



ITS INFORMATION
TECHNOLOGY
SCHOOL

VISOKA ŠKOLA STRUKOVNIH STUDIJA ZA IT

Projektni zadatak

Tehnička dokumentacija

Beograd
Mart , 2018.

Sadržaj

- 1. Uvod**
- 2. Arhitektura Rešenja**
 - 2.1 Korišćeni Alati**
- 3. Dijagrami**
 - 3.1 Proširen Model Objekti Veze**
 - 3.2 Use-Case Dijagrami**
 - 3.3 Dijagram Klasa**
- 4. Modeli(Klase)**
- 5. Kontrole za upravljanje**
- 6. ViewModeli**

1. Uvod

Prvenstveno je kreiran Admin nalog sa najvećim pravom pristupa, pa nakon njega editor i na kraju sam user, odnosno viewer.

Svaki korisnik ima svoj prikaz interfejsa preko kojeg izvršava komunikaciju sa bazom podataka.

U daljem nastavku korisničke specifikacije su dati vizuelni primeri PMOV-a, use-case dijagrama, dijagrama klasa, kontrolera i modela u samoj aplikaciji.

Nalozi:

Šifra:

Admin@admin.com , [Sifra1!](#)

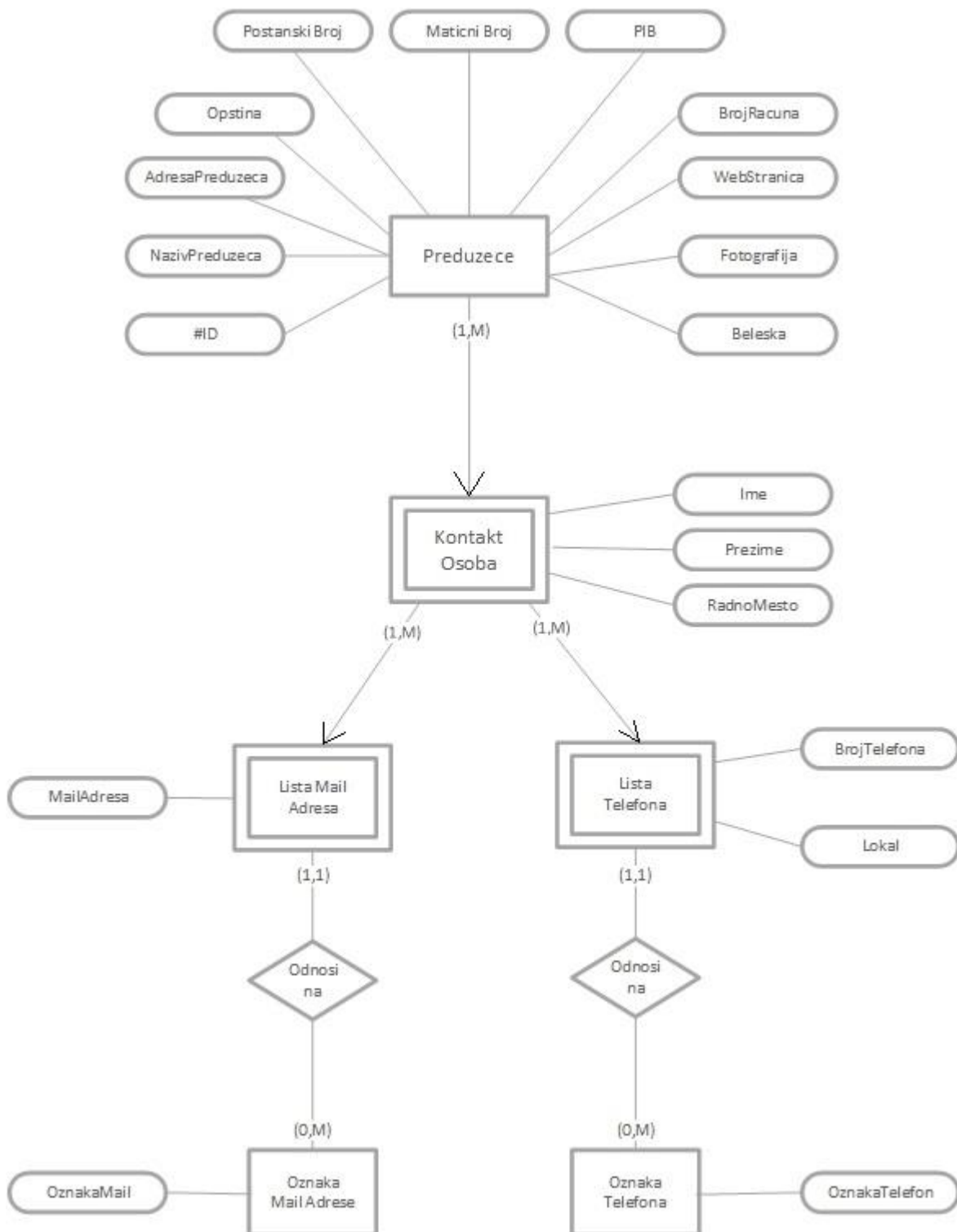
Editor@editor.com , [Sifra2!](#)

