Programsko inženjerstvo

Ak. god. 2022./2023.

# CjenikSvega

## Dokumentacija, Rev. *1*

Grupa: *Životinjsko Carstvo*

Voditelj: *Joško Vrsalović*

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

Nastavnik: *Igor Stančin*

# Sadržaj

# 1. Dnevnik promjena dokumentacije

***Kontinuirano osvježavanje***

| Rev. | Opis promjene/dodatka | Autori | Datum |
|---|---|---|---|
| 0.1 | Napravljen predložak.<br>Napisan opis zadatka. | Vlado Perković | 29.10.2022. |

# 2. Opis projektnog zadatka

***dio 1. revizije***

*Cilj projekta je izgraditi web-aplikaciju koja će omogućiti korisnicima praćenje cijena proizvoda u trgovinama, cijene ažuriraju registrirani korisnici uz prikladne nagrade za aktivnost. Zamislimo sljedeću situaciju: Mario, student Fakulteta elektrotehnike i računarstva, želi minimizirati potrošnju novca na svakodnevne namirnice koje nabavlja u obližnjim trgovinama. Kako bi to učinio, Mario prati kataloge koje pronađe na web-stranicama istih trgovina te nakon pronalaska (ili ne pronalaska) određenih proizvoda razmisli isplati li mu se otići u najbližu trgovinu ili je bolje posjetiti više njih. Ovdje se javljaju očiti problemi da nije sve na jednom mjestu te nerijetko ti katalozi uopće nisu pretraživi. Drugi će se korisnici možda odlučiti na fizičke kataloge koji su loša opcija i za okoliš i za pretraživanje. Još jedna mana kataloga jest da ne prikazuju cijene svih proizvoda, nego samo onih sniženih.*

*Kao rješenje na navedene probleme nastaje aplikacija CjenikSvega. Svi proizvodi s cijenama u stvarnom vremenu na jednom mjestu. Aplikacija će biti od najveće koristi osobi koja živi u blizini nekoliko različitih trgovina, na primjer student u domu ili odrasla osoba u gradskoj četvrti.*

*Ta osoba kao **neregistrirani** korisnik može pregledavati i pretraživati sadržaj aplikacije. Odluči li se **registrirati**, dodatno će moći unositi cijene proizvoda i dodavati oznake (engl. tag) proizvodima. Pri registraciji osoba unosi svoje ime, prezime, nadimak i e-mail adresu. [Korisniku je u interesu registrirati se zbog nagrađivanja aktivnih korisnika i zbog mogućnosti personaliziranog popisa proizvoda koji se automatski ažurira]. Jedan od primjera korištenja aplikacije jest da registrirani korisnik opazi razliku u cijenama proizvoda u trgovini i na aplikaciji te uslika taj proizvod s cijenom. Sliku će preko sučelja postaviti na web odakle će administrator odobriti izmjenu cijene.[Dogodi li se da je to već treća ili neka druga proizvoljna izmjena cijene, trgovina će nagraditi korisnika s npr. 10% popusta pri idućoj kupnji]. Valja napomenuti da korisnik šalje sliku jedino kada postoji odstupanje u cijeni. Ako administrator potvrdi promjenu cijene, šalje se*

*obavijest i korisniku i trgovini. Pohranjuju se sve promjene koje su nastale intervencijom korisnika te se to bilježi na stranici trgovine kao krivi unos.*

***Administrator** ima sve mogućnosti kao i registrirani korisnik. Uz to može zabraniti pristup stranici registriranim korisnicima ili trgovinama i napisati komentar o svakoj pojedinoj trgovini koji je onda istaknut na stranicama trgovine. Dakle, pronađu li se neki korisnici koji pokušaju prevariti sistem lažiranim fotografijama proizvoda i cijena, administrator će ih blokirati i zabraniti pristup aplikaciji. Primjer jednog komentara kojeg bi administator mogao ostaviti trgovini jest: "Trgovina xy u posljednjih mjesec dana nije imala nijedan pogrešan unos cijena."*

