

Programsko inženjerstvo ak.god 2024./2025

Sveučilište u Zagrebu

Fakultet elektrotehnike i računarstva

Noodle

Tim: <TG09.2>

Ime tima: Noodle

Nastavnik: Vlado Sruk

Aplikacija Noodle je digitalno rješenje koje povezuje učenike i školsko osoblje s ciljem unaprjeđenja organizacije školskih aktivnosti i komunikacije unutar srednje škole.

Potencijalna korist ovog projekta

U današnje vrijeme, većina sustava pokušava digitalizirati i pojednostaviti korištenje korisnicima. Platforma Noodle na sličan način donosi brojne koristi za obrazovni sustav i njegove sudionike. Aplikacija je usmjerena na poboljšanje organizacije i komunikacije u srednjoj školi te na centraliziranje informacija i jednostavan pristup za sve korisnike. Cilj aplikacije je olakšati praćenje rasporeda, obaveza, aktivnosti i dostupnih resursa za učenike i nastavnike te podržati međusobnu komunikaciju i suradnju. Učenici, koji se prilikom upisa opredjeljuju za određeni smjer, dobivaju predmete prilagođene svom smjeru u skladu s kurikulumom, čime se osigurava usklađenost s obrazovnim standardima. Aplikacija automatski generira raspored sati, uključujući organizaciju praktične nastave u specijaliziranim prostorima te odlazak na prakse, uz automatsko obavješćavanje svih uključenih osoba o promjenama putem elektroničke pošte. Omogućene su i dodatne funkcionalnosti, poput slanja poruka između korisnika, objava nastavnih materijala isključivo učenicima kojima su namijenjeni te praćenje statistike pristupa materijalima. Učenici također imaju mogućnosti izvaditi potvrdu o upisu za potrebe izrade pokaza za javni prijevoz. Aplikacija omogućuje i korištenje kartografskih usluga u sklopu podrške za terensku nastavu i/ili prakse te prikaz vremenske prognoze za lokaciju na kojoj se škola nalazi. Kako bi platforma bila široko dostupna, nudi responzivni dizajn prilagođen različitim uređajima i podršku za autentifikaciju putem vanjskih servisa poput OAuth2, čime se osigurava zaštita podataka za korisnike. Platforma doprinosi učinkovitosti i organizaciji nastave u školama te omogućava jednostavniju komunikaciju korisnika i dijeljenje materijala na svima razumljiv i dostupan način.

Postojeća slična rješenja

Na internetu već postoje rješenja koja imaju slične funkcionalnosti kao Noodle te koja se koriste u obrazovnim sustavima. Jedan od takvih primjera je Google Classroom, koji omogućuje nastavnicima kreiranje digitalnih razreda, dijeljenje zadataka i materijala te praćenje napretka učenja. Koristeći pomoćne Google alate, poput Google Meeta ili Google Calendara, vrlo je lako ostvarena komunikacija između sudionika i efikasna organizacija vremena. Postoji i aplikacija Moodle, koja služi za dijeljenje materijala i e-učenje putem interaktivnih kvizova. Sličnosti vidimo i na primjeru FER-ovog Intraneta koji omogućuje pristup materijalima za sve predmete koje student sluša, kao i efikasno praćenje rasporeda predavanja i vježbi unutar akademske godine. Razlika u odnosu na navedene primjere je da platforma Noodle nudi integriranu podršku za cjelokupno upravljanje školskim resursima, od optimalnog slaganja rasporeda, gdje se u obzir uzimaju predmeti, dostupnost nastavnika i učionica te mogućnost korištenja opreme, do objave materijala i generiranja potvrda za učenike. Aplikacija veći fokus stavlja na općenitu organizaciju obrazovnog sustava i efikasniju komunikaciju između korisnika, nego na aspekte e-učenja, koje navedeni primjeri implementiraju. Također, aplikacija nudi mogućnost korištenja Google Mapsa kako bi učenicima olakšali odlazak na praksu. Osim toga, omogućena je autentifikacija putem OAuth2 servisa, što nudi veću fleksibilnost i sigurnost pri prijavi korisnika, što može biti poželjno kad su u pitanju učenici srednje škole. Poput i većine aplikacija danas, omogućen je responzivni dizajn kako bi korisnici mogli na bilo kojem uređaju pristupiti platformi, čime se osigurava češće i jednostavnije korištenje.

FER-Moodle

Naslovnica

Moja naslovnica

Moji e-kolegiji

FER - Moodle

Site news

Pretplatite se na ovaj forum

Arhiva Moodle za prethodne godine

napisao/la Admin User

- petak, 17. veljače 2012., 18:06

Arhiva za prethodnu akademsku godinu dostupna je na zahtjev.

Trajna poveznica

Raspravljajte o ovoj temi (0 odgovora za sada)

Popis e-kolegija

Ostali kolegiji (12)

FER-ovi nastavni kolegiji (401)

Proširi sve

Pretraži e-kolegije

Dobrodošli na FER-ov poslužitelj Moodle!

Kalendar

studenoga


2024.


Ned	Pon	Uto	Sri	Čet	Pet	Sub
					1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30


Cijeli kalendar


• Uvezite ili izvezite kalendare


1.1 Prikaz platforme Moodle





 Classroom


 Home


 Calendar

 Archived classes


 Settings






 J

Don't see your classes?
Try another account



Add a class to get started

[Create class](#)[Join class](#)



1.2 Prikaz platforme Google Classroom

Skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje

Rješenje poput Noodlea bilo bi zanimljivo širokom spektru korisnika u obrazovnim institucijama. Krenuvši od samih učenika, kao krajnji korisnici najviše bi profitirali od lakšeg pristupa rasporedu, materijalima i obavijestima. Sama komunikacija bila bi olakšana i pristup osoblju bio bi jednostavniji. Nastavnici bi koristili aplikaciju za organizaciju nastave, upravljanje materijalima te komunikaciju s učenicima. Omogućeno je planiranje rasporeda i uvažavanje njihovih slobodnih termina. Obrazovne institucije, u ovom slučaju srednje škole, zainteresirane su za digitalizaciju i poboljšanje učinkovitosti svojih obrazovnih procesa.

Administrativno osebje i ravnatelj imali bi koristi od centraliziranog upravljanja resursima i školskom logistikom. Aplikacija nudi detaljno praćenje zauzetosti prostorija i specijalizirane opreme, automatsko generiranje potvrda o upisu te izvještaje o izdanim dokumentima, što doprinosi smanjenju administrativnog

opterećenja i boljoj organizaciji. Satničari i administrator bi imali bolji uvid i kontrolu nad složenim rasporedom. Satničaru se smanjuje opseg posla jer samo prati promjene u rasporedu, a uloga administratora dodjeljuje se ravnatelju škole kako bi mogao imati pristup i kontrolu nad svim potrebnim podacima. Aplikacija predstavlja univerzalno rješenje koje je lako primjenjivo za različite potrebe unutar same srednje škole, ali i unutar cijelog sustava srednjih škola koje žele digitalizirati svoj sustav.

Mogućnost prilagodbe rješenja

Moguće prilagodbe za aplikaciju:

- fleksibilno dodavanje korisničkih uloga (npr. mentori, studentska služba, roditelji)
- prilagodba funkcionalnosti prema specifičnim potrebama korisnika (sustav funkcionira za različite škole)
- mogućnost proširenja baze podataka (npr. uvođenje upisivanja ocjena, praćenje izostanaka)
- skalabilnost za veći broj korisnika i podataka (sustav je skalabilan, podržava rast škola i implementaciju za povećan broj osoblja i predmeta)
- sustav je prilagođen za izvođenje na različitim uređajima (kako bi se omogućili korisnicima pristup gdje god se nalazili)
- mogućnost integracije s drugim obrazovnim alatima po potrebi (npr. Zoom, Notebook LM)
- mogućnost promjene dizajna/korisničkog sučelja da odgovara zahtjevima/izgledu škole

Opseg projektnog zadatka

Projekt uključuje izradu aplikacije koja omogućava slaganje rasporeda, praćenje nastavnih aktivnosti i školskih resursa, kao što su specijalizirane prostorije i oprema te olakšava školama organizaciju nastave u skladu s potrebama različitih smjerova. Prvi korak za nove korisnike je registracija. Neregistrirani korisnik može se putem vanjskog servisa za autentifikaciju (OAuth2) ulogirati u sustav te mu se odmah na početku dodjeljuje uloga. Moguće uloge su: učenik, nastavnik, ravnatelj(u ulozi administratora) i satničar. Uzevši u obzir da se radi o školskom sustavu, nije moguće dozvoliti pristup bilo kojem korisniku, već postoji lista čeka (popis imena i prezimena dozvoljenog osoblja) uz pomoć koje ravnatelj dodjeljuje ulogu korisniku. Ravnatelju se dodjeljuje uloga administratora prije pokretanja stranice kako bi mogao voditi evidenciju za školu. Osobi koja sudjeluje u slaganju rasporeda se također dodjeljuje uloga (satničar) i daje mu se ovlast za pristup podacima koji su potrebni za slaganje rasporeda kako bi mogao raditi izmjene i slati obavijesti. Ovisno o dodijeljenoj ulozi, slijede koraci nakon registracije za postavljanje profila na stranici. U slučaju uloge učenika, bira se smjer (tehničar za računalstvo ili tehničar za elektroniku) kako bi se mogli odrediti obavezni i izborni predmeti u skladu s kurikulumom. U slučaju uloge nastavnika, odabiru se predmeti koje nastavnik smije predavati i dostupnost nastavnika tijekom tjedna. Nakon toga slaže se raspored za učenike i nastavnike te se učenici raspodijele u razrede, ovisno o upisanom smjeru. Svakom razredu dodjeljuje se razrednik. U oba slučaja se nakon generiranja rasporeda na elektroničku poštu šalje obavijest korisnicima o rasporedu. Na isti način nastavnici i učenici primaju obavijesti ako dođe do promjena u rasporedu. Nastavnik ima mogućnosti objave materijala u repozitorij za predmete koje predaje, a njihova dostupnost ograničena je na učenike koji slušaju predmet. Također, prati se statistika pristupa i korištenja materijala kako bi nastavnici imali povratnu informaciju. Učenici još putem elektroničke pošte mogu primiti potvrdu o upisu ako to zatraže. Sva ostala komunikacija odvija se unutar grupa za slanje poruka na samoj stranici, koje su automatski generirane po razredima. Unutar grupa razrednici mogu slati obavijesti o predmetima ili materijalima. Nastavnici također mogu od satničara zatražiti da obavi izmjenu u rasporedu po potrebi. Osim navedenih osnovnih funkcionalnosti, postoje još dvije dodane funkcionalnosti stranice. Prvo je

obavješćavanje učenika o praksama i/ili terenskoj nastavi korištenjem usluga Google Mapsa. Prakse su dio rasporeda te se tjedan dana prije na elektroničku poštu učenicima šalje obavijest o lokaciji i kako doći do nje od škole. Zadnja funkcionalnost platforme je prikaz vremenske prognoze za mjesto u kojem se škola nalazi na naslovnoj stranici. Sama stranica implementira responzivni dizajn kako bi se omogućio pristup putem različitih uređaja i time olakšalo korištenje korisnicima.

Moguće nadogradnje projektnog zadatka

Moguće nadogradnje za platformu Noodle:

- proširenje pristupa aplikacije roditeljima kako bi mogli dobivati obavijesti o djetetu i direktno komunicirati s nastavnicima
- mogućnost offline rada kako bi učenici mogli koristiti aplikaciju bez pristupa internetu
- uvođenje mogućnosti videopoziva u svrhu hibridne nastave ili online konzultacija
- integracija s dodatnim obrazovnim alatima poput Kahoota ili Quizleta
- implementacija prikaza ocjena i povratnih komentara nastavnika kako bi učenici imali uvid u svoj napredak
- uvođenje alata za pristupačnost za učenike s poteškoćama u razvoju, npr. čitači ekrana, prilagodljivi font

Funkcionalni zahtjevi

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-001	Sustav omogućuje registraciju neregistriranih korisnika putem sustava za autentifikaciju OAuth2.0	Visok	Zahtjev dionika	Korisnik može napraviti račun putem OAuth2.0 servisa (Google ili Microsoft račun) te primiti potvrdu o registraciji na e-mail
F-002	Sustav omogućuje prijavu korisnika u sustav	Visok	Zahtjev dionika	Korisnik unosi svoje podatke i pristupa svom profilu na stranici
F-003	Sustav omogućuje stvaranje razrednih grupa	Visok	Zahtjev dionika	Učenici se raspodjeljuju u razredne grupe nasumično ovisno o upisanom smjeru te se svakoj grupi nasumično dodjeljuje razredni nastavnik
F-004	Sustav generira raspored sati prema smjerovima i razredima	Visok	Dokument zahtjeva	Raspored sati se generira s obzirom na kapacitet prostorija, dostupnost opreme i nastavnika te u skladu s kurikulumom
F-005	Sustav šalje obavijesti o izradi rasporeda	Visok	Povratne informacije	Obavijesti se šalju svim korisnicima unutar 5 minuta nakon izrade rasporeda
F-006	Sustav omogućuje pravovremeno obavještavanje korisnika o aktivnostima i promjenama u sustavu	Visok	Zahtjev dionika	Sustav šalje obavijesti putem elektroničke pošte svim korisnicima koji bi imali korist od primanja obavijesti
F-007	Sustav omogućuje vađenje potvrde o upisu	Srednji	Zahtjev učenika	Učenici na zahtjev primaju potvrdu o upisu elektroničkom poštom
F-008	Sustav omogućuje upućivanje učenika na stručnu praksu	Srednji	Dokument zahtjeva	Učenici imaju pristup karti s podacima o lokaciji i načinu dolaska na stručnu praksu
F-009	Sustav omogućuje direktnu komunikaciju između korisnika	Visok	Postojeći sustav	Razrednik i učenici mogu slati i primiti poruke u chat unutar razrednih grupa

