

UNIVERZITET U SARAJEVU  
ELEKTROTEHNIČKI FAKULTET  
ODSJEK ZA RAČUNARSTVO I INFORMATIKU

PRAKTIKUM: POSLOVNI INFORMACIONI SISTEMI

# *Service Requirements Specification Dokument*

Studenti:

BOJADŽIĆ *Hanna*, 1421/17311

GRAĐANIN *Ehvan*, 1438/17335

HALILBEGOVIĆ *Kemal*, 1469/16682

KRUPALIJA *Ehlimana*, 1431/17461

KUBAT *Ismar*, 1422/17148

KULOVIĆ *Nejra*, 1519/17484

MUFTIĆ *Belma*, 1423/17260

PRIBIŠIĆ *Tarik*, 1536/2018

Predmetni nastavnik:

doc. dr. ANEL TANOVIĆ, dipl. ing. el.

Juni, 2019

# Sadržaj

<b>1 ITIL procesi</b>	<b>1</b>
1.1 <i>Change Management</i> . . . . .	1
1.2 <i>Event Management</i> . . . . .	2
<b>2 Opis i namjena sistema</b>	<b>4</b>
2.1 Osnovne informacije o kompaniji . . . . .	4
2.2 Poslovni procesi . . . . .	4
2.2.1 Upravljanje promjenama . . . . .	5
2.2.2 Upravljanje događajima . . . . .	6
2.3 Pregled slučajeva upotrebe . . . . .	7
2.4 Metodologija razvoja sistema . . . . .	8
2.5 Prototip sistema . . . . .	11
<b>3 Funkcionalni zahtjevi</b>	<b>14</b>
3.1 FZ 1. Prijava na sistem . . . . .	14
3.1.1 Uvod . . . . .	14
3.1.2 Preduslovi . . . . .	14
3.1.3 Ulazi . . . . .	14
3.1.4 Obrada . . . . .	14
3.1.5 Izlazi . . . . .	15
3.1.6 Prioritet realizacije . . . . .	15
3.2 FZ 2. Odjava sa sistema . . . . .	15
3.2.1 Uvod . . . . .	15
3.2.2 Preduslovi . . . . .	15
3.2.3 Ulazi . . . . .	15
3.2.4 Obrada . . . . .	15
3.2.5 Izlazi . . . . .	15
3.2.6 Prioritet realizacije . . . . .	16
3.3 FZ 3. Slanje zahtjeva za promjenom . . . . .	16
3.3.1 Uvod . . . . .	16
3.3.2 Preduslovi . . . . .	16
3.3.3 Ulazi . . . . .	16
3.3.4 Obrada . . . . .	16
3.3.5 Izlazi . . . . .	17
3.3.6 Prioritet realizacije . . . . .	17
3.4 FZ 4. Prijava grešaka . . . . .	17
3.4.1 Uvod . . . . .	17
3.4.2 Preduslovi . . . . .	17
3.4.3 Ulazi . . . . .	17
3.4.4 Obrada . . . . .	17
3.4.5 Izlazi . . . . .	18
3.4.6 Prioritet realizacije . . . . .	18
3.5 FZ 5. Pregled svih prijava . . . . .	18
3.5.1 Uvod . . . . .	18
3.5.2 Preduslovi . . . . .	18
3.5.3 Ulazi . . . . .	18
3.5.4 Obrada . . . . .	18

3.5.5	Izlazi . . . . .	19
3.5.6	Prioritet realizacije . . . . .	19
3.6	FZ 6. Arhiviranje prijava . . . . .	19
3.6.1	Uvod . . . . .	19
3.6.2	Preduslovi . . . . .	19
3.6.3	Ulazi . . . . .	19
3.6.4	Obrada . . . . .	19
3.6.5	Izlazi . . . . .	20
3.6.6	Prioritet realizacije . . . . .	20
3.7	FZ 7. Pregled svih pristiglih zahtjeva . . . . .	20
3.7.1	Uvod . . . . .	20
3.7.2	Preduslovi . . . . .	20
3.7.3	Ulazi . . . . .	20
3.7.4	Obrada . . . . .	20
3.7.5	Izlazi . . . . .	21
3.7.6	Prioritet realizacije . . . . .	21
3.8	FZ 8. Odlučivanje po zahtjevu . . . . .	21
3.8.1	Uvod . . . . .	21
3.8.2	Preduslovi . . . . .	21
3.8.3	Ulazi . . . . .	21
3.8.4	Obrada . . . . .	21
3.8.5	Izlazi . . . . .	22
3.8.6	Prioritet realizacije . . . . .	22
3.9	FZ 9. Pregled svih događaja u sistemu . . . . .	22
3.9.1	Uvod . . . . .	22
3.9.2	Preduslovi . . . . .	22
3.9.3	Ulazi . . . . .	22
3.9.4	Obrada . . . . .	22
3.9.5	Izlazi . . . . .	23
3.9.6	Prioritet realizacije . . . . .	23
3.10	FZ 10. Slanje zahtjeva na konačno odlučivanje . . . . .	23
3.10.1	Uvod . . . . .	23
3.10.2	Preduslovi . . . . .	23
3.10.3	Ulazi . . . . .	23
3.10.4	Obrada . . . . .	23
3.10.5	Izlazi . . . . .	24
3.10.6	Prioritet realizacije . . . . .	24
3.11	FZ 11. Raspoređivanje događaja . . . . .	24
3.11.1	Uvod . . . . .	24
3.11.2	Preduslovi . . . . .	24
3.11.3	Ulazi . . . . .	24
3.11.4	Obrada . . . . .	25
3.11.5	Izlazi . . . . .	25
3.11.6	Prioritet realizacije . . . . .	25
3.12	FZ 12. Pregled izvještaja . . . . .	25
3.12.1	Uvod . . . . .	25
3.12.2	Preduslovi . . . . .	26
3.12.3	Ulazi . . . . .	26
3.12.4	Obrada . . . . .	26

3.12.5 Izlazi . . . . .	26
3.12.6 Prioritet realizacije . . . . .	26
<b>4 Arhitekturalni <i>stack</i></b>	<b>27</b>
4.1 Dijagram klase . . . . .	27
4.2 <i>Entity Relationship</i> dijagram (ERD) . . . . .	27
<b>5 Tehnološki <i>stack</i></b>	<b>28</b>
5.1 Opis korištenih tehnologija . . . . .	28
<b>6 Nefunkcionalni zahtjevi</b>	<b>29</b>
6.1 Odziv sistema . . . . .	29
6.2 Propusnost . . . . .	29
6.3 Poruke o greškama . . . . .	29
6.4 Dostupnost . . . . .	29
6.5 Skalabilnost . . . . .	30
6.6 Dokumentovanost sistema . . . . .	30
6.7 Autorizacija korisnika . . . . .	30
<b>7 Korisnički <i>interface-i</i></b>	<b>32</b>
<b>8 Zakonska ograničenja dizajna</b>	<b>33</b>

# 1 ITIL procesi

## 1.1 Change Management

Upravljanje promjenama je ITIL (*Information Technology Infrastructure Library*) proces koji pripada skupu procesa u okviru tranzicije usluga. Njegova svrha je upravljanje promjenama koje se dešavaju nad nekim IT servisom, te olakšavanje prilagođavanja promjenama, kao i njihovom uvođenju i što boljem procesiranju zahtjeva za promjenama. U prvim verzijama ITIL-a promjene su bile generičke, no u kasnijim verzijama (od verzije 3 nadalje) prepoznata je potreba za uvođenjem tzv. modela promjena, koji razdvajaju promjene u različite kategorije prema značaju i hitnosti.

Pri upravljanju promjenama potrebno je definisati sljedeće pojmove:

1. **Zahtjev za promjenom (Request for Change - RFC)** - trebaju postojati jasno definisani slučajevi u kojima se prepozna potreba za vršenjem promjena. Promjene mogu inicirati osobe od interesa (*stakeholderi*, ukoliko je u pitanju IT usluga koju koristi neka kompanija i sl.), ili mogu biti vodene odzivom sistema, odnosno inicirane zbog grešaka, pogrešnog ili slabog korištenja nekih dijelova sistema. Zbog toga je važno omogućiti različitim osobama od interesa iniciranje promjena, kao i osigurati da se vrši praćenje korištenja sistema te analiza uspješnosti i frekvencije korištenja istog.
2. **Promjena** - moguće je vršiti različite akcije nad sistemom koje se smatraju promjenom: dodavanje novih komponenti, izmjenu ili potpuno uklanjanje starih, spajanje različitih funkcionalnosti i mnoge druge. Različite vrste promjena trebaju se tretirati na različite načine. Potrebno je jasno definisati šta je moguća promjena, koja je njen kategorija, koliki je njen prioritet te ko je zadužen za implementaciju iste.
3. **Tim za upravljanje promjenama** - treba postojati jasno definisana uloga za osobe koje će moći nadzirati promjene, odobravati vršenje istih ili odbacivati prijedloge. Od ovog osoblja očekuje se da je u stanju brzo i adekvatno reagovati na hitne promjene, kao i da izvrši ispravno rasudivanje da li će neka promjena pozitivno ili negativno utjecati na sam sistem (potrebno je izvršiti analizu koja se dokumentuje u vidu *Projected Service Outage - PSO* dokumenta, u kojem su izlistane sve predvidljive devijacije od plana promjena).

Upravljanje promjenama sastoji se od četiri osnovna podprocesa:

1. **Filtriranje promjena**, u okviru kojeg se pregledaju svi primljeni zahtjevi za vršenje promjena, analiziraju se i vrši se odluka koji zahtjevi će se prihvati, a koji zahtjevi će se odbiti, uz što se prilaže PSO dokument.
2. **Savjetovanje sa CAB (Change Advisory Board i komitetom za hitne promjene**, što se odnosi na argumentovanje prethodno utvrđenih zahtjeva koji su prepoznati kao neophodni, te predstavljanje tih argumenata savjetima i komitetima koji jedini imaju autoritet za odobravanje promjena. Tek nakon što se promjene odobre, počinje se sa njihovom implementacijom. U svakom drugom slučaju, promjene se odbacuju.
3. **Upravljanje promjenama**, što se odnosi na dodjeljivanje posla zaposlenicima ovisno o vrsti zahtjeva koji je potrebno ispuniti te praćenje rada na dodavanju novih, zamjenom ili uklanjanjem postojećih funkcionalnosti u skladu s postojećim zahtjevima. Upravljanje promjenama vrši se stalno, od početka do kraja procesa njihovog uvođenja u sistem.

4. **Izvještavanje o promjenama**, odnosno kreiranje izvještaja o svakoj izvršenoj promjeni, što uključuje argumente za i protiv vršenja promjena, PSO dokument, dozvolu od CAB i drugih komiteta, kao i izvještaj o samom procesu implementacije i uvođenja promjena te kako je uvođenje istih u sistem promijenilo poslovanje, tj. da li je uvođenje promjena imalo pozitivan efekat ili ne.

## 1.2 Event Management

Upravljanje događajima je ITIL (*Information Technology Infrastructure Library*) proces koji pripada skupu procesa u okviru upravljanja uslugama. Njegova svrha je da nadzire sve značajne događaje u sistemu, detektuje nove i procesira ih na adekvatan način. Ovaj proces u *ITIL*-u 2011 povezan je sa upravljanjem pristupom preko interfejsa, budući da su ta dva procesa u bliskoj korelaciji: u sistemu koji nadzire važne, često sigurnosno osjetljive događaje, onemogućavanje neautorizovanog pristupa jedan je od najvažnijih zahtjeva.

Pri upravljanju događajima potrebno je definisati sljedeće pojmove:

1. **Događaj** - u kompleksnim sistemima postoje različite vrste događaja i notifikacija, te je potrebno napraviti razgraničenje kojim vrstama događaja se bavi sistem upravljanjem događajima (događaji se najčešće dijele na informacije, upozorenja i greške).
2. **Prioritet** - različiti događaji imaju različit značaj, pri čemu je neke događaje potrebno hitno procesirati, dok drugi događaji mogu čekati određeno vrijeme. Iz tog razloga neophodno je svim vrstama događaja dodijeliti određene prioritete, koji se najčešće označavaju brojevima.
3. **Odziv** - sistem najčešće treba reagovati na događaje, posebno ukoliko je njihov prioritet veliki. Iz tog razloga potrebno je definisati vrstu i vrijeme odziva na različite događaje, te način njihovog procesiranja.

