

Fakulta matematiky, fyziky a informatiky
Univerzita Komenského v Bratislave

Návrh
Vybrané slová

Martin Masár
Adrián Pavčo
Peter Zach
Michal Hradečný

Tvorba informačných systémov 2015/16

Obsah

0. Úvod.....	4
0.1 Cieľ dokumentu.....	4
1. Popis používateľských rozhraní.....	5
1.1 Neprihlásený používateľ.....	5
1.2 Prihlásený používateľ – žiak.....	5
1.3 Prihlásený používateľ – učiteľ.....	5
2. Entitno-relačný diagram.....	6
3. Stavový diagram.....	7
4. Use-case diagram	8
4.01 Pozri zoznam.....	9
4.02 Uprav zoznam.....	9
4.03 Prihlásenie.....	9
4.04 Zobraz moje výsledky.....	10
4.05 Cvičenie.....	10
4.06 Individuálne cvičenie.....	11
4.07 Dopln správne písmeno.....	11
4.08 Vyber zlé slovo.....	11
4.09 Prirad' slovo k obrázku.....	12
4.10 Zobraz nápovedu.....	12
5. Prílohy.....	13
5.1 Slovník pojmov.....	13
6. Komponenty.....	14
6.1 Komponentový diagram.....	14
6.2 Popis komponentov.....	14
6.2.1 Komponent Autorizácia.....	14
6.2.2 Komponent Správa Používateľov.....	15
6.2.3 Komponent Správa Aktivít.....	15
6.2.4 Komponent Aktivit.....	15
6.2.5 Komponent Template.....	15
6.2.6 Komponent Controllers.....	15
7. Dátový model.....	16
7.1 Zobrazenie dátového modelu.....	16
7.2 Popis dátového modelu.....	16
7.2.1 WybSlova_slova.....	16
7.2.2 WybSlova_ucitelia.....	17
7.2.3 WybSlova_vysledky.....	17
7.2.4 WybSlova_ziaci.....	17
7.2.5 Vzťahy medzi tabuľkami.....	17
8. Analýza technológií.....	18
8.1 Node.js.....	18
8.2 HyperText Markup Language (HTML).....	18
8.3 Javascript.....	18
8.4 Kaskádové štýly CSS.....	18
8.5 AngularJS.....	18
8.6 MongoDB.....	18
8.7 Web Audio API.....	18
9. Objektový návrh.....	19
9.1 Data-flow diagram.....	19
9.1 Sekvenčný diagram prihlasovania.....	20

9.3 Triedny návrh.....	20
9.3.1 Premenná cvičenie.....	20
9.3.2 Premenná úloha.....	21
9.3.3 Premenná slovo.....	21
9.3.4 Funkcie.....	21
9.4 Návrh kontrolerov.....	23
9.4.1 Kontroler student.....	23
9.4.2 Kontroler teacher.....	23
9.4.3 Kontroler frontend_kontroler.....	23
9.4.4 Kontroler backend_kontroler.....	24
9.4.5 Kontroler activities.....	24
9.4.6 Kontroler admin.....	24
9.4.7 Kontroler admin_activity.....	24
9.4.8 Kontroler admin_students.....	24
9.5 MongoDB databáza a MongoJS.....	25
9.6 Testovacie scenáre.....	26
9.6.1 Načítanie slova.....	26
9.6.2 Nájdenie slova.....	26
9.6.3 Kontrola kontroly úlohy.....	26
9.6.4 Tvorba cvičenia.....	26
9.6.5 Výpis úlohy.....	27
9.6.6 Výpis cvičenia.....	27
9.5.7 Kontrola prihlasovania.....	27

0. Úvod

0.1 Cieľ dokumentu

Dokument spája konceptuálnu analýzu, analýzu technológií a návrh. Obsahuje entitno-relačný, stavový a use-case diagramy znázorňujúce fungovanie systému z rôznych pohľadov, popis používateľského rozhrania, formalizáciu dátového modelu systému, jeho dekompozíciu na nezávislé časti(komponenty), zhrnutie technológií, ktoré budú využité pri jeho tvorbe, data-flow diagram, sekvenčný diagram, triedny návrh a testovacie scenáre.

1. Popis používateľských rozhraní

Práca s aplikáciou je rozdelená do nasledovných kategórií na základe vzťahu medzi aplikáciou a používateľom. V prílohe Pouzivatelске_rozhrania.zip sú znázornené graficky.

1.1 Neprihlásený používateľ

Aplikácia poskytuje prístup ku cvičeniam, zoznamu vybraných slov, nápovede a návrat na hlavnú stránku.

Po kliknutí na tlačidlo Cvičenia sa dá používateľovi na výber z ponuky troch cvičení:

1. Dopĺňanie i/y do zobrazených slov a následné kontrolovanie správnosti
2. Sú zobrazené obrázky a slová, ktoré je nutné k sebe správne priradiť
3. Je zobrazená množina slov, z ktorej je potrebné vybrať práve jedno chybné

Po kliknutí na tlačidlo Slovíčka sa zobrazí zoznam všetkých vybraných slov.

Po kliknutí na tlačidlo Nápoveda sa zobrazí nápoveda o aplikácii.

Po kliknutí na tlačidlo Úvod sa zobrazí úvodná stránka aplikácia.

1.2 Prihlásený používateľ – žiak

Poskytuje to isté rozhranie ako pre neprihláseného používateľa, s niekoľkými zmenami:

Žiak pracuje pod svojím kontom na stránke multimedialnej čítanky.

Po kliknutí na tlačidlo Cvičenia sa dá používateľovi na výber medzi individuálnym resp. skupinovým cvičením. Výsledky skupinového cvičenia sa neukladajú do výsledkovej histórie žiaka, individuálne áno. Po kliknutí na jedno z týchto dvoch tlačidiel sa otvorí ponuka troch cvičení ako pre neprihláseného používateľa.

Možnosť kliknúť na tlačidlo Moje výsledky, po čom sa mu zobrazí história jeho cvičení a ich hodnotenie.

1.3 Prihlásený používateľ – učiteľ

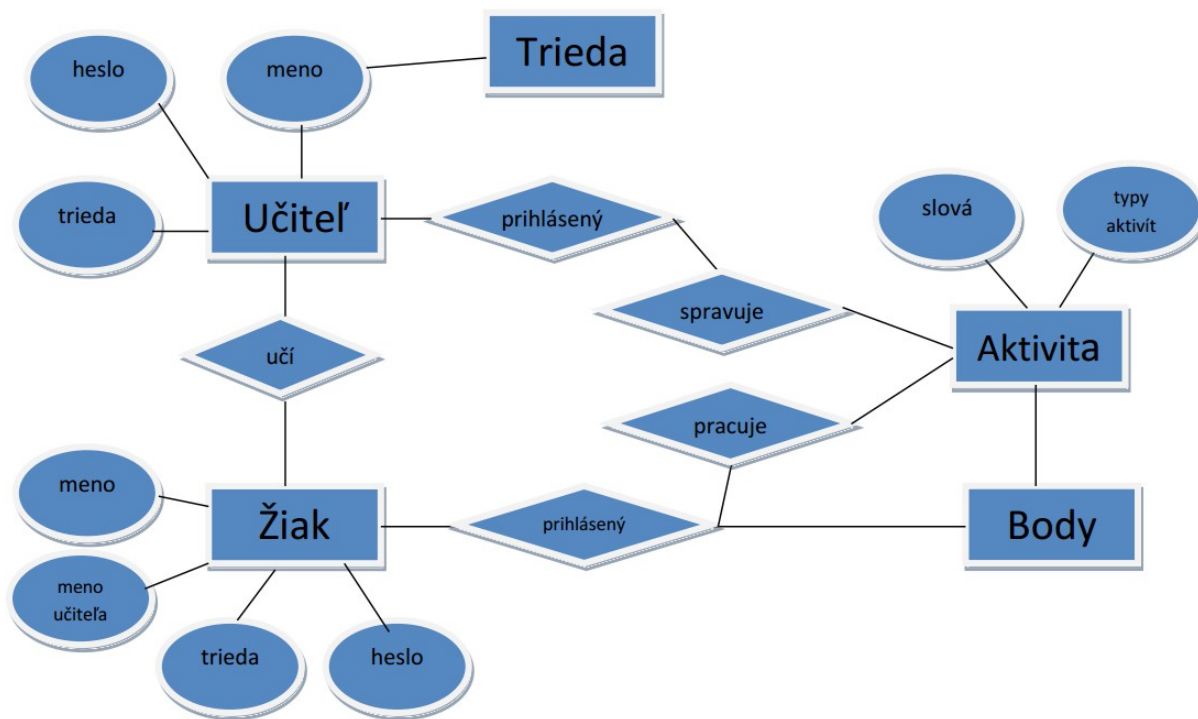
Poskytuje to isté rozhranie ako pre prihláseného žiaka, s niekoľkými zmenami:

