



CORNUX

TEHNIČKA DOKUMENTACIJA

AUTOR

FILIP JURKOVIĆ

Srednja škola za elektrotehniku i računalstvo



MENTOR

Sanja Sochor, mag.educ.inf.

SADRŽAJ

1.	UVOD	1
1.1.	OPIS TRENUTNOG NAČINA RADA	1
1.2.	O AUTORU.....	1
1.3.	IDEJA PROJEKTA.....	1
1.4.	PROBLEMI KOJE CORNUX RJEŠAVA.....	2
1.5.	MOGUĆNOSTI CORNUXA	2
2.	DETALJAN OPIS RADA.....	3
2.1.	ZASLON ZA PRIJAVU.....	3
2.2.	POČETNI ZASLON	3
2.3.	IME KORISNIKA	4
2.4.	SKENIRANJE	5
2.5.	MJESEČNA STATISTIKA	5
2.6.	ZASLON ZA PROVJERU PODATAKA	6
2.7.	BOČNI IZBORNIK	7
2.8.	STATISTIKA	8
2.8.1.	<i>Godišnja statistika</i>	<i>8</i>
2.8.2.	<i>Mjesečna statistika</i>	<i>9</i>
2.9.	O NAMA	10
3.	TEHNIČKE ZNAČAJKE	12
3.1.	LISTA ZNAČAJKI.....	12
3.1.1.	<i>Skeniranje</i>	<i>12</i>
3.1.2.	<i>Provjera.....</i>	<i>12</i>
3.1.3.	<i>Opcije</i>	<i>12</i>
3.1.4.	<i>Profil korisnika</i>	<i>12</i>
3.1.5.	<i>Ostalo</i>	<i>12</i>
3.2.	SISTEMSKA KONFIGURACIJA.....	13
3.3.1.	<i>Minimalna sistemska konfiguracija.....</i>	<i>13</i>
3.3.2.	<i>Preporučena sistemska konfiguracija.....</i>	<i>13</i>
3.3.3.	<i>Potreban softver.....</i>	<i>13</i>
3.3.	TEHNOLOGIJE.....	13
3.4.	KORISNIČKO SUČELJE	15
3.5.	ONLINE KOMUNIKACIJA	15
3.5.1.	<i>Prijava korisnika</i>	<i>15</i>
3.5.2.	<i>Online baza podataka</i>	<i>15</i>
3.5.3.	<i>Sigurnost.....</i>	<i>15</i>
3.6.	PROCES IZRADA APLIKACIJE	16
3.6.1.	<i>Ideja</i>	<i>16</i>
3.6.2.	<i>Razvoj metode za obradu narudžbenica</i>	<i>16</i>
3.6.3.	<i>Stvaranje web servera.....</i>	<i>16</i>
3.6.4.	<i>Dizajn korisničkog sučelja</i>	<i>16</i>
3.6.5.	<i>Prototip</i>	<i>17</i>
3.6.6.	<i>Beta verzija</i>	<i>17</i>
3.6.7.	<i>Prva javna verzija</i>	<i>17</i>
3.7.	ODRŽAVANJE SOFTVERA	17
4.	MARKETING.....	18
5.	ZAKLJUČAK	19

Popis slika

SLIKA 1: UNOS ROBE	2
SLIKA 2: ZASLON ZA PRIJAVU.....	3
SLIKA 3: GLAVNI ZASLON.....	4
SLIKA 4: MOGUĆNOSTI SKENIRANJA.....	5
SLIKA 5: TABLICA DOSTAVNICE	6
SLIKA 6: BOČNI IZBORNIK	7
SLIKA 7: STATISTIKA	8
SLIKA 8: GODIŠNJA STATISTIKA	9
SLIKA 9: MJESEČNA STATISTIKA	9
SLIKA 10: O NAMA	10
SLIKA 11: KONTAKT	11
SLIKA 12: KORIŠTENE TEHNOLOGIJE.....	14

1. Uvod

1.1. Opis trenutnog načina rada

Za potpuno razumijevanje rada i samog razvoja ove aplikacije te cjelokupnog ekosistema potrebno je znati ponešto o fiskalizaciji i unosu robe na skladište. Djelatnici u ugostiteljstvu ali i svim djelatnostima koje se bave nekim načinom prodaje dobara (robe) moraju imati inventar na skladištu kako bi se moglo pratiti koliko je robe prodano i koliko kupljeno, kao i u svakom skladištu. Pri svakoj nabavi robe operater treba unijeti sva kupljena dobra za taj objekt. Danas je unos robe još uvijek vrlo manualan proces te se apsolutno sve mora napraviti ručno, što često traje i više od sat vremena dnevno. Svaki proizvođač radi svoj format narudžbenice te samim time nastaje problem automatizacije unosa jer ne postoji univerzalan format koji možemo prepoznavati iznova za svaku narudžbu. Ovo je primjer jedne narudžbenice.

1.2. O autoru

Moje ime je Filip Jurković , učenik sam Srednje škole za elektrotehniku i računalstvo (SŠER) u Rijeci, smjer tehničar za računalstvo.

U školi sam imao priliku naučiti programirati za razne platforme. Većinu svojeg slobodnog vremena provodim učeći programiranje te sam veliku većinu znanja stekao na internetu i na radionicama Centra tehničke kulture.

Uz SŠER pohađam Centar tehničke izvrsnosti u Rijeci, gdje se upoznajem sa različitim novim tehnologijama što mi je dalo veliku prednost u izradi ovog projekta.

