

Mobilna aplikacija za samostalno učenje odabranih gramatičkih struktura

Makaj, Patrik

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:771658>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-23**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA OSIJEK**

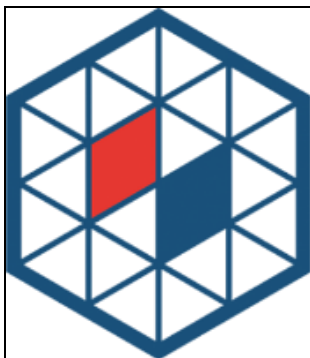
Sveučilišni studij računarstva

**Mobilna aplikacija za samostalno učenje odabranih
gramatičkih struktura**

Završni rad

Patrik Makaj

Osijek, 2023.



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA **OSIJEK**

Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju

Osijek, 15.09.2023.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Patrik Makaj
Studij, smjer:	Sveučilišni prijediplomski studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R 4391, 22.07.2019.
OIB Pristupnika:	08490036351
Mentor:	doc. dr. sc. Dragana Božić Lenard
Sumentor:	prof. dr. sc. Krešimir Nenadić
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna aplikacija za samostalno učenje odabranih gramatičkih struktura
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rad:	U sklopu će se završnoga rada izraditi android aplikaciju za usavršavanje odabranih gramatičkih struktura engleskog jezika. Gramatičke će se strukture birati u skladu s nastavnim planom i programom triju kolegija Engleskoga jezika za posebne svrhe koji se predaju na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija Osijek, stoga su studenti FERIT-a ciljani krajnji korisnici, no aplikacijom će se moći poslužiti i drugi korisnici koje budu zanimala odabrana gramatika poglavlja. Aplikacija će se izraditi u programskom jeziku Kotlinu koristeći Android Studio okruženje. Na početnoj će stranici korisnik odabrati gramatičku cjelinu nakon čijeg će proučavanja teorijske osnove moći riješiti test za provjeru znanja na temelju čijih će rezultata korisniku biti predložena dodatna literatura/naputci za dodano učenje. Rezervirano za Patrika Makaja Sumentor s FERIT-a Krešimir Nenadić
Prijedlog ocjene završnog rada:	Vrlo dobar (4)

Kratko obrazloženje ocjene prema Kriterijima za ocjenjivanje završnih i diplomskih radova:	Primjena znanja stečenih na fakultetu: 3 bod/boda Postignuti rezultati u odnosu na složenost zadatka: 2 bod/boda Jasnoća pismenog izražavanja: 2 bod/boda Razina samostalnosti: 2 razina
Datum prijedloga ocjene od strane mentora:	15.09.2023.
Datum potvrde ocjene od strane Odbora:	24.09.2023.
Potvrda mentora o predaji konačne verzije rada:	Mentor elektronički potpisao predaju konačne verzije.
	Datum:



FERIT

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

IZJAVA O ORIGINALNOSTI RADA

Osijek, 25.09.2023.

Ime i prezime studenta:

Patrik Makaj

Studij:

Sveučilišni prijediplomski studij Računarstvo

Mat. br. studenta, godina upisa:

R 4391, 22.07.2019.

Turnitin podudaranje [%]:

7

Ovom izjavom izjavljujem da je rad pod nazivom: **Mobilna aplikacija za samostalno učenje odabranih gramatičkih struktura**

izrađen pod vodstvom mentora doc. dr. sc. Dragana Božić Lenard

i sumentora prof. dr. sc. Krešimir Nenadić

moj vlastiti rad i prema mom najboljem znanju ne sadrži prethodno objavljene ili neobjavljene pisane materijale drugih osoba, osim onih koji su izričito priznati navođenjem literature i drugih izvora informacija.

Izjavljujem da je intelektualni sadržaj navedenog rada proizvod mog vlastitog rada, osim u onom dijelu za koji mi je bila potrebna pomoć mentora, sumentora i drugih osoba, a što je izričito navedeno u radu.

Potpis studenta:

SADRŽAJ

1. UVOD	6
1.1. Zadatak rada	6
1.2. Motivacija za odabir tehnologija i programskog jezika	7
2. PREGLED PODRUČJA I SLIČNIH GOTOVIH RJEŠENJA	8
2.1. Duolingo.....	9
2.2. Rosetta Stone.....	10
2.3. Babbel.....	11
2.4. Memrise.....	12
2.5. Busuu.....	13
2.6. Osvrt na postojeća rješenja.....	14
3. POSTUPAK RAZVOJA APLIKACIJE	15
3.1. Android Studio i Kotlin.....	15
3.2. Arhitektura i povezanost slojeva	17
3.3. RecyclerView	17
3.4. XML	18
3.5. Cloud Firestore	19
3.6. Radio grupa i gumbi	20
3.7. Objekt Activity	21
3.8. Objekt Intent.....	22
3.9. Kviz	22
4. PRIKAZ FUNKCIONALNOSTI APLIKACIJE	24
4.1. Glavna aktivnost.....	24
4.2. Kviz aktivnost	25
4.3. Scores aktivnost.....	26
4.4. Info aktivnost.....	27
4.5. Usporedba aplikacije s rješenjima navedenim u poglavlju	28
5. ZAKLJUČAK	29

1. UVOD

U današnje vrijeme, znanje stranog jezika potreba je svakoga tko želi napredovati na tržištu rada. Znanje stranih jezika otvara jako puno mogućnosti. Nelson Mandela ovaj je fenomen objasnio s "*If you talk to a man in a language he understands, that goes to his head. If you talk to him in his own language, that goes to his heart.*" Učenje stranog jezika od početka vrlo je težak i vremenski zahtjevan proces.

Pametni telefon jedan je od najvećih izuma prošlog stoljeća. Danas ga posjeduje skoro svatko, a njegove su mogućnosti bezbrojne; od slanja jednostavnih SMS poruka do instaliranja raznih aplikacija s nikad prije mogućim opcijama poput mjerenja udaljenosti, računanja kompleksnih matematičkih operacija pa čak i prevođenja bilo kojeg jezika. Jasno je da mobilni telefoni uvelike olakšavaju život kroz razne mogućnosti i aplikacije. *Mobile-assisted language learning* (MALL) je pojam kojim stručnjaci opisuju učenje stranih jezika potpomognuto mobilnim uređajem. Postoji velik broj aplikacija toga tipa široke namjene dostupne korisnicima na tržištu o kojima se govori u idućem poglavlju, ali većina tih aplikacija ima mane.

U radu su opisane sve korištene tehnologije. Nabrojane su brojne prednosti Androida, Kotlina i Android Studio programa korištenih pri izradi aplikacije. Iz tih prednosti moguće je vidjeti zašto su se koristile baš te tehnologije, npr. na slici 1.4. vidljivo je zašto je izabran Kotlin kao programski jezik za kreaciju aplikacije umjesto Jave. Aplikacija je kreirana baš na Android mobilnom sustavu, jer je prema podacima u poglavlju 1.2. on jedan od najdostupnijih sustava.

U drugom su poglavlju opisane slične aplikacije uz osvrt na prednosti i nedostatke svake od njih. Treće poglavlje govori o postupku izrade aplikacije uz opisivanje i primjere korištenih tehnologija. Četvrto poglavlje prikazuje funkcionalnost aplikacije, sučelja vidljiva korisniku te usporedbu aplikacije sa sličnim rješenjima dostupnim na tržištu. Zadnje je poglavlje zaključak u kojem se kratko rezimiraju motivacija za izradu, svrha i prednosti ove aplikacije.

1.1. Zadatak rada

Zadatak je ovog završnog rada izraditi Android aplikacije u Kotlin programskom jeziku koristeći programsko okruženje Android Studio, čija je svrha omogućiti korisniku jednostavno i interaktivno učenje gramatičkih struktura. Cilj je ove aplikacije, uz interaktivne predloške i kviz pitanja, poučiti korisnika određenim gramatičkim strukturama engleskog jezika potpuno besplatno. Na sučelju aplikacije korisnik bira određeni dio gradiva i točne gramatičke probleme

koje želi savladati. Nakon odabira gradiva, navedena su pravila o uporabi i korektni primjeri korištenja dane gramatičke strukture u praksi. Potom korisnik dobiva nekoliko kviz pitanja koja mu daju povratnu informaciju o tome koliko je dobro savladao tu gramatičku strukturu. Naposljetku dobije postotak po gramatičkoj strukturi kako bi imao točnu povratnu informaciju koliko je dobro savladao tu gramatičku strukturu. Aplikacija bi trebala biti odličan alat za savladavanje gramatičkih struktura engleskog jezika u kratkom vremenu uz zabavno i interaktivno sučelje koje prati korisnikov napredak.

Aplikacija treba imati mogućnost prikazivanja gramatičkih pravila te kviz pitanja kojima se provjerava savladanost gradiva. Aplikacija je napravljena tako da ostavlja mogućnost dodavanja novih gramatičkih struktura. Razvijena je za operacijski sustav Android u programskom jeziku Kotlin. Korištena su znanja Kotlin programskog jezika te svih koncepata koji se koriste u programiranju. Također, potrebno je znanje Android Studio programskog okruženja u kojem je kreirana aplikacija.

