

Oblikovanje programske potpore

Ak. god. 2009/2010

Policijski informacijski sustav

Dokumentacija, Rev. 3



Grupa: *gangsteri*

Voditelj: *Petra Zadro*

Datum predaje: *11. siječnja 2010.*

Nastavnik: *Alan Jović*

Popis članova grupe i zaduženja

Petra Zadro - use case, sekvencijski i dijagrami stanja, koordinacija rada grupe, razvoj i opis baze, upiti baze, korisničke upute, pregled dokumentacije

Marin Biberović - opis projektnog zadatka, dijagrami komponenata, objekata i razmještaja, upiti, opis koda glavne funkcionalnosti

Andrija Čajić - obrasci uporabe, pomoć oko programskog dijela, testiranje

Martina Jakovina - use case, sekvencijski i dijagrami stanja, opis arhitekture, upiti, korisničke upute, rad na class dijagramu

Mirko Jambrošić - programski dio (povezivanje s bazom, programsko sučelje ...), prijedlog i opis rješenja projektnog zadatka, class dijagrami

Ante Kegalj - plan rada, opis detalja oblikovanja, pomoć oko programskog dijela, testiranje

Martina Kocet - kostur funkcionalnih i nefunkcionalnih zahtjeva, razvoj i upiti baze, dijagram aktivnosti i kolaboracijski dijagram, upute za instalaciju

Filip Lasić – obrasci uporabe, istraživanje za dokumentaciju, popunjavanje baze, korišteni alati i tehnologije

Sadržaj:

1. DNEVNIK PROMJENA DOKUMENTACIJE	4
2. OPIS PROJEKTOG ZADATKA.....	6
3. RJEČNIK POJMOVA.....	10
4. FUNKCIONALNI ZAHTJEVI	11
5. NEFUNKCIONALNI ZAHTJEVI	20
6. ARHITEKTURA I DIZAJN SUSTAVA.....	21
6.1. Svrha i detaljniji prikaz rada.....	21
6.2. Opći prioriteti i skica sustava	25
6.3. Temeljna pitanja u oblikovanju	30
6.4. Detalji oblikovanja, algoritmi i strukture podataka	34
7. POVIJEST RADA I TRENUTNI STATUS IMPLEMENTACIJE	36
8. IMPLEMENTACIJA I KORISNIČKO SUČELJE	39
8.1. Dijagram razmještaja.....	39
8.2. Korištene tehnologije i alati	40
8.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava.....	40
8.4. Testiranje programskog rješenja	43
8.5. Upute za instalaciju	52
8.6. Korisničke upute.....	55
8.6.1. Čitanje primljenih poruka.....	56
8.6.2. Nova poruka	56
8.6.3. Zaduženja	57
8.6.4. Zaključavanje, zatvaranje i vidljivost zadatka.....	58
8.6.5. Novi zadatak.....	58
8.6.6. Digitalizacija.....	59
9. ZAKLJUČCI I SMJERNICE ZA DALJNI RAD	61
10. REFERENCE	62
DODATAK A: DNEVNIK SASTAJANJA	63

1. Dnevnik promjena dokumentacije

Rev.	Opis promjene/dodatka	Autor(i)	Datum
0.1	Napravljen predložak.	Zadro	23.09.2009.
0.2	Dodan opis projektnog zadatka.	Biberović	24.09.2009.
0.3	Dodan dio funkcionalnih i nefunkcionalnih zahtjeva.	Kocet	27.09.2009.
0.4	Dodani obrasci uporabe.	Čajić	29.09.2009.
0.5	Dodan plan rada.	Kegalj	30.09.2009.
0.6	Use Case dijagram.	Jakovina	01.10.2009.
0.7	Sequence dijagrami.	Zadro	02.10.2009.
0.8	Ispravci.	Jambrošić	03.10.2009.
0.9	Dopunjeni preostali dijelovi prve revizije.	Kocet	04.10.2009.
1.0	Verzija samo s bitnim dijelovima za 1. ciklus.	Zadro	05.10.2009.
1.1	Ispravci prve revizije	Biberović	01.11.2009.
1.2	Class dijagrami	Jambrošić	10.11.2009.
1.3	Opis modela baze	Zadro	13.11.2009.
1.4	Dijagrami aktivnosti i komunikacije	Kocet	14.11.2009.
1.5	Dijagram stanja, izbor arhitekture	Jakovina	14.11.2009.
1.6	Nadopunjen plan rada	Kegalj	14.11.2009.
1.7	Komponentni i dijagram objekata	Biberović	15.11.2009.
1.8	Slaganje 6.poglavlja u cjelinu	Čajić	15.11.2009.
1.9	Dopunjeni preostali dijelovi druge revizije.	Zadro	16.11.2009.
2.0	Verzija dokumentacije za 2. ciklus.	Zadro	17.11.2009.
2.1	Ispravci druge revizije	Zadro	10.12.2009.
2.2	Upute za korištenje	Jakovina	05.01.2010.
2.3	Novi class dijagrami	Zadro	06.01.2010.
2.4	Korištene tehnologije i alati	Lasić	07.01.2010.
2.5	Upute za instalaciju	Kocet	08.01.2010.
2.6.	Testiranje	Čajić	08.01.2010.

2.7.	Ispravljeni neki dijagrami	Kocet	09.01.2010.
2.8	Dijagram razmještaja, glavna funkcionalnost	Biberović	10.01.2010.
2.9	Povijest rada, zaključci	Lasić	10.01.2010.
3.0	Završna verzija dokumentacije projekta	Zadro	11.01.2010.

2. Opis projektnog zadatka

Cilj ovog projekta je implementacija informacijskog sustava za policiju grada Polisa. Novi sustav mora omogućiti lakše praćenje policijskih službenika, njihovih zadataka te arhiviranje novih i postojećih informacija. Policijski službenici nisu samo policajci nego i službenici unutar policijskih postaja i drugih policijskih objekata koji također imaju svoja zaduženja. Policijska uprava sastoji se od nekoliko zgrada, nešto više policijskih postaja te velikog broja policijskih službenika. Sustav moraju moći funkcionalno koristiti svi službenici bez obzira na razinu informatičkog obrazovanja te mjesta rada. Obavezno je da sustav funkcionira na razini cijele gradske policije, a opcionalno bi moglo biti omogućeno da se koristi sa bilo koje lokacije ako je takav način rada korisniku dozvoljen od strane nadređene osobe.

Sustav mora omogućiti dodjelu zadataka službenicima i njihovu eventualnu kasniju izmjenu. Ove akcije obavljati će administrator zadužen za logistiku na zahtjev službenika koji je nadređen onoj osobi koja dobiva zadatak. Službenici moraju biti u mogućnosti u svakom trenutku pratiti promjene na zadacima koji su im dodijeljeni.

Također mora biti moguće razmjenjivanje multimedijskog sadržaja između dvoje ili više službenika s obzirom na hijerarhijski položaj unutar policijske uprave. Sadržaje mogu razmjenjivati službenici iste razine. Sadržaji mogu biti tekstualni, zvučni zapisi, slike i video zapisi.

Sustav je interni i ne smije biti otvoren prema vanjskoj mreži jer su svi sadržaji unutar samog sustava povjerljivi. Potrebno je i arhivirati sve sadržaje koji se nađu na sustavu te omogućiti unos postojećeg pisanog sadržaja u arhivu kako bi se lakše pregledavao u digitalnom obliku.

Policijska uprava ima četiri hijerarhijske razine zaposlenika od kojih samo najviša razina ima jednake mogućnosti manipulacije sustavom kao i administrator zadužen za logistiku. Svaki službenik u sustavu ima vlastiti kompletan zapis te je dužan koristiti sustav u svakodnevnom radu, u njega unositi promjene sukladno sa pravilima i komunicirati sa drugim zaposlenicima prema potrebi. Službenik, ovisno o

poziciji, može patrolirati određeno područje, istraživati nezakonite radnje, sastavljati izvješća, prisustvovati sastancima vezanim uz posao te obavljati mnoge druge uobičajene policijske poslove.

Svaki posao mora imati naziv, datum početka, očekivano trajanje posla, popis osoba koje rade na tom poslu, prioritet, stupanj povjerljivosti, opis, datum završetka te popis promjena.

Rješenje za dotični problem se svodi na upravljanje bazom podataka. Informacije koje korisnik smije i može vidjeti, urediti, dodati ovise o razini hijerarhije na kojoj se nalazi. Prvo rješenje koje ćemo predstaviti je korištenje samo klijentske aplikacije koja se spaja na bazu podataka na nekom poslužiteljskom računalu. Korisnik se prijavljuje u aplikaciju koja uspostavi vezu, preko sustava za upravljanje bazom podataka (SUBP), sa samom bazom podataka. Nakon provjere autentifikacije i potvrde sa korisničkim imenom i lozinkom korisnik dalje dobiva dozvole za rad. Baza podataka mora biti u javnoj mapi na računalu radi mogućnosti pristupa istoj. Tu dolazimo do prvog problema sigurnosti. Baza podataka se može kopirati i na druge načine pokušavati ući u nju. Možda i najveća mana ovog rješenja je potreba za vezom između klijenta i same baze – SUBP-om. Instalacija SUBP-a na svako klijentsko računalo nije najzgodnije rješenje. Zbog gore navedenih razloga predlažemo sljedeće: rješenje sa *poslužiteljskom aplikacijom*.

Ideja je da se *poslužiteljska aplikacija* i sama *baza podataka* nalaze na jednom računalu, poslužitelju, koji ima pristup lokalnoj mreži. Baza podataka ne mora biti vidljiva ostalim računalima na mreži. Baza podataka pristupa, te konačno i upravlja, samo i isključivo poslužiteljska aplikacija. Klijentska aplikacija spaja se preko TCP/IP protokola, preko određenog porta, na poslužiteljsku aplikaciju. Ideja je dalje ista kao i bez poslužiteljske aplikacije. Komunikacija ide od klijenta, preko poslužitelja, koji preko SUBP-a pristupa bazi podataka. Poslužitelj vraća podatke klijentu kojeg raspoznaje preko IP adrese i već prije poznatog porta. Velika prednost ovoga je korištenje SUBP-a samo na jednom mjestu, mogućnost uvođenja zaštite informacija kod prijenosa te vrlo jednostavno arhiviranje podataka.

Komunikacija klijenta i poslužitelja:

Korisnik u sustav unosi korisničko ime i lozinku. Nakon provjere točnosti podataka od strane poslužitelja, dolazi odgovor u obliku broja značke. Korisnik potražuje tekstualne poruke od poslužitelja slanjem broja značke, poslužitelj ispunjava zahtjev te su povratna informacija sve poruka tog tipa. Ovakvo potraživanje korisnik može obaviti za tekstualne i multimedijalne poruke te vlastite zadatke koji su zadani i neizvršeni. Nadređeni zaposlenici mogu na ovakav način od poslužitelja potraživati i izvršene zadatke od stane podređenih te ih onda zatvoriti.

Prilikom slanja poruke, korisnik zahtjeva od sustava popis osoba kojima može poslati poruku (svi policijski službenici iste razine) te nakon toga šalje poruku koja sadrži njegov broj značke, sadržaj poruke te popis brojeva znački svih primatelja poruke. Ovakav način slanja poruka vrijedi za tekstualne i multimedijske poruke s razlikom da poslužitelj nakon primitka naznake da se radi o multimedijskoj poruci čeka isključivo na sadržaj te poruke, odnosno multimediju, a za vrijeme čekanja ne obrađuje daljnje korisničke zahtjeve nego se oni pamte i stavljaju na čekanje. Nakon što poslužitelj prihvati sadržaj (multimediju) pohranjuje ga na sekundarnu memoriju pod slučajno odabranim imenom te u bazu podataka upisuje samo to ime, ne i sami sadržaj.

Nakon što je neki zadatak izvršen, korisnik ga mora zaključati. Zaključavanje zadatka obavlja se tako da korisnik pošalje poslužitelju broj značke i broj zadataka koji je potrebno zaključati. Poslužitelj obavlja traženu zadaću te šalje povratnu informaciju o obavljenom.

Arhiviranje pisanog sadržaja obavlja se slanjem potrebnih podataka poslužitelju koji taj sadržaj pohranjuje u bazu podataka.

Potencijalno ovaj će informacijski sustav znatno skratiti vrijeme koje policijski službenik troši prilikom traženja informacija, unosa informacija i dijeljenja informacija između ostalih zaposlenika. Također će unaprijediti poslovanje policijske uprave te uvesti jednostavnost u nekad zamorne i dugotrajne procedure koje su vezane za papirologiju.

Kupci ovog proizvoda ne moraju biti samo policijske uprave, već je sustav pogodan za implementaciju na bilo kojem mjestu gdje se posao vrši uzimajući u obzir hijerarhijsku strukturu poduzeća. Vrlo je lako izravno primjenjiv uz sitne preinake u vojne svrhe te bilo koji drugu strogo hijerarhijsku strukturu koja obavlja srodan posao.

3. Rječnik pojmova

Delphi : Programski jezik kojemu je podloga objektni Pascal.

FTP: Protokol za prijenos datoteka.

IP adresa: Jedinstvena bročana oznaka računala na Internetu.

SQL: Programski jezik za upravljanje bazom podataka.

SUBP: Sustav upravljanja bazom podataka.

TCP/IP: Oznaka grupe protokola koja omogućuje komunikaciju preko raznih međusobno povezanih mreža, na njemu zasniva i globalna mreža Internet.

Sekundarna memorija: Memorija koja nije izravno dostupna procesoru računala, vrlo je spora i koristi se za trajnu pohranu podataka.

Poslužitelj: Namjensko računalo koje šalje i/ili prima podatke od jednog ili više klijenata.

UML (Unified Modeling Language): standardizirani jezik za slikovno predstavljanje i modeliranje objekata.

RAID: "*Redundant Array of Inexpensive (Independent) Disks*". RAID je, dakle, metoda kombiniranja više fizičkih diskova u jednu logičku cjelinu

RAID 0: (*striped set* ili *striped volume*) dijeli podatke ravnomjerno preko dva ili više diskova.

4. Funkcionalni zahtjevi

Postoji nekoliko funkcija koje sustav treba ostvariti i time udovoljiti zahtjevima policijskih službenika. Ponajprije treba omogućiti pregled zadataka koji su im dodijeljeni, uvid u promjene na zadacima i unos promjena. Ukratko, prvi zahtjev bi bio lakše praćenje i zadavanje zadataka. Sljedeći je ostvarena komunikacija među službenicima putem poruka ili razmjenom multimedijskog sadržaja. I konačno, mogućnost arhiviranja različitih tipova informacija, odnosno svih obavljenih poslova i razmijenjenih multimedijskih sadržaja.

Osobe koje imaju interes u ovom sustavu, odnosno dionici sustava su:

- Policijska uprava
- Policijski službenici
- Administrator

Aktori koji izravno komuniciraju sa sustavom mogu imati ulogu inicijatora ili sudionika. Inicijatori pokreću procese u sustavu dok sudionici obavljaju određene poslove.

Aktori, njihove uloge i funkcionalni zahtjevi su sljedeći:

- Policijski službenici, inicijatori:
 - Zbog hijerarhijskog sustava postoje podređeni i nadređeni službenici.
 - Podređeni postavljaju upite o dodijeljenim zadacima i razmjenu poruka kroz korisničko sučelje sustava.
 - Nadređeni dodatno dodjeljuju zadatke svojim podređenima i zatvaraju ih ukoliko su obavljeni.
- Administrator, inicijator:
 - Dodjeljuje poslove na zapovijed policijskog službenika nadređenog službeniku kojem je zadan zadatak.
 - Najviša razina hijerarhije (šef policije) ima ujedno i ovlasti kakve ima administrator.

- Tajnik, inicijator:
 - Bavi se digitalizacijom, odnosno unošenjem starih papirnatih dokumenata u arhivu.
- Baza podataka, sudionik:
 - Sprema podatke o poslovima, službenicima, multimedijским sadržajima.
- Poslužitelj (server), sudionik:
 - Pristupa bazi podataka na zahtjev korisnika i osigurava hijerarhijski pristup podacima.