ID zahtjeva	Opis	Prioritet	Izvor	Kriterij prihvaćanja
F-010	Sustav omogućuje nastavnicima objavu materijala	Visok	Dokument zahtjeva	Nastavnici mogu objaviti materijale (prezentacije, PDF dokumenti, video itd.) na repozitorij sustava
F-011	Sustav omogućuje učenicima pristup materijalima	Visok	Zahtjev dionika	Učenici mogu pristupiti samo materijalima iz predmeta koji su dio njihovog rasporeda, mogu pregledavati materijale ili ih skinuti na vlastito računalo
F-012	Sustav prati statistiku pristupa materijalima	Srednji	Povratne informacije	Nastavnici mogu vidjeti broj pregleda i preuzimanja za pojedine materijale koje objavljuju
F-013	Sustav omogućuje praćenje izostanaka učenika	Srednji	Zahtjev dionika	Nastavnici mogu unositi izostanke učenika za pojedine dane u tjednu koje su propustili
F-014	Sustav omogućava promjene rasporeda	Visok	Zahtjev dionika	Nastavnici mogu zatražiti od satničara da provede promjenu u rasporedu, (obavijest o promjeni rasporeda šalje se na e-mail)

Ostali zahtjevi

Nefunkcionalni zahtjevi

Zahtjevi performansi

ID zahtjeva	Opis	Prioritet
NF-1.1	Sustav treba podržavati rad više korisnika u stvarnom vremenu	Visok
NF-1.2	Sustav treba biti dizajniran tako da može lako proširiti funkcionalnosti i podržati veći broj korisnika bez smanjena performansi	Srednji
NF-1.3	Sustav treba dopustiti korisnicima upravljanje aplikacijom s odgovorima na akcije unutar 3 sekunde	Visok

Zahtjevi pouzdanosti

ID zahtjeva	Opis	Prioritet
NF-2.1	Sustav treba biti dostupan 99 % vremena	Visok
NF-2.1.1	Sustav mora imati omogućen pristup iz javne mreže	Srednji
NF-2.2	Sustav treba biti otporan na neispravno korištenje korisničkog sučelja i ne smije dovesti do pada sustava	Srednji

Sigurnosni zahtjevi

ID zahtjeva	Opis	Prioritet
NF-3.1	Sustav treba implementirati autentifikaciju putem OAuth2.0, omogućujući autentifikaciju putem Google ili Microsoft računa	Visok
NF-3.2	Sustav mora ograničiti korisnika na pristup jedino onim resursima kojima ima pristup koristeći role-based access control(RBAC), odnosno u ovisnosti s ulogama korisnika	Visok
NF-3.3	Sustav mora koristiti HTTPS protokol za sigurnu komunikaciju između klijenta i poslužitelja	Visok

Zahtjevi za korisničko iskustvo

ID zahtjeva	Opis	Prioritet
NF-4.1	Korisničko sučelje treba biti jednostavno i intuitivno za korištenje, s jasnim navigacijama i vizualnim elementima	Visok
NF-4.1.1	Sustav treba omogućiti podršku za hrvatski jezik i jasno prikazivanje informacija za sve korisnike	Srednji
NF-4.1.2	Sustav treba implementirati sučelje koje je responzivno i optimizirano za različite uređaje (pametni telefoni, tableti, desktop uređaji)	Visok
NF-4.2	Sustav treba koristiti Google Maps prilikom upućivanja učenika na praksu	Visok
NF-4.3	Sustav treba koristiti OpenWeatherMap za prikaz prognoze na naslovnoj stranici koja se ažurira svakih 30 minuta	Srednji

Zahtjevi za održavanje

ID zahtjeva	Opis	Prioritet
NF-5.1	Sustav treba biti dizajniran tako da se može lako održavati i nadograditi, s jasno definiranim modulima i mogućnošću jednostavne implementacije novih funkcionalnosti	Visok

ID zahtjeva	Opis	Prioritet
NF-5.1.1	Sustav treba imati dovoljnu dokumentaciju	Visok
NF-5.1.1.1	Sustav treba biti opisan putem dokumenta oblikovanja /SRS/	Visok
NF-5.1.2	Sustav treba implementirati koristeći objektno-orijentirane jezike	Visok

Zahtjevi domene

ID zahtjeva	Opis	Prioritet
D-001	Sustav mora osigurati da je raspored za svaki smjer u skladu s informacijama objavljenim na službenim stranicama Ministarstva znanosti, obrazovanja i sporta	Visok
D-002	Sustav mora omogućiti prikaz rasporeda po tjednima u tabličnom obliku za akademsku godinu, uzevši u obzir blagdane, neradne dane i vikende	Visok
D-003	Sustav treba biti usklađen s pravilima zaštite podataka, uključujući GDPR ili slične regulative	Visok
D-004	Sustav treba omogućiti pohranu podataka o učenicima i nastavnicima u skladu s obrazovnim pravilnicima i zakonima o zaštiti podataka	Visok
D-005	Sustav mora omogućiti upravljanje materijalima tako da svi materijali za nastavu budu organizirani prema predmetima i godinama	Srednji
D-006	Sustav mora omogućiti ravnatelju škole da prati podatke o učenicima i nastavnicima	Visok

Dionici

Dionici:

- 1. Razvojni tim
- 2. Administrator sustava
- 3. Korisnici

Aktori:

A-1 Neregistrirani korisnik može:

- 1. Napraviti registraciju (F-001)

A-2 Registrirani korisnik (ravnatelj, satničar, nastavnik ili učenik) može:

- 1. Prijaviti se u sustav (F-002)
- 2. Pregledavati raspored sati (F-004)

3. Primati obavijesti vezano uz školske aktivnosti, raspored i sl. (F-006)

A-3 Učenik može:

1. Odabrati smjer (samo prilikom prve prijave u sustav)
2. Primati obavijesti vezane uz praksu (F-008)
3. Komunicirati u razrednoj grupi (F-003, F-009)
4. Pristupati materijalima (F-011)
5. Zatražiti potvrdu o upisu (F-007)

A-4 Nastavnik može:

1. Komunicirati unutar grupe (F-003, F-009, samo ako je nastavnik ujedno i razrednik)
2. Objavljivati materijale (F-010)
3. Pratiti statistiku korištenja materijala (F-012)
4. Zapisivati i pregledavati izostanke učenika (F-013)
5. Tražiti promjene u rasporedu (F-014)

A-5 Satničar može:

1. Mijenjati raspored sati (F-014)
2. Komunicirati s ravnateljom ili nastavnicima o promjenama u rasporedu

A-6 Ravnatelj (u ulozi administratora) može:

1. Odobriti ili odbiti zahtjev za registraciju (F-001)
2. Dodijeliti ulogu registriranom korisniku (F-001)
3. Nadgledati sigurnost sustava
4. Pregledavati statistiku korištenja materijala i izostanaka učenika (F-012, F-013)

Obrasci uporabe

1. Visokorazinski dijagram obrazaca uporabe cijelog sustava

UC1 - Registracija novih korisnika

UC2 - Prijava korisnika u sustav

UC6 - Generiranje rasporeda sati

UC7 - Objava materijala

UC8 - Pristupanje materijalima

UC10 - Upute za odlazak na praksu

2. Dijagram obrazaca uporabe za ključne funkcionalnosti

UC5 - Generiranje razrednih grupa

UC9 - Izdavanje potvrda o upisu

UC11 - Promjene u rasporedu

UC12 - Unos izostanaka

3. Dijagram obrazaca uporabe za korisničke role

UC3 - Biranje smjera

UC4 - Biranje predmeta i označavanje dostupnosti

UC7 - Objava materijala

UC8 - Pristupanje materijalima

UC11 - Promjene u rasporedu

UC12 - Unos izostanaka

4. Dijagram obrazaca uporabe za osnovne poslovne procese

UC5 - Generiranje razrednih grupa

UC6 - Generiranje rasporeda sati

UC9 - Izdavanje potvrda o upisu

5. Dijagram obrazaca uporabe za kritične sustave i integracije

UC10 - Upute za odlazak na praksu

Opis obrazaca uporabe

UC1 - Registracija novih korisnika

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Omogućiti novom korisniku da se registrira u sustav
- Sudionici: Baza podataka, administrator, OAuth2.0
- Preduvjet: Korisnik mora biti učenik ili djelatnik škole
- Opis osnovnog tijeka:
 1. Korisnik pristupa stranici za registraciju i odabire jedna od računa (Google ili Microsoft) za registraciju u sustav. (F-001)
 2. Korisnik šalje zahtjev za registraciju.
 3. Administrator prima zahtjev i pregledava podatke.
 4. Administrator potvrđuje registraciju.
 5. Korisnik dobiva potvrdu putem e-maila da je uspješno registriran.
- Opis mogućih odstupanja:
 - 3.1 Korisnik ne zadovoljava uvjete za registraciju, ne nalazi se na listi ovlaštenih.
Rješenje: Administrator odbija zahtjev za registraciju.
 - 3.2 Korisnik je već registriran u sustavu.
Rješenje: Prikazati korisniku poruku da je već obavio registraciju i sugerirati prijavu u sustav.
- 5. Potvrda nije poslana.
Rješenje: Korisnik može zatražiti ponovno slanje potvrde.

UC2 - Prijava korisnika u sustav

- Glavni sudionik: Korisnik
- Cilj: Uspješno prijaviti korisnika u sustav
- Sudionici: Baza podataka
- Preduvjet: Korisnik je uspješno obavio registraciju
- Opis osnovnog tijeka:
 1. Korisnik pristupa stranici za prijavu u sustav.
 2. Korisnik se prijavljuje putem Google ili Microsoft računa.
 3. Sustav provodi validaciju unesenih podataka s podacima u bazi podataka. (F-002)
 4. Korisnik se prijavljuje i pristupa naslovnoj stranici.
- Opis mogućih odstupanja:
 3. Autentifikacija putem OAuth2.0 sustava nije uspjela
Rješenje: Prikazuje se obavijest o pogrešnim podacima i korisniku se omogućava ponovna prijava.

UC3 - Biranje smjera

- Glavni sudionik: Učenik
- Cilj: Omogućiti učeniku odabir smjera

- Sudionici: Baza podataka
- Preduvjet: Korisniku je dodijeljena uloga učenika u sustavu
- Opis osnovnog tijeka:
 1. Učenik pristupa stranici za odabir smjera, koja se prikazuje samo nakon 1. prijave u sustav, nakon uspješne registracije.
 2. Sustav prikazuje izbor smjerova (tehničar za računalstvo ili tehničar za elektroniku).
 3. Učenik odabire željeni smjer.
 4. Sustav pohranjuje odabrani smjer u bazu podatka.
 5. Učenik je usmjeren na naslovnu stranicu.
- Opis mogućih odstupanja:
 3. Učenik nije odabrao smjer

Rješenje: Sustav onemogućuje daljnje aktivnosti dok učenik ne odabere barem jedan smjer.

UC4 - Biranje predmeta i označavanje dostupnosti

- Glavni sudionik: Nastavnik
- Cilj: Omogućiti nastavniku odabir predmeta koje predaje i dostupnost za rad
- Sudionici: Baza podataka
- Preduvjet: Korisniku je dodijeljena uloga nastavnika u sustavu
- Opis osnovnog tijeka:
 1. Nastavnik pristupa stranici za odabir, koja se prikazuje samo nakon 1. prijave u sustav, nakon uspješne registracije.
 2. Nastavnik prvo odabire predmete koje može predavati.
 3. Nastavnik zatim označava svoju dostupnost u kalendaru koji prikazuje tjedni raspored (pon-pet, 8-16h).
 4. Nakon završetka odabira, sustav pohranjuje te podatke u bazu podataka.
 5. Nastavnik je preusmjeren na naslovnu stranicu.
- Opis mogućih odstupanja:
 2. Nastavnik nije označio nijedan predmet.

Rješenje: Sustav onemogućuje daljnje aktivnosti dok nastavnik ne odabere barem jedan termin u kojem može predavati.

 3. Nastavnik nije označio nijedan termin u kalendaru.

Rješenje: Sustav onemogućuje daljnje aktivnosti dok nastavnik ne odabere barem jedan termin u kojem može predavati.

UC5 - Generiranje razrednih grupa

- Glavni sudionik: Sustav
- Cilj: Automatski generirati razredne grupe za učenike prema odabranom smjeru i svakoj dodijeliti razrednika
- Sudionici: Baza podataka
- Preduvjet: Sustav ima podatke o svim učenicima i njihovim odabranim smjerovima.
- Opis osnovnog tijeka:

1. Sustav pristupa podacima o učenicima koji su smjer odabrali (tehničar za računalstvo ili tehničar za elektroniku).
2. Sustav nasumično dijeli učenike s obzirom na odabrani smjer, tako da u jednom razredu bude maksimalno 25 učenika.
3. Sustav zatim svakoj grupi nasumično dodjeljuje jednog nastavnika koji postaje razrednik grupe.
4. Grupa i razrednik se pohranjuju u bazu podataka. (F-003)
5. Učenici i nastavnici na e-mail dobivaju obavijest o dodjeli grupe i razrednika. (F-006)
6. Nakon formiranja grupa, sustav automatski stvara privatni chat za svaku grupu, u koji su dodani svi članovi grupe i unutar kojeg je omogućena komunikacija između članova. (F-009)
7. Gotove grupe se prikazuju na korisničkim stranicama za učenike i nastavnike, uključujući chat prostor.

