

Mobilna aplikacija za samoučenje gramatičkih konstrukcija engleskog jezika temeljena na predznanju korisnika

Kazalicki, Krunoslav

Undergraduate thesis / Završni rad

2024

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://urn.nsk.hr/urn:nbn:hr:200:014896>

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

Download date / Datum preuzimanja: **2025-01-22**

Repository / Repozitorij:

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



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

Sveučilišni studij

**MOBILNA APLIKACIJA ZA SAMOUČENJE
GRAMATIČKIH KONSTRUKCIJA ENGLESKOG
JEZIKA TEMELJENA NA PREDZNANJU KORISNIKA**

Završni rad

Krunoslav Kazalicki

Osijek, 2022.

**FERIT**FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA
I INFORMATIJSKIH TEHNOLOGIJA **OSIJEK****Obrazac Z1P - Obrazac za ocjenu završnog rada na preddiplomskom sveučilišnom studiju**

Osijek, 14.09.2022.

Odboru za završne i diplomske ispite

Prijedlog ocjene završnog rada na preddiplomskom sveučilišnom studiju

Ime i prezime Pristupnika:	Krunoslav Kazalicki
Studij, smjer:	Preddiplomski sveučilišni studij Računarstvo
Mat. br. Pristupnika, godina upisa:	R4072, 28.07.2017.
OIB Pristupnika:	56048441707
Mentor:	Doc. dr. sc. Dragana Božić Lenard
Sumentor:	Izv. prof. dr. sc. Mirko Köhler
Sumentor iz tvrtke:	
Naslov završnog rada:	Mobilna aplikacija za samoučenje gramatičkih konstrukcija engleskog jezika temeljena na predznanju korisnika
Znanstvena grana rada:	Programsko inženjerstvo (zn. polje računarstvo)
Zadatak završnog rad:	Cilj je završnoga rada izraditi mobilnu aplikaciju za samoučenje gramatičkih konstrukcija engleskoga jezika. Po pokretanju aplikacije, korisnik će moći riješiti test za procjenu razine znanja na temelju čijega će mu se rezultata odrediti trenutna razina znanja i predložiti gramatičke cjeline koje treba proučiti i usvojiti. Gramatičke će cjeline obuhvaćati teorijske postavke, tvorbu, uporabu, primjere te usporedbe sa srodnim i različitim gramatičkim
Prijedlog ocjene završnog rada:	Izvrstan (5)
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: 3 bod/boda Jasnoća pismenog izražavanja: 3 bod/boda Razina samostalnosti: 3 razina
Datum prijedloga ocjene od strane mentora:	14.09.2022.
Datum potvrde ocjene od strane Odbora:	
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, 16.09.2022.

Ime i prezime studenta:

Krunoslav Kazalicki

Studij:

Preddiplomski sveučilišni studij Računarstvo

Mat. br. studenta, godina upisa:

R4072, 28.07.2017.

Turnitin podudaranje [%]:

4

Ovom izjavom izjavljujem da je rad pod nazivom: **Mobilna aplikacija za samoučenje gramatičkih konstrukcija engleskog jezika temeljena na predznanju korisnika**

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

i sumentora Izv. prof. dr. sc. Mirko Köhler

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	1
1.1. Zadatak završnog rada.....	1
2. ISTRAŽIVANJE POSTOJEĆIH RJEŠENJA	3
2.1. Pregled postojećih rješenja	3
2.1.1. Learn English Grammar.....	3
2.1.2. English Grammar Test	4
2.1.3. English Grammar	4
2.1.4. English Grammar in Use.....	5
2.2. Osvrt na postojeća rješenja.....	5
3. KORIŠTENE RAZVOJNE TEHNOLOGIJE.....	7
3.1. Android.....	7
3.1.1. Arhitektura Androida	8
3.2. Android Studio.....	10
3.3. Kotlin.....	11
3.4. XML	11
3.5. Firebase.....	11
4. IZRADA APLIKACIJE I PRIKAZ FUNKCIONALNOSTI.....	13
4.1. Navigacija aplikacije.....	13
4.2. Dijagram i opis toka aplikacije	13
4.3. Glavni dijelovi aplikacije i funkcionalnosti	15
4.3.1. Home.....	16
4.3.2. Test	17
4.3.3. TestResults.....	19
4.3.4. Learn	20
4.3.5. Lesson	22
4.3.6. Profile	23
4.4. Cloud Firebase baze podataka.....	24
5. TESTIRANJE RADA APLIKACIJE I PRIKAZ KORIŠTENJA ZNAČAJKI.....	25
5.1. Rješavanje inicijalnog testa znanja	25

5.2. Proučavanje gramatičkih struktura.....	27
5.3. Ponovno testiranje i prelazak u novu razinu znanja	28
5.4. Pregled dosadašnjih rezultata testiranja	31
6. ZAKLJUČAK.....	32
LITERATURA	34
SAŽETAK.....	35
ABSTRACT	36

1. UVOD

Pametni su se telefoni još do nedavno smatrali luksuzom, a danas su sastavnim dijelom suvremenog života zbog brojnih mogućnosti koje nam pružaju na dlanu. Gotovo svi ih imamo, a osim kao osobne telefone koje rabimo za komunikaciju, pretraživanje informacija, zabavu i edukaciju, koriste se i u poslovnom kontekstu. Jedna od važnijih značajki pametnog telefona njegova je povezanost s trgovinom aplikacija koja predstavlja centralizirani portal gdje korisnici mogu pretraživati i preuzimati aplikacije za svoje uređaje. Upravo ta trgovina aplikacija omogućava stvaranje i lansiranje vlastitih rješenja u obliku aplikacijskog programa koji u potpunosti te na kreativan i jedinstven način iskorištava sve mogućnosti pametnog telefona.

Engleski je jezik trenutno najrašireniji jezik svijeta čiji veliki postotak čine govornici kojima je engleski strani jezik. To je univerzalni jezik današnjice; standardni jezik u međunarodnom poslovanju, tehnologiji, znanosti, turizmu i mnogim drugim područjima. Poznavanje engleskog jezika otvara brojne mogućnosti poput pristupanja i razumijevanja velikog broja informacija, lakšeg zaposlenja u mnogim međunarodnim tvrtkama, sporazumijevanja s ljudima različitih kulturoloških i lingvističkih podloga kao i mnoge druge. [1]

Upravo se iz ovih dvaju razloga rodila ideja o aplikaciji koja bi pomogla korisniku otvoriti čitav svijet mogućnosti poznavanja engleskog jezika. Praktičnost i jednostavnost mobilnih uređaja omogućuju lako i jednostavno učenje gramatičkih konstrukcija bilo kada i bilo gdje. Upravo ta fleksibilnost i neograničenost vremenom i prostorom pružaju korisniku komfornu zonu učenja u kojoj on sam odabire tempo usvajanja jezika te mogućnost procjene trenutne razine poznavanja.

1.1. Zadatak završnog rada

U okviru ovog završnog rada izradit će se mobilna aplikacija za operacijski sustav Android koja će omogućiti korisnicima samoučenje određenih gramatičkih struktura engleskog jezika kao i evaluaciju trenutnog znanja. Na početku korištenja aplikacije radi se inicijalni test kojim se korisniku dodjeljuje trenutna razina znanja prema kojoj mu se nude točno određene gramatičke cjeline koje bi trebao proučiti kako bi unaprijedio svoje znanje. Te gramatičke cjeline sadrže detaljan opis pravila, slučajeve upotrebe i primjere. Nakon što korisnik prouči predložene mu lekcije, može se ponovo testirati i o tome rezultatu ovisi je li korisnik spreman prijeći na novu razinu, a samim time i zakoračiti u proučavanje naprednijih struktura jezika.

Aplikacija će se napraviti u okruženju Android Studio koristeći programski jezik Kotlin i jezik za označavanje XML. Za izradu će se aplikacije koristiti *Firebase*-ova usluga *Firestore*, koja omogućava stvaranje baza podataka gdje se smještaju podaci koji se prikazuju u aplikaciji i u koje se spremaju podaci temeljeni na određenim izborima korisnika.

2. ISTRAŽIVANJE POSTOJEĆIH RJEŠENJA

Na tržištu aplikacija za učenje engleskog jezika prevladavaju aplikacije za učenje vokabulara kako zbog većeg interesa korisnika za takvim aplikacijama tako i zbog veće motiviranosti korisnika za takvim učenjem koje može biti vizualnije i samim time primamljivije korisnicima. Postoje i aplikacije koje se ne orijentiraju ni na jedno područje jezika, već nude učenje jezika u cjelini, no kako se izrađena aplikacija temelji na gramatici, u ovom se poglavlju daje pregled postojećih rješenja u vidu aplikacija za učenje engleskog jezika čiji je cilj učenje gramatike. Analiziraju se značajke, spektar cjelina gramatike koje se nude te načini provedbe samoučenja i evaluacije. Razmatraju se sličnosti i razlike među postojećim rješenjima te u konačnici i između aplikacije zadatka ovoga rada.

2.1. Pregled postojećih rješenja

Neke od pronađenih postojećih rješenja koja će se analizirati su aplikacije *Learn English Grammar*, *English Grammar Test*, *English Grammar* i *English Grammar in Use*. Sve su aplikacije dostupne za operacijski sustav Android, na kojem su i testirane. Također su sve aplikacije besplatne za preuzimanje, no svaka aplikacija uz plaćanje nudi dodatne značajke i eventualno proširenje sadržaja.

2.1.1. Learn English Grammar

Aplikacija *Learn English Grammar*¹ koncipirana je na način da korisnik sam izabire područje gramatike koje želi proučavati, a svako se područje dijeli na određeni broj gramatičkih cjelina koje korisnik također proizvoljno odabire. Ulaskom u gramatičku cjelinu, korisniku se prikazuje sažeta definicija određene strukture uz primjere te pravilo, također uz primjere. Nakon toga slijedi pitanje koje provjerava znanje prethodnog pravila čime se korisniku daje povratna informacija je li njegov odgovor točan. Zatim se takav niz ponavlja s drugim strukturama, čiji broj ovisi o danoj cjelini i području.

¹ https://play.google.com/store/apps/details?id=com.english.vivoapp.grammar.grammaren&hl=en_US&gl=US

Tako opisana aplikacija može biti izuzetno korisna za proučavanje specifičnih gramatičkih lekcija zbog široke pokrivenosti gramatike i opsežnog broja pravila i primjera svake lekcije. Također, u aplikaciji se gramatika dijeli na manje dijelove zbog čega je jednostavna za navigaciju i korištenje. Sadržaj je aplikacije u velikoj mjeri besplatan, no aplikacija nudi dodatne značajke (poput izgovora) i mogućnost uklanjanja oglasa za određenu naknadu.

2.1.2. English Grammar Test

*English Grammar Test*² aplikacija je koja ne pruža izravan način i resurse za učenje, već to radi kroz testove znanja. Na ulasku se korisniku nude tri razine težine gramatičkih lekcija. Ulaskom u svaku od razina nudi se popis tema čiji se testovi mogu rješavati, a svaka lekcija nudi tri testa. Postoji i mogućnost odabira 30 testova u kojima su teme lekcija izmiješane. Kvizovi su takvi da se korisniku prikaže rečenica u kojoj je potrebno nadopuniti jednu riječ. Korisniku su ponuđena četiri odgovora, a klikom na jedan od njih prikazuje se točan odgovor i objašnjenje. Na taj je način zamišljeno učenje na principu pokušaja i pogrešaka.

Sama aplikacija i njezine osnovne značajke besplatne su, no postoje određene plaćene funkcije. Dodatna je značajka mogućost dopisivanja s ostalim korisnicima aplikacije u vidu soba za razgovor, a cilj je značajke da korisnici jedni drugima pomažu u shvaćanju određenih tema.

