PROGRAMSKO RJEŠENJE ZA UPRAVLJANJE ODABIROM ŽELJENIH POKLONA

Kolonić, Luka

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Algebra University College / Visoko učilište Algebra**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:225:024486

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Algebra Univerity College - Repository of Algebra Univerity College

VISOKO UČILIŠTE ALGEBRA

ZAVRŠNI RAD

PROGRAMSKO RJEŠENJE ZA UPRAVLJANJE ODABIROM ŽELJENIH POKLONA

Luka Kolonić

Zagreb, veljača 2023.

Predgovor

Svome mentoru Đaniju Pašiću, mag. ing. comp., iskazujem duboku zahvalnost za ukazano povjerenje i podršku tijekom pisanja ovog rada. Vaše vodstvo i savjeti značajno su pridonijeli uspjehu ovog rada. Također, želim se zahvaliti mojoj obitelji na neizmjernoj podršci i razumijevanju. Posebna zahvala ide mojoj dragoj Matei koja mi je pružila neprekidno strpljenje, motivaciju i potporu.

Prilikom uvezivanja rada, Umjesto ove stranice ne zaboravite umetnuti original potvrde o prihvaćanju teme završnog rada kojeg ste preuzeli u studentskoj referadi

Sažetak

U ovom radu, fokus je bio na stvaranju Android mobilne aplikacije za darivanje poklona korištenjem programskog jezika Kotlin, Android Studija te baze podataka SQLite. Prije samog početka implementacije, obavljeno je detaljno istraživanje tržišta, uključujući analizu direktnih i indirektnih konkurenata na tržištu.

Tijekom procesa implementacije, definirani su zahtjevi vezani za karakteristike i funkcionalnosti aplikacije, uključujući funkcije poput jednostavne navigacije do fizičkog prodajnog mjesta određenog poklona, kategoriziranog prikaza te unosa liste želja u obliku poklona, anonimnog chata i slično.

Kako bi se osigurala funkcionalnost i kvaliteta aplikacije, ključni dijelovi implementacije prikazani su uz detaljni prikaz koda. Aplikacija omogućuje korištenje jednostavnog sučelja za brzi i efikasan odabir poklona, čime se uvelike pojednostavljuje proces darivanja.

Cilj ovog rada bio je stvoriti jednostavnu i efikasnu aplikaciju za darivanje poklona, koja će korištenjem inovativnih funkcionalnosti olakšati proces odabira i pružiti korisnicima kvalitetno iskustvo.

Ključne riječi: proces darivanja, lista želja, iskustvo, konkurenti

Abstract

In this final thesis, the focus was on creating an Android mobile application for gift giving using Kotlin programming language, Android Studio and SQLite database. Before the very beginning of the implementation, a detailed market research was carried out, including an analysis of direct and indirect competitors on the market.

During the implementation process, requirements related to the characteristics and functionality of the application were defined, including functions such as simple navigation to the physical selling spot of a certain gift, categorized display and entry of a wish list in the form of a gift, anonymous chat and the rest.

In order to ensure the functionality and quality of the application, the key parts of the implementation were shown with a detailed view of the code. The use of a simple interface was enabled for quick and efficient gift selection, which greatly simplified the gift-giving process.

The goal of this thesis was to create a simple and efficient application for gift giving, which will facilitate the selection process and provide users with a quality experience by using innovative functionalities.

Key words: gifting process, wish list, experience, competitors

Sadržaj

1. Uvod 3

2. Trenutno stanje na tržištu 4

2.1. Izravni konkurenti 4

2.2. Neizravni konkurenti 5

3. Arhitektura aplikacije 8

3.1. Oblikovanje arhitekture 8

3.2. Korištene tehnologije 10

4. Implementacija programskog rješenja 14

4.1. Analiza zahtjeva 14

4.1.1. Zahtjevi vezani za funkcionalnost 15

4.1.2. Zahtjevi vezani uz karakteristike 18

4.2. Implementacija zahtjeva 18

4.2.1. Registracija i prijava u aplikaciju 19

4.2.2. Kategorije događaja 21

4.2.3. Kreiranje događaja 24

4.2.4. Kreiranje i odabir liste želja 25

4.2.5. Anonimni chat i statistika 27

5. Rezultati korištenja aplikacije 30

5.1. Upute za korištenje aplikacije 31

5.2. Inovacije s obzirom na postojeća rješenja 41

Zaključak 42

Popis kratica 44

Popis slika 45

Popis tablica..... 46

Popis kodova 47

Literatura 48

1. Uvod

Pokloni su dio svakodnevnog života mnogih ljudi te se često koriste kao način pokazivanja poštovanja i zahvalnosti prema voljenim osobama. Međutim, svake se godine ljudi susreću s problematikom odabira odgovarajućih poklona. Odabir istih može biti dugotrajan, stresan i kompliciran zadatak, osobito ako se dar poklanja dragim osobama. Stoga je programsko rješenje za upravljanje odabirom prikladnih poklona vrijedno istraživanja i razvoja. Prema istraživanju, ljudi koji su dobili iskustvene poklone iskazali su više zadovoljstva i sreće u odnosu na one koji su dobili materijalne stvari. Spomenuto zadovoljstvo se događa jer iskustva više stimuliraju osjećaje i emocije, što može dovesti do dugotrajnijeg zadovoljstva (Goodman, 2014).

Cilj izrade ovog rada je napraviti sustav za upravljanje odabirom željenih poklona koji će imati dalekosežan utjecaj na korisnike. Jednima će pomoći u odabiru priželjkivanih poklona za svaku prigodu, dok će drugima prikazati podatke o preferencijama slavljenika kako bi se odabralo nešto što će se svidjeti baš toj osobi. Sustav će se temeljiti na podacima koje korisnici unesu, te će se isti kroz aplikaciju analizirati i obrađivati. Jedna od glavnih vodilja ovog rada je i smanjenje utrošenog vremena za pronalazak kvalitetnog i prigodnog dara. Također, kroz rad će biti proučeni i analizirani postojeći sustavi za darivanje poklona te će se identificirati njihove prednosti i nedostaci. Na temelju dobivenih rezultata, bit će razvijen novi sustav koji će se temeljiti na najboljim praksama te će biti prilagođen korisnicima. Rad se sastoji od pet poglavlja. Poglavlje „Trenutno stanje na tržištu“ sadrži analizu postojećih sustava za darivanje poklona. Sljedeća dva poglavlja su posvećena opisu razvoja aplikacije, arhitekturi sustava te procesu implementacije. Zadnje poglavlje „Rezultat korištenja aplikacije“ prikazuje način na koji se aplikacija koristi, njenu učinkovitost, doprinose i inovacije.

Sama izrada ovog sustava imat će značajan utjecaj na olakšavanje procesa odabira poklona, što bi moglo doprinijeti pozitivnijem iskustvu korisnika u vezi darivanja materijalnih ili nematerijalnih stvari koji će biti izraz poštovanja i pažnje.

2. Trenutno stanje na tržištu

Trenutna ponuda domaćih i stranih mobilnih aplikacija na zadanu temu ne zadovoljava potrebe tržišta zbog manjka onih koje nisu definirane određenom prigodom. Postoji mnogo različitih aplikacija za darivanje i odabir poklona, svaka sa svojim specifičnim setom funkcija i mogućnosti. Neke od najpopularnijih aplikacija za darivanje poklona su Gifster, VOLO i Wishlistr. Spomenute aplikacije omogućavaju korisnicima da stvore listu želja, podijele istu s obiteljima i prijateljima, ocjene poklone te niz drugih mogućnosti. Također, na tržištu možemo pronaći par zanimljivih aplikacija kao što su Tinggly i SnapFish Gifts koje nude personalizaciju poklona te stavljanje fokusa na darivanje iskustva umjesto materijalnih stvari. U nastavku će se opisati izravni i neizravni konkurenti aplikacije PoklonManija o kojoj se piše u ovome radu.

2.1. Izravni konkurenti

Izravni konkurenti su tvrtke ili proizvodi koji se direktno natječu s drugim tvrtkama ili proizvodima u istom tržišnom segmentu. Analiza izravnih konkurenata je važna za određivanje ključnih faktora uspješnosti u poslovanju jer pomaže u identificiranju prednosti i slabosti u odnosu na konkurente. Za određivanje izravnih konkurenata, potrebno je uzeti u obzir nekoliko faktora poput: tipa proizvoda ili usluge, ciljane publike, cijene usluge, kanala distribucije te snage tržišnog položaja. U ovom slučaju, izravni konkurenti su sve srodne aplikacije koje nude prvenstveno mogućnost unosa želja slavljenika:

- Gifster

Gifster je jedna od najpopularnijih, a samim time i najveći izravni konkurent. Aplikacija je nastala 2009. godine u Sjedinjenim Američkim Državama. Gifster omogućuje korisnicima stvaranje privatnih i javnih popisa želja koje mogu biti podijeljene s prijateljima i obitelji. Najzanimljivija funkcionalnost aplikacije je stvaranje raznih grupa putem kojih je korisnicima omogućen pregled te kupnja željenih poklona. Također, njima je omogućeno dodavanje vlastitih želja u obliku linka koji vodi do odabranog proizvoda. Sustav automatski prepoznaje opis proizvoda, ime, cijenu te razne druge informacije. Danas, Gifster aplikacija broji preko milijun članova te je dostupna mobilna i web verzija aplikacije (Gifster, 2022).

- VOLO

VOLO wishlist je popularna web i mobilna aplikacija pomoću koje je olakšan odabir poklona svojim najdražima. Nudi mogućnost primanja i darivanja poklona za događaje poput rođendana i vjenčanja. Za božićne je blagdane predviđena kategorija „Pismo Djeda Mrazu“, dok je za najmlađu populaciju zamišljena kategorija „Za bebu“. VOLO se ističe od drugih sličnih aplikacija po svojem otvorenom savjetniku za darove gdje korisnici iste spremaju te predlažu drugima (VOLO, 2022).

- Wishlistr

Wishlistr je web aplikacija koja omogućuje korisnicima kreiranje i upravljanje popisom stvari koje žele kupiti ili darovati s bilo kojeg mjesta na webu jednostavnim klikom na gumb. Aplikacija korisnicima pruža opciju organiziranja stavki s popisa želja u odgovarajuće predloške za rođendane, vjenčanja, praznike te razne druge događaje. Listu želja moguće je jednostavno urediti, promijeniti, ukloniti, dodati opis te sliku. Praćenjem prijatelja na stranici olakšava se dijeljenje liste želja te primanje istih (wishlistr, 2022).

Spomenutim aplikacijama, kao i PoklonManiji, zajednička je funkcionalnost unos vlastitih korisničkih želja u obliku poklona ili iskustva. Nakon unosa želja, drugi korisnici mogu na razne načine pokloniti željeni dar. Upravo zbog navedenih funkcionalnosti možemo reći da su spomenute aplikacije izravni konkurenti jer se direktno natječu sa sličnim proizvodom koji ima istu namjenu u tržišnom segmentu.

