

Cilj projekta jest razvoj programske podrške za stvaranje web aplikacije "EduHub" koja će obrazovnim ustanovama omogućiti organiziranje nastave elektroničkim sredstvima komunikacije. Aplikacija će olakšati pristup nastavnim materijalima, raznim resursima škole i njihovom zauzeću te omogućiti jednostavnu komunikaciju među sudionicima u srednjoškolskom obrazovanju. Kao poseban dodatak aplikacija na početnoj stranici pokazuje podatak o vremenskim uvjetima i vremenskoj prognozi za grad u kojoj se škola nalazi.

Prilikom pokretanja aplikacije prikazuje se zahtjev za prijavu u sustav. Za proces prijave koristimo vanjski servis za autentifikaciju - OAuth 2.0. Korisnik upisuje svoju email adresu i lozinku, nakon čega sustav prepoznaje/dodjeljuje ulogu prijavljene osobe (upisan učenik, ne upisan učenik, nastavnik, ravnatelj, satničar, djelatnik službe ili administrator). Djelatnicima škole i upisanim učenicima odmah se otvara njihova početna stranica (ovisno o ulozi), dok se neupisanim učenicima otvara stranica za upis gdje upisuju potrebne podatke.

Registrirani korisnici imaju mogućnosti

- prijave
- odjave
- pregleda rasporeda sati
- pregleda dostupnih resursa
- pregleda sadržaja kolegija i aktivnosti
- komunikacije s ostalim korisnicima

Učenicima će na jednom mjestu biti raspored sati, pregled svih aktivnosti te svi potrebni nastavni materijali. Učenici će biti obaviješteni o svakoj promjeni rasporeda putem elektroničke pošte. U slučaju terenske nastave i upućivanja učenika na stručnu praksu, prilikom upućivanja se u obavijesti koja se šalje učeniku šalje i točan podatak o lokaciji i načinu dolaska od lokacije škole do željene lokacije.

Za upis u prvi razred potrebni su sljedeći podaci:

- ime
- prezime
- datum rođenja
- OIB
- email adresa
- lozinka

Nakon upisa podataka korisnik odabire smjer koji želi upisati te mu se pridjeljuju odgovarajući predmeti i prikazuje sučelje za izbor izbornih predmeta.

*Učenik* je korisnik sa svim ovlastima registriranog korisnika s dodatnim mogućnostima pregleda nastavnih materijala koji su mu namijenjeni i podnošenjem zahtjeva za potvrdom o upisu.

Upis u prvi razred moguć je samo tijekom određenog razdoblja koji je određen početkom školske godine. Nakon završetka roka upisa u srednje škole izrađuje se raspored sati po smjerovima, vodeći pri tome brigu o najvećem mogućem kapacitetu prostorija, kao i o opremi koja je potrebna za izvođenje praktične nastave. Raspored se prikazuje u preglednom tabličnom obliku za svaki tjedan unutar akademske godine. Pri tome se vodi računa o svim državnim praznicima, blagdanima i neradnim danima. Škola radi isključivo ponedjeljak – petak, u jednoj smjeni.

Svakom djelatniku škole se registracijom dodjeljuju prava "registriranog korisnika". Naknadno im administrator informatičkog sustava dodjeljuje jednu od idućih uloga: nastavnik, ravnatelj, satničar, djelatnik studentske službe.

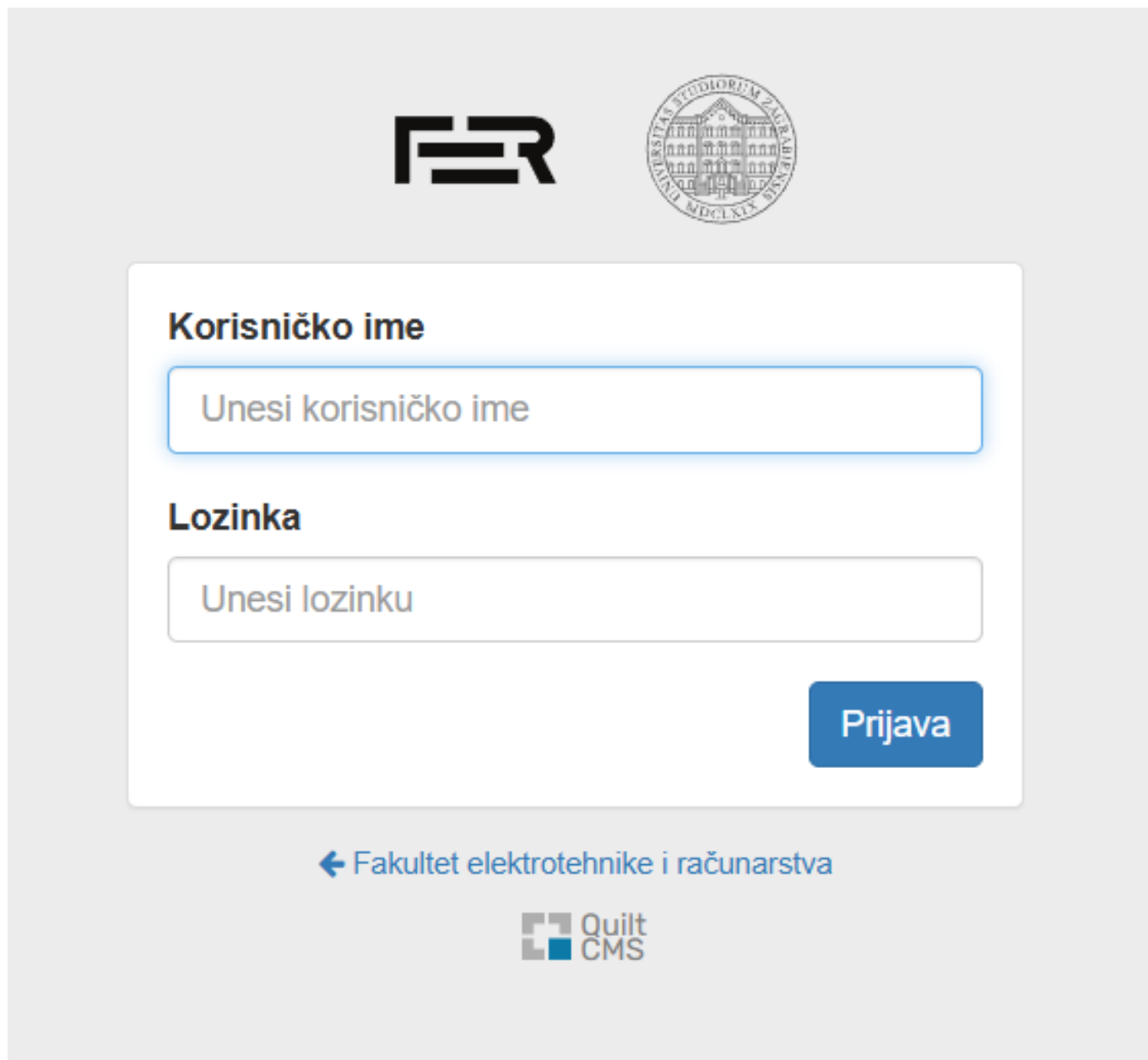
*Nastavnik* je korisnik sa svim ovlastima registriranog korisnika s dodatnom mogućnosti dodavanja nastavnog materijala. Nastavni materijali bit će dostupni samo učenicima kojima su namijenjeni. Pratit će se statistika pregleda i pristupa svim nastavnim materijalima.

*Ravnatelj i djelatnici službe* su korisnici sa svim ovlastima registriranog korisnika s dodatnom mogućnosti pregleda izvještaja o izdavanju potvrda o upisu u školu.

Uz navedene korisnike postoji još jedna vrsta korisnika.

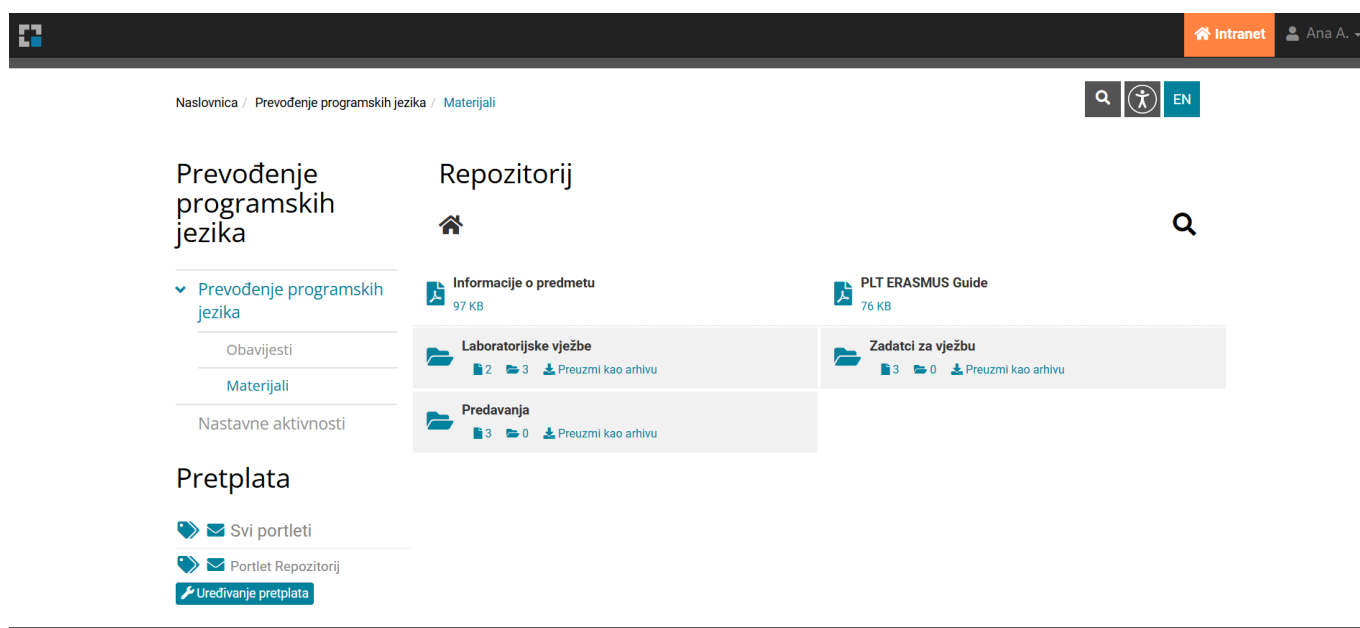
*Administrator informatičkog sustava* je korisnik s najvećim ovlastima. On ima pristup bazi s popisom registriranih korisnika i njihovim podacima. Obavlja dodjelu uloga djelatnicima škole. Određuje vrijeme u kojem je upis u prvi razred moguć. Dostupni su mu svi podaci te može definirati sve potrebne uvjete za ispravan rad sustava.

Aplikacija je izrađena za specifičnu uporabu obrazovnih ustanova koju ispunjavaju i slične aplikacije poput Google classroom, FERWeb i Moodle. Navedene aplikacije imaju pristup materijalima za određen predmet i kalendar s rasporedom. Google classroom i Moodle imaju opciju komunikacije među korisnicima, FERWeb šalje obavijesti elektroničkom poštom.



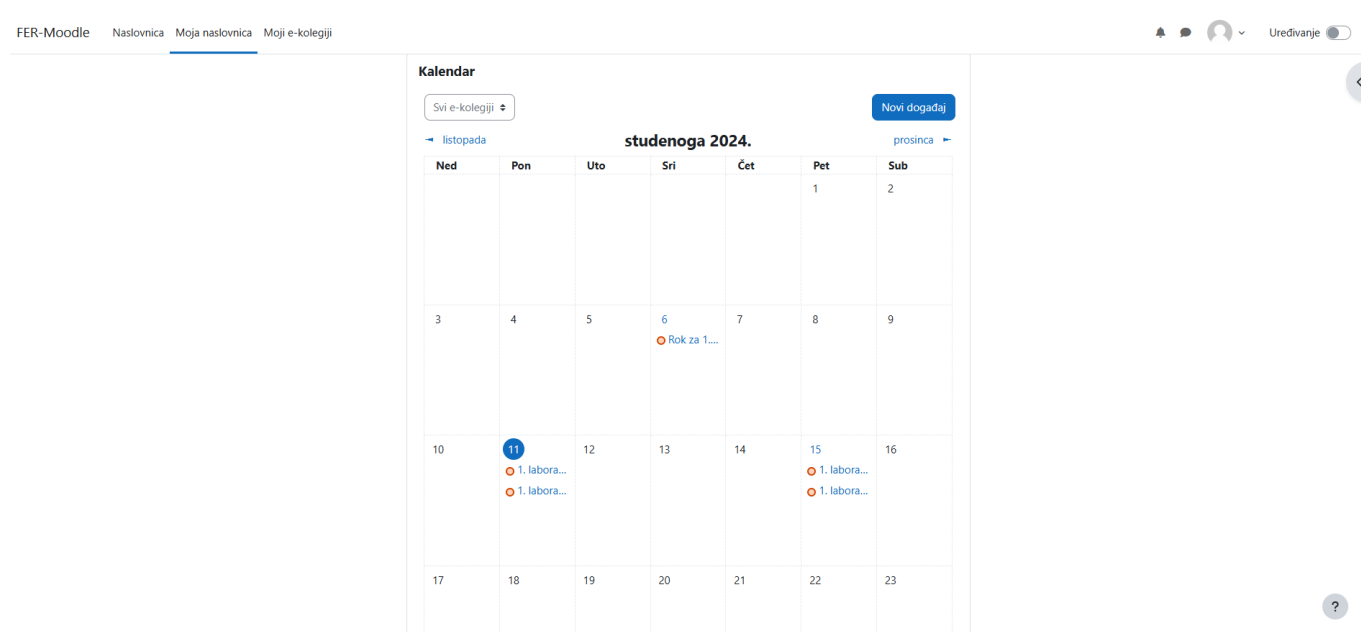
The image shows the FERWeb login interface. At the top, there are two logos: the FER logo on the left and the University of Zagreb seal on the right. Below the logos, there is a white login box with a blue border. Inside the box, the text "Korisničko ime" (Username) is followed by a text input field containing the placeholder "Unesi korisničko ime". Below this, the text "Lozinka" (Password) is followed by another text input field containing the placeholder "Unesi lozinku". To the right of the password field is a blue button labeled "Prijava" (Login). Below the login box, there is a blue arrow pointing left followed by the text "Fakultet elektrotehnike i računarstva". At the bottom center, there is a logo for "Quilt CMS".