1.2. Motivacija za odabir tehnologija i programskog jezika

Korištenje Android sustava olakšava život korisnicima. Primjerice, uporabom GPS-a izbjegava se prometna gužva, pametni sat može slati poruke, a asistent odgovarati na pitanja. To je operacijski sustav instaliran na 2,5 milijarde aktivnih uređaja krećući se od 5G telefona do tableta. *Google Play Protect* skenira sve aplikacije, a softver dobiva redovita sigurnosna ažuriranja i platforma se stalno poboljšava. Android je otvorena platforma za sve developere i kreatore aplikacija. Na Android platformi beskrajne su mogućnosti stvaranja aplikacija.

Prema podacima StatCounter-a [1] iz lipnja 2021. godine, Android je najrasprostranjeniji mobilni operacijski sustav na svijetu. Prema izvješću StatCounter-a Android je imao udio od gotovo 73 % na globalnoj razini, dok je iOS imao oko 26 %. Prema istraživanju za 2020. godinu, više od 1,3 milijarde pametnih telefona prodanih u toj godini bilo je opremljeno Android operacijskim sustavom. To predstavlja impresivnih 85,4 % tržišnog udjela. Android je također popularan u mnogim zemljama u razvoju. Otvoren kod Androida omogućava proizvođačima prilagođavanje Androida svojim uređajima. To rezultira raznovrsnim hardverskim opcijama i prilagođenim korisničkim sučeljima. Otvoren kod također potiče inovaciju jer omogućava programerima da prilagođavaju i unaprjeđuju Android sustav. To rezultira raznovrsnim aplikacijama, funkcionalnostima i poboljšanjima u korisničkom iskustvu. Zbog toga privlači veliku zajednicu programera koja doprinosi razvoju i poboljšanju Androida. To znači da postoje

mnoge aplikacije, alati i resursi dostupni za korisnike. Android je besplatan za korištenje, što znači da proizvođači i programeri mogu lako pristupiti sustavu i razvijati aplikacije bez financijskih ograničenja.

2. PREGLED PODRUČJA I SLIČNIH GOTOVIH RJEŠENJA

Na tržištu postoji jako velik broj aplikacija za učenje stranih jezika koje su dostupne skoro svakom korisniku pametnog telefona. U idućih pet poglavlja opisan će se način funkcioniranja pet najpoznatijih aplikacija te analizirati i usporediti njihove prednosti i nedostatke.

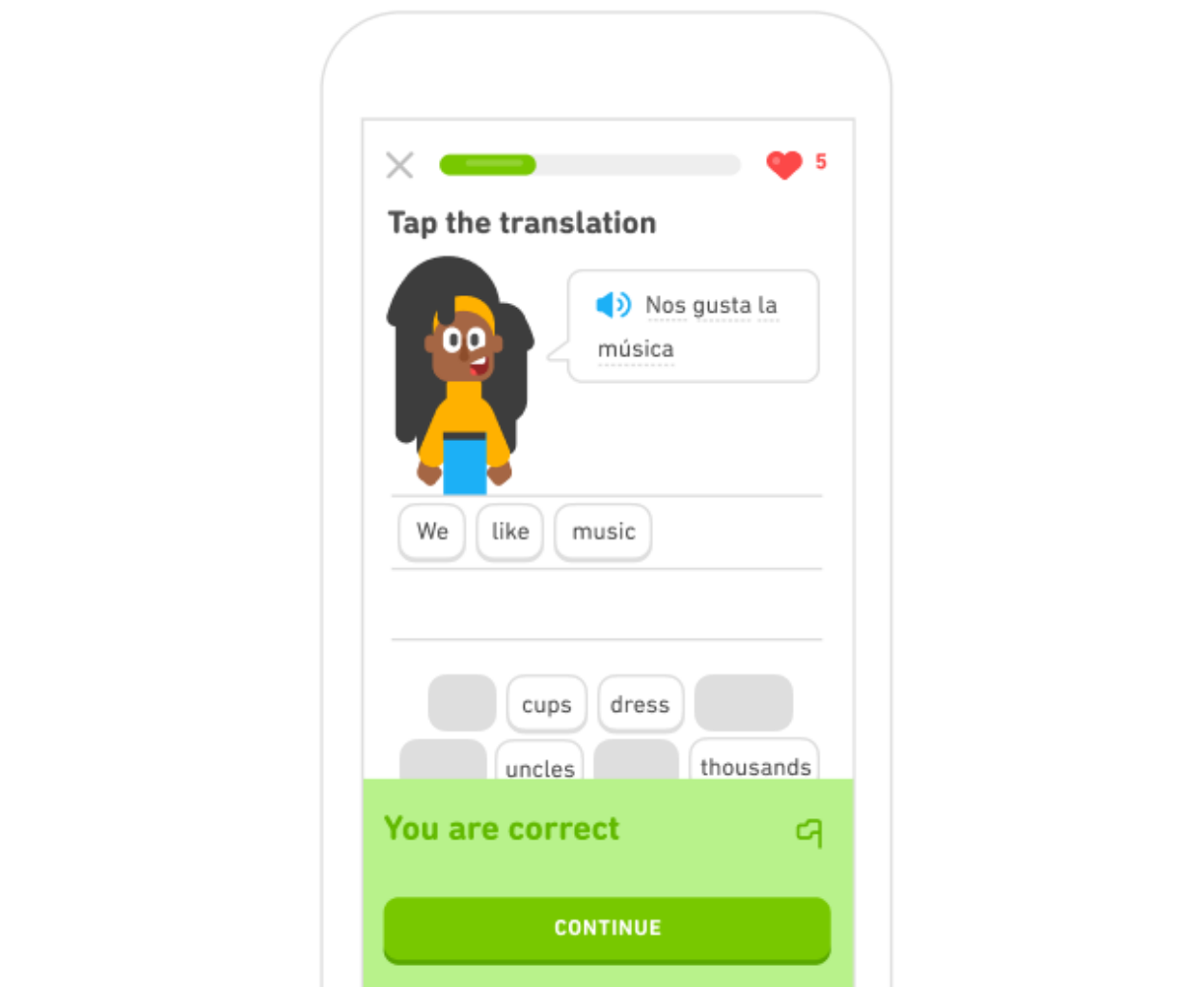
2.1. Duolingo

Duolingo je popularna aplikacija za učenje stranih jezika koja nudi nekoliko funkcionalnosti. Korisnici mogu odabrati 40 različitih jezika i vježbi putem mobilne aplikacije ili mrežne stranice. Aplikacija ima interaktivne lekcije, kvizove i igre kako bi učenje jezika učinila zabavnim iskustvom. [2] Većina se lekcija zapravo i temelji na igrama ili aktivnostima poput slušanja kratkih priča, naručivanja hrane, razgovora o sportu itd. Gramatika se uči putem kratkih testova. *Duolingo* koristi progresivno učenje prilagođavajući se korisnikovom napretku i pružajući zadatke za ponavljanje gradiva. Na samom početku korisnik dobije opciju odabira gdje se svrstava u početnike ili korisnike koji su već učili taj jezik. Rješavanjem igara i testova korisnik dobiva bodove s kojima prelazi razine koje označavaju poznavanje jezika. Najveća je razina 25 za koju je u prosjeku potrebno 387.5 sati.

Prednosti su korištenja *Duolinga* njegova dostupnost jer je aplikacija besplatna, pristupačnost i jednostavno korisničko sučelje. Aplikacija je idealna za početnike i omogućuje korisnicima da uče jezik vlastitim tempom. Također, aplikacija pruža mogućnost vježbanja čitanja, pisanja, slušanja i govora, što pomaže u razvoju sveobuhvatnih jezičnih vještina.

Međutim, *Duolingo* ima i neke nedostatke. Iako pruža početno usvajanje jezika, može se primijetiti da se naglasak i izgovor ne ističu dovoljno u *Duolingovim* vježbama na način da se nakon svake nove naučene riječi putem zvučnika reproducira točan izgovor riječi. Osim toga, korištenje *Duolinga* može postati monotono nakon dugotrajnog korištenja jer se neke lekcije previše ponavljaju. Također, napredniji korisnici mogu dobiti osjećaj da aplikacija ne pruža dovoljno izazova i dublje razumijevanje jezika. Unatoč navedenim nedostacima, *Duolingo* je i dalje vrijedan alat za početno učenje stranih jezika jer pruža mogućnost usvajanja osnovnih vještina te daje motivaciju za daljnje učenje. Aplikacija je popularna zbog svoje jednostavnosti, dostupnosti i zabavnog pristupa učenju jezika. Sama jednostavnost aplikacije je vidljiva na slici 2.1. koja prikazuje jednostavno sučelje za učenje prijevoda rečenice.

1. Complete new lessons



Sl. 2.1. Prikaz sučelja Duolingo aplikacije

2.2. Rosetta Stone

Rosetta Stone je aplikacija za učenje stranih jezika koja korisnike izlaže usvajanju jezika na prirodan način. Aplikacija pruža interaktivne lekcije, vježbe izgovora, audio i video materijale kao što je vidljivo na slici 2.2.

Prednost korištenja *Rosetta Stonea* jest to što se naglasak stavlja na slušanje i govor, što je korisno za razvijanje fluentnosti. To se postiže takozvanim *Live Tutoringom*, gdje korisnik zapravo dobije 25 minuta poziva s učiteljem koji je izvorni govornik jezika koji ispravlja pogreške i daje savjete za izgovor. [3] Aplikacija nudi širok spektar jezika za učenje, a cjeline su strukturirane da korisnik napreduje od osnovnih do naprednih razina. Postoji čak 5 razina po jeziku od kojih se svaka dijeli na 20 cjelina s 4 modula (slušanje, pričanje, izgovor i pisanje).