- Opis mogućih odstupanja:

7. Korisnicima se dodijeljena grupa ne prikazuje nigdje na stranici.

Rješenje: Administrator ručno dodaje korisnika u slobodnu grupu.

UC6 - Generiranje rasporeda sati

- Glavni sudionik: Sustav
- Cilj: Automatski generirati raspored sati po smjerovima
- Sudionici: Baza podataka
- Preduvjet: Svi relevantni podaci (dostupnost nastavnika i učionica, smjerovi svih učenika) su pohranjeni i dostupni u sustavu, stvorene su razredne grupe
- Opis osnovnog tijeka:
 1. Sustav koristi algoritam za generiranje rasporeda na temelju podataka u bazi.
 2. Algoritam u obzir uzima: dostupnost nastavnika, dostupnost i broj učionica, razredne grupe, odabrane smjerove i kurikulum, radne i neradne dane u godini.
 3. Sustav automatski generira raspored za učenike i nastavnike, po razrednim grupama. (F-004)
 4. Raspored se pohranjuje u bazu podataka.
 5. Sustav generira obavijest i šalje učenicima, nastavnicima i satničaru na e-mail adresu. (F-005)
 6. Gotov raspored se prikazuje na stranici.

- Opis mogućih odstupanja:

3. Algoritam ne uspijeva generirati raspored.

Rješenje: Sustav prikazuje obavijest administratoru o grešci u generiranju rasporeda.

UC7 - Objava materijala

- Glavni sudionik: Nastavnik
- Cilj: Omogućiti nastavniku da objavi materijale za predmete na stranicu
- Sudionici: Baza podataka
- Preduvjet: Korisnik ima ulogu nastavnika
- Opis osnovnog tijeka:
 1. Nastavnik pristupa stranici repozitorija u sustavu.
 2. Nastavnik odabire predmet za koji želi objaviti datoteku.
 3. Nastavnik odabire vrstu materijala koju želi dodati (PDF, video, slika, itd.)
 4. Nastavnik učitava materijal na repozitorij. (F-010)
 5. Nastavnik može dodati opis materijala po potrebi.

6. Sustav pohranjuje samo metapodatke materijala u bazu podataka.
 7. Nakon objave, materijal postaje vidljiv samo učenicima upisanim na predmet.
- Opis mogućih odstupanja:
 4. Nastavnik nije uspio učitati materijal (npr. loša internetska veza, neispravan format datoteke).**Rješenje:** Sustav prikazuje poruku o grešci i omogućuje ponovno učitavanje materijala.

UC8 - Pristupanje materijalima

- Glavni sudionik: Učenik
- Cilj: Omogućiti učeniku pristup potrebnim materijalima
- Sudionici: Baza podataka
- Preduvjet: Učenik je prijavljen u sustav i upisan je na predmet čijim materijalima želi pristupiti
- Opis osnovnog tijeka:
 1. Učenik pristupa stranici repozitorija u sustavu.
 2. Učenik odabire predmet za koji želi pregledati materijale.
 3. Sustav prikazuje popis materijala za odabrani predmet.
 4. Učenik odabire materijal koji želi pregledati ili preuzeti. (F-011)
 5. Učenik pregledava materijal.
- Opis mogućih odstupanja:
 2. Učeniku je zabranjen pristup za pregled materijala na odabranom predmetu.**Rješenje:** Učenik može zatražiti pristup materijalima tako da pošalje e-mail administratoru, nakon čega mu administrator ručno dodjeljuje prava ako zaključi da je riječ o pogrešci.

UC9 - Izdavanje potvrda o upisu

- Glavni sudionik: Učenik
- Cilj: Omogućiti učeniku preuzimanje potvrde o upisu
- Sudionici: Baza podataka
- Preduvjet: Učenik je prijavljen u sustavu i upisan u bazu podataka
- Opis osnovnog tijeka:
 1. Učenik pristupa stranici za potvrde unutar sustava.
 2. Učenik odabire opciju "Izdavanje potvrde o upisu".
 3. Sustav generira potvrdu o upisu koristeći podatke o učeniku iz baze.
 4. Potvrda o upisu se šalje ne e-mail učenika u roku 15 minuta. (F-007)
- Opis mogućih odstupanja:
 4. Potvrda nije stigla na email**Rješenje:** Učenik može ponovno zatražiti generiranje potvrde, ali tek nakon 60 minuta od prvog pokušaja.

UC10 - Upute za odlazak na praksu

- Glavni sudionik: Učenik
- Cilj: Omogućiti učeniku prikaz uputa za dolazak na lokaciju prakse od škole
- Sudionici: Baza podataka, Google Maps API
- Preduvjet: Učenik je prijavljen u sustav

- Opis osnovnog tijeka:

1. Praksa se svakom učeniku dodjeljuje prilikom generiranja rasporeda.
2. Podaci o praksi zapisani su u bazi podataka. (F-006)
3. Učenik pristupa stranici prakse unutar sustava.
4. Na stranici se prikazuje karta koristeći Google Maps API-ja s označenom lokacijom prakse.
5. Učenik može zatražiti prikaz rute od škole do prakse. (F-008)
6. Generira se prikaz rute na karti.
7. Sustav pohranjuje ID učenika i datum pristupa karti u bazu podataka.

- Opis mogućih odstupanja:

- 3./4. Karta ili upute nisu ispravno prikazane na stranici.

Rješenje: Učenik može pokušati ponovno učitati stranicu ili obavijestiti administratora o problemu.

UC11 - Promjene u rasporedu

- Glavni sudionik: Nastavnik
- Cilj: Omogućiti nastavniku da zatraži zamjenu u rasporedu putem obrasca u aplikaciji
- Sudionici: Baza podataka, satničar
- Preduvjet: Nastavnik je prijavljen u sustav
- Opis osnovnog tijeka:

1. Nastavnik pristupa stranici za promjene u rasporedu.
2. Nastavnik ispunjava obrazac za promjenu koju želi (unos predmeta, razlog zamjene i datum)
3. Nakon što nastavnik ispunio obrazac, sustav ga automatski proslijeđuje satničaru. (F-014)
4. Satničar prima zahtjev.
5. Satničar prihvaća zahtjev i ručno obavlja zadanu promjenu unutar rasporeda.
6. Promjena se ažurira u bazi podataka.
7. Nastavnik dobiva obavijest o obradi zahtjeva unutar stranice za zamjene.
8. Korisnici dobivaju e-mail o provedenoj promjeni.

- Opis mogućih odstupanja:

2. Nastavnik nije ispunio sve potrebne podatke u obrascu.

Rješenje: Sustav ne dopušta slanje obrasca dok svi podaci nisu ispunjeni.

5. Satničar zbog konflikata ne uspijeva prihvatiti zahtjev nastavnika.

Rješenje: Satničar odbija zahtjev i nastavniku se šalje obavijest o odbijenom zahtjevu za zamjenu. Postupak se ovdje zaustavlja dok se ne pošalje novi zahtjev.

UC12 - Unos izostanaka

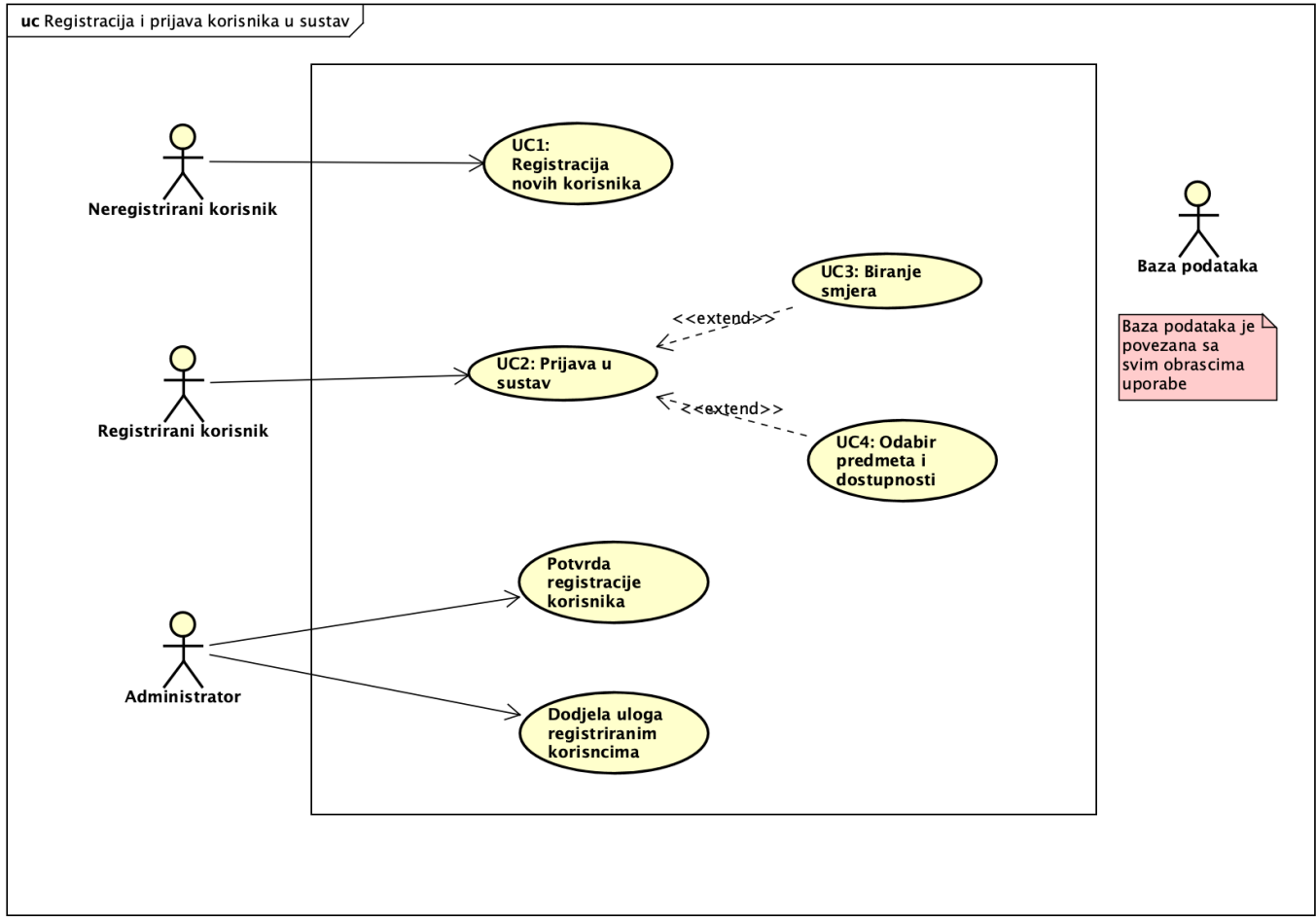
- Glavni sudionik: Nastavnik
- Cilj: Omogućiti nastavniku unos izostanaka za učenike.
- Sudionici: Baza podataka, Google Maps API
- Preduvjet: Nastavnik je prijavljen u sustav i ima ulogu razrednika
- Opis osnovnog tijeka:

1. Nastavnik pristupa stranici za unos izostanaka unutar sustava.
2. Sustav prikazuje popis učenika iz razredne grupe nastavnika.
3. Nastavnik označava učenika/e koji je/su izostao/li.

