

**Fakulta matematiky, fyziky a informatiky**  
Univerzita Komenského v Bratislave

## **Špecifikácia požiadaviek na softvér**

Vybrané slová

Martin Masár  
Adrián Pavčo  
Peter Zach  
Michal Hradečný

Tvorba informačných systémov 2015/16.

# OBSAH

1 Špecifikácia požiadaviek.....	3
1.1 Úvod.....	4
1.2 Celkový prehľad systému.....	4
1.3 Komerčné potreby.....	4
1.4 Slovník pojmov.....	4
1.5 Referencie.....	4
1.6 Prehľad dokumentu.....	5
2 Celkový opis.....	6
2.1 Kontext systému.....	6
2.1.1 Používateľské rozhranie.....	6
2.1.2 Hardvérové rozhranie.....	6
2.1.3 Softvérové rozhranie.....	6
2.2 Funkcie systému.....	6
2.3 Charakteristika používateľov.....	7
2.4 Obmedzenia.....	7
2.5 Závislosti a predpoklady.....	7
3 Špecifikácia požiadaviek.....	8
3.1 REQ-F-01 - Jazyk.....	8
3.2 REQ-U-02 - Zobrazenie.....	8
3.3 REQ-F-03 - Kódovanie textu.....	8
3.5 REQ-U-05 - Jednoduchosť použitia.....	8
3.6 REQ-F-06 - Údaje.....	8
3.7 REQ-F-07 - Tlačidlá.....	8
3.8 REQ-F-08 - Režimy použitia.....	8
3.9 REQ-F-09 - Možnosť úpravy.....	9
3.10 REQ-F-10 - Rozličnosť cvičení.....	9
3.11 REQ-F-11 - Hodnotenie výsledkov.....	9
3.12 REQ-F-12 - Dedičnosť.....	9
3.13 REQ-F-13 - Zvuk.....	9
3.14 REQ-S-14 - Testovanie.....	9
4. Popis používateľských rozhraní.....	10
4.1 Neprihlásený používateľ.....	10
4.2 Prihlásený používateľ – žiak.....	10
4.3 Prihlásený používateľ – učiteľ.....	10
5. Entitno-relačný diagram.....	11

6. Stavový diagram.....	12
7. Use-case diagram.....	13
7.01 Pozri zoznam.....	14
7.02 Uprav zoznam.....	14
7.03 Prihlásenie.....	14
7.04 Zobraz moje výsledky.....	15
7.05 Cvičenie.....	15
7.06 Individuálne cvičenie.....	15
7.07 Dopln správne písmeno.....	16
7.08 Vyber zlé slovo.....	16
7.09 Prirad' slovo k obrázku.....	16
7.10 Zobraz nápovedu.....	17
8. Prílohy.....	18
8.1 Slovník pojmov.....	18
9. Komponenty.....	19
9.1 Komponentový diagram.....	19
9.2 Popis komponentov.....	19
9.2.1 Komponent Autorizácia.....	19
9.2.2 Komponent Správa Používateľov.....	20
9.2.3 Komponent Správa Aktivít.....	20
9.2.4 Komponent Aktivít.....	20
9.2.5 Komponent Template.....	20
9.2.6 Komponent Controllers.....	20
10. Dátový model.....	21
10.1 Zobrazenie dátového modelu.....	21
10.2 Popis dátového modelu.....	21
10.2.1 VybSlova_slova.....	21
10.2.2 VybSlova_ucitelia.....	22
10.2.3 VybSlova_vysledky.....	22
10.2.4 VybSlova_ziaci.....	22
10.2.5 Vzťahy medzi tabuľkami.....	22
11. Analýza technológií.....	23
11.1 Node.js.....	23
11.2 HyperText Markup Language (HTML).....	23
11.3 Javascript.....	23
11.4 Kaskádové štýly CSS.....	23
11.5 AngularJS.....	23
11.6 MongoDB.....	23

11.7 Web Audio API.....	23
12. Objektový návrh.....	24
12.1 Data-flow diagram.....	24
12.1 Sekvenčný diagram prihlasovania.....	25
12.3 Triedny návrh.....	25
12.3.1 Premenná cvičenie.....	25
12.3.2 Premenná úloha.....	26
12.3.3 Premenná slovo.....	26
12.3.4 Funkcie.....	26
12.4 Návrh kontrolerov.....	28
12.4.1 Kontroler student.....	28
12.4.2 Kontroler teacher.....	28
12.4.3 Kontroler frontend_kontroler.....	28
12.4.4 Kontroler backend_kontroler.....	29
12.4.5 Kontroler activities.....	29
12.4.6 Kontroler admin.....	29
12.4.7 Kontroler admin_activity.....	29
12.4.8 Kontroler admin_students.....	29
12.5 MongoDB databáza a MongoJS.....	30
13.6 Testovacie scenáre.....	31
13.6.1 Načítanie slova.....	31
13.6.2 Nájdenie slova.....	31
13.6.3 Kontrola kontroly úlohy.....	31
13.6.4 Tvorba cvičenia.....	31
13.6.5 Výpis úlohy.....	32
13.6.6 Výpis cvičenia.....	32
13.5.7 Kontrola prihlasovania.....	32

# 0. Úvod

## 0.1 Cieľ dokumentu

Dokument záverečná správa spája špecifikáciu požiadaviek, konceptuálnu analýzu, analýzu technológií, návrh a zhodnotenie.

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, testovacie scenáre, výsledky testovania a zhodnotenie projektu z rôznych pohľadov.

## 1 Špecifikácia požiadaviek

### 1.1 Úvod

Tento dokument popisuje používateľské, funkčné a parametrické požiadavky pre systém cvičení vybraných slov. Je určený pre zadávateľa, ktorému slúži ako kontrolný prostriedok, aby mal istotu, že jeho zadanie bude správne zrealizované a vývojovému tímu ako počiatočný bod procesu tvorby softvéru.

### 1.2 Celkový prehľad systému

Webová aplikácia Vybrané slová predstavuje online interaktívny výukový systém pre žiakov základných škôl. Jej cieľom je priniesť hravú a zároveň efektívnu formu výučby. Dostupná na internete, dovoľuje tak cvičenie v pohodlí domova, alebo v školskom prostredí. Aplikácia prináša pre deti zaujímavú audio-vizuálnu stránku na spríjemnenie učenia. Obsahuje niekoľko druhov skupinových, resp. individuálnych cvičení, databanku vybraných slov, taktiež rozhranie pre učiteľa.

