

Programsko inženjerstvo

Ak. god. 2023./2024.

CookBooked

Dokumentacija, Rev. 1

Grupa: *CodeCooks*

Voditelj: *Antonio Hohnjec*

Datum predaje: *<dan>. <mjesec>. <godina>.*

Nastavnik: *Nikolina Frid*

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	4
2.1	Opis ideje aplikacije	4
3	Specifikacija programske potpore	7
3.1	Funkcionalni zahtjevi	7
3.1.1	Obrasci uporabe	8
3.1.2	Sekvencijski dijagrami	12
3.2	Ostali zahtjevi	16
4	Arhitektura i dizajn sustava	17
4.1	Baza podataka	19
4.1.1	Opis tablica	20
4.1.2	Dijagram baze podataka	26
4.2	Dijagram razreda	27
4.3	Dijagram stanja	28
4.4	Dijagram aktivnosti	29
4.5	Dijagram komponenti	30
5	Implementacija i korisničko sučelje	31
5.1	Korištene tehnologije i alati	31
5.2	Ispitivanje programskog rješenja	32
5.2.1	Ispitivanje komponenti	32
5.2.2	Ispitivanje sustava	32
5.3	Dijagram razmještaja	33
5.4	Upute za puštanje u pogon	34
6	Zaključak i budući rad	35
	Popis literature	36

Indeks slika i dijagrama	37
Dodatak: Prikaz aktivnosti grupe	38

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Učitani predložak dokumentacije.	Antonio Hohnjec	19.10.2023.
0.2	Dopunjeni nazivi i sudionici projekta. Nadopunjeno poglavlje "2. Opis projektnog zadatka"	Antonio Hohnjec	25.10.2023.
0.3	Dodani <i>Use Case</i> dijagrami i sekvencijski dijagrami, nefunkcionalni zahtjevi i ostali zahtjevi te arhitektura i dizajn sustava sa bazom podataka i prikladnim tablicama te dijagramom baze podataka.	Benjamin Zrakić, Armis Kantarević, Luka Ivčević, Filip Vrbić, Marko Pavić	30.10.2023.
0.6	Upis poglavlja Arhitektura i dizajn sustava. Dodani dijagrami razreda za prvu predaju	Armis Kantarević, Antonio Zglavnik	9.11.2023.

2. Opis projektnog zadatka

CookBooked je projekt u kojem je potrebno razviti web aplikaciju koja omogućava korisnicima razmjenu recepata za kuhanje i pečenje kolača te povezivanje s autorima recepata.

Neregistrirani korisnici mogu samo pregledavati recepte temeljem kategorija, vrsta kuhinje ili specifičnih sastojaka. Za pristup svim ostalim mogućnostima platforme, korisnici se moraju registrirati s važećom adresom e-pošte.

Autori recepata imaju opciju komunikacije s ostalim korisnicima vezano za svoje recepte, kao npr. razmjenu poruka, čavljanje ili video pozive. Ove značajke omogućuju korisnicima da se povežu s autorima recepata, ali su dostupne samo registriranim korisnicima. Autori recepata također mogu postaviti termine kada su dostupni za komunikaciju (npr. određene sate ili dane).

Registrirani korisnici mogu označavati, komentirati i spremati recepte za buduću referencu. Korisnici mogu pratiti svoje omiljene autore recepata kako bi primili obavijest o novim receptima.

Registrirani korisnici imaju javne profile na kojima prikazuju svoje objavljene recepte, pratitelj i autore koje prate. Također imaju privatne profile gdje mogu upravljati osobnim informacijama, postavkama komunikacije i obavijestima za poruke i aktivnosti povezane s receptima.

Platformu održavaju sistemski administratori koji mogu upravljati korisnicima, mijenjati kategorije recepata ili brisati recepte.

2.1 Opis ideje aplikacije

CookBooked je aplikacija koja omogućava korisnicima razmjenu recepata za kuhanje i pečenje kolača te povezivanje s autorima recepata. Sam potencijal ove aplikacije

cije leži u dobro razvijenom te interaktivnom sučelju koje omogućava registriranim korisnicima lako objavljivanje svojih recepata, dok u isto vrijeme omogućava nesmetano korištenje i pregled samih tih recepata bez mogućnosti objave.

Danas već postoje neke stranice za objavu recepata kao što su Coolinarka, ReciPeci, Zdrave navike i brojne druge. Iako su one već dobrim djelom razvijene te koriste brojne alate, aplikacija CookBooked omogućava veću personalizaciju, odabir jelovnika za pojedini dan te kategorizira jela i namirnice na načine koji su puno bliži i lakši za snalaženje svakom korisniku.



Slika 2.1: Coolinarka



Slika 2.2: ReciPeci

U prijašnjem odlomku spomenuli smo prednost odlične kategorizacije ove aplikacije pa tako gdje se ta prednost pokazuje jest upravo tako što imamo kategorije hrane za mlade, kategorije hrane za osobe sa bolestima te za starije osobe kojima

odgovara lagana prehrana.

Tako naša aplikacija zahvaća razne kategorije ljudi kao što su:

- *mlađe osobe koje tek uče kuhati*
- *osobe koje ne stignu raditi duge pripreme*
- *starije osobe koje ne mogu pripremati i konzumirati svu hranu*
- *odrasli koji žele naučiti kuhati*
- *osobe koje žele izraditi kompliciranije stvari sa nešto manje kulinarskog iskustva*

Sama aplikacija od razvoja pa do konačne aplikacije i poslije je vrlo lako prilagodljiva. Osim izvornih alata pisanja koda, dijelove aplikacije lako je prilagoditi u smislu izgleda, dodavanja novih kategorija, dodatnih opcija privatnog profila. Ako želimo znati zašto, to je jednostavno. Cijela aplikacija u kodu je strukturirana na način da se prate i učitavaju određene strukture sa podacima gdje je tada u slučaju bilo kakve modifikacije potrebno promjenu učiniti na samo jednom mjestu.

Što se samih promjena tiče, aplikacija se sastoji od više verzija. Kod razvoja aplikacije u pogon se pušta prvo prototip koji sadrži osnovnu funkcionalnost zbog potrebe provjere ispravnosti aplikacije. U sljedećim verzijama aplikacija se nadograđuje profilima korisnika, poslovnim i privatnim profilima pa se nadalje dodaje i kategorije, izbornici, i slično te u konačnici mogućnost komunikacije između autora i recenzije njihovih recepata.

Nakon završetka postavljenih ciljeva pri izradi aplikacije, zbog same strukture i čitljivosti pri pisanju, aplikaciju je u naknadnim verzijama moguće nadograditi sa brojnim drugim značajkama.