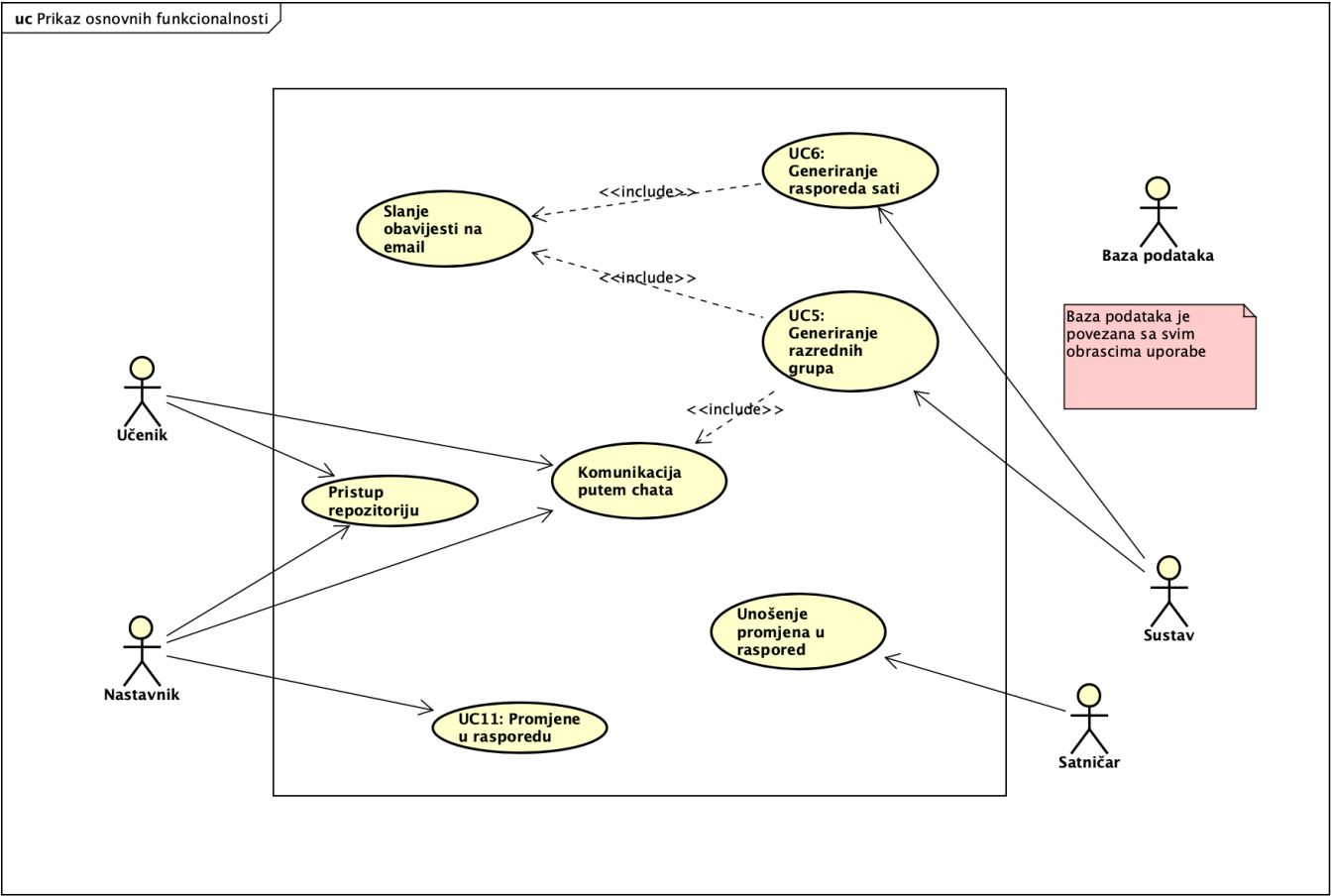
- 4. Nastavnik unosi razlog i datum izostanka za učenika/e.
- 5. Nastavnik potvrđuje unos izostanka.
- 6. Sustav pohranjuje podatke o izostanku u bazu podataka.

- Opis mogućih odstupanja:
 - 4. Nastavnik ne unese sve potrebne podatke (razlog, status).
- Rješenje:** Sustav ne dopušta unos izostanaka dok svi podaci nisu ispunjeni.

Dijagrami obrazaca uporabe

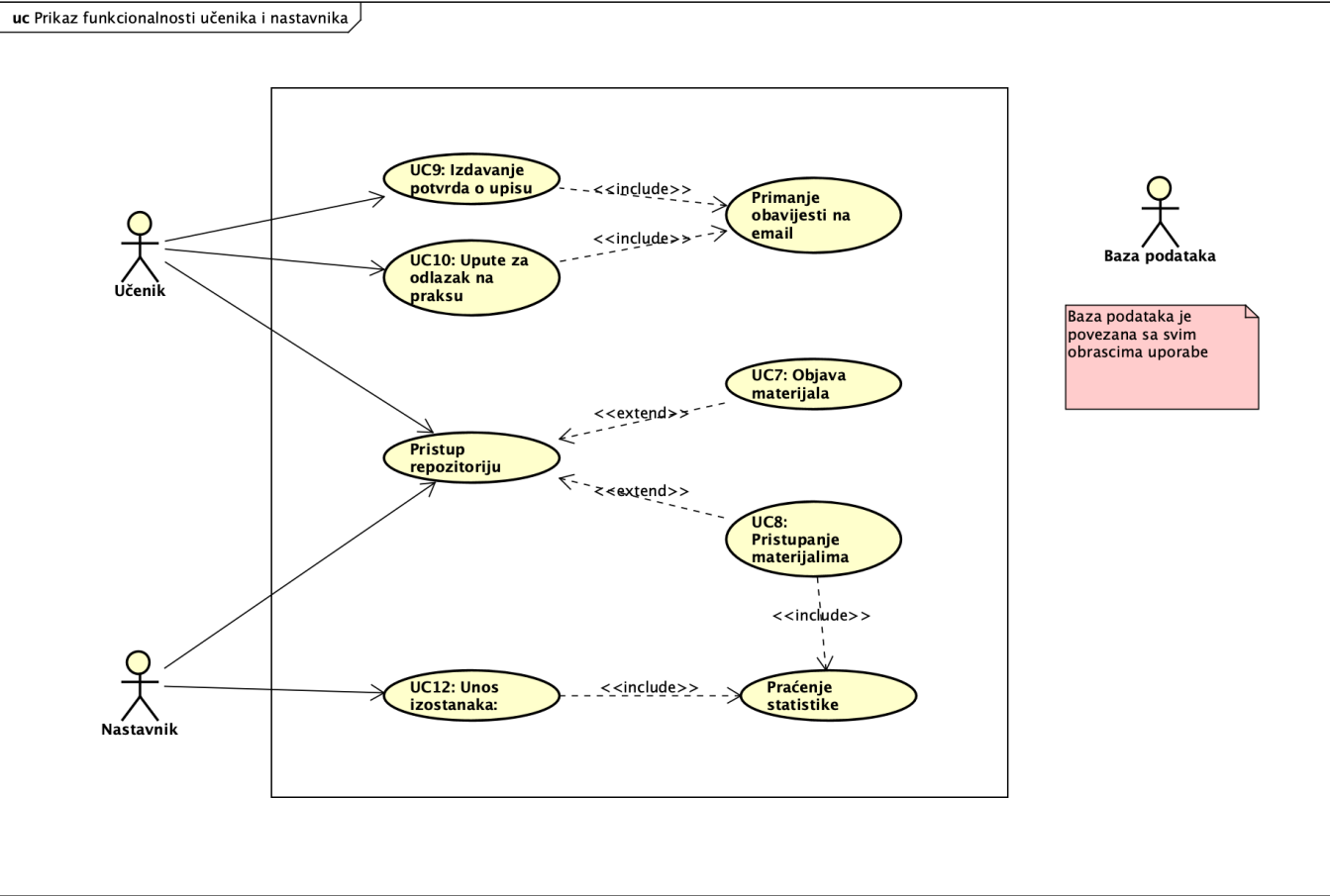


1.1 Prikaz registracije i prijave korisnika u sustav



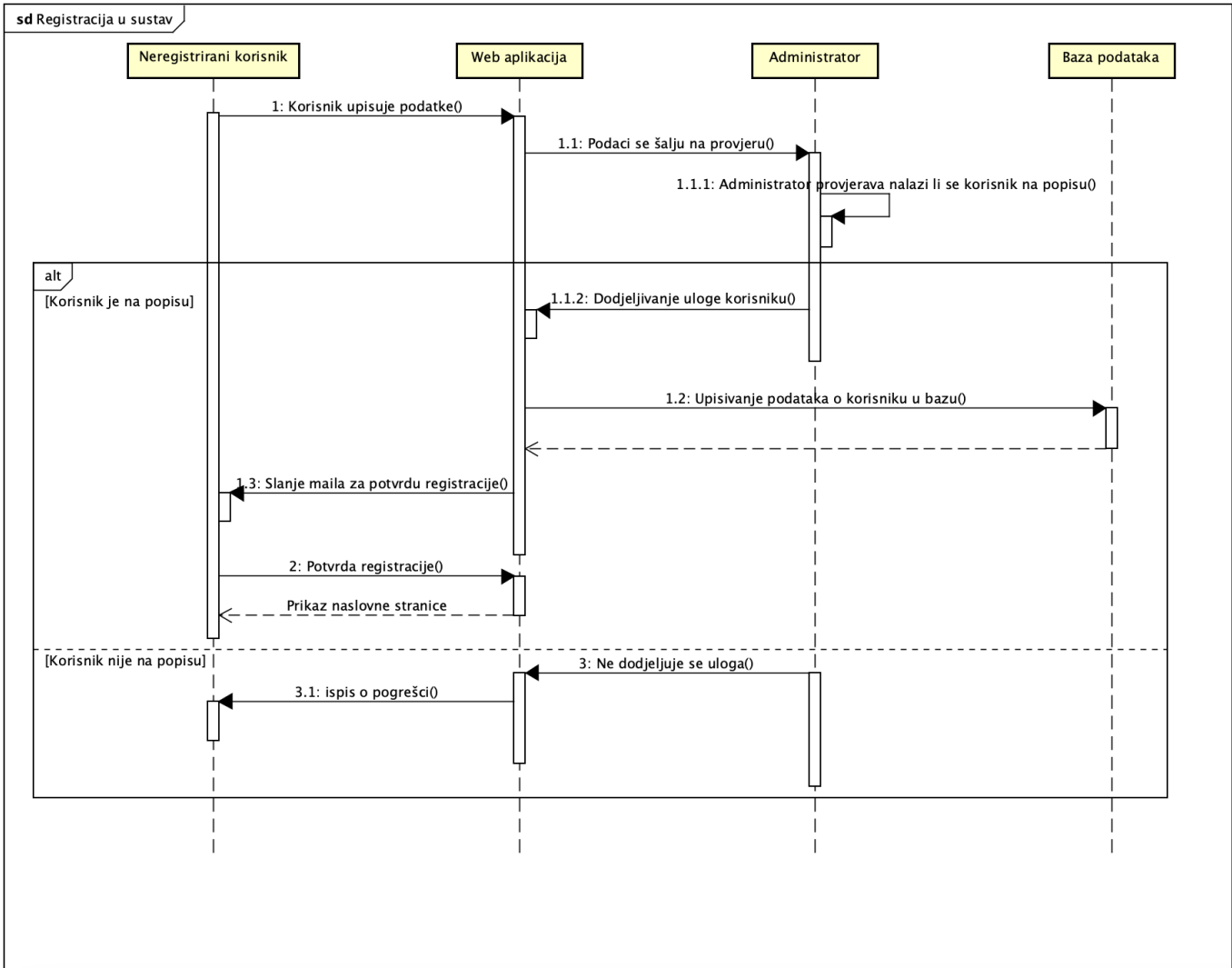
1.2 Prikaz osnovnih funkcionalnosti sustava

*obrasci uporabe UC5 i UC6 odnose se na izvođenje algoritama za generiranje grupa i rasporeda koje razvija razvojni tim, ali smo stavili sustav kao aktora budući da aplikacija, tj. sustav izvodi navedene obrasce

