Izrada aplikacije "Virovitica nevidljivim koracima" koristeći programske tehnologije za Android operacijski sustav

Perković, Brigita

VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstvo

BRIGITA PERKOVIĆ

IZRADA APLIKACIJE „VIROVITICA NEVIDLJIVIM KORACIMA“ KORISTEĆI PROGRAMSKE TEHNOLOGIJE ZA ANDROID OPERACIJSKI SUSTAV

ZAVRŠNI RAD

VIROVITICA, 2023.

VELEUČILIŠTE U VIROVITICI

Preddiplomski stručni studij Računarstvo

IZRADA APLIKACIJE „VIROVITICA NEVIDLJIVIM KORACIMA“ KORISTEĆI PROGRAMSKE TEHNOLOGIJE ZA ANDROID OPERACIJSKI SUSTAV

ZAVRŠNI RAD

Predmet: Programiranje mobilnih aplikacija

Mentor: Student:

Ivan Heđi, dipl.ing., v.pred. Brigita Perković

VIROVITICA, 2023.

Veleučilište u Virovitici

Stručni prijediplomski studij Računarstva - Smjer Programsko inženjerstvo

OBRAZAC 1b

ZADATAK ZAVRŠNOG RADA

Student/ica: **BRIGITA PERKOVIĆ** **JMBAG:** **0307016455**

Imenovani mentor: Ivan Heđi, dipl. ing., v. pred.

Imenovani komentor: -

Naslov rada:

Izrada aplikacije „Virovitica nevidljivim koracima" koristeći programske tehnologije za Android operacijski sustav

Puni tekst zadatka završnog rada:

U vašem radu opišite Android operacijski sustav i programski jezik Java za izradu aplikacija. Kao praktični primjer implementacije teorijskog dijele osmislite i izradite aplikaciju pomoću koje osobe s invaliditetom mogu jednostavno pronaći razne informacije o javnim objektima koje inače ne mogu naći na službenim web stranicama. Prije izrade same aplikacije osmisliti koji bi to skupovi podataka bili od bitnog značaja za osobe s invaliditetom. Za slanje obavijesti korisnicima implementirajte sustav Firebase Cloud Messaging. Za bazu podataka koristiti Firebase. Osim programskog rješenja, u pisanom dijelu završnog rada opišite ukratko korištene tehnologije, opišite detaljno arhitekturu sustava te opišite detaljno odabrane procese i/ili funkcije unutar same aplikacije. Prilikom opisivanja, osim neformalnih koristite i neke od formalnih metoda koje poznajete.

Datum uručenja zadatka studentu/ici: 31.07.2023.
Rok za predaju gotovog rada: 08.09.2023.

Mentor:

Ivan Heđi, dipl. ing., v. pred.

Dostaviti:
1. Studentu/ici
2. Povjerenstvu za završni rad - tajniku

IZRADA APLIKACIJE „VIROVITICA NEVIDLJIVIM KORACIMA“ KORISTEĆI TEHNOLOGIJE ZA ANDROID OPERACIJSKI SUSTAV

Sažetak

Cilj ovog rada bila je izrada mobilne aplikacije koristeći tehnologije za Android operacijski sustav pomoću koje osobe s invaliditetom mogu jednostavno i brzo pronaći informacije o prilagođenosti javnih objekata na području grada Virovitice. Aplikacija je pisana programskim jezikom Java u razvojnom okruženju Android Studio koje je namijenjeno za razvoj Android aplikacija, dok je za bazu podataka korištena Firebase platforma. Ova aplikacija osobama s invaliditetom, odnosno korisnicima aplikacije pruža informacije o objektima koje inače ne mogu pronaći na službenim web stranicama, a za njih su od krucijalne važnosti. Informacije koje se korisnicima pružaju su slika ulaza u objekt, prikaz lokacije na Google Maps platformi za web karte, adresa, kontaktne informacije, web adresa, opis ulaza te unutrašnjost objekta. Opis ulaza u objekt sadrži informacije o pristupačnosti prilikom ulaska koje su vidljive na slici ulaza, širina vrata, vrsta vrata, dok unutrašnjost objekta sadrži informacije o pragovima, širini vrata te dodatne informacije kao što su informacije o postojanju lifta, sanitarnog čvora prilagođenog osobama s invaliditetom i slično. Informacije o objektu u aplikaciji korisnik može pronaći na dva načina, a oni su odabirom kategorije u izborniku kategorija te potkategorije objekta ili pomoću tražilice koja se također nalazi u izborniku kategorija. Osim pregleda informacija o objektu, korisniku se pruža mogućnost samoinicijativnog unosa informacija o objektu za koji ne postoje informacije u aplikaciji, a korisnik posjeduje informacije o istom.

Ključne riječi: *Android operacijski sustav, Android Studio, Firebase, Java, mobilma aplikacija*

CREATION OF THE APPLICATION „VIROVITICA WITH INVISIBLE STEPS“ USING TECHNOLOGIES FOR THE ANDROID OPERATING SYSTEM

Abstract

Goal of this final work was the creation of mobile application using technologies for Android operation system, with which people with disabilities are able to find information about public facilities and their accesaibility in the area of the town Virovitica. Application is written with Java programming language in Android Studio IDE, while Firebase platform was used as database. Users of this application have possibility to get information about facilities that can't be find on official websites, but are of crucial importance to them. Information provided to users are pictures of entrance to facility, location on Google Maps, address, contact data, web address, description of entrance and facility interior. Entrance description has information about facility accessibility when entering, that are visible in entrance picture, such as door width, door type. Facility interior has information about thresholds, door width and additional information like the existance of an elevator, sanitary facilities adapted for people with disabilities and so on. Those information are available in two ways in application, thorugh choosing category in category menu and then choosing subcategory of facility, or with the help of search bar that is also visible in category menu. Except of facility information, user, in his own initiative, is able to enter facility information that are not existing in application, but user has knowledge o fit due to experience.

Keywords: *Android operating system, Android Studio, Firebase, Java, Mobile application*

Sadržaj

1. Uvod

U današnje vrijeme sve više se koriste mobilne aplikacije koje su postale popularnije od web aplikacija i desktop aplikacija radi svoje dostupnosti. Svaka osoba uz sebe ima mobilni uređaj te im on omogućava pristup svim informacijama, a informacije se dobivaju putem aplikacija. Želja je bila osobama s invaliditetom omogućiti dostupnost informacija o prilagođenost objekata koji trebaju posjetiti te im na taj način omogućiti da se mogu unaprijed pripremiti za određene prepreke prilikom ulaza u objekt, a i unutar objekta. U ovome radu govorit će se o programskom jeziku Java koji služi za razvoj mobilnih aplikacija, ali također ima i širu primjenu, može se koristiti i za razvoj web aplikacija, desktop aplikacija, Internet of things i mnoge druge probleme. U radu će biti objašnjeno što je Firebase spremište podataka, android operacijski sustav te prikazati naše rješenje za aplikaciju o kojoj se govorilo ranije u radu.

2. Programske tehnologije i alati

2.1. Android Studio

Android Studio je razvojno okruženje (IDE) namijenjeno za razvoj Android aplikacija koje je temeljeno na moćnom uređivaču koda i razvojnih alata tvrtke IntelliJ IDEA. Osim IntelliJ Android Studio nudi značajke koje povećavaju produktivnost pri izradi Android aplikacija, kao što su:

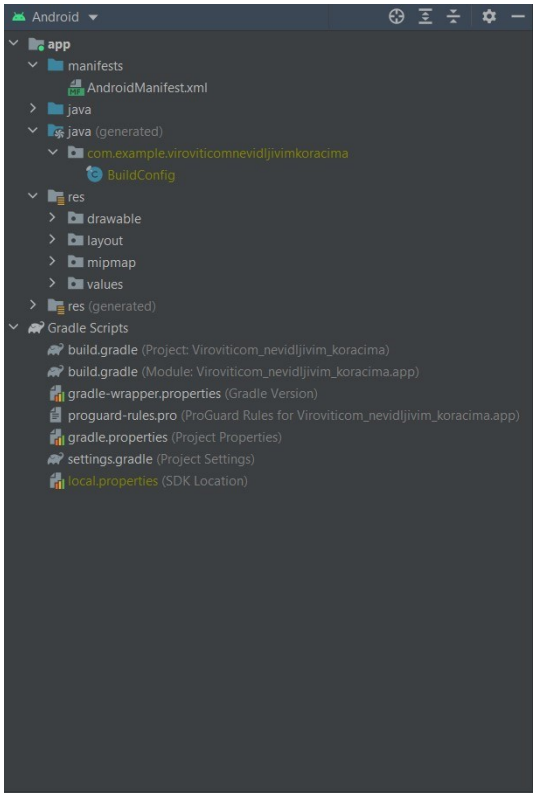
- Fleksibilan sustava izrade temeljen na Gradle-u
- Brz emulator bogat značajkama
- Jedinstveno okruženje u kojem možete razvijati za sve Android uređaje
- Predlošci koda i GitHub integracija koje će vam pomoći da izgradite zajedničke značajke aplikacije
- Opsežni alati i okviri za testiranje
- Live Edit za ažuriranje komponenata u emulatorima i fizičkim uređajima u stvarnom vremenu
- Lint alati za hvatanje performansi, kompatibilnosti verzija, upotrebljivosti i drugih problema
- C++ i NDK podrška
- Ugrađena podrška za Google Cloud Platform, koja olakšava integraciju Google Cloud Messaging i App Engine [1]

Sučelje Android Studia podijeljeno je u tri cjeline:

- Upravitelj datoteka: Nalazi se na desnoj strani Android Studia te omogućava lakšu navigaciju između datoteka koje se nalaze u projektu.
- Editor koda: Nalazi se u središnjem dijelu Android Studia te podržava naprednije funkcionalnost poput:
    - Code completion: Android Studio prati unos koda te predlaže nadopunu. Primjerice, poziva funkcije s njezinim parametrima.

    - Generator koda: Automatski generira predefinirane funkcije i mijenja funkcije sa Lambda izrazima.
    - Lint: Android Studio za izvršavanje provjere sintakse dok korisnik piše posjeduje napredni lint. Također, koristi ga za generiranje upozorenja prilikom pisanja koda.
- Upravitelj konzole: Nalazi se na dnu Android Studia, a služi kao multi izbornik konzola pomoću kojih je moguće pratiti rad aplikacije. Unutar upravitelja konzole nalaze se kontrole poput:
    - Terminal : Korisniku omogućava upravljanje datotekama unutar projekta i uređajem pomoću ADB-a (engl. *Android Debug Bridge*) alata naredbenog retka koji omogućava komunikaciju s uređajem.
    - Build: Korisniku omogućava praćenje procesa izrade projekta.
    - Logcat: Korisniku omogućava praćenje aplikacija za vrijeme njihovog izvršavanja na uređaju.
    - Event log: sadrži informacije zadacima Android Studia te izvještaje istih.

Struktura projekta u Android Studiu sastoji se komponenti. Slika 1 prikazuje osnovne datoteke koje nastaju prilikom kreiranja novog projekta. Prilikom kreiranja novog stvara se manifest datoteka, osnovna aktivnost, pomoćni resursi – slika aplikacije, layout osnovne aktivnosti, prijevodi aplikacije te gradle build skripte [2].



Slika 1. Primjer datoteka unutar Android projekta

2.2. Programski jezik Java

Razvoj programskog jezika Java inicijalno je započeo 1991. godine u svrhu programiranja elektroničkih čipova ugrađenih u kućanske aparate i uređaje posebne namjene [10]. Java je jezik otvorenog koda. Dizajnirao ga je James Gosling 1995. godine za vrijeme svoga rada u tvrtci Sun Mycrosistems. Njegov cilj je bio dizajnirati novi objektno orijentirani jezik koji će biti neovisan o platformi [11]. Objektno orijentiran jezik ima konstrukcije za predstavljanje objekata iz stvarnog svijeta, dok neovisnost o platformi omogućuje da se programi dizajnirani u programskom jeziku Java mogu se bez preinaka izvoditi na operacijskim sustavima za koje postoji JVM (engl. *Java Virtual Machine*) dok je primjerice standardne programe potrebno prilagođavati operacijskom sustavu, što je prednost u odnosu na druge programske jezike [12].

Redizajniranjem programskog jezika Java 2005. godine u jezik za razvoj web aplikacija Java dobija današnje ime, a potom 2010. godine Java postaje vlasništvo tvrtke Oracle čiji

razvojni tim programera nastavlja raditi na ideji da postane više-platformski programski jezik. To je rezultiralo da postane programski jezik koji se koristi za izradu aplikacija za web, mobilnih aplikacija, samostalnih desktop aplikacija te softvera ugrađenog u uređaje. Također, programi pisani u programskom jeziku Java mogu se izvršavati na većini operacijskih sustava. Razlog tome je što je jezik temeljen na jednostavnosti, distribuiranosti, robusnosti, objektno orijentiranosti, sigurnosti, portabilnosti, visokoj učinkovitosti, višedretvenosti, neovisnosti o arhitekturi i dinamičnosti. [10].

Ovu prednost, dostupnost i veliku popularnost programskog jezika Java prepoznali su kreatori Androida te su ga prigrlili kao osnovni programski jezik za kreiranje Android aplikacija [10].

Postoje dvije vrste programskih jezika: interpreterski i kompajlerski. Interpreterski jezici se izvršavaju liniju po liniju te stanu ukoliko dođe do greške ili se izvrše do kraja. Primjer interpreterskog jezika je JavaScript. Programski jezik Java je kompajlerski jezik što znači da se kod prije izvršavanja mora cijeli kompajlirati te se tek onda izvršava. Kada zakompajliramo Java kod tada dobivamo bajt kod datoteke s nastavkom *class* što znači da je kod kompajliran te ga tada možemo izvršiti na bilo kojoj JVM [12].

Java sadrži osam tipova podataka od toga su četiri za cjelobrojne vrijednosti (byte, short, int i long), jedan za čuvanje pojedinačnih znakova (char i string), dva za vrijednosti s decimalnom točkom (float i double) te jedan za logički tip podataka koji mogu biti točno (engl. true) ili netočno (engl. false) (boolean) [12].

Prva verzija Jave Java 1.0 ili JDK 1.0 je objavljena u siječnju 1996. godine. Razvojni programeri su više od 27 godina aktivno razvijali platformu te su sa svakom novom verzijom donijeli mnoge promjene i poboljšanja u tehnologiji tijekom godina. Trenutna verzija Jave je Java 20 ili JDK 20 koja je objavljena 21. ožujka 2023. godine.

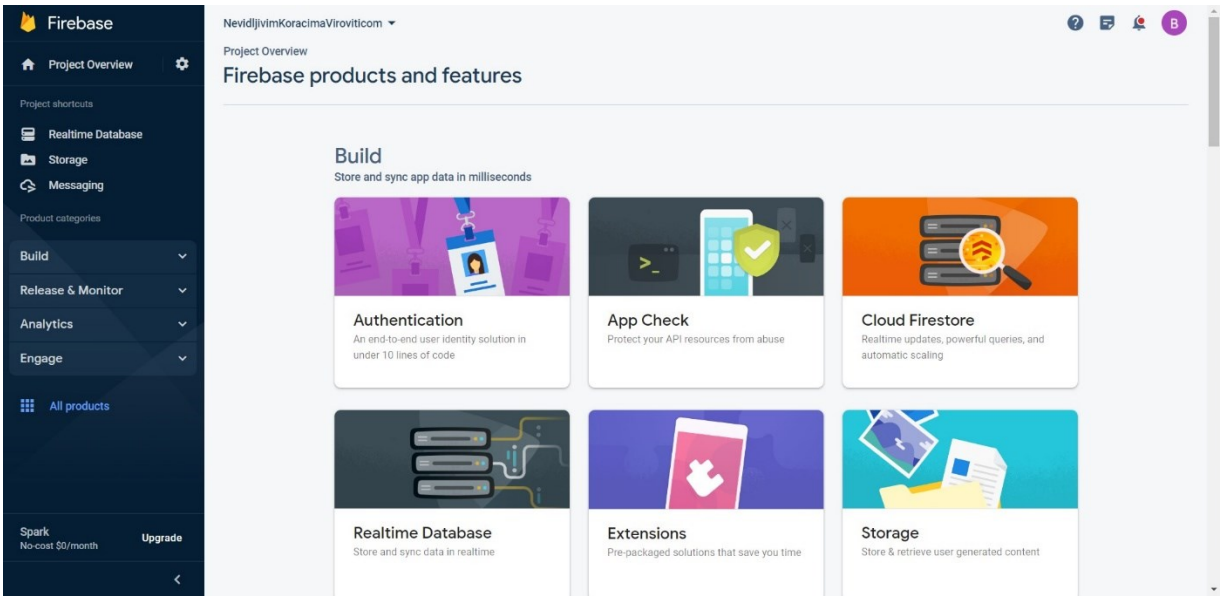
Verzije tijekom godina:

- Verzije 1.0 i 1.1 nazivaju se JDK (engl. *Java Development Kit*)
- Verzije od 1.2 do 5.0 nazivaju se J2SE (engl. Java 2 Standard Edition)
- Verzije od Jave 6 nazivaju se Java SE X [13].