Primjer takvih značajki:

- *komunikacija unutar aplikacije putem videopoziva*
- *AI pomoć pri odabiru recepta i kategorije*
- *AI stvaranje jelovnika prilagođenog za dan i posebne prilike*
- *Stvaranje odjela za događaje izrade i objave novih recepata*
- *Tečajevi za kuhanje putem videa i videopoziva*

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

Navesti **dionike** koji imaju **interes u ovom sustavu** ili **su nositelji odgovornosti**. To su prije svega korisnici, ali i administratori sustava, naručitelji, razvojni tim.

Navesti **aktore** koji izravno **koriste** ili **komuniciraju sa sustavom**. Oni mogu imati inicijatorsku ulogu, tj. započinju određene procese u sustavu ili samo sudioničku ulogu, tj. obavljaju određeni posao. Za svakog aktora navesti funkcionalne zahtjeve koji se na njega odnose.

Dionici:

1. Dionik 1
2. Dionik 2
3. ...

Aktori i njihovi funkcionalni zahtjevi:

1. Aktor 1 (inicijator) može:
 - (a) funkcionalnost 1
 - (b) funkcionalnost 2
 - i. podfunkcionalnost 1
 - ii. podfunkcionalnost 2
 - (c) funkcionalnost 3
2. Aktor 2 (sudionik) može:
 - (a) funkcionalnost 1
 - (b) funkcionalnost 2

3.1.1 Obrasci uporabe

UC1 - Pregled recepata

- Glavni sudionik: Neregistiran korisnik, Registriran korisnik
- Cilj: Pregled recepata temeljem kategorija
- Sudionici: Baza podataka
- Preduvjet: -
- Opis osnovnog tijeka:
 1. Korisnik otvori platformu
 2. Ponuđene su mu kategorije, vrste kuhinje, specifični sastojci
 3. Korisnik odabire kategoriju i prikazuju mu se recepti
- Opis mogućih odstupanja:
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

UC2 - Registracija

- Glavni sudionik: Neregistrirani korisnik
- Cilj: Stvaranje računa za pristup ostalim značajkama sustava
- Sudionici: Baza podataka
- Preduvjet:-
- Opis osnovnog tijeka:
 1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik prima obavijest o uspješnoj registraciji
- Opis mogućih odstupanja:
 - 2.a Odabir već zauzetog korisničkog imena i/ili e-maila, unos korisničkog podatka u nedozvoljenom formatu ili pružanje neispravnoga e-maila
 1. Sustav obavještava korisnika o neuspjelom upisu i vraća ga na stranicu za registraciju
 2. Korisnik mijenja potrebne podatke te završava unos ili odustaje od registracije

UC3 - Prijava u sustav

- **Glavni sudionik:**
- **Cilj:**
- **Sudionici:**
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- **Opis mogućih odstupanja:**
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

UC4 - Objava recepata

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Objava svog recepta na stranici
- **Sudionici:**
- **Preduvjet:** Registracija i prijava korisnika
- **Opis osnovnog tijeka:**
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- **Opis mogućih odstupanja:**
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

UC5 - Razmjena poruka

- **Glavni sudionik:**
- **Cilj:**
- **Sudionici:**
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- **Opis mogućih odstupanja:**
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

UC6 - Čavrljanje

- **Glavni sudionik:**
- **Cilj:**
- **Sudionici:**
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- **Opis mogućih odstupanja:**
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

UC7 - Videopoziv

- **Glavni sudionik:**
- **Cilj:**
- **Sudionici:**
- **Preduvjet:**
- **Opis osnovnog tijeka:**
 - 1.
 - 2.
 - 3.
 - 4.
 - 5.
- **Opis mogućih odstupanja:**
 - 2.a
 - 1.
 - 2.
 - 2.b
 - 3.a

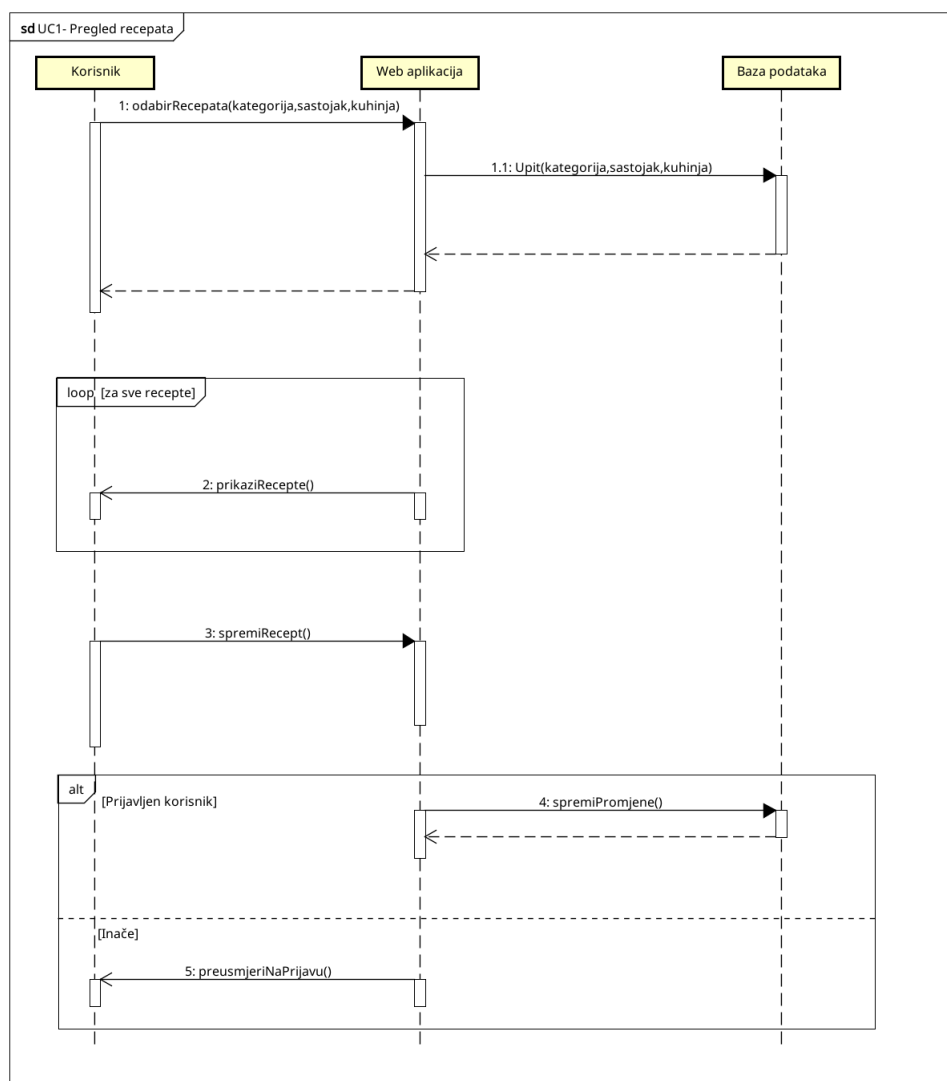
Dijagrami obrazaca uporabe

Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.