Viewer@viewer.com , [Sifra3!](#)

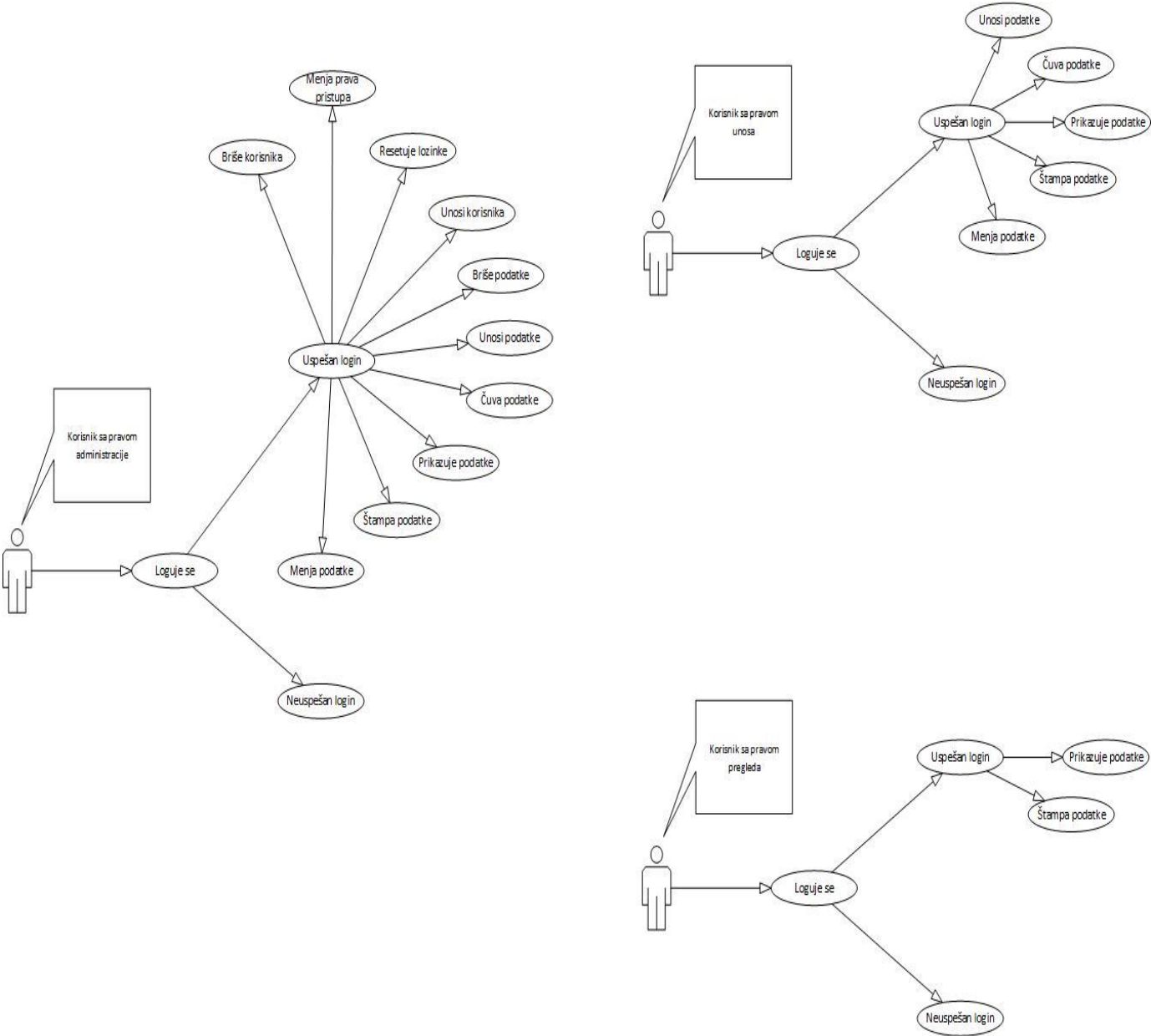
2.1 Korišćeni Alati

Aplikacija je realizovana kao web site, korišćenjem ASP.NET tehnologije na osnovu zadate specifikacije.

