FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Objektno oblikovanje

**Seminar uz projekt eDnevnik**

Bruna Anđelić 0036478714

Josipa Kelava 0036478602

Luka Kovačić 0036479914

Tea Pović 0036476912

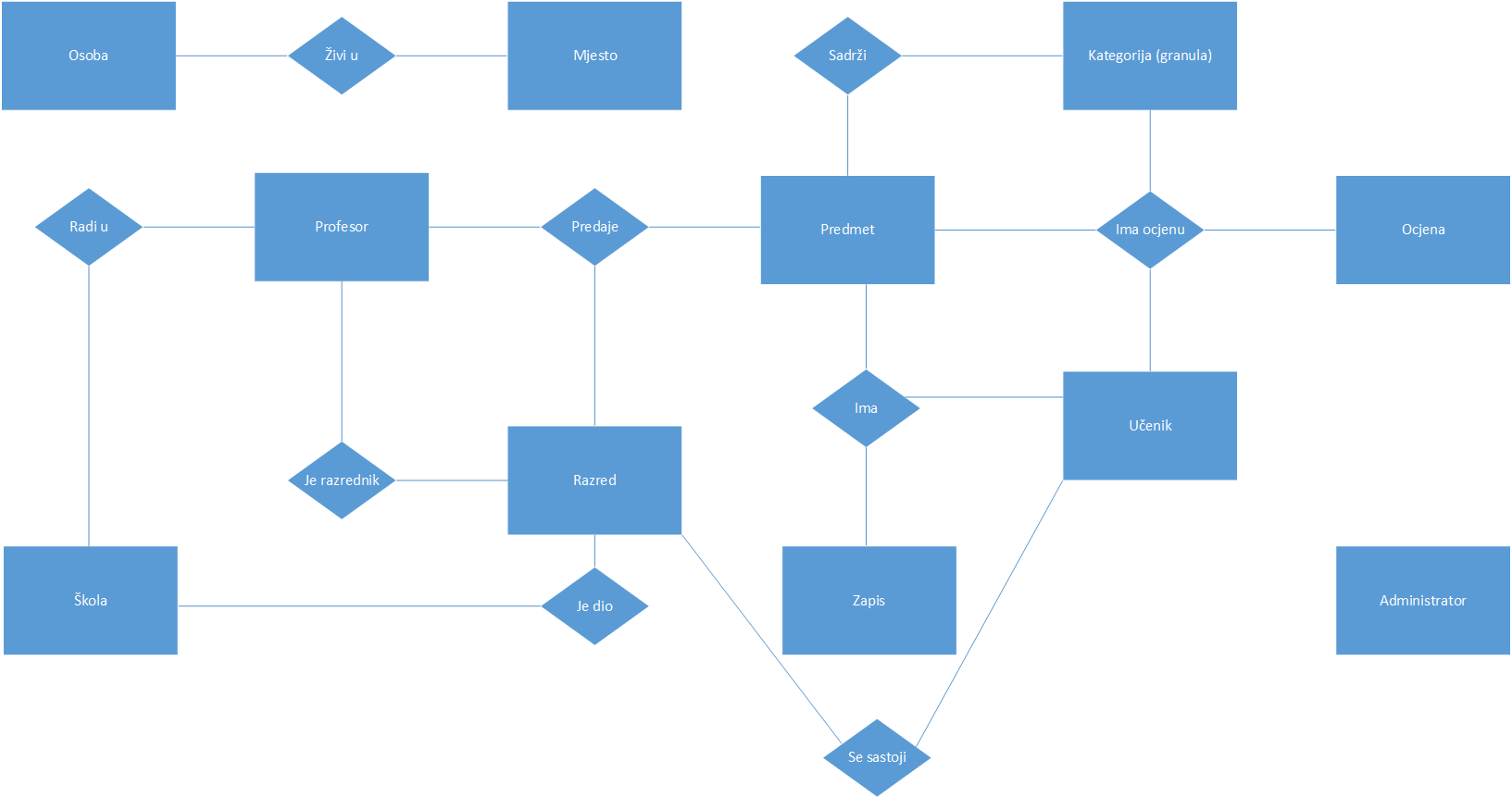
Matija Šekrst 0036477369

Zagreb, 02.02.2017.

# Uvod

U ovom seminarskog radu razradit će se projekt *eDnevnik* ostvaren sklopu kolegija diplomskog studija Fakulteta elektrotehnike i računarstva *Objektno oblikovanje*. Ovim projektom rješava se problem digitalizacije dnevnika za osnovne i srednje škole. Projekt *eDnevnik* osmišljen je i realiziran kako bi profesorima olakšao upisivanje te pregled izostanaka, ocjena i bilješki za svoj predmet ili za razred kojemu je razrednik, a učenicima omogućilo uvid u spomenute podatke kada god to požele. Naravno osim samih korisnika koji su profesori i učenici, u sustavu postoje i administratori koji mogu uređivati podatke kako ne bi došlo do nepotrebnih komplikacija prilikom osvježavanja nekih podataka ili primjerice upisa novih učenika u školu. U sklopu projekta, rješenje je izvedeno na tri različite platforme: web forma, web aplikacija te mobilna aplikacija u Androidu.

Model sustava prikazan je na slici 1.1. iz kojeg su vidljivi glavni dijelovi sustava eDnevnik.



Slika 1.1 Model sustava eDnevnik

Domena problema zahvaća kao najbitnije entitete *Profesora*, *Učenika*, *Administratora*, *Škole*, *Razreda*, *Predmeta*. Također postoji i *Zapis*, *Ocjena*, *Kategorija* za ocjenu te *Mjesto*. Budući da se *Profesor*, *Učenik* i *Administrator* imaju zajedničke karakteristike, svi oni nasljeđuju *Osobu* koja sadrži osobne podatke te je vezana na *Mjesto* kako bi se opisalo mjesto stanovanja svake osobe.

Svaki *Profesor* radi u određenoj *Školi* te predaje 1 ili više *Predmeta*. Također može biti razrednik samo jednom *Razredu* koji je dio *Škole* u kojoj radi. Svaki *Razred* sastoji se od 1 ili više *Učenika* koji slušaju *Predmete* tog *Razreda*. Također *Profesor* svakom *Učeniku* daje uz svoj *Predmet* *Ocjenu* iz određene *Kategorije* za taj *Predmet* te mu također može dodati i *Zapis*. Kroz *Zapis* profesor može ili dati neki komentar na učenikovo ponašanje ili može reći da *Učenik* nije prisutan na *Predmetu*, odnosno upisati mu izostanak.

U nastavku je dano detaljnije objašnjenje konkretnog rješenja kroz *Disign story*.

**Design story**

EDnevnik obavlja evidenciju aktivnosti učenika osnovnih i srednjih škola te omogućuje profesorima unos i pregled podataka o učenicima koji pohađaju njihovu nastavu. Korištenje eDnevnika zaštićeno je prijavom, a prilaz i aktivnosti koje su u njemu moguće definirane su ulogom korisnika. EDnevnik prati četiri različite vrste korisnika od kojih svaki ima jasno definirane mogućnosti i zadaće. Prilikom prijave upisuju se samo korisničko ime i lozinka. Četiri različite uloge u eDnevniku su: učenik, profesor, razrednik te administrator. Neovisno o ulozi svaki se korisnik  koji želi pristupiti eDnevniku prilikom početka rada mora autorizirati sa svojim korisničkim imenom i lozinkom.

Učenik nakon autorizacije na zaslonu vidi popis svih predmeta koje ima upisane u toj školskoj godini. Također osim samog imena predmeta uz svako ime predmeta prikazano je i ime profesora koji ga predaje. Učenik osim pregleda popisa predmeta detaljnije može pogledati i sve podatke o pojedinom predmetu koji uključuju popis ocjena po kategorijama za sve mjesece te bilješke iz pojedinog predmeta. Također učenik može pregledati svoje osobne podatke, ali bez mogućnosti uređivanja. Za svaku informaciju koja je netočna može se obratiti administratoru koji zapis može izmijeniti. Učenik može i pregledati popis izostanaka u kojima ima uvid u datum izostanka te upisan razlog izostanka te informaciju je li sat opravdan ili ne.

Profesor nakon autorizacije prvo odabire razred koji želi pregledati, a zatim predmet koji u tom razredu predaje. Svakom se profesor prikazuju samo razredi kojima on predaje kao i samo predmeti koje u pojedinom razredu održava. Nakon odabira razreda i predmeta koje želi prikazati dobiva uvid svih učenika tog razreda. Osim samog pregleda učenika za svakog od njih može označiti ako dotični nije na nastavi taj dan. Nadalje uz svakog učenika ima pristup svim ocjenama iz predmeta koji je odabran kao i bilješkama. Naravno osim samog pregleda ocjena i bilježaka ima mogućnost dodavanja novih zapisa te uređivanja.

