

Programsko inženjerstvo

Ak. god. 2023./2024.

# Štete javnih površina

Dokumentacija, Rev. 1

Grupa: *CestaFix*

Voditelj: *Fran Fodor*

Datum predaje: **17. 11. 2023.**

Nastavnik: *izv. prof. dr. sc. Vlado Sruk*

# Sadržaj

<b>1 Dnevnik promjena dokumentacije</b>	<b>3</b>
<b>2 Opis projektnog zadatka</b>	<b>4</b>
2.1 Opis projektnog zadatka 'Štete javnih površina' . . . . .	4
<b>3 Specifikacija programske potpore</b>	<b>8</b>
3.1 Funkcionalni zahtjevi . . . . .	8
3.1.1 Obrasci uporabe . . . . .	9
3.1.2 Sekvencijski dijagrami . . . . .	10
3.2 Ostali zahtjevi . . . . .	11
<b>4 Arhitektura i dizajn sustava</b>	<b>12</b>
4.1 Baza podataka . . . . .	12
4.1.1 Opis tablica . . . . .	12
4.1.2 Dijagram baze podataka . . . . .	13
4.2 Dijagram razreda . . . . .	14
4.3 Dijagram stanja . . . . .	15
4.4 Dijagram aktivnosti . . . . .	16
4.5 Dijagram komponenti . . . . .	17
<b>5 Implementacija i korisničko sučelje</b>	<b>18</b>
5.1 Korištene tehnologije i alati . . . . .	18
5.2 Ispitivanje programskog rješenja . . . . .	19
5.2.1 Ispitivanje komponenti . . . . .	19
5.2.2 Ispitivanje sustava . . . . .	19
5.3 Dijagram razmještaja . . . . .	20
5.4 Upute za puštanje u pogon . . . . .	21
<b>6 Zaključak i budući rad</b>	<b>22</b>
<b>Popis literature</b>	<b>23</b>

**Indeks slika i dijagrama** 24

**Dodatak: Prikaz aktivnosti grupe** 25

# 1. Dnevnik promjena dokumentacije

## *Kontinuirano osvježavanje*

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Preuzet predložak.	Sara Podvorec	28.10.2023.
0.2	Dodane osnovne informacije. Napisan početak opisa projekta. Crvenom bojom označeni dijelovi teksta koje treba izmijeniti.	Jan Murić	28.10.2023.
0.3	Nadopuna opisa projekta.	Sara Podvorec	29.10.2023.
0.4	Nadopuna osnovnih informacija. Nadopuna dnevnika sastajanja.	Mateo Jakšić	30.10.2023.
0.5	Stiliziranje opisa projekta.	Sara Podvorec	30.10.2023.
0.6	Nadopuna dnevnika sastajanja.	Mateo Jakšić	31.10.2023.

*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

### 2.1 Opis projektnog zadatka 'Štete javnih površina'

Svakodnevno se suočavamo s brojnim problemima na javnim površinama i cestama u našim gradovima. Problemi kao što su vandalizam, oštećenje pločnika, udarne rupe na cestama, smeće i slično predstavljaju potencijalnu opasnost za građane te poprilično narušavaju kvalitetu njihovog svakodnevnog života.

Neadekvatna briga o oštećenjima i njihovoj sanaciji često ostavlja građane u vrlo nepovoljnoj situaciji kada iste žele prijaviti vlastima. Kako bi suzbili osjećaj nemoći građana i poboljšali kvalitetu života cjelokupne zajednice, potrebno je omogućiti adekvatnu prijavu oštećenja na javnim površinama.

Glavni cilj ovog projekta je razviti programsku podršku za stvaranje web aplikacije **"Unesi ime aplikacije koje moramo smisliti"** za dojavu oštećenja i drugih problema na cestama, parkovima, javnim ustanovama i ostalim javnim mjestima u svrhu olakšavanja dojave, kategorizacije te u konačnici rješenja prijavljenih problema. Ideja je omogućiti građanima da na što jednostavniji način prijavljuju oštećenja i probleme na javnim površinama i cestama svojih gradova te tako pomognu lokalnim vlastima da pravodobno reagiraju na nastale probleme.

S obzirom na velik broj javnih ustanova i njihovih odjeljaka koji se bave različitim problemima, teško je pratiti tko je nadležan za koju vrstu problema i nad kojim područjem. Sustav bi automatski odredio nadležno tijelo prema kategoriji prijave i drugim parametrima te prosljedio prijavu na obradu. Ključno je da proces prijave bude jednostavan i brz.

Prilikom pokretanja sustava prikazuje se početna stranica na kojoj se nalazi karta s označenim lokacijama već podnesenih prijava raznovrsnih oštećenja javnih površina, nekoliko informacija o samoj aplikaciji i navigacijski izbornik putem kojeg se pristupa registraciji/prijavi u sustav te podnošenju prijava kao i provjeri statusa već podnesenih prijava.