3.1.2 Sekvencijski dijagrami

Obrazac uporabe UC1-Pregled recepata

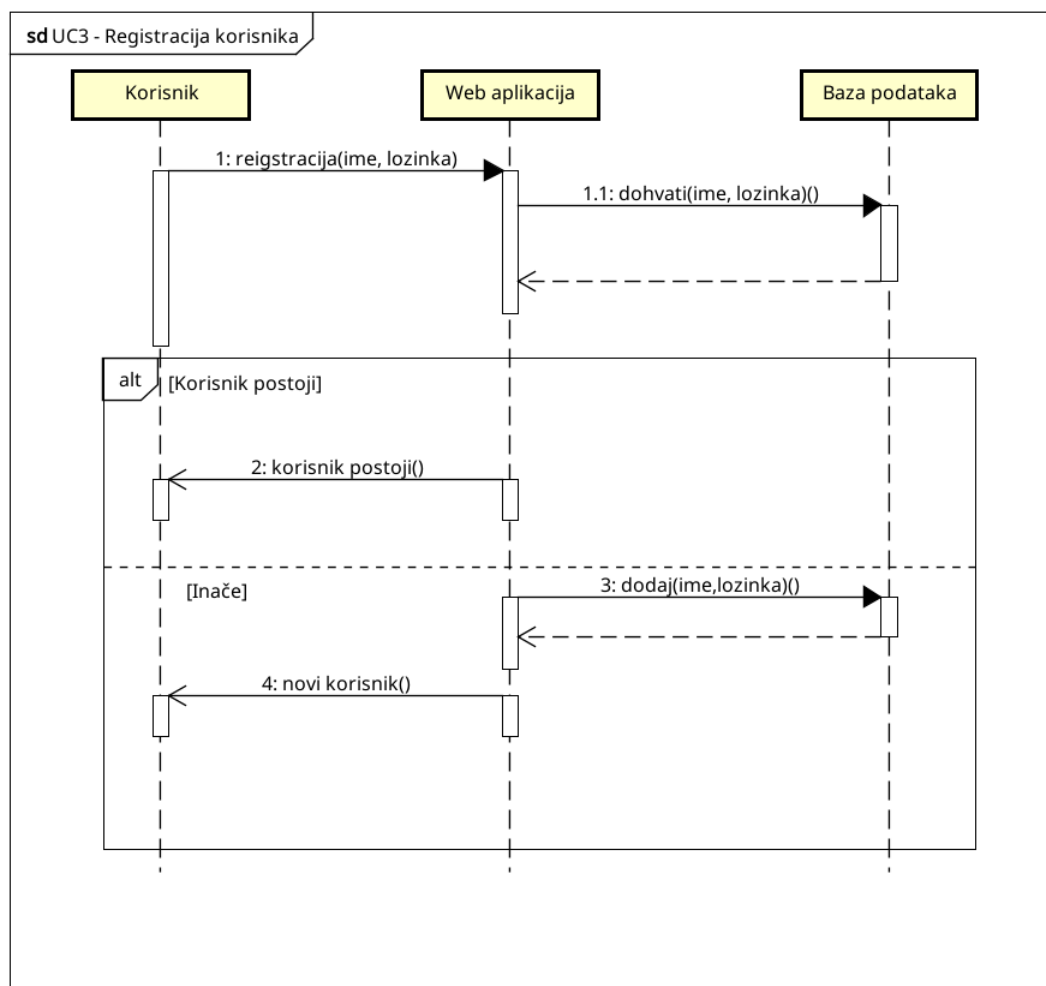
Korisnik šalje zahtjev za prikaz recepata po kategorijama, sastojcima i/ili kuhinjama kojim pripadaju. Poslužitelj dohvaća recepte koji zadovoljavaju uvjete i prikazuje ih korisniku. Korisnik sada može spremiti recepte, ako je prijavljen to se provodi, ako nije preusmjeri ga se na stranicu za prijavu.



Slika 3.1: Sekvencijski dijagram za UC1

Obrazac uporabe UC3-Registracija

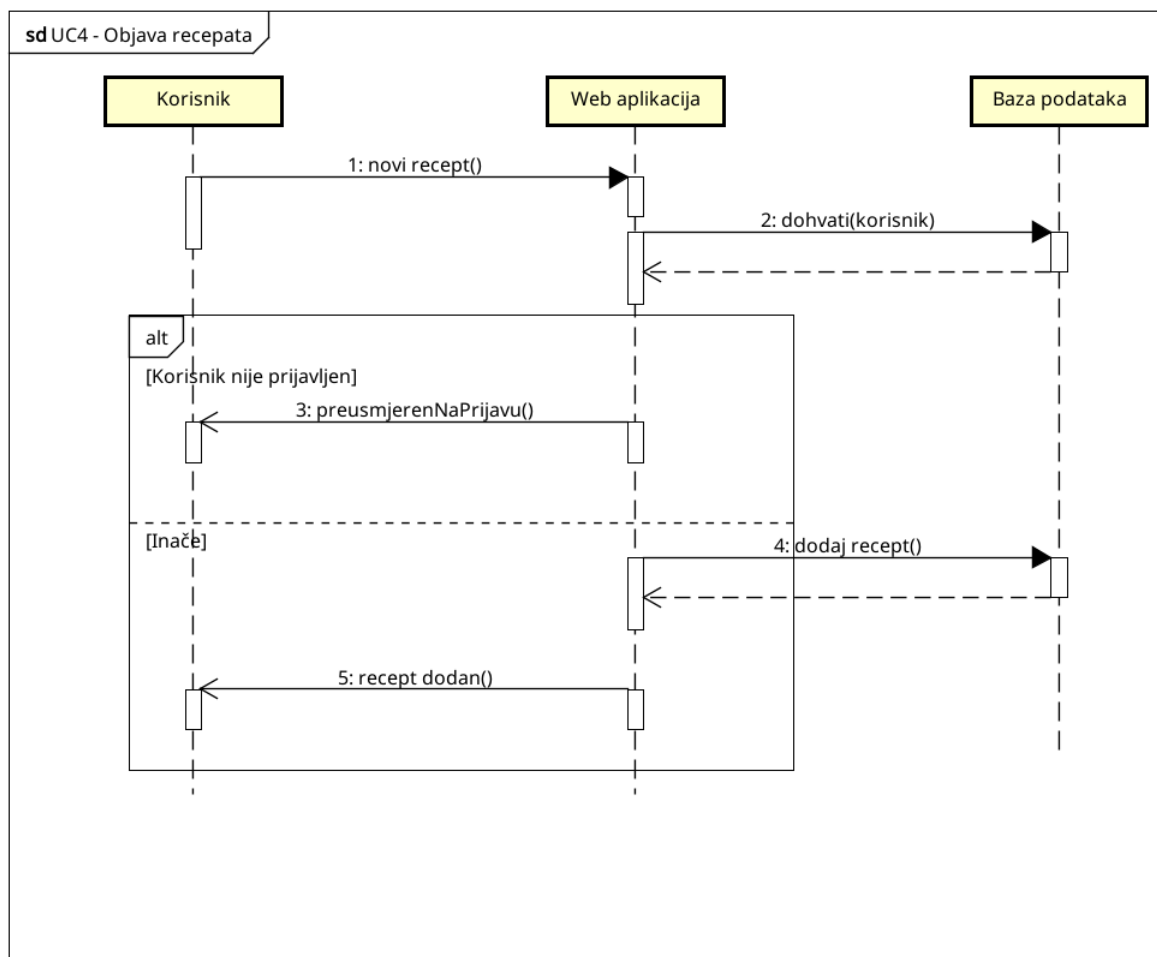
Korisnik se registrira s korisničkim imenom i lozinkom. Ako takav korisnik već postoji, korisniku se ispisiuje greška. Inače, korisnik se uspješno registrirao i to se bilježi u bazi podataka.



