Android aplikacija za pomoć pri obavljanju inventure

Stupar, Renato

Undergraduate thesis / Završni rad

2023

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:996137

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij računarstva

Android aplikacija za pomoć pri obavljanju inventure

Završni rad

Renato Stupar

Osijek, 2023. godina.

SADRŽAJ

1. UVOD **1**

1.1. Android aplikacija za pomoć pri obavljanju inventure **1**

2. RJEŠENJA SLIČNIH WEB, ANDROID ILI IOS APLIKACIJA **2**

2.1. Scan-IT to Office **2**

2.2. QR Reader & Barcode Scanner **3**

2.3. QR code reader **4**

2.4. QR Code & Barcode Scanner **5**

2.5. ScannerLab - QR Code Generator **6**

3. POSTUPAK IZRADE APLIKACIJE **7**

4. E - INVENTURA – ANDROID APLIKACIJA ZA POMOĆ PRI OBAVLJANJU INVENTURE **9**

4.1. Koncept **9**

4.1.1. Web aplikacija: Analiza postojećeg digitalnog proizvoda 9

4.1.2. Mobilna aplikacija: skiciranje arhitekture informacije 13

4.1.3. Mobilna aplikacija: crtanje Mape protoka informacija korisničkog iskustva 15

4.2. Dizajn **15**

4.2.1. Alati i metode dizajniranja korisničkog sučelja 16

4.2.2. Kreiranje sustava elemenata korisničkog sučelja 18

4.2.3. Dizajn korisničkog sučelja 19

4.3. Pretvaranje slike u kod **25**

4.3.1. XML – programski razvoj izgleda 25

4.3.2. Prikaz koda po izgledu i značajkama korisničkog sučelja 26

5. ZAKLJUČAK **37**

LITERATURA **38**

SAŽETAK **39**

ABSTRACT **40**

ŽIVOTOPIS 42

1. UVOD

S obzirom na mnoge promjene unutar institucije Fakulteta elektrotehnike, računarstva i informacijskih tehnologija u kontekstu unapređenja i proširenja fakulteta na dvije odvojene lokacije, povećanja broja zaposlenih pojavila se potreba za prikupljanjem informacija o inventaru djelatnika.

Izv. prof. dr. sc. Ivica Lukić kao mentor u suradnji s prof. dr. sc. Krešimir Nenadić, postavili su temelje Android aplikaciji, izradom iste web aplikacije i njene baze podataka.

Nastavno tome prof. dr. sc. Krešimir Nenadić i student Renato Stupar su dogovorili temu završnog rada, a ta je upravo ovaj završni rad pod imenom Android aplikacija za pomoć pri obavljanju inventure.

Obje aplikacije, web i android, služe u navedenu svrhu prikupljanja informacija o inventaru djelatnika, a mogućnosti iste su unos sadržaja o pojedinom artiklu nakon prethodnog očitavanja QR kodova. Neki atributi su predefinirani ranijim unosom administratora, a ostali mogu biti uneseni i naknadno uređeni od strane inventarnog povjerenika.

Ovaj rad sastoji se od poglavlja navedenih u sadržaju, a ta su opisana u nastavku…

Prvo poglavlje, Postupak izrade aplikacije koji sadrži promišljanje i pristup načinu izrade aplikacije. Primjere sličnih aplikacija. Drugo i ujedno najopširnije poglavlje pod nazivom E - Inventura – Android aplikacija za pomoć pri obavljanju inventure - ključno poglavlje gdje je opisan sav proces vezan uz razvoj aplikacije kroz podnaslove: koncept, dizajn i pretvaranje slike u kod. Posljednje poglavlje, zaključak kao ukupan osvrt na proces razvoja digitalnih proizvoda.

1.1. Android aplikacija za pomoć pri obavljanju inventure

Kratko opisati postupak obavljanja inventure. Osmisliti funkcionalnosti potrebne za provođenje inventure upotrebom Android aplikacije. Izraditi Android aplikaciju i istu povezati s postojećom bazom podataka pomoću izrađenog API-ja. Unutar Android aplikacije omogućiti skeniranje linijskog bar koda ili QR koda pojedine stavke.

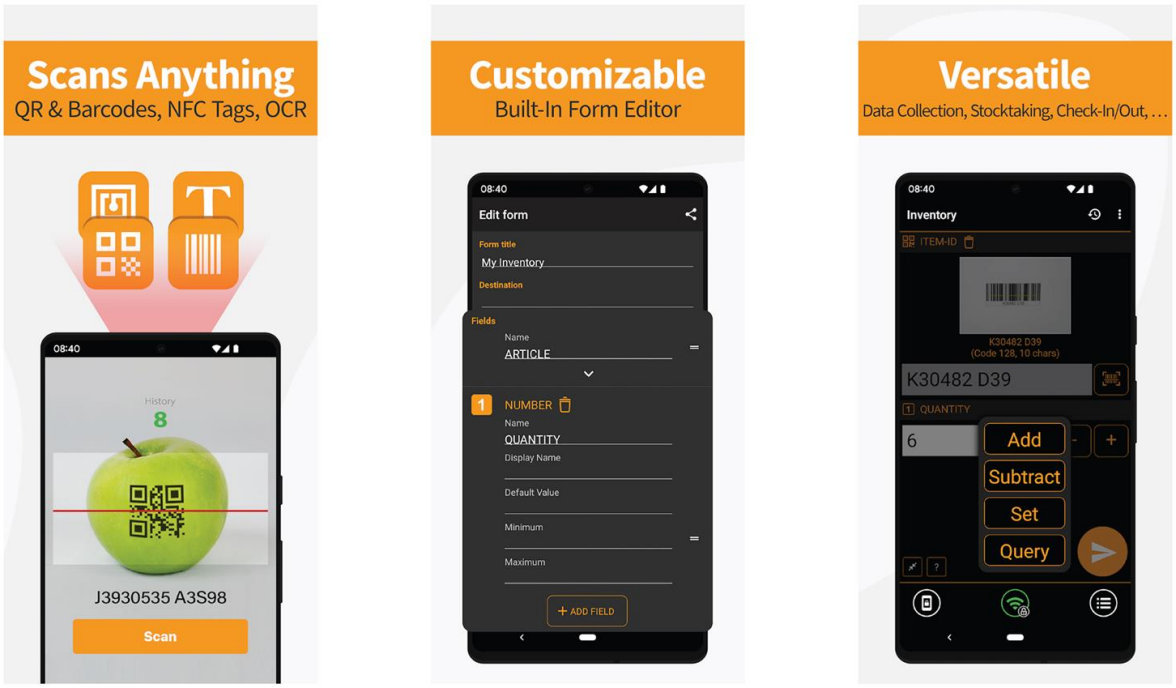
2. RJEŠENJA SLIČNIH WEB, ANDROID ILI IOS APLIKACIJA

U ovom poglavlju su obrađeni slični primjeri aplikacija, a neka od tih rješenja su navedena uz objašnjenja u nastavku u nastavku…

2.1. Scan-IT to Office

Scan-IT to Office je sofisticirana Android aplikacija za mobilno skeniranje QR kodova i barkodova za prikupljanje podataka na temelju obrazaca. Daljinski snimljeni podaci prenose se na računalo ili aplikacije u oblaku u stvarnom vremenu. Aplikacija skenira ili prikuplja linijske kodove, QR kodove, NFC oznake, geolokacije, fotografije, tekstove, korisničke unose, skice i tako dalje. Korisnički definirani obrasci za unos omogućuju prikupljanje više podatkovnih polja odjednom. Izgled sučelja prikazan je na slici 2.1.

Sustav: Android

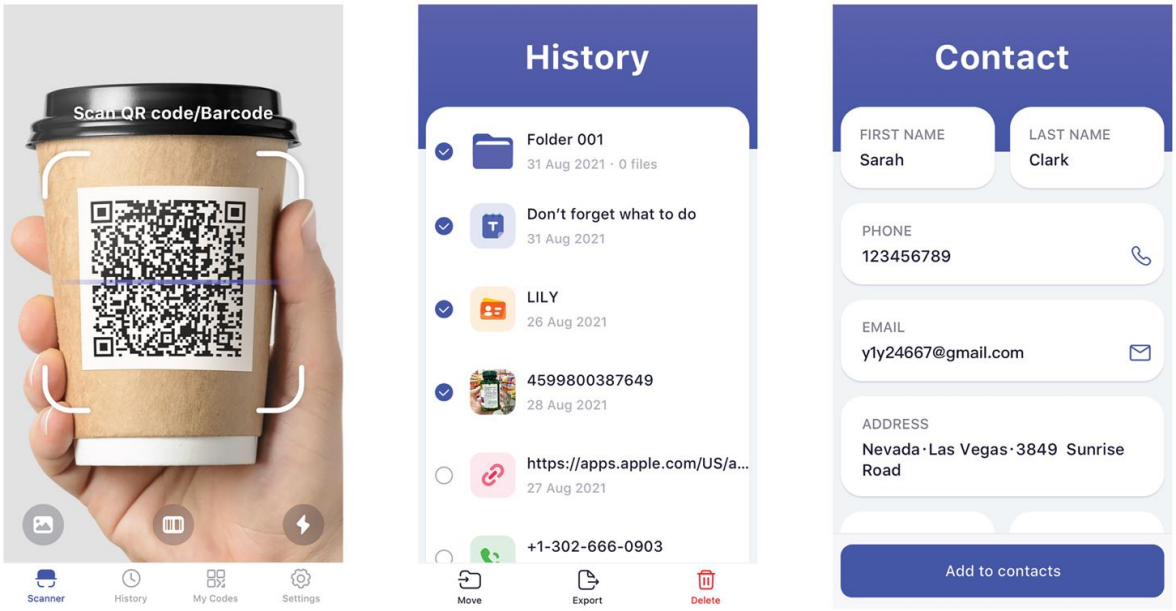


Slika 2.1 Scan-IT to Office: Izgled korisničkog sučelja aplikacije
Izvor: https://play.google.com/store/apps/details?id=com.tecit.android.bluescanner.office

2.2. QR Reader & Barcode Scanner

QR Reader & Barcode Scanner je skener linijskih kodova i QR kodova, a može čitati i dekodirati sve vrste QR kodova i linijskih kodova, uključujući proizvod, URL, Wi-Fi, Google Auth, kalendar, VCard, kontakt, tekst, e-poštu, geolokaciju itd. Sve funkcionalnosti su smještene u podnožje ekrana - donji izbornik (Bottom menu). Izbornih je fiksiran u kontekstu listanja sadržaja, ali je dinamičan i izmjenjiv prema pojedinim zahtjevima i značajkama koje se pokreću pa se prema istima taj izbornik mijenja što je vidljivo na slikama. Izgled sučelja prikazan je na slici 2.2.

Sustav: Android

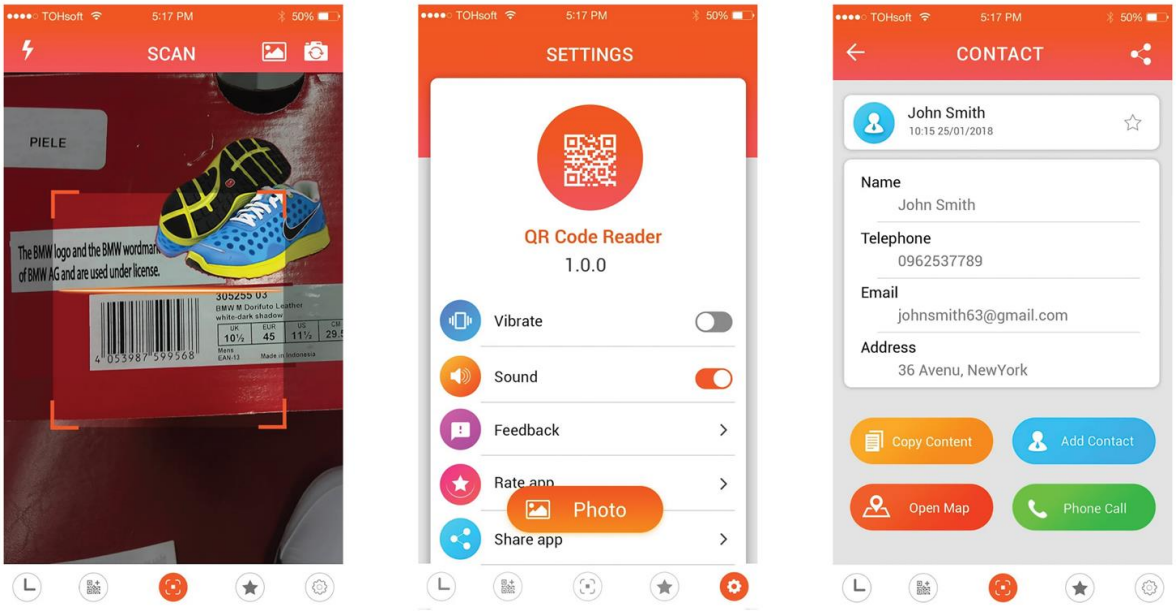


Slika 2.2 QR Reader & Barcode Scanner: Izgled korisničkog sučelja aplikacije
Izvor: https://play.google.com/store/apps/details?id=qrcode.barcode.qr.scanner.reader.scan.free

2.3. QR code reader

Čitač QR koda s glavnim značajkama: čitanje QR koda, skeniranje linijskog koda i stvaranje QR koda uključujući tekst, URL, ISBN, telefonski broj, sms, kontakt, kalendar, e-poštu, lokaciju. Glavne funkcionalnosti su smještene u podnožje ekrana - u donji izbornih (Bottom menu), dok su ostale, sporedne funkcionalnosti smještene u gornji izbornih (Top menu). Izgled sučelja prikazan je na slici 2.3.

