



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Název:	Conjugata - aplikace pro cvičení časování sloves - backend
Student:	Petr Polívka
Vedoucí:	Ing. Filip Křikava, Ph.D.
Studijní program:	Informatika
Studijní obor:	Webové a softwarové inženýrství
Katedra:	Katedra softwarového inženýrství
Platnost zadání:	Do konce zimního semestru 2019/20

Pokyny pro vypracování

Cílem práce je Android aplikace pro procvičování časování sloves ve španělštině, s možností rozšíření na ostatní latinské jazyky.

Funkční požadavky jsou:

- Dva režimy práce: jednoduchý (čas slovesa je zadán) a pokročilý (čas slovesa se musí odvodit z věty).
- Tři druhy klávesnic: jednoduchá (tlačítka reprezentující tvary slovesa), pokročilá (zobrazují se písmena, ze kterých je potřeba složit požadovaný tvar slovesa) a plná (standardní klávesnice Androidu).
- Bodování odpovědí, vytváření cvičení na míru.
- Učební plán pro jednotlivé druhy gramatických časů.

Úkoly:

1. Navrhněte, diskutujte, zvolte technologie a implementujte databázi pro aplikaci a naplňte ji veřejně dostupnými daty.
2. Navrhněte, implementujte a otestujte funkce pro generování cvičení.
3. Celou práci řádně zdokumentujte.
4. Práci koordinujte s Matějem Sedlákem, jehož úkolem je návrh a implementace frontend.

Seznam odborné literatury

Dodá vedoucí práce.

Ing. Michal Valenta, Ph.D.
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.
děkan

V Praze dne 26. února 2018



**FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE**

Bakalářská práce

Conjugata - mobilní aplikace pro cvičení časování sloves - backend

Petr Polívka

Katedra Softwarového inženýrství
Vedoucí práce: Ing. Filip Kříkava, Ph.D.

14. května 2018

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 14. května 2018

.....

České vysoké učení technické v Praze
Fakulta informačních technologií

© 2018 Petr Polívka. Všechna práva vyhrazena.

Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.

Odkaz na tuto práci

Polívka, Petr. *Conjugata - mobilní aplikace pro cvičení časování sloves - backend*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2018.

Abstrakt

Tato bakalářská práce se zabývá analýzou, návrhem a následnou implementací backendu mobilní aplikace pro operační systém Android. Aplikace se jménem Conjugata pomáhá se studiem časování sloves v cizích jazycích, konkrétně ve španělštině. Aplikace se při vytváření cvičení přizpůsobuje odpovědím uživatele, který aplikaci používá. Obsahuje vybrané časy a kategorie používané ve španělském jazyce. Obsahuje také systém výuky, který uživatele provede pravidly, které se při časování ve španělském jazyce používají. Aplikace je užitečná pro všechny studenty španělského jazyka, kteří si chtějí časování sloves procvičit, ale zároveň i pro nováčky, kteří časování neovládají, ale chtějí se ho naučit. Tato aplikace je vytvářena ve spolupráci se studentem ČVUT Matějem Sedlákem, který vytváří frontend aplikace. V příloze se nachází výsledná aplikace ve formátu APK pro mobilní telefony s operačním systémem Android.

Klíčová slova Mobilní aplikace, Android, backend, databáze, výuka jazyka, španělština, Java

Abstract

This bachelors thesis focuses on analysis, schema and subsequent backend implementation of a mobile application for Android operating system. The application called Conjugata aids with learning verb conjugation in foreign languages, concretely in spanish. It also adjusts itself to the answers the end user is choosing. It offers selected tenses and categories used in spanish language. Furthermore, it provides educational system, that walks the user through the rules of spanish verb conjugation. We created a helpful tool for all spanish students, who want to practice verb conjugation, but also for beginners, that are new to verb conjugation and want to learn it. The application was made in cooperation with ČVUT student Matej Sedlak, who worked on the applications frontend. In the attachment you'll find the finished application in APK format for mobile phones with Android operating system

Keywords Mobile application, Android, backend, database, language learning, Spanish, Java

Obsah

Úvod	1
1 Cíl práce	3
2 Analýza	5
2.1 Aplikace s podobnou tematikou	5
2.2 Přehled časování sloves ve španělštině	10
3 Návrh	11
3.1 Požadavky aplikace	11
3.2 Návrh databáze a inicializační soubory	12
3.3 Generování otázek a přizpůsobivost aplikace	13
3.4 Výukový systém	14
4 Implementace	17
4.1 Použité technologie	17
4.2 Databáze	18
4.3 Generátor otázek	20
4.4 Výuka	20
4.5 Konfigurace aplikace	21
4.6 Propojení frontendu a backendu	21
Závěr	23
Literatura	25
A Seznam použitých zkratk	27
B Obsah příloženého CD	29

Seznam obrázků

2.1	Snímky obrazovek aplikace Verbo	6
2.2	Snímky obrazovek aplikace VerbBlitz	7
2.3	Snímky obrazovek aplikace VerbBlitz	8
3.1	Entitní model návrhu databáze	15
3.2	Diagram generování otázek pro cvičení	16
4.1	Graf zastoupení aktivních verzí systému Android	18
4.2	Snímky obrazovek z výsledné aplikace	22

Seznam tabulek

2.1	Tabulka porovnání statistik konkurenčních aplikací	5
2.2	Tabulka srovnání konkurenčních aplikací	9

Úvod

Chytré telefony se v dnešní době staly multifunkčními zařízeními a užitečnými pomocníky v různých situacích, pomocí velké škály aplikací, které v nich můžeme mít. Podle AppBrain.com [1], který vytváří statistiky pro aplikace na operační systém Android, jsou edukativní aplikace nejrozsáhlejší skupinou. Ze všech aplikací na Google play, kterých je více než 3 700 000, je téměř 10 % zaměřeno na vzdělávání. Mnoho z těchto aplikací se různými způsoby věnují studiu cizích jazyků.

Má bakalářská práce, společně s prací studenta Matěje Sedláka, se zabývá právě tímto tématem. Jejím cílem je backend aplikace, která pomáhá studentům cizích jazyků, kteří se chtějí naučit nebo procvičit časování sloves. Časování sloves je jednou z nejdůležitějších a nejtěžších částí při osvojování cizího jazyka. Aplikace nabízí velké množství výukových programů a procvičovací úloh. Pro aktuální řešení byla vybrána španělština, kvůli velkému počtu časů a různorodosti tvarů při časování.

Hlavními výhodami aplikace oproti konkurenci bude výukový program, který uživatele provede vybranými časy ze španělského jazyka, ale nebude ho ve studiu omezovat. Spousta uživatelů již nějaké zkušenosti s časováním mají a nechtějí se učit vše od začátku. Další výhodou bude dynamický systém procvičování, který bude nabízet velkou škálu možností pro výběr specifického cvičení a s možností doplňování sloves do předem připravených vět. Na výběr bude kromě času a kategorie také jeden ze tří systémů odpovědí. Lehký, kde se odpověď vybírá z pěti připravených možností. Dále střední, ve kterém uživatel odpověď tvoří z nabízených písmen. A těžký, kde se odpověď píše pomocí mobilní klávesnice do textového pole.

Motivací k výběru tématu byla možnost vytvoření veřejně prospěšné aplikace, která by moderní formou pomohla lidem se studiem cizích jazyků a která by se přizpůsobovala danému uživateli, který aplikaci používá. Zároveň mě mobilní aplikace zajímají a rád bych se jejich vývoji dále věnoval.

V první kapitole zabývám nastudováním aktuálních řešení daného pro-

blému, pomocí analýzy konkurenčních aplikací. Dále také studiem samotných pravidel časování sloves ve španělském jazyce. V druhé kapitole se věnuji návrhu databáze a systému pro ukládání a generování dat. Poté návrhem systému, který podle předešlých odpovědí uživatele bude připravovat úlohy k procvičování. Po dokončení návrhů se ve třetí kapitole zabývám jejich implementací. Zároveň se budu věnovat možnostem konfigurace aplikace.

