

STREDNÁ PRIEMYSELNÁ ŠKOLA ELEKTROTECHNICKÁ  
PREŠOV

IV.SB - 3, 5

ŠK. R. 2022 – 2023

PČOZ – HYBRIDNÝ  
PROJEKT – SYSTÉM NA  
SPRÁVU UČEBNÝCH  
MATERIÁLOV ŠKOLY

Martin Demčák  
Marek Hrabčák

Konzultant: Ing. Peter Vargovčík

## **Anotácia v slovenskom jazyku**

Táto práca slúži ako centrálne úložisko materiálov pre študentov a učiteľov školy. Poskytuje priestor pre nahrávanie súborov, ich sťahovanie a mazanie. Taktiež tento systém vie spravovať tieto materiály a nastavovať prístup k materiálom podľa roly užívateľa.

Systém pracuje s databázou, kde sa dá nahrať celá hierarchia školy, učitelia, študenti, triedy, predmety, materiály a pod. Za ich správu je zodpovedný školský administrátor s prislúchajúcimi prístupovými právami. Aplikácie sú užívateľsky prívetivé a intuitívne. Databáza je zdieľaná medzi všetkými podporovanými aplikáciami, je vytvorená v jazyku MySQL vo vývojovom prostredí MySQL Workbench a nahratá na XAMPP serveri pomocou webovej aplikácie pre správu databáz phpMyAdmin.

Systém pozostáva z troch aplikácií (webová, desktopová a mobilná). Webová aplikácia je vytvorená pomocou HTML, CSS, JavaScript a PHP v aplikácii PhpStorm a poskytuje možnosť úplnej správy systému administrátorom. Spolu s desktopovou aplikáciou umožňuje prihlásenie sa učiteľov, ktorí môžu spravovať nimi vytvorené materiály pre študentov, ktorých učia a prihlásenie sa študentov, ktorí si tieto materiály vedia prezeráť a sťahovať. Každý učiteľ a študent má prístupný svoj vlastný ToDoList, kde si môže zaznamenávať svoje povinnosti.

Desktopová aplikácia je zakonštruovaná v jazyku Java pre Model a Controller a FXML pre View. Ponúka možnosť kompletnej správy školy.

Nakoniec mobilná aplikácia, zstrojená vo vývojovom prostredí Android Studio, v jazyku Java pre Model a Controller a XML pre View, umožňuje prihlásenie sa študentom, prezeranie si vytvorených štruktúr ich učiteľmi a správu vlastného ToDoListu.

## **Anotácia v anglickom jazyku**

This work serves as a central storage of materials intended for the school's students and teachers. It provides a space for uploading, downloading and deletion of files. Likewise, this system can manage these materials and adjust permissions to the materials depending on the role of the user.

The system works with a database, which is a place for uploading information about the entire school hierarchy, teachers, students, classes, subjects, etc. The school administrator with matching permissions is responsible for their management. The application is user-friendly and intuitive. The database is shared between all of the supported applications, written in the MySQL language inside a development environment called MySQL Workbench and uploaded to a XAMPP server with the use of a web application for database management called phpMyAdmin.

The system consists of three applications (web, desktop and mobile). The web application was made with the use of HTML, CSS, JavaScript and PHP with the use of an application named PhpStorm, and provides the full capability of system management for the site administrator. Together with the desktop application it allows for logging in of the teachers, who can manage the materials intended for their students, and the logging in of the students, who can view and download said materials. Every student and teacher has access to their ToDoList where they can note down their tasks.

The desktop application was made in Java, the Model and Controller part, and FXML, for the View part. It provides the full ability of school management.

Lastly, the mobile application was made in the development environment of Android Studio, in the languages Java, for Model and Controller, and XML, for View, and it provides the ability to log in for the students, while also being able to look through the learning structures created for them by their teachers, while also being able to manage their own ToDoList.

## **Čestné vyhlásenie**

Vyhlasujeme, že celú prácu ročníkového projektu na tému „Hybridný projekt – Systém na správu učebných materiálov školy“ sme vypracovali samostatne, s použitím uvedenej literatúry.

Sme si vedomí zákonných dôsledkov, ak v nej uvedené údaje nie sú pravdivé.

Prešov, 09. máj 2023

.....

*Martin Demčák*

.....

*Marek Hrabčák*

## **Pod'akovanie**

Týmto by sme chceli vyjadriť pod'akovanie konzultantom Ing. Petrovi Vargovčíkovi a Ing. Márii Hedvigovej za ich odbornú pomoc pri vytváraní tejto práce.

# Obsah

<b>Úvod .....</b>	<b>6</b>
<b>1 Ciel' a Metodika práce.....</b>	<b>7</b>
<b>2 Teoretická časť – Úvod do problematiky .....</b>	<b>10</b>
2.1    Mobilná aplikácia .....	10
2.2    Android Studio.....	10
2.3    Java .....	10
2.4    XML.....	10
2.5    IntelliJ Idea .....	11
2.6    JavaFX .....	11
2.7    FXML .....	11
2.8    MySQL .....	11
2.9    SQL .....	12
2.10    Visual studio Code .....	12
2.11    Sublime Text .....	12
2.12    HTML.....	12
2.13    CSS.....	13
2.14    PHP.....	13
2.15    Javascript.....	13
2.16    phpMyAdmin .....	14
2.17    XAMPP .....	14
2.18    Figma.....	14
2.19    PhpStorm .....	14
<b>3 Praktická časť .....</b>	<b>15</b>
3.1    Databáza.....	15
3.2    Desktopová aplikácia .....	16
3.2.1    Trieda Main .....	16
3.2.2    Model.....	18
3.2.2.1    Trieda Data .....	19
3.2.2.2    Trieda DatabaseConnection .....	20
3.2.2.3    Ostatné triedy modelu .....	21
3.2.3    Controller.....	22
3.2.4    View a rýchly návod používania aplikácie (ukážka funkcia.....	23
3.2.4.1    Prihlásovací panel .....	23
3.2.4.2    Registrácia školy .....	24
3.2.4.3    Zabudnuté heslo .....	24
3.2.4.4    Prihlásenie .....	27
3.2.4.5    Administrátorský panel .....	28
3.2.4.6    Učiteľský panel .....	30
3.2.4.7    Študentský panel .....	33
3.3    Mobilná aplikácia .....	35
3.3.1    Súborová štruktúra.....	35
3.3.2    Sekcia Login .....	36
3.3.3    Sekcia Model .....	40
3.3.4    Sekcia Adapter.....	44
3.3.5    View a rýchly návod používania aplikácie (ukážka funkcia.....	45

3.3.5.1	Prihlasovací panel .....	45
3.3.5.2	Zabudnuté heslo .....	46
3.3.5.3	Prihlásenie .....	48
3.3.5.4	Študentský panel .....	48
3.3.5.5	Karta Menu.....	50
3.3.5.6	Karta ToDoList .....	51
3.4	Webová aplikácia.....	53
3.4.1	Súbor ConnectToDB.php .....	53
3.4.2	Súbor SelectDataFunction.php .....	53
3.4.3	Súbor PrintDataFunction.php .....	54
3.4.4	Súbor Využitie funkcie AJAX.....	55
3.4.5	FilesLogic.php .....	56
3.4.6	Trieda PHPMailer.....	57
3.4.7	View a rýchly návod používania aplikácie (ukážka funkcia funkcia) .....	58
3.4.7.1	Prihlasovací panel .....	58
3.4.7.2	Registrácia školy .....	58
3.4.7.3	Zabudnuté heslo .....	60
3.4.7.4	Prihlásenie .....	61
3.4.7.5	Administrátorský panel systému .....	61
3.4.7.6	Administrátorský panel školy.....	64
3.4.7.7	Učiteľský panel .....	64
3.4.7.8	Študentský panel .....	65
<b>4</b>	<b>Výsledky práce a diskusia.....</b>	<b>66</b>
<b>5</b>	<b>Závery práce .....</b>	<b>67</b>
<b>Zhrnutie.....</b>		<b>68</b>
<b>Resumé.....</b>		<b>69</b>
<b>Zoznam použitej literatúry .....</b>		<b>70</b>
<b>Zoznam obrázkov .....</b>		<b>73</b>

# Úvod

Rozhodli sme sa vytvoriť túto aplikáciu preto, lebo na našej škole máme veľa platfóriem, odkiaľ čerpáme materiály pre štúdium. Učitelia nám posielajú materiály mailom, cez MS Teams, stránku školy a EduPage. Pri takomto množstve platfóriem sa ľahko sleduje, kde práve prišli materiály a často krát sa stávalo, že sme si nevšimli, že materiály boli poslané.

Naša aplikácia by teda mala riešiť tento problém a centralizovať materiály na jedno miesto, aby študenti nemuseli hľadať materiály po rôznych platformách pretože to zabera veľa času a energie.

Vymysleli sme systém, ktorý sme distribuovali na troch platformách a to Windows, Android a web. Všetky tri distribúcie sú napojené na jednu a tú istú databázu, čiže sú synchronizované. Každý užívateľ si môže vybrať platformu, ktorá mu najviac vyhovuje a čerpať materiály odtiaľ.

V našej aplikácii je možné zaregistrovať školu, nahrať rozdelenie tried, učiteľov, študentov a nahrávať materiály.

Aplikácia má štyri typy účtov, a to administrátor, administrátor pre danú školu, učiteľ a študent. Administrátor má prístup k administrátorskému panelu, odkiaľ vie spravovať celú databázu. Školský administrátor vie spravovať celú databázu konkrétnej školy, učiteľ vie vytvárať témy a podtémy pre študentov a nahrávať im tam materiály. Študenti si vedia tieto materiály stahovať. Učiteľský účet má navyše možnosť skryť danú tému, podtému alebo súbor, aby ho videl iba on.

Snažili sme sa túto aplikáciu navrhnúť tak, aby bola ľahko ovládateľná a jednoduchá, aby sa v nej užívateľ cítil dobre a dostal sa kam potrebuje na pár klikov.

V dokumentácii je popísaná celá tvorba tohto systému, jeho ciele a metodika práce.

# 1 Ciel' a Metodika práce

Cieľom tejto práce je vytvoriť funkčnú webovú, mobilnú a desktopovú aplikáciu, ktorá by mala slúžiť pre školy na nahrávanie a správu materiálov pre študentov.

Hlavným cieľom je vytvoriť efektívny, jednoduchý, užívateľsky prívetivý systém, aby užívateľ nemal problém zorientovať sa v aplikácii a dostať sa k materiálom na čo najmenší počet klikov.

Na začiatok sme si museli podrobne premysliť a vytvoriť štruktúru databázy, ktorá je nosným pilierom nášho systému. Vytvárali sme ju v jazyku SQL v aplikácii MySQL Workbench. Pri jej vytváraní sme použili naše poznatky, ktoré sme nadobudli počas štúdia a návody na internete, ktoré nám pomohli s tým, čo sme nevedeli.

Webová aplikácia je vytvorená v aplikácii Sublime Text, Visual Studio Code a PhpStorm. S týmito programami sme už pracovali v minulosti, čiže základné prvky sme už ovládali. Stačilo sa naučiť, ako pridať pár pluginov a mohli sme pracovať. Ak sme si s niečím nevedeli rady, hľadali sme to na internete. Webovú aplikáciu sme vytvorili pomocou jazyka HTML, CSS, JavaScript (konkrétnie framework jQuery), PHP a MySQL.

Desktopová aplikácia je vytvorená v aplikácii IntelliJ Idea. S touto aplikáciou pracujeme od prvého ročníka na tejto škole, čiže prácu v nej sme mali dobre nacielenú. Na vytvorenie dizajnu desktopovej aplikácie bol použitý jazyk FXML v aplikácii Scene Builder. Pre komunikáciu s FXML sme používali programovací jazyk Java, konkrétniejsie framework JavaFX. S touto technológiou sme sa už taktiež v minulosti stretli na hodinách objektového programovania. Aj tu platí, že čo sme nevedeli, dohľadali sme si na internete.

Posledná aplikácia je mobilná aplikácia, ktorá bola vytvorená v aplikácii Android Studio, ktoré využíva jazyk XML na vytváranie grafického prostredia a jazyk Java na komunikáciu s ním. Tieto technológie pre nás taktiež neboli žiadoucou novinkou, keďže sme sa ich učili používať na hodinách programovania mobilných aplikácií.

Našimi čiastkovými cieľmi bolo v prvom rade vytvoriť databázu, kde je možné nahrať celú štruktúru školy a materiály pre študentov. Taktiež sa stará o prihlásenie užívateľov do aplikácie. Každá z vyššie spomenutých aplikácií pracuje s tou istou databázou, ktorú sme nahrali na server.

Ďalším čiastkovým cieľom bolo vytvorenie uceleného grafického návrhu našej aplikácie, ktorý je realizovaný pomocou aplikácie Figma. Snažili sme sa nastaviť jednotný dizajn všetkých troch aplikácií. Návrh nášho dizajnu je dostupný na webovej adrese

[https://www.figma.com/file/N9p7Hp9HH0uJcGCGmIKXRC/RPJ\\_WEB DESIGN?node-id=0%3A1&t=MspxAAtCFqmBPzH2-1](https://www.figma.com/file/N9p7Hp9HH0uJcGCGmIKXRC/RPJ_WEB DESIGN?node-id=0%3A1&t=MspxAAtCFqmBPzH2-1)

Ďalej sme museli vytvoriť prihlásovacie menu, ktoré pozostáva z prihlásovacieho formulára, tlačidla pre zabudnuté heslo a registrácie školy. Po kliknutí na zabudnuté heslo užívateľ zadá jeho email, na ktorý mu systém pošle overovací kód, ktorý užívateľ zadá do formulára, ktorý sa mu zobrazí. Po úspešnom overení si užívateľ môže zmeniť heslo. Ak chce niekto do systému zaregistrovať školu, klikne na tlačidlo registrácia školy, kde zadá svoju emailovú adresu a systém pošle upozornenie nášmu administrátorovi, ktorý sa mu ozve a dohodne s ním podrobnosti.

Po prihlásení sme museli vytvoriť samotné prostredie, ktoré je odlišné podľa typu prihláseného účtu. V aplikáciach sme vytvorili navigačné menu, umiestnené podľa potreby resp. aplikácie (mobilná aplikácia má menu na trochu odlišnom mieste ako ostatné aplikácie z dôvodu veľkosti obrazovky). V navigačnom menu môže užívateľ meniť jednotlivé karty aplikácie, alebo sa odhlásiť. Po odhlásení systém presmeruje užívateľa na prihlásovaci stránku.

Náš systém poskytuje štyri typy účtov (administrátor, školský administrátor, učiteľ a študent). Cez webovú aplikáciu sa môžu prihlásiť všetky typy účtov. Desktopová aplikácia neponúka možnosť prihlásenia administrátora (ktorý môže spravovať celú databázu) ale ponúka možnosť prihlásenia školského administrátora, učiteľa a študenta. Mobilná aplikácia je robená čisto pre študentov.

Ak sa prihlási učiteľ uvidí informácie o sebe ako meno, priezvisko, triedu, ktorej je triedny a všetky predmety, ktoré učí. Môže sa nasledovne prepnúť na kartu predmety, kde môže podľa predmetu vytvárať témy a podtémy pre daný predmet a nahrávať do nich materiály pre študentov. Taktiež si tam môže nastaviť, či danú tému vidí len on alebo všetci vrátane študentov.

Ak sa prihlási študent, tak uvidí informácie o sebe ako meno, priezvisko, trieda, do ktorej chodí, jeho skupinu a predmety. Keď prejde na kartu predmety zobrazí sa mu štruktúra s názvami predmetov a tém z jednotlivých predmetov, odkiaľ si môže stiahnuť materiály potrebné k štúdiu.

Ak sa prihlási školský administrátor, tak uvidí rôzne karty, ktoré predstavujú jednotlivé tabuľky databázy, v ktorých môže meniť údaje v databáze, a taktiež pridať a mazať informácie z databázy.

Ak sa prihlási administrátor, tak vidí všetky tabuľky, ktoré vidí školský administrátor ale pre všetky školy v systéme, ktoré môže pridávať, mazať a upravovať.

## **2 Teoretická časť – Úvod do problematiky**

### **2.1 Mobilná aplikácia**

Mobilná aplikácia, najčastejšie označovaná ako aplikácia, je typ aplikačného softvéru navrhnutý na spustenie na mobilnom zariadení, ako je smartfón alebo tablet. Mobilné aplikácie často slúžia na to, aby používateľom poskytovali podobné služby, aké sú dostupné na PC. Aplikácie sú vo všeobecnosti malé, samostatné softvérové jednotky s obmedzenou funkciou. Toto použitie aplikačného softvéru pôvodne spopularizovala spoločnosť Apple Inc. a jej App Store, ktorý ponúka tisíce aplikácií pre iPhone, iPad a iPod Touch. (Rouse, 2020)

### **2.2 Android Studio**

Android Studio je oficiálne integrované vývojové prostredie (IDE) pre vývoj aplikácií pre Android. Je založené na výkonného editore kódu a vývojárskych nástrojoch od IntelliJ IDEA a ponúka ešte viac funkcií, ktoré zvýšia vašu produktivitu pri vytváraní aplikácií pre Android. (Android Developers, 2023)

### **2.3 Java**

Java je objektovo orientovaný programovací jazyk. Je vyvíjaný spoločnosťou Oracle. Jeho syntax vychádza z jazykov C a C++. Zdrojové programy sa nekomplilujú do strojového kódu, ale do medzistupňa, tzv. „byte-code“, ktorý nie je závislý od konkrétnej platformy. Tento byte-code neskôr vykonáva a spracováva interpreter, Java Virtual Machine. (Wikipedia, 2023)

### **2.4 XML**

XML znamená eXtensible Markup Language, v preklade rozšíriteľný značkovací jazyk, ktorý bol vyvinutý a štandardizovaný konzorciov W3C (World Wide Web Consortium) ako pokračovanie jazyka SGML a zovšeobecnenie jazyka HTML. Umožňuje jednoduché vytváranie konkrétnych značkovacích jazykov na rôzne účely a široké spektrum rôznych typov údajov. (Wikipedia, 2017)

## **2.5 IntelliJ Idea**

IntelliJ IDEA je komerčné vývojové prostredie pre programovanie v jazykoch Java, Groovy a ďalších. (Wikipedia, 2023)

## **2.6 JavaFX**

JavaFX je softvérová platforma postavená na báze platformy Java z dielne spoločnosti Sun Microsystems. Slúži na vývoj tzv. RIA aplikácií (Rich Internet applications), v preklade bohatých internetových aplikácií. Ako už z názvu vyplýva, jedná sa predovšetkým o prvky zaistujúce interaktivitu webových stránok, teda stránok, ktoré s užívateľom, oproti tým statickým, nejakým spôsobom komunikujú. V prípade JavaFX webových aplikácií, ktoré sa formou pluginu spúšťajú z prehliadača, navyše platforma umožňuje si aplikáciu stiahnuť a využívať ju ako klasický desktopový program a to aj v offline móde. JavaFX je cielená na vývojárov programov túžiacich po rýchлом a jednoduchom vývoji pre desktopy, internetové prehliadače, ale aj rýchlo rozvíjajúcim sa mobilnú oblast. Vznikla ako reakcia na masové rozšírenie platforiem ako je Adobe Flash a predovšetkým Microsoft Silverlight. (Wikipedia, 2023)

## **2.7 FXML**

FXML je značkovací jazyk používateľského rozhrania založený na XML, ktorý vytvorila spoločnosť Oracle Corporation na definovanie používateľského rozhrania aplikácie JavaFX. FXML predstavuje alternatívu k navrhovaniu používateľských rozhrani pomocou procedurálneho kódu a umožňuje abstrahovať návrh programu od programovej logiky. (Wikipedia, 2021)

## **2.8 MySQL**

MySQL je slobodný a otvorený viacvláknový, viacužívateľský SQL relačný databázový server. MySQL je podporovaný na viacerých platformách (ako Linux, Windows či Solaris) a je implementovaný vo viacerých programovacích jazykoch ako PHP, C++ či Perl. Databázový systém je relačný, typu DBMS (database management system). Každá databáza je v MySQL tvorená z jednej alebo viacerých tabuľiek, ktoré

majú riadky a stĺpce. V riadkoch sa rozoznávajú jednotlivé záznamy, stĺpce udávajú dátový typ jednotlivých záznamov, pracuje sa s nimi ako s poľami. Práca s MySQL databázou je vykonávaná pomocou dopytov (anglicky queries), ktoré sú tvorené programovacím jazykom SQL (Structured Query Language). (Wikipedia, 2022)

## 2.9 SQL

Structured Query Language, skrátene SQL je doménovo špecifický jazyk používaný v programovaní a určený na správu dát uložených v systém riadenia relačnej databázy (RDBMS) alebo na spracovanie toku v systéme riadenia toku relačných údajov (RDSMS). Je to užitočné najmä pri práci so štruktúrovanými údajmi, t. j. údajmi zahŕňajúcimi vzťahy medzi entitami a premennými. (Wikipedia, 2023)

