

PŘÍRODOVĚDECKÁ FAKULTA UNIVERZITY PALACKÉHO
KATEDRA INFORMATIKY

BAKALÁŘSKÁ PRÁCE

Výukový program pro základní školy



Anotace

Práce se zabývá vytvořením počítačového programu pro výuku žáků na základních školách. Program je rozdělen do dvou částí. Serverová část slouží pro přístup k databázi, klientská pak jako aktivní uživatelské rozhraní pro žáka nebo vyučujícího. Výsledkem práce je pak serverový program Server a klientský program Školička. Dále se práce věnuje použitým technologiím, návrhu a implementaci programu pro operační systém Microsoft Windows na platformě .NET Framework 4.0.

Tímto bych chtěl poděkovat především mému vedoucímu práce, RNDr. Arnoštovi Večerkovi, který mi byl kdykoliv nápomocen při tvorbě a následném testování. Dále pak mé přítelkyni a rodině, kteří mi byli vždy oporou a také skupině Jethro Tull, která mě svými tóny provázela po celou dobu tvorby této práce.

Obsah

1. Úvod	7
2. Příprava práce	8
2.1. Současná situace	8
2.2. Řešení problému	8
2.3. Výběr technologií	9
2.3.1. .NET Framework	9
2.3.2. C#	10
2.3.3. LINQ	10
3. Cíle	11
3.1. Jednoduchost	11
3.2. Srozumitelnost	11
3.3. Věrohodnost	11
4. Architektura	12
4.1. Zvolená implementace	12
4.1.1. Server	12
4.1.2. Klient	13
5. Programátorská dokumentace	14
5.1. Klient	14
5.1.1. Adresářová struktura	14
5.1.2. Popis tříd	15
5.2. Server	17
5.2.1. Popis tříd	17
6. Uživatelská dokumentace	19
6.1. Požadavky	19
6.2. Instalace databáze	19
6.3. Přihlášení	19
6.4. Změna hesla	21
6.5. Vyučující	23
6.5.1. Hlavní okno	23
6.5.2. Vytváření testu	25
6.5.3. Vytváření úkolu	27
6.5.4. Žáci testu a úkolu	28
6.5.5. Změna údajů	29
6.5.6. Přidání žáků	29
6.5.7. Přidání vyučujícího	30
6.5.8. Odebrání žáka	31
6.5.9. Odebrání vyučujícího	31

6.5.10. Změna třídy žáka	32
6.5.11. Změna hesla žáka	32
6.5.12. Nový školní rok	33
6.5.13. Seznam žáků a vyučujících	34
6.6. Žák	35
6.6.1. Hlavní okno	35
7. Reálný provoz	39
7.1. Testovací subjekty	39
7.2. Připomínky	40
8. Plány do budoucna	41
Závěr	42
Conclusions	43
Reference	44
A. Obsah přiloženého CD	45
B. Hodnocení uživatele	46

Seznam obrázků

1.	Graf podílu OS na českém trhu ve 4.čtvrtletí roku 2013	10
2.	Schéma použité architektury	13
3.	ER diagram databáze	14
4.	Okno přihlášení	20
5.	Diagram případů užití	21
6.	Okno změny hesla	21
7.	Hlavní okno vyučujícího	23
8.	Okno přidávání otázky	26
9.	Okno s vytvářením úkolu	27
10.	Okno s přidáním žáka	29
11.	Okno s přidáním vyučujícího	30
12.	Okno s odebráním žáka	31
13.	Okno se změnou třídy žáka	32
14.	Okno s posunem třídy do dalšího roku	33
15.	Okno se seznamem žáků	34
16.	Hlavní okno žáka	36
17.	Okno s odevzdáním úkolu	36
18.	Okno s vyplněním testu	39

1. Úvod

Svět jde dopředu obrovskou rychlostí, a tak žáci na základních školách jsou dnes již od prvních ročníků ve styku s moderními technologiemi. Velkou mírou tomu napomáhá i stát zakotvováním předmětů týkajících se počítačů do osnov základních škol. V těchto případech však vyučující z velké míry nejsou schopni znalostí žáků nikterak v oblasti vzdělávání využít. Žáci tak následně své teoretické znalosti mohou zúročit pouze v domácím prostředí. Hlavním důvodem vzniku této práce tedy je vyřešit nebo alespoň redukovat tento problém.

V rámci bakalářské práce se snažím vymyslet co nejjednodušší program, který by byl schopen ovládat i velmi mladý uživatel bez větších znalostí čtení. Bude se jednat o program, který by měl sloužit jak vyučujícím, tak i žákům. Měl by ulehčit jejich výuku a začlenit do ní technologie, které byly doposud opomíjeny. Základní forma, kterou by vyučující měl mít možnost žáky vzdělávat, je vytváření testů či úkolů. Na tyto testy či úkoly pak žáci budou moci provést zpětnou vazbu, díky níž vyučující pozná, zda je žák v této konkrétní situaci dostatečně vzdělán či ne. K sebezdokonalování může vést i fakt, že bude žákovi umožněno po vyplnění testu zobrazit správné odpovědi.

Program se dělí na dvě části – klient a server. Serverová část slouží jako prostředník ke komunikaci klientské části s databází obsahující veškerá data potřebná k fungování programu na konkrétní škole. Další její funkcí je zajišťování úložného prostoru pro soubory přiložené k jednotlivým zadáním a řešením. V serverové části programu tedy nehraje žádnou roli uživatel, neboť jejím jediným přímým výstupem jsou informace přijímané od klientů nebo klientům odesílané. Klientskou část programu pak má již uživatel plně ve svých rukou. Program zde má dva základní možné okruhy použitelnosti v závislosti na faktu, zda se uživatel přihlásí pod právy vyučujícího nebo žáka. V prvním případě je uživateli umožněno vytvářet testy nebo úkoly. Testy se skládají z neomezeného počtu otázek, kde každá z nich může obsahovat obrázkové zadání či odpovědi. Úkoly pak umožňují vyučujícímu k slovnímu zadání připojit také jakýkoli soubor. Mezi další možnosti vyučujícího patří spravování veškerých uživatelů, tedy vyučujících nebo žáků, či správa jednotlivých tříd na jeho škole. Mezi hlavní funkce žáka, po jeho přihlášení, patří především vyplnění zadaného testu nebo vyplnění odpovědi na úkol. K odpovědi může žák také přiložit jakýkoli soubor. Dále pak může žák provádět různá základní nastavení programu a také si změnit své přihlašovací heslo. Jednotlivé funkce programu jsou pro snadnější orientaci opatřeny hlasovou nápovědou.

2. Příprava práce

2.1. Současná situace

Před zahájením samotné práce jsem se nejprve potřeboval seznámit s problematikou mého tématu. Snažil jsem se tedy zjistit co nejvíce informací o aktuálním dění na školách a také o tom, zda nějaký program funkčně podobný mému vůbec existuje.

Pokud by člověk hledal v oddělení soukromého vývoje, existuje velké množství programů určených k výuce žáků na základních školách. Většinou se však jedná pouze o „desktopové“ aplikace, které nijak nekomunikují s vyučujícím. Zde je tedy problém v nulové interakci s vyučujícím a často také s malým oborovým rozsahem konkrétního programu.

V dobách dokončování vývoje mého programu bylo oznámeno budoucí plošné nasazení programu funkčně podobného mému do všech státních základních škol. Již v prvních dnech, kdy byly proškoleni ředitelé základních škol, nebyl tento projekt příliš pozitivně přivítán. Už samotná podstata, že k jeho ovládnutí je vyučující třeba proškolovat, napovídá o náročnosti a použitelnosti programu. Potvrdila se tak má teorie, že další státem vypsany tendr kvalitně neodpovídá svému účelu.

2.2. Řešení problému

Současná situace mě utvrdila v tom, že myšlenka, s kterou aplikaci vyvíjím, je správná a mohla by se tak dočkat i vřelého přijetí z řad zaměstnanců škol. Program bude přínosem zejména v přímé práci vyučujícího s každým jednotlivým žákem. Vyučující bude mít jasný přehled o aktuálních znalostech konkrétního žáka a bude moci na tyto znalosti adekvátně reagovat.

Dalším důležitým přínosem oproti jiným existujícím řešením bude neomezený výběr tématu kterého se tvorba testu či úkolu bude týkat. Výukový materiál tak bude „ušit“ přímo na míru potřebám vyučujících a žáků. Pro vyvarování se problému se složitostí, jsem se rozhodl již v počátcích veškerou mou tvorbu konzultovat s koncovými uživateli. Zjišťoval jsem především, jaké jsou informatické znalosti vyučujících a žáků. U žáků je již v prvních ročnících velmi dobrá úroveň práce s počítačem, jediné s čím je potřeba se vypořádat, je mírná negramotnost. Proto jsem se rozhodl do mého programu zakomponovat hlasovou nápovědu, která tento problém z větší míry odstraní. Co se týče zkušeností vyučujících, nastává u nás momentálně generační přelom, kdy se na základních školách stále nacházejí vyučující, kteří mají k počítačům spíše odpor a pracují s ním pouze z donucení. Proto se často táží mladších kolegů a kolegyně o pomoc. Mé řešení problému vidím v jednoduchosti programu a v obsáhlosti jeho nápovědy. Jistou mírou se na tom podílí i mnou implementovaná možnost sdílení vytvořených materiálů s jiným vyučujícím.

Vyvinul jsem tak program, který by měl být velmi konkurenceschopný, neboť již svoji architekturou klient-server přináší něco nového, u nás nedosažitelného.

2.3. Výběr technologií

Při výběru jazyka bylo pro mne důležitých několik faktorů. Nepočítal jsem mezi ně rychlost, neboť díky jednoduchosti by aplikace neměla být hardwarově náročná. Kvůli rozšířenosti platformy Windows u nás (obr. 1.), jsem si nemusel dělat příliš velkou starost ohledně spouštění programu na jiných platformách než z rodiny Windows. Z možných jazyků odpovídajících mým zkušenostem a požadavkům jsem si pro tvorbu vybral programovací jazyk C#. C# je jednoduchý a výkonný jazyk, který se zaměřuje především na vývojáře aplikací v prostředí .NET Framework. Tato kapitola částečně vychází z publikací [1] a [2].

2.3.1. .NET Framework

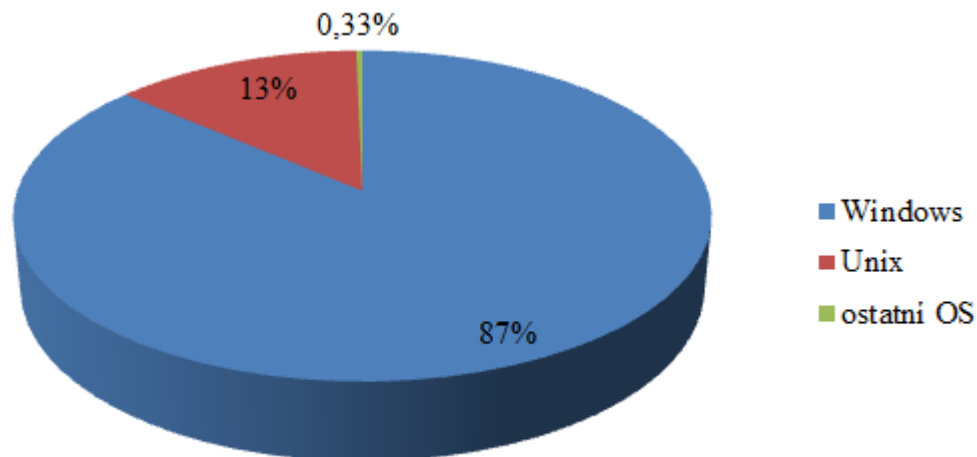
Platforma .NET byla oficiálně představena firmou Microsoft v roce 2000. Znamená novou generaci vývoje aplikací pro operační systém Windows. Hlavním důvodem vzniku této platformy byly:

- nekompatibilita jednotlivých jazyků;
- vysoká chybovost vytvořených aplikací;
- problém s verzemi knihoven;
- zastaralý a nepřehledný způsob vývoje webových aplikací.