Učiteľ pracuje pod svojím kontom na stránke multimedialnej čítanky.

Môže zasahovať do databanky slov tak, že stlačí na tlačidlo Slovíčka a následne na Úprava, po čom sa mu otvorí rozhranie na úpravu slov. Môže ich meniť, pridávať, alebo odstraňovať.

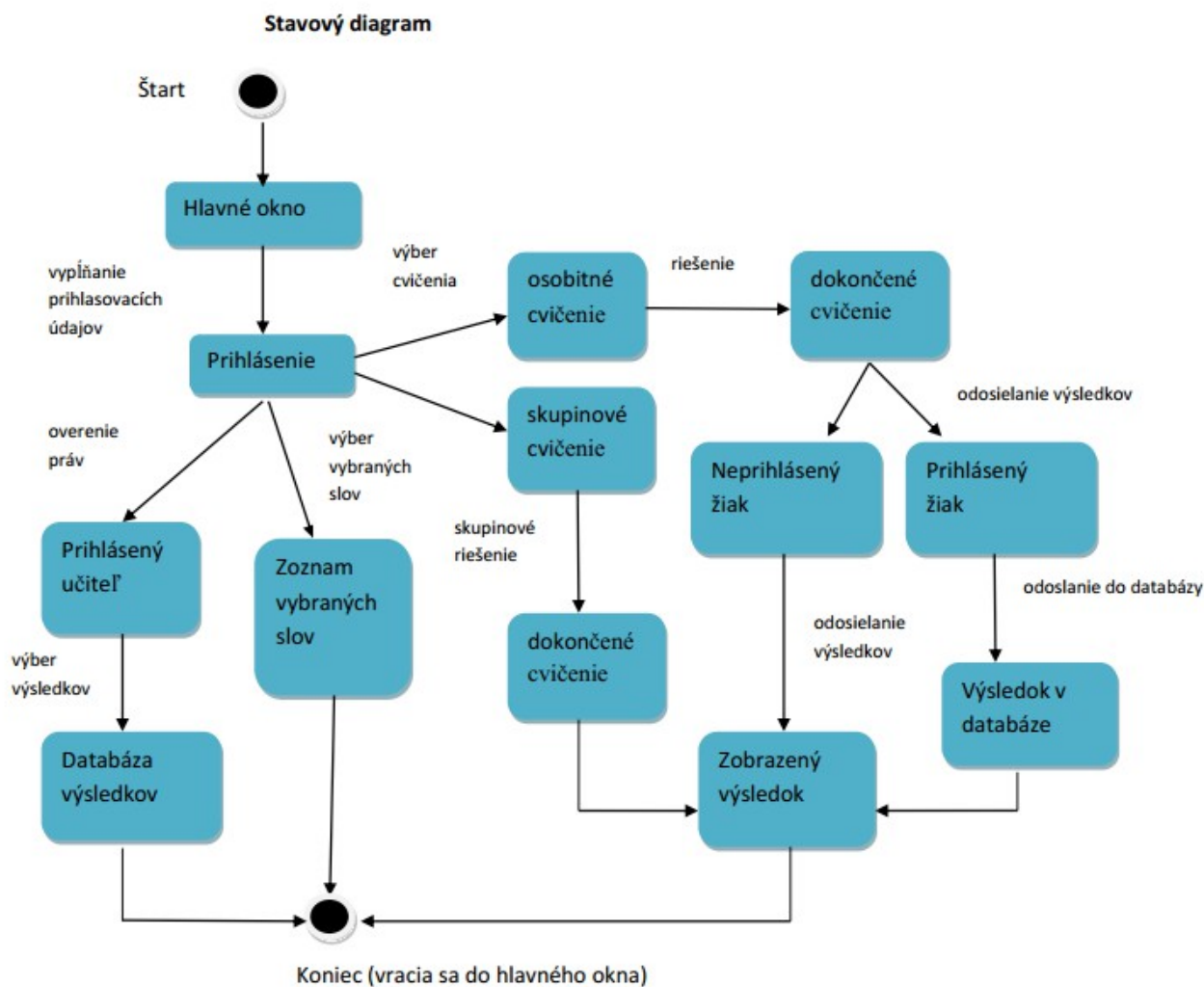
2. Entitno-relačný diagram

Entitno relačný diagram zobrazuje entity a relácie, teda vzťahy medzi jednotlivými entitami vystupujúcimi v diagrame. Tento diagram sa stane základom pre návrh samotného systému a databázy.



