

**UNIVERZITET U BEOGRADU
FAKULTET ORGANIZACIONIH NAUKA**

MILAN V. MILANOVIĆ

**INFORMACIONI SISTEM ZA PRAĆENJE DOKUMENATA U
J2EE OKRUŽENJU**

SPECIJALISTIČKI RAD

Beograd, 2005.

MENTOR:

Dr Siniša Vlajić
docent FON-a

ČLANOVI KOMISIJE:

Dr Božidar Radenković
redovni profesor FON-a

Dr Dragana Bečejski-Vujaklija
vanredni profesor FON-a

Datum odbrane: _____

Informacioni sistem za praćenje dokumenata u J2EE okruženju

Sadržaj:

U ovom specijalističkom radu je prikazan razvoj informacionog sistema za praćenje dokumenata u J2EE okruženju. Opisan je razvoj sistema putem jedinstvenog procesa razvoja softvera i uzora u projektovanju softvera i prikazane sve specifičnosti kod jednog ovakvog sistema.

Detaljno su opisani korisnički zahtevi, kao i analiza, projektovanje i implementacija tih zahteva u obliku softverskog rešenja. Ovakvo rešenje prati proces tehničke dokumentacije u okviru jedne organizacije. Dat je i prikaz korišćenja pojedinih tehnologija koje su u javnom vlasništvu.

U okviru rada je urađena detaljna specifikacija i implementacija dokumentacionog procesa, koji se odvija u lokalnoj računarskoj mreži i preko Interneta, kao i način korišćenja sistema od strane korisnika.

Ključne reči:

Informacioni sistemi, proces dokumentacije, J2EE, jedinstveni proces razvoja softvera, uzori

Information system for managing documentation in J2EE environment

Abstract:

This work presents development of information system for managing documentation process in J2EE environment. System development through Unified Software Development Process and using software patterns have been described in this work.

User requirements have been described in detail, also as analyse, design and implementation of that requirements in form of software solution. This solution trace process of technical documentation in one organisation. Description of some open-source technologies is also been given here.

In the scope of this work, specification and implementation has been done of documentation process that work in local area network and over the Internet, likewise his use from the viewpoint of users.

Key words:

Information systems, documentation process, J2EE, unified software development process, patterns

SADRŽAJ

1. UVOD.....	1
1.1. JAVA	1
1.2. J2EE ARHITEKTURA	1
1.3. JEDINSTVENI PROCES RAZVOJA SOFTVERA (JPRS)	2
1.4. JAKARTA STRUTS FRAMEWORK	3
1.5. UZORI (PATERNI)	4
2. ARCHIDOC WEB DOCUMENT MANAGEMENT SYSTEM	5
2.1. PRIKUPLJANJE ZAHTEVA OD KORISNIKA	5
2.2. ANALIZA	28
2.2.1. <i>Dijagrami sekvenci slučajeva korišćenja</i>	28
2.2.2. <i>Rezultat analize sistemskog dijagrama sekvenci (SO)</i>	56
2.2.3. <i>Definisanje ugovora o sistemskim operacijama</i>	56
2.2.4. <i>Domenski model sa tipiziranim atributima</i>	62
2.2.5. <i>Relacioni model</i>	63
2.2.5. <i>Dijagram klasa strukture</i>	64
2.2.6. <i>Klase strukture</i>	65
2.3. PROJEKTOVANJE	69
2.3.1. <i>Faza projektovanja sistema</i>	69
2.3.2. <i>Arhitektura softverskog sistema</i>	70
2.3.3. <i>Struktura softverskog sistema</i>	72
2.3.3. <i>Ponašanje softverskog sistema</i>	99
2.3.4. <i>Opis arhitekture iz perspektive modela projektovanja</i>	158
2.3.5. <i>Modeli raspoređivanja</i>	162
2.4. IMPLEMENTACIJA	163
2.4.1. <i>Komponente</i>	163
2.4.2. <i>Redosled kompajliranja komponenti</i>	163
2.4.3. <i>Instalacija sistema i pokretanje</i>	163
2.4.4. <i>Veličina sistema</i>	164
2.5. TESTIRANJE	164
3. ZAKLJUČAK	165
4. LITERATURA.....	166

1. UVOD

1.1. JAVA

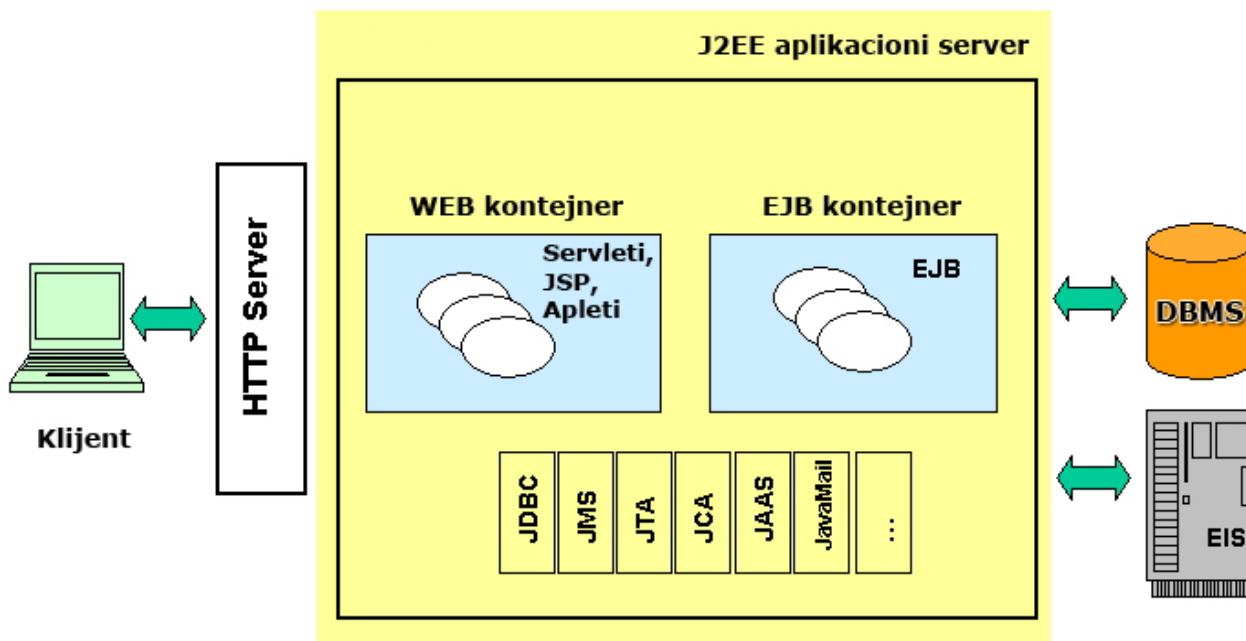
Kad se koristi pojam "Java" tu se govori o nekoliko različitih stvari, zavisno od konteksta u kojem se priča. U osnovnom značenju Java je platformski nezavisan, interpreterski programski jezik razvijen od strane kompanije Sun Microsystems. Ovo je jednoznačan pojam s obzirom da označava jedinstvenu specifikaciju. Kada se govori o izvršavanju Java programa na određenoj platformi, odnosno računaru, govori se o Java virtuelnoj mašini (eng. Java Virtual Machine ili kraće JVM). Java virtuelna mašina predstavlja hardversku ili softversku implementacija jedinstvene specifikacije. Uloga Java virtuelne mašine jeste da interpretira Java bytekod (eng. bytecode) prelazni oblik Java kôda koji se može izvršavati na svim implementacijama Java virtuelne mašine. Prvobitni povod nastanka Jave bili su aparati za domaćinstvo. Java je donela jedno značajno poboljšanje u odnosu na prethodne jezike (kao što su C i C++) jer oni nisu omogućavali platformsku nezavisnost. Prva verzija 1.0 je objavljena 1995. godine. Postoji čitav niz implementacija Java virtuelne mašine od različitih proizvođača. Iako se gotovo sve implementacije pridržavaju specifikacije, ja sam odlučio ograničiti izbor na rad sa standardnom implementacijom Java virtuelne mašine od kompanije Sun Microsystems preciznije JDK 1.5. Da je Java dobar izbor govori i činjenica da već od 2003. preko 75% profesionalnih programera koristi Javu kao primarni programski jezik, a 2005. približno 4.5 miliona programera radi na Javi, dok postoji preko 2.5 milijarde uređaja sa ugrađenom Java tehnologijom. Konsultantska kuća TIOBE koja svakog meseca objavljuje indeks najpopularnijih programskih jezika je objavila u avgustu 2005. da je Java posle 10 meseci stagniranja na drugom mestu izbila na prvo mesto, što se i može videti na slici 1.

Position	(Position)	Programming Language	Ratings	(Ratings)	Status
1	=	Java	21.206%	+4.21%	A
2	=	C	19.473%	+3.14%	A
3	=	C++	10.514%	-4.79%	A
4	↑↑	Perl	9.884%	+1.44%	A
5	↓	PHP	7.873%	-2.55%	A
6	↓	(Visual) Basic	7.757%	-2.38%	A
7	↑↑↑	C#	3.485%	+1.85%	A
8	=	Python	2.883%	-1.82%	A
9	=	JavaScript	1.921%	+0.24%	A
10	↓↓↓	Delphi/Kylix	1.763%	-3.05%	A

Slika 1: Popularnost programskih jezika (Izvor: TIOBE Software BV)

1.2. J2EE ARHITEKTURA

J2EE tehnologija predstavlja specifikaciju za razvoj Enterprise aplikacija. Osnovnu strukturu ove arhitekture čine: Servleti, JSP strane i EJB-ovi. JSP-ovi, servleti i apleti su zaduženi za komunikaciju sa klijentom, dok su EJB-ovi zaduženi za onaj deo poslovne logike aplikacije, koja nije ugrađena u aplikacioni server, kao i za komunikaciju sa bazom.



Slika 2: J2EE arhitektura

Sam aplikacioni server se sastoji iz: Home i Remote interfejsa, Enterprise Java Bean-ova, konteksta i kontejnera. Proces komunikacija između ovih komponenti je sledeći: klijent prvo poziva metodu remote ili home interfejsa EJB-a. Kontejner presreće poziv klijenta i proverava da li je poziv u skladu sa semantikom iz deployment deskriptora. Ako je poziv validan on ga dalje usmerava ka odgovarajućem EJB-u, koji kasnije komunicira sa bazom podataka. Komunikacija sa bazom može biti izvršena i preko kontejner elementa. Kontejner obezbeđuje okruženje na kojem se izvršavaju EJB-ovi. Kontekst element se pravi za svaki EJB, a on sadrži informacije o klijentu, kontejneru i o samom EJB-u. Enterprise Java Bean predstavlja arhitekturu za razvoj složenih aplikacija koje su zasnovane na softverskim komponentama. On se sastoji iz tri dela: EJBHome interfejsa koji je zadužen za operacije: create(), find() i remove(), EJBObject interfejsa koji je zadužen za ostale operacije klijenta i klase koja implementira EJB interfejse. Postoje tri vrste EJB-ova: Session bean, Entity bean i Message-driven bean. Session bean je zadužen za komunikaciju između klijenta i server i za poslovna pravila aplikacija. Entity bean je odgovoran za predstavljanje poslovnih podataka aplikacije i komunikaciju sa bazom podataka. Message-driven bean je zadužen za rad kod kojeg nije potreban trenutni odgovor korisniku. Kod Entity bean-ova postoje dva nivoa perzistentnosti: Container managed Persistent Entity Bean, kod koga se perzistentnost implementira unutar kontejnera i Bean Managed Persistent Entity Bean, kod koga se perzistentnost implementira unutar kontejner elemenata. Za ovaj softverski sistem će biti korišćen Sun Java Application Server 8.1.

1.3. JEDINSTVENI PROCES RAZVOJA SOFTVERA (JPRS)

Jedinstveni proces razvoja softvera [Jacobson1999] predstavlja proces razvoja koji se sastoji iz niza aktivnosti koje su potrebne da se transformišu zahtevi u softverski sistem. On predstavlja okvir za generički proces koji može biti specijalizovan za veliku klasu softverskih sistema. Zasnovan je na komponentama koje su između sebe povezane interfejsom. Njegove osnovne osobine su:

- a) Da se sa njim upravlja na osnovu slučajeva korišćenja;

Slučaj korišćenja (SK) predstavlja deo funkcionalnosti sistema koji korisniku daje neki rezultat, što znači da on opisuje funkcionalne zahteve. Svi slučajevi korišćenja zajedno čine model SK koji opisuje kompletnu funkcionalnost sistema. Ovaj model treba da odgovori na pitanje: Šta sistema treba da radi za svakog korisnika.

- b) Da je usmeravan arhitekturom;

Arhitektura softverskog sistema obuhvata najvažnije statičke i dinamičke aspekte sistema, kao što su: platforme na kojima se izvršava softver, komponente, "legacy" sisteme, itd. Veza između SK i arhitekture je u tome što svaki proizvod ima funkciju i oblik koji moraju biti između sebe povezani. Funkcija korespondira sa slučajem korišćenja, dok oblik saraduje sa arhitekturom. SK moraju, kada se realizuju, da se ugrade u arhitekturu. Oni se razvijaju paralelno.

- c) Da se sastoji iz iteracija, koje kao rezultat daju povećanje svojstva sistema.

Razvoj sistema se sastoji iz više mini projekata, a gde svaki mini projekat predstavlja iteraciju koja rezultuje inkrementom. Mini projekat prolazi kroz faze: analize, projektovanja, implementacije i testiranja.

JPRS se sastoji iz nekoliko ciklusa gde se posle svakog izvršenog ciklusa dobija softverski proizvod. Svaki od tih ciklusa se sastoji iz četiri faze: početak, razvoj, građenje i prelaz. Početna faza definiše širinu projekta i krajnju viziju proizvoda. Tu se kreiraju slučajevi korišćenja. Tokom razvojne faze pravi se plan projekta, slučajevi korišćenja se razrađuju i daje se nacrt arhitekture sistema. U fazi građenja se dobija kompletan softver koji se pridružuje do arhitekture sistema i tu se dolazi do beta verzije softvera. U fazi prelaza se beta verzija prosleđuje korisniku radi testiranja. Nakon ispravke grešaka se prelazi u finalnu verziju proizvoda (softvera).

1.4. JAKARTA STRUTS FRAMEWORK

Za razvoj klijentskog dela ove aplikacije je korišćena podrška Struts framework-a. Struts predstavlja open source framework (okvir) za izgradnju web aplikacija. Jezgro Struts framework-a se sastoji od fleksibilnih kontrolisanih nivoa baziranih na standardnim tehnologijama, kao što su: Java Servleti, JavaBean-ovi, resursi i XML, kao i razni Jakarta Commons paketi. Struts podržava arhitekturu aplikacija baziranu na principu modela 2, koji je jedna verzija klasičnog modela Model-View-Controller (MVC) paterna (uzora). MVC uzor organizuje aplikaciju u tri odvojena modula: jedan za aplikacioni model sa prikazom podataka i biznis logikom, drugi za prezentaciju podataka korisniku i unos podataka, i treći za kontroler koji prosleđuje zahteve i izvršava tok. Većina web aplikacija koriste framework-e sa nekom varijacijom ovog uzora. MVC uzor pruža nekoliko pogodnosti: on odvaja delove sistema (perzistentnost podataka i ponašanje, prezentacija i kontrola), smanjuje ponavljanje koda, centralizuje kontrolu i pravi aplikaciju tako da može lako da se modifikuje. Struts pruža svoju sopstvenu kontroler (Controller) komponentu i integriše je sa drugim tehnologijama da bi pružio Model i View. Za model, Struts može raditi sa standardnim setom tehnologija za pristup podacima, kao što su: JDBC i EJB, kao i većinom paketa drugih kompanija: Hibernate, iBATIS ili Object Relational Bridge. Za View, Struts radi dobro sa JavaServer Pages stranicama, uključujući JSTL i JSF, kao i Velocity Template, XSLT i druge prezentacione sisteme. Struts framework pruža nevidljivu podršku koja svaka profesionalna web aplikacija mora da ima. Struts je projekat Apache Software fondacije i održava se i razvija volonterski.

1.5. UZORI (PATERNI)

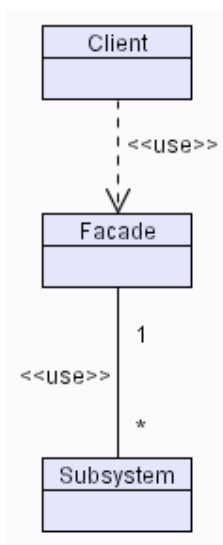
Definicija uzora [Coplien2000] je: Uzor je pravilo za građenje stvari, ali je ono istovremeno i sama stvar. Osnove uzora je napisao Kristofer Aleksander, a primanjivao ih je na projektovanje arhitekture u građevinarstvu. Svaki uzor imma svoj oblik predstavljanja koji pomaže da se shvati: koji problem se razmatra, kontekst problema i objašnjenje rešenja problema. Uzori se obično predstavljaju preko skupa sekcija: ime (važno je da ukaže da svrhu uzora), svrha (frazza koja sumarno objašnjava šta radi uzor), problem (šta uzor rešeava), kontekst (okruženje uzora), sile (teškoće i ograničenja problema), rešenja (potrebno je da bude dovoljno dobro), skica (predstavlja suštinu uzora) i rezultujući kontekst (predstavlja okruženje rešenja). Postoje tri oblika predstavljanja uzora:

- a) GOF oblik (Gang of four);
- b) Aleksanderov oblik;
- c) Copelinov oblik.

Uzori se mogu klasifikovati na: tronivojske uzore (tri nivoa apstrakcije), Aleksanderovo skaliranje, drugi skalirajući pristupi, anti-uzori i meta-uzori. Kod Larmaneove metode (Jedinstvene metode razvoja softvera) koja će biti korišćena u ovom radu, uzori se primenjuju u fazi projektovanja, u toku kreiranja dijagrama saradnje, kada se odgovornosti dodeljuju do objekata i kada se projektuje saradnja između objekata. Dijagrami se prave na osnovu sistemskih operacija (ugovora) koji su definisani u fazi analize. Larmanova definicija uzora [Larman2000] je sledeća: Uzori su imenovani parovi problemi/rešenja, koji definišu savete i principe kod dodeljivanja odgovornosti objektima. Neki GRASP (General Responsibility Assignment Software Patterns) uzori su sledeći:

- Kontroler uzor,
- Kreator uzor,
- Uzor visoke kohezije,
- Uzor slabe povezanosti,
- Uzor indirekcije, ...

Primer jednog uzora (fasada – facade) je prikazan na slici ispod.



Slika 3: Facade (fasada) uzor

2. ARCHIDOC WEB DOCUMENT MANAGEMENT SYSTEM

2.1. PRIKUPLJANJE ZAHTEVA OD KORISNIKA

Prikupljanje zahteva od korisnika se odvija na sledeći način: prvo se pronalaze aktori u sistemu tako što se uzima kao osnova realni sistem iz koje se pravi model slučaja korišćenja softverskog sistema a te slučajeve izvršavaju radnici. Nakon toga je potrebno odrediti prioritetne slučajeve korišćenja, odnosno one koji treba da se razviju u prvim iteracijama. Za svaki slučaj korišćenja je potrebno opisati tog događaja, što uključuje početak i kraj slučaja korišćenja, a i njegovu interakciju sa aktorima. Prototip korisničkog interfejsa se pravi za svaki slučaj korišćenja. I na kraju se radi strukturiranje modela slučaja korišćenja kako bi se dobili detaljni slučajeve korišćenja čije ponašanje mogu korisiti drugi slučajeve korišćenja, i izveli opcioni slučajeve korišćenja.

ARCHIDOC 1: *Korisnički zahtev*

Omogućiti arhiviranje, pretragu i pregled dokumentacija u elektronskom obliku za razne vrste dokumenata i crteža. Dokumenti mogu pripadati objektima koji takođe mogu pripadati i biti deo drugih objekata (hijerarhijski), kao i grupama dokumenata (projekat, elaborat, ...). Uočeno je takođe da korisnici trebaju imati mogućnost komuniciranja između sebe kroz sistem jer postoji potreba stalne saradnje između arhive i autora dokumenata. Potrebno je omogućiti i skladištenje dokumentacije na trajne medijume. Korisnici bi trebalo da imaju restriktivan pristup sistemu u zavisnosti od statusa u sistemu. Postoji više verzija dokumenata i revizija, kao i poreklo (Unutrašnje, Spoljno). Kod grupa dokumenata (Projekata, Elaborata, ...) postoji revizija kao i kod dokumenta, investitor i vrsta te grupe dokumenta, kao i autori istih. Objekat ima svoju lokaciju, vrstu, tip i opciono firmu kojoj pripada. Objekat je realni objekat. Proces dokumentacije ide tako što autori dokumenata šalju dokumente u arhivu, a Arhivar ih unosi u postojeću dokumentaciju i veže za objekte i grupe dokumenata, ako je to potrebno.

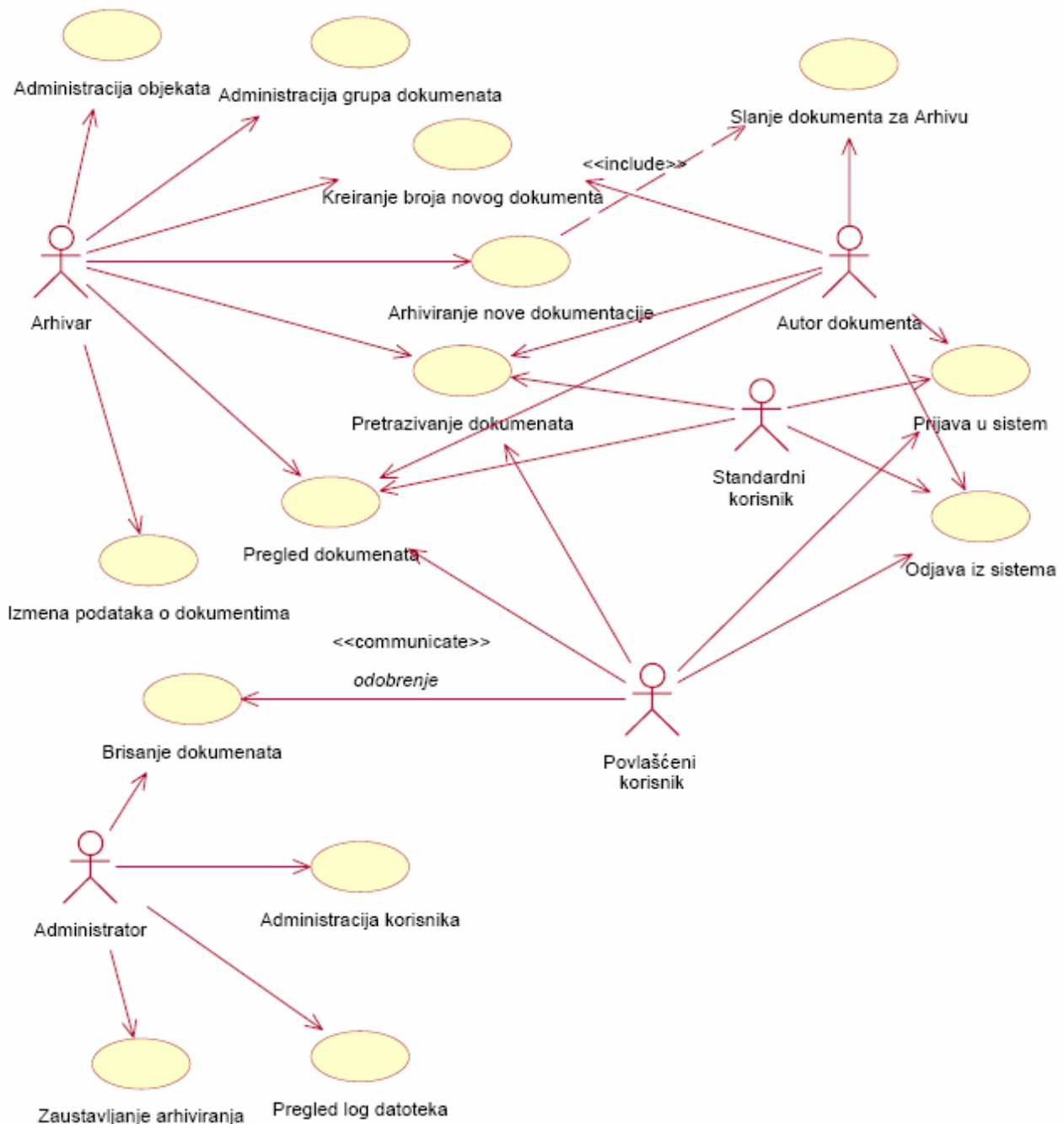
ARCHIDOC 2: *Realni poslovni sistem*

S obzirom na ovako definisan korisnički zahtev odavde se mogu uočiti sledeći poslovni procesi:

- Arhiviranje nove dokumentacije;
- Pretraživanje dokumenata;
- Pregled dokumenta;
- Izmena podataka o dokumentu;
- Administracija objekata;
- Administracija grupa dokumenata;
- Kreiranje broja novog dokumenta;
- Slanje dokumenta u arhivu;
- Brisanje dokumenta;
- Zaustavljanje arhiviranja;
- Administracija korisnika;
- Pregled log datoteka;
- Prijava u sistem;
- Odjava iz sistema.

ARCHIDOC 3: Model slučaja korišćenja

Realni poslovni model je prikazan na slici 4 i modeliran je pomoću dijagrama slučaja korišćenja:



Slika 4: Poslovni model sistema predstavljen preko dijagrama slučaja korišćenja

Ovaj dijagram slučaja korišćenja se sastoji iz četrnaest poslovnih slučaja korišćenja: Arhiviranje nove dokumentacije, Pretraživanje dokumenata, Pregled dokumenta, Izmena podataka o dokumentu, Administracija objekata, Administracija grupa dokumenata, Kreiranje broja novog dokumenta, Slanje dokumenta u arhivu, Brisanje dokumenta, Zaustavljanje arhiviranja, Administracija korisnika, Pregled log datoteka, Prijava u sistem i Odjava iz sistema. A sadrži i pet aktora: Arhivara, Autora dokumenta, Standardnog korisnika, Povlašćenog korisnika i Administratora.

SK1: Slučaj korišćenja - Arhiviranje nove dokumentacije

Naziv SK

Arhiviranje novog dokumenta

Aktori SK

Arhivar dokumenta

Učesnici SK

Arhivar i Sistem

Preduslov

Sistem je uključen i arhivar je prijavljen pod svojom šifrom. Arhivar je prethodno, iz glavnog menija izabrao opciju "Arhiviranje dokumenata" i pokreće se forma za arhiviranje dokumenata, a sistem je prikazuje (kada arhivar pozove izvršenje za ovu formu, sistem prikazuje sve dokumente koji čekaju na arhiviranje).

Kratak opis SK

Ovaj use case se dešava kada Arhivar unosi novi dokument u arhivu.

Osnovni scenario SK

1. Arhivar poziva sistem da prikaže dokumente koji čekaju arhiviranje.
2. Arhivar vrši izbor dokumenta koji hoće da arhivira.
3. Sistem prikazuje arhivaru sve podatke o dokumentu koje je uneo autor.
4. Arhivar može da veže dokument za objekte ili grupe dokumenata.
5. Arhivar kontroliše unete podatke.
6. Arhivar poziva sistem da prihvati podatke o dokumentu sa vezama.
7. Sistem proverava unete podatke.
8. Sistem prihvata unete podatke.
9. Sistem prikazuje poruku Arhivaru o uspešnosti arhiviranja i upisuje tu informaciju u log datoteku.

Postuslov

Ovaj slučaj korišćenja se završava kada je Arhivar primio poruku o uspešnosti arhiviranja dokumenta.

Alternativna scenarija

- 1.1. Ako je administrator zabranio dodavanje dokumenata, arhivar neće moći arhivirati dokument.
- 4.1. Ako objekat ili grupa dokumenata ne postoji Arhivar će dobiti poruku o grešci.
- 6.1. Ako autor dokumenta nije uneo sve informacije o vrsti dokumenta ili o samom dokumentu, prekinuće se izvršavanje scenarija.
- 7.1. Ako dokument ili informacije o dokumentu nisu u redu Arhivar Autoru vraća dokument sa porukom zašto je vraćen isti.

Opis interakcija sistema sa aktorima

1. Arhivar traži od sistema da arhivira dokument.
2. Sistem arhivira dokument i prikazuje poruku.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) dokument koji se arhivira

2. Vrednosti sistema: Arhivirani dokument

Odgovornost učesnika

1. Arhivar bira, povezuje i arhivira dokument.

SK2: Slučaj korišćenja – Pretraživanje dokumenata

Naziv SK

Pretraživanje dokumenata

Aktori SK

Arhivar, Standardni korisnik, Autor, Povlašćeni korisnik i Administrator.

Učesnici SK

Arhivar, Standardni korisnik, Autor, Povlašćeni korisnik, Administrator i Sistem.

Preduslov

Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Proširena pretraga", gde se pokreće se forma za pretragu dokumenata, a sistem je prikazuje ili je direktno ukucana reč u tekstualn polje za brzu pretragu.

Kratak opis SK

Ovaj use case se dešava kada korisnik pretražuje dokumente koji su već arhivirani.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže formu za pretragu dokumenata ili na glavnom ekranu unosi reč(i) za pretragu.
2. Korisnik vrši izbor opcija za pretragu dokumenata.
3. Korisnik poziva sistem da prihvati podatke za pretragu i da pretraži dokumente.
4. Sistem prihvata unete podatke i pretražuje dokumente.
5. Sistem prikazuje rezultate pretrage dokumenata (dokumente koje je pronašao ili poruku ako ni jedan dokument nije pronašao).

Postuslov

Ovaj slučaj korišćenja se završava kada Sistem na ekranu prikaže rezultate pretrage.

Alternativna scenarija

- 3.1. Ako korisnik nije uneo kriterijum pretrage sistem će prijaviti grešku i tražiti da se kriterijum ponovo unese.
- 4.1. U slučaju da ni jedan dokument ne zadovoljava postavljeni uslov sistem će prijaviti poruku o tome da nije mogao da pronađe traženi dokument.
- 5.1. Ako ne postoji ni jedan dokument u sistemu Korisniku će biti prikazana poruka o tome.

Opis interakcija sistema sa aktorima

1. Korisnik traži od sistema da pretraži dokumente.
2. Sistem pretražuje dokumente i prikazuje rezultat pretrage.

Opis korišćenih objekata sistema

1. Poslovni objekti

- a) dokumenti koji se pretražuju
- 2. Vrednosti sistema: nema

Odgovornost učesnika

1. Korisnik (koji u ovom slučaju može pripadati bilo kojoj grupi korisnika) unosi ključne reči za pretragu i opcije pretrage.

SK3: Slučaj korišćenja – Pregled dokumenta

Naziv SK

Pregled dokumenta

Aktori SK

Arhivar, Standardni korisnik, Autor, Povlašćeni korisnik i Administrator.

Učesnici SK

Arhivar, Standardni korisnik, Autor, Povlašćeni korisnik, Administrator i Sistem.

Preduslov

Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Dokumenti", "Objekti" ili "Grupe dokumenata", gde se pokreće se forma za prikaz dokumenata.

Kratak opis SK

Ovaj use case se dešava kada korisnik prikazuje dokument koji je već arhiviran.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže formu za pregled dokumenata.
2. Korisnik vrši izbor dokumenta iz liste dokumenata.
3. Sistem prikazuje detaljne informacije o sistemu i prikazuje opcije za rad sa dokumentom.

Postuslov

Ovaj slučaj korišćenja se završava kada Sistem na ekranu prikaže detalje o dokumentu.

Alternativna scenarija

- 3.1. Ako ne postoji dokument u sistemu Korisnik će dobiti poruku o tome.

Opis interakcija sistema sa aktorima

1. Korisnik traži od sistema da prikaže dokument.
2. Sistem prikazuje dokument i opcije za rad sa dokumentom.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) dokument koji se prikazuje
2. Vrednosti sistema: nema

Odgovornost učesnika

1. Korisnik (koji u ovom slučaju može pripadati bilo kojoj grupi korisnika) vrši izbor dokumenta za pregled.

SK4: Slučaj korišćenja – Izmena podataka o dokumentu**Naziv SK**

Izmena podataka o dokumentu

Aktori SK

Arhivar i Administrator.

Učesnici SK

Arhivar, Administrator i Sistem.

Preduslov

Sistem je uključen i korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Dokumenti", "Objekti" ili "Grupe dokumenata", gde se pokreće se forma za prikaz dokumenata.

Kratak opis SK

Ovaj use case se dešava kada korisnik menja podatke o dokumentu koji je već arhiviran.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže formu za pregled dokumenata.
2. Korisnik vrši izbor dokumenta iz liste dokumenata.
3. Sistem prikazuje detaljne informacije o sistemu i prikazuje opcije za rad sa dokumentom.
4. Iz ponuđenih opcija korisnika bira opciju "Izmeni dokument".
5. Sistem prikazuje formu koja sadrži sve informacije o dokumentu (Naziv, broj, ...).
6. Korisnik unos nove ili menja postojeće informacije.
7. Korisnik traži od sistema da prihvati unete (izmenjene) podatke.
8. Sistem proverava unete podatke.
9. Sistem prihvata unete podatke.
10. Sistem prikazuje poruku Korisniku o uspešnosti izmene dokumenta i upisuje tu informaciju u log datoteku.

Postuslov

Ovaj slučaj korišćenja se završava kada Sistem na ekranu prikaže poruku o uspešnosti izmene dokumenta ili kada Korisnik odustane od izmene.

Alternativna scenarija

- 4.1. Ako je Administrator zabranio izmenu dokumenata opcija za izbor "Izmeni dokument" neće biti prikazana.
- 8.1. Ako korisnik nije uneo korektno (tačno) informacije o dokumentu ili ako je napravio grešku kod izmene Sistem će prijaviti poruku o grešci.

Opis interakcija sistema sa aktorima

1. Korisnik traži od sistema da prikaže dokument.
2. Sistem prikazuje dokument i opcije za rad sa dokumentom.
3. Korisnik menja podatke o dokumentu.
4. Sistem prikazuje poruku o izmeni dokumenta.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) dokument koji se menja.
2. Vrednosti sistema: Izmenjeni dokument.

Odgovornost učesnika

1. Administrator ili Arhiva vrši izbor dokumenta i menja podatke o dokumentu.

SK5: Slučaj korišćenja – Administracija objekata

Naziv SK

Administracija objekata

Aktori SK

Arhivar i Administrator

Učesnici SK

Arhivar, Administrator i Sistem

Preduslov

Sistem je uključen i Korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Administracija objekata" i pokreće se forma za rad sa objektima, i sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Arhivar ili Administrator administrira objekte.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže objekte koji postoje u sistemu.
2. Korisnik kreira novi objekat ili menja postojeći objekat i unos informacije o objektu.
3. Sistem prikazuje Korisniku formu za kreiranje novog objekta.
4. Korisnik unos podatke i veže objekat za neke druge objekte (opciono).
5. Korisnik kontroliše unete podatke.
6. Korisnik poziva sistem da prihvati podatke o objektu.
7. Sistem proverava unete podatke.
8. Sistem prihvata unete podatke.
9. Sistem prikazuje poruku Korisniku o uspešnosti rada sa objektom i prikazuje sve objekte (koji pripadaju izabranom objektu).
10. Korisnik može izabrati objekat da vidi detalje o objektu.

Postuslov

Ovaj slučaj korišćenja se završava kada Korisnik primi poruku o uspešnosti rada sa objektom.

Alternativna scenarija

- 2.1. Ako ne postoji objekat u sistemu Sistem će prikazati poruku.
- 4.1. Ako Korisnik progrešno veže objekte Sistem će prijaviti grešku.
- 5.1. Ako je Korisnik uneo pogrešne podatke o objektu Sistem će prijaviti grešku.
- 5.2. Ako Korisnik pokuša da kreira objekat koji već postoji Sistem će prijaviti grešku.

Opis interakcija sistema sa aktorima

1. Korisnik bira već postojeći objekat ili kreira (menja) postojeći.
2. Sistem kreira (menja) objekat i prikazuje poruku.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) objekat koji se kreira (menja) ili pregleda.
2. Vrednosti sistema: Kreirani objekat.

Odgovornost učesnika

1. Korisnik kreira, menja ili pregleda objekat.

SK6: Slučaj korišćenja – Administracija grupa dokumenata

Naziv SK

Administracija grupa dokumenata

Aktori SK

Arhivar i Administrator

Učesnici SK

Arhivar, Administrator i Sistem

Preduslov

Sistem je uključen i Korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Administracija grupa dokumenata" i pokreće se forma za rad sa grupama dokumenata (Projekat, Elaborat, ...), i sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Arhivar ili Administrator administrira grupe dokumenata.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže vrste grupa dokumenata koji postoje u sistemu.
2. Korisnik bira vrstu grupe dokumenata za prikaz.
3. Korisnik kreira novu grupu dokumenata ili menja postojeću i unos informacije o njoj.
4. Sistem prikazuje Korisniku formu za kreiranje nove grupe dokumenata.
5. Korisnik unos podatke i veže grupu dokumenata za neke objekte (opciono).
6. Korisnik kontrolise unete podatke.
7. Korisnik poziva sistem da prihvati podatke o grupi dokumenata.
8. Sistem proverava unete podatke.
9. Sistem prihvata unete podatke.
10. Sistem prikazuje poruku Korisniku o uspešnosti rada sa grupom dokumenat i prikazuje sve grupe dokumenata za izabranu vrstu grupe dokumenata.
11. Korisnik može izabrati grupu dokumenata da vidi detalje o njoj.

Postuslov

Ovaj slučaj korišćenja se završava kada Korisnik primi poruku o uspešnosti rada sa grupom dokumenata.

Alternativna scenarija

- 1.1. Ako ne postoji ni jedna grupa dokumenata u sistemu biće prikazana poruka.
- 6.1. Ako je Korisnik uneo pogrešne podatke o grupi dokumenata Sistem će prijaviti grešku.

Opis interakcija sistema sa aktorima

1. Korisnik bira već postojeću grupu dokumenata ili kreira (menja) postojeću.
2. Sistem kreira (menja) grupu dokumenata i prikazuje poruku.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) grupa dokumenata koja se kreira (menja) ili pregleda.
2. Vrednosti sistema: Kreirana grupa dokumenata.

Odgovornost učesnika

1. Korisnik kreira, menja ili pregleda grupu dokumenata.

SK7: Slučaj korišćenja – Kreiranje broja novog dokumenta

Naziv SK

Kreiranje broja novog dokumenta

Aktori SK

Arhivar i Autor

Učesnici SK

Arhivar, Autor i Sistem

Preduslov

Sistem je uključen i Korisnik je prijavljen pod svojom šifrom. Korisnik je prethodno, iz glavnog menija izabrao opciju "Slanje poruke" i pokreće se forma za slanje poruka, i sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Autor traži od Arhivara broj za novi dokument.

Osnovni scenario SK

1. Autor poziva sistem da prikaže formu za slanje poruke.
2. Autor popunjava formu i šalje je Arhivaru.
3. Sistem prikazuje Autoru poruku o uspešnosti slanja poruke.
4. Arhivar poziva formu za pregled dobijenih poruka.
5. Vrši izbor nepročitane poruke za pregled.
6. Arhivar preuzima dobijenu poruku od sistema.
7. Sistem prikazuje Arhivaru poruku za novi broj dokumenta koju je dobio od Autora.
8. Arhivar odgovara na poruku tako što šalje Autoru novi broj dokumenta.
9. Sistem proverava unete podatke.
10. Sistem prihvata unete podatke.
11. Sistem prikazuje poruku Arhivaru o uspešnosti slanja poruke.
12. Autor dobija poruku koja sadrži novi broj dokumenta.

Postuslov

Ovaj slučaj korišćenja se završava kada Autor želi da uzme novi broj dokumenta za onaj koji upravo kreira.

Alternativna scenarija

- 2.1. Ako je Autor izabrao pogrešnog korisnika (umesto Arhivara) desiće se poruka o grešci.
- 2.2. Ako Autor nije poslao ispravno poruku Arhivar neće primiti poruku, i on se dobiti poruku o grešci.
- 8.1. Ako Arhivar nije uneo ispravno podatke Sistem će prijaviti grešku.

Opis interakcija sistema sa aktorima

1. Autor šalje poruku Arhivaru kroz Sistem.
2. Sistem prikazuje poruku od Autora Arhivaru čim se prijavi u sistem.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) poruka koja se šalje.
2. Vrednosti sistema: Poslata poruka.

Odgovornost učesnika

1. Autor šalje poruku u kojoj traži novi broj dokumenta.
2. Arhivar prima poruku od Autora i odgovara na nju sa novim brojem dokumenta.

SK8: Slučaj korišćenja – Slanje dokumenta u arhivu

Naziv SK

Slanje dokumenta u arhivu

Aktori SK

Autor

Učesnici SK

Autor i Sistem

Preduslov

Sistem je uključen i Autor je prijavljen pod svojom šifrom. Autor je prethodno, iz glavnog menija izabrao opciju "Slanje dokumenta" i pokreće se forma za slanje dokumenta u arhivu, i sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Autor šalje dokument u arhivu na arhiviranje.

Osnovni scenario SK

1. Autor poziva sistem da prikaže formu za slanje dokumenta.
2. Sistem prikazuje Autoru formu za slanje dokumenta (za izbor datoteke).
3. Autor vrši izbor datoteke na lokalnom (mrežnom) disku.
4. Sistem proverava da li datoteka postoji.
5. Sistem prihvata datoteku.
6. Sistem prikazuje formu za unos podataka (detalja) o dokumentu.
7. Autor unos podatke o dokumentu.
8. Autor poziva Sistem da prihvati podatke o dokumentu.

9. Sistem proverava unete podatke.
10. Sistem prihvata unete podatke.
11. Sistem prikazuje poruku Autoru o uspešnosti slanja dokumenta.

Postuslov

Ovaj slučaj korišćenja se završava kada Autor dobije poruku od Sistema da je dokument otišao u arhivu.

Alternativna scenarija

- 4.1. Ako je Autor izabrao datoteku koju je neko u međuvremenu izbrisao scenario će se zaustaviti.
- 9.1. Ako Autor nije uneo sve potrebne i tačno podatke o dokumentu Sistem će prijaviti grešku i ispisati poruku.

Opis interakcija sistema sa aktorima

1. Autor šalje dokument u arhivu putem Sistema.
2. Sistem prima dokument od Autora.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) dokument koji se šalje.
2. Vrednosti sistema: Poslati dokument.

Odgovornost učesnika

1. Autor šalje dokument u arhivu.
2. Sistem prima dokument i označava ga kao "nearhiviran".

SK9: Slučaj korišćenja – Brisanje dokumenta

Naziv SK

Brisanje dokumenta

Aktori SK

Administrator

Učesnici SK

Administrator i Sistem

Preduslov

Sistem je uključen i Administrator je prijavljen pod svojom šifrom. Administrator je prethodno, iz glavnog menija izabrao opciju "Pregled Dokumenta" i pokreće se forma za pregled dokumenata, i sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Administrator briše dokument iz sistema.

Osnovni scenario SK

1. Administrator poziva sistem da prikaže formu za pregled dokumenata
2. Sistem prikazuje formu za pregled dokumenata.
3. Administrator bira dokument koji hoće da briše.

4. Sistem prikazuje dokument i formu sa opcijama za rad sa dokumentom.
5. Administrator bira opciju "Briši dokument".
6. Sistem proverava da li dokument pripada nekom objektu ili grupi dokumenata.
7. Sistem traži od Administratora potvrdu brisanja dokumenta.
8. Autor potvrđuje brisanje dokumenta.
9. Sistem proverava potvrdu brisanja.
10. Sistem prihvata potvrdu brisanja.
11. Sistem briše dokument i prikazuje poruku Administratoru o uspešnosti brisanja dokumenta.

Postuslov

Ovaj slučaj korišćenja se završava kada Administrator briše dokument iz sistema.

Alternativna scenarija

- 3.1. Ako dokument ne postoji u Sistemu prekida se izvršenje scenarija.
- 6.1. Ako dokument pripada nekom objektu ili grupi dokumenta Sistem prijavljuje grešku.
- 8.1. Ako Autor ne potvrdi brisanje dokumenta scenario se prekida.

Opis interakcija sistema sa aktorima

1. Administrator briše dokument iz Sistema.
2. Sistem fizički briše dokument.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) dokument koji se briše.
2. Vrednosti sistema: nema.

Odgovornost učesnika

1. Administrator briše dokument.
2. Sistem proverava ispravnost izbora i briše fizički dokument (datoteku).

SK10: Slučaj korišćenja – Zaustavljanje arhiviranja

Naziv SK

Zaustavljanje arhiviranja

Aktori SK

Administrator

Učesnici SK

Administrator i Sistem

Preduslov

Sistem je uključen i Administrator je prijavljen pod svojom šifrom. Administrator je prethodno, iz glavnog menija izabrao opciju "Administracija sistema" i pokreće se forma za administriranje sistema, a sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Administrator zaustavlja (trenutno) proces arhiviranja.

Osnovni scenario SK

1. Administrator poziva sistem da prikaže formu za administraciju sistema.
2. Sistem prikazuje formu za administraciju sistema.
3. Administrator bira opciju "Zaustavi arhiviranje".
4. Sistem proverava da li je neki Autor ili Arhivar trenutno u procesu arhiviranja.
5. Sistem prihvata izbor i zaustavlja arhiviranje.
6. Administrator i svi aktivni korisnici dobijaju poruku da je arhiviranje zaustavljeno.

Postuslov

Ovaj slučaj korišćenja se završava kada Administrator zaustavlja arhiviranje zbog održavanja sistema.

Alternativna scenarija

- 4.1. Ako je neki dokument trenutno u procesu arhiviranja, tada se arhiviranje neće moći zaustaviti.

Opis interakcija sistema sa aktorima

1. Administrator poziva zaustavljanje arhiviranja.
2. Sistem zaustavlja arhiviranje i šalje svima poruke.

Opis korišćenih objekata sistema

1. Poslovni objekti
nema.
2. Vrednosti sistema: nema.

Odgovornost učesnika

1. Administrator poziva zaustavljanje arhiviranja.
2. Sistem zaustavlja arhiviranje i prikazuje poruku.

SK11: Slučaj korišćenja – Administracija korisnika**Naziv SK**

Administracija korisnika

Aktori SK

Administrator

Učesnici SK

Administrator i Sistem

Preduslov

Sistem je uključen i Administrator je prijavljen pod svojom šifrom. Administrator je prethodno, iz glavnog menija izabrao opciju "Administracija korisnika" i pokreće se forma za administriranje korisnika, a sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Administrator radi sa korisnicima.

Osnovni scenario SK

1. Administrator poziva sistem da prikaže formu za administraciju korisnika.
2. Sistem prikazuje formu za administraciju korisnika.

3. Administrator bira jednog korisnika.
4. Sistem prikazuje formu za rad sa izabranim korisnikom.
5. Administrator bira opciju za kreiranje novog, izmenu ili brisanje postojećeg korisnika.
6. Administrator unosi nove ili menja postojeće podatke.
7. Arhivar poziva sistem da prihvati podatke o korisniku.
8. Sistem proverava unete podatke.
9. Sistem prihvata unete podatke.
10. Sistem prikazuje poruku Administratoru o uspešnosti kreiranja (izmene ili brisanja) korisnika i upisuje tu informaciju u log datoteku.

Postuslov

Ovaj slučaj korišćenja se završava kada Administrator dobije poruku od sistem o uspešnosti rada sa korisnikom.

Alternativna scenarija

- 8.1. Ako postoji korisnik sa već postojećim korisničkim imenom prekinuće se izvršavanje scenarija.
- 8.2. Ako se kod izmene ili unosa novog korisnika unese pogrešan podatak prekinuće se izvršavanje scenarija i biće prikazana poruka.
- 9.1. Ako je korisnik autor nekog dokumenta kod brisanja će se javiti greška i biće prikazana poruka.

Opis interakcija sistema sa aktorima

1. Administrator poziva rad sa korisnikom.
2. Sistem kreira, menja ili briše korisnika i prikazuje poruku.

Opis korišćenih objekata sistema

1. Poslovni objekti
 - a) korisnik koji se kreira ili menja.
2. Vrednosti sistema: Korisnik.

Odgovornost učesnika

1. Administrator poziva rad sa korisnikom.
2. Sistem prima zahtev od Administratora obrađuje ga i prikazuje rezultat.

SK12: Slučaj korišćenja – Pregled log datoteka

Naziv SK

Pregled log datoteka

Aktori SK

Administrator

Učesnici SK

Administrator i Sistem

Preduslov

Sistem je uključen i Administrator je prijavljen pod svojom šifrom. Administrator je prethodno, iz glavnog menija izabrao opciju "Administracija sistema", pa "Pregled log datoteka" gde se pokreće forma za administriranje korisnika, a sistem je prikazuje na ekranu.

Kratak opis SK

Ovaj use case se dešava kada Administrator pregleda stanje i istoriju sistema.

Osnovni scenario SK

1. Administrator poziva sistem da prikaže formu za pregled log datoteka.
2. Sistem prikazuje formu za pregled log datoteka.
3. Administrator bira opcije za pregled datoteka i konkretnu datoteku (dnevnu, mesečnu, ...).
4. Sistem prikazuje rezultat na ekranu gde se nalazi naziv korisnika, datum i operacija koju je izvršio u sistemu.
5. Administrator može da izbriše log datoteku.

Postuslov

Ovaj slučaj korišćenja se završava kada Administrator pregleda istoriju sistema.

Alternativna scenarija

- 4.1. Ako nema istorije u sistemu sistem će prijaviti to porukom.
- 5.1. Ako datoteka ne postoji a Administrator pokuša da je izbriše dobiće poruku o grešci.

Opis interakcija sistema sa aktorima

1. Administrator poziva sistem da prikaže informacije iz log datoteka.
2. Sistem čita datoteke i prikazuje ih Korisniku.

Opis korišćenih objekata sistema

1. Poslovni objekti
nema.
2. Vrednosti sistema: nema.

Odgovornost učesnika

1. Administrator poziva sistem da izvrši operaciju.
2. Sistem prima zahtev od Administratora obrađuje ga i prikazuje rezultat.

SK13: Slučaj korišćenja – Prijava u sistem**Naziv SK**

Prijava u sistem

Aktori SK

Administrator, Standardni korisnik, Povlašćeni korisnik, Autor, Arhivar.

Učesnici SK

Administrator, Standardni korisnik, Povlašćeni korisnik, Autor, Arhivar i Sistem.

Preduslov

Sistem je uključen i korisnik je pozvao sistem da prikaže formu za prijavljivanje. Sistem je prikazao formu za unos korisničkog imena i lozinke.

Kratak opis SK

Ovaj use case se dešava kada se korisnik prijavljuje u sistem.

Osnovni scenario SK

1. Korisnik poziva sistem da prikaže formu za prijavljivanje u sistem.
2. Sistem prikazuje formu za prijavljivanje u sistem.
3. Korisnik unos korisničko ime i lozinku.
4. Korisnik poziva sistem da prihvati unesene podatke.
5. Sistem proverava unete podatke.
6. Sistem prihvata unete podatke.
7. Sistem prikazuje osnovnu formu za rad sa sistemom prilagođenu grupi kojoj korisnik pripada.
8. Sistem upisuje u log datoteku informaciju o prijavljivanju korisnika.

Postuslov

Ovaj slučaj korišćenja se završava kada se korisniku prikaže njegova prilagođena forma na ekranu.

Alternativna scenarija

- 4.1. Ako je uneseno pogrešno korisničko ime ili lozinka prekida se izvršenje scenarija.

Opis interakcija sistema sa aktorima

1. Korisnik poziva Sistem da prikaže formu za prijavljivanje i unosi korisničko ime i lozinku.
2. Sistem prikazuje formu za prijavljivanje i proverava da li su uneti korisničko ime i lozinka tačni.

Opis korišćenih objekata sistema

1. Poslovni objekti
nema.
2. Vrednosti sistema: nema.

Odgovornost učesnika

1. Korisnika poziva Sistem da proveri korisničko ime i lozinku.
2. Sistem prikazuje formu za prijavljivanje i proverava korisničko ime i lozinku.

SK14: Slučaj korišćenja – Odjava iz sistema**Naziv SK**

Odjava iz sistema

Aktori SK

Administrator, Standardni korisnik, Povlašćeni korisnik, Autor, Arhivar.

Učesnici SK

Administrator, Standardni korisnik, Povlašćeni korisnik, Autor, Arhivar i Sistem.

Preduslov

Sistem je uključen i korisnik je prijavljen pod svojim korisničkim imenom i lozinkom.

Kratak opis SK

Ovaj use case se dešava kada se korisnik odjavljuje iz sistema.

Osnovni scenario SK

1. Korisnik poziva sistem da se odjavi iz njega.

2. Sistem proverava da li je korisnik prijavljen.
3. Sistem prihvata zahtev od korisnika.
4. Sistem upisuje informaciju o odjavljivanju korisnika u log datoteku.
5. Sistem prikazuje korisniku formu za prijavljivanje i odjavnu poruku.

Postuslov

Ovaj slučaj korišćenja se završava kada se korisniku prikaže forma za prijavljivanje u sistem i odjavna poruka.

Alternativna scenarija

2.1. Ako korisnik nije prijavljen ili mu je isteklo vreme za rad sa sistemom prekida se izvršenje scenarija.

Opis interakcija sistema sa aktorima

1. Korisnik poziva Sistem da ga odjavi iz sistema.
2. Sistem proverava da li je korisnik prijavljen i odjavljuje korisnika iz sistema.