Všechny tyto problémy řeší platforma .NET především použitím takzvaného řízeného běhu prostředí. Řízení běhu prostředí znamená, že program při převodu zdrojového kódu do strojového využívá ještě jedné mezivrstvy. Z tohoto mezikódu je pak program převeden do strojového kódu aktuální platformy až při spuštění programu. Tento proces je výkonově náročnější, proto ho není vhodné používat u některých složitějších aplikací, například u počítačových her. Tento mezikód se u platformy .NET nazývá CIL¹ a je spouštěn důležitou součástí .NET framework, nazvanou CLR². Aby platforma splňovala bod ohledně kompatibility jazyků, obsahuje také součást CTS, což je sada datových typů pro programovací jazyky a CLS určující pravidla, která musí každý jazyk využívající tuto platformu splňovat. Pro následný vývoj aplikace na platformě .NET pak lze použít kromě jazyka C# také jazyky jiné, například Visual Basic .NET nebo J#.

¹CIL – Common Intermediate Language - jazyk definovaný specifikací Common Language Infrastructure používaný projekty .NET Framework

²CLR – Common Language Runtime běhové prostředí převádějící kód do strojového kódu konkrétního počítače



Obrázek 1. Graf podílu OS na českém trhu ve 4.čtvrtletí roku 2013

2.3.2. C#

Samotný jazyk C# je vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft. Hlavní funkčnost je založena na jazycích C++ a Java. První zmínky o tomto jazyce přišli společně s Visual Studiem .NET v roce 2000. Jazyk lze, kromě formulářové a databázové aplikace jako v mém případě, využít také například pro tvorbu webových či mobilních aplikací. Dalším důležitým aspektem jazyka je automatická správa paměti. V rámci programu tak funguje takzvaný garbage collector, který se stará o uvolňování zdrojů spuštěného programu.

2.3.3. LINQ

Ve verzi 3.0 jazyk přichází s podporou LINQ³, tedy vlastního dotazovacího jazyka. Tato funkce zavádí jednotné dotazování nad objekty, databázemi, či XML soubory. Dále jsou dostupné i další uživatelsky vytvořené variace použití LINQ, například LINQ to Amazon. Technologie využívá několik stěžejních novinek jazyka, například se jedná o lambda výrazy nebo výrazové stromy.

³Language Integrated Query

3. Cíle

3.1. Jednoduchost

Vzhledem k důvodům, které mě dovedli k vývoji programu, jsem jako hlavní cíl mé tvorby určil jednoduchost. Každá věc ve vytvořeném programu by měla mít své místo a její pozice by měla odpovídat tomu, jak často je využívána. V žádném kroku práce s programem by uživatel neměl být vystaven přílišnému počtu ovládacích prvků, které by mohli vést k jeho dezorientaci.

3.2. Srozumitelnost

Uživatel je zde nejdůležitější článek vývoje programu. Pokud je tedy ztracen, ztrácí program funkčnost, a tím postrádá veškerý smysl. Uživatel by měl v každém kroku práce s programem vědět, co zrovna dělá a k čemu jaký ovládací prvek slouží. Pokud je to jeho první interakce s programem a není si jist, měl by na to s pomocí nápovědy jednoduše a rychle přijít.

Určení mého programu zároveň nastiňuje problém s věkem uživatele. Ten by tedy měl být schopen ovládat program s minimálními zkušenostmi. Mým cílem je tedy co největší možná eliminace tohoto problému, neboť jeho úplné odstranění zřejmě není v mých silách.

3.3. Věrohodnost

Uživateli by měla být zobrazena vždy ta data, která jsou aktuálně uložena v databázi. Měl by tak mít vždy aktuální data a neměl by mít přístup k datům, která již v databázi neexistují.

4. Architektura

Při přihlédnutí k faktu, že můj program musí fungovat mezi více osobními počítači, zde musí existovat komunikace mezi těmito zařízeními. Síťová architektura nám říká, jakým stylem zařízení komunikují a jaké technologie využívají.

Architektura klient-server funguje na bázi jednoho hlavního uzlu nazývaného jako server a více vedlejších uzlů, klientů (obr. 2.). Klient může serveru jak data zasílat, tak od něj data přijímat. Architektura je vhodná, pokud uživatel chce svá data poskytnout více uživatelům. Mezi negativa může patřit potřeba dalšího uzlu pro roli serveru nebo netransparentnost výměny dat. Mezi příklady využívající této architektury patří HTTP server. Na bázi klient a server existuje, kromě této základní, také několik dalších variací této architektury. Každá má své pro a proti vzhledem k účelům vytvářeného programu:

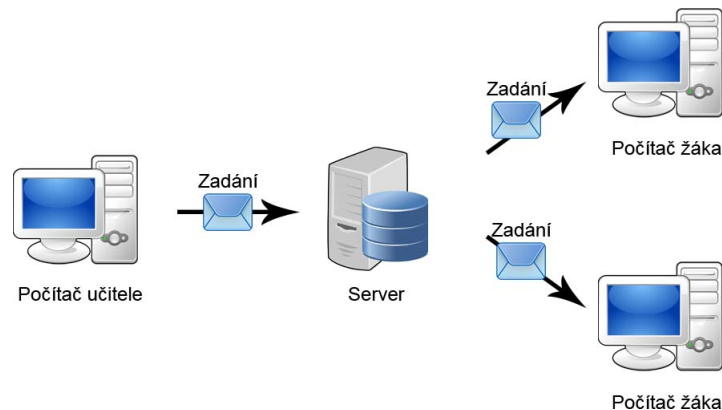
- **Klient-klient:** Základním kamenem této architektury často zvané také peer-to-peer je klient. Ten ke své komunikaci s jiným klientem nepotřebuje žádný další subjekt a jeho data proudí skrz síť přímo k dalšímu klientovi. V tu chvíli tedy počítač slouží jako server a má možnost v případě potřeby data i přijímat a chovat se jako klient. Všechny uzly sítě jsou si tedy rovnocenné. Pro tuto architekturu hraje fakta jako rychlost přenosu dat či bezpečnost, neboť uživatelská data nejsou v průběhu přenosu nikde ukládána. Jako příklad takovéto sítě může sloužit Napster, která v dávných dobách sloužila k výměně dat, především audio nahrávek mezi uživateli.
- **Klient-fronta-klient:** Tato architektura je často nazývána také jako pasivní fronta. Server je v ní omezen pouze na roli fronty, do které mají přístup všichni klienti. Ti mohou z této fronty číst i do ní zapisovat. Jako příklad může sloužit web crawler, tedy systém pro indexování obsahu webů.

4.1. Zvolená implementace

Mnou zvolená architektura je základní varianta klient-server, neboť je v mém případě zřejmě nejvýhodnější. Učitel tak má možnost zvolit větší počet žáků, kterým chce vytvářený test či úkol poskytnout. V následujících bodech jednotlivě popíši, k jakému účelu slouží jednotlivé složky architektury.

4.1.1. Server

Server v mé aplikaci hraje zásadní roli, neboť komunikuje se všemi klienty a reaguje na přijímané zprávy. Klienti tak bez spuštěného serveru a fungujícího přístupu k němu nejsou schopni svůj program ani spustit. Předpokládá se, že server bude spuštěn na jiné než lokální síti, a tudíž bude server s klientem komunikovat přes síť Internet. Základní funkce serveru tak jsou komunikace s databází



Obrázek 2. Schéma použité architektury

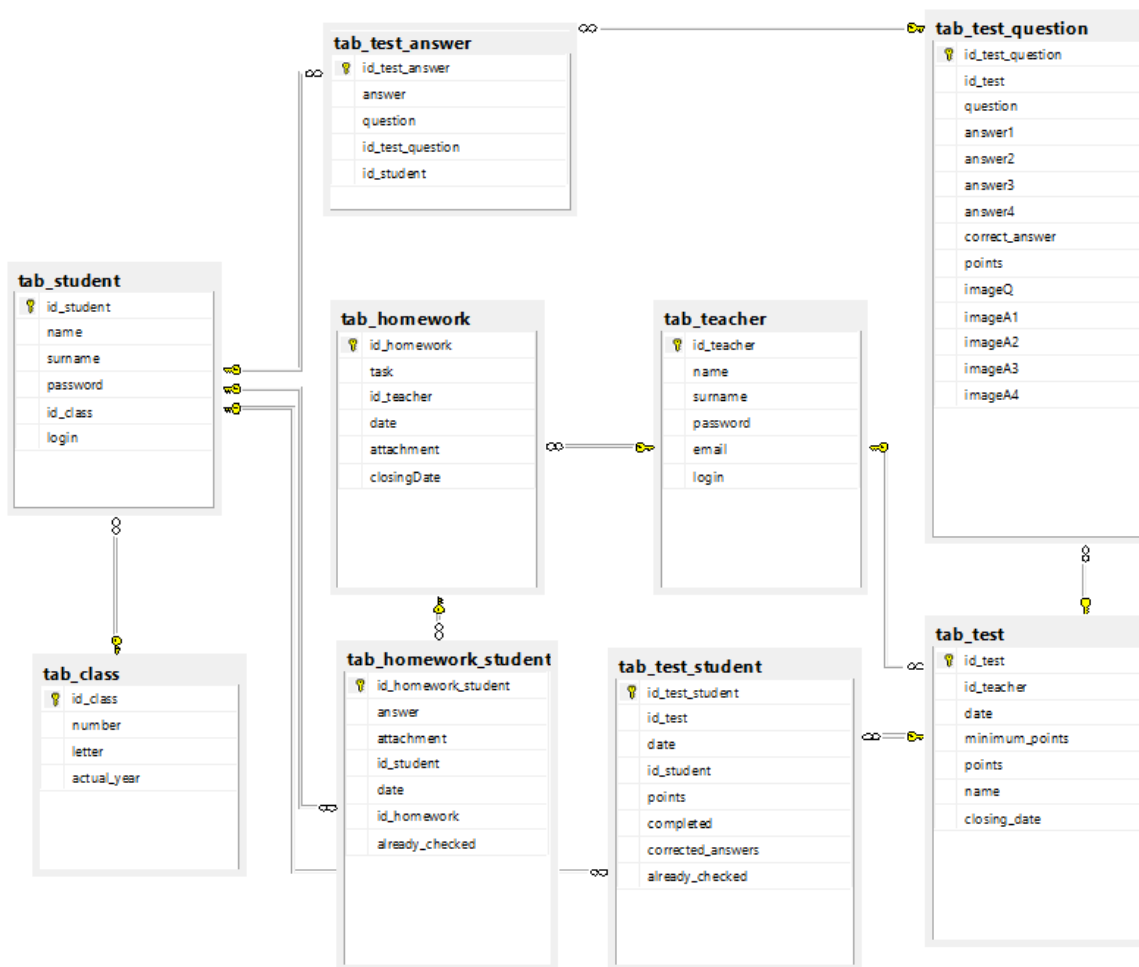
a zprostředkování databáze klientům pomocí komunikace s nimi.

Na počátku komunikace s jednotlivými klienty je provedeno navázání spojení. Pokud toto spojení proběhlo úspěšně, server ho až do jeho ukončení bude značit pod unikátním číslem a spravovat ho v samostatném vlákně. Mezi funkce klienta pak patří také automatická kontrola nových úkolů a testů pro přihlášeného žáka. Kvůli možnému ovlivnění a zatěžování zbylé komunikace je tato kontrola prováděna v samostatně připojeném vlákně. Server po připojení klienta mu neustále naslouchá a je připraven na každou jeho zprávu reagovat.

Většina odpovědí na klientův dotaz vyžaduje spolupráci serveru a databáze. Databáze v mém projektu je založena na relačním modelu (obr. 3.) a server s ní komunikuje pomocí jazyka SQL. Databáze obsahuje devět tabulek týkajících se testů, úkolů a uživatelů přistupujících k programu. Každá tabulka má jeden primární klíč a většina tabulek i klíče cizí. Mezi těmito tabulkami existují relace typu 1:N, přiřazují jednomu záznamu více záznamů z jiné tabulky.

4.1.2. Klient

Klient zde, na rozdíl od serverové části, má zcela jinou úlohu, neboť převážně slouží (pomocí grafického uživatelského rozhraní) jako nástroj k interakci s uživatelem. Jedná se tak o velmi důležitou část projektu, bez níž by tvorba neměla smysl. Klient má v konfiguračním souboru nastavenou adresu serveru, ví tedy kde má potřebný server hledat. V případě, že je server nedostupný, klient informuje o této skutečnosti uživatele a následně se ukončí. Naráz lze přistupovat z jednoho klienta pouze k jednomu serveru. Klientský program tak zasláním krátkých dotazovacích zpráv na server zastupuje práci s databází jak pro vyučujícího, tak pro žáky.