Razrednik ima sve mogućnosti kao i profesor, uz dodatne mogućnosti za razred kojemu je razrednik. Profesoru ulogu razrednika dodaje administrator, a ima posebne ovlasti samo za razred kojemu je razrednik. Podrazumijeva se i jedino je moguće da je svaki razrednik razrednik samo jednom razredu. Za svoj razred može za svakog učenika vidjeti popis ocjena iz svih predmeta koje pojedini učenik pohađa. Za predmete koje on ne predaje razrednik zapise može samo pregledavati, ali ne i uređivati. Još jedna napredna mogućnost razrednika je mogućnost evidencije izostanaka svih učenika iz svog razreda. Za svaki zapis u popisu izostanaka razredniku je omogućeno brisanje zapisa te unos razloga izostanka i je li izostanak opravdan ili ne.

Administrator za razliku od ostalih korisnika eDnevnika ima mogućnost stvaranje novih korisnika te uređivanje podataka o drugim korisnicima. Za svakog učenika može mijenjati osobne podatke kao i pripadnost razredu, ali i popis predmeta koje pojedini učenik sluša. Nadalje svakom profesoru može dodati predmete koje predaje općenito kao i u pojedinom razredu. Također može dodavati zapise u grupu razreda kojima pojedini profesor predaje. Administrator također svakom profesoru može dodijeliti ulogu razrednika te mu pridružiti točno jedan razred u kojem tu ulogu obavlja.

# Zahtjevi

U ovom će se poglavlju pomoću *use caseova* opisati funkcionalni zahtjevi za ostvareni sustava, odnosno skup mogućnosti koje sustav eDnevnik pruža korisnicima.

**Use Case 1 - Prijava u sustav**

**Sažetak**: Korisnik se prijavljuje u sustav

**Obrazloženje**: Korisnik prilikom pristupanja aplikaciji ili web stranici dobiva početni ekran na kojemu je prikazana forma za prijavu u sustav. Korisnik unosi korisničko ime i lozinku u odgovarajuća polja te pritišće na gumb za prijavu.

**Korisnici**: Svi korisnici

**Preduvjeti**: Korisnik ima pristup Internetu

**Uspješan scenarij**: Korisnik se prijavljuje u sustav.

**Neuspješan scenarij**: Korisnik unosi krivo korisničko ime i/ili lozinku te prijava nije uspješna o čemu je korisnik obaviješten porukom na ekranu.

**Use Case 2 - Pregled osobnih podataka**

**Sažetak**: Korisnik pregledava podatke na svom profilu

**Obrazloženje**: Korisnik u navigacijskoj traci odabire karticu profil koja otvara prikaz njegovih osobnih podataka. Podaci se mogu samo pregledavati.  
**Korisnici**: Učenici

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 3 - Pregled predmeta i ocjena**

**Sažetak**: Korisnik pregledava podatke o svojim ocjenama

**Obrazloženje**: Korisnik u navigacijskoj traci odabire karticu predmeti koja otvara prikaz svih predmeta koje korisnik ima upisane. Predmeti su prikazani kao zatvorene kartice. Ukoliko korisnik želi vidjeti detalje predmeta odabire karticu s imenom predmeta kojeg želi pregledati te mu se otvara prikaz kategorija predmeta i unesenih ocjena po mjesecima školske godine. Iza popisa kategorija predmeta korisnik može vidjeti i bilješke koje je dobio na tom predmetu.  
**Korisnici**: Učenici

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 4 - Pregled upisanih izostanaka**

**Sažetak**: Korisnik pregledava popis izostanaka

**Obrazloženje**: Korisnik u navigacijskoj traci odabire karticu izostanci koja otvara prikaz svih izostanaka tog korisnika. Izostanci su prikazani u tablici po datumima te je pored svakog izostanka naveden razlog izostanka te njegova opravdanost ili neopravdanost.  
**Korisnici**: Učenici

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 5 - Odabir razreda**

**Sažetak**: Korisnik pregledava razrede

**Obrazloženje**: Korisniku se nakon prijave ili nakon odabira kartice razreda u navigacijskoj traci otvara prikaz svih razreda kojima korisnik ima pravo pristupiti. Pravo pristupa razredu ostvareno je samo ako korisnik razredu predaje određeni predmet.  
**Korisnici**: Profesori i razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 6 - Odabir predmeta**

**Sažetak**: Korisnik pregledava predmete koje predaje razredu

**Obrazloženje**: Korisnik se nakon odabira razreda prikazuje popis predmeta koje predaje tom razredu. Prikazan mu je minimalno jedan predmet. Odabir predmeta prosljeđuje ga se na prikaz svih učenika odabranog razreda i predmeta. Ukoliko se ovoj opciji pristupa preko navigacijske trake odabirom na karticu prikazuju se predmeti zadnje odabranog razreda.

**Korisnici**: Profesori i razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav te je odabrao razred.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 7 - Pregled i uređivanje podataka o predmetu za sve učenika**

**Sažetak**: Korisnik pregledava učenike odabranih kategorija (razred i predmet).

**Obrazloženje**: Korisniku se nakon odabira razreda i predmeta kojeg u njemu predaje prikazuju svi učenici tog razreda koji pohađaju nastavu odabranog predmeta. Učenici su prikazani kao zatvorene kartice. Ukoliko korisnik želi vidjeti ili urediti ocijene ili bilješke učenika odabire karticu s imenom učenika kojeg želi pregledati te mu se otvara prikaz kategorija predmeta i unesenih ocjena po mjesecima akademske godine. Ispod popisa kategorija predmeta korisnik može vidjeti i urediti bilješke vezane uz odabranog učenika. Ocijene i bilješke se uređuju pritiskom na polje koje se želi urediti.

**Korisnici**: Profesori i razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav. Korisnik je odabrao razred i predmet

**Uspješan scenarij**: Korisniku su prikazani podaci te je promjena uspješno spremljena.

**Neuspješan scenarij**:

**1**.Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**2**. Korisnik pokušava unijeti neispravan podatak o čemu je obaviješten ispisom poruke na ekranu

**Use Case 8 - Prijavljivanje izostanaka**

**Sažetak**: Korisnik prijavljuje izostanke učenika

**Obrazloženje**: Korisniku se nakon odabira razreda i predmeta prikazuje popis svih učenika tog razreda koji pohađaju nastavu predmeta koji predaje. Učenici su prikazani kao zatvorene kartice na kojima piše ime učenika. Svaka kartica ima na sebi polje koje je moguće označiti ukoliko učenik nije prisutan na nastavi.

**Korisnici**: Profesori i razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav te je odabrao razred i predmet

**Uspješan scenarij**: Korisniku su prikazani podaci te je izostanak uspješno pohranjen .

**Neuspješan scenarij**:

**1**. Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**2**. Došlo je do pogreške prilikom spremanje izostanka za pojedinog učenika o kojem je korisnik obaviješten ispisom poruke na ekran.

**Use Case 9 - Pregledavanje podataka o svim predmetima učenika**

**Sažetak**: Korisnik pregledava podatke o svim predmetima učenika

**Obrazloženje**: Ukoliko korisnik u navigacijskoj traci pod karticom razred odabere razred kojem je razrednik, odmah mu se prikazuje popis svih učenika toga razreda. Odabirom učenika prikazuje se popis svih predmeta za tog učenika te se odabirom na pojedini predmet mogu vidjeti sve ocjene tog predmeta zajedno sa bilješkama s tog predmeta vezanih uz tog učenika. Korisnik može samo pregledavati podatke o predmetu kojega ne predaje, a podatke o predmetima koje predaje može i uređivati.

**Korisnici**: Razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav te je odabrao razred kojem je razrednik.

**Uspješan scenarij**: Korisniku su prikazani podaci.

**Neuspješan scenarij**:  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**Use Case 10 - Pregledavanje i uređivanje svih izostanaka**

**Sažetak**: Korisnik pregledava sve izostanke učenika iz razreda kojima je razrednik

**Obrazloženje**: Korisnik u navigacijskoj traci odabire karticu izostanci koja otvara prikaz svih izostanaka učenika razreda kojemu je korisnik razrednik. Izostanci su zapisani u tablici s podacima o imenu i prezimenu učenika, datumu izostanka, razlogu izostanka te  opravdanosti  odnosno neopravdanosti izostanka. Korisnik svaki izostanak može opravdati odnosno ne opravdati ili obrisati krivi unos izostanka.

**Korisnici**: Razrednici

**Preduvjeti**: Korisnik je prijavljen u sustav te je odabran razred kojemu je razrednik.

**Uspješan scenarij**: Korisniku su prikazani podaci te su promjene pohranjene.

**Neuspješan scenarij**:

**1**.  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**2**. Korisnik pokušava unijeti neispravan podatak o čemu je obaviješten ispisom poruke na ekranu

**Use Case 11 - Registracija novih korisnika**

**Sažetak**: Korisnik registrira nove korisnike

**Obrazloženje**: Korisnik odabere opciju dodavanja novog korisnika te unosi novo korisničko ime te lozinku.

**Korisnici**: Administrator

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Novi korisnik je uspješno kreiran te se korisniku ispisuje poruka o uspješnosti operacije.