2.3. Firebase

Firebase je razvijen iz Envolvea, prethodne tvrtke koju su 2011. godine osnovali James Tamplin i Andrew Lee. Envolve je programerima ponudio API koji omogućuje integraciju funkcije online chata na njihove web stranice. Nakon što je usluga puštena, James Tamplin i Andrew Lee su otkrili da se ju razvojni programeri koriste za prosljeđivanje podataka koji nisu chat poruke. Programeri su Envolve koristili u svrhu sinkroniziranja aplikacijskih podataka u stvarnom vremenu. Uvidjevši to ovaj dvojac je odlučio razdvojiti sustav za chat od arhitekture u stvarnom vremenu koja je to omogućavala te je uslijedilo osnivanje Firebase-a kao zasebne tvrtke koja je javnosti predstavljena 2012. godine. Google je Firebase kupio 2014. godine [14].

Firebase je razvojna platforma za aplikacije. Platforma nudi razne servise koji omogućuju razvijanje, nadogradnju i održavanje aplikacija poput analitike podataka o korištenju aplikacije, baze podataka u stvarnom, autentifikacije korisnika prilikom ulaska u aplikaciju vremenu, poruke u oblaku, skladištenje u oblaku i razne druge [15]. Servisima administriramo pomoću Firebase konzole koja je prikazana na slici 2. Konzola nam omogućuje dodavanje i pregled novih servisa.



Slika 2. Firebase konzola

U aplikaciji korišteni su sljedeći servisi:

- Baza podataka u stvarnom vremenu (engl. Realtime Database) je baza podataka koja je smještena u oblaku. Podaci se pohranjuju kao JSON i sinkroniziraju u stvarnom vremenu sa svakim povezanim klijentom.
- Poruke u oblacima (engl. Cloud Messaging) služe za razmjenu poruka i slanje push obavijesti.
- Skladištenje u oblaku (engl. Cloud Storage) služi za pohranu sadržaja kao što su fotografije, videozapisi te ostale datoteke na oblaku [15].

3. Android operacijski sustav

Android je sveobuhvatni operacijski sustav (OS) otvorenog koda dizajniran za mobilne uređaje izgrađen na modificiranoj verziji Linuxa koji je razvio Google. S obzirom da je Android operacijski sustav modularan i prilagodljiv, te osim što se koristi u pametnim telefonima i tabletima, također se koristi u pametnim satovima, automobilima, čitačima elektroničkih knjiga, televizorima i slično, što je omogučeno korištenjem Dalvik i ART virtualnih strojeva[3][4].

Android koristi i podržava:

- Standarde za komunikaciju i interakciju (Flash, MMS, SMS, Bluetooth, ekran na dodir...)
- Digitalne formate za multimedijske sadržaje (PING, JPEG, GIF, MP3, MP4..)
- Digitalni kompas, GPS, fotoaparat/kamera...
- WebKit preglednik
- SQLite relacijsku bazu podataka (brza je i efikasna, što je od iznimne važnosti za uređaje čiji su kapaciteti limitirani) [3].

3.1. Dalvik

Dalvik virtualni stroj (engl. *Dalvik virtual machine*) je licencirani virtualni stroj kreirana za Android i prilagođen za mobilne uređaje s ograničenom memorijom i procesorskim kapacitetom (CPU-om). Ime je dobio po ribarskom selu Dalvik koje se nalazi na Islandu od kuda potječu preci tvorca Dalvika, Dana Bornsteina. Osim Dana Bornsteina na njemu je radio i tim iz Googlea[3][4].

Android aplikacije pišu se u programskom jeziku Java, ali se ne izvršavaju u klasičnom Java virtualnom stroju (engl. *stack virtual based virtual machine*), već u Dalvik virtualnom stroju (engl. *register based virtual machine*). Kompaijlirane (engl. *compiled*) Java klase se pretvaraju u Dalvik izvršne datoteke (engl. *executables*) .dex i izvršavaju kao poseban proces u vlasitoj Dalvik instanci, što omogućuje da ako se jedna aplikacija sruši da nema utjecaj na druge aplikacije. Cjelokupan pristup hardveru i nižim (engl. *low level*) sistemskim funkcionalnostima

omogućuje Dalvika, koji ujedno osigurava razvojnim programerima da ne moraju brinuti o tome za koji uređaj razvijaju aplikaciju [4].

3.2. ART

Dalvikov virtualni stroj za Android nasljednik je ART (engl. *Android Run Time*). On kao i Dalvik izvršava .dex datoteke, samim time kodu koji je napisan za Dalvik omogućeno je da može raditi pod ART-om. No, pojedine tehnologije koje su radile za Dalvik ne rade na ART-u. Postoje i neki benefiti ART-a nad Dalvikom poput Ahead Of Time compilation (AOT) koji služi za ubrzavanje performansi neke aplikacije, poboljšani Garbage Collector (GC), te su jasnije vidljive greške u aplikaciji.

ART još uvijek u potpunosti nije zamijenio Dalvik, jer prijelaz još nije načinjen u potpunosti, ali polako ga zamjenjuje. Android verzije 5.0 kao početnu postavku koriste Dalvik što ne neomogućava prelazak na ART. Osim verzije 5.0 Dalvik kao temeljni virtualni stroj koristi također i verzija 6.0. ART dolazi u prvi plan počevši s 7.0 verzijom [4].

3.3. Optimizirana memorija i proces upravljanja

Osim Androida *run-time* i virtualni stroj koriste .NET i Java. Međutim, razlika je u tome što Android Run Time nadzire trajanje procesa. Kako bi osigurao resurse za aplikacije s većim prioritetom, Android može, prema potrebi, zaustaviti i završiti procese. Interakcija korisnika s određenom aplikacijom utječe na njezin prioritet. Sve aplikacije trebaju biti spremne na mogućnost prekida, ali isto tako trebaju biti postavljene da se mogu brzo pokrenuti u prvobitno stanje [4].

3.4. Povijest Android operacijskog sustava

Android Inc. osnovan je 2003. godine u Kaliforniji. Njegovi osnivači su bili Chris White, Rich Miner, Andy Rubin i Nick Sears. Android operacijski sustav je u svojim počecima bio namijenjen pametnim digitalnim fotoaparatima i kamerama, no ubrzo je Andy Rubin uvidio potencijal i prilagodio ga pametnim telefonima. Od 2005. godine u vlasništvu je Google-a [5].

