Razvoj fiskalne blagajne u web frameworku Django

Anić, Robert

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Pula / Sveučilište Jurja Dobrile u Puli**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:137:836173

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Digital Repository Juraj Dobrila University of Pula

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

ROBERT ANIĆ

RAZVOJ FISKALNE BLAGAJNE U WEB FRAMEWORKU DJANGO

Diplomski rad

Pula, rujan, 2020.

Sveučilište Jurja Dobrile u Puli
Fakultet informatike

ROBERT ANIĆ

RAZVOJ FISKALNE BLAGAJNE U WEB FRAMEWORKU DJANGO

Diplomski rad

JMBAG: 0303031801, izvanredan student

Studijski smjer: diplomski sveučilišni studij Informatika

Znanstveno područje: Društvene znanosti
Znanstveno polje: informacijske i komunikacijske znanosti
Znanstvena grana: informacijski sustavi i informatologija
Predmet: Suvremene tehnike programiranja

Mentor: doc. dr. sc. Siniša Sovilj

Komentor: doc. dr. sc. Nikola Tanković

Pula, rujan, 2020.

IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Robert Anić, kandidat za magistra informatike ovime izjavljujem da je ovaj Diplomski rad rezultat isključivo mojega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Diplomskog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

_____

U Puli, _____, _____ godine

IZJAVA

o korištenju autorskog djela

Ja, Robert Anić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj diplomski rad pod nazivom „Razvoj fiskalne blagajne u web frameworku Django“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

_____

TABLICA SADRŽAJA

1. UVOD 1
2. WEB TEHNOLOGIJE I ALATI 2
2.1. HTML 2
2.2. CSS 3
2.3. PostgreSQL 4
2.4. Python 4
2.5. Javascript 6
2.6. Django Framework 8
2.7. Pycharm 11
3. LINUX OPERACIJSKI SUSTAV 12
4. ANALIZA TRŽIŠTA 14
5. RAZRADA FUNKCIONALNOSTI 16
6. IMPLEMENTACIJA 20
6.1. Model baze podataka 20
6.2. Podešavanje postavki servera 22
6.3. UML dijagram sustava 25
7. APLIKACIJA FISKALNE BLAGAJNE 27
8. CERTIFIKAT 38
9. ZAKLJUČAK 41
SAŽETAK 42
SUMMARY 43
LITERATURA 44
SLIKE 45
TABLICE 45
KOD 46

1. UVOD

Aplikacija Fiskali je web fiskalna blagajna koju mogu koristiti svi poduzetnici koji imaju obvezu izdavati račune, bez obzira na to jesu li obveznici fiskalizacije. Aplikacija je razvijena u programskom jeziku Python pomoću web frameworka Django. Ciljano tržište ove aplikacije jesu obrti i tvrtke koji ne izdaju veliki broj računa dnevno kao što su kafići i restorani, već manji broj računa kao npr. odvjetnici, arhitekti, geodeti, mehaničari, informatičari, kozmetičari itd. Naravno da se aplikacija može koristiti i u slučajevima velikih brojeva dnevnih računa, ali to nije ciljano tržište. Glavna ideja ove aplikacije je pristup s bilo kojeg uređaja koje ima pristup internetu. Tako se izbjegava vezanost korisnika uz jedan uređaj ili prostor. Korištenjem ove aplikacije korisnik koji je voditelj ima centralizirani pregled i upravljanjem svojim tvrtkama, poslovnicama, naplatnim uređajima, artiklima, izdanim računima i radnicima.

Glavni administrator aplikacije ima svoje administratorsko sučelje gdje ima pristup podacima svih korisnika aplikacije te u slučaju problema u radu sustava može lakše intervenirati.

Cilj ovog diplomskog rada je opisati princip rada sustava fiskalne blagajne, analizirati tehnologije u kojem je rađena i objasniti na koji način je izrađena.

U drugom poglavlju opisane su web tehnologije i alati korišteni u razvoju ove aplikacije.
U trećem poglavlju je opisan je Linux operacijski sustav na kojem se projekt razvijao.
U četvrtom poglavlju je napravljena SWOT analiza aplikacije i opisano je ciljano tržište.
U petom poglavlju objašnjena je struktura aplikacije kroz dijagrame. Napravljeni su klasni dijagram, dijagram sekvence i Use Case dijagram.
U šestom poglavlju opisan je model baze podataka, objašnjen je UML dijagram i opisano je postavljanje aplikacije na Linux server.
U sedmom poglavlju opisan je rad sustava kroz snimke ekrana i primjere.
U osmom poglavlju opisan je postupak preuzimanja certifikata za fiskalizaciju od FINE.

2. WEB TEHNOLOGIJE I ALATI

Prilikom izrade ovog projekta korištene su sljedeće web tehnologije:

- HTML,
- CSS,
- PostgreSQL baza podataka,
- Python,
- Javascript,
- Django.

Od alata korišteni su Pycharm kao razvojno sučelje i Git kao program za verzioniranje koda.

2.1. HTML

HTML je kod kojeg web preglednici, kao što su Chrome, Firefox ili Safari, koriste za prikazivanje web-stranice korisnicima. HTML je kratica za "HyperText Markup Language". HyperText znači da je to vrsta teksta koji podržava hiper-veze između stranica. Markup znači da smo uzeli dokument i označili ga s kodom da bismo, u ovom slučaju, rekli pregledniku kako interpretirati stranicu. HTML kod je izgrađen s oznakama, od kojih svaka počinje s „<“ i završava s „>“. Te oznake predstavljaju elemente označavanja. Na slici 1. prikazan je primjer HTML koda korištenog u aplikaciji.

```
<div class="panel panel-default">
    <div class="panel-heading">
        Popis artikala
    </div>
    <div class="panel-body">
        <div class="table-responsive">
            <table class="table table-striped">
                <thead>
                    <tr>
                        <th>Naziv</th>
                        <th>Prodajna cijena</th>
                        <th>Jedinica količine</th>
                        <th>PDV</th>
                        <th>PNP ({{ pnp }}%)</th>
                        <th>Uredi</th>
                    </tr>
                    </thead>
                    <tbody>
                    {% for artikl in object_list %}
                        <tr>
                            <td>{{ artikl.naziv }}</td>
                            <td>{{ artikl.maloprodajna_cijena }} kn</td>
                            <td>{{ artikl.get_jedinica_display }}</td>
                            <td>{{ artikl.pdv }}</td>
                                {% if artikl.pnp %}
                                    <td><i class="fa fa-check green"></i></td>
                                {% else %}
                                    <td><i class="fa fa-times red"></i></td>
                                {% endif %}
                            <td>
                                <a href="{% url 'artikli_izmijeni' pk=artikl.id %}" class="btn btn-primary"><i class="fa fa-edit "></i></a>
                            </td>
                        </tr>
                    {% endfor %}
                </tbody>
            </table>
```

Slika 1. Primjer HTML koda

2.2. CSS

CSS je jezik koji se koristi za opisivanje izgleda i oblikovanja web-stranice napisane za označeni jezik kao što je HTML. CSS je temeljna tehnologija World Wide Weba, zajedno s HTML-om i JavaScriptom. CSS je osmišljen kako bi omogućio odvajanje prezentacije i sadržaja, uključujući izgled, boje i fontove. Ovo odvajanje može poboljšati pristupačnost sadržaja, omogućiti veću fleksibilnost i kontrolu u specifikaciji karakteristika prezentacije, omogućiti višestrukim web stranicama dijeljenje oblikovanja navođenjem relevantnog CSS-a u odvojenoj .css datoteci.

2.3. PostgreSQL

PostgreSQL je objektno relacijski sustav otvorenog koda koji služi za upravljanje podacima. Može raditi na velikoj većini današnjih operacijskih sustava kao što su Windows, macOS i Linux. Ima dobre performanse u slučaju više istodobnih korisnika koji koriste bazu. Neka opća ograničenja PostgreSQL nalaze se u tablici 1.

Tablica 1 . Ograničenja PostgreSQL

Ograničenje (engl. *limitation*)	**Vrijednost (engl. *value*)**
Maksimalna veličina baze podataka	Neograničeno
Veličina tablice	32 TB
Maksimalna veličina reda	1.6 TB
Maksimalna veličina polja	1 GB
Maksimalno redova po tablici	Neograničeno
Maksimalno stupaca po tablici	250 - 1600, ovisno o vrsti stupca
Maksimalno indeksiranje po tablici	Neograničeno

2.4. Python

Razvoj Python-a je počeo kasne 1989. godine. Razvio ga je Guido van Rossum u CWI centru u Nizozemskoj. Objavljen je za javnu distribuciju početkom 1991. godine. Isto kao C, C++, Java i Perl, Python je došao iz istraživačke pozadine gdje je programer imao problema s dobivanjem posla s postojećim alatima koji su se tada mogli koristiti. Objektno orijentirano, strukturno i aspektno orijentirano programiranje stilovi su dopušteni korištenjem Pythona te ova fleksibilnost čini programski jezik Python sve popularnijim. Python je interpretirani programski jezik što znači da je izveden odmah nakon čitanja izvornog koda. Blokovi koda se ne razdvajaju vitičastim zagradama već razmacima kao što se vidi na slici 2. Također iz slike 2. vidimo da u Pythonu nema deklaracije varijabli već se varijabli definiraju dodijeljivanjem neke vrijednosti. Python sam po sebi nije naročito brz što se tiče izvođenja, ali nastoji biti što čitljiviji jezik. Python ima manje

sintaktičkih izuzetaka i posebnih slučajeva kao što je slučaj u C-u ili Pascalu. Python obično koristi prazninu od 4 znaka za granicu kako bi odredio blokove umjesto vitičastih zagrada ili ključnih riječi.

Python podržava proceduralno, objektno orijentirano i funkcionalno programiranje. Iz tog razloga učenjem Pythona svladavaju se uz osnovne koncepte programiranja i paradigme različitih drugih programskih jezika. Python je interpreterski jezik. Zbog toga izvođenje koda programa zna biti višestruko sporije nego kod programa pisanih u C ili C++, gdje se kod prevodi u strojni jezik i u tom obliku prosljeđuje korisniku. U Pythonu se program prevodi u Python bytecode kojega pokreće Python virtualna mašina.

```
# string samoglasnika
vowels = 'aeiou'

ip_str = 'Koliko je samoglasnika u ovom stringu?'

ip_str = ip_str.casefold()

# dictionary gdje je svaki samoglasnik ključ s vrijednosti 0
ukupno = {}.fromkeys(vowels,0)

# zbroj samoglasnika
for char in ip_str:
    if char in ukupno:
        ukupno[char] += 1

print(ukupno)
```

Slika 2. Primjer Python koda

Python se najviše koristi na Linux operacijskom sustavu na kojem dolazi već ugrađen i instaliran, kojeg ću detaljnije opisati u sljedećem poglavlju.

2.5. Javascript

Javascript je skriptni programski jezik koji se najčešće koristi u web razvoju. Programski kod se izvršava na klijentskoj strani. Javascript daje interaktivnost web stranici jer, budući da se izvršava na klijentskoj strani, nije potrebno čekati odgovor servera. Na slici 3. prikazan je dio Javascript koda iz ovog projekta.

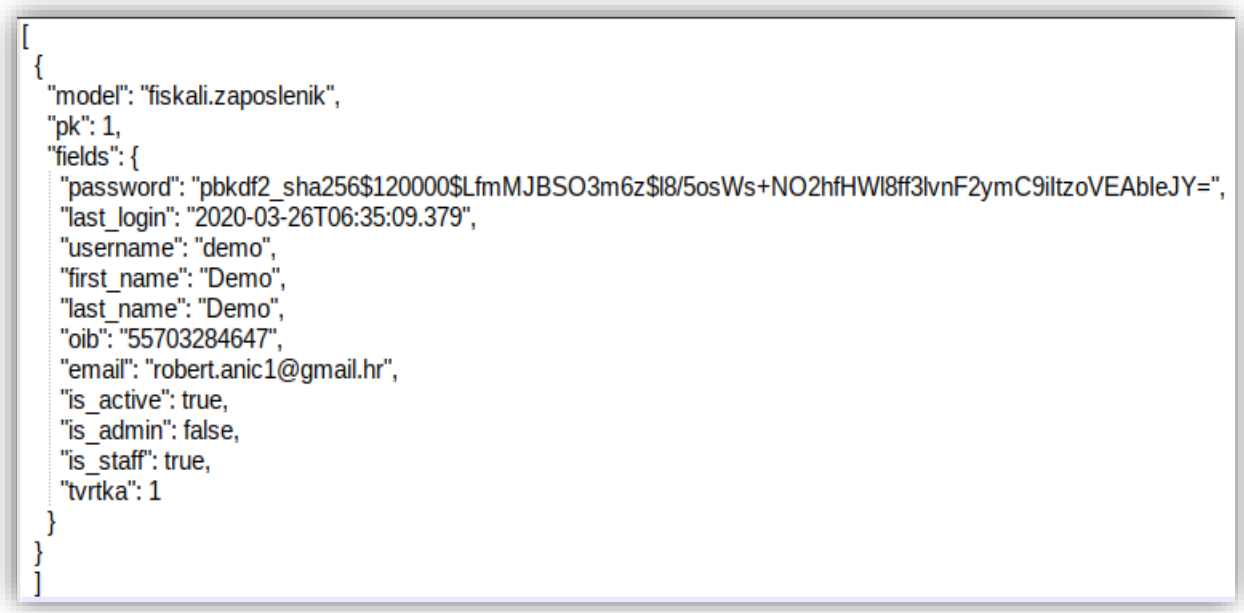
```
function ukupno() {
    if (doAjax) {
        doAjax = false;
        var csrftoken = getCookie( name: 'csrftoken');

        console.log($('#novi-racun').serialize());

        $.ajax({
            type: "POST",
            beforeSend: function (request) {
                request.setRequestHeader("X-CSRFToken", csrftoken);
            },
            url: "/ukupnoAjax",
            data: $('#novi-racun').serialize(),
            success: function (data) {
                $('#osnovica').html(data.osnovica);
                $('#porez').html(data.porez);
                $('#ukupno').html(data.ukupno);
            },
            complete: function (data) {
                doAjax = true
            }
        });
    }
}
```

Slika 3. Javascript kod

Javascript za slanje i primanje podataka koristi JSON (JavaScript Object Notation) format. JSON dolazi iz ECMAScript standarda i oblikuje podatke u standardizirani format. U ovom projektu korišten je JSON format za pohranu podataka o korisnicima, tvrtkama, poslovnicama, naplatnim uređajima i kupcima. Primjer JSON datoteke iz projekta nalazi se na slici 4.