Sustav: Android

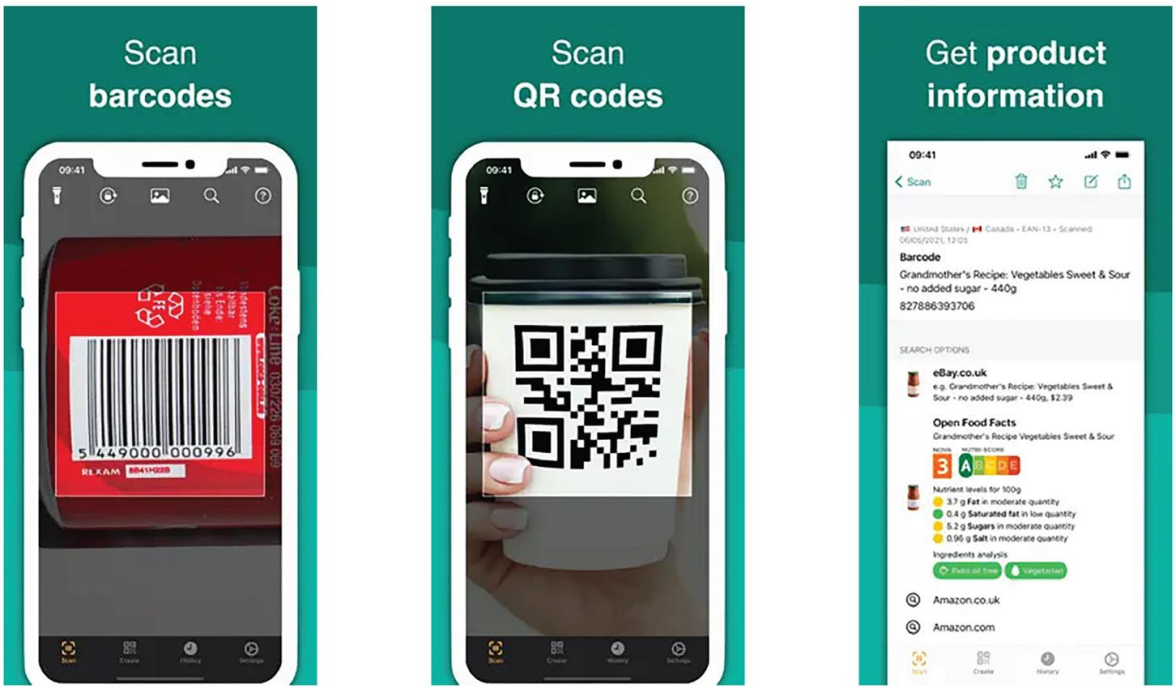


Slika 2.3 QR code reader: Izgled korisničkog sučelja aplikacije
Izvor: https://play.google.com/store/apps/details?id=com.apple.qrcode.reader

2.4. QR Code & Barcode Scanner

Aplikacija za skeniranje QR koda ili linijskog koda. Kreira i dijeli QR kodove za otvaranje URL-ova, povezivanje s WiFi hotspot, dodavanje događaja, čitanje VCards i tako dalje. Značajkama je moguće upravljati složenim inventarom s načinom poslovnog skenera i više. Izgled sučelja prikazan je na slici 2.4.

Sustav: iOS

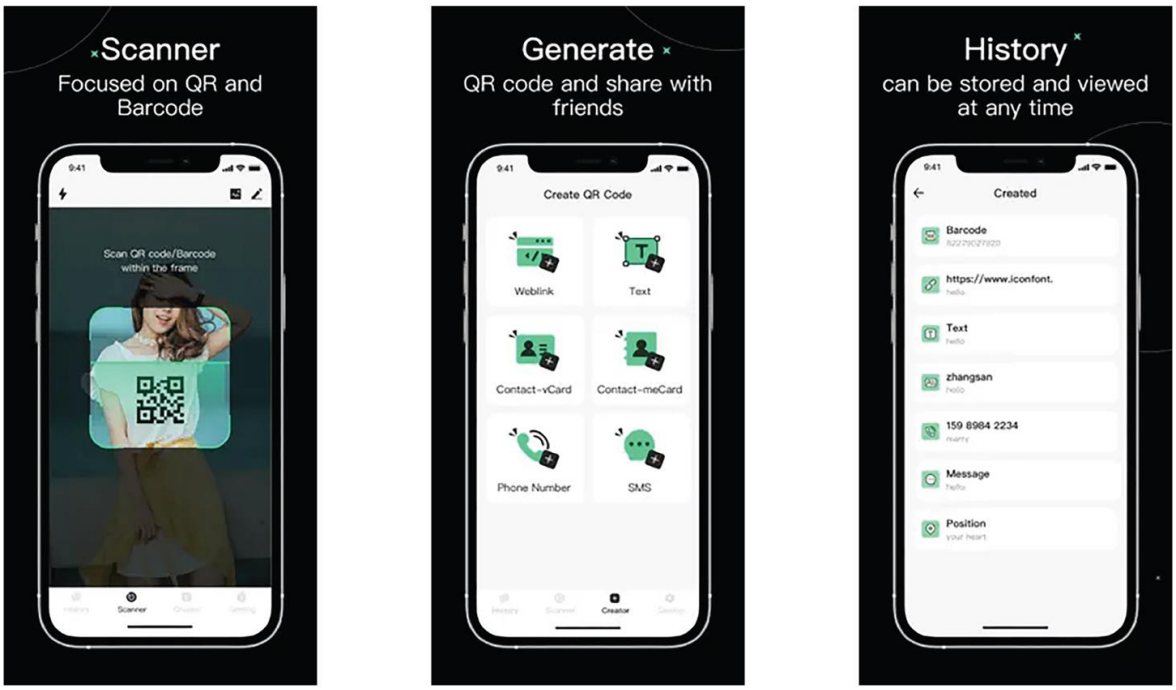


Slika 2.4 QR Code & Barcode Scanner: Izgled korisničkog sučelja aplikacije
Izvor: https://apps.apple.com/hr/app/qr-code-barcode-scanner/id1048473097?platform=iphone

2.5. ScannerLab - QR Code Generator

ScannerLab je aplikacija za iOS sustave kojom je moguće skenirati i generirati QR kodove i linijske kodove. QR kodovi mogu biti dijeljeni sa drugima. Sve glavne funkcionalnosti aplikacije su smještene u podnožje ekrana, u kojem se nalazi fiksirani donji izbornih (Bottom menu). Izgled sučelja prikazan je na slici 2.5.

Sustav: iOS



Slika 2.5 ScannerLab - QR Code Generator: Izgled korisničkog sučelja aplikacije
Izvor: https://apps.apple.com/hr/app/scannerlab-qr-code-generator/id1612611744

3. POSTUPAK IZRADE APLIKACIJE

Budući da se svijest o dizajnu usmjerenom na korisnika sve više naglašava, pristup rješavanja problematike razvoja digitalnih proizvoda je usmjeren na korisnika kroz principe UX dizajna (engl. User eXperience design) - dizajn korisničkog iskustva. UX dizajn u stopu prati i takozvani UI dizajn (engl. User Interface design) - dizajn korisničkog sučelja, a oba pristupa su primijenjena u procesu razvoja ove aplikacije.

Jedna od najvažnijih stvari prilikom izgradnje nekog digitalnog proizvoda su jasne upute i dobro definirani ciljevi i potrebe korisnika koji će koristiti ovaj digitalni proizvod.

"UX proces" ili "popis zadaća prilikom izgradnje korisničkog iskustva i korisničkog sučelja" odgovara na neka od sljedećih pitanja:

- "Koja je svrha aplikacije koja se izrađuje?"
- "Koji su ciljani korisnici aplikacije?"
- "Koji su mogući rizici do kojih se može doći pri izradi aplikacije?"
- "Koji su koraci izrade koji se mogu primijeniti u aplikaciji?"
- "Koji su alati potrebni za realizaciju?"
- "Koja su ograničenja u izradi aplikacije?".

Odgovori na ta pitanja definiraju neke od procesa podijeljenih po fazama koji se koriste prilikom izgradnje nekog digitalnog proizvoda kao što je ova Android aplikacija. Ti procesi, premda nisu utemeljeni kao konačan popis broja zadaća koje dizajner treba izvršiti u svrhu kreiranja izvrsnog digitalnog proizvoda, već ovise o zahtjevima proizvoda i njegovoj kompleksnosti, su navedeni u nastavku:

- Definicija opsega projekta
- Razumijevanje problema projekta
- Istraživanje korisničkog iskustva
- Koncept – skiciranje i izrada maketa i prototipova niske vjernosti
- Izrada makete i prototipova visokih vrijednosti
- Testiranje upotrebljivosti
- Kreiranje sustava elemenata korisničkog sučelja i dizajn korisničkog sučelja
- Programski razvoj digitalnog proizvoda
- Testiranje digitalnog proizvoda: revizija i osiguravanje kvalitete

Kao što je ranije navedeno ova Android aplikacija naslanja se na već postojeću web aplikaciju pa je većina zahtjeva ranije definirana i ispunjena, a ti su: broj korisnika aplikacije, značajke, kategorizacija, pretraga i slično., te se zbog tih definiranih stavki preskaču neki od koraka UX procesa (procesa prilikom izgradnje korisničkog iskustva).

Dio UX procesa koji je korišten prilikom izrade ovog digitalnog proizvoda su faze navedene u nastavku:

- Koncept – Web aplikacija: analiza postojećeg digitalnog proizvoda; Mobilna aplikacija: skiciranje "Arhitekture informacije" i crtanje "Mape protoka informacija korisničkog iskustva"
- Kreiranje sustava elemenata korisničkog sučelja i dizajn korisničkog sučelja
- Programski razvoj digitalnog proizvoda
- Testiranje digitalnog proizvoda: revizija i osiguravanje kvalitete.

Svi procesi su obavljeni i izrađeni skicama na papiru, crtanjem korisničkog sučelja u alatu za crtanje korisničkih sučelja - Figma i programskim razvojem u alatu Android Studio. Navedeni alati će biti obrađeni i razjašnjeni kasnije u tekstu.

4. E - INVENTURA – ANDROID APLIKACIJA ZA POMOĆ PRI OBAVLJANJU INVENTURE

U ovom poglavlju opisani su i obrađeni postupci stvaranja konceptualnog modela mobilne aplikacije, dizajn mobilne aplikacije i pretvaranje slike u kod - programiranje.

Nekima od postupaka konceptualnog stvaranja kao što su: analiza postojeće web aplikacije, skiciranje arhitekture informacija, crtanje mape protoka informacija korisničkog iskustva je postavljen dobar temelj za crtanje korisničkog sučelja.

Dizajn je postupak crtanja korisničkog sučelja koji podrazumijeva kreiranje svih potrebnih elemenata koji sučelje korisniku čine jasnim za korištenje.