## 2.10 Visual studio Code

Visual Studio Code, bežne označovaný aj ako VS Code, je editor zdrojového kódu pre Windows, Linux a macOS vytvorený spoločnosťou Microsoft pomocou Electron Frameworku. Medzi funkcie patrí podpora pre ladenie (debugging), zvýrazňovanie syntaxe, inteligentné dokončovanie kódu, snippetsy, zmena štruktúry kódu a vstavané GUI pre prácu s Gitom. Používateľia môžu meniť tému, klávesové skratky, predvolby a inštalovať rozšírenia, ktoré pridávajú ďalšie funkcie. (Wikipedia, 2023)

## 2.11 Sublime Text

Sublime Text je multiplatformný editor textových súborov a zdrojového kódu napísaný v C++ s programovým API v Pythone. Jeho funkcionalita je rozšíriteľná pluginmi vytváranými komunitou okolo Sublime Textu. Samotný editor nie je open source ani zadarmo, avšak väčšina pluginov áno. (Wikipedia, 2021)

## 2.12 HTML

HyperText Markup Language alebo HTML je štandardný značkovací jazyk pre dokumenty určené na zobrazenie vo webovom prehliadači. Môžu mu pomôcť technológie, ako sú kaskádové štýly (CSS) a skriptovacie jazyky, ako je JavaScript. (Wikipedia, 2023)

## **2.13 CSS**

Cascading Style Sheets (CSS) je jazyk štýlov používaný na popis prezentácie dokumentu napísaného v značkovacom jazyku, ako je HTML alebo XML. CSS je popri HTML a JavaScript základnou technológiou World Wide Web. CSS je navrhnuté tak, aby umožňovalo oddelenie obsahu a prezentácie vrátane rozloženia, farieb a fontov. Toto oddelenie môže zlepšiť dostupnosť obsahu; poskytnúť väčšiu flexibilitu a kontrolu pri špecifikácii prezentačných charakteristik; umožniť viacerým webovým stránkam zdieľať formátovanie špecifikovaním príslušného CSS v samostatnom súbore .css, čo znížuje zložitosť a opakovanie štrukturálneho obsahu; a povoliť ukladanie súboru .css do vyrovnavacej pamäte, aby sa zvýšila rýchlosť načítania stránok medzi stránkami, ktoré zdieľajú súbor, a jeho formátovanie. (Wikipedia, 2023)

## **2.14 PHP**

PHP (Hypertext Preprocessor) je populárny open source programovací jazyk, ktorý sa používa najmä na programovanie klient-server aplikácií (na strane servera) a pre vývoj dynamických webových stránok. (Wikipedia, 2023)

## **2.15 Javascript**

JavaScript, často skracovaný ako JS , je programovací jazyk , ktorý je popri HTML a CSS jednou zo základných technológií World Wide Web . Od roku 2022 používa 98 % webových stránok JavaScript na strane klienta na správanie webových stránok , pričom často obsahujú knižnice tretích strán . Všetky hlavné webové prehliadače majú vyhradený nástroj JavaScript na spustenie kódu na zariadeniach používateľov. JavaScript je jazyk na vysokej úrovni , často kompilovaný práve včas, ktorý je v súlade so štandardom ECMAScript. Má dynamické písanie , objektovú orientáciu založenú na prototype a prvotriedne funkcie. Je to multiparadigma, podporuje udalosťami riadené , funkčné a imperatívne štýly programovania . Má aplikačné programovacie rozhrania (API) na prácu s textom, dátumami, regulárnymi výrazmi, štandardnými dátovými štruktúrami a objektovým modelom dokumentu (DOM). (Wikipedia, 2023)

## **2.16 phpMyAdmin**

PhpMyAdmin je bezplatný softvérový nástroj napísaný v PHP, určený na správu MySQL cez web. phpMyAdmin podporuje širokú škálu operácií na MySQL a MariaDB. Často používané operácie (správa databáz, tabuliek, stĺpcov, vzťahov, indexov, používateľov, oprávnení atď.) je možné vykonávať cez používateľské rozhranie, pričom stále máte možnosť priamo vykonávať ľubovoľný príkaz SQL. (phpMyAdmin, 2006)

## **2.17 XAMPP**

XAMPP je v informatike označenie pre multiplatformou softvérový balíček vyvinutý firmou Apache Friends. Obsahuje voľne dostupný otvorený softvér (webový server Apache, databázy MariaDB, mail server Mercury MTS a interpret programovacích jazykov PHP a Perl). Označenie XAMPP sa skladá z piatich písmen, kde každé reprezentuje jednu spomínanú funkciu: multiplatformnú v preklade Cross-Platform (X), Apache (A), MariaDB (M), PHP (P) a Perl (P). (Wikipedia, 2023)

## **2.18 Figma**

Figma je kooperatívna webová aplikácia pre návrh rozhrania s ďalšími offline funkciami, ktoré umožňujú desktopové aplikácie pre macOS a Windows. Sada funkcií Figma sa zameriava na používateľské rozhranie a dizajn používateľského prostredia s dôrazom na spoluprácu v reálnom čase s využitím rôznych vektorových grafických editorov a nástrojov na vytváranie prototypov. Mobilná aplikácia Figma pre Android a iOS umožňuje prezeranie a interakciu s prototypmi Figma v reálnom čase na mobilných zariadeniach a tabletoch. (Wikipedia, 2023)

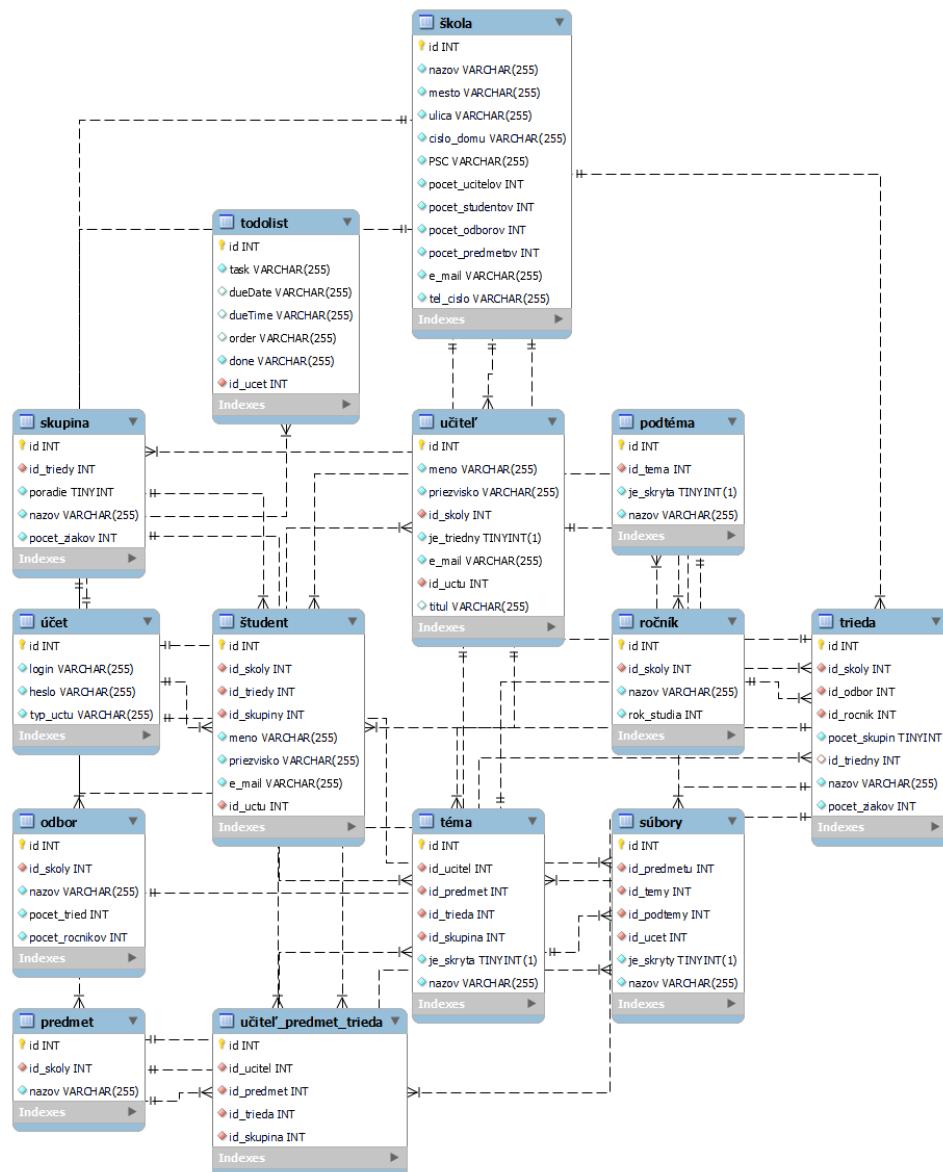
## **2.19 PhpStorm**

PhpStorm je komerčné multi-platformové vývojové prostredie (IDE) pre PHP vyvíjané českou spoločnosťou JetBrains. Hoci je vyvíjaný českou spoločnosťou, jediný jazyk, v ktorom je možné ho používať je angličtina. PhpStorm ponúka editor pre PHP a ďalšie jazyky s analýzou kódu v reálnom čase, prevenciou chýb a automatickým formátovaním. PhpStorm umožňuje našepkávanie kódu, vrátane doplnovania kódu generátorov, cyklov, metód, polí a ďalších. (Wikipedia, 2023)

### 3 Praktická časť

#### 3.1 Databáza

Naša databáza bola naprogramovaná v jazyku MySQL v aplikácii MySQL Workbench. Skladá sa celkovo zo štrnásťich tabuľiek a je rozdelená do štyroch sekcií. Prvá sekcia sa zaoberá zberom informácií o účtoch (login, heslo, typ účtu), druhá sekcia je zameraná na administrátorský panel a všetky informácie o škole, tretia sekcia sa venuje ukladaniu informácií o témach, podtémach a súboroch. Posledná sekcia zabezpečuje ukladanie dát z ToDoListu.



Obrázok 1: ERD diagram databázy

## 3.2 Desktopová aplikácia

Desktopová aplikácia je jednou z troch aplikácií, ktoré ponúkame v rámci nášho projektu. Od webovej aplikácie sa odlišuje tým, že nemá administrátorský panel pre správu celej databázy. Ponúka plnú správu (administráciu) danej školy cez školský administrátorský účet, prihlásenie učiteľov a študentov. Učitelia môžu vytvárať témy, podtémy a súbory pre žiakov, ktorí si ich po zverejnení môžu prezerátať a stiahovať.

Táto aplikácia využíva architektúru MVC (Model, View, Controller), ktorá slúži na zefektívnenie kódu, jeho prehľadnosť a logické rozdelenie na jednotlivé časti.

Desktopová aplikácia je naprogramovaná v jazyku Java (Model a Controller) a v jazyku FXML (View).

### 3.2.1 Trieda Main

Trieda Main v JavaFX aplikácii slúži ako vstupný bod (entry point) pre aplikáciu, teda ako miesto, kde sa aplikácia spustí. V tejto triede sa inicializujú rôzne komponenty a okno aplikácie, ktoré sa zobrazí na obrazovke používateľa.

Trieda Main v JavaFX aplikácii obsahuje metódu main, ktorá sa spustí ako prvá pri spustení aplikácie.

Naša trieda main obsahuje okrem metódy start(), ktorá inicializuje okno aplikácie aj metódy changeScreen(), getStageWidth() a getStageHeight().

Metóda changeScreen() zabezpečuje zmenu okna aplikácie, pomocou čoho „prepíname“ jednotlivé karty aplikácie. Metódy getStageWidth() a getStageHeight() vrátia výšku a šírku okna.

```

public class Main extends Application {

    8 usages
    private static Stage stg;
    2 usages
    private double stageWidth;
    2 usages
    private double stageHeight;

    @Override
    public void start(Stage stage) throws IOException {
        stg = stage;
        FXMLLoader fxmlLoader = new FXMLLoader(Main.class.getResource(name: "login-view.fxml"));
        Scene scene = new Scene(fxmlLoader.load());
        stage.setTitle("Login View");
        stage.setScene(scene);
        stage.setMaximized(false);
        stage.setResizable(false);
        stage.show();
    }

    // metoda, ktora meni sceny aplikacie
    public void changeScrene(String fxml, String name, boolean maximized) throws IOException {
        Parent pane = FXMLLoader.load(getClass().getResource(fxml));
        stg.setTitle(name);
        stg.setMaximized(maximized);
        stg.setResizable(false);
        stg.getScene().setRoot(pane);
    }

    public double getStageWidth() {
        stageWidth = stg.getWidth();
        return stageWidth;
    }

    public double getStageHeight() {
        stageHeight = stg.getHeight();
        return stageHeight;
    }

    3 usages
    public Stage getStage() { return stg; }

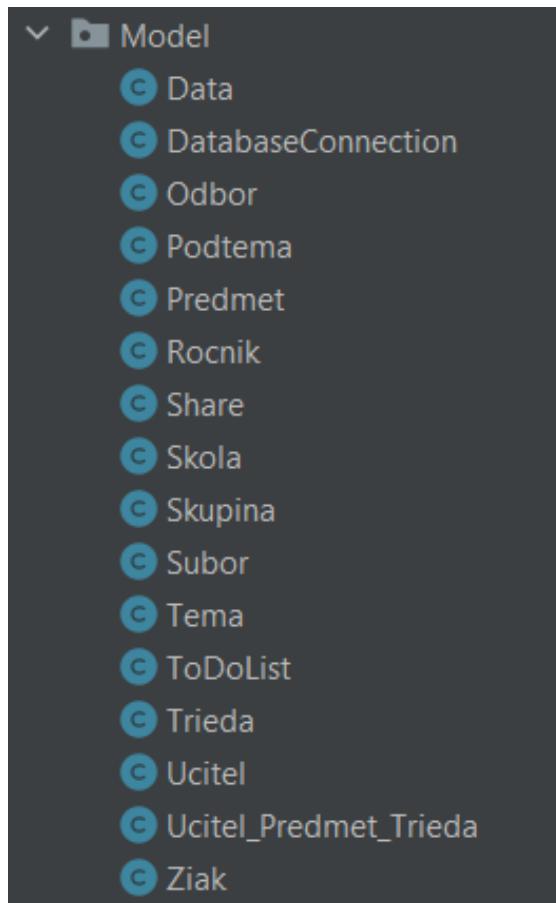
    public static void main(String[] args) { launch(); }
}

```

Obrázok 2: Zdrojový kód triedy „Main“ v desktopovej aplikácii

### 3.2.2 Model

Model našej aplikácie pozostáva z týchto tried:



Obrázok 3: Triedy Modelu desktopovej

aplikácie

Každá trieda z Modelu má svoje špeciálne využitie. Trieda Data nosí všetky informácie, ktoré je potrebné zdieľať medzi ostatnými triedami.

Trieda DatabaseConnection umožňuje komunikáciu s databázou.

Ostatné triedy slúžia ako „formičky“ pre jednotlivé tabuľky databázy, ktoré zobrazujeme v tabuľkách tejto aplikácie. Ich úlohou je teda ukladať dátá z databázy do jednotlivých premenných a tieto údaje následne posielat Controlleru každej tabuľky, ktorý ich zobrazí vo View-e.

### 3.2.2.1 Trieda Data

Táto trieda slúži ako zdieľaná trieda pre uchovávanie potrebných dát z ostatných tried, ktoré si ich navzájom zdieľajú prostredníctvom tejto triedy. Pozostáva z getterov a setterov, pomocou ktorých si môžeme potrebné dátá uložiť a pristúpiť k nim tam, kde to je potrebné.

---

1 usage

```
private Data() {}

public static Data getInstance() { return instance; }

public String getLogin() { return login; }

public void setLogin(String login) { this.login = login; }

public String getIdUctu() { return idUctu; }
```

3 usages

```
public void setIdUctu(String idUctu) { this.idUctu = idUctu; }

public String getIdUcitel() { return idUcitel; }
```

2 usages

```
public void setIdUcitel(String idUcitel) { this.idUcitel = idUcitel; }
```

7 usages

```
public String getIdStudent() { return idStudent; }
```

1 usage

```
public void setIdStudent(String idStudent) { this.idStudent = idStudent; }
```

6 usages

```
public String getIdTriedy() { return idTriedy; }
```

1 usage

```
public void setIdTriedy(String idTriedy) { this.idTriedy = idTriedy; }
```

Obrázok 4: Zdrojový kód triedy „Data“ v desktopovej aplikácii

### 3.2.2.2 Trieda DatabaseConnection

Táto trieda sprostredkováva komunikáciu medzi aplikáciou a databázou prostredníctvom MySQL JDBC Driver-u. MySQL JDBC Driver poskytuje funkcie na pripojenie k databáze MySQL, vytvorenie dotazov, vykonanie dotazov a získanie výsledkov. Tento driver umožňuje Java aplikáciám komunikovať s databázou MySQL prostredníctvom protokolu TCP/IP a poskytuje rozhranie pre vytvorenie a odoslanie SQL príkazov do databázy MySQL.

```
public class DatabaseConnection {  
  
    2 usages  
    public Connection databaseLink;  
  
    public Connection getConnection(){  
  
        String databaseName = "notriggers";  
        String databaseUser = "root";  
        String databasePassword = "";  
        String url = "jdbc:mysql://localhost/" + databaseName;  
  
        try {  
            Class.forName( className: "com.mysql.cj.jdbc.Driver");  
            databaseLink = DriverManager.getConnection(url, databaseUser, databasePassword);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
  
        return databaseLink;  
    }  
}
```

Obrázok 5: Zdrojový kód triedy „DatabaseConnection“ v desktopovej aplikácii

### 3.2.2.3 Ostatné triedy modelu

Ostatné triedy modelu obsahujú konštruktor, gettery a settery potrebné pre získavanie a nastavovanie hodnôt podľa využitia danej triedy. Každá trieda je špeciálne navrhnutá pre konkrétnu tabuľku v aplikácii a databáze, aby bolo možné zobraziť dátu z tabuliek databázy v daných tabuľkách v aplikácii.

```
public class Skola {  
  
    3 usages  
    String id, nazov, mesto, ulica, cisloDomu, psc, pocetUcitelov, po  
  
    1 usage  
    public Skola(String id, String nazov, String mesto, String ulica,  
        this.id = id;  
        this.nazov = nazov;  
        this.mesto = mesto;  
        this.ulica = ulica;  
        this.cisloDomu = cisloDomu;  
        this.psc = psc;  
        this.pocetUcitelov = pocetUcitelov;  
        this.pocetStudentov = pocetStudentov;  
        this.pocetOdborov = pocetOdborov;  
        this.pocetPredmetov = pocetPredmetov;  
        this.email = email;  
        this.telCislo = telCislo;  
    }  
  
    public String getId() { return id; }  
  
    public void setId(String id) { this.id = id; }  
  
    public String getNazov() { return nazov; }  
  
    public void setNazov(String nazov) { this.nazov = nazov; }  
}
```

Obrázok 6: Zdrojový kód triedy „Skola“ v desktopovej aplikácii

### 3.2.3 Controller

Controller zabezpečuje komunikáciu medzi logikou aplikácie a grafickým užívateľským rozhraním. Každý Controller je prepojený s jeho .fxml súborom, s ktorým interaguje, a teda je pre daný .fxml súbor „šitý na mieru“. V javaFX sa Controller prepája s .fxml súborom pomocou atribútu fx:controller, ktorý definuje názov Controllera a tento atribút sa nachádza v elemente, ktorý je definovaný ako koreňový element (root).

```
fx:controller="com.rpj.rpj_desktop.Login.LoginController"
```

Obrázok 7: Prepojenie Controllera s .fxml súborom v desktopovej aplikácii

Controller obsahuje odkazy na jednotlivé komponenty .fxml súboru (ako sú rôzne tlačidlá, tabuľky, kontajnere, a pod.), ktoré budú napĺňa dátami z logiky aplikácie alebo vytvára pre ne eventy (čo sa má stať, ak sa na daný objekt klikne).

```
11 usages
@FXML
ComboBox<String> typUctuDropdown;
2 usages
@FXML
TextField login;
5 usages
@FXML
PasswordField password;
2 usages
@FXML
Button verify;
2 usages
@FXML
Label forgottenPassword;
2 usages
@FXML
Label registraciaSkoly;
```

Obrázok 8: Prepojenie komponentov .fxml súboru  
s Controllerom v desktopovej aplikácii

```

forgottenPassword.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        try {
            main.changeScrene( fxml: "zabudnute_heslo-view.fxml", name: "Zabudnuté heslo", maximized: false);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
});

registraciaSkoly.setOnMouseClicked(new EventHandler<MouseEvent>() {
    @Override
    public void handle(MouseEvent event) {
        try {
            main.changeScrene( fxml: "registracia_skoly-view.fxml", name: "Registrácia školy", maximized: false);
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
});

```

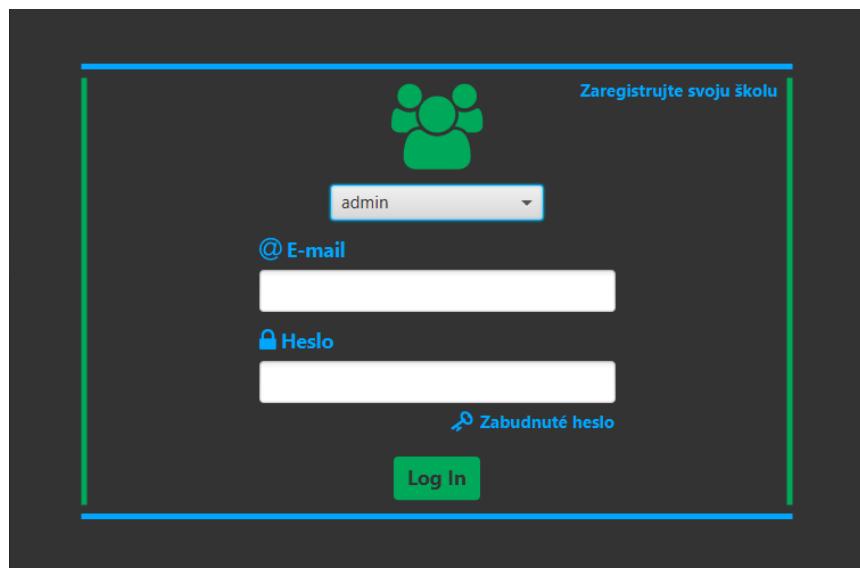
**Obrázok 9:** Vytvorenie eventov s akciami, čo sa má stať po kliknutí na konkrétny komponent v desktopovej aplikácii

### 3.2.4 View a rýchly návod používania aplikácie (ukážka funkcia)

View tejto aplikácie pozostáva z jednotlivých .fxml súborov, ktoré reprezentujú grafickú stránku našej aplikácie. Každý .fxml súbor má svoj Controller, ktorý zabezpečuje interakciu medzi užívateľským rozhraním a logikou aplikácie.

