

E-DNEVNIK

Članovi tima:

1. *Beglerović Vildana*
2. *Buturović Lejla*
3. *Beglerović Vedad*

Opis:

Naša aplikacija je namjena za evidenciju nastavnih aktivnosti elektronskim putem. Jako je bitno da roditelji stalno prate rad i uspjeh svog djeteta u školskim aktivnostima kako bi dijete bilo što uspješnije. E-dnevnik roditeljima pruža mogućnost da blagovremeno budu obavješteni o ocjenama i izostancima svog djeteta, nastavnicima olakšava evidenciju rada, a učenici će imati bolji uvid u svoje ocjene. Korisnik će se moći prijaviti na odgovarajući profil: roditelj, učenik ili nastavnik koji kreira administrator. Nastavnik može biti ujedno i razrednik jednom odjeljenju pa će shodno tome imati dodatne mogućnosti.

Akteri:

- Nastavnik
- Razrednik
- Roditelj
- Učenik
- Administrator

Funkcionalnosti:

Administrator:

- mogućnost kreiranja naloga za nastavnike, učenike i roditelje
- mogućnost kreiranja i ažuriranja profila učenika
- mogućnost kreiranja i ažuriranja profila nastavnika
- mogućnost formiranja odjeljenja

Nastavnik:

- mogućnost uvida u ocjene učenika
- mogućnost evidentiranja časa
- mogućnost upisivanja odsutnih učenika
- mogućnost upisivanja ocjena iz predmeta koji predaje
- mogućnost zakazivanja testova i pismenih zadaća
- mogućnost zaključivanja ocjene
- mogućnost pisanja napomena za razrednike

Razrednik:

- ima sve ovlasti kao i nastavnik za predmete koje predaje
- mogućnost uvida u ocjene i izostanke svih učenika u odjeljenju
- mogućnost uvida u podatke svakog učenika iz odjeljenja

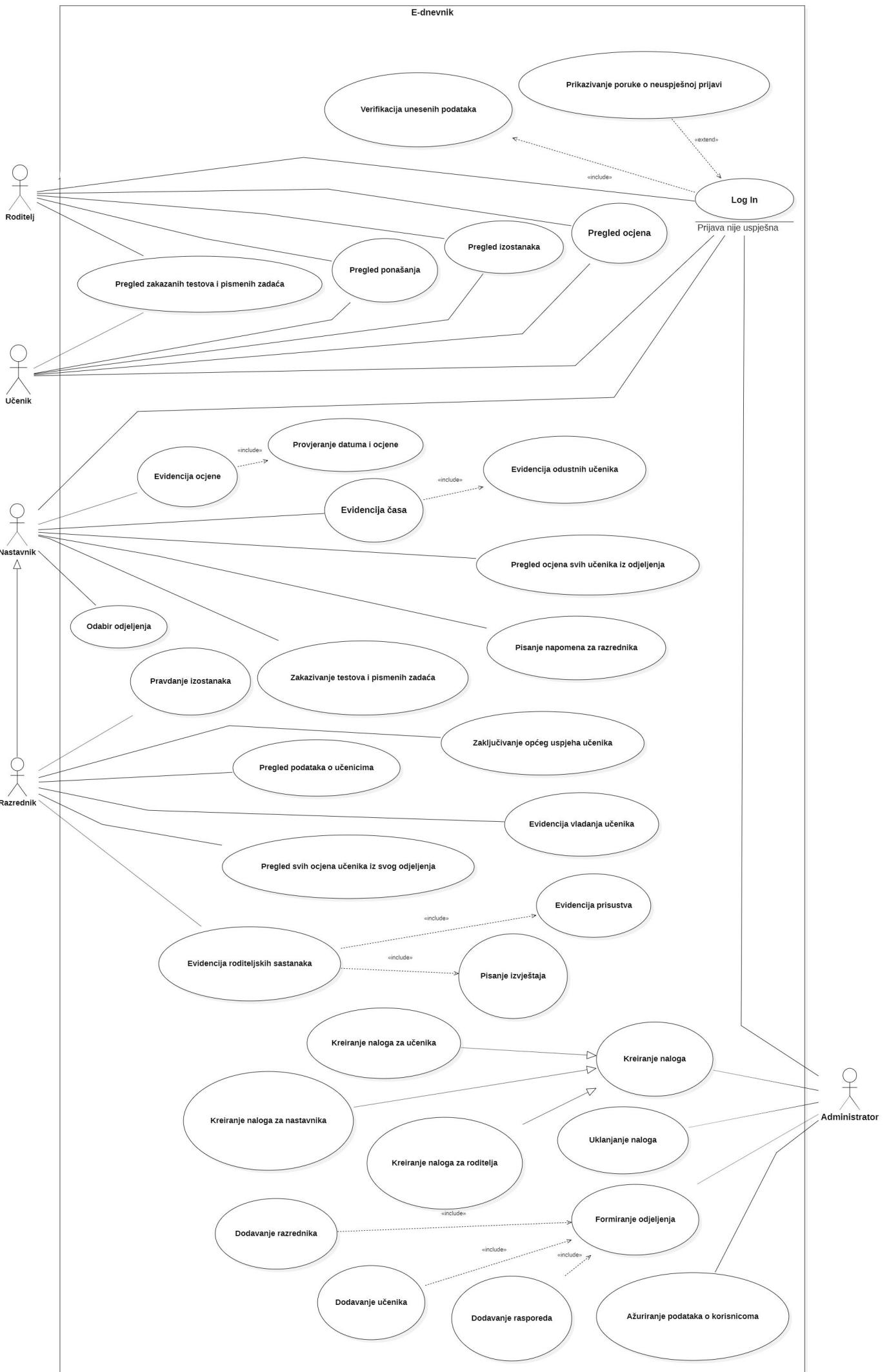
- mogućnost pravdanja i specificiranja vrste izostanaka
- mogućnost zaključivanja općeg uspjeha učenika
- mogućnost pisanja napomena za roditelje
- mogućnost pisanja izvještaja i evidentiranja prisustva roditeljskim sastancima
- mogućnost evidencije o vladanju učenika

Roditelj:

- mogućnost uvida u ocjene i izostanke svog djeteta
- mogućnost uvida u ponašanje svog djeteta
- mogućnost uvida u termine testova i pismenih zadaća

Učenik:

- iste mogućnosti kao i roditelj



SCENARIJ 1- Evidencija roditeljskih sastanaka

Naziv slučaja upotrebe	Evidencija roditeljskih sastanaka
Opis slučaja upotrebe	Razrednik popunjava izvještaj sa roditeljskog sastanka i evidentira prisustvo roditelja
Vezani zahtjevi	/
Preduslovi	1. Uspješna prijava na sistem
Posljedice-uspješan završetak	Trajno evidentiran izvještaj sa roditeljskog sastanka
Posljedice-neuspješan završetak	Izvještaj sa roditeljskog sastanka nije trajno evidentiran
Primarni akteri	Razrednik
Ostali akteri	
Glavni tok	Razrednik u rubriku za roditeljske sastanke upisuje izvještaj sa roditeljskog sastanka (napomene, teme za diskusiju i zaključke). Za svakog roditelja posebno evidentira njegovo prisustvo.
Proširenja/Alternative	/

Tok događaja 1 (uspješan zadatak)

Razrednik	Sistem e-dnevnik
1. Upis datuma	
2. Upis teme roditeljskog sastanka	
3. Upis zaključka sa sastanka	
4. Popunjavanje prisustva	
5. Potvrda unosa	
	6. Trajni zapis izvještaja u dnevnik

Tok događaja 2 (neuspješan završetak)

Razrednik	Sistem e-dnevnik
1. Upis datuma	
2. Upis teme roditeljskog sastanka	
3. Upis zaključka sa sastanka	
4. Izbor liste roditelja	
5. Popunjavanje prisustva	
6. Potvrda unosa	
	7. Ispis poruke o neuspješnom evidentiranju roditeljskog sastanka

SCENARIJ 2 - Upisivanje ocjene

Naziv slučaja upotrebe	Upisivanje ocjene
Opis slučaja upotrebe	Nastavnik upisuje učeniku ocjenu iz predmeta kojeg predaje
Vezani zahtjevi	/
Preduslovi	Uspješna prijava na sistem
Posljedice-uspješan završetak	Ocjena evidentirana u dnevnik
Posljedice-neuspješan završetak	Ocjena nije evidentirana u dnevnik
Primarni akteri	Nastavnik
Ostali akteri	/
Glavni tok	Nakon uspješne prijave na sistem i odabira odjeljenja i predmeta, nastavnik bira učenika kojem treba upisati ocjenu. Zatim unosi datum, pri čemu sistem provjerava validnost datuma, kao i da li je učenik prisutan, potom bira ocjenu i način provjere znanja. Ukoliko nastavnik potvrdi unos, ocjena se evidentira u dnevnik, u suprotnom ocjena neće biti unesena.
Proširenja/Alternative	/

Tok događaja 1 (uspješan zadatak)

Nastavnik	Sistem
1. Odabir odjeljenja i predmeta	
2. Odabir učenika	
3. Unos datuma	Provjera ispravnosti unesenog datuma i da li je učenik prisutan na taj datum
4. Unos ocjene	Provjera ocjene (da li je od 1 do 5)
5. Odabir načina provjere znanja	
6. Potvrda unosa	Evidentiranje ocjene u dnevnik

Tok događaja 2 (neuspješan završetak)

Nastavnik	Sistem
1. Odabir odjeljenja i predmeta	
2. Odabir učenika	
3. Unos datuma	Provjera ispravnosti unesenog datuma i da li je učenik prisutan na taj datum
4. Unos ocjene	Provjera ocjene (da li je od 1 do 5)
5. Odabir načina provjere znanja	
6. Nastavnik ne odobrava unos	Uneseni podaci se poništavaju

SCENARIJ 3-Evidentiranje časa

Naziv slučaja upotrebe	Evidentiranje časa
Opis slučaja upotrebe	Nastavnik evidentira čas(upisuje naziv tematske jedinice,redni broj časa po rasporedu,redni broj časa u godini,tip časa,odsutne učenike)
Vezani zahtjevi	/
Preduslovi	Uspješna prijava na nalog
Posljedice-uspješan završetak	Čas je evidentiran
Posljedice-neuspješan završetak	Poruka o greški i ponovno upisivanje
Primarni akteri	Nastavnik
Ostali akteri	/
Glavni tok	Nakon sto izabere odjeljenje nastavnik može odabrati opciju evidentiraj čas. Popunjava sljedeće podatke: naziv tematske jedinice,redni broj časa po rasporedu i u godini,tip časa(obrada,utvrđivanje,pismena zadaća),odsutni učenici.Vrši se provjera da li su svi podaci popunjeni.Ako je provjera uspješna vraća se potvrda o evidentiranju časa, u suprotnom se javlja poruka o greški i traži se ponovni unos podataka
Proširenja/Alternative	/

Tok događaja 1 (uspješan zadatak)

Nastavnik	Sistem
1. Nastavnik se prijavljuje na nalog	Sistem provjerava podatke
2. Nastavnik bira odjeljenje	
3. Nastavnik bira opciju evidentiraj čas	
4. Nastavnik unosi podatke	Sistem provjerava da li su svi podaci uneseni
	5. Sistem potvrđuje ispravnost i evidentira čas u bazi

Tok događaja 2 (neuspješan završetak)

Nastavnik	Sistem
1. Nastavnik se prijavljuje na nalog	Sistem provjerava podatke
2. Nastavnik bira odjeljenje	
3. Nastavnik bira opciju evidentiraj čas	
4. Nastavnik unosi podatke	Sistem provjerava da li su svi podaci uneseni
	5. Čas nije evidentiran

SCENARIJ 4-kreiranje naloga za nastavnika

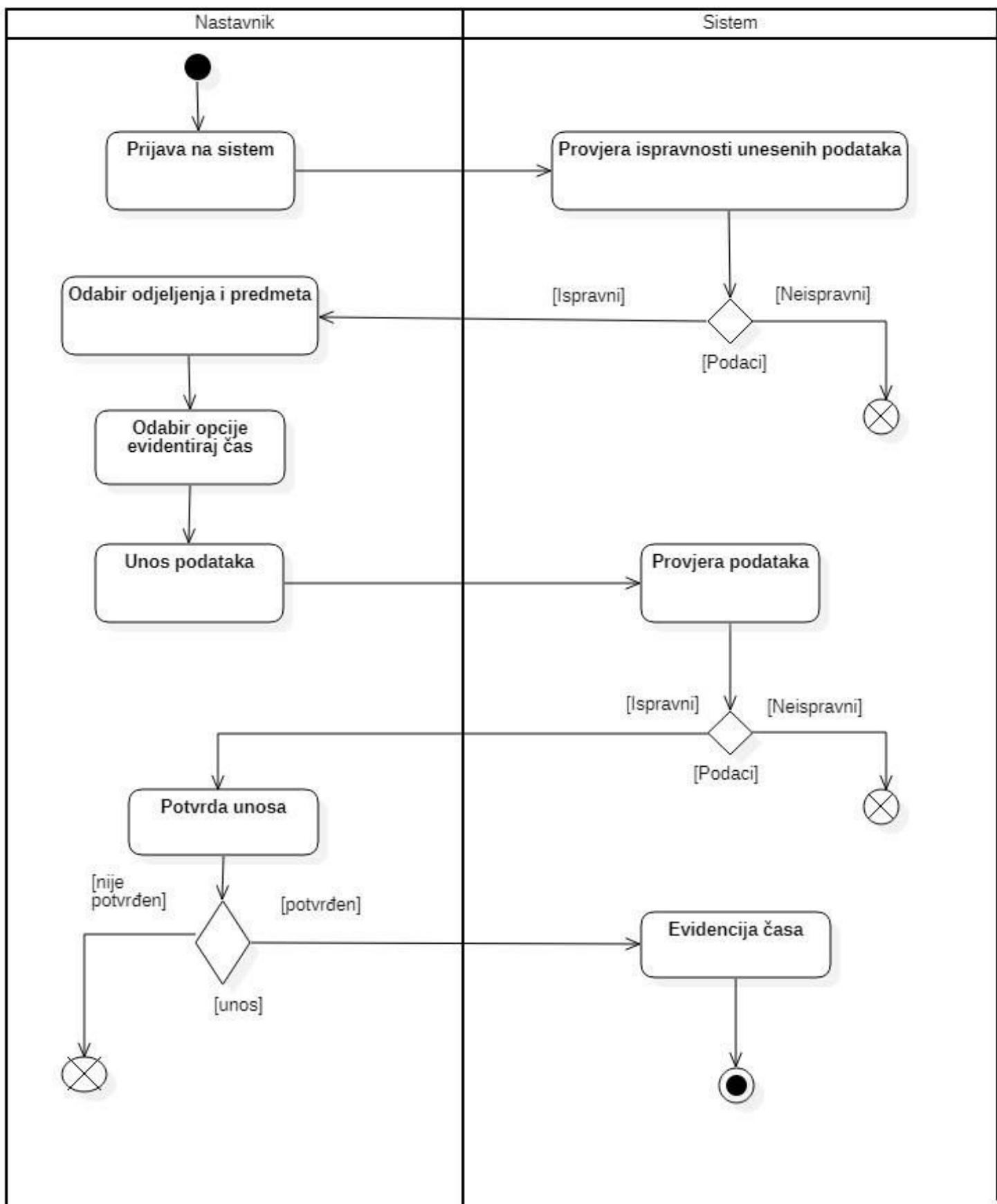
Naziv slučaja upotrebe	Kreiranje naloga za nastavnika
Opis slučaja upotrebe	Kreiranje naloga za nastavnika čiji su osnovni podaci:ime,prezime,datum rođene,JMBG,predmet koji predaje,odjeljenje kojem je razrednik
Vezani zahtjevi	/
Preduslovi	/
Posljedice-uspješan završetak	Kreiran nalog
Posljedice-neuspješan završetak	Prijava greške i ponovni unos
Primarni akteri	administrator
Ostali akteri	/
Glavni tok	Administrator bira opciju kreiraj nalog za nastavnika, nakon čega se pojavi prozor sa osnovnim podacima za nastavnika.Zatim on unosi podatke i vrši se provjera validnosti tih podataka. Ako je unos validan traži se potvrda administratora za kreiranje naloga, u suprotnom se javlja greška i traži ponovni unos podataka.
Proširenja/Alternative	/

