

Prototipni razvoj aplikacija na temelju modela

Sandalj, Luka

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:211:070193>

Rights / Prava: [Attribution-NoDerivs 3.0 Unported](#) / [Imenovanje-Bez prerada 3.0](#)

Download date / Datum preuzimanja: **2024-05-20**



Repository / Repozitorij:

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



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

Luka Sandalj

**PROTOTIPNI RAZVOJ APLIKACIJA NA
TEMELJU MODELA**

ZAVRŠNI RAD

Varaždin, 2023.

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

Luka Sandalj

JMBAG: 0016143622

Studij: Poslovni sustavi

PROTOTIPNI RAZVOJ APLIKACIJA NA TEMELJU MODELA
ZAVRŠNI RAD

Mentor/Mentorica:

Prof. dr. sc. Vjeran Strahonja

Varaždin, Rujan 2023.

Luka Sandalj

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 potvrdio prihvatanjem odredbi u sustavu FOI-radovi

Sažetak

Ovaj završni rad je usredotočen na razvoj prototipa aplikacije na temelju modela. U radu su objašnjeni osnovni koraci razvoja skice i prototipa aplikacije, modeli podataka i procesa, te životni ciklus razvoja aplikacije. Uz to, najveći dio posvećen je razvoju prototipa mobilne aplikacije na temelju primjera, izradi skice, definiranju funkcionalnosti, modela entiteta i veza, te izradi dijagrama slučajeva korištenja. Korištenjem više alata zaokružuje se tema prototipnog razvoja aplikacije na temelju modela. Izrađene su skice, modeli i krajnji protoip mobilne aplikacije za razmijenu biljaka. Cilj ove aplikacije je jednostavna i brza razmjena biljaka između korisnika, te pružanje intuitivnog i funkcionalnog korisničkog sučelja.

Ključne riječi: prototip, skica, Figma, funkcionalnost, entitet, aplikacija, razvoj.

Sadržaj

Sadržaj.....	iii
1. Uvod.....	1
2. Skica.....	2
2.1. Definicija skice	2
2.2. Vrste skica	3
2.2.1. Skice niske vjernosti prikaza	3
2.2.2. Skice srednje vjernosti prikaza.....	3
2.2.3. Skice visoke vjernosti prikaza.....	4
2.3. Prednosti i nedostaci izrade skice	4
2.4. Alati za izradu skica	5
2.4.1. Excalidraw.....	5
2.4.2. Balsamiq Wireframes	5
2.4.3. Sketch.....	6
3. Prototip.....	7
3.1. Definicija prototipa	7
3.2. Prednosti i nedostaci izrade prototipa	7
3.3. Alati za izradu prototipa.....	7
3.3.1. Figma.....	7
3.3.2. Adobe XD.....	8
3.3.3. Proto.io.....	9
3.3.4. Oracle Apex	10
4. Modeliranje podataka.....	11
4.1. Svrha izrade modela podataka.....	11
4.2. Modeliranje entiteta i veza.....	11
4.2.1. Entiteti.....	11
4.2.2. Veze.....	12

4.2.3. Atributi.....	12
5. Metode i tehnike modeliranja procesa.....	13
5.1. BPMN (Business Process Model and Notation)	13
5.2. Dijagram slučajeve korištenja.....	14
6. Životni ciklus razvoja aplikacije	15
6.1. Planiranje projekta	15
6.2. Prikupljanje i analiza zahtjeva	15
6.3. Dizajniranje	15
6.4. Kodiranje i implementacija	16
6.5. Testiranje aplikacije.....	16
6.6. Puštanje aplikacije u rad	16
6.7. Održavanje aplikacije	16
7. Metoda izrade prototipa	18
7.1. Konceptualni razvoj prototipa	18
7.2. Evolucijski razvoj prototipa	18
8. Skica i prototip u okviru životnog ciklusa razvoja aplikacije	19
8.1. Uloga skice u razvoju aplikacije	19
8.2. Uloga prototipa u razvoju aplikacije.....	19
9. Razvoj prototipa aplikacije na primjeru – Biljkoljupci	20
9.1. O aplikaciji	20
9.2. Modeliranje entiteta i veza.....	21
9.3. Modela poslovnog procesa na primjeru.....	22
9.4. Dijagrami slučajeve korištenja.....	22
9.5. Izrada skice.....	25
9.6. Izrada prototipa	29
9.6.1. Boje i tipografija.....	29
9.6.2. Veze prototipa	30
10. Funkcionalnosti prototipa	32
11. Zaključak	38

12.	Literatura	39
-----	------------------	----

1. Uvod

U ovom radu će se razmatrati izazovi i koraci u razvoju prototipa aplikacije, istražujući kako teorijski i praktično primijeniti koncepte iz područja razvoja softverskih rješenja. U teorijskom dijelu rada detaljno ćemo analizirati skicu i prototip kao ključne komponente razvojnog procesa. Kroz modeliranje podataka i procesa, opisivat će se kako precizno definirati zahtjeve i funkcionalnosti aplikacije. Također će se kroz rad proučavati životni ciklus razvoja aplikacije, istražujući kako se skica i prototip uklapaju u taj kontekst. U praktičnom dijelu rada, fokus će biti na konkretnom primjeru razvoja mobilne aplikacije. Ovaj dio istraživanja obuhvaća detaljan pregled same aplikacije, uključujući modeliranje entiteta i veza koje će biti ključne za njeno funkcioniranje. Dijagrami slučajeva korištenja pomoći će u razumijevanju kakva je interakcija korisnika s aplikacijom i koji su njihovi osnovni zadaci. Nadalje, prikazana će biti izrađena skica aplikacije, koristeći vizualne alate za definiranje njenog izgleda i korisničkog sučelja. Kroz proces izrade prototipa, detaljno će se razmatrati kako testirati i iterirati na dizajnu aplikacije prije nego što krenemo u fazu razvoja. Kroz ovaj rad će se pokušati dublje objasniti kako teorijski koncepti o skici, prototipiranju i modeliranju podataka i procesa ulaze u praksu razvoja aplikacija, a praktični primjer mobilne aplikacije pružit će konkretan uvid u ove korake.

2. Skica

2.1. Definicija skice

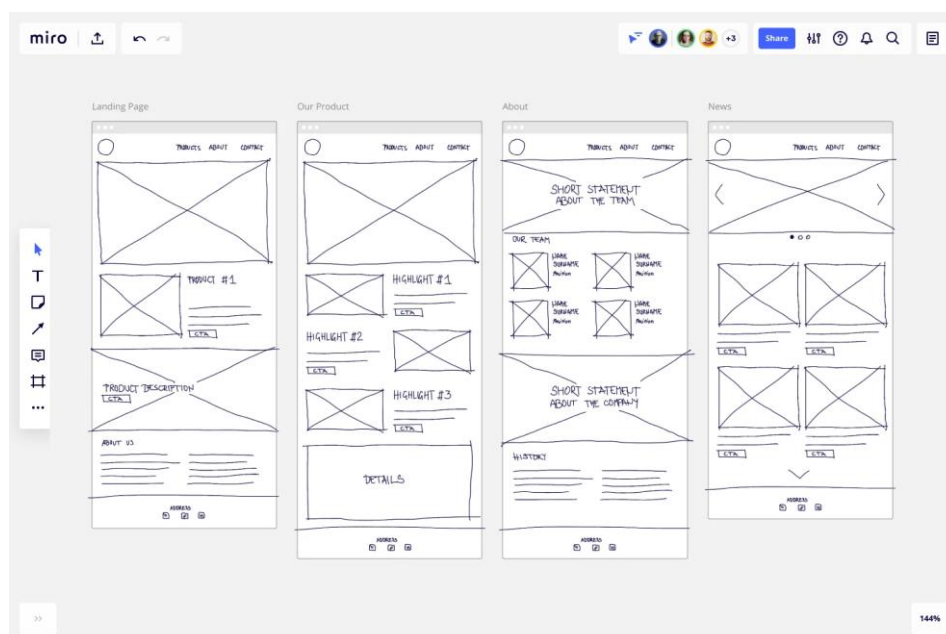
Skica je dvodimenzionalni vizualni prikaz web ili mobilne aplikacije, odnosno aplikacijskog sučelja ili izgleda budućeg proizvoda. Skica obično prikazuje samo funkcionalnost, a ne pravi stil i vizualne elemente konačnog proizvoda. Zato većina skica izgleda jednostavno. Umjesto boja se koriste sivi i neutralni tonovi, izbjegava se korištenje slika, već se samo prikazuju pozicije najmnjenjene budućim slikama, te se također izbjegava tipografija. Ideja izrade skica je da se pomoću njih stvaraju moguća rješenja, te se njima postiže dodirna točka komunikacije između korisnika i aplikacije. Preciznije uz pomoć skice se može komunicirati s korisnikom vezano uz funkcionalnosti usluge te predloženu strukturu aplikacije. Skice obično objedinjuju sadržaj, navigaciju, korisničko sučelje, te interakciju korisnika s aplikacijom. Svrha skice je da odgovori na nekoliko ključnih pitanja. Koji će elementi biti prikazani u korisničkom sučelju, kako će elementi biti organizirani, kako će sučelje funkcionirati i kako korisnik komunicira s aplikacijom.

Skice se mogu dijeliti prema razini detalja prilikom vizualizacije buduće aplikacije i prema metodi izrade. Dije se na tri razine: skice niske vjernosti prikaza (eng. *Low-fidelity wireframes*), skice srednje vjernosti prikaza (eng. *Mid-fidelity wireframes*) i skice visoke vjernosti prikaza (eng. *High-fidelity wireframes*). Skica je početni korak komunikacije između korisnika u procesu razvoja aplikacije. Postoji nekoliko vrsta korisnika. Klijent uz pomoć skice postiže bolje razumijevanje kako će izgledati budući proizvod, te lakše može procijeniti odgovaraju li mu predložena rješenja i jesu li njegove potrebe zadovoljene. Pregledava je li zadovoljan s elementima korisničkog sučelja, čime se postiže lakše ispravljanje problema. Dok je razvoj proizvoda još u fazi skiciranja, kod promjena ne dolazi do velikih troškova, te je puno lakše raditi promjene u ovoj fazi dok još nema izrađenog programskog koda. Dizajneri, voditelj projekta i razvojni programeri su unutarnji korisnici, odnosno korisnici koji zajedno sudjeluju u razvoju proizvoda koji će kasnije predstaviti klijentu ili krajnjem korisniku. Dizajnerima su skice kostur strukture uz pomoć kojeg mogu lakše prolaziti kroz iduće korake u životnom ciklusu razvoja proizvoda. Razvojni programeri koriste skice kako bi stekli razumijevanje funkcionalnosti rješenja i tehničkih zahtjeva. Voditelj projekta vodi brigu da su svi ostali sudionici razvoja na istoj strani, odnosno da je postignuto razumijevanje između zahtjeva klijenta i zahtjeva dizajnera i programera.

2.2. Vrste skica

2.2.1. Skice niske vjernosti prikaza

Skice niske vjernosti prikaza su prvi i najlakši korak pri dizajniranju aplikacije. Izrada skice niske vjernosti je produktivan i jeftin, ali kreativni proces koji može ponuditi mnogo izvanrednih rješenja za razvoj budućeg proizvoda. Skice niske vjernosti prikaza su korisne i za klijente i za timove. S jedne strane, omogućuju timovima da brzo donose najbolje rješenje za krajnjeg korisnika. S druge strane, niskovjerne skice su dovoljne da klijentu pomogne odrediti hoće li raditi promjene funkcionalnosti korisničkog sučelja.



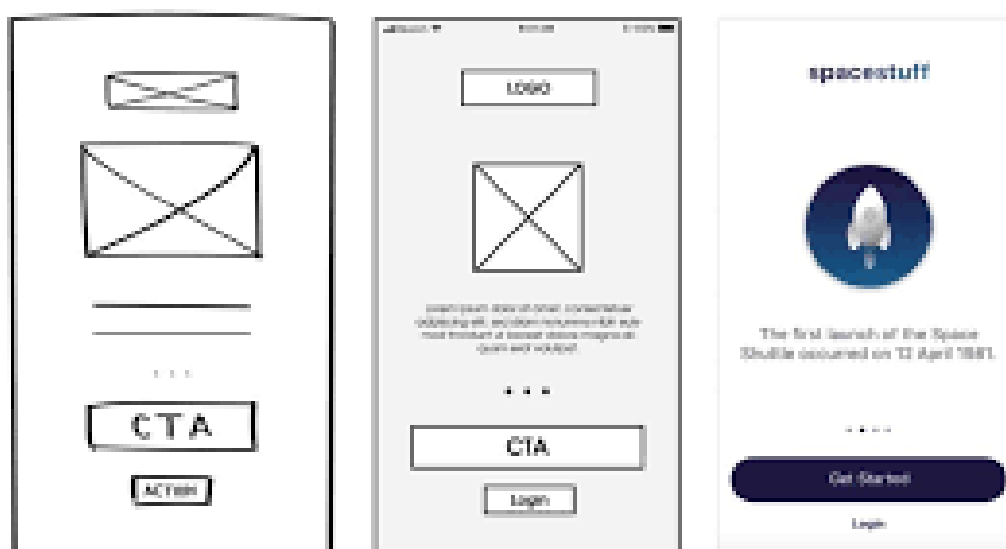
Slika 1: Primjer besplatnog predloška skice niske vjernosti prikaza u alatu Miro

2.2.2. Skice srednje vjernosti prikaza

Skice srednje vjernosti prikaza se najčešće koriste, te one imaju točniji prikaz izgleda od skica niske vjernosti prikaza. Ova razina skica i dalje izbjegava implementaciju slika ili tipografije, ali se posvećuje više detalja određenim komponentama, a funkcije se međusobno jasno razlikuju. Različite težine teksta također se mogu koristiti za odvajanje naslova i sadržaja. Iako su elementi i dalje crno-bijeli, nijanasama sive se postižu i prenose vizualne istaknutosti pojedinih elemenata. Iako su još uvijek relevantne u ranim fazama proizvoda, skice srednje vjernosti prikaza se često stvaraju pomoću digitalnih alata kao što su Balsamiq ili Excalidraw.

2.2.3. Skice visoke vjernosti prikaza

Skice visoke vjernosti prikaza su najviši stupanj izrade skica. Skice visoke vjernosti pružaju najvišu razinu detalja. Visokovjerne skice sadrže informacije o svakom specifičnom elementu na stranici, uključujući tipografiju, radnje povezane s interaktivnim elementima i dimenzije. Preciznije skice visoke vjernosti prikaza su vizualno čišće i realističnije, odnosno bliže krajnjem izgledu sučelja. Prethodni nasumični tekst (eng. *Lorem Ipsum*) zamijenjen je stvarnom tipografijom. Detalji se naglašavaju uz pomoć tonova sive boje. Primjenjuju se različite veličine i vrste fontova, čime se postiže bolja vizualizacija krajnjeg proizvoda.



Slika 2: Usporedba razina vjernosti prikaza skice (Izvor: Tsvetelina Lazarova, 2018.)

2.3. Prednosti i nedostatci izrade skice

Velika prednost izrade skica je da daju ranu vizualnu predodžbu koja se može koristiti za pregled s klijentom. Korisnici ju također mogu vidjeti kao rani mehanizam povratnih informacija za testiranje upotrebljivosti prototipa. Skice su lakše izmjenjive od konceptualnog dizajna, te nakon što ih klijent ili korisnik odobre, one daju povjerenje dizajneru. Iz praktične perspektive, skice osiguravaju da su sadržaj i funkcionalnost stranice pravilno postavljeni na temelju klijentovih/korisničkih potreba. Kako projekt napreduje, skice se mogu koristiti kao dobar predmet za raspravu između članova projektnog tima kako bi se dogovorili o viziji i opsegu projekta.

