**FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY  
UNIVERZITA KOMENSKÉHO**

**ZÁVEREČNÁ SPRÁVA**

*DIDAKTICKÝ SOFTVÉR NA VÝUKU PREVODOV JEDNOTIEK PRE ŽIAKOV SŠ*

Jakub Chlup

Matúš Kalužák

Adam Šabík

Ján Vodila

V Bratislave 19.1.2015

1. Špecifikácia požiadaviek

# Úvod

## Predmet špecifikácie

Táto špecifikácia požiadaviek na softvér (ďalej ŠPS) popisuje používateľské a funkčné požiadavky didaktického systému na výuku prevodov jednotiek pre žiakov SŠ (ďalej DSPV). Špecifikácia je súčasťou ústnej dohody medzi objednávateľom a dodávateľom. Bude slúžiť aj ako východisko pre vyhodnocovanie správnosti softvéru.

## Rozsah projektu a funkcie systému

Didaktický systém na výuku prevodov jednotiek bude poskytovať rozhranie pre výuku prevodov jednotiek. Primárnou úlohou systému je sprístupniť žiakom SŠ 1. ročníka možnosť precvičovania si znalostí, ktoré nadobudli na základnej škole v oblasti prevodov jednotiek. Druhotným zámerom systému je poskytnúť učiteľovi výsledky úloh, ktoré jeho žiaci riešili.

## Slovník pojmov, skratky

|  |  |
| --- | --- |
| ŠPS | Špecifikácia požiadaviek na softvér |
| DSPV | Didaktický systém na výuku prevodov jednotiek |
| SŠ | Stredná škola |
| Používateľ | Žiak alebo učiteľ SŠ |

# Celkový opis

## Kontext systému

DSPV predstavuje viacužívateľské webovské rozhranie pre žiakov stredných škôl na zdokonalenie práce s prevodmi jednotiek. Systém budú využívať žiaci stredných SŠ 1. ročníka, ako aj učitelia.

Učiteľ môže po registrácii do systému vytvárať skupiny, kde bude mať prístup k výsledkom a štatistikám svojich žiakov. Učiteľ môže v skupine zadať päťminutovky, ktoré budú vygenerované automaticky.

Žiak si pri registrácii vyberie skupinu, ktorú vytvoril jeho učiteľ. Po prihlásení môže žiak začať riešiť príklady, alebo riešiť aktuálne zadanú päťminutovku. Príklady budú pozostávať len z veličín, ktoré žiak pozná zo základnej školy (viď prílohu A).

## Funkcie systému

Hlavnou úlohou webovej aplikácie je poskytnúť učiteľom nástroj na testovanie žiakov 1. ročníka stredných škôl z premien jednotiek, ktoré by mali poznať zo základnej školy. Učiteľ môže zadávať päťminútovky pre študentov, ktoré môže ukončiť a následne si pozrieť výsledky päťminútovky. Ďalšou funkciou systému je, že žiak sa môže zdokonalovať v premieňaní jednotiek aj samostatne – riešením automaticky vygenerovaných úloh na prevod jednotiek.

Učiteľ bude môcť vytvárať skupiny. Tie budú mať názov a kľúč, ktoré učiteľ povie študentom, ktorých chce v danej skupine. Študenti sa budú pri registrácii pridávať do učiteľom vytvorenej skupiny. Učitelia si budú môcť prezerať riešenia jednotlivých študentov a štatistiky celej študijnej skupiny (napr. triedy), v ktorej sú študenti prihlásení.

Učitelia budú môcť mazať študentov zo skupiny – tým vymažú študenta zo systému a môže sa registrovať znova.

Študenti aj učitelia sa budú môcť registrovať, pre prístup k jednotlivým funkciám. Systémový administrátor ich môže mazať, pričom sa vymažú aj dáta súvisiace s nimi (napr. keď vymaže učiteľa, tak sa zmažú aj jeho skupiny, študenti v skupinách a ich riešenia).

## Špecifiká jednotlivých typov používateľov

|  |  |
| --- | --- |
| **Typ používateľa** | **Špecifický popis** |
| Žiak | Žiak sa pri registrácii priradí do svojej skupiny. Po prihlásení sa do systému môže riešiť príklady, ktoré mu systém automaticky vygeneruje. Žiak si môže pozrieť úspešnosť svojich posledných riešení. Taktiež mu je k dispozícii pomôcka pri riešení príkladov, ktorá sa zaznamenáva do systému. |
| Učiteľ | Učiteľ vytvára a spravuje skupiny do ktorých sa môžu prihlasovať žiaci (zväčša podľa triedy). Pre skupiny môže zadávať päťminútovky, ktoré budú riešiť žiaci na hodinách. Ku všetkým výsledkom a štatistikám jeho skupín bude mať prístup v systémovej administrácii. |
| Systémový administrátor | Riadi chod celého systému a zabezpečuje administráciu všetkých používateľov. |

## Systémové rozhrania

|  |  |
| --- | --- |
| SR-1 | Webové rozhranie – webová stránka systému |
| SR-2 | Databáza úloh |

## Hardvérové rozhrania

Systém neobsahuje žiadne hardvérové rozhrania.

## Softvérové rozhrania

|  |  |
| --- | --- |
| SSR-1 | DSPV bude vytvorený ako web aplikácia v PHP 5 |
| SSR-2 | Používateľské rozhranie musí korektne fungovať v najnovších webových prehliadačoch a mobilných zariadeniach. |

## Inicializačné požiadavky

|  |  |
| --- | --- |
| IP-1 | Pre vstup do systému je potrebné sa zaregistrovať |

# Konkrétne požiadavky

## Funkčné požiadavky

* Registrácia žiakov a učiteľov
* Prihlasovanie do systému
* Obnova hesla
* Zmena hesla
* Pridanie skupiny
* Automatické generovanie unikátnych príkladov
* Zobrazenie pomôcky pri riešení príkladu
* Vyhodnotenie správnosti riešenia príkladov
* Možnosť zadania päťminutovky žiakom
* Zobrazenie vyriešených príkladov
* Zobrazenie štatistík skupiny učiteľovi
* Vymazanie žiaka zo skupiny
* Zakladanie nových skupín žiakov
* Vymazávanie neaktuálnych príkladov z databázy

### Registrácia

Registrácia zahŕňa získanie informácii o používateľovi:

* Meno a priezvisko
* E-mail
* Heslo
* Skupina
* Typ používateľa (učiteľ/žiak)

Do systému sa môžu registrovať buď žiaci alebo učitelia. Žiak si musí pri registrácii vybrať skupinu zo zoznamu a do textového poľa zadá jej kľúč. Učiteľ si pri registrácii nevyberá žiadnu skupinu. Pri registrovaní sa nepotvrdzuje email užívateľa.

Email bude slúžiť na prihlásenie užívateľa do systému, takže musí byť unikátny.

Po odoslaní údajov je užívateľ automaticky registrovaný.

### Prihlásenie

Po úspešnej registrácii sa môže používateľ prihlásiť do systému pomocou svojho E-mailu a hesla, ktoré zadal v registrácii.

#### Obnova hesla

Ak používateľ zabudne heslo, môže si ho po kliknutí na odkaz pri prihlasovaní zmeniť – zadaním emailu a odoslaním formulára sa mu odošle na e-mail odkaz, kde si môže nastaviť nové heslo.

#### Úspešne prihlásený učiteľ

Po úspešnom prihlásení učiteľa bude presmerovaný na učiteľskú administračnú časť, kde má prístup k funkciám spravovania svoje skupiny – pridávať a mazať skupinu, odstrániť žiaka zo skupiny, prezerať si vypracované príklady svojich žiakov, štatistiky skupiny, zadávať päťminutovky.

#### Úspešne prihlásený študent

Po úspešnom prihlásení študenta bude presmerovaný na stránku, kde má prístup k nasledujúcim funkciám: riešenie úloh a päťminútoviek, môže ich riešiť a nechať si ich vyhodnotiť systémom, prezeranie svojich výsledkov.

### Zmena hesla

Prihlásený používateľ si môže zmeniť heslo – zadá aktuálne heslo, nové heslo a znova nové heslo na potvrdenie. Po úspešnom odoslaní formulára sa prihlasuje s týmto zmeneným heslom.