Tato práce úzce spolupracuje s prací kolegy Matěje Sedláka (student FIT ČVUT v Praze), který má za úkol vytvořit frontend aplikace, neboli návrhy uživatelského rozhraní, systému jednoduchého průchodu aplikací a grafickým prostředím. Je potřeba spolupracovat v průběhu celého projektu, aby spolu obě části dobře komunikovaly a aplikace fungovala správně.

Cíl práce

Cílem práce je navrhnout a vytvořit backend aplikace pro operační systém Android na jednoduché a efektivní procvičování časování sloves ve španělštině.

Cílem analýzy je studium teoretických informací potřebných k vytvoření aplikace. Nejdříve je třeba analyzovat existující aplikace s podobnou tematikou na základě určených kritérií. Důležité je si nastudovat pravidla časování sloves ve španělském jazyce, pro všechny časy. Tyto informace budou dále použity k návrhu databáze a generátoru slovesných tvarů.

Návrhová část připravuje databázový model na základě uživatelských požadavků. Dále také systém, který podle předešlých odpovědí bude pro specifické kategorie a časy připravovat otázky k procvičování. Nakonec navrhnout podobu výukového programu, který musí zahrnovat studijní materiály a průběžné procvičování právě nastudovaných tvarů a celková opakovací cvičení.

Cílem implementační části je vytvoření funkční backendové části aplikace podle návrhů z předešlé části. To zahrnuje v první řadě vytvoření databáze, podle aktuálních postupů pro tvorbu databázových systémů pro operační systém Android (součástí [2]). Vytvoření inicializačních souborů obsahujících potřebná data (Seznamy slov, tvarů, kategorií, atd.) a jejich naplnění do databáze při prvním spuštění aplikace. Dále pak implementace systému generování procvičovacích otázek, na základě zaznamenaných odpovědí, zahrnující vytváření správných i špatných odpovědí. Během vývoje je důležité synchronizovat backendovou část s frontendem (Matěje Seldáka), pro zajištění správné komunikace v celé aplikaci.

Analýza

2.1 Aplikace s podobnou tématikou

V této sekci se zaměřím na popis a zhodnocení konkurenčních aplikací věnujících se výuce a procvičování časování španělských sloves. Na této analýze jsme spolupracovali s kolegou Matějem Sedlákem. Z aplikací zaměřených na toto téma jsme podle hodnocení a počtu stažení vybrali tři aplikace. Jejich statistiky z Google Play jsou popsány v tabulce ??.

Název	Verbo	Spanish Verb Blitz	Spanish Grammar
Vývojář	The Spanish Verbo Company	Geolot Apps	Himalaio
Průměrné hodnocení	4,6/5	4,6/5	4,4/5
Počet hodnotících	61	32	88
Počet stažení	5000+	5 000+	10 000+
Dostupnost	Zdarma, bez reklam	Zdarma, malé reklamy	Zdarma, velké reklamy

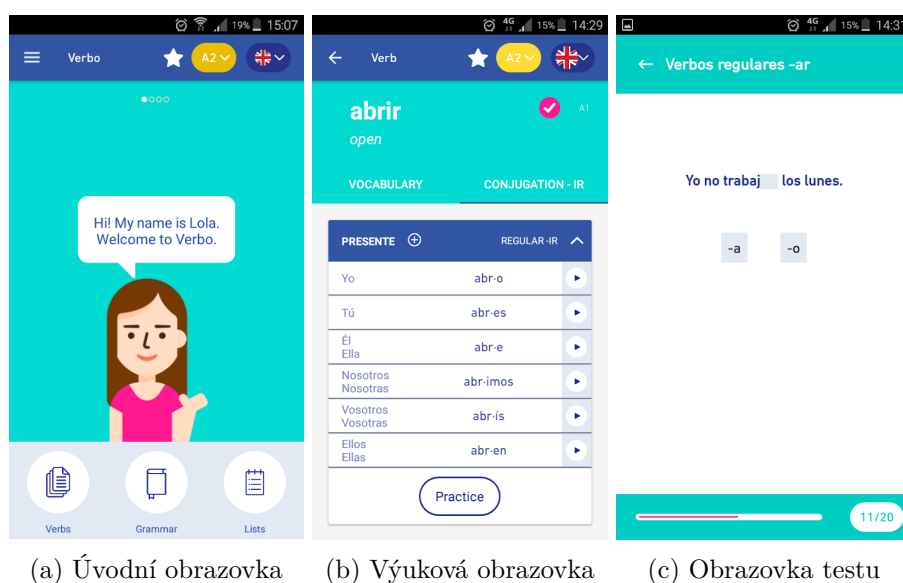
Tabulka 2.1: Tabulka porovnání statistik konkurenčních aplikací

2.1.1 Aplikace Verbo

Verbo [3] je první z vybraných aplikací s podobnou tématikou. Aplikace nabízí možnost výuky, lze vidět na obrázku 2.1b, která uživatele provede pravidly časování španělských sloves. Nabízí výukové materiály ke každé kategorii, následované krátkým testem. Výuka je dobře zpracovaná a rozdělená podle úrovní jazyka (A1, A2, B1, B2, C1, C2) popsanych v [4]. Aplikace je lokalizovaná do španělštiny a angličtiny. Aplikace nenabízí možnost samostatného procvičování, lze pouze opakovat procvičovací testy z výuky. Testy nejsou generované dynamicky, tudíž při opětovném spuštění testu se otázky i odpovědi zobrazí stejné. Testy probíhají formou výběru ze dvou možností, obrázek 2.1c, což způsobuje, že uživatel může lehce odpovědět správně i bez znalosti správné od-

2. ANALÝZA

povědi. V některých testech se podle předepsané věty vybírá správná osoba. V ostatních testech se vybírají pouze koncovky, nebo změny v části sloves, namísto celých tvarů slovesa. To má za následek malý přehled o slovesech patřících do dané kategorie. Seznam sloves v aplikaci je obsáhlý a je možné v něm vyhledávat. Zajímavá je možnost označení sloves, které již známe a třídit tak slovesa pro snadnější vyhledávání. Statistiky úspěšnosti v testech jsou v aplikaci zobrazeny procentuálně podle cvičení pro danou kategorii. Celkově graficky působí aplikace dobře 2.1a. Běh aplikace způsobuje občasné pády a nepřízřívobuje se otáčení obrazovky.



Obrázek 2.1: Snímky obrazovek aplikace Verbo

2.1.2 Aplikace Verb Blitz

Aplikace s názvem Verb Blitz [5] se zaměřuje pouze na procvičování časování sloves. Neobsahuje tedy žádný výukový program. Je ale možné si zobrazit seznam sloves zahrnutých v databázi aplikace a zobrazit si k vybranému slovesu jeho tvary ve všech dostupných časech a jeho překlad do angličtiny. Zároveň je u každého slovesa v seznamu zobrazena jeho procentuální úspěšnost z dříve absolvovaných testů. V aplikaci je možné procvičovat čtyřmi různými způsoby a každý z nich se skládá z deseti otázek. První možností je *Infinitive quiz*, ve kterém k zobrazenému anglickému slovesu vybíráme ze čtyř možností jeho španělský ekvivalent. Druhou alternativou je *Conjunction quiz*, obrázek 2.2c, ve kterém ze čtyř možností vybíráme správný tvar slovesa podle zadané věty. Tyto dvě možnosti jsou v celku podobné, jak vizuálně, tak technicky. Třetí je *Snap quiz*, kde je v horní části obrazovky zobrazena španělská věta a ve spodní části její překlad v anglickém jazyce. Tyto věty se každé tři vteřiny mění a

pokud je význam obou vět totožný, je vaším úkolem kliknout na obrazovku. Po třech neúspěšných pokusech se procvičování ukončí. Poslední možností je *Gapfill* 2.2b, kde do připravené věty skládáte správný tvar slovesa z připravených písmen. Díky více testovacím způsobům se zdá aplikace zábavnější a lépe si udrží uživatele. Otázky jsou generované dynamicky podle množiny časů vybraných v nastavení, tudíž je každý test jiný. Aplikace je jednoduchá a přehledná, je vidět na obrázku 2.2a, lehce se v ní orientuje a vizuálně působí dobře. Aplikace se špatně přizpůsobuje otáčení obrazovky a způsobuje grafické chyby, které však neovlivňují funkčnost aplikace. Aplikace obsahuje malé reklamy v menu. Ty však nenarušují běh aplikace.