2.2. Neizravni konkurenti

Neizravni konkurenti su oni koji se bave sličnim proizvodima ili uslugama, ali ne pružaju iste proizvode ili usluge na identičan način kao konkurent. Neizravni tipovi konkurenata često se bave različitim tržišnim segmentima ili koriste različite metode distribucije. Na primjeru aplikacije koja je opisana u ovom radu, neizravni konkurenti bi bile aplikacije koje pružaju mogućnost personalizacije određenih poklona, poklanjanje poklona isključivo u obliku nekog iskustva ili doživljaja umjesto materijalnog poklona. Također, neizravni

konkurenti su sve aplikacije koje su na bilo koji način povezane s darivanjem posebnih želja odnosno poklona. U nastavku slijedi opis nekoliko neizravnih konkurenata:

- Tinggly

Aplikacija Tinggly korisnicima nudi usluge putovanja i poklona u obliku iskustava. Ona nudi mogućnost pronalaska raznih aktivnosti kao što su izleti, obilasci gradova, sportske aktivnosti, večere u restoranima i još mnogo toga. Nakon odabira poklon iskustva namijenjenog voljenoj osobi, Tinggly šalje poklon vaučer koji se može iskoristiti u više od sto zemalja širom svijeta. Također, dar se može naručiti za sebe, poslati nekome preko DHL-a širom svijeta ili preko USPS Priority servisa za tržište SAD-a. Tinggly poklon iskustvo nema rok isteka, tako da slavljenici mogu rezervirati svoje iskustvo u bilo kojem trenutku. Jedna zanimljiva te pohvalna stvar je da se Tinggly kutije izrađuju od 100% recikliranih materijala te dio profita ide u održive projekte poput proizvodnje, sadnje te zaštite stabala u Keniji, Madagaskaru, Haitiju te mnogim drugim zemljama (Tinggly, 2022).

- Snapfish

Snapfish aplikacija nudi veliki izbor foto proizvoda u mnogo različitih veličina i stilova. Korisnicima nudi personalizaciju poklona za različite prigode poput rođendana, godišnjice, vjenčanja i slično. Vlastite slike se mogu isprobati na različitim proizvodima kao što su šalice, kalendari, albumi, platna te zidne umjetnosti. Odabir fotografija nudi se ne samo s vlastitog uređaja, već su dostupne i opcije preuzimanja istih s različitih društvenih mreža što može biti vrlo korisno. Snapfish također nudi opsežan izbor različitih predložaka boja, dizajna, fontova te ukrasa tako da svoj ili tuđi personaliziran poklon možete učiniti doista unikatnim i nesvakidašnjim (Snapfish, 2022).

- Giftable

Pomoću Giftable aplikacije korisnicima je omogućeno da obave kupnju poklona direktno iz same aplikacije. Nudi se veliki izbor poklona iz raznih kategorija, dodavanje vlastite poruke te mnoštvo drugih zanimljivih funkcionalnosti. Sretnog primatelja poklona moguće je odabrati putem kontakta u telefonskom imeniku ukoliko je primatelj registrirani korisnik

aplikacije. U suprotnom, njemu će putem SMS-a stići obavijest o slanju poklona (Giftable, 2022).

- Christmas Gift List

Aplikacija Christmas Gift List omogućuje korisnicima praćenje preostalog vremena do Božića te kreiranje liste želja za spomenuti blagdan. Listu mogu podijeliti sa svojim prijateljima i obitelji te postaviti određeni prioritet kod svakog poklona. Postoji mogućnost praćenja već dobivenih poklona kako bi se izbjegle neželjene situacije. Također, listu je moguće podijeliti i preko društvenih mreža (Christmas Gift List, 2022).

3. Arhitektura aplikacije

PoklonManija, klijentska mobilna aplikacija, napisana je u programskom jeziku Kotlin te je razvijena koristeći Android operativni sustav. Svaka aplikacija, uključujući i spomenutu treba biti napravljena i dizajnirana prema odgovarajućoj arhitekturi kako bi se osigurala stabilnost, performanse te lakoća korištenja. Arhitektura programske potpore je skup struktura sustava koji sadrže elemente, njihova izvana vidljiva obilježja i odnose između njih (Bass & al., 2012). U početnim fazama razvoja mobilnih aplikacija, arhitekture su bile jednostavne te su se fokusirale na izvođenje trivijalnih zadataka poput prikaza teksta i slike. Kako su s vremenom aplikacije postale sve složenije te su zahtijevale više funkcionalnosti, sama arhitektura aplikacija se trebala prilagoditi kako bi podržala novitete. Danas je arhitektura aplikacija osnovni element oblikovanja i izrade programskih sustava.

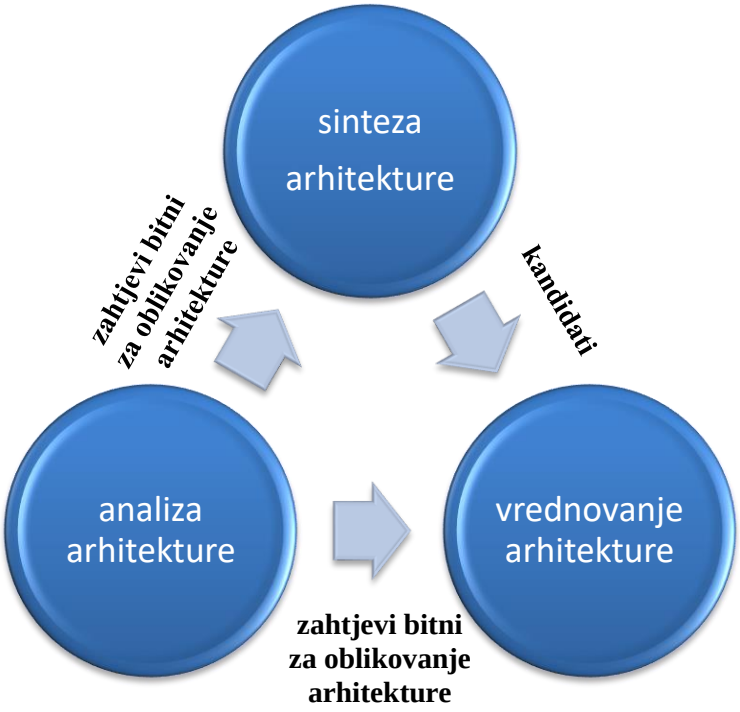
U nastavku će se pružiti uvid u oblikovanje arhitekture sustava te detaljno opisati korištene tehnologije.

3.1. Oblikovanje arhitekture

Oblikovanje arhitekture aplikacije je proces identificiranja i strukturiranja podsustava koji čine cjelinu te okruženja za upravljanje i komunikaciju između tih podsustava (Bass & al., 2012). Glavne prednosti definiranja arhitekture su smanjenje cijene oblikovanja, razvoja i održavanja, poboljšanje kvalitete te mogućnost ponovne uporabe programskog rješenja. Arhitektura također omogućuje razumijevanje cjelokupnog sustava i njegovih funkcionalnosti, olakšanu modifikaciju komponenti te povećava kvalitetu i pouzdanost. Slika 3.1 prikazuje aktivnosti unutar procesa oblikovanja. Proces kreće analizom arhitekture koja započinje grubom skicom modela koja je zasnovana na prikupljanju osnovnih zahtjeva. Zatim se određuju osnovne komponente sustava te se odabire stil arhitekture[1]. On predstavlja najvišu razinu apstrakcije, zasniva se na arhitekturnim obrascima te se prepoznaje temeljem kolekcije komponenata i njihove međusobne interakcije. Razlikujemo objektno usmjereni stil, stil protoka podataka, stil zasnovan na događajima, mikro servisni stil i ostale. S obzirom na mogućnost kombinacije više stilova, u ovom radu odabrani su objektno

[1] Stil arhitekture je obitelj sustava definirana sličnim oblicima strukturne organizacije.

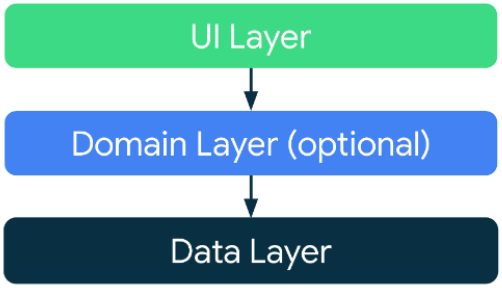
usmjereni i stil zasnovan na događajima. Prvi definira da je sustav sastavljen od skupa objekata u interakciji, dok drugi određuje reakcije na događaje. Slijedi dopunjavanje detaljima te identificiranje pojedinih dijelova. Neki zahtjevi imaju daleko dublji utjecaj na arhitekturu, a obično se definiraju kao arhitekturno značajni zahtjevi. Oni idu u proces sinteze i vrednovanja arhitekture. Sinteza, odnosno razdvajanje sustava je moguće od vrha prema dnu ili obrnuto. Između mnoštva rješenja, cilj je izabrati ono najbolje, odnosno donijeti odluku oblikovanja. To uključuje postavljanje prioriteta, sagledavanje sustava u kontekstu i sustava u cjelini te definiranje potencijalnih izazova. Prilikom tog procesa se koriste razne tehnike i principi.



Slika 3.1 Proces oblikovanja arhitekture (izvor: *Vlastiti rad autora*)

Također, važan je i izbor tehnologija i okruženja u kojemu će se sustav razvijati. Izabrana tehnologija je Android, a odgovarajuća arhitektura mobilne aplikacije se sastoji od tri sloja (Slika 3.2). Prvi je sloj korisničkog sučelja (engl. *UI layer*) koji prikazuje podatke na ekranu. Svaka promjena, bila to interakcija korisnika ili vanjski podražaj, bi trebala ažurirati sučelje. Ovaj sloj se sastoji od elemenata sučelja, poput rasporeda (engl. *layout*) i pogleda (engl. *view*) te držača stanja (engl. *state holder*) koji izlaže podatke. Podatkovni sloj aplikacije sadrži poslovnu logiku. Ona se sastoji od pravila i principa koji određuju kako aplikacija stvara, pohranjuje i mijenja podatke. Preporučuje se izrada klase repozitorija za svaku različitu vrstu podataka. Posljednji sloj je opcionalan, a zove se domenski sloj. Nalazi se

između spomenutih slojeva te je odgovoran je za enkapsulaciju složene poslovne logike koju modeli prikaza (engl. *view model*) ponovno koriste.

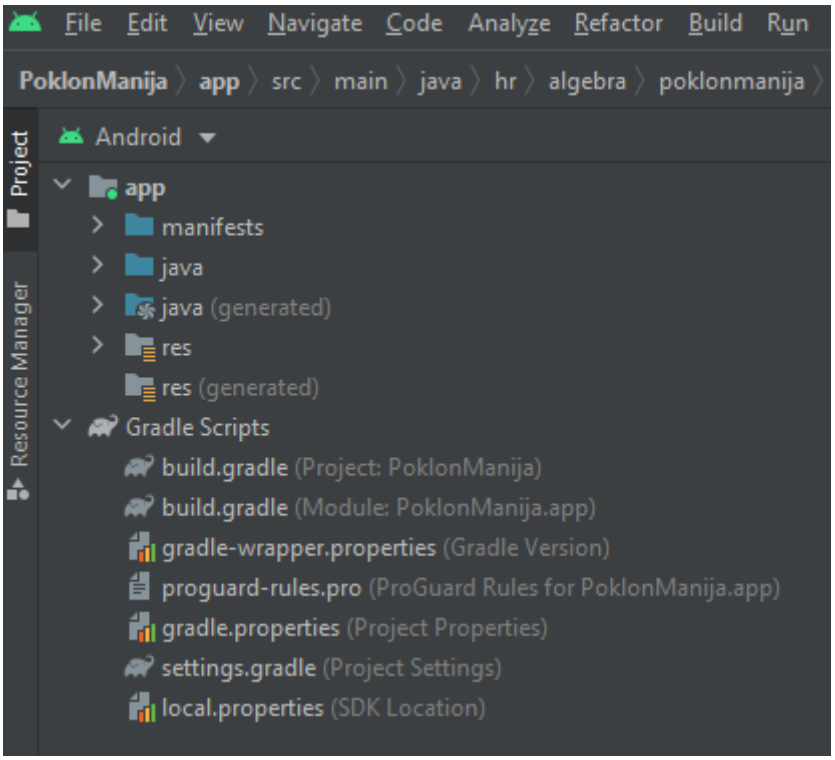


Slika 3.2 Dijagram arhitekture Android mobilne aplikacije (izvor: https://developer.android.com/topic/architecture, preuzeto 28. siječnja 2023.)