Slika 1.1: Primjer FERWeb prijave u sustav

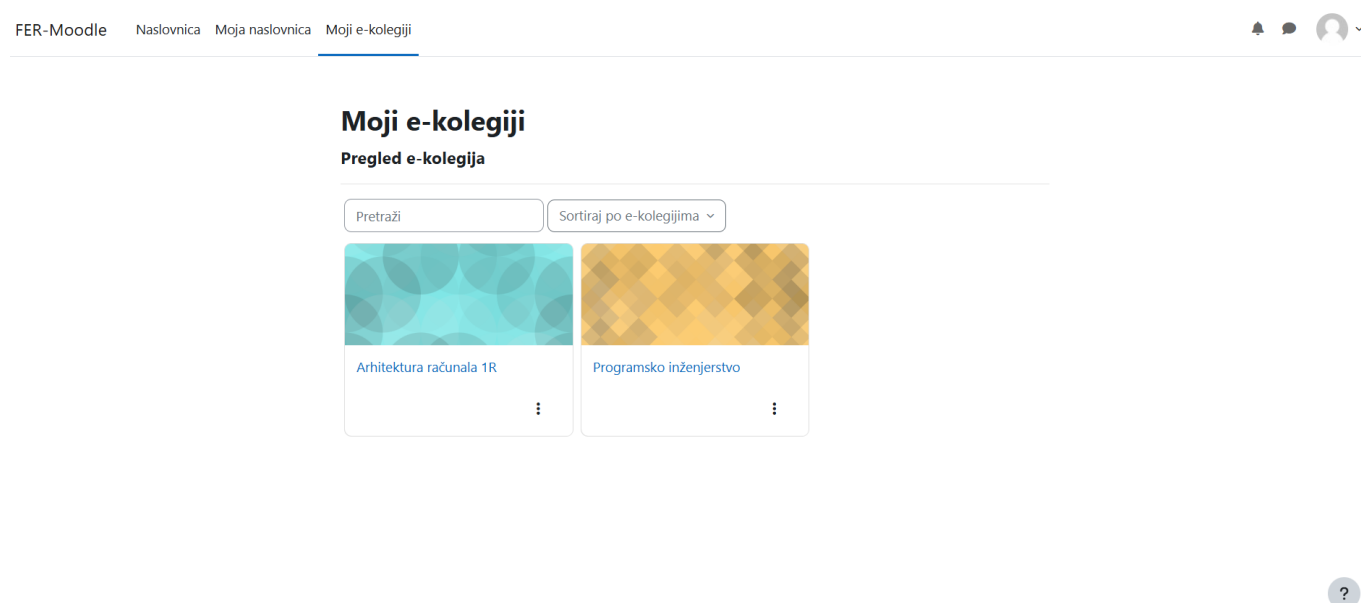


The image shows the FERWeb dashboard interface. At the top, there is a dark header bar with the FER logo on the left, the text "Intranet" in the center, and a user profile icon with the name "Ana A." on the right. Below the header bar, there is a navigation bar with the text "Naslovnica / Prevođenje programskih jezika / Materijali" on the left and search, user, and language icons on the right. The main content area is divided into several sections. On the left, there is a section titled "Prevođenje programskih jezika" with a dropdown menu showing "Prevođenje programskih jezika" (selected), "Obavijesti", "Materijali", and "Nastavne aktivnosti". Below this is a section titled "Pretplata" with a list of items: "Svi portleti", "Portlet Repozitorij", and "Uređivanje pretplata". In the center, there is a section titled "Repozitorij" with a home icon and a search icon. Below this, there are two columns of content. The left column has "Informacije o predmetu" (97 KB), "Laboratorijske vježbe" (2 files, 3 downloads), and "Predavanja" (3 files, 0 downloads). The right column has "PLT ERASMUS Guide" (76 KB) and "Zadaci za vježbu" (3 files, 0 downloads). At the bottom, there is a dark footer bar.

Slika 1.2: Primjer FERWeb izgleda kolegija



Slika 1.3: Primjer moodle kalendara



Slika 1.4: Primjer moodle izbornika kolegija

Neke od mogućih nadogradnja aplikacije su

- Implementiranje dvofaktorske autentikacije
- Automatsko stvaranje @skole.hr mailova za učenike nakon njihovih upisa
- Razvoj mobilne aplikacije za Android i iOS
- Omogućavanje prijave za stručnu praksu kroz aplikaciju

## Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-001	Sustav omogućuje upis u prvi razred.	Visok	Zahtjev dionika	Korisnik se s pomoću elektroničke pošte prijavi u sustav kao učenik. Ako ga sustav ne prepozna kao upisanog učenika prebacuje ga na stranicu za upis u školu gdje se upisuje korištenjem osobnih podataka (ime, prezime, datum rođenja, OIB) i odabirom željenog smjera.
F-002	Sustav omogućuje prijavu korisnika.	Visok	Zahtjev dionika	Postojeći korisnik se prijavljuje točnom email adresom i lozinkom.
F-003	Sustav omogućuje odjavu korisnika.	Srednji	Zahtjev dionika	Korisnik se odjavljuje iz aplikacije i ne može više pristupiti podatcima.
F-004	Sustav omogućuje pregled rasporeda sati.	Visok	Postojeći sustav	Korisniku je omogućen pregled rasporeda sati po danima.
F-005	Sustav omogućuje pregled dostupnih resursa.	Visok	Postojeći sustav	Korisniku je omogućen pregled dostupnih resursa škole. Podatci koje korisnik vidi ovise o njegovoj ulozi u školi.
F-006	Sustav omogućuje pregled sadržaja kolegija i aktivnosti.	Visok	Postojeći sustav	Korisniku je omogućen pregled sadržaja kolegija svakog upisanog kolegija.
F-007	Sustav omogućuje komunikaciju s ostalim korisnicima.	Visok	Zahtjev dionika	Korisnicima je omogućena komunikacija s bilo kojim drugim registriranim korisnikom.
F-008	Sustav nastavnicima omogućuje dodavanje nastavnih materijala.	Visok	Zahtjev dionika	Korisniku (nastavniku) je omogućeno dodavanje nastavnih materijala za predmete koje predaju.
F-009	Sustav ravnatelju i djelatnicima službe omogućuje pregled izvještaja.	Visok	Zahtjev dionika	Korisniku (ravnatelju, djelatniku službe) je omogućen pregled izvještaja.
F-010	Sustav pohranjuje podatke o korisnicima.	Visok	Dokument zahtjeva	Podatci o korisnicima se spremaju u bazu podataka,

ID zahtjeva	Opis	Prioritet	Izvor	Kriteriji prihvatanja
F-011	Sustav pohranjuje podatke o nastavnim materijalima, njihovim pregledima i pristupima.	Visok	Dokument zahtjeva	Podatci o nastavnim materijalima, njihovim pregledima i pristupima se pohranjuju u bazu podataka.
F-012	Sustav poslužuje potrebne podatke vezane za prognozu.	Visok	Postojeći sustav	Korisniku je omogućen pregled vremenske prognoze.
F-013	Sustav omogućuje slanje elektroničke pošte korisnicima.	Visok	Povratne informacije korisnika	Korisnik dobiva elektroničku poštu o promjenama na stranici.
F-014	Sustav omogućuje prikaz navigacije korisnicima.	Visok	Povratne informacije korisnika	Korisnik ima pristup navigaciji od škole do lokacije terenske nastave.

## Ostali zahtjevi

ID zahtjeva	Opis	Prioritet
NF-001	Optimalan raspored sati za nastavnike i učenike. Nema previše nastavnih sati u jednom danu.	Visok
NF-002	Vođenje računa o praznicima. Praznici su nenastavni dani te tim danima kalendar treba biti prazan.	Visok
NF-003	Responzivni dizajn prilagođen upotrebi na različitim uređajima.	Visok
NF-004	Podržavanje višekorisničkog rada bez smanjenja performansi.	Visok
NF-005	Enkripcija osjetljivih podataka za sigurnost svih korisnika.	Visok
NF-006	Pregledno te intuitivno sučelje.	Visok
NF-007	Modularan programski kod.	Visok
NF-008	Visoka pouzdanost aplikacije 99 % <i>uptime</i> .	Visok

## Dionici

1. Obrazovna ustanova (naručitelj)
2. Administrator
3. Razvojni tim
4. Korisnici
5. Učenici
6. Djelatnici škole

### **Aktori i njihovi funkcionalni zahtjevi:**

1. Registrirani korisnik (inicijator) može:
  - prijaviti se
  - odjaviti se
  - pregledati raspored sati
  - pregledati dostupne resurse
  - pregledati sadržaj kolegija i aktivnosti
  - komunicirati s ostalim korisnicima
2. Učenik
  - pregled nastavnih materijala
  - zahtijevanje potvrde o upisu
3. Nastavnik (inicijator) može:
  - dodati nastavne materijale
4. Djelatnici studentske službe i ravnatelj (inicijatori) mogu:
  - pregledati izvještaje
5. Baza podataka (sudionik):
  - pohranjuje podatke o korisnicima
  - pohranjuje podatke o nastavnim materijalima, njihovim pregledima i pristupima
6. Poslužitelj vremenske prognoze (sudionik):
  - poslužuje potrebne podatke vezane za prognozu
7. Poslužitelj elektroničke pošte (sudionik):
  - aplikaciji omogućuje slanje elektroničke pošte korisnicima
8. Poslužitelj navigacije (sudionik):
  - aplikaciji omogućuje prikaz navigacije korisnicima

