

# Mobilní výuková aplikace matematiky pro základní a střední školy

Bc. Jan Hruban

---

Diplomová práce  
2017



Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky

---

Univerzita Tomáše Bati ve Zlíně  
Fakulta aplikované informatiky  
akademický rok: 2016/2017

## ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Bc. Jan Hruban**  
Osobní číslo: **A15418**  
Studijní program: **N3902 Inženýrská informatika**  
Studijní obor: **Informační technologie**  
Forma studia: **kombinovaná**

Téma práce: **Mobilní výuková aplikace matematiky pro základní a střední školy**

Téma anglicky: **Mobile Learning Application of Mathematics for Elementary and Secondary Schools**

Zásady pro vypracování:

1. Vypracujte rešerži na dané téma.
2. Seznamte se s výukovými plány matematiky pro základní a střední školy.
3. Navrhněte mobilní výukovou aplikaci.
4. Naprogramujte základní výukové, graficky realizované pomůcky pro snadnější porozumění matematickým operacím.
5. Realizujte rozhraní testů, správu výsledků z jednotlivých testů a správu uživatelských účtů v aplikaci.
6. Vytvořte uživatelskou dokumentaci a aplikaci otestujte.

Rozsah diplomové práce:

Rozsah příloh:

Forma zpracování diplomové práce: **tištěná/elektronická**

Seznam odborné literatury:

1. LACKO, L'uboslav. Vývoj aplikací pro Android. 1. vyd. Brno: Computer Press, 2015, 472 s. ISBN 978-80-251-4347-6.
2. PETZOLD, Charles. Creating Mobile Apps with Xamarin.Forms. 2015. Redmond, Washington: Microsoft Press, 2015. ISBN 978-0-7356-9723-2.
3. ALESSI, Patrick. Vývoj her pro iPhone a iPad: programování pro iOS. Vyd. 1. Brno: Zoner Press, 2012, 423 s. Encyklopedie Zoner Press. ISBN 978-80-7413-199-8.
4. NASH, Trey. C# 2010: rychlý průvodce novinkami a nejlepšími postupy. Vyd. 1. Brno: Computer Press, 2010, 624 s. ISBN 978-80-251-3034-6.
5. VOŠICKÝ, Zdeněk. Matematika v kostce: pro střední školy. 1. vyd. Havlíčkův Brod: Fragment, 2007, 195 s. Maturita v kostce. ISBN 978-80-253-0191-3.
6. EISLER, Jaroslav. Matematika v kostce: pro základní školy a nižší ročníky víceletých gymnázií. 1. vyd. Havlíčkův Brod: Fragment, 2006, 120 s. V kostce (Fragment). ISBN 80-253-0197-4.

Vedoucí diplomové práce:

**Ing. Dalibor Slovák, Ph.D.**

Ústav počítačových a komunikačních systémů

Datum zadání diplomové práce:

**28. července 2017**

Termín odevzdání diplomové práce:

**29. srpna 2017**

Ve Zlíně dne 28. července 2017

doc. Mgr. Milan Adámek, Ph.D.  
*děkan*



prof. Mgr. Roman Jašek, Ph.D.  
*ředitel ústavu*

### **Prohlašuji, že**

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s tím, že licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

### **Prohlašuji,**

- že jsem na diplomové/bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 28. 8. 2017

  
.....  
podpis diplomanta

## **ABSTRAKT**

Tato diplomová práce se zabývá vývojem mobilní aplikace pro výuku matematiky na základních a středních školách. První oddíl teoretické části práce se věnuje obecnému popisu dvou v současnosti nejpoužívanějších mobilních operačních systémů Android a iOS a také teorií vývoje aplikací pro tyto platformy. Druhý oddíl teoretické části obsahuje popis rámcových vzdělávacích programů matematiky pro základní a střední školy. Praktická část této práce se již věnuje popisu vývoje samotné mobilní aplikace a tvorbě uživatelské dokumentace.

Klíčová slova: Mobilní aplikace, chytré telefony, tablety, Android, matematika, programování

## **ABSTRACT**

This thesis deals with development of mobile application for learning math on elementary and secondary schools. First section of theoretical part is devoted to general description of currently two most widely used mobile operating systems Android and iOS and also of application development theory for these two platforms. Second section of theoretical part contains description of Framework education programme of mathematics for elementary school and high school. Practical part of this thesis deals with development of learning mobile application itself and also of user guide creation.

Keywords: Mobile applications, Smartphones, Tablets, Android, Math, Programming

Touto cestou bych chtěl poděkovat mému vedoucímu diplomové práce panu doktorovi Daliboru Slovákovi za jeho trpělivost a cenné rady, svým rodičům a sestře za vytrvalou podporu během studií a také svým kolegům a nadřízeným ze zaměstnání za jejich odbornou podporu při vypracovávání této práce.

Prohlašuji, že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

# OBSAH

ÚVOD.....	10
<b>I TEORETICKÁ ČÁST.....</b>	<b>11</b>
<b>1 VÝVOJ MOBILNÍCH APLIKACÍ.....</b>	<b>12</b>
1.1 EXISTUJÍCÍ ŘEŠENÍ.....	12
1.1.1 iMathematics.....	12
1.1.2 Mathway.....	13
1.1.3 Mathspace.....	13
1.2 OPERAČNÍ SYSTÉM ANDROID.....	14
1.2.1 Historie androidu.....	15
Architektura androidu.....	16
Linuxové jádro.....	18
Libraries.....	18
Android runtime.....	18
Application Framework a Applications.....	19
1.2.2 Android aplikace.....	19
Activity.....	20
Service.....	21
Broadcast receiver.....	22
Content provider.....	22
1.3 OPERAČNÍ SYSTÉM IOS.....	22
1.3.1 Architektura iOS.....	23
Core OS.....	23
Core Services.....	24
<i>Core Foundation Framework</i> .....	24
Media Layer.....	25
Cocoa Touch.....	26
1.3.2 iOS aplikace.....	27
Hlavní smyčka aplikace.....	28
Stavy aplikace.....	29
1.4 MULTIPLATFORMNÍ VÝVOJ MOBILNÍCH APLIKACÍ.....	30
1.4.1 PhoneGap.....	31
1.4.2 Xamarin.....	32
<b>2 VÝUKOVÉ PLÁNY MATEMATIKY PRO ZÁKLADNÍ A STŘEDNÍ ŠKOLY.....</b>	<b>35</b>
2.1 VÝUKOVÝ PLÁN MATEMATIKY PRO ZÁKLADNÍ ŠKOLY.....	35
2.1.1 Čísla a početní operace (Čísla a proměnné).....	35
Očekávané výstupy – 1. Období (3. třída).....	36
Očekávané výstupy – 2. Období (5. třída).....	36
Očekávané výstupy – 3. Období (9. třída).....	36
2.1.2 Závislost, vztahy a práce s daty.....	37
Očekávané výstupy – 1. Období (3. třída).....	37
Očekávané výstupy – 2. Období (5. třída).....	37
Očekávané výstupy – 3. Období (9. třída).....	37
2.1.3 Geometrie v rovině a v prostoru.....	37
Očekávané výstupy – 1. Období (3. třída).....	38

Očekávané výstupy – 2. Období (5. třída) .....	38
Očekávané výstupy – 3. Období (9. třída) .....	38
2.1.4    Nestandardní aplikační úlohy .....	38
Očekávané výstupy – 2. Období (5. třída) .....	39
Očekávané výstupy – 3. Období (9. třída) .....	39
2.2    VÝUKOVÝ PLÁN MATEMATIKA PRO STŘEDNÍ ŠKOLY (GYMNÁZIA) .....	39
Argumentace a ověřování – očekávané výstupy .....	39
Číslo a proměnná – očekávané výstupy .....	40
Práce s daty, kombinatorik, pravděpodobnost .....	40
Závislost a funkční vztahy .....	40
Geometrie .....	41
<b>II PRAKTICKÁ ČÁST .....</b>	<b>42</b>
<b>3    NÁVRH A REALIZACE MOBILNÍ VÝUKOVÉ APLIKACE .....</b>	<b>43</b>
3.1    NÁVRH APLIKACE .....	43
3.2    POUŽITÉ VÝVOJOVÉ NÁSTROJE .....	44
3.2.1    Visual studio Community 2015 .....	44
3.2.2    Microsoft SQL server .....	45
3.2.3    Git, Bitbucket a SourceTree .....	45
3.2.4    Microsoft Azure .....	45
3.2.5    Xamarin .....	45
3.3    BACKEND APLIKACE .....	46
3.3.1    Struktura projektu .....	46
3.3.2    Databáze .....	48
3.3.3    API Dokumentace .....	49
3.4    MOBILNÍ APLIKACE .....	51
3.4.1    Struktura projektu .....	51
3.4.2    Popis aplikace .....	53
3.5    WEBOVÁ APLIKACE .....	53
3.6    GRAFICKY REALIZOVANÉ POMŮCKY PRO POROZUMĚNÍ MATEMATICKÝM OPERACÍM 55	
3.7    ROZHRANÍ TESTŮ A SPRÁVA VÝSLEDKŮ .....	58
3.8    SPRÁVA UŽIVATELSKÝCH ÚČTŮ .....	59
<b>4    UŽIVATELSKÁ DOKUMENTACE .....</b>	<b>61</b>
4.1    UŽIVATELSKÁ DOKUMENTACE PRO STUDENTY A ŽÁKY .....	61
4.1.1    Registrace do aplikace .....	61
Standardní registrace .....	61
Facebook registrace .....	63
4.1.2    Přihlášení do aplikace .....	64
4.1.3    Úvodní obrazovka .....	65
4.1.4    Přednášky .....	67
4.1.5    Cvičení .....	68
4.1.6    Úkoly .....	71
4.1.7    Hry .....	73
4.1.8    Odhlášení z aplikace .....	73
4.2    UŽIVATELSKÁ DOKUMENTACE PRO UČITELE .....	74
4.2.1    Registrace učitele .....	74



Standardní registrace.....	74
Facebook registrace .....	76
4.2.2 Úvodní obrazovka .....	77
4.2.3 Přednášky .....	78
4.2.4 Nový úkol.....	79
4.2.5 Výsledky .....	82
4.2.6 Hry.....	84
4.3 UŽIVATELSKÁ DOKUMENTACE - PŘIHLÁŠENÍ BEZ REGISTRACE.....	84
<b>5 TESTOVÁNÍ APLIKACE .....</b>	<b>85</b>
5.1 TESTOVÁNÍ WEBOVÉ APLIKACE A REST API.....	85
5.2 TESTOVÁNÍ MOBILNÍ APLIKACE.....	86
5.3 VÝSLEDKY TESTOVÁNÍ.....	87
<b>ZÁVĚR .....</b>	<b>88</b>
<b>6 BIBLIOGRAFIE .....</b>	<b>89</b>
<b>SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK.....</b>	<b>93</b>
<b>SEZNAM OBRÁZKŮ .....</b>	<b>94</b>
<b>SEZNAM TABULEK.....</b>	<b>96</b>

## ÚVOD

Svět chytrých mobilních zařízení teprve nedávno vstoupil do druhé dekády své existence a mnoho lidí už by si život bez nich snad ani nedokázalo představit. Tato poměrně krátká, avšak zcela jistě bouřlivá historie chytrých mobilních zařízení je charakteristická neuvěřitelně rychlým vývojem na poli miniaturizace velmi výkonného hardwaru. Bouřlivým vývojem však prošla chytrá zařízení i z pohledu softwaru. V oblasti mobilních operačních systémů jsme tak mohli být svědky vzestupu téměř neznámého startupového projektu až na samotný vrchol i velkého pohoření softwarového giganta v podobě Microsoftu. Hlavní souboj operačních systémů pro mobilní zařízení se nyní odehrává mezi dvěma hráči: společností Google Inc. s jeho otevřeným operačním systémem Android, podporovaným známými výrobci mobilních zařízení jako Samsung či Sony a společností Apple s jeho iOS určeným výhradně pro zařízení od téže společnosti.

Pro vývojáře, kteří chtějí své aplikace poskytnout, co nejširší skupině uživatelů jsou však i dvě platformy hodně. Stát se expertem na vývoj aplikací pro Android i iOS je velice náročné, protože vývoj pro každou z těchto platform je velice specifický. Řešení však poskytují multiplatformní frameworky umožňující vyvíjet aplikace v jednom programovacím jazyce, v jednom vývojovém prostředí a dokonce (alespoň částečně) pod jedním zdrojovým kódem.

Dnešní mobilní aplikace se uplatňují v mnoha různých segmentech každodenního života od zábavy, přes time-management, navigaci až po .... Zajímavá oblast využití mobilní aplikace, která se podle mého názoru, nevyužívá tak, jak by si zasloužila, je oblast vzdělávání. Mobilní aplikace jsou díky své vysoké schopnosti interakce jako stvořené pro výuku zábavnou formou a tvorba zábavně naučných aplikací zase představuje opravdovou výzvu pro vývojáře. Využití mobilních technologií ve výuce představuje zajímavé možnosti využití nejen pro studenty, ale i pro učitele. Předávání znalostí ve formě zápisků, zadávání a kontrola domácích úkolů či testů by mohla v takovém případě představovat velkou úsporu času a energie vyučujících.

V této diplomové práci se pokusím právě takovou výukovou aplikaci matematiky vytvořit. V teoretické části se zaměřím na teorii vývoje mobilních aplikací a také výuky matematiky. V praktické části se již budu věnovat vývoji samotné aplikace a tvorbě uživatelské dokumentace.

## **I. TEORETICKÁ ČÁST**

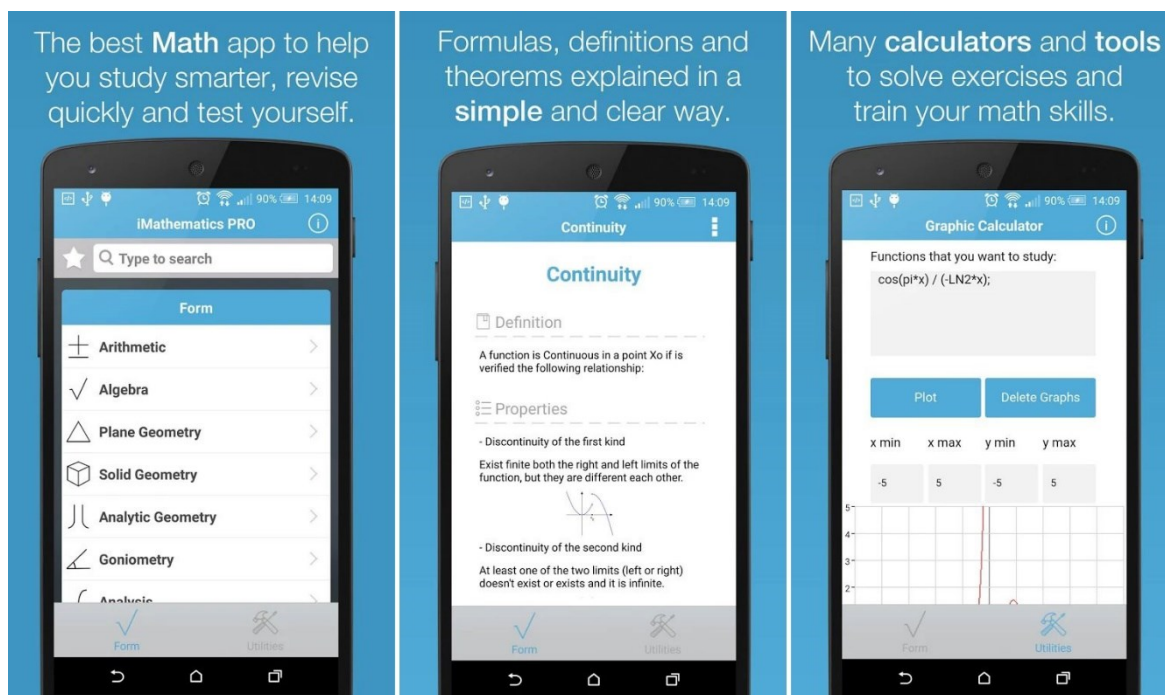
# 1 VÝVOJ MOBILNÍCH APLIKACÍ

## 1.1 Existující řešení

### 1.1.1 iMathematics

Aplikace iMathematics sama sebe uvádí jako „osobní matematický rádce“. Byla vydána v květnu 2009, dohromady má přes čtyři miliony stažení a průměrné hodnocení na AppStore a Google play se pohybuje okolo 3,5/5. Aplikace je dostupná pro operační systémy iOS, Android a Windows Phone. [1]

V základní verzi tato aplikace obsahuje přes 70 matematických témat, umožňuje provádět matematická cvičení s různými interaktivními kalkulátory a umožňuje rychlé vyhledávání řešené problematiky. Cena placené verze aplikace se pohybuje okolo 80 korun a nabízí odemčení dalších matematických témat, interaktivní kvízy a nástroje pro lepší pochopení matematických témat jako například řešení krok po kroku nebo grafické kalkulátory. [2]

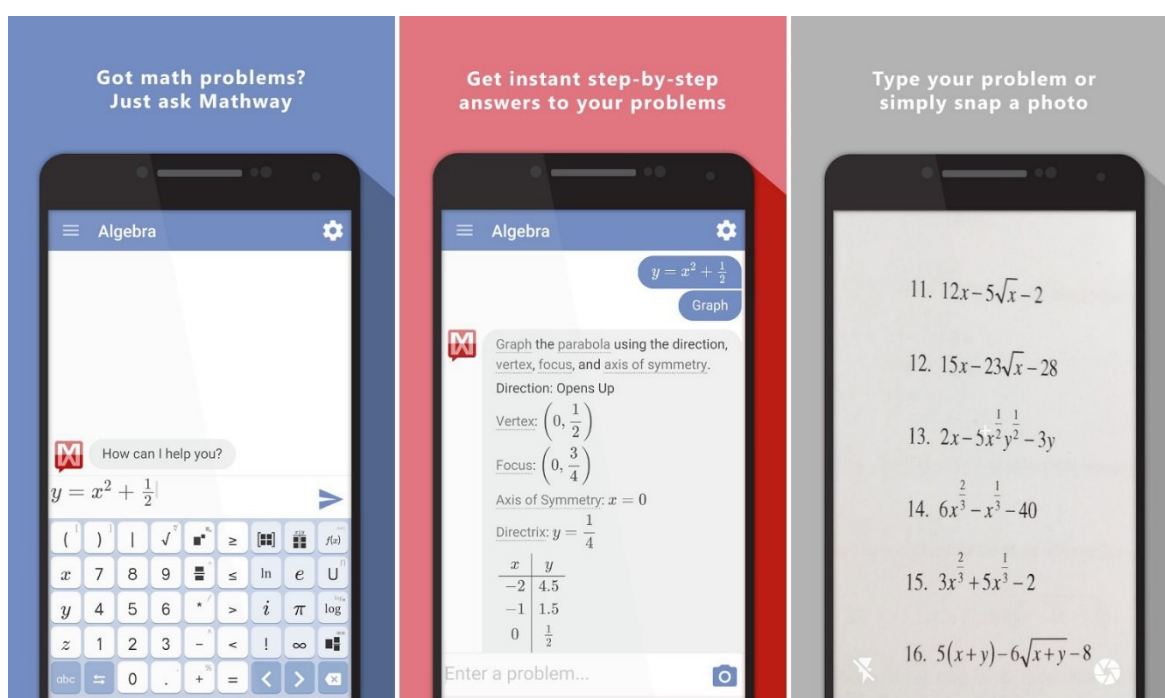


Obr. 1. Ukázka aplikace iMathematics. [1]

Web commonsense.org, který se mimo-jiné zabývá hodnocením výukových aplikací, ohodnotil iMathematics průměrnou známost 3/5. Jako hlavní výhody vidí množství informací a příkladů a využitím interaktivních prvků. Nevýhody vidí v nízké zpětné vazbě, gramatické korekci a malé pestrosti. [3]

### 1.1.2 Mathway

Aplikace Mathway je určena pro operační systémy iOS, Android a je dostupná i jako webová aplikace. Na obou mobilních platformách má aplikace přes milion stažení. A hodnocení se pohybuje na velice slušných 4-4,5/5. Aplikace je řešena formou tzv. „Problem Solver“, kde uživatel napíše nebo vloží obrázek formulace řešeného problému a aplikace se pokusí vrátit výsledek. Aplikace navíc umožňuje procházení řešeného problému krok po kroku. Aplikace taktéž umožňuje práci s grafy. Graficky je aplikace podobná známé chatovací aplikaci Messenger. [4]



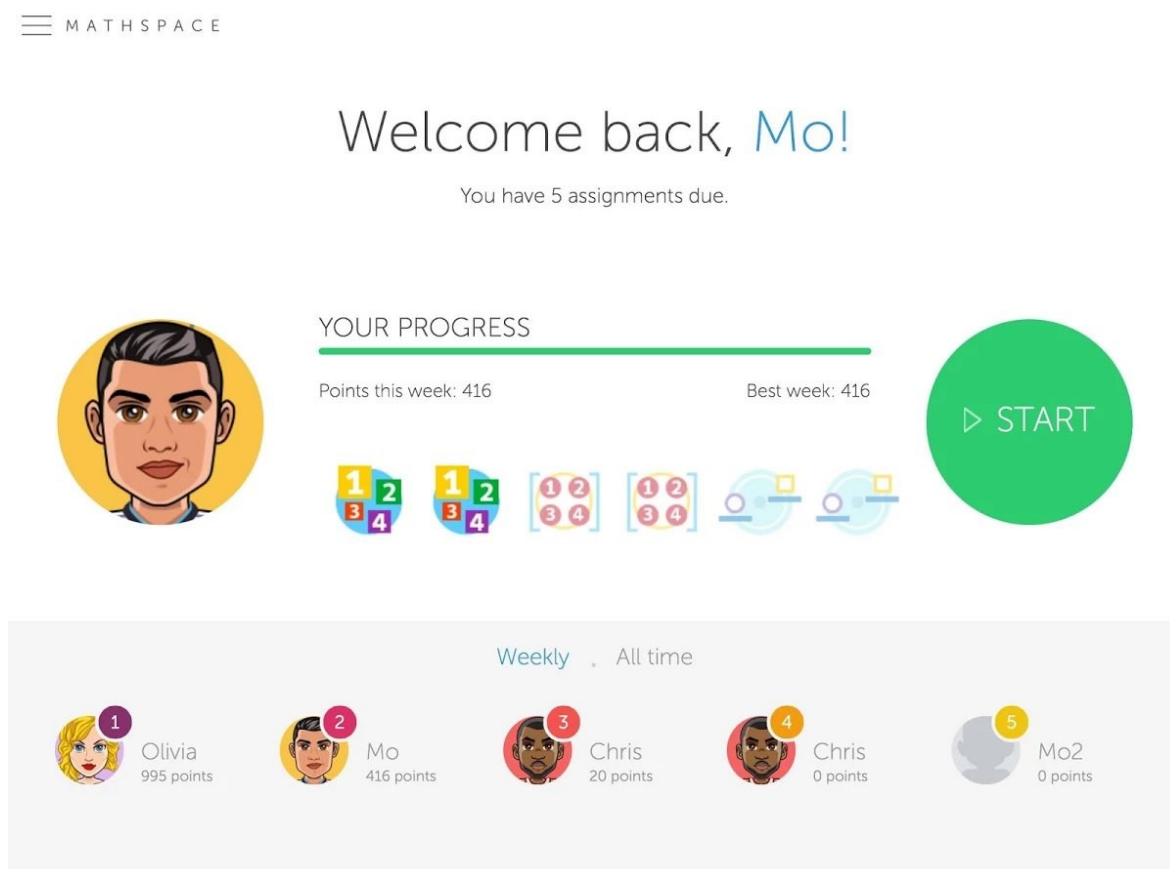
Obr. 2. Ukázka aplikace Mathway [4]

Recenze aplikace oceňují především jednoduché a hezké grafické rozhraní, jako zápory vidí v uzamčení užitečných výukových funkcí do placených přídatků. Osobně mi v aplikaci chybí možnost cvičných příkladů se zpětnou vazbou studentovi nebo učiteli. [5]

### 1.1.3 Mathspace

Mobilní aplikace Mathspace funguje jako učebnice, pracovní sešit i poznámkový blok. Umožňuje studentům řešit matematické problémy online s okamžitou zpětnou vazbou a pomocí v každém kroku řešeného problému. Obsahuje databázi s více jak 20000 interaktivními otázkami pokrývající algebru, geometrii, grafy, pravděpodobnost a statistiku. Obsahuje video ukázky řešených problémů a zpětnou vazbu o progresu studentů pro učitele. Aplikace je

dostupná pro mobilní platformy iOS, Android a pro platformu Windows 10. Počet stažení aplikace se pohybuje v desetitisících a průměrné hodnocení se pohybuje mezi 3,2-3,5/5. [6]



Obr. 3. Ukázka aplikace MathSpace [6]

Recenze oceňují na aplikaci úroveň individualizace a real-time podporu, které se blíží výuce formou jeden žák-jeden učitel. Studenti sdílejí svoji práci, dostávají zpětnou vazbu a mohou se učit vlastním tempem. Záporem aplikace je především zpoplatnění některých funkcí, především pro učitele. [7]

## 1.2 Operační systém Android

Operační systém android v současné době pohání stovky miliónů mobilních zařízení ve více než 190 zemích světa. Jedná se o největší mobilní platformu s rychlým růstem nových uživatelů – každý den přibude kolem milionu poprvé spuštěných android zařízení. [8]

OS Android je Open Source platforma tzn. platforma s otevřeným zdrojovým kódem a jedná se o poměrně bezpečné prostředí inspirované některými základními zabezpečovacími mechanismy Linuxu. Jedním z těchto mechanismů je takzvaný Sandbox. Ten zajišťuje izolovaný běh aplikací pomocí přidělování UID, kde každá aplikace běží v separátním procesu a

je jim přidělena vlastní část souborového systému. Tím je zajištěno, že aplikace nemohou přistupovat k údajům jiné aplikace, pokud k tomu nedostala povolení uživatele. [9]

### 1.2.1 Historie androidu

V říjnu roku 2003 byla založena startup společnost Android Inc. specializující se na vývoj software pro mobilní telefony. V červenci roku 2005 byl tento startup odkoupen firmou Google Inc. s cílem posílit svůj podíl na trhu mobilních technologií. [10]

V roce 2007 bylo založeno uskupení firem zaměřující se na vývoj softwaru a hardwaru pro mobilní zařízení s názvem Open Handset Alliance. V tomto uskupení byly zastoupeny firmy jako Google Inc., T-Mobile, HTC, Qualcomm, Motorola a další. V listopadu 2007 Open Handset Alliance oficiálně oznámilo vývoj OS Android, prvního otevřeného a rozsáhlého operačního systému pro mobilní zařízení. [11]

V září roku 2008 T-Mobile a Google představily první chytré zařízení, poháněné operačním systémem Android ve verzi 1.0 – T-Mobile G1 (známý též jako HTC Dream). Od verze 1.0 jsou pravidelně vydávány aktualizace v podobě nových verzí. Zajímavostí je systém pojmenovávání těchto aktualizací – každá významnější aktualizace je pojmenována po sladkosti a tyto názvy se drží abecedního pořadí. [12]

Tab. 1. Historie verzí Androidu.