### 1.3 Komerčné potreby

Jednou z najhlavnejších priorít výučby Slovenského jazyka na základných školách prvého stupňa je naučiť žiakov rozlišovať a vedieť sa správne rozhodnúť medzi jotou a ypsilonom. Desaťročia znamenal tento proces pre žiakov fádne naučenie sa množiny slov. Vďaka dostupnosti informačných technológií vznikla potreba vytvoriť didaktický software zaoberajúci sa týmto problémom. Riešením sa pokúša byť aj webová aplikácia Vybrané slová, zasadená do prostredia multimedálnej čítanky (1).

### 1.4 Slovník pojmov

didaktika - pedagogická veda zaoberajúca sa vyučovaním

databanka slov – množina nielen vybraných slov, ktorú aplikácia používa

databáza žiakov – zoznam žiakov v systéme multimedálnej čítanky

multimedálna čítanka – didaktická internetová stránka určená pre žiakov prvého stupňa ZŠ

javascript – skriptovací programovací jazyk na báze javy  
Node.js – serverové prostredie pre javascript (2)

## 1.5 Referencie

- (1) <https://www.mmcitanka.sk/> - internetová stránka obsahujúca systém, do ktorého bude projekt integrovaný
- (2) <https://nodejs.org/en/> - domovská internetová stránka Node.js

## 1.6 Prehľad dokumentu

Zvyšok špecifikácie je rozdelený do dvoch častí.

Prvá časť obsahuje celkový opis systému, prehľad jeho vlastností a ako pracuje z pohľadu bežného používateľa.

Druhá časť obsahuje zoznam jednotlivých podmienok a ich detailný popis.

## 2 Celkový opis

### 2.1 Kontext systému

Aplikácia predstavuje nový spôsob zoznámenia sa s vybranými slovami pre žiakov prvého stupňa základných škôl. S aplikáciou budú pracovať dve skupiny ľudí, a to žiaci a učitelia.

Z pohľadu žiaka aplikácia vybrané slová obsahuje niekoľko rozličných cvičení zameraných na výučbu vybraných slov. Medzi nich bude patriť priradenie slova ku obrázku, doplnenie správneho i/y do slova.

Z pohľadu učiteľa táto aplikácia môže byť využívaná na spoločné vyučovanie na jednom počítači, alebo na samostatné učenie sa na jednotlivých počítačoch.

Aplikácia bude obsahovať databázu žiakov, kde bude možnosť hodnotiť ich pri jednotlivých aktivitách.

#### 2.1.1 Používateľské rozhranie

Používateľské rozhranie musí spĺňať požiadavky stanovené zadávateľom projektu nakoľko aplikácia bude integrovaná do už existujúceho webového rozhrania.

Používateľské rozhranie musí byť vytvorené formou web aplikácie.

Používateľské rozhranie musí byť dostatočne jednoduché a intuitívne pre používateľov.

#### 2.1.2 Hardvérové rozhranie

Systém neobsahuje žiadne hardvérové rozhranie, kvôli jeho integrácii do už existujúceho systému, ktorý toto rozhranie zabezpečuje.

#### 2.1.3 Softvérové rozhranie

Aplikácia bude vytvorená ako web aplikácia pomocou javascriptu a Node.js na zjednodušenie integrácie do už existujúceho systému.

Používateľské rozhranie musí správne fungovať v prehliadačoch Internet Explorer, Firefox, Google Chrome a Opera.

### 2.2 Funkcie systému

Systém musí byť schopný vytvoriť, načítať, uložiť a manipulovať s dátovou štruktúrou obsahujúcou slová – databanku, ku ktorým sú viazané zvuky a/alebo obrázky. Manipulácia s obsahom databanky je možná len pri prihlásení sa pod učiteľským kontom. Obsah databanky je zoznam slov, ktoré sú používané v cvičeniach. Cvičenia sú schopné pracovať v dvoch režimoch - osobný na individuálne precvičovanie a skupinový pre vyučujúceho a publikum/žiacov.

Keďže sa dáta v systéme spravujú cez samotný systém, musí byť schopný odfiltrovať zlé vstupy bez pádu celého systému.



## 2.3 Charakteristika používateľov

Používateľov tejto aplikácie budeme rozdeľovať do dvoch kategórií a to: žiakov a učiteľov.

-žiaci:

budú najmä prvého stupňa základných škôl, tj. deti vo veku 6-11 rokov. Predpokladá sa, že k systému budú žiaci najmä pristupovať zo školského prostredia no nakoľko sa jedná o web aplikáciu budú k nej mať prístup aj z domu

-učitelia:

okrem všeobecnej funkčnosti aplikácie, učitelia budú mať k dispozícii aj systém hodnotenia jednotlivých žiakov pomocou bodov, ktoré sa budú ukladať do databázy k jednotlivým hodnoteným žiakom.

## 2.4 Obmedzenia

Systém musí pracovať v rámci jedného DIV elementu na stránke multimedialnej čítanky(1) , kde bude zaradený ako javascriptový objekt. Musí byť schopný pracovať v rámci danej stránky bez narušenia jej chodu.

Ďalšia nutnosť je grafické používateľské rozhranie dostatočne jednoduché, aby s ním vedeli pracovať žiaci prvého stupňa základnej školy, ktorý sú určený ako jedna z cieľových skupín používateľov.

Systém taktiež musí byť spustiteľný na bežnom školskom počítači bez nadmerného spomalenia hardvéru.

Funkcia zmeny údajov v dátovej štruktúre programu musí byť implementovaná tak, aby s ňou vedel pracovať učiteľ na základnej škole bez informatického vzdelania.

Kód systému musí byť napísaný tak, aby sa s ním dalo manipulovať a robiť na ňom úpravy aj pre programátora, ktorý sa nepodieľal na tvorbe systému (kód nesmie byť "čierna skrinka").

## 2.5 Závislosti a predpoklady

Jedinou závislosťou aplikácie je už existujúca elektronická čítanka, do ktorej sa naša aplikácia bude vkladať a je potrebné dodržať kompatibilitu týchto dvoch aplikácií.

Ďalej sa predpokladá, že systém bude hlavne používaný na školských počítačoch.

## 3 Špecifikácia požiadaviek

### 3.1 REQ-F-01 - Jazyk

Projekt má byť napísaný v jazyku javascript a používať prostredie Node.js.

### 3.2 REQ-U-02 - Zobrazenie

Kompletné GUI je navrhnuté tak, aby sa dalo použiť v jednom DIV elemente HTML stránky.

### 3.3 REQ-F-03 - Kódovanie textu

Na kódovanie jazyka je použitá znaková sada UTF-8.

### 3.5 REQ-U-05 - Jednoduchosť použitia

Program obsahuje intuitívne, prehľadné a ľahko použiteľné grafické používateľské prostredie.

### 3.6 REQ-F-06 - Údaje

Má obsahovať databanku slov, ku ktorým môžu byť priradené obrázky.

