SVEUČILIŠTE U SPLITU  
FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Izrada Android aplikacije za prikaz snimljenih GPS koordinata tijekom izvođenja fizičkih aktivnosti

Ante Lovrić

Split, Kolovoz 2019.

SVEUČILIŠTE U SPLITU

FAKULTET ELEKTROTEHNIKE, STROJARSTVA I BRODOGRADNJE

Dodiplomski studij: **Računarstvo**

Smjer/Usmjerenje:

Oznaka programa: 120

Ime i prezime: **ANTE LOVRIĆ**

Broj indeksa: 206-2016

**PRIJAVA ZAVRŠNOG RADA**

Radni naslov: Izrada Android aplikacije za prikaz snimljenih GPS koordinata tijekom izvođenja fizičkih aktivnosti

Zadatak: Ovladati znanjima potrebnim za programiranje Android uređaja, te osmisliti aplikaciju koja prihvaća GPS podatke snimljene tijekom izvođenja fizičke aktivnosti. Osigurati grafički prikaz tih podataka na Google mapi, te grafički prikaz različitih parametara iz kojih se može zaključiti o nivou utreniranosti. Razmotriti mogućnost izmjenjivanja pohranjenih podataka i odgovarajuću komunikaciju između udaljenih uređaja.

Prijava rada: 01.03.2019.

Rok za predaju rada: 20.09.2019.

Rad predan:

Datum obrane:

Predsjednik Odbora: Mentor:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof.dr.sc. Maja Štula Prof.dr. sc. Mirjana Bonković

**SADRŽAJ**

[1 Uvod 2](#_Toc17840034)

[2 Povijest androida i izbor programskog jezika 3](#_Toc17840035)

[2.1 Kotlin 3](#_Toc17840036)

[2.2 Java 4](#_Toc17840037)

[3 Korištene tehnologije 5](#_Toc17840038)

[3.1 Android Studio 5](#_Toc17840039)

[3.2 Git i Github 7](#_Toc17840040)

[4 Android aplikacija 8](#_Toc17840041)

[5 Gotova rješenja i biblioteke 9](#_Toc17840042)

[6 Zaključak 10](#_Toc17840043)

[Literatura 11](#_Toc17840044)

[Popis oznaka i kratica 12](#_Toc17840045)

[Sažetak i ključne riječi 13](#_Toc17840046)

# Uvod

U ovom dokumentu je opisan postupak izrade Android aplikacije za praćenje lokacije korisnika u stvarnom vremenu, načina dohvaćanja i spremanja podataka o lokaciji i kretanju korisnika, te načina obrade i prikaza dotičnih podataka na ekranu uređaja, napisane u Java programskom jeziku.

U poglavljima koji slijede možete pronaći osnovne informacije o Android OS-u, princip „Activity Lifecycle“-a, Android Studio alata za razvoj aplikacija, osnove Android prikaza i prijelaza iz jednog prikaza u drugi, prijenos informacija iz jednog u drugi prikaz, osnovno znanje o namještanju Google Maps lokacijskog servisa za izradu Android aplikacije itd.

Na kraju dokumenta možete pronaći gotove biblioteke koje implementiraju neke funkcionalnosti potrebne za ovu aplikaciju poput crtanja grafova i animiranih gumbova.

# Povijest androida i izbor programskog jezika

Android je besplatni operacijski sustav napravljen na Linux jezgri za mobilne uređaje s velikim ekranima, te raznim mogućnostima kao određivanjem lokacije uređaja, tipkanja na dodir, spajanja na Internet i slično.

Ovaj operacijski sustav je besplatan za korištenje, te je postao najpopularniji operacijski sustav s preko 2 milijarde korisnika u svijetu od 2007. godine kada je predstavljen javnosti.

Budući da je napisan na Linux jezgri, za izradu aplikacija se može birati između velikog broja programskih jezika kao C, C++, Java, Visual Basic, Xamarin, Kotlin itd. Svaki od tih programskih jezika ima svoje dobre i loše strane. Npr. C i C++ zbog brzine trenutno rastu na popularnosti u izradi igara za Android. Od tih jezika, dva su službeno podržana od strane Google-a kao tvrtke koja je razvila Android platformu, a to su Java i Kotlin.

