

# Programsko inženjerstvo

Ak. god. 2023./2024.

## CookBooked

Dokumentacija, Rev. 1

Grupa: *Culinary Chaos Crew*

Voditelj: *Dominik Dejanović*

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

Nastavnik: *Nikolina Frid*

# Sadržaj

<b>1</b>	<b>Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2</b>	<b>Opis projektnog zadatka</b>	<b>5</b>
<b>3</b>	<b>Specifikacija programske potpore</b>	<b>8</b>
3.1	Funkcionalni zahtjevi . . . . .	8
3.1.1	Obrasci uporabe . . . . .	10
3.1.2	Sekvencijski dijagrami . . . . .	17
3.2	Ostali zahtjevi . . . . .	18
<b>4</b>	<b>Arhitektura i dizajn sustava</b>	<b>19</b>
4.1	Baza podataka . . . . .	19
4.1.1	Opis tablica . . . . .	19
4.1.2	Dijagram baze podataka . . . . .	20
4.2	Dijagram razreda . . . . .	21
4.3	Dijagram stanja . . . . .	22
4.4	Dijagram aktivnosti . . . . .	23
4.5	Dijagram komponenti . . . . .	24
<b>5</b>	<b>Implementacija i korisničko sučelje</b>	<b>25</b>
5.1	Korištene tehnologije i alati . . . . .	25
5.2	Ispitivanje programskog rješenja . . . . .	26
5.2.1	Ispitivanje komponenti . . . . .	26
5.2.2	Ispitivanje sustava . . . . .	26
5.3	Dijagram razmještaja . . . . .	27
5.4	Upute za puštanje u pogon . . . . .	28
<b>6</b>	<b>Zaključak i budući rad</b>	<b>29</b>
	<b>Popis literature</b>	<b>30</b>
	<b>Indeks slika i dijagrama</b>	<b>31</b>

**Dodatak: Prikaz aktivnosti grupe**

**32**

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	*	22.08.2013.
0.2	Dopisane upute za povijest dokumentacije. Dodane reference.	*	24.08.2013.
0.5	Dodan <i>Use Case</i> dijagram i jedan sekvencijski dijagram, funkcionalni i nefunkcionalni zahtjevi i dodatak A	*	25.08.2013.
0.6	Arhitektura i dizajn sustava, algoritmi i strukture podataka	*	26.08.2013.
0.8	Povijest rada i trenutni status implementacije, Zaključci i plan daljnjeg rada	*	28.08.2013.
0.9	Opisi obrazaca uporabe	*	07.09.2013.
0.10	Preveden uvod	*	08.09.2013.
0.11	Sekvencijski dijagrami	*	09.09.2013.
0.12.1	Započeo dijagrame razreda	*	10.09.2013.
0.12.2	Nastavak dijagrama razreda	*	11.09.2013.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus	*	11.09.2013.
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene: Timer - Brojilo vremena	*	15.09.2013.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

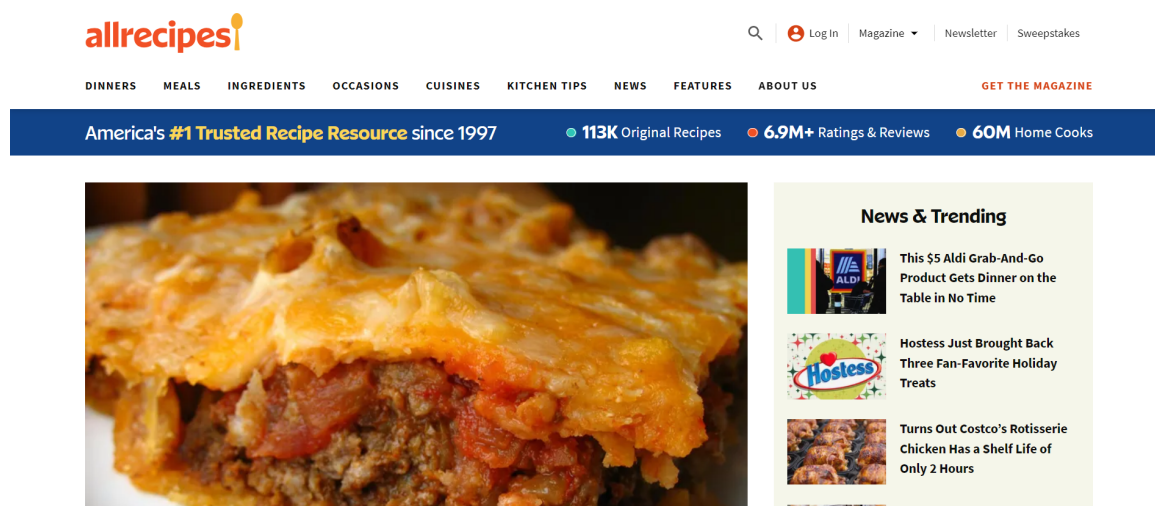
Rev.	Opis promjene/dodatka	Autori	Datum
1.3	Popravljeni dijagrami obrazaca uporabe	*	15.09.2013.
1.5	Generalna revizija strukture dokumenta	*	19.09.2013.
1.5.1	Manja revizija (dijagram razmještaja)	*	20.09.2013.
2.0	Konačni tekst predloška dokumentacije	*	28.09.2013.