Obrázek 3. ER diagram databáze

5. Programátorská dokumentace

5.1. Klient

5.1.1. Adresářová struktura

Adresářová struktura projektu se skládá z těchto položek:

- **graphic/** Adresář obsahující veškeré obrázky týkající se grafického prostředí klientského programu.
- **sounds/** Adresář obsahující zvuky používané v programu.
- **Školička.exe.config** Soubor obsahující základní konfiguraci programu. Jedná se o zjištění údajů potřebných k přihlášení se k serveru, informaci zda

má být spuštěna uvítací hláška a v neposlední řadě také uložené přihlašovací údaje.

5.1.2. Popis tříd

Jednotlivé třídy klientského programu lze rozdělit do několika kategorií.

1. **Společné třídy** Společné třídy jsou takové, s nimiž je umožněno pracovat každému uživateli programu. Jedná se o tyto třídy:

- *Communication* je třída, jenž zprostředkovává veškerou komunikaci mezi klientem a serverem. Vytváří spojení a umožňuje zasílat a přijímat textové zprávy a soubory.
- *Program* je třída, spouštějící se při startu programu. Při této příležitosti naváže spojení se serverem vytvořením instance třídy *Communication*. Zkoumá také, zda je na nastavené adrese dostupný server. Pokud tomu tak není, je o této skutečnosti uživatel informován a program je ukončen. Třída obsahuje také funkce společné pro celý program.
- *User* je třída zastupující přihlášeného uživatele. Instance této třídy je vytvářena při úspěšném přihlášení uživatele do systému. Uvnitř třídy se pak uchovávají některé identifikační údaje uživatele získané z databáze. Na základě těchto údajů je také určeno, jaké má uživatel postavení v programu a tím také do kterých formulářů bude mít přístup.
- *FormFullImage* je třída vytvářející formulář, jenž zobrazuje plnou velikost obrázku. Velikost otevřeného okna se přizpůsobuje velikosti otvíraného obrázku.
- *FormHWAnswer* je třída vytvářející formulář, jenž zobrazí odpověď žáka na zvolený domácí úkol.
- *FormChangePassword* je třída vytvářející formulář, pomocí něhož uživatel může změnit své přihlašovací heslo. Třída obsahuje dva konstruktory, druhý je volán pouze v případě, že uživatel heslo mění při prvním přihlášení a předá se tak i uživatelovo aktuální heslo.
- *FormLogin* je třída vytvářející formulář, jenž umožňuje uživateli se přihlásit do programu.
- *FormTestFinished* je třída vytvářející formulář, v němž uživatel může zobrazit zvolený test a odpovědi na jeho otázky od zvoleného žáka.

2. **Učitel** Kategorie, která obsahuje všechny třídy, jejichž funkčnost využívá pouze vyučující. Ve všech případech se jedná o třídy tvořící formuláře vytvořené v jazyce C# pomocí technologie Windows Forms. Součástí tak jsou automaticky generované částečné třídy s přídomkem *designer*, obsahující informace o ovládacích prvcích formulářů. Jedná se o tyto třídy:

- *FormTeacherAddClass* je formulář, sloužící k přidání nové třídy do databáze systému.
- *FormTeacherAddHW* je formulář, sloužící k zasílání zprávy serveru s požadavkem o přidání nového úkolu.
- *FormTeacherAddHWStudents* je formulář, sloužící k výběru studentů, pro které je určen vytvářený úkol. Třída obsahuje dva konstruktory, neboť může být volána ve dvou různých situacích. V prvním případě při vytváření úkolu, v druhém při úpravě již vytvořeného s tím, že jsou předáni všichni žáci kterým je již úkol zadán.
- *FormTeacherAddQuestions* je formulář, sloužící k přidávání nové otázky do právě vytvářeného testu.
- *FormTeacherAddStudent* je formulář, sloužící k přidání nového žáka do databáze školy. Obsahuje dva konstruktory. První z nich je volán při vytváření žáka přes položku kontextového menu, druhý při vytváření první třídy ve funkci Nový školní rok.
- *FormTeacherAddTeacher* je formulář, sloužící k přidávání nových vyučujících do databáze systému.
- *FormTeacherAddTest* je formulář, sloužící k prvnímu kroku vytváření testu. U testu se v něm vyplní název, termín odevzdání a také údaj, zda jednotlivé otázky mají být různě bodované. Potřebné údaje se pak pošlou na server s dotazem o vytvoření testu v databázi.
- *FormTeacherAddTestStudents* je formulář, sloužící k vybrání těch žáků, kterým je vytvářený test určen. Obsahuje dva různé konstruktory, kde druhý je volán v případě, že test již má nějaké žáky přidání.
- *FormTeacherDeleteStudent* je formulář, sloužící ke smazání vybraného žáka.
- *FormTeacherDeleteTeacher* je formulář, sloužící ke smazání vybraného vyučujícího z databáze školy.
- *FormTeacherEdit* je formulář, sloužící k úpravě osobních údajů vyučujícího.
- *FormTeacherHWStudents* je formulář, sloužící k zobrazení všech žáků, kterým byl úkol zadán. Zobrazí také jejich průběžný stav odevzdání.
- *FormTeacherChangeStudentClass* je formulář, sloužící k přesunu žáků mezi třídami. Není využíván při postupu do dalšího ročníku.

- *FormTeacherChangeStudentPassword* je formulář, sloužící k možnosti změny hesla žáka vyučujícím.
 - *FormTeacherChangeTerm* je formulář, sloužící ke změně termínu odevzdání úkolu či testu.
 - *FormTeacherListTeachers* je formulář, sloužící k výpisu všech vyučujících v databázi a případné extrahování seznamu do aplikace Microsoft Office Excel.
 - *FormTeacherListStudents* je formulář, sloužící k výpisu všech žáků nacházejících se aktuálně v databázi.
 - *FormTeacherMain* je formulář, zobrazující hlavní nabídku učitele.
 - *FormTeacherNextYear* je formulář, sloužící k přesunutí všech žáků do dalšího ročníku.
 - *FormTeacherTestStudents* je formulář, zobrazující všechny žáky, kterým je daný test určen, s jejich průběžným stavem vyplnění.
3. **Žák** Kategorie, jenž obsahuje všechny třídy, které může používat výhradně žák. Veškeré třídy obsahují vytvoření formuláře Windows Forms. Součástími těchto tříd jsou tedy také automaticky generované částečné třídy, jejichž název obsahuje klíčové slovo *designer*. Tyto částečné třídy popisují ovládací prvky na formulářích. Jedná se o tyto třídy:
- *FormStudentHomeworkFill* je formulář, sloužící k zobrazení zadaného úkolu s možností na něj odpovědět.
 - *FormStudentMain* je formulář, jenž slouží jako hlavní okno programu pro přihlášeného žáka.
 - *FormStudentTestFill* je formulář, v němž je žákovi umožněno postupně vyplnit otázky vybraného testu.

5.2. Server

5.2.1. Popis tříd

V této sekci popíší jednotlivé třídy serverové části programu. Jedná se tyto třídy:

- *DataClasses* je třída, komunikující přímo s SQL databází. Vytváří třídu pro každou tabulku obsaženou v databázi a obsahuje základní metody pro práci s databází.
- *Communication* je třída, obsluhující komunikaci s klienty. V první řadě vytváří spojení pak také přijímá a odesílá zprávy jednotlivým připojeným klientům.

- *Database* je třída, která slouží ke komunikaci s databází. Využívá metody třídy *DataClasses* a technologie LINQ to SQL⁴.
- *Message* je třída, která řeší porozumění příchozí zprávě od klienta a následně vytváří na tuto zprávu odpověď¹.
- *Program* je třída, jejíž instance je vytvářena při spuštění programu. Vytváří objekt třídy *Communication*. Kontroluje taky, zda je dostupná databáze na nastavené adrese. Pokud tomu tak není, je o tom uživatel informován a program následně vypnut.

⁴Language Integrated Query to Structured Query Language – viz kapitola Příprava práce

6. Uživatelská dokumentace

V této kapitole popíšeme práci s programem. Vzhledem k tomu, že serverová část programu nepodporuje žádnou interakci s uživatelem, bude u ní popsána pouze instalace SQL databáze. Ve zbytku kapitoly tak bude popsána klientská část. Dokumentace je vcelku detailní, především kvůli možné nízké počítačové gramotnosti koncového uživatele.

6.1. Požadavky

Pro spuštění programu je zapotřebí operační systém Windows, ve verzi XP SP3 a novější. Další důležitou podmínkou je nainstalování knihovny Microsoft .NET Framework verze 4 a vyšší.

6.2. Instalace databáze

Pro správný chod aplikace je zapotřebí mít nainstalovaný SQL server a mít k dispozici program, který ho dokáže spravovat. Svůj příklad uvedu na programu Microsoft SQL Server Management Studio 2012, který je volně ke stažení ze stránek výrobce. Postup instalace vychází z publikace [3]. Po nainstalování těchto aplikací se musí vytvářený server pojmenovat „SQL“. V případě jiného názvu, nebo v případě, že se SQL server nenachází na stejném počítači jako serverová aplikace, je potřeba změnit konfigurační soubor `server.exe.config`. Jméno SQL serveru je k nalezení v části zvané *connectionStrings*, kde v hodnotě *add* a atributu *connectionString* je uvedena cesta k databázi. Uvádí se ve tvaru *adresa serveru s databází \název databáze* kde adresa serveru může být, v případě nainstalování serverové aplikace na stejném serveru, ve tvaru „localhost“.

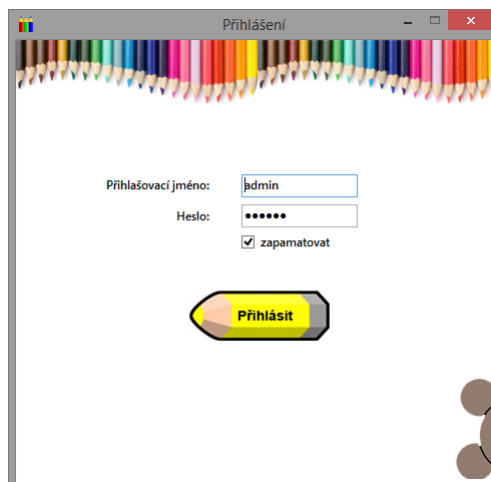
Samotné připojení databáze programu se provádí v aplikaci SQL Server Management Studio. Po jejím spuštění a připojení se k serveru je zapotřebí kliknout pravým tlačítkem myši na složku *Databases* a vybrat položku *Attach*. V otevřeném okně pak kliknout na tlačítko *Add* a vybrat lokaci, kde se soubor databáze programu nachází. Ta je implicitně v adresáři programu, v podsložce *data*.

Po instalaci databáze a spuštění serveru je zapotřebí vytvořit administrátorský účet. K tomu stačí spustit klientskou část programu a následně vyplnit heslo administrátora. Po úspěšném vytvoření již nic nebrání v použití programu.

6.3. Přihlášení

Základní obrazovkou po spuštění programu je obrazovka s přihlášením do programu (obr. 4.). Pokud není dostupný server, je zobrazeno upozornění a poté je program ukončen. V případě delšího trvání by měl uživatel kontaktovat svého

vyučujícího, případně správce sítě na své škole.

