

Данијела Тешендић
Душан Сурла

Коришћење библиотечке грађе у софтверском систему БИСИС



**Иновациони центар за
електронске библиотеке и архиве**

Департман за математику и
информатику

Природно – математички факултет

Универзитет у Новом Саду



Данијела Тешендић

Душан Сурла

Коришћење библиотечке грађе у софтверском систему БИСИС

Нови Сад, 2007.

Назив монографије

Коришћење библиотечке грађе у софтверском систему БИСИС

Аутори

мр Данијела Тешендић, истраживач приправник, Иновациони центар за електронске библиотеке и архиве, Департман за математику и информатику, Природно-математички факултет, Нови Сад

др Душан Сурла, редовни професор, Иновациони центар за електронске библиотеке и архиве, Департман за математику и информатику, Природно-математички факултет, Нови Сад

Рецензенти

др Милош Рацковић, редовни професор, Природно-математички факултет, Нови Сад

др Зора Коњовић, редовни професор, Факултет техничких наука, Нови Сад

др Бранко Милосављевић, доцент, Факултет техничких наука, Нови Сад

Издавач

Природно-математички факултет, Нови Сад

Главни и одговорни уредник

Проф. др Мирослав Весковић, декан Природно-математичког факултета у Новом Саду

Нучно-наставно веће Природно-математичког факултета у Новом Саду, на наставку XII седнице од 21. 11. 2007. године, одобрило је штампање и употребу ове монографије.

Издавач задржава сва права. Без писмене сагласности издавача није дозвољено прештампавање нити преузимање односно репродуковање књиге нити њених појединих делова.

Корице

Данијела Тешендић

Штампа

АМБ Економик

Тираж

150

Предговор

У овој монографији приказан је део добијених резултата на пројекту *Апстрактни модели и примене у рачунарским наукама*, број 144017А, који финансира Министарство науке и заштите животне средине Републике Србије. Носилац истраживања овог пројекта је Природно-математички факултет у Новом Саду.

Предмет истраживања у овој монографији је моделирање и имплементација система за коришћење библиотечке грађе. Софтверска архитектура система треба да задовољи све информационе захтеве електронског пословања градских библиотека. То се посебно односи на рад како у интранет окружењу тако и на Интернету.

Монографија садржи следећа поглавља:

1. Увод
2. Спецификација информационих захтева система за коришћење библиотечке грађе
3. Имплементација система за коришћење библиотечке грађе
4. Опис коришћења апликације

У **првом** поглављу су описана нека од постојећих софтверских решења за позајмицу библиографске грађе и преглед развоја библиотечког софтверског система БИСИС.

Централни део монографије је спецификација информационих захтева система за коришћење библиотечке грађе, која је дата у **другом** поглављу. За моделирање је коришћен обједињени језик моделирања UML 2.0. Спецификација садржи дијаграме случајева коришћења са текстуалним описом, дијаграме класа и дијаграме

секвенци. У овом поглављу је дат и дијаграм класа базе података за коришћење библиотечке грађе.

Имплементација система за коришћење библиотечке грађе описана је у **трећем** поглављу. Систем поседује два интерфејса за комуникацију корисничке апликације са базом података, чиме је омогућен ефикасан рад апликације и у случају кад се база података налази на локалном серверу и у случају кад се налази на удаљеном серверу. Овако предложена софтверска архитектура може се користити и за друге информационе системе.

У **четвртном** поглављу описано је коришћење и администрација апликације од стране библиотекара.

Захваљујемо се рецензентима на корисним примедбама и сугестијама које су допринеле квалитету монографије.

Нови Сад, 2007.

Аутори

Садржај

Предговор	3
Садржај	5
Сажетак	7
Abstract	9
Увод	11
1.1 Преглед постојећих софтверских система за коришћење библиотеке грађе	11
1.2 Библиотечки софтверски систем БИСИС	15
1.2.1 Функције система БИСИС	16
1.2.2 Примена XML технологије у развоју система БИСИС	17
1.3 Коришћење библиографске грађе у систему БИСИС	18
Спецификација информационих захтева система за коришћење библиотеке грађе	21
2.1 Диаграми случајева коришћења	21
2.1.1 Дијаграм случајева коришћења <i>Рад са корисницима</i>	22
2.1.2 Дијаграм случајева коришћења <i>Извештавање</i>	32
2.1.3 Дијаграм случајева коришћења <i>Параметри система</i> ...	38
2.2 Статичка структура система	42
2.2.1 Дијаграм пакета система	42
2.2.2 Класе пакета <i>View</i>	43
2.2.3 Класе пакета <i>Model</i>	45
2.2.4. Пакет <i>Manager</i>	49
2.2.5 Пакет <i>Proxy</i>	50
2.2.6 Пакет <i>RecordUtils</i>	51
2.3. Дијаграми секвенци	53
2.3.1 Дијаграм секвенци <i>Апликација</i>	53
2.3.2 Дијаграм секвенци <i>NoviKorisnik</i>	55
2.3.3 Дијаграм секвенци <i>IndividualniKorisnik</i>	56
2.3.4 Дијаграм секвенци <i>IndividualniKorisnikUpis</i>	57
2.3.5 Дијаграм секвенци <i>IndividualniKorisnikAžuriranje</i>	59
2.3.6 Дијаграм секвенци <i>KolektivniKorisnik</i>	60
2.3.7 Дијаграм секвенци <i>PostojećiKorisnik</i>	62

Имплементација система за коришћење библиотечке грађе	65
3.1 Софтверско окружење	66
3.2 Имплементација корисничког интерфејса	67
3.2 Имплементација ажурирања базе података	72
3.2.1 Имплементација објектног модела базе података	73
3.2.2 Имплементација комуникације са базом података	73
3.2.2.1 Мапирање базе података	74
3.2.2.2 Приступ бази података у локалном интранету	74
3.2.2.3 Приступ бази података преко Интернета	77
3.2.3 Имплементација менаџера	79
3.3 Имплементација рада са библиотечким записима	81
3.3.1 Имплементација интерфејса библиотечког записа	82
3.3.2 Имплементација менаџера записа	84
Опис коришћења апликације	87
4.1 Основна екранска форма	87
4.2 Подаци индивидуалних корисника	90
4.2.1 Упис новог корисника	90
4.2.2 Ажурирање података постојећег корисника	91
4.2.3 Картица <i>Osnovni podaci</i>	93
4.2.4 Картица <i>Dodatni podaci</i>	94
4.2.5 Картица <i>Članarina</i>	95
4.2.6 Картица <i>Zaduženja</i>	97
4.3 Подаци колективних корисника	98
4.3.1 Упис новог корисника	99
4.3.2 Ажурирање података постојећег корисника	100
4.4 Претраживање корисника	100
4.5 Претраживање публикација	103
4.6 Генерисање извештаја	107
4.7 Генерисање опомена	109
4.7.1 Измена текста опомене	112
4.7.2 Регистар опомена	114
4.7.3 Раздуживање корисника којима је послата опомена	115
4.8 Администрација система	117
4.8.1 Администрација шифарника	117
4.8.2 Параметри система	119
Литература	125