### Pridanie skupiny

Prihlásený učiteľ môže pridať skupinu, ktorej nastaví názov a kľúč. S týmito údajmi sa môžu študenti pri registrácii pridať do danej skupiny. Skupinou môže byť iba jedna trieda alebo celý ročník (skupina nie je kapacitne obmedzená). Učiteľ môže svojim skupinám pridať popis a obrázok.

### Generovanie príkladov

Generovanie príkladov bude spočívať v nasledovnom algoritme:  
Žiak si vyberie úroveň obtiažnosti príkladov. Na základe toho bude vygenerovaná jednotka z danej množiny (viď príloha A) a kladné reálne číslo pre ktoré platí, že obsahuje dve, alebo tri platné čislice. Číslo bude ohraničené z dola najviac dvomi nulami za desatinnou čiarkou (napr. 0.00245mm) a z hora číslom 1000 (napr. 230V). Premieňať sa bude vždy na základné jednotky.

### Zobrazenie pomôcky pri riešení príkladu

Žiak pri riešení príkladu môže využiť pomôcku, ktorá mu odkryje buď mocninu desiatky alebo základný tvar čísla (napr. x\*104, alebo 3.28\*10x).

### Vyhodnotenie správnosti riešenia

Ak žiak odošle svoje riešenie, systém v databáze automaticky označí správnosť žiakovho riešenia a zobrazí žiakovi jeho úspešnosť.

### Rýchle testovanie žiakov (päťminútovka)

Učiteľ spustí hromadné testovanie žiakov na hodine (zadá počet príkladov a ich obtiažnosť). Toto testovanie ukončí tlačidlom na stránke, kde ho spustil. Po ukončení testovania sa výsledky vyhodnotia a zobrazia učiteľovi.

### Zobrazenie vyriešených príkladov pre učiteľa

Prihlásený učiteľ si môže pozrieť výsledky žiakov v rámci svojich skupín (riešené príklady svojich žiakov) – po kliknutí na žiaka v skupine sa mu zobrazí, ktoré príklady ako vyriešil (príklad + študentov výsledok a ak riešil zle, alebo nevyriešil, tak aj správny výsledok).

### Zobrazenie štatistík skupiny učiteľovi

V administračnej časti bude mať učiteľ možnosť zobraziť si štatistiky jednotlivých skupín v podobe grafov:

* úspešnosť – koľko príkladov bolo celkovo dobre/zle vyriešených a neriešených
* úspešnosť (poradie) jednotlivých žiakov
* úspešnosť jednotlivých veličín (napr. premeny jednotiek vo veličine objem bolo nesprávnych zo všetkých nesprávnych 53.14%, plošný obsah 23%, ...)

### Vymazanie žiaka zo skupiny

Prihlásený učiteľ môže vymazať žiaka zo svojej skupiny – tým vymaže aj jeho riešené príklady a žiak sa musí nanovo registrovať.

### Vymazávanie príkladov z databázy

Systémový administrátor môže z databázy vymazať príklady, ktoré boli vygenerované v databáze pred dátumom, ktorý zadá.

### Vymazanie neaktívnych žiakov, učiteľov

Systémový administrátor môže mazať neaktívne skupiny a používateľov. Vymazaním učiteľa sa zmažú aj jeho skupiny, žiaci v týchto skupinách a ich príklady.

## Výkonnostné požiadavky

* Kapacita databázy musí byť dostatočne veľká na uloženie aspoň milióna príkladov.
* Vyhodnotenie príkladov musí byť k dispozícii ihneď po odoslaní riešení.

## Dostupnosť

* Systém musí byť dostupný 24 hodín denne
* Systém nesmie byť závislý na jednom type prehliadača
* Systém musí byť dostupný aj z mobilných zariadení (tablet, smartphone, ...)

# Príloha A: Zoznam jednotiek

|  |  |  |  |
| --- | --- | --- | --- |
| **Názov veličiny** | **Základná jednotka** | **Predpony** | **Obtiažnosť** |
| Dĺžka | Meter (m) | pm, nm, µm, mm, cm, dm, km | Normálna |
| Elektrický prúd | Ampér (A) | pA, nA, µA, mA, kA, MA, GA, TA | Normálna |
| Elektrické napätie | Volt (V) | pV, nV, µV, mV, kV, MV, GV, TV | Normálna |
| Elektrický odpor | Ohm (Ω) | pΩ, nΩ, µΩ, mΩ, kΩ, MΩ, GΩ, TΩ | Normálna |
| Výkon | Watt (W) | pW, nW, µW, mW, kW, MW, GW, TW | Normálna |
| Práca/Energia | Joule (J) | pJ, nJ, µJ, mJ, kJ, MJ, GJ, TJ | Normálna |
| Hmotnosť | Kilogram (kg) | pg, ng, µg, mg, g, tona, kiloton, megaton, gigaton | Normálna |
| Sila | Newton (N) | pN, nN, µN, mN, kN, MN, GN, TN | Normálna |
| Tlak | Pascal (Pa) | pPa, nPa, µPa, mPa, kPa, MPa, GPa, TPa | Normálna |
| Frekvencia | Hertz (Hz) | pHz, nHz, µHz, mHz, kHz, MHz, GHz, THz | Normálna |
| Plocha | Meter štvorcový (m2) | pm2, nm2, µm2, mm2, cm2, dm2, km2 | Ťažká |
| Objem | Meter kubický (m3) | pm3, nm3, µm3, mm3, cm3, dm3, km3, ml, cl, dl, l | Ťažká |

1. Konceptuálna analýza

# Úvod

Cieľom tohto je dokumentu je analýza používateľov systému, prostredníctvom diagramov prezentovať funkcionality systému a návrh používateľského rozhrania – GUI. V tomto dokumente vychádzame z platného katalógu požiadaviek. Dokument je štruktúrovaný do nasledujúcich kapitol:

* Analýza používateľov – obsahuje analýzu používateľov tak, ako sú špecifikovaní v katalógu požiadaviek. Konkrétne obsahuje kategorizáciu používateľov podľa používateľských práv na prístup k systému
* Diagramy – obsahuje konkrétne diagramy, menovite entitno-relačný diagram, v ktorom sú zobrazené jednotlivé entity a vzťahy medzi nimi, use-case diagram, ktorý vychádza priamo z platného katalógu požiadaviek, a stavový diagram, v ktorom je možné vidieť jednotlivé stavy, v ktorých sa systém môže nachádzať a ktorý rovnako vychádza priamo z platného katalógu požiadaviek
* Používateľské rozhranie – obsahuje popis a grafické (obrazové) návrhy používateľského rozhrania

# Analýza používateľov

## Role používateľov

Užívateľov systému sme rozdelili na 3 kategórie, ktoré vyplývajú z katalógu požiadaviek

* žiaci/študenti
* učitelia
* administrátor

## Žiaci/študenti

Žiaci sa budú môcť na stránke registrovať. Pri registrácii si vyberú skupinu a zadajú kľúč tejto skupiny, ktoré získajú od správcu skupiny – učiteľa. Následne môžu riešiť zadané úlohy, alebo päťminutovky, ktoré zadáva učiteľ, príklady sa generujú automaticky. Budú mať práva na zobrazovanie svojich starých nevymazaných riešení. Nebudú mať právo na mazanie týchto úloh.