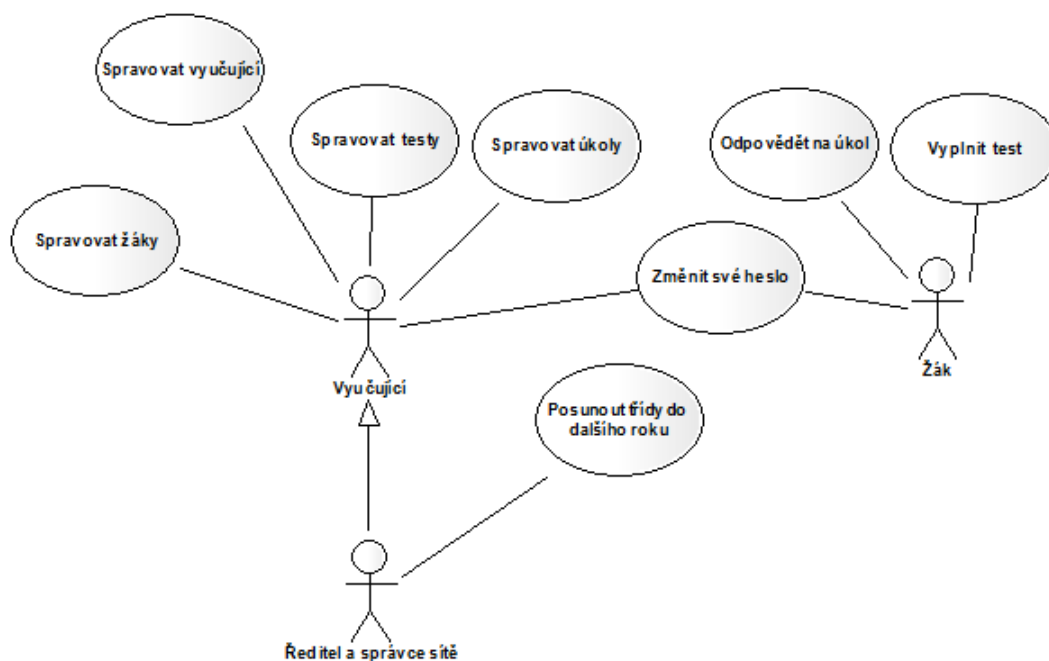


Obrázek 4. Okno přihlášení

V případě dostupnosti serveru je zobrazeno hlavní okno přihlášení a pokud je program spuštěn poprvé, je také automaticky spuštěna hlasová nápověda. Pro samotné přihlášení je potřeba vyplnit uživatelské jméno a heslo. Program podporuje export údajů jak vyučujících, tak žáků. Po přidání konkrétního žáka či vyučujícího se tedy předpokládá předání vygenerovaných přihlašovacích údajů příslušnému uživateli. Uživatel si také, zaškrtnutím položky *zapamatovat*, může vybrat, zda si chce přihlašovací údaje uložit do paměti počítače. Pokud tak učiní, při příštím spuštění počítače mu již program automaticky vyplní přihlašovací jméno a heslo. Pro případné zrušení uložení stačí uživateli položku odškrtnout a následně se do programu přihlásit.

Po kliku na tlačítko *Přihlásit* může nastat několik případů:

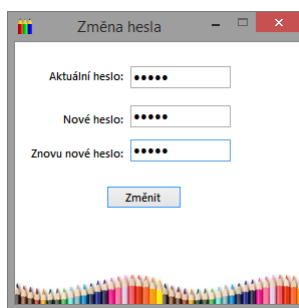
- **Nevyplněné údaje:** V případě nevyplnění jména nebo hesla je uživateli zobrazen pouze červený varovný vykřičník u příslušného nevyplněného pole.
- **Špatné jméno nebo heslo:** Pokud uživatel zadá špatné jméno nebo heslo, bude o tom následně informován varovnou hláškou a vrácen zpět na okno s přihlášením.
- **Správné údaje:** Pokud uživatel zadá správné údaje a je to jeho první přihlášení, bude vyzván z bezpečnostních důvodů ke změně svého hesla. Bez jeho změny mu nebude přihlášení povoleno. Následně se otevře hlavní okno programu.



Obrázek 5. Diagram případů užití

6.4. Změna hesla

Pro změnu hesla je potřeba se nejprve přihlásit pod svým aktuálně používaným heslem. Pokud toto heslo žák nezná, může mu ho změnit jeho vyučující. Pokud heslo nezná vyučující, měl by požádat svého správce sítě. Po přihlášení je funkce dostupná v kontextovém menu v položce *Uživatel*, následně pak v položce *Změna hesla*. Poté se otevře okno změny hesla (obr. 6.).



Obrázek 6. Okno změny hesla

Pro změnu hesla jsou zapotřebí tři kroky, nejprve uživatel musí zadat své

aktuálně platné heslo k přihlášení. Dále bude vyzván k zapsání chtěného nového hesla. Z důvodu bezpečnosti je také povinen jím zvolené heslo zapsat do další položky znovu. Po odkliknutí tlačítka *Změnit* nastane jeden z těchto stavů:

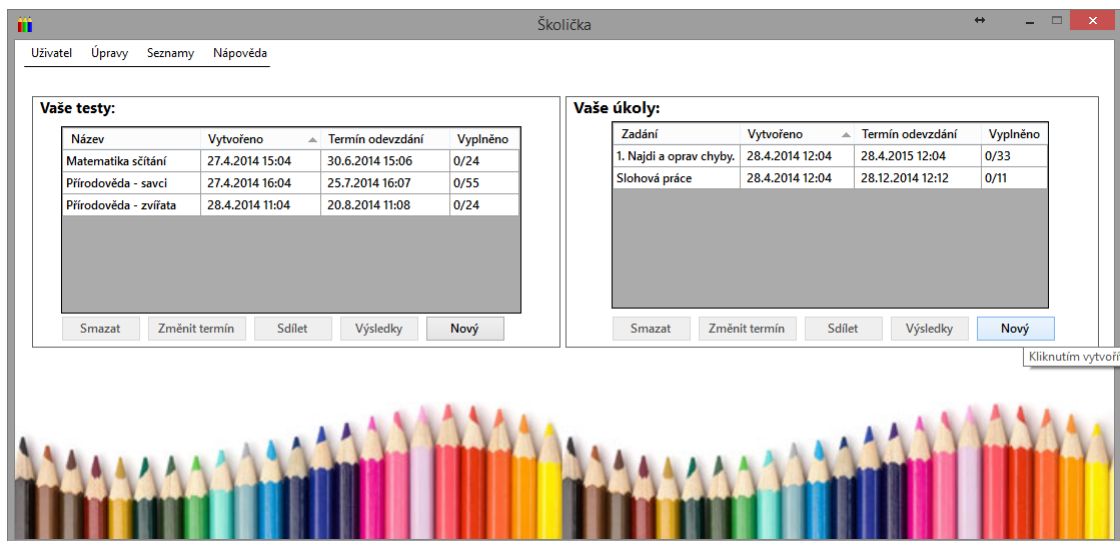
- **Nevyplněná pole:** Pokud uživatel nevyplní alespoň jedno ze tří polí, bude na tuto skutečnost následně upozorněn červenou varovnou ikonou se symbolem vykřičníku u příslušného nevyplněného pole.

- **Stejné heslo jako jméno:** V případě, že uživatel zadá heslo stejné, jako má své přihlašovací jméno, bude na to upozorněn varovným vyskakovacím oknem a změna nebude dokončena.
- **Nesprávné potvrzení hesla:** Do stavu se uživatel dostane, pokud jeho nově zadané heslo není shodné s textem zadaným do pole *Znovu nové heslo*. Změna tak nebude dokončena.
- **Heslo změněno:** V případě, že nenastane žádný z výše popsanych stavů heslo bude úspěšně změněno. Tuto informaci program sdělí uživateli pomocí vyskakovacího okna, po jehož odkliknutí bude okno *Změna hesla* uzavřeno.

6.5. Vyučující

6.5.1. Hlavní okno

Hlavní okno (obr. 7.) je rozděleno na dvě základní části. První z nich je kontextové menu u horního okraje okna. Druhou část tvoří dvě základní nabídky s přehledy učitelem vytvořených testů na levé straně a učitelem vytvořených úkolů na straně pravé.



Obrázek 7. Hlavní okno vyučujícího

Kontextové menu hlavního okna je rozděleno na tři skupiny:

- **Uživatel:** nabídka uživatel představuje činnosti, týkající se momentálně přihlášeného uživatele. Obsahuje tyto položky:
 - **Změnit údaje:** Kliknutím otevře uživatel okno, ve kterém má možnost změnit své základní údaje.

- **Změnit heslo:** Kliknutím otevře uživatel okno s možností změnit své aktuálně používané heslo při přihlašování.
- **Odhlásit:** Po kliknutí na toto tlačítko bude aktuálně přihlášený uživatel odhlášen a aktuální hlavní okno se uzavře.
- **Úpravy:** Při kliknutí na tuto položku kontextového menu je uživateli zobrazena kontextová nabídka týkající se práce s žáky a vyučujícími na jeho škole. Nabídka je rozdělena do tří až čtyř funkčních celků. Obsahuje tyto položky:
 - **Přidat žáky:** Kliknutím uživatel otevře okno umožňující mu přidat do programu nového žáka nebo nové třídy.
 - **Přidat vyučujícího:** Po kliknutí na tuto položku nabídky bude mít uživatel přístup k oknu, které mu umožňuje přidání nového vyučujícího do školního systému.
 - **Odebrat žáka:** Položka obsahující funkci pro odebrání již vytvořeného žáka nebo třídy na škole. Pro odstranění celé třídy při ukončení devátého ročníku by měl uživatel využít funkci *Nový školní rok*.
 - **Změnit třídu žáka:** Po kliknutí bude mít vyučující možnost změnit třídu konkrétního žáka za jinou. Pro přesun celých tříd na konci školního roku slouží funkce *Nový školní rok*.
 - **Změnit heslo žáka:** Kliknutím otevře uživatel okno s možností změnit přihlašovací heslo jakéhokoliv žáka. Pokud žák své aktuální heslo zná, má možnost si ho změnit i sám.
 - **Nový školní rok:** Tato nabídka je dostupná pouze pokud je přihlášen uživatel s přihlašovacím jménem admin. Zpřístupní uživateli možnost přesunout jednotlivé třídy o rok výše a smazat deváté ročníky.
- **Seznamy:** Kliknutím na tuto položku kontextového menu se vyučujícímu otevře kontextová nabídka obsahující odkazy na seznamy vyučujících a studentů. Nabídka obsahuje tyto položky:
 - **Seznam učitelů:** Kliknutím vyučující otevře okno se seznamem učitelů.
 - **Seznam žáků:** Kliknutím vyučující otevře okno se seznamem žáků přidaných do programu.
- **Nápověda:**
 - **Zobrazit nápovědu:** Kliknutím na položku menu vyučující otevře okno s nápovědou. Tuto akci může provést i na jakémkoliv jiném okně stisknutím klávesy F1.

Další částí okna jsou hlavní nabídky. Jedná se o nabídku testů a úkolů. Ty mohou nabývat dvou základních stavů. Pokud tyto nabídky ještě nebyly naplněny, vyučující tedy ještě nepřidal příslušnou položku, je zde zobrazeno pouze tlačítko, po jehož kliknutí začne vytváření příslušné položky. Pokud nabídka již naplněna byla, zobrazí se místo tlačítka seznam všech položek příslušné nabídky.

V nabídce úkolů poskytuje každá svá položka, tedy konkrétní úkol, údaj o jeho *zadání*. Pokud je delší než šířka sloupce ve kterém je zapsáno, je možno si po najetí kurzorem myši prohlédnout celý obsah v zobrazeném okně. Položka dále obsahuje *termín odevzdání*, tedy datum do kterého je žák povinen úkol odevzdat. Pak také informaci o *vytvoření*, kde nám zobrazený datum říká kdy vyučující daný úkol zadal a v neposlední řadě také údaj *vyplněné* obsahující dvě čísla. První z čísel udává kolik žáků úkol odevzdalo, druhé číslo pak kolika žákům byl úkol zadán. Pokud jsou si tato čísla rovna, je úkol zbarven zeleně. V opačném případě pak červeně.

V nabídce testů jsou položky popsány podobnými údaji. První z údajů je *název*. Ten nám říká, pod kterým názvem vyučující test vložil do systému. Pokud je název delší než šířka sloupce, je po najetí kurzorem myši na tento text zobrazen v celém znění. Jako další údaj zde figuruje *vytvořeno*, které udává datum, kdy byl daný test vložen do systému. Dalším datem je *termín odevzdání*, který určuje konkrétní den a minutu, do které musí být test vyplněn žákem. Poslední položkou je pak *vyplněno*, předávající vyučujícímu informace ve dvou číslech. První z nich udává počet žáků kteří test již vyplnili, druhé celkový počet žáků, kterým byl test zadán. Pokud se tato čísla rovnají, je konkrétní test zbarven zeleně, v opačném případě je červený.

V obou nabídkách má vyučující možnost (kliknutím na konkrétní název sloupce nabídky) měnit řazení položek podle svých představ. Jednotlivým sloupcům lze také měnit šířku, a to kliknutím a držením levého tlačítka myši v mezní části sloupce. Dále může jednotlivé položky nabídky vybírat, a to jak pomocí myši, tak pomocí klávesnice. Pokud uživatel vybere jednu položku, zpřístupní se mu tlačítka *Změnit termín* a *Smazání*. První z nich nám zpřístupní okno, ve kterém má vyučující možnost jeho vybranému úkolu či testu změnit termín odevzdání. Druhé nám dává možnost smazat jednotlivé testy či úkoly. Tlačítko *Smazání* je dostupné i při výběru více položek ze seznamu, buď tažením, držením klávesy Shift a posunem šipkou, nebo držením klávesy Control a klikáním na jednotlivé položky nabídky. Tlačítko samotné pak, po zobrazení varovné hlášky a potvrzení uživatelem, smaže veškeré označené položky.

