

Programsko inženjerstvo

Ak. god. 2021./2022.

Radno vrijeme

Dokumentacija, Rev. 0.4

Grupa: *DugonogiProgi*

Voditelj: *Bernard Kazazić*

Datum predaje: *21. listopada 2021.*

Nastavnik: *Miljenko Krhen*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	4
3 Specifikacija programske potpore	8
3.1 Funkcionalni zahtjevi	8
3.1.1 Obrasci uporabe	10
3.1.2 Sekvencijski dijagrami	11
3.2 Ostali zahtjevi	12
4 Arhitektura i dizajn sustava	13
4.1 Baza podataka	13
4.1.1 Opis tablica	13
4.1.2 Dijagram baze podataka	14
4.2 Dijagram razreda	15
4.3 Dijagram stanja	16
4.4 Dijagram aktivnosti	17
4.5 Dijagram komponenti	18
5 Implementacija i korisničko sučelje	19
5.1 Korištene tehnologije i alati	19
5.2 Ispitivanje programskog rješenja	20
5.2.1 Ispitivanje komponenti	20
5.2.2 Ispitivanje sustava	20
5.3 Dijagram razmještaja	21
5.4 Upute za puštanje u pogon	22
6 Zaključak i budući rad	23
Popis literature	24
Indeks slika i dijagrama	25

Dodatak: Prikaz aktivnosti grupe

26

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Raspudić	14.10.2021.
0.1.1	Dopisani dnevnicu sastajanja.	Raspudić	15.10.2021.
0.2	Dodani nefunkcionalni zahtjevi.	Raspudić	16.10.2021.
0.3	Dodani funkcionalni zahtjevi.	Pašalić, Sušac	17.10.2021.
0.4	Dodan opis projekta.	Pavlović, Raspudić	21.10.2021.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za web aplikaciju “Radno vrijeme” koja će omogućiti djelatnicima poduzeća „Mi puno radimo“ praćenje realizacije pojedinih djelatnosti i radnih sati na razini svakog djelatnika, grupe i poduzeća, kao i praćenje zauzeća/raspoloživosti pojedinih djelatnika. Na taj će se način olakšati organizacija istovremenog obavljanja mnogih uslužnih djelatnosti.

Korisnike sustava možemo podijeliti u 4 grupe:

- vlasnika sustava
- voditelja grupa
- zaposlenike
- neregistrirane korisnike

Neregistrirani korisnici mogu samo vidjeti popis i opis djelatnosti koje poduzeće ima u svojem portfoliju. Direktor je omogućeno registriranje korisnika u sustav kreiranjem novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- korisničko ime
- lozinka
- ime
- prezime
- OIB
- adresa e-pošte

Svim djelatnicima kasnije korisničko ime i lozinka služe za prijavu u sustav, a adresa e-pošte kako bi ih se moglo kontaktirati.

Naknadno se djelatnicima, ako već nisu, mogu još i dodijeliti prava voditelja ili vlasnika sustava. Ukoliko radnici idu na intervencije izvan lokacije sjedišta poduzeća, potrebno je za svakoga prikazati mjesto na karti za lokaciju na koju je bio upućen.

Djelatnik poduzeća kroz aplikaciju može vidjeti podatke samo o sebi. Njemu se prikazuju grupe u koje je raspodijeljen te zadatke koji su mu dodijeljeni. Jedan djelatnik može biti raspoređen i na nekoliko radnih zadataka unutar jedne djelatnosti, a i između više njih.

Ukoliko je djelatniku dodijeljen zadatak na nekoj lokaciji različitoj od sjedišta poduzeća, on tu promjenu u lokaciji mora unijeti na karti prije odlaska, ako nije voditelj grupe već unio u sustav.

Voditelji grupa mogu kroz aplikaciju dobiti podatke o sebi, ali i o članovima svoje grupe. Oni razrađuju plan rada svoje grupe te određuju zadatke i pridjeljuju ih zaposlenicima. Za svaki zadatak voditelj određuje očekivani broj potrebnih radnih sati te cijenu radnog sata na razini pojedine djelatnosti i/ili zadatka.

Osim što upravlja grupama, direktor može upravljati zaposlenicima, u smislu da im daje ili oduzima dozvole. Vlasnik može uređivati grupe na način da mijenja članove i voditelje grupa te ima mogućnost brisanja grupa.

