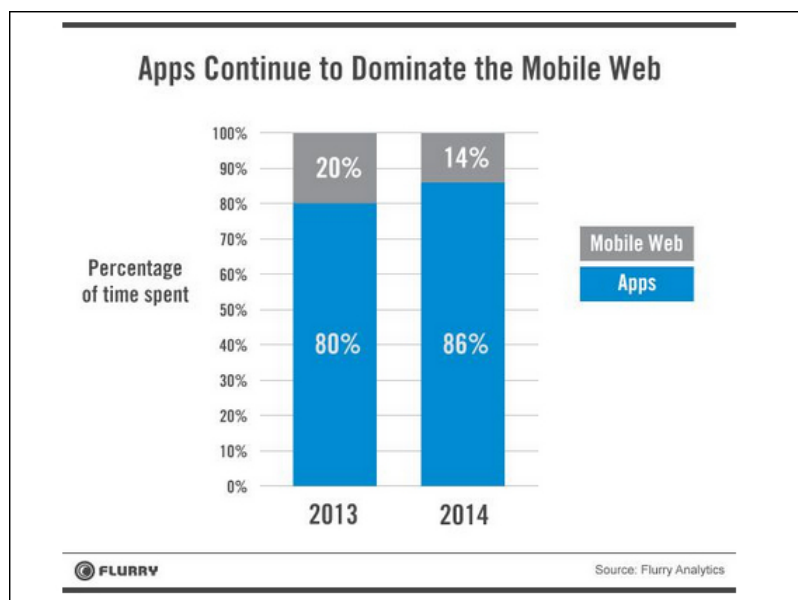
Pro mobilní zařízení je potřeba přizpůsobit nejen rozložení prvků na stránce, ale také jejich velikost. Zatímco při klasickém ovládní přes počítačovou myš si můžeme dovolit použít poměrně malé ovládací prvky, tak u zařízení s malým dotykovým displejem musí být prvky výrazně větší kvůli menší přesnosti ovládní dotykem.

Z pohledu vývoje mobilních webových aplikací existují dva přístupy. Jedním z nich je tvorba dvou oddělených webů. Jeden pro klasické obrazovky počítačů a druhý pro mobilní zařízení. Mezi výhody tohoto staršího přístupu k vývoji webu patří zejména dobrá možnost přesné optimalizace mobilní verze, díky oddělení lze omezit, jaký obsah se na jaké verzi webu bude zobrazovat a uživatel má možnost přepnutí mezi verzemi. Mezi nevýhody patří složitější udržitelnost vývoje. Je potřeba spravovat dva weby. Dalším problémem jsou zejména tablety, ty patří mezi mobilní zařízení, ale mají větší displeje s vysokým rozlišením. Pro ně se většinou nehodí mobilní verze z důvodu neoptimálního využití místa, které zařízení poskytuje, ani klasická verze pro počítače, protože ta nemá ovládací prvky uzpůsobené pro dotykové ovládní. Možným řešením by byla tvorba třetí verze webu.

Druhým přístupem při tvorbě mobilního webu je využití **responzivního designu**. Jedná se o novější způsob vývoje. Zde se neprovádí implementace více webů pro různá zařízení, ale pouze jednoho webu, který se dynamicky přizpůsobí. Lze tedy říci, že responzivní web se na zařízeních s různou velikostí obrazovky zobrazuje jinak. Mezi výhody patří levnější vývoj, lepší udržitelnost a širší záběr podporovaných zařízení. Mezi nevýhody patří, že se pro mobilní zařízení přenáší méně dat než pro plnohodnotný web, ale z pravidla více dat, než u mobilních verzí webu využívajících předchozí přístup. Další nevýhodou je, že se obvykle uživatel nemůže přepnout do klasické plnohodnotné verze webu [16].

V dnešní době se z pohledu vývoje moderních aplikací vedou spory, zdali je lepší pro mobilní zařízení vytvářet responzivní web, či nativní aplikaci [17]. Oba přístupy mají své klady a zápory, nejlepším řešením je v případě finančních prostředků kombinovat oba přístupy. Pro majoritní platformy vytvořit nativní aplikace. Responzivním webem pak pokrýt minoritní mobilní platformy, nebo příležitostné návštěvníky, kteří si nechtějí instalovat aplikaci.

Uživatelé si na mobilních zařízeních zvykli používat nativní nainstalované aplikace, ty jim na rozdíl od webu přinášejí standardizované ovládací prvky pro danou platformu a větší komfort ovládání. Stále větší oblíbenost aplikací před mobilními weby dokazují statistiky. Například v minulém roce více jak 86% uživatelů preferovalo na svém mobilním zařízení nativní aplikaci před webem viz obr. 3.3.



Obrázek 3.3: Graf použití nativních mobilní aplikací a mobilních webů.[18]

### 3.3 Aplikační rozhraní

V dnešní době ve většině případů nelze dělat izolované systémy tak jako dříve. S novou generací webových aplikací je již standardem propojení s dalšími aplikacemi a systémy. Díky takovému vývoji se aplikace stávají více interoperabilní a znovupoužitelné [19]. Pro takový druh integrací je potřeba aplikačních rozhraní, ty se obecně nazývají **webové služby**. Mezi nejpoužívanější standardy pro tvorbu webových služeb lze považovat služby typu **REST** a typu **SOAP**.

SOAP (Simple Object Access Protocol) je hojně používaným protokolem webových služeb založeným na XML formátu. Existuje pro něj velké množství knihoven a nástrojů. Odesílané zprávy musí být v požadovaném formátu, který je popsán ve **WSDL** souboru služby prostřednictvím speciálního jazyka pro popis dané služby a funkcí, které poskytuje. Pro samotné odeslání dat lze využít libovolného protokolu aplikační vrstvy, který taková data dokáže přenášet. Nejčastěji se však využívá protokolu HTTP. Díky datům formátovaným v XML lze velmi přesně transformovat přijatá data do potřebného formátu v cílové aplikaci. Problém ale nastává při posílání velkého množství dat. Zpracování velkých XML souborů je strojově náročná operace. Navíc XML formát přináší nevýhodu velkého množství redundantních popisujících dat. Protokol tedy není šetrný ani na kapacitu přenosového kanálu. Mezi hlavní výhody patří snadná validace dat.

Zkratka REST znamená Representational State Transfer. Jedná se o poměrně starou techniku využívající funkcí, které poskytuje protokol **HTTP**. Poprvé byl tento druh webových služeb představen v disertační práci Roye Fieldinga [20]. Oproti službám typu SOAP lze hovořit o REST jako o lehkých službách. Využívají elementárních operací HTTP protokolu pro přenos dat bez velkého množství redundantních informací. V poslední době se začíná tento druh služeb mnohem více využívat na úkor SOAP právě díky jeho jednoduchosti a nenáročnému zpracování dat. Hlavním důvodem je použití na mobilních platformách a na *front end* části webových aplikací, kde se o zpracování dat stará nejčastěji kód napsaný v jazyce JavaScript. Nevýhodou je však horší popis dat a dokumentace samotné služby.

## 4 Seznam realizovaných funkcí

Na základě průzkumu e-learningových systémů a analýzy byla provedena samotná implementace. Vzhledem k tomu, že tato práce rozšiřuje již funkční systém Teri, tak zde popíši realizované funkce, a pro porovnání i původní funkcionalitu, která nebyla implementována v diplomové práci.

Původní systém byl vytvořen v rámci méj bakalářské práce. Jeho hlavním cílem bylo vytvořit nový multiplatformní výukový portál. Ten měl sloužit pro testování znalostí studentů. Podařilo se tedy implementovat funkční testovací platformu, která byla nasazena v reálném provozu ve výuce.

Seznam hlavních funkcí původního systému [21]:

- Správa uživatelů a jejich rolí.
- Modul pro testování znalostí studentů (výukové testy).
- Část webu pro diskuzi uživatelů.
- Modul ostrých testů pro závazné testování studentů.
- Mobilní verze webu.
- Správa předmětů.
- Vzkazy studentům.
- Export a import otázek.
- Tvorba tisknutelných verzí testů.
- Základní statistiky výuky.
- Nativní aplikace pro mobilní zařízení s operačním systémem Android.
- Klient pro zařízení podporující J2ME.

Vzhledem k tomu, že problematika e-learningu zahrnuje mnohem více oblastí, než jen samotné testování znalostí studentů, tak byl v rámci této práce celý systém rozšířen. Stal se tak mnohem více komplexním než původní verze. Ta byla z větší části přepracována a rozšířena o nové moduly. Výsledný systém je tedy mnohem více použitelný pro reálnou výuku. Byl také značně technologicky modernizován. V původní práci byly vytvořeny mobilní aplikace. V rámci diplomové práce byla vytvořena kompletně nová nativní aplikace pro operační systém Android. Ta nevyužívá

žádné zdrojové kódy z aplikace původní. Jedná se tedy o kompletně novou aplikaci nahrazující původní verzi.

Seznam hlavních funkcí, které byly implementovány v této práci:

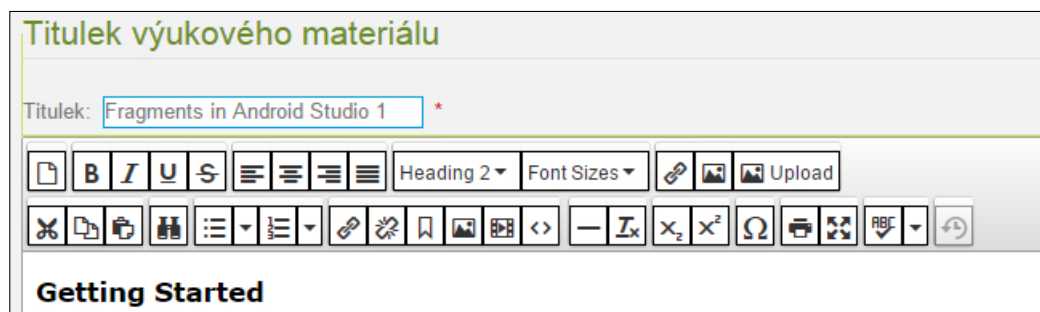
- Authoring modul pro tvorbu výukového obsahu. Možnost čtení výukových článků doplňuje testování a tím jsou lépe pokryty možnosti e-learningu (více v kapitole 5.1).
- Modul pro zpětnou vazbu studentů.
- Systém pro nahrávání příloh a materiálů souvisejících s výukou. Učitelé tak mohou nahrávat již hotové materiály, které chtějí poskytnout studentům.
- Webová služba pro synchronizaci dat s mobilní aplikací (více v kapitole 5.2).
- Přepracovaný a rozšířený modul statistik výuky. Učitelé nyní dostanou přehlednější statistiky svých předmětů doplněné o statistiky výukových článků (více v kapitole 5.3).
- Rozšíření ostrých testů o pokročilé možnosti tvorby, editace a vyhodnocení. Nový modul ostrého testování je nyní komplexním nástrojem pro reálné využití při výuce (více v kapitole 5.5).
- Rozšíření výukových testů o nový druh interaktivní otázky (více v kapitole 5.4).
- Kontextová nápověda. Ta umožňuje rychlé informování uživatele v kontextu dané stránky. Uživatel není nucen číst návod pro klíčové funkce (více v kapitole 5.6).
- Kompletní *redesign* celé webové aplikace. Byla odstraněna původní mobilní verze webu a celý systém byl implementován jako responzivní (více v kapitole 6.1).
- Drobečková navigace. Umožňuje snadného procházení i při hlubším zanoření v hierarchii webu (více v kapitole 5.22).
- Hromadný import studentů. Vyučující má díky hromadnému importu možnost hromadného vytvoření účtů pro studenty svého předmětu (více v kapitole 5.8).
- Skrytí předmětu. Lze tak ovlivňovat viditelnost předmětu pokud vyučující nemá připraveny všechny materiály, nebo je předmět v neaktuálním stavu (více v kapitole 5.9).
- Nová nativní aplikace pro mobilní zařízení s operačním systémem Android umožňující čtení výukového obsahu a testování znalostí studentů (více v kapitole 6.2).

## 5 Webová verze systému

Hlavní částí celého systému je webová aplikace. Studenti si zde mohou testovat své znalosti a využívat výukových materiálů. Nachází se zde také administrace pro učitele. V této kapitole jsou popsány hlavní části, které byly v rámci této práce ve webové aplikaci implementovány.

### 5.1 Authoring modul

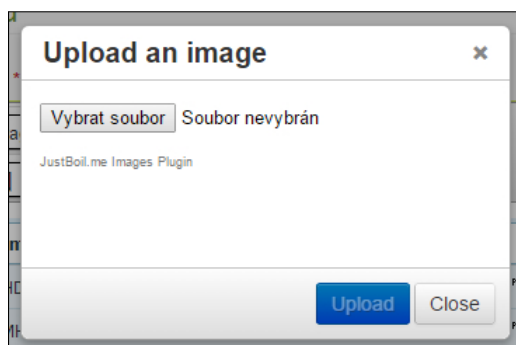
Jedním z hlavních modulů, který byl implementován, je modul pro tvorbu výukového obsahu. Ten učitelům umožňuje vytvářet výukové články a nahrávat studijní materiály. V administraci lze psát články v plnohodnotném textovém procesoru viz obr. 5.1.



Obrázek 5.1: Editor pro tvorbu výukových článků.

Pro implementaci textového procesoru byl zvolen WYSIWYG editor **TinyMCE** ve verzi 4.1.9 [8]. Jedná se o JavaScript *open source* knihovnu pod licencí LGPL. Tato knihovna patří mezi nejrozšířenější webové editory, proto má velkou komunitu a existuje hodně rozšiřujících modulů. Pro možnost přímého nahrávání obrázků, které se vloží do textu článku byl pak použit zdarma dostupný rozšiřující modul **JBImages** [22]. Ten umožní pohodlné vkládání obrázků do tvořeného obsahu viz obr. 5.2. Bez jeho použití by učitel musel obrázky sám nahrávat na server a v článku použít pro zobrazení jejich odkazy.

Výstupem TinyMCE je článek formátovaný jako HTML dokument. To usnadňuje práci při procesu zobrazení článku. V další fázi vývoje se HTML struktura článků osvědčila i pro zobrazení v nativní aplikaci pro mobilní telefony s operačním systémem Android.



Obrázek 5.2: Ukázka modulu pro nahrávání obrázků do článku.

Při tvorbě výukového článku jsou učitelé také nabízena rozšiřující nastavení viz obr. 5.3. Zde může zvolit, aby se článek zobrazoval pouze přihlášeným studentům, časovou viditelnost článku, možnost exportu článku do formátu PDF a funkci zpětné vazby studentů.

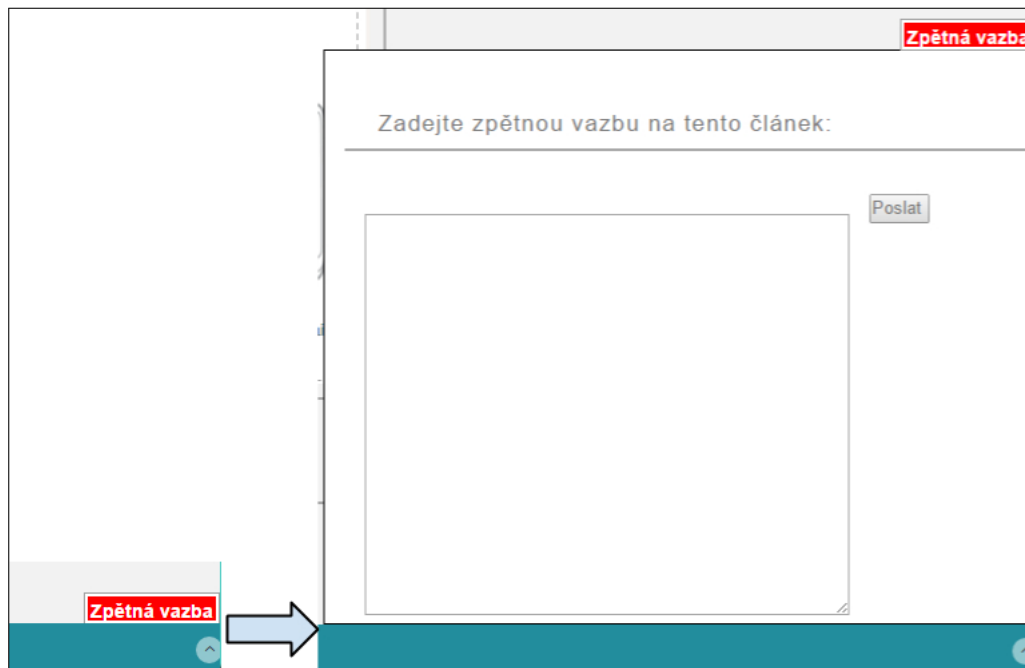
Obrázek 5.3: Rozšířené nastavení při tvorbě článku.

Při aktivní volbě exportu článku do PDF se při prohlížení studentům zobrazí odkaz, který umožní daný obsah vyexportovat. Při exportu je využíváno volně šiřitelné PHP knihovny **mPDF** [23]. Tato knihovna umí vytvářet PDF dokumenty z dat strukturovaných v HTML. Díky této funkci si studenti mohou pohodlně exportovat veškerý studijní obsah. Výsledná PDF také mají vhodný formát pro tisk. To vše může být výhodou i pro učitele. Ten si takto může jednoduše uložit své články a použít je i mimo systém Teri.

Další zajímavou funkcí je možnost zpětné vazby na článek. Pokud se učitel rozhodne tuto možnost studentům povolit, tak se přihlášeným studentům zobrazí při prohlí-



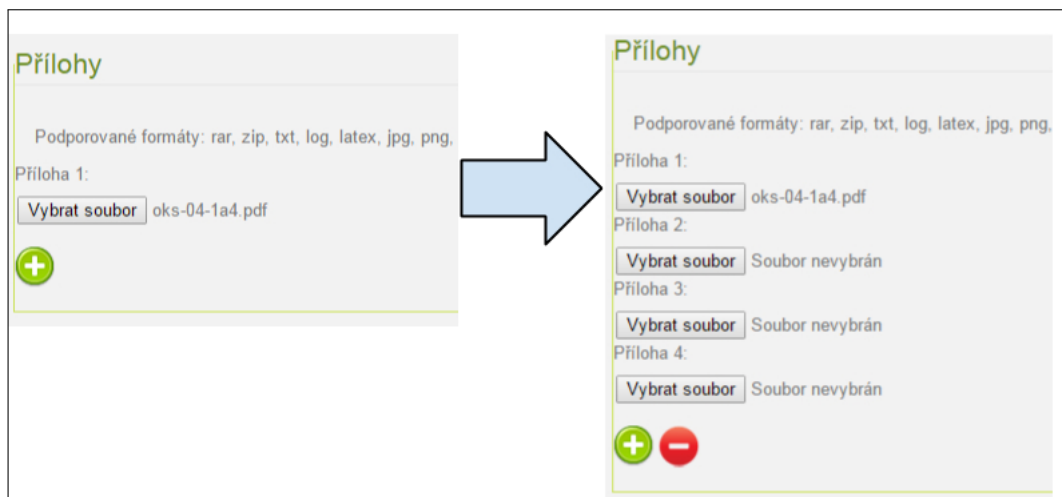
žení článku v pravém dolním rohu tlačítko pro zpětnou vazbu. Po kliknutí na toto tlačítko se rozbalí formulář pro napsání zpětné vazby viz obr. 5.4. Tato zpětná vazba přijde anonymně na e-mail učitele, který článek napsal. Kromě textu napsaného studentem učiteli také přijde informace, k jakému článku byl příspěvek poslán. Aby se při odeslání formuláře stránka nepřekreslovala, tak bylo použito funkce **AJAX** kterou poskytuje knihovna **jQuery** [24], ta dokáže poslat asynchronní POST požadavek na pozadí pomocí jazyka JavaScript. Efekt vyjíždějícího formuláře byl docílen funkcí **slideToggle** obsaženou v knihovně **jQuery UI** [25].



Obrázek 5.4: Rozbalovací formulář pro zpětnou vazbu.

Obdobné asynchronní volání technikou AJAX je použito také pro panel hodnocení článku viz obr. 5.12. Ten se zobrazuje všem čtenářům a dovolí článek kladně, či záporně ohodnotit. Po ohodnocení panel pro daný článek zmizí. Po kliknutí na tlačítko hodnocení se provede asynchronní dotaz na skript, který zaznamená uživatelské hodnocení článku do databáze.

Poslední funkcionalitou při tvorbě článku je přidávání příloh. Učitel tak může na server nahrávat přílohy obsahující studijní materiály. Ty se pak zobrazí ke stažení u daného článku. Vzhledem k tomu, že není předem známý počet příloh, které budou nahrávány, tak byl naprogramován dynamický formulář viz obr. 5.5. Na základě analýzy bylo rozhodnuto, že implementovaný formulář ukládá dokumenty na souborový systém serveru. Hlavním důvodem je snadnější přístup a menší režie. Nevýhoda spojená s možností přerušování transakce při nahrávání dat v našem případě není kritická. Uživatel se dozví, že se data nepodařilo nahrát a akci provede znovu.



Obrázek 5.5: Dynamický formulář pro nahrávání příloh.

Tento formulář funguje tak, že tlačítka se symboly plus a mínus lze přidávat další položky formuláře pro nahrání souboru. Uživatel si tedy sám určí počet nahrávaných souborů. Kód pro tento formulář byl napsán v jazyce JavaScript:

```

1 <script type='text/javascript'>
2 $(document).ready(function() {
3     // zacina od 2. 1 je rezervovan, uzdy zobrazen alespon jeden
4     var counter = DEFAULT_INDEX;
5     // pri zobrazeni jednoho prvku je tlacitko minut skryto
6     $('#del_file').hide();
7     // klik na ikonku plus - pridani dalsiho souboru
8     $('#img#add_file').click(function() {
9         // prida dalsi html input pro soubor
10        // identifikator je hodnota citace
11        $('#file_tools').before(printFileInput(counter));
12        // zobrazi tlacitko pro minus pro odebirani poctu priloh
13        $('#del_file').fadeIn(0);
14        counter++;
15    });
16    // klik na ikonku minus - odebrani prilohy
17    $('#img#del_file').click(function() {
18        // pokud je mazan posledni mozny input
19        if (counter == LAST_DELETABLE_INDEX) {
20            // schovani tlacitka minus
21            $('#del_file').hide();
22        }
23        counter--;
24        // smaze html input ze stranky
25        $('#f' + counter).remove();
26    });
27 });
28 </script>

```

Takové dynamické přidávání a odebírání prvků formuláře je použito pouze pro přílohy. Vytvořená knihovna by šla v budoucnu ale použít i na jiných místech aplikace a udělat formuláře více škálovatelné.

## 5.2 Webová služba

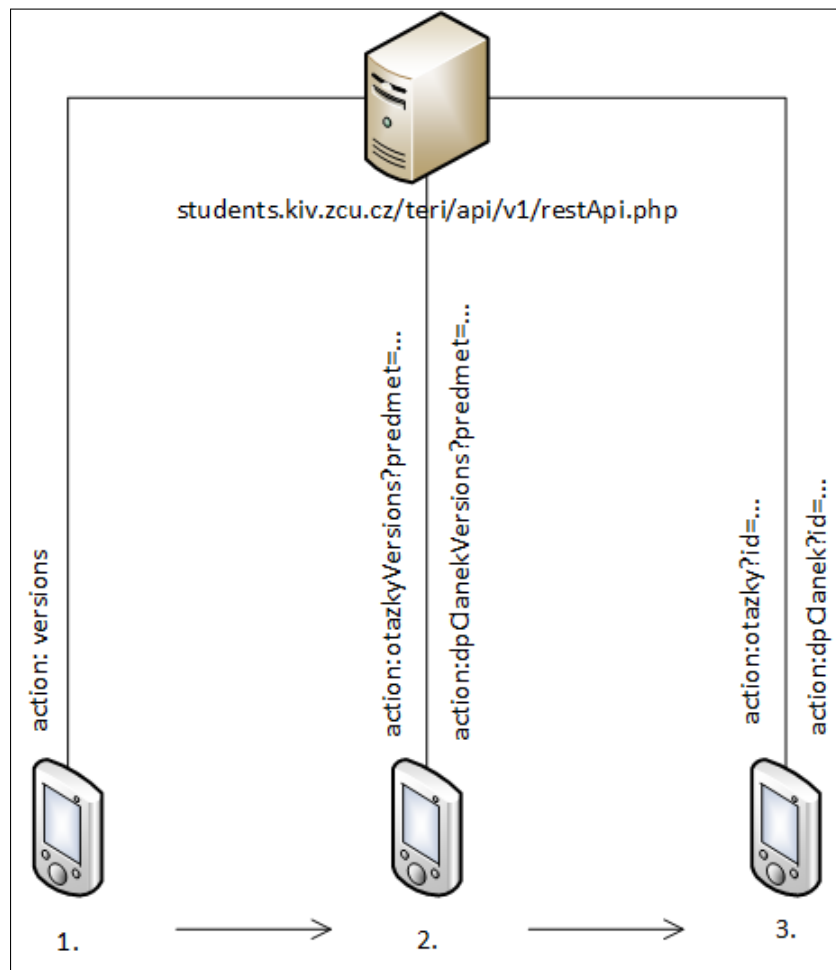
Pro poskytování dat mobilním zařízením byla navržena a implementována webová služba. Mezi nejpoužívanější standardy webových služeb jsou webové služby typu SOAP a REST. SOAP strukturuje posílaná data v XML formátu. Tím je dosaženo dobrého popisu dat, ale je posíláno velké množství redundantních dat a zpracování XML je výpočetně náročné. Oproti tomu jsou REST služby mnohem méně náročné na zpracování (lightweight služby) a vhodnější pro zpracování na méně výkonných zařízeních, jako jsou například mobilní telefony. Více informací naleznete v kapitole 3.3.