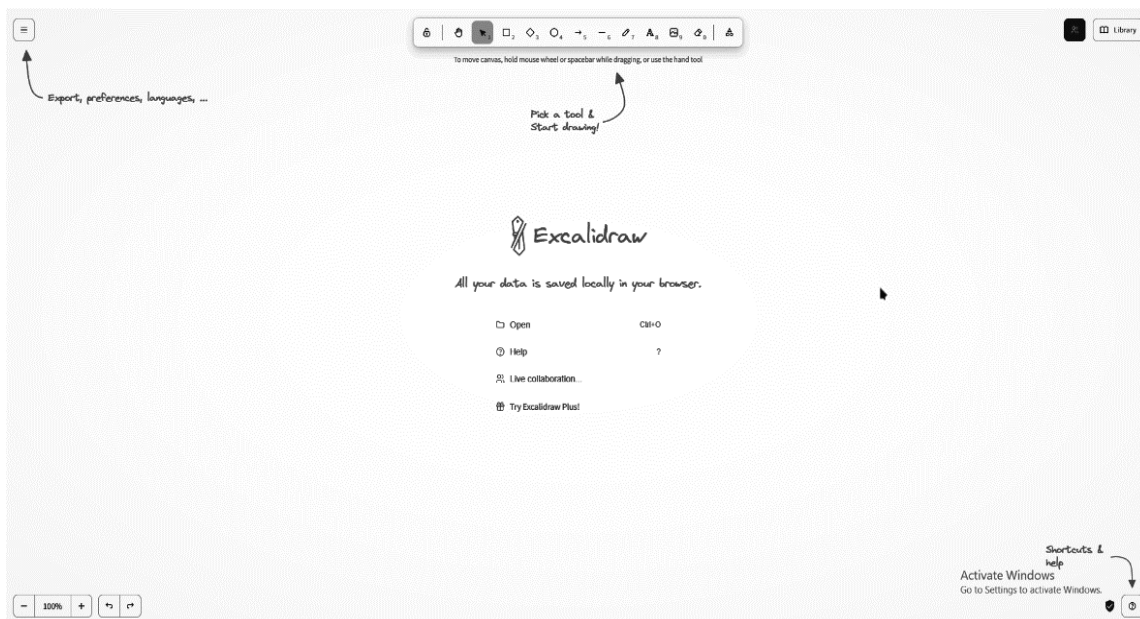
Budući da skice ne uključuju dizajn niti objašnjavaju tehničke implikacije, klijentu može biti teško shvatiti koncept. Dizajner će također morati prevesti skice u prototip, tako da je komunikacija često potrebna kako bi se objasnilo zašto su elementi aplikacije postavljeni onako kako jesu. Također, kada se dodaje sadržaj, u početku bi ga moglo biti previše tipografije koja bi mogla stvarati probleme, tako da dizajner i tekstopisac moraju blisko surađivati kako bi ovaj segment odgovarao prilikom dizajniranja.

2.4. Alati za izradu skica

2.4.1. Excalidraw

Excalidraw je bijela ploča (eng. *whiteboard*), alat koji omogućuje jednostavno skiciranje dijagrama koji imaju dojam crtanja rukom.

Njegove ključne prednosti su jednostavnost i lako rukovanje. Dizajniran je za brzo stvaranje jednostavnih grafika i skica.



Slika 3: Radna površina alata Excalidraw

2.4.2. Balsamiq Wireframes

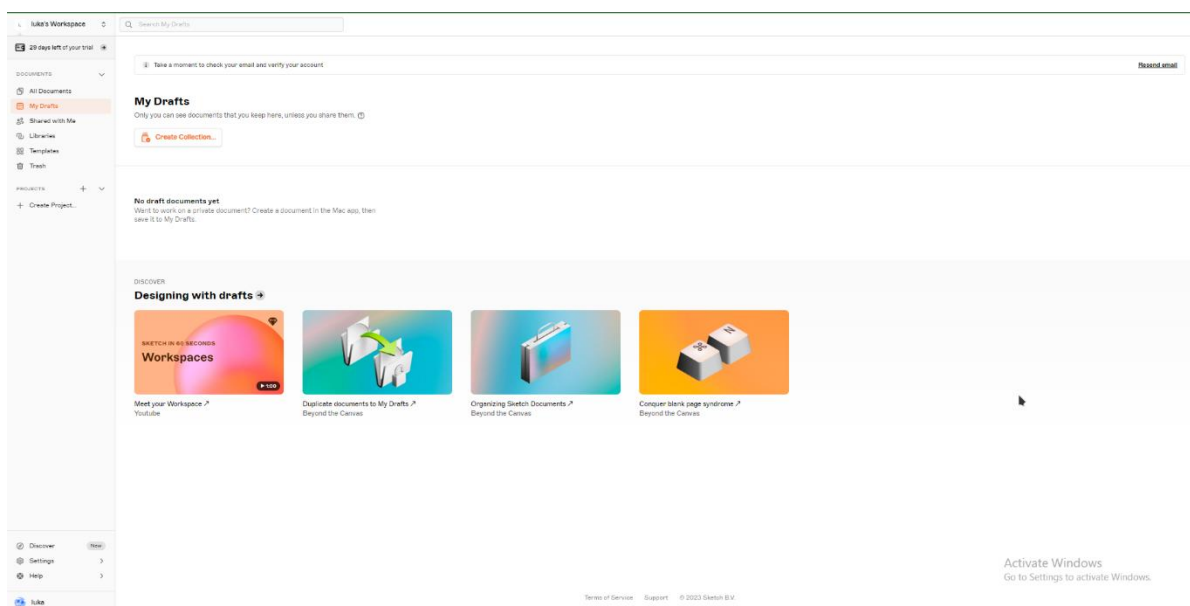
Balsamiq Wireframes je brz alat za stvaranje i skiciranje korisničkog sučelja. To je alat za izradu skice niske vjernosti prikaza koji reproducira iskustvo skiciranja po bilježnici ili bijeloj ploči, ali pomoću računala. Također, nudi opciju online alata Balsamiq Cloud koji omogućuje dizajniranje i pregled korisničkih sučelja na Internetu.



Slika 4: Skica mobilne aplikacije u Balsamiq Wireframes (Izvor: vlastita izrada, 2021.)

2.4.3. Sketch

Sketch je platforma za dizajniranje koju je 2010. godine izradila nizozemska tvrtka Sketch B.V. Namijenjena je korisnicima iOS-a, odnosno Sketch je Mac aplikacija za dizajniranje, skiciranje, izradu prototipa i još mnogo toga. Sketch je osvojio nagradu Apple Design Award 2012. Također ima web aplikaciju koja omogućuje korisnicima dijeljenje datoteka na mreži.



Slika 5: početna stranica alata Sketch

3. Prototip

3.1. Definicija prototipa

Prototip je približna replika onoga kako će izgledati krajnji rezultat proizvoda. Obično je cilj lansirati ga vrlo brzo i implementirati rješenja bez kodiranja, ako je to moguće. Oni uključuju većinu konačnog dizajna korisničkog sučelja i interakcija koje će imati gotov proizvod. Izrada prototipa jedan je od najboljih načina za stvaranje snažnog dizajna korisničkog iskustva (eng. *User experience – UX*).

Smisao prototipa je pomoći u eksperimentiranju i testiranju pretpostavki, kako one ne bi dovele do skupih pogrešaka u proizvodnji. To je osobito istinito u digitalnom prostoru gdje se stvari poput boja, tema i raznih dizajna mogu lako mijenjati.

3.2. Prednosti i nedostaci izrade prototipa

Izrada prototipa, kao i svaki korak razvoja aplikacije, ima svoje prednosti i nedostatke.

Izradom prototipa postižu se smanjeni troškovi i skraćeno vrijeme razvoja. Razvoj prototipa uključuje samog korisnika u proces razvoja aplikacije, čime se postiže veće korisničko zadovoljstvo. Također, prilikom razvoja prototipa prednost je dobivanje kvantitativnih povratnih informacija s kojima razvojni timovi mogu raspolagati, te mogućnost poboljšanja i nadograđivanja sustava od strane programera.

S negativne strane prototipi mogu dovoditi do nepotpune dokumentacij i nedovoljne analize. Prototipi mogu ograničiti programere, te oni mogu postati previše vezani uz sam prototip. Sa korisničkog pogleda, prototipi znaju postaviti korisniku očekivanje da će krajnja aplikacija biti jednaka kao i sam prototip.

3.3. Alati za izradu prototipa

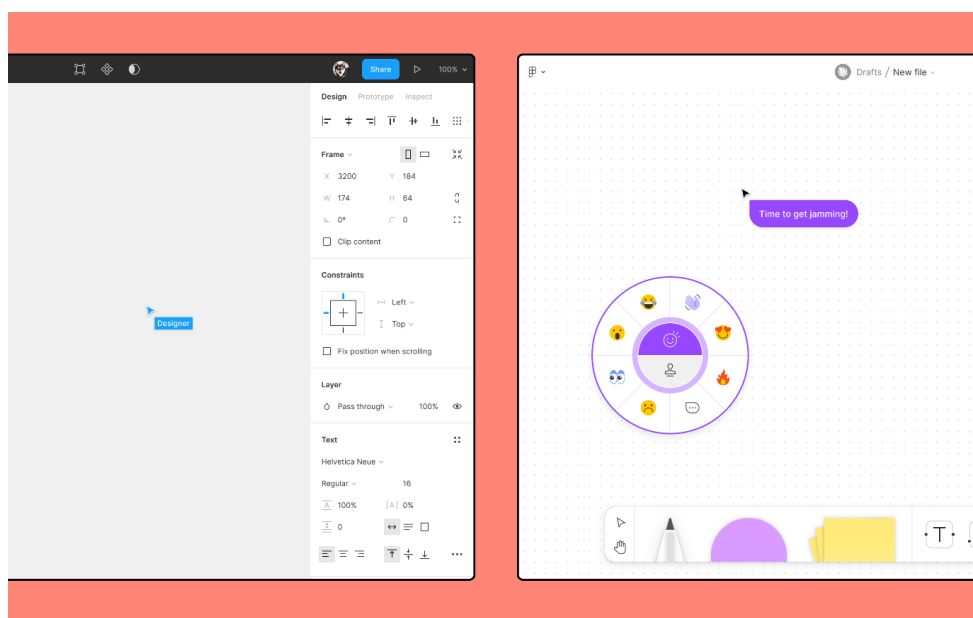
3.3.1. Figma

Misija Figue je učiniti dizajn dostupnim svima. Figma se sastoji od dva proizvoda koji pomažu ljudima da vizualno izraze svoje ideje, te da uz pomoć ove platforme dizajniraju željeni produkt. Figma se dijeli na Figma design i FigJam.

Figma design je namijenjena korisnicima za izradu, dijeljenje i testiranje dizajna za web aplikacije, mobilne aplikacije i druge digitalne proizvode i usluge. Ciljanja skupina korisnika

alata Figma su dizajneri, voditelji proizvoda, pisci i programeri, te ona pomaže svima koji su uključeni u proces dizajna da doprinesu, daju povratne informacije i brže donose bolje odluke.

FigJam korisnicima daje mogućnost izrade ploča na Internetu (eng. *online whiteboards*) kojima svatko u radnom okruženju može pristupiti. FigJam se također često koristi za sastanke, brainstorming, izradu dijagrama, planiranje i istraživanje. Funkcionalnosti koje koristi FigJam su izrada teksta, raznolika oblikovanja, izrada crteže, umetanje i oblikovanje slike, ljepljive bilješke i druge elemente za vizualno predstavljanje ideja. Ideja Figma je da se Figma design i FigJam koriste zajedno, jer zajedno sadrže sve za cijeli proces dizajna. Prilikom razvoja aplikacije skiciranje svoje ideje može se raditi u FigJamu. Nakon skiciranja, Figma design nudi funkcionalnosti za nastavak dizajniranja i izradu prototipa.



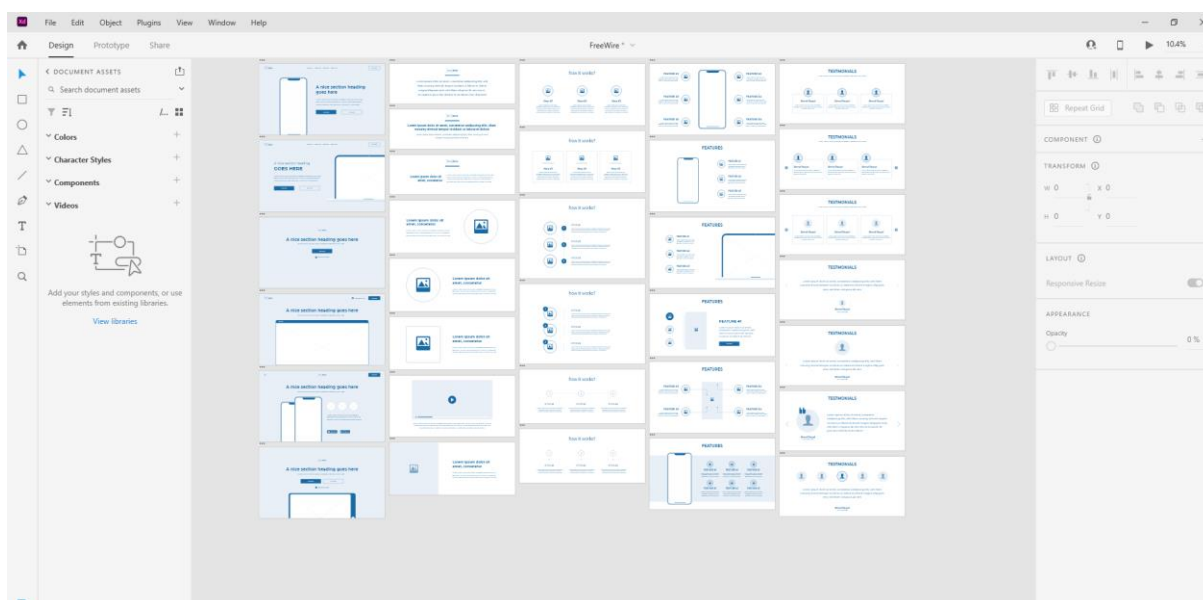
Slika 6: Radna površina alata Figma i FigJam

3.3.2. Adobe XD

Adobe XD je snažan i jednostavan alat za izradu prototipa i dizajna iskustava temeljenih na vektorima. Dostupan je na Mac i Windows sustavima, tako da Adobe XD ispunjava potrebe timova s kompatibilnošću na različitim platformama. Dizajneri i dizajnerski timovi diljem svijeta koriste Adobe XD za suradnju tijekom procesa dizajniranja, od koncepta i izrade skica, izrade prototipa do predaje developerima. Ima mogućnost dijeljenja dizajna i specifikacija, čime članovi tima mogu pregledavati i ostavljati povratne informacije izravno u preglednik, u obliku komentara. Komponente u Adobe XD-u pojednostavljaju proces ponovne upotrebe zajedničkih elemenata poput gumba i navigacije te održavaju njihovu sinkronizaciju u cijelom dizajnu. S glavnom komponentom stvorenom, svaka promjena napravljena na

glavnoj automatski će se pojaviti u svim instancama. Adobe XD također nudi prilagodljivo skaliranje koje razumije vaš raspored, tako da je dizajniranje za novi format jednostavno. U Adobe XD-u može se dizajnirati potpuno audio korisničko iskustvo, koristeći naredbe glasa zajedno s govorom i reprodukcijom zvuka. Tako da Adobe XD ima mogućnost izrade produkta koji se može koristiti putem primjerice, Amazon Alexe ili Google Asisstant-a.