2.1.3. English Grammar

Analizom aplikacije *English Grammar*³ vidljivo je da nudi prilično širok izbor gramatičkih cjelina koje se sastoje od različitog broja lekcija. Lekcije su opsežne i detaljno objašnjene. Svaka cjelina nudi desetak testova znanja koji zahtijevaju odgovore na sva pitanja prije prikazivanja njihovih točnosti. Nakon završetka takvih kvizova korisniku nije ponuđeno objašnjenje za netočno odgovorena pitanja, već se od korisnika očekuje samostalno istraživanje pogreški. Također, nudi se mogućnost dnevnog praćenja napretka u vidu postotka točnih odgovora na navedenim testovima. Aplikacija ne nudi inicijalni test za procjenu znanja korisnika, kao ni sličan sveobuhvatni test koji bi korisniku pružio uvid u njegovu razinu poznavanja jezika.

² <https://play.google.com/store/apps/details?id=english.grammar.test.app>

³ <https://play.google.com/store/apps/details?id=com.kimco.english.grammar.in.use.test.ultimate.englishgrammar>

2.1.4. English Grammar in Use

Aplikacija poznatog britanskog sveučilišta Cambridgea *English Grammar in Use*⁴ podijeljena je na nekoliko cjelina od kojih se svaka sastoji od određenog broja jedinica (lekcija). Cjeline su osnovne, a samo je prva cjelina besplatna, dok se ostale plaćaju pojedinačno ili u paketu. Svaka je lekcija podijeljena na sekcije koje predstavljaju određeno pravilo. U svakoj je sekciji objašnjeno pravilo i dani su primjeri za isto. Lekcija također sadrži dio za procjenu usvojenosti lekcije koji se sastoji od svega nekoliko pitanja višestrukog izbora. Nakon odgovaranja na svako pitanje potrebno je kliknuti na gumb za provjeru točnosti danog odgovora. U slučaju netočno odgovorenog pitanja aplikacija ne nudi nikakvo objašnjenje, već šansu za novi odgovor. Korisnik je prepušten sam sebi kako bi unutar objašnjenja lekcije pronašao zašto njegov odgovor nije točan. U aplikaciji nije implementirana značajka za praćenja napretka korisnika niti značajka za procjenu znanja određene cjeline ili lekcije. Zbog nedostatka takvih funkcionalnosti, aplikacija je korisna za pretraživanje određenih pravila, ali nije pogodna za cjelovito samoučenje engleskog jezika.

2.2. Osvrt na postojeća rješenja

Analizom navedenih aplikacija može se zaključiti da je sadržaj uglavnom podijeljen na veće cjeline koje se dijele na manje jedinice definirane i prikazane primjerom. Sve aplikacije nude neki način procjene poznavanja određenog područja pomoću kvizova višestrukog izbora. Razlog tomu je što je takav način ispitivanja kratak i precizan, no također i interaktivan jer korisnik odmah zna je li odgovorio točno. Kada korisnik ugleda točan odgovor, to može biti okidač pamćenju da točno odgovori i u slučaju da povratna informacija potvrđuje korisniku da je njegov odgovor točan, to ojačava poveznicu između pravilne uporabe određenog pravila odnosno strukture. [2]

Sve aplikacije, osim *English Grammar Test*, koja ima specifičan način podučavanja novih pravila, prikazuju lekcije na sličan, jednostavan i izravan način. Razlikuju se po količini i opširnosti obuhvaćenih područja gramatike gdje neke aplikacije nude poprilično širok izbor lekcija, dok se druge temelje na osnovnim gramatičkim područjima koje, u tom slučaju, detaljnije obrađuju. Većina njih sadržaj nudi besplatno, ali uz neizbježne reklame. Međutim, postoji

⁴ https://play.google.com/store/apps/details?id=org.cambridge.englishgrammar.egiu&hl=en_US&gl=US

moćnost plaćanja za omogućavanje dodatnih značajki, kupnju novih paketa lekcija ili uklanjanje reklama.

Analizirane aplikacije imaju dobro definiran i objašnjen sadržaj, no njihov je zajednički nedostatak taj što ne usmjeravaju korisnika na onaj sadržaj koji mu je primjeren u datom trenutku. Sve se navedene aplikacije oslanjaju na to da korisnik točno zna koje područje i što iz njega želi proučavati. To nije nužno loša izvedba jer tako koncipirane aplikacije mogu služiti kao koristan i pouzdan izvor za referenciranje, podsjećanje i provjeru pravila. Međutim, početnike, ali i korisnike koji napreduju, potrebno je usmjeriti na sadržaj koji bi trebali proučavati kako bi poboljšali svoje trenutno znanje i uvrstili nove strukture u svoje izražavanje. Također, testovi za procjenu znanja koje navedene aplikacije izvode bazični su i ne daju korisniku jasnu sliku o njegovom ukupnom poznavanju jezika na području gramatike.

Uzimajući u obzir navedeno, izrađena će aplikacija implementirati one značajke analiziranih aplikacija za koje je utvrđeno da su korisne i učinkovite. Neke su od njih dobro definirana podjela sadržaja na cjeline i lekcije, detaljno objašnjena pravila i primjene istih, dobro ilustrirani primjeri te kvizovi višestrukog izbora za koje je utvrđeno da su dobra metoda učenja novih koncepata jezika. Osim toga, izrađena aplikacija rješava neke od navedenih nedostataka tako da ne prepušta korisnika samog sebi, nego ga usmjerava na lekcije koje će unaprijediti njegovo trenutno znanje. Također, korisnik ne mora brinuti o samoprocjeni trenutnog poznavanja jezika jer mu aplikacija nudi standardizirani test na temelju kojeg mu se određuje razina znanja.

3. KORIŠTENE RAZVOJNE TEHNOLOGIJE

3.1. Android

Android je operacijski sustav razvijen na modificiranoj verziji Linuxove jezgre. Inicijalno ga je razvila softverska tvrtka Android Inc. koju je 2005. godine kupila tvrtka Google nakon čega ga nastavlja razvijati zajedno s konzorcijem od 84 tvrtke zvanim *Open Handset Alliance*. Iako je operacijski sustav prvenstveno bio namijenjen mobilnim telefonima sa zaslonom osjetljivim na dodir, njegovom popularizacijom uporaba mu se širi na tablete, televizore, pametne satove, medijske uređaje automobila i mnoge druge „pametne“ uređaje. [3]

Otvorenost koda operacijskog sustava omogućilo je da ga bilo koji proizvođač mobilnih telefona može razvijati, unaprijediti, prilagoditi i koristiti. To je proizvođačima olakšalo i pojednostavilo razvoj programske podrške za njihove mobilne uređaje jer nije više svaki proizvođač morao investirati u stvaranje i održavanje vlastitog operacijskog sustava, već su svi mogli koristiti prilagođenu inačicu Androida smanjujući troškove razvoja mobilnih uređaja, a samim time i prodajne cijene tih uređaja. Upravo se iz tog razloga Android popularizirao među proizvođačima mobilnih telefona i postao de facto izbor operacijskog sustava za nove uređaje, što je dovelo do niže cijene uređaja koja je omogućila da se Android proširi među korisnicima u cijelom svijetu. Android se ponajviše raširio u zemljama u razvoju u kojima njegov glavni rival (iOS) nije mogao konkurirati zbog svoje visoke cijene koju si većina stanovnika tih zemalja nije mogla priuštiti. Navedeni su čimbenici pridonijeli tomu da je Android danas u svijetu dominantan operacijski sustav za mobilne uređaje. [4]

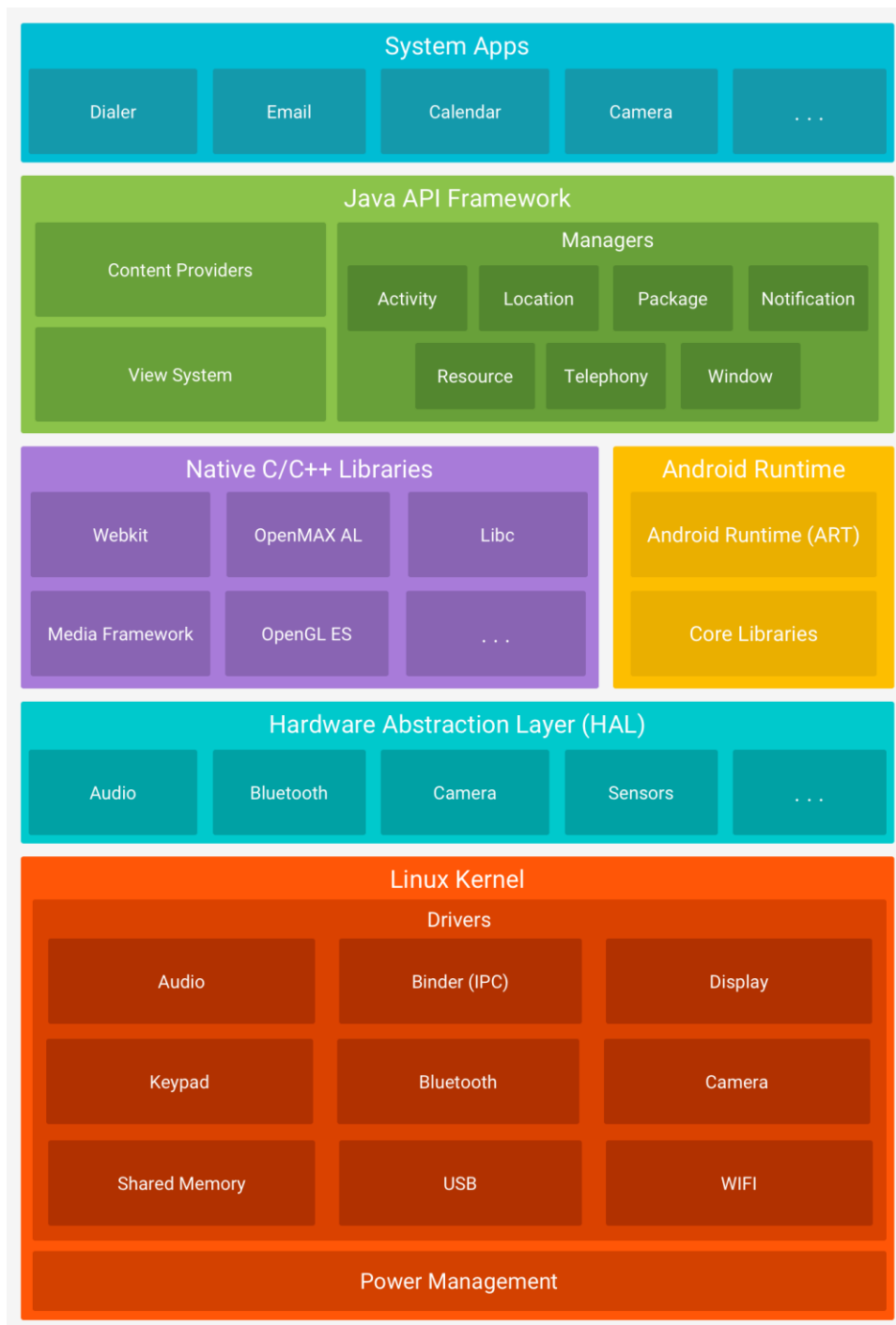
Njegova ga svestranost, otvorenost, proširivost i popularnost čini izvrsnom platformom za razvoj i lansiranje aplikacija koje će dosegnuti mnogobrojne korisnike diljem svijeta.



Slika 3.1 Početni zaslon operacijskog sustava Android

3.1.1. Arhitektura Androida

Arhitektura operacijskog sustava Android stog je programske podrške otvorenog koda koji se temelji na Linuxu, a sastoji se od nekoliko slojeva gdje svaki niži sloj nudi usluge višem sloju. Na taj je način korisnička aplikacija kroz seriju aplikacijskih programskih sučelja različitih slojeva povezana sa sklopovljem uređaja koji pokreće operacijski sustav. Povezanost pojedinih slojeva jasno se može uočiti na slici 3.2.



