

Programsko inženjerstvo

Ak. god. 2020./2021.

Terminko za vešeraj

Dokumentacija, Rev. 1

Grupa: Janezi

Voditelj: Jan Grgić

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

Nastavnik: Izv. Prof. Dr. Sc. Vlado Sruk

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
2.1 Opis problema i motivacija	4
2.2 Postojeća slična rješenja	4
2.3 Potencijalno zainteresirani korisnici	7
2.4 Mogućnost prilagodbe rješenja	7
2.5 Opseg projektnog zadatka (zahtjevi sustava)	7
2.6 Nadogradnje projektnog zadatka i planovi za budućnost	8
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	11
3.1.2 Sekvencijski dijagrami	14
3.2 Ostali zahtjevi	15
4 Arhitektura i dizajn sustava	16
4.1 Baza podataka	16
4.1.1 Opis tablica	16
4.1.2 Dijagram baze podataka	17
4.2 Dijagram razreda	18
4.3 Dijagram stanja	19
4.4 Dijagram aktivnosti	20
4.5 Dijagram komponenti	21
5 Implementacija i korisničko sučelje	22
5.1 Korištene tehnologije i alati	22
5.2 Ispitivanje programskog rješenja	23
5.2.1 Ispitivanje komponenti	23
5.2.2 Ispitivanje sustava	23
5.3 Dijagram razmještaja	24

5.4 Upute za puštanje u pogon	25
6 Zaključak i budući rad	26
Popis literature	27
Indeks slika i dijagrama	28
Dodatak: Prikaz aktivnosti grupe	29

1. Dnevnik promjena dokumentacije

Kontinuirano osvježavanje

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	D. Grgić	20.10.2020.
0.2	Promijenjeni dijelovi predloška.	J. Grgić	21.10.2020.
0.3	Prepisan opis projektnog zadatka iz word dokumenta koji je sastavljen ranije radi prijave vlastite teme.	J. Grgić	22.10.2020.
0.4	Dodani dionici u poglavlju 3, navedeni aktori i funkcionalni zahtjevi za neregistrirane/neprijavljene korisnike. Napisani obrasci uporabe za neregistrirane/neprijavljene korisnike.	J. Grgić	23.10.2020.
0.5	Uređen dnevnik sastajanja i podjela poslova u grupi.	J. Grgić	23.10.2020.
0.6	Dodani funkcionalni zahtjevi za Administratora.	I. Joskić	23.10.2020.

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.