Upravljanje događajima sastoji se od četiri osnovna podprocesa:

1. **Korištenje mehanizama za nadzor događaja**, koji se odnosi na stalnu mogućnost sistema da detektuje događaje, kao i da nadzire postojeće događaje i vrši njihovo procesiranje. Budući da je broj događaja koji konstantno ulaze u sistem veoma veliki, vrlo je važno ispravno izvršiti diferenciranje na različite vrste događaja, kako se ne bi detektovao veliki broj nevažnih ili nekritičnih događaja. S druge strane, sistem mora biti u mogućnosti reagovati na događaje na adekvatan način, posebno u slučaju kad se više različitih događaja treba procesirati u isto vrijeme.
2. **Prvi nivo korelacije: Filtriranje događaja**, odnosno analiza događaja kako bi se odredilo da li je događaj relevantan za sistem ili ne. Ukoliko je događaj samo informacija, najčešće nije potrebno vršiti nikakve akcije te se takvi događaji mogu ignorisati. No, upozorenja i greške se moraju procesirati na adekvatan način, te je vrlo važno imati adekvatne mehanizme kako takvi događaji ne bi bili ignorisani (ali i kako informacije ne bi bile procesirane, ukoliko to nije neophodno).
3. **Drugi nivo korelacije: Izbor odziva**, odnosno dalja analiza događaja koji nisu filtrirani na prethodnom nivou. Budući da je za upozorenja i greške najčešće potrebno imati adekvatan odziv, na ovom nivou potrebno je odrediti da li odziv može biti automatski (u pitanju je neki od poznatih događaja za koje postoji predefinisani koraci odziva), ili je potrebno eskalirati situaciju na viši nivo.

4. **Pregled događaja i njegovo zatvaranje**, koji se odnosi na dodatnu provjeru aktivnih događaja koji se trenutno nadziru. Ukoliko je automatska akcija bila dovoljna da se događaj procesira na adekvatan način i stanje vrati u normalno, te ukoliko je sistem proslijedio informacije o događaju individuama sistema koji su ga procesirali na adekvatan način, takav događaj moguće je zatvoriti, kako bi se novi događaji mogli procesirati bez čekanja.

## 2 Opis i namjena sistema

### 2.1 Osnovne informacije o kompaniji

Kompanija **Idea d.o.o. Sarajevo** je mlada kompanija koja se bavi razvojem i održavanjem informacionih sistema za državne institucije, poput Zavoda za izgradnju Kantona Sarajevo, Zavoda za informatiku i statistiku Kantona Sarajevo, Direkcije za robne rezerve Kantona Sarajevo i mnogih drugih. Ovi informacioni sistemi omogućavaju pregled svih zaposlenika, njihovih personalnih dosjeva (uključujući lične podatke, razne vrste dokumenata, rješenja i ugovora) te iniciranje različitih akcija, poput slanja na seminare, edukacije i stručna usavršavanja, vršenje ocjenjivanja radnika, donošenje različitih vrsta rješenja te vršenje sistematizacije.

Softver koji je razvijen potrebno je svakodnevno održavati. Korisnici sistema su najčešće srednje životne dobi, bez tehničkog znanja na visokom nivou, te se svakodnevno susreću s različitim poteškoćama u korištenju sistema. Ponekad dolazi do pojave grešaka, na koje je potrebno što prije reagovati, kako se ne bi ugrozilo ispravno i kontinualno funkcionisanje sistema. Same institucije često šalju zahtjeve za promjenama sistema, zbog stalnog donošenja novih odredbi te izmjene starih, zbog čega je sistem potrebno stalno ažurirati kako bi odgovarao trenutnom stanju i olakšavao poslovanje ovih institucija.

Iz svih ovih razloga, Idea d.o.o. Sarajevo ima potrebu za razvojem aplikacije koja će im omogućiti upravljanje događajima i promjenama u sistemu. Ovakva aplikacija olakšati će detekciju grešaka i zahtjeva za promjenama, ubrzati reagovanje na hitne promjene te osigurati da sistem kontinualno funkcioniše bez problema, te da njegovi korisnici mogu imati pouzdano i ažurirano okruženje u kojem će im biti jednostavno raditi.

### 2.2 Poslovni procesi

Kompanija želi informacioni sistem koji omogućava sljedeće funkcionalnosti:

- Pregled svih zahtjeva za promjenama od strane tima za upravljanje promjenama;
- Pisanje izvještaja (PSO dokumenta) o predloženoj promjeni te slanje istog komitetu za promjene na pregled;
- Konačno odobravanje ili odbijanje promjena od strane komiteta za promjene;
- Prihvatanje i odbijanje promjena, te njihovo označavanje kompletiranim nakon implementacije;
- Slanje promjena timu za produkciju;
- Prijavljivanje grešaka i problema pri implementaciji od strane tima za produkciju;
- Pregled statističke analize korištenja sistema, uz informacije o slabo ili nikako korištenim dijelovima;
- Prijava problema pri korištenju i slanje zahtjeva za promjene od strane korisnika sistema koji firma održava.

Sve ove funkcionalnosti trebaju biti objedinjene u jednoj aplikaciji, te mora biti osigurana sigurnost i konzistentnost podataka. U nastavku će biti prikazane osnovne informacije vezane za ITIL procese pomoću kojih je moguće modelirati poslovne procese sistema za koji je potrebno napraviti aplikaciju.

### 2.2.1 Upravljanje promjenama

Sama svrha sistema je praćenje korištenja softvera koji se održava te prijem i procesiranje zahtjeva za promjene. U skladu s tim, definišu se dvije kategorije promjena:

- *Obične promjene*, odnosno promjene koje inicijalizira tim za upravljanje promjenama na osnovu zahtjeva koje šalju autorizirani korisnici sistema koje firma Idea razvija i održava, ili koje tim za upravljanje promjenama ocjeni neophodnim na osnovu statističke analize korištenja sistema (slabo ili nikako korišteni dijelovi sistema);
- *Hitne promjene*, koje se detektuju na osnovu prijave problema pri korištenju od strane korisnika sistema koje firma održava. Prijava problema korisnicima je omogućena unutar same aplikacije.

Osim kategorija, promjene imaju i svoje vrste, kako slijedi:

- Dodavanje novih komponenti;
- Izmjena postojećih komponenti;
- Brisanje postojećih komponenti.

Svaka promjena mora imati definisanu kategoriju i vrstu. Tim za upravljanje promjenama vrši njihov pregled i filtriranje - zahtjev se analizira, te se ili odbija uz obrazloženje, ili se šalje komitetu za promjene na dodatnu analizu i odobravanje. Komitet za promjene ima mogućnost pregleda svih promjena koje su im poslane od strane tima za upravljanje promjenama, te odbijanje ili odobravanje promjena. Te informacije prosljeđuju se timu za upravljanje promjenama, te ukoliko se odobri vršenje promjena, iste se dodjeljuju jednom od dostupnih timova za razvoj softvera.

Na osnovu ovih informacija, moguće je dodijeliti određene funkcionalnosti sistema različitim vrstama korisnika, na način kako slijedi:

- Slanje novog zahtjeva za promjenama - **korisnici sistema**;
- Pregled svih zahtjeva za promjenama - **tim za upravljanje promjenama**;
- Pisanje izvještaja (PSO dokumenta) o predloženoj promjeni te slanje istog komitetu za promjene na pregled - **tim za upravljanje promjenama**;
- Prihvatanje i odbijanje promjena, te njihovo označavanje kompletiranim nakon implementacije - **tim za upravljanje promjenama**;
- Konačno odobravanje ili odbijanje promjena - **komitet za promjene**;
- Slanje promjena timu za produkciju - **tim za upravljanje promjenama**;
- Pregled statističke analize korištenja sistema, uz informacije o slabo ili nikako korištenim dijelovima - **tim za upravljanje promjenama**.

### 2.2.2 Upravljanje događajima

Događaje u sistemu predstavljaju prijave problema od strane korisnika sistema koji se održavaju, ili od strane razvojnog tima koji vrši implementaciju izmjena u sistemu. Pregled i nadziranje događaja vrši tim za upravljanje promjenama, dok korisnici sistema i razvojni tim mogu vršiti iniciranje novih događaja.

U skladu s ovim, potrebno je izvršiti definisanje svih različitih događaja u sistemu, njihovih tipova i prioriteta, što je prikazano u Tabeli 1.

Vrsta događaja	Inicijator	Prioritet
Greška u aplikaciji	Korisnik	3
Pogrešno korištenje aplikacije	Korisnik	1
Greška nakon uvođenja promjena	Razvojni tim	2
Nemogućnost implementacije promjene	Razvojni tim	3

Tabela 1: Svi događaji sistema

Svi događaji prikazuju se timu za upravljanje promjenama, koji vrše odziv na događaje, na jedan od tri načina:

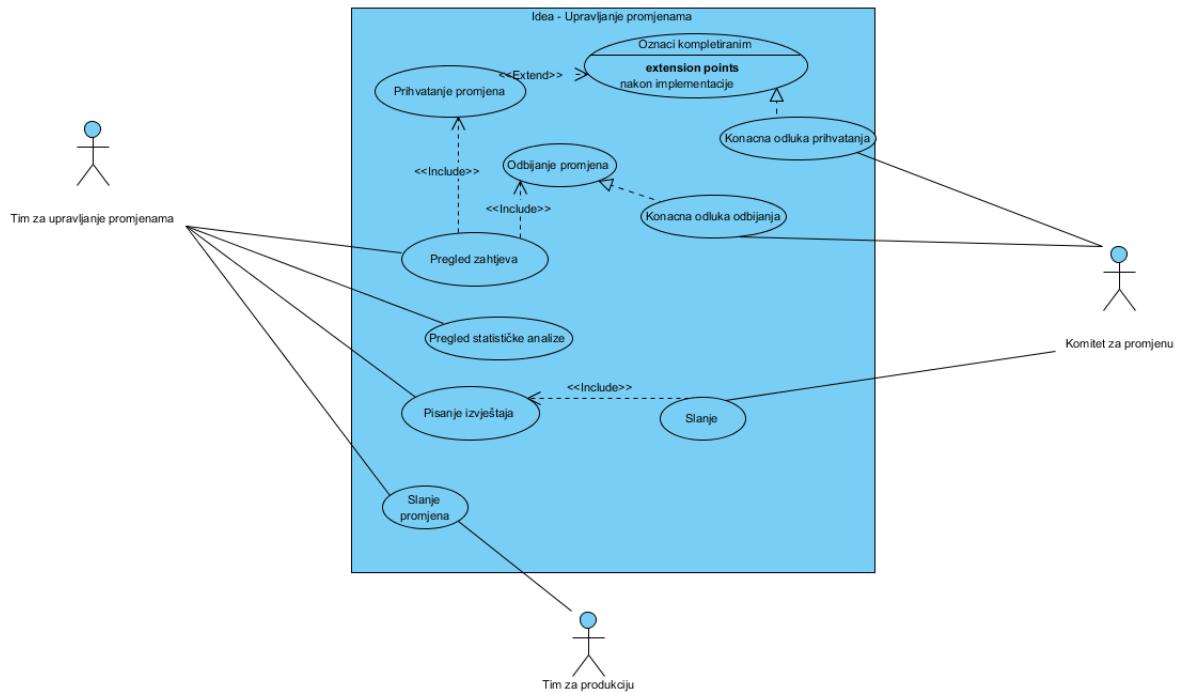
- *Otkazivanje promjena*, koje se vrši ukoliko razvojni tim prijavi da nije u mogućnosti da implementira promjene iz nekog razloga (poput samog načina na koji sistem funkcioniše, nedovoljnog broja informacija o samoj promjeni koju je potrebno izvršiti, nemogućnosti povezivanja sa postojećim sistemom i sl.).
- *Slanje razvojnom timu na popravljanje*, koje se vrši ukoliko korisnici prijave grešku u sistemu prije ili nakon uvođenja promjena ili ukoliko dođe do greške pri testiranju sistema nakon uvođenja promjena.
- *Organizacija edukacije korisnika ili razgovor s korisnikom*, koji se vrše ukoliko korisnik prijavi da ne zna kako ispravno koristiti sistem, što nije dio samog sistema koji se razvija.

Na osnovu ovih informacija, moguće je dodijeliti određene funkcionalnosti sistema različitim vrstama korisnika, na način kako slijedi:

- Prijavljivanje grešaka i problema pri implementaciji promjena - **razvojni tim**;
- Prijava problema pri korištenju sistema koji firma održava - **korisnici sistema koji se održava**.

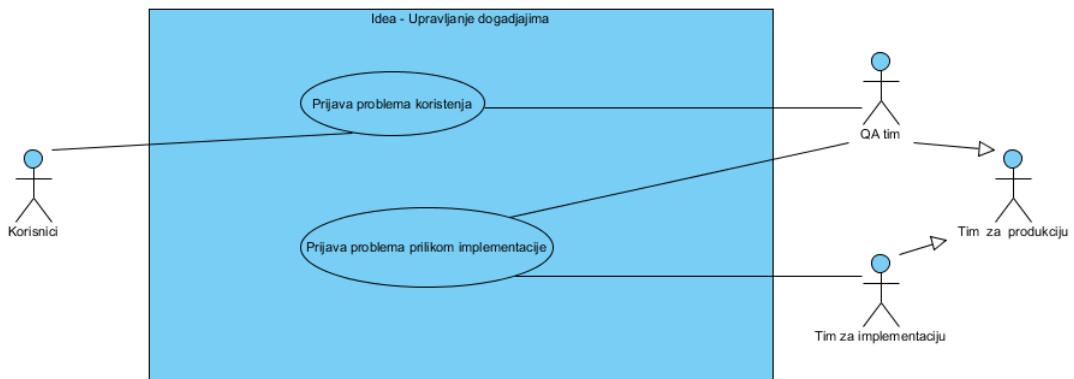
## 2.3 Pregled slučajeva upotrebe

Slučajevi upotrebe sistema koji odgovaraju funkcionalnostima vezanim za upravljanje promjenama prikazani su na Slici 1. Akteri sistema su tim za upravljanje promjenama, komitet za promjene i tim za produkciju, koji učestvuju u realizaciji slučajeva upotrebe.



