Programsko inženjerstvo

Ak. god. 2023./2024.

Digitalizacija

Dokumentacija, Rev. *1*

Grupa: *Jura Hostić i Film*

Voditelj: *Tvrtko Puškarić*

Datum predaje: *17. 11. 2023.*

Nastavnik: *Igor Stančin*

Sadržaj

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Stvoren prazan predložak.	*	05.11.2023.
0.2	Podjela posla, započeta poglavlja 2, 3 i 4.	*	08.11.2023.
0.3	Dodano poglavlje 3.2, sekvencijski dijagrami	Katarina Pešić	16.11.2023.
0.4	Dodane osnovne informacije, zapisnik sastanka	Luka Bradarić Lisić, Tvrtko Puškarić	17.11.2023.
0.5	Dodano poglavlje 2	Tvrtko Puškarić	17.11.2023.
0.6	Dodano poglavlje 4	Andrea Milanović, Jura Hostić, Karlo Grgičin	17.11.2023.

2. Opis projektnog zadatka

Cilj ovog projekta je pružiti specifičnom klijentu, u ovom slučaju računovodstvenom uredu, prilagođeno i primjenjivo rješenje za njihove potrebe. Rješenje se ostvaruje u obliku mobilne aplikacije koja mora biti prilagođena i konfigurirana prema njihovim specifičnim potrebama i strukturi posla što osigurava da klijent dobije optimalno rješenje za digitalizaciju i održavanje postojeće arhive dokumenata, usklađeno s njihovom poslovnom strukturom. To podrazumijeva sagledavanje potreba svih zaposlenika, proučavanje obrazaca upotrebe i pronalaska najboljeg načina za tehnološko ostvarenje. Potrebno je dostaviti gotovo, primjenjivo rješenje koje istovremeno zadovoljava zadane uvjete i obrasce upotrebe te primijeniti valjana inženjerska načela kako bi se osigurala kvaliteta proizvoda, kao i mogućnost lake održivosti te dugovremene upotrebe.

2.1 Skup ciljanih korisnika i upotrebe

Skup ciljanih korisnika čine svi zaposlenici firme, uz dodatnog vanjskog administratora koji početno postavlja sustav. Rješenje mora biti prilagođeno zahtjevima svih vrsta korisnika, odnosno prilagođeno potrebama za rad svih zaposlenika unutar ureda. U daljnjem tekstu će zaposlenici biti vezani s istoimenim ulogama. Važno je napomenuti kako je za sve namijene potrebno ostvariti funkcionalno i jednostavno sučelje.

Kako je računovodstveni ured zatvoreni sustav, nepovezan s vanjskim čimbenicima, generalno nije potrebno imati nadstrukturu koja će upravljati sustavom već ured može upravljati samim sobom, za što je zadužen direktor. Vanjski administrator postoji jedino u vidu početnog postavljanja sustava gdje on stvara direktora i po potrebi druge korisnike. Vanjski administrator također može uređivati korisnike po potrebi ako dođe do nepredviđenih grešaka u radu, no zamišljen je samo kao pomoćni alat koji odstranjuje potrebu za direktnim pristupom bazi podataka. Prvenstveno bi administrator trebao biti vanjski akter, na primjer član razvojnog tima, no ulogu je umjesto toga također moguće dati zaposleniku ureda ovisno o klijentovim željama.

Sukladno zahtjevima zaposlenici ureda se dijele na općenite zaposlenike, revizore, računovođe i direktore. Svi zaposlenici trebaju imati mogućnost digitalizacije dokumenata, kao i pregleda vlastitih, prethodno predanih. Također, valjalo bi omogućiti pregled statusa pojedinog predanog dokumenta u tijeku procesa potvrđivanja. Nadalje, općeniti zaposlenik je onaj koji nema dodatnih ovlaštenja osim digitalizacije pojedinog dokumenta.

Nakon predaje dokumenta potrebno ga je revidirati, čime se bave revizori. Revizorima se dokumenti trebaju dodijeliti ovisno o trenutnom statusu i zaposlenosti, nakon čega je odabranog revizora potrebno obavijestiti. Revizor pojedini dokument može odbiti ili potvrditi, a potvrđene prosljeđuje dalje računovođi. Sustav bi trebao omogućiti lako prosljeđivanje kako bi se proces ubrzao, stoga se mogu ponuditi opcije za automatsko ili ručno prosljeđivanje. U slučaju automatskog prosljeđivanja bira se valjani računovođa s trenutno najmanje posla. Dodatno, valjalo bi revizoru omogućiti pregled prethodno revidiranih dokumenata i njihovog daljnjeg statusa.

Računovođe se dijele ovisno o vrstama dokumenata koje mogu arhivirati. Prije arhiviranja određenog dokumenta računovođa također ima mogućnost prosljeđivanja

dokumenta direktoru na potpis, na koji se treba čekati u tom slučaju. Računovođu valja obavijestiti o dokumentima spremnim za arhiviranje, bili oni novi ili novopotpisani. Kao i prethodnim zaposlenicima valjalo bi računovođi omogućiti pregled dokumenata na kojima je radio, konkretnije prethodno arhiviranih.

Direktor je glavni upravitelj ureda i kao takav mora imati pristup stvaranju i brisanju korisnika, kao i uređivanju podataka i dopuštenja istih. Stoga mu je potrebno omogućiti pregled svih korisnika, njihovih statistika i dokumenata na kojima su radili. Sam direktor također sudjeluje u tijeku digitalizacije dokumenata ako se od njega traži potpis, o čemu ga se mora obavijestiti. Konačno, direktoru se mora omogućiti objavljivanje dokumenata na društvenim mrežama.

Rješenje osim što mora zadovoljavati početne zahtjeve, također mora biti lako prilagodljivo novim zahtjevima, zaposlenicima ili ulogama, zbog čega ga je važno pravilno koncipirati.

2.2 Osnovni zahtjevi

Osnovni zahtjevi koje planirana aplikacija mora ostvariti odnose se na procese digitalizacije i upravljanja prenesenim dokumentima te upravljanja zaposlenicima u sklopu aplikacije. Svaki od navedenih procesa ima svoje specifične zahtjeve, a oni koji nisu detaljnije obrađeni pri pregledu skupa korisnika opisani su u nastavku.