Slika 4. JSON format

JavaScript je jezik sve veće popularnosti, a sastoji se od sintakse koja je bliska engleskom jeziku. Upotrebljava DOM model koji nudi brojne unaprijed napisane funkcionalnosti za različite objekte. JavaScript je relativno brzi jezik za krajnje korisnike jer se kod izvršava na računalu klijenta, rezultat i obrada ovisi o zadanom zadatku. Također, pruža proširenu funkcionalnost za web stranice.

2.6. Django Framework

Django je Python Web okruženje visoke razine koji potiče brzi razvoj te čist i pragmatičan dizajn. Izgrađen od strane iskusnih programera, vodi brigu o velikom broju problema vezanih uz razvoj weba. Radi se o potpuno besplatnom izvoru.

Django se može koristiti za izradu sustava za upravljanje odnosa s klijentima (CRM), sustava za upravljanje sadržajem (CMS) za unutarnju i komercijalnu upotrebu, za komunikacijske platforme, platforme za upravljanje dokumentima i sl. Između ostalog, Django je odličan za rješenja za slanje e-pošte, za sustave provjere, sustave za filtriranje s dinamički mijenjajućim pravilima i naprednim parametrima, za rješenja za analizu podataka i strojno učenje.

Radno okruženje Django je izvrstan izbor za web-razvoj, te se koristi na više od tisuću web lokacija diljem svijeta.

Prednosti Django-a kao radnog okruženja su:

- Brzina.
    - Django je dizajniran da pomogne razvojnim inženjerima da primjene aplikacije od koncepta do završetka što je brže moguće.
- Sigurnost
    - Django ozbiljno shvaća sigurnost i pomaže programerima da izbjegnu mnoge uobičajene sigurnosne pogreške.
- Skalabilnost
    - Neki od najprometnijih web-mjesta utječu na brzu i fleksibilnu skalabilnost Django-a.
- Admin sučelje
    - Admin sučelje osmišljeno je da pomogne razvojnom inženjeru upravljati aplikacijom. Njega radno okruženje Django automatski generira iz Python koda. Mnogo je prostora otvoreno za prilagodbu u admin sučelju Django-a zahvaljujući aplikacijama trećih strana. Osim toga, Django omogućuje

izmjenu sučelja i dodavanje jedinstvenih kontrolnih ploča za potrebe razvoja.

- Dodaci
    - Django se može priključiti i proširiti s brojnim dodacima. Dodaci su softverske komponente koje razvojnim programerima dopuštaju da dodaju određenu značajku aplikaciji, ostavljajući mnogo prostora za prilagodbu. Postoje stotine paketa koji pomažu dodati Google karte, stvoriti složena dopuštenja ili se povezati s uslugom za obradu plaćanja i slično.
- Velik broj biblioteka
    - Svaki programski jezik dolazi s vlastitim skupom biblioteka za rješavanje uobičajenih zadataka. Ona uključuje unaprijed napisan kod, klase, procedure, skripte, konfiguracijske podatke i drugo. U pravilu, biblioteka se dodaje programu kako bi pružila više funkcionalnosti ili automatizirala proces bez ručnog pisanja novog koda.
- Objektno relacijsko mapiranje (ORM)
    - Django je cijenjen zbog svog objektno-relacijskog mapera koji programerima pomaže u interakciji s bazama podataka. Objektno-relacijski maper (ORM) je biblioteka koja automatski prenosi podatke pohranjene u bazama podataka, kao što su PostgreSQL i MySQL, u objekte koji se obično koriste u aplikacijskom kodu. Sposobnost ORM-a da izdvoji informacije, ubrzava razvoj web aplikacija i pomaže razvojnim inženjerima da u najkraćem roku izrade prototipove. Osim toga ORM pomaže programerima u prebacivanju između relacijskih baza podataka, uz minimalne promjene koda.

Na tradicionalnoj web-lokaciji koja se temelji na podacima, web aplikacija čeka HTTP zahtjeve od web preglednika (ili drugog klijenta). Kada je zahtjev zaprimljen, aplikacija radi ono što je potrebno na temelju URL-a i eventualnih informacija u POST podacima ili

GET podacima. Ovisno o tome što se traži, aplikacija može čitati ili pisati podatke iz baze podataka ili obavljati neke druge zadatke kao bi se ispunili zahtjevi. Aplikacija će zatim vratiti odgovor web pregledniku, stvarajući dinamički HTML stranicu za prikazivanje preglednika, tako što će umetnuti dohvaćene podatke u rezervirana mjesta u HTML predlošku.

Django web aplikacije obično grupiraju kod koji obrađuje svaku od ovih komponenti u zasebne datoteke:

- URL-ovi: Iako je moguće obraditi zahtjeve iz svakog pojedinačnog URL-a preko jedne funkcije, mnogo bolji način je pisati zasebnu funkciju prikaza za obradu svakog resursa. URL mapper se koristi za preusmjeravanje HTTP zahtjeva u odgovarajući prikaz na temelju URL-a zahtjeva. URL maper također može odgovarati određenim uzorcima nizova ili znamenki koji se pojavljuju u URL-u i proslijediti ih funkciji prikaza kao podatke.

- View-ovi: Prikaz (engl. *view*) je funkcija koju definira rukovatelj zahtjeva, a prima HTTP zahtjeve i vraća HTTP odgovore. Prikazi pristupaju podacima potrebnim za zadovoljavanje zahtjeva putem modela, te raspodjeljuju oblikovanje odgovora na predloške.

- Modeli: Modeli su Python objekti koji definiraju strukturu podataka aplikacije i osiguravaju mehanizme za upravljanje (dodavanje, mijenjanje, brisanje) i zapisivanje upita u bazi podataka.

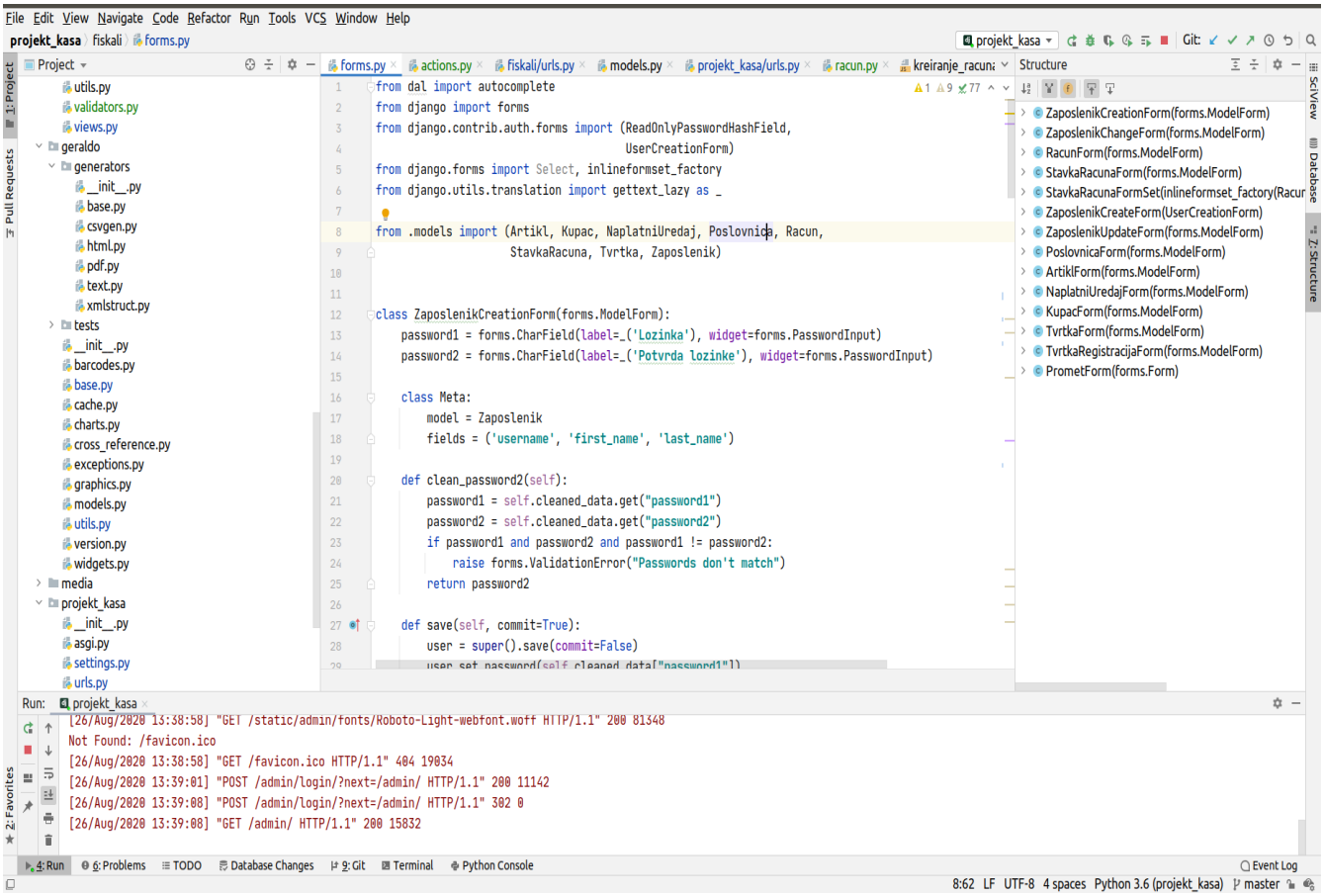
- Template-ovi: Predložak (engl. *template*) je tekstualna datoteka koja definira strukturu ili izgled datoteke kao što je HTML stranica. Prikaz može dinamički stvoriti HTML stranicu pomoću HTML predloška, koja se popunjava podacima iz modela.

U sustav je implementirana verifikacija korisnika putem aktivacijskog emaila nakon kojeg se korisnik može prijaviti u aplikaciju. Implementiran je i ispis računa na kojem su navedeni podaci o tvrtki i logotip.

2.7. Pycharm

Za razvoj aplikacije korišteno je Pycharm razvojno sučelje (IDE). Obično se koristi za programiranje u programskom jeziku Python , a olakšava razvoj i nudi brojne mogućnosti poput dopunjavanja koda, isticanje koda s greškom, preslagivanje koda, Python Debugger i integracija s sustavima za kontrolu verzioniranja. Prilagođen je za rad s web framework-cima, ne moramo napuštati sam IDE da bismo napravili promjenu na lokalnom računalu.

Na slici 5. prikazano je programsko sučelje Pycharm razvojnog sučelja. Na lijevoj strani je struktura projekta, u sredini je glavni ekran za pisanje koda, a lijevo prikaz klasa u otvorenoj datoteci radi bolje preglednosti. U donjem dijelu se nalaze tabovi za Python konzolu, Linux terminal, Git, bazu podataka i greške u radu programa.



Slika 5. Pycharm programsko sučelje

3. LINUX OPERACIJSKI SUSTAV

Na strani poslužitelja (servera), Linux je poznat kao stabilna i pouzdana platforma pružajući bazu podataka i trgovinu za tvrtke kao što su Amazon, US Post Office i mnogi drugi. Naročito su pružatelji i davatelji internetskih usluga zavoljeli Linux firewall, proxy- i web server, te je Linux kutiju (engl. *box*) moguće naći kod svakog UNIX sustava gdje administrator cijeni upravljačku stanicu.

Linux je dobio ime po izvornom autoru Linusu Torvaldsu. Prije nego što je Linus napravio novu jezgru za Intelov mikroprocesor 80386, operacijski sustavi slični Unixu nisu bili primjenjivi ni korišteni za kućnu upotrebu, već samo za istraživačke i uredske poslove. Linus Torvalds je napravio svoju jezgru (engl. *kernel*), a nakon nekog vremena je objavio izvorni kod na internetu, te je pozvao ljude da sudjeluju u daljnjem razvoju. Mnogi programeri su prihvatili poziv, tako da je Linux jezgra zajedničko djelo programera diljem svijeta. Na početku Linux programeri se se usredotočili na umrežavanje i usluge u sustavu. U posljednjih nekoliko godina Linux je postao prihvatljivim izborom kao radno okruženje, pružajući jednostavno korisničko sučelje i Office alate kao što su proračunske tablice, prezentacije i slično.

Linux kao operacijski sustav ima mnoge prednosti:

- Linux je besplatan
    - Linux se može u cijelosti besplatno preuzeti s Interneta. To podrazumijeva neobaveznu registraciju, nikakve troškove po korisniku, besplatno ažuriranje i slobodno dostupan izvorni kod.
- Linux je prenosiv na bilo koju hardversku platformu
    - Dobavljač koji želi prodati novo računalo, pri čemu ne zna koji operacijski sustav će njegovo računalo moći pokrenuti, može jednostavno uzeti Linux Kernel i pogoniti ga na svojem hardveru, jer je njegova dokumentacija besplatno dostupna.
- Linux je napravljen da neprekidno radi

- Kao i kod UNIX-a, Linux sustav očekuje da će raditi bez ponovnog pokretanja cijelo vrijeme. Zato se puno zadataka izvršava noću ili se automatski raspoređuju, što rezultira veću dostupnost tijekom intenzivnijih razdoblja i uravnoteženiju upotrebu hardvera.

- Linux je siguran i svestran
  - Nije predviđen samo za korištenje protiv internetskih napada, nego se podjednako prilagođava i drugim situacijama koristeći visoke standarde sigurnosti.

