

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žan Korečič

**Sistem za upravljanje strank in
prodaje energentov**

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Luka Šajn

Ljubljana, 2017

COPYRIGHT. Rezultati diplomske naloge so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomske naloge je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Za potrebe podjetij, ki se ukvarjajo s prodajo energentov, izdelajte spletno aplikacijo, ki bo uporabnikom omogočala pregled in vodenje podatkov o strankah. V ta namen podrobno preučite zahteve, izberite primerne tehnologije ter predstavite razvoj in uporabnost rešitve. Aplikacija naj omogoča čim prijaznejši uporabniški vmesnik za vnos in pregled podatkov. V zadnjem delu naloge analizirajte opravljeno delo ter predlagajte izboljšave.

Najprej bi se zahvalil svojim staršem, ki so me podpirali med celotnim študijem. Zahvaljujem se tudi mentorju doc. dr. Luki Šajn za vso pomoč in hitro odzivnost pri izdelavi diplomske naloge.

Svojim staršem.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Opis problema	1
1.2	Namen aplikacije	1
2	Uporabljeni programski jeziki in tehnologije za razvoj aplikacije	3
2.1	ASP.NET MVC	3
2.2	C#	4
2.3	Razor	4
2.4	Transact-SQL	4
2.5	HTML	5
2.6	CSS	5
2.7	JavaScript	5
2.7.1	JQuery	6
2.8	Ajax	6
3	Orodja	7
3.1	Visual Studio 2013	7
3.2	Internet Information Services	8
3.3	Google Chrome Developer Tools	8

4	Razvoj spletne aplikacije	9
4.1	Kaj je CRM	9
4.2	Arhitektura aplikacije	10
4.3	Podatkovna baza	11
4.4	Opis aplikacije	12
4.4.1	Aktivnosti	12
4.4.1.1	Dodajanje novih aktivnosti	13
4.4.1.2	Urejanje aktivnosti	14
4.4.1.3	Brisanje aktivnosti	15
4.4.1.4	Podrobnosti aktivnosti	16
4.4.1.5	Filtriranje in razvrščanje	16
4.4.2	Pregled strank	17
4.4.2.1	Iskalnik	17
4.4.2.2	Prikaz podatkov o stranki	19
4.4.2.3	Urejanje podatkov o strankah	20
4.4.2.4	Dodajanje nove stranke	21
4.4.2.5	Brisanje strank	24
4.4.3	Pogodbe	25
4.4.3.1	Dodajanje novih pogodb	26
4.4.3.2	Urejanje in brisanje pogodb	28
4.4.4	Nadzorna plošča	28
4.4.4.1	Dodajanje novih podatkov	29
4.4.4.2	Urejanje podatkov	30
4.4.4.3	Brisanje podatkov	30
5	Analiza	33
5.1	Čas razvoja	33
5.2	Zmogljivost aplikacije	34
5.3	Težave pri razvijanju projekta	34
6	Sklepne ugotovitve	35
6.1	Nadaljnje delo	36

Literatura	36
Priloge	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
CRM	Customer relationship management	Upravljanje odnosov s strankam
HTML	Hypertext markup language	Jezik za označevanje nadbesedila
SQL	Structured query language	Strukturiran povpraševalni jezik
CSS	Cascading style sheets	Predloge za določanje izgleda
XML	eXtensible Markup Language	Razširljiv označevalni jezik
API	Application programming interface	Aplikacijski programski vmesnik
GUI	Graphical User Interface	Grafični uporabniški vmesnik
IDE	Integrated development environment	Integrirano razvojno okolje
HTTP	Hypertext Transfer Protocol	Protokol za prenos hiperteksta
FTP	File transfer protocol	Protokol za prenos datotek

Povzetek

Naslov: Sistem za upravljanje strank in prodaje energentov

Avtor: Žan Korečič

Problem, ki se ga je obravnavalo v diplomski nalogi, je razvijanje sistema za pomoč pri prodaji energentov. Sistem bi pomagal podjetjem, ki se ukvarjajo s prodajo energentov, voditi pregled vseh strank in njihovih informacij, beleženje akcij, ki so jih uporabniki naredili s strankami, in pregled pogodb, ki so sklenjene s podjetjem. Za razvoj aplikacije so se uporabljale spletne tehnologije, ker s tem omogočamo dostop do spletne aplikacije ne glede, kje se uporabnik nahaja, in mu s tem omogočamo pregled podatkov na terenu. Rezultat je delujoča spletna aplikacija, ki jo podjetja lahko že uporabljajo. Njen pregled in razvoj sta opisana v četrtem poglavju diplomske naloge.

Ključne besede: CRM, Upravljanje odnosov s strankami, spletna aplikacija.

Abstract

Title: System for customer management and sale of energy products

Author: Žan Korečič

The problem that was beign discussed in the thesis was the development of a system for assisting sale of energy products. Such a system would help companies that sell energy products to keep an overview of all customers and their information, keeping record of activities done with customers and the ability to review contracts. In order to develop the application, web technologies were beign used because it enables us to access the web application regardless of where the user is located and thereby enable to view the data on the go. The result is a functioning web application that companies can already use. It's course and development is described in the fourth chapter.

Keywords: CRM, customer relationship management, web aplication.

Poglavje 1

Uvod

1.1 Opis problema

Obstajajo podjetja, ki vodijo pregled obstoječih in bodočih partnerjev v Excelovih datotekah ali kot fizične dokumente, kar pri množici podatkov postane nepregledno in nemogoče za resno obdelavo podatkov. Podatki so na voljo v več formatih, kar pomeni ogromno nepotrebnega dela večjega števila ljudi pri pripravi poročil, možnost napak se poveča, nepreglednost podatkov pa preprečuje kakovostna prodajna poročila. Manjka sledljivost prodajnega procesa pri določenem kupcu, pregled cen konkurence na trgu, nimamo zgodovine o posameznem odjemalcu, ni razvidno zakaj na primer ni sklenil pogodbe s podjetjem kot dobaviteljem.

1.2 Namen aplikacije

Namen aplikacije na področju prodaje električne energije in zemeljskega plina poslovnim odjemalcem podjetij, ki v okviru obstoječega informacijskega sistema nimajo na voljo modula CRM, ki bi zadostil prodajnim zahtevam. Za pospešitev prodajnega procesa ter hiter pregled in dostop do podatkov na enem mestu je bilo treba razviti aplikacijo, ki omogoča prodajnim menedžerjem pregled osnovnih podatkov o kupcu, kontaktne osebe, profil od-

jema, dosežen nivo cen in obstoječih pogodb. Z novim sistemom bo možno centralno načrtovati in poročati o prodajnih aktivnostih tako na mesečni, kvartalni kot letni ravni. Preglednost kontaktnih podatkov kupca in pregled zadnjih opravljenih aktivnosti bosta pospešila proces prodaje, hitrejša bo tudi menjava oz. nadomeščanje skrbnikov ključnih kupcev v primeru odsotnosti. Vodstvo bo pridobilo pregled in nadzor aktivnosti po posameznem prodajnem skrbniku. Z odjemalci bo možna dvosmerna komunikacija in nudenje novih storitev obveščanja kot npr. obveščanje o doseženi dogovorjeni ceni posameznega energenta, možna je priprava mailing liste. Za potrebe analitike in bodočih strateških odločitev se vodi zgodovina vseh posameznih vpisov in aktivnosti.

Poglavje 2

Uporabljeni programski jeziki in tehnologije za razvoj aplikacije

V sledečem poglavju bomo opisali vse programske jezike in tehnologije, ki so bili uporabljeni za razvoj aplikacije.

2.1 ASP.NET MVC

ASP.NET MVC je ogrodje za izdelavo spletnih aplikacij z uporabo arhitekture Model-View-Controller (MVC). MVC je arhitekturni vzorec, ki spodbuja striktno izolacijo med individualnimi deli aplikacije. Aplikacijo razdelimo na tri dele: model, pogled in krmilnik.

Vsak razdeljeni del ima svoje specifično delo, in je zadolžen za izvajanje ter ne vpliva na delovanje drugih delov. Model predstavlja osnovno poslovno logiko in podatke, enkapsulira lastnosti in obnašanja entitete ter izpostavlja lastnosti, ki opisujejo entiteto. Pogled je zadolžen za transformiranje modela ali modelov v vizualen prikaz. V spletnih aplikacijah običajno to pomeni generiranje kode HTML, ki se upodobi v spletnem brskalniku. Pogledi se morajo osredotočiti samo na prikazovanje podatkov in ne smejo vsebovati

katerekoli poslovne logike. Za poslovno logiko so zadolženi modeli. Krmilnike se uporablja za nadzor nad logiko aplikacije in delujejo kot koordinator med modelom in pogledom. Krmilniki dobijo vhodne podatke od uporabnikov prek pogledov in potem sodelujejo z modelom za opravljanje specifičnih akcij. Rezultati akcij se pošljejo nazaj na pogled [12].