Svakom neregistriranom korisniku omogućeno je prijavljivanje u sustav s postojećim računom, prilikom čega je potrebno upisati:

- *korisničko ime*

- *lozinku*

ili kreiranje novog računa prilikom čega je potreban unos sljedećih podataka:

- *korisničko ime*
- *lozinka*
- *email adresa*

Također, postoji opcija anonimnog podnošenja prijave na temelju koje se status podnesene prijave prati putem jedinstvenog broja prijave bez iznošenja osobnih podataka.

Registrirani korisnik može pregledavati i mijenjati svoje osobne podatke i izbrisati svoj korisnički račun te postoji mogućnost naknadne dodjele prava službenika ili administratora stupanjem u kontakt naveden pri dnu stranice.

Kako bi olakšali proces podnošenja prijave, aplikacija omogućuje korisnicima da brzo i precizno dokumentiraju probleme. Svaka prijava uključuje naziv problema, kratki opis, geografske koordinate problema, jedinstveni broj prijave te opcionalno fotografije kako bi se problem što bolje opisao.

Korisnici mogu odabirom lokacije na karti, unosom najbliže adrese ili iz meta podataka slika priloženih uz prijavu precizno definirati lokaciju problema na koji su naišli. Sustav također olakšava identifikaciju vremenski bliskih prijava na istoj lokaciji kako bi se građani mogli povezati na već postojeće prijave sličnih problema, čime se smanjuje dupliciranje prijava i ubrzava proces njihovog rješavanja.

Podnesene prijave obrađuju određeni gradski uredi, a svaka promjena u statusu prijave vidljiva je prijavitelju i svim ostalim korisnicima koji su povezani s tom lokacijom.

Gradski uredi također imaju mogućnost objediniti nepovezane prijave na istoj lokaciji, što je vidljivo prijaviteljima. Kako bi gradskim uredima olakšali razvrstavanje i obrađivanje prijava, sustav dopušta prijave za različite kategorije problema. Sve podnesene prijave su javno dostupne i mogu se grupirati prema temi i lokaciji, što omogućuje građanima i vlastima da prate i analiziraju probleme u stvarnom vremenu.

Korisnik s pravom službenika ima mogućnost ažuriranja statusa prijave i pregled profila prijavitelja (ili jedinstvenog broja prijave u slučaju anonimnog prijavitelja) među kojima odabire relevantne prijave. Nakon što sustav proslijedi prijavu u nadležni gradski ured, službenik tog ureda može izabrati problem i krenuti raditi na njegovu rješavanju.

Administrator sustava ima najveće ovlasti među koje pripada i pristup bazi s popisom registriranih korisnika i njihovim podacima te mogućnost brisanja korisnika kao i dodjeljivanje administratorskih prava i prava službenika drugim korisnicima. Statistika o statusima prijava, poput vremena potrebnog za rješavanje problema, također se obrađuje u stvarnom vremenu kako bi poboljšala učinkovitost sustava. Posebna je pozornost posvećena zaštiti svih osobnih podataka te osiguranju korisničkog iskustva koje će biti jednostavno i intuitivno, kako bi se potaknulo građane na aktivno sudjelovanje u poboljšanju svojih zajednica.

Jedan od primjera sličnog sustava koji se aktivno upotrebljava u Hrvatskoj je "Gradsko Oko". Projekt je pokrenut u kolovozu 2017. godine u svrhu prijave komunalnih problema na području grada Bjelovara, a od tad se proširio na još 11 gradova i općina te na prijavu problema na moru.

Postojeća slična rješenja, kao što je "Gradsko Oko", pružaju mogućnost prijave komunalnih problema, ali se razlikuju u nekoliko ključnih aspekata. Naša aplikacija omogućava anonimno podnošenje prijava, prati statistiku rješavanja problema, te provjerava sličnost vremenski bliskih prijava kako bi se smanjilo dupliciranje prijava i ubrzao proces njihovog rješavanja.

Skup korisnika koji bi mogao biti zainteresiran za ovu aplikaciju je vrlo širok. To uključuje građane koji žele prijaviti probleme na javnim površinama, lokalne gradove i njihove uredima koji će obraditi prijave, službenike koji će nadzirati i rješavati prijave te administratore koji će upravljati sustavom i osigurati sigurnost podataka.

Ova web aplikacija ima potencijal za prilagodbu i nadogradnje u budućnosti. Moguće nadogradnje uključuju integraciju s mobilnim aplikacijama kako bi se olakšao proces prijave, dodatne značajke za praćenje rada gradskih ureda, uvođenje sustava za nagrađivanje korisnika koji aktivno sudjeluju u rješavanju problema te proširenje na druge gradove i regije.