## Učitelia

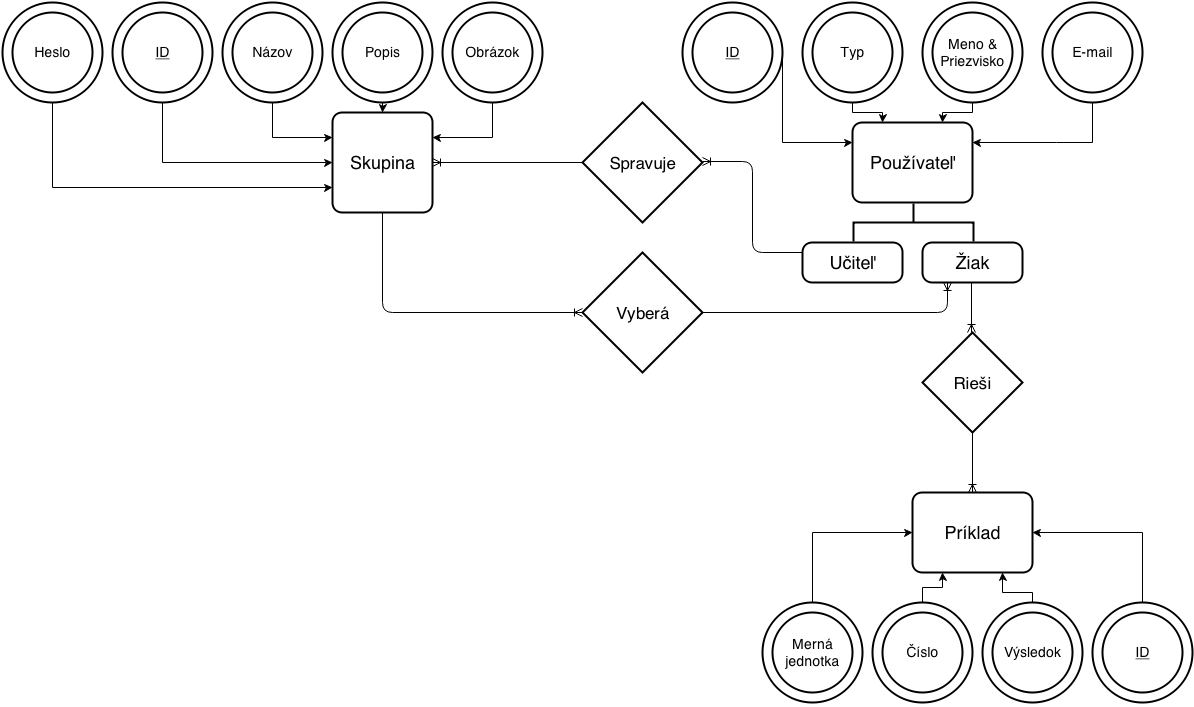
Učitelia sa registrujú. Po prihlásení budú mať práva na vytvorenie skupiny, ktorej musia nastaviť heslo. Ďalej majú práva na vymazanie študentov z ich vytvorenej skupiny, môžu si pozrieť výsledky úloh/päťminutoviek jednotlivých žiakov. Budú mať dostupnú štatistiku celej skupiny (grafy úspešnosti). Majú práva na zrušenie celej skupiny.

## Administrátor

Stará sa o údaje v databáze. Môže zmazať dlhšie neaktívnych žiakov (s nimi aj ich skupiny, žiakov v skupinách a úlohy daných žiakov)/učiteľov, môže vymazať vyriešené príklady staršie ako ním určený dátum.