Međutim, *Rosetta Stone* ima nekoliko nedostataka, a jedna od njih je visoka cijena. Aplikacija je nešto skuplja u usporedbi s drugim dostupnim opcijama za učenje jezika. Plaća se 349 eura za doživotnu licencu, odnosno 11.99 eura mjesečno za tromjesečnu pretplatu. Također, postoji mogućnost nedostatka interakcije s instruktorima ili drugim učenicima. S obzirom da instruktori dobiju 6 korisnika po lekciji od 25 minuta, teško je svakom posvetiti dio vremena i prilagoditi se korisnikovoj razini znanja.

Rosetta Stone ima kvalitetan audio i vizualni sadržaj koji pruža autentičan jezični doživljaj. Potiče se prirodno usvajanje jezika, a aplikacija naglašava i razumijevanje konteksta koristeći slike i situacije. Učenje je fleksibilno jer korisnici mogu pristupiti lekcijama putem mobilnih uređaja i računala. Ova je aplikacija posebno pogodna za one koji preferiraju sustavan pristup učenju i žele razviti temeljne jezične vještine.



Sl. 2.2. Prikaz sučelja Rosetta Stone aplikacije

2.3. Babbel

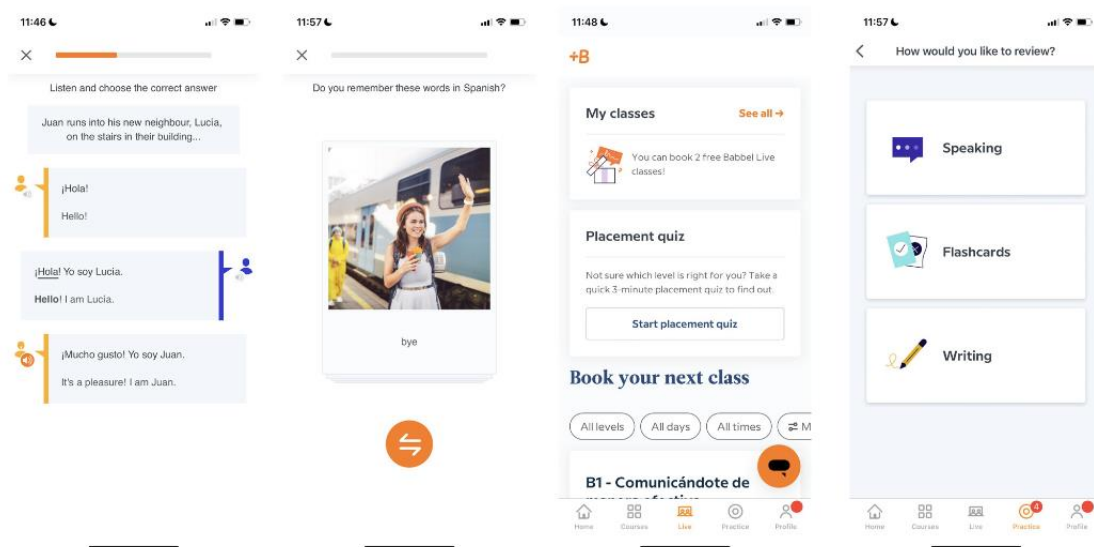
Babbel je aplikacija za učenje stranih jezika koja se fokusira na praktično učenje jezika pružajući korisne fraze i riječi koje se mogu primijeniti u svakodnevnim situacijama.

Prednosti su korištenja *Babbela* njegova jednostavnost i prilagodljivost. Aplikacija nudi širok izbor jezika za učenje i sadrži interaktivne vježbe, kratke priče i mogućnost vježbanja izgovora ponavljanjem riječi i fraza reproduciranih na zvučniku uređaja. *Babbel* podržava učenje 14 jezika. [4] Cijena je 599.99 eura za doživotan pristup učenju svih dostupnih jezika, odnosno 6.55 eura mjesečno za tromjesečnu pretplatu. *Babbel* omogućuje korisnicima učenje vlastitim ritmom i prilagođavanje lekcija korisnikovim ciljevima i razini znanja. Postoji 15 internih razina prema kojima se svrstavaju korisnici te se napredovanjem kroz te razine može pratiti proces učenja.

Također, pruža mogućnost vježbanja čitanja, pisanja, slušanja i govora pomažući u razvijanju sveobuhvatnih jezičnih vještina. Slika 2.3. jasno prikazuje funkcionalnosti aplikacije *Babbel*.

Jedan je od nedostataka *Babbela* temeljenje lekcija na unaprijed definiranim situacijama i nedostatak prilagodljivosti za personalizirano učenje. Također, neke napredne razine učenja nisu dostupne za sve jezike.

Babbel je koristan alat za brzo usvajanje osnovnih jezičnih vještina i razumijevanje konkretnih situacija. Aplikacija je pogodna za one koji žele praktično primijeniti jezik u stvarnom svijetu i brzo se pripremiti za putovanja ili komunikaciju na stranom jeziku.



Sl. 2.3. Prikaz sučelja Babbel aplikacije

2.4. Memrise

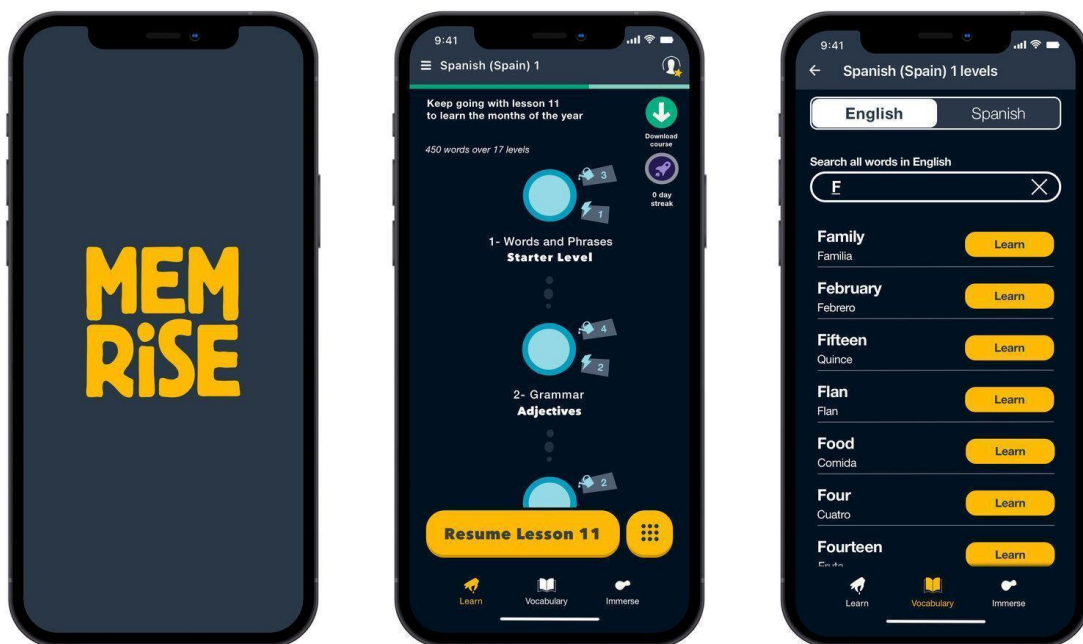
Memrise je aplikacija za učenje stranih jezika koja korisnicima nudi različite funkcionalnosti. Aplikacija sadrži bogatu bazu podataka jezičnih tečajeva s velikim brojem lekcija, rječnika i interaktivnih vježbi. Nudi materijale za učenje čak 23 jezika. [5]

Prednost je korištenja *Memrisea* u naglašavanju memoriranja riječi i izraza korištenjem vizualnih i auditivnih metoda. Aplikacija koristi ponavljanje i testiranje za jačanje memorije, a korisnicima omogućuje i stvaranje vlastitih lekcija kombinacijom unaprijed stvorenih lekcija. To je dosta korisno za učenje vokabulara jer si korisnik na isto mjesto može postaviti nekoliko sličnih dijelova gradiva. *Memrise* pruža mogućnost učenja jezika putem videozapisa, autentičnih materijala i izgovora izvornih govornika, što pomaže u razvoju slušanja i izgovaranja. Ono što ovu

aplikaciju razlikuje od sličnih jest učenje jezika kroz dijaloge s *chatbotom* u određenim situacijama kao što su npr. naručivanje kave u kafiću. *Chatbot* je računalni program ili aplikacija koja korisnicima omogućava komunikaciju i interakciju putem teksta ili govora. *Chatbot* ispravlja korisnika ovisno o pogreškama te daje savjete i dosjetke u slučaju neznanja.

Međutim, neki korisnici primjećuju da se aplikacija previše orijentira na pamćenje pojedinih riječi, a manje na korištenje gramatičkih struktura i razumijevanje konteksta. Također, neki korisnici smatraju da je napredak u učenju ograničen nakon određene razine zato što se većinski fokusira na učenje jezika kroz vokabular što je vidljivo na slici 2.4. gdje u početnom izborniku korisnik bira područje vokabulara koje želi vježbati.

Memrise je pristupačna i interaktivna aplikacija koja nudi mogućnost prilagođavanja lekcija individualnim potrebama korisnika. Aplikacija je pogodna za početnike koji žele brzo usvojiti osnovni vokabular i koristiti ga u svakodnevnim situacijama. Cijena je aplikacije 199.99 dolara za doživotni pristup, odnosno 14.99 dolara mjesečno.