#### 3.2.4.1 Prihlásovací panel

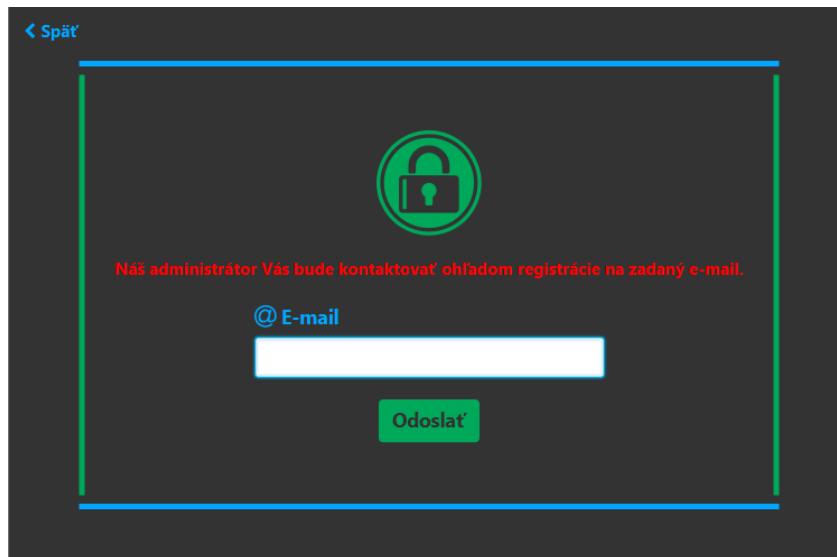
Po spustení aplikácie sa nám zobrazí prihlásovací panel, pomocou ktorého sa môžeme prihlásiť do systému, zmeniť si zabudnuté heslo alebo zaregistrovať školu.



**Obrázok 10:** Prihlásovací panel v desktopovej aplikácii

### 3.2.4.2 Registrácia školy

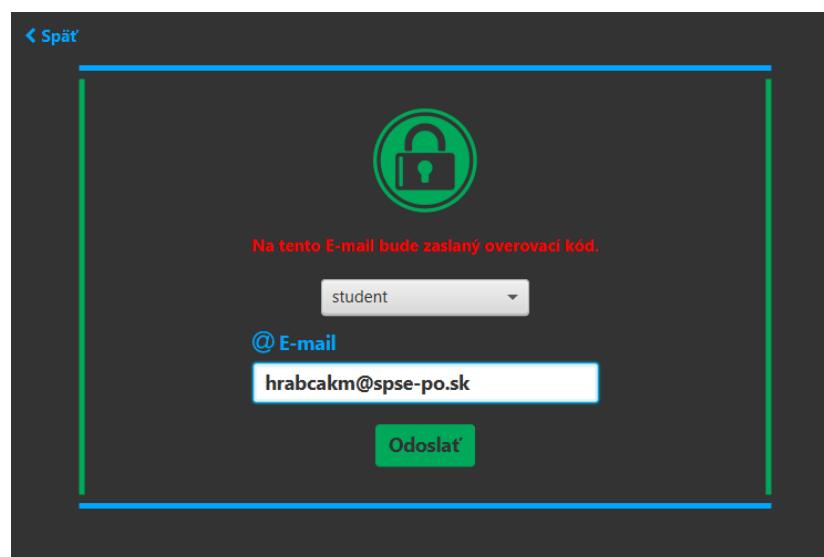
V pravom hornom rohu sa nachádza text „Zaregistrujete svoju školu“. Po kliknutí na tento text sa zobrazí okno, kde môžeme zadať svoju e-mailovú adresu, pomocou ktorej budeme komunikovať s administrátorom systému, ktorý nám pomôže s registráciou školy.



Obrázok 11: Panel pre registráciu školy v desktopovej aplikácii

### 3.2.4.3 Zabudnuté heslo

Pod prihlasovacím formulárom sa nachádza text „Zabudnuté heslo“. Po kliknutí na tento text sa zobrazí okno, kde musíme zadať typ účtu (študent, administrátor alebo učiteľ) a emailovú adresu účtu zaregistrovaného v systéme.



Obrázok 12: Panel pre odoslanie overovacieho kódu na e-mail v desktopovej aplikácii

Po zadaní emailovej adresy a kliknutí na tlačidlo „Odoslat“ nám systém pošle na zadanú emailovú adresu email s overovacím kódom.



Obrázok 13: E-mail s overovacím kódom

Zasielanie e-mailov je realizované prostredníctvom nasledujúceho kódu:

```
String urlLink = "http://localhost/Mail/sendMail.php?email=" + mail + "&code=" + code;
URL url = new URL(urlLink);
HttpURLConnection conn = (HttpURLConnection) url.openConnection();
conn.setRequestMethod("GET");

BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
StringBuffer sb = new StringBuffer();
String line;

while ((line = in.readLine()) != null) {
    sb.append(line);
}
in.close();
System.out.println(sb.toString());
```

Obrázok 14: Zdrojový kód pre odosielanie e-mailu so zdrojovým kódom v desktopovej aplikácii

Vyššie uvedený kód posiela request na server s informáciami (e-mailom a vygenerovaným kódom), ktoré na serveri dostane PHP súbor, ktorý vykoná odoslanie e-mailu užívateľovi pomocou triedy PHPMailer (plnohodnotná trieda na vytváranie a prenos e-mailov pre PHP). Spomínaný PHP súbor vyzerá takto:

```

if(isset($_GET['email']) && isset($_GET['code']) && !isset($_GET['password'])) {
    try {
        //Server settings
        $mail->isSMTP();                                //Send using SMTP
        $mail->Host          = 'smtp.gmail.com';         //Set the SMTP server to send through
        $mail->SMTPAuth     = true;                      //Enable SMTP authentication
        $mail->Username      = 'rpjmail12345@gmail.com'; //SMTP username
        $mail->Password      = 'vfvqamuvucpsfxdgu';       //SMTP password
        $mail->SMTPSecure   = PHPMailer::ENCRYPTION_SMTPS; //Enable implicit TLS encryption
        $mail->Port          = 465;                       //TCP port to connect to; use 587 if you have set 'SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS'

        //Recipients
        $mail->setFrom('from@example.com', 'Mailer');
        $mail->addAddress('rpjmail12345@gmail.com', 'Joe User'); //Add a recipient

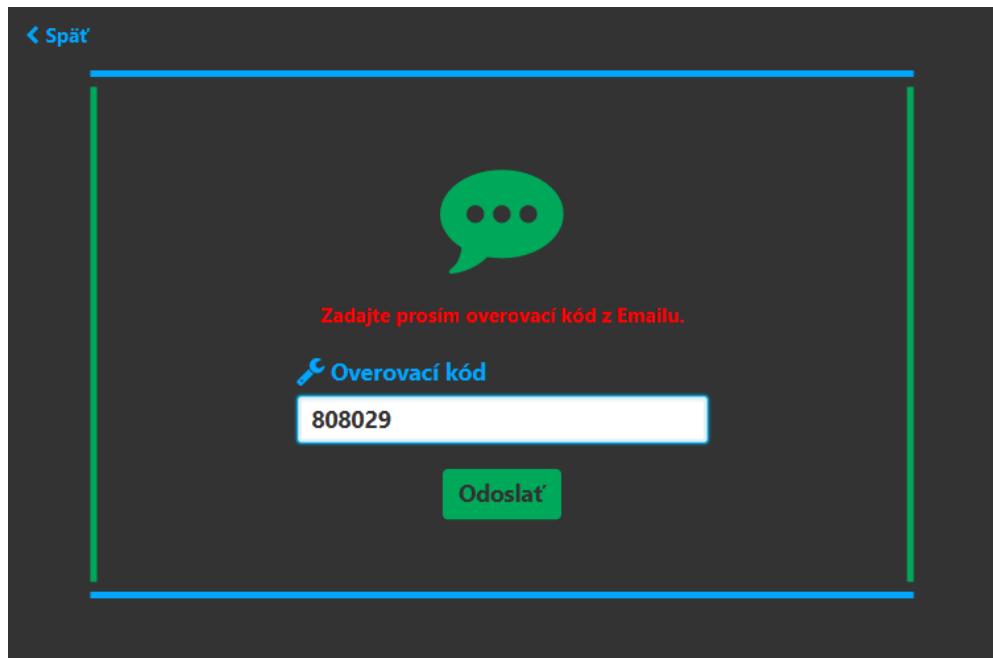
        //Content
        $mail->isHTML(true);                            //Set email format to HTML
        //$mail->Subject = 'Here is the subject';
        //$mail->Body   = 'This is the HTML message body <b>in bold!</b>';
        //$mail->AltBody = 'This is the body in plain text for non-HTML mail clients';
        $mail->Subject = 'Overovací kód';
        $mail->Body   = 'Váš overovací kód je '. '<b>' . $_GET['code'] . '</b>!';
        $mail->AltBody = $_GET['code'];

        $mail->send();
    } catch (Exception $e) {
        echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
    }
}

```

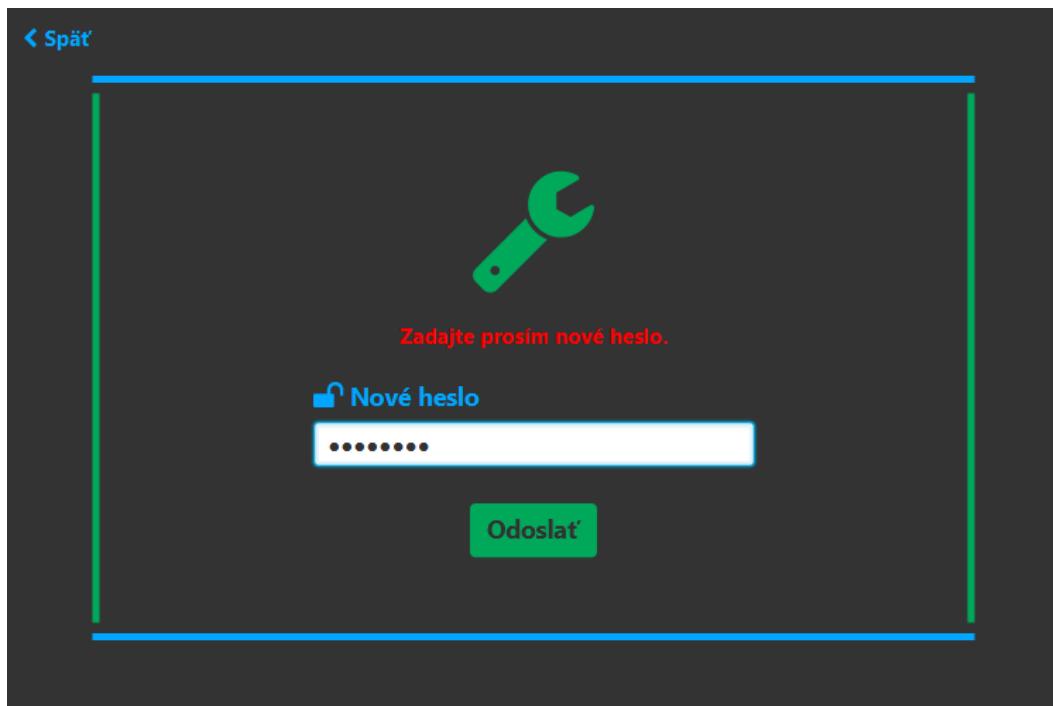
**Obrázok 15: Zdrojový kód PHP súboru na serveri, ktorý zabezpečuje odosielanie e-mailov v desktopovej aplikácii**

Overovací kód následne zadáme do nasledujúceho okna.



**Obrázok 16: Panel pre zadanie overovacieho kódu v desktopovej aplikácii**

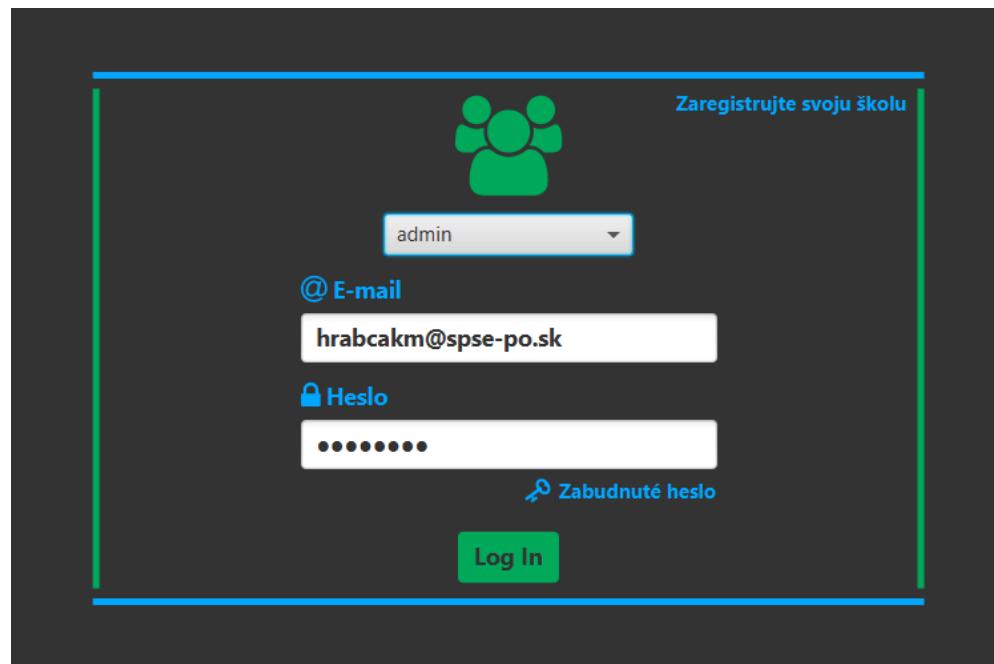
Po overení kódu nám systém dovolí zmeniť heslo.



Obrázok 17: Panel pre zmenu hesla v desktopovej aplikácii

#### 3.2.4.4 Prihlásenie

Pre prihlásenie sa do systému si musíme najprv zvoliť typ účtu, do ktorého sa chceme prihlásiť a potom zadať email a heslo od nášho účtu.



Obrázok 18: Prihlásenie pomocou prihlasovacieho panela v desktopovej aplikácii

### 3.2.4.5 Administrátorský panel

V administrátorskom paneli môže administrátor školy spravovať všetky informácie o škole, pridať, mazať a aktualizovať informácie o škole, odboroch, ročníkoch, predmetoch, učiteľoch, triedach, skupinách a žiakoch. Administrátorský panel je rozdelený na viacero kariet, na ktorých sú zobrazené jednotlivé tabuľky, ktoré je možné podľa potreby upravovať.

Škola	Odbory	Ročníky	Predmety	Učitelia	Triedy	Žiaci	Odhliasiť sa																																																																													
<b>INFORMÁCIE O ŠKOLE</b>																																																																																				
<table border="1"><thead><tr><th>nazov</th><th>mesto</th><th>ulica</th><th>cislo_domu</th><th>PSC</th><th>pocet_uctitelov</th><th>pocet_studentov</th><th>pocet_odborov</th><th>pocet_predmetov</th><th>e-mail</th><th>tel-cislo</th></tr></thead><tbody><tr><td>Stredná priemyselná škola elektrotechnická v Prešove</td><td>Prešov</td><td>Ptzenská</td><td>1</td><td>080 01</td><td>1</td><td>1</td><td>2</td><td>3</td><td>spse@spse-po.sk</td><td>051/772 55 67</td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr><tr><td> </td><td> </td></tr></tbody></table>								nazov	mesto	ulica	cislo_domu	PSC	pocet_uctitelov	pocet_studentov	pocet_odborov	pocet_predmetov	e-mail	tel-cislo	Stredná priemyselná škola elektrotechnická v Prešove	Prešov	Ptzenská	1	080 01	1	1	2	3	spse@spse-po.sk	051/772 55 67																																																							
nazov	mesto	ulica	cislo_domu	PSC	pocet_uctitelov	pocet_studentov	pocet_odborov	pocet_predmetov	e-mail	tel-cislo																																																																										
Stredná priemyselná škola elektrotechnická v Prešove	Prešov	Ptzenská	1	080 01	1	1	2	3	spse@spse-po.sk	051/772 55 67																																																																										

Obrázok 19: Administrátorský panel (karta „Škola“)

Dáta v tabuľkách sa dajú aktualizovať dvojklikom na bunku, v ktorej chceme zmeniť informácie (niektoré bunky sa nedajú editovať a to z ochranného dôvodu).