Slika 3.2 Prikaz arhitekture Androida

Temelj je Android platforme Linuxova jezgra koja predstavlja sloj apstrakcije između sklopovlja uređaja i ostalih komponenti arhitekture operacijskog sustava. Jezgra rukuje upravljačkim programima (primjerice, upravljački program za zaslon ili upravljački program kamere) koji su potrebni za pravilan rad sustava. Osim toga, operacijski se sustav oslanja na Linuxovu jezgru za sve temeljne funkcionalnosti poput upravljanja nitima, memorijom, napajanjem, uređajima itd. Korištenjem jezgre Linuxa Android iskorištava prednosti ključnih sigurnosnih značajki te omogućuje proizvođačima uređaja jednostavan i dobro poznati razvoj upravljačkih programa. [5] [6]

Povrh jezgre nalazi se sloj apstrakcije sklopovlja (HAL, engl. *Hardware Apstraction Layer*) koji nudi standardna sučelja za izlaganje hardverskih mogućnosti uređaja okvirima Java aplikacijskih programskih sučelja više razine. Sastoji se od nekoliko modula biblioteka, svaki od kojih implementira sučelje za određenu vrstu komponente (primjerice senzora uređaja). [7]

Android Runtime (ART) okruženje je koje sadrži osnovne biblioteke (Core Libraries) i Dalvikom virtualni stroj (DVM) te pruža osnovu za okvir aplikacije. Na uređajima koji koriste minimalno Android verziju 5.0 (API razina 21), svaka je pokrenuta aplikacija vlastiti proces te ima vlastitu instancu *Android Runtime-a*. ART je napisan kako bi pokretao više virtualnih strojeva na uređajima koji posjeduju malo memorije na način da izvršava DEX datoteke – format bajtkoda posebno dizajniran za Android. [5] [6]

Brojne su komponente i usluge Androida, poput spomenutih HAL i ART, izgrađene na temelju izvornog koda koji se oslanja na izvorne biblioteke pisane u jezicima C i C++. Android platforma nudi Java okvire aplikacijskih programskih sučelja (API) koji omogućuju aplikacijama korištenje nekih funkcionalnosti navedenih izvornih biblioteka. [5]

Sloj aplikacijskog okvira (eng. *Application Framework*) pruža cjelokupan skup značajki operacijskog sustava dostupnih putem Java aplikacijskih programskih sučelja. Navedeni API-ji pružaju generičku apstrakciju za pristup sklopovlju i predstavljaju osnovne građevne jedinice za izradu Android aplikacije pojednostavljajući višestruko korištenje osnovnih komponenti i usluga sustava. [5]

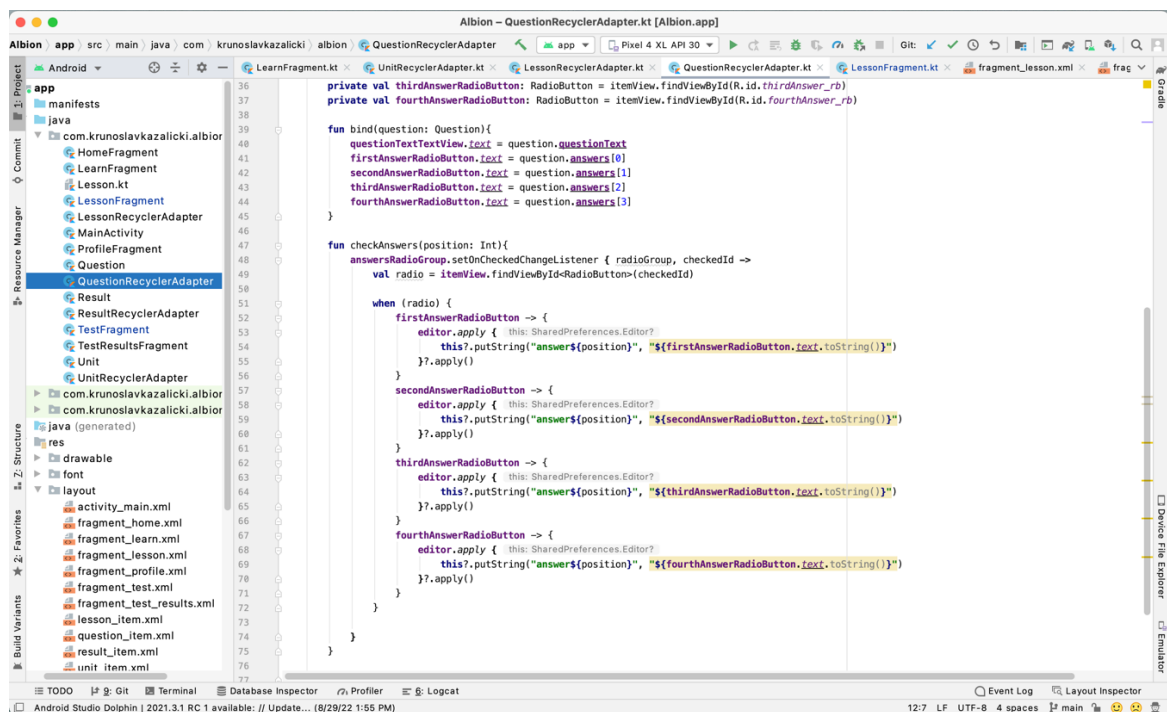
Aplikacije su najviši sloj Android arhitekture. Android dolazi s određenim brojem predinstaliranih aplikacija, no one nemaju poseban status u odnosu na aplikacije koje korisnik sam instalira na uređaj. To znači da aplikacije koje korisnik instalira mogu postati, primjerice, zadani internetski preglednik ili zadana tipkovnica. Sustavske aplikacije mogu pružiti ključne mogućnosti kojima razvojni programeri mogu pristupiti iz vlastite aplikacije. Na primjer, u slušaju da

aplikacija koju programer dizajnira želi slati SMS poruke, tu funkciju nije potrebno implementirati na vlastiti način, već je moguće pozvati bilo koju instaliranu aplikaciju koja omogućava razmjenu SMS poruka. [5]

3.2. Android Studio

Službeno integrirano razvojno okruženje (IDE) naziva Android Studio temelji se na IntelliJ IDEA platformi – moćnom uređivaču koda sa skupom korisnih alata za razvojne programere. Osim toga, Android Studio nudi brojne dodatne značajke specifične za razvoj Android aplikacija koje poboljšavaju produktivnost i jednostavnost korištenja razvojnog alata. Jedinstveno okruženje u kojem je moguć razvoj aplikacija za sve vrste Android uređaja, integrirani brzi emulator, podrška za mnoge programske jezike (C++, Java, Kotlin) i platforme (*Google Cloud Platform*, *Firebase* itd.) te fleksibilan sustav izgradnje aplikacije temeljen na *Gradle-u* samo su neke od korisnih značajki ovog razvojnog okruženja. [8]

Iako nije jedini dostupan alat za razvoj Android aplikacija, ipak je korišten za izradu aplikacije završnog rada zbog širokog spektra značajki i službene podrške razvijatelja operacijskog sustava.



Slika 3.3 Sučelje okruženja Android Studio

3.3. Kotlin

Programski jezik koji se primarno koristio za razvoj aplikacije završnog rada je Kotlin koji, u usporedbi s Javom, omogućuje postizanje iste funkcionalnosti s manje koda te donosi naprednije i modernije značajke programskog jezika, a osim toga omogućuje i korištenje dijelova Java koda.

Kotlin je višepatformski, statički tipiziran programski jezik opće namjene kojeg je razvila tvrtka JetBrains. Izvorno je napravljen na Java platformi i Java virtualnom stroju (JVM, engl. *Java Virtual Machine*) s ciljem interoperabilnosti s programskim jezikom Java. To znači da se program pisan u Kotlinu može koristiti, odnosno pozivati Java biblioteke, aplikacijska programska sučelja i programe. To je jedan od ključnih razloga zašto je Kotlin postao službeni programski jezik za razvoj Android aplikacija; jer je omogućio da nove aplikacije i proširenja za Android mogu upotrijebiti sve postojeće biblioteke, radne okvire te ostali gotovi programski kod pisan u Javi. [9] [10]

3.4. XML

Za dizajniranje, odnosno opisivanje izgleda izrađene aplikacije koristio se XML jezik.

EXtensible Markup Language (XML) jezik je za označavanje koji omogućava pohranu, prijenos i prezentiranje različitih vrsta podataka. Definirani skup pravila osigurava kodiranje dokumenata u oblik koji je čitljiv ljudima i računalima. [11]

U razvoju aplikacija za Android, XML se koristi pri raspoređivanju, oblikovanju i uređivanju vizualnih elemenata zaslona aplikacije. Ti elementi nisu trajno formatirani na inicijalan način, već njima možemo manipulirati tijekom izvođenja aplikacije pomoću napisanog Kotlin koda. To znači da, primjerice, možemo imati grafički element stalne veličine definiran XML-om, čiju veličinu korisnik može promijeniti dodiranjem na isti zbog napisanog Kotlin koda koji zbog korisničke interakcije izmjenjuje dijelove XML-a.

3.5. Firebase

Firebase je platforma raznih usluga i značajki dostupnih programerima na korištenje za razvoj mrežnih i mobilnih aplikacija. Moguće je koristiti i usklađivati različite usluge kako bi odgovarali potrebama razvijane aplikacije. Prvotni je proizvod tima koji razvija ovu platformu bila baza

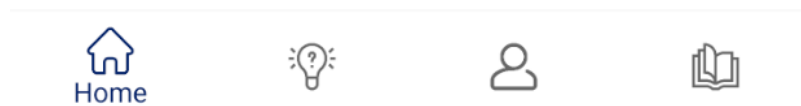
podataka u stvarnom vremenu. No rastom popularnosti platforme, proširila se i linija proizvoda te sada čini potpuni paket za razvoj aplikacija. [12]

Primarna usluga platforme *Firebase* korištena u izradi aplikacije završnog rada je *Cloud Firestore* – fleksibilna i skalabilna baza podataka. To je NoSQL baza podataka, što znači da podatke pohranjuje drugačije od relacijskih tablica te ne koristi SQL jezike za upravljanje bazom podataka. *Cloud Firestore* omogućava sinkronizaciju podataka kroz klijentske aplikacije u stvarnom vremenu, a osim toga nudi i izvanmrežnu podršku, što omogućuje izradu responzivnih aplikacija bez obzira na stabilnost internetske povezanosti. [13]

4. IZRADA APLIKACIJE I PRIKAZ FUNKCIONALNOSTI

4.1. Navigacija aplikacije

Jednostavno i brzo navigiranje kroz dijelove aplikacije ostvareno je korištenjem *BottomNavigationView-a*. U svakom trenutku, bez obzira na to gdje se u aplikaciji nalazili, dostupna je donja traka za navigaciju prikazana na slici 4.1 koja omogućuje prelazak u druge dijelove aplikacije. Osim toga, u pojedinim zaslonima aplikacije dostupni su gumbi koji omogućuju fluidan prelazak na druge, kontekstualno logičke zaslone. Primjerice, u fragmentu *Home*, nakon poglavlja teksta koji korisniku daje upute i objašnjenja o korištenju aplikacije, nudi se gumb koji jasno vodi na fragment za testiranje znanja engleskog jezika, što je logički korak na početku korištenja aplikacije.