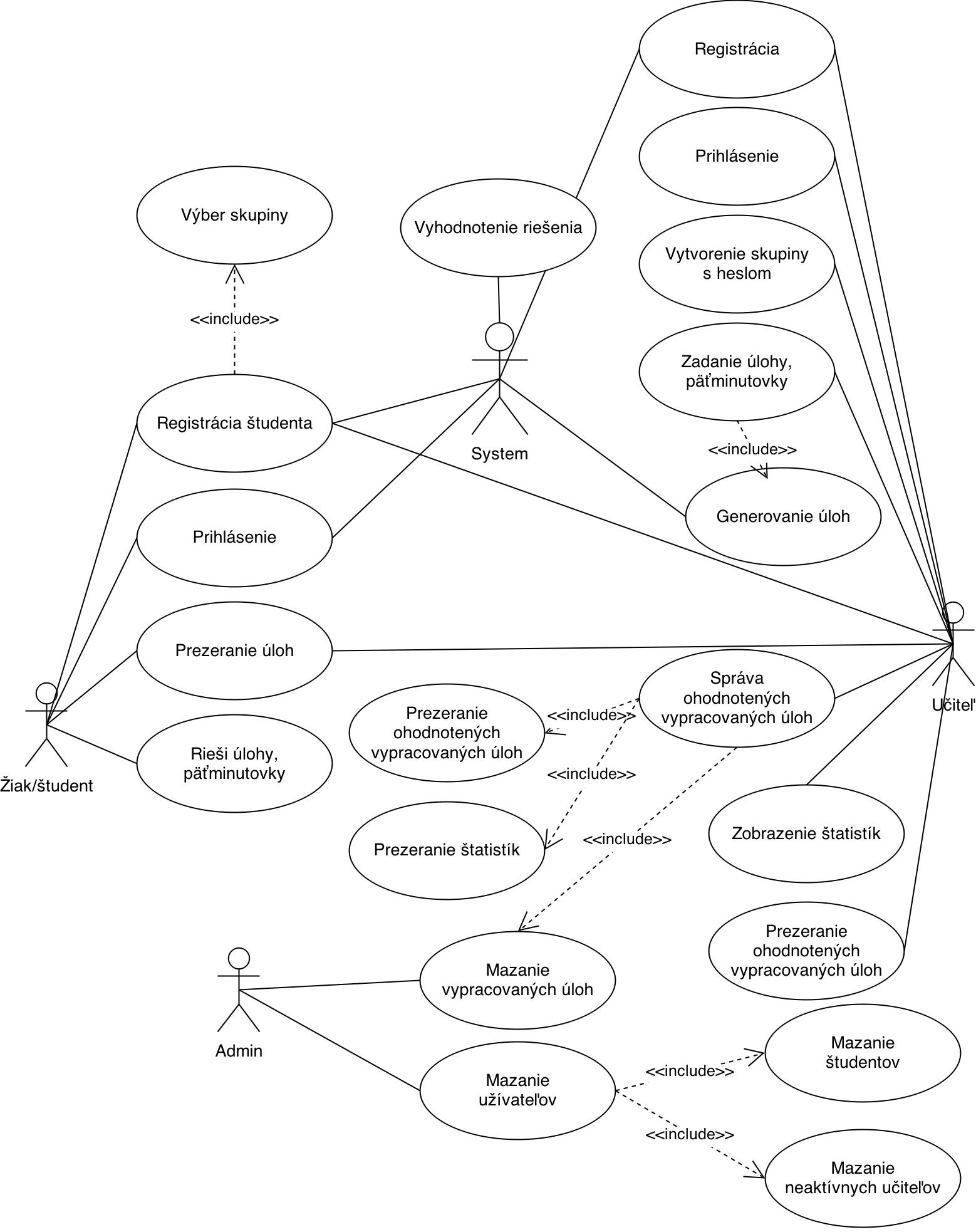
# Diagramy

## Entitno-relačný diagram



*obr. 1 Entitno relačný diagram*

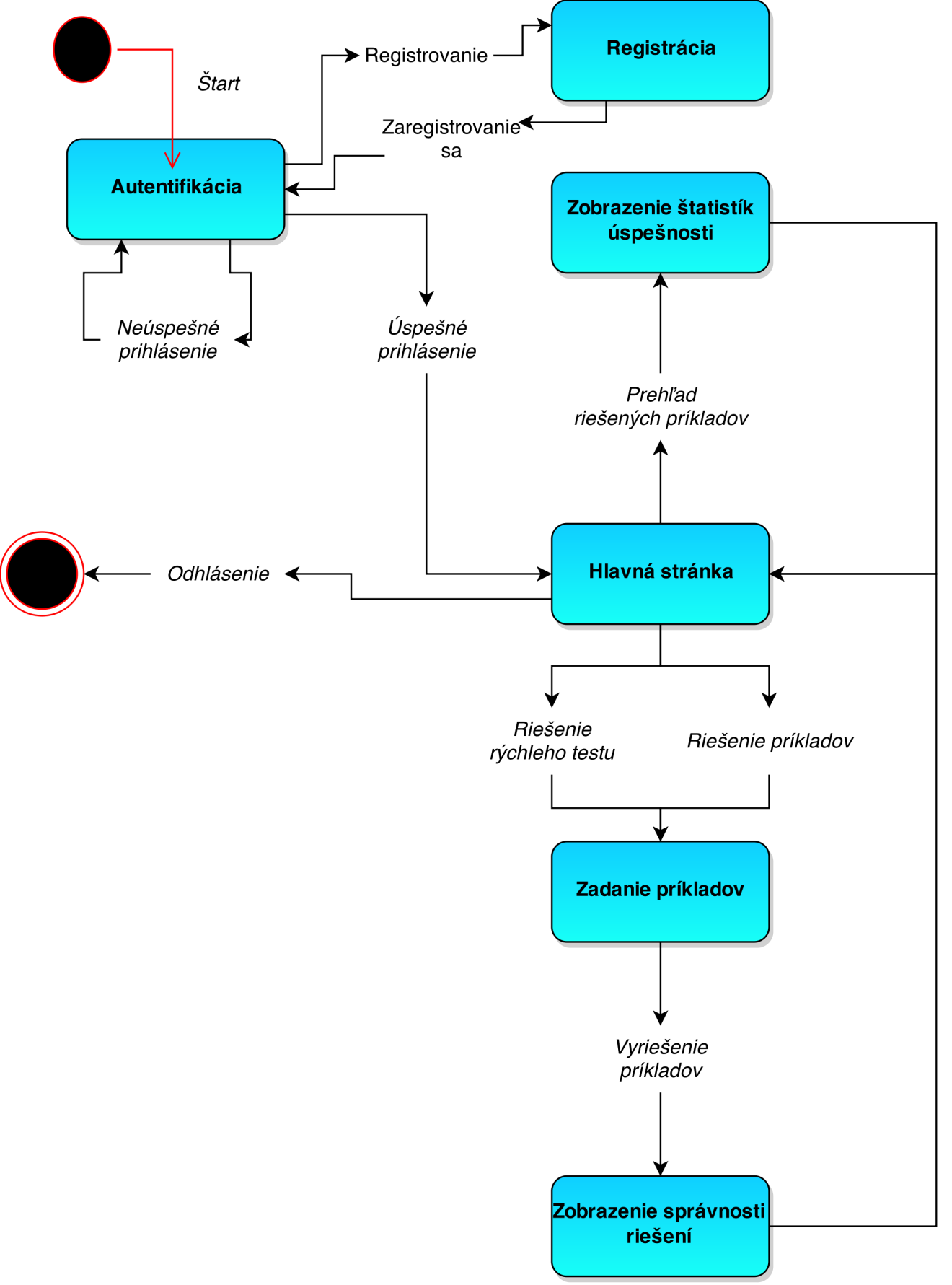
## Use-case diagram



*obr. 2 Use-case diagram*

## Stavový diagram

Stavový diagram možných stavov pre užívateľa typu žiak.



*obr. 3 Stavový diagram*

# Používateľské rozhranie

Táto časť dokumentu sa venuje opisu používateľského rozhrania, teda jednotlivým stránkam, s ktorými sa užívateľ softvéru stretne. Opis rozhrania je rozdelený podľa typov užívateľov a teda na rozhranie pre žiaka, učiteľa a administrátora.

## Rozhranie pre žiakov

### Registrácia

Žiak počas registrácie do formuláru zadá meno a priezvisko, e-mail, heslo, vyberie si skupinu a zadá jej heslo, ktoré dostal od učiteľa a zaškrtne typ používateľa žiak.

### Po prihlásení

Žiakovi sa po úspešnom prihlásení zobrazí úvodná stránka. Na ktorej nájde odkazy na:

* Zadanú päťminútovku
* riešenie ľubovoľných úloh
* prehľad svojich riešení

### Riešenie päťminútoviek

Na tejto stránke sa žiakovi zobrazí učiteľom zadaný počet príkladov, ktoré má riešiť. Pri riešení príkladov žiak nemá k dispozícií nápovedi. Po vyriešení, prípadne po ukončení päťminútovky učiteľom, sa zobrazí stránka s vyhodnotením, na ktorej žiak zistí, ktoré príklady mal dobre, ktoré zle.

### Riešenie ľubovoľných príkladov

Žiak zadá do formuláru počet príkladov, ktoré chce riešiť a aj úroveň obtiažnosti. Následne sa mu zobrazia vygenerované príklady, ktoré môže začať riešiť. Žiakovi je ku každému príkladu prístupná nápoveda. Príklad si môže dať vyhodnotiť každý zvlášť, alebo všetky naraz.

### Prehľad svojich riešení

Na tejto stránke sa žiakovi zobrazia jeho vyriešené príklady, ktoré budú označené, či sú správne, alebo nesprávne a taktiež dátum a čas ich riešenia. Ďalej sa mu zobrazí celkový počet príkladov, ktoré mal dobre/zle.

## Rozhranie pre učiteľov

### Registrácia

Učiteľ počas registrácie do formuláru zadá meno a priezvisko, e-mail, heslo a zaškrtne typ používateľa učiteľ. Skupinu a heslo skupiny nevypĺňa.

### Po prihlásení

Učiteľovi sa zobrazia informácie o poslednej aktívnej skupine (názov skupiny, zoznam mien žiakov v skupine). Taktiež zoznam jeho skupín, po kliknutí na konkrétnu skupinu sa údaje o skupine zmenia. Ďalej bude mať prístupné odkazy na:

* zadanie a spustenie päťminútovky
* prehľad štatistík skupiny
* informácie o žiakovi (zobrazí sa kliknutí na meno žiaka v zozname)

### Zadávanie päťminútoviek

Učiteľ zadá počet a úroveň obtiažnosti príkladov, následne spustí testovanie. Potom sa mu zobrazí tlačidlo na ukončenie päťminútovky. Následne sa mu zobrazia vyhodnotené riešenia jednotlivých žiakov.

### Prehľad štatistík skupiny

Učiteľovi sa zobrazí celková štatistika danej skupiny v podobe grafov a tabuliek.

Konkrétne:

* úspešnosť – koľko príkladov bolo celkovo dobre/zle vyriešených a neriešených
* úspešnosť (poradie) jednotlivých žiakom
* úspešnosť jednotlivých veličín (premeny jednotiek vo veličine objem bolo nesprávnych zo všetkých nesprávnych 53.14%, plošný obsah 23%, ...)

### Informácie o žiakovi

Na tejto stránke sa zobrazí meno, priezvisko vybraného žiaka, celkový počet riešených a správne vyriešených príkladov. Taktiež sa zobrazia konkrétne príklady, ktoré žiak riešil aj s jeho odpoveďami, príklady budú vyznačené podľa správnosti žiakovho riešenia. Na tejto stránke bude mať učiteľ možnosť odstrániť daného žiaka zo skupiny.

## Rozhranie pre administrátora

Po prihlásení sa administrátorovi zobrazí stránka s možnosťou vymazania príkladov, ktoré sú staršie ako ním zadaný dátum. Taktiež sa vypíše zoznam mien učiteľov s možnosťou vymazať ich. Po kliknutí na meno učiteľa sa zobrazia jeho skupiny a po kliknutí na skupinu žiaci danej skupiny. Skupiny ako aj jednotlivých žiakom bude môcť administrátor vymazať.

1. Analýza technológií a návrh

# Analýza technológií

Jednou z požiadaviek zadávateľa je vytvoriť aplikáciu, ktorá je dostupná 24 hodín denne a aby bola dostupné nielen z klasického PC ale aj z mobilných zariadení ako je tablet, smartphone a podobne.

## Použité technológie

Na základe spomenutých požiadaviek sme sa rozhodli vytvoriť webovú aplikáciu postavenú na PHP technológii s využitím Nette Frameworku. Nette sa veľmi dobre hodí pri vytváraní našej aplikácie, keďže sa jedná o objektovo orientovanú štruktúru, ktorá podporuje MVC architektúru, znovupoužiteľnosť kódu a mnoho ďalších predpripravených funkcií. Samozrejme sa počíta s PHP vo verzii 5, ktorá rozširuje možnosti písania objektovo orientovaného kódu, narozdiel od staršej verzii PHP 4, ktorá bola skôr procedurálne založená.

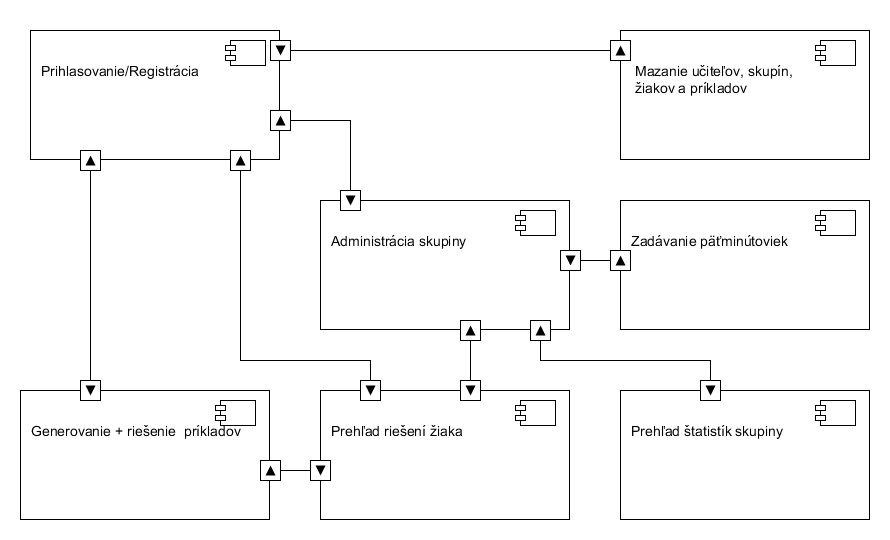
Keďže naša aplikácia bude potrebovať ukladanie záznamov do databázy, rozhodli sme sa použiť MySQL systém. Je pravda, že v porovnaní s Oracle databázou nedisponuje takými bohatými možnosťami a funkciami, no výhodou tohto systému a dôvodom, prečo sme sa rozhodli použit MySQL je fakt, že je to open-source systém a teda oproti Oracle databáze, nepotrebujeme platiť žiadné licenčné poplatky. Spoločne s PHP tak máme dve silné a hlavne bezplatné technológie, vďaka ktorým môžeme vytvoriť našu aplikáciu.