Сажетак

Циљ. Циљ истраживања је моделирање и имплементација система за коришћење библиотечке грађе. Моделирање информационих захтева овог система урађено је у обједињеном језику моделирања (UML 2.0). Имплементација система реализована је у програмском језику *Java*.

Методологија. Коришћена је објектно оријентисана методологија за пројектовање софтверских компоненти као и CASE алати. Архитектура софтверског система подржава рад са различитим библиотечким стандардима и може се интегрисати у различите библиотечке софтверске системе.

Резултат истраживања. Резултат је систем за коришћење библиотечке грађе. Апликација система подржава рад са локалном базом података (интранет), као и са удаљеном базом података (Интернет). Поред тога апликација подржава рад са различитим библиотечким форматима (*UNIMARC, MARC21*, као и други).

Ограничење резултата истраживања. Интеграција овог система у систем БИСИС реализована је тако да се комуникација са системом за обраду библиотечке грађе одвија преко текст сервера система БИСИС. Због тога би интеграција у друге библиотечке софтверске системе захтевала додатно програмирање. Превазилажење овог ограничења може се реализовати тако што би се дефинисао интерфјес за комуникацију са системом за обраду библиотечке грађе. У овом случају интеграција у неки систем би захтевала само имплементацију тог интерфејса од стране другог библиотечког система.

Практична примена. Систем за коришћење библиотечке грађе интегрисан је у четврту верзију система БИСИС. Тестирање и верификација је извршена над подацима факултетских и градских библиотека. Систем БИСИС вер. 3 ради у 36 библиотека (факултетске, јавне и специјализоване). У току је прелазак ових библиотека на четврту верзију система.

Оригиналност резултата истраживања. Оригиналност рада је у архитектури система која се може конфигурисати за рад у локалном интранету, као и на Интернету и не зависи од формата библиотечких записа. Такође, архитектура омогућава интеграцију апликације у различите библиотечке системе.

Abstract

Purpose. The purpose of the research is modelling and implementation of library material usage system. Information requirements of the system are modelled in Unified Modelling Language (UML 2.0). System implementation is realised in Java programming language.

Methodology. The object-oriented methodology as well as CASE tools is used to design the software components. The software architecture supports work with different bibliographic standards and can be integrated into various library software systems.

Findings. The outcome is library material usage system. An application of the system supports work with local database (intranet) as well as work with remote database (Internet). Besides, the application supports work with different bibliographic formats (*UNIMARC*, *MARC21*, and others)

Research limitations. Integration of the application into system *BISIS* is realised in the way that communication with system for processing bibliographic material depends on text server of system *BISIS*. Because of that, integration into other library software systems requires additional programming. This limitation can be overcome by introducing interface for communication with system for bibliographic material processing. In that way, the only requirement for integrating the application into other library systems would be implementation of this interface by that system.

Practical implications. The system for library material usage is integrated into forth version of system *BISIS*. Its testing and verification were done by using data from faculty libraries and public city libraries.

The system BISIS, version 3 is in use in 36 libraries (faculty, public and specialised). The installation of forth version of system BISIS is in progress.

The originality of research findings. The originality lies in system architecture which can be configured for work in local intranet as well as on Internet and it is independent of bibliographic records format. Also, architecture enables integration of the application into various librarian software systems.

Поглавље 1

Увод

Резултати приказани у овој монографији добијени су у оквиру развоја библиотечког софтверског система БИСИС, и то оног дела који се односи на коришћење библиотечке грађе. Сегмент који се односи на коришћење библиотечке грађе, односно праћење коришћења фонда библиотеке представља посебан део библиотечког пословања и у оквиру библиотечког информационог система БИСИС реализован је апликацијом Циркулација.

Библиотечки софтверски систем БИСИС развија се од 1993. године и преглед тог развоја дат је у другом одељку овог поглавља. Актуелна верзија овог система је 3.1. Увођењем ове верзије система Градске библиотеке уочени су недостаци постојеће имплементације апликације Циркулација у односу на захтеве које градске библиотеке имају у свом пословању. На основу анализе захтева и потреба градских библиотека које користе БИСИС, у овој монографији дат је опис моделирања и имплементације апликације система за коришћење библиотечке грађе тако да задовољи потребе електронског пословања градских библиотека.

1.1 Преглед постојећих софтверских система за коришћење библиотечке грађе

Постоји велики број библиотечких информационих система који у свом оквиру имају и систем за коришћење библиотечке грађе. Овде је дат кратак опис неких од тих система.

ALEPH 500 [Aleph] је библиотечки информациони систем развијен од стране *Ex Libris Group*. Систем укључује модуле за

каталогизацију, набавку књига, циркулацију, *web* претраживање, међубиблиотечку позајмицу, извештавање и корисничке сервисе. Систем има подршку за *UNICODE* и вишејезичност екранских форми. Модул за циркулацију омогућава све активности везане за рад са корисницима које обухватају задуживање и раздуживање корисника, као и праћење финансијских трансакција везаних за чланарину корисника. Омогућена је и резервација публикација како од стране библиотекара кроз циркулацију, тако и од стране корисника кроз *web* интерфејс. Постоји део система, *offline* циркулација, која омогућава рад циркулације и у случају прекида везе између клијента и сервера. У том случају подаци се привремено чувају локално на клијету и трансферују се на сервер након успостављања везе са сервером. Систем има подрушку за *SIP2* протокол [*SIP0*] чиме је омогућена употреба бар-кода и *RFID* технологије [*RFID*]. Такође постоји део система за генерисање статистичких извештаја. *ALEPH 500* систем има вишеслојну клијент/сервер архитектуру. Клијентски модули намењени библиотекарима су имплементирани као *Windows* апликације подржане само од стране *Windows* оперативних система. Серверска страна је имплементирана за *UNIX* платформе (*IBM*, *Digital Alpha*, *Sun*, *HP* и *Linux*), а као сервер базе података се користи *Oracle*.