Celý systém Teri je zaměřen na mobilní zařízení, proto byla nakonec implementována webová služba prostřednictvím **REST** služeb. Na první pohled systém Teri z hlediska dat poskytuje pro jednotlivé předměty testovací otázky a výukové články. Pokud bychom stavěli rozhraní pro použití na počítačích připojených do rychlé sítě, pak by byl model aplikačního rozhraní jednoduchý. V našem případě ale bylo API vytvářeno zejména pro mobilní telefony. V dnešní době jsou již mobilní datové přenosy z hlediska rychlosti na dobré úrovni. Stále je však běžné datové omezení **FUP** (Fair User Policy). Z tohoto důvodu bylo API navrženo tak, aby bylo z hlediska množství přenášených dat co nejméně náročné.

Synchronizace dat přes API probíhá ve třech krocích viz obr. 5.6. Nejprve se klient dotazuje na všechny předměty systému. Jako odpověď dostane názvy předmětů a číslo verze. Pokud klient zjistí, že daný předmět nezná, nebo má neaktuální verzi tak pokračuje v druhém kroku. V tom se dotazuje serveru na výukové články a otázky daného předmětu. I v tomto případě se kvůli úspoře dat nepošílají všechna data, ale pouze identifikátory a verze konkrétních článků a otázek. Klient provede porovnání s lokálními daty. Pokud u některého článku či otázky zjistí, že v lokálních datech neexistuje, nebo je zastaralá verze, tak se ve třetím komunikačním kroku dotáže serveru na kompletní data článku nebo otázky podle daného identifikátoru. Takto navržené synchronizační API přenáší minimum zpráv a pouze data, která klient potřebuje. Jednotlivé funkce webové služby jsou uvedeny v tabulce 5.1.

Adresa služby	Parametry	Metoda
/api/v1/restApi.php?action=versions	-	GET
/api/v1/restApi.php?action=otazkyVersions	predmet= <i>název</i>	GET
/api/v1/restApi.php?action=otazky	id= <i>id otázky</i>	GET
/api/v1/restApi.php?action=dpClanekVersions	predmet= <i>název</i>	GET
/api/v1/restApi.php?action=dpClanek	id= <i>id článku</i>	GET

Tabulka 5.1: Popis webové služby



Obrázek 5.6: Schéma komunikace mobilního zařízení se serverem Teri.

Popis jednotlivých služeb:

- **action=versions** tato služba vrací informace o všech předmětech ve tvaru:  
`{"predmet":"KIV TST","version":"342"}`
- **action=otazkyVersions** této službě v GET parametru předáte název předmětu a vrací výpis všech otázek. Opět pouze ve tvaru s názvem otázky a její verzí.
- **action=otazky** v GET parametru se pošle identifikátor otázky a vrátí se všechna data k dané otázce.
- **action=dpClanekVersions** obdoba služby otazkyVersions, akorát pro výukové články.
- **action=dpClanek** v GET parametru se pošle identifikátor článku a vrátí se všechna data k danému článku.

Data, která zasílá server jsou ve formátu **JSON**. Na straně serveru se data uloží do polí a ta jsou posléze zakódována funkcí `json_encode()`.

Ukázka kódu funkce pro získání článků a jejich verzí:

```
1 private function dpClanekVersions() {
2     // získání názvu předmětu
3     $predmet = $_REQUEST[PREDMET];
4     // načtení článku a jejich verzí
5     $versionsArray = DpClanekFacade::findAllByPredmet($predmet);
6     // vytvoření výstupního pole
7     $version_info = array();
8     if (count($versionsArray) > 0) {
9         foreach ($versionsArray as $version) {
10            // vložení dat do pole
11            array_push($version_info,
12                array(ID => $version->getId(),
13                    VERSION => $version->getVersion()));
14        }
15    }
16    // zakódování do JSON a poslání
17    exit(json_encode($version_info));
18 }
```

Funkce nejprve načte z request požadavku jméno předmětu, pro který má zasílat informace o článcích. Poté si z databáze vytáhne potřebné informace. Ty uloží do dvourozměrného pole, které je následně zakódováno a posláno zpět na klienta který informace žádal.

Na straně klienta pak přenesený JSON dekodujeme a zpracujeme.

Ukázka kódu zpracování přijatých zpráv v aplikaci pro operační systém Android:

```

1  /**
2  * Vrátí JSON data ze webové služby
3  * param httpUrl - adresa služby
4  */
5  private static String getDataFromApi(String httpUrl) throws
        ApiException {
6      StringBuilder builder = new StringBuilder();
7      HttpClient client = new DefaultHttpClient();
8      HttpGet httpGet = new HttpGet(httpUrl);
9      try {
10         // poslání dotazu na server
11         HttpResponse response = client.execute(httpGet);
12         // vrácení výsledku
13         StatusLine statusLine = response.getStatusLine();
14         int statusCode = statusLine.getStatusCode();
15
16         if (statusCode == STATUS_OK_200) {
17             // dostali jsme správná data
18             HttpEntity entity = response.getEntity();
19             InputStream content = entity.getContent();
20             BufferedReader reader =
21                 new BufferedReader(new InputStreamReader(content));
22             String line;
23             // čtení dat
24             while ((line = reader.readLine()) != null) {
25                 builder.append(line);
26             }
27         } else {
28             Log.e(ApiHandler.class.toString(), ERR_DOWNLOAD);
29             throw new ApiException(ERR_DOWNLOAD);
30         }
31     }
32
33     } catch (ClientProtocolException e) {
34         Log.e(ApiHandler.class.toString(), e);
35         throw new ApiException(e);
36     } catch (IOException e) {
37         Log.e(ApiHandler.class.toString(), e);
38         throw new ApiException(e);
39     }
40
41     return builder.toString();
42 }

```

Dále v kódu jsou hodnoty načteny do objektu **JSONArray**.

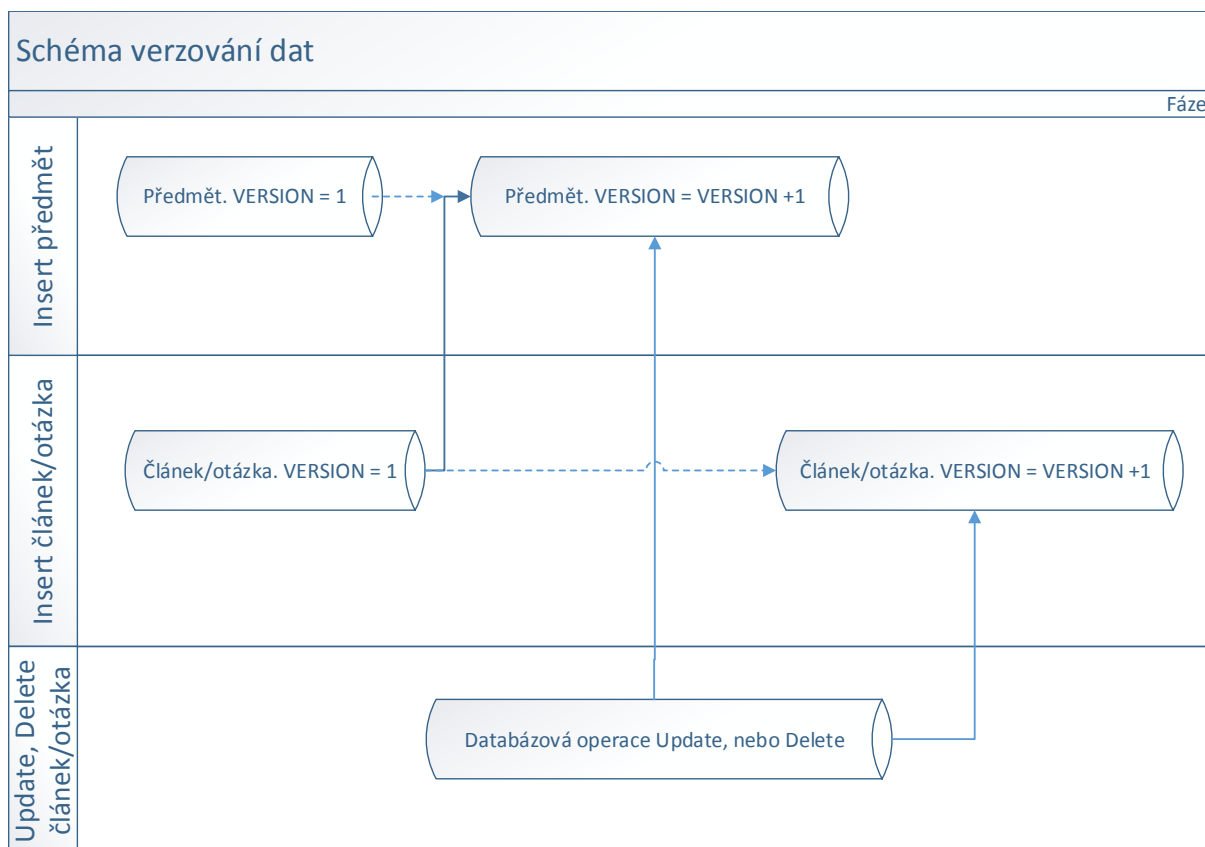
Jak již bylo popsáno výše, tak základem synchronizace jsou verzovaná data na straně serveru. Aby bylo verzování dat co nejvíce efektivní tak byly vytvořeny databázové

triggery, které mají na starost automatickou inkrementaci sloupečku **version** u datových tabulek. Takto je zaručená jednoznačná integrita ve verzování dat na serveru. Verzování v aplikačním kódu by nebylo tak transparentní a bylo by mnohonásobně méně efektivní.

Ukázka databázového triggeru po změně článku:

```
1 # verzovani dat clanku
2 DROP TRIGGER IF EXISTS 'verIncTrg'
3 CREATE TRIGGER 'verIncTrg' BEFORE UPDATE ON 'dp_clanek'
4   FOR EACH ROW BEGIN
5     # inkrementace sloupecku version aktualizovaneho zaznamu
6     SET NEW.version=OLD.version + 1;
7 END
8 //
9 # aktualizace verze predmetu v tabulce verzi
10 DROP TRIGGER IF EXISTS 'contentUpTrg'
11 CREATE TRIGGER 'contentUpTrg' AFTER UPDATE ON 'dp_clanek'
12   FOR EACH ROW BEGIN
13     UPDATE dp_version
14       # inkrementace sloupecku version predmetu zaznamu
15       SET version=version + 1
16       WHERE predmet=NEW.predmet;
17 END
```





Obrázek 5.7: Schéma verzování aplikačních dat.

### 5.3 Rozšíření statistik

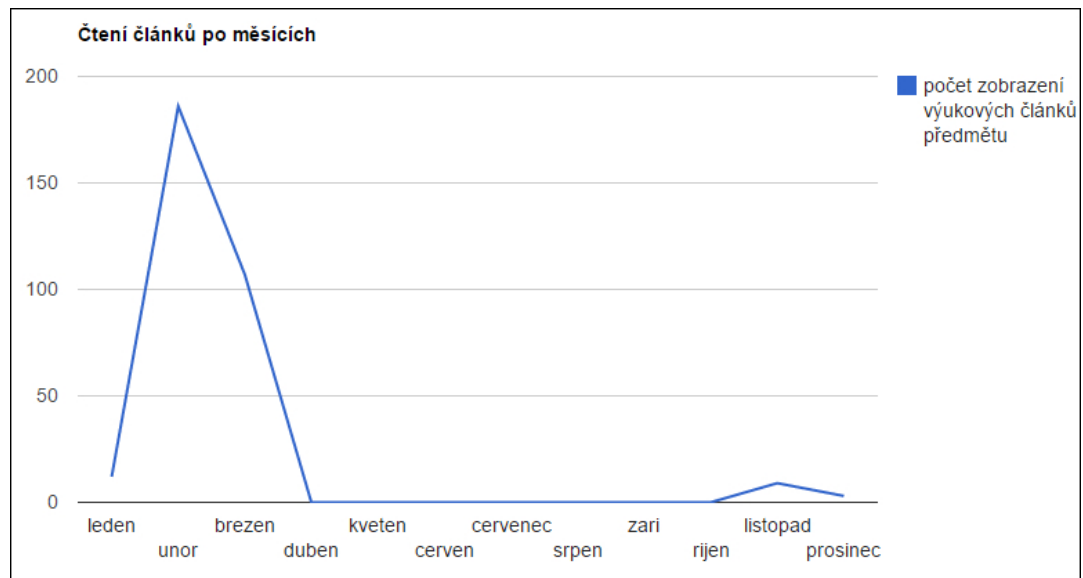
V této kapitole je popsána implementace modulu, který slouží pro zprostředkování interních statistik výuky systému. Samotným naměřeným statistikám a webové analytice se věnuje kapitola 8.

Původní verze systému Teri obsahovala modul pro zobrazování statistik. Ten byl v rámci diplomové práce rozšířen a optimalizován. Dříve byly statistiky zaměřeny na výstupy výukových testů. Nyní modul zpracovává také statistiky výukových článků.

Učitel dostane například statistiku, v jakých hodinách studenti čtou výukové články viz obr. 5.8. Obdobnou statistikou je také graf zobrazující počet čtenářů obsahu po měsících viz obr. 5.9.



Obrázek 5.8: Čtení výukových článků během dne.



Obrázek 5.9: Čtení výukových článků během roku.

Základní statistikou je tabulka, která ukazuje počet zobrazení jednotlivých článků viz obr. 5.10.

Zobrazení výukových článků

	Název článku	Počet zobrazení ▾
1	Fragments in Android Studio 1	307
2	Článek číslo 2	10
3	Zátěžové testy v prostředí Oracle	4
4	Funkční testování	2
5	Jednotkové testy v jazyce Java	1

Obrázek 5.10: Tabulka pro četnost zobrazení článků.



Tato i ostatní tabulky statistik mají funkci řazení. Kliknutím na popis sloupečku lze tedy hodnoty v tabulce pohodlně řadit. Jednou z funkcionalit při tvorbě výukových článků je možnost ukládat přílohy. Aby měl učitel přehled kolik studentů si dané přílohy stahuje, byla přidána také tabulka četnosti stahování příloh viz obr. 5.11.

Statistika stahování příloh

	Příloha	Počet stažení
1	automatic_testovani.pdf	6
2	tdd.png	1

Obrázek 5.11: Tabulka pro četnost zobrazení článků.

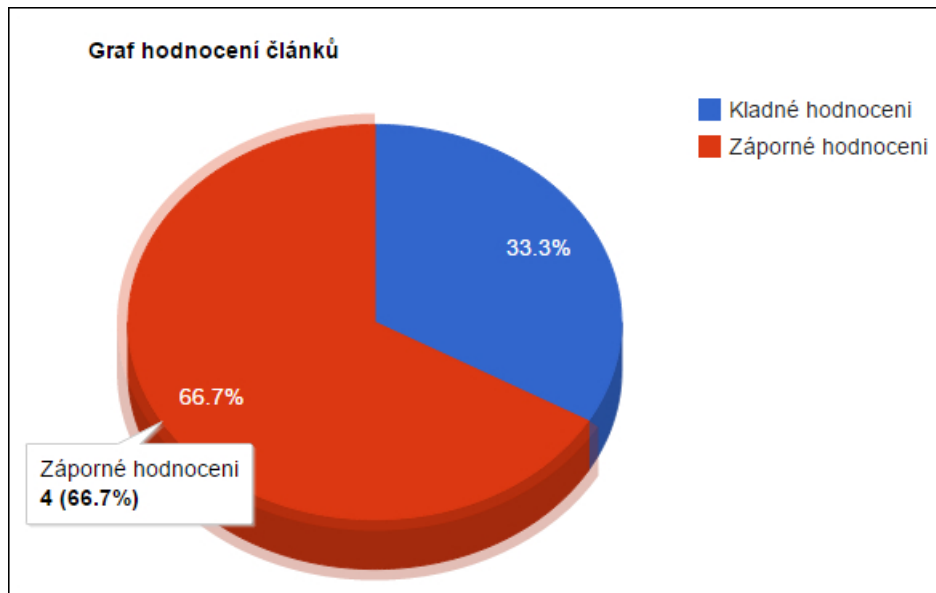
Poslední statistikou z pohledu výukových článků je statistika hodnocení článků. Přihlášení studenti mají při čtení obsahu možnost článek kladně či záporně ohodnotit viz obr. 5.12.

Byl pro vás tento článek přínosný?  

Poslední změna: 26.02.2015

Obrázek 5.12: Uživatelské hodnocení článku.

Učiteli se pak zobrazí tabulka všech článků a jejich hodnocení. Dále se také vykreslí graf poměru kladných a záporných hodnocení viz obr. 5.13.



Obrázek 5.13: Graf hodnocení článků.

Jednotlivé statistiky se ukládají do databázových tabulek **DP\_CLANEK\_STATS**, **DP\_OTAZKA\_STATS** a **DP\_PRILOHA\_STATS**. Pro zobrazování grafů byla použita JavaScript knihovna **Google Charts** [26]. Ta je zdarma dostupná a šířena pod licencí Creative Commons. Na zobrazování grafů byla použita komponenta **Pie Chart** na koláčové grafy a komponenta **Area chart** na vykreslování grafů četností. Knihovna také obsahuje komponentu **Table**, pomocí ní byly implementovány tabulky s funkcí řazení záznamů.

## 5.4 Rozšíření výukových testů

V bakalářské práci se modul výukových testů při výuce předmětu ZOS osvědčil jako přínosný. Proto byl tento modul v rámci diplomové práce dále rozšířen. Prvním vylepšením byla změna vzhledu výukových otázek testu. Otázky byly více zpřehledněny a při vyhodnocení odpovědi se výsledky zobrazují v jednotném stylu notifikací, které jsou na celém webu (více v kapitole 5.6).

Dále byly u všech výukových otázek přidány odkazy s možností prokliku na stránku s výukovými materiály daného předmětu. Student se tedy může rychle dostat k výukovým článkům, aby rozšířil své znalosti.

Nejvýznamnějším zlepšením bylo přidání nového druhu otázky. Jedná se o interaktivní seřazovací otázku. Funkce řazení odpovědí se provádí kurzorem myši metodou

*táhni a pusť*. Tento druh otázky byl zvolen z důvodu snadné tvorby a editace. Není třeba tvorby složitého editačního nástroje pro učitele. Formulář pro tvorbu seřazovací otázky je obdobný jako u ostatních otázek s více odpověďmi. Učitel zadá otázky, které jsou ve správném pořadí viz obr. 5.14.

The image shows a web interface for creating a sorting question. At the top, there are three steps: 1. Otázky, 2. Přidání nové otázky (highlighted), and 3. Tvorba otázky. Below this, the title 'Přidání nové otázky' is displayed. The question text is 'Seřadte datové typy jazyka Java dle jejich velikosti'. Below the question, the instruction 'Odpovědi - Napište odpovědi ve správném pořadí' is shown. There are four input fields for answers, numbered 1 to 4, with the following text: 1. byte, 2. short, 3. int, 4. long.

Obrázek 5.14: Formulář pro tvorbu seřazovací otázky.

Studentům se pak otázka zobrazí s náhodně proházenými odpověďmi viz obr. 5.15. Odpovědi se vypíší do HTML seznamu. Aby tento seznam měl funkci řazení tak byla použita knihovna **jQuery UI**[27]. Tato knihovna je volně šiřitelná pod licencí MIT. Pro potřeby tvorby seřazovací otázky byla využita funkce **sortable** která je volána nad selektorem našeho seznamu. Tím docílíme seřazovacího seznamu. Dále bylo potřeba tento nečíslovaný seznam poslat v aktuálním seřazení přes formulář na stránce.

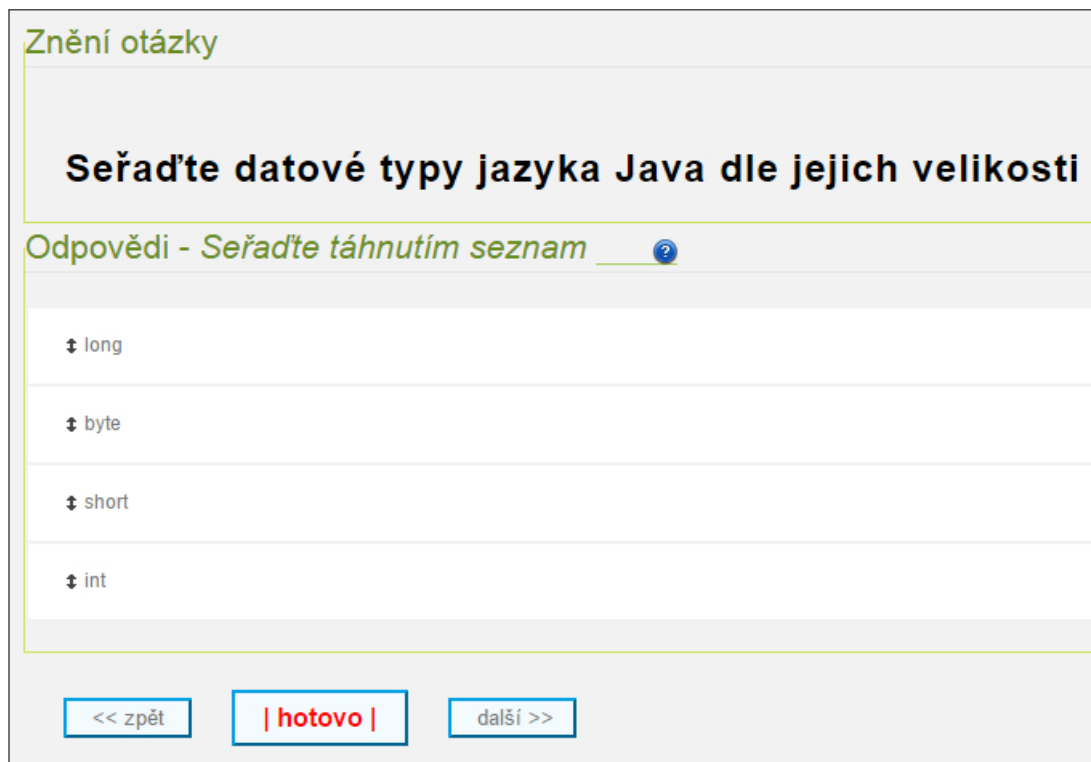
Na to byl napsán vlastní kód v jazyce JavaScript:

```

1 <script type="text/javascript">
2 // script je spousten pri akci odeslani formulare
3 $('#questionForm').submit(function (e) {
4     // zabrani vychozi akci
5     e.preventDefault();
6     // ziskani pole ID polozek dle aktualniho serazeni
7     var idsInOrder = $("#sortable").sortable("toArray");
8     // ziskane poradí nastavíme do hidden prvku formulare
9     $("#resultId").val(idsInOrder);
10    // odeslani formulare
11    this.submit(); // call the submit function
12    });
13 </script>

```

Skript se spouští při události odeslání formuláře s otázkou. Načte si identifikátory položek seznamu tak, jak jsou aktuálně zobrazeny. Tyto položky načte do pole, které následně uloží jako hodnotu skrytého atributu posílaného prostřednictvím formuláře.



Znění otázky

**Seřad'te datové typy jazyka Java dle jejich velikosti**

Odpovědi - Seřad'te táhnutím seznam ?

- long
- byte
- short
- int

<< zpět | hotovo | další >>

Obrázek 5.15: Zobrazená seřazovací otázka.

Po vyhodnocení seřazovací otázky se provede podbarvení jednotlivých položek seznamu. Položky, které jsou ve správném pořadí budou po vyhodnocení podbarveny zeleně a položky chybně seřazené naopak červeně viz obr. 5.16. Uživatel tak dostane přehlednou zpětnou vazbu na svojí odpověď.



Obrázek 5.16: Vyhodnocení seřazovací otázky.

## 5.5 Rozšíření ostrých testů

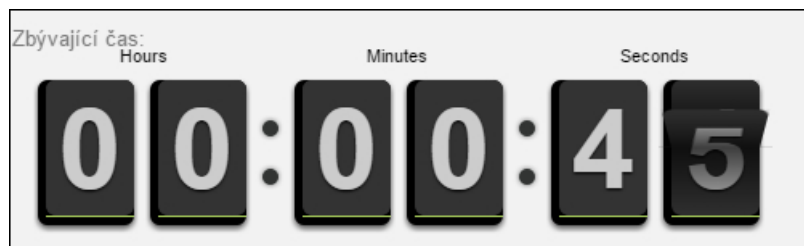
V původní verzi systému byla vytvořena funkcionality ostrého testování. Učitel si mohl vytvořit test, který přihlášení studenti závazně vyplnili a odevzdali. Vyučující pak dostal jejich výsledky. Jedná se o užitečnou funkcionality s potenciálem na využití v prostředí katedry. Z toho důvodu byl původní zjednodušený modul pro ostré testy přepracován a rozšířen tak, aby bylo jeho použití více reálné.

Prvním vylepšením byl přístup k platnosti samotného testu. Dříve učitel zadal pouze datumy a časy rozpětí kdy je test přístupný studentům pro spuštění. Nijak ale nebyla omezena doba vykonávání testu a počet spuštění od jednoho studenta. To bylo změněno tak, že nyní si student daný test může spustit pouze jednou. Rozpětí platnosti zůstalo zachováno s tou změnou, že po vypršení platnosti testu student svůj test již neodevzdá.