2.2 C#

C# je objektno usmerjen programski jezik, ki ga je Microsoft razvil v zgodnjem začetku leta 2000. Objektno usmerjeno programiranje je pristop h programiranju, kjer problem razdelimo na objekte, ki medsebojno sodelujejo. C# je del Microsoftove platforme .Net in je namenjen kot splošno namenski programski jezik za razvijanje različnih tipov aplikacij kot naprimer razvoj spletnih aplikacij, mobilne aplikacije in aplikacij Windows. C# ima sintakso in posebnosti, ki so podobne drugim programskim jezikom kot sta Java in C++. Ogrodje .Net vsebuje zelo veliko knjižnico že napisanih kod in s tem programerjem prihrani veliko časa, da jim ni treba pisati od začetka stvari, ki so že narejene [10].

2.3 Razor

Razor je označevalna sintaksa, ki omogoča vključitev kod C# ali Visual Basic v spletne strani. Z razorjem lahko ustvarjamo dinamično spletno vsebino na spletnih straneh, vključena koda pa se izvaja na strežniku in ne pri odjemalcu. Temelji na ogrodju ASP.NET in je zasnovan za ustvarjanje spletnih aplikacij [7].

2.4 Transact-SQL

Transact-SQL je Microsoftova razširitev strukturiranega povpraševalnega jezika za delo z relacijskimi podatkovnimi bazami. Razširitve, ki jih vključuje v

standard SQL so proceduralno programiranje, uporaba lokalnih spremenljivk, podporne funkcije za obdelavo nizov, procesiranje datumov in računanje, prinaša pa tudi spremembe stavkov za posodabljanje in brisanje podatkov. T-SQL je ključen pri uporabi Microsoftovega strežnika SQL, saj aplikacije, ki komunicirajo z instanco strežnika SQL, delajo tako, da pošiljajo stavke T-SQL strežniku ne glede na uporabniški vmesnik aplikacije [8].

2.5 HTML

HTML je jezik za izdelovanje spletnih strani in spletnih aplikacij. HTML ni programski jezik, ampak označevalni jezik, kar pomeni, da je sistem za identifikacijo in opisovanje različnih komponent dokumenta. HTML označuje osnovno strukturo dokumenta [13].

2.6 CSS

CSS so kaskadne slogovne podloge, ki se uporabljajo za opisovanje prezentacije dokumentov, napisanih v HTML ali XML. CSS opisuje kako, naj se elementi upodobijo na spletni strani [3].

2.7 JavaScript

JavaScript je interpretiran programski jezik z objektno usmerjenimi sposobnostmi. Pri JavaScriptu so spremenljivke brez strogo definiranega tipa kar pomeni, da spremenljivkam ni treba povedati, katerega tipa so. Jedro programskega jezika podpira števila, nize in logične vrednosti kot primitivne podatkovne tipe. Vsebuje tudi podporo za polja, datume in regularne izraze. Najbolj pogosto je uporabljen v spletnih brskalnikih, je razširjen z objekti, ki dovolijo skriptam interakcijo z uporabnikom, nadzor nad spletnim brskalnikom in spreminjanje vsebine dokumenta, ki je prikazan v oknu spletnega brskalnika. JavaScript zaganja skripte, ki so vgrajene znotraj spletnih strani,

to pomeni, da se koda izvaja na odjemalčevem računalniku in ne na strežniku [11].

2.7.1 JQuery

JQuery je knjižnica za JavaScript, ki vsebuje bogat nabor funkcionalnosti. Z uporabo vmesnika uporabniškega programa (API) nam omogoča obhod po dokumentih HTML, manipuliranje z elementi, upravljanje z dogodki, ustvarjanje animacij in lažjo uporabo Ajaxa. JQuery tako kot JavaScript deluje na vseh spletnih brskalnikih [6].

2.8 Ajax

Ajax je skupek spletnih tehnologij za ustvarjanje asinhrono spletne aplikacije. Z Ajaxom spletne aplikacije lahko pošiljajo in pridobivajo podatke s strežnika brez motenja prikaza in obnašanja obstoječe strani. Vsebina spletne strani ali aplikacije se lahko dinamično spreminja brez osveževanja [1].

Poglavje 3

Orodja

3.1 Visual Studio 2013

Za razvoj spletne aplikacije se je uporabljalo Microsoftov Visual Studio 2013. Visual Studio je integrirano razvojno okolje (IDE) za razvijanje programov za Windows, izdelavo spletnih strani, spletnih aplikacij, spletnih storitev in mobilnih aplikacij. Vsebuje Microsoftove razvojne platforme, kot so Windows API, Windows Forms, Windows Presentation Foundation, Windows Store in tudi Microsoft Silverlight. Visual Studio tudi vsebuje komponente, ki nam omogočajo samodejno dopolnjevanje kode (IntelliSense) in avtomatsko preoblikovanje kode. V razvojno okolje je tudi že vključen razhroščevalnik, ki deluje na izvornem in strojnem nivoju. Razhroščevalnik se uporablja za ugotavljanje in iskanje napak ter spremljanje vrednosti spremenljivk, ki so prisotne v programu. Poleg vseh naštetih lastnosti ima Visual Studio še dodatna integrirana orodja, kot so nadzorovanje porabe pomnilnika, prikaz časovne zahtevnosti programa, urejevalnik za izdelavo grafičnih vmesnikov (GUI) in orodje za upravljanje s podatkovnimi bazami. Okolje podpira veliko programskih jezikov. Lahko razvijamo aplikacije za splet, spletne strani, mobilne aplikacije in programe za operacijski sistem Windows [9].

3.2 Internet Information Services

Internet Information Services (IIS) je splošno namenski spletni strežnik, ki ga je razvil Microsoft za streženje zahtevanih HTML strani ali datotek. Spletni strežnik sprejema odjemalčeve zahteve in vrača temu primerne odgovore. Odgovore lahko vrača na različne načine, kot so npr. statične spletne strani, datotečna izmenjava, tekstovne datoteke, slike in podobno. IIS deluje skozi različne standardne jezike in protokole. Hypertext Transfer Protocol (HTTP) je uporabljen kot osnovni komunikacijski protokol za izmenjavo informacij med strežniki in uporabniki. Za večjo varnost se uporablja tudi Secure Sockets Layer (SSL) za enkripcijo komunikacije. Omogoča tudi izmenjavo datotek s pomočjo File Transfer Protocola (FTP) [5].

3.3 Google Chrome Developer Tools

Developer Tools je skupek orodij, ki so vgrajena v Google Chrome za razvijanje in razhroščevanje spletnih strani in spletnih aplikacij. Uporabniku omogočajo dostop do notranjosti brskalnika in njihovih spletnih strani oziroma spletnih aplikacij. Orodje se uporablja za ugotavljanje težav, kot so postavitev elementov, napake v kodi JavaScript, postavitev zaustavitvenih točk in za optimizacijo kode. Chrome Developer Tools je razdeljen na osem skupin in vsaka skupina omogoča delo z različnimi tipi strani ali informacij [2].

Poglavje 4

Razvoj spletne aplikacije

4.1 Kaj je CRM

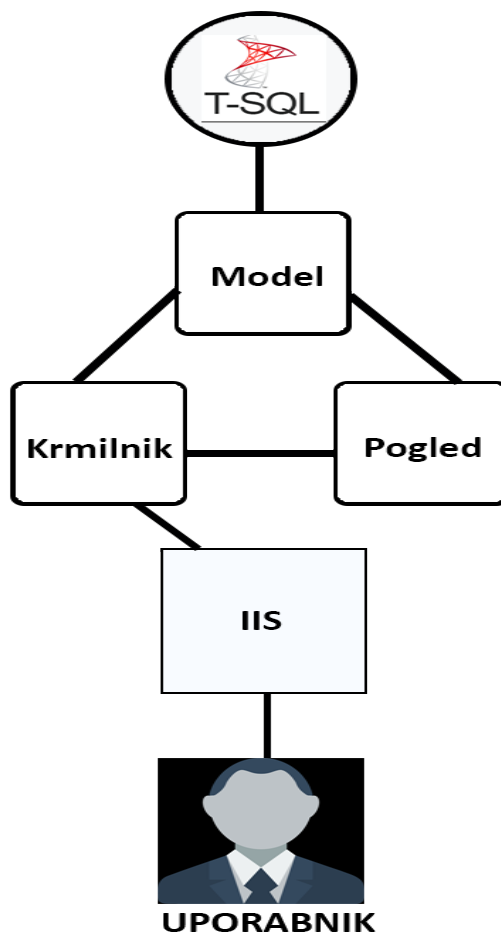
Upravljanje odnosov s strankami je izraz, ki se nanaša na prakse, strategije in tehnologije, ki jih podjetja uporabljajo za upravljanje ter analizo podatkov in interakcij s stranko s ciljem izboljšave poslovnega odnosa med stranko in podjetjem, zadrževanja strank in višanja prodajne rasti. Sistemi CRM so namenjeni zbiranju informacij o strankah po različnih kanalih ali kontaktnih točkah med stranko in podjetjem. Lahko vsebujejo tudi podrobnejše informacije o strankah, kot na primer njihove osebne podatke, zgodovino nakupov, zgodovino interakcij, seznam kontaktov in seznam pogodb, ki so sklenjene. Ostale funkcionalnosti te programske opreme vključujejo tudi avtomatizacijo različnih poslovnih procesov, kot so naloge, koledarji in opozorila, ter možnost beleženja uspešnosti in produktivnosti na podlagi podatkov, ki so bili vneseni v sistem. Vse informacije so združene v eno podatkovno bazo zato, da poslovni uporabniki lažje dostopajo in upravljajo s podatki [4].