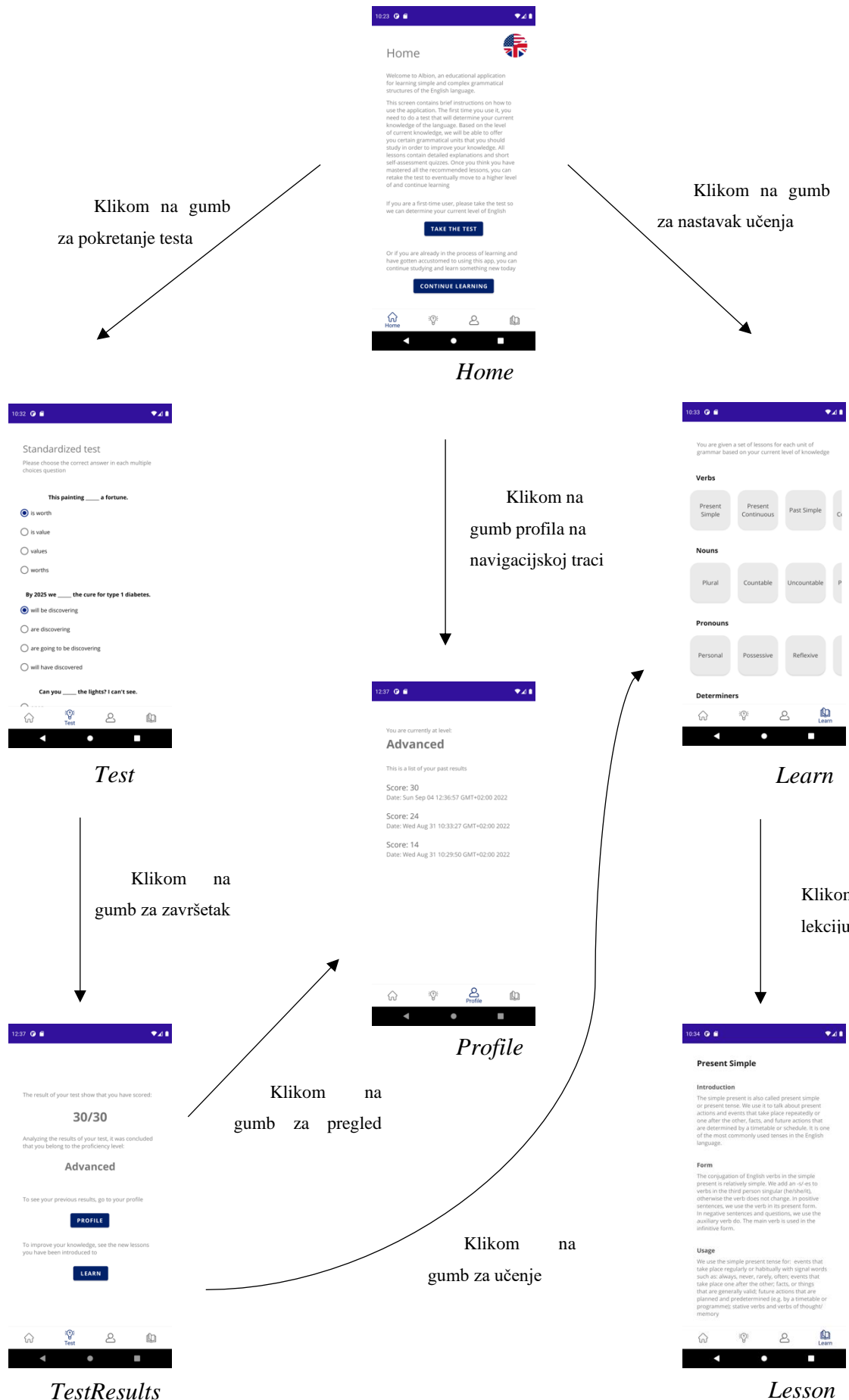


Slika 4.1 Traka za navigaciju s aktivnim fragmentom *Home*

4.2. Dijagram i opis toka aplikacije

Aplikacija je podijeljena na četiri glavna dijela: *Home*, *Profile*, *Test* i *Learn*. Zbog svojstvenog načina navigacije kroz aplikacije putem navigacijske trake, glavni su dijelovi implementirani u obliku fragmenata, a glavna i jedina aktivnost odgovorna je za osiguravanje pravilnog funkcioniranja navigacije.

Navedenim je fragmentima moguće pristupiti iz bilo kojeg drugog fragmenta u bilo koje vrijeme. No aplikacija ima i dva „skrivena“ fragmenta do kojih se dolazi samo određenim akcijama ili nakon ispunjavanja određenih uvjeta u glavnim fragmentima. Ti su skriveni fragmenti *TestResults* i *Lesson*.



Slika 4.2 Dijagram toka aplikacije

Slika 4.2 pokazuje kretanje kroz aplikaciju odnosno tok kroz fragmente. Pri pokretanju aplikacije korisnik se nailazi na zaslonu *Home* koji služi kao polazište za pokretanje drugih dijelova aplikacije. Novim je korisnicima zatim predloženo pokretanje ispita znanja, a to mogu učiniti klikom na gumb koji otvara fragment *Test* gdje korisnik odgovara na postavljena mu pitanja. Klikom na gumb za predaju testa otvara se fragment *TestResults* u kojemu se prikazuje ostvareni rezultat. Iz navedenog zaslona korisnik klikom na gumb može nastaviti proučavati gramatičke lekcije ili ih može započeti proučavati (u slučaju novog korisnika) u fragmentu *Learn*. Također, postoji gumb koji otvara pregled dosadašnjih rezultata u fragmentu *Profile*.

Stalni korisnici aplikacije u polazištu *Home* mogu klikom gumba prijeći na fragment *Learn* gdje im se nudi spektar lekcija primjerenih trenutnoj razini znanja, a iste su podijeljene u cjeline. Klikom na neku od lekcija otvara se fragment *Lesson* u kojem se nalaze definicije, primjeri i primjene odabrane gramatičke strukture.

Tako koncipirani tok omogućuje intuitivnu navigaciju i prelazak s jednog zaslona (fragmenta) aplikacije na drugi. Korisnik se u svakom trenutku može vratiti na početni zaslon ili na neki od glavnih fragmenata.

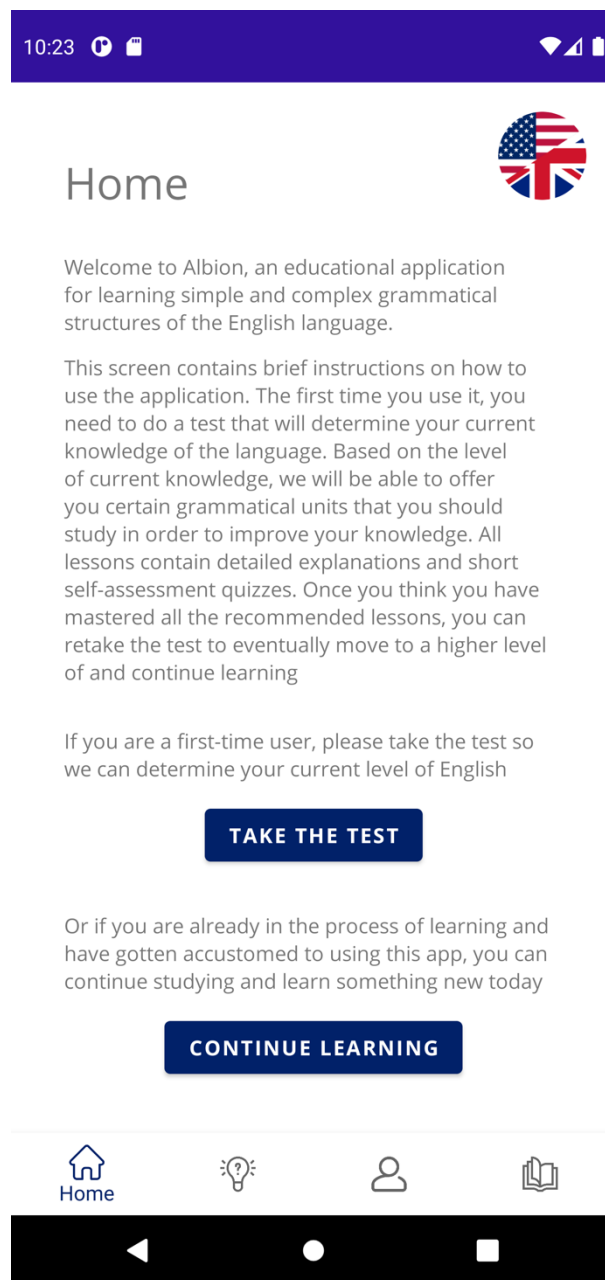
4.3. Glavni dijelovi aplikacije i funkcionalnosti

Fragmenti imaju vlastiti izgled, odnosno datoteke koje opisuju njihov izgled; vlastito ponašanje s vlastitim povratnim pozivima životnog ciklusa, a upravo su zbog toga pogodno sredstvo za implementaciju glavnih dijelova/zaslona aplikacije. Izgled je svakog dijela moguće jednostavno definirati prema sadržaju koji taj određeni fragment prezentira korisniku. Također, izmjene su zaslona jednostavne i agilne.

Svakim se ulaskom u aplikaciju prvobitno prikazuje fragment *Home* koji za korištenje aplikacije usmjerava korisnika na daljnje fragmente čije funkcionalnosti korisnik zatim može upotrebljavati. U sljedećim su poglavljima opisani izgledi, funkcionalnosti koje nude i prikaz načina implementacije pojedinih spomenutih fragmenata.

4.3.1. Home

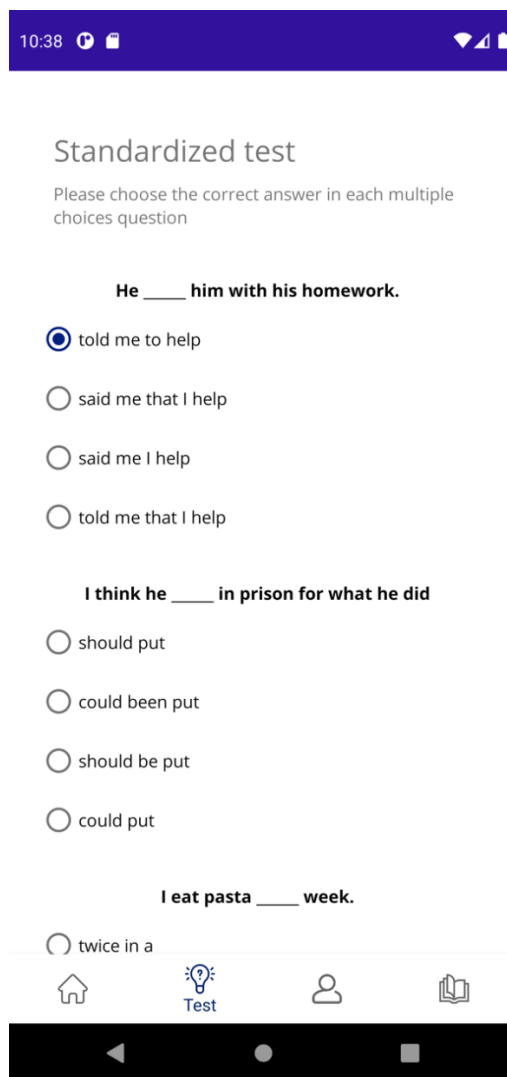
Home je polazni zaslon aplikacije koji se sastoji od nekoliko poglavlja teksta implementiranih pomoću *TextView* elemenata. U tim su poglavljima sadržane okvirne upute o tome kako je aplikacija koncipirana i kako se koristi. Kako je u spomenutim uputama navedeno, kod prvog korištenja aplikacije korisnik prvo rješava standardizirani test kojim se procjenjuje razina njegovog trenutnog poznavanja engleskog jezika. Odlazak na zaslon testa moguć je intuitivno pozicioniranim gumbom, ali i preko navigacijske trake na dnu zaslona.