***Trgovine** se registriraju na aplikaciju kako bi postavljale cijene **vlastitih** proizvoda. Prilikom registracije trgovina mora unijeti popis svih svojih proizvoda i njihove standardne cijene. Trgovina svaki dan ažurira cijene koje odstupaju od standardnih tako što postavi datoteku na računalo koju aplikacija automatski dohvaća. U slučaju da nema promjena, aplikacija pretpostavlja da za taj dan vrijede sve standardne cijene. [Trgovine mogu ponuditi kupone s popustom za aktivne korisnike koji su ispravili određen broj cijena kako bi potaknule aktivnost korisnika, a i svoje radnike u cilju smanjenja pogrešaka].*

*Svaki proizvod može imati do pet oznaka (engl. tag) vrste proizvoda. Primjeri oznaka su pekarski proizvod, mliječni proizvod, hrana, piće i slično. Konkretno za čips oznake bi mogle biti hrana, slane grickalice. Registrirani korisnici mogu predložiti do 5 oznaka po proizvodu. Prikaz oznaka određuje se većinskim glasanjem, odnosno prikazuju se onih pet za koje se najviše ljudi složilo da dobro opisuju navedeni proizvod. Svaki proizvod ima vlastitu stranicu na kojoj su prikazani detalji i slika proizvoda, sve trgovine u kojima se proizvod može kupiti te kako su se cijene mijenjale u zadnjih sedam dana u svakoj pojedinoj trgovini.*

*Rješenje problema pretraživanja jest tražilica u aplikaciji koja proizvode pretražuje po nazivu i/ili po oznakama. Osim proizvoda, mogu se pretraživati i trgovine. [Korisnik može filtrirati rezultate tražilice po cijeni, popularnosti...].*

*Rješenje koje gleda na sličan problem iz druge perspektive koje je dostupno za preuzimanje jest mobilna aplikacija Listonic. Glavna misao vodilja aplikacije jest da se*

*namirnice na popisima generalno ponavljaju te umjesto da se ponovno piše popis, nudi digitalno rješenje. Razlikuje se od našeg rješenja tako što nije povezano s trgovinama koje postavljaju cijene nego korisnici sami zadaju cijene. Pogledamo li malo šire, na primjer aplikacija Basket, mobilna aplikacija nedostupna korisnicima na ovim prostorima koja pruža uslugu dostavljanja proizvoda. Funkcionira na sličan način kao naše rješenje, trgovine postavljaju cijene koje se ažuriraju u aplikaciji te korisnici vrlo lako pronađu proizvode koji ih zanimaju. Razlike su u tome što korisnici ne ispravljaju cijene i u tome što aplikacija nudi dostavu.*

*Rješenje je moguće prilagoditi kao mobilnu aplikaciju, što bi korisnicima olakšalo samo korištenje i postavljanja fotografija. Kako bi učinili cijeli projekt profitabilnim, svakako moguća prilagodba bi bila dodati reklame i premium članstva. Provede li se prilagodba na mobilnu aplikaciju, moguće je naplaćivati samu aplikaciju nakon probnog perioda.*

*Opseg projekta je razviti web-aplikaciju s bazom podataka?*

*Moguće nadogradnje su sustav za planiranje rute, dijeljenje popisa proizvoda s drugim korisnicima, dodavanje proizvoda na popis skeniranjem, prikaz proizvoda gdje se nalazi unutar trgovine. Sustav za planiranje rute bi optimizirao gdje se koji proizvod kupuje tako što uzima u obzir cijenu, udaljenost između trgovina i isplativost dodatnog putovanja. Dijeljenje popisa proizvoda s drugim korisnicima je vrlo korisno u slučaju da osoba svojoj bližnoj osobi sastavi popis i jednostavno podijeli.*

# 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. Dionik 1
2. Dionik 2
3. ...

**Aktori i njihovi funkcionalni zahtjevi:**

1. Aktor 1 (inicijator) može:
   (a) funkcionalnost 1
   (b) funkcionalnost 2
      i. podfunkcionalnost 1
      ii. podfunkcionalnost 2
   (c) funkcionalnost 3
2. Aktor 2 (sudionik) može:
   (a) funkcionalnost 1
   (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 tijek obrasca vratio na osnovni tijek.*

**UC<broj obrasca> -<ime obrasca>**

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

  2.a <opis mogućeg scenarija odstupanja u koraku 2>
    1. <opis rješenja mogućeg scenarija korak 1>
    2. <opis rješenja mogućeg scenarija korak 2>