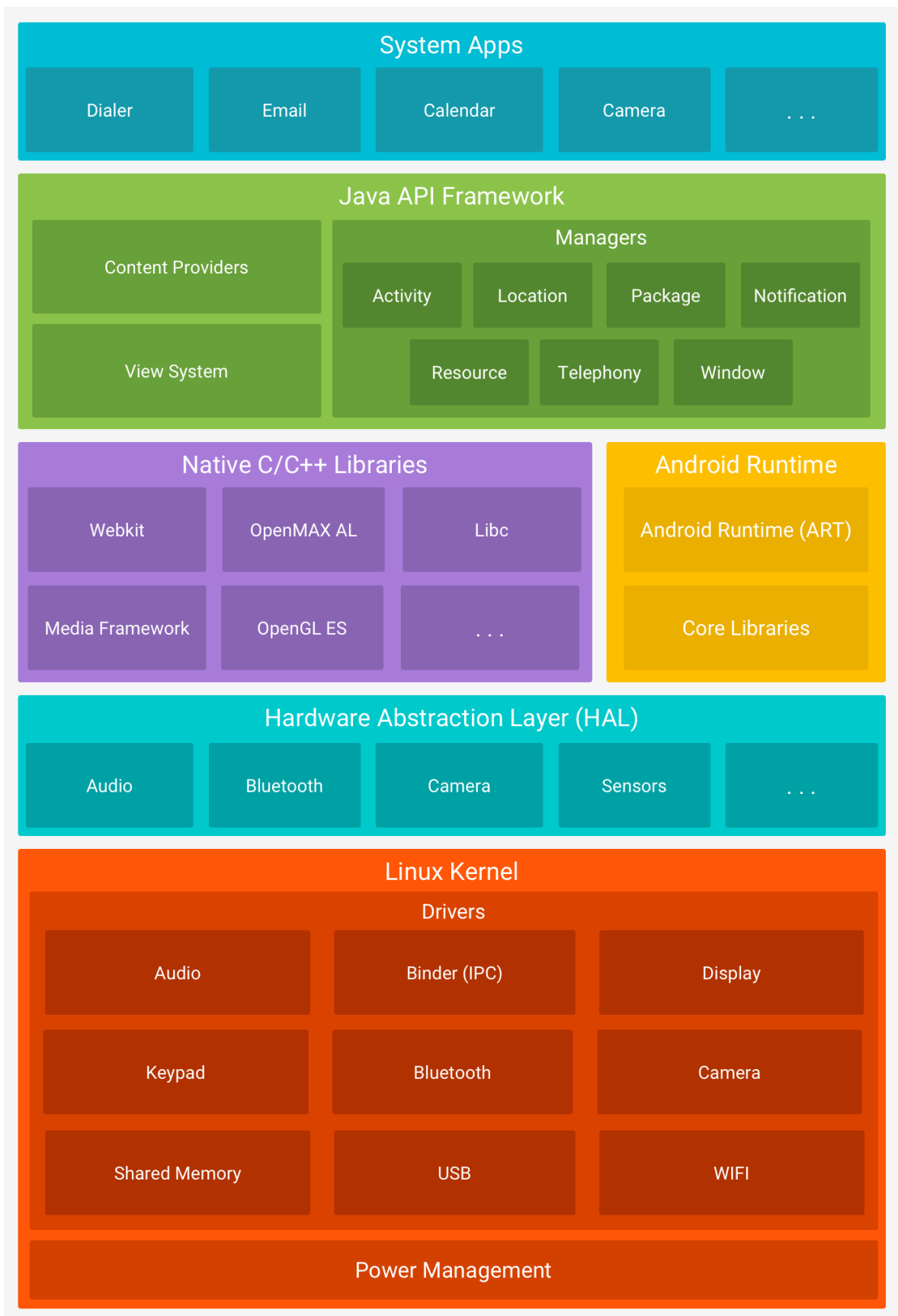
Verze	Název	Rok vydání	Novinky
1.5	Cupcake	2009	Virtuální klávesnice třetích stran, rozšířena podpora video formátů, kopírování do schránky, zlepšení uživatelského rozhraní
1.6	Donut	2009	Vylepšení obchodu s aplikacemi, integrované vyhledávání Google, vylepšené hlasové vyhledávání, podpora displejů s vyšším rozlišením
2.0/2.1	Eclair	2009	Vylepšené uživatelské prostředí, optimalizace výkonu, Bluetooth 2.1, Google Maps 3.x, živé tapety

2.2	Froyo	2010	Nové uživatelské prostředí, možnost instalovat aplikace na paměťové karty, vylepšené zálohování, možnosti Wi-Fi hotspotu, optimalizace výkonu
2.3	Gingerbread	2010	Vylepšena správa napájení, podpora více fotoaparátů, NFC, podpora nových typů senzorů (gyroskop, barometr), podpora internetových hovorů
3.0	Honeycomb	2011	První verze pro tablety, vylepšené uživatelské rozhraní, Action bar, vylepšení notifikací a domovské obrazovky.
4.0	IceCream Sandwich	2011	Tato verze sjednocuje verze pro telefony a tablety, widgety na zamykací obrazovce, adresáře, převod hlasu na text, nový internetový prohlížeč
4.2	Jelly Bean	2012	Google Cloud Messaging, uživatelské účty, Google Now
4.4	KitKat	2013	Optimalizace výkonu, rychlejší multitasking, inteligentní správa procesů, infračervené ovládání
5.0	Lollipop	2014	Optimalizace výkonu, nový engine ART, nové uživatelské rozhraní
6.0	Marshmallow	2015	Skupiny oprávnění pro aplikace, optimalizace výdrže, podpora USB-C, rozpoznávání otisků prstů
7.0	Nougat	2016	Práce s více okny, API pro virtuální realitu, úprava notifikací

### Architektura androidu

OS Android je založen na Linuxovém jádře a skládá se celkem z pěti hlavních vrstev. Každá z vrstev vystupuje z pohledu systému víceméně samostatně a provádí různé operace. V praxi však dochází ke spolupráci jednotlivých vrstev mezi sebou a ty tak nejsou mezi sebou striktně odděleny. [13]





Obr. 4. Architektura OS Android [14]

### *Linuxové jádro*

Linuxové jádro nebo Linux Kernel představuje první a nejnižší vrstvu operačního systému. Úkolem této vrstvy je implementace abstrakce mezi hardwarem zařízení a softwarem na vyšších vrstvách. Při spuštění zařízení je linuxové jádro zavedeno do operační paměti a je mu předáno řízení systému. To zahrnuje především koordinaci činností běžících procesů, správa paměti, správa sítí atd. Na úrovni jádra jsou též implementovány zabezpečení systému, správa napájení nebo základní grafika. [13] [9]

### *Libraries*

Další vrstvou je vrstva knihoven – libraries. Jedná se o nativní knihovny psané v jazyce C/C++, které poskytují přístup vyšším vrstvám k různým komponentám Androidu. Jsou to například tyto knihovny [6]

- **android.util** – knihovna obsahující nízko-úrovňové třídy – řetězcové formátování, nástroje pro parsování XML a další
- **android.os** – knihovna pro přístup k základním službám operačního systému - interprocesní komunikace, hodiny debugging a další.
- **android.graphics** – knihovna pro práci a vykreslování grafických prvků
- **android.telephony** – knihovna pro interakci s komunikačním hardwarem zařízení – telefonní hovory, SMS zprávy
- **android.webkit** – knihovna pro práci s webovým obsahem
- **android.media** – knihovna pro práci s multimediálním obsahem
- **android.location** – knihovna, která zabezpečuje komunikaci s GPS modulem zařízení
- **SQLite** – knihovna pro práci s odlehčenou verzí SQL relační databáze typu SQLite
- **SSL** – knihovna pro práci s šifrovacím protokolem pro bezpečnou komunikaci v rámci protokolu TCP/IP [13]

### *Android runtime*

Android aplikace, podobně jako obyčejné aplikace vyvíjené prostřednictvím jazyka Java se překládají do spustitelné podoby prostřednictvím virtuálních strojů tzv. Virtual Machine. Do verze 4.4. vrstva Android runtime obsahovala tzv. DVM (Dalvik Virtual Machine), jednalo se o virtuální stroj, který byl vyvíjen speciálně pro Android společností Google s přihlédnu-

tím ke specifickým potřebám mobilních zařízení (omezené možnosti napájení, slabší hardware). DVM využíval základní vlastnosti linuxového, např. správu paměti nebo práce s vlákny. Od verze 5 je Dalvik nahrazen ART (Android Runtime), mezi hlavní výhodu ART oproti Dalviku patří použití AOT (Ahead-Of-Time) kompilátoru oproti JIT (Just-In-Time) kompilátoru Dalviku. To znamená, že jednotlivé aplikace jsou z byte kódu do strojového kódu překládány při instalaci aplikace na místo při každém spuštění, což znamenalo zrychlení spuštění aplikací. Vrstva Android Runtime též obsahuje základní knihovny programovacího jazyka Java. S drobnými rozdíly, především v oblasti tvorby uživatelského rozhraní jsou knihovny stejné jako v Java SE (Standard Edition) [13]

### *Application Framework a Applications*

S pohledu vývojáře je vrstva Application Framework nejzajímavější. Právě tato vrstva umožňuje vývojáři přistupovat k nejrůznějším službám, které může využívat ve svých aplikacích např. prvky uživatelského rozhraní, používání hardwaru zařízení, spouštět aplikace na pozadí atd. Skládá se z několika základních komponent: [9]

- **Package manager** – modul pro správu balíčků. Jedná se o databázi všech aplikací nainstalovaných na zařízení.
- **Windows manager** – tento modul spravuje okna, které tvoří aplikace. Například hlavní okno aplikace, dialogová okna atd.
- **View system** – modul pro správu široké palety prvků grafického uživatelského rozhraní např. tlačítka, formulářové vstupy a další.
- **Activity manager** – modul pro správu životního cyklu aplikací.
- **Notification manager** – modul pro správu upozornění
- **Content providers** – umožňuje jedné aplikaci přistupovat a pracovat s obsahem jiných aplikací [9]

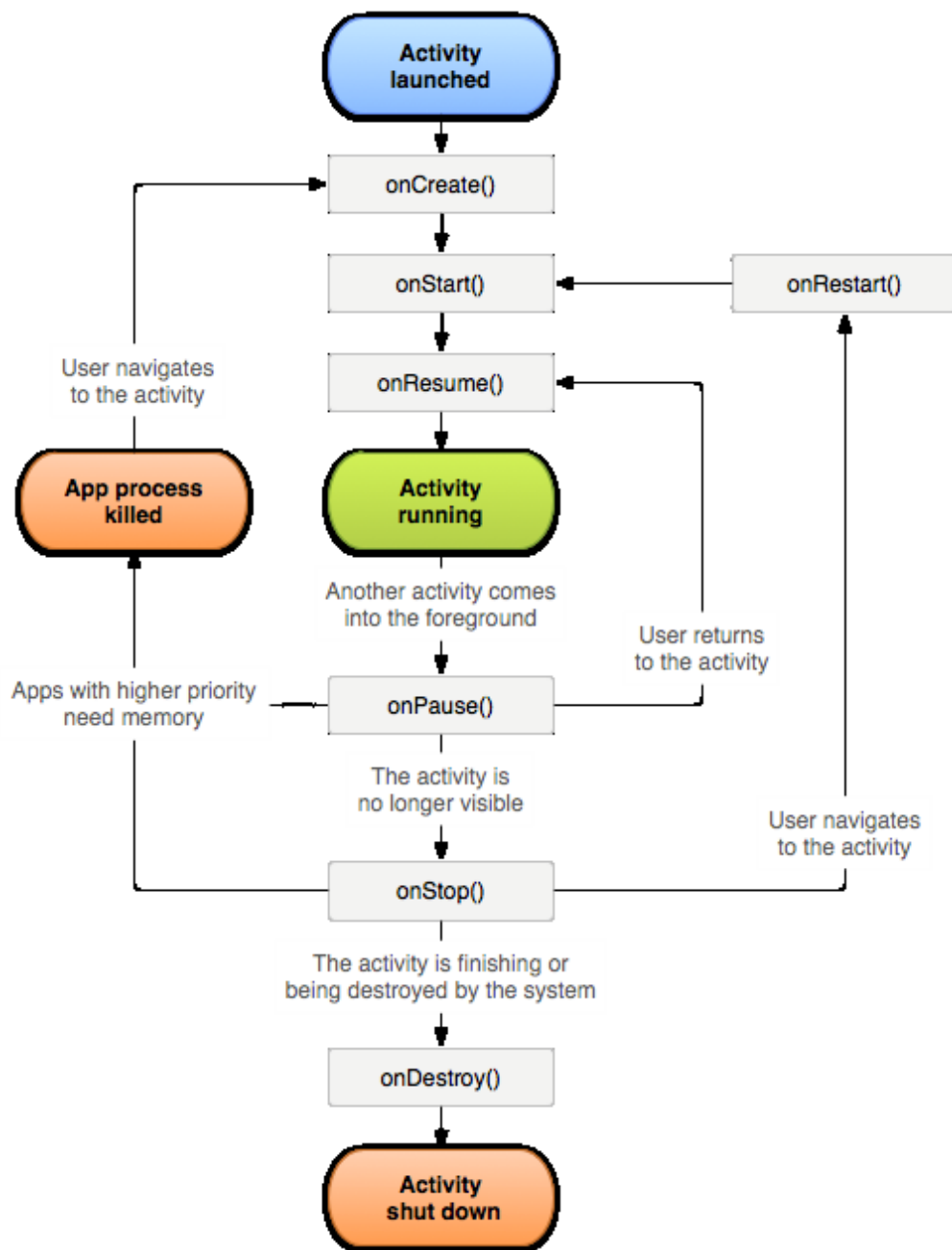
Vrstva Applications již představuje samotné aplikace nainstalované na zařízení. [13]

### **1.2.2 Android aplikace**

Android aplikace jsou psané v jazyce Java a skládají ze čtyř základních komponent: Activity, Service, Broadcast Receiver a Content provider.

### Activity

Activity nebo Aktivita je nejdůležitější část aplikace. Aktivita má vlastní životní cyklus a umožňuje uživateli prostřednictvím grafického uživatelského rozhraní (GUI) přijímat informace od aplikace a ovládat ji. V rámci této třídy je naprogramována největší část samotné aplikace – umožňuje zachytávat události vyvolané uživatelem (například stisknutí tlačítka) a patřičně na ně reagovat. Dále se stará například o správu fragmentů – menších částí Activity s vlastním životním cyklem. [9]

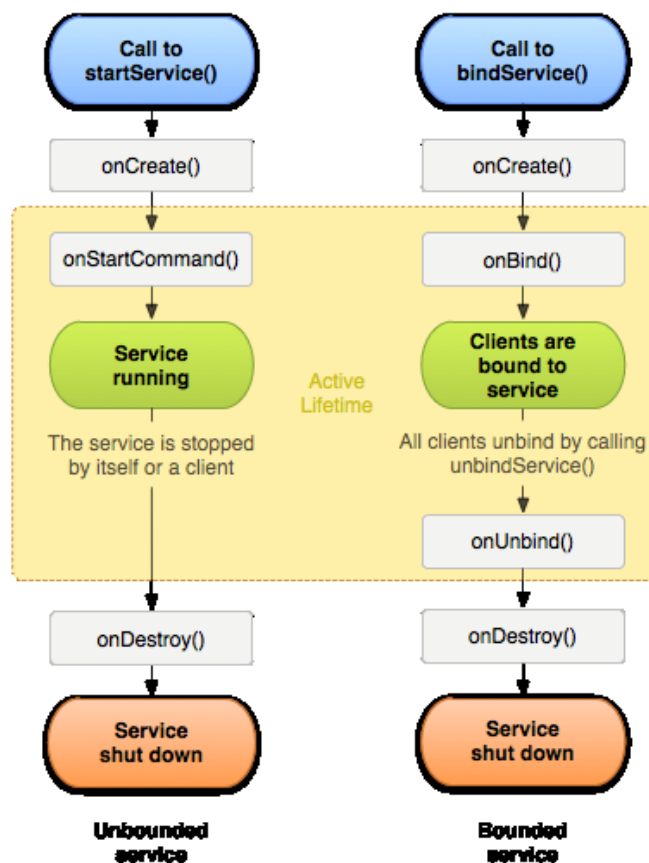


Obr. 5. Životní cyklus aktivity [15]

## Service

Service neboli Služba slouží k vykonávání dlouhotrvajících operací na pozadí a nemá uživatelské rozhraní. Služby je možné spouštět pomocí jiných aplikačních komponent a ty pak mohou běžet na pozadí i při používání jiné aplikace uživatelem. Typickým příkladem služby je přehrávání hudby. Služby mohou být trojího typu:

- Naplánované – tento typ služby je dostupný až od verze androidu 5.0 a využívá tzv. `JobScheduler`, ten umožňuje naplánovat spuštění služby na přesný čas.
- Spouštěné – spouštějí se okamžitě a běží i po ukončení aplikace tak dlouho, dokud nejsou ukončeny samy sebou nebo systémem ve výjimečných případech. Používá se například pro stahování souborů přes internet.
- Vázané – umožňují komunikovat s jinými komponentami aplikací formou client-server struktury, posílat požadavky a přijímat výsledky a to dokonce mezi samotnými procesy pomocí mezi-procesové komunikace (IPC). Vázané služby existují tak dlouho, dokud je na ně navázána nějaká komponenta. Pokud tato vazba zanikne, zaniká i samotná služba. [16]



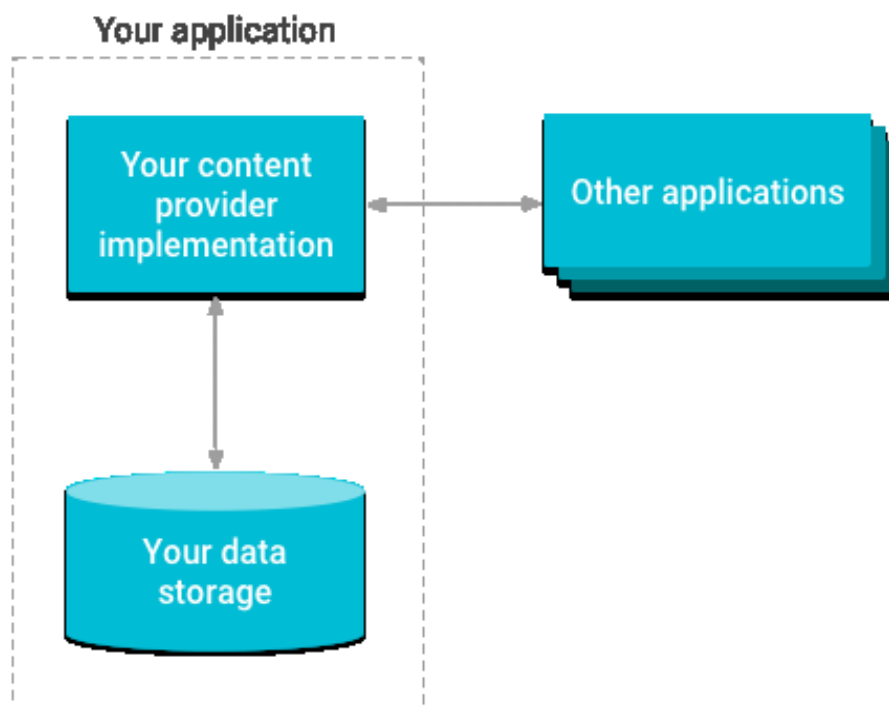
Obr. 6. Životní cyklus služby [16]

### **Broadcast receiver**

Android aplikace mohou posílat a přijímat tzv. broadcast zprávy od systému nebo od jiných aplikací. Tyto zprávy jsou odesílány při zvláštních událostech, jako například navázání internetového spojení nebo informace že začalo nabíjení zařízení. Aplikace též mohou vyslat zprávu například o stažení nových dat, které by mohly být zajímavé pro jiné aplikace.

### **Content provider**

Content providery pomáhají aplikaci spravovat přístup k vlastním datům, datům jiných aplikací a nabízejí možnosti sdílení dat s jinými aplikacemi. Content providery jsou standardní rozhraní, které propojují data v jednom procesu s kódem běžícím v procesu jiném. Představují tak jakýsi most, přes které je možné sdílet data mezi aplikacemi, které za standardních podmínek běží každá ve svém procesu a k datům jiných aplikací nemají přímý přístup. [17]



Obr. 7. Ukázka Content Provideru a správy přístupu k datům aplikace [17]

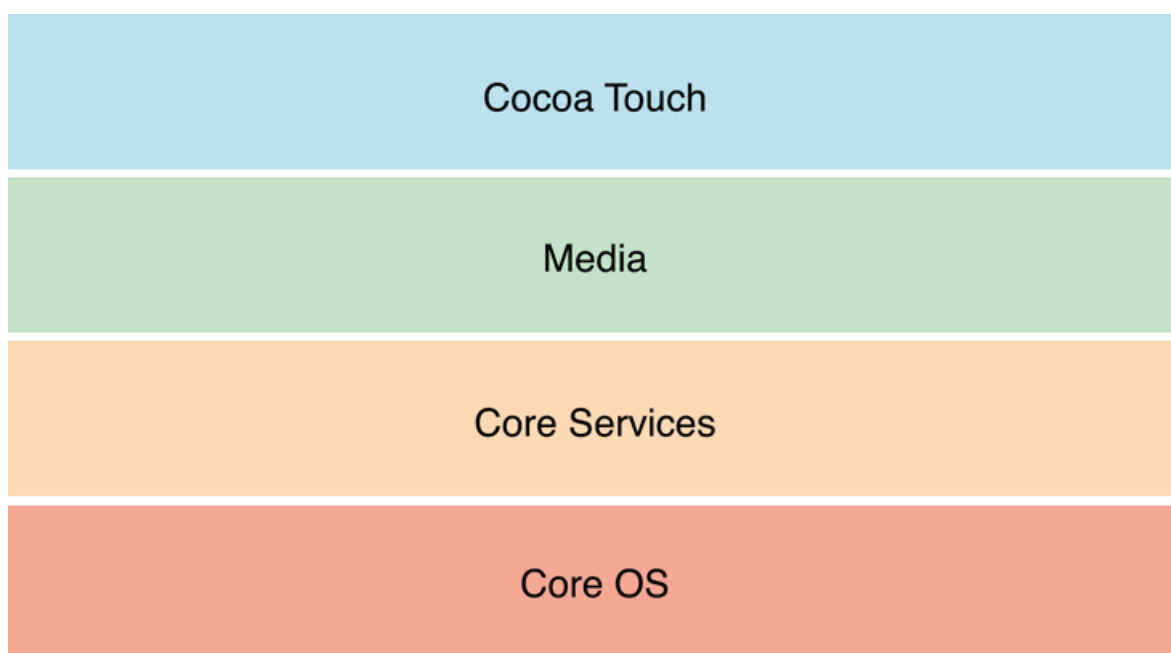
## **1.3 Operační systém iOS**

iOS je operační systém od společnosti Apple Inc., který pohání dotyková zařízení iPhone, iPad a iPod od stejné společnosti. iOS slouží ke správě hardwaru těchto zařízení a poskytuje technologie nezbytné pro vývoj nativních aplikací. Součástí iOS jsou také různé systémové

aplikace jako telefon, email a internetový prohlížeč, poskytující základní servis pro uživatele.

### 1.3.1 Architektura iOS

Aplikace podobně, jako u OS Android nekomunikují s hardwarem přímo, ale prostřednictvím několika vrstev a systémových rozhraní. Tato rozhraní zjednodušují vývoj aplikací, které tak mohou fungovat konzistentně na zařízeních s rozlišeným hardwarem. Nejnižší vrstvy systému obsahují základní služby a technologie a se stoupajícími vrstvami jsou technologie a služby komplexnější a sofistikovanější. [18]



Obr. 8. Architektura systému iOS. [18]

#### **Core OS**

Vrstva Core OS je nejnižší vrstvou a obsahuje nízkourovňové funkce pro práci s hardwarem nebo zabezpečením. Obsahuje jádro systému - Mach kernel, ovladače, a nízkourovňová UNIXová rozhraní. Mach kernel se stará o správu paměti, vlákna, souborový systém, síť a interprocesovou komunikaci. Ovladače poskytují rozhraní mezi dostupným hardwarem zařízení a systémovými frameworky. Z bezpečnostních důvodů je přístup k jádru a ovladačům přístupný pouze systémovým frameworkům a aplikacím. Frameworky v této vrstvě jsou například: [19]

- **Accelerate Framework** – obsahuje rozhraní pro vykonávání digital signal processing (DSP), lineární algebry a výpočty pro zpracování obrázků

- **External Accesory Framework** – poskytuje podporu pro komunikace se zařízeními kompatibilními s iOS, například přes 30-pinový dock nebo bezdrátově přes Bluetooth rozhraní.
- **Generic Security Services Framework** – standardní set bezpečnostních služeb pro iOS aplikace.
- **The Local Authentication Framework** – poskytuje služby pro autentifikaci uživatele prostřednictvím Touch ID (otisku prstu), a pro zabezpečení aplikací autentifikací.
- **Network Extension Framework** – poskytuje podporu pro konfiguraci a ovládání VPN tunelů. [19]

### *Core Services*

Tato vrstva obsahuje základní systémové služby pro aplikace. Klíčovými mezi těmito službami jsou Core Foundation a Foundation Frameworky, které definují základní typy, které jsou využívány všemi aplikacemi. Tato vrstva také obsahuje služby pro práci například s lokacemi, cloudovou službou iCloud, sociálními sítěmi a internetem. [20]

### *Core Foundation Framework*

Obsahuje sety rozhraní psaných v jazyce C, která poskytují základní data management. Jedná se především o:

- Kolekce (pole, listy, sety)
- Nástroje pro formátování textových řetězců
- Nástroje pro formátování data a času
- Správa vláken a smyček
- Komunikace s porty a sokety [20]

### *Foundation Frameworks*

Tento framework představuje Objective-C wrapper pro Core Foundation Framework, to znamená, že umožňuje využívat tato rozhraní pomocí jazyka Objective-C. Navíc obsahuje například:

- Bonjour – propojování kompatibilních zařízení v rámci lokální sítě
- Správa komunikačních portů
- Internacionalizace



- Regulární výrazy
- Podpora Cache [20]

### *Media Layer*

Tato vrstva obsahuje grafické, audio a video technologie a frameworky pro usnadnění nasazení multimediálního obsahu do aplikací. [21]

### *Grafické technologie*

- **UIKit graphics** – definuje vysokoúrovňovou podporu pro kreslení obrázků, Bézierových křivek a animací. Poskytuje třídy pro implementaci kreslení. Představuje rychlou a efektivní cestu pro renderování obrázků a animací.
- **Core graphics** – nativní kreslicí engine pro iOS aplikace. Vhodný pro situace, kdy je potřeba renderovat 2D tvary a obrázky dynamicky
- **Core Animation** – technologie pro optimalizaci animací.
- **Core Images** – poskytuje pokročilou podporu pro manipulaci s videem a statickými obrázky
- **OpenGL ES** - umožňuje pokročilé renderování 2D a 3D grafiky s využitím rozhraní pro akceleraci hardware. Tento Framework je používán hlavně vývojáři her.
- **Metal** – poskytuje pokročilé možnosti grafického renderingu a výpočetních úkolů, pomocí přístupu k GPU řady A7 a novějších.
- **Image I/O** – poskytuje rozhraní pro čtení a zápis většiny obrázkových formátů.
- **Photos Library** – poskytuje přístup k uživatelským fotkám a videím. [21]

### *Audio Technologies*

- **Media Player Framework** - vysokoúrovňový Framework poskytující přístup k iTunes knihovně uživatele a podporu pro přehrávání stop a playlistů.
- **AV Foundation** – Objective-C rozhraní pro správu nahrávání a přehrávání audia a videa.
- **OpenAL** – technologie pro práci s pozičním zvukem
- **Core Audio** – set frameworků, které poskytují jednoduchá i sofistikovaná rozhraní pro nahrávání a přehrávání audia. [21]

### *Video technologie*

- **UIImagePickerController** – Kontroler pro výběr uživatelských médií

- AVKit – Framework poskytující jednoduchá rozhraní pro video prezentaci. Podporuje full-screen i částečné zobrazení při přehrávání.
- Core Media – framework definující nízko úroňové datové typy a rozhraní pro práci s médiálním obsahem. [21]

### *Cocoa Touch*

Představuje klíčovou vrstvu pro vývoj iOS aplikací. Pomocí frameworků na této vrstvě lze definovat konečný vzhled aplikace, poskytují základní aplikační infrastrukturu a podporu pro klíčové technologie jako multitasking, dotyková vstup, push notifikace a mnoho dalších vysoko-úroňových systémových služeb. [22]