Kao i većina ostalih učenika svoje ljetne praznike provodim radeći kao konobar u obiteljskom restoranu što mi je dalo potpuni uvid u problem koji rješavam ovom aplikacijom i potrebne informacije što bi se moglo napraviti bolje i brže.

1.3. Ideja projekta

Unos narudžbenica je vrlo naporan proces te zahtjeva mnogo vremena, ako se radi po zakonskim mjerama. S obzirom da na otoku Krku mnogo ljudi živi od 3 mjeseca turizma, mnogi obrtnici nemaju svakodnevno 1-2 sata vremena za unos robe na skladište što ih dovodi do dva rješenja: mogu platiti zaposlenika da to radi za njih ili mogu ne unositi robu te riskirati kaznu od preko 50.000 kuna. Kada sam ove sezone htio ubrzati taj proces počeo sam tražiti po internetu te sam se jako iznenadio kada nisam našao ni jedno rješenje za ovaj problem koje bi pomoglo obrtnicima i poduzetnicima u cijeloj Hrvatskoj. Samim time sam otkrio da postoji ogromna potražnja za takvim rješenjem i ogromno tržište koje je kompletno otvoreno. Ideja ove aplikacije jest da obrtnik sa svojim pametnim telefonom može slikati narudžbenicu te će klikom jednog gumba sa narudžbenice izvući sve potrebne podatke koji će se automatski unijeti u fiskalnu kasu.

U razvoju ove ideje sam i uspio. Nakon godinu dana istraživanja koda napravio sam glavni dio projekta, servis koji prepoznaje sve potrebne podatke te aplikaciju koja omogućava svim korisnicima da to jednostavno i efikasno iskoriste.

1.4. Problemi koje Cornux rješava

- Ubrzavanje unosa robe
- Mogućnost praćenja mjesečnog unosa
- Unos robe bez pristupa fiskalnoj kasi
- Potpuna kontrola nad procesom

1.5. Mogućnosti Cornuxa

Pomoću ove aplikacije korisnik može unijeti robu na skladište u nekoliko minuta. To može postići na nekoliko načina.

Prvi je trenutačni način, pomoću kamere pametnog telefona. Kada je korisnik u prisustvu narudžbenice, može ju slikati i odmah unijeti u fiskalnu kasu. Tako se korisnik odmah rješava obaveze unosa.

Drugi način je pomoću galerije pametnog telefona. Korisnik može više dana slikati narudžbenice koje dolaze u njegov objekt, te na slobodan dan može otvoriti galeriju i unijeti svu robu isto kao i prvim načinom. Ovo omogućuje korisniku fleksibilnost pri unosu te je sličnije trenutačnom ručnom pristupu.

Tijekom provjere korisnik može provjeriti sve podatke koje unosi u tabličnog formatu te može promijeniti svaki podatak kako ne bi došlo do krivog unosa te samim te do nepodudaranja između skladišta i stvarnog stanja.

Važno je napomenuti kako Cornux aplikacija ne sprema niti zadržava ikakve podatke koje korisnik skenira pošto mnogi potencijalni cijene privatnost te bi to bilo direktno kršenje prava korisnika.

Želim istaknuti da aplikacija vodi računa o jako puno scenarija od nepredvidljivih radnji korisnika, prisutnost internetske veze, pa sve do točne razlučivosti te dimenzija uređaja. U dokumentaciji sam objasnio veće i bitnije situacije. U pozadini se odvija velik broj drugih provjera, kao na primjer : uspostavljanje komunikacije API protokolom, provjera valjanosti unosa slike itd. Cilj svih tih, možda naizgled nebitnih, provjera te vođenje računa o jako velikom broju scenarija je doprinos maksimalnog mogućeg korisničkog iskustva.

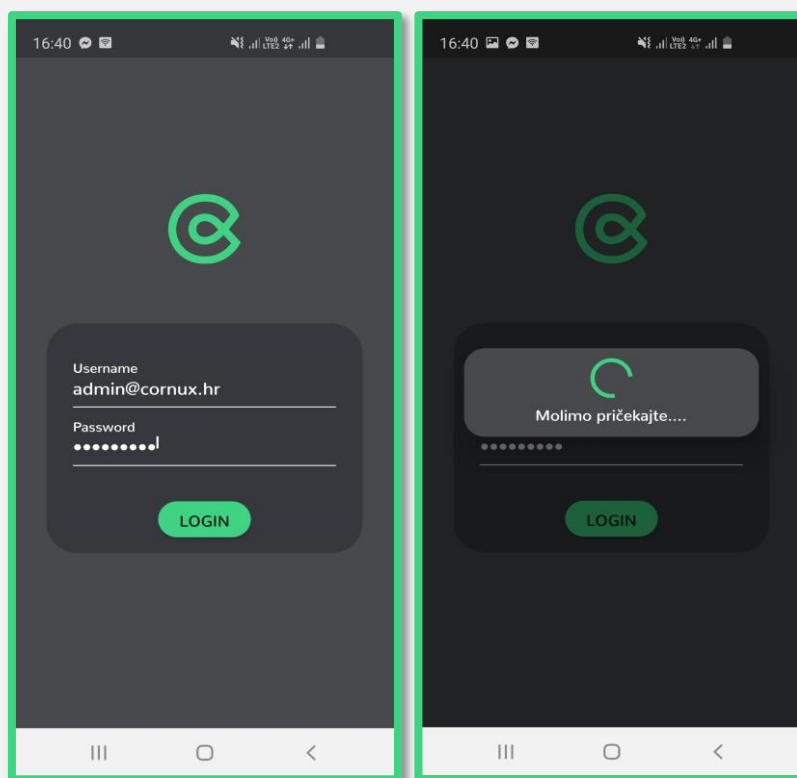


Slika 1: Unos robe

2. Detaljan opis rada

2.1. Zaslون za prijavu

Zaslون za prijavu je prvo što korisnik vidi kada otvori aplikaciju.