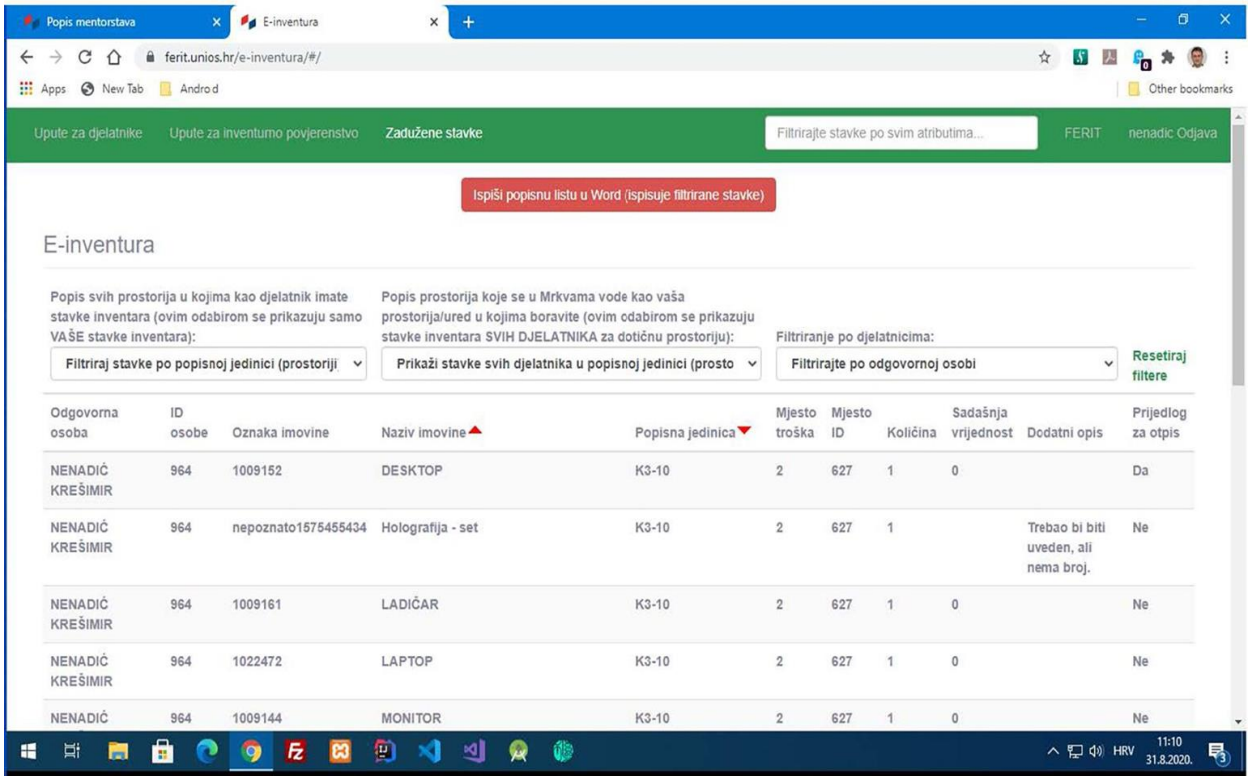
Pretvaranje slike u kod podrazumijeva definiranje oblika dizajniranih elemenata i pridodavanje značajki poput interakcija i ponašanja tim elementima putem teksta koda.

4.1.Koncept

Mobilna aplikacija se oslanja na postojeću web aplikaciju, pa su također i zbog toga preskočeni neki već navedeni, a dobro poznati uvriježeni koraci postupka izgradnje korisničkog iskustva i korisničkog sučelja. Revizijom postojeće web aplikacije prikupljene su informacije o značajkama i ponašanju iste, a te navodim u nastavku…

4.1.1. Web aplikacija: Analiza postojećeg digitalnog proizvoda

Osnovna funkcija web aplikacije je sakupljanje informacija o inventaru djelatnika fakulteta, kraće rečeno “inventura inventara”. Cilj je da jedna osoba (djelatnik fakulteta - inventarni povjerenik) koja ima dozvolu fakulteta, dopuštenje administratora i korisničke podatke za ulazak u aplikaciju, obilazeći zgrade fakulteta (sobe i hodnike) popisuje sav inventar fakulteta koji koriste djelatnici. Početni zaslon web aplikacije je prikazan na slici 4.1.



Slika 4.1 Korisničko sučelje postojeće Web aplikacije - početni zaslon: prikaz više subjekata (snimka zaslona)

Web aplikacija je podijeljena na neke osnovne i dodatne atribute pomoću kojih se bilježe i definiraju subjekti, njihova imenovanja i opisi; i značajke za pretragu, kategorizacija i filtracija podataka. U nastavku su navedeni osnovni i dodatni atributi i značajke.

- Osnovni atributi:
  - Odgovorna osoba
  - ID osobe
  - Naziv imovine
  - Propisana jedinica

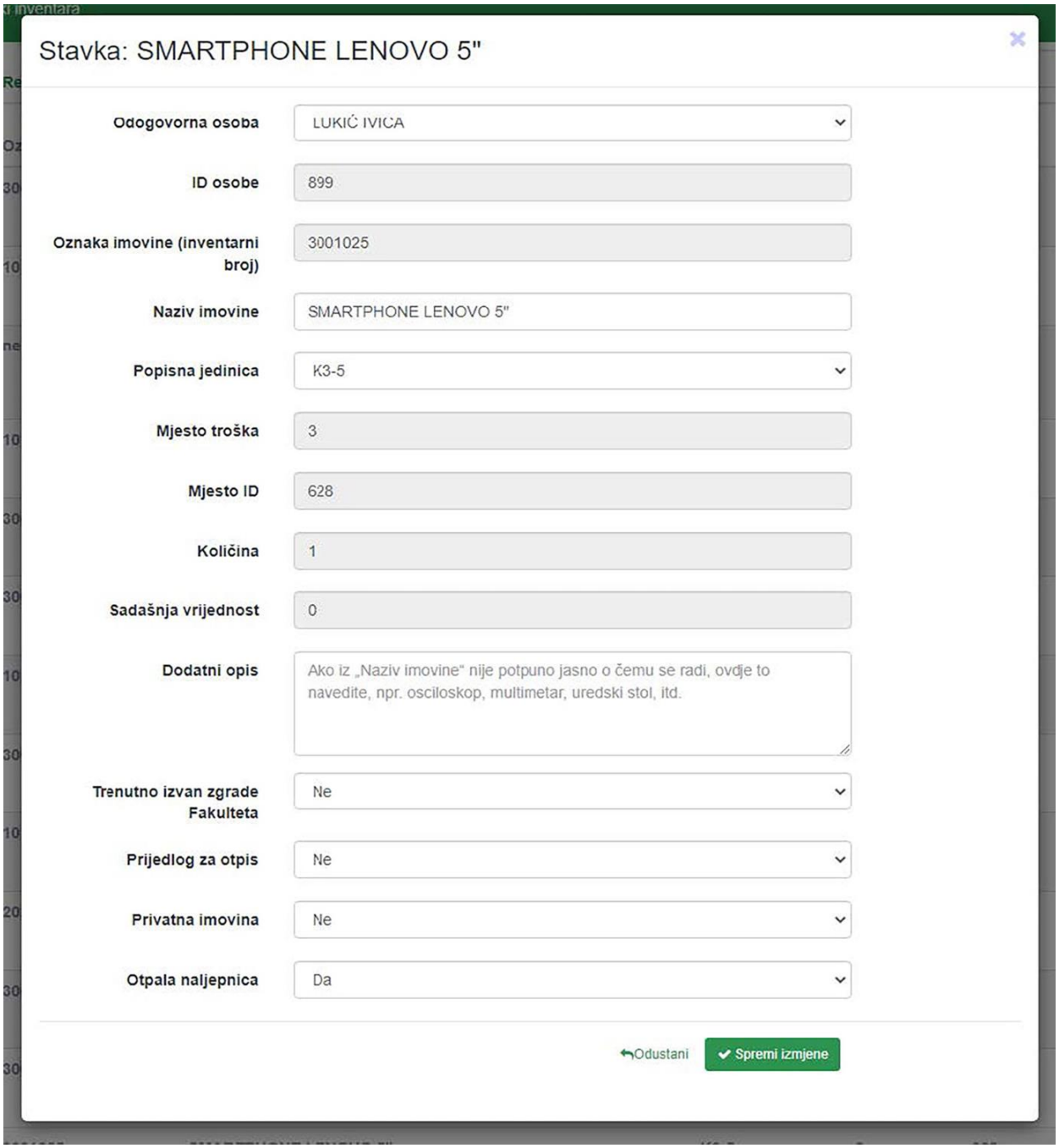
- Dodatni atributi:
  - Oznaka imovine (inventarni broj)
  - Mjesto troška
  - Mjesto ID
  - Količina
  - Sadašnja vrijednost
  - Dodatni opis

    - Trenutno izvan zgrade fakulteta
    - Prijedlog za otpis
    - Privatna imovina
    - Otpala naljepnica

- Osnovne značajke:
    - Pregled subjekata (zadužene stavke)
    - Dodaj subjekt
    - Pretraži subjekte
    - Filtriraj subjekte

- Dodatne značajke:
    - Upute za djelatnike
    - Upute za inventurno povjerenstvo
    - Korištenje aplikacije
    - Ferit (poveznica sa web lokacijom fakulteta)
    - Odjava

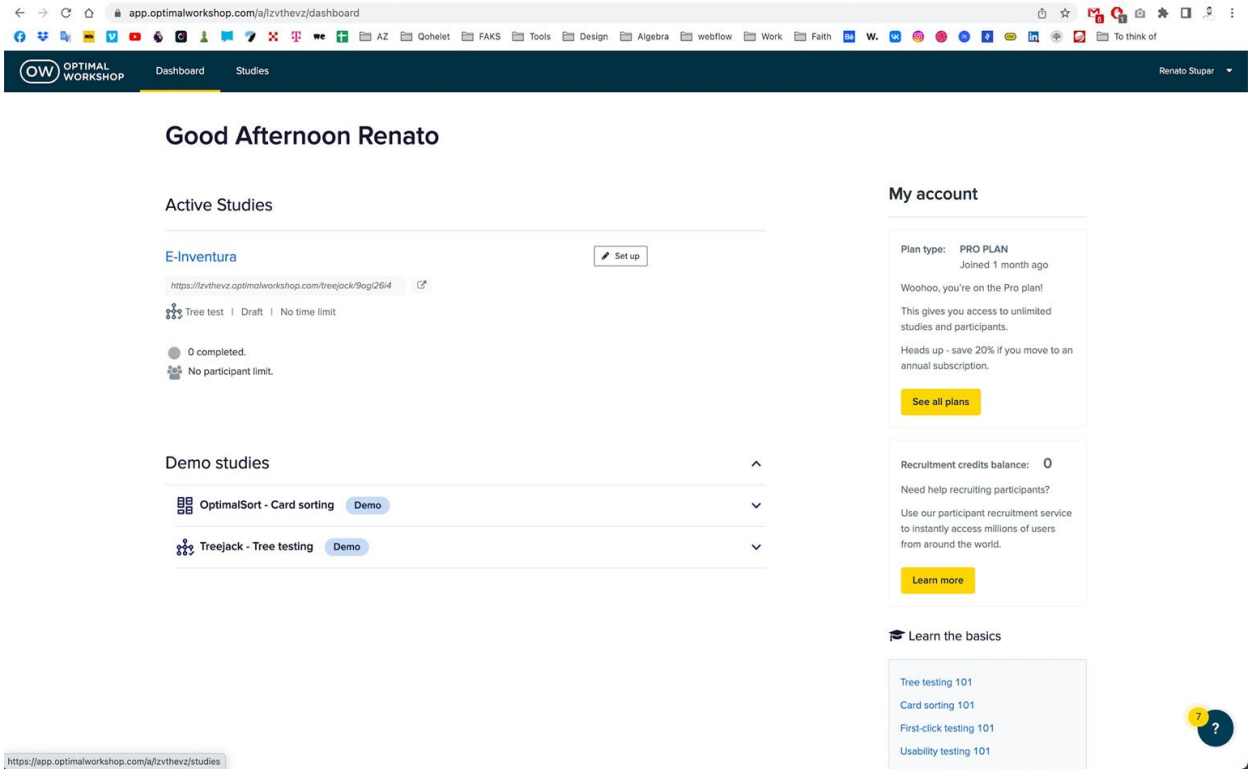
Osnovne i dodatne atribute pojedine stavke mogu se vidjeti na slici 4.2.



Slika 4.2 Pojedine stavke web aplikacije: prikaz atributa jedne stavke (snimka zaslona)

4.1.2. Mobilna aplikacija: skiciranje arhitekture informacije

Alat koji je korišten za izradu arhitekture informacija je web platforma firme Optimal Workshop prikazan na slici 4.3.