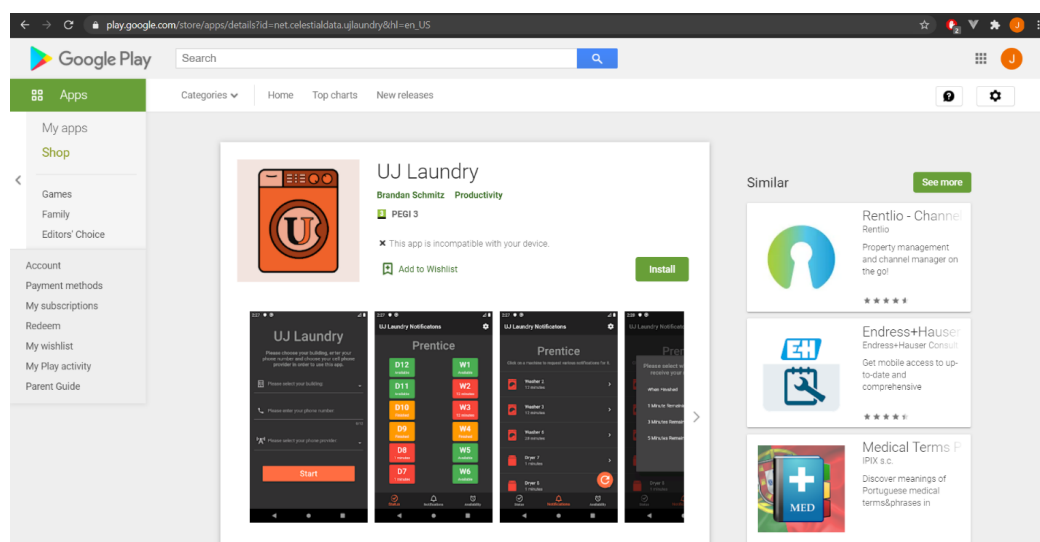
2. Opis projektnog zadatka

2.1 Opis problema i motivacija

Jedan je od glavnih problema života u studentskom domu pranje rublja. Studenti smješteni u domove moraju paziti na raspored pranja više nego na rokove za prijavu ispita, jer što im znači prijavljen ispit ako na njega nemaju u čemu doći. Motivacija za izradu ove web aplikacije leži u nepraktičnom sustavu rezerviranja termina pranja odjeće. Često se zna dogoditi da se student dođe upisati, ali naiđe na zatvorena vrata zbog pauza koje su stalno u drugo vrijeme ili nađe kompletno popunjen kalendar te ne može oprati veš. Želimo da se to iskustvo olakša studentima, ali i zaposlenicima praonice. Na ovaj bi se način mogli puno jednostavnije rezervirati, ali i otkazati termini za pranje koji bi se onda lako popunili te bi se studenti mogli puno bolje organizirati pa čak i upisati za termin koji je netom otkazan. Moderniziralo bi se i plaćanje te penaliziralo često otkazivanje dragocjenih termina.

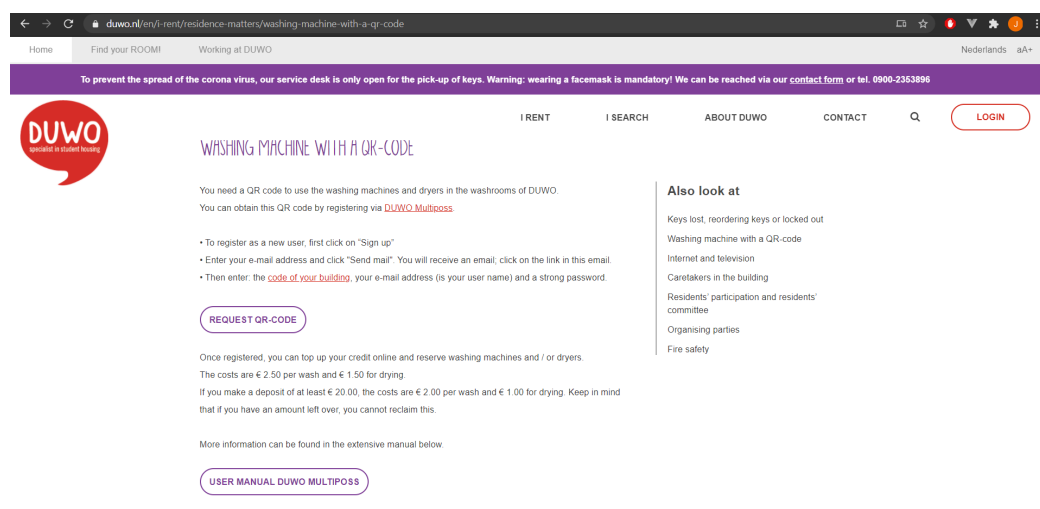
2.2 Postojeća slična rješenja

Istraživanjem tržišta, zaključili smo da postoji nekoliko aplikacija koje se bave navedenim problemima. Jedna je od aplikacija koju smo uspjeli pronaći, a da je slična onome što želimo proizvesti slijedeća: https://play.google.com/store/apps/details?id=net.celestialdata.ujlaundry&hl=en_US (Slika 2.1). Ta je aplikacija razvijena za praonicu na University of Jamestown-u. Dakako, razlika je u tome što je ovo isključivo mobilna aplikacija (postoje verzije za Android i IOS), a mi bismo napravili responzivnu web aplikaciju lako dostupnu svima s bilo kojeg uređaja koji ima pristup internetskoj mreži. Također, imamo i neke dodatne funkcionalnosti koje su navedene u usporedbi sa slijedećom aplikacijom (posudba košare, recenzije radnika, objava slika, oglasi za posao...).