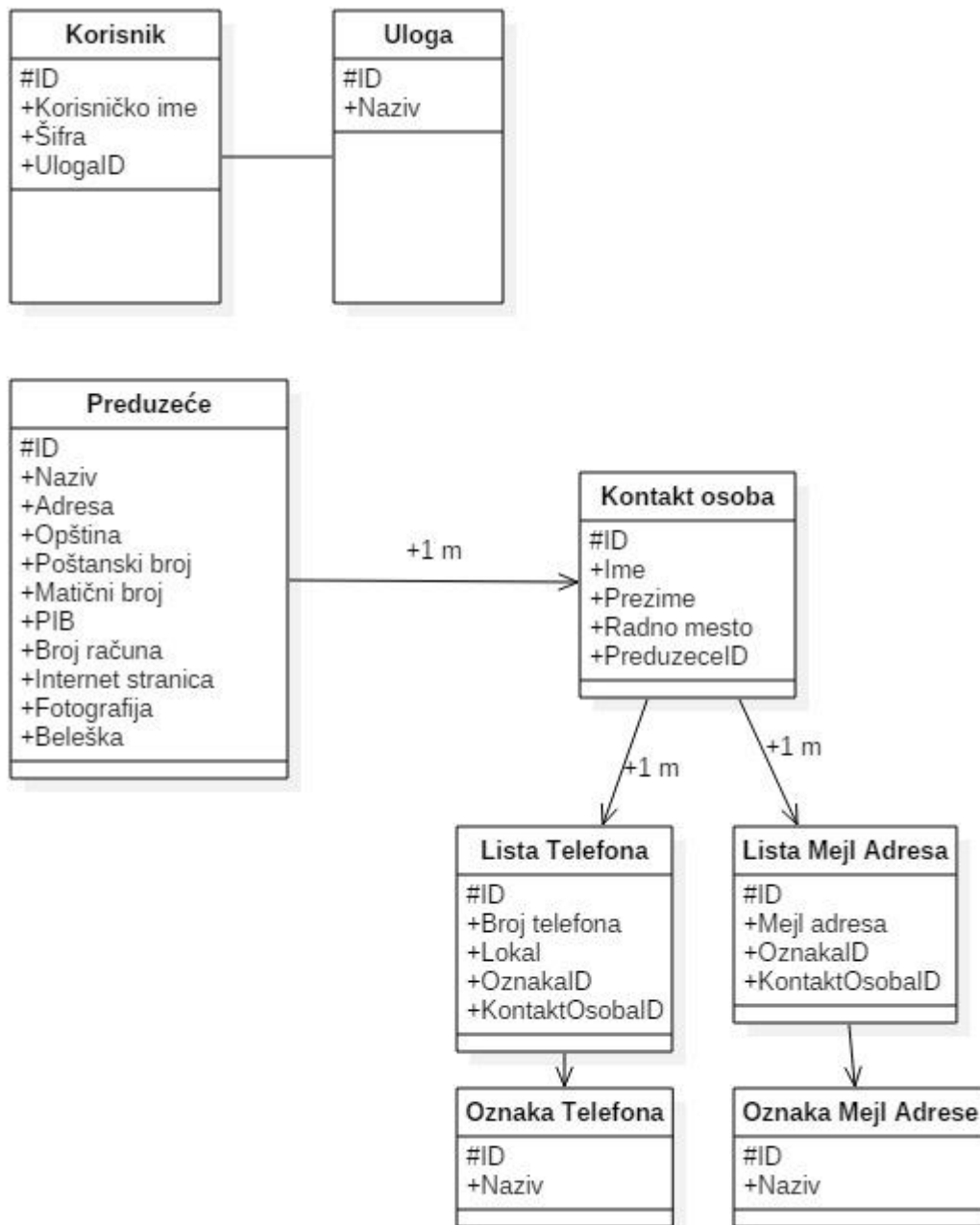
3.1 Proširen Model Objekti Veze



3.2 Use-Case Dijagrami



3.3 Dijagram Klasa



4. Modeli (Klase)

-Preduzece

```
namespace ProjektniCentar1.Models
{
    public class Preduzece
    {
        public int Id { get; set; }
        public string NazivPreduzeca { get; set; }
        public string AdresaPreduzeca { get; set; }
        public string Opstina { get; set; }
        public string PostanskiBroj { get; set; }
        public string MaticniBroj { get; set; }
        public string PIB { get; set; }
        public string BrojRacuna { get; set; }
        public string WebStranica { get; set; }
        public string Fotografija { get; set; }
        public string Beleske { get; set; }
        public List<KontaktOsoba> Osobe { get; set; }
    }
}
```

-Kontakt Osobe

```
namespace ProjektniCentar1.Models
{
    public class KontaktOsoba
    {
        public int Id { get; set; }
        public string Ime { get; set; }
        public string Prezime { get; set; }
        public string RadnoMesto { get; set; }
        public Preduzece Preduzece { get; set; }
        public int PreduzeceId { get; set; }
        public List<ListaMailAdrese> ListaMejlova { get; set; }
        public List<ListaTelefon> ListaTelefona { get; set; }