6.5.2. Vytváření testu

Jednou z hlavních funkcí programu je vytváření testu. To je rozděleno do tří částí a vyučující má několik možností jak se k ní dostat. První z nich se mu naskytá tehdy, pokud ještě nevytvořil žádný test a jedná se o velké tlačítko zobrazené namísto seznamu již vytvořených testů. Druhou možnost má dostupnou

neustále a jedná se o tlačítko *Nový* v levé části hlavního okna, pod seznamem již vytvořených testů.

První část požaduje po vyučujícím vyplnit název testu a termín do kterého mají žáci test vyplnit. Pokud žáci test do termínu nevyplní, již nebudou mít možnost test otevřít. Výběr může provést kliknutím na šipku vedle data a vybráním data z kalendáře, nebo ručním přepsáním data. Dále uživatel zaškrtnutím položky *Různě bodované otázky* vybere, zda chce aby každá otázka mohla být za jiný počet bodů. Poté by každá otázka mohla mít jinou váhu při konečném výpočtu, zda žák v testu uspěl. Pokračovat musí stiskem tlačítka *Přidat otázky*.

Obrázek 8. Okno přidávání otázky

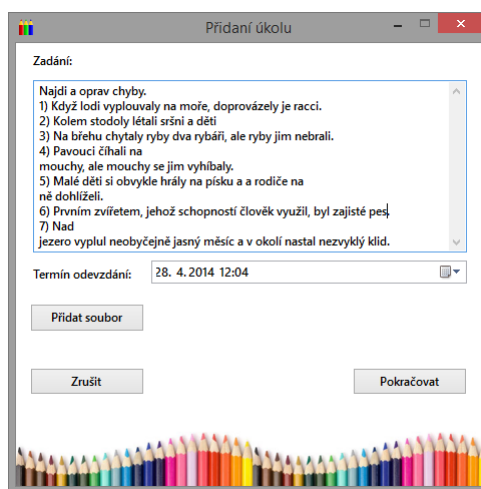
V druhé části vytváření testu je zapotřebí přidat otázky testu. V horním levém rohu je umístěna informace kolikátá otázka je vkládána. Pokud jsou zvoleny různě bodované otázky, zobrazí se v horním pravém rohu pole, do kterého lze zapsat, za kolik je vkládána otázka bodů. Následuje pole pro vepsání otázky. K textu, který vyučující zadá, může vybrat obrázek, který se nachází u něj v počítači. Vybrat mu je dovoleno pouze obrázky ve formátech jpg, gif a png, které nepřekračují velikost 10MBit. Po úspěšném vybrání obrázku bude vedle textu otázky zobrazen zmenšený náhled obrázku. Kliknutím na tento náhled je možné zobrazit obrázek v novém okně v plné velikosti. Zároveň (po přidání obrázku) má vyučující možnost ho smazat stiskem tlačítka s křížkem nad jeho náhledem. Dále má povinnost vyplnit všechny čtyři možné odpovědi. Může tak udělat buď vyplněním textů u jednotlivých písmen otázek nebo opět vybráním obrázku za stejných podmínek jako u otázky. Poslední krok u přidání otázky je výběr správné odpovědi ze se-

znamu čtyř možností. Pokud chce přidat další otázku do testu, musí kliknout na tlačítko s nápisem *Přidat další*. Pokud vyučující nevyplnil všechna potřebná pole, je o tom informován varovnou ikonkou u nevyplněného pole. V opačném případě je otázka vložena k testu a uživatel může zadat další otázku. Pokud si již vyučující nepřeje přidávat další otázky musí kliknout na tlačítko *Dokončit test*. V případě že nebyly vyplněny všechny povinné položky si může vyučující zvolit, zda má být právě vkládaná otázka ignorována.

V poslední části má již vyučující povinnost pouze vyplnit pole s minimálním počtem správně zodpovězených otázek, případně minimálním počtem bodů k úspěšnému splnění testu a následně vybrat studenty, kterým bude test zobrazen. První možnost jak tak může učinit, je vybrání třídy ze seznamu a následné kliknutí na tlačítko *Přidat celou třídu*. Druhou možností (po zvolení třídy) je výběr konkrétního žáka a následný klik na tlačítko *Přidat žáka*. Pokud vyučující do seznamu přidal žáka, kterému test zobrazit nechce, může ho ze seznamu odstranit vybráním a následně kliknutím na tlačítko *Odebrat vybraného*. Po vybrání všech žáků pokračuje stiskem *Dokončit*. Pokud je seznam žáků prázdný, může se vyučující rozhodnout, zda chce žáky přidat později.

6.5.3. Vytváření úkolu

Další důležitou funkcí programu, je vytváření úkolů a jejich zadávání jednotlivým žákům. K této funkci se vyučující může dostat dvěma způsoby. První z jich je dostupný pouze tehdy, pokud vyučující nemá vytvořený ještě žádný úkol. Jedná se o velké tlačítko, zobrazené namísto výpisu vytvořených úkolů. Jako druhý způsob figuruje tlačítko *Nový*, které je umístěné pod seznamem vytvořených úkolů.



Obrázek 9. Okno s vytvářením úkolu

Samotné vytváření úkolu probíhá ve dvou fázích. V první z nich (obr. 9.) žádá program vyučujícího o vyplnění textového zadání, s možností přiložení jakéhokoliv souboru ze svého počítače. To lze provést stiskem tlačítka s názvem *Přidat soubor*. Z důvodu omezené kapacity serveru je povoleno přiložit soubor o maximální velikosti 10MBit. Další položkou v okně vytváření úkolu je výběr data u *Termínu odevzdání*. Vyučující má možnost provést výběr pomocí vestavěného kalendáře kliknutím na malou šipku, nebo může datum a čas zadat ručně přímo do pole. Pokračovat k další fázi může stiskem tlačítka *Pokračovat*. V případě přidání souborové přílohy může dojít k zobrazení posuvníku udávajícího průběh nahrávání souboru na server. Po dokončení tohoto nahrávání okno s vytvářením zmizí a vyučujícímu je zobrazena další fáze.

V druhé a zároveň poslední fázi má vyučující za úkol vybrat ty žáky, kterým je právě vytvářený úkol určen. Přidat může všechny žáky z jedné třídy naráz, nebo pouze jednotlivce. První možnost je dostupná po výběru třídy z nabídky *Třída* a následném kliku na tlačítko *Přidat třídu*. Druhá možnost spočívá ve vybrání konkrétního žáka, samozřejmě po výběru třídy ve které se nachází a následném kliknutí na tlačítko *Přidat žáka*. Pokud vyučující přidá do seznamu žáka, kterému úkol určen není, může ho následně odebrat jeho označením v seznamu a kliknutím na tlačítko *Odebrat*. Dalším krokem vyučujícího, pro dokončení vytváření úkolu, je kliknutí na tlačítko *Dokončit*. Jestliže je seznam žáků prázdný, může se vyučující rozhodnout zda je chce přidat později.

6.5.4. Žáci testu a úkolu

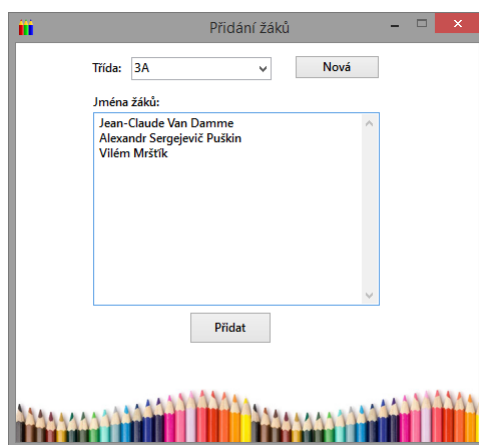
Vyučující může dvojklikem na záznam úkolu nebo testu v hlavním okně otevřít okno s podnabídkou. V té pak nalezne seznam všech žáků, kterým test či úkol zadal. Seznamy obsahují jména těchto žáků i informace, zda již test či úkol žák odevzdal. U testů také počet získaných bodů a u úkolů datum, kdy na něj žák odpověděl. V případě, že na test či úkol žák odpověděl, má vyučující možnost na položku dvakrát kliknout nebo položku v seznamu vybrat a kliknout na tlačítko *Otevřít položku*. Následně se mu zobrazí podrobnější informace o dané položce. V případě testů se jedná o okno, v němž vyučující může procházet jednotlivé otázky testu a u každé z nich si prohlédnout, jakou odpověď žák zvolil. Pokud je odpověď správná, je podbarvena zelenou barvou. Pokud je naopak odpověď špatná, je správná odpověď podbarvena červeně. Pokud otevře vyučující informace o úkolu, může si prohlédnout v otevřeném okně znovu svoje zadání a také žakovu textovou odpověď. Pokud k této odpovědi nebo otázce byl přiložen soubor, může ho vyučující (kliknutím na tlačítko *Zobrazit soubor*) uložit do svého počítače, odkud mu bude soubor následně otevřen.

6.5.5. Změna údajů

Pokud vyučující chce změnit své údaje, může tak učinit po kliknutí na položku *Uživatel* kontextového menu a následně na položku *Změna údajů*. V následujícím okně má uživatel k dispozici jednotlivá pole, týkající se jeho údajů uložených v systému. Při otevření jsou již předvyplněny aktuálními hodnotami. Pokud uživatel chce nějakou z těchto hodnot změnit, musí ji v daném poli přepsat. Pokud již provedl všechny potřebné změny, uloží je do systému stiskem tlačítka *Změnit*.

6.5.6. Přidání žáků

Pokud vyučující má v úmyslu přidat do programu nové žáky, tak mu to program umožní po výběru položky *Úpravy* v kontextovém menu a následně kliknutím na položku *Přidat žáky*.



Obrázek 10. Okno s přidáním žáka

Po otevření okna (obr. 10.) nejprve vyučující vybere v roletové nabídce v horní části okna třídu, do které má daný žák či žáci patřit. Pokud se v něm potřebná třída nenachází, měl by pokračovat stiskem tlačítka s nápisem *Nová*. Stisknutím tohoto tlačítka se vyučujícímu otevře nový formulář s položkami *Ročník*, pro číslici nové třídy a *Písmeno*, pro identifikační písmeno nové třídy. Pokud se vytvářený ročník na škole neopakuje, vyučující není povinen písmeno vyplnit a program následně, po stisku tlačítka *Přidat*, doplní automaticky písmeno „A“.

Po vybrání třídy je potřeba zapsat do pole *Jména žáků*, jejich jednotlivá jména ve tvaru křestní jméno a příjmení jednoho žáka na jednom řádku. Jeden žák může mít více křestních jmen. Po kliknutí na tlačítko *Přidat* mohou nastat tyto události:

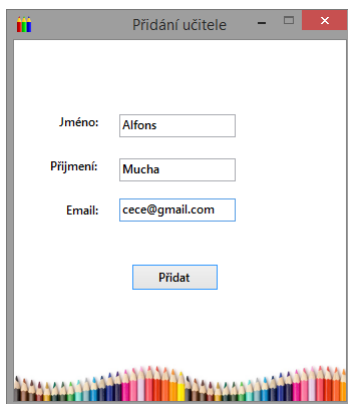
- **Žák zadán ve špatném formátu:** V případě že vyučující napíše na jeden řádek pouze jedno slovo, je informován varovnou hláškou, že jméno žáka je

ve špatném tvaru. Přidání se tak neprovede.

- **Žáci přidání:** Pokud je vše v pořádku, uživatel je informován hláškou, že přidání proběhlo úspěšně. Následně je okno s přidáním žáků uzavřeno a žákovi je vygenerováno přihlašovací jméno a heslo. Jméno se skládá z prvních čtyř písmen žákova příjmení, prvních dvou písmen žákova jména a čísla. Číslo je určeno podle toho, zda se někdo s potencionálně stejným uživatelským jménem v systému již nachází. Pokud tedy tomu tak není, doplní se přihlašovací jméno nulou, pokud je, doplní se o číslovku s počtem shodných přihlašovacích jmen. Z takto vygenerovaného jména je následně odstraněna diakritika a velká písmena převedena na malá. Heslo které je žákovi přiděleno je stejné jako přihlašovací jméno, má však povinnost si ho při prvním přihlášení změnit.

6.5.7. Přidání vyučujícího

Pro přidání nového vyučujícího do systému je v nabídce kontextového menu, pod pojmem *Úpravy*, dostupná položka *Přidat vyučujícího*.