Proces digitalizacije sastoji se od unosa dokumenata u aplikaciju. Mora biti omogućen unos do 50 dokumenata odjednom. Za svaki dokument mora se provesti OCR (optical character recognition), što čini osnovnu funkcionalnost samog procesa. Prije same digitalizacije za svaki predani dokument valja provjeriti zadovoljava li sve predodređene uvjete, što se odnosi na kut slikanja i dobiveni oblik sadržaja. Uvjete valja provjeriti kako bi se ispravno mogao provesti OCR proces. Ako je digitalizacija pojedinog dokumenta uspješna korisniku se prikazuje sažetak dokumenta, nakon čega mora označiti dokument pravilno ili nepravilno skeniranim, ovisno o čemu se prosljeđuje dalje u sklopu tijeka upravljanja prenesenim dokumentima.

Važno je napomenuti kako se skup dokumenata koje je moguće skenirati sastoji od tri tipa dokumenata, odnosno računa, ponuda i internih dokumenta. Svaki tip zadovoljava određeni format po čemu se mogu razvrstati u procesu upravljanja. Računi će u svom tekstu nakon OCR-a imati oznaku računa koja je veliko slovo R te šest znamenaka, oznaka ponude imat će veliko slovo P i devet znamenaka, a oznaka internog dokumenta „INT“ i četiri znamenke. Računi osim oznake sadrže ime klijenta, artikle s cijenama i ukupnu cijenu. Ponude su kao računi, ali ne sadrže ime klijenta. Interni dokumenti sadrže samo nestrukturirani tekst. Očekuje se kako će u slučaju ispravnog predanog dokumenta, valjanog formata, aplikacija biti u stanju prepoznati tip i pravilno ga razvrstati.

Očekuje se kako će pravilno skenirani dokumenti biti spremljeni u bazi podataka te biti dostupni za pregled svim ovlaštenim zaposlenicima, ovisno o stadiju u kojem se nalaze.

2.3 Korisnost rješenja

Planirano rješenje korisno je u okviru procesa digitalizacije i održavanja postojećih arhiva dokumenata zbog mogućnosti za optimizaciju rada, što se odnosi na raspodjelu posla i točno definiranim koracima koji se provode. Rješenje će omogućiti brzu i efikasnu digitalizaciju postojećih papirnatih dokumenata s pomoću algoritama za prepoznavanje teksta i automatsku klasifikaciju dokumenata, što znatno ubrzava proces pretvaranja fizičkih dokumenata u digitalni format. To čini arhiviranje i pretraživanje dokumenata mnogo jednostavnijim, smanjuje potrebu za skladištenjem fizičkih dokumenata i minimizira rizik od gubitka ili oštećenja dokumenata. Aplikacija također omogućuje digitalno-analogno upravljanje dokumentima. Zaposlenici mogu lako pristupiti digitalnim kopijama dokumenata putem sučelja, pregledavati ih, uređivati i označavati kako bi zadovoljili svoje specifične potrebe. Ovo omogućuje rad s dokumentima na način koji je prilagođen njihovim zahtjevima, bez potrebe za povratkom fizičkih dokumenata iz arhive. Najbitnije je što se olakšava raspodjela posla za održavanje arhiva. Sustav omogućuje postavljanje različitih pristupnih razina i ovlaštenja za korisnike, omogućavajući preciznu kontrolu nad tim tko može pristupiti, pregledavati ili uređivati određene dokumente. To olakšava timski rad, omogućava bolju suradnju i smanjuje rizik od neovlaštenog pristupa. Na kraju rješenje pomaže klijentu da iskoristi prednosti digitalizacije, učinkovitije upravlja svojim arhivom dokumenata te ostvaruju uštede u vremenu, prostoru i resursima.

2.4 Slična postojeća rješenja

Uz zahtjeve i želje klijenta nužno je proučiti i druga dostupna rješenja koja se bave sličnom problematikom kako bi se upotpunila lista zahtjeva i poboljšao plan rješenja. Načini na koje postojeća rješenja pristupaju određenim odrednicama problema mogu pomoći pri razradi vlastitog rješenja, a pogotovo ako se klijent već susretao s nekim od alata ili već koristi jedan od istih. U nastavku su navedena najpoznatija i najraširenija postojeća rješenja koja se mogu pronaći u upotrebi, kao i kratki opis svakog.

Adobe Acrobat je poznata platforma za uređivanje i upravljanje dokumentima. S obzirom na proširenja, omogućava napredne mogućnosti za obradu dokumenata, uključujući OCR za pretvaranje skeniranih dokumenata u pretražive tekstualne datoteke. Dodatno, proširenja kao što su Adobe Sign omogućuju digitalno potpisivanje dokumenata, pojednostavljujući poslovne procese.

ABBYY FineReader i FlexiCapture su rješenja specijalizirana za OCR i automatizaciju procesa. FineReader se koristi za OCR, konverziju i prepoznavanje teksta na visokoj razini preciznosti, dok FlexiCapture omogućava automatizaciju prikupljanja podataka iz različitih izvora i vrsta dokumenata, uključujući obrasce i fakturiranje.

Amazon Textract je Amazonova usluga za automatsko prepoznavanje teksta i strukturu dokumenata, a može se integrirati s drugim AWS uslugama kao što su S3, Lambda i Rekognition. Textract kombinira OCR s naprednim algoritmima strojnog učenja kako bi izvlačio podatke iz dokumenata, olakšavajući njihovu analizu i upotrebu.

Google Workspace (ranije poznat kao G Suite) omogućava korisnicima pohranu, zajedničko uređivanje i dijeljenje dokumenata u oblaku. Dodatno, Google Document AI koristi strojno učenje za analizu i ekstrakciju informacija iz dokumenata, čime se olakšava pretraživanje i organizacija dokumenata unutar platforme.

Microsoft SharePoint je platforma za upravljanje sadržajem i suradnju koja omogućava organizacijama da pohrane, dijele i upravljaju svojim dokumentima. Ima ugrađene mogućnosti za OCR, omogućujući pretraživanje i indeksiranje sadržaja dokumenata, čime olakšava njihovo upravljanje i održavanje unutar SharePoint okoline.