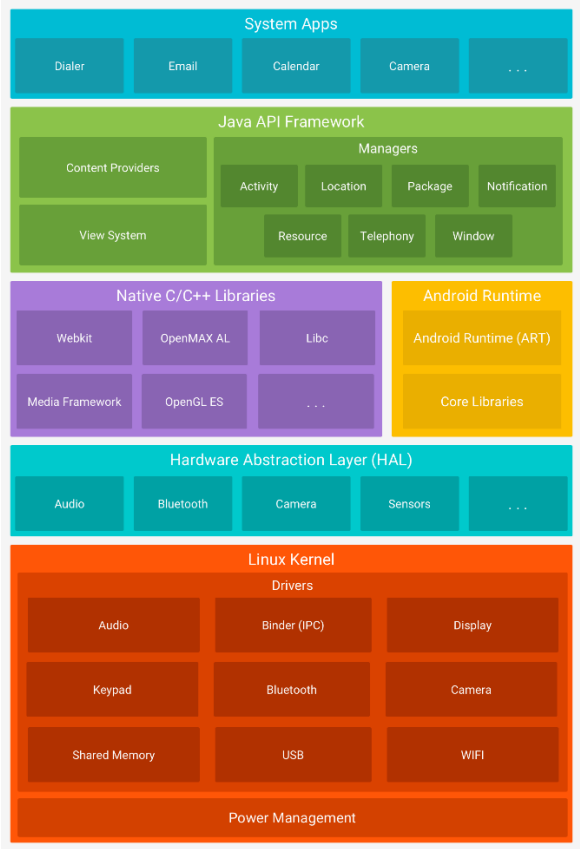
Trenutna verzija Google-a Android operacijskog sustava je Android 13 koja je objavljena 15. kolovoza 2022. godine. Androidove verzije su poznate po nazivima „poslastica“ te su ti nazivi verzija bili poznati javnosti. S verzijom Android 10 prestaje ova tradicija. Interni kodni naziv te verzije je „Quince Tart“, dok trenutne verzije je „Tiramisu“ [6]. Najnovija verzija Android 14 javnosti bi trebala biti dostupna u trećem kvartalu 2023. godine [7].

3.5. Android arhitektura

Android operacijski sustav je sustav koji koristi softverski dizajn u slojevima. Arhitektura se sastoji od četiri sloja te aplikacijskog sloja koji se nalazi na vrhu arhitekture. Aplikacije se pokreću preko virtualnog stroja. Sistemske komponente su napisane u Java, C/C++ i XML programskim jezicima. Slika 3 prikazuje dijagram Android arhitekture.

Slojevi Android arhitekture su:

- Aplikacijski sloj (engl. *System Apps*)
- Java razvojno okruženje (engl. *Java API Framework*)
- Native C/C++ biblioteke (engl. *Native C/C++ Libraries*)
- Izvršavanje Android aplikacije (engl. *Android Runtime*)
- Hardverski sloj aplikacije (engl. *Hardware Abstraction Layer (HAL)*)
- Linux jezgra (engl. *Linux Kernel*) [8].



Slika 3. Android arhitektura [8]

3.5.1. Aplikacijski sloj

Aplikacijski sloj je najviši sloj Android arhitekture koji sadrži neke od osnovnih aplikacija koje uključuju kalendar, web preglednik, kontakte, e-mail poštu, poruke i slične aplikacije. Aplikacije koje se nalaze u ovom sloju napisane su u Java programskom jeziku [8]. Također i sve ostale aplikacije koje korisnik samoincijativno odluči instalirati na svoj uređaj nalaze se u ovom sloju. Primjerice, korisnik nije primoran koristiti zadani preglednik, tipkovnicu ili aplikaciju za SMS poruke, već ih može promijeniti. Jedina aplikacija koju korisnik ne može promijeniti je „Postavke sustava“[9].

3.5.2. Java razvojno okruženje

Java razvojno okruženje je sloj koji sadrži servise kojima se pristupa preko API-ja napisanih u Java programskom jeziku. API-ji su dostupni razvojnim programerima te tvore blokove od kojih je Android aplikacija izgrađena te na taj način pojednostavljuje kod i dijele sistemske komponente i usluge u zasebne module, a oni uključuju [8]:

- View sustav (engl. *View system*): koristi se za izradu i interakciju korisničkog sučelja aplikacije [8].
- Upravitelj obavijestima (engl. *Notification Manager*): pruža mogućnost prikaza obavijesti na statusnoj traci aplikacije [8].
- Upravitelj resursima (engl. *Resource Manager*): omogućuje pristup datotekama koje se nalaze unutar aplikacije [8].
- Pružatelji sadržaja (engl. *Content providers*): omogućuje aplikacijama pristupanje podacima iz ostalih aplikacija, primjerice aplikaciji „Kontakti“ [8].
- Upravitelj aktivnostima (engl. *Activity Manager*): upravlja životnim ciklusom aplikacije [8].

3.5.3. Native C/C++ biblioteke i Izvršavanje Android aplikacije

Sloj ispod javinog razvojnog okruženja se sastoji od dva dijela. On se sastoji od Native C/C++ biblioteke i Izvršavanje Android aplikacije.

Native C/C++ sloj sadrži izvorne biblioteke koje su napisane u programskom jeziku C/C++. Pozivaju se preko Java sučelja te omogućavaju većinu funkcija koje su dostupne u osnovnim bibliotekama programskog jezika Java [8].

Izvršavanje Android aplikacije je sloj koji se sastoji od Core Libraries i Android Runtime (ART) virtualnog stroja o kojem se govorilo nešto ranije u radu.

3.5.4. Linux jezgra

Linux jezgra je najniži sloj Android arhitekture na kojoj se temelje ostali slojevi, odnosno služi kao sloj apstrakcije između sklopovlja i ostalih slojeva programske podrške, a

pruža programsku potporu i upravlja sklopovljem. Android operacijski sustav Linux jezgru koristi za ostvarivanje temeljnih funkcionalnosti sustava poput upravljanja memorijom, upravljanja procesima, mrežne komunikacije i slično [8].

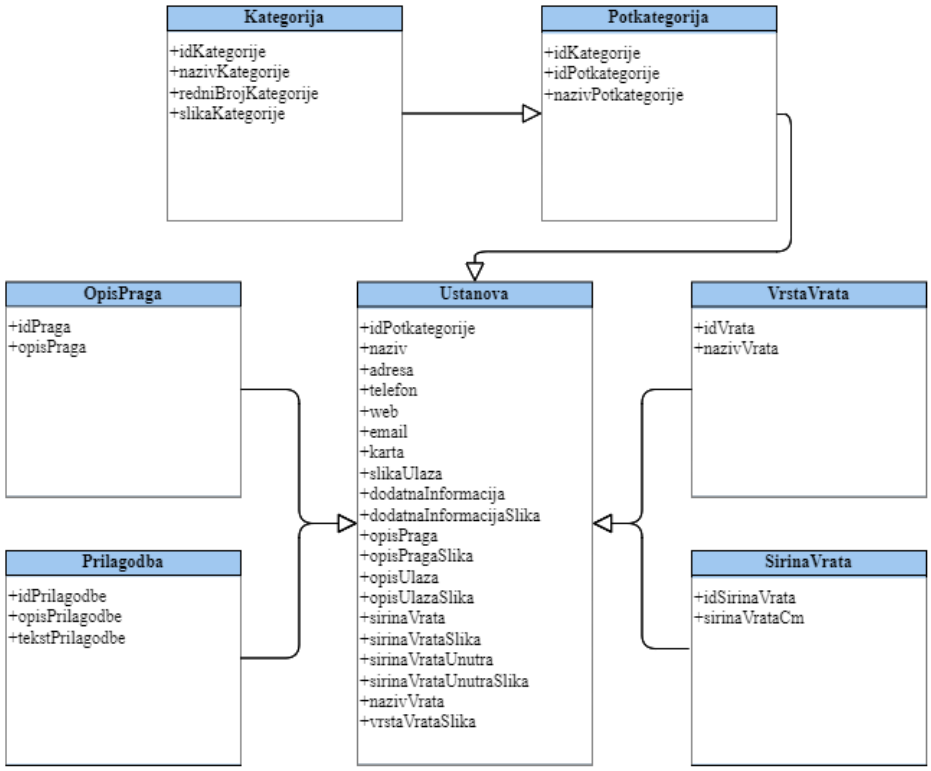
3. Programsko rješenje

Osobama s invaliditetom ponekad je teško pronaći podatke o pristupačnosti objektima koje zbog poslovne ili privatne potrebe trebaju posjetiti. Ova aplikacija im upravo to omogućava. Podaci su spremljeni u Realtime Database bazi podataka koja je dio Firebase servisa.

4.1. Baza podataka

Realtime Database je NoSQL baza podataka koja se nalazi u oblaku te sinkronizira podatke u stvarnom vremenu. Podaci se pohranjuju u obliku JSON datoteke i sinkroniziraju u stvarnom vremenu sa svakim klijenskom aplikacijom. Može se zaključiti da klijenti imaju ažurne podatke koji su upisani pomoću aplikacije koju je programer izradio.

