

Programsko inženjerstvo

Ak. god. 2023./2024.

Wild Track

Dokumentacija, Rev. 1.

Grupa: *Aristos*

Voditelj: *Josipa Udovičić*

Datum predaje: *17. studenog 2023.*

Nastavnik: *<Ime i prezime nastavnika zaduženog za vašu grupu>*

Sadržaj

1 Dnevnik promjena dokumentacije	3
2 Opis projektnog zadatka	5
3 Specifikacija programske potpore	9
3.1 Funkcionalni zahtjevi	9
3.1.1 Obrasci uporabe	12
3.1.2 Sekvencijski dijagrami	13
3.2 Ostali zahtjevi	14
4 Arhitektura i dizajn sustava	15
4.1 Baza podataka	17
4.1.1 Opis tablica	17
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

Rev.	Opis promjene/dodatka	Autori	Datum
0.1	Napravljen predložak.	Franjo Vuković	04.11.2023.
0.2	Dodani opis projektnog zadatka i funkcionalni zahtjevi.	Josipa Udovičić	04.11.2023.
0.2.1	Promijenjen opis i funkcionalni zahtjevi.	Josipa Udovičić	05.11.2023.
0.3	Dodan dio opisa obrazaca uporabe.	Stela Troškot	05.11.2023.
0.3.1	Dodani svi opisi obrazaca uporabe.	Stela Troškot, Domagoj Jurič	05.11.2023.
0.4	Dodani dijagrami obrazaca uporabe.	Franjo Vuković, Marko Kukolj, Marko Pongrac	05.11.2023.
0.5	Dodani sekvencijski dijagrami.	Franjo Vuković, Marko Kukolj, Marko Pongrac	08.11.2023.

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

Rev.	Opis promjene/dodatka	Autori	Datum
0.5.1	Promijenjeni dijagrami obrazaca uporabe i sekvencijski dijagrami.	Stela Troskot, Domagoj Jurič	09.11.2023.
0.5.2	Dodan opis sekvencijskih dijagrama.	Marko Kukolj	09.11.2023.
0.5.3	Dodani ostali zahtjevi.	Josipa Udovičić	09.11.2023.
0.6	Dodan opis arhitekture.	Josipa Udovičić	09.11.2023.
0.7	Napravljena baza podataka.	Franjo Vuković, Marko Pongrac	09.11.2023.
0.7.1	Napravljen dijagram baze podataka.	Franjo Vuković, Marko Pongrac	09.11.2023.

2. Opis projektnog zadatka

Cilj ovog projekta je razviti programsku podršku za web aplikaciju "Wild Track". Ta aplikacija olakšava korisniku pronalazak i praćenje divljih životinja. Prilikom otvaranja aplikacije prikazuje se karta koja pokazuje gdje se nalazi koja životinja. Praćene životinje imaju na sebi gps uređaj koji aplikaciji odašilje njihovu poziciju i tako korisnik može cijelo vrijeme znati njihovu točnu lokaciju. Kad korisnik odabere koju životinju želi pratiti, može vidjeti neke podatke o njoj, kao na primjer povijesne podatke gdje se nalazila, naziv vrste, slika i opis.

Neregistrirani korisnik je ograničen samo s dosad nabrojanim akcijama, ako želi nešto više s aplikacijom omogućeno mu je prijavljivanje u sustav s postojećim računom (potrebno je upisati korisničko ime i lozinku) ili kreiranjem novog računa. Za kreiranje novog računa potrebni su sljedeći podaci:

- uloga za koju se prijavljuje - može biti istraživač, vođitelj postaje ili tragač na terenu
- korisničko ime
- fotografija
- lozinka
- ime
- prezime
- email adresa

Registracijom u sustav korisniku se dodjeljuju prava istraživača, vođitelja postaje, tragača na terenu ili administratora. Registrirani korisnik može, uz opcije koje ima njegova željena uloga, još i pregledati, mijenjati osobne podatke te izbrišati svoj korisnički račun.