Glavno što je zajedničko navedenim rješenjima jest mogućnost baratanja dokumentima, tijek digitalizacije dokumenata kao i mogućnosti za OCR. Prednost vlas-

titog rješenja nalazi se u većoj prilagođenosti specifičnim zahtjevima, što dodatno povlači lakšu integraciju u postojeći tijek rada čime nestaje potreba za proučavanjem mnoštva dostupnih alata i prilagodbe istih klijentovim potrebama.

2.5 Mogućnosti dodatnih proširenja

Mogućnost slikanja dokumenata direktno putem mobilne aplikacije može biti vrlo korisna značajka koja se može ostvariti zbog izabrane platforme. Predstavljala bi velika prednost nad klasičnim, računalnim programima pružajući korisnicima brz i praktičan način za unos i obradu dokumenata. Uporaba kamere mobilnog uređaja omogućuje korisnicima lak unos slike dokumenta i obrade putem OCR-a. Ovo bi bilo posebno korisno zaposlenicima koji su često u pokretu i žele brzo dokumentirati račune, ponude ili interne dokumente.

Povezano s time, razvoj paralelne web inačice aplikacije proširio bi korisničko iskustvo omogućujući pregled dokumenata na računalima. Web inačica omogućila bi korisnicima, posebice računovođama i direktorima, pristup dokumentima putem web preglednika. Ovo je korisno za dublje analize, pregledavanje većih količina dokumenata ili jednostavno pristupanje informacijama na radnom računalu.

Teoretska web inačica zadržala bi sve funkcionalnosti mobilne aplikacije, uključujući pregled sažetaka dokumenata, praćenje povijesti, označavanje ispravno ili krivo skeniranih dokumenata te objavljivanje određenih dokumenata na društvenim mrežama. Ovo proširenje platforme omogućilo bi korisnicima fleksibilnost u radu zbog olakšanog pristupa informacijama iz različitih okruženja, bilo putem mobilnog uređaja ili računala.

2.6 Daljnja prilagodba, razvoj i održavanje

Velika prednost upotrebe cross-platform tehnologije za razvoj mobilne aplikacije je što olakšava buduću prilagodbu i održavanje. Razvojni okvir Flutter omogućuje ponovnu upotrebu koda između operativnih sustava za mobilne uređaje što pojednostavljuje popravak grešaka, poboljšanje postojećih funkcionalnosti, kao i implementaciju novih, čime se smanjuje vrijeme i trošak održavanja sustava. Razvoj inačice za IOS uređaje u budućnosti bi mogao imati veliki prioritet kako bi svi zaposlenici mogli koristiti aplikaciju na vlastitim uređajima.