Baza podataka koja se koristi u aplikaciji sastoji se od sedam tablica kategorija, potkategorija, ustanova, opis praga, prilagodba, vrsta vrata i širina vrata – slika 4. Za pristup bazi koristi se Firebase Realtime Database. NoSQL ne koristi relacije za povezivanje već dokumente u čijim poljima se navode identifikatori (id) prema kojima se pregledavaju dokumenti. Primjerice, kako bi se dohvatile potkategorije koje pripadaju u određenu kategoriju, potrebno je prvo dohvatiti kategoriju i njezin id, a potom dohvaćamo tablicu s potkategorijama i provjeravamo dokumente sadrže li u sebi određeni id kategorije.



Slika 4. Shema baze podataka korištene u aplikaciji

4.2. Nerelacijska baza podataka

Nerelacijske baze podataka su NoSQL baze podataka koje su pogodne za web aplikacije, mobilne i igrica što od baze podataka zahtjeva stabilnost, fleksibilnost, moćnost i visoku funkcionalnost da bi se korisnicima pružilo odlično korisničko iskustvo [16].

Razlozi zašto bi se trebala koristiti nerelacijska baza podataka:

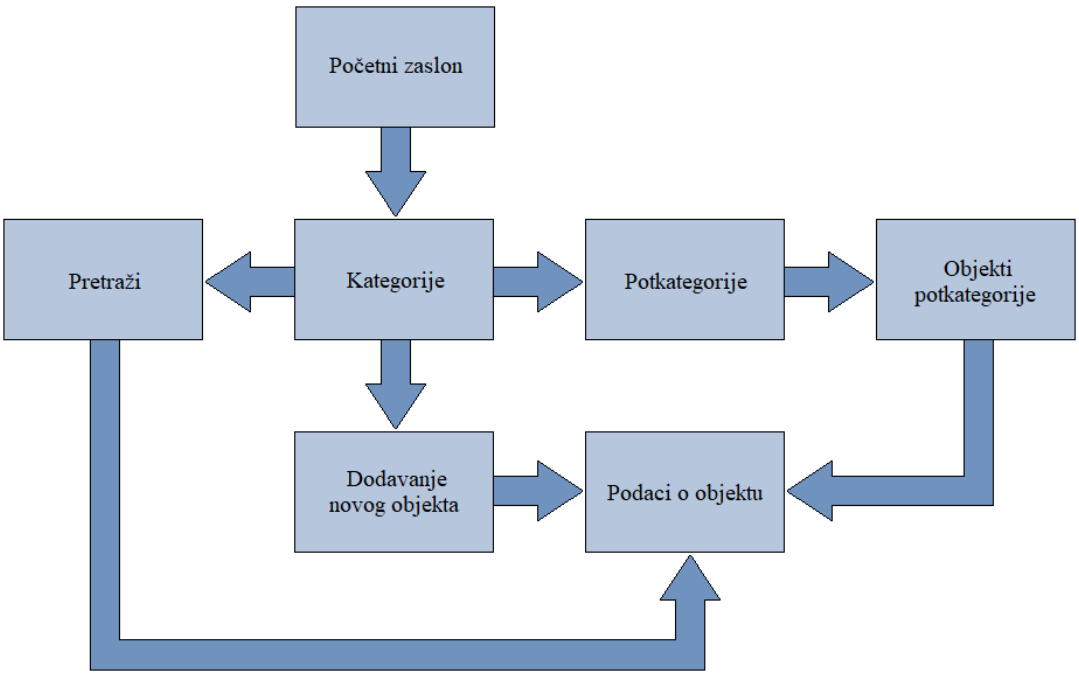
- Skalabilnost: NoSQL baza podataka je dizajnirana za skaliranje na distribuiranim hardverskim klasterima[16].
- Fleksibilnost: NoSQL baza podataka pruža fleksibilni raspored koji omogućuje brz i iterativni napredak. Fleksibilnost modela podataka NoSQL bazu podataka čini pogodnim rješenjem za nestrukturirane i polustrukturirane podatke [16].

- Snažne performanse: NoSQL baza podataka prilagođena je određenim podatkovnim modelima, kao što su primjerice dokumenti [16].
- Visoko funkcionalno: NoSQL baza podataka pruža API-je i tipove podataka posebno izgrađene za pripadajući model podataka [16].

4.3. Ideja i model

Ideja je bila izraditi mobilnu aplikaciju pomoću koje osobe s invaliditetom mogu na jednostavan način pronaći razne informacije o javnim objektima koje inače ne mogu pronaći na službenim web stranicama objekata.

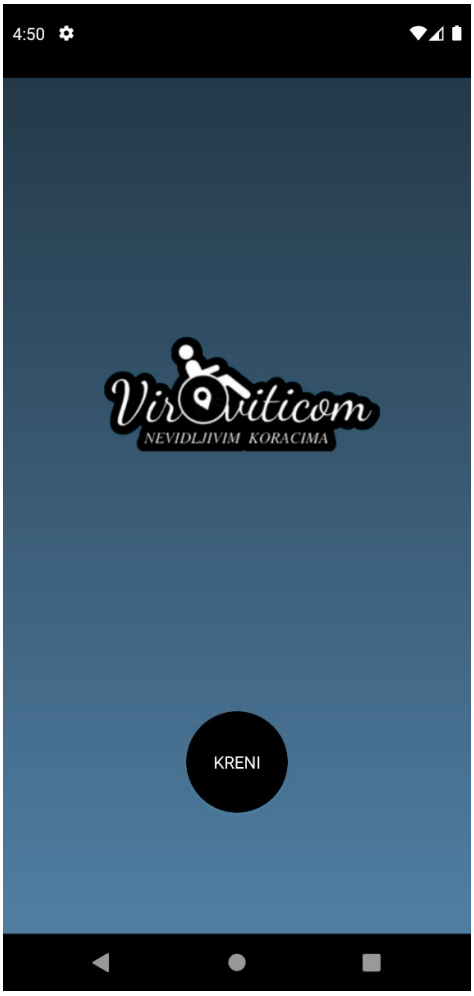
Prije izrade aplikacije važno je osmisliti model kako će aplikacija izgledati. Pomoću dijagrama će biti prikazan slijedni prikaz odvijanja aplikacije – slika 5. Prvi dio aplikacije je početni zaslon koji korisniku omogućuje ulazak u sučelje koje se sastoji od izbornika kategorija, pretraživača i mogućnosti dodavanja novog objekta. U slučaju da korisnik odabere kategoriju otvoriti će mu se sučelje potkategorija. Sučelje potkategorija korisniku će prikazati potkategorije objekata. Odabirom željene potkategorije korisniku se otvara sučelje s objektima potkategorije. Kada korisnik odabere objekt otvoriti će mu se sučelje s podacima objekta. U slučaju da korisnik u sučelju izbornika odabere *Pretraži* otvoriti će mu se sučelje s tražilicom i popisom svih objekata koji se nalaze u bazi podataka. Ovo sučelje korisniku omogućuje da na brz i efikasan način provjeri jesu li u bazi podataka upisani podaci o željenom objektu, ako jesu odabirom na objekt otvara se sučelje s podacima objekta. Na ovaj način korisniku se olakšava pronalazak informacija, ako nije siguran u koju kategoriju, a potom potkategoriju objekt pripada. Ako prilikom pretraživanja objekata korisnik aplikacije nije pronašao objekt te samim time niti podatke o istome, a posjeduje informacije, povratkom na sučelje s izbornikom pruža mu se mogućnost dodavanja novog objekta. Upisom, a potom spremanjem podataka u sučelju za dodavanje novog objekta, korisnik će dobiti povratnu informaciju o objektu čije je podatke upisao.



Slika 5. Model android aplikacije

4.3.1. Početni zaslon

Aplikacija omogućuje korisnicima s invaliditetom pregled informacija o pristupačnosti objektima u gradu Virovitici. Nakon instalacije aplikacije prikazan je početni zaslon. Zaslon sadrži gumb za ulazak na sljedeći zaslon, kao što je prikazano na slici 6. Klikom na gumb *Kreni* okida se događaj onClick funkcije koja omogućuje korisniku aplikacije prelazak na zaslon s kategorijama objekata – Isječak programskog koda 1.