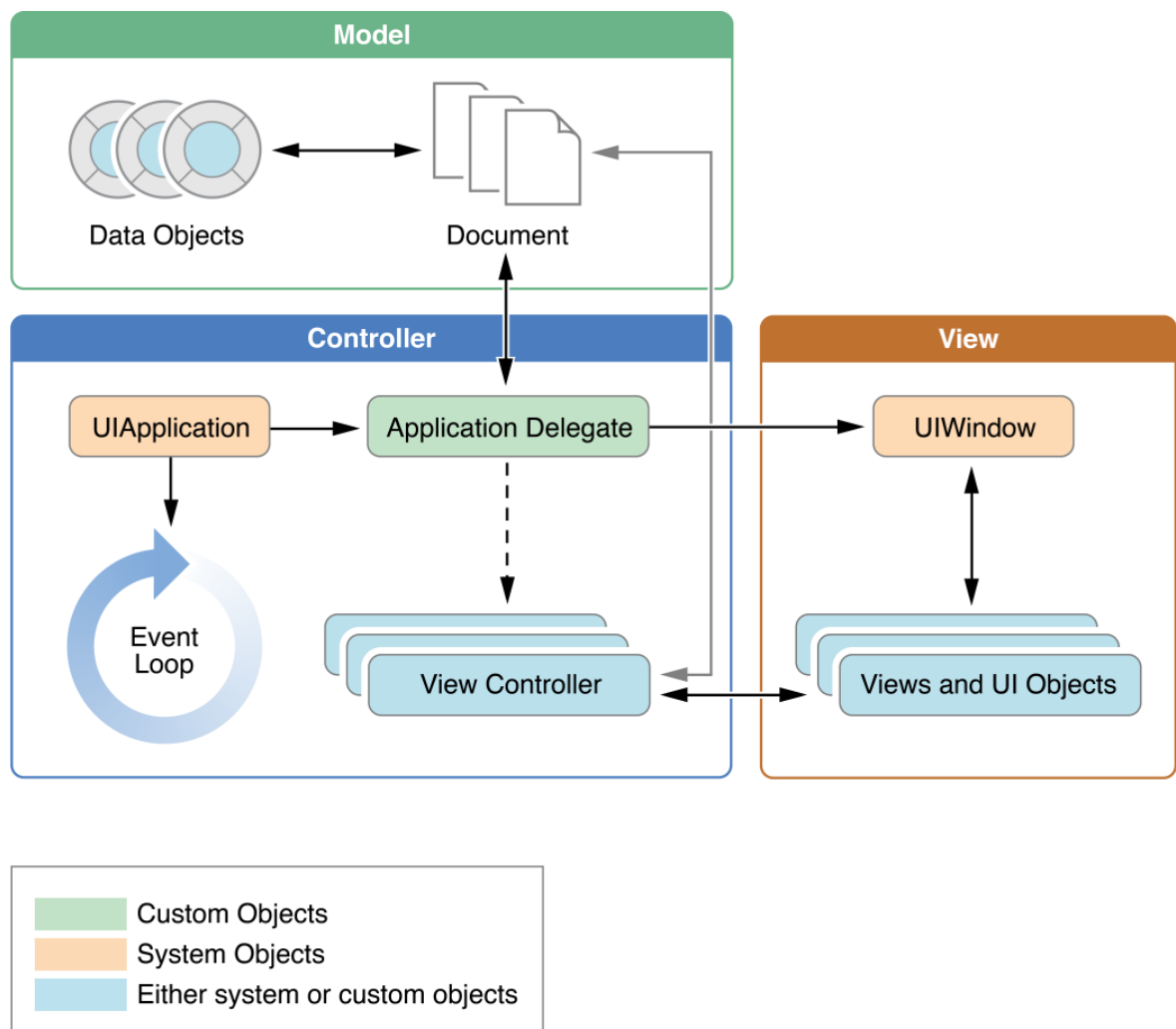
- **Address Book UI Framework** - Objective-C rozhraní, pro zobrazování a manipulaci s kontakty.
- **EventKit UI Framework** - Poskytuje prostředky pro nastavování plánovaných událostí.
- **GameKit Framework** – obsahuje podporu pro herní centrum, které umožňuje uživatelům sdílet své herní výsledky a jiné herní informace.
- **iAd Framework** – poskytuje standardní view pro zobrazování reklamních bannerů
- **MapKit Framework** – poskytuje standardní mapové rozhraní pro umístění do aplikací. Obsahuje také rozhraní pro přizpůsobení vzhledu a manipulaci s mapami.
- **Message UI Framework** – poskytuje podporu pro tvorbu emailů a SMS zpráv z aplikací.
- **Notification Center Framework** – poskytuje podporu pro tvorbu widgetů, které zobrazují informace v notifikačním centru.
- **PushKit Framework** – poskytuje podporu pro VoIP aplikace.
- **Social Framework** – poskytuje UI podporu pro tvorbu a konzumaci obsahu na sociálních sítích
- **UIKit Framework** – poskytuje základní infrastrukturu pro implementaci grafických, událostmi řízených aplikací. Tento Framework obsahuje například:
  - Základní infrastrukturu a nástroje pro správu aplikací
  - Objekty, reprezentující standardní systémová view a ovládací prvky
  - Podporu pro obsluhu událostí založených na interakci s dotykovou obrazovkou
  - Podpora pro multitasking

- Podpora pro tisk
- Podpora pro animaci UI komponent
- Podpora pro textový a webový obsah
- Podpora pro vyjímání, kopírování a vkládání obsahu
- Podpora pro push notifikace a další [22]

### 1.3.2 iOS aplikace

iOS aplikace jsou postaveny na modelu MVC neboli Model-View-Controller, který je charakteristický rozdělením aplikace do tří nezávislých částí – datový model, řídicí logiku a uživatelské rozhraní. [23]

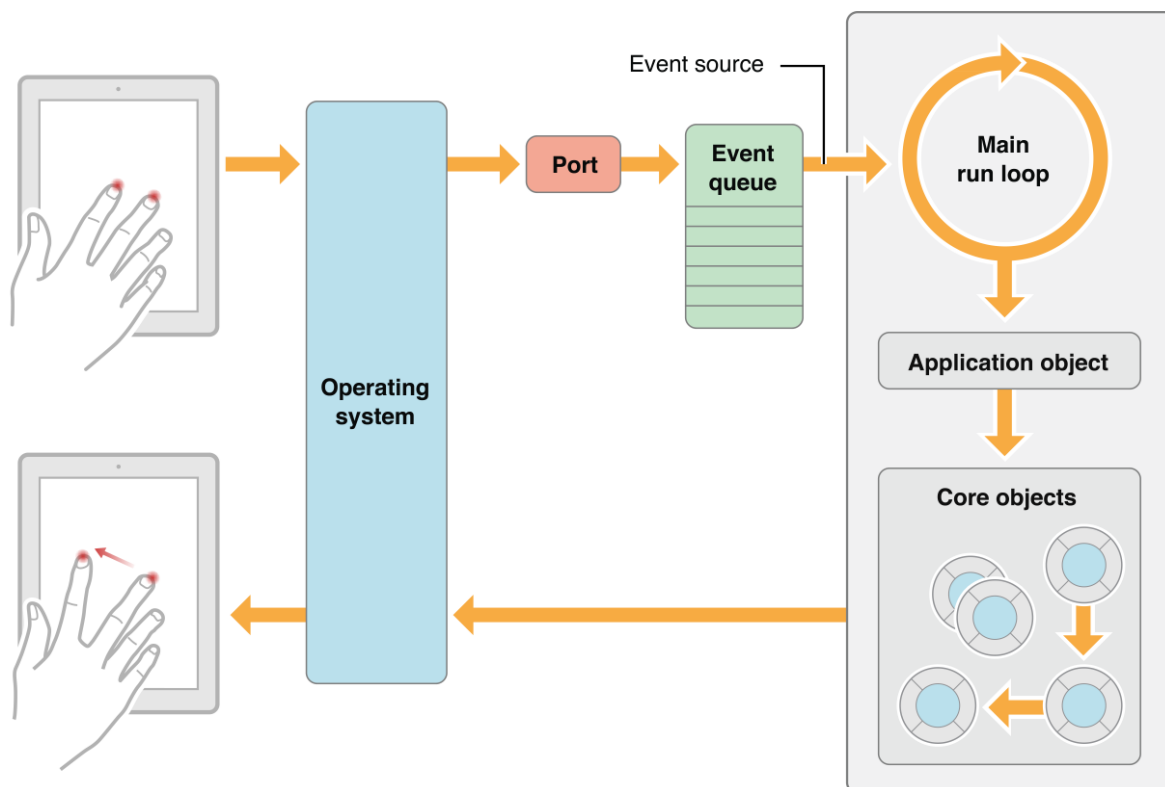
- **UIApplication** – stará se o interakci mezi systémem a jednotlivými komponentami aplikace.
- **Application Delegate** - spolupracuje přímo s UIApplication a má za úkol obsluhu inicializace aplikace, obsluhu přechodu stavů a dalších událostí.
- **Documents a Data model objects** – slouží pro ukládání a poskytování obsahu aplikace například práci s databázovými záznamy, nebo s obrázky
- **View controller** – spravuje obsah a celkovou prezentaci aplikace. V této části se obsah transformuje a vhodným způsobem vkládá do prezentační vrstvy – jednotlivých view (pohledů).
- **UIWindow** – je komponenta, která koordinuje prezentaci jednotlivých view na obrazovce. Většina aplikací má pouze jedno okno, které je prezentováno na hlavní obrazovce, mohou však existovat i aplikace s dodatečnými okny pro obsah zobrazovaný na externí obrazovce.
- **Views a Controls** – představují vizuální reprezentaci aplikačního obsahu. View je komponenta, která se stará o vykreslení obsahu v navržené čtvercové oblasti. Controls představují speciální view pro interakci s uživatelem jako jsou tlačítka, textová pole, přepínače a další. [23]



Obr. 9. Struktura iOS aplikace. [23]

***Hlavní smyčka aplikace***

Hlavní smyčka aplikace obstarává všechny události spojené s interakcí s uživatelem. O zavedení smyčky během startu aplikace se stará komponenta UIApplication a ta také umožňuje nastavit patřičné reakce při identifikaci události zachycené smyčkou a vyvolané uživatelem. Tato smyčka běží v hlavním vlákne aplikace a události vyvolané uživatelem jsou zpracovávány sériově v pořadí, v jakém byly zaznamenány. [23]

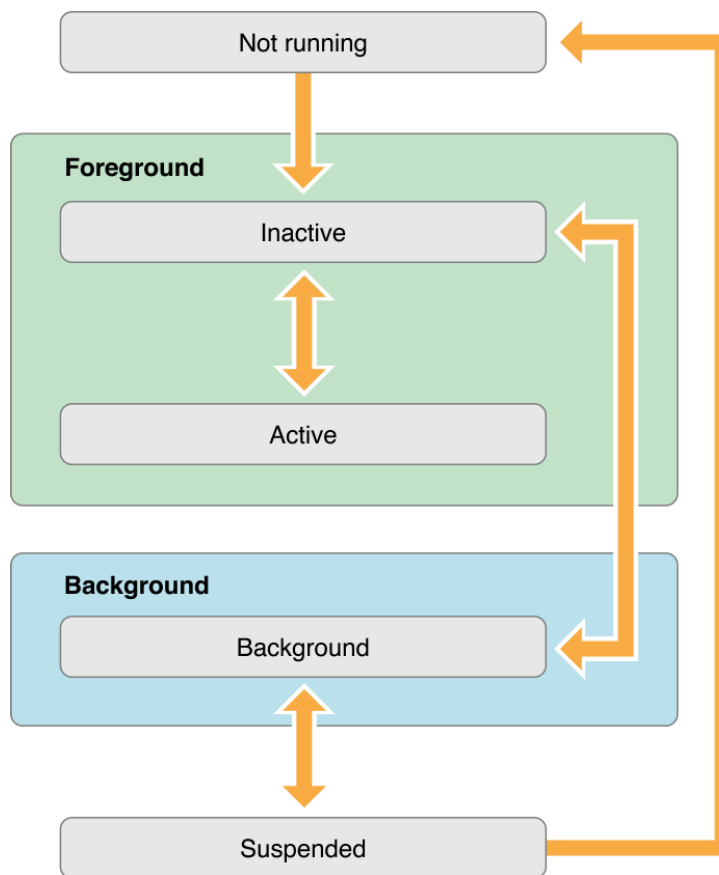


Obr. 10. Ukázka zpracování události v hlavní aplikační smyčce. [23]

### *Stavy aplikace*

V každém momentě se aplikace nachází v jednom z 5 stavů. Operační systém se stará o přechod mezi stavy v reakcích na vnitřní události systému, nebo na základě událostí vyvolaných uživatelem například stlačení domovského tlačítka zařízení, příchozí hovor atd. aplikace se může nacházet v těchto stavech: [23]

- **Nespuštěná** – aplikace nebyla dosud spuštěna, nebo byla spuštěna ale ukončena systémem
- **Neaktivní** – Aplikace běží na popředí, ale nepřijímá žádné události. Tento stav nastává většinou krátce při přechodu mezi jinými stavy.
- **Aktivní** – Aplikace běží na popředí a přijímá události. Běžný stav.
- **Na pozadí** – Aplikace, která provádí operace, ale běží na pozadí. Tento stav většinou nastává krátce před přechodem do stavu **Odložená**.
- **Odložená** – Aplikace, která je na pozadí a neprovádí žádné operace. Takto odložená aplikace zůstává v paměti zařízení. Při nedostatku paměti může systém aplikace z paměti odstranit pro získání více místa pro aplikace běžící na popředí. [23]



Obr. 11. Životní cyklus iOS Aplikací. [23]

#### 1.4 Multiplatformní vývoj mobilních aplikací

Pro vývoj nativních aplikací pro dvě a více platformů klade velké nároky na znalosti vývojářů. Při vývoji složitějších aplikací je často nezbytné mít vývojáře nebo týmy vývojářů specializující se na jednotlivé platformy. Vývojáři tak nejčastěji naráží na čtyři hlavní problémy: [24]

##### **První problém: Rozdílná paradigmatu uživatelského rozhraní**

Tři nejpoužívanější mobilní platformy dneška (iOS, Android a Windows 10 mobile) používají v globálu stejný princip prezentace grafického uživatelského rozhraní za použití multi-dotykových obrazovek. Je zde ale spousta rozdílů v detailech. Každá platforma má rozdílné možnosti navigace skrz aplikace a stránky, rozdílné konvence prezentace dat, rozdílné cesty jak vyvolat a zobrazovat kontextové menu a také rozdílné přístupy k dotykovému ovládní. [24]

**Druhý problém: Rozdílná vývojová prostředí**

Dnešní vývojáři jsou zvyklí pracovat v sofistikovaných integrovaných vývojových prostředích tzv. IDE (Integrated development environment). Takováto IDE existují pro každou ze tří hlavních mobilních platform. [24]

- Xcode pro iOS.
- Android studio pro OS Android.
- Visual Studio pro Windows aplikace.

**Třetí problém: Různá vývojová rozhraní**

Všechny tři hlavní mobilní platformy jsou postavené na různých operačních systémech s různými API. V mnoha případech tyto platformy implementují podobné typy uživatelského rozhraní avšak s různým názvoslovím. [24]

**Čtvrtý problém: Různé programovací jazyky**

Vývoj každé ze tří hlavních mobilních platform je postaven na rozdílném programovacím jazyce. Objective-C pro iOS, Java pro Android a C# pro Windows. I když jsou všechny tyto tři jazyky svým způsobem příbuzné (jsou objektově orientované a mají zděděnou syntaxi od jazyka C) jedná se o příbuzenství poměrně vzdálené.

Existuje však druhá cesta při vývoji multiplatformních aplikací a sice za použití multiplatformních frameworků, které umožňují psát aplikace pomocí jednoho programovacího jazyka nebo souboru všeobecně známých webových technologií. [24]

**1.4.1 PhoneGap**

PhoneGap od společnosti Adobe umožňuje vývojářům vyvíjet aplikace, které mohou běžet na různých mobilních zařízeních a platformách. Široká podpora PhoneGap je založená na vývoji široce podporovaných webových technologií jako jsou HTML, CSS a JavaScript. Vývojové nástroje PhoneGap umožňují tyto technologie zabalit do balíčků, které jsou kompatibilní s jednotlivými platformami. [25]

PhoneGap je podporován mimo jiné následujícími platformami:

- iOS
- Android
- Windows 8
- Windows Phone 7 a 8

- BlackBerry 5.x+
- WebOS
- Symbian
- Tizen [25]

PhoneGap obaluje kombinaci webových technologií HTML, CSS a Javascript do webového pohledu nativní aplikace konkrétní platformy. PhoneGap navíc poskytuje mechanismy, které dovolují JavaScriptu, spouštěného ve webovém pohledu přístup k mnoha základním funkcím zařízení jako jsou souborový systém, fotoaparát, akcelerometr, kontakty a další. [25]

K výhodám PhoneGapu patří:

- Všeobecně široká známost použitých vývojových technologií
- Aplikace se tváří jako nativní – je možné využívat aplikační obchody.
- Je Open Source a zdarma

K nevýhodám patří zejména výkon, ve kterém tyto aplikace zaostávají za nativními. [26]

#### 1.4.2 Xamarin

Xamarin je multiplatformní Framework pro vývoj mobilních aplikací v jazyce C#. Na rozdíl od PhoneGapu, který spadá do rodiny „Write-once, run everywhere“ frameworků, která bývá v různých ohledech omezující, Xamarin umožňuje vyvíjet plně nativní aplikace s využitím uživatelských rozhraní specifických pro jednotlivé platformy. [27]

Při vývoji nativních aplikací pomocí Xamarinu se používají dva postupy. Prvním je sdílení většiny kódu formou sdílených knihoven a následná implementace uživatelského rozhraní pomocí projektů Xamarin.Android a Xamarin.iOS. To vyžaduje jistou míru znalostí obou systémů, ale umožňuje vyvíjet komplexnější aplikace. Druhou možností je využití Xamarin.Forms, která umožňuje vyvíjet aplikace pro operační systémy Android, iOS a Windows v rámci jednoho zdrojového kódu. Prvky uživatelského rozhraní jsou zde unifikovány a nekladou tak nároky na znalost vývoje pro všechny tři platformy. Tento postup ale neumožňuje využití všech vývojových nástrojů jednotlivých platforem. [27]

#### Jazyk C#

Programovací jazyk C# byl představen společností Microsoft v roce 2000 a od této doby je soustavně vyvíjen. Jedná se tak o poměrně nový programovací jazyk (alespoň v porovnání s Java nebo Objective-C). Jazyk C# je vysokoúrovňový objektově orientovaný silně typový



programovací jazyk, silně ovlivněný jazykem C++ (částečně i jazykem JAVA), ale s mnohem čistší syntaxí než C++. Jazyk C# je navíc neustále rozvíjen. Za poslední roky přibýly do jazyka nové funkcionality jako například podpora generičnosti, lambda funkce, LINQ a asynchronní operace. [28]

Od svého počátku je jazyk C# silně spjat s vývojovým frameworkem .NET vyvíjeným taktéž společností Microsoft. Ve svém základu framework .NET poskytuje infrastrukturu pro základní datové typy C# (int, double, string a další). Rozšiřující .NET knihovny poskytují podporu pro mnoho rozdílných typů programování. Jedná se například o: [28]

- Knihovna pro podporu matematických operací
- Debugování
- Reflexe
- Kolekce
- Globalizace
- Práce se soubory
- Síťové operace
- Zabezpečení
- Vícevláknové operace
- Webové služby
- Práce s formáty XML a JSON a mnoho dalších

### **Historie Xamarinu**

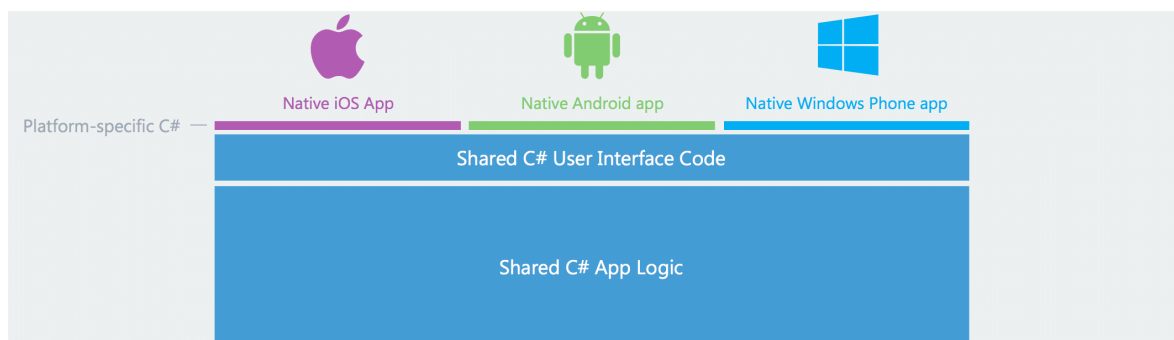
V červnu roku 2000, společnost Ximian ohlásila open-source projekt Mono s cílem vytvořit alternativní implementace C# a .NET kompilera, který by běžel pod Linuxem. O jedenáct let později se zakladatelé Ximian založili novou společností Xamarin, které stále pracovala na projektu Mono, ale tentokrát s cílem adaptovat Mono jako základní multiplatformní řešení pro mobilní platformy. V roce 2014 byla vydána open-source verze C# kompilera s názvem .NET Compler Platform (původně známá pod názvem „Roslyn“) a .NET Foundation převzal záštitu pro open-source .NET technologie ve kterých Xamarin hrál významnou roli. V březnu 2016 dochází k přesunu Xamarinu pod křídla Microsoftu a produkty Xamarin se stávají zdarma dostupné pro všechny uživatele Visual Studia. [24]

Vývojáři dnes mohou používat IDE Visual Studio, jazyk C# a .NET Framework od společnosti Microsoft, jako hlavní vývojové nástroje pro všechny tři hlavní mobilní platformy. Pro

vývoj iOS aplikací je nicméně nutné vlastnit zařízení Mac s nainstalovaným vývojovým prostředím Xcode pro sestavování aplikací. [24]

### Sdílený kód

Hlavní výhoda multiplatformního vývoje pod jedním programovacím jazykem tkví ve schopnosti sdílet kód aplikace. Než jsou aplikace schopné takový kód sdílet, je nutné aby aplikace byly správně strukturované. Strukturováním se rozumí rozdělit zdrojový kód do funkčních vrstev. Jedním z takových rozdělení může být například populární MVC architektura, tedy rozdělení aplikace na vrstvu modelů (datová vrstva), vrstvu pohledů (vizuální reprezentace dat) a kontrolery (obsluha uživatelských vstupů). Modernizovaná verze MVVM, rozděluje aplikaci na vrstvu modelů, pohledů a pohledových modelů (starají se o předávání dat mezi modelem a pohledem). Z pohledu multiplatformního vývoje je vhodné zvolit architekturu MVVM a rozdělit kód na platformě závislé pohledy a platformě nezávislé pohledové modely a modely. Za normálních okolností především pohledové modely potřebují využívat funkcí specifických pro API jednotlivých platforem, jako je například síťová komunikace, kolekce nebo více-vláknové operace. V případě Xamarinu zde přichází na scénu .NET framework, který programování těchto funkcí umožní v platformě nezávislé podobě. Část platformě nezávislé aplikace může být oddělena od zdrojového kódu platformě závislé části a vystupovat jako samostatný projekt v podobě Shared Asset Project (SAP) nebo Portable Class Library (PCL). [24]



Obr. 12. Grafické znázornění technologie Xamarin.Forms. [29]

## 2 VÝUKOVÉ PLÁNY MATEMATIKY PRO ZÁKLADNÍ A STŘEDNÍ ŠKOLY

Matematika poskytuje žákům vědomosti a dovednosti, které jsou potřebné pro orientaci v praktickém životě, vytváří předpoklady pro uplatnění v mnoha oborech a různých směrech studia, rozvíjí intelektuální schopnosti, paměť, tvořivost, abstraktní myšlení, představivost a schopnost logického myšlení. Současně také přispívá k rozvoji osobnostních rysů, jako je vytrvalost, pracovitost a kritičnost. [30]

Výuka matematiky si klade za cíl, aby se žáci naučili

- Provádět výpočetní výkony s přirozenými čísly a zlomky, písemně, pamětně a při složitých úlohách používat kapesní kalkulátor.
- Řešit úlohy z praxe jako jsou procentní počty nebo jednoduché úrokování.
- Provádět odhady výsledků a posuzovat jejich reálnost, provádět racionální zaokrouhlování.
- Používat a číst statistické tabulky a diagramy.
- Používat proměnné, chápat jejich význam, řešit rovnice a nerovnice.
- Chápat, graficky znázorňovat a zapisovat závislosti kvantitativních jevů ve společnosti a v přírodě.
- Řešit geometrické úlohy, vypočítat obsahy a obvody rovinných obrazců.
- Schopnost dokazovat jednoduchá tvrzení a vyvozovat z nich logické závěry.

### 2.1 Výukový plán matematiky pro základní školy

Podle rámcového vzdělávacího programu je předmět Matematika pro základní školy rozdělen do čtyř tematických okruhů: Čísla a početní operace (čísla a proměnné na druhém stupni), Závislosti vztahy a práce s daty, Geometrie v rovině a v prostoru a Nestandardní aplikační úlohy a problémy. [31]

#### 2.1.1 Čísla a početní operace (Čísla a proměnné)

V tomto tematickém okruhu se žáci učí aritmetickým operacím ve třech složkách: dovednosti provádět operaci, algoritmické porozumění (proč se operace řeší pomocí popsaného postupu) a významové porozumění (propojení operací s reálnými situacemi). Studenti se též učí získávat číselné údaje výpočtem, odhadem, měřením či zaokrouhlením. Na druhém

stupni se žáci seznamují s pojmem proměnná a s její rolí při matematizaci reálných situací. [31]

#### ***Očekávané výstupy – 1. Období (3. třída)***

- Umění používat přirozená čísla k modelování reálných situací, v daném souboru počítat předměty a tyto soubory s daným počtem prvků vytvářet
- Číst, zapisovat a porovnávat přirozená čísla do 1000, užívat a umět zapisovat vztahy rovnosti a nerovnosti.
- Zobrazovat čísla na číselné ose.
- Z paměti provádět jednoduché početní operace s přirozenými čísly.
- Řešit a také tvořit úlohy, ve kterých student aplikuje a modeluje osvojené početní operace. [31]

#### ***Očekávané výstupy – 2. Období (5. třída)***

- Pamětně i písemně sčítat i odčítat a využívat při tom komutativnost a asociativnost.
- Provádět písemné výpočetní operace s přirozenými čísly.
- Provádět odhady, zaokrouhlovat přirozená čísla a kontrolovat výsledky početních operací v oboru přirozených čísel.
- Určovat a modelovat část celku, umět používat zápis ve formě zlomku.
- Číst zápis desetinného čísla a vyznačit jej na číselné ose.
- Porozumět záporným číslům a vyznačit je na číselné ose. [31]

#### ***Očekávané výstupy – 3. Období (9. třída)***

- Provádět početní operace v oboru celých a racionálních čísel. Užívat ve výpočtech druhou mocninu a odmocninu.
- S danou přesností provádět zaokrouhlování a odhady, účelně využívat kalkulátor.
- Užívat různé způsoby kvantitativního vyjádření celku a částí (procentem, zlomkem, desetinným číslem, poměrem).
- Matematizovat jednoduché reálné situace s využitím proměnných, určit hodnotu výrazu, provádět rozklad mnohočlenu na součin pomocí vytýkání a užívání vzorců, sčítat a násobit mnohočleny.
- Umět formulovat a řešit reálnou situaci za pomoci rovnic a jejich soustav

- Analyzovat a řešit jednoduché problémy, v nichž využívá matematický aparát v oboru celých a racionálních čísel. [31]

### 2.1.2 Závislost, vztahy a práce s daty

V tomto okruhu se žáci učí rozpoznávat typy změn a závislostí, které jsou projevem každodenních jevů reálného světa, seznamují se s jejich reprezentacemi. Uvědomují si závislosti známých jevů, chápou, že změnou může být růst, pokles ale i nulová hodnota. Žáci se tyto změny a závislosti učí analyzovat z tabulek, grafů a diagramů a v jednoduchých případech je vyjadřují matematickým předpisem. Zkoumání a pochopení těchto závislostí směřují k pochopení pojmu funkce. [31]

#### *Očekávané výstupy – 1. Období (3. třída)*

- Orientace v čase, provádění převodů jednotek času
- Popis jednoduché závislosti z praktického života.
- Doplnovat schémata, tabulky, posloupnosti čísel. [31]

#### *Očekávané výstupy – 2. Období (5. třída)*

- Vyhledávat, sbírat a třídit data.
- Číst a sestavovat jednoduché tabulky a diagramy. [31]

#### *Očekávané výstupy – 3. Období (9. třída)*

- Vyhledávat, zpracovávat, vyhodnocovat a porovnávat data.
- Určovat vztah přímé a nepřímé úměrnosti.
- Vyjadřovat funkční vztah pomocí tabulky, rovnice nebo grafem.
- Matematizace reálných situací využitím funkčních vztahů. [31]

### 2.1.3 Geometrie v rovině a v prostoru

V tomto okruhu se žáci učí znázorňovat geometrické útvary a modelovat geometricky reálné situace. Učí se hledat podobnosti a odlišnosti geometrických útvarů vyskytující se kolem nás. Učí se chápat vzájemné polohy objektů v rovině a prostoru, odhadovat, porovnávat, měřit délky a velikosti úhlů, obvod, obsah, povrch a objem geometrických útvarů. Zkoumáním útvarů a prostoru se žáci učí řešit polohové a metrické úlohy a problémy, které vycházejí z běžných životních situací. [31]

**Očekávané výstupy – 1. Období (3. třída)**

- Rozeznávat, modelovat, pojmenovávat a popisovat základní rovinné útvary a jednoduchá tělesa.
- Porovnávat velikosti útvarů, měřit a odhadovat délku úsečky.
- Rozeznat a modelovat jednoduché souměrné útvary v rovině. [31]

**Očekávané výstupy – 2. Období (5. třída)**

- Znázorňovat a rýsovat základní rovinné útvary (čtverec, obdélník, trojúhelník a kružnici)
- Sčítat a odčítat grafické úsečky, určit délku lomené čáry, určit obvod mnohoúhelníku sečtením délek jeho stran.
- Určit obsah obrazce pomocí čtvercové sítě a užívat základní jednotky obsahu.
- Rozpoznat a znázorní ve čtvercové síti jednoduché osově souměrné útvary. [31]

**Očekávané výstupy – 3. Období (9. třída)**

- Využívat polohové a metrické vlastnosti základních rovinných útvarů při řešení úloh a jednoduchých praktických problémů.
- Třídít a charakterizovat základní rovinné útvary.
- Určit velikost úhlu měřením a výpočtem.
- Odhadnout a vypočítat obvod a obsah základních rovinných útvarů, povrch a objem těles.
- Načrtnout a sestrojít obraz rovinného útvaru v osově a středové souměrnosti.
- Načrtnout a sestrojít síť a obrazy základních těles v rovině. [31]

**2.1.4 Nestandardní aplikační úlohy**

Posledním okruhem rámcového vzdělávání pro základní školy v předmětu matematika jsou Nestandardní aplikační úlohy a problémy, jejichž řešení, do značné míry, může být nezávislé na matematických znalostech a dovednostech, ale při nichž se uplatní především logické myšlení. Takovéto úlohy by měly zasahovat do všech tematických okruhů v průběhu základního vzdělávání. Žáci se učí pochopit, analyzovat a řešit situace, problémy a úlohy z běžného života. Učí se řešit optimalizační úlohy, provádět situační náčrty, třídít údaje a podmínky. Řešení těchto úloh posiluje u žáků schopnosti logického myšlení. [31]

***Očekávané výstupy – 2. Období (5. třída)***

- Řešit jednoduché praktické slovní úlohy problémy. Řešení těchto úloh je do značné míry nezávislé na obvyklých postupech a algoritmech školské matematiky. [31]

***Očekávané výstupy – 3. Období (9. třída)***

- Řešit úlohy a problémy za užití logických úvah a kombinačního úsudku a nalézat různá řešení.
- Řešit úlohy pomocí prostorové představivosti, aplikace a kombinace poznatků a dovedností z různých tematických a vzdělávacích oblastí. [31]

**2.2 Výukový plán matematika pro střední školy (Gymnázia)**

Matematické vzdělávání na gymnáziích slouží k rozvíjení a prohlubování chápání kvantitativních vztahů reálného světa, utváření kvantitativní gramotnosti studentů a jejich schopností grafického vhledu. Prostředky pro nové a hlubší poznání a předpoklady pro další studium jsou ovládnutí matematického aparátu, elementů matematického myšlení, deduktivní úvahy a vytváření hypotéz. [32]

Výuka matematiky na gymnáziích si klade za cíl, rozvíjet u studentů poznání, že matematika nachází uplatnění v mnoha oborech lidské činnosti (technika, ekonomie ale i společenské vědy), že užitečným pomocníkem matematiky jsou moderní technologie a že je ovlivňována vnějšími podněty (například z oblasti přírodních věd). [33]

Matematika na gymnáziích rozvíjí znalosti studentů v pěti oblastech:

1. Argumentace a ověřování
2. Číslo a proměnná
3. Práce s daty, kombinatorika, pravděpodobnost
4. Závislosti a funkční vztahy
5. Geometrie. [33]

***Argumentace a ověřování – očekávané výstupy***

- Číst a zapisovat tvrzení v symbolickém jazyce matematiky.
- Užívat korektní logické spojky a kvantifikátory.
- Rozlišit větu a definici, předpoklad a závěr věty.
- Rozlišit správný a nesprávný úsudek.