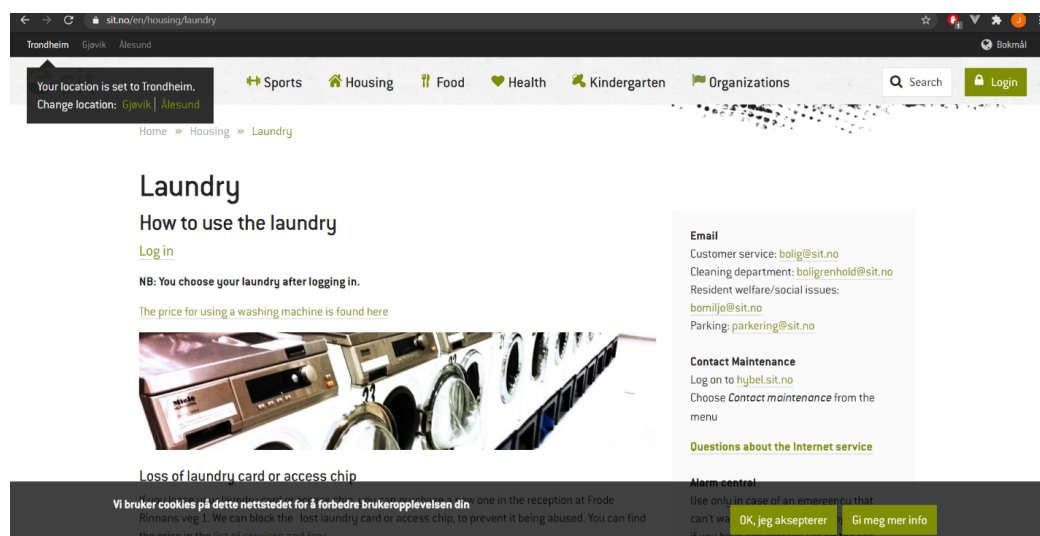
Slika 2.1: Slika aplikacije "UJLAUNDRY"

Također, pronašli smo i slijedeću aplikaciju Nizozemskog porijekla: <https://www.duwo.nl/en/irent/residence-matters/washing-machine-with-a-qr-code> (Slika 2.2). Ta je aplikacija po glavnoj funkcionalnosti poprilično slična našoj (web aplikacija koja omogućuje online rezervaciju termina i plaćanje preko interneta. Razlike su u tome što bi naša aplikacija bila prilagođenija studentskim domovima. Nudila bi funkcionalnost označavanja osobe koja je posudila košaru, recenzije radnika u praoni (što druge aplikacije nemaju jer su većinom vešeraju samoposlužni, duk je u domu zaposlen radnik), mogućnost objavljivanja slika izgubljenih stvari, mogućnost administratora da objavi oglas za posao u praoni, te mogućnost promijene radnog vremena, odnosno vremena pauzi na dnevnoj bazi. Također, ta aplikacija nema mogućnost slanja obavijesti na mail (npr. 1h prije početka termina).



Slika 2.2: Slika aplikacije "Washing machine with a QR code"

Treća pak stranica koju smo pronašli dolazi iz Norveške (Slika 2.3). Sličnih je funkcionalnosti kao i prethodno navedena Nizozemska aplikacija, osim što ima dodatnu funkcionalnost, a to je slanje sms poruka za obavijesti (razlika u odnosu na našu je što bi mi slali na mail). Što se razlika tiče, slične su kao i u gore navedenom primjeru, uz razliku u tome što ova aplikacija, kao i naša, implementira slanje obavijesti korisnicima.