Slika 1: Dijagram slučajeva upotrebe za upravljanje promjenama u sistemu

Slučajevi upotrebe sistema koji odgovaraju funkcionalnostima vezanim za upravljanje događajima prikazani su na Slici 2. Akteri sistema su korisnici sistema i tim za produkciju, koji učestvuju u realizaciji slučajeva upotrebe.



Slika 2: Dijagram slučajeva upotrebe za upravljanje događajima u sistemu

## 2.4 Metodologija razvoja sistema

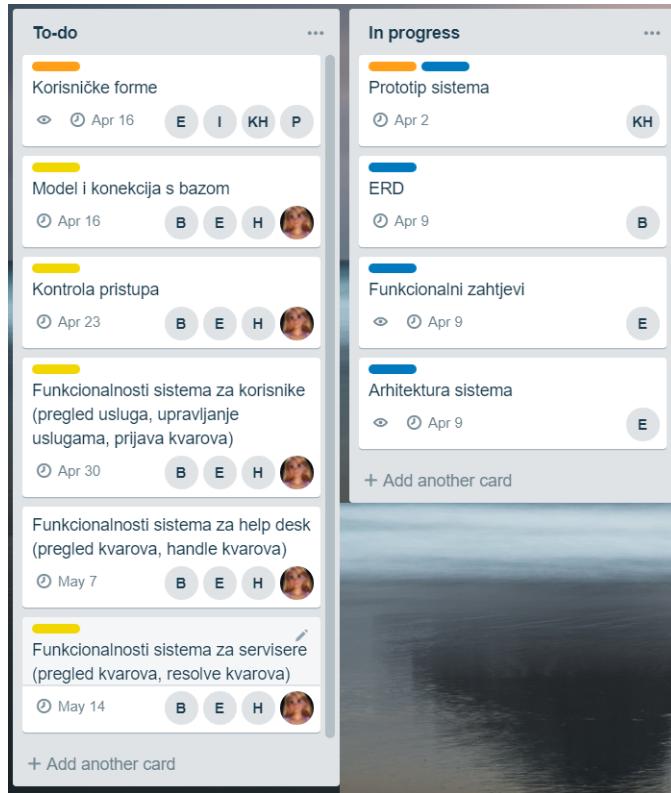
Za razvoj sistema biti će korištena SCRUM metodologija, koja predstavlja jednu od najpoznatijih agilnih metodologija za razvoj softvera. Razvoj sistema obavljati će se u sprintovima, sa trajanjem svakog sprinta od po 7 dana, pri čemu će se funkcionalnosti sistema razdvojiti prema korisničkim ulogama. Dodatno, *task-ovi* će biti razdvojeni na **frontend** i **backend** taskove, a novi zadaci biti će dodavani po potrebi (ovisno o mogućnostima raspodjele većih zadataka na manje podzadatke).

Inicijalni *Trello Board* prikazan je na Slici 3 i 4. Na slikama je vidljivo da su zadaci podijeljeni u nekoliko različitih grupa:

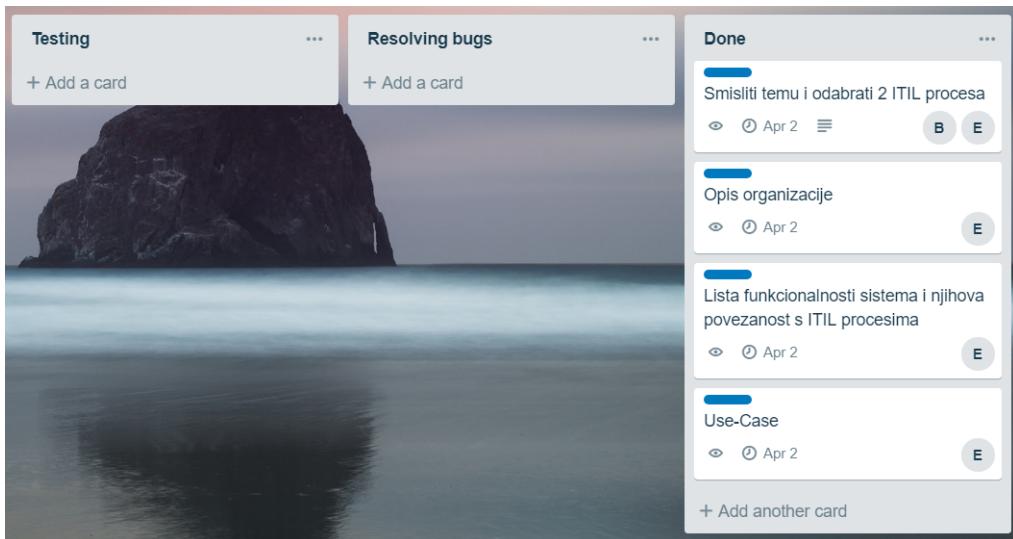
- *To-do* - skup svih zadataka koje je potrebno izvršiti;
- *In progress* - skup zadataka koji su ocijenjeni najvažnijim po određenom kriteriju (vrijeme isporuke ili kritičnost za sistem), te koji se trenutno izvršavaju;
- *Testing* - skup zadataka koji su implementirani no potrebno je izvršiti njihovu verifikaciju;
- *Resolving bugs* - skup zadataka iz prethodnih inkremenata u kojima su pronađene greške, te koje se trenutno ispravljaju;
- *Done* - skup svih zadataka koji su uspješno završeni.

Zadaci se dijele u tri skupine:

1. *Documentation*, što se odnosi na pisanje dokumentacije vezane za projekat;
2. *Backend*, što se odnosi na implementaciju funkcionalnosti te persistencije podataka;
3. *Frontend*, što se odnosi na implementaciju korisničkog interfejsa.

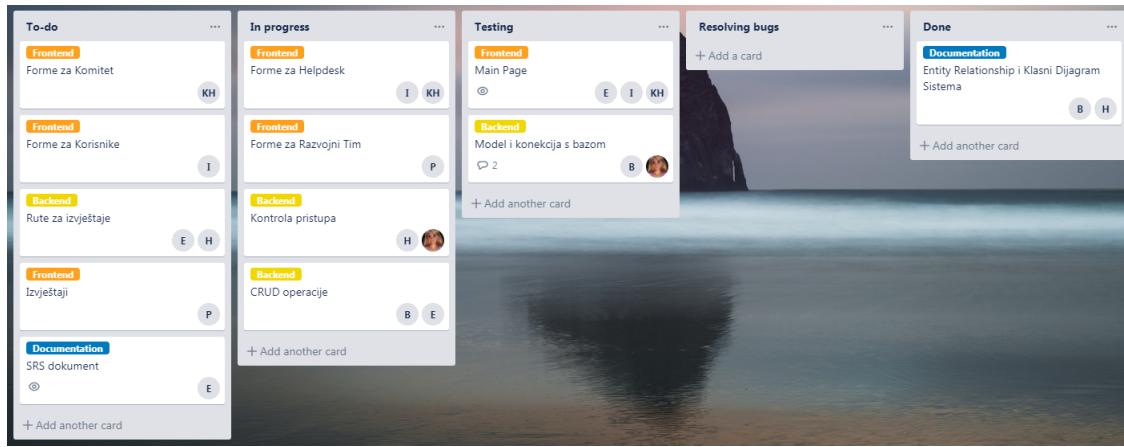


Slika 3: *Trello board* projekta, zadaci koji se trebaju izvršiti i koji se trenutno izvršavaju



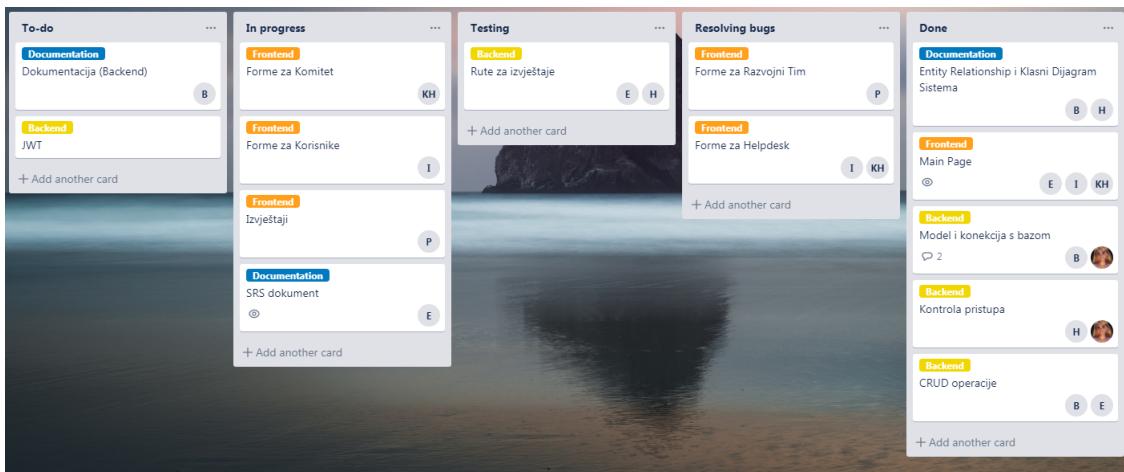
Slika 4: *Trello board* projekta, zadaci koji su završeni te dijelovi sistema koji se testiraju ili popravljaju

Prvi sprint razvoja aplikacije podrazumijevao je postojanje dva odvojena tima za razvoj. Timovi koji su razvijali *backend* i *frontend* dijelove aplikacije radili su odvojeno, budući da je konfiguriranje sistema, kreiranje CRUD operacija i ruta, kao i kreiranje početnih formi bilo nezavisno, te se na taj način sam proces završio brže i jednostavnije. Za dokumentaciju je bila zadužena jedna osoba, čije je zaduženje bilo koordinisanje timova i kreiranje pomoćne dokumentacije za razvoj. Prvi sprint prikazan je na slici 5, odakle je vidljivo da su zadaci timova bili potpuno različiti i nezavisni jedni od drugih.



Slika 5: Prvi sprint projekta

Drugi sprint razvoja aplikacije podrazumijevao je povezivanje dva dijela projekta koji su prethodno razvijeni. U okviru ovog sprinta *frontend* tim izvršio je povezivanje na prethodno kreirane servise *backend* tima, dok je *backend* tim kreirao dodatne rute neophodne za prikaz izvještaja sistema. U ovom sprintu kreirana je formalna dokumentacija sistema koja opisuje funkcionalne i nefunkcionalne zahtjeve, prethodno kreiranu arhitekturu i korištene tehnologije, kao i sve ostale dijelove od kojih se ovaj dokument sastoji. Drugi sprint prikazan je na slici 6.



Slika 6: Drugi sprint projekta

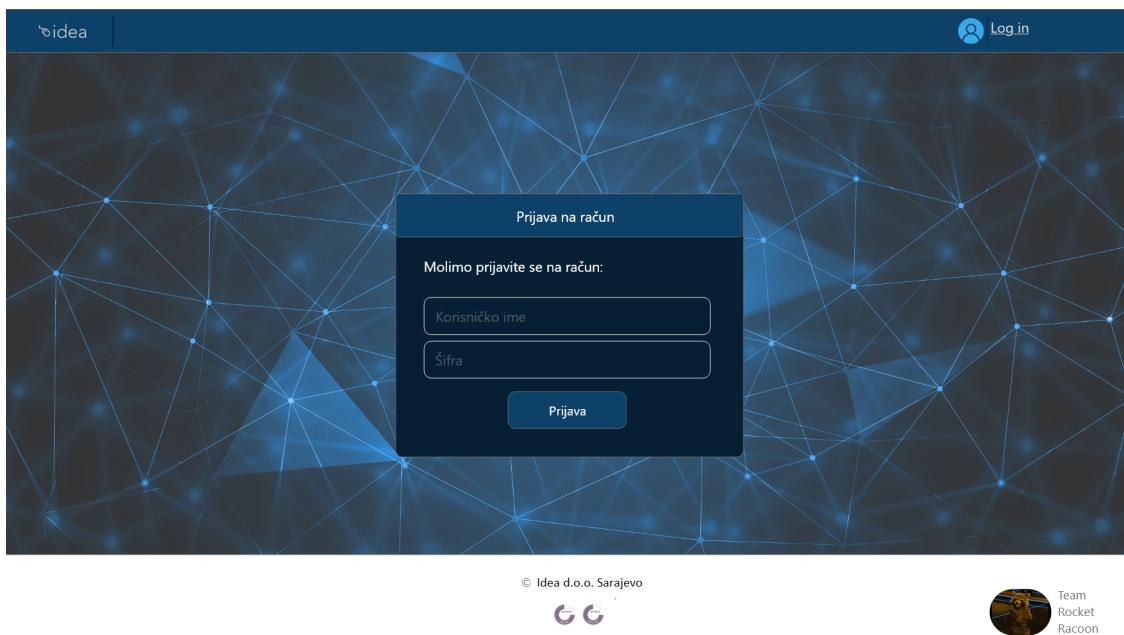
## 2.5 Prototip sistema

Na Slici 7 prikazan je osnovni izgled ekrana aplikacije, koja ima korisnički interfejs kreiran poštujući pravila dobrog dizajna.



