Fakultet elektrotehnike i računarstva

Objektno oblikovanje

2. Domaća zadaća

NannyApp

Sustav za upravljanjem uslugama čuvanja djece

“Lako ćemo”

Zagreb, 14.01.2017.

# 

# **Uvod**

Prilikom izrade ovog zadatka koristit će se ASP.NET programski okvir uz obrazac oblikovanja model – pogled - upravitelj (engl. Model – View - Controller – MVC), razvojna okolina Visual Studio 2013, okvir za objektno relacijsko mapiranje naziva „NHibernate“, SQL Lite te programski jezici JavaScript, C# .

U daljnjem tekstu rada je detaljno opisan zadatak sa svim funkcionalnostima koje je potrebno izraditi. Također opisat ćemo tehnologije pomoću kojih će aplikacija biti razvijena.

# **Opis zadatka**

U sklopu seminara razviti ćemo sustav za upravljanje uslugama čuvanja djece. U današnjem užurbanom svijetu rijetko tko ima vremena upoznavati susjede, a većina obitelji ne živi s bakama i djedovima te je potrebno omogućiti roditeljima online svijet gdje na brzinu mogu pronaći adekvatnu uslugu čuvanja djece. Nadalje, takav način spajanja dadilja i roditelja pogodniji je i za dadilje jer im omogućava jednostavno i brzo oglašavanje svojih usluga. Velika prednost ponude i potražnje usluga čuvanja djece preko Interneta je što korisnicima štedi vrijeme koje bi bilo utrošeno na zvanje i provjeru dostupnosti dadilja ili oglašavanje i traženje roditelja kojima je potrebna usluga. Vidimo postojanje potrebe za ovakvim sustavom kao i smisao realizacije istoga.

Korisnici će sustavu pristupati preko desktop ili preko web aplikacije. Sustav će omogućavati istovremeni rad svih korisnika ovog sustava. Ideja je napraviti jednostavnu aplikaciju koju će biti moguće koristiti bez obzira na razinu informatičkog obrazovanja. To će se postići jedinstvenim, preglednim i intuitivnim sučeljem implementiranim za svakog dionika. Svi relevantni podaci o događajima i korisnicima će biti smješteni u bazi podataka.

Aplikacija se prilagođava za četiri vrste korisnika u sustavu: Neprijavljeni korisnik, Prijavljeni korisnik - Dadilja, Prijavljeni korisnik - Roditelj i Administrator.

**Neprijavljeni korisnik** pokretanjem aplikacije vidi početnu stranicu na kojoj se pruža mogućnost pregleda ponuda i potražnji usluga za čuvanje djece. Neprijavljeni korisnik može zatražiti registraciju ili se prijaviti u sustav ako je registriran. Ako je neprijavljeni korisnik odabrao opciju registriranja u sustav pojavit će mu se odabir nudi li uslugu čuvanja djece, odnosno želi li se registrirati u sustav kao dadilja ili mu je potrebna usluga čuvanja djece, odnosno želi li se registrirati u sustav kao roditelj. Nadalje, kod registracije će biti potrebno ispuniti polja za unos osobnih podataka, kontakt podataka te postavljanje korisničkog imena i lozinke. Ako se korisnik želi registrirati kao dadilja dodatno će trebati ispuniti polja s podacima o cijeni usluge, iskustvu i mjestu čuvanja djece (U domu obitelji korisnika usluge, u domu osobe koja pruža uslugu). Bez registracije, korisnicima će biti omogućeno pregledavanje ponuda i potražnji usluga za čuvanje djece, bez pristupa njihovim detaljima ili kontakt podacima.

**Prijavljeni korisnik** ima mogućnost pregleda ponuda i potražnji usluga čuvanja djece. Također može pregledavati i uređivati svoje podatke i preko izborne trake pristupiti akcijama koje su omogućene za korisnički dio aplikacije.

**Prijavljenom korisniku - Dadilja** se nakon prijave u sustav prikazuju sve potražnje, odnosno oglasi roditelja kojima je potrebna usluga čuvanja djece. Dodatne akcije koje su omogućene za dadilje su unos ponuda usluga čuvanja djece. Prilikom unosa ponude korisnik treba ispuniti polja s cijenom (koliko košta usluga), gradom gdje je moguće izvršavati uslugu, mjestu čuvanja djece (U domu obitelji korisnika usluge, u domu osobe koja pruža uslugu), svojom dostupnošću, kontaktom, iskustvom te dodatan opis ako ima napomena. Prijavljeni korisnik - Dadilja se može prijaviti na potražnje roditelja, pregledavati prijave na svoje ponude usluga čuvanja djece te ih prihvatiti ili odbiti.

**Prijavljenom korisniku - Roditelj** se nakon prijave u sustav prikazuju sve ponude, odnosno oglasi dadilja koje nude usluge čuvanja djece. Dodatne akcije koje su omogućene za roditelje su unos potražnji usluga čuvanja djece. Prilikom unosa potražnje korisnik treba ispuniti polja s cijenom koju su spremni platiti, gradom gdje je moguće izvršavati uslugu, mjestu čuvanja djece (U domu obitelji korisnika usluge, u domu osobe koja pruža uslugu), za koji period je potrebna usluga, rok do kada se potrebno prijaviti, kontaktom, potrebno iskustvo te dodatan opis ako ima napomena. Prijavljeni korisnik - Roditelj se može prijaviti na ponude dadilja, pregledavati prijave na svoje potražnje usluga čuvanja djece te ih prihvatiti ili odbiti.