(a) Úvodní obrazovka

(b) *Gapfill*(c) *Conjunction quiz*

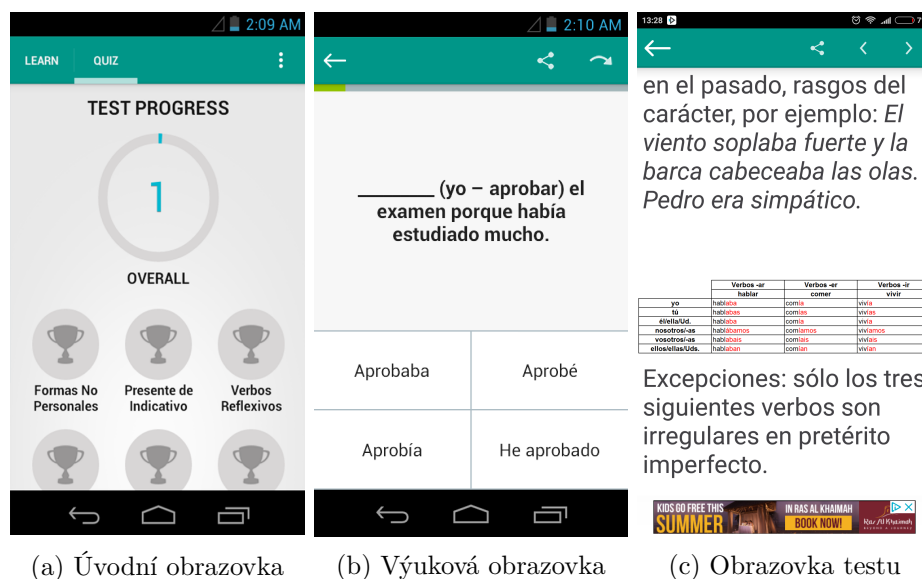
Obrázek 2.2: Snímky obrazovek aplikace VerbBlitz

2.1.3 Aplikace Spanish Grammar

Poslední z vybraných konkurenčních aplikací je Spanish Grammar [6]. Aplikace je rozdělena do dvou částí, výuka a kvízy. Ve výukové části je seznam rozdělený podle časů a kategorií španělského jazyka. Ke každé kategorii je k dispozici popis použití, seznam tvarů a sloves, které do kategorie patří. Informace jsou nepřehledné a špatně vizuálně zpracované, jak vidíme na obrázku 2.3c. Zároveň jsou psané pouze ve španělském jazyce, což pro někoho, kdo se španělsky teprve učí, nemusí být ideální. V kvízové části je na výběr ze 17 připravených cvičení. Každé cvičení obsahuje dvacet otázek, které se při opakování testu nemění. U každého testu je zobrazena jeho úspěšnost, která je vyjádřena počtem správných odpovědí a pracně učenou úspěšností 2.3a. Zároveň je k dispozici ukazatel postupu všemi cvičeními. Při procvičování se určuje správný tvar slovesa podle předepsané věty, obrázek 2.3b. Buď je na výběr ze

2. ANALÝZA

čtyř připravených odpovědí, nebo uživatel tvar tvoří z připravených písmen. Poslední možností je určení, jestli je věta správná. Aplikace působí graficky jednoduše. Aplikace je v dobrém technickém stavu, způsobuje sice občasné grafické chyby, ty ale nenarušují její používání. Aplikace se nepřizpůsobuje otáčení obrazovky a obsahuje spoustu reklam, které zamezují pohodlnému používání aplikace.



Obrázek 2.3: Snímky obrazovek aplikace VerbBlitz

2.1.4 Porovnání a zhodnocení

Každá ze tří vybraných aplikací má trochu jiné zaměření. Aplikace Verbo se zaměřuje na výukový program s průběžným procvičováním, který zpracovává velmi dobře. Pro uživatele, kteří časování sloves neumí a chtějí se ho naučit, je Verbo vhodnou variantou. Aplikace Verb Blitz obsahuje pouze procvičování, má ale zajímavé způsoby testování, které působí zábavně a uživatele u aplikace udrží. Zároveň jako jediná nabízí dynamické generování otázek, takže pro samotné procvičování je tato aplikace nejvhodnější. Spanish Grammar se snaží kombinovat obě tyto funkcionality, bohužel jejich zpracování není moc kvalitní, výuka je nepřehledná a procvičování je příliš jednoduché a málo rozsáhlé. Každá ze tří aplikací má uživatelské rozhraní v angličtině, aplikace Verbo s možností přepnout do španělského jazyka. Databáze sloves je u všech tří aplikací rozsáhlá. Z hlediska vizuální stránky působí nejlépe aplikace Verbo, ale je zároveň tato aplikace v nejhorším technickém stavu.

V tabulce ?? na stránce 9, nalezneme porovnání výše zmiňovaných aplikací, z hlediska několika kategorií.

2.1. Aplikace s podobnou tematikou

Kategorie	Aplikace Verbo	Apikace Verb Blitz	Aplikace Spanish Grammar
Výukový program	Výuka s průběžným procvičováním	Výuka chybí	Pouze seznam kategorií a cvičení
Kvalita výukových materiálů	Jednoduchý, lehce nepřehledný	Pouze seznam tvarů	Nepřehledný seznam kategorií
Grafika výukových materiálů	Jednoduchý, lehce nepřehledný	Základní	Nepřehledný text, zbytečné fotky
Lokalizace výuky	ANG/ŠP	Kombinace ŠP a ANG	ŠP
Lokalizace UI	ANG/ŠP	ANG	ANG
Možnost volného procvičování	Chybí	Jediná část aplikace	Jen předpřipravené kategorie
Počet cvičení	11	Volitelné v závislosti na čase	17
Dynamické generování	Stejně otázky	Dynamické	Stejně otázky
Rozmanitost testovacích módů	2 prakticky stejné	4 naprosto odlišné	3 velmi podobné
Kvalita špatných možností odpovědí	Výběr ze 2	Základní	Špatná, lehce odhadnutelná
Databáze sloves	Střední	Velká	Menší
Přehled informací o slovesech	Dobré	Dobré	Chybí
Vyhledávání	Ve slovesech	Dobré	Ve výukových materiálech
Uživatelská přívětivost	Lehce nepřehledné, přepřácané	Jednoduché, Přehledné	Jednoduché, Přehledné
Celkový vizuál	Zajímavé	Průměrné	Jednoduché
Záznam skóre	Ukazatel postupu cvičeními	Statistiky podle slov	Ukazatel postupu a úspěšnosti testů
Vizuál přehledu postupu	Procentuální podle cvičení	Seznam procent podle slov	Vizuální, graficky přívětivý
Dostupnost	Zdarma bez reklam	Malá reklama v menu	Otravné reklamy
Sdílení	Online tabulky + sdílení	Nelze sdílet	Možnost sdílení otázky, a postupu
Pády aplikace	Občasné	Žádné	Žádné
Chyby v aplikaci	Časté	Grafické chyby	Žádné
Rotace obrazovky	Nelze otáčet	Ve cvičeních způsobuje problémy	Funkční jen v menu, jinak nelze otáčet

Tabulka 2.2: Tabulka srovnání konkurenčních aplikací

2.2 Přehled časování sloves ve španělštině

Tuto sekci jsme vypracovali společně se studentem Matějem Sedlákem, tudíž je v obou bakalářských pracích totožná. Budeme se v ní věnovat pravidlům časování španělských sloves.

