

Programsko inženjerstvo

Ak. god. 2023./2024.

Sustav za prijavu oštećenja javnih površina (name subject to change)

Dokumentacija, Rev. <1 ili 2>

Grupa: *VelicanstveniTimRaketa*

Koordinator: *Ivan Šimunić*

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

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

Sadržaj

1	Dnevnik promjena dokumentacije	3
2	Opis projektnog zadatka	5
2.1	Primjeri u \LaTeX u	9
3	Specifikacija programske potpore	13
3.1	Funkcionalni zahtjevi	13
3.1.1	Obrasci uporabe	15
3.1.2	Sekvencijski dijagrami	21
3.2	Ostali zahtjevi	22
4	Arhitektura i dizajn sustava	23
4.1	Baza podataka	23
4.1.1	Opis tablica	23
4.1.2	Dijagram baze podataka	24
4.2	Dijagram razreda	25
4.3	Dijagram stanja	26
4.4	Dijagram aktivnosti	27
4.5	Dijagram komponenti	28
5	Implementacija i korisničko sučelje	29
5.1	Korištene tehnologije i alati	29
5.2	Ispitivanje programskog rješenja	30
5.2.1	Ispitivanje komponenti	30
5.2.2	Ispitivanje sustava	30
5.3	Dijagram razmještaja	31
5.4	Upute za puštanje u pogon	32
6	Zaključak i budući rad	33
	Popis literature	34

Indeks slika i dijagrama	35
Dodatak: Prikaz aktivnosti grupe	36

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Ivan Šimunić	26.10.2023.
verzija	što je novo napravljeno/dodano.	autori	datum
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.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
1.1	Uređivanje teksta – funkcionalni i nefunkcionalni zahtjevi	* *	14.09.2013.
1.2	Manje izmjene:Timer - Brojilo vremena	*	15.09.2013.
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

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.*
- *moгуćnost prilagodbe rješenja*
- *opseg projektnog zadatka*
- *могуће 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.

U ovo moderno doba i vrijeme, kada su tehnološka sredstva uvijek nadomak ruke, nekolicina entuzijasta se dosjetila kako doskočiti u pomoć svom gradu. Naime, štete po cestama i javnim površinama se ne smanjuju, ali to je upravo cilj jednog ovakvog projekta.

Ono što ovaj projekt nastoji ponuditi jest upravo jedno pravo suvremeno i efikasno rješenje pristupačno svima a usmjereno samo prema jednoj stvari - prijavljivanje štete javnih površina u gradu. Cilj projekta je svim ljudima omogućiti jednostavni pristup aplikaciji preko koje će moći prijavljivati uočena oštećenja javnih površina i cesta u područjima gradova čime sveukupno kumulira poboljšanje života u društvu.

Ukratko, tijekom zamišljenih događaja je idući: Korisnik uoči oštećenu gradsku imovinu ili cestu te istu želi prijaviti gradskom uredu nadležnom za taj tip štete. Korisnik učitava našu aplikaciju te šalje prijavu sa određenim popunjenim podacima. Gradski ured dobiva obavijest o prijavi te kad istu riješi korisnik koji ju je prijavio dobiva obavijest da je ona razriješena.

Pri učitavanju početne stranice aplikacije korisniku je vidljiva karta, opcija za registraciju (ili prijavu), te opcija za podnošenje prijave. Registrirani kao i neregistrirani korisnici imaju mogućnost slanja prijave, ali razlika je u tome što neregistrirani korisnik dobiva jedinstveni broj pomoću kojeg prati status prijave, registrirani korisnik uvijek može samo ući u povijest svojih prijava i vidjeti status svake.