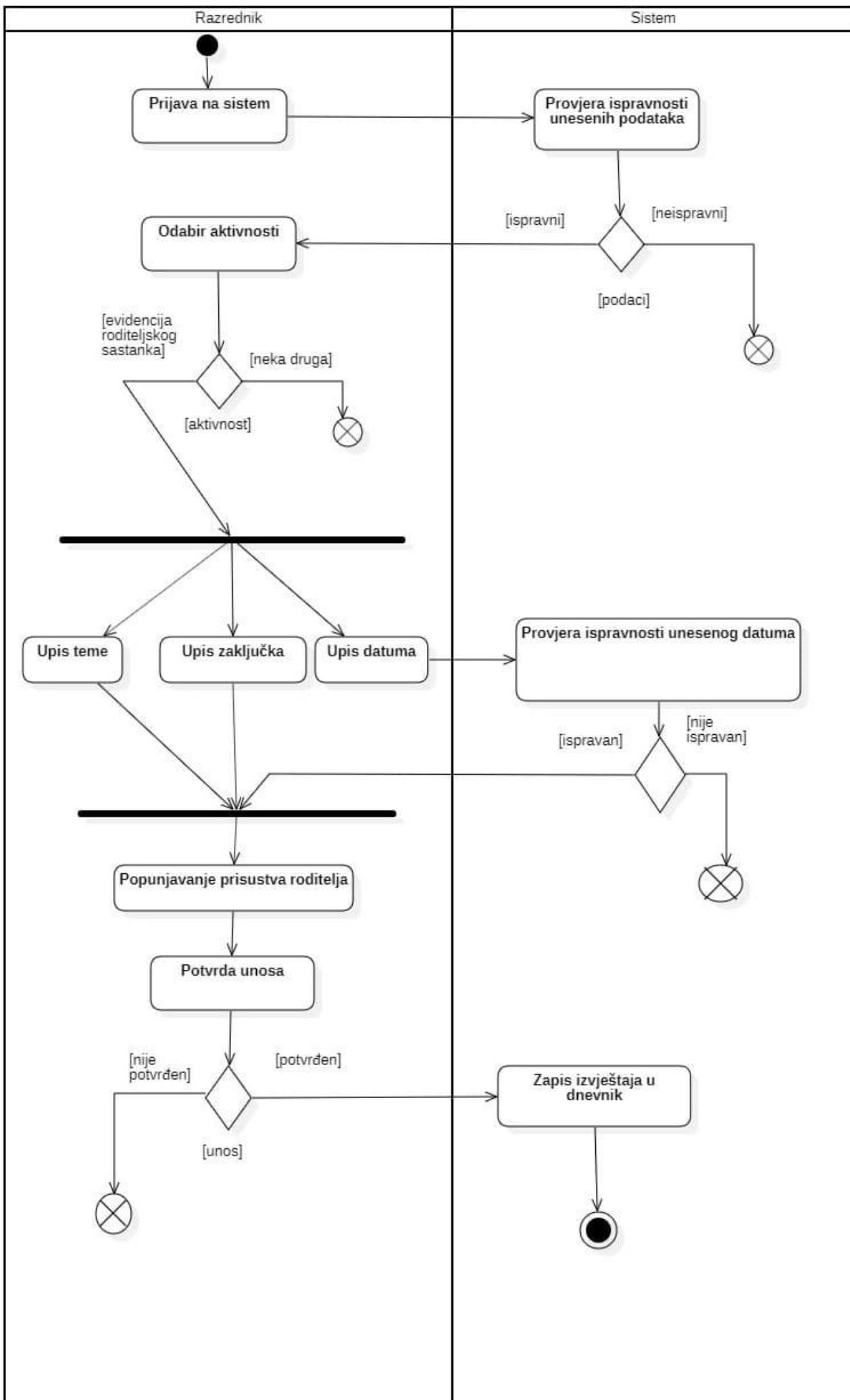
Tok događaja 1 (uspješan zadatak)

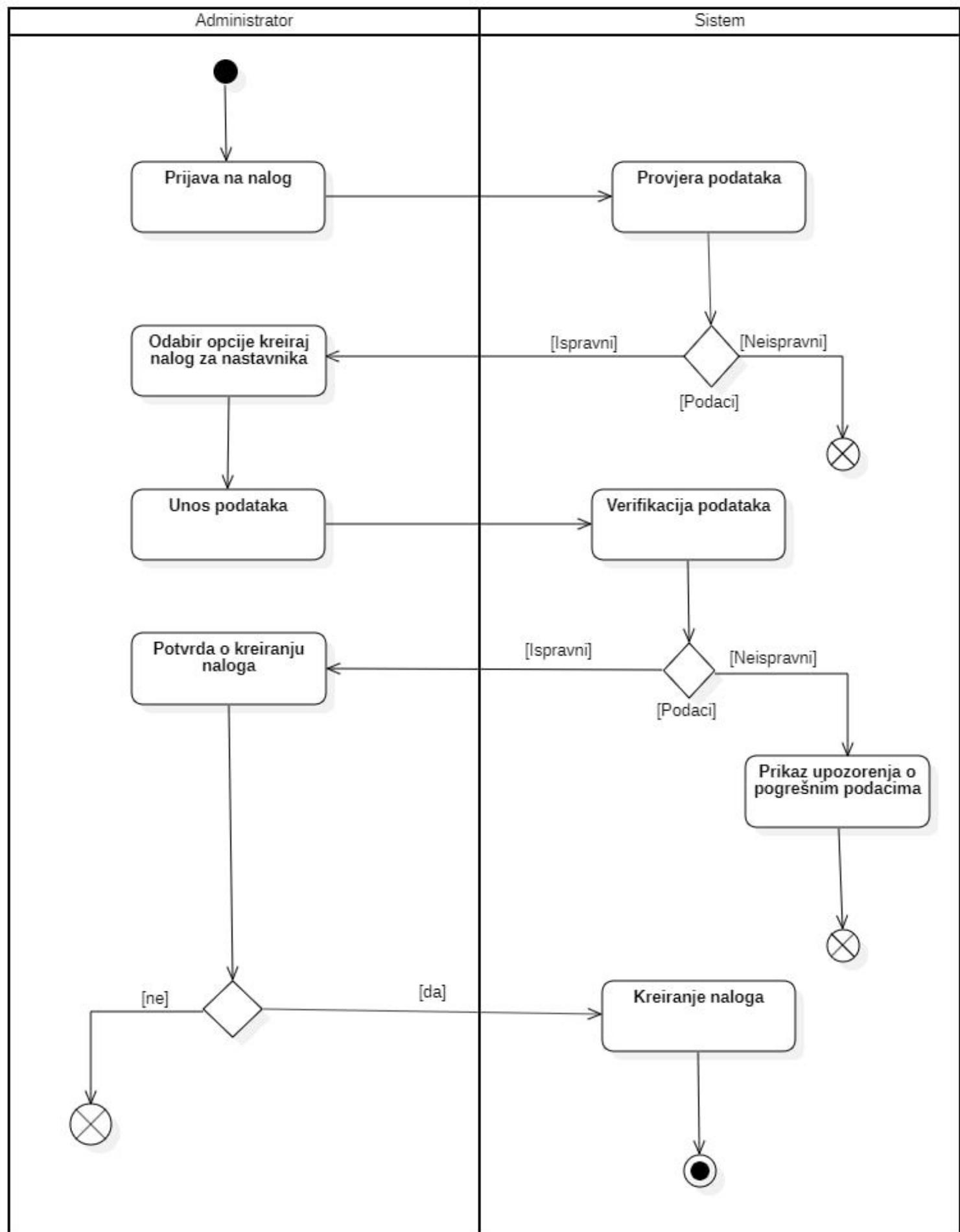
Administrator	Sistem
1. Administrator bira opciju kreiraj nalog za nastavnika	
2. Administrator unosi podatke	Sistem vrši validaciju podataka
3.	Sistem potvrđuje validnost podataka i traži potvrdu administratora da želi kreirati nalog
4. Administrator potvrđuje da želi kreirati nalog	Nalog se kreira,nastavnik se evidnetira u bazi podataka

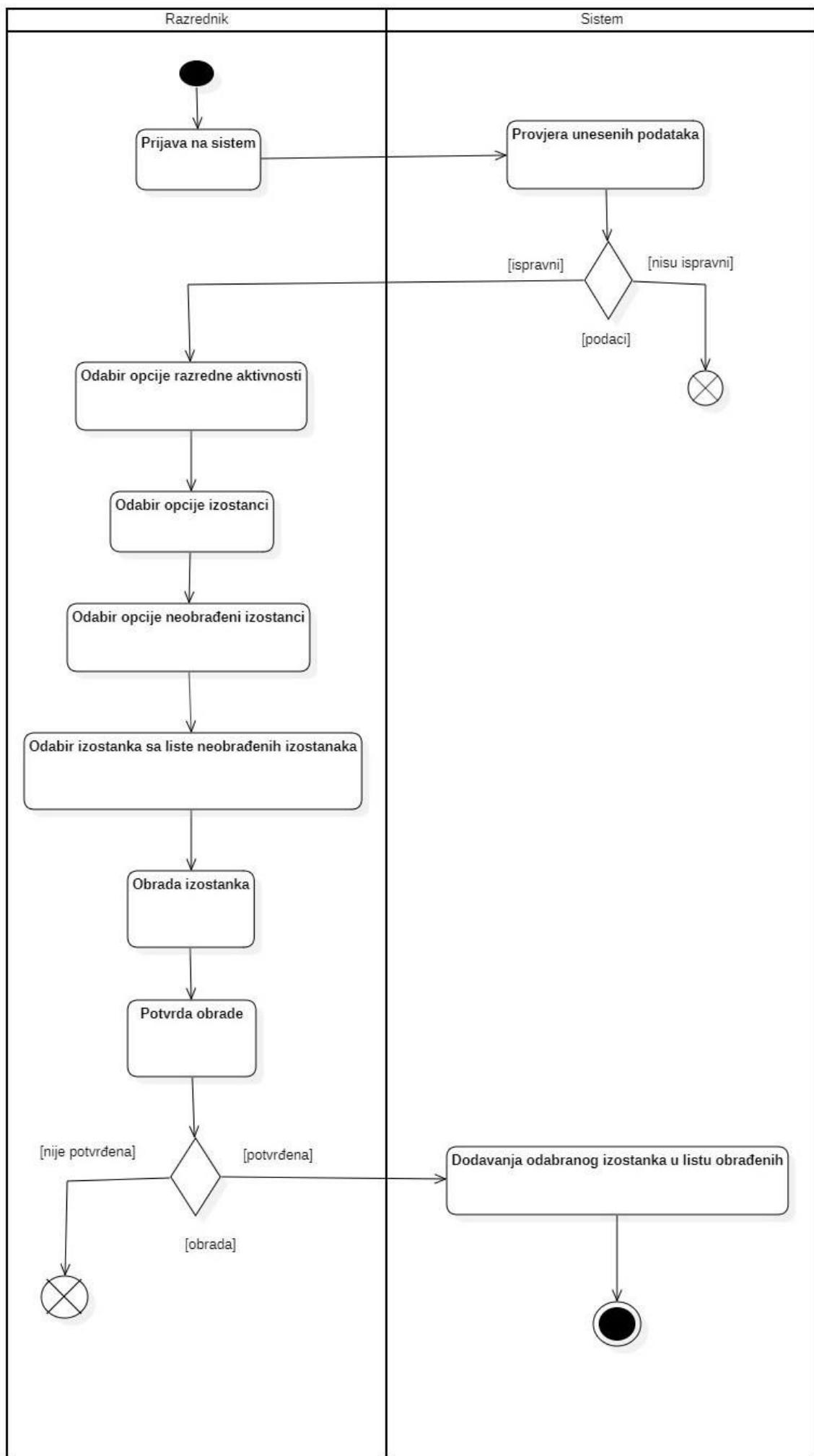
Tok događaja 2 (neuspješan završetak)

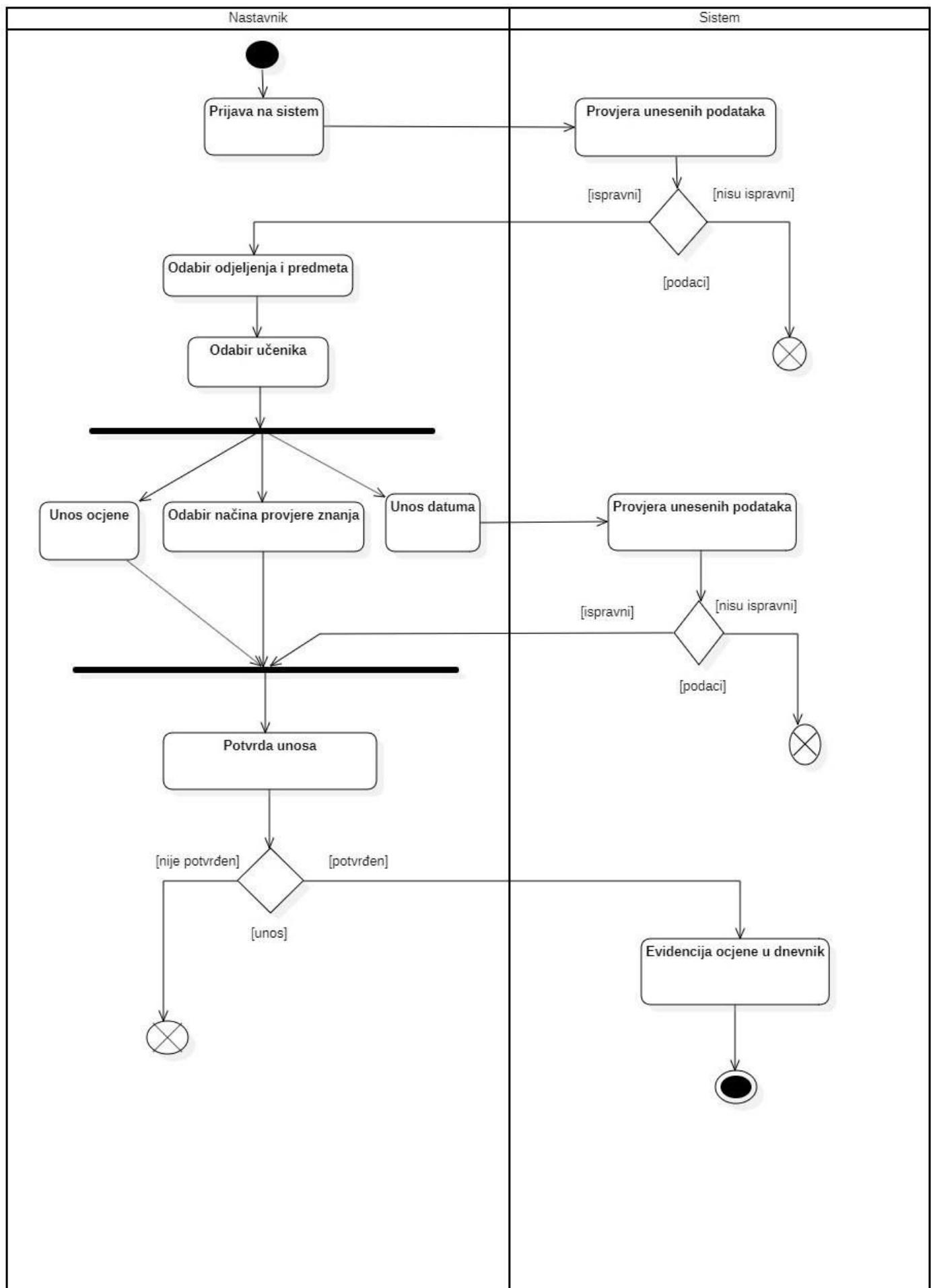
Administrator	Sistem
1. Administrator bira opciju kreiraj nalog za nastavnika	
2. Administrator unosi podatke	Sistem vrši validaciju podataka
	3. Sistem javlja poruku o greški i traži ponovni unos podataka













<https://E-dnevnik.com>



E-dnevnik

Korisničko ime

Šifra

Prijavi se





<https://E-dnevnik.com>



Prijava

Korisničko
š

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)





<https://E-Dnevnik.com>



Dobrodošli, ime prezime učenika

Ocjene

Izostanci

Ponašanje

Pismene zadaće





Ocjene

Ime predmeta, ime i prezime nastavnika ▼

1. polugodište

10.10.2019. (1) (T)
15,10.2019.(4) (U)

2. polugodište

10.02.2020(5)(U)

Zaključna ocjena na polugodištu:

Konačna zaključna ocjena:





https://



Izostanci

Period

Datum	Opravdano	Razlog
DD.MM.YYYY	DA	Bolest
DD.MM.YYYY	NE	Bez razloga

Ukupno izosotanka:

Opravdano:

Neopravdano:





<https://>



Ponašanje

Vladanje

Napomene





https://



Testovi/pismene zadaće

Period

ComboBox



Predmet	datum	Tip
Predmet 1	DD.MM.YYYY	pismena zadaća
Predmet 2	DD.MM.YYYY	test





Dobro došli, ime i prezime!