Slika 3.2: Sekvencijski dijagram za UC3

Obrazac uporabe UC4-Prijava Korisnika

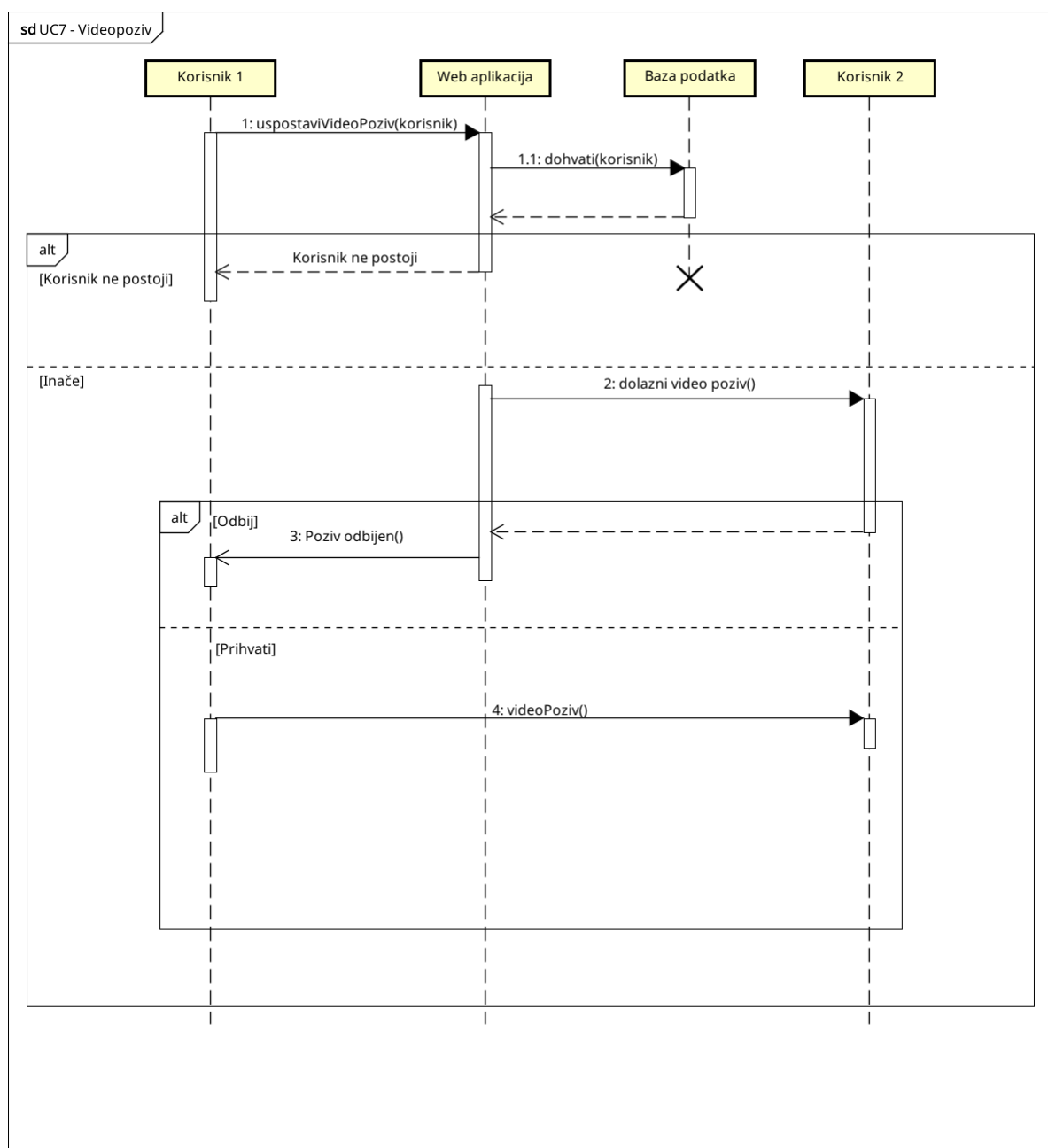
Korisnik pokušava objaviti novi recept. Ako nije prijavljen, preusmjeri ga se na prijavu. Inače, recept se dodaje u bazu podataka i o tome se obavijesti korisnik.



Slika 3.3: Sekvencijski dijagram za UC4

Obrazac uporabe UC8-Videopoziv

Korisnik zatraži video poziv s drugim korisnikom. Ako drugi korisnik ne postoji, poziv se ne uspostavlja i o tome se obavijesti prvog korisnika. Inače se šalje zahtjev za video pozivom drugom korisniku, koji ga može odbiti ili prihvatiti. Ako odbije poziv se ne uspostavlja i o tome se obavijesti prvi korisnik. Ako prihvati uspostavi se video poziv između dva korisnika.



