

Mendelova univerzita v Brně
Provozně ekonomická fakulta

E-learningová aplikácia pre podporu výuky zemepisu pre základnú školu v Nitrianskych Sučanoch

Bakalárska práca

Vedúci práce:
Ing. David Schubert

Elena Fodorová

Brno 2017

Na začiatok by som rada poďakovala vedúcemu mojej bakalárskej práce Ing. Davidovi Schubertovi za vedenie, rady a pripomienky pri vypracovávaní práce a za jeho trpezlivosť.

Čestné prehlásenie

Prehlasujem, že som túto prácu: **E-learningová aplikácia pre podporu výuky zemepisu pre základnú školu v Nitrianskych Sučanoch**

vypracovala samostatne a všetky použité pramene a informácie sú uvedené v zozname použitej literatúry. Súhlasím, aby moja práca bola zverejnená v súlade s § 47b zákona č. 111/1998 Sb., o vysokých školách v znení neskorších predpisov, a v súlade s platnou *Smernicou o zverejňovaní vysokoškolských záverečných prác*

Som si vedomá, že sa na moju prácu vzťahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brne má právo na uzavretie licenčnej zmluvy a použitie tejto práce ako školského diela podľa § 60 odst. 1 Autorského zákona.

Ďalej sa zaväzujem, že pred spísaním licenčnej zmluvy o využití diela inou osobou (subjektom) si vyžiadam písomné stanovisko univerzity o tom, že predmetná licenčná zmluva nie je v rozpore s oprávnenými záujmami univerzity, a zaväzujem sa uhradiť prípadný príspevok na úhradu nákladov spojených so vznikom diela, a to až do ich skutočnej výšky.

Brno, 22.5.2017

.....

Abstract

FODOROVÁ, E. E-learning application to support geography teaching for elementary school in Nitrianske Sučany. Bachelor thesis. Brno, 2017

The bachelor thesis deals with design and creation of an e-learning web application that will be focused on geography and will be helpful in teaching. The first part contains an introduction to the issue of thesis and clarification of certain terms. It also contains analysis of existing solutions. The following section defines the application requirements and design of database represented by the entity-relational model. The thesis continues with the implementation of the application, which was developed by PHP framework Slim and JavaScript framework AngularJS. At the end we can read about the results of the thesis and fulfillment of the assignment as well as about the possible recommendations and suggestions for the next extension of application.

Keywords

web application, e-learning, framework Slim, PHP, AngularJS, template engine Twig

Abstrakt

FODOROVÁ, E. E-learningová aplikácia pre podporu výuky zemepisu pre základnú školu v Nitrianskych Sučanoch. Bakalárska práca. Brno 2017

Bakalárska práca sa zaoberá návrhom a vytvorením e-learningovej webovej aplikácie, ktorá sa bude zaoberať zemepisom a bude nápomocná pri výučbe. Prvá časť obsahuje uvedenie do danej problematiky a vyjasnenie určitých pojmov. Ďalej sa tu nachádza analýza už existujúcich dostupných riešení. V nasledujúcej časti sú definované požiadavky na aplikáciu a návrh databáze zobrazený pomocou entitno-relačného modelu. Práca pokračuje samotnou implementáciou aplikácie na ktorej vývoj bol použitý PHP framework Slim a JavaScript framework AngularJS. Na konci sa môžeme dočítať o výsledkoch práce a splnenia zadania a taktiež o prípadných doporučeníach a návrhoch pri rozširovaní aplikácie.

Kľúčové slová

webová aplikácia, e-learning, framework Slim, PHP, AngularJS, šablónovací systém Twig

Obsah

1	Úvod a cieľ práce	10
1.1	Úvod	10
1.2	Cieľ práce	10
2	Metodika	11
3	Súčasný stav skúmanej problematiky	12
3.1	Webové aplikácie	12
3.2	E-learning	13
3.3	Analýza existujúcich webových aplikácií	13
3.3.1	Seterra	14
3.3.2	Toporopa	14
3.3.3	Slepé Mapy	14
3.3.4	Zhodnotenie analýzy	15
3.4	Prieskum technológií	15
3.5	Vyhodnotenie prieskumu	17
4	Vybrané technológie	18
4.1	Slim	18
4.2	Twig	19
4.3	JavaScript	19
4.4	AngularJS	20
4.5	HTML, CSS	21
4.6	Bootstrap	22
4.7	PHP	22
5	Návrh a implementácia	23
5.1	Požiadavky na aplikáciu	23
5.1.1	Funkčné požiadavky	23
5.1.2	Nefunkčné požiadavky	24
5.2	Návrh databáze	24
5.3	Implementácia aplikácie	25
5.3.1	Štruktúra aplikácie	25
5.3.2	Nastavenie šablónovacieho systému	26
5.3.3	Prístup do aplikácie	27
5.3.4	Vytváranie testu	29
5.3.5	Učebné materiály	30
5.3.6	Slepé mapy	31
6	Testovanie aplikácie	33
6.1	Testovanie používateľom	33
6.2	Výsledky a zhodnotenie testovania	33

7	Diskusia	34
8	Záver	35
9	Zoznam literatúry	36
	Prílohy	38
A	Grafický design úvodnej stránky aplikácie	39
B	Ukážka písania testu	40
C	Ukážka administračného rozhrania	41
D	Návrh databáze	43
E	Priložené CD	44

Zoznam obrázkov

Obrázok 1: Štatistika používania jazykov na strane servera,(MILLARES, 2015)	15
Obrázok 2: Middleware,(slimframework.com, 2017a)	18
Obrázok 3: Architektúra MVC, (codeproject.com, 2014)	20
Obrázok 4: Zloženie CSS príkazu, (w3schools.com, 2017)	21
Obrázok 5: EER Diagram	24
Obrázok 6: Štruktúra aplikácie	25
Obrázok 7: Prihlasovací formulár	27
Obrázok 8: Úkažka vygenerovaného hesla	29
Obrázok 9: Vkladanie otázky s možnosťami	29
Obrázok 10: Vkladanie otázky s možnosťami	30
11 Slepá mapa	32
Obrázok 12: Grafický design úvodnej stránky	39
Obrázok 13: Písanie testu	40
Obrázok 14: Prehľad študentov v administračnom rozhraní	41
Obrázok 15: Formulár na upload učebného materiálu	42
Obrázok 16: EER diagram	43

1 Úvod a cieľ práce

1.1 Úvod

Technologický pokrok sa odráža vo všetkých odvetviach a ovplyvňuje náš každodenný život. Tak ako rôzne oblasti zavádzajú využívanie informačných technológií aj školstvo sa snaží meniť a modernizovať svoje praktiky vo vzdelávaní a celkovo vzdelávací proces. Preto sa pri výuke čoraz viac využívajú moderné informačné technológie, ktorými sa snažia učitelia žiakov zaujať a motivovať na toľko aby sa toľko štúdiu venovali aj mimo školského prostredia. Druh takejto výuky sa nazýva e-learning.

E-learning umožňuje neobmedzený prístup k výučbovým materiálom a tak sa žiaci a študenti vedia k nim dostať aj zo svojich osobných počítačov. Žiaci si tak tiež môžu overiť svoje nadobudnuté schopnosti pomocou testov prispôsobených ich učebným osnovám.

1.2 Cieľ práce

Cieľom tejto práce bude vytvorenie interaktívnej e-learningovej aplikácie pre podporu výučby zemepisu a geografie na základných školách. Aplikácia by mala pomôcť motivovať žiakov k vzdelávaniu a umožniť vyučujúcim učiť jednoduchou, zaujímavou a hravou formou výuky.

Aplikácia bude obsahovať administratívne rozhranie pre jednoduchšie spravovanie. Opora bude umožňovať žiakom overiť si schopnosti pomocou rôznych typov úloh a hier, ako napríklad slepá mapa, testovanie danej problematiky pomocou kvízu. Učebné materiály ako je prehľad jednotlivých krajín s popisom daného štátu a učebných textov prichystaných prostredníctvom vyučujúcich na študovanie budú tiež súčasťou aplikácie.

Aplikácia bude implementovaná pomocou jazyka PHP, HTML, CSS, JavaScript s frameworkom AngularJS a PHP frameworkom, ďalej pomocou Bootstrapu a pomocou systému riadenia báze dát MySQL.

2 Metodika

Po naštudovaní danej problematiky bude vykonaná analýza aplikácii zameraných na rovnakú oblasť, ktorou sa aplikácia bude zaoberať. Analýza sa bude sústrediť predovšetkým na skúmanie silných a slabých stránok daných aplikácii, čo im chýba a čo by bolo vhodné zahrnúť a doplniť do budúcej komplexnejšej aplikácie tak aby vyhovovala požiadavkám stanoveným zadávateľom. Ďalej po oboznámení sa s bežne používanými technologickými prostriedkami pre tvorbu webových aplikácií budú vybrané technológie vyhovujúce pre vývoj danej aplikácie.

Následne po zistení, stanovení a upresnení užívateľských požiadavkou, ktoré budú rozdelené na funkčné a nefunkčné, bude vykonaný návrh jednoduchej a prehľadnej aplikácie tak aby boli všetky požiadavky zohľadnené a splnené. Pre správne fungovanie aplikácie je potrebné vytvoriť adekvátny návrh databáze, ktorý bude zobrazený pomocou entitno-relačného diagramu.

Po navrhnutí aplikácie bude nasledovať samotná implementácia. Aplikácia bude vyvíjaná predovšetkým za pomoci jazyka PHP a JavaScript. Pre zjednodušenie vývoja bude použitý PHP framework, ktorý bude vybraný na základe prieskumu. Čo sa týka vizuálnej stránky budú použité jazyky HTML, CSS spolu s frameworkom Bootstrap, ktorý okrem iného pomáha automaticky prispôsobiť zobrazenie webovej stránky vzhľadom k tomu, na akom zariadení danú stránku prezeráme, teda poskytuje responzívny design. Pre uchovávanie dát bude aplikácia využívať systém riadenia báze dát MySQL.