Sl. 2.4. Prikaz sučelja Memrise aplikacije

2.5. Busuu

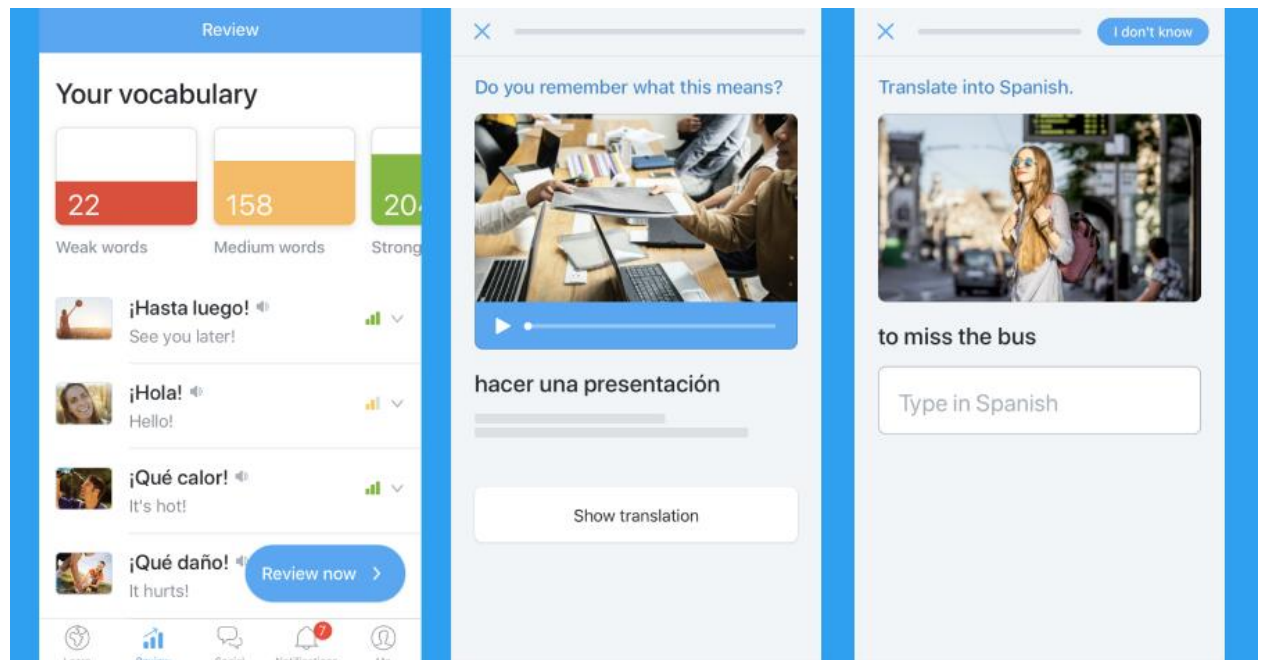
Busuu je aplikacija za učenje stranih jezika koja nudi različite funkcionalnosti. Korisnici mogu pristupiti širokom spektru jezika (14 jezika) i vježbi putem mobilne aplikacije ili mrežne stranice. Aplikacija sadrži interaktivne lekcije, dijaloge, vježbe izgovora (slika 2.5.) i mogućnost

dopisivanja s izvornim govornicima kroz *Busuu* aplikaciju. Cijena je 83.88 eura godišnje, odnosno 14.99 eura mjesečno, što ga svrstava u povoljnije aplikacije. [6]

Jedna je od prednosti *Busuu* fokus na interakciji i komunikaciji. Korisnici mogu dobiti povratne informacije od izvornih govornika jezika koji su dio *Busuu* zajednice. Također, aplikacija nudi mogućnost prilagođavanja učenja individualnim ciljevima i interesima korisnika.

Međutim, besplatna verzija *Busuu* ima ograničen pristup lekcijama, a za pristupanje punom je sadržaju potrebna pretplata.

Prednosti korištenja *Busuu* uključuju mogućnost učenja jezika u interaktivnom okruženju, pristup autentičnom materijalu i korisničkoj podršci. Aplikacija je pogodna za one koji žele razviti jezične vještine koristeći stvarne primjere i komunicirajući s drugim korisnicima. *Busuu* nudi i mogućnost učenja offline, što je pogodno za putovanja i situacije bez internetske veze.



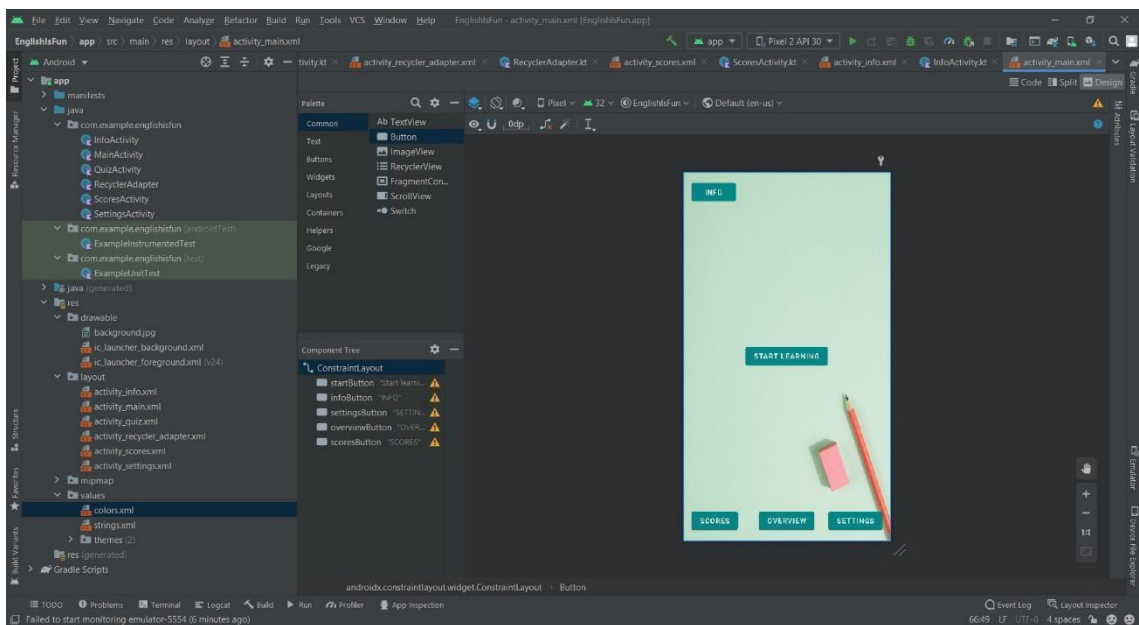
Sl. 2.5. Prikaz sučelja Busuu aplikacije

2.6. Osvrt na postojeća rješenja

U poglavljima 2.1 – 2.5 govori se o najpoznatijim aplikacijama za učenje jezika. Sve aplikacije imaju svoje prednosti i mane pa iz tog razloga svaka ima svoju vjernu bazu korisnika. Većina se aplikacija plaća, što ih čini nedostupnima za velik broj ljudi koji žele učiti jezike, a možda si to ne mogu priuštiti. Sve nabrojene aplikacije nude velik izbor načina učenja, poput igranja igara, kvizova, razgovora s izvornim govornicima pa čak i upotrebe najnovijih tehnologija poput *chatbota*. Većina tih aplikacija fokusira se na učenje vokabulara i govora, dok se gramatika marginalizira, što nije slučaj i s ovom aplikacijom koja sadrži brojne gramatičke lekcije i testove.

3. POSTUPAK RAZVOJA APLIKACIJE

U ovom poglavlju detaljnije je opisan razvoj aplikacije za brzo i interaktivno učenje engleskog jezika kroz okruženje Android Studio. Opisani su i prikazani kroz primjere najznačajniji korišteni programski koncepti. Kroz poglavlje su opisani *RecyclerView*, prenošenje podataka pomoću klase *Intent*, implementacija spremanja rezultata kvizova na vanjski cloud te sama implementacija kviz pitanja u aplikaciju. Na slici 3.1. je prikazan izgled sučelja Android Studio programskog okruženja u procesu kreiranja ove aplikacije gdje se mogu vidjeti sve jedinice koje sačinjavaju prozor koji se prikazuje korisniku.