**Neuspješan scenarij**:  Korisnik unosi već postojeće korisničko ime ili je došlo do pogreške prilikom spremanja novog korisnika te se korisniku na ekran ispisuje poruka o grešci.

**Use Case 12 - Uređivanje osobnih podataka korisnika**

**Sažetak**: Korisnik uređuje osobne podatke za odabranog korisnika

**Obrazloženje**: Korisnik odabire registriranog korisnika čije osobne podatke želi promijeniti te unosi nove podatke i sprema promjene.

**Korisnici**: Administrator

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Podaci o odabranom korisniku su uspješno prikazani te su sve promjene uspješno pohranjene o čemu je korisnik obaviješten porukom na ekranu.

**Neuspješan scenarij**:

**1**.  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**2**. Korisnik pokušava unijeti neispravan podatak o čemu je obaviješten ispisom poruke na ekranu.

**Use Case 13 - Dodavanje predmeta ili razreda profesoru**

**Sažetak**: Korisnik uređuje popis predmeta koje profesor predaje određenom razredu ili općenito ili profesoru dodaje ulogu razrednika.

**Obrazloženje**: Korisnik odabire korisnika koji ima ulogu profesora te uređuje jednu od opcija. Može ili dodati općenito novi predmet koji profesor predaje ili novi razred kojemu predaje određeni predmet. Također profesoru može dodijeliti ulogu razrednika za određeni razred.

**Korisnici**: Administrator

**Preduvjeti**: Korisnik je prijavljen u sustav.

**Uspješan scenarij**: Podaci o odabranom korisniku su uspješno prikazani te su sve promjene uspješno pohranjene o čemu je korisnik obaviješten porukom na ekranu.

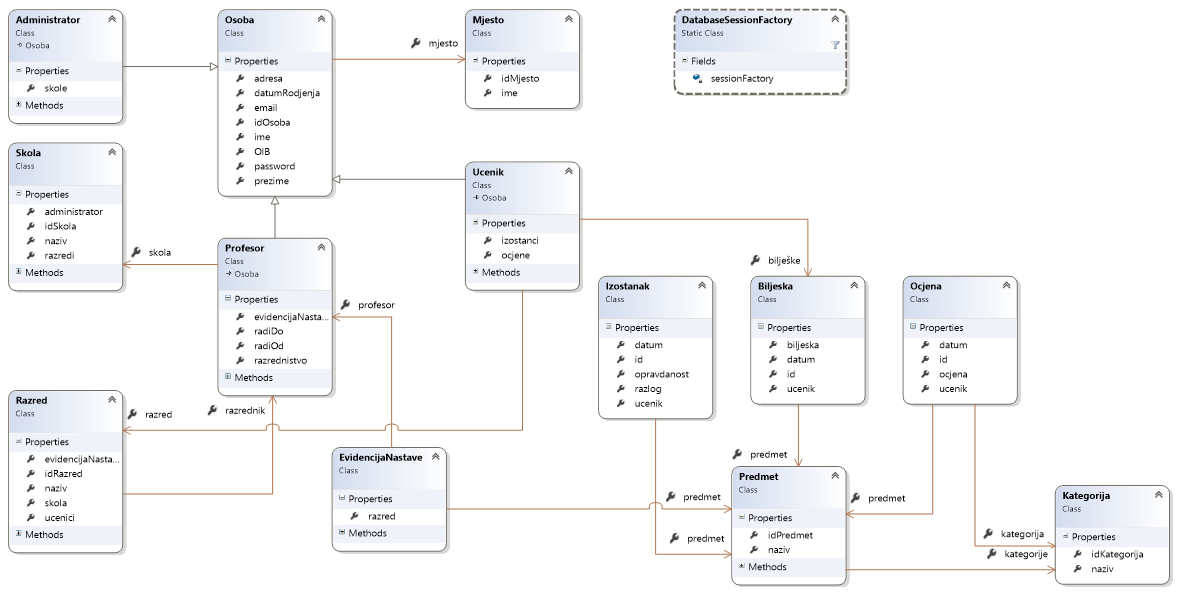
**Neuspješan scenarij**:

**1**.  Korisniku nisu prikazani podaci zbog gubitka internetske veze ili pogreške prilikom dohvata podataka. Korisniku se prikazuje stranica sa obavijesti o grešci ukoliko je scenarij neuspješan zbog greške u sustavu.

**2**. Korisnik pokušava unijeti neispravan podatak o čemu je obaviješten ispisom poruke na ekranu.

# Opis modela

U ovom će se poglavlju predstaviti dijagram klasa objektnog modela projekta eDnevnik. Sam dijagram prikazan je u nastavku na slici 3.1.



Slika 3.1. Dijagram klasa objektnog modela

Vidljivo je sa slike da postoji klasa *Osoba* sa atributima koji opisuju osnovne podatke za neku osobu(adresa, datum rođenja, e-mail itd.). Klasu *Osoba* nasljeđuju klase *Administrator, Profesor* i *Učenik* koji svaki uz osobne podatke koje nasljeđuju iz *Osobe* imaju neke specifične atribute vezane uz svoju ulogu. Također *Osoba* u sebi sadrži podataka tipa *Mjesto* koji govori o mjestu stanovanja osobe.

Kroz model idemo redom. Za početak *Učenik.* Svaki *Učenik* ima podatke o izostancima te ocjenama koje dobiva iz različitih predmeta. Također *Učenik* je vezan i uz klasu *Bilješka* koja je opisana datumom, id-em, učenikom kojem je pridijeljena te samim tekstom bilješka. *Bilješka, Izostanak* i *Ocjena* vezani su uz određeni *Predmet.* *Izostanak* je opisan podatkom o datumu, učeniku koji je izostao, razlogom izostanke te statusom opravdanosti, odnosno je li izostanak opravdan ili neopravdan. Uz samu vrijednost *Ocjene* u klasi se čuva i vrijednost datuma, id-a te učenika kojem je ona pridijeljena. Također svaka ocjena vezana je uz *Kategoriju* iz koje je dodijeljena, dok je sama *Kategorija* opisana svojim id-em i nazivom. Svaki *Predmet* ima za sebe definirane *Kategorije* za ocjenjivanje. Također *Učenik* pripada nekom *Razredu*.

Nadalje postoji klasa *Profesor* za koju su osim atributa opisani unutar klase *Osoba* bitni i atributi o početku i kraju radnog odnosa, podatku o tome je li profesor razrednik ili nije te evidencijom nastave. *Profesor* predaje u određenoj *Školi*, određenom popisu *Razreda*, a ukoliko je razrednik može mu za razredništvo biti dodijeljen samo jedan *Razred*. Također svaki *Profesor* obavlja *Evidenciju nastave.* Klasa *Evidencija nastave* povezuje profesore sa predmetima i razredima kojima predaje. Za *Predmete* kojima predaje u razredima kojima predaje *Profesor* može uređivati podatke iz tih *Predmeta* za sve *Učenike Razreda*. Može upisivati ocjene, bilješke te izostanke. Ukoliko ima status razrednika može i pregledavati podatke svih predmeta za *Učenike* svog *Razreda* te opravdavati izostanke.

Vidljiva je također i klasa *Administratora* koja također nasljeđuje klasu *Osoba*. Administrator je dodijeljen određenoj školi te mu je omogućeno uređivanje podataka o svim drugim korisnicima.

Na dijagramu je vidljiv i DatabaseSessionFactory koji je tvornica koja kreira Session prema bazi podataka.