Slika 2: Zaslون za prijavu

On je osmišljen da aplikaciji mogu pristupiti samo korisnici koji imaju licencu za svoj obrt. Svaki korisnik će moći odabrati koliku količinu računa žele obraditi svaki mjesec te će takva biti i cijena godišnje licence. Za bazu korisnika korišten je Googleova Firebase usluga pomoću koje svaki korisnik dobiva svoju mail adresu te lozinku za pristup aplikaciji. Koristio sam Firebase zato što je to industrijski standard za prijavu i registraciju korisnika te zato što mi Firebase pruža jedinstveni identifikacijski ključ za svakog korisnika te time mogu pratiti mjesečne tarife svakog korisnika. Nakon što korisnik ispuní podatke za prijavu, aplikacija poziva Firebase API kako bi provjerila jesu li korisnički podatci ispravni. Nakon prijave aplikacija kontaktira web server sa identifikacijskim ključem kako bi povukao mjesečnu tarifu, broj obrađenih dokumenata te ime korisnika. Ovaj korak se događa između prijelaza sa zaslona za prijavu na početni zaslon.

2.2. Početni zaslon

Početni zaslon je ujedno i glavni zaslon koji će korisnik koristiti većinu vremena koje provodi na aplikaciji. Ovaj zaslon je osmišljen tako da sadrži samo stavke koje su važne korisniku te ga ne želimo preopterećivati informacijama koje mu nisu potrebne za korištenje aplikacije.

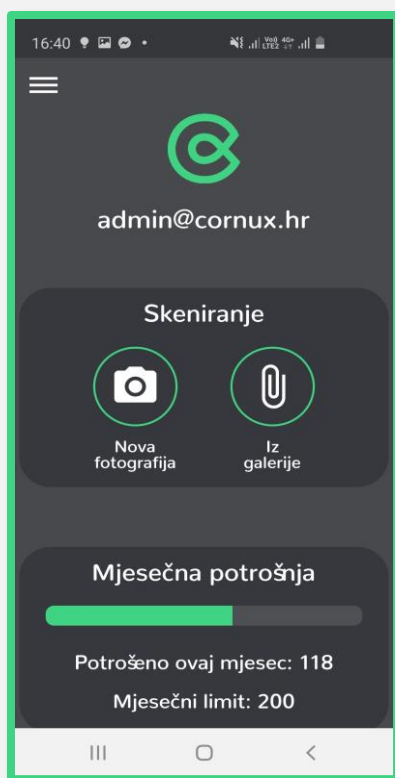
Sa početnog zaslona također možemo pristupiti bočnom izborniku u gornjem lijevom kutu preko kojeg možemo pristupiti svim ostalim zaslonima :

- Statistika
- O nama
- Log out

Bočni izbornik i sve ostale stranice opisujemo kasnije u dokumentaciji.

Glavni zaslon sadrži nekoliko stavki:

- Ime korisnika
- Skeniranje
- Mjesečna statistika



Slika 3: Glavni zaslon

2.3. Ime korisnika

Ova stavka prikazuje ime korisnika te služi za povlačenje svih drugih podataka o korisniku. Pri prijavi u aplikaciju koristimo identifikacijski kod kako bi poslali zahtjev na naš web server koji iz baze podataka prema primarnome ključu izvlači sljedeći podatke:

- Mjesečni limit
- Broj iskorištenih obrada
- OIB firme
- Ime korisnika

Ime korisnika i njegov identifikacijski ključ koristimo kasnije u aplikaciji kada obrađujemo narudžbenicu te ju šaljemo na računalo za unos u fiskalnu kasu.

2.4. Skeniranje

Ovo je glavna funkcija aplikacije te je cijeli Cornux izgrađen oko nje. Skeniranje je prva stavka koju korisnik vidi na početnom zaslonu pošto je to i sama svrha aplikacije. Skeniranje se dijeli na dvije funkcije:

- Nova fotografija
- Iz galerije



Slika 4: Mogućnosti skeniranja

Kada želimo odmah unijeti dobivenu narudžbenicu odabiremo stavku Nova fotografija koja nam otvara fotoaparat pametnog telefona pomoću kojeg slikamo narudžbenicu i odmah je šaljemo na web server pomoću HTTP POST metode koja nam vraća odgovor u JSON formatu.

Ako imamo otprije slikane narudžbenice te ih nismo unesli u fiskalnu kasu odabrati ćemo stavku Iz galerije. Prilikom odabira otvara nam se galerija našeg telefona iz koje možemo odabrati narudžbenice koje želimo obraditi. Nakon odabira slike šaljemo je na web server pomoću HTTP POST metode koja nam vraća odgovor u JSON formatu.

Napomena: prilikom odabira iz galerije možemo odabrati samo jednu sliku istovremeno. Obrada više slika istovremeno nalazi se na listi za daljnji razvoj aplikacije.

Nakon odabira jedne od dvije opcije skeniranja moramo pričekati određeno vrijeme za obradu koje se trenutno kreće oko 10-12 sekundi zato što je server pisan u Python programskom jeziku. Nakon interpretacije u C programski jezik očekuje se znatno ubrzanje procesa od +15%. Nakon obrade slike prelazimo na zaslon za provjeru podataka.