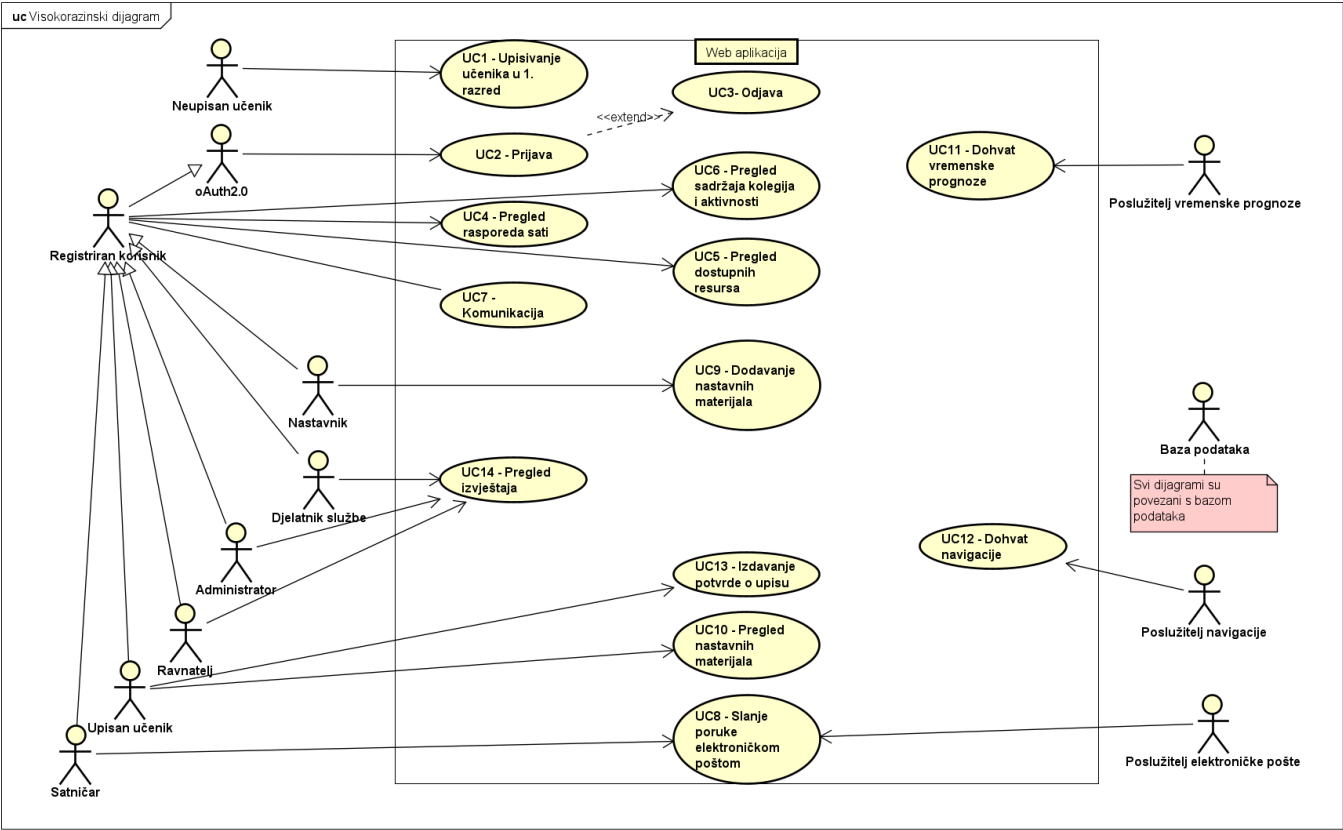
## Obrasci uporabe

---

## Dijagrami obrazaca uporabe

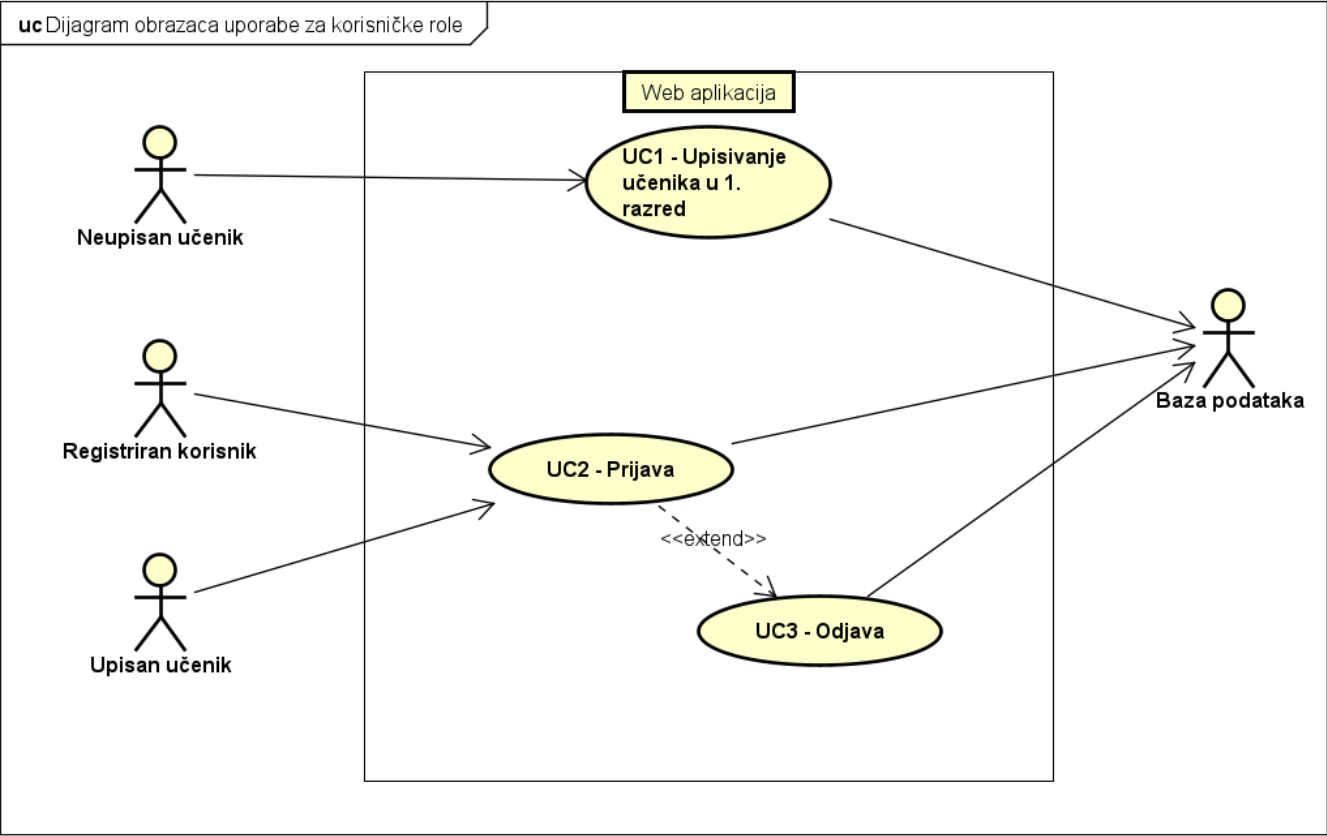
---

### 1. Visokorazinski dijagram obrazaca uporabe cijelog sustava



Slika 3.1: Visokorazinski dijagram obrazaca uporabe cijelog sustava

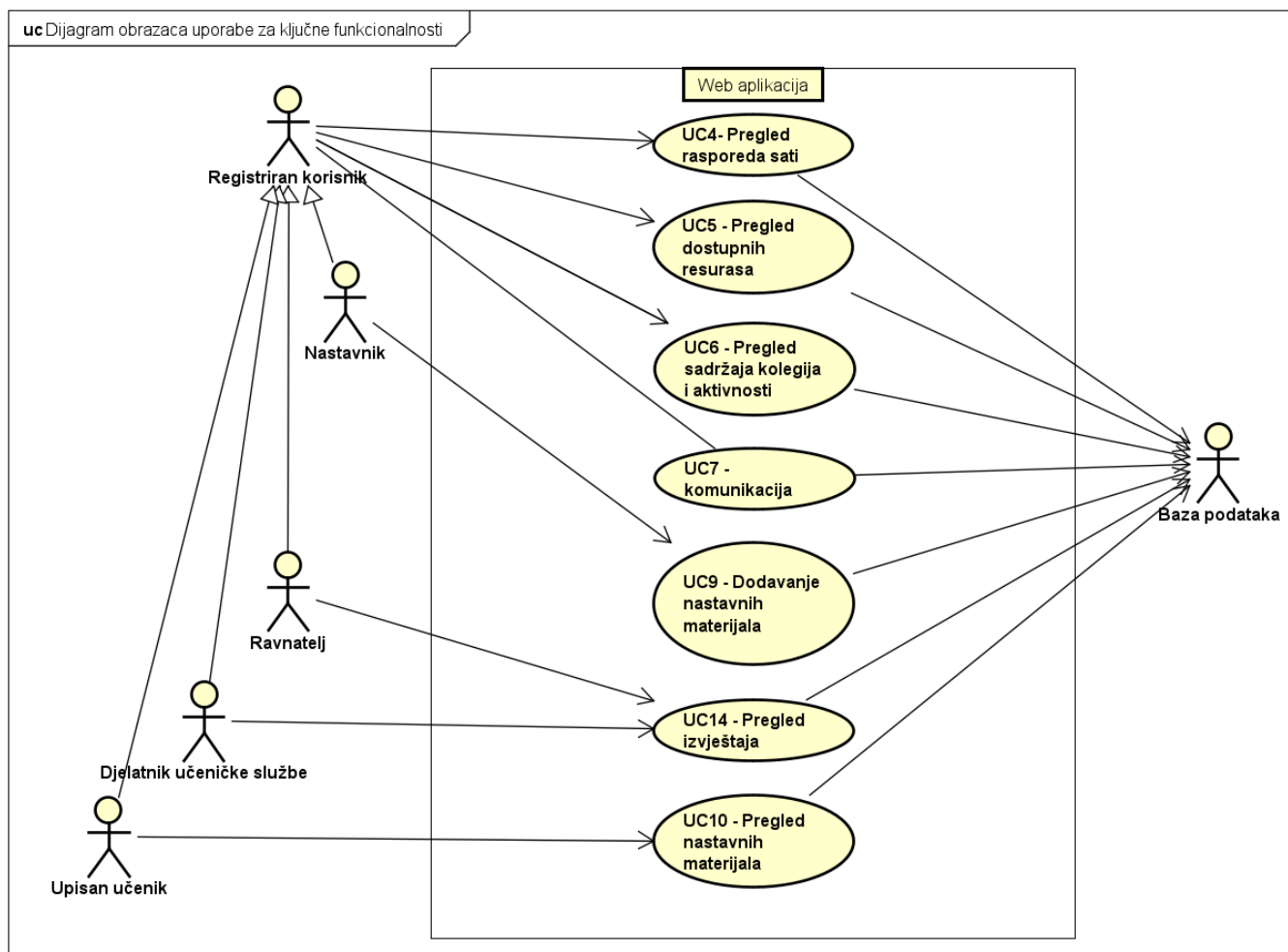
2. Dijagram obrazaca uporabe za ključne značajke



Slika 3.2: Dijagram obrazaca uporabe za ključne značajke



### 3. Dijagram obrazaca uporabe za korisničke uloge



Slika 3.3: Dijagram obrazaca uporabe za korisničke uloge

## Opis obrazaca uporabe

### UC1 - Upisivanje učenika u 1. razred

- **Glavni sudionik:** Neupisan učenik
- **Cilj:** Upisati prvi razred određenog smjera
- **Sudionici:** Baza podataka
- **Preduvjet:** Nema preduvjeta
- **Opis osnovnog tijeka:**
  1. Korisnik se prijavljuje u sustav
  2. Korisnik upisuje osobne podatke
  3. Korisnik odabire smjer koji želi upisati
  4. Korisniku se pridjeljuju odgovarajući predmeti
  5. Korisnik odabire izborne predmete
  6. Otvara se pregled za odabran smjer

### UC2 - Prijava

- **Glavni sudionik:** učenici, nastavnici, djelatnici škole
- **Cilj:** Pristup svim funkcionalnostima namijenjenih korisnikovoj ulozi

- **Sudionici:** Baza podataka
- **Preduvjet:** Nema preduvjeta
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Prijava" te upisuje podatke
  2. Provjera postoji li korisnik u bazi podataka
  3. Korisnik je prijavljen ako postoji
- **Opis mogućih odstupanja:**
  - 2.a Korisnik ne postoji u bazi podataka
    - Sustav obavještava korisnika o neispravnosti podataka

### UC3 - Odjava

- **Glavni sudionik:** Registriran korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju "Odjava"
  2. Korisnik potvrđuje odjavu
  3. Povratak na početni zaslon
- **Opis mogućih odstupanja**
  - 3.a Korisnik odustaje od odjave
    - Povratak na prethodni zaslon

### UC4 - Pregled rasporeda sati

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled rasporeda sati
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju za pregled rasporeda sati
  2. Otvara se prozor s rasporedom sati

### UC5 - Pregled dostupnih resursa

- **Glavni sudionik:** Djelatnik škole
- **Cilj:** Pregled zauzeća postojećih resursa škole (ljudski i materijalni - oprema i prostor)
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju pregleda dostupnih resursa
  2. Otvara se prozor s resursima

### UC6 - Pregled sadržaja kolegija i aktivnosti

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Pregled informacija o obrazovnom sadržaju svih kolegija, nastavnih i nenastavnih aktivnosti u školi

- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju pregleda sadržaja
  2. Otvara se prozor sa sadržajem

## UC7 - Komunikacija

- **Glavni sudionik:** Registriran korisnik
- **Cilj:** Mogućnost direktne komunikacije unutar svake grupe i razreda kao i privatnih poruka između pojedinih osoba
- **Sudionici:** Baza podataka
- **Preduvjet:** Korisnik je prijavljen
- **Opis osnovnog tijeka:**
  1. Korisnik odabire opciju slanja poruke

## UC8 - Slanje poruke elektroničkom poštom

- **Glavni sudionik:** Poslužitelj elektroničke pošte
- **Cilj:** Poslati poruku elektroničkom poštom
- **Sudionici:** Baza podataka
- **Preduvjet:** Generiran ili izmijenjen raspored
- **Opis osnovnog tijeka:**
  1. Slanje poruke elektroničkom poštom nakon generiranja ili izmjene rasporeda

## UC9 - Dodavanje nastavnih materijala

- **Glavni sudionik:** Nastavnik
- **Cilj:** Objava nastavnih materijala
- **Sudionici:** Baza podataka, AWS Cloud
- **Preduvjet:** Prijavljen je nastavnik
- **Opis osnovnog tijeka:**
  1. Nastavnik odabire opciju za dodavanje nastavnih materijala
  2. Nastavnik unosi potrebne podatke
- **Opis mogućih odstupanja:**
  - 9.a Nastavnik odustaje od dodavanja nastavnih materijala

## UC10 - Pregled nastavnih materijala

- **Glavni sudionik:** Upisan učenik
- **Cilj:** Pregled nastavnih materijala koji su namijenjeni pojedinom učeniku
- **Sudionici:** Baza podataka
- **Preduvjet:** Učenik je upisan, prijavljen te je nastavni materijal namijenjen njemu
- **Opis osnovnog tijeka:**
  1. Učenik odabire predmet iz kojeg želi pregledati nastavni materijal
  2. Učenik odabire materijal koji mu je potreban

## UC11 - Dohvat vremenske prognoze

- **Glavni sudionik:** Poslužitelj vremenske prognoze
- **Cilj:** Dohvatiti podatke o vremenskoj prognozi
- **Sudionici:** -
- **Preduvjet:** Škola i grad u kojoj se nalazi evidentirani su u sustavu
- **Opis osnovnog tijeka:**
  1. Aplikacija šalje zahtjev za podacima poslužitelju vremenske prognoze
  2. Vremenska prognoza za grad u kojoj se škola nalazi prikazuje se na početnoj stranici

## UC12 - Dohvat navigacije

- **Glavni sudionik:** Poslužitelj navigacije
- **Cilj:** Slanje točnog podatka o lokaciju i načinu dolaska od lokacije škole do željene lokacije
- **Sudionici:** Baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. Aplikacija šalje zahtjev za podacima poslužitelju navigacije
  2. Slanje obavijesti

## UC13 - Izdavanje potvrde o upisu

- **Glavni sudionik:** Upisan učenik
- **Cilj:** Generiranje i slanje potvrde o upisu
- **Sudionici:** Baza podataka, poslužitelj elektroničke pošte
- **Preduvjet:** Prijavljen je upisan učenik te se potvrda izdaje za definiranu potrebu (javnu uslugu kao što je prijevoz ili prehrana)
- **Opis osnovnog tijeka:**
  1. Učenik odabire opciju izdavanja potvrde o upisu
  2. Sustav generira potvrdu
  3. Sustav šalje potvrdu elektroničkom poštom

## UC14 - Pregled izvještaja

- **Glavni sudionik:** Djelatnik studentske službe, ravnatelj
- **Cilj:** Pregled statistike o izdavanju potvrda
- **Sudionici:** Baza podataka
- **Preduvjet:** Prijavljen je djelatnik studentske službe ili ravnatelj
- **Opis osnovnog tijeka:**
  1. Djelatnik studentske službe ili ravnatelj bira opciju pregleda izvještaja

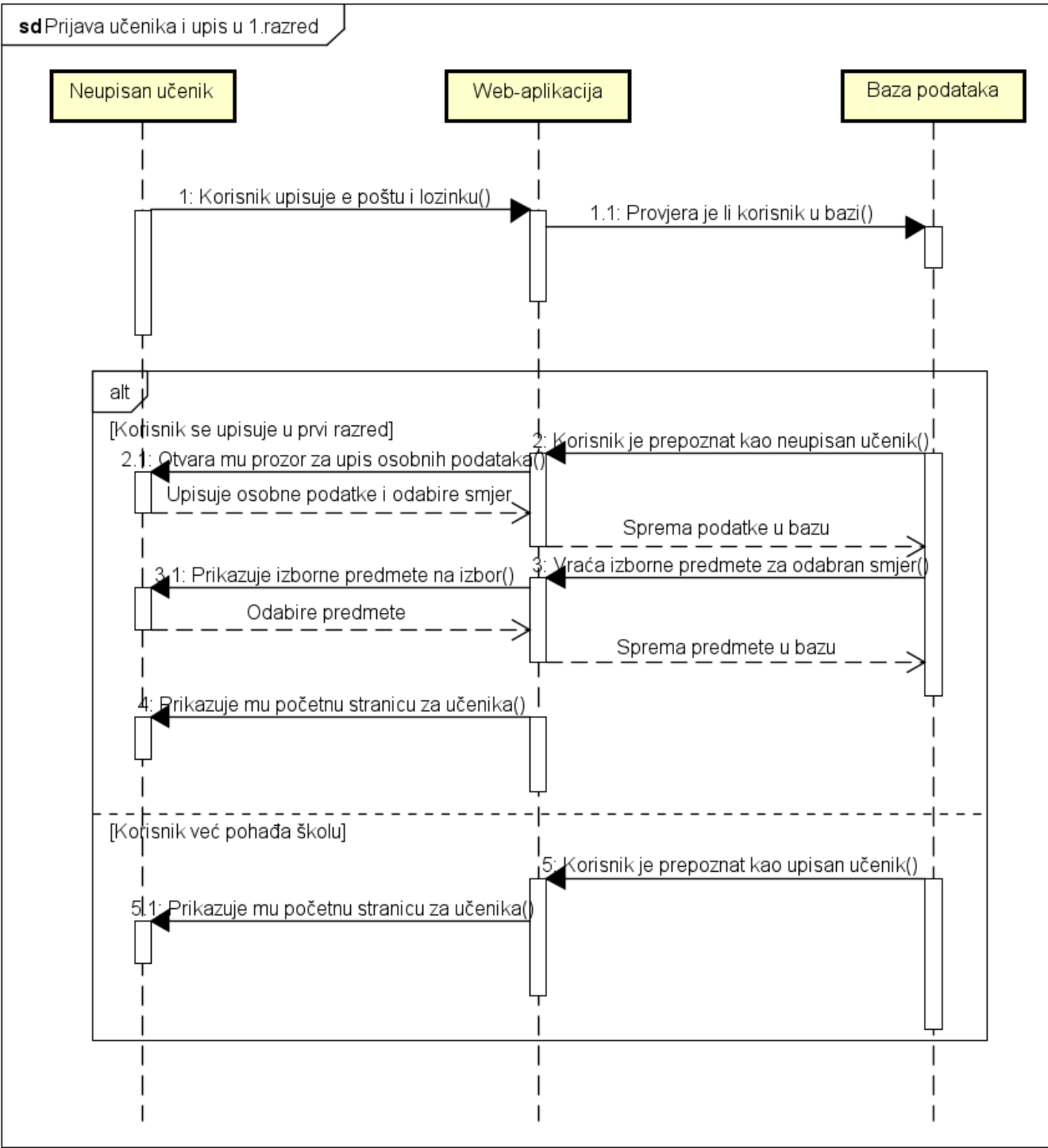
# Sekvencijski Dijagrami

---

## UC1 - Upisivanje učenika u 1. razred

Neupisan učenik prijavljuje se svojom elektroničkom poštom i lozinkom. Pojavljuje se opcija za upis u 1. razred s pomoću upisa osobnih podataka. Nakon upisa osobnih podataka poslužitelj sprema podatke vezane uz korisnika u bazu podataka. Korisnik odabire smjer koji želi upisati te ga sustav sprema u bazu podataka. Nakon uspješnog spremanja podataka baza podataka sustavu šalje dostupne izborne predmete za odabrani