- Vytvářet hypotézy, odůvodnit jejich pravdivost či nepravdivost, vyvracet nesprávná tvrzení.
- Zdůvodňovat své postupy a ověřovat správnost řešení. [33]

### ***Číslo a proměnná – očekávané výstupy***

- Užívat vlastnosti dělitelnosti přirozených čísel.
- Aplikovat geometrický význam absolutní hodnoty, operovat s intervaly.
- Provádět operace s mocninami a odmocninami, upravovat číselné výrazy.
- Upravovat výrazy s proměnnými, určovat definiční obory výrazů.
- Aplikovat rozkládání mnohočlenů při řešení rovnic a nerovnic.
- Geometricky interpretovat číselné, algebraické a funkční vztahy. Graficky znázorňovat řešení rovnic, nerovnic a jejich soustav.
- Analyzovat a řešit problémy, v nichž se uplatní řešení lineárních a kvadratických rovnic a jejich soustav. [33]

### ***Práce s daty, kombinatorik, pravděpodobnost***

- Řešit reálné problémy s kombinatorickým podtextem.
- Výpočet pravděpodobnosti za využití kombinatorických postů. Upravit výrazy s faktoriály a kombinačními čísly.
- Diskutovat a kriticky hodnotit statistické informace.
- Volit a užívat vhodné statistické metody k analýze a zpracování dat.
- Číst a interpretovat tabulky, diagramy a grafy, reprezentovat soubory dat graficky. [33]

### ***Závislost a funkční vztahy***

- Načrtnout grafy jednodušších funkcí zadaných funkčním předpisem a určit jejich vlastnosti.
- Formulovat a zdůvodňovat vlastnosti studovaných funkcí a posloupností.
- Aplikovat vztahy mezi exponenciálními, logaritmickými a goniometrickými funkcemi.
- Řešit aplikační úlohy s využitím poznatků o funkcích a posloupnostech. [33]



### *Geometrie*

- Používat geometrické pojmy, zdůvodňovat a využívat vlastnosti geometrických útvarů v rovině a v prostoru.
- Určovat vzdálenosti, odchylky a vzájemnou polohu lineárních útvarů.
- Aplikovat funkční vztahy, trigonometrii a úpravy výrazů v úlohách početní geometrie.
- Řešit polohové a nepolohové konstrukční úlohy užitím všech bodů dané vlastnosti.
- Řešit planimetrické a stereometrické problémy motivované praxí.
- Využívat charakteristické vlastnosti kuželoseček k určení analytického vyjádření.
- Řešit analyticky metrické a polohové úlohy o lineárních útvarech v rovině.
- Řešit analyticky úlohy na vzájemnou polohu přímky a kuželosečky. [33]

## **II. PRAKTICKÁ ČÁST**

## 3 NÁVRH A REALIZACE MOBILNÍ VÝUKOVÉ APLIKACE

### 3.1 Návrh aplikace

Před přípravou samotného projektu bylo potřeba ujasnit si základní požadavky, které má aplikace splňovat a také za pomoci jakých prostředků těchto cílů dosáhnout.

Na začátku projektu jsem měl dvě představy, kterými by se měl vývoj aplikace ubírat. První z nich byla samostatná mobilní aplikace bez připojení na backendový server a tedy bez interakce na úrovni žák – učitel. Tato verze by byla méně náročná na vývoj, více však na obsah, který by se nemohl dynamicky rozvíjet. Druhá vize, byla aplikace s backend serverem, umožňující obsahovou stránku neustále rozvíjet a navíc rozšířit aplikaci o možnost komunikace mezi žákem a učitelem. Nakonec jsem se přiklonil k druhé vizi, i když jsem tak nechtěně rozšířil zadání diplomové práce. Nicméně tato úprava byla nutná pro širší využití aplikace, než byl původní záměr.

Druhým problémem bylo vybrat si technologie a postupy použitých pro vývoj. Z praxe mám nejvíce zkušeností s vývojem pro platformu .NET v programovacím jazyce C# a proto jsem se rozhodl využít tyto znalosti a všechny části tohoto systému jsem vyvíjel právě pod touto technologií.

Pro vývoj REST API, které bude základem pro Backend server, jsem využil framework ASP.NET Web API 2 od společnosti Microsoft a pro vývoj podpůrného webu jsem použil Framework ASP.NET MVC 5 od téže společnosti. Nativní Android aplikace jsou psány v jazyce Java a oficiální vývoj je doporučován v prostředí Android Studio. Pro mě však bylo výhodné použít framework Xamarin pro multiplatformní vývoj mobilních aplikací v jazyce C# a to nejen z důvodu lepší znalosti programovacího jazyka, ale i z důvodu možnosti budoucího rozšíření aplikace o další platformu.

Jak už bylo řečeno, celý systém se bude skládat ze tří základních komponent: backendový server s REST API, podpůrná webová aplikace a mobilní aplikace. Dalšími důležitými součástmi jsou backendová databáze a služba Google Cloud Messaging pro správu push notifikací. Komunikace mezi jednotlivými částmi systému je zobrazena na následujícím obrázku:

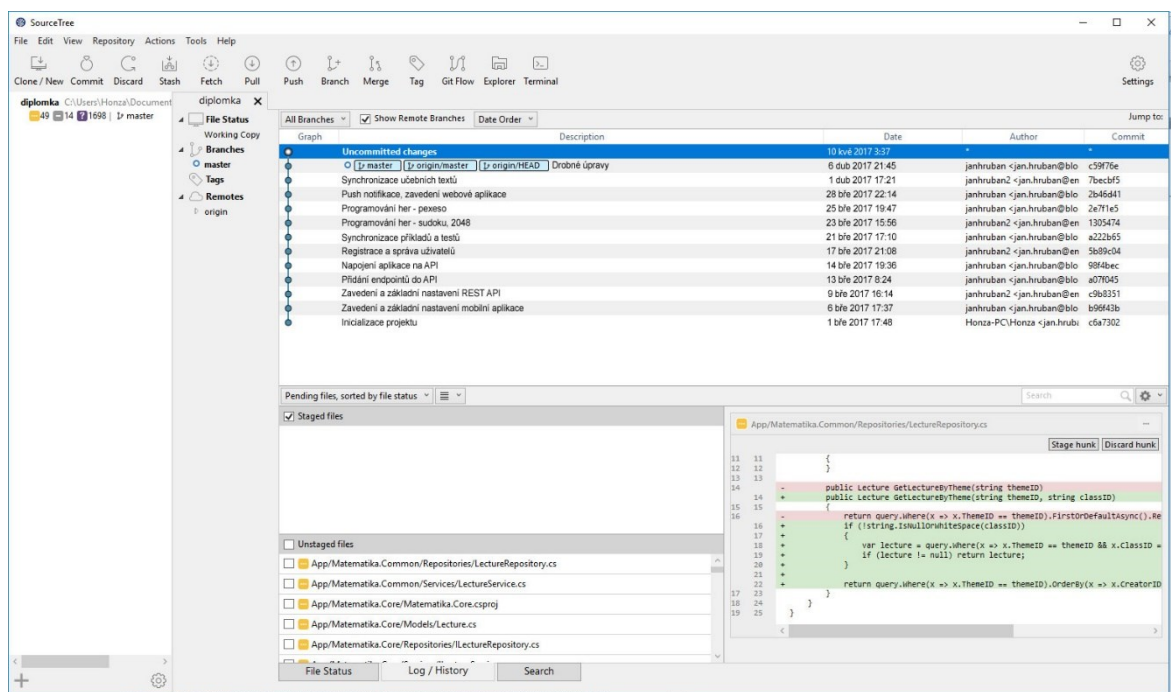


### 3.2.2 Microsoft SQL server

U tohoto projektu jsem pro vývoj databáze použil SQL Server 2016 od společnosti Microsoft a UI rozhraní pro správu databází Microsoft SQL Management Studio 2016. Databáze byla vyvíjena v jazyku T-SQL.

### 3.2.3 Git, Bitbucket a SourceTree

Pro verzování zdrojového kódu jsem použil verzovací systém Git. Pro uchování Git repozitáře jsem využil služeb serveru Bitbucket a pro snadnou správu repozitáře desktopovou aplikaci SourceTree.



Obr. 14. Ukázka verzování softwaru v programu SourceTree.

### 3.2.4 Microsoft Azure

Služba Azure od společnosti Microsoft je cloudová výpočetní platforma. Této službě jsem využil pro hostování backend a webové aplikace a také SQL serveru a databáze.

### 3.2.5 Xamarin

Xamarin je multiplatformní Framework pro vývoj mobilních aplikací. Umožňuje vyvíjet aplikace pod jedním zdrojovým kódem pomocí knihoven Xamarin.Forms nebo vývoj pod jednotlivými platformami s využitím knihoven Xamarin.Android a Xamarin.iOS a možností

sdílených knihoven. Pro tento projekt jsem zvolil vývoj pro platformu android s využitím sdílených (portable) knihoven pro možnost budoucího rozšíření o platformu iOS.

Pro vývoj pod tímto frameworkem jsem využil možnosti integrace vývojových nástrojů do prostředí Visual Studio.

### 3.3 Backend aplikace

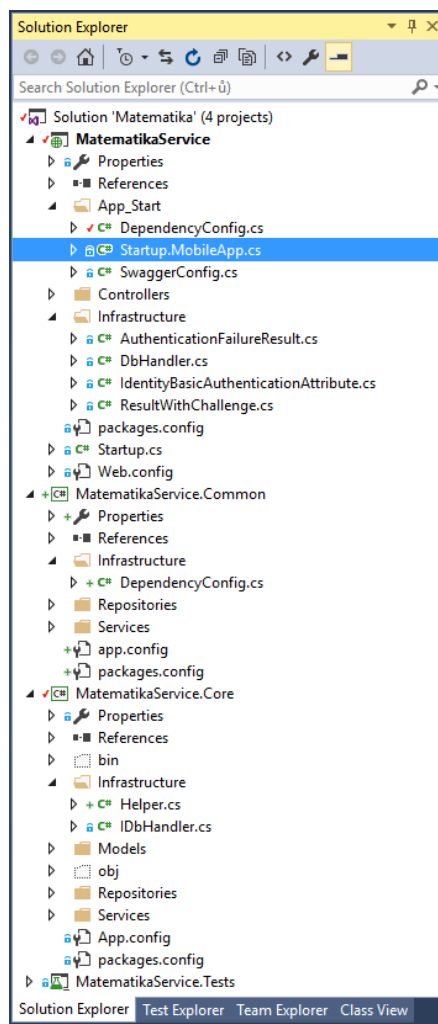
Backend pro mobilní aplikace je REST API postavené na frameworku Web API 2 od Microsoftu. Tento Framework umožňuje jednoduché routování a implementaci zabezpečovacích technik. Tato aplikaci slouží především k propojení hlavní databáze s jednotlivými zařízeními, což umožní dynamickou tvorbu a sdílení obsahu a také přenositelnost uživatelských účtů. Tento projekt využívá knihovnu Autofac pro vkládání závislostí (Dependency injection) a knihovnu Dapper, což je mini ORM knihovna pro komunikaci s databází. Je zde také knihovna Newtonsoft pro snadnou serializaci a deserializaci JSON objektů a knihovna Swagger pro generování API dokumentace.

#### 3.3.1 Struktura projektu

Celá aplikace je pro přehlednost rozdělena do tří projektů:

1. MatematikaService.Core – tento projekt slouží k uchování doménových tříd a rozhraní (interfaců) pro servisové třídy repozitáře. Obsahuje následující adresáře:
  - a. Infrastructure – rozhraní databázového připojení a statické knihovny
  - b. Services – rozhraní servisových tříd, které obsahuje logiku pro zpracování dat
  - c. Repositories – rozhraní repozitářů, které slouží pro vkládání a získávání dat z databáze.
2. MatematikaService.Common – tento projekt obsahuje implementace servisových tříd a repozitářů. Adresáře:
  - a. Infrastructure – obsahuje konfigurační soubor pro vkládání závislostí
  - b. Services – obsahuje implementace servisových tříd
  - c. Repositories – obsahuje implementace repozitářů
3. MatematikaService – Web API projekt, obsahující kontrolery a základní nastavení API. Součástí jsou adresáře:

- a. AppStart – zde se nachází základní nastavení aplikace jako registrace závislostí, implementace zabezpečení a generování projektové dokumentace Swagger
- b. Controllers – obsahuje kontrolery s implementací jednotlivých endpointů REST API
- c. Infrastructure – Obsahuje implementaci třídy pro obsluhu databázového připojení a třídy pro obsluhu zabezpečení.



Obr. 15. Struktura backend aplikace.

### 3.3.2 Databáze

Databáze je koncipována do šesti základních tabulek, které představují šest základních entit v aplikaci. V tomto projektu jsem využil uložené procedury pro snadné oddělení databázového jazyka SQL od zdrojového kódu aplikace. Toto řešení navíc umožňuje ladění databázových operací bez potřeby zasahovat do zdrojových kódů aplikace (při zachování vstupů a výstupů procedur). Dále jsou zde tabulky pro číselníky a vazební tabulky pro entitní vztahy N:M.

Pro synchronizaci mezi mnoha zařízeními bylo nevhodné použití celočíselných identifikátorů s automatickým navyšováním, proto jsem pro entitní tabulky zvolil identifikátor GUID, což je systémový automaticky generovaný identifikátor složený z 32 hexadecimálních znaků a čtyř pomlček, jenž má SQL server jako základní datový typ. Možnost vygenerování dvou stejných GUID identifikátorů v jedné tabulce se limitně blíží nule a proto jsem nepovažoval za nutné zavádět ochranné mechanismy pro tyto případy.

Tab. 2. Přehled a popis tabulek v backend databázi.

Název	Popis
<b>dbo.Classes</b>	Tabulka pro uchování vyučovacích tříd.
<b>dbo.ClassLectures</b>	Tabulka vazeb pro přednášky učitelů pro jednotlivé třídy.
<b>dbo.ClassLevels</b>	Číselníková tabulka tříd (např. 1. Třída základní školy).
<b>dbo.Examples</b>	Tabulka příkladů.
<b>dbo.ExampleTypes</b>	Tato tabulka obsahuje číselník typů příkladů (číselná odpověď, výběrový).
<b>dbo.Lectures</b>	Tabulka učebních textů vytvářených učiteli.
<b>dbo.Roles</b>	Číselník rolí uživatelů (Student, Učitel).
<b>dbo.Sudoku</b>	Tabulka se zadáními pro hru sudoku.
<b>dbo.TaskAnswers</b>	Tabulka uchovávající odpovědi studentů na jednotlivé úkoly, zadané učitelem.
<b>dbo.TaskItems</b>	Vazební tabulka 1:N mezi úkoly a příklady.



<b>dbo.Tasks</b>	Tabulka úkolů, zadaných učitelem.
<b>dbo.Themes</b>	Tabulka učebních témat.
<b>dbo.Users</b>	Tabulka uživatelů.

### 3.3.3 API Dokumentace

Jak již bylo řečeno, mobilní aplikace komunikuje s backend serverem prostřednictvím REST API. K získávání a ukládání informací se mobilní aplikace připojuje přes tzv. endpointy, což jsou jednotlivé URL adresy API s patřičnými parametry. Názvy pro adresy byly voleny tak, aby byly v kombinaci s názvy parametrů samopopisné.

Při každém požadavku na API je vyžadována základní autentifikace (Basic authentication), tzn., že při každém požadavku je vyžadováno umístění autentizačního tokenu do http hlavičky. API z tohoto tokenu získá přihlašovací jméno a heslo aplikace a porovná ho s hodnotami v konfiguračním souboru. Pokud se hodnoty shodují, vrátí API požadovaný výsledek, v opačném případě vrátí http stav 401 – neautorizovaný požadavek.

Tab. 3. Přehled a popis jednotlivých endpointů REST API.

Cesta	Parametry	Popis
/api/classes/class/name	Název třídy	Endpoint pro získání seznamu tříd podle části názvu třídy (použito pro autocomplete).
/api/classes/save	Model třídy	Slouží pro ukládání tříd.
/api/classes/class	ID třídy	Slouží pro získání modelu třídy podle identifikátoru.
/api/classes/classes	Název třídy	Slouží k ověření existence třídy podle přesného názvu.
/api/classes/set	ID uživatele, ID třídy	Slouží pro přiřazení třídy k uživateli.
/api/examples/example	ID příkladu	Vrací model příkladu podle identifikátoru.
/api/examples/examples	Datum poslední synchronizace	Vrací seznam příkladů podle data poslední úpravy. V případě nevyplnění vstupní hodnoty, vrátí všechny aktivní příklady.

/api/examples/save	Model příkladu	Slouží pro uložení příkladu do databáze.
/api/games/sudoku		Vrací seznam zadání pro hru sudoku.
/api/lectures/class	ID třídy	Vrací přednášky podle identifikátoru třídy.
/api/lectures/lecture	ID přednášky	Vrací model přednášky podle identifikátoru.
/api/lectures/save	Model přednášky	Slouží pro uložení přednášky.
/api/tasks/task	ID úkolu	Vrací model konkrétního úkolu podle identifikátoru.
/api/tasks/user	ID uživatele	Vrací seznam úkolů podle identifikátoru uživatele.
/api/tasks/answers	ID úkolu, ID uživatele	Vrací seznam odpovědí na úkol podle identifikátoru úkolu a identifikátoru uživatele.
/api/tasks/save	Model úkolu	Slouží pro uložení zadání úkolu.
/api/answer/save	List odpovědí	Slouží pro uložení odpovědí na úkol.
/api/themes/themes		Vrací seznam učebních témat pro jednotlivé ročníky.
/api/users/authenticate	Přihlašovací jméno a heslo	Slouží pro ověření uživatele podle přihlašovacího jména a hesla.
/api/users/insert	Model uživatele	Slouží pro vložení uživatele.
/api/users/update	Model uživatele	Slouží pro úpravu uživatelských údajů.
/api/users/students	ID třídy	Vrací seznam studentů třídy podle identifikátoru třídy.
/api/users/user	ID uživatele	Vrací model uživatele podle identifikátoru.
/api/users/user	Jméno uživatele	Vrací model uživatele podle přihlašovacího jména.
/api/users/user/facebook	Facebook ID	Vrací model uživatele podle Facebook identifikátoru.

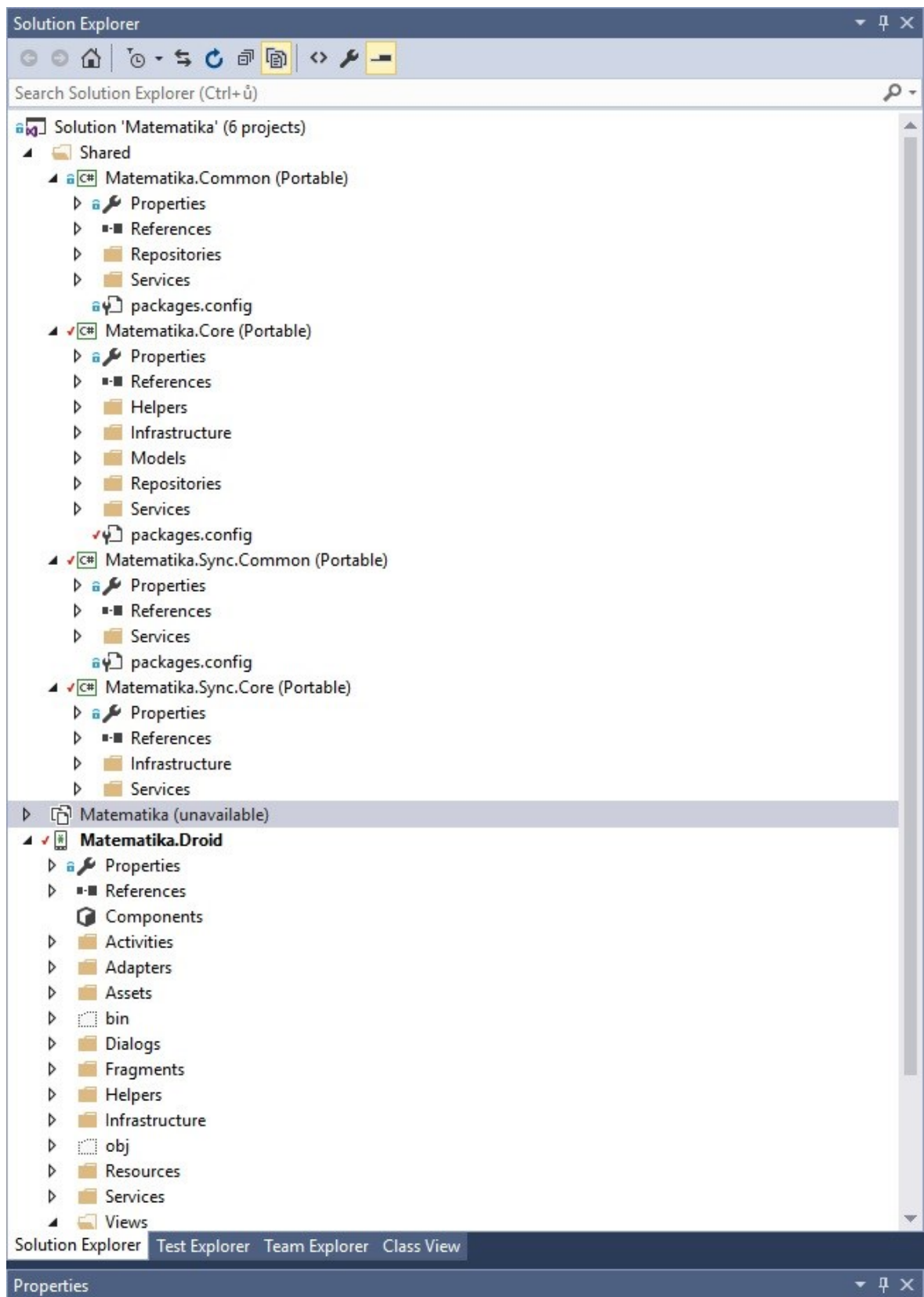
### 3.4 Mobilní aplikace

Mobilní aplikace byla vyvíjena v jazyce C# s využitím multiplatformního frameworku Xamarin pro OS Android, který tvoří nejdůležitější knihovnu aplikace. Dále jsem zde využil opět knihovnu Autofac pro vkládání závislostí a newtonson.json pro serializaci a deserializaci JSON řetězců. Další důležitou knihovnou v tomto projektu je SQLite-net pro práci s relační databází SQLite v prostředí .NET, knihovna Xamarin.Facebook pro jednoduché propojení aplikace se sociální sítí Facebook a Xamarin.GooglePlayServices pro využití služeb společnosti Google.

#### 3.4.1 Struktura projektu

Celá aplikace se skládá z celkem pěti projektů. Jeden projekt je specializovaný projekt samotné aplikace pro OS Android, obsahující nezbytné knihovny a konfigurační soubory. Další čtyři projekty jsou ve formě přenosných (portable) knihoven, který zjednoduší případné rozšíření aplikace o verzi pro iOS nebo Windows. Tyto knihovny slouží pro komunikaci s API a pro obsluhu lokální SQLite databáze. Dále jsou zde pomocné třídy, pro obecné a často používané funkce, jako třeba hashování hesel a algoritmy pro matematické hry.

1. **Matematika.Core** – tato knihovna obsahuje rozhraní pro servisní třídy a repozitáře pro obsluhu lokální SQLite databáze, modely pro základní entity a rozhraní pro konfigurační třídy.
2. **Matematika.Common** – tato knihovna obsahuje implementace pro servisní třídy a repozitáře pro obsluhu lokální SQLite databáze.
3. **Matematika.Sync.Core** – tato knihovna obsahuje rozhraní pro servisní třídy pro obsluhu komunikace s REST API. Dále obsahuje rozhraní pro konfigurační soubory.
4. **Matematika.Sync.Common** – tato knihovna obsahuje implementaci pro servisní třídy pro komunikaci s REST API.
5. **Matematika.Droid** – projekt pro platformu OS Android. Tento projekt implementuje uživatelské rozhraní aplikace a třídy pro obsluhu tohoto rozhraní. Tento projekt obsahuje také reference na ostatní čtyři projekty a využívá jejich funkcí pro synchronizaci a uchovávání informací.



Obr. 16. Struktura projektu mobilní aplikace.

### 3.4.2 Popis aplikace

Mobilní aplikace pro OS Android je rozdělená do dvou základních Aktivit. První aktivita slouží pro obsluhu registrace a přihlašování do aplikace. Druhá aktivita se stará o zobrazování přednášek, zobrazování a zpracovávání příkladů a obsluhu her. Většina vnitřní logiky aplikace je soustředěna právě do aktivitních tříd. Zobrazení a obsluha prvků uživatelského rozhraní je koncipována do fragmentů. Šablony a zdroje pro aplikaci jsou koncipovány do standardních XML souborů.