Neregistrirani korisnik se u svakom trenutku može registrirati, te pri registraciji trebe navesti iduće podatke:

- Ime
- Prezime
- Korisničko ime
- E-mail
- Lozinku

Registrirani korisnici se uvijek mogu prijaviti u sustav pomoću svog korisničkog imena i lozinke. Ako je korisnik prijavljen u sustav, ima mogućnost odjave ponuđenu u gornjem desnom kutu. Registrirani korisnici su podijeljeni na iduć uloge:

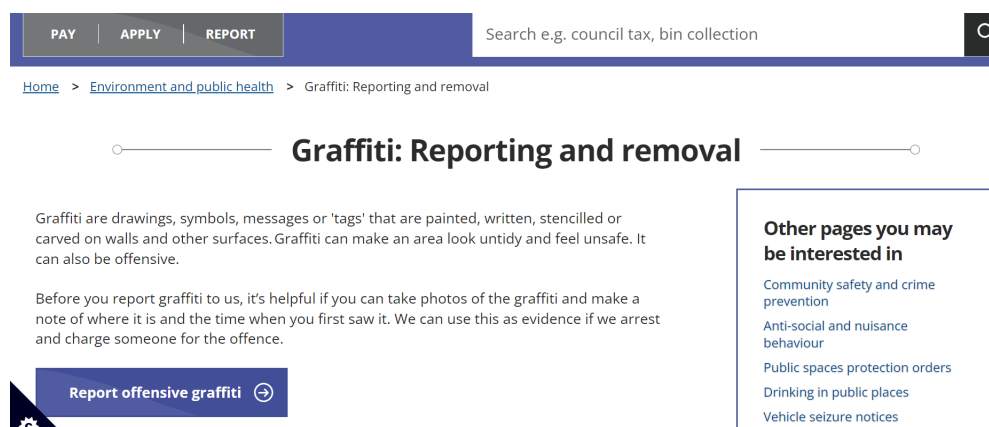
- Klijent
- Administrator
- Gradski ured

Klijent na karti ima označene sve aktivne prijave. Pri pregledu prijava, prijave može filtrirati po tematici (tipu) i lokaciji. Pri predaji prijave, korisnik mora unijeti iduće podatke:

- Naziv
- Kratki opis
- Geografske koordinate

Uz sve to, korisnik ocionalno može poslati i sliku viđenog pa sustav, u slučaju da nisu unesene koordinate, langituda i longituda se daju pročitati sa meta podataka fotografije. Web stranica koja nudi sličnu uslugu, ali je bazirana na vandalizmu i isključivo grafitima u Londonu je:

<https://www.newham.gov.uk/publichealth-safety/graffiti-reporting-removal>, a izgled je na slici 2.1.