Odjeljenje

Upiši čas



Testovi/Pismene zadaće

Upiši ocjenu



Napiši napomenu



Pregled ocjena



Razredne aktivnosti





<https://E-dnevnik.com>



Upiši ocjenu



Učenik

/ /



Ocjena

Način provjere znanja

Odustani

Upiši





Upiši ocjenu



Učenik



Potvrda

Da li ste sigurni da želite upisati
ocjenu?

Da

Ne

Odustani

Upiši





<https://E-dnevnik.com>



Upiši ocjenu



Učenik



Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)

Način provjere znanja



Odustani

Upiši





Upiši ocjenu



Učenik



Uspješan upis ocjene!

[Zatvori](#)

Način provjere znanja



Odustani

Upiši





<https://E-dnevnik.com>



Napomena



/ /



Potvrdi





Napomena

// 

Potvrda

Da li ste sigurni da želite napisati napomenu?

Da

Ne

Potvrdi





<https://E-dnevnik.com>



Napomena



Uspješan unos napomene!

[Zatvori](#)

Potvrди





Upiši čas

Pregledaj prethodne

Naziv tematske jedinice

Odustni učenici

Učenik ▾

Dodaj

Tip časa

Beglerović Vildana
Buturović Lejla

Redni broj časa u godini

Redni broj časa po rasporedu

3 

Odustani

Upiši





Upiši čas

[Pregledaj prethodne](#)

Naziv tematske jedinice

Tip časa

Redni broj časa u godini

Redni broj časa po rasporedu

Potvrda

Da li ste sigurni da želite upisati
čas?

Učenik ▾

Dodaj

Beglerović Vildana
Buturović Lejla

Da

Ne

Odustani

Upiši





Održani časovi

Tematska jedinica	Tip časa	Redni broj	Odsutni učenici	
Sabiranje i oduzimanje polinoma	obrada	1	Buturović Lejla Beglerović Vildana	
Sabiranje i oduzimanje polinoma	utvrđivanje	2	Beglerović Vildana	
Množenje i djeljenje polinoma	obrada	3		





Testovi/pismene zadaće

Datum

Datum	Tip
22.4.2020.	TEST
25.5.	PISMENA ZADAĆA

Zakaži

Otkaži





Testovi/pismene zadaće

Datum

Potvrda

Da li ste sigurni da želite otkazati
test/pismenu zadaću?

ZADAĆA

Da | Ne





Testovi/pismene zadaće

Datum

Potvrda

Da li ste sigurni da želite zakazati
test/pismenu zadaću?

ZADAĆA

Da Ne





Pregled ocjena



Učenik	Prvo polugodište	ZO	Drugo polugodište	KZO
Ahmetspahić Negra	14.10.(5)(A) 25.10.(5)(T)	5	25.02.(5)(T)11.3.(5)(T)	/
Beglerović Vedad	14.10.(5)(A) 25.10.(5)(T)	5	25.02.(5)(T)	/
Beglerović Vildana	25.10.(5)(T)	5	25.02.(5)(T)	/
Buturović Lejla	25.10.(5)(T)	5	25.02.(5)(T) 17.3.(5) (A)	/





Ime i prezime učenika

Prvo polugodište

Ocjena 1
Ocjena 2
Ocjena 3

Drugo polugodište

Ocjena 1
Ocjena 2
Ocjena 3

Zaključna ocjena:

Zaključi ocjenu na polugodištu

Zaključi konačnu ocjenu

Konačna zaključna ocjena:





Ime i prezime učenika

Prvo polugodište

Ocjena 1
Ocjena 2
Ocjena 3

Drugo polugodište

Zaključna ocjena na polugodištu

Izaberite ocjenu:

Zaključna ocjena:

[Odustani](#) [Zaključi](#)

Zaključi konačnu ocjenu

Konačna zaključna ocjena:





Ime i prezime učenika

Prvo polugodište

Ocjena 1
Ocjena 2
Ocjena 3

Drugo polugodište

Konačna zaključna ocjena

Izaberite ocjenu

Zaključna ocjena:

[Odustani](#) [Zaključi](#)

Zaključi konačnu ocjenu

Konačna zaključna ocjena:





<https://>



Podaci o učenicima

Ocjene i vladanje i izostanci učenika

Zapisnici sa roditeljskih sastanaka

Zaključivanje općeg uspjeha učenika

Evidencija vladanja učenika

Pravdanje izostanaka

Evidencija roditeljskog sastanka





https://



Podaci o učenicima

Ime i prezime ▾	Datum rođenja ▾	Mjesto rođenja ▾	JMBG ▾	Broj telefona ▾	Ime oca ▾	Ime majke ▾
Selma Selmanović	07.04.1999.	Sarajevo	999999999999999	9999999999		
Učenik 2						
Učenik 3						
Učenik 4						



Ocjene, vladanje i izostanci učenika

Učenik:

Vladanje:

Datum izmjene:

Ime predmeta	Prvo polugodište	ZO	Drugo polugodište	KZO
Matematika	17.09(5)(A) 09.10(4)(T)	5		
Biologija				
Hemija				
Engleski jezik				
Njemački jezik				
BHS				

Predmet	Datum



Zapisnici sa roditeljskih sastanaka

Sastanak:

09.09.2020.
15.10.2020.
30.10.2020.
15.11.2020.

Naziv teme:

Datum:

Zaključak:

Učenici čiji su roditelji prisutni:

Item One
Item Two
Item Three

Učenici čiji su roditelji odsutni:

Item One
Item Two
Item Three





<https://>



Zaključivanje općeg uspjeha učenika

Učenik ▾

Prvo polugodište

ZO iz predmeta1
ZO iz predmeta2
ZO iz predmeta3

Zaključna ocjena:

[Zaključi ocjenu na polugodištu](#)

Drugo polugodište

ZO iz predmeta1
ZO iz predmeta2
ZO iz predmeta3

Zaključna ocjena:

[Zaključi konačnu ocjenu](#)





https://



Evidencija vladanja učenika

Učenik: ▾

Vladanje: ▾
Vrlodobro
Dobro
Zadovoljava
Loše

Datum:





https://



Pravdanje izostanaka

Učenik:

▼

Neopravdani izostanci:

- Item One
- Item Two
- Item Three

Opravdaj





https://



Pravdanje izostanaka



Uč

Potvrda

Jeste li sigurni da želite opravdati izostanke?

Ne

Da

Opravdaj



Neopravdani izostanci:

- Item One
- Item Two
- Item Three





https://



Evidencija roditeljskog sastanka

Naziv teme:

Datum:

 / 

Zaključak:

Evidentiraj prisustvo

 Evidentiraj sastanak





https://



Evidencija roditeljskog sastanka

Naziv teme:

Datum:

 / /

Zaključak:

Potvrda

Jeste li sigurni da želite
evidentirati sastanak?

Ne

Da

Evidentiraj prisustvo

Evidentiraj sastanak





https://



Evidencija prisustva roditelja

Ime i prezime učenika	Prisutan
Vildana Beglerović	<input checked="" type="checkbox"/>
Vedad Beglerović	<input checked="" type="checkbox"/>
Amina Alagić	<input type="checkbox"/>
Lejla Buturović	<input checked="" type="checkbox"/>





<https://E-Dnevnik.com>



Dobrodošli, ime prezime roditelja

Ime i prezime učenika



Ocjene

Izostanci

Ponašanje

Pismene zadaće





Ocjene

Ime predmeta, ime i prezime nastavnika ▼

1. polugodište

10.10.2019. (1) (T)
15,10.2019.(4) (U)

2. polugodište

10.02.2020(5)(U)

Zaključna ocjena na polugodištu:

Konačna zaključna ocjena:





https://



Izostanci

Period

Datum	Opravdano	Razlog
DD.MM.YYYY	DA	Bolest
DD.MM.YYYY	NE	Bez razloga

Ukupno izosotanka:

Opravdano:

Neopravdano:





<https://>



Ponašanje

Vladanje

Napomene





https://



Testovi/pismene zadaće

Period

ComboBox



Predmet	datum	Tip
Predmet 1	DD.MM.YYYY	pismena zadaća
Predmet 2	DD.MM.YYYY	test





<https://E-Dnevnik.com>



Administrator



Kreiraj nalog

Formiraj odjeljnine

Ažuriraj podatke

Ukloni nalog





<https://>



Kreiraj nalog

Kreiraj nalog za učenika 

Kreiraj nalog za nastavnika 

Kreiraj nalog za roditelja 





https://



Kreiraj nalog za učenika

Ime

JMBG

Prezime

Adresa

Roditelji

 ComboBox ▾

Broj telefona

Datum rođenja

 / /

Šifra

Mjesto rođenja

[Kreiraj nalog za roditelja](#)[Dodijeli korisničko ime](#)[Dodijeli šifru](#)[Kreiraj nalog](#)



https://



Kreiraj nalog za učenika

Ime

JMBG

Prezime

Adresa

Roditelji

 ComboBox ▾

Broj

 ComboBox ▾

Korisničko ime

Datum rođenja

 / /

Šifra

Mjesto rođenja

Šifra

Kreiraj nalog za roditelja

Dodijeli korisničko ime

Dodijeli šifru

Kreiraj nalog

Potvrda
Da li želite kreirati nalog?

Da

Ne



//



https://



Kreiraj nalog za učenika

Ime

Prezime

Roditelji

 ComboBox ▾ ComboBox ▾

Datum rođenja

 / /

Šifra

Mjesto rođenja

[Kreiraj nalog za roditelja](#)

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)





<https://>



Kreiraj nalog za učenika

Ime

Prezime

Roditelji

 ComboBox ▾ ComboBox ▾

Datum rođenja

 / /

Šifra

Mjesto rođenja

[Kreiraj nalog za roditelja](#)

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)

Uspješno ste kreirali nalog!

[Zatvori](#)





https://



Kreiraj nalog za nastavnika

Ime

JMBG

Prezime

Adresa

Predmet

 ComboBox ▼

Broj telefona

Datum rođenja

 / /

Korisničko ime

Mjesto rođenja

Šifra

Dodijeli korisničko ime

Dodijeli šifru

Kreiraj nalog





https://



Kreiraj nalog za nastavnika

Ime

JMBG

Prezime

Predmet

 ComboBox ▾

Datum rođenja

 / /

Mjesto rođenja

Adresa

Broj

Potvrda

Da li želite kreirati nalog?

Da

Ne

Korisničko ime

Šifra

Dodijeli korisničko ime

Dodijeli šifru

Kreiraj nalog





https://



Kreiraj nalog za nastavnika

Ime

JMBG

Prezime

Predmet

 ComboBox ▼

Datum rođenja

 / /

Mjesto rođenja

Adresa

Broj

Ko

Šifra

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)[Dodijeli korisničko ime](#)[Dodijeli šifru](#)[Kreiraj nalog](#)



<https://>



Kreiraj nalog za nastavnika

Ime

Prezime

Predmet

 ComboBox ▾

Datum rođenja

 / /

Mjesto rođenja

Šifra

[Zatvori](#)

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)





https://



Kreiraj nalog za roditelja

Ime

Adresa

Prezime

Broj telefona

Datum rođenja

 / /

Korisničko ime

Mjesto rođenja

Šifra

JMBG

Dodijeli korisničko ime

Dodijeli šifru

Kreiraj nalog





<https://>



Kreiraj nalog za roditelja

Ime

Adresa

Prezime

Datum rođenja

/ /



Mjesto rođenja

JMBG

Šifra

Potvrda

Da li želite kreirati nalog?

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)





<https://>



Kreiraj nalog za roditelja

Ime

Prezime

Datum rođenja

 /

Mjesto rođenja

JMBG

Adresa

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)





<https://>



Kreiraj nalog za roditelja

Ime

Prezime

Datum rođenja

 / 

Mjesto rođenja

JMBG

Uspješno ste kreirali nalog!

[Zatvori](#)

[Dodijeli korisničko ime](#)

[Dodijeli šifru](#)

[Kreiraj nalog](#)





<https://>



Formiraj odjeljenje

Razred

3	
---	--

Razrednik

Ime i prezime nastavnika	
<hr/>	

Odjeljenje

3	
---	--

Učenik 1

Učenik 2

Učenik 3

search

Dodaj učenika

Ukloni učenika

Odaberi predmete

Kreiraj raspored

Formiraj odjeljenje





https://



Formiraj odjeljenje

Razred

3	
---	--

Razrednik

Ime i prezime nastavnika

--

Odjeljenje

3	
---	--

Učenik 1

Učenik 2

Učenik 3

Uspješno ste ažurirali podatke!

[Zatvori](#)

Odaberi predmete

Kreiraj raspored

Formiraj odjeljenje





Odaberi predmete

Predmet

Matematika ▼

Nastavnik

Nastavnik1 ▼

PREDMET	NASTAVNIK
Matematika	Nastavnik 1
BHS jezik i književnost	Nastavnik 2

Dodaj predmet

Ukloni predmet

Sačuvaj





Odaberi predmete

Predmet

Matematik

Nastavnik

Nastavnik

Potvrda

Da li želite sačuvati
promjene?

Da

Ne

Dodaj predmet

	NASTAVNIK
	Nastavnik 1
iževnost	Nastavnik 2

Ukloni predmet

Sačuvaj





Odaberi predmete

Predmet

Matematički

Nastavnik

Nastavnik 1

Dodaj predmet

NASTAVNIK	
ost	Nastavnik 1
ost	Nastavnik 2

Uspješno ste ažurirali

[Zatvori](#)

Ukloni predmet

Sačuvaj





https://



Kreiraj/Izmijeni raspored

Predmet

Nastavnik

Predmet 1:

Predmet 2:

Predmet 3:

Sačuvaj promjene





<https://>



Kreiraj/Izmijeni raspored

Predmet

Predmet 1:

Predmet 2:

Predmet 3:

Nastavnik

Uspješno ste ažurirali podatke!