Novou funkcí, kterou může učitel využít je nastavení doby trvání pro vykonání testu. Učitel tedy při vytváření testu nastaví dobu, za kterou musí student od spuštění

testu test odevzdat. V případě přesáhnutí této mezní doby systém odevzdání nepovolí a studenta informuje chybovou hláškou. V případě, že učitel tuto funkcionalitu nebude chtít využít tak může nechat nastavení časovače na nule a výpočet zbývajících času do odevzdání se provede vůči meznímu datu platnosti testu.

Po zavedení časovače pro zbývajících čas do odevzdání testu bylo potřeba implementovat informační komponentu, která by zobrazovala zbývajících čas. Pro vykreslení hodin byla použita knihovna **FlipClock.js** [28]. Knihovna je volně šiřitelná pod licencí MIT a umožňuje snadnou implementaci čítacích a odečítacích hodin. Po spuštění testu se na straně serveru vypočte zbývajících čas do odevzdání testu a ten je pak nastaven pro zobrazované hodiny v odečítacím módu viz obr. 5.17.












Obrázek 5.17: Hodiny zobrazující zbývajících čas do odevzdání testu.


Hodiny pak dají studentovi jasnou informaci kolik času mu zbývá na odevzdání testu.

Při tvorbě testu má nyní učitel také možnost zvolit, zdali se mají prohazovat otázky testu jednotlivým studentům, prohazovat odpovědi otázek, nebo kombinace obojího, tedy studentům proházet otázky i odpovědi (tato možnost je jako výchozí). Při volbě obou možností každý student dostane množinu otázek ostrého testu v náhodném pořadí včetně náhodně proházených odpovědí. Hlavním důvodem této funkce je zamezit přímého podvádění pokud by test studenti prováděli hromadně v jedné učebně. Pro náhodné prohazování dat se používá PHP funkce **shuffle**.

Vylepšením prošla také stránka s výsledky ostrých testů v administraci učitele. Učitel dostane v tabulce přehledně vypsán seznam studentů a jejich úspěšnost při vyplňování testu viz obr. 5.18.



Výsledky ostrých testů							
Jméno	os. číslo	nick	e-mail	správné odpovědi	špatné odpovědi	% úspěšnost	detail
Josef Kulatý	A12N96237	petr2	<a href="mailto:kulaty@students.zcu.cz">kulaty@students.zcu.cz</a>	4	5	44.4444444444 %	 
František Král	A12N513489	frantisek.k	<a href="mailto:kral@students.zcu.cz">kral@students.zcu.cz</a>	5	4	55.5555555556 %	 
Lenka Davidková	A12N98456	lenda3310	<a href="mailto:davidkova@students.zcu.cz">davidkova@students.zcu.cz</a>	9	0	100 %	 
Eliška Černá	A12N56842	elis.cerna	<a href="mailto:cerna@students.zcu.cz">cerna@students.zcu.cz</a>	7	2	77.7777777778 %	 
Jan Novák	A12N845612	novak.jan	<a href="mailto:novak@students.zcu.cz">novak@students.zcu.cz</a>	4	5	44.4444444444 %	 


[export výsledkové tabulky do PDF](#) 

Obrázek 5.18: Tabulka výsledků ostrého testu.

Učitel tak dostane hned po vyplnění ostrého testu výsledky. Pokud potřebuje vidět detailní záznam, jak student na test odpovídal tak si může ve sloupci detail zobrazit log odpovědí v konzoli, nebo si tento záznam vyexportovat do PDF dokumentu. To může sloužit například pro archivaci. Učitel je také umožněno do PDF uložit celou tabulku s výsledky všech studentů.

V původní verzi šlo ostré testy vytvářet pouze z otázek, které byly dostupné ve výukových testech. Studenti měli při cvičném testování přístup k přesnému znění otázek, které se pak objevily v testu ostrém. To byla komplikace, pokud učitel chtěl ostré testy použít pro testy s vysokou prioritou a chtěl zamezit podvádění a zkresleným výsledkům. Proto byla přidána možnost tvorby otázek, které nejsou veřejně přístupné a jsou určeny pouze pro ostré testy. Učitel při tvorbě nové otázky ve formuláři může zaškrtnout, že se jedná o skrytou otázku určenou pro ostré testy.

Při tvorbě ostrého testu má pak možnost zvolit, zdali se mají pro ostrý test používat otázky z výukových testů, otázky určené pro ostré testy, nebo jejich kombinace viz obr. 5.19.

Skryté otázky 

---

Skryté otázky vynechat

Zahnout skryté otázky

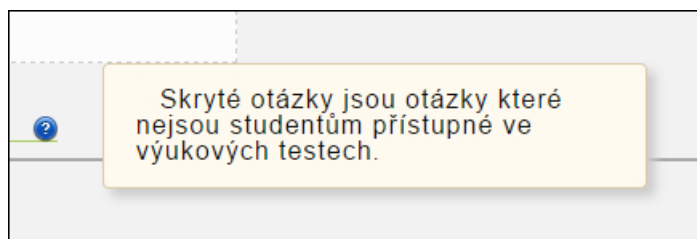
Generovat test pouze ze skrytých otázek

Obrázek 5.19: Volba zahrnutí skrytých otázek při tvorbě ostrého testu.

Pro zlepšení práce s ostrými testy z pohledu učitele byla dále přidána funkce zobrazení ostrého testu. Pokud si chtěl dříve učitel ostrý test prohlédnout, aby viděl jak se zobrazí studentům, tak si musel daný test závazně spustit. V takovém případě se zaznamenaly výsledky a údaje o spuštění testu. Proto byla do administrace přidána funkce náhledu ostrého testu. Vyučující předmětu tak dostane kontrolu nad nově vytvořeným ostrým testem.

## 5.6 Kontextová nápověda a notifikace

Pro zobrazování uživatelských zpráv (notifikací) a kontextové nápovědy byl vytvořen samostatný modul. Na mnoha místech v celé aplikaci jsou pak vykreslovány ikonky otazníčku, na které když uživatel najede myší tak se mu zobrazí detailnější nápověda v plovoucím okně viz obr. 5.20.



Obrázek 5.20: Kontextová nápověda po najetí kurzoru na ikonku otazníku.

Celá komponenta nápovědy využívá pouze možností jazyka **HTML** v kombinaci s kaskádovými styly **CSS3**. Funkce pro vykreslení ikonky s nápovědou:

```

1  /**
2  *  Provede vypis tooltip napovedy jako HTML komponenty
3  *  param $messageResourceName - name identifikator string resource
4  */
5  public static function printTooltip($messageResourceName) {
6      ?>
7      <a href="#" class="tool">
8          
9          <span class="content_help">
10             <?php
11                 // vypis textu napovedy
12                 echo StringResources::getString($messageResourceName);
13             ?>
14         </span>
15     </a>
16     <?
17 }

```

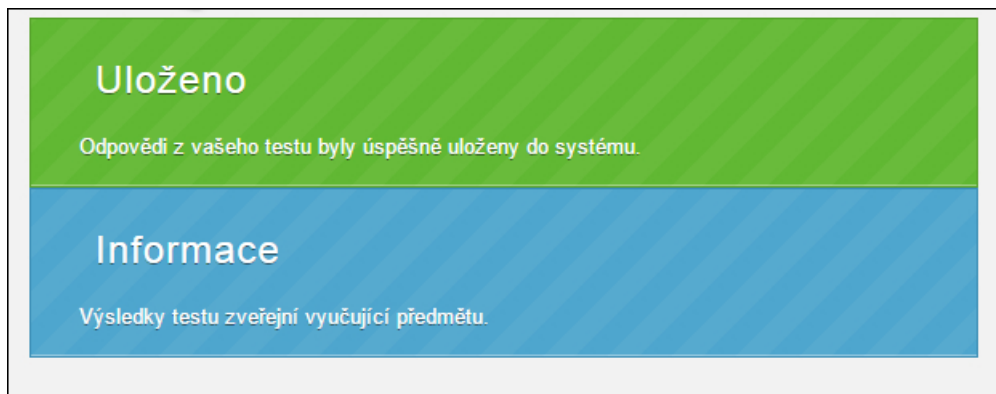
Kaskádové styly pro skrývání a zobrazování textu nápovědy po přejetí kurzoru myši nad ikonou:

```

1  /* Skryte pole s napovedou */
2  #main-content a.tool span {
3      z-index:10; display:none; padding:14px 20px;
4      margin-top:-30px; margin-left:28px;
5      width:340px; line-height:16px;
6      color: white;
7  }
8  /* Po najeti mysi na ikonku zobrazi pole s napovedou */
9  #main-content a.tool:hover span{
10     display:inline; position:absolute; color:#111;
11     border:1px solid #DCA; background:#fffAF0;
12     text-decoration: none;}

```

Další funkcionalitou modulu je zobrazování notifikací. Dříve bylo informování uživatele o prováděných akcích nejednotné. Proto po *redesignu* webu došlo ke sjednocení těchto notifikací pro větší přehlednost viz obr. 5.21.



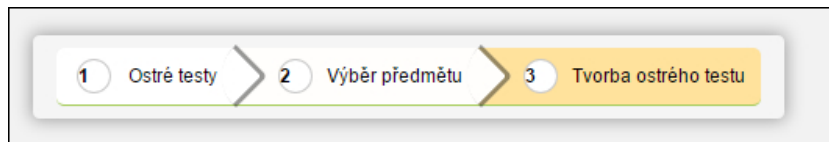
Obrázek 5.21: Ukázka vzhledu sjednocených notifikací.

## 5.7 Drobečková navigace

Jak se celý portál Teri rozrůstá s přibývajícím funkcionalitou, tak se nelze ubránit vícenásobnému zanoření při procházení hlavních sekcí webu. Vícenásobnému zanoření se například nelze vyhnout v některých sekcích administrace. Ve staré verzi webu tento problém nebyl nijak řešen. Uživatel se mohl vracet přes tlačítko zpět ve webovém prohlížeči, anebo se mohl dostat na úvodní stránky sekcí přes hlavní menu aplikace. Aby bylo ovládání aplikace více uživatelsky přívětivé, tak byla implementována komponenta pro drobečkovou navigaci.

Tento druh navigace je poměrně rozšířeným typem navigačního prvku. Na úvodní stránky sekcí se uživatel dostane odkazem v hlavním menu aplikace a poté se mu

v horní části stránky zobrazí drobečkové menu viz obr. 5.22, jehož prostřednictvím se může pohybovat zpět v krocích zanoření. Možností implementace je více, rozhodl jsem se vytvořit navigační menu s velkými tlačítky s ohledem na snadnou použitelnost na zařízeních s dotykovou obrazovkou. Jako oddělovač jednotlivých úrovní zanoření byl použit znak *větší než*. Při volbě tohoto oddělovacího symbolu jsem vycházel z průzkumu drobečkových navigací 4 775 webů [29]. V tomto průzkumu bylo zjištěno, že u webů které drobečkovou navigaci měly použítou byl nejčastějším symbolem (47%) právě znak *větší než*.



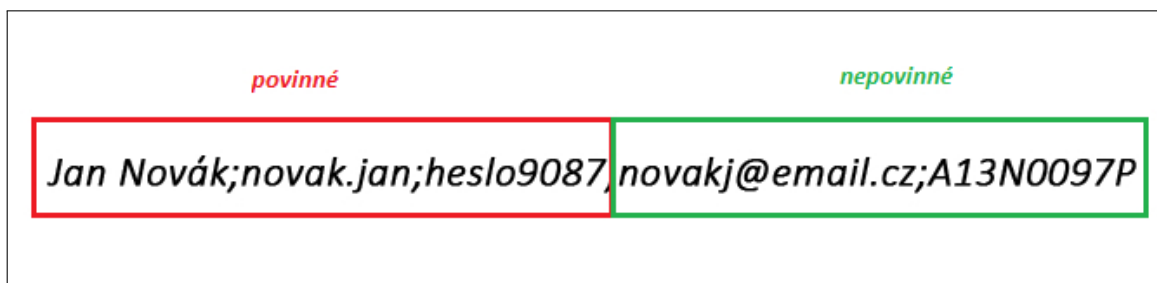
Obrázek 5.22: Drobečková navigace.

## 5.8 Hromadný import studentů

V rámci této práce byl přepracován modul pro ostré testování studentů. V případě, že se učitel rozhodne takto testovat své studenty, tak může narazit na problém, že musí být všichni studenti, kteří by test spouštěli v systému zaregistrováni. To by za předpokladu, že by test probíhal hromadně mohlo zbytečně zdržovat. Z tohoto důvodu byl vytvořen hromadný import studentů. Ten umožní učiteli hromadně zaregistrovat všechny studenty svého předmětu.

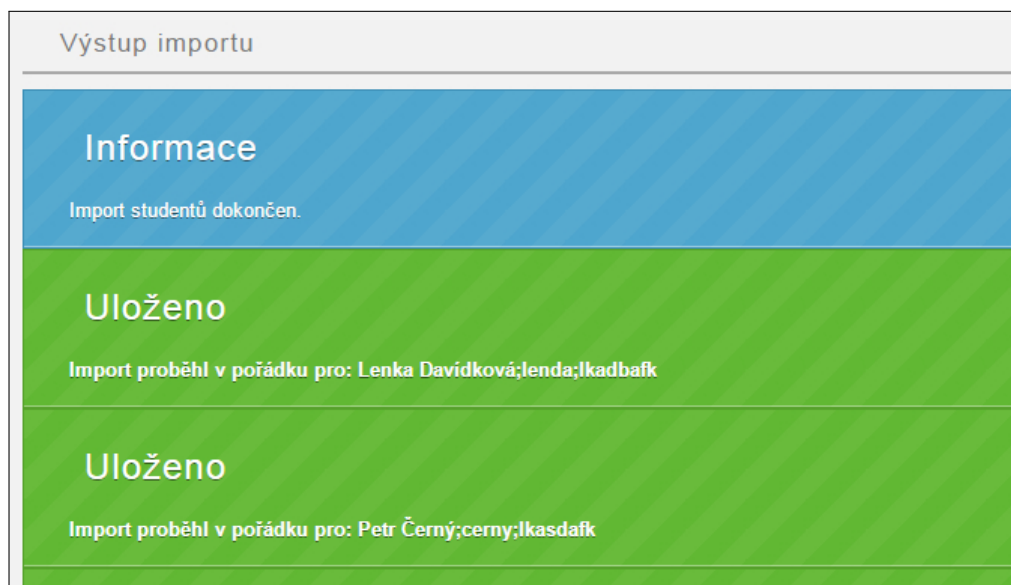
Jako formát importního souboru byl zvolen CSV dokument používající oddělovací znak středník. Učitel si tedy vygeneruje takový importní soubor kde na každém řádku je definice jednoho studenta.

Formát dat musí být ve tvaru: *jméno studenta;login;heslo;e-mail;osobní číslo*. Z toho e-mail a osobní číslo jsou nepovinné údaje viz obr. 5.23. Předpokládané kódování souboru je **UTF-8**.



Obrázek 5.23: Formát importních dat pro hromadné uložení studentů.

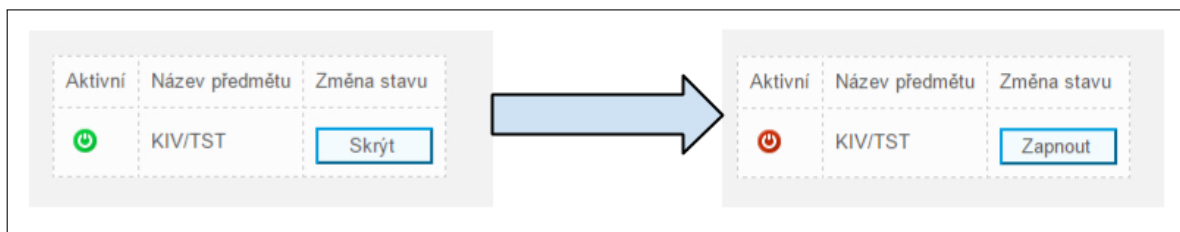
Systém po dokončení importu učitele informuje o dokončení. Pro každého importovaného studenta zobrazí notifikaci, zdali jsou data validní a povedlo se je úspěšně uložit viz obr. 5.24.



Obrázek 5.24: Informační výpis při hromadném importu studentů.

## 5.9 Skrytí předmětu

Další možností, která byla do webové části systému přidána je možnost skrytí předmětu. Učitel si nyní kdykoliv může předmět v administraci skrýt viz obr. 5.25. To zapříčiní, že předmět nebude pro studenty viditelný. Učitel ho ale nadále může spravovat a upravovat prostřednictvím administrace. Tuto funkčnost využije zejména v případech, kdy chce studentům předmět zviditelnit, až když má přidané všechny materiály a výukové testy. Dalším možným případem je předmět skrýt v době provádění ostrých testů nebo v případě neaktuálního obsahu.



Obrázek 5.25: Možnost skrytí předmětu a jeho opětovné zapnutí..

## 6 Podpora mobilních zařízení

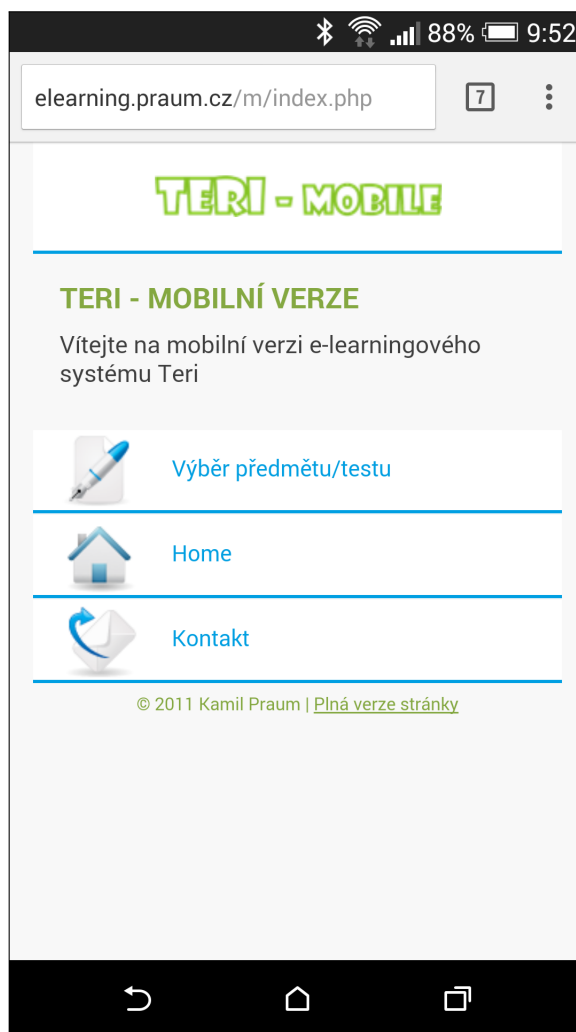
Jedním z hlavních cílů této práce je navrhnout systém s ohledem na použití v mobilních zařízeních. Proto byl kompletně přepracován vzhled webového portálu, aby byl více multiplatformní, dále pro lepší použitelnost byla implementována nativní aplikace pro systém Android.

### 6.1 Redesign webu

Z analýzy vyplývá, že jedním z možných přístupů jak se vypořádat s tímto problémem je tvorba separátních webů. Dostaneme tak klasický web pro počítače, mobilní verzi webu, verzi pro tablety atp. U předchozí verze systému Teri byla naprogramována klasická (viz obr. 6.1) a mobilní verze webu (viz obr. 6.2). Tento přístup je však špatně škálovatelný. Navíc přináší mnohem větší režii a nové problémy a hrozby. Je třeba udržovat, aktualizovat a testovat více webů což není ideální řešení.



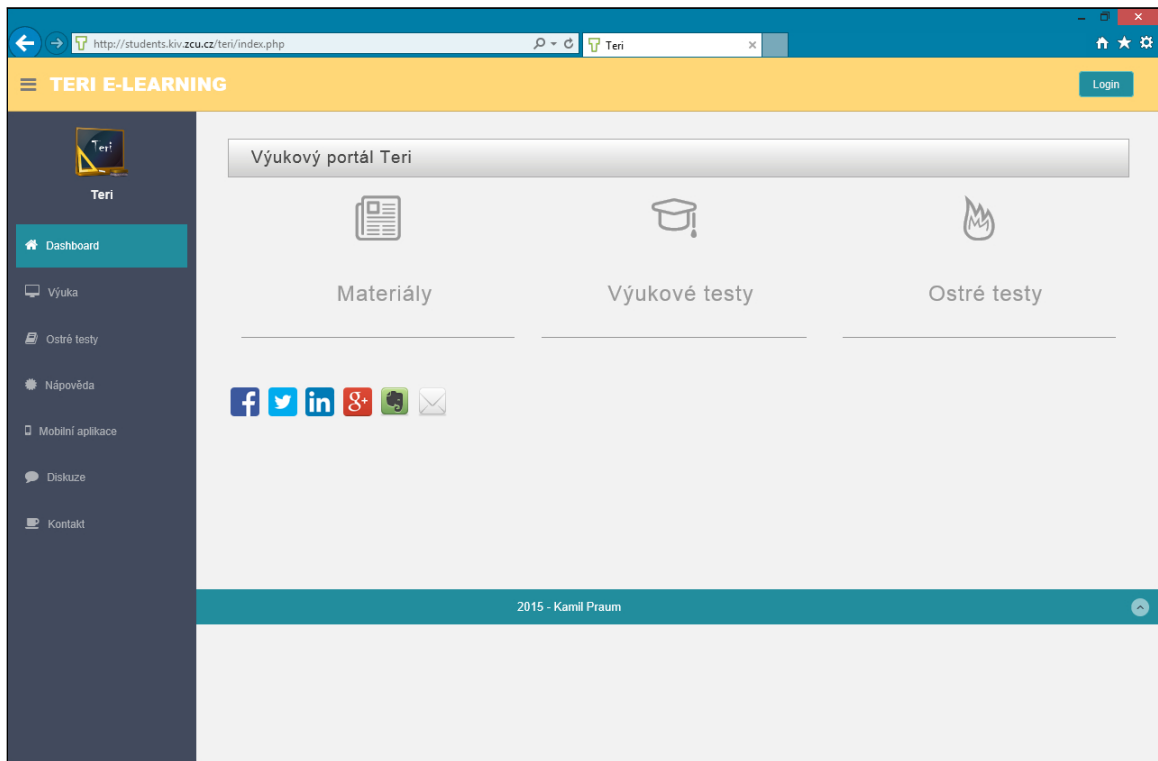
Obrázek 6.1: Původní verze systému Teri - klasické zobrazení.



Obrázek 6.2: Původní verze systému Teri - mobilní verze webu.

Lepším a aktuálním přístupem je použití **responzivního designu**. Responzivní design se nesnaží vytvářet vzhled stránky na míru pro každé ze stále rostoucího počtu zařízení na prohlížení webových stránek, ale vytváří pouze jeden flexibilní design, který se přizpůsobí dynamicky podle velikosti daného zařízení.[30] Mezi silnou stránku responzivního designu jsou mediální dotazy. Přes tyto dotazy lze zjistit například rozlišení cílového zařízení, ale třeba také výška, šířka, barevnost atd. Na základě výstupů dotazů se aplikují jednotlivé kaskádové styly.

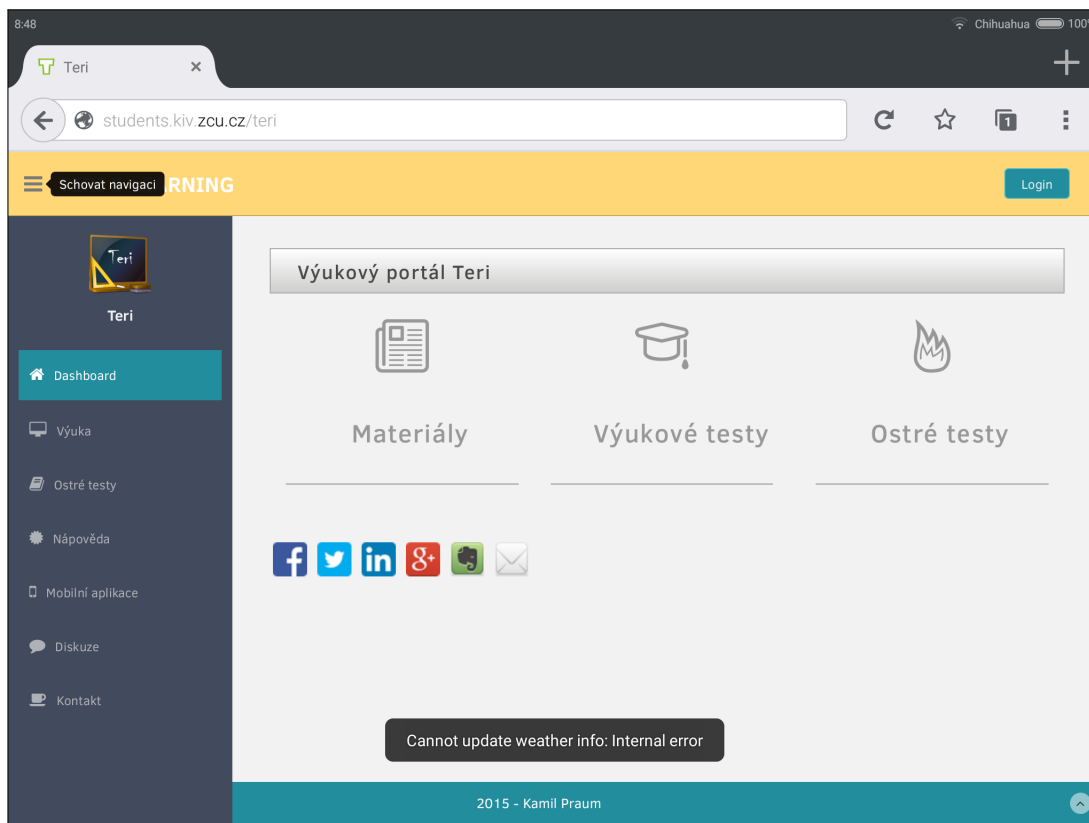
Pro potřeby redesignu portálu Teri byl použit framework **bootstrap** [31]. Bootstrap je volně šiřitelná knihovna pod licencí MIT. Jedná se o moderní *front end* framework, který zjednodušuje práci při tvorbě responzivních webů s podporou mobilních zařízení. Tento nástroj je vyvíjený společností Twitter a v současnosti patří mezi nejpoužívanější pro danou problematiku. Kromě responzivního rozložení webu nabízí bootstrap také komponenty uživatelského rozhraní jako jsou formulářové prvky, ikony, tlačítka atp.