Správný tvar slovesa ve španělském jazyce závisí na následujících kritériích:

1. Slovesný čas – Podle CompleteSpanishGuide.com [7] existuje ve španělském jazyce 14 slovesných časů. Každý z nich má odlišná pravidla pro tvorbu tvarů.
2. Kategorie – Ve španělštině se slovesa dají rozdělit do tří kategorií. První z nich jsou pravidelná, u kterých se mění pouze koncovka slovesa. Dále pravidelně nepravidelná, u kterých se kromě změny koncovky nachází i změna uvnitř slovesa. Poslední jsou nepravidelná, která mají každý tvar jiný, neboli pro ně neexistují souhrnná pravidla. Je tedy nutné si u nepravidelných sloves zapamatovat samotné tvary.
3. Třídy – U pravidelných a pravidelně nepravidelných sloves určujeme také slovesné třídy. Určují se na základě koncovky slovesa v infinitivu. Lze je tedy rozdělit na tři třídy (-AR, -ER, -IR). U pravidelně nepravidelných se slovesa rozdělují také podle změny (E->IE, U->UE, atd.).
4. Osoba – Stejně jako v českém jazyce existují ve španělštině tři osoby v jednotném a množném čísle. V každé osobě a čísle má sloveso jiný tvar. Rozdíl oproti českému jazyku je způsob vykání. Vykání jedné osobě má stejný tvar jako třetí osoba jednotného čísla. Vykání více osobám zase jako třetí osoba množného čísla.

Tyto informace jsou důležité pro návrh databáze a UI aplikace. Pro účely naší aplikace jsme vybrali tři z nejpoužívanějších časů a to Presente (Přítomný prostý), Futuro (Budoucí) a Pretérito perfecto simple (Minulý jednoduchý). V aplikaci jsme se rozhodli pokrýt všechny kategorie a třídy patřící do námi zvolených časů. V budoucnu nebude problém přidat další slovesné časy a kategorie

Návrh

3.1 Požadavky aplikace

Podle informací získaných z analýzy jsme navrhli a specifikovali seznam funkčních požadavků na aplikaci.

3.1.1 Funkční požadavky

1. Výuka časování španělských sloves – Aplikace bude obsahovat systém výuky, který uživatele provede vybranými časy a kategoriemi. Výukový systém se bude skládat z výukových materiálů a průběžných procvičovacích testů, aby si uživatel nově nabyté informace vyzkoušel rovnou v praxi.
2. Procvičování slovesných kategorií – V aplikaci bude možnost samostatného procvičování. Zde si uživatel zvolí kritéria, podle kterých se mu cvičení vygeneruje. Nejdříve si vybere slovesný čas a poté jednu z kategorií (pravidelné, nepravidelné, pravidelně nepravidelné). Dále vybere jednu ze tří obtížností, která určuje způsob zadávání odpovědí. V lehké obtížnosti je na výběr z pěti předvytvořených odpovědí, ve střední se odpověď skládá z dvanácti předpřipravených písmen, ze kterých lze poskládat kromě správné odpovědi i další korektní tvary. V těžké obtížnosti se odpověď píše pomocí standardní mobilní klávesnice. Nakonec si uživatel zvolí, zda chce procvičovat samostatné tvary, nebo odpovědi doplňovat do vět. Cvičení se bude skládat z určitého počtu otázek a po dokončení se uživateli zobrazí přehled správných i špatných odpovědí.
3. Vytvoření vlastních okruhů k procvičování – V aplikaci bude také možné si uložit nastavení kritérií, které budou opakovaně používány ke generování cvičení. Bude možné si je pojmenovat a vybrat si více časů a kategorií najednou.

3. NÁVRH

4. Prohlížení statistik odpovědí – Aplikace bude obsahovat přehled statistik dokončených cvičení. Bude možné sledovat procentuální úspěšnost v závislosti na čase a kategorii. Dále bude možné se zpětně podívat, kdy a jaká cvičení uživatel dokončil.

3.1.2 Nefunkční požadavky

1. Připojení k internetu – Aplikace bude fungovat bez připojení k internetu, aby bylo možné aplikaci používat kdekoli a kdykoliv.
2. Systém android – Aplikace bude podporována na všech systémech Android s verzí 4.0.3 a vyšší. Důvody výběru cílové verze jsou více popsány v sekci 4.1.
3. Lokalizace – Aplikace bude lokalizována do českého jazyka. Lokalizace se týká uživatelského rozhraní a výukových materiálů

3.2 Návrh databáze a inicializační soubory

Tato sekce se zaměřuje na návrh a popis databázového modelu a inicializačních souborů.

3.2.1 Databázový model

Podle pravidel časování sloves popsaných v sekci 2.2 a popisu funkčních požadavků jsem vytvořil návrh databáze pro aplikaci. Návrh databázového modelu na obrázku 3.1 na stránce 15 byl vytvořen na školním portálu předmětu BIDS [8].

Základem databázového modelu je rozdělení podle kategorií na pravidelné, pravidelně nepravidelné a nepravidelné. Pro každou z těchto kategorií byla navržena entita pro slovesa, která do této kategorie patří (*Regular_word*, *Regular_irregular_word*, *Irregular_word*). Pro pravidelná slovesa je navržena další entita *Regular_category* pro všechny třídy, které v dané kategorii mohou být v závislosti na čase (entita *Tense*). Pro vztah mezi entitami je potřeba použít přechodovou tabulku *Rcategory_rword*, jelikož jedno sloveso může v různých časech patřit do jiných tříd. Každá třída má poté vztah s entitou *Ending*, ve které budou uloženy koncovky s ohledem na slovesný čas a osobu (entita *Person*). Pro pravidelně nepravidelné je návrh obdobný, nachází se zde navíc pouze vztah k entitě *Change*, definující změny ve slovesech v závislosti na čase a osobě. Pro nepravidelná slovesa je navržena entita *Irregular_form*, kde budou uloženy celé tvary slovesa pro každý čas a osobu, jelikož neexistují pravidla společná pro více nepravidelných sloves. Díky těmto entitám a vztahům lze získat k libovolnému slovesu informace o koncovkách a změnách potřebných k vytvoření správného tvaru. U každého slovesa v závislosti na čase bude ukládán počet správných a špatných odpovědí ve cvičeních. Tyto informace

budou dále použity při vybírání vhodných sloves pro vytvoření zadání otázek k procvičování.

V návrhu jsou také entity *Exercise* a *Result* určené k ukládání informací o absolvovaných cvičeních. Ukládá se v nich čas a datum začátku, konce celého dokončeného cvičení a pro každé cvičení seznam otázek a jejich odpovědí. Tyto informace budou použity při tvorbě statistik dokončených cvičení. Další součástí návrhu je entita s názvem *User_exercise* pro uživatelské nastavení kritérií pro generování cvičení. Poslední částí návrhu je entita *Sentence* pro ukládání vět. Z nich se poté bude definovat zadání pro cvičení, ve kterých se bude do vět doplňovat.

3.2.2 Inicializační soubory

Důležitou částí aplikace budou také inicializační soubory. Každá entita v databázi bude mít vlastní soubor, ve kterém budou v tabulce uloženy hodnoty, které se do databáze nahrají. Vztahy mezi entitami budou řešeny pomocí cizích klíčů, jejichž hodnoty budou v souborech uloženy také. Je tedy nutné u některých entit definovat jejich identifikační hodnoty (primární klíče), jelikož musí vždy korespondovat s hodnotami cizích klíčů. U ostatních entit se mohou generovat automaticky. Inicializace bude probíhat při prvním spuštění aplikace nebo při změně verze databáze.