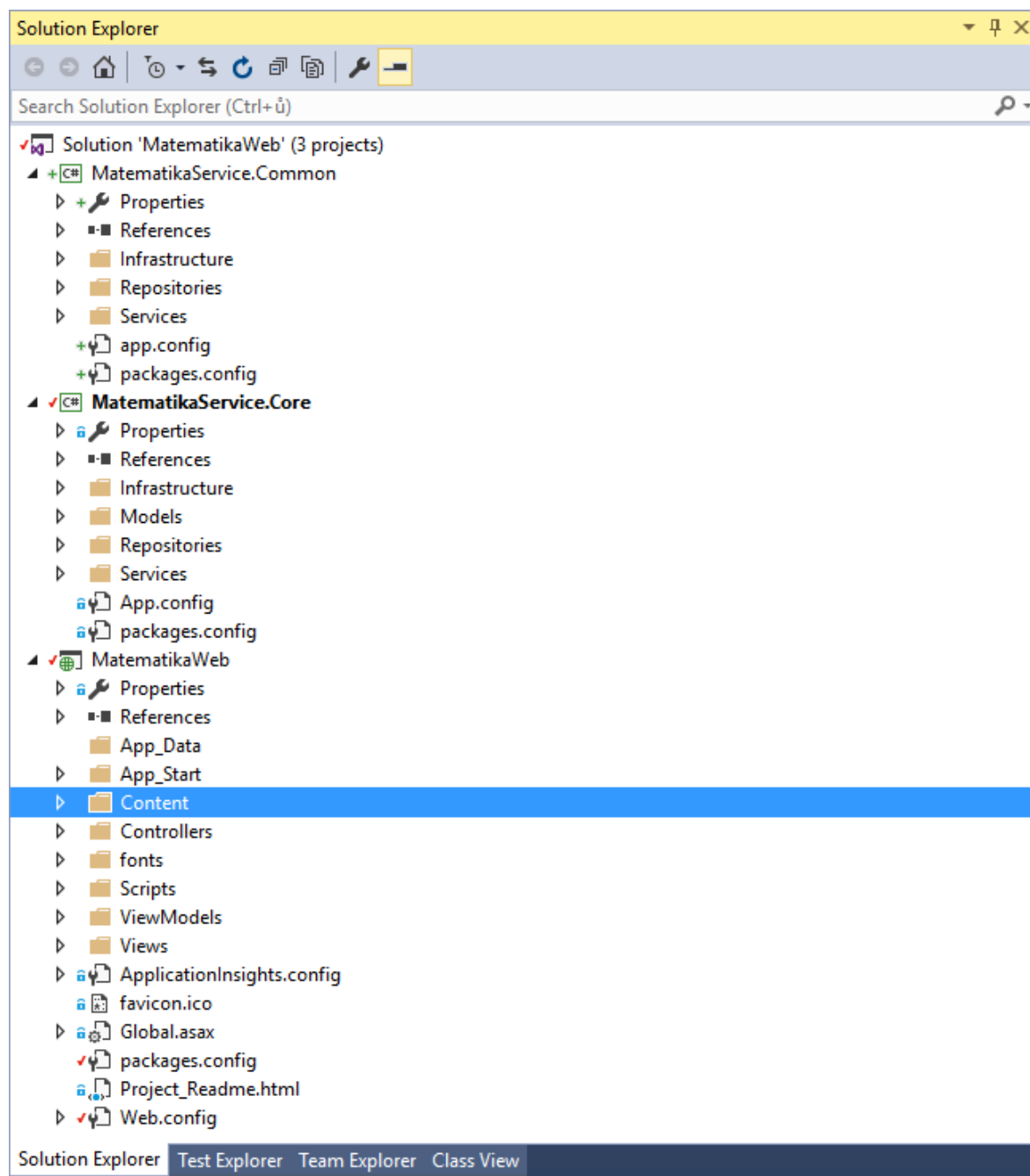
## 3.5 Webová aplikace

Tato aplikace bude sloužit výhradně pro učitele pro tvorbu obsahu učebních textů a příkladů. Webová aplikace byla vyvíjena na frameworku ASP.NET MVC 5, opět je zde použita knihovna Autofac a její rozšíření pro implementaci na MVC5 pro vkládání závislostí. Celý projekt je založený, jak už k tomu samotný Framework vybízí, na vývojovém modelu MVC Model-View-Controller. Ke komunikaci s databází jsem opět použil knihovnu Dapper.

Knihovny s rozhraními a implementacemi servisních tříd a repozitářů pro komunikaci s databází jsou již hotové v Backend aplikaci. Jedná se o projekty MatematikaService.Core a MatematikaService.Common. Proto stačilo tyto projekty nareferencovat do webového projektu.

Celá struktura aplikace je tak složena ze 3 projektů:

1. **MatematikaService.Core** – tato knihovna obsahuje třídy základních entit, rozhraní pro servisní třídy a rozhraní repozitářů pro komunikaci s databází.
2. **MatematikaService.Common** – tato knihovna obsahuje implementace servisních tříd a repozitářů pro komunikaci s databází.
3. **MatematikaWeb**– projekt obsahující kontrolery pro routování a obsluhu jednotlivých stránek, pohledy (views) psané v šablonovacím systému Razer, CSS a javascript knihovny a základní konfigurační soubory.



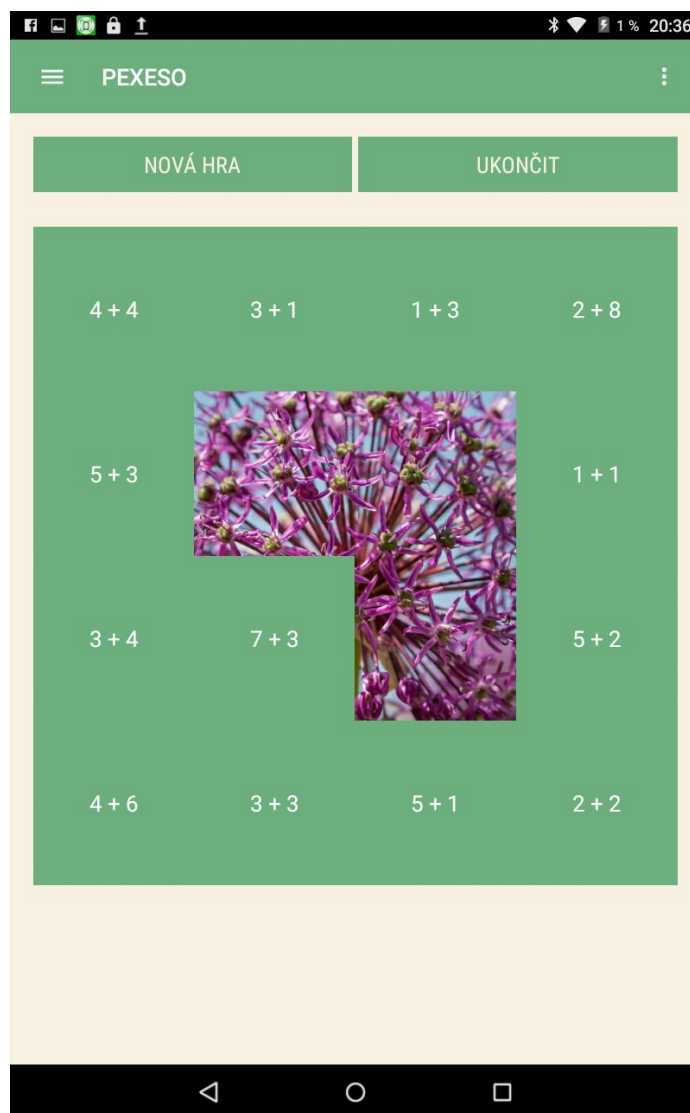
Obr. 17. Struktura webové aplikace.

Pro usnadnění prací se stylováním a programováním front-endové části aplikace jsem zakoupil šablonu Smarty postavenou na responzivní šabloně Bootstrap.

### 3.6 Graficky realizované pomůcky pro porozumění matematickým operacím

Graficky realizované pomůcky pro porozumění matematickým operacím se v aplikaci vyskytují ve dvou formách. První forma je v rukou uživatelů v rolích učitelů, jelikož díky webové aplikaci pro učitele lze do aplikace vkládat obrázky a formátovat text v přednáškách a v zadáních příkladů a umožňuje tak učitelům vytvářet dynamičtější učební texty a příklady podle svého uvážení.

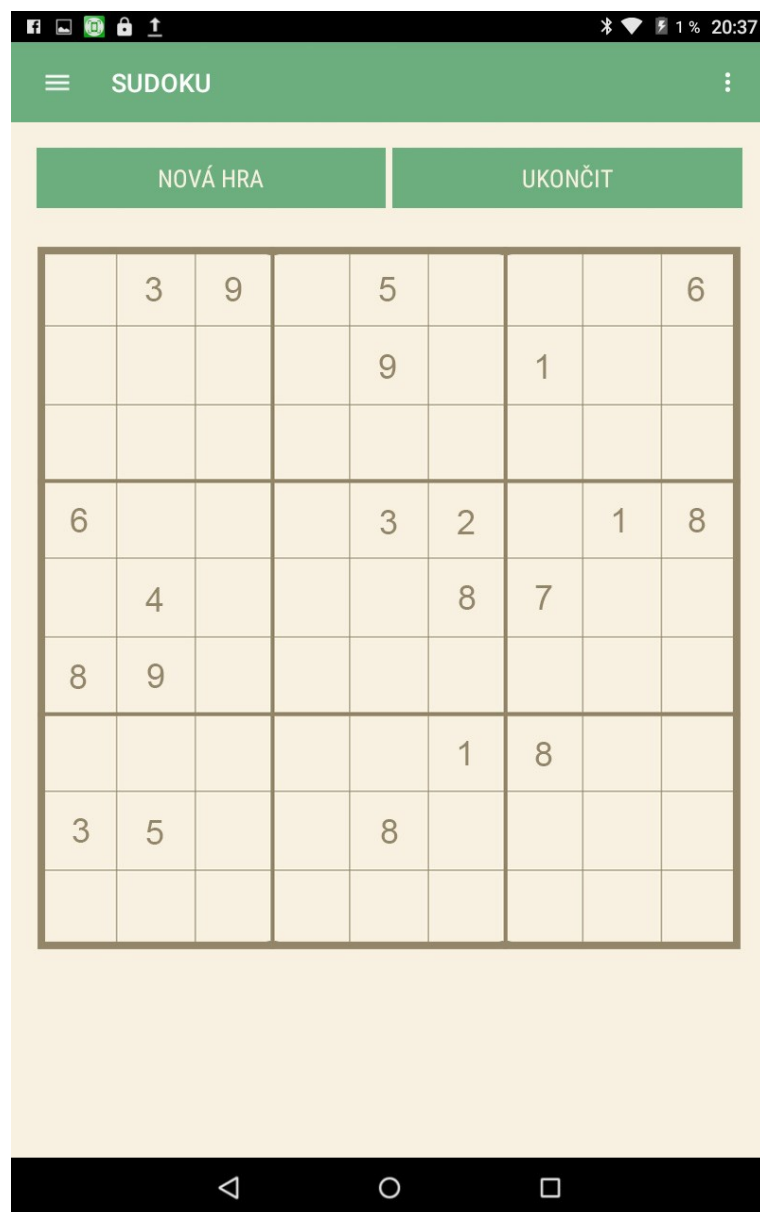
Druhou formou jsou matematické hry. Jednou z nich je matematické pexeso, při níž se vyplňují výsledky jednoduchých příkladů na základní matematické operace. Při správném vyplnění se políčko změní v část obrázku. Hra končí ve chvíli, kdy je poskládán celý obrázek. Tato hra je určena pro nejmladší studenty 1. a 2. ročníku základní školy.



Obr. 18. Hra pexeso.

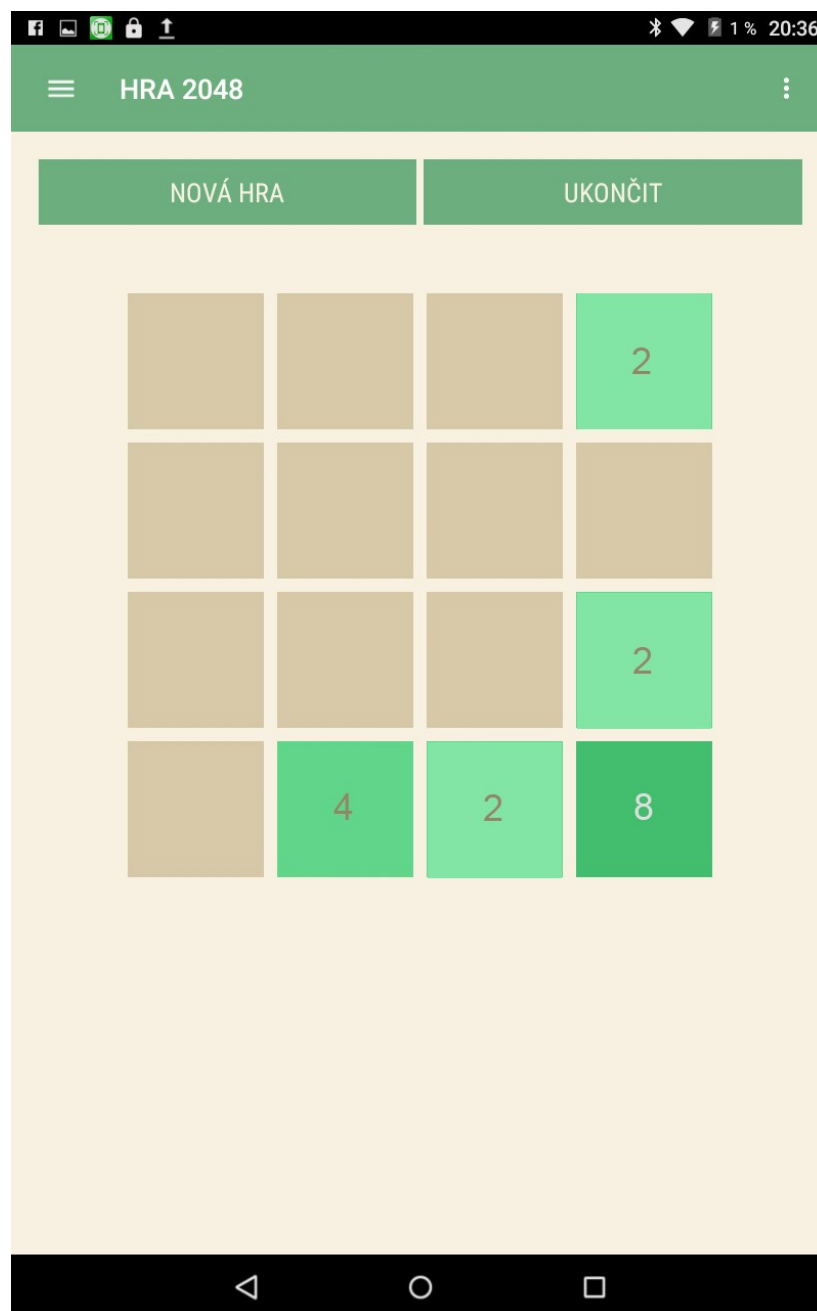
Další hrou je známá hra Sudoku, jejímž cílem je vyplnit tabulku o 9x9 políčkách čísly 1-9, přičemž každé číslo se smí v jednom řádku, sloupci a čtvercové části o velikosti 3x3 vyskytovat pouze jednou. Tato hra je vhodná pro hráče jakéhokoli věku.





Obr. 19. Hra Sudoku.

Třetí hrou je taktéž známá hra s názvem 2048. Nejznámější varianta této hry se hraje na ploše 4x4 políčka. Cílem této hry je posouvat číslice po ploše, přičemž čísla se stejnou hodnotou se spojí do jednoho a sečtou se. Po každém kroku se na ploše objeví nová číslice o hodnotě 2 nebo s menší pravděpodobností s hodnotou 4. Při spojování čísel tak vznikají mocniny dvou a cílem hry je získat políčko s hodnotou 2048. Po získání této hodnoty hráč vyhrává, nicméně může dále pokračovat ve hře. Tato hra je opět vhodná pro hráče jakéhokoli věku.



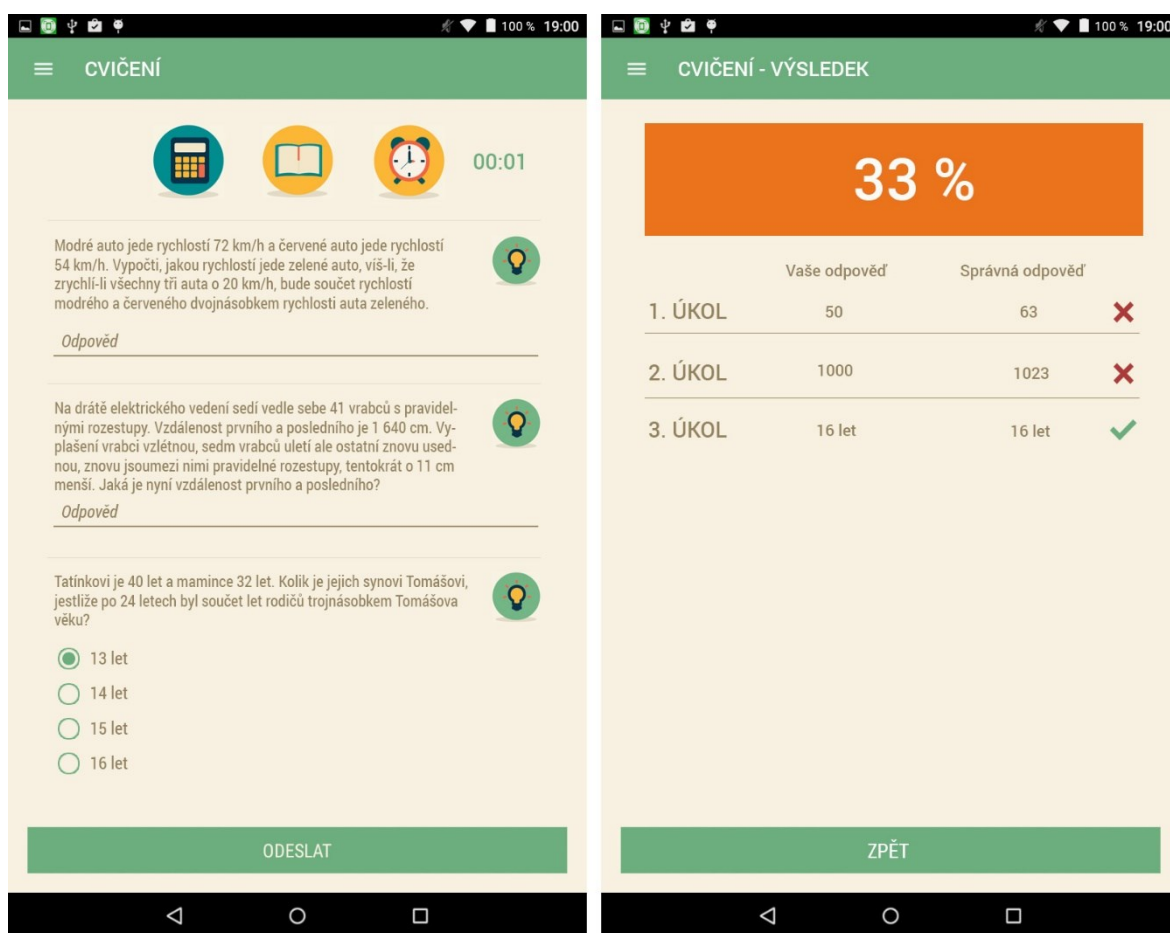
Obr. 20. Hra 2048.

### 3.7 Rozhraní testů a správa výsledků

Rozhraní testů se v aplikaci vyskytuje ve dvou formách: cvičení a úkoly. Cvičení se skládá z 10 náhodně vybraných příkladů z uživatelem vybraných témat. Tyto příklady jsou vybírány z inicializačního balíčku příkladů a dále z příkladů vytvořených učiteli pro druhou formu rozhraní testů „úkolů“. Příklady mohou být dvojího typu číselné a výběrové. Výběrové příklady se řeší výběrem jedné ze čtyř odpovědí, číselné příklady se řeší číselnou odpovědí. V rozhraní testů se navíc nachází kalkulačka, možnost zápisu poznámek např. pro

mezi-výpočty a informace o uplynulém čase. Po odeslání testu následuje vyhodnocení s celkovou úspěšností a se správnými odpověďmi.

Druhá forma rozhraní testů jsou úkoly. Úkoly mohou vytvářet učitelé, kde vytvářejí jednotlivé příklady buď přes mobilní, nebo přes webovou aplikaci. Tyto úkoly jsou následně zaslány všem studentům ve třídě. Forma řešení je stejná, rozdíl je v zadání a vyhodnocení testu. Při cvičeních se výsledky neukládají, ale slouží pouze pro osobní trénink studenta. Výsledky úkolů se naopak ukládají a odesílají přes REST API do sdílené databáze, a jsou k dispozici učitelům. Učitel tak má přehled o úspěšnosti nejen jednotlivých studentů ale i třídy jako celku.



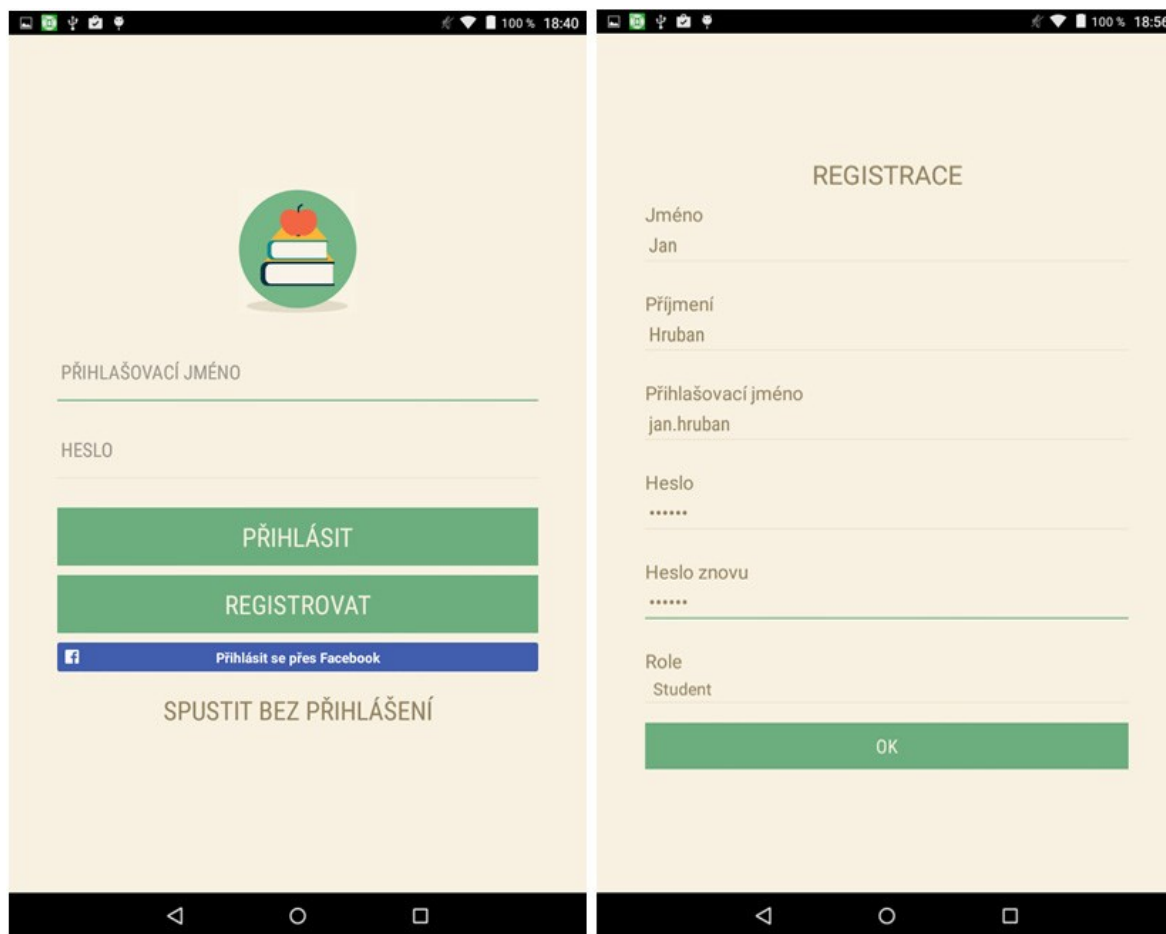
Obr. 21. Rozhraní testů a výsledků.

### 3.8 Správa uživatelských účtů

Mobilní aplikace umožňuje vytvářet a přihlašovat se pomocí uživatelských účtů. Uživatelské účty jsou dvojího typu: Student a učitel. Při registraci učitele, učitel zároveň vytváří i třídu, do které se budou studenti zapisovat. Pro zápis do třídy slouží klíč, který učitel studentům

sdělí. Tento klíč je nutné znát při registraci do aplikace jako student. Uživatelské účty slouží především k zadávání úkolů studentům sdružených ve třídách, vytvářet pro studenty učební texty a ukládat výsledky testů.

Registrace do aplikace je možné přes standardní formulář se zadáním jména, příjmení, uživatelského jména a hesla. Další možností je přihlášení pomocí Facebooku pomocí Facebook SDK.



Obr. 22. Přihlášení a registrace do aplikace.

## 4 UŽIVATELSKÁ DOKUMENTACE

Tato část práce slouží jako uživatelská dokumentace mobilní aplikace pro studenty a učitele a dokumentace webové aplikace pro učitele.

### 4.1 Uživatelská dokumentace pro studenty a žáky

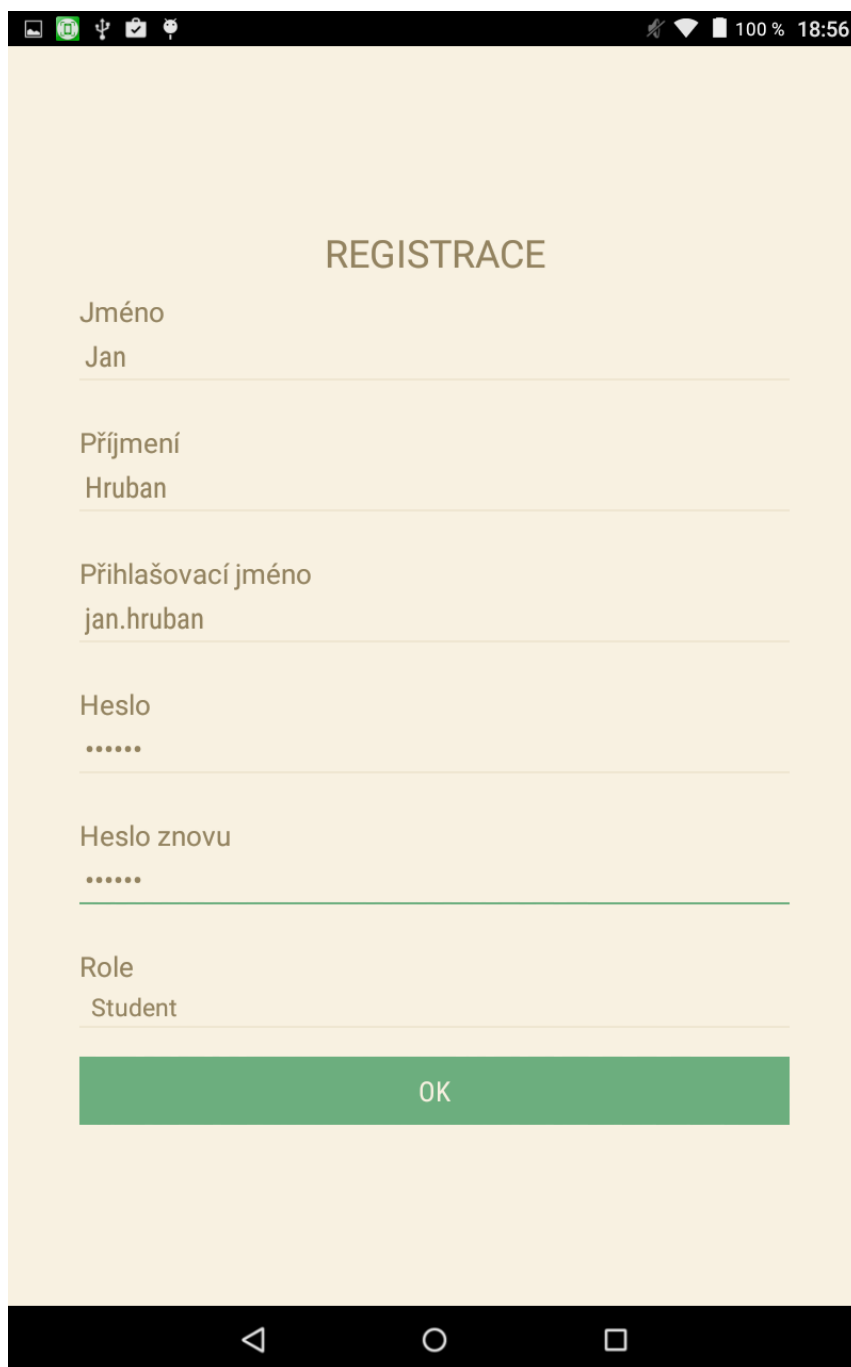
Pro přihlášení do aplikace pod studentským nebo učitelským účtem je potřeba mít na zařízení aktivní připojení k internetu.

#### 4.1.1 Registrace do aplikace

Pro založení studentského účtu jsou k dispozici dvě možnosti. Založení standardního aplikačního účtu s přihlašovacím jménem a heslem, nebo prostřednictvím sociální sítě Facebook.