Obrázek 11. Okno s přidáním vyučujícího

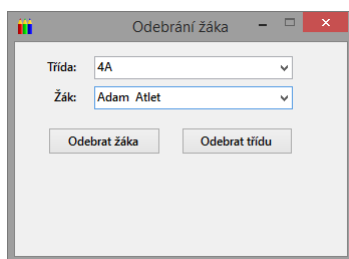
Po kliknutí se vyučujícímu otevře nové okno (obr. 11.) se třemi prázdnými poli. První pro křestní jméno přidávaného vyučujícího, další pak pro jeho příjmení a poslední pro emailovou adresu. Ta slouží pouze ke kontaktování ze strany správce sítě v případě nějakých problémů s programem. Po kliknutí na tlačítko *Přidat* může vyučující dostat jednu z těchto odpovědí:

- **Nevyplněna všechna pole:** Pokud vyučující nevyplní alespoň jedno ze tří polí, je o této skutečnosti informován, v místě nevyplněného pole, varovnou ikonkou.
- **Emailová adresa není ve správném formátu:** Je potřeba zadat správnou emailovou adresu. Přidání tak nebude dokončeno.

- **Vyučující byl přidán:** V případě, že vyučující neporuší žádnou z výše uvedených podmínek, se vyučující přidá. Při vytvoření vyučujícího jsou mu automaticky vygenerovány přihlašovací údaje. Ty se, stejně jako u žáka, skládají z prvních čtyř písmen jeho příjmení, prvních dvou písmen jeho jména a čísla. To se odvíjí od počtu potencionálně shodných přihlašovacích jmen. Z takto vygenerovaného jména je následně odstraněna diakritika a velká písmena převedena na malá.

6.5.8. Odebrání žáka

V případě, že by ze školy odešel nějaký žák, má vyučující možnost ho ze systému odstranit. Může tak učinit otevřením kontextového menu *Úpravy* a následně výběrem položky *Odebrat žáka*. Pokud má vyučující v plánu kvůli ukončení studia odebrat celý devátý ročník, měl by kontaktovat ředitele, aby tak učinil přes funkci *Nový školní rok*.



Obrázek 12. Okno s odebráním žáka

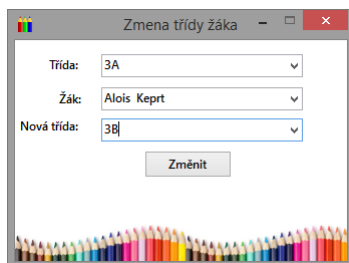
Po otevření okna (obr. 12.) musí uživatel pokračovat nejprve výběrem třídy, ve které se žák nachází. Pokud by měl v plánu odstranit celou třídu, například kvůli chybnému založení, musí pokračovat stiskem tlačítka *Odebrat třídu*. Pokud by měl v plánu odebrat pouze jednoho žáka, musí nejprve pokračovat výběrem jeho jména v druhém seznamu a následně kliknout na tlačítko *Odebrat žáka*. Uživatel by si měl uvědomit, že s odebráním celé třídy zároveň odebírá i všechny žáky v ní obsazené. S každým odebíraným žákem odebírá také veškeré jeho výsledky testů a odpovědi na úkoly.

6.5.9. Odebrání vyučujícího

Vyučující má možnost vyučujícího také ze systému odstranit. Může tak učinit vybráním položky *Odebrat vyučujícího* v nabídce kontextovém menu *Úpravy*.

Vyučující jednoduše vybere jméno vyučujícího, kterého chce odstranit a pokračuje stiskem tlačítka *Odebrat*. Vyučující může odstranit jakéhokoliv vyučujícího kromě sebe, až na výjimku, kterou tvoří administrátorský účet. Při odebrání by měl vyučující brát na vědomí, že s odstraněním vyučujícího se také odstraní všechny jeho testy a zadání úkolů, které vytvořil. Zároveň s nimi se smažou i veškeré reakce žáků na ně.

6.5.10. Změna třídy žáka



Obrázek 13. Okno se změnou třídy žáka

Pokud nějaký žák změni svoji třídu, vyučující ho nemusí mazat a přidávat znovu do třídy nové, ale může využít možnosti *Změnit třídu žáka* v kontextovém menu *Úpravy*. Tuto funkci by měl však využít pouze v případě, že žák mění třídu v rámci svého ročníku. Pokud chce vyučující změnit třídu žáka při příležitosti konce školního roku, měl by požádat svého ředitele, aby využil funkci *Nový školní rok*.

Po otevření okna (obr. 13.) uživatel musí vybrat postupně tři položky. První z nich obsahuje aktuální třídu žáka, kterého chce uživatel přesunout. Po zvolení této třídy, program načte jména žáků do druhé položky, ze které vyučující potřebného žáka vybere. Poslední položka okna představuje třídu, do které uživatel chce žáka přesunout. Pokračovat musí stiskem tlačítka s názvem *Změnit*. Pokud uživatel nevybere nějakou ze tří položek, bude na to upozorněn červenou ikonkou u nevybraného pole.

6.5.11. Změna hesla žáka

Pro změnu hesla žáka je dostupná v kontextovém menu pod nabídkou *Úpravy* položka *Změnit heslo žáka*. Žák si však může své heslo po přihlášení změnit také sám.

V nově otevřeném okně, je zapotřebí nejprve vybrat třídu, ve které se žák nachází. Po výběru konkrétního žáka, je vyučující povinen dvakrát zapsat heslo nové. V druhém případě se jedná pouze o bezpečnostní prvek. Aby došlo ke změně, musí následně vyučující kliknout na tlačítko *Změnit*. Po této akci mohou nastat tyto události:

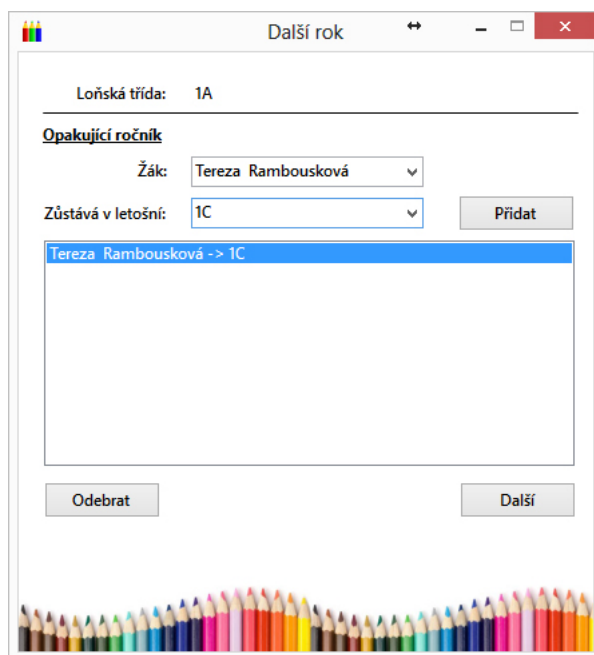
- **Nevyplněné pole či nevybraná položka:** Pokud uživatel nevyplní nějaké pole, nebo nevybere třídu či žáka, bude o tom informován varovnou ikonkou u chybného pole.
- **Hesla se neshodují:** V případě že uživatel nezadá potvrzení nového hesla stejně, jako zadal nové heslo, nebude změna provedena.

- **Heslo změněno:** Pokud vyučující vyplní všechna pole správně, bude informován hláškou o úspěšné změně hesla.

6.5.12. Nový školní rok

V programu je při prvním spuštění dostupný pouze jeden účet. Na tento účet s přihlašovacím jménem „admin“ by měli mít přístup pouze dva lidé a to správce sítě a ředitel či ředitelka školy. Pokud z tohoto účtu uživatel klikne na položku kontextového menu *Úpravy*, uvidí navíc položku *Nový školní rok*. Tuto funkci programu by měl ředitel využít při přechodu školního roku.

Tato funkce programu má tři základní části. V první z nich bude uživatel ihned dotázán, zda již do programu přidal nové první třídy. Při záporné odpovědi se otevře dialogové okno s vytvořením nové třídy. Zde by ředitel měl vytvořit všechny první třídy, které nově nastupují do školy. Po zavření tohoto okna bude pokračovat k další části funkce programu.



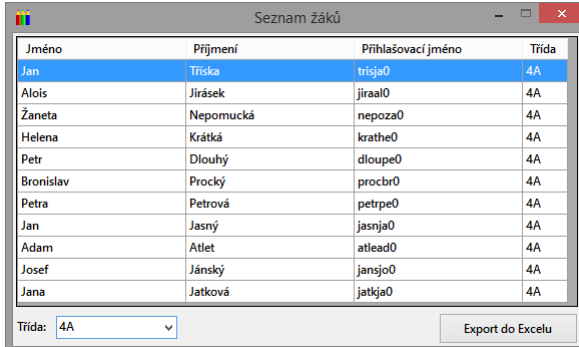
Obrázek 14. Okno s posunem třídy do dalšího roku

Pokud je v systému třída, která ještě nebyla posunuta, je uživateli zobrazeno nové okno (obr. 14.). To slouží k posunu jedné, dosud neposunuté, třídy do dalšího ročníku a zároveň k výběru žáků, kteří ročník opakují. V první části uživatel vidí, kterou třídu zrovna posouvá. V druhé části (pod názvem *Opakující ročník*) je řediteli připravena možnost vybrat ze seznamu žáků postupně ty, kteří nepostupují do dalšího ročníku spolu s ostatními. Po výběru žáka musí vybrat nejprve jeho novou třídu a poté potvrdit přesun tlačítkem *Přidat*. Pokud se ředitel spletě,

má možnost žáka ze seznamu níže odebrat tlačítkem *Odebrat*. Pokud již vybral všechny žáky opakující ročník, může dále pokračovat stiskem tlačítka *Další*. Pokud se v programu nachází další neposunutá třída, musí uživatel druhou část opakovat, dokud neposune všechny třídy. Při příležitosti posunu devátých ročníků, má uživatel pouze za úkol vybrat žáky, kteří devátý ročník opakují. Po stisku tlačítka *Další* pak bude třída i se zbytkem žáků ze systému odstraněna.

6.5.13. Seznam žáků a vyučujících

Po přidání žáků či vyučujících do systému může uživatel vypsat jejich seznam. K této možnosti se dostane kliknutím na položku *Seznamy* v kontextovém menu a následně výběrem položky *Seznam žáků*, případně *Seznam vyučujících*. Předpokládá se, že funkci uživatel bude využívat převážně k zjištění přihlašovacích jmen do systému.



Jméno	Příjmení	Přihlašovací jméno	Třída
Jan	Triska	trjsja0	4A
Alois	Jirásek	jiraa0	4A
Žaneta	Nepomucká	nepoza0	4A
Helena	Krátká	krathe0	4A
Petr	Dlouhý	dloupe0	4A
Bronislav	Procký	procbr0	4A
Petra	Petrová	petrpe0	4A
Jan	Jasný	jasnja0	4A
Adam	Atlet	atlead0	4A
Josef	Jánský	jansjo0	4A
Jana	Jatková	jatkja0	4A

Třída: 4A

Export do Excelu

Obrázek 15. Okno se seznamem žáků

Okno (obr. 15.) obsahuje jednoduchý seznam všech přidávaných žáků či vyučujících, s jejich křestními jmény, příjmeními a přihlašovacími jmény. V seznamu lze (pomocí kliknutí na příslušný název sloupce) měnit výchozí seřazení a také měnit šířku jeho jednotlivých sloupců. Dalším ovládacím prvkem v tomto okně je tlačítko *Export do Excelu*, to nám umožňuje výše zobrazený seznam žáků či vyučujících převést do aplikace Excel, z balíku kancelářských programů Office. Pokud tento program není v počítači uživatele nalezen, je zobrazena pouze varovná hláška. V opačném případě je uživateli spuštěn program Excel s výše zmíněným seznamem. Seznam žáků navíc obsahuje možnost filtrace podle třídy. Pokud vyučující vybere jednu ze tříd v seznamu v levém dolním rohu, seznam žáků se redukuje pouze na žáky z této třídy.