[Zatvori](#)

[Sačuvaj promjene](#)





Raspored V-1

DAN	PONEDJELJAK	UTORAK	SRIJEDA	ČETVRTAK	PETAK
1	Matematika	Likovna kultura	Matematika	Engleski jezik	BHS jezik i književnost
2	Muzička kultura	BHS jezik i književnost	Priroda	Društvo	Matematika
3	Engleski jezik	Priroda	TIZO	BHS jezik i književnost	Kultura življenja
4	BHS jezik i književnost	Društvo		Matematika	TIZO

Izmjeni





<https://>

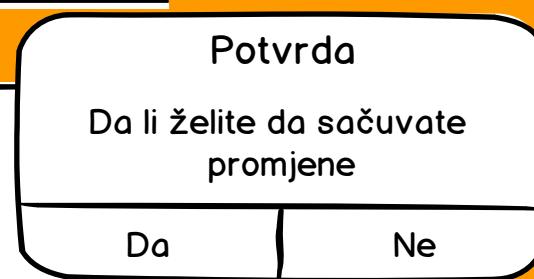


Ažuriraj odjeljenje

Razrednik

Ime i prezime nastavnika

Učenik 1
Učenik 2
Učenik 3



Ukloni učenika

Ukloni učenika

Izmijeni predmete

Izmijeni raspored

Sačuvaj promjene





https://



Predmeti i nastavnici

PREDMET	NASTAVNIK
BHS jezik i književnost	
Engleski jezik	
Matematika	
Priroda	
Društvo	
TIZO	
Likovna kultura	
Muzička kultura	
Kultura življenja	

Izmjeni





<https://>



Ažuriraj podatke

Učenici

Nastavnici

Roditelji

Odjeljenja





https://



Učenici

Odjeljenje

ComboBox ▼

search

Ime i prezime	Roditelj	Roditelj	Datum rođenja	Mjesto rođenja	JMBG	Username	Password
Učenik 1							
Učenik 2							

Ažuriraj učenika





https://



Nastavnici

search

Ime i prezime	Datum rođenja	Mjesto rođenja	JMBG	Predmet	Username	Password
Nastavnik 1						
Nastavnik 2						

Ažuriraj nastavnika





<https://>



Roditelji

search

Ime i prezime	Datum rođenja	Mjesto rođenja	JMBG	Username	Password
Roditelj 1					
Roditelj 2					

Ažuriraj roditelja





https://



Odjeljenja

Odjeljenje	Razrednik	Broj učenika
VII-2	Ime i prezime	20
III-1	Ime i prezime	25

Obriši odjeljenje

Ažuriraj odjeljenje





<https://>



Ažuriraj odjeljenje

Razrednik

Učenik 1
Učenik 2
Učenik 3

search

Dodaj učenika

Ukloni učenika

Izmijeni predmete

Izmijeni raspored

Sačuvaj promjene





<https://>



Button

Ažuriraj odjeljenje

Razrednik

Učenik 1
Učenik 2
Učenik 3

Uspješno ste ažurirali podatke!

enika
enika
enika

[Zatvori](#)

Izmijeni predmete

Izmijeni raspored

Sačuvaj promjene





https://



Ažuriraj nalog za učenika

Ime

JMBG

Prezime

Adresa

Roditelji

 ComboBox ▾

Broj telefona

Datum rođenja

 / /

Šifra

Mjesto rođenja

Korisničko ime

[Kreiraj nalog za roditelja](#)[Dodijeli šifru](#)[Sačuvaj promjene](#)



https://



Ažuriraj nalog za učenika

Ime

JMBG

Prezime

Adresa

Roditelji

 ComboBox ▾

Broj

 ComboBox ▾

Korisničko ime

Datum rođenja

 / /

Šifra

Mjesto rođenja

Potvrda

Da li želite sačuvati
promjene?

Da

Ne

[Kreiraj nalog za roditelja](#)[Dodijeli šifru](#)[Sačuvaj promjene](#)



https://



Ažuriraj nalog za učenika

Ime

Prezime

Roditelji

 ComboBox ▾ ComboBox ▾

Datum rođenja

 / /

Šifra

Mjesto rođenja

Kreiraj nalog za roditelja

Dodijeli šifru

Sačuvaj promjene





https://



Ažuriraj nalog za učenika

Ime

Prezime

Roditelji

 ComboBox ▾ ComboBox ▾

Datum rođenja

 / /

Šifra

Mjesto rođenja

Kreiraj nalog za roditelja

Dodijeli šifru

Sačuvaj promjene





https://



Ažuriraj nalog za nastavnika

Ime

JMBG

Prezime

Adresa

Predmet

 ComboBox ▼

Broj telefona

Datum rođenja

 / /

Korisničko ime

Mjesto rođenja

Šifra

Dodijeli šifruSačuvaj promjene



https://



Ažuriraj nalog za nastavnika

Ime

JMBG

Prezime

Ad

Predmet

 ComboBox ▼

Br

Datum rođenja

 / /

Korisničko ime

Mjesto rođenja

Šifra

Potvrda
Da li želite sačuvati
promjene?

Da

Ne

Dodijeli šifru

Sačuvaj promjene





https://



Ažuriraj nalog za nastavnika

Ime

Prezime

Predmet

 ComboBox ▾

Datum rođenja

 / /

Mjesto rođenja

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)

Šifra

[Dodijeli šifru](#)

[Sačuvaj promjene](#)





<https://>



Ažuriraj nalog za nastavnika

Ime

Prezime

Predmet

 ComboBox ▾

Datum rođenja

 / /

Mjesto rođenja

Šifra

[Zatvori](#)

[Dodijeli šifru](#)

[Sačuvaj promjene](#)





https://



Ažuriraj nalog za roditelja

Ime

Adresa

Prezime

Broj telefona

Datum rođenja

 / /

Korisničko ime

Mjesto rođenja

Šifra

JMBG

Dodijeli šifru

Sačuvaj promjene





<https://>



Ažuriraj nalog za roditelja

Ime

Adresa

Prezime

Datum rođenja

/ /



Mjesto rođenja

JMBG

Šifra

Potvrda

Da li želite sačuvati
promjene?

Da

Ne

Dodijeli šifru

Sačuvaj promjene





<https://>



Ažuriraj nalog za roditelja

Ime

Prezime

Datum rođenja

 /

Mjesto rođenja

JMBG

Uspješno ste ažurirali podatke!

[Zatvori](#)

[Dodijeli šifru](#)

[Sačuvaj promjene](#)





<https://>



Ažuriraj nalog za roditelja

Ime

Prezime

Datum rođenja

 /

Mjesto rođenja

JMBG

Neispravni podaci!

Pokušajte ponovo.

[Zatvori](#)

[Dodijeli šifru](#)

[Sačuvaj promjene](#)





<https://>



Ukloni nalog

 search

Ukloni nalog





<https://>



Ukloni nalog

search

Ukloni nalog

Da li želite ukloniti nalog

Ne

Da





<https://>



Ukloni nalog

[search](#)

Uspješno ste uklonili nalog!

[Zatvori](#)



Analiza sistema

Skola

Atributi:

- ucenici: List(Ucenik)
- roditelji: List(Roditelj)
- odjelenja: List(Odjelenje)
- nastavnici: List(Nastavnik)
- predmeti: List(Predmet)
- razredi: List(Razred)
- administrator: Administrator
- pocetakPrvogPolugodista: Date
- pocetakDrugogPolugodista: Date
- skolskaGodina: String

Metode:

- Konstruktori
- get/set metode
- dodajUcenika(ucenik: Ucenik):void
- obrisiUcenika(ucenik:Ucenik):void
- azurirajUcenika(ucenik: Ucenik):void
- dodajRoditelja(roditelj: Roditelj):void
- azurirajRoditelja(roditelj: Roditelj):void
- obrisiRoditelja(roditelj: Roditelj):void
- dodajNastavnika(nastavnik: Nastavnik):void
- obrisiNastavnika(nastavnik: Nastavnik):void
- dodajOdjeljene(odjelenje: Odjelenje):void
- obrisiOdjeljenje(odjelenje: Odjelenje):void
- azurirajOdjeljenje(odjelenje: Odjelenje):void
- dodajPredmet(predmet: Predmet):void
- obrisiPredmet(predmet: Predmet):void
- azurirajPredmet(predmet: Predmet):void
- dodajRazred(razred: Razred):void
- obrisiRazred(razred: Razred):void
- azurirajRazred(razred: Razred):void
- zapocniNovuSkolskuGodinu(): void

Osoba je apstraktna klasa iz koje su izvedeni Ucenik, Roditelj, Nastavnik, i Administrator.

Atributi:

- id: int
- ime: String
- prezime: String
- datumRodjenja: Date
- mjestoRodjenja: String
- jmbg: String
- adresa: String
- kontakt: String
- korisnickoIme: String
- sifra: String

Metode:

- Konstruktori
- get/set metode
- (-)generisiSifru():void

Ucenik

Atributi:

- roditelj:Roditelj
- Ocjene:List(Ocjene)
- opciUspjehPrvoPolugodiste: double
- opciUspjehNaKraju: double
- vladanjePrvoPolugodiste: Vladanje
- vladanjeNaKraju: Vladanje

Metode:

- konstruktor
- get/set metode
- dajOcjeneZaPredmet(predmet:Predmet): List(Ocjena)
- dajOpravdaneIzostanke(): void
- dajNeopravdaneIzostanke(): void
- prikaziOcjene(): void
- prikaziIzostanke(): void
- pregledajPonasanje(): void
- prikaziZakazaneProvjereZnanja(): void

Nastavnik

Atributi:

- predmeti>List(Predmet)

Metode:

- konstruktor
- get/set metode
- upisiOcjenu(ucenik:Ucenik, predmet:Predmet,vrijednost:int, datum: Date, tip:tipOcjene):void
- napisaNapomenu(datum:Date, sadrzaj:String, odjeljenje:Odjeljenje):void
- zakaziProvjeruZnanja(tip:tipProvjereZnanja, datum:Date, predmet: Predmet, odjeljenje:Odjeljenje):void
- otkaziProvjeruZnanjat(provjeraZnanja, ProvjeraZnanja):void
- upisiCas(odjeljenje:Odjeljenje, tematskaJedinica:String, tip:tipCasa, redniBrojUGodini:int, redniBrojUDanu:int):void
- evidentirajIzostanak(izostanak: Izostanak):void
- zakljuciOcjenu(ucenik: Ucenik, predmet:Predmet, ocjena:int):void
- upisiCas(cas: Cas):void
- pregledajOcjene(predmet: Predmet, ucenik: Ucenik):void
- evidentirajSastanak(sastanak: RoditeljskiSastanak):void
- izaberiOdjeljenje():void
- izaberiPredmet(): void
- opravdajIzostanak(izostanak: Izostanak): void
- zakaziProvjeruZnanja(provjeraZnanja:ProvjeraZnanja): void
- otkaziProvjeruZnanja/provjeraZnanja: ProvjeraZnanja): void
- prikaziPodatkeOUceniku(ucenik: Ucenik): void
- prikaziRoditeljskeSastanke():void
- pregledajRaspored():void

Administrator

Atributi:

- jedinstvenaInstanca: Administrator
- korisnickoIme: String
- lozinka: String

Metode:

- konstruktor
- dajJedinstvenuInstancu(): Administrator
- generisiSifru(): void

Roditelj

Atributi:

Metode:

- Konstruktori
- dajRoditelja(): Roditelj

Razred

Atributi:

- id: int
- predmeti>List(Predmet)

Metode:

- Konstruktor(predmeti>List(Predmet))
- get/set metode
- dodajPredmet(predmet:Predmet)
- obrisiPredmet(predmet:Predmet)

Odjeljenje

Atributi:

- id: int
- razred:Razred
- razrednik:Razrednik
- ucenici>List(Ucenik)
- napomena>List(Napomena)
- odrzaniCasovi>List(Cas)
- raspored:Mapa(Dan, List(Predmet))
- roditeljskiSastanci>List(RoditeljskiSastanak)
- zakazaneProvjereZnanja>List(ProvjeraZnanja)
- zaključneOcjene: List(Ocjena)

Metode:

- konstruktor
- get/set metode
- upisiCas(cas: Cas): void
- zakaziProvjeru(provjera: ProvjeraZnanja): void
- otkaziProvjeru(provjera: ProvjeraZnanja) : void
- dodajNapomenu(napomena: Napomena) : void

- dodajUcenika(ucenik:Ucenik) : void
- dodajRoditeljskiSastanak(sastanak:RoditeljskiSastanak) : void
- dodajZakljucnuOcjenu(ocjena: ZakljucnaOcjena) : void

Ocjena

Atributi:

- ucenik:Ucenik
- predmet:Predmet
- vrijednost:int
- datum:Date
- tip:tipOcjene
- nastavnik: Nastavnik

Metode:

- Konstruktor
- get/set metode

Izostanak

Atributi:

- ucenik: Ucenik
- predmet: Predmet
- datum: Date
- statusIzostanka>StatusIzostanka
- razlog: String
- obradjen: bool
- odjeljenje: Odjeljenje

Metode:

- konstruktor
- get/set metode

Cas

Atributi:

- tematskaJedinica:String
- tip:tipCasa
- datum:Date
- brojCasaUGodini:int
- brojCasaUDanu:int
- predmet: Predmet
- odjeljenje: Odjeljenje
- nastavnik: Nastavnik

Metode:

- konstruktori
- get/set metode

ProvjeraZnanja

Atributi:

- tip:tipProvjereZnanja
- datum: Date
- predmet: Predmet
- odjeljenje: Odjeljenje

Metode:

- konstruktori
- get/set metode

RoditeljskiSastanak

Atributi:

- tema:String
- datum:Date
- sadržaj:String
- odjeljenje: Odjeljenje
- prisutniRoditelji: List(Roditelji)

Metode:

- konstruktor
- get/set metode

Predmet

Atributi:

- ime: String
- razred: Razred

Metode:

- konstruktor
- get/set metode

Napomena

Atributi:

- datum:Date
- sadrzaj:String
- odjeljenje: Odjeljenje

Metode:

- konstruktor
- get/set metode

Enum **tipCasa** {OBRADA, UTVRDJIVANJE, PONAVLJANJE, PROVJERA}

Enum **tipOcjene** {PZ, AKTIVNOST, USMENI, TEST, LEKTIRA}

Enum **statusIzostanka** {OPRAVDANO, NEOPRAVDANO}

Enum **vladanje** {PRIMJERNO, VRLODOBRO, DOBRO, ZADOVOLJAVA, LOŠE}

Enum **tipProvjereZnanja** {TEST, PISMENA ZADAĆA}

Enum **dan** {PONEDJELJAK, UTORAK, SRIJEDA, ČETVRTAK, PETAK}

Veze:

NASTAVNIK-RAZREDNIK: generlizacija, jer je razrednik izveden iz klase nastavnik

OSOBA-UCENIK, ADMINISTRATOR, RODITELJ, NASTAVNIK: generalizacija, jer su razrednik, roditelj, ucenik i administrator izvedeni iz klase osoba

ODJELJENJE-RAZREDNIK: generalizacija, jer je klasa odjeljenje izvedena iz klase razred

RAZREDNIK-ODJELJENJE: kompozicija, jer odjeljenje ne može postojati bez razrednika