Vizuálna časť bude napísaná v HTML dokumente s využitím CSS technológie. Pre zjednodušenie písania CSS sme sa rozhodli použiť voľne dostupný Bootstrap framework, ktorý slúži ako predpripravený nástroj pri vytvárani vizuálnej časti webového dokumentu. Pre lepšie písanie CSS kódu sme sa rozhodli použiť SASS skriptovací jazyk, ktorý nám dovoľuje využiť možnosti, ktoré nám normálne CSS neponúka, napríklad: premenné, dedičnosť, vnorenie, deklarovanie a používanie vlastných ale aj predpripravených funkcií a procedúr.

O dynamickosť na strane klienta sa bude starať JavaScript. V rámci systému sa predpokladá jeho použitie na niektoré vybrané akcie, najmä na zobrazovanie pomôcky pri riešení príkladu. Takisto sa JavaScript, respektíva knižnica jQuery s pluginom CanvasJS, bude starať o vykresľovanie grafov v administratívnej sekcii učiteľa. Vyžaduje sa preto najnovšia verzia HTML 5, ktorá podpuruje canvas element.

## Dekompozícia

### Komponentový diagram



*obr. 1 Komponentový diagram*

### Popis komponentov

#### Prihlasovanie/Registrácia

Tento komponent je určený pre učiteľov a žiakov. Pri registrácií komponent overí zadané údaje a následne ich uloží do databázy. Rozlišuje medzi registráciou učiteľa a žiaka, v prípade žiaka overuje aj zadaného heslo k ním vybranej skupine.V prípade úspešnej registrácie informuje užívateľa o možnosti prihlásiť sa a v opačnom prípade o nastanej chybe. Komponent taktiež slúži pre prihlasovanie všetkých užívateľov systému. Po zadaní mena a hesla sa údaje overia v databáze a prihlásenie sa povolí alebo zamietne. Po prihlásení sa zobrazí stránka určená pre daný typ užívateľa.

#### Generovanie + riešenie príkladov

Komponent podľa zadanej obtiažnosti náhodne vyberie z databázy základnú jednotku a príponu. Ďalej vygeneruje náhodné kladné reálne číslo pre ktoré platí, že obsahuje dve, alebo tri platné číslice. Komponent k vygenerovanému príkladu vytvorí riešenie a výsledok. Bude žiakovi schopný odkryť mocninu desiatky alebo základný tvar čísla, a taktiež vyhodnotiť žiakove riešenie príkladu.

#### Prehľad riešení žiaka

Komponent bude prístupný žiakovi aj učiteľovi. Z databázy bude zobrazovať príklady, ktoré žiak vypočítal aj s jeho riešeniami a ich správnosťou. V prípade nesprávneho riešenia zobrazí aj správny výsledok príkladu.

#### Administrácia skupiny

Tento komponent učiteľovi zobrazuje jeho skupiny a žiakov v nich. Ďalej mu umožňuje vytvoriť novú skupinu. Zadané údaje o nej ako meno, heslo, obrázok a popis uloží do databázy. Učiteľovi umožní vymazať žiaka zo skupiny.

#### Zadávanie päťminútoviek

Komponent sprostredkuje prihlásenému učiteľovi vytváranie päťminútoviek pre danú skupinu. Učiteľ vyberie počet a obtiažnosť príkladov, tieto údaje sa vložia do databázy a komponent ďalej upozorní prihlásených žiakov o nových povinnostiach.

#### Prehľad štatistík skupiny

Komponent slúži učiteľovi pre lepší prehľad o skupine. V podobe grafov a tabuliek vie zobraziť úspešnosť – koľko príkladov bolo celkovo dobre/zle vyriešených, úspešnosť (poradie) jednotlivých žiakov, a ďalej úspešnosť jednotlivých veličín.

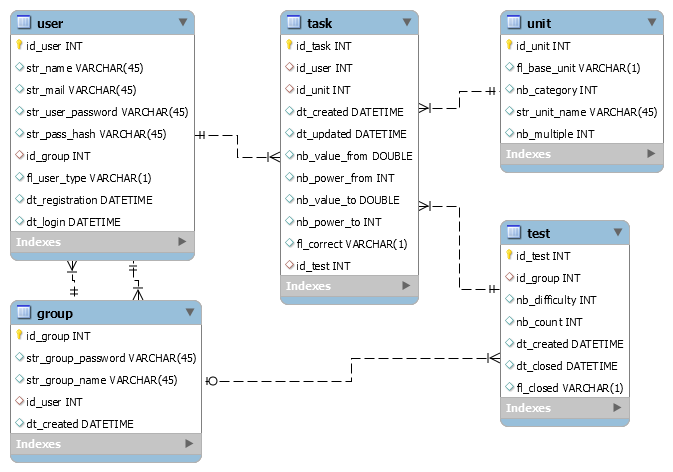
#### Mazanie učiteľov, skupín, žiakov a príkladov

Tento komponent umožňuje prihlásenému administrátorovi spravovať databázu. Administrátorovi sprístupňuje možnosti mazať jednotlivých žiakov, celé skupiny alebo učiteľov aj s ich skupinami a ich žiakmi. Ďalej umožňuje z databázy vymazať riešenia príkladov staršie ako administrátorom zadaný dátum.

## Dátový model

Aplikácia bude pracovať s databázou Mysql. Štruktúra databázy je popísaná v nižšie uvedenom entitno-relačnom diagrame. Diagram zobrazuje jednotlivé tabuľky (entity), ich stĺpce (atribúty), primárne a cudzie kľúče.

### EER diagram



*obr. 2 Dátový model*

### Popis stĺpcov v databáze

|  |  |  |
| --- | --- | --- |
| **Table** | **Column** | **Description** |
| group | id\_group | Primárny kľúč entity group |
| group | str\_group\_password | Heslo k skupine v hash forme |
| group | str\_group\_name | názov skupiny |
| group | id\_user | používateľ, ktorý vytvoril skupinu |
| group | dt\_created | Čas vytvorenia skupiny |
| task | id\_task | Primárny kľúč entity task |
| task | id\_user | používateľ, ktorému patrí daný príklad |
| task | id\_unit | jednotka, z ktorej bolo prevádzané na základnú jednotku |
| task | dt\_created | Čas vytvorenia príkladu |
| task | dt\_updated | Čas update príkladu - zadanie riešenia |
| task | nb\_value\_from | Číslo z akého sa premiena, napr. "12.6" |
| task | nb\_power\_from | Mocnina z akej premieňame |
| task | nb\_value\_to | Základný tvar čísla, ktorý zadal užívateľ |
| task | nb\_power\_to | Mocninu akú užívateľ zadal |
| task | fl\_correct | Správnosť vyriešenia príkladu |
| task | id\_test | Cudzí kľúč na test, ak daný príklad patrí k testu |
| test | id\_test | Primárny kľúč |
| test | id\_group | Cuzdí kľúč na grupu |
| test | nb\_difficulty | Zadaná obtiažnosť |
| test | nb\_count | Počet príkladov |
| test | dt\_created | Čas zadania písomky |
| test | dt\_closed | Čas skončenia |
| test | fl\_closed | Flaga či je ešte aktívna |
| unit | id\_unit | Primárny kľúč |
| unit | fl\_base\_unit | Flaga, či sa jedná o základnú jednotku |
| unit | nb\_category | Kategória, do ktorej patrí veličina |
| unit | str\_unit\_name | Značka jednotky |
| unit | nb\_multiple | Násobok jednotky, nemusí sa vzťahovať k základnej jednotke! |
| user | id\_user | Primárny klúč entity user |
| user | str\_name | Meno usera |
| user | str\_mail | E-mail používateľa |
| user | str\_user\_password | Heslo používateľa v hash forme |
| user | str\_pass\_hash | Zahashovany string potrebny pri obnove hesla |
| user | id\_group | id\_group, kam žiak patrí |
| user | fl\_user\_type | označenie, či sa jedná o učiteľa/žiaka |
| user | dt\_registration | Čas registrácie |
| user | dt\_login | Čas posledného loginu |

# Návrh

## Testovacie scenáre

Táto časť dokumentu popisuje testovacie scenáre, ktoré budú slúži pri testovaní funkčnosti jednotlivých modulov systému.