2.5. Mjesečna statistika

S obzirom da je ovo plaćena usluga i cijena varira ovisno o količini narudžbenica koje se obrađuju svaki mjesec, korisnik mora imati uvid koliko narudžbenica mjesečno može obraditi te koliko ih je već obradio kako ne bi došlo do situacije u kojoj mora nadoplatiti tarifu koju koristi.

2.6. Zaslone za provjeru podataka

Nakon odabira načina skeniranja i obrade narudžbenice na web serveru dolazimo do zaslona gdje su nam prikazani svi podatci koji su nam važni za unos računa u fiskalnu kasu, a to su:

- Ime prodavača
- Broj narudžbenice
- Kupljeni artikli

Ime prodavača i broj narudžbenice prikazujemo u obliku tekstnog polja koje se može izmjenjivati pošto nikada ne možemo reći sa potpunom sigurnošću da će svi izvučeni podatci biti točni.

Kupljeni artikli se prikazuju u tabličnom obliku sa svim njihovim atributima koji variraju od prodavača do prodavača, no nama najvažniji za unos su:

- Ime artikla
- Količina
- Cijena
- PDV
- Cijena + PDV

The screenshots show the 'Tablica dostavnice' application interface. The first screenshot displays the filter menu with options for R.br, Naziv artikla, bk, Mj., 7,13, Cijena, Rab?, Kore, and Vrijednost. The second screenshot shows a list of items with columns for R.br, Naziv artikla, and a list of items. The third screenshot shows a confirmation dialog 'Poslano' over the list.

Col	Cijena	Rab	PDV	Vri
,73	28,80	10,00	25,00	2
,19	50,40	0,00	25,00	4
,02	31,68	10,00	25,00	1
,79		0,00	25,00	4
,51	159,20	22,00	25,00	

R.br	Naziv artikla
1	Svinjska lopatica bk
2	Juneča lopatica/vrat
3	Svinjski but bk
4	Juneći biftek
5	Domaća kobasica
6	Juneći but bk

Slika 5: Tablica dostavnice

S obzirom da svaki prodavač ima različite attribute aplikacija ima mogućnost odabira samo onih atributa koji su nama potrebni za unos robe u fiskalnu kasu. Imamo izbornik u kojem može kvačicom odabrati koje stupce želimo unositi u fiskalnu kasu, a koje nećemo. Ova funkcija je veoma korisna jer

različiti korisnici imaju različite potrebe te time još više generaliziramo aplikaciju, no i dalje ostaje specijalizirana većinom za potrebe ugostiteljstva.

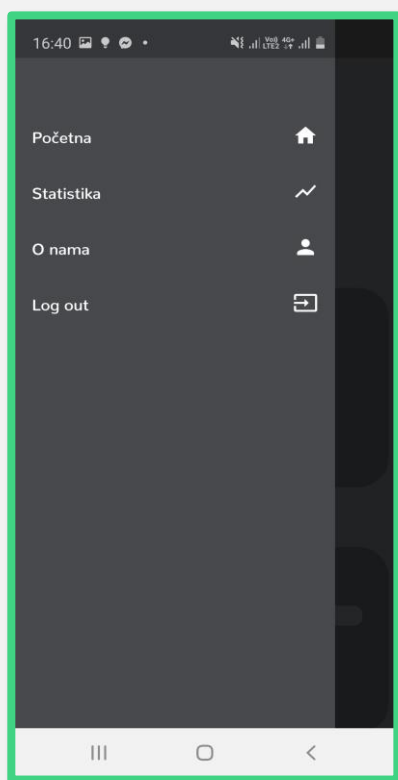
Kao i kod imena prodavača i broja narudžbenice postoji mogućnost da svi podatci neće biti točno obrađeni te zato možemo svaku ćeliju izmjenjivati posebno.

Nakon što smo provjerili da su svi podatci točni u gornjem desnom kutu odabiremo kvačicu čijim pritiskom šaljemo podatke na fiskalnu kasu pomoću REST API protokola. Nakon što su podatci uspješno poslani otvara se dijaloški okvir koji nam naznačuje da su podatci uspješno poslani te nas vraća na početni zaslon.

2.7. Bočni izbornik

Na svakom zaslonu aplikacije u gornjem lijevom kutu imamo ikonu čijim pritiskom otvaramo bočni zaslon koji nam služi za navigaciju kroz aplikaciju. Bočni zaslon sadrži 4 stavke te se uz svaku stavku nalazi ikona koja ju znakovno opisuje, a te stavke su:

- Početna
- Statistika
- O nama
- Log out



Slika 6: Bočni izbornik

Prve tri stavke služe za navigaciju između zaslona aplikacije, no važno jest što za svaku promjenu zaslona moramo prenositi važne stavke kao što su: ime korisnika, identifikacijski ključ te statistika korisnika.

Log out je zadnja stavka na bočnom izborniku te nam ona služi za odjavu iz aplikacije, te njezin odabir aktivira funkciju koja poziva Firebaseov API kako bi najavio odjavu iz sustava.

2.8. Statistika

Zaslon Statistike jest zaslon na kojem korisnik može pratiti koliko narudžbenica obrađuje i koliko robe unosi u fiskalnu kasu na mjesečnoj te godišnjoj razini.