UCENIK-ODJELJENJE: kompozicija, jer odjeljenje ne može postojati bez učenika

PREDMET-RAZRED: kompozicija, jer razred ne može postojati bez predmeta

IZOSTANAK-UCENIK: kompozicija, jer izostanak ne može postojati bez učenika

IZOSTANAK-PREDMET: kompozicija, jer izostanak ne može postojati bez predmeta

OCJENA-UCENIKA: kompozicija, jer ocjena ne može postojati bez učenika

OCJENA-PREDMET:kompozicija, jer ocjena ne može postojati bez predmeta

RODITELJSKISASTANAK-ODJELJENJE:agregacija, jer brisanjem roditeljskog sastanka, odjeljenje idalje postoji

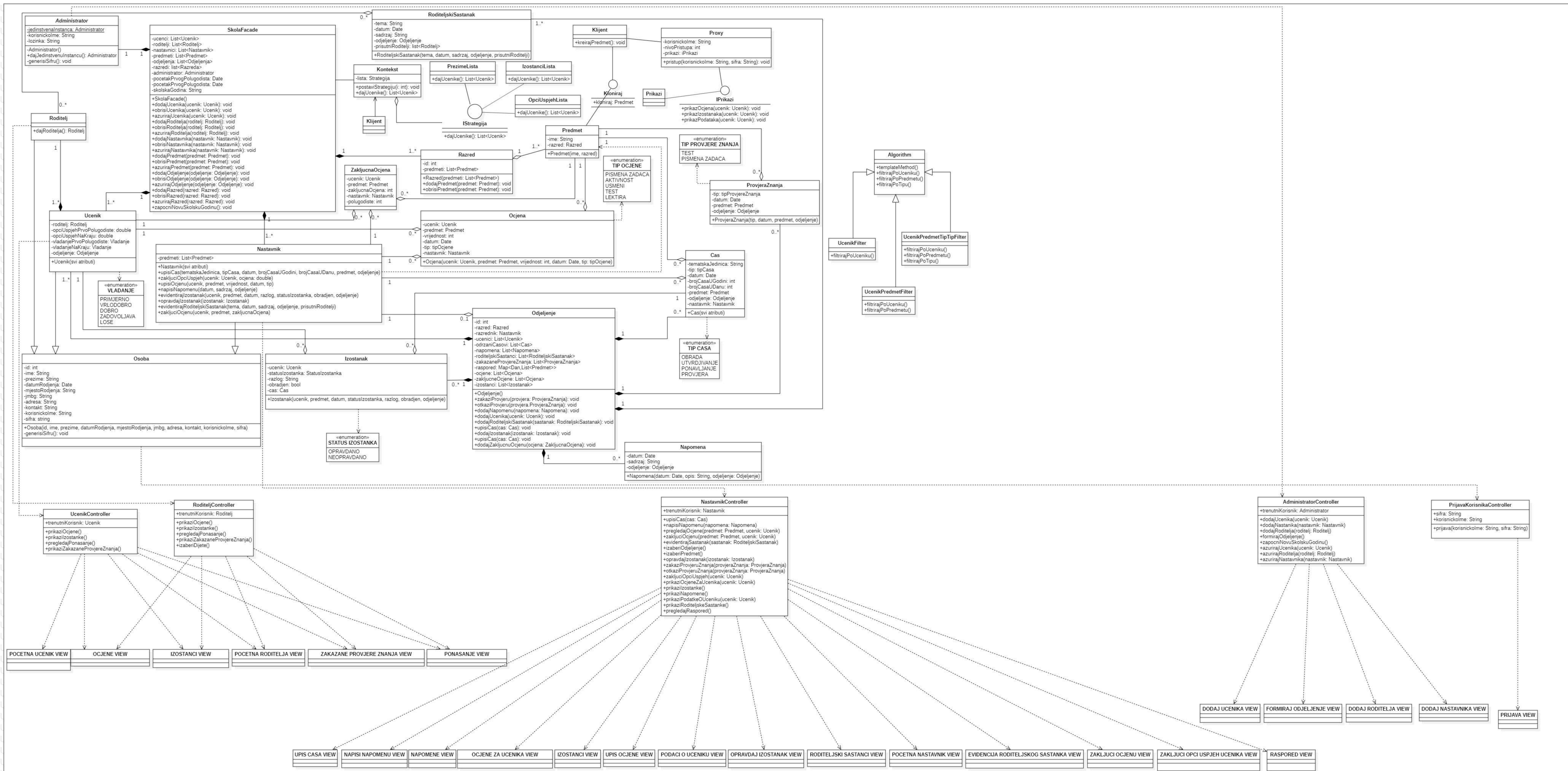
NAPOMENA-ODJELJENJE:agregacija, jer brisanjem napomene, odjeljenje idalje postoji

RODITELJ-UCENIK-agregacija, jer brisanjem roditelja učenik idalje postoji

NASTAVNIK-PREDMET:asocijacija, nastavnik predaje jedan ili više predmeta

SOLID PRINCIPI

- **Single Responsibility Principle:** „Klasa bi trebala imati samo jedan razlog za promjenu.“
Klase „Skola“ je jedina klasa čije metode bi eventualno pristupale bazi ili lokalnoj datoteci. Sve ostale imaju samo jednu odgovornost. Budući da je klasa „Skola“ svakako klasa koja je namijenjena da upravlja cijelim sistemom i ona zadovoljava ovaj princip.
- **Open/Closed Principle:** „Entiteti softvera (klase, moduli, funkcije) trebali bi biti otvoreni za nadogradnju, ali zatvoreni za modifikacije.“
Klase „Skola“ sadrži druge klase kao attribute, ali metode ove klase će se koristiti samo za dodavanje objekata u listu što ne narušava princip.
Iz apstraktne klase „Osoba“ moguće je izvesti nove klase bez potrebe da se modificira već postojeća. Takođe ova klasa je otvorena za nadogradnju.
- **Liskov principle zamjene:** „Podtipovi moraju biti zamjenjivi njihovim osnovnim tipovima.“
Klase „Nastavnik“, „Roditelj“, „Ucenik“ i „Administrator“ su nasljedene iz klase „Osoba“. Klasu „Osoba“ možemo zamijeniti svim ovim klasama. Takođe, klasa „Razrednik“ je nasljedena iz klase „Nastavnik“. Klasu „Nastavnik“ možemo zamijeniti klasom „Razrednik“. Ovaj princip je ispunjen.
- **Interface Segregation Principle:** „Klijenti ne treba da ovise o metodama koje neće upotrebljavati.“
Budući da u našem sistemu još uvijek nema interfejsa, ne dolazi do narušavanja ovog principa.
- **Dependency Inversion Principle:** „Moduli ne bi trebali ovisiti od detalja. Detalji bi trebali ovisiti od apstrakcija.“
Klase „Razrednik“ nasljeđuje klasu „Nastavnik“, koja nije apstraktna. Kako bi zadovoljili ovaj uslov sve zajedničke attribute i metode smo izdvojili u novu klasu „Osoba“ koja nam je bazna klasa i koja je apstraktna.



Strukturalni paterni

Adapter pattern

Adapter patern služi da se postojeći objekat prilagodi za korištenje na neki novi način u odnosu na postojeći rad bez mijenjanja same definicije objekta.

Ovaj patren nismo implementirali u našem sistemu. Recimo da se klasa *RoditeljskiSastanak* ne sastoji od atributa *tema* i *sadrzaj*, nego od *datuma* i *izvještaja* koji je pdf file. Dakle, u klasi *RoditeljskiSastanak* bi imali metodu *evidentiraj (izvjestaj: PDF):void*. Primjenom adapter paterna možemo omogućiti klijentu da evidentira izvještaj i u nekim drugim formatima, npr. docx bez da mijenja postojeću klasu *RoditeljskiSastanak*. Potrebno je definisati interfejs *IEvidencija* sa metodom *evidentirajDOCX():void* i klasu *RoditeljskiSastanakAdapter* koja će implementirati metodu *evidentirajDOCX* tako što će unutar nje pozvati postojeću metodu *evidentiraj* klase *RoditeljskiSastanak* nakon pretvaranja docx u pdf format.

Facade pattern (implementiran)

Ovaj patern se koristi kada sistem ima više podsistema, U našem projektu naš glavni sistem je Skola i ima podsisteme Razred, Ucenik, Odjeljenje, Nastavnik. Da bismo sakrili detalje o kreiranju i modificiranju instanca klasa podistema, dodavanja osoba u sistem, proglašit ćemo klasu Skola Facade klasom – SkolaFacade. Ova klasa komunicira sa prethodno navedenim klasama. Korisnik sistema ne mora razmišljati šta se dešava u pozadini metoda, već je dovoljno da pozove određenu metodu iz fasadne klase.

SkolaFacade
<pre>-ucenci: List<Ucenik> -roditelji: List<Roditelj> -skolskaGodina: SkolskaGodina -nastavnici: List<Nastavnik> -predmeti: List<Predmet> -administrator: Administrator -odjeljenja: List<Odjeljenje> +SkolaFacade(ucenci, roditelji, skolskaGodina, nastavnici, predmeti, administrator) +dodajUcenika(ucenik: Ucenik): void +obrisiUcenika(ucenik: Ucenik): void +azurirajUcenika(ucenik: Ucenik): void +dodajRoditelja(roditelj: Roditelj): void +azurirajRoditelja(roditelj: Roditelj): void +obrisiRoditelja(roditelj: Roditelj): void +dodajNastavnika(nastavnik: Nastavnik): void +obrisiNastavnika(nastavnik: Nastavnik): void +azurirajPredmet(predmet: Predmet): void +dodajPredmet(String predmet): void +obrisiPredmet(String predmet): void +zapocniNovuSkolskuGodinu(): void +dodajOdjeljenje(odjeljenje: Odjeljenje): void +obrisiOdjeljenje(odjeljenje: Odjeljenje): void +azurirajOdjeljenje(odjeljenje: Odjeljenje): void</pre>

Decorator patern

Decorator patern služi za omogućavanja različitih nadogradnji objektima koji svi u osnovi predstavljaju jednu vrstu objekta (odnosno, koji imaju istu osnovu). U našem sistemu nemamo klase čiji objekti bi se mogli nadograđivati, a za neke koji bi i mogli proširiti sa još detalja (recimo klasa Osoba, napočetku mogli bi zahtijevati samo osnovne podatke kao što su ime, prezime, JMBG itd., zatim to proširiti sa brojem telefona, gmailom, zatim slikom) mogli bi to puno jednostavnije postići sa običnom set metodom

Bridge patern

Ovaj patern nismo iskoristili u projektu. Mogli bi ga iskoristiti ukoliko bi aplikaciju koristile privatne škole gdje je potrebno i evidentirati školarinu. Privatne škole nude opcije prevoza ucenika ili produženi boravak što zahtijeva dodatne troškove. U tom slučaju potrebno bi bilo napraviti više klase koje bi predstavljale različite tipove učenika. Svaka klasa bi implementirala interfejs koji sadrži metode za obračun školarine. Klasa Bridge bi tada na osnovni iznos školarine koji plaćaju svi učenici dodala i iznose za svakog učenika posebno.

Composite patern

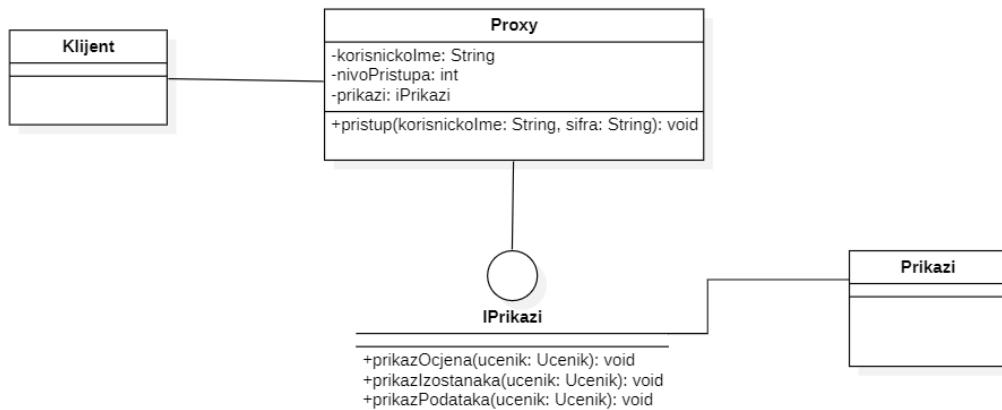
Ovaj patern nismo iskoristili u projektu. Mogli bi ga iskoristit ako je potrebno izračunati školarinu za svakog učenika posebno, na različit način. Tada je potrebno imati interfejs Iučenik koji bi ima metodu dajSkolarinu() i klasu EdnevnikComposite. Ova klasa i sve klase koje predstavljaju tip učenika bi implementovale ovaj interfejs. Metoda dajSkolarinu() u EdnevnikComposite bi obuhvatala školarine svakog učenika posebno.

Proxy pattern (implementiran)

Ovaj patern služi kako bi se kontrolisao pristup određenim objektima. Njegovom primjenom se onemogućava manipulacija objektima ukoliko neki konkretni uslov nije ispunjen. U našem projektu možemo ograničiti pristup pregledu podataka o učeniku i to na način da učenik može vidjeti isključivo svoje ocjene, podatke i izostanke. Također, roditelj može vidjeti vidjeti navedene informacije samo o svom djetu. Da bi se to realizovalo, potrebno je dodati klasu Prikazi, u kojoj će se nalaziti metode: prikazOcjena(ucenik : Ucenik) tipa void, prikazIzostanaka(ucenik: Ucenik) tipa void i prikazPodataka(ucenik: Ucenik). Ova klasa implementira interfejs iPrikazi. Zatim je potrebno dodati klasu Proxy sa atributima string username i metodom pristup (string korisnickoIme, string sifra). U ovoj metodi će se ispitati ispravnost unesenih podataka i ustanoviti vrsta korisnika i korisnicko ime.

Također, potrebno je dodati i metode prikazOcjena(ucenik: Ucenik), prikazIzostanaka(ucenik: Ucenik), prikazPodataka(ucenik: Ucenik). Ukoliko je vrsta korisnika koji pristupa Nastavnik ili Administrator, omogućeni su mu svi prikazi. Ukoliko je vrsta korisnika Učenik provjerava se da

li mu je isti username kao kod učenika kojeg prihvata metoda. Ukoliko je vrsta korisnika roditelj, provjerava se da li mu je korisničko ime jednako korisničkom imenu roditelja od učenika koji je primljen kao parametar. Ukoliko su provjere zadovoljene, poziva se odgovarajuća metoda iz klase Prikazi. Ako ipak nisu, odbija se pristup.



Flyweight pattern

Flyweight patern se koristi kako bi se onemogućilo suvišno stvaranje velikog broja instanci objekta koji svi u suštini predstavljaju jedan objekat.