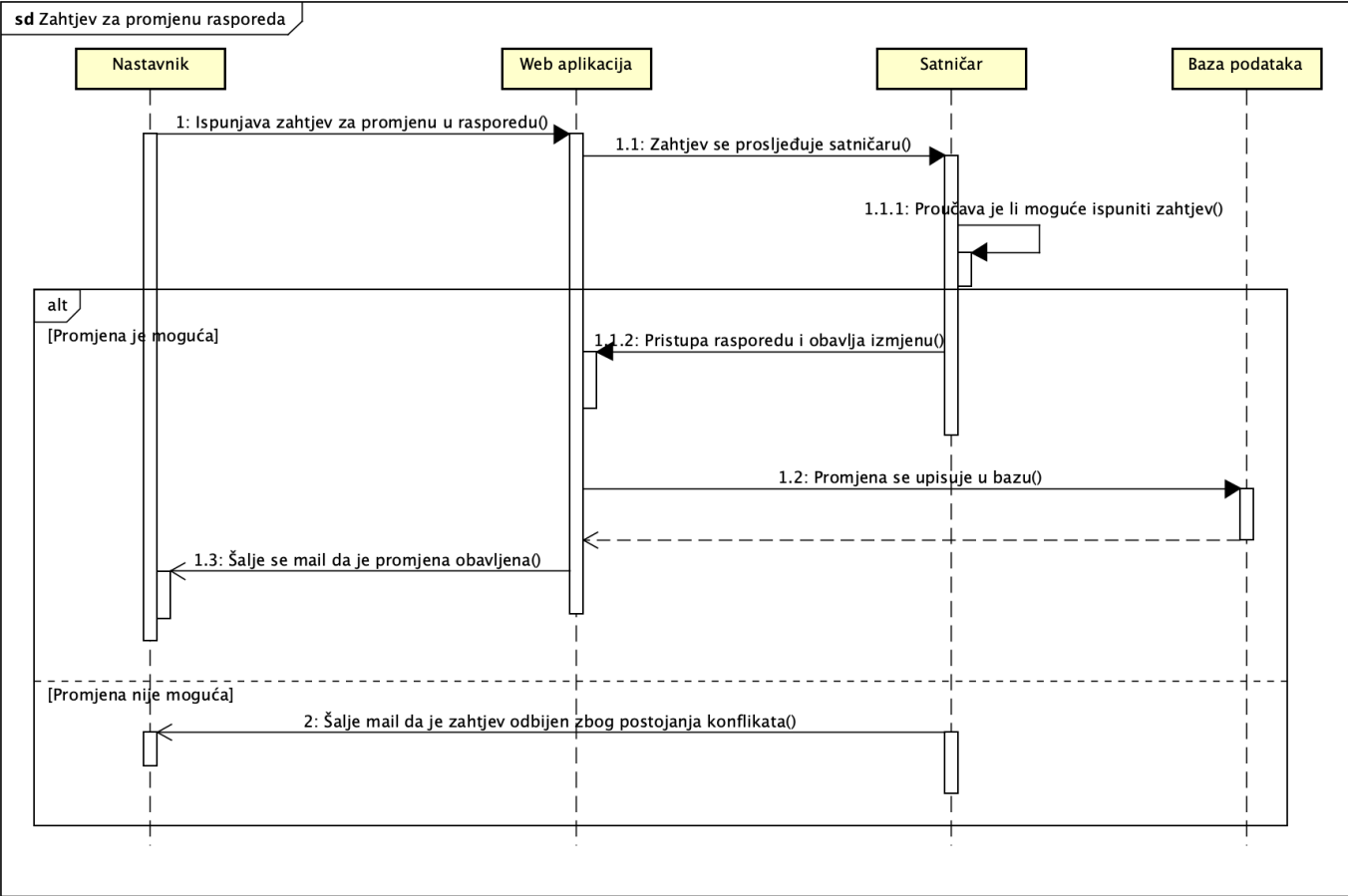


1.3 Prikaz funkcionalnosti učenika i nastavnika

Sekvencijski dijagrami

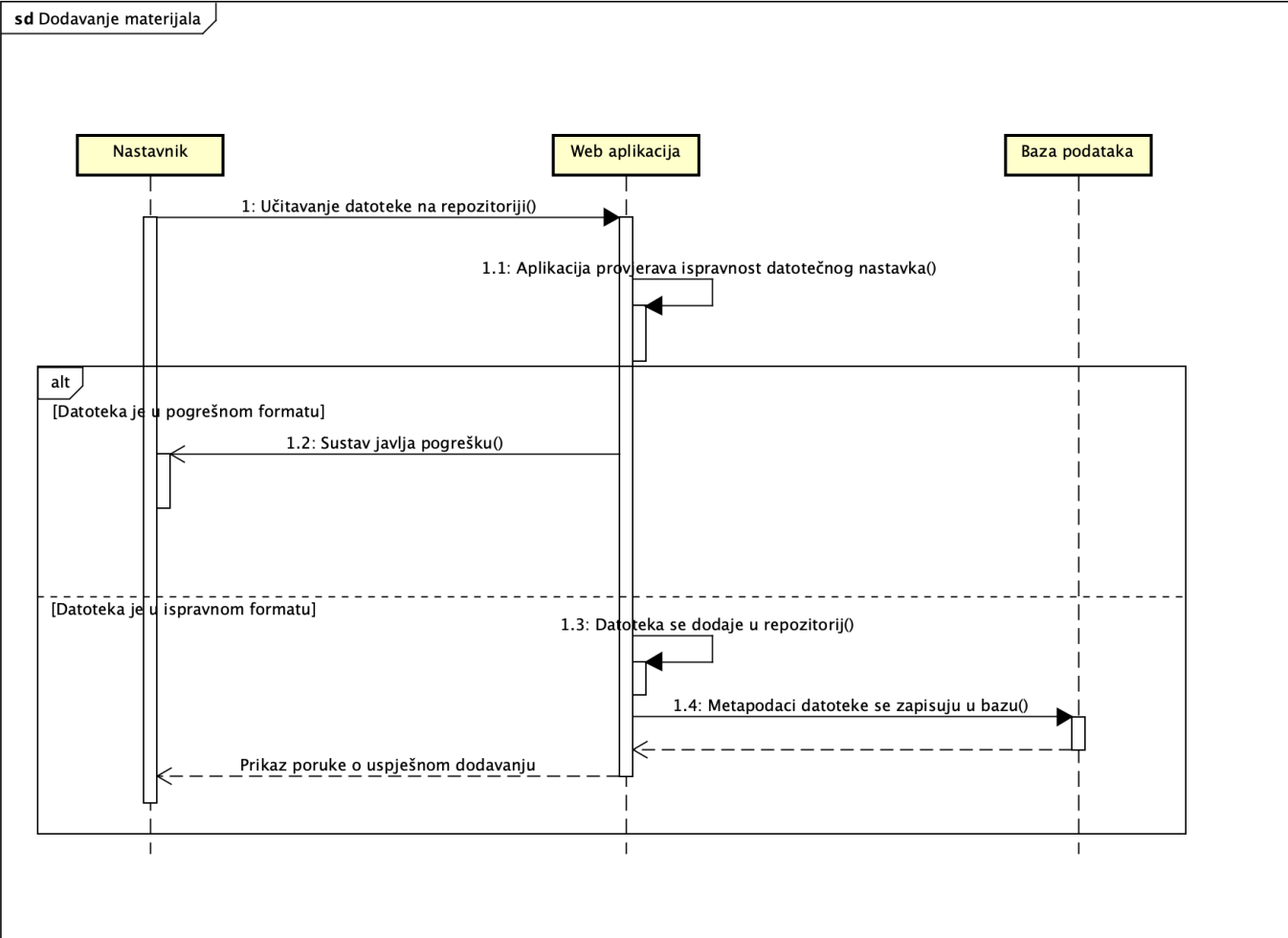


2.1 Sekvencijski dijagram prikazuje kako izgleda registracija novih korisnika u sustav. Prilikom registracije korisnik upisuje podatke, odnosno koristi svoj Google ili Microsoft račun. Te podatke zatim administrator (ravnatelj) osobno provjerava, odnosno ima listu dozvoljenog osoblja i učenika škole te provjerava nalazi li se neregistrirani korisnik na tom popisu. Ako se korisnik nalazi na popisu ravnatelj mu dodjeljuje odgovarajuću ulogu te potvrđuje njegovu registraciju. Podaci o novom korisniku upisuju se u bazu te aplikacija šalje obavijest korisniku na mail da je registracija uspjela uz zahtjev za potvrdu registracije. Jednom kad korisnik potvrdi registraciju, prikazuje mu se naslovna stranica aplikacije. Ako ravnatelj ne pronalazi neregistriranog korisnika na popisu, odbija njegovu registraciju i aplikacija šalje obavijest da registracija nije uspjela.



2.2 Sekvencijski dijagram prikazuje kako nastavnik može zatražiti od satničara da obavi promjenu u rasporedu

Jednom kad nastavnik odluči zatražiti promjenu za rasporedu ispunjava zahtjev na stranici o navedenoj promjeni. Satničar prima ispunjeni obrazac i ručno provjerava je li moguće obaviti promjenu koju je nastavnik zatražio. Ako je promjena moguća satničar pristupa rasporedu i unosi promjenu koja se onda upisuje u bazu podataka. Nakon toga aplikacija šalje obavijest o promjeni na e-mail nastavniku (i učenicima). Ako naveden zahtjev nije bilo moguće provesti, satničar osobno putem e-maila javlja nastavniku da nije uspio obaviti promjenu jer postoje konflikti u rasporedu.



2.3 Sekvencijski dijagram prikazuje kako izgleda kad nastavnik dodaje materijale u repozitorij

Kad nastavnik želi objaviti novi nastavni materijal pristupa stranici repozitorija i odabire predmet u sklopu kojeg želi objaviti datoteku. Zatim odabire format datoteke koji želi objaviti (.pdf, .txt, .mp4, itd.) te učitava datoteku. Aplikacija zatim provjerava odgovara li nastavak datoteka koju treba učitati odabranom nastavku. Ako se formati ne poklapaju, sustav ispisuje poruku o grešci i nastavnik ima mogućnost ponovno učitati datoteku. Inače, datoteka se dodaje u repozitorij predmeta, a njeni metapodaci spremaju se u bazu. Aplikaciju prikazuje poruku o uspješnom dodavanju materijala.

*Metapodaci u ovom slučaju označavaju URL poveznice na materijale, ime autora, veličinu datoteke i sl., odnosno naglašavamo da se same datoteke ne spremaju direktno u bazu

Provjera uključenosti ključnih funkcionalnosti u obrasce uporabe

Obrazac uporabe	Funkcionalni zahtjev
UC1 - Registracija novih korisnika	F-001
UC2 - Prijava korisnika u sustav	F-002
UC3 - Biranje smjera	F-004, F-007
UC4 - Biranje predmeta i označavanje dostupnosti	F-004
UC5 - Generiranje razrednih grupa	F-003, F-006, F-009

Obrazac uporabe	Funkcionalni zahtjev
UC6 - Generiranje rasporeda sati	F-004, F-005
UC7 - Objava materijala	F-010
UC8 - Pristupanje materijalima	F-011
UC9 - Izdavanje potvrda o upisu	F-007
UC10 - Upute za odlazak na praksu	F-006, F-008
UC11 - Promjene u rasporedu	F-014
UC12 - Unos izostanaka	F-013

Arhitektura sustava

Opis arhitekture

Stil arhitekture

Odabrali smo mikroservisnu arhitekturu uz event-driven pristup kako bismo omogućili modularnost, fleksibilno skaliranje i olakšano održavanje aplikacije. Mikroservisi nam omogućuju da različite funkcionalnosti aplikacije (kao što su rasporedi, komunikacija i obavijesti) budu razvijene kao zasebne jedinice, što omogućuje neovisnost i skalabilnost svake funkcionalnosti prema potrebi. Event-driven pristup osigurava brz i asinkron odgovor na događaje (npr. promjene rasporeda), što smanjuje opterećenje glavnih servisa, omogućuje bolju prilagodbu korisnicima i pojednostavljuje integraciju vanjskih servisa poput vremenskih i kartografskih API-ja.

Podsustavi:

Podsustav za upravljanje korisnicima

- Opis: Ovaj podsustav obuhvaća registraciju, autentifikaciju (OAuth2), upravljanje korisničkim profilima i autentifikaciju putem vanjskih servisa (Google, Microsoft). Također, podržava diferencijaciju korisničkih uloga (učenici, nastavnici, administrator, ravnatelj, satničari).
- Funkcije: Prijava i registracija korisnika, autentifikacija, upravljanje korisničkim podacima i pravima pristupa.

Podsustav za upravljanje rasporedom

- Opis: Automatizirano generiranje rasporeda sati za različite smjerove i razrede, uključujući specifične predmete i praktične aktivnosti.
- Funkcije: Generiranje i prilagodba rasporeda, obavješćavanje korisnika o promjenama putem e-pošte, upravljanje posebnim prostorima za praktičnu nastavu.

Podsustav za obavijesti i komunikaciju

- Opis: Zadužen za slanje obavijesti i omogućavanje komunikacije među korisnicima putem poruka.
- Funkcije: Slanje e-mail obavijesti, sustav za razmjenu poruka među korisnicima, mogućnost grupnih obavijesti za određene razrede ili smjerove.

Podsustav za upravljanje nastavnim materijalima

- Opis: Omogućuje objavljivanje i organizaciju nastavnih materijala isključivo za učenike kojima su namijenjeni.
- Funkcije: Postavljanje i distribucija materijala prema smjeru ili razredu, upravljanje pristupom za učenike i nastavnike.

Podsustav za izdavanje potvrda i administraciju dokumenata

- Opis: Automatizira izdavanje potvrda (npr. potvrde o upisu) za učenike radi olakšavanja administrativnih zadataka, poput pribavljanja pokaznih kartica za prijevoz.

- Funkcije: Izdavanje potvrda i drugih potrebnih dokumenata, integracija s dokumentacijskim standardima škole.

Podsustav za praćenje aktivnosti i statistike

- Opis: Praćenje pristupa aplikaciji i statistička obrada korištenja funkcionalnosti kako bi se razumjelo ponašanje korisnika i olakšalo donošenje administrativnih odluka.
- Funkcije: Prikupljanje i analiza podataka o pristupima, kreiranje izvještaja za administraciju i omogućavanje uvida u popularnost različitih dijelova aplikacije.

Podsustav za integraciju kartografskih i vremenskih podataka

- Opis: Podrška za terensku nastavu i prakse, uključujući prikaz vremenske prognoze i kartografskih podataka za lokaciju škole i drugih relevantnih mjesta.
- Funkcije: Prikaz vremenske prognoze, integracija s kartografskim API-jem, prilagodba prema lokaciji korisnika ili aktivnosti.

Preslikavanje na radnu platformu

Za preslikavanje web-aplikacije koristit ćemo platformu Render. Render omogućuje sigurno dodavanje i upravljanje varijablama okruženja, koje su ključne za rad aplikacije u produkciji. Potrebno je definirati sljedeće varijable okruženja:

- JWT_SECRET - tajni ključ za generiranje JSON Web Tokena (JWT), osigurava siguran prijenos i provjeru korisničkih podataka
- OAUTH_CLIENT_ID i OAUTH_CLIENT_SECRET - ključevi za OAuth2 autentifikaciju putem treće strane (npr. Google, Microsoft...)
- DATABASE_URL - URL baze podataka koju koristi aplikacija. Render nudi PostgreSQL baze podataka koje se mogu koristiti, ili se može povezati s vanjskom bazom
- NODE_ENV - postavljena na "production" kako bi aplikacija radila u produkcijskom modu
- PORT - Render postavlja port koji aplikacija sluša putem varijable okruženja PORT

Render također automatski pruža HTTPS, pa je potrebno osigurati redirekciju s HTTP-a na HTTPS. Ovakva konfiguracija aplikacije na platformi Render osigurava optimalne sigurnosne i konfiguracijske postavke za stabilan rad web-aplikacije.

Spremišta podataka

Za spremište podatka koristimo relacijsku bazu podataka (PostgreSQL), što znači da su podaci strukturirani u tablicama koje su međusobno povezane pomoću ključeva, što olakšava organizaciju i pretraživanje. Za modeliranje baze koristimo ERDplus kako bismo jasno prikazali odnose među entitetima, kao što su UČENIK, DJELATNIK, PREDMET itd.