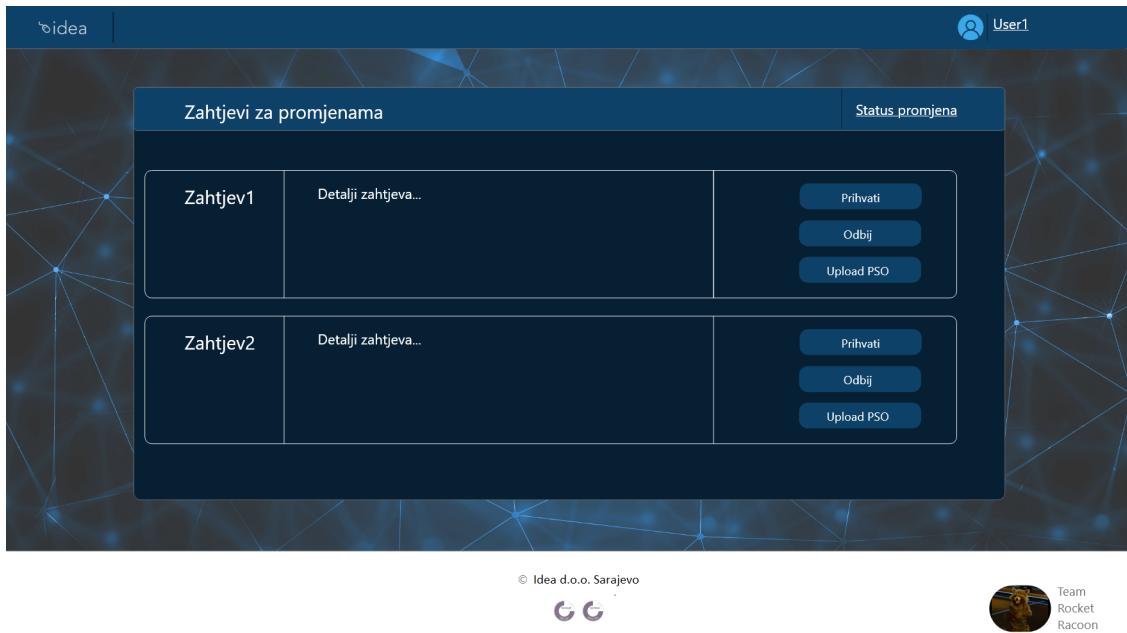
Slika 7: Osnovni izgled ekrana aplikacije

Na Slici 8 prikazan je izgled interfejsa za unos korisničkih podataka, nakon čega se omogućava pristup aplikaciji za različite vrste korisnika.



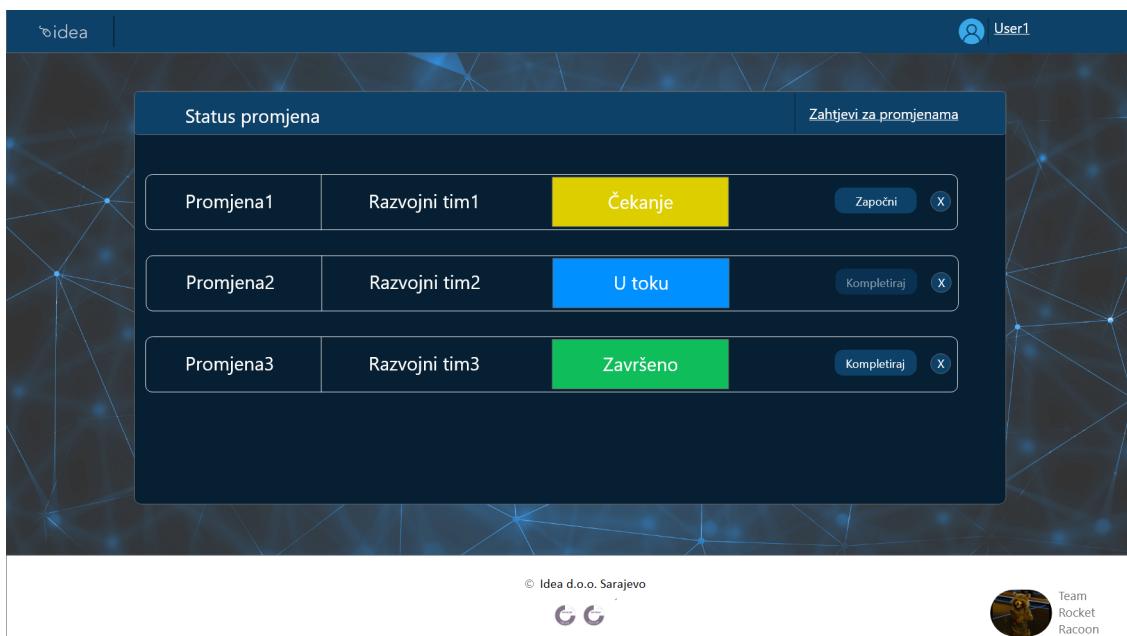
Slika 8: Izgled interfejsa za unos korisničkih podataka

Na Slici 9 prikazan je izgled ekrana za pregled svih zahtjeva za promjenama, koje može vršiti tim za upravljanje promjenama. Nakon pregleda promjene, vrši se njena analiza i *upload-uje* se PSO dokument, te se pritiskom na dugme **Prihvati** promjena automatski šalje komitetu za promjene. Alternativno, promjena se može odmah odbiti.



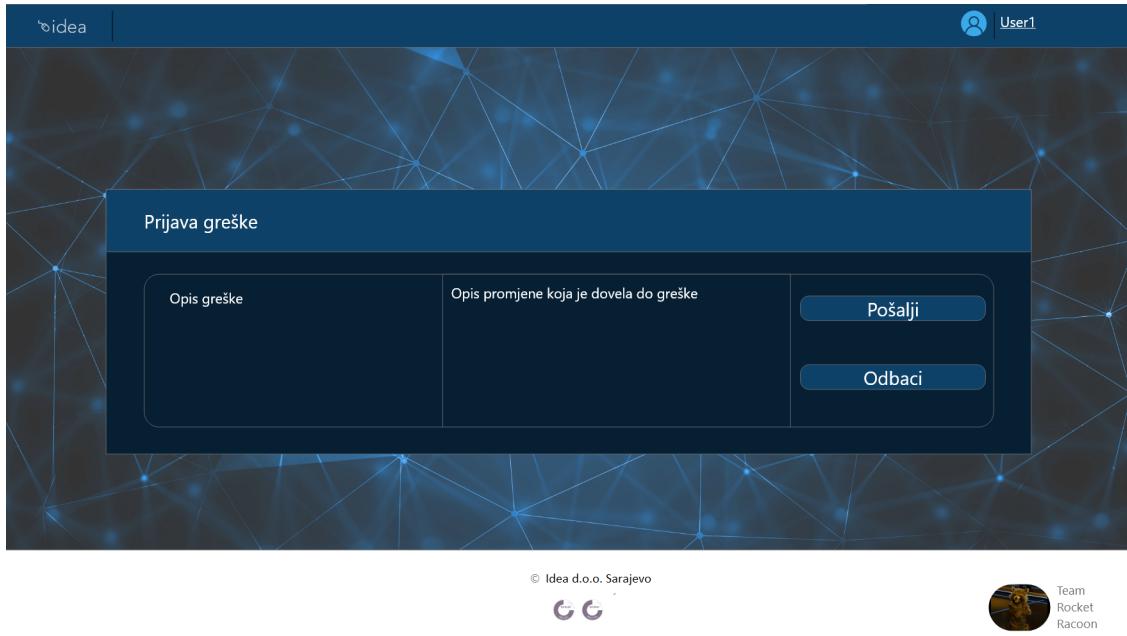
Slika 9: Izgled interfejsa za pregled i procesiranje promjena

Na Slici 10 prikazan je izgled ekrana za pregled svih odobrenih promjena, koje može vršiti tim za upravljanje promjenama. Promjene se mogu poslati razvojnog timu, te se vršiti njihov kontinualni nadzor i otkazivanje, ukoliko se pokaže da je promjenu nemoguće, vrlo teško ili neisplativo ugraditi u sistem.



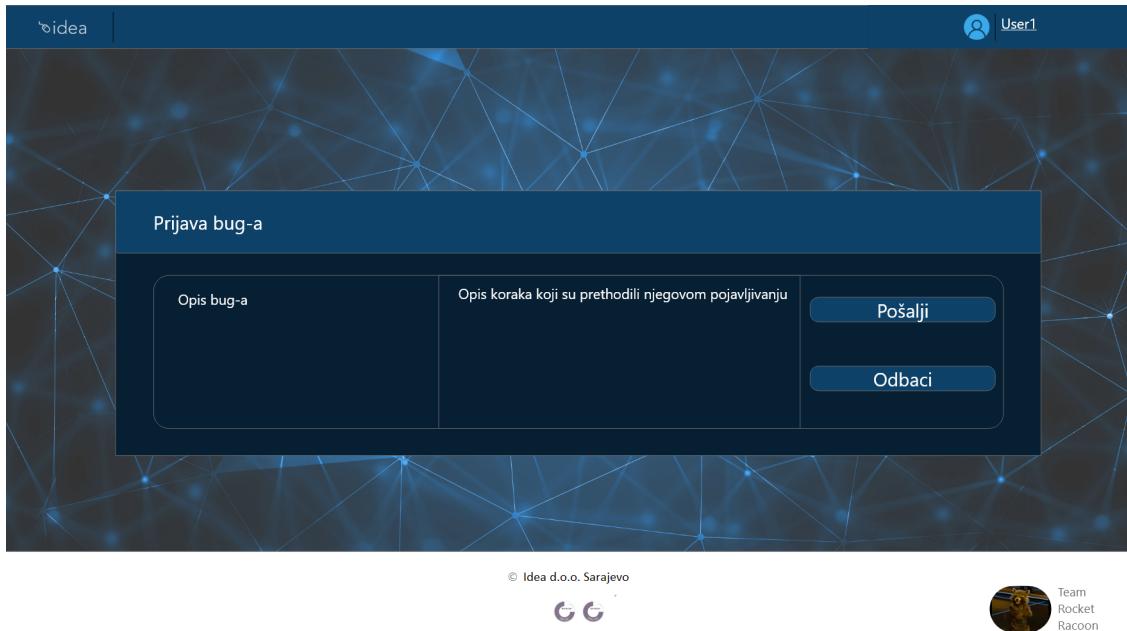
Slika 10: Izgled interfejsa za pregled statusa odobrenih promjena

Na Slici 11 prikazan je izgled ekrana za prijavu greške od strane korisnika sistema koji se održava. Nakon unosa neophodnih informacija, greške se evidentiraju u sistemu kao događaji koje zatim pregleda tim za upravljanje promjenama te na koje adekvatno reaguje.



Slika 11: Izgled interfejsa za prijavu greške u sistemu

Na Slici 12 prikazan je izgled ekrana za prijavu *bug-a* od strane razvojnog tima. Nakon unosa neophodnih informacija, *bug-ovi* se evidentiraju u sistemu kao događaji koje zatim pregleda tim za upravljanje promjenama te na koje adekvatno reaguje.



Slika 12: Izgled interfejsa za prijavu *bug-a* u sistemu

### 3 Funkcionalni zahtjevi

#### 3.1 FZ 1. Prijava na sistem

##### 3.1.1 Uvod

Budući da nije omogućeno javno korištenje sistema, neophodno je izvršiti autorizaciju prije nego se korisniku dozvoli pristup. Osim toga, različite vrste korisnika imaju različite uloge u sistemu, te je potrebno izvršiti provjeru koju ulogu određeni korisnik ima, na osnovu čega se odobrava pristup samo određenim dijelovima sistema.

##### 3.1.2 Preduslovi

Nema preduslova za ovaj funkcionalni zahtjev, budući da je svim korisnicima omogućena prijava na sistem.