Voyager [*Voyager*] је библиотечки информациони систем развијен од стране *Endeavor Information Systems Inc.* који се 2006. године спојио са *Ex Libris Group*. Дизајниран је специфично за академске и истраживачке институције. Укључује модуле за каталогизацију, набавку књига, циркулацију, међубиблиотечку позајмицу, *web* претраживање и извештавање. Модул за циркулацију омогућава све активности везане за задуживање и раздуживање корисника публикацијама. Поред тога постоји и могућност повезивања различитих библиотека у конзорцијуме. У том случају корисник може да позајми или врати књигу у било којој библиотеци у оквиру конзорцијума и информацију о томе ће да има свака библиотека у конзорцијуму. Систем има подрушку за *SIP2* протокол чиме је омогућена употреба бар-кода и *RFID* технологије. Клијентска страна је развијана за *Windows* оперативне системе. Модул за извештавање користи *Microsoft Access* за генерисање извештаја. Серверска страна је имплементирана за *UNIX* платформе, а као сервер базе података се користи *Oracle*.

BookSystems у својој понуди, између осталог, имају и два библиотечка система: *Atrium* и *Concourse*.

Atrium [Atrium] је *web* базиран библиотечки информациони систем и реализован је као *web* апликација. Може да ради и на *Windows* и на *Linux* платформама, а клијентска страна је подржана од стране *browser-a*: *Microsoft Internet Explorer*, *Apple Safari* и *Firefox*. Има све стандардне функционалности као што су каталогизација, набавка књига, циркулација, *web* претраживање и извештавање. Апликација за циркулацију омогућава задуживање и раздуживање корисника, продужење чланарине, штампање реверса и признаница, претраживање корисника и публикација и резервацију публикација.

Concourse [Concourse] је библиотечки информационални систем имплементиран као *Windows* апликација и ради искључиво на *Windows* оперативним системима. Омогућава каталогизацију књига, инвентарисање, циркулацију, *web* претраживање и извештавање, као и разне друге додатне сервисе. Модул за циркулацију омогућава стандардне функције као што су задуживање и раздуживање корисника, резервацију публикација, штампање и слање опомене корисницима за прекорачење рока враћања књига, као и за доступност резервисаних књига. Такође постоји могућност увоза података о кориснику из текстуалног фајла или *CSV* фајла из *Microsoft Excel-a* или *Access-a*. Систем може бити прилагођен за три типа библиотека: школске, јавне/специјалне и црквене. У зависности од типа библиотеке доступан је и одређен скуп функционалности. У систем може бити укључена и функционалност штампања бар-кодова за кориснике и публикације.

CyberTools нуди библиотечки информациони систем ***CyberTools for Libraries*** [CyberTools] који је *web* базиран. Систем има све стандардне функционалности које омогућавају каталогизацију књига, *web* претраживање, циркулацију и извештавање. Модул за циркулацију омогућава све активности везане за рад са корисницима, а такође је омогућено да корисници сами кроз *web* интерфејс продуже своја задужења, резервишу публикације, мењају личне податке и поставе подсетнике који би их одређен број дана

пре истека рока враћања подсетили да треба да врате задужене публикације. *CyberTools* нуди два начина инсталације система у библиотеке. Један начин је да се систем инсталира на локални сервер библиотеке. Други начин је да библиотека нема локални сервер него да систем буде инсталиран на сервер *CyberTools*-а, а библиотека систем користи преко Интернета. У том случају *CyberTools* преузима обавезе везане за одржавања система и сигурност података.

Врло сличну солуцију нуди и *EOS International* кроз свој библиотечки информациони систем *EOS.Web* [EOS]. Систем је *web* базиран и постоји могућност инсталације система на EOS серверима.

Сви досад описани системи су комерцијални системи и поред њих постоји још велики број сличних система који нуде сличне скупове претходно описаних функционалности. Неки од њих су: *Millenium (Inovative Interfaces Inc.)* [Millenium], *Polaris Library Systems* [Polaris], *Athena* и *Spectrum (Sagebrush Corporation)* [Sagebrush], *M3* и *Oasis (Mandarin Library Automation Inc.)* [Mandarin] и многи други.

Постоји и велики број бесплатних и *open-source* библиотечких система. У даљем тексту су описани неки од њих.

Evergreen Integrated Library System (ILS) [Evergreen] је *open-source* библиотечки систем развијен од стране *Georgia Public Library Service*. Поседује модуле за каталогизацију, циркулацију, *web* претраживање и извештавање. Такође има подршку за *SIP2* протокол чиме је омогућена употреба бар-кода и *RFID* технологије. Модул за циркулацију омогућава све стандардне активности у раду са корисницима. Изглед екранске форме за задуживање и раздуживање је приказан на слици 1.4. Корисници такође имају могућност да се преко *web* интерфејса пријаве на свој налог и ту продуже задужења и виде напомене које је библиотекар евентуално ставио на њихов налог. Систем нема подршку за резервацију књига. Постоји могућност *offline* цикулације чиме је омогућен рад и у случају прекида везе између клијента и сервера. Систем је имплементиран већим делом у *Perl*-у, а неки делови су писани у програмском језику *C* и *Python*-у. Као сервер базе података се користи *open-source* база *PostgreSQL*. На сајту постоје сва упутства за

инсталацију и коришћење система, а такође је доступан и код, као и извршна верзија клијента за *Windows* системе.

Koha [Koha] је *open-source* библиотечки систем развијен под *General Public License (GPL)*. Постоји у две верзије: *Koha Classic* и *Koha ZOOM*. *Koha Classic* је решење за мале библиотеке са мањим бројем захтева, а *Koha ZOOM* за велике библиотеке и укључује све функционалности које систем поседује. Систем омогућава каталогизацију, набавку књига, *web* претраживање, циркулацију и извештавање. Модул за циркулацију је *web* базиран и омогућава задуживање и раздуживање корисника, претрагу каталога, резервацију публикација и аутоматско слање опомена корисницима *e-mail*-ом везаних за прекорачење рока враћања и резервације. Такође је корисницима омогућено да кроз *web* интерфејс могу сами мењати свој налог, резервисати књиге и продужити задужења. Систем је имплементиран у *Perl*-у. Као сервер базе података се користи база *MySQL*, а као *web* сервер *Apache* сервер. Са сајта се може скинути код система, као и извршна верзија система.

