

# Napredne mogućnosti operacijskog sustava Android

---

Ivšić, Antonio

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:815126>

Rights / Prava: [Attribution-NonCommercial 3.0 Unported / Imenovanje-Nekomercijalno 3.0](#)

Download date / Datum preuzimanja: **2024-07-17**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU  
FAKULTET ORGANIZACIJE I INFORMATIKE  
VARAŽDIN**

**Antonio Ivšić**

**NAPREDNE MOGUĆNOSTI  
OPERACIJSKOG SUSTAVA ANDROID**

**ZAVRŠNI RAD**

**Varaždin, 2023.**

**SVEUČILIŠTE U ZAGREBU**  
**FAKULTET ORGANIZACIJE I INFORMATIKE**  
**V A R A Ž D I N**

**Antonio Ivšić**

**Matični broj: 0016142443**

**Studij: Informacijski sustavi**

**Napredne mogućnosti operacijskog sustava android**

**ZAVRŠNI RAD**

**Mentor/Mentorica:**

Prof. dr. sc. Ivan Magdalenić

**Varaždin, kolovoz 2023.**

*Antonio Ivšić*

### **Izjava o izvornosti**

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

*Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi*

---

## **Sažetak**

Ovaj završni rad bavi se temom Android operacijskog sustava, njegovom poviješću, procesom razvoja Android aplikacija te prikazom nekih od mogućnosti operacijskog sustava na aplikaciji. Rad je podijeljen na dva dijela, teorijski i praktični. Teorijski dio bavi se poviješću, osnovnim pojmovima vezanim za Android, postupkom i alatima za razvoj Android aplikacija. Za praktični dio izrađena je Android aplikacija sa tri funkcionalnosti. Kod izrade aplikacije korišteno je razvojnog okruženja Android Studio Giraffe, programski jezik Kotlin i alat za izradu izvornog Android korisničkog sučelja, Jetpack Compose. Aplikacija se sastoji od funkcije promjene teme cijele aplikacije, funkcije traženja i prikaza stanja dozvola koje ima aplikacija i funkcije prikaza koordinata korisnikovog dodira na ekranu.

**Ključne riječi:** Android; Android Studio; Kotlin; Jetpack Compose; Google; Dozvole; Teme

# Sadržaj

Sadržaj .....	iii
1. Uvod .....	1
2. Evolucija operacijskog sustava Android .....	2
2.1. Povijest .....	4
2.1.1. Skromni počeci .....	4
2.1.2. Googleov najbolji posao .....	4
2.2. Android operacijski sustav .....	5
2.2.1. Funkcije operacijskog sustava .....	5
2.2.1.1. Multi-tasking .....	5
2.2.1.2. Prilagodljivost .....	6
2.2.1.3. Sigurnost .....	6
2.2.1.4. Podrška za više platformi .....	6
2.2.1.5. Podrška aplikacija .....	6
2.2.2. Verzije i noviteti .....	7
2.2.2.1. Android 1.0 i 1.1 .....	7
2.2.2.2. Android 1.5 Cupcake .....	7
2.2.2.3. Android 1.6 Donut .....	7
2.2.2.4. Android 2.0 i 2.1 Enclair .....	7
2.2.2.5. Android 2.2 Froyo .....	8
2.2.2.6. Android 2.3 Gingerbread .....	8
2.2.2.7. Android 3.0 do 3.2 Honeycomb .....	8
2.2.2.8. Android 4.0 Ice Cream Sandwich .....	8
2.2.2.9. Android 4.1 do 4.3 Jelly Bean .....	8
2.2.2.10. Android 4.4 KitKat .....	9
2.2.2.11. Android 5.0 i 5.1 Lollipop .....	9
2.2.2.12. Android 6.0 Marshmallow .....	9
2.2.2.13. Android 7.0 i 7.1 Nougat .....	9
2.2.2.14. Android 8.0 i 8.1 Oreo .....	9
2.2.2.15. Android 9 Pie .....	10
2.2.2.16. Android 10 .....	10
2.2.2.17. Android 11 .....	10
2.2.2.18. Android 12 .....	10
2.2.2.19. Android 13 .....	11
2.2.2.20. Android 14 .....	11
3. Android ekosustav .....	12

3.1. Komponente Android ekosustava.....	12
3.1.1. Google .....	12
3.1.2. Proizvođači originalne opreme .....	12
3.1.3. Razvojni programeri .....	13
3.1.4. Korisnici .....	13
4. Razvoj Android aplikacije .....	14
4.1. Alati za razvoj.....	14
4.2. Razvijena aplikacija.....	14
4.2.1. Funkcija promjene teme .....	17
4.2.2. Funkcija traženja i prikaza stanja dozvola .....	23
4.2.3. Funkcija prikaza koordinata dodira .....	30
5. Zaključak .....	33
Popis literature .....	34
Popis slika .....	36

# 1. Uvod

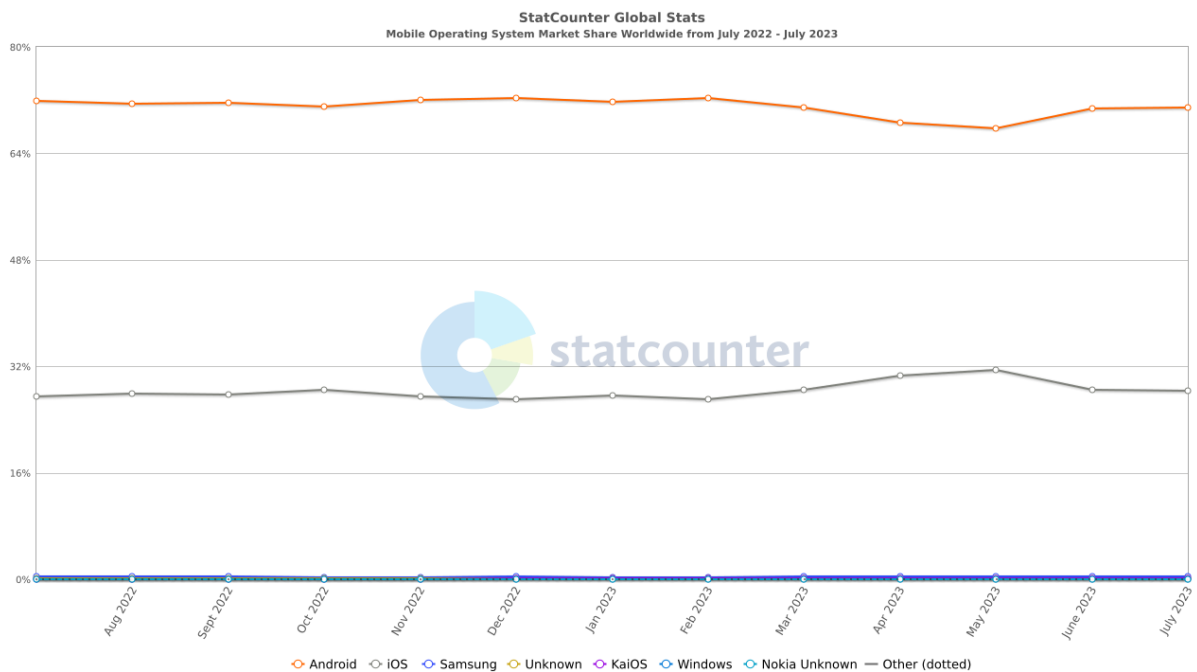
Završni rad baziran je na naprednim mogućnostima operacijskog sustava Android, ali kako bi uopće mogli govoriti o naprednim mogućnostima prvo trebamo znati osnove. U prvom dijelu ovog rada fokusirat ću se na povijesti Androida kao operacijskog sustava, razvoja Android aplikacija i značaju koji Android ima u modernom IT društvu. Većina nas je upoznata sa Androidom kao operacijskim sustavom na našim pametnim telefonima, televizorima, a u novije vrijeme i automobilima. Upravo ta sveprisutnost ovog operacijskog sustava navela me na pisanje ovog rada kako bih se i sam detaljnije upoznao sa Androidom i procesom razvoja aplikacije.

Drugi dio rada fokusiran je na opisu aplikacije koja je napravljena u sklopu istog. Aplikacija je koncipirana na način da prikazuje tri naprednije mogućnosti operacijskog sustava. Funkcionalnošću promjene dizajna cijele aplikacije, funkcionalnošću traženja dozvola od korisnika te prikaz stanja dozvola koje aplikacija ima i koje je korisnik prihvatio ili odbio te funkcionalnošću praćenja korisnikovog dodira i prikazom koordinata dodira. Kroz te tri funkcionalnosti htio sam obuhvatiti širok spektar funkcionalnosti operacijskog sustava od ulaznih uređaja do dozvola i izgleda samog sustava.