3.3 Generování otázek a přizpůsobivost aplikace

Tato sekce se věnuje návrhu postupu při generování otázek pro cvičení. Při generování se musí zohledňovat předešlé odpovědi a kritéria vybraná uživatelem. Jako návrh generování otázek byl vytvořen diagram na obrázku 3.2 na stránce 16.

3.3.1 Postup generování otázek

Poté co uživatel zvolí všechna potřebná kritéria pro vygenerování otázek, odešle se dotaz pro vygenerování otázek. Nejdříve se podle zvolené kategorie (pravidelné, pravidelně nepravidelné, nepravidelné) vyberou všechna slovesa, která do ní patří. Tento výběr se zmenší na přesný počet, který bude možné změnit v konfiguračním souboru podle informací z předchozích cvičení. Pokud si uživatel zvolil procvičování z věty, jsou vybírána pouze slovesa, která mají nějakou větu v databázi vytvořenou. Poté co je seznam sloves vytvořený, se podle kategorie vytvoří správný tvar slovesa. Pokud se generují otázky pro pravidelná slovesa, tak se podle slovesné třídy, slovesného času a náhodně vygenerované osoby určí koncovka pro vytvoření správného tvaru. Pokud se jedná o kategorii pravidelně nepravidelných, tak se kromě koncovky určuje také změna uvnitř slovesa. Je možné, že pro danou osobu se změna při tvoření tvaru nepoužije. Při generování správného tvaru pro nepravidelná slovesa se

3. NÁVRH

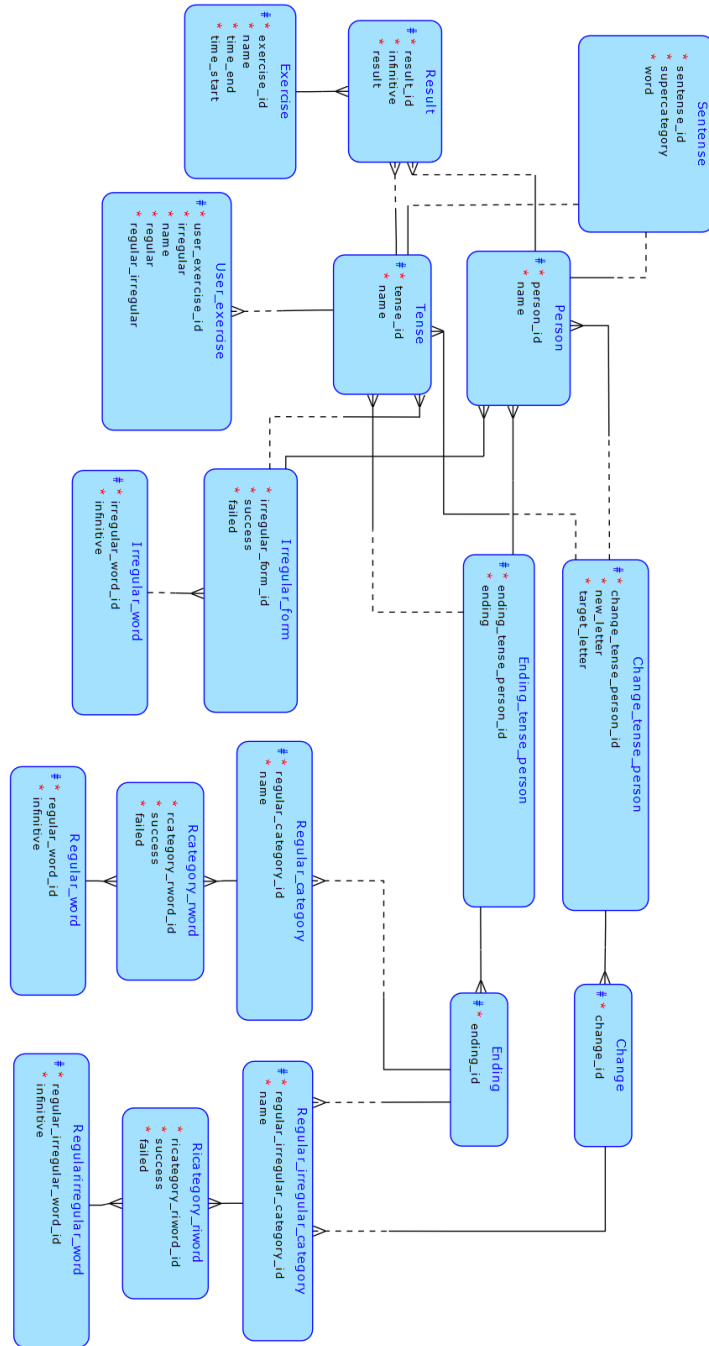
pouze podle času a osoby vybere z databáze odpovídající tvar. Nakonec se vygenerují špatné odpovědi. Pro každou obtížnost je potřeba vytvořit špatné odpovědi v jiné formě. Pokud uživatel procvičuje na lehkou obtížnost, je potřeba vygenerovat 4 špatné odpovědi. Je důležité, aby odpovědi byly podobné, aby uživateli nebyla správná odpověď hned zřejmá. Zároveň odpovědi musí dávat smysl. Při vytváření špatných odpovědí je potřeba vytvořit dvanáct písmen, ze kterých bude možné poskládat správný tvar a přebývající písmena nebudou zcela náhodná, ale bude možné z nich poskládat i jiné tvary. U těžké obtížnosti není potřeba špatné odpovědi generovat, jelikož uživatel odpověď píše na mobilní klávesnici. Nakonec se ke tvarům připojí informace o slovesném času, osobě a infinitivu, ze kterých se vytvoří zadání. V případě, že si uživatel zvolil procvičování z věty, se připojí také celá věta.

3.3.2 Přizpůsobení předešlým odpovědím

Při výběru slov, pro která se budou vytvářet otázky, se budou zohledňovat také předešlé odpovědi. Díky této funkcionalitě se uživateli budou zobrazovat více otázky, které odpověděl špatně nebo otázky, které se uživateli zatím nezobrazili. Seznam sloves bude tvořen ze tří prametrů v poměru, který bude možné jednoduše modifikovat v konfiguračním souboru aplikace. Bude tvořen ze sloves, u kterých má uživatel nejmenší úspěšnost správných odpovědí, tedy slovesa s nejmenší hodnotou rozdílu mezi počtem správných a špatných odpovědí. Dále ze sloves, která se uživateli zobrazovala nejméně, neboli slovesa s nejmenší hodnotou součtu správných a špatných odpovědí. Poslední skupinu budou tvořit slovesa vybraná náhodně. Těch by mělo být ze všech tří skupin nejméně.