  2.b <opis mogućeg scenarija odstupanja u koraku 2>

  3.a <opis mogućeg scenarija odstupanja u koraku 3>

**Dijagrami obrazaca uporabe**

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

### 3.1.2 Sekvencijski dijagrami

***dio 1. revizije***

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

## 3.2 Ostali zahtjevi

***dio 1. revizije***

*Nefunkcionalni zahtjevi i zahtjevi domene primjene dopunjuju funkcionalne zahtjeve. Oni opisuju* ***kako se sustav treba ponašati*** *i koja* ***ograničenja*** *treba poštivati (performanse, korisničko iskustvo, pouzdanost, standardi kvalitete, sigurnost...). Primjeri takvih zahtjeva u Vašem projektu mogu biti: podržani jezici korisničkog sučelja, vrijeme odziva, najveći mogući podržani broj korisnika, podržane web/mobilne platforme, razina zaštite (protokoli komunikacije, kriptiranje...)... Svaki takav zahtjev potrebno je navesti u jednoj ili dvije rečenice.*

# 4. Arhitektura i dizajn sustava

***dio 1. revizije***

*Potrebno je opisati stil arhitekture te identificirati: podsustave, preslikavanje na radnu platformu, spremišta podataka, mrežne protokole, globalni upravljački tok i sklopovsko-programske zahtjeve. Po točkama razraditi i popratiti odgovarajućim skicama:*

- *izbor arhitekture temeljem principa oblikovanja pokazanih na predavanjima (objasniti zašto ste baš odabrali takvu arhitekturu)*
- *organizaciju sustava s najviše razine apstrakcije (npr. klijent-poslužitelj, baza podataka, datotečni sustav, grafičko sučelje)*
- *organizaciju aplikacije (npr. slojevi frontend i backend, MVC arhitektura)*

## 4.1 Baza podataka

***dio 1. revizije***

*Potrebno je opisati koju vrstu i implementaciju baze podataka ste odabrali, glavne komponente od kojih se sastoji i slično.*

### 4.1.1 Opis tablica

*Svaku tablicu je potrebno opisati po zadanom predlošku. Lijevo se nalazi točno ime varijable u bazi podataka, u sredini se nalazi tip podataka, a desno se nalazi opis varijable. Svjetlozelenom bojom označite primarni ključ. Svjetlo plavom označite strani ključ*

| **korisnik - ime tablice** | | |
|---|---|---|
| IDKorisnik | INT | Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod |

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

| **korisnik - ime tablice** | | |
|---|---|---|
| korisnickoIme | VARCHAR | |
| email | VARCHAR | |
| ime | VARCHAR | |
| primjer | VARCHAR | |

### 4.1.2 Dijagram baze podataka

*U ovom potpoglavlju potrebno je umetnuti dijagram baze podataka. Primarni i strani ključevi moraju biti označeni, a tablice povezane. Bazu podataka je potrebno normalizirati. Podsjetite se kolegija "Baze podataka".*

## 4.2 Dijagram razreda

*Potrebno je priložiti dijagram razreda s pripadajućim opisom. Zbog preglednosti je moguće dijagram razlomiti na više njih, ali moraju biti grupirani prema sličnim razinama apstrakcije i srodnim funkcionalnostima.*

***dio 1. revizije***

*Prilikom prve predaje projekta, potrebno je priložiti potpuno razrađen dijagram razreda vezan uz* ***generičku funkcionalnost*** *sustava. Ostale funkcionalnosti trebaju biti idejno razrađene u dijagramu sa sljedećim komponentama: nazivi razreda, nazivi metoda i vrste pristupa metodama (npr. javni, zaštićeni), nazivi atributa razreda, veze i odnosi između razreda.*

***dio 2. revizije***

*Prilikom druge predaje projekta dijagram razreda i opisi moraju odgovarati stvarnom stanju implementacije*

## 4.3 Dijagram stanja

***dio 2. revizije***

*Potrebno je priložiti dijagram stanja i opisati ga. Dovoljan je jedan dijagram stanja koji prikazuje* ***značajan dio funkcionalnosti*** *sustava. Na primjer, stanja korisničkog sučelja i tijek korištenja neke ključne funkcionalnosti jesu značajan dio sustava, a registracija i prijava nisu.*