*Moraju postojati glavne revizije dokumenata 1.0 i 2.0 na kraju prvog i drugog ciklusa. Između tih revizija mogu postojati manje revizije već prema tome kako se dokument bude nadopunjavao. Očekuje se da nakon svake značajnije promjene (dodatka, izmjene, uklanjanja dijelova teksta i popratnih grafičkih sadržaja) dokumenta se to zabilježi kao revizija. Npr., revizije unutar prvog ciklusa će imati oznake 0.1, 0.2, ..., 0.9, 0.10, 0.11.. sve do konačne revizije prvog ciklusa 1.0. U drugom ciklusu se nastavlja s revizijama 1.1, 1.2, itd.*

## 2. Opis projektnog zadatka

### Uvod

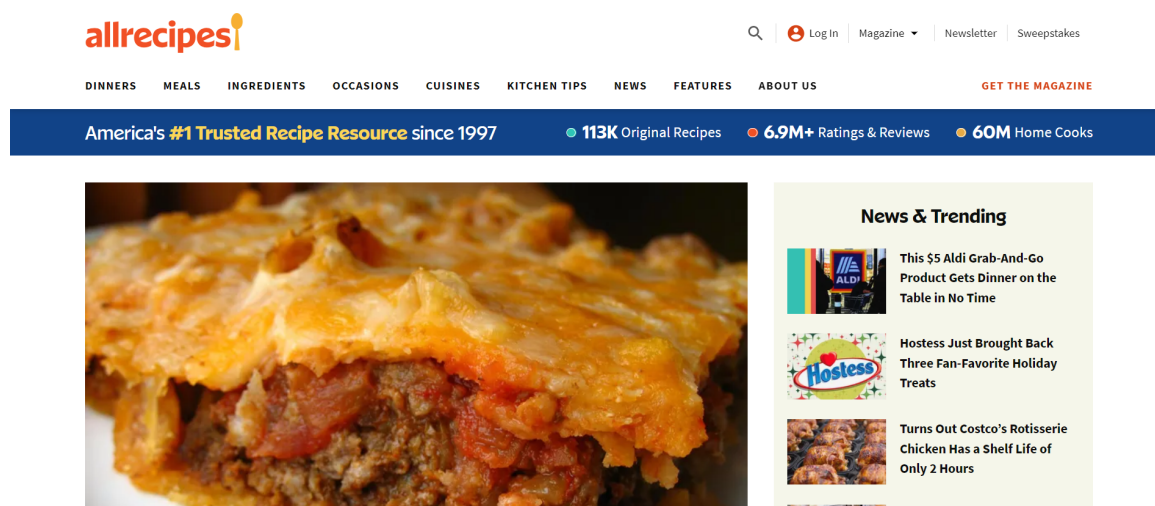
Cilj ovog projekta je razviti *CookBooked*, platformu za zajednicu ljubitelja kuhanja i pečenja kolača, koja omogućava korisnicima razmjenu i otkrivanje recepata iz cijeloga svijeta. Osim toga, platforma treba omogućiti bolje razumijevanje i komunikaciju između korisnika koji dijele istu strast prema kuhanju. Autori recepata će dobiti priliku izgraditi svoju reputaciju, povećati svoju publiku i razmijeniti iskustva s drugima. To će stvoriti motivaciju za kontinuirano stvaranje visokokvalitetnih recepata i sadržaja. S druge strane, korisnici će dobiti priliku unaprijediti vlastite vještine i pričati s autorima recepata te dobiti inspiraciju za vlastite eksperimente u kuhinji. Programska potpora za aplikaciju implementirat će se u obliku web aplikacije.