Adobe XD nudi dvije moćne funkcije za dizajniranje s pregledima u stvarnom vremenu. Ako dizajnirate na pametnom telefonu ili tabletu, možete ga priključiti na svoje računalo i dobiti pregled u stvarnom vremenu na uređaju. Ako nemate uređaj za koji dizajnirate, možete koristiti prozor za pregled u Adobe XD-u kako biste generirali pregled u stvarnom vremenu dok dizajnirate.

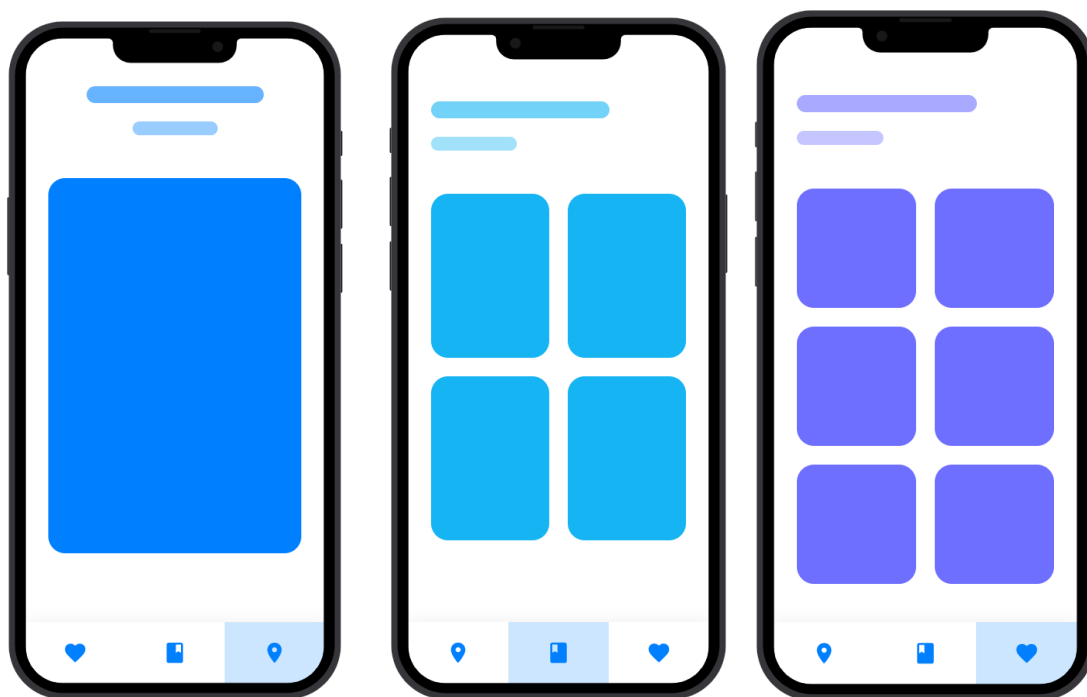


Slika 7: Primjer besplatnog predloška u alata Adobe XD

3.3.3.Proto.io

Slogan proizvođača alata Proto.io je „Oživite svoju ideju u tren oka. Prototipna rješenja za sve vaše potrebe. Za UX dizajnere, poduzetnike, upravitelje proizvodima, marketinške stručnjake i sve koji imaju sjajnu ideju“. Ukratko, Proto.io je platforma za izradu prototipa aplikacija pokrenuta 2011. godine, a razvila ju je tvrtka PROTOIO Inc.

Prilikom izrade prototipa Proto.io nudi sljedeće funkcionalnosti: knjižnice komponenti korisničkog sučelja, predloške, obrasce, biblioteku ikona, biblioteku zvučnih efekata, audio i video komponente, prilagođene komponente, komponente HTML koda, jednostavan „drag and drop upload“, učitavanje slike, alate za maskiranje, standardne i prilagođene predloške veličina uređaja itd.



Slika 8: Primjer besplatnog predloška prototipa mobilne aplikacije u alatu Proto.io

3.3.4. Oracle Apex

Oracle Apex je platforma niske razine kodiranja za razvoj koja omogućava izgradnju skalabilnih i sigurnih poslovnih aplikacija s vrhunskim značajkama koje se mogu implementirati bilo gdje. Oracle APEX je najpopularnija svjetska platforma za razvoj poslovne aplikacije s niskom razinom koda koja vam omogućuje izradu skalabilnih, sigurnih web i mobilnih aplikacija s vrhunskim značajkama koje se mogu implementirati bilo gdje. Oracle Apex nudi razvoj od koncepta uporabljivog prototipa. Pomoću Apexa mogu se definirati nove strukture podataka, zajedno s oglednim podacima, koristeći Quick SQL. Quick SQL omogućuje generiranje SQL skripti na temelju jednostavne skraćene sintakse. Ove se skripte mogu spremi i jednostavno pokrenuti unutar vašeg Oracle APEX okruženja za stvaranje tablica baze podataka, okidača i prikaza. Mogu se dodavati dodatne funkcije, kao što je kontrola pristupa, izvješćivanje o aktivnostima i još mnogo toga, kako bi se u kratkom vremenskom roku izradila potpuno opremljena aplikacija. Razvoj s Oracle APEX-om temelji se na brznoj iteraciji aplikacije i pregledavanju iste s krajnjim korisnicima. Developeri mogu jednostavno raditi s krajnjim korisnicima i implementirati promjene i zahtjeve za poboljšanja. APEX je jednako brz u održavanju i poboljšanju postojećih aplikacija kao i u izradi prototipa.

4. Modeliranje podataka

Model podataka je pojednostavljeni prikaz sustava koji opisuje skupove podataka koji su potrebni za rad sustava, attribute koji opisuju te podatke i odnose između podataka. Omogućuje prikaz stanja sustava. Modele podataka izrađuju projektanti informacijskog sustava, a njihovu ispravnost potvrđuju krajnji korisnici. Rezultati procesa modeliranja temelje se na metodologiji. Također, model podataka je osnovni preduvjet za izradu baze podataka.

4.1. Svrha izrade modela podataka

Bez unaprijed proizvedene dokumentacije i izloženih modela podataka za izgradnju budućih informacijskih sustava, razvoj informacijskog sustava može koštati i trajati dulje od planiranog. Bez dokumentiranog modela podataka informacijskog sustava njegovo održavanje postaje složenije i zahtjevnije i postaje primarni posao inženjera. Promjene u poslovnim sustavima bit će sporije i teže provedive ako ne sadrže dokumentaciju koja prikazuje podatkovni model organizacije njihovih informacijskih sustava. Bez dokumentacije modela podataka, nekontrolirana redundancija podataka i problemi s njihovim ažuriranjem mogu se pojaviti zbog loše organizacije podataka.

4.2. Modeliranje entiteta i veza

Ideja modela entiteta i veza (eng. *Entity-relationship model – ERA model*) je da taj model uz pomoć dijagrama prikazuje stvarni svijet. Izumitelj ovog modela je bio 1976. Peter Chen. Ovo je model za definiranje i opis onih podataka koji su stvoreni i potrebni procesima u području poslovanja. U praktičnom dijelu rada biti će prikazan ERA model na primjeru, gdje će detaljnije biti objašnjen odnos entiteta, veza i atributa.

4.2.1. Entiteti

Svaki entitet ima poseban atribut koji identificira entitet, odnosno svaki entitet ima svoj identitet. Entiteti su međusobno povezani jedni s drugima. Odnosno minimalno dva entiteta mogu biti u međusobnoj vezi. Entiteti se dijele na jake i slabe entitete. Jaki entiteti mogu postojati samostalno, odnosno nisu ovisni o nekom drugom entitetu, dok su slabi entiteti ovisni o postojanositi nekog drugog entiteta.

4.2.2. Veze

Veze služe za povezivanje dvaju entiteta međusobno ili za povezivanje različite pojave istog tipa entiteta. Vezu unutar modela definiraju tip i kardinalnost. Veza prema redu može biti unarna ili binarna. Dok kardinalnost veze, preciznije odnos između dvaju entiteta, može biti u omjeru 1:1, 1:M ili N:M.

4.2.3. Atributi

„Atribut (obilježje, svojstvo) je podatak koji opisuje entitet ili omogućava njegovo prepoznavanje.“ [1, str. 11] Atributi se mogu dijeliti na opisne, izvedene i identifikacijske.

5. Metode i tehnike modeliranja procesa

Model procesa je pojednostavljeni prikaz znanja o sustavu u smislu poslovnih procesa koje on obavlja. Opisuje procese koji se odvijaju na podacima i kroz djelovanje tih procesa postiže ciljeve promatranog sustava. Prikazuje skup procesa koji izvode promjene sustava i tvore izlaz sustava. Model procesa prikazuje skup procesa nad podacima, odnosno skup poslova na skupu podataka. „Modeli procesa su formalizirani opisi hijerarhijskih struktura procesa, unutarnje logike procesa, međusobnih odnosa procesa, događaja u objektnim sustavima, te odnosa procesa i okolice.“ [2, str. 3]

5.1. BPMN (Business Process Model and Notation)

BPMN (eng. *Business Process Model*) predstavlja strukturirani pristup za bilježenje, dizajn, provedbu, dokumentiranje, mjerenje, nadzor i kontrolu procesa, bez obzira na to jesu li automatizirani ili ne, s ciljem ostvarivanja poslovnih ciljeva i strategije tvrtke. S korištenjem BPM-a, tvrtke postižu bolje rezultate brže i s većom fleksibilnošću zahvaljujući sustavnom i svjesnom upravljanju svojim procesima. BPM omogućava usklađivanje procesa s poslovnom strategijom, poboljšavajući njihove performanse putem optimizacije kako unutar pojedinih odjela tako i izvan granica tvrtke. Naglasak je na razumijevanju i unaprjeđenju cijelog procesa, a ne samo njegovih pojedinačnih dijelova.

Kako bi se procesi bolje automatizirali, razvijena je BPMN (Business Process Model Notation) kao grafička notacija za modeliranje i opisivanje poslovnih procesa. Ova standardizirana grafička notacija olakšava komunikaciju i bolje razumijevanje poslovnih procesa, čime se stvara zajednički jezik između poslovne strane i informacijske tehnologije. Svaki model poslovnog procesa se sastoji od objekata toka, a to su: događaji, aktivnosti i skretnice. Aktivnost ima funkciju opisivanja svakog rada unutar poslovnog procesa, aktivnost je označena pravokutnikom sa zaobljenim rubovima i unutar njega nalazi se opis posla koji ta aktivnost izvršava. Aktivnosti mogu biti ponavljajuće, kompenzacijske, višestruko sekvencijalne i višestruko paralelne. Svaki proces ima minimalno dva događaja, početni i završni. Događaj je radnja koja se događa, ali nema trajanje i potrošnju resursa, prilikom izvođenja procesa. Događaj se označuje simbolom kruga. Skretnicama se određuje kojim će se redoslijedom izvoditi aktivnosti u procesu. Postoji više vrsta skretnica, a najčešće korištene su ekskluzivna i paralelna skretnica.

5.2. Dijagram slučajeve korištenja

Dijagrami slučajeve korištenja se koriste za prikaz ponašanja određenih dijelova sustava, cijelog sustava ili određene klase sustava. Sustav se opisuje pomoću korisnika sustava koji definiraju sudionike i pomoću ciljeva. Preciznije, dijagrami slučajeve korištenja prikazuju ponašanje sustava isključivo iz percepcije korisnika, te ne objašnjavaju izvedenost funkcionalnosti unutar prototipa. Ovaj oblik dijagrama odgovara na pitanje što sustav radi, ali ne i kako radi. Dijagram slučajeve korištenja je definiran sa tri osnovna koncepta: slučaj korištenja, veza i učesnik.

Slučaj korištenja se definira kao skup akcija koje neki sustav može izvoditi u međudjelovanju s učesnicima. Veza povezuje odnos između učesnika i slučajeve korištenja, te objašnjava njihovo međudjelovanje i ponašanje. Postoje četiri tipa veza: pridruživanje, generalizacija, uključivanje i proširenje. Pridruživanje je veza kojom se povezuju sudionici sa slučajevima korištenja u kojima sudjeluju. Generalizacija je veza koja povezuje dva učesnika ili dva slučaja korištenja. Uključivanjem se povezuju dva slučaja korištenja tako da jedan slučaj u tijeku svog izvođenja u potpunosti izvede drugi, uključeni slučaj korištenja. Vezom proširenja se povezuju dva slučaja korištenja, gdje jedan proširuje funkcionalnost drugog.

Učesnik ili sudionik je segment izvan samog promatranog sustava koji je sa sustavom u međudjelovanju. Učesnik može biti stvarna osoba, npr. korisnik ili neki drugi sustav, npr. skladište. Postoje dva tipa učesnika, primarni i sekundarni. Primarni učesnik je onaj koji pokreće slučaj korištenja, dok je sekundarni učesnik onaj koji sudjeluje nakon što je slučaj korištenja pokrenut.

6. Životni ciklus razvoja aplikacije

Životni ciklus razvoja aplikacije je sustavan proces izgradnje aplikacije koji osigurava kvalitetu i ispravnost iste. Cilj ovog procesa je proizvesti visokokvalitetni proizvod koji ispunjava očekivanja kupaca. Razvoj sustava trebao bi biti dovršen u unaprijed definiranom vremenskom okviru i trošku. Životni ciklus razvoja se sastoji od detaljnog plana koji objašnjava kako planirati, izgraditi i održavati određenu aplikaciju. Svaka faza životnog ciklusa ima vlastiti proces i rezultate koji se prenose na sljedeću fazu. Životni ciklus razvoja aplikacije se još naziva i životni ciklus razvoja softvera (eng. *Systems development life cycle - SDLC*). Životni ciklus razvoja aplikacije se danas dijeli na sedam faza.

6.1. Planiranje projekta

Prva faza životnog ciklusa razvoja aplikacije se odnosi na planiranje projekta. U ovoj fazi se postavlja ključno pitanje, Što želimo postići? Ovo je bitna faza razvoja iz razloga što u ovoj početnoj fazi projektni tim procjenjuje troškove i definira zahtjeve i prepreke buduće aplikacije.

6.2. Prikupljanje i analiza zahtjeva

Prikupljanje informacija i zahtjeva klijenta ili korisnika, te njihova detaljna analiza čine drugu fazu razvoja aplikacije. U ovoj fazi se sa klijentom razgovara o svakom detalju i specifikacije budućeg produkta. Nakon prikupljenih informacija i zahtjeva, projektni tim analizira iste. Prilikom analize zahtjeva projektni tim bi trebao imati na umu pitanja poput ostvarivosti dizajna zahtjeva ili pak izvedivosti i način prilikom pisanja koda. Neki od primjera zahtjeva su: sigurnost aplikacije, dizajn, hardver, operativni sustav, programiranje itd. Glavni cilj ove faze je da svi sudionici razvoja aplikacije razumiju zahtjeve od osnovnih zahtjeva sve do najdetaljnijih.

6.3. Dizajniranje

Tijekom faze dizajniranje aplikacije projektni tim pažljivo provjerava zadovoljava li pripremljeni softver sve zahtjeve krajnjeg korisnika, te je li projekt tehnološki, praktično i financijski izvediv za klijenta. Nakon što se programer odluči za najbolji pristup dizajnu, on odabire programske jezike prikladne za aplikaciju, npr. Java ili Python. Nakon izrade projektnih specifikacija, sve zainteresirane strane pregledavaju plan i daju komentare i prijedloge.

Prikupljanje povratnih informacija dionika i njihovo uključivanje u dokument je ključan, jer mala pogreška može dovesti do prekoračenja troškova.