**Administrator** ima pristup svim dijelovima sustava. Uvid u sve akcije sustava i omogućeno mu je ažuriranje svih podataka u bazi podataka.

# **Funkcionalni zahtjevi**

## Dionici sustava:

* Administrator
* Neprijavljeni korisnik
* Prijavljeni korisnik - Dadilja
* Prijavljeni korisnik - Roditelj
* Baza podataka

## Aktori i njihovo sudjelovanje u sustavu:

### Administrator, inicijator

* + Ima pristup svim dijelovima sustava
  + Uvid u sve akcije sustava i omogućeno mu je ažuriranje svih podataka baze podataka

### Neprijavljeni korisnik, inicijator

* + Može pregledavati ponude usluga čuvanja djece, bez detalja o kontakt podacima
  + Može pregledavati potražnje usluga čuvanja djece, bez detalja o kontakt podacima
  + Može zatražiti registraciju ili se prijaviti ako je registriran

### Prijavljeni korisnik - Dadilja, inicijator

* + Može pregledavati potražnje usluga čuvanja djece
  + Može pregledavati i uređivati svoje podatke
  + Može unositi ponude usluga čuvanja djece
  + Može se prijaviti na potražnje roditelja
  + Može pregledavati prijave na svoje ponude usluga čuvanja djece
  + Može prihvatiti ili odbiti prijavu na svoje ponude usluga čuvanja djece

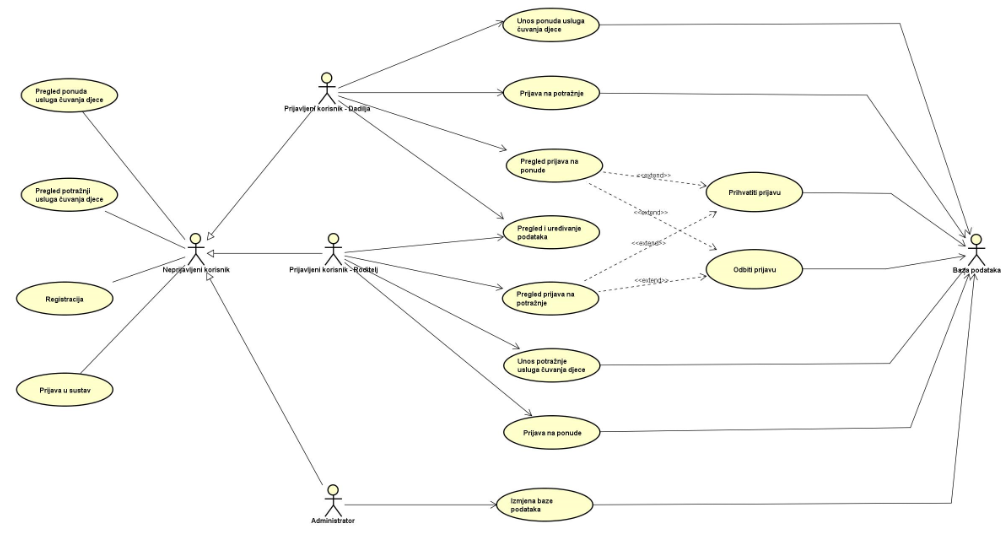
### Prijavljeni korisnik - Roditelj, inicijator

* + Može pregledavati ponude usluga čuvanja djece
  + Može pregledavati i uređivati svoje podatke
  + Može unositi potražnje za usluge čuvanja djece
  + Može se prijaviti na ponude dadilja
  + Može pregledavati prijave na svoje potražnje usluga čuvanja djece
  + Može prihvatiti ili odbiti prijavu na svoje potražnje usluga čuvanja djece

### Baza podataka, sudionik

* + Sprema podatke o korisnicima i događajima

# **Obrasci uporabe**



Slika 1 - Dijagram obrazaca uporabe

## OPIS OBRAZACA UPORABE

### UC1 - *PregledPonuda*

* + **Glavni sudionik**: Neprijavljeni korisnik.
  + **Cilj**: Pregled aktualnih ponuda usluga čuvanja djece.
  + **Sudionici**: Prijavljeni korisnik - Roditelj, Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja.
  + **Rezultat**: Prikaz aktualnih ponuda usluga čuvanja djece i dobivanje informacija.
  + **Željeni scenarij**:

1) Neprijavljeni korisnik šalje zahtjev poslužitelju za prikaz aktualnih ponuda usluga čuvanja djece.

2) Poslužitelj provjerava bazu podataka.

3) Poslužitelj prikazuje podatke neprijavljenom korisniku.

* + **Mogući drugi scenariji**: Ne postoje trenutno aktualne ponude usluga čuvanja djece, ispisuje se poruka „Nema aktualnih ponuda usluga čuvanja djece“.

### UC2 - *PregledPotražnji*

* + **Glavni sudionik**: Neprijavljeni korisnik.
  + **Cilj**: Pregled aktualnih potražnji usluga čuvanja djece.
  + **Sudionici**: Prijavljeni korisnik - Dadilja, Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja.
  + **Rezultat**: Prikaz aktualnih potražnji usluga čuvanja djece i dobivanje informacija.
  + **Željeni scenarij**:

1) Neprijavljeni korisnik šalje zahtjev poslužitelju za prikaz aktualnih potražnji usluga čuvanja djece.