Slika 2.1: Dodati logo umjesto ove slike

## Postojeća slična rješenja

Iako postoje mnoge web stranice i aplikacije za dijeljenje recepata, *CookBooked* se izdvaja po svojoj fokusiranosti na interakciju između korisnika. Primjeri postojećih sličnih rješenja uključuju *AllRecipes*, *Tasty* i *Food Network*, no *CookBooked* se razlikuje po svojim značajkama za komunikaciju među korisnicima i praćenje autora.



Slika 2.2: Izgled naslovne stranice *AllRecipes*

## Skup korisnika

Platformu *CookBooked* može koristiti bilo koji zaljubljenik u hranu i njenu pripremu. Smatramo da će platforma biti najpogodnija ovim skupinama:

- kuhari i ljubitelji kuhanja
- osobe s posebnim prehrambenim potrebama (vegetarijanci, vegani, osobe koje ne jedu gluten...)
- ljudi koji traže inspiraciju za obroke
- autori recepata koji žele podijeliti svoje vještine

## Opseg projektnog zadatka

Za ostvarenje našeg ambicioznog projekta potrebno je slijedeće:

- razvoj platforme sa svim navedenim funkcionalnostima
- stvaranje korisničkog sučelja koje je jednostavno za korištenje
- implementaciju sustava za komunikaciju među korisnicima
- kreiranje profila za sve tipove korisnika
- upravljanje korisnicima i receptima od strane administratora

## Moguće nadogradnje projektnog zadatka

Iako je ovaj projekt unio neke nove ideje i funkcionalnosti u svijet kulinarstva, još uvijek postoje neke nadogradnje i funkcionalnosti koje bi mogle biti implementirane u budućnosti. Neki primjeri nadogradnji su:

- dodavanje dodatnih značajki za napredno pretraživanje recepata
- razvijanje mobilne aplikacije za pristup platformi putem mobilnih uređaja
- uvođenje funkcionalnosti za online prodaju kuhinjskih proizvoda
- integraciju s društvenim mrežama radi širenja korisničke baze
- unapređenje sigurnosti i privatnosti podataka korisnika



## 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. Korisnik
2. Administrator
3. Razvojni tim

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

1. Neregistrirani korisnik (inicijator) može:
  - (a) vidjeti listu recepata svih korisnika
  - (b) pretražiti recepte po naslovu i korisniku
    - i. filtrirati listu prema kategoriji, vrsti kuhinje i sastojcima
  - (c) pregledati detalje zasebnih recepata
  - (d) stvoriti novi korisnički račun
2. Registrirani korisnik (inicijator) može:
  - (a) ulogirati se pomoću vlastitog korisničkog računa
  - (b) objaviti nove recepte
    - i. odrediti kategoriju recepta

- ii. dodati posebne značajke i tagove na recept
- iii. priložiti slike ili videozapise receptu
- (c) pregledati listu vlastitih recepata
- (d) obrisati vlastiti recept
- (e) komunicirati s drugim registriranim korisnicima
  - i. razmjenjivati poruke i pozive
  - ii. postaviti termine otvorene za komunikaciju
- (f) označiti, komentirati i spremati recepte
- (g) pratiti omiljene autore i primati obavijesti o njihovim novim objavama
- (h) promijeniti detalje korisničkog računa:
  - i. korisničko ime
  - ii. lozinku
- (i) deaktivirati vlastiti korisnički račun

3. Administrator (inicijator) može:

- (a) pregledati korisničke podatke registriranih korisnika
- (b) pregledati objavljene recepte
- (c) promijeniti korisničke podatke registriranih korisnika
- (d) promijeniti podatke o receptu
- (e) izbrisati korisnički račun
- (f) izbrisati recept

4. Baza podataka (sudionik):

- (a) pohranjuje podatke o:
  - i. korisnicima
  - ii. receptima

5. API (sudionik):

- (a) omogućava dohvaćanje i strukturirano slanje podataka iz baze podataka

6. Web stranica(sudionik):

- (a) omogućava korisnicima interakciju sa:
  - i. drugim korisnicima
  - ii. listom recepata