Pogoste funkcionalnosti:

- avtomatizacija marketinga: orodja s sposobnostjo avtomatizacije ponavljajočih se nalog za izboljšanje trženjskih prizadevanj (ang. marketing efforts) za stranke,
- avtomatizacija prodajne sile: preprečevanje dvojnih prizadevanj med prodajalcem in stranko,
- avtomatizacija centra kontaktov: programska orodja za poenostavljanje postoka pri obvladovanju zahtev strank
- geolokacijske storitve: nekateri sistemi vključujejo tudi ustvarjanje geografskih marketinških akcij na podlagi fizičnih lokacij strank.

4.2 Arhitektura aplikacije

Aplikacija deluje po principu odjemalec - strežnik (Slika 4.1). Odjemalec pošilja zahteve in strežnik jih procesira ter vrne odgovor nazaj odjemalcu. Takšna arhitektura nam pomaga pri varnosti, grafični vmesnik je neodvisen in omogoča boljšo avtonomijo podatkovne baze.



Slika 4.1: Arhitektura spletne aplikacije

4.3 Podatkovna baza

Za delovanje sistema je treba imeti podatkovno bazo, kjer bomo hranili vse podatke. Za postavitev podatkovne baze smo uporabili Microsoft SQL Server 2012. Vse tabele in njihove pripadajoče povezave so prikazane v prilogi [2].

4.4 Opis aplikacije

V naslednjih poglavjih bomo prikazali arhitekturo, izgled podatkovne baze, izgled aplikacije in njene funkcionalnosti.

4.4.1 Aktivnosti

Ob uspešnem vpisu v aplikacijo se nam odpre stran Aktivnosti. Stran je razdeljena na dva dela. Prvi del je tabela s podatki in drugi del so filtri s katerimi upravljamo nad podatki v tabeli. Izgled strani je prikazan v prilogi [1].

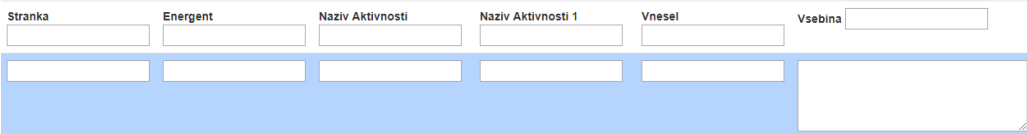
Tabela je sestavljena iz desetih stolpcev:

1. stranka: vsebuje ime stranke oziroma podjetja na katerega se veže aktivnost,
2. energent: za kateri energent smo ustvarili aktivnost (elektrika, zemeljski plin, itd.),
3. naziv aktivnosti: kratek opis kaj smo naredili s stranko,
4. naziv aktivnosti 1: podrobnejši opis prvega naziva aktivnosti,
5. vnesel: kdo je ustvaril aktivnost,
6. vsebina: komentar oziroma dodatna objasnitev,
7. datum aktivnosti: vsebuje datum do kdaj mora uporabnik dokončati aktivnost,
8. datum vnosa: vsebuje datum vnosa aktivnosti,
9. status: prikazuje trenutno stanje aktivnosti in pripadajočo barvo glede na status (Novo, V obdelavi in Končano),
10. upraviteljski stolpec: v tem stolpcu imamo izbiro za urejanje, podrobnejši pogled in brisanje aktivnosti.

Filtri za upravljanje s podatki so vidni nad tabelo in v stolpcih same tabele kot spustni sezname in tekstovna polja. Filtri, ki so nad tabelo, nam omogočajo izbiro po datumih aktivnosti, po datumih vnosa in koliko vrstic naj prikaže. Preostali del filtrov, ki so v stolpcih, pa nam omogočajo filtriranje po ostalih podatkih, kjer vpišemo želen niz v tekstovna polja in se nam prikažejo vse možnosti v spustnem seznamu, ki vsebujejo vpisani niz.

4.4.1.1 Dodajanje novih aktivnosti

Dodajanje novih aktivnosti sprožimo s klikom na povezavo *Dodaj novo aktivnost*. Po kliku na povezavo se izvede JavaScript koda, ki spremeni stil prve vrstice iz nevidne na vidno in tako se nam doda nova vrstica, obarvana modro pod stolpci tabele (Slika 4.2).



Stranka	Energent	Naziv Aktivnosti	Naziv Aktivnosti 1	Vnesel	Vsebina
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

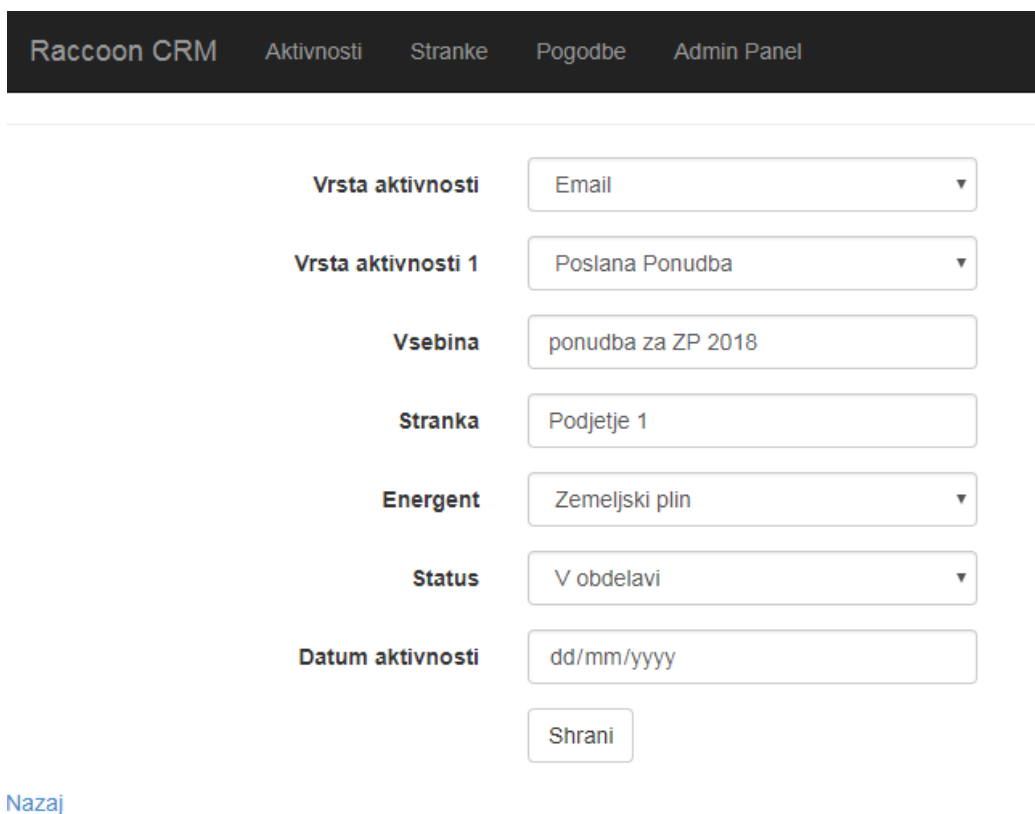
Slika 4.2: Izgled vrstice za dodajanje nove aktivnosti

Dodana vrstica vsebuje spustne sezname, ki vsebujejo vse možne izbire razen pri stranki, kjer je tekstovno polje za iskanje po nazivih podjetij, pri vsebini je samo tekstovno polje, kjer uporabnik napiše dodatna pojasnila oziroma komentarje, in za izbiro datumov se ob kliku nanj odpre koledar za izbiro datuma namesto, da uporabnik ročno vpisuje. Za vnos izpolnjene aktivnosti mora uporabnik potrditi vnos z *Dodaj*, drugače se ob osvežitvi strani ali prehodu na drugo stran podatki izgubijo. Izvede se Ajax klic na krmilnik, ki pošlje vrednosti vseh vnosnih polj in nam prikaže vnesene aktivnosti na vrhu tabele. Uporabnik ima tudi možnost izbrisa vnesenih podatkov aktivnosti s *Počisti*. Vsi podatki, ki so bili izpolnjeni v vrstici za dodajanje se bodo izbrisali in vrstica bo še vedno vidna, dokler uporabnik ne doda aktivnosti ali pa skrije vrstico s klikom na *Skrij*. Vrstica za dodajanje se samo skrije, podatki pa še vedno ostanejo, če se uporabnik odloči, da bo kasneje

dodal aktivnost.

4.4.1.2 Urejanje aktivnosti

Uporabniku moramo tudi omogočiti urejanje vnesenih aktivnosti. Na strani *Aktivnosti* se v vsaki vrstici na koncu v upraviteljskem stolpcu nahaja povezava *Uredi*. Uporabniku se odpre stran za urejanje izbrane aktivnosti, kjer ima možnost urejanja vrste aktivnosti, vrste aktivnosti 1, vsebine, stranke, energenta, statusa in datuma aktivnosti (Slika 4.3).