Nakoniec po zhotovení aplikácie bude nasledovať testovanie aplikácie a zhodnotenie dosiahnutých výsledkov na základe stanoveného cieľa a splnenia stanovených požiadavkou a návrhov.

3 Súčasný stav skúmanej problematiky

V tejto kapitole sa pokúsím zachytiť a definovať skúmanú problematiku v súčasnosti. V kapitole sa nachádza taktiež analýza vybraných obdobných aplikácií a jej zhodnotenie a záver. Ďalej predstavím technológie, ktoré by som pre vývoj aplikácie chcela použiť.

3.1 Webové aplikácie

Dve dôležité súčasti modernej webovej stránky sú flexibilné webové prehliadače a webové aplikácie, oboje dostupné všetkým bez akýchkoľvek nákladov.

Dnešné webové stránky sú vzdialené od statických textových a grafických stránok zo začiatku a polovice deväťdesiatych rokov. Moderné webové stránky umožňujú používateľom podľa individuálnych preferencií a nastavení manipuláciu s dynamickým obsahom.

Web aplikácie umožňujú návštevníkom webových stránok odosielať a získavať dáta do / z databázy cez internet pomocou svojho webového prehliadača. Údaje sa následne zobrazia používateľovi za pomoci prehliadača vďaka tomu, že informácie sa generujú dynamicky webovou aplikáciou prostredníctvom webového servera.

V prípade technicky orientovanejších webových aplikácií sa aplikácia dotazuje na obsahový server a dynamicky vytvára webové dokumenty, ktoré slúžia klientovi. Dokumenty sa generujú v štandardnom formáte, ktorý umožňuje podporu všetkými prehliadačmi. JavaScript je jedna forma skriptu na strane klienta, ktorá dovoľuje dynamické prvky (napr. obrázok sa zmení, akonáhle používateľ naň nasmeruje myš).

Ďalšou významnou výhodou budovania a údržby webových aplikácií je, že vykonávajú svoju funkciu bez ohľadu na operačný systém a prehliadač na strane klienta. Webové aplikácie sú rýchlo nasadené kdekoľvek bezplatne a bez potreby inštalácie (acunetix.com, 2017).

Prvé webové aplikácie sa začali vytvárať krátko predtým, než si Web získal svoju popularitu. Napríklad, Larry Wall vytvoril Perl, populárny server-side skriptovací jazyk, v roku 1987. To bolo sedem rokov, než internet naozaj začal získavať popularitu mimo akademických a technologických kruhov. Prvé tradičné webové aplikácie sú relatívne jednoduché, ale na konci 90. rokov sa vývoj aplikácií postupne posúval smerom k zložitejším webovým aplikáciám.

Väčšina webových aplikácií je založených na architektúre klient-server, kde klient zadá informácie, zatiaľ čo server ukladá a načítava informácie. Nový impulz pre webové aplikácie je posunúť sa k takým aplikáciám, ktoré za normálnych okolností nepotrebujú server pre ukladanie informácií. Napríklad textový editor ukladá dokumenty na vašom počítači a nepotrebuje k tomu server. Webové aplikácie môže poskytovať rovnakú funkčnosť a dosiahli výhodu pracovať na rôznych platformách. Aj webová aplikácia môže pôsobiť ako textový editor, ukladajúci informácie a umožňujúci stiahnutie dokumentu na svoj súkromný pevný disk (Nations, 2016).

3.2 E-learning

E-learning definujeme ako inštrukcie prezentované pomocou digitálnych zariadení, ako je napríklad počítač alebo mobilné zariadenie, ktoré sú určené na podporu vzdelávania. Existuje viac foriem e-learningu s nasledujúcimi vlastnosťami:

- Ukladá a / alebo prenáša lekcie na CD-ROM, lokálne interné alebo externé pamäte alebo servery na internete alebo intranete.
- Zahŕňa obsah relevantný k cieľu učenia.
- Používa mediálne prvky, ako text a obraz na zobrazenie obsahu.
- Využíva výučbové metódy, ako príklady, cvičenia a spätné väzby na podporu štúdia.
- Môžu byť pod vedením inštruktora (synchronný e-learning), alebo určená pre individuálne štúdium (asynchronný e-learning).
- Pomáha študentom vytvárať nové znalosti a zručnosti spojené s cieľmi individuálneho učenia alebo na zlepšenie organizačného výkonu.

E-learningové kurzy zahŕňajú ako obsah (informácie) taktiež aj učebné metódy (postupy), ktoré pomáhajú ľuďom naučiť sa jeho obsah.

E-learningové kurzy sú sprostredkované prostredníctvom digitálnych zariadení, ako sú počítače a chytré telefóny a používajú slová v podobe hovoreného alebo tlačeného textu a obrázky ako sú ilustrácie, fotografie, animácie alebo video. Niektoré formy e-learningu nazývané asynchronný e-learning sú určený pre individuálne samoštúdium. Tieto kurzy sú obvykle založené na vlastnom tempe, čo umožňuje jednotlivým študentom prístup k výcviku kedykoľvek alebo z ľubovoľného miesta na vlastnú päsť. Iné formy zvané virtuálne učebne, webináre ¹, alebo synchronný e-learning, ktorý je určený pre učenie v reálnom čase vedené inštruktorom. Synchronný e-learning teda umožňuje študentom rôznych časti zeme pripojiť sa k online výučbe vedenej inštruktorom v reálnom čase. Avšak, synchronne spôsoby vyučovania sú tiež často zaznamenané, čo umožňuje jednotlivým študentom učenie vo vlastnom tempom. Synchronne a asynchronne formy e-learningu môžu podporovať spoluprácu s ostatnými nástrojmi ako sú YouTube, chat, diskusné fóra či e-mail (Clark, Mayer, 2011).

3.3 Analýza existujúcich webových aplikácií

Testovacích web aplikácií zameraných na zemepis môžeme nájsť na internete niekoľko, nie však všetky majú požadované vlastnosti a mnohé z nich sú zamerané len na nejakú alternatívu testovania. Mnoho aplikácií je zameraných len na testovanie ale neobsahujú materiály k naštudovaniu daného učiva a neumožňujú tvorenie vlast-

¹Webinár je seminár prednášaný prostredníctvom internetu.

ných testov prispôsobených k aktuálnej výuke. K analýze som vybrala tri webové aplikácie ktoré sú prevažne zamerané na testovanie pomocou máp.

3.3.1 Seterra

Seterra je webová aplikácia, ktorá je spustiteľná v rôznych webových prehliadačoch. Môžete ju taktiež stiahnuť do svojho mobilného telefónu či tabletu ako mobilnú aplikáciu. Dostupná je ako pre Android taktiež pre iOS. Táto webová aplikácia ponúka rôzne druhy slepých máp, kde môžete otestovať znalosti z polohy štátov, riek, hlavných miest, ostrovov a polostrovov, pohorí a iných oblastí. Sú to mapy sveta, jednotlivých kontinentov a niektorých vybraných štátov, u ktorých je možné precvičiť si mestá, vodné plochy či regióny a oblasti. Aplikácia tiež ponúka u jednotlivých máp mód učenia, pri ktorom na základe kliknutia na daný prvok zistíte o aký prvok sa jedná. Ďalšou možnosťou je preskúšať sa z vlajok pomocou priradzovania vlajok ku štátom. (online.seterra.com, 2017).

Do aplikácie nie je možné sa prihlásiť a ukáže vám percentuálnu úspešnosť a čas len z aktuálneho preskúšania. Tieto výsledky si nemôžete spätne prezerat'. Je zameraná len na testovanie pomocou máp, neobsahuje žiadne testy či náučné texty.

3.3.2 Toporopa

Toporopa bola vydaná holandskou spoločnosťou EastDock Media v Amsterdame. Obsah je zameraný na otázky spojené s orientáciou na mape a týkajú sa štátov, miest, vlajok, prístavov, ostrovov ale aj historických bitiek na území Európy. Všetky tieto mapy sú sústredené len na európsky kontinent (toporopa.eu, 2017).

Táto aplikácia neumožňuje prihlásenie a po splnení úloh vám poskytne počet chýb. Tieto výsledky sa zobrazujú taktiež iba pri aktuálnom preskúšaní a nemáte prístup k výsledkom z iných testov.

3.3.3 Slepé Mapy

Slepé mapy sú projektom Fakulty informatiky na Masarykovej univerzite, konkrétne v rámci skupiny Adaptive Learning. Prostredníctvom aplikácie si môžete otestovať znalosti zo zemepisu pomocou slepých máp celého sveta. Precvičovať môžete tak ako u predchádzajúcich webových aplikáciách rôzne oblasti zemepisu. Ponúka dva typy otázok a to odpovedanie na konkrétnu otázku alebo výber z niekoľkých možností a taktiež máte možnosť nedopovedať na otázku a preskočiť ju pomocou kliknutia na tlačidlo „Neviem“. V tejto aplikácii si svoje výsledky môžete zobrazit' pomocou vizualizácie vedomostí. Z tejto mapy viete vyzistiť, ktoré oblasti vám idú najlepšie, a v ktorých častiach máte medzery. Výborná vlastnosť je, že aplikácia sa učí z dát od všetkých užívateľov, to znamená že výber otázok je založený na týchto dátach (slepemapy.cz, 2017).