Opis korišćenih objekata sistema

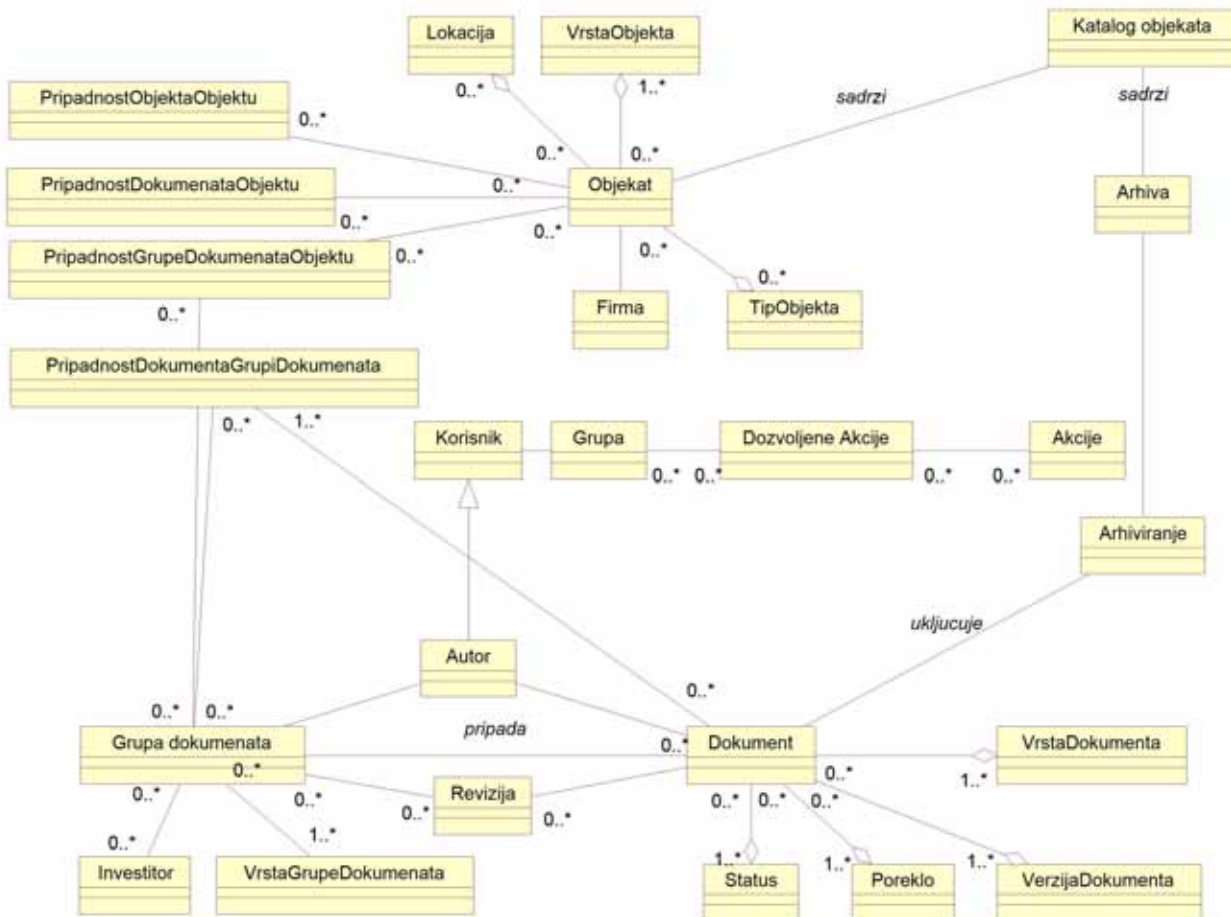
1. Poslovni objekti
nema.
2. Vrednosti sistema: nema.

Odgovornost učesnika

1. Korisnika poziva Sistem da ga odjavi iz sistema.
2. Sistem ga odjavljuje iz sistema i prikazuje poruku.

ARCHIDOC 4: Domenski model

Na osnovu korisničkog zahteva se dobijaju osnovni koncepti sistema koji su predstavljeni preko dijagrama klasa:



Slika 5: Domenski model sistema predstavljen preko dijagrama klasa

ARCHIDOC 5: Rečnik termina

Na osnovu poslovnog i domenskog modela pravi se rečnik termina:

Korisnik, GrupaKorisnika, DozvoljeneAkcije, Akcije, Dokument, StatusDokumenta, Poreklo dokumenta, VerzijaDokumenta, VrstaDokumenta, Revizija, GrupaDokumenata, Investitor, VrstaGrupeDokumenata, Objekat, VrstaObjekta, Lokacija, Firma, TipObjekta, PripadnostObjektaObjektu, PripadnostDokumentaObjektu, PripadnostGrupeDokumenataObjektu, PripadnostDokumentaGrupiDokumenata, ArhiviranjeDokumenta, SlanjeDokumentaUARhivu, Arhiva.

ARCHIDOC 6: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Arhiviranje nove dokumentacije

Šifra: PR01

Opis: Forma za arhiviranje dokumentacije.

Prototip treba da sadrži sledeća polja: *NazivDokumenta*, *BrojDokumenta*, *Napomena*, *VrstaDokumenta*, *NazivDatoteke*, *RevizijaDokumenta*, *VerzijaDokumenta*, *Poreklo*, *StatusDokumenta*, *Autor*, *PoveziSaObjektima* i *PoveziSaGrupamaDokumenata*.

ARCHIDOC 6.1: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Arhiviranje nove dokumentacije

Šifra: PR02

Opis: Forma za arhiviranje dokumentacije.

Prototip treba da sadrži sledeća polja: *NazivDokumenta*, *BrojDokumenta*, *Napomena*, *VrstaDokumenta*, *NazivDatoteke*, *RevizijaDokumenta*, *VerzijaDokumenta*, *Poreklo*, *StatusDokumenta*, *Autor*.

Šifra: PR03

Opis: Sistem prikazuje objekte i grupe dokumenata za vezivanje.

Prototip treba da sadrži sledeća polja: *NazivDokumenta*, *BrojDokumenta*, *Napomena*, *VrstaDokumenta*, *NazivDatoteke*, *RevizijaDokumenta*, *VerzijaDokumenta*, *Poreklo*, *StatusDokumenta*, *Autor*, *PoveziSaObjektima* i *PoveziSaGrupamaDokumenata*.

ARCHIDOC 6.2: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Pretraživanje dokumenata

Šifra: PR04

Opis: Forma za pretraživanje dokumenata.

Prototip treba da sadrži sledeća polja: *SveReci*, *TacnaFraza*, *BarJednaRec*, *BezReci*, *Velicina*, *Datum*, *KljucnaRec*.

ARCHIDOC 6.3: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Pregled dokumenta

Šifra: PR05

Opis: Forma za pregled dokumenta.

Prototip treba da sadrži sledeća polja: *NazivDokumenta*, *TipDokumenta*, *BrojDokumenta*, *Datum*, *Napomena*, *VrstaDokumenta*, *NazivDatoteke*, *RevizijaDokumenta*, *VerzijaDokumenta*, *Poreklo*, *StatusDokumenta*, *Autor*.

ARCHIDOC 6.4: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Izmena podataka o dokumentu

Šifra: PR06

Opis: Forma za izmenu dokumenta.

Prototip treba da sadrži sledeća polja: *NazivDokumenta, BrojDokumenta, Napomena, VrstaDokumenta, RevizijaDokumenta, VerzijaDokumenta, Poreklo, StatusDokumenta.*

ARCHIDOC 6.5: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Administracija objekata

Šifra: PR07

Opis: Forma za rad sa objektima.

Prototip treba da sadrži sledeća polja: *NazivObjekta, BrojObjekta, Datum, Napomena, Lokacija, VrstaObjekta, TipObjekta, Firma.*

ARCHIDOC 6.6: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Administracija grupa dokumenata

Šifra: PR08

Opis: Forma za rad sa grupama dokumenata.

Prototip treba da sadrži sledeća polja: *NazivGrupeDokumenata, BrojGrupeDokumenata, Napomena, Datum, BrojUgovora, Lokacija, Vrsta, Autori, Investitor, Revizija.*

ARCHIDOC 6.7: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Kreiranje broja novog dokumenta

Šifra: PR09

Opis: Forma za slanje systemske poruke.

Prototip treba da sadrži sledeća polja: *Posiljalac, NaslovPoruke, TekstPoruke.*

Šifra: PR10

Opis: Ekran za prikaz systemske poruke.

Prototip treba da sadrži sledeća polja: *Posiljalac, NaslovPoruke, TekstPoruke.*

ARCHIDOC 6.8: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Slanje dokumenta u arhivu

Šifra: PR11

Opis: Forma za slanje dokumenta u arhivu.

Prototip treba da sadrži sledeća polja: *NazivDatoteke, NazivDokumenta, BrojDokumenta, Napomena, VrstaDokumenta, NazivDatoteke, RevizijaDokumenta, VerzijaDokumenta, Poreklo, StatusDokumenta, Autor.*

ARCHIDOC 6.9: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Brisanje dokumenta

Šifra: PR12

Opis: Forma za brisanje dokumenta.

Prototip treba da sadrži sledeća polja: *NazivDatoteke, NazivDokumenta, Tip, BrojDokumenta, Datum.*

ARCHIDOC 6.10: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Administracija korisnika

Šifra: PR13

Opis: Forma za administraciju korisnika.

Prototip treba da sadrži sledeća polja: *KorisnickoIme, Sifra, Ime, Prezime, Telefon, EMail, GrupaKorisnika, StatusKorisnika.*

ARCHIDOC 6.11: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Pregled log datoteka

Šifra: PR14

Opis: Ekran za pregled log datoteka.

Prototip treba da sadrži sledeća polja: *RedniBrojDogadjaja, Korisnik, Dogadjaj, Datum.*

ARCHIDOC 6.12: Prototipovi korisničkog interfejsa za sistemski slučaj korišćenja - Prijava u sistem

Šifra: PR15

Opis: Forma za prijavu korisnika u sistem.

Prototip treba da sadrži sledeća polja: *KorisnickoIme i Sifra.*

ARCHIDOC 7: Rečnik termina

Na osnovu poslovno - sistemskih slučajeva korišćenja se dobijaju novi termini koji se dodaju postojećim:

Postojeći termini:

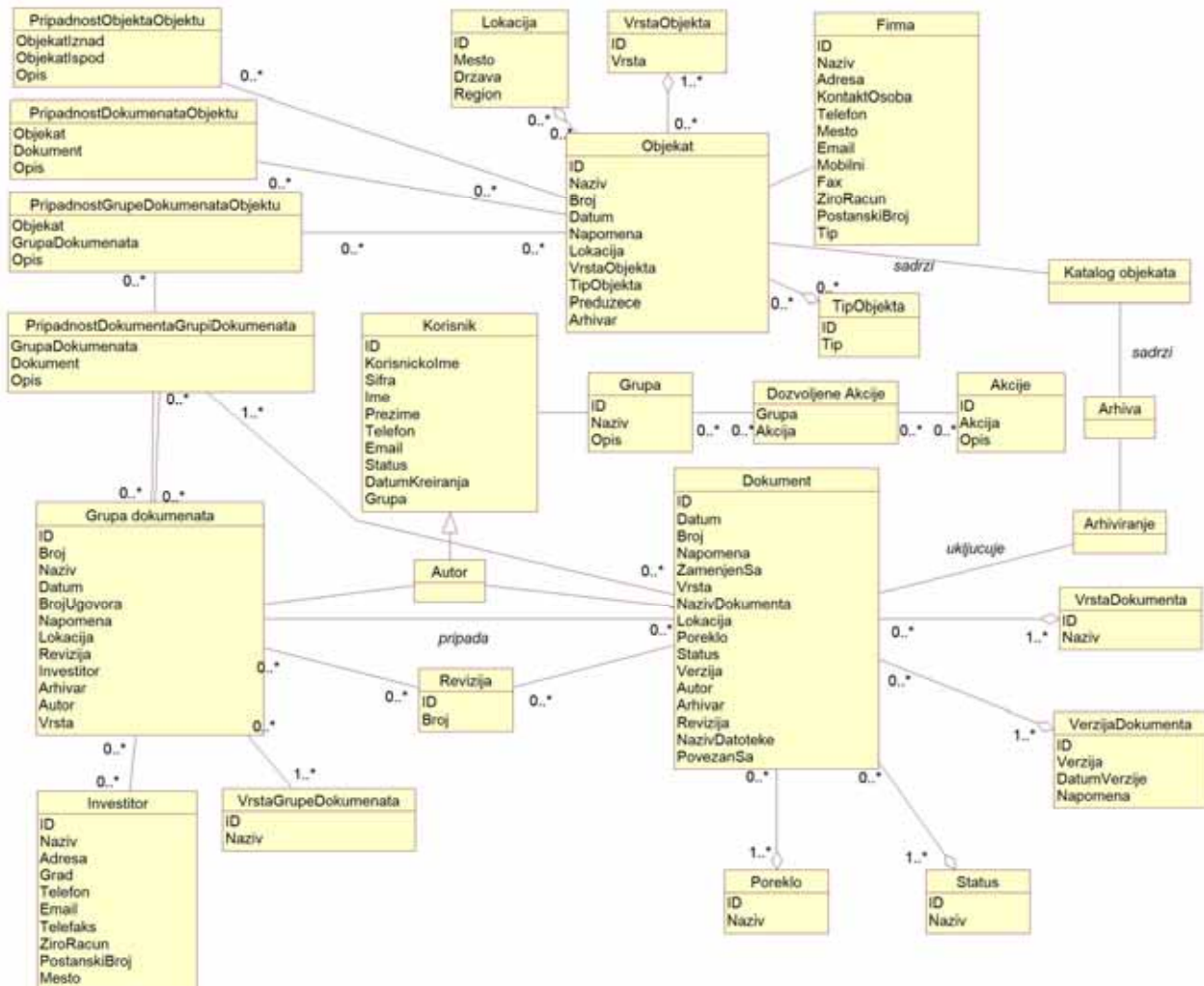
Korisnik, GrupaKorisnika, DozvoljeneAkcije, Akcije, Dokument, StatusDokumenta, Poreklo dokumenta, VerzijaDokumenta, VrstaDokumenta, Revizija, GrupaDokumenata, Investitor, VrstaGrupeDokumenata, Objekat, VrstaObjekta, Lokacija, Firma, TipObjekta, PripadnostObjektaObjektu, PripadnostDokumentaObjektu, PripadnostGrupeDokumenataObjektu, PripadnostDokumentaGrupiDokumenata, ArhiviranjeDokumenta, SlanjeDokumentaUArhivu, Arhiva.

Novi termini:

NazivDokumenta, BrojDokumenta, Napomena, VrstaDokumenta, NazivDatoteke, RevizijaDokumenta, VerzijaDokumenta, Poreklo, StatusDokumenta, Autor, PoveziSaObjektima i PoveziSaGrupamaDokumenata, SveReci, TacnaFraza, BarJednaRec, BezReci, Velicina, Datum, KljucnaRec, NazivObjekta, BrojObjekta, Datum, Napomena, VrstaObjekta, TipObjekta, Firma, NazivGrupeDokumenata, BrojGrupeDokumenata, BrojUgovora, Lokacija, Vrsta, Autori, Investitor, Posiljalac, NaslovPoruke, TekstPoruke, KorisnickoIme, Sifra, Ime, Prezime, Telefon, EMail, GrupaKorisnika, StatusKorisnika, RedniBrojDogadjaja, Dogadjaj.

ARCHIDOC 8: Domenski model sa atributima

Postojećim konceptima iz domenskog modela se dodaju atributi, što je predstavljeno dijagramom klasa na slici 6.



Slika 6: Domenski model sa atributima

2.2. ANALIZA

Zahtevi koji se analiziraju se predstavljaju preko modela slučaja korišćenja, detaljnim opisom svakog slučaja korišćenja i prototipovima korisničkih interfejsa za svaki slučaj korišćenja. Analiza daje mogućnost daljeg razrađivanja slučaja korišćenja, tj. formalan opis realizacija slučaja korišćenja. Struktura modela analize se oslanja na strukturu modela slučaja korišćenja. Tok analize se sastoji od: analize slučajeva korišćenja, analize paketa, analize klasa i analize arhitekture. Slučajevi korišćenja se analiziraju da bi se: identifikovale klase analize čiji su objekti neophodni kako bi se realizovali slučajevi korišćenja, distribuiralo ponašanje slučaja korišćenja na objekte analize koji omogućavaju ovo ponašanje i prikupili posebni zahtevi koji su definisani za slučajeve korišćenja.

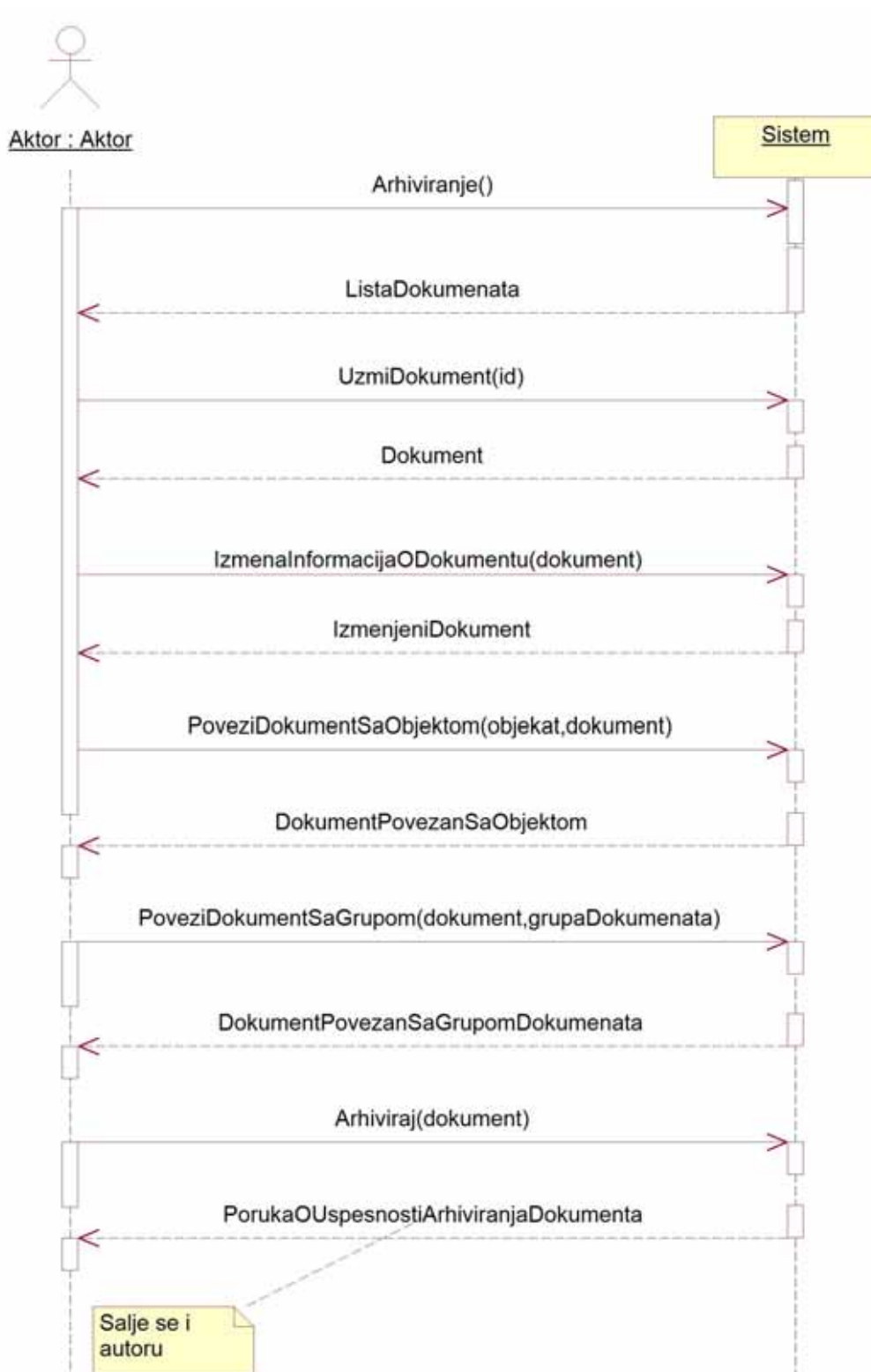
Faza analize opisuje logičku strukturu i ponašanje softverskog sistema. Ponašanje softverskog sistema je opisano pomoću sistemskih dijagrama sekvenci, koji se prave za svaki slučaj korišćenja, i pomoću ugovora o sistemskim operacijama. Struktura softverskog sistema se opisuje pomoću konceptualnog i relacionog modela.

2.2.1. Dijagrami sekvenci slučajeva korišćenja

DS1: *Dijagrami sekvenci slučaja korišćenja - Arhiviranje nove dokumentacije*

Osnovni scenario:

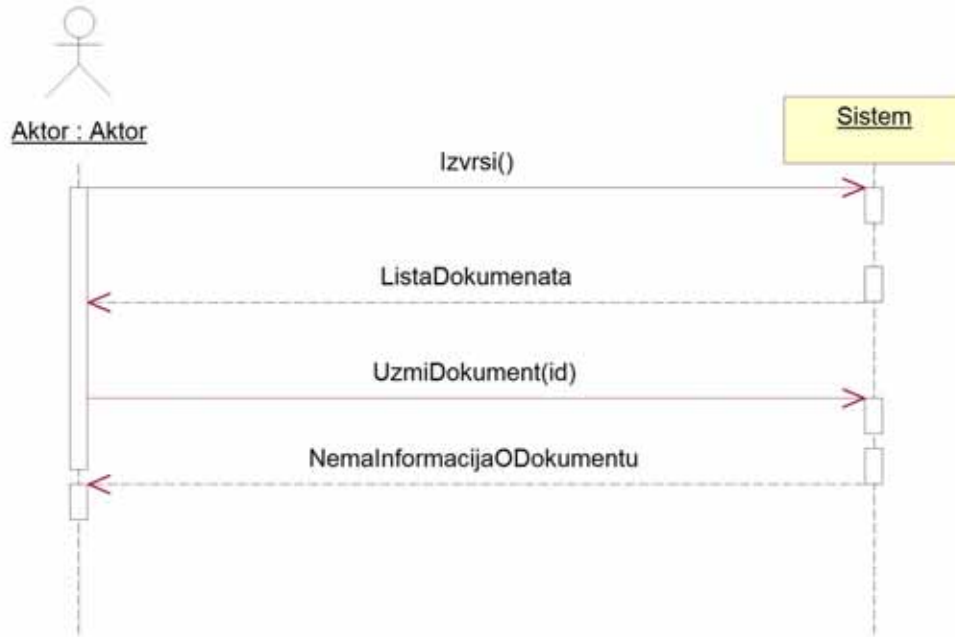
1. Arhivar poziva sistem da prikaže dokumente koji čekaju arhiviranje.
2. Arhivar vrši izbor dokumenta koji hoće da arhivira.
3. Sistem prikazuje arhivaru sve podatke o dokumentu koje je uneo autor.
4. Arhivar može da veže dokument za objekte ili grupe dokumenata.
5. Arhivar kontroliše unete podatke.
6. Arhivar poziva sistem da prihvati podatke o dokumentu sa vezama.
7. Sistem proverava unete podatke.
8. Sistem prihvata unete podatke.
9. Sistem prikazuje poruku Arhivaru o uspešnosti arhiviranja i upisuje tu informaciju u log datoteku.



Slika 7: Osnovni scenario - SK Arhiviranje nove dokumentacije

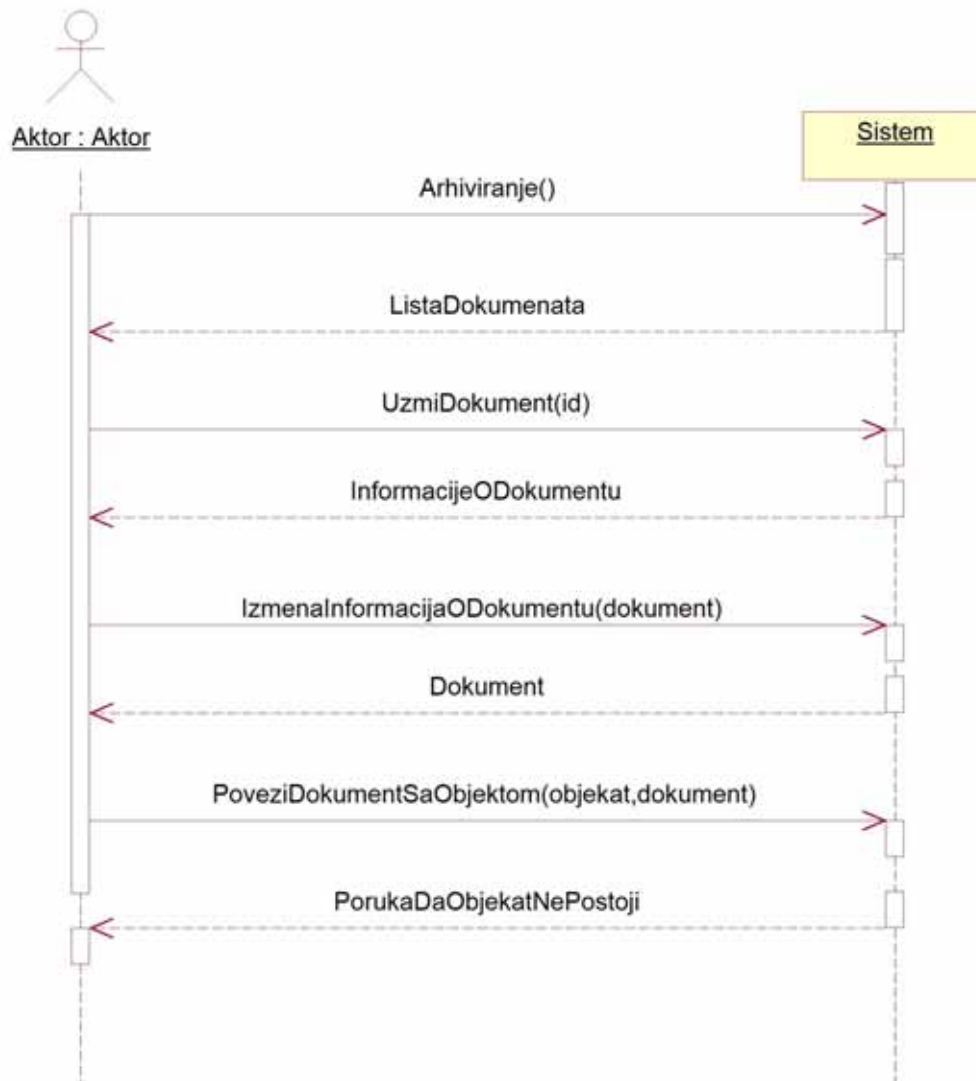
Alternativni scenariji

2.1. Ako je administrator zabranio dodavanje dokumenata, arhivar neće moći arhivirati dokument.



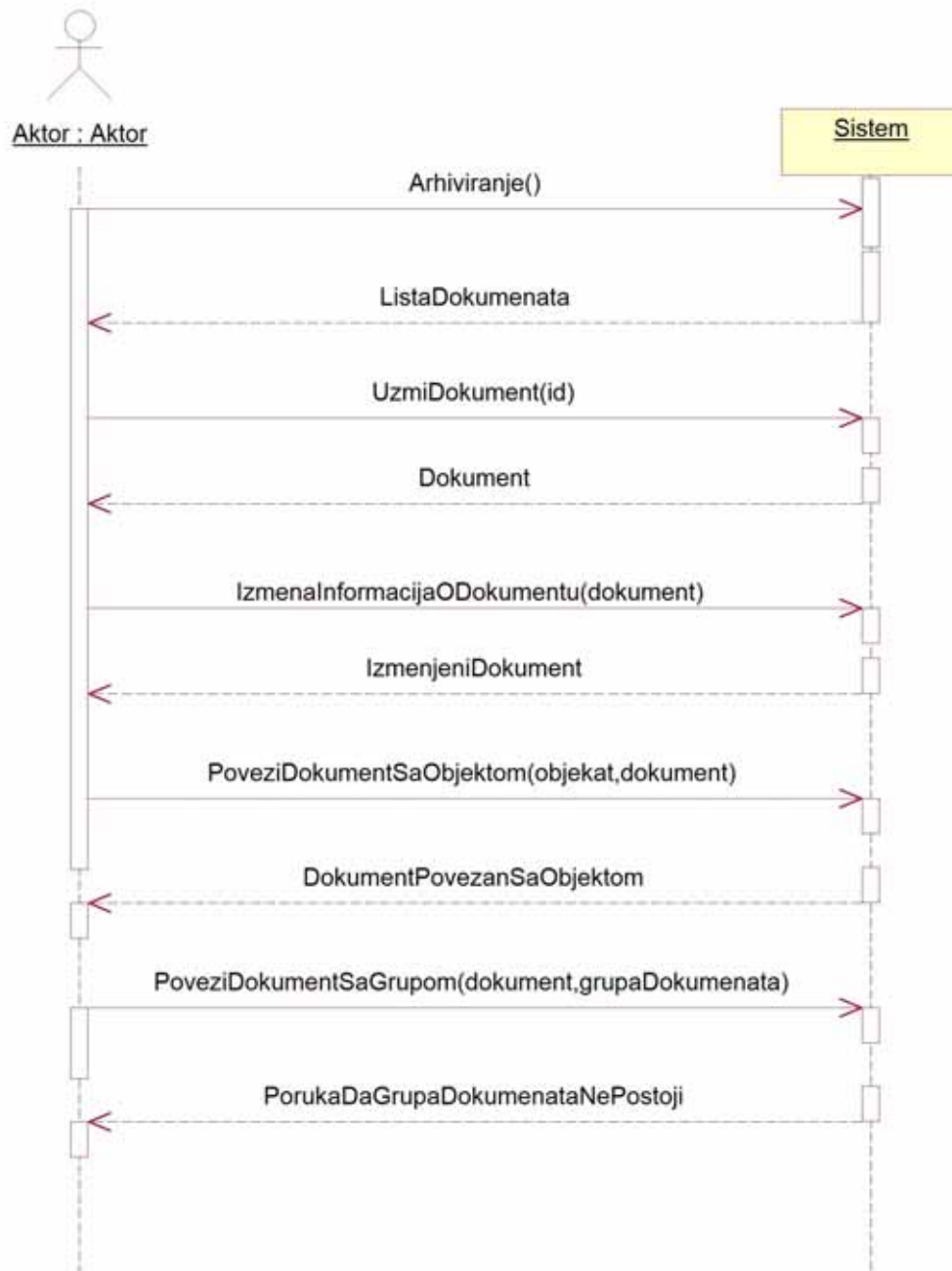
Slika 8: Alternativni scenario 1 - SK Arhiviranje nove dokumentacije

5.1. Ako objekat ne postoji Arhivar će dobiti poruku o grešci.



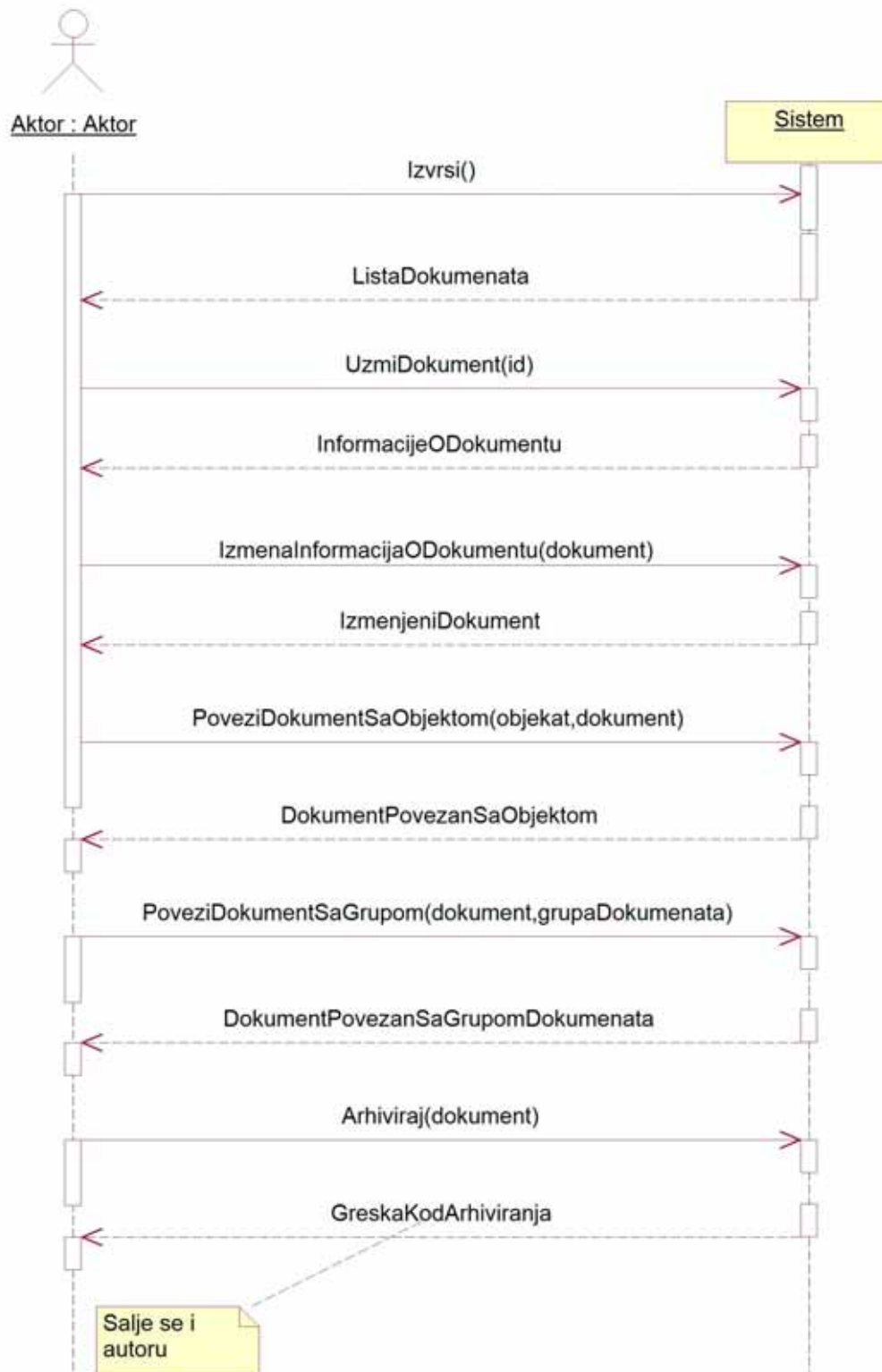
Slika 9: Alternativni scenario 2 - SK Arhiviranje nove dokumentacije

5.2. Ako grupa dokumenata ne postoji Arhivar će dobiti poruku o grešci.



Slika 10: Alternativni scenario 3 - SK Arhiviranje nove dokumentacije

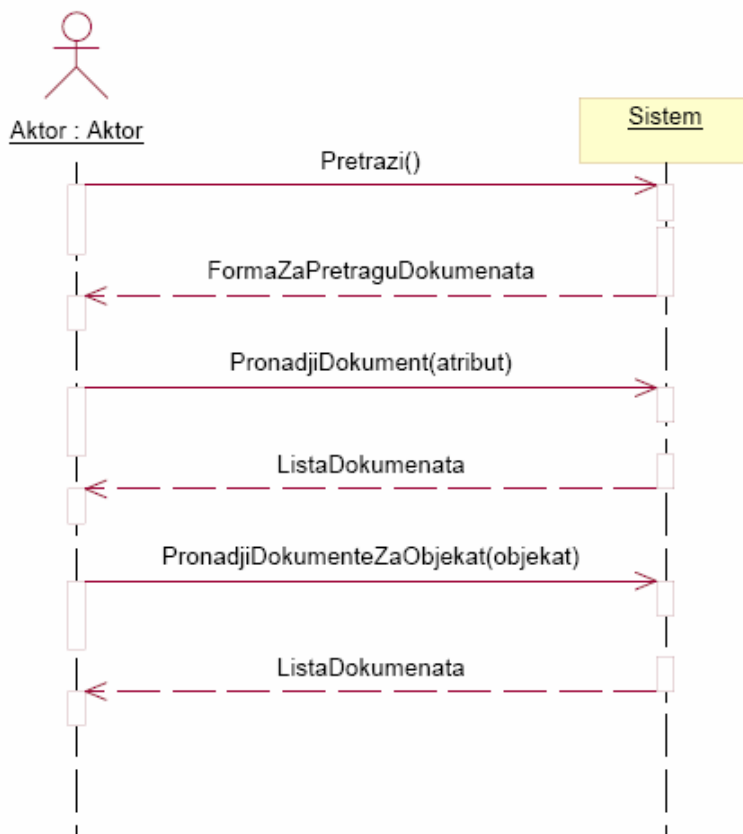
7.1. Ako autor dokumenta nije uneo sve informacije o vrsti dokumenta ili o samom dokumentu, prekinuće se izvršavanje scenarija.



Slika 11: Alternativni scenario 4 - SK Arhiviranje nove dokumentacije

DS2: Dijagrami sekvenci slučaja korišćenja - Pretraživanje dokumenata**Osnovni scenario:**

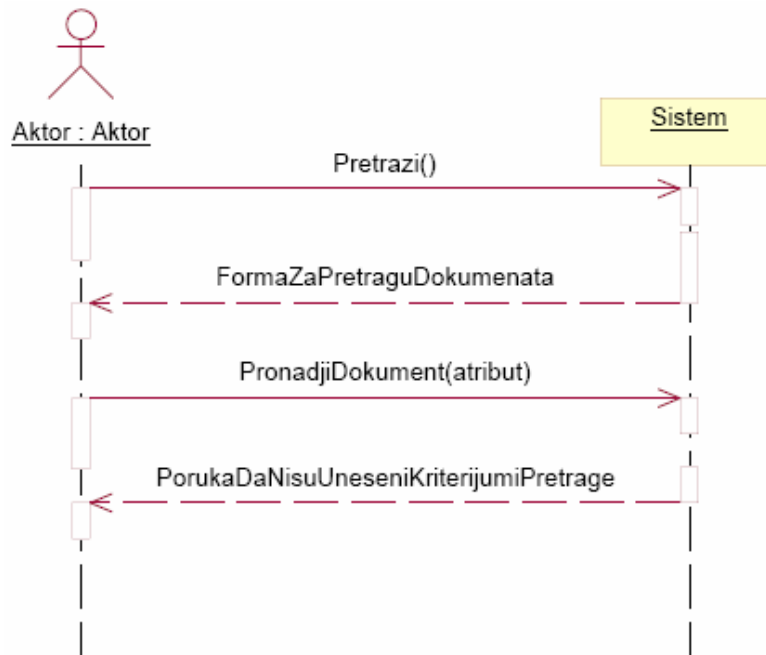
1. Korisnik poziva sistem da prikaže formu za pretragu dokumenata ili na glavnom ekranu unosi reč(i) za pretragu.
2. Korisnik vrši izbor opcija za pretragu dokumenata.
3. Korisnik poziva sistem da prihvati podatke za pretragu i da pretraži dokumente.
4. Sistem prihvata unete podatke i pretražuje dokumente.
5. Sistem prikazuje rezultate pretrage dokumenata (dokumente koje je pronašao ili poruku ako ni jedan dokument nije pronašao).



Slika 12: Osnovni scenario - SK Pretraživanje dokumenata

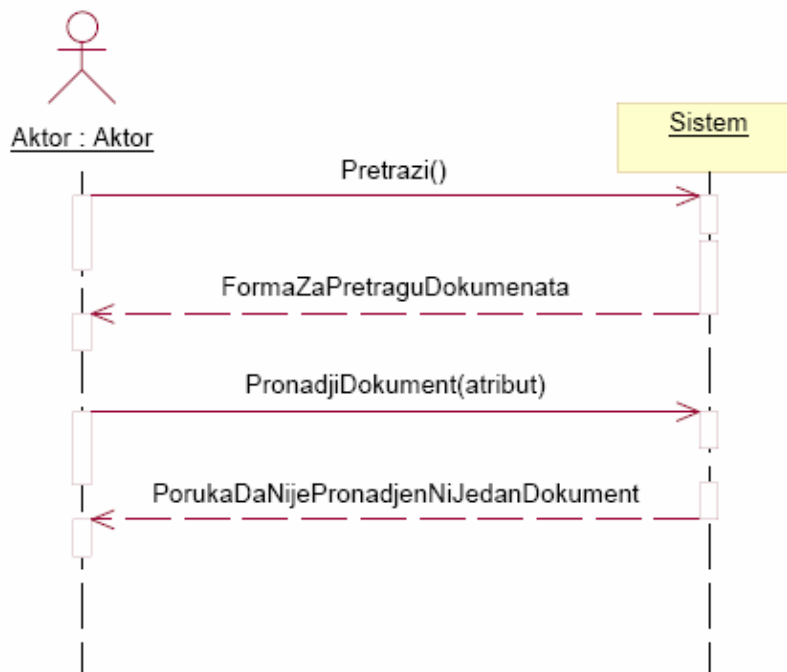
Alternativni scenariji

- 3.1. Ako korisnik nije uneo kriterijum pretrage sistem će prijaviti grešku i tražiti da se kriterijum ponovo unese.



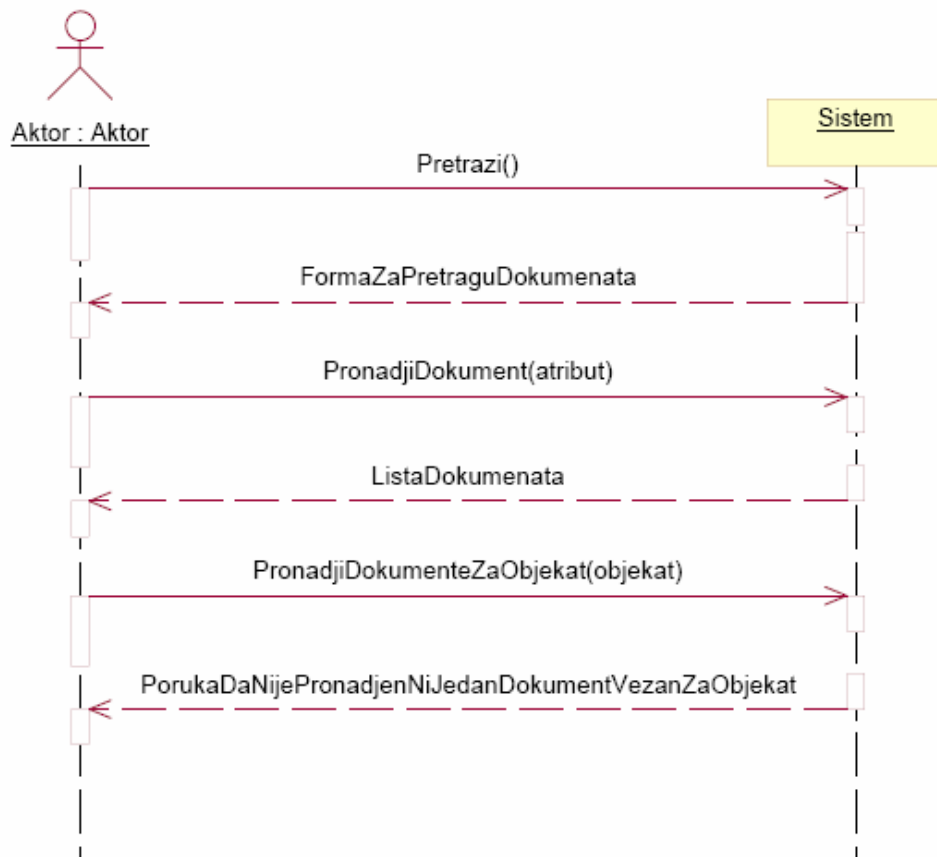
Slika 13: Alternativni scenario 1 - SK Pretraživanje dokumenata

5.1. U slučaju da ni jedan dokument ne zadovoljava postavljeni uslov sistem će prijaviti poruku o tome da nije mogao da pronađe traženi dokument.



Slika 14: Alternativni scenario 2 - SK Pretraživanje dokumenata

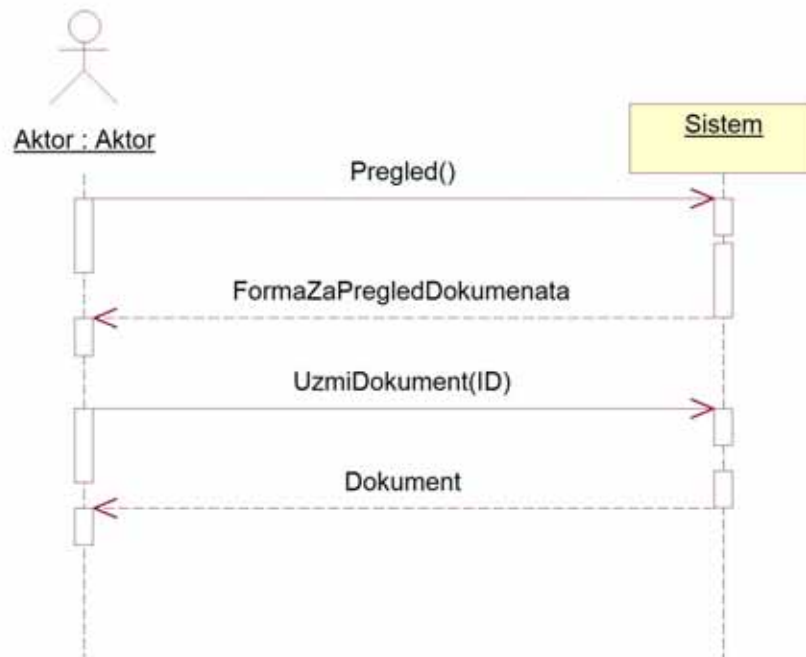
5.2. U slučaju da ni jedan dokument ne zadovoljava uslov vezanosti za objekat sistem će prijaviti poruku o tome da nije mogao da pronađe traženi dokument.



Slika 14: Alternativni scenario 3 - SK Pretraživanje dokumenata

DS3: Dijagrami sekvenci slučaja korišćenja - Pregled dokumenta**Osnovni scenario:**

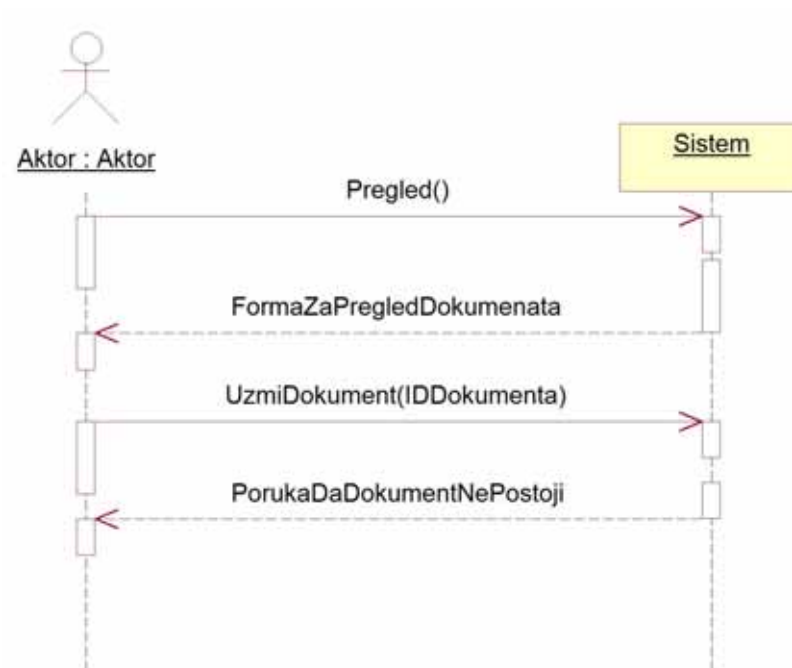
1. Korisnik poziva sistem da prikaže formu za pregled dokumenata.
2. Korisnik vrši izbor dokumenta iz liste dokumenata.
3. Sistem prikazuje detaljne informacije o sistemu i prikazuje opcije za rad sa dokumentom.



Slika 15: Osnovni scenario - SK Pregled dokumenta

Alternativni scenariji

2.1. Ako ne postoji dokument u sistemu Korisnik će dobiti poruku o tome.

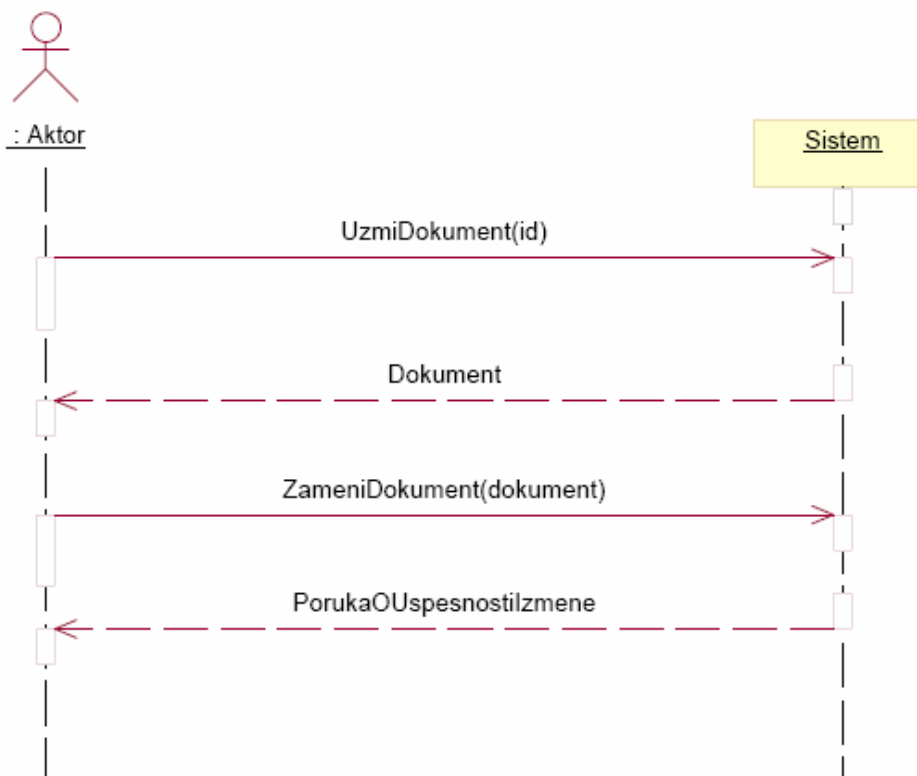


Slika 16: Alternativni scenario 1 - SK Pregled dokumenta

DS4: Dijagrami sekvenci slučaja korišćenja - Izmena podataka o dokumentu**Osnovni scenario:**

1. Korisnik poziva sistem da preuzme dokument za izmenu.
2. Sistem korisniku šalje dokument.

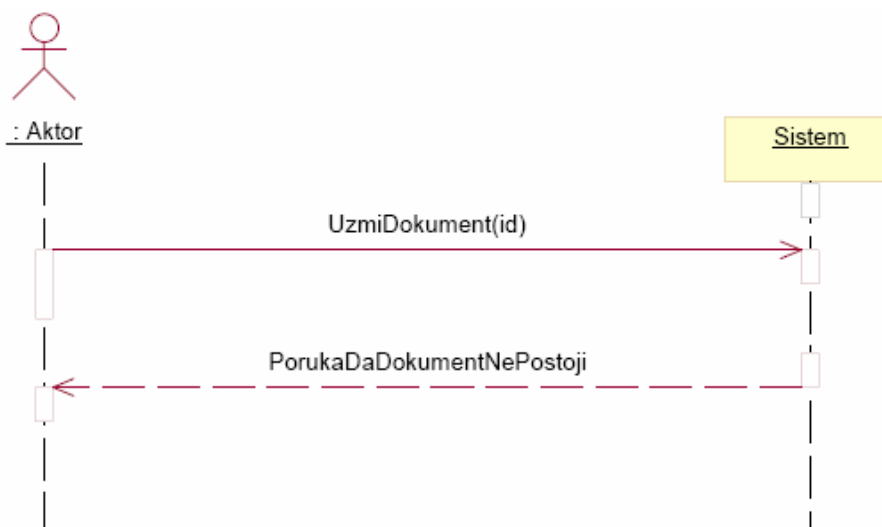
3. Korisnik vrši izmenu dokumenta i izmenjen dokument šalje nazad sistemu.
4. Sistem prihvata izmenu i šalje korisniku poruku o uspešnosti izmene.



Slika 17: Osnovni scenario - SK Izmena podataka o dokumentu

Alternativni scenariji

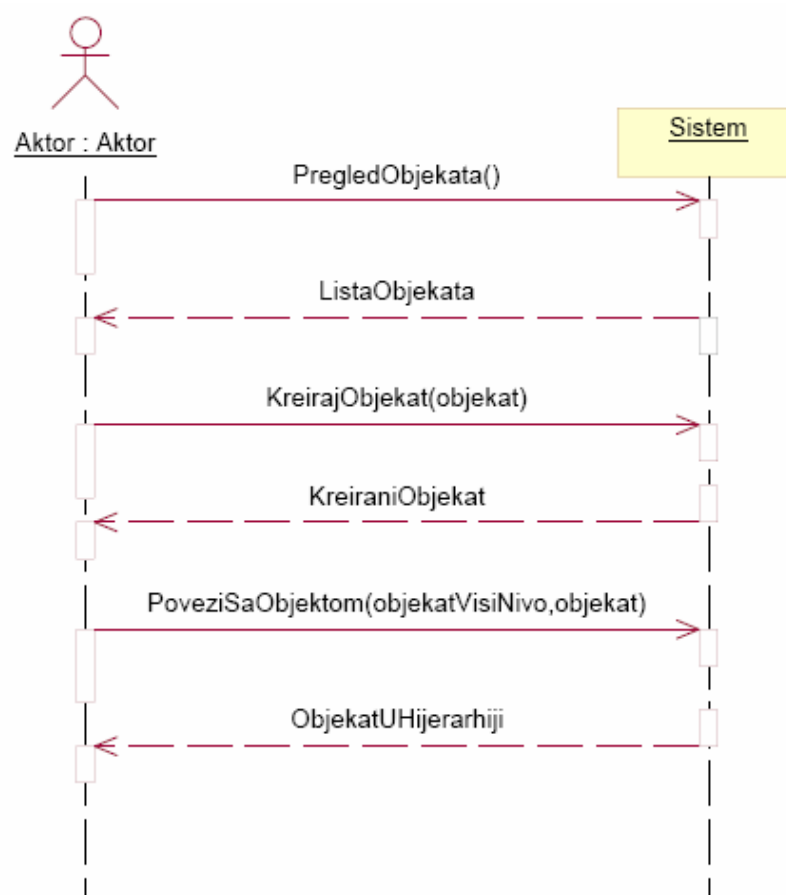
- 1.1. Ako je korisnik od sistema tražio dokument koji ne postoji, sistem mu šalje poruku obaveštenja.