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>

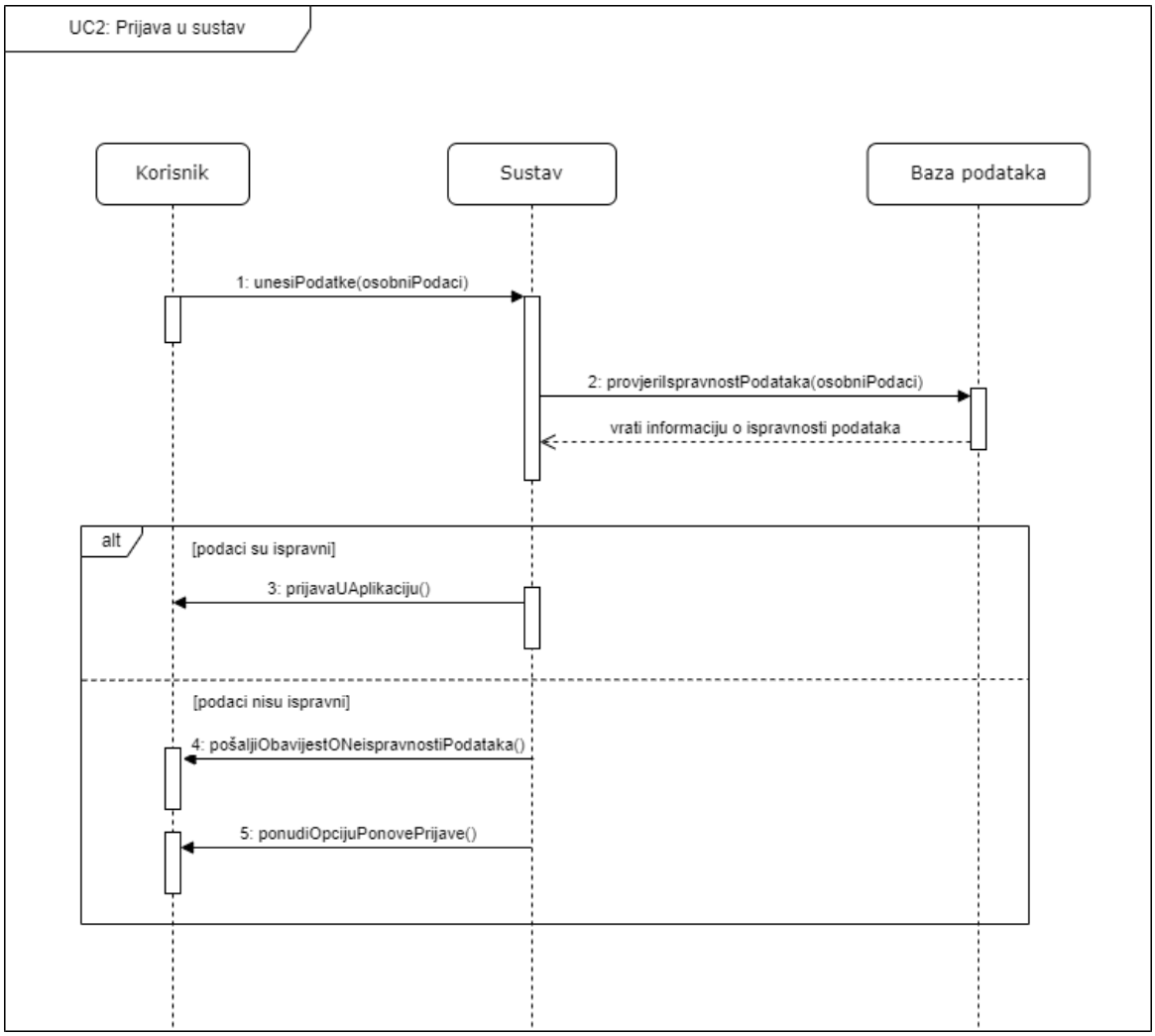
Dijagrami obrazaca uporabe

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

Obrazac uporabe UC2 - Prijava u sustav

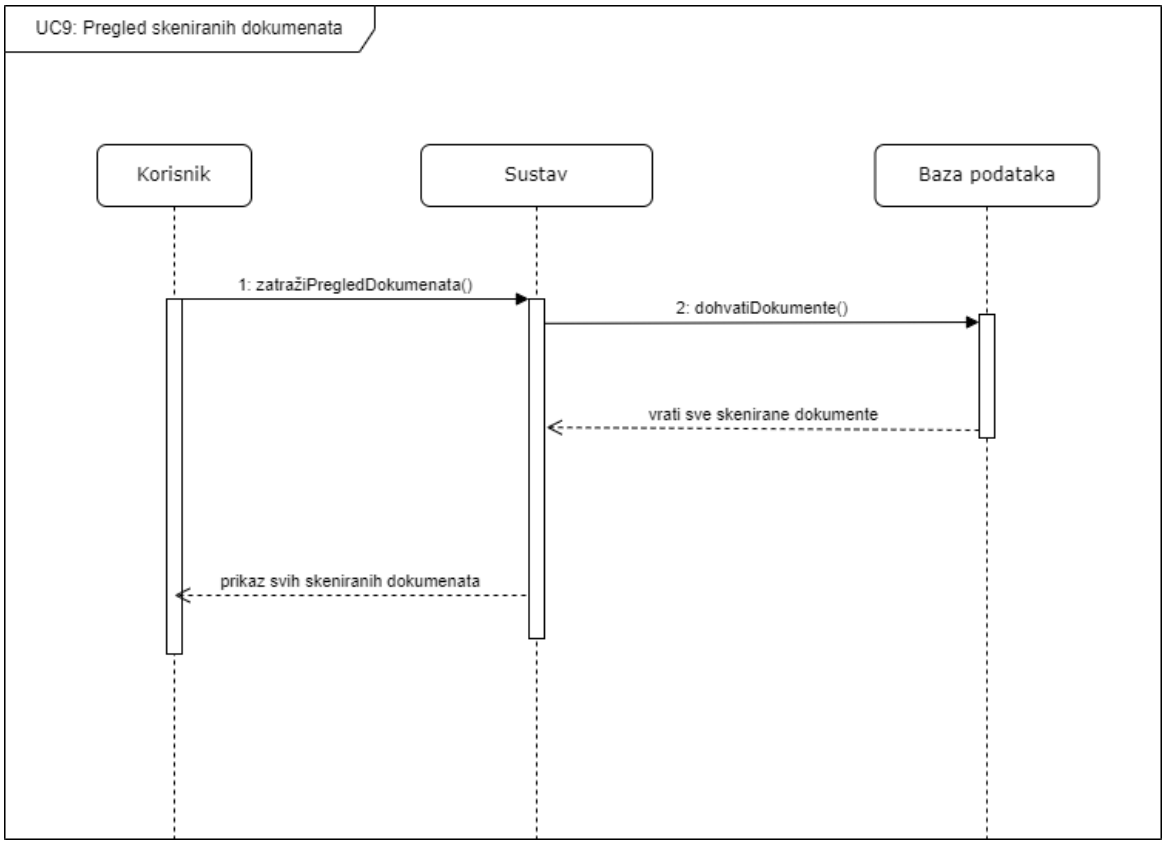
Korisnik unosi svoje osobne podatke. Sustav provjerava u bazi podataka jesu li uneseni podaci ispravni. Ako jesu, korisnik se prijavi u sustav. No ako uneseni podaci ne odgovaraju niti jednom registriranom korisniku u bazi, sustav korisniku šalje obavijest da su osobni podaci neispravni i daje mu priliku za ponovnu prijavu.



Slika 3.1: Slika broj: Sekvencijski dijagram za UC2

Obrazac uporabe UC9 - Pregled skeniranih dokumenata

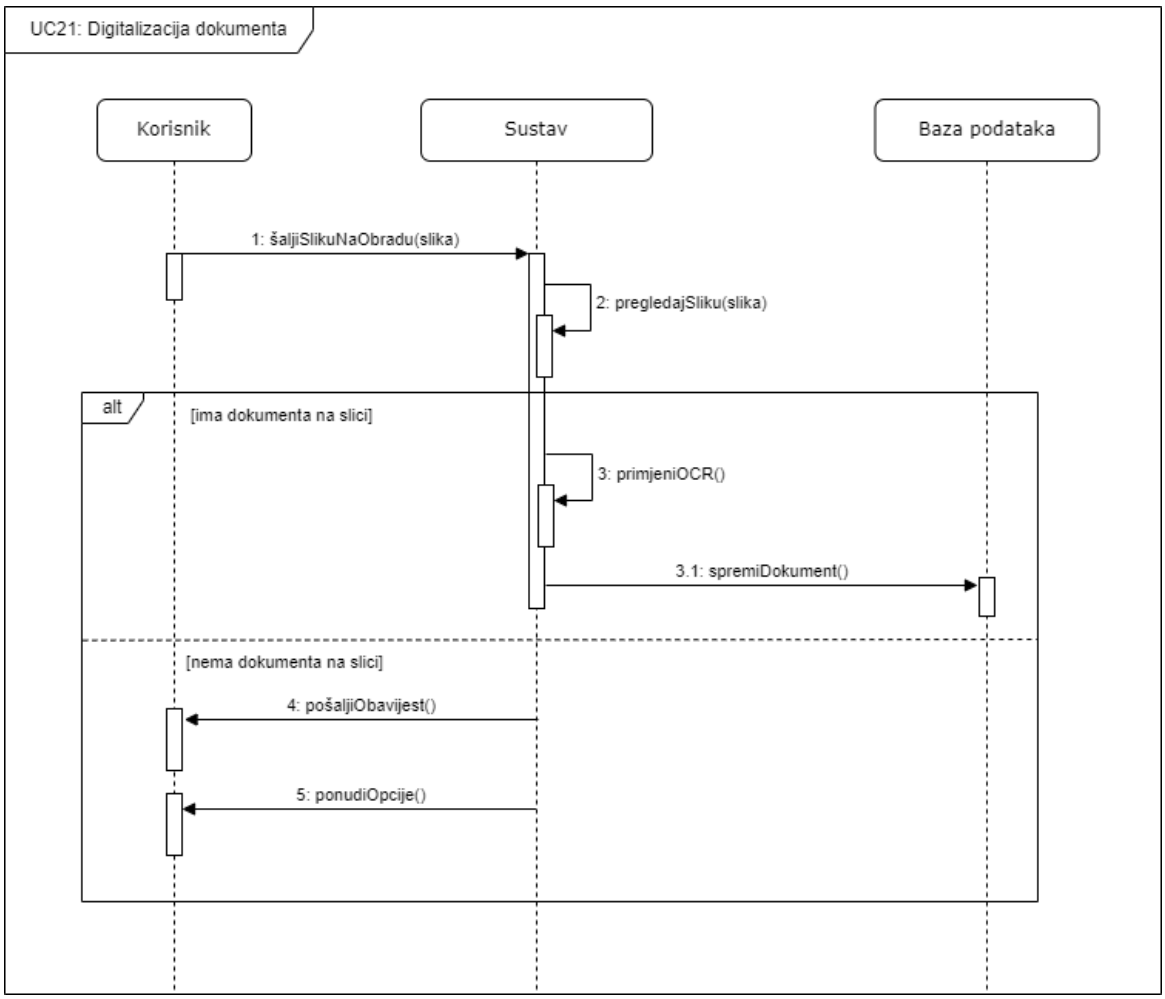
Korisnik odabire pregled popisa svih dokumenata koje je skenirao. Sustav pretraži u bazi njegove skenirane dokumente i vraća ih korisniku na uvid.



