

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

**PLATFORMA PRE KOLABORATÍVNE VYUČOVANIE
MATEMATIKY HEJNÉHO METÓDOU**

Diplomová práca

Bratislava, 2018

Bc. Daniel Linhart

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

PLATFORMA PRE KOLABORATÍVNE VYUČOVANIE
MATEMATIKY HEJNÉHO METÓDOU

Diplomová práca

Študijný program:	Aplikovaná informatika
Študijný odbor:	2511 Aplikovaná informatika
Školiace pracovisko:	Katedra aplikovanej informatiky
Školiteľ:	RNDr. Peter Borovanský, PhD.

Bratislava, 2018

Bc. Daniel Linhart

Zadanie prace

Čestné vyhlásenie

Čestne vyhlasujem, že som túto diplomovú prácu vypracoval samostatne pod vedením vedúceho diplomovej práce, s použitím uvedenej literatúry a zdrojov dostupných na internete.

V Bratislave, dňa xx.xx.20xx

Meno a priezvisko

Pod'akovanie

Abstrakt

LINHART, Daniel: *Platforma pre kolaboratívne vyučovanie matematiky Hejného metódou (Diplomová práca)* – Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky; Katedra aplikovanej informatiky. – Školiteľ: RNDr. Peter Borovanský, PhD.: FMFI UK, 2018

Kľúčové slová:

Abstract

LINHART, Daniel: *The Hejny method educational software – Adding triangles (Bachelor thesis)* - Comenius University in Bratislava, Faculty of mathematics, physics and informatics; Department of applied informatics - Adviser: RNDr. Peter Borovanský, PhD.: FMFI UK, 2016

Key words:

Obsah

Úvod	10
1 Analýza	11
1.1 Podobné riešenia	11
1.2 Použité technológie	11
2 Návrh	14
2.1 Špecifikácia	14
2.1.1 Cieľ projektu	14
2.1.2 Cieľová skupina	14
2.2 Mobilná aplikácia – návrh	14
2.2.1 Návrh databázy	15
2.2.2 Návrh prihlásenie	18
2.2.3 Návrh loading	18
2.2.4 Návrh registrácia	20
2.2.5 Návrh výber triedy	21
2.2.6 Návrh zdieľať s	22
2.2.7 Návrh zdieľať – blackboard	23
2.3 Webová aplikácia - návrh	23
2.4 Nápady na dopracovanie a zlepšenie	25
3 Implementácia	26

4	Testovanie	27
	Záver	28
	Literatúra a internetové zdroje	29
	Prílohy	30

Zoznam obrázkov

Obrázok 1 Návrh komunikácie	15
Obrázok 2 Návrh DB – vrchol Classes	16
Obrázok 3 Návrh DB - vrchol Users.....	17
Obrázok 4 Návrh DB - vrchol shared screen	17
Obrázok 2 Prihlasovanie	18
Obrázok 3 Nahrávanie dát z DB	19
Obrázok 4 Kontrola internetového pripojenia.....	19
Obrázok 5 Registračný formulár s ukázkou chybného vstupu.....	20
Obrázok 6 Scéna kde si žiak volí triedu do ktorej chce vstúpiť. V spodnej časti môžeme vidieť možnosť pridania triedy	21

Úvod

Dovoľte mi aby som nadviazal na úvod z mojej bakalárskej práce, v ktorom spomínam ako je čím ďalej náročnejšie udržať pozornosť dieťaťa štandardnými metódami výuky. Preto som sa rozhodol vytvoriť mobilnú aplikáciu na podporu výuky matematiky Hejného metódou. Pri testovaní aplikácie sme sa presvedčili, že tento spôsob výuky je pre žiakov zaujímavý a oplatí sa v ňom pokračovať. Nakoľko sme ale objavili rozpor s Hejného metódou, rozhodli sme sa, že cieľom Diplomovej práce bude vytvoriť platformu pre kolaboratívnu výuku matematiky, v ktorej bude tento rozpor odstránený.

Diplomovou prácou chceme prispieť k používaniu tabletov. Na vyučovacích hodinách matematiky by si za pomoci našej aplikácie žiaci prvého stupňa základných škôl mohli precvičiť matematiku a zároveň sa podeliť o zaujímavé úlohy zo svojimi spolužiakmi. Zároveň do platformy pribudla aj rola učiteľa, ktorý môže žiakom zadať rôzne úlohy a zároveň kontrolovať ich aktivitu. Našou úlohou bude vytvoriť platformu, ktorá bude pre žiaka ale aj pre učiteľa zaujímavá a intuitívna.

1 Analýza

1.1 Podobné riešenia

1.2 Použité technológie

Základom pre realtime-movú platformu bolo vhodné zvoliť technológie. Z dôvodu multiplatformovosti musela technológia na prenos dát mať podporu tak ako webového rozhrania, tak aj mobilných rozhraní (android, iOS). Z tohoto dôvodu sa nám výber technológií značne zúžil. Rozhodovali sme sa medzi variantom (Node.js, SQL Databázou), ktorá bude napojená na aplikáciu za pomoci web servisov a technológiou Firebase od firmy Google. Technológia Firebase zvíťazila nakoľko do jej použitia nebolo potrebné nijak viac investovať, nakoľko všetkými parametrami vyhovovala našim požiadavkám. Dokonca behom používania mielo prekvapila a to ďalšou funkciou, ktorou bola autentifikácia.

1.2.1 Firebase

Firebase je webová a mobilná vývojárska platforma vyvinutá firmou Firebase, Inc. Firebase je Backend-as-a.Service (BaaS), vďaka čomu je firebase našim serverom, API aj databázou. Firebase má tieto komponenty.

- *Firebase Analytics*
- *Firebase Cloud Messaging*
- *Firebase Auth*
- *Firebase Database*
- *Firebase Storage*
- *Firebase Hosting*
- *Firebase Test Lab for Android*
- *Firebase Crash Reporting*

Tieto komponenty môžeme využívať jednotlivo ale aj ako jeden celok. V našej práci využívame Firebase Auth a Firebase Database.