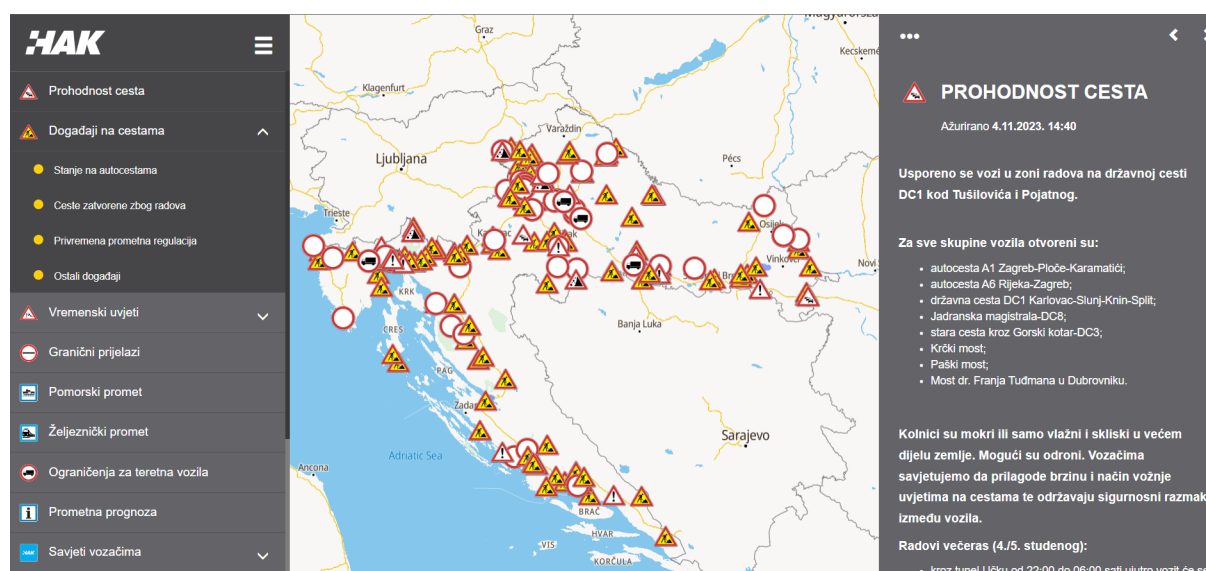
Slika 2.1: Izgled navedene stranice

Administrator je uloga koja ima najveće ovlasti. On je tu da kontrolira sve vezano za stranicu pa tako ima mogućnost uređivanja podataka pojedinih prijava kojih smatra nevaljalima kao i njihovo brisanje iz baze podataka. Pored toga, može brišati profile registriranih korisnika za koje procjeni da krše pravila ponašanja na aplikaciji.

Uz navedene korisnike sustava, još postoje i gradski uredi koji primaju prijave na temelju tematike problema za koju su zaduženi. Tako će puknuće na cestama

primati isključivo javni zavod za ceste, probleme sa zgradama će obrađivati ured za izgradnju i prostorno uređenje, probleme sa vodovom ured za vodovod... Dodatno, svaki gradski ured uvijek ima mogućnost promjene statusa prijave koji će prijaviteljima omogućiti da jasno vide ukoliko je prijava razriješena ili nije. Uz to, ured može povezati one prijave koje korisnici nisu povezali ukoliko shvati da se radi o istom problemu iste tematike.

Sustav će sve prijave obrađivati u stvarnom vremenu pa će korisnici u bilo kojem trenutku iamti uvid da li je npr. neko puknuće na cesti razriješeno te da li se tom cestom promet uopće može odvijati. Sličnu mogućnost nudi stranica HAK-a na idućoj adresi: <https://www.hak.hr/info/stanje-na-cestama/#prohodnost-cesta> na slici refhak.



Slika 2.2: HAK - stanje na cestama

Ovakva aplikacije sama po sebi već ima klijentelu i predispozicije za uspješno korištenje i popularizaciju. Uz štete javnih površina i cesta moglo bi se dodati prijava za zastoje primjerice tramvaja na određenoj lokaciji (naravno u gradovima gdje je tramvajski prijevoz omogućen) kako bi bilo vidljivo svim korisnicima aplikacije u stvarnom vremenu. Uz to moglo bi se područje aplikacije proširiti na prometne nesreće kako bi korisnici uz status da li je cesta zatvorena ili ne, mogli uz sliku procijeniti prohodnost iste. Neke vizualne stvari za korisnike koje se mogu još dodati bi bile gledanje tuđih profila, kao i dopisivanje porukama kako bi se doznale

konkretnije informacije o prijavi od osobe koja je tu prijavu napravila. Kako je Hrvatska turistička zemlja, aplikaciji bi se još mogli dodati ostali osnovni jezici za inozemne korisnike. Kako bi se što više promoviralo prijavljivanje šteta, u aplikaciju bi se dala ugraditi neka online nagrađivanja; primjerice dostupnost povećanja levela i stjecanja points-a.

2.1 Primjeri u L^AT_EXu

Ovo potpoglavlje izbrisati.