Raccoon CRM Aktivnosti Stranke Pogodbe Admin Panel

Vrsta aktivnosti Email ▼

Vrsta aktivnosti 1 Poslana Ponudba ▼

Vsebina ponudba za ZP 2018

Stranka Podjetje 1

Energent Zemeljski plin ▼

Status V obdelavi ▼

Datum aktivnosti dd/mm/yyyy

Shrani

[Nazaj](#)

Slika 4.3: Stran za urejanje aktivnosti

Urejanje datuma vnosa in imena uporabnika, ki je ustvaril aktivnost, je izključeno, saj če bi dovolili urejanje teh podatkov, bi lahko ustvarili veliko zmešnjavo. Ob zaključku urejanja potrdimo s *Shrani*, da se spremembe

shranijo v podatkovno bazo, sicer izgubimo, kar smo hoteli popraviti, in nas potem vrne na prejšnjo stran. Na strani za urejanje obstaja tudi povezava *Nazaj*, če se uporabnik premisli in noče nič uredit in ga vrne na prejšnjo stran.

4.4.1.3 Brisanje aktivnosti

Poleg funkcije *Uredi* imamo tudi možnost brisanja vnosov aktivnosti v primeru, da se je stanje spremenilo ali pa se je uporabnik zmotil. Povezava *Zbriši* pripelje uporabnika na stran, kjer so izpisani vsi podatki o aktivnosti (Slika 4.4).



Slika 4.4: Brisanje aktivnosti

Tukaj pregledamo vse vrednosti, ki so izpisane na seznamu, in s klikom na *Izbriši* odstranimo to aktivnost iz podatkovne baze. Če pa uporabnik ugotovi, da hoče izbrisati napačno aktivnost, se lahko vrne nazaj na prejšnjo stran z *Nazaj*.

4.4.1.4 Podrobnosti aktivnosti

Ko uporabnik hoče dodati novo aktivnost oziroma je v kontaktu s stranko, mora imeti pred seboj tudi informacije o stranki. Funkcija *Podrobnosti* nam v spletnem brskalniku odpre novo okno in v njem prikaže vse informacije o stranki, na katero je vezana aktivnost. Podrobnejši opis tega okna je v poglavju 4.4.2.2.

4.4.1.5 Filtriranje in razvrščanje

Pri veliki količini podatkov pride hitro do zmešnjav in nepreglednosti, zato je bila vključitev filtrov in možnosti razvrščanja nujna. Implementacija razvrščanja je narejena tako, da podpira razvrščanje po več stolpcih. Razvrščanje se sproži s klikom na ime stolpca. Ob prvem kliku bodo podatki razvrščeni naraščajoče, ob drugem padajoče in ob tretjem kliku se razvrščanje odstrani za tisti stolpec. V ozadju se generira seznam, ki vsebuje vse tri kombinacije razvrščanja. Ko uporabnik sproži razvrščanje, najprej pogleda na seznam, na katerem mestu se trenutno nahaja, in izbere naslednjega v vrsti ter prikaže znak na imenu stolpca, da vemo, ali gre za naraščajoče, padajoče ali pa brez. (Slika 4.5).

```
var currentOrder = this.getAttribute('value');  
  
var orders = [this.getAttribute('value').split(' ')[0].replace(' ', '').replace(' ', '') + ' X',  
             this.getAttribute('value').split(' ')[0].replace(' ', '').replace(' ', '') + ' ASC',  
             this.getAttribute('value').split(' ')[0].replace(' ', '').replace(' ', '') + ' DESC'];  
  
var nextOrder = orders[(orders.indexOf(currentOrder) + 1) % orders.length];  
  
this.setAttribute('value', nextOrder);  
  
var Params = this.getAttribute('value').split(' ');
```

Slika 4.5: Izbira načina razvrščevanja

Za razvrščanje po več stolpcih imamo definiran seznam, ki hrani vrstni red. Ob vsakem kliku na stolpce se shranita ime stolpca in način razvrščanja. Filtriranje sprožimo z vpisom teksta v tekstovna polja, ki so pod imeni stolpcev, ob pisanju v tekstovna polja pa se uporabniku prikažejo vsi rezultati, ki

vključujejo vpisani niz (Slika 4.6).

Stranka	Energent [▲]	Naziv Aktivnosti [▲]	Naziv Aktivnosti 1
<input type="text" value="Podjetje 1"/>	<input type="text" value="Elektrika"/>	<input type="text" value="Email"/>	<input type="text"/>

Slika 4.6: Prikaz uporabe filtrov

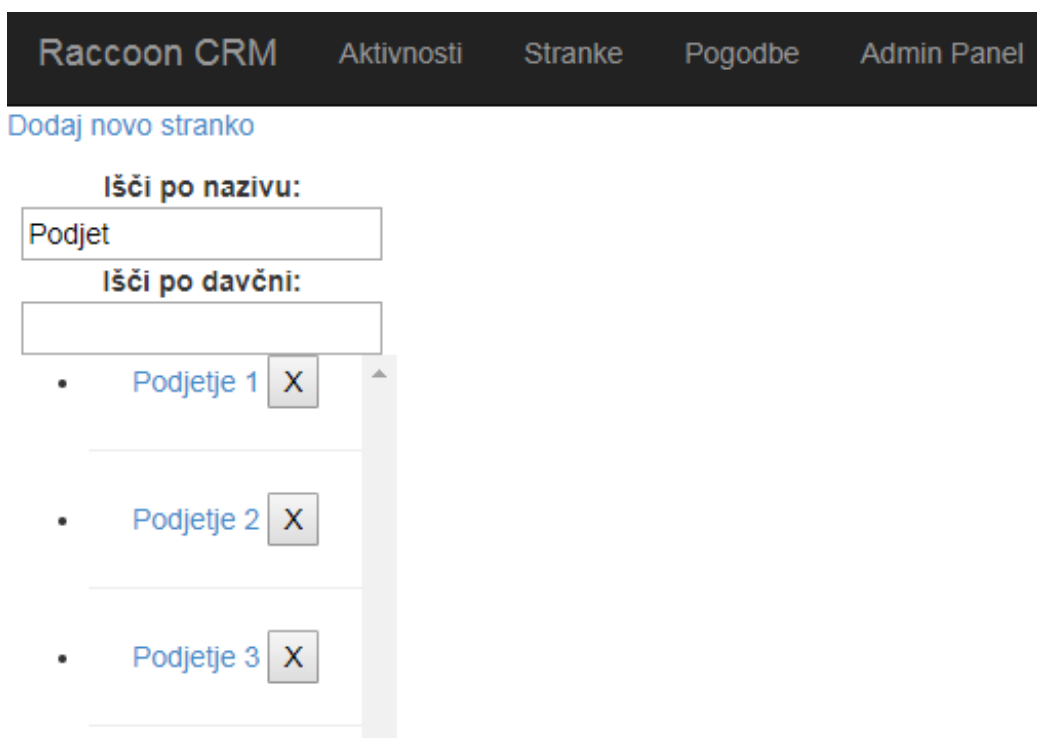
Po izpolnitvi vseh iskalnih kriterijev mora uporabnik klikniti *Search*, da pošlje poizvedbo na podatkovno bazo in vrne nazaj iskane podatke.

4.4.2 Pregled strank

Pod zavihkom *Stranke* dobimo vpogled nad vsemi strankami, ki se nahajajo v podatkovni bazi. Na tej strani imamo možnost dodajanja novih podjetij, pregled informacij, kot so nazivi podjetij, davčne številke, matične številke, naslovi podjetij, njihovi kontakti, sklenjene pogodbe s temi strankami in zgodovina aktivnosti za izbrano stranko. Poleg pregleda informacij imamo tudi možnost urejanja podatkov o strankah, naslovih in njihovih kontaktih.

4.4.2.1 Iskalnik

Na levi strani se nahaja iskalnik, s katerim iščemo podjetja po podatkovni bazi. Možnost iskanja imamo po nazivu podjetja ali po davčni številki, ker lahko obstaja več podjetij, ki vsebujejo vpisani niz, ali pa je ime podjetja prekratko in dobimo še nazive drugih podjetij, ki vsebujejo ta niz (Slika 4.7).



Slika 4.7: Iskalnik

Iskanje se izvede z vpisom teksta v polje za iskanje po nazivu ali pa po davčni številki in se po pol sekunde pošlje poizvedba v podatkovno bazo (Slika 4.8).

```
$(function () {  
    $(".SearchBox").on('propertychange input', function (e) {  
        e.preventDefault();  
  
        clearTimeout(typingTimer);  
        typingTimer = setTimeout(searchDB, doneTypingInterval);  
    })  
})
```

Slika 4.8: Koda za izvedbo iskanja po določenem času

Rezultati, ki vsebujejo iskani niz, se nam avtomatično prikažejo pod iskalnikom kot seznam. S klikom na zeleno podjetje se nam odprejo podatki o

izbranem podjetju na desni strani. Imamo tudi možnost uporabe tipke enter za prikaz informacij, vendar če pritisnemo tipko enter, se nam prikažejo informacije podjetja, ki je prvo na seznamu. Ob vsakem nazivu podjetja se tudi prikaže gumb za brisanje.

4.4.2.2 Prikaz podatkov o stranki

Ko uporabnik izbere željeno podjetje, se mu stran dopolni z vsemi informacijami o izbranem podjetju.