4. ANALIZA TRŽIŠTA

Na tržištu u Republici Hrvatskoj postoje različite aplikacije za fiskalizaciju računa. Neke od najvećih i najboljih aplikacija sličnih funkcionalnosti su Web Račun, Neosalon i Billy-easy. Neosalon i BillyEasy nude i mali prijenosni uređaj na kojem se mogu kreirati računi uz mogućnost trenutnog ispisa.

Ciljano tržište ove aplikacije su obrtnici i tvrtke koji imaju obvezu izdavati fiskalizirane račune. U prvom redu to su razni terenske djelatnici, odvjetnici, informatičari, vlasnici paušalnih obrta i dr. Ovakvih obrta i firmi je sve više i time se povećava tržište na koje bi plasirali ovu aplikaciju. Također se sve više diže svijest o štetnosti i kriminalnoj odgovornosti osoba koje ne izdaju fiskalizirane računa i time izbjegavaju platiti porez. Ova aplikacija se može dodatno proširiti na strance koji imaju obrte u Republici Hrvatskoj, a ne znaju hrvatski jezik dodavanjem mogućnosti odabira više stranih jezika.

Kao alat za analizu strategije poduzeća koristi se SWOT analiza (tablica 2). SWOT analiza dobila je naziv prema engleskim nazivima za snagu, slabosti, prilike i prijetnje (engl. *strengths, weakness, opportunities, threats*). Ona uključuje četiri faktora. Faktori snage i slabosti su unutarnji čimbenici poduzeća, a faktori šanse i prijetnje su čimbenici okoline poduzeća. Pomoću SWOT analize lakše procjenjujemo usklađenost poduzeća s okolinom. Cilj ovakvog načina analize je minimiziranje slabosti i prijetnji okoline i povećanje prednosti poduzeća uz što bolje iskorištavanje prilika.

Tablica 2. SWOT analiza

<table>
<tr><th>Snage</th><th>Slabosti</th><th>Prilike</th><th>Prijetnje</th></tr>
<tr><td>Web pristup</td><td rowspan="3">Potreban pristup internetu</td><td>Sve veći broj obrta i firmi</td><td>Ekonomska situacija u državi</td></tr>
<tr><td>Jednostavnost korištenja</td><td rowspan="2">Dostupnost internet veze</td><td>Moguća financijska kriza</td></tr>
<tr><td>Centralizirani pregled i upravljanje podacima</td><td>Konkurencija</td></tr>
</table>

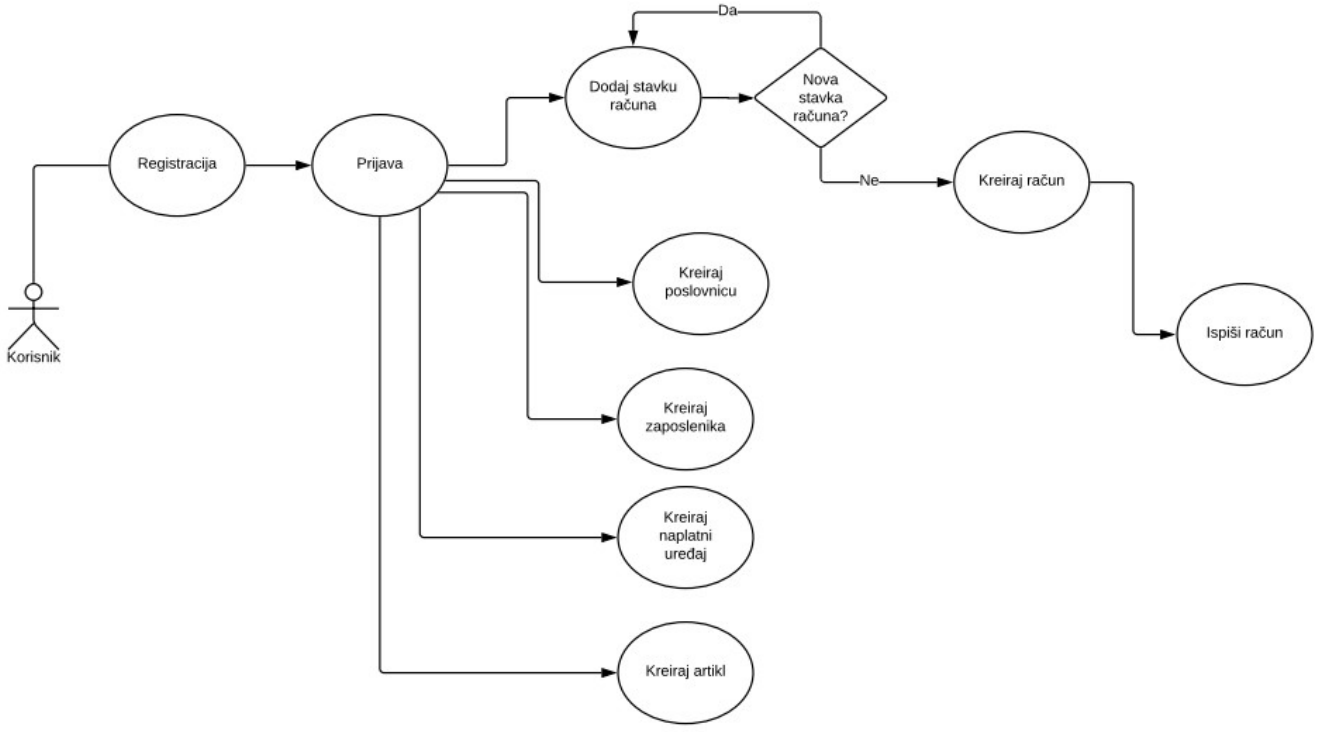
Za uvođenje aplikacije fiskalne blagajne potreban je pristup web servisu porezne uprave. Da bi pristupili tom web servisu tvrtka koja uvodi ovo programsko rješenje mora od Fine zatražiti certifikat kojim se potvrđuje vjerodostojnost poslovnog subjekta i omogućava slanje računa na fiskalizaciju. Koristi od uvođenja ove aplikacije, osim naravno korisnika, ima i porezna uprava.

5. RAZRADA FUNKCIONALNOSTI

Razlikujemo 3 vrste korisnika naše aplikacije. Korisnik može biti radnik, voditelj ili administrator. Radnik ima pravo pregledavati podatke o naplatnim uređajima koji su mu dodijeljeni, tvrtki i poslovnici u kojoj radi, izdanim računima i artiklima unutar svoje tvrtke.

Voditelj ima veća prava. On može dodavati nove radnike i dodjeljivati im naplatne uređaje i poslovnice u kojima rade. Voditelj može i pregledavati izdane račune svih zaposlenika unutar njegove tvrtke.

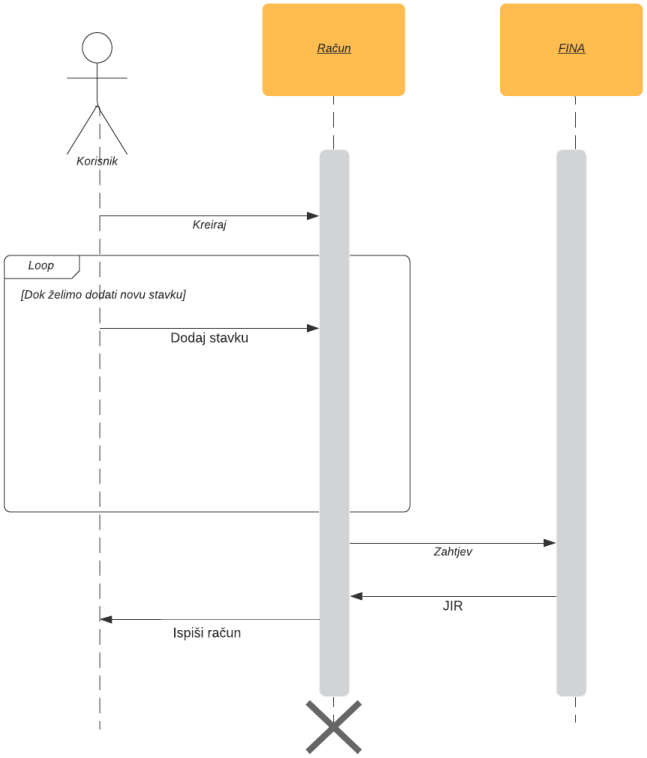
Administrator ima svoje administratorsko sučelje u kojem ima pravo na pregled i izmjenu podataka svih korisnika aplikacije



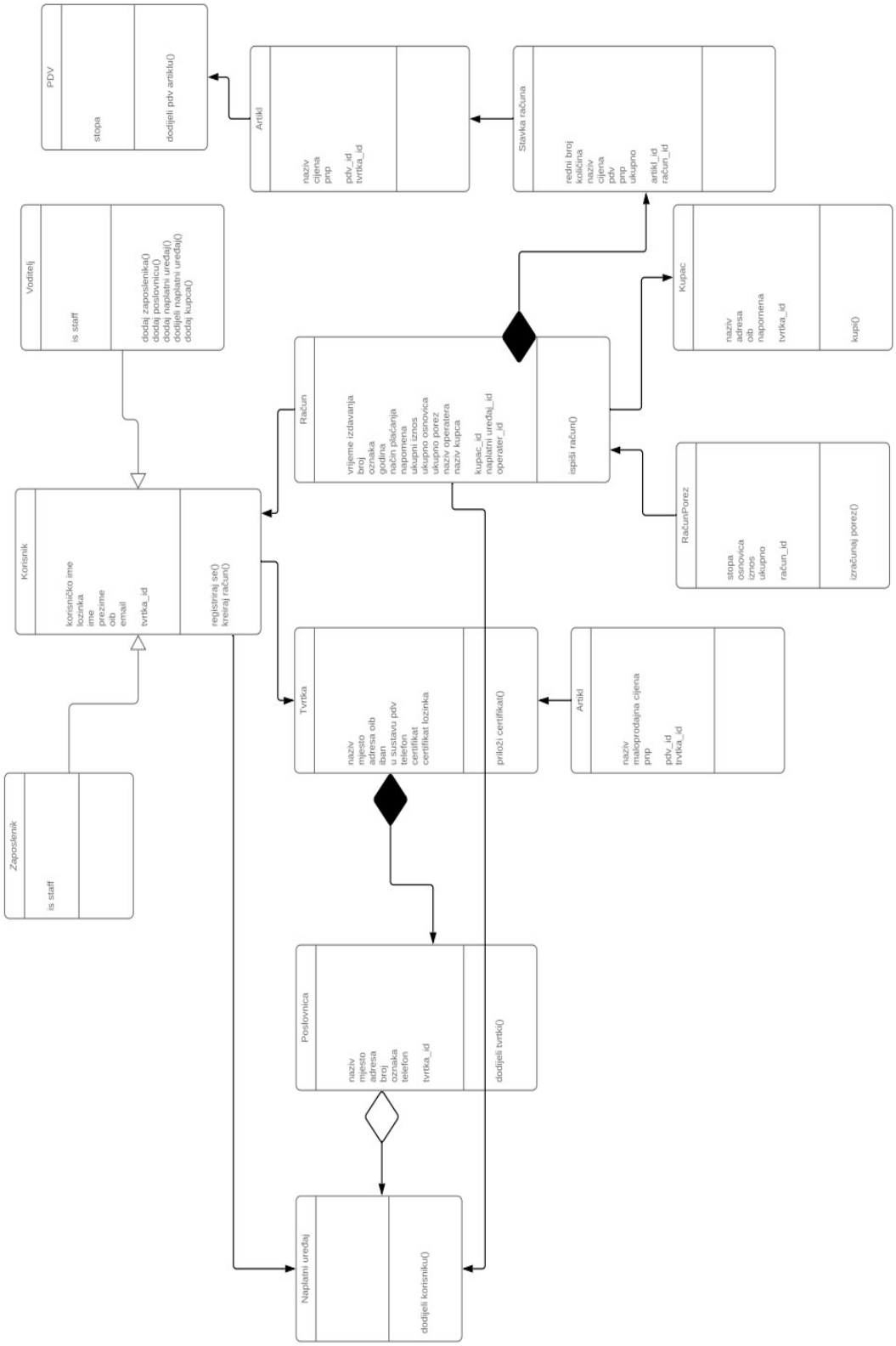
Slika 6. Use Case dijagram

Na slici 6. prikazan je Use Case dijagram sustava. Korisnik se nakon registracije prijavljuje u aplikaciju, ima pravo dodavati i mijenjati podatke o artiklima, naplatnim uređajima, zaposlenicima i poslovnicama. Račun se kreira dodavanjem stavki na račun. Nakon kreiranja računa korisnik može zatražiti ispis računa.

Na slici 7. je prikazano dodavanje stavki na račun i na kraju kreiranje i ispis računa.



Slika 7. Dijagram sekvence



Slika 8. Klasni dijagram

Na slici 8. prikazan je klasni dijagram sustava. Iz njega se može vidjeti da korisnik može biti zaposlenik ili voditelj. Voditelj ima dodatne mogućnosti, a to su dodavanje

zaposlenika, dodavanje poslovnica, dodavanja i dodjela naplatnog uređaja i dodavanje kupca. Klase Račun i Stavka računa povezane su vezom kompozicije jer stavka računa ne postoji bez računa kao i obrnuto. Na Klasu Račun asocijacijom su povezane klase Kupac i Račun Porez u koji spremamo iznos poreza na pojedinom računu. Na klasu Tvrtka vezan je artikl jer svaka tvrtka ima svoje artikle i Poslovnica. Na poslovnicu je agregacijom povezana klasa Naplatni Uređaj.

6. IMPLEMENTACIJA

6.1. Model baze podataka

Aplikacija je razvijena u programskom jeziku Python pomoću web frameworka Django.

Za ispis računa korišten je Geraldo. To je alat koji se naslanja na biblioteku ReportLab i služi za izradu pdf dokumenata.

Baza podataka koja se koristi je PostgreSQL. U bazi podataka nalaze se sljedeće tablice:

- Tvrtka

Polja: naziv, mjesto, adresa, oib, iban, u_sustavu_pdv, telefon, certifikat, certifikat_lozinka,inicijalizacija

- Poslovnica

Polja: naziv, mjesto, adresa, broj, oznaka, telefon, is_active

Strani ključevi: tvrtka

Stranim ključem smo napravili vezu između tablica Tvrtka i Poslovnica jer tvrtka može imati jednu ili više poslovnica.