U nastavku se nalaze različiti primjeri kako koristiti osnovne funkcionalnosti L^AT_EXa koje su potrebne za izradu dokumentacije. Za dodatnu pomoć obratiti se asistentu na projektu ili potražiti upute na sljedećim web sjedištima:

- Upute za izradu diplomskog rada u L^AT_EXu - https://www.fer.unizg.hr/_download/repository/LaTeX-upute.pdf
- L^AT_EX projekt - <https://www.latex-project.org/help/>
- StackExchange za Tex - <https://tex.stackexchange.com/>

podcrtani tekst, **podebljani tekst**, *nagnuti tekst*

primjer primjer primjer primjer **primjer** primjer

- primjer
- primjer
- primjer
 1. primjer
 - 1.a primjer
 - b primjer
 2. primjer

primjer url-a: <https://www.fer.unizg.hr/predmet/proinz/projekt>

posebni znakovi: # \$ % & { } _ | < > ^ ~ \

naslov unutar tablice		
IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	

Tablica 2.1: Naslov s referencom izvan tablice

IDKorisnik	INT	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
korisnickoIme	VARCHAR	
email	VARCHAR	
ime	VARCHAR	
primjer	VARCHAR	



Slika 2.3: Primjer slike s potpisom



Slika 2.4: Primjer slike s potpisom 2

Referenciranje slike 2.4 u tekstu.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Regitrirani korisnik
 - (a) Klijent
 - (b) Administrator
2. Neregistrirani (anonimni) korisnik
3. Razvojni tim
4. Gradski ured

Aktori i njihovi funkcionalni zahtjevi:

1. Klijent (inicijator) može:
 - (a) Prijaviti se u sustav koristeći svoj email i lozinku
 - (b) Odjaviti se iz sustava
 - (c) Pregledati osobne podatke
 - (d) Mijenjati osobne podatke
 - (e) Izbrisati svoj profil
 - (f) Podnijeti prijavu preko web sučelja koja sadrži naziv, opis, geografske koordinate, te opcionalno fotografiju
 - (g) Odabrati koordinate preko karte ili unijeti najbližu adresu
 - (h) Povezati svoju prijavu na postojeću (ako takva postoji)
 - (i) Vidjeti promjene u statusu svoje prijave
 - (j) Odabrati aktivne prijave na karti
 - (k) Imati uvid u sve prijave u sustavu
 - (l) Pregledati povijet svojih prijava
 - (m) Filtrirati pregled prijava po lokaciji i temi
 - (n) Mijenjati podatke aktivne prijave u povijesti prijava

2. Administrator (inicijator) može:

- (a) Imati uvid u popis svih prijava
- (b) Imati uvid u popis svih registriranih profila
- (c) Uređivati podatke prijava
- (d) Brisati prijave
- (e) Imati uvid u popis svih gradskih ureda
- (f) Uklanjati (brisati) korisničke profile

3. Neregistrirani korisnik (incijator) može:

- (a) Napraviti novi korisnički račun klijenta
- (b) Podnijeti prijavu preko web sučelja koja sadrži naziv, opis, geografske koordinate, te opcionalno fotografiju
- (c) Odabrati koordinate preko karte ili unijeti najbližu adresu
- (d) Povezati svoju prijavu na postojeću (ako takva postoji)
- (e) Imati uvid u sve prijave na stranici, te iste grupirati po temi i lokaciji

4. Baza podataka (sudionik) može:

- (a) Imati popis svih postojećih korisničkih računa
- (b) Sadržavati sve jedinstvene oznake prijava neregistriranih oznaka
- (c) Pohraniti svaku prijavu

3.1.1 Obrasci uporabe

dio 1. revizije

Opis obrazaca uporabe

UC01 - Registracija korisnika u sustav

- **Glavni sudionik:** Neregistrirani korisnik
- **Cilj:** Stvoriti korisnički račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Null
- **Opis osnovnog tijeka:**
 1. Klijent bira opciju "registracija" na sučelju web aplikacije
 2. Klijent unosi tražene podatke
 3. Korisnik je upisan u bazu podataka
- **Opis mogućih odstupanja:**
 - 2.a Klijent unosi neispravni/postojeći username ili email
 1. Sustav obavještava korisnika o problemu i briše mu unesena polja
 2. Korisnik mijenja podatke u ispravne i registracija uspješno se završava