Nov pogled je razdeljen na tri dele. Prvi del vsebuje vso zgodovino aktivnosti (Slika 4.9), ki so vezane na to stranko, in ima isto funkcionalnost kot pri zavihku *Aktivnosti*, vendar z razliko, da ne vsebuje vseh stolpcev. Drugi del je sestavljen s podatki o podjetju, kjer izpiše vse informacije o tem podjetju, naslov podjetja in njihove kontakte (Slika 4.10). Zadnji del pa predstavlja kratek pregled vseh pogodb, ki so ali so bile sklenjene s tem podjetjem (Slika 4.11).




Datum aktivnosti od: do

Število prikazanih vrstic:

Energent	Naziv Aktivnosti	Naziv Aktivnosti 1	Vnesel	Vsebina	Datum Aktivnosti
Zemeljski plin	Email	Poslana Ponudba	Uporabnik 2	ponudba za ZP 2018	08.11.2017
Zemeljski plin	Email	Realizirana pogodba	Uporabnik 1	prijava menjave dobavitelja	21.08.2017

Slika 4.9: Prikaz zgodovine aktivnosti o stranki

 Stranka:		 Naslov stranke:	
	Naziv: Podjetje 1		Ulica: Ulica
	Kratek naziv:		Hišna številka: 2
	Segment: VPO		Dodatek k hišni številki:
	Prodajni kanal: Trgovanje		Pošta: Ljubljana
	Šifra partnerja:		Poštna številka: 2017
	Matična številka:		Država: Slovenija
	Skrbnik kupca: Uporabnik 3		
	Davčna številka: xxxxxxxx		

 Kontaki:	
	 
	Kontakt:
	Energent
	Telefon:
	Email:
	Funkcija:
	Opombe:

Slika 4.10: Prikaz podatkov o stranki

Pogodbe:	Energent	Dobavitelj	Obdobje dobave od	Obdobje dobave do	Tip pogodbe	Dobava / odkup	Marža	EPCP
	Elektrika	Dobavitelj 1	01.01.2017	31.12.2017	X	Y	0	0

Slika 4.11: Prikaz sklenjenih pogodb s stranko

4.4.2.3 Urejanje podatkov o strankah

Podatke o podjetjih lahko tudi urejamo. Gumb za urejanje je prisoten pri vsaki sekciji informacij. Način urejanja sprožimo s klikom na gumb za urejanje in tekst se zamenja z vnosnimi polji (Slika 4.12).

<input checked="" type="checkbox"/> Stranka:	<input checked="" type="checkbox"/> Naslov stranke:
Naziv: Podjetje 1	Ulica: Ulica
Kratek naziv:	Hišna številka: 2
Segment: VPO	Dodatek k hišni številki:
Prodajni kanal: Trgovanje	Pošta: Ljubljana
Šifra partnerja:	Poštna številka: 2017
Matična številka:	Država: Slovenija
Skrbnik kupca: Uporabnik 3	
Davčna številka: xxxxxxxx	

<input type="checkbox"/> Kontakti:	
<input checked="" type="checkbox"/> <input type="checkbox"/>	
Kontakt:	
Energent:	
Telefon:	
Email:	
Funkcija:	
Opombe:	

Slika 4.12: Urejanje informacij o stranki

Ko uporabnik zaključi z vnosom novih vrednosti, klikne nazaj na isti gumb kot za urejanje in s tem potrdi spremembe. Pri seznamu kontaktov ima vsak kontakt svoj gumb za urejanje in tako lahko urejamo vsakega posamično. Poleg urejanja kontaktov imamo tudi možnost dodajanja novih kontaktov. S klikom na gumb za dodajanje prikažemo vnosna polja, kjer izpolnimo podatke o novem kontaktu. Po zaključitvi vnosa podatkov potrdimo spremembe in dodamo nov kontakt za trenutno odprto stranko. Vsak kontakt ima tudi gumb za brisanje. Kontakt mogoče ne obstaja več pri podjetju in ga je treba izbrisati. To izvedemo s klikom na gumb za brisanje pri zelenem kontaktu in potrdimo izbiro.

4.4.2.4 Dodajanje nove stranke

Dodajanje sprožimo s povezavo *Dodaj novo stranko*. Odpre se nam nova stran, ki je ločena na tri dele z navpično črto, kjer bo uporabnik izpolnil vse informacije o novi stranki. Izpolnjevanje se izvede z leve proti desni strani. Najprej se izpolnijo podatki o naslovu podjetja, če ta naslov ne obstaja v

podatkovni bazi ali pa če naslov že obstaja, lahko izpustimo ta vnos in ga samo poiščemo v spustnem seznamu za naslov pri izpolnjevanju informacij o podjetju (Slika 4.13).

Vnos novega naslova:

Ulica: *

Hišna številka:

Dodatek hišni številki:

Pošta:

Poštna številka:

Država:

Shrani

Slika 4.13: Izgled izpolnjevanja novega naslova

Za vnos novega naslova je treba izpolniti vsa polja, ki so označena z zvezdico, in klikniti na gumb shrani. Nov naslov se bo shranil samo, če so polja, označena z zvezdico, izpolnjena in bo avtomatično izbran pri izpolnjevanju informacij o stranki, sicer bo uporabnika obvestil, da naslov ni bil dodan. Naslednji korak je izpolnjevanje podatkov o novem podjetju in veljajo ista pravila kot za naslov. Ob uspešnem vpisu se bo novo podjetje avtomatično izbralo pri vnosu kontaktov ali pa bo uporabnika obvestilo, da novo podjetje ni bilo dodano (Slika 4.14).

Vnos nove stranke:

Naziv: *

Kratki naziv:

Segment: *
VPO ▼

Prodajni kanal: *
Trgovanje ▼

Matična številka:

Šifra partnerja:

Skrbnik kupca: *
Uporabnik 1 ▼

Davčna številka: *

Naslov: *

Vnesel: *

Slika 4.14: Izgled vnašanja informacij o novi stranki

Zadnji del pa je dodajanje kontaktov (Slika 4.15). Kontaktov je lahko več, zato se pri izpolnitvi vseh potrebnih podatkov po kliku na *Shrani* doda nov kontakt za to stranko in se vsa polja razen naziva podjetja izbrišejo za dodajanje drugega kontakta. Ko uporabnik zaključi z dodajanjem, se lahko vrne nazaj na prejšnjo stran z *Nazaj*.

Vnos novega kontakta:

Ime : *

Priimek: *

Funkcija:

Email:

Telefon:

Energent: *

Stranka: *

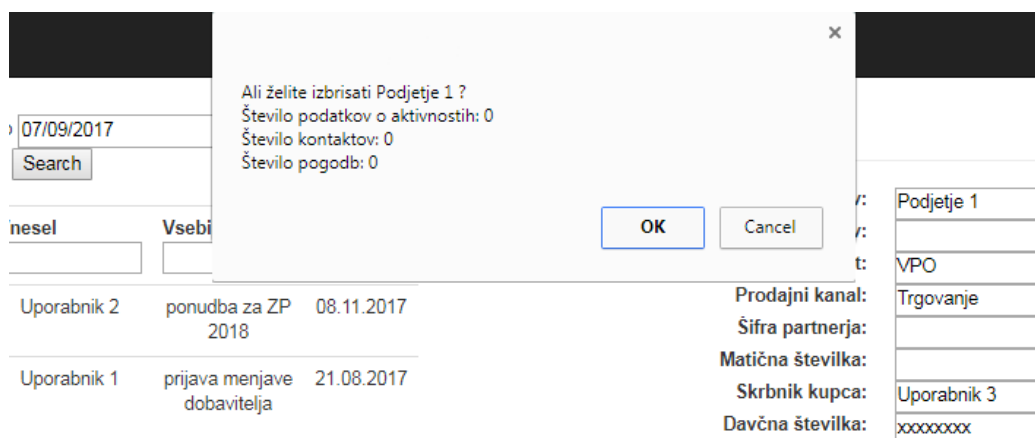
Opombe:

Vnesel: *

Slika 4.15: Dodajanje kontaktov za stranko

4.4.2.5 Brisanje strank

Ko iščemo podjetja, imamo pri iskalniku poleg nazivov podjetij tudi gumb za brisanje. Če neke stranke ne potrebujemo več ali stranka ne obstaja več, jo lahko izbrišemo s klikom na gumb za brisanje. Pojavilo se nam bo okno, ki nam sporoči, koliko podatkov je še vezanih na to stranko (Slika 4.16).



Slika 4.16: Brisanje strank

Pokazalo nam bo število aktivnosti, ki so zapisane za to stranko, koliko kontaktov ima in koliko pogodb. Potrditev izbrisa stranke bo tudi izbrisala vse ostale informacije, ki so vezane na njo. V ozadju se izvedejo tri poizvedbe, ki najprej izbršejo vse kontakte strank, nato vse aktivnosti in na koncu še samo stranko. Vrednosti se morajo izbrisati v takšnem vrstnem redu zaradi povezav med tabelami v podatkovni bazi.

4.4.3 Pogodbe

Stran *Pogodbe* služi uporabniku za pregled vseh vnesenih pogodb v sistemu, dodajanje novih pogodb in urejanje obstoječih pogodb. Pogled je sestavljen iz tabele pogodb in filtrov, s katerimi upravljamo nad podatki v tabeli. Funkcionalnost je ista kot pri aktivnostih vendar z razliko, da tukaj upravljamo s pogodbami. Izgled strani je prikazan v prilogi [3].