On također može definirati nove uslužne djelatnosti kojima se poduzeće bavi te će te promjene biti vidljive u popisu djelatnosti.

Vlasniku sustava se kroz aplikaciju prikazuju zauzetost i realizacija (stvarna i materijalna) za sve djelatnike te odnos planiranih i realiziranih troškova/dobiti. Informacija o zauzetosti djelatnika pomaže mu pri stvaranju grupa kako ne bi preopteretio zaposlenike prevelikom količinom zadataka te na taj način osigurao veću količinu i kvalitetu odrađenog posla.

Ukoliko postoje aktivnosti izvan sjedišta poduzeća, direktor u svakom trenutku mora moći vidjeti na karti gdje se nalazi (ili se nalazio) koji djelatnik. Podaci o adresi intervencije moraju biti prethodno uneseni u sustav.

Na kraju svakog radnog dana svaki pojedini djelatnik (uključujući i voditelje grupa) upisuju broj odrađenih radnih sati taj dan.

Sustav mora omogućiti istovremeni rad svih korisnika sustava i mora omogućiti unos hrvatskih dijakritičkih znakova.

Sustav također mora biti prilagođen i u potpunosti poštivati sva prava radnika, uključujući radno vrijeme (početak i kraj), duljinu radnog vremena, slobodne dana i ostala prava koja radnik ima temeljem općih i posebnih uvjeta definiranih zakonom.

Gledajući dostupne programe koji bi omogućili praćenje produktivnosti zaposlenika možemo vidjeti razne kategorije. Od najpopularnijih, najskupljih, najjednostavnijih pa sve do najjeftinijih. Ujedno je problem što je toliko veliki izbor između svih programa. To otežava voditeljima poduzeća posao odmjerenja jakosti i slabosti svakog programa. Pošto su svi ti programi namjenjeni široj publici, svaki od tih programa će doći sa svojim opcijama koje nisu nužne niti potrebne poduzeću "Mi puno radimo". Ukoliko odaberu neki program, tu dolazi još jedan problem. Učenje korištenja tog programa (engl. *learning curve*). Potrebno je vrijeme i trud prije nego što počnu koristiti taj program optimizirano.

Tu dolazimo mi. Mi dajemo priliku poduzeću "Mi puno radimo" sa krojenim programom prema njihovim potrebama. Ne trebaju plaćati za opcije koje neće koristiti, ne trebaju odustati od svojih zahtjeva kako bi koristili jeftinija i/ili lošija rješenja. Dajemo im priliku da napravimo program koji će oni sami intuitivno znati koristiti jer će sudjelovati u svakom koraku korisničkog sučelja i njegovih funkcionalnosti.

Projekt je uvijek moguće nadograditi. Ovaj projekt je savršen za male, srednje i velike tvrtke. Možemo uz dogovor s klijentima povezati program s njihovim drugim servisima za koje drugi programi nemaju mogućnost, a niti im je u planu jer većina ne bi koristila. Trenutni cilj projekta nije specifičan da ga može samo jedno poduzeće koristiti. Na primjer, trenutna ideja je da zaposlenici sami unesu broj radnih sati za pojedini dan, moguće je nadograditi da se to automatski računa

uz pametne kartice, opcionalno koje bismo mi enkriptirali. Također, ukoliko neke tvrtke imaju već brojače sati za zaposlenike, ali im treba većina naših opcija, uz dogovor možemo proširiti naš program da dobije podatke iz njihovog brojača te samo proširimo program za te servise koje oni traže.

Prednost našeg rješenja je što ne zahtjeva instalaciju na svaki uređaj koji treba pristupiti servisu. Svatko može sa svojeg pametnog mobitela ili računala doći do svojih zadataka, preinaka i novih informacija. Samo trebaju znati svoje korisničko ime i lozinku. Iako ova prednost može doći kao i mana, tu opet dolazimo do naše prethodne snage. Mi možemo, ukoliko je potrebno, ojačati sigurnost programa. Na način da program dozvoli pristup samo ovlaštenim računalima/mobitelima. Na primjer, zabrana pristupa osim dozvoljenim MAC adresama, dozvola pristupa računalima i mobitelima sa instaliranim certifikatom koji mi napišemo i sl. Naša tvrtka kroji proizvod svojim klijentima.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Vlasnik sustava (direktor poduzeća/naručitelj)
2. Djelatnici poduzeća
 - Voditelji grupa
 - Ostali djelatnici
3. Razvojni tim