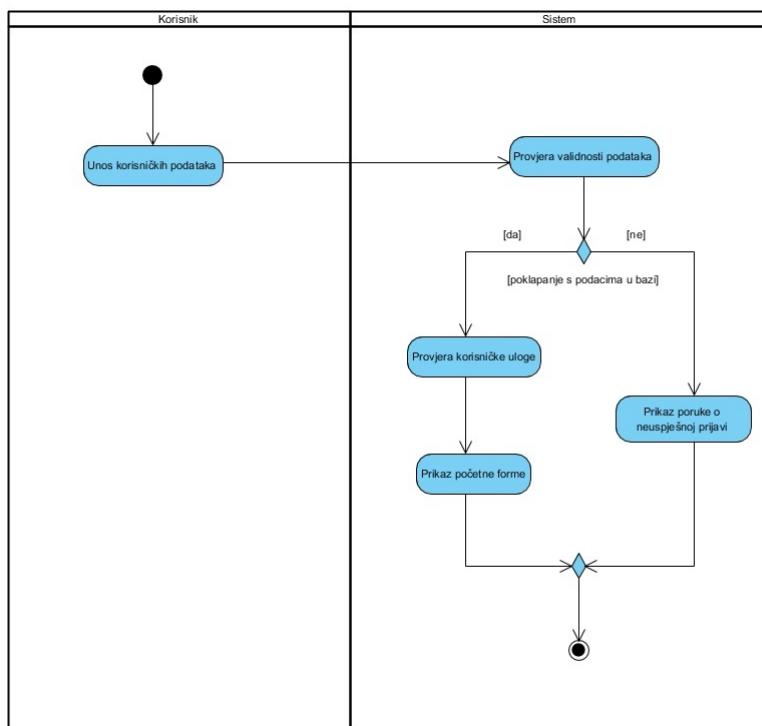
##### 3.1.3 Ulazi

Ulazi u proces prijave na sistem su korisnički podaci korisnika, koji se sastoje od *korisničkog imena* i *šifre*.

##### 3.1.4 Obrada

Nakon što korisnik sistema unese svoje podatke, vrši se njihova usporedba s podacima iz baze podataka. Radi sigurnosti, u bazi se šifre čuvaju kao *hash*-iranе, te se pri provjeri validnosti podataka vrši *hash*-iranje unesene šifre i usporedba s dobavljenim podacima iz baze. Ukoliko se pronade poklapanje korisničkog imena i šifre, sistem preuzima podatke o ulozi korisnika u sistemu, te ovisno o ulozi korisnika preusmjerava na određenu početnu formu.

Cijeli proces prikazan je na Slici 13 u vidu dijagrama aktivnosti.



Slika 13: Proces prijave na sistem

### 3.1.5 Izlazi

Postoje dva moguća izlaza iz sistema - u slučaju unosa ispravnih korisničkih podataka, korisniku se omogućava pristup sistemu i preusmjerava se na odgovarajuću početnu formu, dok se u suprotnom slučaju prikazuje poruka o neispravnom unosu.

### 3.1.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je visok, budući da je vrlo važno omogućiti svim korisnicima prijavu na svoje korisničke račune.

## 3.2 FZ 2. Odjava sa sistema

### 3.2.1 Uvod

Nakon što korisnik završi s korištenjem sistema, može se odjaviti s istog. Svim korisnicima sistema omogućeno je vršenje odjave sa sistema na isti način. Zbog sigurnosti, nakon vršenja odjave sa sistema završava se korisnička sesija, te se onemogućava pristup sistemu dok se ponovo ne unesu ispravni korisnički podaci.

### 3.2.2 Preduslovi

Da bi se korisnik mogao odjaviti, prvo mora biti prijavljen na svoj korisnički račun.

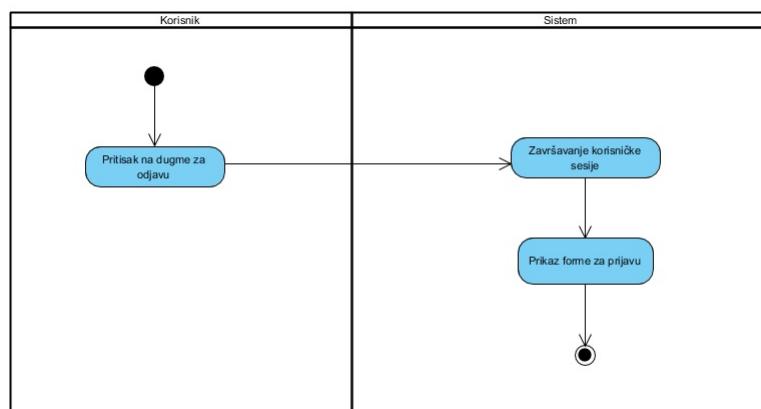
### 3.2.3 Ulazi

Ulaz u sistem predstavlja pritisak na dugme za odjavu.

### 3.2.4 Obrada

Nakon što korisnik pritisne na dugme za odjavu, sistem vrši završavanje korisničke sesije, odnosno onemogućava datom korisniku dalje korištenje sistema. Nakon toga, korisnik se preusmjerava na formu za prijavu.

Cijeli proces prikazan je na Slici 14 u vidu dijagrama aktivnosti.



Slika 14: Proces odjave sa sistema

### 3.2.5 Izlazi

Izlaz iz sistema je preusmjeravanje na formu za prijavu.

### 3.2.6 Prioritet realizacije

Prioritet realizacije ovog zahtjeva je visok, jer neuspješna odjava sa sistema omogućava zlonamjernim korisnicima zloupotrebu aktivne sesije.

## 3.3 FZ 3. Slanje zahtjeva za promjenom

### 3.3.1 Uvod

Korisnici aplikacija koje firma Idea razvija mogu poslati zahtjeve za promjenama. Kako bi to izvršili, potrebno je unijeti odgovarajuće podatke o zahtjevu te poslati isti kako bi ga nakon toga nadležna lica mogla obraditi.

### 3.3.2 Preduslovi

Slanje zahtjeva za promjenom omogućeno je samo korisnicima koji su prijavljeni kao korisnici softvera koje firma Idea razvija.

### 3.3.3 Ulazi

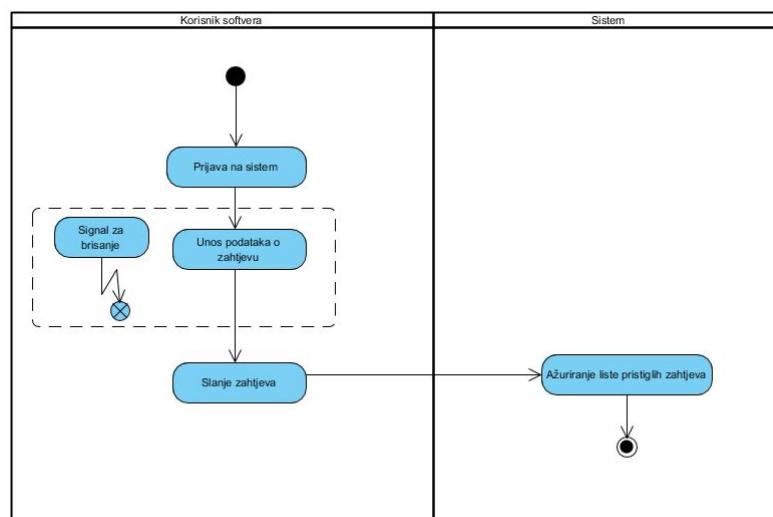
Ulez u sistem predstavlja sam zahtjev za promjenom, koji je opisan putem sljedećih podataka:

- Naziv zahtjeva;
- Opis zahtjeva;
- Vrsta zahtjeva (jedna od predefinisanih vrijednosti iz liste: nova funkcionalnost, izmjena postojeće aktivnosti).

### 3.3.4 Obrada

Nakon što se korisnik softvera prijavi na sistem, treba unijeti podatke o zahtjevu i odabrati opciju za slanje. U svakom trenutku prije slanja omogućeno je brisanje zahtjeva.

Cijeli proces prikazan je na Slici 15 u vidu dijagrama aktivnosti.



Slika 15: Proces slanja zahtjeva za promjenom

### **3.3.5 Izlazi**

Izlaz iz sistema predstavlja poruka o uspješnom slanju zahtjeva za promjenom.

### **3.3.6 Prioritet realizacije**

Prioritet realizacije ovog funkcionalnog zahtjeva je nizak, jer je slanje zahtjeva za promjenom proces koji se može odložiti u slučaju nedostupnosti sistema.

## **3.4 FZ 4. Prijava grešaka**

### **3.4.1 Uvod**

Korisnici sistema kojima je omogućeno prijavljivanje grešaka su korisnici softvera i članovi tima za razvoj i produkciju. Korisnici sistema mogu prijavljivati greške pri korištenju softvera, dok članovi razvojnog tima mogu prijaviti *bug-ove* (funkcionalnosti koje ne rade ispravno te koje je potrebno popraviti) i nemogućnost implementacije nekog od zahtjeva za promjene. U svim slučajevima, prijave se proslijeduju *helpdesk-u* koji vrši njihovo raspoređivanje.

### **3.4.2 Preduslovi**

Prijava grešaka omogućena je korisnicima softvera koje firma Idea razvija i članovima razvojnog tima firme.

### **3.4.3 Ulazi**

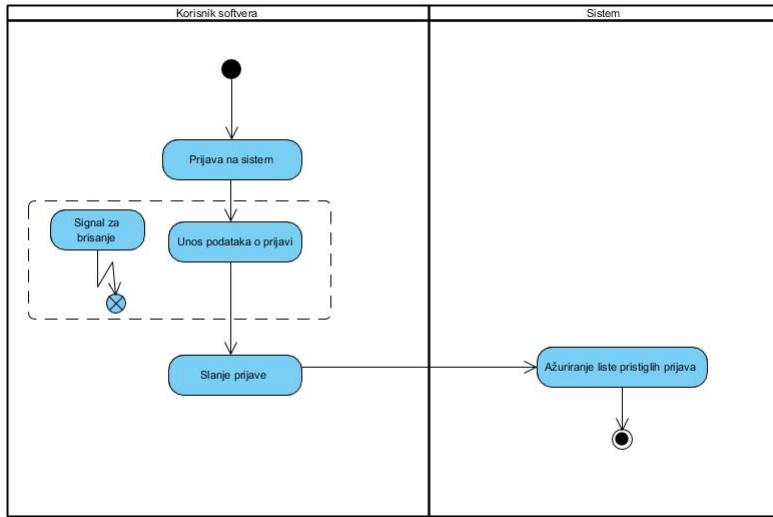
Ulas u sistem predstavlja prijava greške, koja je opisana putem sljedećih podataka:

- Opis greške;
- Opis načina na koji se dolazi do greške;
- Vrsta greške (u slučaju razvojnog tima, moguće je grešku klasificirati kao *bug* ili kao nemogućnost implementacije nekog zahtjeva).

### **3.4.4 Obrada**

Nakon što se korisnik prijavi na sistem, treba unijeti podatke o grešci i odabrati opciju za slanje. U svakom trenutku prije slanja omogućeno je brisanje prijave.

Cijeli proces prikazan je na Slici 16 u vidu dijagrama aktivnosti.



Slika 16: Proces slanja prijave greške

### 3.4.5 Izlazi

Izlaz iz sistema predstavlja poruka o uspješnom slanju prijave greške.

### 3.4.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je srednji, jer je važno ispraviti greške što prije kako bi korisnici mogli nesmetano koristiti aplikacije firme Idea.

## 3.5 FZ 5. Pregled svih prijava

### 3.5.1 Uvod

Članovi razvojnog tima mogu pregledati listu svih pristiglih prijava koje su im dodijeljene od strane *helpdesk-a*. U okviru liste koju mogu pregledati nalaze se greške u sistemu (koje su prijavili korisnici ili članovi razvojnog tima), kao i zahtjevi koje je potrebno implementirati. Nakon uspješne implementacije ili popravljanja grešaka, omogućeno je njihovo arhiviranje.

### 3.5.2 Preduslovi

Pregled svih prijava omogućen je članovima razvojnog tima, budući da je lista prijava prilagođena aktivnostima koje razvojni tim vrši (implementacija novih zahtjeva i ispravljanje grešaka). Da bi se prijava prikazala u listi, potrebno je da je *helpdesk* prvo rasporedi datom razvojnom timu.

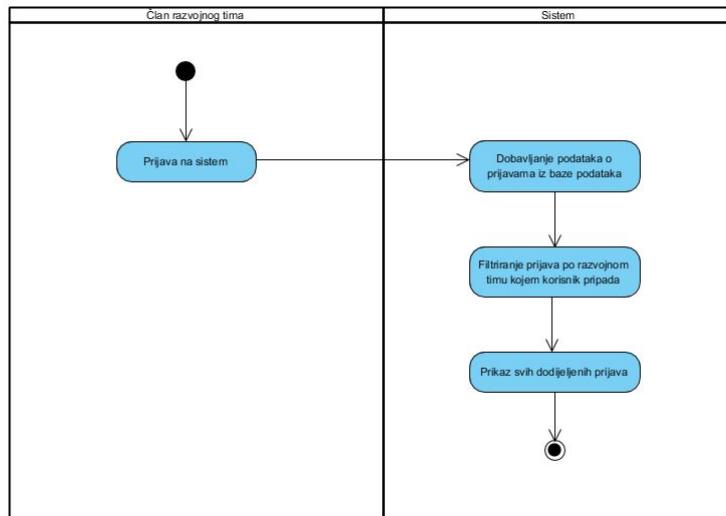
### 3.5.3 Ulazi

Nema ulaza u sistem - nakon uspješne prijave, svim članovima razvojnog tima omogućen je isti pregled svih prijava.