Firebase Auth – je služba, ktorá dokáže autentizovať používateľov pomocou kódu na strane klienta. Táto služba ďalej podporuje poskytovateľov sociálnych služieb Google,

Twitter, Facebook a GitHub. Služba obsahuje aj správu používateľov, pomocou ktorej sa je možné prihlasovať prostredníctvom e-mailu a hesla uloženého v Firebase.

Firebase Database – je realtime databáza. Služba poskytuje rozhranie API umožňujúce synchronizáciu dát medzi klientmi a ukladaním dát na cloudových úložiskách spoločnosti Firebase. Databáza je dostupná pomocou rozhrania REST API a väzieb na pár javascriptových frameworkov (AngularJS, React). Spoločnosť dodáva aj klientske knižnice, ktoré umožňujú integráciu s aplikáciami Java, Objective-C, JavaScript, Android, iOS, Node.js, swift, Unity. Vývojári používajúci databázu v reálnom čase, majú možnosť zabezpečiť si svoje dáta radou bezpečnostných nastavení na strane servera.

1.2.2 Unity 3D

Unity 3D je multiplatformový herný engine, ktorý bol vytvorený firmou Unity Technologies. Táto platforma sa používa na vývoj hier pre počítače, konzoly, VR/AR, mobilné zariadenia. Prvoplánovo bola platforma vyvinutá pre Apple, no časom sa rozšírila o ďalšie platformy. Unity 3D je zaujímavé hlavne vďaka podpore mutlplatformosti. Platformy, ktoré Unity podporuje sú Windows, OSX, Android, iOS, BlackBerry, Windows Phone 8 a platformy herných konzol.

1.2.3 C#

C# (v angličtine c sharp – výslovnosť sí šarp) je objektovo – orientovaný programovací jazyk. Tento jazyk bol vyvinutý spoločnosťou Microsoft. Základom nového jazyka C# sa stali jazyky Java a C++. Jazyk C# bol vyvíjaný s úmyslom spojiť silu jazyka C++ s možnosťou rýchleho programovania.

1.2.4 Nette Framework

Nette Framework je open source framework na tvorbu webových stránok v PHP. Framework je zameraný na eliminovanie bezpečnostných hrozieb. Podporuje AJAX, DRY, KISS, MVC a znovu použiteľnosť kódu. Nette je z väčšej časti postavený na používaní komponent a využíva udalosťami riadené programovanie.

1.2.5 jQuery

jQuery je jednoduchšia javascriptová knižnica, riadiaca sa pravidlom ("write less, do more"). Účelom knižnice je uľahčiť používanie JS na našich web stránkach. S jQuery dokážeme nahradiť veľa riadkov kódu volaním na jeden riadok s identickou funkcionalitou. jQuery taktiež dokáže zjednodušiť komplikované AJAXové volania alebo manipuláciu s DOM. Knižnica jQuery obsahuje:

- *Manipuláciu s DOM/HTML*
- *Manipuláciu s CSS*
- *Html udalosti*
- *Efekty a animácie*
- *AJAX*

2 Návrh

2.1 Špecifikácia

2.1.1 Cieľ projektu

Cieľom práce je vytvoriť platformu pre vyučovanie matematiky Hejného metódou, pomocou ktorej budú môcť žiaci spolupracovať. Platforma používateľom umožní zdieľať úlohy, ktoré im vygeneruje aplikácia ale aj tie, ktoré si sami vytvoria. V našej aplikácii sa nezabudlo ani na učiteľov, pre ktorých sú pripravené funkcie ako sledovanie aktivity a zadávanie úloh žiakom.