Slika 2.3: Slika aplikacije za rezervaciju termina iz Norveške

Postoji i još nekoliko aplikacija sličnoga karaktera, no naveli smo 3 najpopularnije. Većina ostalih aplikacija dijele slične razlike u usporedbi s našom kao i gore navedene pa bi bilo redundantno navoditi ih.

2.3 Potencijalno zainteresirani korisnici

Za ovaj projekt očekujemo da će najveći interes pokazati studentski domovi s obzirom da je cilj projekta olakšavanje poslovanja već postojećih praonica veša u sklopu domova. No unatoč prvobitnom usmjerenju projekta prema studentskim domovima, smatramo da je ovakvo rješenje također opće primjenjivo i na ostale javne praonice jer primjenom ovakvog rješenja sa mogućnošću online rezervacije termina korisnik ne bi morao riskirati dolazak u punu praonicu te bi mogao izbjeći nepotrebno čekanje. Projekt bi također bilo moguće uvesti u ostale ustanove koje nude mogućnosti pranja veša kao što su npr. hoteli, kampusi itd.

2.4 Mogućnost prilagodbe rješenja

Kao što smo već naveli gore, projekt se vrlo lako može prilagoditi potrebama krajnjeg korisnika pa tako u vidu imamo i potrebne prilagodbe za javne praonice veša kao i mogućnost izvoza proizvoda van hrvatske s obzirom da u planu imamo ponuditi i cijelu web aplikaciju na engleskom jeziku. Mogućnost izbora jezika također bi olakšala upotrebu aplikacije stranim studentima koji borave u hrvatskim studentskim domovima.

2.5 Opseg projektnog zadatka (zahtjevi sustava)

Korisnici ove aplikacije su: osoblje u SC-u zaduženo za rad praonice rublja u ulozi administratora, studenti, osobe zaposlene u praonici i neregistrirani korisnici. Neregistrirani korisnici u mogućnosti su samo napraviti svoj korisnički račun, vidjeti oglase za posao i radno vrijeme te nemaju mogućnosti ni za kakvu drugu radnju. Administratori (osoblje u SC-u zaduženo za rad praonice) jedini mogu postaviti oglas za posao u praonici na koji se mogu prijaviti ljudi koji su registrirani u aplikaciji. Oni također mogu mijenjati radno vrijeme praonice te zabilježiti eventualne promijene cijena pranja i sušenja rublja. Ako je potrebno, administrator može i blokirati pristup aplikaciji bilo kojem korisniku. Sljedeća su kategorija korisnika osobe zaposlene u praonici. Te osobe, poput administratora, mogu mijenjati radno vrijeme praonice te pauze. S obzirom da se termini za pranje i sušenje veša mogu rezervirati svakih 1:30h vremena (npr. ako praonica radi od 8, onda su mogući termini 8, 9:30, 11, 12:30...), pauze mogu trajati maksimalno jedan sat (između 2 termina) te se nikako ne smije dogoditi da pauza bude kad je vrijeme zamjene veša. U aplikaciji osoba koja trenutno radi može označiti da je netko posudio košaru za