Slika 6. Prikaz početnog zaslona

Isječak programskog koda 1: Metoda otvoriGlavniActivity()

```
1.  private void otvoriGlavniActivity(){
2.  Intent intent = new Intent( packageContext: this, IzbornikActivity.class);
3.  startActivity(intent);
4.  }
```

4.3.2. Zaslon s kategorijama

Nakon okidanja gumba *Kreni* prikazuje se zaslon s kategorijama, kao što je prikazano na slici 7. Zaslon se sastoji od izbornika kategorija unutar kojeg se nalazi mogućnost pretraživanja objekata klikom na *Pretraži*, osim što zaslon sadrži izbornik sadrži i gumb koji se nalazi u desnom donjem kutu, a korisniku aplikacije omogućava dodavanje novog objekta.

Izbornik kategorija sastoji se od osam kategorija, a one su *Obrazovanje, Zdravstvo, Socijalna skrb, Uprava, Kultura, Sport, Usluge i Ugostiteljstvo.*

Slika 7. Prikaz zaslona s kategorijama

Isječak programskog koda 2 prikazuje postavljanje podataka iz izbornika koji su dohvaćeni iz baze podataka u izbornik od tri stupca pomoću GridLayoutManagera koji je od RecyclerView.LayoutManager implementacija za postavljanje stavki u rešetke.

Isječak programskog koda 2: Postavljanje izbornika u rešetke

```
1.  GridLayoutManager gridLayoutManager = new GridLayoutManager(context: this, spanCount: 3);
2.  recyclerView.setLayoutManager(gridLayoutManager);
```

4.3.3. Zaslon s potkategorijama

Klikom na kategoriju u izborniku kategorija otvara se zaslon s potkategorijama. Zaslon omogućuje pregled potkategorija objekata koje su nalaze u odabranoj kategoriji. Osim izbornika potkategorija na zaslonu u donjem desnom kutu se nalazi gumb koji korisniku omogućuje povratak na prethodni zaslon, odnosno na zaslon izbornika s kategorijama.

Kao primjer dohvaćanja zaslona potkategorije na zaslonu s izbornikom kategorija prikazana je slika 8 koja prikazuje zaslon s potkategorijama kategorije obrazovanje. Potkategorija obrazovanje sadrži potkategorije *Dječji vrtići, Fakulteti, Osnovne škole, Srednje škole i Učilišta.*

Slika 8. Prikaz zaslona s potkategorijama kategorije obrazovanje

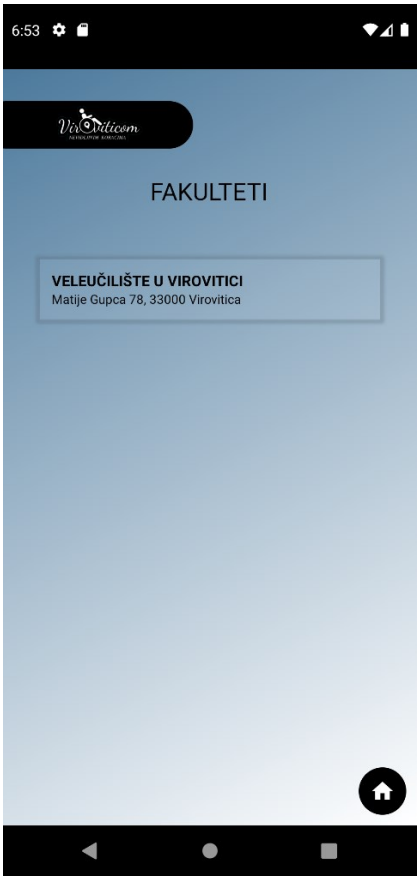
Podaci potkategorija dohvaćaju se pomoću Realtime Database baze podataka. Kako bi se u određenu kategoriju dohvatila određena potkategorija potrebno je koristiti Bundle za vraćanje vrijednosti pridružene id vrijednosti. Odnosno, ako imamo više podataka koji se prosljeđuju dohvatimo ih u jedan objekt u Bundle. Isječak programskog koda 3 prikazuje dohvaćanje potkategorije određene kategorije iz baze podataka.

Isječak programskog koda 3: Dohvaćanje podataka iz baze podataka

```
1.  String kategorija = getIntent().getExtras().getSerializable(key:
    “idKategorije”).toString();
2.  databaseReference = FirebaseDatabse.getInstance().getReference().child(“potkategorija”;
3.  recyclerView.setHasFixedSize(true);
4.  recyclerView.setLayoutManager(new LinearLayoutManager(context: this));
5.  potkategorijaArrayList = new ArrayList<>();
6.  potkategorijaArrayListPom = new ArrayList<>();
7.  adapterPotkategorija = new AdapterPotkategorija(context: this, potkategorijaArrayList);
8.  recyclerView.setAdapter(adapterPotkategorija);
9.  databaseReference.assValueEventListener(new ValueEventListener(){
10.     @Override
11.     public void onDataChange(@NonNull DataSnapshot snapshot){
12.         for(DataSnapshot dataSnapshot : snapshot.getChildren())
13.             Potkategorija potkategorija = dataSnapshot.getValue(Potkategorija.class);
14.             potkategorijaArrayListPom.add(potkategorija);
15.         }
16.         for(Potkategorija pot : potkategorijaArrayListPom){
17.             if(pot.idKategorije.equals(kategorija)){
18.                 potkategorijaArrayList.add(pot);
19.             }
20.         }
21.         adapterPotkategorija.notifyDataSetChanged():
22.      }
23.      @Override
24.      public void onCancelled(@NonNull DatabaseError error){
25. });
```

4.3.4. Zaslon s objektima potkategorije

Odabirom potkategorije na zaslonu s potkategorijama korisniku se otvara zaslon s objektima potkategorije. Slika 9 prikazuje zaslon s objektima potkategorije *Fakulteti* čija je kategorija *Obrazovanje*. Objekti zaslona sadrže ime i adresu. U potkategoriji *Fakulteti* možemo pronaći samo jedan objekt s obzirom na to da se na području grada Virovitice nalazi samo jedan registrirani fakultet, Veleučilište u Virovitici. Osim liste objekata koja se nalazi na središnjem dijelu zaslona korisnik u donjem djelu zaslona, odnosno u donjem desnom kutu može pronaći i gumb za povratak na zaslon s izbornikom kategorija.

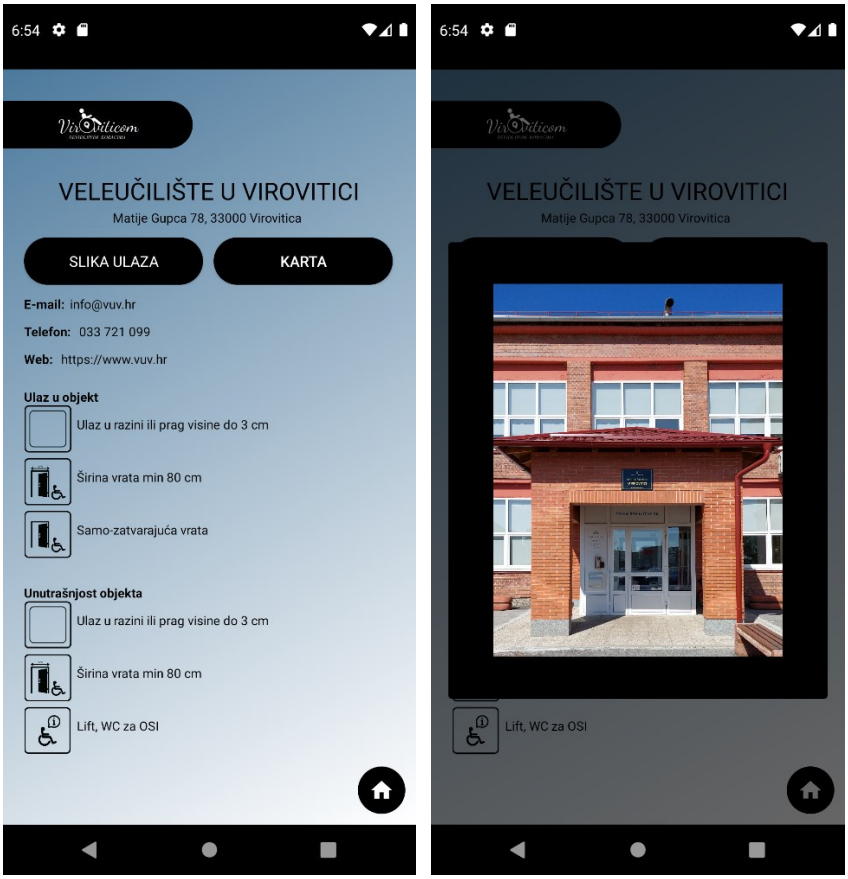


Slika 9. Prikaz zaslona s objektima potkategorije

4.3.5. Zaslon s podacima objekta

Klikom na objekt potkategorije korisniku se prikazuje zaslon sa svim podacima o objektu. Na zaslonu koji je prikazan na slici 10 nalazi se naziv objekta, adresa, kontaktni podaci, podaci o ulazu u objekt, podaci o unutrašnjosti objekta te gumbi za prikaz slike ulaza, karte i povratak na zaslon izbornika s kategorijama.