UC02 - Unos nove prijave u sustav

- **Glavni sudionik:** Korisnik
- **Cilj:** Podnijeti novu prijavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Null
- **Opis osnovnog tijeka:**
 1. Korisnik upisuje tražene podatke pri unosu prijave
 2. Sustav javlja ako postoji vremenski bliska prijava na toj lokaciji
 3. Korisnik može povezati svoju prijavu na postojeću (ako takva postoji)
 4. Prijava se predaje i zapisuje u sustav
- **Opis mogućih odstupanja:**
 - 1.a Korisnik nije naveo sve zahtijevane podatke
 1. Sustav obavještava korisnika o problemu i javlja mu da popuni tražena polja

UC03 - Pregled prijava

- **Glavni sudionik:** Korisnik
- **Cilj:** Pregled postojećih prijava
- **Sudionici:** Baza podataka
- **Preduvjet:** Null
- **Opis osnovnog tijeka:**
 1. Korisnik otvara pregled svih postojećih prijava
 2. Korisniku se nudi opcija filtriranja po temi i lokaciji
 3. Na sučelju se prikazuju filtrirane prijave

UC04 - Prijava u sustav

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Prijaviti se svojim profilom u sustav
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik upisuje korisničko ime i lozinku
 2. Sustav javlja potvrdu ispravnosti unesinih podataka
 3. Korisniku se učitava njemu prilagođeno sučelje
- **Opis mogućih odstupanja:**
 - 1.a Korisnik krivo unio korisničko ime/lozinku
 1. Sustav obavještava korisnika o problemu i javlja mu da ispravi tražena polja

UC05 - Uređivanje podataka prijave

- **Glavni sudionik:** Administrator
- **Cilj:** Korigirati podatke vezane za odabranu prijavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Prikazuju se sve prijave
 2. Administrator može filtrirati prijave
 3. Administrator izmjenjuje podatke prijave
 4. Izmjenjena prijava se sprema u bazu podataka

UC06 - Brisanje prijave

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati određenu prijavu
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Administratoru se prikazuje pregled prijava
 2. Administrator bira prijavu koju želi izbrisati
 3. Prijava se uklanja iz baze podataka i više nije vidljiva u aplikaciji

UC07 - Pregled korisnika

- **Glavni sudionik:** Administrator
- **Cilj:** Uvid u profile registriranih korisnika
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za pregled svih profila
 2. Otvara mu se lista svih registriranih profila

UC08 - Pregled osobnih podataka

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati osobne podatke svog profila
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik ulazi u opis svog profila
 2. Sustav mu prikaže username, e-mail i lozinku

UC09 - Odjava

- **Glavni sudionik:** Registrirani korisnik
- **Cilj:** Odjaviti se iz sustava
- **Sudionici:** Baza podataka
- **Preduvjet:** Aktivna prijava
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju odjava
 2. Sustav ga vraća na početnu stranicu web aplikacije

UC10 - Brisanje kroisničkog računa

- **Glavni sudionik:** Administrator
- **Cilj:** Obrisati određeni račun
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator bira profil koji želi ukloniti
 2. Sustav ga za provjeru pita da potvrdi odluku
 3. Administrator potvrđuje i profil se uklanja iz baze podataka

UC11 - Uvid povijest prijava

- **Glavni sudionik:** Klijent
- **Cilj:** Pregledati svu povijest privedenih prijava
- **Sudionici:** Baza podataka
- **Preduvjet:** Registracija
- **Opis osnovnog tijeka:**
 1. Korisnik bira opciju za prikazivanje povijesti prijava
 2. Sustav mu na sučelju prikazuje sve njegove prijave
 3. Korisnik dodatno može filtrirati iste po temi i lokaciji