Slika 4.3 Zaslon fragmenta *Home*

4.3.2. Test

Podloga izgleda Fragmenta *Test* je *ScrollView* koji omogućava lako listanje i pravilan prikaz svih elemenata. Izgled fragmenta sastoji se od *TextView*-a koji navodi na kojem se zaslonu korisnik nalazi te sadrži kratku uputu o ispunjavanju standardiziranog testa kojem će korisnik pristupiti. Ispod njih nalazi se *RecyclerView* – element koji omogućava lako i resursno nezahtevno prezentiranje više elemenata koji dijele isti izgled, a koji se razlikuju u sadržaju, odnosno podacima koje prikazuju. U kontekstu navedenog fragmenta to znači da *RecyclerView* element reciklira elemente s pitanjem i odgovorom koji su uvijek na isti način vizualno definirani, ali se kod svakog elementa mijenja tekst s postavljenim pitanjem i eventualnim ponuđenim odgovorima. Ti su elementi *CardView*-ovi koji u sebi imaju *TextView* elemente s tekстом pitanja, *RadioGroup* element s određenim brojem *RadioButton* elemenata koji predstavljaju višestruki izbor odgovora.



Slika 4.4 Zaslón fragmenta *Test*

Kako je i prikazano na slici 4.4, fragment je implementiran na način da poziva funkciju za dohvaćanje podataka s *Cloud Firestore* baze podataka. Unutar poziva navedene funkcije deklarira se lista pitanja te se deklarira svako pitanje unutar petlje koja dohvaća attribute pitanja iz baze podataka i inicijalizira to pitanje koje se zatim dodaje u navedenu listu. Lista se pitanja prosljeđuje odgovarajućoj adapter klasi unutar poziva funkcije za inicijalizaciju *RecyclerView* elementa.

```
val db = Firebase.firestore
db.collection( collectionPath: "TestQuestions")
    .get()
    .addOnSuccessListener { result ->
        val questions: MutableList<Question> = ArrayList<Question>()

        for (document in result){
            var questionText = document.data.get("questionText").toString()
            var answers: List<String> = document.data.get("answers") as List<String>
            var correctAnswer = document.data.get("correctAnswer").toString()
            var question = Question(questionText, answers, correctAnswer)

            questions.add(question)
        }

        view.findViewById<RecyclerView>(R.id.questions_rv).apply { this: RecyclerView!
            layoutManager = LinearLayoutManager(this@TestFragment.context)
            adapter = QuestionRecyclerViewAdapter(questions, sharedPreferences.editor)
        }
    }
    .addOnFailureListener { exception ->
        Log.w(ContentValues.TAG, msg: "Error getting documents.", exception)
    }
}
```

Slika 4.5 Implementacije funkcije za dohvaćanje podataka s *Cloud Firestore*-a

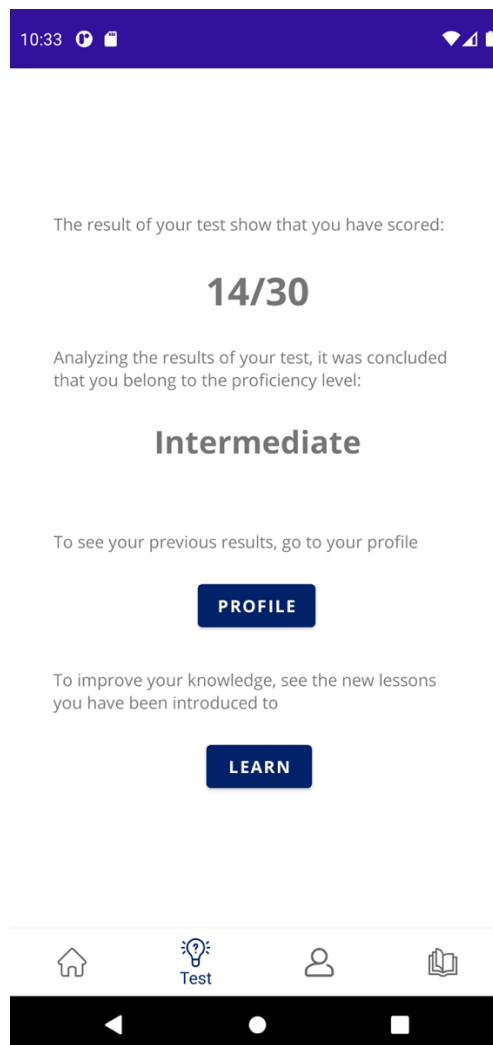
Klikom na gumb za predaju testa, unutar poziva funkcije za dohvaćanje podataka iz baze podataka, broje se točni odgovori na način da se unutar petlje dohvaćaju korisnički dani odgovori spremljeni u *sharedPreferences* datoteku i uspoređuju s odgovarajućim točnim odgovorima iz baze podataka. Nakon izračunatog broja točnih odgovora, postignuti se rezultat, zajedno s vremenom predaje ispita, sprema u bazu podataka *Results* koja sadrži podatke o rezultatima. Nakon toga pokreće se fragment *TestResults* koji korisniku prezentira izračunati brojevi uvid u rezultat, kao i razinu znanja koju taj rezultat donosi, odnosno kojoj korisnik u tom trenutku pripadne.

Pitanja korištena u standardiziranom testu aplikacije nalaze se u unaprijed postavljenoj bazi podataka *Cloud Firestore*-a. Svako pojedino pitanje ima atribut koji predstavlja tekst tog pitanja,

niz mogućih odgovora koji će biti ponuđeni korisniku, kao i atribut koji označava točan odgovor i koji služi da se pri usporedbi na pravilan način vrednuju korisnički odgovori za svako pojedino pitanje.

4.3.3. TestResults

Izgled fragmenta *TestResults* čini nekoliko *TextView* elemenata koji opisuju rezultat, razinu znanja i daju upute o radnjama koje pojedini gumbovi pokreću. U centru su pozornosti dva veća *TextView* elementa koja prikazuju prave podatke o ostvarenom bodovnom rezultatu i razini znanja korisnika. Spomenuti gumbovi omogućuju prelazak u fragmente za učenje i pregled rezultata.



Slika 4.6 Zaslon fragmenta *TestResults*

Fragment je implementiran tako da dohvaća zadnji ostvoreni rezultat iz baze podataka rezultata koji je netom prije pokretanja fragmenta spremljen. Na temelju tog rezultata odlučuje se razina znanja i svi se navedeni podaci prikazuju u sučelju kao što je navedeno na slici 4.7.

```
db.collection( collectionPath: "Results")
    .orderBy( field: "timestamp", Query.Direction.DESENDING).limit( limit: 1)
    .get()
    .addOnSuccessListener { result ->
        for (document in result) {
            var score = document.data.get("result").toString()
            var level: String

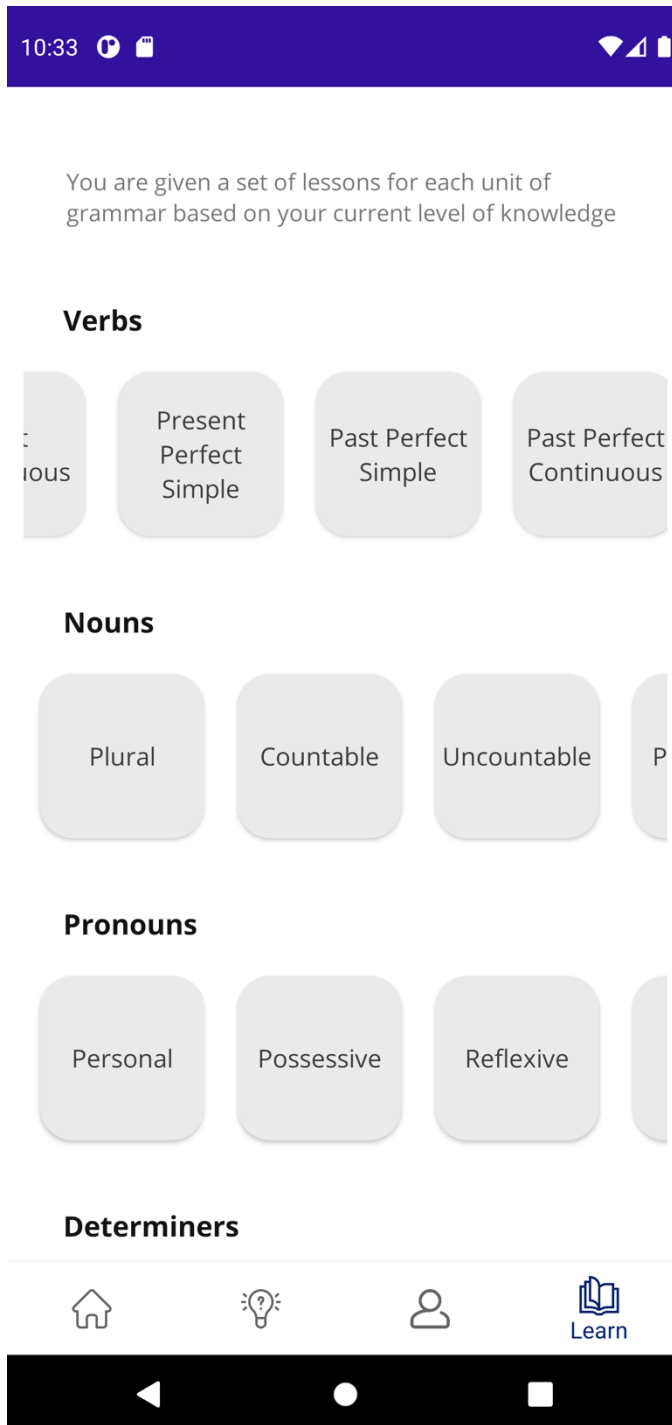
            if (score.toInt() >= 21) {
                level = "Advanced" //A
            } else if (score.toInt() >= 12 && score.toInt() < 21) {
                level = "Intermediate" //B
            } else {
                level = "Beginner" //C
            }

            numberResultTextView.text = "${score}/30"
            levelResultTextView.text = level
        }
    }
    .addOnFailureListener { exception ->
        Log.w(ContentValues.TAG, msg: "Error getting documents.", exception)
    }
}
```

Slika 4.7 Implementacija dohvaćanja rezultata iz baze podataka

4.3.4. Learn

Fragment *Learn* najvažniji je dio aplikacije jer sadržava skup svih gramatičkih cjelina, lekcija, primjera, objašnjenja i primjena istih. Podijeljen je u nekoliko skupina od kojih svaka predstavlja jednu cjelinu koja sadrži skup lekcija jedinstven za tu cjelinu. Svaka je lekcija smještena u karticu koja, ovisno o lekciji, sadrži određeni broj pravila, objašnjenja za isto, primjene tog pravila i realnih primjera korištenja za svaku od primjena u slučaju da ih je više. Kako je aplikacija koncipirana na temelju razina poznavanja jezika, tako se pokretanjem fragmenta *Learn* prikazuju samo do sada naučene lekcije i lekcije koje korisnik tek treba proučiti.