Slika 4.3 Optimal Workshop: alat za testiranja i izradu korisničkih iskustava (snimka zaslona)

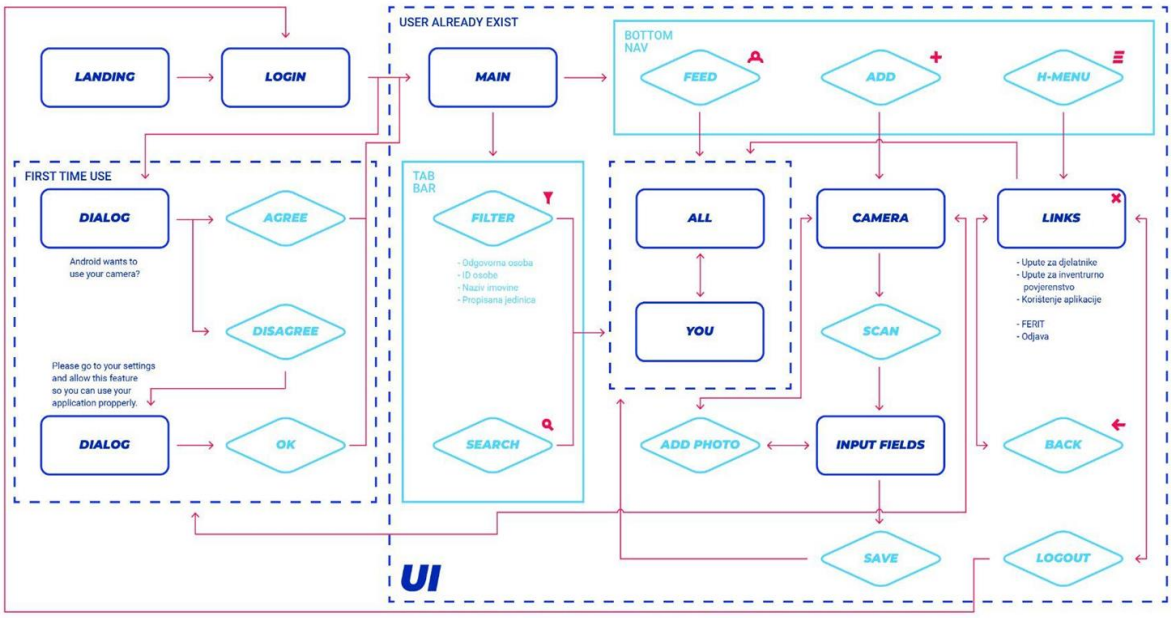
Kako bi analizirani podaci bili dobro smješteni u mobilnu aplikaciju, informacije koje su prikupljene bilo je potrebno razvrstati u "drvo grananja informacija" koje se koristi u procesu ideacije, a naziva se "Arhitektura informacija" prikazan na slici 4.4.

Ovo je jednostavan, ali efikasan, postupak slaganja informacija bez da pozornost oduzimaju crtani grafički elementi korisničkog sučelja, već je fokus samo na informacijama i njihovom povezivanju.

Slika 4.4 Ahitektura informacija (snimka zaslona)

4.1.3. Mobilna aplikacija: crtanje Mape protoka informacija korisničkog iskustva

Prikupljeni podaci organizirani metodom grananja “Arhitektura informacije”, primijenjeni su i prikazani kroz postupak “Mape protoka informacija korisničkog iskustva” (engl. user flow map), principom modeliranje entiteta, veza i atributa; a uobličene geometrijskim likovima poput pravokutnika, romba i linije prikazanu na slici 4.5.



Slika 4.5 Mapa protoka informacija korisničkog iskustva (User flow map)

4.2.Dizajn

Teorijsko tematiziranje procesa dizajna promatra se u članku kao sastavni dio utemeljenja profesije i stoga na razini teorije, ali i historiografije, odnosno u povezivanju povijesnih i teorijskih uvida, osnova za konstituiranje povijesti projektiranja za masovnu proizvodnju (onoga što se danas u hrvatskom jeziku standardno naziva – dizajn). Projektiranje za masovnu proizvodnju, bilo uporabnih predmeta ili simboličnih tvorevina/značenja, ima u hrvatskom kontekstu i svoj teorijski aspekt, o čemu svjedoči velik broj članaka koji su na tu temu objavljivani od 1950. do 1990. članak sagledava dizajn kao dio općega procesa društvene modernizacije i uspostavlja metodičku osnovu na teorijskoj aparaturi iz djela teoretičara dizajna u zemljama razvijenoga kapitalizma, s obzirom na polaznu pretpostavku da je društveno-ekonomski razvoj bio temeljnom tendencijom društvenih zajednica liberalnoga kapitalizma i onih nominalnoga socijalizma. Iznosi se i retrospekcija

najvažnijih teorijskih ideja o dizajnu u Hrvatskoj u navedenom razdoblju, čime se ostvaruju sljedeći rezultati: katalogizacija ideja i popis referentnih članaka kao okvir za daljnje istraživanje. U zaključku članka povlači se distinkcija između teorijske pozicije pojma oblikovanje i pojma dizajn, kao osnova za daljnja istraživanja i teorijska preispitivanja. [1]

Priča o suvremenom dizajnu se u teoriji spominje u knjizi Feđe Vukića: Teorija i povijest dizajna, Kritička antologija, koja navodi tekst John Ruskin-a pod nazivom "Moderna proizvodnja i dizajn", a koji je osvrt na predavanje u Bradfordu u ožujku 1859. godine. U jednom dijelu teksta koji govori o formi ornamenta nalazi se navod koji spominje primjenjivost elemenata i značajki poput kontrasta, niza i simetrije, koji kao takvi čine funkcionalnost tog sustava kojeg zovemo ornament. Samostalno, elementi ornamenta ma koji god oni bili, ukoliko nisu primijenjeni sa ove tri značajke čine zasebne figure gdje god one bile prikazane.

Kasnije kroz vrijeme, od tog događaja do danas dizajn se razvija usporedno sa proizvodima koje formira, bilo da se radi o fizičkom proizvodu poput automobila, perilice za rublje ili sušila za kosu, ili bilo kojem statičkom ili interaktivnom grafičkom prikazu nekih informacija korisnih za čovjeka. Rješenje oblikovanja formi koje se kroz dizajn kreiraju moraju imati primijenjene obje karakteristike, a te su: estetika i funkcionalnost. Sve dok oba kriterija nisu zadovoljena, taj dizajn možemo smatrati djelomično upotrebljivim ili čak i neupotrebljivim.

Kroz industrijsku revoluciju i njenu evoluciju pa do danas susrećemo proizvode koji pretpostavljaju njihovo korištenje, a gdje svakako korisnik tih proizvoda mora učiti metode i načine korištenja istih.

Dublje izraženo teoretiziranje o dizajnu i primjenu istog na proizvode primjećujemo u principima njemačkog dizajnera Dieter Rams-a koji je zaključio kako dobar dizajn mora biti inovativan, činiti proizvod korisnim, biti estetski prihvatljiv, biti razumljiv, biti nenametljiv, biti pošten, biti dugotrajan, biti temeljit do zadnjeg detalja, biti ekološki prihvatljiv i uključivati što manje dizajna.

S obzirom da živimo u doba interaktivnosti danas poznajemo princip kreiranja proizvoda koji gleda iz perspektive korisnika, a taj se zove Dizajn usmjeren na korisnika (engl. Human centered design).

Deset principa za dobar dizajn Dietera Ramsa

Dobar dizajn je:

1. Inovativan
2. Čini proizvod korisnim
3. Estetski
4. Čini proizvod razumljivim
5. Nenametljiv
6. Pošten
7. Dugotrajan
8. Temeljito do zadnjeg detalja
9. Ekološki prihvatljiv
10. Uključuje što je moguće manje dizajna [2]

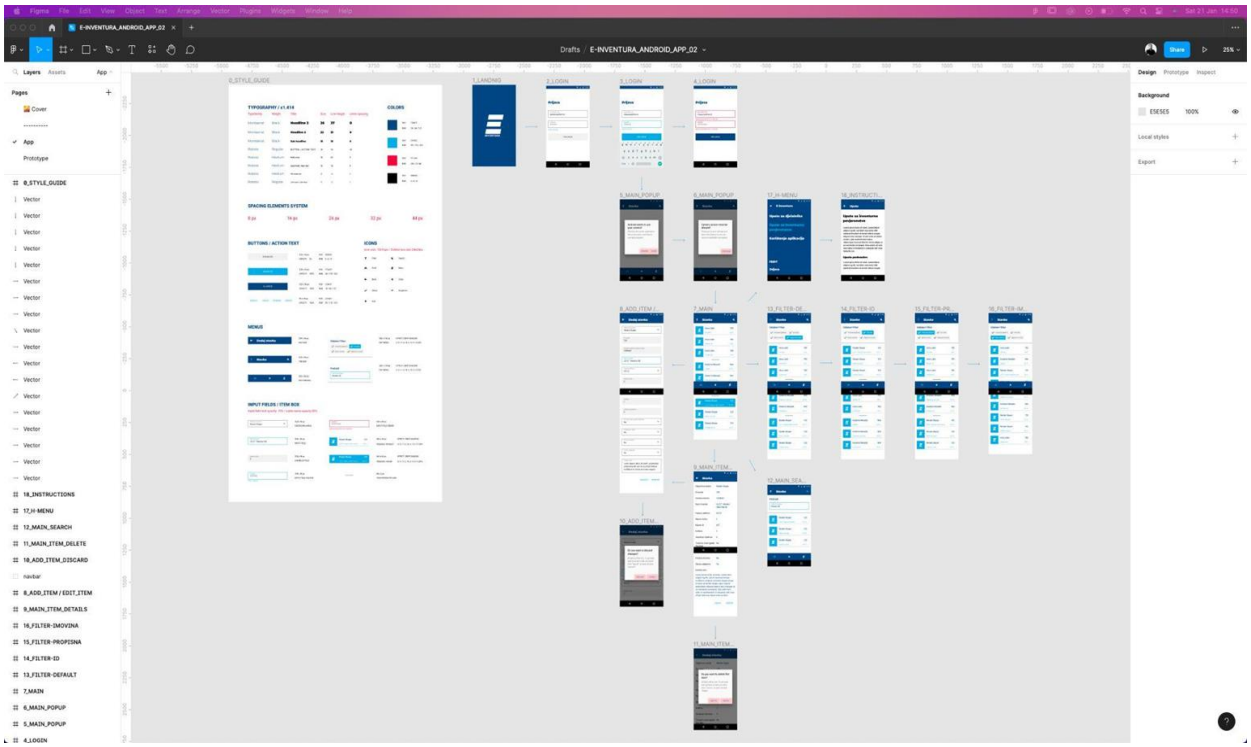
4.2.1. Alati i metode dizajniranja korisničkog sučelja

Dizajn korisničkih sučelja ide korak uz korak s razvitkom računala i sustava korištenih na istima poput Windows, MacOS, Ubuntu, Linux i slično. Prva korisnička sučelja koja su se implementirala u internet i očitavala internetskim pregledima čistim HTML-om (engl. HyperText Markup Language) su se stilizirala kodom. Pojavom izuma CSS-a (engl. Cascading Style Sheets) sadržaj se stilizirao tablično, a nakon toga bilo je potrebno određeno vrijeme kako bi se korisnička sučelja krenula dizajnirati u pravom smislu te riječi.

Jedan od procesa kreiranja korisničkog iskustva koje je usko povezano s korisničkim sučeljem je crtanje samog korisničkog sučelja. Crtanje uključuje kreiranje sadržaja poput: navigacija, linkova, "CTA button" (engl. Call To Action button), mjesto za postavljanje digitalnih slika, određivanje tipografije, boja, piktograma ili ikonica, određivanje horizontalnih razmaka elemenata, vertikalnog ritma (vertikalni razmaci između grafičkih i drugih elemenata) i slično. Sučelja su se razvijala u raznim digitalnim rasterskim i vektorski orjentiranim alatima za crtanje poput Corel Draw ili jednog od Adobe-ovih alata poput "Photoshop", "Illustrator" ili "InDesign". Od toga doba do danas su se razvili neki od digitalnih alata poput "Sketch", "Adobe XD" i "Figma".