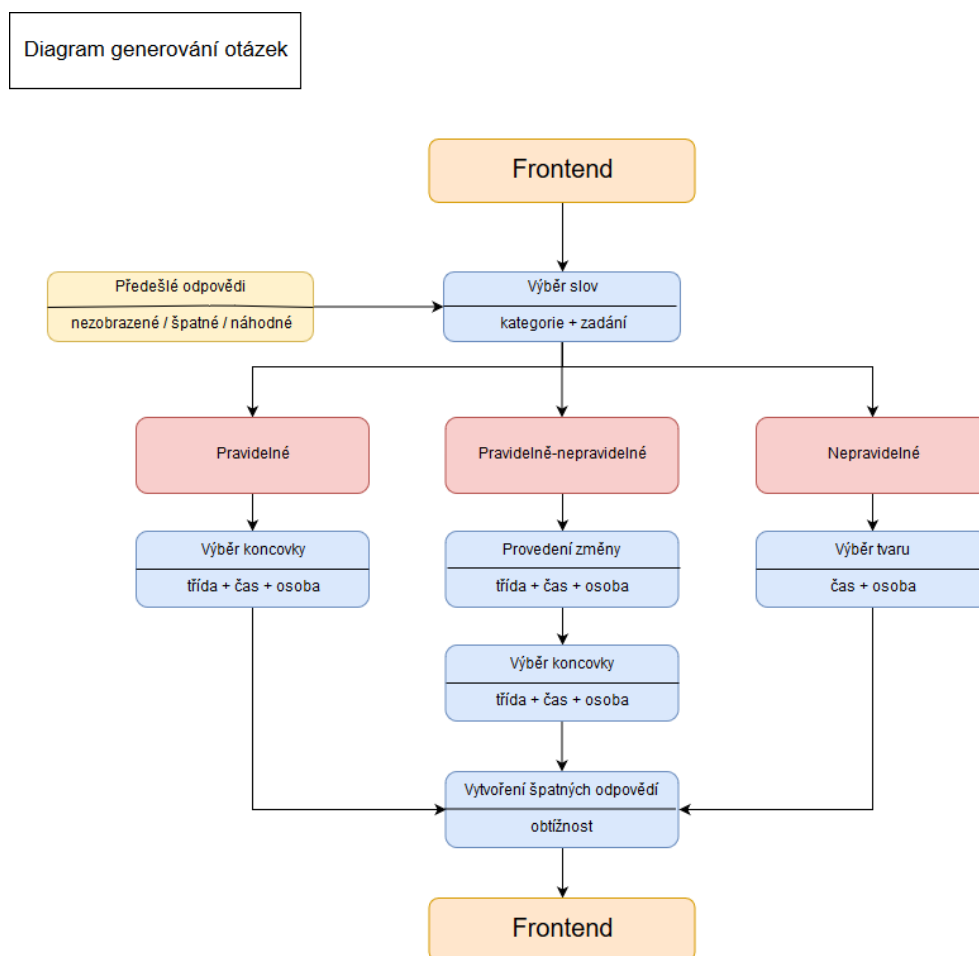
3.4 Výukový systém

Obsahem této sekce je popis návrhu výukového systému. Výuka se bude skládat z výukových materiálů a průběžných testů. Bude uspořádána podle slovesných časů a kategorií. Na začátku každého slovesného času bude krátký přehled o informacích z daného času. Následně bude představena nová kategorie a vysvětleny postupy při tvorbě správných tvarů. Následovat bude vždy krátké cvičení na právě představenou kategorii. Po dokončení všech testů z daného slovesného času se uživateli zobrazí větší opakovací test na celý slovesný čas. Ve výukovém programu bude možné přeskokovat dopředu pro případ, že uživatel již má nějaké slovesné časy a kategorie nastudované. Zároveň se bude možné vracet k již absolvovaným testům a výukovým materiálům pro případ, že si chce uživatel nějakou část zopakovat.



Obrázek 3.1: Entitní model návrhu databáze

3. NÁVRH

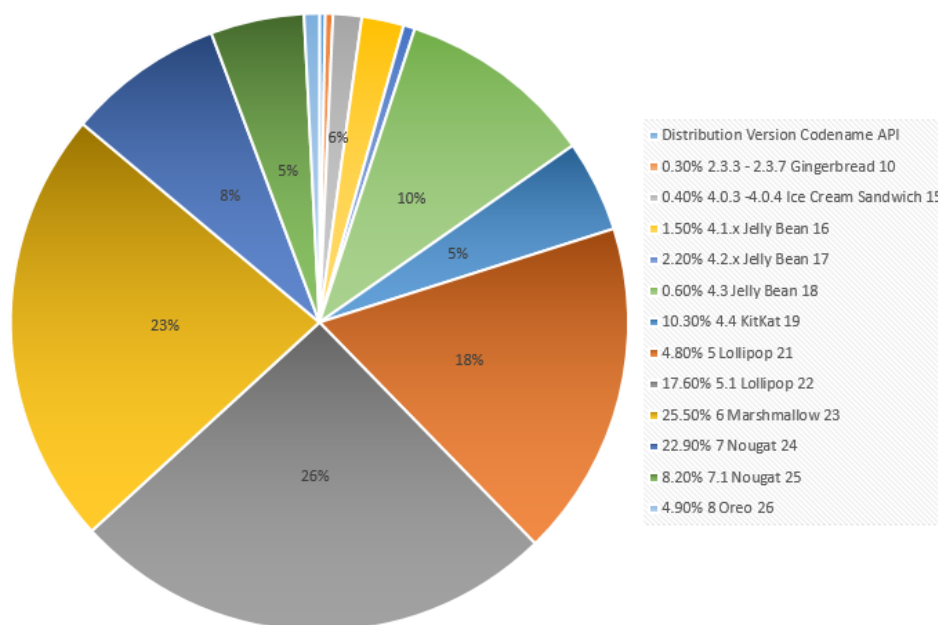


Obrázek 3.2: Diagram generování otázek pro cvičení

Implementace

4.1 Použité technologie

Jako vývojové prostředí jsme zvolili Android Studio [9] z důvodu, že se jedná o oficiální nástroj od firmy Google, která operační systém Android spravuje. Z tohoto důvodu má dobrou podporu jak od firmy Google, tak od komunity vývojářů. Android studio má podporu pro dva programovací jazyky, Java a Kotlin. Vzhledem ke zkušenostem z předmětu BI-AND, kde se učí principy programování Android aplikací pomocí jazyka Java, jsme se rozhodli použít pro tuto aplikaci stejný programovací jazyk. Pro možnost sdílení obou částí projektu a průběžnému verzování aplikace jsme použili školní webovou aplikaci GitLab [10]. Podle informací z Distribution dashboard [11] o aktuálně používaných verzích operačního systému Android (lze vidět na obrázku 4.1) jsme jako minimální podporovanou verzi zvolili 4.0.3 (Ice Cream Sandwich), abychom pokrývali co největší procento používaných verzí systému Android. Aplikace bude mít jednoduchý design, proto není nutnost používat funkce dostupné pouze v novějších verzích systému.



Obrázek 4.1: Graf zastoupení aktivních verzí systému Android

4.2 Databáze

V této sekci se budu věnovat implementaci databáze a její inicializací podle návrhu v sekci 3.2 a popisu použitých postupů a technologií.

4.2.1 Tvorba databáze

Pro správu životního cyklu databáze a jejích tabulek byla použita třída `DatabaseHelper`. Tato třída spravuje vytváření tabulek databáze při prvním spuštění aplikace a v závislosti na aktuální verzi databáze (definované číselnou hodnotou) v ní provádí úpravy. V této aplikaci to znamená zrušení některých tabulek a jejich opětovné vytvoření. Tabulky, které uchovávají data přímo z aplikace (uložené odpovědi a cvičení) nechceme při každé změně v databázi rušit, proto to musí být explicitně nastaveno.

Každá tabulka je reprezentována pomocí samostatné třídy. Tato třída obsahuje metody pro samotné vytvoření dané tabulky se všemi atributy a cizími klíči pro její bezpečné zrušení a opětovné vytvoření. Všechny názvy atributů a cizích klíčů jsou uloženy ve statické třídě `DbConstants`, aby mohly být jednoznačně určeny a při jejich změně se aktualizovala hodnota v celém kódu.

4.2.2 Přístup do databáze

Přístup do databáze je spravován pomocí třídy `ConjugataContentProvider` rozšiřující Android třídu `ContentProvider`. Použití této třídy je standardní volbou v moderních Android aplikacích, jelikož nabízí jednoduché rozhraní a díky tomu, že je oddělená od ostatních aplikačních částí, je možné od ní získávat data odkudkoliv, například i z jiných aplikací. Identifikace přístupu do databáze se provádí pomocí URI adres. Pomocí URI adresy se identifikuje tabulka, ke které chceme přistupovat a operace, kterou nad ní chceme provést. `ContentProvider` nabízí metody pro přidávání nebo mazání dat z databáze a pro vytváření předepsaných pohledů na databázi. Tyto pohledy jsou vytvářeny ve formě instancí třídy `Cursor`, díky kterým se z daného pohledu získávají hodnoty zvolených atributů. K jejich zobrazování do UI je používán `CursorLoader`. Tento `CursorLoader` běží na samostatném vlákne, tudíž při získávání většího objemu dat nezastavují chod hlavního vlákna, na kterém běží UI. Jejich hlavní výhodou je, že si data po načtení uschovávají a například v případě otočení obrazovky je nemusí načítat znovu. Zároveň při jakékoliv změně právě načtených dat z databáze se pomocí upozornění od třídy `ContentProvider` data ihned načtou znova. K vytváření objektů z `Cursor`ů je použita statická třída `ConjugataCursorDataParser`.