Opis obrazaca uporabe:

- UC1 – *Spajanje na sustav (Login)*:
 - **Glavni sudionik**: policijski službenik
 - **Cilj**: spajanje na sustav
 - **Sudionici**: poslužitelj, baza podataka
 - **Preduvjeti**: dostupnost poslužitelja
 - **Rezultat**: korisnik je spojen na sustav, i poslužitelj čeka daljnje naredbe
 - **Željeni scenarij**:
 1. korisnik unese korisničko ime i lozinku
 2. poslužitelj preko baze podataka zatraži listu korisničkih imena i pripadnih lozinki
 3. provjerava odgovaraju li korisničko ime i lozinka te, ukoliko odgovaraju, korisnika spaja na sustav
 - **Mogući drugi scenariji**:
 3. ako nema dotične kombinacije korisničkog imena i lozinke na listi, korisniku nije omogućen pristup sustavu
- UC2 - *Slanje sadržaja*:
 - **Glavni sudionik**: policijski službenik
 - **Cilj**: razmjena informacija
 - **Sudionici**: poslužitelj, baza podataka
 - **Preduvjeti**: korisnik spojen na sustav
 - **Rezultat**: informacije poslane drugim službenicima

- **Željeni scenarij:**
 - 1. korisnik šalje zahtjev poslužitelju za slanje poruka
 - 2. poslužitelj preko baze pronalazi i izlista popis policijskih službenika kojima dotični korisnik može poslati poruku
 - 3. korisnik koji šalje poruku sastavlja poruku i označava službenike sa liste kojima je želi poslati
 - 4. baza pohranjuje poruku za one kojima je namijenjena
 - **Mogući drugi scenariji:**
 - 4. poruka pogrešno primljena
- **UC3 – Čitanje primljenog sadržaja:**
- **Glavni sudionik:** policijski službenik
 - **Cilj:** čitanje poruka primljenih od drugih službenika
 - **Sudionici:** poslužitelj, baza podataka
 - **Preduvjeti:** korisnik spojen na sustav i u bazi podataka postoje pohranjene poruke poslane njemu
 - **Rezultat:** korisnik ima pristup primljenim porukama
 - **Željeni scenarij:**
 - 1. korisnik šalje zahtjev poslužitelju da dohvati primljene poruke
 - 2. poslužitelj preko baze dohvaća poruke te ih prikazuje korisniku koji ih može čitati
 - **Mogući drugi scenariji:**
 - 2. korisnik nema primljenih poruka
- **UC4 – Zadavanje zadatka:**
- **Glavni sudionik:** nadređeni policijski službenik
 - **Cilj:** određivanje zadatka i zadavanje zadatka podređenim policijskim službenicima
 - **Sudionici:** administrator, poslužitelj, baza podataka
 - **Preduvjeti:** korisnik spojen na sustav i nadređeni je policijskim službenicima kojima želi zadati zadatak
 - **Rezultat:** zadatak dodijeljen službenicima koji na njemu trebaju raditi
 - **Željeni scenarij:**
 - 1. korisnik šalje zahtjev poslužitelju za stvaranje novog zadatka

2. poslužitelj preko baze podataka pronalazi i izlistava popis službenika koji su podređeni korisniku

3. korisnik sastavlja novi zadatak i odabire službenike kojima ga želi zadati

4. zadatak se preko poslužitelja sprema u bazu za službenike kojima je dodijeljen, ali ga oni mogu vidjeti tek kada ga administrator odobri

▪ UC5 – *Čitanje i uređivanje promjena u stanju zadatka:*

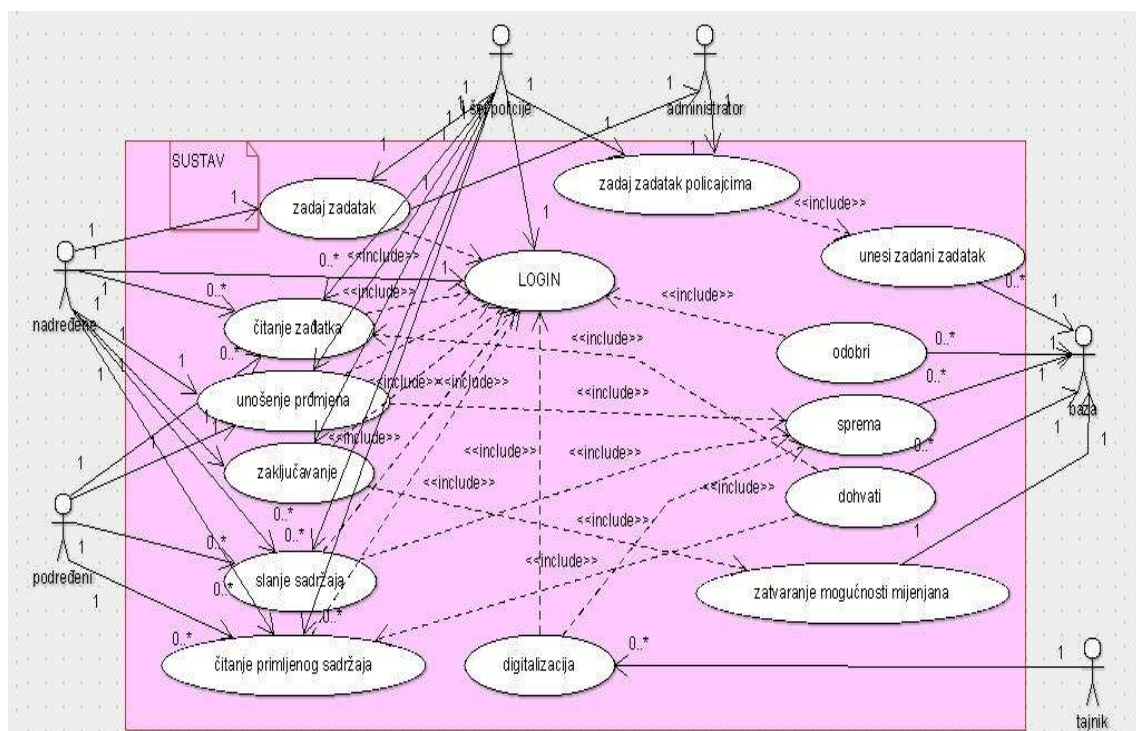
- **Glavni sudionik:** policijski službenik
- **Cilj:** pregled nad zadanim zadacima te unošenje promjena u razvoju zadatka
- **Sudionici:** poslužitelj, baza podataka
- **Preduvjeti:** korisnik spojen na sustav i postoje zadaci koji su mu dodijeljeni
- **Rezultat:** promjena statusa zadatka kako bi se mogao pratiti razvoj zadatka
- **Željeni scenarij:**
 1. korisnik šalje zahtjev za pregledavanje svojih zadataka
 2. nakon otvaranja zadatka nudi mu se mogućnost unošenja promjena u razvoju zadatka
 3. ako želi, korisnik unosi promjene koje se zatim preko poslužitelja spremaju u bazu

▪ UC6 – *Zaključavanje zadatka:*

- **Glavni sudionik:** nadređeni policijski službenik
- **Cilj:** završavanje rada na zadatku, zaključavanje informacija o njemu
- **Sudionici:** poslužitelj, baza podataka
- **Preduvjeti:** korisnik spojen na sustav i glavni je na zadatku
- **Rezultat:** zadatak je gotov, zaključan i na njemu se više ne mogu vršiti nikakve promjene i preinake
- **Željeni scenarij:**
 1. policijski službenici mijenjaju status zadatka u zaključani
 2. nadređeni šalje zahtjev za zatvaranjem zadatka

3. poslužitelj u bazi zatvara mogućnost unošenja promjena na zadatku i dobiva oznaku gotovog zadatka

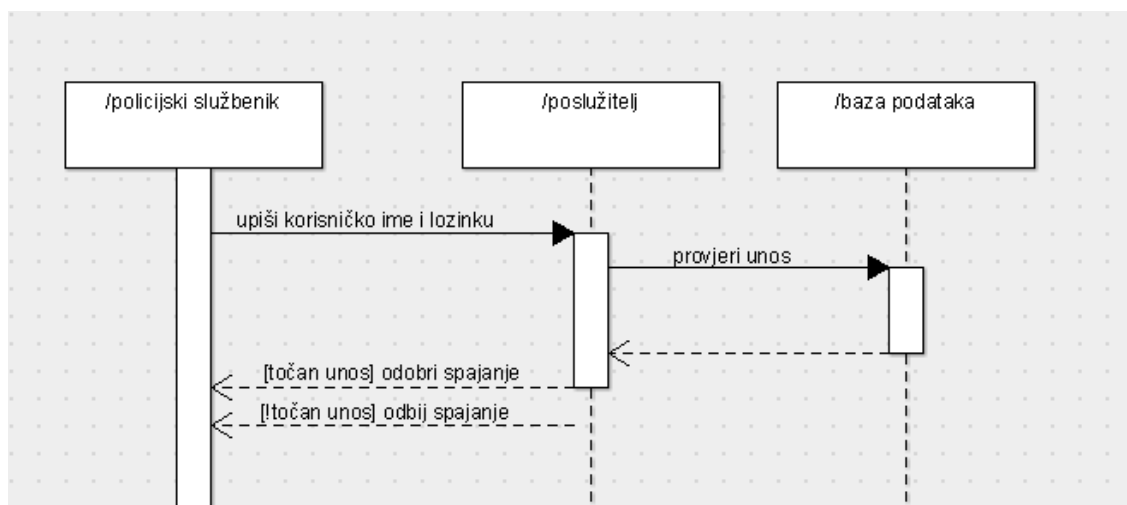
- UC7 – *Digitalizacija pisanog sadržaja (Arhiviranje)*:
 - **Glavni sudionik:** tajnik
 - **Cilj:** spremanje starih podataka radi očuvanja i lakšeg pristupa u digitalni oblik
 - **Sudionici:** poslužitelj, baza podataka
 - **Preduvjeti:** korisnik spojen na sustav
 - **Rezultat:** svi stari dokumenti zapisani u digitalnom obliku i pohranjeni
 - **Željeni scenarij:**
 1. tajnik unosi podatke sa starih dokumenata u sustav
 2. podaci se pohranjuju u bazu podataka



Slika 4.1. Obrazac uporabe za UC1 do UC7.

Sekvencijski dijagrami:**Slučaj korištenja UC1:**

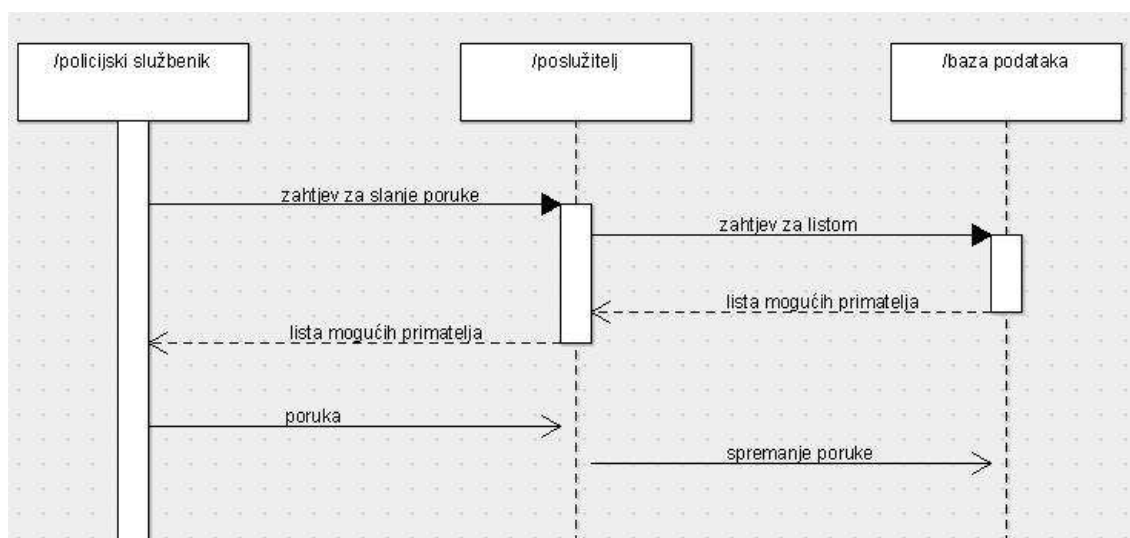
Korisnik se želi spojiti na sustav, pa upisuje svoje korisničko ime i lozinku. Poslužitelj preko baze podataka provjerava kombinaciju korisničkog imena i lozinke. U slučaju da je unos odgovarajući, korisnik se spaja na sustav. Za krivi unos korisniku se odbija zahtjev za spajanjem na sustav. Odgovarajući dijagram je prikazan na *Slici 4.2*.



Slika 4.2. Sekvencijski dijagram za UC1 (Spajanje na sustav)

Slučaj korištenja UC2:

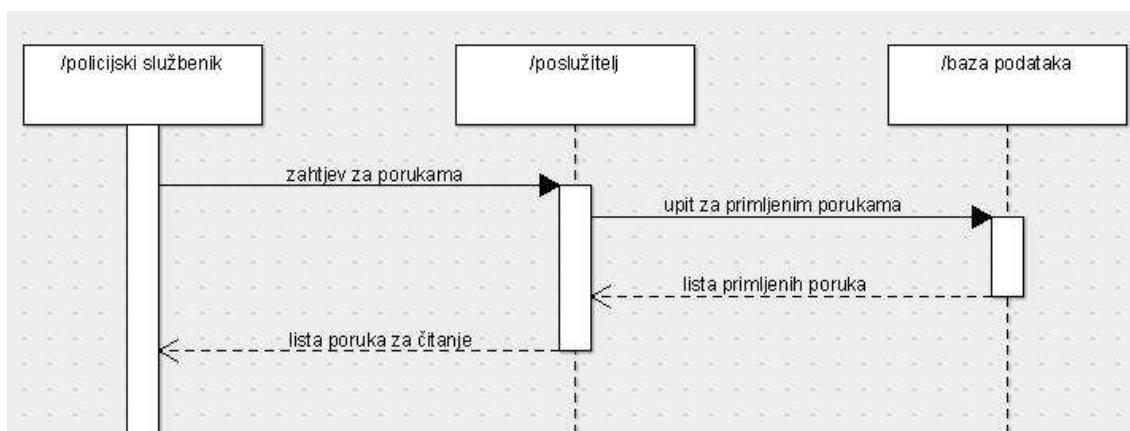
Korisnik šalje poslužitelju zahtjev za slanje multimedijske poruke. Poslužitelj od baze podataka traži listu policijskih službenika kojima korisnik može slati poruku. Korisnik sastavlja poruku i označava policijske službenike, s dobivene liste, kojima želi poslati poruku. Poruka se sprema u bazu podataka. Odgovarajući dijagram je prikazan na *Slici 4.3*.



Slika 4.3. Sekvencijski dijagram za UC2 (Slanje sadržaja)

Slučaj korištenja UC3:

Korisnik šalje poslužitelju zahtjev za pregledom primljenih poruka i ostalih sadržaja. Poslužitelj od baze podataka dohvaća listu primljenih poruka te ih prosljeđuje korisniku na čitanje. Odgovarajući dijagram je na *Slici 4.4*.

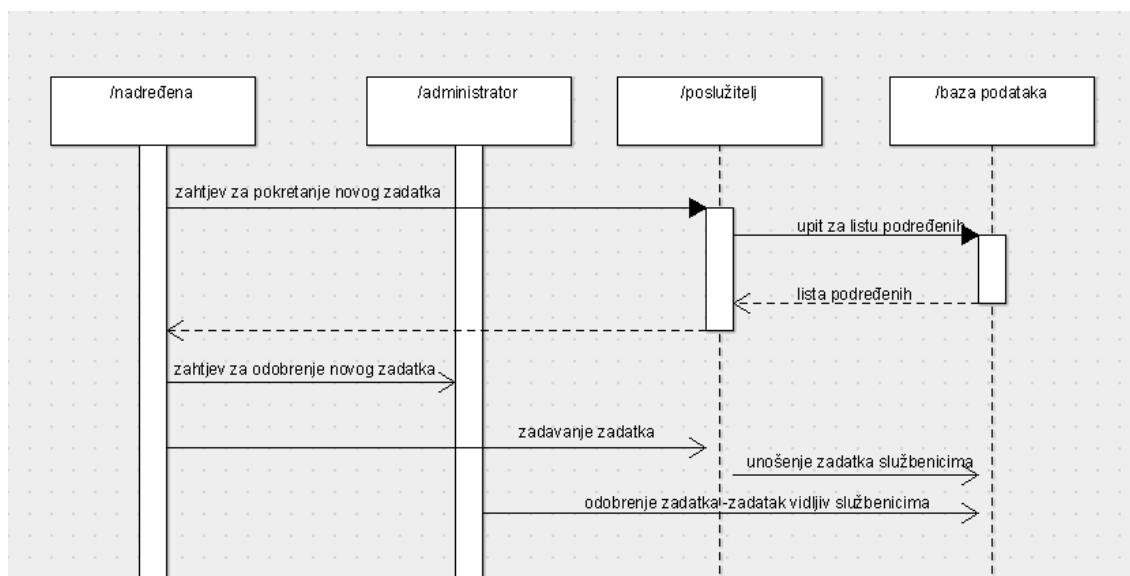


Slika 4.4. Sekvencijski dijagram za UC3 (Čitanje primljenih sadržaja)

Slučaj korištenja UC4:

Korisnik šalje poslužitelju zahtjev za stvaranje novog zadatka. Poslužitelj od baze podataka dohvaća listu službenika koji su podređeni korisniku koji je podnio zahtjev. Korisnik stvara novi zadatak, odabire službenike sa liste, kojima želi zadatak dodijeliti, te šalje podatke preko poslužitelja bazi podataka. Da bi novi zadatak postao

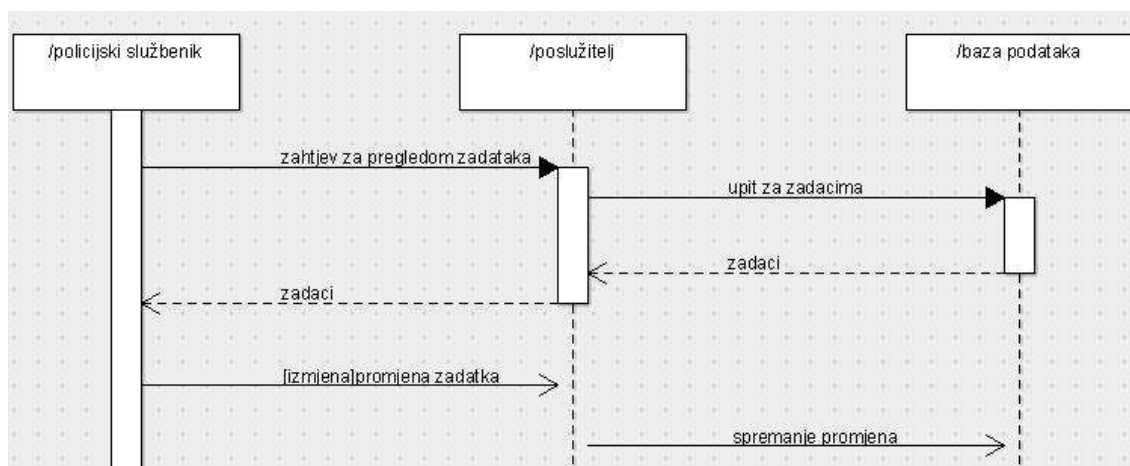
aktualan i vidljiv policijskim službenicima koji su za njega zaduženi, administrator ga mora odobriti. Odgovarajući dijagram je na *Slici 4.5*.



