

**Dokument projektnog zadatka**

**Mobilna aplikacija za saniranje problema u životnoj sredini**

Predmet: Student:

IT355 - Veb Sistemi II Aleksa Stojanović 2285

Profesor: Vladimir Milićević Datum: 14.11.2016.

Asistent: Vuk Vasić Beograd

### Sadržaj

[Sadržaj 2](#_Toc466909429)

[1. Uvod 3](#_Toc466909430)

[1.1. Svrha dokumenta 3](#_Toc466909431)

[1.2.Pregled dokumenta 3](#_Toc466909433)

[2. Korisnici aplikacije 4](#_Toc466909434)

[3. Opis 4](#_Toc466909435)

[3.1. Neophodni moduli/stranice 4](#_Toc466909436)

[3.2. Pretpostavke I ograničenja sistema 5](#_Toc466909437)

[4. Zahtevi I specifikacije 5](#_Toc466909438)

[4.1.Funkcionalni zahtevi 6](#_Toc466909439)

[Krajnji korisnik ( eng. End user ) 7](#_Toc466909440)

[Državni službenik 10](#_Toc466909441)

[4.2. Nefunkcionalni zahtevi 12](#_Toc466909442)

[5. Tehnologije I metodologije 13](#_Toc466909443)

[5.1.1. Tehničke specfikacije 13](#_Toc466909444)

[6.Projektovanje I arhitektura aplikacije 14](#_Toc466909445)

[6.1. Slučaji upotrebe (eng. Use cases) 14](#_Toc466909446)

[6.1.1. Administrator 14](#_Toc466909447)

[6.1.2. Korisnik 19](#_Toc466909448)

[6.1.3. Državni službenik 26](#_Toc466909449)

[6.2. Dizajn baze podataka 31](#_Toc466909450)

[6.2.1.Konceptualni model baze podataka 31](#_Toc466909451)

[6.2.2.Fizički model baze podataka 33](#_Toc466909452)

[6.3.Arhitektura sistema 35](#_Toc466909453)

[6.4. Infrastruktura sistema 38](#_Toc466909454)

[6.5. Dijagrami sekvenci 39](#_Toc466909455)

[6.5.1. Krajnji korisnik 40](#_Toc466909456)

[6.5.2. Državni službenik 50](#_Toc466909457)

[6.5.3. Administrator 57](#_Toc466909458)

# 1. Uvod

Tema ovog projekta jeste sam proces “predevolpment-a”, odnosno dokumentacija sistema ( mobilne aplikacije ) koji će koristiti civilni građani Republike Srbije, kako bi se kreirala specifična interakcija između njih i političkih predstavnika, odnosno, službi za koji su isti zaduženi, a sve u cilju poboljšanja kvaliteta životne sredine.

### 1.1. Svrha dokumenta

## Svrha ovog dokumeta jeste da timu za razvoj sličnih softvera pruži detaljan uvid u principe funkcionisanja samog sistema. Oni će biti razloženi I objašnjeni kroz niz dijagrama i tabela (engl. Unified Modeling Language). Softver koji se koristio u izradi samih modela jeste Power Designer. Ovaj dokument predstavlja “predevelopment” fazu, a ujedno i uputstvo za korišćenje korisnicima koji se , do sada, nisu susretali sa ovakvim vidom aplikacije/sistema.

### 1.2.Pregled dokumenta

Dokumet između ostalog sadrži:

* Detaljnan uvid u dijagram slučaja korišćenja
* Detaljan uvid u svaki slučaj korišćenja
* Dijagrame sekvenci svih slučajeva korišćenja
* Konceptualni model baze podataka
* Fizički model baze podataka
* Dijagram infrastrukture sistema
* Dijagram arhitekture sistema

# 2. Korisnici aplikacije

Analizirajući sve aspekte i funkcionalnosti koje sistem treba da obezbedi, možemo razlikovati tri osnovna tipa korisničkih grupa. Prvu grupu bi predstavljali korisnici (eng. End user-i), odnosno klijenti koji imaju potrebe za uslugama ove aplikacije, drugu administratori koji imaju zadatak da kontrolišu i modifikuju sadržaj koji će samim korisnicima biti na raspolaganju, a treću državni službenici koji su zaduženi da postavljene probleme saniraju. Svaki od pomenutih modula će biti posebno analiziran i projektovan.

Tri korisničke role koje se mogu ustanoviti u okviru samog sistema su:

* Klijent
* Administrator
* Službenik ( U funkciji same države )

Zahtevi, odnosno, slučajevi upotrebe su definisani na nivou svake korisničke uloge u sistemu.

# 3. Opis

Pre izlaganja samih zahteva aplikacije, treba se upoznati sa idejom aplikacijem: na koji način ona funkcioniše I kako je zamišljena od strane kreatora. Napraljen je uvid u neophodne module I stranice kao I u ograničenja Sistema (koje su to prepreke u izradi I koji su nedostaci).

### 3.1. Neophodni moduli/stranice

## 

Modul za krajnje korisnike - U prvoj fazi razvoja trebalo je implementirati nešto najsličnije veb sajt-u, koji će realizovati sve osnovne funkcionalnosti. U ovoj fazi je dovoljna najosnovnija HTML struktura, koja će kasnije biti upotpunjena. Ovde su se projekti I informacije o istima direktno unosili u bazu, samo iz razloga da se verifikuju najosnovnije funkcionalnosti namenjene korisnicima, kao što su mogućnost pregleda svih projekata, odabir konkretnog projekta, komentarisanje odabranog projekta, “Up vote-ovanje” odabranog projekta I slično. Određeni parametri, kao što su specifikacije projekta su unošeni ručno, kako bi bilo izvodljivo testirati sve komponente.

Modul za administratora – Sledeći korak u prvobitnoj fazi razvoja je bio potpuno funkcionalan modul za administratora. Akcenat po pitanju modula bi bio na što jednostavnijem dizajnu. Što se funkcionalnosti tiče, ovaj modul je u suštini klasičan CRUD (engl. Create Read Update Delete) upotpunjen sa nekolicinom dodatnih mogućnosti kao što su pregled liste korisnika kao I uklanjanje određenih projekata, ukoliko dođe do kršenja specifiranih pravila (opisanom u daljem tekstu). U skladu sa potrebama samog administratora, mogu se definisati I implementirati dodatne operacije nad podacima uskladištenim u samoj bazi.

Modul za službenika – Poslednji korak u prvobitnoj fazi razvoja je bio potpuno funkcionalan modul za službenika ( u funkciji države ). Akcenat po pitanju modula bi, takođe, bio na što jednostavnijem dizajnu. Što se funkcionalnosti tiče, fokus je na glavnoj, a ona se odnosi na menjaje statusa samog problema iz “On pending” u “Rešeno”. Kako bi ovo funkcionisalo, službenicima se mora omogučiti efikasan modul za “upload-ovanje” fotografija saniranih problema.

### 3.2. Pretpostavke I ograničenja sistema

Pretpostavka je da je aplikacija veoma jednostavna, stoga korisnik neće imati problema pri samom korišćenju.

Ograničenje trenutnog sistema je činjenica da, treutno, postoji samo jedna kategorija problema kao i jedna služba koja je zadužena za njihovo saniranje, što se ne poklapa sa realnim uslovima. Takođe, proces izrade i testiranja će se realizovati lokalno, stoga postoji šansa da ne uzmemo u obzir nešto što bi poremtilo funkcionisanje u procesu produkcije.

# 4. Zahtevi I specifikacije

Na početku razvoja softvera treba pre svega prikupiti sve korisničke ( odnosno funkcionalne ), a zatim utvrditi I nefunkcionalne zahteve i ovo spada u oblast “Requierments engineering-a”. Taj proces prikupljanja je od neizmerne važnosti, jer na osnovu dobro definisianih zahteva nastaje dobar proizvod I potrebno je izvršiti manje ispravki kada otpočne proces produkcije, što može znatno utičati na troškove testiranja i održavanja samog softvera.

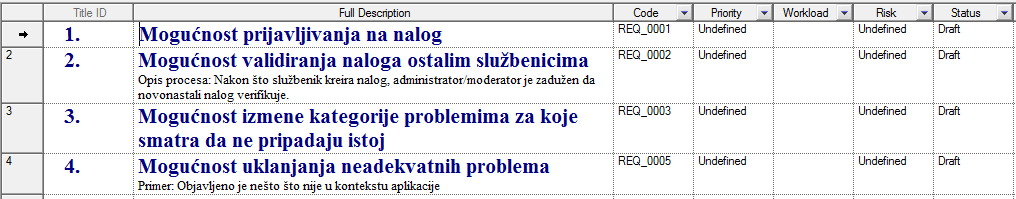
### 4.1.Funkcionalni zahtevi

U nastavku će biti predstavljeni funkcionalni zahtevi aplikacije, uzimajući u obzir najbitnije “stakeholder-e” sistema.

#### **Administrator:**

Administrator ima mogućnost da se prijavi na nalog. Nakon prijave, ima mogućnost da odabere opciju/tab “Validacija naloga”, kako bi verifikovao odnosno odbio novonastale naloge u svojstvu službenika. U slučaju da uoči postavljenih problem (Sa glavne stranice) koji ne odgovara navedenoj kategoriji, on ima mogućnost da istu izmeni. Ukoliko se ispostavi da je uočio problem koji se ne uklapa u sam kontekst aplikacije ( banalan primer - neko je postavio fotografiju hrane ), administrator bi trebalo da ukloni isti.

U nastavku će biti predstavljen tzv. ''Requierments dijagram'' sa svim zahtevima koji se odnose na korisnčku rolu: administrator.



#### *Mogućnost p**rijavljivanja na nalog*

Da bi adminstrator mogao da pristupi svom nalogu (modulu) neophodno je da se prijavi unošenjem korisničkog imena i lozinke nakon čega će mu se pružiti mogućnost obavljanja ostalih aktivnosti ukoliko uspešno prođe procese autentikacije i autorizacije.

Prijava administratora je neophodna kako bi se zaštitilo obavljanje funkcionalnosti za koje će samo administrator biti privilegovan.

* Izmena kategorije postojećeg problema

Administrator može da izmeni kategoriju postojećeg problema. Pre svega administrator treba da odabere problem, iz liste objavljenih problema. Nakon toga, potrebno je da odabere opciju “Izmeni kategoriju” nakon čega će moći da odabere ispravnu kategoriju za odabrani projekat.

* Uklanjanje objavljenog problema

Administrator može da ukloni objavljeni problem, za koji smatra da se ne uklapa u kontekst aplikacije. Odabirom ove opcije, administrator prima poruku da potvrdi uklanjanje. Ukoliko potvrdi, objavljeni problem će biti obrisan.

* Mogućnost validacije naloga novonastalih naloga ( u ulozi službenika )

Poseban modul prikazuje administratoru listu novonastalih korisnika, a on bi trebalo da ih prihvati odnosno odbije.