Tabela je sestavljena iz dvajsetih stolpcev:

1. sifra partnerja: dodatna identifikacijska številka za stranko,
2. energent: za kateri energent je bila pogodba sklenjena,
3. dobavitelj: kdo dobavlja energent,

4. stranka: naziv podjetja s katerim je sklenjena pogodba,
5. vnesel: kdo je ustvaril pogodbo,
6. tip pogodbe: za kakšen tip pogodbe gre
7. dobava odkup: ali gre za dobavo ali odkup,
8. letni odjem: količina odjema,
9. VT, MT, ET: cena,
10. EPCP: cena s pribitkom,
11. marža: dobiček,
12. obdobje dobave: od kdaj do kdaj velja pogodba,
13. datum sklenitve: kdaj je bila pogodba sklenjena,
14. datum vnosa: kdaj je bila pogodba vnešena,
15. datum spremembe: kdaj je bila pogodba spremenjena.

4.4.3.1 Dodajanje novih pogodb

Novo pogodbo dodamo z *Dodaj novo pogodbo*. Odpre se nam stran z vnosnimi polji, kamor bo uporabnik vnesel vse znane podatke o pogodbi (Slika 4.17).

Raccoon CRM Aktivnosti Stranke Pogodbe Admin Panel

Šifra partnerja

Energent

Tip pogodbe

Dobava / odkup

Dobavitelj

Letni odjem [MWh]

VT [€/MWh]

MT [€/MWh]

ET [€/MWh]

EPCP [€/MWh]

Marža [€/MWh]

Obdobje dobave od

Obdobje dobave do

Datum sklenitve

Stranka

Vnesel

Datum vnosa

Slika 4.17: Dodajanje nove pogodbe

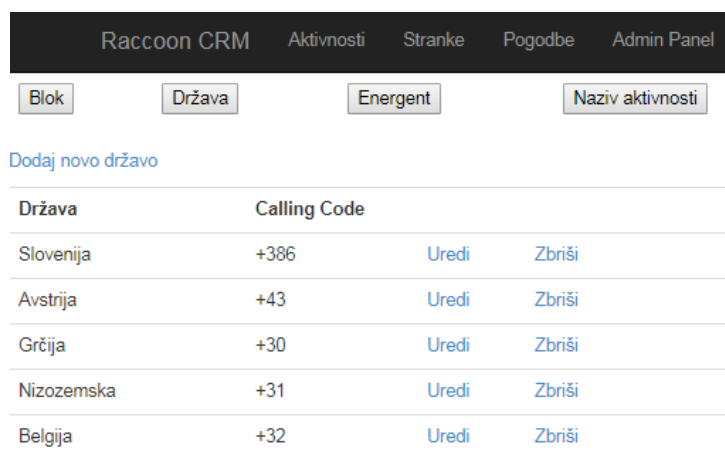
Uporabniki lahko dodajajo pogodbe, ki so sklenjene s podjetjem, ali pa pogodbe, ki so sklenjene z drugimi podjetji. Pogodbe, ki so sklenjene z drugim podjetjem, lahko uporabnik izpolni z vsemi znanimi podatki. S tem vidijo, do kdaj je neka stranka pri drugem dobavitelju, in lahko pred iztekom veljavnosti pogodbe začnejo pošiljati ponudbe tej stranki. Po zaključitvi izpolnjevanja podatkov uporabnik potrdi vnos z *Ustvari*, da se pogodba vnese v podatkovno bazo.

4.4.3.2 Urejanje in brisanje pogodb

Urejanje in brisanje pogodb deluje po istem principu kot pri *Aktivnosti*. V zadnjem stolpcu pri tabeli, kjer so pogodbe, sta povezavi *Uredi* in *Zbriši*. Vsaka povezava pripelje na novo stran, kjer uporabnik posodobi podatke o pogodbi ali pa izbriše pogodbo iz sistema. Pred brisanjem ali posodabljanjem je treba potrditi spremembe.

4.4.4 Nadzorna plošča

Stran nadzorna plošča je bila ustvarjena z namenom, da imajo skrbniki sistema nadzor nad podatki, ki se uporabljajo za filtriranje, aktivnosti in pogodbe. Ta stran vsebuje sezname predhodno definiranih podatkov. Za primer vzamemo seznam držav (Slika4.18), ki vsebuje vsa imena držav in njihove številke.



The screenshot shows a dashboard for 'Raccoon CRM' with navigation tabs for 'Aktivnosti', 'Stranke', 'Pogodbe', and 'Admin Panel'. Below the navigation are filter buttons for 'Blok', 'Država', 'Energent', and 'Naziv aktivnosti'. A link 'Dodaj novo državo' is present above a table. The table has two columns: 'Država' and 'Calling Code'. Each row contains a country name, its calling code, and two buttons: 'Uredi' and 'Zbriši'.

Država	Calling Code		
Slovenija	+386	Uredi	Zbriši
Avstrija	+43	Uredi	Zbriši
Grčija	+30	Uredi	Zbriši
Nizozemska	+31	Uredi	Zbriši
Belgija	+32	Uredi	Zbriši

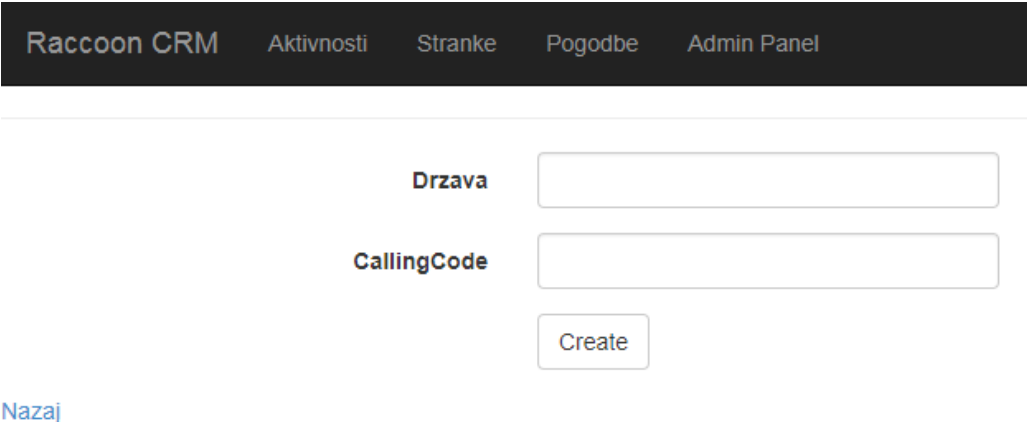
Slika 4.18: Izgled nadzorne plošče

Podatki so že definirani, da uporabniku ni treba na novo vpisovati teh informacij in jih na primer pri aktivnosti v spustnem seznamu samo izbere oziroma napiše v tekstovno polje želeno državo in mu v spustnem seznamu prikaže vse vrednosti, ki vsebujejo vpisani niz. Nazivi aktivnosti so tudi že predhodno definirani, da tudi uporabnikom ni treba vedno na novo vpisovati

vrednosti. Drugi razlog, zakaj je to tako narejeno, je, da pri filtriranju ne pride do težav, če bi na primer pri vpisu novih aktivnosti uporabniki sami vpisovali nazive aktivnosti in bi pri filtriranju prišlo do težav, ker bi se lahko uporabniki po pomoti zatipkali ali pa bi imeli dva ista naziva aktivnosti pod dvema različnima imenoma. Isto velja za vse ostale sezname, ki so prikazani na nadzorni plošči.

4.4.4.1 Dodajanje novih podatkov

Dodajanje novih podatkov izvedemo tako, da najprej izberemo seznam, ki ga hočemo dopolniti, s klikom na pripadajoči gumb seznama in se nam v tabeli prikažejo vsi podatki, ki so vsebovani pod to kategorijo. Z *Dodaj nov x*, kjer je x ime kategorije, se odpre nova stran, kjer vnesemo podatke (Slika 4.19).



The screenshot shows a dark navigation bar at the top with the following menu items: Raccoon CRM, Aktivnosti, Stranke, Pogodbe, and Admin Panel. Below the navigation bar, there is a form with two input fields. The first field is labeled 'Drzava' and the second is labeled 'CallingCode'. Below the second field is a 'Create' button. At the bottom left of the form area, there is a blue link labeled 'Nazaj'.

Slika 4.19: Dodajanje novih podatkov

Ko uporabnik vnese vse potrebne podatke (vsak seznam ima različno število podatkov, ki jih je treba izpolniti), mora na koncu potrditi vnos z *Ustvari*.

4.4.4.2 Urejanje podatkov

Za ureditev nekega vnosa je treba najprej izbrati kategorijo podatkov. Ko se nam prikažejo vsi podatki pod to kategorijo v tabeli, poiščemo vnos, ki ga hočemo popraviti, in kliknemo na *Uredi*. Odpre se nam stran za urejanje, kjer bo uporabnik vnesel želene popravke in potrdil spremembe s *Shrani* (Slika 4.20).