Slika 4.5. Sekvencijski dijagram za UC4 (Zadavanje zadatka)

Slučaj korištenja UC5:

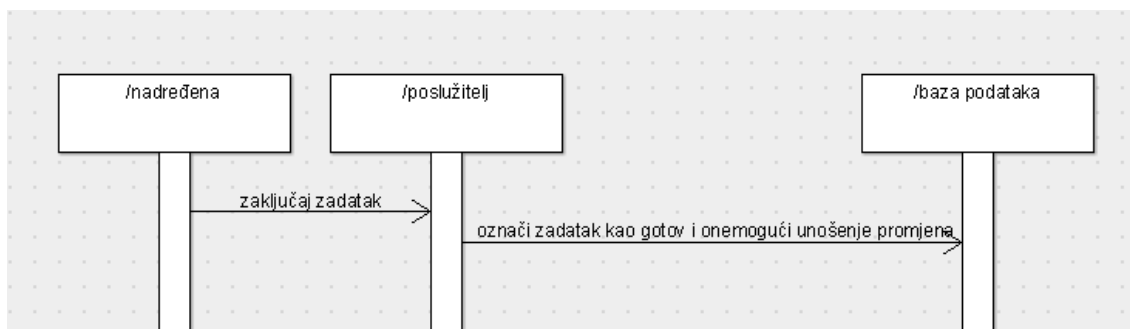
Korisnik šalje poslužitelju zahtjev za pregledavanje svojih zadataka. Poslužitelj od baze podataka dohvaća listu zadataka na kojima korisnik radi te ju prosljeđuje korisniku na čitanje. Nakon otvaranja zadatka korisniku se osim čitanja, nudi mogućnosti unošenja promjena u razvoju zadatka. Promjene se spremaju u bazu podataka. Odgovarajući dijagram je na *Slici 4.6*.



Slika 4.6. Sekvencijski dijagram za UC5 (Čitanje i uređivanje zadatka)

Slučaj korištenja UC6:

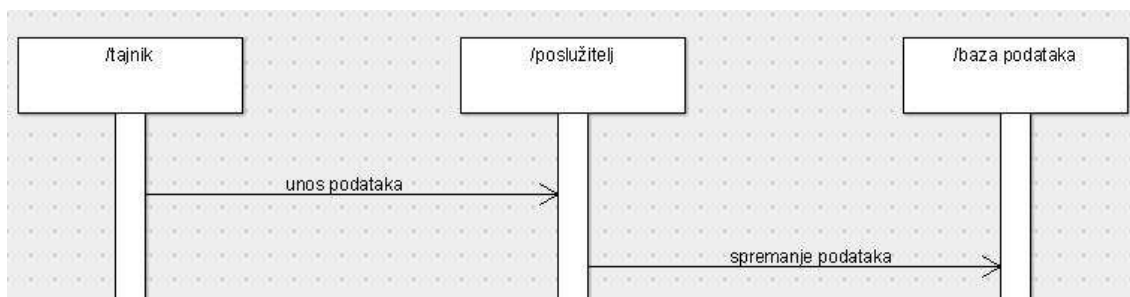
Kada policijski službenik označi da je zadatak zaključan, nadređena osoba zadužena za taj zadatak šalje poslužitelju zahtjev za zatvaranjem. Poslužitelj preko baze podataka označava zadatak kao gotov i onemogućava daljnje izmjene na njemu. Odgovarajući dijagram je na *Slici 4.7*.



Slika 4.7. Sekvencijski dijagram za UC6 (Zaključavanje zadatka)

Slučaj korištenja UC7:

Tajnik unosi podatke sa starih dokumenata u sustav. Podaci se preko poslužitelja koji pristupa bazi podataka spremaju. Odgovarajući dijagram je prikazan na *Slici 4.8*.



Slika 4.8. Sekvencijski dijagram za UC7 (Digitalizacija)

5. Nefunkcionalni zahtjevi

Zahtjevi kvalitete:

- Sustav bi trebao biti napravljen da ga svi znaju koristiti bez dodatnih uputa.
- Nepravilne akcije policijskih službenika unutar korisničkog sučelja ne smiju narušiti funkcionalnost sustava.
- Sustav bi trebao jamčiti pouzdanost i točnost informacija.
- Odgovor na korisnikov upit u prosjeku ne bi trebao trajati duže od 1 sekunde.
- Moguće zagušenje veze kod prijenosa "velikih" multimedijских poruka
- Sustav bi trebao biti napravljen tako da se daljnje nadogradnje čine bez narušavanja ostatka sustava i sustava u cjelini.

Ograničenja:

- Nadogradnja sustava nije moguća bez izvornog koda i programskog jezika u kojem je aplikacija pisana (Delphi).
- Unos policijskih službenika nije moguć bez izravnog pristupa bazi podataka, odnosno ne mogu se unositi putem aplikacije.
- Moguće je slanje samo jednog multimedijского sadržaja u jednoj poruci.
- Veličina multimedijского sadržaja ograničena je na 1MB.
- Nije moguće pokrenuti dvije aplikacije, različitih korisnika s iste IP adrese.
- Potrebno je podešavanje routera u slučaju da se korisnik nalazi "iza" njega (Internet - router - korisnik ... router prema Internetu ima IP adresu X, dok prema korisniku ima IP adresu Y. Tu nastaje problem te su potrebna podešavanja routera da proslijedi tu poruku sa X na Y).

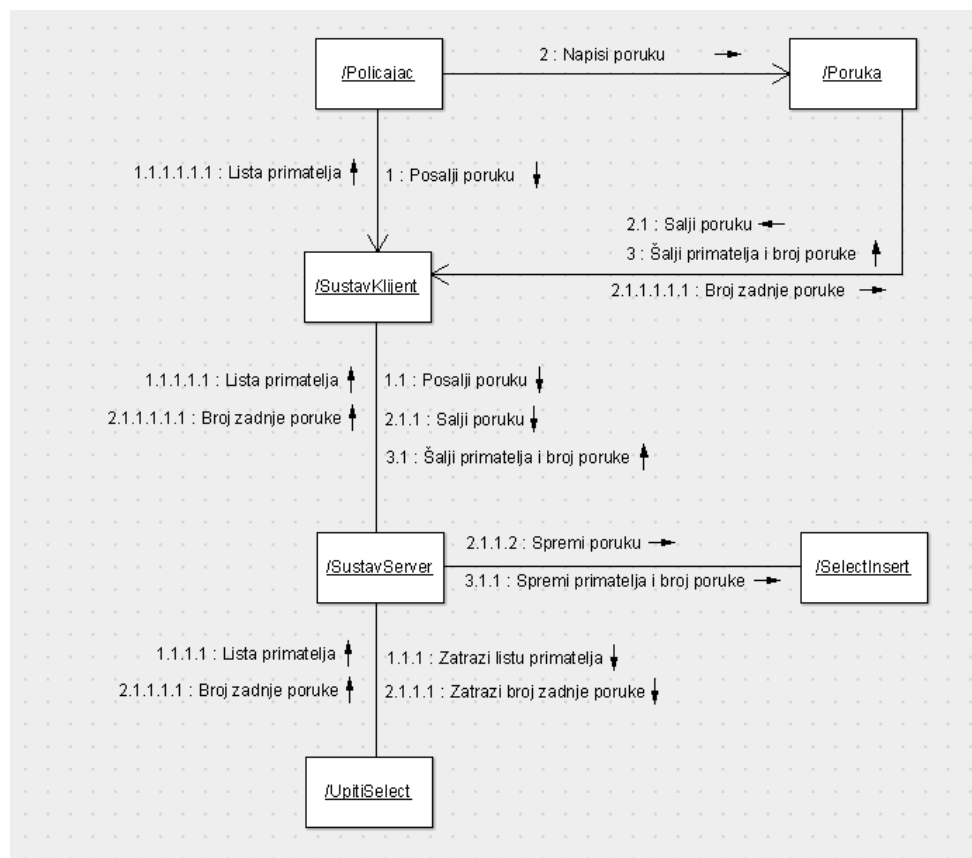
6. Arhitektura i dizajn sustava

6.1. Svrha i detaljniji prikaz rada

Ovaj dokument opisuje policijski informacijski sustav koji omogućuje komunikaciju između policijskih službenika. Svrha sustava opisana je zahtjevima koji su već prethodno navedeni, a neki od njih prikazat će se dijagramima radi boljeg razumijevanja njihove implementacije.

- Slanje poruka

Jedan od zahtjeva koje sustav treba ostvariti je slanje poruka i/ili multimedijских sadržaja kao što je navedeno u poglavlju 4. Funkcionalni zahtjevi : UC2-Slanje sadržaja. Kako se odvija slanje poruka prikazuje komunikacijski (kolaboracijski) dijagram na *Slici 6.1*.



Slika 6.1: Komunikacijski (kolaboracijski) dijagram – Slanje poruke

Objekti koji sudjeluju u komunikaciji su Policajac, Poruka, SustavServer, SustavKlijent, UpitiInsert i UpitiSelect.

Prvi korak u slanju poruke je dobiti listu policijskih službenika kojima se ona može poslati. Komunikacija započinje slanjem zahtjeva objekta Policajac za slanjem poruke prema SustavServeru preko SustavKlijenta. Zatim SustavServer komunicira sa objektom UpitiSelect kako bi kao povratnu poruku dobio listu primatelja.

Proces dobivanja liste potencijalnih primatelja označen je u dijagramu od 1:Pošalji poruku do 1.1.1.1.1:Lista primatelja.

Kada dobije listu primatelja, policijski službenik označava kome želi poslati poruku i napiše ju. To je komunikacija Policajac – Poruka (u dijagramu označeno pod 2: Napiši poruku). Kada je poruka napisana šalje se preko SustavKlijenta do SustavServera.

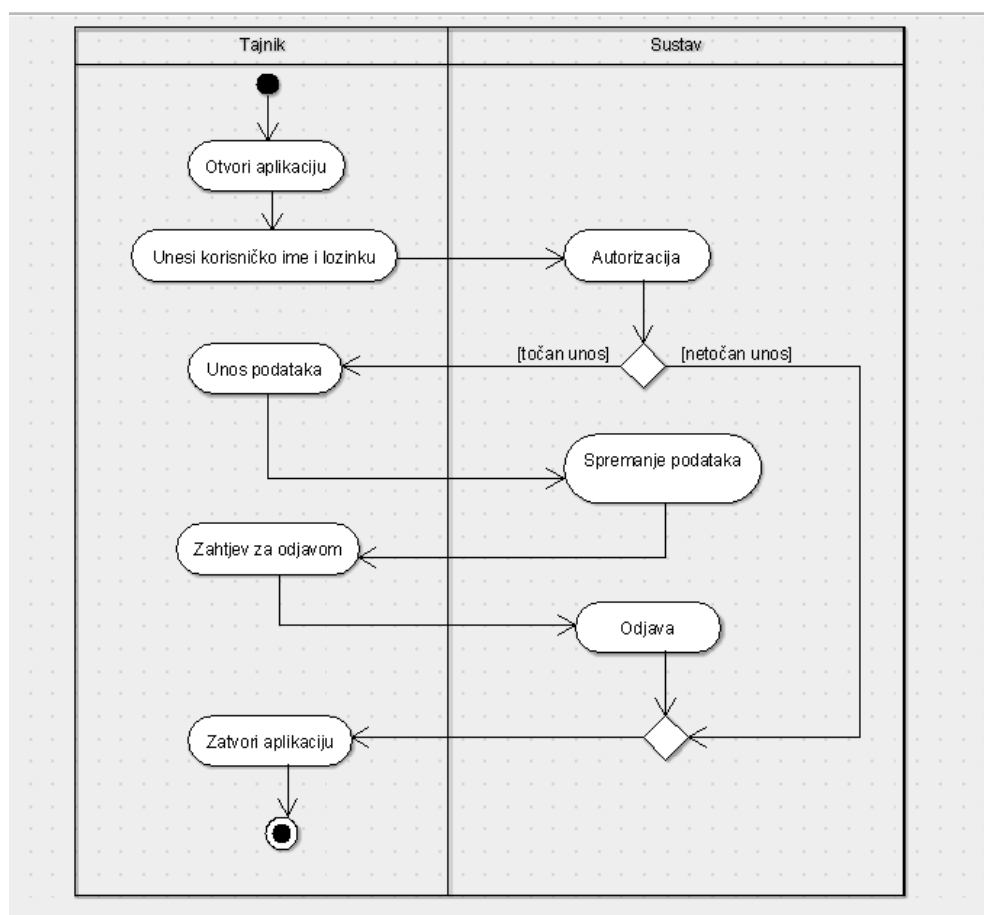
Nakon dolaska do SustavServera šalje se zahtjev za brojem zadnje poruke u bazi (komunikacija prema UpitiSelect 2.1.1.1:Zatraži broj zadnje poruke). Zatim se poruka sprema u bazu (2.1.1.2:Spremi poruku prema UpitiInsert), a taj broj pod kojim je poruka spremljena šalje se natrag prema objektu Poruka preko veze SustavServer - SustavKlijent.

Slijedeći i završni korak je bilježenje primatelja kojima je poruka namijenjena. To se radi tako da se u bazu spremi zajedno oznaka primatelja i broj napisane poruke. Dakle, prvo slijedi komunikacija do SustavKlijenta pa do SustavServera (stavke 3. i 3.1. na dijagramu). Zatim se odvija spremanje u bazu (3.1.1:Spremi primatelja i broj poruke). Ovaj postupak se odvija pojedinačno za svakog primatelja, što znači da će se izvesti onoliko puta koliko primatelja ima.

Kada se spremu svi primatelji, završen je proces slanja poruke.

- Digitalizacija

Digitalizacija je zahtjev opisan u poglavlju 4.Funkcionalni zahtjevi : UC7-Digitalizacija pisanog sadržaja, a ovdje je pojašnjena dijagramom aktivnosti na *Slici 6.2. zajedno sa prijavom na sustav.*



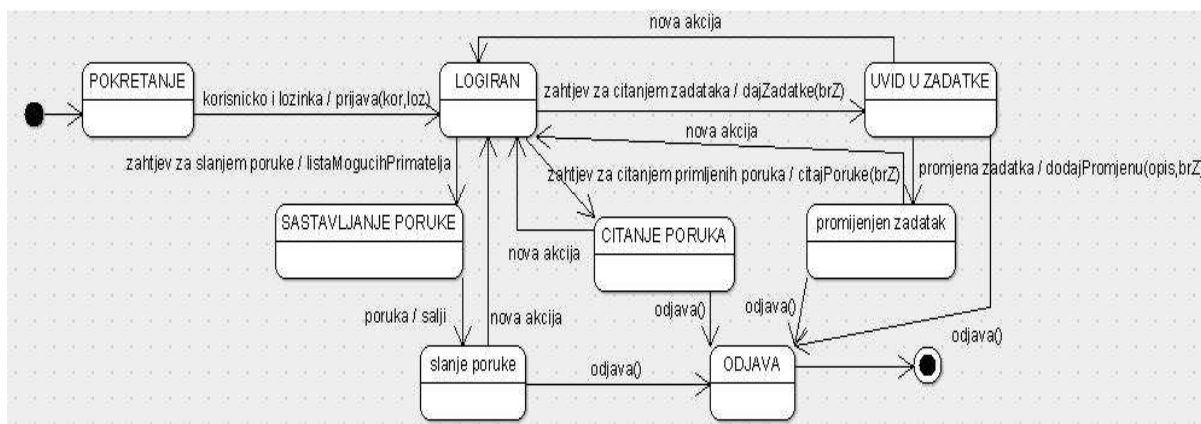
Slika 6.2: Dijagram aktivnosti – Digitalizacija sadržaja

Digitalizaciju obavlja tajnik koji se prvotno mora spojiti na sustav. Unosi svoje korisničko ime i lozinku, a sustav provjerava točnost podataka. Moguća su dva ishoda autorizacije, što je na dijagramu označeno točkom odlučivanja.

Ako je unos netočan, tajnik zatvara aplikaciju i time označava kraj rada. Ako je unos točan, tajnik zapisuje podatke koje sustav sprema. Kada završi s upisivanjem podataka, tajnik traži zahtjev za odjavom. Sustav ga odjavljuje, tajnik zatvara aplikaciju i završava s radom.

- Moguće akcije podređenog

Gledajući cjelokupnu sliku use case dijagrama u poglavlju 4. Funkcionalni zahtjevi, vide se akcije koje može, i za koje ima dozvolu, uloga podređenog u sustavu. Ovdje je to prikazano dijagramom stanja.



Slika 6.3: Dijagram stanja - Podređeni

Dijagram stanja modelira ponašanje sustava upravljano događajima. Nakon što se spoji na sustav, korisnik u ulozu podređenog može slati i čitati poruke te čitati i uređivati zadatke koji su mu dodijeljeni.

Po pokretanju aplikacije, korisnik unosi korisničko ime i lozinku. Time je prijavljen na sustav, te može slati i čitati poruke ili raditi na zadacima koji su mu dodijeljeni.