Figma je vektorski orijentiran alat za digitalno crtanje korisničkih sučelja, odnosno široko rasprostranjen alat za uređivanje i izradu maketa raznih vrsta digitalnih proizvoda i kreiranje njihove interaktivnosti. Figma je crtačko-kolaboracijski alat zasnovan na ideji o upotrebi na lokalnoj jedinici putem internetskog preglednika ili desktop aplikacije, a kolaborativno oslonjen na dijeljenje sadržaja putem servera. Ovu ideju smislili su Dylan Field i Evan Wallace kao studenti

Sveučilišta Brown u Providenceu 2011. godine. Na ideji su krenuli raditi 2012. godine te je Figma četiri godine kasnije prvi puta objavljena 27. rujna 2016. godine. Prvenstveno je zasnovana na mreži dok su dodatne izvanmrežne značajke omogućili računalni programi na macOS i Windows platformama. Glavna funkcionalnost, uvedena 22. kolovoza 2019. godine, uz izradu dizajna jest mogućnost suradnje više osoba u stvarnom vremenu. Ovom funkcionalnošću je pristup dizajnu omogućen ne samo dizajnerima nego i cijelome timu koji radi na projektu. Korisničko sučelje alata prikazano je na slici 4.6.



Slika 4.6 Figma: prikaz korisničkog sučelja

4.2.2. Kreiranje sustava elemenata korisničkog sučelja

Dizajn i razvoj digitalnih proizvoda u kontekstu moderne i aktualne IT industrije općenito podrazumijeva dvije usko povezane kategorije dizajna, a te su dizajn korisničkog iskustva (eng. user experience design) i dizajn korisničkog sučelja (engl. user interface design).

Dizajn korisničkog iskustva (engl. user experience design) ili skraćeno "UX design" je poznat i kao "Dizajn usmjeren na korisnika" (engl. User-centered design).

Izraz 'dizajn usmjeren na korisnika' nastao je u istraživačkom laboratoriju Donalda Normana na Kalifornijskom sveučilištu San Diego (UCSD) 1980-ih i postao je široko korišten nakon

objavljivanja koautorske knjige pod naslovom: Dizajn sustava usmjeren na korisnika: Nove perspektive o interakciji čovjek-računalo (Norman & Draper, 1986). Norman je dalje nadogradio koncept UCD-a u svojoj temeljnoj knjizi The Psychology of Everyday Things (POET) (Norman, 1988). [3]

Nakon što je raznim metodama istraživanja i testiranja definirano zadovoljavajuće korisničko iskustvo, isto je potrebno prikazati na zaslonu ekrana na kojem se odvija primjena tog istog iskustva. Tu počinje priča o korisničkom sučelju.

4.2.3. Dizajn korisničkog sučelja

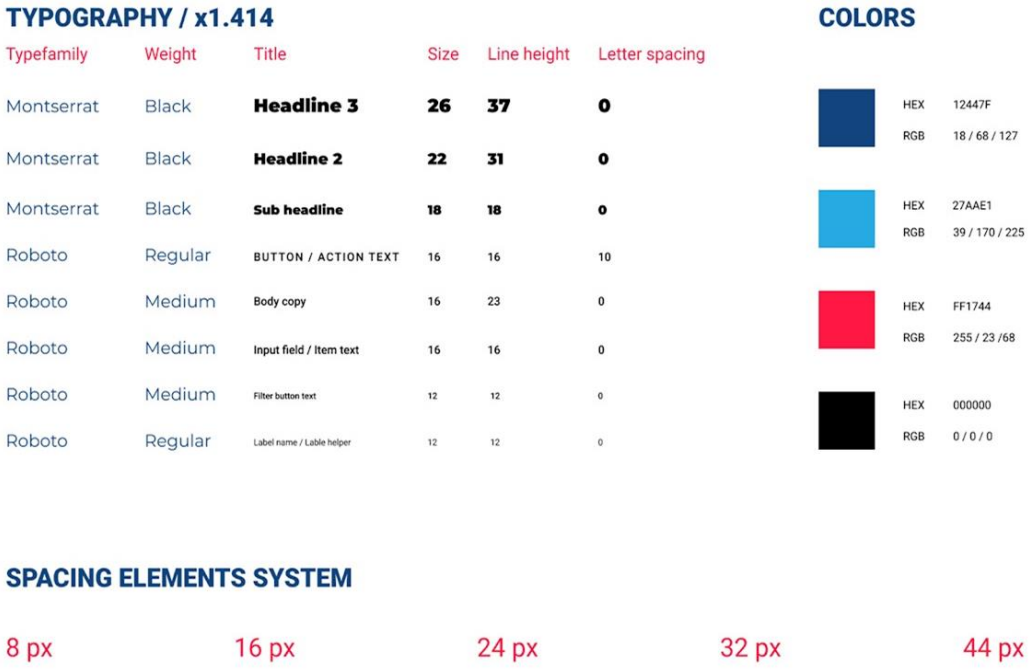
Dizajn korisničkog sučelja (engl. User interface design) ili skraćeno "UI dizajn" je prema definiciji organizacije "Interaction design foundation" sljedeće:

Dizajn korisničkog sučelja (UI) proces je koji dizajneri koriste za izradu sučelja u softveru ili računalnim uređajima, fokusirajući se na izgled ili stil. Dizajneri nastoje stvoriti sučelja koja će korisnicima biti laka za korištenje i koja će biti ugodna. Dizajn korisničkog sučelja odnosi se na grafička korisnička sučelja i druge oblike—npr. sučelja kojima se upravlja glasom. [4]

Nakon što su prikupljene sve potrebne informacije i organizirane alatima i metodama koje su ranije navedene u poglavlju 3.1 Koncept preostala je još jedna faza kreiranja mobilne aplikacije. Aplikaciju je bilo potrebno nacrtati na minimalan format ekrana Android uređaja, a taj je 360 x 640 piksela.

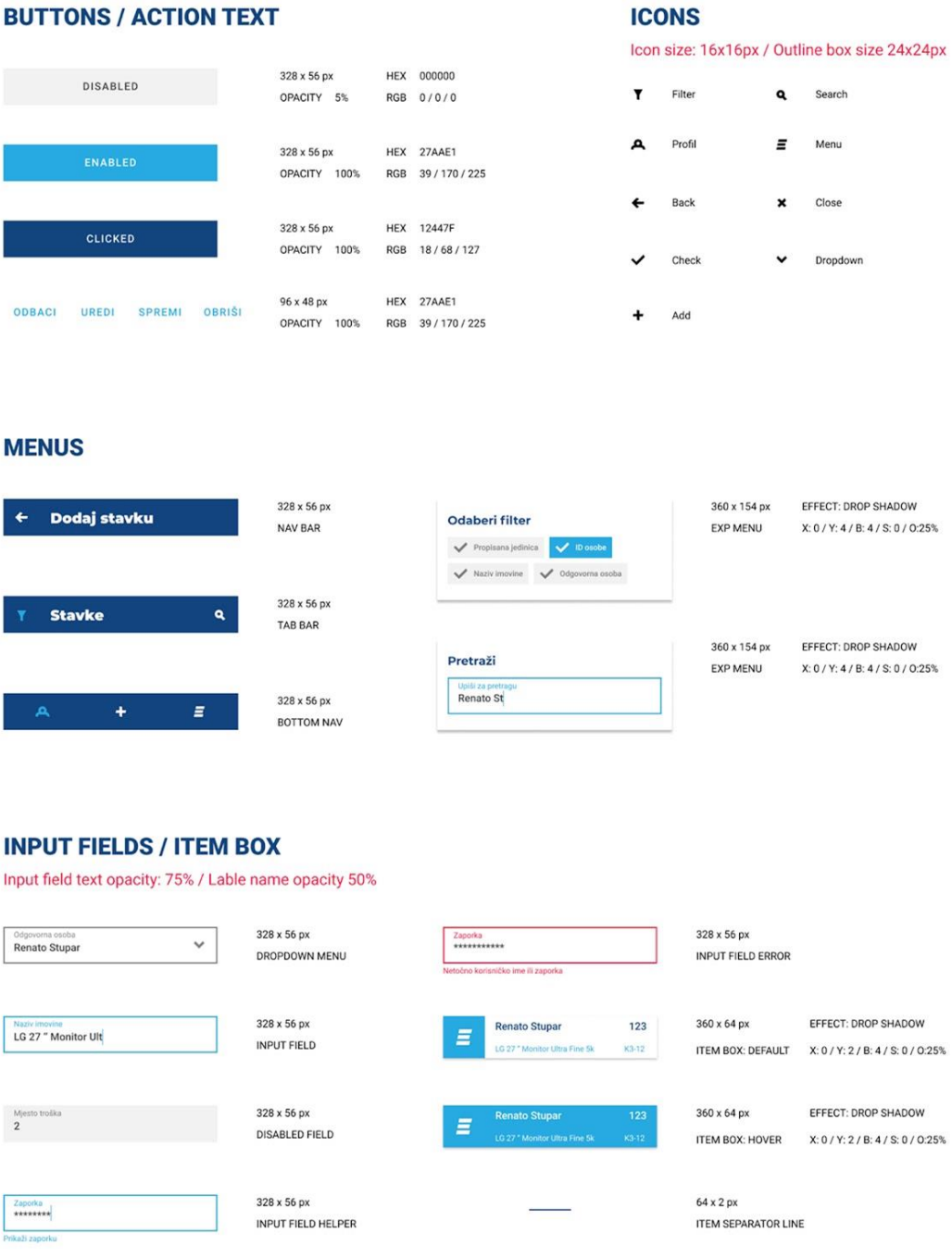
U nastavku se nalaze slike s opisima stilova komponenti i drugih elemenata korištenih za crtanje - (engl. style guide) i digitalne makete svih ekrana aplikacije.

Na slici 4.7 se nalazi prvi dio spomenutog "style guide" na kome su prikazani definirani tekstualni stilovi, stilovi korištenih boja i razmak korišten između slaganja pojedinih elemenata prilikom crtanja mobilne aplikacije.



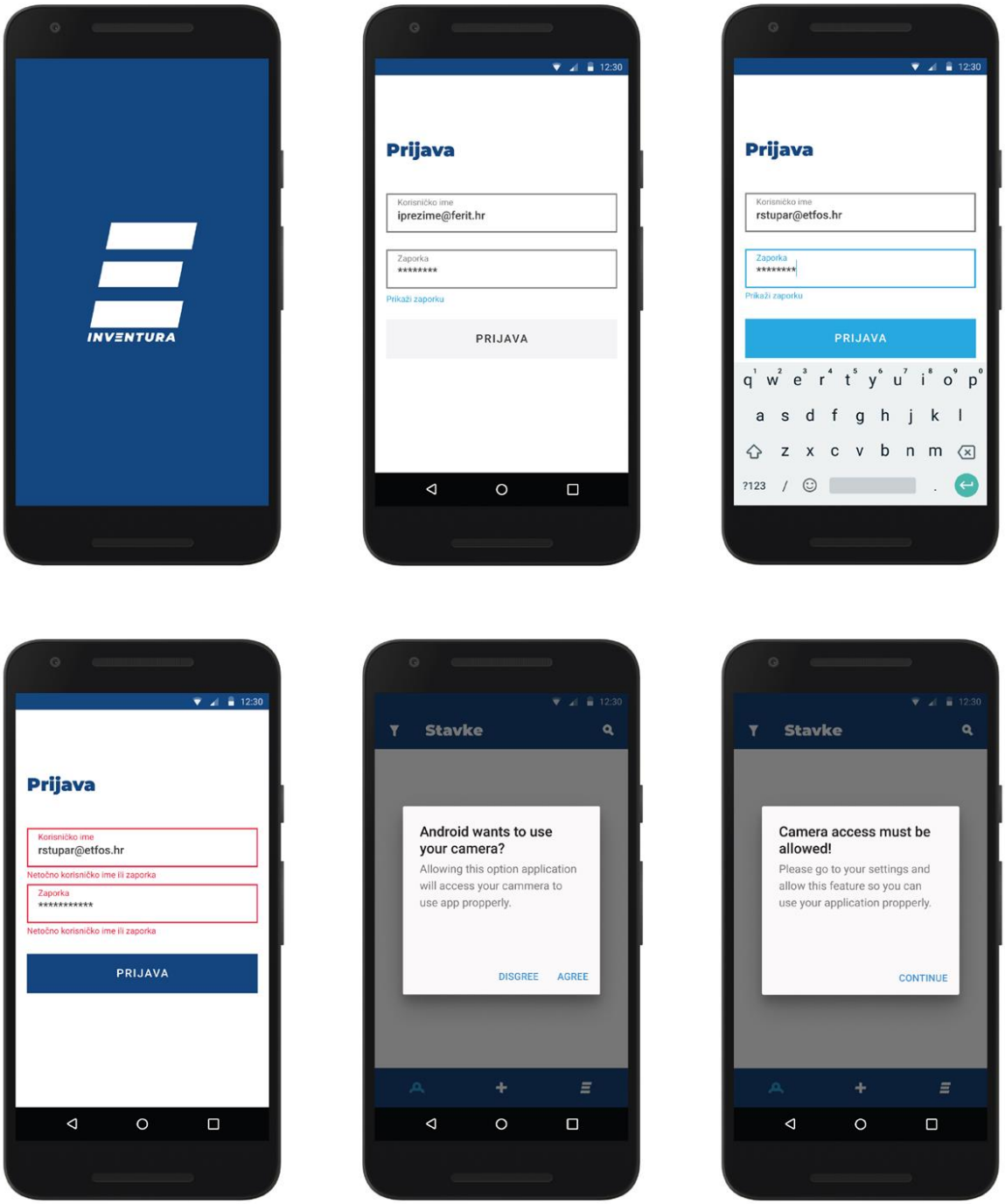
Slika 4.7 Style Guide 1/2

Slika 4.8 prikazuje drugi dio “style guide” na kome se nalaze definirani elementi poput: CTA (engl. Call To Action button), ikonice, navigacije i polja unosa.