UC12 - Pregled gradskih ureda

- **Glavni sudionik:** Administrator
- **Cilj:** Pregledati popis svih postojećih gradskih ureda
- **Sudionici:** Baza podataka
- **Preduvjet:** Dodijeljena prava administratora
- **Opis osnovnog tijeka:**
 1. Administrator odabire opciju za pregled svih gradskih ureda zapisanih u sustavu
 2. Sustav mu na sučelje prikazuje gradske urede zapisane u bazi podataka

UC13 - Povezivanje na postojeću prijavu

- **Glavni sudionik:** Korisnik
- **Cilj:** Vezati se ne vremenski blisku prijavu
- **Sudionici:** Baza podataka
- **Opis osnovnog tijeka:**
 1. Korisnik predaje prijavu
 2. Sustav mu javlja za vremenski blisku prijavu na toj lokaciji
 3. Korisnik bira hoće li se vezati na postojeću prijavu ili kreirati vlastitu

UC14 - Odabir aktivnih prijava sa karte

- **Glavni sudionik:** Klijent
- **Cilj:** Pogladati na karti neriješene prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje aktivnih prijava
- **Opis osnovnog tijeka:**
 1. Klijent na karti odabire opciju za uvid u aktivne prijave
 2. Sustav mu sve lokacije aktivnih prijava prikazuje na karti
- **Opis mogućih odstupanja:**
 - 1.a U sustavu nema aktivnih prijava
 1. Sustav korisnika izbacuje iz pregleda karte i javlja mu da nema aktivnih prijava

UC15 - Mijenjanje sadržaja aktivne prijave

- **Glavni sudionik:** Klijent
- **Cilj:** Promijeniti sadržaj vlastite aktivne prijave
- **Sudionici:** Baza podataka
- **Preduvjet:** Postojanje aktivne prijave
- **Opis osnovnog tijeka:**
 1. Klijent odabire svoju aktivnu prijavu koju želi izmijeniti
 2. Klijent mijenja atribut u prijavi
 3. Prijava s ažuriranim podakom se zapisuje u bazu podataka
- **Opis mogućih odstupanja:**
 - 1.a U trenutku odabira prijave, prijava više nije aktivna
 1. Sustav korisniku javlja da prijava više nije aktivna
 2. Korisnik je preusmjeren na početnu stranicu

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.

TODO: ako treba prenijeti dokumentaciju u wiki pa nacrtati specificirane dijagrame obrazaca uporabe i sekvencijske

Obrasci uporabe UC02, UC03, UC13, UC14 - unos nove prijave, pregled postojećih prijava, povezivanje na postojeću prijavu, uvid u aktivne prijave sa karte Korisnik (prijavljeni ili anonimni) može podnijeti novu prijavu sustavu. Korisnik odabire opciju unosa nove prijave nakon čega poslužitelj vraća formu za upis podataka prijave. Korisnik šalje prijavu te sustav provjerava unos korisnika i provjerava u postoje li vremenski i prostorno bliske aktivne prijave te ako postoje, vraća formu korisniku gdje se može spojiti na neku od bliskih prijava. Korisnik šalje poslužitelju hoće li se i na koju povezati te poslužitelj sprema novu prijavu u bazu podataka. Korisnik također može poslati zahtjev za pregledom svih postojećih prijava uz opcionalne filtere po temi i lokaciji na što poslužitelj reagira dohvaćanjem tih podataka iz baze te slanjem istih korisniku. Korisnik također može kliknuti na neku od aktivnih prijava na karti na što se poslužitelju šalje zahtjev na koji on reagira dohvaćanjem detalja te prijave iz baze podataka te odgovaranjem korisniku istima.

Obrazac uporabe UC01 - registracija korisnika u sustav

Neprijavljeni korisnik može poslati zahtjev za registracijom na što poslužitelj odgovara formom za registraciju. Kad korisnik pošalje podatke, poslužitelj ih validira te sprema u bazu podataka.