Slika 18: Alternativni scenario 1 - SK Izmena podataka o dokumentu

Osnovni scenario:

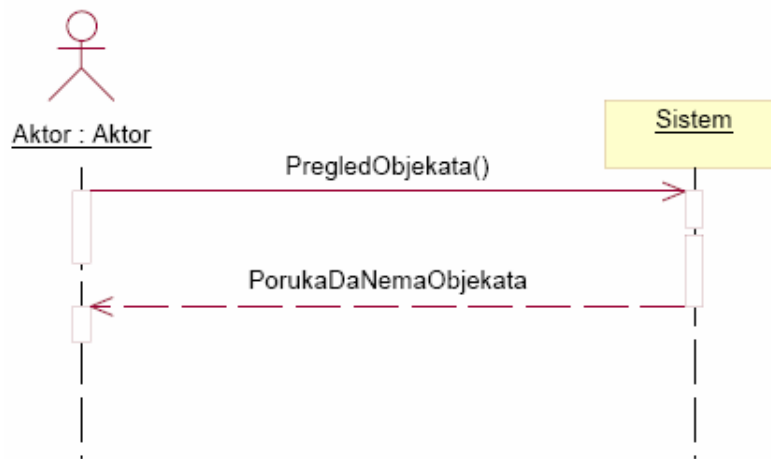
1. Korisnik poziva sistem da prikaže objekte koji postoje u sistemu.
2. Korisnik kreira novi objekat ili menja postojeći objekat i unos informacije o objektu.
3. Sistem prikazuje Korisniku formu za kreiranje novog objekta.
4. Korisnik unos podatke i veže objekat za neke druge objekte (opciono).
5. Korisnik kontrolise unete podatke.
6. Korisnik poziva sistem da prihvati podatke o objektu.
7. Sistem proverava unete podatke.
8. Sistem prihvata unete podatke.
9. Sistem prikazuje poruku Korisniku o uspešnosti rada sa objektom i prikazuje sve objekte (koji pripadaju izabranom objektu).



Slika 19: Osnovni scenario - SK Administracija objekata

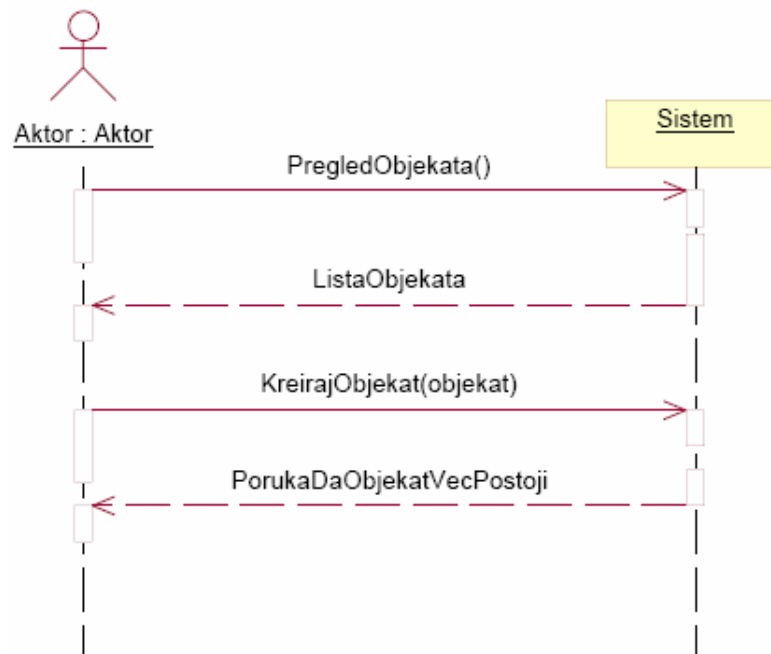
Alternativni scenariji

- 2.1. Ako ne postoji objekat u sistemu Sistem će prikazati poruku.



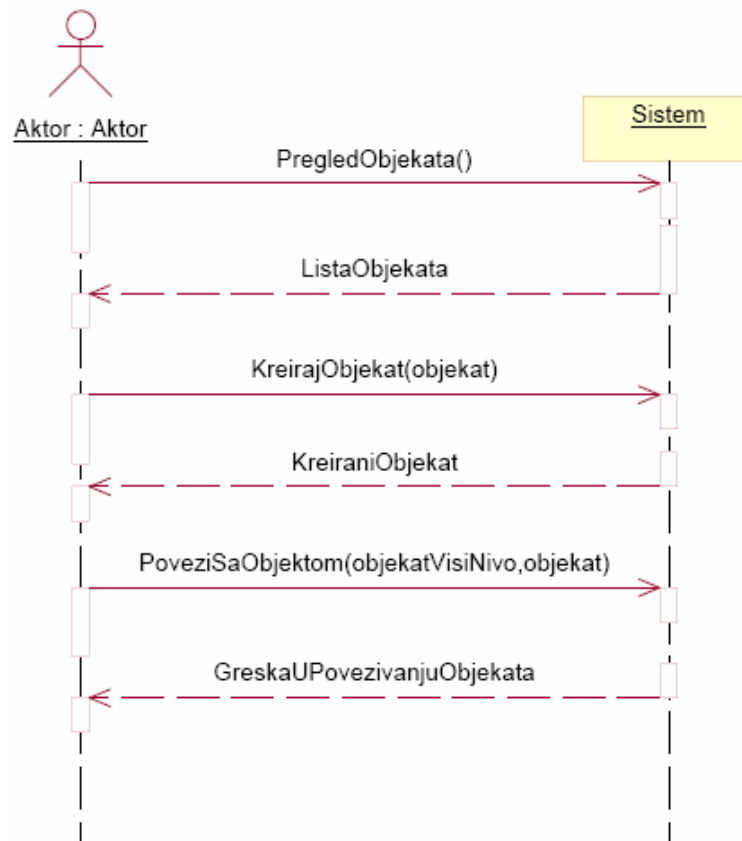
Slika 20: Alternativni scenario 1 - SK Administracija objekata

2.2. Ako Korisnik pokuša da kreira objekat koji već postoji Sistem će prijaviti grešku.



Slika 21: Alternativni scenario 2 - SK Administracija objekata

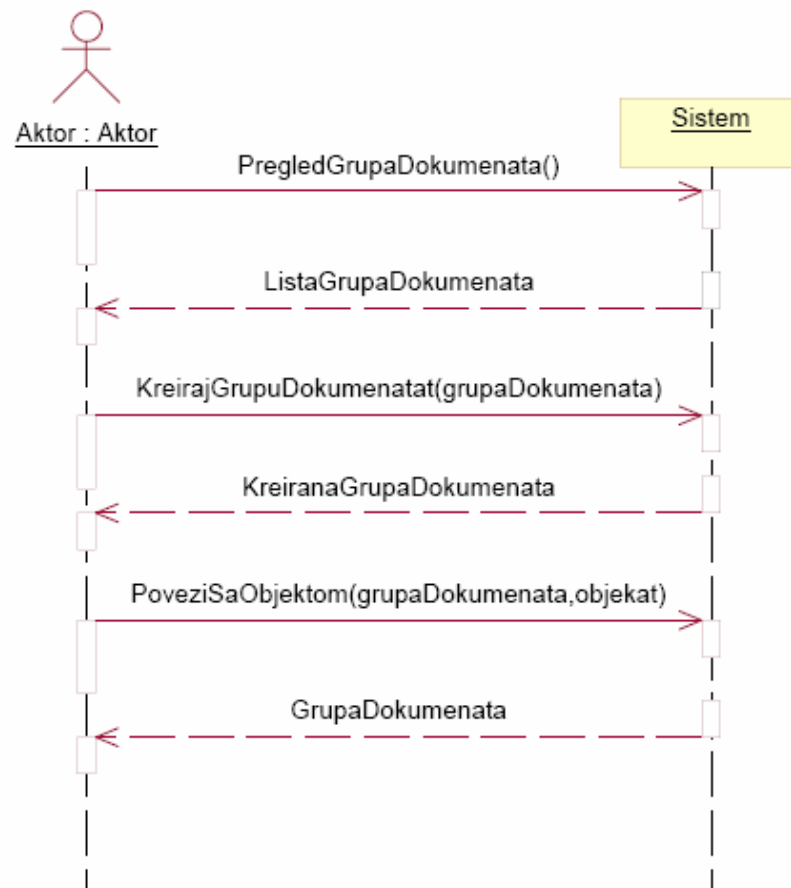
4.1. Ako Korisnik progrešno veže objekte Sistem će prijaviti grešku.



Slika 22: Alternativni scenario 3 - SK Administracija objekata

DS6: Dijagrami sekvenci slučaja korišćenja - Administracija grupa dokumenata**Osnovni scenario:**

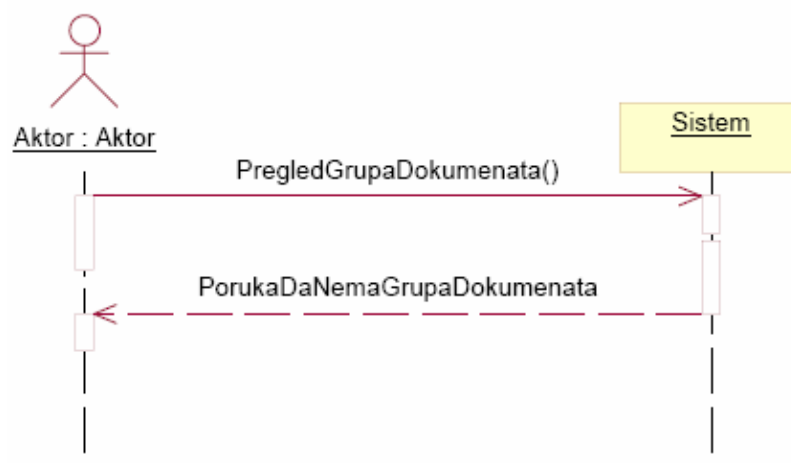
1. Korisnik poziva sistem da prikaže vrste grupa dokumenata koji postoje u sistemu.
2. Korisnik bira vrstu grupe dokumenata za prikaz.
3. Korisnik kreira novu grupu dokumenata ili menja postojeću i unos informacije o njoj.
4. Sistem prikazuje Korisniku formu za kreiranje nove grupe dokumenata.
5. Korisnik unos podatke i veže grupu dokumenata za neke objekte (opciono).
6. Korisnik kontrolira unete podatke.
7. Korisnik poziva sistem da prihvati podatke o grupi dokumenata.
8. Sistem proverava unete podatke.
9. Sistem prihvata unete podatke.
10. Sistem prikazuje poruku Korisniku o uspešnosti rada sa grupom dokumenata i prikazuje sve grupe dokumenata za izabranu vrstu grupe dokumenata.



Slika 23: Osnovni scenario - SK Administracija grupa dokumenata

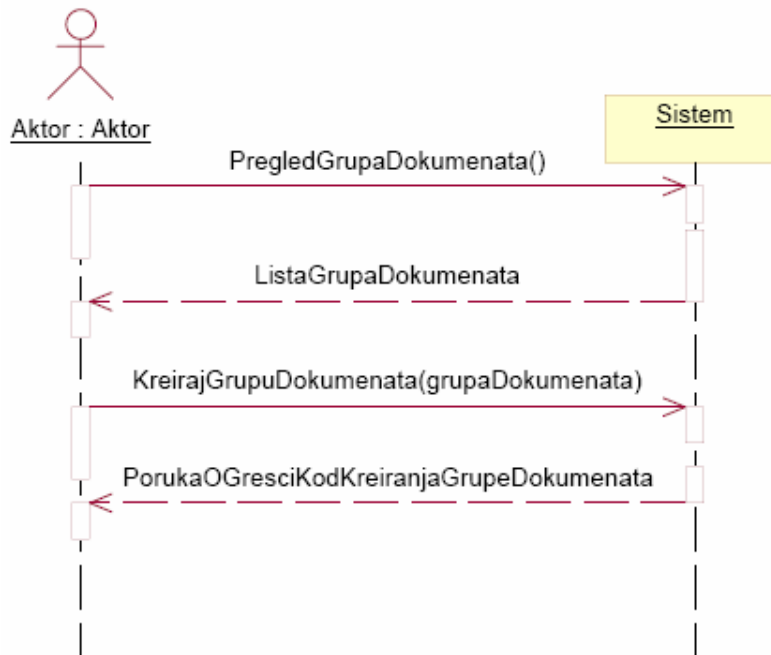
Alternativni scenariji

1.1. Ako ne postoji ni jedna grupa dokumenata u sistemu biće prikazana poruka.



Slika 24: Alternativni scenario 1 - SK Administracija grupa dokumenata

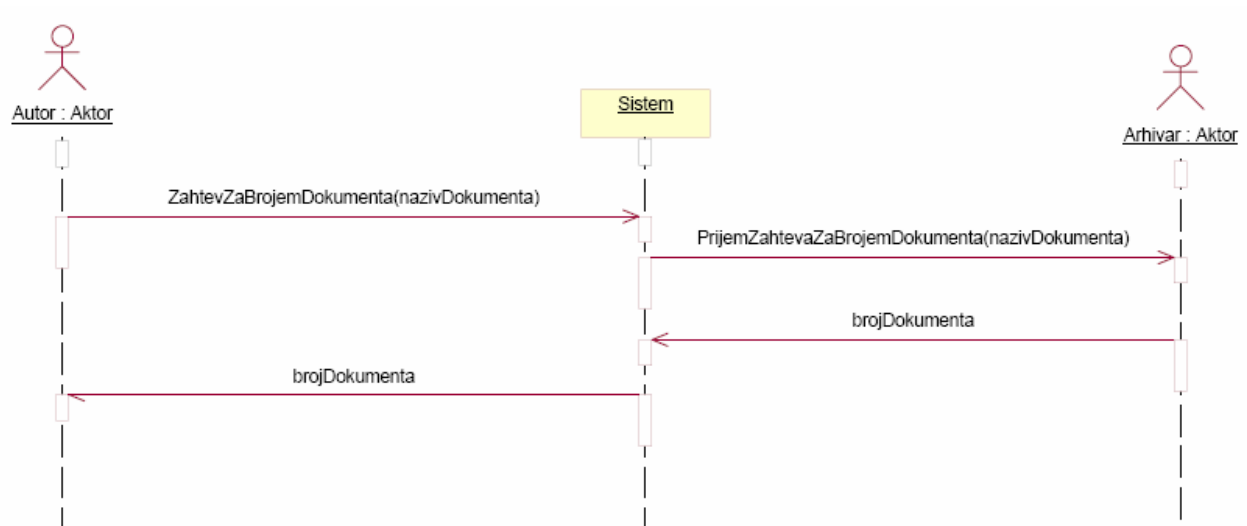
3.1. Ako je Korisnik uneo pogrešne podatke o grupi dokumenata Sistem će prijaviti grešku.



Slika 25: Alternativni scenario 2 - SK Administracija grupa dokumenata

DS7: Dijagrami sekvenci slučaja korišćenja - Kreiranje broja novog dokumenta**Osnovni scenario:**

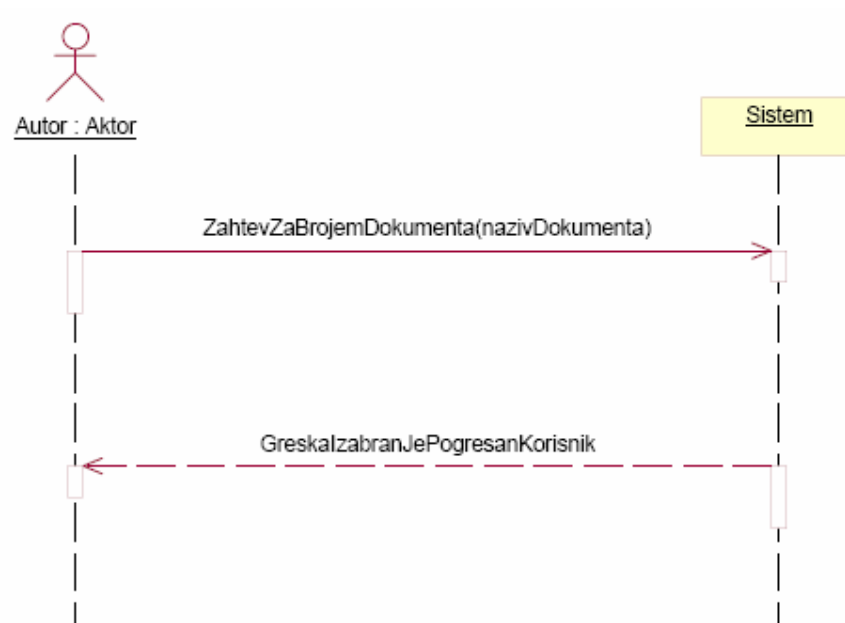
1. Autor poziva sistem da prikaže formu za slanje poruke.
2. Autor popunjava formu i šalje je Arhivaru.
3. Sistem prikazuje Autoru poruku o uspešnosti slanja poruke.
4. Arhivar poziva formu za pregled dobijenih poruka.
5. Vrši izbor nepročitane poruke za pregled.
7. Arhivar preuzima dobijenu poruku od sistema.
8. Sistem prikazuje Arhivaru poruku za novi broj dokumenta koju je dobio od Autora.
9. Arhivar odgovara na poruku tako što šalje Autoru novi broj dokumenta.
10. Sistem proverava unete podatke.
11. Sistem prihvata unete podatke.
12. Sistem prikazuje poruku Arhivaru o uspešnosti slanja poruke.
13. Autor dobija poruku koja sadrži novi broj dokumenta.



Slika 26: Osnovni scenario - SK Kreiranje broja novog dokumenta

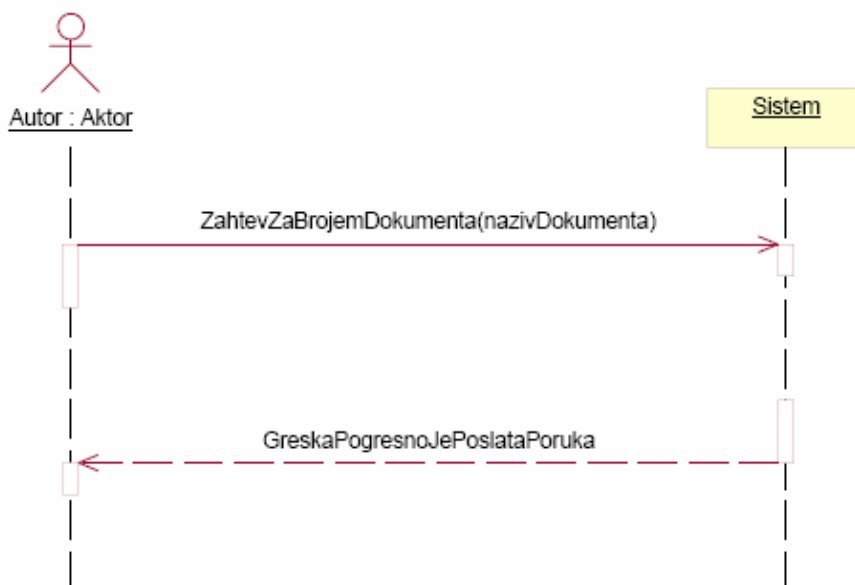
Alternativni scenariji

2.1. Ako je Autor izabrao pogrešnog korisnika (umesto Arhivara) desiće se poruka o grešci.



Slika 27: Alternativni scenario 1 - SK Kreiranje broja novog dokumenta

3.1. Ako Autor nije poslao ispravno poruku Arhivar neće primiti poruku.



Slika 28: Alterativni scenario 2 - SK Kreiranje broja novog dokumenta

9.1. Ako Arhivar nije uneo ispravno podatke Sistem će prijaviti grešku.



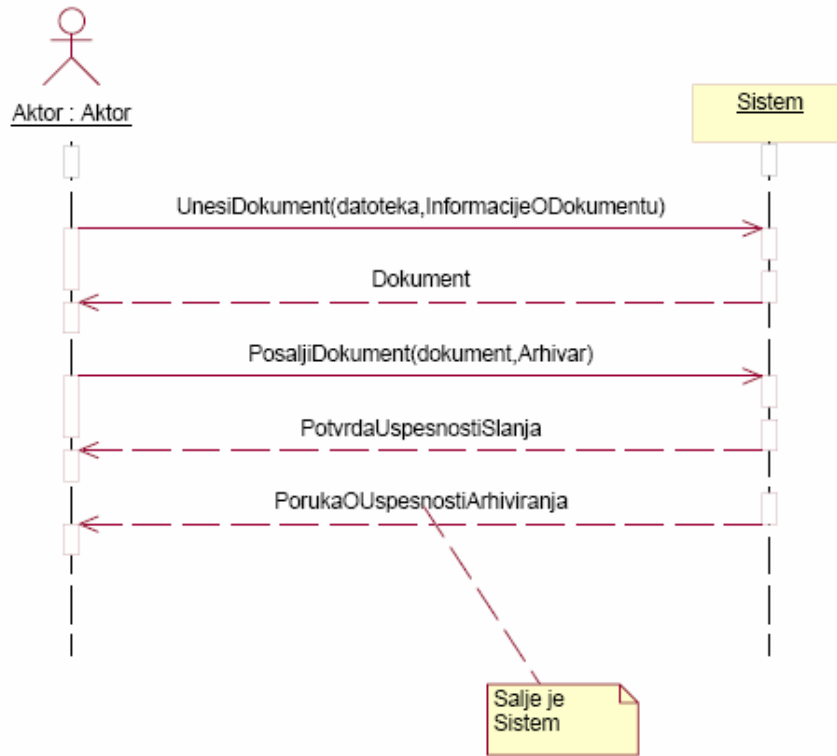
Slika 29: Alterativni scenario 3 - SK Kreiranje broja novog dokumenta

DS8: Dijagrami sekvenci slučaja korišćenja - Slanje dokumenta u arhivu

Osnovni scenario:

1. Autor poziva sistem da prikaže formu za slanje dokumenta.
2. Sistem prikazuje Autoru formu za slanje dokumenta (za izbor datoteke).
3. Autor vrši izbor datoteke na lokalnom (mrežnom) disku.
4. Sistem proverava da li datoteka postoji.
5. Sistem prihvata datoteku.
6. Sistem prikazuje formu za unos podataka (detalja) o dokumentu.
7. Autor unosi podatke o dokumentu.

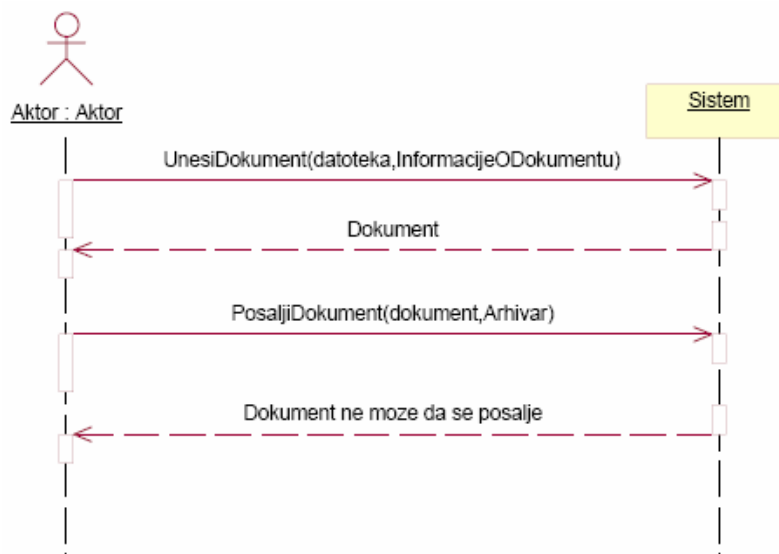
8. Autor poziva Sistem da prihvati podatke o dokumentu.
9. Sistem proverava unete podatke.
10. Sistem prihvata unete podatke.
11. Sistem prikazuje poruku Autoru o uspešnosti slanja dokumenta.



Slika 30: Osnovni scenario - SK Slanje dokumenta u arhivu

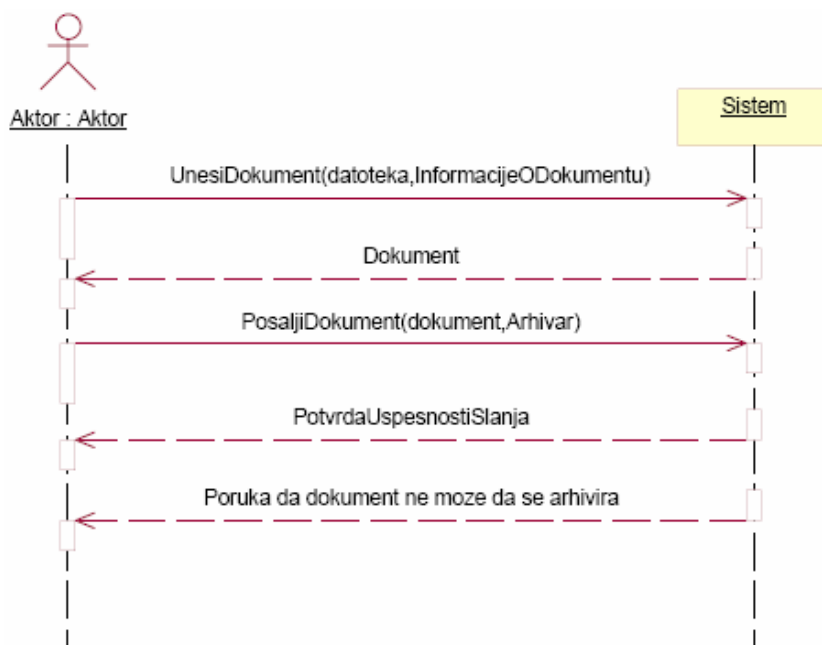
Alternativni scenariji

4.1. Ako je Autor izabrao datoteku koju je neko u međuvremenu izbrisao scenario će se zaustaviti.



Slika 31: Alternativni scenario 1 - SK Slanje dokumenta u arhivu

10.1. Ako Autor nije uneo sve potrebne i tačne podatke o dokumentu Sistem će prijaviti grešku i ispisati poruku.

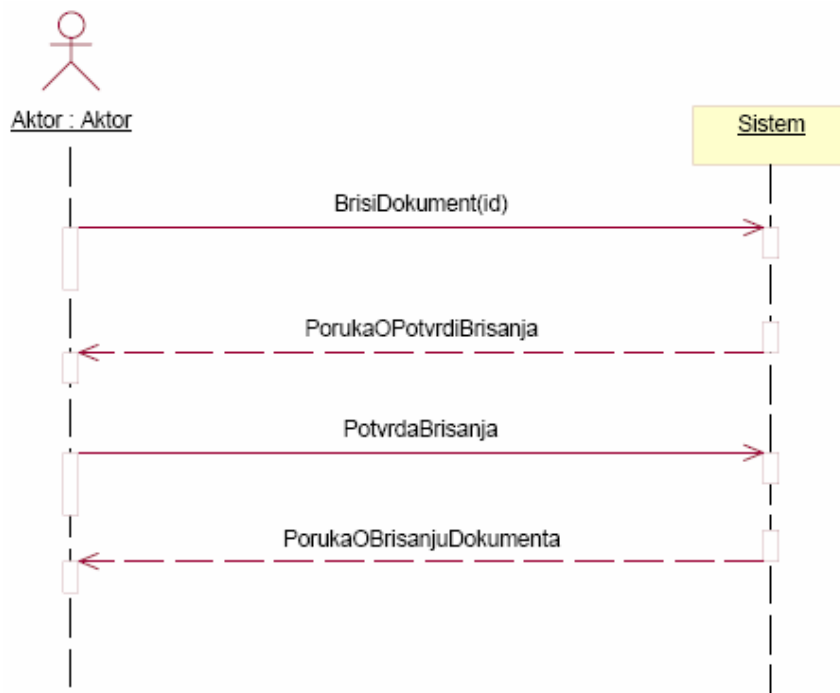


Slika 32: Alternativni scenario 2 - SK Slanje dokumenta u arhivu

DS9: Dijagrami sekvenci slučaja korišćenja - Brisanje dokumenta

Osnovni scenario:

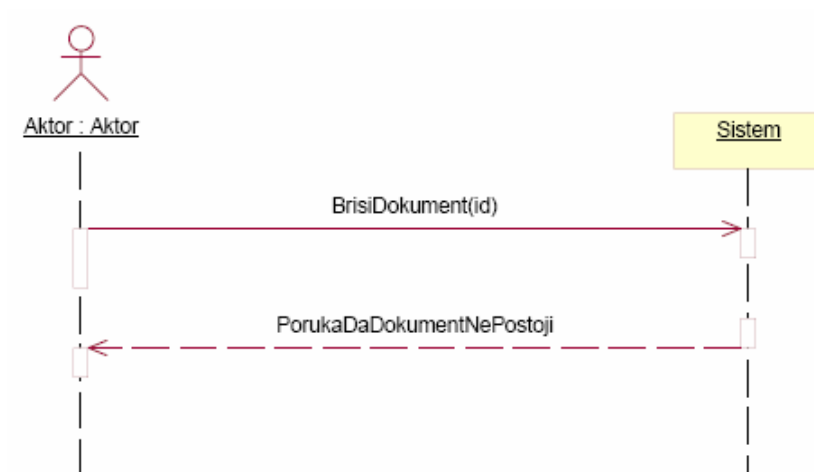
1. Administrator poziva sistem da prikaže formu za pregled dokumenata
2. Sistem prikazuje formu za pregled dokumenata.
3. Administrator bira dokument koji hoće da briše.
4. Sistem prikazuje dokument i formu sa opcijama za rad sa dokumentom.
5. Administrator bira opciju "Briši dokument".
6. Sistem proverava da li dokument pripada nekom objektu ili grupi dokumenata.
7. Sistem traži od Administratora potvrdu brisanja dokumenta.
8. Autor potvrđuje brisanje dokumenta.
9. Sistem proverava potvrdu brisanja.
10. Sistem prihvata potvrdu brisanja.
11. Sistem briše dokument i prikazuje poruku Administratoru o uspešnosti brisanja dokumenta.



Slika 33: Osnovni scenario - SK Brisanje dokumenta

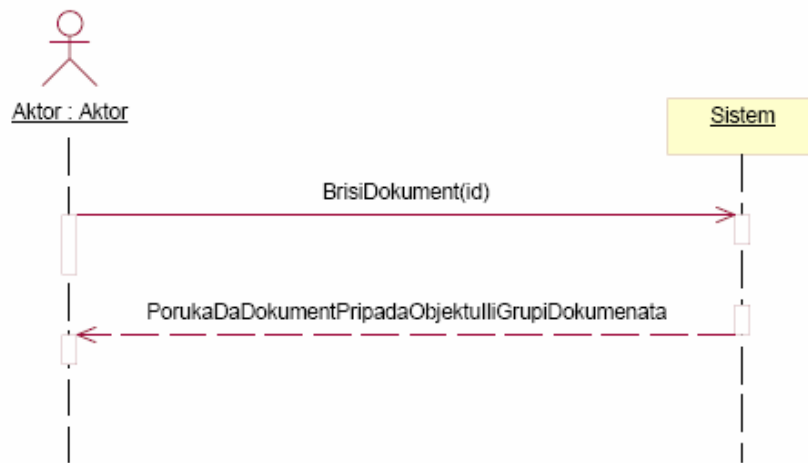
Alternativni scenariji

3.1. Ako dokument ne postoji u Sistemu prekida se izvršenje scenarija.



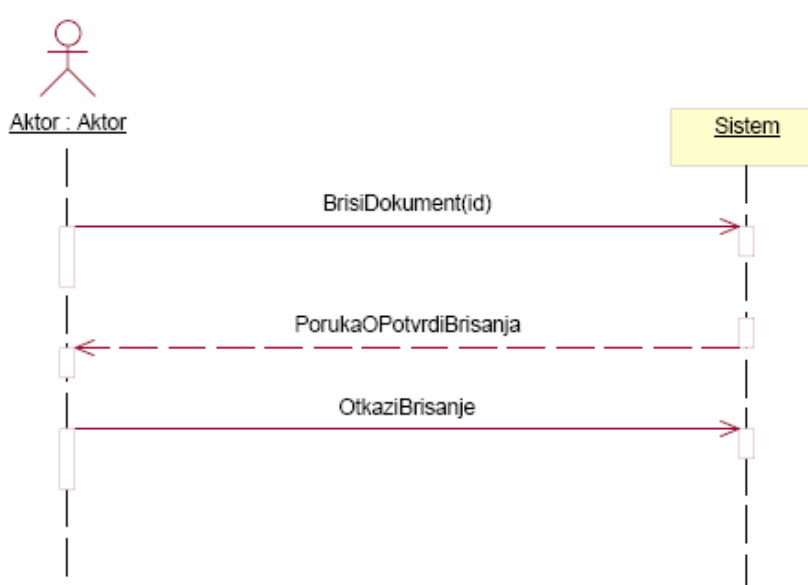
Slika 34: Alternativni scenario 1 - SK Brisanje dokumenta

6.1. Ako dokument pripada nekom objektu ili grupi dokumenta Sistem prijavljuje grešku.



Slika 35: Alternativni scenario 2 - SK Brisanje dokumenta

3. Ako Autor ne potvrdi brisanje dokumenta scenario se prekida.

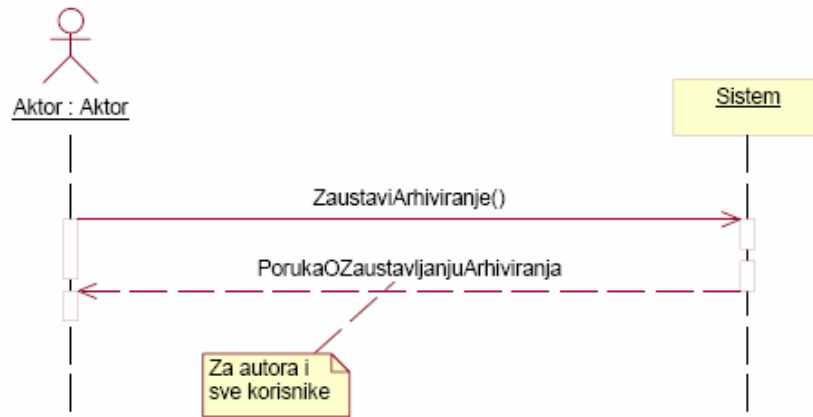


Slika 36: Alternativni scenario 3 - SK Brisanje dokumenta

DS10: Dijagrami sekvenci slučaja korišćenja - Zaustavljanje arhiviranja

Osnovni scenario:

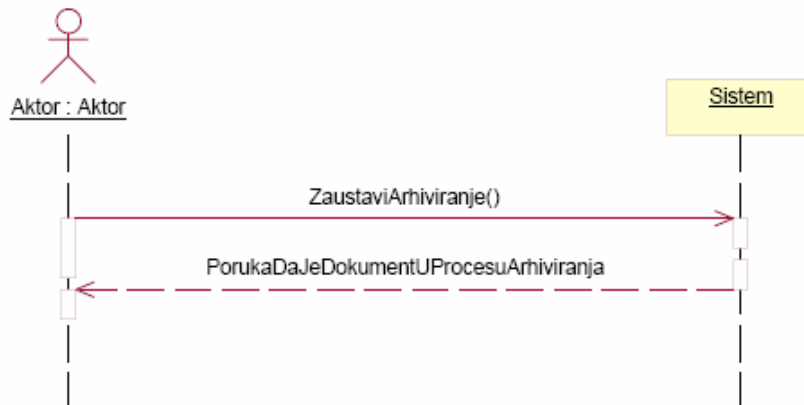
1. Administrator poziva sistem da prikaže formu za administraciju sistema.
2. Sistem prikazuje formu za administraciju sistema.
3. Administrator bira opciju "Zaustavi arhiviranje".
4. Sistem proverava da li je neki Autor ili Arhivar trenutno u procesu arhiviranja.
5. Sistem prihvata izbor i zaustavlja arhiviranje.
6. Administrator i svi aktivni korisnici dobijaju poruku da je arhiviranje zaustavljeno.



Slika 37: Osnovni scenario - SK Zaustavljanje arhiviranja

Alternativni scenariji

4.1. Ako je neki dokument trenutno u procesu arhiviranja, tada se arhiviranje neće moći zaustaviti.

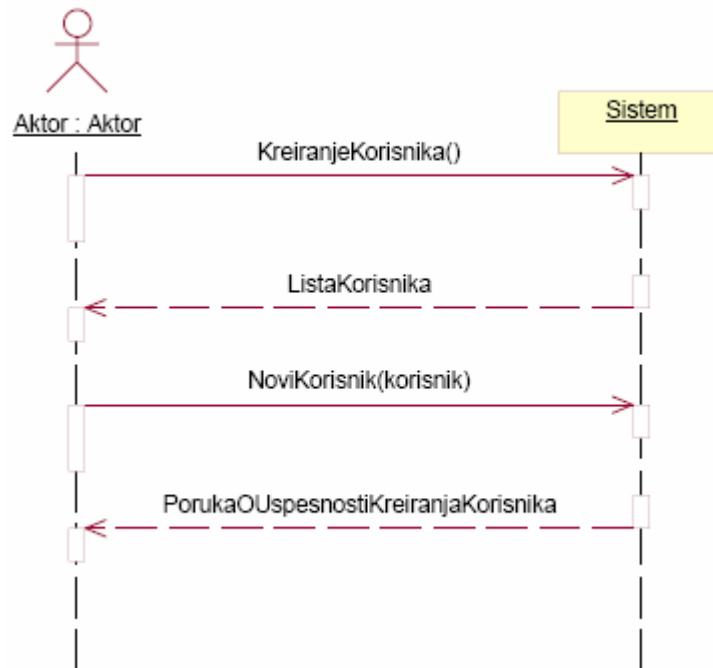


Slika 38: Alternativni scenario 1 - SK Zaustavljanje arhiviranja

DS11: Dijagrami sekvenci slučaja korišćenja - Administracija korisnika**Osnovni scenario:**

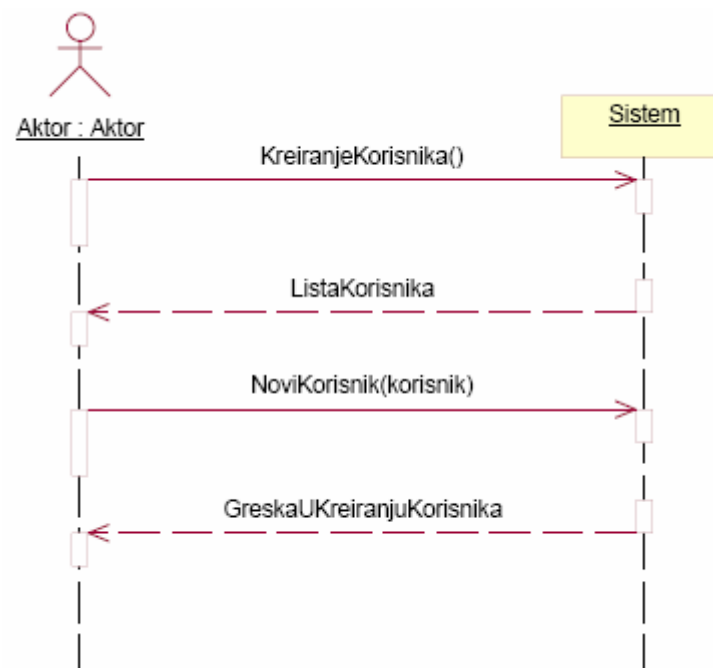
1. Administrator poziva sistem da prikaže formu za administraciju korisnika.
2. Sistem prikazuje formu za administraciju korisnika.
3. Administrator bira jednog korisnika.
4. Sistem prikazuje formu za rad sa izabranim korisnikom.
5. Administrator bira opciju za kreiranje novog, izmenu ili brisanje postojećeg korisnika.
6. Administrator unos nove ili menja postojeće podatke.
7. Arhivar poziva sistem da prihvati podatke o korisniku.
8. Sistem proverava unete podatke.
9. Sistem prihvata unete podatke.
10. Sistem prikazuje poruku Administratoru o uspešnosti kreiranja (izmene ili brisanja) korisnika i upisuje tu informaciju u log datoteku.

Alternativni scenariji



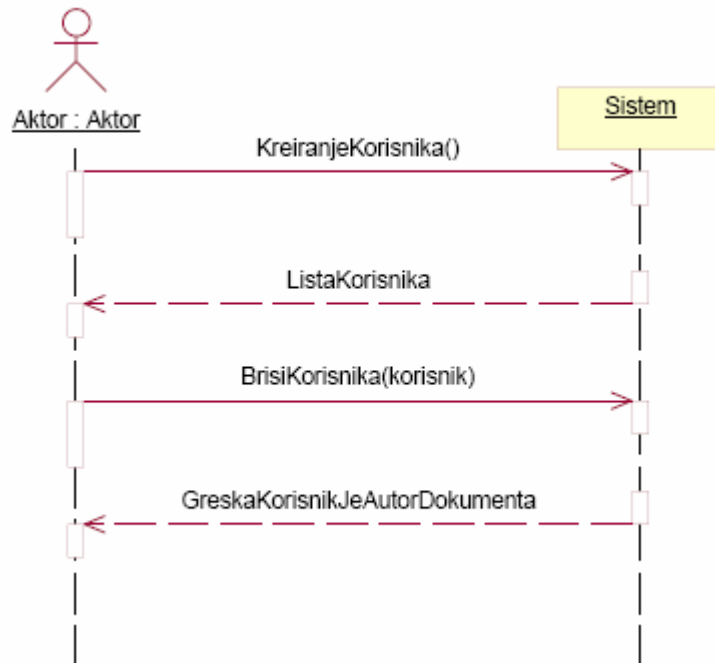
Slika 39: Osnovni scenario - SK Administracija korisnika

7.1. Ako postoji korisnik sa već postojećim korisničkim imenom prekinuće se izvršavanje scenarija.



Slika 40: Alternativni scenario 1 - SK Administracija korisnika

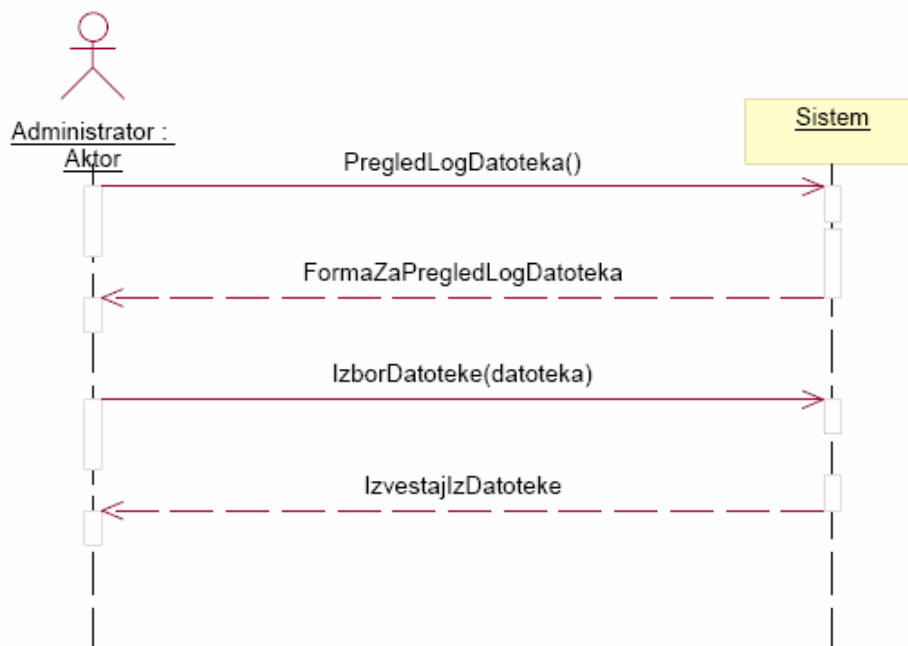
7.1.2. Ako je korisnik autor nekog dokumenta kod brisanja će se javiti greška i biće prikazana poruka.



Slika 41: Alternativni scenario 2 - SK Administracija korisnika

DS12: Dijagrami sekvenci slučaja korišćenja - Pregled log datoteka**Osnovni scenario:**

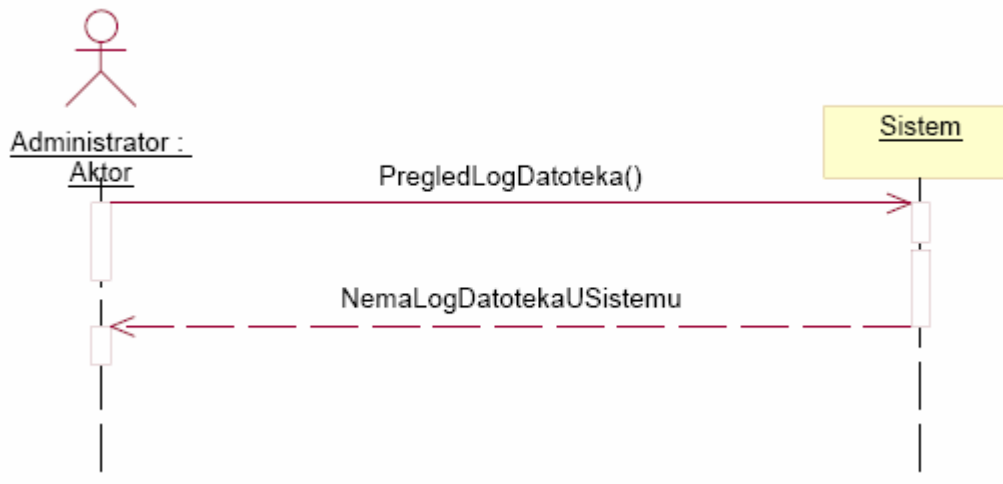
1. Administrator poziva sistem da prikaže formu za pregled log datoteka.
2. Sistem prikazuje formu za pregled log datoteka.
3. Administrator bira opcije za pregled datoteka i konkretnu datoteku.
4. Sistem prikazuje rezultat na ekranu gde se nalazi naziv korisnika, datum i operacija koju je izvršio u sistemu.



Slika 42: Osnovni scenario - SK Pregled log datoteka

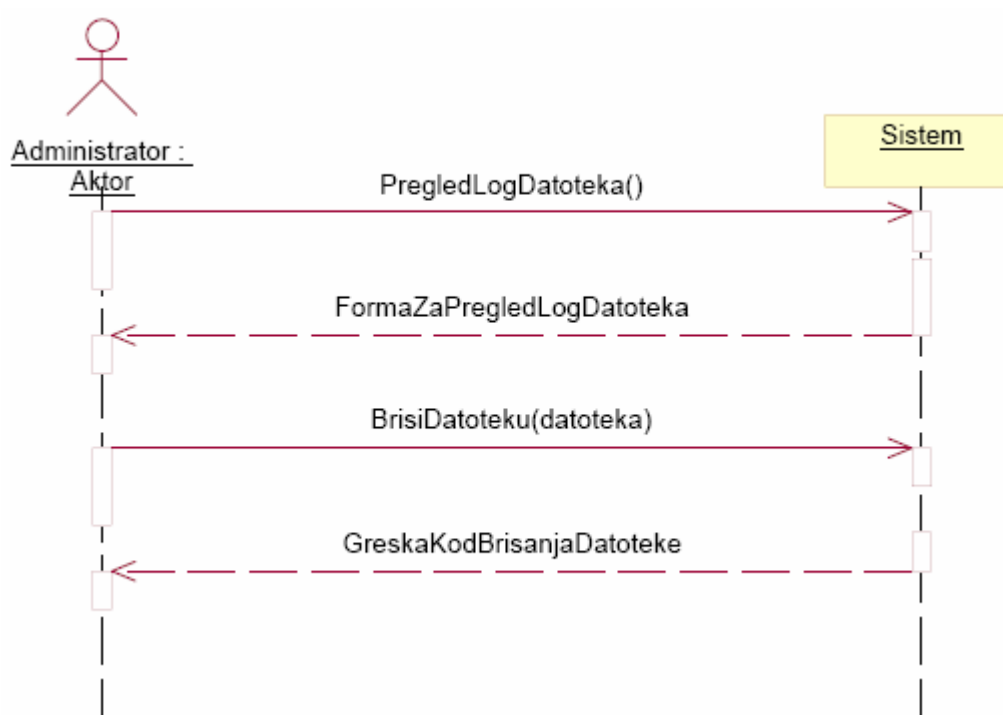
Alternativni scenariji

3.1. Ako nema istorije u sistemu sistem će prijaviti to porukom.



Slika 43: Alternativni scenario 1 - SK Pregled log datoteka

4.1. Ako datoteka ne postoji a Administrator pokuša da je izbriše dobiće poruku o grešci.



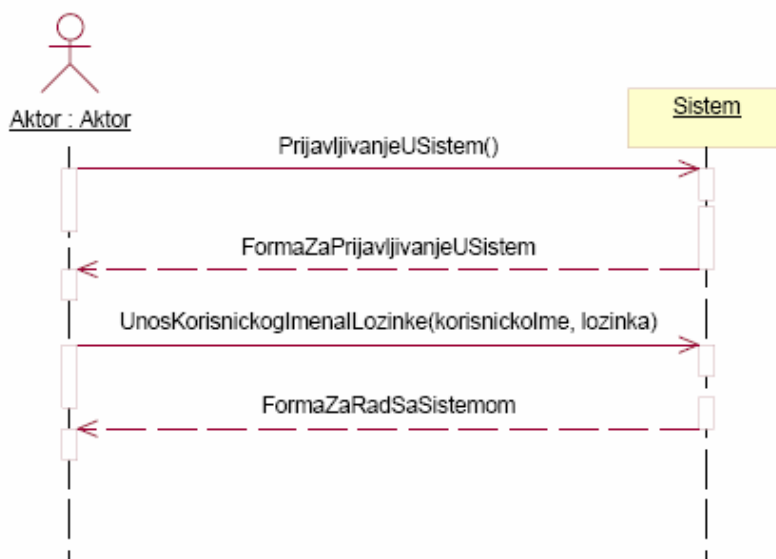
Slika 44: Alternativni scenario 2 - SK Pregled log datoteka

DS13: Dijagrami sekvenci slučaja korišćenja - Prijava u sistem

Osnovni scenario:

1. Korisnik poziva sistem da prikaže formu za prijavljivanje u sistem.
2. Sistem prikazuje formu za prijavljivanje u sistem.
3. Korisnik unosi korisničko ime i lozinku.
4. Korisnik poziva sistem da prihvati unesene podatke.
5. Sistem proverava unete podatke.

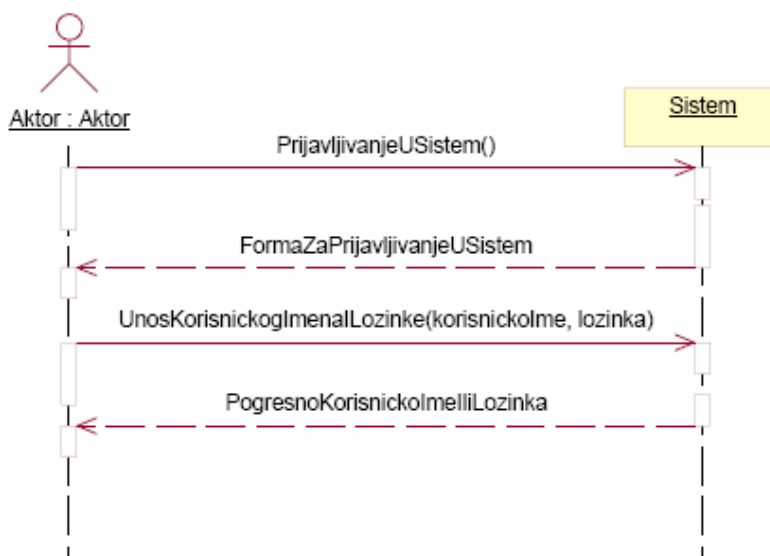
6. Sistem prihvata unete podatke.
7. Sistem prikazuje osnovnu formu za rad sa sistemom prilagođenu grupi kojoj korisnik pripada.



Slika 45: Osnovni scenario - SK Prijava u sistem

Alternativni scenariji

- 5.1. Ako je unešeno pogrešno korisničko ime ili lozinka prekida se izvršenje scenarija.



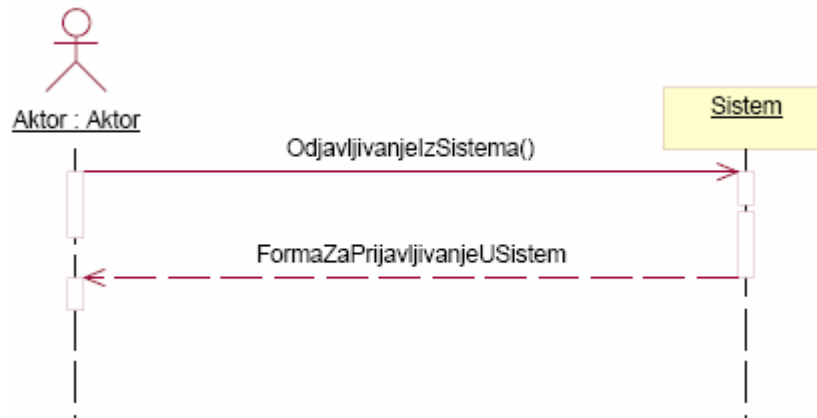
Slika 46: Alternativni scenario 1 - SK Prijava u sistem

DS14: Dijagrami sekvenci slučaja korišćenja - Odjava iz sistema

Osnovni scenario:

1. Korisnik poziva sistem da se odjavi iz njega.
2. Sistem proverava da li je korisnik prijavljen.
3. Sistem prihvata zahtev od korisnika.