Slika 3.4: Sekvencijski dijagram za UC7

3.2 Ostali zahtjevi

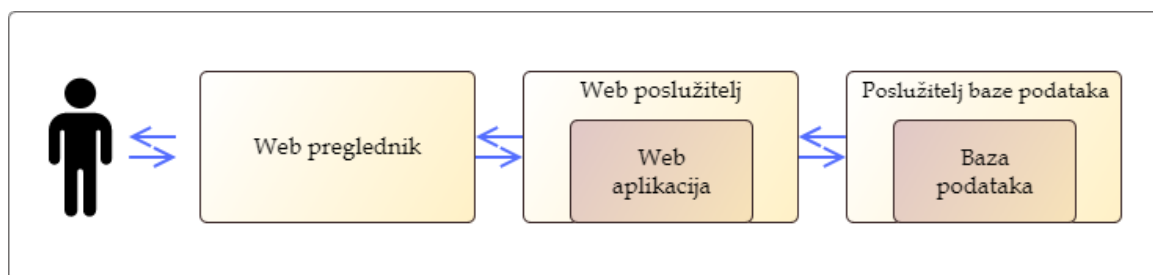
Nefunkcionalne zahtjeve koji će se navesti u nastavku teksta pojašnjavaju dodatne zahtjeve koje web aplikacija treba koristiti ili već koristi.

- Sustav treba biti implementiran kao web aplikacija koristeći objektno-orijentirane paradigme i norme kako bi se omogućilo ponovno korištenje dijelova koda/modula.
- Nepravilno korištenje web aplikacije ne smije rezultirati padom sustava, odnosno potrebno je usmjeriti korisnika na pravilno korištenje informacionim obavještenjima.
- U sustavu treba biti omogućen rad i korištenje web aplikacije od strane više korisnika (cca. 50).
- Sustav treba omogućiti komuniciranje korisnika s ostalim korisnicima putem tekstualnog chat-a i video poziva.
- Sustav mora podržati dijakritičke znakove hrvatskoj jezika, dakle treba podržati hrvatsku abecedu. Omogućit će se i promjena jezika s hrvatskog na engleski jezik.
- Hrvatski jezik je zadani jezik unutar web aplikacije.
- Sustav mora imati intuitivno sučelje koje neće stvarati nedoumice kod korisnika odnosno web-aplikacija treba biti jednostavna za korištenje.
- Konekcija s bazom podataka mora imati brz odziv. Bilo kakav pokušaj neovlaštenog pristupa informacijama u bazi podataka potrebno je spriječiti i istu adekvatno zaštititi.
- Web aplikacija koristi proces kriptiranja lozinki za prijavu korisnika koji se tako u hash-ovima spremaju u bazu podataka.
- Sustav mora omogućiti korištenje određenih funkcionalnosti samo prijavljenim/registriranim korisnicima.
- Učitavanje početne stranice web aplikacije ne smije trajati duže od 5 sekundi.
- Pristup sustavu i razmjena podataka se vrši HTTPS protokolom.
- Pri razvoju web aplikacije koristi se React Native i Spring framework u Java programskom jeziku.

4. Arhitektura i dizajn sustava

Arhitektura sustava je bazirana na tri komponente koje komuniciraju jedna s drugom. Odnosno, sustav je podijeljen u tri dolje navedena sloja, a koji se mogu prikazati slikom ispod.

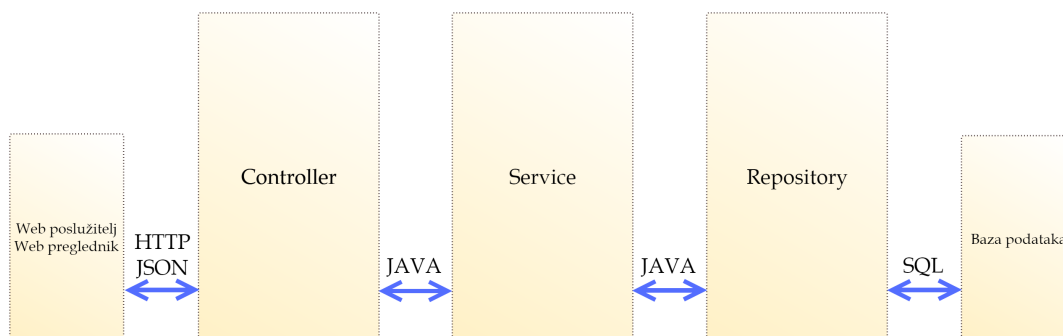
- *Web poslužitelj*
- *Web aplikacija*
- *Baza podataka*



Slika 4.1: Arhitektura sustava

Web preglednik omogućava korisniku prikaz web stranice koja pruža određene funkcionalnosti. Omogućava prikaz web stranice (prikaz videa, slika i ostalog multimedijalnog sadržaja) onako kako je definirana u datotekama, dakle omogućuje interpretiranje koda u koristan oblik "običnom" korisniku. Putem web preglednika korisnik šalje zahtjev za željenu radnju koja se onda proslijedi idućim komponentama na obradu, te nakon obrade opet se prikazuju u vidljivom obliku kao vid povratne informacije.

Web poslužitelj je ključna komponenta u obradi korisničkih zahtjeva. Dakle, to je **središnji dio** aplikacije koji omogućava komunikaciju korisnika s aplikacijom. U središnjem sloju se odvijaju procesi koji su zaslužni za komunikaciju s bazom podataka ukoliko je to potrebno, a to se odvija putem kontrolera, servisa i pristupu bazi podataka. U ovom sloju za komunikaciju koristi se Java programski jezik. Odnos komponenti predstavljen je slikom ispod.

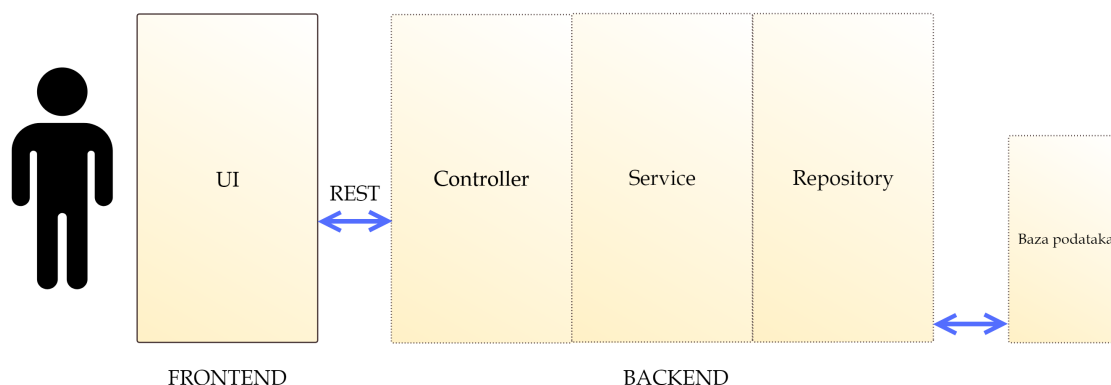


Slika 4.2: Arhitektura sustava backend

Za izradu ovakve web aplikacije kako bi se pokrile navedene specifične funkcionalnosti koristit će se Spring framework za Javu, te React za prikaz na web pregledniku u radnom okruženju IntelliJIDEA. Specifično za Spring framework, koristit će se navedeni tip arhitekture kao što je naveden slikom iznad, jer će se putem metoda JpaRepository moći upravljati zahtjevima korisnika koji se vežu za upite u bazi podataka. Ukratko, obavljaju se operacije nad **bazom podataka H2**.