    }
}
```

-Lista Mail Adresa

```
namespace ProjektniCentar1.Models
{
    public class ListaMailAdrese
    {
        public int Id { get; set; }
        public String MailAdresa { get; set; }
        public OznakaMailAdrese OznakaMailAdrese { get; set; }
        public int OznakaMailAdreseId { get; set; }
        public KontaktOsoba KontaktOsoba { get; set; }
        public int KontaktOsobaId { get; set; }
    }
}
```

-Lista Telefona

```
namespace ProjektniCentar1.Models
{
    public class ListaTelefon
    {
        public int Id { get; set; }
        public string Lokal { get; set; }
        public string BrojTelefona { get; set; }
        public OznakaTelefon OznakaTelefon { get; set; }
        public int OznakaTelefonId { get; set; }
        public KontaktOsoba KontaktOsoba { get; set; }
        public int KontaktOsobaId { get; set; }
    }
}
```

-Oznaka Mail Adrese

```
namespace ProjektniCentar1.Models
{
    public class OznakaMailAdrese
    {
        public int Id { get; set; }
        public string Naziv { get; set; }
    }
}
```

-Oznaka Telefona

```
namespace ProjektniCentar1.Models
{
    public class OznakaTelefon
    {
        public int Id { get; set; }
        public string Naziv { get; set; }
    }
}
```