Aktori i njihovi funkcionalni zahtjevi:

1. Vlasnik sustava (direktor poduzeća) može:
 - (a) definirati uslužne djelatnosti koje će poduzeće raditi
 - (b) dodijeliti djelatnosti voditeljima grupa
 - (c) registrirati novog djelatnika (uključujući i voditelja grupa) i pritom mu dodijeliti ulogu
 - (d) vidjeti zauzetost i realizaciju za sve djelatnike
 - (e) vidjeti trenutno i prošlu poticiju svih djelatnika koji su izašli na intervencije na karti
2. Voditelji grupa mogu:
 - (a) definirati zadatke i dodjeljivati ih djelatnicima
 - (b) zabilježiti procjenu radnih sati potrebnih za zadatak
 - (c) odrediti cijenu sata rada ovisno o djelatnosti ili zadatku
 - (d) pregledati podatke za sebe i svoju grupu
 - (e) upisati broj odrađenih sati za svaki dan

3. Ostali djelatnici mogu:

- (a) vidjeti koji su mu zadaci dodijeljeni i u kojim se grupama nalazi
- (b) pregledati vlastite podatke
- (c) upisati broj odrađenih radnih sati za svaki dan

4. Neregistrirani korisnik može:

- (a) vidjeti popis i opis djelatnosti poduzeća

3.1.1 Obrasci uporabe

Opis obrazaca uporabe

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

UC<broj obrasca> -<ime obrasca>

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

Dijagrami obrazaca uporabe

Prikazati odnos aktora i obrazaca uporabe odgovarajućim UML dijagramom. Nije nužno nacrtati sve na jednom dijagramu. Modelirati po razinama apstrakcije i skupovima srodnih funkcionalnosti.

3.1.2 Sekvencijski dijagrami

dio 1. revizije

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

3.2 Ostali zahtjevi

Sustav treba:

- omogućiti korištenje hrvatskih dijakritičkih znakova pri unosu i prikazu tekstualnog sadržaja
- podržati višekorisnički rad u realnom vremenu
- dati odgovor na traženi upit unutar nekoliko sekundi kada se dohvaćaju podaci iz baze podataka
- imati intuitivno i jednostavno za korištenje korisničko sučelje
- biti implementiran kao web aplikacija koristeći objektno-orijentirane jezike
- neispravno korištenje korisničkog sučelja ne smije narušiti rad sustava
- osigurati sigurnu, brzu i otpornu na vanjske greške vezu s bazom podataka

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

Dodatak: Prikaz aktivnosti grupe

Dnevnik sastajanja

1. sastanak

- Datum: 13. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, A.Pašalić, L.Raspudić, P.Sušac, V.Žunar
- Teme sastanka:
 - 1. sastanak s asistentom i demonstratorom
 - raščišćavanje nejasnoća

2. sastanak

- Datum: 14. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, A.Pašalić, L.Raspudić, V.Žunar
- Teme sastanka:
 - ažuriranje LaTeX dokumentacije
 - dodjela zadataka članovima tima
 - platforme za komunikaciju

3. sastanak

- Datum: 18. listopada 2021.
- Prisustvovali: B.Kazazić, M.Erlić, B.Pavlović, L.Raspudić, P.Sušac, V.Žunar
- Teme sastanka:
 - raspodjela zadataka
 -
 -

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Bernard Kazazić	Marijan Erlić	Barbara Pavlović	Ante Pašalić	Luka Raspudić	Petar Sušac	Veronika Žunar
Upravljanje projektom							
Opis projektnog zadatka							
Funkcionalni zahtjevi							
Opis pojedinih obrazaca							
Dijagram obrazaca							
Sekvencijski dijagrami							
Opis ostalih zahtjeva					60		
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					120		
Zaključak i budući rad							
Popis literature							

	Bernard Kazazić	Marijan Erlić	Barbara Pavlović	Ante Pašalić	Luka Raspudić	Petar Sušac	Veronika Žunar
<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.