Registracija je završena kad korisnik preko svoje email adrese potvrdi, osim ako je korisnik izabrao biti istraživač ili vođitelj postaje. U tom slučaju, administrator mora potvrditi njihovu ulogu. Administrator sustava ima najveće ovlasti. On ima ovlasti da vidi popis svih registriranih korisnika i njihovih osobnih podataka, odnosno pristup bazi s popisom registriranih korisnika, te može mijenjati njihova dodijeljena prava i osobne podatke.

Voditelj postaje može izabrati koji će tragači biti dio njegove postaje i bira na koji način će oni izvoditi pretraživanje životinja. Postaje su određena mjesta na karti koja voditelj postaje bira, to na primjer može biti postaja Biokovo ili postaja Lonjsko polje.

Ako se korisnik prijavi kao tragač, njemu se na karti prikazuju zadaci koje mora obaviti, trenutna pozicija ostalih tragača koji su aktivni na istoj akciji, te trenutna pozicija životinja koje prate. Tragač tijekom svoje akcije može ostavljati komentare o životinji koju je pratio te također može ostaviti komentar ostalim tragačima i istraživačima, to jest drugim sudionicima u akciji. Također, cijeli put koji tragači prođu se bilježi, odnosno označavaju se staze kojima su putovali i način kojim su se kretali. To će biti potrebno istraživačima koji vizualiziraju njihovo kretanje u obliku toplinskih karata te potom to koriste za analiziranje kretanja životinja i njihovih omiljenih staništa.

Načini kojima se tragači kreću mogu biti različiti, kao na primjer:

- pješke
- dronom
- automobilom
- cross motorom
- brodom
- helikopterom

Ovisno o tome kojom je tragač metodom osposobljen za obavljanje zadataka, svaka metoda je drugačije prikazana na karti. Svaka metoda pruža različitu vidljivost i područje pokrivanja te se na prikladne načine prikazuje na kartama. Ako je tragač odabran da ide pješke njegova će karta biti detaljnija i prikazana na manjem prostoru nego karta tragača koji putuje helikopterom. Tragači koji obavljaju zadatak s pomoću drona, helikoptera ili plovila, moraju na karti imati pravocrtnu rutu. Svaki tragač je osposobljen za samo jednu vozilo i tijekom te akcije se njegov tip prijevoza ne mijenja.

Ako je korisnik odlučio biti istraživač, on može stvoriti nove akcije pretraživanja i praćenja životinja s detaljima o određenim vrstama, jedinkama ili staništima za proučavanje. Svaki istraživač je zadužen za jednu akciju. Ako je istraživaču potreban tragač za pomoć pri istraživanju, istraživač može poslati voditelju postaje zahtjev za tragačima s opisom o potrebnim kvalifikacijama. Voditelj će na taj zahtjev odabrati tragače koji odgovaraju opisu i postaviti ih da sudjeluju u toj akciji. Tragač će biti gotov s akcijom kad završi sve potrebne zadatke. Istraživač, kad

dobije određene tragače, zadaje preko karte zadatke pojedinačno svakom tragaču. Zadaci mogu biti različiti, kao na primjer prolazak određenom rutom i dolazak do određene lokacije te postavljanje kamere ili uređaja za praćenje. Prilikom postavljanja zadatka, istraživač može i ostaviti neke dodatne komentare tragačima. Informacije o poziciji životinje, tragaču i postaji se istraživaču prikazuju preko interaktivne karte. Istraživač može birati da se tijekom izrade karte koriste i neke određene informacije. Na primjer:

- povijesne pozicije praćenih životinja
- filtriranje životinja po vrsti ili pojedinačno po jedinki
- trenutne pozicije praćenih životinja
- povijesne pozicije svih tragača na nekoj akciji
- filtriranje po tipu prijevoza ili pojedinačno po tragaču
- trenutne pozicije tragača koji su aktivni na akciji

Ovaj projekt može biti korisan svima koji bi htjeli više naučiti o divljim životinjama, ali isto tako i ljudima koji se bave proučavanjem životinja može olakšati posao. Svaka osoba koja želi može se ulogirati i postati tragač, u svrhu zabave, dodatnog znanja... A time, dok ljudi to rade iz zabave, odnosno svojevoljno, osobe koji su istraživači mogu koristiti njihove informacije i time im se smanjuje dio posla.