- Naplatni uređaj

Polja: broj,is_active

Strani ključevi:poslovnica

- Kupac

Polja: naziv, adresa, oib, napomena,is_active

Strani ključevi:tvrtka

- Zaposlenik

Polja: password, last_login, username, first_name, last_name, oib, email, is_active, is_admin,is_staff

Strani ključevi: tvrtka

- PDV

Polja:stopa

Izdvojili smo PDV u posebnu tablicu radi lakšeg održavanja aplikacije u slučaju promjene stope poreza.

- Artikl

Polja: naziv, maloprodajna_cijena, pnp, jednica,is_active

Strani ključevi: pdv,tvrtka

- Račun

Polja: vrijeme_izdavanja, broj, oznaka, godina, nacin_placanja, zki, jir, napomena, ukupni_iznos, ukupno_osnovica, ukupno_porez, naziv_operatera, naziv_kupca,zakljucan

Strani ključevi: kupac, naplatni_uredaj, operater StavkaRacuna

Polja: rbr, kolicina, naziv_artikla, cijena_artikla, pdv_artikla, pnp_artikla, ukupno, zakljucan

Strani ključevi: artikl, racun

- Račun Porez

Polja: stopa_poreza, osnovica_poreza, iznos_poreza, ukupno

Strani ključevi: račun

Tablica Račun Porez služi da bi spremali izračunane vrijednosti poreza za evidenciju.

Django postavke za bazu podataka u datoteci settings.py prikazane su na kodu 1.:

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql_psycopg2',
        'NAME': 'fiskalna',
    }
}
```

Kod 1. Django postavke

6.2. Podešavanje postavki servera

Ova web aplikacija je postavljena na Linux serveru. Aplikacija koristi PostgreSQL bazu podataka koja se preuzima naredbom „sudo apt-get install postgresql postgresql-contrib“. U ovoj naredbi „sudo“ označava da koristimo administratorska prava na sustavu za instalaciju, apt-get je linux alat za preuzimanje paketa s kojim je instalacija globalna, a „postgresql postgresql-contrib“ su imena paketa koja instaliramo. Nakon toga potreban nam je driver koji omogućuje Djangu da radi s PostgreSQL bazom podataka. Taj driver preuzimamo naredbom:

```
(venv) r@r-HP-ProBook-4740s:~/projects/fiskalna$ pip install django psycopg2
```

Kod 2. Instalacija drivera Psycopg2

Da bi našu web aplikaciju postavili na server treba nam paket Git. To je najpopularniji svjetski alat za verzioniranje programskog koda. Smatram da je pri izradi projekta obvezno koristiti neki od alata za verzioniranje. Time smanjujemo rizik od mogućeg gubitka podataka, ali osim toga nam pruža i neke korisne funkcionalnosti, kao što su pregled povijesti rada, mogućnost da više developera radi na istom projektu u isto vrijeme i mogućnost pregleda svih promjena na kodu. Također, jako nam je koristan alat kada imamo neki projekt u produkciji gdje ne smijemo riskirati da prestane raditi. Tada napravimo dvije grane, na jednoj (master grana) imamo program koji radi, a na drugoj grani (develop grana) radimo neke promjene. Kada želimo primijeniti te promjene grane

ćemo spojiti (merge). Ako nešto pođe po zlu uvijek se možemo vratiti na neko prošlo stanje projekta (revert). Paket Git se instalira naredbom Kod 3:

```
(venv) r@r-HP-ProBook-4740s:~/projects/fiskalna$ sudo apt-get install git
```

Kod 3. Instalacija programa Git

Aplikacija se nalazi na linku https://github.com/robizd/fiskali, a preuzima se naredbom (Kod 4):

```
(venv) r@r-HP-ProBook-4740s:~/projects$ git clone https://git.pakel.hr/fiskalna_projekt.git
```

Kod 4. Preuzimanje projekta

Zatim je potrebno postaviti virtualno okruženje (virtual environment). Ono izolira postavke na razini projekta. Što znači da ako npr. Na serveru je postavljen Python 2.7, ali u našem virtualnom okruženju možemo instalirati Python 3.6, a da nam na serveru verzija Pythona i dalje ostane ista. Kreiranje i aktiviranje virtualnog okruženja izvršava se naredbama (Kod 5):

```
r@r-HP-ProBook-4740s:~/projects$ python3 -m venv myvenv
r@r-HP-ProBook-4740s:~/projects$ source myvenv/bin/activate
(myvenv) r@r-HP-ProBook-4740s:~/projects$
```

Kod 5. Kreiranje i aktiviranje virtualnog okruženja

Nakon uspješne aktivacije virtualnog okruženja instaliramo pakete potrebne za rad web aplikacije. Imena paketa smo spremili u tekstualnu datoteku requirements.txt, a pakete navedene u toj datoteci instaliramo unutar našeg virtualnog okruženja naredbom (Kod 6.):

```
(myvenv) r@r-HP-ProBook-4740s:~/projects$ pip install -r requirements.txt
```

Kod 6. Instalacija paketa iz datoteke requirements.txt

Bazu podataka stvaramo naredbom (Kod 7.), gdje je createdb naredba za stvaranje baze, a drugi argument je ime novostvorene baze podataka.

```
(myvenv) r@r-HP-ProBook-4740s:~/projects$ createdb fiskali
```

Kod 7. Stvaranje PostgreSQL baze podataka

Sada kada smo stvorili praznu bazu podataka koristimo naredbu (Kod 8.) s kojom Django stvara tablice u koje ćemo spremati podatke.

```
(myvenv) r@r-HP-ProBook-4740s:~/projects$ python manage.py makemigrations
```

Kod 8. Django migracije

Nakon stvorene baze i primjenjenih migracija podatke možemo stvoriti administratorskog korisnika naredbom (Kod 9)**:**

```
(myvenv) r@r-HP-ProBook-4740s:~/projects$ python manage.py createsuperuser
```

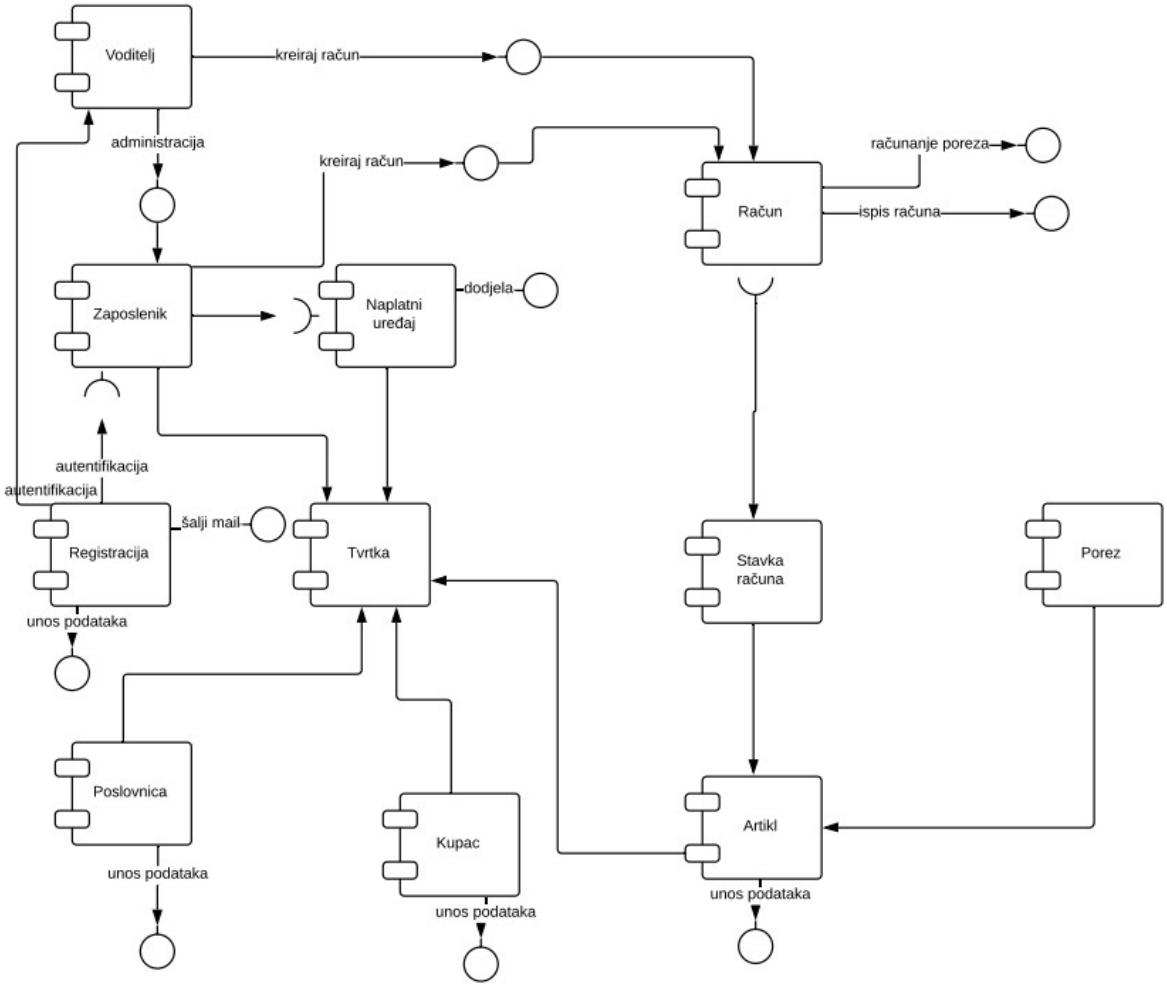
Kod 9. Stvaranje administratorskog korisnika

Za potrebe razvoja i za lakše postavljanje aplikacije u projekt je postavljena datoteka u json formatu koja sadrži korisnika, poslovnicu, naplatni uređaj, kupca i porez koju unosimo u bazu naredbom (Kod 10.).