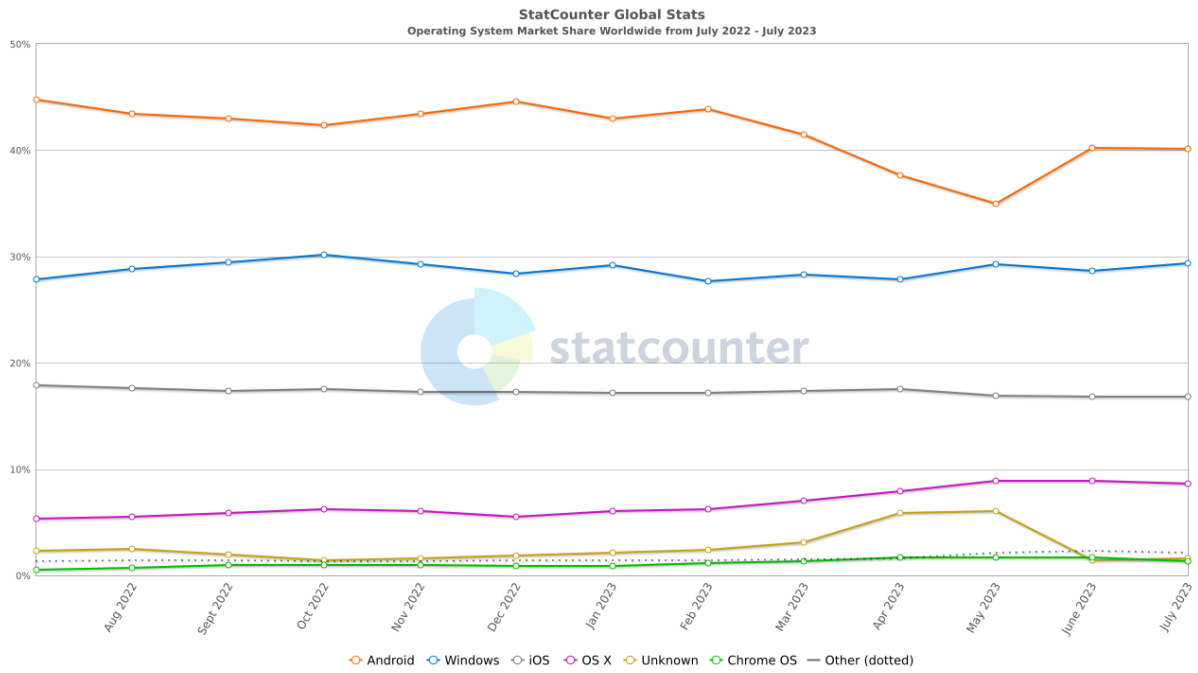


## 2. Evolucija operacijskog sustava Android

Operacijski sustav Android trenutno je najrašireniji mobilni operacijski sustav. Prema podacima iz srpnja 2023. inačice Androida nalaze se na 70,9% pametnih telefona [1]. Zanimljiviji podatak je da je Android najrašireniji operacijski sustav općenito. Prema podacima iz srpnja 2023. čak 40,15% svih operacijskih sustava u korištenju otpada na neku od inačica Androida [2].



Slika 1: Tržišni udio Android operacijskog sustava (mobilni) [1]



Slika 2: Tržišni udio Android operacijskog sustava (općenito) [2]

## 2.1. Povijest

U nastavku je detaljnije objašnjena povijest operacijskog sustava Android od njegovih skromnih početaka do tržišnog giganta koji je danas. Također ću ukratko proći kroz verzije i bitne novitete koje je svaka verzija implementirala.

### 2.1.1. Skromni počeci

Android Inc. Osnovan je u listopadu 2003 u Palo Altu, Kaliforniji. Osnivači su Andy Rubin, Rich Miner, Nick Sears i Chris White. Prvotni cilj tvrtke bio je razviti operacijski sustav namijenjen pametnim mobilnim uređajima. Prvi od pametnih uređaja koji su bili u fokusu su bile digitalne kamere, ta ideja brzo je odbačena te se fokus prebacio na mobilne uređaje. Prve dvije godine bile su teške i tvrtka se borila za opstanak dok su bili odbijani od industrijskih titana kao što su Samsung i HTC. Naposljetku u srpnju 2005. Android Inc. Kupio je Google za oko 50 milijuna dolara. [3],[4],[5]

### 2.1.2. Googleov najbolji posao

Najbolji posao, upravo te riječi upotrijebio je 2010. tadašnji podpredsjednik korporativnog razvitka tvrtke Google, David Lawee. Kupnjom Android Inc-a 2005. Google su se pridružili ključni zaposlenici Rubin, Miner, Sears i White. Google je daljnji razvoj Androida držao u tajnosti, znalo se samo da razvijaju novi program za mobilne uređaje. Tim zadužen za razvoj predvodio je njegov osnivač Andy Rubin. [5]

Rani prototip operacijskog sustava bio je namijenjeni uređajima sa fizičkom tipkovnicom i ekranom kako bi bili konkurentni tadašnjem broju jedan, BlackBerry-u. Sve se promijenilo kada je 2007. Apple lansirao iPhone, uređaj sa ekranom osjetljivim na dodir i operacijskim sustavom iPhone OS koji je dizajniran za rad sa ekranima osjetljivim na dodir. Google je promijenio specifikacije Androida kako bi podržavao ekrane osjetljive na dodir. Istim putem krenuli su Nokia i BlackBerry koji su 2008. lansirali pametne telefone sa ekranom osjetljivim na dodir kako bi konkurirali Appleu i iPhoneu 3G. [4]

## 2.2. Android operacijski sustav

Android je mobilni operacijski sustav baziran na modificiranoj verziji Linux kernela i softveru otvorenog koda (eng. *Open-Source Software*). Primarno je namijenjen mobilnim uređajima sa ekranom osjetljivim na dodir kao što su pametni telefoni, pametni satovi i tableti. Android razvija konsortium pod imenom Open Handset Alliance [6]. Ali dakako najpoznatiju verziju Androida razvija Google [7].

Android operacijski sustav je u jezgri poznat pod nazivom Android Open Source Project (kraće AOSP) i on je besplatan i softver otvorenog koda pod licencom Apache Licences [8]. Iako je besplatan većina uređaja koristi vlasničku verziju tvrtke Google koja dolazi sa softverom zatvorenog koda (eng. *Closed-Source Softwarom*, kraće CSS). Najbitniji CSS koji Google pruža je Google mobilne usluge (eng. *Google Mobile Services*) koji uključuje aplikacije kao što su Google Chrome, Google Play i Google Play usluge (eng. *Google Play Services*) [9].

Iako je AOSP besplatan ime Android i logo su pod vlasništvom tvrtke Google što ograničava njegovo korištenje izvan Google ekosustava. Zbog toga preko 70% uređaja koji koriste AOSP koriste upravo Googleovu verziju koja je poznata pod nazivom Android [9].

Činjenica da je AOSP otvorenog koda dovela je do razvoja varijanti operacijskog sustava za razne vrste mobilnih uređaja. Najpoznatije varijante su naravno Android TV i Wear OS tvrtke Google [9].

### 2.2.1. Funkcije operacijskog sustava

Android je operacijski sustav primjereno namijenjen mobilnim uređajima te je sukladno tome velika većina funkcija orijentirana mobilnim korisnicima. Uz standardne funkcije operacijskih sustava Android se odlikuje posebnostima koje su u nastavku pobrojane i ukratko opisane [10].

#### 2.2.1.1. Multi-tasking

Ova funkcija operacijskog sustava omogućava korisniku korištenje više aplikacija istovremeno. Postoji nekoliko načina korištenja, prvi je da se jedna aplikacija prikazuje dok druge rade u pozadini, a druga je da se nekoliko aplikacija pokazuje i koristi istovremeno. Ova funkcionalnost povećava produktivnost omogućuje paralelno obavljanje više stvari na uređaju [10].

### **2.2.1.2. Prilagodljivost**

Korisnici Androida mogu svoj uređaj prilagoditi sebi. Mogu lako promijeniti font, temu, pozadinu, raspored aplikacija i još mnogo toga što se tiče izgleda. Također mogu kreirati widžete, odnosno male elemente korisničkog sučelja (eng. *User Interface*, kraće UI) koji prikazuju informacije određenih aplikacija sve u svrhu skraćivanja vremena potrebnog za pronalazak istih [10].

### **2.2.1.3. Sigurnost**

Android ima velik broj ugrađenih sigurnosnih mogućnosti koje osiguravaju sigurnost uređaja i podataka na njemu. Za svako korištenje elemenata uređaja na primjer kamere svaka aplikacija od korisnika mora dobiti dozvolu za korištenje. Korisnik ima veliku kontrolu nad razinom dozvola koje će dati aplikaciji i samim trajanjem te dozvole. Uz dozvole dodatnu razinu sigurnosti osigurava ugrađeno kriptiranje podataka na uređaju kao i više faktorska autentifikacija. U slučaju gubitka uređaja postoji opcija „Pronađi moj uređaj“ kojoj je moguće pristupiti preko drugog uređaja i pomoću svojih korisničkih podataka dobiti informacije o lokaciji i statusu uređaja. Sve ove funkcionalnosti rade zajedno kako bi osigurali sigurnost i privatnost korisnika i uređaja [10].