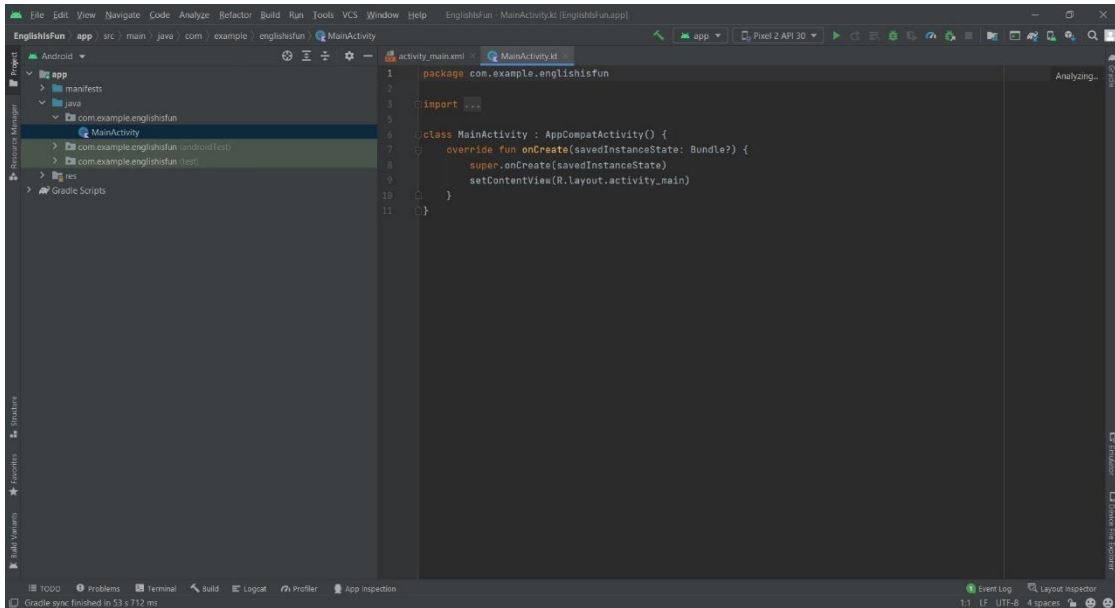


Sl. 3.1. Prikaz svih Activity struktura uz njihove .xml datoteke

3.1. Android Studio i Kotlin

Za potrebe razvojnog okruženja za razvijanje ovog projekta odabran je Android Studio i programski jezik Kotlin. On nudi velik broj značajki koje poboljšavaju produktivnost pri izradi Android aplikacija kao što su:

- brz emulator bogat značajkama koje pomažu pri testiranju aplikacije;
- jednostavno sučelje gdje se mogu razviti sve značajke za aplikaciju vidljivo na slici 3.2.;
- ekstenzivni alati za testiranje aplikacije;
- podrška za C++ i NDK(engl. *The Native Development Kit*) [7].



Sl. 3.2. Prikaz sučelja programskog okruženja Android Studio

Kotlin je moderan programski jezik čiji je cilj učiniti programere bržima i učinkovitijima. Koncizan je, siguran, interoperabilan s Javom i drugim jezicima te pruža mnoge načine za ponovnu upotrebu koda između više platformi za produktivno programiranje.

Prednosti Kotlinu u odnosu na Javu su brojne:

- jednostavnije programiranje;
- kraće vrijeme pisanja programa;
- lakša čitljivost koda;
- besplatno preuzimanje JVM-a(engl. *Java virtual machine*);
- standardno predinstaliran JVM [8].



Sl. 3.3. Prikaz jednostavnosti Kotlinu u odnosu na Javu

Kotlin kao programski jezik je 100% interoperabilan s Javom iz razloga što Kotlin standardna biblioteka ovisi o Javi i kodu pisanom u Javi. Jedina razlika je što je taj kod u Kotlinu koncizniji i lakši za pisanje što je vidljivo na slici 3.3. Kotlin je dizajniran po uzoru na Javu te koristi naprednije i kompleksnije metode za kreiranje Android aplikacija. [9]

3.2. Arhitektura i povezanost slojeva

Android je mobilni operacijski sustav temeljen na Linux jezgri. Njegova se arhitektura sastoji od nekoliko ključnih komponenti. Linux jezgra pruža hardversku apstrakciju, upravljanje memorijom i uređajima. Android Runtime (ART) pretvara Java kod u izvršni kod, dok je Dalvik VM prethodna implementacija ART-a. Sustavne biblioteke pružaju ključne funkcionalnosti kao što su grafika, mreža i sigurnost. Okvir za aplikacije omogućuje razvoj aplikacija, a korisničko sučelje (UI, engl. *User Interface*) koristi se za interakciju s korisnikom. Sustavni procesi upravljaju osnovnim operacijama poput pokretanja aplikacija i upravljanja resursima. Konačno, aplikacije se izvode unutar sandbox okruženja s ograničenim pristupom resursima. Ova arhitektura omogućuje fleksibilnost, sigurnost i široku kompatibilnost, što je doprinijelo uspjehu Androida kao vodećeg mobilnog operacijskog sustava. [10]

3.3. RecyclerView

RecyclerView je UI komponenta koja je nastala kao proširenje za *ListView* komponentu, a koja prikazuje listu sadržaja. *RecyclerView* delegira posao rasporeda elemenata *LayoutManageru*, pa je tako moguće postići pomicanje sadržaja vertikalno i horizontalno. Elementi koji su izašli iz prikaza se ne brišu iz memorije nego ostaju u njoj kako bi ponovno mogli biti prikazani bez kreiranja. *RecyclerView* reciklira (ponovno koristi) poglede koji su trenutno vidljivi na ekranu kako bi se minimizirala potrošnja memorije i optimizirala brzina prikazivanja. To znači da se samo oni dijelovi liste koji su trenutno vidljivi prikazuju u memoriji. Taj način prikaza elemenata značajno smanjuje zahtjeve prema računalnim resursima. *RecyclerView* zahtjeva korištenje *ViewHolder* oblikovnog obrasca koji omogućuje manje poziva *findViewById* metoda nad *View* objektima, što dodatno ubrzava rad. Kod rada s *RecyclerViewom* nužno je napisati vlastitu adapter klasu koja će omogućiti prilagodbu prikaza sadržaja iz objekata modela u *View* objekte. Za razliku od *ListViewa*, *RecyclerView* omogućuje bolju interakciju s elementima iz liste, animaciju pojedinih elemenata,

prilagodbu razmaka između elemenata i slično. Njegovo je postavljanje složenije u odnosu na *ListView*. *ViewHolder* je klasa koja predstavlja jedan element unutar *RecyclerViewa* i upravo se ona koristi prilikom recikliranja elemenata. *RecyclerView* će napraviti onoliko instanci *ViewHoldera* koliko je potrebno da bi se korisnički ekran popunio podacima. Za sve podatke koji nisu trenutno prikazani na ekranu reciklirat će se već instancirani *ViewHolder* s novim sadržajem. Prilikom stvaranja *ViewHolder* objekta potrebno je kroz konstruktor poslati *View objekt*, odnosno već napisani izgled objekta u XML-u (engl. *Extensible Markup Language*). Adapter je klasa koja se brine o stvaranju *ViewHoldera* kao i držanju podataka koji će se prikazivati unutar *RecyclerViewa*. Postoje tri systemske metode koje se moraju upotrijebiti kako bi adapter funkcionirao. Metoda *getItemCount* je metoda koja vraća ukupni broj elemenata koji se nalaze u *RecyclerViewu*. Metoda koja je zadužena za kreiranje i recikliranje *ViewHoldera* naziva se *onCreateViewHolder*, a treća metoda naziva se *onBindViewHolder* i služi za povezivanje *ViewHoldera* i podataka koji se nalaze u adapteru. [11]

U ovoj aplikaciji *RecyclerView* se koristi za prikazivanje svih mogućih poglavlja, odnosno gramatičkih struktura (slika 4.4.) koje korisnik može odabrati kako bi mogao proći teorijski dio te nastaviti na dio s kvizom.

3.4. XML

U kreiranju ove Android mobilne aplikacije u Kotlinu koristi se XML za definiranje korisničkog sučelja (UI) aplikacije. XML datoteke koriste se za opisivanje izgleda ekrana i grafičkih elemenata, kao što su tipke, tekstualna polja i prikazi. U XML-u su definiralni rasporedi svih elemenata, njihovi stilovi i poveznice s Kotlin kodom putem jedinstvenih identifikatora (ID-ova). Tijekom izvođenja aplikacije, Kotlin kod komunicira s XML-om kako bi postavio dinamičke sadržaje, rukovao korisničkim interakcijama i ažurirao sučelje. Korištenjem XML-a omogućeno je jasno odvajanje dizajna od logike, što olakšava održavanje i prilagodbu aplikacije. Na slici 3.4. prikazan je dio XML kod glavnog sučelja vidljivog korisniku. Kroz kod je jasno vidljivo kreiranje gumbova, postavljanje njihovih prostornih ograničenja i boja. Takav kod postoji za svaki grafički element u sučelju te je on zapravo zadužen za sve grafičke elemente vidljive korisniku.

```

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/background"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/startButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/teal_700"
        android:text="Start learning"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.499" />

    <Button
        android:id="@+id/infoButton"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:backgroundTint="@color/teal_700"
        android:text="INFO"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.05"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintVertical_bias="0.023" />

    <Button

```

SI 3.4. XML kod glavnog sučelja

3.5. Cloud Firestore

Cloud Firestore je fleksibilna i skalabilna NoSQL baza podataka koja se nalazi na oblaku računala. Putem nje se spremaju ili sinkroniziraju podaci za potrebe klijentske i poslužiteljske strane aplikacije. Omogućava sinkronizaciju u stvarnom vremenu, što znači da korisnik ima pristup svojim podacima bez obzira ima li pristup internetu ili ne. Kako je Cloud Firestore NoSQL baza podataka podaci se spremaju kao dokumenti s ključ/vrijednost načinom. U ovoj aplikaciji koristi se kao baza za pohranjivanje točnih odgovora za kvizove. [12]

U prvoj liniji koda u kod implementirani su paketi Google Firebase-a kako bi se taj kod mogao koristiti u ovoj aplikaciji. U samoj funkciji, dohvaća se instanca *FirebaseFirestore* objekta, gdje se nakon toga kreira nekoliko testnih objekata klase sa slike 3.7. Nakon kreiranja samog polja objekata pitanja, ta pitanja se kroz for petlju dodaju u kolekciju *Firestore*. Primjer koda prikazan je na slici 3.5.

```

fun saveExampleData() {
    val firestore = FirebaseFirestore.getInstance()
    val questions = ArrayList<Question>(listOf(
        Question(id = 1, questionText = "Which sentence uses the correct subject-verb agreement",
        Question(id = 2, questionText = "Choose the correct sentence with proper punctuation.",
        Question(id = 3, questionText = "Identify the sentence with the correct use of \"its\"")
    ))
    val questionsCollection = firestore.collection("exampleQuestions")

    for (question in questions) {
        questionsCollection.document(question.id)
            .set(question)
            .addOnSuccessListener {
                println("Question ${question.id} successfully added to Firestore")
            }
            .addOnFailureListener { e ->
                println("Error adding question ${question.id} to Firestore: $e")
            }
    }
}

```

SI 3.5. Prikazano učitavanje testnih pitanja na Cloud Firestore

3.6. Radio grupa i gumbi

Radio grupa je kontejner koji grupira radio gumbe zajedno i osigurava da se samo jedan radio gumb može odabrati unutar te grupe. To znači da ako jedan radio gumb postane označen, svi ostali radio gumbi u istoj grupi će automatski postati neoznačeni. Radio gumb je pojedinačni element korisničkog sučelja koji predstavlja jednu opciju između više mogućih opcija. Koristi se za označavanje odabrane opcije.