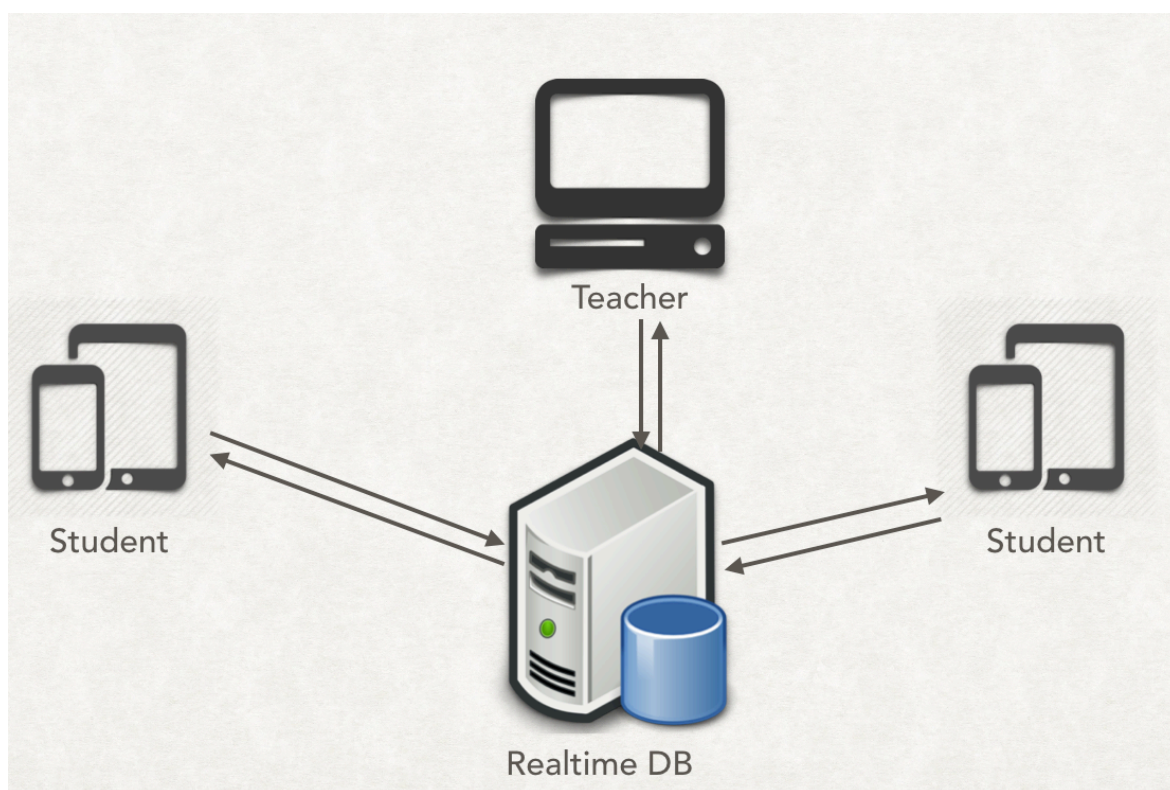
2.1.2 Cieľová skupina

Cieľovou skupinou sú všetci žiaci prvého stupňa základných škôl a ich učitelia nezávisle od toho akým spôsobom sa na školách vyučuje. Nakoľko časť aplikácie je nezávislá od pripojenia na internet, môžu aplikáciu využívať aj deti v predškolskom veku.

2.2 Mobilná aplikácia – návrh

Je určená pre žiakov základných škôl. Po spustení budú mať žiaci na výber medzi online a offline verziou aplikácie. Po zvolení online verzie sa spraví overenie či používateľ je prihlásený alebo nie. Po tejto kontrole môžu nastať 3 scenáre. Prvým scenárom je, že žiak už je prihlásený. Vtedy sa stiahnu dáta prihláseného používateľa z DB a načíta sa nám scéna výberu triedy. V prípade, ak žiak nie je prihlásený zobrazí sa nám scéna prihlásiť a tretím scenárom je nezaregistrovaný používateľ. V tomto prípade sa zobrazí scéna prihlásiť a žiak musí vyplniť krátky registračný formulár. Po prihlásení/registrácii sa užívateľovi (žiakovi) zobrazí možnosť vstúpiť do triedy v prípade že sa už nachádza v nejakej z tried, ak sa žiak nenachádza v žiadnej triede má možnosť zadať heslo triedy a automaticky vstúpi do triedy. Každé heslo triedy sa automaticky žiakovi uloží do DB medzi jeho triedy. Po vstupe do triedy bude mať používateľ k dispozícii tú istú funkcionálnosť ako neprihlásený, ktorá je rozšírená o blackboard (tabuľu) a zdieľanie úloh. Po otvorení blackboard sa žiakovi zobrazia všetky úlohy, ktoré zadal učiteľ ale aj úlohy, ktoré si zadávajú žiaci medzi sebou.

Druhou hlavnou časťou našej práce je funkcia Shared with ... (Zdieľať s ...), cieľom tejto funkcionality je aby žiaci svoje úlohy riešili spolu (Jedna úloha, ktorú rieši niekoľko žiakov na viacerých tabletoch). Hlavným cieľom u tejto funkcie bolo ju navrhnuť tak aby sme neprenášali obrovské dáta, nakoľko táto funkcia musí mať podporu real time. Funkcie Shared with ... ako aj blackboard sú funkčné pre žiakov len z rovnakých tried. V modelovom prípade Janko je v triede 1.A a Marienka v 1.A títo žiaci môžu medzi sebou zdieľať úlohy ale aj si pozrieť tabuľu, ktorú majú spoločnú. Druhým prípadom je že Janko bude v inej triede ako Marienka. V tomto prípade sa žiakom medzi sebou spojenie nadviazať nepodarí, keďže cieľom je nadviazať komunikáciu v rámci tried a nie mimo nich.