### **2.2.1.4. Podrška za više platformi**

Android je baziran na linuxu što u prijevodu znači da je otvorenog koda i samim time podržan na velikom broju različitih uređaja. Iako je najpoznatija verzija Androida ona za mobilne telefone koju razvija Google u posljednjih nekoliko godina pojavile su se verzije za druge pametne uređaje koji su sve više prisutni. Među najrasprostranjenijima su Wear OS i Android TV. Wear OS je inačica Androida koju razvija Google i namijenjena je pametnim satovima. Radi se o modificiranoj verziji mobilnog operacijskog sustava koja omogućuje lakše i intuitivnije korištenje aplikacija, widžeta i povezivanje između pametnih satova i pametnih telefona. Android TV također razvija Google i on je namijenjen pametnim televizorima. Dizajniran je kako bi olakšao navigaciju kroz postavke uređaja i korištenje aplikacija i funkcija koje su najzastupljenije na velikim ekranima [10].

### **2.2.1.5. Podrška aplikacija**

Android kao operacijski sustav nudi puno funkcionalnosti, ali također omogućava korisnicima proširivanje svojih mogućnosti aplikacijama. Aplikacije mogu biti banalne kao aplikacije za pisanje bilješki ili kompleksne kao mobilne igre. Svrha aplikacija je omogućiti korisnicima pristup svim funkcijama koje su im potrebne za obavljanje željenih stvari na mobilnom uređaju bilo to kreiranje popisa za kupovinu, uređivanje fotografija i videa, slanje mailova i obavljanje video poziva i jednostavno razbibrige igrajući mobilnu igru [10].

## **2.2.2. Verzije i noviteti**

U nastavku su pobrojane verzije Android operacijskog sustava od verzije 1.0 sve do trenutno najnovije verzije 14. Ukratko su opisane bitne informacije i noviteti koje je svaka verzija donijela.

### **2.2.2.1. Android 1.0 i 1.1**

Prva inačica Androida objavljena 2008. godine bez danas prepoznatljivog kodnog imena. Ovdje se radilo o jednostavnoj verziji operacijskog sustava koja nije podržavala instalaciju aplikacija i koristila je fizičku tipkovnicu. Operacijski sustav imao je ugrađene rane verzije Google aplikacija, Gmail, Karte i YouTube. [11]

### **2.2.2.2. Android 1.5 Cupcake**

2009. godine objavljena je verzija 1.5. kodnog imena Cupcake čime je započela Googleova tradicija imenovanja verzija prema Američkim slasticama i slatkišima abecednim redom. U ovoj verziji dodana je podrška za tipkovnicu na ekranu, „razvojni okvir“ (eng. *Framework*) za widžete trećih strana i dodala opciju snimanja video zapisa. [11]

### **2.2.2.3. Android 1.6 Donut**

Na jesen 2009. objavljena je verzija 1.6 kodnog imena Donut koja je dodala podršku za ekrane različitih dimenzija što je bilo ključno za daljnji razvoj operacijskog sustava. Također je dodana podrška za CDMA mreže što je osiguralo da svi Američki tele operateri mogu podržavati mobilne uređaje sa Android operacijskim sustavom. [11]

### **2.2.2.4. Android 2.0 i 2.1 Enclair**

Samo šest tjedana nakon objavljivanja verzije 1.6 objavljena je verzija 2.0 kodnog imena Enclair, a nekoliko mjeseci nakon i verzija 2.1 istog kodnog imena. Ovo je prva verzija Androida koja je bila u središtu marketinške kampanje koju je vodila telekomunikacijska tvrtka Verizon promovirajući originalni Motorola Droid. [11]

Ovo je jedna od značajnijih verzija koja je dodala funkcije glasovne navigacije, praćenja prometa u stvarnom vremenu, animirane pozadinske fotografije i prvu verziju funkcije „govor u tekst“ (eng. *Speech-to-Text*) te dotada ekskluzivno iOS funkciju „uštipni za zumiranje“ (eng. *Pinch-to-Zoom*). [11]

### **2.2.2.5. Android 2.2 Froyo**

Samo 4 mjeseca nakon objavljivanja verzije 2.1 objavljena je verzija 2.2 kodnog imena Froyo. Ova verzija donijela je uglavnom poboljšanje performansi operacijskog sustava, ali donijela je i nekoliko novih funkcija. Podrška preglednika za Flash, dodavanje danas standardne ladice za navigaciju i prvu verziju glasovnih akcija koja je omogućavala pokretanje jednostavnih akcija kao što su traženje uputa i rađenje podsjetnika glasom. [11]

### **2.2.2.6. Android 2.3 Gingerbread**

Ova verzija kodnog imena Gingerbread bavila se isključivo vizualnim identitetom Androida. Cijeli UI promijenjen je kako bi bio u stilu sada klasične svijetlo zelene boje maskote Androida. Cijeli UI bazirao se na crnoj i zelenoj boji čime je Android dobio svoj klasični vizualni identitet. [11]

### **2.2.2.7. Android 3.0 do 3.2 Honeycomb**

Ove verzije Androida kodnog imena Honeycomb bile su ekskluzivne za tablete i potpuno su promijenile dizajn UI-a kako bi on više odgovarao većim ekranima tableta. Iako ova verzija nije bila objavljena za pametne telefone, postavila je temelje daljnjeg razvoja za njih sa prvim potpuno softverskim rješenjem navigacije i promjenama u UI-u za nedavno korištene aplikacije. [11]

### **2.2.2.8. Android 4.0 Ice Cream Sandwich**

2011. objavljena je verzija 4.0 Ice Cream Sandwich koja je dovela Android u eru modernog dizajna. Ova verzija ujedinila je elemente UI-a za tablete i pametne telefone te kreirala jednu viziju UI-a za cijeli Android. Zadržani on-screen navigacijski gumbi i funkcija nedavnih aplikacija, a gesta poteza prstom postala je ključni dio interakcije sa sustavom. [11]

### **2.2.2.9. Android 4.1 do 4.3 Jelly Bean**

U razdoblju od 2012. do 2013. godine objavljene su tri verzije Androida kodnog imena Jelly Bean. Ove verzije težile su tome da Android učine primamljivijim novim korisnicima donoseći mnoge inovacije. Najznačajnije su Google Now, interaktivne obavijesti, poboljšani sustav glasovnog pretraživanja, podrška za više korisničkih profila na jednom uređaju i napredniji sustav pretraživanja i prikaza rezultata pretraživanja. [11]

#### **2.2.2.10. Android 4.4 KitKat**

Krajem 2013. objavljena je verzija kodnog imena KitKat koja je Android uvela u suvremeni svijet dizajna. Tamne boje prijašnjih verzija zamijenjene su svijetlim i vedrim bojama, prozirna statusna traka i bijele ikone dali su Androidu moderni izgled. Također nastavljeno je razvijanje Google Now-a i dolazi do prve pojave prve verzije danas poznate „OK, Google“ funkcije. [12]

#### **2.2.2.11. Android 5.0 i 5.1 Lollipop**

Na jesen 2014. godine objavljena je verzija kodnog imena Lollipop čime je ponovno redizajnirao cijeli vizualni identitet Androida lansiranjem Material Design standarda koji je i danas u uporabi. Ovaj standard donio je novi izgled koji se primijenio na operacijskom sustavu, aplikacijama, ali i ostalim Googleovim proizvodima. Ova verzija utemeljila je UI obrazac baziran na karticama i donio novitete u prioritiziranju obavijesti, kao i potpuni „bez ruku“ (eng. *Handsfree*) način kontroliranja uređaja sa unapređenjem funkcije „OK, Google“. [13]

#### **2.2.2.12. Android 6.0 Marshmallow**

Ova verzija kodnog imena Marshmallow objavljena je 2015. godine i bila je samo inkrementalna promjena u odnosu na prijašnju verziju. Nije donijela puno noviteta u smislu UI-a, ali je omogućila detaljnije upravljanje dozvolama aplikacija te dodala podršku za USB-C i čitaće otisaka prstiju. [14]

#### **2.2.2.13. Android 7.0 i 7.1 Nougat**

2016. objavljena je verzija kodnog imena Nougat koja je donijela puno noviteta. Dodana je podrška za podijeljeni ekran (eng. *Split-screen*), novi način grupiranja i prikaza obavijesti, funkciju štednje podatkovnog prometa. Najznačajniji novitet bio je lansiranje Google asistenta (eng. *Google Assistant*) koji je danas prisutan u svim Googleovim proizvodima i jedan je od glavnih proizvoda ove tvrtke. [15]

#### **2.2.2.14. Android 8.0 i 8.1 Oreo**

Verzija kodnog imena Oreo objavljena je 2017. i donijela je puno stvari koje su olakšale korištenje operacijskog sustava i učinile ga primamljivijim. Dodana je podrška za sliku u slici (eng. *Picture-in-Picture*), opcija stišavanja obavijesti kao i detaljnija kontrola nad obavijestima pojedinih aplikacija. [16]