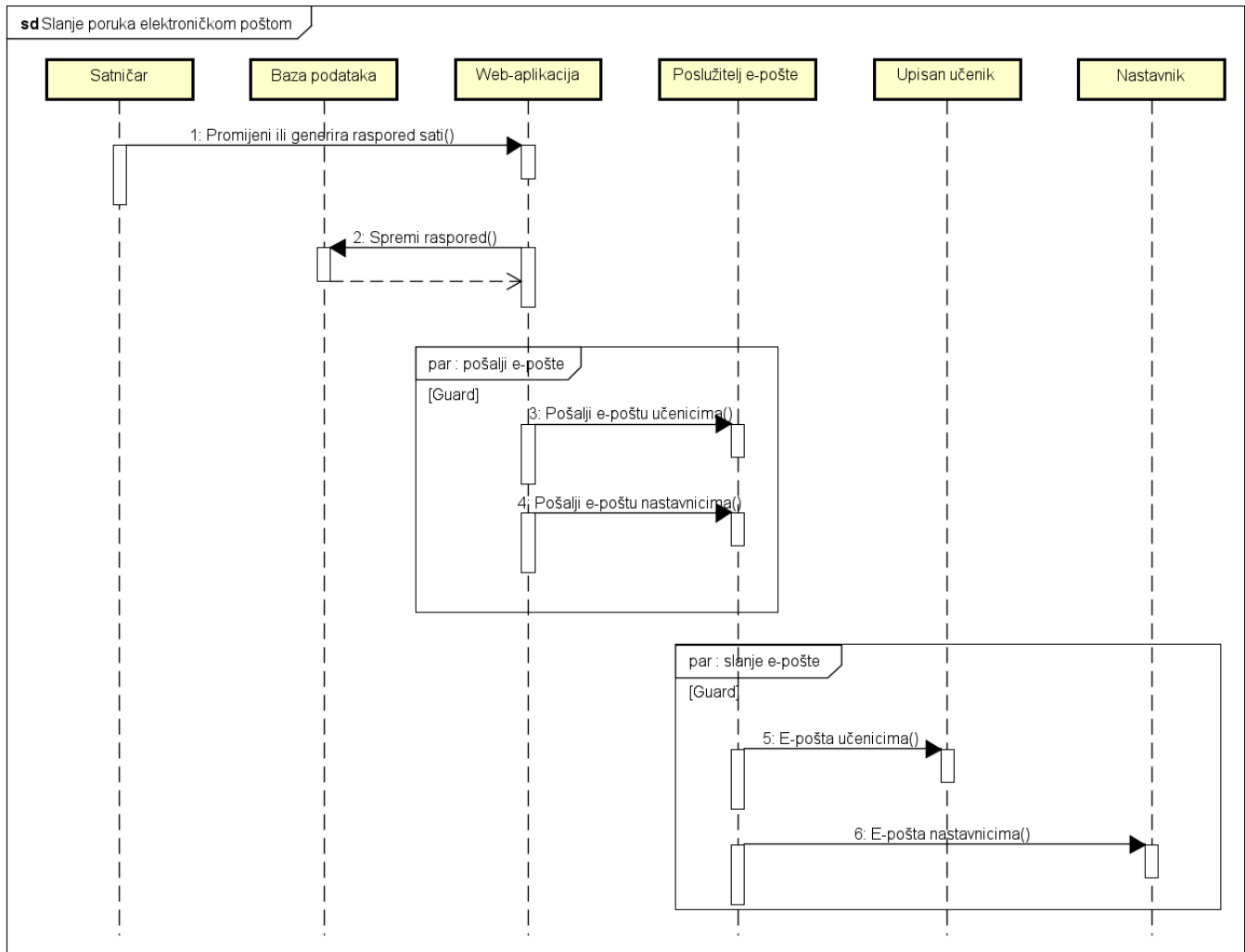
smjer te sustav korisniku nudi odabir izbornog predmeta. Korisnik odabire predmete te se nakon njihovog spremanja u bazu podataka prikazuje stranica odabranog smjera. Dodatno ako je učenik koji se želi upisati već upisan u školu automatski mu se otvara početna stranica.



Slika 3.4: Sekvencijski dijagram za UC1 - Upisivanje učenika u 1. razred

UC8 - Slanje poruke elektroničkom poštom

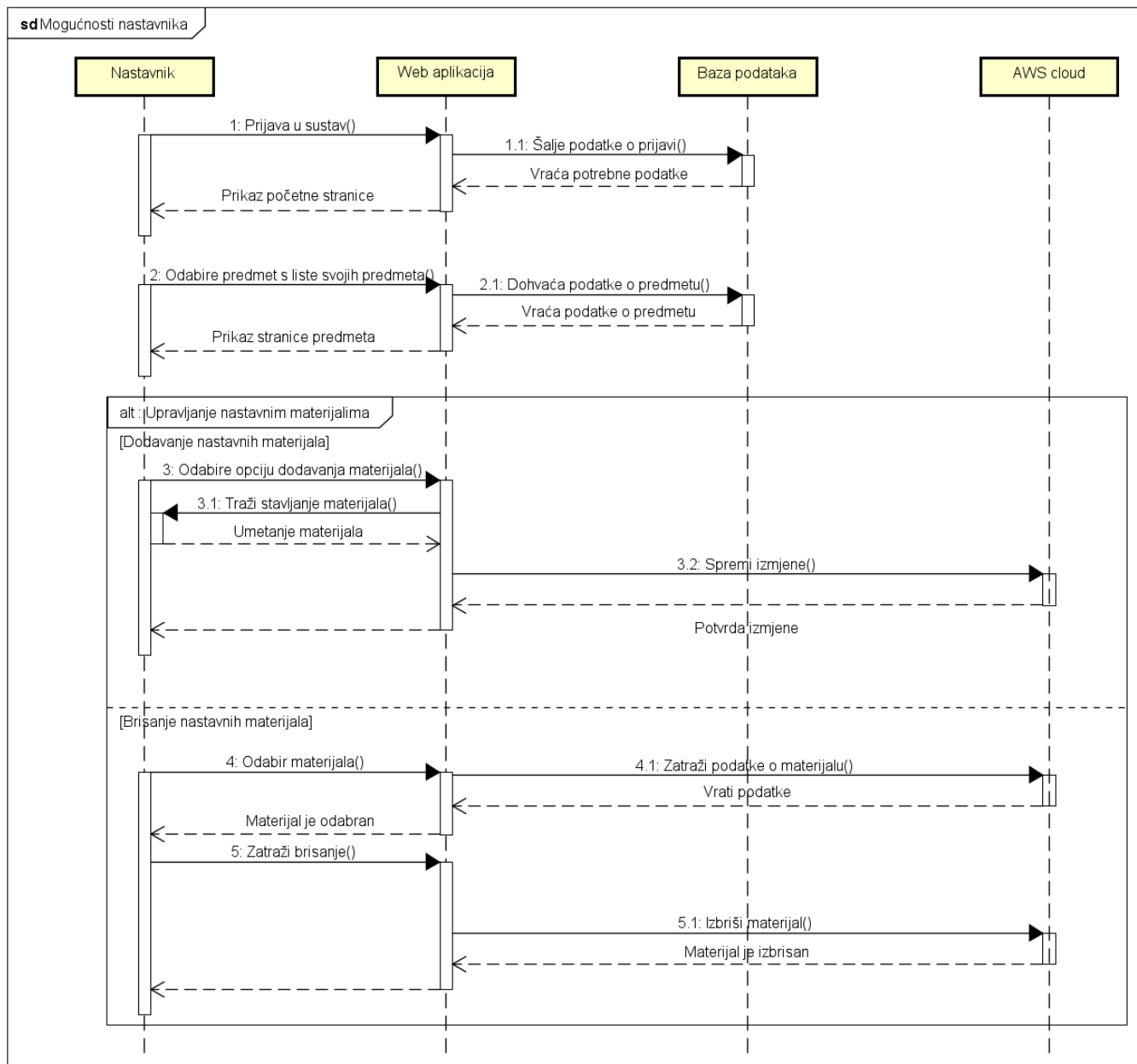
Svakom promjenom rasporeda sustav poslužitelju e-pošte šalje zahtjev za paralelno slanje e-pošte učenicima i nastavnicima u svrhu obavješćavanja o promjenama. Poslužitelj svakom učeniku i nastavniku šalje poruku.



Slika 3.5: Sekvencijski dijagram za UC8 - Slanje poruke elektroničkom poštom

## UC9 - Dodavanje nastavnih materijala

Nastavnik se prijavljuje u Web aplikaciju nakon čega mu se otvara početna stranica. S liste predmeta koje predaje odabire onaj kojem želi dodati ili obrisati nastavni materijal. Ako želi dodati materijal odabire tu opciju nakon čega ga sustav traži da ih učitava. Nakon stavljanja materijala treba potvrditi izmjene te se materijali spremaju u bazu. Ako želi izbrisati nastavni materijal treba odabrati isti. Baza mu vraća podatke o materijalu te nastavnik može zatražiti brisanje. Sustav traži potvrdu promjena i nakon toga ih briše iz baze.



Slika 3.6: Sekvencijski dijagram za UC9 - Dodavanje nastavnih materijala Arhitektura je podijeljena u tri podsustava:

- Web poslužitelj
- Web aplikacija
- Baza podataka

Internetski preglednik služi za pregled web-stranica i njihovog višemedijskog sadržaja, a one komuniciraju s poslužiteljima slanjem zahtjeva i primanjem odgovora. Preglednik ima sposobnost interpretacije koda kojim je pisana stranica u ljudski čitljiv oblik.

Web poslužitelj je posrednik između korisnika i aplikacije, te služi kao osnova rada aplikacije. Komunikacija se ostvaruje korisničkim HTTP (engl. Hyper Text Transfer Protocol) zahtjevima koji obično u zaglavlju imaju definiranu GET ili POST metodu za dohvaćanje odnosno predaju podataka. Poslužitelj na njih odgovara dostavom traženog sadržaja.

Web aplikacija po nalogu poslužitelja pristupa ili mijenja podatke iz baze podataka i vraća HTML dokument koji se potom prikazuje korisniku u sučelju preglednika. Programski jezik koji smo odabrali za razvoj pozadinskog dijela aplikacije je "Java" zajedno sa "Spring" radnim okvirom, a za dizajn korisničkog sučelja

odabrali smo "Java", s radnim okvirom "React". Odabrano razvojno okruženje je "IntelliJ IDEA" i "eclipse". Arhitektura sustava će se temeljiti na stilističkoj varijaciji arhitekture zasnovane na događajima (engl. event based architecture) - MVC obrascu. Spring radni okvir podržava MVC (Model-View-Controller) obrazac, te kao takav ima gotove predloške koji nam olakšavaju razvoj web aplikacije. MVC obrazac omogućuje odvojen razvoj navedenih slojeva aplikacije što znatno olakšava ispitivanje, razvijanje i dodavanje novih svojstava u sustav. Dijelovi MVC obrasca su:

- Model(Model) - rješava problem interakcije korisnika s bazom podataka, predstavlja bazu i služi za komunikaciju upravljača s bazom, te je na taj način zadužen za logiku vezanu za podatke i njihov prijenos
- Prikaz(View) - prikaz podataka i njihovih reprezentacija na korisničkom sučelju
- Upravljač(Controller) - povezuje komponente modela i prikaza i služi kao posrednik između tih komponenti i korisnika, a sam nije zadužen za obradu podataka, međudjeluje s modelom kako bi dohvatio podatke i s prikazom kako bi ih prikazao korisniku

## Baza podataka

---

Kao baza podataka odabrana je relacijska baza podataka. Prvenstveno je odabrana zbog jednostavnosti organizacije podataka, pristupa istima te povezivanjem više tablica kako bi se ostvarili odnosi između dijelova baze podataka.

## Opis tablica

---

### SMJER

Atribut	Tip podatka	Opis varijable
sifSmjer	INT	primarni ključ
nazSmjer	VARCHAR(n)	naziv smjera

### PREDMET

Atribut	Tip podatka	Opis varijable
sifPredmet	INT	primarni ključ
nazPredmet	VARCHAR(n)	naziv predmeta
ukBrSatiTjedno	INT	broj sati u tjednu (opterećenje)
sifSmjer	INT	strani ključ - predmet pripada samo jednom smjeru

### NASTAVNIK

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ



Atribut	Tip podatka	Opis varijable
imeNastavnik	VARCHAR(n)	ime nastavnika
prezimeNastavnik	VARCHAR(n)	prezime nastavnika
email	VARCHAR(n)jedinstvena	e-mail adresa

## SATNIČAR

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ
imeSatnicar	VARCHAR(n)	ime satničara
prezimeSatnicar	VARCHAR(n)	prezime satničara
email	VARCHAR(n)	jedinstvena e-mail adresa

## RAVNATELJ

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ
imeRavnatelj	VARCHAR(n)	ime ravnatelja
prezimeRavnatelj	VARCHAR(n)	prezime ravnatelja
email	VARCHAR(n)	jedinstvena e-mail adresa

## AKTIVNOST

Atribut	Tip podatka	Opis varijable
sifAktivnost	INT	primarni ključ
nazAktivnost	VARCHAR(n)	naziv izvannastavne aktivnosti

## ADMIN

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ
imeAdmin	VARCHAR(n)	ime administratora
prezimeAdmin	VARCHAR(n)	prezime administratora
email	VARCHAR(n)	jedinstvena e-mail adresa

## DJELTANIK

Atribut	Tip podatka	Opis varijable
---------	-------------	----------------

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ
imeDjel	VARCHAR(n)	ime djelatnika
prezimeDjel	VARCHAR(n)	prezime djelatnika
email	VARCHAR(n)	jedinstvena e-mail adresa

## UČIONICA

Atribut	Tip podatka	Opis varijable
oznakaUc	VARCHAR(n)	oznaka/naziv učionice - primarni ključ
kapacitet	INT	kapacitet učionice

## RAZRED

Atribut	Tip podatka	Opis varijable
razred	VARCHAR(2)	naziv razreda (brojka i slovo) - primarni ključ
sifSmjer	INT	pripadni smjer - strani ključ

## UČENIK

Atribut	Tip podatka	Opis varijable
oib	VARCHAR(11)	OIB
prezimeUcenik	VARCHAR(n)	prezime
imeUcenik	VARCHAR(n)	ime
datumRodenja	DATE	datum rođenja
email	VARCHAR(n)	jedinstvena e-mail adresa učenika
spol	VARCHAR(1)	spol M/Ž
razred	VARCHAR(2)	strani ključ - određuje učenikov razred
id	VARCHAR(n)	primarni kjuč
vjeronauk	BOOLEAN	ide li na vjeronauk (ako ne - etika)

## SAT

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ - identifikacija

## RAZREDPREDMETNASTAVNIK

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ - identifikacija

## VRIJEMESATA

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ - identifikacija
dan	VARCHAR(n)	dan u tjednu
pocetakSata	TIME	vrijeme početka sata
krajSata	TIME	vrijeme kraja sata

## OBAVIJESTI

Atribut	Tip podatka	Opis varijable
sifObavijest	INT	primarni ključ - identifikacija
adresaLokacije	VARCHAR(n)	adresa lokacije terenske nastave
državaLokacije	VARCHAR(n)	država lokacije terenske nastave
gradLokacije	VARCHAR(n)	grad lokacije terenske nastave
datumObavijesti	DATE	datum obavijesti
imeNastavnik	VARCHAR(n)	ime nastavnika
prezimeNastavnik	VARCHAR(n)	prezime nastavnika
naslovObavijesti	VARCHAR(n)	naslov obavijesti
sadržajObavijesti	VARCHAR(n)	sadržaj obavijesti

## CHATMESSAGE

Atribut	Tip podatka	Opis varijable
ID	LONG	primarni ključ - identifikacija
imeGrupe	VARCHAR(n)	ime grupe
oznakaVremena	TIMESTAMP	lokalno vrijeme
pošiljatelj	VARCHAR(n)	pošiljatelj
primatelj	VARCHAR(n)	primatelj
sadržaj	VARCHAR(n)	sadržaj poruke