Slika 4.8 Zaslona fragmenta *Learn*

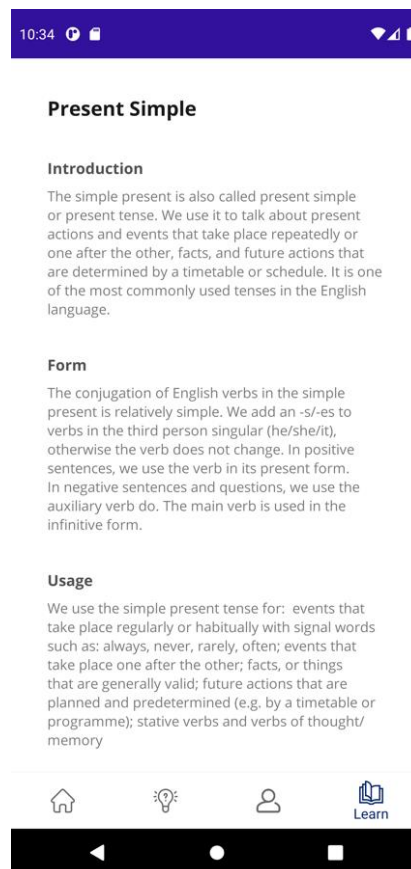
Izgled je fragmenta definiran pomoću *RecyclerView*-a, čiji su elementi datoteke izgleda cjeline, a koji su opisani *TextView*-om (za naslov) i *RecyclerView*-om (za listu lekcija). Radi se o ugniježđenim *RecyclerView* elementima, gdje se prvi lista vertikalno i prikazuje *RecyclerView* elemente unutar sebe izlistane horizontalno. Elementi unutrašnjeg *RecyclerView*-a su *CardView*-ovi koji nose naziv određene lekcije.

U implementaciji fragmenta inicijalizirano je nekoliko popisa lekcija za svaku cjelinu, a ovisno o razini znanja, popis sadrži manje ili više lekcija za tu cjelinu. Pozivom funkcije dohvaćaju se podaci iz baze podataka rezultata gdje se uzima zadnji ostvareni rezultat i prema njemu se određuje koja će od deklariranih listi lekcija biti prosljeđena funkciji za prikaz *RecyclerView* elementa.

Pri inicijalizaciji svakog elementa unutar vanjskog *RecyclerView*-a, inicijalizira se unutrašnji *RecyclerView* koji inicijalizira elemente s nazivima lekcija. Klikom na jednu od tih lekcija ime se prosljeđuje fragmentu *Lesson* koji je zadužen za prikazivanje baš te kliknute lekcije.

4.3.5. Lesson

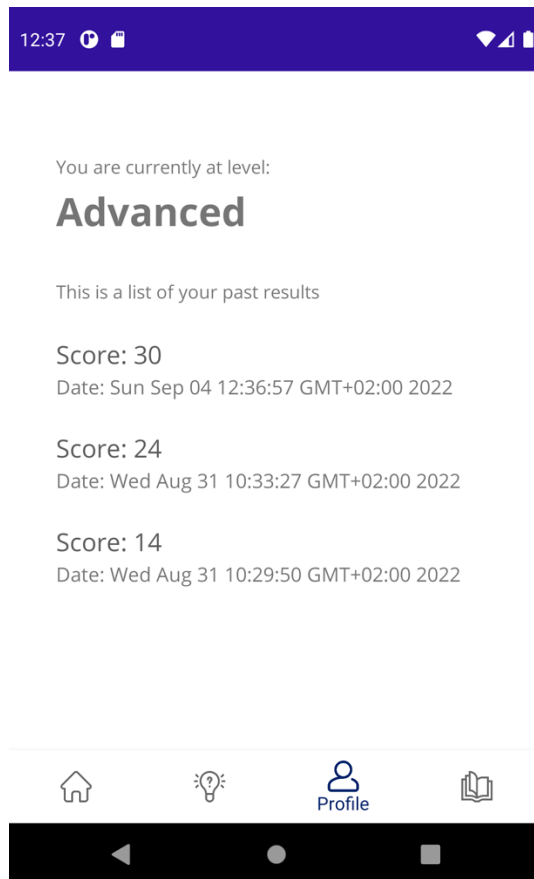
Izgled je fragmenta *Lesson* definiran s nekoliko *TextView* elemenata koji korisniku prezentiraju lekciju s primjenama, tvorbom i primjerima korištenja pravila, dok je za pokretanje samog fragmenta, odnosno njegovu inicijalizaciju, potreban ključ lekcije. Njega prosljeđuje fragment *Learn* i to onaj ključ lekcije na koju je korisnik kliknuo. Pomoću spomenutog se ključa iz baze podataka dohvaćaju točno određeni podaci koji se zatim prikazuju korisniku.



Slika 4.9 Zaslون fragmenta *Lesson*

4.3.6. Profile

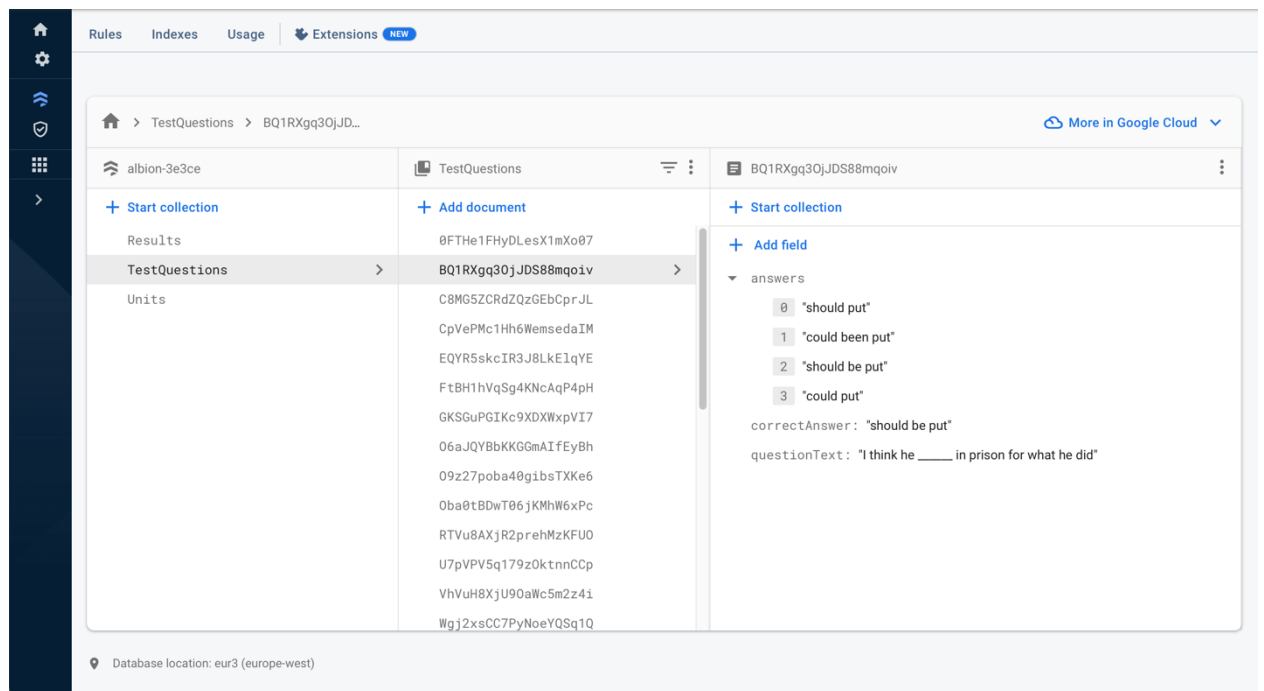
Profile je dio aplikacije koji služi informiranju korisnika o njegovoj trenutnoj razini poznavanja jezika, rezultatima i napredcima ostvarenim testiranjem znanja standardiziranim testom. Izgled fragmenta implementiran je jednostavnim korištenjem nekoliko *TextView* elemenata koji sadrže razne opise, poput trenutne razine korisnika, rezultate testova i slično. Dio se *TextView* elemenata koji sadrži promjenjive podatke inicijalizira unutar bloka koda za dohvaćanje podataka s *Cloud Firestore*-a kako bi se spremljene informacije mogle prikazati korisniku.



Slika 4.10 Zaslون fragmenta *Profile*

4.4. Cloud Firestore baze podataka

Za dohvaćanje i spremanje podataka koristi se već opisana tehnologija *Firebase*, odnosno njezina usluga *Cloud Firestore*. Pomoću navedenog omogućeno je lako i brzo dohvaćanje podataka iz različitih baza podataka, bez pisanja SQL koda. Omogućeno je stvaranje gotovih baza podataka koje sadrže pitanja standardiziranog testa i informacije o svakoj pojedinoj lekciji, kao i baze podataka koje se popunjavaju podacima ovisno o korisnikovim rezultatima.



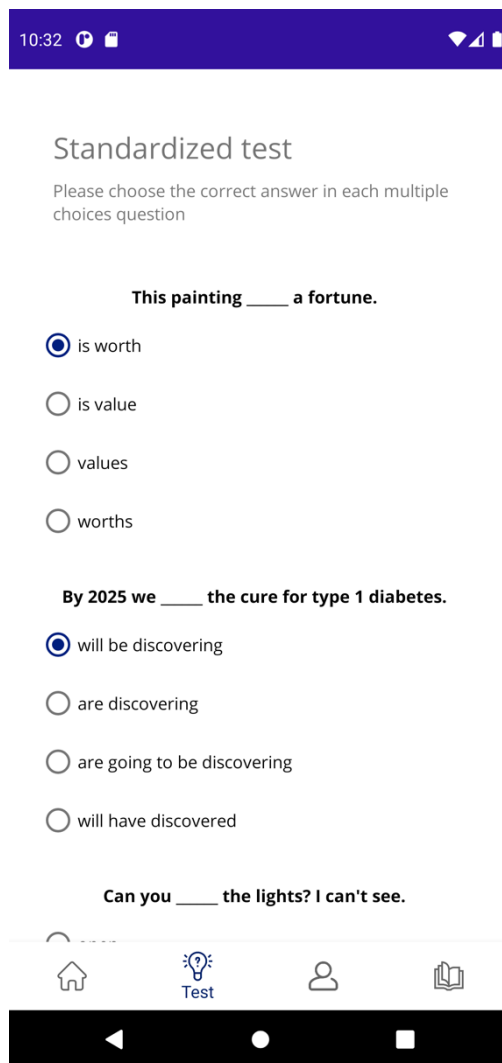
Slika 4.11 *Cloud Firestore* baze podataka korištene u aplikaciji

5. TESTIRANJE RADA APLIKACIJE I PRIKAZ KORIŠTENJA ZNAČAJKI

U sljedećim će se poglavljima opisati rad same aplikacije i prikaz korištenja značajki iste čija je implementacija prethodno opisana.

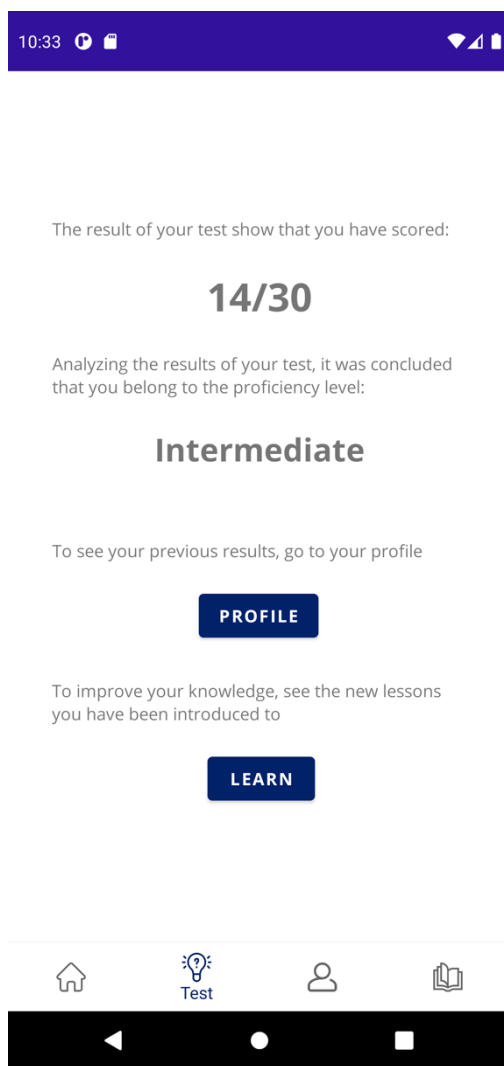
5.1. Rješavanje inicijalnog testa znanja

S početnog zaslona pomoću gumba otvaramo zaslon za inicijalno testiranje i određivanje trenutne razine znanja. Test ćemo simulirati na način da ćemo polovicu pitanja odgovoriti točno, a preostalu polovicu netočno. Slika 5.1 pokazuje navedenu simulaciju testa znanja.