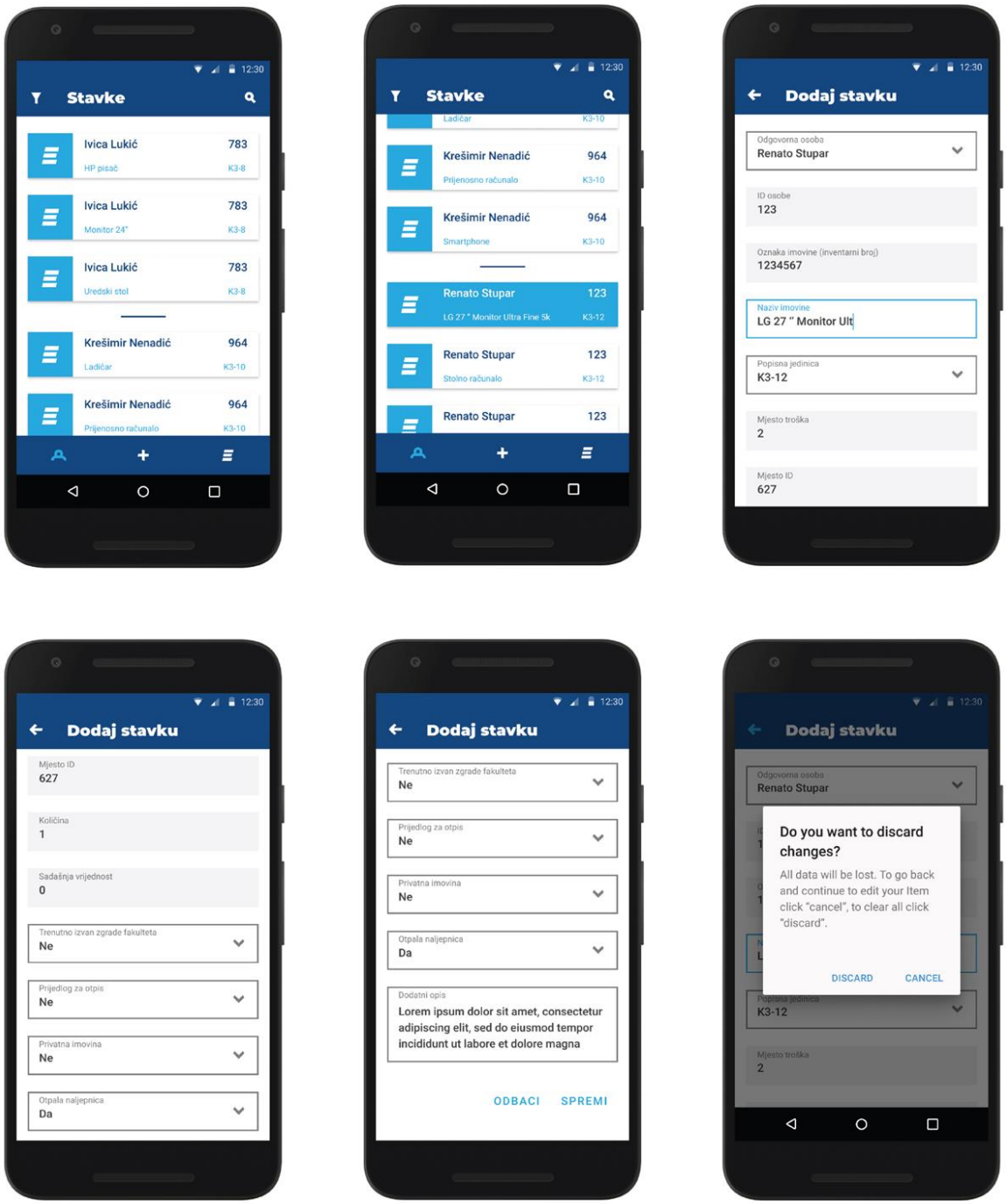
Slika 4.8 Style Guide 2/2

Na slici 4.9 se vide makete dizajniranih ekrana koje prikazuju: ulazni ekran (engl. splash screen), ekran za prijavu u aplikaciju (engl. login screen) i skočne poruke sustava za odobrenje korištenja nekih aplikacija na pametnom telefonu korisnika (engl. pop-up messages, modals).



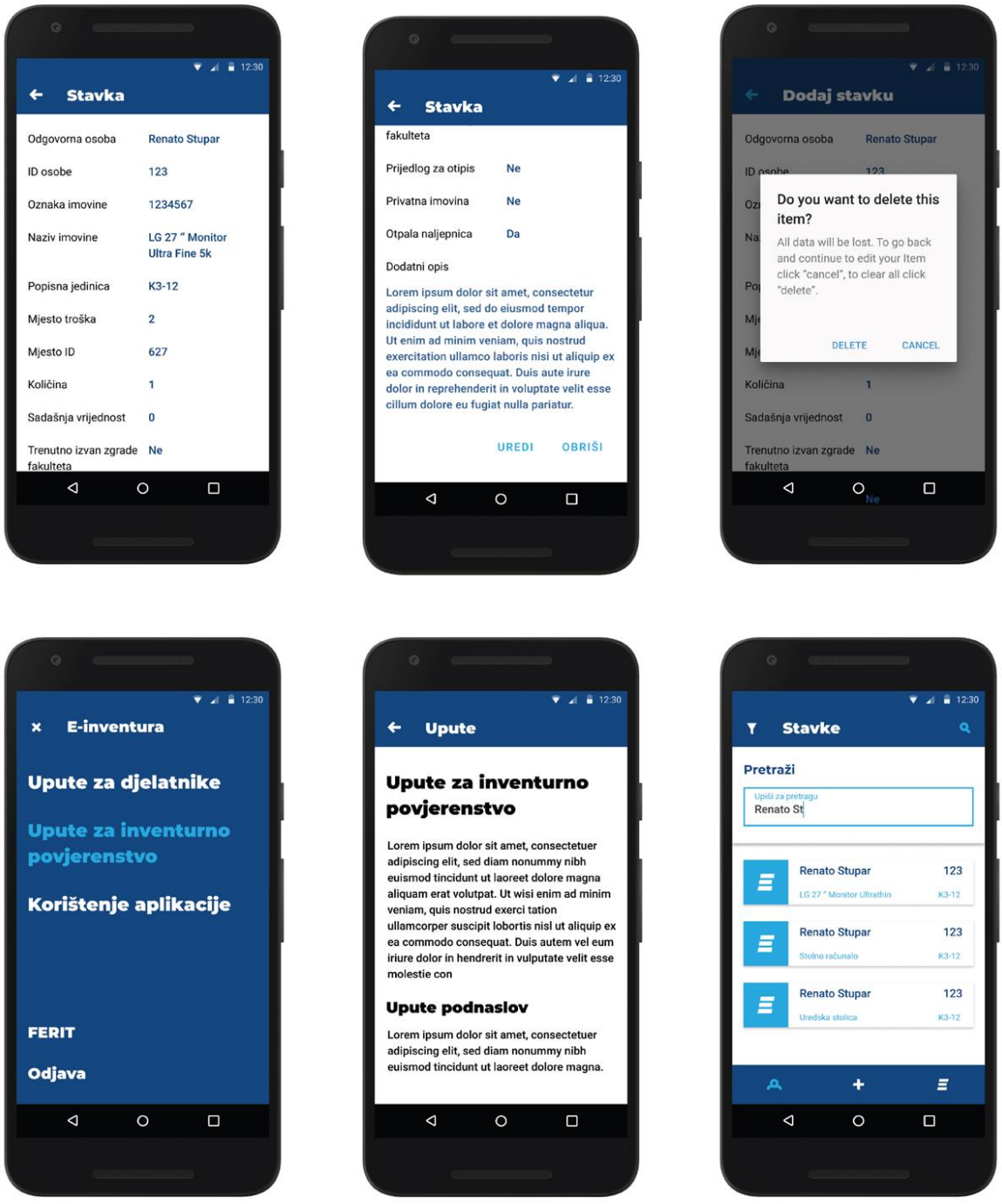
Slika 4.9 Makete ekrana 1/4

Slika 3.10 prikazuje sljedeće makete ekrana na kojima se vide: početni zaslon - sučelje svih stavki koje vidi inventarni povjerenik, dodavanje i uređivanje pojedine stavke; skočna poruka za potvrdu odbacivanja promjena napravljenih u pojedinoj stavci.



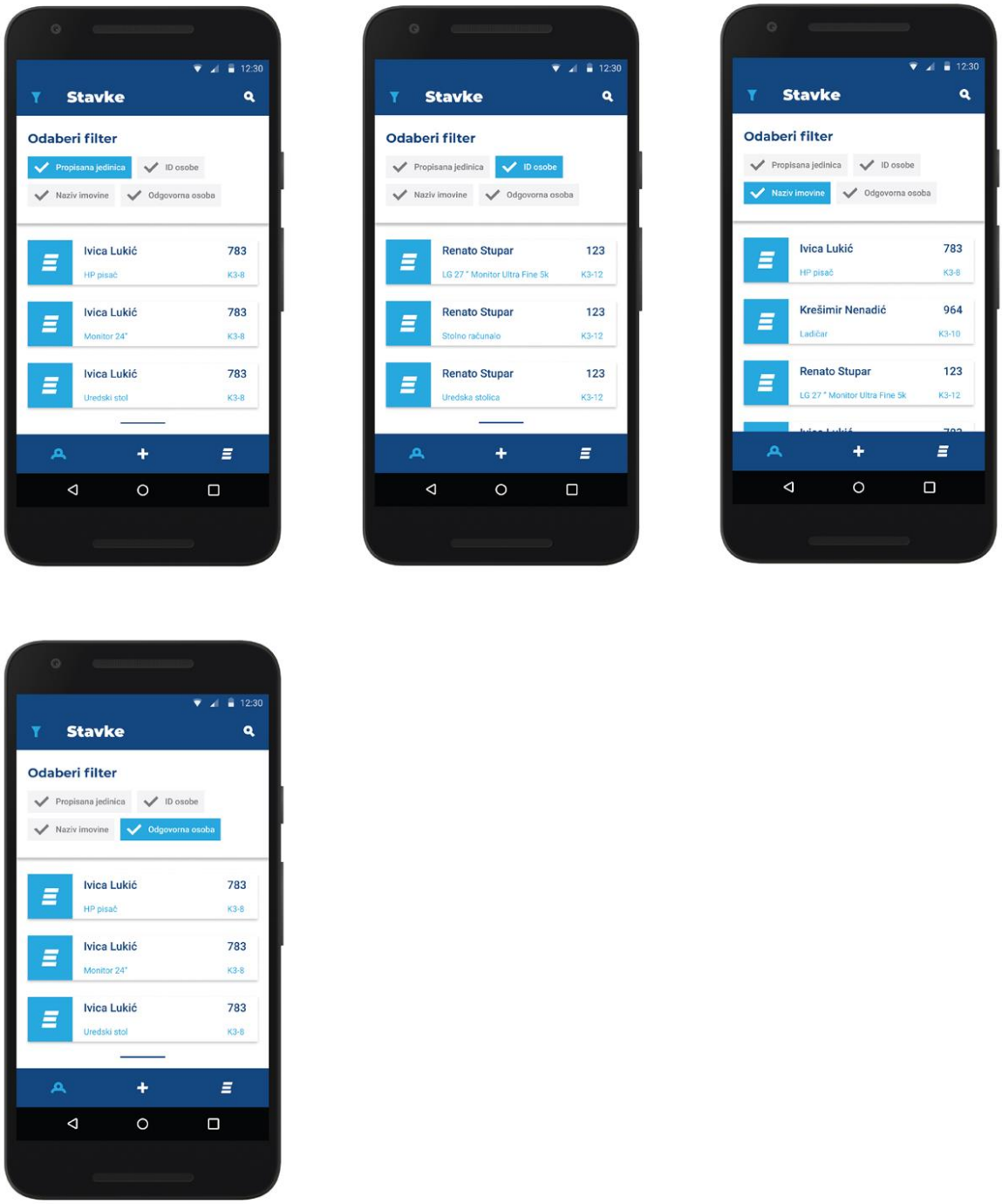
Slika 4.10 Makete ekrana 2/4

Sljedeći ekrani maketa aplikacije na slici 4.11 prikazuju: prikaz pojedine stavke i mogućnosti manipuliranja istom, skočnu poruku za potvrdu brisanja pojedine stavke, izbornik s dodatnim opcijama aplikacije (engl. hamburger menu) i prikaz pretrage pojedinih stavki.