6.4. Kodiranje i implementacija

Ovo je faza u kojoj se počinje kodirati. Dizajna se prevodi na računalnu čitljiv jezik. U ovoj četvrtoj fazi SDLC-a, zadaci su podijeljeni na segmente i dodijeljeni različitim programerima. Programeri će tada početi graditi cijeli sustav pisanjem koda koristeći prethodno odabrane programske jezike. Ova faza se smatra jednom od najdužih u životnom ciklusu razvoja aplikacije. Programeri trebaju određene unaprijed definirane smjernice za kodiranje i alate za programiranje kao što su tumači, kompajleri, program za ispravljanje pogrešaka za implementaciju koda. U slučaju da ima potrebe za promjenom određenih funkcionalosti, programer konzultira voditelja tima i analitičara kako bi zajednički našli odgovarajuće rješenje problema.

6.5. Testiranje aplikacije

Nakon izgradnje aplikacije dolazi faza testiranja. U ovoj fazi tim za testiranje testira funkcionalnosti aplikacije. Provođenjem testiranja se osigurava da aplikacija u potpunosti radi prema svim zahtjevima klijenta. Nakon testiranja, tim za osiguranje kvalitete (eng. *Quality Assurance* - QA) i tim za testiranje mogli bi pronaći neke greške ili nedostatke i obavijestiti programere. Razvojni tim zatim ispravlja pogreške i šalje ga timu za QA na ponovno testiranje. Ovaj proces se ponavlja i traje sve dok aplikacija u potpunosti ne ispunjava sve zahtjeve klijenta.

6.6. Puštanje aplikacije u rad

Nakon što testiranje završi i proizvod bude spreman, bit će pušten korisnicima. Veličina projekta određuje složenost izvedbe. Korisnici tada dobivaju vodič i dokumentaciju koja im pomaže u korištenju aplikacije.

6.7. Održavanje aplikacije

Stvarni problem počinje kada kupac stvarno počne koristiti razvijeni sustav i treba ga povremeno rješavati. Zadnja faza životnog ciklusa razvoja aplikacije je održavanje. U ovoj fazi

se vodi briga o razvijenoj aplikaciji. Cilj je da aplikacija bude u skladu s tehnološkim i korisničkim okruženjem. Taj cilj se ostvaruje pravovremenim ažuriranjem aplikacije.

7. Metoda izrade prototipa

Metoda izrade prototipa je iterativna razvojna metoda. Planirani zahtjevi se dodatno usavršavaju kroz svaku iteraciju, a implementirani sustav se detaljno verificira. Prototip se može razviti konceptualno i evolucijski. Konceptni prototip nikada ne postaje konačni sustav i razvija se isključivo za testiranje različitih zahtjeva ili za testiranje različitih načina implementacije određenih zahtjeva. Evolucijski prototip postat će konačni sustav. Kontinuirano je nadograđivan kroz uzastopni razvoj generacija prototipa.

7.1. Konceptualni razvoj prototipa

Ovaj prototip nikada nije finalni proizvod. Cilj konceptualnog razvoja prototipa je raspravljati i dodatno poboljšavati razumijevanje zahtjeva korisnika ili klijenta. Više prototipa može biti predstavljeno klijentu istovremeno. Svaki od ovih prototipovi mogu biti rezultat različitog pristupa rješenja istog problema ili mogu predstavljati rješenja za različite dijelove istog problem. Često se konceptualni prototipovi nazivaju brzi, eksperimentalni ili istraživački prototipovi. Riječ koncept u kontekstu razvoja prototipa je predstavljena kao pomoć u razvoju i modificiranju zahtjeva i specifikacija. [3]

7.2. Evolucijski razvoj prototipa

Ova vrsta prototipa bi trebala biti preteča do konačnog proizvoda. Drugim riječima, inicijalni prototip će na kraju postati konačni proizvod. Svaka sljedeća generacija prototipova je poboljšanje u odnosu na svog prethodnika. Krajnja razina funkcionalnosti se postiže tamo gdje je prototip postaje finalni proizvod. Evolucijski prototipiranje iznimno dobro funkcionira za softverska rješenja, budući da su troškovi proizvodnje i distribucije minimalni u usporedbi s većinom drugih proizvoda. Mnogi programeri softvera proširuju ovaj evolucijski pristup prototipiranju čak i dalje. U suštini, smatraju svaku novu verziju svog proizvoda evolucijskim prototipom. To znači da se tradicionalna faza održavanja postaje sastavni dio 'definiranja i razumijevanja problema'. Drugim riječima, rješenje nikada nije pronađeno, već se kontinuirano traži u uzastopnim verzijama i ažuriranjima. Ovom metodom priznaje se da softverski proizvodi nikada nisu savršeni, ali da programeri kvalitetnog softvera moraju neprestano težiti tom savršenstvu. Također, ovom metodom priznaje se da su developeri za održavanje i podršku ljudi koji su kontinuirano uključeni u komunikaciju s korisnicima softvera i stoga bi trebala biti iznimno važna njihova uloga u definiranju i razumijevanju problema i očekivanju proizvoda. [3]

8. Skica i prototip u okviru životnog ciklusa razvoja aplikacije

8.1. Uloga skice u razvoju aplikacije

Skica je važan element u razvoju aplikacija, prije svega može vizualizirati ideje korisnika i razvojnog tima. Stoga se skice mogu koristiti kao vizualne bilješke dizajnera za proučavanje i razumijevanje vizualnih preferencija korisnika, čime se postiže jednostavnija izrada dizajna aplikacije. Također, skica se još kroisti kao predložak za klijente koji su željeli vidjeti dizajn aplikacije što je prije moguće i temeljem skica od dizajnera traže izradu prototipa buduće aplikacije.

8.2. Uloga prototipa u razvoju aplikacije

Osim izrade prototipa aplikacije koji nam omogućuje uvid u funkcionalnost buduće aplikacije, u većini slučajeva fokus je na izradi visokokvalitetnog dizajna koji nam daje pozitivno korisničko iskustvo. Ukoliko se prototip ne izradi, budući će korisnici biti nezadovoljni jer se softverski proizvod može vidjeti samo u fazi razvoja i više se ne može mijenjati. Nezadovoljstvo korisničkog iskustva ima posljedice kao što su dodatni troškovi razvoja aplikacije i dodatni utrošak vremena te u konačnici odbijanje aplikacije od strane korisnika. Uloga prototipa je značajno poboljšao kvalitetu i kvantitetu komunikacije između programera i analitičara.

9. Razvoj prototipa aplikacije na primjeru – Biljkoljupci

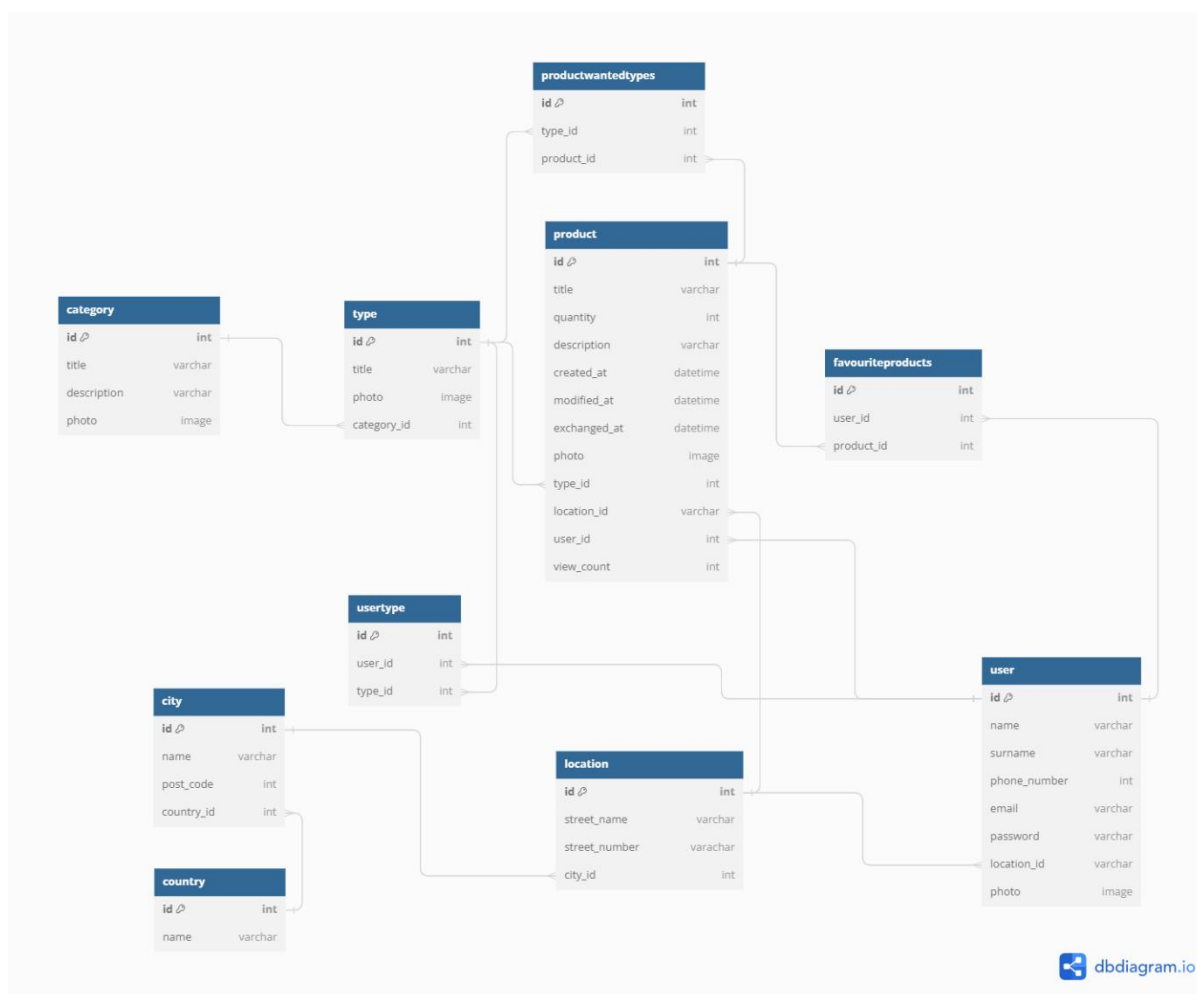
9.1. O aplikaciji

Praktični dio rada će se baviti razvojem i izradom prototipa mobilne aplikacije Biljkoljupci. Biljkoljupci je aplikacija koje bi trebala služiti korisnicima za jednostavnu razmijenu biljaka putem svojih mobilnih uređaja. Korisnici imaju mogućnost pretraživanja i nuđenja biljaka za razmijenu. Razvoj prototipa ove mobilne aplikacije se sastoji od tri ključna dijela. Prvo, prototip se razvija na temelju modela podataka i procesa, zatim dolazi faza prototipnog razvoja na temelju modela sučelja i korisničkog iskustva. Unutar faze izrade modela podataka i procesa će biti prikazan model entiteta i veza (eng. *Entity-relationship model - ERA model*) prototipa, koji će detaljno opisivati međusobnu povezanost entiteta ovog prototipa i slučajevi korištenja koji će prikazivati funkcionalne zahtjeve prototipa, odnosno što aplikacija treba raditi. Unutar faze prototipnog razvoja na temelju modela sučelja i korisničkog iskustva će biti prikazana izrađena skica aplikacije u programu Balsamiq Wireframes, pomoću koje se u sljedećem koraku izrađivao i sam prototip mobilne aplikacije. Izrada prototipa će prikazivati povezanost korisničkih sučelja izrađenih pomoću alata Figma.

Aplikacija korisniku nudi registraciju, prijavu i odjavu, definira lokaciju korisnika koja filtrira ponudu biljaka na temelju blizine. Lokacija se dijeli na ulicu, grad i državu, ovisno o želji korisnika. Prilikom pretrage proizvoda, odnosno ponude, korisnik može pretraživati prema kategoriji biljaka, tipu biljaka i samom nazivu biljke. Kategorija se odnosi na vrste biljaka, kao npr. vrtne biljke, začinsko bilje, grmovi i ljekovito bilje. Tip biljke je podkategorija vrste biljaka koja prikazuje u koju skupinu određena biljka, odnosno proizvod pripada, npr. ruže. Krajnja podkategorija pretraživanja ponude su sami nazivi biljaka, npr. Engleska ruža koja pripada tipu ruže, dok tip ruža pripada kategoriji vrtne biljke. Protip također korisniku nudi funkcionalnost slikovne i opisne objave proizvoda kojeg nudi drugima. Detaljnije, korisnik na jednostavan način, odabirom slike iz albuma ili slikanjem u stvarnom vremenu, može nuditi svoji proizvod. Opis proizvoda definira kategoriju, tip i naziv biljke, količinu biljaka koju korisnik nudi, te tip biljaka koji potražuje zauzvrat. Korisniku se, na njegovom profilu, nudi mogućnost uređivanja fotografije profila, te pregled i uređivanje njegovih objava.

9.2. Modeliranje entiteta i veza

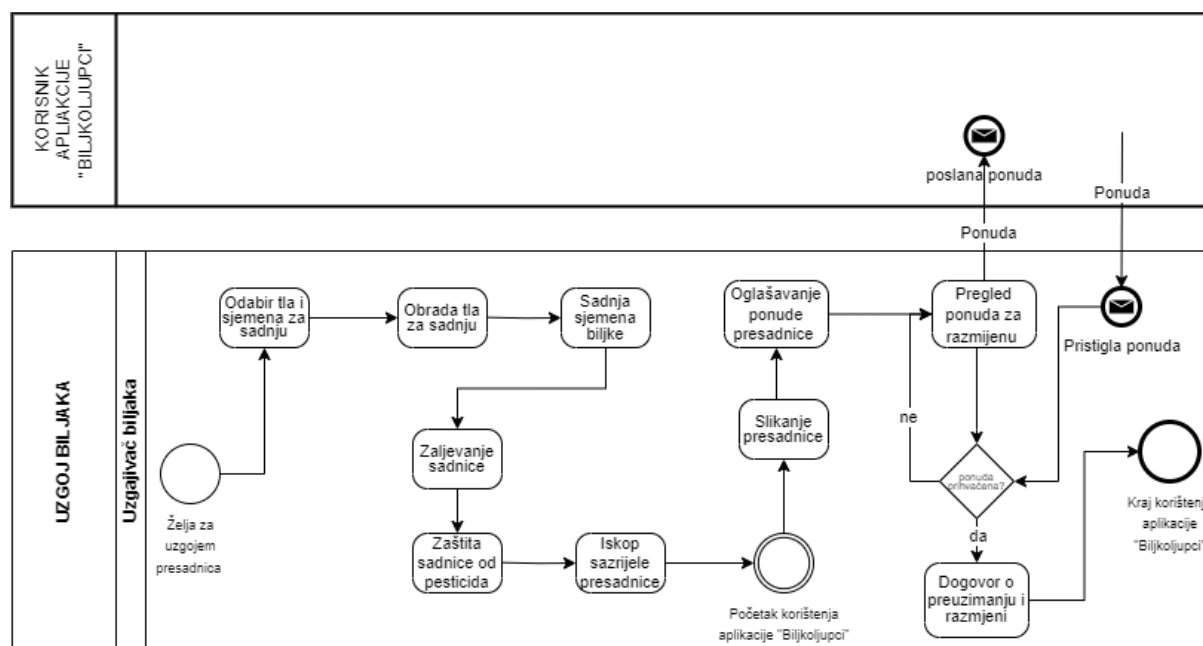
Slika 9 prikazuje ERA model prototipa aplikacije Biljkoljupci. Ovaj model se sastoji od deset entiteta: kategorije (eng. *category*), tipa (eng. *type*), proizvoda (eng. *product*), korisnika (eng. *user*), tipa korisnika (eng. *usertype*), lokacije (eng. *location*), grada (eng. *city*), države (eng. *country*), omiljenih proizvoda (eng. *favourite products*) i željenih tipova proizvoda (eng. *product wanted types*). Entiteti su međusobno povezani vezama jedan naprema jedan (1:1), jedan naprema više (1:M) i više naprema više (M:N). Entiteti su definirani atributima i vidljivo je da je identifikacijski dokument (eng. *Identity Document - ID*) primarni ključ svakog entiteta. Na primjeru entiteta *country*, *city* i *location* može se opisati njihov odnos. Više različitih gradova može biti u jednoj državi, a više lokacija postoji u jednom gradu. Preciznije, ovaj dio modela podataka se primjenjuje kod funkcionalnosti odabira lokacije prilikom prijave u aplikaciju. Korisnik tako može definirati svoju lokaciju na više razina. Unos adrese neke lokacije sa sobom veže grad, a grad veže državu u kojoj se nalazi lokacija. Ukoliko korisnik želi unijeti samo grad za lokaciju, neće vezan biti uz atribut naziva ulice i kućnog broja kojik su atributi entiteta *location*.