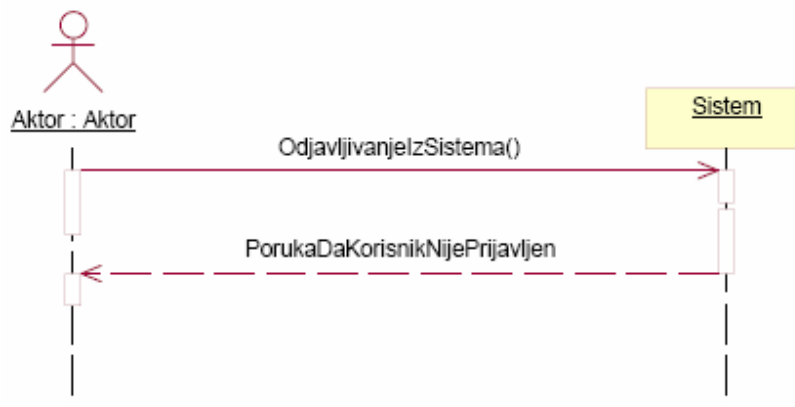
4. Sistem upisuje informaciju o odjavljivanju korisnika u log datoteku.
5. Sistem prikazuje korisniku formu za prijavljivanje i odjavnu poruku.



Slika 46: Osnovni scenario - SK Odjava iz sistema

Alternativni scenariji

2.1. Ako korisnik nije prijavljen ili mu je isteklo vreme za rad sa sistemom prekida se izvršenje scenarija.



Slika 47: Alternativni scenario 1 - SK Odjava iz sistema

2.2.2. Rezultat analize sistemskog dijagrama sekvenci (SO)

Kao rezultat analize dobijene su sledeće sistemske operacije (SO) koje treba projektovati:

1. Dokument **UnesiDokument**(datoteka, InformacijeODokumentu);
2. PotvrdaUspesnostiSlanja **PosaljiDokument**(dokument, Arhivar);
3. ListaDokumenata **Arhiviranje**();
4. IzmenjeniDokument **IzmenaInformacijaODokumentu**(dokument);
5. DokumentPovezanSaObjektom **PoveziDokumentSaObjektom**(objekat, dokument);
6. DokumentPovezanSaGrupomDokumenata **PoveziDokumentSaGrupom**(dokument, grupaDokumenata);
7. ListaObjekata **PregledObjekata**();
8. KreiraniObjekat **KreirajObjekat**(objekat);
9. ObjekatUHijerarhiji **PoveziSaObjektom**(objekatVisiNivo, objekat);
10. FormaZaPretraguDokumenata **Pretrazi**();
11. ListaGrupaDokumenata **PregledGrupaDokumenata**();
12. GrupaDokumenata **PoveziSaObjektom**(grupaDokumenata, objekat);
13. KreiranaGrupaDokumenata **KreirajGrupuDokumenata**(grupaDokumenata);
14. ListaDokumenata **PronadjiDokument**(atribut);
15. ListaDokumenata **PronadjiDokumenteZaObjekat**(objekat);
16. FormaZaPregledDokumenata **Pregled**();
17. ListaKorisnika **KreiranjeKorisnika**();
18. PorukaOUspesnostiKreiranjaKorisnika **NoviKorisnik**(korisnik);
19. Dokument **UzmiDokument**(id);
20. PorukaOUspesnostiIzmene **ZameniDokument**(dokument);
21. BrojDokumenta **ZahtevZaBrojemDokumenta**(nazivDokumenta);
22. PorukaOPotvrdiBrisanja **BrisiDokument**(id);
23. PorukaOBrisanjuDokumenta **PotvrdaBrisanja**();
24. PorukaOZaustavljanjuArhiviranja **ZaustaviArhiviranje**();
25. FormaZaPregledLogDatoteka **PregledLogDatoteka**();
26. IzvestajIzDatoteke **IzborDatoteke**(datoteka);
27. FormaZaPrijavlivanjeUSistem **PrijavlivanjeUSistem**();
28. FormaZaRadSaSistemom **UnosKorisnickogImenaILOzinke**(korisnickoIme, lozinka);
29. PorukaDaKorisnikNijePrijavljen **OdjavljivanjeIzSistema**();

2.2.3. Definisane ugovora o sistemskim operacijama

Sistemska operacija opisuje ponašanje sistema. Ona je javna i njoj se može pristupiti iz okruženja softverskog sistema. Ugovori se prave za sistemske operacije i opisuju njeno ponašanje. Ugovori opisuju šta sistemska operacija treba da radi, a ne kako. Sekcije ugovora uključuju:

- Operaciju – Ime operacije i njeni ulazni argumenti;
- Veza sa SK – imena SK u kojima se poziva sistemska operacija;
- Preduslov – sistem mora biti u odgovarajućem stanju pre izvršenja sistemske operacije;
- Postuslov – sistem mora biti u odgovarajućem stanju posle izvršenja sistemske operacija ili se poništava rezultat sistemske operacije.

UGOVOR UG1: UnesiDokument

Operacija: UnesiDokument(datoteka:File, InformacijeODokumentu:Dokument, dokument:Dokument) : signal;

Veza sa SK: DS8

Preduslovi: Korisnik je prijavljen u sistem.

Postuslovi: Dokument je unesen u sistem.

UGOVOR UG2: PosaljiDokument

Operacija: PosaljiDokument(dokument:Dokument, Arhivar:Korisnik, PotvrdaUspesnostiSlanja:String) : signal;

Veza sa SK: DS8

Preduslovi: Dokument je unesen u sistem.

Postuslovi: Dokument je poslat u Arhivu.

UGOVOR UG3: Arhiviranje

Operacija: Arhiviranje(dokumenti []:Dokument) : signal;

Veza sa SK: DS1

Preduslovi: -

Postuslovi: Dokument je arhiviran.

UGOVOR UG4: IzmenaInformacijaODokumentu

Operacija: IzmenaInformacijaODokumentu(dokument:Dokument) : signal;

Veza sa SK: DS1

Preduslovi: Dokument je izabran.

Postuslovi: Dokument je izmenjen.

UGOVOR UG5: PoveziDokumentSaObjektom

Operacija: PoveziDokumentSaObjektom(objekat:Objekat, dokument:Dokument) : signal;

Veza sa SK: DS1

Preduslovi: Objekat postoji u sistemu.

Postuslovi: Dokument je povezan sa objektom.

UGOVOR UG6: PoveziDokumentSaGrupom

Operacija: PoveziDokumentSaGrupom(dokument:Dokument, grupaDokumenata:GrupaDokumenata) : signal;

Veza sa SK: DS1

Preduslovi: Grupa dokumenata postoji u sistemu.

Postuslovi: Dokument je povezan sa grupom dokumenata;

UGOVOR UG7: PregledObjekata

Operacija: PregledObjekata(Objekat []obj) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: -

UGOVOR UG8: KreirajObjekat

Operacija: **KreirajObjekat**(objekat:Objekat) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: Objekat je kreiran u sistemu.

UGOVOR UG9: PoveziSaObjektom

Operacija: **PoveziSaObjektom**(objekatVisi:Objekat, ObjekatNizi:Objekat) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: Objekat je povezan sa drugim objektom, visim u hijerarhiji.

UGOVOR UG10: Pretrazi

Operacija: **Pretrazi**(objekat:Objekat) : signal;

Veza sa SK: DS2

Preduslovi: -

Postuslovi: Kreirana je forma za pretragu dokumenata.

UGOVOR UG11: PregledGrupaDokumenata

Operacija: **PregledGrupaDokumenata()** : signal;

Veza sa SK: DS6

Preduslovi: -

Postuslovi: Vraća sve grupe dokumenata u sistemu.

UGOVOR UG12: PoveziSaObjektom

Operacija: **PoveziSaObjektom**(grupaDokumenata:GrupaDokumenata, objekat:Objekat) : signal;

Veza sa SK: DS6

Preduslovi: U sistemu moraju postojati grupa dokumenata i objekat koji se povezuju.

Postuslovi: Grupa dokumenata je vezana za objekat.

UGOVOR UG13: KreirajGrupuDokumenata

Operacija: **KreirajGrupuDokumenata**(grupaDokumenata:grupaDokumenata) : signal;

Veza sa SK: DS6

Preduslovi: Prikazane su sve grupe dokumenata.

Postuslovi: Kreira novu grupu dokumenata.

UGOVOR UG14: PronadjiDokument

Operacija: **PronadjiDokument**(atribut:String, Dokument [] dokumenti) : signal;

Veza sa SK: DS2

Preduslovi: -

Postuslovi: Dokumenti koji su pronađeni mogu da se pogledaju.

UGOVOR UG15: PronadjiDokumenteZaObjekat

Operacija: **PronadjiDokumenteZaObjekat**(objekat:Objekat, Dokument [] dokumenti) : signal;

Veza sa SK: DS2

Preduslovi: Objekat mora da postoji u sistemu.

Postuslovi: Dokumenti koji su pronađeni mogu da se pogledaju.

UGOVOR UG16: Pregled

Operacija: **Pregled()** : signal;

Veza sa SK: DS3

Preduslovi: -

Postuslovi: Kreirana je forma za pregled dokumenata.

UGOVOR UG17: KreiranjeKorisnika

Operacija: **KreiranjeKorisnika**(Korisnik [] korisnici) : signal;

Veza sa SK: DS11

Preduslovi: Mora postojati bar jedan korisnik u sistemu (administrator).

Postuslovi: Preuzeta je lista svih korisnika iz sistema.

UGOVOR UG18: NoviKorisnik

Operacija: **NoviKorisnik**(korisnik:Korisnik, poruka:String) : signal;

Veza sa SK: DS11

Preduslovi: -

Postuslovi: Kreiran je novi korisnik u sistemu.

UGOVOR UG19: UzmiDokument

Operacija: **UzmiDokument**(id:integer, dokument:Dokument) : signal;

Veza sa SK: DS4

Preduslovi: Dokument mora da postoji u sistemu.

Postuslovi: Dokument je preuzet iz sistema.

UGOVOR UG20: ZameniDokument

Operacija: **ZameniDokument**(dokument:Dokument, poruka:String) : signal;

Veza sa SK: DS4

Preduslovi: Dokument mora da postoji u sistemu.

Postuslovi: Dokument u sistemu je zamenjen sa postojećim.

UGOVOR UG21: ZahtevZaBrojemDokumenta

Operacija: **ZahtevZaBrojemDokumenta**(nazivDokumenta:String, brojDokumenta:String) : signal;

Veza sa SK: DS7

Preduslovi: -

Postuslovi: Dobijen je novi broj dokumenta.

UGOVOR UG22: BrisiDokument

Operacija: BrisiDokument(id:integer, poruka:String) : signal;

Veza sa SK: DS9

Preduslovi: Dokument mora postojati u sistemu.

Postuslovi: -.

UGOVOR UG23: PotvrdaBrisanja

Operacija: PotvrdaBrisanja(poruka:String) : signal;

Veza sa SK: DS9

Preduslovi: Dokument mora biti u režimu brisanja i ne sme biti vezan za neki objekat ili grupu dokumenata.

Postuslovi: Dokument je izbrisan iz sistema.

UGOVOR UG24: ZaustaviArhiviranje

Operacija: ZaustaviArhiviranje(poruka:String) : signal;

Veza sa SK: DS10

Preduslovi: Ni jedan dokument ne sme biti u režimu arhiviranja.

Postuslovi: Arhiviranje dokumenata je zaustavljeno.

UGOVOR UG25: PregledLogDatoteka

Operacija: PregledLogDatoteka() : signal;

Veza sa SK: DS12

Preduslovi: Log datoteke moraju postojati.

Postuslovi: Vraćena je forma za izbor log datoteke.

UGOVOR UG26: IzborDatoteke

Operacija: IzborDatoteke () : signal;

Veza sa SK: DS12

Preduslovi: Prikazana je forma za izbor log datoteke.

Postuslovi: Prikazan je izveštaj iz izabrane log datoteke.

UGOVOR UG27: PrijavljivanjeUSistem

Operacija: PrijavljivanjeUSistem() : signal;

Veza sa SK: DS13

Preduslovi: -.

Postuslovi: Prikazana je forma za prijavljivanje u sistem.

UGOVOR UG28: UnosKorisnickogImenaILozinke

Operacija: UnosKorisnickogImenaILozinke(korisnickoIme:String, lozinka:String) : signal;

Veza sa SK: DS13

Preduslovi: Prikazana je forma za prijavljivanje u sistem.

Postuslovi: Korisnik je prijavljen u sistem i prikazana mu je forma za rad sa sistemom.

UGOVOR UG29: OdjavljivanjeIzSistema

Operacija: `OdjavljivanjeIzSistema()` : signal;

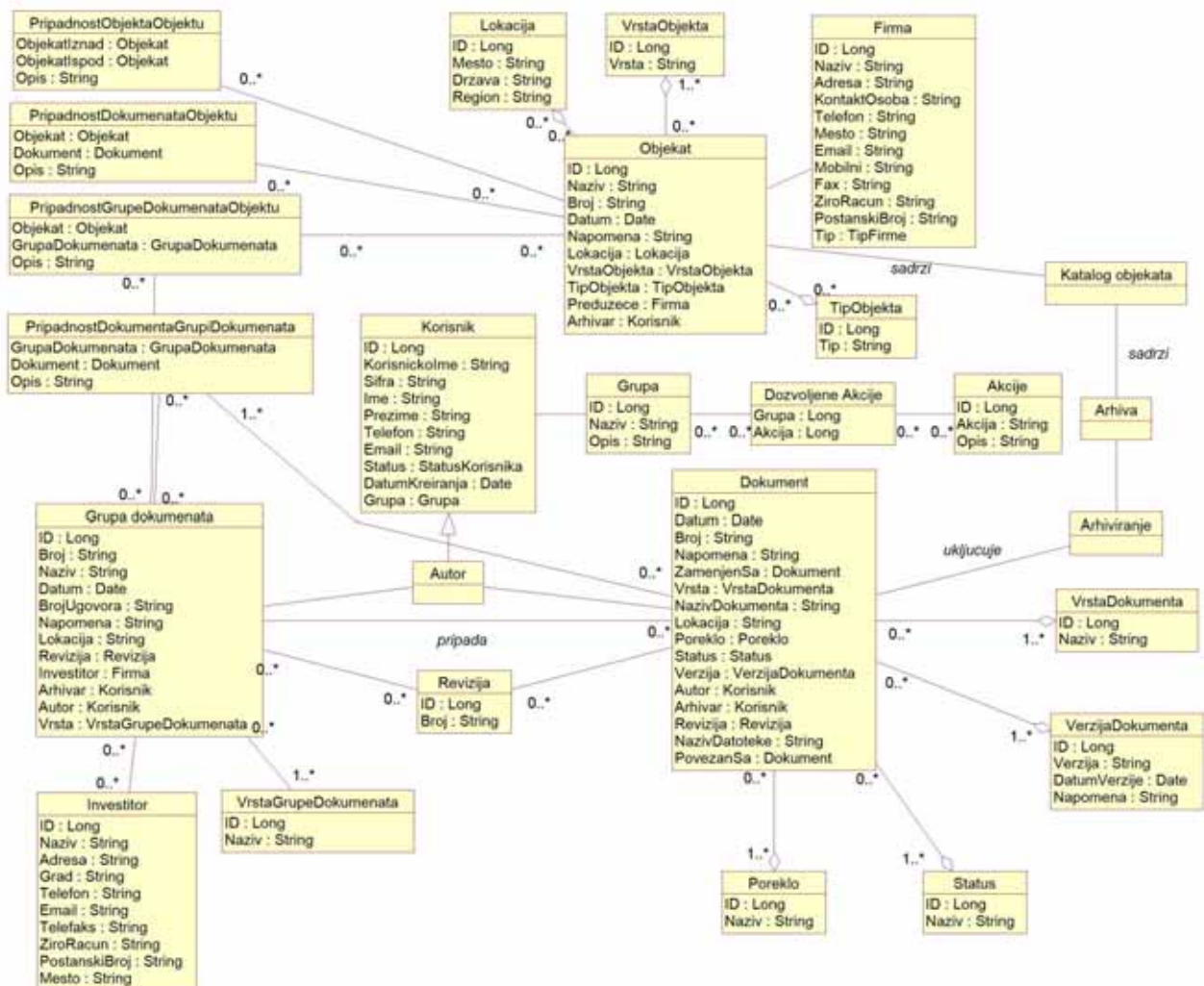
Veza sa SK: DS14

Preduslovi: Korisnik je prijavljen u sistem.

Postuslovi: Korisnik je odjavljen iz sistema i prikazana mu je forma za prijavljivanje.

2.2.4. Domenski model sa tipiziranim atributima

Za svaki atribut domenskih entiteta se određuje konceptulani tip.



Slika 48: Domenski model sa tipiziranim atributima.

2.2.5. Relacioni model

Na osnovu domenskog modela se pravi relacioni model, što je i prikazano na slici 49.

Vezano za objekat:

Objekat (ID, Naziv, Broj, Datum, Napomena, Lokacija, Vrsta, TipObjekta, Firma, Arhivar);

Lokacija (ID, Mesto, Drzava, Region)

VrstaObjekta (ID, Vrsta);

Firma (ID, Naziv, Adresa, KontaktOsoba, Telefon, Mesto, Email, Mobilni, Fax, ZiroRacun, PostanskiBroj, Tip);

TipFirme (ID, Tip);

TipObjekta (ID, Tip);

Vezano za dokument:

Dokument (ID, Datum, Broj, Napomena, ZamenjenSa, VrstaDokumenta, NazivDokumenta, NazivDatoteke, Lokacija, Autor, Arhivar, VerzijaDokumenta, Status, Poreklo, Revizija);

VerzijaDokumenta (ID, Verzija, DatumVerzije, Napomena);

VrstaDokumenta (ID, Naziv);

Status (ID, Naziv);

Poreklo (ID, Naziv);

Revizija (ID, Broj);

Vezano za grupu dokumenata:

GrupaDokumenata (ID, Broj, Naziv, Datum, BrojUgovora, Napomena, Revizija, Autor, Arhivar, Investitor, Vrsta, Lokacija);

VrstaGrupeDokumenata (ID, Naziv);

Veze izmedju objekata i dokumenata:

PripadnostObjektaObjektu (ObjekatIznad, ObjekatIspod, Opis);

PripadnostDokumenataObjektu (Objekat, Dokument, Opis);

PripadnostGrupeDokumenataObjektu (Objekat, GrupaDokumenata, Opis);

PripadnostDokumentaGrupiDokumenata (GrupaDokumenata, Dokument, Opis);

Vezano za korisnike:

Korisnik (ID, KorisnickoIme, Sifra, Ime, Prezime, Telefon, Email, DatumKreiranja, Grupa, Status);

StatusKorisnika (ID, Status);

Grupa (ID, Naziv, Opis);

Vezano za akcije:

Akcije (ID, Akcija, Opis);

DozvoljeneAkcije (Grupa, Akcija);

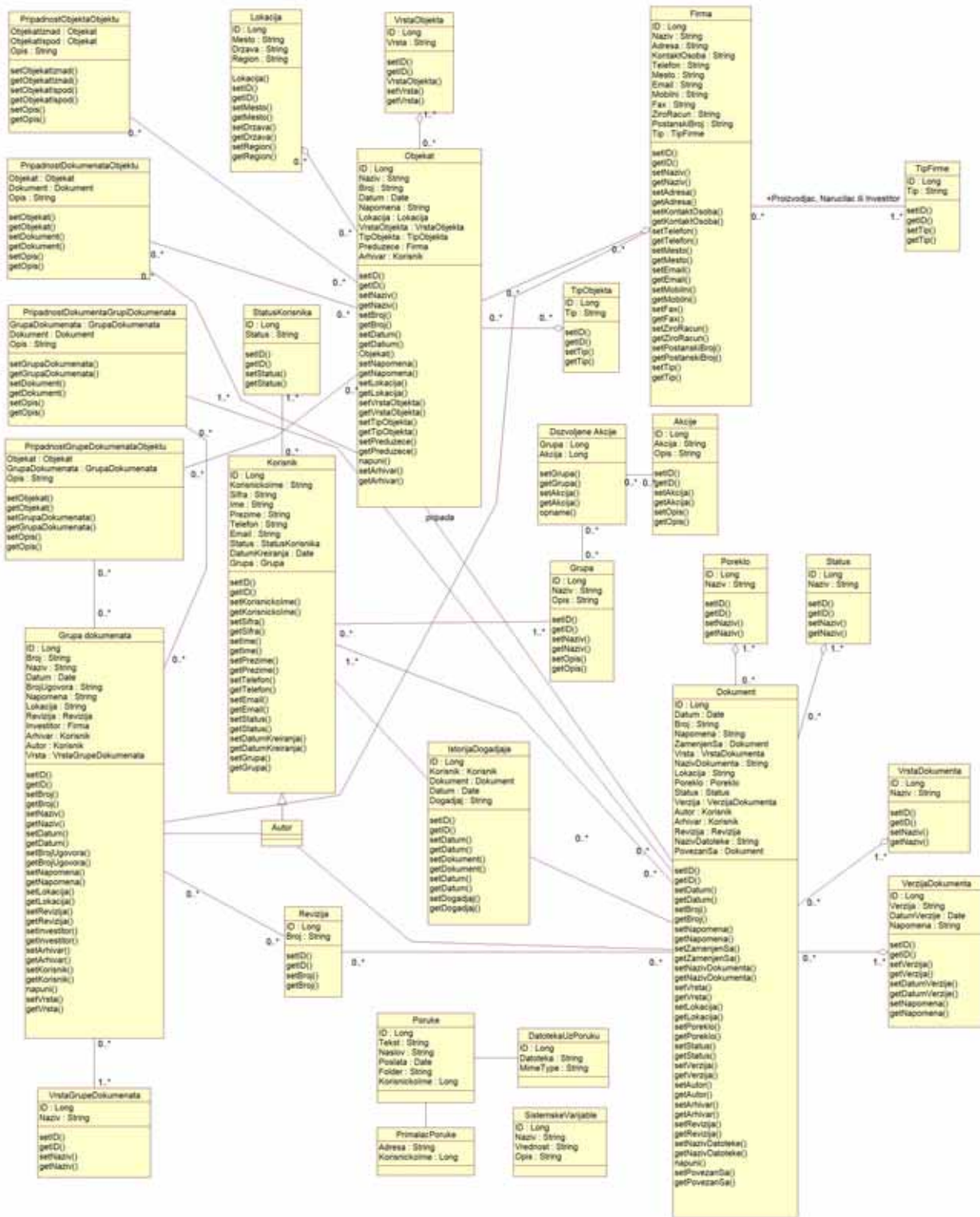
Vezano za sistem:

SistemskeVarijable (ID, Naziv, Vrednost, Opis);

Atributi koji su podvučeni predstavljaju primarni ključ. U daljem tekstu će biti detaljno opisani elementi relacionog modela.

2.2.5. Dijagram klasa strukture

Na osnovu domenskog modela se pravi dijagram klasa strukture:



Slika 49: Dijagram klasa strukture

2.2.6. Klase strukture

Dijagram klasa strukture se sastoji iz sledećih klasa¹ strukture:

```
public class Akcije {  
  
    private long ID;  
    private String Akcija;  
    private String Opis;  
    private long roditeljAkcija;  
    private String akcijaIkonica;  
    private String akcijaPutanja;  
    private String akcijaUradi;  
  
    ...  
  
    /**  
     * Vrednosni objekat koji preslikava tabelu Dokumenti  
     * @author Milan Milanović  
     * @version 1.0  
     */  
    public class Dokument {  
  
        private long ID;  
        private Date datum;  
        private String broj;  
        private String napomena;  
        private long zamenjenSa;  
        private long povezanSa;  
  
        ...  
  
    public class DozvoljeneAkcije {  
  
        private long grupa;  
        private long akcija;  
  
        ...  
  
    public class Firma {  
  
        private long ID;  
        private String Naziv;  
        private String Adresa;  
        private String KontaktOsoba;  
        private String Telefon;  
        private String Mesto;  
  
        ...  
  
    public class Grupa {  
  
        private long ID;  
        private String Naziv;  
        private String Opis;  
  
        ...  
  
    public class GrupaDokumenata {  
  
        private long ID;
```

¹ Dato je samo nekoliko prvih redova klasa (atributa) bez metoda zbog uštede prostora.

```
private String broj;
private String naziv;
private Date datum;
private String brojUgovora;
...
public class IstorijaDogadjaja {

    private long ID;
    private long korisnik;
    private long dokument;
    private Date datum;
    private String dogadjaj;
    ...

public class Korisnik {

    private long ID;
    private String KorisnickoIme;
    private String Sifra;
    private String Ime;
    private String Prezime
    ...

public class Lokacija {

    private long ID;
    private String Mesto;
    private String Drzava;
    private String Region;
    ...

public class Objekat {

    private long ID;
    private String Naziv;
    private String Broj;
    private Date Datum;
    private String Napomena;
    private long lokacija;
    private long vrstaObjekta;
    ...

public class Poreklo {

    private long ID;
    private String Naziv;
    ...

public class PripadnostDokumentaGrupiDokumenata {

    private long grupaDokumenata;
    private long dokument;
    private String opis;
    ...

public class PripadnostDokumentaObjektu {

    private long objekat;
    private long dokument;
```



```
    private String opis;
...
public class PripadnostGrupeDokumenataObjektu {

    private long objekat;
    private long grupaDokumenata;
    private String opis;
...

public class PripadnostObjektaObjektu {

    private long objekatIznad;
    private long objekatIspod;
    private String opis;
...

public class Revizija {

    private long ID;
    private String Broj;
...

public class SistemskeVarijable {

    private long ID;
    private String Naziv;
    private String Vrednost;
    private String Opis;
...

public class SistemskaPoruka {

    private long ID;
    private long autor;
    private long primalac;
    private String tekstPoruke;
    private String naslov;
...

public class Status {

    private long ID;
    private String Naziv;
...

public class StatusKorisnika {

    private long ID;
    private String Status;
...

public class TipFirme {

    private long ID;
    private String Tip;
...

```

```
public class TipObjekta {  
  
    private long ID;  
    private String Tip;  
    ...  
  
public class VerzijaDokumenta {  
  
    private long ID;  
    private String Verzija;  
    private Date DatumVerzije;  
    private String Napomena;  
    ...  
  
public class VrstaDokumenta {  
  
    private long ID;  
    private String Naziv;  
    ...  
  
public class VrstaGrupeDokumenata {  
  
    private long ID;  
    private String Naziv;  
    ...  
  
public class VrstaObjekta {  
  
    private long ID;  
    private String Vrsta;  
    ...
```

2.3. PROJEKTOVANJE

2.3.1. Faza projektovanja sistema

U ovoj fazi se daje oblik celom sistemu, koji treba da podrži sve postavljene zahteve korisnika, kao i druga ograničenja. Kroz ovu fazu treba da se shvate zahtevi i ograničenja programskih jezika, komponente koje se mogu ponovo koristiti, operativni sistemi, itd. Model projektovanja treba da sadrži: interfejse, klase i podsisteme i on predstavlja ulaz u fazu implementacije. Model implementacije treba da sačuva strukturu modela projektovanja i detaljno je razradi. I odrede interfejsi između podsistema.

Radni tok projektovanja se sastoji iz sledećih faza:

- Projektovanje arhitekture;
- Projektovanje slučaja korišćenja;
- Projektovanje klasa i
- Projektovanje podsistema.

Projektovanje arhitekture pruža

- *Računarske čvorove i konfiguraciju mreže;*
Ova konfiguracija ima važan uticaj na softversku arhitekturu, tj. na distribuciju funkcija softverskog sistema na različite čvorove mreže. Tu se obično koristi tronivojski uzor, gde su klijentu u jednom, logika poslovne aplikacije u drugim a funkcije za rad sa bazom podataka u trećem nivou.
- *Podsisteme i njihove interfejse;*
Oni predstavljaju elemente modela koji obezbeđuju da se model projektovanja organizuje preko skupa podsistema kojima se može upravljati. Identifikovanje podsistema započinje na aplikaciono - specifičnom i generalnom sloju gde se podsistemi mogu jednoznačno identifikovati na osnovu paketa analize.
- *Klase projektovanja koje su značajne za arhitekturu;*
Klase projektovanja se mogu dobiti na osnovu klasa koje su dobijene u fazi analize. Takođe, klasa projektovanja je odgovorna za realizaciju nekih od sistemskih operacija.
- *Generičke mehanizme projektovanja za distribuciju, perzistentnost, zaštitu, itd;*
Ovde se razmatraju mogućnosti obrade specifičnih zahteva, koji su identifikovani u fazi analize, na osnovu postojećih tehnologija projektovanja i implementacije. Generički mehanizmi projektovanja nastaju kao rezultat ovog koraka i oni predstavljaju rešenja postavljenih specijalnih zahteva. Ovi mehanizmi se smeštaju u srednji sloj arhitekture.

Projektovanje slučaja korišćenja

Slučajevi korišćenja se projektuju kako bi se identifikovale klase projektovanja, podsistemi i interfejsi čiji su objekti potrebni da bi se realizovali slučajevi korišćenja, opisala interakcija između objekata kod realizacije slučajeva korišćenja, opisala interakcija između podsistema.

Projektovanje klasa

Klase treba da realizuju slučajeve korišćenja. Tokom projektovanja klasa se izvode sledeći koraci: pravi se nacrt klasa projektovanja, identifikuju se njihove operacije i atributi, definišu veze između njih, vrši identifikovanje generalizovanih klasa projektovanja, projektuju metode (realizacija operacija), opisuju stanja klasa i obrađuju specijalni zahtevi.

Projektovanje podsistema

Podsistemi treba da budu što je moguće više nezavisni od drugih podsistema, treba da obezbede dobre interfejsne i da sadrže korektne realizacije potrebnih operacija.

2.3.2. Arhitektura softverskog sistema

Faza projektovanja opisuje fizičku strukturu i ponašanje softverskog sistema (arhitekturu softverskog sistema). U ovom radu je primjenjena klasična trionivojska arhitektura.

Nivoi sistema su:

- Korisnički interfejsi i reprezentacija softverskog sistema;
- Aplikaciona logika, koja uključuje strukturu i ponašanje softverskog sistema;
- Skladište podataka koje čuva stanje softverskog sistema.

Aplikaciona logika uključuje:

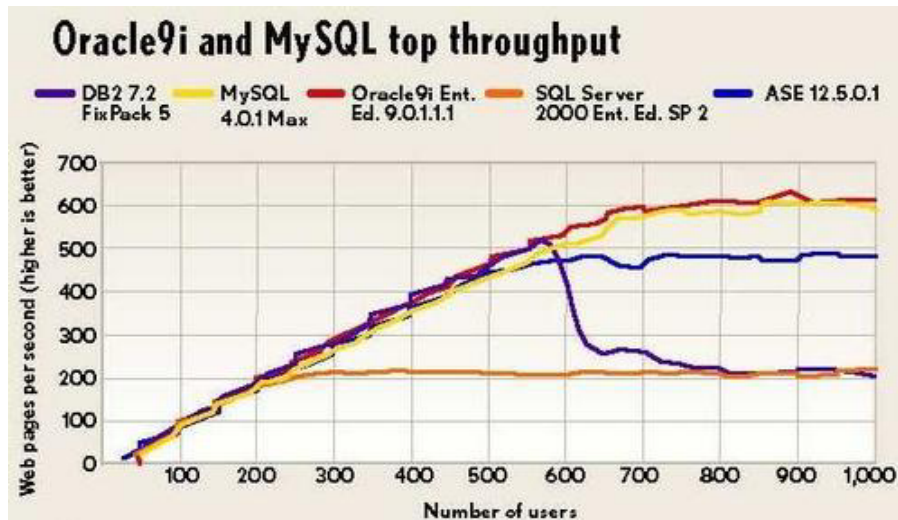
- Kontroler;
- Poslovna logika;
- Deo za komunikaciju sa nekim skladištem podataka (u ovom slučaju DAO uzor).

Projektovanje sistemskog sloja softvera

Kako ova aplikacija treba da bude univerzalna i da se izvršava na što većem broju platformi (operativnih sistema i hardvera), ovu aplikaciju je trebalo uraditi u nekom programskom jeziku koji bi bio što univerzalniji. Izbor je pao na Java programski jezik koji omogućava da se njegovi programi prenose bez problema na skoro sve raspoložive operativne sisteme i platforme, što uključuje i mobilne uređaje.

Izbor sistema za upravljanje bazom podataka

Pošto će za pristup bazi podataka biti korišćen DAO Fabrika uzor, koji implementira pristup bazi podataka preko više slojeva pristupa i gde je moguće napraviti pristup bilo kojoj vrsti podataka (čak i XML datotekama ili običnim datotekama), kao i bilo kojem sistemu za upravljanje bazom podataka to uopšte nije važno koji će se konkretan sistem koristiti. Za ovu aplikaciju je uzeta za test MySQL baza podataka jer predstavlja jedan efikasan sistem koji je istovremeno i besplatan za korišćenje. U februaru 2002. je Ziff Davis Media Inc. izvršio testiranje nekoliko baza podataka u odnosu na brzinu izvršavanja u tu je MySQL izašao kao pobednik. Testirane baze su bile: IBM DB2 7.2, Microsoft SQL Server 2000, MySQL-Max 4.0.1, Oracle 9i 9.0.1.1.1 i Sybase ASE 12.5.0.1., a rezultat se može videti na slici 50.



Slika 50: Rezultat testiranja brzine baza podataka (izvor: Ziff Davis Media Inc.)

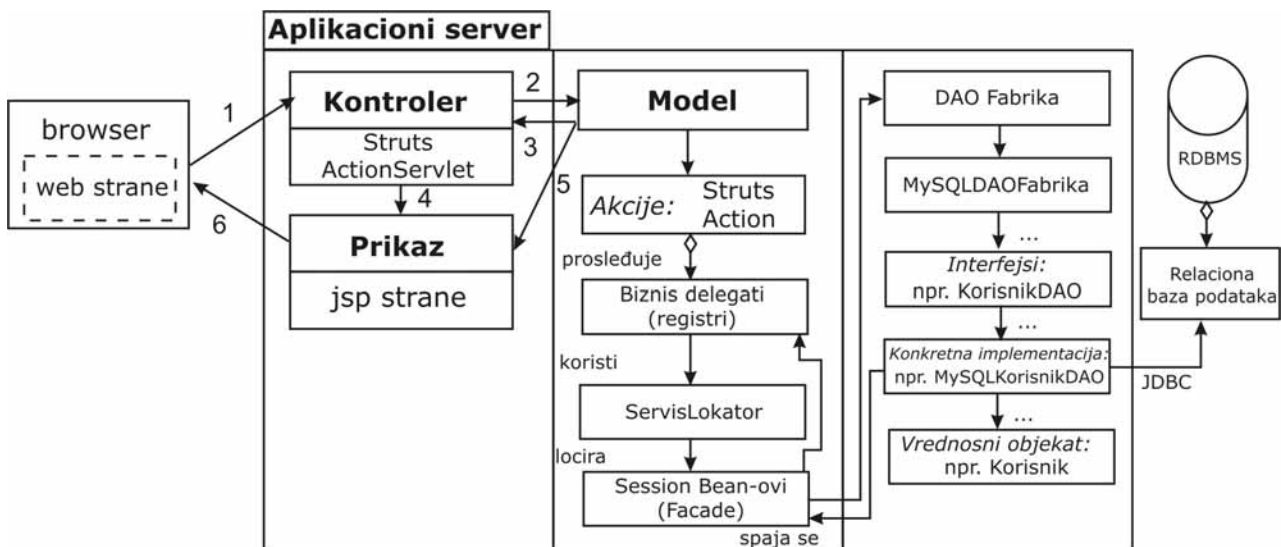
Projektovanje srednjeg sloja

Kod projektovanja srednjeg sloja trebalo bi da se koriste:

- Čuvanje objekata i obrada transakcija preko Javinih klasa koje se nalaze u paketu java.sql.
- Komunikacija klijenata i servera će biti ostvarena mrežom (tcp/ip) protokolom koristeći web čitače.
- Korisnički interfejs će biti urađen pomoću jsp strana.
- Greške u aplikaciji će biti obrađene preko Javinih standardnog mehanizma izuzetaka u Javinom paketu java.lang.

Projektovanje aplikacionog sloja

Aplikacioni sloj je u skladu sa osnovnim principima J2EE tehnologije (uzora). Struktura celog aplikacionog servera je prikazana na slici 51.



Slika 51: Struktura unutar aplikacionog servera

Strukturu aplikacionog servera čine sledeći objekti:

1. Kontroler: u ovom slučaju, pošto će biti korišćen okvir za razvoj web aplikacija Struts, to će biti Struts-ov ActionServlet, koji na osnovu zahteva koji stigne od korisnika (iz browser-a), prosleđuje taj zahtev na osnovu mapiranja zahtev-akcija, ka akciji koja odgovara tom zahtevu.

2. Prikaz: prikaz predstavljaju web stranice (jsp) koje se pozivaju i pronalaze od strane kontrolera. Za njihovo popunjavanje i prosleđivanje podataka ka njima je zadužen model.

3. Model: uključuje akcije koje su potrebne i definisane za izvršavanje pojedinih operacija. Akcije u sebi uključuju Biznis delegate koji implementiraju viši sloj apstrakcije u odnosu na Session Bean-ove. Ova tri objekta čine Model-View-Controller uzor, koji čini osnovu Modela 2.

4. Biznis delegati (registri): oni enkapsuliraju u sebi reference ka Session Bean-ovima koje kreiraju i pronalaze pomoću ServisLokator-a. Korišćenjem ovih delegata je od krajnjeg korisnika sakriven deo pronalaženja, stvaranja i inicijalizacije Session Bean-ova.

5. Session Bean-ovi: implementiraju pojedine operacije i koriste uzor DAO Fabrika za pristup i izvršavanje operacija nad bazom podataka. Oni koriste interfejsa za pojedine tipove objekata kojima se pristupa u bazi. Zajedno Biznis delegati i Session Bean-ovi čine srednji (biznis) sloj aplikacije koji čuva biznis logiku. Pomoću njih je omogućeno jasno odvajanje prezentacionog sloja aplikacije od sloja za rad sa podacima.

6. Servis Lokator: se koristi za apstrakovanje svih JNDI poziva i sakrivanje kompleksnosti kreiranja inicijalnog konteksta, traženja objekta EJB home, i ponovne kreacije EJB objekata. Više klijenata mogu koristiti ovaj objekat da smanje kompleksnost kôda, koji pruža jednu tačku kontrole i poboljšava performanse pružajući mogućnost "keširanja".

7. DAO Fabrika: se koristi kao osnovna fabrika za sve pojedinačne fabrike konkretnih baza podataka. Ona u sebi sadrži sve apstraktne interfejsa ka pojedinim DAO objektima za pristup bazi. Ona implementira uzor Data Access Object. Suština ovog uzora je da omogućava jednostavan način za rad sa bazom podataka, kao i jednostavnu promenu baze podataka.

Standardi

Kod projektovanja ovog informacionog sistema se koristi standardni UML jezik, koji predstavlja standard za specifikaciju i dokumentovanje svih elemenata ovog sistema.

2.3.3. Struktura softverskog sistema

Relacioni model

Relacioni model će se povezati direktno sa određenim sistemom za upravljanje sa bazom podataka. Pristup bazi podataka u ovoj aplikaciji je izveden preko već pomenutog Data Access Object (DAO) uzora. Kod ove aplikacije nisu korišćeni Entity Bean-ovi, zbog problema sa njihovim korišćenjem i njihove nefleksibilnosti koja bi trebala biti ispravljena u novoj specifikaciji 3.0. Korišćenje ovog uzora omogućava apstrakovanje i enkapsulaciju svih pristupa podacima. DAO je odgovoran za konekciju ka izvoru podataka, tj. za upis i čitanje podataka. Izvor podataka može biti perzistentat, kao što je RDMBS, ili eksterni servis (npr. B2B razmena), soket ili čak bilo koja datoteka. Biznis komponente se oslanjaju na DAO koristeći jednostavan interfejs koji DAO izlaže za svoje

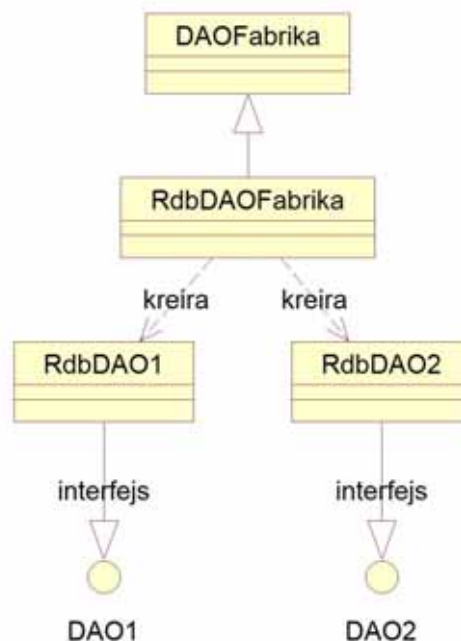
klijente. DAO kompletno sakriva implementaciju izvora podataka od svojih klijenata. Zato što je interfejs izložen klijenti ne moraju da se menjaju ako se promeni izvor podataka ili njegova implementacija, ovaj uzor omogućava DAO-u da prilagodi različite šeme smeštanja podataka bez uticaja na klijente ili biznis komponente. On se ponaša kao adapter između komponenata i izvora podataka. Na slici 52. je prikazam dijagram koji reprezentuje relacije kod DAO uzora.



Slika 52: Data Access Object uzor (pattern)

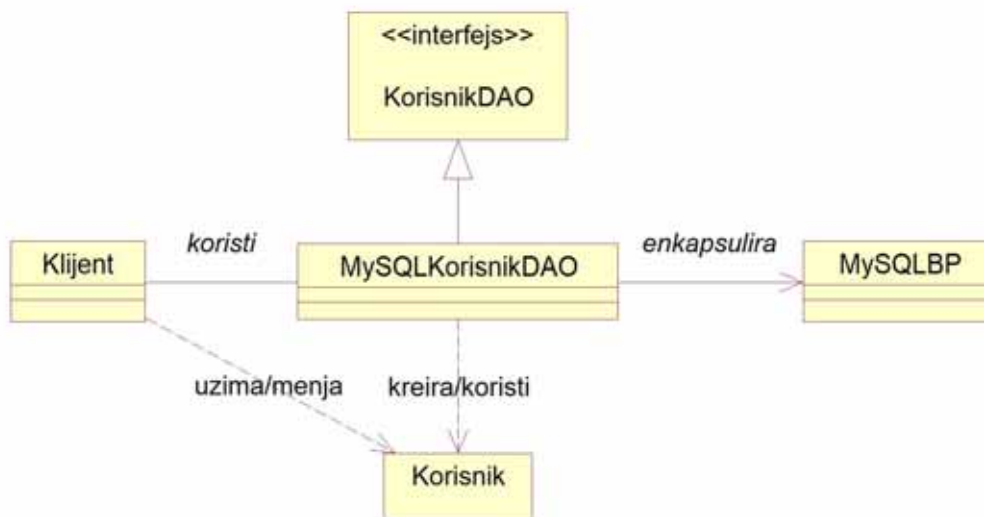
Biznis objekat reprezentuje klijenta. Ovaj objekat traži podatke i šalje podataka u bazu. Biznis objekat je implementiran kao Session Bean. Data Access Object je glavni objekat u ovom uzoru. On apstrakuje pristup podacima ispod sebe i Biznis objektima omogućava transparentan pristup podacima. Izvor podataka može biti baza podataka, XML datoteka ili neka druga datoteka, a u ovom slučaju će to biti MySQL baza podataka. ObjekatZaTransferPodataka predstavlja nosioca podataka. Njega koristi DAO za prenos podataka do klijenta. On može uključivati veći broj podataka zbog prenosa mrežom, pogotovo ako je klijentska aplikacija na računaru korisnika (što nije u ovom slučaju), i on se realizuje TransferObject uzorom.

DAO uzor se može napraviti izuzetno fleksibilnim implementirajući Apstraktna Fabrika [GoF] i Factory Metod [GoF] uzore. Kako sama baza podataka nije subjekat promene sa jedne implementacije na drugu, ova stragija može biti implementirana koristeći Factory Metod uzor, koji proizvodi onaj broj DAO-a koji zahteva aplikacija. Taj dijagram je prikazan na slici 53.



Slika 53: Fabrika za DAO strategiju koristeći Factory Metod

U ovom slučaju ova strategija pruža apstraktni objekat DAO fabrike koji može konstruisati više tipova konkretnih DAO fabrika, gde svaka fabrika implementira različitu vrstu implementacije perzistentnosti. Ovde će biti prikazan način korišćenja DAO fabrika u ovoj aplikaciji na primeru klase Korisnik, koja čuva informacije o korisnicima.



Slika 54: Implementacija DAO uzora na primeru Korisnik-a

Kako je u ovoj aplikaciji primenjen uzor Apstraktna Fabrika, moguće je napraviti pristup kao više sistema za upravljanje bazom podataka. U slučaju da postoje tri različite baze podataka, pristup kod aplikacije će izgledati kao na slici 55 (napomena, dat je primer samo za klasu Korisnik, zbog smanjenja prostora).



Slika 55: Implementiranje Fabrike za DAO strategiju koristeći uzor Apstraktna Fabrika.

Kôd za apstraktnu DAO fabriku je prikazan ispod, s tim da je on skraćen za jedan broj DAO klasa radi smanjenja prostora:

```

/*
 * DAOFabrika.java
 *
 * Created on June 7, 2005, 1:25 PM
 *
 * Implementira J2EE Core patern DAOFactory
 */
package org.archidoc.DAO;

/**
 * Apstraktna klasa za kreiranje objekata
 * @author Milan Milanovic
 * @version 1.0
 */
public abstract class DAOFabrika {

```

```
/**
 * DAO tip koji je podrzan od strane fabrike
 */
public static final int MYSQL = 1;
public static final int POSTGRESQL = 2;
public static final int MSSQL = 3;
public static final int ORACLE = 4;

/**
 * Creates a new instance of DAOFabrika
 */
public DAOFabrika() {
}

/**
 * Uzima DAO klasu korisnika
 * @return DAO klasa korisnika
 */
public abstract KorisnikDAO getKorisnikDAO();

/**
 * Uzima DAO klasu grupa
 * @return DAO klasa grupe
 */
public abstract GrupaDAO getGrupaDAO();

/**
 * Uzima DAO klasu statusa korisnika
 * @return DAO klasa statusa korisnika
 */
public abstract StatusKorisnikaDAO getStatusKorisnikaDAO();

/**
 * Uzima DAO klasu lokacije
 * @return DAO klasa lokacije objekta
 */
public abstract LokacijaDAO getLokacijaDAO();

/**
 * Uzima DAO klasu vrste objekta
 * @return DAO klasa vrste objekta
 */
public abstract VrstaObjektaDAO getVrstaObjektaDAO();

/**
 * Uzima DAO klasu tipa firme
 * @return DAO klasa tipa firme
 */
public abstract TipFirmeDAO getTipFirmeDAO();

/**
 * Uzima DAO klasu firme
 * @return DAO klasa firme
 */
public abstract FirmaDAO getFirmaDAO();

/**
 * Uzima DAO klasu tipa objekta
 * @return DAO klasa tipa objekta
 */
```

```

public abstract TipObjektaDAO getTipObjektaDAO();

/**
 * Uzima DAO klasu vrste dokumenata
 * @return DAO klasa vrste dokumenta
 */
public abstract VrstaDokumentaDAO getVrstaDokumentaDAO();

/**
 * Uzima DAO klasu statusa dokumenta
 * @return DAO klasa statusa dokumenta
 */
public abstract StatusDAO getStatusDAO();

/**
 * Uzima DAO klasu porekla
 * @return DAO klasa porekla dokumenta
 */
public abstract PorekloDAO getPorekloDAO();

// ... ovde dolaze i preostale metode koje vraćaju DAO objekte.

/**
 * Uzima odredjenu DAO Fabriku
 *
 * 1: MYSQL
 * @return DAO klasa izabrane fabrike
 * @param fabrika DAO fabrika
 *
 * 1: MYSQL
 */
public static DAOFabrika getDAOFabrika(int fabrika) {

    switch(fabrika) {
        case MYSQL: return new MySQLDAOFabrika();
        case POSTGRESQL: return new PostgreSQLDAOFabrika(); break;
        case MSSQL: return new MSSQLDAOFabrika(); break;
        case ORACLE: return new OracleDAOFabrika(); break;
        Podrazumevano : return null;
    }
}
}

```

Implementacija DAOFabrike za MySQL bazu podataka pored specifičnih metoda uključuje i implementaciju JDBC drajvera, URL-a ka bazi podatka i razlike u SQL sintaksi (ako postoje).

```

/*
 * MySQLDAOFabrika.java
 *
 * Created on June 7, 2005, 1:38 PM
 *
 * Konkretna DAO fabrika
 */

package org.archidoc.DAO;

import java.sql.*;

/**
 * Konkretna DAO fabrika za MySQL bazu

```

```
* @author Milan Milanovic
*/
public class MySQLDAOFabrika extends DAOFabrika{
    /**
     * Drajver za MySQL bazu
     */
    public static final String DRAJVER = "org.gjt.mm.mysql.Driver";
    /**
     * URL MySQL baze podataka
     */
    public static final String BazaURL =
"jdbc:mysql://localhost:3306/archidoc?useUnicode=true&characterEncoding=utf8";
    /**
     * Korisnicko ime za bazu podataka
     */
    public static final String KorisnickoImeZaBazu = "";
    /**
     * Sifra za bazu podataka
     */
    public static final String SifraZaBazu = "";

    /** Creates a new instance of MySQLDAOFabrika */
    public MySQLDAOFabrika() {
    }

    /**
     * Kreiraj konekciju ka bazi podataka
     *
     * Konekcija se kreira koristeći DRAJVER, URL
     * ka bazi, Korisnicko Ime i Sifra za bazu
     * @return Otvorena konekcija ka bazi
     */
    public static Connection kreirajKonekciju() {
        try {
            Class.forName(DRAJVER);
            Connection kon = DriverManager.getConnection(BazaURL,
KorisnickoImeZaBazu, SifraZaBazu);
            return kon;
        } catch (ClassNotFoundException e1) {
            System.out.println(e1.toString());
        } catch (SQLException e2) {
            System.out.println(e2.toString());
        } catch (Exception e3) {
            System.out.println(e3.toString());
        }
        return null;
    }

    /**
     * Uzmi DAO korisnika
     * @return DAO Korisnika
     */
    public KorisnikDAO getKorisnikDAO() {
        // MySQLKorisnikDAO implementira KorisnikDAO
        return new MySQLKorisnikDAO();
    }

    /**
     * Uzmi DAO grupa
     * @return DAO Grupe
     */
}
```

```

public GrupaDAO getGrupaDAO() {
    return new MySQLGrupaDAO();
}

/**
 * Uzmi DAO statusa korisnika
 * @return DAO Statusa korisnika
 */
public StatusKorisnikaDAO getStatusKorisnikaDAO() {
    return new MySQLStatusKorisnikaDAO();
}

/**
 * Uzmi DAO lokacija
 * @return DAO Lokacija
 */
public LokacijaDAO getLokacijaDAO() {
    return new MySQLLokacijaDAO();
}

/**
 * Uzmi DAO vrste objekata
 * @return DAO Vrste objekata
 */
public VrstaObjektaDAO getVrstaObjektaDAO() {
    return new MySQLVrstaObjektaDAO();
}

/**
 * Uzmi DAO tipa firme
 * @return DAO Tipa firme
 */
public TipFirmeDAO getTipFirmeDAO() {
    return new MySQLTipFirmeDAO();
}

/**
 * Uzmi DAO firme
 * @return DAO firme
 */
public FirmaDAO getFirmaDAO() {
    return new MySQLFirmaDAO();
}
...
}

```

Interfejs `KorisnikDAO` (kao i drugi DAO interfejsi) je prikazan ispod. On definiše DAO metode za perzistentni objekat `Korisnik` koje su implementirane od strane svih DAO implementacija, kao što su `MySQLKorisnikDAO`, `MSSQLKorisnikDAO`, itd. Na sličan način su implementirani i drugi interfejsi, kao što su `MySQLFirmaDAO`, `MySQLObjekatDAO`, `MySQLDatotekaDAO` i ostali koji ovde nisu prikazani zbog mesta.

```

/*
 * KorisnikDAO.java
 *
 * Created on June 7, 2005, 2:10 PM
 *
 * Interfejs koji svi KorisnikDAO-i moraju da podrže
 */