Obrázek 6.3: Nový vzhled Teri - zobrazení na desktopovém prohlížeči.

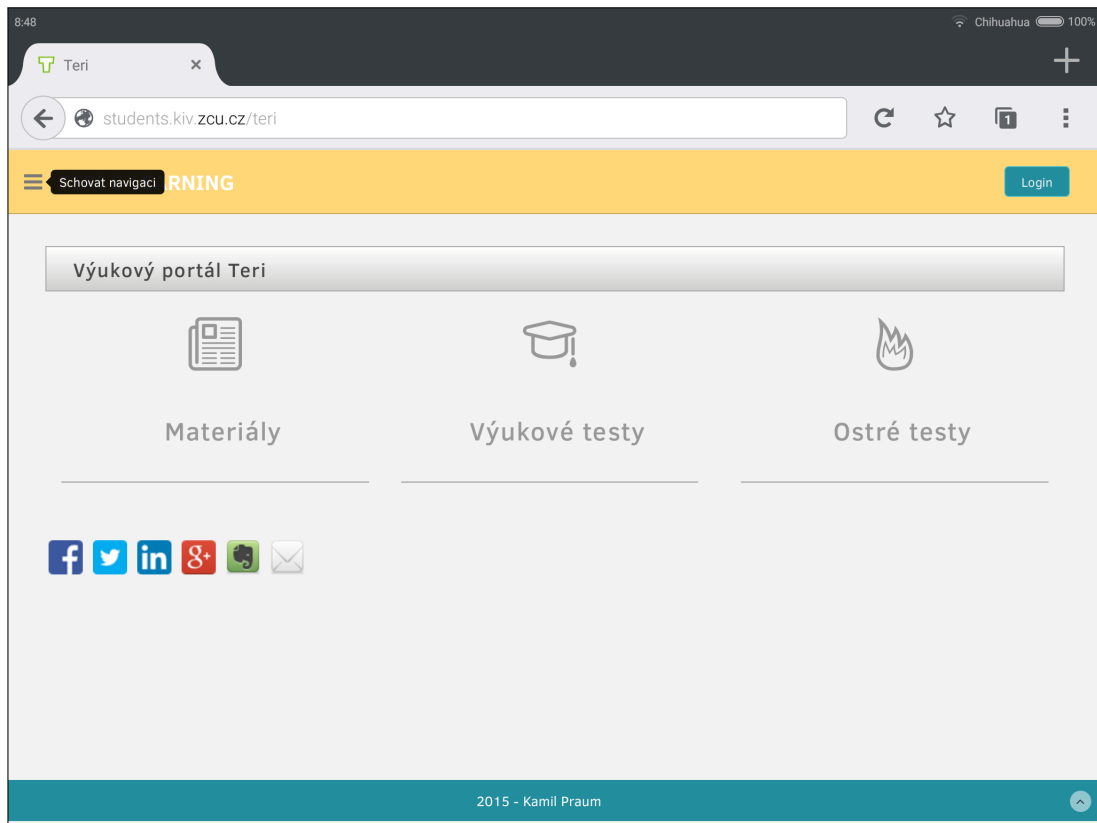
Nový vzhled webu je tedy díky použitému frameworku moderní a na všech běžných prohlížečích se zobrazuje korektně viz obr. 6.3. Responzivní přístup řešení dynamicky reaguje na výšku a šířku cílového zařízení. Na následujících obrázcích můžete vidět jak se vzhled přizpůsobí menšímu displeji tabletu s vyšší denzitou než má monitor počítače viz obr. 6.4.



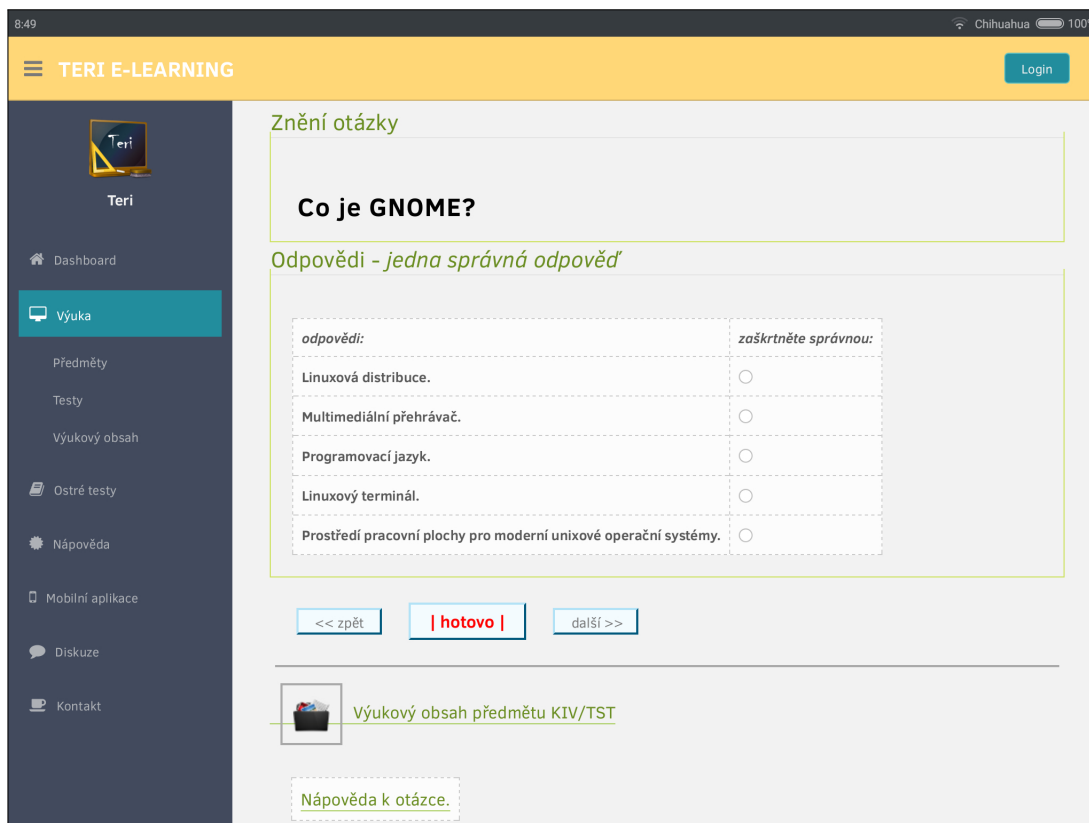


Obrázek 6.4: Nový vzhled Teri - zobrazení na tabletu (Xiaomi Mi Pad).

Na obrázku je vidět, že na displeji menšího tabletu došlo k mírnému zvětšení ovládacích prvků. Ovládání webu na dotykovém zařízení pro pohodlnou práci vyžaduje větší prvky. Dále má návštěvník webu možnost schovat levé menu viz obr. 6.5. To umožňuje lépe využít plochu displeje pro zobrazení obsahu.

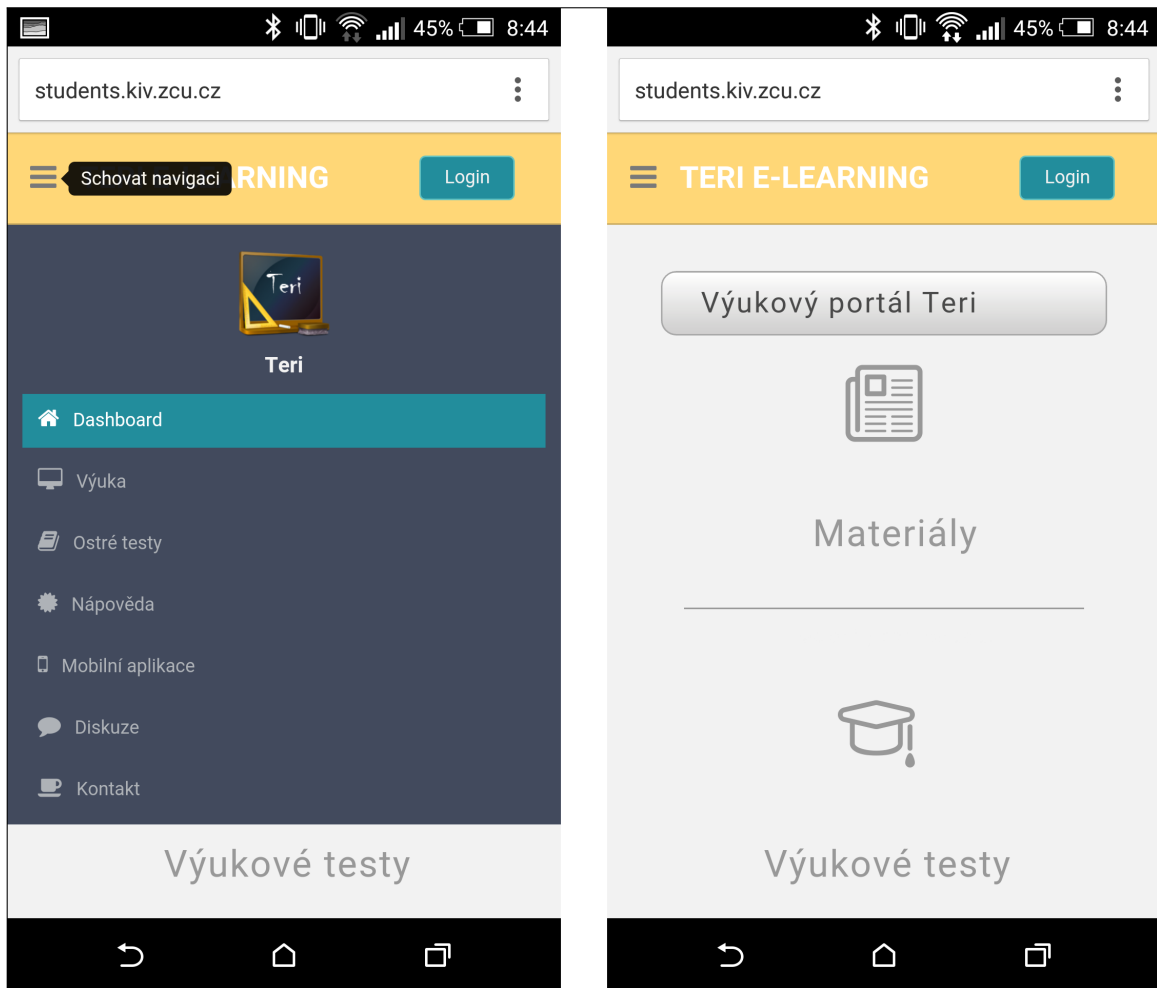


Obrázek 6.5: Nový vzhled Teri - zobrazení se skrytým menu na tabletu (Xiaomi Mi Pad).



Obrázek 6.6: Nový vzhled Teri - zobrazení otázky na tabletu (Xiaomi Mi Pad).

Poslední dva obrázky ukazují jak se web zobrazuje na mobilním telefonu. Automaticky je rozpoznána malá šířka zařízení, to vede ke zmenšení obsahové části a automatickému schování hlavního menu viz obr. 6.7. Uživatel si toto menu může zobrazit sám přes ikonku menu v levém horním rohu.



Obrázek 6.7: Nový vzhled Teri - zobrazení na mobilním telefonu (HTC One M8).

## 6.2 Implementace aplikace pro operační systém Android

Díky responzivnímu webu lze pokrýt za rozumné náklady všechna mobilní zařízení. Dalším možným přístupem je naprogramovat aplikaci v klasických webových technologiích (HTML, CSS, JavaScript) a s pomocí speciálních knihoven tuto aplikaci přeložit na libovolné mobilní platformy a udělat z nich nativní aplikaci. Mezi nejznámější knihovny pro tvorbu takových aplikací patří **PhoneGap** [32]. V rámci diplomové práce byl naprogramován prototyp takové hybridní aplikace. Nakonec jsem se ale rozhodl implementovat klasickou nativní aplikaci. Hlavním důvodem bylo, že u hybridní aplikace lze velmi obtížně dosáhnout takového vzhledu aplikace, aby uživatel nepoznal rozdíl od klasické nativní aplikace a dostal standardní komponenty uživatelského rozhraní pro danou platformu. V době, kdy už byla aplikace pro systém Teri hotová představila společnost Facebook novou knihovnu **React Native** [33]. Ta řeší tyto problémy a dokáže hybridní aplikaci překompilovat do nativního kódu platformy. Lze tedy dosáhnout požadovaného vzhledu a chování aplikace pro danou platformu. V případě, že bude systém Teri dále vyvíjen, tak bych se zaměřil na tento směr vývoje mobilní aplikace.

Nativní aplikace byla naprogramována pro operační systém **Android**. Důvod, proč byla vybrána tato platforma je jednoznačně největší podíl na trhu chytrých mobilních zařízení a majoritní zastoupení mezi studenty. Podle průzkumu společnosti IDC měl Android na konci roku 2014 tržní podíl **76,6%** [34].

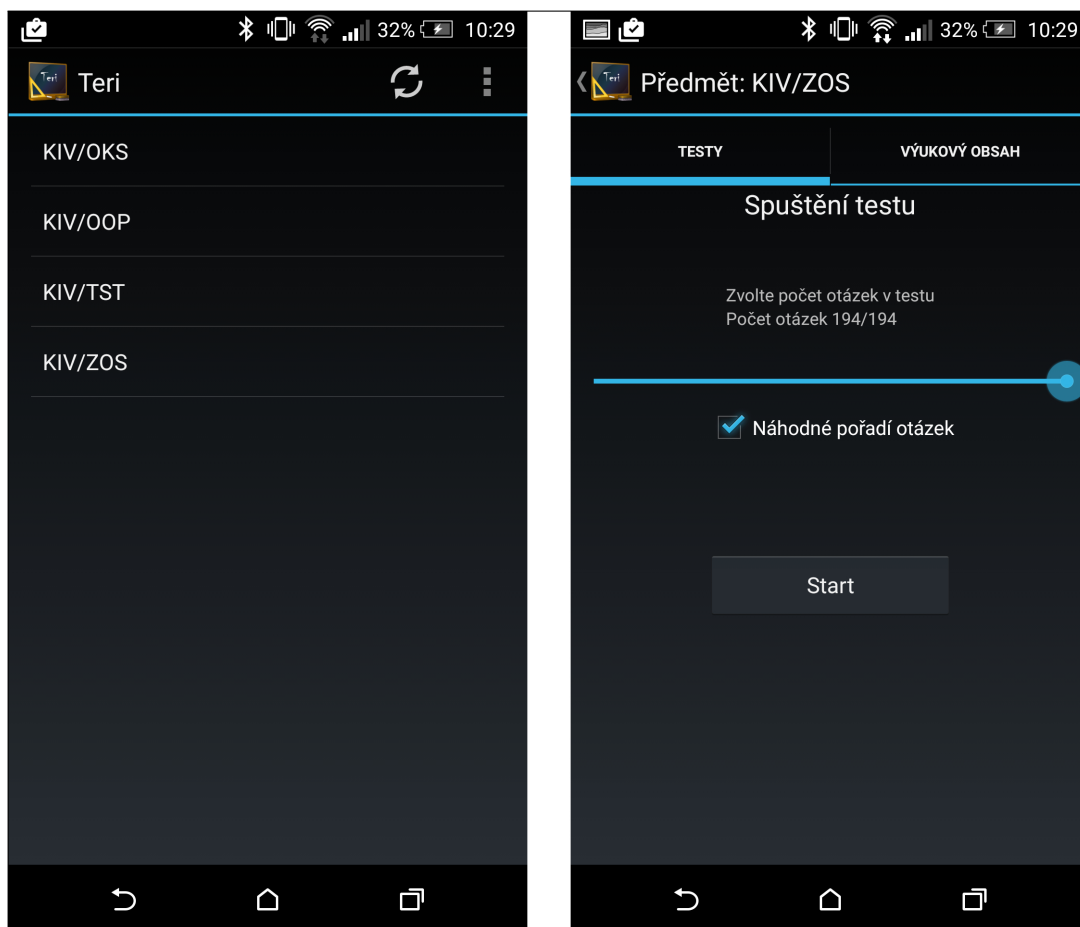
Aplikace má hlavní seznam, který zobrazuje studentům dostupné předměty. Po kliknutí na název předmětu se zobrazí možnost čtení výukového obsahu, nebo spuštění testu. Vzhled aplikace je naprogramován metodou fragmentů. Fragments byly poprvé implementovány ve verzi Android 3.0. Od verze 4.0<sup>1</sup> jsou pak součástí zdrojového kódu systému [35]. Fragments přinášejí další vrstvu nad klasické *activity*. Přinášejí možnost měnit nastavení aktivity v reakci na velikost obrazovky cílového zařízení. Zjednodušeně lze tedy říci, že fragmenty umožňují tvorbu dynamického uživatelského rozhraní pro mobilní telefony a tablety. Není tedy třeba implementovat verzi aplikace pro mobilní telefony a zvláště také pro tablety. Operační systém Android je také dostupný pro chytré televize, brýle, hodinky atd. To přináší velké množství zařízení s odlišnou velikostí obrazovek. Tvorba univerzálních aplikací je aktuálním a poměrně složitým tématem.

V našem případě byla hlavní část aplikace rozdělena na dva fragmenty. První fragment slouží pro zobrazení všech předmětů. Druhý fragment pak pro obsah daného předmětu. Konkrétně spuštění testu a prohlížení výukového obsahu. Pokud je zařízení detekováno jako zařízení s úzkým displejem (typicky mobilní telefon) tak se tyto fragmenty

---

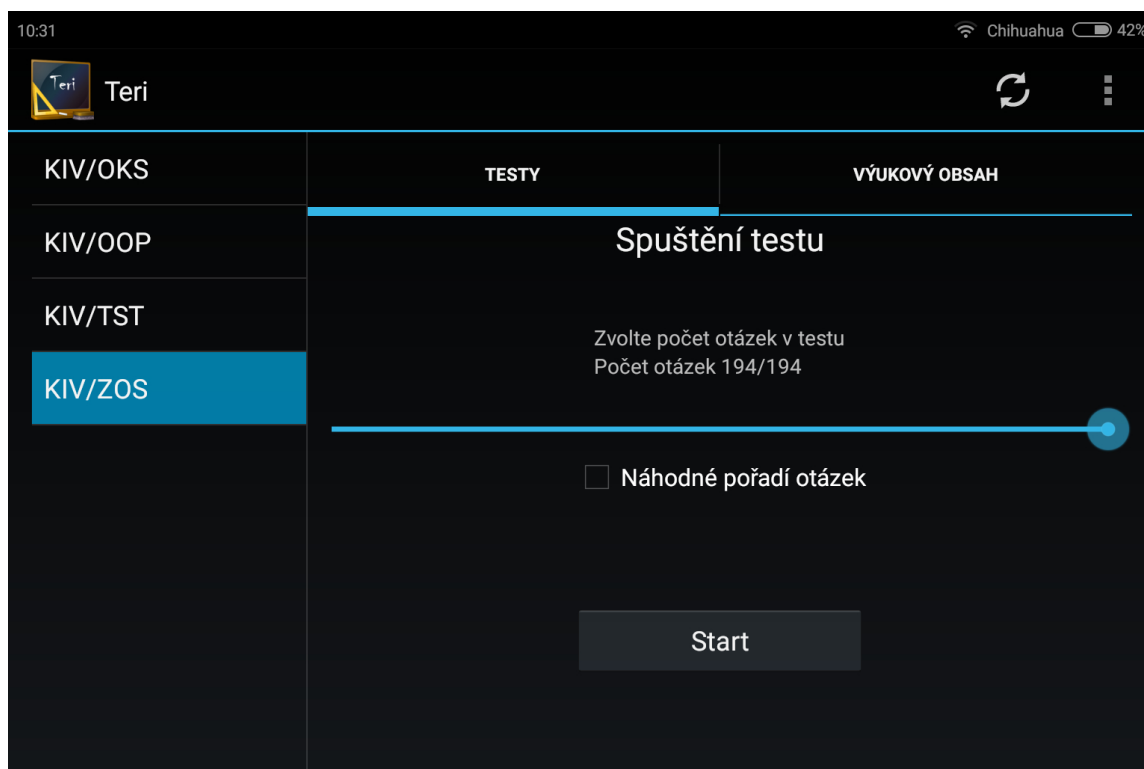
<sup>1</sup>Z důvodu použití fragmentů je pro aplikaci požadována minimální verze operačního systému Android ve verzi 4.0

zobrazují odděleně. To znamená že se fragmenty chovají z pohledu uživatele jako klasické aktivity viz obr. 6.8.



Obrázek 6.8: Nativní Android aplikace zobrazená na chytrém telefonu.

Po kliknutí na název předmětu dojde k zobrazení nového okna s druhým fragmentem. V případě zobrazení na tabletu ale dojde k vykreslení obou fragmentů v jedné aktivitě. Je tak docíleno lepšího využití velké plochy displeje tabletu viz obr. 6.9.

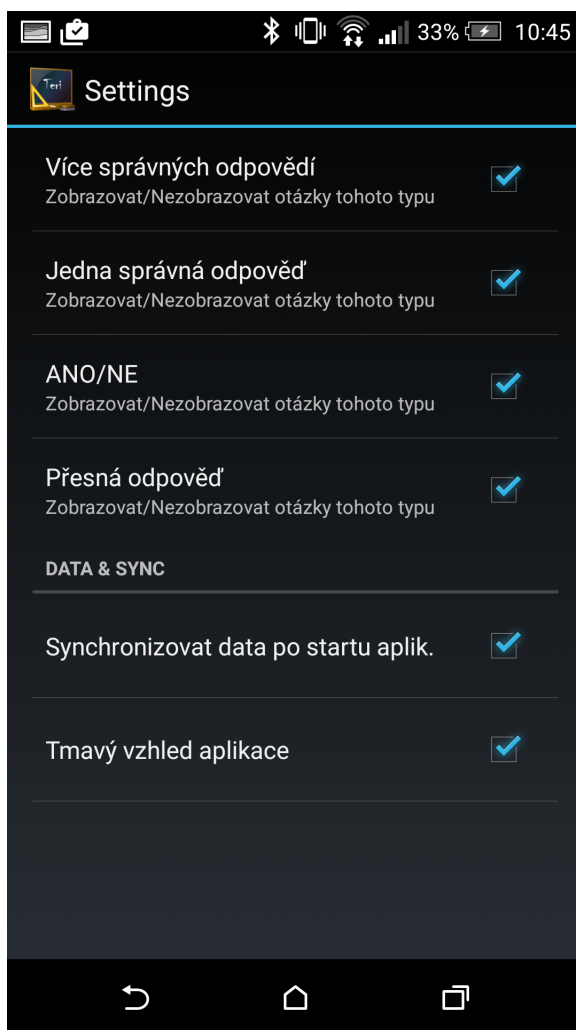


Obrázek 6.9: Nativní Android aplikace zobrazená na tabletu.

Aplikace využívá webové služby portálu Teri k synchronizaci dat. Jsou načítány identifikátory dat a jejich verze. To je pak v procesu synchronizace porovnáno s lokálními daty aplikace. V případě, že aplikace zjistí že má některá data neaktuální, tak si je prostřednictvím REST rozhraní stáhne. Detailnější popis komunikace aplikace s webovou službou naleznete v sekci 5.2. Pro synchronizační proces bylo vytvořeno samostatného asynchronního vlákna, které běží na pozadí nezávisle na uživatelském rozhraní. Uživatel tak může s aplikací během synchronizace normálně interagovat. Pro vlákno nebyly použity standardní mechanismy jazyka Java, ale bylo odděleno od třídy **AsyncTask**. Ta je součástí SDK Android od verze 1.5 [35]. Oproti klasickým vláknům přináší další abstrakci. Systém vlákna sám alokuje a udržuje frontu běžících úloh.

Veškerá data si aplikace ukládá do **SQLite** databáze, kterou poskytuje běhové prostředí systému. Pro pohodlnější práci byl v aplikaci naprogramován pokročilý objektový datový model s odpovídajícími manager třídami a databázovými fasádami.