Evergreen и *Koha* су потпуно бесплатни системи и слободни за корићење. За библиотеке које немају информатичко особље које би инсталирало и одржавало систем постоји солуција коју нуди *LibLime* [LibLime]. *LibLime* нуди потпуну техничку подршку за системе *Evergreen* и *Koha* која укључује инсталацију система, даље одржавање, евентуалну миграцију података из неког претходно коришћеног система, као и даљи развој система за специфичне потребе библиотеке. Таква подршка од стране *LibLime* није бесплатна.

1.2 Библиотечки софтверски систем БИСИС

Библиотечки софтверски систем БИСИС развија се од 1993. године. Тренутно је актуелна трећа верзија система. Систем је базиран на *YUMARC* формату који је настао из *UNIMARC*-а и из захтева пројекта Библиотечка мрежа система научних и технолошких информација Србије. Развој и пројектовање овог система описан је у монографији [Surla04a].

У верзији 3 развијен је сопствени текст сервер за индексирање и претраживање библиографских записа у *UNIMARC* формату.

Главне карактеристике овог сервера су: специјализованост која резултује бољим перформансама, трослојна архитектура, коришћење *Java* платформе и независност од коришћеног релационог система за управљање базом података. Подршка за *Unicode* стандард доследно је спроведена у целокупном систему БИСИС верзија 3.

1.2.1 Функције система БИСИС

У раду [Lazarevic96] описане су функције прве верзије система БИСИС. У наредним верзијама система додате су још и друге функције система, тако да трећа верзија система подржава следеће функције:

- Дефинисање стандарда за обраду и претраживање
- Формирање библиографских записа
- Библиотечко извештавање и документовање
- Корисничко претраживање
- Преузимање записа
- Коришћење библиотечке грађе.

Дефинисање стандарда за обраду и претраживање омогућава опис структуре појединих стандарда и њихово представљање у бази података преко којих ће се обављати основне функције БИСИС-а.

Формирање библиографских записа подразумева обраду докумената и структурирање записа према усвојеном стандарду, преузимајући параметре конкретне обраде на основу унапред дефинисаних модела обраде докумената [Surla00 ,Pupovac].

Библиотечко извештавање и документовање омогућава формирање специфичних библиотекарских извештаја и каталога о документима. Формирање извештаја и каталога се врши на основу Базе библиографске грађе и Базе стандарда за обраду библиографских докумената [Zubic].

Корисничко претраживање обезбеђује обраду упита корисника и структурирање одговора, односно резултата претраживања, на основу садржаја Базе библиографске грађе. Структурирање резултата претраживања врши се на основу унапред дефинисаних структура, садржаја и формата приказа који су

дефинисани у Бази стандарда за обраду библиографских докумената [Surla98, Milosavljevic97].

Преузимање библиографских записа унутар система БИСИС имплементирано је преко протокола који је базиран на размени порука у виду XML документа што је детаљно описано у радовима [Zaric03, Zaric04a, Zaric04b, Surla04c, Todoric].

Коришћење библиотечке грађе омогућава евидентирање корисника библиотеке, као и праћење коришћења фонда библиотеке, резервисање и задуживање докумената. Због тога је, поред базе података о библиографској грађи, потребно обезбедити и базу података о корисницима библиотеке [Tosic, Surla96]. Овај сегмент у актуелној трећој верзији система задовољава потребе за пословање факултетских библиотека.

1.2.2 Примена XML технологије у развоју система БИСИС

Примена XML технологија у развоју библиотечког софтверског система БИСИС обухвата следеће:

- Опис библиографских формата и библиографских записа помоћу XML шема језика
- Контролу квалитета XML библиографских записа
- Употребу XML native технологија
- Генерисање каталожних листића на основу XML библиографских записа
- Имплементацију XML едитора за библиографске формате и библиографске записе

У радовима [Zeremski04, Zeremski01a, Zeremski01b] показано је да се помоћу XML шема језика могу моделирати сви концепти UNIMARC формата, као и сви елементи тих концепата.. XML шема библиографских записа описана је у радовима [Zeremski01c, Zeremski02, Zeremski01d]. Међутим, у радовима [Dimic07a, Milosavljevic07] дат је предлог XML шема које описују концепте библиографских формата (YUMARC, UNIMARC, MARC21) као што су поља, потпоља, индикатори, шифарници, док појаве тих концепата садрже конкретне податке о формату, односно спецификацију појединачних поља, потпоља, индикатора и друго.

Поред тога, у овим радовима описане су и XML шеме библиографских записа формираних по тим форматима. Монографија [Budimir04] посвећена је контроли квалитета библиографских записа. Циљ истраживања приказаних у радовима [Skrbic04, Skrbic05] је испитивање могућности употребе XML native технологија..У радовима [Radjenovic06a, Radjenovic06b] приказано је моделирање и имплементација софтверског пакета за формирање библиотечких каталожских листића. Различити аспекти имплементације XML едитора за библиографске формате и библиографске описану су радовима [Surla03, Mijic03, Skrbic05, Dimic07b].

1.3 Коришћење библиографске грађе у систему БИСИС

Увођењем система БИСИС вер.3 у градске библиотеке констатовано је да постојећа верзија система за коришћење библиотечке грађе не задовољава све захтеве библиотечког пословања градских библиотека. Информациони захтеви процеса коришћења библиотечке грађе односе се на следеће:

- Податке о кориснику библиотеке
- Податке о чланству
- Податке о коришћењу библиотечке грађе
- Опомене корисницима библиотеке
- Статистичке извештаје
- Перформансе корисничког интерфејса
- Архивирање података

Подаци о кориснику библиотеке. При упису у библиотеку потребан је одређен скуп података о кориснику. Овај скуп података дефинисан је од стране библиотека. Неки од тих података из скупа треба да буду обавезни при упису корисника и ту обавезност дефинише такође библиотека. Скуп обавезних података се разликује од библиотеке до библиотеке и зато је потребно обезбедити подешавање обавезности података посебно за сваку библиотеку.