# Opis implementacije perzistencije

Perzistencija podataka ostvarena je korištenjem relacijske baze podataka *Azure SQL Database* koja se nalazi u *Microsoftovom* oblaku. Za automatsku i transparentnu pohranu objekata u tablice relacijske baze korišten je *NHibernate Fluent.* *NHibernate Fluent* ima jednostavno sučelje za CRUD[[1]](#footnote-1) operacije nad objektima te za upite nad razredima i njihovim svojstvima. Ključna stvar pri korištenju ovog ORM-a[[2]](#footnote-2) je definiranje načina na koji će svaki objekt biti pohranjen u pripadnu tablicu baze podataka, odnosno definiranje načina mapiranja objekta. Pritom, zbog razlika u objektnoj i relacijskoj paradigmi najveći problem predstavlja mapiranje nasljeđivanja. U sustavu eDnevnik bazni razred za sve korisnike je razred *Osoba*, a izvedeni razredi su *Ucenik, Korisnik* i *Administrator*. Pošto izvedeni razredi definiraju puno dodatnih svojstva, pri implementaciji nasljeđivanja korišen je *Table per subclass* pristup kojim se nasljeđivanje u bazi podataka meodelia stranim ključevima, odnosno za svaku klasu (i apstraktnu i konkretnu) postoji tablica u bazi podataka. Način mapiranja za razrede *Osoba* i *Ucenik* prikazan je u nastavku.

|  |  |
| --- | --- |
| public class OsobaMap : ClassMap<Osoba>  {  public OsobaMap()  {  Id(x => x.idOsoba).GeneratedBy.Increment();  Map(x => x.ime).Not.Nullable();  Map(x => x.prezime).Not.Nullable();  Map(x => x.datumRodjenja).Not.Nullable();  Map(x => x.adresa).Not.Nullable();  Map(x => x.OIB).Not.Nullable();  Map(x => x.email).Not.Nullable();  Map(x => x.password).Not.Nullable();  References(x => x.mjesto);  Table("Osoba");  }  } | public class UcenikMap : SubclassMap<Ucenik>  {  public UcenikMap()  {  References(x => x.razred); HasMany(x => x.biljeske).Cascade.All;  HasMany(x => x.izostanci).Cascade.All;  HasMany(x => x.ocjene).Cascade.All;  Table("Ucenik");  }  } |

CRUD operacije nad objektima nisu ostvarene direktno u kontrolerima nego im se pristupa indirektno korištenjem *Repository* obrasca. Za svaki entitet iz baze podataka ostvarno je *repository* sučelje i razred koji ga implementira. Kontroler *repositoryju* pristupa putem sučelja, a unutar konkretne *repository* klase otvaraju se sjednica i izvršava željena transakcija. Korištenjem ovog obrasca stvoren je posrednički sloj između poslovne logike i logike pristupa podacima. Na taj način omogućuje se jednostavno testiranje, ali i korištenje jednom napisanog koda pri implementaciji sve tri aplikacije. Povezivanje MVC i *Repository* obrasca prikazano je na Slici 4.1.

**View**

**Repository**

**Nhibernate**

**Fluent**

**Database**

**Controller**

Slika 4.1. Povezivanje MVC i *Repository patterna*

# Opis izgrađene desktop aplikacije

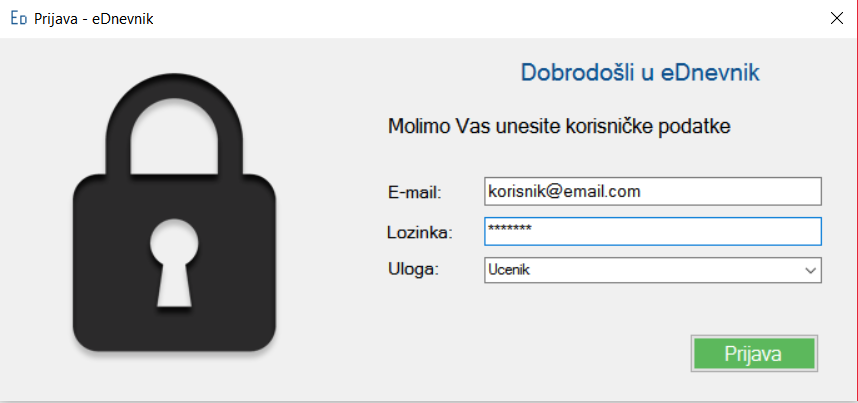
Desktop aplikacija za *eDnevnik* izgrađena je korištenjem *Windows Forms*radnog okvira. Implementacija je izvedena uporabom MVP[[3]](#footnote-3) obrasca.

*Model* dio obrasca MVP su uglavnom klase dobivene kroz *NHibernate* mapiranje uz iznimku klase *OcjenaImenik.cs*koja služi za modeliranje ocjena i bilješka koje pregledavaju učenici, a pregledavaju i uređuju profesori. Zadaća navedenog razreda je organizacija ocjena i nastavnikovih bilješki u tablice koje se zatim prikazuju korisnicima kroz *Data Grid View. Kontroleri* su centralni dio MVP obrasca. Njihova je zadaća dohvat podataka, izveden uporabom repozitorija, i obrada složenijih akcija koje generira korisnik kroz sučelje. *Kontroler* akcije dobiva od *prezentera* najčešće na način da *prezenter* detektira akciju, te tu akciju i odgovarajuće parametre prosljeđuje *kontroleru* koji zna akciju obraditi i odrediti što će se iduće prikazati korisniku. *Prezenter* dio se sastoji od formi koje se prikazuju korisnicima.

Važno je napomenuti da desktop dio pokriva samo učenike i profesore, dok je za administratora namijenjena web aplikacija opisana u poglavlju 6. U nastavku slijedi detaljniji opis implementacije.

# Prijava u sustav

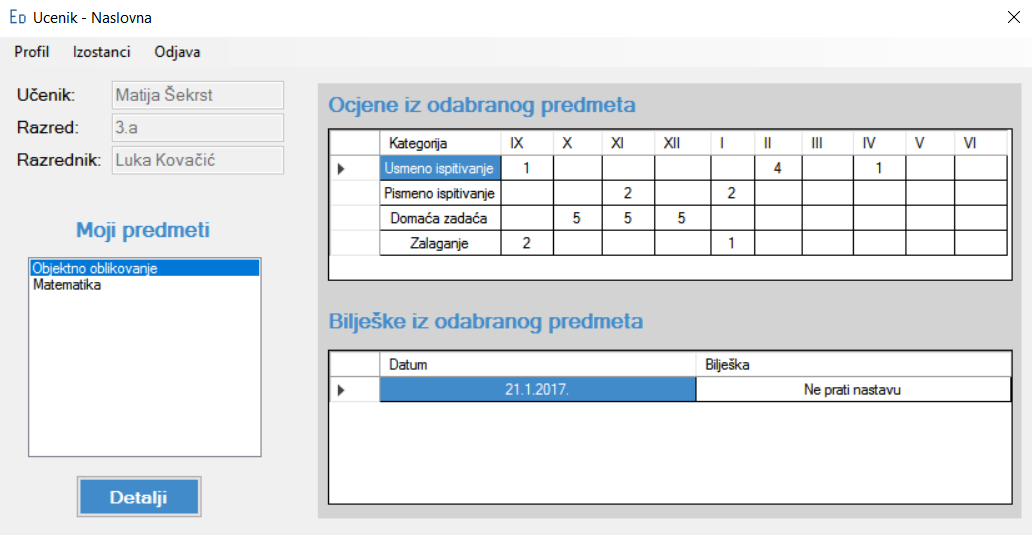
Na slici 5.1 prikazana je prijava u sustav. Korisnik upisuje E-mail i lozinku, te odabire ulogu u sustavu. Ako se upisani E-mail, lozinka i uloga poklapaju s zapisom u bazi podataka, korisniku se dozvoljava ulaz u sustav. Ovisno o ulozi korisniku se prikazuje naslovna forma. Za prikaz odgovarajuće forme kao i prijavu u sustav brine se *HomeController.cs* koji sadrži referencu na repozitorij zadužen za autorizaciju, *AuthorizationRepository.cs.* Ukoliko se u sustav želi prijaviti administrator prikazuje mu se poruka da je njemu namijenjena web aplikacija. U slučaju unosa netočnih podataka korisniku se prikazuje odgovarajuća poruka.



Slika 5.1 Prijava u sustav

# Učenik – Naslovna

Ukoliko je prijavljeni korisnik učenik onda mu se prikazuje naslovna stranica za učenike prikazana na slici 5.2.



Slika 5.2 Naslovna forma za učenike

Naslovna forma za učenike sastoji se od navigacijske trake na vrhu, osnovnih podataka o učeniku, popisa predmeta koje učenik pohađa, te konačno od ocjena i bilješki za željeni predmet.