4.2.3 Inicializace

Inicializace probíhá pomocí `TaskFragmentu` na samostatném vlákne, aby výpočet nezpomaloval hlavní vlákno aplikace. Inicializace probíhá při prvním spuštění aplikace nebo při změně verze databáze na samostatné obrazovce. Průběžně po načtení každého souboru jsou aktivitě `InitializeActivity`, spravující inicializační obrazovku, posílány informace o procentuálním postupu inicializace. Každá tabulka má vlastní soubor, ve kterém jsou uložena data, která se do tabulky načítají. Některé tabulky, například ty obsahující seznam sloves, mají kvůli přehlednosti více souborů, ze kterých se data načítají. Soubory jsou uloženy ve složce `assets` (rozdělené do složek) ve formátu `CSV`, ze kterého je možné data zobrazovat a editovat v tabulkových procesorech. Na prvním řádku každého inicializačního souboru je seznam atributů (oddělených čárkou) totožný s názvy atributů v databázi (třída `DbConstants`). Každý záznam (řádek v databázi) je psán na samostatném řádku v souboru a pořadí jeho atributů musí odpovídat pořadí názvů atributů na prvním řádku souboru. Pokud má nějaký atribut pro všechny záznamy v souboru stejnou hodnotu, je možné za název atributu napsat `=` a jeho hodnotu. Tato hodnota se přiřadí všem záznamům, tudíž není nutné ji zmiňovat na každém řádku. Tato implementace inicializačních souborů dovoluje jednoduché přidávání nových dat.

4.3 Generátor otázek

O generování otázek se stará samostatná třída `QuestionGenerator`. Tato třída obsahuje veřejnou metodu pro generování otázek. Jako parametry požaduje všechny informace potřebné k vytvoření otázek a vrací list objektů třídy `Question`. Tato třída je abstraktní a implementují ji tři další třídy `EasyQuestion`, `MediumQuestion` a `HardQuestion`. Ty se liší v atributech a metodách, které jsou požadované pro odpovídající obtížnost odpovídání (lehkou, střední, těžkou). Z výsledného listu otázek se poté vytvoří samotné cvičení, lze vidět na obrázku 4.2c. Postup generování je implementován podle návrhu v sekci 3.3. Při výběru vhodných slov k procvičování se zohledňují předchozí odpovědi uživatele. Původně se zpracovávala data všech odpovědí a podle nich se vybírala vhodná slovesa. Při velkém objemu uložených odpovědí však tento proces trval příliš dlouho, proto jsou nyní data o úspěšnosti odpovědí uložena u samotných sloves. Díky tomu je výpočet stabilní a neztrácí na rychlosti dlouhodobějším používáním aplikace.

Další zajímavou implementační otázkou bylo generování špatných odpovědí. Při tvorbě špatných odpovědí pro lehkou obtížnost (výběr z pěti tvarů slovesa) se u pravidelných sloves z databáze vyberou náhodné koncovky. Pro tyto účely byly do databáze přidány „falešné koncovky“, které mají oproti ostatním pouze malou změnu. Například z „-emos“ vznikne falešná koncovka „-amos“. Výsledný tvar poté vypadá tak, že by mohl být správnou odpovědí. U pravidelně nepravidelných sloves jsou takto generovány i „falešné změny“ uvnitř sloves. U nepravidelných sloves jsou jako špatné odpovědi vybírány tvary i z ostatních časů, které často vypadají podobně jako správný tvar. Pro střední obtížnost (sestavení odpovědi z dvanácti písmen) se písmena, kromě těch, která jsou použita k vytvoření správné odpovědi, vybírají tak, aby bylo možné vytvořit co nejvíce tvarů a písmena se často neopakovala. Tato implementace by měla nutit uživatele při výběru odpovědi více přemýšlet a při následném používání tvarů v praxi dělat méně chyb.

4.4 Výuka

Výuka je spravována třídou `LearningController`, která je navržena tak, aby přidávání dalších prvků výuky (výukový materiál, nebo test) bylo co nejjednodušší. Třída obsahuje seznam prvků pro výuku, které se při otevření obrazovky ze třídy načtou a zobrazí. Prvky jsou implementací abstraktní třídy `LearnItem`. Prvek definující výukový materiál je reprezentován třídou `LessonItem` a obsahuje cestu k souboru, ve kterém je uložen výukový materiál ve formátu HTML. Prvek výuky představující testovou úlohu je reprezentován třídou `QuizItem`. Ta v sobě uchovává informace o slovesném času, kategorii a seznam tříd, podle kterých se má test vygenerovat. Obdobou této třídy je `BigQuizItem` který představuje velký opakovací test pro celý slovesný čas.

Ukládá se také aktuální postup ve výukovém programu a podle něj se na obrazovce (obrázek 4.2b) zvýrazní prvek, který by měl uživatel splnit jako další. Zároveň uživatele postup výukou neomezuje a může si tak otevřít libovolný prvek výuky. Při vytváření výukových materiálů se čerpalo z webových stránek [12] a [13].

4.5 Konfigurace aplikace

Backend aplikace je navržen tak, aby bylo možné některé parametry pro generování otázek jednoduše upravovat. Ve frontendové části je vše generováno dynamicky a tak změny nezpůsobí v UI žádné problémy. Informace o konfiguraci se nachází v samostatné třídě `ModelConfiguration`. V této třídě je možné například nastavit maximální počet otázek ve cvičení (testu). Omezení je pouze shora z důvodu, že pokud se generuje cvičení pouze pro jednu specifickou třídu, často do této třídy mnoho sloves nepatří a nechceme, aby se otázky opakovaly. Pro účely testování a prezentace výsledné aplikace byla tato hodnota nastavena na pět. Dalším parametrem, který lze nastavit, je poměr tří hodnot podle kterých se vybírají vhodná slovesa k procvičení. První je počet sloves, u kterých má uživatel nejmenší úspěšnost správných odpovědí. Druhá jsou ta, která se uživateli zatím zobrazila nejméně. Poslední je počet náhodně vybraných sloves. Aktuálně jsou tyto hodnoty nastaveny v poměru 3:2:1. Výběr probíhá tak, že se slovesa seřadí podle daného kritéria a do výsledného výběru se přidá počet definovaný hodnotou poměru a celkového počtu otázek pro cvičení.