Подаци о чланству. Потребно је евидентирати податке о продужењима чланства корисника као и давати индикације о истеку чланства. Постоје различите врсте чланства и категорије

корисника које се дефинишу од стране библиотеке. Врстом чланства се дефинише временски период трајања чланства корисника, а категоријом корисника максималан број публикација које корисник може да има на задужењу, као и максималан временски период задржавања тих публикација. Висина чланарине за корисника зависи од категорије и врсте чланства корисника и дефинише се од стране библиотеке. Потребно је такође за сваки упис или продужење чланства водити евиденцију који библиотекар га је извршио и на ком одељењу библиотеке је извршено.

Подаци о коришћењу библиотечке грађе. То су подаци о задужењима и раздужењима корисника публикацијама. При раду библиотекара са корисником потребно је следеће: да се приказују само тренутна задужења; да се даје индикације о евентуалном прекорачењу рока враћања публикација; да се може приказати историја задужења корисника. За свако задужење, раздужење и продужење задужења треба евидентирати податак о библиотекару који га је извршио као и одељењу библиотеке на ком је извршено.

Опомене корисницима библиотеке. Потребно је омогућити периодично штампање опомена корисницима који касне са враћањем публикација. Постоји више типова опомена и садржај који се приказује на њима се разликује од библиотеке до библиотеке тако да треба библиотеци омогућити измену садржаја опомена. При раду библиотекара са корисником треба да је видљива информација о актуелним посланим опоменама кориснику. Такође треба да се омогући штампање годишњег регистра опомена којег библиотека чува у својој архиви.

Статистички извештаји. Очекује се могућност генерисања одређеног скупа статистичких извештаја о корисницима и коришћењу фонда библиотеке. То су следећи извештаји: књига уписа; структура уписаних корисника по категорији, врсти чланства и полу; структура посетилаца по категорији, врсти чланства и полу; историја задужења корисника; историја задужења публикације; најчитаније књиге; највреднији читаоци; издате и враћене публикације по УДК групама; издате и враћене публикације по УДК грапама и категорији корисника и др. Такође се очекује могућност претраживања корисничке базе података по свим могућим критеријумима чиме би се омогућило добијање разних

статистичких података специфичних за библиотеку којих нема у предефинисаном скупу извештаја.

Перформансе корисничког интерфејса. У циљу ефикаснијег рада са корисницима посебну пажњу треба посветити смањењу броја операција које библиотекар треба да изврши да би обрадио захтев корисника као и брзом одзиву софтверског система.

Архивирање података. Такође, у циљу ефикаснијег рада са корисницима потребно је омогућити да се подаци о корисницима који нису продужили чланство склоне из актуелне базе корисника тј. да се архивирају. Један предлог архивирања података о корисницима преко XML докумената је описан у [Tesendic].

На основу приказаних информационих захтева и прегледа постојећих софтверских ситета урађено је моделирање и имплементација система за коришћење библиотечке грађе у оквиру нове (четврте) верзије библиотечког информационог система БИСИС. Тестирање и верификација ове апликације извршено је на подацима једне градске библиотеке и констатовано је да су испуњени сви наведени информациони захтеви библиотека.

Спецификација информационих захтева система за коришћење библиотечке грађе

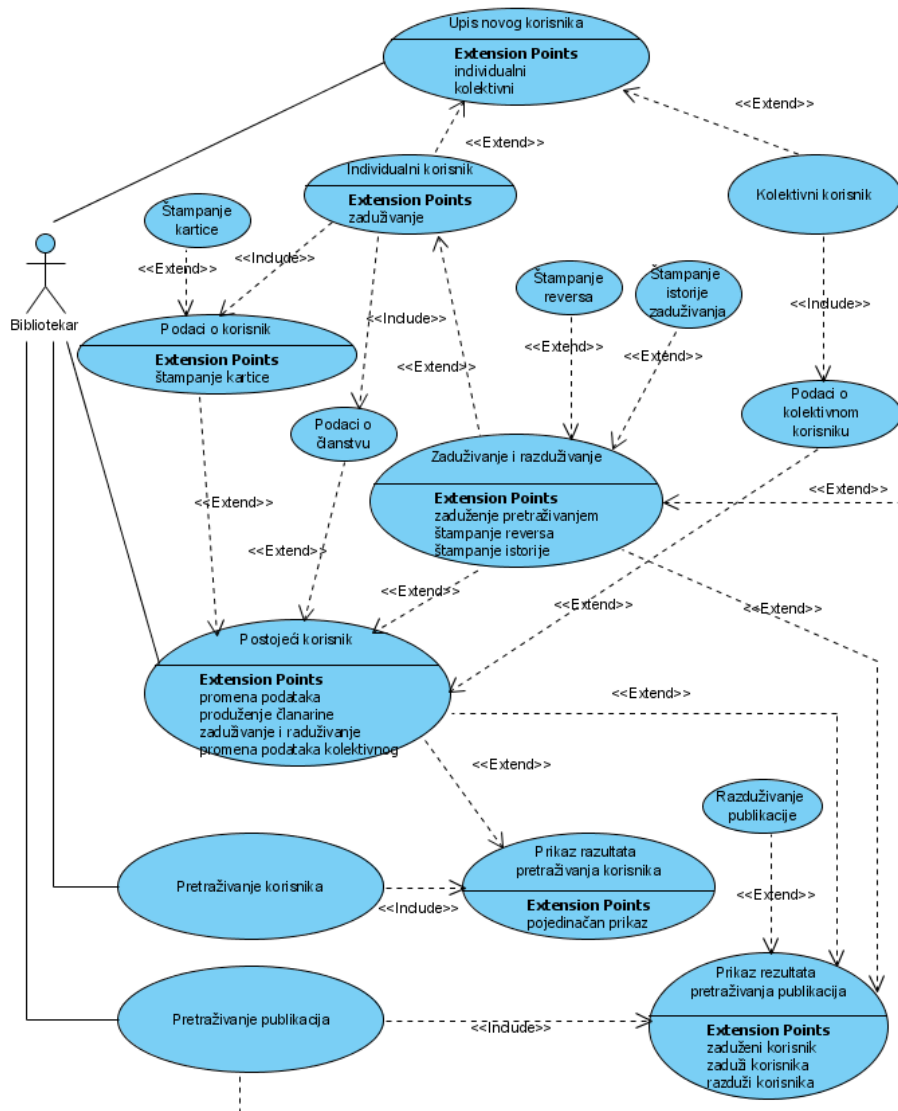
У овом поглављу је описано моделирање система за коришћење библиотечке грађе. На основу овог модела је извршена имплементација апликације *Циркулација* вер. 4. За моделирање је коришћен обједињени језик моделирања (UML 2.0) и CASE алат *Visual Paradigm for UML 5.3* [VisParadigm]. У првом одељку овог поглавља су приказани дијаграми случајева коришћења и дат је текстуални опис свих случајева коришћења са тих дијаграма. У другом одељку је приказана статичка структура система дефинисана преко дијаграма пакета, дијаграма класа и текстуалног описа класа. Динамичка структура система је приказана у трећем одељку и дата је дијаграмима секвенци који описују сценарије случајева коришћења.