## ABSTRACTAUDITABLE

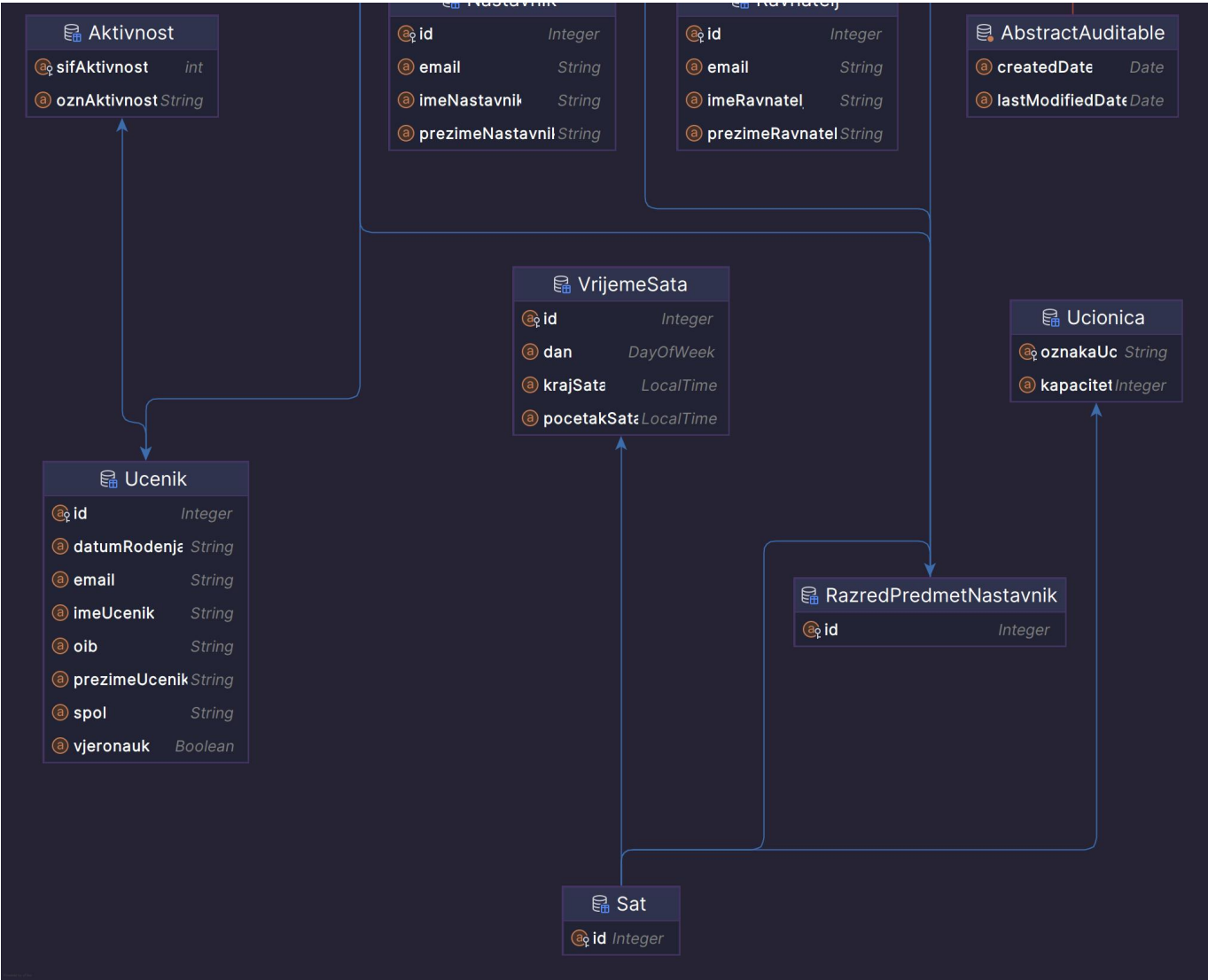
Atribut	Tip podatka	Opis varijable
createdDate	DATE	datum
lastModifiedDate	DATE	datum

# ABSTRACTPERSISTABLE

Atribut	Tip podatka	Opis varijable
id	INT	primarni ključ

## Dijagram baze podataka

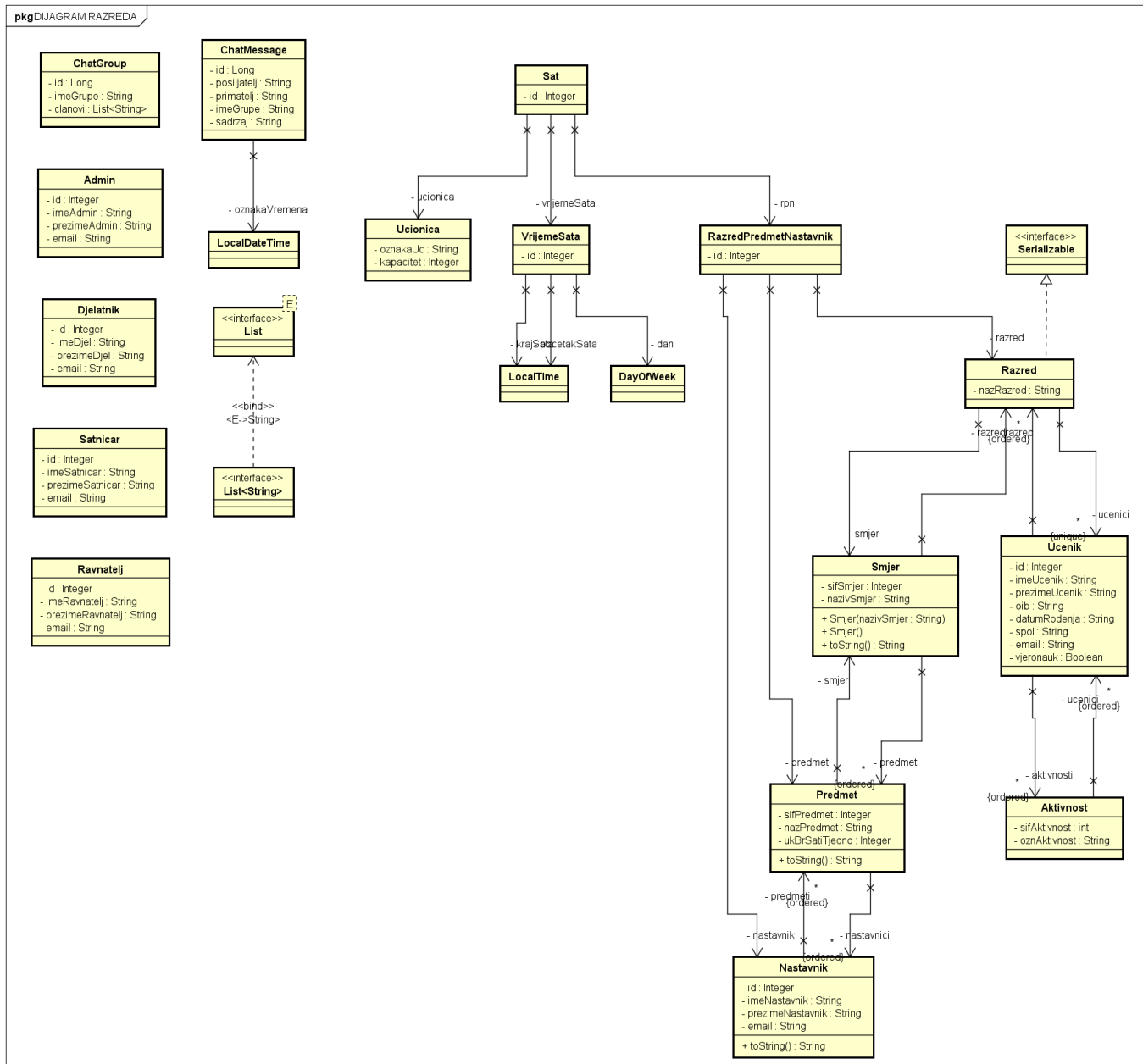




Slika 4.1: Dijagram baze podataka

## Dijagram razreda

Dijagram razreda predstavlja sve glavne atribute i entitete baze podataka. Sadrži sve djelatnike škole i učenike, sve potrebno za izradu rasporeda sati i mogućnosti aplikacije.



Slika 4.2: Dijagram razreda baze podataka

## Dinamičko ponašanje aplikacije

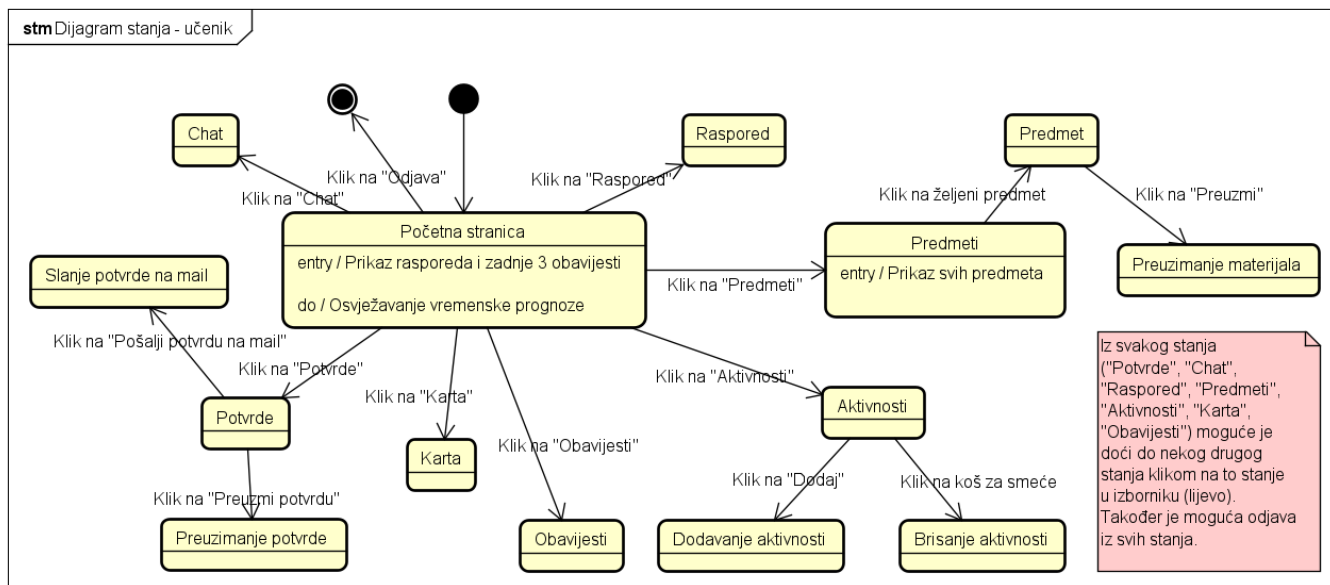
Dinamičko ponašanje aplikacije odnosi se na način na koji objekti u sustavu evoluiraju kroz vrijeme, uključujući prijelaze između različitih stanja. To uključuje aktivnosti, događaje, odluke i interakcije unutar aplikacije. UML dijagrami stanja omogućuju vizualizaciju tih promjena i olakšavaju razumijevanje dinamike sustava.

Razumijevanje promjena stanja neophodno je za pravilno funkcioniranje aplikacije jer pruža uvid u interakcije među objektima, komponentama i korisnicima tijekom rada sustava. Korištenjem UML dijagrama stanja i aktivnosti moguće je vizualizirati prijelaze i stanja objekata, identificirati potencijalne probleme, osigurati točnu implementaciju te poboljšati komunikaciju među članovima tima.

## UML dijagram stanja

Učeniku se nakon prijave otvara početna stranica na kojoj se nalazi vremenska prognoza, zadnje tri obavijesti te raspored sati. U svakom trenutku imaju mogućnost odjaviti se. Klikom na "Chat" imaju mogućnost odabrati

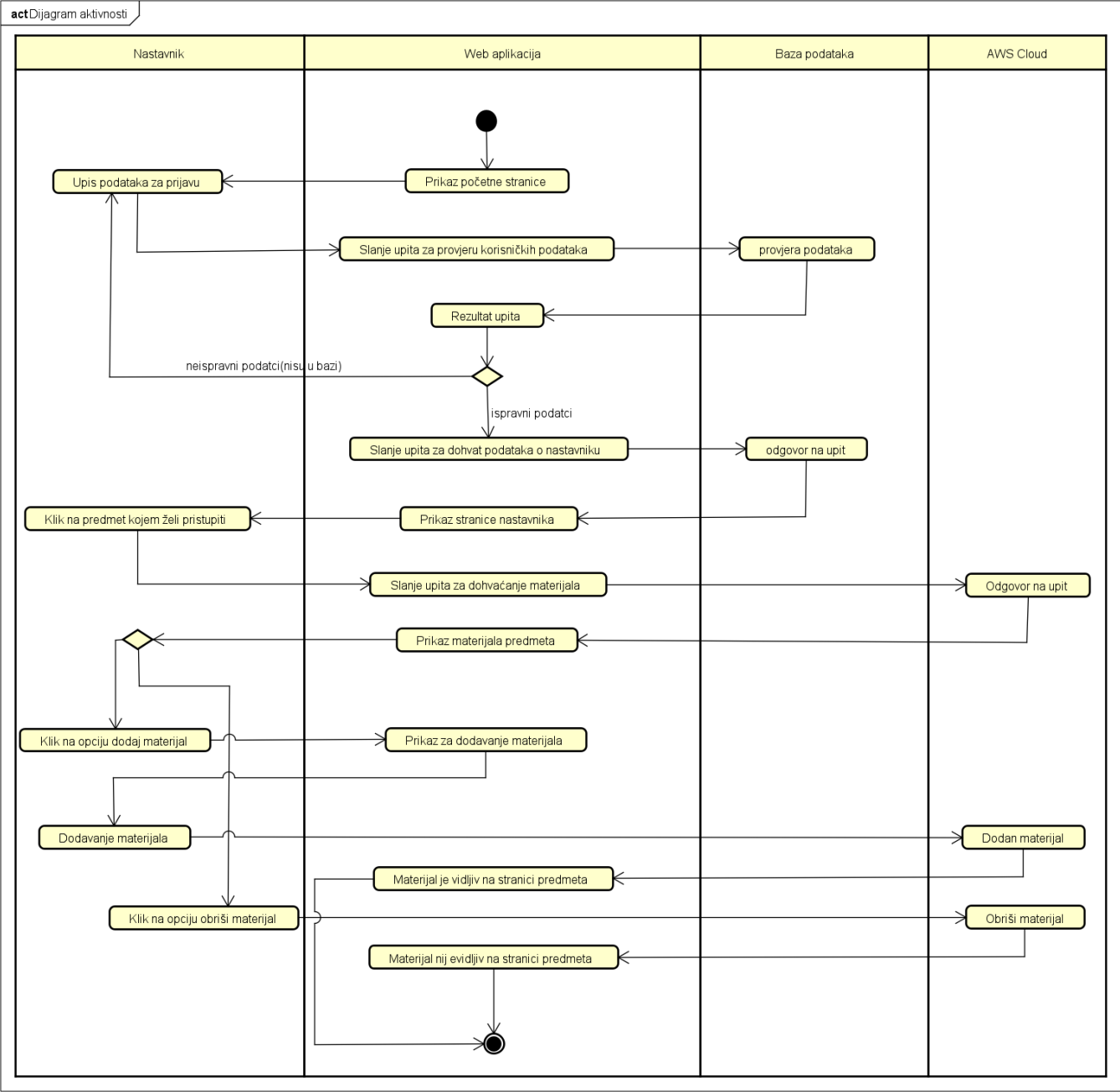
korisnika s kojim žele komunicirati te im poslati poruku. Klikom na "Potvrde" imaju mogućnosti preuzeti svoju potvrdu o školovanju u školskoj godini. Klikom na "Raspored" imaju prikaz cijelog rasporeda. Na "Aktivnosti" im se prikazuju njihove upisane aktivnosti, a imaju mogućnosti obrisati ili dodati iste. Klikom na "Predmeti" prikazuju im se svi upisani predmeti, te klikom na svaki predmet otvara im se prikaz materijala iz tog predmeta te imaju mogućnost skinuti isti.



Slika 4.3: UML dijagram stanja

## UML dijagrami aktivnosti

Dijagram aktivnosti na slici 5.4 prikazuje mogućnosti nastavnika kod dodavanja, odnosno brisanja materijala. On na svojoj stranici odabire kojem predmetu želi dodati/obrisati materijal. Nakon odabira predmeta obavlja željenu akciju. Materijali se spremaju na AWS Cloud.



Slika 4.4: UML dijagrami aktivnosti Arhitektura sustava predstavlja temeljni okvir za razumijevanje i implementaciju svih njegovih funkcionalnosti. U kontekstu razvojne dokumentacije aplikacija, dijagrami komponentata i razmještaja odlučujući su za prikaz povezanosti i rasporeda različitih komponentata sustava. Ovi dijagrami omogućuju sudionicima projekta razumijevanje i vizualizaciju fizičkog i logičkog dizajna sustava, uključujući interakcije između dijelova aplikacije, što je odlučujuće za efikasnu implementaciju i dugoročnu održivost sustava.