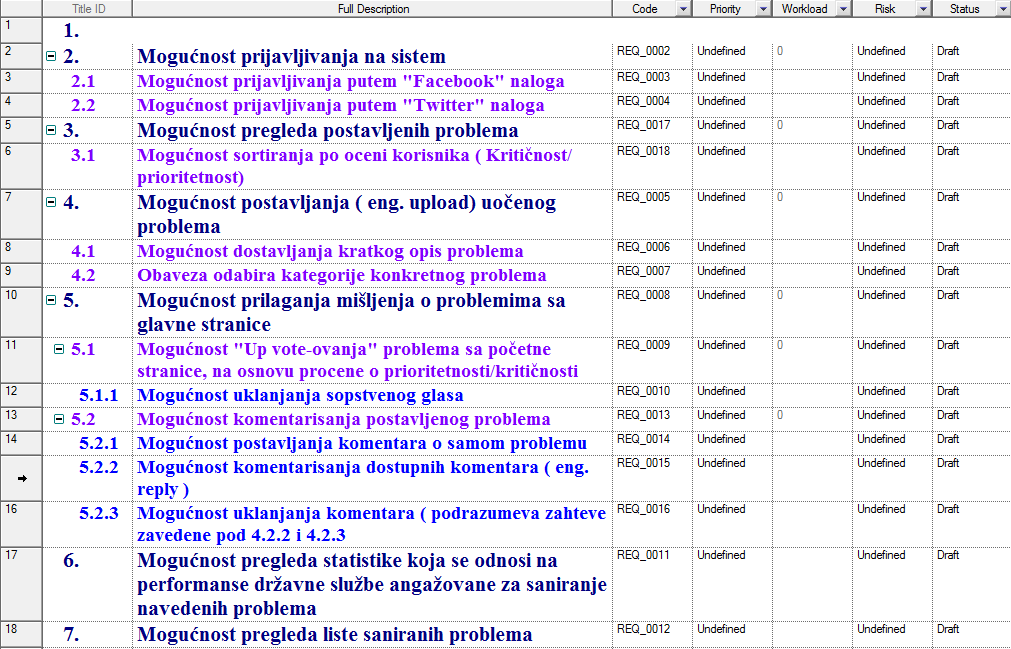
* Mogućnost brisanja neprikladnih komenatara na projektu

Ukoliko shvati da određeni komentar sadrži neprikladan sadržaj, administrator ga može odmah ukloniti.

## **Krajnji korisnik ( eng. End user )**

Na početku korišćenja sistema, korisnik nije u obavezi da napravi nalog. Da bi izabrao određeni projekat, pre svega treba da ode na stranicu gde se nalaze svi projekti. Ukoliko se ispostavi da želi da prokomentariše trenutni projekat, onda je u obavezi da kreira sopstveni nalog. Takođe, ukoliko poželi da i sam postavi određeni problem, bića obavezan da ili kreira nalog, iskoristi već postojeće ( Facebook, odnosno Twitter).

U nastavku će biti predstavljen takozvani ''Requierments dijagram'' sa svim zahtevima koji se odnose na korisnčku rolu: administrator.



#### *Mogućnost kreiranja naloga*

Pri odabiru ove opcije, sistem user-u nudi formu koju treba da popuni. Forma zahteva lične podatke korisnika kao što su: email adresa, korisničko ime i lozinka.

* Mogućnost prijavljivanja na nalog

Ukoliko je prvobitno kreirao nalog, korisnik može da se uloguje. Stranicu mu traži korisničko ime i lozinku koju je korisnik uneo pri praviljenju samog naloga. Druga opcija je prijavljivanje putem “Facebook” ili “Twitter” naloga.

* Mogućnost pregleda postavljenih problema

Odmah nakon prijavljivanja na sistem, korisnik biva preusmeren na glavnu stranicu, stoga ima pregled svih postavljenih problema. Takođe, može da ih sortira po oceni kritičnosti od strane ostalih korisnika.

* Mogućnost pregleda/odabira konkretnog projekta

Nakon što je odabrao opciju “Pregled svih projekata”, korisnik može da odabere konkretan projekat – biće preusmeren na stranicu sa informacijama o tom projektu.

* Mogućnost komentarisanja konkretnog projekta

U slučaju da korisnik poželi da prokomentariše projekat, potrebno je da unese sadržaj komenatara u polje predviđeno za to. Preduslov za ovaj slučaj je prijavljivanje na nalog. ( Komentar se može odnositi na projekat, odnosno na komentar projekta)

* Mogućnost postavljanja uočenog problema

Ukoliko korisnik odabere opciju “postavi problem”, kamera će se automatski uključiti kako bi korisnik mogao da kreira u “upload-uje” fotografiju uočenog problema. Takođe, korisnik moće da ostavi komentar uz fotografiju.

* Mogućnost pregleda statistike koja se odnosi na državne službe angažovane za saniranje navedenih problema

Odabirom opcije “pregled statistike”, korisniku se ukazuje procent rešenih, odnosno nerešenih problema.

* Mogućnost pregleda liste saniranih problema

U slučaju da želi da se pogleda listu saniranih problema, korisnik treba da odabere opciju “Sanirani problemi”.

* Mogućnost “Up vote-ovanja” konkretnog projekta

Ukoliko korisnik želi da “Up vote-uje” određeni projekat ( na osnovu sopstvene procene o kritičnosti/ prioritetnosti), potrebno je da sa glavne stranice, ili nakon što je odabrao konkretan projekat odabere gore navedenu opciju.

* Mogućnost uklanjanja komentara sa projekta

U slučaju da korisnik želi da ukloni komentar sa projekta, potrebno je da odabere opciju “Ukloni komentar”.

* Mogućnost uklanjanja glasa sa projekta

U slučaju da korisnik poželi da ukloni prethodni “Up vote” na konkretnom projektu, potrebno je samo da pritistne ponovo istu opciju.

## **Državni službenik**

Kako bi otpočeo sa korišćenjem sistema, službenik je u obavezi da kreira nalog. Nakon što mu administrator verifikuje novonastali nalog, on može da otpočne sa korišćenjem aplikacije. Slučajevi korišćenja su najsličniji krajnjem korisniku, osim što postoji dodatna opcija – izmena statusa problema u “Rešeno”. Za ovo je potrebno dostaviti fotografiju rešenja, kako bi korisnici stekli uvid o sanaciji postavljenog problema.

U nastavku će biti predstavljen takozvani ''Requierments dijagram'' sa svim zahtevima koji se odnose na korisnčku rolu: administrator.



#### *Mogućnost kreiranja naloga*

Pri odabiru ove opcije, sistem user-u nudi formu koju treba da popuni. Forma zahteva lične podatke korisnika kao što su: email adresa, korisničko ime i lozinka. Takođe, neophodno je da korisnik odabere opciju “u funkciji službenika”.

* Mogućnost prijavljivanja na nalog

Ukoliko je prvobitno kreirao nalog, korisnik može da se uloguje. Stranicu mu traži korisničko ime i lozinku koju je korisnik uneo pri praviljenju samog naloga. Druga opcija je prijavljivanje putem “Facebook” ili “Twitter” naloga.

* Mogućnost pregleda postavljenih problema

Odmah nakon prijavljivanja na sistem, korisnik biva preusmeren na glavnu stranicu, stoga ima pregled svih postavljenih problema. Takođe, može da ih sortira po oceni kritičnosti od strane ostalih korisnika.

* Mogućnost pregleda/odabira konkretnog projekta

Nakon što je odabrao opciju “Pregled svih projekata”, korisnik može da odabere konkretan projekat – biće preusmeren na stranicu sa informacijama o tom projektu.

* Mogućnost komentarisanja konkretnog projekta

U slučaju da korisnik poželi da prokomentariše projekat, potrebno je da unese sadržaj komenatara u polje predviđeno za to. Preduslov za ovaj slučaj je prijavljivanje na nalog. ( Komentar se može odnositi na projekat, odnosno na komentar projekta)

* Mogućnost postavljanja uočenog problema

Ukoliko korisnik odabere opciju “postavi problem”, kamera će se automatski uključiti kako bi korisnik mogao da kreira u “upload-uje” fotografiju uočenog problema. Takođe, korisnik moće da ostavi komentar uz fotografiju.

* Mogućnost pregleda statistike koja se odnosi na državne službe angažovane za saniranje navedenih problema

Odabirom opcije “pregled statistike”, korisniku se ukazuje procent rešenih, odnosno nerešenih problema.

* Pregled mape na kojoj su istaknute lokacije postavljenih problema

Odabirom opcije “Mapa”, službenik može steći uvid u lokacije ( na mapi ) svih postavljenih problema, kako bi kreirao rutu za sanaciju istih.

* Mogućnost menjanja statusa problema u rešeno

U slučaju da je sanirao ( ili učestvovao u sanaciji problema), korisnik može da promeni status postavljenog problema u “rešeno”. Neophodno je da odabere konkretan problem, a nakon toga da odabere gore navedenu opciju. Automatski će se uključiti kamera, kako bi korisnik mogao da kreira fotografiju sanacije.

### 4.2. Nefunkcionalni zahtevi



Nefunkcionalni zahtevi su zahtevi koji nisu direktno povezani sa servisima koje sistem treba da obezbedi svojim korisnicima, već sa svojstvim tih servisa, kao što su: performanse, pouzdanost, brzina odgovora, zauzeće memorije, bezbednost, raspoloživost, prenosivost, lakoća održavanja. Oni definišu ograničenja implementaciji sistema, kao što su na primer, svojstva U/I uređaja, ili predstavljanje podataka na interfejsu sa drugim sistemima.

* Performanse

Performanse su krucijalni zahtev za većinu sistema. U konretnom primeru, one nisu imperativ, iz razloga što sistem nije kritičan, ali to ne znači da celokupno kotiranje samog softvera neće biti uvećano ukoliko se sve izvršava u optimalnom vremenu. S obzirom da se sistem skladišti na dva servera (jedan za samu aplikaciju, drugi za bazu podataka) šanse za sporo učitavanje stranica ili izvlačenja sadržaja, tj. informacija iz baze podataka (putem krajnje jednostavnih upita, koji nisu zahtevni ni vremenski ni memorijski) su maksimalno redukovane.

* Lakoća upotrebe

Sistem zahteva veoma jednostavnu, jednočasovnu obuku, u okviru koje će se administratori upoznati i ovladati svim funkcijama sistema, odnosno modula.

* Zaštita

Sistem ne zahteva visok stepen zaštite, razlog je što je mogućnost manipulacije sa sadržajem baze podataka ograničen isključivo na modul za administratora, a sami administratori su obezbeđeni autentikacijom i autorizacijom.

* Raspoloživost

Sajt treba da bude dostupan za korišćenje na svim tipovima uređaja poput tablet urađaja i mobilnih telefona što zahteva prilagodljiv (responsive) dizajn.

* Lakoća održavanja

Akcenat je na tome da se olakša razvoj eventualnih naknadnih funkcionalnosti tako što treba težiti da se u što većoj meri koriste postojeće tj. već razvijene komponente.

* Bezbednost

Najkritičnija komponenta sistema je sama mogućnost elektronskog plaćanja. S da se sve transkacije izvršavaju uz pomoć tzv. Trećeg lica, preostalo je samo da se iskoriste dobri metodi(koji podrazumevaju dobar izbor algoritama uz ''salt'' tehniku) za šifre korisnika pri beleženju i autentikaciji prilikom rada sa bazom podataka. Takođe, zamisao je da se informacije o potencijalnom investitoru ne čuvaju u bazi podataka, stoga se može konstatovati da će problemi ovog tipa biti eskivirani.

# 5. Tehnologije I metodologije

Što se tiče razvojnih metodologija s obzirom da se radi o sistemu u kome su od strarta zahtevi precizno definisani opredelio sam se za se vodopad (engl. Waterfall) metodu razvoja softvera u kombinaciji sa agilnom metodom koja bi se manifestovala kroz inkrementalni razvoj. To bi značilo da je akcenat na realizaciji početnih, tj. osnovnih zahteva zarad ispunjenja funkcionalnosti, a zatim će naredne verzije softvera proširivati sistem u skladu sa dodatnim potrebama.