### Registrácia a prihlásenie učiteľa

1. Na hlavnej stránke zvolíme registráciu nového užívateľa a zaregistrujeme nového učiteľa.
2. Vyplníme všetky údaje registračného formulára s tým, že najprv vyplníme políčka chybne, aby sme overili, či naozaj náš systém kontroluje jednotlivé vstupy:
   1. Formulár necháme nevyplnený a odošleme
   2. Zadáme krátke vstupy (meno, heslo)
   3. Email vyplníme v zlom formáte (bez '@')
3. Všetky údaje registračného formulára vyplníme správne, vyberieme si typ používateľa „učiteľ“ a odošleme formulár.
4. Registrácia je úspešná a systém nás presmeroval späť na hlavnú stránku s formulárom pre prihlásenie.
5. Prihlasovací formulár vyplníme chybne, aby sme overili, či naozaj náš systém kontroluje jednotlivé vstupy:
   1. Formulár necháme nevyplnený a odošleme
   2. Zadáme chybné používateľské meno
   3. Zadáme chybné heslo
6. Všetky údaje pre prihlasovanie vyplníme správne a odošleme formulár.
7. Systém nás presmeruje na úvodnú stránku učiteľa.

### Vytvorenie skupiny

1. V administratívnej časti učiteľa vyberieme možnosť „vytvoriť novú skupinu“.
2. Podobne ako pri registrácii, vyplníme políčka chybne, aby sme overili, či naozaj náš systém kontroluje jednotlivé vstupy.
3. Všetky údaje registračného formulára vyplníme správne a formulár odošleme.
4. Skontrolujeme či sa naša novovytvorená skupina nachádza v zozname skupín.

### Registrácia a prihlásenie žiaka

1. Postupujeme rovnako ako pri registrácii a prihlasovaní učiteľa s nasledujúcimi zmenami:
   1. Formulár otestujeme, či kontroluje ak zadáme meno alebo e-mail, ktorý sa už v databáze nachádza.
   2. Otestujeme, či pri vybraní typu užívateľa „žiak“ sa odkryjú vstupné políčka pre názov skupiny a jej tajný kľúč a po zmene na „učiteľa“ sa tieto polička skryjú. (ráta sa so zapnutým JavaScript-om)
   3. Overíme kontrolu skupiny a jej tajného kľúča. Zadáme chybnú dvojicu (systém upozorní na nesprávne zadaný kľúč).
2. Po prihlásení nás systém presmeruje na úvodnú stránku žiaka.

### Generovanie a riešenie príkladov

1. Do systému sa prihlásime ako žiak.
2. Vyberieme si možnosť riešiť príklady.
3. Zadáme počet príkladov a obtiažnosť, ktoré chceme riešiť.
4. Skontrolujeme, či je nastavený limit pre počet príkladov (1-20) a či sa daný počet príkladov aj skutočne vygeneroval.
5. Skontrolujeme, či vygenerované príklady obsahujú 2-3 platné číslice a zodpovedajú ostatným požiadavkám.
6. Skontrolujeme, či vygenerované jednotky spadajú pod danú obtiažnosť.
7. Vyplníme riešenie príkladov (niektoré nevyplníme).
8. Odošleme výsledok a skontrolujeme, či systém skontroloval správne výsledky.
9. Tento test vykonáme viackrát, aby sme sa uistili, že funguje náhodné generovanie príkladov.

### Administrácia skupiny

1. Do systému sa prihlásime ako učiteľ.
2. Zo zoznamu si vyberieme ľubovoľnú skupinu a klikneme na odkaz „správa skupiny“.
3. Zmeníme názov skupiny, popis, tajný kľúč a overíme, či sa nové údaje uložili.
4. Vymažeme ľubovoľného žiaka a otestujeme, či sa naozaj vymazal (skúsime sa prihlásiť s údajmi práve vymazaného žiaka)
5. Vymažeme celú skupinu a overíme, či sa vymazali aj jej žiaci (rovnako ako v 4. bode).

### Prehľad riešení žiaka

1. Do systému sa prihlásime ako žiak a vyrátame zopár príkladov.
2. Zapíšeme si počet správnych, nesprávnych a neriešených príkladov, aby sme mohli skontrolovať, či systém generuje správne štatistiky.
3. Skontrolujeme naše údaje s tými, ktoré nám vygeneruje systém.
4. Overíme správnosť zobrazovaných grafov.
5. Prihlásime sa ako učiteľ, vyhľadáme si žiaka a vykonáme testy z bodu 3 a 4.

### Zadávanie päťminútovky

1. Prihlásime sa ako učiteľ, vyberieme si skupinu a odštartujeme päťminútovku kliknutím na tlačidlo „zadať päťminútovku“.
2. Podobne ako pri riešení príkladov, zadáme počet príkladov, obtiažnosť a kliknutím na tlačítko „Štart“ odštartujeme päťminútovku
3. Prihlásime sa ako žiaci z danej skupiny, klikneme na tlačidlo „riešiť päťminútovku“ a overíme správnosť vygenerovaných príkladov.
4. Ďalej kontrolujeme podobne ako pri riešení príkladov.
5. Vyplníme riešenie príkladov a odošleme naše riešenie.
6. Systém nám vyhodnotí naše riešenie a skontrolujeme, či systém skontroloval správne výsledky.
7. Znova sa prihlásime ako učiteľ a danú päťminútovku zastavíme a overíme správnosť riešení.
8. Skontrolujeme, či je päťminútovka naozaj zastavená a teda, že sa nedá ďalej odosielať výsledky, keď sme prihlásený ako žiak.

### Záver z testovania

Na základe výsledkov jednotlivých testov budeme náš systém považovať za otestovaný a funkčný.

## Triedny Návrh

### Modely

Umožňujú prácu s databázou využitím ORM, ktoré poskytuje Nette. Každá tabuľka má svoj model (resp. repository).

#### UserRepository

Slúži pre vyberanie a vkladanie a mazanie dát pre tabuľku User.

**Metódy**

* getUserByEmail($email): vráti záznam o užívateľovi podľa zadaného parametra $email
* getUserById($id): vráti záznam o užívateľovi podľa id užívateľa
* getUserTimeCreated($id): vráti čas, kedy bol užívateľ s daným id vytvorený
* addUser($user\_array): pridá nového užívateľa do databázy
* removeUser($user\_id): odstráni užívateľa z databázy a aj jeho príklady (foreign key v Task a Test tabuľkách)
* updateUserPassword($user\_id, $new\_password, $hash): zmení heslo užívateľovi
* getUserByHash($hash): vráti užívateľa, ktorý si chce zmeniť heslo – podľa prideleného $hash
* generateHashForUser($hash): pridelí hash užívateľovi, ktorý si chce zmeniť heslo
* removeStudentFromGroup($group\_id, $student\_id): odstráni študenta z danej skupiny
* getStudentsByGroup($group\_id): vráti záznamy o študentoch z danej skupiny z tabuľky

#### GroupRepository

Spravuje tabuľku group, kde sú záznamy o učiteľom vytvorených skupinách

**Metódy**

* addGroup($teacher\_id, $group\_name, $group\_key, $description): pridá skupinu do tabuľky
* removeGroup($group\_id): odstráni skupinu z tabuľky (a tým kaskádovito aj žiakov a ich príklady)
* getTeacherGroups($teacher\_id): vráti záznamy o skupinách daného učiteľa