```

```
package org.archidoc.DAO.Korisnici;

import java.util.Collection;
import javax.sql.RowSet;

/**
 * DAO interfejs za korisnike
 * @author Milan Milanovic
 * @version 1.0
 */
public interface KorisnikDAO {

    /**
     * Kreira novog korisnika
     * @param korisnik Korisnik koji se kreira
     * @return Sifra novo kreiranog korisnika ili -1 ako nije kreiran
     */
    public long kreirajKorisnika(Korisnik korisnik);

    /**
     * Brise korisnika
     * @param sifra Sifra korisnika
     * @return Informacija o uspesnosti brisanja korisnika
     */
    public boolean brisiKorisnika(long sifra);

    /**
     * Pronalazi korisnika po nazivu
     * @param naziv Naziv korisnika
     * @return Pronadjeni korisnik ili null ako nije pronadjen
     */
    public Korisnik pronadjiKorisnika(String naziv);

    /**
     * Pronalazi korisnika po sifri
     * @param sifra Sifra korisnika
     * @return Pronadjeni korisnik ili null ako nije pronadjen
     */
    public Korisnik pronadjiKorisnika(long sifra);

    /**
     * Pronalazi sifru korisnika po korisnickom imenu
     * @param korisnickoIme Korisnicko ime korisnika
     * @return Jedinstvena sifra korisnika ili -1 ako nije pronadjen
     */
    public long pronadjiKorisnikaPoKorisnickomImenu(String korisnickoIme);

    /**
     * Pronalazi korisnike koji pripadaju grupi
     * @param grupa Grupa korisnika
     * @return Kolekcija korisnika ili null ako nije pronadjen ni jedan
     */
    public Collection pronadjiKorisnike(Grupa grupa);

    /**
     * Pronalazi korisnike koji imaju ovaj status
     * @param statusKorisnika Status korisnika
     * @return Kolekcija korisnika ili null ako nije pronadjen ni jedan
     */
    public Collection pronadjiKorisnike(StatusKorisnika statusKorisnika);

    /**
     * Pronalazi korisnike koji imaju ovaj status i pripadaju ovog grupi
     * @param statusKorisnika Status korisnika
     * @param grupa Grupa korisnika
     * @return Kolekcija korisnika ili null ako nije pronadjen ni jedan
     */
}
```

```

    public Collection pronadjiKorisnike(StatusKorisnika statusKorisnika, Grupa
grupa);

    /**
     * Vrsi izmenu korisnika
     * @param korisnik Korisnik sa kojim se menja postojeci
     * @return Informacija o uspesnosti izmene korisnika
     */
    public boolean izmeniKorisnika(Korisnik korisnik);
    /**
     * Vraca RowSet korisnika
     * @param sifra Sifra korisnika
     * @return RowSet korisnika ili null ako ni jedan nije pronadjen
     */
    public RowSet izaberiKorisnikovRS(long sifra); // Vraca rowset za sifru
korisnika

    /**
     * Vraca kolekciju svih korisnika
     * @return Kolekcija svih korisnika (vrednosnih objekata) ili null ako ni jedan
nije pronadjen
     */
    public Collection izaberiKorisnikovTO(); // Vraca kolekciju Transfer Objekata

    /**
     * Proverava da li je korisnik validan
     * @return Tacno ako je korisnik validan
     */
    public boolean daLiJeKorisnikValidan(String korisnickoIme, String sifra);
}

```

MySQLKorisnikDAO (kao i sve ostale DAO klase) implementira KorisnikDAO, kao što je prikazano na primeru ispod. Ovde, nije naravno prikazana implementacija svih metoda zbog smanjenja prostora. Implementacija ostalih DAO klasa je izvedena na sličan način.

```

package org.archidoc.DAO.Korisnici;
/*
 * MySQLKorisnikDAO.java
 *
 * Created on June 7, 2005, 2:13 PM
 *
 * MySQLKorisnikDAO implementira KorisnikDAO interfejs.
 * Ova klasa sadrzi sve MySQL specificne kodove i SQL upite.
 * Klijent je zasticen od poznavanja detalja implementacije
 */

import java.sql.*;
import java.util.Collection;
import javax.sql.RowSet;
import com.sun.rowset.CachedRowSetImpl; // u avgustu 2005. nova verzija
import java.util.ArrayList;
import javax.sql.rowset.CachedRowSet;

/**
 * Implementacija DAO interfejsa korisnika za MySQL bazu podataka
 * @author Milan Milanovic
 * @version 1.0
 */
public class MySQLKorisnikDAO implements KorisnikDAO {

```

```

Connection konekcija = MySQLDAOFabrika.kreirajKonekciju();

/** Creates a new instance of MySQLKorisnikDAO */
public MySQLKorisnikDAO() {
    // inicijalizacija
    try {
        konekcija.setAutoCommit(false);
    } catch( SQLException se ) { se.printStackTrace( ); }
}

/**
 * Kreira novog korisnika
 * @param korisnik Korisnik koji se kreira
 * @return Sifra novo kreiranog korisnika ili -1 ako nije kreiran
 */
public long kreirajKorisnika(Korisnik korisnik)
{
    // Implementiraj unosenje korisnika
    // vraca se novo kreirani broj korisnika ili -1 za gresku
    try {
        // Prvo treba pronaci najveću sifru i sabrati je sa 1
        Statement statement = konekcija.createStatement();
        ResultSet rezultat = statement.executeQuery ("SELECT max(ID) FROM
Korisnik");
        long id = 1;
        if(rezultat.next()) { id = rezultat.getLong(1); id++; }
        // nova sifra grupe
        statement.close();

        PreparedStatement stmt = konekcija.prepareStatement("INSERT INTO
Korisnik " + "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)");

        // Popuni statement iz VO objekta
        stmt.setLong(1, id);
        stmt.setString(2, korisnik.getKorisnickoIme());
        stmt.setString(3, korisnik.getSifra());
        stmt.setString(4, korisnik.getIme());
        stmt.setString(5, korisnik.getPrezime());
        stmt.setString(6, korisnik.getTelefon());
        stmt.setString(7, korisnik.getEmail());
        // konverzija java.util.date u java.sql.date
        stmt.setDate(8, new
java.sql.Date(korisnik.getDatumKreiranja().getTime()));
        stmt.setLong(9, korisnik.getGrupa());
        stmt.setLong(10, korisnik.getStatusKorisnika());

        stmt.execute(); // Izvrši upit
        konekcija.commit(); // Komit!

        stmt.close();
        return id;
    } catch( Exception e ) {
        e.printStackTrace( );
        try { konekcija.rollback(); } // Ako se desio izuzetak Rollback!
        catch( SQLException se ) { se.printStackTrace( ); }
    } finally {
        if( konekcija != null ) {
            try { konekcija.close( ); } catch( Exception e ) { e.printStackTrace( ); }
        }
    }
}

```



```

        return -1L;
    }

    /**
     * Brise korisnika
     * @param sifra Sifra korisnika
     * @return Informacija o uspesnosti brisanja korisnika
     */
    public boolean brisiKorisnika(long sifra)
    {
        // Implementacija brisanja korisnika ovde
        try {
            PreparedStatement stmt = konekcija.prepareStatement("DELETE FROM
Korisnik WHERE ID = ?");
            stmt.setLong(1, sifra);

            stmt.execute(); // Izvrsi upit
            konekcija.commit(); // Komit!

            boolean vrati = ( stmt.getUpdateCount() > 0 ) ? true : false;

            stmt.close();

            return vrati;
        } catch( Exception e ) {
            e.printStackTrace( );
            try { konekcija.rollback(); } // Ako se desio izuzetak Rollback!
            catch( SQLException se ) { se.printStackTrace( ); }
        } finally {
            if( konekcija != null ) {
                try { konekcija.close( ); } catch( Exception e ) {
e.printStackTrace( ); }
            }
        }
        return false;
    }
    ...
}

```

Korisnikov objekat za transfer ili kako se još zove vrednosni objekat (Value object) je prikazan ispod. On se koristi od strane DAO-a za slanje i primanje podataka od klijenata i ka njima. Ovde je prikazan Korisnik objekat, a na sličan način su realizovani i ostali vrednosni objekti.

```

/*
 * Korisnik.java
 *
 * Created on June 7, 2005, 2:19 PM
 *
 */

package org.archidoc.DAO.Korisnici;

import java.util.Date;

/**
 * Klasa za korisnike
 * @author Milan Milanovic
 * @version 1.0
 */

```

```
public class Korisnik extends Object implements java.io.Serializable {

    private long ID;
    private String KorisnickoIme;
    private String Sifra;
    private String Ime;
    ...

    /** Creates a new instance of Korisnik */
    public Korisnik() {
        setID(0);
        setKorisnickoIme("");
        setSifra("");
        setIme("");
        setPrezime("");
        ...
    }

    /**
     * Vraca sifru korisnika
     * @return Sifra korisnika
     */
    public long getID() {
        return ID;
    }

    /**
     * Postavlja sifru korisnika
     * @param ID Sifra korisnika
     */
    public void setID(long ID) {
        this.ID = ID;
    }

    /**
     * Vraca korisnicko ime
     * @return Korisnicko ime
     */
    public String getKorisnickoIme() {
        return KorisnickoIme;
    }

    /**
     * Postavlja korisnicko ime
     * @param KorisnickoIme Korisnicko ime
     */
    public void setKorisnickoIme(String KorisnickoIme) {
        this.KorisnickoIme = KorisnickoIme;
    }

    /**
     * Vraca sifru korisnika
     * @return Sifra korisnika
     */
    public String getSifra() {
        return Sifra;
    }
    ... // ovde ide još kôda.
}
```

Primer korišćenja ovih klasa je prikazan ovde.

```
// Kreiranje nove fabrike
DAOFabrika MySQLFabrika = DAOFabrika.getDAOFabrika(DAOFabrika.MYSQL);
KorisnikDAO korisnikDAO = MySQLFabrika.getKorisnikDAO();

// Kreiranje novog korisnika
int noviKorisnikBroj = korisnikDAO.kreirajKorisnika();

// Pronadji korisnika. Vрати njegov Transfer Object
Korisnik korisnik = korisnikDAO.pronadjiKorisnika();

// Promeni vrednosti u transfer objektu
korisnik.setEmail();

// Izmeni korisnika koristeći DAO
korisnikDAO.izmeniKorisnika(korisnik);

// Brisi korisnika
korisnikDAO.brisiKorisnika();
```

Projektovanje skladišta podataka

Na osnovu softverskih klasa strukture projektovane su tabele relacionog sistema za upravljanje bazom podataka.

Tabela: Akcije

Kolone:

akcije					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Akcija		varchar(255)			
Opis	Y	varchar(255)			
RoditeljAkcija	Y	int(10)			
AkcijaIkonica	Y	varchar(255)			
AkcijaPutanja	Y	varchar(255)			
AkcijaUradi	Y	varchar(255)			

Indeksi:

akcije					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: Dokument

Kolone:

dokument					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Datum		date		0000-00-00	
Broj		varchar(255)			
Napomena	Y	varchar(255)			
ZamenjenSa	Y	int(10) unsigned	MUL		
PovezanSa	Y	int(10) unsigned	MUL		
VrstaDokumenta		int(10) unsigned	MUL	0	
NazivDokumenta	Y	varchar(255)			
NazivDatoteke	Y	varchar(255)			
Lokacija	Y	varchar(255)			
Autor	Y	int(10) unsigned	MUL		
Arhivar	Y	int(10) unsigned	MUL		
Revizija	Y	int(10) unsigned	MUL		
VerzijaDokumenta		int(10) unsigned	MUL	0	
Status		int(10) unsigned	MUL	0	
Poreklo		int(10) unsigned	MUL	0	

Indeksi:

dokument					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
ZamenjenSa	BTREE	True	zam_ind	YES	
PovezanSa	BTREE	True	pov_ind	YES	
VrstaDokumenta	BTREE	True	vrs_ind		
Autor	BTREE	True	aut_ind	YES	
Arhivar	BTREE	True	arh_ind	YES	
Revizija	BTREE	True	rev_ind	YES	
VerzijaDokumenta	BTREE	True	ver_ind		
Status	BTREE	True	sta_ind		
Poreklo	BTREE	True	por_ind		

Tabela: DozvoljeneAkcije

Kolone:

dozvoljeneakcije					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
Grupa		int(10) unsigned	PRI	0	
Akcija		int(10) unsigned	PRI	0	

Indeksi:

dozvoljeneakcije					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
Grupa	BTREE	False	PRIMARY		
Akcija	BTREE	False	PRIMARY		
Grupa	BTREE	True	gru_ind		
Akcija	BTREE	True	akc_ind		

Tabela: Email

Kolone:

email					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
TekstPoruke	Y	varchar(255)			
Autor	Y	varchar(30)			
Naslov	Y	varchar(255)			
DatumSlanja	Y	varchar(20)			
AdresaAutora	Y	varchar(255)			
KorisnickoIme	Y	varchar(30)			
Folder	Y	varchar(30)			

Indeksi:

email					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: Firma

Kolone:

firma					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra

ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			
Adresa	Y	varchar(255)			
KontaktOsoba	Y	varchar(255)			
Telefon	Y	varchar(255)			
Mesto	Y	varchar(255)			
Email	Y	varchar(255)			
Mobilni	Y	varchar(255)			
Fax	Y	varchar(255)			
ZiroRacun	Y	varchar(255)			
PostanskiBroj	Y	varchar(255)			
Tip		int(10) unsigned	MUL	0	

Indeksi:

firma					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
Tip	BTREE	True	tip_ind		

Tabela: Grupa

Kolone:

grupa					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			
Opis	Y	varchar(255)			

Indeksi:

grupa					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: GrupaDokumenata

Kolone:

grupadokumenata					
-----------------	--	--	--	--	--

Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Broj		varchar(255)			
Naziv		varchar(255)			
Datum	Y	date			
BrojUgovora	Y	varchar(255)			
Napomena	Y	varchar(255)			
Revizija	Y	int(10) unsigned	MUL		
Arhivar		int(10) unsigned	MUL	0	
Investitor	Y	int(10) unsigned	MUL		
Vrsta		int(10) unsigned	MUL	0	
Lokacija	Y	varchar(255)			

Indeksi:

grupadokumenata					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
Revizija	BTREE	True	rev_ind	YES	
Arhivar	BTREE	True	arh_ind		
Investitor	BTREE	True	inv_ind	YES	
Vrsta	BTREE	True	vrs_ind		

Tabela: IstorijaDogadjaja**Kolone:**

istorijadogadjaja					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Korisnik		int(10) unsigned	MUL	0	
Dokument	Y	int(10) unsigned	MUL		
Datum	Y	date			
Dogadjaj	Y	varchar(255)			

Indeksi:

istorijadogadjaja					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Korisnik	BTREE	True	kor_ind		
Dokument	BTREE	True	dok_ind	YES	

Tabela: Korisnik

Kolone:

korisnik					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
KorisnickoIme		varchar(255)			
Sifra		varchar(255)			
Ime	Y	varchar(255)			
Prezime	Y	varchar(255)			
Telefon	Y	varchar(255)			
Email	Y	varchar(255)			
DatumKreiranja		date		0000-00-00	
Grupa		int(10) unsigned	MUL	0	
Status		int(10) unsigned	MUL	0	

Indeksi:

korisnik					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
Grupa	BTREE	True	gru_ind		
Status	BTREE	True	sta_ind		

Tabela: Lokacija

Kolone:

lokacija					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Mesto	Y	varchar(255)			
Drzava	Y	varchar(255)			
Region	Y	varchar(255)			

Indeksi:

lokacija					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: **Objekat**

Kolone:

objekat					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			
Broj		varchar(255)			
Datum		date		0000-00-00	
Napomena	Y	varchar(255)			
Lokacija	Y	int(10) unsigned	MUL		
Vrsta		int(10) unsigned	MUL	0	
TipObjekta	Y	int(10) unsigned			
Firma	Y	int(10) unsigned	MUL		
Arhivar		int(10) unsigned	MUL	0	

Indeksi:

objekat					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
Lokacija	BTREE	True	lok_ind	YES	
Vrsta	BTREE	True	vrs_ind		
Firma	BTREE	True	fir_ind	YES	
Arhivar	BTREE	True	arh_ind		

Tabela: **Poreklo**

Kolone:

poreklo					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			

Indeksi:

poreklo					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: PrilogUzEmail

Kolone:

priloguzemail					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
DeoID		int(10) unsigned	PRI	0	
NazivDatoteke	Y	varchar(255)			
MimeTip	Y	varchar(255)			

Indeksi:

priloguzemail					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
DeoID	BTREE	False	PRIMARY		
ID	BTREE	False	PRIMARY		
ID	BTREE	True	pri_ind		

Tabela: PripadnostDokumentaGrupiDokumenata

Kolone:

pripadnostdokumentagrupidokumenata					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
GrupaDokumenata		int(10) unsigned	PRI	0	
Dokument		int(10) unsigned	PRI	0	
Opis	Y	varchar(255)			

Indeksi:

pripadnostdokumentagrupidokumenata					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
GrupaDokumenata	BTREE	False	PRIMARY		

Dokument	BTREE	False	PRIMARY		
GrupaDokumenata	BTREE	True	gdo_ind		
Dokument	BTREE	True	dok_ind		

Tabela: PripadnostDokumentaObjektu

Kolone:

pripadnostdokumentaobjektu					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
Objekat		int(10) unsigned	PRI	0	
Dokument		int(10) unsigned	PRI	0	
Opis	Y	varchar(255)			

Indeksi:

pripadnostdokumentaobjektu					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
Objekat	BTREE	False	PRIMARY		
Dokument	BTREE	False	PRIMARY		
Objekat	BTREE	True	obj_ind		
Dokument	BTREE	True	dok_ind		

Tabela: PripadnostGrupeDokumenataObjektu

Kolone:

pripadnostgrupedokumenataobjektu					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
Objekat		int(10) unsigned	PRI	0	
GrupaDokumenata		int(10) unsigned	PRI	0	
Opis	Y	varchar(255)			

Indeksi:

pripadnostgrupedokumenataobjektu					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
Objekat	BTREE	False	PRIMARY		
GrupaDokumenata	BTREE	False	PRIMARY		
Objekat	BTREE	True	obj_ind		
GrupaDokumenata	BTREE	True	gdo_ind		

Tabela: PripadnostObjektaObjektu

Kolone:

pripadnostobjektaobjektu					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ObjekatIznad		int(10) unsigned	PRI	0	
ObjekatIspod		int(10) unsigned	PRI	0	
Opis	Y	varchar(255)			

Indeksi:

pripadnostobjektaobjektu					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ObjekatIznad	BTREE	False	PRIMARY		
ObjekatIspod	BTREE	False	PRIMARY		
ObjekatIznad	BTREE	True	oiz_ind		
ObjekatIspod	BTREE	True	ois_ind		

Tabela: Revizija

Kolone:

revizija					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Broj		varchar(255)			

Indeksi:

revizija					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: SistemskaPoruka

Kolone:

sistemskaporuka					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	

Autor		int(10) unsigned	MUL	0
Primalac		int(10) unsigned	MUL	0
TekstPoruke	Y	text		
Naslov		varchar(255)		
DatumSlanja		date		0000-00-00
VaziDana	Y	int(11)		
Prioritet	Y	int(10) unsigned		
PotvrdaPrijema	Y	int(10) unsigned		
Procitana		int(10) unsigned		0

Indeksi:

sistemskaporuka					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		
Autor	BTREE	True	kor_ind		
Primalac	BTREE	True	pri_ind		

Tabela: Sistemske Varijable

Kolone:

sistemskevarijable					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv	Y	varchar(255)			
Vrednost	Y	varchar(255)			
Opis	Y	varchar(255)			

Indeksi:

sistemskevarijable					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: Status

Kolone:

status					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra

ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			

Indeksi:

status					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: StatusKorisnika

Kolone:

statuskorisnika					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Status		varchar(255)			

Indeksi:

statuskorisnika					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: TipFirme

Kolone:

tipfirme					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Tip	Y	varchar(255)			

Indeksi:

tipfirme					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: TipObjekta

Kolone:

tipobjekta					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Tip	Y	varchar(255)			

Indeksi:

tipobjekta					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: VerzijaDokumenta

Kolone:

verzjadokumenta					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Verzija		varchar(255)			
DatumVerzije		date		0000-00-00	
Napomena	Y	varchar(255)			

Indeksi:

verzjadokumenta					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: VrstaDokumenta

Kolone:

vrstadokumenta					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv	Y	varchar(255)			

Indeksi:

vrstadokumenta					
----------------	--	--	--	--	--

Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: VrstaGrupeDokumenta

Kolone:

vrstagrudedokumenata					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Naziv		varchar(255)			

Indeksi:

vrstagrudedokumenata					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Tabela: VrstaObjekta

Kolone:

vrstaobjekta					
Polja	Null	Tip	Ključ	Podrazumevano	Ekstra
ID		int(10) unsigned	PRI	0	
Vrsta	Y	varchar(255)			

Indeksi:

vrstaobjekta					
Naziv kolone	Tip indeksa	Nije jedinstven	Naziv ključa	Null	Komentar
ID	BTREE	False	PRIMARY		

Klase projektovanja strukture

Klase strukture iz analize su realizovane preko sledećih klasa projektovanja:

Akcije, Dokument, DozvoljeneAkcije, Email, Firma, Grupa, GrupaDokumenata, IstorijaDogadjaja, Korisnik, Lokacija, Objekat, Poreklo, PrilogUzEmail, PripadnostDokumentaGrupiDokumenata, PripadnostDokumentaObjektu, PripadnostGrupeDokumenataObjektu, PripadnostiObjektaObjektu, Revizija, SistemskeVarijable, SistemskaPoruka, Status, StatusKorisnika, TipFirme, TipObjekta, VerzijaDokumenta, VrstaDokumenta, VrstaGrupeDokumenata i VrstaObjekta.

Svaka od ovih klasa je izvedena iz klase Object i realizuje interfejs Serializable da im omogućila mehanizam serijalizacije radi prenosa preko mreže, a i zbog upisa u datoteke.

2.3.3. Ponašanje softverskog sistema

Sistemske operacije opisuju ponašanje sistema, odnosno predstavljaju atomske funkcije sistema, koje se mogu koristiti iz okruženja sistema. Kompletan kôd je raspoloživ na CD-u.

Konceptualna rešenja sistemskih operacija

UGOVOR UG1: UnesiDokument

Operacija: UnesiDokument(datoteka:File, InformacijeODokumentu:Dokument dokument:Dokument) : signal;

Veza sa SK: DS8

Preduslovi: Korisnik je prijavljen u sistem.

Postuslovi: Dokument je unesen u sistem.



Slika 56: Sekvencni dijagram metode UnesiDokument()

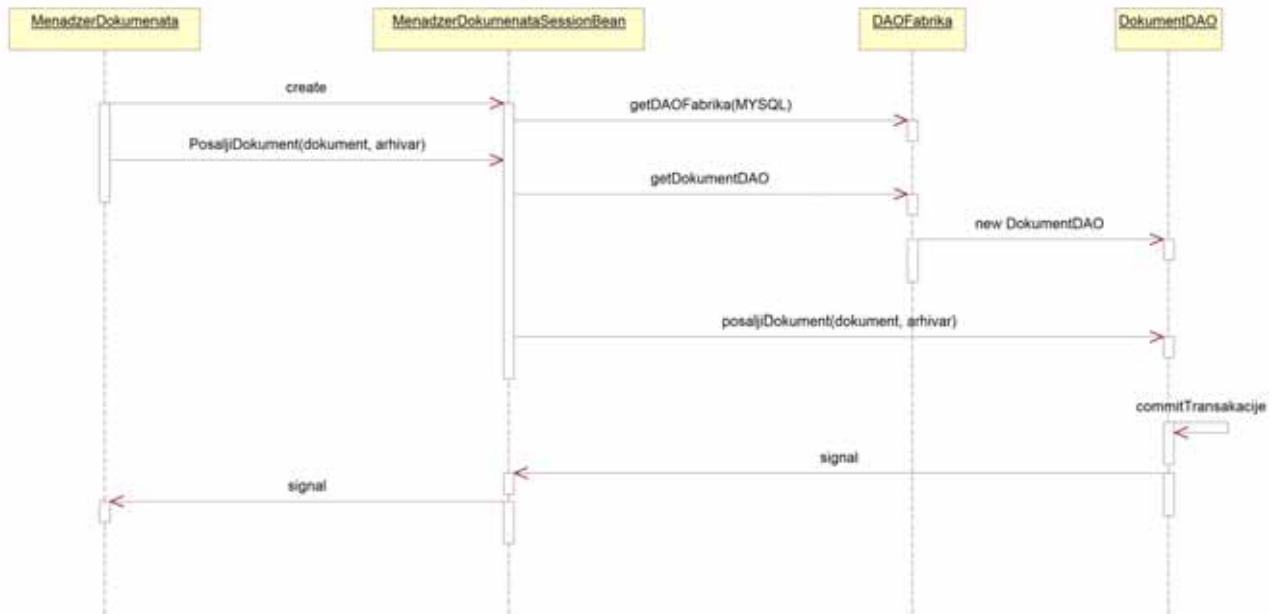
UGOVOR UG2: PosaljiDokument

Operacija: PosaljiDokument(dokument:Dokument, Arhivar:Korisnik, PotvrdaUspesnostiSlanja:String)
: signal;

Veza sa SK: DS8

Preduslovi: Dokument je unesen u sistem.

Postuslovi: Dokument je poslat u Arhivu.



Slika 57: Sekvencni dijagram metode PosaljiDokument()

UGOVOR UG3: Arhiviranje

Operacija: Arhiviranje(dokumenti []:Dokument) : signal;

Veza sa SK: DS1

Preduslovi: -

Postuslovi: Dokument je arhiviran.



Slika 58: Sekvencni dijagram metode Arhiviranje()

UGOVOR UG4: IzmenaInformacijaODokumentu

Operacija: `IzmenaInformacijaODokumentu(dokument:Dokument) : signal;`

Veza sa SK: DS1

Preduslovi: Dokument je izabran.

Postuslovi: Dokument je izmenjen.



Slika 59: Sekvenčni dijagram metode `IzmenaInformacijaODokumentu()`

UGOVOR UG5: PoveziDokumentSaObjektom

Operacija: PoveziDokumentSaObjektom(objekat:Objekat, dokument:Dokument) : signal;

Veza sa SK: DS1

Preduslovi: Objekat postoji u sistemu.

Postuslovi: Dokument je povezan sa objektom.



Slika 60: Sekvenčni dijagram metode `PoveziDokumentSaObjektom()`

UGOVOR UG6: PoveziDokumentSaGrupom

Operacija: PoveziDokumentSaGrupom(dokument:Dokument, grupaDokumenata:GrupaDokumenata)
: signal;

Veza sa SK: DS1

Preduslovi: Grupa dokumenata postoji u sistemu.

Postuslovi: Dokument je povezan sa grupom dokumenata;



Slika 61: Sekvencni dijagram metode PoveziDokumentSaGrupom()

UGOVOR UG7: PregledObjekata

Operacija: PregledObjekata(Objekat []obj) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: -



Slika 62: Sekvencni dijagram metode PregledObjekata()

UGOVOR UG8: KreirajObjekat

Operacija: KreirajObjekat(objekat:Objekat) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: Objekat je kreiran u sistemu.



Slika 63: Sekvencni dijagram metode KreirajObjekat()

UGOVOR UG9: PoveziSaObjektom

Operacija: PoveziSaObjektom(objekatVisi:Objekat, ObjekatNizi:Objekat) : signal;

Veza sa SK: DS5

Preduslovi: -

Postuslovi: Objekat je povezan sa drugim objektom, visim u hijerarhiji.



Slika 64: Sekvencni dijagram metode PoveziSaObjektom()

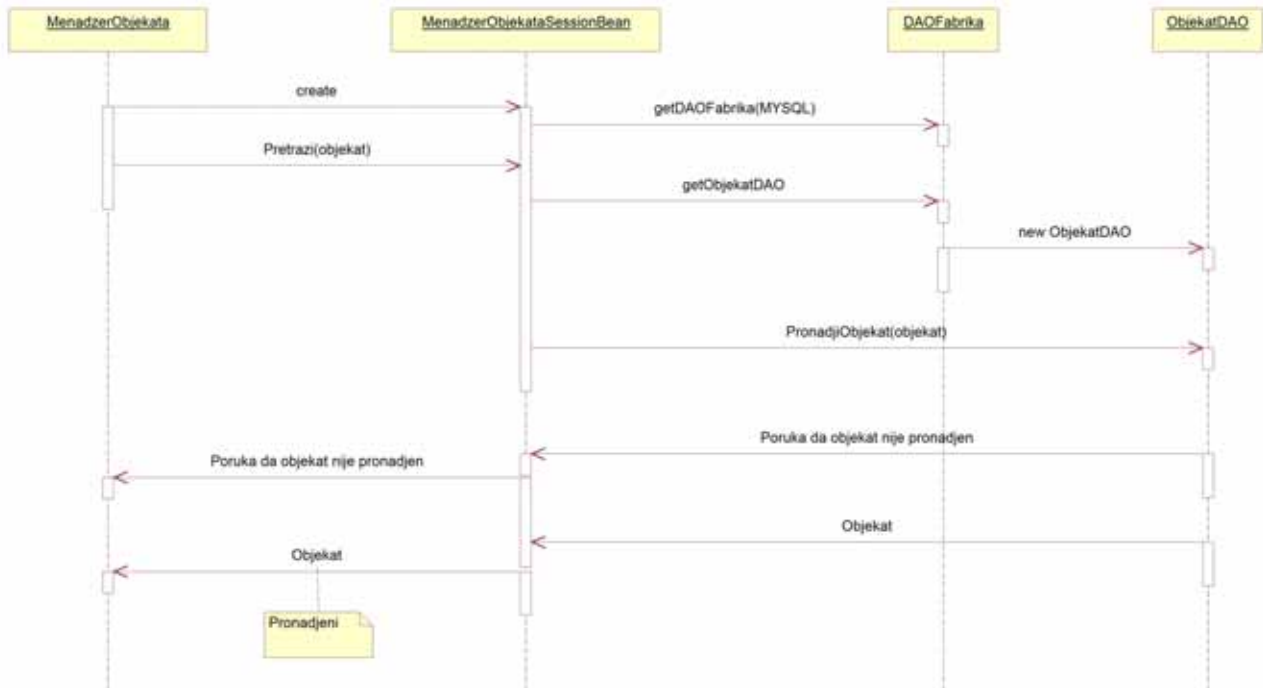
UGOVOR UG10: Pretrazi

Operacija: Pretrazi(objekat:Objekat) : signal;

Veza sa SK: DS2

Preduslovi: -

Postuslovi: Kreirana je forma za pretragu dokumenata.



Slika 65: Sekvencni dijagram metode Pretrazi()

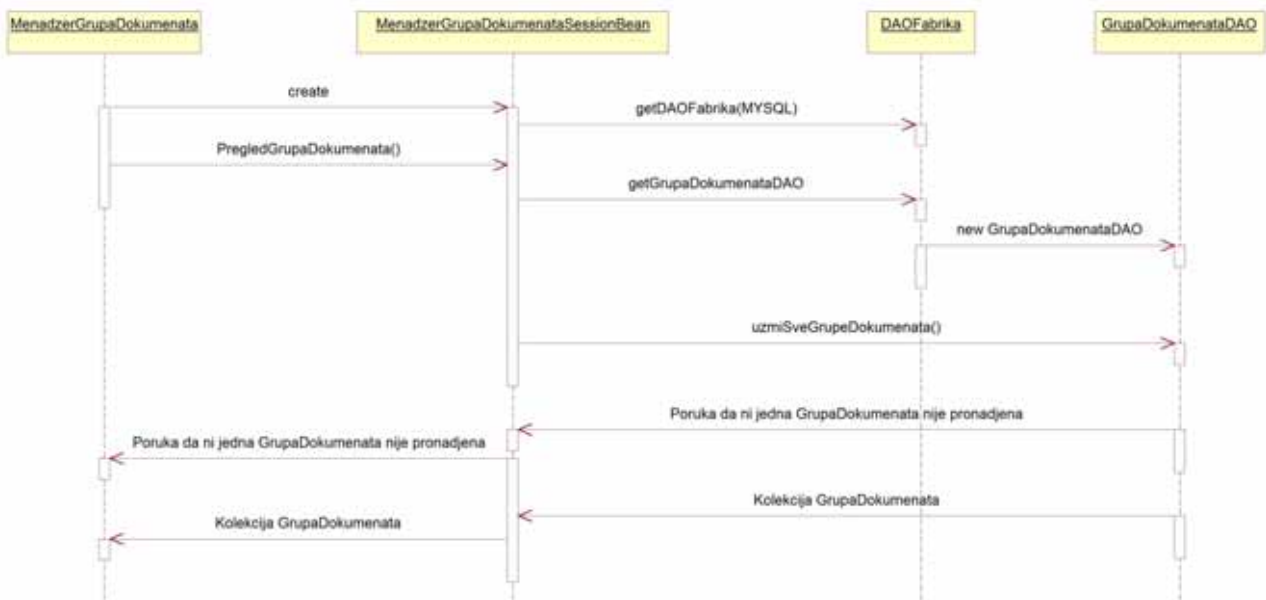
UGOVOR UG11: PregledGrupaDokumenata

Operacija: PregledGrupaDokumenata() : signal;

Veza sa SK: DS6

Preduslovi: -

Postuslovi: Vraća sve grupe dokumenata u sistemu.



Slika 66: Sekvencni dijagram metode PregledGrupaDokumenata()

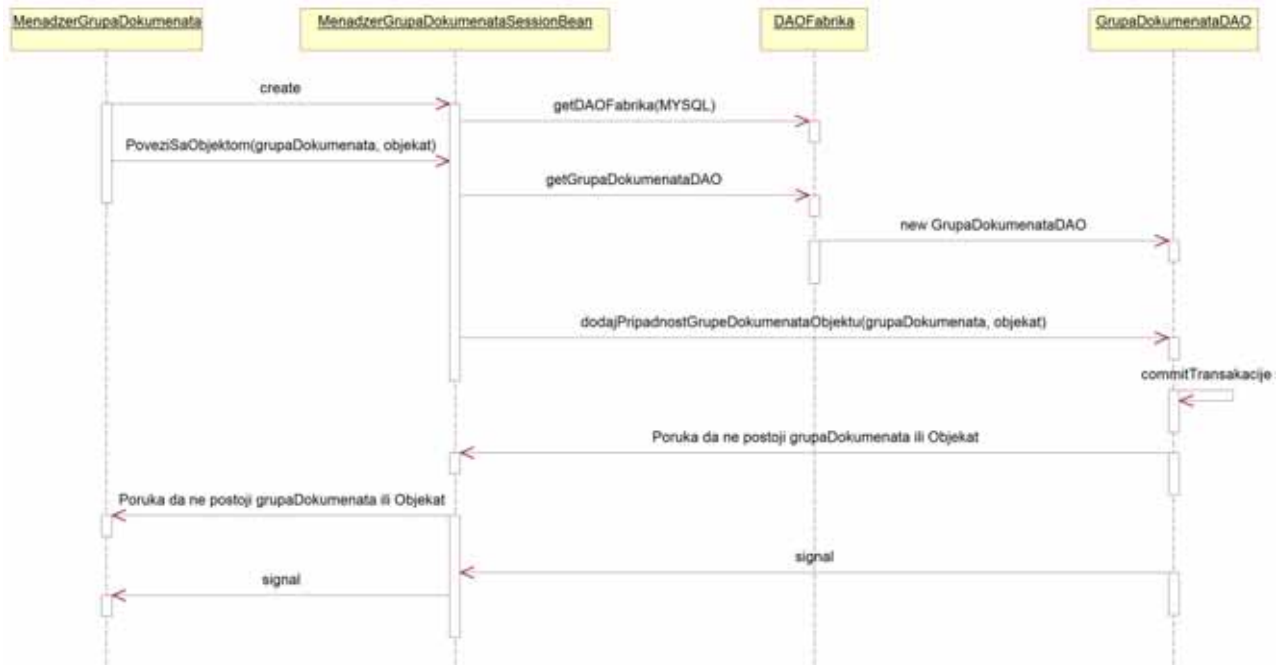
UGOVOR UG12: PoveziSaObjektom

Operacija: PoveziSaObjektom(grupaDokumenata:GrupaDokumenata, objekat:Objekat) : signal;

Veza sa SK: DS6

Preduslovi: U sistemu moraju postojati grupa dokumenata i objekat koji se povezuju.

Postuslovi: Grupa dokumenata je vezana za objekat.



Slika 67: Sekvencni dijagram metode PoveziSaObjektom()

UGOVOR UG13: KreirajGrupuDokumenata

Operacija: KreirajGrupuDokumenata(grupaDokumenata:grupaDokumenata) : signal;

Veza sa SK: DS6

Preduslovi: Prikazane su sve grupe dokumenata.

Postuslovi: Kreira novu grupu dokumenata.



Slika 68: Sekvencni dijagram metode KreirajGrupuDokumenata()

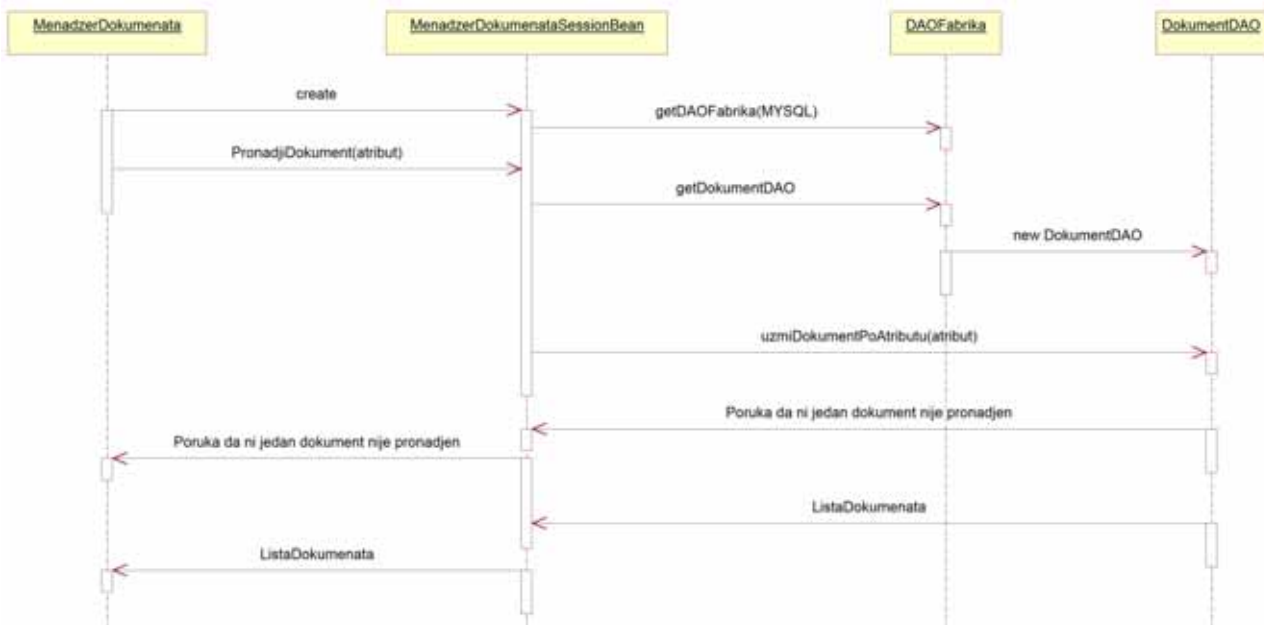
UGOVOR UG14: PronadjiDokument

Operacija: PronadjiDokument(atribut:String, Dokument [] dokumenti) : signal;

Veza sa SK: DS2

Preduslovi: -

Postuslovi: Dokumenti koji su pronađeni mogu da se pogledaju.



Slika 69: Sekvencni dijagram metode PronadjiDokument()

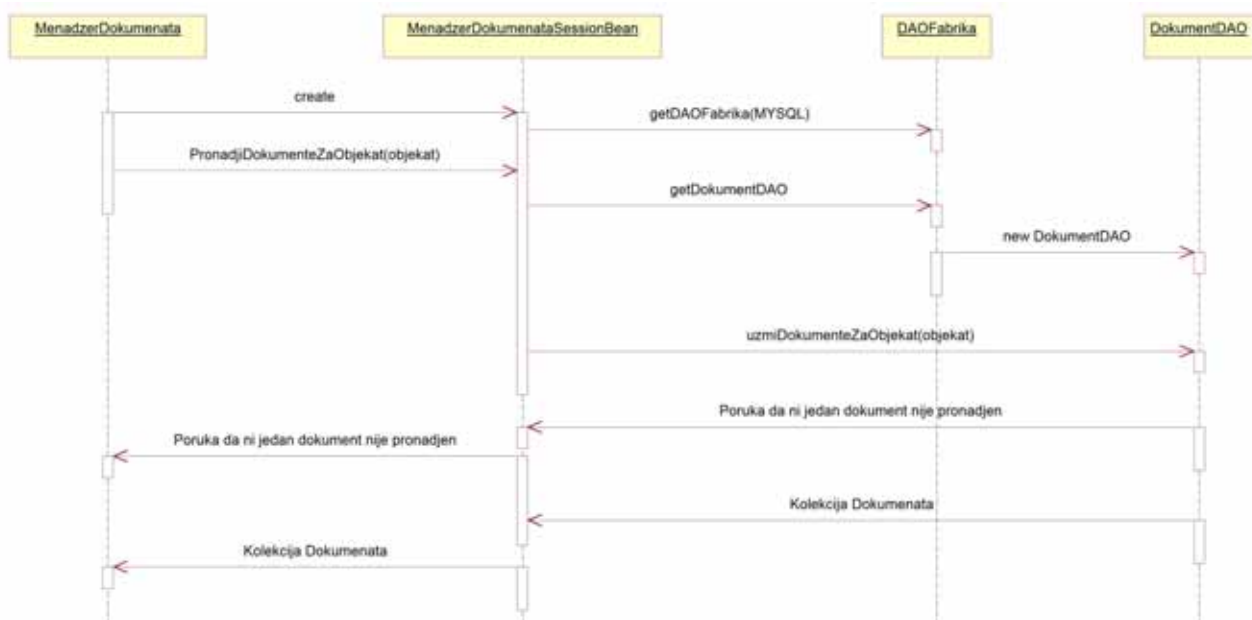
UGOVOR UG15: PronadjiDokumenteZaObjekat

Operacija: PronadjiDokumenteZaObjekat(objekat:Objekat, Dokument [] dokumenti) : signal;

Veza sa SK: DS2

Preduslovi: Objekat mora da postoji u sistemu.

Postuslovi: Dokumenti koji su pronađeni mogu da se pogledaju.



Slika 70: Sekvencni dijagram metode PronadjiDokumenteZaObjekat()

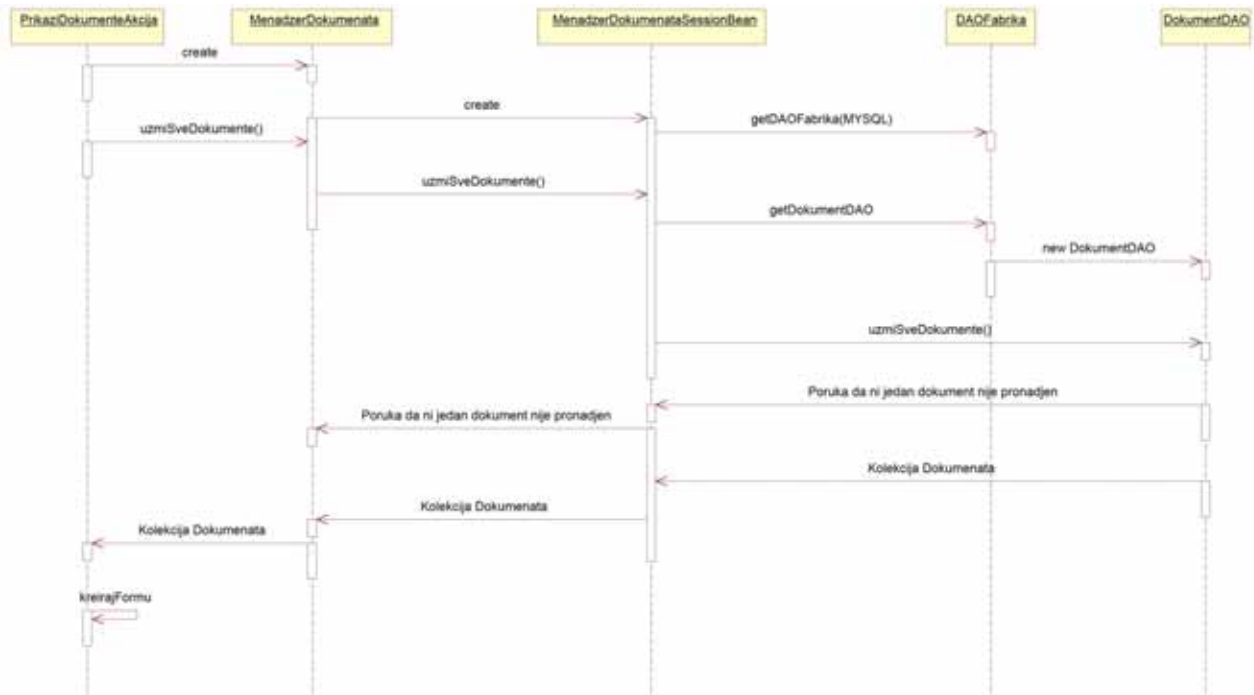
UGOVOR UG16: Pregled

Operacija: Pregled() : signal;

Veza sa SK: DS3

Preduslovi: -

Postuslovi: Kreirana je forma za pregled dokumenata.



Slika 71: Sekvencni dijagram metode PronadjiDokumenteZaObjekat()

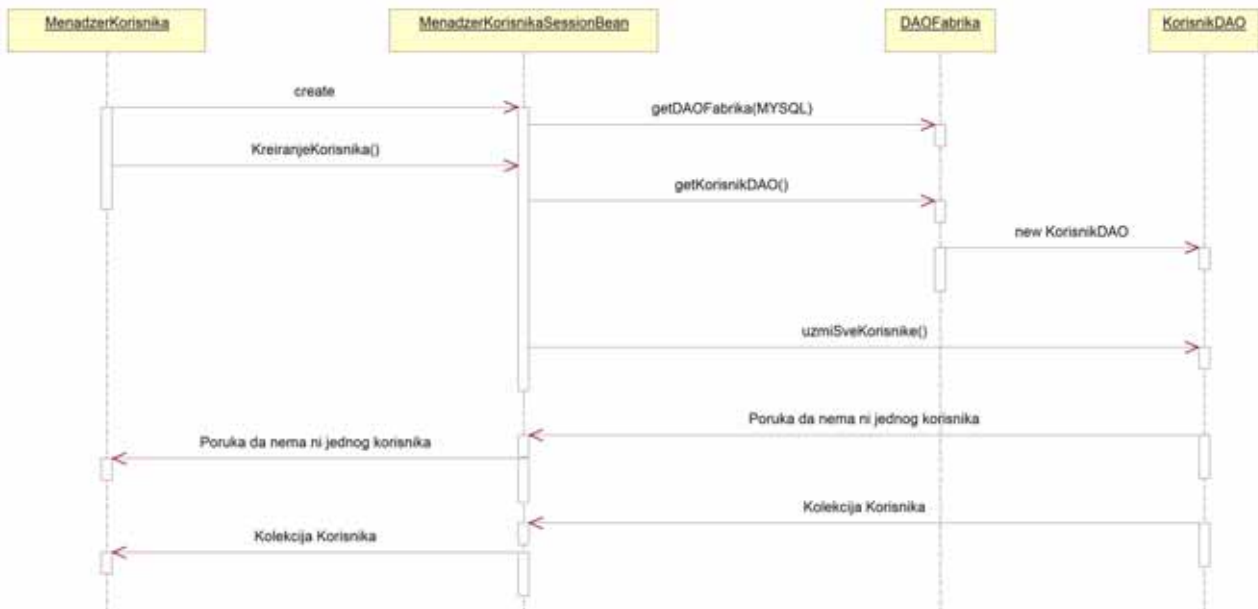
UGOVOR UG17: KreiranjeKorisnika

Operacija: KreiranjeKorisnika(Korisnik [] korisnici) : signal;

Veza sa SK: DS11

Preduslovi: Mora postojati bar jedan korisnik u sistemu (administrator).

Postuslovi: Preuzeta je lista svih korisnika iz sistema.



Slika 72: Sekvencni dijagram metode KreiranjeKorisnika()

UGOVOR UG18: NoviKorisnik

Operacija: NoviKorisnik(korisnik:Korisnik, poruka:String) : signal;

Veza sa SK: DS11

Preduslovi: -

Postuslovi: Kreiran je novi korisnik u sistemu.



Slika 73: Sekvencni dijagram metode NoviKorisnik()

UGOVOR UG19: UzmiDokument

Operacija: `UzmiDokument(id:integer, dokument:Dokument) : signal;`

Veza sa SK: DS4

Preduslovi: Dokument mora da postoji u sistemu.

Postuslovi: Dokument je preuzet iz sistema.



Slika 74: Sekvencni dijagram metode UzmiDokument()

UGOVOR UG20: ZameniDokument

Operacija: **ZameniDokument**(dokument:Dokument, poruka:String) : signal;

Veza sa SK: DS4

Preduslovi: Dokument mora da postoji u sistemu.

Postuslovi: Dokument u sistemu je zamenjen sa postojećim.



Slika 75: Sekvencni dijagram metode ZameniDokument()

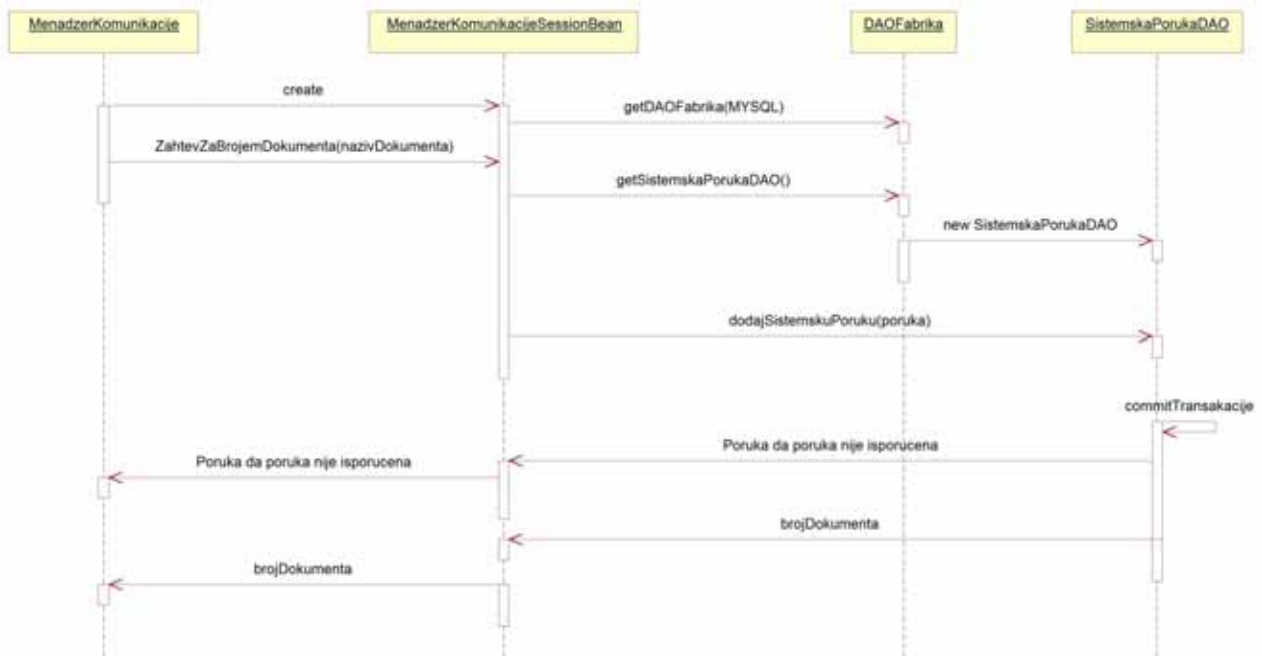
UGOVOR UG21: ZahtevZaBrojemDokumenta

Operacija: `ZahtevZaBrojemDokumenta(nazivDokumenta:String, brojDokumenta:String) : signal;`

Veza sa SK: DS7

Preduslovi: -

Postuslovi: Dobijen je novi broj dokumenta.



Slika 76: Sekvencni dijagram metode `ZahtevZaBrojemDokumenta()`

UGOVOR UG22: BrisiDokument

Operacija: **BrisiDokument**(id:integer, poruka:String) : signal;

Veza sa SK: DS9

Preduslovi: Dokument mora postojati u sistemu.

Postuslovi: -.



Slika 77: Sekvenčni dijagram metode BrisiDokument()

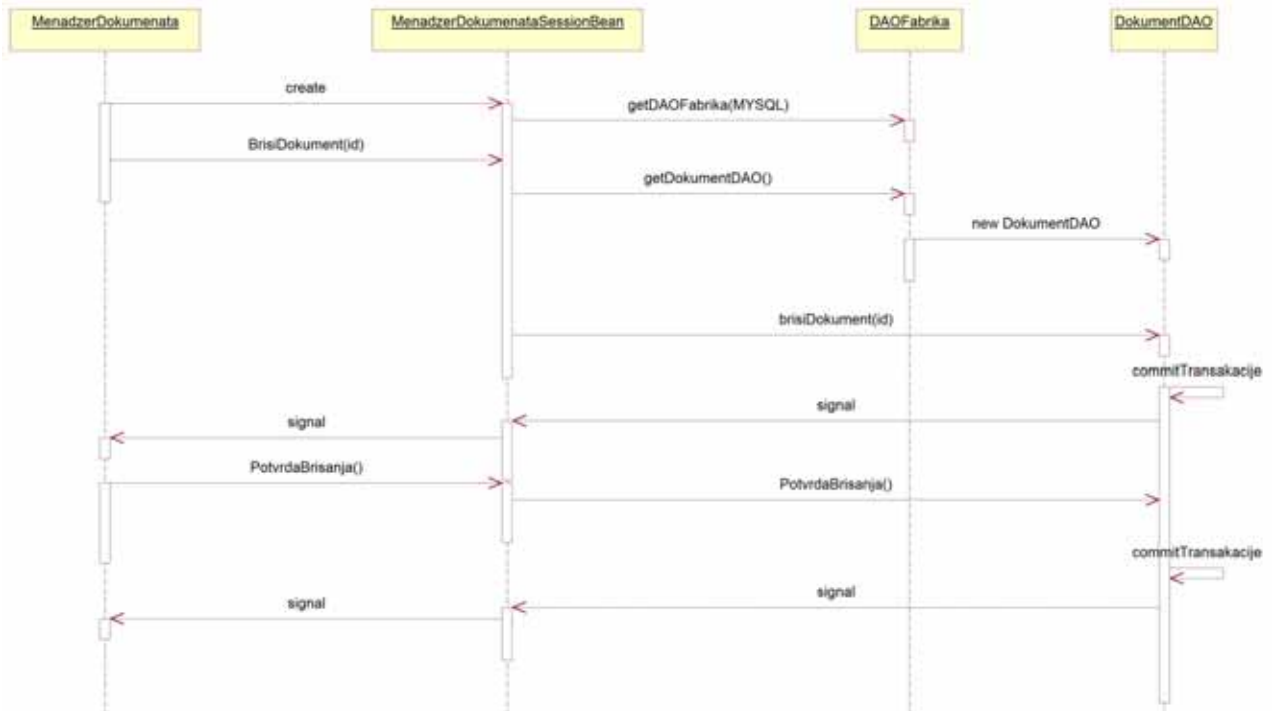
UGOVOR UG23: PotvrdaBrisanja

Operacija: PotvrdaBrisanja(poruka:String) : signal;

Veza sa SK: DS9

Preduslovi: Dokument mora biti u režimu brisanja i ne sme biti vezan za neki objekat ili grupu dokumenata.