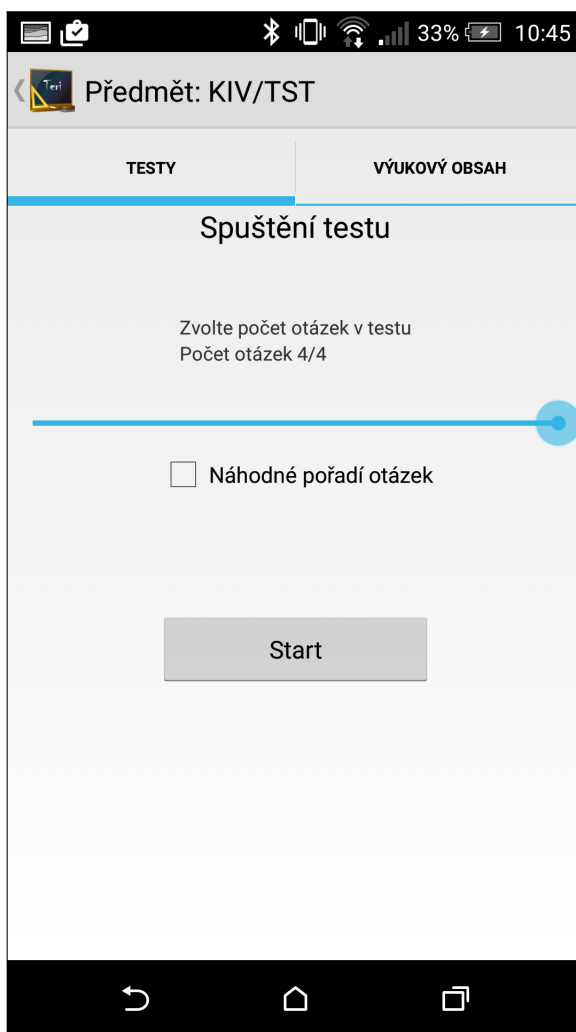
V hlavním kontextovém menu aplikace se nachází položka nastavení. Ta zobrazí uživateli obrazovku s nastavením aplikace viz obr. 6.10.



Obrázek 6.10: Nastavení android aplikace.

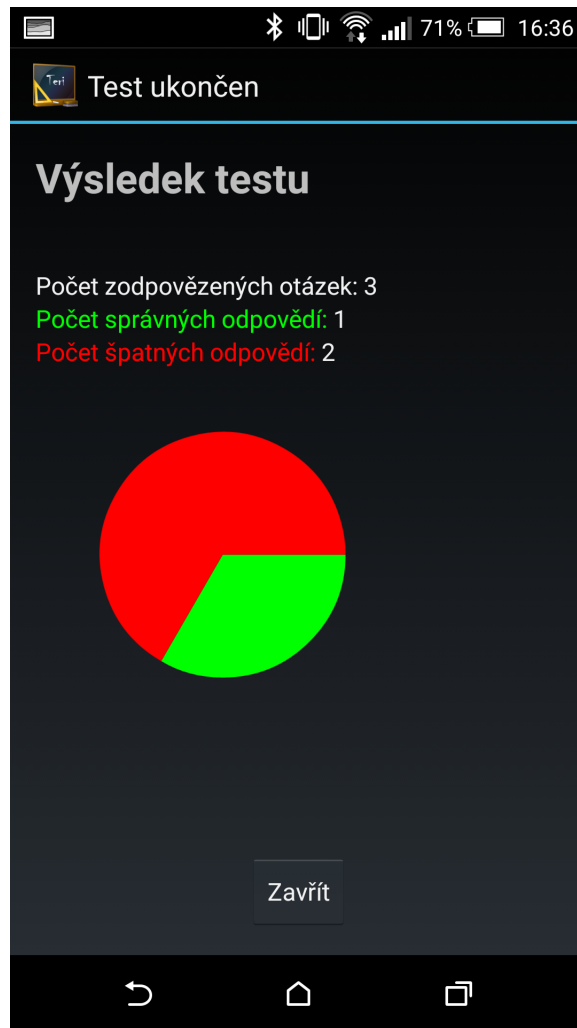
Zde je uživateli umožněno si zvolit jaké druhy otázek se mu budou v testech zobrazovat, zdali se po startu aplikace má provádět synchronizace dat se serverem a poslední položkou je volba vzhledu aplikace. Pro pohodlnější používání zařízení byla implementována možnost uživatelského přepnutí mezi tmavým a světlým téma vzhledu aplikace. Uživatel si tak může sám zvolit barevné rozložení, které mu více vyhovuje. Typicky při učení v nočních hodinách se například hodí tmavé téma které je šetrnější pro oči viz obr. 6.11.





Obrázek 6.11: Světlý vzhled android aplikace.

Při spouštění výukového testu se studentovi zobrazí posuvník pro volbu velikosti testu. Jedná se o komponentu **SeekBar**, které je při inicializaci nastavena maximální možná hodnota. Dále student může zaškrtnout *checkbox* pro náhodné proházení odpovědí. Během provádění samotného testu je student informován o správnosti odpovědi barevným podbarvením možností. Zeleně jsou podbarveny správné odpovědi a červeně chybné. V průběhu provádění testu jsou odpovědi zaznamenávány. Po dokončení testu se zobrazí statistiky testu s grafem udávajícím poměr správných a špatných odpovědí viz obr. 6.12.



Obrázek 6.12: Aktivita po ukončení testu.

Pro vykreslení grafu nebyla použita žádná externí knihovna. Byla implementována vlastní komponenta dědící od třídy **View**. Samotné vykreslení grafu bylo prováděno s třídami **Paint** a **Canvas**.

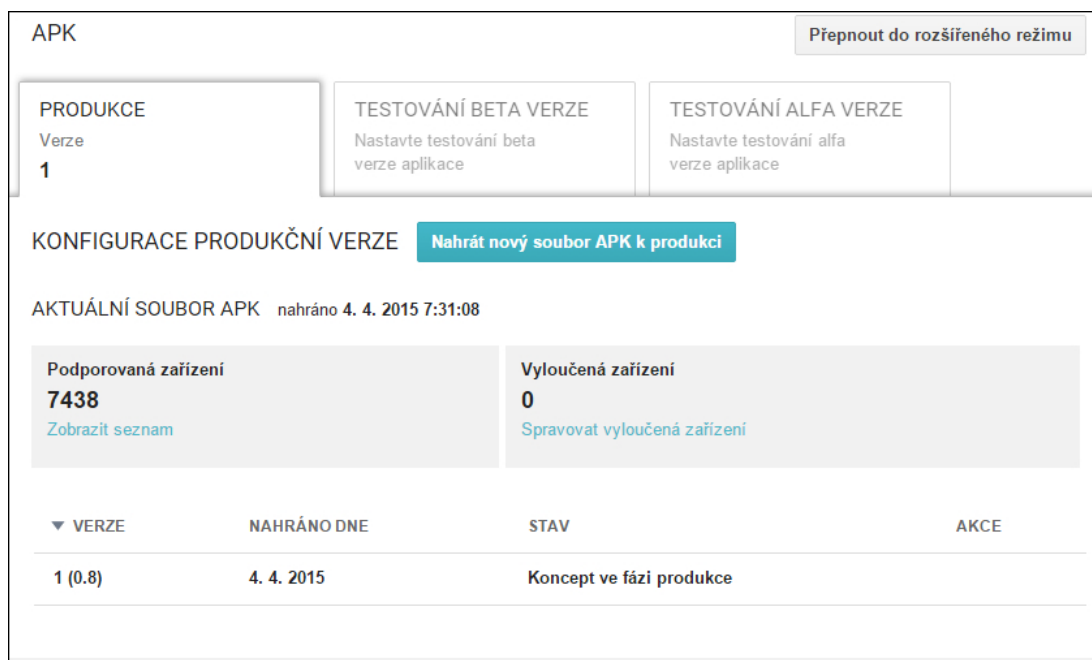
Výukový obsah je ukládán ve formě HTML článku. Proto bylo pro zobrazování těchto materiálů využito komponenty **WebView**.

### 6.3 Publikace mobilní aplikace

Po implementaci celé aplikace bylo potřeba provést publikaci. U původní staré aplikace se dal instalační **APK** balíček na web portálu ke stažení. Jedná se ale o nešťastné řešení. Prvním problémem je, že uživatel musel povolit ve svém zařízení instalaci aplikací z nedůvěryhodných zdrojů. To může spoustu uživatelů odradit. Druhý problém je distribuce aktualizací, uživatel si totiž musel sám stahovat nové

verze a ručně instalovat. Proto byla publikace nové verze aplikace zveřejněna v oficiálním obchodě **Google Play** [36].

Abych mohl zveřejňovat své aplikace na Google Play, tak jsem si musel zřídit vývojářský účet. Ten byl po zaplacení 25 USD aktivován. Po procesu aktivace je nutné aplikaci sestavit a podepsat privátním **JKS** klíčem. Tento klíč si vývojář musí pro danou aplikaci vygenerovat. Výstupem je podepsaný instalační APK balíček. Posledním krokem bylo nahrát mojí aplikaci přes vývojářskou konzoli Google Play viz obr. 6.13. Po nahrání aplikace je nutné vyplnit údaje o aplikaci a nahrát obrázky aplikace.









Obrázek 6.13: Vývojářská konzole Google Play.

Posledním povinným krokem, který je pro Google dost podstatný je určení vhodnosti aplikace z pohledu zakázaného a škodlivého obsahu. Proto je nutné vyplnit rozsáhlý dotazník. Na základě tohoto dotazníku vám Google vygeneruje osvědčení všech nejznámějších standardů pro určování škodlivosti obsahu na celém světě viz obr. 6.14. Konzole nabízí pro vývojáře spoustu užitečných informací. Jsou zde pokročilé analytické funkce pro sledování uživatelů aplikace.

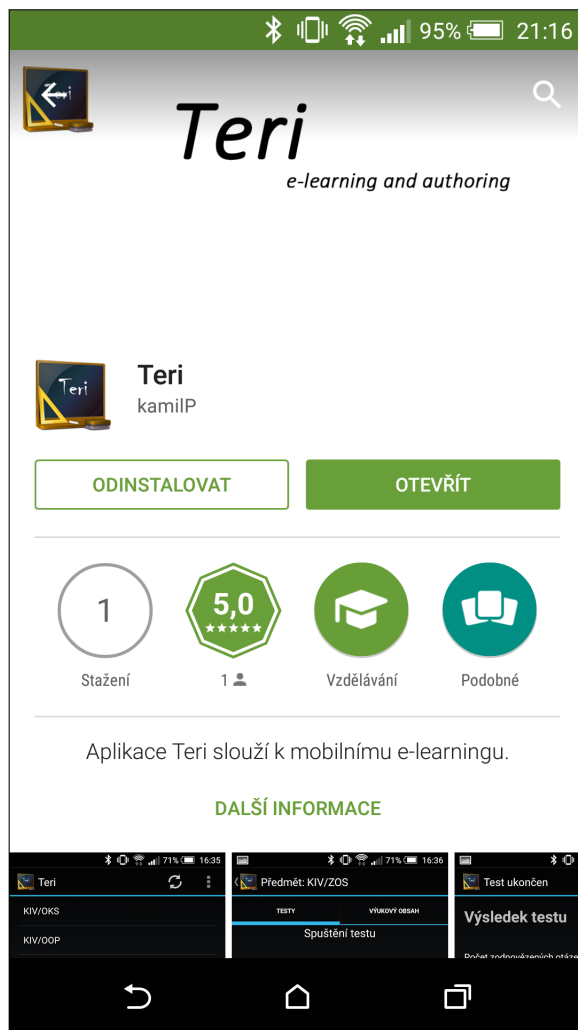
Dále jsou zde rozsáhlé možnosti jak aplikaci monetizovat. Google nabízí vložení reklamy do vaší aplikace, či její zpoplatnění. Při publikaci aplikace si můžete určit, zdali se jedná o produkční aplikaci anebo jí zveřejnit jako beta, či alfa verzi. U všech verzí si lze určit pro jaké regiony bude aplikace dostupná. Pro beta a alfa verze také například pro jaký počet uživatelů, nebo pro jaké konkrétní uživatele. Díky této funkcionalitě lze tedy velmi snadno vytvořit klasický systém pozvánek pro dobro-

volné testery. Poslední zajímavou funkcí, kterou zde zmíním je možnost lokalizace textů. Za poplatek si lze nechat přeložit texty popisující aplikaci, nebo samotnou aplikaci při poskytnutí resource souborů s texty.

VYPOČÍTANÉ HODNOCENÍ <a href="#">Další informace</a>	
Systém hodnocení	Kategorie hodnocení    Deskriptory
<b>Classificação Indicativa (ClassInd)</b> Brazílie	 Všechny věkové skupiny
<b>Entertainment Software Rating Board (ESRB)</b> Severní Amerika	 Všichni
<b>Pan-European Game Information (PEGI)</b> Evropa	 PEGI 3
<b>Unterhaltungssoftware Selbstkontrolle (USK)</b> Německo	 USK: všechny věkové skupiny
<b>IARC Generic</b> Ostatní oblasti	 Od 3 let
<b>Google Play</b> <a href="#">Další informace</a>	 Od 3 let

Obrázek 6.14: Hodnocení škodlivosti aplikace v Google Play.

Po nahrání první produkční verze aplikace trvá několik hodin, než Google rozdistribuuje tuto aplikaci ve všech zvolených regionech. V době nahrání aplikace Teri byla provedená automatická analýza a ve vývojářské konzoli bylo zobrazeno, že tato aplikace je kompatibilní se 7438 zařízeními. Naopak nekompatibilní zařízení s požadovanou verzí operačního systému Android nebylo nalezeno ani jedno. Po úspěšném dokončení distribuce byla aplikace veřejně dostupná ke stažení v Google Play viz obr. 6.15.



Obrázek 6.15: Aplikace Teri v obchodu Google Play.

V případě nahrání nových verzí APK balíčku přes vývojářskou konzoli došlo k automatické instalaci aktualizací na testovaných zařízeních do několika minut.

## 7 Automatické testování

Realizovaný systém je komplexní a rozsáhlou aplikací. Z důvodu udržitelnosti kvality a minimalizování chyb jsem se v rámci této práce zabýval automatickým testováním kódu a aplikací.

Automatické testování patří stále k podceňovaným částem vývoje software. Automatické testy manuální testy ve většině případů zcela nenahradí, ale často se k tomu blíží a v případech, které se dají těžko nasimulovat, mají dokonce automatické testy navrch. Je prokázáno, že automatické testy zvyšují kvalitu výsledného produktu a v konečném důsledku přinášejí úsporu času a peněz. Čím dříve se chyba nalezne, tím levnější je její odstranění. Psaní automatických testů nemusí být jen doplňkovou činností programátora, ale pro případ vývoje řízeného testy se může jednat o základní prvek vývoje aplikace.

Základní kategorie automatických testů:

- **Jednotkové testy**
- **Integrační testy**
- **Funkční testy**
- **Systémové testy**
- **Load testy**
- **Stress testy**

### 7.1 Jednotkové testy

Jednotkové testování má za úkol otestovat programové jednotky. Pod pojmem jednotka si můžeme představit fragment zdrojového kódu s pevně danou funkcionalitou, kterou chceme otestovat. Klasicky se jedná například o metody třídy. Snahou je testovat co nejmenší jednotky s konkrétní jednoznačnou funkcionalitou. Jednotkový test tedy správně nemá testovat co nejvíce funkcionality v jednom testu, ale naopak by mělo být cílem psát testy, které testují jenom jednu funkcionalitu kódu a žádnou jinou. [37]

Jednotkové testy by měly být nezávislé. Například by správně napsaný jednotkový test neměl být závislý na pořadí. Mělo by být tedy jedno v jakém pořadí se dané

jednotlivé jednotkové testy provedou. Testovací Unit frameworky mají implementovanou funkcionalitu pro určení pořadí jednotkových testů, které na sobě závisí. Obecně se ale dá říci, že takto závislé testy jsou špatně napsané testy.

Jednotkový test by správně neměl záviset na výsledku nebo výstupu jiného testu. Například že by jeden test nastavoval objekt, se kterým dále pracuje jiný test. Další zrádné závislosti na které by si měl dát programátor při psaní jednotkového testu pozor jsou datумы, náhodná čísla, síťová komunikace atp. Všechny takové veličiny a objekty by měl mít programátor dobře ošetřené. Možným řešením je si napsat vlastní API pro práci s datумы či jinými objekty v testech. Pokud programátor tyto závislosti neošetří tak hrozí, že testy budou závislé a může docházet k jejich selhání. Dokonce si lze představit situaci kdy budou testy některý den v týdnu procházet a jiný ne. A to vše jen kvůli špatné závislosti na datumu. Z toho logicky vyplývá, že správně napsaný jednotkový test by měl být nezávislý a měl by za každé situace po svém proběhnutí dávat stejné výsledky pro stejný testovaný kód a data. Jednotkový test tedy musí být deterministický.

Při psaní jednotkových, ale i integračních, funkčních a jiných testů je dobré dbát na kvalitu kódu testu. Test by měl být totiž snadno udržitelný. Je tedy vhodné dbát na kvalitu kódu testu minimálně stejně jako na kvalitu kódu ostrého. Dále by měla být snaha, aby byly testy krátké, dobře čitelné a jednoznačné. Test by měl testovat pouze jednu konkrétní funkcionalitu. Neměl by se snažit otestovat vše. Je lepší napsat více nezávislých testů pro více funkcí programu. Tím se docílí lepší čitelnosti a udržitelnosti testů.

V případě, že píšeme jednotkový test, tak nás zajímá pouze funkcionalita dané testované programové jednotky. V reálném programu ale často nějaký úsek kódu, například metoda či třída závisí na velkém množství jiných objektů či modulů. V těchto případech je v hodné testovaný kód od těchto závislostí odizolovat. Pro tyto případy se využívají falešné objekty tzv. Mocky. Pro vytváření mocků existuje ve všech běžných programovacích jazycích velké množství frameworků. Mezi nejznámější patří **Mockito**, **PowerMock**, **EasyMock**, **PHAKE** a **Mockery**.

Takto vytvořený falešný dvojník reálného objektu nám přináší výhodu v rychlosti. Není totiž třeba implementovat veškerou funkcionalitu závislého objektu, ale jen tu kterou potřebujeme. Jako programátor mocku si nadefinujeme veškeré jeho chování, proto můžeme nastavit prostřednictvím mocku i situace, které bychom jinak velmi těžko pro konkrétní závislosti nasimulovali. Například vrácení výjimek pro výpadek databázového serveru atp. [38] Dále tím, že máme nad mock objektem absolutní kontrolu a definujeme jeho chování přispíváme k tomu, aby byl test deterministický a tím pádem i opakovatelný. Máme totiž jistotu že daná závislost bude vykazovat pokaždé stejné chování, které jsme si určili. Docílíme tím tedy i zdokumentování pro jaké chování závislého objektu test projde, tedy je kód funkční.

Pro případ závislých objektů je díky mockování přístup k testování poměrně snadný. Horší situace však může nastat pro závislosti na jiných systémech či databázích. V takových případech je programátor většinou nucen překrýt API závislého systému pro účely testování. V případě databází se pak často používají například in-memory databáze. Taková databáze se vytvoří před zahájením testu v paměti. A po ukončení testu se opět zruší. Někdy programátoři v testu pracují s klasickou databází a pouze provádějí důkladný úklid, aby po testu nezůstala v databázi žádná data. Tento přístup však neřeší problém se závislostí na dostupnosti databázového serveru. Může pak nastat že testy budou díky takové závislosti neúspěšné i přesto že testovaný kód je správný. [37]

Na závěr bych zmínil i některé nevýhody jednotkových testů. Jako nevýhodu spatřuji fakt, že jednotkovým testem netestujeme provázanost testovaného kódu se zbytkem aplikace. Pro závislosti většinou používáme mocky takových objektů. Proto může nastat situace, že jednotkové testy procházejí ale v reálném nasazení aplikace nebude správně fungovat. Proto je nutné nespolehat pouze na jednotkové testy, ale doplnit je testy integračními a manuálním testováním nasazené aplikace.

## 7.2 Integrační testy

Zatímco jednotkové testy se snaží otestovat jenom funkcionalitu fragmentu kódu bez ohledu na jeho závislosti, tak integrační testy se snaží naopak provádět testy pro otestování správné funkčnosti vnitřních vazeb a závislostí objektů, modulů, systémů atp. Integračními testy se testují i závislosti na hardware či operační systém. Integrační testy se píšou až když máme naprogramováno více jednotek, které jsou samostatně otestovány a nyní je potřeba otestovat jejich bezchybnou provázanost. U složitých vícevrstevných aplikací mohou být integrační testy poměrně komplikované, ale dovolí vám otestovat vazby end-end.

## 7.3 Funkční testy

Funkční testování se zabývá testy aplikace či jejích částí z pohledu uživatelské role. Typicky se zde kontroluje správná funkčnost uživatelského rozhraní či webové služby. Testuje se zdali aplikace přechází do správných stavů a vrací správné informace.

Pro testování webových služeb je vhodný například nástroj **SoapUI**. S tímto nástrojem se dají vytvářet funkční testy pro otestování správného chování, ale i testy zátěžové, což je pro webové služby velmi důležitý test.



K testování uživatelského rozhraní existuje velké množství knihoven a nástrojů. Mezi nejznámější patří **Selenium**, **CasperJS**, nebo **Mocha**.

## 7.4 Výhody automatického testování aplikací

Zde bych ve zkratce rád shrnul základní výhody, které automatické testy přináší. Základní výhodou a funkcionalitou jednotkových testů je ověření správného chování testovaného kódu, dále si při psaní jednotkových testů ověříme správný design testovaného kódu. Dostáváme totiž okamžitou zpětnou vazbu o použitelnosti daného kódu, jeho čitelnosti a o správném pojmenování objektů a metod.

Dále je velkou výhodou automatických testů jejich rychlost a opakovatelnost. Provádění jednotlivých testů probíhá velice rychle. Obvykle jsou testy spouštěny v operační paměti a nepřistupují na disk. Testy je možné spouštět kdykoliv je potřeba. Což je základní výhodou oproti klasickému manuálnímu testování softwaru člověkem. S trochou nadsázky lze říci, že automatický test na rozdíl od člověka nemá žádnou pracovní dobu ani proměnlivou kvalitu svého testování.

Dalším velkým přínosem automatických testů je ten, že do budoucna fungují jako jakási záchranná síť. Pokud tedy v budoucnu provádíme změny v kódu aplikace nebo refactoring kódu, tak spuštěním testů ověříme, že naše změny neměly žádný negativní dopad na funkčnost ostatních částí aplikace. Čím více ostrého kódu aplikace máme pokryto kvalitními testy tím je tato záchranná síť lepší.

## 7.5 Psaní automatických testů

Při psaní testů by mělo platit pravidlo, že kód testu by měl být minimálně stejně kvalitní, jako kód ostrý. Testy je totiž třeba měnit v průběhu času tak jak se mění kód, který testují. Napsáním čistého a přehledného testu docílíme jeho snadnosti úpravy v budoucnu i pro případy, že bude test přepisovat jiný programátor. V ideálním případě by úprava testu neměla být časově o moc náročnější nežli úprava samotného ostrého kódu.

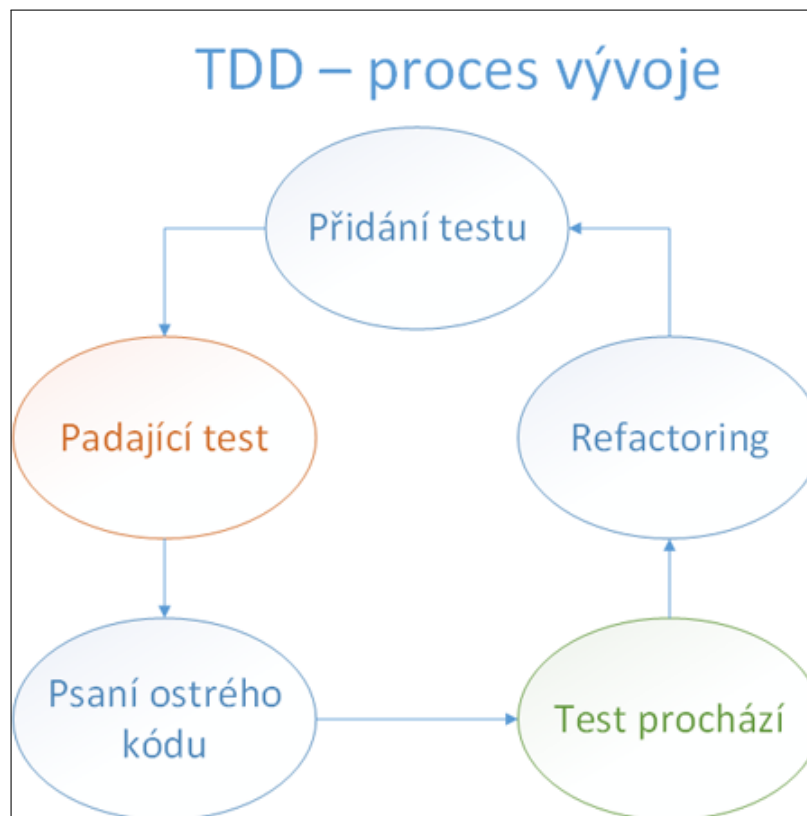
Často se stává, že špatně napsané testy způsobí v budoucnu velké problémy. Dochází pak k velkému nárůstu časových odhadů pracností změn v kódu a v krajních případech pak dojde k odstraňování či vypínání testů z důvodu nemožnosti jejich další udržitelnosti. Takovému případu je třeba se vyhnout a testy psát od začátku co nejvíce kvalitně a přehledně.