Slika 9: ERA model prototipa aplikacije Biljkoljupci

9.3. Modela poslovnog procesa na primjeru

Na slici 10 možemo vidjeti primjer procesa uzgoja i razmijene biljaka (presadnica) u obliku BPMN dijagrama koji prikazuje prikaz stvarnog svijeta i trenutak u kojem dolazi do potrebe za korištenjem mobilne aplikacije „Biljkoljupci“. Dijagram se sastoji od dva sudionika, uzgajivača biljke i korisnika aplikacije, tri događaja i deset aktivnosti. Početni događaj je želja ili potreba uzgajivača za uzgojem presadnica. U prvoj aktivnosti uzgajivač odabire pogodno tlo i sjeme za sadnju. U sljedeće dvije aktivnosti obrađuje i sadi sjeme željene sadnice. Zatim, zaljeva i zaštićuje sadnicu od pesticida. Zadnja aktivnost prije novog događaja, početka korištenja aplikacije, je iskop sazrijele presadnice. Kako je presadnica spremna za sađenje na novo tlo, uzgajivač višak biljaka određene sorte odlučuje zamjeniti za drugi tip biljaka, te u ovom događaju dolazi do potrebe za aplikacijom „Biljkoljupci“. Korisnik unutar aplikacije slika presadnicu i oglašava ponudu. Pregledavanjem ponude uzgajivač može poslati ponudu ili korisnik može vidjeti objavljenu ponudu uzgajivača. Ukoliko se oba sudionika dogovore za razmijenu dolazi se do završnog događaja jer više nema potrebe za korištenjem aplikacije u određenom primjeru.

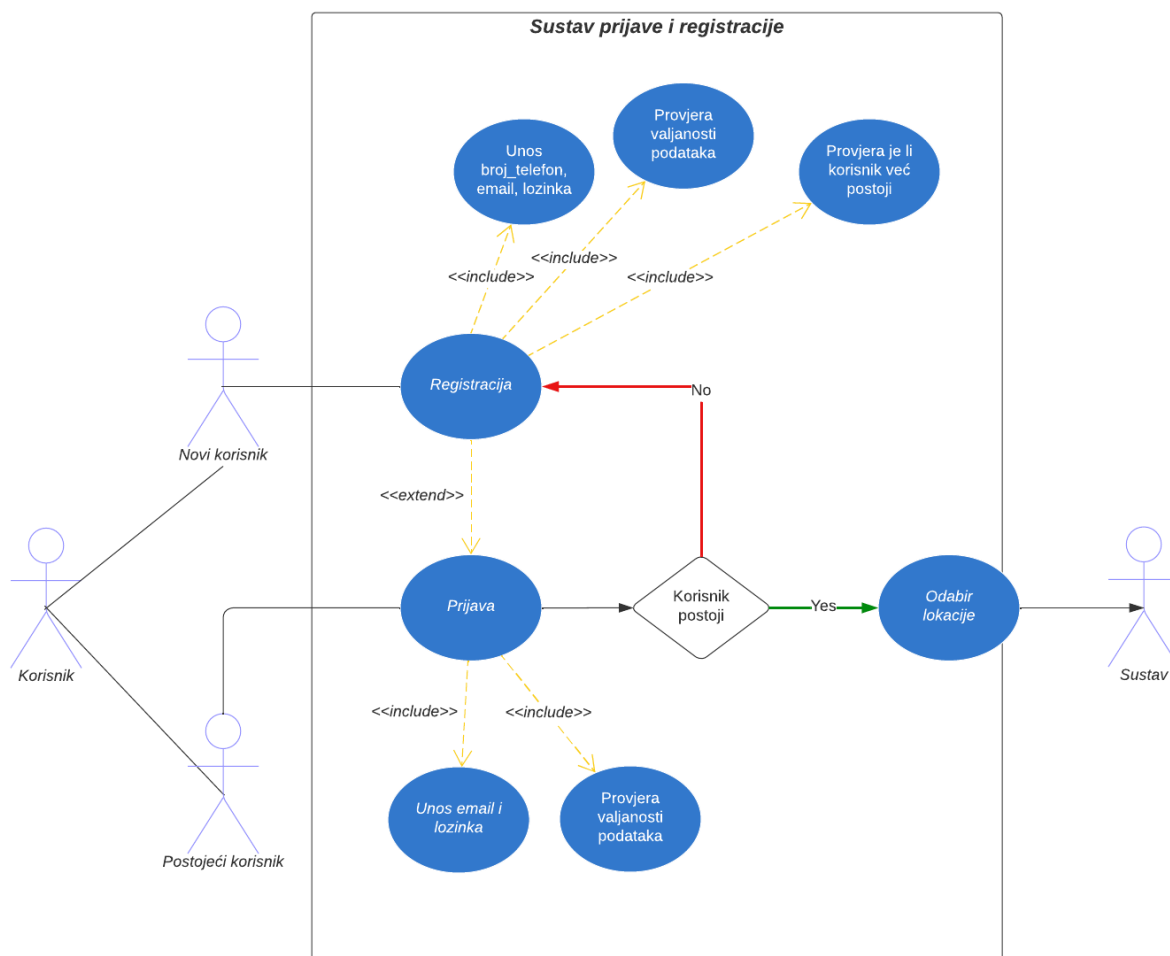


Slika 10: BPMN dijagram na primjeru aplikacije „Biljkoljupci“

9.4. Dijagrami slučajeva korištenja

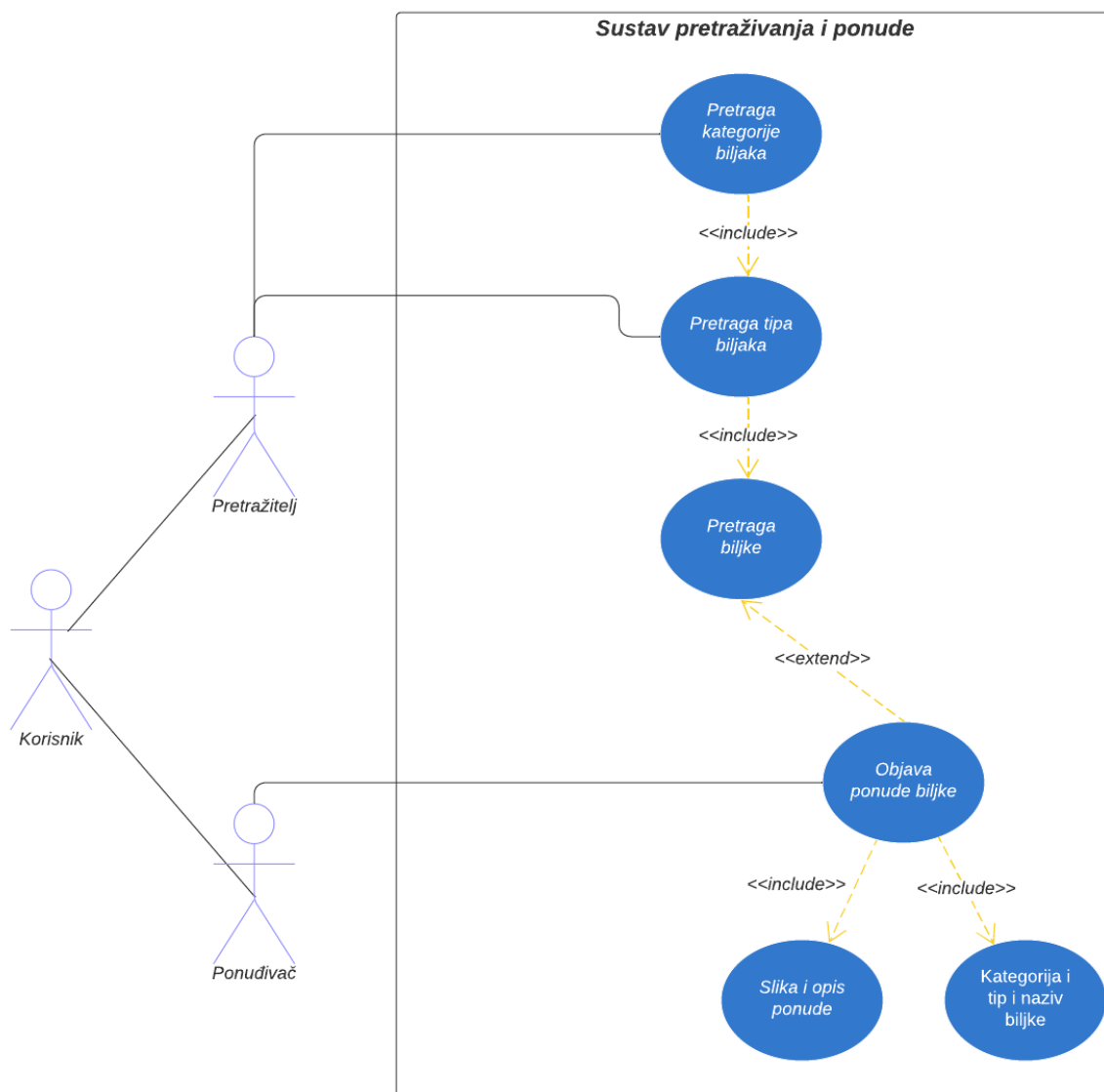
U ovom poglavlju su prikazani dijagrami slučajeva korištenja na primjeru prototipa mobilne aplikacije Biljkoljupci. Slika 10 prikazuje dijagram prijave i registracije u sustav. Primarni korisnici ovog sustava su Novi korisnik i Postojeći korisnik, a sekundarni sudionik je

Sustav. Korisnik prisutpa sustavu na način da se prilikom ulaska u aplikaciju mora prijaviti. Ukoliko je korisnik postojeći, prilikom prijave unosi email i lozinku i pristupa slučaju korištenja Odabir lokacije, a zatim i samom sustavu. Ukoliko je korisnik nov, pristupa registraciji koja sadrži podatke za registraciju, te se u ovom slučaju korištenja provjerava postojanost korisnika i valjanost unešenih podataka. Registracija korisnika u slučaju da korisnik nema postojeći račun proširuje Prijavu.



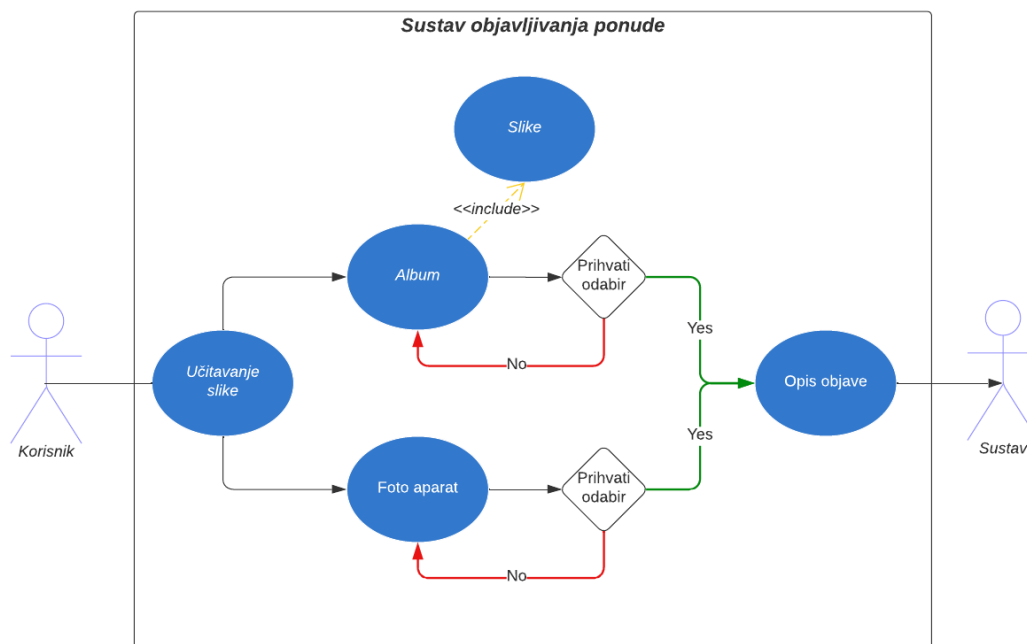
Slika 11: Dijagram slučajeva korištenja prijave i registracije

Dijagram slučajeva korištenja sa slike 11 opisuje način pretraživanja i ponude. Korisnik je učesnik koji se dijeli na pretražitelja i ponuđača. Pretražitelj ima mogućnost pretrage kategorije biljaka, koja sadrži pretragu biljaka, koja sadrži pretragu određene biljke. Korisnik ponuđač je u međuodnosu sa slučajem objave ponude biljke. Ponuda biljke sadrži sliku i opis, te kategoriju, tip i naziv biljke. Objava ponude biljke proširuje slučaj korištenja pretraga biljke.



Slika 12: Dijagram slučajeva korištenja pretraživanja i ponude

Dijagram slučajeva korištenja na slici 12 prikazuje preciznije sustav objavljivanja ponude. Korisnik učitava sliku koja će definirati buduću objavu. Ovaj slučaj korištenja je u međuođnosu sa Albumom i Foto aparatom. Ukoliko korisnik želi slikati proizvod buduće ponude, bira foto aparat. Album sadrži slike i ukoliko korisnik želi odabrati prethodno slikanu sliku, odabire Album. Ova dva slučaja korištenja se ponavljaju sve dok korisnik nije zadovoljan s odabranom slikom biljke. Nakon odabira slike, korisnik stvara opis za objavu i objavljuje ju u sustav.