Obrázok 20:  
Editácia buniek  
v tabuľke  
v desktopovej  
aplikácii

Pod niektorými tabuľkami sa nachádza aj tlačidlo „Pridať“, pomocou ktorého sa dostaneme na kartu, odkiaľ môžeme pridať dátu do databázy.

názov	pocet tried	deleteicon
Informačné a sietové technológie	1	trash
Elektrotechnika	0	trash

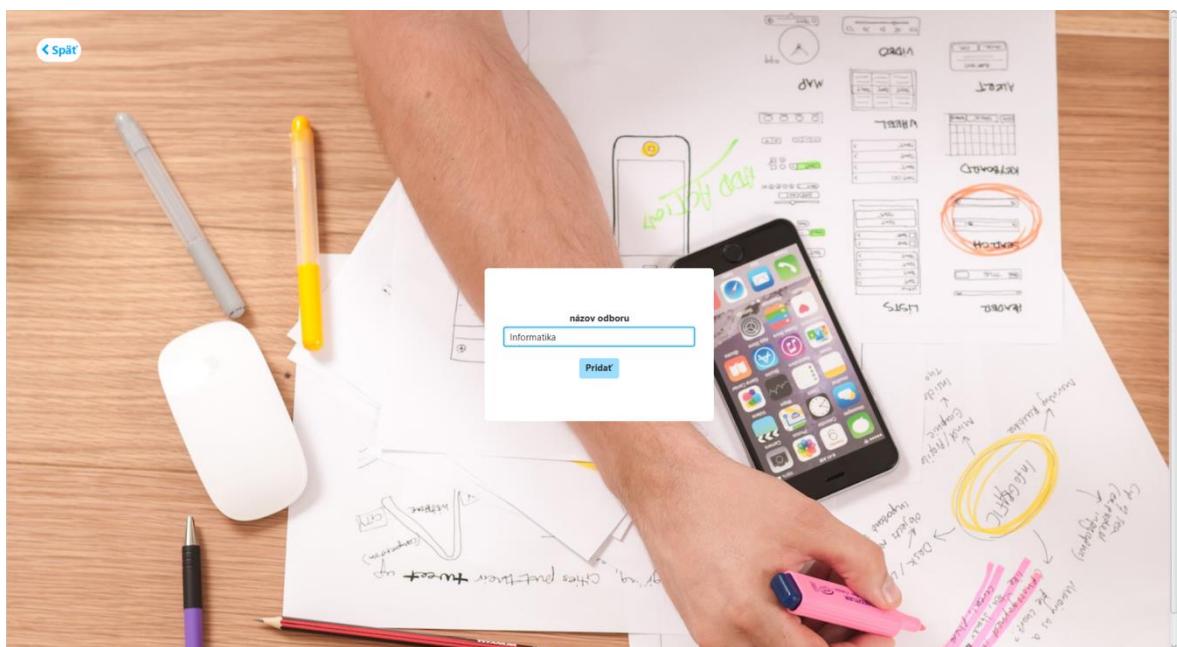
**INFORMÁCIE O ODBOROV**

Odhliásiť sa

[Pridať](#)

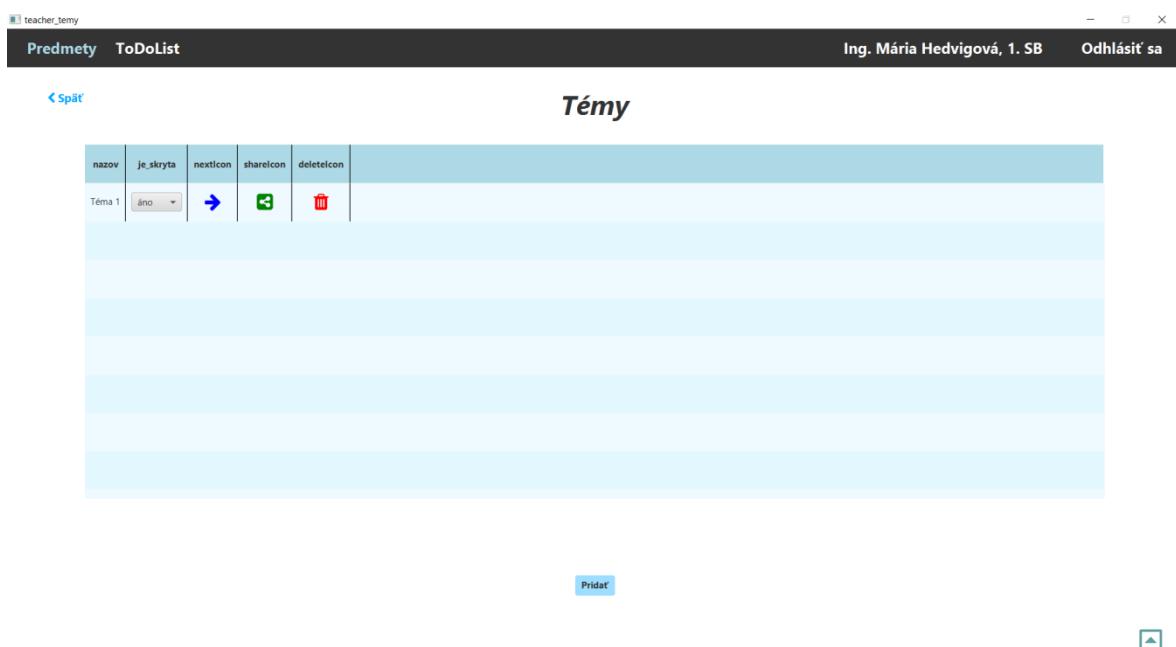
Obrázok 21: Administrátoršký panel (karta „Odbory“)

Po kliknutí na tlačidlo „Pridať“ sa zobrazí karta s formulárom pre pridanie novej položky, ktorý vyplníme a stlačením tlačidla „Pridať“ odošleme dátu do databázy.



Obrázok 22: Formulár pre pridávanie nových odborov v desktopovej aplikácii

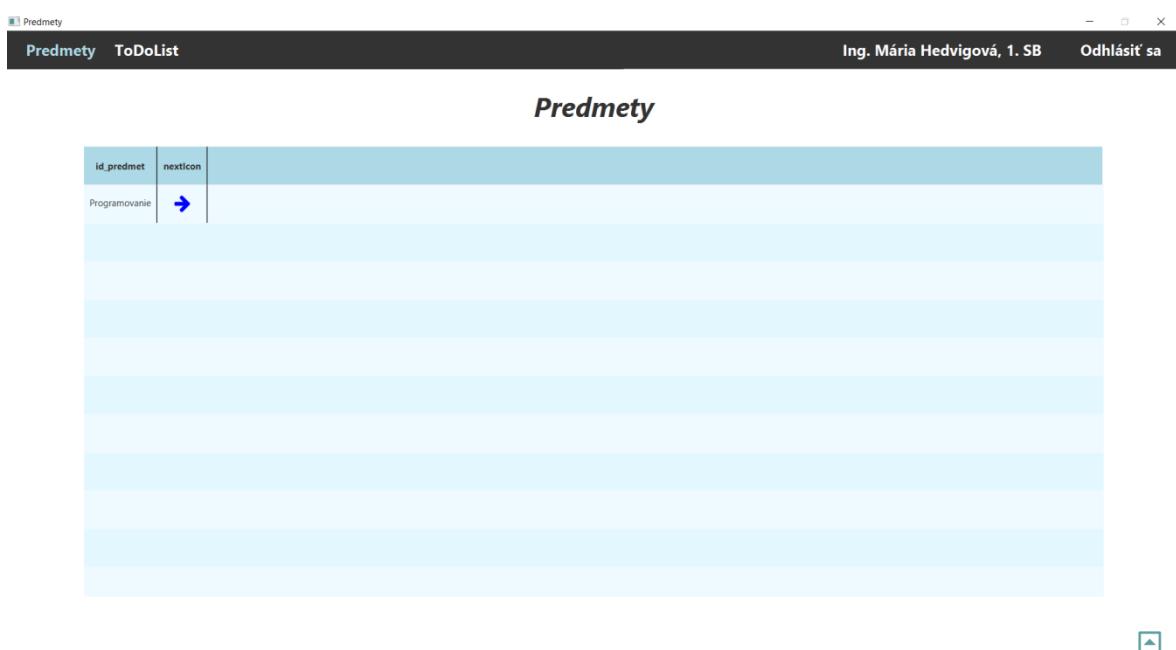
Po pridaní novej položky sa nám hned' ukáže v tabuľke, do ktorej sme pridali danú informáciu.



Obrázok 23: Zobrazenie pridaného riadka v desktopovej aplikácii

### 3.2.4.6 Učiteľský panel

Ak sa do systému prihlásime ako učiteľ, dostaneme sa na kartu, kde vidíme všetky predmety, ktoré učí daný učiteľ. Vedľa každého predmetu je modrá šípka, po kliknutí na ktorú sa prepne karta na jej podkartu (Predmet → Trieda → Skupina → Téma → Podtéma → Súbor). Využíva sa tu takzvaná stromová štruktúra.



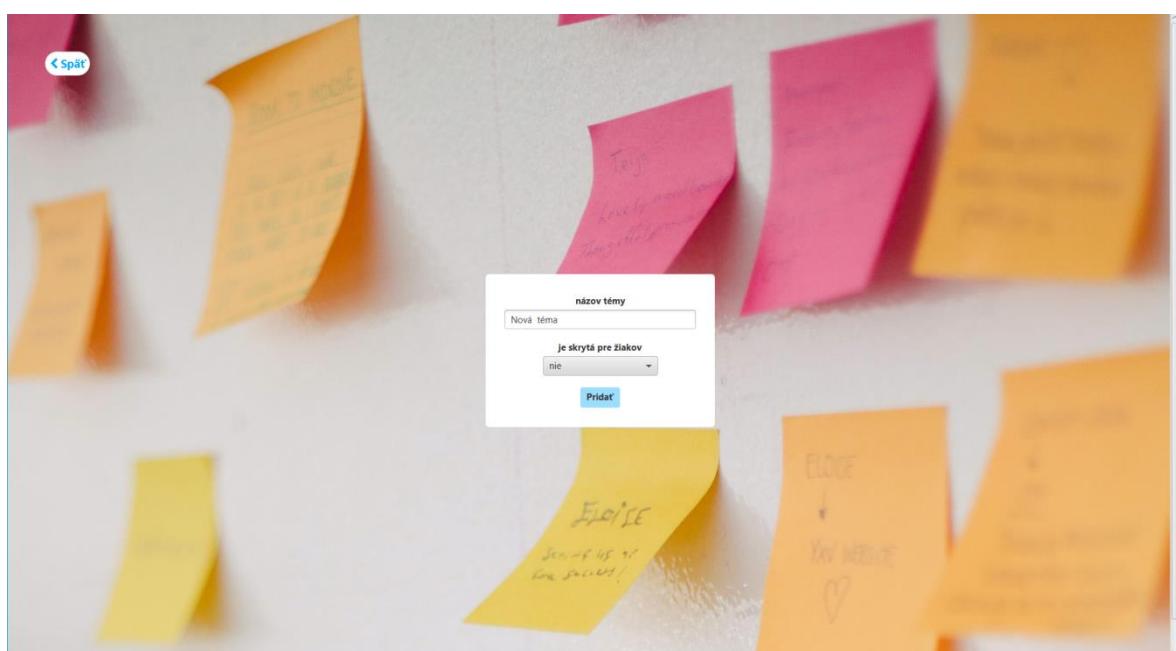
Obrázok 24: Učiteľský panel (karta „Predmety“) v desktopovej aplikácii

nazov	je_skytna	nexticon	shareicon	deleteicon
Téma 1	áno			
Nová téma	nie			

[Pridať](#)

Obrázok 25: Učiteľský panel (karta „Témy“) v desktopovej aplikácii

Po pridaní sa nám daná informácia zobrazí v tabuľke.



Obrázok 26: Formulár pre pridávanie nových tém v desktopovej aplikácii

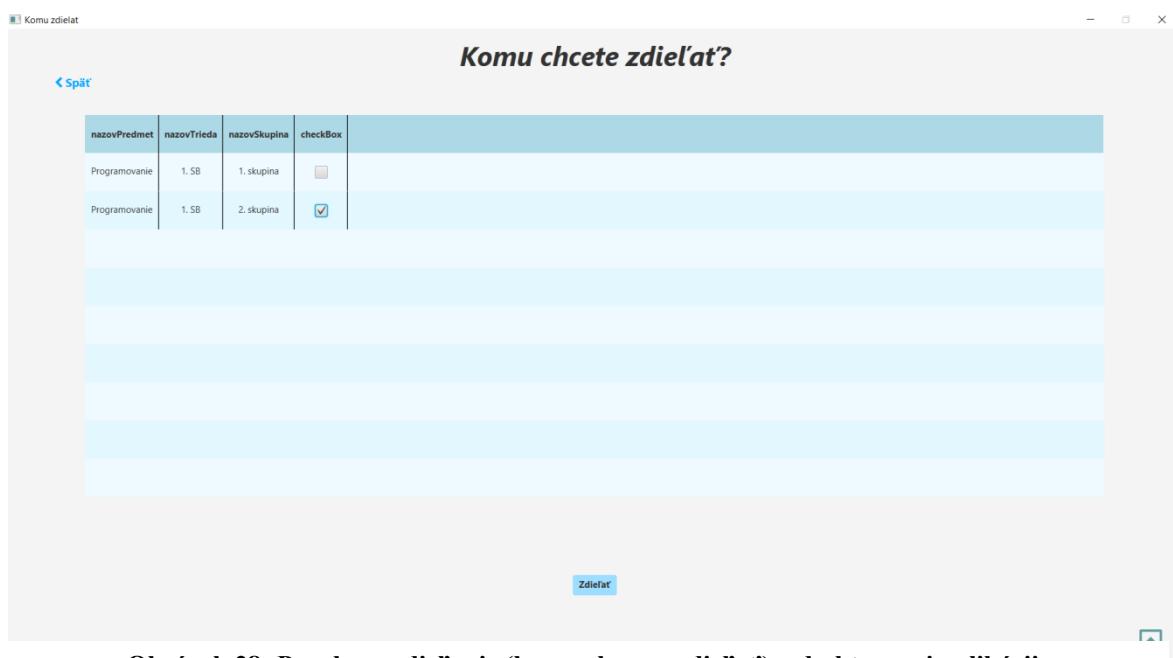
V tabuľkách téma, podtéma a súbor sa okrem modrej šípky nachádzajú aj ikonky pre vymazanie daného objektu a ikonka share. Pri mazaní si treba dávať pozor, pretože ak zmažeme tému všetky podtémy a súbory, ktoré obsahuje sa vymažú spolu s ňou. Po kliknutí na ikonku share sa dostaneme na kartu, odkiaľ môžeme zdieľať témy, podtémy alebo súbory ostatným triedam alebo skupinám.

V prvej časti si zvolíme, čo chceme zdieľať.



Obrázok 27: Panel pre zdieľanie (čo chceme zdieľať) v desktopovej aplikácii

Po kliknutí na tlačidlo „Ďalej“ sa nám zobrazí karta, kde si vyberieme komu chceme zdieľať.

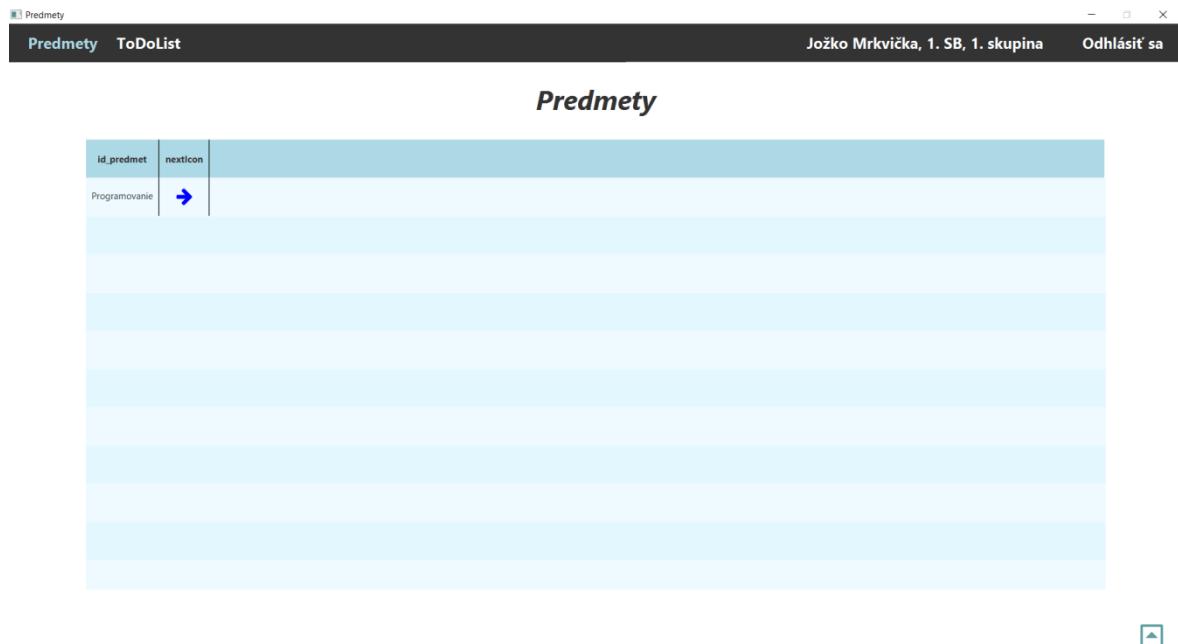


Obrázok 28: Panel pre zdieľanie (komu chceme zdieľať) v desktopovej aplikácii

Po kliknutí na tlačidlo „Zdieľať“ sa dané informácie zazdieľajú požadovaným skupinám alebo triedam.

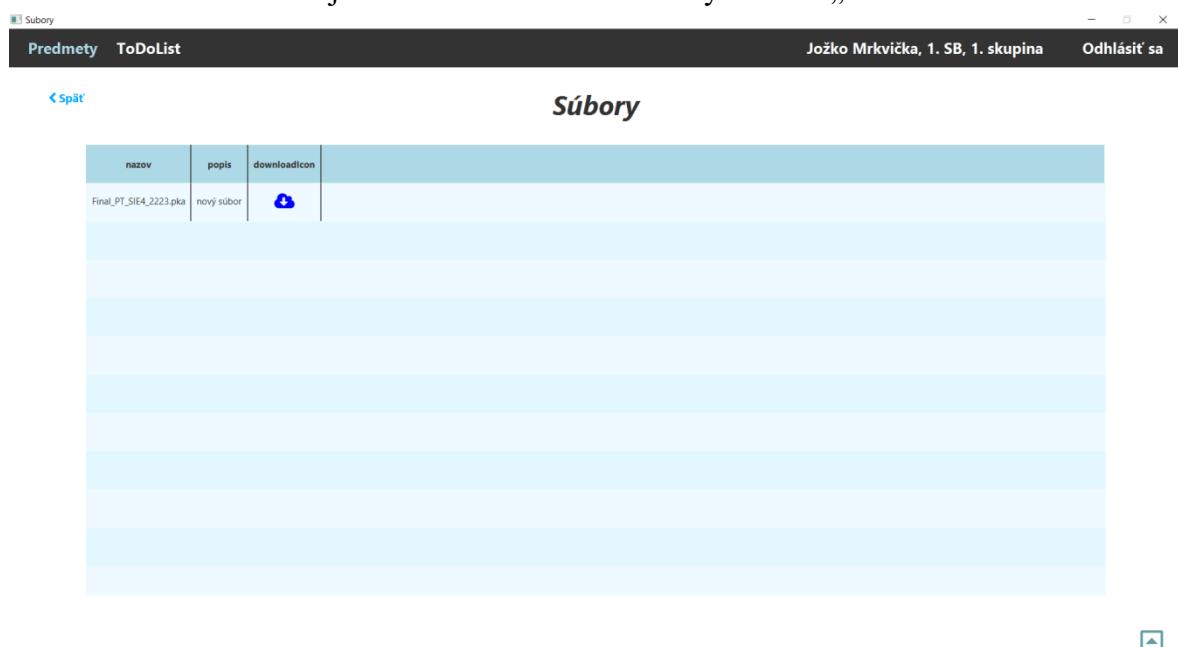
### 3.2.4.7 Študentský panel

Po prihlásení ako študent sa dostaneme do študentského panela, kde sa nám zobrazia všetky predmety, ktoré sa učí daný žiak. Po kliknutí na modrú šípku sa presunie o kartu nižšie (Predmety → Témy → Podtémy → Súbory).



Obrázok 29: Študentský panel (karta „Predmety“) v desktopovej aplikácii

Na karte súbory si môže študent stiahnuť poskytované súbory učiteľom a uložiť na ľubovoľné miesto na svojom disku kliknutím na modrý obláčik „download“.



Obrázok 30: Študentský panel (karta „Súbory“) v desktopovej aplikácii

V študentskom a učiteľskom paneli sa taktiež nachádza karta „ToDoList“, kde si užívatelia môžu zapisovať úlohy, ktoré im treba urobiť v najbližšom čase.

The screenshot shows a desktop application window titled "student todolist". The top navigation bar includes links for "Predmety" and "ToDoList", and displays the user information "Jožko Mrkvíčka, 1. SB, 1. skupina" along with a "Odhlásiť sa" button. The main content area is titled "ToDoList" and contains a table with four columns: "task", "dueDate", "dueTime", and "deleteIcon". There are two rows of data:

task	dueDate	dueTime	deleteIcon
Nová úloha 1	02.05.2023	12:00	
Nová úloha 2	25.02.2023	15:00	

Below the table is a blue "Pridať" (Add) button. In the bottom right corner of the application window, there is a small square icon with an upward-pointing arrow.

Obrázok 31: Študentský panel (karta „ToDoList“) v desktopovej aplikácii

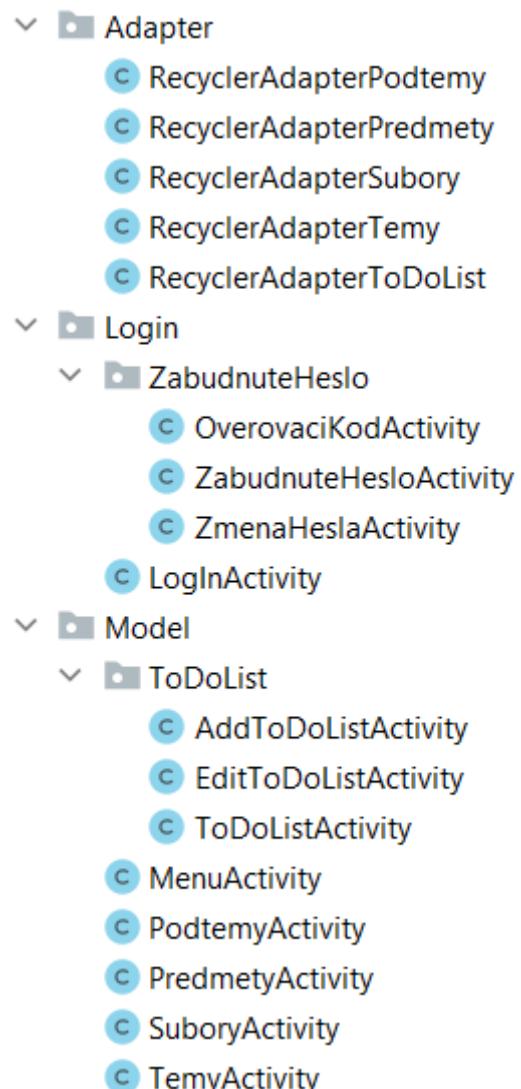
### 3.3 Mobilná aplikácia

Mobilná aplikácia je vytvorená len pre študentov. Ponúka možnosť zmeny zabudnutého hesla, prehľadávanie tém, podtém a súborov, ktoré boli zazdieľané učiteľom a ToDoList, ktorý je synchronizovaný pomocou databázy s mobilnou a desktopovou verziou aplikácie.