### 3.5.4 Obrada

Obrada zahtjeva sastoji se u dobavljanju svih podataka o prijavama te vršenje njihovog filtriranja ovisno o tome kojem razvojnom timu korisnik pripada.

Cijeli proces prikazan je na Slici 17 u vidu dijagrama aktivnosti.



Slika 17: Proces pregleda svih prijava

### 3.5.5 Izlazi

Izlaz iz sistema predstavlja prikaz liste svih dodijeljenih prijava sa svim pripadajućim podacima.

### 3.5.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je srednji, budući da je neophodno što prije prikazati prijave timu za produkciju kako bi se greške mogle ispraviti.

## 3.6 FZ 6. Arhiviranje prijava

### 3.6.1 Uvod

Članovi razvojnog tima vrše popravljanje prijavljenih grešaka te implementiranje dodijeljenih zahtjeva za promjenama. Nakon uspješnog vršenja implementacije, vrše arhiviranje istih kako bi se uklonile s liste aktivnih prijava.

### 3.6.2 Preduslovi

Arhiviranje prijava omogućeno je isključivo članovima tima za razvoj. Da bi se prijava arhivirala, prvo se mora početi s njenom implementacijom (njen status mora biti *U toku*).

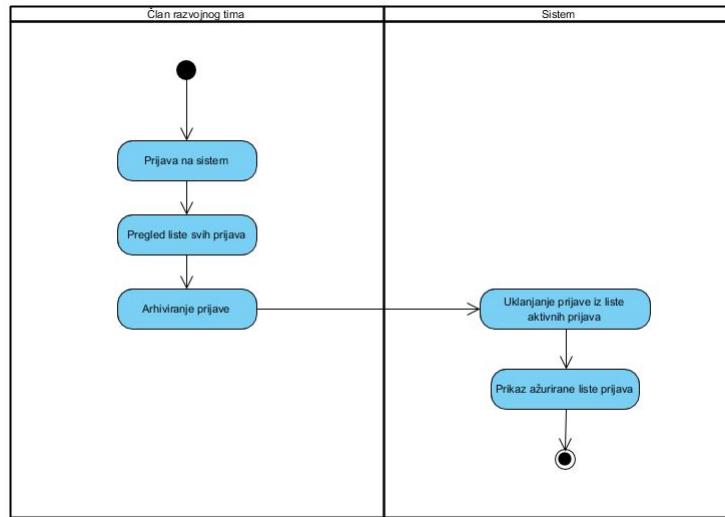
### 3.6.3 Ulazi

Ulaz u sistem je pritisak na dugme za arhiviranje prijave.

### 3.6.4 Obrada

Nakon što član razvojnog tima pritisne na dugme za arhiviranje, sistem vrši označavanje prijave arhiviranim te je uklanja iz liste aktivnih prijava.

Cijeli proces prikazan je na Slici 18 u vidu dijagrama aktivnosti.



Slika 18: Proces arhiviranja prijave

### 3.6.5 Izlazi

Izlaz iz sistema je ažurirana lista prijava, koja ne sadrži arhiviranu prijavu.

### 3.6.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je nizak, budući da arhiviranje implementirane prijave nije prioritetna akcija.

## 3.7 FZ 7. Pregled svih pristiglih zahtjeva

### 3.7.1 Uvod

Nakon što korisnik pošalje zahtjev za promjenom i *helpdesk* napravi studiju izvodljivosti i odobri zahtjev, on se šalje komitetu za promjene na pregled i konačno odlučivanje. Iz tog razloga komitet za promjene treba imati omogućen pregled svih pristiglih zahtjeva o kojima je potrebno odlučivati.

### 3.7.2 Preduslovi

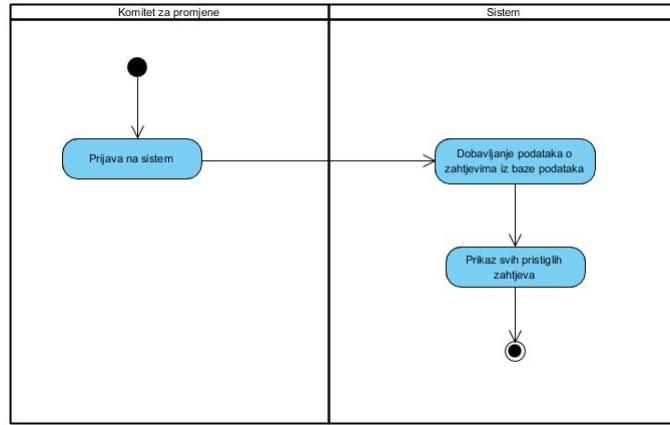
Pregled svih pristiglih zahtjeva omogućen je samo komitetu za promjene. Da bi se zahtjev prikazao u listi, *helpdesk* ga prvo mora odobriti i priložiti PSO dokument za komitet.

### 3.7.3 Ulazi

Nema ulaza u sistem - nakon uspješne prijave, svim članovima komiteta za promjene omogućen je isti pregled svih pristiglih zahtjeva.

### 3.7.4 Obrada

Obrada zahtjeva sastoji se u dobavljanju svih podataka o zahtjevima koje je *helpdesk* odobrio. Cijeli proces prikazan je na Slici 19 u vidu dijagrama aktivnosti.



Slika 19: Proces pregleda svih pristiglih zahtjeva

### 3.7.5 Izlazi

Izlaz iz sistema predstavlja prikaz liste svih dodijeljenih prijava sa svim pripadajućim podacima.

### 3.7.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je srednji, budući da je potrebno što prije pregledati novopristigle zahtjeve, no ukoliko bi sistem bio nedostupan, ta akcija bi mogla biti odložena.

## 3.8 FZ 8. Odlučivanje po zahtjevu

### 3.8.1 Uvod

Nakon pregleda svih informacija o pristiglom zahtjevu (uključujući detalje zahtjeva i PSO dokument), komitet za promjene donosi konačnu odluku. Ovisno o tome kakva je konačna odluka, zahtjev se ili arhivira, ili vraća u listu za *helpdesk* kako bi se moglo početi s njegovom implementacijom.

### 3.8.2 Preduslovi

Odlučivanje po zahtjevima omogućeno je samo komitetu za promjene. Da bi se odlučivalo o nekom zahtjevu, on prvo mora biti u listi pristiglih zahtjeva (mora biti odobren od strane *helpdesk*-a).

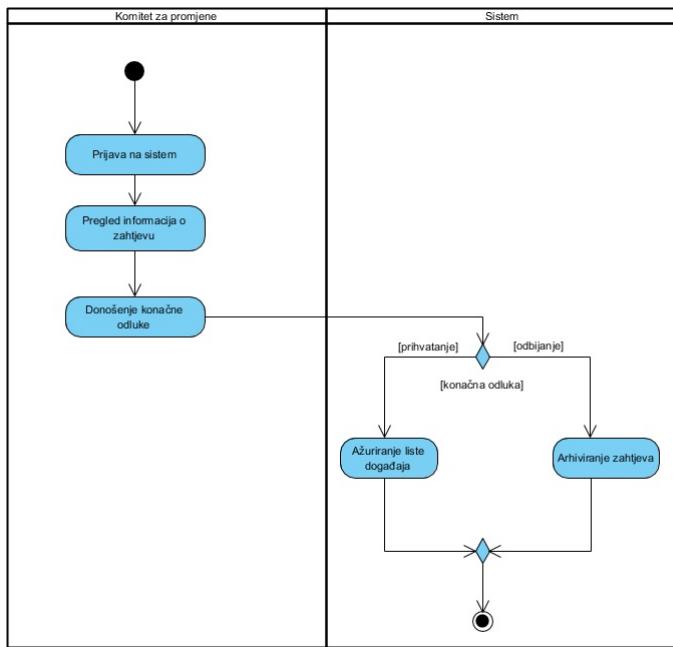
### 3.8.3 Ulazi

Ulaz u sistem predstavlja konačna odluka komiteta - odobravanje ili odbijanje zahtjeva.

### 3.8.4 Obrada

Nakon unosa konačne odluke o zahtjevu (pritiska na dugme za odobravanje ili odbijanje zahtjeva), sistem vrši ažuriranje liste događaja *helpdesk*-a ili arhiviranje zahtjeva.

Cijeli proces prikazan je na Slici 20 u vidu dijagrama aktivnosti.



Slika 20: Proces odlučivanja po zahtjevu

### 3.8.5 Izlazi

Izlaza iz sistema nema - nakon što se odluči po zahtjevu, isti se uklanja iz liste pristiglih zahtjeva tako da član komiteta za promjene može postupati po drugim zahtjevima.

### 3.8.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je srednji, budući da je potrebno što prije poslati zahtjev na implementaciju (ukoliko se odobri), no to nije akcija s najvišim prioritetom.

## 3.9 FZ 9. Pregled svih događaja u sistemu

### 3.9.1 Uvod

Članovi *helpdesk*-a trebaju vršiti raspoređivanje zahtjeva i prijava odgovarajućim korisnicima sistema na dalju obradu. Iz tog razloga komitet za promjene treba imati omogućen pregled svih događaja (pristiglih zahtjeva i prijava) koje je potrebno raspoređivati.

### 3.9.2 Preduslovi

Pregled svih događaja u sistemu omogućen je samo *helpdesk*-u.

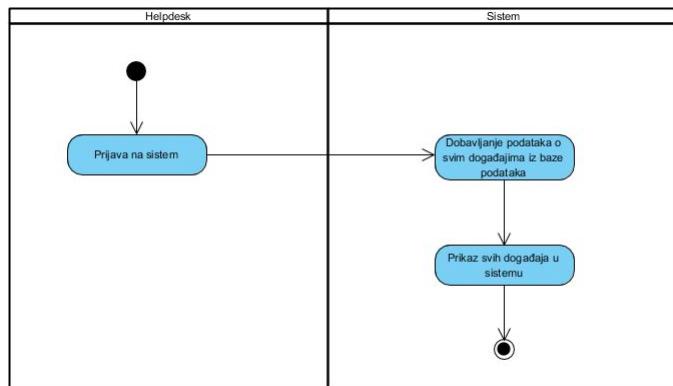
### 3.9.3 Ulazi

Nema ulaza u sistem - nakon uspješne prijave, svim članovima *helpdesk*-a omogućen je isti pregled svih događaja u sistemu.

### 3.9.4 Obrada

Obrada zahtjeva sastoji se u dobavljanju svih podataka o događajima (svi pristigli zahtjevi, odobreni zahtjevi te prijave korisnika i članova razvojnog tima).

Cijeli proces prikazan je na Slici 21 u vidu dijagrama aktivnosti.



Slika 21: Proces pregleda svih događaja u sistemu

### 3.9.5 Izlazi

Izlaz iz sistema predstavlja prikaz liste svih događaja u sistemu sa svim pripadajućim podacima.

### 3.9.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je visok, jer *helpdesk* mora moći nesmetano vršiti raspoređivanje događaja, a preduslov za to je mogućnost pregledanja svih događaja.

## 3.10 FZ 10. Slanje zahtjeva na konačno odlučivanje

### 3.10.1 Uvod

Nakon što korisnik softvera pošalje zahtjev, on se prikazuje *helpdesk*-u u listi svih događaja u sistemu. Nakon toga članovi *helpdesk*-a vrše procjenu izvodljivosti koja se, zajedno sa svim ostalim relevantnim informacijama, nalazi u PSO dokumentu. Tada se zahtjev šalje komitetu za promjene na konačno odlučivanje.

### 3.10.2 Preduslovi

Slanje zahtjeva na konačno odlučivanje omogućeno je samo *helpdesk*-u. Zahtjev prvo treba ispuniti uslove za odobravanje, odnosno studija izvodljivosti treba dati pozitivne rezultate.

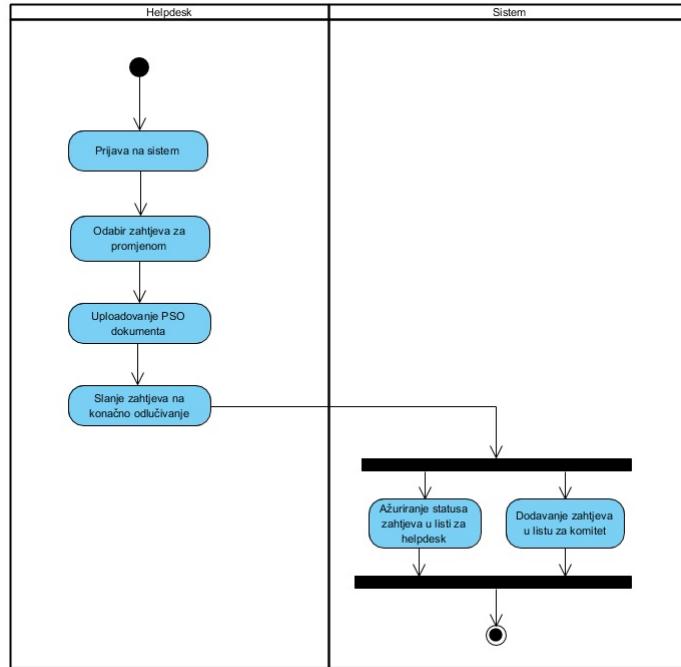
### 3.10.3 Ulazi

Ulas u sistem predstavlja PSO dokument, koji je neophodno priložiti uz zahtjev kako bi se isti poslao na konačno odlučivanje. PSO dokument prilaže se u obliku PDF dokumenta.