S radio gumbovima i radio grupama u Kotlinu, može se jednostavno implementirati izbor između različitih opcija unutar aplikacije ili korisničkog sučelja. Konkretni primjer Radio Grupe prikazan je na slici 3.6. gdje je vidljivo kako se u ovoj aplikaciji koristi radio grupa s radio gumbima. Kad korisnik odabere neki radio gumb unutar iste radio grupe, automatski će biti poništeno označavanje svih drugih radio gumbova unutar te iste grupe. Ovo osigurava da se može odabrati samo jedna opcija iz skupa ponuđenih opcija.

Radio grupa u ovom projektu kreira se svaki puta kada se korisniku postavi pitanje, u naslov radio grupe postavlja se varijabla klase *Question*, odnosno *questionText*. U svaki gumb za tekst postavlja se jedan odgovor iz liste odgovora *possibleAnswers*. Na taj način korisnik zapravo dobije

moгуćnost odabrati samo jedan odgovor na jasno vidljivom i shvatljivom sućelju kako ne bi bilo zbunjujuće.

3.7. Objekt Activity

U Kotlin programiranju, aktivnost(engl. *Activity*) se obićno odnosi na komponentu Android aplikacije koja predstavlja jedan ekran ili interakciju s korisnikom. Aktivnosti su osnovna građevna jedinica Android aplikacije i koriste se za prikazivanje korisnićkog sućelja, obradu korisnićkih interakcija i upravljanje životnim ciklusom aplikacije.

Izdvojeno je nekoliko ključnih aspekata aktivnosti u Kotlin programiranju:

1. **Prikaz korisnićkog sućelja:** Aktivnosti se koriste za prikazivanje korisnićkog sućelja na ekranu uređaja. Svaka aktivnost mođe sadržavati različite grafićke elemente kao što su gumbi, tekst, slike i drugi vizualni elementi.
2. **Obrada korisnićkih interakcija:** Aktivnosti obrađuju korisnićke interakcije kao što su dodirivanje ekrana, unos teksta ili pritisak gumba. Programer mođe definirati logiku za reagiranje na te interakcije unutar aktivnosti.
3. **Životni ciklus:** Aktivnosti imaju svoj životni ciklus koji se sastoji od razlićitih stanja kao što su stvaranje, pokretanje, zaustavljanje, uništenje itd. Razumijevanje životnog ciklusa važno je za upravljanje resursima i oćuvanje stanja aplikacije.
4. **Povezivanje između aktivnosti:** Aplikacije često sadrđe više aktivnosti koje su međusobno povezane. Na primjer, jedna aktivnost mođe pozvati drugu kako bi prikazala drugi ekran ili prikazala dodatne informacije.
5. **Integracija s manifest datotekom:** Svaka aktivnost mora biti deklarirana u manifest datoteci Android aplikacije kako bi Android operativni sustav znao kako ih koristiti. Manifest datoteka sadrži informacije o aktivnostima, njihovim namjenama i dozvolama.

Kotlin je popularan jezik za razvoj Android aplikacija jer nudi moderne znaćajke, bolju sigurnost tipova i čistiju sintaksu u usporedbi s Java jezikom. Programiranje aktivnosti u Kotlinu omogućuje razvoj visokokvalitetnih Android aplikacija s manje koda i lakše odrđavanje. Više o aktivnostima u ovoj aplikaciji govori se u poglavlju 4, gdje je detaljno opisana svaka aktivnost i njezina funkcionalnost.

3.8. Objekt Intent

Ako se Android aplikacija sastoji od više aktivnosti, potrebno je navigirati od jedne do druge aktivnosti. Mehanizam koji omogućuje tu navigaciju naziva se *Intent*. Korištenjem odgovarajućeg Intenta moguće je pokrenuti bilo koju aktivnost što je vidljivo na slici 3.6. On je zapravo objekt koji enkapsulira zahtjev, odnosno skup informacija, čiji su osnovni elementi akcija (opća radnja koju je potrebno izvršiti) i podaci nad kojima se radnja izvršava. Razlikujemo implicitne i eksplicitne Intente. Eksplicitni definiraju ciljanu komponentu u samom Intentu, dok implicitni traže od sustava da procijeni komponente temeljeno na podacima ili radnji koju Intent sadrži. Osim promjene Activitya, Intent omogućuju i komunikaciju među drugim komponentama Android aplikacije. Intent je, u slučaju ove aplikacije, korišten za navigiranje između nekoliko različitih „prikaza“ korisniku. Navigiranjem kroz njih korisnik dolazi do različitih funkcionalnosti aplikacije. [13]

```
val intent = Intent(this, SecondActivity::class.java)
startActivity(intent)
```

Sl. 3.6. Primjer korištenja eksplicitnog intenta kakav je korišten u aplikaciji

3.9. Kviz

Glavni je dio ove aplikacije kviz koji korisnik može pokrenuti kada prođe teorijski dio neke gramatičke strukture. Kako bi korisnik otključao iduću gramatičku strukturu, mora odgovoriti točno na barem 50% pitanja koja dobije u kratkom kvizu nakon što prouči teorijski dio strukture koji bude prikazan na zaslonu uređaja prije prikazivanja samog kviza. Kada završi teorijski dio strukture, korisnik klikom na gumb *EVALUATE* poziva *StartQuiz* funkciju koja prikazuje 10 nasumičnih pitanja na koja korisnik daje odgovore. Primjer jednog pitanja prikazan je na slici 4.4.

U aktivnosti se nakon pokretanja kviza, ovisno o strukturi koja je odabrana, učitavaju pitanja iz datoteka koje su napravljene pri pokretanju aplikacije te izmiješana kako bi korisnik svaki put dobio drukčiji set pitanja.

Uz pomoć klase *Question* definirana su polja za pitanje uz polje za točan odgovor i lista netočnih odgovora. Klasa *Question* i njezina roditeljska klasa *GrammarStructure* prikazane su na slikama 3.7. i 3.8.

```

data class Question (
    val id: Int,
    val questionText: String,
    val possibleAnswers: ArrayList<String>,
    val correctAnswerIndex: Int
)

```

Sl. 3.7. Klasa *Question* koja pohranjuje svako pojedino pitanje uz njegov ID(jedinstveni broj koji se daje svakom pitanju), tekst pitanja, broj indeksa točnog odgovora iz liste odgovora i listu mogućih odgovora

Pritiskom bilo kojeg odgovora prvo se provjerava točnost odgovora koja se korisniku javlja putem *Toast* poruke koja se ispisuje na zaslonu. Toast poruka u programskom jeziku Kotlin (kao i u Android razvoju općenito) je korisnički sučeljni element koji se koristi za prikazivanje kratkih obavijesti ili poruka korisnicima mobilnih aplikacija. Ove poruke obično se pojavljuju kao male prozorići s tekстом ili ikonom na dnu ili vrhu zaslona i automatski nestaju nakon određenog vremena ili nakon što ih korisnik zatvori. Toast poruke koriste se za informiranje korisnika o kratkotrajnim događajima ili radnjama, poput potvrde uspješno izvršene akcije, prikaza trenutnih podataka (npr. temperatura ili status mreže), ili obavijesti o greškama. Ako je odgovor točan, dodaje se jedan bod na varijablu brojača koja se na kraju podijeli s ukupnim brojem pitanja kako bi korisnik dobio povratnu informaciju o uspjehu. MainActivity je glavna aktivnost te je ona zapravo prvo što korisnik može vidjeti na svom zaslonu kao što je prikazano na slici 4.1. Nakon uspješno odrađenog kviza korisnik se vraća na početni zaslon, odnosno na glavnu aktivnost. U aplikaciju je dodano 10 gramatičkih struktura a svaka se sastoji od 15 pitanja, no ako bi se aplikacija razvijala dalje, s obzirom da je korišten RecyclerView, može se vrlo lako dodati nova gramatička struktura ili povećati izbor pitanja za pojedinu gramatičku strukturu. Pitanja se postavljaju tako da se nasumično odabere jedno pitanje čiji je tekst prikazan u vrhu ekrana, dok su ispod pitanja prikazani odgovori.

```

data class GrammarStructure (
    val name: String,
    val difficulty: Int,
    val questionArray: ArrayList<Question>
)

```

Sl.3.8. Klasa *GrammarStructure* koja sprema ime lekcije, težinu lekcije u obliku broja te listu pitanja

4. PRIKAZ FUNKCIONALNOSTI APLIKACIJE

U ovom poglavlju prikazan je izgled zaslona pojedinih aktivnosti na uređaju i opisane su njihove funkcije u pojedinim aktivnostima. Na slici 4.1. vidljiva je glavna aktivnost (prva koja je prikazana korisniku) te je u poglavlju 4.1. opisana njezina funkcionalnost, odnosno funkcionalnost svih gumbova prikazanih na njoj. Prikazani su i kod i izgled info gumba na slikama 4.2. i 4.3.