Slika 4.11 Makete ekrana 3/4

Slika 4.12 prikazuje makete ekrana aplikacije na kojima se vidi: značajka filtera stavke po pojedinim oznakama, a te su: Odgovorna osoba, ID osobe, Naziv imovine, Propisana jedinica.



Slika 4.12 Makete ekrana 4/4

4.3.Pretvaranje slike u kod

Nakon što su opisno i crtanjem korisničkog sučelja kreirani svi elementi koji određuju dizajn, sljedeća faza razvoja mobilne aplikacije je pretvaranje slika u kod - programski razvoj. Zasloni koji su nacrtani najprije se kreiraju u obliku XML dokumenta pa im se pomoću Java i Kotlin programskih jezika dodaju funkcionalnosti. Za dovršavanje ovog koraka korišten je programski alat Android Studio.

Android Studio je službeno integrirano razvojno okruženje (integrated development environment) za Googleov operativni sustav Android, izgrađeno na JetBrainsovom IntelliJ IDEA softveru i dizajnirano posebno za Android razvoj. Dostupan je za preuzimanje na Windows, macOS i Linux operativnim sustavima. [5]

Kada se kreira Android aplikacija u Android Studio to podrazumijeva kreiranje zasebnog projekta za svaku novu aplikaciju koju razvijamo. Projekt podrazumijeva radni prostor koji se sastoji od izvornog koda, resursa, testiranja koda i izgradnje konfiguracija, te se sastoji se od niza dokumenata i klasa koji definiraju aplikaciju.

4.3.1. XML – programski razvoj izgleda

XML (engl. eXtensible Markup Language) je softverski i hardverski neovisan alat za pohranu i transport podataka koji se koristi za razvoj izgleda digitalnog proizvoda - aplikacije, kako je ranije već spomenuto. XML datoteke pohranjivanjem sadrže podatke o izgledu, stilu, poziciji svih elemenata koji su vidljivi na zaslonu korisničkog sučelja. Svaki element predstavlja klase u programskom jeziku. Razvojem izgleda nastaju objekti na kojima su vršene određene radnje.

Android Studio podrazumijeva tri vrste izgleda elemenata koji definiraju način slaganja elemenata, a ti su: linearni (engl. LinearLayout), relativni (engl. RelativeLayout) i izgled s ograničenjima (engl. ConstraintLayout).

U linearnom izgledu elementi se slažu jedan ispod drugoga, dok se u relativnom izgledu slažu u odnosu jedni na druge. Izgled s ograničenjima je izgled koji nudi prilagodljive i fleksibilne načine prikaza.

Svaki od elemenata mora sadržavati atribute poput visine i širine. Svakome elementu dodjeljena je jedinstvena oznaka (id) pomoću koje se pristupa pojedinom elementu. Osim visine i širine

ukoliko se koristi izgled s ograničenjima potrebno je postaviti ograničenja svakome elementu. Ograničenja se mogu postaviti u odnosu na roditeljski element ili bilo koji susjedni element.

Osim visine i širine elementu je moguće postaviti margine i uvlake, boju, veličinu teksta, pozadinu, poravnanje i razne druge atribute koji ih opisuju.

4.3.2. Prikaz koda po izgledu i značajkama korisničkog sučelja

U nastavku su prikazani dijelovi koda koji opisuju izgled i značajke aplikacije za:

- Login
- Main
- Add item
- Scan code
- Hamburger menu
- Search
- Filter

Login

Slika 4.13 prikazuje kod za *LoginActivity* koji predstavlja početni ekran nakon što se aplikacija pokrene. U njemu se nalaze gumb za ulazak u glavni dio aplikacije, dva polja za unos korisničkog imena i zaporke, naslovi pojedinih polja te poruke u slučaju neispravnih podataka za prijavu. Izgled sučelja opisan kodom prikazan je na slici 4.9.

```
LoginBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String username = UsernameEt.getText().toString();
        String password = PasswordEt.getText().toString();

        if ("admin@ferit.hr".equals(username) && "admin".equals(password)) {
            Intent i = new Intent(MainActivity.this, ContentActivity.class);
            startActivity(i);
            finish();
        } else if (!"admin@ferit.hr".equals(username) &&
!"admin".equals(password)) {
            UsernameErrorTv.setVisibility(View.VISIBLE);
            PasswordErrorTv.setVisibility(View.VISIBLE);
        } else if (!"admin@ferit.hr".equals(username) &&
"admin".equals(password)) {
            UsernameErrorTv.setVisibility(View.VISIBLE);
            PasswordErrorTv.setVisibility(View.GONE);
        } else if ("admin@ferit.hr".equals(username) &&
!"admin".equals(password)) {
            UsernameErrorTv.setVisibility(View.GONE);
            PasswordErrorTv.setVisibility(View.VISIBLE);
        } else {
            UsernameErrorTv.setVisibility(View.GONE);
            PasswordErrorTv.setVisibility(View.GONE);
        }
    }
});
```

Slika 4.13 Isječak koda: Login

Main

Slika 4.14 prikazuje kod za *MainActivity*, koji glavni dio aplikacije jer se u njemu nalazi najveći dio funkcionalnosti. U njemu se nalazi lista svih unesenih podataka koje se može uređivati i brisati. Uređivanje je moguće kratkim pritiskom na željeni element liste dok se brisanje vrši dugim pritiskom na željeni element. Prilikom brisanja elementa pojavljuje se dijalog kao dodatna provjera prije brisanja. Također, u njemu se nalaze gumbovi za filtiriranje unesenih podataka, pretraživanje unesenih podataka, skeniranje novog elementa te otvaranje Hamburger menu-a. Izgled sučelja opisan kodom prikazan je na slici 4.10.

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getSupportActionBar().hide();
        setContentView(R.layout.activity_content);
        presenter = new ItemListPresenter(this,
DatabaseManager.getDatabaseInstance());

    }

    @Override
    protected void onResume() {
        super.onResume();

        presenter.setView();
    }

    @Override
    public void initView() {
        PropisanaJedinicaCB = findViewById(R.id.pJedinicaCB);
        NazivImovineCB = findViewById(R.id.nazivImovineCB);
        IdOsobeCB = findViewById(R.id.idOsobeCB);
        OdgovornaOsobaCB = findViewById(R.id.odgovornaOsobaCB);
        ScanIB = findViewById(R.id.scanBtn);
        SearchTv = findViewById(R.id.searchTv);
        searchLL = findViewById(R.id.searchLL);
        filterLL = findViewById(R.id.filterLL);
        ItemListLv = findViewById(R.id.itemListLv);
        ItemListLv.setTextFilterEnabled(true);
        InfoIB = findViewById(R.id.infoIB);
        AppIB = findViewById(R.id.appIB);
        FilterIB = findViewById(R.id.filterIB);
        SearchEt = findViewById(R.id.searchEditText);
      SearchIB = findViewById(R.id.searchIB);
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    requestWindowFeature(Window.FEATURE_NO_TITLE);
    getSupportActionBar().hide();
    setContentView( @Override
public void navigateToAddItem() {
    Intent intent = new Intent(this, AddItemActivity.class);
    startActivity(intent);
}


@Override
public void showItems(List<Item> itemList) {
    itemListAdapter = new ItemListAdapter(this, itemList);
    ItemListLv.setAdapter(itemListAdapter);
}
```

Slika 4.14 Isječak koda: Main

Add item

Slika 4.15 prikazuje kod za dodavanje podataka pokreće se automatski nakon skeniranja barkoda ili QR koda. Sva polja za unos se automatski popunjavaju podacima dobivenim skeniranjem nakon čega je pojedine moguće dodatno urediti. Unutar *AddItemActivity* nalaze se dva gumba, jedan za dodavanje podatka u listu i drugi za odbacivanje skeniranih podataka. Klikom na gumb za odbacivanje pojavljuje se dodatan dijalog kao provjera. Izgled sučelja opisan kodom prikazan je na slici 4.10.

```
    @Override
      public void initView() {
           odgovornaOsoba = getIntent().getStringExtra("odgovornaOsoba");
           idOsobe = getIntent().getStringExtra("idOsobe");
           oznakaImovine = getIntent().getStringExtra("oznakaImovine");
           nazivImovine = getIntent().getStringExtra("nazivImovine");
           popisnaJedinica = getIntent().getStringExtra("popisnaJedinica");
           mjestoTroska = getIntent().getStringExtra("mjestoTroska");
           mjestoId = getIntent().getStringExtra("mjestoId");
           kolicina = getIntent().getStringExtra("kolicina");
           sadasnjaVrijednost = getIntent().getStringExtra("sadasnjaVrijednost");
           trenutnoIzvanZgradeFakulteta =
getIntent().getStringExtra("trenutnoIzvanZgradeFakulteta");
           prijedlogZaOtpis = getIntent().getStringExtra("prijedlogZaOtpis");
           privatnaImovina = getIntent().getStringExtra("privatnaImovina");
           otpalaNaljepnica = getIntent().getStringExtra("otpalaNaljepnica");
           dodatanOpis = getIntent().getStringExtra("dodatanOpis");

           String[] odgovornaOsobaArr = {odgovornaOsoba};
           String[] popisnaJedinicaArr = {popisnaJedinica};
           String[] trenutnoIzvanZgradeFakultetaArr =
{trenutnoIzvanZgradeFakulteta};
           String[] prijedlogZaOtpisArr = {prijedlogZaOtpis};
           String[] privatnaImovinaArr = {privatnaImovina};
           String[] otpalaNaljepnicaArr = {otpalaNaljepnica};

           idOsobeTv.setText(idOsobe);
           oznakaImovineTv.setText(oznakaImovine);
           nazivImovineEt.setText(nazivImovine);
           mjestoTroskaTv.setText(mjestoTroska);
           mjestoIdTv.setText(mjestoId);
           kolicinaTv.setText(kolicina);
sadasnjaVrijednostTv.setText(sadasnjaVrijednost);
           dodatanOpisEt.setText(dodatanOpis);
```

```
 odgovornaOsobaSp = findViewById(R.id.odgovornaOsobaSp);
        idOsobeTv = findViewById(R.id.idOsobeTv);
        oznakaImovineTv = findViewById(R.id.oznakaImovineTv);
        nazivImovineEt = findViewById(R.id.nazivImovineEt);
        popisnaJedinicaSp = findViewById(R.id.popisnaJedinicaSp);
        mjestoTroskaTv = findViewById(R.id.mjestoTroskaTv);
        mjestoIdTv = findViewById(R.id.mjestoIdTv);
        kolicinaTv = findViewById(R.id.kolicinaTv);
        sadasnjaVrijednostTv = findViewById(R.id.sadasnjaVrijednostTv);
        trenutnoIzvanZgradeFakultetaSp =
findViewById(R.id.trenutnoIzvanZgradeFakultetaSp);
        prijedlogZaOtpisSp = findViewById(R.id.prijedlogZaOtpisSp);
        privatnaImovinaSp = findViewById(R.id.privatnaImovinaSp);
        otpalaNaljepnicaSp = findViewById(R.id.otpalaNaljepnicaSp);
        dodatanOpisEt = findViewById(R.id.dodatanOpisEt);

        ArrayAdapter<String> odgovornaOsobaAA = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, odgovornaOsobaArr);
        odgovornaOsobaSp.setAdapter(odgovornaOsobaAA);
        odgovornaOsobaSp.setEnabled(false);

        AddItemBtn = findViewById(R.id.addItemBtn);
        DiscardItemBtn = findViewById(R.id.discardItemBtn);

        AddItemBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                presenter.addItemClicked(odgovornaOsoba, idOsobe,
oznakaImovine, nazivImovineEt.getText().toString(),
                        popisnaJedinica, mjestoTroska, mjestoId, kolicina,
sadasnjaVrijednost, trenutnoIzvanZgradeFakulteta,
                        prijedlogZaOtpis, privatnaImovina, otpalaNaljepnica,
dodatanOpisEt.getText().toString());
                Intent intent = new Intent(AddItemActivity.this,
ContentActivity.class);
                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
                startActivity(intent);
                finish();
            }
        });
    }
}
```

Slika 4.15 Isječak koda: Add item

Scan code

Slika 4.16 prikazuje kod za skeniranje barkoda ili QR koda, a pokreće se klikom na "+" gumb koji se nalazi u *MainActivity*. Nakon toga se otvara kamera mobilnog uređaja i skeniranjem željenog barkoda dohvaćaju se podaci koje on sadrži te se oni prosljeđuju u formu za dodavanje podataka. Izgled sučelja opisan kodom prikazan je na slici 4.10.