```
myvenv) r@r-HP-ProBook-4740s:~/projects/fiskalna$ python manage.py 'kasa/fixtures/user.json'
```

Kod 10. Učitavanje podataka iz JSON datoteke

6.3. UML dijagram sustava



Slika 9. UML (engl. *Unified Modelling Languge*) Component Dijagram sustava

Na slici 9. prikazan je UML prikaz rada sustava. Korisnik se prvo registrira na sustav, nakon toga sustav šalje email s linkom za potvrdu registracije. Nakon uspješne potvrde registracije korisnik se prijavljuje u sustav. Tako stvoreni korisnik je ujedno i voditelj tvrtke te ima pravo dodavati nove zaposlenike, mijenjati njihove podatke ili ih brisati. Automatski su za korisnika stvoreni poslovnica i naplatni uređaj koji je dodijeljen korisniku. Korisnik sada mora unijeti artikle za prodaju. Nakon toga korisnik može kreirati račun dodavajući artikle koji tada postaju stavke računa. Izradom računa automatski se računa porez i

korisnik može zatražiti ispis računa u pdf obliku čime cijeli proces završava.Zakonom o fiskalizaciji je propisano da brojevi računa moraju slijediti neprekinuti niz od broja 1 unutar kalendarske godine. Bez preskočenih brojeva, po svakom poslovnom prostoru ili po naplatnom uređaju u poslovnom prostoru. Traži se od svih tvrtki da u broj računa uključe tri elementa: oznaku poslovnog prostora, oznaku naplatnog uređaja i redni broj računa. Aplikacija Fiskali prati navedene naputke. Tako npr. za oznaku računa “6/POSL1/1” broj 6 označava redni broj računa; “POSL1” označava poslovni prostor (u ovom slučaju poslovni prostor broj 1), a broj 1 na kraju označava naplatni uređaj.

7. APLIKACIJA FISKALNE BLAGAJNE

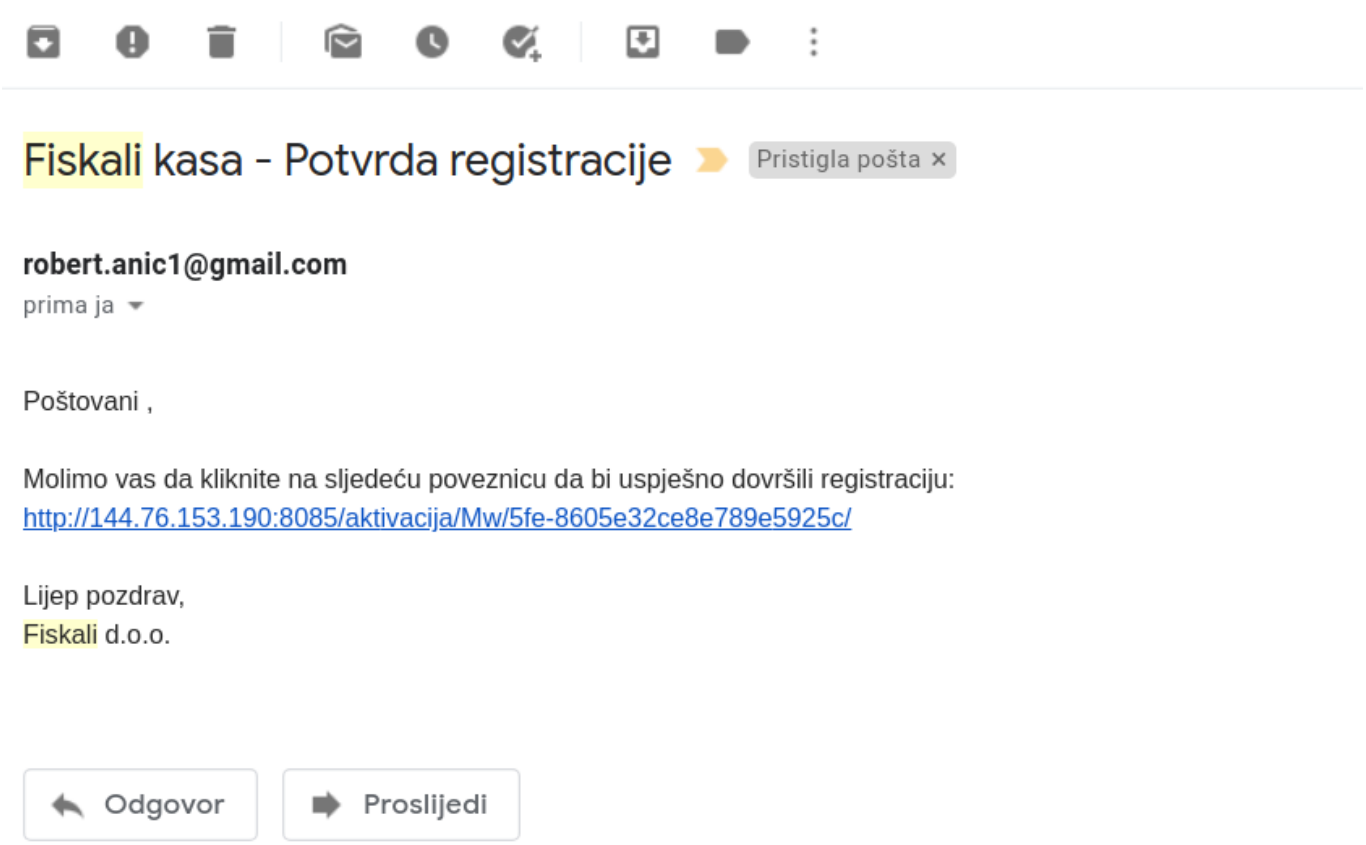
Aplikacija se nalazi na linku http://144.76.153.190:8085/ . Korisnik dolazi na početnu stranicu web aplikacije. Klikom na dugme “Registracija” otvara se web forma u koju korisnik mora unijeti podatke o sebi i tvrtki u kojoj radi. Nakon uspješno ispunjene web forme aplikacija šalje aktivacijski email kojim se potvrđuje da je korisnik unio valjane podatke o svom emailu i tek nakon što korisnik otvori poslani email i klikne na dobiveni link aplikacija verificira novog korisnika i dopušta mu prijavu.