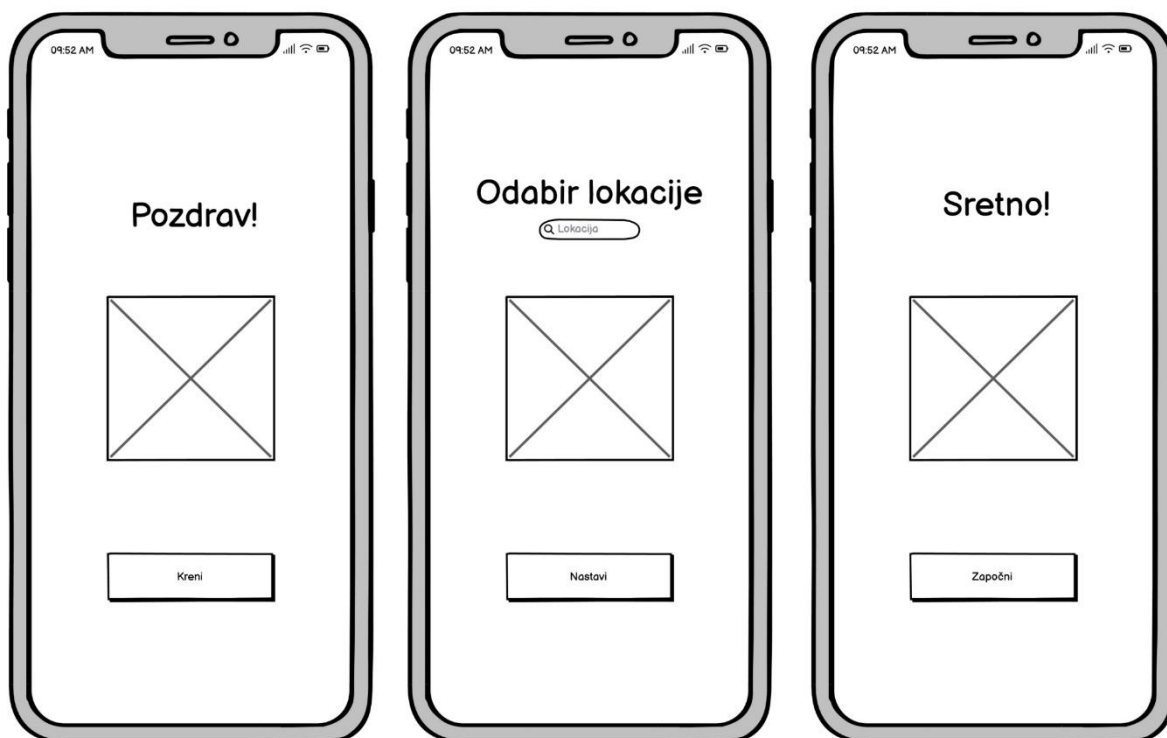
Slika 13: Dijagram slučajeva korištenja objavljivanja ponude

9.5. Izrada skice

Svrha izrade ove skice je postizanje lakšeg dizajniranja krajnjeg prototipa mobilne aplikacije Biljkoljupci. Skica je izrađivana u alatu Balsamiq Wireframes koji služi za vrlo brzo u jednostavnu izradu skica niske i srenje razine. Skica prikazana u idućim slikama je prema vrsti skica srenje razine, što se može prepoznati korištenjem isključivo crne i bejele boje, te sivih tonova. Osnovni gumbi i naslovi ekrana su ispunjeni namjenskim tekstom dok su neki tekstovi ne definirani samo „pošarani“. O tim segmentima se još u ovoj fazi izrade skice ne razmišlja, te je dovoljno samo prikazati kako će se u izradi budućeg prototipa na tom mjestu nalaziti određeni tekst. Također, umjesto slika su postavljena prazna polja koja ukazuju na okvirnu veličinu i željenu poziciju budućih fotografija. Skica je podijeljena na šest osnovnih funkcionalnosti: registracija i prijava korisnika, odabir lokacije korisnika, pregled naslovne stranice, kategorija i podkategorija, pregled oglasa, opis i objavljivanje ponude i pregled vlastite objave.



Slika 14: Skica registracije i prijave korisnika u alatu Balsamiq Wireframes



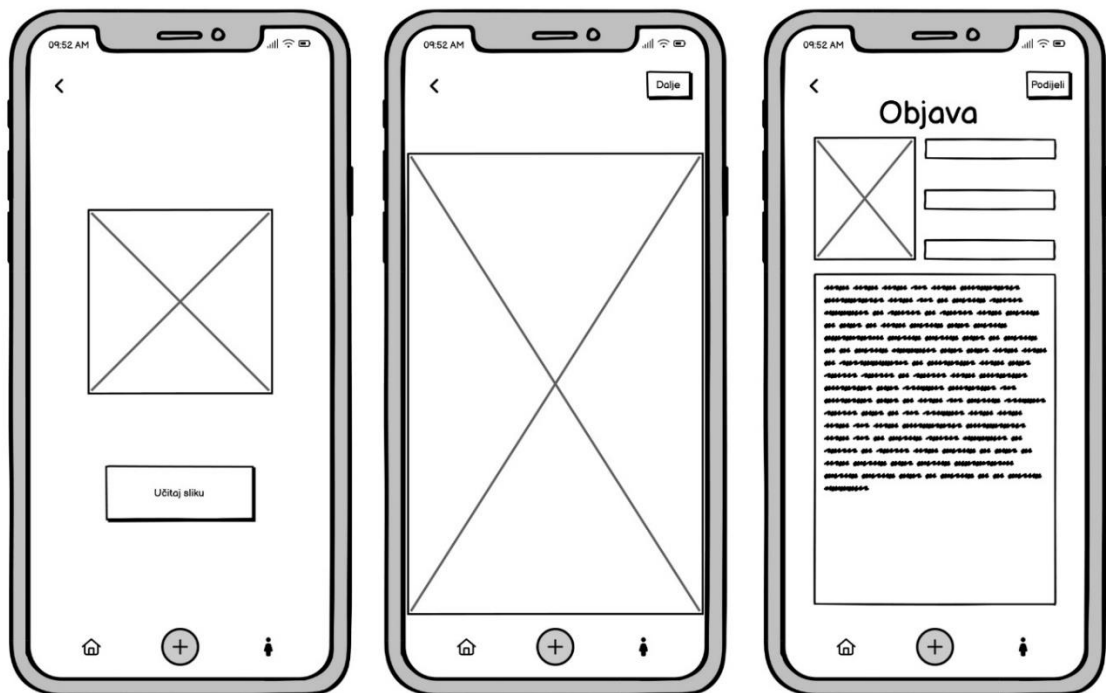
Slika 15: Skica odabira lokacije korisnika u alatu Balsamiq Wireframes



Slika 16: Skica naslovne stranice, kategorija i podkategorija u alatu Balsamiq Wireframes



Slika 17: Skica oglasa u alatu Balsamiq Wireframes



Slika 18: Skica opisa ponude i objavljivanja ponude u alatu Balsamiq Wireframes

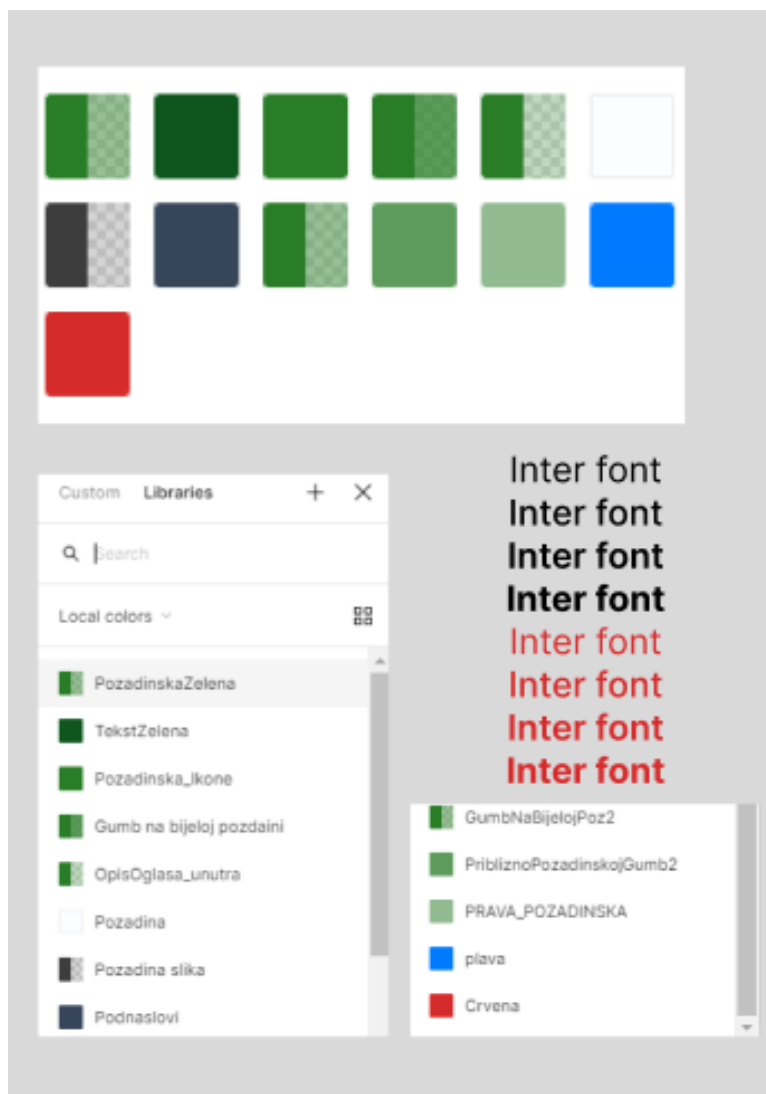


Slika 19: Skica vlastite objave u alatu Balsamiq Wireframes

9.6. Izrada prototipa

9.6.1. Boje i tipografija

Na 19, ispod teksta, je prikazana paleta boja korištena u izradi prototipa. Koristilo se 13 nijansi boja, od kojih je 8 varijacija zelene boje koja obilježava identitet ove aplikacije. Za glavnu pozadinsku boju ekrana korištena je bijela boja šifre #FFFFFF sa vrijacijama prozirnosti izražene u postotcima. Nijanse crvene i plave boje su korištene pretežito na opcionalnim gumbima s kojima se postiže intuitivnost korištenja aplikacije, crvena obilježava gumb za brisanje i odjavu, a plava gumb odabira i gumb za spremanje i dijeljenje sadržaja.



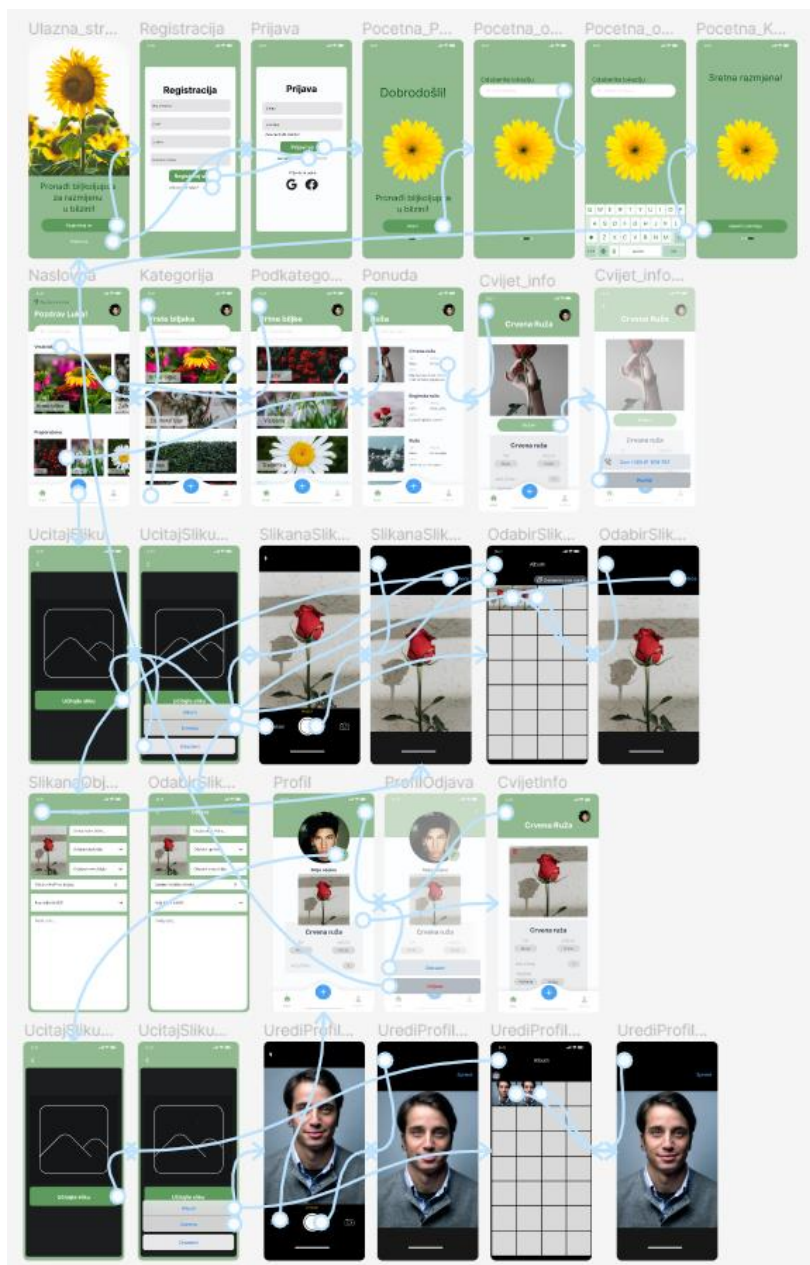
Slika 20: prikaz boja i tipografije u alatu Figma

Tipografija prototipa je definirana fontom, debljinom fonta i veličinom fonta izraženom u pikselima. Prototip je izrađivan na podlozi veličine Iphone uređaja. Kroz cijeli prototip se

koristi jedna vrsta fonta pod nazivom Inter. Prilikom definiranja veličine fonta u obzir se uzimala najmenja korisnicima iOS uređaja, te se iz tog razloga veličina fontova proteže između 16 piksela i 48 piksela. Preporuča se, radi izbjegavanja povećanja polja za unos podataka , izbjegavanje fonta manjeg od 16 piksela kako bi se izbjegla potreba da korisnik moram ručno smanjivat prikaz nakon upisa teksta u određeno polje. Preciznije, veličine fonta korištene u izradi prototipa su: 16, 17, 18 20, 24, 30, 32 i 48 piksela. Debljina fonta je prikazana na slici ispod, gdje se može vidjeti kako se koristilo četiri debljine fonta: Regular, Medium, Semi Bold i Bold, redom od najtanje do najdeblje debiljine.

9.6.2. Veze prototipa

Kako bi se postigao smisao izrade prototipa, korisnička sučelja se povezuju. Međusobnim povzivanjem različitih ekrana se postiže prikaz zamišljenog rada buduće aplikacije. Na slici ispod su prikazane veze između ekrana, koje definiraju i simuliraju čemu određeno područje na ekranu služi i kamo ono vodi. Na primjeru veze s prvog ekrana se može vidjeti kako pritiskakna gumb „Registrij se“ vodi na obrazac za registraciju korisnika. Više veza može voditi na isto odredište, tako pritisak na tipku „Home“ na bilo kojem ekranu ima istu završnu destinaciju, da korisnika vrati na naslovnu stranicu aplikacije. Što je detaljnija povezanost između funkcionalnosti i tipki to se prikaz buduće aplikacije preciznije može prezentirati klijentu.