Táto aplikácia, podobne ako desktopová aplikácia, využíva architektúru MVC (Model, View, Controller).

Mobilná aplikácia je naprogramovaná v aplikácii Android Studio, v jazyku Java (Model a Controller) a v jazyku XML (View).

#### 3.3.1 Súborová štruktúra



Obrázok 32: Súborová štruktúra v mobilnej aplikáции

### 3.3.2 Sekcia Login

Súborová štruktúra sa je rozdelená na 3 časti. Prvou časťou je prihlásование (Login). Táto časť sa skladá zo 4 súborov (OverovaciKodAktivity, ZabudnuteHesloActivity, ZmenaHeslaActivity a LogInActivity). Súbor LogInActivity je hlavný spúšťací súbor aplikácie (po spustení aplikácie sa spustí ako prvý). Spolu s jeho view-om zabezpečuje prihlásование užívateľa do systému.



```
// získanie url adresy k serveru zo shared preferences
SharedPreferences sh = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
String shared_url = sh.getString( s: "url", s1: "");

// poslanie requestu na server (kontrola emalu a hesla s datami z databazy)
String url_connect = shared_url + "selectLogin.php?login=" + "login";
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_connect, response -> {
    try {
        JSONArray data = new JSONArray(response);
        for (int i = 0; i < data.length() ; i++) {
            JSONObject object = data.getJSONObject(i);
            String id = object.getString( name: "id");
            String login = object.getString( name: "login");
            String heslo = object.getString( name: "heslo");
            String typ_uctu = object.getString( name: "typ_uctu");

            // kontrola emalu a hesla
            if (login.equals(editTextEmailAddress.getText().toString()) && heslo.equals(editTextPassword.getText().toString())) {

                // uloženie uctu prihláseneho do shared preferences
                SharedPreferences sharedPreferences = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
                SharedPreferences.Editor myEdit = sharedPreferences.edit();
                myEdit.putString( s: "id_uctu", id);
                myEdit.commit();

                // prepnutie scény
                Intent intent = new Intent(getApplicationContext(), PredmetyActivity.class);
                startActivity(intent);
                break;
            }
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
});
```

Obrázok 33: Zdrojový kód triedy „LogInActivity“ (overenie zadaného e-mailu a hesla) v mobilnej aplikácii

Ďalšie 3 súbory z prvej časti zabezpečujú zmenu zabudnutého hesla. Ako prvý sa spustí súbor ZabudnuteHesloActivity, ktorý sa stará o získanie e-mailovej adresy od užívateľa, overenie jej výskytu v databáze a následné zaslanie e-mailu s vygenerovaným overovacím kódom.

```

// poslanie requestu na server (kontrola emalu)
String url_connect = shared_url + "overEmail.php?email=" + email;
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_connect, response -> {
    try {
        JSONArray data = new JSONArray(response);
        for (int i = 0; i < data.length() ; i++) {
            JSONObject object = data.getJSONObject(i);
            String id = object.getString( name: "id");
            String login = object.getString( name: "login");
            String heslo = object.getString( name: "heslo");
            String typ_uctu = object.getString( name: "typ_uctu");

            // ak sa email zhoduje
            if (login.equals(email)) {

                // zapisanie loginu (emailu) do shared preferences
                SharedPreferences sharedPreferences = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
                SharedPreferences.Editor myEdit = sharedPreferences.edit();
                myEdit.putString( s: "login", login);
                myEdit.commit();

                // odoslanie vygenerovaneho overovacieho kodu na email uzivatela
                odosliEmail(email, generateCode());

            } else Toast.makeText( context: this, text: "Nesprávny e-mail", Toast.LENGTH_SHORT).show();
        }
    }
}

```

**Obrázok 34: Zdrojový kód triedy „ZabudnuteHesloActivity“ (overenie výskytu zadanej e-mailovej adresy v databáze) v mobilnej aplikácii**

```

// ziskanie url adresy k serveru zo shared preferences
SharedPreferences sh = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
String shared_url = sh.getString( s: "url", s1: "");

// poslanie requestu na server (odoslanie overovacieho kodu na email)
String url_connect = shared_url + "sendMail.php?email=" + email + "&code=" + code;
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_connect, response -> {

    Log.i( tag: "tagconvertstr", msg: "[" +response+"]");
    Toast.makeText( context: this, response, Toast.LENGTH_SHORT).show();

    // ulozenie vygenerovaneho overovacieho kodu do shared preferences
    SharedPreferences sharedPreferences = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
    SharedPreferences.Editor myEdit = sharedPreferences.edit();
    myEdit.putString( s: "code", code);
    myEdit.commit();

    // prepnutie sceny
    Intent intent = new Intent(getApplicationContext(), OverovaciKodActivity.class);
    startActivity(intent);
}

```

**Obrázok 35: Zdrojový kód triedy „ZabudnuteHesloActivity“ (odoslanie e-mailu s overovacím kódom) v mobilnej aplikácii**

Po zaslaní e-mailu prichádza na rad OverovaciKodActivity, ktorá sa stará o overenie kódu od užívateľa.

```
// nastavenie eventu po kliknutí na tlačidlo
overKod.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // získanie vygenerovaného kódu zo shared preferences
        SharedPreferences sh = getSharedPreferences( name: "MySharedPref", MODE_PRIVATE);
        String code = sh.getString( s: "code", s1: "");

        // overenie kódu
        if (code.equals(overovaciKod.getText().toString())) {

            // prepnutie scény
            Intent intent = new Intent(getApplicationContext(), ZmenaHeslaActivity.class);
            startActivity(intent);

        } else Toast.makeText( context: OverovaciKodActivity.this, text: "nesprávny kód", Toast.LENGTH_SHORT).show();
    }
});
```

**Obrázok 36: Zdrojový kód triedy „OverovaciKodActivity“ (overenie overovacieho kódu zadaného užívateľom) v mobilnej aplikácii**

Ak užívateľ zadá správny kód, je vpustený do aktivity ZmenaHeslaActivity, kde si môže zmeniť svoje heslo, ktoré sa aktualizuje v databáze.

```
String url = shared_url + "updateHeslo.php?heslo=" + heslo + "&login=" + login;
RequestQueue myRequest = Volley.newRequestQueue( context: this);
StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Log.i( tag: "tagconvertstr", msg: "["+response+"]");
        Toast.makeText(getApplicationContext(), response, Toast.LENGTH_LONG).show();
    }
}, new Response.ErrorListener() {
    2 usages
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_LONG).show();
    }
});
{
    4 usages
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> map = new HashMap<>();

        // odoslanie údajov na server
        map.put( k: "heslo", heslo);
        map.put( k: "login", login);

        return map;
    }
};
myRequest.add(stringRequest);
```

**Obrázok 37: Zdrojový kód triedy „ZmenaHeslaActivity“ (aktualizácia hesla) v mobilnej aplikácii**

Ked' už je reč o heslách, bolo by dobré spomenúť, že systém heslá hashuje pomocou algoritmu SHA256 a tento vygenerovaný hash hesla ukladá do databázy. Je to kvôli tomu, že heslá sa radia medzi citlivé dátu a často ich útočníci odchytávajú. Takto sa internetom neposiela heslo v textovej podobe, ale oproti databáze sa overuje len hash tohto hesla, ktorý je útočníkovi, ktorý ho odchytí zbytočný, keďže sa jedná o jednosmerný algoritmus (nedá sa z hashu dostať heslo späťne).

```
// metoda, ktorá zabezpečuje zahashovanie hesla
private static String hash(String textToHash) {
    MessageDigest digest = null;
    try {
        digest = MessageDigest.getInstance(algorithm: "SHA-256");
    } catch (NoSuchAlgorithmException e) {
        throw new RuntimeException(e);
    }
    String text = textToHash;
    byte[] hash = digest.digest(text.getBytes(StandardCharsets.UTF_8));
    String hashedText = bytesToHex(hash);

    return hashedText;
}

// metoda, ktorá zabezpečuje transformáciu bytov na hash
1 usage
private static String bytesToHex(byte[] hash) {
    StringBuilder hexString = new StringBuilder(capacity: 2 * hash.length);
    for (int i = 0; i < hash.length; i++) {
        String hex = Integer.toHexString(i: 0xff & hash[i]);
        if(hex.length() == 1) {
            hexString.append('0');
        }
        hexString.append(hex);
    }
    return hexString.toString();
}
```

Obrázok 38: Zdrojový kód (metóda, ktorá zabezpečuje hashovanie hesla) v mobilnej aplikácii

### 3.3.3 Sekcia Model

Druhou časťou našej súborovej štruktúry je sekcia Model. Triedy z tejto sekcie zabezpečujú hlavne získavanie údajov z databázy a ich presúvanie do tried zo sekcie Adapter (Controller).

```
// poslanie requestu na server (vyberie tem z databazy)
String url_connect = shared_url + "selectTema.php?id_trieda=" + id_trieda_studenta + "&id_skupina=" + id
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_connect, response -> {
    try {
        Log.i( tag: "tagconvertstr", msg: "[" + response + "]");
        JSONArray data = new JSONArray(response);
        for (int i = 0; i < data.length(); i++) {
            JSONObject object = data.getJSONObject(i);
            String id = object.getString( name: "id");
            String id_ucitel = object.getString( name: "id_ucitel");
            String id_predmet = object.getString( name: "id_predmet");
            String id_trieda = object.getString( name: "id_trieda");
            String id_skupina = object.getString( name: "id_skupina");
            String je_skryta = object.getString( name: "je_skryta");
            String nazov = object.getString( name: "nazov");

            // osetenie, aby sa zobrazovali len temy, ktore nastavil učiteľ ako viditeľné pre studenta
            if (!je_skryta.equals("1")) {
                // pridanie údajov o temach do arraylistov
                temyList.add(nazov);
                idList.add(id);
            }
        }
    }

    // upozornenie recycleradaptera na zmenu v listoch
    recyclerAdapter.notifyItemRangeInserted(temyList.size(), temyList.size());
    recyclerAdapter.notifyItemRangeInserted(idList.size(), idList.size());
}
```

Obrázok 39: Zdrojový kód triedy z Modiu v mobilnej aplikácii

Triedy PredmetyActivity, TemyActivity, PodtemyActivity a SuboryActivity slúžia pre získavanie údajov z datbázy o jednotlivých predmetoch, témach, podtémach alebo súboroch. Trieda MenuActivity slúži ako navigačný panel medzi zdieľanými prvками od učiteľa, ToDoListom a možnosťou odhlásiť sa.

```

// nastavenie eventu po kliknuti na tlacidlo
logOut.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // prepnutie sceny
        Intent intent = new Intent(getApplicationContext(), LogInActivity.class);
        startActivity(intent);
    }
});

// nastavenie eventu po kliknuti na tlacidlo
predmety.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // prepnutie sceny
        Intent intent = new Intent(getApplicationContext(), PredmetyActivity.class);
        startActivity(intent);
    }
});

// nastavenie eventu po kliknuti na tlacidlo
ToDoList.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        // prepnutie sceny
        Intent intent = new Intent(getApplicationContext(), ToDoListActivity.class);
        startActivity(intent);
    }
});

```

**Obrázok 40:** Zdrojový kód triedy `MenuActivity` (akcie po kliknutí na jednotlivé komponenty v mobilnej aplikácii)

Triedy, ktoré sa starajú o `ToDoList` sú `ToDoListActivity`, `EditToDoListActivity` a `AddToDoListActivity`. Trieda `AddToDoListActivity` sa stará o pridávanie položiek do `ToDoListu`.

```

// poslanie requestu na server (pridanie noveho tasku do databazy)
String url = shared_url + "insertTask.php";
RequestQueue myRequest = Volley.newRequestQueue( context: this);
StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Log.i( tag: "tagconvertstr", msg: "["+response+"]");
        Toast.makeText(getApplicationContext(), response, Toast.LENGTH_LONG).show();
    }
}, new Response.ErrorListener() {
    2 usages
    @Override
    public void onErrorResponse(VolleyError error) {
    }
}){

    4 usages
    @Nullable
    @Override
    protected Map<String, String> getParams() throws AuthFailureError {
        Map<String, String> map = new HashMap<~>();
        // nastavenie udajov, ktoré posielame na server
        map.put( k: "task", task);
        map.put( k: "dueDate", dueDate);
        map.put( k: "dueTime", dueTime);
        map.put( k: "order", order);
        map.put( k: "done", done);
        map.put( k: "id_ucet", id_ucet);
        return map;
    }
};


```

**Obrázok 41: Zdrojový kód triedy „AddToDoListActivity“ (pridatie položiek) v mobilnej aplikácii**

Trieda EditToDoListActivity sa stará o upravovanie položiek v ToDoListe alebo o ich vymazanie.

```

// poslanie requestu na server (vymazanie tasku)
String url = shared_url + "deleteTask.php";
RequestQueue myRequest = Volley.newRequestQueue( context: this);
StringRequest stringRequest = new StringRequest(Request.Method.POST, url, new Response.Listener<String>() {
    @Override
    public void onResponse(String response) {
        Toast.makeText(getApplicationContext(), response, Toast.LENGTH_LONG).show();
        // prepnutie sceny
        Intent intent = new Intent(getApplicationContext(), ToDoListActivity.class);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
        startActivity(intent);
    }
}, new Response.ErrorListener() {
    2 usages
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_LONG).show();
    }
}){

    4 usages
    @Override
    protected Map<String, String> getParams() throws AuthFailureError{
        Map<String, String> map = new HashMap<~>();
        // nastavenie udajov, ktoré posielame na server
        map.put( k: "id", id);
        return map;
    }
};

myRequest.add(stringRequest);


```

**Obrázok 42: Zdrojový kód triedy „EditToDoListActivity“ (vymazávanie položiek) v mobilnej aplikácii**

Trieda ToDoListActivity zabezpečuje vypisovanie položiek ToDoListu v Recycler View-e a možnosť ich presúvať systémom drag & drop.

```
// poslanie requestu na server (vybratie informácií o taskoch z databazy)
String url_connect = shared_url + "selectTask.php?id_ucet=" + id_ucet_student;
StringRequest stringRequest = new StringRequest(Request.Method.POST, url_connect, response -> {
    try {
        Log.i( tag: "tagconvertstr", msg: "["+response+"]");
        JSONArray data = new JSONArray(response);
        for (int i = 0; i < data.length() ; i++) {
            JSONObject object = data.getJSONObject(i);
            String id = object.getString( name: "id");
            String task = object.getString( name: "task");
            String dueDate = object.getString( name: "dueDate");
            String dueTime = object.getString( name: "dueTime");
            String order = object.getString( name: "order");
            String done = object.getString( name: "done");
            String id_ucet = object.getString( name: "id_ucet");

            // naplnenie arraylistov hodnotami
            taskList.add(task);
            dueDateList.add(dueDate);
            dueTimeList.add(dueTime);
            idList.add(id);
            orderList.add(order);
        }

        // upozornenie recycleradaptera na zmenu v listoch
        recyclerAdapter.notifyItemRangeInserted(taskList.size(), taskList.size());
        recyclerAdapter.notifyItemRangeInserted(dueDateList.size(), dueDateList.size());
        recyclerAdapter.notifyItemRangeInserted(dueTimeList.size(), dueTimeList.size());
        recyclerAdapter.notifyItemRangeInserted(idList.size(), idList.size());
    }
}
```

Obrázok 43: Zdrojový kód triedy „ToDoListActivity“ (vypisovanie položiek) v mobilnej aplikácii

```
// metoda, ktorá zabezpečuje presuvanie itemov
4 usages
@Override
public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull RecyclerView.ViewHolder viewHolder,
                      int fromPosition, int toPosition) {
    // nastavenie vymeny pozícii jednotlivých ítemov v arraylistoch
    Collections.swap(taskList, fromPosition, toPosition);
    Collections.swap(dueDateList, fromPosition, toPosition);
    Collections.swap(dueTimeList, fromPosition, toPosition);
    Collections.swap(orderList, fromPosition, toPosition);
    Log.i( tag: "tagconvertstr", msg: "["+idList+"]");
    Collections.swap(idList, fromPosition, toPosition);
    Log.i( tag: "tagconvertstr", msg: "["+idList+"]");

    // upozornenie recycleradaptera na zmenu pozicie ítemu
    recyclerView.getAdapter().notifyItemMoved(fromPosition, toPosition);

    // aktualizácia poradia taskov v databaze
    updateOrderinDatabase(idList);

    return false;
}
```

Obrázok 44: Zdrojový kód triedy „ToDoListActivity“ (drag & drop systém) v mobilnej aplikácii

### 3.3.4 Sekcia Adapter

Poslednou časťou súborovej štruktúry mobilnej aplikácie je sekcia Adapter. Triedy tejto sekcie slúžia pre komunikáciu medzi užívateľom a Recycler View-om. Recycler View je komponent, ktorý slúži na zobrazenie zoznamov položiek v našej aplikácii. V ňom sú uložené a zobrazené všetky dátá z databázy. Umožňuje ak klikanie na jednotlivé položky zoznamu.

```
// metoda, ktorá slúži na vykonanie akcie po kliknutí na dany item
@Override
public void onClick(View view) {
    Toast.makeText(context, text: taskList.get(getAdapterPosition()) + " " + idList.get(getAdapterPosition()),

    // uloženie údajov do shared preferences
    SharedPreferences sharedPreferences = context.getSharedPreferences(s: "MySharedPref", MODE_PRIVATE);
    SharedPreferences.Editor myEdit = sharedPreferences.edit();
    myEdit.putString(s: "task", taskList.get(getAdapterPosition()));
    myEdit.putString(s: "dueDate", dueDateList.get(getAdapterPosition()));
    myEdit.putString(s: "dueTime", dueTimeList.get(getAdapterPosition()));
    myEdit.putString(s: "id_task", idList.get(getAdapterPosition()));
    myEdit.putString(s: "order", orderList.get(getAdapterPosition()));
    myEdit.commit();

    // zmena scény
    Intent intent = new Intent(context, EditToDoListActivity.class);
    intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK | Intent.FLAG_ACTIVITY_MULTIPLE_TASK);
    context.startActivity(intent);
}

}
```

**Obrázok 45: Zdrojový kód triedy zo sekcie Adapter (vykonanie akcie po kliknutí na položku v zozname) v mobilnej aplikácii**

```
// metoda, ktorá slúži na zobrazenie zoznamu položiek v recycler view-e (z akého xml suboru sa ma brať "vzhľad" položiek)
// anotácia "@NotNull" pred parametrom "parent" znamená, že sa očakáva, že tato hodnota nikdy nebude null.
@NotNull
@Override
public ViewHolder onCreateViewHolder(@NotNull ViewGroup parent, int viewType) {

    LayoutInflator layoutInflater = LayoutInflator.from(parent.getContext());
    View view = layoutInflater.inflate(R.layout.row_item_todolist, parent, attachToRoot: false);
    ViewHolder viewHolder = new ViewHolder(view);
    return viewHolder;
}

// tato metoda nastaví textové hodnoty pre TextView elementy vo ViewHolder objekte,
// ktorý reprezentuje položku zoznamu na danom indexe (pozicii).
@Override
public void onBindViewHolder(@NotNull ViewHolder holder, int position) {
    holder.nazovTask.setText(taskList.get(position));
    holder.dueDate.setText(dueDateList.get(position));
    holder.dueTime.setText(dueTimeList.get(position));
    holder.id.setText(idList.get(position));
}
```

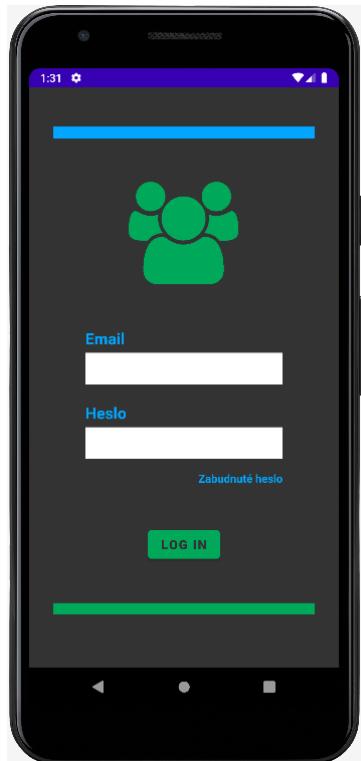
**Obrázok 46: Zdrojový kód triedy zo sekcie Adapter (zobrazenie zoznamu položiek a nastavenie ich hodnôt) v mobilnej aplikácii**

### **3.3.5 View a rýchly návod používania aplikácie (ukážka funkcia)**

View mobilnej aplikácie je tvorený jednotlivými XML súbormi. Každý súbor má svoj controller, ktorý sa stará o interakciu s užívateľom a vypisovanie dát.