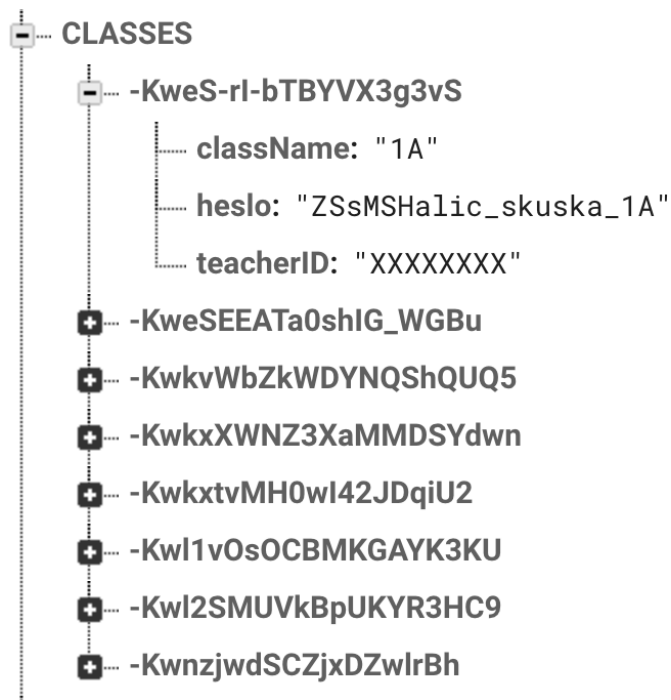


Obrázok 1 Návrh komunikácie

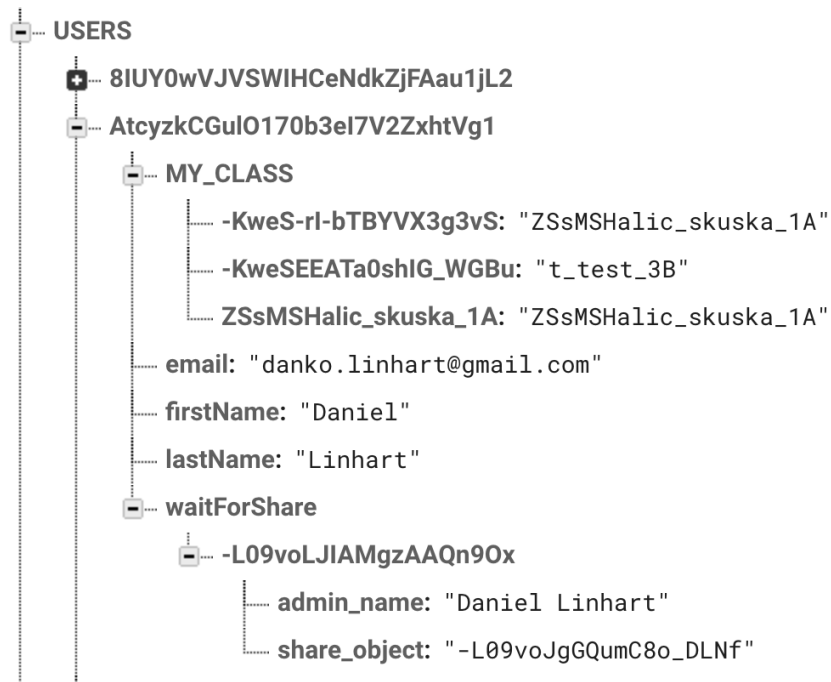
2.2.1 Návrh databázy

Nakoľko na našej platforme nepoužívame SQL databázu bol návrh databázy jednoduchší. Pri SQL databáze by sme museli mať všetky možné prepojenia tabuliek dopredu vymyslené a rovnako je to aj so stĺpcami tabuliek. Pri našej NoSql databáze na tom až tak nezáleží. Firebase DB nieje obyčajná databáza s tabuľkami. Táto databáza je jeden objekt JSON. Každú tabuľku reprezentuje vrchol databázy. Na začiatku databázy máme root. V našom

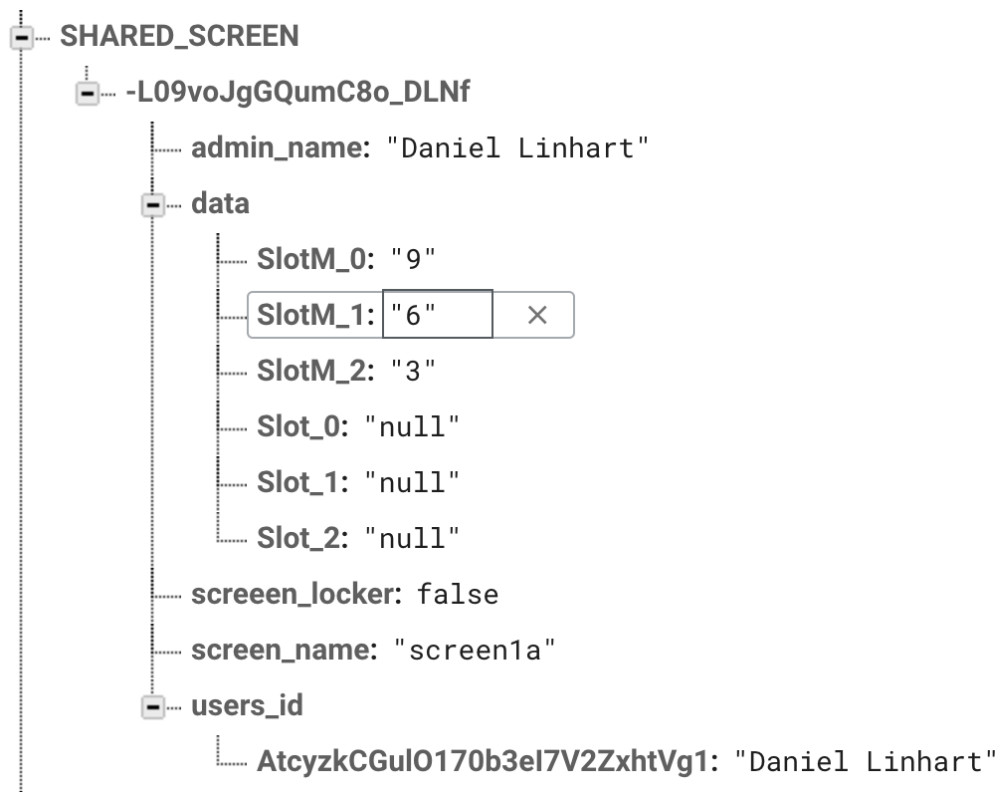
případe sa root vola sumrectangle, následne máme CLASSES, SHARED_SCREEN, USERS, EXAMS, každé z týchto „detí“ obsahuje zoznam id, ktoré reprezentujú deti daného rodičovského objektu. V prípade, že by niektorý z rodičovského objektu neobsahoval žiadne ID celý objekt sa zmaže. Z tohoto dôvodu je návrh jednoduchší databáza sa vytvára sama postupom používania a my len určujeme cestu, ktoré dáta sa majú do ktorého rodiča zapísať. Ako sme navrhli náš databázový model môžete vidieť na priložených obrázkoch.