Slika 3.2: Slika broj: Sekvencijski dijagram za UC9

Obrazac uporabe UC21 - Digitalizacija dokumenta

Korisnik šalje sliku u sustav koji zatim tu sliiku učitava i pregledava je li na slici dokument. Ako je, onda na njega primjenjuje OCR te novonastali dokument sprema u bazu podataka. Ako na slici nije očitan dokument, sustav korisniku šalje obavijst da učitana slika ne sadrži dokument te mu daje opciju da priloži novu sliku.



Slika 3.3: Slika broj: Sekvencijski dijagram za UC21

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

Arhitektura rješenja se sastoji od 3 sustava: Mobilne aplikacije(Frontend), Web aplikacije(Backend) i Baze podataka.

Mobilna aplikacija je softver namjenjen pokretanju na mobilnim uređajima poput mobitela i tableta. Ona pomoću API-a **(Application programming interface)** komunicira s web aplikacijom koja se brine za logiku. API je je skup pravila, protokola i definicija koje omogućuje komunikaciju između različitih programa ili aplikacija. Sama komunikacija se odvija pomoću HTTPS-a **(Hypertext Transfer Protocol Secure)**, što je sigurnija inačica HTTP-a **(Hypertext Transfer Protocol)** koji omogućuje kkomunikaciju na internetu. Kada web aplikacija primi zahtjev od mobilne aplikacije, ona ovisno o zahtjevu, pristupa bazi podataka kako bi dohvatila, promjenila ili dodala informacije. Ovisno o zahtjevu, web aplikacija vraća odgovor na orginalni zahtjev mobilne aplikacije i mobilna aplikacija prikazuje dobivene informacije kroz svoje sučelje.

Mobilna aplikacija je razvijena pomoću programskog jezika *Dart* te radnog okvira *Flutter* unutar razvojnog okruženja *Android Studio*. Web aplikacija je razvijena pomoću programskog jezika *Python* te radnog okvira *FastAPI* unutar razvojnog okruženja *PyCharm* te je postavljena na *render.com*. Sve podatke pohranjujemo unutar *PostgreSQL* baze podataka koja je na *Azure*-u.

Aplikacija se temelji na MVC konceptu u modernijoj izvedbi koja je na web aplikaciji bliža tipičnoj arhitekturi fastapi aplikacija:

- **Model** - Model predstavlja strukture podatakae. On pristupa bazi podataka, te upravlja podacima. Unutar našeg rješenja, postoji poseban sloj u web aplikaciji koji komunicira s bazom podataka (CRUD klase).

- **View** - Pogled predstavlja korisničko sučelje aplikacije te prikazuje sve informacije korisniku. Pomoću njega, korisnik može imati interakciju s ostatkom aplikacije. U našoj arhitekturi, to bi bila mobilna aplikacija koja prikazuje sve informacije i sama po sebi ima minimalnu logiku te primarna svrha je prikaz informacija korisniku.

- **Controller** - Kontroler postoji između Viewa i Modela te služi da bi primao

zahtjeve od Viewa, slao odgovore natrag Viewu, vrši poslovnu logiku te prosljeduje zahtjeve pripadnom djelu modela. Unutar naše izvedbe, Controller je dio Web aplikacije koji prima zahtjeve od mobilne aplikacije, vrši poslovnu logiku nad njima te prosljeđuje potrebne podatke iz zahtjeva modelu te od modela dobiva natrag podatke koje u prikladnom obliku prosljeđuje mobilnoj aplikaciji. U našoj izvedbi je izveden pomoću route funckija i Service klasa.