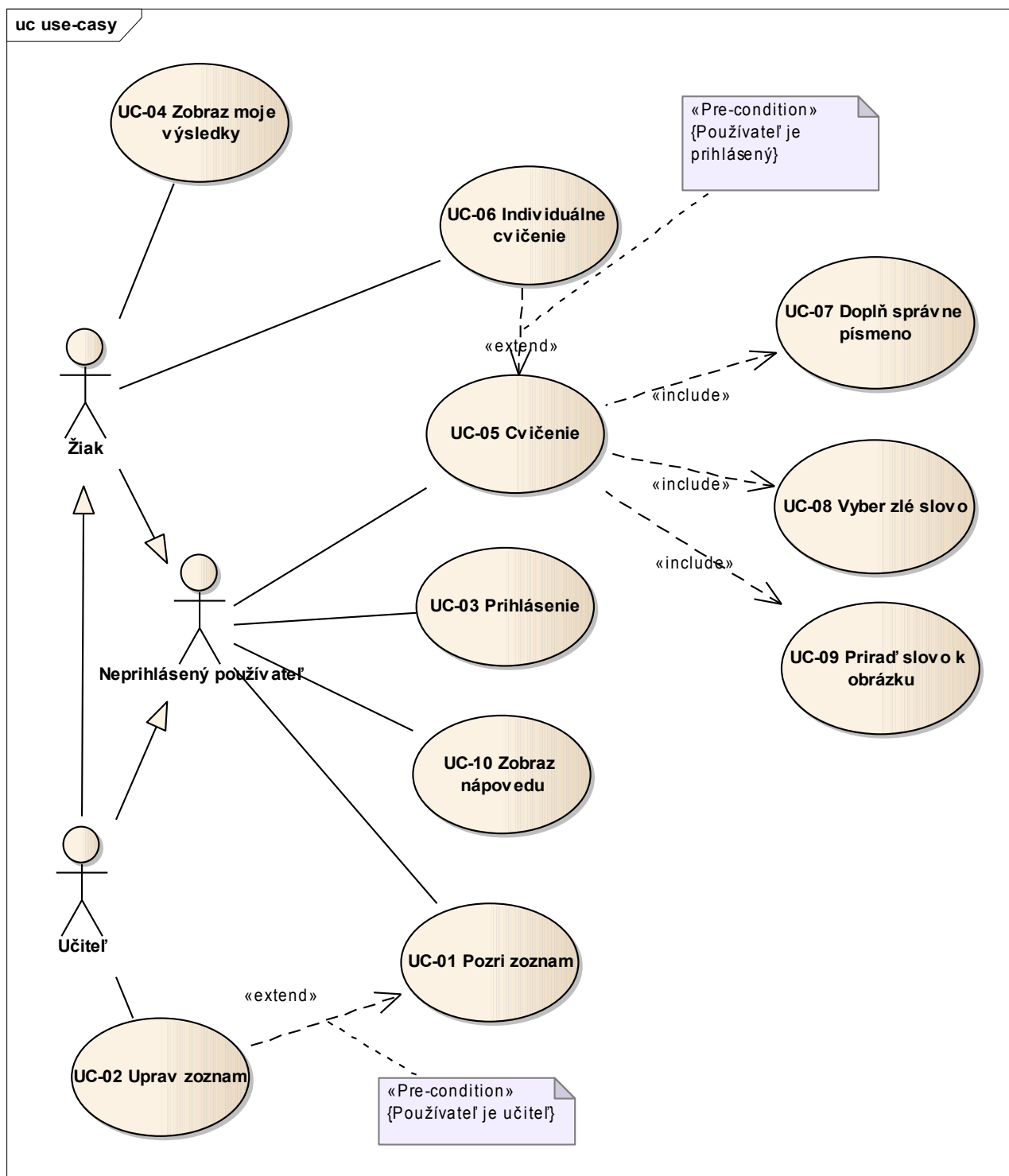
3. Stavový diagram

Stavový diagram predstavuje množinu stavov (znázornených v modrých obdĺžnikoch) vyskytujúcich sa v aplikácii po spustení a procesov (popri smerovaných čiarach medzi obdĺžnikmi), ktorými sa aplikácia do daných stavov dostáva. V našom prípade sú stavy v aplikácii do veľkej miery ovplyvnené tým, aký typ používateľa je momentálne prihlásený.



4. Use-case diagram

Use-case diagram hovorí ako sa program správa z pohľadu používateľa. Zobrazuje, čo musí používateľ urobiť, aby sa dostal k požadovanému výsledku a ako to z pohľadu systému súvisí so zvyškom programu. Taktiež ukazuje možné spôsoby opätovného použitia prvkov a ako so sebou súvisia rôzni používatelia.



4.01 Pozri zoznam

Identifikátor	UC-01	
Názov	Pozri zoznam	
Opis	Zobrazí používateľovi zoznam vybraných slov	
Používatelia	K zoznamu majú prístup všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Používateľ stlačí tlačidlo „Slovíčka“
	2	Systém zaznamená stlačenie
	3	Systém zobrazí zoznam vybraných slov

4.02 Uprav zoznam

Identifikátor	UC-02	
Názov	Uprav zoznam	
Opis	Učiteľ upraví zoznam vybraných slov	
Vstup. podm.	Používateľ musí byť prihlásený ako učiteľ	
Výstup. podm.	Musia byť potvrdené zmeny, inak sa nevykonajú	
Používatelia	Kto ho používa	
Základná postupnosť	Krok	Činnosť
	1	UC-01 Pozri zoznam
	2	Učiteľ stlačí tlačidlo „Úprava“
	3	Systém zobrazí rozhranie na úpravu, mazanie a pridávanie slov
	4	Učiteľ vykoná zmeny
	5	Učiteľ stlačí tlačidlo „Uložiť“
	6	Systém zmeny uloží
	7	Systém zobrazí pozmenený zoznam vybraných slov

4.03 Prihlásenie

Identifikátor	UC-03	
Názov	Prihlásenie	
Opis	Po vyplnení údajov a potvrdení prihlási doteraz neprihláseného používateľa	
Vstupné podmienky	Používateľ nie je prihlásený	
Výstupné podmienky	Používateľ zadal správne meno a heslo	
Používatelia	Prihlásenie používajú žiaci a učitelia	
Základná postupnosť	Krok	Činnosť
	1	Neprihlásený používateľ vyplní meno a heslo

	2	Neprihlásený používateľ stlačí tlačidlo „prihlás“
	3	Systém porovná údaje
	4	Ak sa zhodujú, systém prihlási používateľa
Alternatívna postupnosť	4.a	Systém vypíše chybu prihlásenia („Zlé meno alebo heslo“)

4.04 Zobraz moje výsledky

Identifikátor	UC-04	
Názov	Zobraz moje výsledky	
Opis	Zobrazí výsledky práve prihláseného používateľa	
Vstup. podm.	Používateľ musí byť prihlásený	
Používatelia	Hlavne žiaci, poprípade učitelia	
Základná postupnosť	Krok	Činnosť
	1	Prihlásený používateľ stlačí tlačidlo Moje výsledky
	2	Systém zobrazí výsledky pre daného používateľa

4.05 Cvičenie

Identifikátor	UC-05	
Názov	Cvičenie	
Opis	Používateľ si vyberie a vypracuje niektoré z ponúkaných cvičení.	
Používatelia	Všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Používateľ stlačí tlačidlo „cvičenia“
	2	Systém používateľovi zobrazí zoznam ponúkaných cvičení
	3	Používateľ si vyberie jedno z ponúkaných cvičení
Alternatíva 1	4.a	UC-07
Alternatíva 2	4.b	UC-08
Alternatíva 3	4.c	UC-09
	5	Systém spracuje odpovede
	6	Systém zobrazí výsledky
Poznámky	Cvičenia sú určené aj pre návštevníkov, alebo na prezentáciu pred triedou	

4.06 Individuálne cvičenie

Identifikátor	UC-06	
Názov	Individuálne cvičenie	
Opis	Rozšírenie cvičenia pre samostatné hodnotenie s archiváciou výsledkov	
Vstup. podm.	Používateľ je prihlásený	
Výstup. podm.	Pripojenie na internet nebolo prerušené	
Používatelia	Používajú ho žiaci, prípadne učitelia	
Základná postupnosť	Krok	Činnosť
	1	UC-05
	2	Systém pošle výsledky vzdialenému serveru
	3	Server výsledky archivuje a doplní do výsledkov používateľa
Poznámky	Používa sa aj na precvičovanie, sledovanie zmien prospechu (ide mi to lepšie alebo horšie ako pred týždňom?), ale dá sa využiť aj na písomky a iné hodnotené skúšanie.	