### 5.1.1. Tehničke specfikacije

U ovom odeljku će ukratko biti prikazane tehnologije koje će biti korišćenje u izradi same aplikacije. Prvenstveno treba naglasiti da je arthitektura aplikacije MVC ( eng. Model View-Controller), kako bismo mogli raščlanimo sistem na komponente i detaljnije opišemo tehnologije zadužene za kreiranje svake od njih.

S obzirom da se radi o mobilnoj aplikaciji koja ne spada u igrice, niti zahteva specifičan rad sa animacijama, odlučio sam se za “Ionic” framework koji će uz pomoć alata kao što su “Phone gap” i “Cordova” konvertovati Angular2 ( praktično rešenje koje radi u web browser-u ) u Native mobilnu aplikaciju za “IOS” i “Android” platformu. Iz ovoga se nameće zaključak da ćemo View-Controller komponente kreirati koriseći Angular2 Framework ( kombinacija HTML5 i TypeScript-a). Što se samog modela tiče, on će komunicirati sa bazom podataka putem RESTful servisa koji će biti izgrađeni u poznatom Java framework-u - “Spring”. U procesu izrade aplikacije biće korišćen “MySQL”, dok će u produkciji biti korišćen “PostgreSQL”Datatbase Management System.

# 6.Projektovanje I arhitektura aplikacije

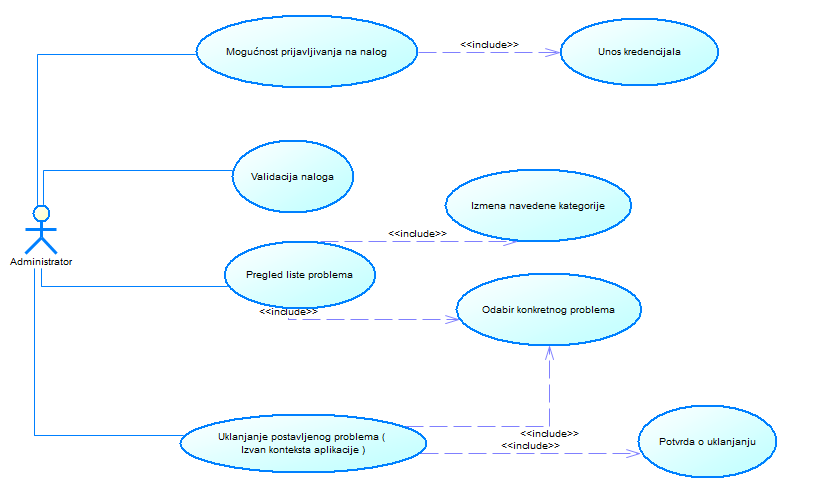
Projektovanje Sistema se zasniva na dijagramima koji prikazuju ponašanje sistema, stanja u kojima se nalaze, bazu podataka, procese koje vrši sistem itd.. Tu spadaju UML dijagrami kao što su klasni dijagram, konceptualni, dijagram stanja, dijagram aktivnosti, flowchart dijagram...

### 6.1. Slučaji upotrebe (eng. Use cases)

Slučaji upotrebe prikazuju koje mogućnosti korisnici sistema imaju I kako su one međusobno povezane. Svi slučaji upotrebe nastaju direktno iz funkcionalnih zahteva korisnika. U nastavku će biti prikazani i detaljno opisani svi navedeni slučajevi korišćenja.

### 6.1.1. Administrator

Na slici 1 možemo videti potencijalne slučajeve korišćenja koji odgovaraju administratoru.

****

**Prijavljivanje na nalog**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost prijavljivanja na nalog. |
| Preduslovi: | Administrator ima pristup internetu. |
| Učesnici: | Administrator i sistem (pasivno) |
| Opis slučaja korišćenja: | Administrator unosi korisničko ime I lozinku kako bi pristupio nalogu I samim tim bio u mogućnosti da koristi aplikaciju kroz odgovarajući modul koji je dizajniran za njega. |
| Osnovi scenario: | Administrator pristupa stranici na kojoj se nalazi login forma koju treba da ispuni.  Popunjava login formu unosom koriničkog imena i lozinka i potvrđuje unos klikom na dugme.  Ukoliko su korisničko ime i lozinka odgovarajući, korisnik će biti obavešten u uspešnom logovanju i biće preusmeren na stranicu za administrator-a. |
| Alternativni scenariji: | Korisničko ime i lozinka ne postoje, administratoru će se prikazati poruka da je uneo pogrešno korisničko ime ili šifru što će ga ponovo vratiti na korak 2.  Sistem prikazuje poruku o grešci i nudi korisniku opciju da kreira nalog. |
| Postuslovi: | Administrator je obavešten o uspešnom prijavljivanju, preusmeren je na svoju stranicu i može da nastavi sa korišćenjem aplikacije. |

**Pregled svih projekata**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost pregleda svih projekata. |
| Preduslovi: | Administrator ima pristup internetu.  Administrator je odabrao opciju ‘’Pregled svih problema'’. |
| Učesnici: | Administrator i sistem (pasivno) |
| Opis slučaja korišćenja: | Administrator želi da pregleda postavljenje probleme. |
| Osnovi scenario: | Administrator odabira opciju “Pregled svih problema”, stoga mu sistem vraća komponentu koja sadrži sve projekte. |
| Alternativni scenariji: | NA |
| Postuslovi: | Administrator je, uspešno, stekao uvid u postojeće projekte. |

**Mogućnost pregleda/odabira konkretnog projekta**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost odabira konkretnog projekta. |
| Preduslovi: | Administrator ima pristup internetu.  Administrator je odabrao opciju ‘’Pregled svih problema'’. |
| Učesnici: | Administrator i sistem (pasivno) |
| Opis slučaja korišćenja: | Administrator želi da pregleda određeni projekat. |
| Osnovi scenario: | Administrator odabira konkretan projekat iz liste svih dostupnih projekat, sistem mu vraća komponentu koja sadrži sve informacije o projektu. |
| Alternativni scenariji: | NA |
| Postuslovi: | Administrator je, uspešno, stekao uvid u određeni projekat. |

***Validiranje novonastalnih naloga***

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Validiranje korisničkih naloga ( Cilj: Omogućiti pristup službenicima) |
| Preduslovi: | Administrator se uspešno prijavio na sistem. |
| Učesnici: | Administrator i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Administrator iz liste naloga potvrđuje, odnosno “decline-uje” kreirane naloge. |
| Osnovi scenario: | 1. Administrator otvara stranicu sa listom novonastalih naloga. 2. Administrator odabira konkretan nalog i bira opciju “Accept” ili “Decline” |
| Alternativni scenariji: | NA |
| Postuslovi: | Administrator je uspešno “accept-ovao” odnosno “decline-ovao” naloge službenika |

***Izmena kategorije objavljenog problema***

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Izmena kategorije objavljenog problema |
| Preduslovi: | Administrator se uspešno prijavio na sistem. |
| Učesnici: | Administrator i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Administrator sa početne stranice ( koja se sadrži listu svih problema) odabira problem i odlučuje se za izmenu kategorije ( razlog: Neadekvatnost). |
| Osnovi scenario: | 1. Adminstrator pristupa glavnoj stranici ( Sadrži sve postavljene probleme) 2. Nakon toga odabira konkretan problem, i bira opciju “Izmeni kategoriju”. 3. Nakon izmene, potvrđuje unos. |
| Alternativni scenariji: | NA |
| Postuslovi: | Administrator je uspešno izmenio neodgovarajuću kategoriju određenog problema. |

***Uklanjanje postavljenog problema***

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Ukalanjanje postavljenog problema |
| Preduslovi: | Administrator se uspešno prijavio na sistem. |
| Učesnici: | Administrator i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Administrator sa početne stranice ( koja se sadrži listu svih problema) odabira problem i odlučuje se za uklanjanje istog ( Razlog: ne uklapa se u kontekst aplikacije) |
| Osnovi scenario: | 1. Adminstrator pristupa glavnoj stranici ( Sadrži sve postavljene probleme) 2. Nakon toga odabira konkretan problem, i bira opciju “Ukloni objavljeni problem”. 3. Nakon odabira, potvrđuje izbor. |
| Alternativni scenariji: | NA |
| Postuslovi: | Administrator je uspešno uklonio neodgovarajući problem. |

### 6.1.2. Korisnik

Na slici 2 se može videti dijagram svih slučajeva korišćenja koji se odnose na krajnjeg korisnika.

Illustration 1: Slika 2: Slučajevi korišćenja, korisnička rola - krajnji korisnik

**Mogućnost kreiranja naloga**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost kreiranja naloga. |
| Preduslovi: | Korisnik ima pristup internetu. |
| Učesnici: | Korisnik i sistem (pasivno) |
| Opis slučaja korišćenja: | Korisnik kreira nalog, kako bi prokomentarisao određeni projekat, ili doprineo, finansijski. |
| Osnovi scenario: | 1. Korisnik zahteva formu za pravljenje naloga. 2. Sistem vraća formu za kreiranje naloga. 3. Korisnik popunjava formu za prijavu unosom koriničkog imena, lozinke i email adrese. 4. Korisnik potvrđuje unos klikom na dugme. 5. Sistem vraća poruku o uspešno kreiranom nalogu. |
| Alternativni scenariji: | 3.1.1. Korisnik je uneo već postojeće korisničko ime.  3.1.2. Sistem prikazuje poruku o grešci.  3.2.1. Korisnik je uneo dve različite lozinke.  3.2.2. Sistem izbacuje poruku o grešci.  3.3.1. Korisnik je uneo pogrešan format lozinke.  3.3.2. Sistem izbacuje poruku o grešci. |
| Postuslovi: | 1. Korisnik je napravio nalog. |

**Prijavljivanje na nalog**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Prijavljivanje na nalog |
| Preduslovi: | Krajnji korisnik ima pristup internetu. |
| Učesnici: | Korisnik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Korisnik unosi sopstvene kredencijale, tj. Korisničko ime i lozinku kako bi pristupio sistemu. U slučaju da se odluči za prijavljivanje na sistem putem “Facebook” ili “Twitter” naloga, potrebno je samo da odabere jednu od tih opcija. |
| Osnovi scenario: | 1.Korisnik pristupa stranici na kojoj se nalazi login forma koju treba da ispuni ( Ukoliko se odlučio za kreiranje “tradicionalnog” naloga). U suprotnom samo odabira neke od navedenih opcija.  2.Popunjava login formu unosom koriničkog imena i lozinke i potvrđuje unos klikom na dugme ( ili pritiskanjem taster „Enter“.  3.Ukoliko su korisničko ime i lozinka odgovarajući, korisnik će biti preusmeren na stranicu kreiranu za njegove potrebe |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Korisnik se uspešno prijavio na sistem. |

**Mogućnost pregleda svih projekata**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost pregleda svih projekata. |
| Preduslovi: | Korisnik ima pristup internetu.  Korisnik je odabrao opciju ‘’Pregled svih projekata'’. |
| Učesnici: | Korisnik i sistem (pasivno) |
| Opis slučaja korišćenja: | Korisnik želi da pregleda dostupne projekte. |
| Osnovi scenario: | Korisnik odabira opciju “Pregled svih projekata”, stoga mu sistem vraća komponentu koja sadrži sve projekte. |
| Alternativni scenariji: | NA |
| Postuslovi: | Korisnik je, uspešno, stekao uvid u postojeće projekte. |