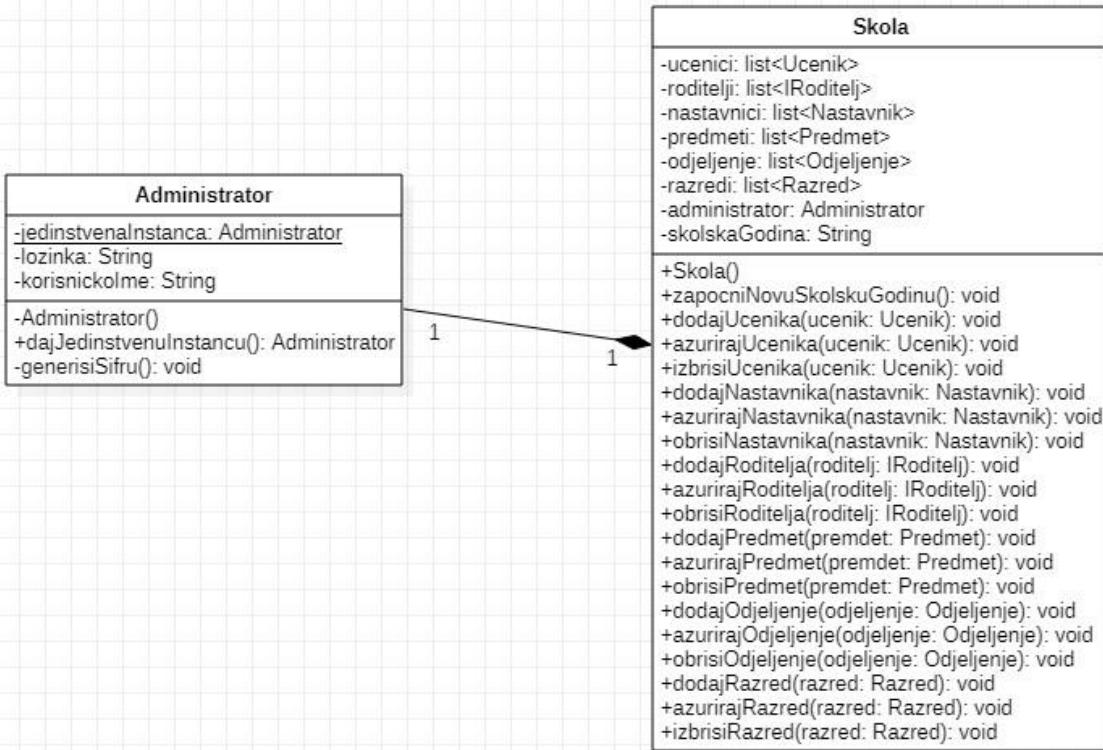
Ovaj patern nismo implementirali u našem sistemu. Naš sistem je zamišljen tako da je prilikom dodavanja novog učenika u listu u klasi *Skola*, potrebno dodati i roditelja u listu roditelja. Međutim, roditelj može da iste školske godine ili nekad kasnije upiše još djece u tu školu. Recimo da naš sistem prilikom kreiranja učenika, kreira i novu instancu roditelja bezobzira što je taj roditelj možda već upisao drugo dijete u školu nekada ranije. S obzirom da se on ni po čemu ne razlikuje od one prethodne iskoristili bismo Flyweight patern. Na dijagramu klasa bi dodali interfejs *IRoditelj* koji sadrži metodu *dajRoditelja():Roditelj*. Klasa *Roditelj* bi implementirala ovaj interfejs, što znači da bi imala metodu *dajRoditelja()*, a klasa *Skola* umjesto atributa *roditelji:list(Roditelj)* tada bi imala atribut *roditelji:list(IRoditelj)*. Umjesto metoda *dodajRoditelja(roditelj:Roditelj):void*, *azurirajRoditelja(roditelj:Roditelj):void* i *obrisiRoditelja(roditelj:Roditelj):void*, bile bi metode *dodajRoditelja(roditelj:IRoditelj):void*, *azurirajRoditelja(roditelj:IRoditelj):void* i *obrisiRoditelja(roditelj:IRoditelj):void*. Ovim bi uštedili memoriju u slučaju da postoji veći broj roditelja koji imaju više od jedno dijete.

Kreacijski paterni

Singleton pattern (implementiran)

Uloga Singleton paterna je da osigura da se neka klasa može instancirati samo jednom.

U našem sistemu takva klasa je klasa *Administrator* pa smo na nju primijenili ovaj patern kako bi uštedili memoriju. Konstruktor ove klase smo proglašili privatnim, zatim smo dodali statički atribut *jedinstvenaInstanca:Administrator* i metodu *dajJedinstvenuInstancu():Administrator* preko koje se pristupa toj jedinstvenoj instanci.

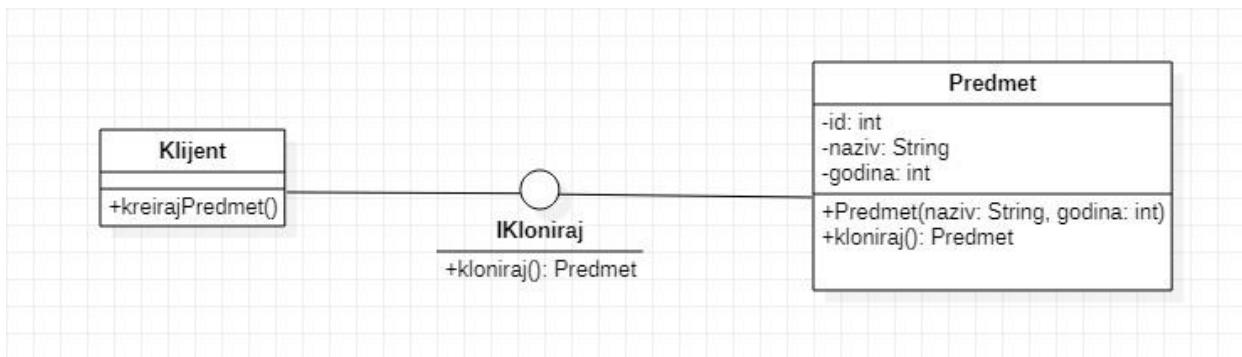


Prototype pattern (implementiran)

Uloga Prototype paterna je da kreira nove objekte klonirajući jednu od postojećih prototip instanci (postojećih objekata).

U našem sistemu ovaj patern smo primijenili na klasu *Predmet*. U toj klasi postoje tri atributa: *id:int*, *naziv:String* i *godina:int*. S obzirom na način na koji smo zamislili da funkcioniše naš sistem može postojati više predmeta koji imaju isti *naziv*, a drugačiji tip *godina*. Recimo, mogu postojati objekti čiji je *naziv* Matematika i *godina* bilo koji broj od 1 do 9.

Uz pomoć ovog paterna moguće je klonirati sve instance klase *Predmet* po istom nazivu, a zatim naknadno promijeniti atribute koji se razlikuju. Takođe, neki objekti klase *Predmet* mogu imati istu *godinu*, a različit *naziv* (svi predmeti sa iste godine). Kako bi primijenili ovaj patern dodali smo klasu *Klijent* i interfejs *IPrototype*. Klasa *Klijent* zahtjeva kloniranje postojećih objekata preko interfejsa *IKloniraj*, a klasa *Predmet* implementira kloniranje postojećih objekata.



Factory method pattern

Ovaj patern omogućava instanciranje različitih vrsta podklasa. U toku izvršavanja programa se odlučuje koja podkласa će se instancirati. Ovaj patern je moguće iskoristiti u našem projektu na način opisan u nastavku. Klase izvedene iz klase Osoba su Ucenik, Nastavnik, Roditelj. Potrebno je napraviti Creator klasu koja predstavlja klasu za pravljenje osoba. U toj klasi ćemo imati metodu FactoryMethod koja će primati sve podatke potrebne za osobu, kao i int tipOsobe koji će se odrediti na osnovu tipa korisnika koji će biti unaprijed definiran u bazi podataka. Npr. 1 – Nastavnik, 2 – Ucenik, 3 – Roditelj. Na osnovu tipa osobe će se napraviti odgovarajuća osoba. Klijent koristi FactoryMethod iz Creator klase.

Abstract factory pattern

Abstract factory pattern omogućava da se kreiraju familije povezanih objekata. Koristi se kada ne želimo da kod zavisi o konkretnim klasama tih objekata. U našem projektu ovaj patern se ne bi se mogao upotrijebiti. Hipotetički, ukoliko bi naš projekat podržavao više vrsta ocjenjivanja i školovanja (npr. po sistemu neke druge države) mogao bi se napraviti fabrički interfejs AbstractFactory i dvije konkretnе fabrike za pravljenje ocjena i provjera znanja, a koje bi implementirale interfejsе IOcjena i IprovjeraZnanja. Na taj način, ovisno o sistemu školovanja ocjene i provjere znanja bi bile odgovarajuće, a da toga klijent nije svjestan.

Builder pattern

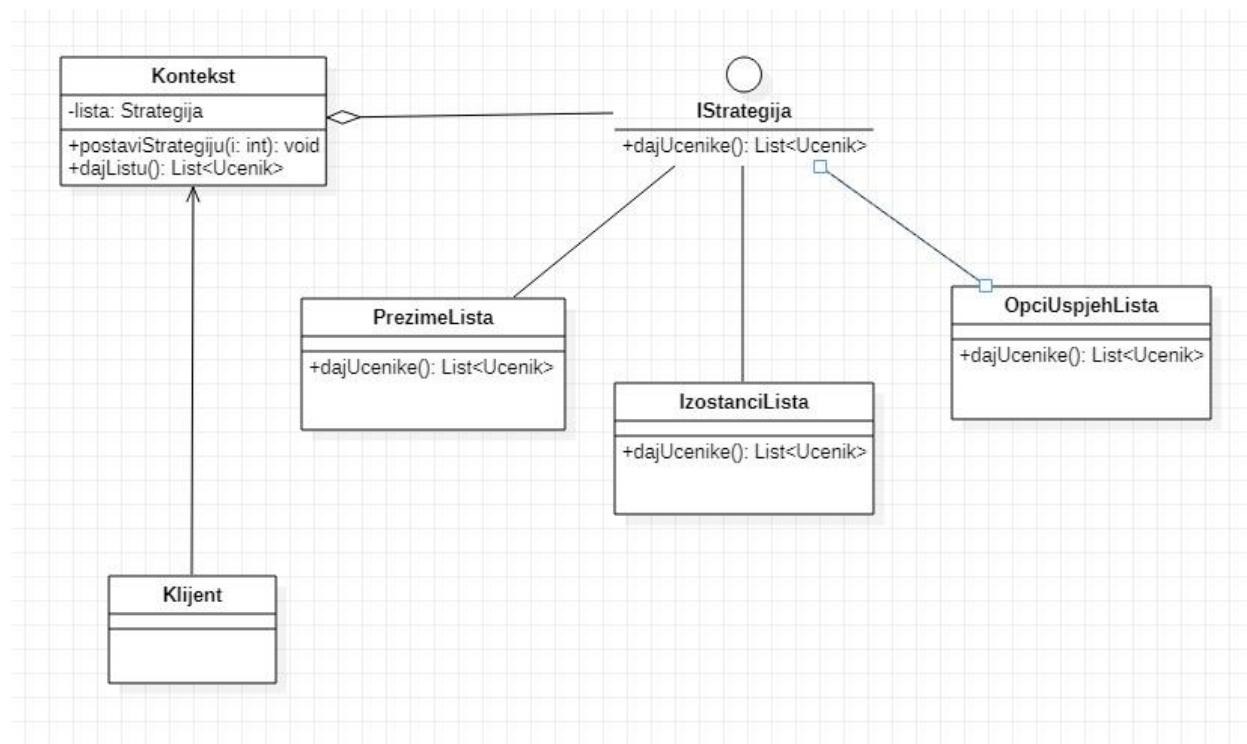
Uloga Builder paterna je odvajanje specifikacije kompleksnih objekata od njihove stvarne konstrukcije. U našem projektu, ovaj patern bi se mogao iskoristiti za pravljenje instanca klase Skola zato što se sastoji od lista kompleksnih objekata. Klasa Product bi zapravo bila Skola. Dodali bismo interfejs iBuilder, kao i Builder klasu koja bi implementirala metode napraviUcenika(), napraviNastavnika(), napraviRoditelja(), napraviAdministratora() u kojima bi se pravile instance klasa Ucenik, Nastavnik, Roditelj i Administrator i dodavale u odgovarajuće liste instance klase Skola koju Builder klasa ima kao svoj atribut. Dodali bismo metodu dajRezultat() koja bi vraćala produkt, odnosno formiranu školu.

Paterni ponašanja

Strategy pattern (implementiran)

Strategy patern izdvaja algoritam iz matične klase i uključuje ga u posebne klase.

Razrednik ima mogućnost pregleda liste učenika iz svog odjeljenja po različitom kriteriju. Dakle, on može zahtijevati listu učenika sortiranih abecedno po prezimenu, zatim listu učenika sortiranu po općem uspjehu na kraju polugodišta ili na kraju školske godine, zatim listu učenika sortiranu po ukupnom broju izostanaka. Primjenom ovog paterna omogućeno je da se u budućnosti doda još kriterija za sortiranje učenika jednog odjeljenja. Klasa *Kontekst* ima atribut *lista* tipa interfejs *IStrategija*. Njega implementiraju klase *ProsjekLista*, *PrezimeLista* i *IzostanciLista* koje preko metode *dajUcenike():List<Ucenik>* vraća listu učenika sortiranu po odgovarajućem kriteriju.



State pattern

State patern mijenja način ponašanja objekta na osnovu trenutnog stanja.

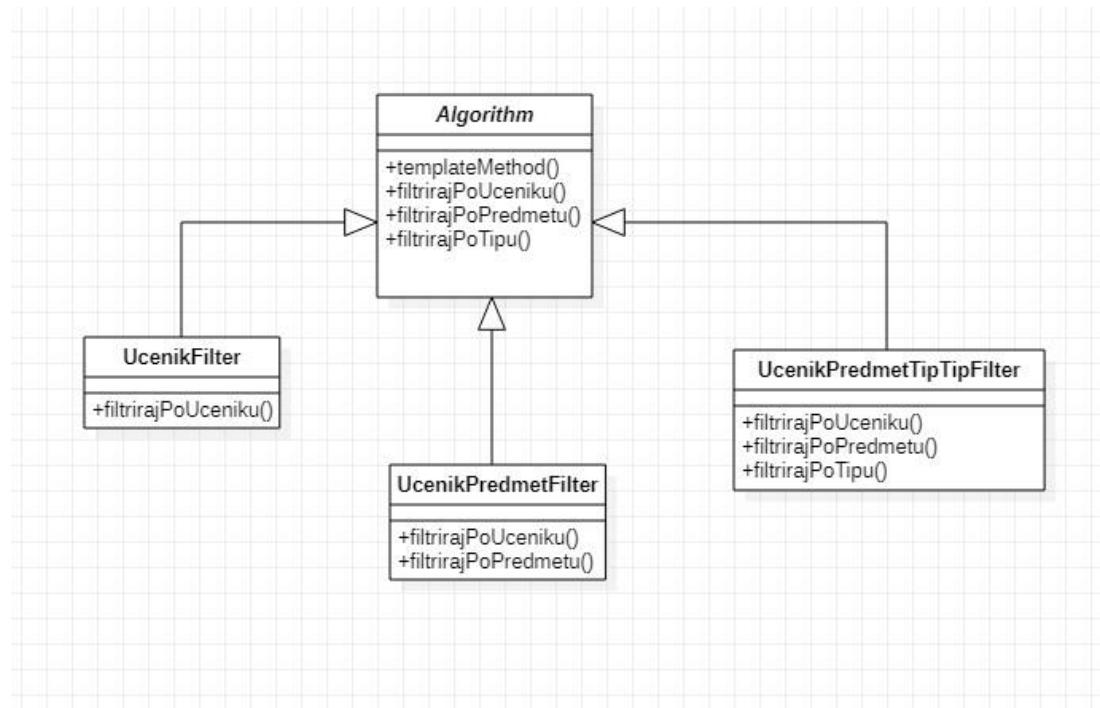
Ovaj patern nismo primijenili u našem sistemu. Mogli bismo ga primijenit na sljedeći način. Nastavnik može imati dva stanja u odnosu na jedno odjeljenje: samo nastavnik ili razrednik tom odjeljenju. Ukoliko je samo nastavnik, ne bi imao mogućnost pravdanja izostanaka, evidentiranja izvještaja sa roditeljskog sastanka, zaključivanja općeg uspjeha itd., dok bi kao razrednik to mogao. Vrijednost ovog atributa bi se mijenjala sa promjenom odjeljenja sa kojim radi. Da bismo primijenili ovaj patern, trebali bismo u klasu *Nastavnik* dodati atribut *stanje:Stanje* i

metodu *promjeniStanje()*. Zatim, bismo dodali interfejs *Stanje* sa metodama iz klase *Nastavnik* koje su vezane za upis ocjene, upis časa, pravdanje izostanaka itd. Taj interfejs bi implementirali klase *NastavnikOdjeljenju* i *RazrednikOdjeljenju*. Ukoliko bi se pozvala metoda *opravdajIzostanak*, a nastavnik je u stanju *NastavnikOdjeljenje* ne bi se desilo nista, a ukoliko je *RazrednikOdjeljenju* metoda bi se ispravno izvršila. Ovaj patern nismo primijenili, jer smo zamislili da naš sistem drugačije funkcioniše.

