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](#_Toc18182289)

[2 Povijest androida i izbor programskog jezika 3](#_Toc18182290)

[2.1 Kotlin 3](#_Toc18182291)

[2.2 Java 4](#_Toc18182292)

[3 Korištene tehnologije 5](#_Toc18182293)

[3.1 Android Studio 5](#_Toc18182294)

[3.2 Git i Github 7](#_Toc18182295)

[4 Android aplikacija 8](#_Toc18182296)

[4.1 Koncept 8](#_Toc18182297)

[4.2 Podaci o aktivnostima 8](#_Toc18182298)

[4.3 Baza podataka 10](#_Toc18182299)

[4.3.1 Biblioteka 10](#_Toc18182300)

[4.3.2 Definiranje i kreiranje tablica 10](#_Toc18182301)

[4.3.3 Unos podataka 11](#_Toc18182302)

[4.3.4 Dohvaćanje podataka 12](#_Toc18182303)

[5 Gotova rješenja i biblioteke 13](#_Toc18182304)

[6 Zaključak 14](#_Toc18182305)

[Literatura 15](#_Toc18182306)

[Popis oznaka i kratica 16](#_Toc18182307)

[Sažetak i ključne riječi 17](#_Toc18182308)

# 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.

Za izradu ove aplikacije koristiti će se Java programski jezik.

# Korištene tehnologije

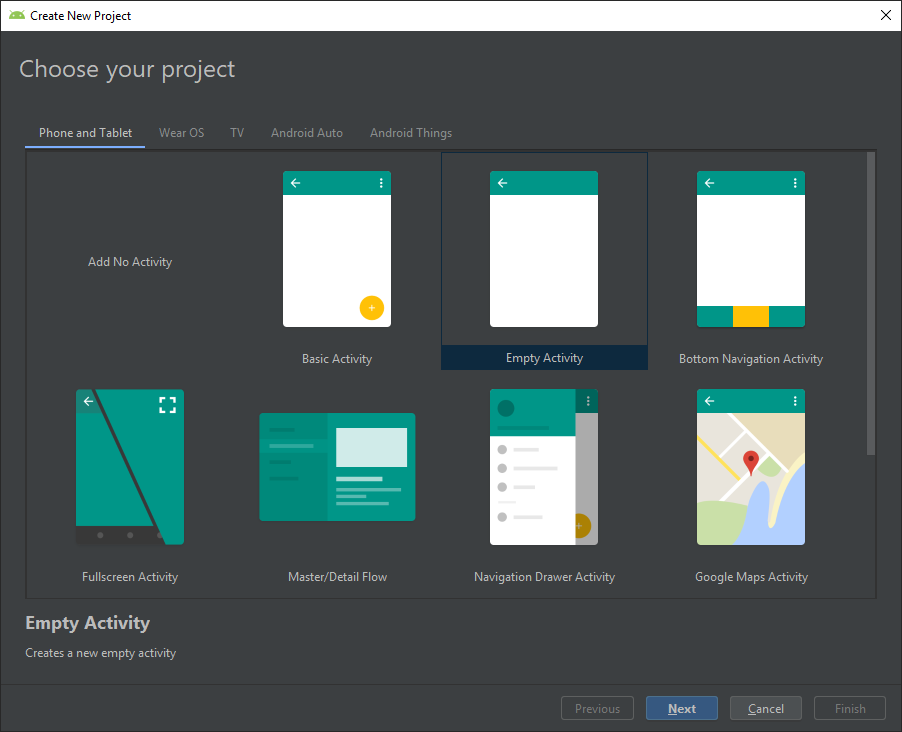
Alati i aplikacije koje su se koristile u razvoju aplikacije:

## Android Studio

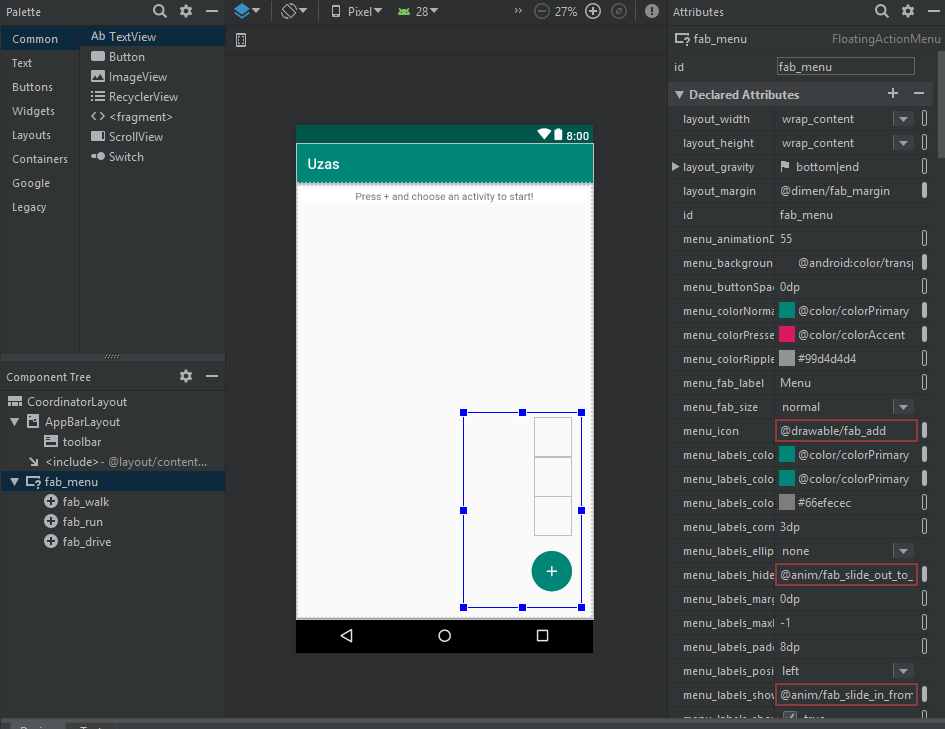
Android Studio je razvojno okruženje koju je razvio Google u svrhu lakšeg razvoja aplikacija za Android s podrškom za programske jezike Java i Kotlin.

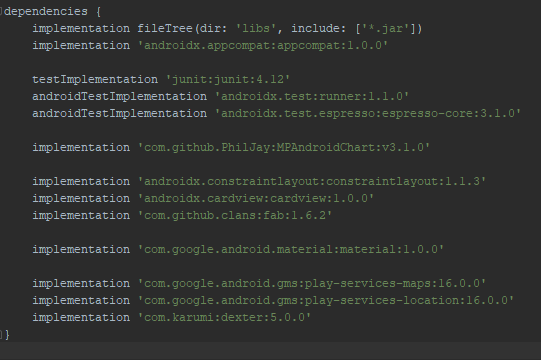
Pogledajmo neke od mogućnosti Android Studio-a:

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 isječku (Isječak koda 1 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.
2. 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.

Isječak koda 1 Korištenje biblioteka

1. 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 su potrebne sljedeće vještine:

* Programski jezik Java
* Osnove objektno orijentiranog programiranja
* Osnove baza podataka
* Znanje o funkcioniranju Android aplikacija i razne Android biblioteke
* Korištenje Google Maps API-ja za dohvaćanje i prikaz trenutne lokacije korisnika.

## Podaci o aktivnostima

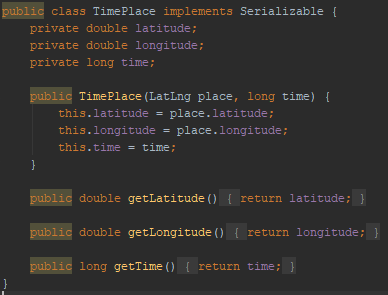
Podaci koje Google Maps lokacijski servis pruža su trenutna geografska širina (engl. Latitude) i dužina (engl. Longitude). Trenutak u kojem se dohvati lokacija se može odrediti pozivom metode „System.CurrentTimeMillis()“ koja vraća trenutno vrijeme i datum kroz vremensku oznaku (engl. Timestamp) u milisekundama. Ti podaci su osnova ove aplikacije i iz njih se mogu izračunati sve potrebne informacije da se korisniku prikaže statistika za neku aktivnost.