**Mogućnost pregleda/odabira konkretnog projekta**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Mogućnost odabira konkretnog projekta. |
| Preduslovi: | Korisnik ima pristup internetu.  Korisnik je odabrao opciju ‘’Pregled svih projekata'’. |
| Učesnici: | Korsinik i sistem (pasivno) |
| Opis slučaja korišćenja: | Korisnik želi da pregleda određeni projekat. |
| Osnovi scenario: | Korisnik odabira konkretan projekat iz liste svih dostupnih projekat, sistem mu vraća komponentu koja sadrži sve informacije o projektu. |
| Alternativni scenariji: | NA |
| Postuslovi: | Korisnik je, uspešno, stekao uvid u određeni projekat. |

**Mogućnost ostavljanja komenatara**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Komentarisanje problema |
| Preduslovi: | 1. Krajni korisnik se prijavio na sistem.  2. Krajnji korisnik je odabrao konkretan problem I “odabrao opciju komentariši”. |
| Učesnici: | Službenik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Korisnik odabira konkretan problem, i ostavlja sopstveni komentar. Postoji mogućnost i ostavljanja komentara na komentar sa odabranog problema. |
| Osnovi scenario: | 1. Korisnik je odabrao konkretan problem sa stranice.  2.U slučaju da se odlučio za komentar ( na navedeni problem) unosi tekst u predviđeno polje i potvrđuje unos.  U slučaju da se odlučio za komentar na neki od komentara, odabira opciju “reply”, i nakon toga unosi tekst u predviđeno polje i potvrđuje sopstveni unos. |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Korisnik je uspešno prokomentarisao konkretan problem, odnosno komentar konkretnog problema. |

**Postavljanje problema**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Postavljanje uočenog problema. |
| Preduslovi: | 1.Korisnik se prijavio na sistem.  2.Korisnik je odabrao opciju “postavi problem”. |
| Učesnici: | Korisnik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Korisnik odabira opciju “Postavi problem”, što će prouzrokovati automatsko uključivanje kamere. Nakon što korisnik napravi fotografiju, biće upitan da li je adekvatna. Ukoliko jeste, dobiće opciju da doprinese i komentarom ( o konkretnom problemu). |
| Osnovi scenario: | 1. Korisnik je odabrao opciju “Postavi problem”.  2. Kamera se automatski uključuje, kako bi korisnik mogao da kreira fotografiju samog problema.  3. Sistem “prompt-uje” pitanje “Da li ste zadovoljni fotografijom?”  4. Ukoliko korisnik odabere opciju “ne”, ponavljaju se koraci 2. i 3.  5. Ukoliko je sistem zadovoljan fotografijom, ostaje mu samo da predviđeno polje ispuni tekstom, koji služi kao deskripcija problema). |
| Alternativni scenariji: | 1. Sistem “Upload-ovanja” ne fukcioniše, korisnik biva obavešten, a napravljena fotografija se čuva u memoriji telefona. |
| Postuslovi: | Krajnji korisnik je uspešno postavio problem. |

**Pregled postavljenih problema**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Pregled postavljenih problema. |
| Preduslovi: | 1.Korisnik se prijavio na sistem. |
| Učesnici: | Korisnik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Nakon što se korisnik prijavio na sistem, biće preusmeren na “glavnu stranicu”, tako da može steći uvid u postavljene probleme. |
| Osnovi scenario: | 1. Korisnik se prijavio na sistem, automatski biva preusmeren na “glavnu stranicu”.  2. Korisnik ima mogućnost da sortira probleme na osnovu “ocene kritičnosti”. |
| Alternativni scenariji: | NA |
| Postuslovi: | Korisnik stiče uvid u postavljene probleme ( od strane ostalih korisnika ). |

**“Up vote-ovanje” problema**

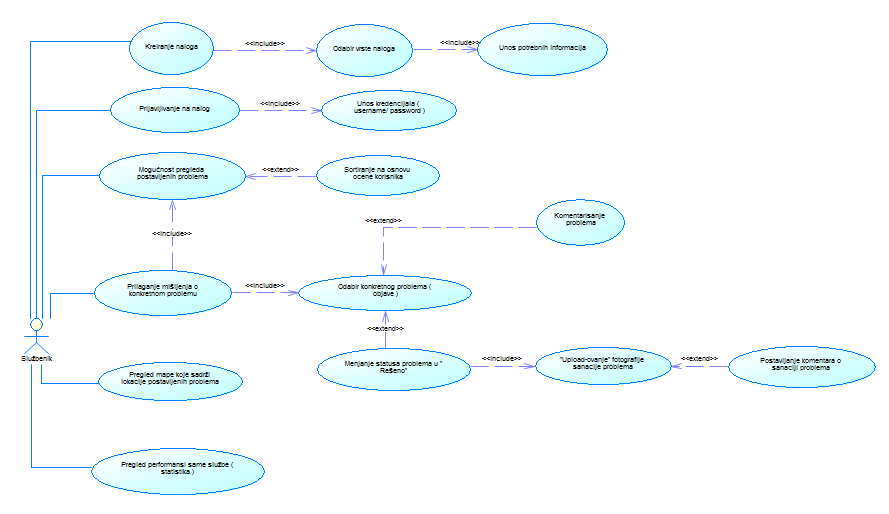
|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | “Upvote-ovanje” problema. |
| Preduslovi: | 1.Korisnik se prijavio na sistem.  2.Korisnik se nalazi na glavnoj/početnoj stranici. |
| Učesnici: | Korisnik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Nakon što se korisnik prijavio na sistem, biće preusmeren na “glavnu stranicu”, tako da može steći uvid u postavljene probleme. Ukoliko mu se određeni problem učini kritičnim, odabira opciju “upvote”. |
| Osnovi scenario: | 1. Korisnik se prijavio na sistem, automatski biva preusmeren na “glavnu stranicu”.  2.Korisnik “odabira” konkretan problem.  3. Korisnik odabira opciju “Upvote” kako bi ukazao na to da bi bilo od većeg značaja sanirati konkretan problem. |
| Alternativni scenariji: | NA |
| Postuslovi: | Korisnik je uspešno ukazao na kritičnost određenog problema. |

**Pregled saniranih problema**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Pregled saniranih problema. |
| Preduslovi: | 1.Korisnik se prijavio na sistem.  2. Korisnik je odabrao opciju “Sanirani problemi”. |
| Učesnici: | Korisnik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Nakon što je korisnik odabrao opciju “Sanirani problemi” biće preusmeren na stranicu sa listom saniranih problema. |
| Osnovi scenario: | 1. Korisnik odabira opciju ( iz menija ) “Sanirani problemi”.  2. Biva preusmeren na stranicu koja sadrži listu saniranih problema. |
| Alternativni scenariji: | NA |
| Postuslovi: | Korisnik stiče uvid u probleme rešene od strane službe. |

### 6.1.3. Državni službenik

Na slici 3 se može videti dijagram svih slučajeva korišćenja koji se odnose na državnog službenika.

Illustration 2: Slika 3: Dijagram slučajeva korišćenja, korisnička rola - državni službenik

**Kreiranje naloga**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Kreiranje naloga |
| Preduslovi: | Pristup internetu. |
| Učesnici: | Administrator i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Službenik se registruje kako bi kasnije mogao da pristupa sistemu. |
| Osnovi scenario: | 1. Službenik pristupa formi predviđenoj za kreiranje naloga. 2. Odabira opciju “službenik”. 3. Predviđenu formu ispunjava svim neophodnim informacijama. 4. Potvrđuje unos. |
| Alternativni scenariji: | 1. Nisu uneti podaci (ili nisu validni) - sistem obaveštava službenika o tome. 2. Službenik ponovo unosi informacije u polja na koje mu je sistem ukazao. |
| Postuslovi: | Službenik je uspešno kreirao nalog i iščekuje potvrdu od strane administratora. |

**Prijavljivanje na nalog**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Prijavljivanje na nalog |
| Preduslovi: | Administrator ima pristup internetu. |
| Učesnici: | Administrator i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Administrator unosi sopstvene kredencijale, tj. Korisničko ime i lozinku kako bi pristupio sistemu. |
| Osnovi scenario: | 1. Korisnik pristupa stranici na kojoj se nalazi login forma koju treba da ispuni. 2. Popunjava login formu unosom koriničkog imena i lozinke i potvrđuje unos klikom na dugme ( ili pritiskanjem taster „Enter“. 3. Ukoliko su korisničko ime i lozinka odgovarajući, korisnik će biti preusmeren na stranicu kreiranu za njegove potrebe |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Korisnik (službenik ) se uspešno prijavio na sistem. |

**Komentarisanje problema**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Komentarisanje problema |
| Preduslovi: | Službenik se, prvenstveno prijavio na nalog. |
| Učesnici: | Službenik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Službenik odabira konkretan problem, i ostavlja sopstveni komentar. |
| Osnovi scenario: | 1. Službenik je odabrao konkretan problem sa stranice. 2. U slučaju da se odlučio za komentar ( na navedeni problem) unosi tekst u predviđeno polje i potvrđuje unos. 3. U slučaju da se odlučio za komentar na neki od komentara, odabira opciju “reply”, i nakon toga unosi tekst u predviđeno polje i potvrđuje sopstveni unos. |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Korisnik ( službenik ) je uspešno prokomentarisao konkretan problem. |

**Menjanje statusa problema u “Rešeno”**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Menjanje statusa problema u “Rešeno”. |
| Preduslovi: | 1. Službenik se prijavio na sistem. 2. Službenik je odabrao konkretan problem i odabrao opciju “izmeni status”. |
| Učesnici: | Službenik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Službenik odabira konkretan problem, i menja status iz “on pending” u “solved” tako što prilaže fotografiju sanacije kao ( obavezno) uz komentar ( opciono ). |
| Osnovi scenario: | 1. Službenik je odabrao konkretan problem sa stranice I odabrao opciju “Izmeni status”.   2. Sistem obavezuje službenika da “Upload-uje” fotografiju koja predstavlja sanaciju problema.  3. Sistem nudi opciju prilaganja komentara uz fotografiju uz “Upload-ovanu” fotografiju. |
| Alternativni scenariji: | 1. Sistem “Upload-ovanja” ne fukcioniše, stoga je službenik obavešten da bi trebalo ponovo da ponovi proces. |
| Postuslovi: | Službenik je uspešno promenio status problema u “solved”. |

**Pregled mape koja sadrži lokacije postavljenih problema**

|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Pregled mape koja sadrži lokacije postavljenih problema. |
| Preduslovi: | 1.Službenik se prijavio na sistem.  2.Službenik je odabrao opciju “Mapa”. |
| Učesnici: | Službenik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Službenik odabira opciju “Mapa” kako bi stekao uvid u lokacije problema koje treba rešiti. |
| Osnovi scenario: | 1. Službenik je odabrao opciju “Mapa” kako bi stekao uvid u lokacije objavljenih problema.  2. Sistem mu vraća mapu ispunjenu “Pinpoint-ovima” koji ukazuju na lokacije postavljenih probelma. |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Sistem “Upload-ovanja” ne fukcioniše, stoga je službenik obavešten da bi trebalo ponovo da ponovi proces. |