Ďalšou z možností je aj prihlásenie do aplikácie, kde je vaše skóre zaznamenávané a máte k nemu nepretržitý prístup. Aplikácia je navrhnutá tak aby zadávala

také otázky pri opakovaní testov, ktoré pomôžu v zlepšení oblasti, v ktorých ste pôvodne získali horšie výsledky.

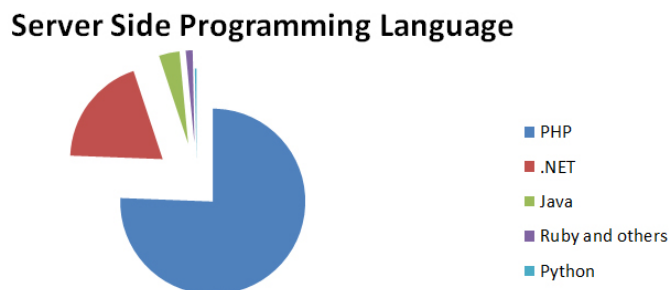
3.3.4 Zhodnotenie analýzy

Na základe analýzy je možné vidieť, že existuje viacero takýchto aplikácií, avšak nie sú celkom prispôsobené pre dlhodobé vzdelávanie. Vzhľadom k tomu, že aplikácia by mala obsahovať okrem iného nejaké učebné materiály, poprípade by sa prostredníctvom nej dali testovať znalosti nie len z oblasti lokalizácie krajín či iných geografických prvkov, nevyhovujú ako podpora pre kompletne vzdelávanie. Pre plnohodnotnú oporu, ktorá bude zabezpečovať aj vytváranie vlastných testov prostredníctvom vyučujúcich, či prístup k študijným materiálom, ktoré spĺňajú požiadavky a sú prispôbolené aktuálnym učebným osnovám, je nutné vytvoriť ucelenú aplikáciu so všetkými týmito vlastnosťami, o ktorú sa pokúsím v tejto práci.

3.4 Prieskum technológií

Pri tvorbe webových aplikácií existuje množstvo možností a kombinácií aké technológie používať, hlavne čo sa týka rôznych frameworkov alebo knižníc. V tejto kapitole sa pokúsím stručne načrtnúť niektoré z nich.

Aplikácia bude vyvíjaná predovšetkým v jazyku PHP. Podľa článku zverejneného na stránke Stone River eLearning, ktorého autorom je Gerard Millares, skriptovací jazyk PHP je používaný až v 75% prípadoch webových stránok ako jazyk na strane servera. Štatistiku server-side jazykov môžete vidieť na Obrázku 1.



Obrázok 1: Štatistika používania jazykov na strane servera, (MILLARES, 2015)

Pri používaní tohto jazyka je k dispozícii mnoho moderných frameworkov. Väčšina frameworkov funguje na princípe MVC architektúry - Model, View a Controller.

V krátkosti sa pokúsím načrtnúť niekoľko vybraných PHP frameworkov:

- Laravel - tento framework je relatívne mladý, ktorý bol vydaný v roku 2011. Napriek tomu sa v najnovších štatistikách magazínu SitePoint pohybuje na prvej priečke ako najčastejšie používaný PHP framework. Označuje sa za veľmi

prehľadný a jednoduchý na naučenie, tak isto vhodný pre začiatočníkov v oblasti PHP frameworkov. Laravel ma mnoho výhod, medzi ktoré parí aj:

- Jednoduchosť implementácie autentifikácie - takmer všetko je nakonfigurované mimo frameworku.
 - Integrácia s mail serverom - poskytuje ovládače pre SMTP², čo umožňuje odosielanie pošty prostredníctvom lokálnej alebo cloudovej služby. Okrem toho podporuje aj odosielanie iných upozornení vrátane napríklad SMS.
 - Automatizácia testovania - podporuje testovanie s PHPUnit, kde phphunit.xml súbor je predpripravený a nastavený pre aplikáciu.(belitsoft.com, 2016)
- Nette - framework Nette vytvoril v roku 2004 český programátor, podnikateľ a publicista David Grudl. Preto je framework hlavne rozšírený na území Slovenska a Českej republiky, kde patrí medzi najobľúbenejšie. I napriek tomu sa Nette dostalo na 3. priečku štatistiky vyššie spomínaného magazínu SitePoint, ktorá prebehla v roku 2015.

Výhody frameworku:

- Bezpečnosť pred zraniteľnosťami - framework zahrňuje technológiu, ktorá ochráni aplikáciu pred rizikom Cross-Site Scriptingu, prostredníctvom toho, že všetky výstupy ošetrí automaticky.
 - Podpora moderných technológií a koncepcii - Nette podporuje AJAX / AJAJ, Dependency Injection, SEO, DRY, KISS, MVC, Web 2.0, cool URL
 - Ladiaci nástroj - umožňuje jednoduchšie odhaľovať chyby, ktoré nastávajú pri vývoji danej aplikácie
 - Objektový návrh - čistý objektový návrh, ktorý využíva nové vlastnosti PHP 7.1 komponent a udalosti riadeného modelovania (nette.org, 2017)
- Slim Slim je PHP mikro framework a bol vytvorený ako inšpirácia frameworkom Sinatra. Mikro frameworky bývajú minimalistické a výborne vyhovujú menším aplikáciám, ktoré by klasický framework plne nevyužili.

Výhody mikro frameworku Slim:

- HTTP Smerovač - rýchle a prehľadné smerovania v aplikácii
- Middleware - slúži na manipuláciu s objektami Request a Respose. Pomáha chrániť aplikácie pred cross-side request a pri autentifikovaní pri spustení aplikácie.

²Simple Mail Transfer Protocol - internetový protokol určený pre prenos správ elektronickej pošty

- HTTP Caching - Slim Framework umožňuje uchovávať aktíva, ktoré sa nemenia často, ako sú favicon³, CSS súbory, čo zlepšuje celkový výkon a čas načítania webových stránok
- Vytváranie Rest aplikácií - Slim je tiež skvelý pre vytváranie REST API, to uľahčuje vývojárom umožniť tretím stranám prístup k aplikačným funkciám
- Dependency Injection - umožňuje vývojárovi mať plnú kontrolu nad externými nástrojmi a použiť akýkoľvek Container-Interop kontajner
- Podpora PSR-7 - Slim podporuje akúkoľvek implementáciu HTTP správ PSR-7. Umožňuje prezeráť a ovládať metódu správ HTTP, stav, URI, hlavičky, súbory cookie a telo (agriya.com, 2016)

Pri vývoji takýchto aplikácií sa rovnako používa HTML, CSS pre vizualizáciu. Pri tom môžeme tiež použitie zjednodušiť frameworkom pri CSS je to predovšetkým najčastejšie používaný framework Bootstrap.

Funkcionalita sa dá ďalej implementovať pomocou JavaScriptu, ktorý môžeme použiť spolu s JavaScriptovými frameworkami, ktorých je hneď niekoľko, no pre túto aplikáciu som sa rozhodla pracovať s frameworkom AngularJS.

3.5 Vyhodnotenie prieskumu

Každý framework ma určitú skupinu zastúpenčov a často je to len o subjektívnom názore, ktorý framework danej skupine vyhovuje, či ktorý je aktuálne vhodný pre vyvíjajúcu aplikáciu. Všetky vyššie uvedené frameworky majú plno výhod, no vzhľadom k tomu, že framework Nette z určitej časti ovládam a Bakalársku prácu považujem za výbornú možnosť naučiť sa niečo nové a rozšíriť svoje znalosti v tejto oblasti, som si zvolila framework Slim. K rozhodnutiu mi taktiež dopomohla skutočnosť, že aplikácia ktorú sa chystám vyvíjať nebude taká rozsiahla aby využila naplno všetky možnosti frameworku Laravel.

³je malá ikona, ktorá spojená s určitou web stránkou alebo webom

4 Vybrané technológie

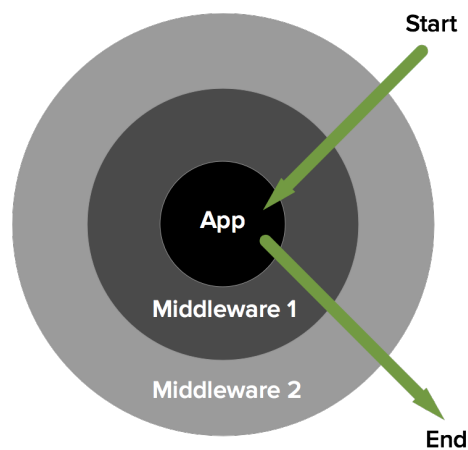
Vývoj technológií pre weby či webové aplikácie sa posúva stále dopredu. Pre vývoj takýchto aplikácií existuje niekoľko možností ako skombinovať rôzne jazyky s frameworkami či knižnicami, ktorých je k dispozícii v súčasnej dobe hromada. Niektoré boli uvedené v predchádzajúcej kapitole.

V nasledujúcej časti predstavím technológie, s ktorými som sa rozhodla pracovať pri vývoji aplikácie. Pokúsim sa stručne vysvetliť a popísať hlavné znaky a princípy jednotlivých technológií.

4.1 Slim

Slim je PHP micro framework, ktorý napomáha k rýchlemu písaniu jednoduchých a zároveň výkonných webových aplikácií a API, teda rozhraní pre programovanie aplikácií. Hlavnými znakmi a vymoženostami tohto frameworku sú:

- HTTP router (smerovač) – umožňuje vykonávať rýchle a výkonné presmerovania, ktoré mapujú spätné volania pre špecifické požiadavkové metódy a URI. Podporuje parametre a vyhľadávanie vzorcov.
- Middleware – pomocou middleware je možné vyladiť HTTP požiadavky a odozvy objektov okolo Slim aplikácie. Middleware je vhodné použiť napríklad v prípadoch, keď chcete aplikáciu ochrániť pred CSRF (Cross-Site Request Forgery) či keď chcete vykonať autentizáciu ešte pred spustením aplikácie.