Slika 5.1 Rješavanje inicijalnog testa znanja

Odgovorivši na pitanja, završavamo test klikom na gumb za završetak ispita, nakon čega nam se otvara novi zaslon koji prikazuje ostvareni rezultat i dodijeljenu razinu znanja, što se može vidjeti na slici 5.2.

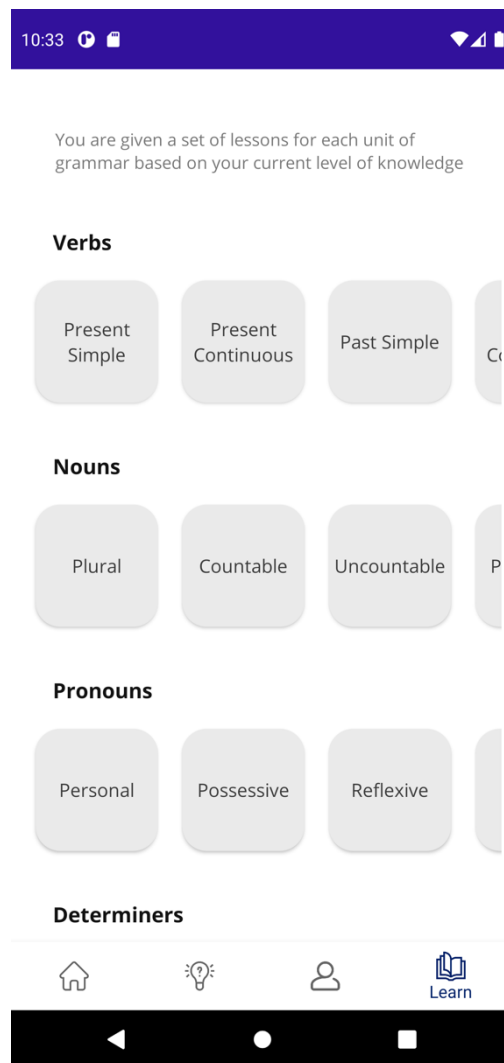


Slika 5.2 Inicijalno dodijeljena razina znanja

Kako smo naveli, otprilike polovica je odgovora bila točno riješena (14 točnih odgovora od 30 mogućih) i shodno tomu dodijeljena nam je srednja razina znanja, odnosno *intermediate*.

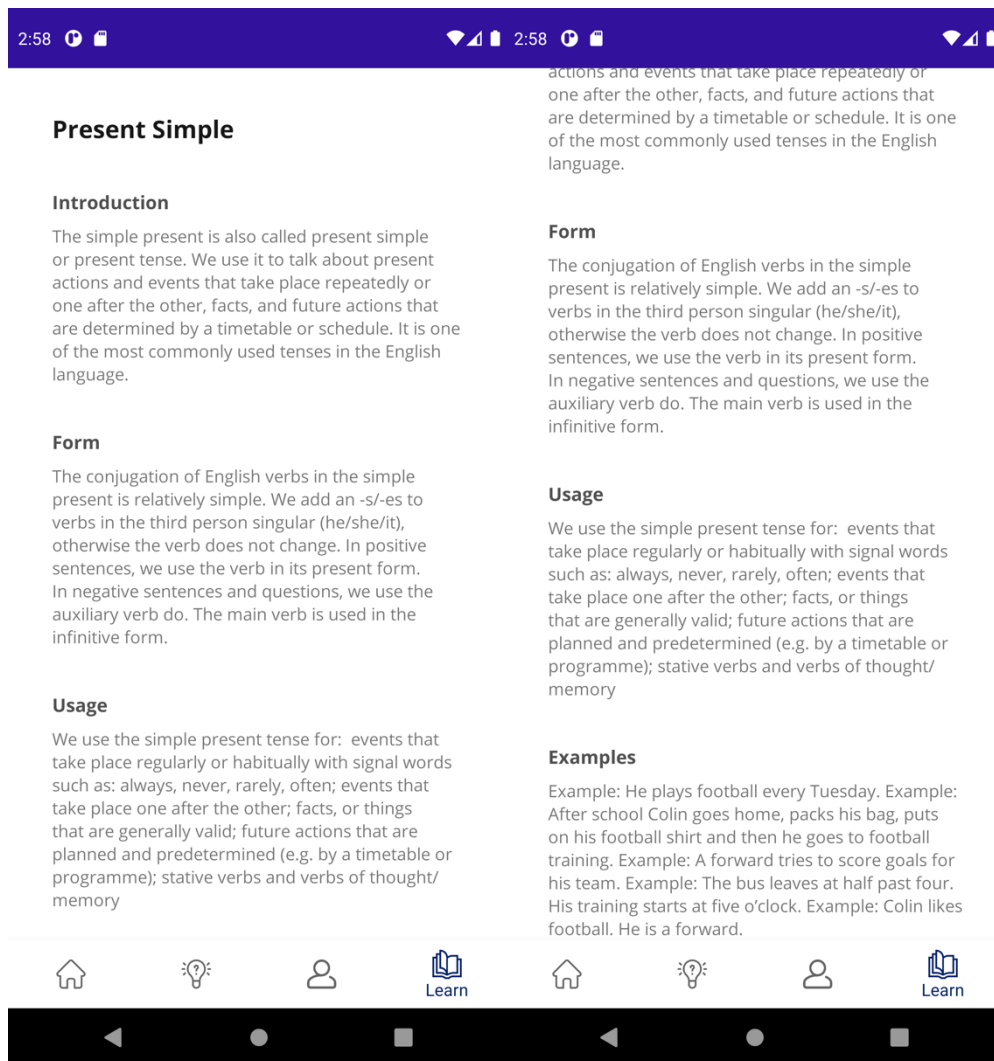
5.2. Proučavanje gramatičkih struktura

Kako smo inicijalnim rješavanjem standardiziranog testa smješteni u srednju kategoriju znanja, tako su nam na izbor zaslona *Learn* ponuđene samo one lekcije koje odgovaraju trenutnoj razini znanja. Za primjer ćemo promatrati cjelinu *Verbs* u kojoj su trenutno ponuđena samo jednostavna i neka složenija glagolska vremena te glagolska vremena koja služe kao podloga za formiranje struktura složenih glagolskih vremena koja će korisniku tek biti ponuđena.



Slika 5.3 Ponuđene lekcije za srednju razinu znanja

Za primjer ćemo ući u jednu od lekcija; konkretno, radi se o lekciji *Present Simple* unutar cjeline *Verbs*. Stvarnim korištenjem aplikacije korisnik bi proučavao ponuđene mu cjeline i usvajao znanja koja one nude.

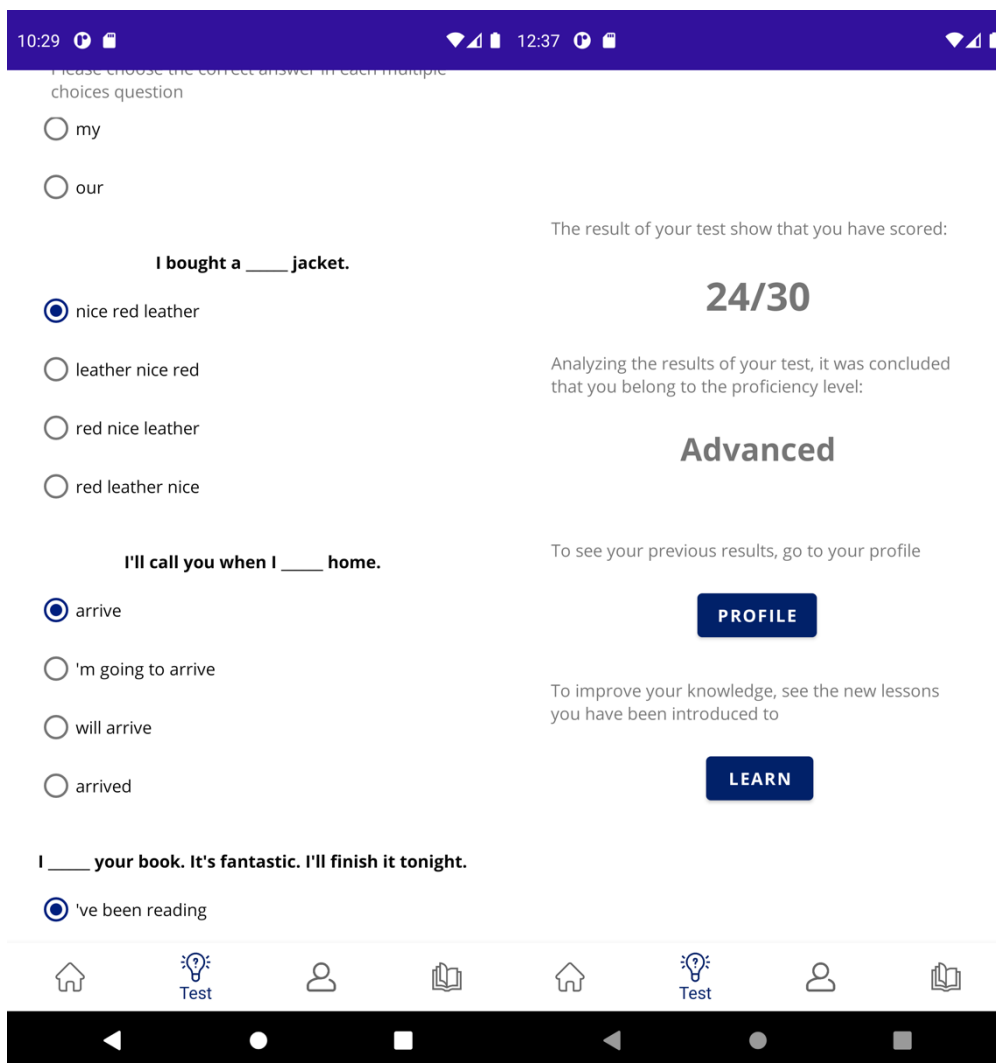


Slika 5.4 Zaslonek lekcije *Present Simple*

Kao što je vidljivo na slici, korisniku je najprije dan kratki uvod u cjelinu koji sažeto iznosi sadržaj iste. Zatim, u konkretnom primjeru, slijedi tvorba – odnosno dano je pravilo o tvorbi glagola navedenog glagolskog vremena. Nakon toga su objašnjenje situacije u kojima se navedeno glagolsko vrijeme koristi u engleskom jeziku. Naposljetku se nalaze primjeri rečenica u zadanom glagolskom vremenu.

5.3. Ponovno testiranje i prelazak u novu razinu znanja

Sada ćemo ponovno pokrenuti test kao simulaciju korisnika koji je usvojio znanja novih lekcija te odgovoriti na većinu pitanja točno.