Slika 7: Statistika

Na zaslonu statistike imamo dvije stavke:

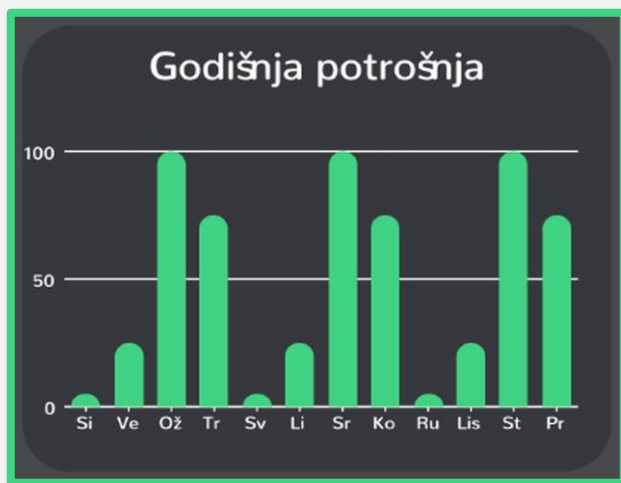
- Godišnja statistika
- Mjesečna statistika

2.8.1. Godišnja statistika

S obzirom da su ciljani korisnici ove aplikacije većinom ugostitelji i obrtnici, u aplikaciji je ugrađena stavka za praćenje godišnje količine obrađenih narudžbenica. Također je važno da korisnik može vidjeti koliki je broj narudžbenica u odnosu na mjesec u godini stoga je godišnja statistika napravljena kao stupčasti grafikon za 12 mjeseci u godini i on se sastoji od:

- Broj narudžbenica na Y osi
- Mjeseci na X osi
- Stupcići kronološki poredani

Godišnja statistika pruža korisniku mogućnost provjere podudaranosti između fiskalne kase i stvarnog stanja kako ne bi došlo do kažnjavanja pri provjeri porezne uprave.



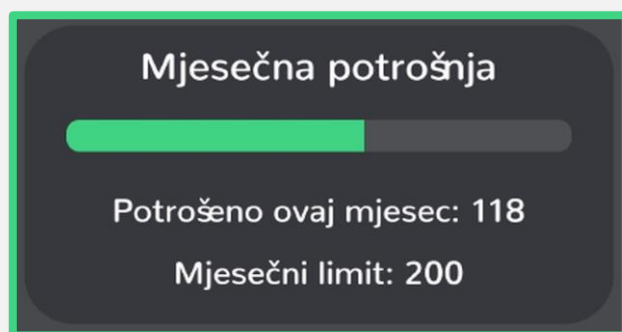
Slika 8: Godišnja statistika

2.8.2. Mjesečna statistika

Pošto je ovo plaćena usluga i cijena varira ovisno o količini narudžbenica koje se obrađuju svaki mjesec korisnik mora imati uvid koliko narudžbenica mjesečno može obraditi te koliko ih je već obradio kako ne bi došlo do situacije u kojoj mora nadoplatiti tarifu koju koristi.

Stavka mjesečne statistike ima tri dijela:

- Progresivni graf
- Mjesečni limit
- Broj skeniranja



Slika 9: Mjesečna statistika

Progresivni grafikon je uveden kako bi korisnik imao vizualni prikaz svog korištenja aplikacije za taj mjesec te nakon što korisnik prijeđe 90% plaćenog limita indikator mijenja boju u crveno.

Mjesečni limit prikazuje koju tarifu koristi korisnik aplikacije te se ona mijenja ukoliko korisnik promijeni svoju tarifu.

Broj skeniranja prikazuje koliko je korisnik napravio skeniranja ovaj mjesec te se povećava sa svakom obradom narudžbenice.

Stavka mjesečne statistike se nalazi na početnom zaslonu kako bi privukla korisniku pažnju o njegovoj uporabi aplikacije.

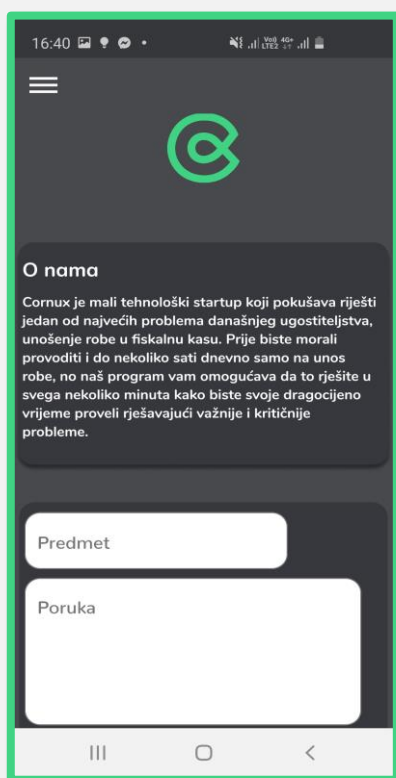
2.9. O nama

Stranica O nama služi korisniku kako bi pronašao podatke o Cornux firmi te mogućnosti kontaktiranja u slučaju potrebe za korisničkom podrškom ili ostalih nepredvidljivih zahtjeva.