```
@Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getSupportActionBar().hide();
        setContentView(R.layout.activity_scan_code);

        intentIntegrator.setPrompt("Scan a barcode or QR Code");
        intentIntegrator.setOrientationLocked(true);
        intentIntegrator.initiateScan();
    }

    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable
Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        IntentResult intentResult =
IntentIntegrator.parseActivityResult(requestCode, resultCode, data);
        if (intentResult != null) {
            if (intentResult.getContents() == null) {
                Intent intent = new Intent(ScanCodeActivity.this,
ContentActivity.class);
                startActivity(intent);
            } else {
                String qrContentData = intentResult.getContents();
                try {
                    JSONObject qrJsonData = new JSONObject(qrContentData);
                    Iterator<String> dataIterator = qrJsonData.keys();
                    Intent i = new Intent(ScanCodeActivity.this,
AddItemActivity.class);

                    while (dataIterator.hasNext()) {
                        String itemKey = dataIterator.next();
                        try {
                            Object itemValue = qrJsonData.get(itemKey);
                            i.putExtra(itemKey, itemValue.toString());
                        } catch (JSONException e) {
                            e.printStackTrace();
                        }
                    }
                    startActivity(i);
                    finish();
```

```
      } catch (JSONException e) {
                    e.printStackTrace();
                    Intent intent = new Intent(ScanCodeActivity.this,
ContentActivity.class);
                    startActivity(intent);
                }
            }
        } else {
            Intent intent = new Intent(ScanCodeActivity.this,
ContentActivity.class);
            startActivity(intent);
        }
    }
```

Slika 4.16 Isječak koda: Scan code

Hamburger menu

Slika 4.17 prikazuje kod za "Hamburger menu" sastoji se od klikabilnih tekstova - gumbova. Pomoću njih se otvaraju zasloni koji sadrže: upute za djelatnike, upute za inventurno povjerenstvo te kratke upute za korištenje aplikacije. Preostala dva gumba služe za odlazak na web stranicu FERIT-a te za odjavu iz aplikacije. Nakon odjave iz aplikacije prikazuje se početni ekran kao i nakon pokretanja aplikacije. Izgled sučelja opisan kodom prikazan je na slici 4.11.

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        requestWindowFeature(Window.FEATURE_NO_TITLE);
        getSupportActionBar().hide();
        setContentView(R.layout.activity_app_info);

        DjelatniciBtn = findViewById(R.id.uputeZaDjelatnikeButton);
        PovjerenstvoBtn =
findViewById(R.id.uputeZaInventurnoPovjerenstvoButton);
        KoristenjeBtn = findViewById(R.id.koristenjeAplikacijeButton);
        FeritBtn = findViewById(R.id.feritButton);
        OdjavaBtn = findViewById(R.id.odjavaButton);
        CloseIBtn = findViewById(R.id.closeIButton);
```

Slika 4.17 Isječak koda: Hamburger menu

Search

Slika 4.18 prikazuje dio koda za pretraživanje unesenih podataka otvara se klikom na pripadajuću gumb koji je u obliku povećala. Sastoji se od polja za unos i na temelju unesene vrijednosti pretražuje se cijela lista i prikazuju se samo oni koji zadovoljavaju zadani kriterij. Izgled sučelja opisan kodom prikazan je na slici 4.11.

```
SearchIB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (searchLL.getVisibility() == View.VISIBLE) {
            searchLL.setVisibility(View.GONE);
            filterLL.setVisibility(View.GONE);

        } else {
            filterLL.setVisibility(View.GONE);
            searchLL.setVisibility(View.VISIBLE);
            SearchEt.addTextChangedListener(new TextWatcher() {
                @Override
                public void beforeTextChanged(CharSequence s, int start, int
count, int after) {

                }

                @Override
                public void onTextChanged(CharSequence s, int start, int
before, int count) {
                    itemListAdapter.getFilter().filter(s.toString());
                }

                @Override
                public void afterTextChanged(Editable s) {

                }
            });
        }
    }
});
```

Slika 4.18 Isječak koda: Search

Filter

Slika 4.19 prikazuje dio koda za filtriranje unesenih podataka. Ova značajka se otvara klikom na pripadajući gumb koji je u označen ikonicom filtera. Sastoji se od četiri polja - "checkbox" ikonice. Svaki od ta četiri polja predstavlja jednu stavku unesenog podatka, te se klikom na svaki od njih podaci sortiraju prema stavci koju to polje predstavlja, a definirano je imenima glavnih atributa koju svaka pojedina inventarna stavka posjeduje. Izgled sučelja opisan kodom prikazan je na slici 4.12.

```
FilterIB.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (filterLL.getVisibility() == View.VISIBLE) {
            searchLL.setVisibility(View.GONE);
            filterLL.setVisibility(View.GONE);

        } else {
            searchLL.setVisibility(View.GONE);
            filterLL.setVisibility(View.VISIBLE);
        }
        PropisanaJedinicaCB.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                itemList = dbInterface.getItems();
                if (PropisanaJedinicaCB.isChecked()) {
                    Collections.sort(itemList, new Comparator() {
                        @Override
                        public int compare(Object o1, Object o2) {
                            Item p1 = (Item) o1;
                            Item p2 = (Item) o2;
                            return
p1.getPopisnaJedinica().compareToIgnoreCase(p2.getPopisnaJedinica());
                        }
                    });
                    itemListAdapter = new
ItemListAdapter(getApplicationContext(), itemList);
                    ItemListLv.setAdapter(itemListAdapter);
                } else {
                    itemListAdapter = new
ItemListAdapter(getApplicationContext(), itemList);
                    ItemListLv.setAdapter(itemListAdapter);
                }
            }
        });
```

Slika 4.19 Isječak koda: Filter

5. ZAKLJUČAK

Razvoj digitalnih proizvoda poput android aplikacije zahtjeva konstantan razvoj i razvoj unutar procesa koji je u korak sa vremenom i razvojem tehnologije. Povrh toga, ljudi su ti za koje razvijaju digitalni proizvodi, a čovjek i njegova svijest se razvijaju usporedno sa razvojem tehnologije. Premda nisu svi digitalni proizvodi namijenjeni da budu direktna pomagala u svakodnevnom životu, čak i oni koji su prisutni isključivo u svrhu zabave neposredno sudjeluju u olakšavanju svakodnevnog života. Kao i u svemu postoji tendencija da se novonastala pomagala koriste u krivu svrhu, a takva nastojanja treba apostrofirati i stvarati svijest o zlouporabi.

Aplikacija je izvezena u .apk formatu, instalirana i testirana na HUAWEI Mate 40 Pro pametnom telefonu koji koristi Android operacijski sustav. Sve osmišljene značajke funkcioniraju besprijekorno. Mobilna aplikacija spremna je za upotrebu.

LITERATURA

[1] Vukić, Feđa. PRILOG POZNAVANJU TEORIJE DIZAJNA U HRVATSKOJ, hrčak - Portal hrvatskih znanstvenih i stručnih časopisa, 2006., https://hrcak.srce.hr/18282

[2] Klemp, Klaus. Dieter Rams: The Complete Works. Phaidon Press, 2020. Google books., https://books.google.hr/books/about/Dieter_Rams.html?id=6aeCzQEACAAJ&redir_esc=y

[3] Abras, Chadia, et al. "User-Centered Design." User-Centered Design, Sage Publications, 2004, https://d1wqtxts1xzle7.cloudfront.net/6190316/10.1.1.94.381-libre.pdf?1390844883=&response-content-disposition=inline%3B+filename%3DUser_centered_design.pdf&Expires=1676460722&Signature=PBpuPWxYx6WlBzuh6JHzZe0~vo5hHFarzexCA4O71j0~kRIhCZDEnh26l0T-hndS9fFwXKh6m4nzbH3dIlC8tm-t6nt2E~P0l3RUZbud~hzt3uptoMFTO7qe01GefOISyvVKqsG-ZTHuF2NF8vk6sjowT8wHoFhc~GotVnVwagjj8GYpecHDT7xzCKBsHjuvK970CBAJUBZ6KPbj-hipolLi1bRV9X7aLcixy0OZXzoGdkM-WnYHKmkHLIIccIDQ9KEtebQz7VAJOcbymcgSPMdSMnSfv-0~VxxT5O-dpXRiG66BAa1wS30k~dCX-OEZcb6PwiTg36GU0-d7IyO95g__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA

[4] Foundation, Interaction Design. "What is User Interface (UI) Design? | IxDF." Interaction Design Foundation, 2022, https://www.interaction-design.org/literature/topics/ui-design

[5] Android Studio., https://en.wikipedia.org/wiki/Android_Studio

SAŽETAK

Naslov: Android aplikacija za pomoć pri obavljanju inventure

Ovaj rad sadrži sve elemente razvojnog procesa kreiranja mobilne aplikacije počevši od koncepta koji se sastoji od analize postojećeg digitalnog proizvoda, preko postavljanja temeljnih uvjeta za vizualni prikaz svog sadržaja aplikacije, vizualizacija sadržaja i dodavanje ponašanja značajkama mobilne aplikacije.

Aplikacija služi kao pomagalo inventurnom povjereniku prilikom sakupljanja i bilježenja informacija o inventaru.

Ključne riječi: android, aplikacija, inventura

ABSTRACT

Title: Android application for stocktaking

This work contains all the elements of the development process for creating a mobile application starting from the concept which consists of the analysis of the existing digital product, through the setting of the basic conditions for the visual display of the application's content, the visualization of the content and the addition of behaviour to the mobile application's features.

The application serves as an aid to the inventory commissioner when collecting and recording information about the college's inventory.

Keywords: android, application, inventory

ŽIVOTOPIS

Renato Stupar rođen je 10.10.1986. godine u Osijeku. Od tada do danas živi, školuje se i radi u Osijeku. Osnovnu školu “Mladost” i osnovnu glazbenu školu “Franje Kuhača” pohađa i završava u Osijeku 2001. godine. Nakon osnovne škole upisuje četverogodišnju srednju “Graditeljsko - geodetsku školu Osijek”, smjer “Građevinski tehničar Visokogradnje”, te istu završava 2005. godine u redovnom roku. Nakon srednje škole, te iste 2005. godine upisuje “Elektrotehnički fakultet u Osijeku”. Stjecajem niza okolnosti poput samofinanciranja troškova života i plaćanja iznosa participacija za školarinu svake pojedine akademske godine zapošljava se sljedeće godine nakon upisa studija. Sve te okolnosti su utjecale na dug period studiranja u kojem se i ime samog fakulteta mijenja u “Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku”, koji uz puno napora dovršava u 2023. godini. Trenutno ima svoje malo poduzeće koje funkcionira kao paušalni obrt za pružanje usluga dizajna i naučavanja istog, te uz mnoge klijentske projekte, radi kao predavač Grafičkog i UX dizajna na Visokom učilištu Algebra. Zaručen je sa zakazanim datumom vjenčanja u narednoj godini.

U Osijeku, 2023. godine

Renato Stupar

_____