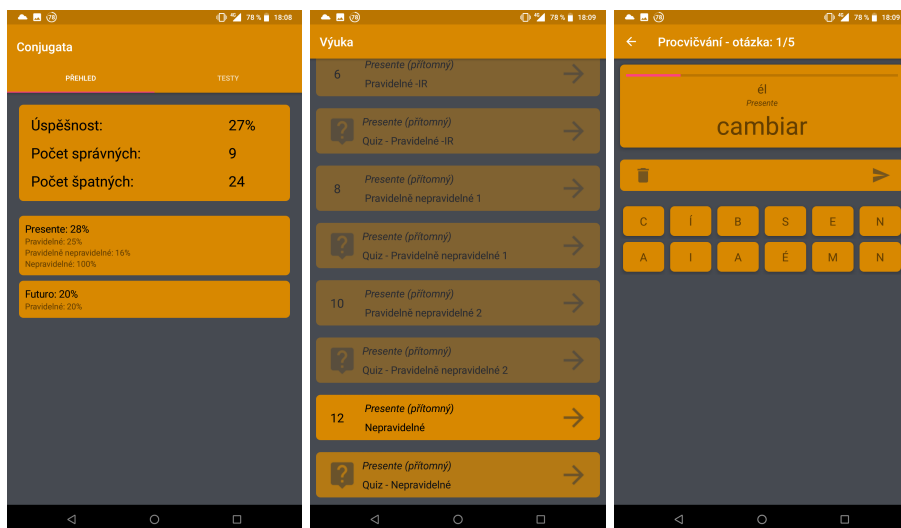
4.6 Propojení frontendu a backendu

Aplikace byla vyvíjena v úzké spolupráci se studentem ČVUT FIT Matějem Sedlákem, který pracuje na frontendové části aplikace. To bylo důležité hlavně v implementační části z důvodu toho, aby spolu obě části (frontend a backend) dobře komunikovaly. Pro komunikaci mezi nimi byla vytvořena třída `ConjugataModel`. Tato třída zajišťuje veškeré operace, které UI vyžaduje. První z nich je generování otázek popsané v sekci 4.3. Pro vygenerování otázek je implementována veřejná metoda `getQuestions`. Tato metoda je přetížená a má tři možnosti jejího volání. Je možné ji zavolat s parametry pro generování otázek pro jednu ze tří kategorií (pravidelné, pravidelně nepravidelné, nepravidelné) nebo pro specifické seznamy tříd z každé kategorie. Poslední možností je generování pro uložená uživatelská kritéria.

Další metodou této třídy je `saveQuestionResult`. Tato metoda uloží seznam zaznamenaných otázek a odpovědí do databáze pro další použití ve statistikách (obrázek 4.2a) nebo při výběru vhodných sloves k procvičování. Zbylé metody se týkají uložení uživatelských kritérií. První s názvem `existUserExercise` zjistí, jestli se již kritéria s daným názvem v databázi nenacházejí. Pomocí

4. IMPLEMENTACE

druhé metody *setUserExercise* se poté do databáze kritéria uloží. Záznamy z databáze se pro UI načítají pomocí třídy *CursorLoader*, popsáno v sekci 4.2.2.



(a) Snímek statistik (b) Výuková obrazovka (c) Obrazovka cvičení

Obrázek 4.2: Snímky obrazovek z výsledné aplikace

Závěr

V této práci byl vytvořen backend pro výukovou aplikaci cizích jazyků na operační systém Android a úspěšně propojena s frontendovou částí, která byla tématem bakalářské práce Matěje Sedláka [14]. Výsledně tak vznikla aplikace dostupná na Google Play [15].

Aplikace spravuje ukládání a získávání dat z databáze, implementuje systém generování procvičovacích testů, ukládá odpovědi a podle nich přizpůsobuje testování danému uživateli, který aplikaci používá. Aplikace obsahuje výukový systém, který uživatele provádí pravidly časování sloves ve španělském jazyce, s průběžnými procvičovacími úlohami. V aplikaci je také k dispozici přehled statistik, obsahující procentuální úspěšnost v závislosti na čase a kategorii. A také historii absolvovaných cvičení s možností shlédnutí špatných a správných odpovědí.

Apliacce byla vyvíjena jako prototyp. Proto existuje spousta možností k jejímu rozšíření. Hlavním rozšířením je doplnění ostatních slovesných časů španělského jazyka a vytvoření výuky pro tyto časy. Dále zvětšení databáze sloves pro různé kategorie a třídy a tím zvýšit rozmanitost vygenerovaných otázek. Aplikace byla vyvíjena tak, aby bylo možné tyto rozšíření vytvářet co nejjednodušeji. Poté by bylo možné rozšířit aplikaci pro další světové jazyky, nejlépe o další latinské jazyky, jelikož mají podobné principy časování jako právě implementovaná španělština.

Další možností by bylo vylepšit výukový program, který by pracoval s reálným časem. Například připomínal uživateli každý den v určitou dobu, aby se studiem pokračoval nebo ho upozornil, že by bylo vhodné si dnes zopakovat určitou kategorii, kterou se před nedávném naučil. Bylo by také možné rozšíření modelu postupu. Vytvořit propracovanější systém získávání zkušenostních bodů a úrovní. Tyto informace by bylo možné sdílet s ostatními uživateli a podporovat se tak vzájemně při dalším studiu. Je možné také vytvořit připojení se serverem, který by zaznamenával data všech uživatelů. Tyto informace by mohly pomoci dalšímu rozvoji aplikací v tomto odvětví.

Literatura

- [1] Number of Android Applications [online]. 28 duben 2018, [cit. 2018-25-04]. Dostupné z: <https://www.appbrain.com/stats>
- [2] Google Inc.: Developer Guides [online]. [cit. 2018-25-04]. Dostupné z: <https://developer.android.com/guide/>
- [3] THE SPANISH VERBO COMPANY: Verbo 1.0.12 [software]. [cit. 2018-25-04]. Požadavky na systém: Android 4.0.3 a vyšší. Dostupné z: <https://play.google.com/store/apps/details?id=es.hubiquus.verbo>
- [4] Jazykové úrovně dle Společného evropského referenčního rámce [online]. [cit. 2018-14-05]. Dostupné z: <http://www.jazykove-urovne.cz/>
- [5] Geoglot Apps: Spanish Verb Blitz 1.4.15 [software]. [cit. 2018-25-04]. Požadavky na systém: Android 4.1 a vyšší. Dostupné z: <https://play.google.com/store/apps/details?id=com.rusmate.verbblitzspanish>
- [6] Himalaio: Spanish Grammar: Tenses 1.0 [software]. [cit. 2018-25-04]. Požadavky na systém: Android 4.0 a vyšší. Dostupné z: <https://play.google.com/store/apps/details?id=com.himalaio.spanish>
- [7] Learn Spanish Free - Online Lessons [online]. [cit. 2018-11-05]. Dostupné z: <http://www.completespanishguide.com/>
- [8] ČVUT FIT: Data modeller [software]. [cit. 2018-15-04]. Dostupné z: <https://dbs.fit.cvut.cz/dbsdm/>
- [9] Google Inc.: Android Studio 3.1.2 [software]. [cit. 2018-05-04]. Dostupné z: <https://developer.android.com/studio/>
- [10] Gitlab FIT [software]. [cit. 2018-12-04]. Dostupné z: <https://gitlab.fit.cvut.cz/>

LITERATURA

- [11] Distribution dashboard [online]. květen 2018, [cit. 2018-05-12]. Dostupné z: <https://developer.android.com/about/dashboards/>
- [12] Španělština-zdarma [online]. únor 2017, [cit. 2018-05-08]. Dostupné z: <http://www.spanelstina-zdarma.cz/>
- [13] estudiante.cz – španělština od nuly [online]. [cit. 2018-05-08]. Dostupné z: <http://estudiante.cz/>
- [14] Sedlák, M.: Conjugata - aplikace pro cvičení časování sloves - frontend. Praha, 2018. Bakalářská práce (Bc). ČVUT FIT, Katedra softwarového inženýrství.
- [15] Petr Polívka a Matěj Sedlák: Conjugata 1.0 [software]. [cit. 2018-11-05]. Požadavky na systém: Android 4.0.3 a vyšší. Dostupné z: <https://play.google.com/store/apps/details?id=cz.cvut.fit.bp.conjugata>

Seznam použitých zkratk

URI Uniform Resource Identifier

UI User Interface

CSV Comma-separated values

HTML Hypertext Markup Language

APK Android application package

Obsah přiloženého CD

readme.txt	stručný popis obsahu CD
apk	
└─ conjugata.apk	spustitelný soubor aplikace pro Android
src	
└─ impl	projekt obsahující zdrojové kódy implementace
└─ thesis	zdrojová forma práce ve formátu \LaTeX
doc	
└─ uzivatelska_prirucka.pdf	uživatelská příručka ve formátu PDF
└─ conjugata_dokumentace	dokumentace ke zdrojovým kódům
text	
└─ thesis.pdf	text práce ve formátu PDF