4.1 Baza podataka

Koristimo relacijsku bazu podataka s implementacijom pomoću PostgreSQL-a. Za opis naše baze podataka koristimo DBML. DBML (Database Markup Language) je jezik označavanja koji omogućuje deklarativno opisivanje relacijskih baza podataka. Time možemo jednostavno definirati strukture baze podataka pomoću tekstualnog zapisa. Zadaća baze podataka je da olakša i ubrza pristup podacima za daljnju obradu ili pronalaženje arhiviranih podataka. Glavne komponente naše baze podataka su:

- **documents**
- **users**
- **signatures**
- **audits**
- **archive**
- **image**
- **user_role**
- **roles**

4.1.1 Opis tablica

Documents Sadrži informacije o dokumentima, uključujući vrstu dokumenta, sliku dokumenta, sažetak, status, vlasnika, oznaku o ispravnom skeniranju i vremensku oznaku skeniranja. Ovaj entitet u vezi je *One-to-Many* s entitetom **users** preko atributa *ownerID*, u vezi je *One-to-Many* s entitetom **signatures** preko atributa *documentID*, u vezi je *One-to-Many* s entitetom **audits** preko atributa *documentID*, u vezi je *One-to-Many* s entitetom **archive** preko atributa *documentID*, u vezi je *One-to-One* s entitetom **image** preko atributa *imageID*, u vezi je *Many-to-One* s entitetom **users** preko atributa *ownerID*.

documents		
documentID	INT	identifikacijski broj dokumenta
documentType	ENUM	tip skeniranog dokumenta: račun, ponuda ili interni
imageID	INT	identifikacijski broj slike dokumenta
summary	TEXT	tekst koji sadrži skenirani dokument
status	ENUM	trenutni status dokumenta: odobren, odbijen, potpisan, reviziran, potpisan i arhiviran ili arhiviran
correctlyScanned	BOOL	je li dokument označen kao ispravno skeniran ili nije
ownerID	INT	identifikacijski broj korisnika koji je skenirao dokument
scanTimestamp	DATETIME	oznaka datuma i vremena skeniranja dokumenta

Users Sprema informacije o korisnicima, uključujući korisničko ime i lozinku, ime i prezime korisnika te email. Ovaj entitet u vezi je *One-to-Many* s entitetom **documents** preko atributa *userID*, u vezi je *One-to-Many* s entitetom **signatures** preko atributa *userID*, u vezi je *One-to-Many* s entitetom **audits** preko atributa *userID*, u vezi je *One-to-Many* s entitetom **archive** preko atributa *userID*, u vezi je *One-to-Many* s entitetom **user_role** preko atributa *userID*.

users		
userID	INT	identifikacijski broj korisnika
email	VARCHAR	email korisnika
firstName	VARCHAR	ime korisnika
lastName	VARCHAR	prezime korisnika
username	VARCHAR	korisničko ime za prijavu korisnika
password	VARCHAR	šifa za prijavu korisnika

Signatures Služi za praćenje potpisa na dokumentima, s informacijama o ko-

risniku koji je potpisao, statusu potpisa i vremenskoj oznaci potpisa. Ovaj entitet u vezi je *Many-to-One* s entitetom **users** preko atributa *signedBy*, u vezi je *One-to-One* s entitetom **documents** preko atributa *documentID*.

signatures		
signatureID	INT	identifikacijski broj potpisanog dokumenta
documentID	INT	identifikacijski broj dokumenta
signedBy	INT	identifikacijski broj korisnika koji je potpisao ili treba potpisati dokument
status	ENUM	status dokumenta za potpisivanje: potpisan ili čeka na potpisivanje
signedAt	DATETIME	oznaka datuma i vremena potpisivanja dokumenta

Audits Sadrži podatke o revizijama dokumenata, uključujući korisnika koji je obavio reviziju, status revizije i vremensku oznaku. Ovaj entitet u vezi je *Many-to-One* s entitetom **users** preko atributa *auditedBy*, u vezi je *One-to-One* s entitetom **documents** preko atributa *documentID*.

audits		
auditID	INT	identifikacijski broj reviziranog dokumenta
documentID	INT	identifikacijski broj dokumenta
auditedBy	INT	identifikacijski broj korisnika koji je revizirao ili treba revizirati dokument
status	ENUM	status dokumenta za reviziju: reviziran ili čeka na reviziranje
auditedAt	DATETIME	oznaka datuma i vremena reviziranja dokumenta

Archive Koristi se za praćenje arhiviranja dokumenata, s informacijama o korisniku koji je arhivirao dokument, statusu arhiviranja, vremenskoj oznaci i broju arhive. Ovaj entitet u vezi je *One-to-One* s entitetom **documents** preko atributa *documentID*, u vezi je *Many-to-One* s entitetom **users** preko atributa *archiveBy*.

archive		
archiveNumber	INT	identifikacijski broj arhiviranog dokumenta
documentID	INT	identifikacijski broj dokumenta
archiveBy	INT	identifikacijski broj korisnika koji je arhivirao ili treba arhivirati dokument
status	ENUM	status dokumenta za arhiviranje: arhiviran ili čeka na arhiviranje
archivedAt	DATETIME	oznaka datuma i vremena arhiviranja dokumenta

Image Sprema slike dokumenata. Ovaj entitet u vezi je *One-to-One* s entitetom **documents** preko atributa *imageID*.

image		
imageID	INT	identifikacijski broj slike dokumenta
image	BLOB	spremljena slika dokumenta