### 3.1.1 Obrasci uporabe

#### *dio 1. revizije*

#### Opis obrazaca uporabe

*Funkcionalne zahtjeve razraditi u obliku obrazaca uporabe. Svaki obrazac je potrebno razraditi prema donjem predlošku. Ukoliko u nekom koraku može doći do odstupanja, potrebno je to odstupanje opisati i po mogućnosti ponuditi rješenje kojim bi se tijekom obrasca vratio na osnovni tijek.*

#### UC1 - stvaranje korisničkog računa

- **Glavni sudionik: korisnik**
- **Cilj: stvaranje novog korisničkog računa**
- **Sudionici: web stranica, API, baza podataka**
- **Preduvjet: -**
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb za prijavu
  2. korisnik upiše željeno korisničko ime i lozinku
  3. korisnik klikne na gumb "Registracija"
  4. šalje se zahtjev na API
  5. API zapisuje novog korisnika u bazu podataka
  6. ako je registracija uspješna web stranica se preusmjeri na homepage, ali sa korisnikom prijavljenim u sustav
- **Opis mogućih odstupanja:**
  - 1.a korisničko ime je zauzeto (već postoji)
    1. prikazuje se poruka "Korisničko ime zauzeto!"
  - 1.b lozinka je slaba (sadrži manje od 6 znakova i/ili sadrži samo slova)
    1. prikazuje se poruka "Lozinka mora imati najmanje 6 znakova te barem jedan broj ili specijalan znak!"
  - 1.c unesena adresa e-pošte nije važeća
    1. prikazuje se poruka "Adresa e-pošte ne postoji ili nije važeća!"

#### UC2 - prijava u korisnički račun

- **Glavni sudionik: korisnik**
- **Cilj: prijava korisnika u vlastiti račun**

- **Sudionici: web stranica, API, baza podataka**
- **Preduvjet: -**
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb za prijavu
  2. korisnik unosi korisničko ime i lozinku
  3. korisnik klikne na gumb "Prijava"
  4. šalje se zahtjev sa korisničkim podacima na API
  5. API provjerava točnost podataka u bazi podataka
  6. u slučaju da je korisnik unio točne podatke, web stranica se preusmjeri na homepage sa korisnikom koji je sada prijavljen
- **Opis mogućih odstupanja:**
  - 2.a korisničko ime ne postoji u bazi podataka
    1. prikazuje se poruka "Korisničko ime ne postoji!"
  - 2.b pogrešna lozinka
    1. prikazuje se poruka "Kriva lozinka!"

### UC3 - promjena korisničkih podataka

- **Glavni sudionik: korisnik**
- **Cilj: promjena korisničkog imena i/ili lozinke**
- **Sudionici: web stanica, API, baza podataka**
- **Preduvjet: korisnik je prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb "Postavke"
  - 2.a korisnik promijeni korisničko ime upisivanjem novog imena
  - 2.b korisnik promijeni lozinku upisivanjem nove lozinke
  3. korisnik klikne na gumb "Pohrani promjene"
  4. šalje se zahtjev za promjenom podataka na API
  5. u bazu podataka se upisuju promjene
  6. korisnik se automatski odjavi za stranice
- **Opis mogućih odstupanja:**
  - 3.a novo korisničko ime je zauzeto
    1. prikazuje se poruka "Korisničko ime zauzeto!"
  - 3.b lozinka je slaba (sadrži manje od 6 znakova i/ili sadrži samo slova)
    1. prikazuje se poruka "Lozinka mora imati najmanje 6 znakova te barem jedan broj ili specijalan znak!"

#### UC4 - brisanje korisničkog računa

- **Glavni sudionik:** korisnik
- **Cilj:** brisanje korisničkog računa i svih podataka vezanih uz račun
- **Sudionici:** web stanica, API, baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb "Postavke"
  2. korisnik klikne na gumb "Obriši račun"
  3. pojavljuje se obavijest "Ovim postupkom će se obrisati svi Vaši receptii vaš račun, jesti li sigurni da želite nastaviti?"
  4. klikom na gumb "Da" se šalje zahtjev za brisanje računa API-u
  5. API briše račun iz baze podataka
  6. korisnik se preusmjerava na homepage, te više nije prijavljen u sustav
- **Opis mogućih odstupanja:**
  - 4.a korisnik kliknuo na gumb "Ne"
    1. obavijest se zatvara te korisnik ostaje u postavkama