Obrázok 2: Middleware, (slimframework.com, 2017a)

- Podpora PSR-7 – Slim podporuje všetky PSR-7 HTTP implementácie správ, to vedie k tomu, že je možné kontrolovať a manipulovať s metódami HTTP správ, statusmi, URI, hlavičkami, cookies a telom správy.
- Dependency Injection (Vkladanie závislostí) – umožňuje úplnú kontrolu nad externými nástrojmi. Slim používa kontajner (dependency container) a pod-

poruje ďalšie kontajnery, ktoré implementujú Container-Interrom rozhranie. Je možné použiť vstavovaný kontajner alebo kontajnery tretích strán ako sú Acclimate alebo PHP-DI (slimframework.com, 2017b).

4.2 Twig

Twig je moderný šablónovací systém pre PHP. Jeho 3 najdôležitejšie vlastností sú:

- Rýchlosť – Twig kompiluje šablóny až na jednoduchý optimalizovaný PHP kód. Režijný v porovnaní s bežným PHP kódom bol zredukovaný na úplné minimum.
- Bezpečnosť – Twig má sandbox⁴ mód na vyhodnotenie nedôveryhodného kódu šablóny. To umožňuje Twig-u aby bol použitý ako šablónovací jazyk pre aplikácie kde užívatelia môžu upravovať návrh šablón.
- Flexibilita – Twig je poháňaný flexibilným lexikálnym a syntaktickým analyzátorom. To zabezpečuje, že vývojár môže definovať jeho vlastné značky a filtre a vytvoriť vlastné DSL (twig.sensiolabs.org, 2017).

4.3 JavaScript

Keď sa JavaScript prvýkrát objavil v roku 1995, jeho hlavným cieľom bolo spracovať niektoré vstupné validácie, ktoré boli predtým ponechané na jazyky na strane servera ako napríklad Perl.

Od tej doby, JavaScript rozrástol do dôležitej vlastnosti každého väčšieho webového prehliadača na trhu. Už nie je požívaný len na jednoduché overenie dát, JavaScript teraz spolupracuje s takmer všetkými aspektami okna prehliadača a jeho obsahom. JavaScript je uznávaný ako plnohodnotný programovací jazyk, ktorý je schopný zložitých výpočtov a interakcií, vrátane anonymných (lambda) funkcií a dokonca metaprogramovania⁵. JavaScript sa stal tak dôležitou súčasťou webu, že aj alternatívne prehliadače, vrátane tých v mobilných telefónoch ho podporujú. Dokonca aj Microsoft, s vlastným skriptovacím jazykom na strane klienta s názvom VBScript, skončil zahrnutím vlastnej implementácie JavaScriptu v prehliadači Internet Explorer od jeho najstarších verzii.

Vzostup JavaScript-u z jednoduchého vstupného validátora na mocný programovací jazyk nebolo možné predvídať. JavaScript je zároveň veľmi jednoduchý a veľmi zložitý jazyk, ktorý sa v priebehu krátkej doby naučíte, ale trvá roky ho zvládnuť. (ZAKAS, 2012).

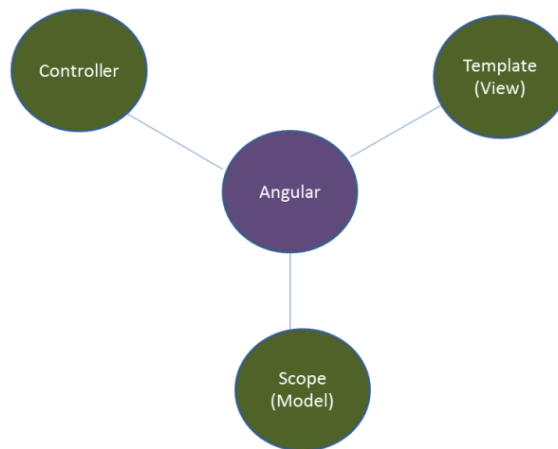
⁴Sandbox je izolované prostredie, v ktorom program alebo súbor môže byť spustený bez vplyvu na aplikáciu, v ktorej beží.

⁵Metaprogramovanie je vytváranie programov, ktoré manipulujú s programami ako s dátami.

4.4 AngularJS

AngularJS je open source JavaScript knižnica, ktorá je podporovaná a udržiavaná spoločnosťou Google. AngularJS sa ponára do niekoľko najlepších aspektov rozvoja na strane servera a využíva ich k zlepšeniu HTML v prehliadači, vytváraním základu, ktorý umožňuje vybudovať bohaté aplikácie jednoduchšie a ľahšie. AngularJS aplikácie sú postavené na princípe návrhového vzoru s názvom Model-View-Controller (MVC), ktorý kladie dôraz na vytváranie aplikácií, ktoré sú:

- Rozšíriteľné: Je ľahké prísť na to, ako komplexná AngularJS aplikácia funguje, akonáhle pochopíte základy. To znamená, že môžete ľahko rozšíriť aplikáciu pre vytváranie nových a užitočných funkcií pre užívateľov.
- Udržiavateľné: AngularJS aplikácie sa ľahko ladia a opravujú, to znamená, že dlhodobé udržiavanie aplikácie je zjednodušené.
- Testovateľné: AngularJS má dobrú podporu pre end-to-end⁶ testovanie, čo znamená, že si môžete ľahko nájsť a opraviť chyby pred tým ako ich nájde užívateľ.
- Štandardizované: AngularJS vychádza z vlastných schopností webového prehliadača, bez toho aby sa vám dostal do cesty, to umožňuje vytvárať štandardizované webové aplikácie, ktoré využívajú najnovšie funkcie (napríklad HTML5 API) a známe nástroje a frameworky (Freeman, 2014).



Obrázok 3: Architektúra MVC, (codeproject.com, 2014)

⁶End-to-end testovanie je metóda použitá na testovanie, či tok aplikácie sa správa od začiatku do konca tak ako sa predpokladá. Účelom je identifikovať závislosti systému a aby zabezpečili, že správne informácie sa odovzdávajú medzi rôznymi komponentmi systému a systémom.

4.5 HTML, CSS

HTML pochádza z prototypu vytvoreným Timom Berners-Lee v roku 1992. Berners-Lee cítil, že existovala možnosť prepojenia dokumentov dohromady pomocou hypertextu, a vyvinul koncept HTML. Nevýhodou bolo, že komerčné hypertextové balíčky, ktoré boli k dispozícii v tom čase, ako je Zog a Intermedia, boli upravené tak, aby vyhovovali rôznym typom počítačov ale boli príliš nejednoznačné.

Berners-Lee vytvoril jazyk HyperText Markup Language (HTML) a zároveň vyvinul protokol pre prístup k textu z iných dokumentov cez hypertextové odkazy, ktorý nazval HTTP. HTML sám bol odvodený od značkovacieho jazyka s názvom Standard Generalized Markup Language (SGML).

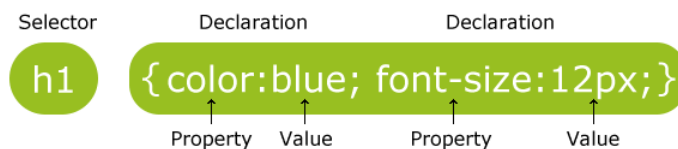
Ďalšie prielom v oblasti bolo zavedenie CSS spolu s HTML 4.0. Prioritou zavedenia CSS bolo, že web dizajnéri a vývojári používajú HTML pre účely formátovania. Formátovanie a štylizovanie webovej stránky pomocou HTML marilo účel HTML, pretože HTML prvky a atribúty by mali definovať iba štruktúru webovej stránky. Účelom CSS bolo oddeliť štylizovanie od štruktúrnych značiek. So zavedením CSS môžeme oddeliť prezentáciu od obsahu. Výsledkom je, že formátovanie sa oddelilo od HTML dokumentu a je uložené v samostatnom súbore, ktorý môže byť pripojený do dokumentu pomocou pripojovacej značky (SHENOY, 2014).

CSS funguje tak, že umožňuje spojiť pravidlá s prvkami, ktoré sa objavujú na webovej stránke. Tieto pravidlá upravujú, ako by mal byť obsah týchto prvkov zobrazený. CSS pravidlo sa skladá z dvoch častí:

- Selektor - ten označuje, ku ktorému prvku alebo prvkom deklarácie sa vzťahuje
- Deklarácia - stanovuje, ako by mali byť prvky uvedené v selektore štylizované

Deklarácia je tiež rozdelená do dvoch častí, oddelených dvojbodkou:

- Vlastnosť - ktorá sa vzťahuje k vybranému prvku, ktorý chcete ovplyvniť.
- Hodnota - ktorá je špecifikáciou pre túto vlastnosť (Duckett, 2010).



Obrázok 4: Zloženie CSS príkazu, (w3schools.com, 2017)

4.6 Bootstrap

Bootstrap je open source produkt vytvorený Markom Ottom a Jacobom Thorntonom. Od kedy bol Bootstrap uvedený v Auguste 2011, stal sa okamžite populárny. Vyvíjal sa z projektu, ktorý bol celý riadený CSS a zahrňal celý rad JavaScript pluginov a ikon. Umožňuje responzívny web design a ako podklad obsahuje dvanásť stĺpcovú, 940 pixelov širokú mriežku.

Jednou z výborných vymožeností je budovací nástroj na webovej stránke Bootstrap-u, kde si môžete prispôbiť kód podľa vlastných potrieb výberom, ktoré CSS a JavaScript funkcie chcete zahrnúť do svojho projektu. Toto všetko umožňuje vývoju frontendu webu postupovať vpred, budovať na stabilných základoch inovatívneho designu a vývoja(Sakla, 2015).