Slične aplikacije poput aplikacije "Wild Track" na području Hrvatske ne postoje. Neke aplikacije koje se bave divljim životinjama su: "eWildLife" i "Divlje životinje".

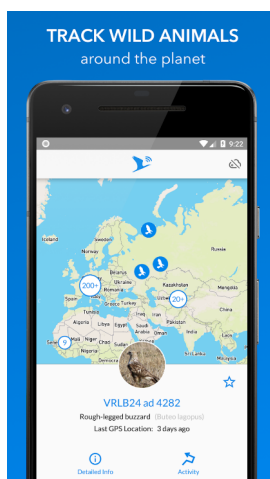
"eWildLife" je aplikacija razvijena u realnom vremenu, ljudi prate i izvještavaju o ubijanju divljih životinja, sukobe čovjek-životinja i viđenja divljih životinja, također mogu i spasiti delfine.

"Divlje životinje" (Slika 2.1) je Android aplikacija koja omogućuje istraživanje svijeta životinja. Aplikacija nudi opsežan katalog divljih životinja, uključujući šumske životinje, životinje iz zooloških vrtova, cirkuske životinje, afričke životinje, šumske životinje itd. Aplikacija pruža detaljan opis svake životinje, uključujući njezinu vrstu, veličinu, kao i podatke o ponašanju i navikama. Također možete pristupiti raznim slikama koje će vam pomoći da bolje razumijete životinje.



Slika 2.1: Aplikacija Divlje Životinje

No, slična aplikacija "Wild Track" aplikaciji izvan Hrvatske postoji. Zove se "Animal Tracker" (Slika 2.2). S pomoću "Animal Tracker" aplikacije moguće je pratiti kretanje divljih životinja diljem svijeta koje se prate u gotovo stvarnom vremenu. Kretanja se prikupljaju GPS oznakama koje životinje nose i pohranjuju se u Movebank (internetska infrastruktura koju koriste istraživači za upravljanje, dijeljenje, analizu i arhiviranje podataka o kretanju životinja).



Slika 2.2: Aplikacija Animal Tracker

Poboljšanja na aplikacijama se uvijek mogu napraviti, pa tako može i na ovoj. Na primjer, moguće je pratiti otkucaje srca životinja, disanje... Time je moguće određivati zdravlje životinje, kretanje životinje, da li životinja spava ili je budna... Tu smo dobili nove komponente aplikacije koje se mogu nazvati trenutne aktivnosti životinje i zdravlje životinje te tu tragači i istraživači imaju još više informacija o životinjama, a ne samo one općenite. Isto tako možemo dodati tragačima opciju da slikaju i snimaju životinje te im dati mogućnost da slike i videe objavljuju na aplikaciju. Time je opet doživljaj određene životinje bolji, a ne općenit.

3. Specifikacija programske potpore

3.1 Funkcionalni zahtjevi

Dionici:

1. Tragač na terenu
2. Istraživač
3. Voditelj postaje
4. Administrator

Aktori i njihovi funkcionalni zahtjevi:

1. Neregistrirani korisnik (inicijator) može:
 - (a) se registrirati u sustav
 - i. dati svoje podatke: korisničko ime, fotografija, lozinka, ime, prezime i email adresa
 - ii. odabrati svoju ulogu(tragač, istraživač ili voditelj postaje)
 - iii. potvrditi registraciju na svojoj email adresi
2. Korisnik (inicijator) može:
 - (a) se prijaviti u sustav
 - (b) upravljati svojim podacima
 - i. pregled osobnih podataka
 - ii. promjena osobnih podataka
 - iii. brisanje korisničkog računa
3. Tragač na terenu (inicijator) može:
 - (a) vidjeti zadatke tijekom akcije
 - i. označiti da je zadatak riješen
 - (b) završiti akciju
 - (c) vidjeti gdje se nalaze ostali tragači na akciji
 - (d) vidjeti koje su dostupne akcije