#### **3.3.5.1 Prihlásovací panel**

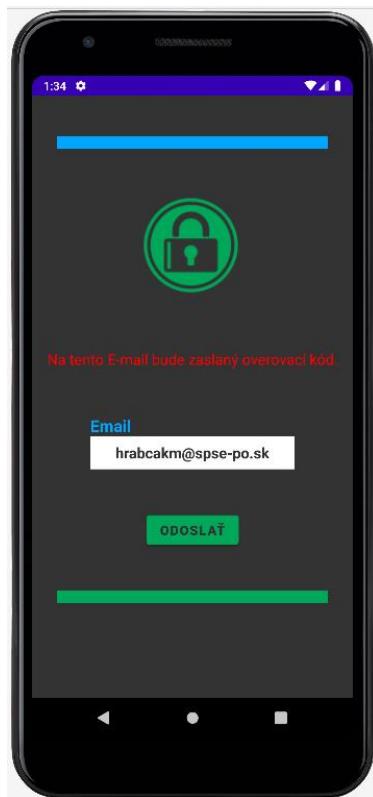
Po spustení aplikácie sa nám zobrazí prihlásovací panel, pomocou ktorého sa môžeme prihlásiť do systému, zmeniť si zabudnuté heslo alebo zaregistrovať školu.



**Obrázok 47: Prihlásovací panel  
v mobilnej aplikácii**

### 3.3.5.2 Zabudnuté heslo

Pod prihlásovacím formulárom sa nachádza text „Zabudnuté heslo“. Po kliknutí na tento text sa zobrazí okno, kde musíme emailovú adresu účtu zaregistrovaného v systéme.



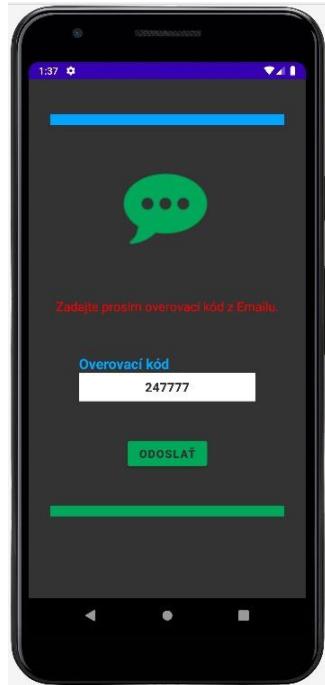
Obrázok 48: Panel pre odoslanie  
overovacieho kódu na e-mail v  
mobilnej aplikácii

Po zadaní emailovej adresy a kliknutí na tlačidlo „Odoslať“ nám systém pošle na zadanú emailovú adresu email s overovacím kódom.



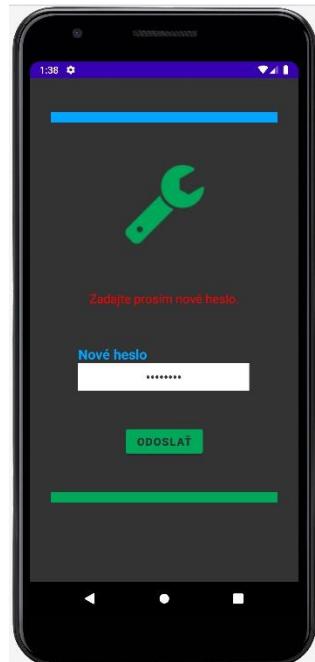
Obrázok 49: Zaslaný e-mail s overovacím kódom

Overovací kód následne zadáme do nasledujúceho okna.



**Obrázok 50:** Panel pre  
overenie hesla v mobilnej  
aplikácii

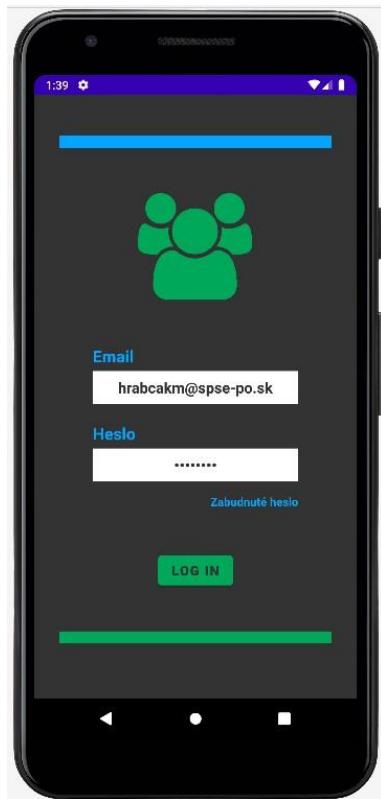
Po overení kódu nám systém dovolí zmeniť heslo.



**Obrázok 51:** Panel pre  
zmenu hesla v mobilnej  
aplikácii

### **3.3.5.3 Prihlásenie**

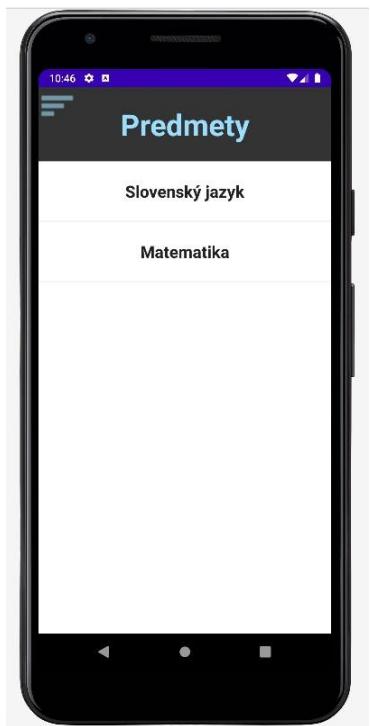
Pre prihlásenie sa do systému musíme zadať email a heslo od nášho účtu.



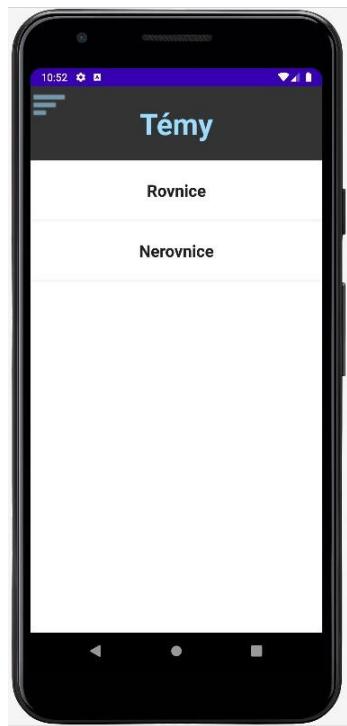
**Obrázok 52:** Prihlasovací panel  
v mobilnej aplikácii

### **3.3.5.4 Študentský panel**

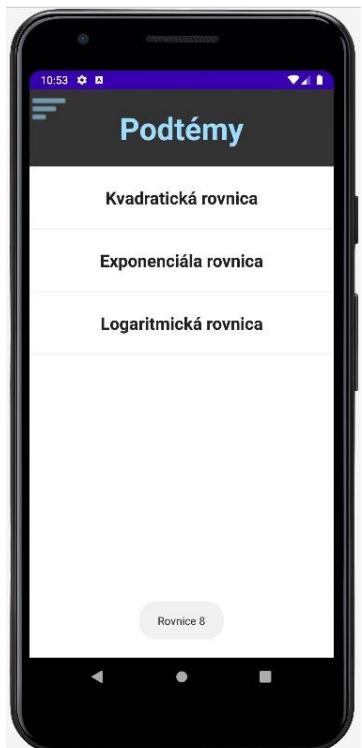
Po prihlásení ako študent sa dostaneme do študentského panela, kde sa nám zobrazia všetky predmety, ktoré sa učí daný žiak. Po kliknutí na daný predmet sa presunie o kartu nižšie (Predmety → Témy → Podtémy → Súbory).



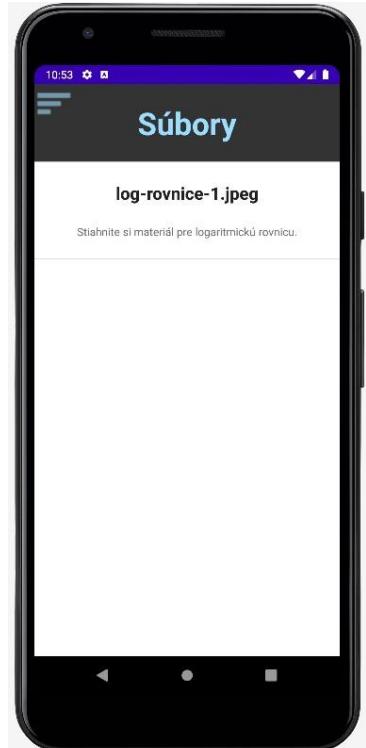
Obrázok 56: Študentský panel (Predmety) v mobilnej aplikácii



Obrázok 55: Študentský panel (Témy) v mobilnej aplikácii

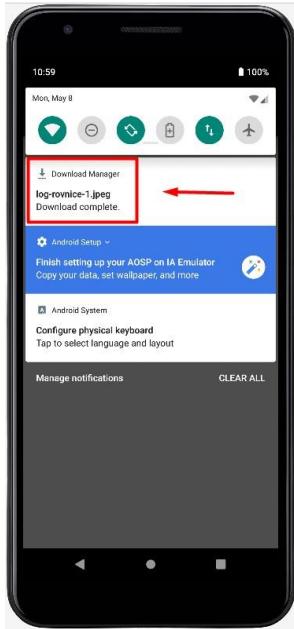


Obrázok 54: Študentský panel (Podtémy) v mobilnej aplikácii



Obrázok 53: Študentský panel (Súbory) v mobilnej aplikácii

Na karte súbory si môže študent stiahnuť poskytované súbory učiteľom a uložiť do svojho zariadenia kliknutím na daný súbor.



**Obrázok 57: Študentský panel – stiahnutie súboru v mobilnej aplikácii**

### 3.3.5.5 Karta Menu

Po kliknutí na ikonku „hamburger“ v ľavej hornej časti obrazovky sa užívateľ môže prepnúť na kartu menu. Táto karta slúži ako navigačný panel medzi zdieľanými prvkami od učiteľa, ToDoListom a možnosťou odhlásiť sa.



**Obrázok 58: Karta „Menu“ v mobilnej aplikácii**

### 3.3.5.6 Karta ToDoList

Na karte „ToDoList“ si užívatelia môžu zapisovať úlohy, ktoré im treba urobiť v najbližšom čase.



Obrázok 59: Katra  
„ToDoList“ v mobilnej  
aplikácii

Po kliknutí na ikonku „plus“ v pravom hornom rohu obrazovky si užívateľ môže pridať ďalšiu úlohu.



Obrázok 60: Pridávanie  
novej úlohy do  
ToDoListu v mobilnej  
aplikácii

Po kliknutí na danú úlohu môžeme editovať jej obsah alebo ju vymazat.



**Obrázok 61: Editácia  
a vymazanie úlohy  
z ToDoListu v mobilnej  
aplikácii**

Dlhším pridržaním prstu na danej téme ju vieme presúvať a meniť jej pozíciu a usporiadanie úloh.

## 3.4 Webová aplikácia

Webová aplikácia je jednou z troch aplikácií, ktoré náš projekt ponúka. Predstavuje hlavné miesto pre správcov školy, ale aj správcov stránky. Ponúka možnosť administrácie všetkých škôl, účtov, ale aj priestor pre školských administrátorov na správu ich konkrétnej školy. Administrátori, rovako ako učitelia, dokážu vytvárať témy a podtémy pre žiakov. Učitelia tiež dokážu zverejňovať a sprístupňovať súbory pre svojich žiakov v daných predmetoch.

Stránka bola zhotovená pomocou HTML, CSS, JavaScript-u (konkrétnie pomocou knižnice JQuery, ktorá poskytuje možnosť ľivej interakcie s databázou bez potreby znovunačítania webstránky) a PHP. Na stylizáciu stránky bola použitá knižnica Bootstrap.

### 3.4.1 Súbor ConnectToDB.php

Tento súbor je najviac využívaný pri webovej aplikácii, pretože obsahuje funkciu, ktorá otvára spojenie s databázou. Využíva funkciu mysli\_connect();

```
<?php
function returnCon(){
    return mysqli_connect( host: "localhost", user: "root", password: "", database: "notriggers");
}
?>
```

Obrázok 62: Funkcia, ktorá vracia pripojenie k databáze v súbore ConnectToDB.php

### 3.4.2 Súbor SelectDataFunction.php

Tento súbor obsahuje rôzne funkcie, ktoré vracajú hodnoty z databázy potrebné pre administráciu školy. Každá z funkcií má svoje špeciálne využitie, či už pri vytváraní výpisov, hľadaní mien alebo počítaní stĺpcov a dát. V podstate predstavuje Model webovej aplikácie, pretože neinteraguje priamo s grafickými prvkami webovej aplikácia, ale komunikuje s databázou a vracia dátá v potrebných tvaroch.

```

//Funkcia na vrátenie názvov všetkých škôl
function getSchoolName($id_skoly){
    $sql = "SELECT nazov FROM `škola` WHERE id = '$id_skoly'";
    $result = mysqli_query($returnCon(), $sql);
    return mysqli_fetch_array($result);
}

//Funkcia na vrátenie počtu škôl
function getCountSkola(){
    $sql = "SELECT id FROM `škola`";
    $result = mysqli_query($returnCon(), $sql);
    return count(mysqli_fetch_array($result));
}

//Funkcia, ktorá vráti číslo id-čiek školy
function getIDNumsSkola(){
    $sql = "SELECT id FROM `škola`";
    $result = mysqli_query($returnCon(), $sql);
    $arrayNums = array();
    if (mysqli_num_rows($result) > 0){
        while ($row = mysqli_fetch_array($result)){
            $arrayNums[] = $row["id"];
        }
    }
    return $arrayNums;
}

//Funkcia na vrátenie informácií o škole
function getInfoSchool($id_skoly){
    $sql = "SELECT id, nazov, mesto, ulica, cislo_domu, PSC, pocet_uctielov, pocet_studentov, pocet_odborov, pocet_predmetov, e_mail, tel_cislo FROM `škola` WHERE id = '$id_skoly'";
    $result = mysqli_query($returnCon(), $sql);
    return mysqli_fetch_array($result);
}

```

Obrázok 63: Ukážka funkcií, ktoré vracajú hodnoty z databázy

### 3.4.3 Súbor PrintDataFunction.php

Tento súbor poskytuje už predštylizované dátá tabuľky, pripravené pre dynamickú prácu s nimi. S využitím funkcie AJAX-u JQuery, tento súbor realizuje živé premietanie zmien v databáze. Pomocou konkatonácie ku premennej dátového typu String je možné poskladať si vizuálny výpis do tabuľky podľa potreby. Využíva funkcie zo súboru SelectDataFunction.php, ale aj volania funkcií, ktoré sa nachádzajú v súboroch, ktoré už stránku zobrazujú.

```

<?php
require "SelectDataFunctions.php";

$funkcia = $_POST['funkcia'];

$output = "";

//Funkcia, ktorá vyplie do tabuľky názvy škôl.
if ($funkcia == 'printSchoolNames'){
    for ($i = 0; $i < count(getIDNumsSkola()); $i++){
        $output .= '|  |  |  |
| --- | --- | --- |
| '.getSchoolName(getIDNumsSkola()[$i])['nazov'].'   | <i>x</i> | |
';
    }
    echo $output;
}
$_POST = array();

```

Obrázok 64: Ukážka PHP kódu predštylizovaných výpisov do tabuľky

### 3.4.4 Súbor Využitie funkcie AJAX

AJAX(Asynchronous JavaScript and XML) je technológia vývoja webových aplikácií, ktorá umožňuje asynchronné (bez obnovenia celej stránky) komunikovať s webovým serverom a aktualizovať časti webovej stránky dynamicky.

Použitím AJAX-u môžete zabezpečiť, aby užívateľ nemusel na každú akciu na stránke čakať, kým sa celá stránka znova načíta. Týmto spôsobom je možné dosiahnuť plynulejších interakcií so stránkou a vylepšiť užívateľské skúsenosti.

AJAX používa JavaScript na posielanie a prijímanie dát z webového servera, často v formáte XML alebo JSON. V praxi sa však stále viac používa JSON namiesto XML, pretože je jednoduchší na použitie a ľahšie sa spracováva. V tomto projekte sa využíva formát JSON, pretože PHP ho vie pomocou POST metódy prekonvertovať na pole, ktoré je uložené v globálnej premennej `$_POST`.

```
<!-- Tabuľka pre témy -->
<h2 style="..." class="mt-4">Témy</h2>
<table id="skola_temy" class="table">
    <tbody>
        <!-- Výpis témy z DB -->
        <script>
            function printTema(){
                $.ajax({
                    url:"../../php/PrintDataFunction.php",
                    method:"POST",
                    data:{funkcia:'printTema', id_skoly:<?php echo $id_skoly;?>},
                    dataType:"text",
                    success: function(data) {
                        $('.reloadableTema').remove();
                        $('#skola_temy').html(data);
                    }
                });
            }
            printTema();
        </script>
    </tbody>
</table>
```

Obrázok 65: Ukážka využitia funkcie AJAX na dynamický výpis témy na stránke

### 3.4.5 FilesLogic.php

Pomocou tohto súboru je realizované nahrávanie a stiahovanie súborov zo servera. Pri nahrávaní súborov kontroluje typ súboru a veľkosť súboru, aby bola realizovaná kontrola nahrávaných súborov. Pri stiahovaní súborov využíva dátu z databázy, ktorá obsahuje cestu k jednotlivým súborom na serveri. Po kontrole existencie súboru zostaví header pre daný súbor a využije funkciu readfile() na jeho stiahnutie.

```

if (isset($_POST['save'])){
    // Name of the uploaded file
    $filename = $_FILES['myfile']['name'];
    // Destination of the file on the server
    $destination = '../FileUploadDownload/uploads/' . $filename;
    // Get file extension
    $extension = pathinfo($filename, options PATHINFO_EXTENSION);
    // The physical file on a temporary uploads directory on the server
    $file = $_FILES['myfile']['tmp_name'];
    $size = $_FILES['myfile']['size'];

    if(!in_array($extension, ['zip', 'rar', 'pdf', 'docx', 'txt'])) {
        echo "Nesprávny typ súboru";
    } elseif ($_FILES['myfile']['size'] > 5000000){ //File cap 5MB
        echo "Prilis veľký súbor";
    } else{
        // Move the uploaded (temporary) file to the specified destination
        if (move_uploaded_file($file, $destination)) {
            $sql = "INSERT INTO súbory ('id_predmetu','id_temy','id_podtemy','id_ucet','je_skryty','je_ucitelsky','je_studentsky','link') values (4,6,5,10,0,1,0,$filename)";
            if (mysqli_query($conn, $sql)) {
                header( string: "Location: ../php/Subory.php");
            }
        } else {
            echo "Failed to upload file.";
        }
    }
}

```

Obrázok 67: Ukážka PHP kódu pre nahrávanie súborov na server

```

if (isset($_GET['file_id'])) {
    $id = $_GET['file_id'];

    // fetch file to download from database
    $sql = "SELECT * FROM súbory WHERE id=$id";
    $result = mysqli_query($conn, $sql);

    $file = mysqli_fetch_assoc($result);
    $filepath = '../FileUploadDownload/uploads/' . $file['link'];
    if (file_exists($filepath)) {
        header( string: 'Content-Description: File Transfer');
        header( string: 'Content-Type: application/octet-stream');
        header( string: 'Content-Disposition: attachment; filename=' . basename($filepath));
        header( string: 'Expires: 0');
        header( string: 'Cache-Control: must-revalidate');
        header( string: 'Pragma: public');
        header( string: 'Content-Length: ' . filesize($filepath));
        ob_clean();
        flush();
        readfile($filepath);
    }
}