## 7.6 Vývoj řízený testy

TDD je zkratkou z anglického **Test Driven Development** a jedná se o metodiku vývoje řízenou testy.

Základní myšlenkou TDD je, že by měl programátor nejprve napsat testy a teprve až poté by měl napsat samotný kód. Dále programátor nepíše další kód ani testy do té doby, než stávající testy neprojdou což mu říká, že daná programová jednotka ostrého kódu vykazuje požadovanou vlastnost a chování. [39]

Výhodou této metody je, že jsou testy psány průběžně s kódem, čímž se docílí toho, že je téměř všechen zdrojový kód pokryt testy. Nevýhodou však je, že se testy píšou i na místech, kde jsou zbytečné a spíše prodlužují a tím i prodražují vývoj. Na to navazuje také problém s udržitelností. Udržování takového množství testů je do budoucna nákladné a náročné.



Obrázek 7.1: TDD

Proto se v praxi metoda TDD nedodrhuje většinou zcela striktně, ale třeba jen v modulech aplikace, kde má takové metoda vývoje svůj význam a opodstatnění. Například business logika aplikace, rozhraní aplikace či modulů, persistentní vrstva atd.

Velkými průkopníky v oblasti TDD jsou například velké open source projekty. Zde bývá obvyklé, že programátoři ke každé nové naprogramované funkcionalitě dodají také testy. Test pak v takovém případě slouží jako dokumentace korektního chování kódu.

## 7.7 Implementace testů

V rámci práce jsem si vyzkoušel jednotkové testování. V online verzi systému Teri jsem použil knihovnu **PHPUnit** [40]. Při použití této knihovny stačí aby testovací třída byla oddělena od třídy **PHPUnit\_Framework\_TestCase**. Takto vytvořený test umožňuje dělat asserce a knihovna po spuštění testu tyto asserce vyhodnotí.

Příklad asserce:

```
$this->assertEquals($vystup, $ocekavanyVystup);
```

Dále tato knihovna umožňuje použití hromadného volání testu z předpřipravených dat. Funkce, která vrací pole se vstupními daty pro test se nazývá **Data Provider**. Aby test věděl jakého providera má použít, tak je nutné v dokumentačním komentáři uvést jeho jméno:

```

1  /**
2   * 'YYYY-MM-DD' convert to 'DD.MM.YYYY'
3   *
4   * dataProvider datToStringCSDateProvider
5   */
6  public function testDatToStringCS($date, $ocekavanyVystup){
7      $vystup = DateUtils::datToStringCS($date);
8      // assert
9      $this->assertEquals($vystup, $ocekavanyVystup);
10 }
11 /**
12 * Data provider pro test testDatToStringCS
13 */
14 public function datToStringCSDateProvider()
15 {
16     return array(
17         array("2013-02-01", "01.02.2013"),
18         array("2013-02.01", ""),
19         array("0000-0000", ""),
20         array(null, ""),
21         array("0000.00.00", "")
22     );
23 }
```

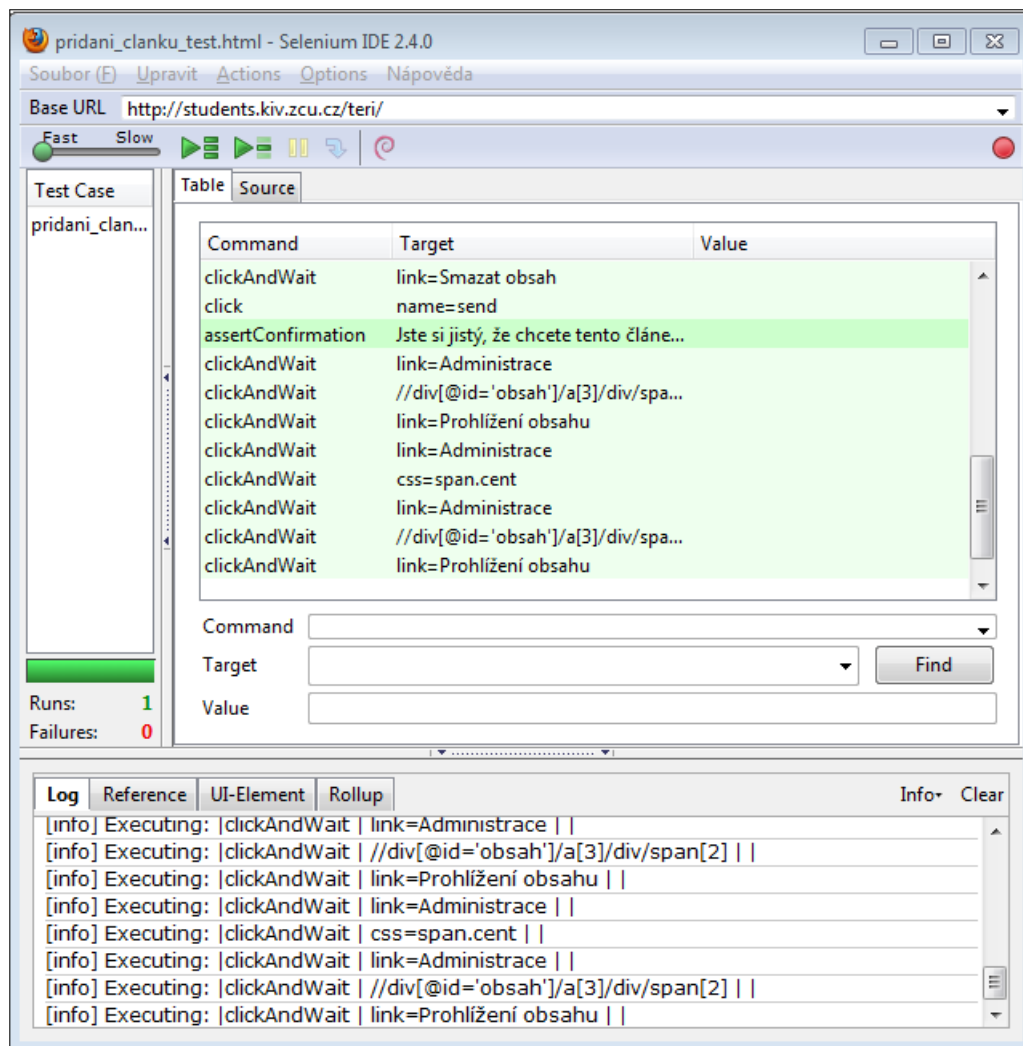
V ukázce je test `testDatToStringCS`, který testuje funkci `datToStringCS` ve třídě `DateUtils`. Vstupní data pro tento test vrací funkce `datToStringCSDateProvider`. Definice tohoto providera je proto v dokumentačním komentáři testu jako:

```
@dataProvider datToStringCSDateProvider
```

Byla napsána sada jednotkových testů pro otestování hlavní business logiky aplikace. V případě pokračování na tomto projektu bych rád pokračoval v psaní jednotkových testů pro co největší pokrytí zdrojového kódu aplikace.

Jednotkové testy jsem psal i při vývoji android aplikace. Zde jsem použil knihovnu **TestNG** [41], ta vychází ze známé knihovny **JUnit**, která je součástí JDK. Přináší však malé výhody zejména pro přípravu a úklid testu. TestNG také lépe pracuje s operační pamětí.

Kromě jednotkového testování jsem implementoval testy funkční. Pro automatické funkční testování jsem použil nástroj **Selenium** [42]. Tento zdarma dostupný nástroj patří mezi nejznámější nástroje pro tvorbu funkčních testů. Umožňuje spouštět testy na jádrech webových prohlížečů. Já jsem pro své testy použil jádro webového prohlížeče **Mozilla Firefox**. Firefox driver je v základu knihovnou Selenium podporován. Takto napsaný test otevře prohlížeč Firefox a automaticky kliká na webové stránce místo uživatele. Ověří tak základní funkčnost aplikace, lze zde také provádět asserce, které ověří zdali se na stránce vypisují správné hlášky atp. Funkční test má oproti testování uživatelem výhodu zejména v rychlosti a opakovatelnosti. Pokud bychom si vytipovali několik desítek základních scénářů pro kritické testy aplikace, tak lze během několika sekund zjistit zdali jsme nějakou úpravou kódu nenarušili základní funkčnost aplikace. Bylo by totiž velmi nákladné po každé větší úpravě provádět ruční otestování veškeré kritické funkcionality aplikace.



Obrázek 7.2: Selenium IDE.

Selenium testy lze spouštět v **Selenium IDE** viz obr. 7.2. Tento nástroj lze doinstalovat do prohlížeče Firefox jako vývojářský doplněk. Usnadní programátorovi tvorbu testu a umožňuje tyto testy také spouštět. Dále si lze testy vyexportovat jako Unit test do libovolného z podporovaných programovacích jazyků. Například si můžeme testy vyexportovat do jazyka Java a tyto testy pak hromadně automaticky spouštět stejně jako spouštíme jednotkové testy.

## 8 Funkčnost řešení a statistiky

Po první provozuschopné implementaci nové verze webového portálu Teri byl systém nasazen na server **students.kiv.zcu.cz/teri**, kde předtím běžela jeho stará verze. Nasazení bylo provedeno v červenci roku 2014. Díky tomu byl systém dlouhou dobu v ostrém provozu. Systém byl tedy odladěn na reálných uživateliích a bylo nasbíráno velké množství statistických informací, které budou prezentovány v této kapitole. Prezentované statistiky byly nasbírány od 1.7.2014 do 5.4.2015.

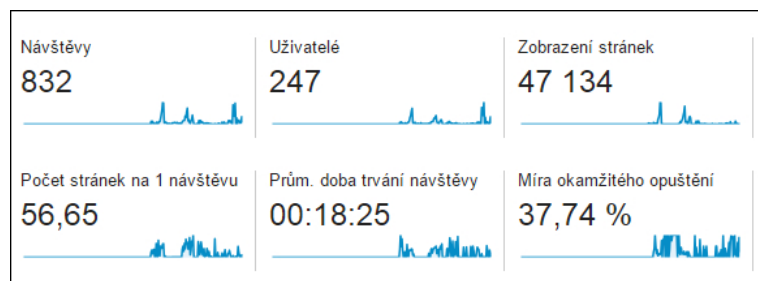
Webová verze systému byla testována na prohlížečích Google Chrome (verze 42), Mozilla Firefox (verze 37.0.1), Internet Explorer (verze 11), Opera (verze 12.14) na operačních systémech Windows 8.1, Windows 7 a Ubuntu 14.04. Dále bylo provedeno také testování na operačním systému OS X Yosemite s poslední verzí prohlížeče Safari 8.0.5.

Pan Ing. Ladislav Pešička již dříve systém naplnil výukovými otázkami pro předmět **Základy operačních systémů** (KIV/ZOS). Studenti byli upozorněni na možnost online výuky, což se pozitivně projevilo na významné návštěvnosti. Výsledný systém byl podroben ověřovacímu provozu na reálných uživateliích. Toto nasazení při výuce se ukázalo jako vynikající zpětná vazba při vývoji.

Během téměř ročního provozu systém nezaznamenal žádné problémy s funkčností ani stabilitou. Problémy se neprojevovaly ani během zkouškového období kdy byla zátěž nejvyšší.

Statistiky vztahující se k samotné výuce si systém ukládá a zpracovává sám. Pro zaznamenání ostatních statistik přístupů byla použita služba **Google Analytics** [43]. Jedná se o zdarma dostupný nástroj pro pokročilou webovou analytiku.

Za dva semestry provozu bylo na portále Teri přes 800 návštěvníků (odhadovaných 250 reálných uživatelů) a bylo zobrazeno téměř 50 tisíc stránek viz obr. 8.1.



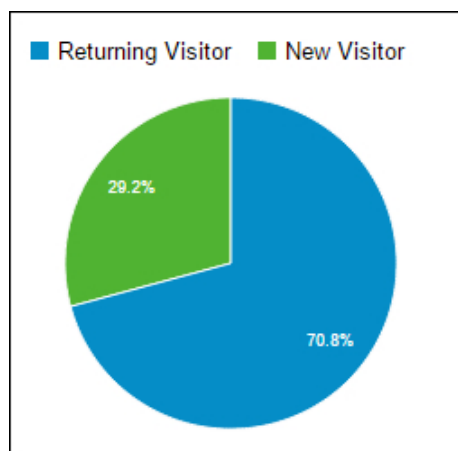
Obrázek 8.1: Hlavní statistika návštěvnosti webu.

Průměrná doba trvání jedné návštěvy se blíží dvaceti minutám. Jedná se na poměry webových stránek na relativně vysoké číslo. Lze z toho usuzovat že velká část návštěvníků se na portále opravdu učila. Tomu nasvědčuje také statistika počtu stránek na jednu návštěvu, která má hodnotu 56 stránek. Detailnější zobrazení statistik doby setrvání na stránce viz obr. 8.2. Na ní je hezky vidět, že uživatelé kteří na webu stráví delší množství času jsou aktivní a proklikají velké množství stránek.

Doba trvání návštěvy	Návštěvy	Zobrazení stránek
Počet sekund: 0-10	344	395
Počet sekund: 11-30	30	150
Počet sekund: 31-60	26	161
Počet sekund: 61-180	58	700
Počet sekund: 181-600	72	1 924
Počet sekund: 601-1800	120	9 495
Počet sekund: 1801+	182	34 309

Obrázek 8.2: Detailní statistika opuštění webu.

Poměr nových návštěvníků nám ukazuje, že téměř 3/4 uživatelů se na web vrací pravidelně viz obr. 8.3.

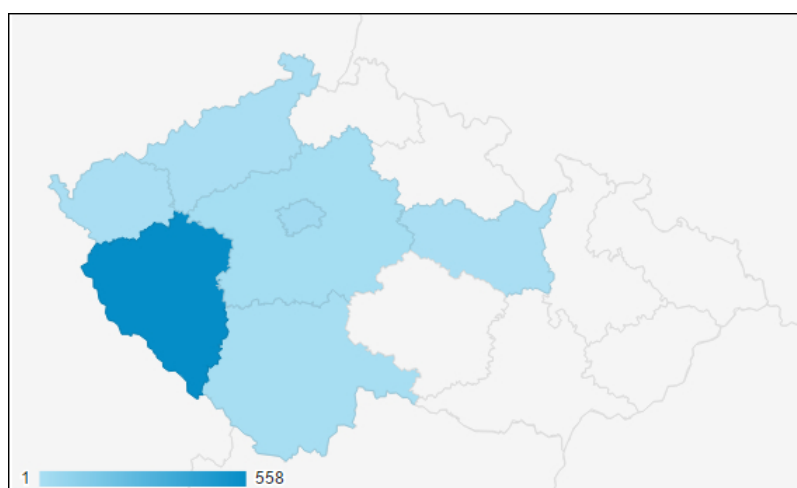


Obrázek 8.3: Graf poměru nových a vracejících se návštěvníků.

Google Analytics nám také nabízí geografické statistiky viz obr. 8.4. Ty nepřinášejí žádné překvapení. Systém běží pro výuku na ZČU, proto je majoritní počet návštěv z Plzně (přes 50%). Další nejčastější lokality jsou velká města v západní části Čech viz obr. 8.5. Jediným nečekaným městem je Moskva. I z detailních přístupových logů bylo zjištěno, že největší počet webových robotů prohledávající stránky bylo z Ruska.

Město	Návštěvy	Návštěvy v %
1. Plzeň	433	52,04 %
2. Rokycany	116	13,94 %
3. (not set)	58	6,97 %
4. Moscow	56	6,73 %
5. Prague	28	3,37 %
6. Samara	10	1,20 %
7. Klatovy	9	1,08 %
8. Zelenec	8	0,96 %
9. Tabor	6	0,72 %
10. Píbram	5	0,60 %

Obrázek 8.4: Tabulka nejčastějších geografických lokalit návštěv webu.



Obrázek 8.5: Geografická mapa přístupů na web.

Z hlediska vývoje byla zajímavá tabulka ukazující zastoupení webových prohlížečů mezi návštěvníky webu viz obr. 8.6. Jednoznačně majoritní postavení má prohlížeč **Google Chrome**, který je následovaný prohlížečem **Mozilla Firefox**. Ostatní prohlížeče mají minimální zastoupení. Web byl testován na nejčastěji používaných prohlížečích posledních verzí, díky této statistice ale byla největší pozornost při testování věnována právě těmto dvěma.



Prohlížeč	Návštěvy	Návštěvy v %
1. Chrome	530	63,70 %
2. Firefox	237	28,49 %
3. Opera	49	5,89 %
4. Safari	7	0,84 %
5. Internet Explorer	5	0,60 %
6. Android Browser	4	0,48 %

Obrázek 8.6: Statistika zastoupení webových prohlížečů.

Podobnou statistikou je dále tabulka zobrazující operační systémy návštěvníků viz obr. 8.7. Na první příčce je operační systém **Windows**. V aktuálním měření je zastoupen u 66 % uživatelů. Pro zajímavost u staré verze systému Teri měl tento systém četnost 91 %. Je tedy vidět, že na úkor stále více oblíbeného **Mac OS** začíná operační systém Microsoftu ztrácet. Také oproti měření na staré verzi webu posílil systém **Linux**. Ten se svými 18 % ukazuje na jeho stále velkou oblíbenost zejména mezi akademiky a zkušenými uživateli.

Zajímala nás také návštěvnost na mobilních platformách. Zde bylo nejvíce návštěv z operačního systému **Android** (73 %). Zbylé mobilní návštěvy byly ze systému **iOS**. Z operačního systému Windows Phone nebyla učiněna ani jedna návštěva.

Operační systém	Návštěvy	Návštěvy v %
1. Windows	551	66,23 %
2. Macintosh	152	18,27 %
3. Linux	106	12,74 %
4. Android	17	2,04 %
5. iOS	6	0,72 %

Obrázek 8.7: Operační systémy návštěvníků webu.

Další zajímavá čísla nám dává statistika ukazující na poskytovatele internetového připojení návštěvníků. Na prvním místě se umístila **Západočeská Univerzita**. Z toho se dá usuzovat, že velká část studentů se učí během volných chvil ve škole.

Poskytovatel služeb	Návštěvy	Návštěvy v %
1. zapadoceska univerzita	155	18,63 %
2. arenis s.r.o.	140	16,83 %
3. xdsl network-adsl	74	8,89 %
4. hosting telesystems network	55	6,61 %
5. (not set)	38	4,57 %
6. upc ceska republika a.s.	28	3,37 %
7. pilsfree.net member workstations nat 1	23	2,76 %
8. telefonica czech republic a.s.	20	2,40 %
9. upc ceska republika s.r.o.	19	2,28 %
10. internet 7d - jaroslav novak	18	2,16 %

Obrázek 8.8: Poskytovatelé internetového připojení návštěvníků webu.

Poslední statistikou z Google Analytics kterou zde popíši je graf chování uživatelů na webu Teri viz obr. 8.9. Z ní je vidět, že web je dostatečně přímočarý a umožňuje na dvě kliknutí spouštět výukové testy. Ty jsou dle grafu nejžádanější částí webu mezi návštěvníky.



Obrázek 8.9: Graf chování uživatelů.

Následující statistiky se vztahují k předmětu **KIV/ZOS**. Jedná se o statistiky ukládané a zobrazované modulem systému Teri. Vztahují se ke statistikám výukových testů. Statistika pro články byly implementovány až ke konci vývoje a čísla by nebyla relevantní.

Za dva semestry bylo spuštěno **2144** výukových testů, během nichž bylo zodpovězeno téměř **74** tisíc otázek. Kompletně dokončeno bylo více než **700** testů. Jedná se o vysoká čísla. Studenti portál pro výuku tohoto předmětu nejvíce využívali během zimního semestru, kdy je tento předmět vyučován. Dále pak před termíny státních závěrečných zkoušek viz obr. 8.10.



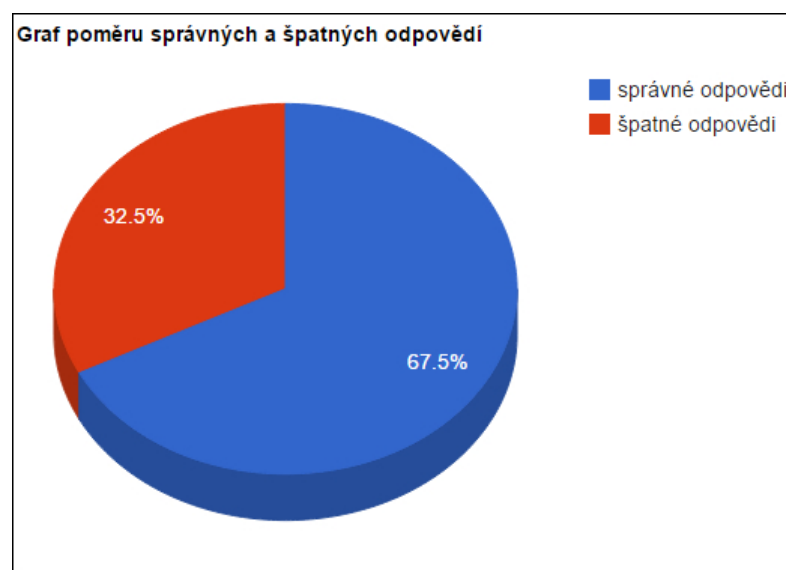
Obrázek 8.10: Graf počtu výukových testů předmětu KIV/ZOS během roku.

Tento graf dále doplňuje další, který vykresluje učení v průběhu dne viz obr. 8.11. Je vidět, že studenti nejvíce využívají e-learningové testy v odpoledních a nočních hodinách. Naopak v brzkých ranních hodinách se neučí nikdo.



Obrázek 8.11: Graf počtu výukových testů předmětu KIV/ZOS během dne.

Poslední statistikou je graf poměru správných a špatných odpovědí viz obr. 8.12. Na něm je vidět, že je u tohoto předmětu obtížnost otázek dobře vyvážená.



Obrázek 8.12: Graf poměru správných a špatných odpovědí na otázky z KIV/ZOS.

## 9 Možnosti dalšího rozšíření systému

Vytvořený systém je již v dlouhodobém provozu. Dá se předpokládat jeho širší využití při výuce dalších předmětů. Bude vhodné systém dále rozšiřovat a reagovat na potřeby studentů a vyučujících.

Možným rozšířením systému by mohlo být vytvoření dalších nativních aplikací pro nejrozšířenější mobilní platformy. V současnosti je implementována pouze mobilní aplikace pro operační systém Android. Implementace aplikací pro systémy **iOS** a **Windows Phone** by pokryla veškerá chytrá mobilní zařízení mezi studenty.

Z pohledu synchronizace dat by tyto nové mobilní aplikace mohly využívat již vytvoření REST API. To bylo implementováno a ověřeno v rámci této diplomové práce a má univerzální využití.

Dále by bylo možné systém propojit s již zavedeným portálem ZČU. Přínos v takové integraci vidím zejména ve sjednocení systémů souvisejících s výukou. Zejména napojení aplikace Teri na přihlašovací systém **WebAuth** ZČU. Stávající systém přihlašování a registrací by byl zachován pro externí výuku či školení osob, které nestudují na ZČU. Přihlašování přes WebAuth by pak využili studenti a vyučující, kterým by odpadla nutnost registrace a v rámci systému by byla snazší jejich identifikace a ověření pravosti.

Velký potenciál má modul ostrých testů, proto si myslím, že by tento modul mohl být také dále rozšířen o pokročilé možnosti vyhodnocení testů a analýzy výsledků. Pro vyučující by mohlo být zajímavé, kdyby systém uměl spojovat ostré testy do logických celků a tyto celky vyhodnocovat. Například, že by v rámci semestru bylo vytvořeno několik ostrých testů. Systém by pak sám uměl sjednotit výsledky studentů z jednotlivých testů a dát vyučujícímu informaci, zdali student splnil podmínky pro získání zápočtu.

## 10 Závěr

V rámci diplomové práce byly prozkoumány vybrané systémy a aplikace určené pro e-learning a tvorbu výukového obsahu. Jedním z hlavních kritérií, které byly v průzkumu zohledněny, byla použitelnost na mobilních zařízeních.

Na základě průzkumu a analýzy byl rozšířen stávající výukový portál Teri. Byl vytvořen modul pro e-learningový obsah, rozšířeny výukové testy a kompletně přepracován a rozšířen modul ostrých testů.

Hlavním cílem celé práce bylo, aby byl systém co nejvíce multiplatformní pro použití na mobilních zařízeních. Byl kompletně přepracován vzhled celé webové aplikace. Nový responzivní design zaručuje možnost použití na zařízeních s různými velikostmi obrazovky. Aby byla zaručena ještě lepší použitelnost na mobilních zařízeních, tak byla v rámci práce implementována nativní aplikace pro operační systém Android.

Výsledný rozšířený výukový portál a mobilní aplikaci se podařilo implementovat v plném rozsahu stanovených cílů. Bylo vytvořeno a rozšířeno velké množství modulů pro snadnou práci s výukovým obsahem a testy. Výsledná aplikace byla nasažena pro reálný provoz v prostředí katedry. Jedná se tedy o již používanou aplikaci s možností dalšího rozšíření a dlouhodobého provozu.

Mezi hlavní přednosti realizovaného portálu je jeho snadná instalace i použitelnost. Další výhodou je jeho kompaktnost, díky které na jednom místě v jednotném rozhraní dostanou studenti nástroj, obsahující vše potřebné pro e-learning. Systém běží na adrese <http://students.kiv.zcu.cz/teri/> a během ročního testovacího provozu bylo zaznamenáno více jak 800 návštěv.