4.07 Dopln správne písmeno

Identifikátor	UC-07	
Názov	Dopln správne písmeno	
Opis	Cvičenie na doplnenie správneho písmena do vybraného slova	
Používatelia	Používajú ho všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Systém zobrazí cvičenie (teda sadu slov s miestom, kde sa dá doplniť písmeno)
	2	Používateľ vyplní chýbajúce písmená
	3	Používateľ stlačí tlačidlo na potvrdenie, že je hotový
Alternatíva	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 2

4.08 Vyber zlé slovo

Identifikátor	UC-08	
Názov	Vyber zlé slovo	
Opis	Cvičenie na doplnenie výber zlého slova z pomedzi niekoľkých správnych	
Používatelia	Používajú ho všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Systém zobrazí niekoľko slov, z ktorých je práve jedno zlé
	2	Používateľ vyberie nesprávne slovo
	3	Používateľ stlačí tlačidlo na potvrdenie, že je hotový
Alternatíva	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 1

4.09 Prirad' slovo k obrázku

Identifikátor	UC-09	
Názov	Prirad' slovo k obrázku	
Opis	Cvičenie na priradenie slova k správne mu obrázku, napr. sa zobrazí slovo „výr“ a obrázky pre slová „vír“ a „výr“	
Používatelia	Používajú ho všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Systém zobrazí cvičenie (teda obrázky a slovo, ktoré ku niektorému z nich patrí)
	2	Používateľ vyberie právny obrázok
	3	Používateľ stlačí tlačidlo na potvrdenie, že je hotový
Alternatíva	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 2
Poznámky	V prípadoch chýbajúcich obrázkov sa načíta náhrada obrázkov	

4.10 Zobraz nápovedu

Identifikátor	UC-10	
Názov	Zobraz nápovedu	
Opis	Zobrazí nápovedu ohľadom práce s aplikáciou	
Používatelia	Používajú ho všetci používatelia	
Základná postupnosť	Krok	Činnosť
	1	Používateľ stlačí tlačidlo „nápoveda“
	2	Systém zobrazí nápovedu

5. Prílohy

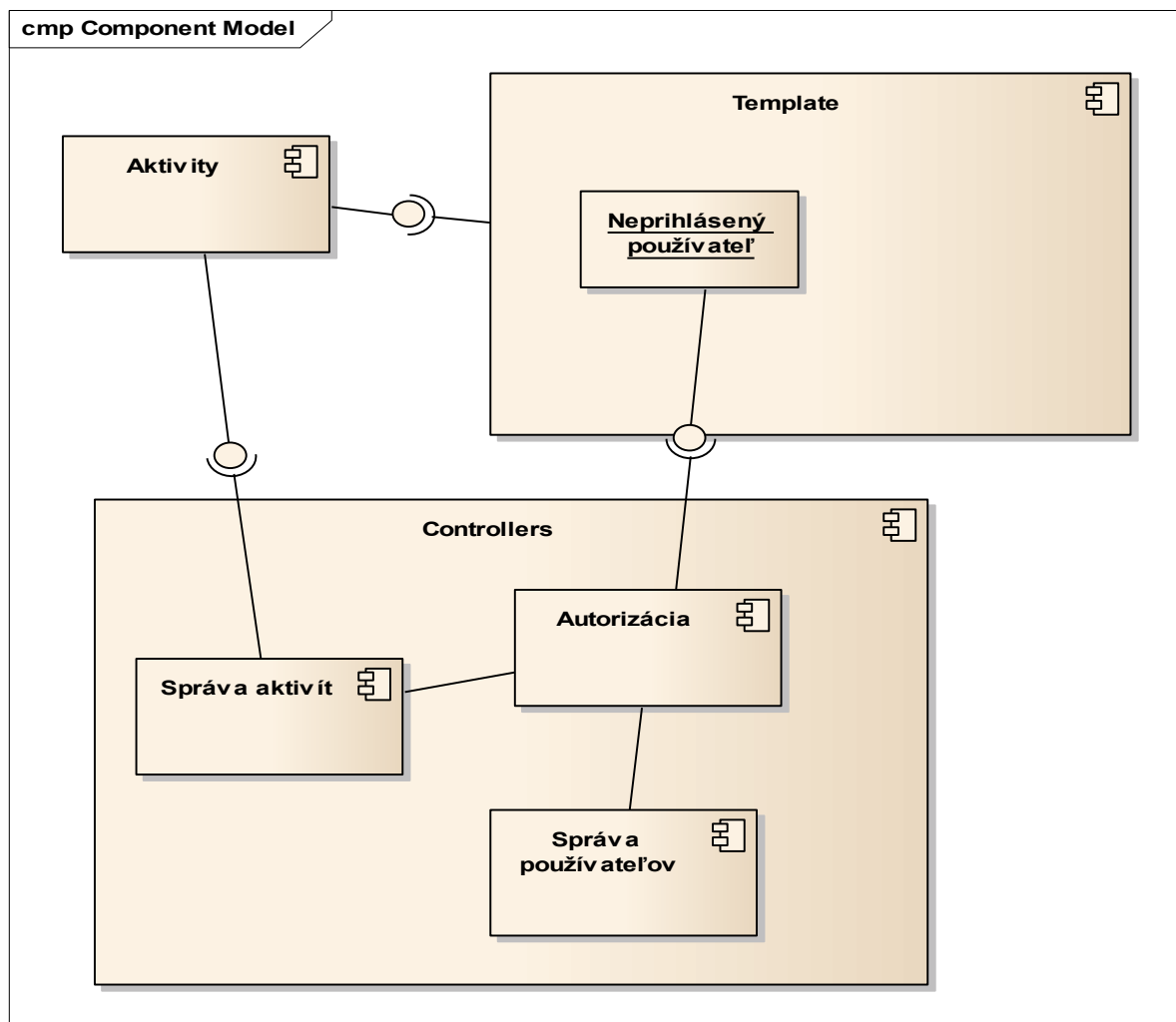
1. Súbor Pouzivatske_rozhrania.zip, ktorý obsahuje návrh grafického rozhrania a rozmiestnenia komponentov GUI.

5.1 Slovník pojmov

- 5.1.1 Používateľ – osoba využívajúca funkcie systému
- 5.1.2 žiak – prihlásená osoba bez práv na zásahy do systému, len jeho používanie s rozšírenými možnosťami
- 5.1.3 učiteľ - prihlásená osoba s právami na zmeny systému
- 5.1.4 cvičenie - aktivita, pri ktorej sa buď učí alebo skúša znalosť vybraných slov
- 5.1.5 individuálne cvičenie – cvičenie, ktorého výsledky sa archivujú, predpokladá sa že sa robí samostatne
- 5.1.6 GUI – grafické používateľské rozhranie

6. Komponenty

6.1 Komponentový diagram



6.2 Popis komponentov

6.2.1 Komponent Autorizácia

Pomocou tohto komponentu sa budú môcť používatelia správnym vyplnením prihlasovacieho formulára prihlásiť do systému. Komponent je používaný na prihlasovanie bežných už zaregistrovaných používateľov ako aj používateľov s administrátorskými právami. Z formulára sa načítajú údaje a porovnávajú sa so záznamami v databáze. Pri zhode sa používateľ prihlási do systému a podľa typu účtu má k dispozícii rôzne funkcionality systému.

6.2.2 Komponent Správa Používateľov

Komponent prístupný iba používateľom s administrátorskými právami. Slúži na správu registrovaných/neregistrovaných účtov. Funkcionalita:

- pridávanie používateľov
- mazanie používateľov
- modifikácia práv používateľov
-

6.2.3 Komponent Správa Aktivít

Komponent prístupný iba používateľom s administrátorskými právami. Slúži na správu Aktivít. Funkcionalita:

- pridávanie aktivít
- mazanie aktivít
- modifikácia aktivít

6.2.4 Komponent Aktivita

Tento komponent je prístupný všetkým používateľom a slúži na kontrolu vstupov pri práci s aktuálnou aktivitou.