Obrázok 2 Návrh DB – child Classes



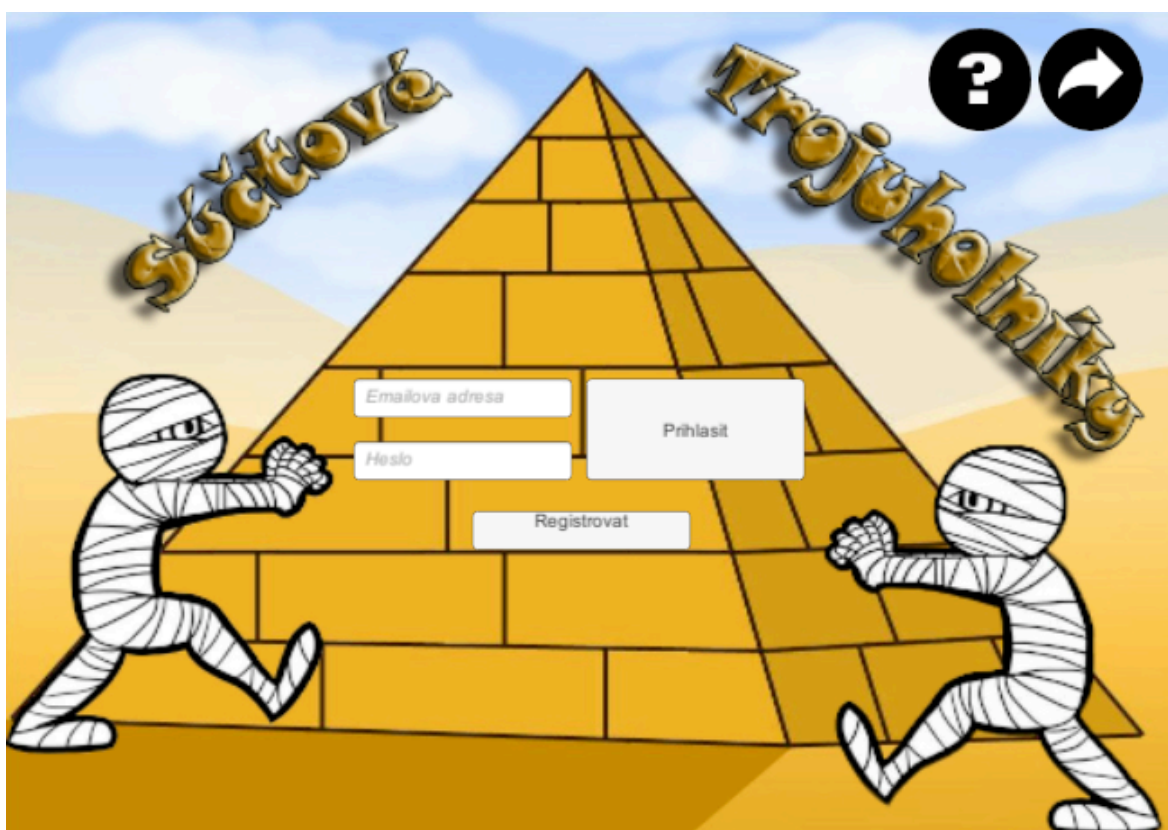
Obrázok 3 Návrh DB - child Users



Obrázok 4 Návrh DB - child shared screen

2.2.2 Návrh prihlásenie

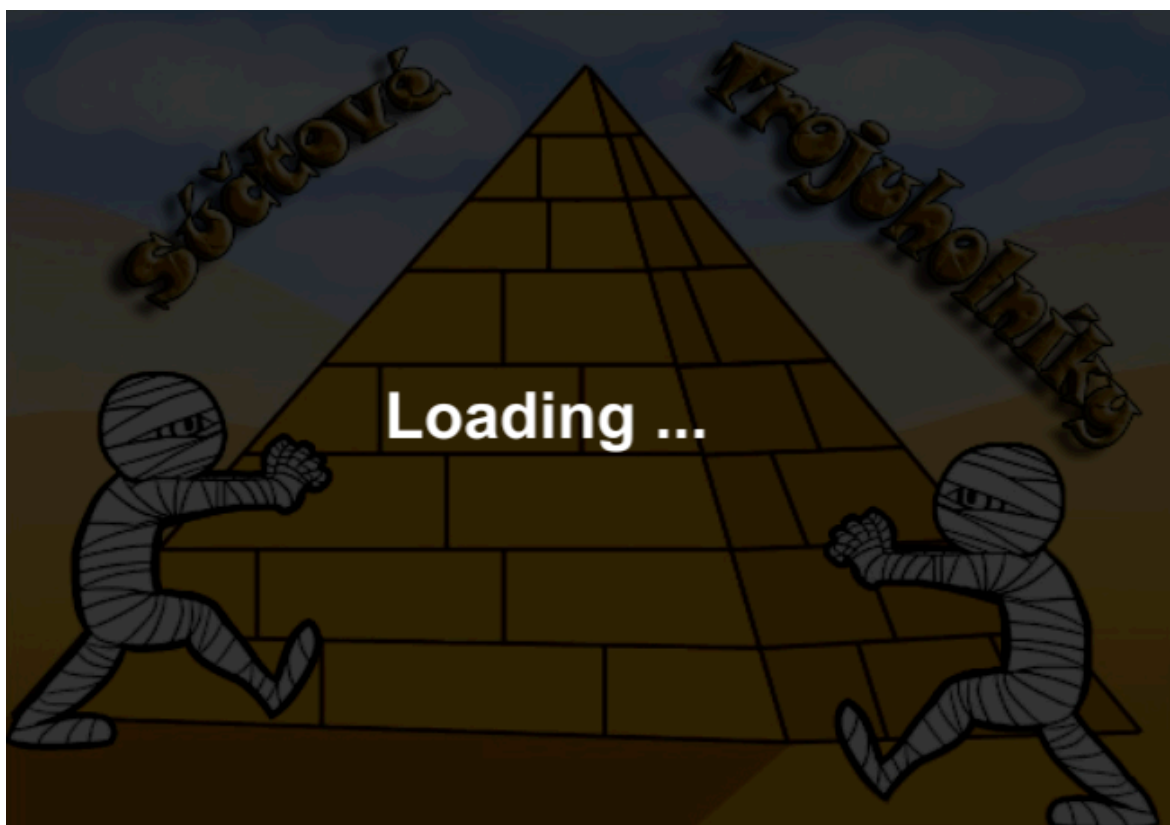
Používateľ sa prihlasuje pomocou emailu a hesla, ktoré si sám zvolí v prípade že aplikáciu používa prvý krát, žiak sa pred prihlásením musí zaregistrovať. Po úspešnom prihlásení používateľ bude presmerovaný na scénu s výberom triedy. Prihlásenie na zariadení trvá dovtedy pokiaľ sa žiak z aplikácie sám neodhlási alebo neprebehne aktualizácia aplikácie. V prípade, ak prihlasovanie z nejakej príčiny zlyhá bude mu príčina oznámená formou informatívneho výpisu.