## Seznam obrázků

2.1	Elektronická učebnice nakladatelství Fraus. . . . .	7
2.2	Jak z pohledu učitelů splnila elektronická učebnice jejich očekávání? .	8
2.3	Samsung smart school.[5] . . . . .	9
2.4	Android aplikace Moodle. . . . .	10
2.5	Android aplikace eDoceo. . . . .	11
3.1	Ukázka WYSIWYG editoru NicEdit. . . . .	12
3.2	Využití mobilních zařízení na sociálních sítích.[14] . . . . .	14
3.3	Graf použití nativních mobilní aplikací a mobilních webů.[18] . . . . .	15
5.1	Editor pro tvorbu výukových článků. . . . .	19
5.2	Ukázka modulu pro nahrávání obrázků do článku. . . . .	20
5.3	Rozšířené nastavení při tvorbě článku. . . . .	20
5.4	Rozbalovací formulář pro zpětnou vazbu. . . . .	21
5.5	Dynamický formulář pro nahrávání příloh. . . . .	22
5.6	Schéma komunikace mobilního zařízení se serverem Teri. . . . .	24
5.7	Schéma verzování aplikačních dat. . . . .	29
5.8	Čtení výukových článků během dne. . . . .	30
5.9	Čtení výukových článků během roku. . . . .	30
5.10	Tabulka pro četnost zobrazení článků. . . . .	31
5.11	Tabulka pro četnost zobrazení článků. . . . .	31
5.12	Uživatelské hodnocení článku. . . . .	31
5.13	Graf hodnocení článků. . . . .	32
5.14	Formulář pro tvorbu seřazovací otázky. . . . .	33
5.15	Zobrazená seřazovací otázka. . . . .	34

5.16	Vyhodnocení seřazovací otázky. . . . .	35
5.17	Hodiny zobrazující zbývající čas do odevzdání testu. . . . .	36
5.18	Tabulka výsledků ostrého testu. . . . .	37
5.19	Volba zahrnutí skrytých otázek při tvorbě ostrého testu. . . . .	37
5.20	Kontextová nápověda po najetí kurzoru na ikonku otazníku. . . . .	38
5.21	Ukázka vzhledu sjednocených notifikací. . . . .	39
5.22	Drobečková navigace. . . . .	40
5.23	Formát importních dat pro hromadné uložení studentů. . . . .	40
5.24	Informační výpis při hromadném importu studentů. . . . .	41
5.25	Možnost skrytí předmětu a jeho opětovné zapnutí. . . . .	41
6.1	Původní verze systému Teri - klasické zobrazení. . . . .	42
6.2	Původní verze systému Teri - mobilní verze webu. . . . .	43
6.3	Nový vzhled Teri - zobrazení na desktopovém prohlížeči. . . . .	44
6.4	Nový vzhled Teri - zobrazení na tabletu (Xiaomi Mi Pad). . . . .	45
6.5	Nový vzhled Teri - zobrazení se skrytým menu na tabletu (Xiaomi Mi Pad). . . . .	46
6.6	Nový vzhled Teri - zobrazení otázky na tabletu (Xiaomi Mi Pad). . . . .	47
6.7	Nový vzhled Teri - zobrazení na mobilním telefonu (HTC One M8). . . . .	48
6.8	Nativní Android aplikace zobrazená na chytrém telefonu. . . . .	50
6.9	Nativní Android aplikace zobrazená na tabletu. . . . .	51
6.10	Nastavení android aplikace. . . . .	52
6.11	Světlý vzhled android aplikace. . . . .	53
6.12	Aktivita po ukončení testu. . . . .	54
6.13	Vývojářská konzole Google Play. . . . .	55
6.14	Hodnocení škodlivosti aplikace v Google Play. . . . .	56



6.15	Aplikace Teri v obchodu Google Play. . . . .	57
7.1	TDD . . . . .	62
7.2	Selenium IDE. . . . .	65
8.1	Hlavní statistika návštěvnosti webu. . . . .	66
8.2	Detailní statistika opuštění webu. . . . .	67
8.3	Graf poměru nových a vracejících se návštěvníků. . . . .	67
8.4	Tabulka nejčastějších geografických lokalit návštěv webu. . . . .	68
8.5	Geografická mapa přístupů na web. . . . .	68
8.6	Statistika zastoupení webových prohlížečů. . . . .	69
8.7	Operační systémy návštěvníků webu. . . . .	69
8.8	Poskytovatelé internetového připojení návštěvníků webu. . . . .	70
8.9	Graf chování uživatelů. . . . .	70
8.10	Graf počtu výukových testů předmětu KIV/ZOS během roku. . . . .	71
8.11	Graf počtu výukových testů předmětu KIV/ZOS během dne. . . . .	72
8.12	Graf poměru správných a špatných odpovědí na otázky z KIV/ZOS. . . . .	72

## Seznam tabulek

5.1	Popis webové služby . . . . .	24
-----	-------------------------------	----

## Reference

- [1] William Horton. *e-Learning by Design*. Amazon ISBN 978-0470900024, 2. vydání edition, 2011.
- [2] Blended learning [online], cit. 12.3.2004. Informační zdroj dostupný z URL: <<http://www.csvs.cz/aula/clanky/06-2004-3-blended-learning.pdf>>.
- [3] Jižná Korea do roku 2015 nahradí všechny papírové učebnice tablety [online], cit. 7.7.2011. Informační zdroj dostupný z URL: <<http://goo.gl/GA80m8>>.
- [4] One-to-one Tablets in Secondary Schools [online], cit. 4.12.2012. Informační zdroj dostupný z URL: <[http://www.e-learningfoundation.com/Websites/elearningfoundation/images/PDF%20Documents/Honeywood\\_2012\\_Barbie\\_Clark.pdf](http://www.e-learningfoundation.com/Websites/elearningfoundation/images/PDF%20Documents/Honeywood_2012_Barbie_Clark.pdf)>.
- [5] Samsung Smart School [online], cit. 2.3.2014. Informační zdroj dostupný z URL: <<http://technomania25.blogspot.cz/2013/05/samsung-smart-school.html>>.
- [6] Moodle LMS [online], cit. 31.3.2015. Informační zdroj dostupný z URL: <<https://moodle.org/>>.
- [7] eDoceo [online], cit. 10.4.2015. Informační zdroj dostupný z URL: <<http://www.edoceo.cz/>>.
- [8] TinyMCE - WYSIWYG editor [online], cit. 28.3.2015. Informační zdroj dostupný z URL: <<http://www.tinymce.com/>>.
- [9] NicEdit[online], cit. 11.4.2015. Informační zdroj dostupný z URL: <<http://nicedit.com/>>.
- [10] CKEditor [online], cit. 11.4.2015. Informační zdroj dostupný z URL: <<http://ckeditor.com/>>.
- [11] Twitter usage [online], cit. 10.11.2014. Informační zdroj dostupný z URL: <<https://about.twitter.com/company>>.
- [12] Facebook's new stats: 1.32 billion users, 30 percent only use it on their phone [online], cit. 23.7.2014. Informační zdroj dostupný z URL: <<http://goo.gl/iip6Nt>>.
- [13] Facebook reports 1 billion active users on mobile devices and 200 million users on Instagram [online], cit. 25.3.2014. Informační zdroj dostupný z URL: <<http://goo.gl/CHJg1N>>.

- [14] How Mobile Are Social Networks? [online], cit. 31.3.2014. Informační zdroj dostupný z URL: <http://www.statista.com/chart/2091/mobile-usage-of-social-networks/>.
- [15] Mobile First Design: Why It's Great and Why It Sucks [online], cit. 5.3.2013. Informační zdroj dostupný z URL: <http://designshack.net/articles/css/mobilefirst/>.
- [16] Mobilní, nebo responzivní web? [online], cit. 11.4.2015. Informační zdroj dostupný z URL: <http://www.artweby.cz/blog/mobilni-stranky-nebo-responzivni-web>.
- [17] How to decide between a responsive website or a native mobile app [online], cit. 8.2.2014. Informační zdroj dostupný z URL: <http://thenextweb.com/dd/2014/02/08/decide-responsive-website-native-mobile-app/>.
- [18] Apps Continue to Dominate the Mobile Web [online], cit. 5.3.2013. Informační zdroj dostupný z URL: <http://www.mobinett.com/2014/04/29/>.
- [19] Diplomová práce - REST a webové služby v jazyce Java [online], cit. 20.5.2010. Informační zdroj dostupný z URL: [https://is.muni.cz/th/143417/fi\\_m/143417\\_kadlec\\_jiri\\_diplomova\\_prace.txt](https://is.muni.cz/th/143417/fi_m/143417_kadlec_jiri_diplomova_prace.txt).
- [20] Architectural Styles and the Design of Network-based Software Architectures [online], cit. 20.5.2010. Informační zdroj dostupný z URL: [https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding\\_dissertation.pdf](https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation.pdf).
- [21] Kamil Praum. *Bakalářská práce - Systém pro online testování s podporou mobilních zařízení*. Plzeň, 2012.
- [22] JBIimages - image upload plugin for TinyMCE [online], cit. 28.3.2015. Informační zdroj dostupný z URL: <http://justboil.me/>.
- [23] mPDF - PHP class which generates PDF files from UTF-8 encoded HTML [online], cit. 29.3.2015. Informační zdroj dostupný z URL: <http://www.mpdf1.com/mpdf/index.php>.
- [24] Marián Margorín. *jQuery bez předchozích znalostí*. Computer Press ISBN 978-80-251-3379-8, Brno, 2011.
- [25] jQuery UI - Toggle [online], cit. 29.3.2015. Informační zdroj dostupný z URL: <https://jqueryui.com/toggle/>.
- [26] Google Charts [online], cit. 26.3.2015. Informační zdroj dostupný z URL: <https://developers.google.com/chart/>.
- [27] jQuery UI - Sortable [online], cit. 28.3.2015. Informační zdroj dostupný z URL: <https://jqueryui.com/sortable/>.

- [28] FlipClock.js [online], cit. 30.3.2015. Informační zdroj dostupný z URL: <<http://flipclockjs.com/>>.
- [29] Drobecková navigace na webu [online], cit. 4.8.2006. Informační zdroj dostupný z URL: <<https://www.interval.cz/clanky/drobeckova-navigace-na-webu/>>.
- [30] Tim Kadlec. *Responzivní design Profesionálně*. Zoner Press ISBN 978-80-7413-280-3, 1. vydání edition, 2014.
- [31] Bootstrap [online], cit. 19.3.2015. Informační zdroj dostupný z URL: <<http://getbootstrap.com/>>.
- [32] PhoneGap [online], cit. 6.4.2015. Informační zdroj dostupný z URL: <<http://phonegap.com/>>.
- [33] React Native [online], cit. 6.4.2015. Informační zdroj dostupný z URL: <<http://www.reactnative.com/>>.
- [34] Smartphone OS Market Share, Q4 2014 [online], cit. 6.4.2015. Informační zdroj dostupný z URL: <<http://www.idc.com/prodserv/smartphone-os-market-share.jsp>>.
- [35] Grant Allen. *Android 4*. Computer Press ISBN 978-80-251-3782-6, Brno, 2013.
- [36] Teri - Google Play [online], cit. 6.4.2015. Informační zdroj dostupný z URL: <<https://play.google.com/store/apps/details?id=cz.praum.teri.app>>.
- [37] Robert C. Martin. *Čistý kód*. Computer Press ISBN 978-80-251-2285-3, Brno, 2009.
- [38] Tomek Kaczanowski. *Bad Tests, Good Tests*. kaczanowscy.pl Tomasz Kaczanowski ISBN 978-83-938471-0-5, 2013.
- [39] Nat Pryce Steve Freeman. *Growing Object-Oriented Software, Guided by Tests*. Pearson Education ISBN 978-0-321-50362-6, 2010.
- [40] PHPUnit [online], cit. 4.3.2014. Informační zdroj dostupný z URL: <<http://phpunit.de/>>.
- [41] TestNG [online], cit. 31.3.2014. Informační zdroj dostupný z URL: <<http://testng.org/>>.
- [42] Selenium [online], cit. 5.2.2014. Informační zdroj dostupný z URL: <<http://docs.seleniumhq.org/>>.

- [43] Google Analytics [online], cit. 31.3.2015. Informační zdroj dostupný z URL: [<http://www.google.com/analytics/>](http://www.google.com/analytics/).
- [44] Steve Suehring. *JavaScript krok za krokem*. Computer Press ISBN 978-80-251-2241-9, Brno, 1. vydání edition, 2008.
- [45] Derick Rethans Andi Gutmans, Stig Saether Bakken. *Mistrovství v PHP 5*. Computer Press ISBN 978-80-251-1519-0, Brno, 2008.
- [46] Luboslav Lacko. *PHP 5 a MySQL 5 Hotová řešení*. Computer Press ISBN 978-80-251-1695-1, Brno, 2007.
- [47] Jakub Vrána. *1001 Tipů a triků pro PHP*. Computer Press ISBN 978-80-251-2940-1, Brno, 2010.
- [48] Frank Salim Peter Lubbers, Brian Albers. *HTML5*. Computer Press ISBN 978-80-251-3539-6, Brno, 2011.
- [49] Flexibook 1:1 [online], cit. 28.10.2014. Informační zdroj dostupný z URL: [<http://www.fraus.cz/flexibook-11/>](http://www.fraus.cz/flexibook-11/).
- [50] JavaFX [online], cit. 5.2.2014. Informační zdroj dostupný z URL: [<http://www.oracle.com/technetwork/java/javafx/>](http://www.oracle.com/technetwork/java/javafx/).
- [51] JavaFX Scene Builder [online], cit. 5.2.2014. Informační zdroj dostupný z URL: [<http://www.oracle.com/technetwork/java/javafx/tools/index.html>](http://www.oracle.com/technetwork/java/javafx/tools/index.html).
- [52] Apache Derby [online], cit. 5.2.2014. Informační zdroj dostupný z URL: [<http://db.apache.org/derby/>](http://db.apache.org/derby/).
- [53] Kerio - Přednáška o Apache Derby [online], cit. 5.2.2014. Informační zdroj dostupný z URL: [<http://www.java.cz/dwn/1003/27517\\\_czjug-apache-derby-slides.pdf>](http://www.java.cz/dwn/1003/27517\_czjug-apache-derby-slides.pdf).

## Seznam zkratek

**AJAX** Asynchronous JavaScript and XML

**API** Application Programming Interface

**APK** Android application package

**BLOB** Binary large object

**CSS** Cascading Style Sheets

**CSV** Comma-separated values

**ERP** Enterprise Resource Planning

**FUP** Fair User Policy

**GUI** Graphical user interface

**HTML** HyperText Markup Language

**HTTP** Hypertext Transfer Protocol

**IDE** Integrated Development Environment

**J2ME** Java Platform, Micro Edition

**JDK** Java Development Kit

**JKS** Java KeyStore

**JSON** JavaScript Object Notation

**LGPL** General Public License

**LMS** Learning Management System

**REST** Representational State Transfer

**SDK** Software development kit

**SOAP** Simple Object Access Protocol

**TDD** Test Driven Development

**URL** Uniform Resource Locator

**WSDL** Web Services Description Language

**WYSIWYG** What you see is what you get

**XML** Extensible Markup Language

# Přílohy

**Příloha 1:** Uživatelská a instalační příručka

Západočeská univerzita v Plzni  
Fakulta aplikovaných věd  
Katedra informatiky a výpočetní techniky

**E-learningový systém Teri**  
**Uživatelská a instalační příručka**



# Obsah

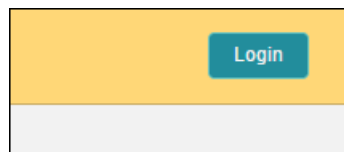
<b>1</b>	<b>Nápověda pro studenty</b>	<b>2</b>
1.1	Registrace a přihlášení . . . . .	2
1.2	Ověřování znalostí . . . . .	3
1.3	Výukový obsah . . . . .	5
1.4	Ostré testy . . . . .	6
1.5	Aplikace pro operační systém Android . . . . .	6
<b>2</b>	<b>Nápověda pro učitele</b>	<b>9</b>
2.1	Přidání předmětu . . . . .	9
2.2	Přidání okruhu předmětu . . . . .	9
2.3	Přidání vzkazu studentům . . . . .	9
2.4	Skrytí předmětu . . . . .	9
2.5	Správa otázek . . . . .	10
2.6	Správa ostrých testů . . . . .	10
2.7	Správa výukového obsahu . . . . .	13
2.8	Export . . . . .	14
2.9	Import . . . . .	15
<b>3</b>	<b>Instalace</b>	<b>16</b>

# 1 Náповěda pro studenty

Na adrese <http://students.kiv.zcu.cz/teri/> se nachází webová verze portálu Teri. Zde je popsána uživatelská příručka pro studenty.

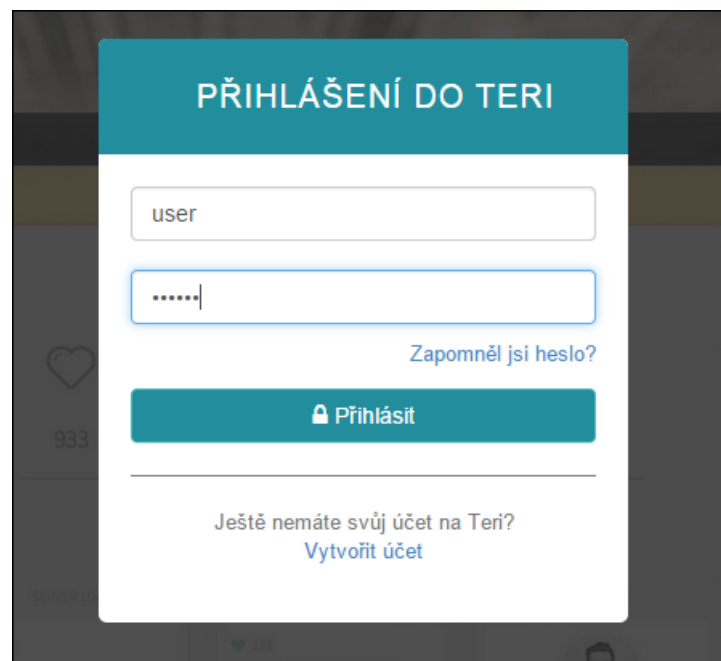
## 1.1 Registrace a přihlášení

Při vstupu na web systému Teri máte v pravém horním rohu tlačítko pro přihlášení viz obr. 1.1.



Obrázek 1.1: Tlačítko pro přihlášení.

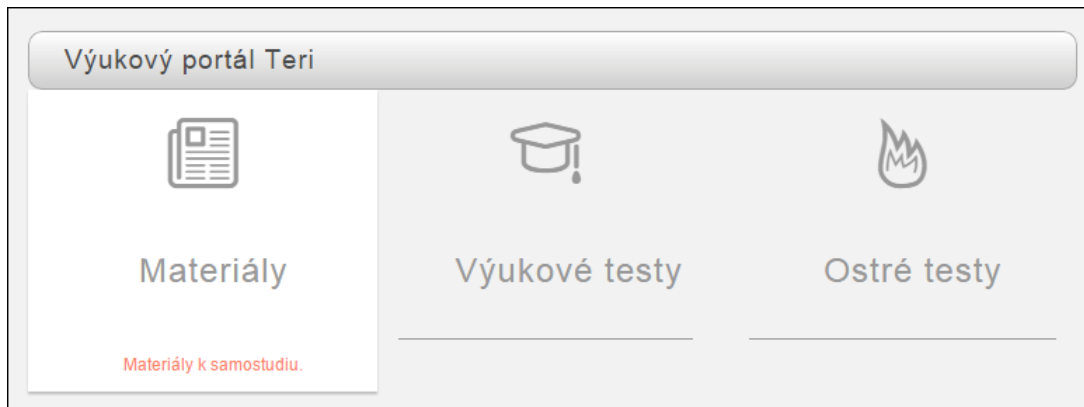
Po kliknutí na toto tlačítko se otevře přihlašovací obrazovka do systému viz obr. 1.2. Zde zadejte své přihlašovací údaje. Pokud jste na webu poprvé, tak klikněte na odkaz **Vytvořit účet**. Ten vás přesměruje na registrační formulář. V případě, že jste již v aplikaci registrován, ale ztratil jste své heslo, tak můžete využít funkcionalitu obnovy hesla. Stačí kliknout na odkaz **”Zapomněl jsi heslo?”**.



Obrázek 1.2: Přihlášení uživatele.

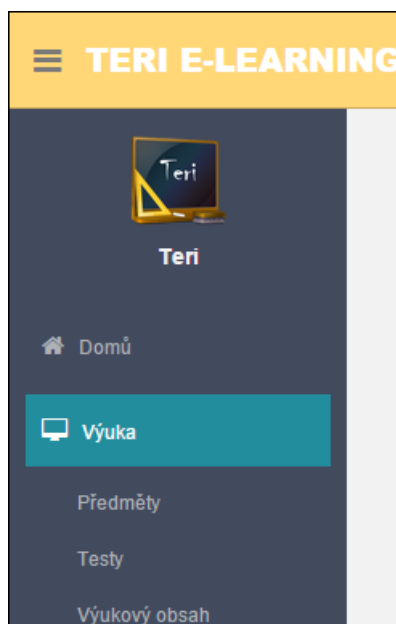
## 1.2 Ověřování znalostí

Pokud chcete ověřit své znalosti v daném předmětu, tak využijte funkci výukových testů. Na úvodní stránce se nachází rychlé odkazy viz obr. 1.3. Zde stačí kliknout na odkaz **Výukové testy**.



Obrázek 1.3: Odkazy na úvodní obrazovce.

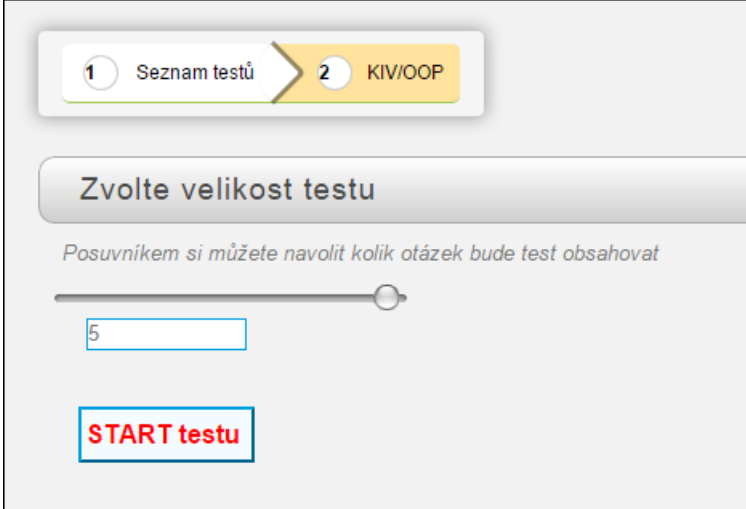
Pokud se zrovna nacházíte v jiné části webu, než na úvodní stránce, tak pro výukové testy můžete také použít kontextové menu (**Výuka**>**Testy**) viz obr. 1.4.



Obrázek 1.4: Hlavní kontextové menu.

V obou případech se dostanete na stránku s výběrem předmětu a daného výukového testu. Po kliknutí na název testu se zobrazí stránka s posuvníkem. Zde si zvolte

počet otázek testu viz obr. 1.5. V případě, že vyučující předmětu pošle studentům vzkaz, tak se zobrazí na této stránce.



The image shows a user interface for selecting the number of questions in a test. At the top, there are two tabs: '1 Seznam testů' and '2 KIV/OOP'. Below the tabs is a section titled 'Zvolte velikost testu'. Underneath this title is a slider control with a handle and a text input field containing the number '5'. Below the slider is a red button labeled 'START testu'.

Obrázek 1.5: Volba velikosti testu.

Nyní se vám spustí výukový test. Můžete zlepšovat své znalosti odpovídáním na zobrazené otázky. Pro kontrolu odpovědi na otázku klikněte na tlačítko **hotovo**. Pokud chcete otázku přeskočit, tak volte tlačítko **další**, nebo v případě návratu na předešlou otázku tlačítko **zpět**.

Během testu se vám může zobrazit pět druhů otázek. Otázka s přesnou odpovědí. Zde musíte do textového pole zadat požadovaný výraz. Dále pak otázka s jednou, nebo více správnými odpověďmi, otázka typu ano/ne a posledním druhem otázky je seřazovací otázka. U seřazovací otázky odpovědi seřad'te pomocí kurzoru myši metodou *táhni a pusť*.

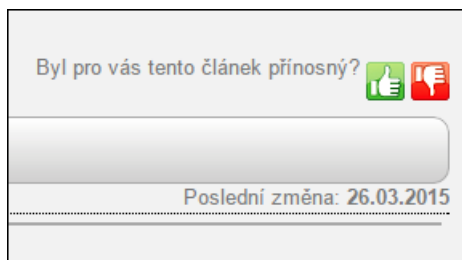
The screenshot shows a quiz interface with a red header bar containing the text "Špatná odpověď" (Wrong answer). Below the header, the question is displayed: "Znění otázky" (Question text) followed by "Seřadte datové typy jazyka Java dle jejich velikosti" (Sort the data types of the Java language by their size). The answer options are listed in a list: "byte", "short", "long", and "int". The "byte" and "short" options are highlighted in green, while "long" and "int" are highlighted in pink. At the bottom of the interface, there are three buttons: "<< zpět" (Previous), "| hotovo |" (Finished), and "další >>" (Next).