5. Kontrole za upravljanje

-Preduzece Kontroler

```
namespace ProjektniCentar1.Controllers
{
    [Authorize]
    public class PreduzeceController : Controller
    {
        private ApplicationDbContext _context;

        public PreduzeceController()
        {
            _context = new ApplicationDbContext();
        }
        protected override void Dispose(bool disposing)
        {
            _context.Dispose();
        }
        // GET: Preduzece
        public ActionResult Index()
        {
            var preduzeca = _context.Preduzeca.ToList();
            return View(preduzeca);
        }
        //GET: Preduzece/Detalji
        public ActionResult Detalji(int id)
        {
            var model = new ViewModels.KontaktOsobaViewModel();
            model.Preduzeca = _context.Preduzeca.Include(k => k.Osobe).SingleOrDefault(
p => p.Id == id);
            model.Osoba=_context.KontaktOsobe.Include(p=>p.Preduzece).Where(p => p.Pred
uzeceId == id).ToList();

            // var preduzeca = _context.Preduzeca.SingleOrDefault(p => p.Id == id);
            return View(model);
        }
        [Authorize(Roles = "Admin , Editor")]
        public ActionResult NovoPreduzece()
        {
            //var kontakOsobe = _context.KontaktOsobe.ToList();
            return View();
        }
        [Authorize(Roles = "Admin , Editor")]
        [HttpPost]
        public ActionResult PreduzeceCreate(Preduzece preduzece)
        {
            if (preduzece.Id == 0)
            {
                _context.Preduzeca.Add(preduzece);
            }
            else
            {
                var preduzeceBaza = _context.Preduzeca.Single(p => p.Id == preduzece.Id
);
                TryUpdateModel(preduzeceBaza);
            }
        }
    }
}
```

```

        }
        _context.SaveChanges();
        return RedirectToAction("Index", "Preduzece");
    }
    [Authorize(Roles = "Admin , Editor")]
    public ActionResult Edit(int id)
    {
        var preduzece = _context.Preduzeca.SingleOrDefault(p => p.Id == id);
        if (preduzece == null)
            return HttpNotFound();
        return View("NovoPreduzece" , preduzece);
    }

    [Authorize(Roles = "Admin")]
    public ActionResult izbrisiPreduzece(int id)
    {
        var itemToRemove = _context.Preduzeca.SingleOrDefault(x => x.Id == id); //r
        eturns a single item.

        if (itemToRemove != null)
        {
            _context.Preduzeca.Remove(itemToRemove);
            _context.SaveChanges();
        }

        return RedirectToAction("Index");
    }
}

```

-Kontakt Osoba Kontroler

```

namespace ProjektniCentar1.Controllers
{
    [Authorize]
    public class KontaktOsobaController : Controller
    {
        private ApplicationDbContext _context;

        public KontaktOsobaController()
        {
            _context = new ApplicationDbContext();
        }
        protected override void Dispose(bool disposing)
        {
            _context.Dispose();
        }
        // GET: KontaktOsoba
        public ActionResult Index()
        {
            var osobe = _context.KontaktOsobe.OrderByDescending(s => s.Preduzece.NazivP
            reduzeca).Include(k => k.Preduzece).ToList();
            return View(osobe);
        }

        [Authorize(Roles = "Admin , Editor")]
        // GET: KontaktOsoba/NovaKontaktOsoba
        public ActionResult NovaKontaktOsoba()
    }
}

```

```

    {
        NovaOsobaViewModel podaci = new NovaOsobaViewModel();

        podaci.Preduzeca = _context.Preduzeca.ToList();
        podaci.OznakeMaila = _context.OznakaMailAdresa.ToList();
        podaci.OznakeTelefona = _context.OznakaTelefona.ToList();

        return View("NovaKontaktOsoba", podaci);
    }

    public ActionResult Detalji(int id, int preduzeceid)
    {
        var model = new ViewModels.OsobeViewModel();

        List<ListaMailAdrese> mejlovi = _context.ListaMailAdresa.Where(p => p.KontaktOsobaId == id).ToList();
        List<OznakaMailAdrese> tempOznake = new List<OznakaMailAdrese>();
        List<OznakaMailAdrese> oznake = _context.OznakaMailAdresa.ToList();

        List<ListaTelefon> telefoni = _context.ListaTelefona.Where(p => p.KontaktOsobaId == id).ToList();
        List<OznakaTelefon> tempOznakeT = new List<OznakaTelefon>();
        List<OznakaTelefon> oznakeT = _context.OznakaTelefona.ToList();

        model.Preduzeca = _context.Preduzeca.Where(p => p.Id == preduzeceid).ToList();
        model.Osoba = _context.KontaktOsobe.SingleOrDefault(o => o.Id == id);
        model.ListaMailAdresa = _context.ListaMailAdresa.Where(p => p.KontaktOsobaId == id).ToList();
        model.ListaTelefona = _context.ListaTelefona.Where(p => p.KontaktOsobaId == id).ToList();

        foreach (var mejl in mejlovi)
        {
            tempOznake.Add(oznake.Where(o => o.Id == mejl.OznakaMailAdreseId).SingleOrDefault());
        }
        model.OznakeMailAdresa = tempOznake;

        foreach (var telefon in telefoni)
        {
            tempOznakeT.Add(oznakeT.Where(o => o.Id == telefon.OznakaTelefonId).SingleOrDefault());
        }
        model.OznakeTelefona = tempOznakeT;

        return View(model);
    }
    [Authorize(Roles = "Admin , Editor")]
    [HttpPost]
    public ActionResult KontaktOsobaCreate(NovaOsobaViewModel novaOsoba)
    {
        if (novaOsoba.KontaktOsoba.Id == 0)
        {