6.6. Žák

6.6.1. Hlavní okno

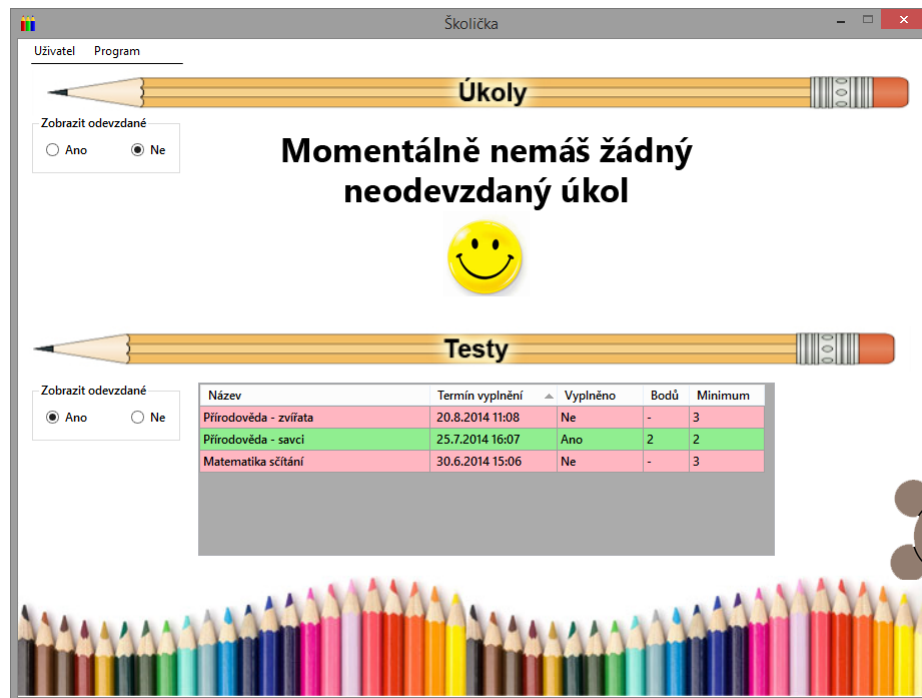
Hlavní okno žáka (obr. 16.) je rozděleno na tři základní části. První z nich je kontextové menu umístěné v horní části okna, druhou částí je seznam přidělených testů, který může být prázdný a posledním z nich je seznam přidělených úkolů, který taktéž může být prázdný.

Kontextové menu obsahuje tyto položky:

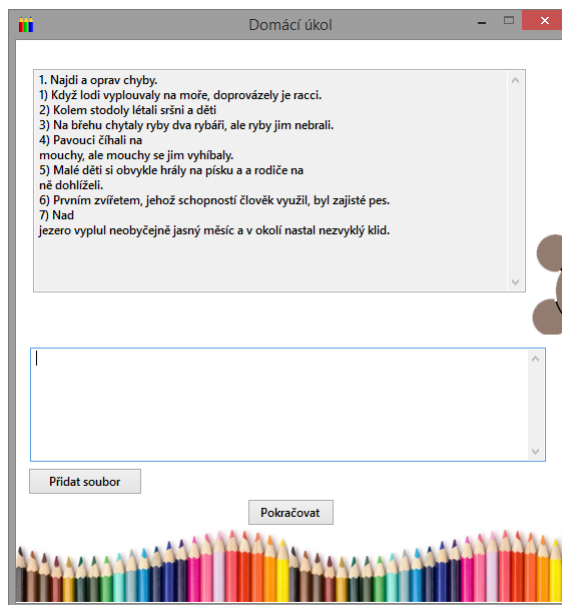
- **Uživatel:** Položka obsahující základní funkce spojené s uživatelským účtem. Obsahuje tyto položky:
 - **Změnit heslo:** Položka sloužící k změně aktuálně používaného hesla k žakovskému účtu.
 - **Odhlásit:** V případě, že uživatel odchází od počítače, může využít možnosti odhlášení svého účtu z programu. Po odhlášení mu bude zobrazeno přihlašovací okno.
- **Program:** Položka obsahující nastavení základních funkcí programu. Obsahuje tyto položky:
 - **Vypnout uvítací hlášku:** Pokud žák provedl první přihlášení, byla mu automaticky vypnuta uvítací hláška hlasové nápovědy. Pokud by ji chtěl znovu zapnout, provede tak kliknutím na tuto položku menu.
 - **Zastavit kontrolu:** V případě, že žák chce zastavit pravidelnou kontrolu nových úkolů a testů, může tak provést po kliknutí na tuto položku. Žák pak nové úkoly a testy zjistí pouze opětovným otevřením hlavního okna.

V části týkající se úkolů, má žák nejprve k dispozici přepínač s nadpisem *Zobrazit odevzdané* a možnostmi ano či ne. Ty určují, zda se v sousedním seznamu úkolů objeví i ty úkoly, na které již žák odpověděl. Samotný seznam tvoří jednotlivé položky úkolů s několika údaji. *Zadání*, to lze, v případě větší délky, zobrazit celé najetím kurzoru myši. Dále *Termín odevzdání* a stav, který říká zda na úkol již bylo odpovězeno. To také určuje, že jednotlivé úkoly v seznamu jsou barevně rozlišeny. V případě, že na úkol ještě odpovězeno nebylo, je podbarven červeně. V opačném případě zeleně. Žák může na jednotlivé položky seznamu klikat, nebo je může označit a následně použít tlačítko *Otevřít*, čímž může vyvolat několik událostí závislých na položkách seznamu. Pokud je úkol:

- **Po termínu odevzdání a nevyplněný:** Žákovi je zobrazeno okno s informací, že daný úkol je již po termínu odevzdání a nelze ho tak otevřít. Po odkliknutí tohoto okna je žák vrácen do hlavního okna.



Obrázek 16. Hlavní okno žáka



Obrázek 17. Okno s odevzdáním úkolu

- **Před termínem odevzdání a nevyplněný:** V případě, že žák ještě na úkol neodpověděl a jeho termín na odpověď ještě nevypršel, je mu předloženo jednoduché okno (obr. 17.). V jeho horní části nalezne pole

s textovým zadáním. V případě, že je k zadání přiložen soubor, lze ho zobrazit kliknutím na tlačítko *Zobrazit* a následně vybrat umístění, kam chce soubor na svém disku uložit. Pokud má uživatel na svém počítači program na otvírání tohoto souboru, bude soubor následně otevřen. V další části okna musí žák vložit svou textovou odpověď do příslušného pole. Má také možnost přiložení souboru, po stisku tlačítka *Přidat soubor*. Maximální velikost tohoto souboru však nesmí přesáhnout 10MBit. Pokud žák vložil špatný soubor, může ho ze své odpovědi ještě před odevzdáním smazat kliknutím na tlačítko s ikonkou křížku. Svoji odpověď může žák odeslat stisknutím příslušného tlačítka. Pokud nejsou vyplněny všechny potřebné položky, je na to upozorněn ikonou s červeným vykřičníkem u nevyplněného pole. Pokud během vyplňování vyučující zadaný úkol smazal, bude o této skutečnosti žák informován a jeho odpověď nebude nikde zaznamenána. V případě, že se žádná z těchto skutečností nenaskytne, bude odpověď vložena do systému. Pokud k odpovědi byl přiložen soubor, může žákovi být zobrazen posuvník, který značí stav odesílání souboru na server a žák musí vyčkat než bude odeslání dokončeno.

- **Vyplněný:** Žákovi je zobrazeno okno s odpovědí na úkol. Pokud byl k zadání či odpovědi přiložen soubor, může ho žák znovu zobrazit a to kliknutím na příslušné tlačítko *Zobrazit*. V následujícím dialogu musí nejprve vybrat umístění a název souboru, po jeho uložení se soubor (pokud je to možné) automaticky ze svého umístění spustí.

Poslední část hlavního okna je věnovaná žakovým testům. Obsahuje jejich seznam, kde každý jednotlivý test obsahuje informace o termínu vyplnění, informaci zda byl test již vyplněn, počet získaných bodů, potřebný minimální počet bodů pro splnění a název testu (v případě, že je delší než šířka pole, ve kterém je zapsán, bude po najetí kurzorem zobrazen v plné délce). Žákovi je umožněno měnit položkám seznamu jejich uspořádání kliknutím na název sloupce, podle kterého chce žák položky seřadit. Jednotlivé testy v seznamu jsou barevně rozlišeny u příležitosti stavu jejich vyplnění. Pokud byl test již vyplněn, je podbarven zeleně, v opačném případě červeně. Zobrazení odevzdaných testů může žák měnit přepínačem s nápisem *Zobrazit odevzdané*, umístěným vlevo od seznamu testů. Každý test může žák otevřít, a to buď pomocí dvojkliku na jeho položku v seznamu, nebo pomocí tlačítka *Otevřít*. Po otevření může nastat několik možností odvozených od stavu, ve kterém se test zrovna nachází. Pokud je test ve stavu:

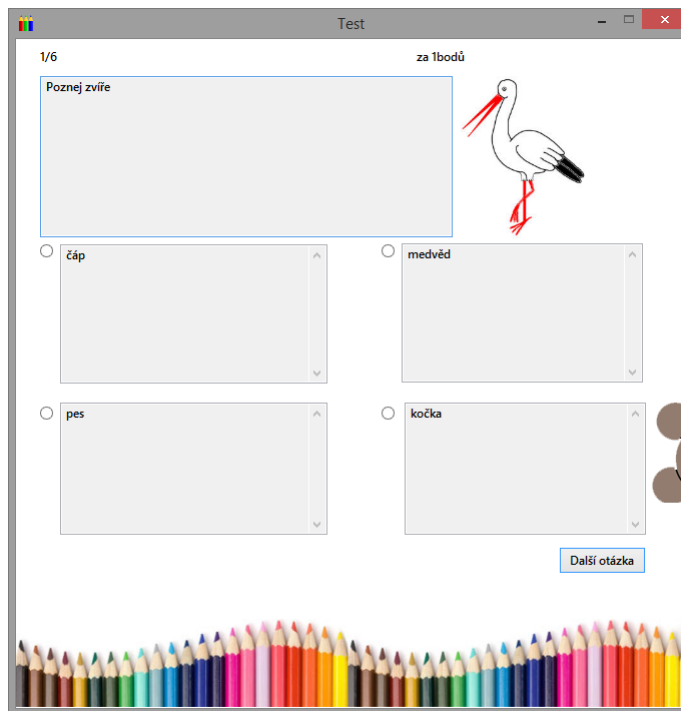
- **Po termínu odevzdání a nevyplněný:** Zobrazí se žákovi hláška informující o již skončeném termínu odevzdání a žákovi nebude umožněno test vyplnit.
- **Před termínem odevzdání a nevyplněný:** V případě, že žák má možnost test vyplnit, bude mu zobrazeno okno s vyplněním testu (obr. 18.).

Žákovi se budou zobrazovat postupně všechny otázky a na každou z nich má povinnost odpovědět. Na horním okraji okna v levém rohu má možnost žák sledovat, u kolikáté otázky z celkového počtu se momentálně nachází. Napravo od tohoto údaje pak vidí, za kolik bodů je právě otevřená otázka. Pod těmito údaji nalezne žák text otázky, která byla položena. Pokud je text doplněn o obrázek, zobrazí se žákovi jeho náhled. Po kliknutí na tento náhled může žák zobrazit obrázek v jeho plné velikosti. V další části okna se již nachází čtyři možnosti odpovědi na zobrazenou otázku. U každé odpovědi může být text, obrázek nebo jejich kombinace. Pokud je to poslední možnost, bude textové pole odpovědi po kliknutí na něj rozšířeno. Žák má opět (stejně jako u obrázku otázky) možnost zvětšit obrázek po kliknutí myši. Žák je povinen vybrat ke každé otázce odpověď. To provede kliknutím na příslušné kolečko u zvolené odpovědi, které se tak označí. Pokračovat na další otázku může stiskem tlačítka *Další otázka*. Pokud je již u poslední otázky, bude namísto něho zobrazeno tlačítko *Dokončit*. Pokud proběhlo vše v pořádku, bude mu zobrazena další otázka (případně zpráva zda v testu uspěl).

Z vyplňování testu může žák kdykoliv odejít a to stiskem tlačítka s křížkem v pravém horním rohu okna. Žák se může k vyplňování testu kdykoli vrátit a pokračovat od první nevyplněné otázky. V testu je zároveň přítomna hlasová nápověda, k níž má žák přístup po najetí na obrázek uší u pravého kraje obrazovky a následném kliknutí na zobrazený obrázek myši.

- **Vyplněný:** V případě, že test byl již vyplněn, má žák možnost procházet své odpovědi. Mezi jednotlivými otázkami může procházet pomocí tlačítek *Předchozí* a *Další*. Pokud byla jeho odpověď na otázku správná, bude puntík u této odpovědi zbarven zeleně, v opačném případě bude puntík u správné odpovědi podbarven červeně.