### 3.10.4 Obrada

Nakon uspješne prijave i odabira zahtjeva iz liste svih događaja u sistemu, član *helpdesk*-a prilaže PSO dokument i vrši pritisak na dugme za slanje zahtjeva na konačno odlučivanje. Nakon toga se ažurira status zahtjeva u listi koju *helpdesk* vidi, te se zahtjev dodaje u listu pristiglih zahtjeva o kojima komitet treba odlučivati.

Cijeli proces prikazan je na Slici 22 u vidu dijagrama aktivnosti.



Slika 22: Proces slanja zahtjeva na konačno odlučivanje

### 3.10.5 Izlazi

Izlaz iz sistema predstavlja prikaz ažurirane liste svih događaja u sistemu, u kojoj poslani zahtjev ima promijenjen status.

### 3.10.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je nizak, budući da slanje zahtjeva na konačno odlučivanje nije hitno i može se odložiti u slučaju nedostupnosti sistema.

## 3.11 FZ 11. Raspoređivanje događaja

### 3.11.1 Uvod

*Helpdesk* predstavlja skup korisnika sistema koji mogu pregledati sve događaje u sistemu, i čija je dužnost njihovo raspoređivanje onim korisnicima sistema koji trebaju postupati po njima. Jedna od najvažnijih funkcionalnosti ovih korisnika je olakšavanje rada ostalim korisnicima, jer će svi vidjeti samo one događaje koji su za njih relevantni.

### 3.11.2 Preduslovi

Raspoređivanje događaja može vršiti samo *helpdesk*.

### 3.11.3 Ulazi

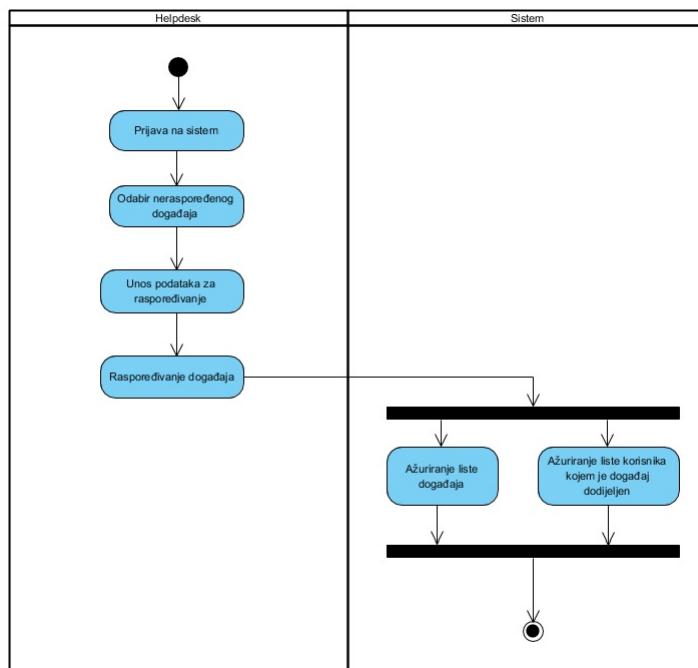
Postoje sljedeći ulazi u sistem:

- Odabir vrste korisnika kojem se dati događaj treba dodijeliti;
- Odabir prioriteta događaja;
- Odabir statusa događaja.

### 3.11.4 Obrada

Nakon prijave na sistem, član *helpdesk-a* odabire događaj koji je potrebno rasporediti. Nakon unosa neophodnih podataka i pritiska na dugme za raspoređivanje, prikazuje se ažurirana lista događaja na kojoj je prikazano kome je događaj dodijeljen i kakav je njegov status.

Cijeli proces prikazan je na Slici 23 u vidu dijagrama aktivnosti.



Slika 23: Proces raspoređivanja događaja

### 3.11.5 Izlazi

Izlaz iz sistema predstavlja prikaz ažurirane liste svih događaja u sistemu, u kojoj raspoređeni događaj ima promijenjen status.

### 3.11.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je visok, jer *helpdesk* treba u svakom trenutku moći izvršiti raspoređivanje događaja (poput grešaka zbog kojih su aplikacije nedostupne korisnicima) adekvatnim korisnicima sistema.

## 3.12 FZ 12. Pregled izvještaja

### 3.12.1 Uvod

Članovi *helpdesk-a*, osim mogućnosti pregleda svih trenutno aktivnih događaja u sistemu, trebaju biti u mogućnosti pregledati i historijat svih događaja. To uključuje sve događaje arhivirane od strane članova tima za razvoj (greške koje su uspješno ispravljene, promjene koje su uspješno implementirane) ili od strane komiteta za promjene (promjene koje su odbačene), posebno ukoliko je potrebno pregledati sve faze u kojima su se neki zahtjevi ili prijava nalazili od početka do kraja svoje aktivnosti u sistemu.

### 3.12.2 Preduslovi

Pregled izvještaja omogućen je samo *helpdesk*-u.

### 3.12.3 Ulazi

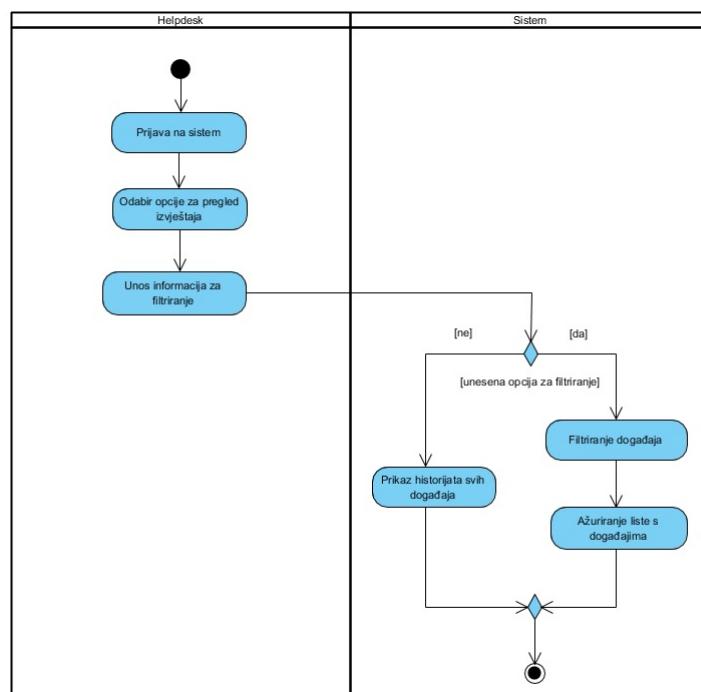
Ulaz u sistem predstavljaju podaci o događajima prema kojima se historijat treba filtrirati:

- Vrsta događaja (zahtjev ili prijava);
- Status događaja (na čekanju, u obradi, završen);
- Datum aktivnosti.

### 3.12.4 Obrada

Nakon prijave na sistem i odabira opcije za pregled izvještaja, član *helpdesk*-a unosi informacije prema kojima je potrebno filtrirati događaje (ukoliko se ne odabere nijedna opcija za filtriranje, prikazuju se svi događaji u historijatu sistema). Nakon toga lista svih događaja se ažurira i prikazuju se samo relevantni događaji sistema.

Cijeli proces prikazan je na Slici 24 u vidu dijagrama aktivnosti.



Slika 24: Proces pregleda izvještaja i filtriranja događaja

### 3.12.5 Izlazi

Izlaz iz sistema predstavlja prikaz ažurirane liste svih događaja u sistemu, u kojoj su prikazani samo događaji koji ispunjavaju uslov filtriranja.

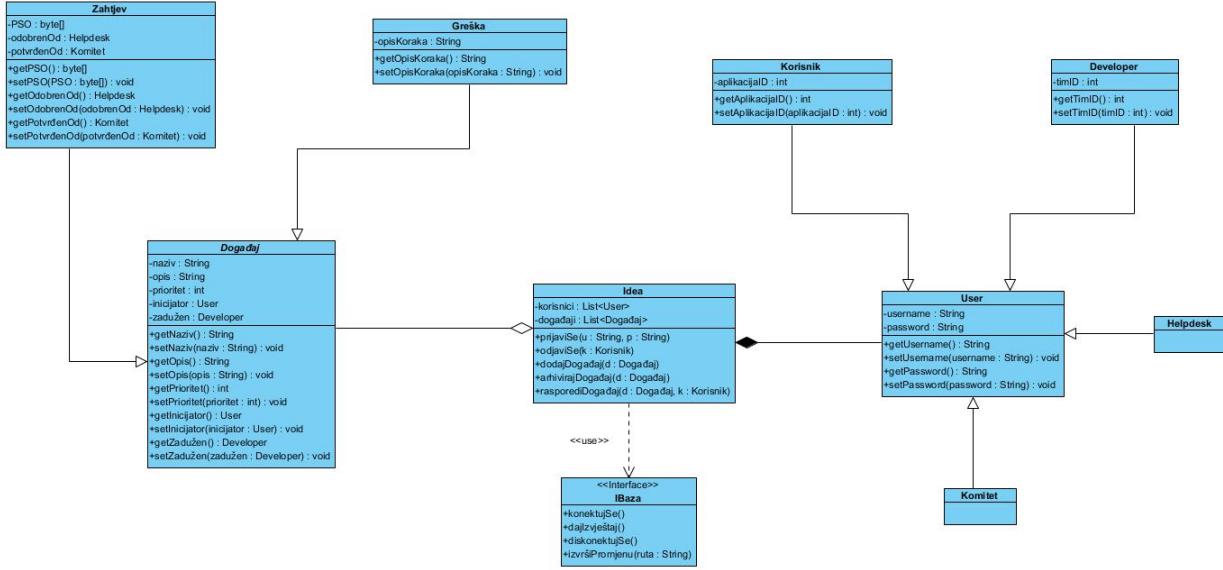
### 3.12.6 Prioritet realizacije

Prioritet realizacije ovog funkcionalnog zahtjeva je nizak, jer je pregled izvještaja statican i nema snažan utjecaj na funkcionisanje sistema.

## 4 Arhitekturalni *stack*

## 4.1 Dijagram klase

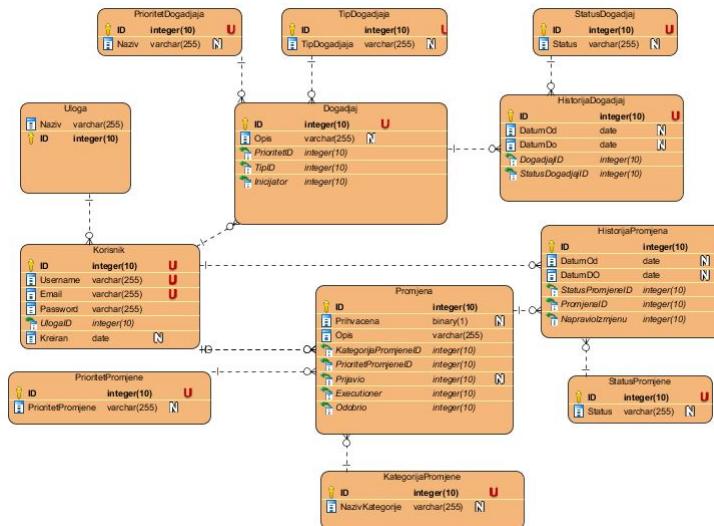
Najjednostavniji način za opis arhitekture sistema je putem dijagrama klasa, na kojem su vidljive najvažniji dijelovi modela (klase) i veze između njih (generalizacija i asocijacija - kompozicija ili agregacija). Ovaj dijagram prikazan je na Slici ??.



Slika 25: Dijagram klasa sistema

#### 4.2 Entity Relationship diagram (ERD)

Još jedan način za opis arhitekture sistema omogućen je putem *entity relationship* dijagrama sistema, u kojem je prikazan model baze podataka koju sistem koristi. Ovaj dijagram prikidan je na Slici 26.