### ***dio 1. revizije***

*Na osnovi projektnog zadatka detaljno opisati korisničke zahtjeve. Što jasnije opisati cilj projektnog zadatka, razraditi problematiku zadatka, dodati nove aspekte problema i potencijalnih rješenja. Očekuje se minimalno 3, a poželjno 4-5 stranica opisa. Teme koje treba dodatno razraditi u ovom poglavlju su:*

- potencijalna korist ovog projekta*
- postojeća slična rješenja (istražiti i ukratko opisati razlike u odnosu na zadani*

*zadatak). Dodajte slike koja predočavaju slična rješenja.*

- skup korisnika koji bi mogao biti zainteresiran za ostvareno rješenje.*
- mogućnost prilagodbe rješenja*
- opseg projektnog zadatka*
- moguće nadogradnje projektnog zadatka*

*Za pomoć pogledati reference navedene u poglavlju „Popis literature“, a po potrebi konzultirati sadržaj na internetu koji nudi dobre smjernice u tom pogledu.*



## 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.

#### **Dionici:**

1. Vlasnik
2. Građani
3. Gradski uredi
4. Službenici gradskih ureda
5. Administratori
6. Razvojni tim

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

1. Neprijavljeni korisnik (inicijator):
  - (a) Podnošenje anonimne prijave
  - (b) Praćenje prijave jedinstvenim kodom
  - (c) Uvid u postojeće prijave
  - (d) Prijava u korisnički račun
  - (e) Registracija korisničkog računa
2. Prijavljeni korisnik (inicijator):
  - (a) Podnošenje prijave pod vlastitim profilom
  - (b) Praćenje vlastitih prijava
  - (c) Uvid u postojeće prijave
  - (d) Izmjena podataka korisničkih računa
  - (e) Pregled profila drugih korisnika

### 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.*

#### UC<broj obrasca> -<ime obrasca>

- **Glavni sudionik:** <sudionik>
- **Cilj:** <cilj>
- **Sudionici:** <sudionici>
- **Preduvjet:** <preduvjet>
- **Opis osnovnog tijeka:**
  1. <opis korak jedan>
  2. <opis korak dva>
  3. <opis korak tri>
  4. <opis korak četiri>
  5. <opis korak pet>
- **Opis mogućih odstupanja:**
  - 2.a <opis mogućeg scenarija odstupanja u koraku 2>
    1. <opis rješenja mogućeg scenarija korak 1>
    2. <opis rješenja mogućeg scenarija korak 2>
  - 2.b <opis mogućeg scenarija odstupanja u koraku 2>
  - 3.a <opis mogućeg scenarija odstupanja u koraku 3>

#### **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

# Dodatak: Prikaz aktivnosti grupe

## Dnevnik sastajanja

### *Kontinuirano osvježavanje*

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

#### 1. sastanak

- Datum: 17. listopada 2023.
- Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
- Teme sastanka:
  - formiranje tima
  - dogovor načina komunikacije
  - dogovor oko korištenih tehnologija

#### 2. sastanak

- Datum: 19. listopada 2023.
- Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
- Teme sastanka:
  - dogovor podjele poslova u početnoj fazi
  - analiza projektnog zadatka

#### 3. sastanak

- Datum: 24. listopada 2023.
- Prisustvovali: Fran Fodor, Vedran Knežević, Ante Prkačin
- Teme sastanka:
  - sastanak s nastavnikom
  - upoznavanje s temom projekta

#### 4. sastanak

- Datum: 27. listopada 2023.

- Prisustvovali: Fran Fodor, Mateo Jakšić, Vedran Knežević, Leon Sattvik Kolenc, Jan Murić, Sara Podvorec, Ante Prkačin
- Teme sastanka:
  - definiranje funkcijskih zahtjeva
  - definiranje baze podataka

#### 5. sastanak

- Datum: 31. listopada 2023.
- Prisustvovali: Fran Fodor, Mateo Jakšić, Sara Podvorec, Ante Prkačin
- Teme sastanka:
  - sastanak s nastavnikom
  - pojašnjenje zadatka

## Tablica aktivnosti

### Kontinuirano osvježavanje

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

	Fran Fodor	Mateo Jakšić	Vedran Knežević	Leon Sattvik Kolenc	Jan Murić	Sara Podvorec	Ante Prkačin
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

	<b>Fran Fodor</b>	<b>Mateo Jakšić</b>	<b>Vedran Knežević</b>	<b>Leon Sattvik Kolenc</b>	<b>Jan Murić</b>	<b>Sara Podvorec</b>	<b>Ante Prkačin</b>
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>							

## 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.*