3.2. Korištene tehnologije

U praktičnom primjeru ovog rada napravljena je mobilna aplikacija. Razvijena je u Android Studio razvojnom okruženju, koristeći programski jezik Kotlin. Za pohranu podataka korišten je skriptni jezik SQL u okruženju SQLite. Proces pokretanja i testiranja funkcionalnosti aplikacije rađen je putem korištenja Android Emulatora te u stvarnom okruženju na osobnom Android mobilnom uređaju.

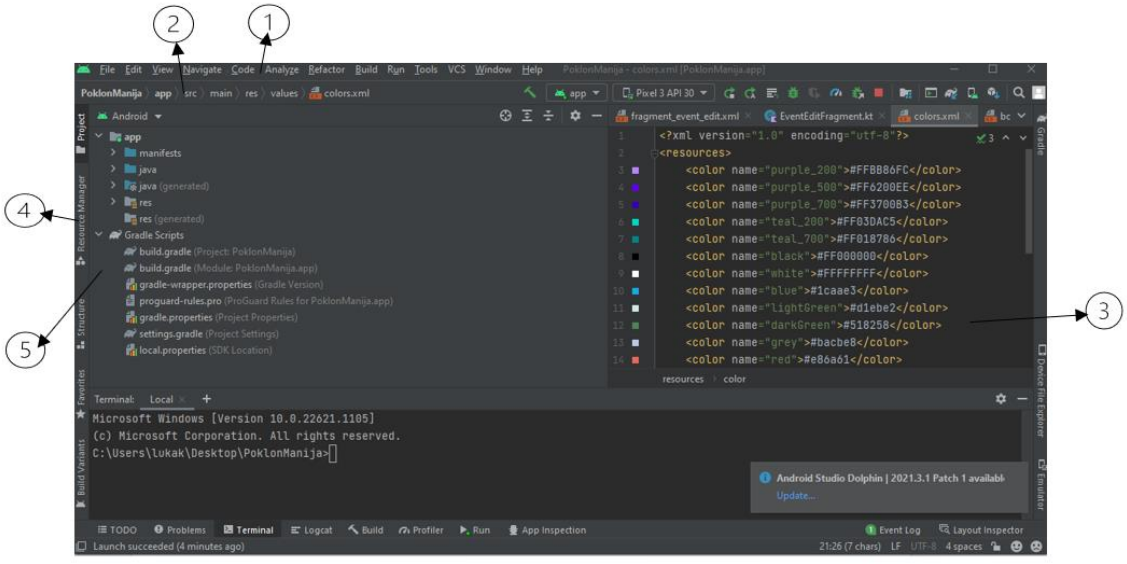
Android Studio službeno je integrirano razvojno okruženje (engl. *integrated development environment*, skraćeno IDE) koje je razvio Google za razvoj aplikacija na Android operativnom sustavu. Odluka za izradu aplikacije u Android Studiju je upravo zbog toga što pruža sve neophodne alate za razvoj aplikacije na jednom mjestu. Koristeći ga, programeri imaju pristup intuitivnom *editoru* za pisanje koda, pouzdanim alatima za otklanjanje neispravnosti, automatiziranim alatima za testiranje, visoko funkcionalnom grafičkom sučelju za dizajn te integraciji s različitim službenim servisima Google-a. Također, on podržava različite verzije Android platforme, što omogućava programerima da stvaraju aplikacije za različite uređaje i verzije operativnog sustava. Prema zadanim postavkama, Android Studio prikazuje projektne datoteke aplikacije PoklonManija što je vidljivo na slici (Slika 3.3).



Slika 3.3 Projektne datoteke (izvor: *Vlastiti rad autora*)

Ovakva organizacija prikaza projektnih datoteka napravljena je kako bi se omogućio brzi pristup ključnim izvornim datotekama projekta.

Glavni prozor Android Studija sastoji se od nekoliko logičkih područja prikazanih na slici (Slika 3.4).



Slika 3.4 Glavni prozor Android Studija (izvor: *Vlastiti rad autora*)

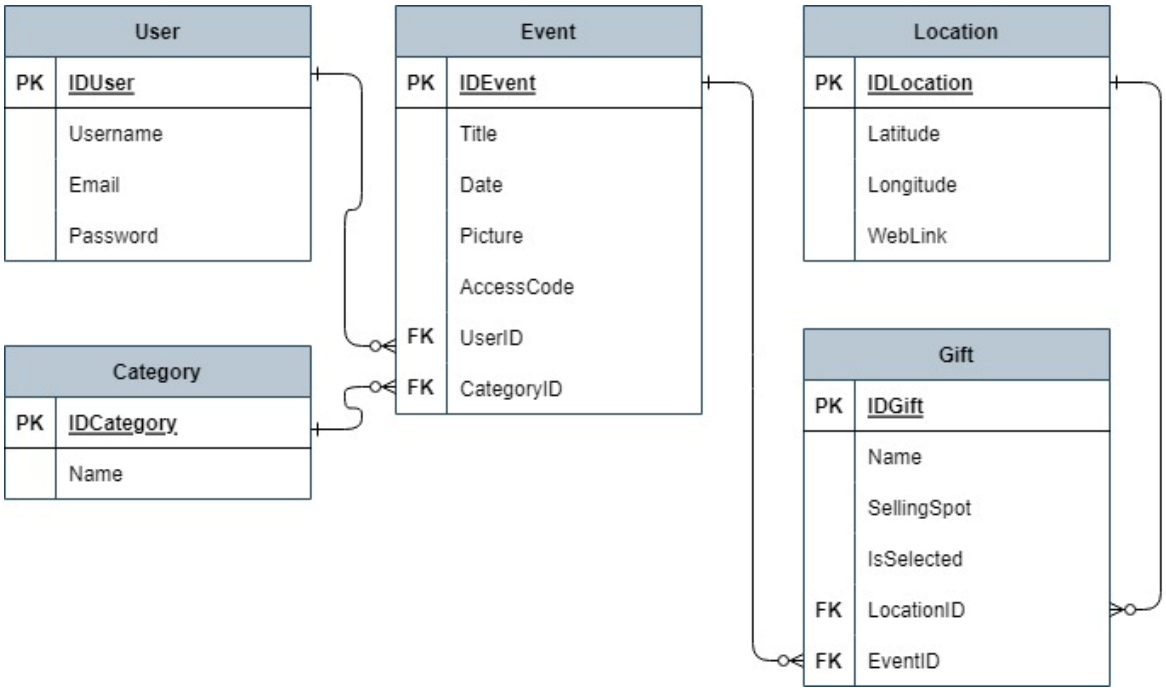
Logičko područje označeno brojem jedan prikazuje alatnu traku koja služi za izvršavanje širokih raspona radnje, uključujući pokretanja aplikacije i pokretanje Android alata. Pod brojem dva označena je navigacijska traka pomoću koje je omogućeno lakše praćenje te kretanje kroz aplikaciju. Prozor označen brojem tri namijenjen je za pisanje te uređivanje koda same aplikacije. U zavisnosti od datoteke, prozor te uređivač koda se može razlikovati. Klikom na traku alatnog prostora koja je označena brojem četiri omogućeno je proširenje ili suženje pojedinačnih prozornih alata. Broj pet prikazuje prozore alata putem kojih pristupamo određenim zadacima (Android Studio, 2023).

Kotlin je programski jezik koji je dizajniran da bude interoperabilan s programskim jezikom Javom, što znači da podržava postojeći Java kod i biblioteke. Međutim, Kotlin ima nekoliko naprednijih funkcionalnosti koje Java nema te je iz tog razloga aplikacija PoklonManija rađena u programskom jeziku Kotlin. Kotlin je statički tipizirani jezik, što znači da se tipovi podataka moraju definirati unaprijed. Spomenuta funkcionalnost omogućuje veću sigurnost koda i olakšava razvoj. Također, Kotlin podržava objektno-orijentirano i funkcionalno programiranje te ima niz naprednih funkcionalnosti poput podrške za multiplatformno programiranje koje omogućuje razvoj aplikacija na više platformi.

Aplikacija koja se opisuje u ovom radu, rađena je na Android operacijskom sustavu. To je operacijski sustav otvorenog koda, temeljen na Linuxu i stvoren za široku lepezu uređaja (Platform Architecture, 2023). Što se tiče arhitekture, ona se proteže kroz nekoliko slojeva. Prvi čini Linux jezgra koja upravlja raznim procesima. Sljedeći je sloj hardverske apstrakcije zadužen za integraciju s komponentama poput kamere, zvuka i senzora. Nekadašnja Dalvik virtualna mašina je od verzije 5.0 zamijenjena Android izvršnim okruženjem (engl. *Android Runtime*, skraćeno ART) koje omogućuje optimizirano izvršavanje aplikacijskog koda i eksploataciju resursa. U višim slojevima se nalaze biblioteke i modularne komponente koje su potrebne za izradu Android aplikacija.

Baze podataka su sustavi za organizaciju i upravljanje podacima. One se koriste za pohranu, pristup i ažuriranje podataka u strukturiranom obliku kako bi se isti kasnije lakše pretraživali i dohvaćali. Postoje različiti tipovi, a baza podataka koja se koristi u ovom radu je relacijskog tipa. Relacijske baze podataka koriste relacijske modele za organiziranje podataka. To znači da se podaci organiziraju u tablice, a svaka tablica ima određene stupce i retke. Tablice se mogu povezati jedna s drugom koristeći strane ključeve. Za pohranu podataka u ovome radu korišten je skriptni jezik SQL (engl. *Structured Query Language*). Korištena je biblioteka

SQLite napisana u C jeziku čiji je izvorni kod u javnoj domeni i besplatan je za bilo koju svrhu (SQLite, 2023). Ona implementira mali, brzi, samostalni, visoko pouzdani i potpuno opremljeni SQL pogon baze podataka. Slika 3.5 prikazuje tablice u obliku dijagrama. S lijeve strane imamo tablicu User s pripadajućim stupcima, odnosno podacima koji se unose prilikom kreiranja računa. Svaki User može kreirati jedan ili više Eventa koji su obilježeni naslovom, lokacijom i datumom održavanja, slikom, pristupnim kodom te pripadajućom kategorijom koja sadrži samo naziv. Unutar svakog Eventa postoji mogućnost kreiranja Gifta, odnosno poklona čija brojnost nije ograničena. Poklon sadrži naziv, prodajno mjesto, stanje zauzetosti te prodajnu lokaciju. Location tablica se sastoji od koordinata i/ili imena web stranice.