4.7 PHP

PHP je najpopulárnejší skriptovací jazyk na strane servera pre vývoj webových aplikácií, ktorý poháňa odhadom 78,9 % z celkového počtu webových stránok. Bol vytvorený Rasmusom Lerdorfom v roku 1995, PHP bolo pôvodne akronymom "Personal Home Page (Tools)", aj keď teraz je viac známy pre skratku "PHP: Hypertext Preprocessor". Jazyk je riadený, sledovaný a vyvinutý skupinou vývojárov známou ako PHP Group.

PHP kód je najčastejšie interpretovaný, spracovávaný a zobrazovaný pomocou webového serveru s nainštalovaným PHP procesorovým modulom, umožňujúcim uloženie PHP kódu medzi značky HTML v súbore s príponou .php.

Dnes je PHP vývoj zameraný hlavne na skriptovanie na strane servera a nie na skriptovanie univerzálnych úloh. PHP sa používa predovšetkým na spracovanie komplexných dát ktoré umožňuje dynamickým dátam zobrazenie na webových stránkach, napríklad matematických výpočtov alebo interakciu s databázou. To umožňuje vývojárom, aby kedysi statický obsah HTML umožnil reagovať na požiadavky používateľov alebo urobiť to isté s permanentne uloženými dátami, ktoré sú umiestnené v databáze. PHP je zamerané na vývoj webových aplikácií, čo z neho robí jasnou voľbou pre vývojárov pri vytváraní webových aplikácií alebo webových stránok (Hopkins, 2013).

5 Návrh a implementácia

V nasledujúcej kapitole sa budem zaoberať návrhom aplikácie a jej následným implementovaním. V prvej časti uvediem stanovené požiadavky na aplikáciu. Následne bude znázornený a popísaný entitno-relačný diagram a na koniec bude kapitola obsahovať ukážky a popísanie implementácie aplikácie.

5.1 Požiadavky na aplikáciu

Na začiatok pred samotným návrhom a implementáciou je dôležité popísať požiadavky, ktoré má aplikácia spĺňať. Požiadavky na aplikáciu stanovujú nároky užívateľa definujúce funkcionality, chovanie a vzhľad aplikácie. Tieto požiadavky sú rozdelené na požiadavky funkčné a nefunkčné.

5.1.1 Funkčné požiadavky

Aplikácia by mala byť v prvom rade jednoducho ovládateľná ako pre žiakov tak pre učiteľov. Z hľadiska učiteľského prístupu k aplikácii, teda administratívneho, by aplikácia mala spĺňať nasledujúce požiadavky:

- Prihlásenie do aplikácie.
- Vzhľadom k tomu, že aplikácia je tvorená na mieru pre danú školu registráciu užívateľov bude vykonávať admin.
- Vytváranie a pridávanie testov.
- Pri vytváraní jednotlivých otázok bude tvorca môcť vybrať typ danej otázky.
- Pridávanie materiálov na študovanie.
- Prehľad študentov.
- Prehľad výsledkov odovzdaných testov.

Z pohľadu študenta:

- Prihlasovanie sa do aplikácie pomocou pridelených prihlasovacích údajov.
- Prehľad materiálov pre daný ročník, v ktorom sa študent nachádza.
- Prehľad testov pre navštevovaný ročník.
- Precvičovanie znalostí pomocou slepých máp.
- Prehľad výsledkov svojich odovzdaných testov.

5.1.2 Nefunkčné požiadavky

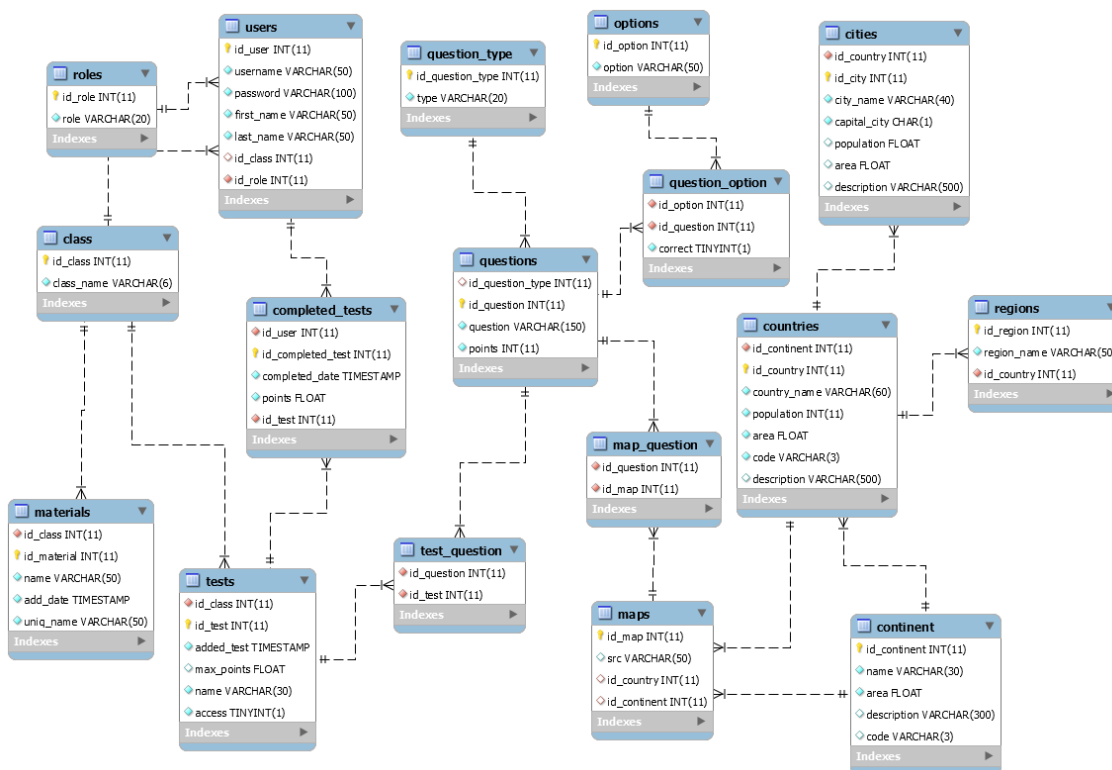
Nefunkčné požiadavky definujú vlastnosti systému, ktoré by mohli určitým spôsobom obmedziť či narušiť funkčnosť aplikácie.

Medzi nefunkčné požiadavky patria:

- Aplikácia bude fungovať na bežne používaných webových prehliadačoch ako sú Google Chrome, Mozilla Firefox, Opera či Internet Explorer.
- Rozhranie aplikácie by malo byť prijateľné voči užívateľom.
- Aplikácia bude vytvorená pomocou jazykov ako sú PHP, HTML, CSS, JavaScript a s využitím databáze MySQL.

5.2 Návrh databáze

Databáza je jednou z najdôležitejších zložiek pri návrhu aplikácie. Využívame ju pre ukladanie informácií a dát. Aby bolo toto uchovávanie efektívne a práca s databázou jednoduchá je potrebné vytvoriť správny a zodpovedajúci návrh. Návrh databáze pre tvorenú aplikáciu je zobrazený na nasledujúcom obrázku.



Obrázok 5: EER Diagram

Informácie o užívateľoch uchováva tabuľka *users*. Obsahuje základne informácie ako sú prihlasovacie meno a heslo, ktoré v prípade študentov bude pridelené prostredníctvom admina, a taktiež meno a priezvisko. V prípade študenta sa bude v tabuľke ukladať aj údaj o triede, ktorú aktuálne navštevuje, tá je uložená v tabuľke *class*. To či ide o admina alebo bežného používateľa vieme rozlíšiť na základe jeho role, pričom typy rolí sú definované v samostatnej tabuľke *roles*.

Ďalšou z hlavných tabuliek je tabuľka *tests*, ktorá ukladá testy vytvorené adminom, teda učiteľom. Testy sa viažu k jednotlivým triedam. Každý test sa skladá z niekoľkých otázok, pri čom je možné aby jeden test obsahoval rôzne typy otázok. Tieto typy nájdeme v tabuľke *question_type*. Pri otázkach s niekoľkými možnosťami sa tieto možnosti opäť ukladajú v samostatnej tabuľke s názvom *options*.

Databáza obsahuje aj tabuľku s názvom *maps*. Tá uchováva mapy, ktoré sa viažu buď ku kontinentom alebo krajinám.

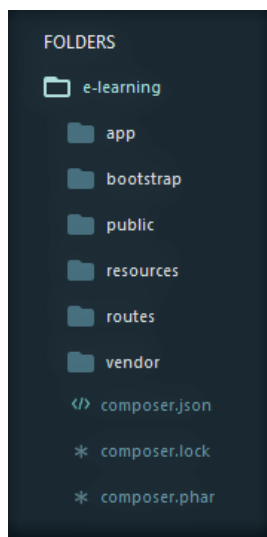
Okrem toho sa budú využívať tiež tabuľky ako *completed_tests* pre zoznam odovzdaných testov používateľmi či *materials* pre vkladanie študijných materiálov učiteľmi.

5.3 Implementácia aplikácie

Ďalšia časť práce sa zaoberá postupmi pri implementácii a samotnou implementáciou. Budú v nej zahrnuté niektoré hlavné ukážky kódu aplikácie a grafického rozhrania. Na začiatok sa pokúsím priblížiť štruktúru aplikácie a obsah niektorých dôležitých častí.

5.3.1 Štruktúra aplikácie

Obrázok pod textom zachytáva základu štruktúru aplikácie.