**Pregled performansi same službe**

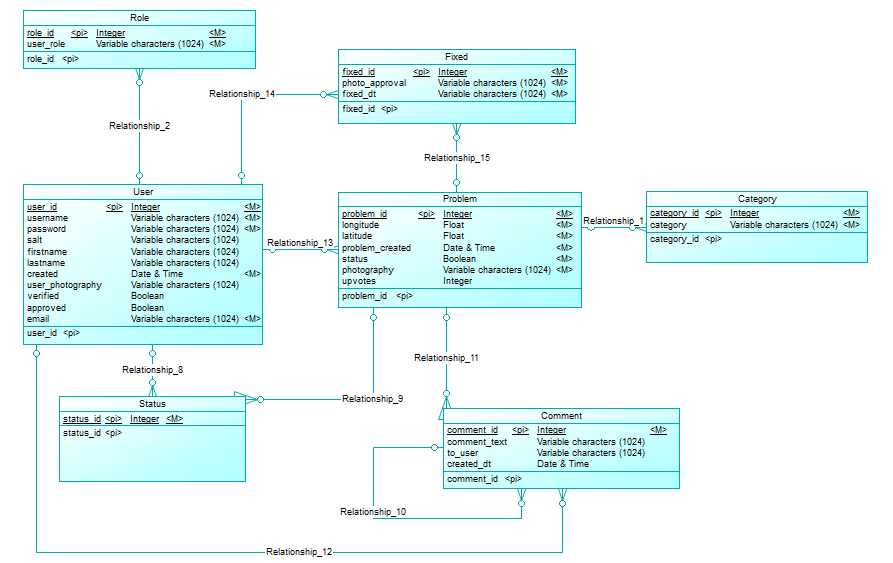
|  |  |
| --- | --- |
| Naziv slučaja korišćenja: | Pregled performansi same službe. |
| Preduslovi: | 1.Službenik se prijavio na sistem.  2.Službenik je odabrao opciju “Pregled statistike”. |
| Učesnici: | Službenik i sistem ( pasivno ) |
| Opis slučaja korišćenja: | Službenik odabira opciju “Performanse” kako bi stekao uvid u odnos rešenih – nerešenih problema. |
| Osnovi scenario: | 1. Službenik je odabrao opciju “Performanse” kako bi stekao uvid u broj rešenih, odnosno nerešenih problema.  2. Sistem mu vraća grafik koji ukazuje na uspešnost same službe. |
| Alternativni scenariji: | NA |
| Postuslovi: | 1. Službenik je uspešno stekao uvid u odnos rešenih i nerešenih problema. |

### 6.2. Dizajn baze podataka

Svaki složeniji sistem mora da sadrži bazu podataka. U zavisnosti od toga šta se čuva u samoj bazi, zavisi i veličina baze. Međutim, bez obzira na veličinu, mora da postoji neki prikaz iste koji je ukratko opisuje. Iz tog razloga se kreira konceptualni model baze podataka kao i fizički koji je njegova konkretizacija.

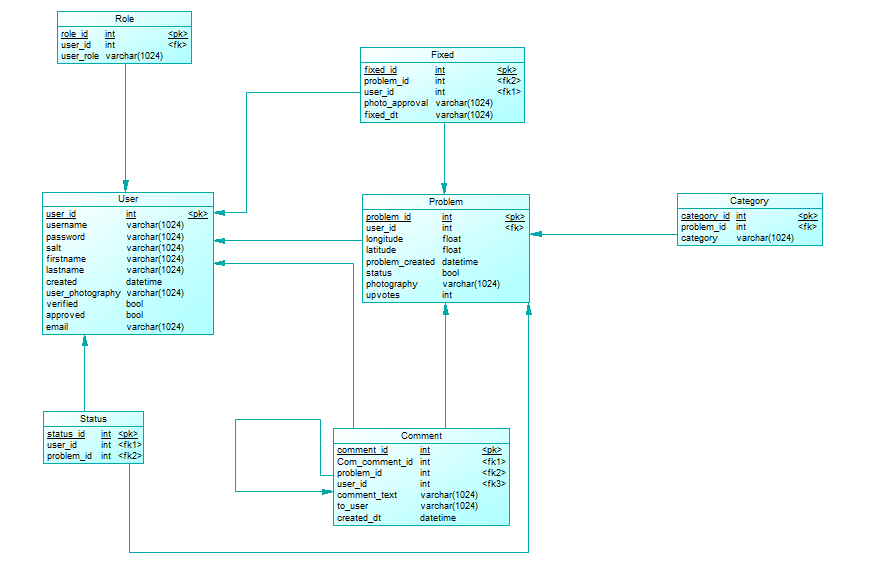
### 6.2.1.Konceptualni model baze podataka

Na slici 4. je prikazan konceptualni model baze podataka.

Illustration 3: Slika 4: Konceptualni model baze podataka

### 6.2.2.Fizički model baze podataka

Na slici 5 je prikazan fizički model baze podataka, koji je generisan na osnovu prethodno prikazanog modela.

Illustration 4: Slika 5: Fizički model baze podataka

Na osnovu fizičkog modela baze podataka može uočiti nekoliko jednostavnih entiteta, a u nastavku će svaki od njih, pojedinačno, biti detaljnije opisan kako bi se stekao što bolji uvid u deo sistema koji čuva naše informacije.

1. Entitet “User”:

Ovaj entitet predstavlja realnog korisnika, stoga se u njemu od atributa nalaze “userID” ( Auto inkrementalna vrednost), “ , “username” - kolona koja čuva odabrano korisničko ime, “password”, kolona koja čuva odabrani password ( hešira se pri kreiranju naloga, bezbednosti radi) I smešta u atribut “salt”. Od atributa koji nisu obavezni imamo: “Firstname”, “Lastname”, “User photography”, kako bismo sačuvali ime, prezime, odnosno fotografiju ( profilnu sliku ) korisnika. Atribut “Verified” služi kako bi korisnik verifikovao svoj account ( verifikaciju putem mail adrese ), a atribut “approved” se odnosi isključivo na službenike – putem njega proveramo da li je sam administrator verifikovao novonastalni nalog.

1. Entitet “Role”

Ovaj entitet je veoma jednostavan, a svrha je da čuva uloge dodeljene svakom korisniku.

1. Entitet “Comment”

Ovaj entitet je, takođe, veoma jednostavan, s tim što se pri nastanku svakog komentara u ovu tabelu unose ID korisnika, kao i ID projekta kako bismo imali predstavu ko je i na kojem projektu ostavio komentar. Takođe, atribut “Com\_comment\_id” nam služi da čuvamo referencu na tzv. Parrent comment – u slučaj da je komentar odgovor na neki od već postojećih komentara. Atribut “to\_user” služi da sačuva referencu na korisnika kojem je komentar upućen.

1. Entitet “Problem”

Moglo bi se reći da je ovo entitet od najvećeg značaja za čitav projekat, a od atributa sadrži: "problem\_id” - Auto increment atribut koji čuva ID samog problema, “user\_id” ( eng. Foreign key) entiteta User, kako bismo imali predstavu o korisniku koji je postavio problem, “longitude” i “latitude” kako bismo imali predstavu o lokaciji problema, “problem\_created” - vreme i datum postavljavanja problema, “status” - atribut koji čuva status samog problema ( on pending i solved, po kreiranju problema je - “on pending” ), “photography”, “Upvotes” - kako bismo čuvali broj glasova kao i “description” koji čuva opis samog problema.

1. Entitet “Fixed”

Ovaj entitet služi da čuva informacije o rešenim problemima. Od atributa sadrži: “fixed\_id” - auto increment atribut koji čuva jedinstveni broj rešenog problema, “problem\_id” - > atribut koji ukazuje na to koji problem je rešen ( strani ključ ), “user\_id” koji nam ukazuje na korisnika (rola: službenik) koji je postavio sanaciju problema, “photo\_approval” - atribut koji čuva adresu fotografije priloženu zarad potvrde sanacije, kao i “fixed\_dt” koji čuva datum i vreme sanacije problema.

1. Entitet “Status”

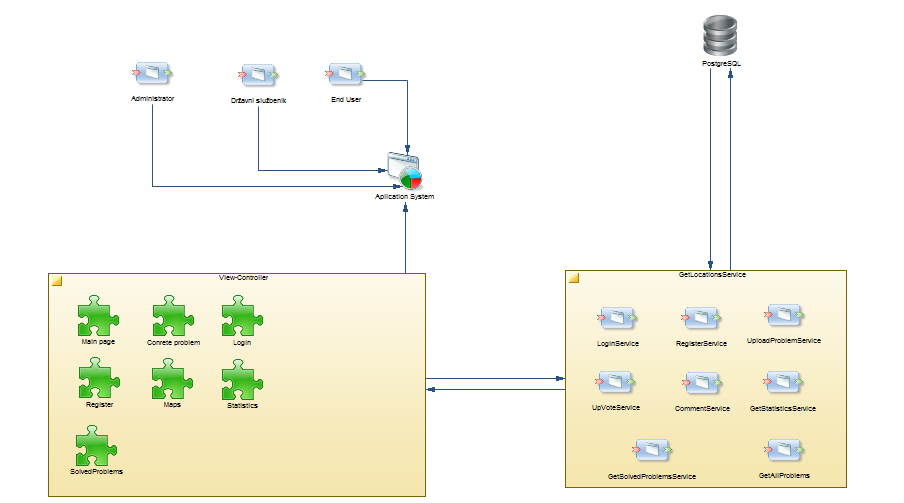
Ovaj entitet je krajnje jednostavan, a za svrhu ima čuvanje informacija o “Up vote-ovima”, tj. Koji korisnik je “Up vote-ovao” određeni projekat, kao i vreme i datum kada je to uradio.

1. Entitet “Category”

Ovaj entitet ima za svrhu čuvanje informacija o kategoriji samog problema. Inicijalno postoji jedinstvena kategorija, “rupe na putevima”, ali će ih kasnije biti više, a ovaj entitet ima za svrhu obezbeđivanje dodatnih kategorija.

### 6.3.Arhitektura sistema

Na slici 6.4.1. je prikazan dijagram arhitekture sistema.



Slika 6.4.1. – Dijagram arhitekture

Prvenstveno je potrebno obezediti bazu podataka koja predstavlja relacionu mySQL bazu podataka. U slučaju nepredviđenih problema(kao što su pad servera), bilo bi poželjno da postoji rezervna baza podataka, ali to nije imperativ ( ukoliko se redovno vrši proces tzv. Back up-ovanja podataka). Svi servisi služe za manipulaciju nad bazom podataka s tim što svi nude osnovne CRUD (Create Read Update Delete) operacije, a I malo kompleksnije upite.

Komponente Login, Register, Problem, Konkretan Problem, Mape i Statistika predstavljaju Angular2 komponente koje čine samu aplikaciju. Sve ove komponente kao svoj template koriste HTML stranice, kao i, po potrebi, određene forme.

Softverska arhitekture same aplikacije, kao što se vidi na priloženom dijagramu predstavlja takozvanu “Model – View – Controller” arhitekturu, ili preciznije – softverski dizajn pattern.

MVC arhitektura je patern koji odvaja prikaz informacija od interakcije korisnika sa tim informacijama. Model se sastoji od podataka aplikacije, pravila, logike i funkcija. View može da bude bilo koji izlazni prikaz podataka. Kontroler uzima ulazne podatke i konvertuje ih u komande za model ili view. Arhitektura samog Angular2 framework-a predstavlja arhitekturu sačinjenu iz komponenti, ali uz ovakvu implementaciju modela ( tj. Servisa ) može se reći da ovo jeste MVC arhitektura. Zamisao ovakvih aplikacije je da podele čitav sistem na tri dela, tj. “View” koji predstavlja korisnički interfejs, “Controller” koji je aktivan listener na akcije samog korisnika, i “Model” koji direktno upravlja bazom podataka, logikom i pravilima definisanim u okviru same aplikacije. U nastavku sleduje grafički prikaz “Angular2 component” arhitekture.