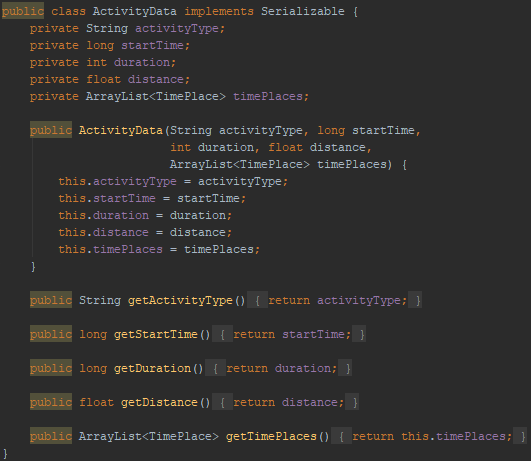
Pomoćni podaci koji će se spremati u bazu podataka su:

* Vrsta aktivnosti
* Vrijeme početka aktivnosti
* Trajanje aktivnosti
* Ukupna prijeđena udaljenost

Korisnik treba imati opciju biranja vrste aktivnosti koju će izvoditi – šetnja, trčanje ili vožnja. Ostali pomoćni podaci se mogu izračunati iz onih podataka koji će svakako biti u bazi podataka, međutim radi jednostavnosti su ostavljeni.

U isječku (Isječak koda 2 Klasa za pohranu informacija o lokaciji) vidimo klasu „TimePlace“ u koju će se spremati osnovni podaci o trenutnoj lokaciji korisnika.

Isječak koda 2 Klasa za pohranu informacija o lokaciji

Klasa TimePlace sama za sebe nema mnogo značaja i može sadržavati samo jednu lokaciju od mnogih u aktivnosti, tako da sljedeća klasa imena „ActivityData“ (Isječak koda 3 Klasa za pohranu informacija o aktivnosti) sadržava sve potrebne informacije o jednoj aktivnosti.

Isječak koda 3 Klasa za pohranu informacija o aktivnosti

Ova klasa sadržava sve pomoćne informacije koje će se spremati u bazu podataka uz niz informacija o lokacijama korisnika kroz vrijeme koje trajanja aktivnosti.

## Baza podataka

Baza podataka se uvodi jer omogućava spremanje i čitanje svih novih i starih podataka o aktivnostima.

### Biblioteka

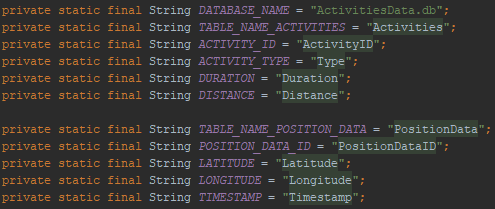
SQLite je standardna biblioteka za kreiranje relacijskih baza podataka pomoću SQL jezika na Androidu. Korištenje SQLite zahtjeva kreiranje pomoćne (engl. Helper) klase koja će naslijediti baznu klasu „SQLiteOpenHelper“ (Isječak koda 2 Deklaracija pomoćne klase) koja sadrži sve potrebne funkcionalnosti za upravljanje relacijskom bazom podataka.

Isječak koda 4 Deklaracija pomoćne klase

### Definiranje i kreiranje tablica

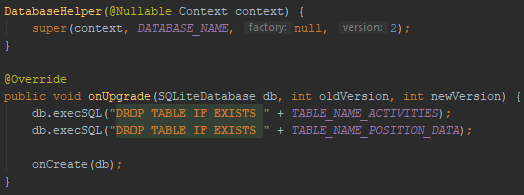
Prikaz baze podataka možetete vidjeti na slici ().

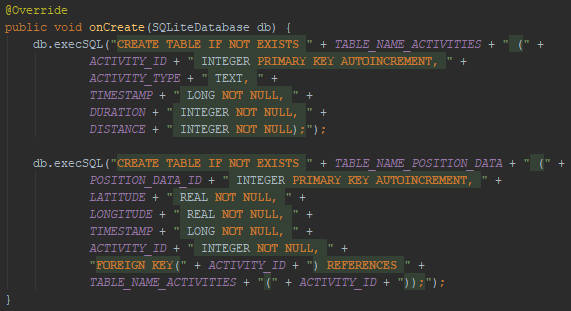
**TODO:SLIKA tablica**

U pomoćnoj klasi su definirane vrijednosti imena stupaca u tablicama (Isječak koda 3 Definicija imena u pomoćnoj klasi) kako bi se izbjegla greška pri implementaciji pretrage podataka u tablici i unosa podataka u tablicu.

Isječak koda 5 Definicija imena u pomoćnoj klasi

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 koje su neispravne zbog novije verzije istih.

Isječak koda 6 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 7 Kreiranje tablica u bazi podataka

### Unos podataka

Nakon svake aktivnosti u bazu se unose novi podaci. Ovo ostvarujemo služeći se SQL naredbama za umetanje (engl. Insert)

### Dohvaćanje 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