## 4.4 Dijagram aktivnosti

***dio 2. revizije***

*Potrebno je priložiti dijagram aktivnosti s pripadajućim opisom. Dijagram aktivnosti treba prikazivati značajan dio sustava.*

## 4.5 Dijagram komponenti

***dio 2. revizije***

*Potrebno je priložiti dijagram komponenti s pripadajućim opisom. Dijagram komponenti treba prikazivati strukturu cijele aplikacije.*

# 5. Implementacija i korisničko sučelje

## 5.1 Korištene tehnologije i alati

***dio 2. revizije***

*Detaljno navesti sve tehnologije i alate koji su primijenjeni pri izradi dokumentacije i aplikacije. Ukratko ih opisati, te navesti njihovo značenje i mjesto primjene. Za svaki navedeni alat i tehnologiju je potrebno* ***navesti internet poveznicu*** *gdje se mogu preuzeti ili više saznati o njima.*

## 5.2 Ispitivanje programskog rješenja

***dio 2. revizije***

*U ovom poglavlju je potrebno opisati provedbu ispitivanja implementiranih funkcionalnosti na razini komponenti i na razini cijelog sustava s prikazom odabranih ispitnih slučajeva. Studenti trebaju ispitati temeljnu funkcionalnost i rubne uvjete.*

### 5.2.1 Ispitivanje komponenti

*Potrebno je provesti ispitivanje jedinica (engl. unit testing) nad razredima koji implementiraju temeljne funkcionalnosti. Razraditi* ***minimalno 6 ispitnih slučajeva*** *u kojima će se ispitati redovni slučajevi, rubni uvjeti te izazivanje pogreške (engl. exception throwing). Poželjno je stvoriti i ispitni slučaj koji koristi funkcionalnosti koje nisu implementirane. Potrebno je priložiti izvorni kôd svih ispitnih slučajeva te prikaz rezultata izvođenja ispita u razvojnom okruženju (prolaz/pad ispita).*

### 5.2.2 Ispitivanje sustava

*Potrebno je provesti i opisati ispitivanje sustava koristeći radni okvir Selenium*[1]*. 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

***Kontinuirano osvježavanje***

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

1. sastanak
    - Datum: u ovom formatu: 29. listopada 2022.
    - Prisustvovali: I.Prezime, I.Prezime
    - Teme sastanka:
        - opis prve teme
        - opis druge teme
2. sastanak
    - Datum: u ovom formatu: 29. listopada 2022.
    - Prisustvovali: I.Prezime, I.Prezime
    - Teme sastanka:
        - opis prve teme
        - opis druge teme

# Tablica aktivnosti

***Kontinuirano osvježavanje***

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

| | **Ime Prezime voditelja** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** |
|---|---|---|---|---|---|---|---|
| Upravljanje projektom | | | | | | | |
| Opis projektnog zadatka | | | | | | | |
| Funkcionalni zahtjevi | | | | | | | |
| Opis pojedinih obrazaca | | | | | | | |
| Dijagram obrazaca | | | | | | | |
| Sekvencijski dijagrami | | | | | | | |
| Opis ostalih zahtjeva | | | | | | | |
| Arhitektura i dizajn sustava | | | | | | | |
| Baza podataka | | | | | | | |
| Dijagram razreda | | | | | | | |
| Dijagram stanja | | | | | | | |
| Dijagram aktivnosti | | | | | | | |
| Dijagram komponenti | | | | | | | |
| Korištene tehnologije i alati | | | | | | | |
| Ispitivanje programskog rješenja | | | | | | | |

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

| | **Ime Prezime voditelja** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** | **Ime Prezime** |
|---|---|---|---|---|---|---|---|
| Dijagram razmještaja | | | | | | | |
| Upute za puštanje u pogon | | | | | | | |
| Dnevnik sastajanja | | | | | | | |
| Zaključak i budući rad | | | | | | | |
| Popis literature | | | | | | | |
| | | | | | | | |
| *Dodatne stavke kako ste podijelili izradu aplikacije* | | | | | | | |
| *npr. izrada početne stranice* | | | | | | | |
| *izrada baze podataka* | | | | | | | |
| *spajanje s bazom podataka* | | | | | | | |
| *back end* | | | | | | | |
| | | | | | | | |

# 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.*