Template Method pattern (implementiran)

Template Method patern omogućava izdvajanje određenih koraka algoritma u odvojene podklase.

Primijenili smo Template Method patern kako bi omogućili izdvajanje određenih ocjena iz liste ocjena jednog odjeljenja. Recimo, želimo pregled svih ocjena nekog učenika, zatim možemo specificirati predmet iz kojeg želimo vidjeti ocjene, zatim odabrati tip ocjene (PISMENA ZADACA, TEST, AKTIVNOST itd.). Kako bi primijenili ovaj patern u naš model dodali smo klase *Algorithm*, *UcenikFilter*, *UcenikPredmetFilter* i *UcenikPredmetTipFilter*. Klasa *Algorithm* je apstraktna klasa koja ima metode *templateMethod()*, *filtrirajPoUceniku()*, *filtrirajPoPredmetu()* i *filtrirajPoTipu()*. Dakle, ukoliko želimo ocjene samo jednog učenika primijenit će se klasa *UcenikFilter*, ukoliko odredimo i predmet, primijenit će se klasa *UcenikPredmetFilter* i ukoliko navedemo sva tri parametra primijenit će se klasa *UcenikPredmetTip*.



Observer pattern

Uloga Observer paterna je da uspostavi relaciju između objekata tako da kada jedan objekat promijeni stanje drugi zainteresirani objekti se obavještavaju.

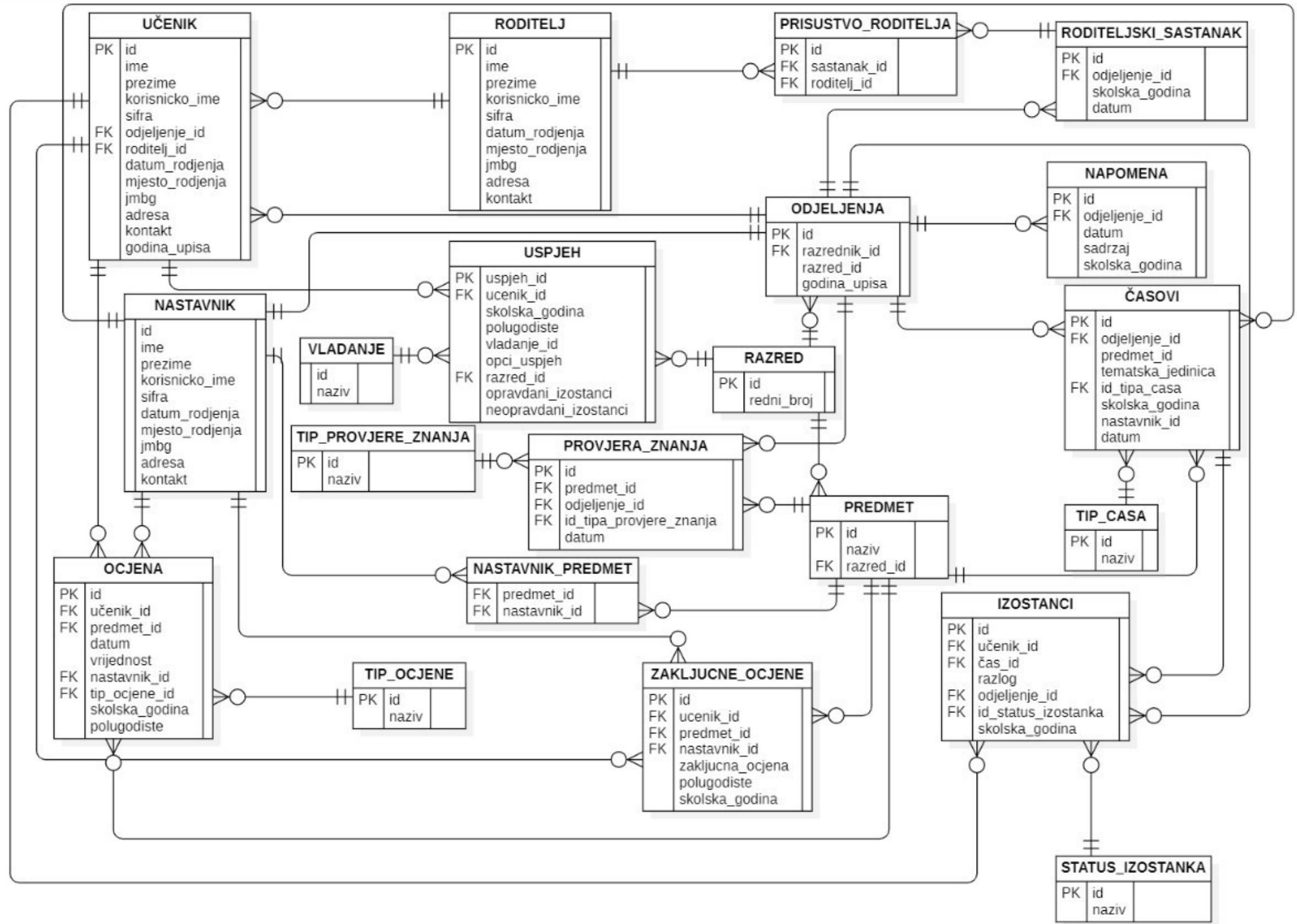
Ovaj patern nismo primijenili u našem sistemu, ali bismo ga mogli primijeniti ukoliko želimo da se roditelj i razrednik obavijeste ukoliko izostanak učenika nije opravadan u periodu od 7 dana od dana kada se desio. U tom slučaju bi trebali definisati interfejs *IPratilac* koji bi imao metodu *update()*. Klase *Razrednik* i *Roditelj* bi implementirali interfejs *IPratilac*. U klasi *Izostanak* pored navedenih atributa bi imali atribut *pratioci:list<Osoba>* u kojem bi čuvali listu pratioca (u našem slučaju to bi bili samo razrednik i roditelj, pa bi ih mogli čuvati odvojeno, ali kako bi ostavili mogućnost za dodavanje još pratioca koristili bi listu), zatim metode *obavijestiPratioce()* koja je privatna metoda, *dodajPratioca(osoba:Osoba)* i *izbrisipratioca(osoba:Osoba)* (u slučaju da se promijeni razrednik) i metodu *provjeriStanje()* koja bi provjerila da li je prošao zadani period i u slučaju da jeste poziva se metoda *obavijestiPratioce()*.

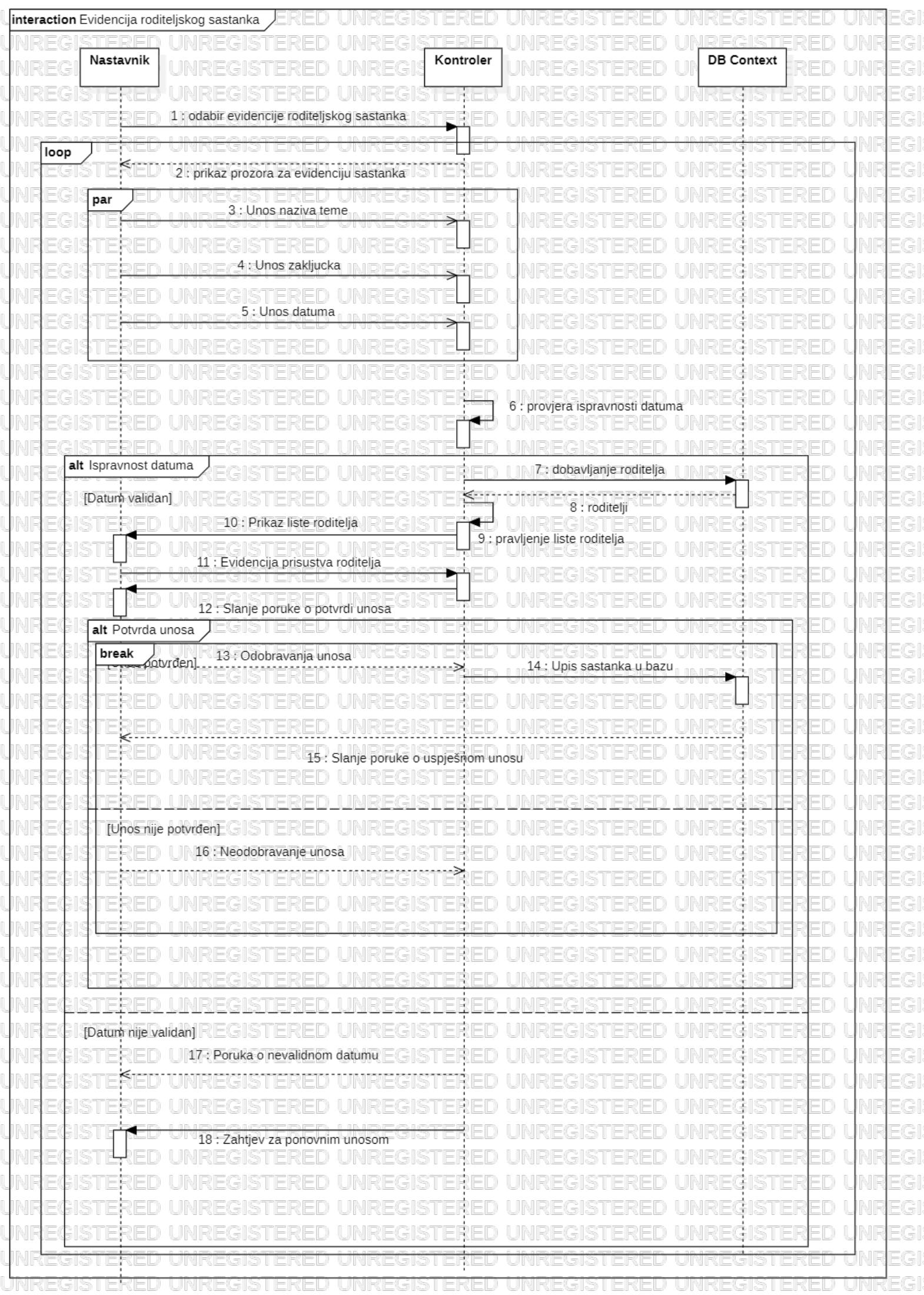
Iterator pattern

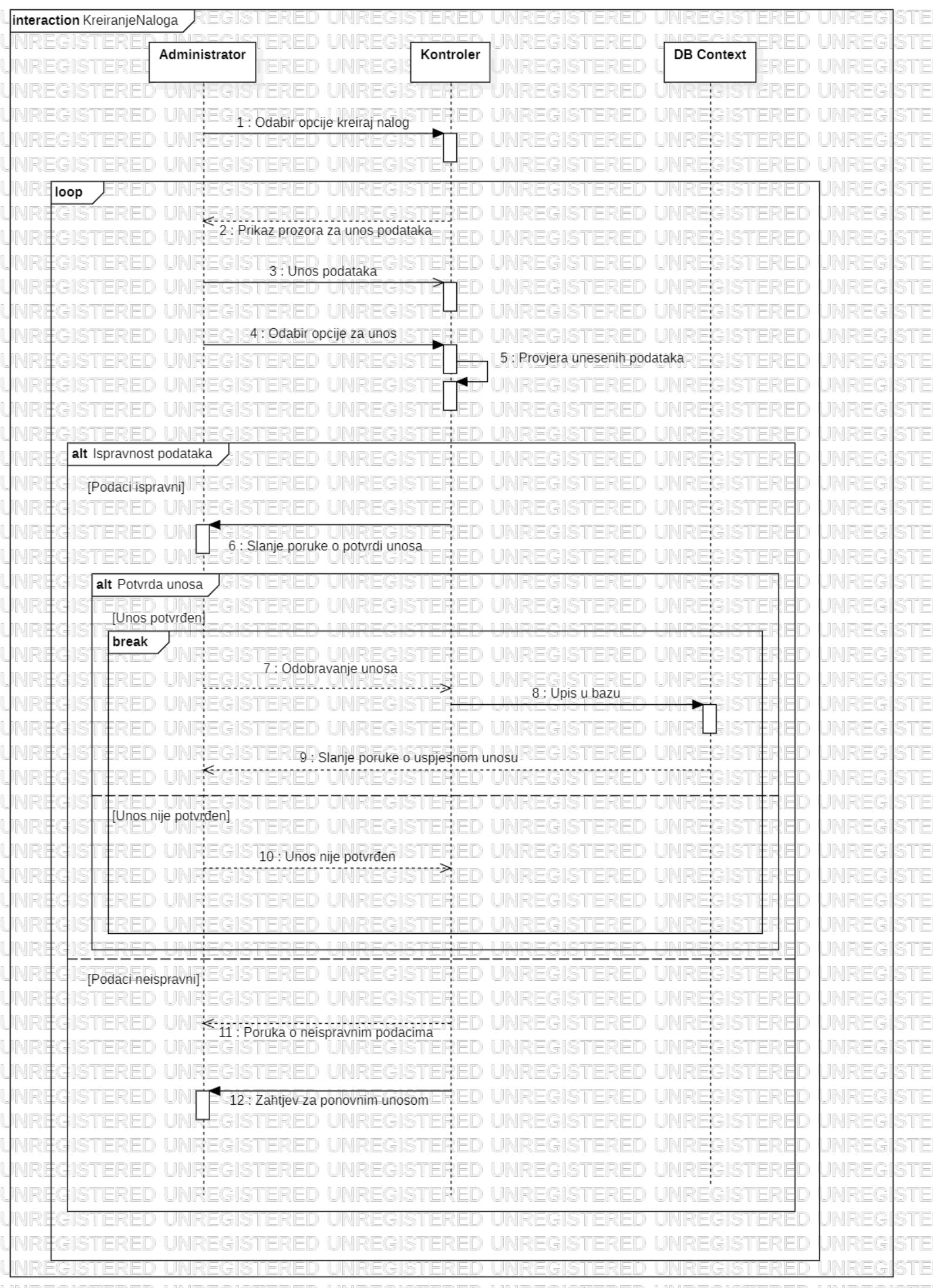
Iterator patern omogućava sekvensijalni pristup elementima kolekcije bez poznavanja kako je kolekcija struktuirana.

Ovaj patern nismo primijenili u našem sistemu.