veš i ako ne vrati zna se kod koga je. Također, ako osoba ne dođe pokupiti svoj veš, može mu poslati obavijest na mail da je veš gotov. Posljednja su vrsta korisnika ove aplikacije studenti. Oni će se moći prijaviti svojim AAI eduHr računom ako nam CARNET dozvoli korištenje AAI eduHr prijave, dok će u suprotnom moći napraviti korisnički račun te će morati čekati da im osobe zaposlene u praonici potvrde račun (nakon slanja zahtjeva za izradu računa morat će doći osobno u praonicu kako bi im osoba koja trenutno radi potvrdila račun). Glavna je funkcionalnost za ovu vrstu korisnika mogućnost rezervacije i otkazivanja termina za pranje i sušenje veša. Njima je na naslovnoj stranici prikazan kalendar sa slobodnim i popunjenim terminima te lagano mogu odabrati vrijeme i datum za željenu rezervaciju. Prilikom rezervacije moguće je odabrati način plaćanja pranja – putem aplikacije ili na blagajni doma. Također, student može postaviti bilješku kod rezervacije npr. ne mogu doći po veš zbog obaveza na fakultetu, kasnit ću 2 sata. Termin se smije otkazati do 24 sata prije termina, a ako netko otkaže neposredno prije ili nakon termina za otkaz dobiva negativne bodove (ako se učestalo ponavlja korisnik je dužan platiti kaznu). Studentu će sat vremena prije njegovog termina na mail stići obavijest kako ne bi zaboravio doći ili otkazati termin. Studenti mogu i ocijeniti radnika u praoni nakon pranja veša i to samo onoga tko je bio u njihovom terminu jedanput po pranju. U ovoj će aplikaciji postojati i zid s objavama na kojem osoba zaposlena u praoni objavljuje slike izgubljenih stvari. Ako je stvar izgubljena dulje od mjesec dana objava se briše i stvar se donira u dobrotvorne svrhe.

2.6 Nadogradnje projektnog zadatka i planovi za budućnost

Moguća nadogradnja bi bila uključivanje više usluga koje se nude studentima u sklopu doma npr. rezervacija termina u teretani (koja ima ograničen broj mjesta I inače, a osobito sada u vrijeme pandemije), pregledi menija u menzi i radnog vremena, kupnja karti za Kino SC, Teatar & TD i slično. Sljedeći korak u razvoju bi mogao biti razvoj mobilne aplikacije jer bi većina studenata sustavu pristupala na taj način. Nakon razvitka mobilne aplikacije, može se pristupiti izgradnji interaktivnog vodiča za bruce. Vodič bi naše bruce na licu mjesta mogao provesti kroz način funkcioniranja veš mašina, ponuditi im savjet koji program služi za njihovu robu, te bi se time smanjila interakcija između zaposlenika i studenta. Preinakama originalne aplikacije mogli bi ju ponuditi i ostalim samoposlužnim vešerajima po Hrvatskoj ali i svijetu. Morali bi istražiti koje su im funkcionalnosti bitne, a koje bi trebalo ukloniti u odnosu na aplikaciju specifičnu za vešeraje u studentskim domovima.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

dio 1. revizije

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

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

Dionici:

1. Studentski centar Sveučilišta u Zagrebu (naručitelj)
2. Korisnici praonice veša
3. Zaposlenici praonice veša
4. Administrator sustava
5. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani/neprijavljeni korisnik (inicijator) može:
 - (a) napraviti novi korisnički račun za koji su mu potrebni adresa e-pošte, lozinka, ime, prezime i broj kreditne kartice
 - (b) vidjeti oglase za posao (osobe koje ne žive u domu i nisu studenti, odnosno nemaju pravo koristiti praonicu, mogu se zaposliti u praonici, prijave za oglas ne idu putem aplikacije, već še molba i životopis šalju na adresu e-pošte navedenu u oglasu)
 - (c) vidjeti podatke o praonici
 - i. radno vrijeme praonice

ii. cijene pranja i sušenja

2. Registrirani korisnik (inicijator) može:

- (a) marija i dino
- (b) funkcionalnost 2

3. Zaposlenik (inicijator) može:

- (a) ink
- (b) funkcionalnost 2

4. Administrator (inicijator) može:

- (a) Objaviti oglas za posao
- (b) Promijeniti radno vrijeme praonice
- (c) Obaviti promjenu cijene pranja
- (d) Obaviti promjenu cijene sušenja
- (e) Blokirati pristup aplikaciji bilo kojem korisniku

5. Baza podataka (sudionik) može:

- (a) spigl
- (b) funkcionalnost 2

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>

UC1 - Registracija

- **Glavni sudionik:** Korisnik
- **Cilj:** Stvoriti korisnički račun za pristup sustavu
- **Sudionici:** Baza podataka + pitanje (ako radnik treba potvrditi, je li i on sudionik?)
- **Preduvjet:** -
- **Opis osnovnog tijeka:**