Obrázok 5 Prihlasovanie

2.2.3 Návrh loading

Scéna loading bola navrhnutá z dôvodu doťahovania dát z databázy. Nakoľko databáza firebase disponuje iba s asynchrónnym volaniami bolo potrebné navrhnuť spôsob akým by sme zabránili postupu z jednej scény do druhej kým nemáme stiahnuté všetky potrebné dáta. Z tohoto dôvodu sme túto scénu navrhli. Nakoľko by ale bolo neustále doťahovanie dát otravné a obmedzujúce rozhodli sme sa po prihlásení stiahnuť všetky dáta potrebné k chodu našej aplikácie.



Obrázok 6 Nahrávanie dát z DB



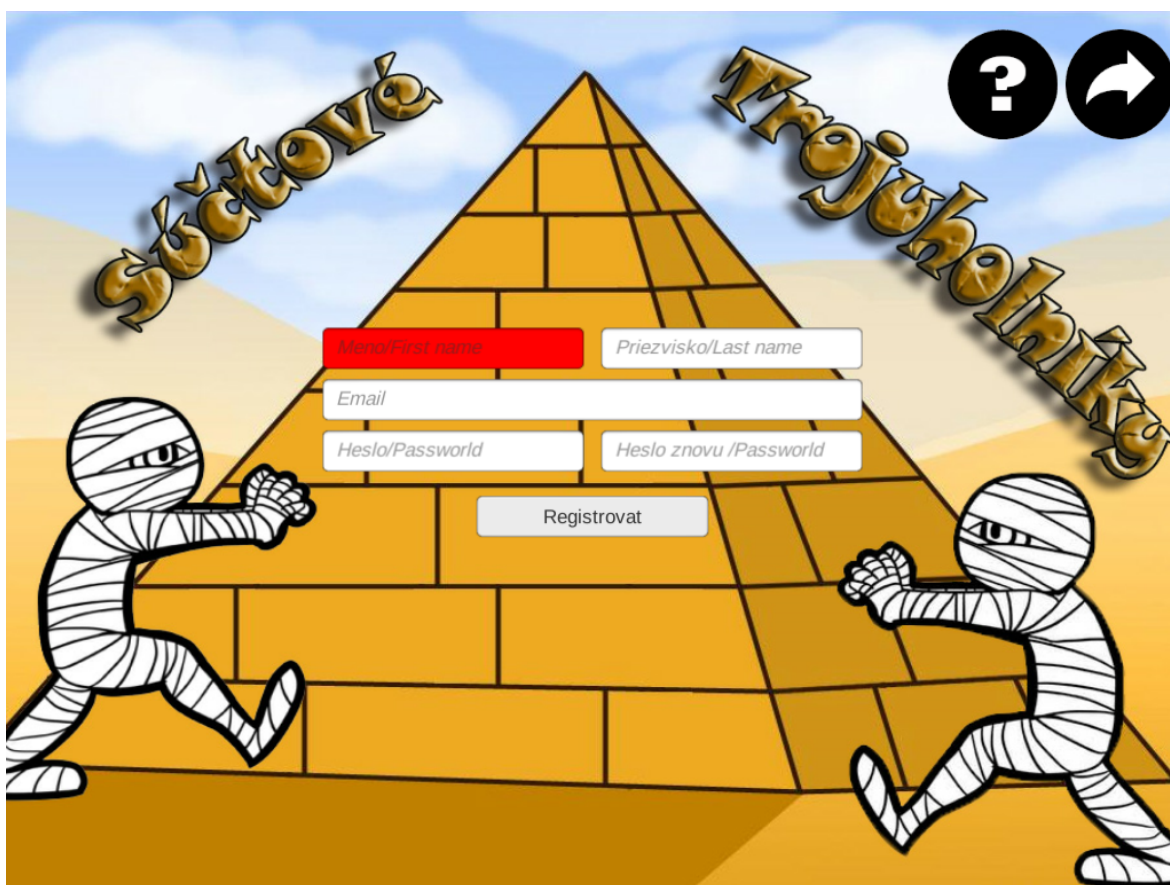
Obrázok 7 Kontrola internetového pripojenia

2.2.4 Návrh registrácia

Nezaregistrovanému používateľovi sa zobrazí krátky formulár ktorý obsahuje :

- Email
- Meno
- Priezvisko
- Heslo

Po vyplnení formulára a odoslaní prebehne registrácia a po úspešnom zaregistrovaní bude žiak automaticky presmerovaný na scénu s výberom tried. V prípade, že žiak nesplní niektorú z požiadaviek políčko sa rozsvieti na červeno spolu s textom v ktorom sú vypísané podmienky fiedlu.