### 3.7 REQ-F-07 - Tlačidlá

Aplikácia poskytuje panel tlačidiel, a to nasledovné:

1. Úvod – po stlačení sa zobrazí hlavná úvodná stránka
2. Cvičenia – po stlačení sa zobrazí ponuka cvičení
3. Slovíčka – po stlačení sa zobrazí zoznam vybraných slov, učiteľskému kontu sa tu zobrazí tlačidlo Úprava, ktoré otvorí rozhranie na manipuláciu so slovami v databanke
4. Nápoveda – po stlačení sa zobrazí nápoveda

### 3.8 REQ-F-08 - Režimy použitia

Program má nasledujúce módy použitia:

1. Učiteľ, ktorý má možnosť meniť údaje v databanke
2. Skupinové cvičenie (s použitím projektora)
3. Individuálne cvičenie

### 3.9 REQ-F-09 - Možnosť úpravy

S údajmi v databanke sa dá manipulovať cez rozhranie v programe, pri prihlásení pod učiteľským kontom.

Príklad scenára:

1. Učiteľ je prihlásený – systém po stlačení tlačidla Slovíčka a následne Úprava dovoľuje vykonávať zmeny v databanke slov
2. Neprihlásený používateľ/žiak – systém nezobrazuje tlačidlo Úprava a tým zabraňuje neoprávnenej zmene.

### 3.10 REQ-F-10 - Rozličnosť cvičení

Aplikácia obsahuje niekoľko druhov cvičení na tréning vybraných slov.

Príklad scenára:

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

### 3.11 REQ-F-11 - Hodnotenie výsledkov

Systém si pamätá počet správnych, resp. nesprávnych odpovedí pri cvičeniach, na konci ktorých tento počet oznámi a podľa výsledku používateľa slovne aj zvukovo ohodnotí.

### 3.12 REQ-F-12 - Dedičnosť

Program má byť navrhnutý tak, aby bola vytvorená jedna hlavná trieda, z ktorej sú odvodené jednotlivé cvičenia.

### 3.13 REQ-F-13 - Zvuk

Audio komponent je riešený cez web audio API.

### 3.14 REQ-S-14 - Testovanie

Použiteľná verzia bude otestovaná aj v školskom prostredí.

## 4. Popis používateľských rozhraní

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

### 4.1 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.

### 4.2 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.

### 4.3 Prihlásený používateľ – učiteľ

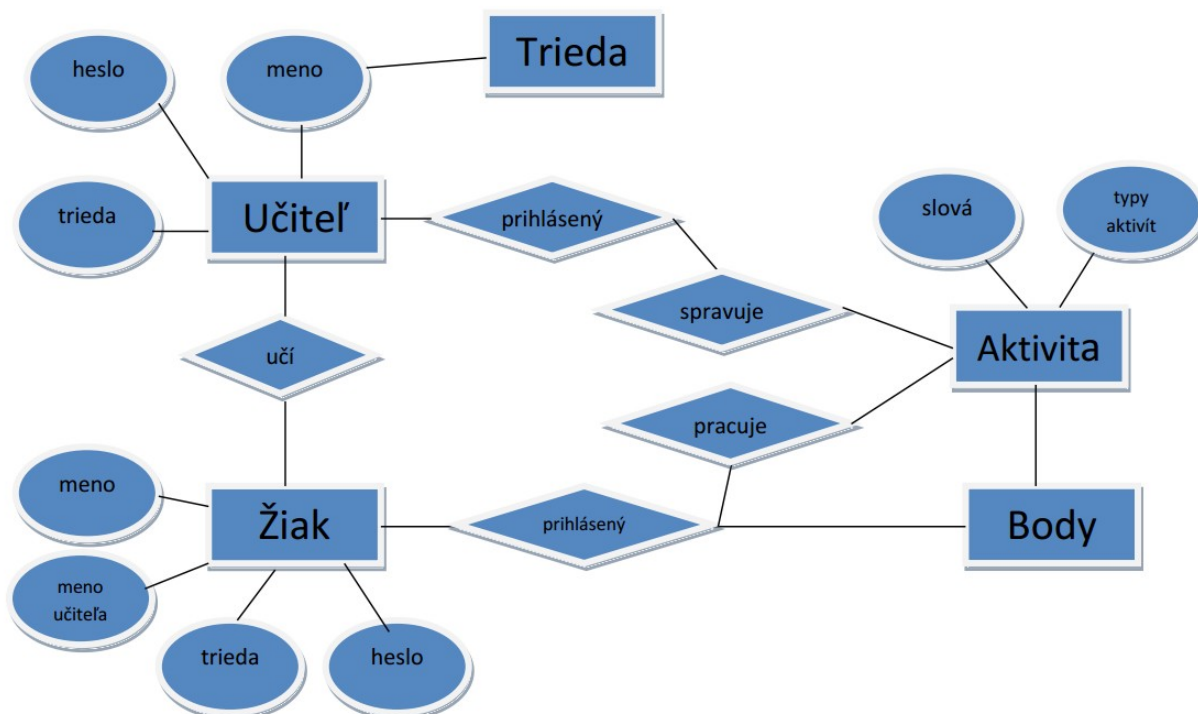
Poskytuje to isté rozhranie ako pre prihláseného žiaka, s niekoľkými zmenami:

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.