2) Poslužitelj provjerava bazu podataka.

3) Poslužitelj prikazuje podatke neprijavljenom korisniku.

* + **Mogući drugi scenariji**: Ne postoje trenutno aktualne potražnje usluga čuvanja djece, ispisuje se poruka „Nema aktualnih potražnji usluga čuvanja djece“.

### UC3 - *Registracija*

* + **Glavni sudionik**: Neprijavljeni korisnik.
  + **Cilj**: Uspješna registracija i prijava korisnika u sustav.
  + **Sudionici**: Poslužitelj, baza podataka .
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja.
  + **Rezultat**: Korisnik je registriran u sustav. Prilikom registracije ispunjava obrazac sa osnovnim podatcima.
  + **Željeni scenarij**:

1) Korisnik koji još nije registriran unosi podatke u obrazac.

2) Poslužitelj provjerava jesu li svi podaci pravilno uneseni i postoji li u bazi podataka korisnik s istim podacima.

3) Korisnik je uspješno registriran u sustav.

* + **Mogući drugi scenariji**: Neuspješna prijava korisnika u sustav zbog krivo unesenih podataka ili postojanja korisnika s istim podacima.

### UC4 - *PrijavaUSustav*

* + **Glavni sudionik**: Neprijavljeni korisnik.
  + **Cilj**: Uspješna prijava korisnika u sustav .
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je registriran u sustavu.
  + **Rezultat**: Korisnik je uspješno prijavljen u sustav te može koristiti opcije sustava za koje ima dozvolu.
  + **Željeni scenarij**:

1) Korisnik sustava upisuje svoje korisničko ime i lozinku.

2) Poslužitelj u bazi podataka provjerava postoji li uređeni par korisničkog imena i lozinke.

3) Korisnik sustava uspješno je prijavljen u sustav.

* + **Mogući drugi scenariji**: Neuspješno spajanje korisnika na sustav jer uređeni par korisničkog imena i lozinke nisu pronađeni u bazi.

### UC5 - *UnosPonude*

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja.
  + **Cilj**: Uspješan unos ponude usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno unio ponudu usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Dadilja unosi podatke u obrazac za unos ponude usluge čuvanja djece.

2) Korisnik je uspješno unio ponudu usluge čuvanja djece.

* + **Mogući drugis cenariji**: Neuspješan unos ponude usluge čuvanja djece u sustav zbog greške na poslužitelju.

### UC6 - *PrijavaNaPotražnje*

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja.
  + **Cilj**: Uspješna prijava na potražnje usluga čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik se uspješno prijavio na potržnju usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Dadilja odabire prijavu na potražnju

2) Korisnik se uspješno prijavio.

* + **Mogući drugis cenariji**: Neuspješna prijava zbog greške na poslužitelju.

### UC7 - *PregledPrijavaNaPonude*

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja.
  + **Cilj**: Uspješan pregled prijava na ponudu usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno pregledao prijave na ponudu usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Dadilja odabire pregled prijava na ponudu usluge čuvanja djece

2) Korisniku se prikaže popis prijava.

* + **Mogući drugis cenariji**: Neuspješan pregled prijava zbog greške na poslužitelju

### UC8 - *PregledPrijavaNaPotražnje*

* + **Glavni sudionik**: Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješan pregled prijava na potražnje usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno pregledao prijave na potražnju usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Roditelj odabire pregled prijava na potražnju usluge čuvanja djece

2) Korisniku se prikaže popis prijava.

**Mogući drugis cenariji**: Neuspješan pregled prijava zbog greške na poslužitelju.

### UC9 - *PrihvatitiPrijavu*

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja, Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješno prihvaćanje prijava na ponudu ili potražnju usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno potvrdio prijavu na ponudu ili potražnju usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Dadilja ili Prijavljeni korisnik - Roditelj nakon pregleda prijava odabire prihvatiti prijavu na ponudu ili potražnju usluge čuvanja djece

2) Prijava je prihvaćena i ostvarena je suradnja između dadilje i roditelja

* + **Mogući drugis cenariji**: Neuspješno prihvaćanje prijava zbog greške na poslužitelju.

### UC10 - *OdbitiPrijavu*

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja, Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješno odbijanje prijava na ponudu ili potražnju usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno odbio prijavu na ponudu ili potražnju usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Dadilja ili Prijavljeni korisnik - Roditelj nakon pregleda prijava odabire odbiti prijavu na ponudu ili potražnju usluge čuvanja djece

2) Prijava je odbijena

* + **Mogući drugis cenariji**: Neuspješno odbijanje prijava zbog greške na poslužitelju.

### UC11 - PregledUređivanjePodataka

* + **Glavni sudionik**: Prijavljeni korisnik - Dadilja, Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješan pregled i uređivanje podataka korisnika.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno pregledao i/ili izmijenio svoje podatke.
  + **Željeni scenarij**:

1) Korisnik šalje poslužitelju zahtjev za pregledom i/ili uređivanjem podataka.

2) Korisnik je uspješno pregledao i/ili uredio svoje podatke.