#### TaskRepository

Spravuje údaje o vyriešených príkladoch študentov

**Metódy**

* addTask($id\_user, $id\_unit, $id\_test, $convert\_array, $isTest=false): pridá príklad do tabuľky. Pole $convert\_array bude mať info o príklade (nb\_value\_from, nb\_power\_from, nb\_value\_to, nb\_power\_to)
* markTaskAsCorrect($id\_task, $is\_correct): označí v tabuľke, či žiak správne vyplnil úlohu
* getTasksByStudent($student\_id): vráti príklady na základe id študenta
* getTasksBytTest($test\_id): vráti dvojrozmerné pole, kde prvý rozmer budú študenti a druhý ich riešené príklady

#### UnitRepository

Slúži na komunikáciu s tabuľkou unit, kde sú informácie o fyzikálnych jednotkách

**Metódy**

* getUnitById($unit\_id): vráti záznam o fyz. jednotke podľa id jednotky
* getBaseUnitByDifficulty($difficulty): vráti náhodne záznam o základnej jednotke podľa obtiažnosti
* getRandomUnitForBaseUnit($unit\_id): vráti náhodnú jednotku ku základnej jednotke (okrem nej) z tej istej veličiny

#### TestRepository

Pracuje s tabuľkou, kde sú uložené informácie o testoch pre študentov

**Metódy**

* addTest($id\_group, $count, $difficulty=1): pridá nový test
* closeTest($test\_id): ukončí daný test

### Controllery

Controllery MVC frameworku Nette slúžia na komunikáciu s Modelmi (cez ktoré dopytujú dáta z databázy a modifikujú ich). Tieto dáta môžu ďalej spracovávať a poskytnúť ich View-om, ktoré slúžia ako HTML šablóny. Ďalej spracovávajú dáta získané z formulárov. Každý Controller dedí z triedy BasePresenter.

#### Auth Controller

Riadi autentifikáciu používateľa (študenta/učiteľa) – registráciu, prihlasovanie, odhlasovanie, zmenu hesla používateľa. Tento controller komunikuje s triedou \Nette\Security\User, ktorá uchováva informáciu o skupine, užívateľovi a obsahuje metódy na prácu s užívateľom – registrácia, prihlásenie, odhlásenie, zmenu hesla v databáze.

**Metódy**

* createComponentNewRegisterUser(): vráti formulár na registráciu užívateľa (žiaka, alebo učiteľa)
* newRegisterUserSubmitted(): zvaliduje a spracuje odoslaný registračný formulár pomocou triedy \Nette\Security\User
* createComponentNewLoginForm(): vráti formulár na prihlásenie užívateľa (žiaka, alebo učiteľa)
* newLoginFormSubmitted(): spracuje odoslaný prihlasovací formulár pomocou triedy \Nette\Security\User. Z tejto triedy sa zavolá metóda login($user, $password), kde sa pomocou parametrov pošle email a heslo zadané užívateľom vo formulári. Tá prihlási užívateľa nastavením Sessions, kde si pamätá informácie o užívateľovi.
* newLogoutFormSubmitted(): metóda odhlasuje prihláseného užívateľa

#### Student Controller

Riadi požiadavky od prihláseného študenta. Pri každej z týchto akcií musí byť študent prihlásený. To sa overuje v metóde Startup() tejto triedy.

**Metódy**

* actionDefault(): zobrazí študentovi jeho domovskú stránku po prihlásení
* actionSolutions(): spracuje a zobrazí študentovi jeho riešené úlohy
* createComponentNewTask(): vygeneruje žiakovi príklady pomocou triedy UnitConversion, uloží do databázy prostredníctvom modelu a zobrazí mu ich na riešenie
* createComponentNewTest(): vygeneruje žiakovi príklady z danej päťminutovky podľa nastavenej obtiažnosti pomocou triedy UnitConversion a uloží do databázy. Príklady na riešenie zobrazí študentovi
* newTaskSubmitted(isTest = false): vyhodnotí odoslané príklady žiaka a uloží jeho výsledky do databázy pomocou modelu a presmeruje ho na actionShowResult()
* actionShowResult(): zobrazí študentovi výsledky (ktoré príklady mal dobre/zle, alebo nerieš)

#### Teacher Controller

Riadi požiadavky od prihláseného učiteľa. Pri každej z týchto akcií musí byť učiteľ prihlásený. Pri každej z týchto akcií musí byť študent prihlásený.

**Metódy**

* actionDefault (): zobrazí učiteľovi jeho domovskú stránku po prihlásení. Tu mu vylistuje jeho pridané skupiny a možnosť vytvoriť nové skupiny.
* createComponentNewGroup(): vráti formulár na vytvorenie skupiny.
* newGroupSubmitted(): vytvorí novú skupinu zo zadaných údajov vo formulári a presmeruje na actionDefault()
* actionRemoveGroup($group\_id): odstráni učiteľovu skupinu a s ňou jej študentov a ich príklady
* actionShowStudentsInGroup($group\_id): zobrazí učiteľovi študentov z danej skupiny
* actionShowStudentsResults($student\_id, $only\_tests=false): zobrazí učiteľovi študentove riešené príklady
* actionSetTest($count, $difficulty): učiteľ zadá test pre žiakov (prostredníctvom modelu sa nastaví v db., že daná skupina má test)
* actionCloseTest(): učiteľ ukončí testovanie (skončí sa v DB) a presmeruje na actionShowTestResults()
* actionShowTestResults(): zobrazia sa učiteľovi výsledky žiakov z ukončenej päťminútovky
* actionShowGroupStatistics(): vráti štatistiky danej skupiny v podobe grafov vyšpecifikovaných v predošlých dokumentoch

### Knižnice, samostatné triedy

#### UnitConversion

Táto trieda bude slúžiť na generovanie príkladov na premeny jednotiek a ich kontrolu. Bude pri tom využívať potrebné modely na prácu s databázou.

* generateConversion($difficulty = null, $unit = null): metóda vygeneruje a vráti príklad v tvare [‘id\_phys\_quantity’=>1, ‘from\_unit’=>’centimeter’, ‘from\_unit\_symbol’=>’cm’, ‘to\_unit’=>’meter’, ’to\_unit\_symbol’=>’m’, ‘value’=>166.3]
* generateManyConversions($count, $difficulty = null, $unit = null): vygeneruje taký počet príkladov, ako sa nastaví parameter $count. Metóda vráti tieto príklady ako pole, ktorého prvky budú mať rovnaký tvar, aký vracia metóda generate(…)
* checkConversion($id\_phys\_quantity, $from\_unit, $value, $result\_to\_check): skontroluje, či je správne premenená jednotka z $from\_unit na základnú jednotku. Metóda vracia true/false.
* getBaseFormatNumberHint($number): metóda vráti základný tvar čísla. Napríklad 166.3 vráti 1.66, alebo z 0.017 vráti 1.7

getExp10Hint($id\_phys\_quantity, $from\_unit, $value): metóda vracia pomôcku na výsledok prevodu jednotiek v tvare mocniny 10. Napr. 0.003dm=[…]\*10^**-4**

1. Záver z testovania

# Priebeh tetovania č. 1

## Registrácia a prihlásenie učiteľa

Na hlavnej stránke zvolíme registráciu nového užívateľa a zaregistrujeme nového učiteľa. Vyplníme všetky údaje registračného formulára s tým, že najprv vyplníme políčka chybne, aby sme overili, či naozaj náš systém kontroluje jednotlivé vstupy:

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Odoslanie nevyplneného formuláru | áno |
| Zadanie krátkych vstupov (meno, heslo) | áno |
| Zadaný email v zlom formáte | áno |
| Správne vyplnené údaje a zvolený typ užívateľa učiteľ (registrácia úspešná, presmerovanie na hlavnú stránku) | áno |

Prihlasovací formulár vyplníme chybne, aby sme overili, či naozaj náš systém kontroluje jednotlivé vstupy:

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Odoslanie nevyplneného formuláru | áno |
| Zadané chybné meno | áno |
| Zadané chybné heslo | áno |
| Správne zadané prihlasovacie údaje (prihlásenie, presmerovanie na úvodnú stránku učiteľa) | áno |

## Vytvorenie skupiny

V administratívnej časti učiteľa vyberieme možnosť „vytvoriť novú skupinu“.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Chybne vyplnené údaje o novej skupine | áno |
| Správne zadané údaje (novovytvorená skupina v zozname) | áno |