Obrázek 1.6: Seřazovací otázka.

### 1.3 Výukový obsah

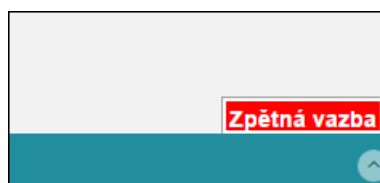
Pro zobrazení článků s výukovým obsahem využijte odkaz **Materiály** na úvodní stránce viz obr. 1.3, nebo odkaz **Výuka>Výukový obsah** v kontextovém menu viz obr. 1.4. Po výběru předmětu se zobrazí výpis všech aktuálních článků předmětu. Klikněte na název článku pro jeho zobrazení. Zobrazí se stránka s daným článkem. V dolní části obrazovky pod článkem se pak nachází odkazy na stáhnutelné přílohy. Dále se zde nachází odkaz na export článku do PDF (pokud vyučující tuto funkci nevyplne).

V pravém horním rohu se nachází panel pro anonymní hodnocení článku viz obr. 1.7. Zde můžete přidělit jednorázově kladné, či záporné hodnocení článku.



Obrázek 1.7: Hodnocení článku.

V případě, že jste přihlášení, tak můžete autorovi článku také poslat zpětnou vazbu. Formulář pro odeslání zpětné vazby se nachází v dolním rohu viz obr. 1.8. Po kliknutí na tlačítko se rozbalí formulář. Odeslaná zpráva je anonymní.



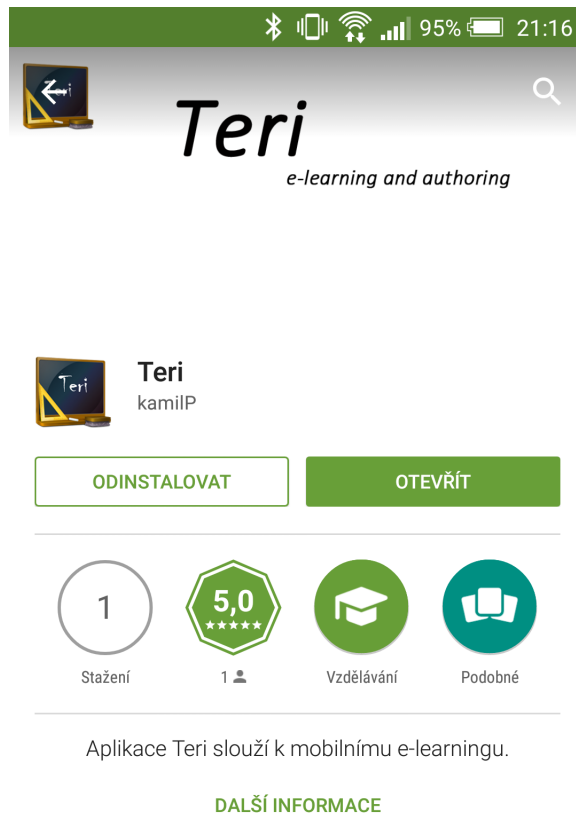
Obrázek 1.8: Zpětná vazba na článek.

## 1.4 Ostré testy

Pro zobrazení sekce ostrých testů použijte odkaz **Ostré testy**. Ten se nachází na úvodní obrazovce i v hlavním kontextovém menu. Abyste mohli spouštět ostré testy, tak musíte být přihlášení. Zobrazí se vám výpis aktuálně spuštěných testů. Po kliknutí na název se test spustí. Pozor, spuštění testu je závazné a nelze si jeden ostrý test spustit vícekrát. Během vyplňování testu se v horní části zobrazují hodiny, které ukazují zbývající čas do odevzdání. Test po vypršení limitu nelze odevzdat. Po odevzdání testu dostane vyučující předmětu výsledky. Ty poté sám zveřejní.

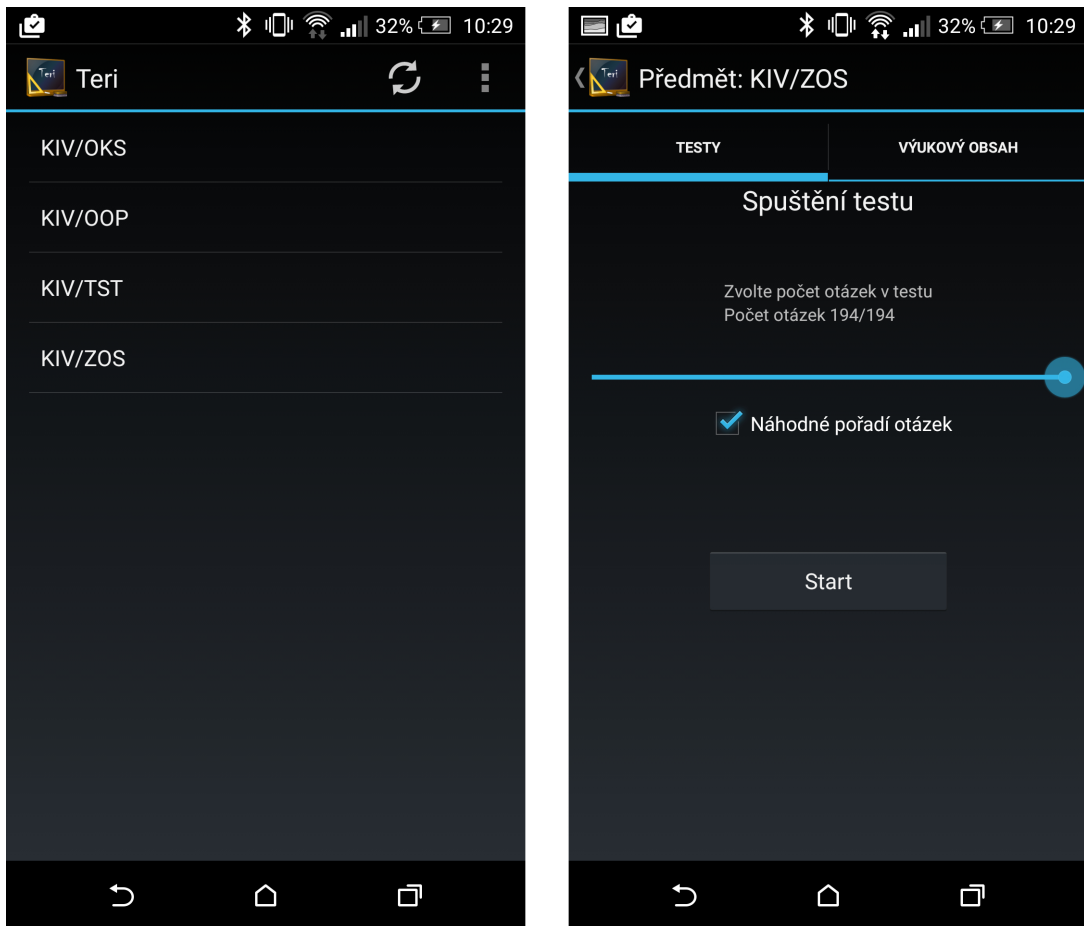
## 1.5 Aplikace pro operační systém Android

Pokud vlastníte mobilní zařízení s operačním systémem Android 4.0 a vyšší, tak můžete využít mobilní aplikaci Teri. Stačí v obchodě **Googla Play** zadat název **Teri** a aplikaci standardní cestou nainstalovat. (odkaz: <https://play.google.com/store/apps/details?id=cz.praum.teri.app>).



Obrázek 1.9: Aplikace Teri v obchodu Google Play.

Po spuštění se provede načtení výukových dat ze serveru. Po této synchronizaci se vám zobrazí seznam všech dostupných předmětů. Po kliknutí na název předmětu si můžete spustit výukové testy viz obr. 1.10, nebo prohlížet články s obsahem.



Obrázek 1.10: Android aplikace na mobilním telefonu.



## 2 Nápověda pro učitele

Učitelé mohou využívat stejných funkcionalit jako studenti. Navíc mají také přístup do administrace, kde mohou spravovat obsah. Zde je popis funkcí a ovládání administrační části přístupné pouze vyučujícím.

### 2.1 Přidání předmětu

Položka se nachází v sekci **Předměty**. Zde se na jednoduché stránce nachází formulář pro tvorbu vašeho předmětu. Vyplňte název předmětu. Je zde také druhé nepovinné textové pole pro zadání detailnějších informací.

### 2.2 Přidání okruhu předmětu

Položka se nachází v sekci **Předměty**. Tuto volbu budete potřebovat, pokud chcete svůj předmět rozdělit do více částí. Například na otázky k zápočtu a otázky ke zkoušce. V horní části formuláře si vyberete váš předmět a napíšete název nového testu. V případě, že nechcete dělit otázky k předmětu, tak není třeba tuto funkci používat.

### 2.3 Přidání vzkazu studentům

Položka se nachází v sekci **Předměty**. Z tabulky si vyberte předmět, ke kterému chcete vzkaz přiřadit. Dále vyplňte znění vzkazu a s využitím kalendáře určete, do kdy bude vzkaz platný. Tento vzkaz se bude studentům zobrazovat pokaždé, když zahájí výuku vašeho předmětu. Je to vhodná forma, jak studentům sdělit nějakou důležitou informaci, kterou by neměli přehlédnout.

### 2.4 Skrytí předmětu

Položka se nachází v sekci **Předměty**. Na stránce se vám vypíší všechny vaše předměty, pomocí tlačítka **Skrýt/Zapnout** můžete ovlivnit, zdali se předmět studentům zobrazí. Skrytý předmět však vy jako vyučující máte stále dostupný v administraci. Lze tak například předmět plnit obsahem a zveřejnit ho až je vše připravené.



Obrázek 2.11: Skrytí předmětu.

## 2.5 Správa otázek

Položky pro správu otázek se nachází v sekci **Otázky**. První položkou je **Přidat otázky**. Tuto položku vyberete v případě, že chcete přidávat nové testovací otázky do systému. V prvním kroku nejprve vyberete typ otázky a předmět/test, pro který otázku přidáváte.

V dalším kroku vyplňte potřebné údaje k dané otázce. Tedy znění otázky a odpovědi. Máte také možnost k otázce připojit obrázek či audio nahrávku ve formátu MP3. Můžete dále nastavit obtížnost otázky. Ta je brána v potaz při generování testů. Poslední funkcionalitou je nápověda. Tu doporučuji vyplnit v případě obtížných otázek. Rozumně napsaná nápověda často studentům pomáhá k rychlému pochopení otázky při učení.

Po úspěšném uložení otázky můžete pokračovat ve tvorbě dalších otázek nebo tuto akci ukončit.

Další položkou je **Editovat otázky**. V tabulce se zobrazují všechny vaše otázky. Přes tlačítko **editovat** se dostanete k formuláři, který vám dovolí otázku upravovat. V tabulce jsou pouze znění otázek. Pokud chcete zobrazit celou otázku včetně odpovědí, tak stačí kliknout na ono znění.

Poslední položkou je pak **Mazání otázek**. Obdobně jako u editace otázek se zobrazí tabulka s otázkami. Kliknutím na tlačítko **smazat** otázku ze systému smažete.

*Pozor, jedná se o nevratný proces.*

## 2.6 Správa ostrých testů

Položky pro správu otázek se nachází v sekci **Ostré testy**. První položkou je **Vytvořit ostrý test**. Jako učitel máte možnost v systému vytvářet i tzv. *ostré testy*. Tuto volbu můžete zvolit třeba pro zápočtové nebo zkouškové testy. Nejprve si vyberte předmět, ke kterému test bude patřit. Dále pak máte možnost využít automatického generátoru testu nebo manuální výběr otázek do testu. Při využití první možnosti

pouze zaškrtnete, jaké typy otázek chcete, aby bral generátor v potaz a zdali má generátor zahrnovat i skryté otázky viz obr. 2.12. Skryté otázky jsou otázky, které nejsou veřejně přístupné studentům ve výukových testech. V dalším kroku pak zvolíte počet otázek, které bude test obsahovat.

Obrázek 2.12: Tvorba ostrého testu - skryté otázky.

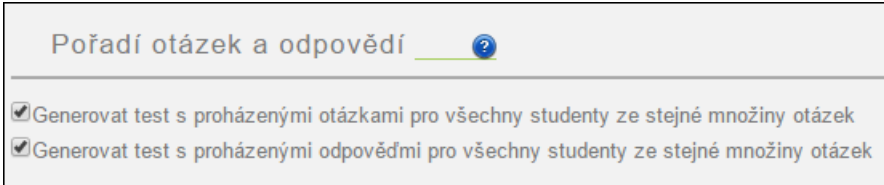
Systém pak za vás náhodně vybere otázky. V posledním kroku už pouze určíte platnost testu zadáním začátku a konce, viz obr. 2.13. Je vhodné volit dobu testu například na dobu cvičení, abyste mohli zajistit fyzický dohled nad vyplňováním a tím zabránit podvádění. Během vyplňování otázek systém studenta nepustí k standardnímu učení otázek. Poslední informace, kterou musíte zadat, je název testu.

Obrázek 2.13: Formulář při tvorbě ostrého testu.

Volitelně můžete také zadat časové omezení na vyplnění testu studentem. V takovém případě se vámi zadaná hodnota v minutách nastaví pro odpočet zbývajících času na odevzdání. Tento odpočet je zahájen spuštěním daného testu. Pokud tuto volbu

necháte na hodnotě 0, tak se po spuštění provede výpočet zbývajících času vzhledem k meznímu datumu platnosti testu.

Před uložením ostrého testu máte ještě možnost zvolit, zdali se vygenerovaný test má studentům zobrazovat s proházenými odpověďmi, proházenými otázkami, nebo kombinací obojího viz obr. 2.14. Ve výchozím stavu jsou zvolené obě možnosti. Zamezí se tak přímému podvádění, pokud v učebně sedí studenti vedle sebe a vidí si do monitorů.



Obrázek 2.14: Volba náhodného pořadí otázek a odpovědí.

Druhá možnost vytvoření ostrého testu je manuální výběr otázek. Zde si ručně zaškrtnete konkrétní otázky, které chcete, aby test obsahoval.

Ostatní je stejné jako u předchozí možnosti.

Další položkou je **Smazat ostrý test**. Zde můžete mazat vaše ostré testy. Jsou zde i testy, které již byly ukončeny.

*Mějte na vědomí, že smazáním předmětu smažete i jeho výsledky.*

Dále následuje položka **Výsledky ostrých testů**. Zde vyberte daný test. Zobrazí se vám tabulka, ve které jsou zaznamenány výsledky všech studentů, kteří ho absolvovali. Na každé řádce vidíte identifikační informace, počet správných a špatných odpovědí, procentuální úspěšnost a poslední položkou jsou ikonka papíru s tužkou a ikonka PDF dokumentu. Pokud na ikonku s papírem a tužkou kliknete, tak se zobrazí záznam toho, co konkrétně student v testu vyplnil. Jedná se o záznam, který se může hodit například při reklamaci výsledků. Je možné, že například u otázek s přesnou odpovědí student odpověděl správně, ale ne dostatečně přesně, jak by si systém představoval. V případě druhé ikonky se vám stejný záznam vygeneruje do PDF dokumentu. Tato funkce je přidána z důvodu možnosti archivace výsledků mimo systém.

Máte zde i možnost exportovat celou tabulku s výsledky do PDF dokumentu. Tato možnost je dobrá pro archivaci či zveřejnění výsledků testu.

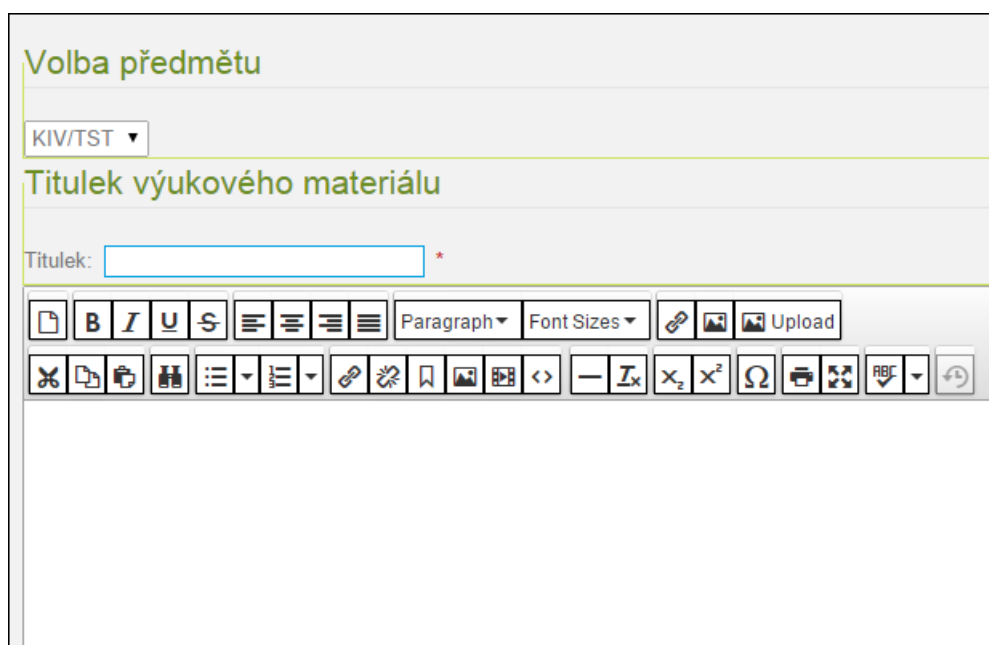
Další položkou je **Změna platnosti ostrých testů**. Zde můžete editovat již vytvořené ostré testy a měnit jim jejich platnost.

Předposlední položkou je **Zobrazit ostrý test**. Zde si vyučující může zobrazit náhled, jak vytvořený ostrý test vypadá a jak se bude zobrazovat studentům.

Poslední položkou je **Export otázek do PDF**. Na této stránce můžete vytvořit tisknutelnou verzi testu. Tvorba samotného testu je obdobná jako při tvorbě ostrého testu. Po dokončení tvorby dostanete PDF dokument s otázkami testu.

## 2.7 Správa výukového obsahu

V této sekci máte možnost tvorby, editace, mazání a náhledu výukových článků. V případě tvorby nového výukového článku se vám zobrazí formulář s možností pokročilé editace textu viz obr. 2.15.



The screenshot shows a web form titled "Volba předmětu" (Subject Selection). At the top, there is a dropdown menu with "KIV/TST" selected. Below this is a section titled "Titulek výukového materiálu" (Title of educational material) with a text input field labeled "Titulek:" followed by a red asterisk. Underneath the input field is a rich text editor toolbar containing various icons for text formatting (bold, italic, underline, strikethrough, bulleted list, numbered list), alignment (left, center, right), paragraph styles, font sizes, link management, image upload, and other editing functions.

Obrázek 2.15: Tvorba výukového obsahu.

Při tvorbě článku máte také možnost volby pokročilých nastavení viz obr. 2.16. Můžete zde omezit, kdy bude článek studentům viditelný, umožnit studentům export daného článku do PDF a možnost zpětné vazby na článek přihlášeným studentům.

Vydat tento výukový obsah: [?](#)  
 Zobrazit článek pouze přihlášeným uživatelům  
 Omezení zobrazení článku:  
 od:    
 do:    
 Umožnit uživatelům stáhnout tento článek v PDF  
 Zpětná vazba [?](#)  
**Skrýt pokročilá nastavení**

Obrázek 2.16: Pokročilá nastavení výukového obsahu.

Poslední možností při tvorbě článku je možnost nahrávání příloh. K nahrávání příloh můžete využít dynamického formuláře viz obr. 2.17. Symboly plus a mínus si můžete přidávat a odebírat počet nahrávaných příloh. Podporované formáty nahrávaných příloh jsou zobrazeny na formuláři.

**Přílohy**  
 Podporované formáty: rar, zip, txt, log, latex, jpg, png, gif, jpeg, pdf, xls, xlsx, doc, docx, odt, pptx, ppt  
 Příloha 1:  
 proguard-rules.txt  
 Příloha 2:  
 app-release.apk  
 Příloha 3:  
 Soubor nevybrán

Obrázek 2.17: Formulář příloh.

## 2.8 Export

Systém umožňuje export otázek výukových testů a výukových článků do XML dokumentu. Lze exportovat pouze vámi vytvořený obsah.

## 2.9 Import

Systém kromě exportu otázek podporuje také import otázek z XML souboru. Přes formulář nahrajte XML soubor, který musí mít přesnou strukturu XML souboru, která vzniká při využití funkce exportu. Nahrávaná XML musí být validní. Přímo na formuláři naleznete odkaz na XSD soubor, vůči kterému můžete validitu zkontrolovat.

Další možností importu je hromadný import studentů. Tuto možnost využije učitel například, pokud chce systém využít pro hromadné ostré testování. Pro spuštění ostrého testu musí být student zaregistrovaný a přihlášen. Ruční registrace všech studentů je v případě hromadného spuštění testů zdlouhavá. Proto může vyučující využít hromadného importu a studentům jejich účty založit. Pro import se využívá **CSV** dokumentu, který používá jako oddělovací znak středník. Na každém řádku importního souboru je definice jednoho importovaného studenta. Soubor musí být kódován v **UTF-8**.

Formát dat musí být ve tvaru: *jméno studenta;login;heslo;e-mail;osobní číslo*. Z toho e-mail a osobní číslo jsou nepovinné údaje viz obr. 2.18.

*povinné* *nepovinné*

<i>Jan Novák;novak.jan;heslo9087</i>	<i>novakj@email.cz;A13N0097P</i>
--------------------------------------	----------------------------------

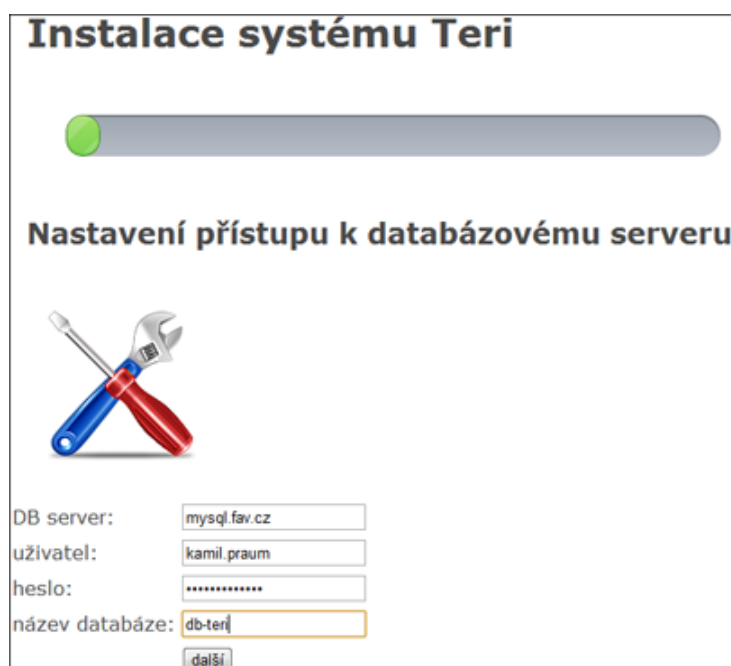
Obrázek 2.18: Formát importního CSV souboru pro import studentů.

### 3 Instalace

Rozbalte ZIP archiv obsahující soubory systému Teri do libovolného adresáře. Obsah tohoto adresáře nahrajte na váš server<sup>1</sup>. Je potřeba HTTP server s podporou jazyka PHP verze 5 a vyšší a databázový server MySQL. Nyní spustíte instalaci zadáním adresy ve tvaru

`doména_webu/teri/install.php` <sup>2</sup>.

Spustí se jednoduchý průvodce instalací systému Teri. V prvním kroku musíte vyplnit údaje k databázovému serveru. Databázi musíte mít před samotnou instalací vytvořenou. Do formuláře musíte vyplnit adresu databázového serveru, uživatelské jméno, heslo a název databáze, viz obr. 3.19.



The screenshot shows a web-based installation wizard titled "Instalace systému Teri". At the top, there is a progress bar with a green circle on the left. Below the title, the subtitle reads "Nastavení přístupu k databázovému serveru". There is an icon of crossed wrench and screwdriver. The form contains the following fields:

DB server:	<input type="text" value="mysql.fav.cz"/>
uživatel:	<input type="text" value="kamil.praum"/>
heslo:	<input type="password" value="*****"/>
název databáze:	<input type="text" value="db-teri"/>
	<input type="button" value="další"/>

Obrázek 3.19: První krok při instalaci systému Teri.

Dle vašich zadaných údajů se provede vygenerování souboru pro spojení s databází. Pokud tato akce proběhne v pořádku, tak se v dalším kroku provádí kontrola spojení s databázovým serverem. Při úspěšném spojení následuje předposlední krok instalace, při kterém se vytvářejí všechny potřebné databázové tabulky. Posledním krokem je vyplnění formuláře s údaji k administrátorskému účtu systému, viz obr. 3.20.

Nyní je systém úspěšně nainstalován a připraven k použití.


*Nezapomeňte z vašeho serveru vymazat soubor `install.php`.*

<sup>1</sup>Nejčastěji se pro tuto činnost používá protokol FTP.


<sup>2</sup>Za `doména_webu` dosad'te adresu, na které web poběží, například `http://www.zcu.cz/teri/install.php`



### Instalace systému Teri



#### Vytvoření administrátorského účtu



nick:

heslo:

jméno:

e-mail:

Obrázek 3.20: Tvorba účtu administrátora.