Stranica O nama ima 3 stavke:

- O nama
- Kontaktirajte nas
- Gdje nas možete naći

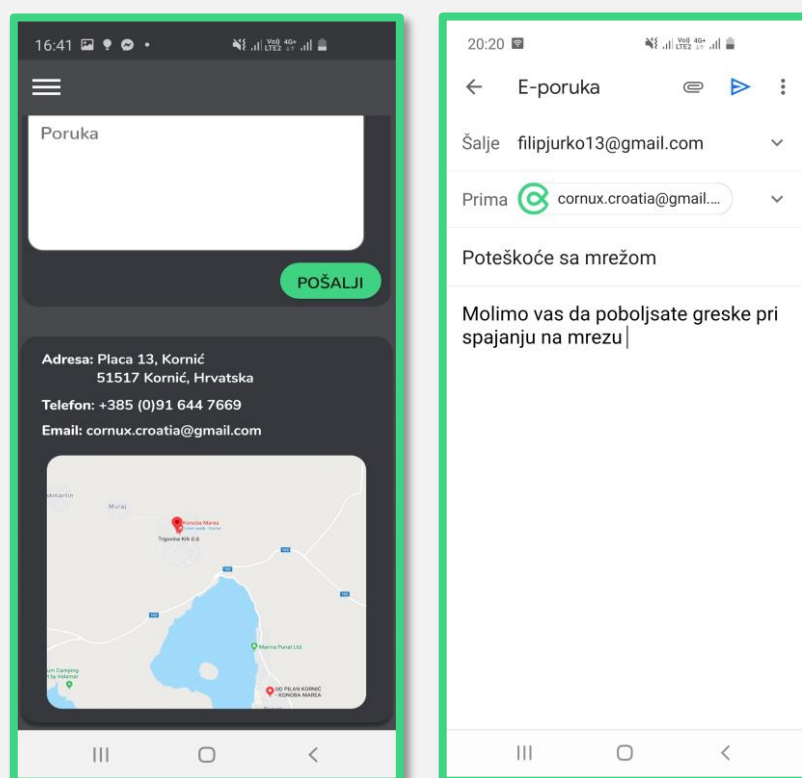
Stavka O nama ukratko govori o firmi i zašto je aplikacija napravljena. Ono služi ukoliko korisnik želi znati ponešto o firmi.



Slika 10: O nama

Stavka Kontaktirajte nas sadrži format za slanje maila na službeni mail firme Cornux.

Korisnik može nas kontaktirati pri bilo kojoj poteškoći sa aplikacijom te će dobiti odgovor na svoju e-mail adresu.



Slika 11: Kontakt

Treća i zadnja stavka sadrži općenite informacije o firmi Cornux te prikaz Google Karta sa lokacijom Cornux firme. Prilikom pritiska na prozor otvara se aplikacija Google Karta pomoću koje možete doći u firmu ako postoji ta potreba.

3. Tehničke značajke

3.1. Lista značajki

3.1.1. Skeniranje

- Slikanje narudžbenice kamerom
- Odabir narudžbenice iz galerije
- Udaljeni pristup fiskalnoj kasi

3.1.2. Provjera

- Promjena teksta
- Odabir određenih atributa
- Odabir stupaca za unos u fiskalnu kasu
- Automatski unos u fiskalnu kasu

3.1.3. Opcije

- Pregled godišnje statistike
- Pregled mjesečne statistike
- Povećanje tarife
- Kontaktiranje korisničke podrške

3.1.4. Profil korisnika

- Prijava putem e-mail adrese
- Sigurnost podataka
- Online baza podataka
- Sinkronizacija online i lokalnih podataka
- Bilježenje obrade narudžbenica

3.1.5. Ostalo

- Podrška za Android i iOS
- Podrška za praktički sve Android mobitele i tablete
- Intuitivno sučelje
- Mogućnost rada na SD kartici
- Mogućnost kontaktiranja podrške

3.2. Sistemska konfiguracija

Navedena sistemska konfiguracija služi kao bolja predodžba potrebnih performansi uređaja. Nju je bitno znati u slučaju da je softver namijenjen uređajima čije komponente mogu mnogo varirati ili da softver traži dobre performanse uređaja. Aplikacija podržava Android i iOS uređaja. Svi mobilni uređaji i tableti koji imaju android verziju 4.1 (Jelly Bean) ili više te Apple uređaji čija je iOS verzija 6 ili više bit će sposobni pokrenuti ovu aplikaciju.

3.3.1. Minimalna sistemska konfiguracija

- Mobilni uređaj ili tablet sa android operacijskim sustavom verzije 4.1 (Jelly Bean)
- Mobilni uređaj ili tablet sa iOS operacijskim sustavom verzije 6 ili više
- CPU Intel Atom® Processor Z2520 1.2 GHz
- Arhitektura procesora: 32 ili 64-bitna
- 128MB slobodne RAM memorije
- 50MB slobodnog prostora
- Zaslon na dodir (ili sa nekom softverskim rješenjem simulacije dodira) rezolucije 144p
- Povezanost sa Internet infrastrukturom

3.3.2. Preporučena sistemska konfiguracija

- Mobilni uređaj ili tablet sa android operacijskim sustavom verzije 5.0 (Lollipop) ili više
- Mobilni uređaj ili tablet sa iOS operacijskim sustavom verzije 10 ili više
- CPU Dual-core 1.7 GHz ili bolje
- 64-bit
- 1 GB ili više RAM memorije
- 50MB ili više slobodnog prostora
- Povezanost sa Internet infrastrukturom

3.3.3. Potreban softver

- Zadani softver podržavane android verzije
- Zadani softver podržavane iOS verzije

3.3. Tehnologije

Aplikacija Cornux izrađena je u programu Android Studio koji je službeni Googleov program za izradu android aplikacija.

Backend aplikacije je pisan u Flutteru. Flutter je najnoviji Googleov framework za pisanje aplikacija koje su kompatibilne na više operativnih sustava. U ovom slučaju to iskorištavamo izradom jedne aplikacije za Android i iOS.

S obzirom da Flutter koristi svoj programski jezik pod imenom Dart u isto vrijeme pišemo kod za backend i frontend kako bi osigurali kompatibilnost između platformi.