Kada korisnik klikne na gumb *SLIKA ULAZA* poziva se metoda za prikazivanje slike ulaza objekta u dijalogu.



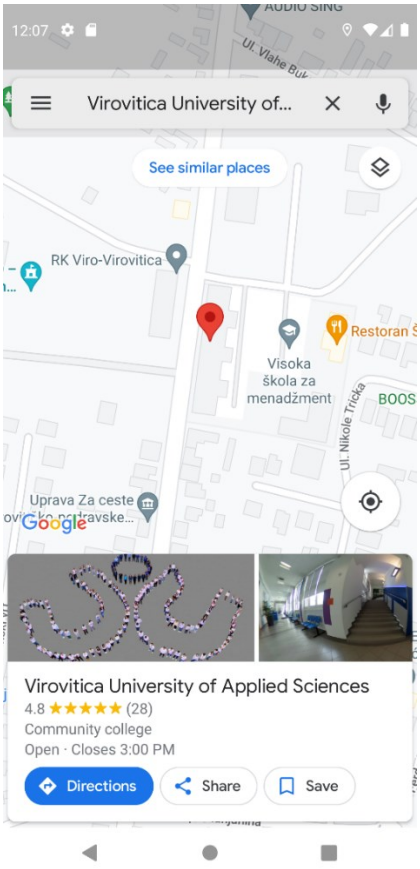
Slika 10. Prikaz zaslona s podacima objekta

Isječak programskog koda 4 prikazuje metodu *prikaziSlikuUlaza()* koja dohvaća podatke u dijalogu, odnosno implementaciju prilagođenog AlertDialog View-a.

Isječak programskog koda 4: Metoda prikaziSlikuUlaza()

```
1.  private void prikaziSlikuUlaza(){
2.  AlertDialog.Builder alertDialog = new AlertDialog.Builder(context: this,
    R.style.MyTransparentDialog);
3.  LayoutInflater = layoutInflater = LayoutInflater.from(this);
4.  View slikaUlazaDialog = layoutInflater.inflater(R.layout.dialog_slika_ulaza, root: null);
5.  ImageView slikaUlaza;
6.  slikaUlaza = (ImageView} slikaUlazaDialog.findViewById(R.id.slikaUlaza);
7.  Picasso.get().load(getIntent().getStringExtra(name: “slikaUlaza”)).into(slikaUlaza);
8.  alertDialog.setView(slikaUlazaDialog);
9.  AlertDialog dialog = alertDialog.create();
10. dialog.show();
11. }
```

Klikom na gumb Karta korisniku se otvara karta s lokacijom odabranog objekta, u ovom slučaju Veleučilišta u Virovitici – slika 11.



Slika 11. Prikaz zaslona lokacije objekta

Isječak programskog koda 5 prikazuje metodu *prikaziLokaciju()* za dohvat podatka lokacije objekta pomoću URL-a (engl. Uniform Resource Location) u Android web pregledniku iz aplikacije. URL je adresa određenog izvora na web-u, odnosno on je mehanizam koji preglednici koriste za dohvaćanje određenog izvora na web-u. U teoriji, svaki URL koji je važeći upućuje na jedinstveni resurs. Resursi mogu biti HTML (engl. *HyperText Markup Language*) stranica, CSS (engl. *Cascading Style Sheet*s) dokument, slika i slično. Primjer URL-a lokacije objekta je „https://goo.gl/maps/c1T6H2a1WrGNUcCs9“ koja je preuzeta s Google Maps-u. No, o tome će se govoriti u nastavku završnog rada prilikom dodavanja novog objekta [17].

Isječak programskog koda 5: Metoda prikaziLokaciju()

```
1.  private void prikaziLokaciju() {
2.  LayoutInfalater layoutInfalaterLokacija = LayoutInfalater.from(this);
3.  View lokacija = layoutInfalaterLokacija.infalate(R.layout.lokacija, root: null);
4.  WebView karta;
5.  Karta = (WebView) lokacija.findViewById(R.id.karta);
6.  Karta.getSettings().setJavaScriptEnabled(true);
7.  Karta.loadUrl(getIntent().getStringExtra(name: “karta”));
8.  }
```

4.3.6. Zaslon za pretraživanje objekata pomoću tražilice

Povratkom na zaslon kategorija, odnosno izbornika nudi nam se mogućnost odabira kategorije *Pretraži*. Klikom na *Pretraži* korisniku se prikazuje zaslon s tražilicom i popisom svih objekata na području grada Virovitice koji su upisani u bazu podataka. Pretragom objekata te klikom na objekt korisniku aplikacije se otvara zaslon s podacima o objektu – slika 12.

Slika 12. Prikaz zaslona za pretraživanje objekata pomoću tražilice

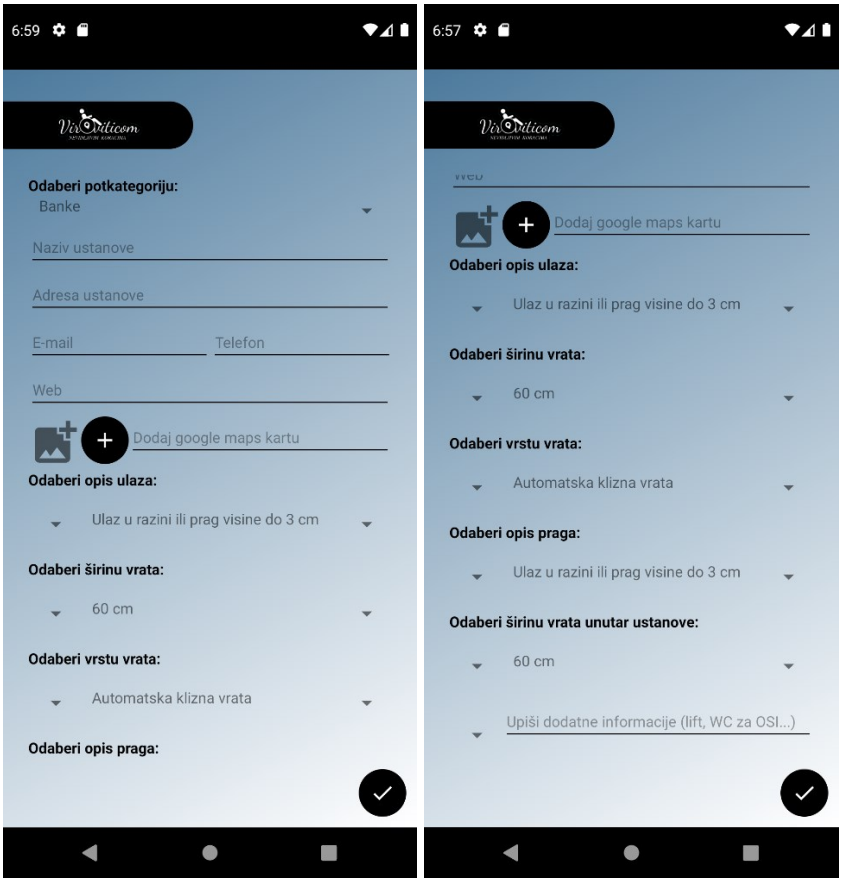
Isječak programskog koda 6 prikazuje metodu *filterList()* koja dohvaća listu objekata te ih pretražuje po imenu, ako objekt nije pronađen ispisuje se poruka „Ustanova nije pronađena.“.