### **2.2.2.15. Android 9 Pie**

U kolovozu 2018. objavljena je verzija kodnog imena Pie koja je donijela novi način kretanja kroz operacijski sustav. Standardna navigacija pomoću tri gumba na zaslonu zamijenjena je hibridnim sustavom sa jednim home gumbom i gestama za navigaciju. Dodane su funkcionalnosti preporučenog odgovora na poruke i pozive, kontrolna ploča digitalnog zdravlja, poboljšani su sustavi upravljanja svjetlinom i potrošnjom, poboljšani način upravljanja povezivanja na WiFi i poboljšani sustav čuvanja baterije. [17]

### **2.2.2.16. Android 10**

U rujnu 2019. objavljena je verzija 10, prva kodnog se riješila kodnog imena. Ova verzija unaprijedila je navigaciju prijašnje oslanjajući se na navigaciju gestama. Najznačajnija promjena došla je u obliku upravljanja dozvolama aplikacija koje je transparentno i jednostavno za shvatiti. Također je dodana podrška za Tamni način na razini cijelog sustava i novi Fokus način rada. [18]

### **2.2.2.17. Android 11**

Android 11 bio je jedan od najznačajnijih verzija do sad koja je objavljena u rujnu 2020. Ova verzija donijela je mnogo površinskih i promjena ispod površine. Teme ove verzije je privatnost i upravljanje onime što pojedina aplikacija zna o vama kao korisniku. Unaprijeđeno je upravljanje dozvola aplikacija kako bi bilo još detaljnije i transparentnije, dodano je automatsko brisanje dozvola nekorištenih aplikacija, ograničeno skupljanje informacija u pozadini. Što se tiče površinskih promjena unaprijeđen je sustav prikaza obavijesti, medija, dodana je mogućnost povijesnog prikaza obavijesti, dodana je opcija snimanja ekrana i unaprijeđeno upravljanje spojenih uređaja. [19]

### **2.2.2.18. Android 12**

U listopadu 2021. objavljena je verzija 12. U odnosu na posljednjih nekoliko verzija ova je donijela pretežito površinske promjene vezana za dizajn. Google je ovom verzijom osmislio Material You, svoju verziju Material Design standarda. Material You donio je potpuno novi izgled i osjećaj korištenja Androida kao i Googleovih proizvoda općenito. Cilj ovog standarda je da se korisniku da kontrola nad izgledom operacijskog sustava uvođenjem odabira paleta boja, vrste fonta kao i dinamičke stilizacije sustava temeljem palete pozadinske fotografije. Uz redizajn površinskog sloja dodane su i brojni noviteti i optimizacije. Redizajniran je način korištenja widgeta, unaprijeđene su performanse, sigurnost i privatnost. [20]

### **2.2.2.19. Android 13**

U kolovozu 2022. objavljena je verzija 13 koja je ujedno jedna od najambicioznijih i najsuptilnijih verzija do sad. Koji krajnost će korisnik doživjeti ovisi o tipu uređaja koji koristi. Ova verzija donosi potpuno novi izgled korisničkog sučelja za korisnike tableta i preklopnih pametnih telefona. Stavlja se fokus na poboljšanje lakoće korištenja uređaja velikih ekrana uvođenjem novog razvojnog okvira (eng. *Framework*), uputa za optimizaciju aplikacija i poboljšanja funkcije multitaskinga. Za korisnike klasičnih pametnih telefona ova verzija donosi većinom inkrementalna poboljšanja performansi, optimizacije i privatnosti uz dodavanje poneke funkcionalnosti kao „native“ mogućnosti skeniranja QR koda i mogućnosti uređivanja kopiranog teksta. [21]

### **2.2.2.20. Android 14**

Prva verzija dostupna razvojnim programerima objavljena je u veljači 2023., a prve verzije dostupne široj populaciji očekuju se početkom jeseni. Prema prvim podacima uz standardno inkrementalno poboljšanje performansa, sigurnosti i privatnosti Google razmatra uvođenje opcije dvojnih aplikacija koja bi korisnicima omogućila korištenje dvije verzije iste aplikacije istovremeno. Ova opcija primjereno je namijenjena razdvajanju privatnog od poslovnog, odnosno odvajanju poslovnog od privatnog identiteta na istom uređaju. [22]

## 3. Android ekosustav

Android ekosustav je pojam koji se odnosi na međuzavisnost korisnika, proizvođača opreme i razvojnih programera. Kako bi mogli pričati o razvoju aplikacija za Android trebamo poznavati i razumijete Android ekosustav i njegove međuzavisnosti. Sustav se temelji na međuovisnosti između Googlea, proizvođača opreme (pametnih uređaja) i korisnika gdje nijedan ne može opstati bez drugog. Android ekosustav baziran je oko Googleove trgovine mobilnih aplikacija koja uključuje aplikacije i korisnike. [23], [24]

### 3.1. Komponente Android ekosustava

Android ekosustav je skup fizičkih uređaja, aplikacija i servisa koji zajedno tvore Android operacijski sustav koji svi znamo. U nastavku su pobrojane i opisane komponente.

#### 3.1.1. Google

Google je prvi i najbitniji dio ovog ekosustava. On je razvio Android i pružio platformu za razvoj aplikacija. Google sakuplja korisničke podatke koje su korisnici aplikacija i samog Androida pristali podijeliti. Sakuplja podatke o povijesti pretraživanja, podatke o korištenju aplikacija, uređajima i slično. Svrha skupljanja podataka je dobiti kompletnu sliku trenutnog stanja onoga što korisnici koriste, što smatraju dobrim, što im je potrebno kako bi mogli pružiti što bolje usluge i proizvode. [24]

#### 3.1.2. Proizvođači originalne opreme

Proizvođači originalne opreme (eng. *Original Equipment Manufacturer*, OEM) proizvode fizičke uređaje na kojima je instaliran Android operacijski sustav i aplikacije. Iako je svima prva pomisao pametni telefon, Android je prisutan na mnogo različitih uređaja. Android je operacijski sustav koji se koristi na pametnim satovima i televizorima, tabletima, bežičnim zvučnicima i IoT uređajima kao pametnih termostata, digitalnih okvira za slike i slično. Bez proizvođača koji proizvode uređaje koji podržavaju Android operacijski sustav i koji zajedno sa Googleom rade na tome da korisnicima osiguraju najbolje iskustvo korištenja Androida cijeli ekosustav ne bi funkcionirao. [24]

### **3.1.3.Razvojni programeri**

Razvojni programeri kao pojedinci ili unutar razvojnih tvrtki pridonose ekosustavu proizvodnjom aplikacija koji sustav čine većim i raznovrsnijim. Svojom postojanjem osiguravaju zaposlenje velikog broja ljudi, pridonose iz dana u dan sve većem katalogu Android aplikacija što besplatnih što plaćenih. Osiguravaju zadržavanje korisnika na Androidu proizvodnjom korisnih, zabavnih i neophodnih aplikacija. Android kao operacijski sustav bez podrške aplikacija raznih kategorija ne bi imao smisla i ne bi bio gdje je danas. [24]

### **3.1.4.Korisnici**

Android korisnici su ljudi koji kupuju fizičke uređaje sa Android operacijskim sustavom, ljudi koji koriste besplatne i kupljene aplikacije i veliki dio ekosustava. Procjenjuje se da Android ima 2,8 milijardi korisnika. Ta nevjerojatna brojka govori o uspjehu Androida i njegovim pozitivnim atributima. Upravo sloboda koju pruža Android i velika količina uređaja koji ga koriste zajedno sa velikim brojem aplikacija i proizvođača čine ga primamljivim ovolikom broju ljudi bez kojih ovaj ekosustav ne bi postojao. [24]

## 4. Razvoj Android aplikacije

Kada govorimo o razvoju Android aplikacije mislimo na dizajniranje i programiranje programa na nekom programskom jeziku koji se pokreće na operacijskom sustavu Android. Iako se za razvoj Android aplikacija mogu koristiti razni programski jezici pretežito se koriste dva, Kotlin i Java. Od ta dva jezika Google je 2019. fokus stavio na Kotlin kao de facto standardni programski jezik za razvoj „native“ Android aplikacija. [25], [26]