Postuslovi: Dokument je izbrisan iz sistema.



Slika 78: Sekvencni dijagram metode PotvrdaBrisanja()

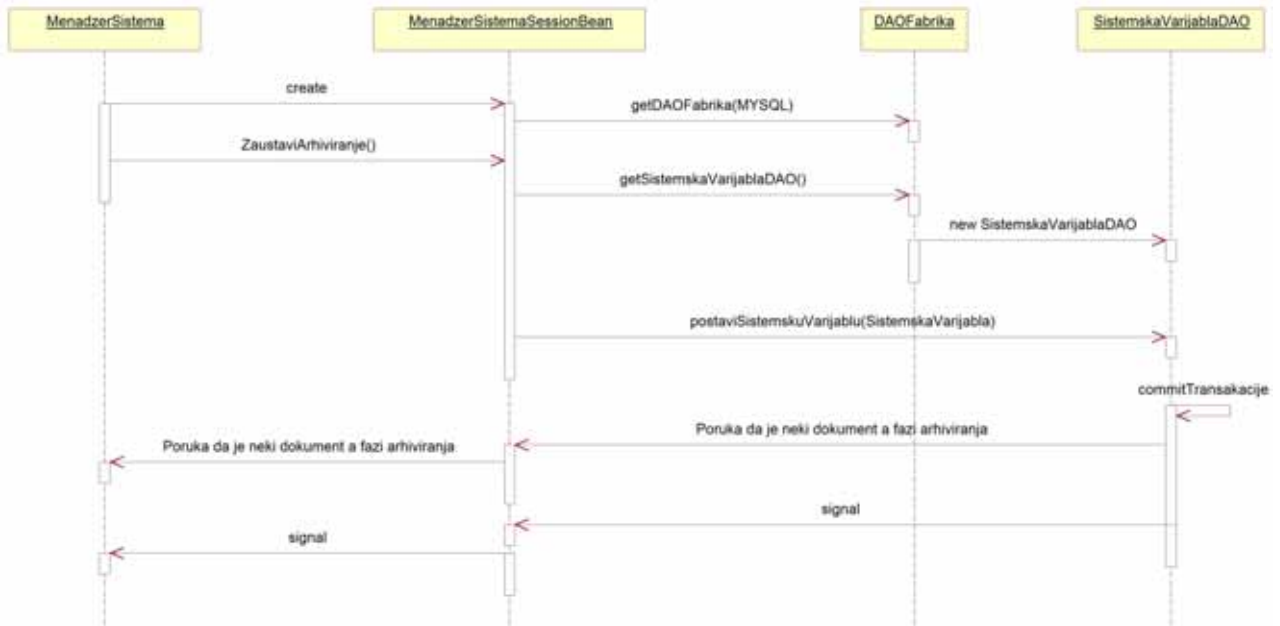
UGOVOR UG24: ZaustaviArhiviranje

Operacija: ZaustaviArhiviranje(poruka:String) : signal;

Veza sa SK: DS10

Preduslovi: Ni jedan dokument ne sme biti u režimu arhiviranja.

Postuslovi: Arhiviranje dokumenata je zaustavljeno.



Slika 79: Sekvencni dijagram metode ZaustaviArhiviranje()

UGOVOR UG25: PregledLogDatoteka**Operacija:** PregledLogDatoteka() : signal;**Veza sa SK:** DS12**Preduslovi:** Log datoteke moraju postojati.**Postuslovi:** Vraćena je forma za izbor log datoteke.

Slika 80: Sekvenčni dijagram metode PregledLogDatoteka()

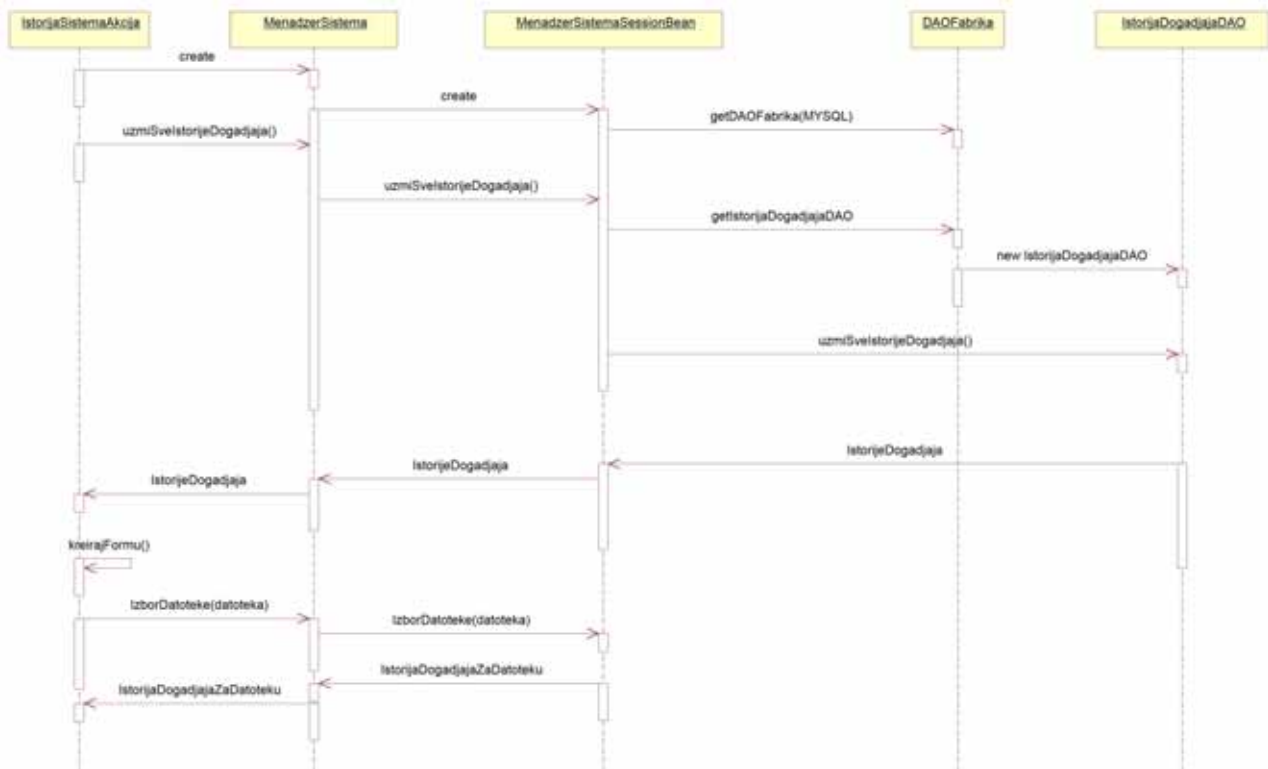
UGOVOR UG26: IzborDatoteke

Operacija: IzborDatoteke () : signal;

Veza sa SK: DS12

Preduslovi: Prikazana je forma za izbor log datoteke.

Postuslovi: Prikazan je izveštaj iz izabrane log datoteke.



Slika 81: Sekvenčni dijagram metode IzborDatoteke()

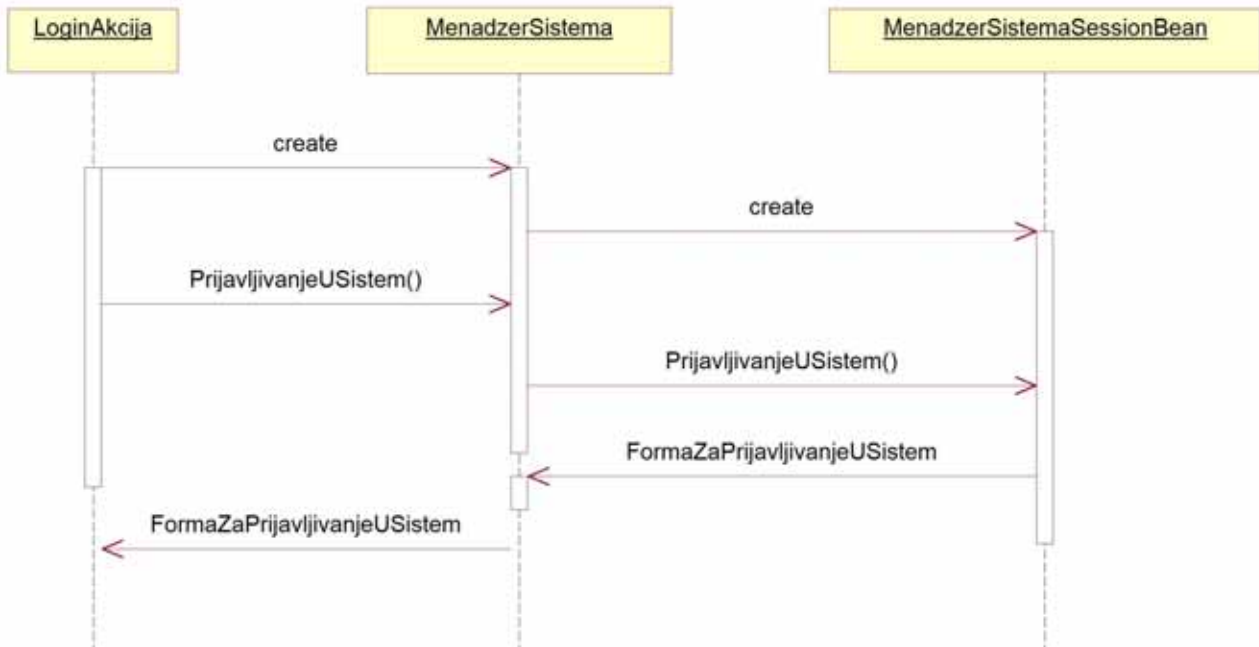
UGOVOR UG27: PrijavljivanjeUSistem

Operacija: PrijavljivanjeUSistem() : signal;

Veza sa SK: DS13

Preduslovi: -.

Postuslovi: Prikazana je forma za prijavljivanje u sistem.



Slika 82: Sekvencni dijagram metode PrijavljivanjeUSistem()

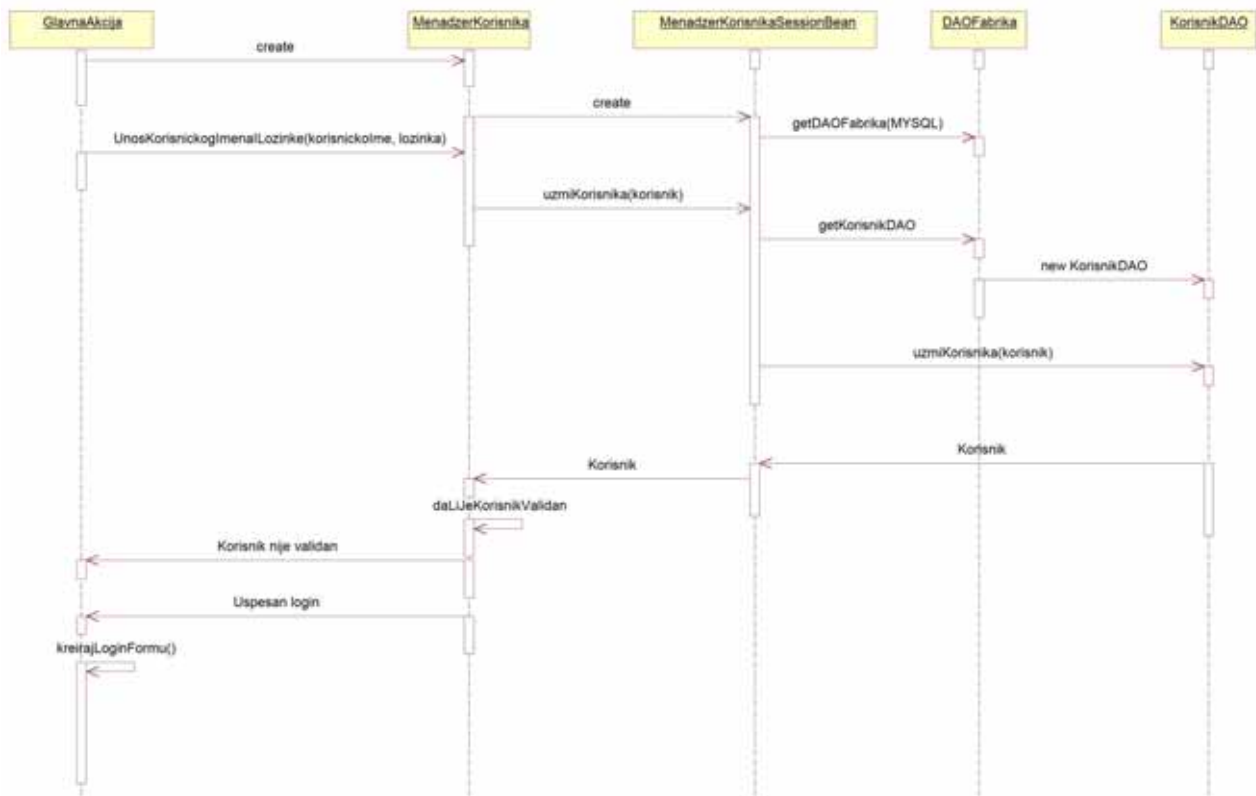
UGOVOR UG28: UnosKorisnickogImenaILOzinke

Operacija: UnosKorisnickogImenaILOzinke(korisnickoIme:String, lozinka:String) : signal;

Veza sa SK: DS13

Preduslovi: Prikazana je forma za prijavljivanje u sistem.

Postuslovi: Korisnik je prijavljen u sistem i prikazana mu je forma za rad sa sistemom.



Slika 83: Sekvencni dijagram metode UnosKorisnickogImenaILOzinke()

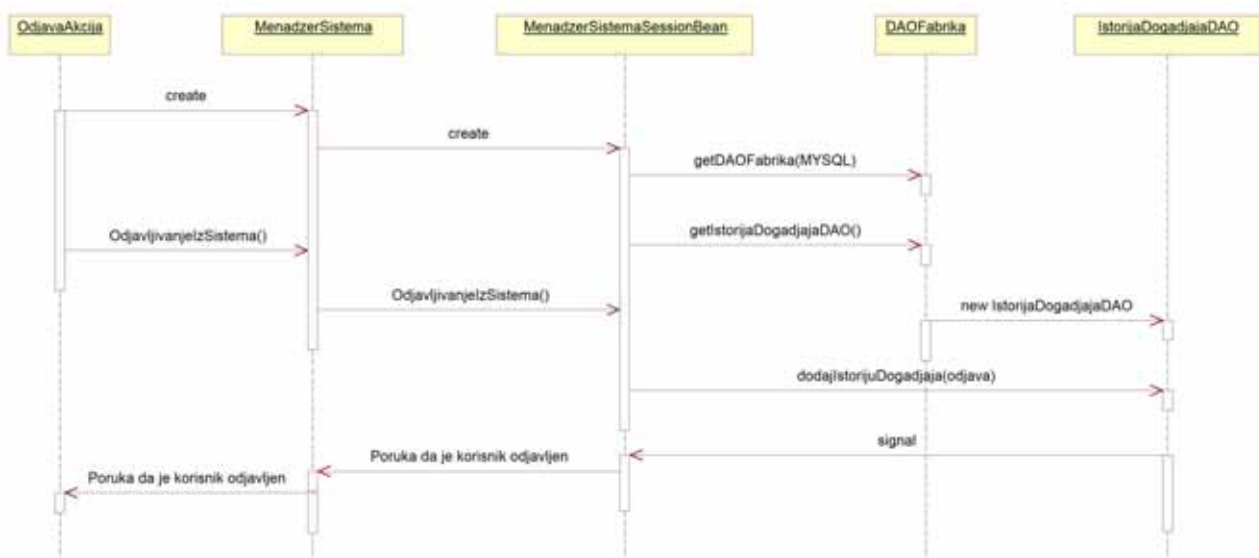
UGOVOR UG29: OdjavljivanjeIzSistema

Operacija: `OdjavljivanjeIzSistema()` : signal;

Veza sa SK: DS14

Preduslovi: Korisnik je prijavljen u sistem.

Postuslovi: Korisnik je odjavljen iz sistema i prikazana mu je forma za prijavljivanje.



Slika 84: Sekvenčni dijagram metode `OdjavljivanjeIzSistema()`

Klase projektovanja ponašanja i interfejsi

Klase projektovanja koje predstavljaju izlaz iz projektovanja ponašanja sistema i projektovanja strukture su sledeće (ovde su navedeni samo nazivi klasa, zbog smanjenja prostora), po paketima:

DAO:

Akcije: `AkcijeDAO` (interfejs), `MySQLAkcijeDAO` (konkretna klasa).

Dokumenti: `DokumentDAO`, `MySQLDokumentDAO`.

DozvoljenaAkcija: `DozvoljeneAkcijeDAO`, `MySQLDozvoljeneAkcijeDAO`.

Emailovi: `EmailDAO`, `MySQLEmailDAO`.

Firme: `FirmaDAO`, `MySQLFirmaDAO`.

Grupe: `GrupaDAO`, `MySQLGrupaDAO`.

GrupeDokumenata: `GrupaDokumenataDAO`, `MySQLGrupaDokumenataDAO`.

IstorijeDogadjaja: `IstorijaDogadjajaDAO`, `MySQLIstorijaDogadjajaDAO`.

Korisnici: `KorisnikDAO`, `MySQLKorisnikDAO`.

Lokacije: `LokacijaDAO`, `MySQLLokacijaDAO`.

Objekti: `ObjekatDAO`, `MySQLObjekatDAO`.

Porekla: `PorekloDAO`, `MySQLPorekloDAO`.

PriloziUzEmail: `PrilogUzEmailDAO`, `MySQLPrilogUzEmailDAO`.

PripadnostiDokumenataGrupamaDokumenata: `PripadnostDokumentaGrupiDokumenataDAO`, `MySQLPripadnostDokumentaGrupiDokumenataDAO`.

PripadnostiDokumenataObjektima: `PripadnostDokumentaObjektuDAO`, `MySQLPripadnostDokumentaObjektuDAO`.

PripadnostiGrupaDokumenataObjektima: PripadnostGrupeDokumenataObjektuDAO, MySQLPripadnostGrupeDokumenataObjektuDAO.

PripadnostiObjekataObjektima: PripadnostObjektaObjektuDAO, MySQLPripadnostObjektaObjektuDAO.

Revizije: RevizijaDAO, MySQLRevizijaDAO.

SistemskaVarijabla: SistemskeVarijableDAO, MySQLSistemskeVarijableDAO.

SistemskePoruke: SistemskaPorukaDAO, MySQLSistemskaPorukaDAO.

Statusi: StatusDAO, MySQLStatusDAO.

StatusiKorisnika: StatusKorisnikaDAO, MySQLStatusKorisnikaDAO.

TipoviFirmi: TipFirmeDAO, MySQLTipFirmeDAO.

TipoviObjekata: TipObjektaDAO, MySQLTipObjektaDAO.

VerzijeDokumenata: VerzijaDokumentaDAO, MySQLVerzijaDokumentaDAO.

VrsteDokumenata: VrstaDokumentaDAO, MySQLVrstaDokumentaDAO.

VrsteGrupaDokumenata: VrstaGrupeDokumenataDAO, MySQLVrstaGrupeDokumenataDAO.

VrsteObjekata: VrstaObjektaDAO, MySQLVrstaObjektaDAO.

Registri:

Dokumenti: MenadzerDokumenata (biznis delegat), MenadzerDokumenataSessionBean.

Ovaj biznis delegat i SessionBean implementiraju neke sistemske operacije, a spisak njihovih metoda je naveden ispod:

```
public interface MenadzerDokumenataSessionLocalBusiness {
    long dodajVrstuDokumenta(VrstaDokumenta vrstaDokumenta);
    boolean brisiVrstuDokumenta(long sifraVrsteDokumenta);
    boolean izmeniVrstuDokumenta(VrstaDokumenta vrstaDokumenta);
    VrstaDokumenta uzmiVrstuDokumenta(long sifraVrsteDokumenata);
    Collection uzmiSveVrsteDokumenta();
    long dodajStatusDokumenta(Status status);
    boolean brisiStatusDokumenta(long sifraStatusaDokumenta);
    boolean izmeniStatusDokumenta(Status status);
    Status uzmiStatusDokumenta(long sifraStatusaDokumenta);
    Collection uzmiSveStatuseDokumenta();
    long dodajPorekloDokumenta(Poreklo poreklo);
    boolean brisiPorekloDokumenta(long sifraPoreklaDokumenta);
    boolean izmeniPorekloDokumenta(Poreklo poreklo);
    Poreklo uzmiPorekloDokumenta(long sifraPoreklaDokumenta);
    Collection uzmiSvaPoreklaDokumenta();
    long dodajRevizijuDokumenta(Revizija revizija);
```

```
boolean brisiRevizijuDokumenta(long sifraRevizijeDokumenta);
boolean izmeniRevizijuDokumenta(Revizija revizija);
Revizija uzmiRevizijuDokumenta(long sifraRevizijeDokumenta);
Collection uzmiSveRevizijeDokumenta();
long dodajVerzijuDokumenta(VerzijaDokumenta verzijaDokumenta);
boolean brisiVerzijuDokumenta(long sifraVerzijeDokumenta);
boolean izmeniVerzijuDokumenta(VerzijaDokumenta verzijaDokumenta);
VerzijaDokumenta uzmiVerzijuDokumenta(long sifraVerzijeDokumenta);
Collection uzmiSveVerzijeDokumenta();
long dodajDokument(Dokument dokument);
boolean brisiDokument(long sifraDokumenta);
boolean izmeniDokument(Dokument dokument);
Dokument uzmiDokument(long sifraDokumenta);
Dokument uzmiDokumentPoNazivu(java.lang.String nazivDokumenta);
Dokument uzmiDokumentPoBroju(java.lang.String brojDokumenta);
Collection uzmiDokumenteZaGrupuDokumenata(GrupaDokumenata grupaDokumenata);
Collection uzmiDokumenteZaObjekat(Objekat objekat);
Collection uzmiSveDokumente();
Collection uzmiSveDokumenteKojiNemajuPripadnosti();
RowSet uzmiSveDokumenteKojiNemajuPripadnostiRS();
Collection uzmiSveStatuseOsimoVogStatusa(java.lang.String naziv);
Status uzmiStatusDokumenta(java.lang.String naziv);
Collection uzmiDokumentePoStatusu(Status status);
int uzmiBrojDokumenataSaStatusom(Status status);
Collection uzmiSveDokumenteZaArhivara(Korisnik arhivar);
boolean daLiPostojiIstiDokument(String nazivDokumenta, String nazivDatoteke,
long revizija, long verzija);
int uzmiBrojDokumenataZaArhivara(Korisnik arhivar);
Collection uzmiSveDokumenteSaLimitom(long min, long max);
long ukupanBrojDokumenata();
Collection uzmiSveDokumenteKojiNemajuPripadnostSaLimitom(long min, long max);
```

```

long ukupanBrojNedodeljenihDokumenata();

Collection uzmiSveDokumentePoVrsti(VrstaDokumenta vrstaDokumenta);

Collection uzmiSveDokumente(Poreklo poreklo, VrstaDokumenta vrstaDokumenta);

Collection uzmiSveDokumeteZaAutora(Korisnik autor);

Collection uzmiSveDokumenteOsimOvogDokumenta(Dokument dokument);

int uzmiBrojDokumenataZaAutora(Korisnik autor);

int uzmiBrojKorisnikaZaArhivara(Korisnik arhivar);

Collection uzmiDokumentSasRevizijamaIVerzijama(Dokument dokument);
}

```

GrupeDokumenata: MenadzerGrupaDokumenata, MenadzerGrupeDokumenataSessionBean.

Ovaj biznis delegat i SessionBean implementiraju systemske operacije, a spisak njihovih metoda je naveden ispod:

```

public interface MenadzerGrupeDokumenataSessionLocalBusiness {
    long dodajVrstuGrupeDokumenata(VrstaGrupeDokumenata vrstaGrupeDokumenata);

    boolean brisiVrstuGrupeDokumenata(long sifraVrsteGrupeDokumenata);

    boolean izmeniVrstuGrupeDokumenata(VrstaGrupeDokumenata vrstaGrupeDokumenata);

    VrstaGrupeDokumenata uzmiVrstuGrupeDokumenata(long sifraVrsteGrupeDokumenata);

    Collection uzmiSveVrsteGrupaDokumenata();

    long dodajGrupuDokumenata(GrupaDokumenata grupaDokumenata);

    boolean brisiGrupuDokumenata(long sifraGrupeDokumenata);

    boolean izmeniGrupuDokumenata(GrupaDokumenata grupaDokumenata);

    GrupaDokumenata uzmiGrupuDokumenata(long sifraGrupeDokumenata);

    Collection uzmiSveGrupeDokumenata();

    Collection uzmiGrupeDokumenata(Objekat objekat);

    boolean
dodajPripadnostDokumentaGrupiDokumenata(PripadnostDokumentaGrupiDokumenata
pripadnostDokumentaGrupiDokumenata);

    boolean
brisiPripadnostDokumentaGrupiDokumenata(PripadnostDokumentaGrupiDokumenata
pripadnostDokumentaGrupiDokumenata);

    Collection uzmiPripadnostDokumentaGrupiDokumenata(GrupaDokumenata
grupaDokumenata);

    Collection uzmiPripadnostDokumentaGrupiDokumenata(Dokument dokument);

    Collection uzmiGrupeDokumenata(VrstaGrupeDokumenata vrstaGrupeDokumenata);
}

```



```

int uzmiBrojDokumenataZaGrupuDokumenata (GrupaDokumenata grupaDokumenata);
long uzmiPripadnostDokumentaSvimGrupama (Dokument dokument);
Collection uzmiSveGrupeDokumenataOsimVezanih (Dokument dokument);
Collection uzmiGrupeDokumenataZaBroj (java.lang.String broj);
Collection uzmiGrupeDokumenataPoNazivu (java.lang.String naziv);
Collection uzmiGrupeDokumenataZaBrojUgovora (java.lang.String brojUgovora);
Collection uzmiGrupeDokumenataZaNapomenu (java.lang.String napomena);
boolean dodajAutoraZaGrupuDokumenata (AutoriGrupaDokumenata autori);
boolean brisiAutoraZaGrupuDokumenata (AutoriGrupaDokumenata autori);
boolean brisiSveVezeSaAutorimaZaGrupuDokumenata (GrupaDokumenata
grupaDokumenata);
Collection uzmiSveAutoreZaGrupuDokumenata (GrupaDokumenata grupaDokumenata);
int uzmiBrojGrupaDokumenataZaAutora (Korisnik autor);
int uzmiBrojGrupaDokumenataZaArhivara (Korisnik arhivar);
}

```

Komunikacija: MenadzerKomunikacije, MenadzerKomunikacijeSessionBean.

Ovaj biznis delegat i SessionBean implementiraju systemske operacije, a spisak njihovih metoda je naveden ispod:

```

public interface MenadzerKomunikacijeSessionLocalBusiness {
    long dodajSistemskuPoruku (SistemskaPoruka sistemskaPoruka);

    boolean brisiSistemskuPoruku (long sifraSistemskePoruke);

    boolean izmeniSistemskuPoruku (SistemskaPoruka sistemskaPoruka);

    SistemskaPoruka uzmiSistemskuPoruku (long sifraSistemskePoruke);

    Collection uzmiSveSistemskePoruke ();

    int brisiIsteklePoruke ();

    Collection uzmiPorukeZaPrimaoca (Korisnik korisnik);

    int uzmiBrojNeprocitanihPorukaZaPrimaoca (Korisnik korisnik);

    long dodajEmail (Email email);

    boolean brisiEmail (long sifraEmaila);

    boolean izmeniEmail (Email email);

    Email uzmiEmail (long sifraEmaila);

    Collection uzmiSveEmailove ();
}

```

```
boolean dodajPrilogUzEmail(PrilogUzEmail prilogUzEmail);
boolean brisiPrilogUzEmail(PrilogUzEmail prilogUzEmail);
boolean izmeniPrilogUzEmail(PrilogUzEmail prilogUzEmail);
PrilogUzEmail uzmiPrilogUzEmail(PrilogUzEmail prilogUzEmail);
Collection uzmiSvePrilogeUzEmailove();
boolean brisiSveSystemskePoruke(long primaocID);
}
```

Korisnici: MenadzerKorisnika, MenadzerKorisnikaSessionBean.

Ovaj biznis delegat i SessionBean implementiraju systemske operacije, a spisak njihovih metoda je naveden ispod:

```
public interface MenadzerKorisnikaSessionLocalBusiness {
    long dodajKorisnika(Korisnik korisnik);

    boolean izmeniKorisnika(Korisnik korisnik);
    boolean brisiKorisnika(long sifraKorisnika);
    Korisnik uzmiKorisnika(long sifraKorisnika);
    Collection uzmiKorisnike(Grupa grupa);
    Collection uzmiKorisnike(StatusKorisnika statusKorisnika);
    long dodajGrupu(Grupa grupa);
    boolean izmeniGrupu(Grupa grupa);
    Collection uzmiSveKorisnike();
    boolean brisiGrupu(long sifraGrupe);
    Grupa uzmiGrupu(long sifraGrupe);
    Collection uzmiSveGrupe();
    long dodajStatusKorisnika(StatusKorisnika statusKorisnika);
    boolean izmeniStatusKorisnika(StatusKorisnika statusKorisnika);
    boolean brisiStatusKorisnika(long sifraStatusaKorisnika);
    StatusKorisnika uzmiStatusKorisnika(final long sifraStatusaKorisnika);
    Collection uzmiSveStatuseKorisnika();

    boolean proveridiDaLiJeKorisnikValidan(java.lang.String korisnickoIme,
    java.lang.String Sifra);

    long uzmiIDKorisnikaZaKorisnickoIme(java.lang.String korisnickoIme);
    StatusKorisnika uzmiStatusKorisnika(java.lang.String nazivStatusa);
}
```

```
Korisnik uzmiKorisnika(java.lang.String naziv);
Grupa uzmiGrupu(java.lang.String naziv);
Collection uzmiKorisnike(Grupa grupa, StatusKorisnika statusKorisnika);
}
```

Objekti: MenadzerObjekata, MenadzerObjekataSessionBean.

Ovaj biznis delegat i SessionBean implementiraju systemske operacije, a spisak njihovih metoda je naveden ispod:

```
public interface MenadzerObjekataSessionLocalBusiness {
    long dodajLokaciju(Lokacija lokacija);

    boolean brisiLokaciju(long sifraLokacije);

    boolean izmeniLokaciju(Lokacija lokacija);

    Collection uzmiSveLokacije();

    Lokacija uzmiLokaciju(long sifraLokacije);

    long dodajVrstuObjekta(VrstaObjekta vrstaObjekta);

    boolean brisiVrstuObjekta(long sifraVrsteObjekta);

    boolean izmeniVrstuObjekta(VrstaObjekta vrstaObjekta);

    VrstaObjekta uzmiVrstuObjekta(long sifraVrsteObjekta);

    Collection uzmiSveVrsteObjekata();

    long dodajTipFirme(TipFirme tipFirme);

    boolean brisiTipFirme(long sifraTipaFirme);

    boolean izmeniTipFirme(TipFirme tipFirme);

    TipFirme uzmiTipFirme(long sifraTipaFirme);

    Collection uzmiSveTipoveFirmi();

    long dodajFirmu(Firma firma);

    boolean brisiFirmu(long sifraFirme);

    boolean izmeniFirmu(Firma firma);

    Firma uzmiFirmu(long sifraFirme);

    Collection uzmiSveFirme();

    long dodajTipObjekta(TipObjekta tipObjekta);

    boolean brisiTipObjekta(long sifraTipaObjekta);
```

```
boolean izmeniTipObjekta(TipObjekta tipObjekta);

TipObjekta uzmiTipObjekta(long sifraTipaObjekta);

Collection uzmiSveTipoveObjekata();

long dodajObjekat(Objekat objekat);

boolean brisiObjekat(long sifraObjekta);

boolean izmeniObjekat(Objekat objekat);

Objekat uzmiObjekat(long sifraObjekta);

Collection uzmiSveObjekte();

boolean dodajPripadnostObjektaObjektu(PripadnostObjektaObjektu
pripadnostObjektaObjektu);

boolean brisiPripadnostObjektaObjektu(PripadnostObjektaObjektu
pripadnostObjektaObjektu);

Collection uzmiSveObjekteIznad(Objekat objekatIspod);

Collection uzmiSveObjekteIspod(Objekat objekatIznad);

boolean dodajPripadnostGrupeDokumenataObjektu(PripadnostGrupeDokumenataObjektu
pripadnostGrupeDokumenataObjektu);

boolean brisiPripadnostGrupeDokumenataObjektu(PripadnostGrupeDokumenataObjektu
pripadnostGrupeDokumenataObjektu);

Collection uzmiSveGrupeDokumenata(Objekat objekat);

Collection uzmiSveObjekteZaGrupuDokumenata(GrupaDokumenata grupaDokumenata);

boolean dodajPripadnostDokumentaObjektu(PripadnostDokumentaObjektu
pripadnostDokumentaObjektu);

boolean brisiPripadnostDokumentaObjektu(PripadnostDokumentaObjektu
pripadnostDokumentaObjektu);

Collection uzmiSveDokumenteZaObjekat(Objekat objekat);

Collection uzmiSveObjekteZaDokument(Dokument dokument);

Collection uzmiObjekteBezPripadnosti();

Collection uzmiSveObjekteOsimVezanih(Objekat objekat);

int uzmiBrojObjekataZaLokaciju(Lokacija lokacija);

int uzmiBrojObjekataZaTipObjekta(TipObjekta tipObjekta);

int uzmiBrojObjekataZaFirmu(Firma firma);

boolean daLiImaDece(Objekat objekat);

TipFirme uzmiTipFirme(java.lang.String tipFirme);

Collection uzmiFirme(TipFirme tipFirme);
```

```
Collection uzmiSveObjekteOsimVezanih(GrupaDokumenata grupaDokumenata);
int uzmiBrojObjekataZaGrupuDokumenata(GrupaDokumenata grupaDokumenata);
long uzmiPonavljanjeDokumentaZaSveObjekte(Dokument dokument);
Collection uzmiSveObjekteOsimVezanih(Dokument dokument);
Collection uzmiObjektePoNazivu(java.lang.String naziv);
Collection uzmiObjektePoBroju(java.lang.String broj);
Collection uzmiObjektePoNapomeni(java.lang.String napomena);
int uzmiBrojObjekataZaArhivara(Korisnik arhivar);
}
```

Sistem: MenadzerSistema, MenadzerSistemaSessionBean.

Ovaj biznis delegat i SessionBean implementiraju systemske operacije, a spisak njihovih metoda je naveden ispod:

```
public interface MenadzerSistemaSessionLocalBusiness {
long dodajAkciju(Akcije akcije);

boolean brisiAkciju(long sifraAkcije);

boolean izmeniAkciju(Akcije akcije);

Akcije uzmiAkciju(long sifraAkcije);

Collection uzmiSveAkcije();

boolean dodajDozvoljenuAkciju(DozvoljeneAkcije dozvoljenaAkcija);

boolean brisiDozvoljenuAkciju(DozvoljeneAkcije dozvoljeneAkcije);

Collection uzmiDozvoljeneAkcije(Grupa grupa);

Collection uzmiDozvoljeneAkcije(Akcije akcije);

Collection uzmiSveDozvoljeneAkcije();

long dodajSistemskeVarijablu(SistemskeVarijable sistemskeVarijable);

boolean brisiSistemskeVarijable(long sifraSistemskeVarijable);

boolean izmeniSistemskeVarijablu(SistemskeVarijable sistemskeVarijable);

SistemskeVarijable uzmiSistemskeVarijable(long sifraSistemskeVarijable);

Collection uzmiSveSistemskeVarijable();

long dodajIstorijuDogadjaja(IstorijaDogadjaja istorijaDogadjaja);

boolean brisiIstorijuDogadjaja(long sifraIstorijeDogadjaja);

boolean brisiIstorijuDogadjaja(Korisnik korisnik);
```

```

boolean brisiIstorijuDogadjaja(Dokument dokument);
Collection uzmiIstorijeDogadjaja(Korisnik korisnik);
Collection uzmiIstorijeDogadjaja(Dokument dokument);
IstorijaDogadjaja uzmiIstorijuDogadjaja(long sifraIstorijeDogadjaja);
Collection uzmiSveIstorijeDogadjaja();
Collection uzmiSveDozvoljeneAkcije(Grupa grupa, Akcije roditeljAkcija);
Collection uzmiIstorijuDogadjaja(Korisnik korisnik, int limit);
SystemskeVarijable uzmiSystemskuVarijablu(java.lang.String naziv);
Akcije uzmiAkciju(java.lang.String nazivAkcije);
boolean daLiJeAkcijaDozvoljenaGrupi(Grupa grupa, Akcije akcija);
boolean brisiSveIstorijeDogadjaja();
}

```

ServisLokator:

ServisLokator i KesiraniServisLokator.

Util:

Datum.

Akcije:

Akcije realizuju direktno neke od sistemskih operacija.

GlavnaAkcija, GlavniMeniAkcija, LoginAkcija, OdjavaAkcija, OSistemuAkcija, PomocAkcija.

Akcije za dokumente: ArhivirajDokumentAkcija, PopuniInformacijeODokumentuAkcija, PosaljiDokumentAkcija, PoveziDokumentAkcija, SacuvajDokumentAkcija, SacuvajUploadAkcija, AdministracijaDokumenataAkcija, DokumentMeniAkcija, DokumentUProzoruAkcija, IstorijaDokumentaAkcija, IzmeniDokumentAkcija, PoveziDokumentSaGrupomAkcija, PoveziDokumentSaObjektomAkcija, PoveziSaDokumentomAkcija, PregledDokumentaAkcija, PreuzmiDokumentAkcija, PrikaziDokumentAkcija, PrikaziDokumenteAkcija, SacuvajIzmenjenDokumentAkcija, SacuvajPoveziDokumentSaGrupomAkcija, SacuvajPoveziDokumentSaObjektomAkcija, SacuvajPoveziSaDokumentomAkcija, SacuvajZameniSaDokumentomAkcija, VerzijeDokumentaAkcija, ZameniSaDokumentomAkcija.

Akcije za transfer: ZipIzvozAkcija.

Utility akcije: HTMLReviser.

Akcije za firme: BrisiFirmuAkcija, IzmeniFirmuAkcija, KreirajFirmuAkcija, PrikaziFirmeAkcija, PromeniFirmuAkcija, SacuvajFirmuAkcija.

Akcije za grupe dokumenata: AdministracijaGrupaDokumenataAkcija, BrisiGrupuDokumenataAkcija, IzmeniGrupuDokumenataAkcija, KreirajGrupuDokumenataAkcija, PoveziGrupuSaObjektomAkcija, PrikaziGrupeDokumenata, PrikaziGrupuDokumenata,

PrikaziVrstuGrupeDokumenata, PromeniGrupuDokumenataAkcija,

SacuvajGrupuDokumenataAkcija, SacuvajPoveziGrupuSaObjektomAkcija.

Akcije za komunikacije: BrisiSvePorukeAkcija, IzaberiSistemskuPorukuAkcija, IzbrisiSistemskuPorukuAkcija, KreirajNovuPorukuAkcija, PrikaziSistemskePorukeAkcija, SacuvajNovuPorukuAkcija.

Akcije za korisnike: AdministracijaKorisnikaAkcija, BrisiKorisnikaAkcija, IstorijaKorisnikaAkcija, IzmeniKorisnikaAkcija, KreirajKorisnikaAkcija, PrikaziGrupeKorisnikaAkcija, PrikaziKorisnikaAkcija, PrikaziKorisnikeAkcija, PromeniKorisnikaAkcija, SacuvajKorisnikaAkcija.

Akcije za lokacije: BrisiLokacijuAkcija, IzmeniLokacijuAkcija, KreirajLokacijuAkcija, PrikaziLokacijeAkcija, PromeniLokacijuAkcija, SacuvajLokacijuAkcija.

Akcije za objekte: AdministracijaObjekataAkcija, BrisiObjekatAkcija, IzmeniObjekatAkcija, KreirajObjekatAkcija, objektiDrvoLevoFrejmBean, PoveziObjekatSaObjektomAkcija, PrikaziObjekteAkcija, PrikaziObjekteHijerarhijskiAkcija, RaskiniVezuAkcija, SacuvajIzmenjenObjekatAkcija, SacuvajObjekatAkcija, SacuvajPoveziObjekatSaObjektomAkcija.

Akcije za pretragu: DokumentIndekser, Pretraga, PretragaOpcije, PretraziAkcija, PretraziObjekteIGrupeAkcija, PretraziSledeciAkcija, ProsirenaPretragaAkcija, Rezultat, SquareSimilarity.

Akcije za sistem: AdministracijaSistemaAkcija, BrisiSveDogadjajeAkcija, DirektorijiAkcija, IndeksirajPonovoAkcija, IstorijaSistemaAkcija, OtkljucajAkcija, PretragaPodesavanjaAkcija, PromeniDirektorijeAkcija.

Akcije za tipove objekata: BrisiTipObjektaAkcija, IzmeniTipObjektaAkcija, KreirajTipObjektaAkcija, PrikaziTipoveObjekataAkcija, PromeniTipObjektaAkcija, SacuvajTipObjektaAkcija.

Forme:

DokumentForma, FirmaForma, GrupaDokumenataForma, IndeksForma, KonekcijaForma, KorisnikForma, LoginForm, LokacijaForma, NovaPorukaForm, ObjekatForma, PoveziDokumentForma, PoveziGrupuForma, PoveziObjekatForma, PretraziForm, PutanjaForma, SistemForma, TipObjektaForma, UploadForma.

Osnovno:

DatotekaBean, JarBean, KriptoBean, MenadzerSesije, XMLBean, ZipBean.

FajlSistem: Cuvar, FajlSistemCuvar.

Konfiguracija: DBMSKonfigurator, KonfiguratorRasporedaStrane, KonfiguratorSkladistaPodataka.

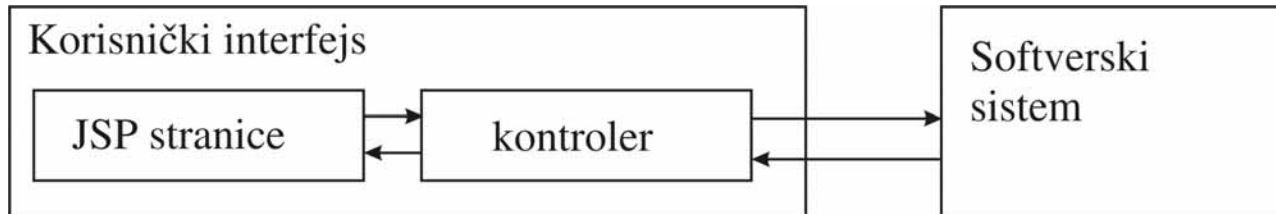
Tekst: AnalizirajTekst, ParserTeksta, AnalizaFabrika, AnalizaRezultat, Analyzer, RangerReci, Rec, RecUlaz, Stemmer, StopReci, StopTabela, TabelaReci, EngleskiTezinar, IzbornikJezika, NemackiTezinar, SrpskiTezinar, Tezinar, DOCParser, HTMLParser, Parser, ParserFabrika, PDFParser, PPTParser, PSParser, RTFParser, TXTParser, XLSParser, XLSRecordListener, XMLParser.

Transfer: Izvoz, Writer2Stream, ZipIzvoz.

Korisnički interfejsi

Korisnički interfejs se sastoji od:

- Ekranskih formi u obliku jsp strana (korišćenjem Struts tag biblioteka) i
- Kontrolera korisničkog interfejsa (ovde je to Struts-ov ActionServlet).



Slika 85: Struktura korisničkog interfejsa

Projektovano je i implementirano sedamdeset i pet ekranskih formi koje su podeljene i organizovane u osam posebnih celina shodno njihovoj nameni. Pri pokretanju programa prikazuje se forma koja je uvijek pokrenuta i predstavlja osnovno korisnikovo okruženje za rad u programu. Iz forme "Glavna" korisnik, shodno poslu koji želi da uradi, pokreće neku od formi tako što prvo bira stavku iz glavnog menija levo, a broj stavki i njihova vrsta će zavisiti od grupe kojoj korisnik pripada. Ovde će biti prikazane osnovne forme za rad sa sistemom.

Glavni: (mogu svi da ga vide)

- Login;
- Osnovna (koja u sebi uključuje: MeniLevo, TrakaGore, TrakaDole i GlavnaDesno);
- BrzaPretraga;
- ProsirenaPretraga;
- Pomoc;
- KreirajPoruku;
- ČitajPoruku.

Arhiviranje: (može da je vidi samo Administrator i Arhivar)

- DokumentiUARhivi;
- InformacijeODokumentuOdAutora;
- DokumentArhiviran.

Administracija korisnika: (vide je samo Administratori)

- IzborRadaSaKorisnicima;
- GrupeKorisnika;
- Korisnici;
- KreirajKorisnika.

Slanje dokumenata: (vide je Administratori i Autori)

- SlanjeDokumenta;
- UnosInformacijaODokumentu;
- DokumentPoslat.

Objekti: (vide ga svi, osim Lokacija, TipovaFirmi i Firmi koje vide samo Arhivari i Administratori)

- KreirajObjekat;

- PrikaziObjekte;
- PrikaziObjekteHijerarhijski;
- Lokacije;
- TipoviFirmi;
- Firme.

Grupe dokumenata: (vide ga svi, osim Kreiranja grupe dokumenata kojeg vide samo Arhivari i Administratori)

- KreiranjeGrupeDokumenata;
- PregledGrupaDokumenata.

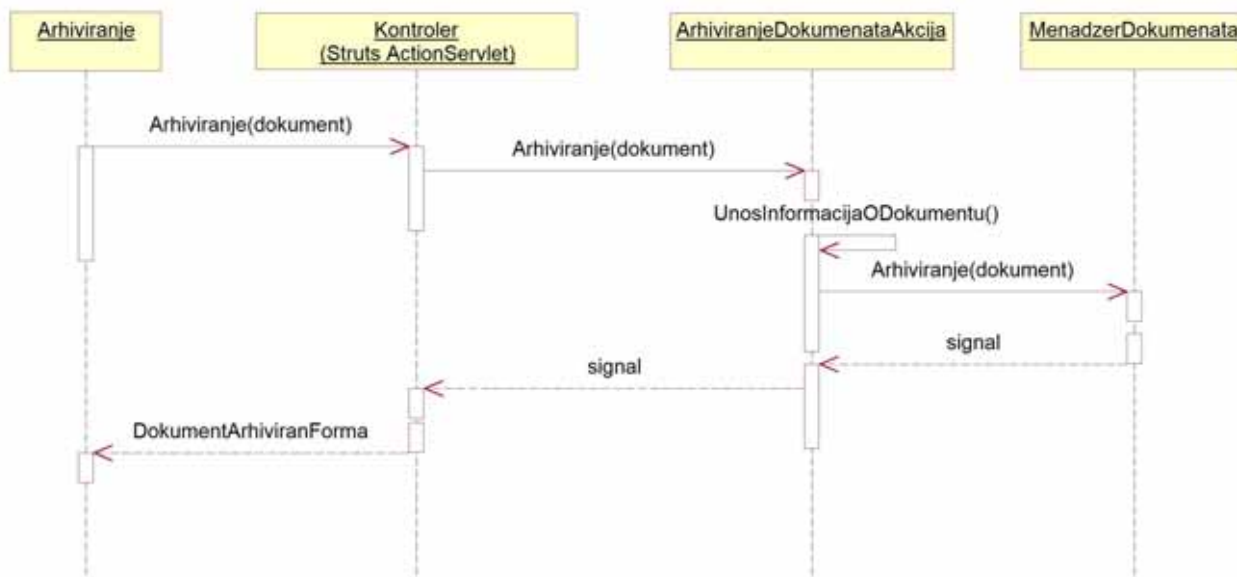
Dokumenti: (vide ga svi)

- PrikazSvihDokumenata;
- PrikazSvihNedodeljenihDokumenata;
- PrikazDokumenataPoVrsti;
- PrikazDokumenataPoPoretku;
- PrikazDokumenataPoAutoru.

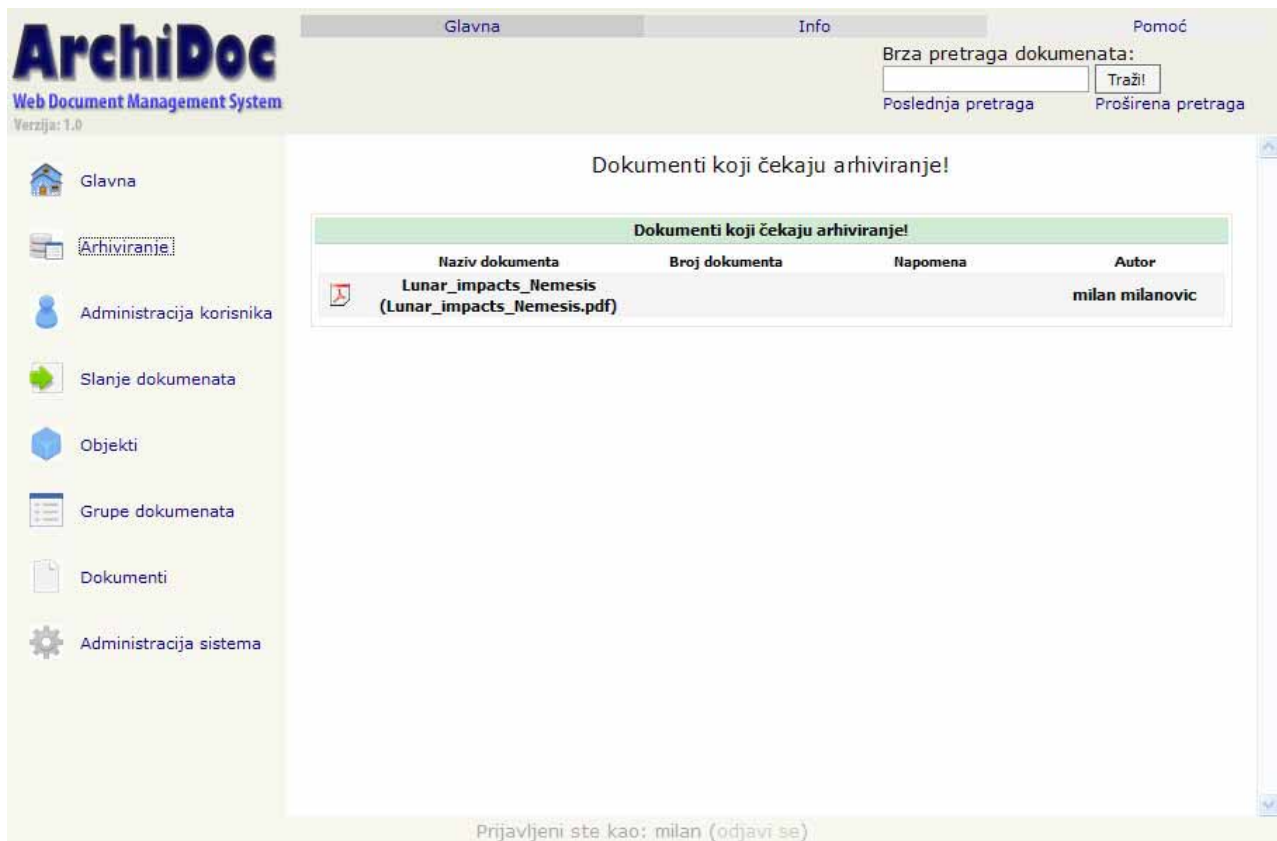
Administracija sistema: (vide ga samo Administratori)

- Direktoriji;
- PodesavanjePretrage;
- IstorijaSistema.