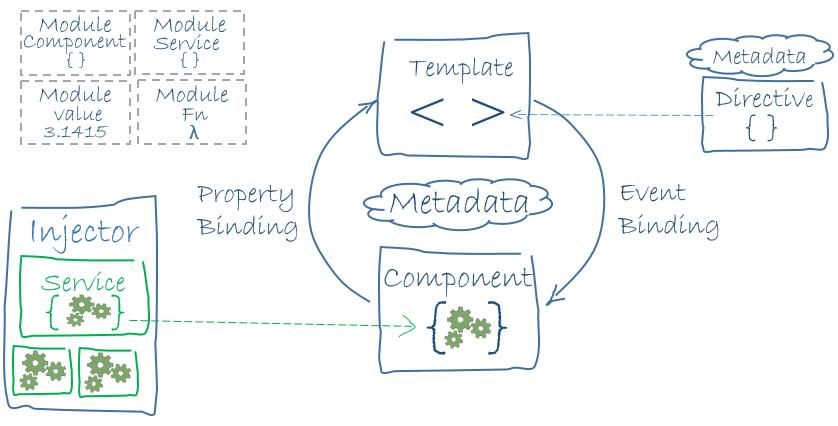
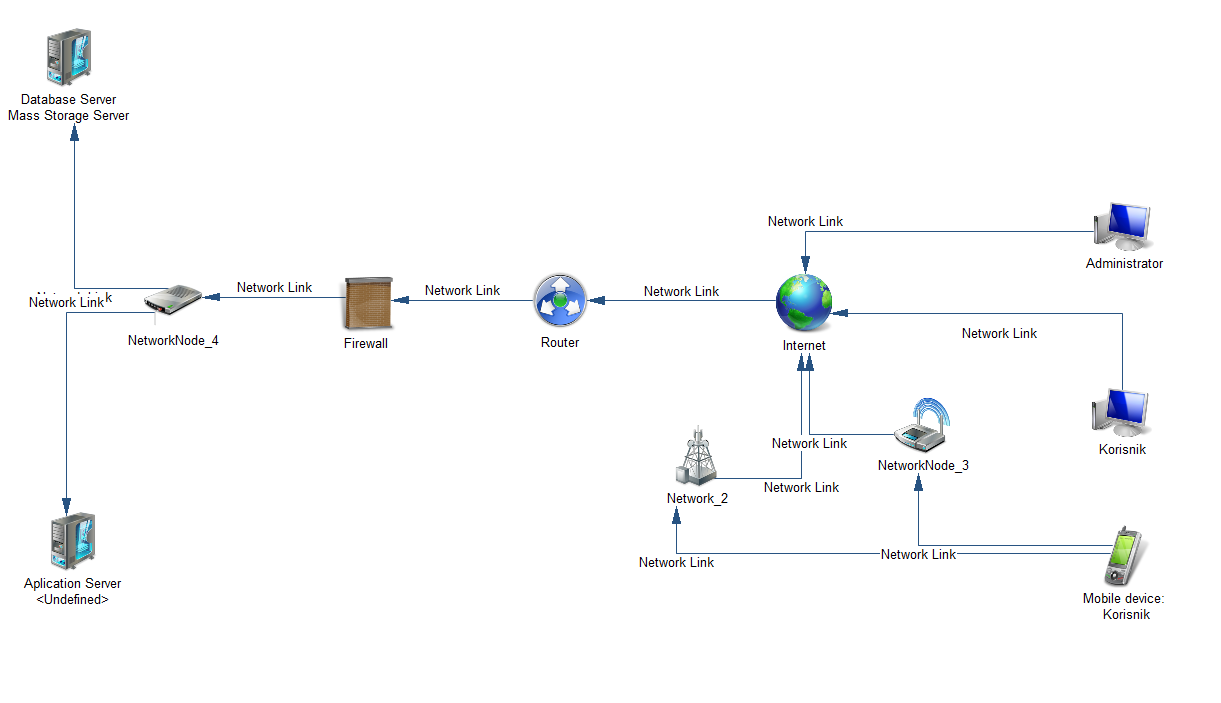


Illustration 5: Predstavljanje najvažniji gradivnih blokova Angular2 aplikacije

### 6.4. Infrastruktura sistema

Na slici 6.5.1. se može videti dijagram infrastrukture sistema.



Slika 6.5.1. – Dijagram infrastrukture

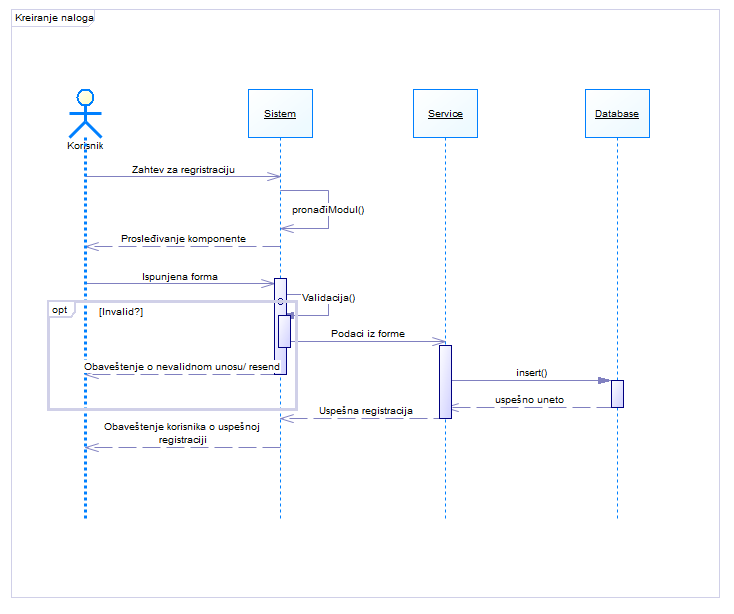
Ovaj dijagram ukazuje na krajnje jednostavnu infrastrukturu – uključuje najosnovnije elemente, neopohodne za funkcionisanje jednog veb sistema. Administrator može pristupiti sistemu putem interneta preko bilo kog računara (izostavljena je mogućnost pristupa putem mobilnog telefona, iz bezbednosnih razloga), dok je klijentima omogućen pristup sistemu takođe koristeći računar, ali i putem mobilnih uređaja. Firewall nije neizostavan element, s obzirom da dobro konfigurisani serveri nemaju nikakve beneficije od istog. Treba istaći hub koji kontroliše tok saobraćaja, između dva servera, jedan koji služi za skladištenje same aplikacije, drugi kao skladište same baze podataka. Na taj način se može izbeći preterano opterećenje jednog od ova dva servera, ukoliko sam sistem mora da odgovara na zahteve mnogobrojnih korisnika. Na serveru bi se nalazili svi elementi koji su navedeni u dijagramu arhitekture softvera.

### 6.5. Dijagrami sekvenci

Za svaki od osnovnih scenarija koji su opisani biće reprezentovan dijagram sekvenci i pored toga će biti opisan tok interakcije u izuzetnim situacijama.

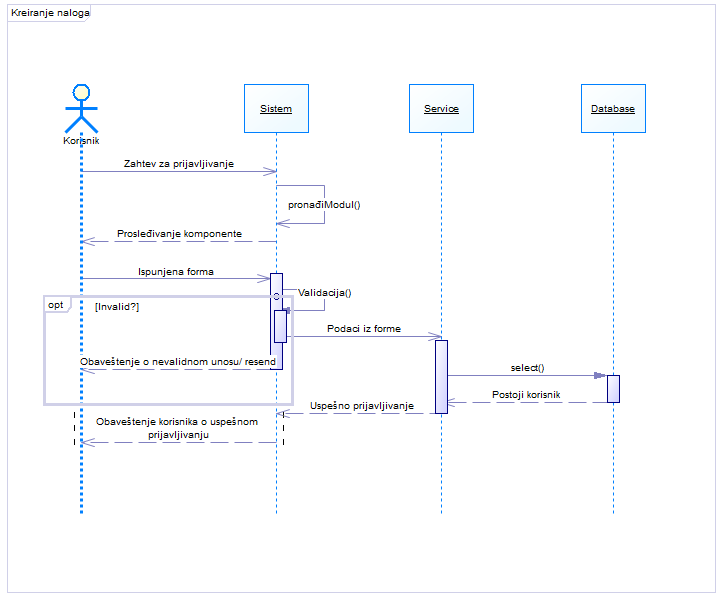
### 6.5.1. Krajnji korisnik

***Kreiranje naloga/registracija***

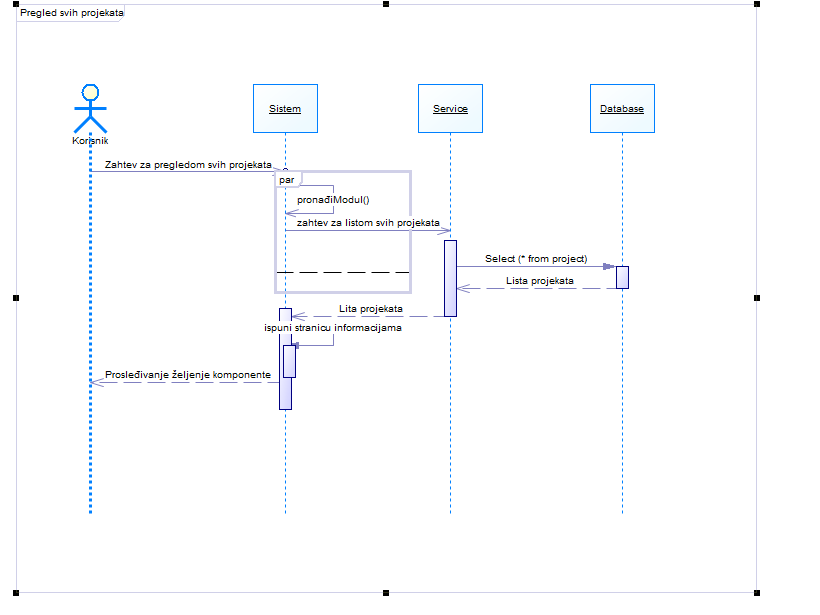


Pri pravljenju naloga, sistem treba da proveri da li je korisnik uneo podatke validnog formata. Ukoliko je to ispunjeno, “registerService” će kreirati nalog korisniku. Ukoliko se to ne desi, na slici dijagrama se može videti kako sistem reaguje i koje poruke razmenjuje sa korisnikom i bazom.