```

```

        _context.KontaktOsobe.Add(novaOsoba.KontaktOsoba);
        _context.ListaMailAdresa.Add(novaOsoba.ListaMailAdresa);
        _context.ListaTelefona.Add(novaOsoba.ListaTelefona);
    }
    else
    {
        var model = new ViewModels.NovaOsobaViewModel();

        model.KontaktOsoba = _context.KontaktOsobe.Single(o => o.Id == novaOsoba.KontaktOsoba.Id);
        model.ListaMailAdresa = _context.ListaMailAdresa.SingleOrDefault(o => o.KontaktOsobaId == novaOsoba.KontaktOsoba.Id);
        model.ListaTelefona = _context.ListaTelefona.SingleOrDefault(p => p.KontaktOsobaId == novaOsoba.KontaktOsoba.Id);

        TryUpdateModel(model);

    }
    _context.SaveChanges();

    return RedirectToAction("Index", "KontaktOsoba");

}
[Authorize(Roles = "Admin , Editor")]
public ActionResult Edit(int Id)
{
    NovaOsobaViewModel podaci = new NovaOsobaViewModel();

    podaci.Preduzeca = _context.Preduzeca.ToList();
    podaci.OznakeMails = _context.OznakeMailAdresa.ToList();
    podaci.OznakeTelefona = _context.OznakeTelefona.ToList();

    podaci.KontaktOsoba = _context.KontaktOsobe.SingleOrDefault(o => o.Id == Id);
    podaci.ListaMailAdresa = _context.ListaMailAdresa.FirstOrDefault(o => o.KontaktOsobaId == podaci.KontaktOsoba.Id);
    podaci.ListaTelefona = _context.ListaTelefona.FirstOrDefault(p => p.KontaktOsobaId == podaci.KontaktOsoba.Id);

    return View("NovaKontaktOsoba", podaci);
}

[Authorize(Roles = "Admin")]
public ActionResult izbrisiOsobu(int id)
{
    var itemToRemove = _context.KontaktOsobe.SingleOrDefault(x => x.Id == id);
    //returns a single item.

    if (itemToRemove != null)
    {
        _context.KontaktOsobe.Remove(itemToRemove);
        _context.SaveChanges();
    }

    return RedirectToAction("Index");
}
}

```

6. View Modeli

-KontaktOsobaViewModel

```
namespace ProjektniCentar1.ViewModels
{
    public class KontaktOsobaViewModel
    {
        public Preduzece Preduzeca { get; set; }
        public IEnumerable<KontaktOsoba> Osoba { get; set; }
        public IEnumerable<ListaMailAdrese> ListaMailAdresa { get; set; }
        public IEnumerable<ListaTelefon> ListaTelefona { get; set; }
        public OznakaMailAdrese OznakeMailAdrese { get; set; }
        public OznakaTelefon OznakeTelefona { get; set; }
    }
}
```

-NovaOsobaViewModel

```
namespace ProjektniCentar1.ViewModels
{
    public class NovaOsobaViewModel
    {
        public IEnumerable<Preduzece> Preduzeca { get; set; }
        public KontaktOsoba KontaktOsoba { get; set; }
        public ListaMailAdrese ListaMailAdresa { get; set; }
        public ListaTelefon ListaTelefona { get; set; }
        public IEnumerable<OznakaMailAdrese> OznakeMaila { get; set; }
        public IEnumerable<OznakaTelefon> OznakeTelefona { get; set; }
    }
}
```

-OsobeViewModel

```
namespace ProjektniCentar1.ViewModels
{
    public class OsobeViewModel
    {
        public IEnumerable<Preduzece> Preduzeca { get; set; }
        public KontaktOsoba Osoba { get; set; }
        public IEnumerable<ListaMailAdrese> ListaMailAdresa { get; set; }
        public IEnumerable<ListaTelefon> ListaTelefona { get; set; }
        public IEnumerable<OznakaMailAdrese> OznakeMailAdresa { get; set; }
        public IEnumerable<OznakaTelefon> OznakeTelefona { get; set; }
    }
}
```