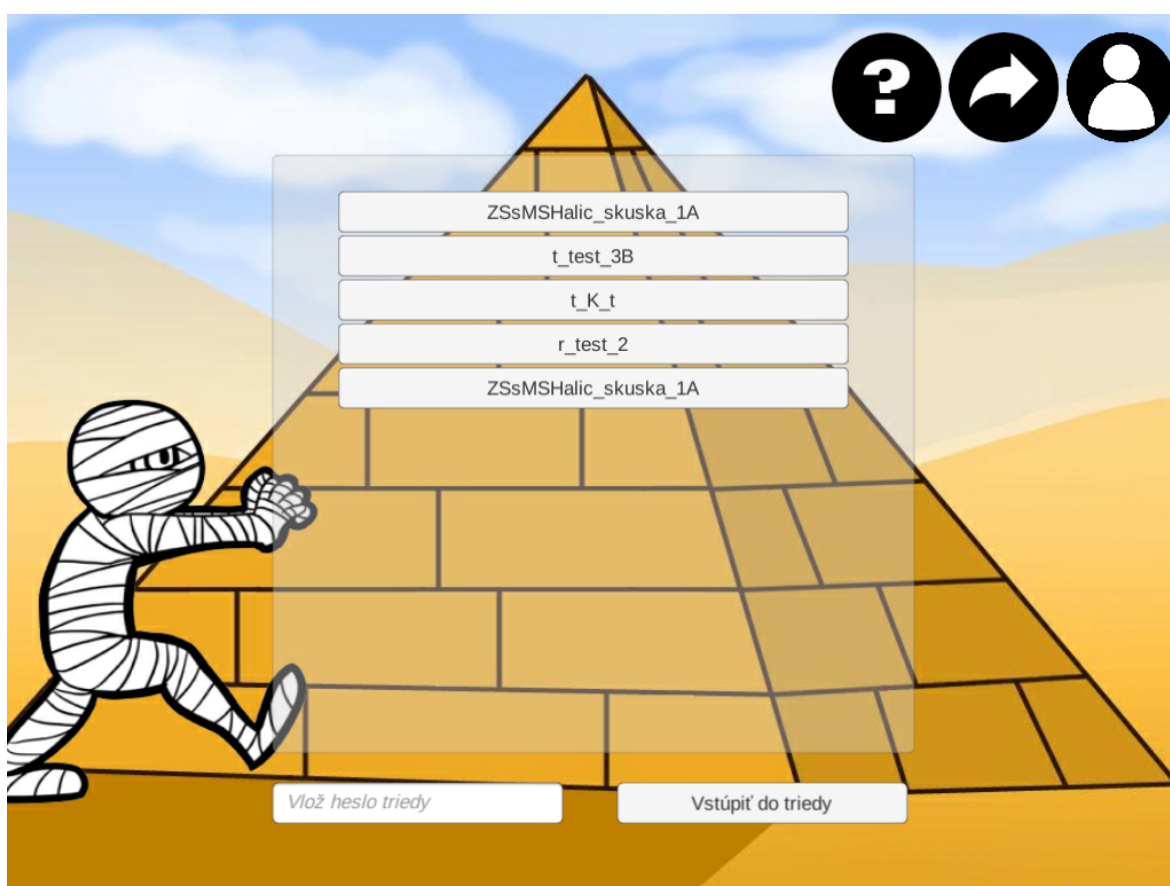


Obrázok 8 Registračný formulár s ukážkou chybného vstupu

2.2.5 Návrh výber triedy

Táto scéna bola navrhnutá ako 2v1 scéna funguje na zvolenie triedy kliknutím na názov triedy v scrollbare v ktorom bude zoznam tried vypísaný. Na tomto objekte je zavesený handler, ktorý kontroluje či si žiak nepridal ďalšiu triedu v prípade, že áno doplní nám triedu do nášho scrollbaru. Tento handler počúva aj na zmenu názvu triedy prípadne zmazaniu. V prípade, že by učiteľ triedu premenoval zrovna v momente, keď si žiak volí triedu automaticky by sa mu názov triedy zmenil, prípadne by trieda zmizla z ponuky.

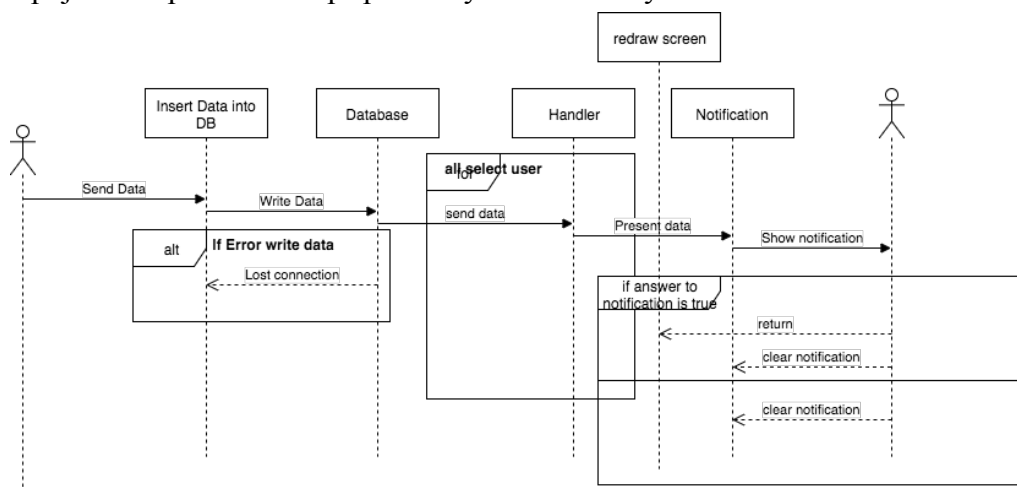
Druhou funkciou tejto scény je ukladanie tried. Funkcia funguje na základe hesla triedy. Žiak zadá heslo triedy, ktoré im učiteľ určí. Po zadaní hesla prebehne kontrola, či na základe hesla existuje takáto trieda, ak systém nájde danú triedu v databáze automaticky žiaka do tejto triedy priradí. V opačnom prípade notifikuje žiaka, že trieda s daným heslom neexistuje.



Obrázok 9 Scéna kde si žiak volí triedu do ktorej chce vstúpiť. V spodnej časti môžeme vidieť možnosť pridania triedy

2.2.6 Návrh zdieľať s ...

Jedna z hlavných funkcionalít platformy a aj najťažším orieškom na návrh. Ako som už spomínal bolo potrebné dodržať určité dôležité podmienky ako sú real time, malý obsah prenášaných dát, nadviazanie spojenia a princíp lock/unlock obrazovky. Posledný spomenutý problém bolo najproblematickejšie vyriešiť nakoľko sa nám naskytalo niekoľko možností ako ísť na tento problém. Prvou možnosťou bolo lockovanie obrazoviek všetkým neaktívnym používateľom. Čiže používateľ s označeným objektom by zablokoval prístup ostatným. V tomto spôsobe bol problém ten, že vždy by sa našiel žiak, ktorý by lockoval screen ostatným a znechutil im výuku. Druhou metódou bola rozšírená prvá metóda o časovač. Tento návrh bol založený na princípe klientskych časovačov, kde by sa spustil časovač pri označení objektu. V prípade, že by žiak do určitého času neumiestnil objekt bol by objekt od značený a obrazovka odblokovaná. Táto metóda bola lepšia, avšak vysvetliť deťom prečo sa im od značil objekt by bolo problematické. Metóda úplne neriešila blokovanie ostatných. Ďalšou metódou, ktorá prichádzala do úvahy bol spôsob kto prv príde ten prv mele. Princíp bol vtom, že všetci mohli označiť ten rovnaký objekt, avšak všetkým sa objekt od značil po tom ako ho ten najrýchlejší umiestnil na novú pozíciu. Tento spôsob sa ukázal ako najvhodnejší na použitie. Či to bolo správne rozhodnutie sa uvidí pri prvom testovaní. Nadviazanie spojenia je zabezpečené pomocou handllera, ktorý počúva na vrchole "waitForShare". V tom momente ako nastane zmena v tomto vrchole zobrazí sa žiadosť o nadviazanie spojenia. V momente potvrdenia žiadosti sa zosynchronizuje obrazovka s navrhovateľom. Týmto momentom je vytvorené spojenie medzi používateľmi až do odpojenia sa používateľa prípadne vyriešenia úlohy.