### 4.1. Alati za razvoj

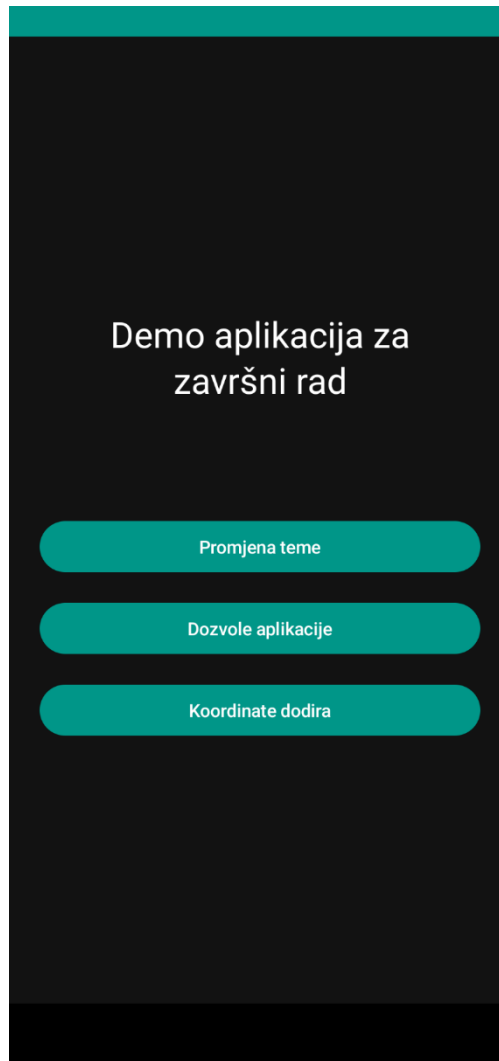
Kako bi mogli razvijati Android aplikacije potreban nam je Android komplet za razvoj softvera (eng. *Software development kit*, kraće SDK). SDK je set razvojnih alata koji uključuje program za ispravljanje pogrešaka (eng. *Debugger*), softverske biblioteke (eng. *Software libraries*), emulator, dokumentaciju, testni kod i vodiče. Sljedeća stvar je programski jezik koji Android podržava, kako je Kotlin posljednjih godina postao industrijski standard u ovom projektu je korišten on. Također nam je potrebno razvojno okruženje koje će koristiti SDK, ali 2014. godine Google nam je olakšao razvoj aplikacija predstavljanjem Android Studia. [26], [27]

Android Studio je integrirano razvojno okruženje (eng. *Integrated Development Environment*, kraće IDE) bazirano na IntelliJ IDEA. Android Studio objedinjuje Android SDK, programski jezik Kotlin, podršku za JetPack Compose i ostale potrebne alate za razvoj Android aplikacije. [28]

U ovom projektu su za razvoj aplikacije korišteni Android Studio Giraffe 2022.3.1., programski jezik Kotlin i Jetpack Compose, alat za izradu modernog „native“ UI-a

### 4.2. Razvijena aplikacija

U svrhu prikaza naprednih mogućnosti operacijskog sustava Android i ovog završnog rada napravio sam android aplikaciju koristeći gore navedene alate. Ova aplikacija koristi Material 3 dizajn, Jetpack Compose za izradu modernog UI-a, ugrađene Android biblioteke i dva Google programska sučelja aplikacija (eng. *Application Programming Interface*, kraće API) za ostvarenje funkcionalnosti.



Slika 3: Početni ekran aplikacije

Aplikacija se sastoji od 3 funkcionalnosti do kojih se dolazi preko početnog ekrana sa slike 3. Za navigaciju kroz aplikaciju koristio sam Navigation Compose koji omogućuje laganu i intuitivnu implementaciju navigacije u Jetpack Compose aplikacijama. Navigacija je ostvarena kroz zatvorenu klasu u `Navigator.kt` u kojoj su navedeni moguće rute navigacije:

```
sealed class Ekran(val route: String) {  
    object Home : Ekran("pocetniEkran")  
    object PromjenaTeme : Ekran("promjenaTeme")  
    object Dozvole : Ekran("dozvole")  
    object KoordinateDodira : Ekran("koordinataDodira")  
}
```

Klasa `MainActivity` je prva koja se pokreće kod pokretanja aplikacije te se u njoj deklariraju tri potrebne varijable i kreira `NavHost` koji uz navedene podatke kreira navigacijski graf aplikacije i postavlja aplikaciju na navedeni composable. U nastavku je programski kod klase `MainActivity`:

```

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            val navController = rememberNavController()
            val odabranaTema = rememberSaveable {
mutableStateOf(Teme.System) }
            val koordinataDodira = remember { mutableStateOf(Offset(0f,
0f)) }

            DemoZavršniRadTheme(odabranaTema.value) {
                NavHost(navController = navController, startDestination =
Ekran.Home.route) {
                    composable(route = Ekran.Home.route) {
                        PocetniEkran(navController)
                    }
                    composable(route = Ekran.PromjenaTeme.route) {
                        PromjenaTemeEkran(odabranaTema, navController)
                    }
                    composable(route = Ekran.Dozvole.route) {
                        DozvoleEkran(navController)
                    }
                    composable(route = Ekran.KoordinataDodira.route) {
                        KoordinataDodiraEkran(navController,
koordinataDodira)
                    }
                }
            }
        }
    }
}

```

U gore navedenom programskom kodu početna destinacija postavljena je na `startDestination = Ekran.Home.route` koja pokreće `composable` funkciju `PocetniEkran`:

```

@Composable
fun PocetniEkran(navController: NavController) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colorScheme.background)
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Demo aplikacija za završni rad",
            style = MaterialTheme.typography.headlineMedium,
            color = MaterialTheme.colorScheme.onBackground,
            modifier = Modifier.padding(16.dp),
            textAlign = TextAlign.Center
        )
        Spacer(modifier = Modifier.height(64.dp))
        GumbPocetna("Promjena teme", navController, "promjenaTeme")
        GumbPocetna("Dozvole aplikacije", navController, "dozvole")
        GumbPocetna("Koordinate dodira", navController, "koordinataDodira")
    }
}

```

U ovoj funkciji nalazi se izbornik koji se sastoji od tri gumba, po jedan za svaku funkcionalnost. Kako bi dizajn gumbova bio dosljedan, kod za njih odvojen je u composable funkciju `GumbPocetna`.

Programski kod composable funkcije `GumbPocetna`:

```
@Composable
fun GumbPocetna(text: String, navController: NavController, route: String)
{
    Button(
        onClick = { navController.navigate(route) },
        modifier = Modifier
            .fillMaxWidth()
            .padding(8.dp)
    ) {
        Text(text)
    }
}
```

### 4.2.1. Funkcija promjene teme

Funkcionalnost promjene teme aplikacije omogućava korisniku odabir jedne od tri opcija, tamna, svijetla i tema sustava kao što je prikazano na slici 4. Svijetla tema prikazana je na slici 5, a tamna na slici 6. Tema sustava ovisi o opciji pametnog telefona o tome koristi li svijetli ili tamni način te sukladno tome postavlja temu aplikacije na svijetlu ili tamnu.