Mrežni protokoli

Web-aplikacija koristi nekoliko protokola kako bi garantirala sigurnu i efikasnu komunikaciju između korisnika i servera:

1. TCP/IP (Transmission Control Protocol/Internet Protocol): TCP/IP je bazični protokol za sigurni prijenos podataka preko mreža. IP usmjerava pakete, čime osigurava da su došli do točnog odredišta,

dok TCP osigurava integritet poslanih paketa, što osigurava da su došli na odredište u dobrom obliku. Svi protokoli koji rade na stranici djeluju preko TCP/IP-a, što osigurava točnost i pouzdanost prijenosa.

2. TLS/SSL (Transport Layer Security/Secure Sockets Layer): TLS/SSL su kriptografski protokoli koji služe za osiguravanje prijenosa podataka između klijenta i servera. Ovi protokoli omogućuju enkripciju, integritet i autentifikaciju podataka čime garantiraju da podatci ostanu privatni i nepromijenjeni.
3. HTTPS (HTTP Secure): HTTPS se nalazi unutar svakog web-prometa i osigurava sigurni prijenos podataka između klijenata i servera koristeći TLS/SSL enkripcije. Ovaj protokol je osnova osiguravanja privatnosti i integriteta stranice osobito pri prijavi korisnika u sustav ili poruka između korisnika.
4. OAuth 2.0: OAuth se koristi za autentifikaciju korisnika preko "third-party" poslužiteljima kao što su Google i Microsoft. Ovaj protokol omogućava sigurni pristup tako što dopušta korisnicima da se prijave koristeći postojeće račune bez upisivanja lozinke.
5. SMTP (Simple Mail Transfer Protocol): SMTP služi slanju email-ova od aplikacije do korisnika. Ovaj protokol pomaže pri slanju obavijesti, potvrda i sličnih poruka na siguran način. Kada je ovaj protokol u kombinaciji s TLS/SSL protokolima, osigurava enkripciju prometa, obavijesti i poruka.
6. WebSocket: WebSocket služi za razmjenjivanje poruka između korisnika. Pomoću ovog protokola poruke mogu biti razmijenjene bez potrebe da korisnik osvježava stranicu ili da se podatci o poruci traže sa servera. Također WebSocket protokol također koristi TLS/SSL protokole za enkripciju poruka.

Globalni upravljački tok

Web-aplikacija sadrži Node.js i Express.js kao glavne tehnologije za backend, te React za frontend. Za prijavu korisnika koristi se OAuth2.0 autentifikacija, nakon čega se korisnikov access token sprema u cookie s istekom od sat vremena. Refresh token se sprema u bazu podataka i koristi se za dobivanje novog access tokena kako se korisnik ne bi morao često ponovno prijavljivati u stranicu. Svaki korisnik ima određena dopuštenja, koja se kontroliraju JSON Web Tokenom (provjera autentičnosti). Ako korisnik nema dopuštenje za neku radnju na stranici bit će mu odbijen pristup u slučaju pokušaja odrađivanja te radnje. Korisnički podatci se spremaju u bazu podataka, kao i dokumenti koji će biti učitani na stranicu. Općeniti tok kroz aplikaciju izgleda otprilike ovako:

1. Korisnik interakcijom s frontendom šalje HTTPS zahtjev prema backendu.
2. Provjera autentifikacije: Radi se provjera ima li korisnik valjan JWT za radnju koju pokušava odraditi.
3. Ako je zahtjev ovlašten, aplikacija obrađuje podatke i izvršava potrebnu logiku (npr. vraća korisničke podatke, obrađuje prijavu, sprema dokumente, itd.)
4. Aplikacija šalje odgovor u JSON formatu frontendu koji korisniku prezentira rezultat radnje (npr. uspjela radnja, greška u izradi, odbijen pristup, itd.)

Sklopovskoprogramski zahtjevi

Za uspješno pokretanje aplikacije potrebna je stabilna internetska veza za podršku HTTPS zahtjeva (preporučena brzina je najmanje 1 Mbps). Aplikacija podržava većinu operacijskih sustava (Windows, Linux i sl.). Za razvoj potreban je minimalno dvojezgreni procesor te minimalno 1 GB RAM-a (za produkciju preporučeno je 2 GB RAM-a ili više). Potrebna pohrana je oko 500 MB, ali u rijetkim slučajevima može biti potrebno i više ako se radi s nekim većim datotekama.

Softverski zahtjevi za razvoj uključuju Node.js (14.x ili noviji), Express.js, Passport.js, JWT, bazu podataka (SQL, MongoDB) te razne knjižnice za razvoj poput cookie-parsera, body-parsera, dotenv i sl. Potrebne su i varijable okruženja koje će biti uključene putem Render platforme.

Obrazloženje odabira arhitekture

Izbor arhitekture temeljen na principima oblikovanja:

Razdvajanje odgovornosti - mikroservisi omogućuju jasno razdvajanje funkcionalnosti, što olakšava održavanje i nadogradnju aplikacije.

Fleksibilnost i skalabilnost - arhitektura omogućava skaliranje pojedinačnih servisa prema specifičnim zahtjevima (npr. učestalost korištenja obavijesti ili rasporeda).

Asinkrona obrada i brzi odziv - event-driven pristup podržava asinkronu komunikaciju, smanjujući kašnjenja i poboljšavajući iskustvo korisnika.

Sigurnost i izolacija - svaki mikroservis funkcionira kao zasebna cjelina, čime se poboljšava sigurnost i smanjuje utjecaj eventualnih problema na cijeli sustav.

Fleksibilna integracija vanjskih servisa - event-driven pristup olakšava povezivanje s vanjskim servisima za podršku, poput vremenskih i kartografskih aplikacija, koje se mogu automatski pozvati prema potrebi.

Razmatrane alternative:

Monolitna arhitektura

- Opis: Jednostavnija za implementaciju i manji troškovi održavanja u početnoj fazi. Svi dijelovi aplikacije su povezani u jedan deployment paket.
- Prednosti: Jednostavnija razvojna i održavana struktura za manje timove. Konzistentnost podataka jer svi podaci koriste istu bazu. Manji troškovi infrastrukture u odnosu na mikroservisnu arhitekturu.
- Nedostaci: Teže skaliranje specifičnih dijelova aplikacije. Kako aplikacija raste, održavanje postaje složenije, a ažuriranja mogu izazvati veće smetnje. Primjena: Prikladna za početne verzije aplikacije ili ako se očekuje manji broj korisnika. Kasnije se može prijeći na mikroservisnu arhitekturu ako aplikacija postane zahtjevnija.

Modularni monolit

- Opis: Sličan monolitnoj arhitekturi, ali unutar jedinstvenog deployment paketa organizira funkcionalnosti u jasno odvojene module (npr. modul za raspored, komunikaciju, resurse).
- Prednosti: Bolja organizacija koda zahvaljujući modularnosti. Fleksibilnost za buduće podjele na mikroservise. Manji troškovi u usporedbi s mikroservisima, ali nudi dobru osnovu za skaliranje.
- Nedostaci: Nema potpune neovisnosti modula kao u mikroservisima. Skalabilnost je ograničena jer svi moduli rade unutar jedne aplikacije. Primjena: Odlična opcija za aplikacije u ranoj fazi koje će možda s vremenom trebati biti preuređene u mikroservise.

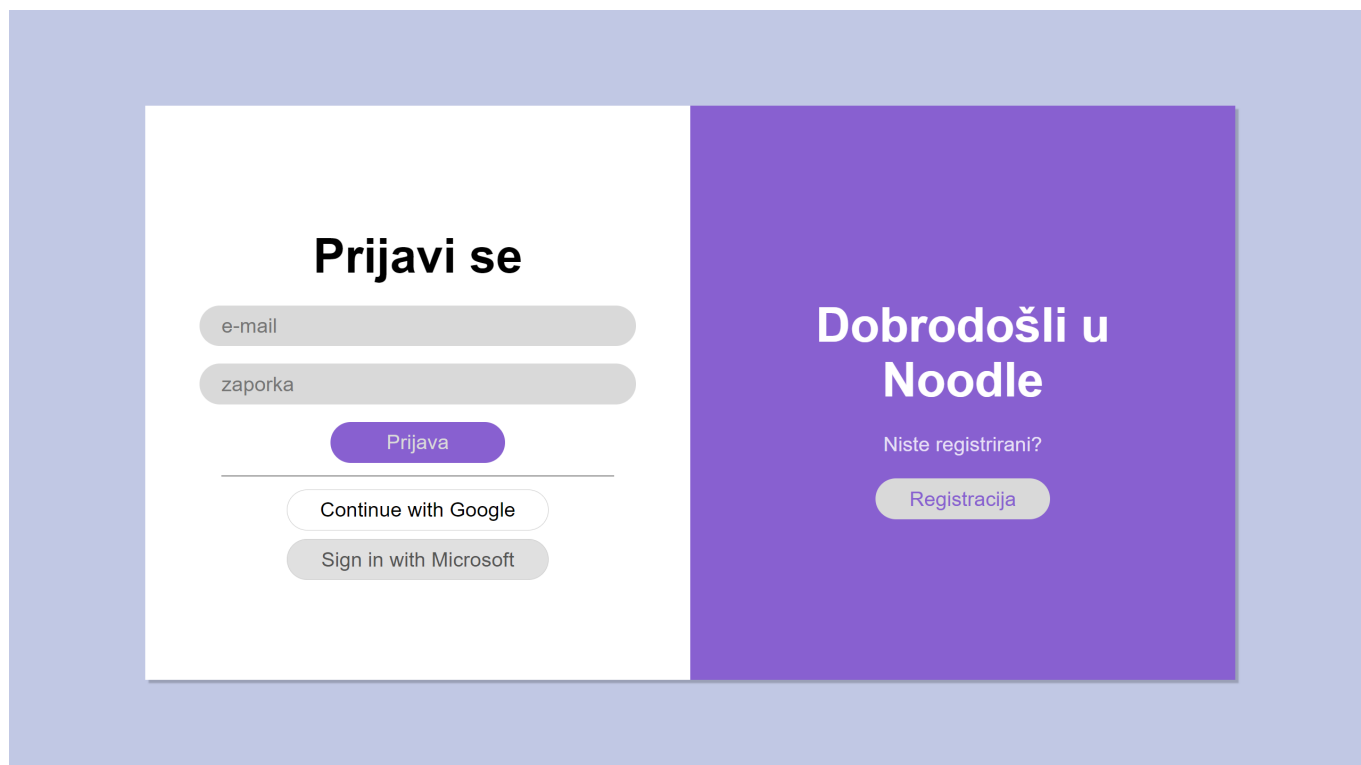
MVC (Model-View-Controller) arhitektura

- Opis: MVC je arhitektonski obrazac koji razdvaja aplikaciju na tri glavna dijela: Model (podatkovni sloj i poslovna logika), View (korisničko sučelje) i Controller (upravljanje korisničkim zahtjevima i komunikacija između Modela i Viewa).

- **Prednosti:** Jasna separacija odgovornosti koja olakšava održavanje i razvoj. Jednostavnije testiranje komponenti zasebno. Pogodnost za timski rad jer različiti timovi mogu paralelno raditi na različitim slojevima.
- **Nedostaci:** Može postati kompleksan u vrlo velikim aplikacijama. Nije inherentno skalabilan kao mikroservisna arhitektura, ali može se kombinirati s drugim arhitektonskim stilovima. Primjena: Prikladno za aplikacije koje zahtijevaju jasnu organizaciju koda i odvojene slojeve poslovne logike i prezentacije. Može biti dobra alternativa za manje do srednje kompleksne aplikacije ili se može koristiti unutar modularnog monolita.

Organizacija sustava na visokoj razini

- **Klijent-poslužitelj:** Aplikacija slijedi klijent-poslužitelj arhitekturu što znači da se sastoji od dva glavna dijela: **klijent** (frontend) i **poslužitelj** (backend). Klijent je odgovoran za prikaz korisničkog sučelja, slanje zahtjeva i prijem i prikaz odgovora, dok poslužitelj zaprima zahtjeve, obrađuje podatke i izvršava poslovnu logiku. Poslužitelj pristupa bazi podataka kada je potrebno, generira odgovor i šalje ga nazad klijentu u JSON formatu koji prezentira podatke na odgovarajući način. Komunikacija između klijenta i poslužitelja odvija se preko HTTPS protokola, a za siguran pristup koristi se JWT za autorizaciju korisnika.
- **Baza podataka:** Baza podataka je sastavljena od osobnih podataka o korisnicima aplikacije, repozitorija sa svim linkovima koji se koriste, predmetima u školskom kurikulumu, prostorijama škole te rasporeda sati. Služi nam kako bismo mogli spremati sve to na lako dostupno mjesto za potrebe aplikacije kao što su dohvat tih podataka, razlikovanju korisnika po pristupu, slaganja rasporeda itd.
- **Datotečni sustav:** Baza podataka koristi spremljene linkove koji su povezani na mape na cloudu kako bi dohvaćali potrebne datoteke.
- **Grafičko sučelje:** Web aplikacija s responzivnim dizajnom prilagođenim korištenju na različitim uređajima (računala, tableti, mobilni uređaji) koja korisnicima (učenicima, nastavnicima, administratorima) omogućuje jednostavan pristup svim funkcionalnostima sustava putem intuitivnog sučelja. Frontend je implementiran korištenjem Reacta.