```

Obrázok 66: Ukážka PHP kóde pre stiahovanie súborov zo servera

### 3.4.6 Trieda PHPMailer

Pomocou tejto PHP triedy je realizované posielanie e-mailov. Inštancia tejto triedy potrebuje nastaviť rôzne vlastnosti ako protokol, prihlásovacie údaje pre účet, ktorý bude maily posielat, ale aj samotné telo e-mailu. V našom projekte využívame služby Gmail a SMTP protokol na posielanie e-mailov.

```
$mail = new PHPMailer(TRUE);

try {
    //Server settings
    $mail->isSMTP();
    $mail->Host = 'smtp.gmail.com';
    $mail->SMTPAuth = true;
    $mail->Username = 'rpjservice202223@gmail.com';
    $mail->Password = 'ebwvelogurvmirbb';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
    $mail->Port = 465;
    //Recipients
    $mail->setFrom('RPJ@servicemail.com', 'Mailer');
    $mail->addAddress($login, 'Joe User');
    //Add a recipient
    //Content
    $mail->isHTML(true);
    $mail->Subject = 'Vas Novy Administratorsky Ucet';
    $mail->Body = 'Login: ' . $login . '  
 Heslo: ' . $heslo . '';
    $mail->AltBody = 'Login: ' . $login . ' <br> Heslo: ' . $heslo . '';
    $mail->send();
} catch (Exception $e) {
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
}
```

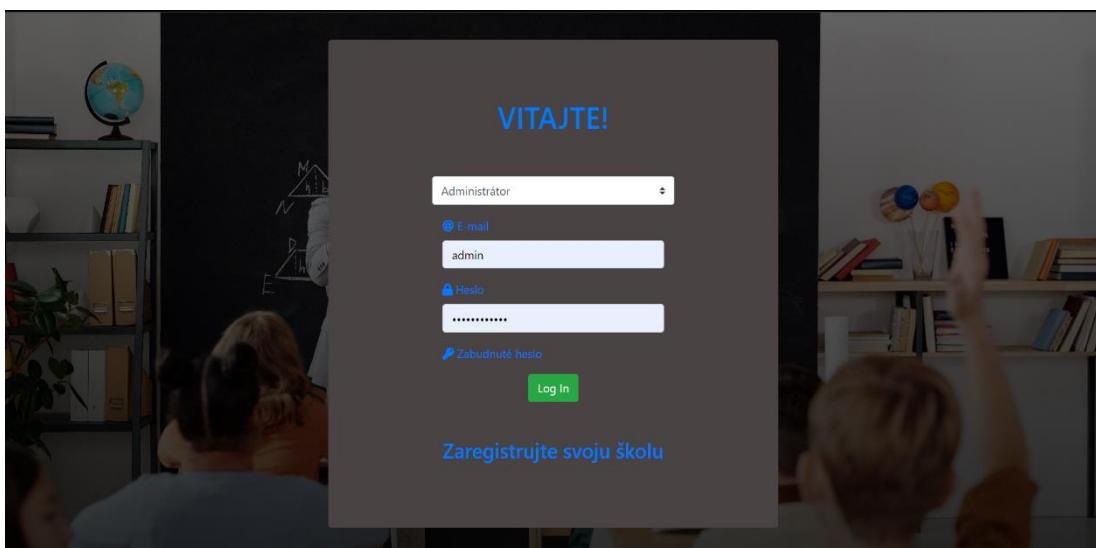
Obrázok 68: Ukážka využitia triedy PHPMailer na poslanie používateľských údajov e-mailom po registrácii nového učiteľského účtu

### 3.4.7 View a rýchly návod používania aplikácie (ukážka funkcia)

View webovej aplikácie resp. webstránky je tvorený jednotlivými PHP súbormi, ktoré dokážu okrem vykonávania PHP kódu na strane servera aj zobrazovať HTML značky ako normálny HTML súbor. Tieto HTML značky, ale aj JavaScript, resp. JQuery kód sa starajú o interakciu s užívateľom a zobrazovanie prijatých dát.

#### 3.4.7.1 Prihlásovací panel

Po načítaní stránky sa nám zobrazí prihlásovací panel, pomocou ktorého sa môžeme prihlásiť do systému, zmeniť si zabudnuté heslo alebo zaregistrovať školu.



Obrázok 69: Ukážka prihlásovacieho panela webstránky

#### 3.4.7.2 Registrácia školy

Celkom dole, posledný link obsahuje text „Zaregistrujte svoju školu“. Po kliknutí na tento link nás presmeruje na formulár, v ktorom vyplníme základné informácie o škole, ktoré budú e-mailom zaslané administrátorovi systému.

**Zaregistrujte svoju školu**

Názov školy

Mesto

Ulica

Číslo domu

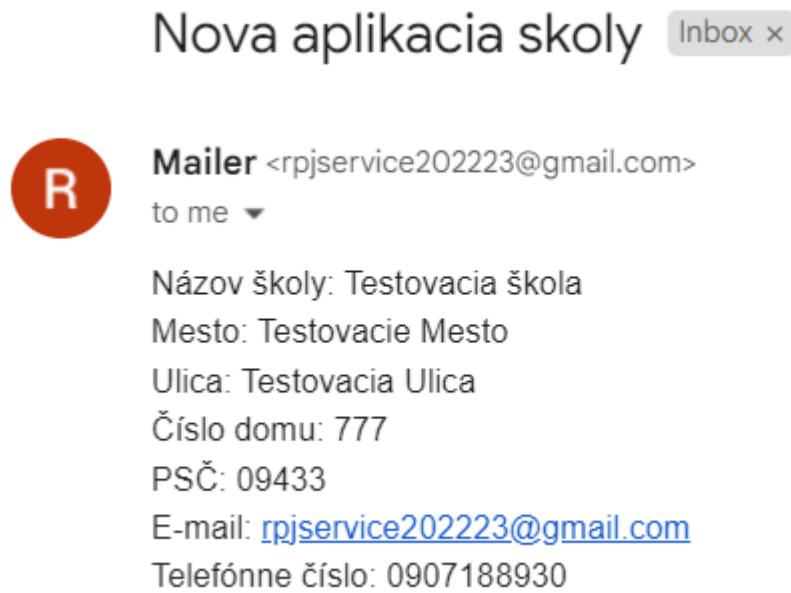
PSČ

E-mail

Telefónne číslo

**Obrázok 70: Ukážka formulára pre registráciu školy**

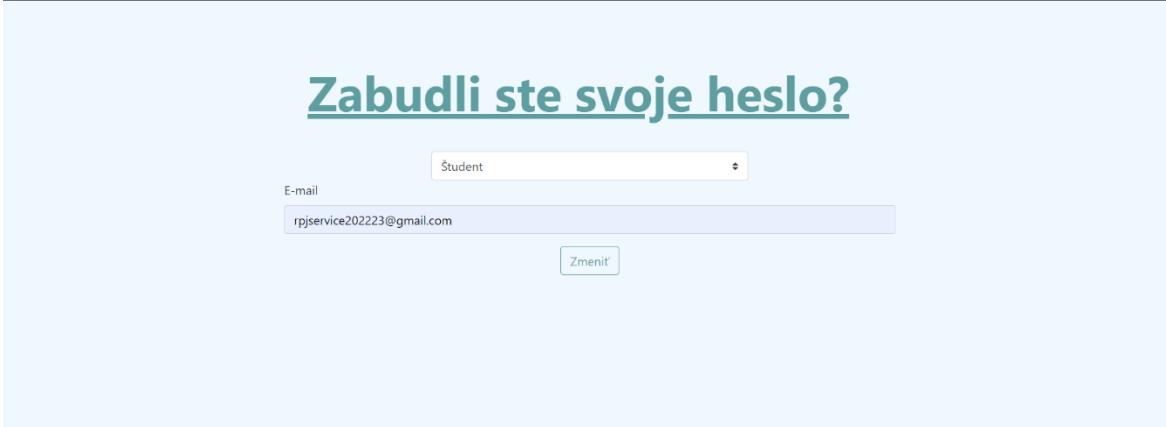
Dáta z tohto formulára budú zobrazené v tabuľke v tele e-mailu pre administrátora systému.



**Obrázok 71: Ukážka stránkou vygenerovaného e-mailu pre registráciu školy administrátorovi systému**

### 3.4.7.3 Zabudnuté heslo

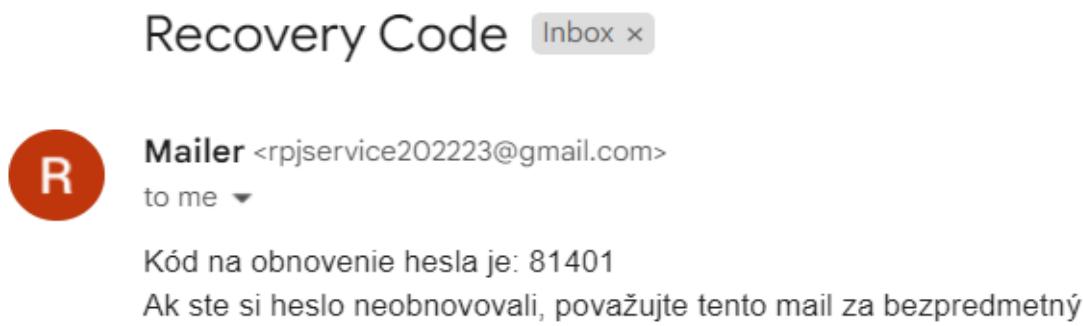
Pod prihlásovacím formulárom sa nachádza link s textom „Zabudnuté heslo“. Po kliknutí na tento link nás presmeruje na podstránku, v ktorej musíme zadať typ účtu (študent, administrátor alebo učiteľ) a e-mail adresu účtu zaregistrovanú v systéme.



The screenshot shows a web-based password recovery form. At the top, it says "Zabudli ste svoje heslo?". Below that, there's a dropdown menu set to "Student". An input field contains the email address "rpjservice202223@gmail.com". At the bottom of the form is a blue "Zmeniť" (Change) button.

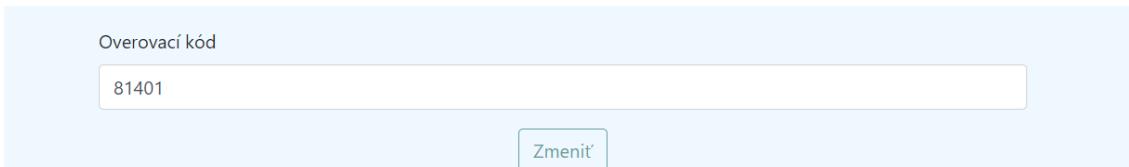
Obrázok 72: Ukážka formulára pre odoslanie e-mailu na obnovu hesla

Po zadaní e-mail adresy a kliknutí na tlačidlo „Zmeniť“ nám systém odošle na zadanú adresu e-mail s overovacím kódom.



Obrázok 73: Ukážka stránkou vygenerovaného e-mailu s kódom pre obnovu hesla

Overovací kód následne zadáme do prislúchajúceho políčka.



The screenshot shows a web-based verification form. It has an input field labeled "Overovací kód" containing the value "81401". Below the input field is a blue "Zmeniť" (Change) button.

Obrázok 74: Ukážka formulára pre overenie kódu na obnovu hesla

Po overení kódu nám systém dovolí zmeniť heslo.

Nové heslo

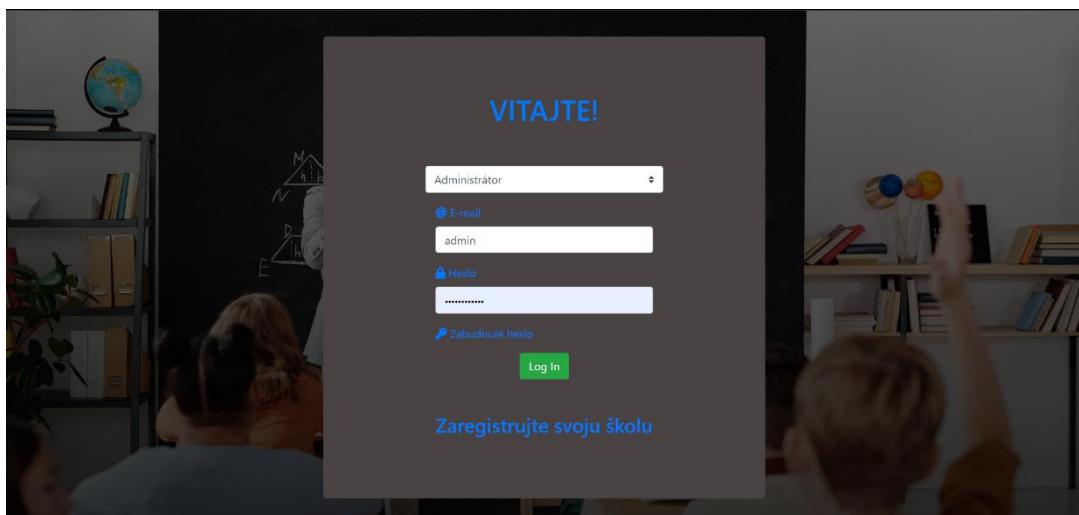
Nové heslo...

Zmeniť

Obrázok 75: Ukážka formulára na vytvorenie nového hesla

#### 3.4.7.4 Prihlásenie

Pre prihlásenie sa do systému musíme najprv zvoliť typ účtu, do ktorého sa chceme prihlásiť, a potom zadať e-mail a heslo daného účtu.



Obrázok 76: Ukážka prihlásovacieho formulára stránky pre rôzne typy účtov

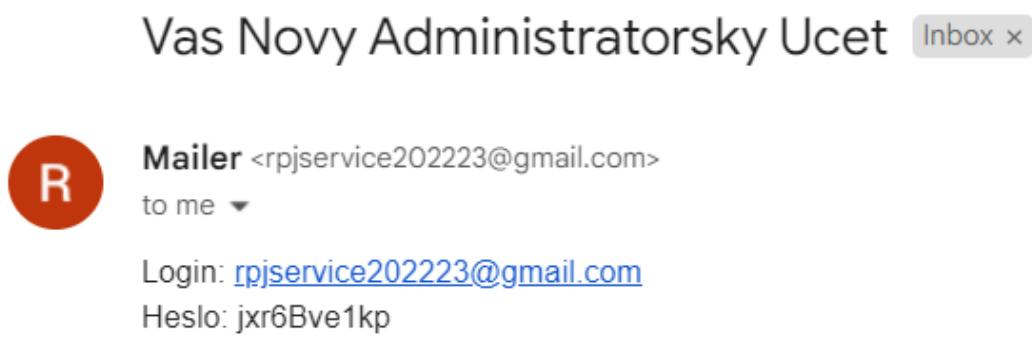
#### 3.4.7.5 Administrátorský panel systému

V administrátorskom paneli systému môže jeho administrátor pomocou formulára vložiť novú školu do systému. Po vyplnení formulára a kliknutí na tlačidlo „Pridať novú školu“ sa škola pridá do zoznamu a administrátorovi danej školy bude poslaná e-mail správa s prihlasovacími údajmi k jeho účtu.

**Administrátorovský panel**  
**Zoznam škôl**

<p>Nová škola</p> <p>Názov školy  <input type="text" value="Názov školy..."/></p> <p>Mesto  <input type="text" value="Mesto..."/></p> <p>Ulica  <input type="text" value="Ulica"/></p> <p>Číslo domu  <input type="text" value="Číslo domu..."/></p> <p>PSČ  <input type="text" value="PSČ..."/></p> <p>E-mail  <input type="text" value="E-mail..."/></p> <p>Telefónne číslo  <input type="text" value="Telefónne číslo..."/></p> <p>Meno školského administrátora  <input type="text" value="Meno školského administrátora..."/></p> <p>Priezvisko školského administrátora  <input type="text" value="Priezvisko školského administrátora..."/></p> <p>Titul školského administrátora (nepovinné)  <input type="text" value="Titul školského administrátora..."/></p> <p>E-mail školského administrátora  <input type="text" value="E-mail školského administrátora..."/></p>	<p>Administrácia</p>
<input type="button" value="Pridať novú školu"/>	

**Obrázok 78: Ukážka formulára pre pridanie školy v administrátorovskom paneli systému**



**Obrázok 77: Ukážka stránkou vygenerovaného e-mailu s prihlásovacími údajmi pre školského administrátora, po pridaní danej školy do systému**

Po kliknutí na tlačidlo „Administrácia“ sa nám zobrazí zoznam škôl, ktoré je možné spravovať, kliknútím na ikonku pera, alebo spravovať, kliknutím na ikonku X.

<u><b>Administrátorský panel</b></u> <u><b>Zoznam škôl</b></u>		
Nová škola	Administrácia	
Stredná priemyselná škola elektrotechnická v Prešove		
Testovacia škola		

**Obrázok 79:** Ukážka zoznamu škôl pre administráciu na administrátorskom paneli systému

Po vybratí si školy môžeme spravovať všetky tabuľky danej školy, pridať, mazat alebo meniť rôzne údaje školy.

<u><b>Testovacia škola</b></u>		
<u><b>Informácie o škole</b></u>		
<b>Názov</b>	Testovacia škola	
<b>Mesto</b>	Testovacie Mesto	
<b>Ulica</b>	Testovacia Ulica	
<b>Číslo domu</b>	777	
<b>PSČ</b>	09433	
<b>E-mail</b>	rpjservice202223@gmail.com	
<b>Tel. číslo</b>	777777	
<u><b>Odbory</b></u>		
<b>Názov</b>	Testovací odbor 1	
<b>Názov</b>	Testovací odbor 2	
<b>Názov</b>	Testovací odbor 3	
<b>Názov</b>	Testovací odbor 4	
<b>Názov</b>		

**Obrázok 80:** Ukážka tabuľiek spravovania konkrétnej školy v administrátorskom paneli systému

### 3.4.7.6 Administrátorový panel školy

V administrátorskom paneli školy môže školský administrátor spravovať všetky informácie danej školy, pridávať, mazať a aktualizovať informácie o danej škole, odboroch, ročníkoch, predmetoch, učiteľoch, triedach, skupinách a žiakoch. Tento panel je rozdelený na viacero sekcií, v ktorých môžeme nájsť zobrazené jednotlivé tabuľky, ktoré je možné podľa potreby upravovať.

The screenshot shows a web-based administrative interface for a school. At the top, there is a navigation bar with links: Škola, Odbory, Ročníky, Predmety, Učitelia, Triedy, Žiaci, and Odhlásiť sa. The main content area is titled "Informácie o škole". Below the title is a table containing the following data:

Názov	Testovacia škola	
Mesto	Prešov	
Ulica	Testovacia	
Číslo domu	7777	
PSČ	094337	
Počet učiteľov	66	
Počet študentov	799	
Počet odborov	4	
Počet predmetov	19	

Obrázok 81: Ukážka administrátorského panela konkrétnej školy po prihlásení sa ako administrátor danej školy

### 3.4.7.7 Učiteľský panel

Ak sa užívateľ prihlási do systému ako pomocou učiteľského konta, dostane sa do sekcie, kde môže vidieť všetky predmety, ktoré učí. Vedľa každého predmetu je modrá šípka, po kliknutí na ktorú sa prepne karta na jej podkartu (Predmet → Trieda → Skupina → Téma → Podtéma → Súbor). Využíva sa tu takzvaná stromová štruktúra. V danej štruktúre je možné pridávať a mazať témy, podtémy a súbory. Taktiež je možné ich zdielat' medzi jednotlivými triedami a ich skupinami.

The screenshot shows a user interface for managing subjects. At the top, there are navigation links: 'Predmety' (Subjects) and 'ToDoList'. On the right, it displays the name 'Ing. Jozef Mrkvička' and a link 'Odhlašiť sa' (Logout). The main title 'Predmety' is centered above a table. The table lists two subjects: 'Testovací\_predmet' and 'Matematika', each with a blue arrow icon to its right. The table has two rows and two columns.

1	Testovací_predmet	↗
2	Matematika	↗

Obrázok 82: Ukážka sekcie predmety daného učiteľa po prihlásení sa učiteľským kontom

### 3.4.7.8 Študentský panel

Ak sa užívateľ prihlási do systému ako pomocou študentského konta, dostane sa do sekcie, kde môže vidieť všetky predmety, ktoré sa učí. Po kliknutí na modrú šípku sa presunie o kartu nižšie (Predmety → Témy → Podtémy → Súbory). Taktiež, ako aj pri ostatných aplikáciách, užívateľ (študent tak ako aj administrátor) majú prístup k svojím osobnými ToDoList-om.

The screenshot shows a user interface for managing subjects. At the top, there are navigation links: 'Predmety' (Subjects) and 'ToDoList'. On the right, it displays the name 'John Doe, 1.TEST, 1.Skupina' and a link 'Odhlašiť sa' (Logout). The main title 'Predmety' is centered above a table. The table lists two subjects: 'Testovací\_predmet' and 'Matematika', each with a blue arrow icon to its right. The table has two rows and two columns.

1	Testovací_predmet	↗
2	Matematika	↗

Obrázok 83: Ukážka sekcie predmety daného študenta po prihlásení sa študentským kontom

## **4 Výsledky práce a diskusia**

Výsledkom nášho projektu je plne funkčná webová, desktopová a mobilná aplikácia. Všetky tri aplikácie medzi sebou komunikujú prostredníctvom zdieľanej databázy a spolu tvoria systém pre správu učebných materiálov školy. Zo začiatku bola práca na tomto projekte veľmi náročná, keďže sme si museli navrhnúť každú aplikáciu, jej triedy, jej dizajn a funkcionality, museli sme si premysliť ako budú aplikácie medzi sebou komunikovať a navrhnúť funkčnú a efektívnu databázu, ku ktorej budú mať aplikácie prístup, ďalej sme si museli premysliť, čo má daný systém robiť a túto myšlienku potom aplikovať do praxe.