Ako želi poslati poruku, zahtjeva listu mogućih primatelja, sastavlja i šalje poruku odabranima. Ako poruku želi čitati, zahtjeva listu primljenih poruka te ih čita. Nakon slanja ili čitanja poruke, korisnik može obavljati novu operaciju čime se vraća na stanje logiran ili se može odjaviti čime prelazi u stanje odjava.

Ako korisnik želi čitati i uređivati zadatak, prvo šalje zahtjev za pregledom zadataka te dobiva uvid u zadatke. Zadatak može po želji i potrebi čitati ili uređivati. Nakon čitanja ili promjene zadatka, korisnik može obavljati novu operaciju čime se vraća na stanje logiran ili se može odjaviti čime prelazi u stanje odjava.

6.2. *Opći prioriteti i skica sustava*

Arhitektura sustava je objektno usmjerena. Nove značajke se mogu jednostavno dodavati uz vrlo male izmjene što arhitekturu čini stabilnom. Osigurava se veća pouzdanost i mogućnost jednostavnog održavanja sustava.

Sustav je podijeljen na klase. Podjela na manje dijelove tj. klase pridonosi jednostavnosti, povećava razumljivost te olakšava zamjenu dijelova bez opsežne intervencije u cijeli sustav.

Funkcijska kohezija je velika jer je kôd koji obavlja pojedinu operaciju grupiran u metodu klase. Time su međusobno povezani elementi grupirani što olakšava razumijevanje i promjene u sustavu. Klasa ima dobru komunikacijsku koheziju jer obavlja samo upravljanje podacima tj. grupira sve podsustave koji pristupaju ili mijenjaju određene podatke. Međuovisnost u objektnom oblikovanju smanjena je prijenosom jednostavnijih varijabli u argumentima metoda klasa.

Objektno oblikovanje smanjuje složenost sustava. Zadržavanje što više razine apstrakcije omogućava razumijevanje biti podsustava bez nepotrebnih detalja. Atributi klase su podatkovne apstrakcije, dok metode klase predstavljaju proceduralne apstrakcije.

Uporabom klasa oblikovanje je pojednostavljeno, a ponovna uporabivost povećana. Klase također omogućuju laku nadogradnju sustava te povećavaju uporabu postojećeg.

Model klijent-poslužitelj

U procesu oblikovanja programske potpore teži se povećanju razine apstrakcije. Organizaciju sustava s visoke razine apstrakcije moguće je prikazati modelom klijent-poslužitelj. Klijent zahtijeva određene usluge od poslužitelja, a poslužitelj osigurava usluge za klijenta.

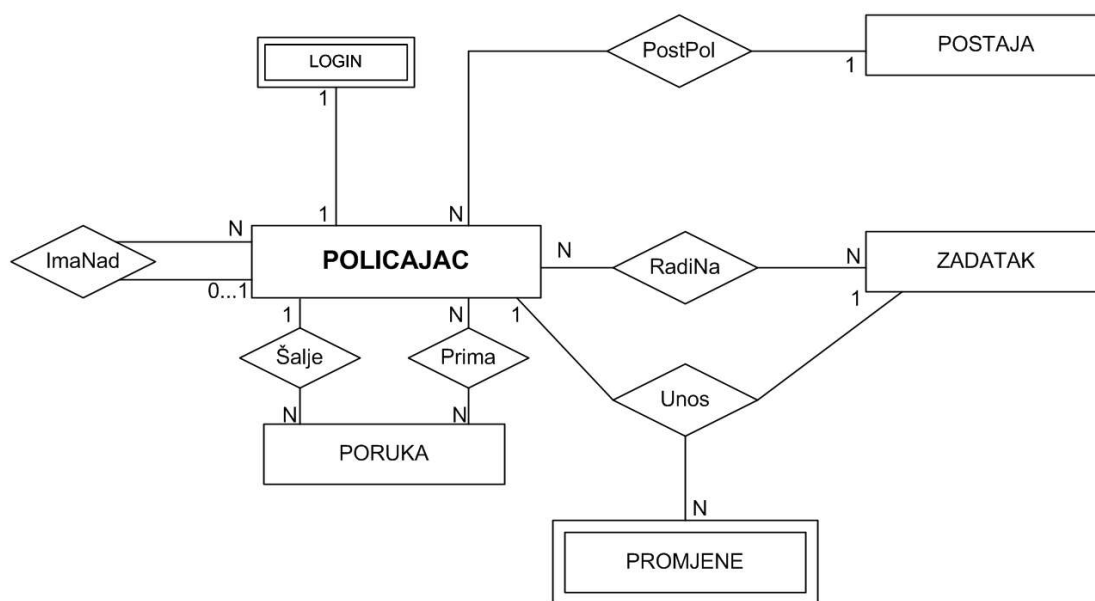
Poslužitelj započinje s radom i osluškuje komunikacijski kanal tj. čeka na dolazak klijentskog zahtjeva. Klijent započinje s radom i obavlja razne operacije koje

iziskuju interakciju s poslužiteljem. Poslužitelj čeka na poruke koje dolaze od spojenih klijenata. Kada pristigne poruka nekog klijenta, poslužitelj poduzima akcije kao odziv na poruku.

Npr. ako policijski službenik želi uređivati zadatak, mora pristupiti bazi podataka da bi dobio uvid u zadatke. Komunikacija s bazom podataka vrši se preko poslužitelja. Poslužitelj također sudjeluje u pohrani promjena na zadatku koje je policijski službenik napravio, a koje moraju biti ažurirane u bazi podataka.

Osnovna prednost modela klijent-poslužitelj je u tome što nije potreban sustav za upravljanje bazom podataka (SUBP) na svakom klijentskom računalu, već se klijent s bazom podataka povezuje preko aplikacije. Time se postiže veća sigurnost i zaštita podataka.

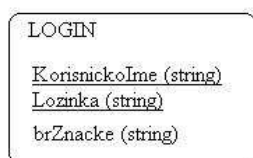
Baza Podataka



Slika 6.4. ER model baze podataka

Koristit ćemo relacijsku bazu podataka u kojoj se nalaze relacije (tablice) povezane i u odnosima kako je prikazano na *Slici 6.4*. Slijedi detaljniji opis po relacijama:

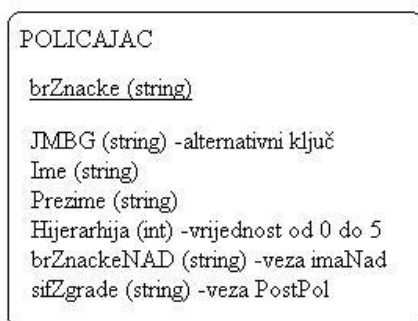
1. LOGIN:



Relacija Login koristi se za početak rada na sustavu. U njoj se pohranjuju sve važeće kombinacije korisničkog imena i lozinke, te se preko broja značke povezuju sa relacijom Policajac, te time i sa podacima osobe koja se

pokušava spojiti na sustav.

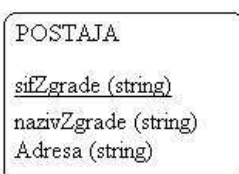
2. POLICAJAC (+imaNad+PostPol):



Odredili smo da se svima koji se spajaju na sustav dodjeljuje broj značke, tako da je za policajce to njihov stvarni broj značke, koji, recimo, uvijek počinje sa 1, dok ostale uloge u sustavu (npr. administrator i tajnik) imaju drugačiji broj značke, koji uvijek počinje sa 0. Osim JMBG-a, imena i

prezimenaa, za svaku osobu piše i razina u hijerarhiji. Postoje 4 razine među policajcima, gdje je razina 1 najviša, a razina 4 najniža. Razina 0 označava administratora, a razina 5 tajnike koji se brinu o digitalizaciji. Atribut 'broj značke nadređenog' povezuje relaciju samu sa sobom kako bi se za svakog policajca znalo tko mu je izravni nadređeni, ako ga ima. Atribut 'šifra zgrade' povezuje relaciju sa relacijom Postaja, kako bi se znalo u kojoj zgradi osoba radi.

3. POSTAJA:



4. ZADATAK:

```
ZADATAK
nazivZ (string)
trajeOD (date)
trajeDO (date)
vrijeme (time) -očekivano trajanje
prioritet (int)
povjerljivost (int)
opis (string)
```

Naziv zadatka je jedinstvena šifra zadatka (npr. neki redni broj pa ime, zajedno). Ostali atributi pobliže određuju zadatak.

5. PROMJENE (+unos):

```
PROMJENE
nazivZ (string)
rbrPromjene (int)
brZnacke(string)
opisPromjene (string)
```

U svakom zadatku promjene se razlikuju po rednom broju (kako su se odvijale) koji za svaki zadatak kreće od 1. Broj značke povezuje sa relacijom Policajac, a time i podacima o tome tko je dotičnu promjenu unio.

6. PORUKA (+Šalje):

```
PORUKA
rbr (int)
nazivP (string)
datumSLANJA (date)
brZnackeSALJE (string)
putanja (int)
poruka (string)
tipDat (string)
```

Svaka poruka ima svoj jedinstveni redni broj, uz koji se sprema njen naziv, datum slanja, broj značke osobe koja ju šalje, što je poveznica sa relacijom Policajac, te putanja na lokaciju u memoriji gdje je sadržaj (multimedijski) i sam tekst tekstualne poruke. Uz putanju tu je i tip podatka, koji označava

radi li se o slici ,zvuku, videu... (mp3, jpg,...).

7. veza Prima:

```
PRIMA
brZnackePRIMA (string)
rbr (int)
```

Kako jedan policajac jednu poruku može poslati više drugih policajaca, ovom relacijom se povezuje redni broj poruke koja je poslana sa svim policajcima kojima je poslana.

8. veza RadiNa:

```
RADINA
nazivZ (string)
brZnacke (string)
zatvoren (int) -0 ili 1, nadređeni
zaključan (int) -0 ili 1, podređeni
vidljiv (int) -0 ili 1
```

Kombinacija naziva zadatka i broja značke povezuje zadatak sa svim policajcima kojima je zadan. Kada nadređeni zada zadatak, vidljiv je samo administratoru (vidljiv=0), dok ga on ne odobri, kada postaje vidljiv i svim podređenima kojima je zadan. Kada neki od podređenih označi da je zadatak gotov (zaključan=1), zadatak postaje vidljiv samo nadređenom koji onda treba označiti da je zadatak stvarno završen (zatvoren=1).

6.3. Temeljna pitanja u oblikovanju

Izbor razreda objekata, odabranih tako da na što bolji način zajedno ostvaruju implementaciju željenog sustava, prikazan je pomoću dva class dijagrama, jednog sa strane klijenta, i jednog sa strane poslužitelja.

Klijent:

Klasa **Sustav** je središnja klasa na klijentskoj strani sustava. Ona nam je prozor prema poslužitelju, pa i samoj bazi s podacima. Osnovni cilj klase je poslati podatke prema poslužitelju i primiti podatke od poslužitelja. Dvije javne metode *primiPoruku()* i *saljiPoruku(poruka: String)* su zadužene za komunikaciju. Kod slanja poruke metoda kao argument prima varijablu koja je tipa *string*. Ta poruka se šalje preko TCP/IP protokola prema poslužitelju. Kad metoda *primiPoruku()* primi poruku, ona poziva privatnu metodu *obradiPoruku(poruka: String)* koja ne vraća nikakve podatke. Navedene metode se koriste svakih nekoliko stotina milisekundi „ugniježdene“ u obliku *obradiPoruku(primiPoruku(poruka))*.

Klasa **Poruka** sadrži podatke o svakoj (multimedijskoj) poruci koju policijski službenik može vidjeti. Klasa sadrži slijedeće attribute: *datoteka*, *brojPoruke*, *datumSlanja*, *prima*, *šalje*, *tekst*, *naslov* i *tipDatoteke*. Atributi klase su istog tipa kao i atributi relacije u bazi podataka predviđene za pohranu poruka.

Klasa **Policajac** služi kao temeljna klasa za identifikaciju korisnika koji je logiran na sustav. Klasa sadrži dvije javne metode koje se koriste za prijavu i odjavu. Metoda *prijava* preko klase *Sustav* šalje zahtjev za logiranjem do poslužitelja koji mu odgovara u slučaju točnih podataka za prijavu.

Klasa **OvlastiPodređeni** nasljeđuje klasu *Policajac*. U našem sustavu prava vezana uz zadatke koja ima svaki podređeni policijski službenik su čitanje i izmjena te zaključavanje zadatka. Zbog toga postoje javne metode *popisZadataka()*, *dodajPromjenu(nazivZ: String, pro: Promjena)* i *zakljucajZadatak(nazivZad: String)*.

Klasa **OvlastiNadredeni** je klasa po hijerarhiji viša od klase *OvlastiPodredeni*, ali zbog dvije uloge koje može obavljati jedan policijski službenik (biti u isto vrijeme i podređeni i nadređeni), klasa *OvlastiNadredeni* nasljeđuje klasu *OvlastiPodredeni*. Neke dodatne javne metode su *zadajZadatak()* i *zatvoriZadatak(brojZadatka: int)* prema ovlastima koje policijski službenici imaju u sustavu. Nadređeni zadatak može zadavati podređenima te zaključani zadatak može zatvoriti.

Klasa **OvlastiSefPolicije** nasljeđuje *OvlastiNadredeni* i *OvlastiAdministrator*. Zbog programskog ograničenja nasljeđivanja samo jedne klase, klasa *OvlastiSefPolicije* nasljeđuje samo *OvlastiNadredeni* i ima neke dodatne metode kao i klasa *OvlastiAdministrator*.

Administrator je osoba koja nema izravne veze s policijom i policijskim poslom, već dodjeljuje zadatke na zapovijed nadređenih. Zbog toga klasa **OvlastiAdministrator** nasljeđuje klasu *Policajac*. Administrator također može postavljati vidljivost zadataka zadanih od strane nadređenih što radi preko javne metode *prikaziZadatak(nazivZ: String)*.

Posljednja klasa osoba koje mogu koristiti sustav je **OvlastiTajnik**. Jedina mogućnost tajnika je unos zadataka za što se koristi javna metoda *digitaliziraj(zad: Zadatak)*.

Klasa **Zadatak** sadrži attribute kao i relacije *Zadatak* i *RadiNa* te attribute *popisPromjena* i *brojPromjena*. Za nove zadatke koristi se javna metoda *napraviZadatak()*.

Promjena je klasa koja se koristi preko klase *Zadatak*. Sadrži attribute *brojZnačke* i *opisPromjene* tipa *string* te atribut *redniBroj* koji je tipa *integer*. Klasa ima jednu javnu metodu *dodajPromjenu()* koja kao argumente prima opis same promjene i broj značke policijskog službenika koji vrši promjenu.

Klasa **FTP Klijent** služi za dohvaćanje i spremanje datoteka za što koristi metode *salji()* i *primi()*.

Poslužitelj:

Kao i na strani klijenta, i na strani poslužitelja postoji središnja klasa koja je zadužena za komunikaciju. To je klasa **SustavServer**. Cilj klase je komunikacija s klijentom.

Cijeli *poslužitelj* rastavili smo na dijelove za rad nad bazom te na dijelove zadužene za obradu zahtjeva pristiglih od strane klijenta. Dio za upravljanje i korištenje baze podataka rastavljen je na tri klase.

Klasa **UpitiSelect** se koristi kada želimo pročitati neke podatke iz baze. Klasa sadrži javne procedure s upitima koji daju različite rezultate. Neke od metoda su *brojZadnjePoruke()* koja vraća podatak tipa *integer*, *popisIsteRazine()* koji prima atribut *hijerarhija* tipa *string*, a daje popis svih policijskih službenika koji imaju hijerarhiju koju proslijeđujemo u metodu.

UpitiInsert je klasa koja ujedinjuje metode koje sadrže upite za upisivanje podataka u bazu. Neke od tih metoda su *dodajPoruku()*, *dodajZadatak()* i sl.

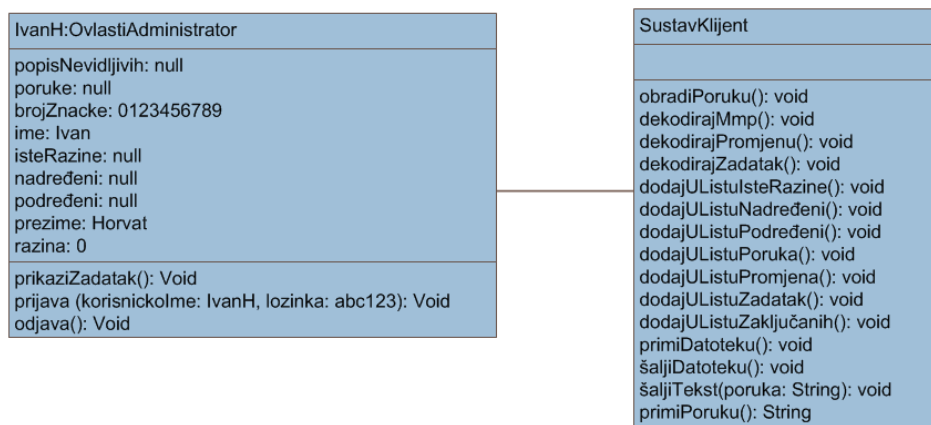
Zadnja klasa za rad nad bazom je **UpitiUpdate**. Kao i što sam naziv govori, sadrži upite za promjenu podataka. Promjene koje sustav vrši su postavljanje vidljivosti zadatka, zaključavanje i zatvaranje zadataka. Prema tome postoje tri javne metode sličnih naziva *postaviVidljiv()*, *zakljucajZadatak()* i *zatvoriZadatak()*. Sve tri metode primaju jedan argument, a to je naziv zadatka koji je tipa *string*.