Další důležitou funkcí, kterou program disponuje, je minimalizace do systémové lišty. Po minimalizaci hlavního okna programu se v liště operačního systému vedle hodin, objeví ikonka s pastelkami. Pokud je program spuštěný, ať už minimalizovaný či ne, stále naslouchá serveru, zda přihlášenému žákovi nebyl zadán nový úkol či test. Pokud tomu tak je, žák je o této skutečnosti informován vyskakovací zprávou u ikonky pastelky v liště. Pokud na tuto zprávu klikne, vrátí se do hlavního okna programu. Žák tak může bez problému pracovat na počítači a o jakémkoliv novém zadání bude okamžitě informován. Tuto funkci může žák vypnout a to tak, že v kontextové menu pod položkou *Program*, zaškrtnutím položku *Zastavit kontrolu*.



Obrázek 18. Okno s vyplněním testu

7. Reálný provoz

7.1. Testovací subjekty

Mezi testovací subjekty v rolích žáků, patřili žáci první třídy základní školy. Testování proběhlo v počítačové učebně základní školy. Po zjištění aktuální počítačové gramotnosti žáků byly svěřeni do péče vyučujícího. Ten jim nejprve vysvětlil, jak program spustit a poté se do něj přihlásit. To také následně názorně převedl na svém počítači připojeném na projektor. Žákům se to ve větší míře povedlo, v případě překlepů byly schopni sami přečíst a porozumět zobrazené chybové hlášce. Největším problémem při přihlášení pak bylo vymyšlení nového hesla. Proto pro následné ostré nasazení programu do provozu bude vyučujícím doporučeno, aby první přihlášení s žákem první třídy provedl buď osobně, nebo tuto povinnost přenechal s vysvětlením jeho rodičům. Po úspěšném přihlášení a další krátké ukázce od vyučujícího byly žáci schopni program bez větších problémů ovládat.

V roli vyučujícího zde figurovala třídní učitelka první třídy. Po krátkém vysvětlení, jak se má přihlásit, byla již sama schopna bez jakýchkoliv problémů program ovládat. Z mé strany vyučující nepotřebovala žádné další informace a všechny funkce programu dokázala ovládat zcela intuitivně.

Testování tak proběhlo úspěšně a nebylo zapotřebí program do větší míry

upravovat. Škola nadále souhlasila s možným budoucím využitím testované aplikace v ostrém provozu.

7.2. Připomínky

Od žáků byly připomínky, až na první přihlášení, pouze kladné. Největším pozitivem zde byla určitě tvorba hlasové nápovědy, která je vytvořeno uživatelsky velmi přívětivou formou a žáky zaujala. Problém s prvním přihlášením a následnou změnou hesla bohužel není možné nijak ze strany žáků řešit, neboť se jedná o podstatný bezpečnostní prvek který nemůže být vypuštěn. O tomto problému a jeho možném řešení tak bude informován vyučující při krátké instruktáži k programu.

Vyučující nenašel žádný problém, který by program obsahoval. Pochválena byla samotná myšlenka programu a jeho přínos do výuky. Plusem byla také označena jednoduchost programu a všude přítomná textová nápověda. Kladně ohodnocena pak byla také přítomnost tlačítek pro zadávání aritmetických znamének.

8. Plány do budoucna

V budoucnu bych rád program dostal do ostrého provozu. Před touto fází bude zajisté potřeba ještě několik měsíců testování. Proto již nyní domlouvám další spolupráci s již použitými testovacími subjekty. Dále by také program mohl sloužit jako jednoduchá komunikační síť. V první vlně mezi vyučujícími, v případě zájmu také mezi žáky. Síť by měla sloužit k tomu, aby si vyučující sdíleli informace o práci s programem a o výsledcích jednotlivých žáků.

V budoucnu by byla možná také konverze klientské části aplikace do mobilní platformy a její optimalizování pro dotykové ovládání. Žáci by tak mohli své testy vyplnit například na tabletech či jiných mobilních zařízeních.

Závěr

Hlavním cílem práce bylo vyvinout program, který splní nejdůležitější cíl – jednoduchost a zároveň svými funkcemi pomůže ve větší míře ve výuce jak vyučujícím, tak i žákům. Vznikl tak program, jenž umožňuje vyučujícím jednoduchou formou vytvářet testy a úkoly, které mohou přiřadit třídám či konkrétním žákům. Z důvodu soběstačnosti si pak vyučující může sám spravovat všechny žáky na jeho škole, i třídy, ve kterých se nacházejí. V rámci tvorby testu může k jednotlivým otázkám a odpovědím přiřadit pro zjednodušení obrázky. Pokud vyučující chce zadat žákům například slohovou práci, může využít funkci úkolů, na které mají žáci možnost odpovídat připojením jakéhokoliv souboru.

Hlavní přínos pro žáka vidím v tom, že stačí, když po startu systému program spustí a následně se do něj přihlásí. Program již pak pravidelně kontroluje, zda přihlášený žák nemá od vyučujícího nové zadání testu či úkolu a pokud nějaké nalezne, bude o tom okamžitě, až do přečtení této zprávy, žáka formou vyskakovacího okna informovat. Žák má možnost se do systému přihlásit jak v hodině informatiky ve své škole, tak i v rámci plnění školních povinností u sebe doma. Vázán je tak pouze platformou Windows.

Program je funkční a splňuje všechny cíle, které byly požadovány při počátcích jeho tvorby. V průběhu tvorby programu se nenaskytly žádné větší problémy.

Conclusions

Main goal of my work was to develop a program, which will fulfil the most important objective – simplicity and also have functions that will provide extra help to teachers and students during classes. Program was created, which enables teachers in an easy way to make tests and exercises, which they can assign to whole classes or specific students. For the reason of self-sufficiency, a teacher can by oneself administer all of his students at school, and also classes, which are they attending. While creating a test, there can be inserted a picture to the question or answer for achieving bigger simplicity. If a teacher wants to assign for example a stylistic exercise, he can use an exercise function and students can attach any kind of file.

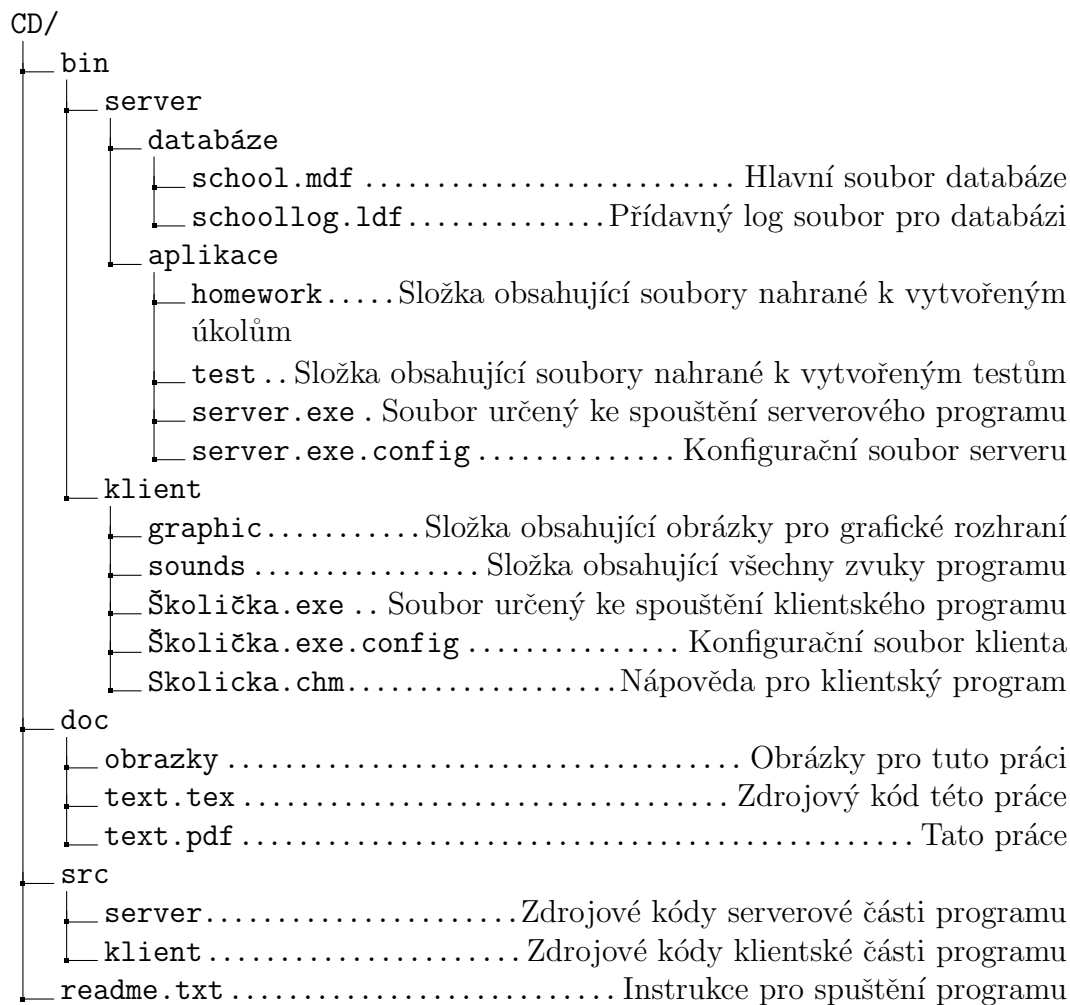
I see the main benefit for the student in the fact, that after start of the system he only has to turn on the program and log in once. Program then regularly controls if the student has a new assignment (test or exercise) and if it finds same, then it will put on a pop-up alert window until the message is read. Student has the possibility to log in the program during informatics classes at school or during completing homework at home. Only necessity for program is platform Windows. Program is functional and fulfils every goal, which was required at the beginning of its development. There were not any bigger problems during creation.

Reference

- [1] Sharp, John. *Microsoft Visual C# 2010 krok za krokem* 1. vyd. Brno: Computer Press a.s., 2010. 696 s. ISBN 978-80-251-3147-3.
- [2] Nagel, Christian a kol. *C# Programujeme profesionálně* 1. vyd. Brno: Computer Press a.s., 2009. 1160 s. ISBN 978-80-251-2401-7.
- [3] Hotek, Mike. *Microsoft SQL Server 2008 krok za krokem* 1. vyd. Brno: Computer Press a.s., 2009. 488 s. ISBN 978-80-251-2466-6.

A. Obsah přiloženého CD

Přiložené CD má tuto strukturu:



B. Hodnocení uživatele

Byla jsem požádána, abych se vyjádřila k výukovému programu pro základní školy „Školička“. S tímto programem jsem měla možnost pracovat na 1.stupni ZŠ Karla st. ze Žerotína Bludov. Využívala jsem jej k tvorbě testů pro žáky první a třetí třídy. Jelikož mám s několika obdobnými programy zkušenost, velmi jsem ocenila několik důležitých věcí.

Největší plus „Školičky“ vidím v její jednoduchosti. Vkusný a jednoduchý design přispívá k rychlé orientaci na ploše. Neztráceli jsme se v množství textu, tvoření testů i ovládání programu je jednoduché, intuitivní. V případě nejistoty obsahoval program kvalitní nápovědu. Mluvenou nápovědu rády využívaly zejména děti prvního ročníku, a to nejen kvůli nejistotě při práci s programem, ale i pro sympatickou „mluvčí“. Mluvená forma nápovědy je velmi vhodná rovněž při práci dětí s vývojovými poruchami učení.

Já i žáci jsme velmi ocenili možnost využívání obrázků, čímž se testy stávají pro žáky nejen názornější, ale i přitažlivější. Díky možnosti vložení obrázků je tudíž možno program využít i pro děti s elementárními znalostmi čtení.

K učitelům velmi „přívětivá“ je možnost vložení termínu, do kterého je nutno test vyplnit a rychlý přehled, kdo zadaný úkol nevypracoval. Testy utvořené ve „Školičce“ tedy mohou být i domácími úkoly nebo fungovat jako úlohy na procvičení pouze některým vybraným žákům. Velmi jsem uvítala rovněž funkci zobrazení počtu získaných bodů a možnost prohlédnout si, ve kterých otázkách žák chyboval.

Jako velmi užitečná se mi jeví možnost přiložení souboru, ať už při práci ve škole či z domova. Výukový program „Školička“ zcela jistě pomáhá pedagogům při výuce, jednoduchou formou umožňuje prověřování znalostí žáků, upevňuje jejich dovednost čtení s porozuměním a rozvíjí počítačovou gramotnost.

Mgr. Marie Matysová