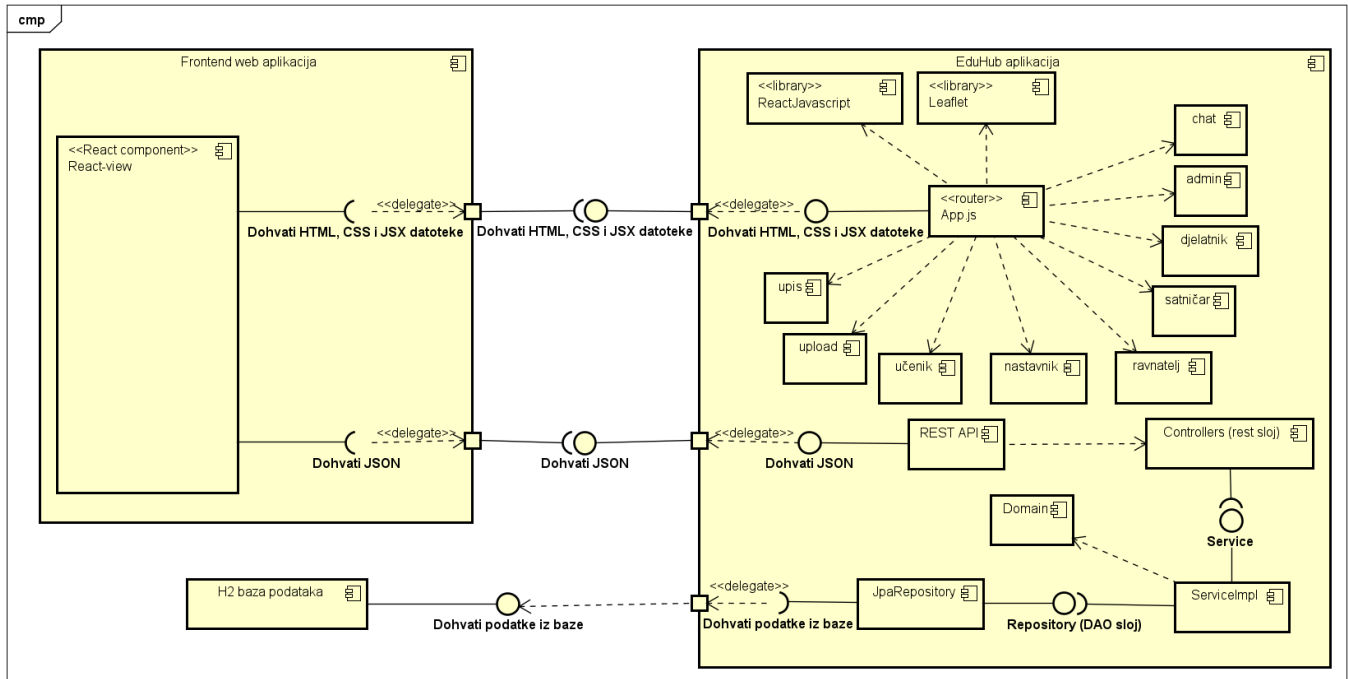
Arhitektura sustava, u kontekstu dijagrama komponentata i razmještaja, pruža uvid u strukturu i raspored ključnih dijelova aplikacije. Ovi dijagrami nisu korisni samo tijekom faza oblikovanja i implementacije, već služe i kao alati za održavanje i optimizaciju sustava u budućnosti.

## Dijagram komponentata

Komponente sustava predstavljaju bitne dijelove aplikacije koji obavljaju specifične funkcije. Svaka komponenta je autonomna jedinica s vlastitim odgovornostima, ali je povezana s drugim komponentama



kako bi sustav u cjelini funkcionirao. Sustavu se pristupa preko dva različita sučelja. Preko sučelja za dohvat HTML, CSS i JSX datoteka poslužuju se datoteke koje pripadaju prednjem dijelu aplikacije. App.js (router) je komponenta koja na upit s url određuje koja datoteka će se poslužiti na sučelje. Frontend dio se sastoji od niza JavaScript datoteka koje su raspoređene u logičke cjeline nazvane po dijelu aplikacije za koji se koriste. Sve JavaScript datoteke ovise o React i Leaflet bibliotekama iz koje dohvaćaju gotove komponente. Preko sučelja za dohvat JSON podataka pristupa se REST API komponenti. REST API poslužuje podatke koji pripadaju backend dijelu aplikacije. JpaRepository je zadužen za dohvaćanje tablica iz baze podataka s pomoću SQL upita.

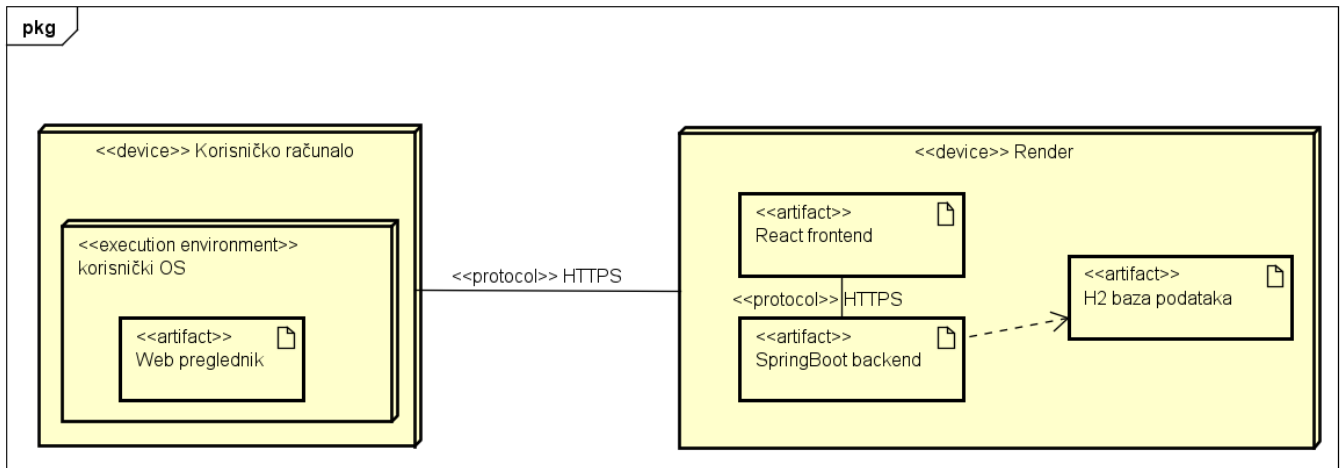


Slika 5.1: UML dijagram komponenta

## Dijagram razmještaja

UML dijagram razmještaja prikazuje fizičku ili virtualnu raspodjelu komponenta sustava unutar infrastrukture. Cilj je prikazati kako su komponente raspoređene (npr. na poslužiteljima, u okruženjima oblaka ili na uređajima krajnjih korisnika) te način komuni čije, API-ja ili drugih komunikacijskih protokola. Na poslužiteljskom računalu nalazi se Render. Na poslužitelju je aktivan proces programa aplikacije frontend i backend koji komunicira s bazom podataka. Predviđeno je da korisnik koristi mrežni preglednik na vlastitom računalu za komunikaciju s aplikacijom na mrežnom poslužitelju. Komunikacija između korisnika i poslužitelja

odvija se preko HTTPS veze.



Slika 5.2: UML dijagram razmještaja

Ovo poglavlje treba opisati provedena ispitivanja implementiranih funkcionalnosti na razini komponenti i sustava. Fokus je na odabiru i izvedbi ispitnih slučajeva koji obuhvaćaju redovne, rubne uvjete i testiranje grešaka, kao i upotrebu odgovarajućih alata za provedbu testiranja.

## Ispitivanje komponenti

Cilj ispitivanja komponenti je provjera osnovnih funkcionalnosti implementiranih u razredima sustava. Ovdje je potrebno izolirati svaku komponentu kako bi se testirala njezina ispravnost i reakcija na različite scenarije.

### 1. Uspješno dodavanje satničara

#### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj testira funkcionalnost dodavanja novog satničara u sustav pomoću metode `addSatnicar` u klasi `AdminServiceJpa`. Provjerava se je li sustav ispravno pohranio podatke o satničaru i vratio spremljeni objekt.

#### Ulazni podaci

Email: `satnicar@eduxhub.onmicrosoft.com`

Ime: `ImeSatnicara`

Prezime: `PrezimeSatnicara`

#### Očekivani rezultati

- Spremljeni `Satnicar` objekt ne smije biti `null`.
- Spremljeni `Satnicar` treba imati email `satnicar@eduxhub.onmicrosoft.com`.
- Spremljeni `Satnicar` treba imati ime `ImeSatnicara`.
- Spremljeni `Satnicar` treba imati prezime `PrezimeSatnicara`.
- Metoda `save()` na repozitoriju treba biti pozvana točno jednom.

#### Dobiveni rezultati

- Spremljeni `Satnicar` nije `null`.
- Spremljeni `Satnicar` ima email `satnicar@eduxhub.onmicrosoft.com`.
- Spremljeni `Satnicar` ima ime `ImeSatnicara`.
- Spremljeni `Satnicar` ima prezime `PrezimeSatnicara`.
- Metoda `save()` bila je pozvana točno jednom.

**Rezultat testa:** Prolaz

## Postupak provođenja ispitivanja

1. Mockati `SatnicarRepository` kako bi metoda `save()` vraćala `Satnicar` objekt.
2. Pripremiti `AddDTO` objekt s potrebnim ulaznim podacima (email, ime, prezime).
3. Pozvati metodu `addSatnicar` u servisu `AdminServiceJpa`.
4. Provjeriti da spremljeni `Satnicar` objekt ima očekivane vrijednosti za email, ime i prezime.
5. Osigurati da je metoda `save()` pozvana točno jednom na `SatnicarRepository`.

[//video](#)

## 2. Greška kod dodavanja dupliciranog administratora

### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj testira kako sustav reagira na pokušaj dodavanja novog administratora s emailom koji već postoji u sustavu. Cilj je osigurati da sustav baci odgovarajuću iznimku i ne pokuša spremiti duplicirani unos.

### Ulazni podaci

Postojeći administrator:

- Email: `josip.jelavic@eduxhub.onmicrosoft.com`
- Ime: `Josip`
- Prezime: `Jelavić`

Novi administrator (pokušaj dodavanja):

- Email: `josip.jelavic@eduxhub.onmicrosoft.com`
- Ime: `NovoIme`
- Prezime: `NovoPrezime`

### Očekivani rezultati

- Sustav baca iznimku `IllegalArgumentException` s porukom: `"Admin s ovim e-mailom već postoji."`
- Metoda `save()` na repozitoriju ne smije biti pozvana.

### Dobiveni rezultati

- Sustav je bacio iznimku `IllegalArgumentException` s točnom porukom: `"Admin s ovim e-mailom već postoji."`
- Metoda `save()` nije bila pozvana.

**Rezultat testa:** Prolaz

## Postupak provođenja ispitivanja

1. Mockati `AdminRepository` tako da metoda `findByEmail()` za email `josip.jelavic@eduxhub.onmicrosoft.com` vraća postojećeg administratora.
2. Pripremiti `AddDTO` objekt s podacima o novom administratoru (email, ime, prezime).
3. Pozvati metodu `addAdmin` u servisu `AdminServiceJpa`.
4. Provjeriti da sustav baca `IllegalArgumentException` s odgovarajućom porukom.
5. Osigurati da metoda `save()` na `AdminRepository` nije pozvana.

[//video](#)

## 3. Uspješno dodavanje aktivnosti učeniku

### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj testira funkcionalnost dodavanja nove aktivnosti postojećem učeniku s pomoću metode `dodajAktivnostiPoNazivu` u servisu `UcenikServiceJpa`. Cilj je osigurati da se nova aktivnost pravilno doda na popis postojećih aktivnosti učenika i da se spremi u bazu podataka.

### Ulazni podaci

Učenik:

- Email: `nikola.nikic@eduxhub.onmicrosoft.com`
- Ime: `Nikola`
- Prezime: `Nikić`
- Razred: `1a`
- Datum rođenja: `2009-10-29`
- Spol: `M`
- Vjeronauk: `true`
- Postojeće aktivnosti:
  - `Nogomet`
  - `Dodatna_nastava_Matematika`
  - `Dodatna_nastava_Engleski_jezik`

Nova aktivnost za dodavanje:

- `Šah`

### Očekivani rezultati

- Nova aktivnost `Šah` treba biti dodana na popis aktivnosti učenika.
- Učenikov popis aktivnosti treba sadržavati:
  - `Nogomet`
  - `Dodatna_nastava_Matematika`
  - `Dodatna_nastava_Engleski_jezik`
  - `Šah`

- Metoda `save()` na repozitoriju učenika (`ucenikRepository`) treba biti pozvana točno jednom.

### Dobiveni rezultati

- Nova aktivnost `Šah` je dodana na popis aktivnosti učenika.
- Učenikov popis aktivnosti sadržava:
  - `Nogomet`
  - `Dodatna_nastava_Matematika`
  - `Dodatna_nastava_Engleski_jezik`
  - `Šah`
- Metoda `save()` na `ucenikRepository` bila je pozvana točno jednom.

### Rezultat testa: Prolaz

### Postupak provođenja ispitivanja

1. Mockati `ucenikRepository` tako da metoda `findByEmail()` za email `nikola.nikic@eduxhub.onmicrosoft.com` vraća postojećeg učenika s definiranim podacima.
2. Mockati `aktivnostService` tako da metoda `findByOznAktivnosti()` za oznaku `Šah` vraća listu koja sadrži aktivnost `Šah`.
3. Pozvati metodu `dodajAktivnostiPoNazivu` na servisu `UcenikServiceJpa` s emailom učenika i oznakom nove aktivnosti.
4. Provjeriti da popis aktivnosti učenika sada sadrži i novu aktivnost `Šah`.
5. Osigurati da je metoda `save()` na `ucenikRepository` pozvana točno jednom.

[//video](#)

### 4. Promjena kapaciteta učionice

#### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj provjerava funkcionalnost promjene kapaciteta učionice putem metode `promijeniKapacitet` unutar kontrolera `RavnateljController`. Cilj je osigurati da se kapacitet učionice uspješno ažurira te da odgovarajući status i poruka budu vraćeni klijentu.

#### Ulazni podaci

Učionica:

- Oznaka: `A202`
- Trenutni kapacitet: `20`
- Novi kapacitet: `50`

#### Očekivani rezultati

- Kapacitet učionice se uspješno mijenja iz `20` u `50`.
- Vraća se HTTP status `200 OK` s porukom `"Učionici uspješno promijenjen kapacitet"`.
- Metoda `findById()` na servisu učionica (`ucionicaService`) pozvana je točno dva puta (jednom za dohvaćanje prije promjene i jednom nakon).

## Dobiveni rezultati

- Kapacitet učionice je promijenjen iz 20 u 50.
- HTTP status je 200 OK, a poruka je "Učionici uspješno promijenjen kapacitet".
- Metoda `findById()` na servisu `ucionicaService` bila je pozvana točno dva puta.

**Rezultat testa:** Prolaz

## Postupak provođenja ispitivanja

1. Mockati `ucionicaService` tako da metoda `findById()` za oznaku učionice A202 vraća učionicu s kapacitetom 20.
2. Pozvati metodu `promijeniKapacitet` na kontroleru `RavnateljController` s oznakom učionice i novim kapacitetom 50.
3. Provjeriti da je HTTP status odgovora 200 OK i da je poruka odgovora "Učionici uspješno promijenjen kapacitet".
4. Provjeriti da je metoda `findById()` pozvana dva puta.