***Prijavljivanje na nalog***

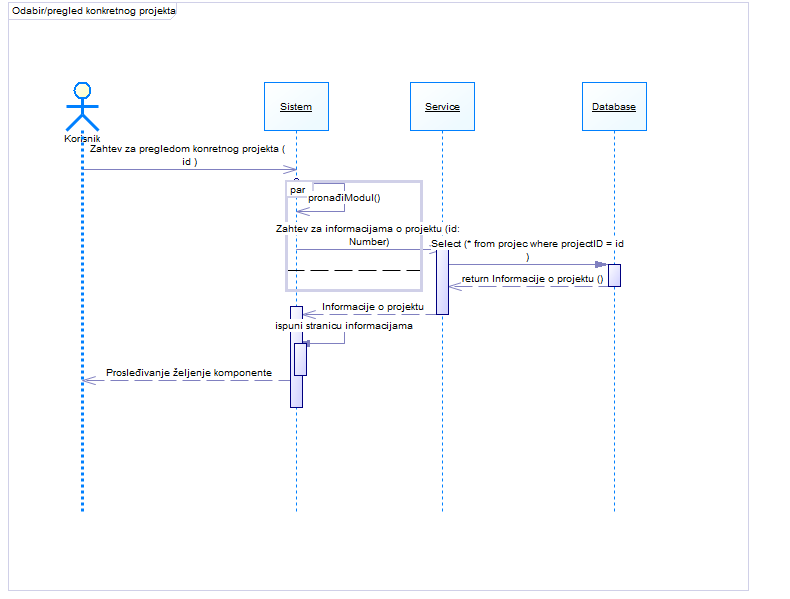
******

***Pregled svih problema***

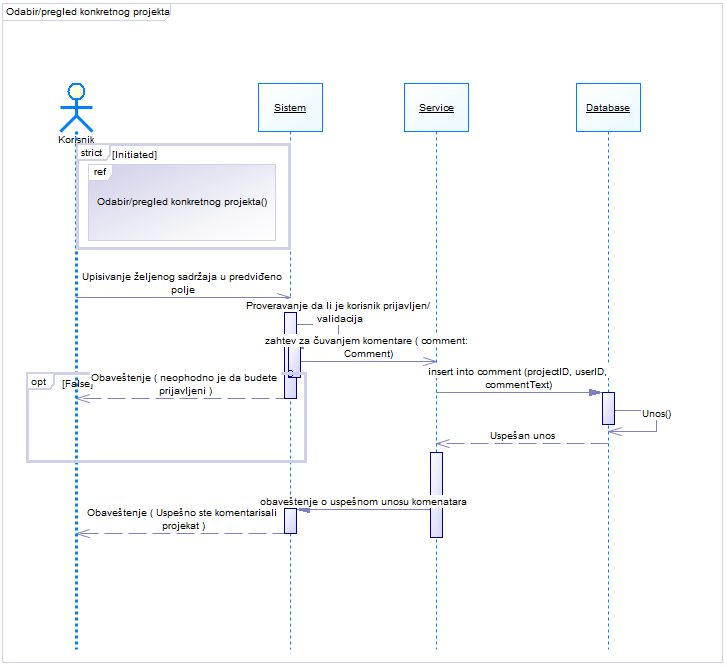


6.6.2.5.– Sekvencijalni dijagram: Pregled svih projekata

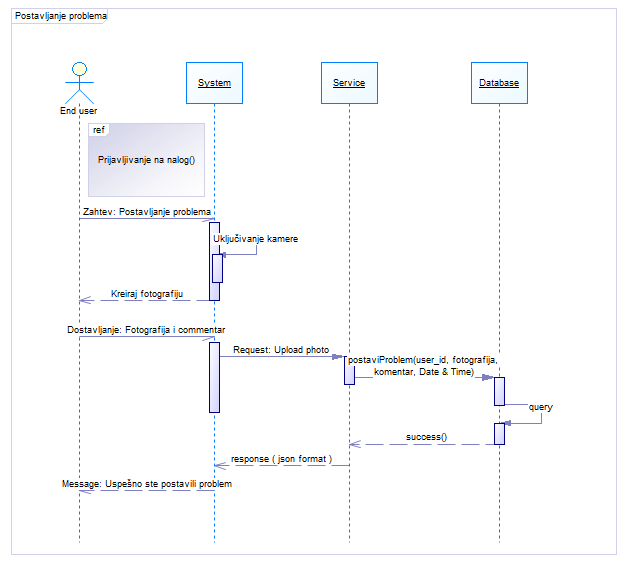
***Odabir konkretnog problema***

******

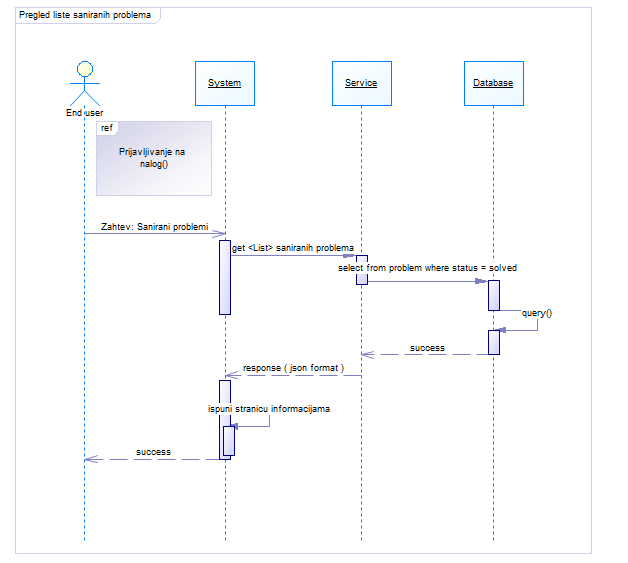
***Komentarisanje konkretnog problema***