- i. prihvatiti akciju
 - ii. odbiti akciju
- (e) vidjeti pozicije praćenih životinja
- (f) odabrati vozilo kojim će ići u istraživanje
- (g) vidjeti informacije o životinjama
 - i. dodati komentar
 - ii. izbrisati komentar

4. Istraživač (inicijator) može:

- (a) vidjeti informacije o životinjama
 - i. dodati komentar
 - ii. izbrisati komentar
- (b) vidjeti kartu
 - i. izraditi kartu: na temelju životinja ili tragača
- (c) stvoriti novu akciju
 - i. dodati nove zadatke
- (d) poslati voditelju postaje zahtjev za tragačima
 - i. dodati opis o željenim tragačima
- (e) zadati zadatke tragačima
 - i. upravljati zadacima
 - A. dodavati zadatke
 - B. uređivati zadatke
 - C. brisati zadatke
 - ii. dodati komentare na zadatke
- (f) pregledati informacije

5. Voditelj postaje (inicijator) može:

- (a) pregledati sve tragače
- (b) dodati tragača na akciju
- (c) dodijeliti postaju tragaču

6. Administrator (inicijator) može:

- (a) vidjeti popis svih registriranih korisnika i njihovih osobnih podataka
- (b) obrisati korisnika
- (c) mijenjati dodijeljena prava i osobne podatke registriranim korisnicima

(d) potvrditi istraživača i voditelja postaje

7. Baza podataka (sudionik):

(a) pohranjuje sve podatke o korisnicima

(b) pohranjuje sve podatke o životinjama

(c) pohranjuje staze kojima tragači putuju(i način kojim su se kretali)

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

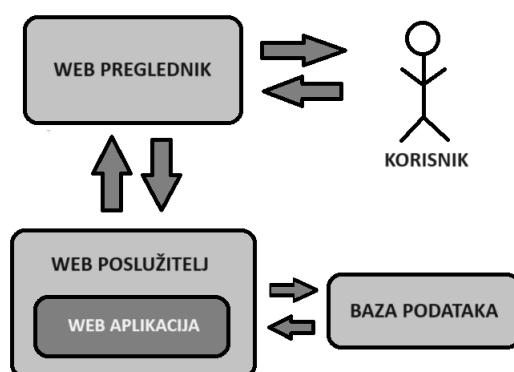
- Sustav treba podržavati istovremeni rad više korisnika u stvarnom vremenu
- Korisničko sučelje i sustav moraju podržavati hrvatsku abecedu
- Pristup bazi podataka trebao bi biti učinkovit, s vremenom izvršavanja unutar nekoliko sekundi
- Za izradu sustava kao web aplikacije, koriste se objektno-orijentirani jezici
- Pogrešna uporaba korisničkog sučelja ne bi smjela imati negativan utjecaj na funkcionalnost i rad sustava
- Sustav treba biti jednostavan za korištenje
- Pri nadogradnji sustava, ne smiju se narušavati postojeće funkcionalnosti
- Veza s bazom podataka mora biti zaštićena, brza i otporna na vanjske greške
- Pristup sustavu treba biti moguć iz javne mreže, uz korištenje HTTPS-a radi sigurne komunikacije

4. Arhitektura i dizajn sustava

Arhitektura sustava može se podijeliti na tri ključna podsustava:

1. Web poslužitelj:
 - (a) Srce web aplikacije.
 - (b) Odgovoran za interakciju između klijenta i aplikacije.
 - (c) Koristi HTTP/HTTPS protokol za prijenos informacija na webu.
 - (d) Inicira pokretanje web aplikacije i prosljeđuje zahtjeve.
2. Web aplikacija:
 - (a) Procesira korisničke zahtjeve i obrađuje ih.
 - (b) Pristupa bazi podataka prema potrebi.
 - (c) Generira odgovore u obliku HTML dokumenata za prikaz u web pregledniku.
3. Baza podataka:
 - (a) Sprema podatke koji se koriste ili modificiraju unutar web aplikacije.