2.1 Дијаграми случајева коришћења

Спецификација захтева система је представљена са три дијаграма случајева коришћења. Основна функционалност система је представљена на дијаграму случајева коришћења *Рад са корисницима*. Он обухвата функције које библиотекар свакодневно врши у раду са корисницима. Ту спадају учлањење нових корисника, продужење чланарине, задуживање и раздуживање корисника и претраживање фонда. На дијаграму *Извештавање* је представљена функцијоналност дела система за генерисање извештаја и дела система за генерисање опомена које се периодично

шаљу корисницима који касне са враћањем задужених публикација. Дијаграм *Параметри система* обухвата функције које врши администратор система да би прилагодио ситем за рад у конкретној библиотеци.

2.1.1 Дијаграм случајева коришћења *Рад са корисницима*



Слика 2.1 Дијаграм случајева коришћења *Рад са корисницима*

Текстуални опис случајева коришћења:

Случај коришћења: Упис новог корисника

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

*Опис: Библиотекар уписује новог корисника у библиотеку. Постоје две врсте корисника: индивидуални и колективни корисници. Библиотекар бира врсту корисника и зависно од избора реализује се случај коришћења *Индивидуални корисник*, односно случај коришћења *Колективни корисник*.*

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Уписан је нови корисник у библиотеку.

Случај коришћења: Индивидуални корисник

Учесници: Библиотекар

*Услови који морају бити задовољени пре извршавања: У случају коришћења *Упис новог корисника* изабран је упис индивидуалног корисника.*

*Опис: Библиотекар уписује индивидуалног корисника у библиотеку. Индивидуални корисник је појединац који може да користи фонд библиотеке. Појединац може да се упише у библиотеку и преко колектива који је већ уписан у библиотеку као колективни корисник. У том случају постоји информација преко ког колективног корисника је уписан тај индивидуални корисник. У оквиру овог случаја коришћења реализују се случајеви коришћења *Подаци о кориснику* и *Чланарина*. Ако корисник при упису жели да задужи неке публикације реализује се и случај коришћења *Задуживање* и *раздуживање*.*

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Уписан је индивидуални корисник у библиотеку.

Случај коришћења: Подаци о кориснику

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења Упис новог корисника изабран је упис индивидуалног корисника или је при реализацији случаја коришћења Постојећи корисник био изабран индивидуални корисник.

Опис: Библиотекар уноси или ажурира податке о индивидуалном кориснику. У те податке спадају: име и презиме, име родитеља, адреса и место становања, број телефона, *e-mail* адреса, ЈМБГ, број документа и место издавања документа, пол, узраст, додатна адреса, степен образовања, занимање, звање, организација у којој је корисник запослен, матерњи језик, интересовања и напомена. Библиотека одређује обавезност уноса података. Ако библиотекар жели да штампа картицу реализује се случај коришћења Штампање картице.

Изузеци:

[погрешно унети подаци] Поправити унос.

[нису унети обавезни подаци] Унети обавезне податке.

Услови који морају бити задовољени после извршавања: Ажурирани су подаци о кориснику.

Случај коришћења: Подаци о чланству

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају

коришћења *Упис новог корисника* изабран је упис индивидуалног корисника или је при реализацији случаја коришћења *Постојећи корисник* био изабран индивидуални корисник.

Опис: Библиотекар уноси или ажурира податке о чланству корисника у библиотеку. У те податке спадају: број корисника, ако се корисник уписује преко колективног корисника тада се повезује са подацима о колективном кориснику, категорија корисника, врста учлањења, датум од кад важи чланарина, датум до кад важи чланарина, одељење на које се корисник уписује, цена, и број призанице која се издаје кориснику за плаћену чланарину. Библиотека одређује обавезност уноса података.

Изузеци:

[погрешно унети подаци] Поправити унос.

[нису унети обавезни подаци] Унети обавезне податке.

Услови који морају бити задовољени после извршавања: Ажурирани су подаци о чланству корисника.

Случај коришћења: Задуживање и раздуживање

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења *Упис новог корисника* изабран је упис индивидуалног корисника и обавезни подаци о кориснику и чланству су унешени или је при реализацији случаја коришћења *Постојећи корисник* био изабран индивидуални корисник или је при реализацији случаја коришћења *Приказ резултата претраживања публикација* изабрано задуживање одређеног корисника неком од публикација из резултата.

Опис: Приказује се списак тренутних задужења корисника. Ако корисник враћа неку од задужених публикација библиотекар може да га раздужи или може да продужи задужење. Ако корисник жели да задужи неку публикацију библиотекар може задужење да изврши уношењем инвентарног броја публикације или

проналажењем публикације претраживањем. У случају проналажења публикације претраживањем реализује се случај коришћења *Претраживање публикација*. Ако библиотекар жели да штампа реверс кориснику, тада се реализује случај коришћења *Штампање реверса*. Ако библиотекар жели да види или штампа претходна задужења корисника, тада се реализује случај коришћења *Штампање историје задуживања*.

Изузеци:

[погрешно унет инвентарни број] Исправно унети инвентарни број.

Услови који морају бити задовољени после извршавања: Корисник је раздужен публикацијама које је вратио, задужен новим публикацијама и продужено је тренутно задужење.

Случај коришћења: Штампање картице

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења *Упис новог корисника* изабран је упис индивидуалног корисника и обавезни подаци о кориснику и чланству су унешени или је при реализацији случаја коришћења *Постојећи корисник* био изабран индивидуални корисник.

Опис: Библиотекару се приказује изглед картице коју жели да штампа. Библиотекар шаље захтев за штампање.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Одшtamпана је картица кориснику.