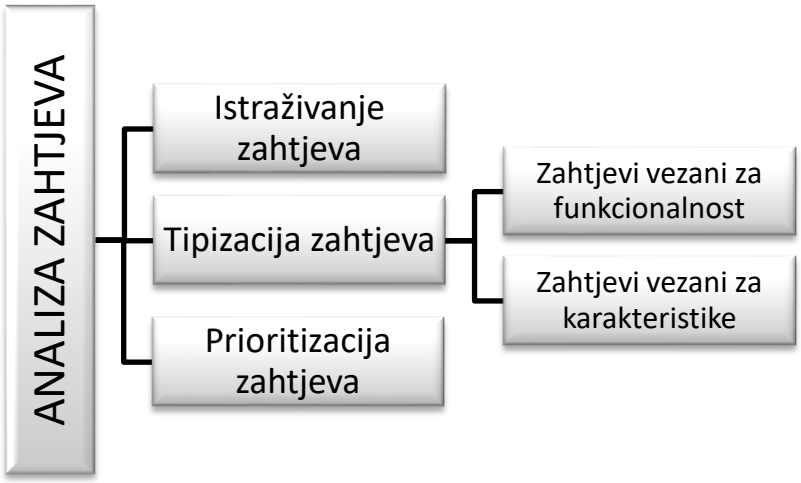
Slika 3.5 Prikaz tablica (izvor: *Vlastiti rad autora*)

4. Implementacija programskog rješenja

Implementacija programskog rješenja je proces koji uključuje pretvaranje razvojnog plana u funkcionalni program. To uključuje kodiranje, testiranje i integriranje svih komponenti aplikacije.

4.1. Analiza zahtjeva

Analiza zahtjeva aplikacije je proces identificiranja i dokumentiranja potreba te očekivanja korisnika i poslovnih procesa. Analiza poslovnog zahtjeva je vrlo važan korak u razvoju aplikacija jer otkriva svrhu projekta na kojem se radi, razlog zašto se izrađuje upravo ta aplikacija te očekivanja nakon plasiranja aplikacije na tržište. Svrha aplikacije PoklonManija je olakšati korisnicima proces odabira kvalitetnog i odgovarajućeg poklona. Samim korištenjem aplikacije, korisnici će uštedjeti vrijeme na odabiru poklona, traženju istih te će slavljenici biti zadovoljni dobivenim. Prikazom grafa (Slika 4.1) vidljivi su glavni koraci u analizi zahtjeva. Prvi je istraživanje zahtjeva koji obuhvaća procese identifikacije, dokumentiranja te evaluacije. Cilj je razjasniti i utvrditi domenu primjene, definirati usluge i ograničenja u radu sustava. Zatim se utvrđeni zahtjevi grupiraju s obzirom na vrstu. Razlikujemo funkcionalne koji opisuju funkcionalnosti koje programska podrška mora pružati te nefunkcionalne koji predstavljaju ograničenja u radu. Tipizacija pomaže pri planiranju i upravljanju te poboljšava kvalitetu sustava. Na kraju se zahtjevi razvrstavaju po važnosti ili nužnosti implementacije istih. To omogućuje fokusiranje na one dijelove o kojima ovisi uspješnost izvedbe. U sljedećim poglavljima detaljno su opisani i navedeni zahtjevi aplikacije.



Slika 4.1 Proces analize zahtjeva (izvor: *Vlastiti rad autora*)

4.1.1. Zahtjevi vezani za funkcionalnost

Zahtjevi vezani uz funkcionalnost aplikacije su specifični zadaci ili radnje koje aplikacija mora biti u stanju izvršiti kako bi zadovoljila potrebe korisnika. Zadaci mogu uključivati osnovne radnje kao što su mogućnost stvaranja, čitanja, ažuriranja i brisanja podataka, kao i naprednije radnje poput pružanja stvarnih vremenskih ažuriranja, obrade velikih količina podataka ili integriranja s drugim sustavima. Zahtjevi vezani uz funkcionalnost navode te opisuju što aplikacija mora raditi i kako to mora raditi.

Zahtjevi vezani za funkcionalnost aplikacije PoklonManija:

- Registracija korisnika

Mobilna aplikacija je dostupna samo registriranim korisnicima. Registracija se sastoji od popunjavanja obrasca registracije, koji zahtijeva od korisnika unos svojih podataka, kao što su korisničko ime, e-mail adresa i lozinka. E-mail adresa i korisničko ime moraju biti jedinstveni, što osigurava da svaki korisnik ima autentični profil. Nakon uspješno popunjenog obrasca i potvrde registracije, korisnik se smatra registriranim i ima pristup aplikaciji. Registracija je neophodna kako bi se osiguralo da samo autorizirani korisnici imaju pristup funkcionalnostima aplikacije, što je bitno za sigurnost i integritet podataka u aplikaciji.

- Prijava korisnika

Prijava u aplikaciju je proces koji omogućuje registriranim korisnicima da se prijave koristeći svoj korisnički račun i pristupe funkcionalnostima i informacijama unutar aplikacije. Da bi se prijavili, korisnici moraju unijeti svoje korisničko ime i lozinku koje su kreirali prilikom registracije. Aplikacija će provjeriti odgovaraju li uneseni podaci onima spremljenim u bazi podataka i ako su podaci ispravni, korisnik će biti prijavljen i imat će pristup aplikaciji PoklonManija.

- Pregled i odabir kategorija

Nakon što se korisnik prijavi u aplikaciju, prva stvar koju vidi je ekran s prikazom određenih kategorija. Ovaj ekran predstavlja pregled svih dostupnih kategorija i omogućuje korisniku da odabere onu koja mu najviše odgovara. Kategorije koje se nude uključuju: vjenčanja, rođendane, božićne proslave, promocije i ostale događaje. Odabirom kategorije Ostalo, korisnici imaju mogućnosti kreiranja događaja prema svojim specifičnim potrebama i željama. Ova opcija omogućuje fleksibilnost i prilagodljivost, što znači da korisnici mogu koristiti aplikaciju za organizaciju različitih tipova događaja.

- Kreiranje vlastitih događaja

Ako korisnik želi koristiti funkcionalnost kreiranja novog događaja, prvo mora odabrati kategoriju u kojoj želi da se isti nalazi. Nakon toga, klikom na gumb za dodavanje događaja, otvara se novi prozor u kojem korisnik mora unijeti tražene podatke za kreiranje događaja. Prvo se unosi naslov, koji bi trebao biti jasan i precizan, kako bi svim uzvanicima bilo jasno o kojem je događaju riječ. Zatim se klikom na gumb u obliku poklona, korisniku omogućuje korištenje funkcionalnost dodavanja poklona u željenom redoslijedu. Također, korisnik može unijeti željenu sliku za svaki događaj kako bi se isti lakše prepoznao i atraktivnije prezentirao. Slika bi trebala biti relevantna za događaj i privlačna za posjetitelje. Kako bi sve ključne informacije bile pohranjene na jednom mjestu, moguće je unijeti mjesto i vrijeme održavanja događaja. Nakon završenog unosa podataka, događaj će se kreirati.

- Uređivanje postojećih događaja

Odabirom određene kategorije, korisnicima aplikacije se prikazuju svi događaji koje su kreirali unutar iste kategorije. Pomoću jednostavnog klika na postojeći događaj, otvara se obrazac za uređivanje koji je sličan onome za kreiranje novog događaja. Svi podaci koji su prethodno bili uneseni bit će već upisani u obrazac, a korisnik će ih moći urediti ili dodati nove želje, bile one iskustva ili materijalne stvari. Spremanjem promjena, postojeći događaj će se ažurirati te će biti dostupan svima s novim promjenama.

- Dodavanje željenih poklona

Prilikom kreiranja događaja, korisnicima je omogućeno dodavanje liste svojih vlastitih želja u obliku poklona. Kikom na gumb u obliku poklona, korisnici stvaraju poklon po poklon koji žele dobiti za određeni događaj. Svaki poklon mora sadržavati naziv te opcionalno link do web stranice ili naziv fizičke prodavaonice u kojoj je moguće kupiti isti.

- Integracija s Google Maps servisom

Kako bi se uzvanicima olakšala sama kupnja željenog poklona, ukoliko je uneseno fizičko prodajno mjesto te koordinate, isto će se prikazati na karti. Time će se skratiti vrijeme potrebno za pronalazak poklona.

- Pristupanje određenom događaju

Prilikom izrade događaja, generira se jedinstveni kod za pristup događaju. Ovaj jedinstveni kod organizator može poslati svim pozvanicima koji su pozvani na događaj. Pozvani gosti putem dobivenog koda mogu pristupiti događaju i odabrati poklon koji žele pokloniti slavljeniku, te pronaći informacije o mjestu i vremenu održavanja događaja. Ovaj proces omogućuje organizaciju i kontrolu pristupa događaju te olakšava proces odabira poklona i pronalaženja informacija o istom.

- Odabir poklona

Pozvani gosti imaju pristup događaju i listi želja slavljenika. Oni mogu pregledati sve želje i odabrati poklon koji namjeravaju pokloniti i koji su u mogućnosti kupiti. Odabirom istog, on više nije dostupan ostalim uzvanicima. Time se izbjegava situacija kupnje istih poklona od strane različitih gostiju. Također, ovaj proces olakšava organizaciju i kontrolu izbora poklona.

- Anonimni chat

Funkcionalnost anonimnog chata pruža uzvanicima mogućnost razjašnjenja nejasnoća vezanih uz kupnju poklona. Oni mogu postaviti bilo kakva pitanja koja ih zanimaju vezano uz poklone, predložiti svoje poklone koji žele pokloniti te vidjeti jesu li odgovarajući prema željama slavljenika, dok im kreator može pomoći u rješavanju nedoumica. Uzvanici će također moći komentirati listu želja slavljenika. Ova funkcionalnost će omogućiti uzvanicima da osiguraju kupnju odgovarajućeg dara za slavljenika.

- Statistika o danim poklonima

Svaki kreator događaja će imati priliku vidjeti statistiku o odabranim poklonima. Međutim, slavljenik neće znati od koga će dobiti koji poklon, već će samo vidjeti broj odabranih poklona s obzirom na sve upisane poklone. Ovim pristupom će se sačuvati faktor iznenađenja.

4.1.2. Zahtjevi vezani uz karakteristike

Definicija karakteristika aplikacije je skup zahtjeva koji se odnose na performanse, kvalitetu i pouzdanost aplikacije. One direktno utječu na korisničko iskustvo i mogućnost korištenja aplikacije. Zahtjevi vezani uz karakteristike aplikacije obično uključuju stvari poput brzine, sigurnosti, pouzdanosti, skalabilnosti, jezika, prilagodljivosti, lakoće uporabe, dostupnosti, mogućnosti proširivanja, pomoći i podrške.

Zahtjevi vezani uz karakteristike aplikacije PoklonManija:

- Aplikacija je dostupna na hrvatskom jeziku
- Aplikacija mora biti stabilna te raditi bez prekida
- Aplikacija mora biti intuitivna i jednostavna za korištenje
- Dostupnost aplikacije mora biti podržana na svim Androrid uređajima
- Korištene tehnologije:
  - Programski jezik Kotlin korišten u Android Studio okruženju
  - SQLite
- Samo registrirani korisnici će moći koristiti aplikaciju
- Uzvanici će isključivo putem koda moći pristupiti određenom događaju

4.2. Implementacija zahtjeva

Što se tiče same implementacije rješenja, u nastavku će biti detaljnije objašnjena izvedba glavnih funkcionalnosti aplikacije uz prikaz koda.