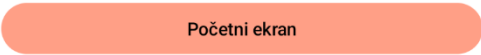
## Funkcija promjene teme

Taman tema


Svijetla tema

Tema sustava

Početni ekran



Slika 4: Ekran funkcionalnosti promjene teme




## Demo aplikacija za završni rad

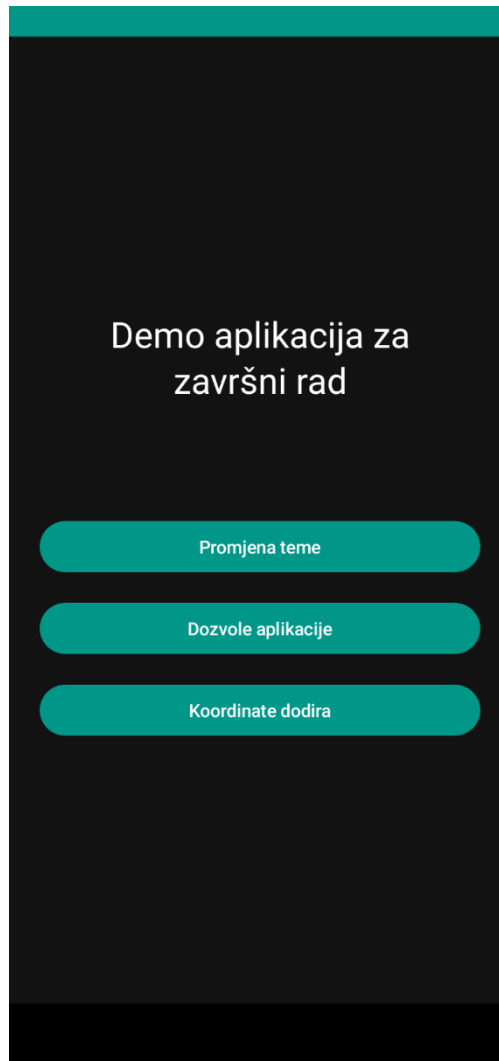
Promjena teme

Dozvole aplikacije

Koordinate dodira



Slika 5: Početni ekran u svijetloj temi



Slika 6: Početni ekran u tamnoj temi

Funkcionalnost promjene teme ostvarena je composable funkcijom `PromjenaTemeEkran`. Ova funkcija kreira tri gumba, po jedan za svaku opciju teme. Klikom na gumb vrijednost parametar `odabranaTema` postavlja se na odgovarajuću vrijednost enumeracijske klase `Tema`. Kako bi odabrana tema bila primjenjena na cijelu aplikaciju ona se prosljeđuje composable funkciji `DemoZavršniRadTheme` koja se pokreće prva unutar klase `MainActivity` te određuje temu aplikacije. `DemoZavršniRadTheme` na temelju dva parametra `odabranaTema` i `tamniNacin` postavlja vrijednost varijable `paletaBoja`. Vrijednost varijable prosljeđuje se javnoj composable funkciji `MaterialTheme` koja je predefinirana funkcija koju koristimo kako bi aplikacija slijedila Material stil dizajna. U nastavku je prikaza programski kod composable funkcija `PromjenaTemeEkran` i `DemoZavršniRadTheme`.

```

@Composable
fun PromjenaTemeEkran(odabranaTema: MutableState<Teme>, navController:
NavController) {
    var tamnaTema by remember { mutableStateOf(false) }
    var svijetlaTema by remember { mutableStateOf(false) }
    var systemskaTema by remember { mutableStateOf(false) }
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colorScheme.background),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Funkcija promjene teme",
            style = MaterialTheme.typography.headlineMedium,
            color = MaterialTheme.colorScheme.onBackground,
            modifier = Modifier.padding(16.dp),
            textAlign = TextAlign.Center
        )

        Spacer(modifier = Modifier.height(64.dp))

        Button(
            onClick =
            {
                odabranaTema.value = Teme.Dark
                tamnaTema = true
                svijetlaTema = false
                systemskaTema = false
            },
            enabled = !tamnaTema)
        {
            Text(text = "Taman tema")
        }
        Button(
            onClick =
            {
                odabranaTema.value = Teme.Light
                tamnaTema = false
                svijetlaTema = true
                systemskaTema = false
            },
            enabled = !svijetlaTema)
        {
            Text(text = "Svijetla tema")
        }
        Button(
            onClick =
            {
                odabranaTema.value = Teme.System
                tamnaTema = false
                svijetlaTema = false
                systemskaTema = true
            },
            enabled = !systemskaTema)
        {
            Text(text = "Tema sustava")
        }

        Spacer(modifier = Modifier.height(64.dp))
    }
}

```

```

        Button(
            onClick = { navController.popBackStack() },
            modifier = Modifier
                .fillMaxWidth()
                .padding(8.dp)
        ) {
            Text(text = "Početni ekran")
        }
    }
}

@Composable
fun DemoZavršniRadTheme(
    odabranaTema: Tema,
    tamniNacin: Boolean = isSystemInDarkTheme(),
    content: @Composable () -> Unit
) {
    val paletaBoja = when (odabranaTema) {
        Tema.Dark -> DarkColorScheme
        Tema.Light -> LightColorScheme
        else -> {
            if (tamniNacin) DarkColorScheme else LightColorScheme
        }
    }

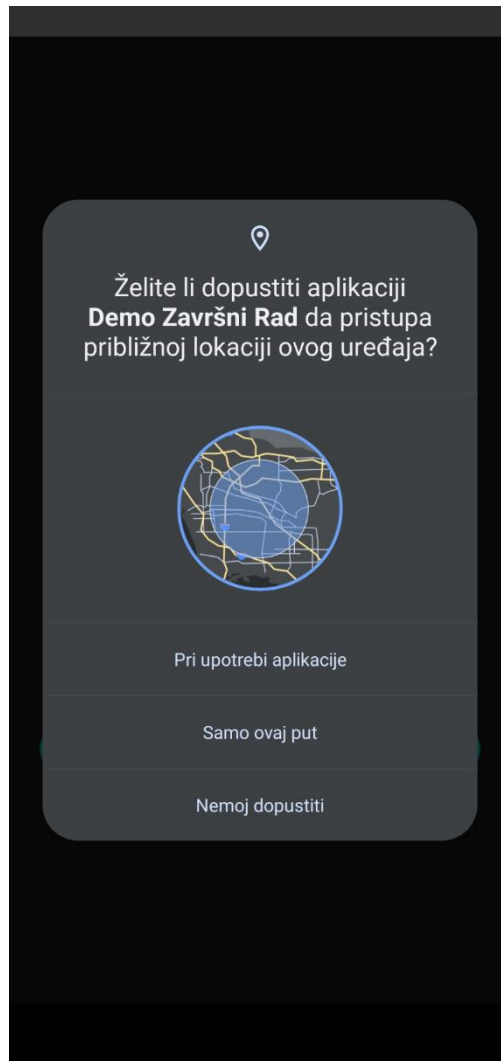
    MaterialTheme(
        colorScheme = paletaBoja,
        typography = Typography,
        content = content
    )
}

```

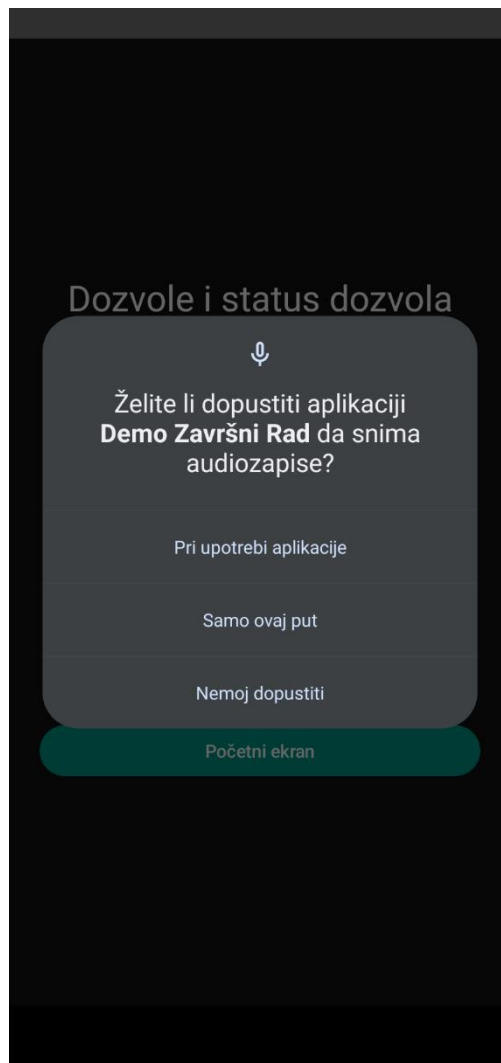
## 4.2.2. Funkcija traženja i prikaza stanja dozvola

Ova funkcionalnost korisnika aplikacije traži dozvole za pristup informacijama uređaja, konkretno dozvolama za lokaciju, stanje mreže i pristup mikrofONU. Dozvole koje aplikacija traže čisto su demonstracijske prigode i moguće ih je jednostavno promijeniti da aplikacija traži druge što pokazuje jednostavnost koju Android pruža. Za traženje i upravljanje dozvolama koristio sam Googleov Permissions API koji omogućuje traženje više dozvola istovremeno i prikaz njihovog stanja kako bih demonstrirao što se događa kada jednostavno prihvatimo ili odbijemo aplikaciji pristup neki podacima uređaja.

Klikom na gumb koji vodi na ovu funkcionalnost sa početnog ekrana pokreće se composable funkcija `DozvoleEkran` koja kod prvog korištenja od korisnika zatraži dozvole koje su ranije navedene. Svaka dozvola ima tri moguća stanja vidljive na slikama 7 i 8, davanje dozvole dok se koristi aplikacija, davanje dozvole samo ovaj put i odbijanje. Nakon odabira željenih postavki dozvola korisniku se prikazuje ekran sa slike 9 sa popisom dozvola koje aplikacija ima i njihovim stanjem. Ako je dozvola odbijena jednom prikazuje se razlog zbog kojeg je ona potrebna. Ako korisnik kod sljedećeg korištenja aplikacije ponovno odbije istu dozvolu ona se odbija trajno što znači da aplikacija više neće tražiti dozvolu te ju korisnik mora ručno prihvatiti u postavkama privatnosti uređaja.