Za testiranje koristim svoj osobni mobilni uređaj (Samsung Galaxy S10) te više prijateljevih mobilnih uređaja (Apple Iphone 7, Apple Iphone 5c, Huawei P30 Pro) kako bih osigurao kompatibilnost među platformama i veličinama. Također sam koristio i emulator ugrađen u Android studio. U njemu sam testirao rad aplikacije na najnovijim verzijama androida na mobilnom i tablet uređaju. Koristim i BlueStacks emulator kako bih bolje testirao performanse aplikacije.

Za kontrolu verzija aplikacije koristim GitHub privatni repozitorij koji sam integrirao u Android Studio.

Za neke male dodatke također sam koristio GitHub projekte otvorenog koda te računalni kod iz Googleovog tečaja za android na internetskoj stranici Udacity koji sam prethodno završio. Kako slične aplikacije ne postoje, ne postoji niti projekti otvorenog koda sličnih aplikacija.

Za korisničke račune koristio sam Google Firebase authentication API, a za online bazu podataka Google Firebase Cloud Firestore API. Google Firebase je sigurna platforma sa besplatnim planom koji je za ovu aplikaciju nudi dovoljno mogućnosti i malo ograničenja.

Za obradu samih narudžbenica izradio sam privatni web server koristeći Python programski jezik i Flask Framework radi njegove jednostavnosti. Komunikacija između aplikacije i servera se odvija pomoću HTTP POST GET protokola. Kako bi svaki korisnik mogao pristupiti web serveru, server koristiti Amazonov AWS(Amazon Web services) EC2 hosting.

Radi slanja podataka između aplikacije i servera svi podatci su bili u JSON formatu.



Slika 12: Korištene tehnologije

3.4. Korisničko sučelje

Korisničko sučelje Cornuxa osmišljeno je kako bi bilo jednostavno i intuitivno za korištenje raznim dobnim skupinama te na raznim uređajima. Ono se dobro skalira između zaslona različitih razlučivosti uključujući tablete i mobilne uređaje.

Sve je osmišljeno kako bi bilo dobro vidljivo te „nadohvat ruke“ korisniku. Najlakše područje koje korisnik može pritisnuti na uređaju je njegova sredina. Zato se sve bitno uvijek dešava u sredini. Dosta teški za stisnuti su kutovi zaslona zato što korisnik mora rastegnuti ruku kako bi mogao pritisnuti gornji kut ili stisnuti kako bi pritisnuo donji. U kutovima se, zato, nalazi ono što će korisnik stiskati manje puta: tipke za izlazak, opcije, te dodatne mogućnosti pojedinih prikaza. Sve ovo doprinosi kvaliteti korisničkog iskustva – UX (User eXperience).

Raspored u opcijama je isto tako promišljeno raspoređen. U odjeljku za više postavka određene postavke podijeljene su u polja sukladno sa njihovom svrhom.

3.5. Online komunikacija

Sva se mrežna komunikacija izvršava preko Google Firebase API-ja te preko HTTP protokola za komunikaciju sa serverom. Firebase je mobilni alat razvijen od strane Googlea namijenjen brzom razvoju aplikacija visoke kvalitete. Dakle, sva je komunikacija podržana sa strane veoma pouzdanih Google servera od kojih možemo očekivati neprestan i besprijekoran rad.

3.5.1. Prijava korisnika

Korisnik se u aplikaciji prijavljuje sa e-mail adresom pribavljenom od strane Cornuxa s obzirom da je Cornux za sada zatvorena aplikacija. Korisnik će pomoću svoje Firebase e-mail adrese komunicirati sa Python web serverom za obradu narudžbenica.

3.5.2. Online baza podataka

Kao što sam već prije spominjao Firebase je u ovoj aplikaciji povezan na vrlo visokoj razini sa osobnim web serverom koji koristimo za obradu narudžbenica. S obzirom da aplikacija mora pristupati serveru, aplikacija mora imati internetsku vezu da bi radila.

Web server ima svoju bazu podataka napravljenu u CSV formatu kako bi se mogla ručno mijenjati pomoću Microsoft Excela ili potpuno automatski pomoću Python programskog jezika.

3.5.3. Sigurnost

Sva online komunikacija odvija se strogo po zadanim pravilima napisanim u programskom rješenju aplikacije. Firebase je povezan sa aplikacijom pomoću SHA-1 (Secure Hash Algorithm) ključa. SHA-1 je algoritam za provjeru autentičnosti datoteke, poruke ili, u ovom slučaju, aplikacije. Na ovaj sam način osigurao da samo ova aplikacija može komunicirati s tim posebno izrađenim firebase projektom. Kao što sam već spomenuo, korisnik u aplikaciji ima pristup samo njegovim podacima. Na taj je način ovaj sustav zatvoren i osiguran od vanjskih prijetnji.

Komunikacija je također osigurana raznim postupcima kao što je ograničavanje prijava korisnika. Točnije, više od 100 prijava sa jedne IP adrese u jednom satu neće biti moguće. Na ovaj sam se način zaštitio od zlonamjernih korisnika. Velik broj prijava mogao bi više opteretiti server te umanjiti brzinu prijave drugim korisnicima.

3.6. Proces izrade aplikacije

Proces izrade složen je proces u kojemu se nalaze mnogi koraci pri razvoju aplikacije. Kroz samu izradu aplikacije prošao sam kroz razne pogreške u razvoju koje sam putem ispravljao kako bi aplikacija bila funkcionalna, a kod kvalitetan. Za dosadašnji razvoj aplikacije, a pogotovo web servera trebalo mi je preko godinu dana.