##### *Standardní registrace*

Pro založení účtu prostřednictvím standardní registrace, klikněte na přihlašovací obrazovce na tlačítko „Registrovat“. Objeví se obrazovka s několika políčky. Vyplňte vaše jméno, příjmení, přihlašovací jméno, heslo a heslo znovu. Ve výběrovém políčku vyberte roli „Student“ a potvrďte registraci. Pokud registrace není úspěšná, opravte příslušný údaj a registraci opakujte.



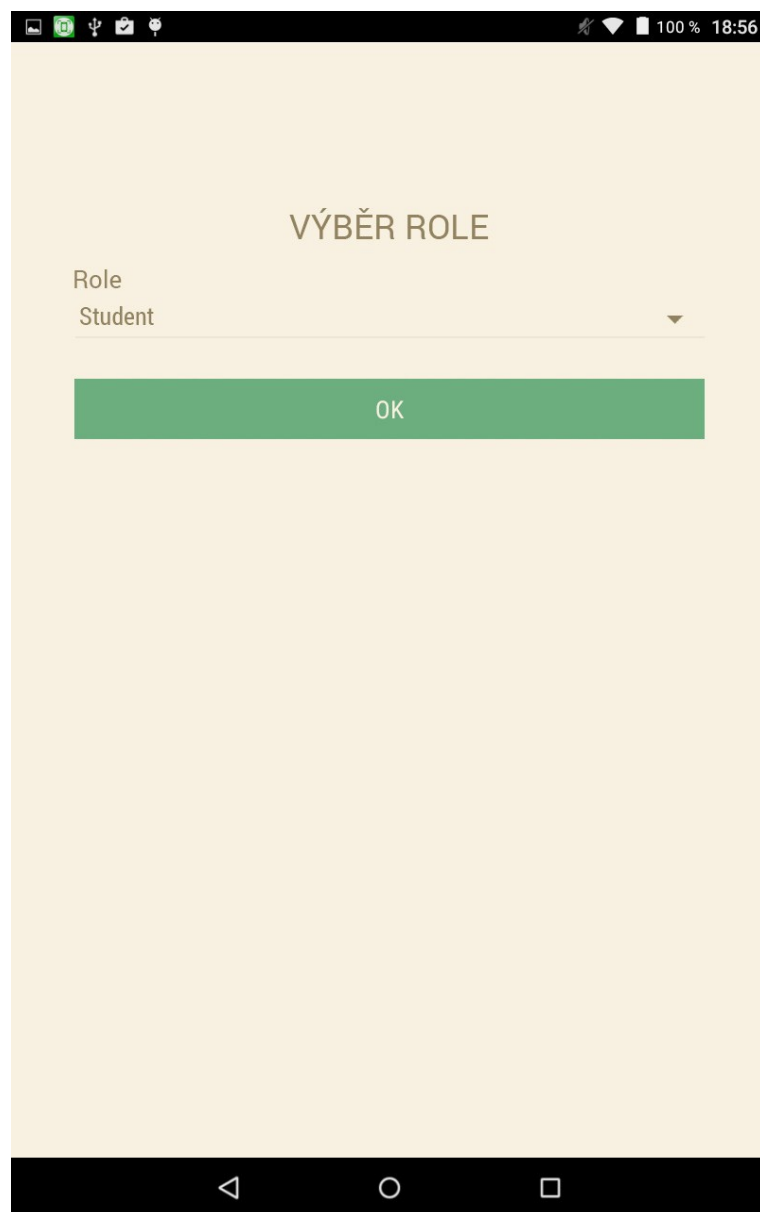
The image shows a mobile application registration screen with a light beige background. At the top, there is a status bar with icons for signal, Wi-Fi, battery (100%), and time (18:56). The title 'REGISTRACE' is centered in a dark grey font. Below the title are several input fields with labels and pre-filled text: 'Jméno' (Jan), 'Příjmení' (Hruban), 'Přihlašovací jméno' (jan.hruban), 'Heslo' (masked with six dots), 'Heslo znovu' (masked with six dots), and 'Role' (Student). A green button labeled 'OK' is positioned at the bottom of the form. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Obr. 23. Obrazovka registrace.

V druhém kroku registrace vyplňte do políčka „Název třídy“ název nebo část názvu příslušné třídy, do které se chcete přihlásit a následně třídu vyberte z nabídky. Pro přihlášení do třídy a dokončení registrace vyplňte heslo, které vám sdělí učitel, a potvrďte. Po tomto kroku dojde k dokončení registrace a prvnímu přihlášení do aplikace.

### *Facebook registrace*

Pro založení účtu prostřednictvím sociální sítě Facebook klikněte na přihlašovací obrazovce na modré tlačítko „Přihlásit přes Facebook“. Pokud máte na zařízení aktivní Facebookový účet, dojde k automatické synchronizaci základních údajů, pokud na zařízení aktivní Facebookový účet není, zobrazí se přihlašovací obrazovka naFacebook. Po přihlášení dojde k synchronizaci základních údajů. V následujícím kroku si uživatel vybírá roli. Vyberte roli „Student“ a potvrďte. Následuje druhý krok registrace: výběr třídy. Do políčka „Název třídy“ zadejte název třídy nebo jeho část a vyberte tuto třídu z nabídky. Následně vyplňte klíč pro přihlášení do třídy a potvrďte. Registrace je dokončena a proběhne první přihlášení do aplikace.



Obr. 24. Obrazovka výběru role.

#### 4.1.2 Přihlášení do aplikace

Pokud byl na zařízení přidán uživatelský účet a uživatel se z aplikace neodhlásil, probíhá přihlášení automaticky. V případě prvního přihlášení do aplikace na zařízení, nebo přihlášení po předchozím odhlášení vyplňte na přihlašovací obrazovce uživatelské jméno a heslo a stiskněte tlačítko „Přihlásit“ nebo klikněte na tlačítko „Přihlásit se přes Facebook“, pokud máte uživatelský účet založený přes Facebook. Proběhne vzdálené ověření přihlašovacích údajů a dojde k přihlášení do aplikace.

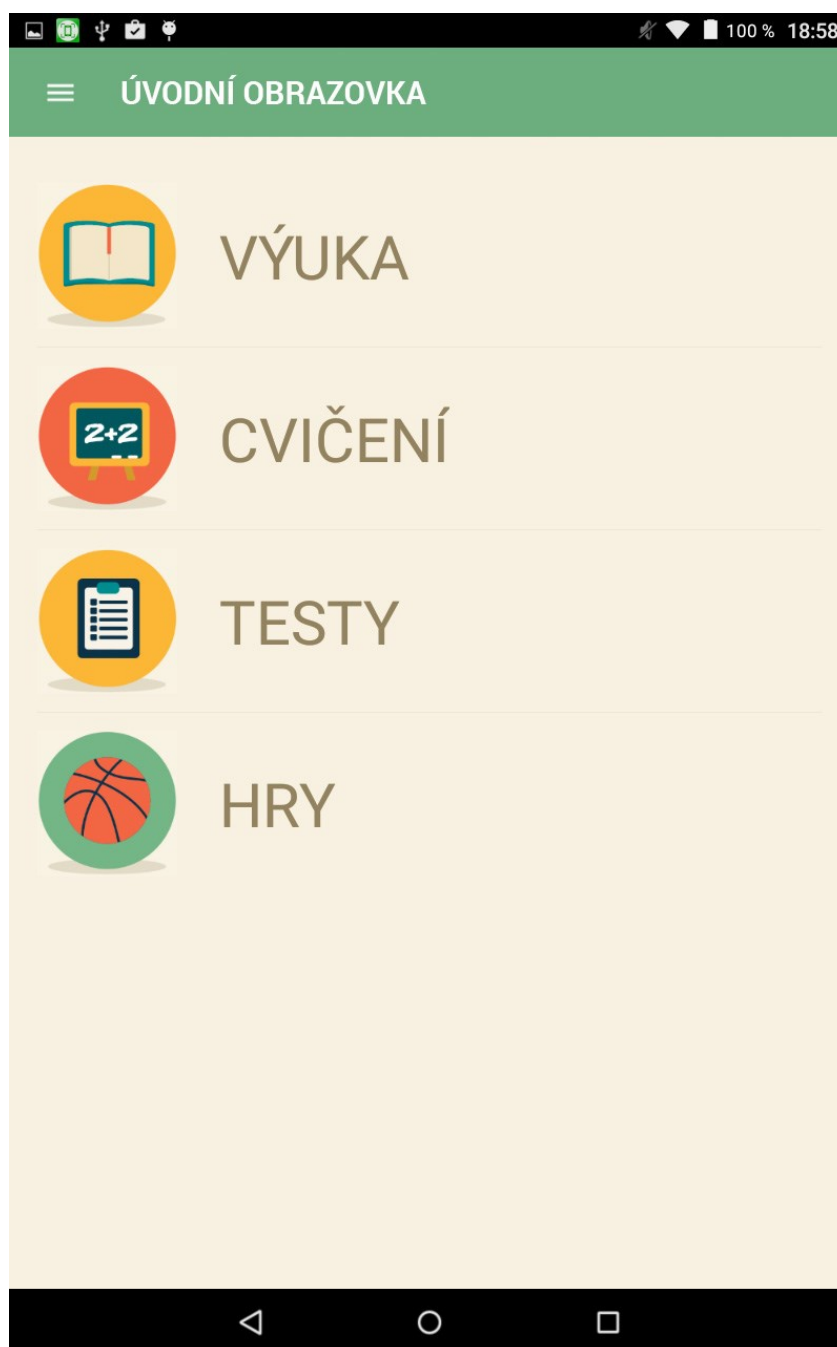




Obr. 25. Přihlašovací obrazovka.

#### 4.1.3 Úvodní obrazovka

Bezprostředně po přihlášení do aplikace v případě aktivního připojení na internet dojde ke spuštění synchronizace databáze příkladů, úkolů a učebních textů. Vyčkejte, dokud synchronizace neskončí (v případě prvního přihlášení na zařízení, může synchronizace trvat trochu déle).



Obr. 26. Úvodní obrazovka pro studenty.

V horní části aplikace se nachází zelený informační řádek, který obsahuje nadpis pro jednotlivé obrazovky a také tlačítko pro vyvolání navigačního menu. Tato část je pro všechny obrazovky stejná. Na úvodní obrazovce jsou dále čtyři tlačítka:

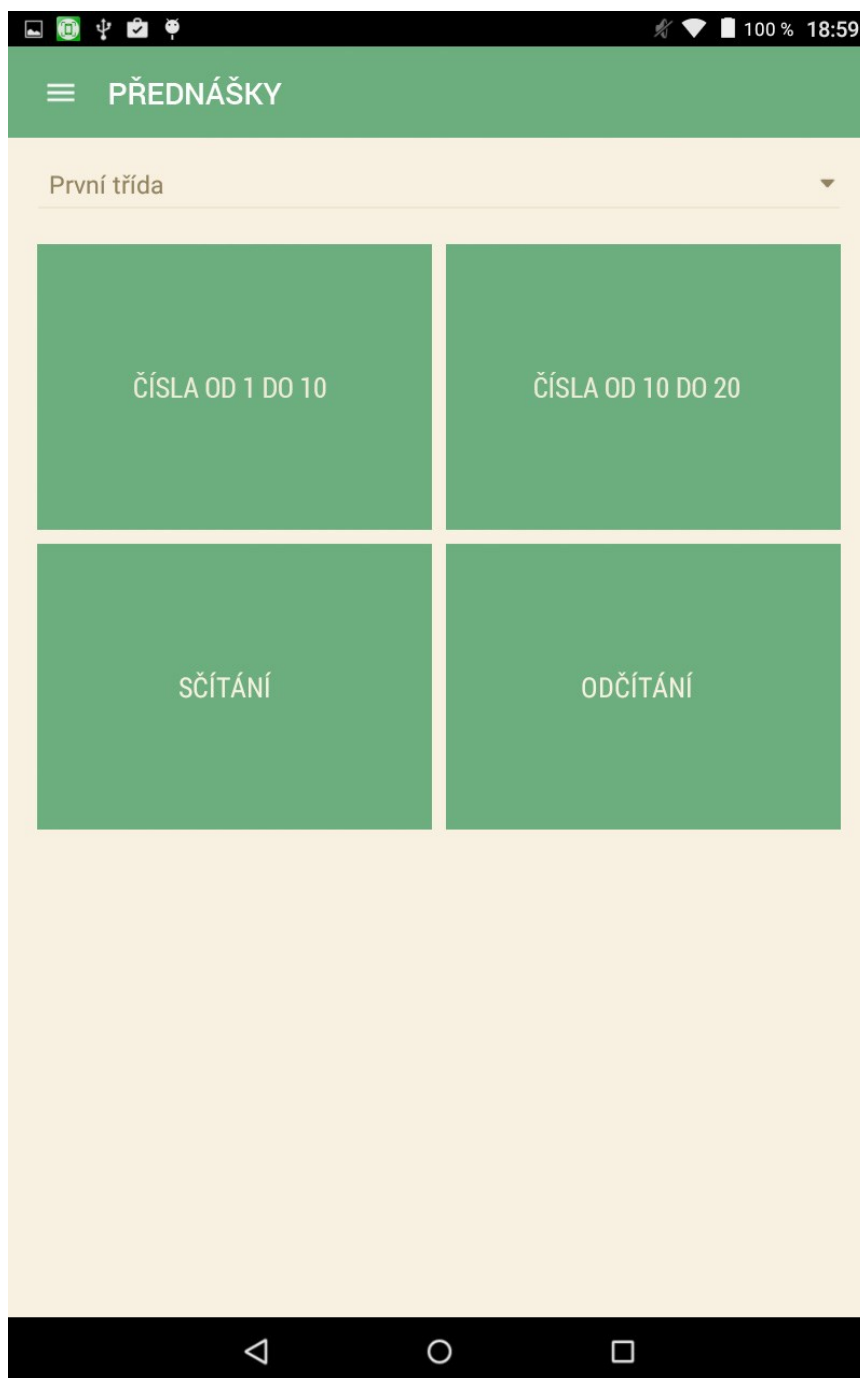
1. Výuka – stisknutím přejdete na obrazovku pro výběr učebního textu pro vybrané téma
2. Cvičení – stisknutím přejdete na obrazovku pro výběr cvičného testu

3. Testy – stisknutím přejdete na přehled splněných a nově zadaných úkolů.
4. Hry – stisknutím přejdete na přehled her.

#### **4.1.4 Přednášky**

V horní části obrazovky se nachází výběrový seznam pro filtrování témat podle jednotlivých ročníků. Tlačítka pro výběr tématu se objeví ve frontě zelených čtverců. Po kliknutí na vybrané téma se dostanete na detail přednášky.

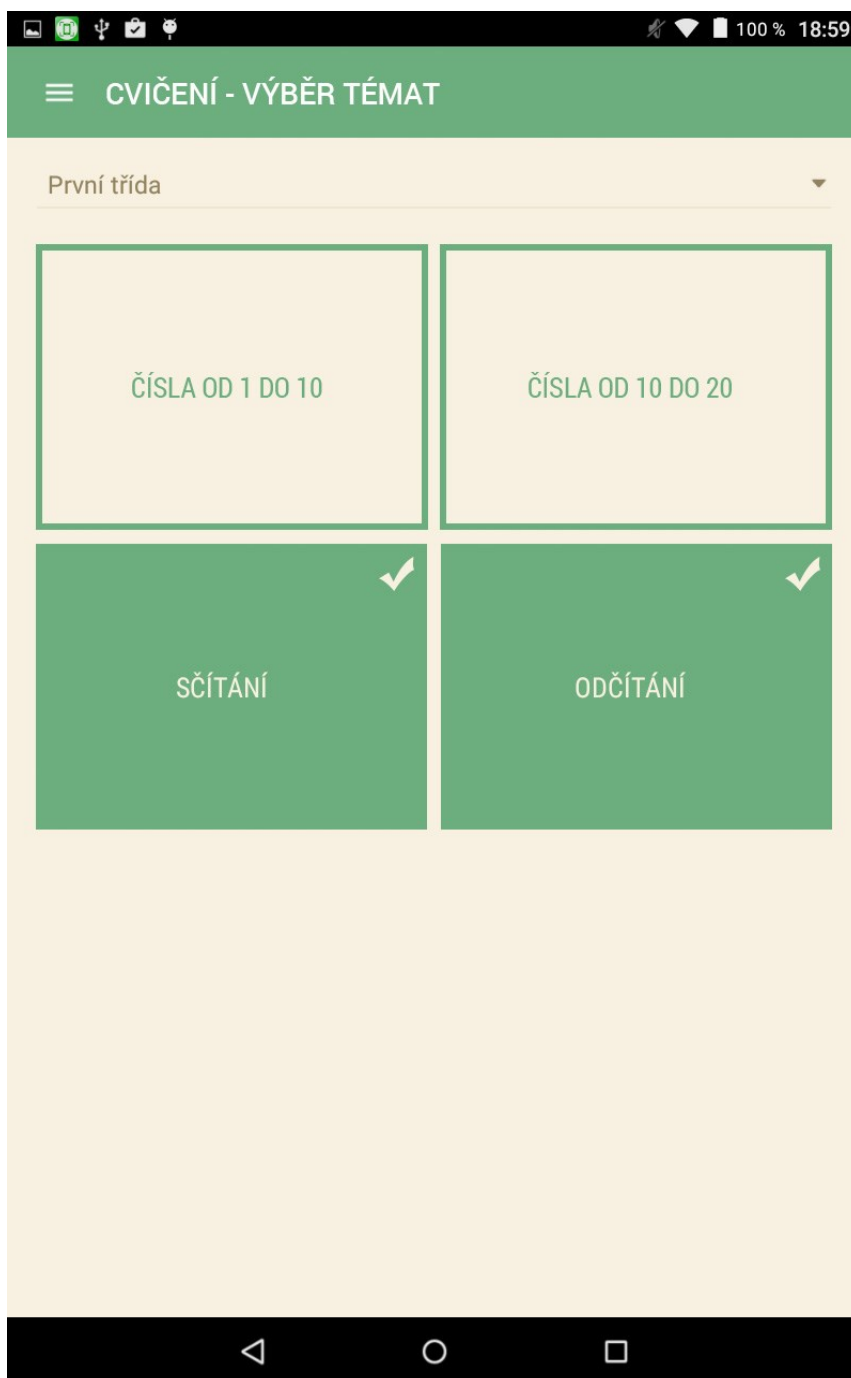
Na detailu přednášky je už samotný učební text univerzální nebo zpracovaný vyučujícím. Pro navigaci zpět do výběru témat můžete použít tlačítko ve spodní části obrazovky.



Obr. 27. Obrazovka výběru přednášek.

#### 4.1.5 Cvičení

V horní části obrazovky se nachází výběrový seznam pro filtrování témat podle jednotlivých ročníků. Jednotlivá témata vybírejte kliknutím na příslušné téma, vybrané téma se podbarví zeleně. Po zvolení témat stiskněte tlačítko „Začít test“. Následně bude náhodně vybráno 10 příkladů z vybraných témat.



Obr. 28. Obrazovka výběru příkladů pro cvičení.

Na obrazovce detailu testu se v horní části nachází tři tlačítka.

- Tlačítko s obrázkem kalkulačky slouží k vyvolání kalkulátoru.
- Tlačítko s obrázkem knihy slouží k vyvolání poznámkového plátna.
- Tlačítko budíku je informativní a ukazuje uplynulý čas.

Pod těmito tlačítky se již nacházejí příklady. Příklady mohou být dvojího typu:

1. Číselný příklad – zde odpovídáte ve formě číselných odpovědí.

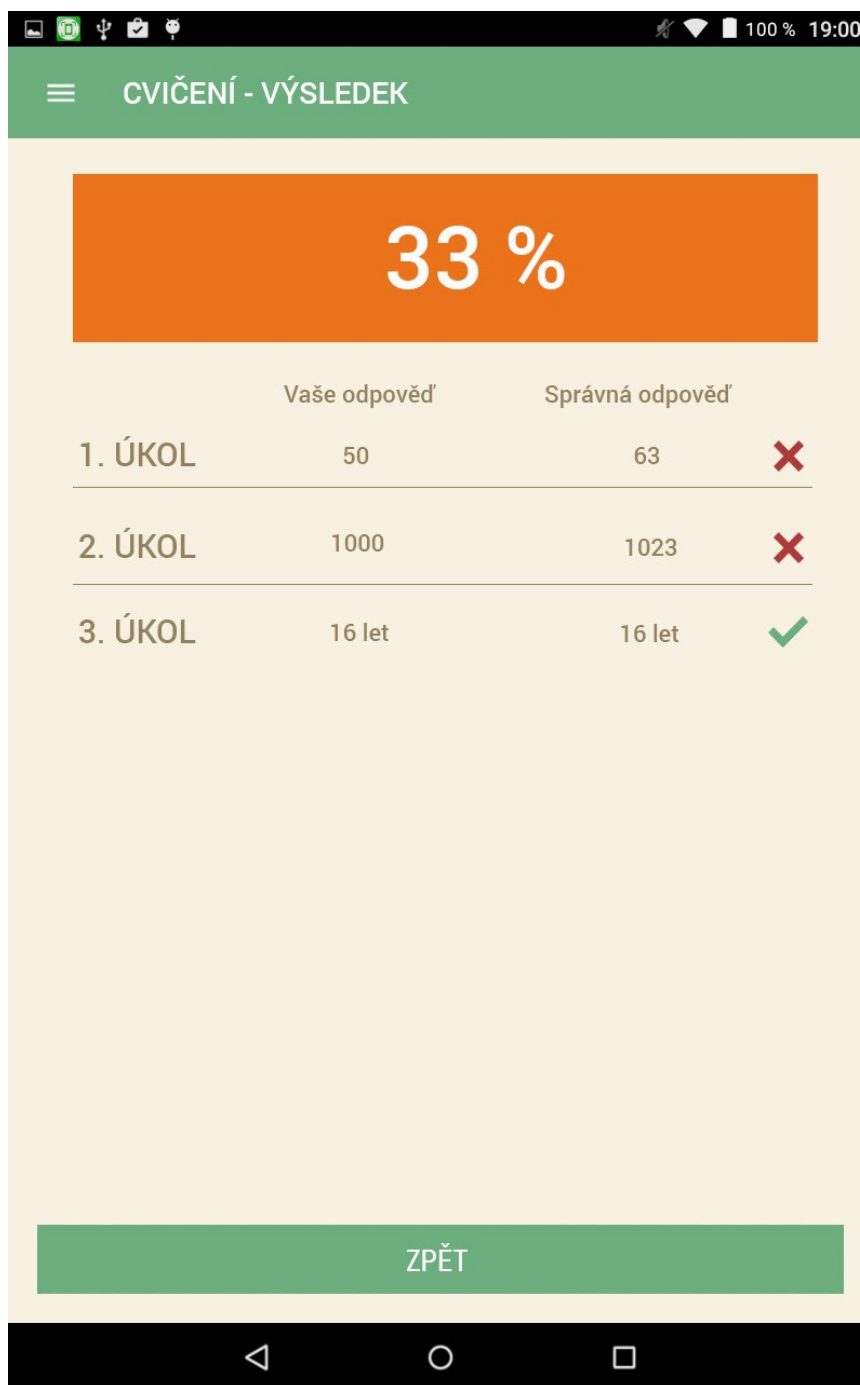
## 2. Výběrový příklad – zde odpověď vybíráte ze čtyř možností.

U každého příkladu se nachází slovní zadání a tlačítko s obrázkem žárovky. Po kliknutí na toto tlačítko (pokud je aktivní) se zobrazí rada učitele.



Obr. 29. Obrazovka testu.

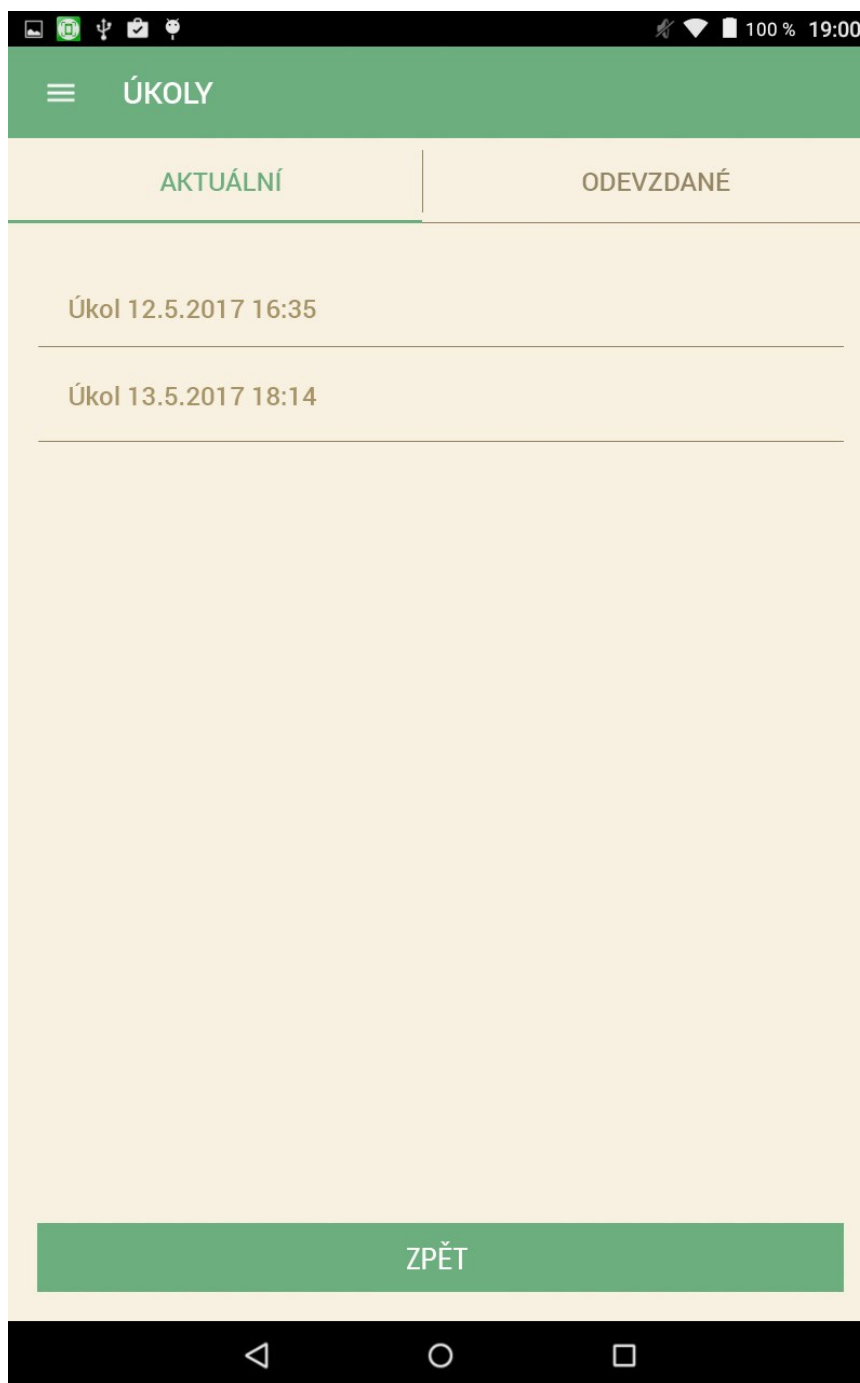
Po vyplnění odpovědí stisknete tlačítko „Dokončit test“ v dolní části obrazovky. Na následující obrazovce je zobrazen procentuálně úspěch v testu a přehled vašich a správných odpovědí. Ve spodní části obrazovky se nachází tlačítko pro návrat na úvodní obrazovku.



Obr. 30. Obrazovka výsledku testu.

#### 4.1.6 Úkoly

V horní části obrazovky se nachází přepínač pro přepnutí mezi aktuálními a odevzdanými úkoly. Při kliknutí na dokončený úkol se zobrazí procentuální výsledek a vyhodnocení jednotlivých odpovědí.



Obr. 31. Obrazovka výběru úkolů.