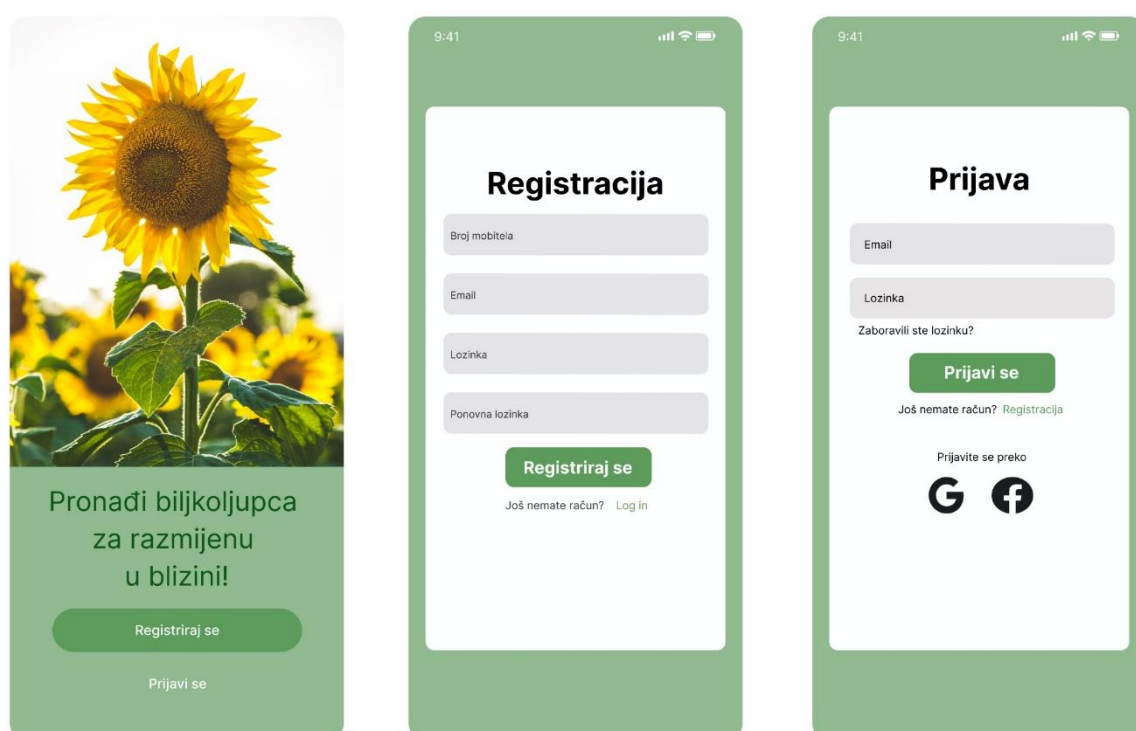


Slika 21: prikaz veza između sučelja u alatu Figma

10. Funkcionalnosti prototipa

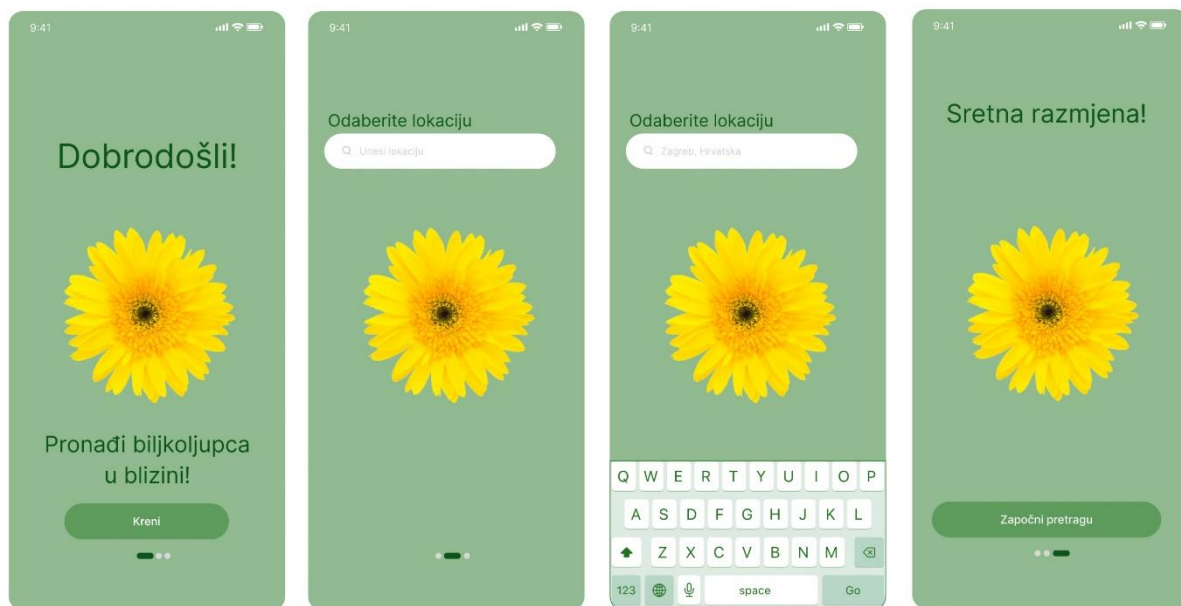
U ovo poglavlju će biti prikazane i opisane funkcionalnosti gotovog prototipa mobilne aplikacije Biljkoljupci. Radi lakšeg razumijevanja rada prototipa aplikacije, kroz devet slika spojenih ekrana biti će podijeljen funkcionalnosti prototipa. Slika 1 prikazuje funkcionalnost registracije novog i prijave postojećeg korisnika.

Slika 21 prikazuje početnu stranicu prototipa na kojoj se korisniku nude opcije registracije i prijave u aplikaciju, dok ekrani dva i tri prikazuju obrasce koje ispunjuje korisnik za registraciju i prijavu.



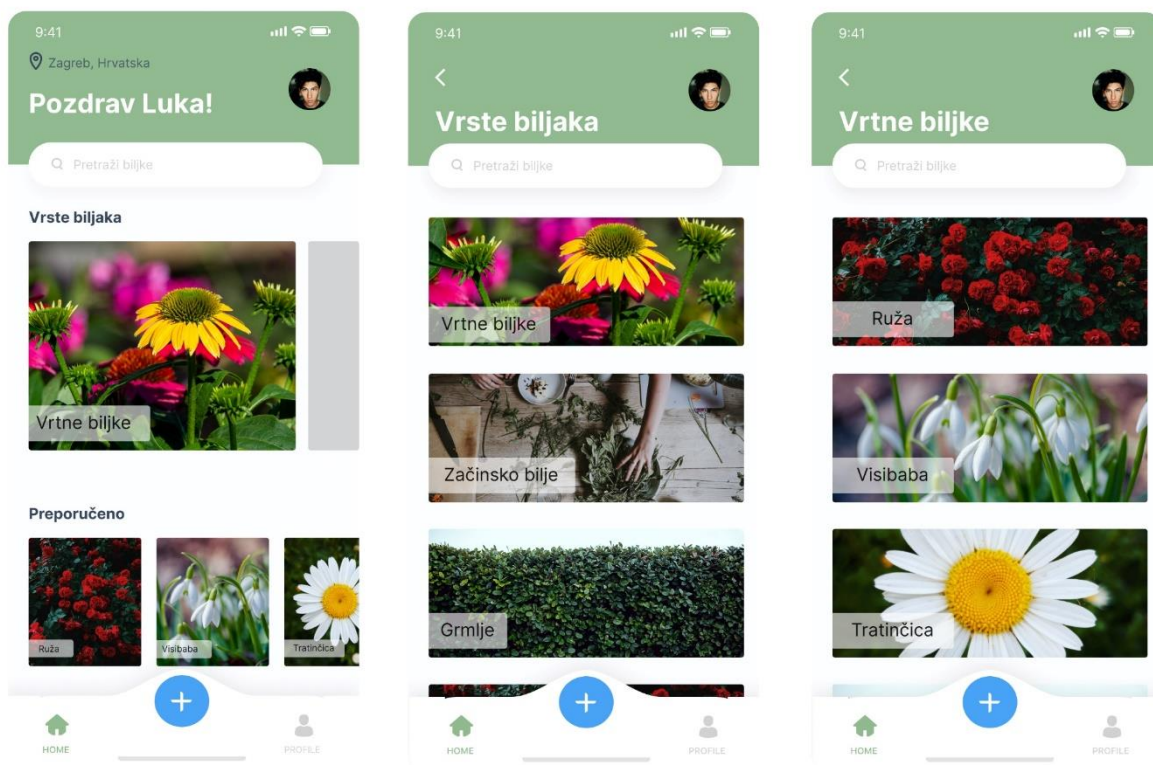
Slika 22: Registracija i prijava korisnik u alatu Figma

Druga funkcionalnost je odabir lokacije korisnika. U ovom koraku korisnik započinje svoje korištenje aplikacije pritiskom na gumb „Kreni“, zatim unosi svoju trenutnu lokaciju kojom određuje prikaz ponude proizvoda u svojoj blizini. Korisnik lokaciju može definirati na tri razine, a to su: ulica, grad i država, ovisno o blizini prikaza željenog sadržaja. Nakon postavljanja lokacije korisnik pritiskom na gumb „Započni pretragu“ započinje pretraživanje željenog sadržaja.



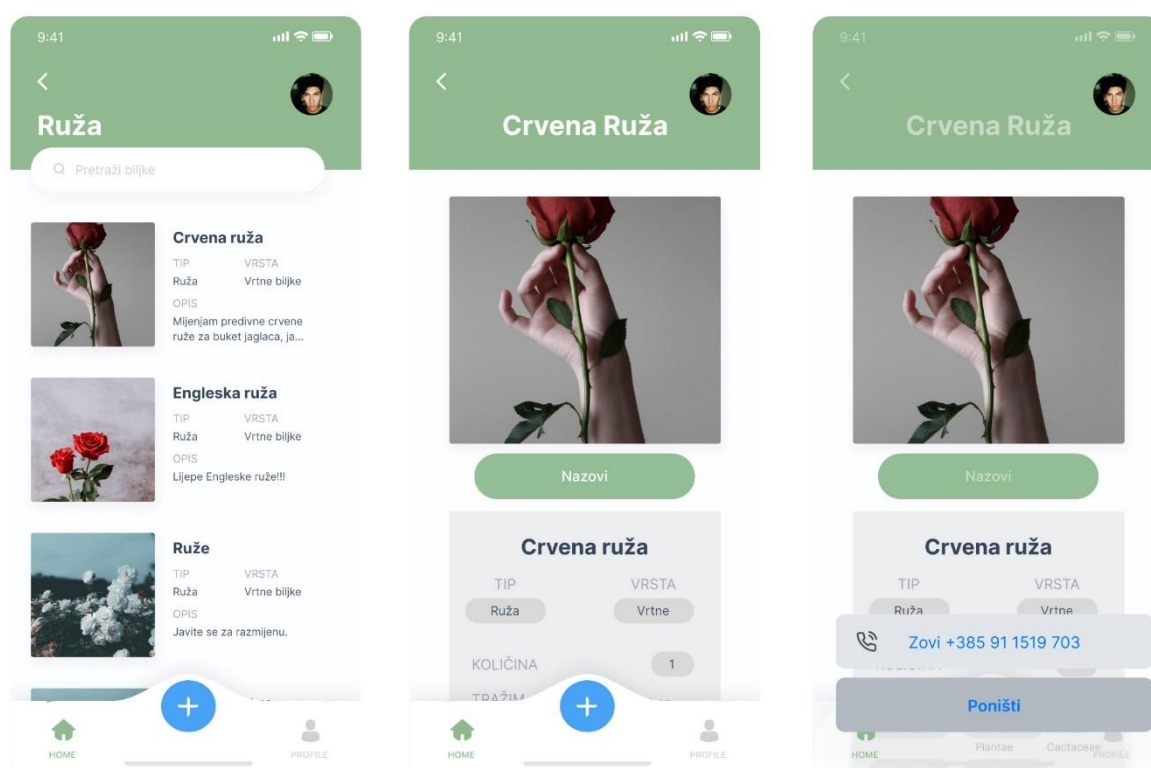
Slika 23: Odabir lokacije korisnika u alatu Figma

Iduća funkcionalnost je pregled naslovne stranice, kategorija i podkategorija unutar kojih pretražuje sortirane ponude proizvoda. Naslovna stranica nudi prikaz kategorija, odnosno vrsti biljaka i preporučeni sadržaj, odnosno podkategorije koje je najčešće pretraživao.



Slika 24: Pregled naslovne stranice, kategorija i podkategorija u alatu Figma

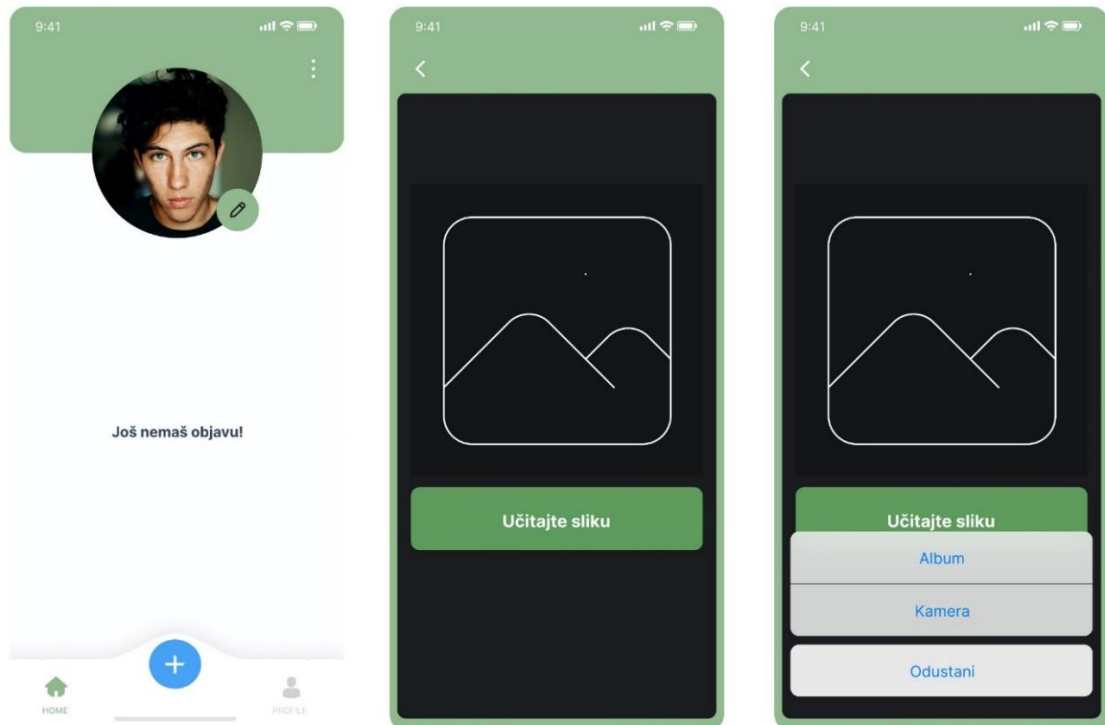
Ova funkcionalnost se nastavlja na prethodnu i nudi korisniku detaljniji pregled ponude. Prikazuje pregled svih oglasa određenog tipa biljaka, pregled pojedinačnog oglasa biljke. Zadnji ekran ove slike prikazuje način stupanja u kontakt dvaju korisnika putem poziva. Na svakom pojedinačno oglasu aplikacija generira broj mobitela oglašivača i prikazuje ga na istom. Ideja rada aplikacije je da se ovim putem sporazumjevaju dva korisnika oko razmjene biljaka, zajedničkog nalaska i ostalih potrebnih informacija oko dogovora. Ovom funkcionalnosti se želi postići što jednostavnija uporaba aplikacije. Razgovorom se postiže najbrži dogovor, te je to razlog izbjegavanja funkcionalnosti za razmjenu poruka unutar sustava.