Klasa **Zahtjev** opisuje zahtjeve koje klijent šalje poslužitelju. Klasa sadrži attribute *ipAdresa*, *akcija* i *podaci* koji su svu tipa *string*.

Općenito:

Svi atributi klasa su privatni. Za svaki od njih napisane su dvije funkcije: *get()* i *set()*. Prva funkcija, *get()* vraća podatak onog tipa kojeg tipa je atribut. Ne prima nikakav argument. Ako je vrijednost *null*, tj ne postoji, vraća vrijednost *-1*. Funkcija *set()* ne vraća nikakvu vrijednost već se koristi za postavljanje vrijednosti atributima u klasi. Prima argument ili više njih onih tipova koji se koriste za attribute u klasama.

Primjer korištenja instanci razreda, objekata:



Slika 6.7. Objektni dijagram –Login Administratora

Objektni dijagram prikazuje stanje u sustavu nakon što se administrator prijavi na sustav. Iz dijagrama razreda vidljivo je da će naslijediti klasu Policajac te tako dobiti sve attribute i metode iz te klase. Instanca klase OvlastiAdministrator ima na raspolaganju instancu klase SustavKlijent u kojemu se nalaze sve metode koje može koristiti u radu.

6.4. *Detalji oblikovanja, algoritmi i strukture podataka*

Kako bi se ostvarila komunikacija između poslužiteljske i klijentske aplikacije potrebno je bilo osmisliti način na koji oni komuniciraju. Ta komunikacija odvija se preko poruka specijalnog formata koje si aplikacije međusobno šalju. Format poruke za slanje dobiva se kodiranjem poruke na kraju aplikacije koja šalje poruku te dekodiranjem poruke na početku aplikacije koja prima poruku. Komunikacija može biti obostrana, dakle poslužitelj šalje poruku klijentu i obratno. Ako klijent šalje zadatak prema poslužitelju (bazi) ta poruka se kodira te će biti npr. oblika :

"ZZnaziv;trajanjeOD;trajanjeDO;trajanje;policajac1;policajac2;policajac3".

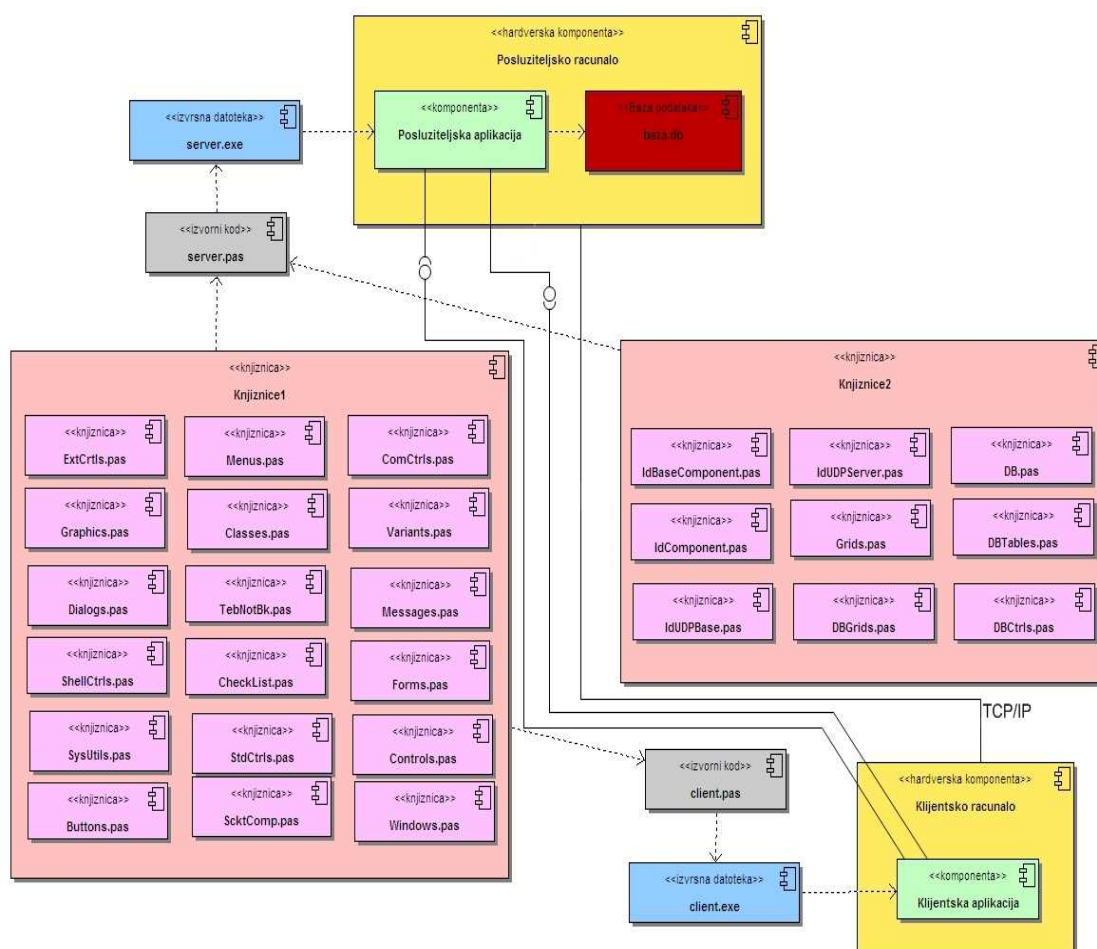
Poslužitelj zatim prima poruku, dekodira ju i radi određenu akciju s njom.

Aplikacije za skladištenje podataka u svome radu većim dijelom koriste nove složene tipove podataka, jedinstvene za naš sustav. Tako klasa Zadatak u sebi nosi podatke o zadatku određenog policijskog službenika i sve podatke koji su potrebni da bi taj zadatak detaljno definirali. Klasa Policajac ima mogućnosti čuvanja svih podataka vezanih za jednog policijskog službenika i njegovoj ulozi u policijskoj postaji. Komunikacija među zaposlenicima odvija se preko klase Poruka koja sve podatke vezane za uspješnu komunikaciju sadrži u sebi. Od ugrađenih struktura podataka koriste se polja. Polja instanci Zadatak, polja instanci Policajac, itd.

Komponentni dijagram na *Slici 6.8* predstavlja grubi prikaz komponenata od kojih je izgrađen sustav. Sklopovske komponente na kojima se i nalaze sve ostale programske komponente osiguravaju sve funkcionalnosti fizičkog svijeta te se u ovom odjeljku time ne bavimo. Možemo samo navesti da su računala generička te nema nikakvih posebnih zahtjeva na njihovu arhitekturu i funkcionalnosti. U ovom su slučaju dva računala, klijentsko i poslužiteljsko, spojena na IP mrežu te komuniciraju preko TCP protokola radi većeg osiguranja prijenosa informacija. Klijentskih računala može biti više, ovisno o mediju povezivanja i snazi poslužiteljskog računala.

Na klijentskom računalu pokreće se klijentska aplikacija koja za svoj rad koristi izvršnu datoteku client.exe. Client.exe ima svoj izvorni kod client.pas napisan u Pascalu koji koristi skupinu knjižnica (grupirane u „Knjiznice1“) u kojima se nalaze funkcije i potprogrami koji su bitni za izvođenje aplikacije. Klijentska aplikacija nudi sučelje poslužiteljskoj aplikaciji.

Na poslužiteljskom računalu pokreće se poslužiteljska aplikacija koja za svoje izvođenje koristi svoju izvršnu datoteku server.exe. Server.exe sadrži server.pas izvorni kod, također napisan u Pascal-u, koji koristi dvije skupine knjižnica, Knjiznice1 i Knjiznice2. Poslužiteljska aplikacija za spremanje podataka koristi bazu podataka te nudi sučelje klijentskoj aplikaciji.



Slika 6.8. Komponentni dijagram sustava

7. Povijest rada i trenutni status implementacije

- Programski dio:
 - Implementirano:
 - Kreirana je baza i napravljeni su kompletni upiti nad bazom potrebni za rad sustava.
 - Korisnici razgovaraju preko tekstualnih poruka. Svaka poruka na početku sadrži podatak koji poslužiteljskoj ili klijentskoj aplikaciji govori što treba napraviti s tom porukom.
 - Riješeno povezivanje na bazu podataka s poslužiteljem preko gotovih komponenti za tu namjenu.
 - Klijentska i poslužiteljska aplikacija komuniciraju povezujući se na bazu podataka. Svaka aplikacija zasebno šalje upit te na njega dobiva odgovor od poslužitelja.
- Dokumentacija:
 - Napisano od prošle revizije:
 - Implementacija i korisničko sučelje, koja sadrži dijagram razmještaja, korištene tehnologije i alate, isječak programskog koda vezan za temeljnu funkcionalnost sustava, testiranje programskog rješenja, upute za instalaciju te korisničke upute.
 - Zaključci i smjernice za daljnji rad.

Problemi u realizaciji:

- Ekipa Gangsteri imala je poteškoća u pronalaženju mjesta sastajanja. Nije se mogao naći prostor te se dosta dragocjenog vremena trošilo na traženje drugih lokacija. Stalni sastanci cijelog tima su se pokazali nepraktičnima i neproduktivnima, pa se komunikacija prenijela većim dijelom na Internet, a kraći sastanci uživo se i dalje odvijaju radi pregledavanja napravljenog i dogovora oko podjele budućeg posla.
- Previše obaveza svakog člana tima posebno (za što je poprilično zaslužan fakultet) dovelo je do povremeno slabe komunikacije među članovima te otežanog rada na projektu.

- Službeno pružene informacije i opis projektne dokumentacije drugog dijela bili su nejasni pa se i uz obaveznu, a i neobaveznu, literaturu teško moglo raditi. Osim nejasno opisanog tekstualnog dijela, radilo se i o podosta dijagrama, koji su intuitivski svi slični jedan drugome, a iz literature nismo mogli doznati velike razlike niti jesu li te razlike ispravne. Teško je bilo razlučiti što i kako. Na taj način opet smo gubili vrijeme kojeg nemamo na pretek.
- Najveći problem trećeg ciklusa su praznici preko kojih je teško tražiti produktivnost od članova tima te je poslije potrebno vrijeme da se svi ponovno uhodaju.

Podjela posla:

Mirko Jambrošić razradio je i predložio rješenje povezivanja s bazom, programskog sučelja te je radi svog znanja programiranja u Delphiju zadužen za razvoj istoga. Kako je i najviše zadužen za programiranje samog programa napravio je i dijagram razreda uz pomoć ostalih članova tima. U programskom dijelu su mu, proučavajući Delphi i rad na njemu, pomogli Andrija Čajić i Ante Kegalj, koji su se uz to pozabavili razradom plana rada i obrascima uporabe.

Na pisanju obrazaca uporabe i radu s bazom podataka radio je Filip Lasić koji je istraživao za potrebe dokumentacije. Njegova zadatak bila je popuniti bazu podataka koju su kreirali Mirko Jambrošić i Petra Zadro, kako bi Martina Jakovina, Marin Biberović, Martina Kocet te Petra Zadro, koja je uz to napravila i ER model baze, mogli pisati upite nad njim. Martina Jakovina i Petra Zadro učile su i izrađivale use case i sekvencijske dijagrame te zajedno napravile i dijagram stanja, a Martina Kocet i Marin Biberović bavili su se opisom projektnog zadatka i razradom funkcionalnih i nefunkcionalnih zahtjeva. Što se ostalih dijagrama tiče, za dijagrame aktivnosti i kolaboracijske dijagrame zadužena je sveprisutna Martina Kocet, dok je dijagram komponenata za cijeli sustav i primjer objektnog dijagrama razradio i napravio Marin Biberović.

O izboru arhitekture temeljem principa oblikovanja pisala je Martina Jakovina, a odlično je obavila i izradu loga programa. Filip Lasić je u trećem ciklusu od Ante Kegalja preuzeo pisanje povijesti rada te dodao zaključke i daljnje smjernice rada te korištene tehnologije i alate.

Mirko Jambrošić, Ante Kegalj i Andrija Čajić su iskomentirali odsječke koda, dok su ogromnom poslu testiranja programa svoj doprinos dali svi članovi tima. Pred kraj su ostali još samo zahtjevni poslovi pisanja korisničkih uputa te uputa za instalaciju, koje su odradili Petra Zadro i Martina Jakovina te Martina Kocet. I na kraju, trebalo je promijeniti neke dijagrame, kao i ovo poglavlje dokumentacije kako bi sve odgovaralo ostvarenim ciljevima u ovom projektu.

Treba napomenuti da su članovi tima odlično komunicirali, kako iz prvog lica, tako i preko google grupe koja se pokazale neizostavnom.

8. Implementacija i korisničko sučelje

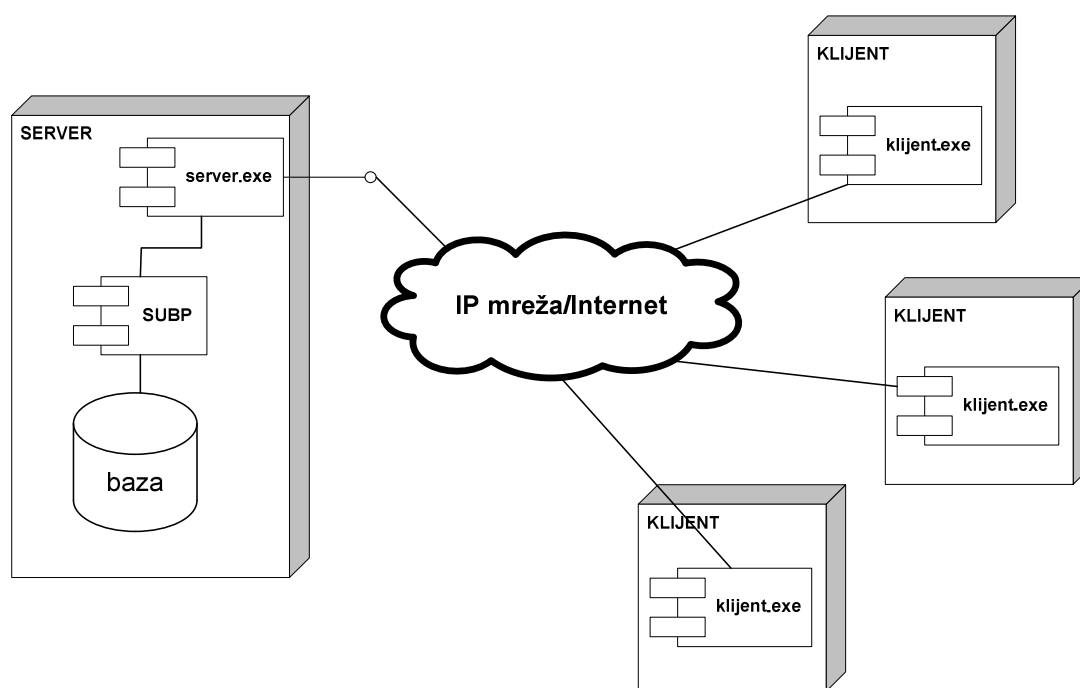
8.1. Dijagram razmještaja

Dijagram razmještaja prikazuje generalnu topologiju sustava koji se koristi kako bi resursi bili optimalno raspoređeni. Predstavlja statički pogled na razmještaj sklopovskih i programskih komponenta.

Komunikacija između računala SERVER i računala KLIJENT odvija se preko bilo koje „Internet Protocol“ (IP) mreže kako bi se omogućila najveća moguća mobilnost i kompatibilnost klijenata na bilo kojem području. Računalo SERVER sadrži sve potrebne datoteke i aplikacije, a računalo KLIJENT klijentsku aplikaciju koja je vrlo „lagana“, odnosno ponaša se kao „thin client“ u tzv. klijent-server arhitekturi.

U bazi podataka sadržane su sve potrebne informacije i datoteke koje korisnik može zahtijevati. Samo se eksplicitnim zahtjevom korisnika materijali sa SERVER-a (iz baze) umnožavaju na računalo KLIJENT.

Vidljivo je da ovakav razmještaj komponenta pogoduje sigurnosti koja je za aplikaciju ove namjene iznimno bitna te da fizička raspodjela omogućuje iznimno laganu izradu sigurnosnih kopija sustava.



8.2. Korištene tehnologije i alati

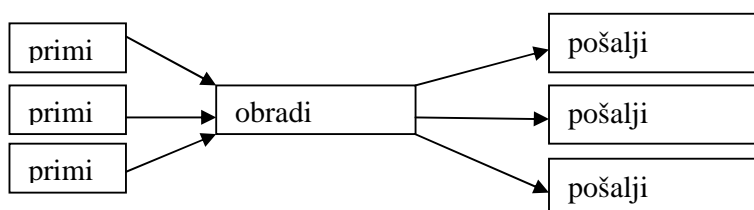
U izradi cijelog programa korištena je Borland Delphi razvojna okolina. Za bazu je korišten Delphi Database Desktop. Korišteni programski jezik je Delphi, kojemu je podloga objektni Pascal. Za dijagrame je korišten AgroUML(Unified Modeling Language). Na računalu na kojem će se instalirati server, treba biti instaliran i Delphi 6.0 ili inter base server 6.0 i inter base client 6.0

Prilikom izrade nekih slika je korišten i Microsoft Visio koji nudi veliki broj predefiniranih oblika i simbola pa tako olakšava i ubrzava rad. Za zaštitu podataka koristi se raid0.