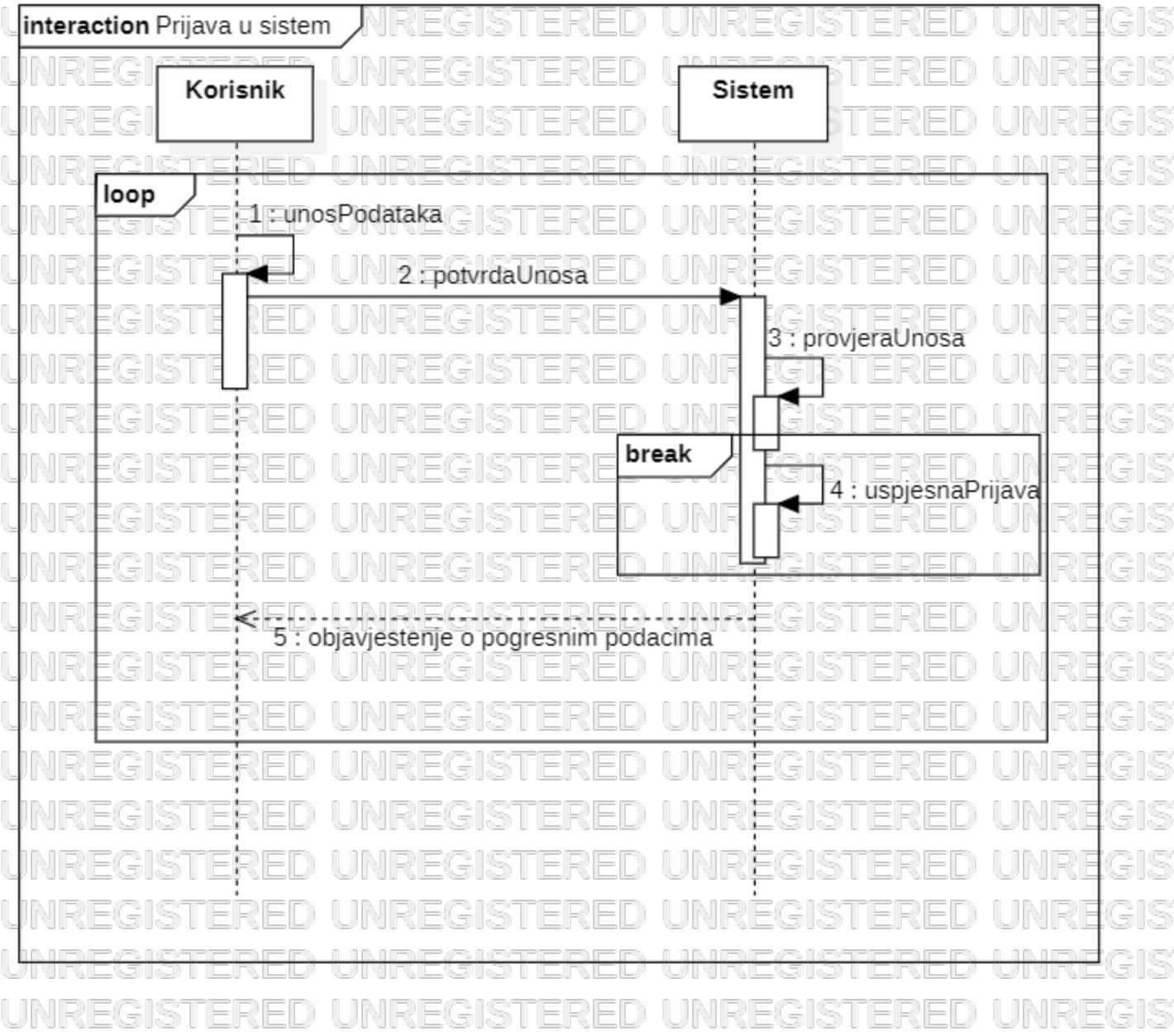
U klasi *Odjeljenje* čuvamo listu učenika tog odjeljenja. Primjenom ovog paterna bismo omogućili iteriranje foreach petljom kroz listu učenika abecedno po prezimenu. Dodali bismo novu klasu *UcenikIterator* i dva nova interfejsa *Iterator* i *Aggregate*. Interfejs *Aggregate* bi imao metodu *createIterator()*, a klasa *Odjeljenje* bi implementirala taj interfejs. Klasa *UcenikIterator* bi implementirala interfejs *Iterator* koji bi imao metode *hasNext()* i *next()*.



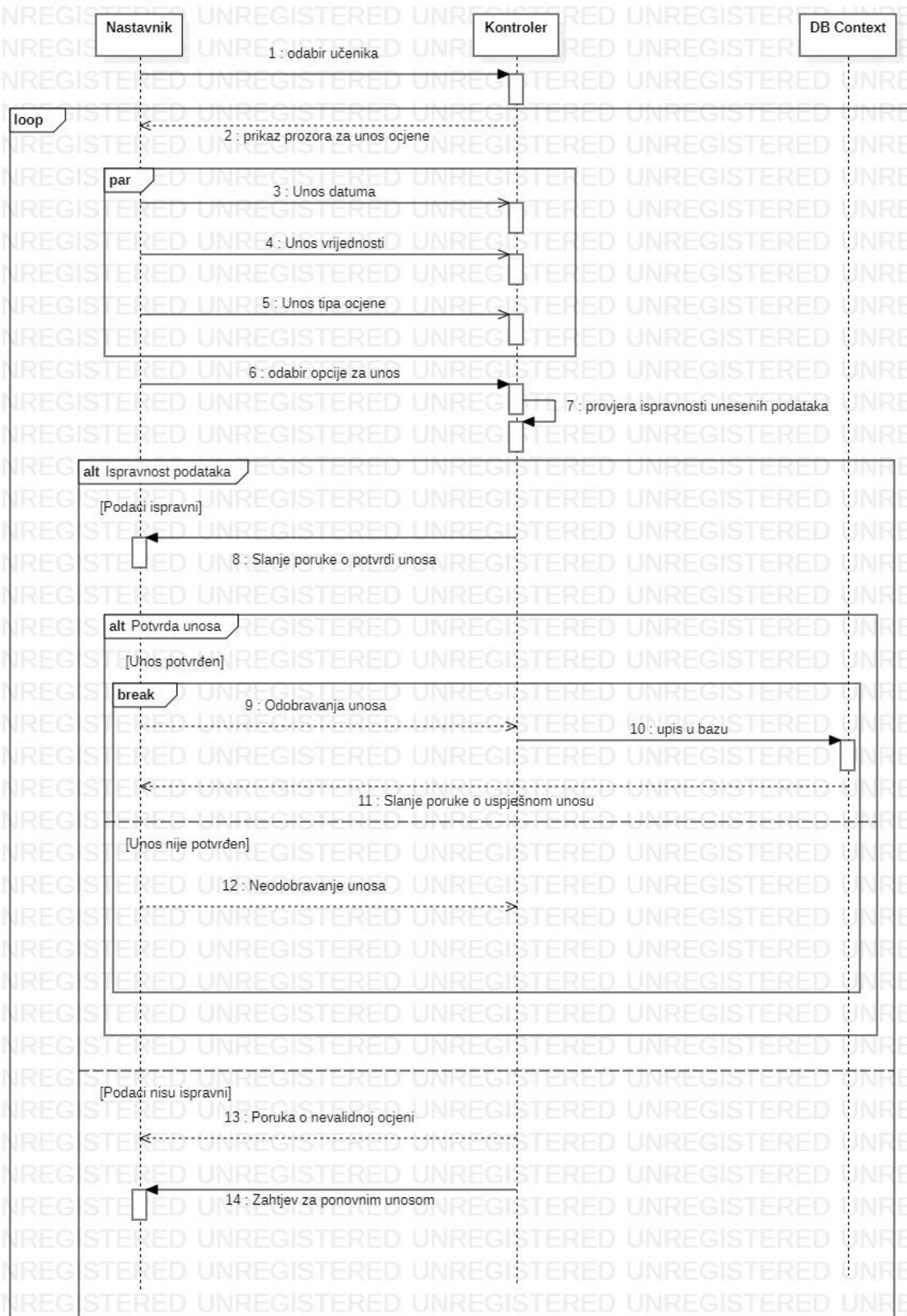


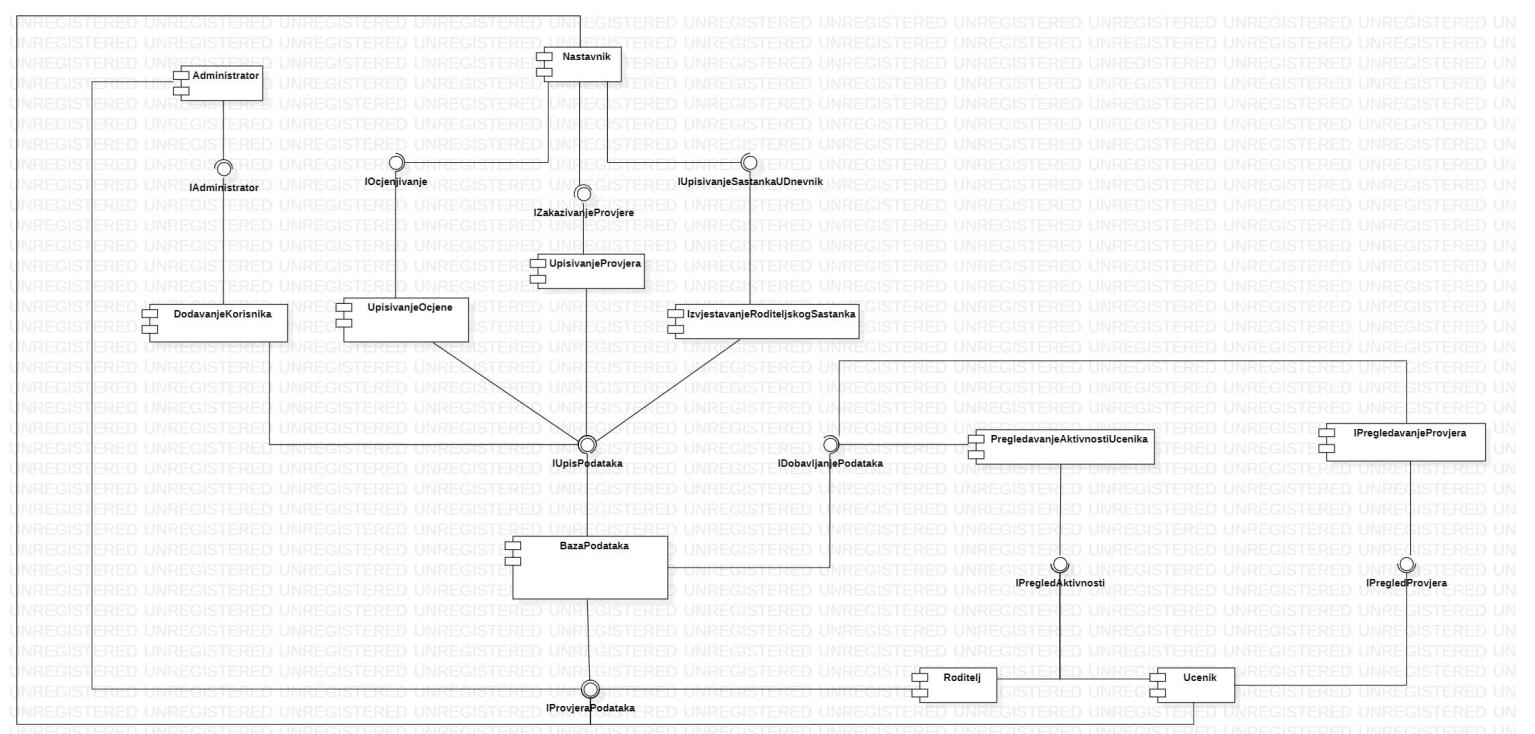


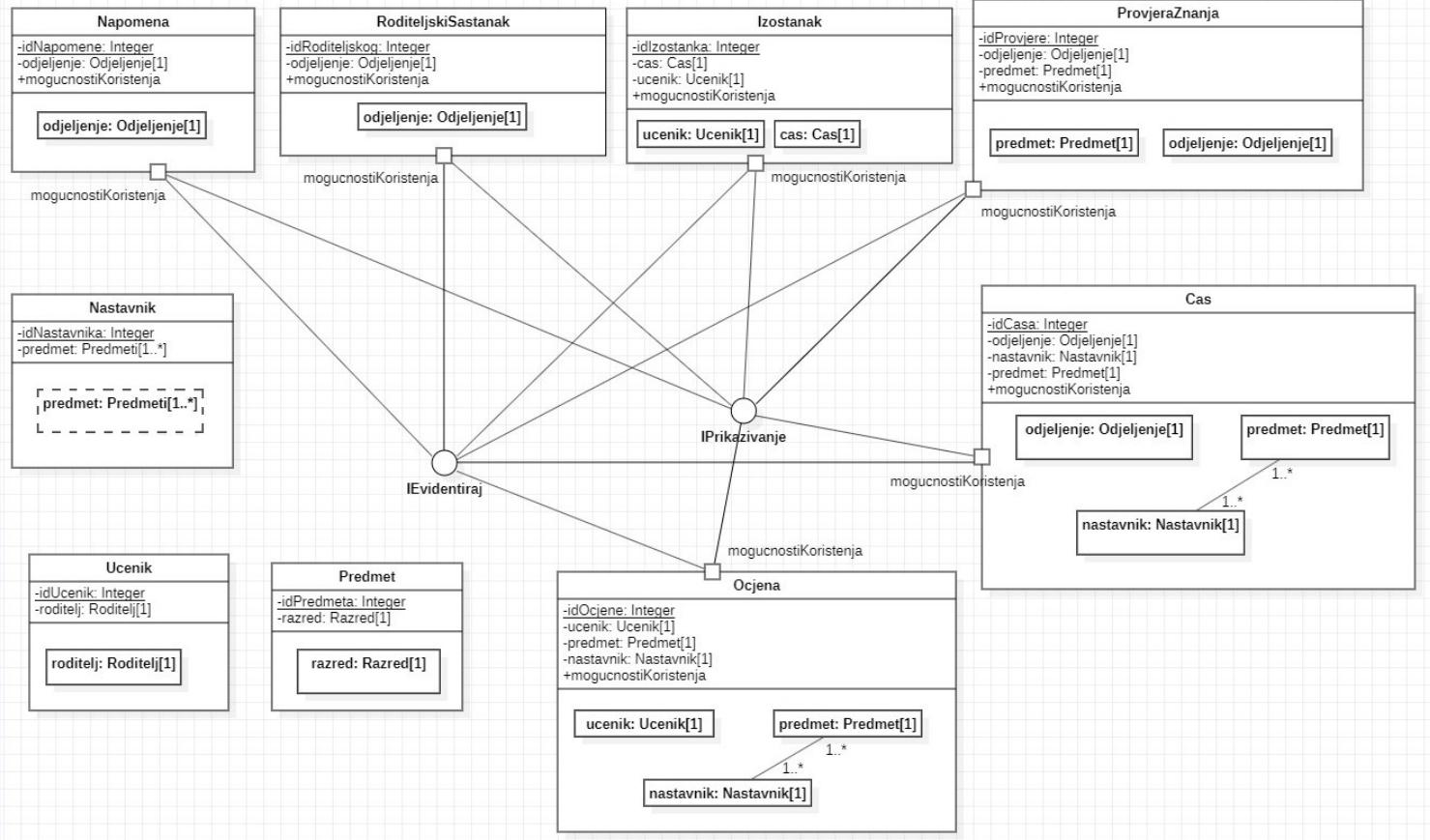
interaction Prijava u sistem



interaction UpisOcjene







Web servis

Funkcionalnost koja se može iskoristiti u našoj aplikaciji kao web servis je preuzimanje informacija o državnim praznicima i neradnim danima. To će omogućiti lakše kreiranje rasporeda i planiranje časova. Ove informacije se mogu preuzeti sa stranice <https://mon.ks.gov.ba/>.