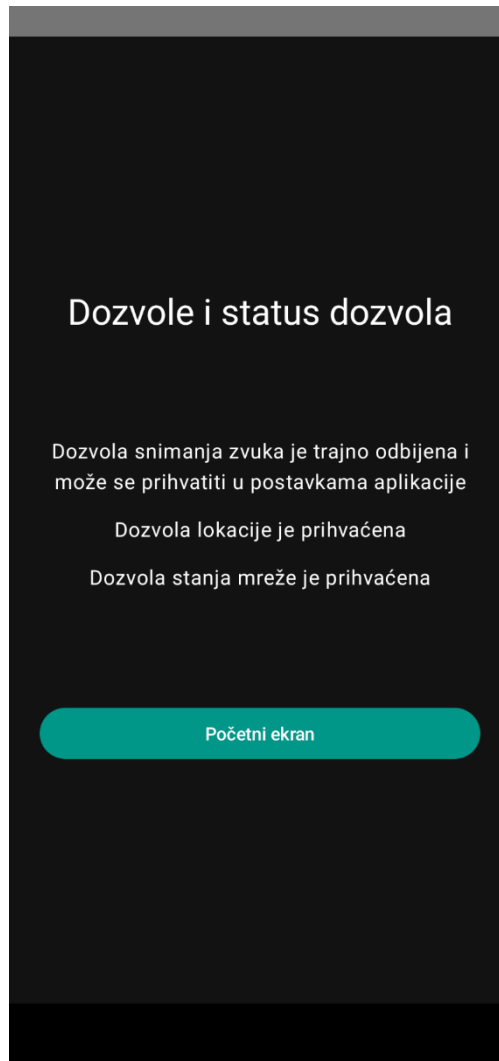


Slika 7: Traženje dozvola približne lokacije



Slika 8: Traženje dozvola snimanja audio zapisa





Slika 9: Popis i stanje dozvola aplikacije

Na početku composable funkcije `DozvoleEkran` definirana je lista dozvola koja sprema podatke o njihovom stanju:

```
val stanjeDozvola = rememberMultiplePermissionsState(  
    permissions = listOf(  
        Manifest.permission.RECORD_AUDIO,  
        Manifest.permission.ACCESS_COARSE_LOCATION,  
        Manifest.permission.ACCESS_NETWORK_STATE  
    )  
)
```

Traženje ovih dozvola odvija se pokretanje funkcionalnosti sa početnog ekrana i ono se odvija svakim pokretanjem aplikacije kako bi osigurali da ako korisnik odabere opciju davanja dozvole samo jednom tu istu dozvolu uskratili gašenje aplikacije i ponovno ju zatražili sljedeći put. Za implementaciju ove funkcionalnosti koristio sam ugrađenu funkciju `DisposableEffect` koja svakim pokretanjem aplikacije provjerava stanje dozvola i ako se ono promijenilo u odnosu na prethodno zapamćeno stanje ponovno pokreće traženje dozvola čije stanje je promijenjeno.

Kod ispisa provjerava se svaka dozvola i ovisno o njenom stanju ispisuje se određeni tekst. U nastavku je cijeli programski kod composable funkcije `DozvoleEkran`.

```

@OptIn(ExperimentalPermissionsApi::class)
@Composable
fun DozvoleEkran(navController: NavController) {

    val stanjeDozvola = rememberMultiplePermissionsState(
        permissions = listOf(
            Manifest.permission.RECORD_AUDIO,
            Manifest.permission.ACCESS_COARSE_LOCATION,
            Manifest.permission.ACCESS_NETWORK_STATE
        )
    )

    val lifecycleOwner = LocalLifecycleOwner.current
    DisposableEffect(key1 = lifecycleOwner,
        effect = {
            val observer = LifecycleEventObserver { _, event ->
                if (event == Lifecycle.Event.ON_START) {
                    stanjeDozvola.launchMultiplePermissionRequest()
                }
            }
            lifecycleOwner.lifecycle.addObserver(observer)

            onDispose {
                lifecycleOwner.lifecycle.removeObserver(observer)
            }
        })

    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colorScheme.background)
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Dozvole i status dozvola",
            style = MaterialTheme.typography.headlineMedium,
            color = MaterialTheme.colorScheme.onBackground,
            modifier = Modifier.padding(16.dp),
            textAlign = TextAlign.Center
        )

        Spacer(modifier = Modifier.height(64.dp))

        stanjeDozvola.permissions.forEach { perm ->
            when (perm.permission) {
                Manifest.permission.RECORD_AUDIO -> {
                    when {
                        perm.hasPermission -> {
                            Text(
                                text = "Dozvola snimanja zvuka je
prijhvaćena",
                                color =
MaterialTheme.colorScheme.onBackground,
                                textAlign = TextAlign.Center
                            )
                        }
                    }
                }
            }
        }
    }
}

```

```

        perm.shouldShowRationale -> {
            Text(
                text = "Dozvola snimanja zvuka je potrebna
za testiranje",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }

        perm.isPermanentlyDenied() -> {
            Text(
                text = "Dozvola snimanja zvuka je trajno
odbijena i može se prihvatiti u postavkama aplikacije",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }
    }
    Spacer(modifier = Modifier.height(16.dp))
}

Manifest.permission.ACCESS_COARSE_LOCATION -> {
    when {
        perm.hasPermission -> {
            Text(
                text = "Dozvola lokacije je prihvaćena",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }

        perm.shouldShowRationale -> {
            Text(
                text = "Dozvola lokacije je potrebna za
testiranje",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }

        perm.isPermanentlyDenied() -> {
            Text(
                text = "Dozvola lokacije je trajno odbijena
i može se prihvatiti u postavkama aplikacije",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }
    }
    Spacer(modifier = Modifier.height(16.dp))
}

```

```

Manifest.permission.ACCESS_NETWORK_STATE -> {
    when {
        perm.hasPermission -> {
            Text(
                text = "Dozvola stanja mreže je
prihvaćena",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }
        perm.shouldShowRationale -> {
            Text(
                text = "Dozvola stanja mreže je potrebna za
testiranje",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }
        perm.isPermanentlyDenied() -> {
            Text(
                text = "Dozvola stanja mreže je trajno
odbijena i može se prihvatiti u postavkama aplikacije",
                color =
MaterialTheme.colorScheme.onBackground,
                textAlign = TextAlign.Center
            )
        }
    }
    Spacer(modifier = Modifier.height(16.dp))
}
}

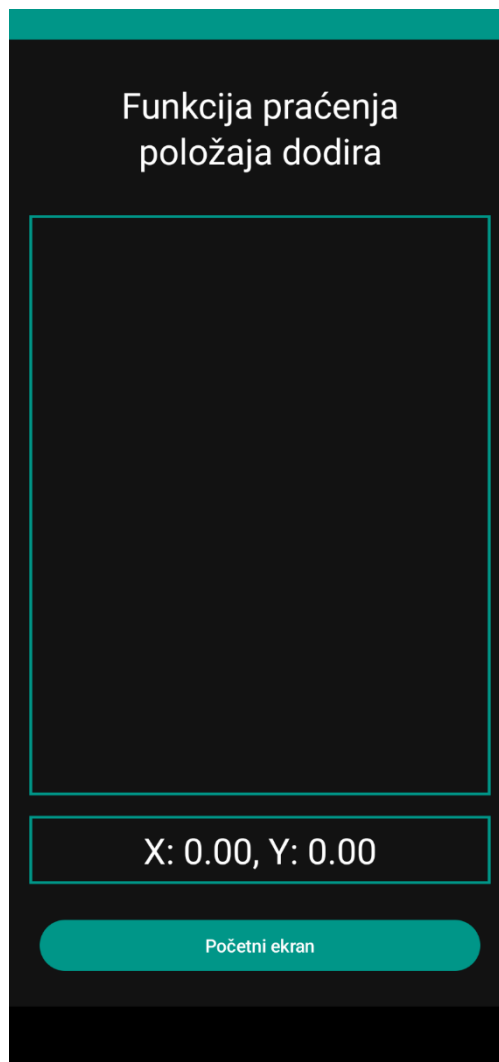
Spacer(modifier = Modifier.height(64.dp))

Button(
    onClick = { navController.popBackStack() },
    modifier = Modifier
        .fillMaxWidth()
        .padding(8.dp)
) {
    Text(text = "Početni ekran")
}
}
}

```

### 4.2.3. Funkcija prikaza koordinata dodira

Funkcionalnost prikaza koordinata korisnikovog dodira koncipirana je tako da se na ekranu, vidljivom na slici 10, nalazi okvir unutar kojeg se prati položaj dodira i ispisa trenutnih X i Y koordinata dodira. Koordinate se računaju od gornjeg lijevog kuta okvira odnosno koordinate gornjeg lijevog kuta okvira su X,Y(0,0). Kretanjem horizontalno mijenja se vrijednost X koordinate, pozitivno u desno i negativno u lijevo. Kretanjem vertikalno mijenja se vrijednost Y koordinate, pozitivno prema dolje i negativno prema gore.