1. Korisnik odabire opciju za registraciju
 2. Korisnik unosi potrebne korisničke podatke
 3. Korisnik dobiva obavijest o slanju registracije na potvrdu
 4. Korisnik odlazi osobno u praonicu s potvrdom da stanuje u domu
 5. Radnik u praonici potvrđuje prijavu nakon uvjeravanja da osoba stanuje u domu
 6. Korisnik prima obavijest o uspješnoj registraciji
- **Opis mogućih odstupanja:**
 - 2.a Odabir već zauzete ili nepostojeće adrese e-pošte te neispravno popunjavanje obrasca
 1. Sustav obavještava korisnika o neuspjeloj registraciji i vraća ga na natrag stranicu za registraciju korisnika
 2. Korisnik mijenja potrebne podatke te se ponovno pokušava registrirati ili odustaje od registracije
 - 4.a Korisnik zaboravi odnijeti potvrdu o stanovanju u studentskom domu u praonicu
 1. Sustav obavještava korisnika da odnese potvrdu u praonicu
 2. Nakon određenog vremena pokušaj registracije briše se iz sustava
????
 - 5.a Radnik u praonici nije potvrdio registraciju korisnika
 1. Korisnik ponovno odlazi u praonicu s potvrdom
 - 5.a Radnik u praonici potvrdio je ili odbio zahtjev za registracijom krivog korisnika
 1. Radnik obavještava administratora o tome kojeg je korisnika zabunom potvrdio ili odbio, te administrator promijeni podatke u bazi podataka

UC2 -Pregled podataka o praonici

- **Glavni sudionik: Korisnik**
- **Cilj: Prikaz podataka o praonici**
- **Sudionici: Baza podataka**
- **Preduvjet: -**
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prikaz podataka o praonici
 2. Korisniku se prikazuju podaci o praonici

UC3 -Pregled oglasa za posao

- **Glavni sudionik: Korisnik**
- **Cilj: Prikaz oglasa za posao**
- **Sudionici: Baza podataka**
- **Preduvjet: -**
- **Opis osnovnog tijeka:**
 1. Korisnik odabire opciju za prikaz oglasa za posao
 2. Korisniku se prikazuju oglasi

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 tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam

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	Slika aplikacije "UJLAUNDRY"	5
2.2	Slika aplikacije "Washing machine with a QR code"	6
2.3	Slika aplikacije za rezervaciju termina iz Norveške	6

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

Kontinuirano osvježavanje

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

1. sastanak

- Datum: 2. listopada 2020.
- Prisustvovali: I. Joskić, D. Grgić, B. Spiegl, D. Šmigovec, M. Dragošević, L. Inkret, J. Grgić
- Teme sastanka:
 - smišljanje teme za projekt

2. sastanak

- Datum: 15. listopada 2020.
- Prisustvovali: I. Joskić, D. Grgić, B. Spiegl, D. Šmigovec, M. Dragošević, L. Inkret, J. Grgić
- Teme sastanka:
 - rasprava o funkcionalnostima sustava
 - rasprava o bazi podataka

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Jan grgić	Dunja Šmigovec	Marija Dragošević	Bernard Spiegl	Dino Grgić	Ivan Joskić	Leonard Inkret
Upravljanje projektom	3						
Opis projektnog zadatka	5						
Funkcionalni zahtjevi	2					1	
Opis pojedinih obrazaca	2						
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							
Dnevnik sastajanja	0.2						
Zaključak i budući rad							
Popis literature							
<i>izrada frontenda</i>							

	Jan grgić	Dunja Šmigovec	Marija Dragošević	Bernard Spiegl	Dino Grgić	Ivan Joskić	Leonard Inkret
<i>izrada backenda</i>	2						
<i>spajanje frontenda s back endom</i>							
<i>izrada baze podataka</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.