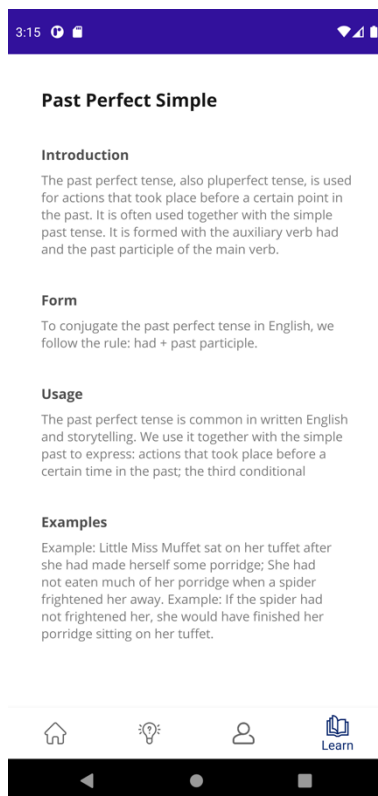
Slika 5.5 Rješavanje ispita s većinom točnih odgovora i rezultati istoga

Završetkom ispita odgovorili smo točno na 24 od 30 pitanja, što znači da smo spremni prijeći u novu razinu znanja kao što je i očekivano. Time su nam sada ponuđene nove lekcije kojima prije nismo mogli pristupiti jer su bile neprimjerene prethodnoj razini znanja.

Na promatranom primjeru cjeline *Verbs* sada su nam ponuđene dodatne lekcije koje prije nismo vidjeli niti smo im mogli pristupiti dok smo se nalazili u srednjoj razini znanja. To jasno možemo vidjeti na slici 5.6 gdje su nam sada ponuđena nova i složenija glagolska vremena poput *Past Perfect Simple* i slično.



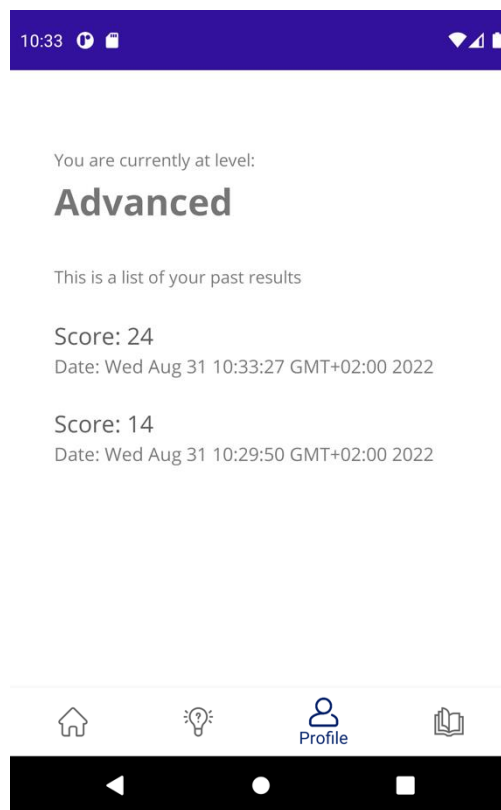
Slika 5.6 Prikaz ponuđenih lekcija više razine znanja



Slika 5.7 Zaslona lekcije *Past Perfect Simple*

5.4. Pregled dosadašnjih rezultata testiranja

U ovome se poglavlju prikazuje još jedna od funkcionalnosti aplikacije – pregled prethodnih rezultata testiranja i prikaz trenutne razine poznavanja jezika. Iz bilo kojeg dijela aplikacije i na različite načine možemo otvoriti fragment *Profile*. Na slici 5.8 možemo vidjeti kako su uspješno zabilježeni i prikazani rezultati ostvareni tijekom testiranja aplikacije, odnosno simuliranja testova znanja. Također, jasno su prikazani datumi i vrijeme testiranja. Isto tako, prikazano je da se korisnik nalazi u naprednoj razini poznavanja jezika.



Slika 5.8 Pregled dosadašnjih rezultata

6. ZAKLJUČAK

Zadatak završnog rada bio je dizajnirati i isprogramirati mobilnu aplikaciju za proučavanje gramatičkih struktura engleskog jezika koje ovise o trenutnoj razini znanja korisnika.

U uvodnim je poglavljima opisana važnost poznavanja engleskog kao stranog jezika, sveprisutnost mobilnih uređaja u ljudskih života te kako se iz tih dviju spoznaja rodila ideja o izrađenoj aplikaciji. Analizirana su neka od postojećih rješenja na tržištu te su uspoređene različite aplikacije. Određeni su aspekti analiziranih aplikacija poslužili kao inspiracija za implementaciju nekih funkcionalnosti izrađene aplikacije. Android je izabran kao operacijski sustav za izradu aplikacije zbog svoje raširenosti, broja korisnika i dostupnosti različitih tehnologija za izradu aplikacija za tu platformu. Android Studio je program koji je korišten za izradu same aplikacije, a izabran je zbog svoje integracije s Androidom, raznih alata koji olakšavaju programiranje aplikacijskih rješenja te podrške koju nudi u razvoju aplikacija. Aplikacija je pisana programskim jezikom Kotlin, koji je službeni jezik za razvoj Android aplikacija, a ima moćne i suvremene značajke programskog jezika koje omogućuju ostvarivanje više funkcionalnosti u manje linija koda otklanjajući ponavljajući redundantni kod. Kao sustav za upravljanje bazom podataka odabran je *Google-ov* servis *Cloud Firestore* koji omogućuje jednostavnu integraciju s aplikacijom koja se izrađuje; brzo i jednostavno spremanje i dohvaćanje podataka u realnom vremenu, a upravo su to zahtjevi izrađene aplikacije i iz tog je razloga izabrana ova tehnologija.

Izrada aplikacije započela je određivanjem zaslona i funkcionalnosti koje će svaki od zaslona obavljati. Navigacija kroz aplikaciju ostvarena je korištenjem fragmenata za prikazivanje glavnih elemenata aplikacije, a donja traka za navigaciju implementirana je koristeći *BottomNavigationView*. Pomoću navedene trake moguće je navigirati između pet glavnih zaslona aplikacije, dok je dvama „skrivenim“ zaslonima moguće pristupiti iz glavnih dijelova određenim akcijama (poput ispunjavanja testa). Pitanja za test znanja kojem korisnik pristupa dohvaćaju se iz odgovarajuće baze podataka spomenutog servisa *Cloud Firestore*, a u isti se servis u bazu podataka za rezultate spremaju ostvareni rezultati pokušaja rješavanja standardiziranog testa. Iz navedenog se također dohvaćaju podaci o lekcijama koji se korisniku prikazuju na njegov zahtjev. Zaslone za izbor lekcija implementiran pomoću ugniježđenih *RecyclerView* elemenata koji pružaju pregledan i kategoriziran izbor lekcija. Tijekom izrade aplikacije pojavljivali su se određeni problemi poput problema pohrane korisničkog odgovora unutar *RecyclerView* elemenata, koji je riješen korištenjem *SharedPreferences* datoteke u koju su pohranjeni odgovori.

Izrađena aplikacija, za razliku od analiziranih sličnih aplikacija, pruža korisniku uvid u njegovo trenutno poznavanje jezika i usmjerava ga na određene gramatičke strukture koje bi trebao proučiti, dok kod analiziranih aplikacija korisnici izabiru koje će lekcije proučavati ne znajući pri tome koje su pogodne za njihovu razinu znanja niti koja je to razina znanja. No neke od analiziranih aplikacija imaju bolje prezentirane lekcije i veći izbor istih.

Sve su planirane funkcionalnosti aplikacije ostvarene i realizirane. No kako se radi o manjem projektu, postoje mogućnosti za proširenje aplikacije – poput određenih estetskih poboljšanja, uvođenja mogućnosti više profila, stvaranje vizualnih elemenata za reklame koje bi mogle služiti kao eventualni izvor prihoda i slično. Također, jedno od zahtjevnijih poboljšanja bilo bi napraviti modulirano sučelje za prikaz lekcija čiji bi se izgledao morao mijenjati u ovisnosti o tome koja je lekcija otvorena. Zbog toga što svaka pojedina lekcija može imati različitu strukturu, različit broj primjera ili primjena, odnosno svaka bi se pojedina lekcija morala prikazivati na sebi svojstven način, a implementacija takvog sučelja može biti izazovna.

LITERATURA

- [1] The most spoken languages in the world, Berlitz, dostupno na: <https://www.berlitz.com/blog/most-spoken-languages-world>, [30. svibnja 2022.]
- [2] Why Do Multiple-Choice Questions Help Learning?, EducationQuizzes, dostupno na: <https://www.educationquizzes.com/knowledge-bank/why-do-multiple-choice-questions-help-learning/>, [31. svibnja 2022.]
- [3] Android Operating System, Investopedia, dostupno na: <https://www.investopedia.com/terms/a/android-operating-system.asp>, [31. svibnja 2022.]
- [4] The history of Android: The evolution of the biggest mobile OS in the world, AndroidAuthority, dostupno na: <https://www.androidauthority.com/history-android-os-name-789433/>, [31. svibnja 2022.]
- [5] Platform Architecture, Android Developers, dostupno na: <https://developer.android.com/guide/platform>, [14. lipnja 2022.]
- [6] Android Architecture, GeeksforGeeks, dostupno na: <https://www.geeksforgeeks.org/android-architecture/>, [14. lipnja 2022.]
- [7] D. Huang, H. Wu, Mobile Cloud Computing, Morgan Kaufmann, 2018, stranice 31-64
- [8] Meet Android Studio, Android Developers, dostupno na: <https://developer.android.com/studio/intro>, [15. lipnja 2022.]
- [9] Kotlin, Gentoo, dostupno na: <https://wiki.gentoo.org/wiki/Kotlin>, [15. lipnja 2022.]
- [10] Kotlin (programming language), Wikipedia, dostupno na: [https://en.wikipedia.org/wiki/Kotlin_\(programming_language\)](https://en.wikipedia.org/wiki/Kotlin_(programming_language)), [15. lipnja 2022.]
- [11] XML, Wikipedia, dostupno na: <https://en.wikipedia.org/wiki/XML>, [16. lipnja 2022.]
- [12] Google Firebase, Google Wiki, dostupno na: https://google.fandom.com/wiki/Google_Firebase, [25. lipnja 2022.]
- [13] Cloud Firestore, Google Firebase, dostupno na: <https://firebase.google.com/docs/firestore>, [25. lipnja 2022.]

SAŽETAK

U sklopu završnog rada izrađena je mobilna aplikacija koja služi za samoučenje gramatičkih cjelina engleskog jezika, analizirane su postojeće aplikacije za učenje gramatike engleskog jezika kao i tehnologije primijenjene u izradi aplikacije. Programsko je rješenje izrađeno za operacijski sustav Android u razvojnom okruženju Android Studio koristeći programski jezik Kotlin i jezik za označavanje XML. Za potrebe baza podataka korištena je *Firebase*-ova usluga *Cloud Firestore*. Opisan je tok kroz dijelove aplikacije, princip kojim su oni povezani te način izrade pojedinih dijelova aplikacije. Aplikacija omogućava procjenu korisnikova znanja engleskog jezika, a na temelju dobivenih rezultata nudi mu određene gramatičke lekcije na proučavanje. Na kraju su rada testirane značajke aplikacije simulirajući novog korisnika sa srednjom razinom znanja koji, nakon proučavanja gramatičkih struktura, ponovo pristupa testu znanja i čiji rezultat pokazuje prelazak korisnika na višu razinu znanja.

Ključne riječi: Android aplikacija, gramatika, standardizirani test, učenje engleskog jezika

ABSTRACT

Title: Mobile application for self-studying of English language grammatical units based on the user's prior knowledge

The final paper elaborates on the development of a designed mobile application for self-studying of English language grammatical units, analyzes the existing applications for learning English grammar, as well as the technologies used in the application development. The application was created for the Android operating system in Android Studio integrated development environment using the Kotlin programming language and the XML markup language. Firebase's Cloud Firestore service was used as a database. The paper describes a step-wise application development process and connection mechanisms. Using the application, the user can assess his/her English language knowledge and use the results to study specific grammatical units. At the end of the paper, the application features were tested by simulating a new user with an intermediate knowledge level. Upon studying the grammatical structures, the user retakes the standardized test the result of which informs him/her of his/her advancement to a higher knowledge level.

Keywords: Android application, grammar, standardized test, learning English language