6.2.5 Komponent Template

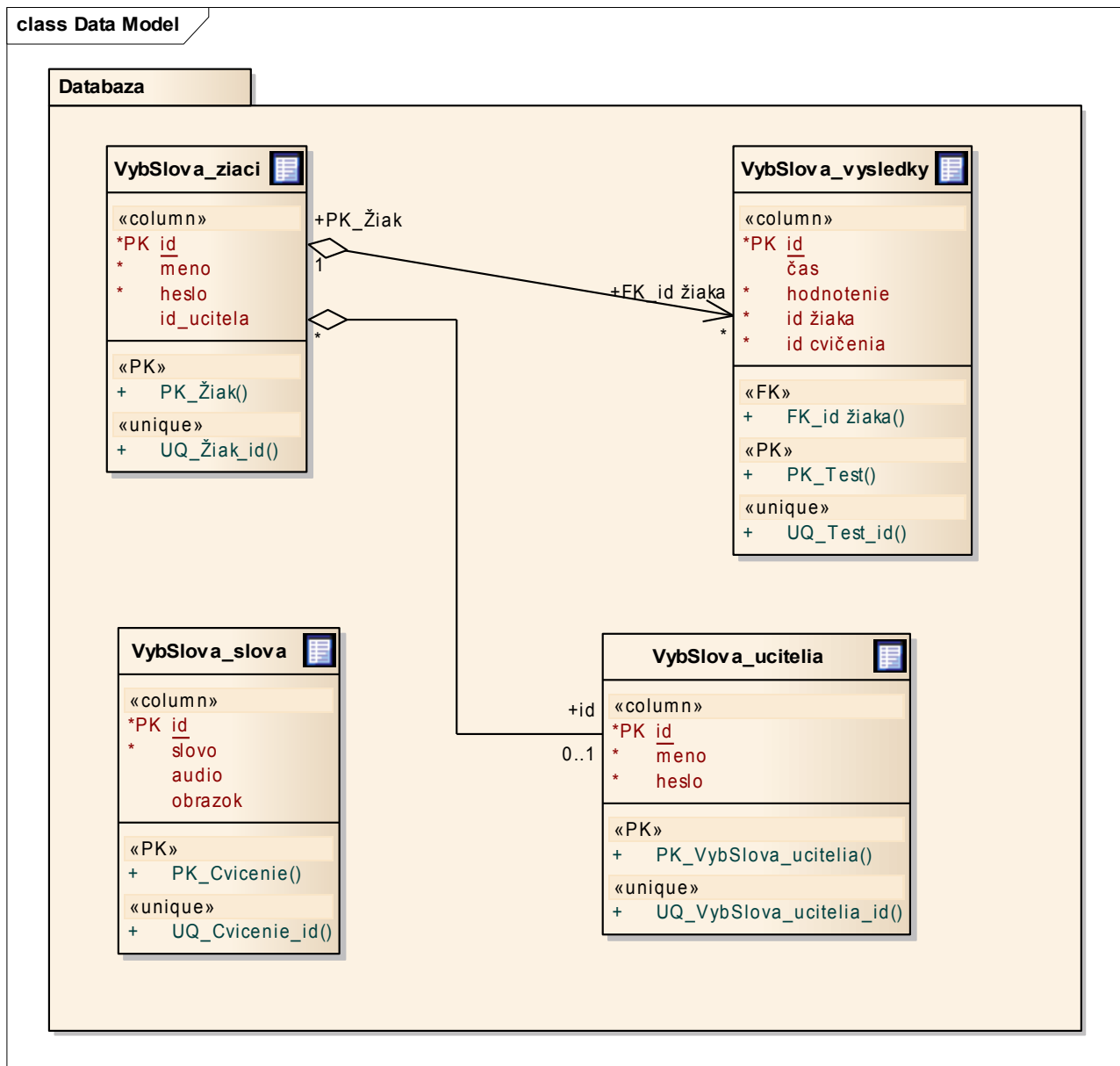
Tento komponent obsahuje cele GUI aplikácie. Spolupracuje so všetkými ostatnými komponentmi. Zabezpečuje zobrazenie obsahu.

6.2.6 Komponent Controllers

Komponent, ktorý obsahuje všetky kontroly vstupov.

7. Dátový model

7.1 Zobrazenie dátového modelu



7.2 Popis dátového modelu

7.2.1 VybSlova_slova

Tabuľka obsahuje údaje o každom slove v systéme. Slovo sa dá identifikovať podľa kľúča id typu int a obsahuje položky slovo typu string, čo je samotné slovo a odkazy na lokácie audio súboru kde sa nachádza prečítané slovo a odkaz na lokáciu obrázka.

7.2.2 VybSlova_ucitelia

Táto tabuľka obsahuje zoznam učiteľov, ktorý majú priradené id, ktoré je zároveň aj primárny kľúč, meno pod ktorým sa prihlasujú typu string a heslo.

7.2.3 VybSlova_vysledky

Tak isto ako ostatné tabuľky databázy obsahuje primárny kľúč id typu int, čas kedy bol test odovzdaný, výsledné hodnotenie testu, id cvičenia a id žiaka, ktorý cvičenie odovzdal.

7.2.4 VybSlova_ziaci

Pre každú položku, teda jedného žiaka, je v tabuľke uložené jeho id, ktoré je pre databázu primárny kľúč, prihlasovacie meno typu string, heslo a môže obsahovať id učiteľa, ku ktorému je žiak priradený.

7.2.5 Vzťahy medzi tabuľkami

Tabuľka žiaci sa viaže k tabuľke učitelia tak, že jeden žiak môže a nemusí byť priradený k učiteľovi a učiteľ môže mať ľubovoľný počet žiakov.

Tabuľka výsledky je viazaná na tabuľku žiakov tak, že každý výsledok(riadok tabuľky výsledky) je priradený práve jednému žiakovi, ale žiak samotný môže mať ľubovoľný počet výsledkov.

8. Analýza technológií

8.1 Node.js

Node.js je open-source prostredie pre používanie serverových webových aplikácií založených na Javascripte. V systéme bude použité na tvorbu back-endu.

8.2 HyperText Markup Language (HTML)

HTML je značkový jazyk, ktorý slúži ako štandard na tvorbu webových stránok. V systéme bude použitý ako základ pre zobrazenie používateľského rozhrania.

8.3 Javascript

Javascript je programovací jazyk využívaný pri tvorbe dynamického obsahu na webových stránkach. V kontexte systému bude slúžiť na tvorbu obsahu stránok a bude zabezpečovať reagovanie na vstup používateľa. Môže byť použitý buď priamo alebo cez framework.

8.4 Kaskádové štýly CSS

CSS slúži na upravenie výzoru grafického používateľského rozhrania stránky. V systéme bude použitý na tvorbu vzhľadu.

8.5 AngularJS

Je to Javascriptový framework určený na zjednodušenie práce s Javascriptom a na tvorbu jednostránkových webových aplikácií.

8.6 MongoDB

Je to open-source databázový systém, ktorý dobre spolupracuje s AngularJS a Node.js. V systéme bude slúžiť na správu databázy.