4.2.1. Registracija i prijava u aplikaciju

Na primjeru procesa registracije objašnjena je implementacija i komunikacija s bazom podataka. Kod 4.1 prikazuje klasu `SqlHelper` koja nasljeđuje ugrađenu klasu `SQLiteOpenHelper` zaduženu za lako korištenje SQLite baze podataka. Ona omogućuje implementaciju funkcija za kreiranje, mijenjanje, brisanje i dohvaćanje podataka, kao i praćenje verzije same baze. U ovom primjeru je vidljivo da se pri kreiranju baze izvrši upit za kreiranje tablice `User`. Pri ažuriranju baze bi se ta ista tablica obrisala (engl. *drop*) i ponovno kreirala. Klasa `SqlHelper` implementira `UserRepository` te samim time i metodu `insertUser`.

```
private const val TABLE_USER = "user"
private const val DB_NAME = "poklonmanija.db"
private const val VERSION = 1

private val CREATE_USER = "create table $TABLE_USER(" +
        "${User::_id.name} integer primary key autoincrement,
" +
        "${User::username.name} text not null, " +
        "${User::email.name} text not null, " +
        "${User::password.name} text not null " +
        ")"
private const val DROP_USER = "drop table $TABLE_USER"

class SqlHelper(
    context: Context?
) : SQLiteOpenHelper(context, DB_NAME, null, VERSION),
UserRepository {

    override fun onCreate(db: SQLiteDatabase) {
        db.execSQL(CREATE_USER)
    }

    override fun onUpgrade(db: SQLiteDatabase, oldVersion:
Int, newVersion: Int) {
        db.execSQL(DROP_USER)
        onCreate(db)
    }
```

```
    override fun insertUser(values: ContentValues?)
           =writableDatabase.insert(TABLE_USER, null, values)
}
```

Kod 4.1 SqlHelper klasa

Nadalje, slijedi implementacija `ContentProvidera`. On služi za pristup podacima na siguran i standardiziran način te se često koristi u kombinaciji sa SQLite bazom. Također koristi standardizirane URI adrese za identifikaciju resursa. Kod 4.2 prikazuje implementaciju `UserProvider` klase te pripadajuće metode za unos korisnika koja poziva prethodno implementiranu metodu `insertUser`.

```
private const val AUTHORITY = "hr.algebra.poklonmanija"
private const val PATH = "user"
val PROVIDER_URI = Uri.parse("content://$AUTHORITY/$PATH")

class UserProvider : ContentProvider() {

    private lateinit var repository: UserRepository

    override fun insert(uri: Uri, values: ContentValues?):
Uri? {
        val id = repository.insertUser(values)
        return ContentUris.withAppendedId(PROVIDER_URI, id)
    }
}
```

Kod 4.2 UserProvider klasa

Napokon, `contentResolver` unosi podatke putem ContentProvidera koji se nalaze na određenoj URI adresi pozivom metode `registerUser`.

```
private fun registerUser(user: User) {
      try {
          context.contentResolver.insert(PROVIDER_URI,
values)
       } catch (e: Exception) {
          throw Exception(e.message)
       }
}
```

Kod 4.3 Registracija korisnika

Na sličan način je implementirana prijava korisnika koristeći metodu za dohvaćanje korisnika s unesenim korisničkim imenom i lozinkom.

4.2.2. Kategorije događaja

Lista kategorija se prikazuje koristeći `RecyclerView` koji olakšava učinkovito prikazivanje velikih skupova podataka. On dinamički stvara elemente i upravlja istima. Kao što i samo ime kaže, on reciklira stare elemente te ih ponovno koristi za prikaz novih nakon pomicanja zaslona (engl. *screen scrolling*) (Android for Developers, 2023).

Njegovi elementi su u primjeru (Kod 4.4) definirani koristeći `MaterialCardView` iz biblioteke *com.google.android.material.card.MaterialCardView*. On pruža dosljedan izgled i stil kontejnera za svaku stavku iz neke liste. Jednom definirani CardView služi kao prikaz za dinamički definirane podatke svakog elementa (Android for Developers, 2023). U primjeru je vidljivo da sadrži relativni raspored koji elemente unutar sebe prikazuje u relativnom odnosu s obzirom na roditeljski element ili ostale elemente. Pa tako element `TextView` prikazuje preko elementa `ImageView`.

```
<androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerview"
        android:layout_width="500dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="35dp"
        android:layout_gravity="center_horizontal"
        android:layout_marginTop="210dp"
        tools:listitem="@layout/card_view_category" />

<com.google.android.material.card.MaterialCardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="wrap_content"
    android:layout_height="175dp"
    app:cardUseCompatPadding="true"
    android:rotation="0"
    app:strokeWidth="1dp"
    app:strokeColor="@color/ribbon"
```

```
        app:cardCornerRadius="2dp"
        android:id="@+id/cardView">

         <RelativeLayout
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_gravity="center">

                <ImageView
                    android:id="@+id/ivIcon"
                    android:layout_width="420dp"
                    android:layout_height="192dp"
                    android:scaleType="centerCrop"
                    tools:src="@drawable/weeding2" />

                <TextView
                    android:id="@+id/tvCategory"
                    android:layout_width="wrap_content"
                    android:layout_height="42dp"
        android:fontFamily="@font/beauty_luxury_regular"
                    android:letterSpacing="0.1"
                    tools:text="Vjenčanja"
                    android:textAllCaps="true"
                    android:layout_centerInParent="true"
                    android:textColor="@color/ribbon"
                    android:textSize="27sp"
                    android:textStyle="bold" />
            </RelativeLayout>

    </com.google.android.material.card.MaterialCardView>
```

Kod 4.4 Kreiranje RecyclerView komponente za prikaz kategorija

Adapter i ViewHolder se koriste za prikaz RecyclerView komponente. Adapter spaja podatke iz modela s prikazom na RecyclerView koristeći u primjeru prikazanu metodu `bind` (Kod 4.5). ViewHolder sprema reference na objekte unutar RecyclerViewa što ubrzava prikaz velikog broja podataka. Također sadrži slušatelje događaja (engl. *event listener*). U ovom primjeru osluškuje klik na sam CardView koji rezultira pokretanjem nove aktivnosti (engl. *activity*) u Androidu naziva `EventActivity` o kojem detaljnije u nastavku.

```
class CategoryAdapter(private val context: Context, private
val categories: MutableList<Category>)
    : RecyclerView.Adapter<CategoryAdapter.ViewHolder>() {

    class ViewHolder(categoryView: View) :
RecyclerView.ViewHolder(categoryView) {

        private val tvCategory =
itemView.findViewById<TextView>(R.id.tvCategory)
        private val ivIcon =
itemView.findViewById<ImageView>(R.id.ivIcon)
        fun bind(item: Category) {
            tvCategory.text = item.name
            ivIcon.setImageResource(item.icon)
        }
    }

    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder =
        ViewHolder(categoryView =
LayoutInflater.from(parent.context)
            .inflate(R.layout.card_view_category, parent,
false))

    override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
        holder.itemView.setOnClickListener {
           context.startActivity<EventActivity>()
        }
        holder.bind(categories[position])
    }

    override fun getItemCount() = categories.size
}
```

Kod 4.5 Upravljanje kategorijama koristeći Adapter i ViewHolder

4.2.3. Kreiranje događaja

EventActivity prikazuje fragmente odnosno korisnička sučelja koja služe za višekratnu upotrebu i dinamičko mijenjanje sadržaja bez pokretanja nove aktivnosti. Jedan od implementiranih fragmenata je EventEditFragment koji služi za dodavanje i uređivanje događaja. On se sastoji od unosa naslova, lokacije, slike i datuma te popisa želja. Naslov i lokacija se unose ručno kroz komponentu EditText, dok za datum postoji posebna komponenta `DatePicker`. Ona je vrlo korisna zbog intuitivnosti i jednostavnosti te sprječava unošenje krivog formata datuma.

```
<DatePicker
            android:id="@+id/datePicker"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:datePickerMode="spinner"
            android:layout_gravity="center"
            android:calendarViewShown="false"/>
```

Ikona, odnosno slika događaja se može odabrati iz galerije na uređaju koja se otvara na duži klik gumba za dodavanje slike. Metoda `handleImage` učitava sliku i postavlja je u ImageView koristeći Picasso biblioteku.

```
ivAddIcon.setOnLongClickListener {
            handleImage()
            true}

 private fun handleImage() {
        val intent = Intent(Intent.ACTION_GET_CONTENT).apply{
            addCategory(Intent.CATEGORY_OPENABLE)
            type = IMAGE_TYPE
            imageResult.launch(this)
        }
    }

    private val imageResult =
registerForActivityResult(ActivityResultContracts.StartActivi
tyForResult()) {
            if (it.resultCode == Activity.RESULT_OK &&
it.data != null) {
```

```
                val dir =
context?.applicationContext?.getExternalFilesDir(null)
                val file = File(dir,
File.separator.toString() + UUID.randomUUID().toString() +
".jpg")

context?.contentResolver?.openInputStream(it.data?.data as
Uri).use { inputStream ->
                    FileOutputStream(file).use { fos ->
                        val bitmap =
BitmapFactory.decodeStream(inputStream)
                        val bos = ByteArrayOutputStream()

bitmap.compress(Bitmap.CompressFormat.JPEG, 100, bos)
                        fos.write(bos.toByteArray())
                        picPath = file.absolutePath

      Picasso.get()
             .load(File(picPath))
             .transform(RoundedCornersTransformation(50, 5))
             .into(binding.ivAddIcon)
                    }
                }
            }
        }
```

Kod 4.6 Unos slike iz galerije

4.2.4. Kreiranje i odabir liste želja

Tijekom kreiranja događaja, unosi se i popis želja, odnosno željenih poklona. Lista mora sadržavati barem jedan element, u suprotnom nije moguće kreirati događaj. Klikom na gumb u obliku poklona se stvara nova instanca LinearLayout komponente koja sadrži element za unos teksta i gumb za unos lokacije. Gumb za unos lokacije otvara dijalog putem kojeg se unose fizička i/ili web adresa prodajnog mjesta na kojem je moguće kupiti poklon. Ukoliko korisnik unese barem jednu adresu, to označava da je prodajno mjesto navedeno te gumb promijeni boju u zelenu.

```
btnWish.setOnClickListener {
     createInputFieldForGift(container)
```

```
 }

 private fun createInputFieldForGift(container: LinearLayout)
{
        val linearLayout = LinearLayout(requireContext())
        linearLayout.layoutParams =
LinearLayout.LayoutParams(
            LinearLayout.LayoutParams.MATCH_PARENT,
            LinearLayout.LayoutParams.WRAP_CONTENT
        )
        linearLayout.orientation = LinearLayout.HORIZONTAL
        container.addView(linearLayout)

        val editText = EditText(requireContext())
        editText.layoutParams =
LinearLayout.LayoutParams(900, 110)

editText.setTextAppearance(R.style.TextAppearance_AppCompat_M
edium)
        editText.setTextColor(Color.rgb(7, 72, 192))
        editText.setPadding(70, 20, 10, 10)
        editText.maxLines = 1
        editText.background =
resources.getDrawable(R.drawable.border)
        editText.gravity = Gravity.LEFT
        linearLayout.addView(editText)

        giftEditTextList.add(editText)

        btnStore = Button(requireContext())
        btnStore.layoutParams =
LinearLayout.LayoutParams(80,80)
        btnStore.gravity = Gravity.END
        btnStore.gravity = Gravity.CENTER
        btnStore.background =
resources.getDrawable(R.drawable.addlocation)
        linearLayout.addView(btnStore)

        setupListenerOnBtnStore()
    }
```

```
    private fun setupListenerOnBtnStore() {
        btnStore.setOnClickListener {
            val fragment = DialogAddLocationFragment()
            fragment.setTargetFragment(this, 0)
            val fragmentManager = fragmentManager
            val fragmentTransaction =
fragmentManager!!.beginTransaction()
            fragmentTransaction.add(fragment,
"DialogAddLocationFragment")
            fragmentTransaction.commit()
        }
    }

override fun onDataPass(data: Any) {
        storeLocation = data as GiftStoreLocation
   if (data.physicalAddress != "" || data.webAddress != "") {
btnStore.setBackgroundResource(R.drawable.locationgreen)
        }
    }
```

Kod 4.7 Dodavanje željenih poklona

Vezano uz odabir poklona, implementiran je ekran s prikazanim detaljima događaja i popisom poklona. Pokraj svakoga se nalazi potvrdni okvir kojeg je, ukoliko je omogućen i dostupan, moguće kliknuti i time odabrati željeni dar.

```
<CheckBox
      android:id="@+id/cbSelect"
      android:layout_width="40dp"
      android:layout_height="40dp"
      android:layout_marginLeft="20dp"
      android:clickable="true" />
```

4.2.5. Anonimni chat i statistika

Za implementaciju anonimnog chata i statistike korištene su vanjske biblioteke.