[//video](#)

## 5. Dodavanje učenika u nepostojeći razred

### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj provjerava ponašanje sustava kada se pokuša dodati učenika u razred koji ne postoji. Cilj je osigurati da se u takvom slučaju generira odgovarajuća greška i da učenik nije spremljen u bazu podataka.

### Ulazni podaci

#### Podaci učenika:

- Email: `ucenik@eduxhub.onmicrosoft.com`
- Ime: `Ime`
- Prezime: `Prezime`
- OIB: `12345678912`
- Spol: `M`
- Vjeronauk: `true`
- Datum rođenja: `2000-01-01`

#### Podaci razreda:

- Naziv razreda: `5a`
- Razred ne postoji u sustavu (mock vraća `null`).

### Očekivani rezultati

- Metoda `adminService.addUcenik()` baca `IllegalArgumentException` s porukom "Razred s ovim nazivom ne postoji".
- Metoda `ucenikRepository.save()` nije pozvana jer unos učenika nije uspješan.

## Dobiveni rezultati

- Exception `IllegalArgumentException` bačen je s odgovarajućom porukom.
- Metoda `ucenikRepository.save()` nije bila pozvana, što potvrđuje da učenik nije spremljen u bazu.

**Rezultat testa:** Prolaz

## Postupak provođenja ispitivanja

1. Mockati `razredService` tako da metoda `findByNazivRazred("5a")` vraća `null`.
2. Kreirati instancu objekta `AdminAddUcenikDTO` s podacima učenika i nepostojećim razredom (`5a`).
3. Pozvati metodu `adminService.addUcenik()` s DTO objektom.
4. Provjeriti da je bačen exception `IllegalArgumentException` s točno definiranim opisom greške.
5. Provjeriti da metoda `ucenikRepository.save()` nije bila pozvana.

//video

## 6. Testiranje nepostojeće funkcionalnosti

### Funkcionalnost koju testiramo:

Ovaj ispitni slučaj provjerava kako sustav reagira kada se pozove metoda koja nije implementirana. Simuliramo exception koji označava nepostojeću funkcionalnost i testiramo očekivano ponašanje.

### Ulazni podaci

#### Pozvana metoda:

- `deleteAdminByName("Nikola")`

#### Simulirani exception:

- `UnsupportedOperationException` s porukom "Metoda nije implementirana."

### Očekivani rezultati

- Metoda `deleteAdminByName("Nikola")` baca `UnsupportedOperationException` s točno definiranim opisom greške.
- Metoda `deleteAdminByName()` je pozvana točno jednom.

## Dobiveni rezultati

- Exception `UnsupportedOperationException` bačen je s odgovarajućom porukom.
- Metoda `deleteAdminByName("Nikola")` pozvana je jednom, što je u skladu s očekivanjem.

**Rezultat testa:** Prolaz

## Postupak provođenja ispitivanja

1. Mockati instancu `AdminServiceJpa` koristeći Mockito.

2. Postaviti simulirano ponašanje za metodu `deleteAdminByName("Nikola")` tako da baca `UnsupportedOperationException` s porukom "Metoda nije implementirana."
3. Koristiti `assertThatThrownBy` za potvrdu da je exception bačen i da sadrži očekivanu poruku.
4. Provjeriti pomoću `verify` da je metoda `deleteAdminByName("Nikola")` pozvana točno jednom.

[//video](#)

## Ispitivanje sustava

---

Cilj ispitivanja sustava je testiranje ponašanja cijelog sustava u uvjetima stvarnog korištenja, uz posebnu pažnju na međusobnu povezanost svih komponenti. Ispitivanje treba obuhvatiti sve aspekte sustava i njegovu interakciju s korisnicima.

### 1. Prijava korisnika s odgovarajućom ulogom (administrator) u sustav - Redovni slučaj

#### Ulazi

- E-mail korisnika: `josip.jelavic@eduxhub.onmicrosoft.com`
- Lozinka: `Lozinka2`
- Klikanje gumba za prijavu
- Unos emaila i lozinke

#### Koraci ispitivanja

1. Otvoriti aplikaciju na linku i maksimizacija prozora
2. Pronaći gumb "Prijava uz Microsoft"
3. Unijeti e-mail
4. Kliknuti gumb "Next"
5. Unijeti lozinku
6. Kliknuti gumb "Sign in"
7. Kliknuti gumb "No" na pitanje o ostanku prijavljenim
8. Dohvatiti tekst dobrodošlice s vrha stranice
9. Provjeriti očekivani `Pozdrav, Josip Jelavić!` i dobiveni rezultat koji će se prikazati u konzoli

#### Očekivani izlaz

Nakon prijave aplikacija se preusmjerava na početnu stranicu s pozdravom. Interno se provjerava odgovara li tekst dobrodošlice očekivanom i nakon toga se zatvara stranica.

#### Dobiveni izlaz

[//video](#)

### 2. Prijava korisnika s krivom lozinkom u sustav - Redovni slučaj

#### Ulazi

- E-mail korisnika: `josip.jelavic@eduxhub.onmicrosoft.com`



- Lozinka: **krivaLozinka**
- Klikanje gumba za prijavu
- Unos emaila i lozinke

### Koraci ispitivanja

1. Otvoriti aplikaciju na linku i maksimizacija prozora
2. Pronaći gumb "Prijava uz Microsoft"
3. Unijeti e-mail
4. Kliknuti gumb "Next"
5. Unijeti lozinku
6. Kliknuti gumb "Sign in"
7. Provjeriti postojanje poruke o pogrešnoj lozinci

### Očekivani izlaz

Nakon pokušaja prijave ispisuje se pogreška da je netočna lozinka. Interno se provjerava pojavljuje li se poruka o pogrešnoj lozinci i nakon toga se zatvara stranica.

### Dobiveni izlaz

[//video](#)

### 3. Dodavanje ravnatelja s postojećim e-mailom - Rubni slučaj

#### Ulazi

- E-mail korisnika: **josip.jelavic@eduxhub.onmicrosoft.com**
- Lozinka: **Lozinka2**
- Klikanje gumba za prijavu
- Unos emaila i lozinke
- Klikanje gumba za dodavanje ravnatelja
- Ime ravnatelja: **Davor**
- Prezime ravnatelja: **Čerić**
- E-mail ravnatelja: **davor.ceric@eduxhub.onmicrosoft.com**

### Koraci ispitivanja

1. Otvoriti aplikaciju na linku i maksimizacija prozora
2. Pronaći gumb "Prijava uz Microsoft"
3. Unijeti e-mail
4. Kliknuti gumb "Next"
5. Unijeti lozinku
6. Kliknuti gumb "Sign in"
7. Kliknuti gumb "No" na pitanje o ostanku prijavljenim
8. Pronaći u lijevom *sidebar*-u "Ravnatelji"
9. Unijeti ime, prezime i e-mail ravnatelja
10. Kliknuti gumb "Dodaj ravnatelja"

## Očekivani izlaz

Nakon klika na dodavanje ravnatelja aplikacija javlja grešku te se provjerava ispis greške s očekivanim **Greška: Greška prilikom dodavanja korisnika..** Nakon provjere zatvara se stranica.

## Dobiveni izlaz

[//video](#)

4. Administrator pokušava preuzeti potvrdu (dostupna učeniku) - Poziv nepostojećih funkcionalnosti

## Ulazi

- E-mail korisnika: **josip.jelavic@eduxhub.onmicrosoft.com**
- Lozinka: **Lozinka2**
- Klikanje gumba za prijavu
- Unos emaila i lozinke
- Unos URL-a za učenikovu podstranicu **https://eduhub-rfsg.onrender.com/ucenik**
- Klikanje gumba "Potvrde" i "Preuzmi potvrdu"

## Koraci ispitivanja

1. Otvoriti aplikaciju na linku i maksimizacija prozora
2. Pronaći gumb "Prijava uz Microsoft"
3. Unijeti e-mail
4. Kliknuti gumb "Next"
5. Unijeti lozinku
6. Kliknuti gumb "Sign in"
7. Kliknuti gumb "No" na pitanje o ostanku prijavljenim
8. Unos URL-a za učenikovu podstranicu
9. Pronaći u lijevom *sidebar*-u "Potvrde"
10. Kliknuti gumb preuzmi potvrdu

## Očekivani izlaz

Nakon klika na preuzimanje potvrde aplikacija javlja grešku te se provjerava ispis greške s očekivanim **Došlo je do greške prilikom generiranja potvrde.** jer administrator ne može preuzeti potvrdu. Nakon provjere zatvara se stranica. U konzoli se ispisuje uspješan test.

## Dobiveni izlaz

[//video](#)

# Korištene tehnologije i alati

---

**Cilj:** Jasno i precizno opisati tehnologije korištene u projektu kako bi se olakšalo održavanje, proširenje i suradnja u timu.

Uključite informacije:

### 1. Programski jezici:

- Java 21

### 2. Radni okviri i biblioteke:

- Spring Boot 3.3.5: Glavni framework za razvoj backend aplikacije. Odabran zbog jednostavne konfiguracije i podrške za različite module. Korišteni moduli uključuju:
  - Spring Boot Starter Web: Omogućava razvoj REST API-ja.
  - Spring Boot Starter Data JPA: Pruža integraciju s bazom podataka
  - Spring Boot Starter Security: Implementacija sigurnosnih protokola
  - Spring Boot Starter Mail: Slanje emailova.
  - Spring Boot Starter WebSocket: Podrška za WebSocket veze
  - Spring Boot Starter Thymeleaf
  - Spring Boot Starter OAuth2: implementacija OAuth2
- iTextPDF 5.5.13.3: Korišten za generiranje PDF dokumenata
- Amazon S3 SDK 2.20.93: Integracija s Amazon S3 za pohranu podataka u oblaku
- Jackson-Datatype-JSR310: Korišten za obradu podataka vezanih uz vrijeme i datume.
- Lombok: pruža anotacije za lakše upravljanje objektima
- React 18.3.1: Glavni framework za razvoj korisničkog sučelja. Omogućava izradu dinamičkih i responzivnih aplikacija.
- React Router DOM 6.27.0: Koristi se za upravljanje rutama unutar aplikacije.
- Leaflet 1.9.4: Biblioteka za prikaz interaktivnih karata i rad s geolokacijskim podacima.
  - React-Leaflet 4.2.0: React integracija za Leaflet.
  - Leaflet Control Geocoder 3.1.0: Korišten za geokodiranje i pretragu adresa.
  - Leaflet Routing Machine 3.2.12: Omogućava izračunavanje i prikaz ruta na kartama.
- Axios 1.7.9: Korišten za slanje HTTP zahtjeva i rad s REST API-jem.
- Recharts 2.15.0: Biblioteka za prikaz vizualnih podataka kroz grafikone i dijagrame.
- React Icons 5.4.0: Korištena za prikaz vektorskih ikonica unutar aplikacije.
- React Scripts 5.0.1: Paket koji omogućava jednostavno pokretanje, buildanje i testiranje React aplikacije.
- OpenWeather API: Korišten za dohvat ikonica za vremenske uvjete
- WeatherAPI: Korišten za dohvat temperature i osnovnih podataka o vremenskim uvjetima

### 3. Baza podataka:

- H2 baza podataka

### 4. Razvojni alati:

- Eclipse IDE(2024-09 4.33.0)
- IntelliJ IDEA (2024.3)
- Postman: Korišten za testiranje REST API-ja
- Maven 4.0.0
- Visual Studio Code 1.96.4

### 5. Alati za ispitivanje:

- JUnit

- Selenium 4.27.0

#### 6. Alati za razmještaj:

- Render

#### 7. Cloud platforma:

- AWS (Amazon Web Services)

#### 8. Mail provider

- Brevo Ovaj odjeljak dokumentacije treba dati detaljne smjernice za instalaciju, konfiguraciju, pokretanje i administraciju aplikacije. Cilj je olakšati postavljanje aplikacije na razvojnom, ispitnom i produkcijskom okruženju.

## 1. Instalacija

### Preduvjeti

- Node.js 22.x
- Docker 20.x
- IntelliJ 2024.x ili sličan IDE

### Koraci

1. Kloniranje repozitorija:

```
git clone https://github.com/bornaodobasic/eduhub.git
cd eduhub
git checkout deploy-bigMerge
```

2. Instalacija ovisnosti:

- npm install

## 2. Postavke

### Konfiguracijske datoteke

- backend\src\main\resources\db - inicijalna baza podataka
- .env - podaci o svim ključevima

Svaka izmjena frontend foldera zahtijeva navigaciju basha u **frontend** te pokretanje **npm run build**. Nakon toga u \frontend stvara se build folder koji treba izrezati i zalijepiti u backend\src\main\resources\db. Preimenovati ga u **static**.

## 3. Pokretanje aplikacije

- aplikacija ima kodirane linkove koji su prilagođeni za online puštanje u pogon

- u slučaju želje za lokalnim testiranjem najbolje je prvo otvoriti projekt u IntelliJ IDEA
- Ctrl+Shift+F otvara mogućnost pretrage po svim datotekama i treba tražiti `.onrender.com` dio
- za potrebe lokalnog testiranja promijeniti link u `http://localhost:8080`

## Inicijalna baza podataka

Automatski se pokreće kada se pokrene aplikacija

## Pokretanje aplikacije

- pokretanje `backend\src\main\java\fer\progi\backend\BackendApplication.java`
- otvara se na `localhost:8080`

## 4. Upute za vanjske servise

### 1. Microsoft Entra admin center

- prijava na `https://entra.microsoft.com`
- odabir željene aplikacije, linkova i postavljanje ključeva
- rukovođenje svim korisnicima (njihove uloge i sl.)

### 2. AWS

- registracija i kreiranje *bucketa*
- kopiranje potrebnih ključeva
- `backend\src\main\resources\db\aws-credentials.properties` sadrži zapis ključeva

### 3. Brevo

- registracija računa
- dohvat ključeva

## 5. Ažuriranje aplikacije

- `git pull origin deploy-bigMerge`
- `npm install`
- `npm run build`
- nadalje standardna procedura opisana u prethodnim poglavljima

## 6. Postavljanje na Render

- Prijava na `render.com`
- Povezivanje GitHub računa s Render računom.
- Kreiranje Dockerfilea.
- Postavljanje Maven "slike" u Dockerfileu i instalacija aplikacije.
- Upisivanje uputa za pokretanje aplikacije u Dockerfile.
- Dodavanje Dockerfilea u root GitHub grane.
- Odabir novog Web Servicea na Renderu i odabir određenog repozitorija.
- Odabir naziva aplikacije, grane s koje se pokreće i Docker runtimea.
- Kopiranje ključeva iz `.env` ili ručno dodavanje u `Environment Variables`

- Proces puštanja u pogon.
- Dohvat URL-a na vrhu stranice.
- Automatski je odabran auto-deploy, svakim novim ažuriranjem grane ponovno se pokreće puštanje aplikacije u pogon.

## 7. Korištenje aplikacije

- ovisi o postavljenoj ulozi u **Microsoft Entra admin centru**
- na temelju nje korisnik se prijavljuje putem Microsofta
- ovisno o ulozi prikazuje mu se naslovna stranica i izbornik u kojem su implementirane brojne funkcionalnosti
- u datoteci **keysInfo.txt** nalaze se podaci za prijavu Cilj projekta bio je razviti upotrebljivu aplikaciju od koje bi koristili imali svi učenici i zaposlenici neke škole. Sve potrebne informacije i materijali bili bi dostupni na jednom mjestu što bi olakšalo proces školovanja svima uključenima.

Tim koji je radio na aplikaciji sastojao se od šest članova, a rad na projektu trajao je sedamnaest tjedana, podijeljenih na dva dijela. Tim se odmah podijelio u tri grupe, pri čemu su tri člana bila zadužena za pozadinski (engl. backend), dva člana za prednji (eng. frontend), te jedan član za dokumentaciju.