Organizacija aplikacije

Frontend i backend slojevi

- **Slojevi backenda omogućuju sigurnost i održivost aplikacije:**

1. Sloj za rute - sloj za rute upravlja HTTP zahtjevima i usmjerava ih na odgovarajuće kontrolere koji obrađuju logiku.
2. Kontrolni sloj - obrađuje logiku i komunicira s bazom podataka ili drugim servisima.
3. Sloj za autentifikaciju i autorizaciju - sloj odgovoran za prijavljivanje korisnika i osigurava pristup zaštićenim dijelovima aplikacije samo ovlaštenim korisnicima.
4. Sloj za Middleware - detektira zahtjeve prije nego što dođu do kontrolera i odgovore nakon što su kreirani. Omogućava dodatne funkcionalnosti kao rukovanje kolačićima, parsiranje tijela zahtjeva i sl.
5. Sloj za upravljanje okruženjem - koristi varijable okruženja kako bi osigurao sigurnost osjetljivih podataka i konfiguracija kao tokeni, port
6. Sloj za povezivanje s bazom podataka - omogućava pohranu i dohvaćanje podataka iz baze
7. Sloj za sigurnost - sloj koji koristi HTTPS i SSL (HTTP sigurnosni sloj) kako bi se osigurala sigurnost podataka i spriječili napadi

- **Slojevi frontenda omogućuju interaktivnost i optimalno korisničko iskustvo aplikacije:**

1. Sloj za prikaz - sloj za prikaz odgovoran je za vizualno predstavljanje aplikacije, koristeći alate i tehnologije za stiliziranje elemenata (npr. CSS).
2. Sloj za upravljanje komponentama - upravlja stvaranjem, organizacijom i ponovnom upotrebom komponenti kako bi bile modularne i jednostavne za održavanje, što olakšava razvoj aplikacije.
3. Sloj za komunikaciju s API-jem - sloj koji upravlja slanjem zahtjeva prema backendu za dohvaćanje ili slanje podataka te obrađuje odgovore, uključujući rukovanje pogreškama i učitavanjem podataka.
4. Sloj za optimizaciju performansi - sloj koji poboljšava korisničko iskustvo smanjujući vrijeme učitavanja i ubrzavajući aplikaciju.
5. Sloj za validaciju unosa - provjerava ispravnost podataka koje korisnici unose prije slanja na backend, kao što su ispravnost e-maila, formata broja ili obavezna polja.

Baza podataka

Za svaku školu postoji zasebna baza podataka koja uključuje osobne podatke korisnika, podatke korisnika ovisno o njihovoj ulozi, predmete iz školskog kurikuluma, repozitorij školskih materijala i obavijesti te podatke o prostorijama škole.

Opis tablica

ŠKOLA

Atribut	Tip podatka	Opis varijable
školaID	INT	Jedinstveni identifikator škole

- primarni ključ: školaID
- **imaRaspored** - relacija koja označava da škola ima raspored

KORISNIK

Atribut	Tip podatka	Opis varijable
OIB	INT	Jedinstveni identifikator korisnika
ime	VARCHAR	Ime korisnika
prezime	VARCHAR	Prezime korisnika
spol	VARCHAR	Spol korisnika
adresa	VARCHAR	Adresa korisnika
e-mail	VARCHAR	Elektronička pošta korisnika
zaporka	VARCHAR	Zaporka korisnika
školaID	INT	Jedinstveni identifikator škole

- primarni ključ: OIB
- strani ključ: školaID
- predstavlja bilo kojeg korisnika koji se registrira/prijavljuje na stranici škole
- **sudionik** - relacija koja označava sve korisnike koji se nalaze u školi (gosti, učenici i djelatnici)

GOST

Atribut	Tip podatka	Opis varijable
gostID	INT	Jedinstveni identifikator gosta
datumPristupa	DATETIME	Označava kad je gost pristupio stranici
OIB	INT	Jedinstveni identifikator korisnika

- primarni ključ: OIB
- strani ključ: OIB
- unique: gostID
- predstavlja podtip registrirane osobe koja čeka dozvolu za ulazak na stranicu

DJELATNIK

Atribut	Tip podatka	Opis varijable
djelatnikID	INT	Jedinstveni identifikator
mobBroj	INT	Broj mobitela djelatnika
razred	VARCHAR	Popis razreda kojima djelatnik predaje
status	VARCHAR	Određuje uloga djelatnika
OIB	INT	Jedinstveni identifikator korisnika

- primarni ključ: OIB
- strani ključ: OIB
- unique: djelatnikID

- predstavlja podtip registriranog korisnika koji je djelatnik u školi
- djelatnik može biti: ravnatelj, profesor i satničar (određeno atributom status)
- podtipovi djelatnika imaju različite dozvole pristupa, ovisno što im je potrebno
- jedna osoba može biti svaki podtip odjednom po potrebi
- ravnatelj može biti samo jedan, a odobrava ga administrator te ima pristup odobravanju daljnjih registriranih osoba i dodjeljivanju uloga te sve ostale mogućnosti administratora unutar škole
- satničar ima pristup promjeni rasporeda i objavljivanju obavijesti
- profesor ima pristup objavljivanju obavijesti i materijala te podacima svakog učenika kojem je razrednik
- **razrednik** - relacija koja označava kojem učeniku je profesor razrednik
- **predaje** - relacija koja označava predmete koje profesor predaje

UČENIK

Atribut	Tip podatka	Opis varijable
učenikID	INT	Jedinstveni identifikator učenika
razred	VARCHAR	Označava kojem razredu učenik pripada
škGod	VARCHAR	Označava godinu početka školovanja učenika
OIB	INT	Jedinstveni identifikator korisnika
smjer	VARCHAR	Koji smjer pohađa učenik

- primarni ključ: OIB
- strani ključ: OIB
- unique: učenikID
- predstavlja podtip registriranog korisnika koja pohađa školu
- sadrži još i razred, školsku godinu početka školovanja (škGod) i svoj jedinstveni ID (učenikID)
- **jelmao** (IZOSTANAK) - relacija koja označava je li i koliko je učenik imao izostanaka

IZOSTANAK

Atribut	Tip podatka	Opis varijable
izostanakOpis	VARCHAR	Opisuje uzrok izostanka
izostanakaDatum	DATE	Označava datum izostanka
izostanakStatus	VARCHAR	Označava je li izostanak opravdan ili ne
OIB	INT	Jedinstveni identifikator korisnika

- primarni ključ: OIB, izostanakDatum
- strani ključ: OIB
- veže se za učenika i označava svaki izostanak zasebno

PREDMET

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

Atribut	Tip podatka	Opis varijable
predmetID	INT	Jedinstveni identifikator predmeta
imePredmet	VARCHAR	Ime predmeta
brojSatova	INT	Broj sati koje predmet ima u tjednu
brojLab	INT	Broj laboratorija/vježbi koje predmet ima u tjednu
školaID	INT	Jedinstveni identifikator škole
smjer	VARCHAR	Na kojem smjeru se drži predmet

- primarni ključ: predmetID
- strani ključ: školaID
- predstavlja predmet koji se drži u školi
- **uKurikulumu** - relacija koja označava koje predmete škola drži

RASPORED

Atribut	Tip podatka	Opis varijable
datum	DATETIME	Datum i vrijeme kada se održava sat
razred	VARCHAR	Razred kojem je dodijeljen raspored
oznaka	VARCHAR	Jedinstveni identifikator prostorije
imePredmet	VARCHAR	Ime predmeta
školaID	INT	Jedinstveni identifikator škole

- predstavlja raspored koji se dodjeljuje učenicima i profesorima
- primarni ključ: školaID
- strani ključ: školaID

PROSTORIJA

Atribut	Tip podatka	Opis varijable
oznaka	VARCHAR	Jedinstveni identifikator
kapacitet	INT	Označava kapacitet prostorije
tipProstorije	VARCHAR	Označava je li prostorija učionica ili dvorana
školaID	INT	Jedinstveni identifikator škole

- predstavlja prostorije koje se nalaze u školi
- primarni ključ: oznaka
- strani ključ: školaID
- **ima (PROSTORIJA)** - relacija koja označava koje prostorije škola ima

REPOZITORIJ

Atribut	Tip podatka	Opis varijable
repID	INT	Jedinstveni identifikator repozitorija
imeRep	VARCHAR	Naziv repozitorija
metaData	VARCHAR	Dodatne informacije poput opisa ili tagova mapa
školaID	INT	Jedinstveni identifikator škole

- primarni ključ: repID
- strani ključ: školaID
- predstavlja repozitorij škole
- **unutar** - relacija koja označava da je svaki repozitorij zapravo folder na cloudu i može imati svoje subfoldere
- **imaOnline** - relacija koja označava da škola ima svoj repozitorij odnosno obavijesti i materijale spremljene u foldere na cloud

LINK

Atribut	Tip podatka	Opis varijable
autor	VARCHAR	Predstavlja korisnika koji je objavio materijale
brojPregleda	INT	Označava koliko se puta pristupilo dokumentu
datumObjave	DATETIME	Datum i vrijeme objave dokumenta
linkTekst	VARCHAR	Jedinstveni identifikator linka
repID	INT	Jedinstveni identifikator repozitorija

- primarni ključ: linkTekst
- strani ključ: repID
- predstavlja sve što se može nalaziti u repozitoriju odnosno DATOTEKA ili OBAVIJEST
- **sadrži** - relacija koja označava u kojem se folderu nalazi LINK

OBAVIJEST

Atribut	Tip podatka	Opis varijable
naslov	VARCHAR	Naslov obavijesti
tekst	VARCHAR	Tekst obavijesti
linkTekst	VARCHAR	Jedinstveni identifikator linka

- primarni ključ: linkTekst
- strani ključ: linkTekst
- predstavlja podtip linka koji je obavijest

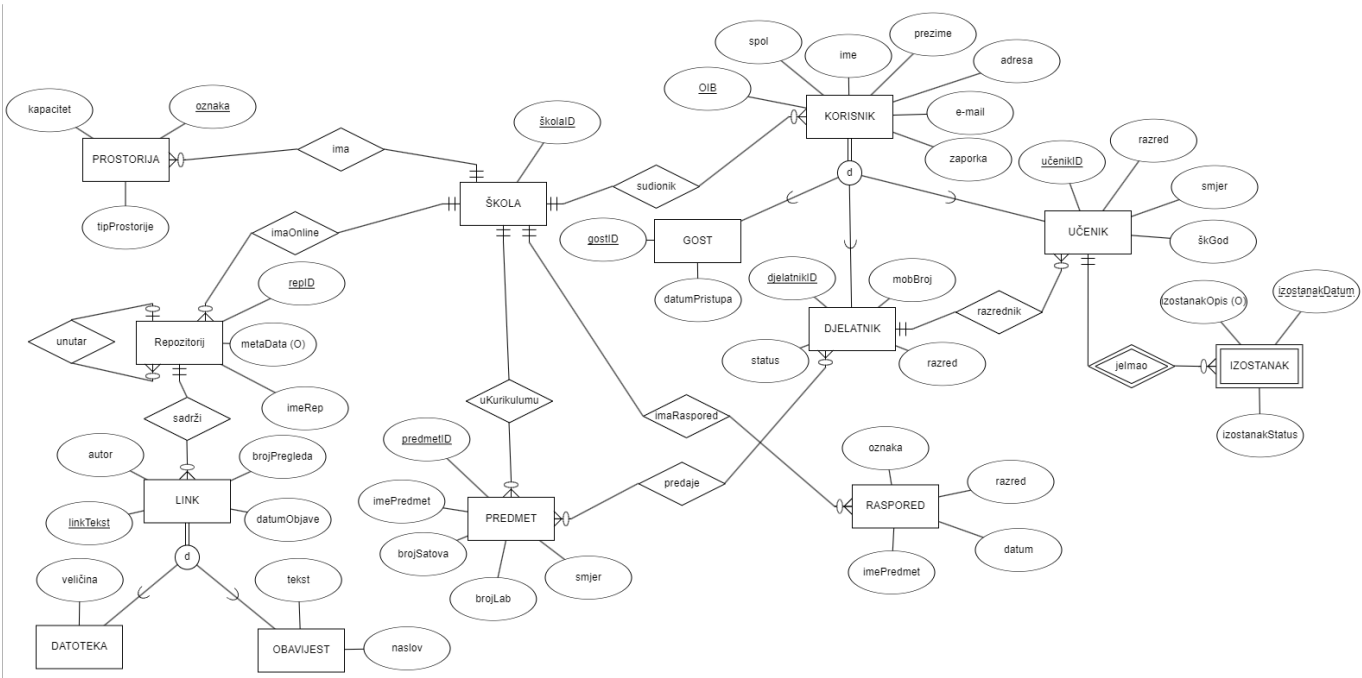
DATOTEKA

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

Atribut	Tip podatka	Opis varijable
veličina	INT	Veličina datoteke u MB
linkTekst	VARCHAR	Jedinstveni identifikator linka