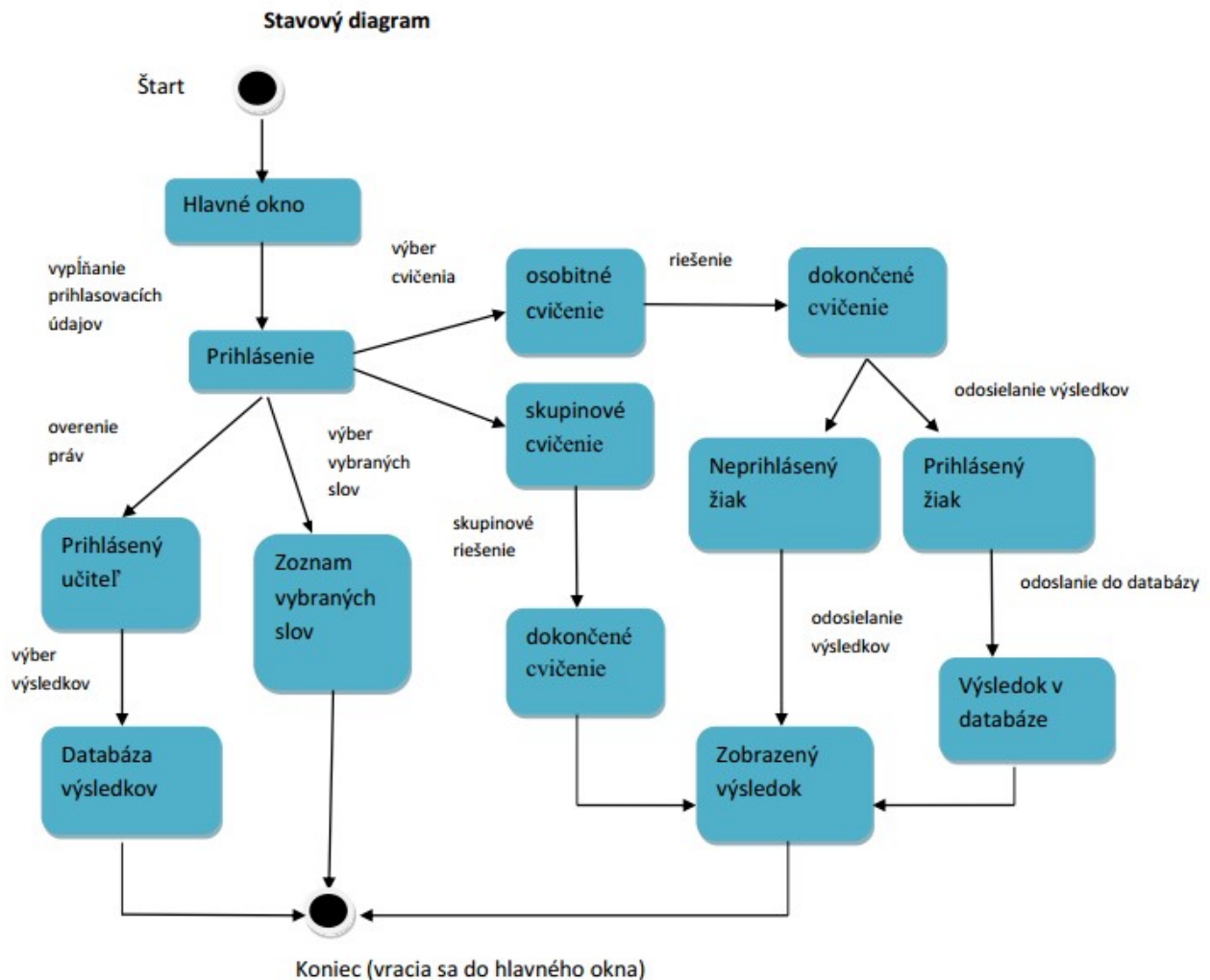
## 5. Entitno-relačný diagram

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.



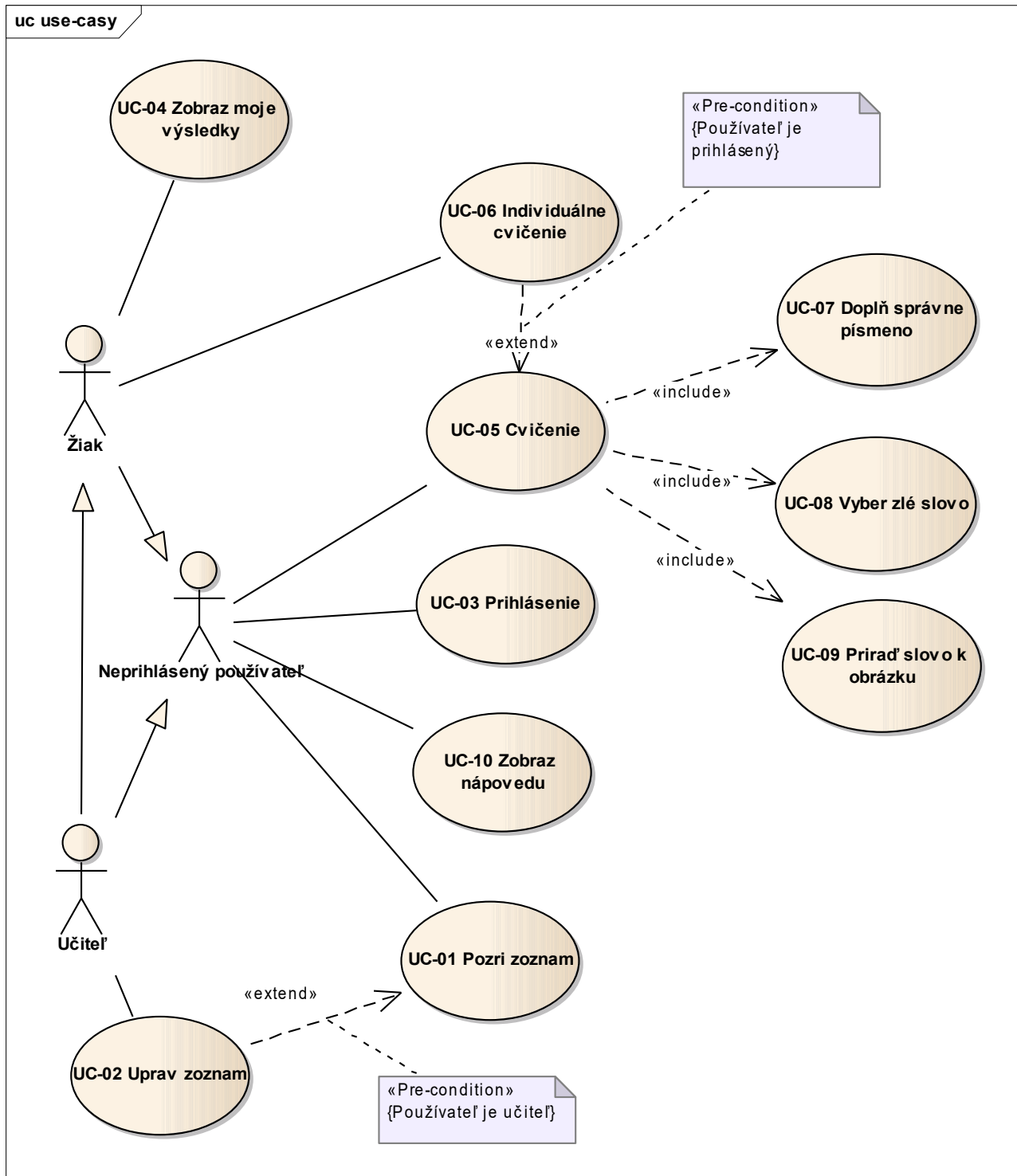
## 6. Stavový diagram

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.