Za slanje registracijskog emaila koristi se biblioteka django.core.mail. Funkciju iz prethodno navedene biblioteke zovemo s 3 parametra (naslov, tekst, email adresu primatelja i token za autentifikaciju). Da bi uspješno poslali email poruku potrebno je navesti par atributa u settings.py datoteku.

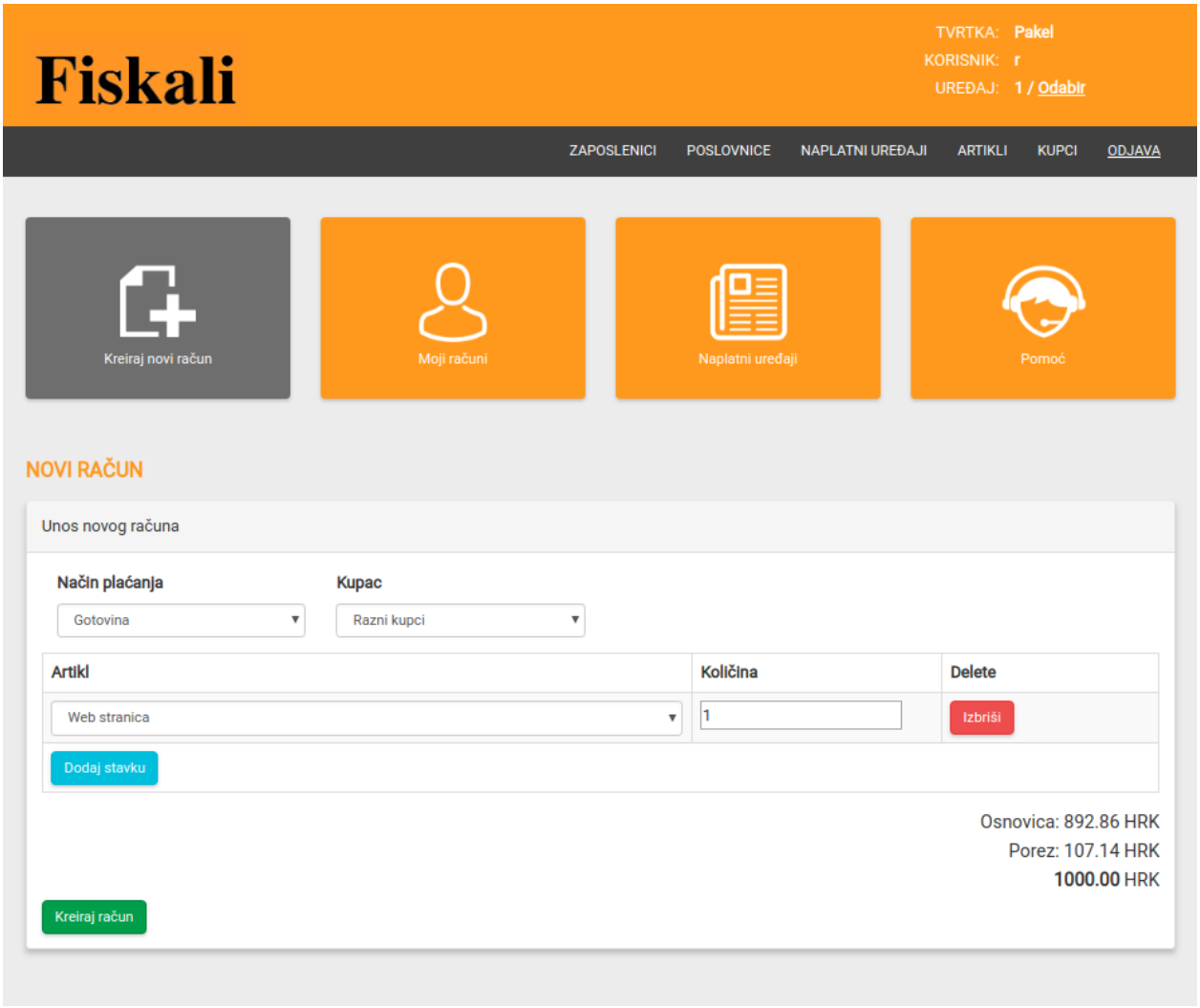
```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'webmail.unipu.hr'
EMAIL_USE_TLS = True
EMAIL_PORT = 587
```

Kod 11. Postavke za slanje email-a

EMAIL_BACKEND je Django biblioteka koju koristimo, uz nju navodimo email hosta, protokol, port, korisničko ime i lozinku.



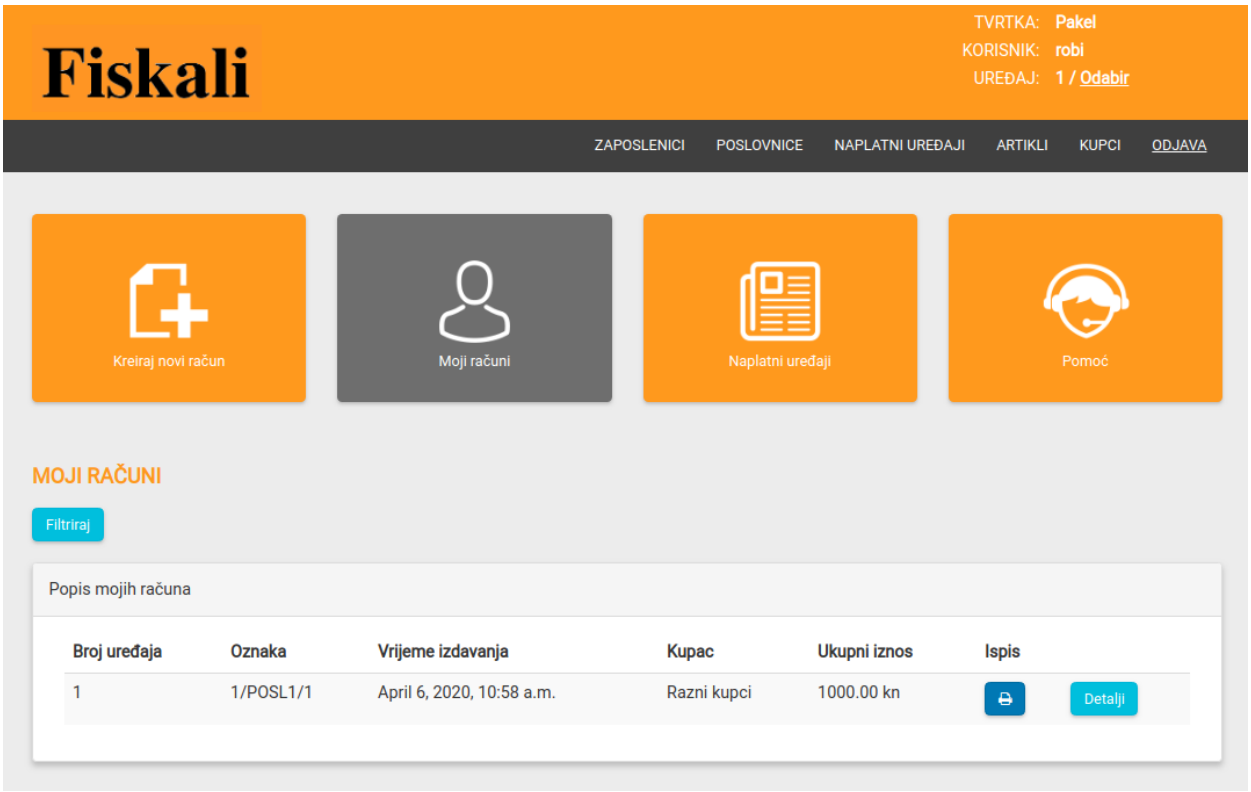
Slika 10. Verifikacijski e-mail poslan iz aplikacije



Slika 11. Kreiranje novog računa

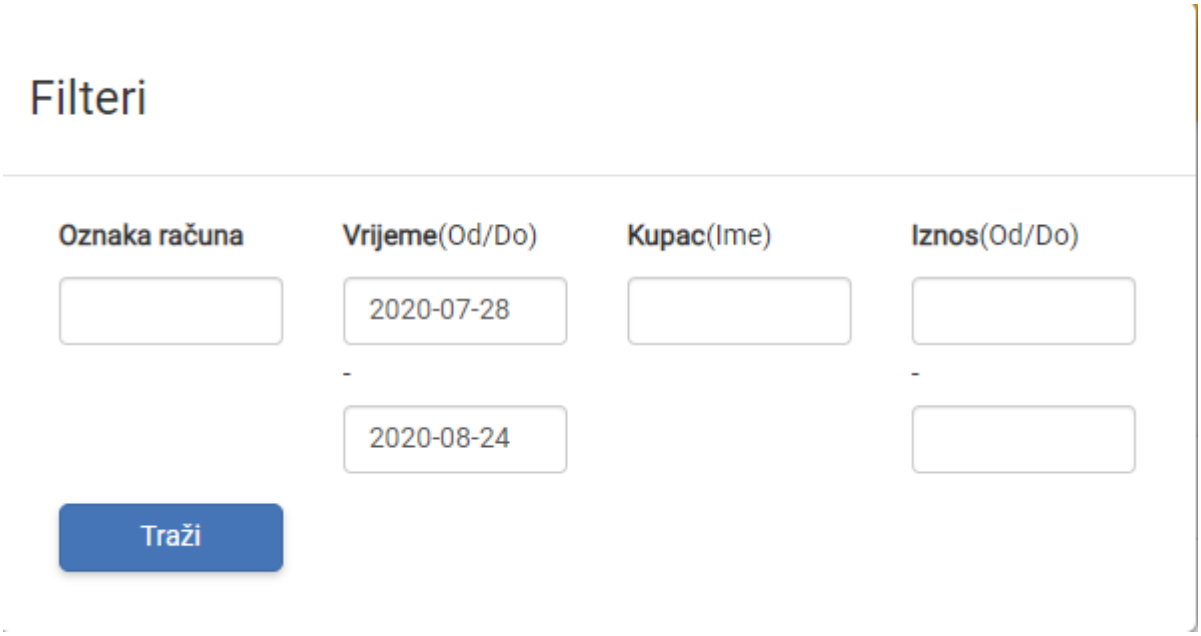
Na slici 11. prikazana je početna stranica aplikacije(ako ste prijavljeni) na kojoj ujedno i stvaramo novi račun. Odabiremo način plaćanja i kupca (ako je nepoznat odabiremo kategoriju “Razni kupci”). Zatim se dodaju artikli kao stavke računa koji su prethodno uneseni u bazu podataka i odabire se količina. Postoje opcije brisanja i dodavanja stavki računa. Kad dodamo novu ili izbrišemo staru stavku aplikacija automatski računa

osnovicu, porez i ukupni iznos koji kupac treba platiti. Račun kreiramo klikom na dugme “Kreiraj račun”. Aplikacija nas tada vodi na sliku 12.



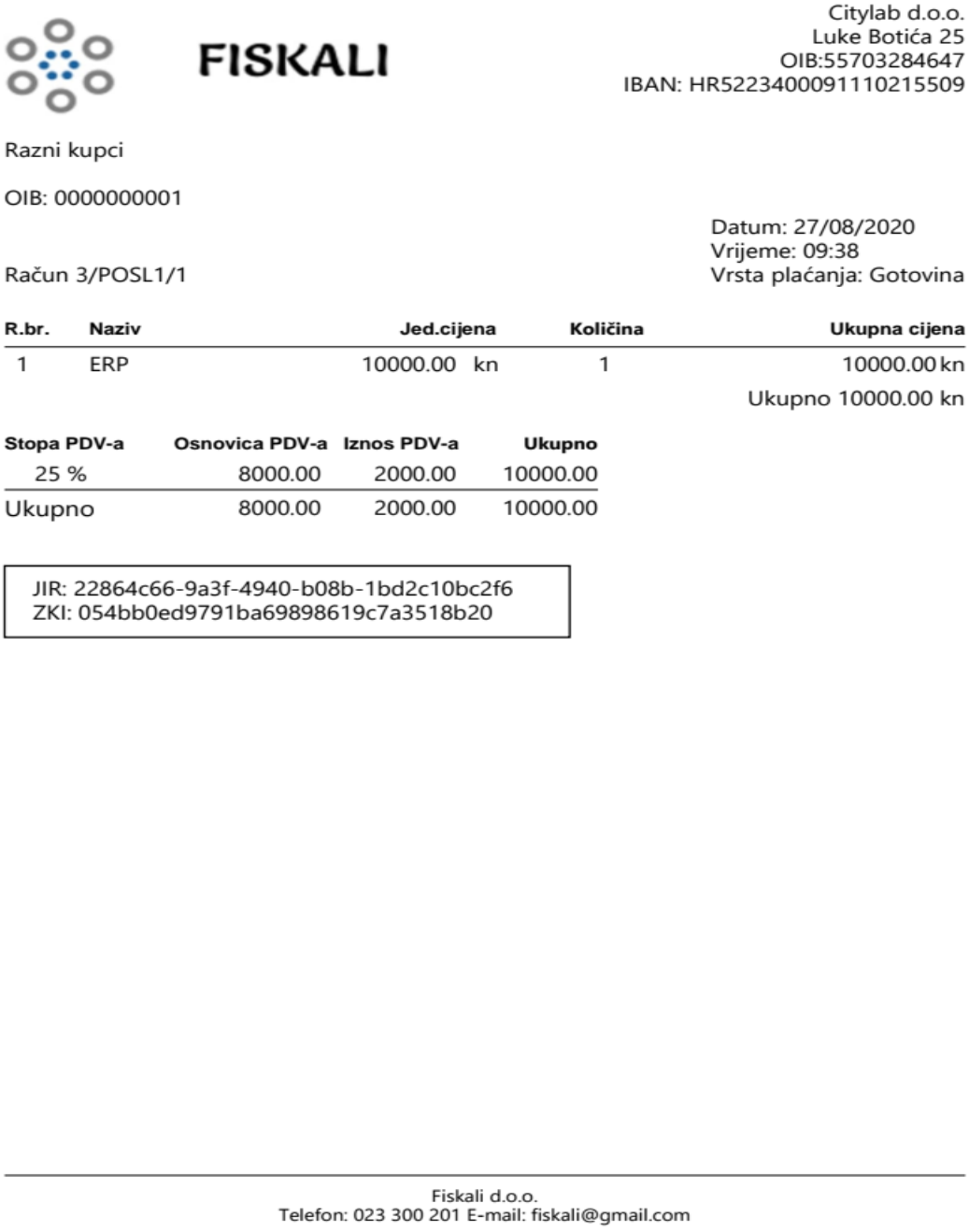
Slika 12. Popis izdanih računa

Na slici 12. nalazi se popis naših izdanih računa s osnovnim podacima. Klikom na dugme “Filteri” iskače skočni prozor gdje račune možemo filtrirati po oznaci, vremenu, kupcima i iznosu. Prikaz filtera je prikazan na slici 13.



Slika 13. Filteri izdanih računa

Filteri su napravljeni tako da je moguće filtriranje po oznaci računa, vremenu izdavanja računa, kupcima i iznosu.



FISKALI

Citylab d.o.o.
Luke Botića 25
OIB:55703284647
IBAN: HR5223400091110215509

Razni kupci

OIB: 0000000001

Datum: 27/08/2020
Vrijeme: 09:38
Račun 3/POSL1/1