Při kliknutí na aktuální úkol je vyvolána obrazovka s příklady. Stejně jako u příkladů jsou zde tlačítka kalkulačka, poznámky a ukončení. Tlačítka kalkulačka a poznámky ale nemusí být aktivní, pokud učitel při zadání úkolů tak rozhodne. Po vyplnění odpovědí příkladů stisknete tlačítko „Dokončit“. Zobrazí se procentuální výsledek, a přehled jednotlivých odpovědí. Výsledek je zároveň odeslán a přístupný učiteli. Pro návrat na úvodní obrazovku klikněte na tlačítko ve spodní části obrazovky

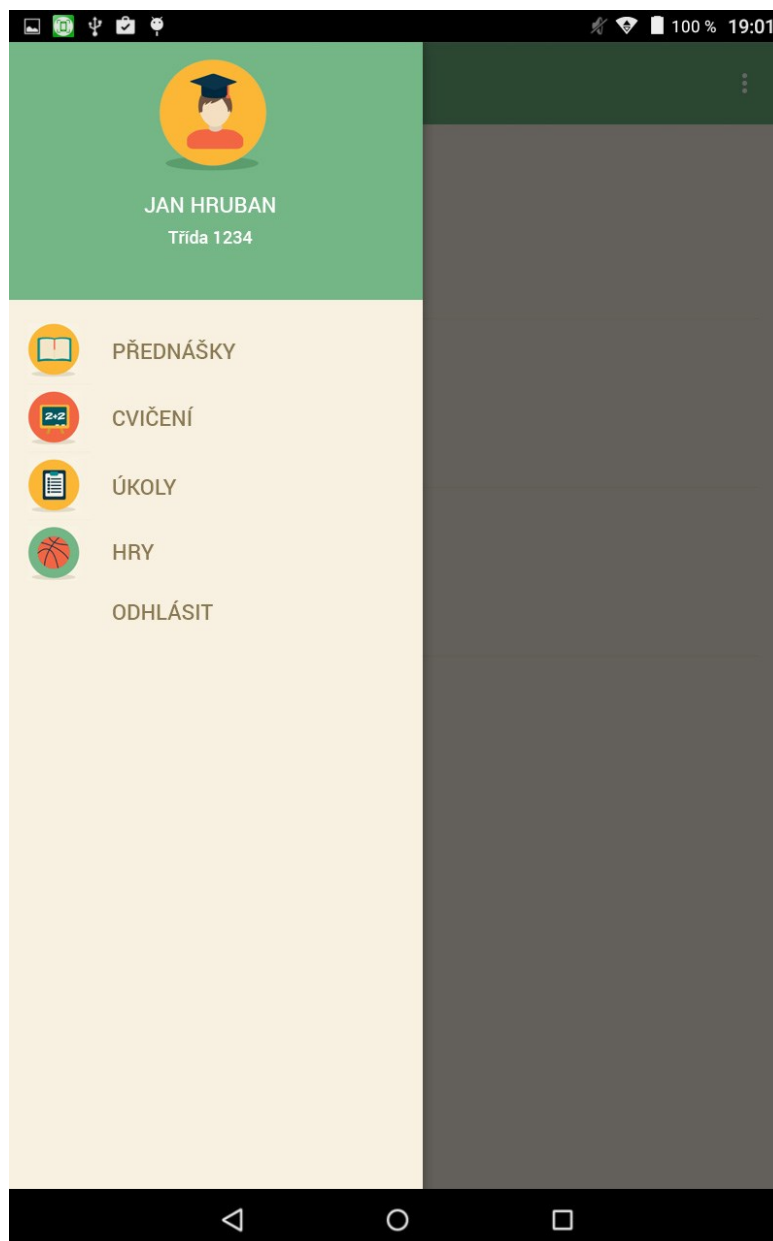


#### 4.1.7 Hry

V přehledu her se nachází tlačítka pro tři hry: Matematické pexeso, sudoku, 2048.

#### 4.1.8 Odhlášení z aplikace

Pro odhlášení z aplikace vyvolejte navigační menu v horní části obrazovky a následně v navigačním menu klikněte na „Odhlásit“.



Obr. 32. Obrazovka hlavního menu aplikace.

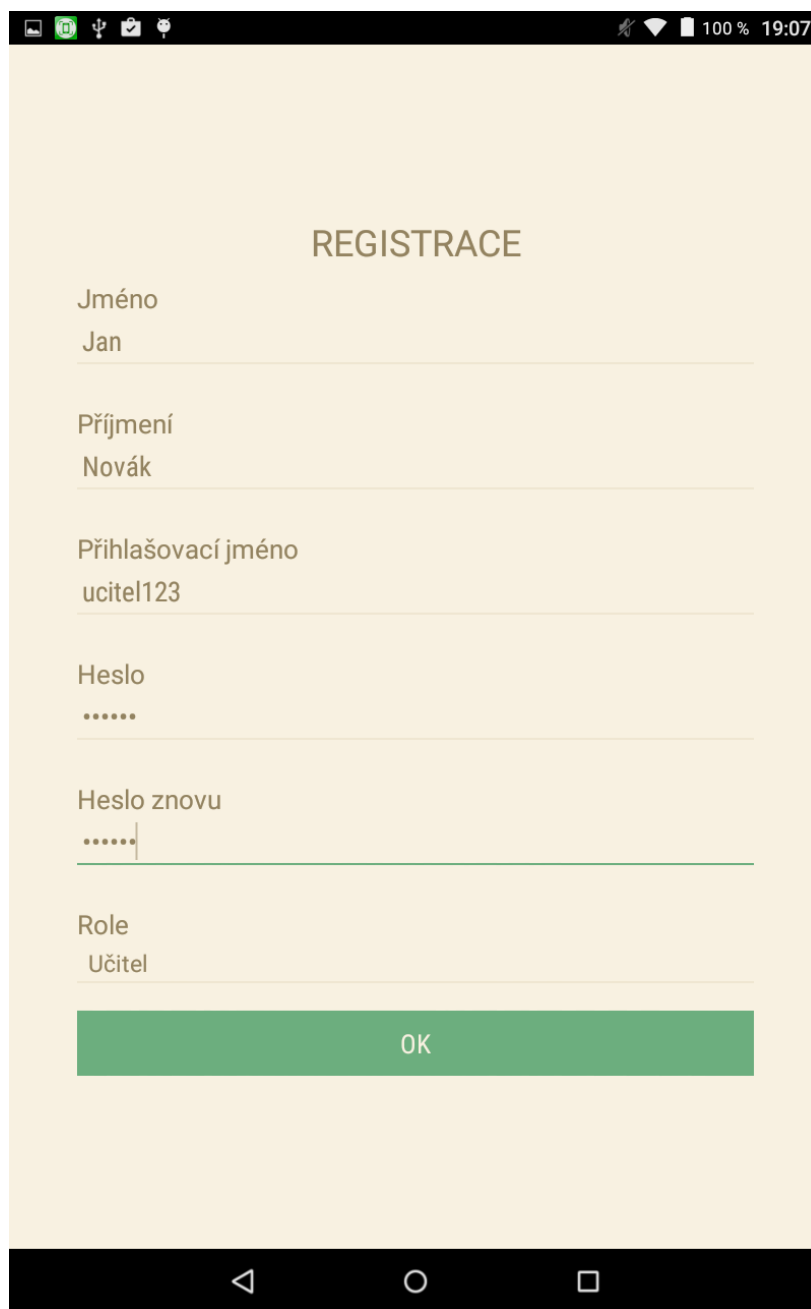
## 4.2 Uživatelská dokumentace pro učitele

Pro přihlášení do aplikace pod studentským nebo učitelským účtem je potřeba mít na zařízení aktivní připojení k internetu.

### 4.2.1 Registrace učitele

#### *Standardní registrace*

Pro založení účtu učitele prostřednictvím standardní registrace, klikněte na přihlašovací obrazovce na tlačítko „Registrovat“. Objeví se obrazovka s několika políčky. Vyplňte vaše jméno, příjmení, přihlašovací jméno, heslo a heslo znovu. Ve výběrovém políčku vyberte roli „Učitel“ a potvrďte registraci. Pokud registrace není úspěšná, opravte příslušný údaj a registraci opakujte.



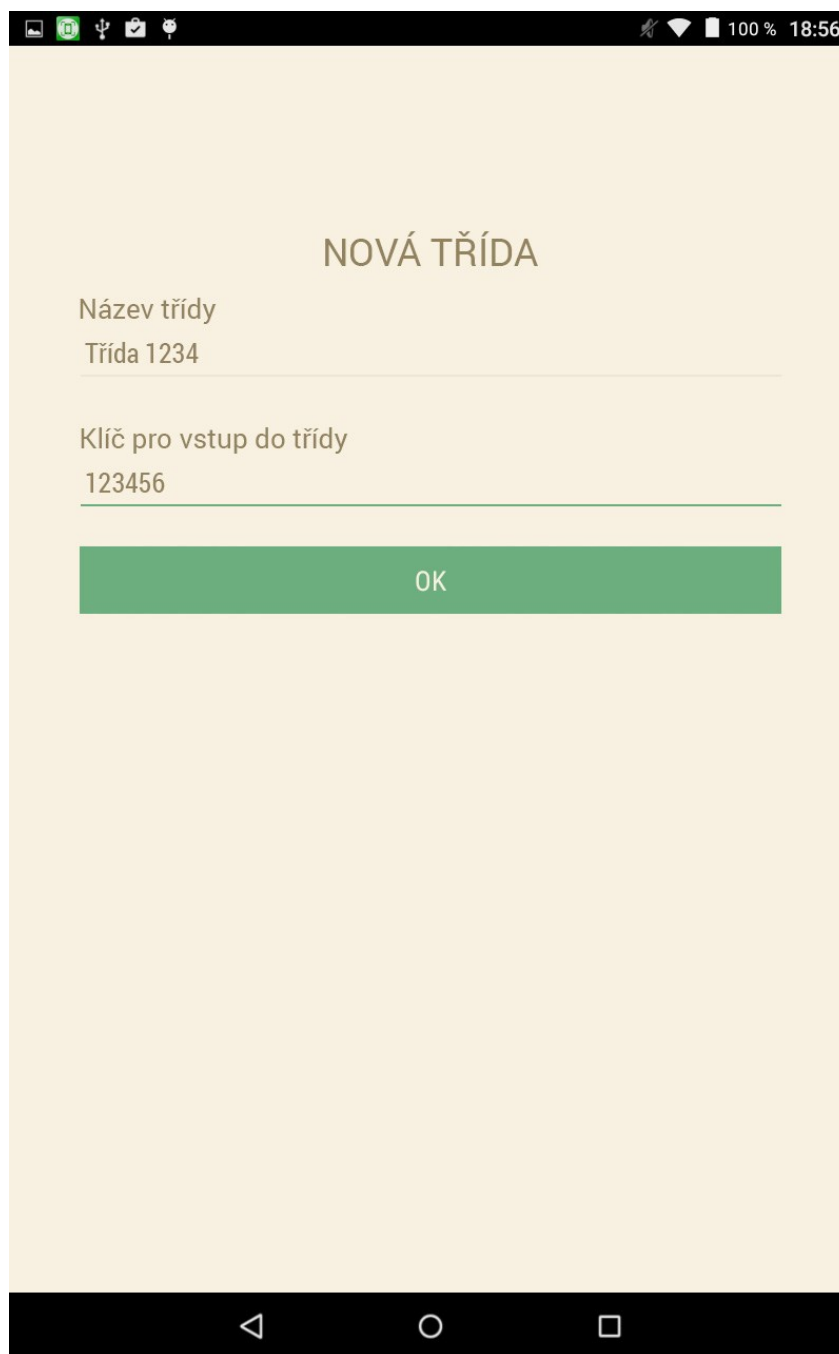
The image shows a mobile application registration screen with a light beige background. At the top, the title 'REGISTRACE' is centered in a dark grey font. Below the title, there are several input fields with labels and pre-filled text:

- Jméno**: Input field containing 'Jan'.
- Příjmení**: Input field containing 'Novák'.
- Přihlašovací jméno**: Input field containing 'ucitel123'.
- Heslo**: Input field with five dots representing a password.
- Heslo znovu**: Input field with five dots and a cursor, for re-entering the password.
- Role**: Input field containing 'Učitel'.

At the bottom of the form is a large green button with the text 'OK' in white. The top of the screen shows a black status bar with icons for signal, Wi-Fi, battery (100%), and time (19:07). The bottom of the screen shows a black navigation bar with three white icons: a triangle, a circle, and a square.

Obr. 33. Obrazovka registrace.

V druhém kroku registrace vyplňte do políčka třída název třídy a do políčka „Klíč pro vstup do třídy“ vyplňte heslo, prostřednictvím kterého se budou studenti do této třídy přihlašovat. Po založení třídy dojde k dokončení registrace a prvnímu přihlášení do aplikace.



Obr. 34. Obrazovka založení třídy.

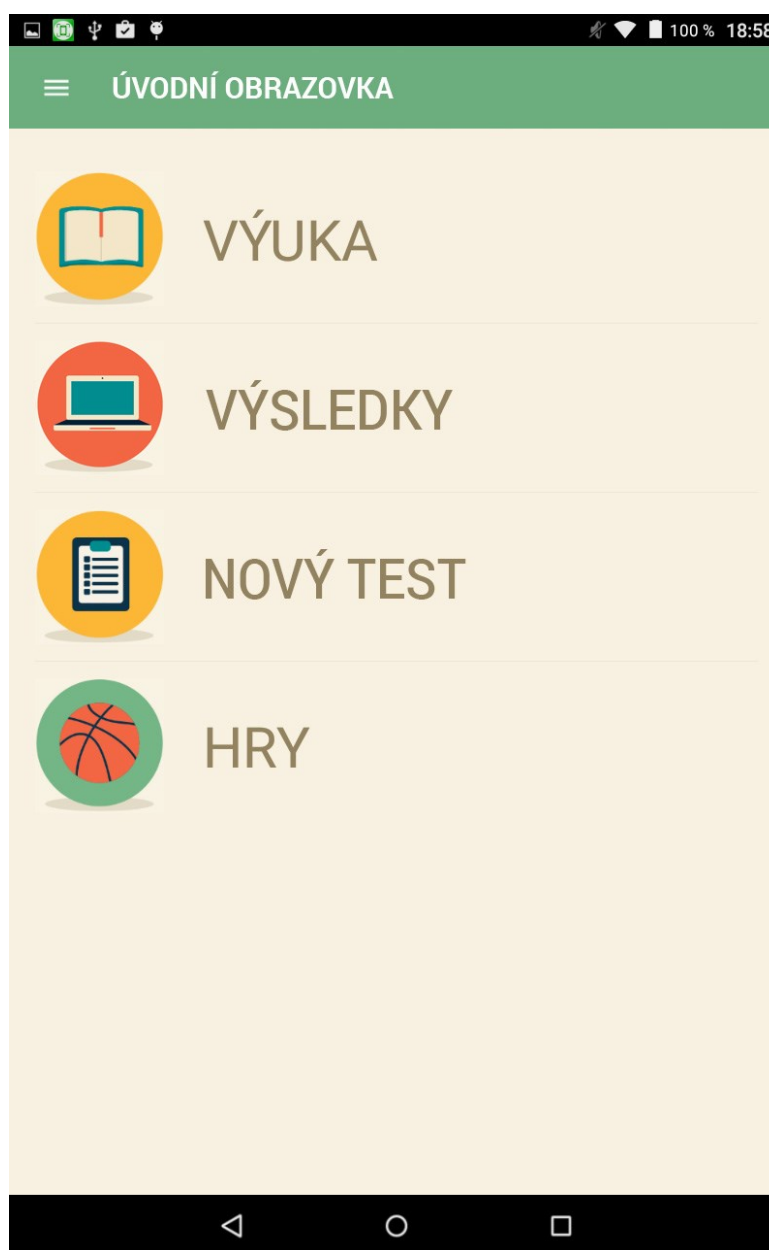
### ***Facebook registrace***

Pro založení účtu prostřednictvím sociální sítě Facebook klikněte na přihlašovací obrazovce a modré tlačítko „Přihlásit se přes Facebook“. Pokud máte na zařízení aktivní Facebookový účet, dojde k automatické synchronizaci základních údajů, pokud na zařízení aktivní Facebookový účet není, zobrazí se přihlašovací obrazovka na Facebook. Po přihlášení dojde k synchronizaci základních údajů. V následujícím kroku si uživatel vybírá roli. Vyberte roli

„Učitel“ a potvrďte. Následuje druhý krok registrace: založení třídy. Do políčka třída zadejte název třídy. Následně vyplňte klíč, prostřednictvím kterého se budou studenti přihlašovat do této třídy, a potvrďte. Registrace je dokončena a proběhne první přihlášení do aplikace.

#### 4.2.2 Úvodní obrazovka

Bezprostředně po přihlášení do aplikace v případě aktivního připojení na internet dojde ke spuštění synchronizace databáze příkladů, úkolů a učebních textů. Vyčkejte, dokud synchronizace neskončí (v případě prvního přihlášení na zařízení, může synchronizace trvat trochu déle).



Obr. 35. Úvodní obrazovka pro učitele.

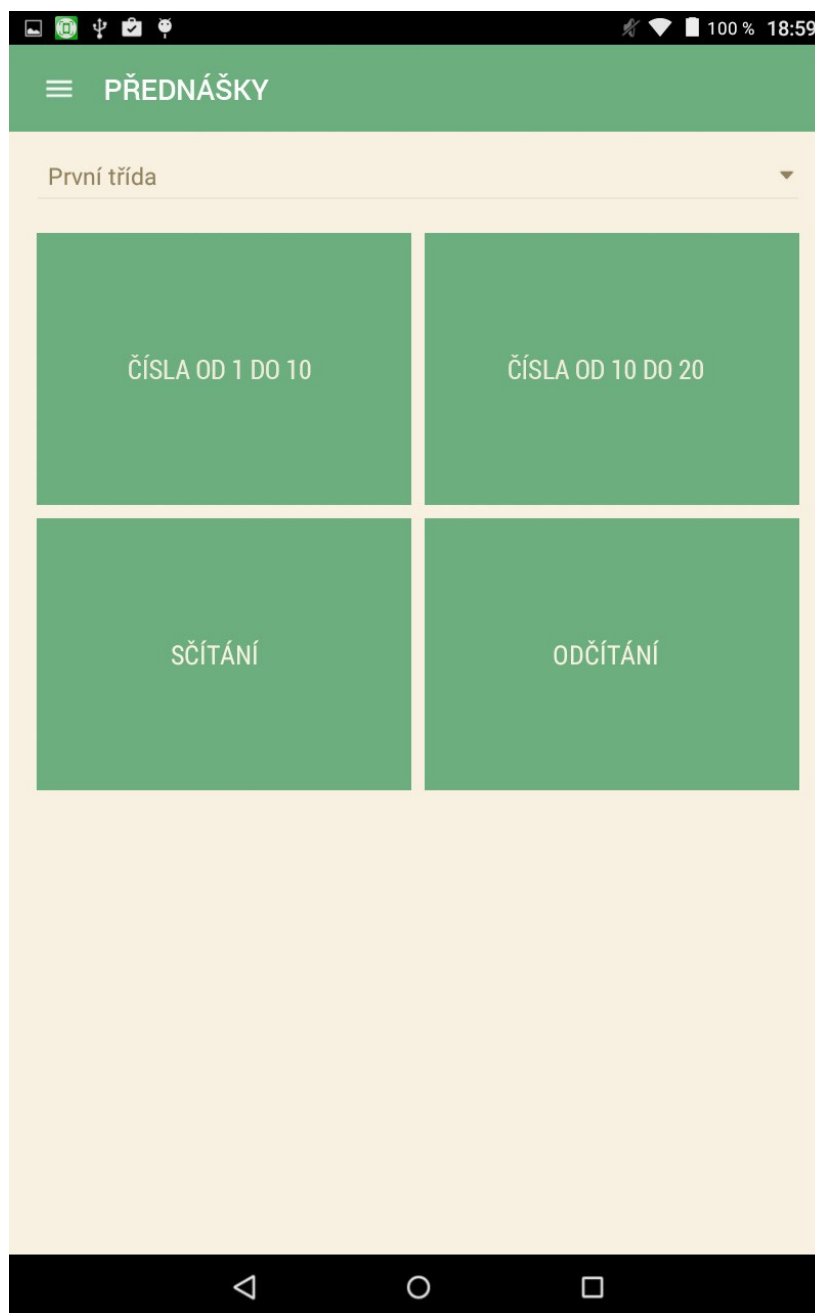
Na úvodní obrazovce pro učitele se nachází čtyři tlačítka:

1. Výuka – slouží pro přehled přednášek
2. Výsledky – přehled výsledků jednotlivých studentů z úkolů
3. Nový test - slouží pro založení úkolu pro studenty ve třídě
4. Hry – obsahuje hry Sudoku, 2048 a výpočetní pexeso

#### **4.2.3 Přednášky**

V horní části obrazovky se nachází výběrový seznam pro filtrování témat podle jednotlivých ročníků. Tlačítka pro výběr tématu se objeví ve frontě zelených čtverců. Po kliknutí na vybrané téma se dostanete na detail přednášky. Dále se zde nachází tlačítko „Přidat přednášku“. Po kliknutí na toto tlačítko dojde k vyvolání webového klienta a přesměrování na webovou aplikaci, prostřednictvím které je možné upravovat učební texty třídy.

Na detailu přednášky je už samotný učební text univerzální nebo zpracovaný vyučujícím. Pro navigaci zpět do výběru témat můžete použít tlačítko ve spodní části obrazovky.



Obr. 36. Obrazovka přehledu přenášek.

#### 4.2.4 Nový úkol

Na obrazovce je přehled přidávaných příkladů. V horní části obrazovky se nachází tlačítko „Přidat příklad“. Než bude možné založit nový úkol, je potřeba alespoň jeden příklad přidat. Jsou zde také dva přepínače, které slouží k povolení nebo zakázání kalkulačky a poznámek v úkolu (úkol může sloužit i jako test).

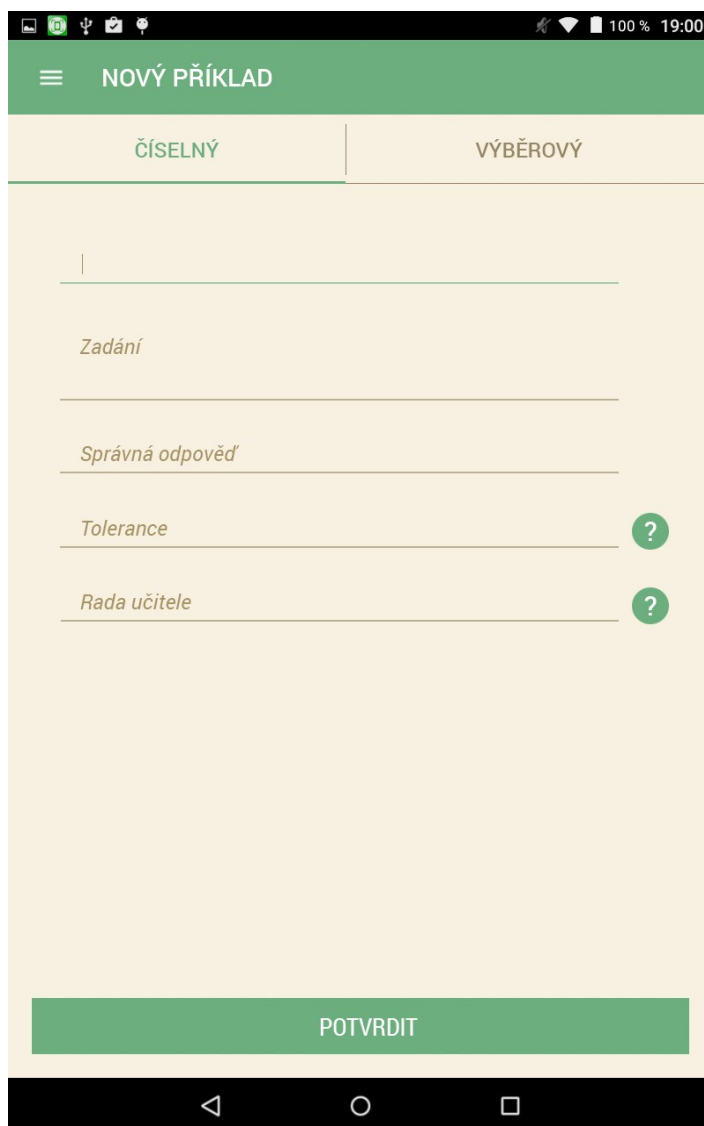
Na obrazovce nový příklad se v horní části nachází přepínač mezi číselným a výběrovým zadáním příkladu. Při zakládání číselného příkladu jsou k dispozici políčka

- **Správná odpověď** – zde zadejte co nepřesnější číselnou odpověď.
- **Tolerance** – zde zadejte toleranci, ve které bude číselná odpověď akceptována (například v příkladech, kde se předpokládá zaokrouhlování)
- **Rada učitele** – pokud chcete studentům poskytnout užitečnou radu k řešení příkladu, vyplňte ji zde.

Při zakládání výběrového příkladu jsou k dispozici následující políčka:

- **Správná odpověď** – zde vyplňte správnou odpověď
- **Nesprávná odpověď 1 – 3** – zde vyplňte nesprávné odpovědi, které budou součástí výběru.
- **Rada učitele** – pokud chcete studentům poskytnout užitečnou radu k řešení příkladu, vyplňte ji zde.

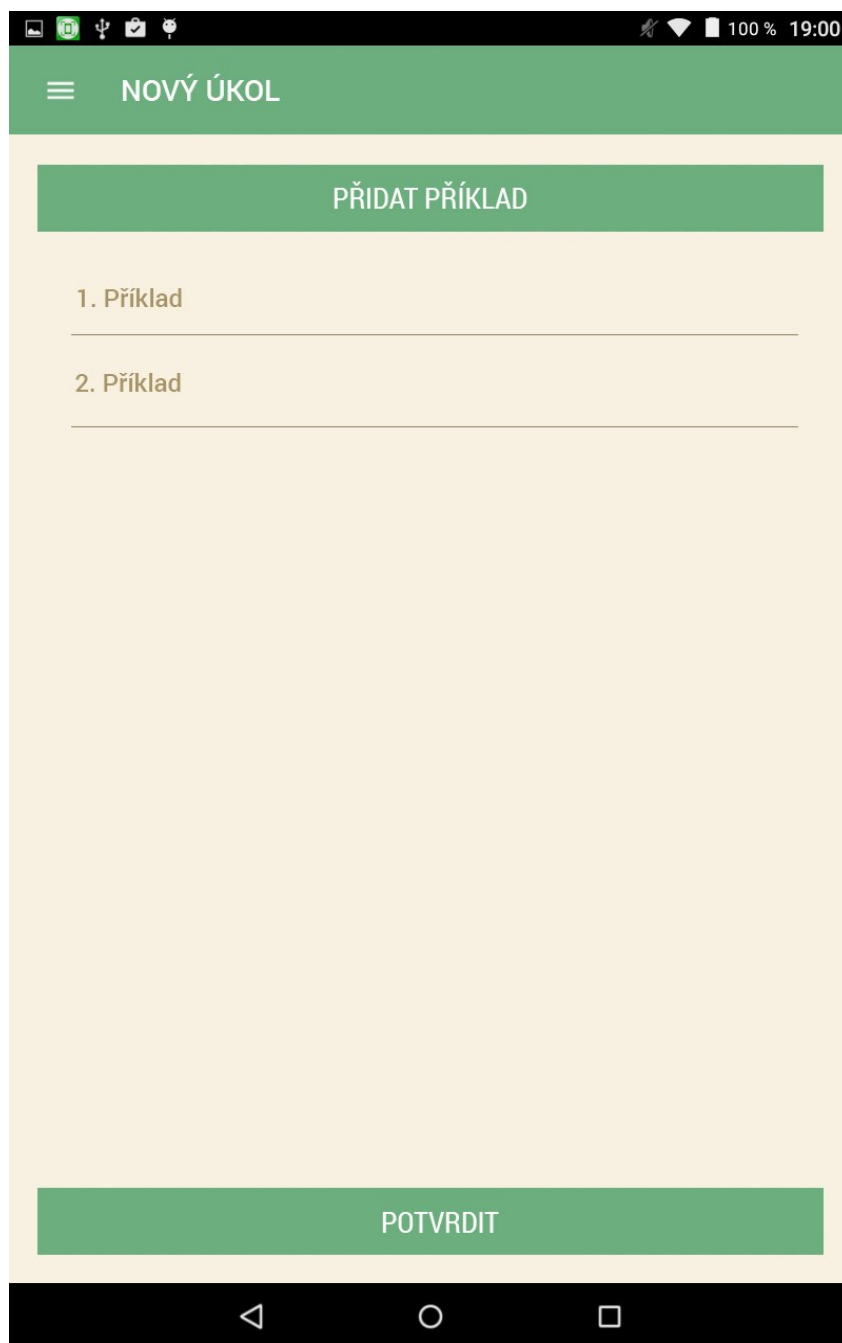




The screenshot shows a mobile application interface for creating a new example. At the top, there is a green header bar with a hamburger menu icon and the text "NOVÝ PŘÍKLAD". Below the header, there are two tabs: "ČÍSELNÝ" (selected) and "VÝBĚROVÝ". The main content area is a light beige color and contains several input fields with labels: "Zadání", "Správná odpověď", "Tolerance", and "Rada učitele". The "Tolerance" and "Rada učitele" fields have a green question mark icon to their right. At the bottom of the form, there is a green button labeled "POTVRDIT". The status bar at the top shows various icons, signal strength, Wi-Fi, 100% battery, and the time 19:00. The bottom navigation bar shows the standard Android navigation icons.