Raccoon CRM Aktivnosti Stranke Pogodbe Admin Panel

Država

Calling code

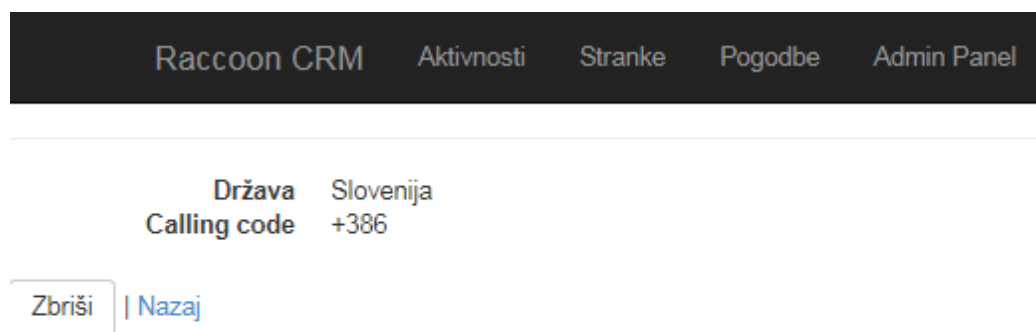
Create

[Nazaj](#)

Slika 4.20: Urejanje vnosov

4.4.4.3 Brisanje podatkov

Brisanje podatkov, ki jih ne želimo več, se izvede podobno kot za urejanje. Uporabnik mora najprej izbrati kategorijo, iz katere izvira vnos, ki ga hočemo izbrisati, in nato ob koncu vrstice, v kateri se nahaja vnos, kliknemo na povezavo *Izbriši*. Uporabnika bo preusmerilo na stran, kjer so izpisani vsi podatki o vnosu (Slika 4.21).



Slika 4.21: Brisanje vnosov

Ko uporabnik pregleda izpis podatkov in ugotovi, da hoče izbrisati pravi vnos, potrdi izbris z *Izbriši*. Lahko pa tudi pride do pomote in ugotovimo, da hočemo izbrisati napačen vnos, zato se samo vrnemo na prejšnjo stran z *Nazaj*.

Poglavje 5

Analiza

V tem poglavju bomo predstavili kakovost rešitve, koliko časa potrebuje aplikacija za obdelovanje podatkov, njen obseg in težave, ki so se pojavljale med razvijanjem. Končna različica je delujoča spletna aplikacija, ki jo lahko končni uporabniki že uporabljajo.

5.1 Čas razvoja

Za razvoj celotne spletne aplikacije smo potrebovali okoli devet tednov. Približno dva tedna smo porabili samo za zajem vseh funkcionalnosti, ki morajo biti vključene v sistem, en teden za načrtovanje podatkovne baze, štiri tedne za razvoj aplikacije in uporabniškega vmesnika v tehnologiji ASP.NET po arhitekturnem modelu MVC [12], dva tedna pa smo porabili za testiranje aplikacije in popravljanje napak, ki so se pojavile med testiranjem.

Celoten projekt je sestavljen iz štirih krmilnikov, štirinajst pogledov in osemnajst modelov. Celotne kode je približno 5000 vrstic in od tega je 2000 vrstic kod HTML in Razor skupaj, 1800 vrsti kode C# kode, 1000 vrstic kode JavaScript in približno 100 vrstic kode CSS.

5.2 Zmogljivost aplikacije

Ob testiranju aplikacije na strežniku se izkaže, da za naložitev strani *Aktivnosti*, *Pogodbe*, *Stranke* ali *Admin Panel* potrebuje nekje od 1,1 do 1,82 sekunde. Dodajanje novih vrednosti v sistem potrebuje 71 milisekund, brisanje vnosov porabi 952 milisekund, urejanje vnosov potrebuje 990 milisekund. Za filtriranje podatkov po tabeli potrebuje 79 milisekund in za sortiranje potrebuje 94 milisekund. Iskalnik pri strankah povprečno porabi 104 milisekunde za pridobitev seznama strank, prikaz informacij o stranki se izvede v 1,20 milisekunde. V meritvah je že upoštevan čas, ki potrebuje zahtevek, da pride od uporabnika do strežnika prek omrežja. Testiranje je bilo izvedeno na strežniku, ki je povezan v optično omrežje in vsebuje Intelov i7 procesor, dva trda diska v velikosti 1TB in dva 16GB DDR4 pomnilnika. Aplikacija v celoti deluje na vseh brskalnikih razen v Internet Explorerju. Internet Explorer predstavlja težave s samo hitrostjo aplikacije in tudi ne podpira nekaterih funkcij, ki jih ostali brskalniki imajo kot na primer izbira datumov s koledarjem.

5.3 Težave pri razvijanju projekta

Največji problem, ki se je pojavil pri razvijanju sistema, je bila implementacija mehanizma za preverjanje pristnosti. Rešitev za to je veliko, ampak je odvisno od uporabnikov oziroma podjetij, kakšno implementacijo hočejo. V diplomski nalogi ni opisanega mehanizma za preverjanje pristnosti, ker je ta del prepuščen podjetjem, da implementirajo lastne rešitve. Nekatera podjetja imajo že lasten mehanizem ali portal kjer ta rešitev že obstaja, in morajo le integrirati preverjanje pristnosti v novi sistem.

Poglavje 6

Sklepne ugotovitve

V okviru diplomske naloge smo razvili spletno aplikacijo, ki je lahko dostopna v spletnih brskalnikih na osebnih računalnikih in mobilnih telefonih ter postavitev podatkovne baze kjer hranimo podatke. Cilj diplomske naloge je bil razviti spletno aplikacijo, ki bi olajšala prodajo energentov končnim uporabnikom in preglednosti pri vodenju evidence strank. Predstavili smo vse funkcionalnosti, ki jih aplikacija obsega in opisali njena delovanja. Rezultat je delujoča spletna aplikacija, ki jo lahko podjetja implementirajo in uporabljajo.

Aplikacijo bi lahko z dodatnim vloženim delom še optimizirali in poskrbeli, da bi še hitreje delovala. Pohitrili bi lahko filtre pri iskanju strank tako, da bi se poslala poizvedba v podatkovno zbirko ko uporabnik zaključi z vpisovanjem niza. Trenutno je narejeno tako, da so vse stranke shranjene v en seznam in to zna predstavljati težavo pri veliki količini strank. Izboljšave bi lahko naredili tudi na samem grafičnem vmesniku in tudi na varnosti aplikacije. Grafični vmesnik bi lahko naredili bolj kompakten, da ne bi zasedoval celoten prostor.

Ob zaključku diplomske naloge smo pridobili veliko znanja in novih izkušenj, ki nam bodo pomagali v praksi in nadaljnjih projektih. Poglobili smo znanje o spletnih in novih tehnologijah, ki prihajajo ven in se nadgrajujejo.

6.1 Nadaljnje delo

Po zaključku testiranja in uporabljanja aplikacije smo dobili še veliko idej, kako bi lahko nadgradili sistem in dodali nove funkcionalnosti. Ena izmed nadgradenj bi bila integracija s spletno stranjo Ajpes, kjer hranijo javno dostopne podatke o podjetjih. S tem bi lahko pri vnosu novih strank avtomatično dopolnili vse podatke o novi stranki prek strani Ajpes, da potem uporabniku ni treba napisati vseh podatkov. Za to razširitev bi potrebovali dokumentacijo za API, če ga imajo razvitega, drugače bi morali razviti dodatne funkcije, ki bi pridobivale podatke direktno iz dokumentov HTML. Druga razširitev bi bila vključitev orodij za analitične namene. S temi orodji bi analitiki na podjetju dobili odgovore na analitična vprašanja, kot so na primer, kdo od zaposlenih je sklenil največ pogodb, kdo je najbolj aktiven s strankami, koliko dobička je kdo prinesel itd. V aplikacijo bi se dodalo novo stran, ki bi služila samo temu namenu in bi bila dostopna samo za analitike. Tukaj bi lahko imeli vnosna polja za izvajanje stavkov SQL in prikazano celotno strukturo podatkovne baze, ki bi jim olajšala iskanje podatkov. Tretja razširitev bi lahko bila vključitev sistema za pošiljanje ponudb po elektronski pošti strankam, katerim se pogodba v kratkem izteče. S tem bi uporabnikom prihranili čas pregledovanja po zbirki pogodb, katerim strankam v kratkem poteče veljavnost pogodb. Ta razširitev bi potrebovala postavitve poštnega strežnika. Uporabniki bi že vnaprej določili izgled ponudb ali pa imeli opcijo, da določenim strankam pošljejo drugačen izgled ponudbe. Trenutno je aplikacija narejena samo za osebne računalnike. Treba bo razviti tudi mobilno aplikacijo za uporabnike, ki gredo tudi na teren. Vmesnik bi bilo treba prilagoditi za male ekrane in vključiti samo funkcije, ki jih nujno potrebujejo, ko so na terenu. Poleg same mobilne aplikacije bi bilo treba razviti tudi spletno storitev ali API, ki bi olajšala komunikacijo med podatkovno bazo in mobilno napravo.