Vrsta plaćanja: Gotovina

R.br.	Naziv	Jed.cijena	Količina	Ukupna cijena
1	ERP	10000.00 kn	1	10000.00 kn

Ukupno 10000.00 kn

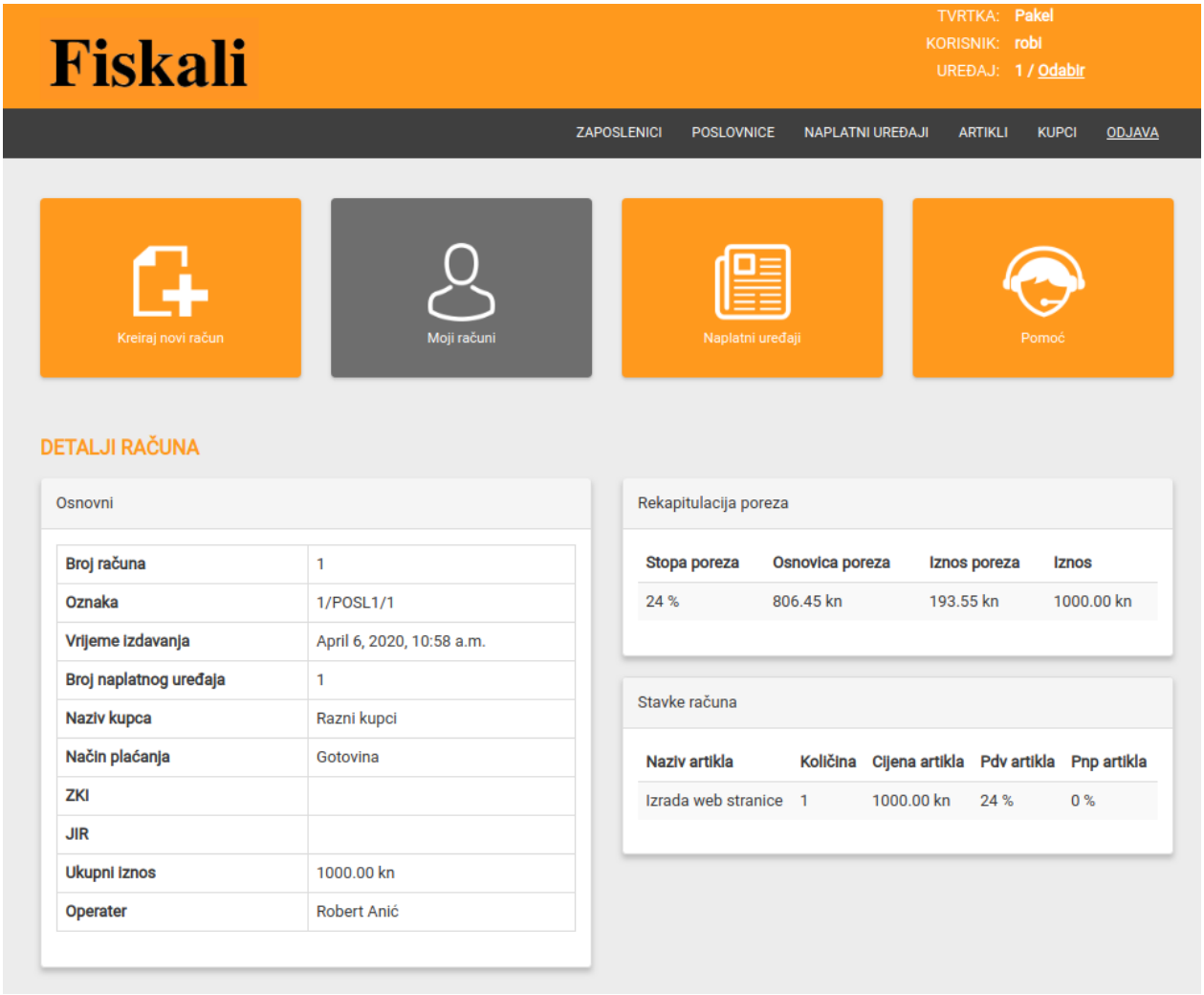
Stopa PDV-a	Osnovica PDV-a	Iznos PDV-a	Ukupno
25 %	8000.00	2000.00	10000.00
Ukupno	8000.00	2000.00	10000.00

JIR: 22864c66-9a3f-4940-b08b-1bd2c10bc2f6
ZKI: 054bb0ed9791ba69898619c7a3518b20

Fiskali d.o.o.
Telefon: 023 300 201 E-mail: fiskali@gmail.com

Slika 14. Ispis računa

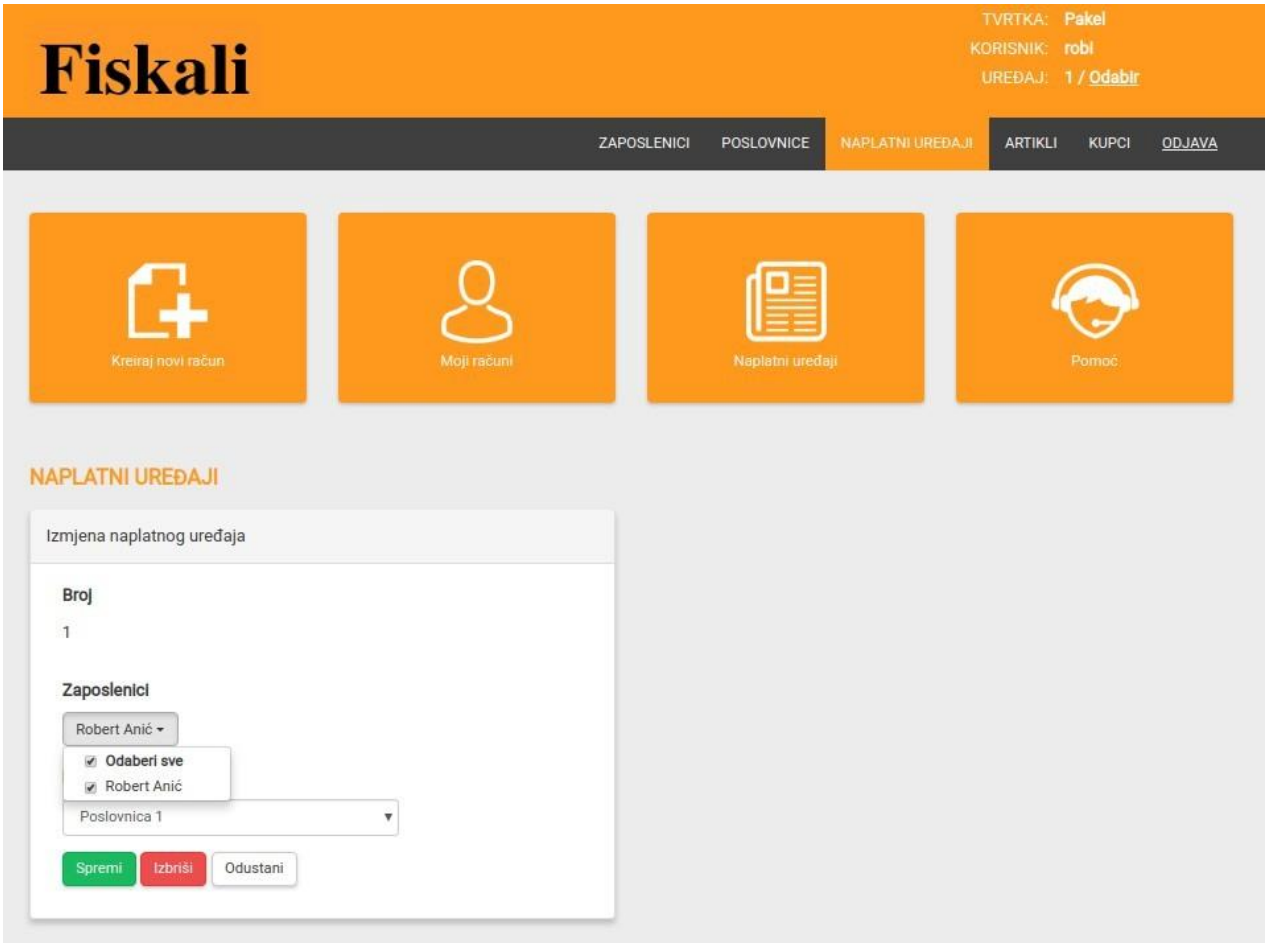
Klikom na ispis računa otvara se račun u pdf formatu koji je prikazan na slici 14. Na ispisu računa navedeni su podaci o tvrtki, stavke računa i logotip.



Slika 15. Detalji računa

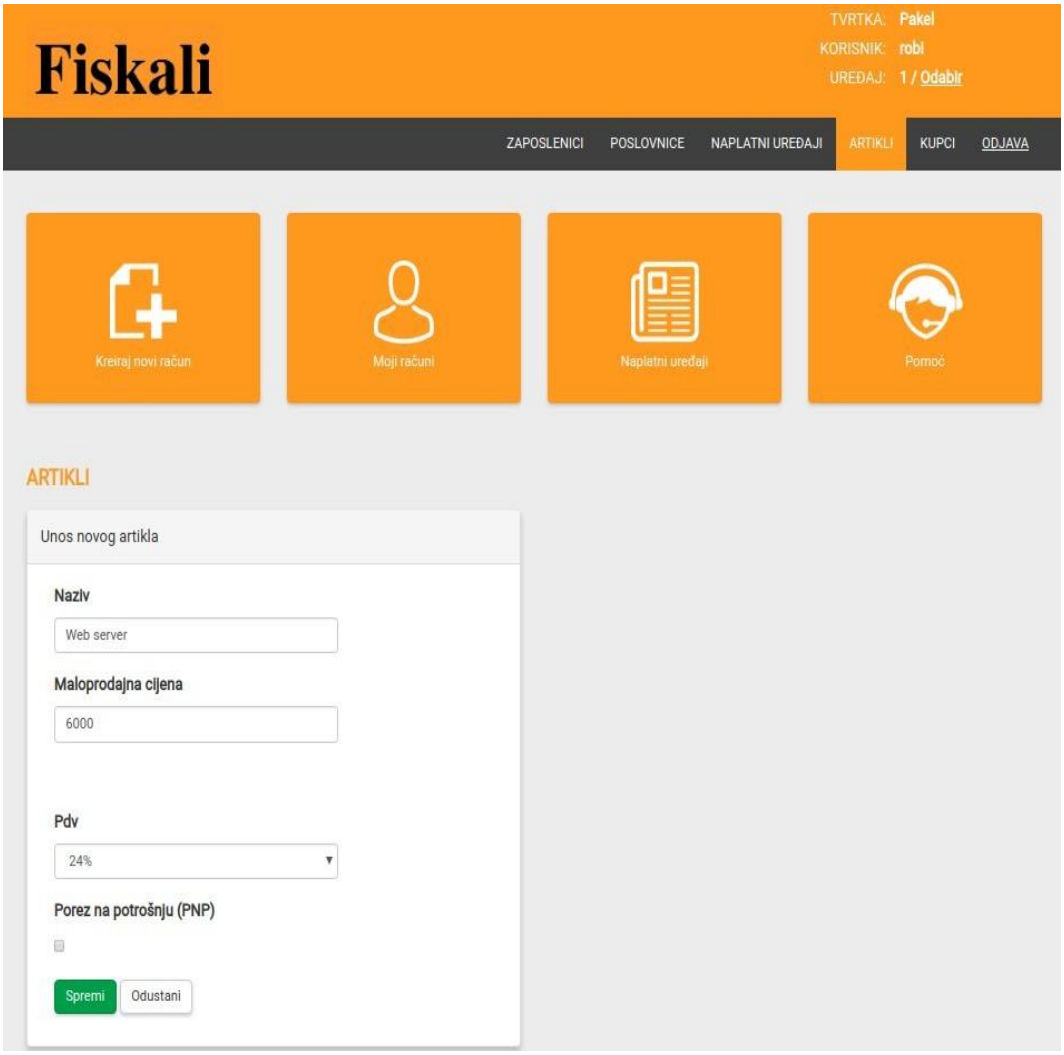
Na slici 15. prikazani su detalji jednog izdanog računa. Podaci su podijeljeni na 3 skupine:

- osnovni
- rekapitulacija poreza
- stavke računa



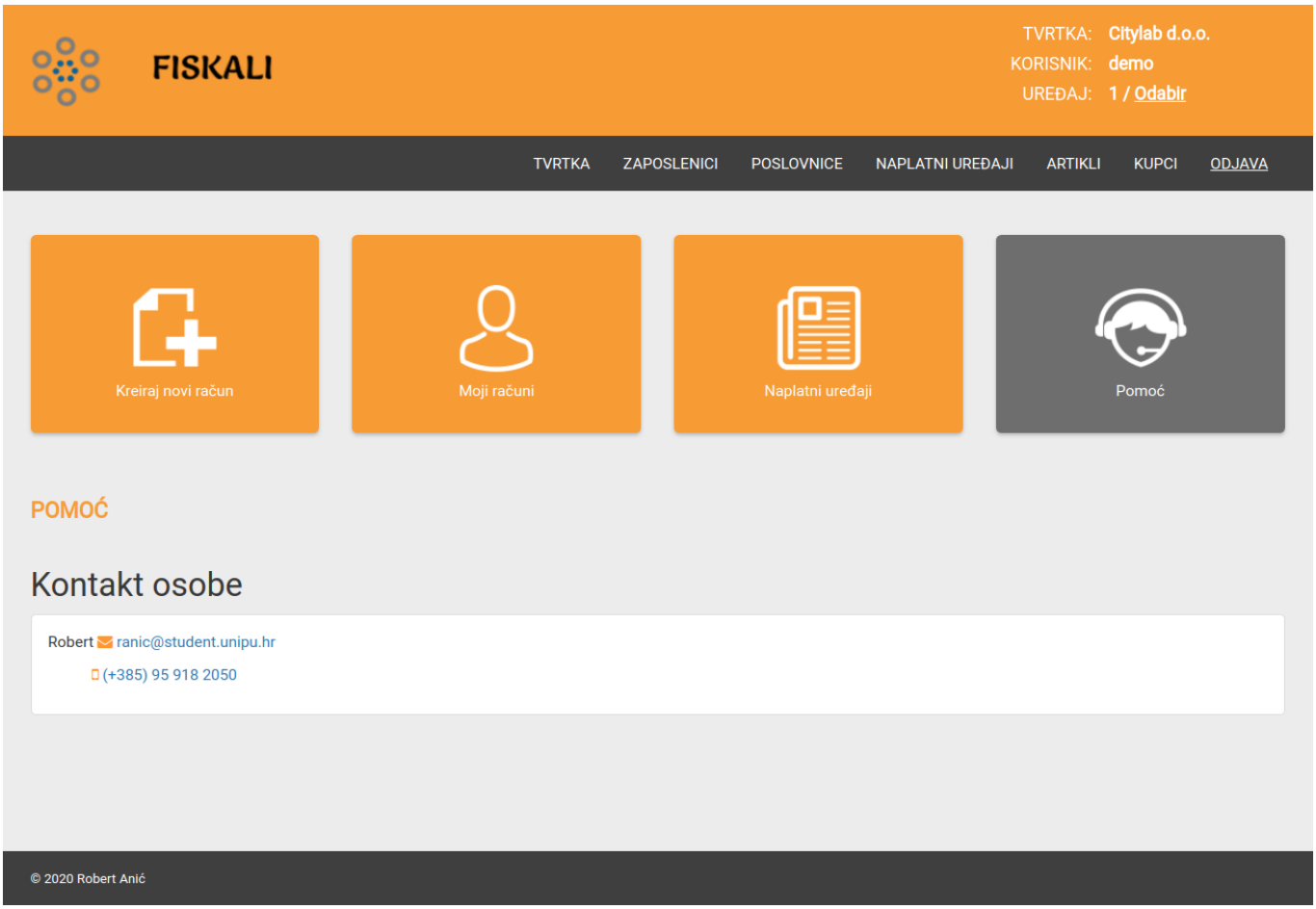
Slika 16. Dodjela naplatnog uređaja zaposleniku

Na slici 16. nalazi se dodjela naplatnog uređaja zaposleniku. Na jedan naplatni uređaj može biti vezano više zaposlenika. Naplatne uređaje možemo dodjeljivati zaposlenicima svoje tvrtke, premještati ih iz jedne poslovnice u drugu, brisati i dodavati nove uređaje.



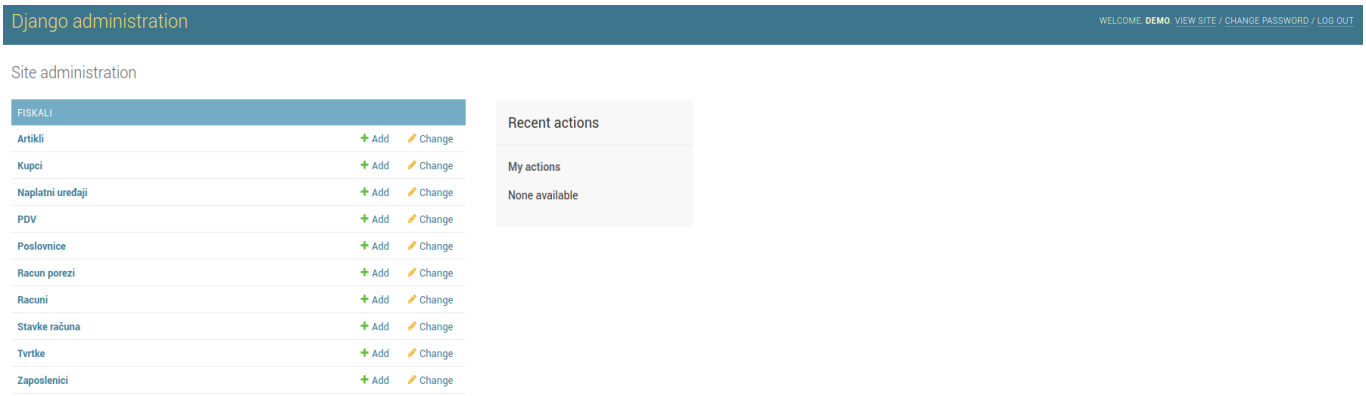
Slika 17. Unos novog artikla

Na slici 17. prikazano je dodavanje novog artikla. Klikom na dugme Izrada novog artikla otvara se forma u koju korisnik mora unijeti naziv, maloprodajnu cijenu, PDV i da li artikl podliježe porezu na potrošnju. Nakon toga artikl je unesen u bazu podataka i možemo ga vidjeti klikom na dugme artikli i možemo ga koristiti kao stavku računa u izradi novog računa.



Slika 18. Korisničko sučelje - Pomoć

Na slici 18. je prikazan ekran ”Pomoć” na kojem je naveden kontakt email i broj mobilnog uređaja podrške na koji se korisnici mogu obratiti i zatražiti pomoć pri korištenju aplikacije.



Slika 19. Administratorsko sučelje

Na slici 19. prikazano je administratorsko sučelje web aplikacije. Django izrađuje administratorsko sučelje, trebamo samo registrirati naše modele u datoteku admin.py.