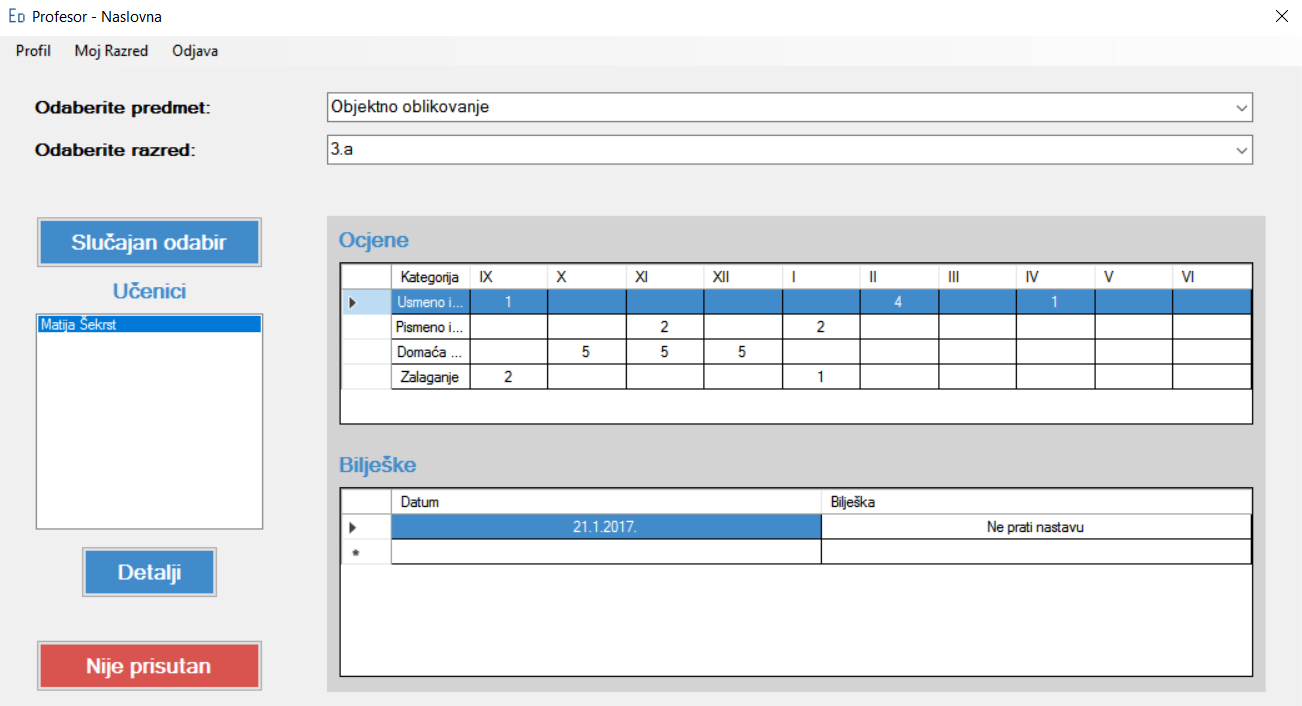
Navigacijska traka omogućuje učenicima pregled profila, popis učenikovih izostanaka i odjavu iz sustava.

Popis predmeta služi za odabir predmeta na način da se u listi odabire željeni predmet, te se pritiskom na gumb «Detalji» prikazuju ocjene i bilješke za odabrani predmet. Učenik je u mogućnosti samo pregledavati sve stavke i ne može mijenjati, dodavati i brisati podatke.

Za rad forme brine se *UcenikNaslovnaController.cs* koji dohvaća podatke iz baze podataka i obrađuje akcije za navigaciju, dok se navigacija kroz podatke odvija u samom *prezenteru*.

# Profesor – Naslovna

Ukoliko je prijavljeni korisnik profesor onda mu se prikazuje naslovna stranica za profesore prikazana na slici 5.3.



Slika 5.3 Naslovna stranica za profesora

Naslovna stranica za profesora sastoji se od navigacijske trake, panela za odabir predmeta i razreda, gumba za slučajni odabir, popisa učenika, te ocjena i zapisa za učenika.

Navigacijska traka omogućava profesorima pregled svoga profila s osobnim informacijama. Pregled izostanaka za svoj razred i opravdavanje istih, te odjavu iz sustava.

Odabir predmeta i razreda je polazište svakog profesora, jer profesor prvo mora odabrati koji predmet želi pregledati (u slučaju da predaje više predmeta) i u kojem razredu (ako predaje isti predmet u više razreda). Nakon odabira predmeta i razreda prikazani su učenici toga razreda. Profesor može odabrati učenika i pritiskom na «Detalji» pogledati ocjene i bilješke za učenika, ili odabrati nasumičnog učenika pritiskom na «Slučajan odabir». Važna je i evidencija odsustva koja se provodi pritiskom na gumb «Nije prisutan» na način da se odabere učenik u listi i pritisne se gumb. Pritisak gumba prikazuje dijalog za potvrdu odabira, te ovisno o ishodu sprema ili odbacuje izostanak.

Glavna zadaća profesora je upisivanje ocjena i bilješka o učenicima. To se provodi kroz *Data Grid View* komponente. Važno je napomenuti da u svaku čeliju tablice smije ići po jedna ocjena.

Formu kontrolira *ProfesorNaslovnaController.cs* koji služi za dohvat podataka, obradu akcija navigacijske trake i spremanje bilješki i ocjena. *Kontroler* je također zadužen i za osvježavanje podataka pri svakoj promjeni.

# Profil

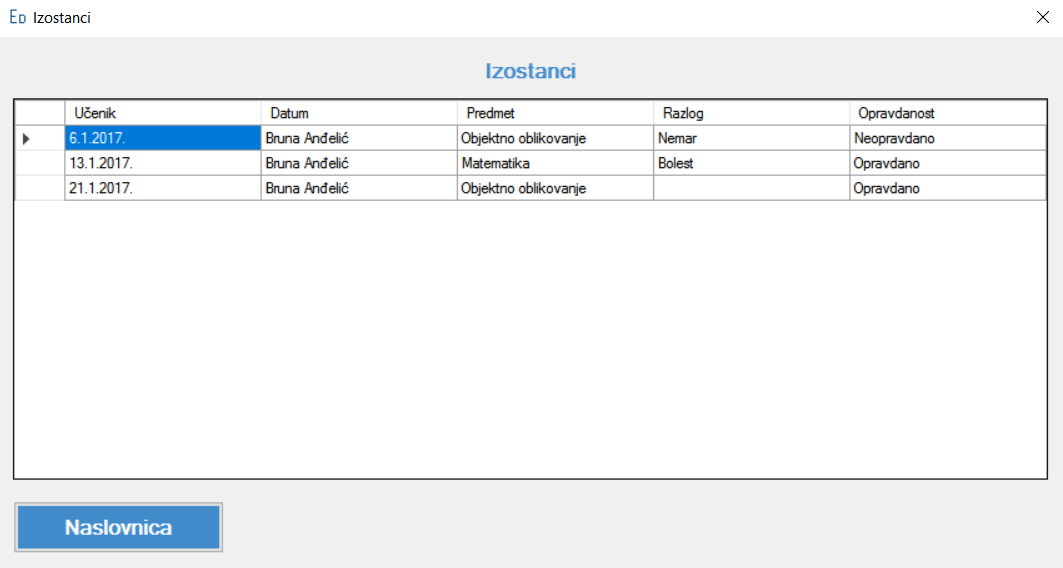
Profil prikazuje osobne podatke korisnika i identičan je za sve korisnike. Izgled profila prikazan je na slici 5.4. Profil služi za prikaz osobnih podataka o korisniku. Korisnici ne smiju mijenjati podatke pa im ta funkcionalnost nije omogućena. Pritiskom na gumb «Naslovna» korisnici se vraćaju na njihovu naslovnu stranicu. Za kreiranje profila koristi se objekt tipa *Osoba.*



Slika 5.4 Profil korisnika

# Moj Razred

Moj razred se prikazuje samo profesorima koji su razrednici. Forma se sastoji od popisa svih izostanak učenika u razredu i služi za pregled i opravdavanje izostanaka. Izgled forme prikazan je na slici 5.5. Svaki izostanak se prilikom upisa smatra neopravdanim. Razrednik tako ima zadaću opravdavati opravdane izostanke, upisivanjem razloga i opravdanosti. Da bi se opravdao izostanak dostatno je pod opravdanost upisati «opravdano», a razlog se može ali i ne mora navesti. Za formu je zadužen *IzostanciController.cs* koji služi za spremanje promjena podataka, dok se forma kreira sa popisom izostanaka.



Slika 5.5 Popis izostanaka

# Opis izgrađene web aplikacije

Web aplikacija za *eDnevnik* je izvedena uporabom MVC obrasca. Model dio obrasca MVC su razredi kojima je omogućeno spajanje podataka dobivenih iz baze podataka s vrijednostima koje će biti potrebno prikazati. Razredi koji se predaju viewu osim za prikaz i za uređivanje podataka sadrže u sebi potrebne validacije za polja. Također određeni razredi sadrže metode toList() koje nam služe za dobivanje liste objekata tog modela.