Obrázok 6: Štruktúra aplikácie

Môžeme tu vidieť priečinok **app**, ktorý sa ďalej člení na zložky *Controller*, *Model* a *Middleware*.

- **Controller** - tu sa nachádza niekoľko samostatných controllerov, ktorých obsahom je predovšetkým práca s databázou, následne sa tu renderujú šablóny či definujú presmerovania pri určitých úkonoch. Priečinok obsahuje základný *Controller*, od ktorého dedia ostatné kontroléry, obsahuje len vytvorený kontajner pre prístup k databáze či routeru, ktorý slúži na presmerovanie. Ďalej sa tu nachádzajú *AuthController*, *MapController*, *MaterialController*, *StudentController*, *TestController* a *UserController*.
- **Model** - pre každý controller je vytvorený model, ktorého úlohou je uchovávať funkcie pre zjednodušenie a hlavne prehľadný kód šablón. V modeli je možné si predpripraviť napríklad jednoduché funkcie pre zobrazovanie určitých údajov vo formáte aký chceme aby bol zobrazený a nemusíme tieto funkcie definovať v šablóne ale ich len jednoducho zavolať.
- **Middleware** - táto zložka obsahuje middlewary, ktoré nám zadovážia aby sa užívateľ nedostal k informáciám alebo funkciám aplikácie ku ktorým nemá oprávnenie.

V priečinku **bootstrap** sa nachádza jediný súbor a to *app.php*, ktorý v sebe ukrýva základne nastavenia aplikácie a konfiguráciu databáze či šablónovacieho systému ako je uvedené nižšie v texte.

Všetky súbory, ktoré slúžia na štylizovanie aplikácie sú umiestnené v zložke **public**, kde sa tiež vyskytujú JavaScriptové súbory.

Zložka **resources** zahŕňa všetky obrázky, ktoré sú použité v aplikácii a to v priečinku *images*, ďalej zložku *views*, ktorá pozostáva zo všetkých šablón a ako posledná zložka ktorá sa tu nachádza je *uploads*.

A dôležitý súbor, ktorý sa nachádza v zložke **routes**, obsahuje všetky základné routy aplikácie. Názornú ukážku takéhoto routu môžete vidieť následne.

```
$this->get('/materials/{id}', MaterialController::class  
    .':showMaterials')->setName('materials');
```

5.3.2 Nastavenie šablónovacieho systému

Ako šablónovací systém bol použitý systém Twig. Na začiatok bolo potrebné Twig nastaviť v hlavnom súbore *app.php*, ktorý je uložený v priečinku *bootstrap*. Nastavenie bolo vykonané pomocou nasledujúceho kódu.

```
$container = $app->getContainer();

$container['view'] = function ($container) {
    $view = new \Slim\Views\Twig(__DIR__.'../../resources/views', [
        'cache' => false
    ]);

    $basePath = rtrim(str_ireplace('index.php', '', $container['request']
        ->getUri()->getBasePath()), '/');
    $view->addExtension(new \Slim\Views\TwigExtension($container['router'],
        $basePath));

    return $view;
};
```

5.3.3 Prístup do aplikácie

Vzhľadom na to, že aplikácia je vytvorená pre súkromné účely a pre konkrétnu osobu, bude možné sa do aplikácie prihlasovať iba ak máte k dispozícii pridelené prihlasovacie údaje a teda registrácia bežného návštevníka nebude možná. Na základe tejto skutočnosti bol pre prihlásenie vytvorený jednoduchý formulár. V prípade registrácie nového prístupového účtu môže túto registráciu vykonať admin.

The image shows a login form centered on a light blue background. The form itself is a yellow-green rectangle. It contains three white input fields: the top one is labeled 'Prihlasovacie meno', the middle one is labeled 'Heslo', and below them is a white button labeled 'Prihlásiť'.

Obrázok 7: Prihlasovací formulár

Zabezpečenie toho aby sa do aplikácie nedostal neprihlásený užívateľ je riešená pomocou middleware s názvom *Unauthenticate* v príslušnom priečinku.

```
public function __invoke($request, $response, $next)
{
    if (!isset($_SESSION['signed_in'])) {
        $response = $response->withRedirect($this->router->
            pathFor('sign_in_from'));
    }
    return $next($request, $response);
}
```

Na základe toho, že nie každý žiak a predovšetkým žiaci nižších ročníkov základných škôl má zriadenú e-mailovú adresu, pri strate hesla či zabudnutí sa môže žiak obrátiť na vyučujúceho s požiadavkou o vygenerovanie nového hesla. Nové heslo sa vygenerujú na základe definovanej funkcie, ktorá je zobrazená nižšie.

```
public function newPswd()
{
    $alphabet = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
        PQRSTUVWXYZ1234567890';
    $pswd = array();
    $alphaLength = strlen($alphabet) - 1;
    for ($i = 0; $i < 8; $i++) {
        $n = rand(0, $alphaLength);
        $pswd[] = $alphabet[$n];
    }
    $password = implode($pswd);
    return $password;
}
```

Na Obrázku 8 môžete vidieť zobrazenie oznámenia o vygenerovaní nového hesla do aplikácie pre dané prihlasovacie meno.

Pre login **j.novak** bolo vygenerované heslo: **9Mw8Lvek**

Späť

Obrázok 8: Úkažka vygenerovaného hesla

5.3.4 Vytváranie testu

Prostredníctvom aplikácie majú učitelia príležitosť vytvárať si vlastné testy pre jednotlivé triedy alebo testy, ku ktorým majú prístup všetci žiaci. Testy sú vytvárané pomocou formulára ktorý má dva hlavné kroky. V prvom kroku zadáte názov testu a vyberiete pre ktorú triedu je test určený. Po potvrdení sa dostanete k druhému formuláru, ktorý po vyplnení spracuje a vloží už samotnú otázku, kde musíte určiť o aký typ otázky sa jedná. Následne sa môžete rozhodnúť či chcete pridať ďalšiu otázku alebo je test kompletný a test vložíte. Formulár vyzerá nasledovne.



Vloženie otázky

Znenie otázky:

Typ otázky:

Možnosti

Správna:

Ďalšia:

Ďalšia:

Ďalšia:

Počet bodov:

+ Ďalšia otázka

Vložť test

Obrázok 9: Vkladanie otázky s možnosťami

Učitelia môžu sprístupniť a odoprieť prístup k testu kedykoľvek pomocou jedného tlačítka v prehľade testov, ako je možné vidieť na Obrázku 10, kde ikona preškrtnutého oka znázorňuje neviditeľnosť testu (Test č.2), a test s ikonou normálneho oka (Test č.1) v stĺpci *Prístup* je sprístupnený pre danú triedu. To im umožňuje testy prichystať vopred.

Dátum vytvorenia	Názov testu	Trieda	Maximálny počet bodov	Pisaf test	Prístup
2017-05-19 16:58:20	Test č.1	3A	4	→	👁
2017-05-19 16:58:09	Test č.2	3A	4	→	👁

Obrázok 10: Vkladanie otázky s možnosťami

5.3.5 Učebné materiály

Súčasťou aplikácie sú tiež učebné materiály, ktoré učitelia pridávajú a sprístupňujú jednotlivým triedam. Pre úspešne načítanie a uloženie do databázy platí, že súbor nemôže byť väčší než 5 MB s určitou odchýlkou smerom na hor.

Prácu so súborami ich vkladanie do databáze zabezpečuje kód napísaný v jazyku PHP uvedený nižšie.

```

$file = $_FILES['file'];
$fileName = $_FILES['file']['name'];
$fileTmpName = $_FILES['file']['tmp_name'];
$fileSize = $_FILES['file']['size'];
$fileError = $_FILES['file']['error'];
$fileType = $_FILES['file']['type'];

$fileExtension = explode('.', $fileName);
$fileActualExt = strtolower(end($fileExtension));

if($fileError == 0){
    if($fileSize < 6000000){

        $fileUniqName = uniqid('', true) . "." . $fileActualExt;
        $fileDestination = '../resources/uploads/' . $fileUniqName;
        move_uploaded_file($fileTmpName, $fileDestination);

        $stmt= $this->c->db->prepare("INSERT INTO materials
            ."(id_class ,name, uniq_name) VALUES (:id ,: name ,:uName)");
        $stmt->bindValue(":name", $postValues['name']);
        $stmt->bindValue(":id", $postValues['class']);
        $stmt->bindValue(":uName", $fileUniqName);
    }
}

```

```

        $stmt->execute ();
    }
}

```

Tieto materiály si môžu žiaci následne stiahnuť. Sťahovanie umožňuje nasledujúca funkcia `download($request,$response,$args)`.

```

public function download($request , $response , $args)
{
    $file = $this->c->db->query("SELECT * from materials where "
        . " id_material = " . $args['id'])
        ->fetchAll(PDO::FETCH_ASSOC)[0];
    $path="../resources/uploads/" . $file['uniq_name'];

    header('Content-Type: applicacion/octet-stream');
    header('Content-Disposition: attachment; filename="'.basename($path).');
    header('Content-Length: ' . filesize($path));
    readfile($path);
}

```

5.3.6 Slepé mapy

Aplikácia tiež zahŕňa slepé mapy. Žiaci si tu môžu precvičovať svoje znalosti a cíbit orientačné schopnosti.

Mapy v aplikácii sú vkladané ako svg súbor, prostredníctvom JavaScriptu za použitia frameworku AngularJS ako môžeme vidieť na ukážke. Tento kód je situovaný v súbore `directives.js`.