#### UC5 - pregled liste recepata

- **Glavni sudionik:** korisnik
- **Cilj:** prikazati recepte na web stranici
- **Sudionici:** web stranica, baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. korisnik odabere filtere i način sortiranja recepata
  2. korisnik u polje za pretraživanje upiše tekst
  3. web stranica šalje zahtjev za dohvaćanje recepata API-u
  4. API dohvaća recepte iz baze podataka i šalje ih web stranici
  5. na web stranici se prikazuje lista recepata
- **Opis mogućih odstupanja:**
  - 5.a nije upisan tekst po kojem se pretražuje
    1. prikazuju se svi recepti

#### UC6 - stvaranje novog recepta

- **Glavni sudionik:** korisnik
- **Cilj:** kreiranje novog recepta

- **Sudionici: web stranica, API, baza podataka**
- **Preduvjet: korisnik prijavljen u sustav**
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb "Novi recept"
  2. upišu se podaci o receptu kao postupak, slike, videi, kategorije itd.
  3. klikom na objavi recept se šalje zahtjev API-u
  4. API sprema recept u bazu podataka
  5. recept postaje vidljiv drugim korisnicima
- **Opis mogućih odstupanja:**
  - 6.a recept nema naslov
    1. prikazuje se poruka "Potrebno je upisati ime recepta"
  - 6.b nije upisana lista sastojaka
    1. prikazuje se poruka "Potrebno je upisati listu sastojaka"
  - 6.c ne postoji ni jedan korak pripreme
    1. prikazuje se poruka "Potrebno je upisati barem jedan korak pripreme"
  - 6.d nije uneseno vrijeme kuhanja
    1. prikazuje se poruka "Potrebno je upisati vrijeme kuhanja"

#### UC7 - promjena podataka o receptu

- **Glavni sudionik: korisnik**
- **Cilj: promjena bilo kojeg podatka o receptu kojeg je korisnik napravio**
- **Sudionici: web stranica, API, baza podataka**
- **Preduvjet: korisnik je prijavljen u sustav te je on autor recepta kojeg želi promijeniti**
- **Opis osnovnog tijeka:**
  1. korisnik klikne na svoj recept
  2. korisnik klikne na gumb za uređivanje recepta
  3. web stranica se preusmjerava na poveznicu za promjenu recepta
  - 4.a korisnik promijeni podatke o receptu i klikne gumb "Spremi promjene"
  - 4.b korisnik klikne na gumb "Izbriši recept"
  5. na API se šalje zahtjev sa novim podacima o receptu
  6. u bazu podataka se spremaju promjene
  7. web stranica se preusmjerava na poveznicu na kojoj se može pregledati recept ukoliko nije obrisao

- **Opis mogućih odstupanja:**

7.a problem u novim podacima (referencirati se na odstupanja u UC6)

### UC8 - prijava na recepte korisnika

- **Glavni sudionik:** korisnika
- **Cilj:** prijava korisnika na recepte autora kako bi primio obavijesti kada taj korisnik napravi novi recept
- **Sudionici:** web stranica, API, baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. prijavljeni korisnik otvara recept nekog autora
  2. klikom na gumb "???" se šalje zahtjev na API
  3. u bazu podataka se zapisuje da je korisnik prijavljen na recepte odabranog autora
  4. recepti autora se pokazuju u listi obavijesti
- **Opis mogućih odstupanja:**

8.a korisnik je već prijavljen na recepte autora

  1. klikom na gumb za prijavu na recepte autora se korisnik odjavljuje sa recepata tog autora

### UC9 - pregled obavijesti

- **Glavni sudionik:** korisnik
- **Cilj:** pregled recepata autora koje korisnik prati
- **Sudionici:** web stanica, API, baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav
- **Opis osnovnog tijeka:**
  1. korisnik klikne na gumb koji izgleda kao zvonce
  2. šalje se zahtjev na API
  3. iz baze podataka se čitaju obavijesti korisnika te se podaci vraćaju web stranici
  4. prikazuju se recepti svih autora koje korisnik prati
- **Opis mogućih odstupanja:**