## 7. 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.



## 7.01 Pozri zoznam

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

## 7.02 Uprav zoznam

<b>Identifikátor</b>	UC-02	
<b>Názov</b>	Uprav zoznam	
<b>Opis</b>	Učiteľ upraví zoznam vybraných slov	
<b>Vstup. podm.</b>	Používateľ musí byť prihlásený ako učiteľ	
<b>Výstup. podm.</b>	Musia byť potvrdené zmeny, inak sa nevykonajú	
<b>Používatelia</b>	Kto ho používa	
<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	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

## 7.03 Prihlásenie

<b>Identifikátor</b>	UC-03	
<b>Názov</b>	Prihlásenie	
<b>Opis</b>	Po vyplnení údajov a potvrdení prihlási doteraz neprihláseného používateľa	
<b>Vstupné podmienky</b>	Používateľ nie je prihlásený	
<b>Výstupné podmienky</b>	Používateľ zadal správne meno a heslo	
<b>Používatelia</b>	Prihlásenie používajú žiaci a učitelia	
<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	1	Neprihlásený používateľ vyplní meno a heslo
	2	Neprihlásený používateľ stlačí tlačidlo „prihlás“
	3	Systém porovná údaje
	4	Ak sa zhodujú, systém prihlási používateľa
<b>Alternatívna postupnosť</b>	4.a	Systém vypíše chybu prihlásenia („Zlé meno alebo heslo“)



## 7.04 Zobraz moje výsledky

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

## 7.05 Cvičenie

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

## 7.06 Individuálne cvičenie

<b>Identifikátor</b>	UC-06	
<b>Názov</b>	Individuálne cvičenie	
<b>Opis</b>	Rozšírenie cvičenia pre samostatné hodnotenie s archiváciou výsledkov	
<b>Vstup. podm.</b>	Používateľ je prihlásený	
<b>Výstup.</b>	Pripojenie na internet nebolo prerušené	

<b>podm.</b>		
<b>Používatelia</b>	Používajú ho žiaci, prípadne učitelia	
<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	1	UC-05
	2	Systém pošle výsledky vzdialenému serveru
	3	Server výsledky archivuje a doplní do výsledkov používateľa
<b>Poznámky</b>	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.	

## 7.07 Dopln správne písmeno

<b>Identifikátor</b>	UC-07	
<b>Názov</b>	Dopln správne písmeno	
<b>Opis</b>	Cvičenie na doplnenie správneho písmena do vybraného slova	
<b>Používatelia</b>	Používajú ho všetci používatelia	
<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	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
<b>Alternatíva</b>	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 2

## 7.08 Vyber zlé slovo

<b>Identifikátor</b>	UC-08	
<b>Názov</b>	Vyber zlé slovo	
<b>Opis</b>	Cvičenie na doplnenie výber zlého slova z pomedzi niekoľkých správnych	
<b>Používatelia</b>	Používajú ho všetci používatelia	
<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	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
<b>Alternatíva</b>	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 1

## 7.09 Prirad slovo k obrázku

<b>Identifikátor</b>	UC-09	
<b>Názov</b>	Prirad slovo k obrázku	
<b>Opis</b>	Cvičenie na priradenie slova k správnej obrázku, napr. sa zobrazí slovo „výr“ a obrázky pre slová „vír“ a „výr“	
<b>Používatelia</b>	Používajú ho všetci používatelia	

<b>Základná postupnosť</b>	<b>Krok</b>	<b>Činnosť</b>
	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
<b>Alternatíva</b>	4.a	Systém zobrazí pokračovanie cvičenia a vráti sa na krok 2
<b>Poznámky</b>		V prípadoch chýbajúcich obrázkov sa načíta náhrada obrázkov

## 7.10 Zobraz nápovedu

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

## 8. Prílohy

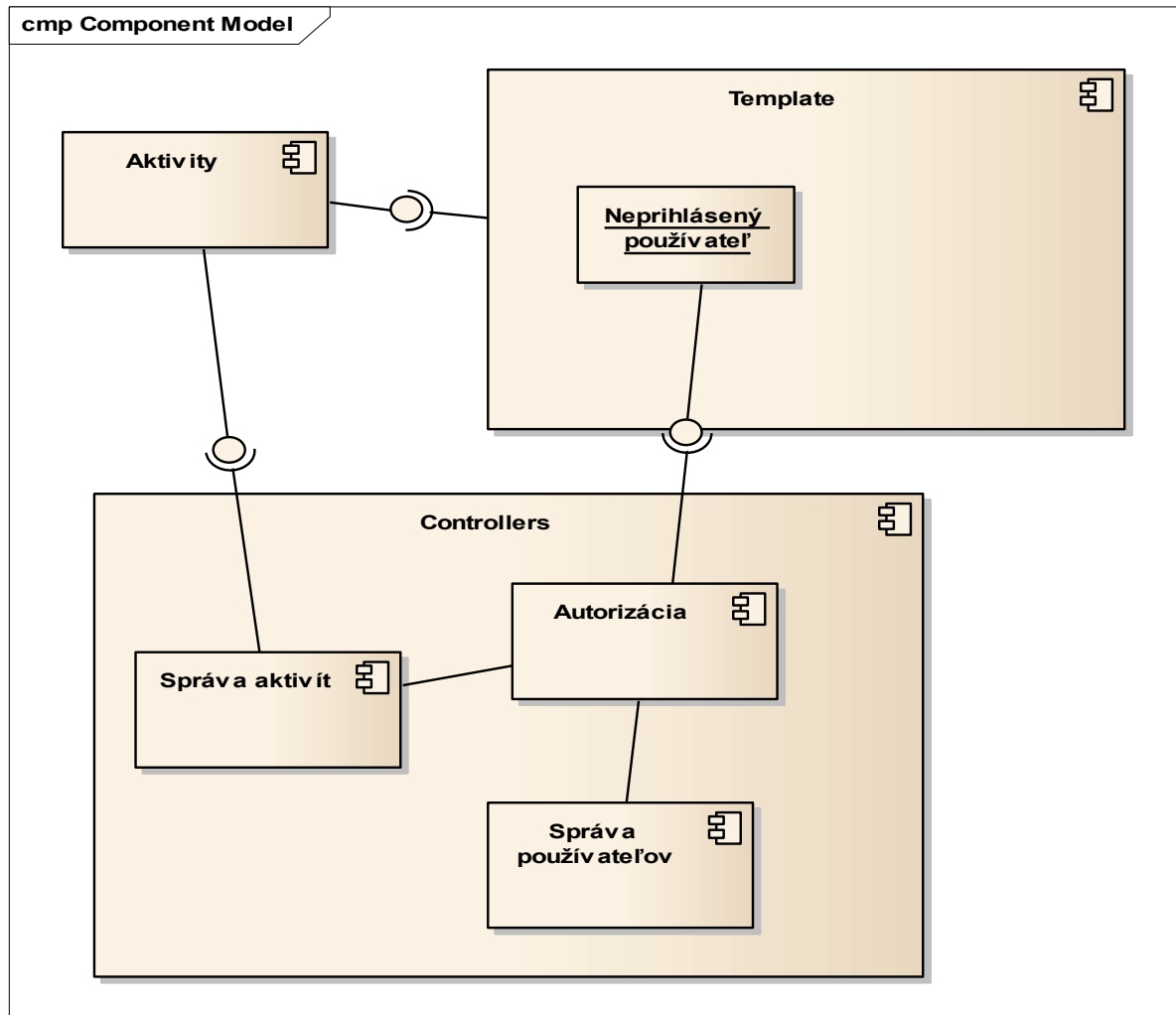
1. Súbor Pouzivatske\_rozhrania.zip, ktorý obsahuje návrh grafického rozhrania a rozmiestnenia komponentov GUI.