Kontroler upravlja korisničkim zahtjevima i prosljeđuje ih dalje prema **Service** gdje se obavlja logika nad upitima i zahtjevima. Service komunicira s **bazom podataka** preko JpaRepository.

UI će biti povezan sa ostalim dijelovima web aplikacije putem **REST servisa**, odnosno REST servis komunicira s navedena tri sloja: Controller, Service i Repository. Takva vrsta komunikacije je predstavljena idućom slikom. UI je izveden uz pomoć **React** framework-a koji se koristi komponentama za organizaciju prikaza.



Slika 4.3: Arhitektura sustava backend i frontend

4.1 Baza podataka

Kao rješenje za web aplikaciju koju izrađujemo, odlučili smo se na relacijski model baze podataka jer nam ona omogućava precizno oblikovanje i modeliranje elemenata iz stvarnog svijeta. Međusobni odnos tablica se zasniva na relaciji dok se svaka tablica sastoji od naziva entiteta te njegovih pripadajućih atributa koji opisuju dani entitet. Ovakav tip baze podataka u ovom slučaju nam omogućava brzo spremanje, dohvat i izmjenu (uređivanje ili brisanje) podataka koji cirkulišu u web aplikaciji. Baza podataka koja je spremna za izradu ove web aplikacije se sastoji od sljedećih elemenata/tablica:

- Korisnik
- Recept
- Video
- Objava
- Kategorija
- ReceptKategorije
- ReceptSastojci
- Sastojak
- Komentar
- OmiljeniAutor
- OznačenRecept
- SpremljenRecept
- VrstaKuhinja
- Obavijesti

4.1.1 Opis tablica

Korisnik Entitet Korisnik služi za evidenciju podataka o korisnicima koji koriste web aplikaciju. Atributi koji ga opisuju su korisnickoIme, lozinkaKorisnik, imeKorisnik, prezimeKorisnik, brojTelefona, emailKorisnik, razinaOvlasti i Dostupan što označava kada je dostupan za komunikaciju s drugim korisnicima unutar web aplikacije. Entitet Korisnik je u *One-To-Many* vezi s entitetom Recept. Entitet Korisnik je i u *Many-To-Many* vezi s entitetom OmiljeniAutor koji bilježi korisnike koje ovaj korisnik prati. Također, ovaj entitet je u *Many-To-Many* vezi s entitetima SpremljenRecept, OznačenRecept, a u *One-To-Many* vezi s entitetima Objava i Komentar.

Korisnik		
IDKorisnik	INT	jedinstveni identifikator korisnika
KorisnickoIme	VARCHAR	korisničko ime
LozinkaKorisnik	VARCHAR	spremljena hash lozinka za pristup
ImeKorisnik	VARCHAR	ime korisnika
PrezimeKorisnik	VARCHAR	prezime korisnika
BrojTelefona	VARCHAR	broj telefona korisnika
EmailKorisnik	VARCHAR	e-mail adresa korisnika
RazinaOvlasti	VARCHAR	razina ovlasti u aplikaciji
Dostupan	TIME	vrijeme dostupnosti

Komentar Entitet Komentar služi za evidenciju komentara koji korisnici međusobno dijele u web aplikaciji. Opisuje ga atributi IDKomentar, IDKorisnik, IDObjava, te NaslovKomentar, SadržajKomentar i DatumKomentar. Ovaj entitet je u *Many-To-One* vezi s entitetom Korisnik, te u istoj vezi s entitetom Objava.

Komentar		
IDKomentar	INT	jedinstveni identifikator komentara
IDKorisnik	INT	jedinstveni identifikator korisnika
IDObjava	INT	jedinstveni identifikator objave
NaslovKomentar	VARCHAR	naziv komentara
OpisKomentar	VARCHAR	sadržaj komentara
DatumKomentar	DATE	datum komentara

OmiljeniAutor Entitet OmiljeniAutor služi za evidenciju korisnika koje jedan korisnik zaprati. Opisan je sljedećim atributima: IDKorisnik i IDAutor. U *Many-To-Many* je vezi s entitetom Korisnik unutar web aplikacije.

OmiljeniAutor		
IDKorisnik	INT	jedinstveni identifikator pratitelja
IDAutor	INT	ID korisnika koji je zapaćen

Recept Entitet Recept služi za evidenciju podataka o receptima koji cirkulišu i nastaju u web aplikaciji. Opisan je atributima: IDRecept, NazivRecept, IDKategorija, IDSastojak, IDVrstaKuhinja, PripremaRecept, VrijemeKuhanja, IDOznaka, SlikaRecept i IDVideoRecept. Entitet Recept je u *Many-To-One* vezi s entitetom Korisnik, a u *Many-To-Many* vezi s entitetom Sastojak, te u *Many-To-Many* vezi s Kategorija, a *One-To-One* s VrstaKuhinja. Također je u *One-To-One* vezi s entitetom Objava.

Recept		
IDRecept	INT	jedinstveni identifikator recepta
NazivRecept	VARCHAR	naziv recepta
IDKategorija	INT	ID kategorije kojoj recept pripada
IDSastojak	INT	ID sastojka u receptu
IDVrstaKuhinja	INT	ID vrste kuhinje recepta
PripremaRecept	VARCHAR	Opis pripreme recepta
VrijemeKuhanja	TIME	Vrijeme potrebno za pripremu recepta
IDOznaka	INT	ID oznake recepta
SlikaRecept	VARCHAR	Slika recepta
IDVideoRecept	INT	ID Video recepta

Video Entitet Video služi za evidenciju videa koji se objavljuju uz recepte. Opisani su sljedećim atributima: IDVideo, NazivVideo i TrajanjeVideo što omogućuje pretragu recepata po dužini pripremanja. U *One-To-One* je vezi s entitetom Recept unutar web aplikacije.

Video		
IDVideo	INT	jedinstveni identifikator videa
NazivVideo	VARCHAR	naziv videa
TrajanjeVideo	TIME	trajanje videa

Objava Entitet Objava služi za evidenciju objava recepata koje korisnici objavljuju unutar web aplikacije. Opisani su atributima: IDObjava, IDKorisnik, IDRecept i DatumObjava. U *Many-To-One* vezi je s entitetom Korisnik, te u *Many-To-Many* vezi s entitetom Komentar i u *One-To-One* s entitetom Recept.