## Kotlin

Kotlin je programski jezik koji je razvila tvrtka JetBrains čiji je najpoznatiji proizvod IntelliJ razvojno okruženje.

Krajem 2017. godine je podržan od strane Google-a koji je izjavio da će neke mogućnosti prvo isporučivati za Kotlin, pa tek nakon za Javu. Sredinom 2019. godine je određen kao preferirani jezik za razvoj android aplikacija.

Kotlin se koristi jer:

1. Napravljen je s ciljem smanjenja količine potrebnog koda za izradu neke funkcionalnosti. Time se i smanjuje količina koda u kojoj može nastati greška i količina koda koja se mora testirati.
2. Sam provjerava neke greške koje je potrebno ručno provjeravati u drugim programskim jezicima.
3. Podržava biblioteke napisane za Java i JavaScript jezike. S time je omogućeno pisanje aplikacija za JVM, Android i Internet preglednik.
4. Podržavaju ga sve razvojne okoline koje podržavaju Java jezik.

## Java

Java programski jezik je do sredine 2019. godine jedini službeno podržani jezik od strane Google-a za izradu Android aplikacija. Kao objektno orijentirani i najpopularniji programski jezik, postao je službenim jezikom i u njemu je pisan Android OS.

Prednosti Jave:

1. Java kod se ne mora prilagođavati ovisno o platformi na kojoj se pokreće, već to radi JVM. Java kod se pretvara u Java „byte code“ te se pokreće pomoću JVM-a.
2. Postoji velik broj ljudi stručnih u Java programskom jeziku.
3. Velika podrška od strane zajednice, uz veliku količinu biblioteka i alata besplatnih za upotrebu (Open Source).
4. Budući da nekakvo virtualno računalo pokreće Java kod, moguće je upravljati i ograničiti resursima, i tako osigurati računalo od štete koju bi mogla napraviti aplikacija neodgovornim korištenjem resursa.

Kao osobni izbor, za izradu ove aplikacije koristiti će se Java programski jezik.

# Korištene tehnologije

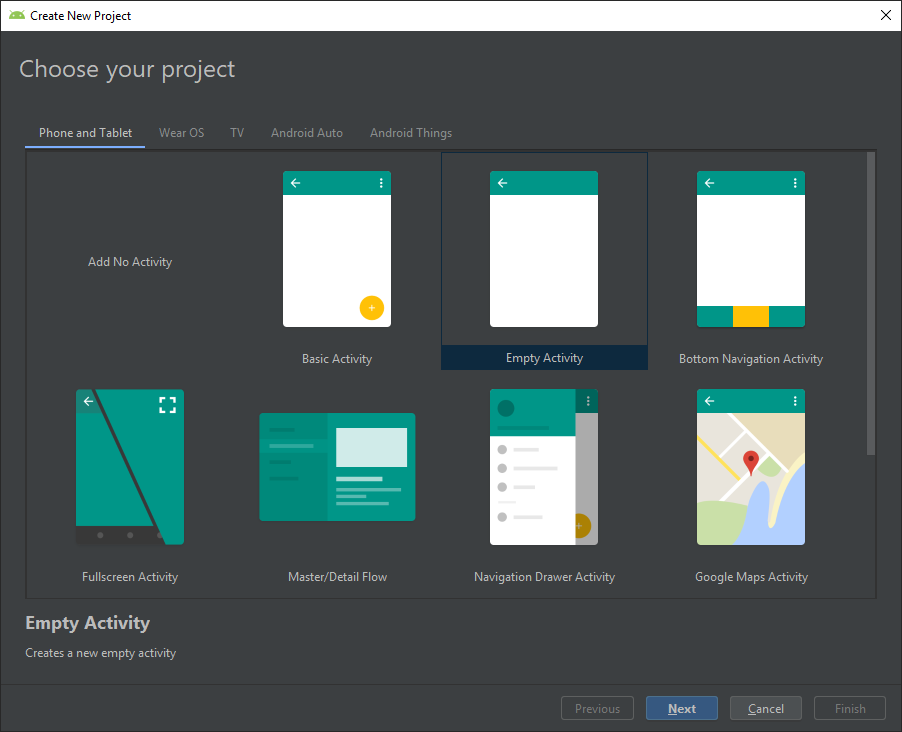
Alati i aplikacije koje su se koristile u razvoju aplikacije:

## Android Studio

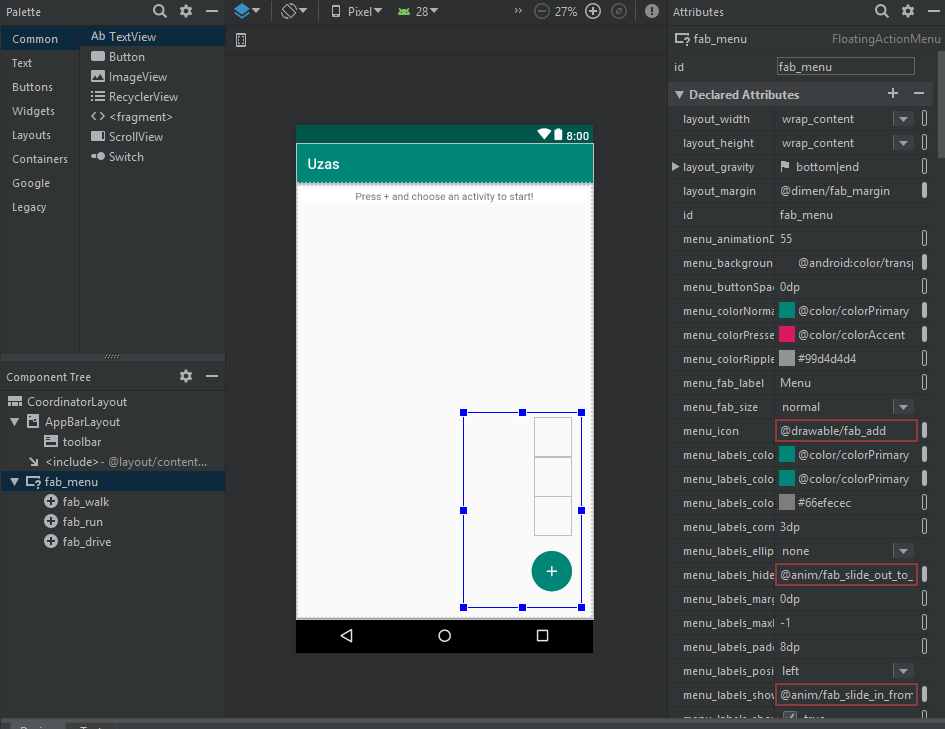
Android Studio je razvojno okruženje koju je razvio Google u svrhu lakšeg razvoja aplikacija za Android s podrškom za programske jezike Java i Kotlin.

Pogledajmo neke od mogućnosti Android Studio-a:

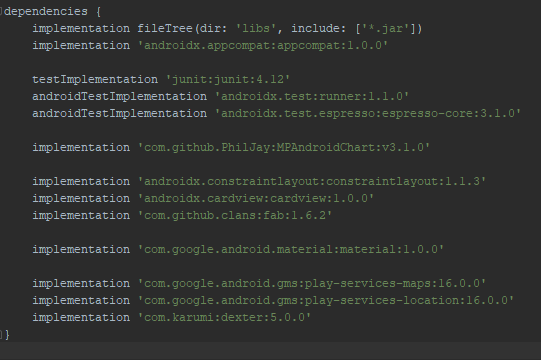
1. Izrada novog prikaza u aplikaciji ili pokretanje novog projekta s predloška prikaza (Slika 1 Prikaz izbornika za kreiranje novog projekta) gdje ovisno o tome što želimo da aplikacija radi ili što da prikazuje možemo imati gotovi kod za osnovnu aktivnost.

Slika 1 Prikaz izbornika za kreiranje novog projekta

1. Pregled projektnih datoteka i uređivanje koda se podrazumijevaju, uz opcije automatskog oblikovanja koda, i preoblikovanja koda, automatskog preimenovanja varijabli/klasa itd.
2. Android Studio omogućava odabir bilo kojeg elementa na ekranu, te uređivanje prikaza, dodavanje i brisanje stavki s virtualnog ekrana, te uređivanje parametara prikaza aplikacije (Slika 2 Prikaz sučelja za uređivanje izgleda aplikacije). Ovaj dio će biti dalje pojašnjen u nastavku dokumenta.