Случај коришћења: Штампање реверса

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Корисник има бар једно задужење.

Опис: Библиотекару се приказује изглед реверса који жели да штампа. Библиотекар шаље захтев за штампање. Реверс је документ на ком се приказују тренутна задужења корисника. Стављањем потписа на реверс корисник потврђује тачност података приказаних на реверсу.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Одштампан је реверс кориснику.

Случај коришћења: Штампање историје задуживања

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Корисник има бар једно задужење.

Опис: Библиотекару се приказује списак претходних и тренутних задужења корисника који библиотекар може да одштампа на захтев корисника.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Одштампана је историја задуживања корисника.

Случај коришћења: Колективни корисник

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења Упис новог корисника изабран је упис колективног корисника.

Опис: Библиотекар уписује колективног корисника у библиотеку. Колективни корисник је организација која учлањењем у библиотеку својим члановима омогућава одређене погодности при њиховом учлањењу. Те погодности зависе искључиво од уговора између колективног члана и библиотеке. У оквиру овог случаја коришћења се реализује случај коришћења Подаци о колективном кориснику.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Уписан је колективни корисник у библиотеку.

Случај коришћења: Подаци о колективном кориснику

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења Упис новог корисника изабран је упис колективног корисника или је при реализацији случаја коришћења Постојећи корисник био изабран колективни корисник.

Опис: Библиотекар уноси или ажурира податке о колективном кориснику. У те податке спадају: број корисника, назив организације, адреса и место организације, број телефона, број телефакса, *e-mail* адреса, додатна адреса, подаци о контакт особи из организације у које спадају име, презиме и *e-mail* адреса особе. Библиотека одређује обавезност уноса података.

Изузеци:

[погрешно унети подаци] Поправити унос.

[нису унети обавезни подаци] Унети обавезне податке.

Услови који морају бити задовољени после извршавања: Ажурирани су подаци о колективном кориснику.

Случај коришћења: Постојећи корисник

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Библиотекар је унео број корисника или је при реализацији случаја коришћења *Приказ резултата претраживања корисника* изабран преглед података о неком кориснику из резултата или је при реализацији случаја коришћења *Приказ резултата претраживања публикација* изабран приказ података о кориснику код кога је задужена публикација.

Опис: У завистности од тога да ли је корисник индивидуални или колективни реализују се следећи случајеви коришћења:

- у случају индивидуалног корисника случајеви коришћења *Подаци о кориснику*, *Чланарина* и *Задуживање и раздуживање*.
- у случају колективног корисника случај коришћења *Подаци о колективном кориснику*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Ако је вршено ажурирање података, оно је успешно извршено.

Случај коришћења: Претраживање корисника

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Омогућено је претраживање по сваком податку о кориснику који се чува у систему. Претраживање се врши укрштањем до пет података операторима *and*, *or* и *not*. Поред тога ти подаци се могу укрстити са периодом у ком је корисник задуживан или

раздуживан и у ком је вршио продужење чланарине. У оквиру овог случаја коришћења се реализује случај коришћења *Приказ резултата претраживања корисника*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Библиотекар је добио резултате за задати упит.

Случај коришћења: Приказ резултата претраживања корисника

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Извршено је претраживање корисника.

Опис: Приказује се број погодака и погодци. За сваки погодак, тј. Корисника који је задовољио упит приказује се: број корисника, име, презиме, име родитеља, адреса и место становања. Ако библиотекар жели да види остале податке, тада се реализује случај коришћења *Постојећи корисник*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Библиотекар је добио резултате за задати упит.

Случај коришћења: Претраживање публикација

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Омогућено је претраживање по дефинисаним префиксима текст сервера библиотечког система БИСИС. Претраживање се врши укрштањем до пет префикса операторима *and*, *or* и *not*. Поред тога префикси се могу укрстити са периодом у ком је публикација

задужена или раздужена. У оквиру овог случаја коришћења се реализује случај коришћења *Приказ резултата претраживања публикација*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Библиотекар је добио резултате за задати упит.

Случај коришћења: Приказ резултата претраживања публикација

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Извршено је претраживање публикација.

Опис: Приказује се број погодака и погодци. Омогућен је приказ погодака на нивоу записа и на нивоу примерака. За сваки примерак је означено да ли може да се задужи или не. Ако је примерак задужив, тада се случај коришћења може проширити реализацијом случаја коришћења *Задуживање и раздуживање*. Ако је примерак задужен, тада се случај коришћења може проширити реализацијом случаја коришћења *Раздуживање*. Ако библиотекар жели да види податке о кориснику код кога је примерак задужен, тада се реализује случај коришћења *Постојећи корисник*. За сваки погодак се приказује кратак инфо о публикацији који садржи одређени подскуп информација из записа.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Библиотекар је добио резултате за задати упит.

Случај коришћења: Раздуживање

Учесници: Библиотекар

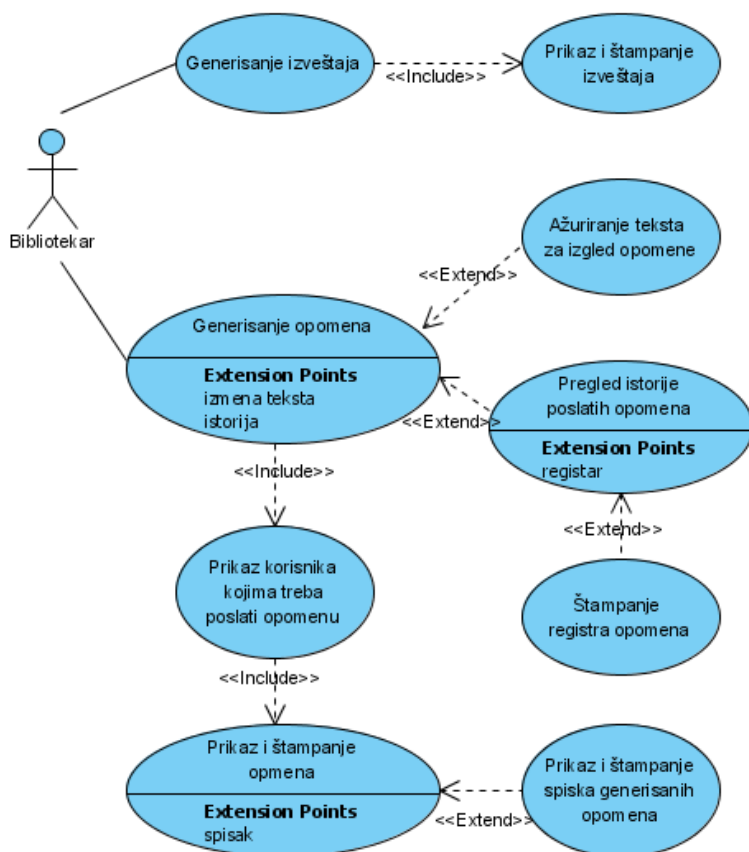
Услови који морају бити задовољени пре извршавања: Публикација је задужена код неког корисника.