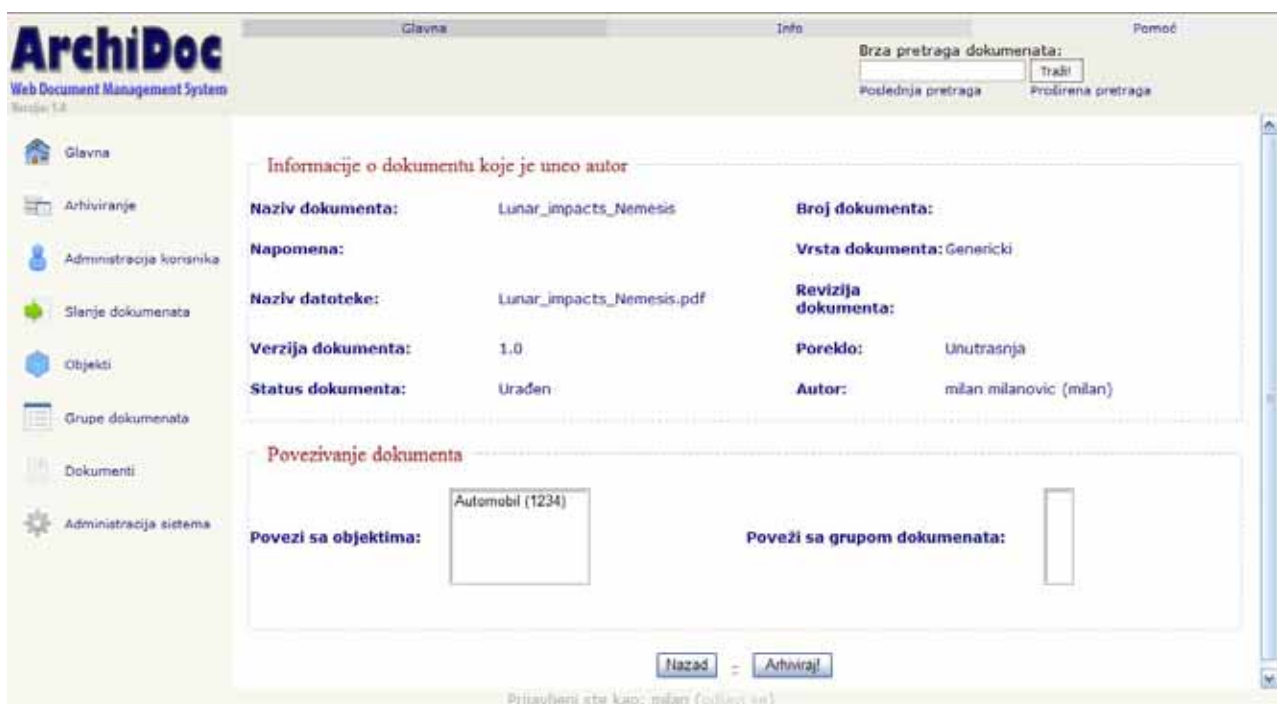
Korisnički interfejs za slučaj korišćenja - *Arhiviranje nove dokumentacije*



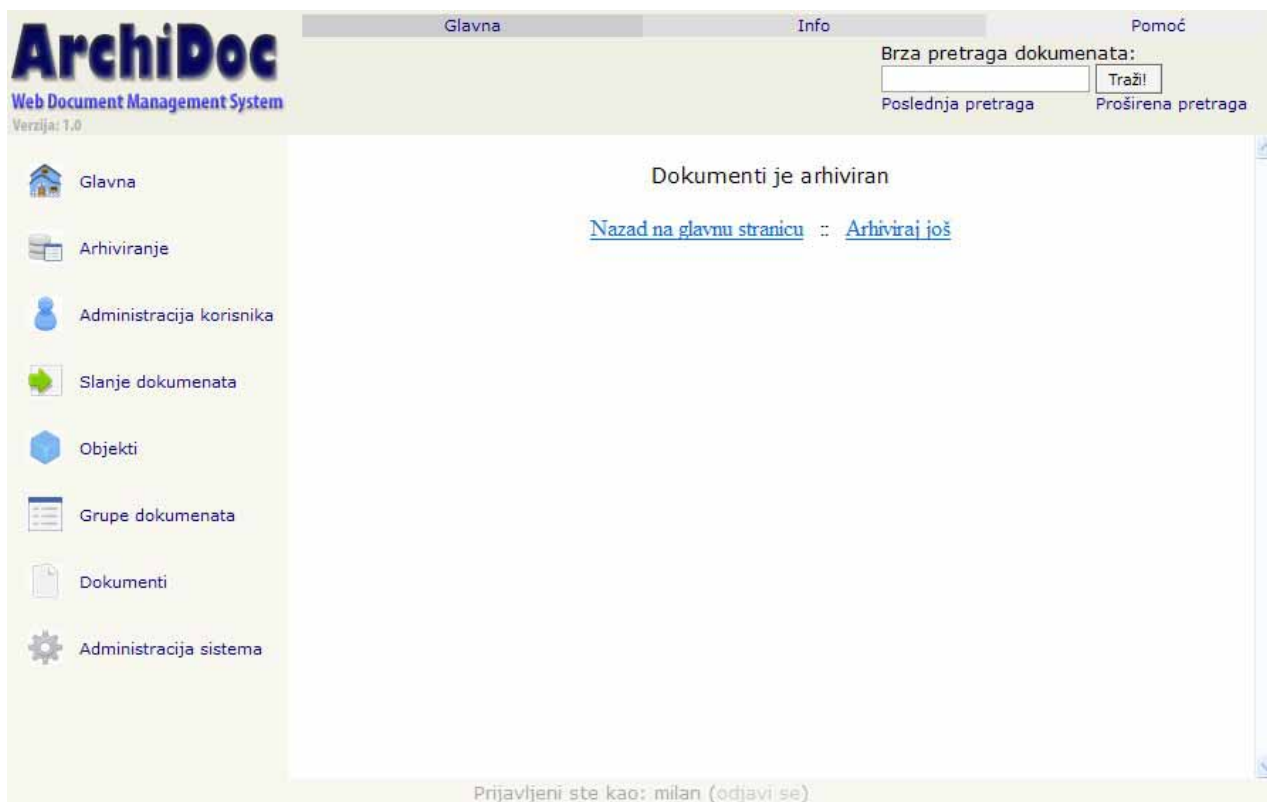
Slika 86: Sekvencni dijagram za SK - Arhiviranje nove dokumentacije



Slika 87: SK Arhiviranje nove dokumentacije - Dokumenti čekaju arhiviranje



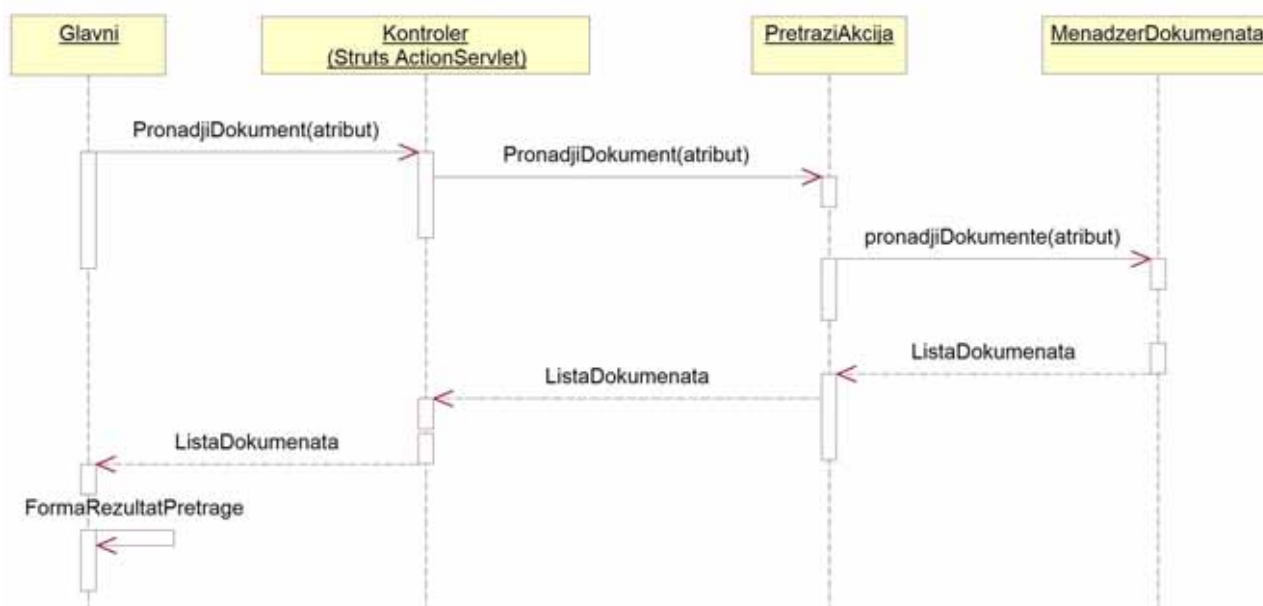
Slika 88: SK Arhiviranje nove dokumentacije - Povezivanje dokumenta sa objektima ili grupama dokumenata



Slika 89: SK Arhiviranje nove dokumentacije - Dokument arhiviran

Korisnički interfejs za slučaj korišćenja - *Pretraživanje dokumenata*

Ovde je prikazan slučaj proširene pretrage, a brza pretraga se pokreće direktno iz glavnog menija. Rezultat pretrage je lista dokumenata, ili objekata.



Slika 90: Sekvenčni dijagram za SK - Pretraživanje dokumenata

Proširena pretraga

Opcije pretrage

Sa svim rečima:	<input type="text"/>
Sa tačnom frazom:	<input type="text"/>
Sa bar jednom od reči:	<input type="text"/>
Bez reči:	<input type="text"/>

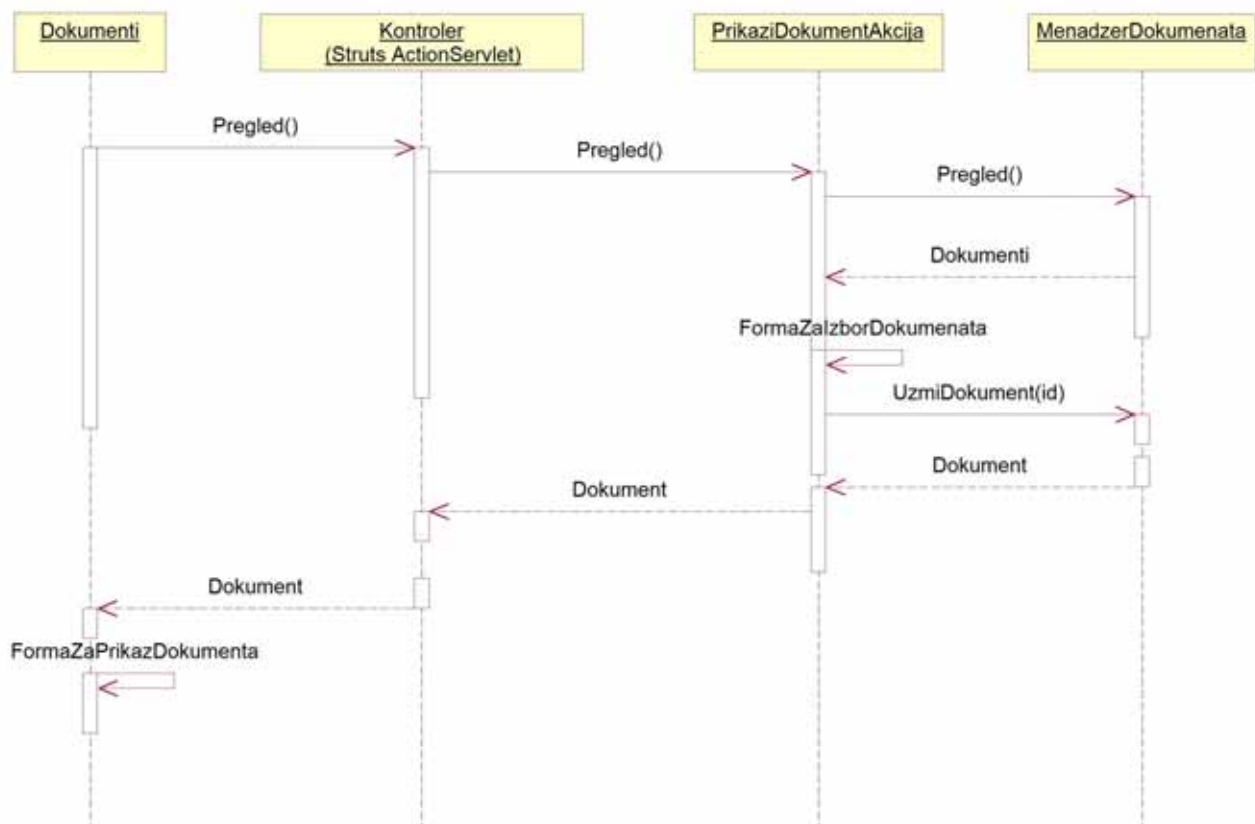
Informacije o pretrazi

Veličina:	<input type="text"/>	Relacije:	<input type="button" value="Vece od"/> ▾
	Bajta	Relacije:	<input type="button" value="Posle"/> ▾
Datum:	<input type="text"/> <input type="button" value="📅"/>	Format:	<input type="button" value="Svi formati"/> ▾
Fuzzy pretraga:	<input type="checkbox"/>		

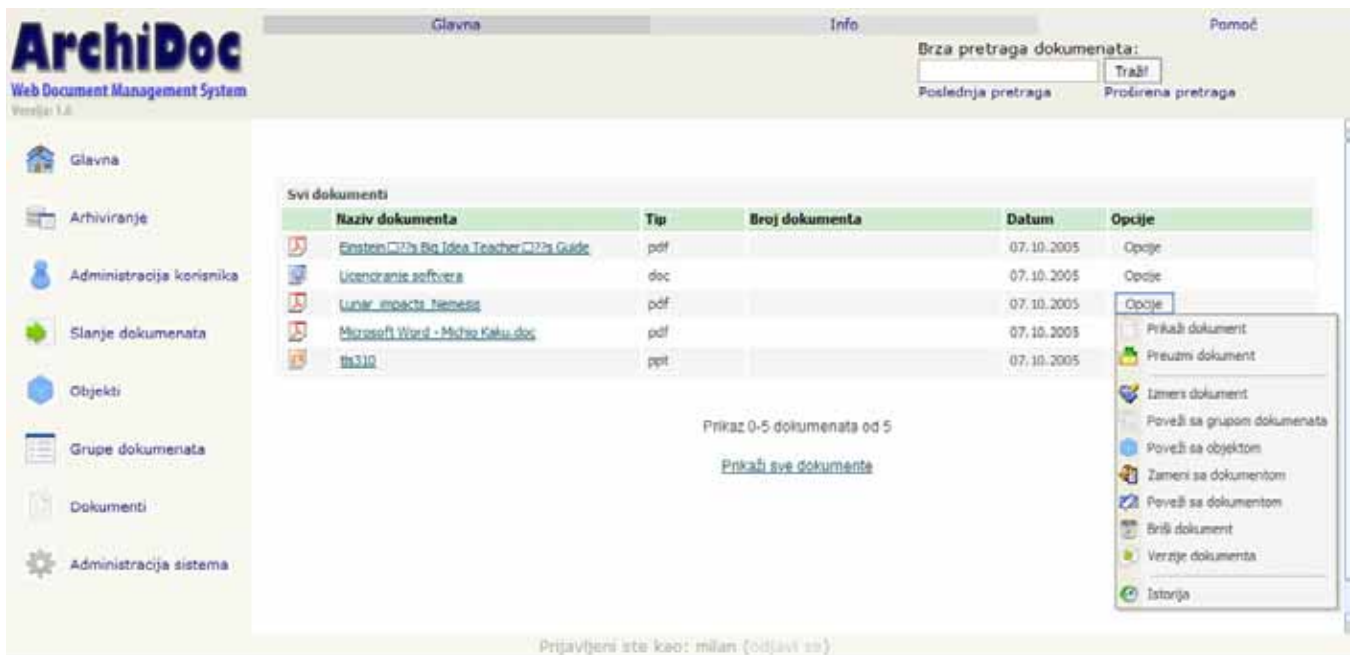
Pretraga objekata i grupa dokumenata

Unesite ključnu reč:

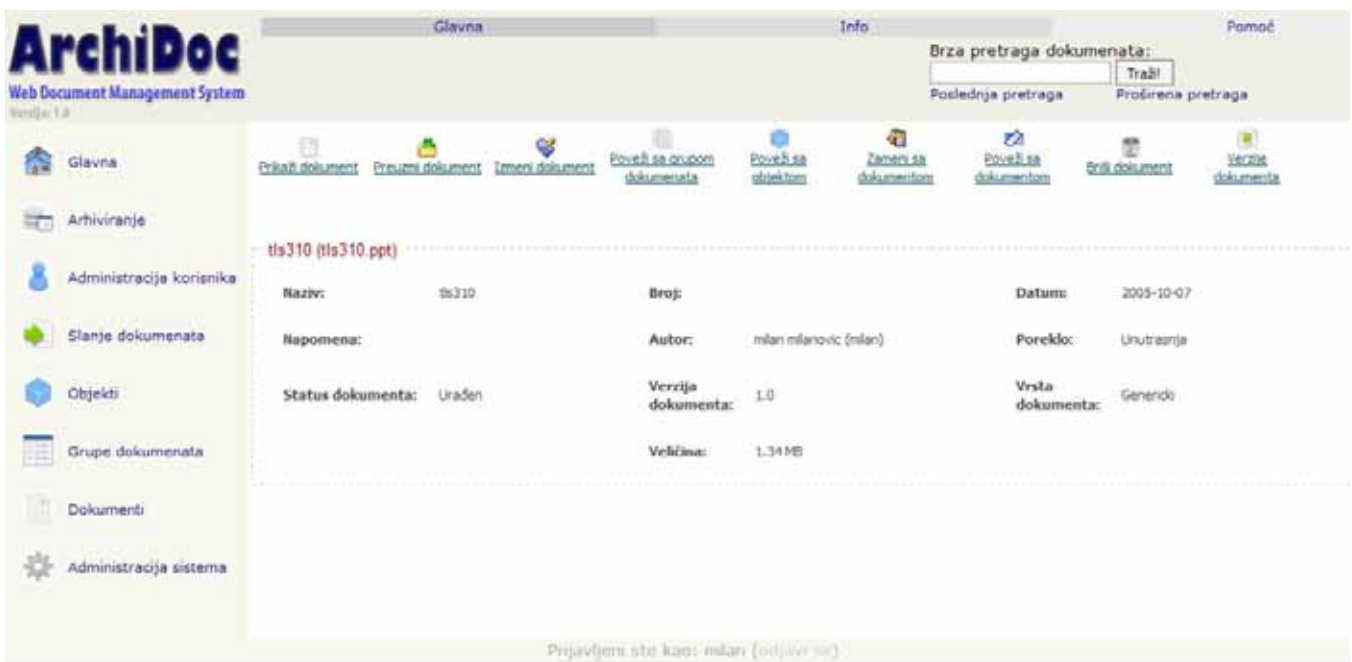
Slika 91: SK Pretraživanje dokumenata - Proširena pretraga

Korisnički interfejs za slučaj korišćenja - *Pregled dokumenta*

Slika 92: Sekvenčni dijagram za SK - Pregled dokumenta

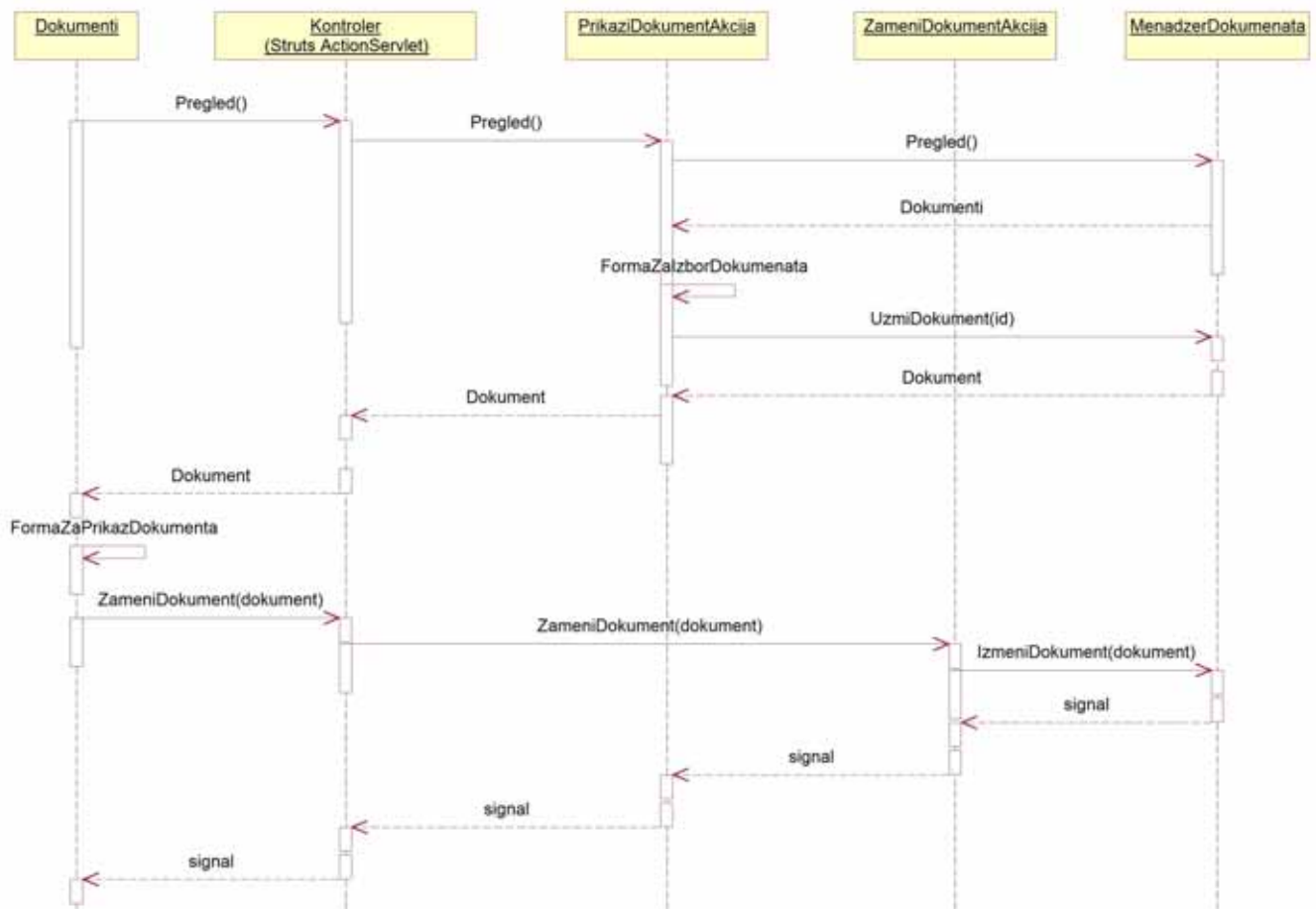


Slika 93: SK Pregled dokumenta - Pregled dokumenata



Slika 94: SK Pregled dokumenta - Pregled detalja o dokumentu

Korisnički interfejs za slučaj korišćenja - *Izmena podataka o dokumentu*

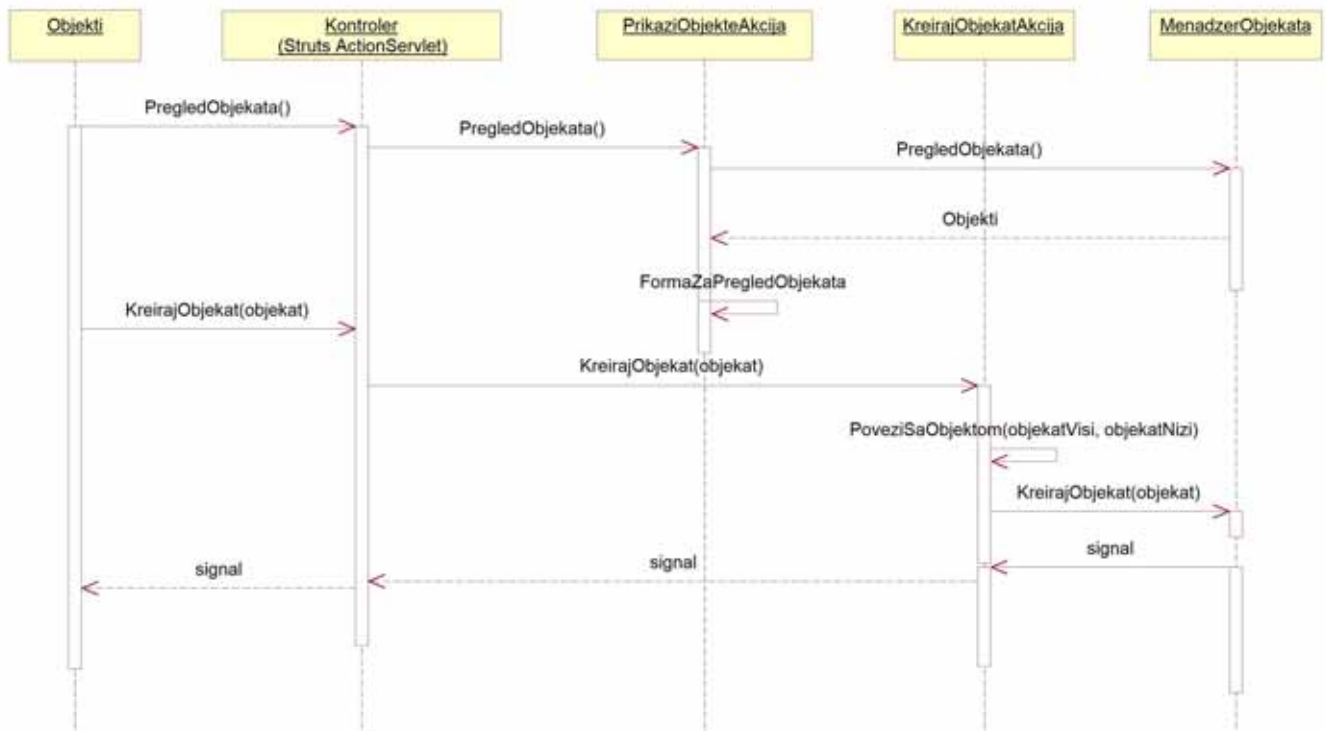


Slika 95: Sekvencni dijagram za SK - Izmjena podataka o dokumentu

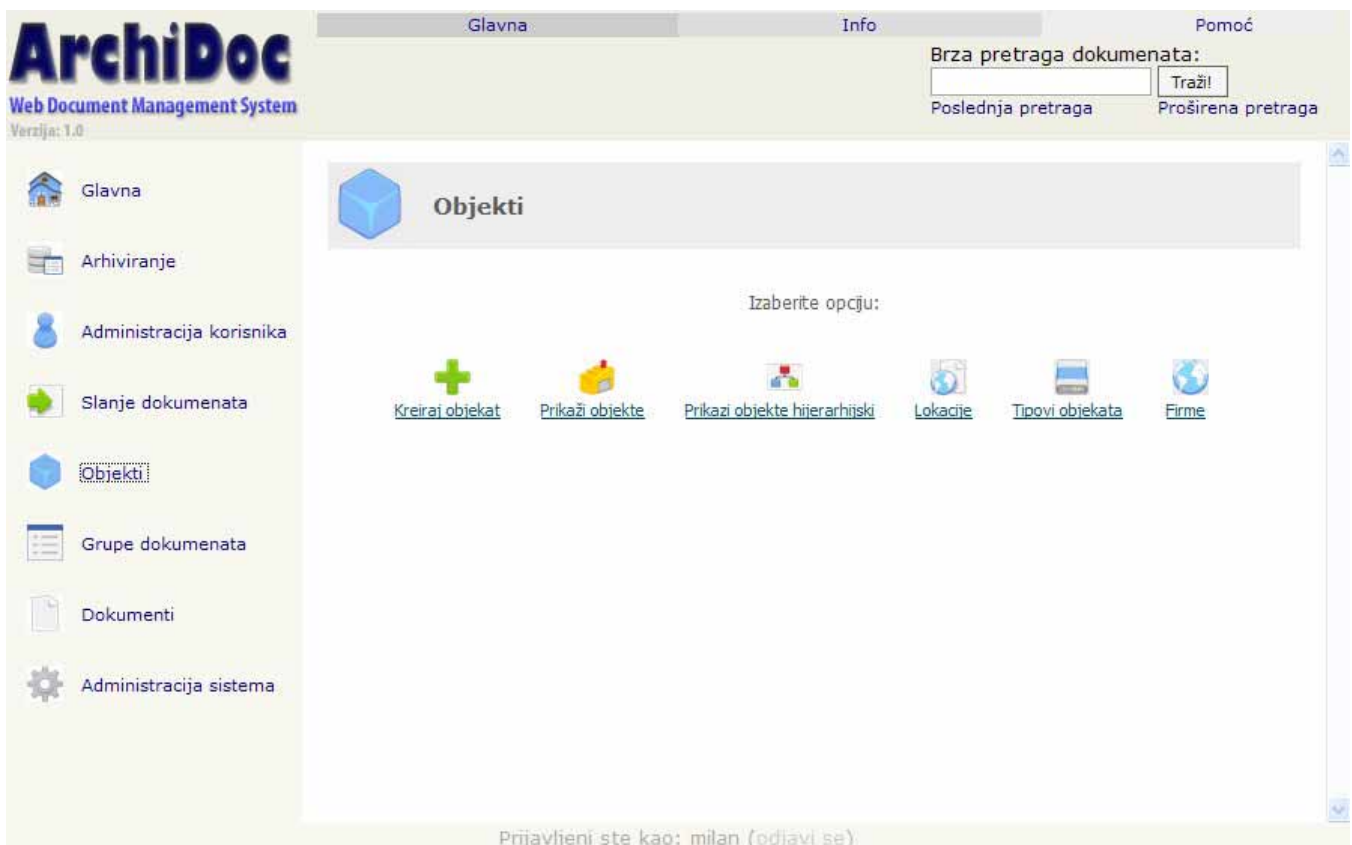


Slika 96: SK Izmjena podataka o dokumentu - Izmjena dokumenta

Korisnički interfejs za slučaj korišćenja - *Administracija objekata*



Slika 97: Sekvencni dijagram za SK - Administracija objekata



Slika 98: SK Administracija objekata - Objekti

ArchiDoc
Web Document Management System
Verzija: 1.0

Glavna Info Pomoć

Brza pretraga dokumenata:
 Traži!
 Poslednja pretraga Proširena pretraga

Kreiranje novog objekta

Naziv objekta*: Broj objekta: Napomena:

Datum*: Lokacija: Vrsta objekta*: **Sistem**

Tip objekta: Firma:

Polja označena * su obavezna

Kreiraj !

Prijavljeni ste kao: milan (odjavi se)

Slika 99: SK Administracija objekata - Kreiranje novog objekta

ArchiDoc
Web Document Management System
Verzija: 1.0

Glavna Info Pomoć

Brza pretraga dokumenata:
 Traži!
 Poslednja pretraga Proširena pretraga

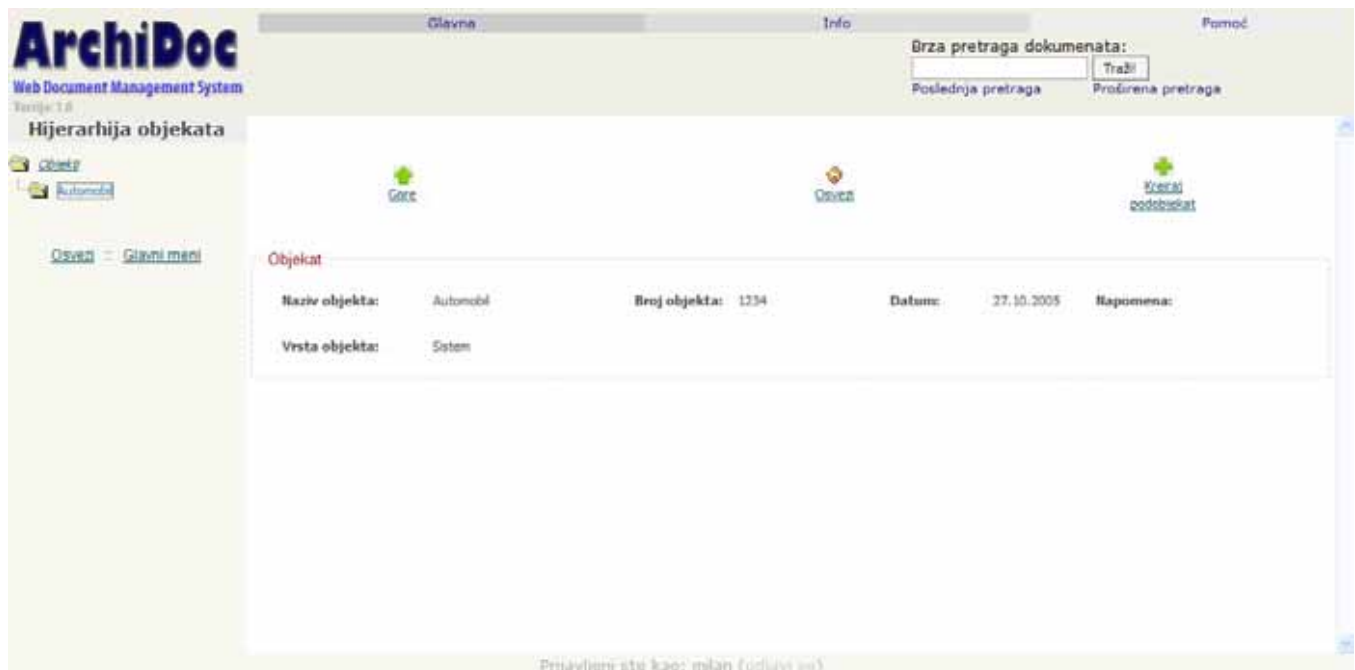
Osveži + Kreiraj objekat

Naziv objekta	Broj objekta	Datum	Opcije
Automobil	1234	27.10.2005	Opcije

- Izaberi objekat
- Izmena objekta
- Brisanje objekta
- Poveži sa drugim objektom
- Izvezi kao zip
- Izvesi sve nivoe kao zip

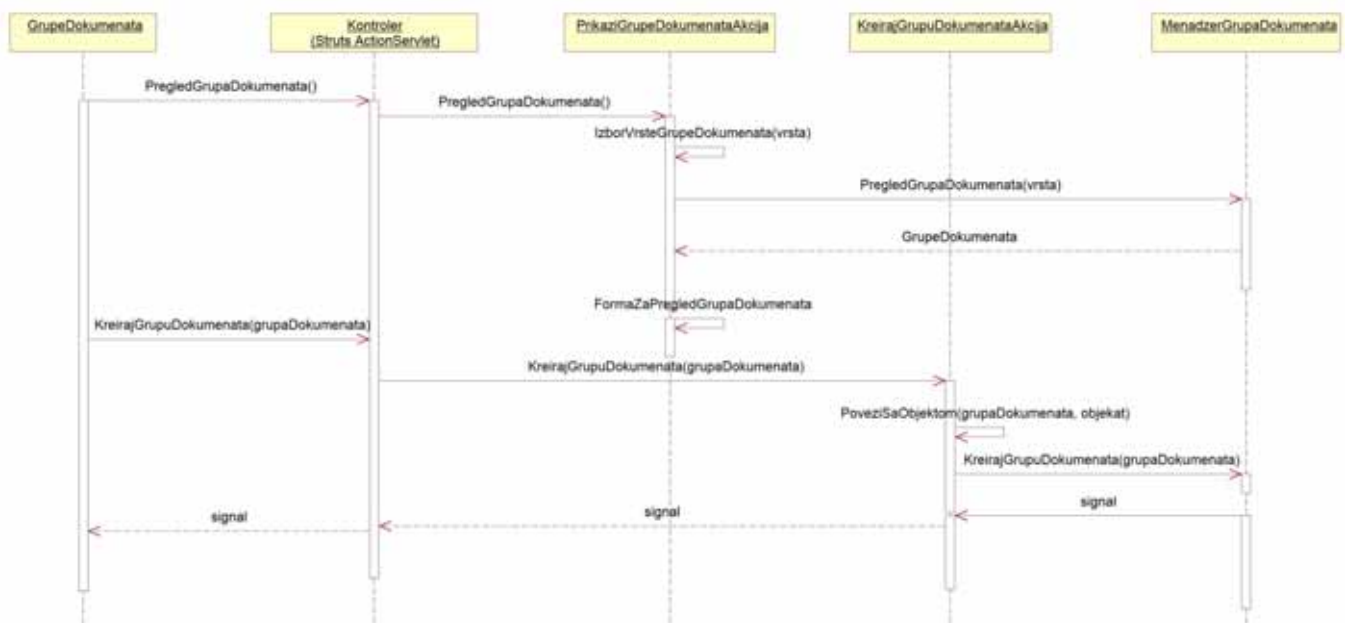
Prijavljeni ste kao: milan (odjavi se)

Slika 100: SK Administracija objekata - Pregled objekata



Slika 101: SK Administracija objekata - Hijerarhija objekata

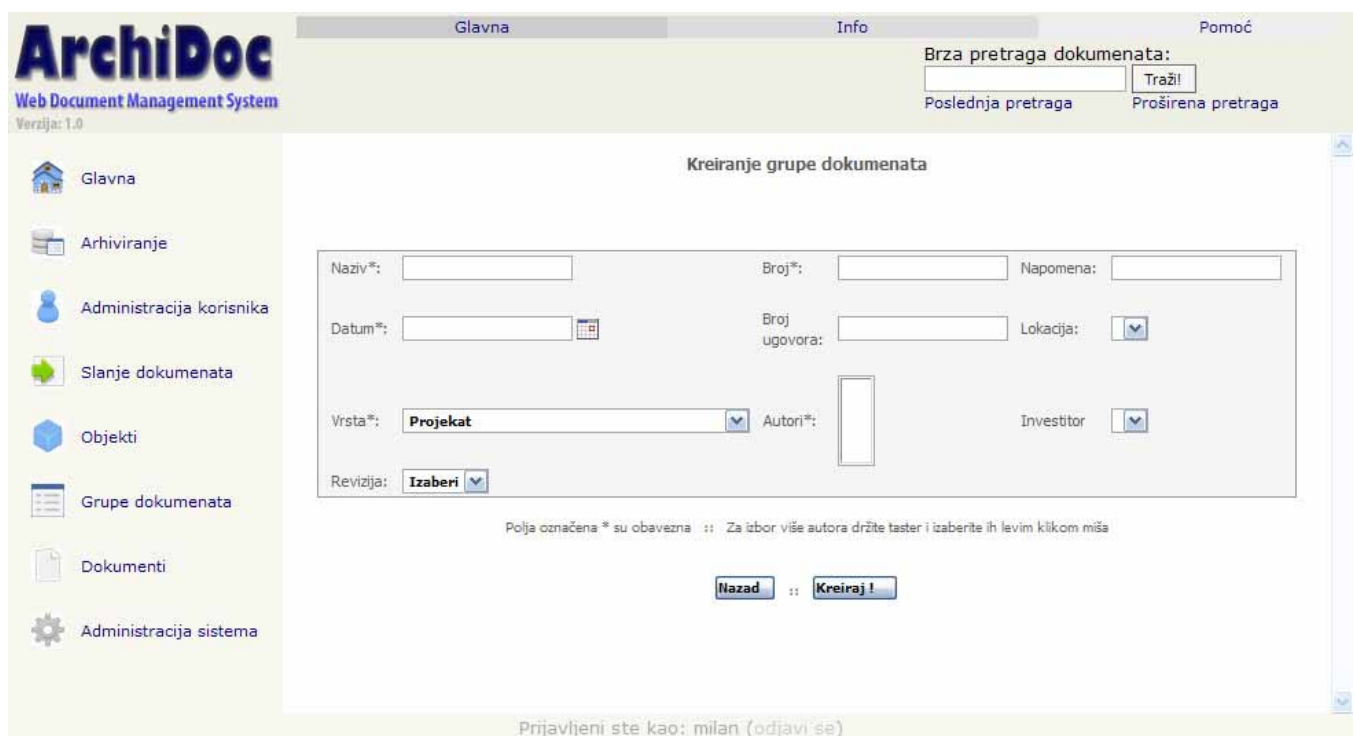
Korisnički interfejs za slučaj korišćenja - *Administracija grupa dokumenata*



Slika 102: Sekvenčni dijagram za SK - Administracija grupa dokumenata



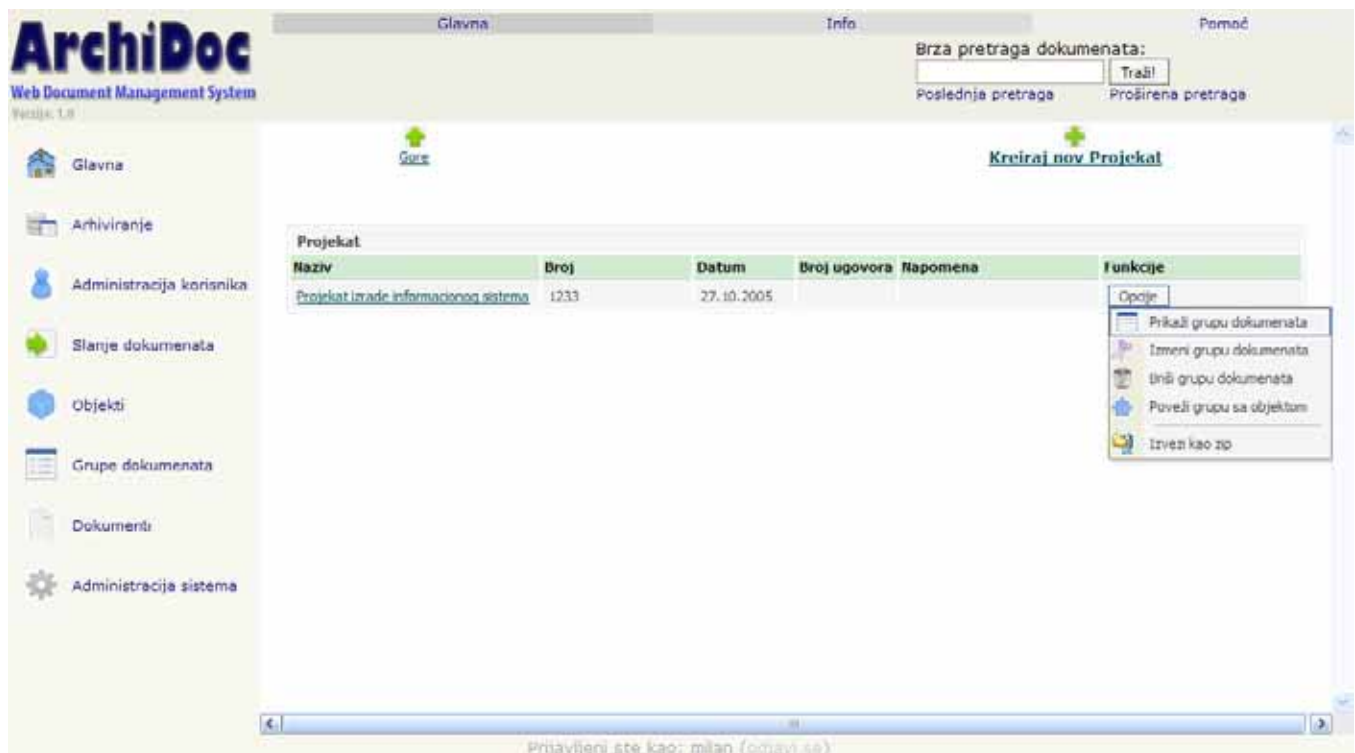
Slika 103: SK Administracija grupa dokumenata - Izbor opcije



Slika 104: SK Administracija grupa dokumenata - Kreiranje grupe dokumenata



Slika 105: SK Administracija grupa dokumenata - Izbor vrste grupe dokumenata



Slika 106: SK Administracija grupa dokumenata - Prikaz konkretne vrste grupe dokumenata (Projekti)

The screenshot shows the ArchiDoc Web Document Management System interface. The main content area displays details for a document group named 'Projekat'. The details include:

- Projekat**
- Naziv:** Projekat izrade informacionog sistema
- Broj:** 1233
- Datum:** 27.10.2005
- Napomena:**
- Broj ugovora:**
- Autor:** Uros Pavlevic
- Vrsta grupe dokumenata:** Projekat

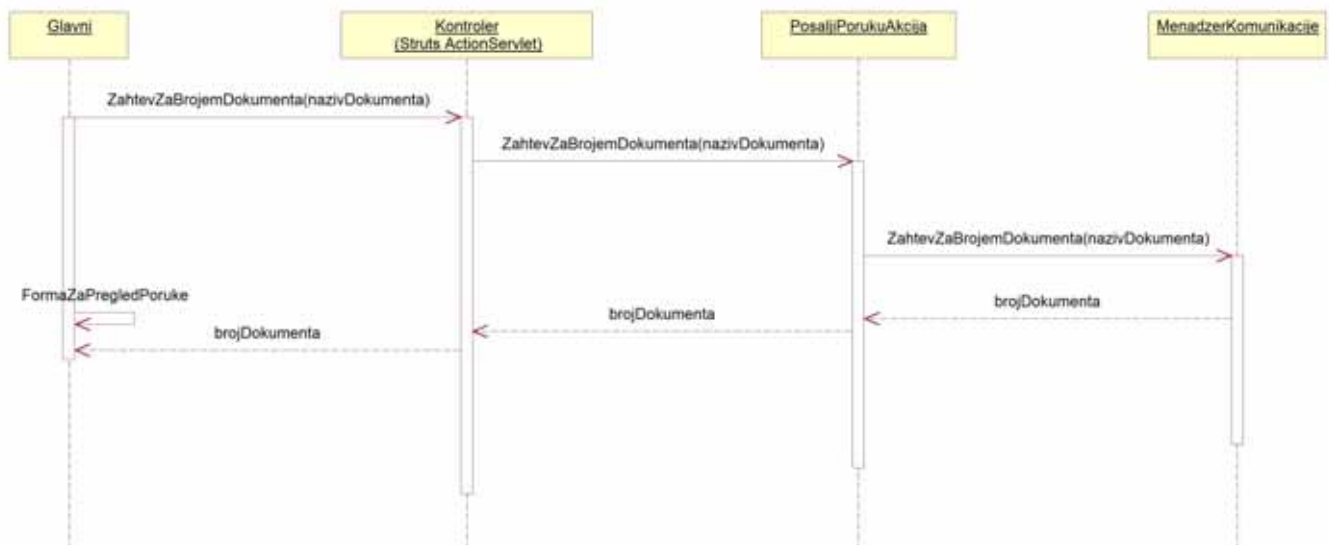
Below the details is a table of documents:

Naziv dokumenta	Broj dokumenta	Datum	Opcije
Licenciranje softvera		07.10.2005	Opcije

The interface also includes a sidebar with navigation options like 'Glavna', 'Arhiviranje', 'Administracija korisnika', 'Slanje dokumenata', 'Objekti', 'Grupe dokumenata', 'Dokumenti', and 'Administracija sistema'. At the top right, there is a search bar for 'Brza pretraga dokumenata' with 'Traži' and 'Poslednja pretraga' buttons. The user is logged in as 'milan (nisi svi se)'.

Slika 107: SK Administracija grupa dokumenata - Prikaz konkretne grupe dokumenata (Projekat)

Korisnički interfejs za slučaj korišćenja - *Kreiranje broja novog dokumenta*



Slika 108: Sekvencni dijagram za SK - Kreiranje broja novog dokumenta

ArchiDoc
Web Document Management System
Verzija: 1.0

Glavna Info Pomoć

Brza pretraga dokumenata:

 Poslednja pretraga Proširena pretraga

Kreiraj poruku

Naslov poruke

Primalac*: milan milanovic (milan) ▼
 Naslov*:
 Potvrda: **Prioritet:** Nizak ▼
 Period važenja (u danima):

Tekst poruke

Tekst poruke:

Polja označena * su obavezna :: [Nazad na glavnu stranicu](#)

Prijavljeni ste kao: milan (odjavi se)

Slika 109: SK Kreiranje broja novog dokumenta - Kreiranje nove poruke

ArchiDoc
Web Document Management System
Verzija: 1.0

Glavna Info Pomoć

Brza pretraga dokumenata:

 Poslednja pretraga Proširena pretraga

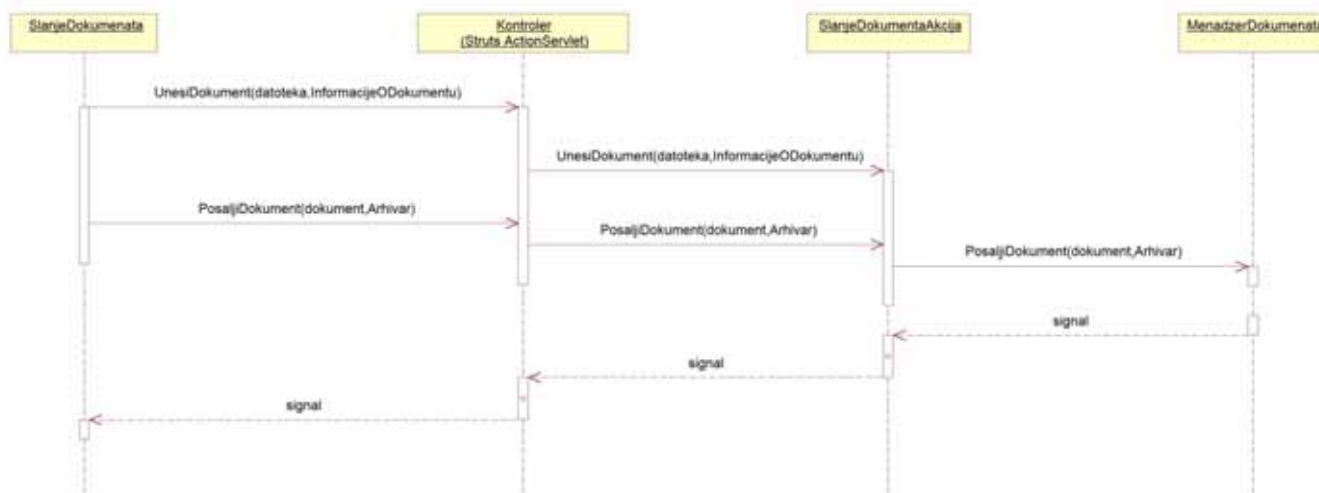
Čitanje poruka

Prioritet	Naslov	Posiljalac	Datum	Brisi
	<u>Poruka u vezi rada!</u>	milan milanovic	07.10.2005	

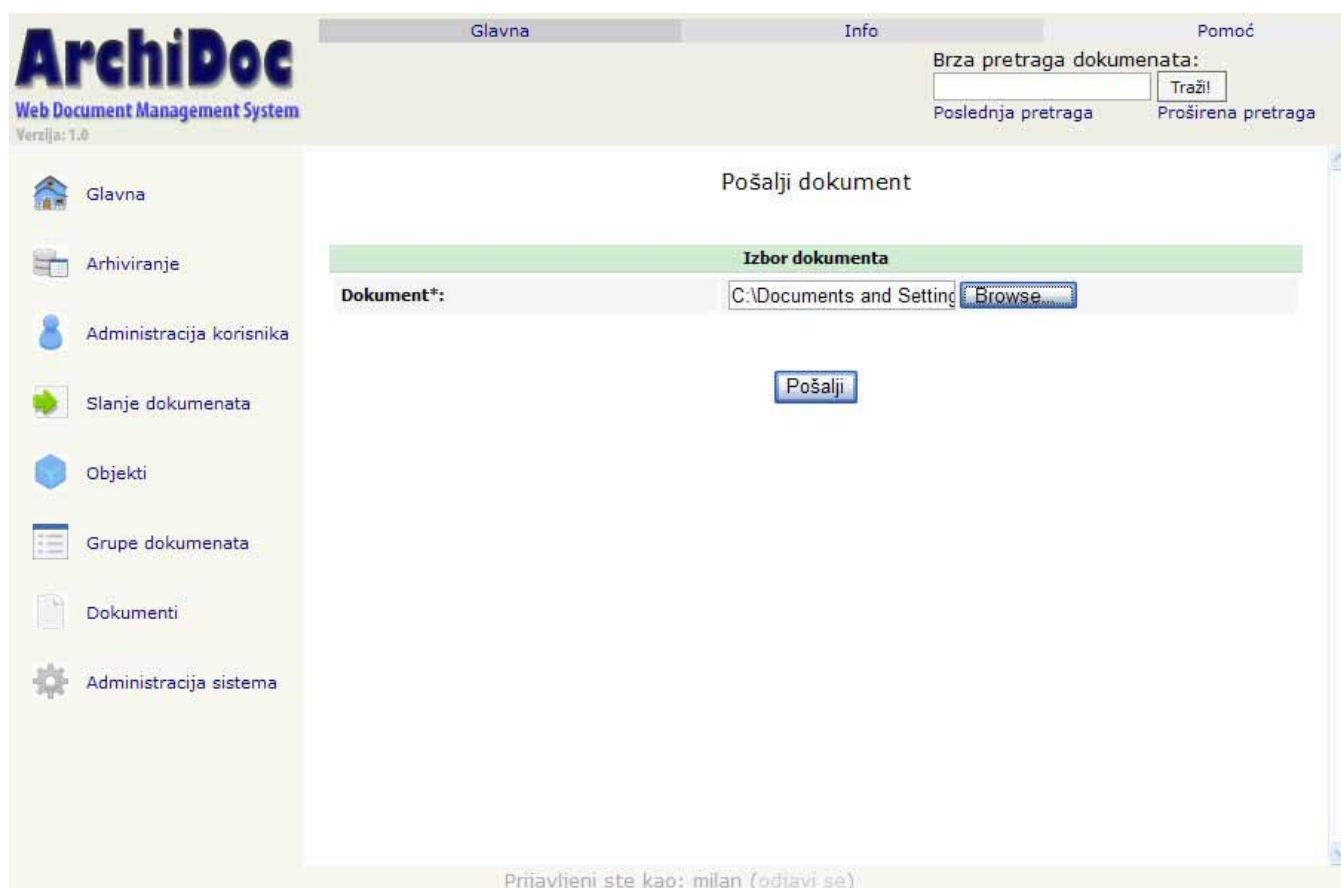
[Brisi sve poruke](#) :: [Nazad na glavnu stranicu](#)

Prijavljeni ste kao: milan (odjavi se)

Slika 110: SK Kreiranje broja novog dokumenta - Prijem i čitanje poruke

Korisnički interfejs za slučaj korišćenja - *Slanje dokumenta u arhivu*

Slika 111: Sekvencni dijagram za SK - Slanje dokumenta u arhivu



Slika 112: SK Slanje dokumenta u arhivu - Izbor dokumenta na disku

The screenshot shows the ArchiDoc Web Document Management System interface. The main content area is titled "Unesite osnovne informacije o dokumentu" (Enter basic information about the document). The form is for a document named "Lunar_impacts_Nemesis.pdf".