4.1. Glavna aktivnost

U glavnoj aktivnosti nalazi se 5 gumba od kojih svaki otvara novu aktivnost.

- Gumb *INFO* – prebacuje zaslon s MainActivity na InfoActivity;
- Gumb *START LEARNING* – prebacuje zaslon s MainActivity na QuizActivity;
- Gumb *SCORES* – prebacuje zaslon s MainActivity na ScoresActivity;
- Gumb *OVERVIEW* – prebacuje zaslon s MainActivity na ScoresActivity;
- Gumb *SETTINGS* – prebacuje zaslon s MainActivity na SettingsActivity.

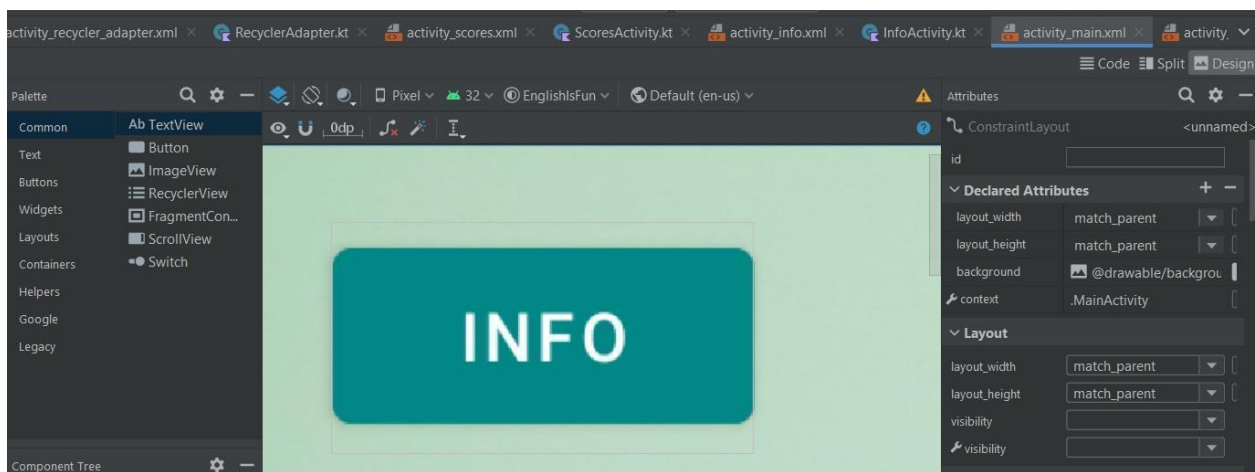


Sl.4.1. Prikazana glavna aktivnost

Na slici 4.2. moguće je vidjeti sam kod gumba, te je iz koda moguće vidjeti njegov id, njegova ograničenja i poziciju, njegovu boju te tekst koji prikazuje. Na slici 4.3. prikazan je sam gumb, odnosno ono što XML kod na slici 4.2. predstavlja korisniku.

```
<Button
    android:id="@+id/infoButton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:backgroundTint="@color/teal_700"
    android:text="INFO"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.05"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.023" />
```

Sl. 4.2. XML kod gumba prikazanog na slici 4.3.



Sl. 4.3. Prikaz info gumba u programskom sučelju Android Studio

4.2. Kviz aktivnost

Kao što je prikazano na slici Sl. 4.4. i objašnjeno u poglavlju 3.9., kviz aktivnost izbacuje 10 nasumičnih pitanja jedno po jedno na ekran. Na toj aktivnosti također postoji jedan gumb koji služi za prekidanje trenutnog kviza i vraćanje na početnu aktivnost, odnosno u glavni izbornik. Na slici 4.4. prikazano je sučelje vidljivo korisniku u trenutku postupka odgovaranja na pitanja. Taj proces

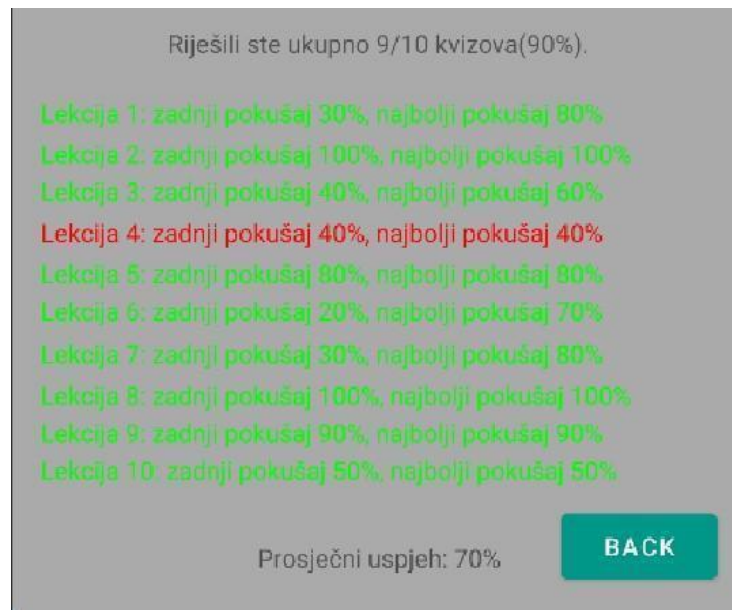
je bolje objašnjen u poglavlju 3.9. gdje je objašnjen svaki element ove slike te kako je on kreiran u pozadini koju korisnik ne vidi.

A screenshot of a quiz question displayed on a grey background. The question is "_____ old is Mike?". Below the question are four radio button options: "What", "How", "When", and "If". In the bottom right corner, there is a teal button labeled "BACK".

Sl. 4.4. Prikaz jednostavnog kviz pitanja uz korištenje Radio grupe s 4 Radio gumba

4.3. Scores aktivnost

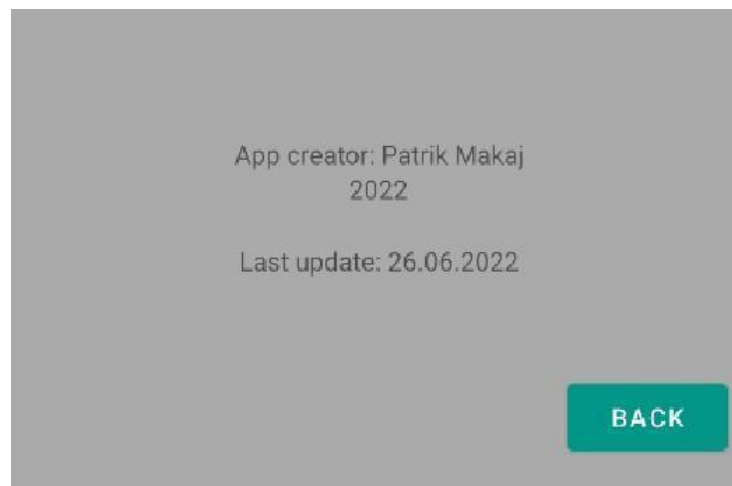
U *Scores* aktivnosti oprimjerenoj na slici 4.5. prikazani su svi dosadašnji pokušaji rješavanja kvizova uz najveći ostvareni rezultat po kvizu. Također, prikazana je sveukupna riješenost kvizova uz prosječan rezultat. Na toj aktivnosti također postoji jedan gumb koji služi za prekidanje trenutnog kviza i vraćanje na početnu aktivnost, odnosno u glavni izbornik.



Sl. 4.5. Prikazana Scores aktivnost

4.4. Info aktivnost

U *Info* aktivnosti prikazane su informacije o aplikaciji, kreatoru aplikacije te zadnjem ažuriranju aplikacije. Aktivnost služi za prikazivanje informacija o kreatoru aplikacije i zadnjem ažuriranju. Primjer aktivnosti je prikazan na slici 4.6. gdje je vidljivo tko je kreator aplikacije te kad je aplikacija bila zadnji put ažurirana.



Sl.4.6. Prikazano sučelje nakon pritiska info gumba

4.5. Usporedba aplikacije s rješenjima navedenim u poglavlju

Cilj ove aplikacije nije isti kao i cilj ostalih aplikacija na tržištu - naučiti što više jezika u što kraćem vremenskom roku kao na primjer 23 jezika i aplikaciji Memrise, ili 14 jezika u aplikaciji Busuu. Cilj je usavršiti engleski jezik na interaktivan, zanimljiv i jednostavan način. Fokusiranje na učenje jednog jezika umjesto više jezika može imati brojne prednosti:

1. Dublje razumijevanje: Fokusiranjem na jedan jezik, možete se bolje udubiti u njegove gramatičke strukture, vokabular i kulturni kontekst, što omogućuje dublje razumijevanje jezika.
2. Brži napredak: Koncentrirajući se na samo jedan jezik, možete brže napredovati jer ćete imati više vremena za učenje i praksu.
3. Veća kompetencija: Usredotočivanje na jedan jezik omogućava vam da razvijete veću kompetenciju u tom jeziku, što može rezultirati boljim govorom, pisanjem i razumijevanjem.
4. Smanjena konfuzija: Učenje više jezika istovremeno može dovesti do konfuzije između gramatičkih pravila, vokabulara i izgovora. Fokusom na jedan jezik, možete izbjeći takvu konfuziju.
5. Lakše zapošljavanje: Ako postanete stručnjak za određeni jezik, možete imati veće šanse za zapošljavanje ili suradnju u specifičnim industrijama kao prevoditelj ili jezični stručnjak.