* + **Mogući drugis cenariji**: Neuspješan pregled i/ili uređivanje podataka zbog greške na poslužitelju.

### UC12 - *UnosPotražnje*

* + **Glavni sudionik**: Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješan unos potražnje usluge čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik je uspješno unio potražnju usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Roditelj unosi podatke u obrazac za unos potražnje usluge čuvanja djece.

2) Korisnik je uspješno unio potražnju usluge čuvanja djece.

* + **Mogući drugis cenariji**: Neuspješan unos potražnje usluge čuvanja djece u sustav zbog greške na poslužitelju.

### UC13 - *PrijavaNaPonude*

* + **Glavni sudionik**: Prijavljeni korisnik - Roditelj.
  + **Cilj**: Uspješna prijava na ponude usluga čuvanja djece.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja, glavni sudionik je prijavljen u sustav.
  + **Rezultat**: Korisnik se uspješno prijavio na ponudu usluge čuvanja djece.
  + **Željeni scenarij**:

1) Prijavljeni korisnik - Roditelj odabire prijavu na ponudu

2) Korisnik se uspješno prijavio.

* + **Mogući drugis cenariji**: Neuspješna prijava zbog greške na poslužitelju.

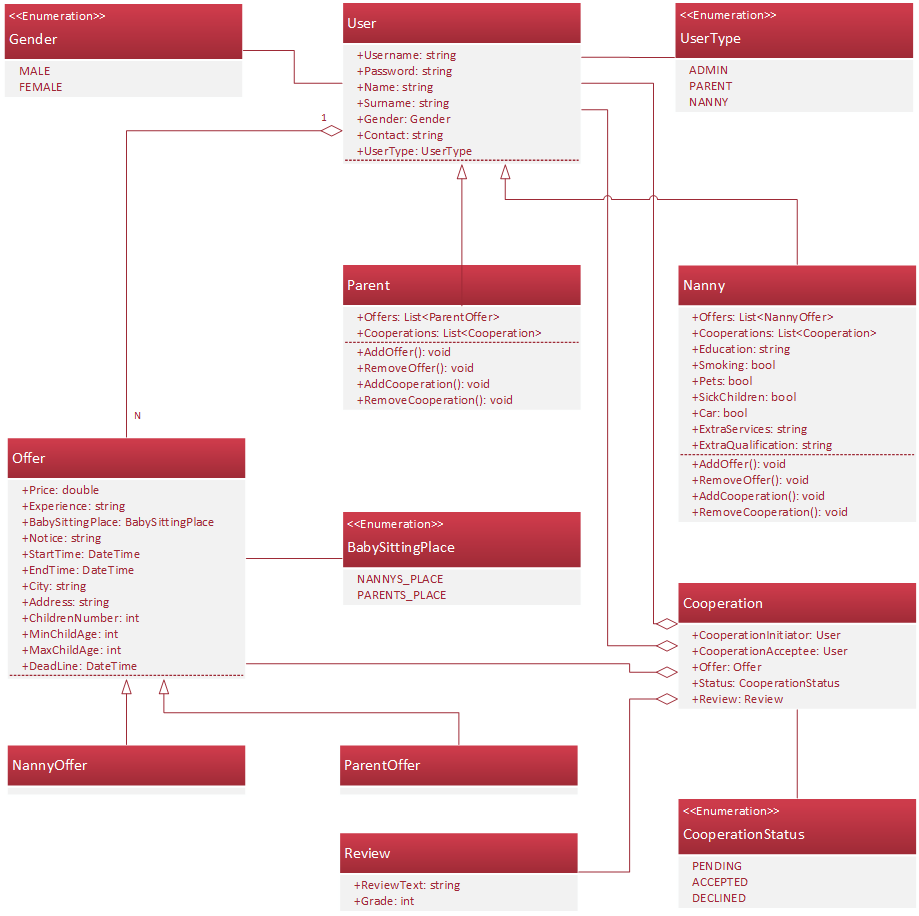
### UC14 - IzmjenaBazePodataka

* + **Glavni sudionik**: Administrator.
  + **Cilj**: Izmjeniti podatke u bazu.
  + **Sudionici**: Poslužitelj, baza podataka.
  + **Preduvjeti**: Mogući pristup Internetu, dostupnost poslužitelja.
  + **Rezultat**: Izmjenjeni su podaci u bazi.
  + **Željeni scenarij**:

1) Administrator mijenja podatke u bazi.

* + **Mogući drugi scenariji**: Ne postoji.

**Opis modela**



Slika 2 – dijagram razreda

Razredi entiteta u modelu su User, Offer, Cooperation i Review, a razreda vrijednosnih objekata nema. U modelu osim navedenih razreda postoje i repozitoriji CooperationRepository, OfferRepository i UserRepository, koji se bave dohvatom podataka o suradnjama, ponudama i korisnicima iz baze podataka, te tvornice CooperationFactory, OfferFactory i UserFactory koje se bave proizvodnjom odgovarajućih objekata suradnje, ponude i korisnika.