Pri realizácii tohto projektu sme sa naučili, že je veľmi dôležité mať do detailov premyslené všetky prvky aplikácie, že je potrebné si na začiatku presne zadefinovať čo, ako, kedy a prečo budeme robiť a že komunikácia medzi všetkými zúčastnenými na projekte je veľmi podstatná a kľúčová. Taktiež sme sa naučili, že práca v tíme vie byť niekedy veľmi náročná na organizáciu ale na druhej strane stále platí známa veta „Viac hláv, viac rozumu.“

Sme radi, že sme sa pustili práve do tohto projektu, ktorý má ešte veľa možností na napredovanie a potenciál do budúcnosti.

## 5 Závery práce

Na záver by sme chceli zhodnotiť dosiahnuté výsledky počas práce na projekte a splnenie stanovených cieľov pri realizácii nášho systému. Po dlhom čase od začiatku práce na tomto projekte, tmavých aj svetlých momentoch, tvrdej práci a námahe sme finálne zavŕšili prácu na tomto projekte, ktorý ponúka školám trochu iné riešenie zdieľania učebných materiálov medzi učiteľmi a študentmi na troch platformách. Sústredili sme sa hlavne na funkciaľnosť a využiteľnosť ale nepozabudli sme ani na grafickú stránku a snažili sme sa zo seba dostať to najlepšie, čo sme sa za 4 roky na našej škole naučili.

Pri vytváraní tohto systému sme natrafili na množstvo zaujímavých a pre nás nových poznatkov, ako napríklad odosielanie e-mailov, javaScriptová funkciaľnosť, ktorá dokáže aktualizovať informácie z databázy na webovej stránke bez nutnosti refreshovania stránky alebo drag & drop systém v ToDoListe.

Podľa nášho názoru má táto aplikácia využitie v školskom prostredí ako celok. Ponúka ešte veľa možností na vylepšovanie a nadstavbu. Na záver by sme chceli ešte podotknúť, že sme hrdí na to, že sme dokázali vytvoriť hybridný projekt, splniť ciele, ktoré sme si zadali na začiatku a že sme aj takto prakticky mohli získať nové a nesmierne cenné informácie zo sveta IT a programovania.

Dúfame, že táto aplikácia si nájde obľubu medzi študentmi a učiteľmi a že im ukáže, že z myšlienky sa po určitom čase a námahe môže stať skutočnosť.

## Zhrnutie

Cieľom našej práce je vytvoriť prehľadný a intuitívny systém správy materiálov školy realizovaný hybridne, prostredníctvom troch databázou prepojených aplikácií. Hlavnou zásadou našej práce je vytvoriť systém správy materiálov, ktorý je efektívny a jednoduchý na používanie. Práca je rozdelená na teoretickú a praktickú časť, ktoré sa zaobrajú jednotlivými nástrojmi, jazykmi a prostrediami, ktoré používame na realizáciu jednotlivých častí práce, ale aj praktickými postupmi zahrnutými vo vytváraní webovej, desktopovej a mobilnej aplikácie. Praktické časti opísané v tejto práci poukazujú na to, že je možné dosiahnuť prepojenosť medzi rôznymi metódami vývoja, prostrediami a jazykmi, ale dôležitosť správneho návrhu základných stavebných blokov systému. Tento návrh a jeho realizácia umožňujú správu škôl, jej učiteľov a študentov, ako aj predmetov, tém a k nim prislúchajúcich materiálov, ktoré je možné zdieľať učiteľmi triedam a ich skupinám.

## **Resumé**

The goal of our work is to create an easy-to-use and intuitive system of managing school materials realized hybridly, through the use of three database-connected applications. The fundamental principle of our work is creating a system for managing materials which is effective and simple to use. The work is divided into theoretical and practical parts dealing with individual tools, languages and environments used in the realization of different parts of our work, but also with practical procedures used in creating the web, desktop and mobile applications. The practical parts described in this work pertain to the fact that it's possible to achieve interconnectedness between different methods of development, environments and languages, but also to the importance of correctly designing the basic building blocks of a system. This design and its realization allow for the management of schools, their teachers and students, as well as their subjects, topics and materials of said topics, which can be shared by the teachers with the classes and class groups.

## Zoznam použitej literatúry

**Almiray, Andres.** 2022. Icon packs for Java applications. *Kordamp.org*. [Online] 13 03 2022. [Cited: 21 03 2023.] Dostupné na: <[https://kordamp.org/ikonli/#\\_javafx](https://kordamp.org/ikonli/#_javafx)>.

**Android Studio developers.** 2023. Meet Android Studio. *developer android*. [Online] 03 05 2023. [Cited: 09 05 2023.] Dostupné na: <<https://developer.android.com/studio/intro>>.

**Bootstrap.** 2021. Forms. *getbootstrap.com*. [Online] 19 01 2021. [Cited: 06 12 2022.] Dostupné na: <<https://getbootstrap.com/docs/4.6/components/forms/#select-menu>>.

**Flores, William.** 2023. How to Change Row Color of Selected Rows in JavaFX TableView Using CSS. *opyprogramming.com*. [Online] 27 01 2023. [Cited: 20 04 2023.] Dostupné na: <<https://copyprogramming.com/howto/how-to-change-row-color-of-selected-rows-in-javafx-tableview-using-css>>.

**jQuery.com.** 2007. `.siblings()`. *api.jquery.com*. [Online] 30 08 2007. [Cited: 05 04 2023.] Dostupné na: <<https://api.jquery.com/siblings/>>.

—. 2007. `click` event. *api.jquery.com*. [Online] 02 09 2007. [Cited: 02 04 2023.] Dostupné na: <<https://api.jquery.com/click/>>.

—. 2007. `find()`. *api.jquery.com*. [Online] 30 08 2007. [Cited: 06 04 2023.] Dostupné na: <<https://api.jquery.com/find/>>.

**Mysql.com.** 2005. 12.15 Information Functions. *dev.mysql.com*. [Online] 13 10 2005. [Cited: 08 03 2023.] Dostupné na: <<https://dev.mysql.com/doc/refman/8.0/en/information-functions.html>>.

**Php.net.** 2009. `explode`. *Php.net*. [Online] 06 05 2009. [Cited: 19 01 2023.] Dostupné na: <<https://www.php.net/manual/en/function.explode.php>>.

—. 2006. `flush`. *Php.net*. [Online] 27 03 2006. [Cited: 04 04 2023.] Dostupné na: <<https://www.php.net/manual/en/function.flush.php>>.

—. 2006. `mysql_insert_id`. *Php.net*. [Online] 26 03 2006. [Cited: 08 03 2023.] Dostupné na: <<https://www.php.net/manual/en/function.mysql-insert-id.php>>.

—. 2010. `require`. *Php.net*. [Online] 05 06 2010. [Cited: 01 05 2023.] Dostupné na: <<https://www.php.net/manual/en/function.require.php>>.

**phpMyAdmin.** 2006. Bringing MySQL to the web. *phpMyAdmin*. [Online] 01 03 2006. [Cited: 09 05 2023.] Dostupné na: <<https://www.phpmyadmin.net>>.

**Redko, Alla.** 2013. Table View. *docs.oracle.com*. [Online] 15 09 2013. [Cited: 15 04 2023.] Dostupné na: <[https://docs.oracle.com/javafx/2/ui\\_controls/table-view.htm](https://docs.oracle.com/javafx/2/ui_controls/table-view.htm)>.

**Rouse, Margaret. 2020.** Mobile Application. *techopedia*. [Online] 07 08 2020. [Cited: 09 05 2023.] Dostupné na: <<https://www.techopedia.com/definition/2953/mobile-application-mobile-app>>.

**Wikipedia. 2023.** CSS. *Wikipedia the free encyclopedia*. [Online] 09 05 2023. [Cited: 09 05 2023.] Dostupné na: <<https://en.wikipedia.org/wiki/CSS>>.

—. 2023. Figma (software). *Wikipedia the free encyclopedia*. [Online] 16 04 2023. [Cited: 09 05 2023.] Dostupné na: <[https://en.wikipedia.org/wiki/Figma\\_\(software\)](https://en.wikipedia.org/wiki/Figma_(software))>.

—. 2021. FXML. *Wikipedia the free encyclopedia*. [Online] 25 01 2021. [Cited: 09 05 2023.] Dostupné na: <<https://en.wikipedia.org/wiki/FXML>>.

—. 2023. HTML. *Wikipedia the free encyclopedia*. [Online] 09 05 2023. [Cited: 09 05 2023.] Dostupné na: <<https://en.wikipedia.org/wiki/HTML>>.

—. 2023. IntelliJ IDEA. *Wikipedia the free encyclopedia*. [Online] 26 04 2023. [Cited: 09 05 2023.] Dostupné na: <[https://en.wikipedia.org/wiki/IntelliJ\\_IDEA](https://en.wikipedia.org/wiki/IntelliJ_IDEA)>.

—. 2023. Java (programovací jazyk). *Wikipédia Slobodná encyklopédia*. [Online] 19 04 2023. [Cited: 09 05 2023.] Dostupné na: <[https://sk.wikipedia.org/wiki/Java\\_\(programovac%C3%AD\\_jazyk\)](https://sk.wikipedia.org/wiki/Java_(programovac%C3%AD_jazyk))>.

—. 2023. JavaFX. *Wikipédie otevřená encyklopedie*. [Online] 07 04 2023. [Cited: 09 05 2023.] Dostupné na: <<https://cs.wikipedia.org/wiki/JavaFX>>.

—. 2023. JavaScript. *Wikipédia the free encyclopedia*. [Online] 29 03 2023. [Cited: 09 05 2023.] Dostupné na: <<https://en.wikipedia.org/wiki/JavaScript>>.

—. 2022. MySQL. *Wikipédia Slobodná encyklopédia*. [Online] 30 03 2022. [Cited: 09 05 2023.] Dostupné na: <<https://sk.wikipedia.org/wiki/MySQL>>.

—. 2023. PHP (skriptovací jazyk). *Wikipédia Slobodná encyklopédia*. [Online] 12 02 2023. [Cited: 09 05 2023.] Dostupné na: <[https://sk.wikipedia.org/wiki/PHP\\_\(skriptovac%C3%AD\\_jazyk\)](https://sk.wikipedia.org/wiki/PHP_(skriptovac%C3%AD_jazyk))>.

—. 2023. PhpStorm. *Wikipédie otevřená encyklopedie*. [Online] 17 03 2023. [Cited: 15 04 2023.] Dostupné na: <<https://cs.wikipedia.org/wiki/PhpStorm>>.

—. 2023. SQL. *Wikipedia the free encyclopedia*. [Online] 21 04 2023. [Cited: 2023 05 09.] Dostupné na: <<https://en.wikipedia.org/wiki/SQL>>.

—. 2021. Sublime Text. *Wikipédia Slobodná encyklopédia*. [Online] 06 10 2021. [Cited: 09 05 2023.] Dostupné na: <[https://sk.wikipedia.org/wiki/Sublime\\_Text](https://sk.wikipedia.org/wiki/Sublime_Text)>.

—. 2023. Visual Studio Code. *Wikipedia the free encyclopedia*. [Online] 08 05 2023. [Cited: 09 05 2023.] Dostupné na: <[https://en.wikipedia.org/wiki/Visual\\_Studio\\_Code](https://en.wikipedia.org/wiki/Visual_Studio_Code)>.

—. 2023. XAMPP. *Wikipédie otevřená encyklopedie*. [Online] 22 04 2023. [Cited: 09 05 2023.] Dostupné na: <<https://cs.wikipedia.org/wiki/XAMPP>>.

—. 2017. XML. *Wikipédia slobodná encyklopédia*. [Online] 13 05 2017. [Cited: 09 05 2023.] Dostupné na: <<https://sk.wikipedia.org/wiki/XML>>.

## Zoznam obrázkov

Obrázok 1: ERD diagram databázy .....	15
Obrázok 2: Zdrojový kód triedy „Main“ v desktopovej aplikácii .....	17
Obrázok 3: Triedy Modelu desktopovej aplikácie .....	18
Obrázok 4: Zdrojový kód triedy „Data“ v desktopovej aplikácii .....	19
Obrázok 5: Zdrojový kód triedy „DatabaseConnection“ v desktopovej aplikácii .....	20
Obrázok 6: Zdrojový kód triedy „Skola“ v desktopovej aplikácii .....	21
Obrázok 7: Prepojenie Controllera s .fxml súborom v desktopovej aplikácii.....	22
Obrázok 8: Prepojenie komponentov .fxml súboru s Controllerom v desktopovej aplikácii .....	22
Obrázok 9: Vytvorenie eventov s akciou, čo sa má stať po kliknutí na konkrétny komponent v desktopovej aplikácii .....	23
Obrázok 10: Prihlásovací panel v desktopovej aplikácii.....	23
Obrázok 11: Panel pre registráciu školy v desktopovej aplikácii.....	24
Obrázok 12: Panel pre odoslanie overovacieho kódu na e-mail v desktopovej aplikácii ...	24
Obrázok 13: E-mail s overovacím kódom.....	25
Obrázok 14: Zdrojový kód pre odosielanie e-mailu so zdrojovým kódom v desktopovej aplikácii .....	25
Obrázok 15: Zdrojový kód PHP súboru na serveri, ktorý zabezpečuje odosielanie e-mailov v desktopovej aplikácii .....	26
Obrázok 16: Panel pre zadanie overovacieho kódu v desktopovej aplikácii .....	26
Obrázok 17: Panel pre zmenu hesla v desktopovej aplikácii .....	27
Obrázok 18: Prihlásovanie pomocou prihlásovacieho panela v desktopovej aplikácii.....	27
Obrázok 19: Administrátorský panel (karta „Škola“) .....	28
Obrázok 20: Editácia buniek v tabuľke v desktopovej aplikácii .....	28
Obrázok 21: Administrátorský panel (karta „Odbory“) .....	29
Obrázok 22: Formulár pre pridávanie nových odborov v desktopovej aplikácii .....	29
Obrázok 23: Zobrazenie pridaného riadka v desktopovej aplikácii .....	30
Obrázok 24: Učiteľský panel (karta „Predmety“) v desktopovej aplikácii .....	30
Obrázok 25: Učiteľský panel (karta „Témy“) v desktopovej aplikácii .....	31
Obrázok 26: Formulár pre pridávanie nových tém v desktopovej aplikácii .....	31
Obrázok 27: Panel pre zdieľanie (čo chceme zdieľať) v desktopovej aplikácii.....	32
Obrázok 28: Panel pre zdieľanie (komu chceme zdieľať) v desktopovej aplikácii.....	32

Obrázok 29: Študentský panel (karta „Predmety“) v desktopovej aplikácii .....	33
Obrázok 30: Študentský panel (karta „Súbory“) v desktopovej aplikácii .....	33
Obrázok 31: Študentský panel (karta „ToDoList“) v desktopovej aplikácii .....	34
Obrázok 32: Súborová štruktúra v mobilnej aplikácii.....	35
Obrázok 33: Zdrojový kód triedy „LogInActivity“ (overenie zadaného e-mailu a hesla) v mobilnej aplikácii .....	36
Obrázok 34: Zdrojový kód triedy „ZabudnuteHesloActivity“ (overenie výskytu zadanej e-mailovej adresy v databáze) v mobilnej aplikácii .....	37
Obrázok 35: Zdrojový kód triedy „ZabudnuteHesloActivity“ (odoslanie e-mailu s overovacím kódom) v mobilnej aplikácii .....	37
Obrázok 36: Zdrojový kód triedy „OverovaciKodActivity“ (overenie overovacieho kódu zadaného užívateľom) v mobilnej aplikácii .....	38
Obrázok 37: Zdrojový kód triedy „ZmenaHeslaActivity“ (aktualizácia hesla) v mobilnej aplikácii .....	38
Obrázok 38: Zdrojový kód (metóda, ktorá zabezpečuje hashovanie hesla) v mobilnej aplikácii .....	39
Obrázok 39: Zdrojový kód triedy z Modiu v mobilnej aplikácii.....	40
Obrázok 40: Zdrojový kód triedy MenuActivity (akcie po kliknutí na jednotlivé komponenty) v mobilnej aplikácii.....	41
Obrázok 41: Zdrojový kód triedy „AddToDoListActivity“ (pridatie položiek) v mobilnej aplikácii .....	42
Obrázok 42: Zdrojový kód triedy „EditToDoListActivity“ (vymazávanie položiek) v mobilnej aplikácii .....	42
Obrázok 43: Zdrojový kód triedy „ToDoListActivity“ (vypisovanie položiek) v mobilnej aplikácii .....	43
Obrázok 44: Zdrojový kód triedy „ToDoListActivity“ (drag & drop systém) v mobilnej aplikácii .....	43
Obrázok 45: Zdrojový kód triedy zo sekcie Adapter (vykonanie akcie po kliknutí na položku v zozname) v mobilnej aplikácii.....	44
Obrázok 46: Zdrojový kód triedy zo sekcie Adapter (zobrazenie zoznamu položiek a nastavenie ich hodnôt) v mobilnej aplikácii .....	44
Obrázok 47: Prihlásovací panel v mobilnej aplikácii.....	45
Obrázok 48: Panel pre odoslanie overovacieho kódu na e-mail v mobilnej aplikácii .....	46
Obrázok 49: Zaslaný e-mail s overovacím kódom.....	46

Obrázok 50: Panel pre overenie hesla v mobilnej aplikácii .....	47
Obrázok 51: Panel pre zmenu hesla v mobilnej aplikácii .....	47
Obrázok 52: Prihlásovací panel v mobilnej aplikácii .....	48
Obrázok 53: Študentský panel (Súbory) v mobilnej aplikácii.....	49
Obrázok 54: Študentský panel (Podtémy) v mobilnej aplikácii .....	49
Obrázok 55: Študentský panel (Témy) v mobilnej aplikácii.....	49
Obrázok 56: Študentský panel (Predmety) v mobilnej aplikácii.....	49
Obrázok 57: Študentský panel – stiahnutie súboru v mobilnej aplikácii .....	50
Obrázok 58: Karta „Menu“ v mobilnej aplikácii.....	50
Obrázok 59: Katra „ToDoList“ v mobilnej aplikácii .....	51
Obrázok 60: Pridávanie novej úlohy do ToDoListu v mobilnej aplikácii.....	51
Obrázok 61: Editácia a vymazanie úlohy z ToDoListu v mobilnej aplikácii.....	52
Obrázok 62: Funkcia, ktorá vracia pripojenie k databáze v súbore ConnectToDB.php .....	53
Obrázok 63: Ukážka funkcií, ktoré vracajú hodnoty z databázy.....	54
Obrázok 64: Ukážka PHP kódu predstylizovaných výpisov do tabuľky .....	54
Obrázok 65: Ukážka využitia funkcie AJAX na dynamický výpis tém na stránke .....	55
Obrázok 66: Ukážka PHP kóde pre sťahovanie súborov zo servera.....	56
Obrázok 67: Ukážka PHP kódu pre nahrávanie súborov na server.....	56
Obrázok 68: Ukážka využitia triedy PHPMailer na poslanie používateľských údajov e-mailom po registrácii nového učiteľského účtu.....	57
Obrázok 69: Ukážka prihlásovacieho panela webstránky .....	58
Obrázok 70: Ukážka formulára pre registráciu školy.....	59
Obrázok 71: Ukážka stránkou vygenerovaného e-mailu pre registráciu školy administrátorovi systému.....	59
Obrázok 72: Ukážka formulára pre odoslanie e-mailu na obnovu hesla.....	60
Obrázok 73: Ukážka stránkou vygenerovaného e-mailu s kódom pre obnovu hesla.....	60
Obrázok 74: Ukážka formulára pre overenie kódu na obnovu hesla .....	60
Obrázok 75: Ukážka formulára na vytvorenie nového hesla .....	61
Obrázok 76: Ukážka prihlásovacieho formulára stránky pre rôzne typy účtov .....	61
Obrázok 77: Ukážka stránkou vygenerovaného e-mailu s prihlásovacími údajmi pre školského administrátora, po pridaní danej školy do systému .....	62
Obrázok 78: Ukážka formulára pre pridanie školy v administrátorskem paneli systému... 62	
Obrázok 79: Ukážka zoznamu škôl pre administráciu na administrátorskem paneli systému .....	63

Obrázok 80: Ukážka tabuľiek spravovania konkrétnej školy v administrátorskom paneli systému .....	63
Obrázok 81: Ukážka administrátorského panela konkrétnej školy po prihlásení sa ako administrátor danej školy .....	64
Obrázok 82: Ukážka sekcie predmety daného učiteľa po prihlásení sa učiteľským kontom .....	65
Obrázok 83: Ukážka sekcie predmety daného študenta po prihlásení sa študentským kontom .....	65