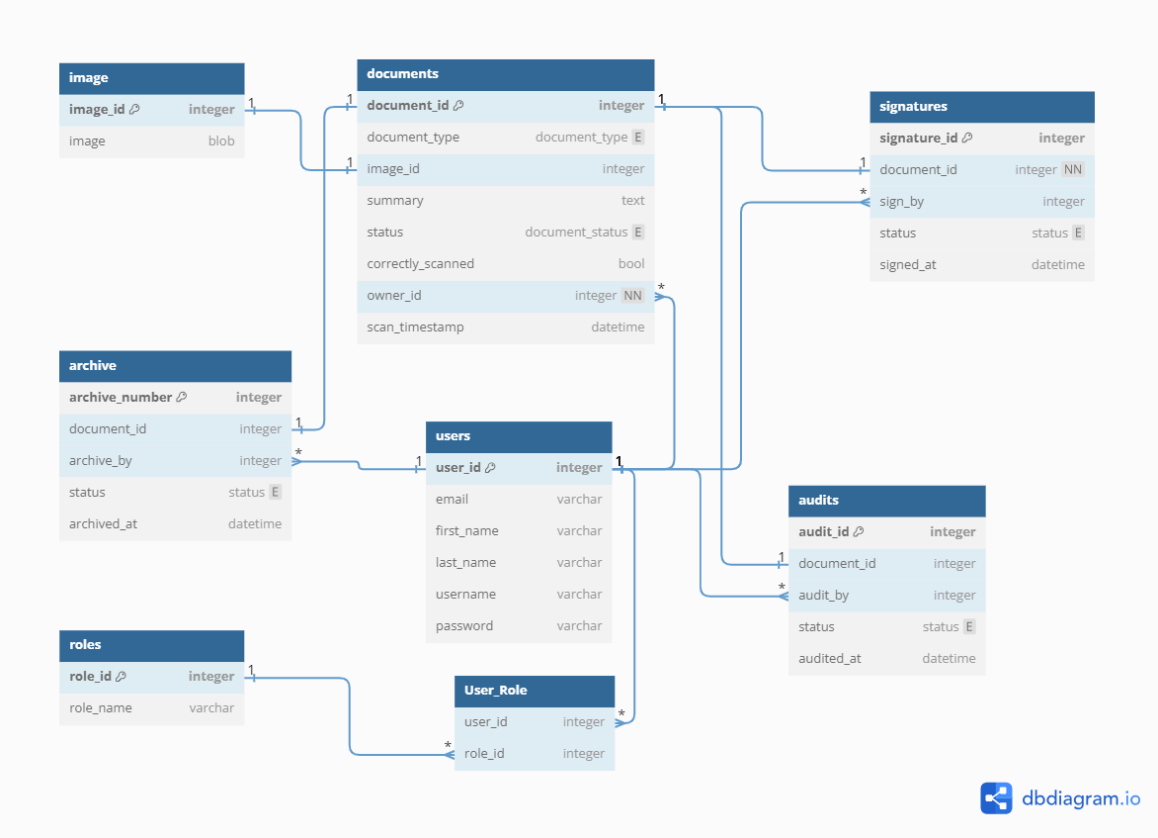
Roles Definira različite uloge korisnika u sistemu. Ovaj entitet u vezi je *One-to-Many* s entitetom **user_role** preko atributa *roleID*.

roles		
roleID	INT	identifikacijski broj kategorije korisnika
roleName	VARCHAR	ime kategorije korisnika

User_role Tablica koja povezuje korisnike s njihovim ulogama. Ovaj entitet u vezi je *Many-to-One* s entitetom **roles** preko atributa *roleID*, u vezi je *Many-to-One* s entitetom **users** preko atributa *userID*.

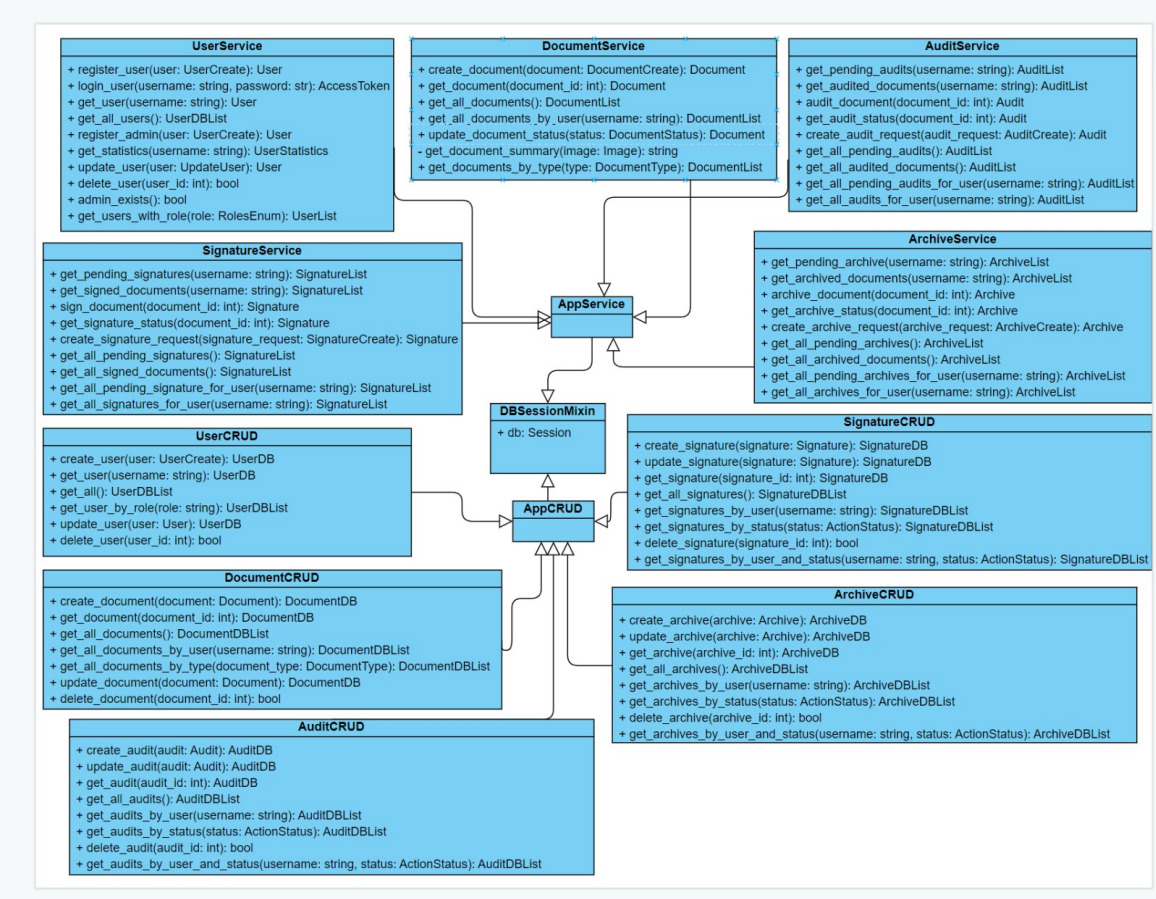
user_role		
userID	INT	identifikacijski broj korisnika
roleID	INT	identifikacijski broj kategorije korisnika