Objava		
IDObjava	INT	jedinstveni identifikator objave

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Objava		
IDKorisnik	INT	jedinstveni identifikator korisnika
IDRecept	INT	jedinstveni identifikator recepta
DatumObjava	DATE	datum objave

OznačenRecept Entitet OznačenRecept služi za evidenciju označenih recepata od strane korisnika. Opisan je atributima: IDRecept i IDKorisnik. U *Many-To-Many* vezi je s entitetom Korisnik.

OznačenRecept		
IDRecept	INT	jedinstveni identifikator recepta
IDKorisnik	INT	ID korisnika koji je označio recept

SpremljenRecept Entitet SpremljenRecept za evidenciju spremljenih recepata od strane korisnika. Opisan je atributima: IDRecept i IDKorisnik. U *Many-To-Many* vezi je s entitetom Korisnik.

SpremljenRecept		
IDRecept	INT	jedinstveni identifikator recepta
IDKorisnik	INT	ID korisnika koji je spremio recept

Kategorija Entitet Kategorija služi za evidenciju kategorija. Opisan je atributima: IDKategorija, NazivKategorija i OpisKategorija.

Kategorija		
IDKategorija	INT	jedinstveni identifikator kateogrije
NazivKategorija	VARCHAR	naziv kategorije
OpisKategorija	VARCHAR	opis kategorije

ReceptKategorije Entitet ReceptKategorije služi za evidenciju kategorija i recepata pod tim kategorijama. Opisan je atributima: IDKategorija, NazivKategorija i OpisKategorija. U *Many-To-Many* vezi je s entitetom Recept.

ReceptKategorije		
IDRecept	INT	jedinstveni identifikator kateogrije
IDKategorija	INT	jedinstveni identifikator recepta

VrstaKuhinja Entitet VrstaKuhinja služi za evidenciju različitih vrsta kuhinja. Opisan je atributima: IDVrstaKuhinja, NazivVrstaKuhinja i OpisKategorija. U *One-To-One* vezi je s entitetom Recept.

VrstaKuhinja		
IDVrstaKuhinje	INT	jedinstveni identifikator vrste kuhinje
NazivVrstaKuhinja	VARCHAR	naziv vrste kuhinje

Sastojak Entitet Sastojak služi za evidenciju različitih sastojaka. Opisan je atributima: IDSastojak, NazivSastojak.

Sastojak		
IDSastojak	INT	jedinstveni identifikator sastojka
NazivSastojak	VARCHAR	naziv sastojka

ReceptSastojci Entitet ReceptSastojci služi za evidenciju različitih sastojaka unutar recepta. Opisan je atributima: IDRecept, IDSastojak i KolicinaSastojak. U *Many-To-Many* vezi je s entitetom Recept.

ReceptSastojci		
IDRecept	INT	jedinstveni identifikator recepta

Nastavljeno na idućoj stranici

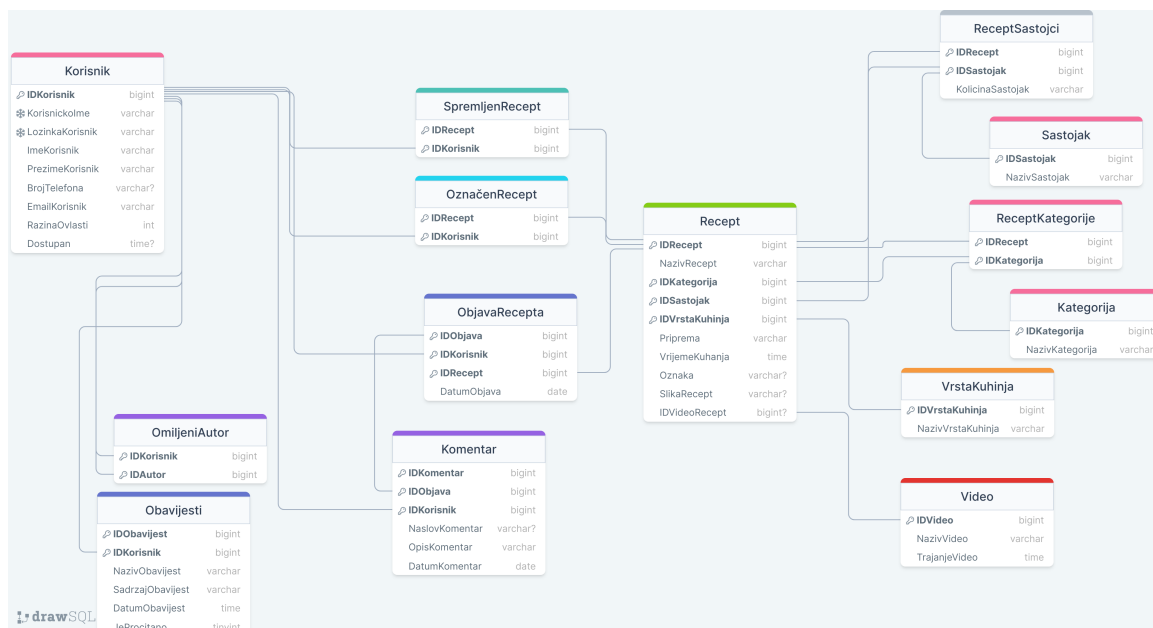
Nastavljeno od prethodne stranice

ReceptSastojci		
IDSastojak	INT	jedinstveni identifikator sastojka
KolicinaSastojak	VARCHAR	količina sastojka

Obavijesti Entitet Obavijesti služi za slanje obavijesti korisnicima koji prate određene autore receptata. Opisan je atributima: IDObavijest, IDAutor, NazivObavijest, SadržajObavijest, DatumObavijest i JeProcitano. U *Many-To-One* vezi je s entitetom Korisnik.

Obavijesti		
IDObavijest	INT	jed. identifikator obavijesti
IDKorisnik	INT	jedinstveni identifikator korisnika
NazivObavijest	VARCHAR	naziv obavijesti
SadržajObavijest	VARCHAR	sadržaj obavijesti
DatumObavijest	DATE	datum obavijesti
JeProcitano	TINYINT	oznaka pročitane obavijesti

4.1.2 Dijagram baze podataka



Slika 4.4: Dijagram relacijske baze podataka za web aplikaciju

4.2 Dijagram razreda

Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.

dio 1. revizije

*Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz **generičku funkcionalnost** sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.*

dio 2. revizije

Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije

4.3 Dijagram stanja

dio 2. revizije

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje **značajan dio funkcionalnosti** sustava. Na primjer, stanja korisničkog sučelja i tijekom korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

4.4 Dijagram aktivnosti

dio 2. revizije

Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.