Obrazac uporabe UC05, UC06, UC07, UC10 - uređivanje podataka prijave, brisanje prijave, pregled korisnika, brisanje korisničkog računa, pregled gradskih ureda

Administrator može poslati zahtjev poslužitelju za uređivanje podataka iz pojedine prijave na što poslužitelj zahtjeva podatke te prijave iz baze te ih u formi za uređivanje vraća administratoru. Nakon uređivanja administrator šalje nove podatke poslužitelju na što ih on sprema u bazu. Administrator također može poslati

poslužitelju zahtjev za brisanjem prijave na što poslužitelj briše prijavu iz baze. Osim toga, administrator ima uvid i u sve registrirane korisnike kao liste koju na zahtjev poslužitelj zahtijeva iz baze te vraća administratoru. Osim korisnika, na isti način administrator može pregledati i gradske urede.

Obrazac uporabe UC04, UC08, UC11, UC15 - prijava korisnika u sustav, pregled osobnih podataka, pregled povijesti vlastitih prijava, mijenjanje sadržaja aktivne prijave

Registrirani korisnik može poslati zahtjev za prijavom u sustav na što poslužitelj odgovara formom za prijavu. Kad korisnik prijavu pošalje, poslužitelj ju validira uz pomoć baze podataka te obavještava korisnika o uspješnosti prijave. Prijavljeni korisnik može poslati poslužitelju zahtjev za pregledom povijesti vlastitih prijava na što ih poslužitelj dohvaća iz baze podataka te ih vraća korisniku. Na isti način prijavljeni korisnik može pregledati osobne podatke. Ukoliko prijavljeni korisnik želi mijenati svoju aktivnu prijavu, može odabrati jednu iz liste aktivnih prijava na što se poslužitelju šalje zahtjev za uređivanjem iste. Poslužitelj tad dohvaća podatke te prijave iz baze podataka te korisniku šalje formu s podacima prijave. Korisnik tad uređuje podatke te ih šalje poslužitelju na što ih poslužitelj validira te sprema u bazu podataka.

3.2 Ostali zahtjevi

- Sustav treba biti implementiran u obliku web aplikacije
- Aplikacija treba biti uvijek dostupna
- Aplikacije treba pružati usluge u stvarnom vremenu
- Učitavanje aplikacije ne smije trajati duže od 2 sekunde
- Pristup bazi podataka ne smije trajati duže od 2 sekunde
- Sustav treba biti organiziran u obliku MVC
- Sustav na poslužiteljskoj strani je napisan u programskom jeziku Java te radnom okviru Spring Boot
- Sustav na klijentskoj strani je implementiran programskim jezikom JavaScript te radnom okviru React.js
- Podaci se spremaju u bazu podataka koristeći JPA
- Arhitektura aplikacije mora biti u obliku klijent-poslužitelj
- Pri pristupu aplikaciji se koristi protokol HTTPS
- Aplikacije treba biti prilagođena i desktop uređajima kao i mobilnim uređajima

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 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	Izgled navedene stranice	7
2.2	HAK - stanje na cestama	8
2.3	Primjer slike s potpisom	11
2.4	Primjer slike s potpisom 2	12

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: 23. listopada 2023.
- Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Ivan Šimunić, Dominik Zoričić
- Teme sastanka:
 - raspodjela uloga po članovima
 - odabir izvedbenih tehnologija

2. sastanak

- Datum: 26. listopada 2023.
- Prisustvovali: Nikola Botić, Nino Ćurko, Ivan Elez, Davor Najev, Josip Šare, Dominik Zoričić
- Teme sastanka:
 - TODO

3. sastanak

- Datum: u ovom formatu: 7. studenoga 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.

	Ivan Šimunić	Nino Ćurko	Davor Najev	Nikola Botić	Dominik Zoričić	Josip Šare	Ivan Elez
Upravljanje projektom	1	2	2	2	2	2	2
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							
Upute za puštanje u pogon							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ivan Šimunić	Nino Ćurko	Davor Najev	Nikola Botić	Dominik Zoričić	Josip Šare	Ivan Elez
Dnevnik sastajanja	1						
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 stranice, u izborniku Repository, pritiskom na stavku Contributors.