**Implementacija perzistencije**

Perzistencija je u projektu ostvarena pomoću objektno relacijskog mapera Nhibernate, alata za rad s istim Fluent Nhibernate i bazom podataka SQLite. Mapiranja se nalaze u datotekama CooperationMap.cs, OfferMap.cs, ReviewMap.cs i UserMap.cs. U mapiranju je određeno da će baza podataka sadržavati tablice: za suradnje, roditelje, dadilje, ponude roditelja, ponude dadilja i osvrte. U tablicama za roditelje i dadilje i za ponude roditelja i ponude dadilja je osiguran polimorfizam prilikom dohvata podataka iz baze naredbama UseUnionSubclassForInheritanceMapping() u nadklasama i Abstract() u podklasama, koje se načaze u datotekama OfferMap.cs (za ponude) i UserMap.cs (za korisnike).

Dohvaćanje i spremanje podataka odvija se na sljedeći način: prilikom dodavanja, dohvata, brisanja ili uređivanja zapisa iz baze, dohvaća se na početku izvođenja stvoreni objekt razreda Session u kojemu se onda otvara objekt razreda Transaction pomoću kojega se obavlja tražena radnja nad izabranim objektom. Nakon završetka radnje, nad objektom razreda Transaction se vrši metoda Commit() čime se sprema promjena koju je radnja obavila nad bazom.

# **Web aplikacija**

Web aplikacija korisniku kroz sučelje omogućava pregled podataka i provođenje raznih akcija nad tim podacima. U opisu zadatka spomenuto je kako postoje različite vrste korisnika, sukladno tome implementirani su različiti dijelovi aplikacije ovisni o ulozi korisnika.

**MVC (Model View Controller)** predstavlja oblikovni obrazac koji dijeli aplikaciju na tri glavne komponente odnosno sloja: Model, View (pogled) i Controller (kontroler). Osnovna ideja ovakvog pristupa je razdvajanje logike aplikacije od prezentacijskog dijela vidljivog korisniku.

**Model** je podatkovni sloj, implementira logiku domene podataka problema kojim se aplikacija bavi.Model je zadužen za komunikaciju s bazom podataka.

**View** je komponenta koja oblikuje korisničko sučelje, to je html predložak (.cshtml ekstenzija) koji se puni pomoću Razor stroja (eng. engine). Razor predstavlja programsku sintaksu koja omogućuje umatanje poslužiteljski baziranog koda (pisanog C# ili Visual Basic jezikom) unutar web stranica.

**Controller** koordinira korisničke interakcije, po potrebi komunicira s Model slojem i u konačnici koristeći View vraća korisničko sučelje nakon obavljene akcije. Controller-i predstavljaju skup razreda iz objektno orijentirane paradigme koji programski ostvaruju funkcionalnosti aplikacije. ASP.NET okvir podržava izradu Controller razreda u C# i VisualBasic programskom jeziku. Za izradu ovog rada korišten je C# jezik.

Interakcija korisnika s aplikacijom se provodi tako da korisnik poziva akcije klikom na različite objekte koji se nalaze u View-u te se putem HTML poveznica pozivaju različite metode koje su implementirane unutar odgovarajućih Controller-a.

Primjer poveznice je @Html.ActionLink("Tekst poveznice", "Metoda", "Controler") (Razor sintaksa).

## Model

Za upravljanje podacima koristili smo NHibernate i SQLite, što smo opisali u poglavlju opisa perzistencije. Podaci su organizirani na način opisan u poglavlju opisa modela

## Controller

Controller preuzima zahtjeve korisnika i izvršava potrebne radnje da bi odgovorio na zahtjev. Prilikom obrade zahtjeva controller šalje view-u model da prezentira rezultat korisniku.

* HomeController
  + Index()
* NannyController()
  + Index()
  + CreateNanny()
  + Delete()
  + Details()
  + Edit()
* ParentController()
  + Index()
  + CreateParent()
  + Delete()
  + Details()
  + Edit()
* NannyOfferController()
  + Index()
  + Create()
  + Delete()
  + Details()
  + Edit()
* ParentOfferController()
  + Index()
  + Create()
  + Delete()
  + Details()
  + Edit()
* UserController()
  + Index()
  + Login()
  + Logout()
* CooperationController
  + Index()
  + Details()
  + Create()

*HomeController* upravlja sadržajem početne stranice.

*NannyController* služi za implementaciju korisničkog dijela aplikacije. Omogućuje pregled i uređivanje osobnih podataka te registraciju novih dadilja.

*ParentController* služi za implementaciju korisničkog dijela aplikacije. Omogućuje pregled i uređivanje osobnih podataka te registraciju novih roditelja.

*NannyOfferController* omogućuje izradu, pregled i uređivanje ponuda za usluge čuvanja djece.

*ParentOfferController* omogućuje izradu, pregled i uređivanje potražnji za usluge čuvanja djece.

*UserController* služi za prijavu i odjavu korisnika.

*CooperationController* služi za prihvaćanje ili odbijanje ponuda, odnosno stvaranje suradnje između roditelja.

## View

Za navedene metode kontrolera implementiran je istoimeni View. Preko View-a se korisniku isporučuje rezultat zatražene akcije. Prilikom razvoja web aplikacije View datoteke su smještene u direktorije istoimene kontrolerima te ih iz tog razloga nisam detaljnije navodila u ovom poglavlju.

Dodatno, u direktoriju *Shared* nalazi se dijeljena View datoteka *\_mytemplate.cshtml* koja je zadužena za prikaz sadržaja, odnosno dizajna koji je zajednički za sve podstranice aplikacije.