## Registrácia a prihlásenie žiaka

Postupujeme rovnako ako pri registrácii a prihlasovaní učiteľa.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Test na kontrolu zadania už existujúceho emailu v databáze | áno |
| Test zobrazenia vstupných políčok pre názov, heslo skupiny pri zadaní typu užívateľa žiak | áno |
| Kontrola overenia správne zadaných údajov o skupine | áno |
| Prihlásenie so správnymi údajmi (presmerovanie na úvodnú stránku žiaka) | áno |

## Generovanie a riešenie príkladov

Do systému sa prihlásime ako žiak. Vyberieme si možnosť riešiť príklady.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Test generovania príkladov (zobrazenie príkladov) | áno |
| Kontrola, či vygenerované príklady obsahujú 2-3 platné číslice a zodpovedajú ostatným požiadavkám | áno |
| Kontrola funkčnosti hintov a ich správnosť | áno |
| Kontrola dopĺňania výsledku pri zadávaní základného tvaru čísla | nie, na serveri sa 0 neprepisuje zadaným číslom |
| Test kontroly správnych výsledkov | áno |
| Opakovanie testovacieho scenáru viac krát | nie, viď neúspešný test vyžšie |

## Administrácia skupiny

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Zmazanie žiaka zo skupiny, pokus o jeho prihlásenie | áno |
| Zmazanie celej skupiny, pokus o prihlásenie žiakov do nej patriacich | áno |

## Prehľad riešení žiaka

Do systému sa prihlásime ako žiak a vyrátame zopár príkladov. Zapíšeme si ich výsledky, aby sme mohli skontrolovať, či systém generuje správne tie isté.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Skontrolujeme naše údaje s tými, ktoré nám vygeneruje systém. | áno |
| Kontrola riešení cez učiteľa | áno |

## Zadávanie päťminútovky

Prihlásime sa ako učiteľ, vyberieme si skupinu a odštartujeme päťminútovku kliknutím na tlačidlo „zadať päťminútovku“.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Zadáme počet príkladov, obtiažnosť a odštartujeme päťminútovku. | áno |
| Riešenie päťminútovky ako žiak | nie, pri zadaní obtiažnosti dva sa žiakovi päťminútovka nezobrazí |
| Kontrola počtu a obtiažnosti vygenerovaných príkladov | áno (pre obtiažnosť 1) |
| Kontrola riešení | áno (pre obtiažnosť 1) |
| Zastavenie päťminútovky učiteľom | áno |
| Ako žiak skontrolujeme či je možné riešenia odoslať | áno (pre obtiažnosť 1) |

# Priebeh tetovania č. 2

Všetky úspešné testy z predchádzajúceho testovania boli opäť úspešné a preto ich výsledky už nebudeme uvádzať.

## Generovanie a riešenie príkladov

Do systému sa prihlásime ako žiak. Vyberieme si možnosť riešiť príklady.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Test generovania príkladov (zobrazenie príkladov) | áno |
| Kontrola, či vygenerované príklady obsahujú 2-3 platné číslice a zodpovedajú ostatným požiadavkám | áno |
| Kontrola funkčnosti hintov a ich správnosť | áno |
| Kontrola dopĺňania výsledku pri zadávaní základného tvaru čísla | áno |
| Test kontroly správnych výsledkov | áno |
| Opakovanie testovacieho scenáru viac krát | áno |

## Zadávanie päťminútovky

Prihlásime sa ako učiteľ, vyberieme si skupinu a odštartujeme päťminútovku kliknutím na tlačidlo „zadať päťminútovku“.

|  |  |
| --- | --- |
| Popis testu | Úspešný |
| Zadáme počet príkladov, obtiažnosť a odštartujeme päťminútovku. | áno |
| Riešenie päťminútovky ako žiak (obtiažnosť 1 aj 2) | áno |
| Kontrola počtu a obtiažnosti vygenerovaných príkladov | áno |
| Kontrola riešení | áno |
| Zastavenie päťminútovky učiteľom | áno |
| Ako žiak skontrolujeme či je možné riešenia odoslať | áno |

1. Zhodnotenia diela

# Jakub Chlup

Našu tímovú prácu hodnotím veľmi pozitívne, najmä vďaka dobrej komunikácií a organizovanosti tímu. Všetci sme si plnili svoje úlohy, na ktorých sme sa dohodli na čas, navzájom sme si pri ich plnení pomáhali a diskutovali o riešeniach problémov.

Komunikácia prebiehala najme na spoločných stretnutia, ale taktiež po prednáškach v škole a samozrejme aj na facebooku. Na stretnutiach sme vždy preberali čo treba spraviť, prípadne dokončiť či upraviť z predchádzajúcej etapy. Následne sme sa dohodli čo kto spraví, každý člen si viac menej, s pár kompromismi, vybral čo chce spraviť. Myslím, že aj preto sme dokázali efektívne pracovať a postupovať s projektom. Rozdelenie práce bolo rovnomerné ako pri návrhu tak aj pri implementácií projektu, takže sa nedá povedať, že by niektorý člen tímu pracoval viac ako ostatní.

S výsledkom projektu som spokojný a najmä na jeho priaznivé prijatie zadávateľom, ktorému sme splnili jeho očakávania. Výsledný systém mu uľahčí testovanie žiakov, aj samotným žiakom pomôže pri učení, a dúfam že po odskúšaní ho zadávateľ posunie ďalším učiteľom. Pokračovanie projektu by som videl v dotiahnutí drobností, ako napríklad pridanie ďalších grafov pre učiteľa pre lepší prehľad o žiakoch.

Tímový projekt bola skvelá skúsenosť do budúcnosti, myslím že všetci sme sa naučili skvele spolupracovať a komunikovať a tak spoločne dotiahnuť projekt do konca. Veľa som sa naučil o postupe a tvorbe informačného systému, taktiež som sa oboznámil s Nette Frameworkom, ktorý sme použili pri tvorbe systému.

# Ján Vodila

V prvom rade musím pochváliť veľmi dobrú organizovanosť a rozdelenie úloh v našom tíme. Vždy sme sa vedeli dohodnúť na tom, kto, ktorú časť práce urobí. Stretnutia tak prebiehali veľmi rýchlo, kde sme sa dohodli kto čo urobí najbližší týždeň, prešli sme si úlohy, ktoré sme mali spraviť v danom týždni a nikdy tak nebol problém stihnúť prácu do najbližšieho deadline-u, teda stanovený plán sme dokázali vďaka tomu úspešne napĺňať, aj keď pár drobností sme ešte sem tam doťahovali aj neskôr, nejednalo sa však o veľké zmeny či už v dokumentoch alebo v samotnom projekte.

Pri implementácii samotného návrhu sme sa snažili aby každý člen tímu urobil nejakú časť a tak sa nestalo, že niekto robil viac ako druhý. Z toho vyplýva, že na projekte sa každý podieľal rovnakým dielom, teda nikto sa nemôže sťažovať, že robil viac alebo menej ako niekto iný.

Počas konzultácii so školiteľom a pri prezentovaní diela zadávateľovi mal väčšinou slovo ten, kto danú časť práce, o ktorej bola práve reč, robil. Teda nemôžem povedať, že v tíme bol nejaký líder, ktorý to riadil, skôr sme si všetci boli rovnocenní. Podľa mňa, práve vďaka tomu, že sme si všetci boli seberovní, nevznikali tak žiadne spory a iné nežiadúce vplyvy, ktoré by mohli ovplyvniť efektívnosť a súdržnosť tímu.

Čo sa týka výsledného projektu, podľa reakcie zadávateľa môžem povedať, že jeho očakávania sme naplnili. Systém je vyriešený čo najjednoduchšie, objektovo pomocou MVC architektúry, takže prípadné zmeny by nemali byť žiadnym problémom, keďže stačí len pridávať nové metódy do existujúcich tried. Odporúčal by som do budúcna sa sústrediť na detaily, aby bol systém viac atraktívny a ponúkal viac doplnkových funkcií, ktoré by mohli zjednodušiť prácu učiteľovi ale aj žiakom. Jediným zádrhelom je potreba ovládať Nette Framework, na ktorom je naše dielo postavené.

Aby som to zhrnul, s naším projektom som nadmieru spokojný a chválim zodpovednosť každého člena tímu, že aj napriek práci a školským povinnostiam si každý tú svoju čast urobil na čas a svedomito. Veľa veci som sa naučil, či už z hľadiska tvorby informačného systému so všetkým, čo k tomu patrí, ale aj vedieť pracovať v kolektíve a nepozerať tak len na seba ale brať ohľad aj na ostatných členov.