### 8.1 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

## 9. Komponenty

### 9.1 Komponentový diagram



### 9.2 Popis komponentov

#### 9.2.1 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.

### 9.2.2 Komponent Správa Používateľov

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

3. pridávanie používateľov
4. mazanie používateľov
5. modifikácia práv používateľov
- 6.

### 9.2.3 Komponent Správa Aktivít

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

- (3) pridávanie aktivít
- (4) mazanie aktivít
- (5) modifikácia aktivít

### 9.2.4 Komponent Aktivita

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

### 9.2.5 Komponent Template

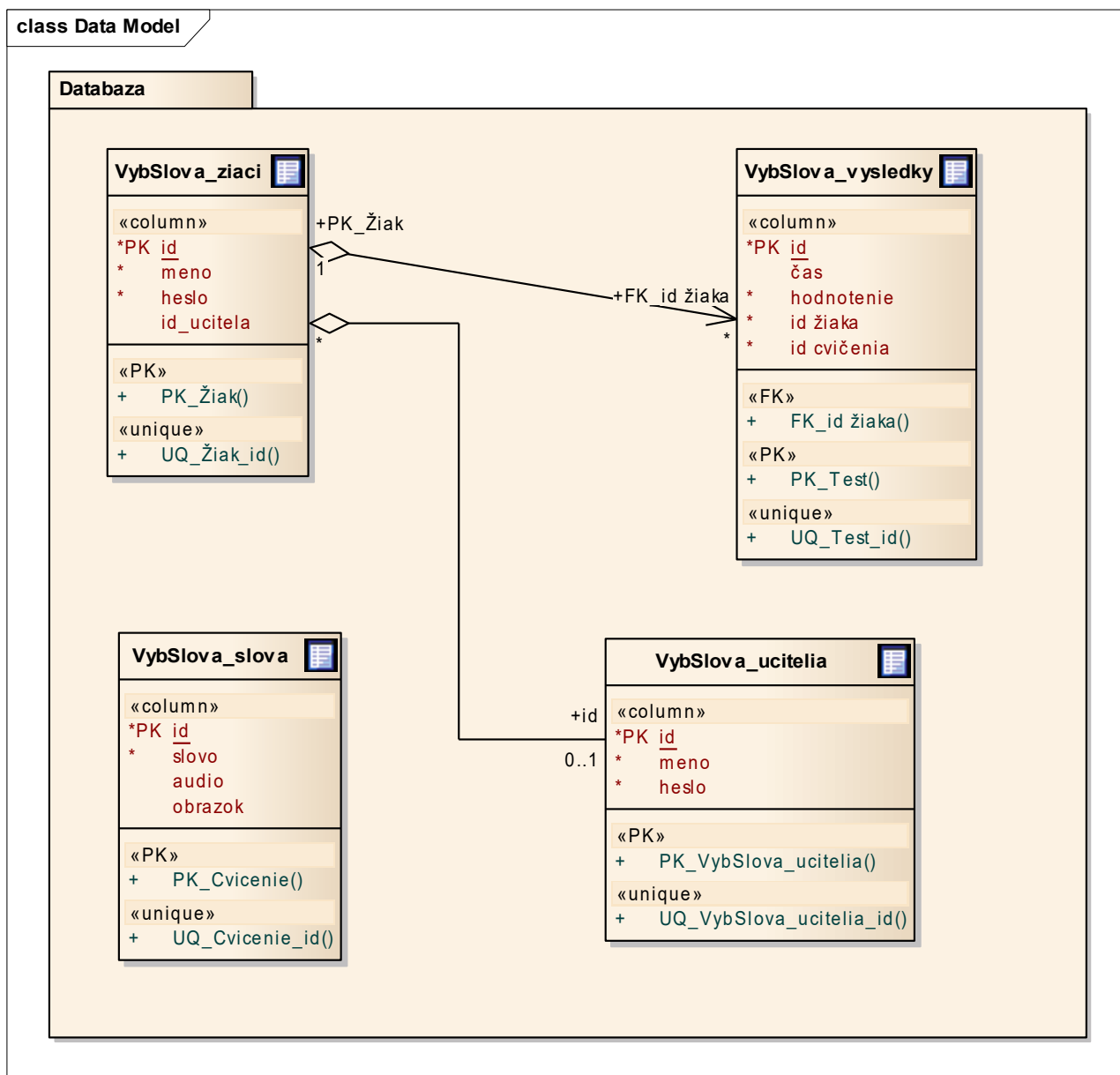
Tento komponent obsahuje cele GUI aplikácie. Spolupracuje so všetkými ostatnými komponentmi. Zabezpečuje zobrazenie obsahu.

### 9.2.6 Komponent Controllers

Komponent, ktorý obsahuje všetky kontroly vstupov.