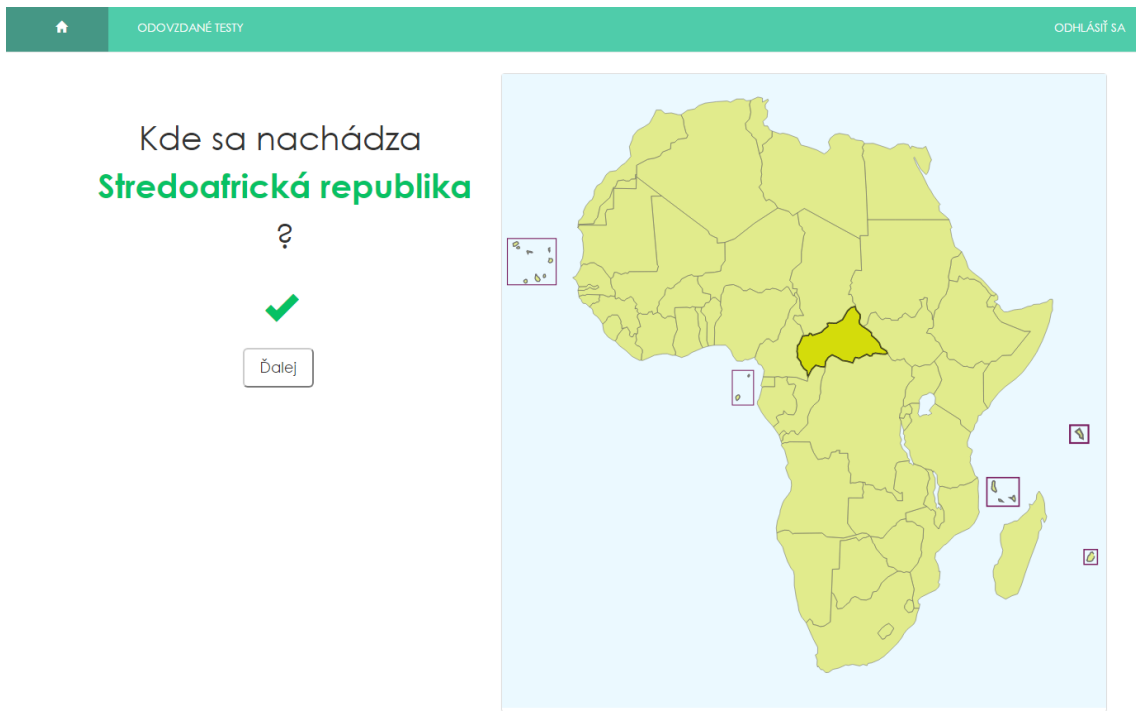
```

angular.module('OutlineMap').directive('map', ['$compile',
function ($compile) {
    return {
        restrict: 'A',
        templateUrl: function(elem, attr) {
            console.log( attr.data);
            return '/e-learning/resources/img/'+ attr.src;
        },
        link: function (scope, element, attrs) {
            var states = element[0].querySelectorAll('.land');
            angular.forEach(states, function (path, key) {
                var stateElement = angular.element(path);
                stateElement.attr("state", key);
                $compile(stateElement)(scope);
            })
        }
    }
}

```

```
    },  
    controller: "MainCtrl"  
  }  
});  
  
angular.module('OutlineMap').directive('state', ['$compile',  
function ($compile) {  
  return {  
    restrict: 'A',  
    link: function (scope, element, attrs) {  
      scope.elementId = element.attr("id");  
      element.attr("ng-click", "regionClick()");  
      element.removeAttr("state");  
      $compile(element)(scope);  
    },  
    controller: "MainCtrl"  
  }  
});
```

V rámci aplikácie práca s mapou vyzerá nasledovne. Systém vám položí otázku a keď zvládnete nájsť správnu odpoveď, zobrazí sa tlačítko **Ďalej** a môžete sa posunúť na ďalšiu.



Obrázok 11: Slepá mapa

6 Testovanie aplikácie

Po dokončení implementácie bolo potrebné aplikáciu otestovať či je aplikácia pripravená pre nasadenie a používanie v praxi.

Pre testovanie aplikácie bolo zvolené akceptačné testovanie, ktoré vykonáva zákazník. Pri tomto testovaní nie je hlavným cieľom zistiť prípadné chyby ale hlavne či je aplikácia zhotovená adekvátne a vyhovujúco pre danú cieľovú skupinu. Môžeme tiež zistiť určité nedostatky a na základe týchto testov upraviť aplikáciu do požadovanej formy.

6.1 Testovanie používateľom

Testovanie začalo jednoduchým prihlásením a následne vyskúšaním všetkých funkcií, ktorými aplikácia disponuje. Testovateľ nemal pri skúšaní aplikácie žiadny problém čo sa týkalo orientácie v aplikácii, či hľadaním určitých aktivít.

Testovanie zahrňovalo:

- Registrácia študentov/učiteľov.
- Vytváranie testov a ich sprístupňovanie žiakom.
- Pridávanie učebných materiálov prostredníctvom súborov rôzneho typu a následne ich stiahnutie z aplikácie do počítača.
- Funkčnosť testovania orientácie prostredníctvom slepých máp.
- Zobrazovanie prehľadov vypísaných testov, vložených materiálov, odovzdaných testov či prehľad žiakov.

6.2 Výsledky a zhodnotenie testovania

Všetky vyššie uvedené body zhrňujúce testovanie mali pozitívnu odozvu. Testovanie funkčnosti slepých máp taktiež prebehlo v poriadku, ale boli tu určité výhrady čo sa týkalo zobrazenia. Slepá mapa ktorá obsahovala menšie štáty či ostrovy, ktoré mali malú plochu zakliknutia, boli ťažšie zaznačiteľné. To mohlo byť pre žiakov nízkych ročníkov náročnejšie z dôvodu ich stále vyvíjajúcich sa motorických funkcií.

Na základe týchto pripomienok bola mapa upravená zväčšením celkového zobrazenia mapy a menšie plochy boli ešte zväčšene, pri čom samostatné ostrovy či súostrovia boli taktiež zväčšené a zvýraznené prostredníctvom orámovania.

7 Diskusia

Vytvorená aplikácia spĺňa všetky požiadavky, ktoré boli stanovené pred samotnou realizáciou. Napriek tomu sa dá aplikácia rozširovať a dopĺňať ďalšími funkciami a vymoženosťami. Po pri implementácii sa vynámali rôzne možnosti ako by šlo aplikáciu rozšíriť.

Jednou z požiadavkou bolo aby sa dali vytvárať testy pre jednotlivé triedy. To by sa dalo rozšíriť databázou otázok, ktoré by boli priradené k samostatne definovaným témam. Testy by sa následne pri výbere určitej témy samostatne generovali. Táto možnosť však nebola implementovaná do súčasnej aplikácie.

Rovnako by sa do aplikácie dalo implementovať viacej hier zameraných na rôzne témy, čo by pomohlo ešte viac prilákať a zaujať žiakov na škole.

Pre ešte plnohodnotnejší a komplexnejší e-learningový systém by sa dalo do webovej aplikácie implementovať poprípade aplikáciu napojiť na elektronickú žiacku knižku. Tak by mali žiaci prehľad o svojich známkach na jednom mieste.

Ďalším rozšírením by mohlo byť zadávanie úloh a zadaní na doma do systému. Výhodou by napríklad bolo pre žiakov, ktorý z nejakého dôvodu zameškali nejaké hodiny výuky.

Aplikácia by sa taktiež dala rozšíriť tak, že by nebola zameraná len na jeden predmet, v tomto prípade zemepis, ale zahrňovala by aj iné predmety, ktoré sa na škole vyučujú.

8 Záver

Cieľom tejto práce bolo vytvoriť jednoduchú a prehľadnú webovú aplikáciu, ktorá bude slúžiť ako e-learningová opora pre základnú školu. Vzhľadom k tomu, že aplikácia mala byť vytvorená aj pre žiakov základnej školy nižších ročníkov, jej užívateľské prostredie malo byť prívetivé a aplikácia jednoduchá na ovládanie. Aplikácia bola realizovaná predovšetkým na základe stanovených požiadavkou.

Po oboznámení sa s hlavnou témou bakalárskej práce bola následne vytvorená analýza už existujúcich aplikácií zameraných na rovnakú problematiku. K analýze boli vybrané presne tri aplikácie. Napokon nasledovalo ich zhodnotenie a výsledok celkovej analýzy.

Práca pokračuje špecifikáciou funkčných a nefunkčných požiadavkou, kde funkčné požiadavky boli rozdelené na základe pohľadu používateľa a to z pohľadu študenta a z pohľadu učiteľa. Návrh databáze zobrazený pomocou entitno-relačného modelu bol vytvorený s ohľadom na tieto požiadavky.

K procesu implementácie boli vybrané technológie určené na vývoj aplikácií. Zjednodušenie implementácie jednoznačne umožnili použité frameworky a to hlavne mikro framework Slim, framework AngularJS a čo sa týka designu a responzivity framework Bootstrap.

V konečnom dôsledku môžem povedať, že aplikácia splňuje požiadavky stanovené zadávateľom a pôsobí ako základ pre prípadne rozšírenia a doplnenia ďalších funkcií.