Korisnik, putem web preglednika, šalje zahtjeve web poslužitelju. Web poslužitelj zatim inicira rad web aplikacije, koja procesira zahtjeve, pristupa bazi podataka po potrebi i vraća odgovore u obliku HTML dokumenata. Ova interakcija omogućuje korisnicima pregled i manipulaciju sadržajem putem web sučelja.



Slika 4.1: Arhitektura sustava

React je biblioteka koju smo odabrali za izradu naše web aplikacije, zajedno sa Spring Boot radnim okvirom te programskim jezikom JavaScript. Odabrano razvojno okruženje je Visual Studio Code. Arhitektura sustava temeljiti će se na MVC(Model-View-Controller) konceptu.

React je biblioteka otvorenog koda koja se koristi za izgradnju korisničkih sučelja.

- Jezik: Povezan je s JavaScriptom
- Slučajevi upotrebe: Često se koristi za izradu jednostranih aplikacija, gdje je potrebna brza i prilagodljiva interakcija
- Platforma: Nema specifičnu platformu; može se koristiti u web aplikacijama na različitim platformama

Spring Boot je radno okruženje koje se koristi za izgradnju aplikacija.

- Jezik: prvenstveno povezan s Javom
- Radno okruženje: Spring Boot dio je Spring Frameworka za Java razvoj
- Slučajevi upotrebe: naširoko se koristi za izradu web aplikacija temeljenih na Javi
- Ekosustav: Java ekosustav, s jakom integracijom s tehnologijama kao što su Spring MVC, Spring Data itd.
- Platforma: Java Virtual Machine (JVM)

Spring Boot podržava koncept MVC (Model-View-Controller) arhitekture, a to se postiže kroz Spring Web MVC modul. Spring Boot podržava MVC koncept:

- Model: Spring Boot omogućava korištenje Java objekata kao modela. Ovi objekti predstavljaju podatke koji se koriste u aplikaciji. Spring Data može se integrirati za jednostavno upravljanje podacima i komunikaciju s bazom podataka.
- View: Spring Boot pruža fleksibilnost u odabiru tehnologije za prikazivanje korisničkog sučelja. Prikazi se često implementiraju kroz HTML datoteke, a moguće je koristiti različite template engines (Thymeleaf, FreeMarker, JSP). Pomoću konfiguracija view resolvera jednostavno se integriraju odabrane tehnologije za prikazivanje podataka korisnicima.
- Controller: Anotacije poput @Controller i @RestController omogućuju jednostavno označavanje klasa koje djeluju kao kontroleri. @RequestMapping i slične anotacije omogućuju mapiranje HTTP zahtjeva na određene metode kontrolera. Spring Boot automatski prepoznaje i konfigurira komponente kontrolera.

Primjena MVC koncepta u Spring Boot-u olakšava održavanje i proširivost aplikacija.

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
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	Aplikacija Divlje Životinje	8
2.2	Aplikacija Animal Tracker	8
4.1	Arhitektura sustava	15

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: 9. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

2. sastanak

- Datum: u ovom formatu: 9. studenoga 2023.
- Prisustvovali: I.Prezime, I.Prezime
- Teme sastanka:
 - opis prve teme
 - opis druge teme

Tablica aktivnosti

Kontinuirano osvježavanje

Napomena: Doprinosi 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							
Dijagram razmještaja							

Nastavljeno na idućoj stranici

Nastavljeno od prethodne stranice

	Ime Prezime voditelja	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime	Ime Prezime
Upute za puštanje u pogon							
Dnevnik sastajanja							
Zaključak i budući rad							
Popis literature							
<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.