# 10. Dátový model

## 10.1 Zobrazenie dátového modelu



## 10.2 Popis dátového modelu

### 10.2.1 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.

### 10.2.2 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.

### 10.2.3 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.

### 10.2.4 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.

### 10.2.5 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.



## **11. Analýza technológií**

### **11.1 Node.js**

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

### **11.2 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.

### **11.3 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.

### **11.4 Kaskádové štýly CSS**

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

### **11.5 AngularJS**

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

### **11.6 MongoDB**

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

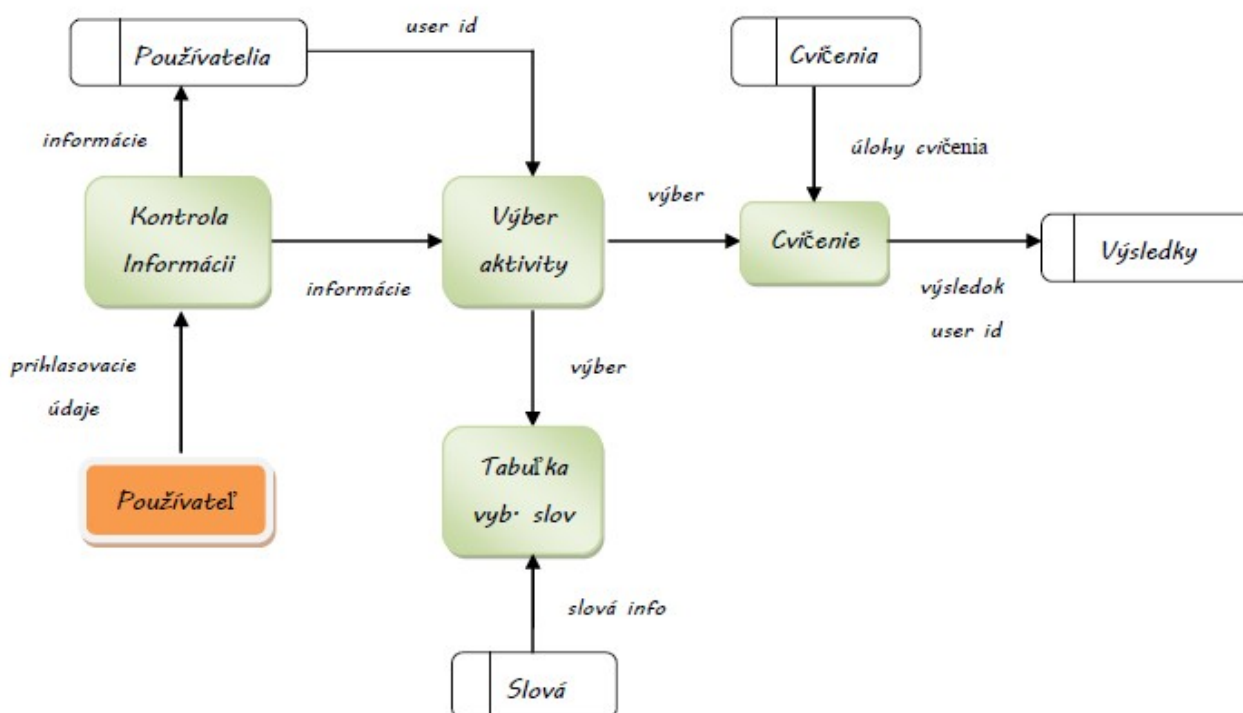
### **11.7 Web Audio API**

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).

## 12. Objektový návrh

### 12.1 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



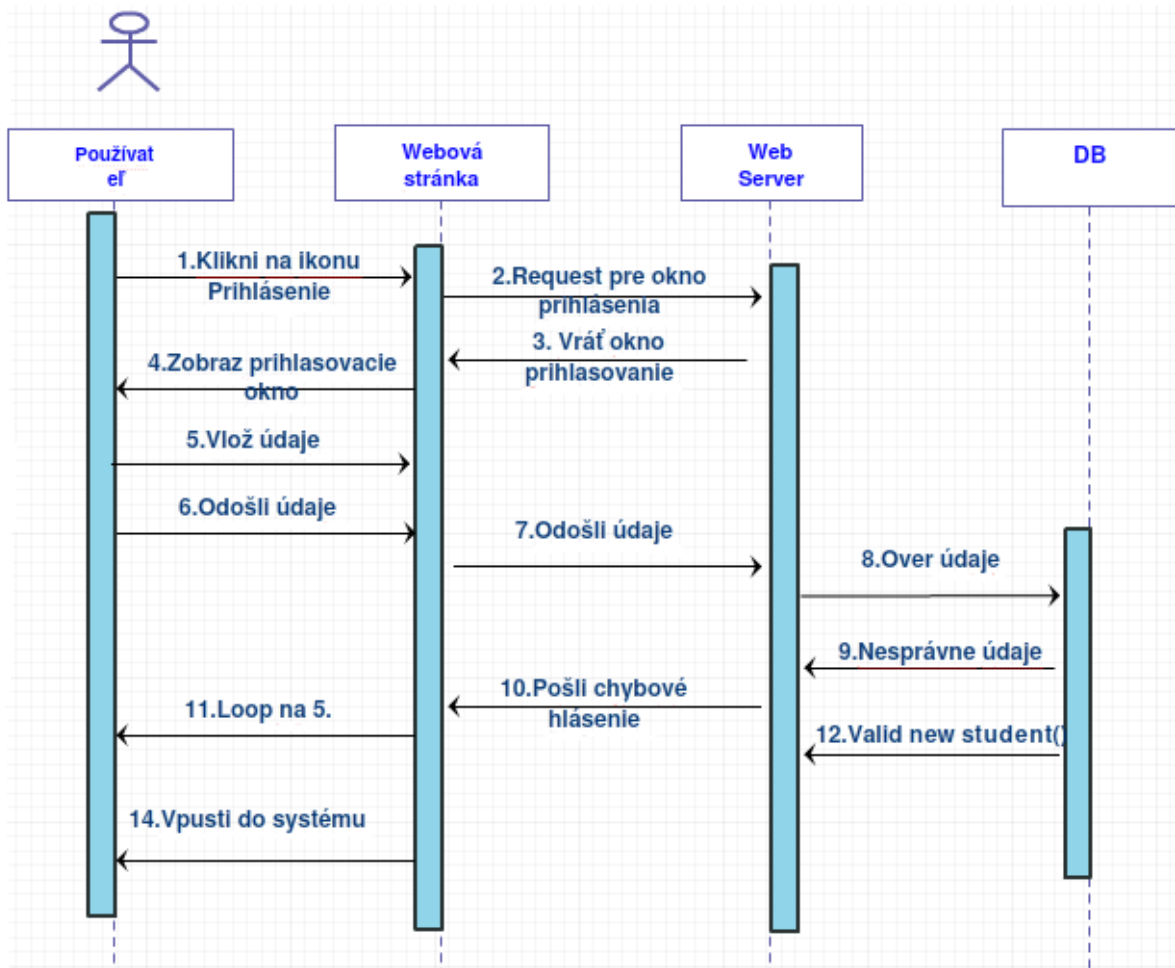
- Vonkajšia entita (človek, systém)



- Proces - funkcia ktorá manipuluje s dátami a transformuje ich

## 12.1 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.



## 12.3 Triedny návrh

### 12.3.1 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:

7. *id integer* - identifikácia cvičenia, primárny kľúč, hodnota 0-9
8. *nazov string* - označuje názov cvičenia, má parametre unique a required
9. *popis string* - je zobrazený ako popis cvičenia, ak by názov nestačil, je nepovinný
10. *napoveda string* - obsahuje napovedu na ovládanie cvičenia

## 11. *ulohy uloha[]* - zoznam úloh použitých v danom cvičení

### 12.3.2 Premenná úloha

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

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

### 12.3.3 Premenná slovo

Premenná slovo obsahuje všetky údaje o 1 slove. Zahŕňa:

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

### 12.3.4 Funkcie

- (14) *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.