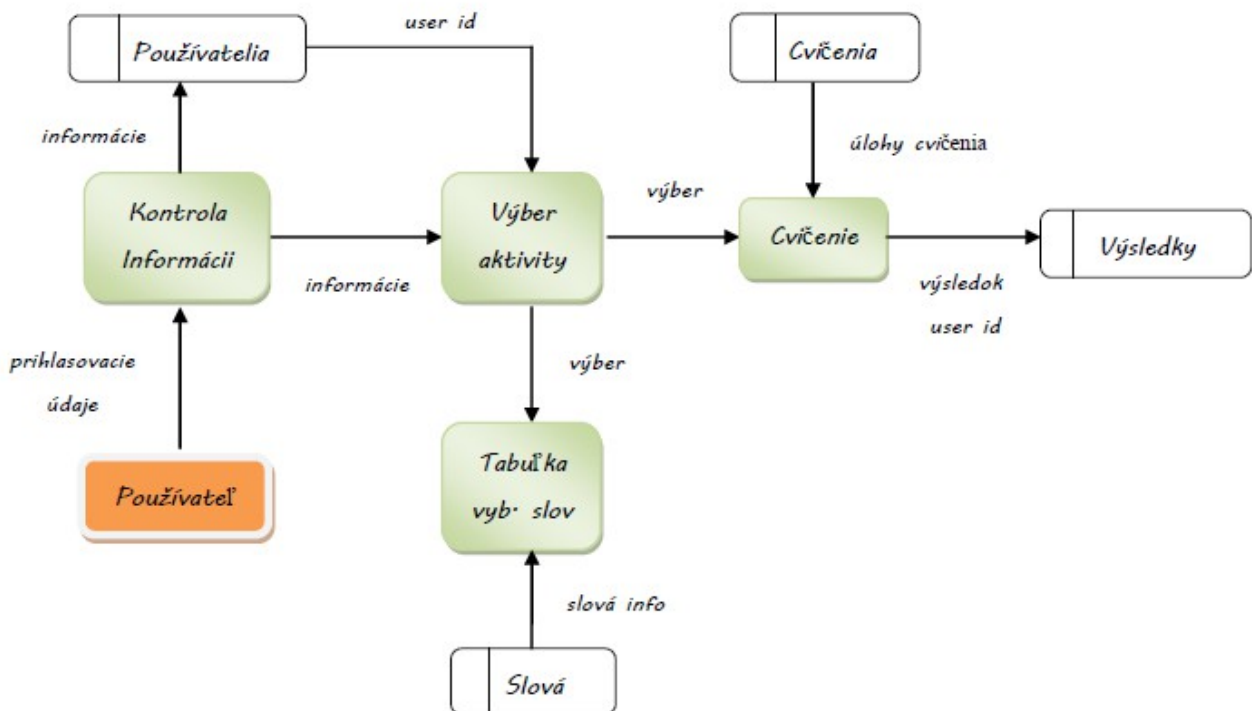
8.7 Web Audio API

Slúži ako aplikačné rozhranie pre Javascript za účelom tvorby a spracovávania zvukov. V systéme bude slúžiť hlavne na prehrávanie zvukov ktoré zodpovedajú slovám (inak povedané ako sa to slovo číta).

9. Objektový návrh

9.1 Data-flow diagram

Používateľ začína vyplnením prihlasovacieho formulára, ktorý sa skontroluje a databáza vráti id používateľa. Pokračuje výberom aktivity. Pri výbere vybraných slov databáza vráti tabuľku vybraných slov a proces ich zobrazí. Pri výbere konkrétneho cvičenia databáza vráti úlohy spojené s daným cvičením a proces spustí dané cvičenie. Nakoniec sa výsledok pošle do databázy spolu s používateľovým id číslom.



- Tabuľka v databáze - uchováva dáta vyžadované procesmi



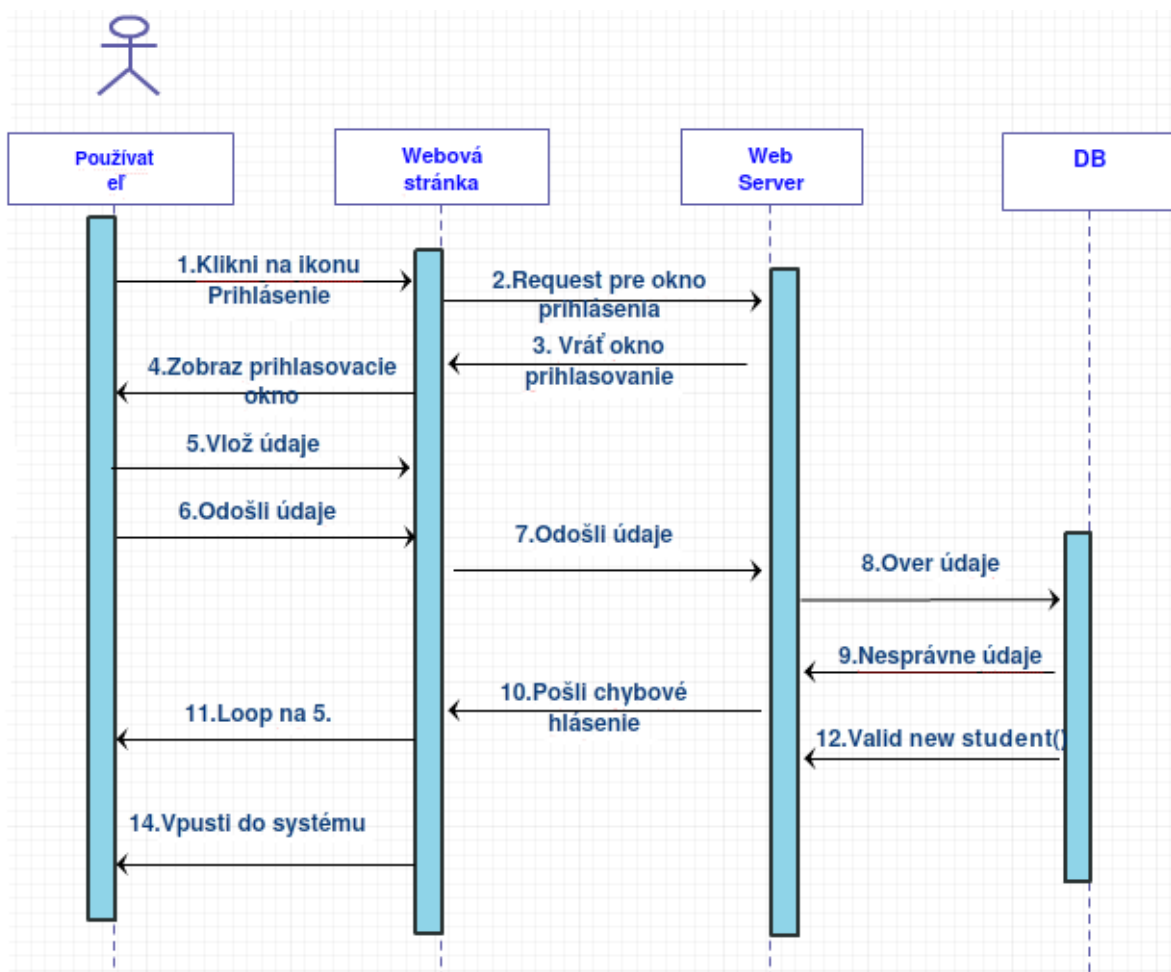
- Vonkajšia entita (človek, systém)



- Proces - funkcia ktorá manipuluje s dátami a transformuje ich

9.1 Sekvenčný diagram prihlasovania

Sekvenčný diagram na obrázku zobrazuje postup pri prihlasovaní používateľa pomocou vyplnenia prihlasovacieho formuláru. Údaje z neho sa kontrolujú s údajmi uloženými v databáze. Ak sú údaje správne, používateľ bude prihlásený. V prípade kedy používateľ zadá nesprávne údaje, systém vypíše chybovú správu a vráti používateľa späť ku prihlasovaciemu formuláru, kde môže prihlásenie opakovať.