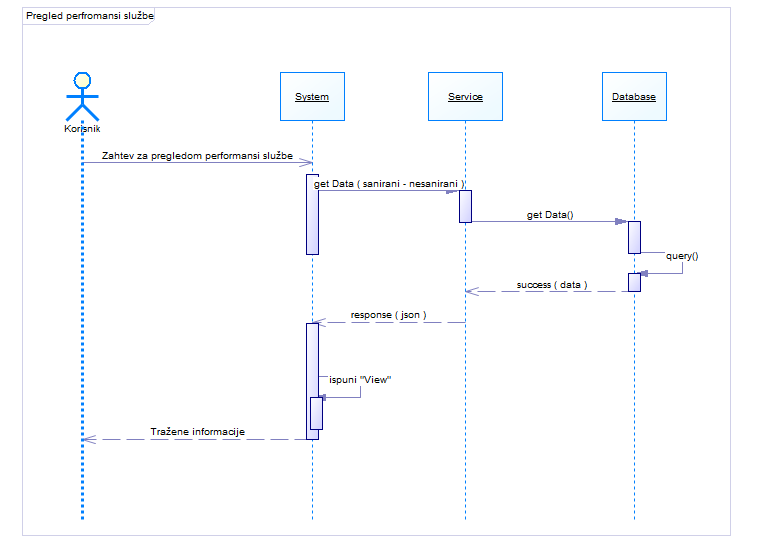
***Postavljanje problema***



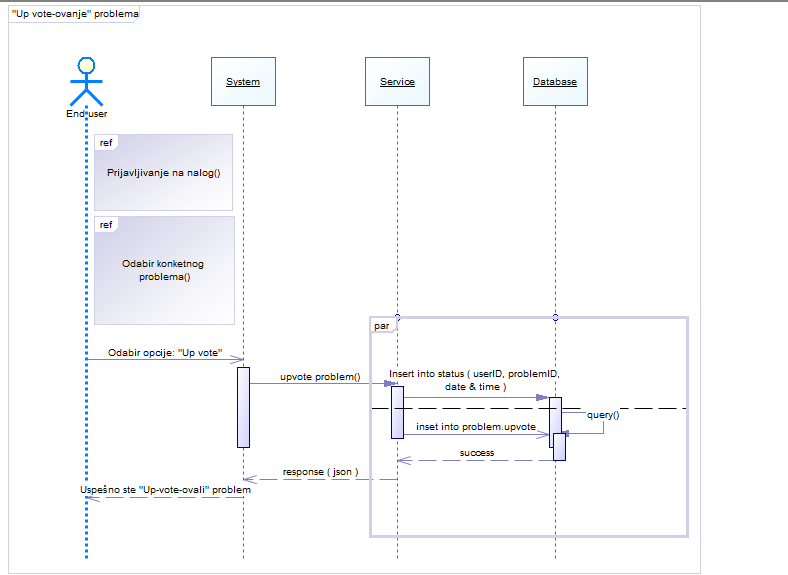
***Pregled liste saniranih problema***

******

***Pregled statitstike***

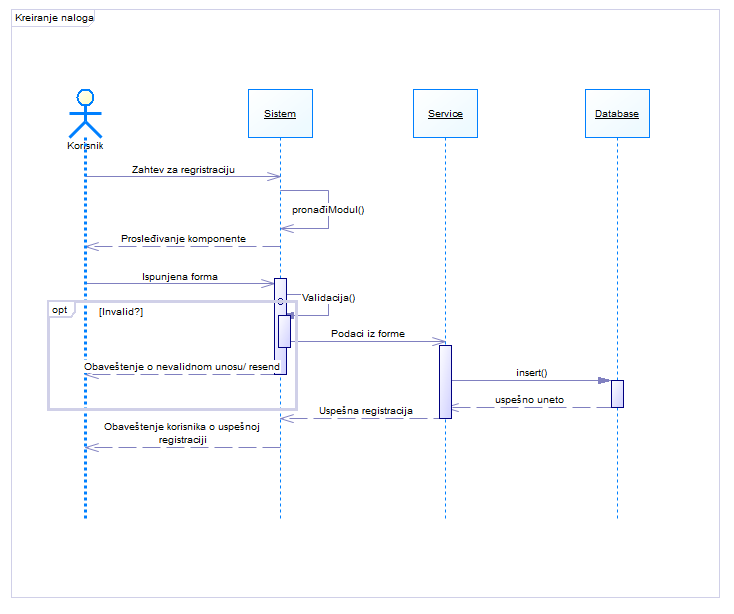


***“Up vote-ovanje” problema***

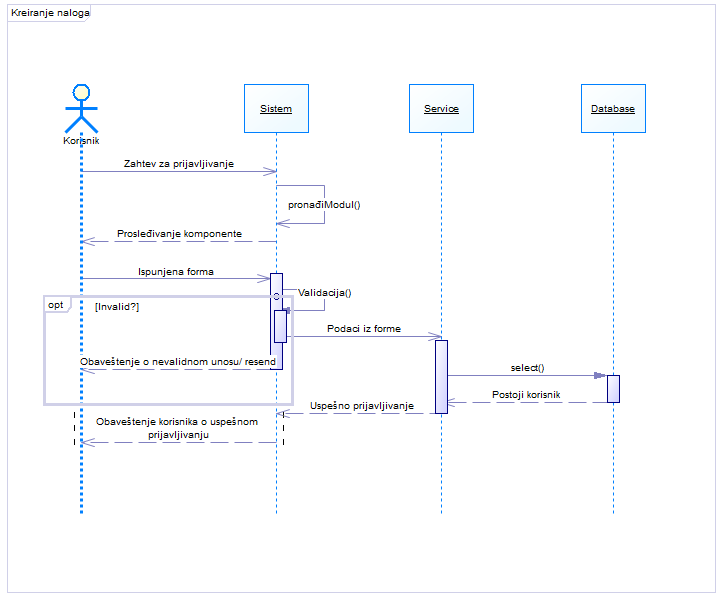
******

### 6.5.2. Državni službenik

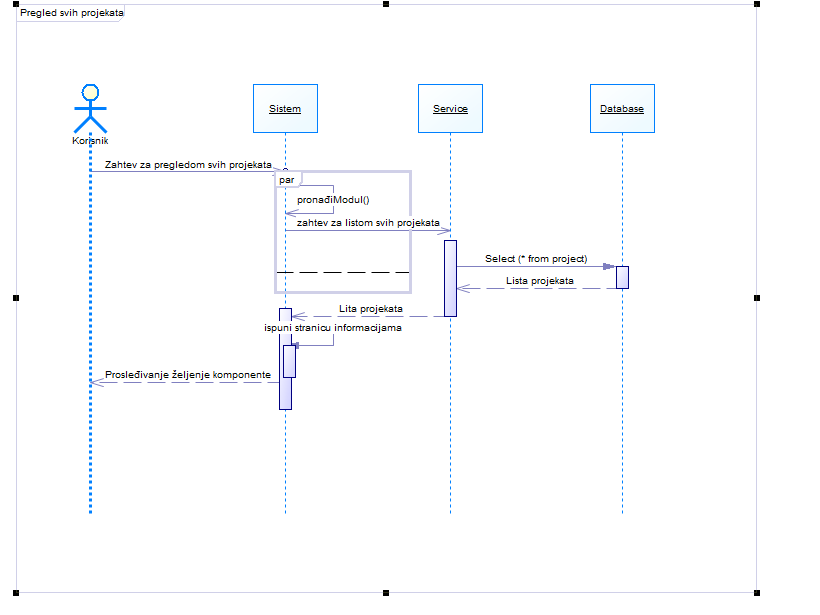
***Kreiranje naloga/registracija***



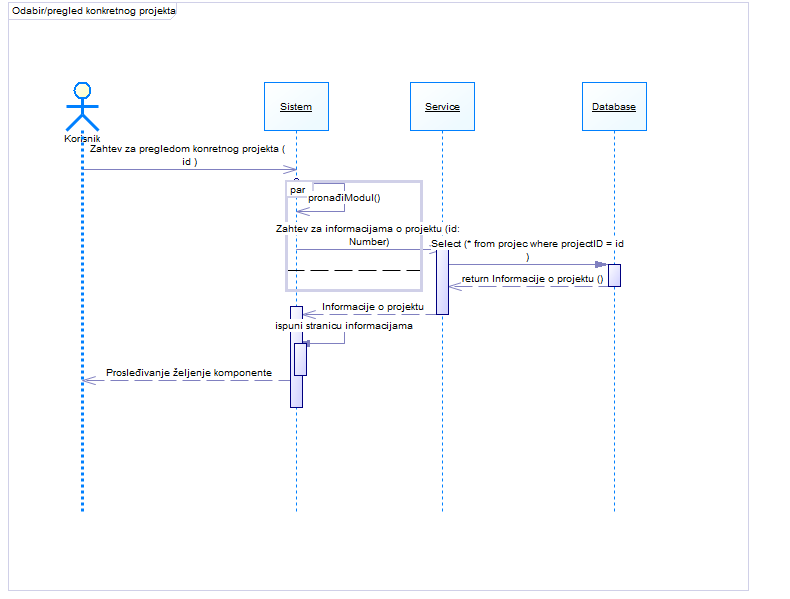
***Prijavljivanje na nalog***



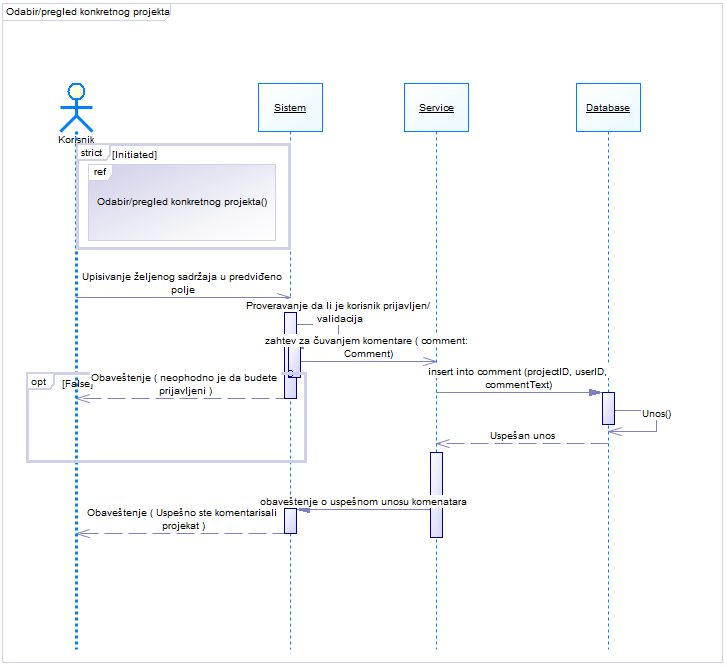
***Pregled svih projekata/problema***



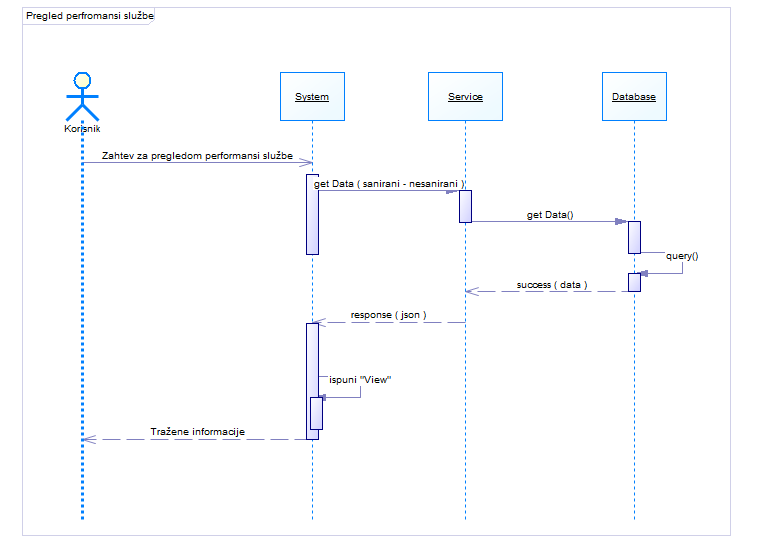
***Odabir konkretnog projekta/problema***



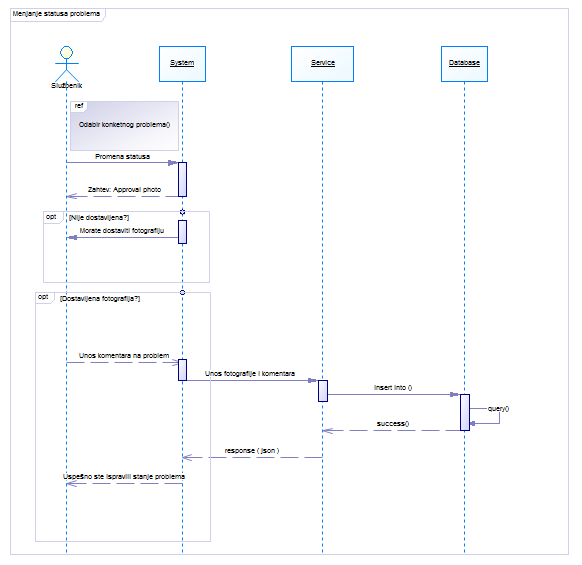
***Komentarisanje konkretnog projekta/problema***



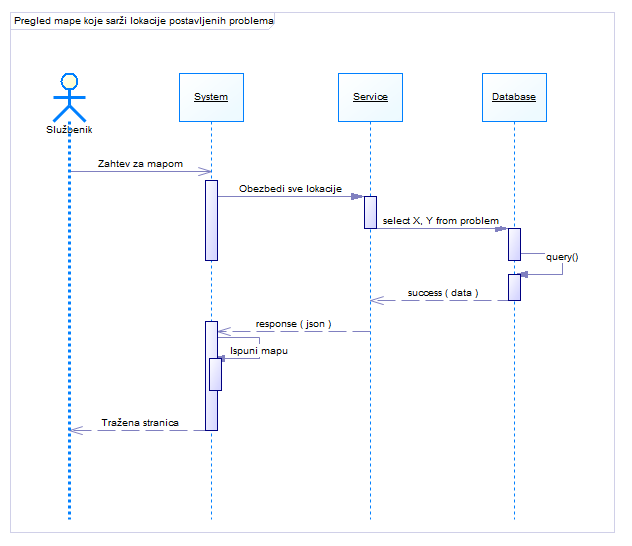
***Pregled performansi službe***



***Menjanje statusa problema***



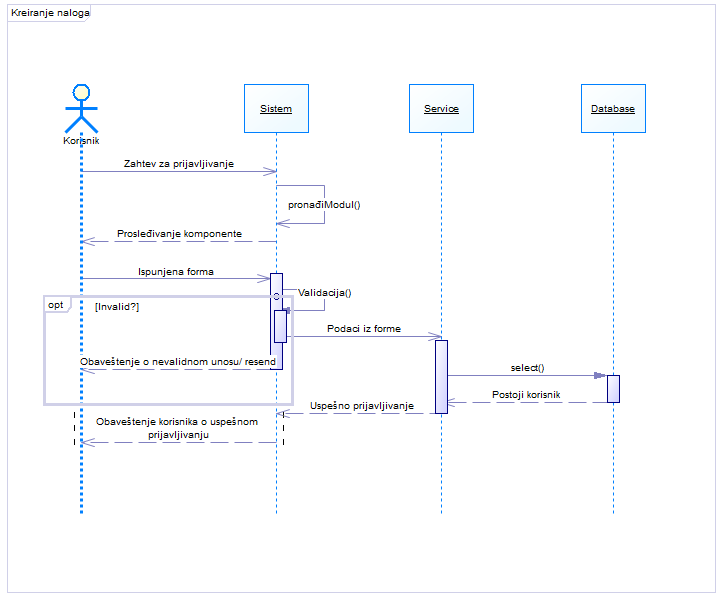
***Pregled mape sa lokacijama postavljenih problema***



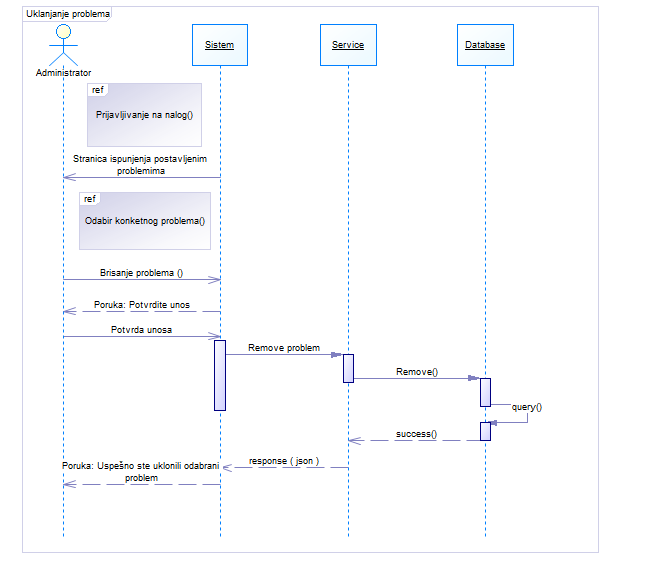
### 6.5.3. Administrator

U nastavku će biti prikazani dijagrami sekvenci koji obuhvataju sve slučajeve korišćenja za user role- “Administrator”.

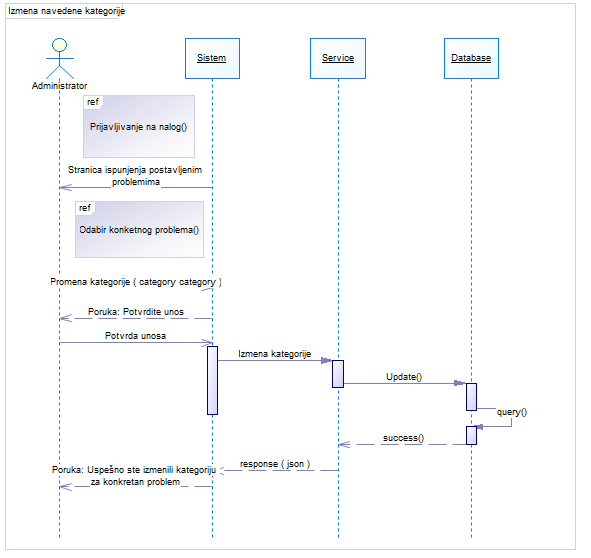
***Prijavljivanje na nalog***



***Uklanjanje problema***



***Izmena navedene kategorije određenog problema***



***Validacija/verifikacija naloga***