8.3. Isječak programskog koda vezan za temeljnu funkcionalnost sustava

Temeljna funkcionalnost sustava je komunikacija između klijenta i servera. Server sluša na TCP portu 30093, klijent sluša odgovore od servera na TCP portu 30094. Komponenta koja je zadužena za primanje poruka kod klijenta i kod servera konstantno bez prestanka osluškuje i čeka na inicijalizaciju komunikacije sa druge strane.

Server slijedno prima poruke te ih onda paralelno obrađuje, nakon obrade šalje tražene informacije prema zasebnim klijentima.




```
procedure TForm1.ServerSocket1ClientRead(Sender: TObject;  
  Socket: TCustomWinSocket);  
var  
  i:integer;  
  sRec : string;  
begin  
  for i := 0 to ServerSocket1.Socket.ActiveConnections-1 do  
  begin  
    with ServerSocket1.Socket.Connections[i] do  
    begin  
      sRec := ReceiveText;  
      if sRec <> "" then  
      begin  
        sRec:=sRec+'<'+RemoteAddress+'>';  
        server.lista_cekjanja(sRec);  
      end;  
    end;  
  end;  
end;  
end;
```

Svaki klijent koji se spoji na server otvara novu konekciju. Za svaku konekciju gleda se da li ima nova poruka, te se zajedno sa izvorišnom IP adresom odakle poruka dolazi *RemoteAddress* šalje na obradu. Nakon završetka obrade poruke poziva se funkcija *salji_klijentu* (*poruka, ip*).

```
procedure Sustav.salji_klijentu(const poruka:String; adresa:string);  
begin  
  with SaljiPorukuDretva.Create(true) do  
  begin  
    podatak:=poruka;  
    ip_adresa:=adresa;  
    inicijalizacija();  
    FreeOnTerminate:=true;  
    Resume;  
  end;  
end;
```

Ova funkcija stvara novu dretvu, preko metoda *set (property)* postavlja podatke klase te postavlja zastavice koje su potrebne da nakon završetka dretva sama sebe uništi.

Procedura kojom inicijaliziramo IP adresu klijenta kojemu je poruka namijenjena i TCP port preko kojega se odvija komunikacija.

```
{procedura kojom inicijaliziramo 'klijenta' (gdje ce se  
spajati)  
autor: mirkec}  
procedure SaljiPorukuDretva.inicijalizacija();  
begin  
inc(Form1.broj_dretvi);  
KlientSoket:=TClientSocket.Create(KlientSoket);  
KlientSoket.Host:=ip;  
KlientSoket.Port:=94;  
KlientSoket.Active:=true;  
end;
```

Procedura koja pokreće samu dretvu te neumorno pokušava slati poruku prema klijentu nema nikakvu zaštitu od eventualnog prekida veze koji je moguć u svakodnevnim uvjetima (nestanak struje, problemi sa konekcijom na Internet...).

Prilikom uspješnog slanja dretva se uništava sa *Terminate()*.

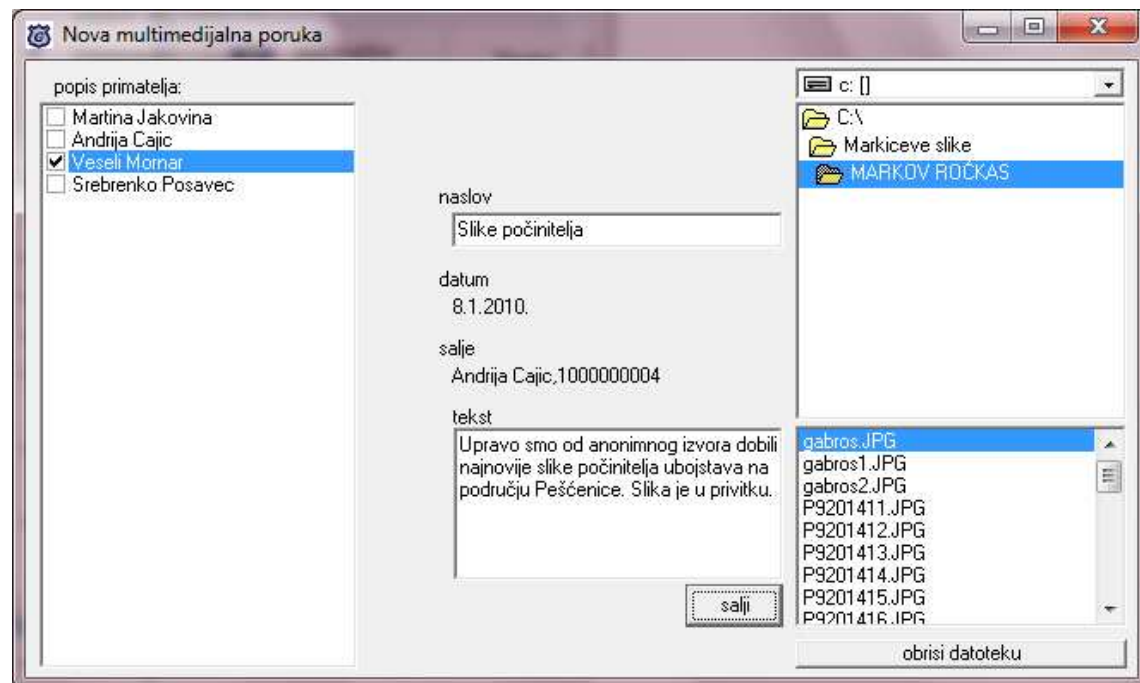
```
{procedura koja se pokrece kod pokretanja dretve  
autor: mirkec}  
procedure SaljiPorukuDretva.Execute();  
begin  
while (not(KlientSoket.Socket.Connected)) do  
begin  
pokazi(IntToStr(Self.ThreadID)+' nije spojen');  
Sleep(100); //spava 100 ms  
end;  
KlientSoket.Socket.SendText(podaci);  
pokazi(IntToStr(Self.ThreadID)+' spojen');  
dec(Form1.broj_dretvi);  
Terminate;  
end;
```

8.4. Testiranje programskog rješenja

Test 1

Ulaz:

- Pokretanje poslužiteljske aplikacije, pokretanje klijent aplikacije
- Prijava korisnika acajic
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Nova poruka“
- Odabir primatelja: Veseli Mornar
- Popunjavanje potrebnih polja za stvaranja poruke te odabir .jpg datoteke u izborniku za odabir datoteke
- Odjava korisnika acajic



- Prijava korisnika vmornar
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Poruke“
- Otvara se novi prozor sa poljima za prikaz detalja poruke

Očekivani izlaz:

- U polju „popis poruka“ dodan je naslov poruke poslane od korisnika acajic
- Pri odabiru te poruke ostala polja u prozoru popunjavaju se detaljima o odabranoj poruci (pošiljalatelj, datum...)
- Na klik gumba „Preuzmi“ u desnom gornjem dijelu prozora prikazuje se .jpg datoteka poslana od strane korisnika acajic
- Na klik gumba „Otvori datoteku“ otvara se mapa u kojoj je spremljena datoteka poslana od strane korisnika acajic

Izlaz:

- Izlaz odgovara očekivanom izlazu



Test 2

Ulaz:

- Pokretanje poslužiteljske aplikacije, pokretanje klijent aplikacije
- Prijava korisnika pzadro
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Novi zadatak“
- Odabir primatelja: Martina Jakovina
- Popunjavanje potrebnih polja za stvaranja novog zadatka
- Odjava korisnika pzadro
- Prijava korisnika pzadro
- Postavljanje vidljivosti novog zadatka
- Odjava korisnika pzadro

The image displays two screenshots of a software application interface. The left window, titled "Novi zadatak", is a form for creating a new task. It features a list of subordinates on the left, with "Martina Jakovina" selected. To the right, there are input fields for "naziv:" (Svemirci napadju), "početak:" (15. 1. 2010.), "trajanje:" (5 dana), "prioritet:" (4), "povjerljivost:" (4), and a "tekst:" area containing the text "Svemirci će nas napasti u našoj najslabijoj točki.". A "Napravi" button is located at the bottom left. The right window, titled "N.", is a confirmation dialog. It shows "Svemirci napadju" selected under the "nevidljivi:" label. A "Vidljivo!" button is at the bottom.

- Prijava korisnika mjakovina
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Zaduženja“
- Otvara se novi prozor sa poljima za prikaz zaduženja korisnika mjakovina

Očekivani izlaz:

- U padajućem izborniku „popis zaduženja“ dodan je novi zadatak stvoren od korisnika pzadro
- Pri odabiru tog zaduženja ostala polja u prozoru popunjavaju se detaljima o odabranom zaduženju (naziv zadatka, prioritet, tajnost...)

Izlaz:

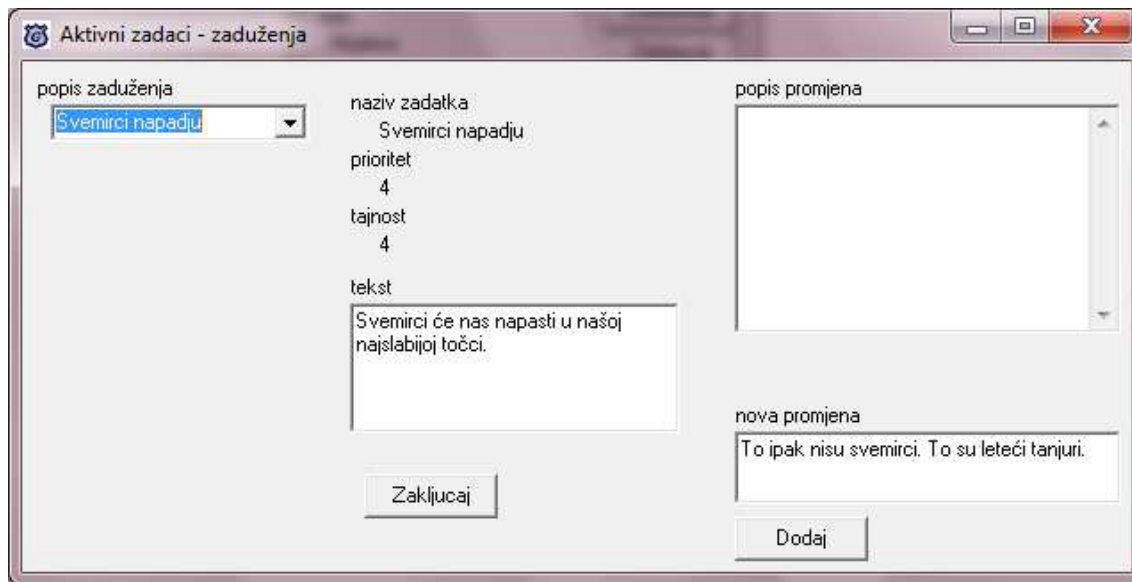
- Izlaz odgovara očekivanom izlazu

The screenshot shows a window titled "Aktivni zadaci - zaduženja" with a task management interface. On the left, a dropdown menu labeled "popis zaduženja" has "Svemirci napadju" selected. To the right, the task details are displayed: "naziv zadatka" is "Svemirci napadju", "prioritet" is "4", and "tajnost" is "4". Below these, a text area labeled "tekst" contains the message "Svemirci će nas napasti u našoj najslabijoj točki." At the bottom left is a "Zaključaj" button. On the right side, there is a large empty text area labeled "popis promjena" and a text input field labeled "nova promjena" with a "Dodaj" button below it.

Test 3

Ulaz:

- Nastavak testa 2 (Korisnik mjakovina)
- Upisivanje teksta u polje „nova promjena“
- Pritisak gumba „Dodaj“
- Pritisak gumba „Zaključaj“
- Izlaz iz prozora „Aktivni zadaci-zaduženja“



- Odjava korisnika mjakovina

Očekivani izlaz:

- Prijava korisnika pzadro
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Zatvori“
- Otvara se novi prozor sa listom zadataka koji čekaju potvrdu za zatvaranje
- Na listi je dodan zadatak koji je zaključan od strane korisnika mjakovina
- Označivši taj zadatak i kliknuvši gumb „Zatvori“ zadatak je službeno zatvoren i u budućem korištenju sustava neće biti vidljiv ni korisniku pzadro ni mjakovina

Izlaz:

- Izlaz odgovara očekivanom izlazu



Test 4

Ulaz:

- Pokretanje poslužiteljske aplikacije, pokretanje klijent aplikacije
- Prijava korisnika pzadro
- Nakon završenog procesa spajanja na poslužitelj, odabir opcije „Novi zadatak“
- Odabir primatelja: Andrija Cajic
- Popunjavanje potrebnih polja za stvaranja novog zadatka
- Odjava korisnika pzadro

The image shows two screenshots of a software application. The left window, titled "Novi zadatak", is for creating a new task. It features a list of subordinates on the left with "Andrija Cajic" selected. The right side contains fields for task details: "naziv" (Kriminal na otoku Cresu), "početak" (14.12.2009.), "trajanje" (6 dana), "prioritet" (4), "povjerljivost" (2), and "tekst" (neke kuće se grade bez dozvole. pp). A "Napravi" button is at the bottom. The right window, titled "N.", shows a "nevidljivi" checkbox checked and a "Vidljiv!" button at the bottom.

- Prijava korisnika pzadro
- Postavljanje vidljivosti novog zadatka
- Odjava korisnika pzadro

Očekivani izlaz:

- Prijava korisnika acajic
- Nakon završenog procesa spajanja na server, odabir opcije „Zaduzenja“
- Otvara se novi prozor sa poljima za prikaz zaduženja korisnika acajic
- U padajućem izborniku „popis zaduženja“ dodan je novi zadatak stvoren od korisnika pzadro

- Pri odabiru tog zaduženja ostala polja u prozoru popunjavaju se detaljima o odabranom zaduženju (naziv zadatka, prioritet, tajnost...)

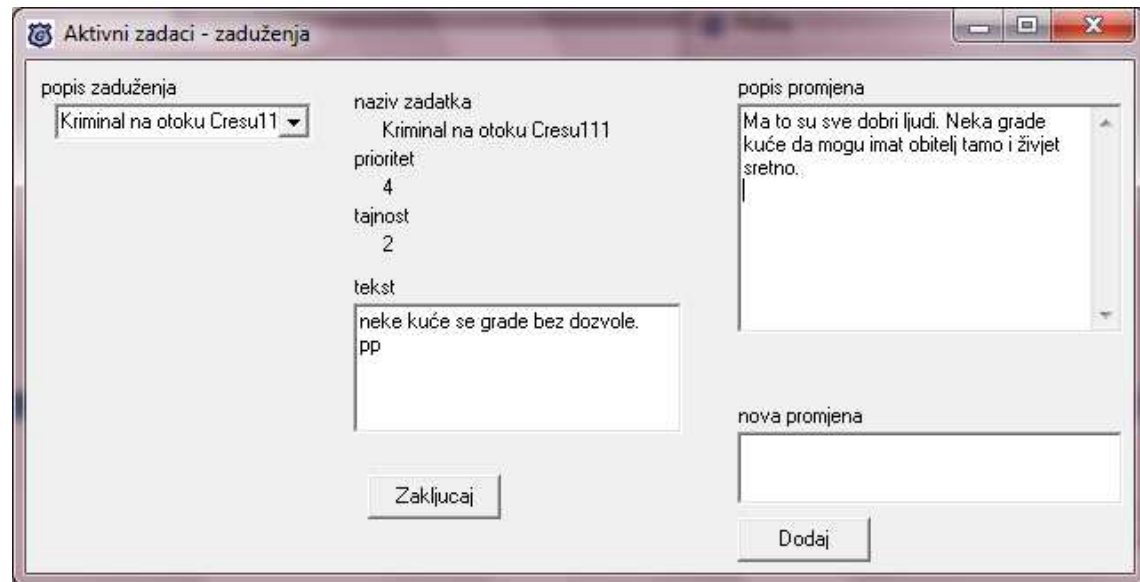
The screenshot shows a window titled "Aktivni zadaci - zaduženja". It contains several fields and buttons:

- popis zaduženja:** A dropdown menu with "Kriminal na otoku Cresu111" selected.
- naziv zadatka:** "Kriminal na otoku Cresu111"
- prioritet:** "4"
- tajnost:** "2"
- tekst:** A text area containing "neke kuće se grade bez dozvole. pp".
- popis promjena:** An empty text area.
- nova promjena:** An empty text input field.
- Zaključaj:** A button.
- Dodaj:** A button.

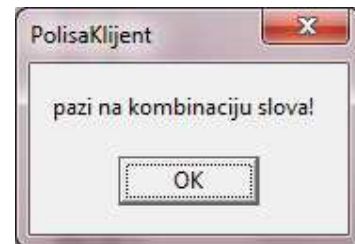
- Unos nove promjene

This screenshot shows the same window as the previous one, but with the "nova promjena" field filled with the text: "Ma to su sve dobri ljudi. Neka grade kuće da mogu imat obitelj tamo i živjet sretno." The "Dodaj" button is now highlighted with a dashed border, indicating it is the next step in the process.