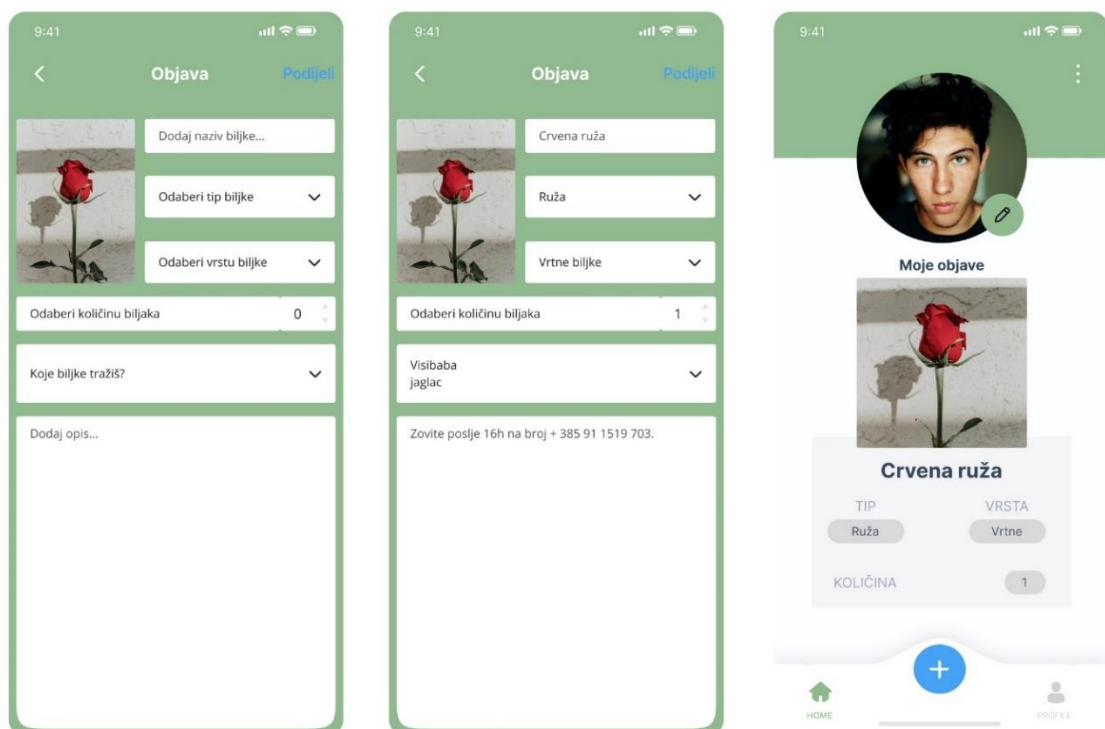
Slika 25: Pregled oglasa i kontaktiranje oglašivača u alatu Figma

Slika 25 i slika 26 prikazuju povezane funkcionalnosti objavljivanja slike ponude i opisivanje iste. Svaki korisnik pritiskom na središnji gumb plave boje sa znakom plus ima mogućnost objave svoje ponude. Prvo, aplikacija korisnika vodi na ekran sa gumbom „Učitajte sliku“ koja nudi opciju odabira slike iz albuma i slikanje foto aparatom. Nakon što je korisnik zadovoljan slikom koju je predodredio za objavu ispunjava obrazac unutar kojeg opisuje objavu. U ovom koraku korisnik definira ime, tip i vrstu biljke. Zatim određuje količinu biljaka koje nudi za razmijenu. Prije samog tekstualno opisa, korisnik uz pomoć padajućeg izbornika odabire tipove biljaka koje potražuje za uzvrat. Tako, svaki korisnik prilikom pregledavanja

ponuda može vidjeti imaju li i potraživač i ponuđač zajedničke osnove za moguću razmijenu biljaka.

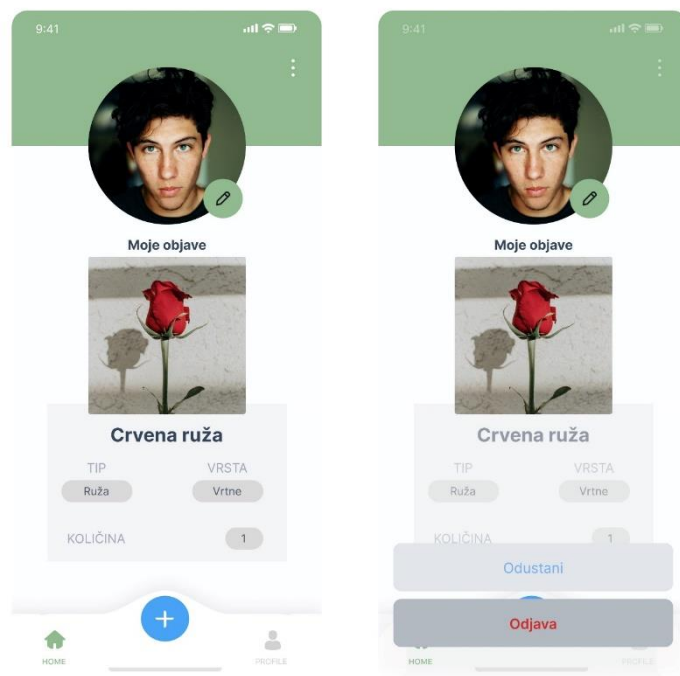


Slika 26: Objavljivanje ponude u alatu Figma

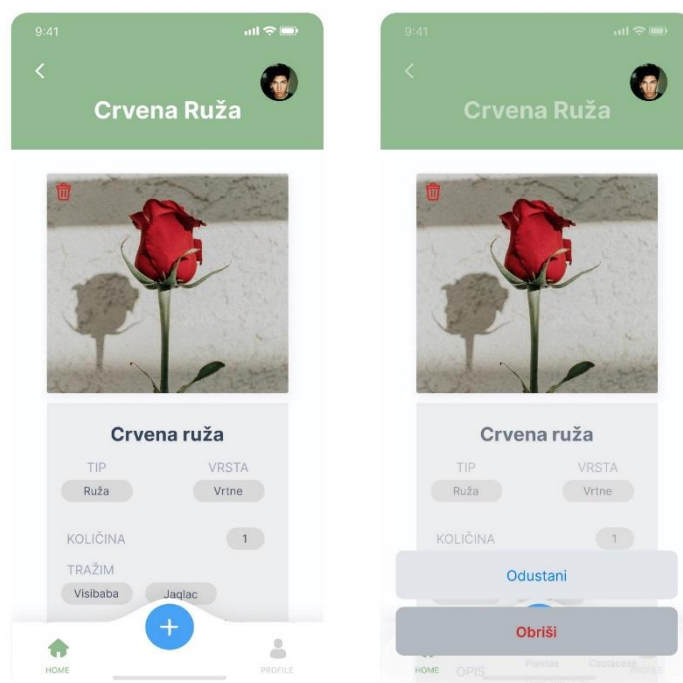


Slika 27: Opisivanje ponude u alatu Figma

Funkcionalnosti na slici 27 i slici 28 prikazuju mogućnost objave sa korisničkog računa, pritiskom na gumb sa tri točkice u desno gornjem kutu ekrana i mogućnost brisanja vlastite objave pritiskom na crven koš u lijevom kutu fotografije objave.

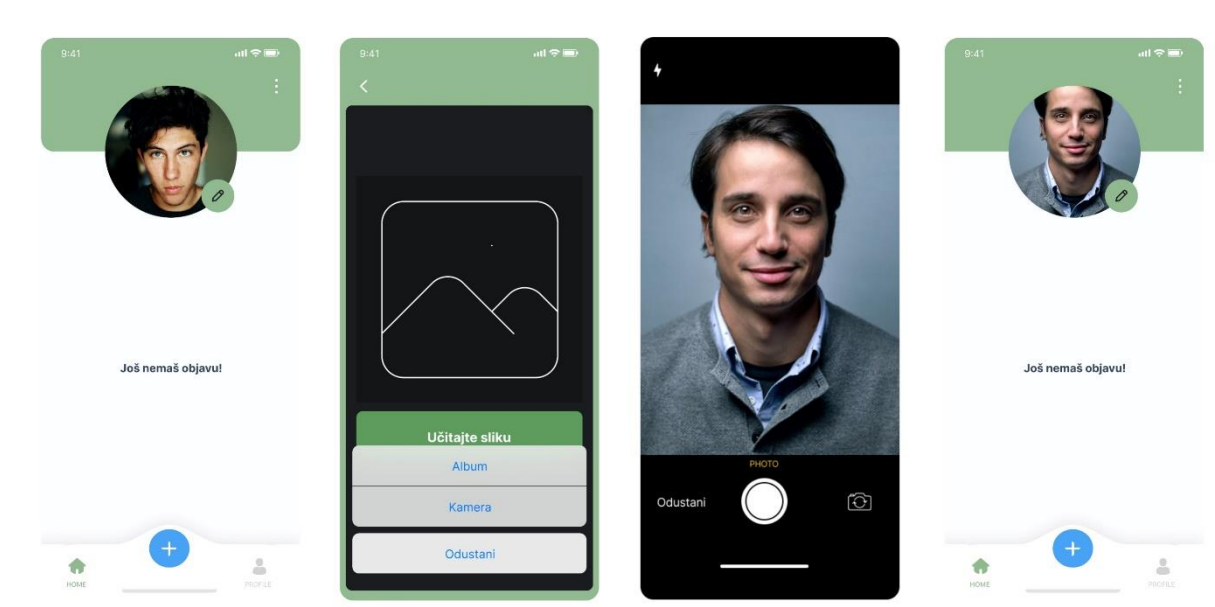


Slika 28: Odjava iz korisničko računa u alatu Figma



Slika 29: Brisanje objave u alatu Figma

Zadnja funkcionalnost na slici 29 prikazuje promjenu slike profila korisnika. Svaki korisnik odlaskom na svoj profil unutar mobilne aplikacije pritiskom na olovku za uređivanje može promijeniti svoju sliku profila.



Slika 30 : Promjena slike profila korisnika u alatu Figma

11. Zaključak

U zaključku ovog istraživanja o razvoju prototipa aplikacije, mogu se sažeti ključne spoznaje i postignuća u teorijskom i praktičnom dijelu rada. Teorijski dio pružio je temeljno razumijevanje koncepta skice i prototipa, modeliranja podataka i procesa te životnog ciklusa razvoja aplikacije. Naglasak je bio stavljen na to kako skica i prototip utječu na razvoj aplikacije,.

U praktičnom dijelu rada, koristeći primjer mobilne aplikacije, uspješno su primjenjeni teorijski koncepti. Aplikacija je detaljno opisana, izrađen je model entiteta i veza kako bis se preciznije definirale funkcionalnosti prototipa aplikacije. Dijagrami slučajeva korištenja omogućili su nam bolje razumijevanje korisničkih interakcija, dok se kroz izradu skice i prototipa postigao vizualni prikaz kako bi mobilna aplikacija trebala funkcionirati i izgledati. Kroz ovaj rad, istaknuta je važnost skice i prototipa u procesu razvoja aplikacija. Ovim koracima omogućuje se rano testiranje ideja i dizajna, čime se smanjuju troškovi i rizik od kasnijih problema, te samim time poboljšava kvaliteta krajnjeg proizvoda. Razvoj mobilne aplikacije služio nam je kao konkretan primjer kako se teorijski koncepti primjenjuju u stvarnom svijetu.

Kroz daljnje istraživanje i razvoj, moguće je dodatno poboljšati proces razvoja prototipa aplikacije, uzimajući u obzir nove tehnologije i metode. U konačnici, razvoj aplikacija ostaje dinamično područje koje zahtijeva kontinuirano učenje i prilagodbu kako bi se isporučile inovativne i korisnički usmjerene aplikacije.

12. Literatura

- [1] V. Strahonja, "Logički model podataka" nastavni materijali na predmetu Projektiranje informacijskih sustava [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2022.
- [2] V. Strahonja, " Modeli procesa - Funkcionalno raščlanjivanje i dijagrami toka podataka" nastavni materijali na predmetu Projektiranje informacijskih sustava [Moodle], Sveučilište u Zagrebu, Fakultet organizacije i informatike, Varaždin, 2022.
- [3] S. Davis, Software Design and Development: the HSC course, Drugo izd., Janine Fendall, Ed.: Parramatta Education Center, 2012.
- [4] T. Lazarova, Low Fidelity Wireframes vs High Fidelity Wireframes, 2018., [Na internetu]. Dostupno: <https://mentormate.com/blog/low-fidelity-wireframes-vs-high-fidelity-wireframes/> [pristupano: 22.08.2023.]
- [5] A. Wulf, Wireframes: A Great Way to Start Development Projects, 2012., [Na internetu]. Dostupno: <https://www.infoq.com/articles/wireframes-start-development-projects/> [pristupano: 22.08.2023.]
- [6] Experienceux, What is wireframing?, bez dat., [Na internetu]. Dostupno: <https://www.experienceux.co.uk/faqs/what-is-wireframing/> [pristupano: 23.08.2023.]
- [7] Excalidraw Team, Introducing Excalidraw+, 2021., [Na internetu]. Dostupno: [Introducing Excalidraw+ | Excalidraw Blog](https://excalidraw.com/blog/introducing-excalidraw-plus/) [pristupano: 23.08.2023.]
- [8] P. Lopusushinsky, Prototyping vs Wireframing: What's The Difference?, bez dat., [Na internetu]. Dostupno: <https://theproductmanager.com/topics/prototyping-vs-wireframing/> [pristupano: 23.08.2023.]
- [9] Figma Learn, What is Figma?, bez dat., [Na internetu]. Dostupno: [What is Figma? – Figma Learn - Help Center](https://www.figma.com/learn/what-is-figma/) [pristupano: 23.08.2023.]
- [10] R. Thampy, 7 Stages Of SDLC: How To Keep Development Teams Running, 2019., [Na internetu]. Dostupno: <https://www.betsol.com/blog/7-stages-of-sdlc-how-to-keep-development-teams-running/> [pristupano: 24.08.2023.]
- [11] M. Martin Software Development Life Cycle (SDLC) Phases & Models, 12.08.2023., [Na internetu]. Dostupno: <https://www.guru99.com/software-development-life-cycle-tutorial.html> [pristupano: 24.08.2023.]
- [12] K. Gordon, How to Draw a Wireframe (Even if You Can't Draw), 2021., [Na internetu]. Dostupno: <https://www.nngroup.com/articles/draw-wireframe-even-if-you-cant-draw/> [pristupano: 24.08.2023.]
- [13] A. Shepherd, Top 5 Low Fidelity Prototype Tools, 23.03.2023., [Na internetu]. Dostupno:

<https://mockitt.wondershare.com/prototyping/low-fidelity-prototype-tools.html>

[pristupano: 25.08.2023.]

[14] P. Martinez, What is a Paper Prototype? In-Depth Guide, 23.03.2023., [Na internetu].

Dostupno: <https://mockitt.wondershare.com/prototyping/paper-prototype.html>

[pristupano: 25.08.2023.]

[15] V. Brathwaite, Wireframe Design 101: 6 Steps to Create a Wireframe, 2020., [Na internetu]. Dostupno: <https://xd.adobe.com/ideas/process/wireframing/wireframe-design-101>

[pristupano: 25.08.2023.]

Korišteni alati:

[1] RealtimeBoard, Inc. 2011., Miro, [alat za izradu skica].

Dostupno: <https://miro.com/templates/low-fidelity-wireframes/>

[2] Balsamiq Studios, LLC 2018., Balsamiq Wireframes, [alat za izradu skica].

Dostupno: <https://balsamiq.com/wireframes/>

[3] C. Chedeau 2020., Excalidraw, [alat za izradu skica]. Dostupno: <https://excalidraw.com/>

[4] Figma, Inc. 2016., Figma, [alat za izradu prototipa].

Dostupno: [Figma: The Collaborative Interface Design Tool](https://www.figma.com/)

[5] Lucid Software Inc. 2008., Lucidchart, [alat za izradu dijagrama].

Dostupno: <https://lucid.app/>

[6] dbdiagram.io 2018., dbdiagram.io, [alat za izradu dijagrama].

Dostupno: <https://dbdiagram.io/>

[7] JGraph Ltd, Diagrams.net, 2010., [alat za izradu dijagrama].

Dostupno: <https://app.diagrams.net/>