4.1.2 Dijagram baze podataka

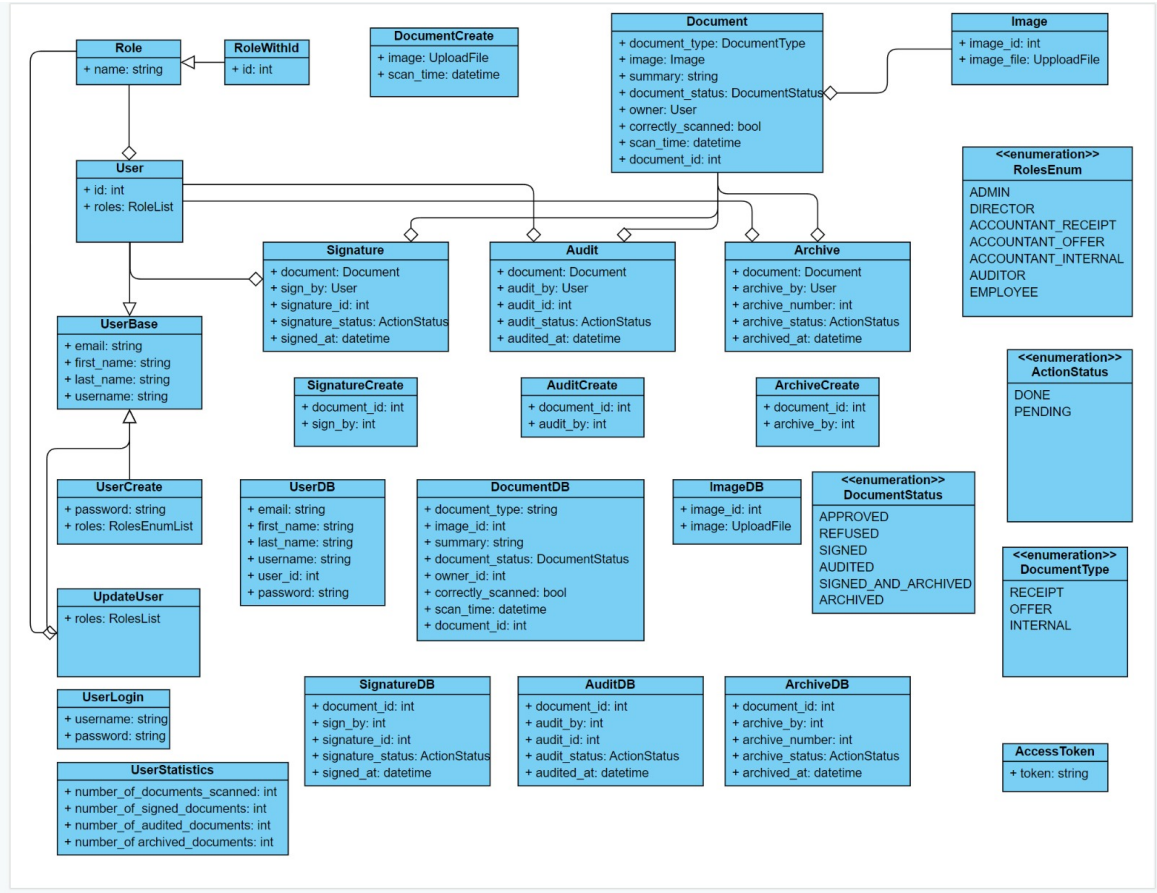


Slika 4.1: Dijagram baze podataka

4.2 Dijagram razreda



Slika 4.2: Dijagram razreda - servisi i CRUD



Slika 4.3: Dijagram razreda - shema i model

4.3 Dijagram stanja

4.4 Dijagram aktivnosti

4.5 Dijagram komponenti

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

1. sastanak
    - Datum: 13. listopad 2023.
    - Prisustvovali: Tvrtko Puškarić, Luka Bradarić Lisić, Karlo Grgičin, Jura Hostić, Andrea Milanović, Katarina Pešić, Martin Subotić
    - Teme sastanka:
        - Početni dogovori vezano za odabir teme projekta
        - Upoznavanje s kolegama
        - Pregled predložene teme
        - Razrada novo-predložene teme
2. sastanak
    - Datum: 28. listopad 2023.
    - Prisustvovali: Tvrtko Puškarić, Luka Bradarić Lisić, Karlo Grgičin, Jura Hostić, Andrea Milanović, Katarina Pešić, Martin Subotić
    - Teme sastanka:
        - Odabir konačne teme
        - Razrada odabranog projekta
        - Odabir platforme za izradu aplikacije
        - Raspodjela zadataka
        - Upoznavanje s LaTex dokumentacijom
3. sastanak
    - Datum: 5.studenoga 2023.
    - Prisustvovali: Tvrtko Puškarić, Luka Bradarić Lisić, Karlo Grgičin, Jura Hostić, Andrea Milanović, Katarina Pešić, Martin Subotić
    - Teme sastanka:
        - Stvoren početni predložak dokumentacije
        - Upoznavanje kolega zaduženih za mobilnu aplikacijom s Flutter razvojnim okvirom

  - Upoznavanje kolega zaduženih za pozadinski server s FastAPI razvojnim okvirom
  - Održana interna radionica upotrebe Git alata
  - Raspodjela zadataka za izradu dokumentacije

4. sastanak
   - Datum: 8.studenoga 2023.
   - Prisustvovali: Tvrtko Puškarić, Luka Bradarić Lisić, Karlo Grgičin, Jura Hostić, Andrea Milanović, Katarina Pešić, Martin Subotić
   - Teme sastanka:
     - Početak izrade dokumentacije
     - Početak izrade mobline aplikacije
     - Početak izrade pozadinskog servera
     - Pregled modela baze podataka i njezino implementiranje
5. sastanak
   - Datum: 13.studenoga 2023.
   - Prisustvovali: Tvrtko Puškarić, Luka Bradarić Lisić, Karlo Grgičin, Jura Hostić, Andrea Milanović, Katarina Pešić, Martin Subotić
   - Teme sastanka:
     - Interne diskusije oko statusa dokumentacije
     - Diskusije oko statusa usvajanja znanja odabranih tehnologija

Tablica aktivnosti

Kontinuirano osvježavanje

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

	Tvrtko Puškarić	**Luka Bradarić Lisić**	**Karlo Grgičin**	**Jura Hostić**	**Andrea Milanović**	**Katarina Pešić**	**Martin Subotić**
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							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Tvrtko Puškarić	**Luka Bradarić Lisić**	**Karlo Grgičin**	**Jura Hostić**	**Andrea Milanović**	**Katarina Pešić**	**Martin Subotić**
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.