- Odjava korisnika acajic
- Prijava korisnika acajic
- Nakon završenog procesa spajanja na server, odabir opcije „Zaduzenja“
- Odabir zadatka na kojem se dodala promjena
- U popisu promjena navedena promjena dodana od strane korisnika acajic

**Izlaz:**

- Pri zadavanju zadatka prilikom pisanja niza znakova „PP“ program je dojavio poruku:
- Zatim je prepravio napisani niz znakova „PP“ u „pp“
- Prilikom unosa imena zadatka koristilo se ime koje je već dvaput korišteno, pa su prilikom spremanja imenu na kraju dodane tri jedinice („Kriminal na otoku Cresu111“)
- Ostatak izlaza odgovara očekivanom izlazu

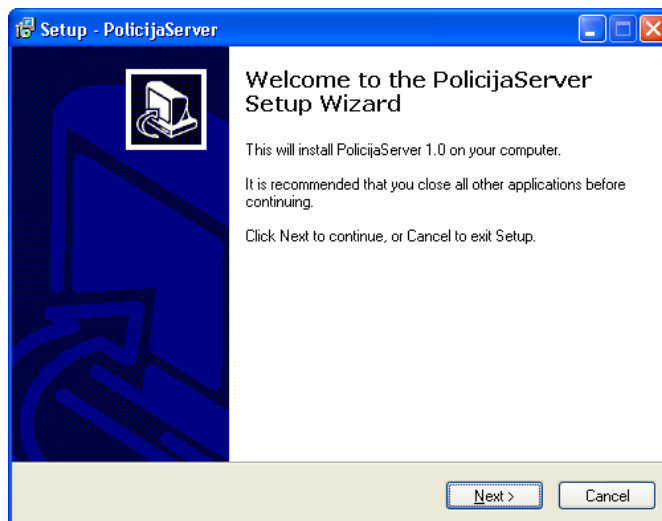


8.5. Upute za instalaciju

Za pokretanje aplikacije potrebno je instalirati program *PolicijaServer* na poslužitelju, te program *PolicijaKlijent* na klijentskim računalima.

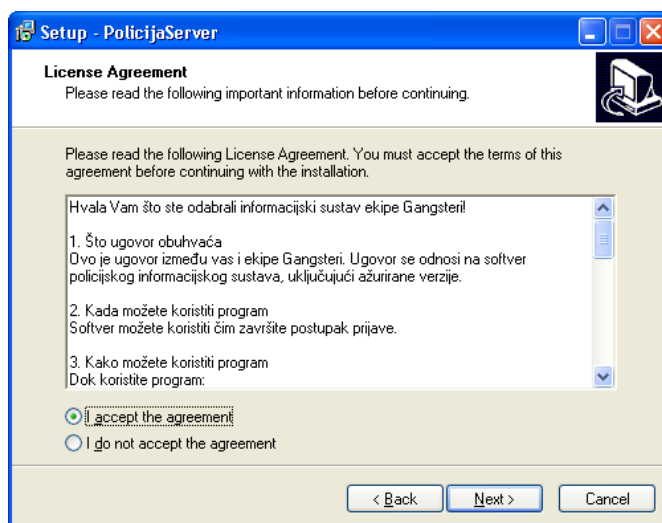
- Instalacija na poslužitelju

Potrebno je umetnuti instalacijski CD. Pokrenite ikonu *setupServer* za otvaranje prozora koji vas vodi kroz korake instalacije. U novo otvorenom prozoru koji prikazuje poruku dobrodošlice pritisnite *Next*.



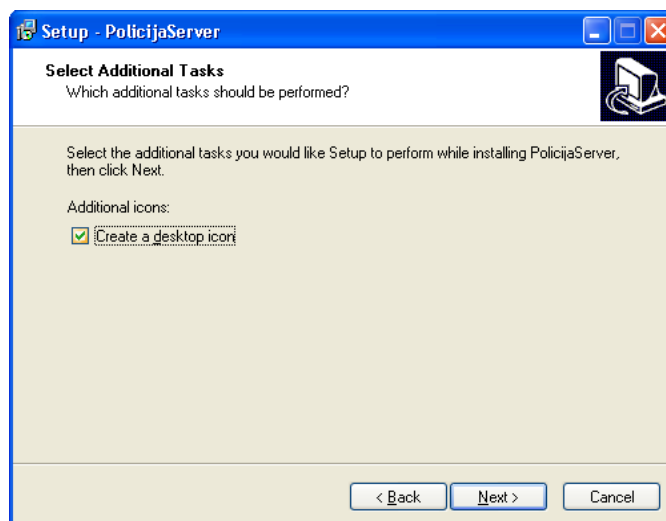
Slika 8.5.1: Prozor s porukom dobrodošlice

U slijedećem prozoru pojavit će se ugovor s uvjetima korištenja i ukoliko se slažete s navedenim pritisnite *I accept the agreement* i zatim *Next*. U protivnom ne možete nastaviti s instalacijom.



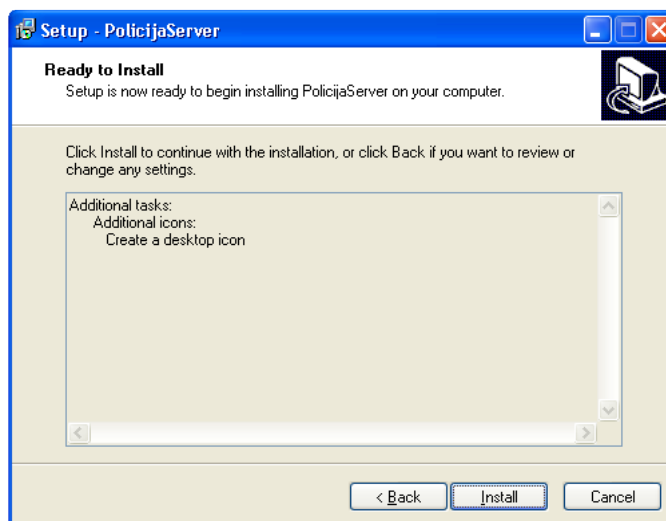
Slika 8.5.2: Prozor s uvjetima korištenja

Ukoliko želite kreirati ikonu na radnoj površini, u slijedećem prozoru koji se otvori pritisnite *Create a desktop icon* (preporučeno) i zatim *Next*:



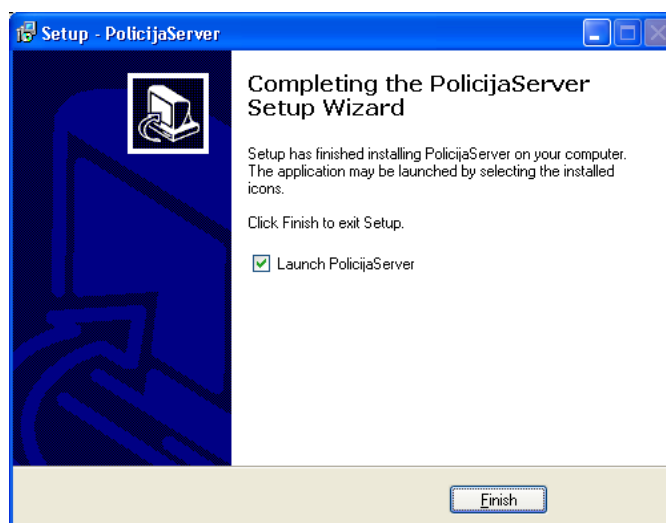
Slika 8.5.3: Stvaranje ikone na radnoj površini

U slijedećem prozoru prikazana je dodatna opcija za stvaranje ikone (ukoliko ste ju odabrali). Pritisnite *Install*:



Slika 8.5.4: Pritisnite Install za instalaciju

U posljednjem prozoru označite ukoliko želite pokrenuti program odmah nakon završetka instalacije odabirom opcije *Launch PolicijaServer* i pritisnite *Finish* za završetak:



Slika 8.5.5: Završetak instalacije

Prilikom instalacije izvršnog programa na poslužitelju, instalirat će se i baza podataka na lokaciji *C:\Program Files\Common Files\Borland Shared\Data* te datoteka *PolisaServer* na lokaciji *C:* koja omogućava izvršnom programu ostvarenje prijenosa podataka. Također je stvoren prečac u *Start Menu* za pokretanje programa.

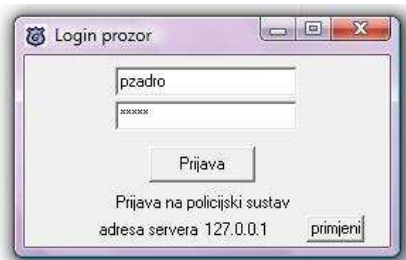
- Instalacija na računalu klijenta

Potrebno je umetnuti instalacijski CD. Pokrenite ikonu *setupKlijent* za otvaranje prozora koji vas vodi kroz korake instalacije. Koraci su isti kao kod instalacije na poslužitelju, dakle slijedite iste upute kao za poslužitelja.

Rezultat instalacije je izvršni program *PolicijaKlijent*, datoteka *PolisaKlijent* na lokaciji *C:* za prijenos sadržaja i prečac u *Start Menu* za pokretanje programa.

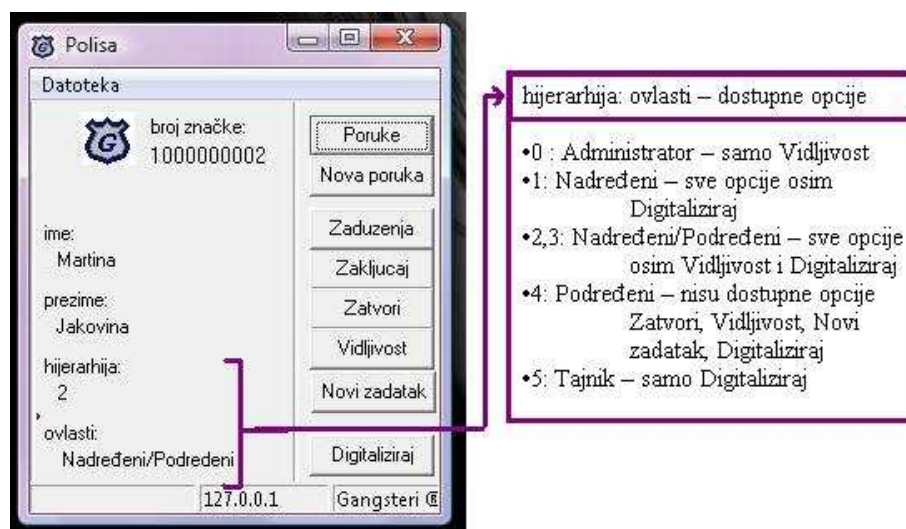
8.6. Korisničke upute

Za početak rada na sustavu potrebno je prijaviti se sa korisničkim imenom i lozinkom. Prilikom pokretanja programa 'PolisaKlijent' korisniku se otvara prostor za prijavu kao na *Slici 8.6.1* Osim unošenja korisničkog imena i lozinke, kliktanjem miša na području adrese servera, moguće je tu istu adresu promijeniti, nakon čega je potrebno odabrati tipku 'Primjeni'. Nakon unosa željenih podataka, korisnik se na sustav prijavljuje pritiskom na tipku 'Prijava'.



Slici 8.6.1: Prijava na sustav

Nakon prijave, korisniku se otvara glavni prozor na kojemu su na lijevoj strani prikazani njegovi osnovni podaci, a na desnoj moguće akcije unutar sustava. Na *Slici 8.6.2.* je uz izgled glavnog prozora prikazan i način na koji hijerarhija određuje ovlasti, koje pak određuju koje od akcija na desnoj strani prozora su dostupne. Pokuša li korisnik pokrenuti neku od akcija za koju nema ovlasti javlja se upozorenje.

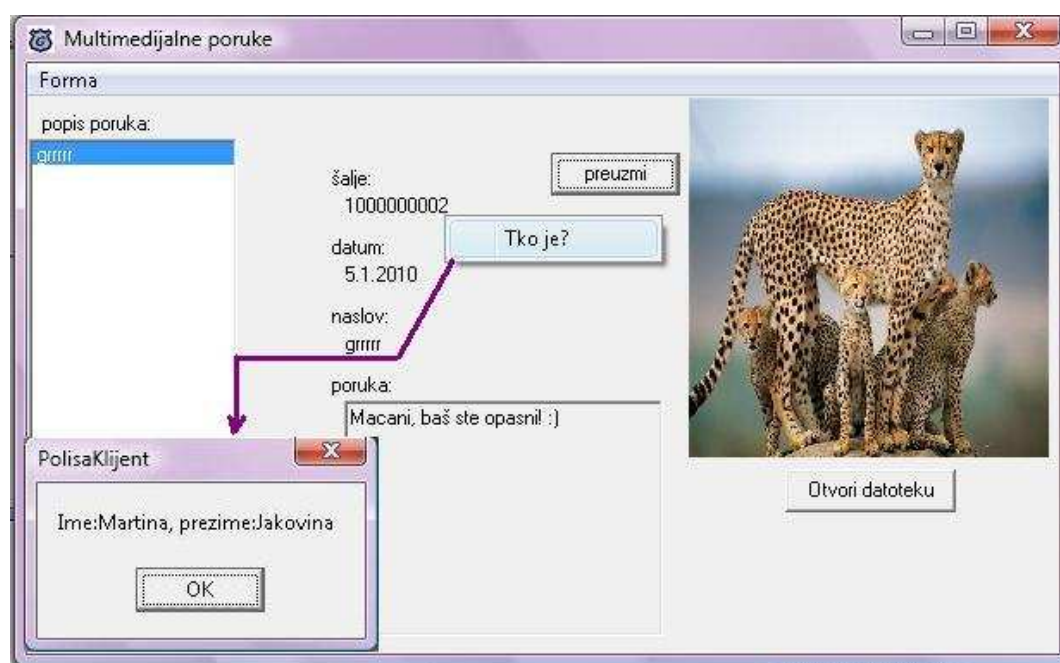


Slici 8.6.2.: Glavni prozor prijavljenog korisnika

PAZI! NAPOMENA: Nije dozvoljeno korištenje jednostrukih navodnika, a prilikom upisa točke-zareza (;), automatski se mijenja u samo zarez (.). Uz točku-zarez, server za svoje interne poruke koristi i neke kombinacije od dva velika tiskana slova, i ako korisnik pokuša negdje upisati baš jednu od tih kombinacija velikim slovima, da ne bi došlo do pogrešne interpretacije, ta se slova automatski prebacuju u mala tiskana slova.

8.6.1. Čitanje primljenih poruka

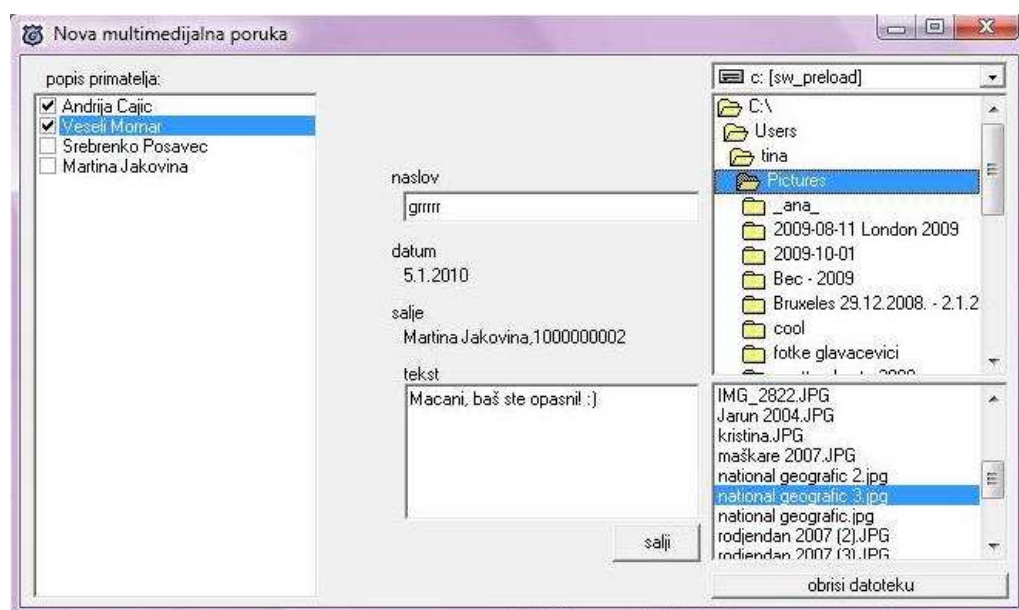
Odabirom opcije 'Poruke', korisnik dobiva uvid u primljene poruke (Slika 8.6.3.) Na lijevoj strani se nalazi popis poruka a odabirom neke od njih prikazuju se njeni podaci i sadržaj, na desnoj strani prozora. Za prikaz multimedijalnog sadržaja potrebno je još kliknuti na tipku 'Preuzmi' (koja se, kao i tipka 'Otvori datoteku', može kliknuti samo ako u primljenoj poruci i postoji multimedijalni sadržaj), a za otvaranje lokacije na kojoj je sadržaj spremljen (radi pokretanja videa npr.) potrebno je kliknuti na tipku 'Otvori datoteku'. Pošiljalatelj je označen brojem značke, ali desnim klikom na taj broj otvara se pitanje 'Tko je?' pritiskom na koje se prikazuje prozorčić sa imenom i prezimenom odgovarajuće osobe, kao što je prikazano na Slika 8.6.3.