```
admin.site.register(
    [
        Zaposlenik,
        ZaposlenikPrilog,
    ]
)
```

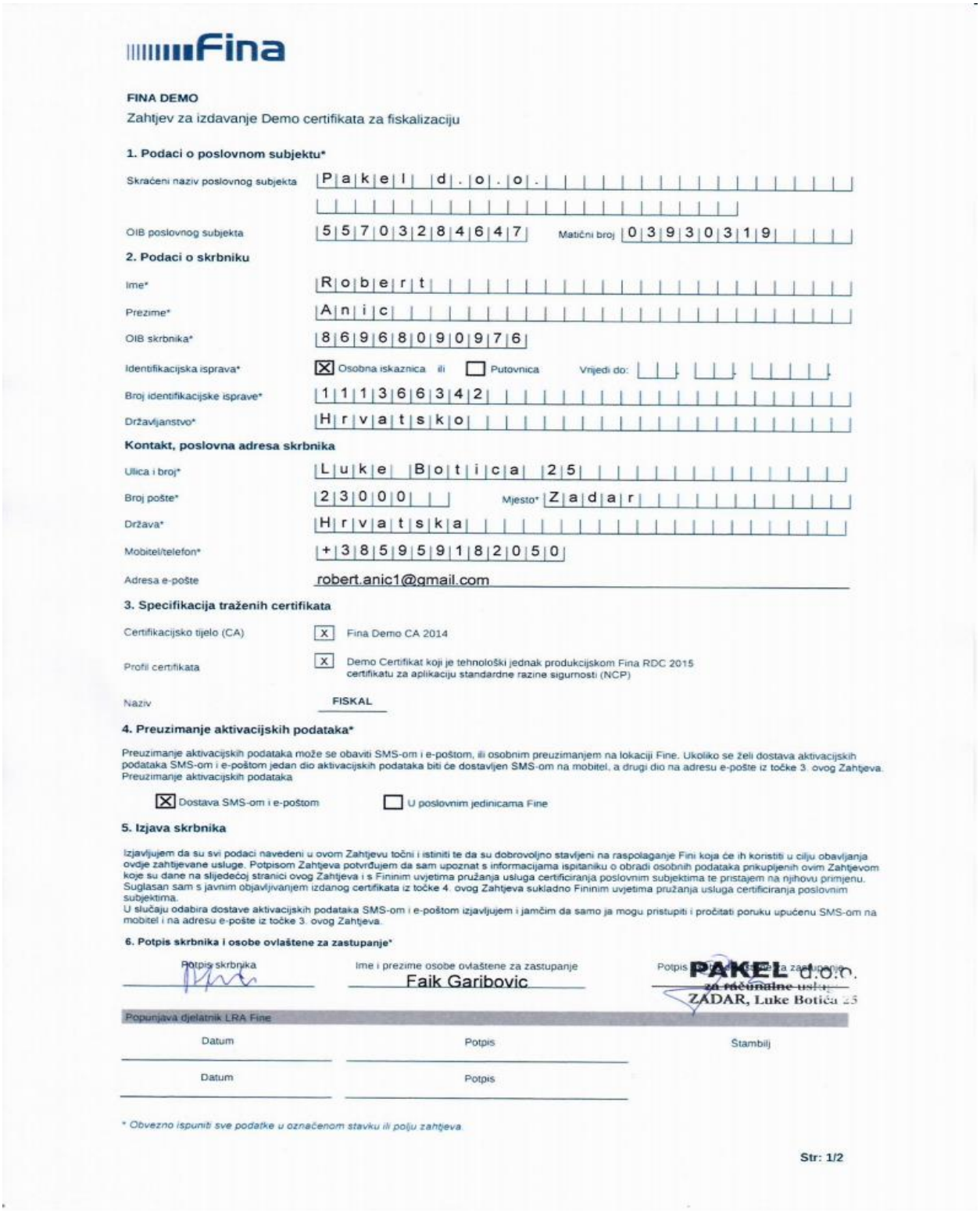
Kod 12. Registriranje modela u administratorsko sučelje

8. CERTIFIKAT

Fina izdaje aplikacijske certifikate koji su potrebni za fiskalizaciju u soft obliku. Za potrebe diplomskog rada zatražio sam od Fine Demo aplikacijski certifikat. To je testni certifikat koji se ne naplaćuje i namijenjen je informatičkim firmama koje razvijaju fiskalizacijska rješenja. Digitalni certifikat se koristi za elektroničko potpisivanje elemenata računa i za identifikaciju obveznika fiskalizacije. Postupak nabave certifikata počinje ispunjavanjem Zahtjeva za izdavanje Demo certifikata za fiskalizaciju (https://www.fina.hr/documents/52450/155573/7+Zahtjev+za+RDC_fiskalizacija+-+Demo_06092018.pdf/8c70682a-bd32-c32f-84f0-ce0441dba8ca), nakon toga ispunjen zahtjev skupa s preslikom osobne iskaznice skrbnika certifikata šaljemo na službeni email Fine (certifikati-fiskalizacija@fina.hr).

Na slici 20. je ispunjen Zahtjev za izdavanje Demo certifikata za fiskalizaciju. Na njemu su navedeni podaci o tvrtki koja šalje zahtjev i podaci o skrbniku certifikata i odgovornoj osobi. Za potrebe ovog projekta zahtjev je zatražila tvrtka Pakel d.o.o. iz Zadra.

Nakon što Fina provjeri podatke i odobri zahtjev na navedeni email Fina šalje referentni broj, a autorizacijski kod na SMS. Pomoću tih podataka preuzimamo certifikate s službene stranice Fine (https://mojcert.fina.hr/finacms/).



Fina

FINA DEMO

Zahtjev za izdavanje Demo certifikata za fiskalizaciju

1. Podaci o poslovnom subjektu*

Skraćeni naziv poslovnog subjekta: Pakel d.o.o.

OIB poslovnog subjekta: 55703284647 Matični broj: 03930319

2. Podaci o skrbniku

Ime*: Robert

Prezime*: Anic

OIB skrbnika*: 86968090976

Identifikacijska isprava*: [X] Osobna iskaznica ili [ ] Putovnica Vrijedi do: ..

Broj identifikacijske isprave*: 111366342

Državljanstvo*: Hrvatsko

Kontakt, poslovna adresa skrbnika

Ulica i broj*: Luke Botica 25

Broj pošte*: 23000 Mjesto*: Zadar

Država*: Hrvatska

Mobitel/telefon*: +385959182050

Adresa e-pošte: robert.anic1@gmail.com

3. Specifikacija traženih certifikata

Certifikacijsko tijelo (CA): [X] Fina Demo CA 2014

Profil certifikata: [X] Demo Certifikat koji je tehnološki jednak produkcijskom Fina RDC 2015 certifikatu za aplikaciju standardne razine sigurnosti (NCP)

Naziv: FISKAL

4. Preuzimanje aktivacijskih podataka*

Preuzimanje aktivacijskih podataka može se obaviti SMS-om i e-poštom, ili osobnim preuzimanjem na lokaciji Fine. Ukoliko se želi dostava aktivacijskih podataka SMS-om i e-poštom jedan dio aktivacijskih podataka biti će dostavljen SMS-om na mobitel, a drugi dio na adresu e-pošte iz točke 3. ovog Zahtjeva.
Preuzimanje aktivacijskih podataka

[X] Dostava SMS-om i e-poštom [ ] U poslovnim jedinicama Fine

5. Izjava skrbnika

Izjavljujem da su svi podaci navedeni u ovom Zahtjevu točni i istiniti te da su dobrovoljno stavljeni na raspolaganje Fini koja će ih koristiti u cilju obavljanja ovdje zahtijevane usluge. Potpisom Zahtjeva potvrđujem da sam upoznat s informacijama ispitaniku o obradi osobnih podataka prikupljenih ovim Zahtjevom koje su dane na slijedećoj stranici ovog Zahtjeva i s Fininim uvjetima pružanja usluga certificiranja poslovnim subjektima te pristajem na njihovu primjenu.
Suglasan sam s javnim objavljivanjem izdanog certifikata iz točke 4. ovog Zahtjeva sukladno Fininim uvjetima pružanja usluga certificiranja poslovnim subjektima.
U slučaju odabira dostave aktivacijskih podataka SMS-om i e-poštom izjavljujem i jamčim da samo ja mogu pristupiti i pročitati poruku upućenu SMS-om na mobitel i na adresu e-pošte iz točke 3. ovog Zahtjeva.

6. Potpis skrbnika i osobe ovlaštene za zastupanje*

Potpis skrbnika | Ime i prezime osobe ovlaštene za zastupanje: Faik Garibovic | Potpis osobe ovlaštene za zastupanje

PAKEL d.o.o. za računalne usluge ZADAR, Luke Botića 25

Popunjava djelatnik LRA Fine

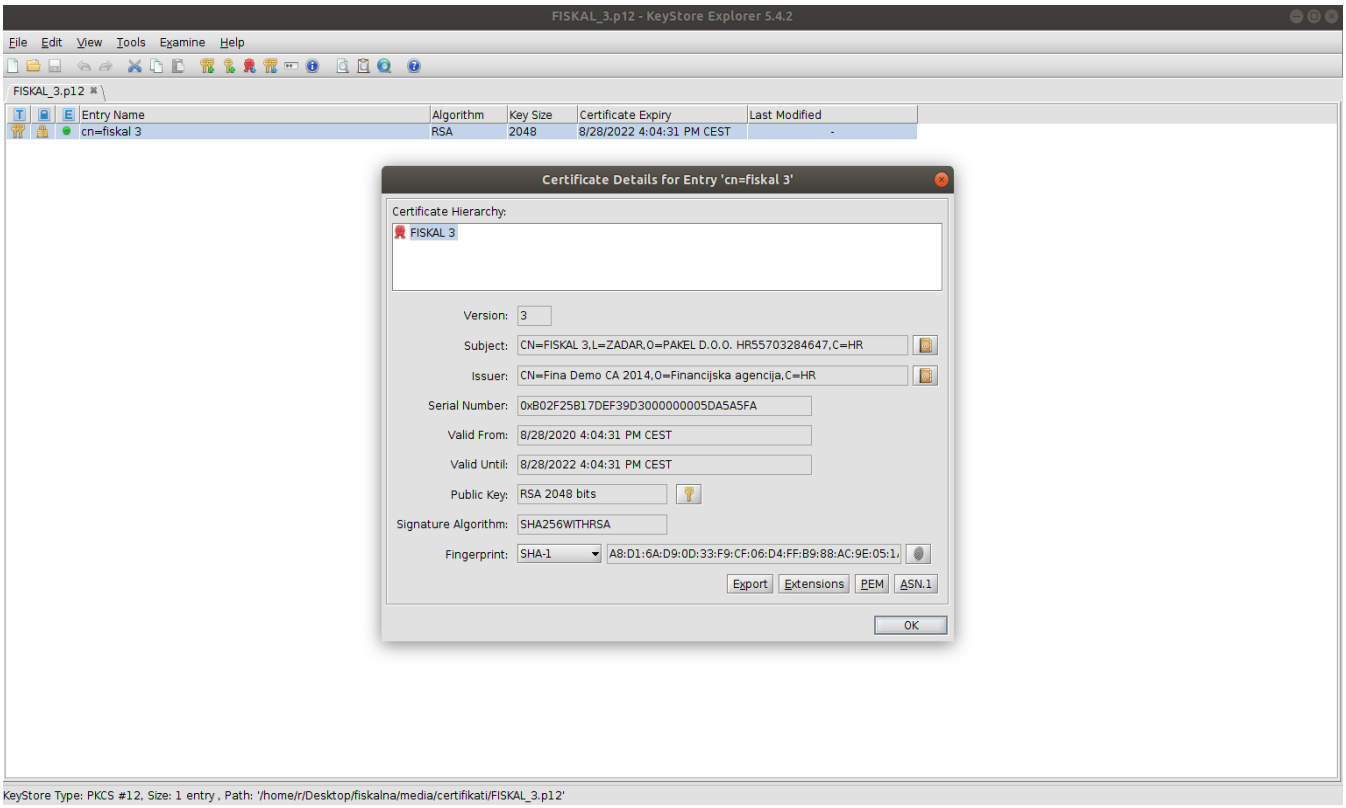
Datum | Potpis | Štambilj

Datum | Potpis

* Obvezno ispuniti sve podatke u označenom stavku ili polju zahtjeva.

Str: 1/2

Slika 20. Zahtjev za Demo certifikatom



Slika 21. Detalji certifikata

Certifikat preuzimamo u .p12 formatu. Na slici 21. su detaljni podaci o certifikatu otvorenom u programu Keystore Explorer 5.4.2.. Za pristup detaljima certifikata postavili smo lozinku pri preuzimanju.

9. ZAKLJUČAK

Cilj ovog rada je bio razviti web aplikaciju za izdavanje fiskaliziranih računa. U uvodnom dijelu rada opisan je koncept fiskalne blagajne i navedene su osnovne tehnologije za razvoj web aplikacija. Detaljnije su prikazane tehnologije HTML i CSS za razvoj dizajna aplikacije, zatim PostgreSQL baza za pohranu svih podataka, programski jezici Python i Javascript i okruženje Django Framework za pristup i rad sa svim podacima koji su pohranjeni u bazi podataka. Prilikom izrade projekta stečeno je teorijsko iskustvo s područja web tehnologija koje su korištene u izradi same web stranice, kao i praktično iskustvo s područja programiranja u Python i Javascript programskim jezicima i korištenja baze podataka. Unaprijeđeno je i praktično znanje CSS jezika koji je korišten za grafički dizajn web aplikacije koja je prilagođena za prikaz na različitim uređajima.

U radu su detaljno opisane funkcionalnosti web aplikacije koristeći razne dijagrame i kroz slike je prikazan izgled svakog ekrana korisničkog sučelja.

SAŽETAK

Potreba za web aplikacijama na tržištu danas je sve veća, a glavni razlog tome je nagli razvoj interneta i brojnih web tehnologija. Primjenom novih tehnologija, svakodnevno se razvijaju multimedijalne aplikacije kojima korisnici mogu pristupati u svakom trenutku zahvaljujući „pametnim“ uređajima. Jedna od takvih aplikacija je i fiskalna blagajna Fiskali koja je prilagođena svim dimenzijama ekrana, uključujući najmanji mobilni ekran pa sve do velikih monitora.

Tema ovog diplomskog rada je razvoj web aplikacije Fiskali, fiskalne blagajne koja poštuje hrvatske zakone o fiskalizaciji. Korištenjem ove web aplikacije korisnik upravlja podacima o svojoj tvrtki, poslovnicama, naplatnim uređajima, zaposlenicima, kupcima i artiklima. Budući da se radi o web aplikaciji, samim time prednost iste je ta što se ne mora instalirati na klijentsko računalo već je moguć online pristup.

U ovom diplomskom radu detaljno su opisane korištene tehnologije kao i rad sustava kroz dijagrame i upute za rad s aplikacijom.

Kjučne riječi: aplikacija, Django, fiskalizacija, Linux, Python, web

SUMMARY

Nowadays, the market demand for web applications is increasing mainly due to the rapid development of the Internet and numerous web technologies. By applying the newly developed technologies multimedia applications are being created daily, granting the users access at any moment via ''smart'' devices. One such application is the fiscal cash register ''Fiskali'', which adapts to all screen dimensions, ranging from the smallest mobile screens up to large format computer monitors.

The topic of this thesis is the development of the web application ''Fiskali'', a fiscal cash register that operates in accordance with Croatian fiscalization laws. By using the application the user manages company data, business branches, payment terminals, employees, buyers and product items. The main advantage of the product being a web application is that it can be accessed online, therefore it isn't necessary to have it installed on the client computer.

This thesis describes in detail the technologies used as well as the performance of the system via diagrams and application operating instructions.

Keywords: application, Django, fiscalization, Linux, Python, web

LITERATURA

1. About JavaScript [Online] Dostupno na: https://www.postgresql.org/docs/current/limits.html [Pristupljeno: 23. kolovoza 2020.].
2. Bray T. (2014) RFC 7159:The JavaScript Object Notation (JSON) Data Interchange Format (2014) [Online] Dostupno na: https://tools.ietf.org/html/rfc7159 [Pristupljeno: 22. kolovoza 2020.].
3. Ellingwood J. (2016) How To Install and Use PostgreSQL on Ubuntu 16.04 [Online] Dostupno na: https://www.digitalocean.com/community/tutorials/how-to-install-and-use-postgresql-on-ubuntu-16-04 [Pristupljeno: 20. kolovoza 2020.].
4. PostgreSQL About PostgreSQL [Online] Dostupno na: https://www.postgresql.org/about [Pristupljeno: 5. kolovoza 2020.].
5. Django [Online] Dostupno na: https://docs.djangoproject.com [Pristupljeno: 10. kolovoza 2020.].
6. Morris J. (2018) How To Create a Django App and Connect it to a Database [Online] Dostupno na: https://www.digitalocean.com/community/tutorials/how-to-create-a-django-app-and-connect-it-to-a-database [Pristupljeno: 11. kolovoza 2020.].
7. Kuchling A. Python PEP 206 -- Python Advanced Library [Online] Dostupno na: https://www.python.org/dev/peps/pep-0206/ [Pristupljeno: 12. kolovoza 2020.].
8. Crockford, D. (2008) *JavaScript: The Good Parts* Sebastopol: O'Reilly Media.
9. Lutz, M. (2013) *Learning Python.* 5.Izdanje. Sebastopol: O'Reilly Media.
10. Kalafatić, Z., S. Šegvić, J. Šribar (2016.) *Python za znatiželjne.* Zagreb*:*Element.

SLIKE

Slika 1. Primjer HTML koda 3
Slika 2. Primjer Python koda 5
Slika 3. Javascript kod 6
Slika 4. JSON format 7
Slika 5. Pycharm programsko sučelje 11
Slika 6. Use Case dijagram 16
Slika 7. Dijagram sekvence 17
Slika 8. Klasni dijagram 18
Slika 9. UML (engl. *Unified Modelling Languge*) Component Dijagram sustava 25
Slika 10. Verifikacijski e-mail poslan iz aplikacije 28
Slika 11. Kreiranje novog računa 29
Slika 12. Popis izdanih računa 30
Slika 13. Filteri izdanih računa 31
Slika 14. Ispis računa 32
Slika 15. Detalji računa 33
Slika 16. Dodjela naplatnog uređaja zaposleniku 34
Slika 17. Unos novog artikla 35
Slika 18. Korisničko sučelje - Pomoć 36
Slika 19. Administratorsko sučelje 37
Slika 20. Zahtjev za Demo certifikatom 39
Slika 21. Detalji certifikata 40

TABLICE

Tablica 1 . Ograničenja PostgreSQL 4
Tablica 2. SWOT analiza 15

KOD

Kod 1. Django postavke 22
Kod 2. Instalacija drivera Psycopg2 22
Kod 3. Instalacija programa Git 23
Kod 4. Preuzimanje projekta 23
Kod 5. Kreiranje i aktiviranje virtualnog okruženja 23
Kod 6. Instalacija paketa iz datoteke requirements.txt 23
Kod 7. Stvaranje PostgreSQL baze podataka 23
Kod 8. Django migracije 24
Kod 9. Stvaranje administratorskog korisnika 24
Kod 11. Postavke za slanje email-a 27
Kod 12. Registriranje modela u administratorsko sučelje 37