9.3 Triedny návrh

9.3.1 Premenná cvičenie

Premenná cvičenie obsahuje údaje o cvičení, ktoré sú použité pri výpise pre používateľa. Popis a napoveda sú nepovinné položky, môžu byť prázdne. Zahŕňa:

- *id integer* - identifikácia cvičenia, primárny kľúč, hodnota 0-9
- *nazov string* - označuje názov cvičenia, má parametre unique a required
- *popis string* - je zobrazený ako popis cvičenia, ak by názov nestačil, je nepovinný

- *napoveda string* - obsahuje nápovedu na ovládanie cvičenia
- *ulohy uloha[]* - zoznam úloh použitých v danom cvičení

9.3.2 Premenná úloha

Premenná uloha obsahuje údaje pre jeden riadok alebo jednu časť cvičenia. Zahŕňa:

- *id integer* - má parameter min 10, max 999, t.j. má 989 vyhradených id
- *slova slovo[]* - je to množina slov využitá v úlohe, má minimálnu veľkosť 1
- *popis string* - ako vo cvičení je to popis úlohy, je nepovinný a slúži len ako vysvetlivka, ak ju treba
- *odpoved string[]* - odpoveď obsahuje správne riešenie úlohy, počet položiek je od 1 po počet slov

9.3.3 Premenná slovo

Premenná slovo obsahuje všetky údaje o 1 slove. Zahŕňa:

- *id integer* - má parameter min 1000
- *text string* - má parametre unique, trim a required (nesmie byť bez textu)
- *link na obrázok* - obsahuje odkaz na miesto, odkiaľ je obrázok pre slovo v tvare, aby sa dal priamo použiť na stránke
- *link na zvuk* - ako link na obrázok, ale pre zvukovú stopu

9.3.4 Funkcie

- *konštruktory pre premenné*

Pri pridávaní novej premennej vytvorí jednotný interface.

Každá má vlastný konštruktor, ktorý je obmedzený aj podľa databázy aj podľa programu (musí prejsť kontrolami pre obe v prípade slova). Konštruktor pre slovo je špecifický tým, že slová sa ukladajú, narozdiel od úloh, ktoré sú generované cez RNG podľa predlohy, a cvičení, ktoré sú okrem dĺžky pevne dané.

- *vypis ulohy(id_cvicenia integer, pocet integer)*

Vypíše zadaný počet úloh pre dané cvičenie.

Zavolá generátor náhodných čísel, podľa ktorých vygeneruje "pocet" náhodných čísel, až pokiaľ nemá dostatočný počet rôznych (duplikáty zahodí). Tieto hodnoty použije na výber úloh pre dané cvičenie. Pre niektoré cvičenia sa úlohy vypisujú postupne, nová po dokončení predošlej, pri iných sa zobrazia naraz pod seba. Konkrétna implementácia bude predbežne rozdelená, na konci testovania sa zvolí lepší spôsob pre každé cvičenie

- *skontroluj_ulohu(id_cvicenia, id, vysledok)*

Skontroluje výsledok úlohy podľa id úlohy a cvičenia.

Najskôr skontroluje hodnoty, či je id v danom cvičení, a ak áno či je správna dĺžka výsledku. Ak nie tak vráti považuje úlohy za nesplnené a výsledku priradí prázdnu alebo inú predpripravenú hodnotu. Následne porovná výsledok s výsledkom pre danú úlohu. Ak sa zhoduje, vráti informáciu o správnosti, inak o nesprávnom riešení.

- *vypis_ulohu(id integer)*

Vypíše úlohu s daným id, volá sa pri zavolaní funkcie vypis ulohy.

Pre cvičenie vyberie úlohu podľa zadaného id a vráti hodnoty pre stránku, aby ju mohla zobrazit' pre používateľa.

- *vyrob_cvicenie(id integer, pocet_uloh integer)*

Vráti údaje pre kompletne cvičenie s daným počtom úloh.

Cvičenia sú pevne nastavené, označujú rôzne druhy cvičení a vytvárame nové inštancie.

Najskôr vynuluje množinu úloh. Nasledovne pre daný počet úloh naplní zoznam cez volania vytvor_ulohu(id) nastaví názov, komentáre a pod. podľa predlohy a zavolá výpis úlohy, aby ju mohol používateľ riešiť.

- *vytvor_ulohu(id cvicenia)*

Vytvorí úlohu podľa vzoru pre dané cvičenie.

Slová pre danú úlohu sa vyberú náhodne bez opakovania cez RNG a načítajú do zoznamu.

Výsledok úlohy sa generuje podľa cvičenia, buď je daný podľa časti slova(1) alebo zlého slova(2).

(1) V slove nájde prvé y, ktoré nahradí prázdny miestom, a kontroluje sa s pôvodným slovom.

(2) Vypíše sa niekoľko slov, pričom niektoré nie je vybrané, id takéto slova je správne riešenie, alebo to slovo samotné.

(3) Pri priradovaní obrázku ku slovu sa kontroluje, či je link obrázka zhodný s linkom k obrázku pre dané slovo.

Nakoniec vráti úlohu ako výsledok.

- *nacitaj_slovo(id)*

Zavolá databázu a požiada údaje o slove.

Vstupným parametrom je id slova, ktoré funguje ako jeho primárny kľúč v databáze.

Podľa tohto id nájde slovo, u ktorého sa id zhoduje, ak také slovo neexistuje, vráti prázdne slovo(dá sa použiť na kontrolu, či slovo existuje/či je id voľné). Výsledok funkcie je premenná slovo, ktorú vráti.

- *autentifikuj(name string, password string)*

Ak je niečo napísané v premennej, kde má byť meno a heslo, tak ich overí podľa údajov v databáze. Ak sa zhodujú, vráti správu o úspešnom prihlásení a nastaví, že používateľ je

prihlásený, inak vráti chybovú hlášku (chýba meno/heslo, nesprávne heslo)

- *je_prihlaseny(req, res, next())*

Funkcia dostane premenné, ktoré identifikujú aktuálneho používateľa a zistí, či je prihlásený. Ak áno, zavolá nasledujúcu funkciu (napr. *ma_opravnenie*)

- *ma_opravnenie(req, res, next())*

Dostane ako parameter akciu, ktorú chce používateľ vykonať a jeho aktuálne údaje pod ktorými je prihlásený. Podľa týchto hodnôt buď vráti chybu o nedostatočných oprávneniach alebo akciu povolí.

- *pridaj_slovo(nove slovo)*

Skontroluje, či je používateľ prihlásený, a ak áno či má na túto akciu oprávnenie (t.j. či je učiteľ), inak akciu zamietne a vyvolá chybovú hlášku. Skontroluje, či je slovo v správnom formáte, či sú všetky údaje správneho typu. Ak slovo prejde cez testy, potom zavolá funkciu na uloženie slova do databázy

- *uloz_slovo(slovo)*

Server-side funkcia na ukladanie slov do databázy.

Ak je id nezadané alebo zlé (pod 1000), potom generuje nové id pred uložením, inak prepíše už existujúce id a považuje sa za modifikáciu už existujúceho slova. Uloží slovo do databázy s príslušným id.

9.4 Návrh kontrolerov

Kontrolery sú triedy, ktoré sú zodpovedné za obsluhu používateľských požiadaviek, vracajú používateľom odpovede vo forme html, xml a pod.