U prvoj fazi projekta uglavnom se stjecalo potrebno znanje i usavršavale se vještine za rad s odabranim radnim okvirima i knjižnicama. Dogovoreni su funkcijski zahtjevi i željeni izgled same aplikacije.

Završetak prvog dijela obilježilo je postavljanje potrebnih poslužitelja, te na njima uspostava početne baze podataka i početne inačice aplikacije. Prije toga se provelo prikupljanje i analiza zahtjeva, osmišljavanje funkcionalnosti aplikacije i dizajn arhitekture sustava, što je uključivalo odabir radnih okruženja i alata. Glavnina vremena ove etape iskoristila se na detaljan ispis obrazaca uporabe i crtanje dijagrama na temelju kojih bi se kasnije napisao kod.

U drugoj fazi krenulo se dublje u pisanje koda, pazeći na svaki detalj. Po uzoru na odabrano rješenje pisao se nastavak dokumentacije, odnosno njezini dijelovi koji ovaj put služe kako bi opisali funkcionalnost programirane aplikacije.

Razvoj pozadinskog dijela sustava na početku se vršio neovisno od prednjeg. Kad su oba dijela bili zasebno funkcionalni, povezalo ih se u cjelinu nakon čega je krenulo konačno ispitivanje i dorada, kao i intenzivniji rad na dokumentiranju funkcionalnosti napisanog koda.

Komunikacija unutar grupa bila je uspostavljena putem whatsapp i discorda, a prema potrebi se sazvala nekolicina timskih sastanaka, koji su bili relativno učestali tijekom prve faze razvoja, a sukladno s izmijenjenom metodom rada u drugoj fazi puno rjeđi.

Tijekom rada na aplikaciji svi članovi su nailazili na tehničke izazove vezane uglavnom uz neiskustvo u korištenju odabranih tehnologija i alata, zbog čega se značajan dio vremena trošio na njihovo proučavanje i rješavanje grešaka nastalih u kodu. Jedan od problema bilo je i usklađivanje obaveza grupe zbog ostalih fakultetskih i drugih obaveza, te je na trenutke komunikacija izostajala što je otežalo rad na projektu. No vođa tima izradio je odličan plan rada koji se morao pratiti i do krajnjeg roka je sve traženo bilo dovršeno.

Znanja stečena radom na projektu u korištenju radnih okvira za razvoj poslužiteljske strane aplikacije i dizajn korisničkog sučelja, metodama i standardima implementiranja njihove komunikacije, pa čak i radom s alatima za pisanje valjane dokumentacije, kao i samo iskustvo rada u timu pokazat će se korisnima u budućim projektima ili nastavku razvijanja ovog.

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen wiki predložak	L.Batinić	22.10.2024.
0.2	Nabrojani funkcionalni zahtjevi	L.Batinić	22.10.2024.
0.3	Definirani obrasci uporabe	L.Batinić	22.10.2024.
0.3.1	Opisani obrasci uporabe	L.Batinić	22.10.2024.
0.4	Dodani aktori	L.Batinić	22.10.2024.
0.5	Dodane stranice na wiki	L.Batinić	23.10.2024.
0.6	Dodan mali dio opisa projekta	L.Batinić	29.10.2024.
0.7	Napravljene male promjene u README.md i Home na wiki	L.Batinić	29.10.2024.
0.8	Dodana stranica 2., dionici i ostali zahtjevi	L.Batinić	31.10.2024.
0.9	Uređena stranica 4., obrasci uporabe	L.Batinić	31.10.2024.
0.10	Dodani funkcionalni i nefunkcionalni zahtjevi u tablice	L.Batinić	3.11.2024.
0.11	Dodana 2 sekvencijska dijagrama	L.Batinić	6.11.2024.
0.12	Dodan kratki opis arhitekture	L.Batinić	6.11.2024.
0.13	Dodan opis tablica	L.Batinić	7.11.2024.
0.14	Manje jezične promjene na 2. stranici	L.Batinić	8.11.2024.
0.15	Izmjena rasporeda 4. stranice	L.Batinić	8.11.2024.
0.16	Dodan dijagram baze podataka	L.Batinić	8.11.2024.
0.17	Dodan opis projektnog zadatka	L.Batinić	11.11.2024.
0.18	Manje promjene u tekstu na 5.stranici	L.Batinić	12.11.2024.
0.19	Dodan učenik kao aktor i dionik	L.Batinić	12.11.2024.
0.20	Promjenjen dijagram glavnih funkcionalnosti	L.Batinić	12.11.2024.
0.21	Dodan dijagram razreda	L.Batinić	12.11.2024.
0.22	Maknut s popisa član koji je odustao, dodan 4. sastanak	L.Batinić	13.11.2024.
0.23	Manje jezične promjene, uređene stranice prema predlošku	L.Batinić	15.11.2024.
0.24	Dodani promijenjeni dijagrami	L.Batinić	15.11.2024.
1.1	Promijenjen dijagram razreda	L.Batinić	10.12.2024.
1.2	Uređen wiki po novom predlošku	L.Batinić	28.12.2024.
1.3	Prilagođeni funkcijski zahtjevi radu aplikacije	L.Batinić	12.1.2025.
1.4	Prilagođeni obrasci uporabe radu aplikacije	L.Batinić	14.1.2025.
1.5	Dodani promijenjeni dijagrami	L.Batinić	17.1.2025.

Rev.	Opis promjene/dodatka	Autori	Datum
1.6	Promjenjen dijagram baze i izgled 4. stranice	L.Batinić	19.1.2025.
1.7	Dodan zaključak	L.Batinić	19.1.2025.
1.8	Promjenjen dijagram baze i tablice baze	L.Batinić	22.1.2025.
1.9	Dodani dijagrami stanja i aktivnosti	L.Batinić	23.1.2025.
1.10	Napisane upute za puštanje u pogon	B.Odobašić	24.1.2025.
1.11	Opisana ispitivanja programskog rješenja	B.Odobašić	24.1.2025.
1.12	Dodani dijagrami komponenata i razmještaja	L.Batinić	24.1.2025.
1.13	Manje jezične promjene	L.Batinić	24.1.2025.
1.14	Opisani dijagrami, dodane tehnologije i uređen plan rada	L.Batinić	24.1.2025.
1.15	Dodan dijagram pregleda promjena	L.Batinic	24.1.2025.

1. Programsko inženjerstvo, FER ZEMRIS, <http://www.fer.hr/predmet/proinz>
2. I. Sommerville, "Software engineering", 8th ed, Addison Wesley, 2007.
3. T.C.Lethbridge, R.Langanieri, "Object-Oriented Software Engineering", 2nd ed. McGraw-Hill, 2005.
4. I. Marsic, Software engineering book", Department of Electrical and Computer Engineering, Rutgers University, <http://www.ece.rutgers.edu/~marsic/books/SE>
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>

## Dnevnik sastajanja

---

### 1. sastanak

- Datum: 17. listopada 2024.
- Prisustvovali: L.Batinić, M.Bilješko, B.Odobašić, D.Rožić, M.Stolar, K.Vrdoljak
- Teme sastanka:
  - upoznavanje tima
  - dogovor načina komunikacije (Discord server, WhatsApp grupa)

### 2. sastanak

- Datum: 21. listopada 2024.
- Prisustvovali: L.Batinić, M.Bilješko, B.Odobašić, M.Stolar, K.Vrdoljak
- Teme sastanka:
  - okvirno pisanje funkcionalnih zahtjeva i obrazaca uporabe i dogovor da se kasnije postave na wiki
  - razmjena pitanja koje će grupa postaviti na laboratorijskim vježbama

### 3. sastanak

- Datum: 8. studenoga 2024.



- Prisustvovali: B.Odobašić, D.Rožić, M.Stolar
- Teme sastanka:
  - dogovor oko izgleda početne stranice i stranice za prijavu
  - odabrana je funkcionalnost koju ćemo imati za prvi termin predaje projekta
  - započeli smo rad na Spring-u i inicijalnu bazu podataka (glavne entitete, atribute, ključeve i osnovne metode)

#### 4. sastanak

- Datum: 12. studenoga 2024.
- Prisustvovali: L.Batinić, M.Bilješko, B.Odobašić, D.Rožić, M.Stolar, K.Vrdoljak
- Teme sastanka:
  - završni dogovori oko 1.predaje projekta, spajanju front-enda i back-enda

## Plan rada

---

### Prvi dio

Dogovor oko funkcijskih zahtjeva i učenje novih tehnologija uzelo je veći dio ciklusa. Zadnja dva tjedna dogovor je bio da se radi baza i početne funkcionalnosti.

### Drugi dio

Vođa tima napravio je raspored rada po tjednima:

#### 1. tjedan 16.12.-22.12.

- logout handler
- gore desno - ime i prezime osobe izvučeno iz baze
- podaci o vremenu na /home, /ucenik...
- definiranje točne lokacije škole
- admin može dodavati učenike viših razreda
- admin može brisati bilo što iz baze (ako su podaci neupotrebljivi)
- admin nastavniku dodaje predmete (ili uklanja)
- učenik može dodavati aktivnosti (prije početka 2025/26 godine)
- omogućiti učeniku ispis potvrde na mail (ravnatelj i djelatnik mogu vidjeti izvještaj)

#### 2. tjedan 23.12.-29.12.

- pregled upisanih predmeta i aktivnosti učenika
- pregled zauzeća učionica
- dodavanje materijala - nastavnik
- praćenje pristupa materijalima
- slanje mailova
- posebna obavijest u vezi terenske nastave koja ima ugrađenu rutu od škole do tog mjesta

#### 3. tjedan 30.12.-5.1.

- dovršetak generiranja rasporeda sati
- omogućavanje komunikacije (privatni razgovori, grupe)

- nastavnik bira sate kad ne može predavati
- pregled kalendara rasporeda
- bug fixes
- novi zahtjevi

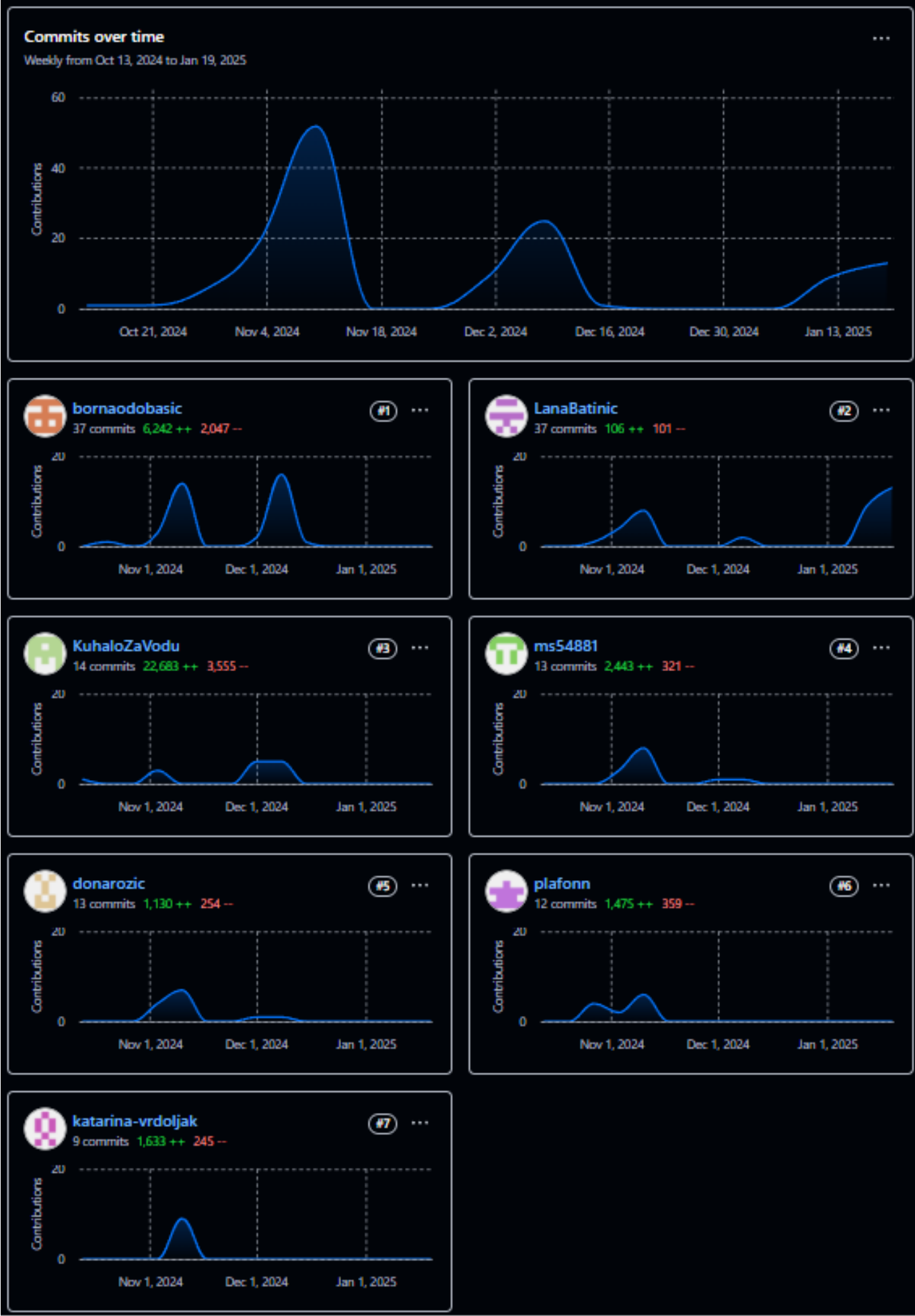
Nakon toga rad na zaostacima i greškama.

## Tablica aktivnosti

	Batinić Lana	Bilješko Marija	Odobasić Borna	Rožić Donatela	Stolar Maja	Vrdoljak Katarina
Upravljanje projektom						
Opis projektnog zadatka	4					
Funkcionalni zahtjevi	2		1			
Opis pojedinih obrazaca	7					
Dijagram obrazaca	6					
Sekvencijski dijagrami	5					
Opis ostalih zahtjeva	0.5					
Arhitektura i dizajn sustava	5		3			
Baza podataka			11	2	2	
Dijagram razreda	1					
Dijagram stanja	1.5					
Dijagram aktivnosti	2					
Dijagram komponenti	1				0.5	
Korištene tehnologije i alati	0.5		2			
Ispitivanje programskog rješenja		3	15	5	5	2
Dijagram razmještaja	1					
Upute za puštanje u pogon			3			
Dnevnik sastajanja	0.5					
Zaključak i budući rad	1.5					

	Batinić Lana	Bilješko Marija	Odobasić Borna	Rožić Donatela	Stolar Maja	Vrdoljak Katarina
Popis literature						
Učenje novih tehnologija	1	10	20	11	11	10
izrada korisničkog sučelja		20	1		1	15
spajanje s bazom podataka		3	5	3	4	4
Izrada backenda			70	52	50	
Izrada frontenda		45	4		5	40
Postavljanje na server			6			

## Dijagram pregleda promjena



Slika B.1: Dijagram pregleda promjena

# Ključni izazovi i rješenja

---

Izazova je bilo tijekom cijele izrade projekta. Prvi izazov bilo je neznanje. Učenje novih tehnologija bio je prvi zadatak koji su svi članovi tima trebali riješiti. Jedan od izazova je i manjak direktne komunikacije i usklađivanje svačijih obaveza sa zajedničkim radom na projektu. Radilo se na puno grana, pa je proces spajanja svega trajao nepotrebno dugo.

## Programsko inženjerstvo ak.god. 2024./2025.

---

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

EduHub

---

**Grupa: <TG09.3>**

**Tim:**

Batinić Lana

Bilješko Marija

Odobasić Borna

Rožić Donatela

Stolar Maja

Vrdoljak Katarina

**Nastavnik: Vlado Sruk**

EduHub