Slika 8.6.3.: Pregled primljene poruke

8.6.2. Nova poruka

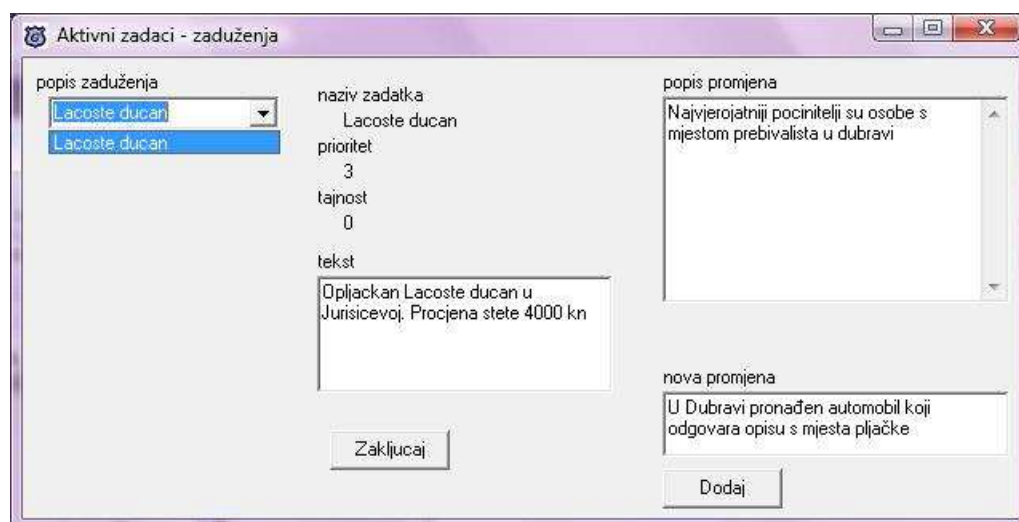
Odabirom opcije 'Nova poruka', korisniku se otvara obrazac za pisanje poruke kao na Slici 8.6.4. Na lijevoj strani se nalazi popis osoba koje su na istoj hijerarhijskoj razini, te se među njima odabiru primatelji poruke. Na desnoj strani se popunjavaju polja za naslov i tekst poruke, a ako se želi poslati multimedijalni sadržaj potrebno je još u desnom izborniku naći mapu u kojoj je sadržaj spremljen, te u najnižem prozorčiću među datotekama u odabranoj mapi pronaći onu koja se želi poslati. Na kraju se klikne 'Šalji', a prilikom sljedećeg prijavljivanja na sustav, svi primatelji poslane poruke, imat će je među primljenim porukama.



Slici 8.6.4. : Pisanje nove poruke

8.6.3. Zaduženja

Odabirom opcije 'Zaduženja', otvara se prozor prikazan na Slici 8.6.5 pomoću kojeg se mogu pregledavati zadaci, te vršiti promjene nad istima.



Slici 8.6.5: Pregled aktivnih zadataka i unos promjene na odabranom zadatku

U klizećem prozoru 'Popis zaduženja' korisnik ima uvid u sve zadatke koji su mu zadani i trenutno aktivni. Odabirom jednog od ponuđenih zadataka, prikažu se podaci vezani uz taj zadatak, uključujući sve promjene na zadatku unešene do trenutka prijavljivanja korisnika na sustav. Ako korisnik želi unijeti promjenu, tekst izmjene unosi u prozor 'Nova promjena', a klikom miša na tipku 'Dodaj' promjena se dodaje ostalim promjenama na zadatku. Omogućeno je i zaključavanje zadatka

izravno iz ove opcije, pomoću tipke 'Zaključaj'. Unesena promjena ili zaključavanje zadatka vidljiva je tek prilikom sljedeće prijave na sustav.

8.6.4. Zaključavanje, zatvaranje i vidljivost zadatka

Odabirom akcije 'Zaključaj' s glavnog prozora prijavljenog korisnika, prikaže se prozor prikazan na *Slici 8.6.6a*. Korisniku su ponuđeni svi trenutno aktivni zadaci na kojima radi te mu je pružena opcija, ukoliko je zadatak gotov, da ga označi i zaključa. Nakon toga je na nadređenom da taj zadatak zatvori.



Slici 8.6.6: a) Zaključavanje

b) Zatvaranje

c) Vidljivost

Akcija 'Zatvori' otvara prozor prikazan na *Slici 8.6.6b*. U novootvorenom prozoru korisniku su ponuđeni zadaci koje je zadao, a netko od podređenih prethodno zaključao, te ih sada može označiti te pritiskom na tipku 'Zatvori' zatvoriti.

Akcija 'Vidljivost' otvara prozor prikazan na *Slici 8.6.6c*. Korisniku (administratoru ili šefu policije) su ponuđeni zadaci koje je prethodno zadao neki nadređeni, ali da bi se aktivirao, tj postao vidljiv podređenima kojima je namijenjen, administrator ga mora učiniti vidljivim klikom na tipku 'Vidljiv!'.

8.6.5 Novi zadatak

Svaki policijski službenik koji ima ovlasti nadređenog, može zadavati novi zadatak. To čini klikom miša na tipku 'Novi zadatak' s glavnog prozora prijavljenog korisnika, čime se otvara prozor prikazan na *Slici 8.6.7*.

Korisnik s popisa svojih podređenih odabire policijske službenike kojima želi zadati zadatak. Na desnoj strani prozora ispunjava podatke vezane uz zadatak. Početak izvršavanja zadatka može odabrati s kalendara prikazanog u klizećem

prozoru. Prioritet i povjerljivost kreću se u rasponu od 0-4. Upisivanjem manje ili veće vrijednosti korisniku se prikazuje poruka o krivom unosu. U prozor pod imenom 'tekst' korisnik unosi opis zadatka. Nakon što korisnik unese sve podatke o zadatku, klikom miša na tipku 'Napravi' novi zadatak je zadan i poslan administratoru i šefu policije na odobravanje.

Slici 8.6.7.: Zadavanje novog zadatka

Nakon što ga administrator ili šef policije učine vidljivim, zadatak će vidjeti i moći na njemu raditi svi odabrani podređeni. U slučaju da korisnik ne unese sve podatke o zadatku, sustav javlja pogrešku. Ako korisnik unese ime zadatka koji već postoji, novi zadatak se automatski preimenuje u odabrano ime sa sufiksom 1 (svaki novi zadatak tog istog imena dobiva još jednu jedinicu na kraju imena, npr: HAC, HAC1, HAC11...).

8.6.6 Digitalizacija

Digitalizaciju može obavljati samo tajnik. Digitalizacija se provodi pomoću prozora prikazanog na Slici 8.6.9..

Korisnik označava koji su sve policijski službenici sudjelovali na zadatku te unosi sve potrebne podatke. Početak i kraj izvršavanja zadatka može odabrati s kalendara prikazanog u klizećem prozoru. Korisnik također može dodati promjenu te označiti koji policijski službenici su sudjelovali u promjeni. Klikom miša na tipku 'Digitaliziraj' zadatak se pohranjuje.

The screenshot shows a software window titled "Digitaliziraj". On the left, a list of names is under the heading "bili na zadatku". In the center, there are input fields for "naziv", "pocetak", "kraj", "prioritet", "tajnost", "trajanje", and a "tekst" area. On the right, there is a "promjenu napravili" list, a "promjena" field, and a "Digitaliziraj" button at the bottom right.

bili na zadatku

- ☐ Martina Kocet
- ☒ Ante Kegelj
- ☒ Marin Biberovic
- ☐ Andrija Cajic
- ☐ Martina Jakovina
- ☐ Veseli Momar
- ☐ Super Cifrek
- ☐ Srebreško Posavec
- ☐ Renato Jovic

naziv
sudar

pocetak
2. 9. 1990

kraj
2. 9. 1990

dodaj promjenu

prioritet
1

tajnost
0

trajanje
1d

tekst
Na križanju Frankopanske i Ilice
sudarili se stojadin i fiček.

promjenu napravili
(prvi oznacen se evidentira)

- ☐ Martina Kocet
- ☐ Ante Kegelj
- ☒ Marin Biberovic
- ☐ Andrija Cajic
- ☐ Martina Jakovina
- ☐ Veseli Momar
- ☐ Super Cifrek

promjena
zastoj

Digitaliziraj

Slici 8.6.9.: Digitalizacija

9. Zaključci i smjernice za daljnji rad

Nakon puna četiri mjeseca zajedničkog rada, lijep je osjećaj osvrnuti se na napravljeno. Naravno da je bilo i nesuglasica, ali definitivno ostaju lijepa iskustva s projekta na predmetu Oblikovanje programske potpore. U godinama u kojima je nekad teško iskoordinirati samog sebe, naša voditeljica Petra je morala iskoordinirati osmero ljudi. Kako je svaki čovjek drukčiji, za svakoga je trebala drukčija taktika. Dok su neki dobrovoljno obavljali zadane zadatke, od drugih se rad mogao očekivati samo poslije prijetnji, a treći su sami sebi smišljali posao.

No, sve je to iza nas, bogatiji smo za važno iskustvo rada u timu, a svima je jasno da pravi rad u timu tek počinje kada se završi fakultet. Svi smo mi položili Baze podataka, Programiranje i programsko inženjerstvo te Algoritme i strukture podataka, ali rijetki su imali posla s “pravim” programima. Naše najvažnije stečeno znanje, osim naravno rada u timu, je iskustvo rada na nečemu opipljivom, od čega netko kasnije može imati koristi.

Kad bi nekim slučajem semestar trajao jedan ciklus duže, program bi dotjerali do savršenstva. Za početak, bolja preglednost zadataka je jedna od lako izvedivih promjena koja bi dodatno popravila ukupni dojam programa. Među smjernicama za budući rad je i omogućivanje službenicima da poruke šalju svojim nadređenima i podređenima, ako bi to imalo za rezultat efikasniju komunikaciju između službenika u policijskim postajama. Također, trenutno se promjene na zadacima i slične vide tek nakon što se korisnik ponovno ulogira na sustav, što je i više nego prihvatljivo s obzirom na trenutne uvjete, ali bi se definitivno na tome poradilo kad bi vremena bilo više. Sustav bi se mogao proširiti i pregledom nad većom količinom podataka vezanih uz policijske službenike, ili mogućnošću da se ti podaci promjene, pa čak i dodaju novi službenici.

Sve u svemu, velika lekcija u daljnjem radu je da ljudi nisu strojevi, te da 8 najboljih programera na svijetu neće kvalitetno odraditi zadatak ako nema dobre atmosfere i kvalitetne komunikacije među članovima. Shvatili smo kako stvari koje se uče na “društvenim” predmetima i nisu tako bespotrebne kao što smo bili uvjereni.

10. Reference

- ¹ Oblikovanje programske potpore, FER ZEMRIS, <http://www.fer.hr/predmet/opp>
- ² Oblikovanje programske potpore, FER ZEMRIS,
<http://www.zemris.fer.hr/predmeti/opp> -predavanja
- ³ <http://argouml-downloads.tigris.org/nonav/argouml-0.28/quickguide-0.28.pdf> -
ArgoUML brzi vodič
- ⁴ <http://argouml-downloads.tigris.org/nonav/argouml-0.28/manual-0.28.pdf> -
ArgoUML korisnički priručnik
- ⁵ 06a_UML_komplet.ppt → Software architecture and the UML (prikaznice)
- ⁶ SQL in Delphi, <http://delphi.about.com/od/database/l/aa060199.htm>
- ⁷ SQL pisanje upita, Wikipedia, <http://en.wikipedia.org/wiki/SQL>

Dodatak A: Dnevnik sastajanja

1.ciklus:

Utorak, 22.09.2009., 11.00-11.45h

Prisutni: Mirko Jambrošić, Martina Jakovina, Andrija Čajić, Ante Kegalj, Marin Biberović, Petra Zadro, Martina Kocet, Filip Lasić.

Sadržaj sastanka:

Svi članovi grupe su se upoznali, analiziran je zadatak i predložena su okvirna rješenja i područja rada pojedinih članova na cjelokupnom projektu.

Srijeda, 23.09.2009., 14.00-14.20h

Prisutni: Petra Zadro, Mirko Jambrošić, Andrija Čajić, Ante Kegalj.

Sadržaj sastanka:

Razgovor s asistentom, raspitivanje oko detalja projekta, rješavanje nejasnoća.

Četvrtak, 24.09.2009., 12.00-13.30h

Prisutni: Petra Zadro, Mirko Jambrošić, Martina Jakovina, Andrija Čajić, Ante Kegalj, Marin Biberović.

Sadržaj sastanka:

Razrađen je opis projekta i dogovoreni su poslovi koje treba obaviti tijekom prvog ciklusa:

Mirko Jambrošić – Razrada plana za programsko ostvarenje zadatka.

Marin Biberović – Detaljni opis projektnog zadatka.

Martina Kocet – Pisanje funkcionalnih i nefunkcionalnih zahtjeva.

Andrija Čajić – Razrada i pisanje obrazaca uporabe.

Filip Lasić – Razrada i pisanje obrazaca uporabe.

Ante Kegalj – Izrada plana rada za projekt.

Martina Jakovina – Izrada use case i sekvencijskih dijagrama.

Petra Zadro – Koordinacija grupe, vođenje dnevnika, pomoć pri izradi dijagrama.

Ponedjeljak, 28.09.2009., 17.00-18.00h

Prisutni: Mirko Jambrošić, Martina Jakovina, Petra Zadro, Martina Kocet, Filip Lasić.

Sadržaj sastanka:

Detaljno objašnjeni obrasci uporabe i ostvarenje komunikacije u sustavu. Uvid u obavljene zadatke.

Izvješće o napretku i budući rad:

Mirko Jambrošić – Napravio ideju za komunikaciju u sustavu.

Martina Kocet – Napravila kostur zahtjeva.

Martina Jakovina – Proučila dijagrame i nacrtat će ih nakon što se razrade obrasci uporabe .

Petra Zadro – Napisala ukratko obrasce uporabe i pomoći će pri izradi dijagrama

Filip Lasić – Proučio obrasce uporabe, u sljedeća 3 dana će ih detaljno opisati i poslati svima na uvid.

Petak, 2.10.2009., 17.00-18.15h

Prisutni: Mirko Jambrošić, Petra Zadro, Martina Jakovina, Andrija Čajić, Ante Kegalj, Marin Biberović.

Sadržaj sastanka:

Pregledana je napisana dokumentacija, dogovoreni su završni poslovi i demonstriran rad u Delphiju.

Izvešće o napretku i budući rad:

Mirko Jambrošić – Prikaz rada u Delphiju.

Martina Jakovina – Napravila i pojasnila dio dijagrama.

Petra Zadro – Pomogla oko dijagrama te uredila dokumentaciju.

Andrija Čajić – Dovršio obrasce uporabe.

Ante Kegalj – Predstavio plan rada.

Marin Biberović – Pomoć oko nadopunjavanja preostalih dijelova dokumentacije.

2.ciklus:

Petak, 6.11.2009., 12.00-12.30h

Prisutni: Mirko Jambrošić, Martina Jakovina, Petra Zadro, Martina Kocet, Filip Lasić, Marin Biberović, Andrija Čajić, Ante Kegalj

Sadržaj sastanka:

Pregledavanje uputa za drugu reviziju, dogovaranje oko podjele posla, izgleda baze.

Utorak, 10.11.2009., 17.00-19.00h

Prisutni: Mirko Jambrošić, Martina Jakovina, Petra Zadro, Martina Kocet, Marin Biberović, Ante Kegalj

Sadržaj sastanka:

Proučavanje class dijagrama, detalji oko razumijevanja druge revizije, objašnjavanje okvirnog izgleda programa.

Izvešće o napretku i budući rad:

Mirko Jambrošić – Radio na implementaciji, prijedlozi oko baze, proučavanje class dijagrama.

Martina Kocet – Rad na bazi, proučavanje dijagrama (aktivnosti i kolaboracijski)

Martina Jakovina – Proučava teoriju za pisanje detalja o arhitekturi, dijagram stanja

Marin Biberović – Proučavanje dijagrama (komponenti i objekata), crtanje.

Ante Kegalić – Skupljanje natuknica oko detalja implementacije.

Petra Zadro – Rad na bazi, pregled potrebnih stvari za drugu reviziju.

Srijeda, 11.11.2009., 14.00-14.15h

Prisutni: Petra Zadro, Mirko Jambrošić, Marin Biberović, Martina Kocet, Martina Jakovina.

Sadržaj sastanka:

Razgovor s asistentom, pojašnjavanje detalja oko pisanja druge revizije i crtanje dijagrama, pregled prijedloga class dijagrama.

Četvrtak, 12.11.2009., 16.00-18.00h

Prisutni: Petra Zadro, Mirko Jambrošić, Marin Biberović, Martina Kocet, Martina Jakovina.

Sadržaj sastanka:

Zajednički rad na dijagramima.

3.ciklus:

Utorak, 15.12.2009., 14.00-16.00h

Prisutni: Mirko Jambrošić, Martina Jakovina, Petra Zadro, Martina Kocet, Marin Biberović

Sadržaj sastanka:

Rad na upitima potrebnim za program, dogovor oko nekih detalja vezanih uz program, te uz podjelu posla oko dokumentacije. Podjela poslije kasnije napravljena putem interneta:

Mirko Jambrošić – implementacija

Martina Kocet – sređivanje instalacije, upute za instalaciju

Martina Jakovina – korisničke upute

Petra Zadro – korisničke upute, pregled stare verzije revizije

Marin Biberović – dijagram razmještaja, opis osnovne funkcionalnosti programa

Utorak, 5.1.2010., 12.00-14.00h

Prisutni: Mirko Jambrošić, Martina Jakovina, Petra Zadro, Marin Biberović, Ante Kegalj

Sadržaj sastanka:

Proučavanje rada programa, testiranje, preciznije dogovaranje posla, rješavanje nejasnoća

Petak, 8.1.2010., 12.00-12.30h

Prisutni: Petra Zadro, Mirko Jambrošić, Marin Biberović, Martina Kocet, Martina Jakovina, Ante Kegalj, Andrija Čajić, Filip Lasić

Sadržaj sastanka:

Pregled napravljenog posla oko dokumentacije, dogovor oko zadnjih detalja u sređivanju dokumentacije prije predaje projekta.