Chat je kreiran uz pomoć komunikacijske platforme MirrorFly, koja omogućuje brzo i efikasno implementiranje poruka jedan na jedan, poziva, video poziva, notifikacija i ostalog. Platforma se temelji na Cloud infrastrukturi te je dostupna u obliku SDK-a. Nakon

implementirane registracije korisnika i spajanja na server, izrađene su metode za slanje i primanje poruka.

```
binding.btnSend.setOnClickListener {
            sendMessages()}

private fun sendMessages() {
        val input = findViewById<EditText>(R.id.etMessage)
        val myMessage = findViewById<TextView>(R.id.tvMy)
        var messageToSend = input.text.toString()
        FlyMessenger.sendTextMessage("12345609812@xmpp-
preprod-sandbox.mirrorfly.com", messageToSend, listener =
object : SendMessageListener {
            override fun onResponse(isSuccess: Boolean,
chatMessage: ChatMessage?) {
                if (isSuccess){
                    Log.i(LTAG, "Message sent")
                    myMessage.text = "Vi: " + messageToSend
                    input.setText("")
                } else {
                    Log.i(LTAG, "Message failed")
                }
            }
        })
    }

override fun onMessageReceived(message: ChatMessage) {
        super.onMessageReceived(message)
        var rmesssage =
message.getMessageTextContent().toString()
        Log.i(LTAG, rmesssage)
        var messageView =
findViewById<TextView>(R.id.tvMessage)
        messageView.text = "Kreator: " + rmesssage
    }
```

Kod 4.8 Primanje i slanje poruka koristeći MirrorFly

Grafički prikaz statistike je implementiran koristeći MPAndroidChart biblioteku koja omogućuje brzo i jednostavno stvaranje različitih vrste grafikona te vizualizaciju podataka u aplikacijama. Kod 4.9 prikazuje implementaciju unutar Statistics aktivnosti.

```
val dataSet = PieDataSet(entries, "")
        dataSet.colors = listOf(Color.rgb(62, 115, 222),
Color.rgb(200, 203, 207))
        dataSet.valueTextSize = 20f
        dataSet.valueTypeface = Typeface.DEFAULT_BOLD
        dataSet.valueTextColor = Color.WHITE

        val data = PieData(dataSet)
        chart.data = data
        chart.description.text =
getString(R.string.statisticsTitle)
        chart.holeRadius = 0f
        chart.description.textSize = 13f
        chart.transparentCircleRadius = 10f
        chart.isRotationEnabled = true
        chart.rotationAngle = 25f
        chart.isHighlightPerTapEnabled = true

        chart.invalidate()
```

Kod 4.9 Implementacija grafa

5. Rezultati korištenja aplikacije

Rezultati korištenja aplikacije su promjene ili efekti koji se javljaju prilikom susretanja korisnika s aplikacijom. Sami rezultati mogu se mjeriti i analizirati kako bi se odredile korisnosti aplikacije te poduzela potrebna poboljšanja. Darivanje je jedan od najljepših načina pokazivanja ljudima koliko nam je stalo do njih. Međutim, ponekad je teško pronaći savršeni poklon za nekoga, osobito ako nemamo dovoljno vremena ili se nalazimo daleko od trgovina. Upravo zbog toga je razvijena aplikacija PoklonManija, koja omogućuje pronalazak te odabir odgovarajućeg poklona bez napora i briga. Aplikacija PoklonManija za odabir i darivanje poklona pruža širok raspon značajki koje mogu pomoći korisnicima pri odabiru poklona koji su relevantni i prilagođeni potrebama slavljenika. Rezultati korištenja aplikacije opisani u ovom radu vidljivi su u tablici (Tablica 5.1).

Rezultati	Opis
Povećanje efikasnosti i organizacije	Aplikacija pomaže korisnicima pri kreiranju i organiziranju svojih događaja, poput rođendana, vjenčanja, Božića i promocija, pozivanju uzvanika te stvaranju liste želja koje bi voljeli dobiti na poklon, što može smanjiti vrijeme i trud potreban za organizaciju događaja.
Povećanje zadovoljstva korisnika	Aplikacija pruža slavljenicima, odnosno kreatorima događaja opciju za kreiranje liste željenih poklona koji su prilagođeni potrebama događaja te slavljenikovim zamislima. To povećava vjerojatnost da će poklon biti uspješno prihvaćen i cijenjen. Uzvanicima događaja također je omogućeno da odaberu poklon koji mogu i

	žele kupiti, što može povećati njihovo zadovoljstvo.
Povećanje učinkovitosti kupnje poklona	Uzvanicima događaja aplikacija omogućuje pregledavanje te odabir poklona direktno iz aplikacije, što rezultira smanjenjem stresa povezanim s kupnjom poklona. Uzvanicima je također omogućeno da pregledaju lokacije na Google Mapsu gdje se mogu kupiti određeni pokloni, što može povećati učinkovitost kupnje.
Povećanje komunikacije	Uzvanicima događaja je omogućeno korištenje anonimnog chata, što olakšava komunikaciju s kreatorom i kupnju idealnog poklona.

Tablica 5.1 Rezultati korištenja aplikacije (izvor: *Vlastiti rad autora*)

5.1. Upute za korištenje aplikacije

Spomenute funkcionalnosti zajedno čine aplikaciju PoklonManija cjelovitom i autentičnom te su u nastavku predstavljene detaljne upute za korištenje aplikacije kako bi korisnici mogli iskoristiti sve njene prednosti.

Registracija korisnika je prvi i ključni korak za korištenje aplikacije PoklonManija. Proces registracije je brz te se odvija u nekoliko jednostavnih koraka. Nakon što korisnici otvore aplikaciju, na zaslonu su im dostupne opcije za registraciju i prijavu. Kako bi se registrirali, korisnici trebaju unijeti tražene podatke i kliknuti na gumb „Registriraj se“. Korisnici se registriraju unošenjem svojih osobnih podataka, kao što su e-mail adresa, korisničko ime i lozinka. Registracija je važan korak za korištenje aplikacije jer omogućuje korisnicima da pristupaju svim funkcionalnostima aplikacije. Slika 5.1 prikazuje pravilan te uspješan postupak registracije u samu aplikaciju.

Slika 5.1 Registracija korisnika (izvor: *Vlastiti rad autora*)

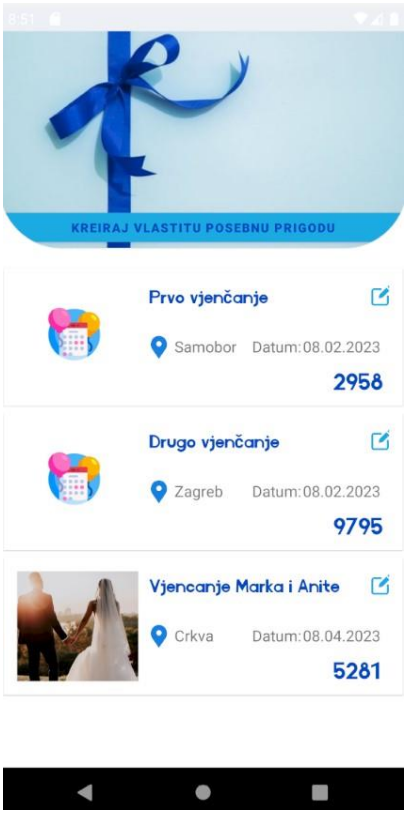
Nakon uspješne registracije korisnici se moraju, svaki put kada žele pristupiti aplikaciji, prijaviti putem svojeg korisničkog imena i lozinke (Slika 5.2). Ova mjera sigurnosti osigurava da samo registrirani korisnici mogu pristupiti aplikaciji i koristiti njene funkcionalnosti. Prijava se odvija na zaslonu aplikacije koji se otvara klikom na gumb „Imaš račun? Prijavi se“. Ukoliko se unesu ispravni podaci, otvara se novi ekran s prikazom kategorija.

Slika 5.2 Prijava korisnika (izvor: *Vlastiti rad autora*)

Sljedeća funkcionalnost omogućuje korisnicima da odaberu kategoriju za svoj događaj, a prikaz ekrana je vidljiv na slici (Slika 5.3.) Kategorije koje su ponuđene su Vjenčanja, Rođendani, Božić, Promocije i Ostalo. One olakšavaju organizaciju i pretraživanje kreiranih događaja. Klikom na određenu kategoriju, otvara se novi prozor u kojem korisnici imaju mogućnost stvaranja novih događaja i uređivanja postojećih.

Slika 5.3 Prikaz kategorija (izvor: *Vlastiti rad autora*)

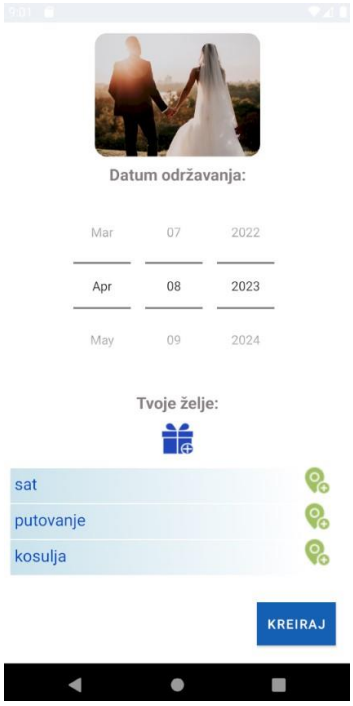
Odabirom određene kategorije otvara se ekran s prikazom prethodno kreiranih korisnikovih događaja unutar nje. Slika 5.4 prikazuje tri događaja s pripadajućim podacima. Prva dva događaja imaju predefiniranu sliku jer ista nije odabrana kroz galeriju. Svaki događaj ima i svoj pristupni kod koji se generira nakon kreiranja i prikazuje u desnom donjem kutu događaja.