Slika 26: *Entity relationship* dijagram sistema

## 5 Tehnološki *stack*

### 5.1 Opis korištenih tehnologija

Pri razvoju sistema korištene su sljedeće tehnologije:

- **Frontend:**

*Bootstrap* za generisanje izgleda formi (koristeći HTML i CSS);  
*Angular framework* za izgradnju web-aplikacije;

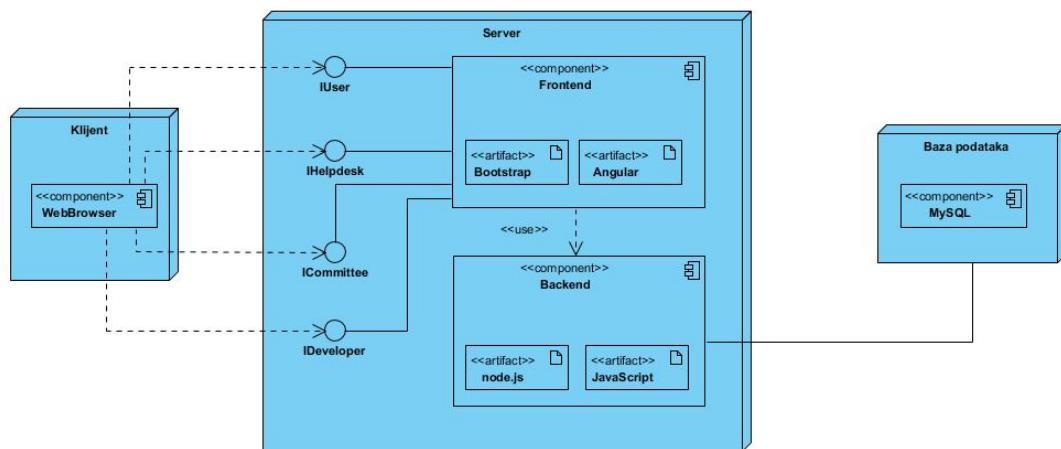
- **Backend:**

*Node.js* tehnologija (programski jezik: *JavaScript*);  
*MySQL* baza podataka;

- **Dokumentacija:**

*LaTex* za generisanje SRS dokumenta;  
*Visual Paradigm* za kreiranje UML dijagrama.

Tehnološki *stack* sistema najlakše je predstaviti koristeći dijagram raspoređivanja, koji je prikazan na Slici 27. Na dijagramu je vidljivo da korisnik koristi *interface*-e koje pruža *frontend* dio sistema. *Frontend* koristi servise backenda, a *backend* jedini ima pristup bazi podataka. Na dijagramu je prikazano i koji dio sistema koristi koje od tehnologija koje su prethodno navedene.



Slika 27: Dijagram raspoređivanja sistema

## 6 Nefunkcionalni zahtjevi

### 6.1 Odziv sistema

Odziv sistema predstavlja količinu vremena od slanja zahtjeva do početka slanja odgovora sistema. Odziv sistema je veoma važna karakteristika za performanse sistema, te spor odziv (odnosno, dugo vrijeme odziva sistema) uzrokuje slabije performanse sistema. Vrijeme odziva sistema ovisi o mnogo faktora, a primarno o mogućnosti sistema da se nosi s velikim brojem korisnika bez da dođe do zagruženja mreže, te se zbog tog razloga hardverske komponente sistema trebaju prilagoditi na način da omogućavaju normalno funkcioniranje sistema i pri velikom broju zahtjeva od strane korisnika.

Budući da je sistem koji se razvija web-aplikacija koju će koristiti veliki broj, kako zaposlenika firme Idea, tako i kompanija za koje firma Idea razvija aplikacije, neophodno je da odziv sistema bude brz i u slučajevima kada u jednom trenutku bude poslan veliki broj zahtjeva. Ovakav odziv omogućiti će se putem osposobljavanja više kopija istog servisa na koji će se preusmjeravati zahtjevi ovisno o stanju u redovima čekanja (koji trebaju biti približno isto napunjeni).

### 6.2 Propusnost

Propusnost sistema se definiše kao količina informacija koje sistem može obraditi po jedinici vremena, te je također veoma važna za performanse sistema. Ona ovisi o mnogo faktora, uključujući brzinu procesiranja pojedinih informacija, kao i vrijeme odziva sistema, s kojim je propusnost direktno povezana. Budući da se propusnost odnosi na količinu informacija koje se mogu prenijeti preko mreže, pročitati ili napisati na medije za spašavanje podataka ili na broj operacija ulazno-izlaznih (I/O) uređaja, i propusnost sistema ovisi o više hardverskih komponenti, te je potrebno analizirati koja od naprijed navedenih propusnosti je esencijalna za funkcioniranje sistema te se analogno prilagoditi i hardverske komponente.

Propusnost sistema koji se razvija u vezi je s odzivom sistema koji je prethodno definisan - propusnost sistema treba biti velika (veliki broj istovremeno prijavljenih korisnika), što će se omogućiti povećanjem kapaciteta redova čekanja, odnosno povećanjem broja redova čekanja (budući da je skaliranje u širinu jeftinije).

### 6.3 Poruke o greškama

Za normalno funkcioniranje sistema potrebno je predvidjeti da korisnici neće uvijek na korektan način unositi tražene podatke i vršiti zadane operacije na ispravan način te se trebaju omogućiti poruke o greškama. Ukoliko korisnik napravi grešku, sistem istu treba prepoznati i prikazati korisniku adekvatne informacije kako bi se ponavljanje iste greške spriječilo.

Ovaj nefunkcionalni zahtjev ispunjen je tokom razvoja aplikacije, budući da je napravljen intuitivan korisnički *interface* koji je jednostavan za korištenje.

### 6.4 Dostupnost

Dostupnost sistema definiše se kao vjerovatnoća da će poslani zahtjev od strane korisnika rezultovati odgovorom od strane sistema. Čak i mala količina vremena nedostupnosti sistema neprihvatljiva je (jer uzrokuje velike štete, što je dokazano u mnogim studijama) te se stoga

treba osigurati da će sistem biti u stanju da uvijek ispravno funkcioniše, te ukoliko to ne bude moguće, da će se svaki kvar brzo i efikasno otkloniti od strane tima za održavanje, te da će korisnici biti obaviješteni kada i koliko dugo će sistem biti nedostupan (ukoliko je to moguće).

Dostupnost sistema također je u korelaciji s odzivom sistema, kao i s njegovom propusnošću, te još jednom potvrđuje neophodnost kreiranja većeg broja redundantnih kopija istih servisa - što više kopija istog servisa postoji, to je manja mogućnost da će u jednom trenutku sve dostupne kopije biti nedostupne, te da nijedan od servisa neće biti u mogućnosti odgovoriti na korisnički zahtjev.

## 6.5 Skalabilnost

Skalabilnost je sposobnost sistema da se prilagođava sve većim zahtjevima u pogledu resursa i performansi. Pri dizajnu sistema i izbora hardverskih komponenti potrebno je uzeti u obzir informacije o broju korisnika koji će isti koristiti, vrsti usluga koje će sistem pružati te da li je potrebno omogućiti horizontalno ili vertikalno skaliranje (u skladu sa predviđenim budžetom). Budući da je pri dizajnu sistema korištena klijent-server arhitektura, skalabilnost je omogućena odvajanjem funkcionalnosti sistema u zasebne servise koji će se zasebno i skalirati (te time smanjiti i troškovi), te će sistem biti u stanju da funkcioniše i ukoliko jedna od komponenti ne bude funkcionalisala (ili se bude nadogradila), čime se direktno utječe i na dostupnost sistema.

Iako firma Idea ne planira da uskoro širi svoje poslovanje na veći broj firmi (jer je njena primarna djelatnost održavanje postojećih sistema), sistem koji se dizajnira svakako treba biti skalabilan, ukoliko se u budućnosti ipak odluči za ovakvo nešto. Skalabilnost je postignuta putem korištenja servis-klijent arhitekture sistema, koja omogućava dodavanje proizvoljnog broja kopija istog servisa, modifikaciju i nadogradnju određenih servisa bez utjecanja na ostale servise te čak i potpunu nezavisnost o operativnim sistemima koje korisnici koriste, budući da korisnik jedino treba posjedovati *web-browser* kako bi pristupio sistemu.

## 6.6 Dokumentovanost sistema

Za ispravno funkcionisanje sistema neophodno je da isti posjeduje što više što detaljnije dokumentacije, kako bi tim za održavanje u svakom trenutku mogao pronaći relevantne informacije za održavanje konkretnog servisa ili sistema kao cjeline. Osim dokumentacije za tim za održavanje, i korisnici trebaju imati adekvatnu dokumentaciju u obliku uputstava za korištenje, kako bi u svakom trenutku mogli pronaći relevantna uputstva za funkcionalnosti sistema koje žele koristiti. Uputstva za korištenje trebaju biti precizna i jasna, te koristiti terminologiju prilagođenu korisnicima sistema, kako ih ne bi zbunjivala te kako bi se rješenje nastalih problema pronašlo što brže i lakše.

Dokumentacija sistema predstavljena je u obliku SRS dokumenta, koji definiše sve bitne funkcionalnosti i ostale detalje o samom sistemu. U budućnosti je planirano i pravljenje korisničkog priručnika, kako bi korisnici mogli pronaći odgovore na sva eventualna pitanja i rješenja za sve nedoumice pri korištenju sistema.

## 6.7 Autorizacija korisnika

Pristup neautorizovanim korisnicima treba biti onemogućen, te se iz tog razloga pri ulasku u sistem korisnici trebaju autorizovati, a to će uraditi putem unošenja ispravnih i jedinstvenih

korisničkih podataka za koje će se izvršiti provjera unutar sistema. Ukoliko se uneseni korisnički podaci pronađu u sistemu, korisniku će biti omogućen pristup istom, te ukoliko se uneseni korisnički podaci ne pronađu u sistemu, korisniku će se prikazati poruka o neuspješnoj prijavi bez omogućavanja korištenja većine funkcionalnosti sistema.

Problem autorizacije u sistemu riješen je koristeći korisničke uloge i sesije. Postoje predefinirane vrijeme nakon kojeg se korisnik automatski odjavljuje iz sistema, kako bi se smanjila mogućnost zloupotrebe ukoliko se korisnik zaboravi odjaviti sistema. Rute sistema nisu blokirane na način da su samo sakriveni putevi koji vode do stranica za različite vrste korisnika, već se pri pristupu svakoj stranici provjerava uloga korisnika te se pristup dopušta isključivo ukoliko je korisnik autorizovan za pristup. Šifre korisnika u bazi podataka se čuvaju *hash*-irane, tako da eventualno vršenje *SQL Injections* neće ugroziti sigurnost korisnika.

## 7 Korisnički *interface-i*

## 8 Zakonska ograničenja dizajna

Sistem će biti razvijen u skladu s ograničenjima koja su propisana u sljedećim zakonima:

### 1. **Zakon o privrednim društvima Federacije Bosne i Hercegovine:**

*Član 38.* Poslovnom tajnom smatraju se informacije o poslovanju za koje je očito da bi prouzrokovale značajnu štetu društvu ako dođu u posjed trećeg lica bez saglasnosti društva.

*Član 39.* (1) Nadležni organ društva dužan je pismenim aktom odrediti informacije koje imaju karakter poslovne tajne i lica odgovorna za njihovo korištenje i zaštitu. U skladu sa prethodno navedenim članom, zakonska ograničenja predstavljaju i svi pisani akti doneseni od strane vrtića, a kojima se definišu podaci za koje je potrebno onemogućiti pristup svim neovlaštenim licima.

Naprijed navedeni članovi (kao i svi pisani akti doneseni od strane firme Idea, a koji se odnose na podatke koje je potrebno zaštititi od neovlaštenog pristupa) postavljaju ograničenja pristupa različitim vrsta korisnika podacima za koje nisu ovlašteni. Budući da će razvijenoj aplikaciji biti u mogućnosti pristupiti osobe koje nisu zaposlenici firme Idea, sve podatke je potrebno zaštititi putem obavezne autorizacije pri pokušaja pristupa svim funkcionalnostima sistema.

### 2. **Zakon o zaštiti ličnih podataka Bosne i Hercegovine:** *Član 11.* (1) Kontrolor podataka i, u okviru svoje nadležnosti, obrađivač podataka staraju se o bezbjednosti podataka te preduzimaju sve tehničke i organizacione mjere i utvrđuju pravila postupka, koji su neophodni kako bi se sproveo ovaj zakon i drugi propisi u vezi sa zaštitom i tajnošću podataka.

*Član 11.* (2) Kontrolor i obrađivač dužni su da preduzmu mjere protiv neovlašćenog ili slučajnog pristupa ličnim podacima, mijenjanja, uništavanja ili gubitka podataka, neovlašćenog prenosa, drugih oblika nezakonite obrade podataka, kao i mjere protiv zloupotrebe ličnih podataka. Ova obaveza ostaje na snazi i nakon završene obrade podataka.

Naprijed navedeni članovi postavljaju ograničenja na pristup podacima za korisnike sistema. Pregled korisnika će biti onemogućen te će biti poduzete sve mjere zaštite od krađe podataka korisnika (njihovih korisničkih imena i šifri, te svih ostalih relevantnih podataka) od strane zlonamjernih korisnika aplikacije.

Dokument je napisan u skladu sa IEEE 830-1998 standardom (*IEEE Recommended Practice for Software Requirements Specification*).