Isječak programskog koda 6: Metoda filterList()

```
1.  private void filterList(String text) {
2.  ArrayList<Ustanova> filterList = new ArrayList<>();
3.  for (Ustanova ustanova : ustanovaArrayList){
4.  if(ustanova.getNaziv().toLowerCase().contains(text.toLowerCase())){
5.  filterList.add(ustanova);
6.  }
7.  }
8.  if (filterList.isEmpty()){
9.  Toast.makeText(this, "Ustanova nije pronađena.", Toast.LENGTH_SHORT).show();
10. }
11. else {
12. adapterTrailica.setFilterList(filterList);
13. }
14. }
```

4.3.7. Zaslon za dodavanje novog objekta

Osim što se korisniku povratkom na izbornik pruža odabir do sada spomenutih mogućnosti pruža mu se i mogućnost dodavanja novog objekta klikom na gumb +. Klikom na gumb korisniku se otvara zaslon s formularom za unos podataka o objektu – slika 13. Formular se popunjava odabirom podataka u padajućem izborniku i upisom, primjerice kontaktne podatke. Klikom na gumb + korisniku se pruža mogućnost odabira slike ulaza iz galerije mobilnog uređaja ili slikanjem dok za dodavanje lokacije je potrebno kopirati URL iz Google Maps-a. Nakon što je korisnik upisao sve podatke klikom na gumb koji se nalazi u donjem desnom kutu sprema ih te mu se otvara ekran s podacima upisanog objekta.



Slika 13. Prikaz zaslona dodavanje novog objekta

5. Zaključak

Život bez mobilnih uređaja postao je nezamisliv, jer oni olakšavaju i poboljšavaju kvalitetu života u svim segmentima svakodnevnice. Stoga, kako bi se osobama s invaliditetom olakšala organizacija posjeta nekom objektu napravljena je mobilna aplikacija pomoću koje osobe s invaliditetom mogu jednostavno pronaći informacije o prilagođenosti javnih objekata koje ne mogu pronaći na službenim web stranicama. S obzirom da veliki broj korisnika kod nas, a i u svijetu koristi mobilne uređaje s Android operacijskim sustavom ideja je bila napraviti mobilnu aplikaciju za taj operacijski sustav. Sama aplikacija je vrlo jednostavna za korištenje.

Aplikacija je napravljena u Android Studio-u te koristi Firebase bazu podataka za pohranu podataka o objektu. Za izradu mobilne aplikacije bilo je važno usvojiti te poznavati osnovne koncepte objektno orijentiranog programiranja te programskog jezika Java. Aplikaciju koja je je moguće pokrenuti na mobilnim uređajima koji imaju Android operacijski sustav.

Aplikaciju je u budućnosti moguće doraditi proširenjem baze podataka te izmjenom vizualnog izgleda korisničkog sučelja.

Popis literature

[1] Android for Developers. Pristupljeno: 1. kolovoza 2023. [Online]. Dostupno na: https://developer.android.com/studio/intro

[2] Priručnik za laboratorijske vježbe iz predmeta Programiranje mobilnih aplikacija, Veleučilište u Virovitici, 2022.

[3] Marko Gargenta(2011): Naučite Android. Virovitica: Gradska knjižnica i čitaonica Virovitica.

[4] Maja Karaga, Marko Stojanović (2019): Programiranje aplikacija za Android. Virovitica: Gradska knjižnica i čitaonica Virovitica.

[5] Jay Alabaster: PCWorld. Pristupljeno: 8. kolovoza 2023. [Online]. Dostupno na: https://www.pcworld.com/article/451350/android-founder-we-aimed-to-make-a-camera-os.html.

[6] How-To Geek. Pristupljeno: 8. kolovoza 2023. [Online]. Dostupno na: https://www.howtogeek.com/345250/whats-the-latest-version-of-android/.

[7] Boris Radošević: tportal. Pristupljeno: 8. kolovoza 2023. [Online]. Dostupno na: https://www.tportal.hr/tehno/clanak/stize-android-14-ovo-je-sve-sto-trebate-znati-o-njemu-foto-20221221.

[8] Android for Developers. Pristupljeno: 10. kolovoza 2023. [Online]. Dostupno na: https://developer.android.com/guide/platform.

[9] Li Ma, Lei Gu, Jin Wang: Research and Development of Mobile Application for Android Platform. Pristupljeno: 10. kolovoza 2023. [Online]. Dostupno na: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.430.9736&rep=rep1&type pdf.

[10] Zlatko Stapić, Ivan Švogor, Davor Fodrek: Priručnik za 4. razred gimnazije Razvoj mobilnih aplikacija. Pristupljeno: 12. kolovoza 2023. [Online]. Dostupno na: http://www.ss-ivanec.hr/images/8_Prirucnik_Razvoj_mobilnih_aplikacija.pdf.

[11] Yakov Fain (2013): Programiranje Java. Virovitica: Gradska knjižnica i čitaonica Virovitica.

[12] Dražen Mušanović: Java (Prigramski jezik) – Računarstvo. Pristupljeno: 15. kolovoza 2023. [Online]. Dostupno na: https://www.scribd.com/doc/183038286/Java Programski-Jezik-racunarstvo.

[13] Code Java. Pristupljeno: 15. kolovoza 2023. [Online]. Dostupno na: https://www.codejava.net/java-se/java-se-versions history#:~:text=This%20article%20gives%20you%20an,Java%20version%20on%20our%20computer).

[14] Steven Melendez: Fast Company. Pristupljeno: 17. kolovoza 2023. [Online]. Dostupno na: https://www.fastcompany.com/3031109/sometimes-youre-just-one-hop-from-something-huge.

[15] Firebase. Pristupljeno: 17. kolovoza 2023. [Online]. Dostupno na: https://firebase.google.com.

[16] ITpedia. Pristupljeno: 18. kolovoza 2023. [Online]. Dostupno na: https://hr.itpedia.nl/2019/06/03/wat-zijn-nosql-databases/.

[17] Mmdn_. Pristupljeno: 24. kolovoza 2022. [Online]. Dostupno na: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_URL.

Popis slika

Popis isječaka programskog koda

Veleučilište u Virovitici

OBRAZAC 5

IZJAVA O AUTORSTVU

Ja, Brigita Perković

izjavljujem da sam autor/ica završnog/diplomskog rada pod nazivom

Izrada aplikacije „Virovitica nevidljivim koracima" koristeći programske tehnologije za Android operacijski sustav

Svojim vlastoručnim potpisom jamčim sljedeće:
- da je predani završni/diplomski rad isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu, a što pokazuju korištene bilješke i bibliografija,
- da su radovi i mišljenja drugih autora/ica, koje sam u svom radu koristio/la, jasno navedeni i označeni u tekstu te u popisu literature,
- da sam u radu poštivao/la pravila znanstvenog i akademskog rada.

Potpis studenta/ice

Brigita Perković

Veleučilište u Virovitici

OBRAZAC 6

ODOBRENJE ZA OBJAVLJIVANJE ZAVRŠNOG/DIPLOMSKOG RADA U DIGITALNOM REPOZITORIJU

Ja Brigita Perković

dajem odobrenje za objavljivanje mog autorskog završnog/diplomskog rada u javno dostupnom digitalnom repozitoriju Veleučilišta u Virovitici sadržanom u Dabar (Digitalni akademski arhivi i repozitoriji) te u javnoj internetskoj bazi završnih radova Nacionalne i sveučilišne knjižnice bez vremenskog ograničenja i novčane nadoknade, a u skladu s odredbama članka 58. stavka 5., odnosno članka 59. stavka 4. Zakona o visokom obrazovanju i znanstvenoj djelatnosti (NN 119/22).
Potvrđujem da je za pohranu dostavljena završna verzija obranjenog i dovršenog završnog/diplomskog rada. Ovom izjavom, kao autor navedenog rada dajem odobrenje i da se moj rad, bez naknade, trajno javno objavi i besplatno učini dostupnim na sljedeći način:

a) Rad u otvorenom pristupu
b) Rad dostupan nakon: 18.9.2023. (upisati datum nakon kojeg želite da rad bude dostupan)
c) Pristup svim korisnicima iz sustava znanosti i visokog obrazovanja RH
d) Pristup korisnicima matične ustanove
e) Rad nije dostupan (*u slučaju potrebe dodatnog ograničavanja pristupa Vašem završnom/diplomskom radu, podnosi se pisani obrazloženi zahtjev*).

Potpis studenta/ice

Brigita Perković

U Virovitici, 8.9.2023.