Proces razvoja može varirati od programera do programera. U nastavku su glavni koraci koje sam ja prošao prilikom izrade ovog projekta.

3.6.1. Ideja

Svaki projekt započinje sa idejom. Prije početka razvoja aplikacije dobro sam razradio ideju aplikacije te sam istražio sve moguće tehnologije koje su mi se činile dobre za izradu ove aplikacije. Osim toga, provjerio sam postoje li već slične aplikacije na tržištu te ako i postoje, koje probleme mogu riješiti svojom aplikacijom.

3.6.2. Razvoj metode za obradu narudžbenica

Kako na tržištu nije postojalo ni jedno rješenje ove prirode, bilo je potrebno osmisliti potpuno nov način kako uzimati slike različitih formata i izvlačiti iz njih različite podatke, a da to bude veoma precizno i pouzdano. Nakon više mjesecnog istraživanja, odlučio sam se da ću koristiti Python programski jezik zbog njegove razvijenosti u području umjetne inteligencije i obrade slika.

3.6.3. Stvaranje web servera

Nakon stvaranja osnovne funkcionalnosti ovog rješenja bilo je potrebno naći način kako ga i upotrijebiti. S obzirom da je to veoma zahtjevan proces za računalno sklopovlje, bilo bi previše varijacija među uređajima kada bi ta funkcija bila ugrađena u mobilnu aplikaciju. Drugi način je bio napraviti web server koji će raditi 24 sata te će sada performanse ovisiti samo o dobroj internetskoj vezi.

3.6.4. Dizajn korisničkog sučelja

Nakon što sam osmislio kako će aplikacija raditi te koje će sve značajke imati, počeo sam osmišljavati grafički izgled korisničkog sučelja. Ovaj korak u procesu potreban je kako bi se mogla predložiti sama funkcionalnost aplikacije.

3.6.5. Prototip

Nakon što sam osmislio sve potrebno krenuo sam sa izgradnjom prototipa. U tom procesu najviše sam se fokusirao na sam izgled aplikacije koji sam sa vremenom dorađivao. Prototip mi je poslužio kako bih mogao predočiti kako će završni proizvod izgledati te kako bih mogao razviti detaljniji plan izrade aplikacije.

3.6.6. Beta verzija

U razvoju softvera, beta verzija je naziv za prvu funkcionalnu verziju koja nije dostupna svim korisnicima. Ovu verziju sam dao ograničenom broju korisnika na testiranje. Cilj toga bio je skupljanje povratnih informacija, ideja te nalaženje grešaka (bugova).

3.6.7. Prva javna verzija

Prva javna verzija aplikacije sadržavala je samo najbitnije funkcionalnosti aplikacije te samo ono što je korisniku uistinu potrebno. Ova verzija još uvijek sadrži razne bugove koji će se rješavati pri izbacivanju svake nove verzije. Ovom verzijom aplikacije pristupio sam na županijsko natjecanje.

3.7. Održavanje softvera

Svaka se verzija sastoji od više podverzija koje objavljujem na Trgovini play kako bih ispravio neke greške. Održavanje softvera ključan je dio njegova razvoja. Njegova svrha je ispravljanje grešaka te dodavanje novih značajki koje postanu potrebne sa vremenom te dolaskom novih tehnologija na tržište.

4. Marketing

Osobno, mislim da je cilj izrade dobrog softvera izraditi čim bolje programsko rješenje, po čim pristupačnijoj cijeni.

Cornux je prvo programsko rješenje ove prirode na tržištu čime ostavlja meni, kao proizvođaču aplikacije na volju koju cijenu postaviti za aplikaciju. Pošto je ova aplikacija namijenjena za ugostitelje i obrtnike, aplikacija će biti plaćana na godišnjoj razini kao i većina ostalih poslovnih programa u ugostiteljstvu. Kao i mnoga druga programska rješenja, aplikacija će imati 3 ponuđene razine plaćanja koje će se podudarati sa mjesečnim brojem obrađenih narudžbenica.

Kako korisnik već plaća tantijeme za aplikaciju i cijeli programski paket, aplikacija mora imati čim jednostavnije i intuitivnije korisničko iskustvo koje se neće prekidati nikakvim reklamama ili oglasima. Reklame ću možda integrirati kada smislim kako to napraviti, a da ne smanjim kvalitetu korisničkog iskustva. To je jako osjetljiva tema koju treba osmisliti tako da bi krajnji korisnik imao dobro iskustvo. Zato, što se toga tiče, postupam jako oprezno.

5. Zaključak

Razvoj Cornuxa je za mene kao prvo bio veliki proces učenja te još uvijek je. Prije razvoja ove aplikacije nisam se previše bavio jezicima visoke razine, no ovo mi je znatno pomoglo u savladavanju Python-a te poznavanja koncepata umjetne inteligencije.

Nakon rada na ovako složenom projektu smatram da sam poboljšao svoje znanje u radu aplikacija za Android. Upravo zbog toga vjerujem da ću moći nastaviti ažurirati ovu aplikaciju s novim mogućnostima, a prema željama korisnika.

Drago mi je što sam ujedinio dva znanja iz dvoje velikih područja, informatike i ugostiteljstva, te sam u procesu stekao još puno znanja u oba područja.

Na kraju bi htio zahvaliti svima koji su mi pomogli u dosadašnjem razvoju i izradi ovog projekta. Također želim zahvaliti mentorici Sanji Sochor za pomoć u rješavanju nedoumica i pripremi za natjecanje.