### UC10 - spremanje/bookmarkanje recepta

- **Glavni sudionik:** korisnik

- **Cilj:** spremanje recepata kako bi se kasnije mogli pregledati omiljeni recepti
- **Sudionici:** web stanica, API, baza podataka
- **Preduvjet:** korisnik prijavljen u sustav
- **Opis osnovnog tijeka:**
  - 1.a korisnik klikne na gumb za spremanje recepta iz liste recepata
  - 1.b korisnik klikne na gumb za spremanje recepata prilikom prikaza pojedinog recepta
  2. šalje se zahtjev na API
  3. u bazu podataka se upisuje recept kao omiljen od korisnika
- **Opis mogućih odstupanja:**
  - 10.a recept je već spremljen kao omiljen
    1. recept se miče is liste omiljenih recepata

#### UC11 - pregled specifičnog recepta

- **Glavni sudionik:** korisnik
- **Cilj:** prikaz svih podataka vezanih uz recept
- **Sudionici:** web stranica, API, baza podataka
- **Preduvjet:** -
- **Opis osnovnog tijeka:**
  1. korisnik kliknuo na recept
  2. šalje se zahtjev na API za dohvaćanje podataka o receptu
  3. iz baze podataka se čitaju podaci o receptu
  4. na web stranici se prikazuju dohvaćeni podaci
- **Opis mogućih odstupanja:**

#### UC12 - komentiranje na recept

- **Glavni sudionik:** korisnik
- **Cilj:** ostavljanje komentara na nekom receptu
- **Sudionici:** web stranica, baza podataka
- **Preduvjet:** korisnik je prijavljen u sustav te je otvorio stranicu recepta
- **Opis osnovnog tijeka:**
  1. scrollanjem do kraja recepta se dolazi do dijela sa komentarima
  2. upisuje se tekst u polje za komentar
  3. klikom na gumb "Objavi komentar" se šalje zahtjev na API



4. u bazu podataka se upisuje komentar
  5. web stranica se osvježava
- **Opis mogućih odstupanja:**
    - 12.a tekst komentara je prazno
      1. gumb za objavljivanje komentara je zasivljen te je nemoguće kliknuti na njega

## UC

- **Glavni sudionik:**
- **Cilj:**
- **Sudionici:**
- **Preduvjet:**
- **Opis osnovnog tijeka:**
  
- **Opis mogućih odstupanja:**

## **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

#### *dio 1. revizije*

*Nacrtati sekvencijske dijagrame koji modeliraju najvažnije dijelove sustava (max. 4 dijagrama). Ukoliko postoji nedoumica oko odabira, razjasniti s asistentom. Uz svaki dijagram napisati detaljni opis dijagrama.*

## 3.2 Ostali zahtjevi

### *dio 1. revizije*

*Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju **kako se sustav treba ponašati** i koja **ograničenja** treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

## 4. Arhitektura i dizajn sustava

### dio 1. revizije

Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programске zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:

- izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu)
- organizaciju sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje)
- organizaciju aplikacije (npr. slojevi frontend i backend, MVC arhitektura)

### 4.1 Baza podataka

#### dio 1. revizije

Potrebno je opisati koju vrstu i implementaciju baze podataka ste odabrali, glavne komponente od kojih se sastoji i slično.

#### 4.1.1 Opis tablica

Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ

korisnik - ime tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

korisnik - ime tablice		
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

#### 4.1.2 Dijagram baze podataka

*U ovom potpoglavlju potrebno je umetnuti dijagram baze podataka. Primarni i strani ključevi moraju biti označeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".*

## 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 Selenium<sup>1</sup>. 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.*

---

<sup>1</sup><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	Dodati logo umjesto ove slike . . . . .	5
2.2	Izgled naslovne stranice <i>AllRecipes</i> . . . . .	6



# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

- Datum: u ovom formatu: 27. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

#### 2. sastanak

- Datum: u ovom formatu: 27. listopada 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
  - opis prve teme
  - opis druge teme

## Tablica aktivnosti

### Kontinuirano osvježavanje

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

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva							
Arhitektura i dizajn sustava							
Baza podataka							
Dijagram razreda							
Dijagram stanja							
Dijagram aktivnosti							
Dijagram komponenti							
Korištene tehnologije i alati							
Ispitivanje programskog rješenja							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<i>Dodatne stavke kako ste podijelili izradu aplikacije</i>							
<i>npr. izrada početne stranice</i>							
<i>izrada baze podataka</i>							
<i>spajanje s bazom podataka</i>							
<i>back end</i>							

## 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](https://gitlab.com) stranice, u izborniku Repository, pritiskom na stavku Contributors.*