Slika 10: Ekran funkcionalnosti praćenja položaja dodira

Ova funkcionalnost implementirana je pomoću Google Compose UI API koji omogućuje dohvat podataka o položaju korisnikovog dodira unutar compose elementa, u ovom slučaju elementa Box. Unutar ugrađene funkcije `pointerInteropFilter` prati se događaj dodira i ovisno o vrsti događaja mijenja se vrijednost `koordinataDodira`. U nastavku je prikazan cijeli programski kod composable funkcije `KoordinateDodiraEkran`.

```

@OptIn(ExperimentalComposeUiApi::class)
@Composable
fun KoordinateDodiraEkran(navController: NavController, koordinataDodira:
MutableState<Offset>) {
    Column(
        modifier = Modifier
            .fillMaxSize()
            .background(MaterialTheme.colorScheme.background)
            .padding(16.dp),
        verticalArrangement = Arrangement.Center,
        horizontalAlignment = Alignment.CenterHorizontally
    ) {
        Text(
            text = "Funkcija praćenja položaja dodira",
            style = MaterialTheme.typography.headlineMedium,
            color = MaterialTheme.colorScheme.onBackground,
            modifier = Modifier.padding(16.dp),
            textAlign = TextAlign.Center
        )

        Spacer(modifier = Modifier.height(16.dp))

        Box(
            modifier = Modifier
                .fillMaxSize()
                .weight(1f)
                .border(
                    width = 2.dp,
                    color = MaterialTheme.colorScheme.primary,
                    shape = RectangleShape
                )
            .pointerInteropFilter { event ->
                when (event.action) {
                    MotionEvent.ACTION_DOWN -> {
                        koordinataDodira.value = Offset(event.x,
event.y)

                        true
                    }

                    MotionEvent.ACTION_MOVE -> {
                        koordinataDodira.value = Offset(event.x,
event.y)

                        true
                    }

                    MotionEvent.ACTION_UP -> {
                        koordinataDodira.value = Offset(0f, 0f)
                        true
                    }

                    else -> false
                }
            }
        )

        Spacer(modifier = Modifier.height(16.dp))
    }
}

```

```

Box(
    modifier = Modifier
        .fillMaxWidth()
        .border(
            width = 2.dp,
            color = MaterialTheme.colorScheme.primary,
            shape = RectangleShape
        )
) {
    Text(
        text = "X: %.2f, Y: %.2f".format(koordinataDodira.value.x,
koordinataDodira.value.y),
        style = MaterialTheme.typography.headlineMedium,
        color = MaterialTheme.colorScheme.onBackground,
        modifier = Modifier
            .padding(8.dp)
            .align(Alignment.Center)
    )
}

Spacer(modifier = Modifier.height(16.dp))

Button(
    onClick = { navController.popBackStack() },
    modifier = Modifier
        .fillMaxWidth()
        .padding(8.dp)
) {
    Text(text = "Početni ekran")
}
}
}

```

## 5. Zaključak

Korištenjem besplatnog razvojnog okruženja Android Studio, programskog jezika Kotlin i dva besplatna API-ja tvrtke Google kreirana je aplikacija koja prikazuje tri napredne funkcionalnosti operacijskog sustava Android. Aplikacija prikazuje funkcionalnosti promjene teme cijele aplikacije, traženja i prikaza statusa dozvola aplikacije i funkcionalnost praćenja i prikaza trenutnog položaja korisnikovog dodira. Kroz implementaciju ovih funkcionalnosti prikaz je širok spektar mogućnosti operacijskog sustava Android od praćenja korisnikovih ulaznih podataka (dodira) do upravljanja izgledom.

Kroz funkcionalnosti promjene teme prikazana je mogućnost prilagođavanja koja je jedna od glavnih prednosti Androida. Mogućnost da korisnik odabere način na koji se aplikacija prikazuje, promijeni cjelokupnu temu aplikacije. Također je vidljiva lakoća kojom razvojni programeri mogu omogućiti korisniku tu mogućnost u svega nekoliko linija koda.

Kroz funkcionalnost traženja i prikaza statusa dozvola prikazana je velika kontrola i transparentnost koju Android dopušta korisniku u pitanju privatnosti. Razvojni programeri mogu od korisnika zatražiti sve moguće dozvole koje su im potrebne za kreiranje i ispravan rad aplikacije, ali korisnik ima potpunu kontrolu nad dozvolama i njihovim stanjima. Korisnik odlučuje koje dozvole je spreman ustupiti i kada te time osigurava svoju sigurnost i privatnost.



## Popis literature

1. Statcounter, „Operating System Market Share Worldwide - August 2023“, 2023. [Na internetu]. Dostupno: <https://gs.statcounter.com/os-market-share/mobile/worldwide> [pristupano 26.08.2023.]
2. Statcounter, „Mobile Operating System Market Share Worldwide - August 2023“, 2023. [Na internetu]. Dostupno: <https://gs.statcounter.com/os-market-share> [pristupano 26.08.2023.]
3. Phone Arena, „Google's Android OS: Past, Present, and Future“, 2011. [Na internetu]. Dostupno: [https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future\\_id21273](https://www.phonearena.com/news/Googles-Android-OS-Past-Present-and-Future_id21273) [pristupano 28.08.2023.]
4. J. Alabaster, „Android founder: We aimed to make a camera OS“, 2013. [Na internetu]. Dostupno: <https://www.pcworld.com/article/451350/android-founder-we-aimed-to-make-a-camera-os.html> [pristupano 29.08.2023.]
5. A. McAfee i E. Brynjolfsson, *Machine, Platform, Crowd: Harnessing Our Digital Future*. New York, W. W. Norton & Company, Inc. 2017.
6. Open Handset Alliance, (bez dat.) Android [Na internetu]. Dostupno: [https://www.openhandsetalliance.com/android\\_overview.html](https://www.openhandsetalliance.com/android_overview.html) [pristupano 29.08.2023.]
7. Google, (bez dat.) Android [Na internetu]. Dostupno: <https://www.android.com/> [pristupano 29.08.2023.]
8. Android Open Source Project, (bez dat.) [Na internetu]. Dostupno: <https://source.android.com/> [pristupano 29.08.2023.]
9. Android Open Source Project, (bez dat.) Frequently asked questions [Na internetu]. Dostupno: <https://source.android.com/docs/setup/about/faqs> [pristupano 29.08.2023.]
10. A. Sareen, „Features of Android Operating System“, 2023. [Na internetu]. Dostupno: <https://www.codingninjas.com/studio/library/features-of-android-operating-system> [pristupano 29.08.2023.]
11. J. R. Raphael, „Android versions: A living history from 1.0 to 14“, 2023. [Na internetu]. Dostupno: <https://www.computerworld.com/article/3235946/android-versions-a-living-history-from-1-0-to-today.html> [pristupano 27.08.2023.]
12. Google, (bez dat.) Android KitKat [Na internetu]. Dostupno: <https://developer.android.com/about/versions/kitkat> [pristupano 28.08.2023.]
13. Google, (bez dat.) Android Lollipop [Na internetu]. Dostupno: <https://developer.android.com/about/versions/lollipop> [pristupano 28.08.2023.]

14. Google, (bez dat.) Android 6.0 Marshmallow [Na internetu]. Dostupno: <https://developer.android.com/about/versions/marshmallow> [pristupano 28.08.2023.]
15. Google, (bez dat.) Android 7.0 Nougat [Na internetu]. Dostupno: <https://developer.android.com/about/versions/nougat> [pristupano 28.08.2023.]
16. Google, (bez dat.) Android Oreo [Na internetu]. Dostupno: <https://developer.android.com/about/versions/oreo> [pristupano 28.08.2023.]
17. Google, (bez dat.) Android 9 Pie [Na internetu]. Dostupno: <https://developer.android.com/about/versions/pie> [pristupano 28.08.2023.]
18. Google, (bez dat.) android 10 [Na internetu]. Dostupno: <https://developer.android.com/about/versions/10> [pristupano 29.08.2023.]
19. Google, (bez dat.) android 11 [Na internetu]. Dostupno: <https://developer.android.com/about/versions/11> [pristupano 29.08.2023.]
20. Google, (bez dat.) android 12 [Na internetu]. Dostupno: <https://developer.android.com/about/versions/12> [pristupano 29.08.2023.]
21. Google, (bez dat.) android 13 [Na internetu]. Dostupno: <https://developer.android.com/about/versions/13> [pristupano 29.08.2023.]
22. Google, (bez dat.) android 14 Beta [Na internetu]. Dostupno: <https://developer.android.com/about/versions/14> [pristupano 29.08.2023.]
23. Google, (bez dat.) Better together [Na internetu]. Dostupno: <https://www.android.com/better-together/> [pristupano 29.8.2023.]
24. GeeksforGeeks, „Android Ecosystem“, 2023. [Na internetu]. Dostupno: <https://www.geeksforgeeks.org/android-ecosystem/> [pristupano 01.09.2023.]
25. Google, (bez dat.) Android for Developers [Na internetu]. Dostupno: <https://developer.android.com/> [pristupano 26.08.2023.]
26. P. Ruhil, „Introduction to Android Development“, 2023. [Na internetu]. Dostupno: <https://www.geeksforgeeks.org/introduction-to-android-development/#article-meta-div> [pristupano 26.08.2023.]
27. Google, (bez dat.) What is modern Android development [Na internetu]. Dostupno: <https://developer.android.com/modern-android-development> [pristupano 27.08.2023.]
28. Google, (bez dat.) Android Studio [Na internetu]. Dostupno: <https://developer.android.com/studio> [pristupano 27.08.2023.]

# Popis slika

Slika 1: Tržišni udio Android operacijskog sustava (mobilni) [1].....	2
Slika 2: Tržišni udio Android operacijskog sustava (općenito) [2].....	3
Slika 3: Početni ekran aplikacije .....	15
Slika 4: Ekran funkcionalnosti promjene teme .....	18
Slika 5: Početni ekran u svijetloj temi.....	19
Slika 6: Početni ekran u tamnoj temi.....	20
Slika 7: Traženje dozvola približne lokacije.....	24
Slika 8: Traženje dozvola snimanja audio zapisa.....	25
Slika 9: Popis i stanje dozvola aplikacije.....	26
Slika 10: Ekran funkcionalnosti praćenja položaja dodira .....	30