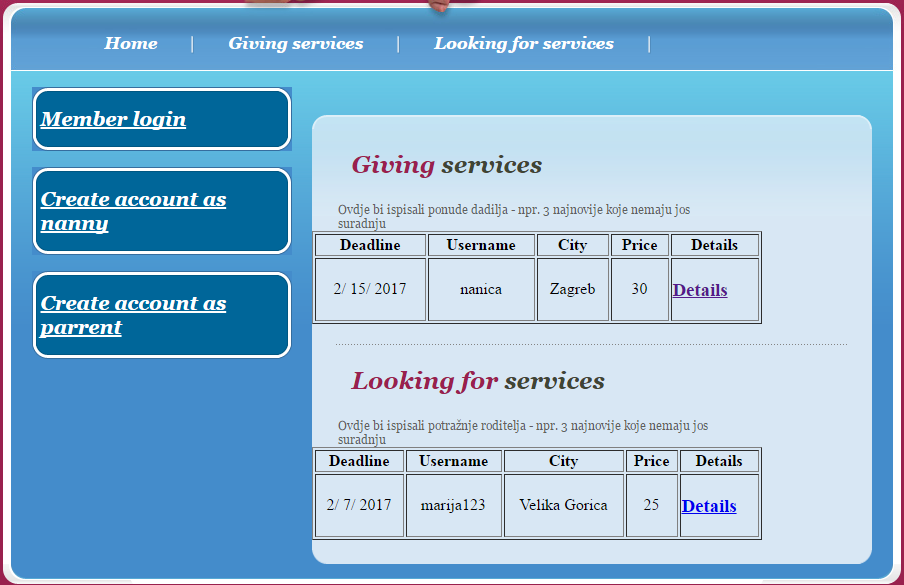
U View-u *\_mytemplate.cshtml* je definiran izgled izborne trake (eng. Menu bar) na vrhu stranice i izgled pozadine. Ovisno o vrijednosti varijable *Session([''Id''])* koja se svakom prijavljenom korisniku s u sustavu postavlja prilikom prijave u sustav, pojaviti će se dodatne poveznice.

## Dizajn

Vizualni dizajn aplikacije ostvaren je tako da su u direktorij Content te poddirektorij MyTemplate projekta aplikacije dodane .css datoteke s dizajnerskim predloškom, preuzete sa stranice <http://www.free-css.com/free-css-templates/page107/play-school>.

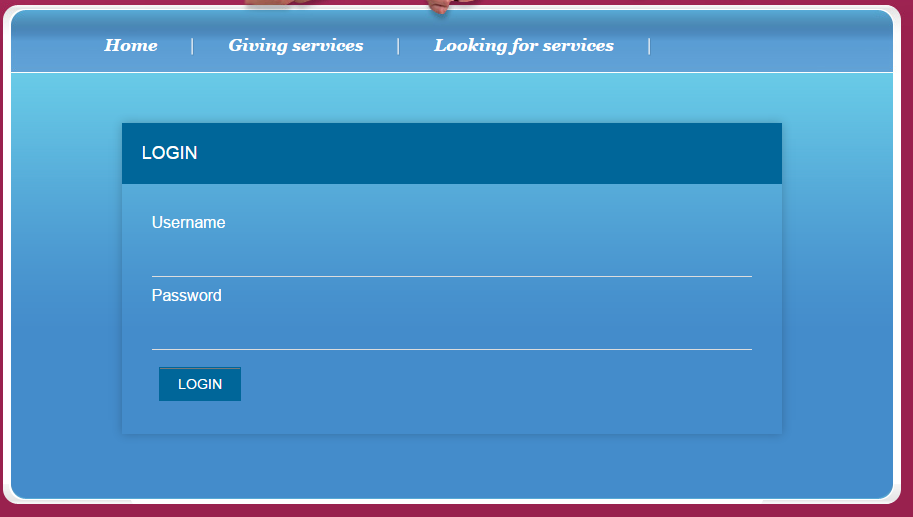
## Upute za korištenje

Najvažnije akcije web aplikacije korisnik može pronaći pri vrhu stranice na izbornoj traci (eng. Menu bar). Web aplikacija se sastoji odpočetne stranice i pod stranica koje se prikazuju ispod izborne trake nakon odabira željene akcije. Sadržaj izborne trake je različit za svaku vrstu korisnika. U nastavku ovog poglavlja slijede upute za svaku pojedinu vrstu korisnika.



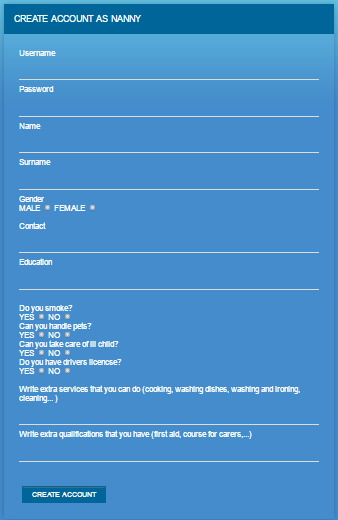
Slika 3 - Početna stranica

**Neprijavljeni korisnik** pokretanjem aplikacije vidi prikaza kao na slici 1. Vidljivo je kako Neprijavljeni korisnik ima s lijeve strane aplikacije mogućnost prijave (eng. Member login) ili stvaranja računa (eng. Create account). Može izabrati hoće li stvoriti račun kao dadilja ili kao roditelj. Dodatno na početnoj stranici se prikazuje nekoliko najnovijih ponuda i potražnji usluga čuvanja djece pod nazivima “Giving services” i ”Looking for services”.