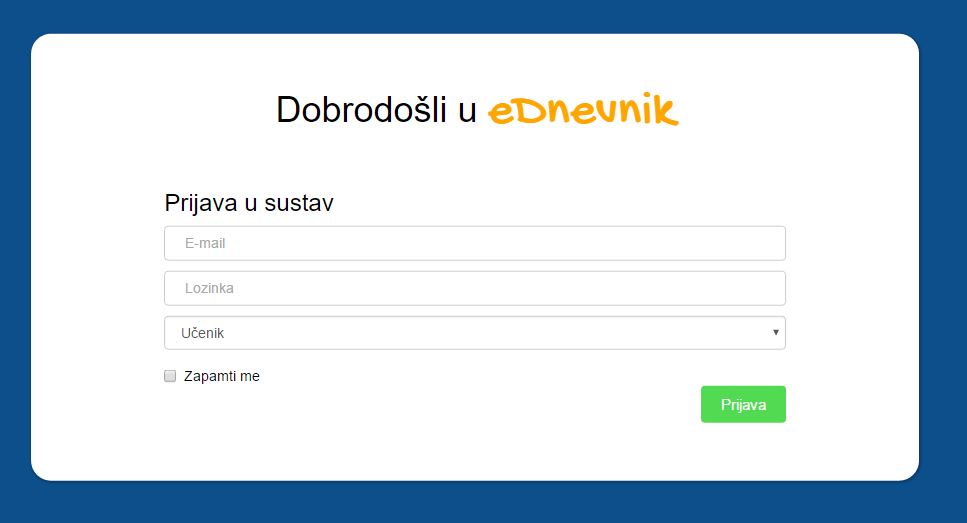
Kontroler kao središnji dio MVC obrazaca sluša korisničke zahtjeve, dohvaća iz baze podatke pomoću repozitorija i spaja ih s modelima. Izgrađene modele predaje *Viewu.*

*View* dio modele dobivene od kontrolera prikazuje korisniku. On isto tako poziva određene metode kontrolera ovisno o korisničkom odabiru.

Web aplikacija izrađena je za tri uloge: učenik, profesor i administrator, a u nastavku su opisani dijelovi implementcije.

# Prijava u sustav

Korisnik za prijavu u sustav upisuje e-mail i lozinku te odabire ulogu u sustavu. Ako se upisani e-mail, lozinka i uloga poklapaju sa zapisom u bazi podataka, korisniku se dozvoljava ulaz u sustav. Ovisno o ulozi, korisniku se prikazuje naslovna forma. Za prikaz odgovarajuće stranice kao i prijavu u sustav brine se *AutorizacijaController.cs.* Prikaz prijave u sustav vidljiv je na slici 6.1.

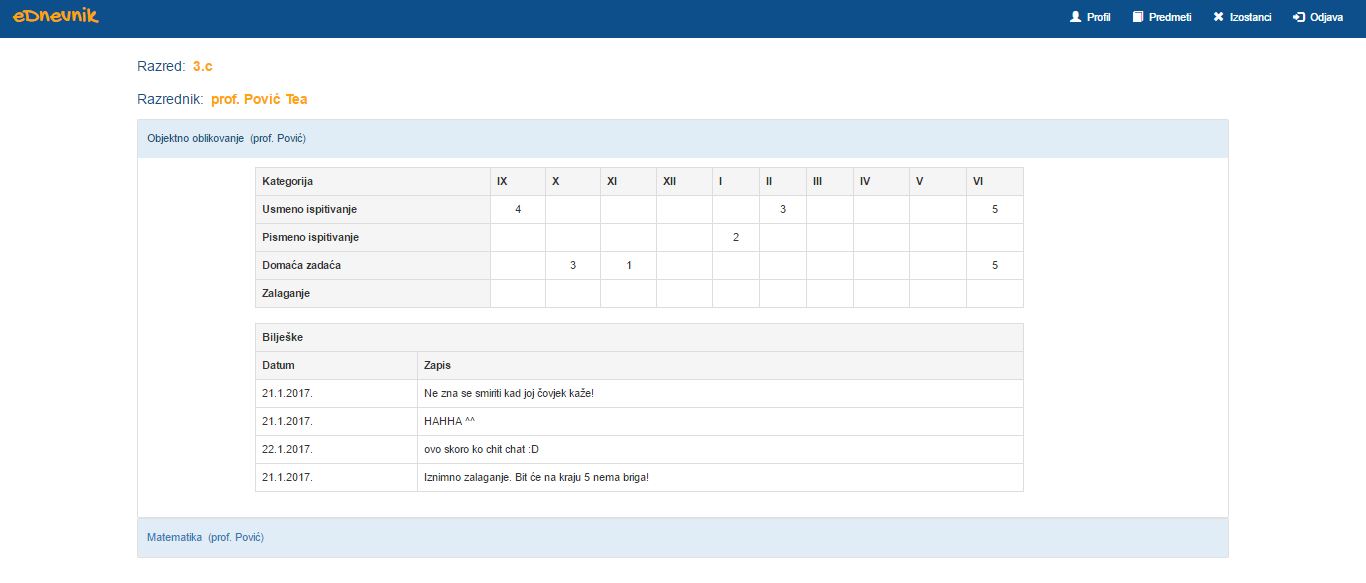


Slika 6.1 Prijava u sustav

# Učenik

Ukoliko je prijavljeni korisnik učenik onda mu se kao početna stranica prikazuje njegov profil. Profil je izgledom identičan i u slučaju prijave profesora.   
Prikaz, osim profila, sadrži i navigacijsku traku na kojoj se nalaze kartice za Profil, Predmeti, Izostanci, Odjava.

Odabirom Predmeti učeniku se prikazuje popis predmeta koje učenik sluša. Ima pregled svih upisanih ocjena u akademskoj godini te svih upisanih bilješki za taj predmet. Na slici 6.2 prikazan je izgled kartice Predmeti.



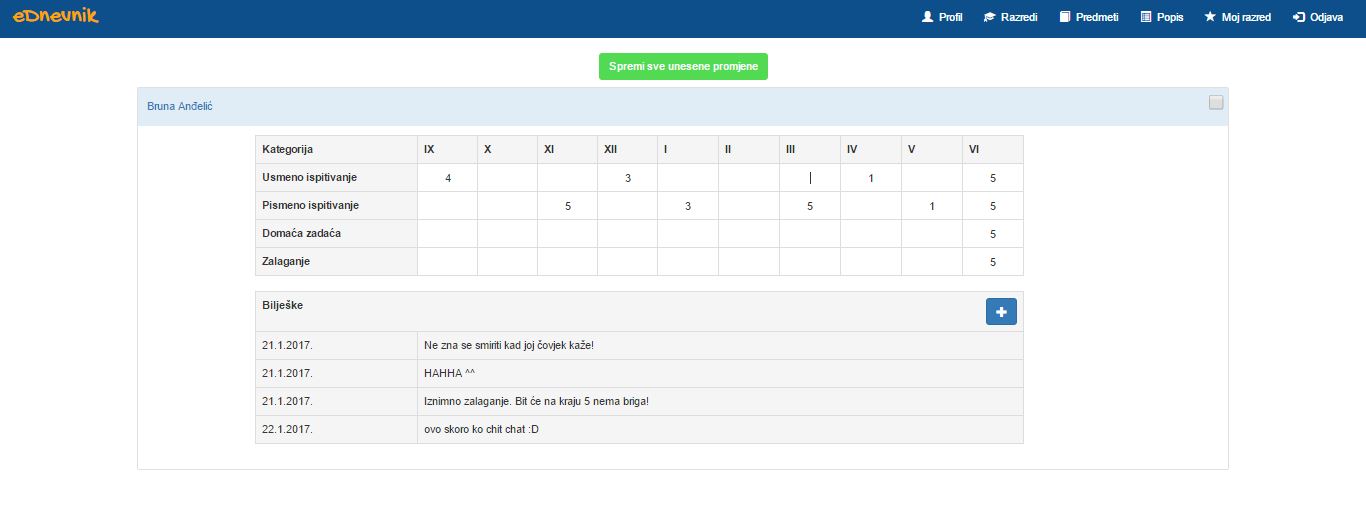
Slika 6.2 Prikaz predmeta učenika

Odabirom kartice Izostanci učeniku se prikazuje svi satovi na kojima je izostao. Može vidjeti da li je izostanak opravdan ili neopravdan. Odabirom Odjave korisnik se vraća na početni prikaz, odnosno prijavu u sustav.

Za prikaz podataka učeniku zadužen je *UcenikController.cs*.

# c.Profesor

Ukoliko je prijavljeni korisnik profesor onda mu se prikazuje popis svih razreda kojima predaje. Odabirom razreda pikazuje se popis predmeta koje profesor predaje tom razredu. Profesor mora odabrati jedan predmet da bi nastavio. Odabirom predmeta prikazuje se popis učenika i njihovih ocjena. Prikaz razreda prikazan je na slici 6.3.