Slika 2 Prikaz sučelja za uređivanje izgleda aplikacije

1. Android Studio omogućava automatsko preuzimanje biblioteka s interneta i njihovu instalaciju, sve što je potrebno je dodati vezu na lokaciju gdje se nalazi projekt. Npr. Na slici (Slika 3 Isječak koda za korištenje biblioteka) vidite da za dodati biblioteku sa github repozitorija sve što treba dodati je implementation 'com.github…'. Dalje će Android Studio sve sam napraviti i biblioteka je spremna za korištenje.

Slika 3 Isječak koda za korištenje biblioteka

1. Postoji mogućnost kreiranja i pokretanja virtualnog uređaja u sklopu Android Studio aplikacije čime je moguće direktno testiranje aplikacije, ili spajanje vlastitog mobilnog uređaja na računalo uz prethodno postavljanje razvojnih mogućnosti na tom uređaju, čime se također omogućuje testiranje aplikacije te uz sve to čitanje zapisnika o tome što se sve dešava u uređaju. Korisnik ne treba brinuti o tome kako se aplikacija prenosi na mobilni uređaj i instalira.
2. Na posljetku, postoji mogućnost dodavanja mnoštva dodataka kao npr. Git integriran u Android Studio.

## Git i Github

Ako postoji ijedan alat koji je apsolutno potreban u razvoju bilo kakve aplikacije, onda je to git. Git je alat za upravljanje verzijama dokumenata. Način na koji radi je tako da ne pamti kompletne verzije dokumenata nego samo određene promjene poput linije u dokumentu koja je uklonjena, dodana ili promjenjena.

Razlozi zašto je neizostavan su kako slijedi:

1. Omogućava istovremeno uređivanje dokumenata između više ljudi, te spajanje verzija na najlakši mogući način (ako nema sukoba među njihovim verzijama). Ako postoje sukobi među verzijama, najvjerojatnije se mogu lako razriješiti.
2. Bilo kakve promjene koje se naprave na projektu i koliko god davno su reverzibilne
3. Moguće je segmentiranje funkcionalnosti na jedinične „commit“-ove i raspoznavanje što se kada napravilo i koje su promjene točno napravljene za neku funkcionalnost.
4. Podaci se istovremeno nalaze na više lokacija, pa se pri gubitku podataka na jednom računalu gube samo promjene koje nisu spremljene na repozitorij. Zato se ujedno i preporučava često isporučivanje promjena na repozitorij.

# Android aplikacija

U ovom poglavlju će biti opisani svi dijelovi aplikacije i zašto su takvi kakvi jesu, počevši od koncepta pa do korisničkog sučelja i koda.

## Koncept

Cilj je napraviti aplikaciju koja će pratiti korisnikovu lokaciju kroz vrijeme u periodu koji korisnik određuje, spremati određene podatke i prikazati ih na ekranu u uređenom obliku. Te podatke treba trajno pohraniti na uređaju tako da se mogu ponovno prikazati pri sljedećem paljenju aplikacije.

Za izradu aplikacije su stoga potrebne vještine pisanja koda u jeziku Java, osnove objektno orijentiranog programiranja, znanje o izradi i korištenju baza podataka, znanje o tome kako funkcionira Android aplikacija i razne Android biblioteke, te na poslijetku znati namjestiti i koristiti Google Maps API za prikaz i dohvaćanje trenutne lokacije korisnika.

Kroz ovaj dokument, radnja kojom korisnik prati svoju lokaciju će se zvati aktivnost.

## Podaci o aktivnostima i baza podataka

Kasnije će se objasniti Google Maps, ali sada možemo reći da znamo da će nam glavni podaci koje nam pruža lokacijski servis biti geografska širina (engl. latitude) i dužina (engl. longitude). Pri tom ćemo vremenske podatke dohvaćati s Android OS-a sustavskim pozivom za dohvaćanje vremena u milisekundama.

Ti podaci (geografska širina i dužina i vrijeme) čine osnovu ove aplikacije i iz njih možemo izračunati sve što nam treba da korisniku prikažemo statistiku za njegovu aktivnost, ali da bi pamtili različite aktivnosti, potrebne su nam dodatne informacije.