Fields in the form include:

- Naziv dokumenta*:** Lunar_impacts_Nemesis
- Broj dokumenta:** (empty text box)
- Napomena:** (empty text box)
- Vrsta dokumenta*:** Genericki (dropdown menu)
- Revizija dokumenta:** Izaberi (dropdown menu)
- Verzija dokumenta*:** 1.0 (dropdown menu)
- Poreklo*:** Unutrasnja (dropdown menu)
- Status dokumenta*:** Urađen (dropdown menu)

At the bottom of the form, there is a note: "Polja označena * su obavezna ::" followed by a "Pošalji dokument!" button.

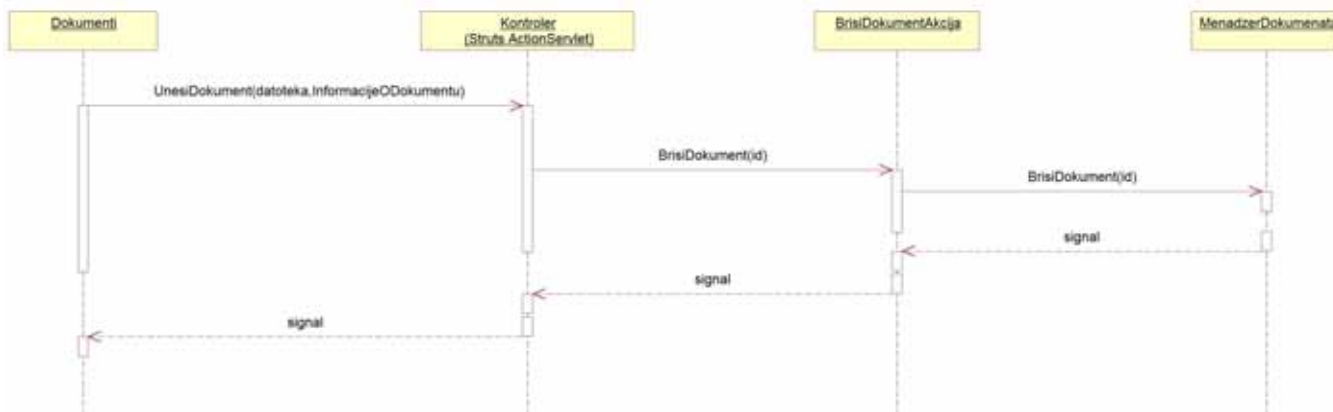
The left sidebar contains navigation options: Glavna, Arhiviranje, Administracija korisnika, Slanje dokumenata, Objekti, Grupe dokumenata, Dokumenti, and Administracija sistema.

The top navigation bar includes "Glavna", "Info", and "Pomoć". On the right, there is a search section: "Brza pretraga dokumenata:" with a search box and a "Traži!" button, and "Poslednja pretraga" and "Proširena pretraga" options.

At the bottom of the page, it says "Prijavljeni ste kao: milan (odjavi se)".

Slika 113: SK Slanje dokumenta u arhivu - Unos informacija o dokumentu

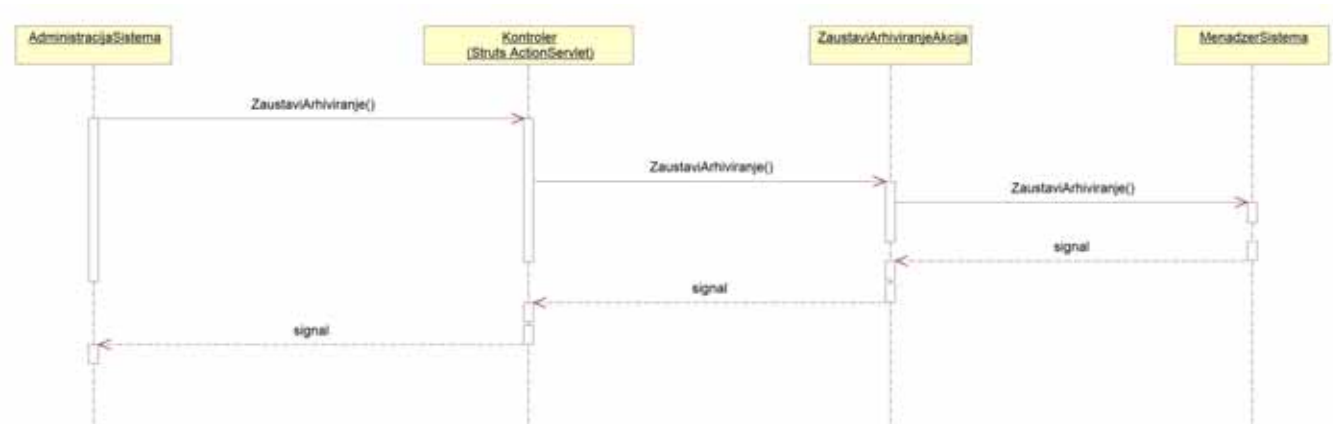
Korisnički interfejs za slučaj korišćenja - *Brisanje dokumenta*



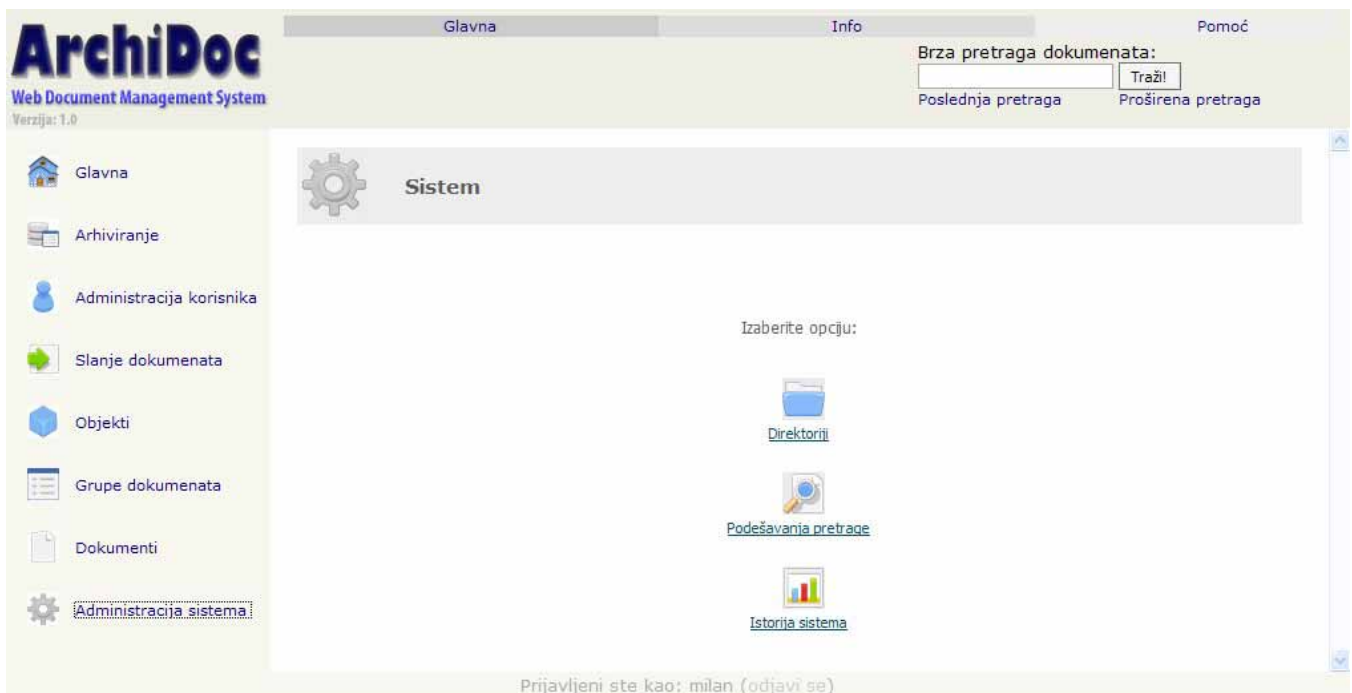
Slika 114: Sekvencni dijagram za SK - Brisanje dokumenta

Brisanje dokumenta se vrši na SK Pregled dokumenta izborom opcije "Briši dokument", što je i prikazano na slici 93.

Korisnički interfejs za slučaj korišćenja - *Zaustavljanje arhiviranja*



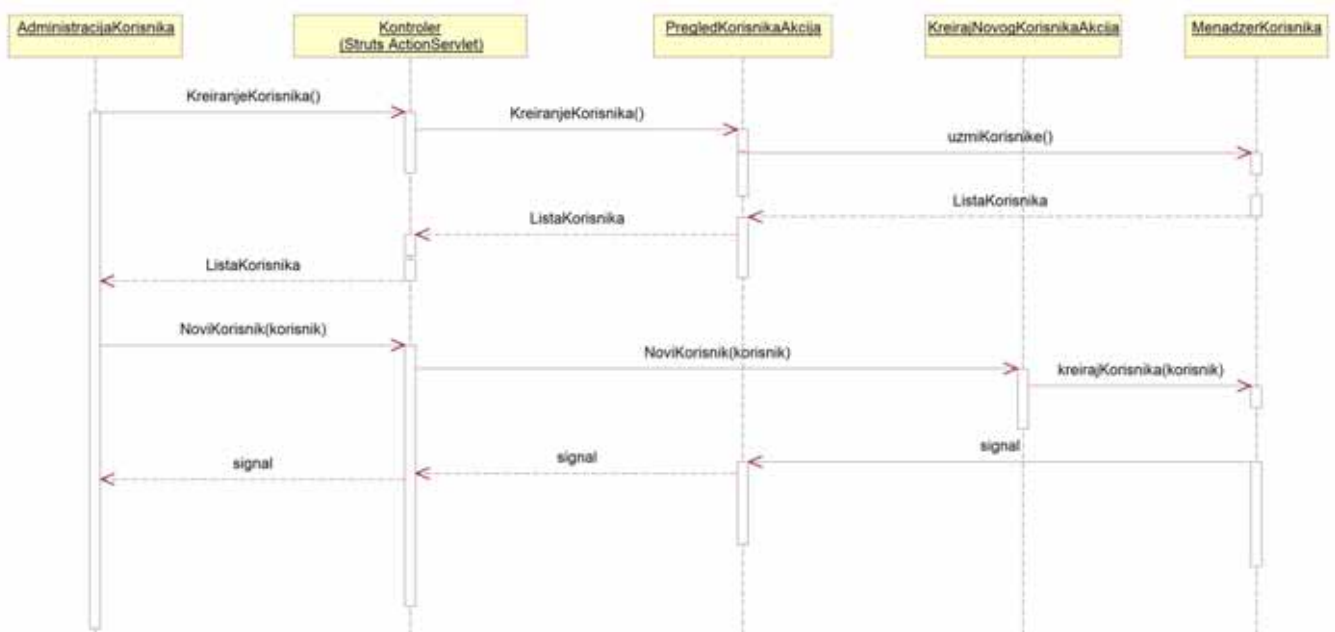
Slika 115: Sekvenčni dijagram za SK - Zaustavljanje arhiviranja



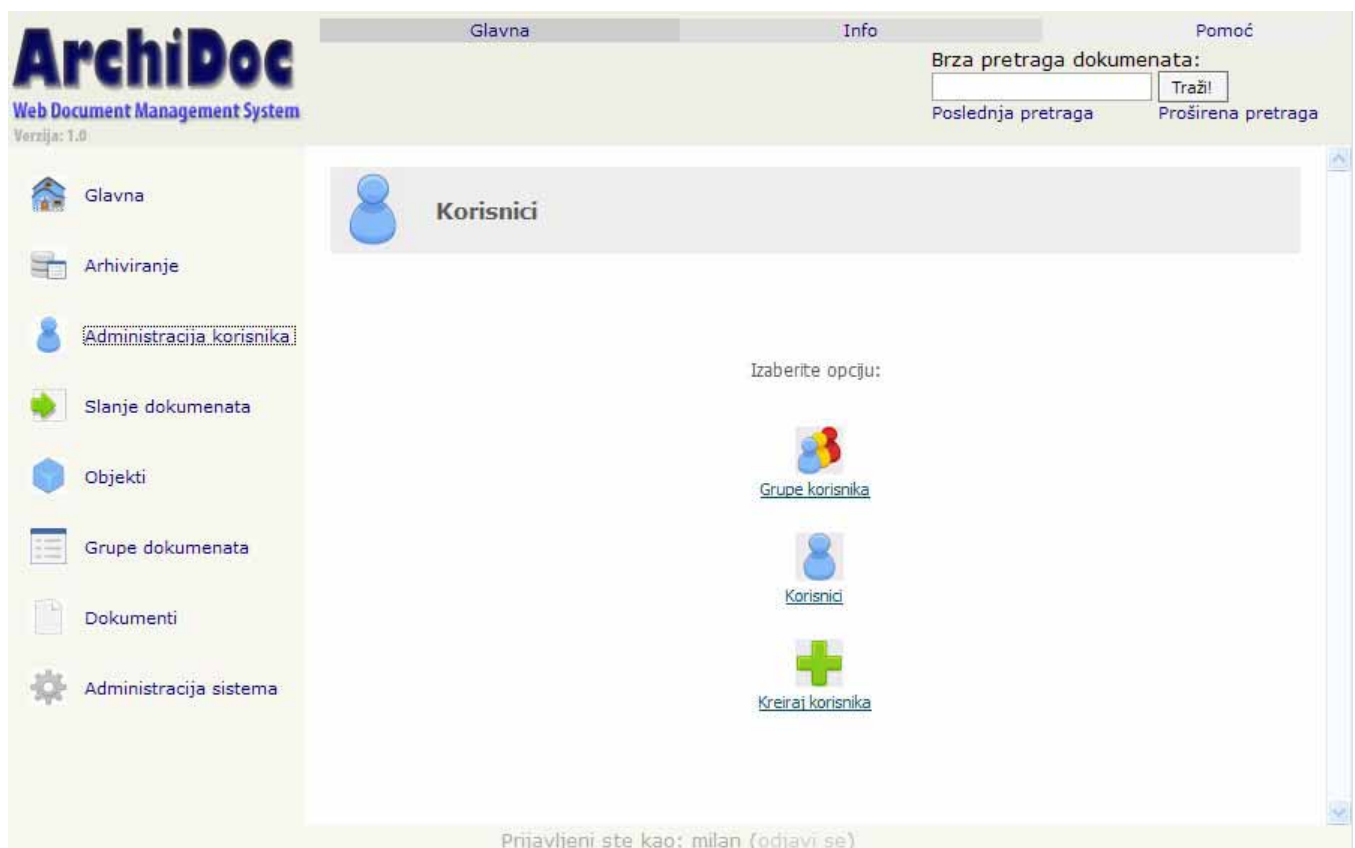
Slika 116: SK Zaustavljanje arhiviranja - Sistem

Zaustavljanje arhiviranja se vrši izborom opcije Direktorji, pa "Zaustavi arhiviranje".

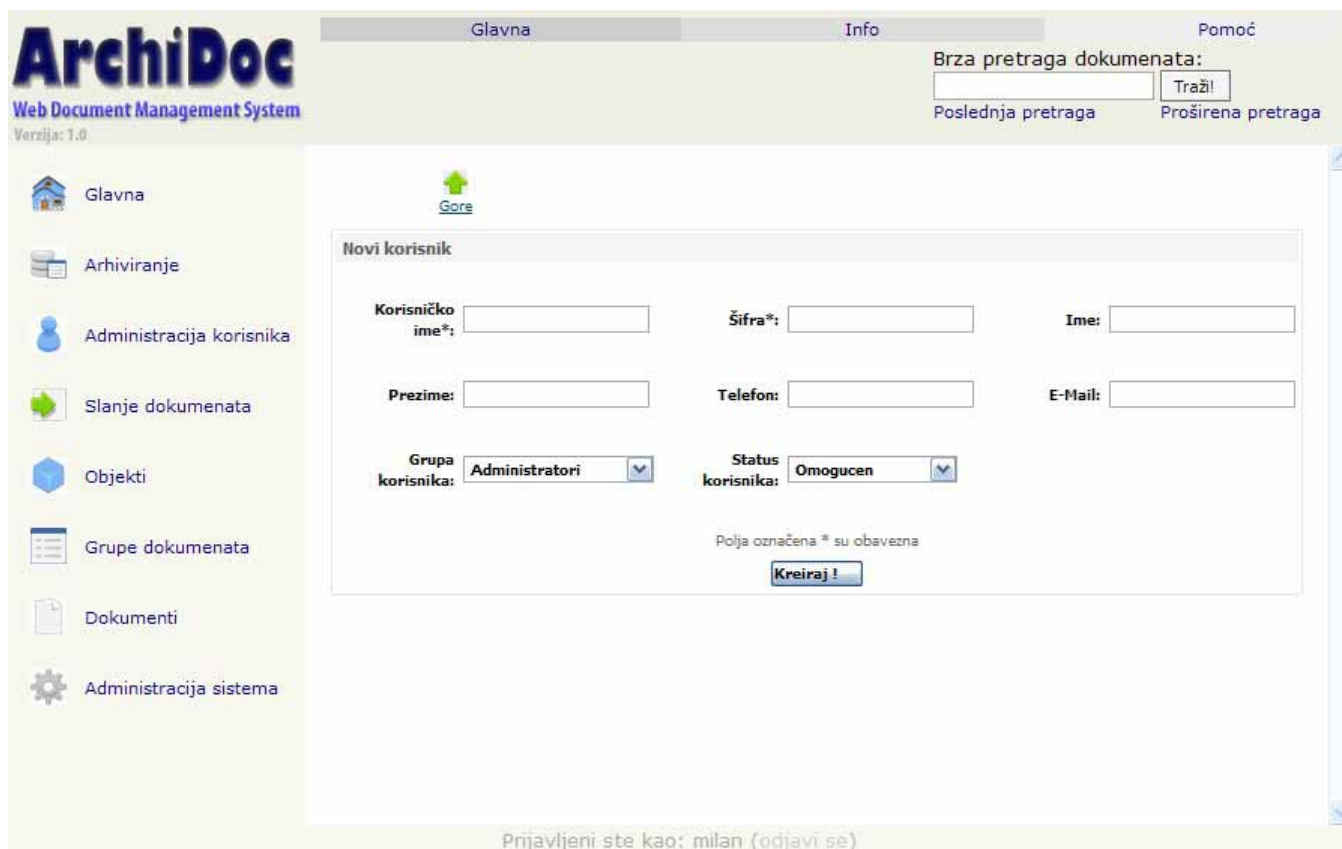
Korisnički interfejs za slučaj korišćenja - *Administracija korisnika*



Slika 117: Sekvencni dijagram za SK - Administracija korisnika

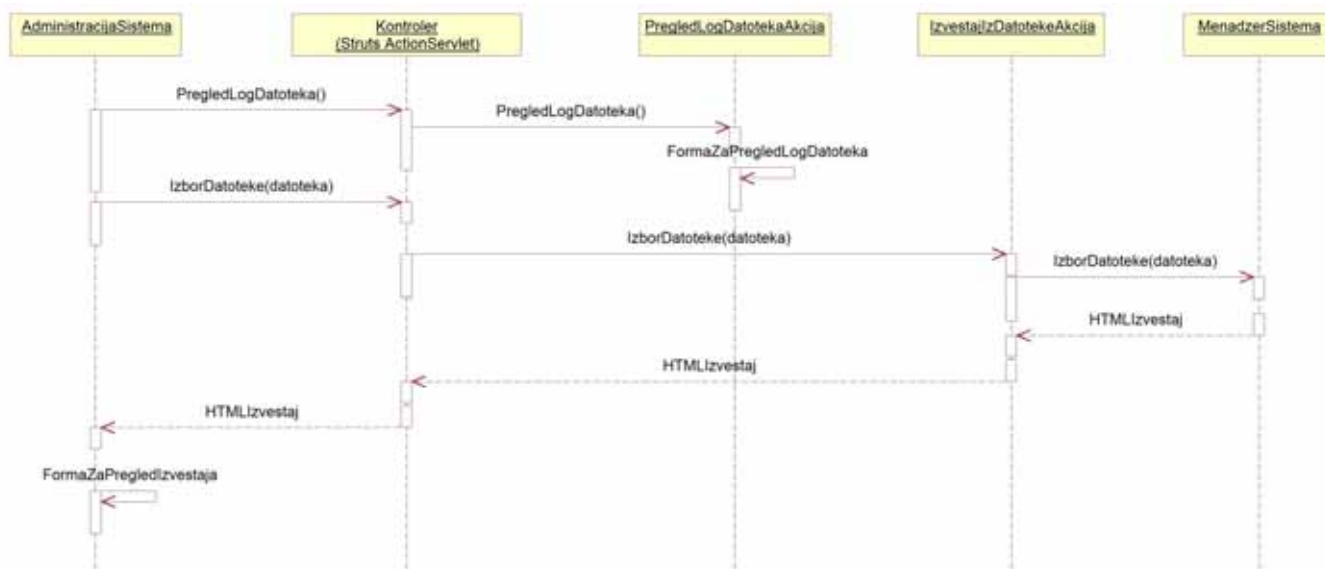


Slika 118: SK Administracija korisnika - Izbor opcije



Slika 119: SK Administracija korisnika - Kreiranje korisnika

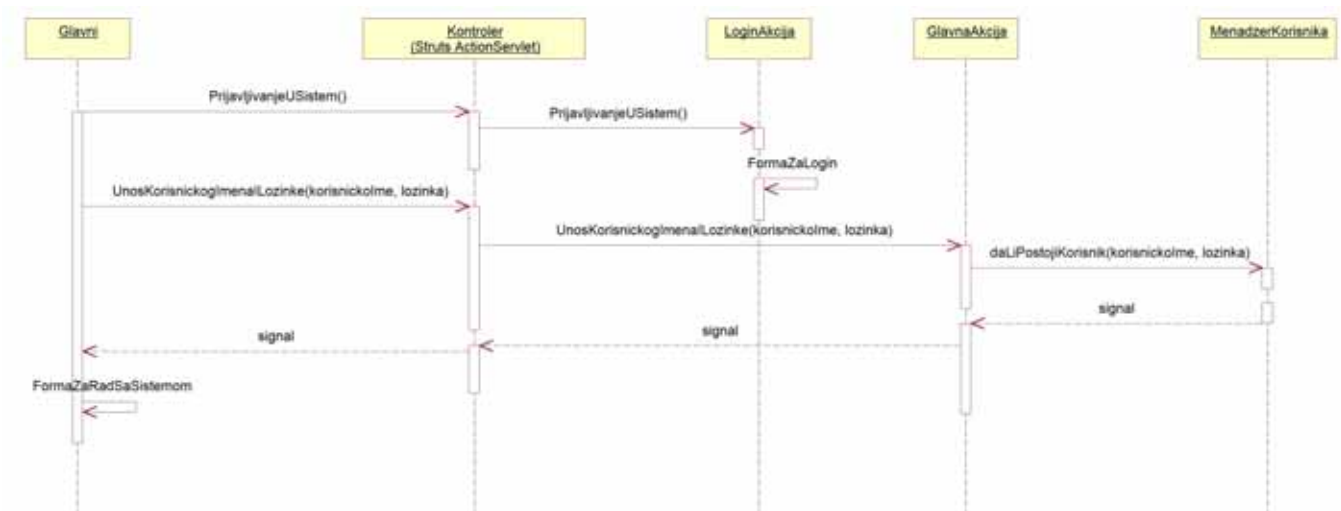
Korisnički interfejs za slučaj korišćenja - *Pregled log datoteka*



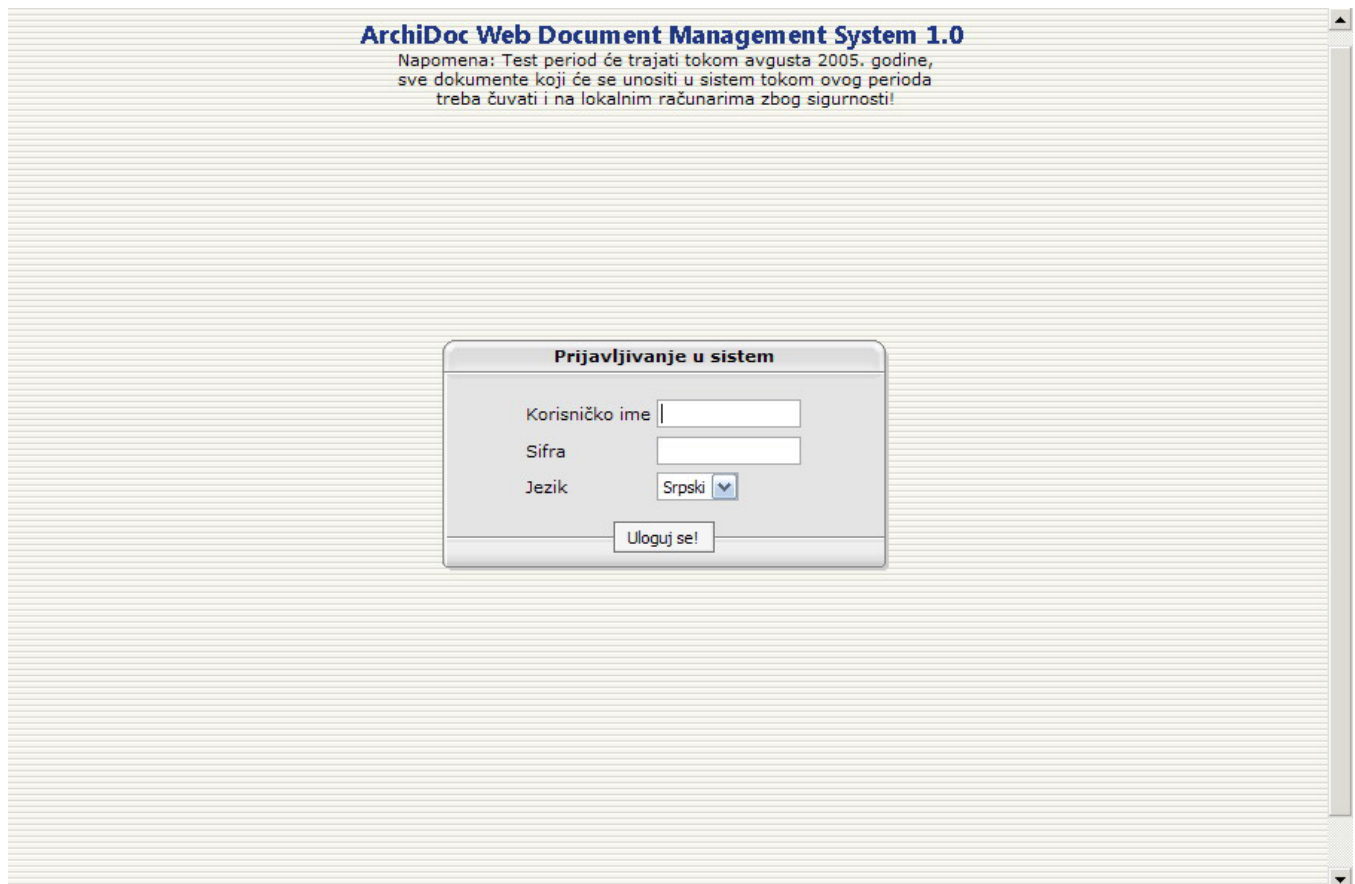
Slika 120: Sekvencni dijagram za SK - Pregled log datoteka

Pregled log datoteka se vrši izborom na opciju "Istorija sistema" u delu Administracija sistema, što se može videti na slici 116, kod SK Zaustavljanje arhiviranja.

Korisnički interfejs za slučaj korišćenja - *Prijava u sistem*

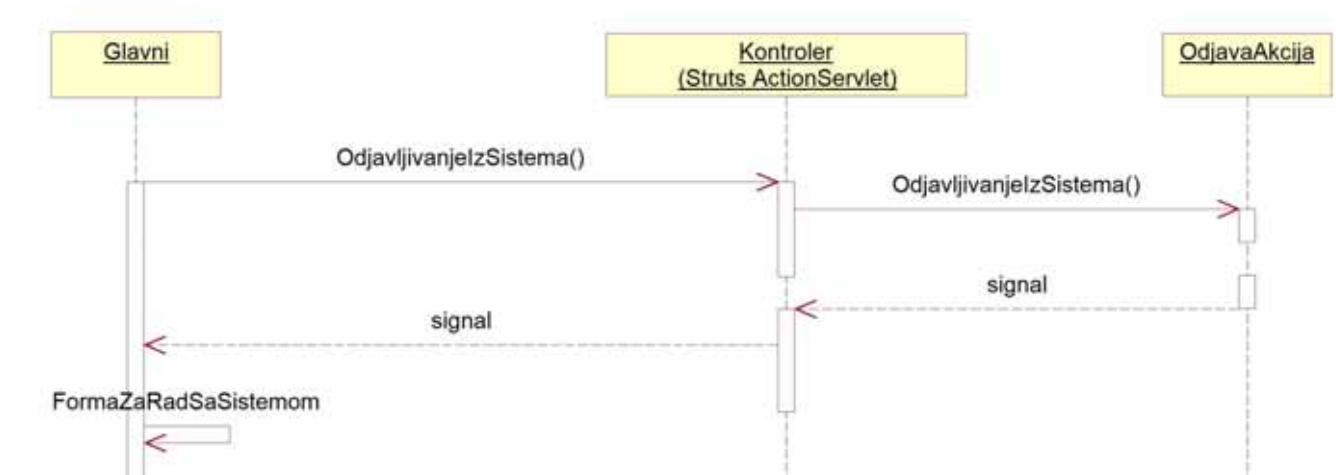


Slika 121: Sekvenčni dijagram za SK - Prijava u sistem



Slika 122: SK Prijava u sistem - Prijavljivanje u sistem

Korisnički interfejs za slučaj korišćenja - *Odjava iz sistema*

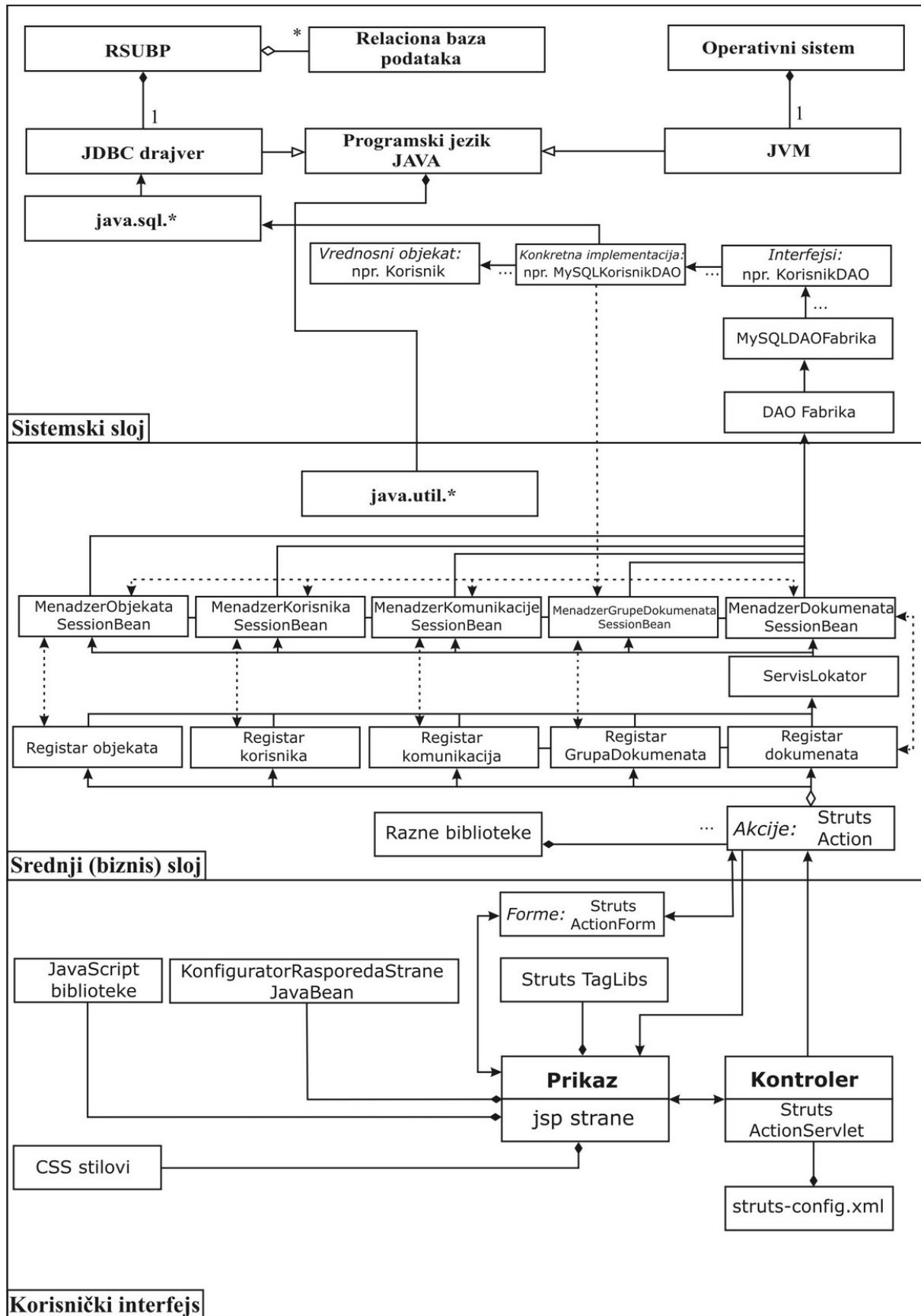


Slika 123: Sekvencni dijagram za SK - Odjava iz sistema

Odjavu iz sistema je moguće pokrenuti sa bilo kojeg ekrana klikom na "odjavi se" na dnu ekrana.

2.3.4. Opis arhitekture iz perspektive modela projektovanja

Rezultat kompletnog razvoja informacionog (softverskog) sistema je predstavljen arhitekturom softverskog sistema na slici ispod.



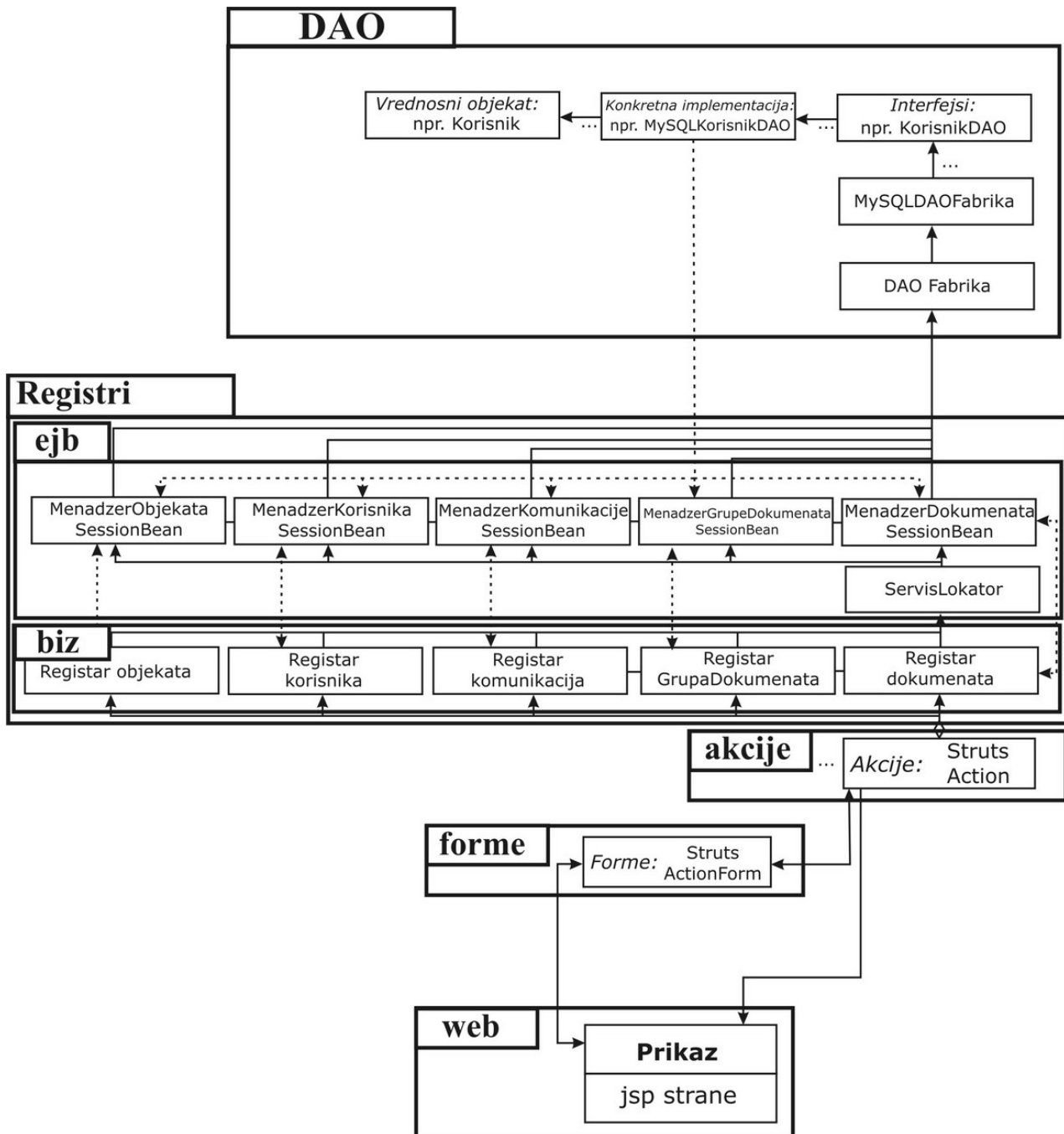
Slika 124: Arhitektura softverskog sistema

Arhitektura softverskog sistema je proširila okruženje sistema, tako što je:

- Povezala pojedine akcije sa prikazom jsp strana preko Struts-ovog kontrolera;
- Uredila način korišćenja određenih registara preko akcija;
- Spojila Srednji sloj (Session Bean-ove) sa bazom podataka koristeći DAO uzor;
- Uvela dodatne JavaBean-ove.

Aplikacioni podsistemi projektovanja

Iz arhitekture sistema se identifikuju sledeći podsistemi:



Slika 125: Aplikacioni podsistemi projektovanja

1. DAO:

- Ima ulogu da se spoji sa bazom podataka;
- Izvršava operacije na bazom;
- Da popuni vrednosne objekte iz baze;
- I da ih prenese pomoću Transfer objekata do srednjeg sloja.

2. Registri (biz i ejb):

- Realizuju Fasada uzor;
- Realizuju pojedine systemske operacije;
- Povezuje se sa DAO slojem i preuzima podatke od njega.

3. Akcije:

- Realizuje neke systemske operacije i sve ostale akcije vezane za sistem;
- Predstavljaju vezu između web strana i srednjeg sloja (Registara);
- Kreiraju i razmenjuju podatke sa registrima.

4. Forme:

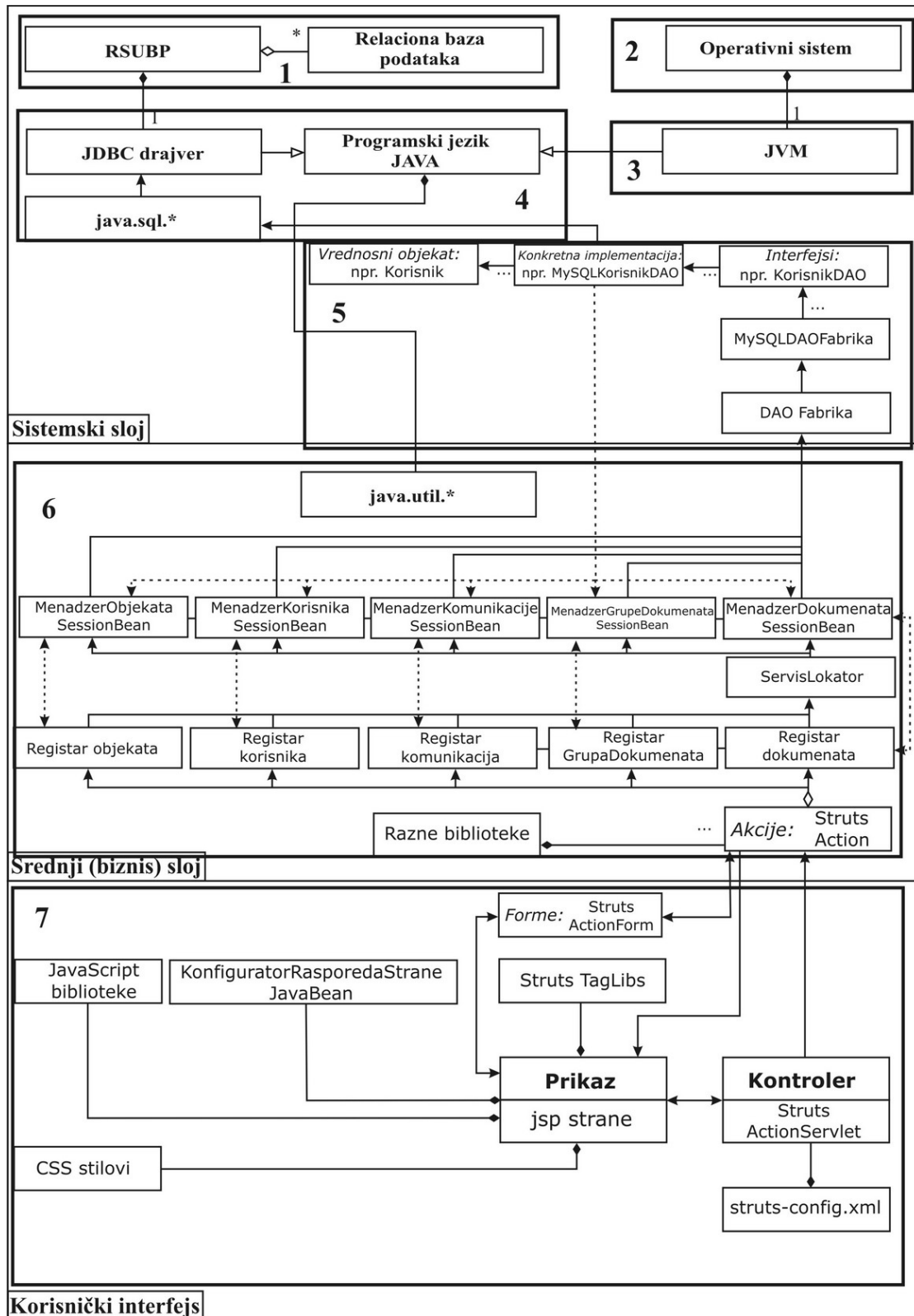
- Sadrži podatke koji se prenose od web strana do akcija i obratno.

5. Web:

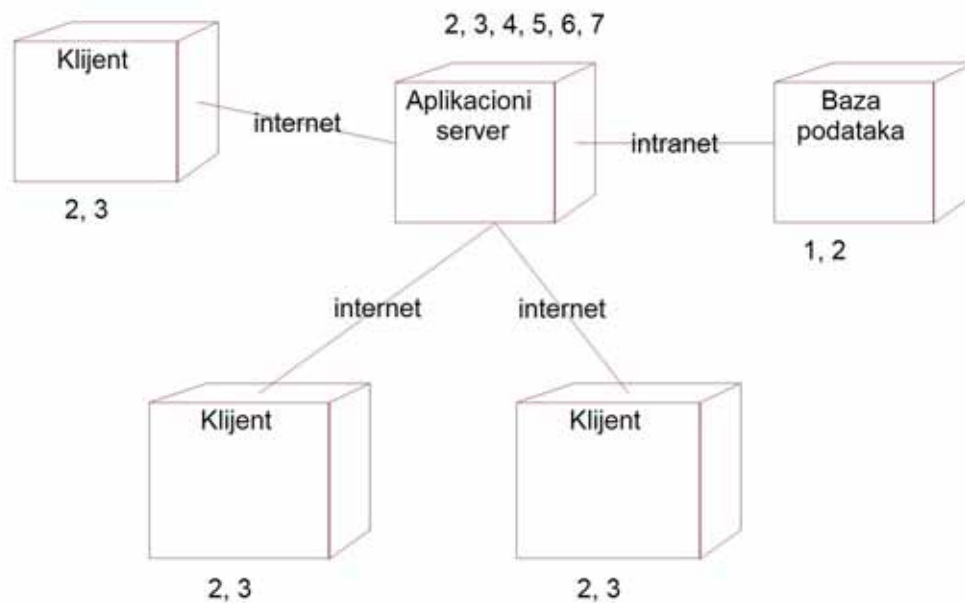
- Predstavljaju web stranice u obliku jsp stranica;
- Prihvataju podatke od korisnika i prikazuju podatke iz akcija.

2.3.5. Modeli raspoređivanja

Model raspoređivanja obuhvata kompletnu arhitekturu softverskog sistema. Ovde se obraća pažnja na sve podsisteme. Ovi podsistemi se dodeljuju do čvorova modela raspoređivanja.



Slika 126: Model raspoređivanja podsistema



Slika 127: Model raspoređivanja

2.4. IMPLEMENTACIJA

Kod implementacije se prikazuju komponente koje se proizašle iz arhitekture softverskog sistema, kao i redosled njihovog izvršavanja i međusobna zavisnost.

2.4.1. Komponente

Komponente se dobijaju iz klasa i interfejsa ili podsistema aplikacionih slojeva. Samo komponente su raspoređene po paketima koji su prikazani u modelu raspoređivanja i datotekama koje su navedene ranije u ovom radu.

2.4.2. Redosled kompajliranja komponenti

Sve komponente (koje su razvijene u NetBeans-u 4.1) su spakovane kao jedinstvena aplikacija u datoteku archidoc.ear (Enterprise archive). U njoj se nalaze dva osnovna modula: ArchiDoc-EJBModule.jar i ArchiDoc-WebModule.war (Web archive). U prvom su smeštene DAO klase i klase srednjeg sloja (registri), a u drugom web (jsp) stranice. Takođe u ovoj datoteci se nalaze i sve potrebne (.jar) biblioteke i application.xml, kao i sun-application.xml za podešavanje Sun-ovog aplikacionog servera. Tu su i datoteke struts web framework-a. Redosled kompajliranja komponenti je naveden u datoteci build.xml i za to se koristi Ant. Tu je navedeno da se kompajliraju prvo biblioteke, pa potom EJB paket (koji je priključen web paketu kao komponenta), pa potom WEB paket.

2.4.3. Instalacija sistema i pokretanje

Za instalaciju su urađene posebne skripte i uputstvo za instalaciju. Posle instalacije web servera, servera baze podataka i aplikacionog servera, radi se prenos aplikacije (deploy) na aplikacioni server. Pokretanje instalacije se vrši unosom adrese u web čitač: <http://localhost:8080/ArchiDoc/setup>. Proces instalacije se kreće kroz više formi koje dopuštaju korisniku izbor osnovnih opcija za instalaciju sistema

i kreiranja osnovnog administratorskog naloga. Pokretanje aplikacije se vrši na kraju instalacije pomoću linka koji je aplikacija ponudila.

2.4.4. Veličina sistema

Veličina ArchiDoc Web Document Management sistema je opisana na sledeći način:

- Meseci rada: 8.
- Projektanata: 1.
- Biznis komponenti: 6.
- Zavisnost od eksternih komponenti: 7.
- Linija Java kôda: EJB (21.694), JSP (7.947), WEB (17.361), Ukupno: 47.002.

2.5. TESTIRANJE

Testiranje je obavljeno detaljno, i to:

1. Sve DAO komponente (njihove metode) su detaljno testirane pomoću Junit biblioteke, tako što su, npr. za klasu MySQLDokumentDAO kreirane klase MySQLDokumentDAOTest koje implementiraju TestCase klasu i omogućavaju testiranje celog sistema samo jednim klikom.
2. Kod testiranja DAO komponenti su korišćeni assertion-si, iz JDK 1.5.0.
3. Detaljno su testirani svi slučajevi korišćenja sistema, od automatskog pokretanja aplikacionog servera, do prihvatanja zahteva od klijenata za izvršavanje sistemskih operacija.
4. Ceo sistem je testiran pod uticajem više korisnika (od 25-27), gde je ograničenje sistema 30 korisnika.

3. ZAKLJUČAK

ArchiDoc Web Document Management sistem, kao informacioni sistem za rad sa dokumentacijom u elektronskom obliku je projektovan u cilju da obuhvati ceo proces arhiviranja dokumentacije u jednom projektnom birou. Ovaj sistem prati proces arhiviranje po zvaničnom uputstvu projektnog biroa. Zahvaljujući primenjenoj arhitekturi, prikazani softverski sistem se vrlo lako održava, proširuje i što je najvažnije, jednostavno integriše u sisteme šire namene. Ovde se vidi sva snaga objektnog pristupa programiranju koja je tek došla do izražaja kod sistema ovako većeg obima. Kod projektovanja sistema je primenjen jedinstveni proces razvoja softvera koji je omogućio da se na jednostavan način implementira kompleksan sistem i da se poveća robustnost sistema zbog kasnijih proširivanja i održavanja. Za sistem je važno da se zaštita (bekap) svih podataka može izvesti na jednostavan način putem bilo kojeg bekap sistema.

4. LITERATURA

- [Alur2000] Alur, D., Malks, D. (2000) *Core J2EE Patterns*, Sun Microsystems, Inc.
- [Boggs2002] Boggs, W., Boggs, M. (2002) *UML i Rational Rose 2002*, Kompjuter biblioteka, Čačak.
- [Brogden2001] Brogden, B., Minnick, C. (2001) *Java i razvoj aplikacija elektronskog poslovanja pomoću XML i JSP*, Kompjuter biblioteka, Čačak.
- [Cattell2001] Catell, R., Inscore, J. (2001) *J2EE Technology in Practice: Building Business Applications With the Java 2 Platform, Enterprise Edition*, Pearson Education.
- [Cooper1998] Cooper, J. (1998) *The Design Patterns: Java Companion*, Addison Wesley.
- [Coplien2000] Coplien, O., J. (2000) *Software Patterns*, Bell Labs, The Hillside Group.
- [Crawford2003] Crawford, W., Kaplan, J. (2003) *J2EE Design Patterns*, O'Reilly.
- [DeMichiel2003] DeMichiel, G., L. (2003) *Enterprise JavaBean Specification, Version 2.1*, Sun Microsystems, Inc.
- [Gamma1995] Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1995) *Gotova rešenja: elementi objektno orijentisanog softvera za višekratnu upotrebu*, CET, Beograd.
- [Heller2005] Heller, P., Roberts, S. (2005) *Complete Java 2 Certification: Study Guide*, Sybex, Inc., 5th Edition.
- [Hubert2002] Hubert, R. (2002) *Convergent Architecture - Building Model-Driven J2EE Systems with UML*, John Wiley & Sons.
- [Jacobson1999] Jacobson, I., Booch, G., Rumbaugh, J. (1999) *The Unified Software Development Process*, Rational Software Corporation, Addison-Wesley.
- [Jones2005] Jones, A., Stephens, K., R., Plew, R., R., Garrett, F., R., Kriegel, A. (2005) *SQL Functions Programmer's Reference*, Wiley Publishing, Inc.
- [Kurniawan2002] Kurniawan, B. (2002) *Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions*, New Riders Publishing.
- [Larman2000] Larman, C. (2000) *Applying UML and Patterns*, Prentice Hall, 2nd Edition.
- [Lazarević2004] Lazarević, B., Marijanović, Z., Aničić, N., Babarogić, S. (2004) *Baze podataka*, Fakultet organizacionih nauka, Beograd.
- [Lemay1998] Lemay, L., Cadenhead, R. (1998) *Naučite za 21 dan - java 1.2*, Kompjuter biblioteka, Čačak.
- [Marinescu2002] Marinescu, F. (2002) *EJB Design Patterns - Advanced Patterns, Processes and Idioms*, John Wiley & Sons.

- [Milićev2001] Milićev, D. (2001) *Objektno orijentisano modelovanje na jeziku UML*, Mikro knjiga, Beograd.
- [Monson-Haefel2001] Monson-Haefel, R. (2001) *Enterprise JavaBeans*, O'Reilly.
- [Pekowsky2003] Pekowsky, L. (2003) *JavaServer Pages*, Addison Wesley, 2nd Edition.
- [Reese2000] Reese, G. (2000) *Database Programming with JDBC and Java*, O'Reilly, 2nd Edition.
- [Reese2001] Reese, G., Yarger, J., R., King, T. (2001) *Managing & Using MySQL*, O'Reilly.
- [Riordan2005] Riordan, R. (2005) *Designing Effective Database Systems*, Addison Wesley.
- [Roman2005] Roman, E. (2005) *Mastering Enterprise JavaBeans*, Wiley Computer Publishing.
- [Rosenberg2001] Rosenberg, D., Scott, K. (2001) *Applying Use Case Driven Object Modeling with UML: An Annotated e-Commerce Example*, Addison Wesley.
- [Schildt2001] Schildt, H. (2001) *Kompletan priručnik, JAVA 2*, Mikro knjiga, Beograd.
- [Shannon2003] Shannon, B. (2003) *Java 2 Platform Enterprise Edition Specification*, v1.4, Sun Microsystems, Inc.
- [Shenoy2005] Shenoy, S. (2005) *Struts Survival Guide*, ObjectSource LLC.
- [Simsion2005] Simsion, G., Witt, G. (2005) *Data Modeling Essentials*, Morgan Kaufmann.
- [Singh2002] Singh, I., Stearns, B., Johnson, M. (2002) *Designing Enterprise Applications with the J2EE Platform*, Addison Wesley.
- [Tindall2002] Tindall, P. (2002) *Developing Enterprise Applications*, Que.
- [Zakas2005] Zakas, N. (2005) *Professional JavaScript for Web Developers*, Wiley Publishing Inc.
- [Zukowski2001] Zukowski, J. (2001) *Java Collections*, Apress.
- [Vlajić2004] Vlajić, S. (2004) *Projektovanje programa*, skripta, Univerzitet u Beogradu, Fakultet organizacionih nauka.
- [Vlajić2003] Vlajić, S., Ćirić, V., Savić, D. (2003) *Projektovanje programa*, praktikum, Univerzitet u Beogradu, Fakultet organizacionih nauka.