Slika 5.4 Prikaz događaja (izvor: *Vlastiti rad autora*)

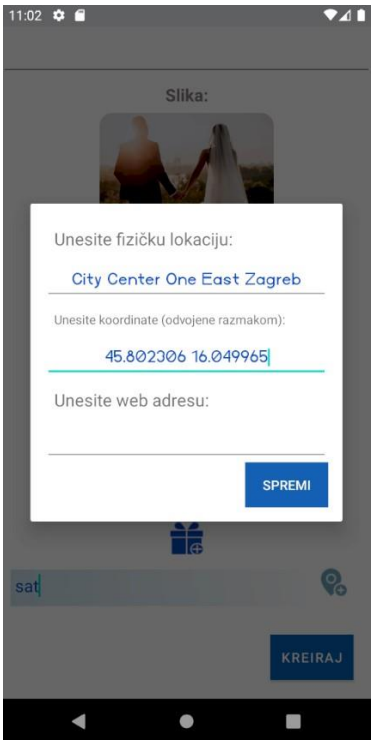
Procesi kreiranja i uređivanja događaja otvaraju isti ekran, odnosno formu prikazanu na slikama (Slika 5.5, Slika 5.6). Uređivanje se pokreće klikom na gumb u desnom gornjem kutu kartice događaja te se otvara forma s prethodno unesenim podacima. Podaci koji su već uneseni su spremljeni, pa korisnik može promijeniti ili dodati nove podatke prema svojim željama. Pritiskom na gumb za kreiranje posebne prigode, korisnicima se otvara prazna forma. Stvaranje događaja podrazumijeva popunjavanje podataka kao što su naslov događaja te mjesto i datum održavanja, dodavanje slike te navođenje željenih poklona. Naslov bi trebao biti jasan jer će olakšati razumljivost i prepoznavanje događaja uzvanicima. Što se tiče slike, ona se prikazuje na stranici detalja događaja, čineći je atraktivnijom za goste. Može biti slika mjesta događaja, dizajn ili logo vezan uz događaj ili čak fotografija osoba koje organiziraju događaj. Željeni pokloni se dodaju u listu želja klikom na gumb u obliku poklona. Svakim klikom na spomenuti gumb, korisnici dodaju zasebno poklon po poklon.

Slika 5.5 Kreiranje događaja (izvor: *Vlastiti rad autora*)



Slika 5.6 Kreiranje događaja (izvor: *Vlastiti rad autora*)

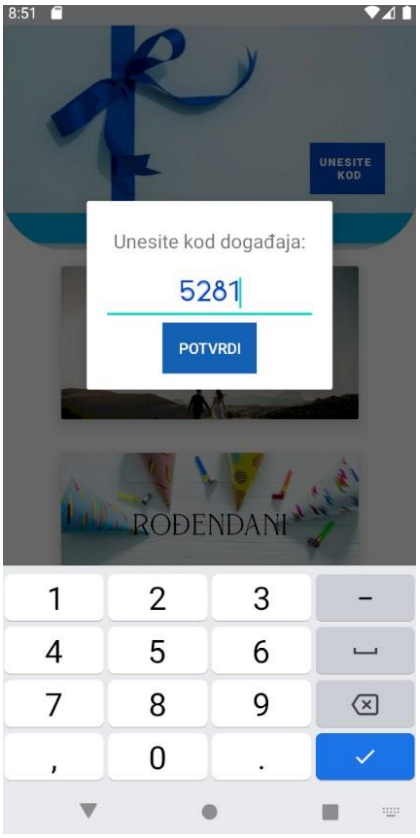
Za svaki poklon je moguće unijeti lokaciju na kojoj se isti prodaje. Klikom na ikonu lokacije otvara se dijalog koji zahtjeva unos fizičkog ili web prodajnog mjesta te spremanja istog (Slika 5.7). Fizička lokacija zahtjeva unos koordinata kako bi se ista mogla pokazati na karti. Nakon uspješnog unosa gumb promijeni boju u zelenu (Slika 5.8).



Slika 5.7 Unos lokacije prodajnog mjesta poklona (izvor: *Vlastiti rad autora*)

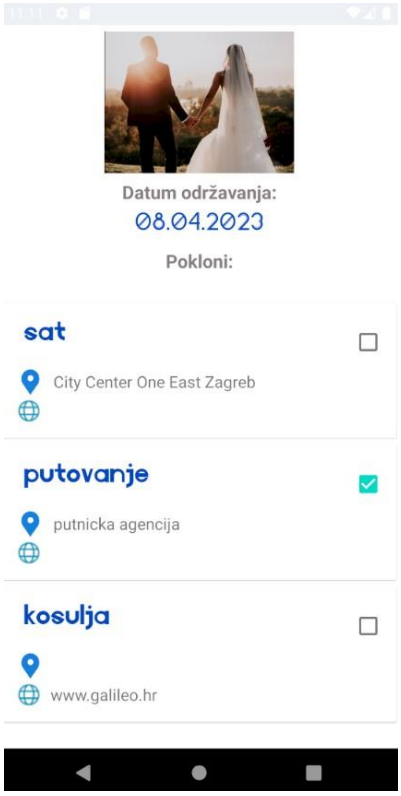
Slika 5.8 Promjena boje gumba (izvor: *Vlastiti rad autora*)

Nakon kreacije događaja, korisnik može podijeliti događaj na način da proslijedi generirani kod događaja uzvanicima. Svaki uzvanik koji želi vidjeti popis poklona mora izraditi račun i prijaviti se u aplikaciju. Zatim na ekranu sa slike (Slika 5.3) odabire gumb za unos koda. Tamo, na primjer, unosi kod 5281 od događaja naziva „Vjenčanje Marka i Anite“ (Slika 5.9). Klikom na gumb Potvrdi otvara se zaslon s detaljima događaja i popisom poklona (Slika 5.10, Slika 5.11). Klikom na potvrdni okvir kraj poklona je isti moguće izabrati.



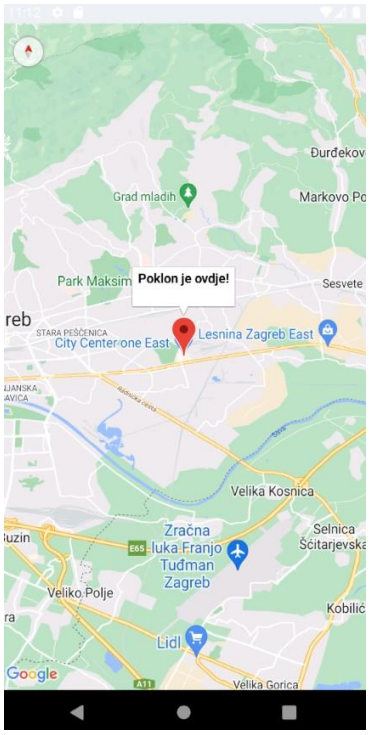
Slika 5.9 Unos pristupnog koda (izvor: *Vlastiti rad autora*)

Slika 5.10 Detalji događaja (izvor: *Vlastiti rad autora*)



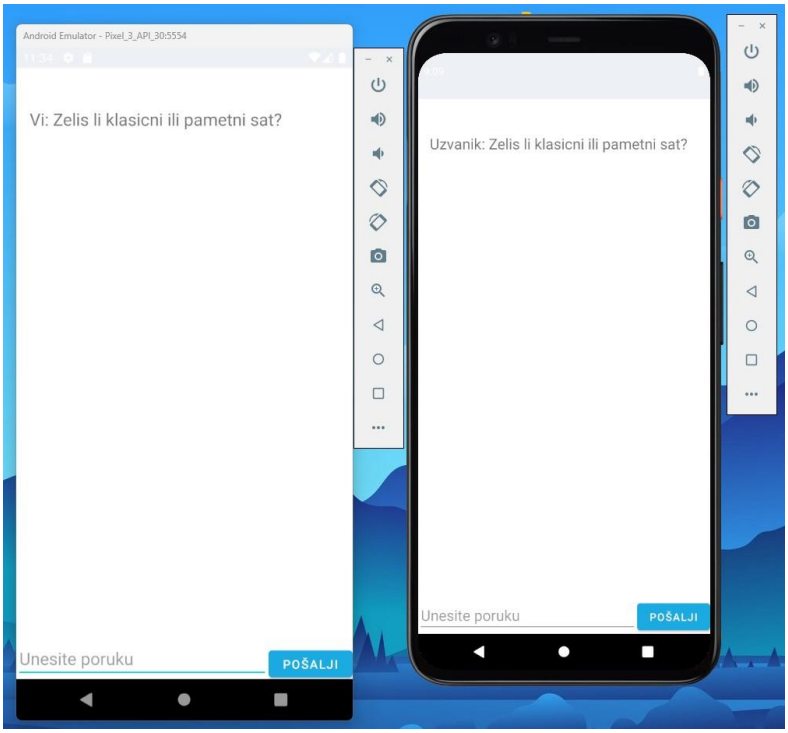
Slika 5.11 Detalji događaja (izvor: *Vlastiti rad autora*)

Klikom na ikonu lokacije pokraj imena fizičkog prodajnog mjesta, u ovom slučaju City Center One East Zagreb, otvara se Google Maps karta koja pokazuju gdje se lokacija nalazi, odnosno gdje se poklon može kupiti (Slika 5.12).

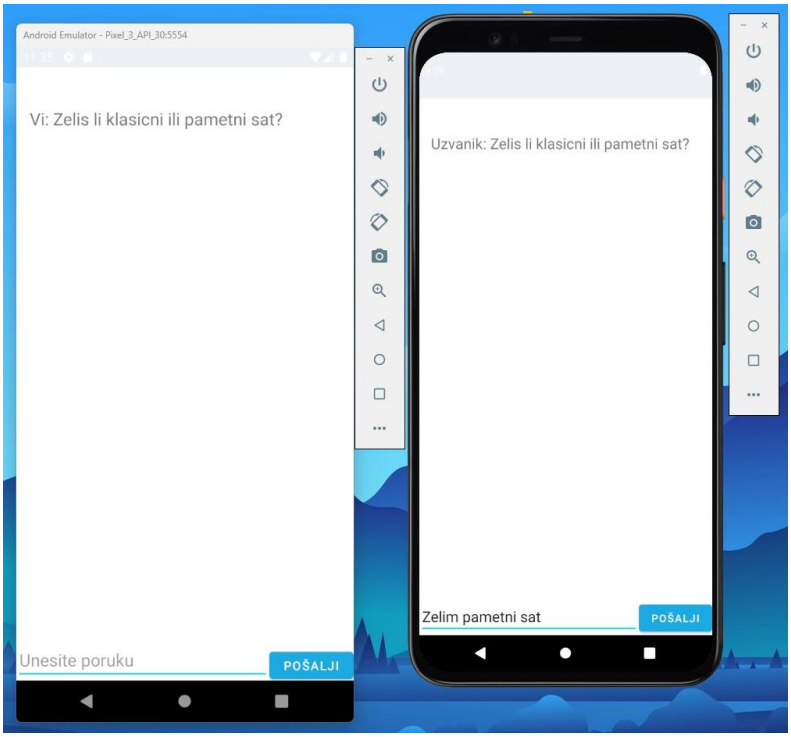


Slika 5.12 Prikaz lokacije na karti (izvor: *Vlastiti rad autora*)

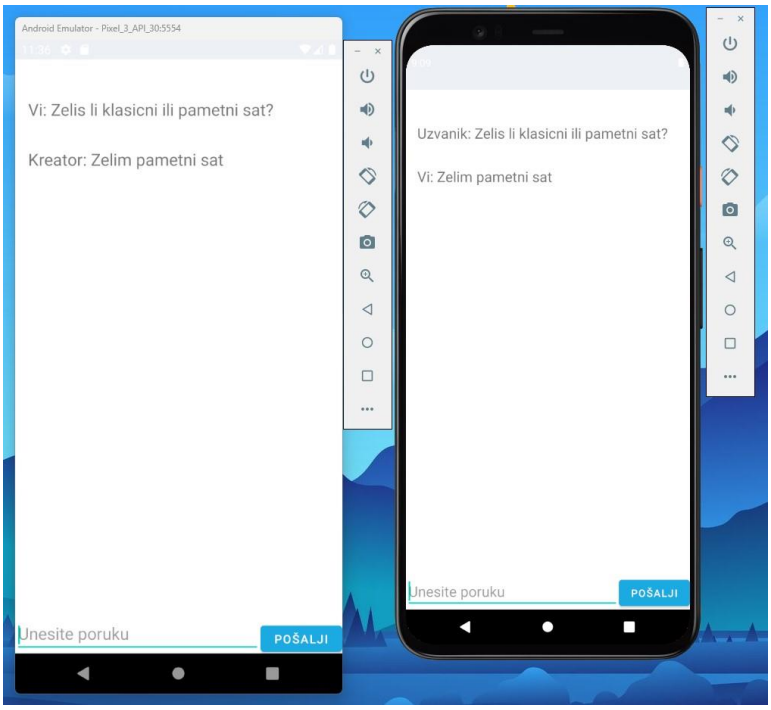
Uzvanici događaja imaju mogućnost korištenja funkcionalnosti anonimnog chata za postavljanje bilo kakvih pitanja vezanih uz željene poklone. To im omogućuje detaljnije informiranje u vezi poklona bez otkrivanja svog identiteta. U nastavku je slikovno prikazan tijek komunikacije između kreatora i uzvanika (Slika 5.13, Slika 5.14, Slika 5.15).