Literatura

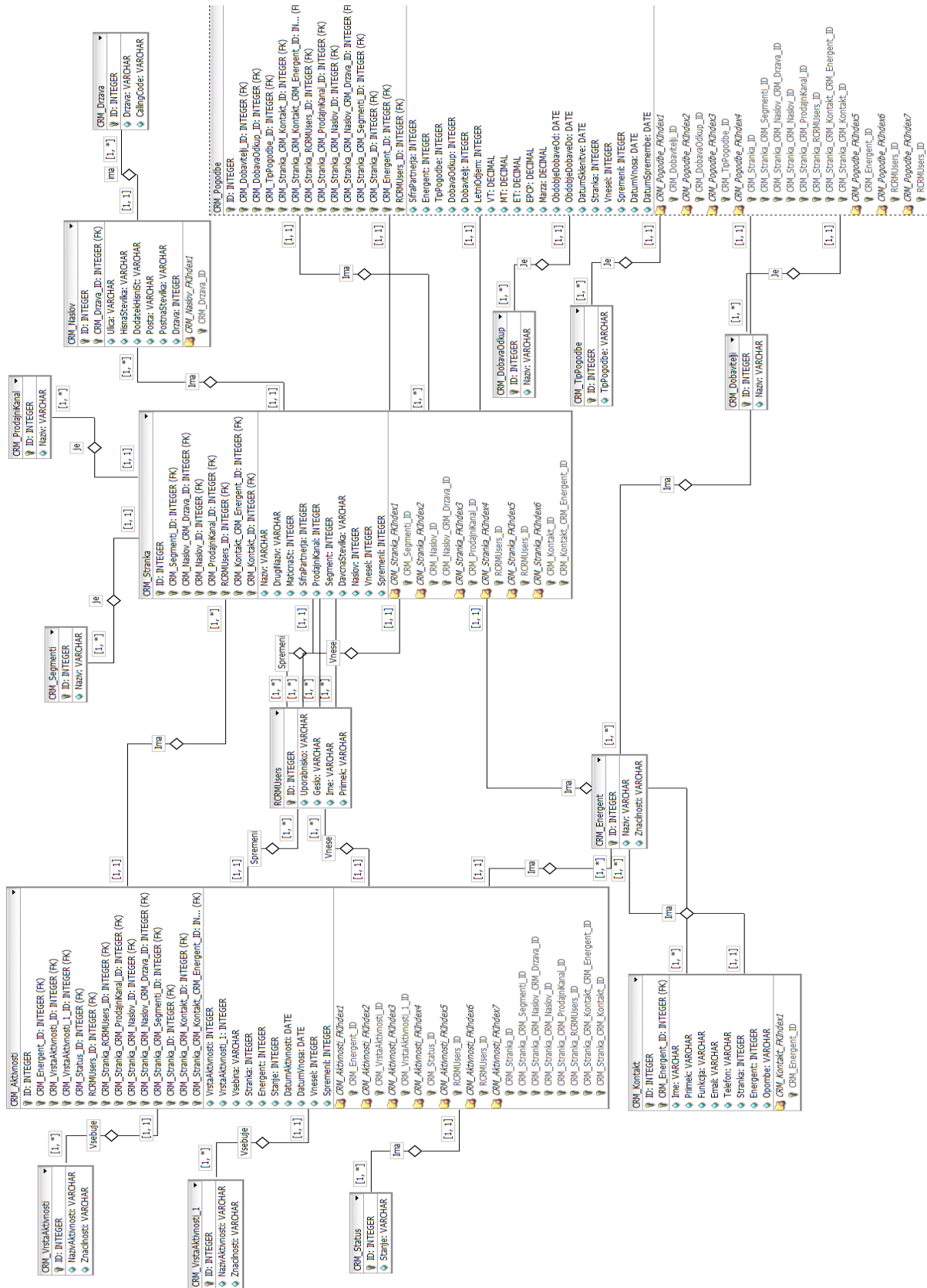
- [1] Ajax. Dosegljivo: [https://en.wikipedia.org/wiki/Ajax_\(programming\)](https://en.wikipedia.org/wiki/Ajax_(programming)). [Dostopano 26. 8. 2017].
- [2] Chrome developer tools. Dosegljivo: <https://developer.chrome.com/devtools>. [Dostopano 28. 8. 2017].
- [3] Css. Dosegljivo: <https://developer.mozilla.org/en-US/docs/Web/CSS>. [Dostopano 26. 8. 2017].
- [4] Customer relationship management. Dosegljivo: <http://searchcrm.techtarget.com/definition/CRM>. [Dostopano 30. 8. 2017].
- [5] Internet information services. Dosegljivo: <http://searchwindowsserver.techtarget.com/definition/IIS>. [Dostopano 28. 8. 2017].
- [6] jquery. Dosegljivo: <https://jquery.com/>. [Dostopano 26. 8. 2017].
- [7] Razor. Dosegljivo: https://www.w3schools.com/asp/razor_intro.asp. [Dostopano 26. 8. 2017].
- [8] Transact sql. Dosegljivo: <https://en.wikipedia.org/wiki/Transact-SQL>. [Dostopano 26. 8. 2017].
- [9] Visual studio 2013. Dosegljivo: <https://reviews.financesonline.com/p/microsoft-visual-studio/>. [Dostopano 28. 8. 2017].

- [10] Jamie Chan. *Learn C# in One Day and Learn It Well: C# for Beginners with Hands-on Project (Learn Coding Fast with Hands-On Project) (Volume 3)*. CreateSpace Independent Publishing Platform, 2015.
- [11] David Flanagan. *JavaScript: the definitive guide*. O'Reilly Media, Inc., 2006.
- [12] Jon Galloway, Brad Wilson, K. Scott Allen, and David Matson. *Professional ASP.NET MVC 5 1st Edition*. O'Reilly Media, Inc., 2012.
- [13] Jennifer Niederst Robbins. *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics 4th Edition*. Wrox, 2014.

Priloge

Dodaj novo aktivnost

Stranka	Emergent	Naziv Aktivnosti	Naziv Aktivnosti 1	Vrstei	Vsebina	Datum Aktivnosti	Datum Vnosa	Status
Podjetje 1	Zenerjski plin	Email	Poslana Ponudba	Uporabnik 2	ponudba za ZP 2018	08.11.2017	16.08.2017	V obdelavi Uredi Pobrabi Završi
Podjetje 2	Zenerjski plin	Email	Poslana Ponudba	Uporabnik 3	ponudba za ZP 2018	08.11.2017	16.08.2017	V obdelavi Uredi Pobrabi Završi
Podjetje 5	Elektrika	Klic	Kontaktiranje kupca	Uporabnik 3	ponovna ponudba	22.08.2017	17.08.2017	V obdelavi Uredi Pobrabi Završi
Podjetje 5	Elektrika	Obisk	Obisk kupca	Uporabnik 3	Sestanek	29.08.2017	18.08.2017	Novo Uredi Pobrabi Završi
Podjetje 3	Zenerjski plin	Email	Realizirana pogodba	Uporabnik 1	pošlj novo pogodbo za leto 2018	21.08.2017	18.08.2017	V obdelavi Uredi Pobrabi Završi
Podjetje 2	Zenerjski plin	Email	Realizirana pogodba	Uporabnik 1	prijava menjave obnavilca	21.08.2017	18.08.2017	V obdelavi Uredi Pobrabi Završi
Podjetje 6	Elektrika	Obisk	Pogajanja	Uporabnik 1		21.08.2017	18.08.2017	Končano Uredi Pobrabi Završi



Datum dobave od: [dd/mm/yyyy] do [dd/mm/yyyy]
 Datum sklenitve od: [dd/mm/yyyy] do [dd/mm/yyyy]
 Datum vnosa iz: [dd/mm/yyyy] do [dd/mm/yyyy]
 Datum spremembe od: [dd/mm/yyyy] do [dd/mm/yyyy]
 Število prikazanih vstic: 10 Search

Dodaj novo pogodbo

Šifra partnerja	Energent	Dobavitelj	Stranka	Vesel	Tip pogodbe	Dobava odkup	Letni odjem [MWh]	VT [€/MWh]	MT [€/MWh]	ET [€/MWh]	Meša [€/MWh]	EFCP [€/MWh]	Obdobje dobave od	Obdobje dobave do	Datum sklenitve	Datum vnosa	Datum spremembe	Spremenil	
xxxxx	Elektrika	Dobavitelj 1	Podjetje 2	Uporabnik 1	X	Y	Z				xxxx		01.01.2017	31.12.2017	12.12.2016			Uporabnik 1	Uredi Podrobnosti Zriši
xxxxx	Elektrika	Dobavitelj 2	Podjetje 2	Uporabnik 1	X	Y	Z						01.01.2018	31.12.2019				Uporabnik 1	Uredi Podrobnosti Zriši
xxxxx	Elektrika	Dobavitelj 3	Podjetje 5	Uporabnik 2	X	Y	Z						01.01.2018	31.12.2018		18.08.2017		Uporabnik 2	Uredi Podrobnosti Zriši
xxxxx	Elektrika	Dobavitelj 1	Podjetje 1	Uporabnik 2	X	Y	Z	xxxx	xxxx	xxxx			01.01.2018	31.12.2019	15.07.2017	18.08.2017	21.08.2017	Uporabnik 2	Uredi Podrobnosti Zriši
xxxxx	Elektrika	Dobavitelj 1	Podjetje 3	Uporabnik 2	X	Y	Z	xxxx	xxxx	xxxx			01.01.2018	31.12.2019	15.07.2017	18.08.2017		Uporabnik 2	Uredi Podrobnosti Zriši