Obrázok 10 Sekvenčný diagram nadviazania spojenia

2.2.7 Návrh zdieľať – blackboard

Časť blackboard je ďalšou dôležitou funkciou našej platformy. Funkcia share with priamo spájala žiakov a nútila ich spolupracovať. Táto funkcia umožňuje podeliť sa so zaujímavými príkladmi v rámci triedy. Takže, ak žiak vymyslí náročný alebo zaujímavý príklad, prípadne mu ho vygeneruje generátor úloh bude sa môcť sním jednoducho podeliť zo svojimi spolužiakmi. V tejto časti bolo dôležité vhodne navrhnuť databázu nakoľko sa úlohy môžu opakovať. V rámci databázy ale aj tabule už priamo v programe by tým pádom vznikali zbytočné duplicity a tak by sme žiakov otravovali rovnakými úlohami zbytočne viackrát. Druhým dôležitým krokom je vyriešiť označovanie úloh za vyriešené. Po vyriešení úlohy sa žiakovi príklad už nebude zobrazovať. Po vyriešení týchto úvodných funkcií bude tabuľa fungovať takýmto scenárom. Žiak vytvorí príklad a pridá ho na tabuľu. V tom momente sa zobrazí na tabuli všetkým žiakom v rámci triedy.

O pridaní úloh na tabuľu nebudú žiakom chodiť notifikácie. Notifikácie by v tomto prípade mohli byť príliš časté a rušili by používateľov. Takto je na zvážení používateľa ako často si skontroluje tabuľu. Na tabuľu môže pridávať úlohy aj učiteľ zo svojho rozhrania.

(diagram, obrázok)

2.3 Webová aplikácia - návrh

Webová aplikácia je platforma navrhnutá výhradne pre učiteľov. Je to z dôvodu množstva funkcionality, ktorú bude táto tento web obsahovať. Pre túto technológiu sme sa rozhodli z dôvodu, že mobilná aplikácia je prispôbena žiakom. Preto sme sa rozhodli oddeliť detskú hravú časť platformy od tej pedagogickej časti avšak tieto 2 aplikácie sú prepojené spoločnou databázou. Webová časť obsahuje tieto funkcie:

- *Vytvorenie/editovanie triedy*
- *Pridanie úlohy na tabuľu*
- *Zobrazenie všetkých tried*
- *Zobrazenie žiakov v triedach*

- *Odobratie žiakov z tried*
- *Vytvorenie úlohy*
- *Zobrazenie štatistík žiaka*
- *Zobrazenie obrazovky žiaka*

2.3.1 Návrh vytvorenie/editovanie triedy

Z dôvodu, že aplikáciu môžu používať žiaci z rôznych tried a škôl rozhodli sme sa vytvoriť množinu tried. Tieto triedy slúžia učiteľom na začlenenie si žiakov do tried. Každý z učiteľov si vytvorí triedu. Na vytvorenie triedy má v svojom rozhraní pripravený krátky formulár. Po vyplnení formulára a odoslaní sa trieda automaticky zobrazí v DB a je prístupná žiakom. Žiaci na vstup do danej triedy potrebujú vedieť vstupný kľúč, ktorý je špecifický. Učiteľ môže názov a heslo triedy kedykoľvek editovať. Editovanie hesla a názvu žiakov neobmedzí. Žiaci nachádzajúci sa v triede nové heslo zadávať nemusia týkať sa to bude len nových žiakov.

2.3.2 Zobrazenie všetkých tried

Funkcia bude slúžiť používateľovi ako prehľad svojich vytvorených tried. Zároveň každá trieda obsahuje tlačidlá na editáciu a zmazanie triedy.

2.3.3 Zobrazenie žiakov v triedach

2.3.4 Odobratie žiakov z tried

2.3.5 Vytvorenie úlohy

2.3.6 Pridanie úlohy na tabuľu

2.3.7 Zobrazenie obrazovky žiaka

2.3.8 Zobrazenie štatistík žiaka

2.4 Nápady na dopracovanie a zlepšenie

3 Implementácia

4 Testovanie

Záver

Literatúra a internetové zdroje

- [1] Marek, DIRNER Alexander–DEMKO Jozef–DOMARACKÝ, FRANKO František–HLAVÁČOVÁ Júlia–MARTINSKÁ Gabriela, and MURÍN Pavol–Šerý Michal. "VIRTUÁLNA KOLABORÁCIA. HĽADANIE NOVÝCH FORIEM VZDELÁVANIA S VYUŽITÍM NOVÝCH IKT V ROZVOJI FYZIKÁLNEHO POVEDOMIA MLÁDEŽE."
- [2] Dirner, A., et al. "Virtuálna kolaborácia. Využitie nových informačno-komunikačných technológií vo výučbe fyziky."
- [3] DEPEŠOVÁ, Jana. "Virtual communication in educational system." Journal of technology and information (2013).
- [4] Dirner, Alexander, et al. "2. Virtuálna kolaborácia."
- [5] Firebase Realtime Database [online].[19.5.2017] Dostupné na <https://firebase.google.com/docs/database/>
- [6] Unity User Manual (5.6) [online].[19.5.2017] Dostupné na <https://docs.unity3d.com/Manual/index.html>

Prílohy