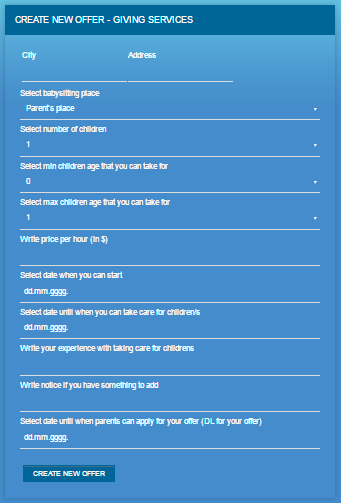
Slika 4 - Obrazac za prijavu u sustav

Klikom na „Login“ korisniku se na stranici pojave polja za unoskorisničkog imena i lozinke (Slika 2). Nakon što korisnik unese tražene podatke sustav će provjeriti postoji li korisnik s tim podacima u bazi podataka i ako postoji provjerit će jesu li lozinke identične. Kada su uneseni točni podaci korisnik će biti preusmjeren na svoj profil, gdje može nastaviti rad.



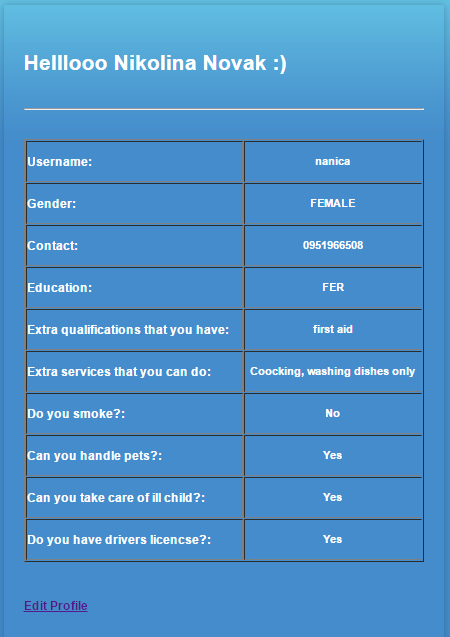
Slika 5 - Obrazac za registiranje korisnika u sustav

Ako korisnik nije registriran u sustav, klikom na „Create account as nanny“ otvorit će mu se obrazac za unos osobnih podataka i postavljanje lozinke (Slika 3). Obrazac za registriranje roditelja je ekvivalentan, samo s manje polja za unos.



Slika 6 - Obrazac za stvaranje nove ponude

Prijavljeni korisnik - Dadilja klikom na poveznicu “New offer” dobije prikaz kao na slici 4. Potrebno je ispuniti sva polja te klikom na gumb “Create new offer” šalje podatke kontroleru koji sprema novu ponudu u bazu. Ekvivalentan obrazas se prikaže i prijavljenom korisniku - Roditelj sa malo izmjenjenim nazivima polja.



Slika 7 - Profil prijavljenog korisnika - Dadilja

Prijavljeni korisnik - Dadilja klikom na poveznicu “Profile” dobije prikaz kao na slici 5. Može uređivati podatke klikom na poveznicu “Edit Profile”. Ako odabere poveznicu “Edit Profile” pojaviti će joj se obrazac kao i za kreiranje računa ali sa već postojećim podacima koje može urediti.

# **Desktop aplikacija**

Nad istim modelom domene korisnicima je, pored web aplikacije, omogućeno i sučelje putem Windows forms desktop aplikacije. Aplikacija, uz neke prilagodbe, prati sučelje web aplikacije te nudi jednak skup akcija i operacija nad podacima. Priikom pokretanja korisniku je prikazano naslovno sučelje, a za daljnji rad se omora prijaviti ili kreirati račun kao tip korisnika – roditelj ili dadilja. Ulogirani korisnici mogu pregledavati i prijavljivati se na ponude usluga, pregledavati profile drugih korisnika, ostvarivati suradnje i pisati recenzije nakon završetka usluge što se prati i prikazuje kao pomoć pri odabiru ponuđača neke usluga.

Aplikacija je razvijena koristeći C# programski jezik i Microsoftovu .NET platformu. Arhitektura je izvedena prema Model View Presenter oblikovnom uzorku koji definira podjelu sustava na tri dijela: Model, Prezenter i View, gdje Model definira podatke koji će se prikazzivati korisniku i nad kojima će se vršiti neke radnje, Prezenter obavlja operacije nad modelom i viewom, dohvaća podatke iz repozitorija i formatira ih za prikaz u view-ima, dok je view pasivno sučelje koje prikazuje podatke te prihvaća i preusmjerava korisnikove akcije prezenteru koji ih onda izvršava nad podacima.

Takva slojevita podjela implementirana je skupom sučelja za upravljanje podacima, sučelja za definiranje view razreda te kontrolera.