- (15) *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

- (16) *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.

(17) *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.

(18) *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.

(19) *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.

(20) *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.

(21) *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 (chyba meno/heslo, nesprávne heslo)

(22) *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`)

(23) `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.

(24) `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

(25) `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.

## 12.4 Návrh kontrolerov

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

### 12.4.1 Kontroler student

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

### 12.4.2 Kontroler teacher

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

### 12.4.3 Kontroler frontend\_kontroler

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

#### 12.4.4 Kontroler backend\_kontroler

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

#### 12.4.5 Kontroler activities

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

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

#### 12.4.6 Kontroler admin

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

1. index zistí, či je administrátor prihlásený alebo nie
2. login zobrazí prihlasovaciu obrazovku
3. do\_login obdrží prihlasovacie údaje od akcie login a pokúsi sa prihlásiť administrátora

#### 12.4.7 Kontroler admin\_activity

Umožňuje spravovať záznamy o aktivitách a meniť ich obsah a dané zmeny ukladať

Má tieto akcie:

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

#### 12.4.8 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

## 12.5 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);
  });
});

```

## 13.6 Testovacie scenáre

### 13.6.1 Načítanie slova

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

### 13.6.2 Nájdenie slova

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

### 13.6.3 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

### 13.6.4 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.

### 13.6.5 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.

### 13.6.6 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.

### 13.5.7 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.

## 14. Výsledky testovania

### 14.1 Testovanie a validácia

Testovanie prebiehalo nepretržite počas vývoja. Kvôli rozdielom v návrhu a realizácii nebolo možné otestovať fungovanie systému podľa všetkých testovacích scenárov. Testovalo sa náhodné vyberanie slov z asociatívneho poľa, výber a vypisovanie cvičení, kontrolovanie cvičení a kontrolovanie výsledkov cvičení. Správnosť testov sa určovala prevažne testovacím výpisom hodnôt do konzoly a interakciou používateľa s aplikáciou. S finalizáciou produktu boli jednotlivé verzie postupne posielané zadávateľovi a na základe jeho odozvy a pripomienok aj aktualizované, či už z funkčnej, alebo vizuálnej stránky.

## 15. Zhodnotenie diela

### 15.1 Úvod

Tento dokument zhrňuje prácu a výsledky práce projektového tímu, zodpovedá ich v niekoľkých kapitolách venovaných zhodnoteniu diela, odchýlok od pôvodného plánu a zhodnotenia práce členov tímu.

### 15.2 Zhodnotenie diela

Napriek tomu, že sme sa stretli s pár problémami, konštatujeme, že sme s výsledným dielom pomerne spokojný nielen my, ale aj zadávateľ projektu. Počas vývoja dochádzalo ku zmenám jednotlivých častí funkcionality projektu, nakoľko presné požiadavky na jednotlivé časti neboli stanovené a bolo na nás aby sme ich vymysleli a implementovali. Práve kvôli tomuto dochádzalo ku častým zmenám vo vývoji a dolad'ovaniu jednotlivých myšlienok do finálnej podoby za chodu. Najpodstatnejší problém nastal so systémom prihlasovania. Keďže má byť systém integrovateľný s už existujúcimi systémami, tak tvorba dodatočného prihlasovacieho systému by bola enormne zložitá, lebo by musel byť schopný pracovať s ľubovoľným iným riešením. Preto sme sa rozhodli databázu výsledkov uschovávať lokálne a v prípade potreby je možné sa k nej dostať skopírovaním na iný počítač a následným otvorením.

### 15.3 Odchýlky od pôvodného plánu

Pri pôvodnom pláne sme si sami stanovili vytvorenie databázy jednotlivých vybraných slov, pričom do tejto databázy by bolo možné pridávať a odoberať jednotlivé slová. Táto podmienka nám neskôr prišla ako zbytočná, nakoľko vybraných slov, ktoré sa učia na prvých stupňoch základných škôl je fixný počet a pridávanie ďalších slov, je zbytočné. Na základe tohto sme vytvorili niekoľko asociatívnych polí pre vybrané a nevybrané slová, spolu s ich príslušnými obrázkami a od myšlienky vytvorenia databázy slov sme upustili.

Ďalším našim osobným plánom bolo pridávanie výsledkov z jednotlivých aktivít do databázy. Od tohto nápadu sme sa takisto odtrhli a vytvorili sme zaznamenávanie hodnôt pomocou html5 nástroja localStorage. Tieto výsledné hodnoty si každý z používateľov môže pozrieť v osobitnom okne aplikácie a v prípade potreby ich následne aj vymazať.

Vo všeobecnosti ako už bolo spomenuté dochádzalo počas implementácie k niekoľkým zmenám pri návrhu, alebo dizajne, kým sme nedosiahli požadovaného výsledku, s ktorým by všetci stakeholderi súhlasili.

## 15.4 Zhodnotenie tímovej práce

Tímová práca v našom tíme Mudkips bola nad očakávania priaznivá a všetci členovia tímu sa konštantne zapájali do jednotlivých úloh. Treba podotknúť, že práca na projekte prebiehala aj počas vianočných prázdnin, čo je pomerne jednoznačným ukazovateľom prístupu ku práci.

Na začiatku vývoja sme si rozdelili prácu medzi jednotlivých členov tímu, ale s postupom času každý z nás prispieval pri implementácii tam, kde bolo potrebné.

Zápisnice sa písali priebežne každý týždeň, ich tvorcovia sa pravidelne obmieňali.

Kód má svojich tvorcov uvedených ako komentáre.

Krátke zhodnotenie podľa členov:

Martin:

- vedúci teamu, šéf, sprostredkovateľ, navrhovateľ, novátor a objaviteľ
- najväčší podiel pri tvorbe dokumentov
- vytvorenie jadra backendu cvičenia 1 a 2
- úprava cvičenia 1 a 2
- backend lokálneho ukladania výsledkov
- úprava frontendu

Peter:

- novátor, overovateľ, podporovateľ a dokončovateľ
- podiel pri tvorbe dokumentov
- vytvorenie jadra backendu cvičenia 3
- úprava cvičení 1, 2 a 3
- úprava frontendu

Michal:

- organizátor, overovateľ, objaviteľ, a podporovateľ
- podiel pri tvorbe dokumentov
- úprava cvičení 1, 2 a 3
- úprava frontendu

Adrián:

- koordinátor, organizátor, podporovateľ a dokončovateľ
- podiel pri tvorbe dokumentov
- úprava cvičení 1, 2 a 3
- tvorba a úprava frontendu
- refaktORIZÁCIA backendu