Dovoljna informacija bi bila ID aktivnosti čime bi mogli jednostavno određivati kojoj aktivnosti pripada koji skup podataka, ali zbog jednostavnosti ćemo spremati podatke o vremenu kad je aktivnost započela, ukupna prijeđena udaljenost, utrošeno vrijeme, prosječna brzina, koje ćemo računati prilikom praćenja lokacije korisnika. Osim toga, dati ćemo korisniku opciju da bira koji tip aktivnosti će izvoditi između šetnje, trčanja i vožnje, pa ćemo i tu informaciju spremiti.

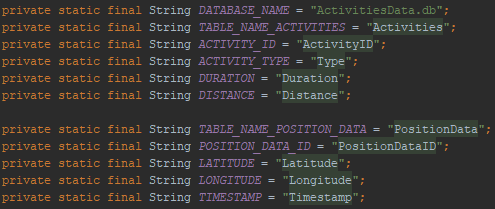
S tim zaključcima dolazimo do relacijske baze podataka s tablicama koje možetete vidjeti na slici ().

**TODO:SLIKA tablica**

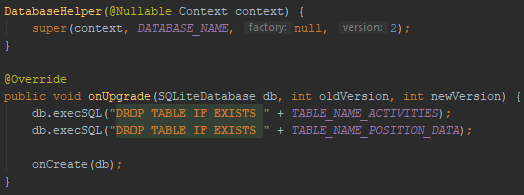
Što se tiče izvedbe, biblioteka SQLite je standardna i jedna od mnogih dostupnih koje omogućuju kreiranje SQL baza podataka na Androidu. Da bi koristili SQLite, potrebno je napraviti pomoćnu (engl. Helper) klasu koja će naslijediti baznu klasu „SQLiteOpenHelper“ koja sadrži sve funkcionalnosti koje su nam potrebne (Isječak koda 1 Implementacija pomoćne klase za upravljanje bazom podataka).

Isječak koda 1 Implementacija pomoćne klase za upravljanje bazom podataka

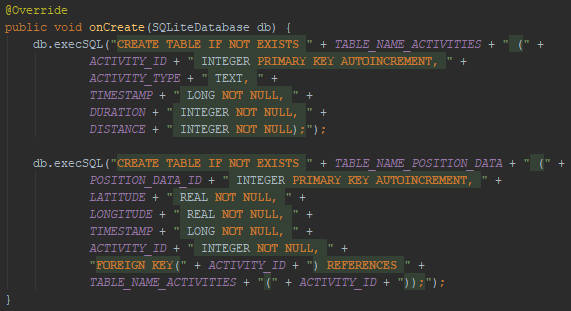
U pomoćnoj klasi su definirane vrijednosti (Isječak koda 2 Definicija imena baze podataka, tablica i stupaca u tablicama) imena stupaca u tablicama kako bi se izbjegla greška pri implementaciji pretrage podataka u tablici i unosa podataka u tablicu.

Pri inicijalizaciji pomoćne klase (pozivu konstruktora) potrebno je pozvati konstruktor SQLiteOpenHelper klase kojim specificiramo ime baze podataka i verziju. Ako je verzija baze podataka novija od prošlog puta kad se pokretala aplikacija, pokreće se metoda „onUpdate“ koju također trebamo implementirati (Isječak koda 3 Konstruktor pomoćne klase i onUpgrade metoda), u kojoj bi trebali izbrisati postojeće tablice i njihove retke koje su neispravni zbog novije verzije podataka (ili prebaciti sve te podatke u podatke pogodne noviju verziju).

Isječak koda 2 Definicija imena baze podataka, tablica i stupaca u tablicama

Isječak koda 3 Konstruktor pomoćne klase i onUpgrade metoda

Metoda „onCreate“ (Isječak koda 4 Kreiranje tablica u bazi podataka) se poziva pri paljenju aplikacije i služi za kreiranje tablica unutar baze podataka.

Isječak koda 4 Kreiranje tablica u bazi podataka

# Gotova rješenja i biblioteke

# Zaključak

# Literatura

1. [https://www.android.com](https://www.android.com/)
2. [https://stackoverflow.com](https://stackoverflow.com/)
3. [https://www.quora.com](https://www.quora.com/)
4. <https://specialties.bayt.com/en/specialties/q/4553/why-do-we-use-java-for-creating-android-application/>
5. [https://kotlinlang.org](https://kotlinlang.org/)

# Popis oznaka i kratica

JVM – Java Virtual Machine

# Sažetak i ključne riječi