Još jedna važna razlika između ove aplikacije i onih na tržištu je ta da se ova aplikacija ne plaća. Sve aplikacije se u prosjeku plaćaju više od 10 eura mjesečno što je stvarno visok iznos za nekog studenta, učenika ili umirovljenika. Ova aplikacija omogućava pristup učenju jezika ljudima širom svijeta, bez obzira na njihovu financijsku situaciju. Ova aplikacija je u potpunosti besplatna, dok se Babbel plaća čak 599.99 eura za pristup. To pomaže u smanjenju barijera za obrazovanje. Besplatne aplikacije promiču demokratizaciju znanja i obrazovanja, čime omogućavaju širokom spektru ljudi da steknu jezične vještine, bez obzira na njihovo geografsko podrijetlo ili društveni status. Osim toga, ljudi su skloniji isprobati nešto novo i nastaviti učiti kada ne moraju plaćati. Besplatne aplikacije mogu potaknuti veću motivaciju za učenje jezika. Također, lakše je privući raznoliku skupinu korisnika, što može stvoriti bogatiju zajednicu učenja i omogućiti razmjenu iskustava iz različitih kultura. Aplikacije koje se ne plaćaju često privlače veći broj korisnika, što može rezultirati većim dosegom i utjecajem, posebno među mlađom populacijom koja je često ciljna skupina za učenje jezika.

5. ZAKLJUČAK

Porastom korištenja mobilnih uređaja poput mobitela i tableta, mobilne aplikacije postale su neizostavan dio svakodnevice. Ljudi na svojim mobilnim uređajima imaju vrlo velik broj aplikacija koje pomažu i olakšavaju izvršavanje svakodnevnih zadataka. Jedan od najvrjednijih alata koje pružaju aplikacije su za učenje jezika. Prema izvješću Statista.com, u 2022. godini 10 aplikacija s najvećim brojem preuzimanja za učenje stranih jezika bile su preuzete sveukupno 23 336 525 puta. [14] Širenjem tehnologije i rastom prometne povezanosti – jezici postaju potreba, a aplikacije za učenje rješenje.

Aplikacija koja se fokusira na učenje jednog jezika detaljnija je i funkcionalnija od aplikacija koje sadrže više mogućnosti iz razloga što je moguće bazu podataka napuniti materijalima specifičnima samo za taj jezik, što uvelike pomaže za detaljnije i specifičnije učenje određenog jezika. Osim toga, aplikacija je u potpunosti besplatna za razliku od sličnih aplikacija kod kojih je za gotovo svaku napredniju lekciju potrebno plaćati mjesečno članstvo ili kupiti aplikaciju. U ovoj aplikaciji moguće je detaljno naučiti gramatiku i vokabular zbog detaljno selektirane baze podataka.

Nakon analize već postojećih rješenja, izdvojene su brojne mane postojećih aplikacija za učenje. Iako su te aplikacije bile inspiracija za ovu, izbačeni su neki od najvećih nedostataka. Ovaj je rad napravljen s ciljem učenja gramatičkih struktura engleskog jezika. Aplikacija je razvijena u Android Studio okruženju, čija je razvojna okolina jednostavna za shvaćanje i rad. Aplikacija je vrlo korisna za učenike i studente koji nemaju puno vremena za učenje, ali već imaju nekakvo predznanje pa im je dovoljno samo ponoviti tvorbe gramatičkih struktura uz nekoliko zadataka kako bi uspješno savladali taj dio gradiva. U aplikaciji nema početnog testiranja, već svaki korisnik kreće učiti od početka. Na taj način svi korisnici imaju dovoljne temelje za učenje kompleksnijih poglavlja. Svi pokušaji rješavanja prikazuju se na način da se prikaže postotak riješenosti, a cilj je da su svi kvizovi riješeni sa stopostotnom točnošću. Sučelje je aplikacije vrlo jednostavno, intuitivno i razumljivo svim korisnicima. Aplikacija se može nadograditi dodatnim gramatičkim strukturama i novim kvizovima, a element koji namjerno nedostaje ovoj aplikaciji jest višejezičnost jer je primarni fokus ove aplikacije na učenju i usavršavanju engleskog jezika.

LITERATURA

- [1] Službena stranica statcounter, dostupno na:
<https://gs.statcounter.com/os-market-share/mobile/worldwide> (lipanj 2023.)
- [2] Blog aplikacije Duolingo, dostupno na:
<https://blog.duolingo.com/educator-and-parent-guide-making-the-most-out-of-staying-at-home/> (kolovoz 2023.)
- [3] Blog aplikacije Rosetta Stone, dostupno na:
<https://blog.rosettastone.com/rosetta-stone-unveils-new-features/> (kolovoz 2023.)
- [4] Službena stranica screenlane, dostupno na:
<https://screenlane.com/screens/product/babbel> (kolovoz 2023.)
- [5] Službena stranica Sandy Higuero, dostupno na:
<https://www.sendyhiguero.com/memrise> (rujan 2023.)
- [6] Blog aplikacije Busuu, dostupno na:
<https://blog.busuu.com/busuu-launches-ai-powered-language-learning/> (rujan 2023.)
- [7] Službena stranica programskog jezika Kotlin, kotlinlang.org, dostupno na:
<https://kotlinlang.org/> (lipanj 2022.)
- [8] Razvoj mobilnih aplikacija-priručnik za edukaciju Osijek: Elektrotehnički fakultet Osijek, 2013.
- [9] Phillips, Bill; Stewart, Chris; Hardy, Brian; Marsicano, Kristin Android Programming: The Big Nerd Ranch Guide (2nd Edition) Atlanta: Big Nerd Ranch, LLC., 2015.
- [10] Blog Android Pit, dostupno na
<http://android-pit.blogspot.com/2012/01/arhitektura-androida.html> (lipanj 2023.)
- [11] Sebesta, R.W. Programming the World Wide Web (2nd Ed.) Addison-Wesley, Boston, MA, 2004.
- [12] Službena stranica Firebasea, dostupno na:
<https://firebase.google.com/> (lipanj 2022.)
- [13] Službena stranica Androida, android.com, dostupno na:
<https://www.android.com/what-is-android/> (lipanj 2022.)

[14] Službena stranica Statista, dostupno na:

<https://www.statista.com/statistics/1239522/top-language-learning-apps-downloads>

(lipanj 2023.)

[15] Službena stranica Androida, developer.android.com, dostupno na:

<https://developer.android.com/studio/intro> (lipanj 2022.)

[16] Službena stranica Java, dostupno na:

<https://www.java.com> (lipanj 2022.)

[17] Predložak s nastave

OSNOVE RAZVOJA WEB I MOBILNIH APLIKACIJA, LV 7: Aktivnosti i fragmenti

(lipanj 2022.)

SAŽETAK

Za potrebe je ovog završnog radu razvijena Android aplikacija EnglishIsFun namijenjena za učenje gramatičkih struktura engleskog jezika. Korisnici na zaslonu dobivaju informacije o tvorbi i korištenju gramatičkih struktura nakon čega imaju mogućnost samoprovjere uz kviz od 10 pitanja. Aplikacija je izrađena u okruženju Android Studio u programskom jeziku Kotlin i bazi podatka Cloud Firestore. U radu je objašnjena teorijska pozadina svih korištenih koncepata. Također, u posljednjem je poglavlju objašnjen izgled i funkcionalnost aplikacije. Postoji nekoliko mogućih nadogradnji za aplikaciju, npr. zanimljivije bodovanje i nagrađivanje ili čak prebacivanje cijele aplikacije na druge platforme (iOS ili web) kroz Kotlin Multiplatform tehnologiju.

Ključne riječi: Android aplikacija, gramatika, kviz, učenje engleskog jezika

ABSTRACT

Mobile application for self-learning selected grammatical units

For the purposes of this final thesis, an Android application entitled "EnglishIsFun" has been developed for learning grammatical structures of the English language. Users receive information on the formation and usage of grammatical structures on the screen, followed by the opportunity for self-assessment through a 10-question quiz. The application was created within the Android Studio environment using the Kotlin programming language and the Cloud Firestore database. The theoretical background of all the concepts used is explained in this paper. Additionally, the appearance and functionality of the application are explained in the fourth chapter. There are several potential enhancements for the application, such as more engaging scoring and reward systems, or even transitioning the entire application to other platforms (iOS or web) through Kotlin Multiplatform technology.

Keywords: Android application, English language learning, grammar, quiz

ŽIVOTOPIS

Patrik Makaj rođen je 9. veljače 2001. godine u Našicama. Pohađao je Osnovnu školu Ivane Brlić Mažuranić Koška. Odličnim završenim uspjehom osnovnoškolskog obrazovanja upisuje Opću gimnaziju u Našicama. Srednju školu također završava odličnim uspjehom uz odličan uspjeh na maturi. Nakon završetka srednje škole upisuje preddiplomski sveučilišni studij, smjer računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija, Osijek. Ima izvrsno znanje engleskog jezika te razumijevanje njemačkog jezika. Koristi C, C++, C#, Javu, Kotlin, SQL, MySQL, HTML, CSS i XML.