9 Zoznam literatúry

- ACUNETIX. *Web Applications: What are They? What of Them?* [online]. 2014 [cit. 2017-04-21]. Dostupné z: <https://www.acunetix.com/websitesecurity/web-applications/> .
- CLARK, R. C., MAYER, R. E. *E-learning and the science of instruction: proven guidelines for consumers and designers of multimedia learning*. 3rd ed. San Francisco, CA: Pfeiffer, 2011. ISBN 978-047-0874-301. .
- CODEPROJECT. *AngularJS With MVC Web API (ASP.NET MVC RESTful Service)* [online]. 2014 [cit. 2017-04-21]. Dostupné z: <https://www.codeproject.com/Articles/826307/AngularJS-With-MVC-Web-API> .
- DUCKETT, Jon. *Beginning HTML, XHTML, CSS, and JavaScript*. Chichester: John Wiley [distributor], 2010. ISBN 978-0-470-54070-1. .
- FREEMAN, Adam. *Pro AngularJS*. New York, NY: Apress, 2014. ISBN 978-1-4302-6448-4. .
- HOPKINS, Callum. *Jump start PHP*. Collingwood, Victoria: SitePoint, 2013. ISBN 978-0-9874674-1-6. .
- NATIONS, Daniel. *Improve Your Understanding of Web Applications*. Lifewire [online]. 2016 [cit. 2017-04-14]. Dostupné z: <https://www.lifewire.com/what-is-a-web-application-3486637> .
- SAKLA, R. A. *Learn 'Mobile First' Web Design Using Bootstrap 3* [online]. 2015 [cit. 2017-04-05]. Dostupné z: <https://www.free-ebooks.net/ebook/Learn-Mobile-First-Web-Design-Using-Bootstrap-3> .
- SETERRA. *Seterra Onilne* [online]. 2017 [cit. 2017-04-02]. Dostupné z: <http://online.seterra.com/> .
- SHENOY, Aravind. *Thinking in HTML* [online]. Birmingham, UK: Packt Publishing, 2014 [cit. 2017-04-14]. Dostupné z: <https://www.packtpub.com/packt/free-ebook/thinking-in-html> .
- SLEPEMAPY. *Slepé mapy* [online]. 2017 [cit. 2017-04-02]. Dostupné z: <https://sk.slepemapy.cz/> .
- SLIMFRAMEWORK. *Middleware* [online]. 2017a [cit. 2017-04-21]. Dostupné z: <https://www.slimframework.com/docs/concepts/middleware.html> .
- SLIMFRAMEWORK. *Slim - a micro framework for PHP* [online]. 2017b [cit. 2017-04-02]. Dostupné z: <https://www.slimframework.com/> .

- TOPOROPA. *Toporopa* [online]. 2017 [cit. 2017-04-02]. Dostupné z: <http://www.toporopa.eu> .
- TWIG. *Twig - The flexible, fast, and secure template engine for PHP* [online]. 2017 [cit. 2017-04-05]. Dostupné z: <http://twig.sensiolabs.org> .
- W3SCHOOLS. *CSS Syntax and Selectors* [online]. 2017 [cit. 2017-04-21]. Dostupné z: https://www.w3schools.com/css/css_syntax.asp .
- NETTE. *Nette Framework* [online]. 2016 [cit. 2017-04-19]. Dostupné z: <https://nette.org/cs/> .
- ZAKAS,Nicholas C. *Professional JavaScript for web developers*. 3rd ed. Indianapolis: Wiley, 2012. ISBN 978-1-118-02669-4. .
- MILLARES, Gerard *Top 5 Programming Languages Used In Web Development*. Stone River eLearning [online]. 2015 [cit. 2017-04-19]. Dostupné z: <http://blog.stoneriverelearning.com/top-5-programming-languages-used-in-web-development/> .
- BELITSOFT, *The TOP 10 Advantages of Using Laravel PHP framework*. [online]. 2016 [cit. 2017-04-19]. Dostupné z: <https://belitsoft.com/laravel-development-services/10-benefits-using-laravel-php-framework> .
- AGRIYA *New Amazing Features of Slim Framework*. [online]. 2016 [cit. 2017-04-19]. Dostupné z: <https://www.agriya.com/blog/2016/10/25/new-amazing-features-and-advantages-slim-framework/> .
- NOVOSIOLOV, Andrzej *AngularJS advantages and limitations*. Softelegance's blog [online]. 2015 [cit. 2017-04-19]. Dostupné z: <http://blog.softelegance.com/angularjs/angularjs-advantages-and-limitations/> .

Prílohy

A Grafický design úvodnej stránky aplikácie



Obrázok 12: Grafický design úvodnej stránky

B Ukážka písania testu

ODHLÁSIŤ SA

ODOVZDANÉ TESTY

TEST 1

Aké je hlavné mesto Islandu?

Oslo

Belehrad

Berlín

Reykjavík

Ako sa nazýva najdlhšia rieka na území Slovenska?

Dunaj

Váh

Hron

Morava

Najvyšším vrcholom slovenska je Gerlachovský štít.

Áno

Nie

Odoslať test

Obrázok 13: Písanie testu

C Ukážka administračného rozhrania

Prehľad študentov Prehľad učiteľov Učebné materiály Testy Pridať používateľa ODHLÁSIŤ SA

Študenti

Trieda:

Meno	Priezvisko	Prihlasovacie meno	Trieda	Generovať nové heslo
Janko	Kováčik	kovacik	2B	↑
Jurko	Mrkvička	mrkvička	2B	↑
Ján	Novák	jnovak	2B	↑
Elena	Fodorová	fodorova	3B	↑

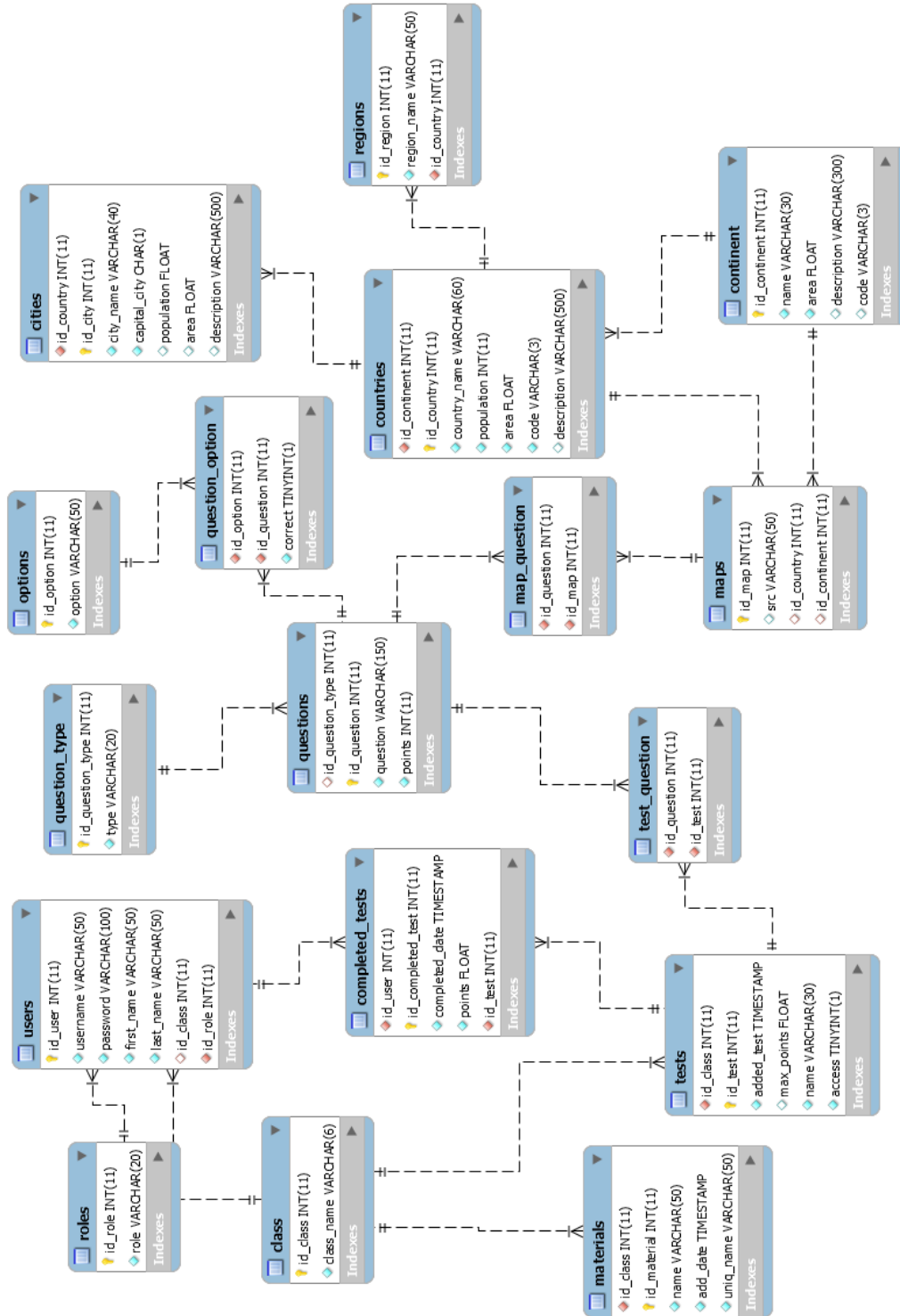
Obrázok 14: Prehľad študentov v administračnom rozhraní

The image shows a web interface for adding educational material. At the top is a green navigation bar with the following items from left to right: a home icon, 'Prehľad študentov', 'Prehľad učiteľov', 'Učebné materiály', 'Testy', 'Pridať používateľa', and 'ODHLÁSIŤ SA'. Below the navigation bar, the main content area has a title 'Pridanie materiálu' and a form with the following fields:

- Názov učebného materiálu:** A text input field.
- Určený pre triedu:** A dropdown menu with '1.A' selected.
- Súbor do 5MB:** A file upload area with a 'Choose File' button, the text 'No file chosen', and a 'Pridaj' button.

Obrázok 15: Formulár na upload učebného materiálu

D Návrh databáze



Obrázok 16: EER diagram

E Priložené CD

Priložené CD obsahuje:

- Súbory so zdrojovým kódom webovej aplikácie
- Súbor s exportovanou databázou
- Obrázky použité v Bakalárskej práci
- Bakalárska práca v PDF súbor.