Slika 5.13 Slanje poruke kreatoru (izvor: *Vlastiti rad autora*)



Slika 5.14 Pisanje odgovora uzvaniku (izvor: *Vlastiti rad autora*)



Slika 5.15 Slanje odgovora uzvaniku (izvor: *Vlastiti rad autora*)

Kreatori događaja mogu vidjeti statistiku odabranih poklona kroz grafičko sučelje. Klikom na svaki kreirani događaj, prikazuje se ekran s brojem dobivenih poklona i pripadajućim grafom (Slika 5.16). Vizualne informacije lakše su razumljive od brojčanih podataka, stoga graf može pomoći u brzom i jasnom uvidu uspješnosti.

Slika 5.16 Prikaz statistike (izvor: *Vlastiti rad autora*)

5.2. Inovacije s obzirom na postojeća rješenja

Aplikacija PoklonManija je dizajnirana da pruži korisnicima jedinstveno i učinkovito iskustvo odabira poklona, što je čini idealnim izborom za one koji znaju što žele dobiti te za one koji žele ispuniti svaku slavljenikovu želju. Njene funkcionalnosti omogućuju korisnicima da odaberu savršeni poklon bez napora i brige. Početni dio izrade ove aplikacije i ovog rada bilo je istraživanje tržišta konkurenata. Postoji mnogo aplikacija za odabir poklona koje su namijenjene samo za jedan specifičan događaj te ih dugoročno nema smisla koristiti. Cilj aplikacije PoklonManija je pružiti korisnicima jedinstvenu, zanimljivu i prvenstveno korisnu aplikaciju. U usporedbi s direktnim konkurentima, poput aplikacija Gifster, VOLO i Wishlistr, aplikacija PoklonManija nudi niz različitih funkcionalnosti. Jedne od glavnih inovacija su anonimni chat i statistika poklanjanja poklona. Korisnici mogu, koristeći anonimni chat, točno specificirati detalje željenog poklona u razgovoru s kreatorom. Kreatori događaja mogu pratiti tijek poklanjanja želja putem statističkih grafova. Također, PoklonManija nudi korisnicima mogućnost korištenja aplikacije za bilo koji događaj u bilo koje doba godine. Mogu sami odabrati kategorije, događaje, mjesto i vrijeme održavanja događaja. Sve spomenute funkcionalnosti zajedno čine aplikaciju PoklonManija drugačijom od svih ostalih konkurenata na tržištu te omogućuju korisnicima da pronađu i dobe željeni poklon za bilo koju prigodu.

Zaključak

Kako bi se stvorio kvalitetan sustav, bilo je ključno ispitati što tržište nudi i koje su potrebe korisnika. Nakon detaljne analize, utvrđeno je da na tržištu nedostaju funkcionalne aplikacije za odabir i definiranje željenih poklona. Većina dostupnih aplikacija je namijenjena samo određenoj kategoriji, poput Božića, što korisnicima nije dovoljno zanimljivo. Upravo ta nedostatna ponuda potaknula je na stvaranje novog sustava koji će zadovoljiti potrebe tržišta i pružiti korisnicima funkcionalnost i dugotrajnu upotrebljivost. Sustav, koji je namijenjen odabiru poklona, implementiran je koristeći programski jezik Kotlin. Proces implementacije je obuhvatio analizu zahtjeva i potrebe korisnika te definiranje funkcionalnosti i implementaciju istih. Ideja izrade ovog sustava je bila osigurati korisnicima jednostavan i učinkovit način za odabir željenih poklona te olakšati proces darivanja koji je nerijetko iscrpljujući. Analizom rezultata korištenja i spomenutim inovacijama prikazane su prednosti ovog sustava, poput poboljšane organizacije događaja te olakšane komunikacije. Međutim, uvijek postoji prostor za napredak i poboljšanja. Stoga je u budućnosti moguće uvesti nove funkcionalnosti poput personaliziranih poklona, direktne kupnje putem aplikacije i slično.

Student vlastoručno potpisuje Završni rad na prvoj stranici ispred Predgovora s datumom i oznakom mjesta završetka rada te naznakom:

„Pod punom odgovornošću pismeno potvrđujem da je ovo moj autorski rad čiji niti jedan dio nije nastao kopiranjem ili plagiranjem tuđeg sadržaja. Prilikom izrade rada koristio sam tuđe materijale navedene u popisu literature, ali nisam kopirao niti jedan njihov dio, osim citata za koje sam naveo autora i izvor, te ih jasno označio znakovima navodnika. U slučaju da se u bilo kojem trenutku dokaže suprotno, spreman sam snositi sve posljedice uključivo i poništenje javne isprave stečene dijelom i na temelju ovoga rada".

U Zagrebu, datum.

Ime Prezime

Popis kratica

SQL	*Structured Query Language*	Strukturni jezik
DHL	*Dalsey Hillblom Lynn*	Njemačka logistička tvrtka
USPS	*United States Postal Service*	Američka poštanska služba
SAD	*Sjedinjene Američke Države*	Savezna republika u Sjevernoj Americi
SMS	*Short Message Service*	Slanje tekstualnih poruka
UI	*User Interface*	Korisničko sučelje
IDE	*Integrated development enviroment*	Integrirano razvojno okruženje
ART	*Android Runtime*	Okruženje za izvođenje aplikacija
URI	*Uniform Resource Identifier*	Jedinstveni identifikator resursa

Popis slika

Slika 3.1 Proces oblikovanja arhitekture (izvor: *Vlastiti rad autora*) 9

Slika 3.2 Dijagram arhitekture Android mobilne aplikacije (izvor: https://developer.android.com/topic/architecture, preuzeto 28. siječnja 2023.)..... 10

Slika 3.3 Projektne datoteke (izvor: *Vlastiti rad autora*) 11

Slika 3.4 Glavni prozor Android Studija (izvor: *Vlastiti rad autora*) 11

Slika 3.5 Prikaz tablica (izvor: *Vlastiti rad autora*) 13

Slika 4.1 Proces analize zahtjeva (izvor: *Vlastiti rad autora*)..... 15

Slika 5.1 Registracija korisnika (izvor: *Vlastiti rad autora*) 32

Slika 5.2 Prijava korisnika (izvor: *Vlastiti rad autora*)..... 33

Slika 5.3 Prikaz kategorija (izvor: *Vlastiti rad autora*)..... 33

Slika 5.4 Prikaz događaja (izvor: *Vlastiti rad autora*)..... 34

Slika 5.5 Kreiranje događaja (izvor: *Vlastiti rad autora*)..... 35

Slika 5.6 Kreiranje događaja (izvor: *Vlastiti rad autora*)..... 35

Slika 5.7 Unos lokacije prodajnog mjesta poklona (izvor: *Vlastiti rad autora*)..... 36

Slika 5.8 Promjena boje gumba (izvor: *Vlastiti rad autora*)..... 36

Slika 5.9 Unos pristupnog koda (izvor: *Vlastiti rad autora*)..... 37

Slika 5.10 Detalji događaja (izvor: *Vlastiti rad autora*)..... 37

Slika 5.11 Detalji događaja (izvor: *Vlastiti rad autora*)..... 38

Slika 5.12 Prikaz lokacije na karti (izvor: *Vlastiti rad autora*) 38

Slika 5.13 Slanje poruke kreatoru (izvor: *Vlastiti rad autora*)..... 39

Slika 5.14 Pisanje odgovora uzvaniku (izvor: *Vlastiti rad autora*)..... 39

Slika 5.15 Slanje odgovora uzvaniku (izvor: *Vlastiti rad autora*) 40

Slika 5.16 Prikaz statistike (izvor: *Vlastiti rad autora*)..... 40

Popis tablica

Tablica 5.1 Rezultati korištenja aplikacije (izvor: *Vlastiti rad autora*)..... 31

Popis kodova

Kod 4.1 SqlHelper klasa 20

Kod 4.2 UserProvider klasa 20

Kod 4.3 Registracija korisnika 20

Kod 4.4 Kreiranje RecyclerView komponente za prikaz kategorija 22

Kod 4.5 Upravljanje kategorijama koristeći Adapter i ViewHolder 23

Kod 4.6 Unos slike iz galerije 25

Kod 4.7 Dodavanje željenih poklona 27

Kod 4.8 Primanje i slanje poruka koristeći MirrorFly 28

Kod 4.9 Implementacija grafa 29

Literatura

Android for Developers. (Siječan 2023). Dohvaćeno iz https://developer.android.com/develop/ui/views/layout/cardview

Android for Developers. (Siječan 2023). Dohvaćeno iz https://developer.android.com/develop/ui/views/layout/recyclerview

Android Studio. (Siječanj 2023). Dohvaćeno iz https://developer.android.com/studio/intro

Bass, & al., e. (2012). *Software Architecture in Practice.* Addison-Wesley Professional.

Christmas Gift List. (Prosinac 2022). Dohvaćeno iz https://play.google.com/store/apps/details?id=com.xmaslist&hl=en_US&gl=US

Gifster. (prosinac 2022). Dohvaćeno iz https://www.giftster.com/

Giftable. (Prosinac 2022). Dohvaćeno iz https://www.giftable.com/

Goodman, J. (2014). Giving Happiness: Do Experiential Gifts Lead to More Happiness? *Association for Consumer Reasearch*.

Platform Architecture. (Siječanj 2023). Dohvaćeno iz https://developer.android.com/guide/platform

Snapfish. (Prosinac 2022). Dohvaćeno iz https://www.snapfish.com/home

SQLite. (Siječanj 2023). Dohvaćeno iz https://www.sqlite.org/index.html

Tinggly. (Prosinac 2022). Dohvaćeno iz https://www.tinggly.com/how-it-works

VOLO. (Prosinac 2022). Dohvaćeno iz https://volowishlist.com/en

wishlistr. (Prosinac 2022). Dohvaćeno iz https://www.wishlistr.com/

ALGEBRA
VISOKO
UČILIŠTE

PROGRAMSKO RJEŠENJE ZA UPRAVLJANJE ODABIROM ŽELJENIH POKLONA

Pristupnik: Luka Kolonić, 0016120507

Mentor: mag. ing. comp. Đani Pašić