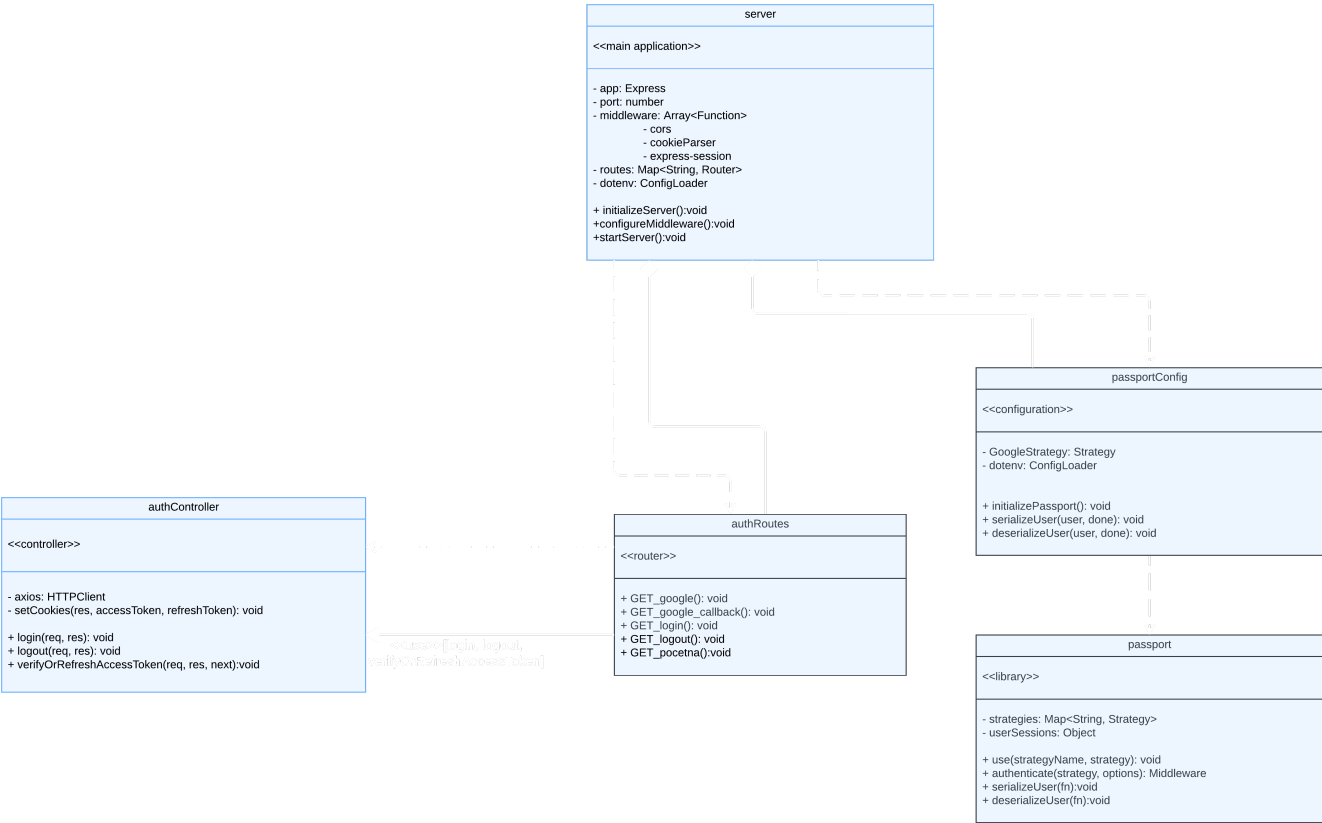
- primarni ključ: linkTekst
- strani ključ: linkTekst
- predstavlja podtip linka koji je datoteka

Dijagram baze podataka

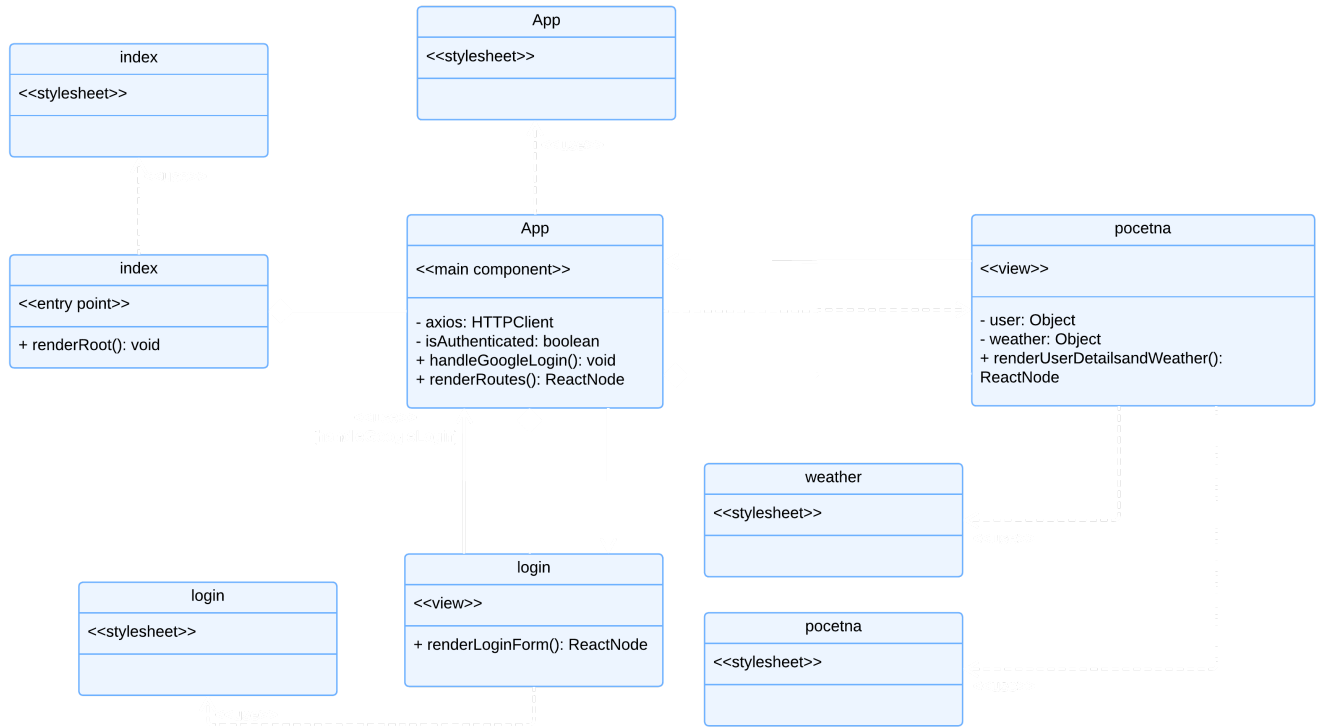


Dijagram razreda

Backend dijagram razreda:



Frontend dijagram razreda:



Dijagram stanja

Dijagram aktivnosti

Dijagram komponenti

Korištene tehnologije i alati

Ispitivanje programskog rješenja

Ispitivanje komponenti

Ispitivanje sustava

Dijagram razmještaja

Upute za puštanje u pogon

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri pisanju projekta.

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/](http://www.ece.rutgers.edu/~marsic/books/SE) books/SE
5. The Unified Modeling Language, <https://www.uml-diagrams.org/>
6. Astah Community, <http://astah.net/editions/uml-new>
7. Tehnička škola Ruđer Bošković, <https://www.tsrh.hr/>

Dnevnik sastajanja

1. sastanak

- Datum: 14. listopada 2024.
- Prisustvovali: J.Gazdek, S.Horvatić, J.Jovanović, D.Popović, F.Vuletić-Antić
- Teme sastanka:
 - i. Upoznavanje s temom - diskutiranje ideja, odabir imena
 - ii. Raspored poslova - odluka tko radi na kojem dijelu aplikacije
 - iii. Dogovorena komunikacija - izrada Whatsapp grupe i Discord kanala

2. sastanak

- Datum: 20. listopada 2024.
- Prisustvovali: J.Gazdek, J.Jovanović
- Teme sastanka:
 - i. Izrada GitHub repozitorija - dogovor za pisanje dokumentacije i postavljanje GitHub repozitorija

3. sastanak

- Datum: 28. listopada 2024.
- Prisustvovali: J.Čižić, J.Gazdek, S.Horvatić, J.Jovanović, R.Markov, D.Popović, F.Vuletić-Antić
- Teme sastanka:
 - i. Razrada analize zahtjeva aplikacije - diskusija o funkcionalnim i ostalim zahtjevima, dogovor o načinu implementacije i alatima za izradu
 - ii. Dogovor za izradu Figma sučelja za prikaz izgleda aplikacije

4. sastanak

- Datum: 29. listopada 2024.
- Prisustvovali: J.Čižić, J.Gazdek, S.Horvatić, J.Jovanović, R.Markov, D.Popović, F.Vuletić-Antić
- Teme sastanka:
 - i. Detaljnija podjela poslova - dogovor za izradu baze podataka, unapređivanje UML dijagrama
 - ii. Dogovoren plan rada - razrada i rokovi za dodijeljene zadatke

5. sastanak:

- Datum: 6. studenog 2024.
- Prisustvovali: J.Čižić, J.Gazdek, S.Horvatić, J.Jovanović, R.Markov, D.Popović, F.Vuletić-Antić
- Teme sastanka:
 - i. Dogovor za frontend i backend - početak implementacije login stranice i naslovne stranice, implementacija logina

6. sastanak:

- Datum: 11. studenog 2024.
- Prisustvovali: J.Čižić, J.Gazdek, S.Horvatić, J.Jovanović, R.Markov, D.Popović, F.Vuletić-Antić
- Teme sastanka:
 - i. Dogovor za arhitekturu sustava - raspravljanje o mogućim rješenjima i podjela posla za pisanje

dokumentacije

ii. Finalni dogovori - rasprava o detaljima koje treba popraviti prije 1. revizije

Plan rada

Zadatak	14.10-21.10	21.10-28.10	28.10-3.11	3.11-11.11	11.11-15.11
Planiranje	Razumjeti ideju projekta i postaviti osnovne ciljeve, podjela poslova	Razraditi zahtjeve i implementaciju sustava, pisati potrebnu dokumentaciju	Nadopunjavanje dokumentacije, detaljnija razrada zahtjeva, izrada baze podataka	Izrada Figma sučelja i razrada baze podataka	Finalni dogovori za izradu frontenda i backenda te deployment aplikacije
Dokumentiranje	Napraviti predložak, napisati opis projektnog zadatka	Napisati analizu zahtjeva i specifikacije zahtjeva sustava	Usavršiti zahtjeve i UML dijagrame	Dodati arhitekturu i dizajn sustava	Razrada arhitekture i dovršavanje dokumentacije za 1. reviziju
Izrada baze podataka			Započeti izradu baze podataka	Nadopuna baze podataka po potrebi	
Frontend				Napraviti login i homepage	Deployment frontenda
Backend				Implementirati login putem sustava za autentifikaciju	Deployment backenda
Deployment					Omogućiti deployment aplikacije

Tablica aktivnosti

Kontinuirano osvježavanje

Napomena: Doprinosi u aktivnostima treba navesti u satima po članovima grupe po aktivnosti.

	Juraj Čižić	Jana Gazdek	Siniša Horvatić	Jana Jovanović	Roko Markov	Damjan Popović	Filip Vuletić- Antić
Upravljanje projektom		5		0.5			
Opis projektnog zadatka		2					
Funkcionalni zahtjevi		3					
Opis pojedinih obrazaca		4					
Dijagram obrazaca		3					
Sekvencijski dijagrami		4					
Opis ostalih zahtjeva		2					
Arhitektura i dizajn sustava	2	1	0.5	2	2.5	2.5	2
Baze podataka					5	5	
Dijagram razreda	1		1				
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja	1						
Dijagram razmještaja							
Upute za puštanje u pogon							
Dnevnik sastajanja		0.5					
Zaključak i budući rad							
Popis literature		0.5					
Dodatne stavke kako smo podijelili izradu aplikacije							
Izrada baze podataka							
Spajanje s bazom podataka							

	Juraj Čižić	Jana Gazdek	Siniša Horvatić	Jana Jovanović	Roko Markov	Damjan Popović	Filip Vuletić- Antić
Backend	10						3
Frontend			9	2			
Figma			1	3			

Dijagram pregleda promjena

Prenijeti dijagram pregleda promjena nad datotekama projekta. Potrebno je na kraju projekta generirane grafove s githuba prenijeti u ovo poglavlje dokumentacije. Dijagrami za vlastiti projekt se mogu preuzeti s github.com stranice, u izborniku Repository, pritiskom na stavku Contributors.

Ključni izazovi i rješenja

- Zaključno
- Opis izazova
- Rješenja

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Jana Gazdek	27.10.2024
0.2	Dodana analiza zahtjeva i obrasci uporabe. Dodane reference.	Jana Gazdek	28.10.2024
0.3	Dodani sekvencijski dijagrami i dijagrami obrazaca uporabe.	Jana Gazdek	29.10.2024
0.4	Napravljena baza podataka, dodan opis tablica i dijagram baze podataka.	Damjan Popović, Roko Markov	5.11.2024
0.5	Dodani novi sekvencijski dijagrami i dijagrami obrazaca uporabe.	Jana Gazdek	5.11.2024
0.6	Napravljen predložak za arhitekturu i dizajn sustava.	Jana Gazdek	11.11.2024
0.7	Dodani opis i obrazloženje arhitektura sustava. Dodana organizacija sustava i aplikacije, dodani dijagrami razreda.	Juraj Čičić, Jana Jovanović, Roko Markov, Damjan Popović, Filip Vuletić-Antić	12.11.2024
0.8	Popravljeni funkcionalni i ostali zahtjevi. Popravljeni sekvencijski dijagrami i dijagrami obrazaca uporabe.	Jana Gazdek	13.11.2024
0.9	Revizija cijele dokumentacije.	Jana Gazdek	14.11.2024
0.10	Dodani ostali dijagrami razreda.	Siniša Horvatić	15.11.2024
1.0	Dovršena verzija dokumentacije s dijelovima bitnim za 1. reviziju.	Jana Gazdek	15.11.2024