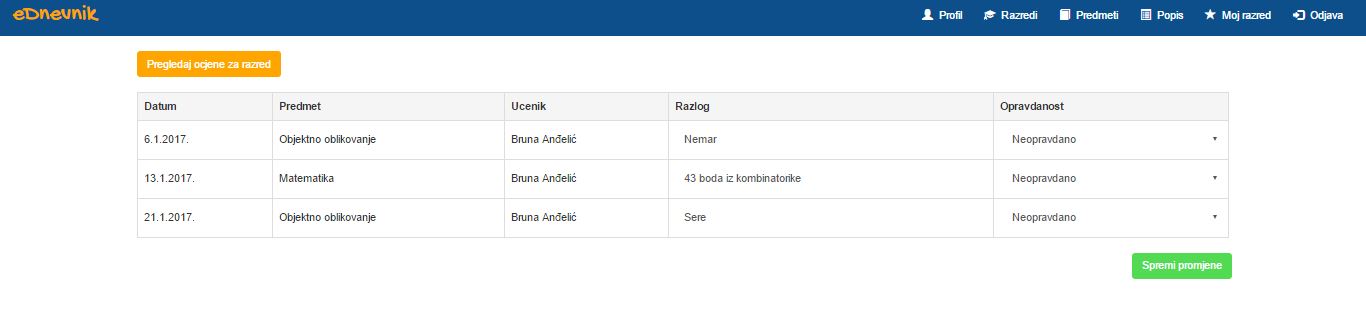


Slika 6.3 Prikaz odabranog razreda

Profesor ima mogućnost uređivanja unesenih ocijena i unosa novi. Također odabirom oznake sa znakom + nudi mu se mogućnost unosa nove bilješke za učenika. Ukoliko učenik trenutačno nije na predavanju profesor može označiti polje pored njegovog imena te će se time učeniku upisati izostanak. Sve promjene koje je profesor unio potrebno je spremiti pritiskom gumba „Spremi sve unesene promjene“.

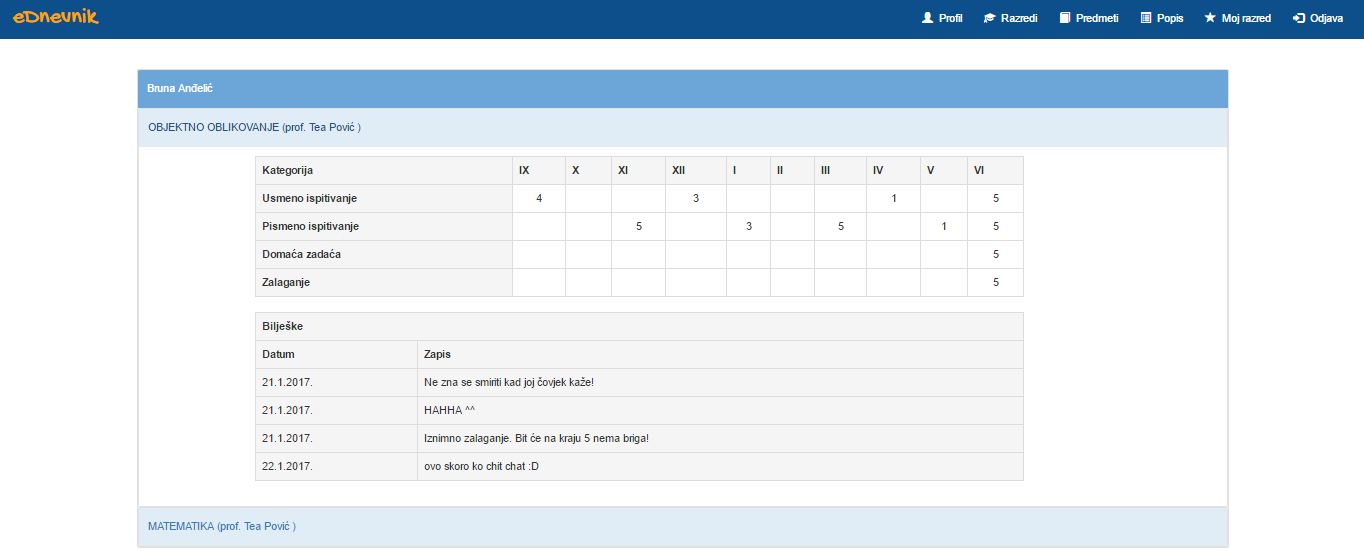
Navigacijska traka omogućava profesorima pregled svoga profila s osobnim informacijama, pregled podataka svog razreda, pregled odabranog razreda i ponovni odabir razreda te odjavu iz sustava.

Odabirom kartice Moj razred otvara mu se prikaz prikazan na slici 6.4.



Slika 6.4 Prikaz Moj razred

Profesor može svom razredu opravdati izostanke te dodati obrazloženje. Unesene promjene sprema pritiskom gumba „Spremi promjene“.Odabirom „Pogledaj ocjene za razred“ prikazuje mu se popis svih učenika i njihovih predmeta te ocjena iz predmeta. Te ocjene razrednik može samo pregledavati. Na slici 6.5. prikazan je popis učenika sa razredom.



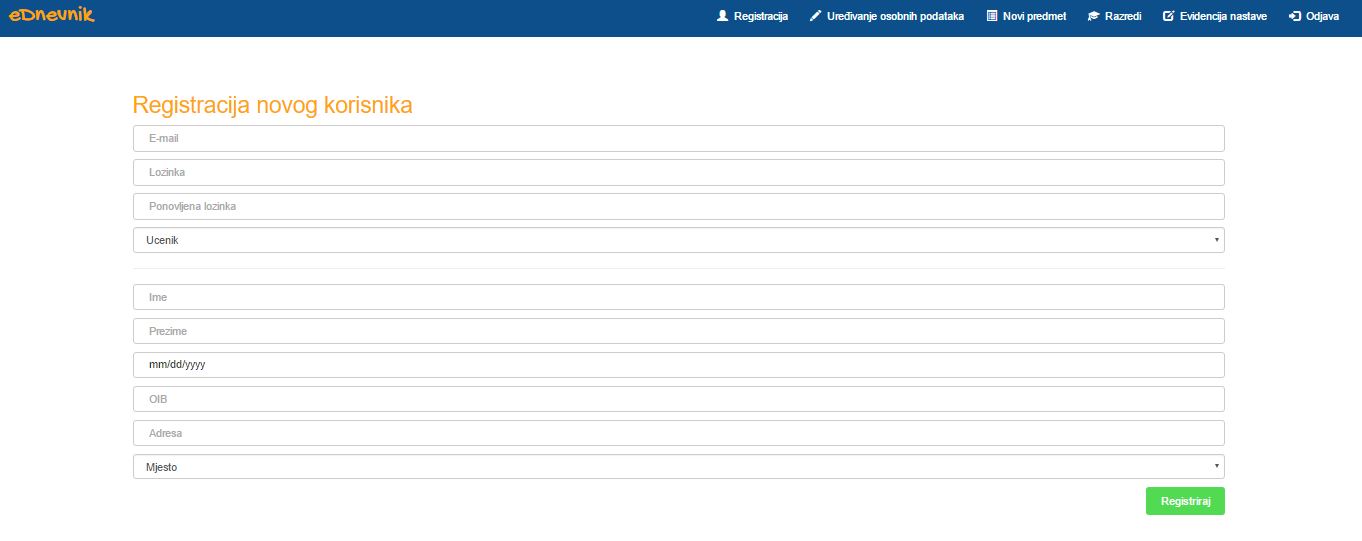
Slika 6.5 Prikaz ocjena za razrednika

Odabirom Odjave korisnik se vraća na početni prikaz, odnosno prijavu u sustav.

Za prikaz podataka profesoru zadužen je *ProfesorController.cs*.

# d.Administrator

Ukoliko je prijavljeni korisnik administrator početni prikaz mu je Registracija novog korisnika. Prikaz registracije prikazan je na slici 6.6.

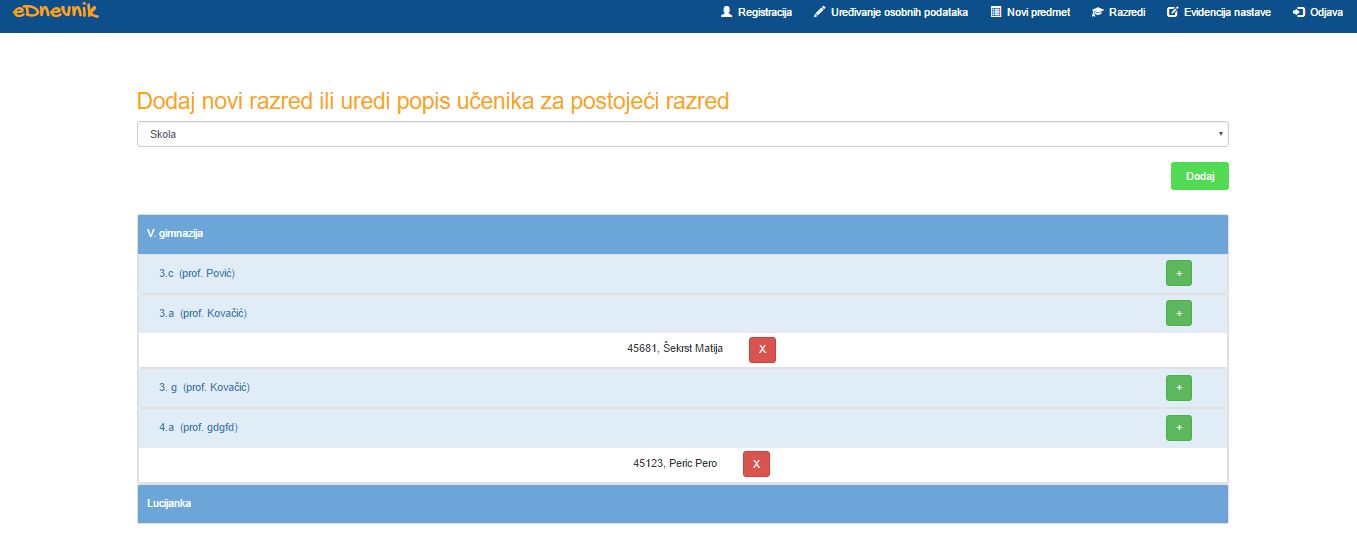


Slika 6.6 Registracija novog korisnika

Potrebno je popuniti sve podatke te se tek tada novi korisnik može registrirati.