Obr. 37. Obrazovka formuláře pro nový příklad.

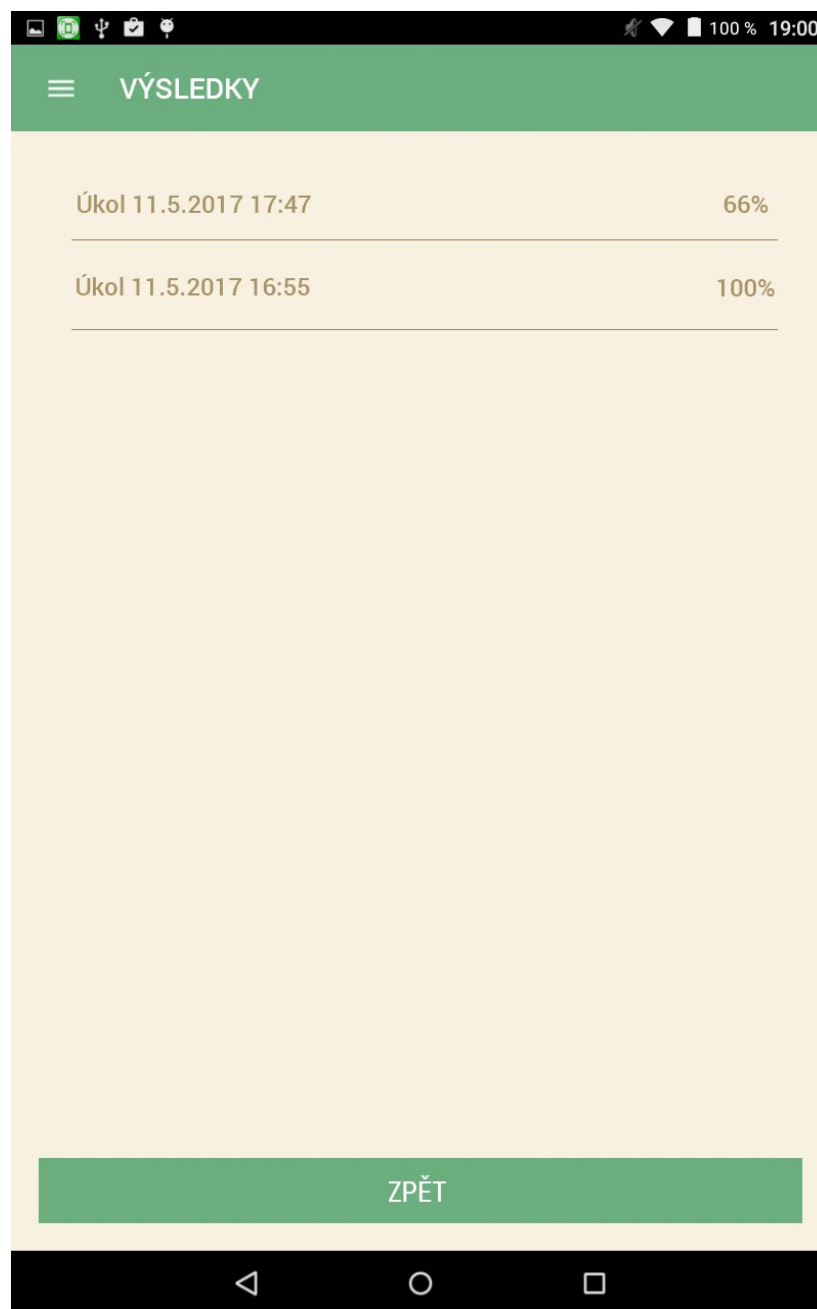
Po přidání jednoho nebo více příkladů je možné založit úkol tlačítkem ve spodní části obrazovky. Po založení úkolu bude zaslán studentům ve třídě k vypracování.



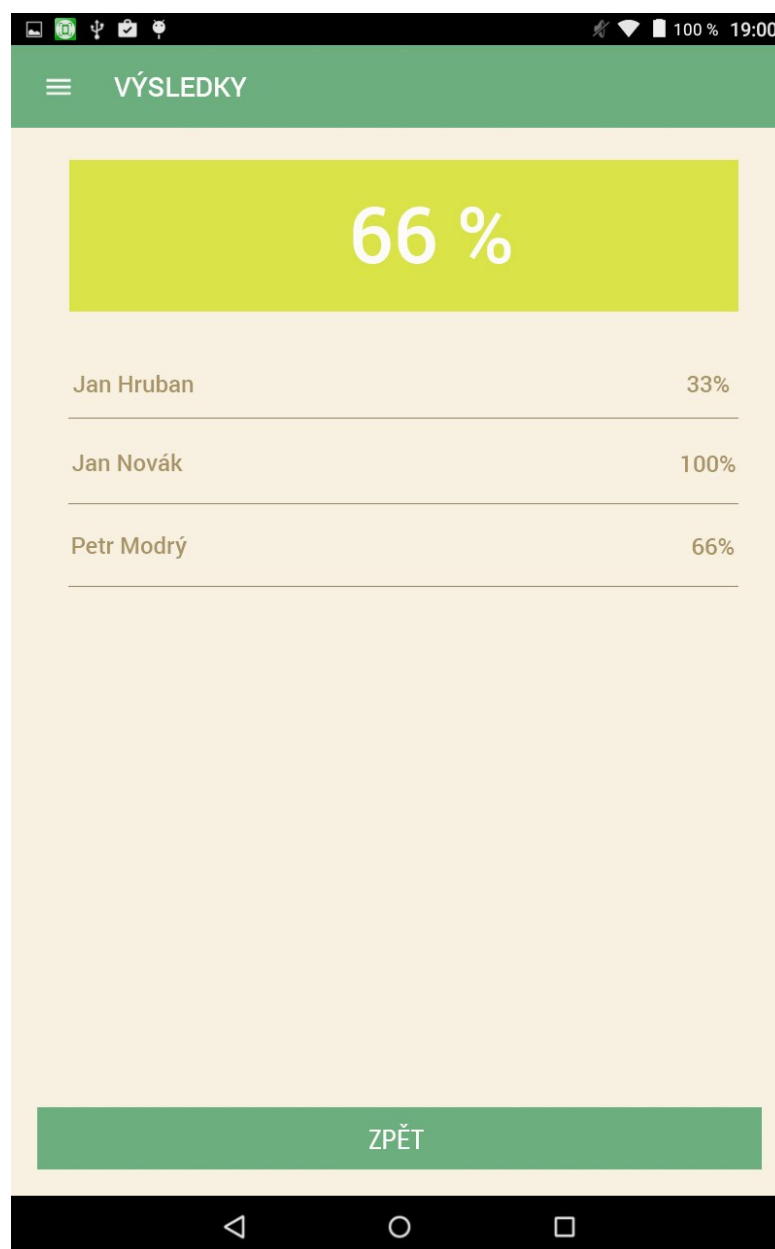
Obr. 38. Obrazovka pro zadání nového úkolu.

#### 4.2.5 Výsledky

Na obrazovce výsledků se nachází souhrnná informace o úspěšnosti studentů v testech a také lze rozkliknout jednotlivé testy a zobrazit výsledky jednotlivých studentů.



Obr. 39. Obrazovka výsledků z úkolů.



Obr. 40. Obrazovka úspěšnosti studentů z úkolu.

#### 4.2.6 Hry

Viz 4.1.7

### 4.3 Uživatelská dokumentace - přihlášení bez registrace

Do mobilní aplikace se lze přihlásit i bez registrace. V takovém případě se aplikace chová jako ke studentovi s výjimkou toho, že není k dispozici sekce Úkoly. Jako učební texty jsou použity univerzální texty. Pro první přihlášení je nutné připojení k internetu, aby mohlo dojít k synchronizaci příkladů a učebních textů.

## 5 TESTOVÁNÍ APLIKACE

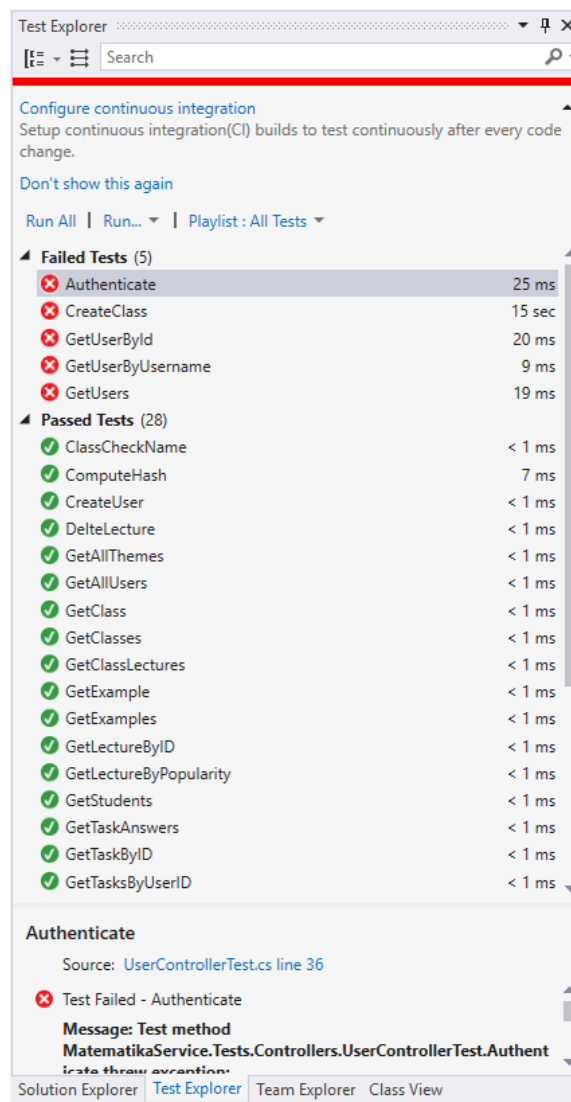
Samotné testování aplikace bylo nutné rozdělit do dvou částí. V první části bylo nutné testovat správnou funkčnost REST API, v druhé části testovat mobilní aplikaci.

### 5.1 Testování webové aplikace a REST API

Pro testování REST API a backendové části webové aplikace jsem využil funkci automatizovaných unit testů, které jsou základní součástí vývojového prostředí Visual Studio. Pro každý endpoint API byl vytvořen jeden unit test. Tyto unit testy byly napsány formou porovnávání známých výstupů, v reakci na známé vstupy.

Takto napsané automatizované unit testy mají výhodu v podobě testování integrity aplikace po případných změnách, opravách a rozšířeních. Tedy například, že po optimalizaci dotazů do databáze, bude aplikace vracet stále stejné výsledky.

Před každým vystavením aplikace do produkčního prostředí jsem provedl hromadné spuštění unit testů a teprve po úspěšném doběhnutí všech testů, byla aplikace vystavena do prostředí Azure.



Obr. 41. Přehled unit testů API v prostředí  
Visual Studio

## 5.2 Testování mobilní aplikace

Testování mobilní aplikace jsem rozdělil do dvou fází. První fází bylo testování aplikace podle testovacích scénářů. Sepsání těchto scénářů jsem přenechal další osobě, kterou jsem nejprve s funkcemi aplikace podrobně seznámil. Tyto scénáře jsou psány pro různé části aplikace (např. registrace, přihlášení, cvičení) a jsou psány formou kroků s očekávaným výsledkem. Příklad takového scénáře lze vidět v příloze PI. Ukázka testovacího scénáře. Veškeré testovací scénáře jsou pak dostupné v elektronické příloze na CD.

Další fází testování byla jistou obdobou pilotních testů, kdy byla aplikace testována jednou dospělou osobou v roli učitele a třemi dětmi v roli studentů. Ideální situací by bylo, kdyby

byla testovací skupina tvořena skutečným učitelem a skutečnou třídou dětí, to však v letním období nebylo možné.

### 5.3 Výsledky testování

Testování REST API pomocí unit testů bylo velice výhodné při různých úpravách a optimalizacích v průběhu vývoje. Testování pomocí unit testů považuji za předpokladu dostatečného pokrytí pro tento druh aplikace za dostatečné.

Testování mobilní aplikace pomocí testovacích scénářů provedli 4 různí uživatelé, většina těchto testů skončila neúspěchem z důvodů různě závažných chyb. Hledání těchto chyb bylo usnadněno popisem místa aplikace a podmínek za jakých k těmto chybám došlo. Tyto chyby byly z aplikace postupně odstraněny. Po odstranění chyby byl tento chybný scénář zopakován, dokud nebyl úspěšný. Touto formou testování bylo odhaleno a opraveno největší množství chyb ze všech typů testů (včetně vlastního testování při vývoji).

Nezávislé „pilotní“ testování v úzké skupince odhalil několik méně závažných chyb, většinou ve formě špatného zobrazení prvků uživatelského rozhraní. Toto testování bylo výhodné spíše z hlediska podnětů pro přidání nových nebo úpravou existujících funkcionalit.

## ZÁVĚR

V této práci bylo mým úkolem vytvořit mobilní výukovou aplikaci matematiky pro základní a střední školy. Téma této práce jsem si vybral, protože je mi blízké jak z profesního, tak z osobního hlediska. V obrovském množství mobilních aplikací jsem postrádal výběr opravdu kvalitních aplikací z oblasti vzdělávání, a proto jsem v tomto tématu spatřil zajímavou výzvu, takovouto aplikaci vytvořit.

Mojí ambicí bylo vytvořit aplikaci, která bude graficky přívětivá a přitom jednoduchá a na první pohled srozumitelná pro uživatele. Aplikaci, která umožní uživatelům neustále rozvíjet její obsah. Aplikaci, která staví na interakci žáka a učitele. A konečně aplikaci, která bude nejen přispívat ke vzdělání, ale která bude i zábavná.

V teoretické části této práce jsem se věnoval popisu dvou nejpoužívanějších mobilních operačních systémů současnosti Android a iOS. Popsal jsem architekturu obou systémů i specifiky vývoje pro tyto dvě platformy. Také jsem se dotkl tématu multiplatformního vývoje pro mobilní operační systémy. V teoretické části jsem také shrnul aktuální rámcové vzdělávací programy výuky matematiky pro základní a střední školy. V praktické části už jsem se věnoval popisu vývoje mobilní aplikace a tvorbě uživatelské dokumentace.

Abych docílil interakce žáka a učitele, přenositelnosti uživatelských účtů a možnosti neustálého rozšiřování aplikace, musel jsem vedle samotné mobilní aplikace vytvořit i backend aplikaci ve formě REST API s vlastní sdílenou databází. Pro lepší možnosti vytváření obsahu jsem vytvořil navíc i podpůrný web pro učitele, který umožňuje vytvářet učební texty s formátovaným textem a obrázky. Největším problémem, se kterým jsem se při tvorbě aplikace setkal, bylo zpracování uživatelského rozhraní pro specifické úlohy matematiky, jako např. geometrické úlohy. Zpracování takového uživatelského rozhraní by daleko přesahovalo časový fond, který jsem měl pro tvorbu této aplikace k dispozici. Z tohoto důvodu se aplikace orientuje především numerická témata a úkoly.

Zpracování této diplomové práce mi přineslo mnoho nových poznatků z oblasti vývoje mobilních aplikací.



## 6 BIBLIOGRAFIE

- [1] IMathematics™ – Aplikace pro Android ve službě Google Play. *Google Play* [online]. Mountain View: Google, Inc., 2009 [cit. 2017-08-18]. Dostupné z: <https://play.google.com/store/apps/details?id=it.droidtech.imathematica&hl=cs>
- [2] IMathematics. *Mobixee* [online]. Londýn: Mobixee Limited, 2015 [cit. 2017-08-22]. Dostupné z: <http://www.mobixee.com/portfolio/item/imathematics/>
- [3] IMathematics Pro Educator Review | Common Sense Education. *Common Sense* [online]. San Francisco, Kalifornie: Common Sense Media, 2003 [cit. 2017-08-19]. Dostupné z: <https://www.commonsense.org/education/app/imathematics-pro>
- [4] Mathway - Math Problem Solver – Aplikace pro Android ve službě Google Play. *Google Play* [online]. Mountain View: Google, Inc., 2009 [cit. 2017-08-18]. Dostupné z: <https://play.google.com/store/apps/details?id=com.bagatrix.mathway.android&hl=cs>
- [5] Mathway Educator Review | Common Sense Education. *Common Sense* [online]. San Francisco, Kalifornie: Common Sense Media, 2003 [cit. 2017-08-19]. Dostupné z: <https://www.commonsense.org/education/app/mathway>
- [6] Mathspace – Aplikace pro Android ve službě Google Play. *Google Play* [online]. Mountain View: Google, Inc., 2009 [cit. 2017-08-18]. Dostupné z: <https://play.google.com/store/apps/details?id=co.mathspace&hl=cs>
- [7] Mathspace Educator Review | Common Sense Education. *Common Sense* [online]. San Francisco, Kalifornie: Common Sense Media, 2003 [cit. 2017-08-19]. Dostupné z: <https://www.commonsense.org/education/website/mathspace>
- [8] Android, the world's most popular mobile platform. *Android developers* [online]. Mountain View: Google, Inc., 2009 [cit. 2017-04-30]. Dostupné z: <https://developer.android.com/about/android.html>
- [9] LACKO, Ľuboslav. *Vývoj aplikací pro Android*. 1. vyd. Brno: Computer Press, 2015. ISBN 978-80-251-4347-6.

- [10] ELGIN, Ben. Google Buys Android for Its Mobile Arsenal. *Bloomberg: Businessweek* [online]. 2005 [cit. 2017-04-30]. Dostupné z: <http://webcitation.org/5wk7sIvVb>
- [11] Industry Leaders Announce Open Platform for Mobile Devices. *Open Handset Alliance* [online]. Mountain View, 2007 [cit. 2017-04-30]. Dostupné z: [http://www.openhandsetalliance.com/press\\_110507.html](http://www.openhandsetalliance.com/press_110507.html)
- [12] HRON, Michal. T-Mobile a Google představily přímého konkurenta pro iPhone. *IDNES* [online]. 2008 [cit. 2017-04-30]. Dostupné z: [http://mobil.idnes.cz/t-mobile-a-google-predstavily-primeho-konkurenta-pro-iphone-pla-/telefony.aspx?c=A080923\\_170454\\_telefony\\_hro](http://mobil.idnes.cz/t-mobile-a-google-predstavily-primeho-konkurenta-pro-iphone-pla-/telefony.aspx?c=A080923_170454_telefony_hro)
- [13] VÁVRŮ, Jiří a Miroslav UJBÁNYAI. *Programujeme pro Android. 2.*, rozš. vyd. Praha: Grada, 2013. Průvodce (Grada). ISBN 978-80-247-4863-4.
- [14] Platform Architecture | Android Developers. *Android Developers* [online]. Mountain View, Kalifornie: Google Inc., 2008 [cit. 2017-08-27]. Dostupné z: <https://developer.android.com/guide/platform/index.html>
- [15] The Activity Lifecycle. In: *Android Developers* [online]. Mountain View, Kalifornie: Google Inc., 2008 [cit. 2017-04-30]. Dostupné z: <https://developer.android.com/guide/components/activities/activity-lifecycle.html>
- [16] Services. *Android Developers* [online]. Mountain View, Kalifornie: Google Inc., 2008 [cit. 2017-04-30]. Dostupné z: <https://developer.android.com/guide/components/services.html>
- [17] Services. *Android Developers* [online]. Mountain View, Kalifornie: Google Inc., 2008 [cit. 2017-04-30]. Dostupné z: <https://developer.android.com/guide/components/services.html>
- [18] About the iOS Technologies. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2014 [cit. 2017-05-02]. Dostupné z: <https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/Introduction/Introduction.html>

- [19] Core OS Layer. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2014 [cit. 2017-05-02]. Dostupné z: [https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html///apple\\_ref/doc/uid/TP40007898-CH11-SW1](https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreOSLayer/CoreOSLayer.html///apple_ref/doc/uid/TP40007898-CH11-SW1)
- [20] Core Services Layer. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2014 [cit. 2017-05-02]. Dostupné z: [https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html///apple\\_ref/doc/uid/TP40007898-CH10-SW5](https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/CoreServicesLayer/CoreServicesLayer.html///apple_ref/doc/uid/TP40007898-CH10-SW5)
- [21] Media Layer. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2014 [cit. 2017-05-02]. Dostupné z: [https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html///apple\\_ref/doc/uid/TP40007898-CH9-SW4](https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/MediaLayer/MediaLayer.html///apple_ref/doc/uid/TP40007898-CH9-SW4)
- [22] Cocoa Touch Layer. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2014 [cit. 2017-05-02]. Dostupné z: [https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html///apple\\_ref/doc/uid/TP40007898-CH3-SW1](https://developer.apple.com/library/content/documentation/Miscellaneous/Conceptual/iPhoneOSTechOverview/iPhoneOSTechnologies/iPhoneOSTechnologies.html///apple_ref/doc/uid/TP40007898-CH3-SW1)
- [23] The App Life Cycle. *Apple Developer* [online]. Cupertino, Kalifornie: Apple Inc., 2017 [cit. 2017-05-02]. Dostupné z: <https://developer.apple.com/library/content/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/TheAppLifeCycle/TheAppLifeCycle.html>
- [24] PETZOLD, Charles. *Creating Mobile Apps with Xamarin.Forms: Cross-platform C# programming for iOS, Android, and Windows*. 1. Redmond, Washington: Microsoft Press, 2016. ISBN 978-1-5093-0297-0.
- [25] TRAEG, Peter. Four Ways To Build A Mobile Application, Part 3: PhoneGap. In: FRIEDMAN, Vitaly a Sven LENNARTZ. *Smashing Magazine* [online]. Německo, 2014 [cit. 2017-05-04].

- [26] Pros and Cons of the Top 5 Cross-Platform Tools. In: *Developer Economics* [online]. Londýn: VisionMobile Ltd, 2013 [cit. 2017-05-04]. Dostupné z: <https://www.developereconomics.com/pros-cons-top-5-cross-platform-tools>
- [27] Introduction to Mobile Development. *Xamarin Developer* [online]. San Francisco, Kalifornie: Microsoft, 2011 [cit. 2017-05-04]. Dostupné z: [https://developer.xamarin.com/guides/cross-platform/getting\\_started/introduction\\_to\\_mobile\\_development/](https://developer.xamarin.com/guides/cross-platform/getting_started/introduction_to_mobile_development/)
- [28] NASH, Trey. *Accelerated C# 2010*. New York: Distributed to the book trade worldwide by Springer Science Business Media, 2010. Expert's voice in C#. ISBN 978-143-0225-379.
- [29] Mobile Application Development to Build Apps in C# - Xamarin. *Xamarin* [online]. San Francisco, Kalifornie: Microsoft, 2011 [cit. 2017-05-04]. Dostupné z: <https://www.xamarin.com/platform>
- [30] EISLER, Jaroslav. *Matematika v kostce: pro základní školy a nižší ročníky víceletých gymnázií*. 1. vyd. Havlíčkův Brod: Fragment, 2006. V kostce (Fragment). ISBN 80-253-0197-4.
- [31] *Rámcový vzdělávací program pro základní vzdělávání*. Praha: NÚV - Národní ústav pro vzdělávání, 2016.
- [32] VOŠICKÝ, Zdeněk. *Matematika v kostce: pro střední školy*. 1. vyd. Havlíčkův Brod: Fragment, 2007. Maturita v kostce. ISBN 978-1430225379.
- [33] BALADA, Jan. *Rámcový vzdělávací program pro gymnázia: RVP G*. Praha: Výzkumný ústav pedagogický v Praze, 2007. ISBN 978-80-87000-11-3.

**SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK**

API	Application Programming Interface
CSS	Cascading Style Sheets
DVM	Dalvik Virtual Machine.
GPU	Graphics processing unit.
GUID	Globaly unique identifier
HTML	HyperText Markup Language
IDE	Integrated Development Environment
JSON	JavaScript Object Notation
OS	Operační systém
REST	Representational state transfer
UWP	Universal Windows Platform
VoIP	Voice over Internet Protocol

**SEZNAM OBRÁZKŮ**

Obr. 1. Ukázka aplikace iMathematics. [1] .....	12
Obr. 2. Ukázka aplikace Mathway [4].....	13
Obr. 3. Ukázka aplikace MathSpace [6] .....	14
Obr. 4. Architektura OS Android [14].....	17
Obr. 5. Životní cyklus aktivity [15] .....	20
Obr. 6. Životní cyklus služby [16].....	21
Obr. 7. Ukázka Content Provideru a správy přístupu k datům aplikace [17].....	22
Obr. 8. Architektura systému iOS. [18].....	23
Obr. 9. Struktura iOS aplikace. [23] .....	28
Obr. 10. Ukázka zpracování události v hlavní aplikační smyčce. [23] .....	29
Obr. 11. Životní cyklus iOS Aplikací. [23] .....	30
Obr. 12. Grafické znázornění technologie Xamarin.Forms. [29].....	34
Obr. 13. Schéma jednotlivých aplikací a propojení.....	44
Obr. 14. Ukázka verzování softwaru v programu SourceTree. ....	45
Obr. 15. Struktura backend .....	47
Obr. 16. Struktura projektu mobilní aplikace. ....	52
Obr. 17. Struktura webové aplikace.....	54
Obr. 18. Hra pexeso. ....	56
Obr. 19. Hra Sudoku. ....	57
Obr. 20. Hra 2048. ....	58
Obr. 21. Rozhraní testů a výsledků.....	59
Obr. 22. Přihlášení a registrace do aplikace.....	60
Obr. 23. Obrazovka registrace. ....	62
Obr. 24. Obrazovka výběru role. ....	64
Obr. 25. Přihlašovací obrazovka.....	65
Obr. 26. Úvodní obrazovka pro studenty.....	66
Obr. 27. Obrazovka výběru přednášek. ....	68
Obr. 28. Obrazovka výběru příkladů pro cvičení. ....	69
Obr. 29. Obrazovka testu. ....	70
Obr. 30. Obrazovka výsledku testu.....	71
Obr. 31. Obrazovka výběru úkolů. ....	72
Obr. 32. Obrazovka hlavního menu aplikace. ....	73

---

Obr. 33. Obrazovka registrace. ....	75
Obr. 34. Obrazovka založení třídy. ....	76
Obr. 35. Úvodní obrazovka pro učitele. ....	77
Obr. 36. Obrazovka přehledu přenášek. ....	79
Obr. 37. Obrazovka formuláře pro nový příklad. ....	81
Obr. 38. Obrazovka pro zadání nového úkolu. ....	82
Obr. 39. Obrazovka výsledků z úkolů. ....	83
Obr. 40. Obrazovka úspěšnosti studentů z úkolu. ....	84
Obr. 41. Přehled unit testů API v prostředí. ....	86

**SEZNAM TABULEK**

Tab. 1. Historie verzí Androidu. ....	15
Tab. 2. Přehled a popis tabulek v backend databázi. ....	48
Tab. 3. Přehled a popis jednotlivých endpointů REST API. ....	49



## PŘÍLOHA P I: UKÁZKA TESTOVACÍHO SCÉNÁŘE

Název		Registrace studenta				
Projekt		Matematika				
Vytvořil		Zuzana Hrubanová				
Upravil		Jan Hruban				
Celkový čas testu		Datum vytvoření				
Stručný popis změn		Datum úpravy				
ID	Název	Podmínky	Kroky (časová náročnost)	Očekávaný výsledek	Poznámka	Výsledek
#1	Registrace 1. krok		<ol style="list-style-type: none"> <li>Spustím mobilní aplikaci.</li> <li>Na přihlašovací obrazovce kliknu na „Registrovat“.</li> <li>Vyzkouším validaci kliknutím na „OK“.</li> <li>Vyplním všechny údaje, do pole „Heslo znovu“ napíšu jiné heslo než do pole „Heslo“.</li> <li>Vyplním obě hesla stejně a potvrdím.</li> </ol>	<ol style="list-style-type: none"> <li>Mobilní aplikace se spustí a jsem na přihlašovací obrazovce.</li> <li>Dostanu se na obrazovku registrace, vidím políčka pro vyplnění jména a přihlašovacích údajů.</li> <li>U políček se zobrazí validační hlášky s výzvou k vyplnění.</li> <li>U pole „heslo znovu“ se zobrazí validační hláška o neshodném hesle.</li> <li>Dostanu se na obrazovku 2. kroku registrace.</li> </ol>		
#2	Registrace 2. krok		<ol style="list-style-type: none"> <li>Do pole „Vyberte třídu“ zadám část názvu třídy a nějakou výberu.</li> <li>Vyplním špatný klíč a potvrdím.</li> <li>Vyplním správný klíč a potvrdím.</li> </ol>	<ol style="list-style-type: none"> <li>Do pole se vyplní název vybrané třídy.</li> <li>U políčka se zobrazí validační hláška, že klíč není správný.</li> <li>Proběhne zaregistrování uživatele a přihlášení do aplikace.</li> </ol>		