9.4.1 Kontroler student

Kontroler, ktorý je zodpovedný za zobrazenie žiaka a informácie o jeho hodnotení.

9.4.2 Kontroler teacher

Kontroler, ktorý je zodpovedný za zobrazenie učiteľa a informácie o jeho triede žiakov.

9.4.3 Kontroler frontend_kontroler

Kontroler pre frontend, teda časť stránky s ktorou pracuje bežný používateľ. Slúži na zavedenie všetkých spoločných knižníc pre frontend kontrolery.

9.4.4 Kontroler backend_kontroler

Kontroler pre backend, teda časť stránky s ktorou pracuje bežný používateľ. Slúži na zavedenie všetkých spoločných knižníc pre backend kontrolery.

9.4.5 Kontroler activities

Kontroler zobrazuje formulár s aktivitou pre žiaka alebo učiteľa a takisto vyhodnocuje odpovede na otázky. Zoznam akcií:

- index zobrazí usporiadanú množinu príkladov zodpovedajúcich danej aktivite a odpovede zodpovedajúce danej aktivite. Odpovede sa budú viazať na id žiaka
- verify dostane id odpovede a vyhodnotí správnosť/nesprávnosť

9.4.6 Kontroler admin

Úvodný kontroler pre administráciu aplikácie najmä využívanú učiteľmi a má tento zoznam akcií:

- index zistí, či je administrátor prihlásený alebo nie
- login zobrazí prihlasovaciu obrazovku
- do_login obdrží prihlasovacie údaje od akcie login a pokúsi sa prihlásiť administrátora

9.4.7 Kontroler admin_activity

Umožňuje spravovať záznamy o aktivitách a meniť ich obsah a dané zmeny ukladať. Má tieto akcie:

- index zobrazuje zoznam súčasných aktivít s možnosťami upravovať ich obsah
- new zobrazí formulár na doplnenie nového slova do aktivity
- delete zobrazí formulár na zmazanie slova z aktivity
- save umožní uložiť zmeny a vráti ich do tabuľky

9.4.8 Kontroler admin_students

Umožňuje spravovať záznam študentov nasledovne:

- index zobrazuje zoznam existujúcich študentov
- new zobrazí formulár na pridanie nového študenta do databázy
- delete zobrazí formulár na zmazanie študenta z databázy
- save umožní uložiť zmeny a vráti ich do tabuľky

9.5 MongoDB databáza a MongoJS

Ukážka práce v databáze so žiakmi – pseudokód.

- Vytvorenie databázy s priliehajúcimi kolekciami (tabuľkami) v MongoJS module pre Node.JS a zabezpečenie pripojenia.

```
var dburl = 'ipdatabazy/nazovdatabazy';
var collections = ['ziaci','slova','vysledky','ucitelia']
var mongojs = require('mongojs');
var db = mongojs(dburl, collections);
```

- Funkcia žiak sa dá chápať ako objekt predstavujúci žiaka s menom a heslom.

```
function žiak(name , password) {
    this.name = name
    this.password = password
}
```

- Pridanie unikátneho indexu k menu žiaka aby sa do databázy nedali pridávať viacerí žiaci s rovnakým menom.

```
db.users.ensureIndex( {meno:1}, {unique:true});
```

- Premenná žiak1 vytvára nového žiaka pomocou žiak funkcie.

```
var žiak1 = new žiak(jozko mrkva, hesloniake)
```

- Pridanie žiaka do databázy s výpisom úspešnosti. Vypíše chybu ak sa nepodarilo žiaka uložiť do databázy aj s príslušnou chybou.

```
db.ziaci.save(žiak1,function(err,savedUser)){
    if (err || !savedUser ) console.log("Používateľ " + žiak.name + " nebol uložený kvôli chybe " + err);
    else console.log("Používateľ " + savedUser.name + " uložený")
});
```

- Nájdenie žiaka v databáze s výpisom úspešnosti. Vypíše chybu ak sa nepodarilo žiaka nájsť v databáze aj s príslušnou chybou.

```
db.ziaci.find(ziak1,function(err,ziaci) {
    if (err || !ziaci.length) console.log("Používateľ " + ziak.name + " nebol
nájdený" );
    else ziaci.forEach(function(user) {
        console.log(("Používateľ nájdený = " + ziak.name);
    });
});
```

9.6 Testovacie scenáre

9.6.1 Načítanie slova

Pridanie slova s id 1000 do databázy, zavolá sa nacistaj_slovo(1000), skontroluje sa výsledok funkcie s pôvodným vzorom.

9.6.2 Nájdenie slova

Do databázy sa pridá niekoľko slov, zavolá sa nacistaj_slovo(id) pre náhodné z nich určené RNG, výsledok sa skontroluje so vzorom.

9.6.3 Kontrola kontroly úlohy

Vytvorí sa samostatná úloha, zadá sa do nej jednoduchý správny výsledok(napr. "a"), zavolá sa skontroluj_ulohu(id_cvicenia, "a"), odchyť sa výsledok(napr. na konzolu alebo do premennej). Ak je pravdivá, prvá časť testu prešla, tým istým spôsobom sa zavolá a skontroluje volanie skontroluj_ulohu(id_cvicenia, "b"). Ak vráti false, test prešiel kompletne

9.6.4 Tvorba cvičenia

Zavolá sa vyrob_cvicenie(id, 1)(id meníme podľa toho, ktoré cvičenie momentálne kontrolujeme). Vypíšeme údaje v cvičení a zistíme, či sú správne. Na dodatočnú kontrolu zavoláme skontroluj_ulohu(id, spravny_vysledok), aby sme zistili, či správne tvorí úlohy. Ak sme doteraz nenarazili na zlý údaj alebo chybovú hlášku, pokračujeme, inak test neprešiel. Použijeme vyrob_cvicenie(id, 4), zistíme, či má zoznam ulohy správnu dĺžku. Ak áno, skontrolujeme každú úlohu, či je správne riešiteľná a iná. Ak všetko prejde bez chyby, test je splnený.

9.6.5 Výpis úlohy

Vytvoríme si falošné cvičenie(nastavíme mu hodnoty napevno) a zavoláme výpis pre úlohu. Skontrolujeme údaje, či sa zhodujú s tými v cvičení. Ak áno, test prešiel.

9.6.6 Výpis cvičenia

Najskôr skúsime vypísať falošné cvičenie z predošlého testu s 1 úlohou. Skontrolujeme správnosť výpisu. Ak je správny, tak zavoláme vytvorenie a výpis pre viac úloh. Výsledky skontrolujeme, ak sa zhodujú s tými v cvičení, tak je test splnený.

9.5.7 Kontrola prihlasovania

Vytvoríme si falošnú databázu s 2 používateľmi(žiak a učiteľ). Skúsime sa prihlásiť so zlými údajmi, ak vypíše chybovú hlášku, pokračujeme. Skúsime sa prihlásiť s údajmi žiaka, pokračujeme, ak sa úspešne prihlásime. Skúsime zavolať funkciu určenú pre žiaka, napr. aby mu vrátil systém jeho výsledky. Ak ich vráti(aj ako prázdnu množinu), pokračujeme. Nasledovne skúsime zavolať funkciu, ku ktorej by nemal mať prístup (ako je vytvorenie nového slova). Skontrolujeme, či vráti chybovú hlášku. Tie isté testy ako pre používateľa skúsime aj pre učiteľa, ak nám povolí všetky úpravy/zobrazenia, tak je test splnený.