Administrator u navigacijskoj traci ima kartice: Registracija, Uređivanje osobnih podataka, Novi predmet, Razredi, Evidencija Nastave. Odabirom uređivanja osobnih podataka potrebno je prvo odabrati korisnika koji se želi urediti te tada unijeti željene promjene. Prikaz uređivanja je isti kao u Registracija uz dodatak odabira osobe.

Ukoliko administrator odabere karticu Novi predmet ima mogućnost dodavanja novog predmeta. Prilikom dodavanja novog predmeta potrebno je paziti da predmet s tim nazivom već ne postoji u bazi. Ukoliko postoji sustav neće dopustiti dodavanje istog.  
Odabirom kartice Razredi prikazuje se prikaz na slici 6.7.

 Slika 6.7 Uređivanje razreda

Administratoru se nude dvije mogućnosti, dodavanje novog razreda ili dodavanje učenika postojećim razredima. Za dodavanje novog razreda potrebno je prvo odabrati školu. Odabirom škole administratoru se prikazuje polje za naziv razreda i odabir razrednika. Dodavanje se završava pritiskom tipke Dodaj. Za dodavanje učenika razredu potrebno je pritisnuti oznaku + pored naziva razreda. Pritiskom se otvara modal s popisom učenika koji nisu dodani ni jednom razredu.

Zadnja opcija koja se nudi administratoru je dodavanje evidencije nastave, odnosno dodavanje predmeta razredu. Za dodavanje predmeta razredu potrebno je odabrati i profesora koji će predavati taj predmet. Za sve prikaze podataka administratoru zadužen je *AdministratorController.cs*.

# Opis izgrađene mobilne aplikacije

Aplikacija je izgrađena za mobilnu platformu za operacijski sustav Android. Korišteni IDE za razvoj je Android Studio, a jezik Java. Aplikacija nudi sve funkcionalnosti kao desktop i web aplikacija uz ograničenje samo za Učenik i Profesor uloge.

Komunikaciju s klijentske strane za nas obavlja framework Retrofit (<https://square.github.io/retrofit/>) koji se temelji na protokolu HTTP.

Inicijalizacija Retrofit frameworka se izvodi kao u sljedećem primjeru:

Retrofit retrofit = **new** Retrofit.Builder()  
 .baseUrl(Utils.*getADDRES*())  
 .build();

Nad klasom Retrofit izvodi se statička metoda *baseUrl* kojoj je kao argument predana IP adresa računala na kojem je naš server pokrenut. IP adresa se dohvaća preko pomoćne *Utils* klase.

U sučelju RestApi su popisane rute na koje se možemo povezivati npr.

@GET(**"api/Ucenik/Profil"**)  
Call<User> getProfile(@Header(**"Authorization"**) String token);

Ovdje vidimo metodu *getProfile* koju možemo pozivati unutar koda i koja će vraćati objekt tipa *User* a kao argument prima token sjednice u tipu String. Kada se *getProfile* moteoda pozove na rutu *api/Ucenik/Profil* se pošalje *GET* zahtjev sa svim potrebnim podacima.

Nakon poziva neke od metoda koja je usmjerena na unaprijed predodređenu rutu izvodi se asinkroni poziv te se odgovor mora očekivati u *Callbacku* u metodama *onResponse* i *onFailure*.

Poslužitelju se pristupa preko vrata 3000. U nastavku je opisan api poslužitelja:

|  |  |
| --- | --- |
| URL | /api/Login |
| OPIS | Prijava u sustav. |
| METODA | POST |
| HEADER PARAMETRI | / |
| URL PARAMETRI | / |
| DATA PARAMETRI | Email – email adresa korisnika  Password – password korisnika  Role – uloga korisnika (Ucenik, Profesor, Administrator) |
| SUCCESS RESPONSE | 200 OK   * Token korisnika json obliku |
| ERROR RESPONSE | 400 BAD REQUEST  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Ucenik/Profil |
| OPIS | Dohvat osobnih informacija o učeniku. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token učenika |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Osobni podaci o učeniku u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Ucenik/Subjects |
| OPIS | Dohvat svih predmeta koje učenik ima upisane u tekućoj školskoj godini. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token učenika |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis predmeta koje učenik sluša u tekućoj školskoj godini u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Ucenik/Subject/{subjectId}/Cmn |
| OPIS | Dohvat svih informacija o pojedinom učeniku za predmet sa zadanim subjectId. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token učenika |
| URL PARAMETRI | subjectId – integer – id predmeta |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * kategorije predmeta sa zadanim subjectId * ocjene i bilješke učenika na predmetu sa zadanim subjectId |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Ucenik/Absences |
| OPIS | Dohvat popisa izostanaka učenika. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token učenika |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis svih izostanaka učenika json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Profil |
| OPIS | Dohvat osobnih informacija o profesoru. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Osobni podaci o profesoru u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Grades |
| OPIS | Dohvat svih razreda kojima profesor predaje. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis svih razreda kojima profesor predaje u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/MyGrade |
| OPIS | Dohvat popisa učenika kojima je profesor razrednik. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis učenika kojima je profesor razrednik u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Profil |
| OPIS | Dohvat osobnih informacija o profesoru. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Osobni podaci o profesoru u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Subjects/{gradeId} |
| OPIS | Dohvat popisa predmeta koje profesor predaje razredu sa zadanim gradeId. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | gradeId – integer – id razreda |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis predmeta koje profesor predaje željenom razredu u json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Grades/{gradeId}/Subjects/{subjectId}/Students |
| OPIS | Dohvat popisa svih učenika razreda sa zadanim gradeId koji slušaju predmet sa zadanim subjectId. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | gradeId – integer – id razreda  subjectId – integer – id predmeta |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * Popis svih učenika razreda sa zadanim gradeId koji slušaju predmet sa zadanim subjectId json obliku |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/Grades/{gradeId}/Subjects/{subjectId}/Students/{studentId} |
| OPIS | Dohvat svih informacija o učeniku sa zadanim subjectId za predmet sa zadanim subjectId. |
| METODA | GET |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | gradeId – integer – id razreda  subjectId – integer – id predmeta  studentId – integer – id učenika |
| DATA PARAMETRI | / |
| SUCCESS RESPONSE | 200 OK   * kategorije predmeta sa zadanim subjectId * ocjene i bilješke učenika da zadanim studentId na predmetu sa zadanim subjectId |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/AddNote |
| OPIS | Dodavanje bilješke za učenika. |
| METODA | POST |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | studentId – intger - id učenika  subjectId – integer - id predmeta  date – date - datum  note – string – tekst bilješke |
| SUCCESS RESPONSE | 200 OK |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/AddAbsence |
| OPIS | Prijavljivanje izostanka učenika. |
| METODA | POST |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | studentId – intger - id učenika  subjectId – integer - id predmeta  date – date - datum  reason – string – razlog izostanka |
| SUCCESS RESPONSE | 200 OK |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

|  |  |
| --- | --- |
| URL | /api/Profesor/AddMark |
| OPIS | Dodavanje ocjene učeniku. |
| METODA | POST |
| HEADER PARAMETRI | Authorization - token profesora |
| URL PARAMETRI | / |
| DATA PARAMETRI | studentId – intger - id učenika  subjectId – integer - id predmeta  categoryId – integer – id kategorije  date – date - datum  mark – integer – ocjena |
| SUCCESS RESPONSE | 200 OK |
| ERROR RESPONSE | 401 UNAUTHORIZED  500 SERVER ERROR |

1. **C**reate, **R**ead, **U**pdate, **D**elete [↑](#footnote-ref-1)
2. *Object/Relational Mapping* [↑](#footnote-ref-2)
3. ***M****odel****V****iew****P****resenter –*arhitekturni obrazac namijenjen za izradu korisničkog sučelja [↑](#footnote-ref-3)