4.5 Dijagram komponenti

dio 2. revizije

Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.

5. Implementacija i korisničko sučelje

5.1 Korištene tehnologije i alati

dio 2. revizije

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno **navesti internet poveznicu** gdje se mogu preuzeti ili više saznati o njima.*

5.2 Ispitivanje programskog rješenja

dio 2. revizije

U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.

5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi **minimalno 6 ispitnih slučajeva** u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium¹. Razraditi **minimalno 4 ispitna slučaja** u kojima će se ispitati redovni slučajevi, rubni uvjeti te poziv funkcionalnosti koja nije implementirana/izaziva pogrešku kako bi se vidjelo na koji način sustav reagira kada nešto nije u potpunosti ostvareno. Ispitni slučaj se treba sastojati od ulaza (npr. korisničko ime i lozinka), očekivanog izlaza ili rezultata, koraka ispitivanja i dobivenog izlaza ili rezultata.*

Izradu ispitnih slučajeva pomoću radnog okvira Selenium moguće je provesti pomoću jednog od sljedeća dva alata:

- *dodatak za preglednik **Selenium IDE** - snimanje korisnikovih akcija radi automatskog ponavljanja ispita*
- ***Selenium WebDriver** - podrška za pisanje ispita u jezicima Java, C#, PHP koristeći posebno programsko sučelje.*

Detalji o korištenju alata Selenium bit će prikazani na posebnom predavanju tijekom semestra.

¹<https://www.seleniumhq.org/>

5.3 Dijagram razmještaja

dio 2. revizije

Potrebno je umetnuti **specifikacijski** dijagram razmještaja i opisati ga. Moguće je umjesto specifikacijskog dijagrama razmještaja umetnuti dijagram razmještaja instanci, pod uvjetom da taj dijagram bolje opisuje neki važniji dio sustava.

5.4 Upute za puštanje u pogon

dio 2. revizije

*U ovom poglavlju potrebno je dati upute za puštanje u pogon (engl. deployment) ostvarene aplikacije. Na primjer, za web aplikacije, opisati postupak kojim se od izvornog kôda dolazi do potpuno postavljene baze podataka i poslužitelja koji odgovara na upite korisnika. Za mobilnu aplikaciju, postupak kojim se aplikacija izgradi, te postavi na neku od trgovina. Za stolnu (engl. desktop) aplikaciju, postupak kojim se aplikacija instalira na računalo. Ukoliko mobilne i stolne aplikacije komuniciraju s poslužiteljem i/ili bazom podataka, opisati i postupak njihovog postavljanja. Pri izradi uputa preporučuje se **naglasiti korake instalacije uporabom natuknica** te koristiti što je više moguće **slike ekrana** (engl. screenshots) kako bi upute bile jasne i jednostavne za slijediti.*

Dovršenu aplikaciju potrebno je pokrenuti na javno dostupnom poslužitelju. Studentima se preporuča korištenje neke od sljedećih besplatnih usluga: Amazon AWS, Microsoft Azure ili Heroku. Mobilne aplikacije trebaju biti objavljene na F-Droid, Google Play ili Amazon App trgovini.

6. Zaključak i budući rad

dio 2. revizije

U ovom poglavlju potrebno je napisati osvrt na vrijeme izrade projektnog zadatka, koji su tehnički izazovi prepoznati, jesu li riješeni ili kako bi mogli biti riješeni, koja su znanja stečena pri izradi projekta, koja bi znanja bila posebno potrebna za brže i kvalitetnije ostvarenje projekta i koje bi bile perspektive za nastavak rada u projektnoj grupi.

Potrebno je točno popisati funkcionalnosti koje nisu implementirane u ostvarenoj aplikaciji.

Popis literature

Kontinuirano osvježavanje

Popisati sve reference i literaturu koja je pomogla pri ostvarivanju 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.Langaniere, "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>

Indeks slika i dijagrama

2.1	Coolinarka	5
2.2	ReciPeci	5
3.1	Sekvencijski dijagram za UC1	12
3.2	Sekvencijski dijagram za UC3	13
3.3	Sekvencijski dijagram za UC4	14
3.4	Sekvencijski dijagram za UC7	15
4.1	Arhitektura sustava	17
4.2	Arhitektura sustava backend	18
4.3	Arhitektura sustava backend i frontend	18
4.4	Dijagram relacijske baze podataka za web aplikaciju	26

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

U ovom dijelu potrebno je redovito osvježavati dnevnik sastajanja prema predlošku.

1. Uvodni sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: A.Hohnjec, M.Pavić, A.Kantarević, L.Ivčević, B.Zrakić, F.Vrbić, A.Zglavnik
- Teme sastanka:
 - Upoznavanje sudionika projektnog tima

2. Sastanak o raspodjeli poslova

- Datum: 25. listopada 2023.
- Prisustvovali: A.Hohnjec, M.Pavić, A.Kantarević, L.Ivčević, B.Zrakić, F.Vrbić, A.Zglavnik
- Teme sastanka:
 - Dogovor o raspodjeli poslova pri izradi aplikacije
 - Dogovor o raspodjeli poslova pri izradi dokumentacije
 - Dogovor o načinu izvršavanja projekta

3. Merganje pojedino izrađene dokumentacije

- Datum: 30. listopada 2023.
- Prisustvovali: A.Hohnjec, M.Pavić, A.Kantarević, L.Ivčević, B.Zrakić, F.Vrbić, A.Zglavnik
- Teme sastanka:
 - Merganje pojedino izrađene dokumentacije

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Antonio Hohnjec	Marko Pavić	Armis Kantarević	Luka Ivčević	Benjamin Zrakić	Filip Vrbić	Antonio Zglavnik
Upravljanje projektom	10						
Opis projektnog zadatka	2						
Funkcionalni zahtjevi							
Opis pojedinih obrazaca				1	1	1	
Dijagram obrazaca					3	2	
Sekvencijski dijagrami				2			
Opis ostalih zahtjeva			1				
Arhitektura i dizajn sustava		1					
Baza podataka		4	3				
Dijagram razreda							2
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Antonio Hohnjec	Marko Pavić	Armis Kantarević	Luka Ivčević	Benjamin Zrakić	Filip Vrbić	Antonio Zglavnik
Dnevnik sastajanja	X						
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>		1			1		1
<i>izrada baze podataka</i>			1	1		1	
<i>spajanje s bazom podataka</i>	1						
<i>deployment</i>	1						

Dijagrami pregleda promjena

dio 2. revizije

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