Sučelja za rad s podacima:

IUserRepository

IOfferRepository

ICooperationRepository

Sučelja koja definiraju viewove:

ICooperationView

ICooperationsView

ILoginView

IMainView

INannyView

IOfferView

IParentView

IReview

IReviews

IStartView

Controlleri:

AccountController

CooperationController

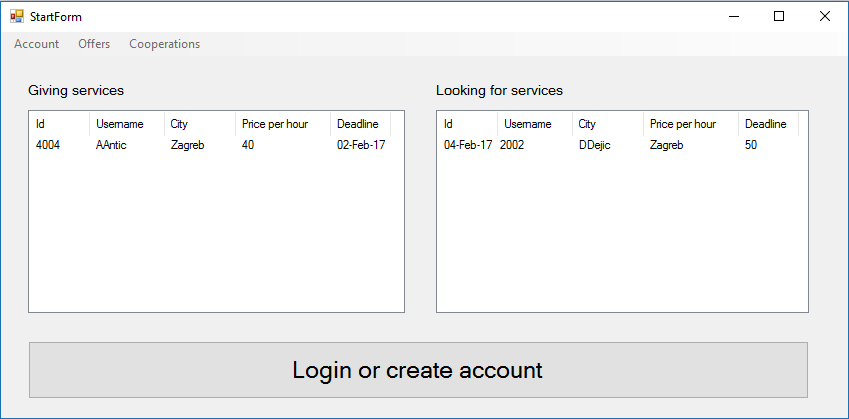
MainFormController

NannyController

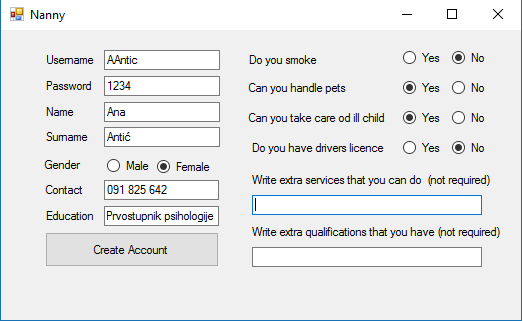
ParentController

ReviewController

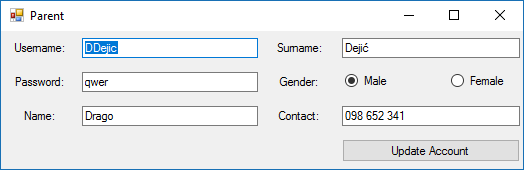
Tipičan tok rada aplikacije je da korisnik pokrene neku akciju pritiskom na neku opciju na formi. Forma to dalje proslijeđuje do MainFormControllera koji raspoređuje pristigle zahtjece odgovarajućim kontrolerima te pritom instancira odgovarajuće view objekte koji se stvaraju u tvornici opisanoj sučeljem IwindowsFormFactory. Svaka forma koja se može prikazati na ekranu implementira jedno od navedenih view sučelja koja uglavnom propisuju metode za postavljanje i dohvaćanje vrijednosti koje se prikazuju i uređuju, uz pokoju dodatnu funkciju za uređivanje prikaza. AccountController obavlja akcije vezane uz prijavu korisnika u sustav, Parent i Nanny kontroleri izvršavaju operacije vezane uz prikaz dadilja i roditelja, a analogno kontroleri OfferController, CooperationControler i ReviewController ostvaruju funkcionalnosti iz područja vezana za postavljanje ponuda, ostvarivanje i prikaz suradnji i korisničkih recenzija. Implementirane forme su prikazane na sljedećim slikama.



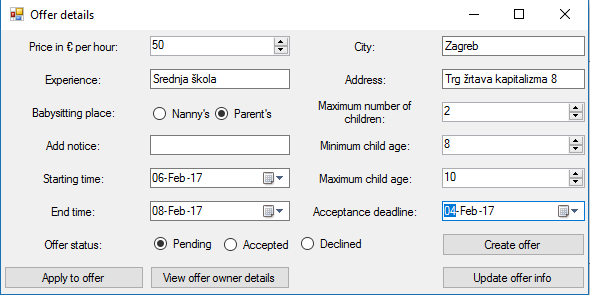
Slika 8 – glavna forma



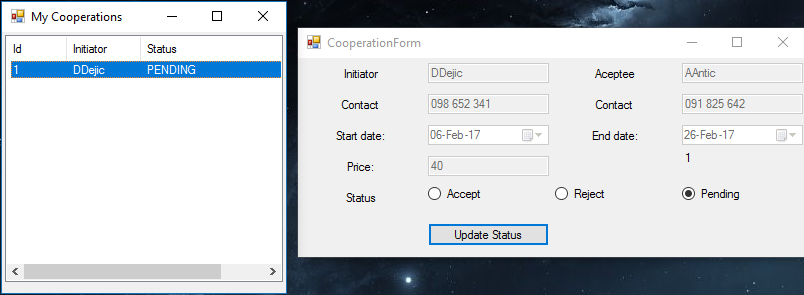
Slika 9 – forma s podacima o dadilji



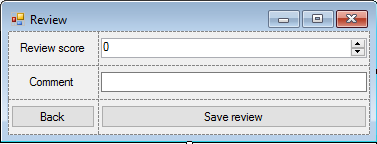
Slika 10 – forma s podacima o roditelju



Slika 11 – forma s podacima o ponudi



Slika 12 – forma s popisom suradnji i forma s podacima o suradnji



Slika 13 – forma s pregledom suradnje