Опис: Публикација се раздужује.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Публикација је раздужена.

2.1.2 Дијаграм случајева коришћења *Извештавање*



Слика 2.2 Дијаграм случајева коришћења *Извештавање*

Текстуални опис случајева коришћења:

Случај коришћења: Генерисање извештаја

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Постоји више предефинисаних врста извештаја у систему, као што су *Књига уписа*, *Структура уписаних корисника*, *Најчитаније књиге* и други. Библиотекар задаје период за који се генерише извештај и врсту извештаја који се генерише. Реализује се случај коришћења *Приказ и штампање извештаја*.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Изгенерисан је изабрани извештај за задати период.

Случај коришћења: Приказ и штампање извештаја

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења *Генерисање извештаја* је изабрана врста извештаја који се генерише.

Опис: Библиотекару се приказују изабрани извештај за задати период. Библиотекар шаље захтев за штампање.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Извештај је одштампан.

Случај коришћења: Генерисање опомена

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Опомена је документ који библиотека шаље кориснику у случају када је кориснику истекао рок враћања за задужене публикације. Постоје два типа документа: Опомена и Опомена пред утужење. Библиотекар задаје период за који се генеришу опомене и тип документа који се генерише. Реализује се случај коришћења *Приказ корисника којима треба послати опомене*. Постоје предефинисани текстови за сваки тип документа. Ако библиотекар жели да мења текст на изглед опомене, тада се реализује случај коришћења *Ажурирање текста за изглед опомене*. Ако библиотекар жели да види претходно послате опомене, тада се реализује случај коришћења *Преглед историје послатих опомена*.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Изгенерисане су опомене за задати период.

Случај коришћења: Приказ корисника којима треба послати опомену

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Библиотекару се приказује списак корисника којима за задати период треба послати опомену. Библиотекар може да изостави поједине кориснике са тог списка. За сваког корисника се приказује број корисника, име, презиме и број публикација које дугује. Реализује се случај коришћења *Приказ и штампање опомена*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Изабрани су корисници за које ће се генерисати и штампати опомене.

Случај коришћења: Приказ и штампање опомена

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења Приказ корисника којима треба послати опомену постоји бар један корисник.

Опис: Библиотекару се приказују опомене за изабране кориснике. Библиотекар шаље захтев за штампу штампачу. Ако жели библиотекар може да одштампа и списак изгенерисаних опомена. У том случају се реализује случај коришћења Приказ и штампање списка генерисаних опомена.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Опомене су одштампане.

Случај коришћења: Приказ и штампање списка генерисаних опомена

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: У случају коришћења Приказ и штампање опомена генерисана је бар једна опомена.

Опис: Библиотекару се приказују списак изгенерисаних опомена. Библиотекар шаље захтев за штампање.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Одштампан је списак генерисаних опомена.

Случај коришћења: Ажурирање текста за изглед опомене

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

Опис: За сваки тип документа постоји шаблон у коме се налази комплетан текст који се исписује на документу. У тај текст спадају: назив и адреса библиотеке, назив опомене, текст који се исписује на опомени, подаци о трошковима слања опомене и висини казне. Библиотекар може да мења тај текст, као и писмо на ком ће се штампати опомена.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Шаблон је измењен.

Случај коришћења: Преглед историје послатих опомена

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

*Опис: Библиотекар може да погледа листу послатих опомена за одређени период. За сваку опомену се зна број опомене, датум, корисник коме је послата и публикације за које је послата. Библиотекар може да штампа регистар опомена. У том случају се реализује случај коришћења *Штампање регистра опомена*.*

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Добијен је

преглед историје за задати период.

Случај коришћења: Штампање регистра опомена

Учесници: Библиотекар

Услови који морају бити задовољени пре извршавања: Нема.

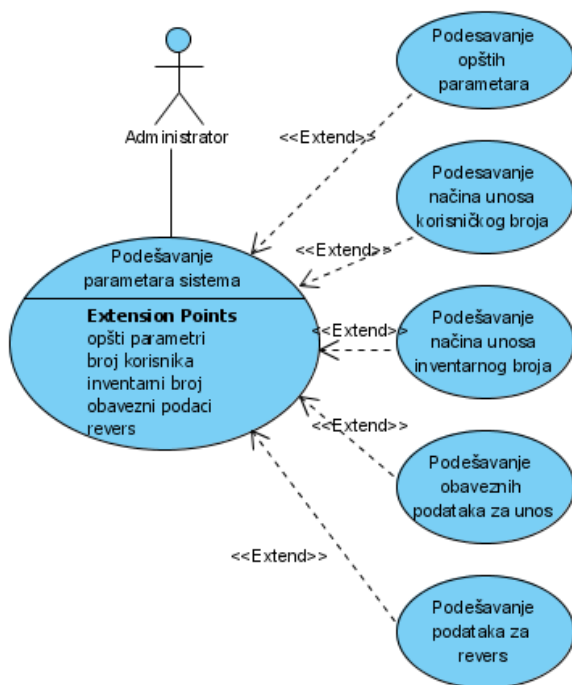
Опис: Библиотекару се приказује регистар опомена за одређени период. За сваку опомену се приказује број опомене, датум слања, корисник коме је послата и публикације за које је послата. Библиотекар шаље захтев за штампање.

Изузеци:

[штампач не ради] Оспособити штампач за рад.

Услови који морају бити задовољени после извршавања: Регистар опомена је одштампан.

2.1.3 Дијаграм случајева коришћења *Параметри система*



Слика 2.3 Дијаграм случајева коришћења *Параметри система*

Текстуални опис случајева коришћења:

Случај коришћења: Подешавање параметара система

Учесници: Администратор

Услови који морају бити задовољени пре извршавања: Нема.

Опис: Администратор подешава параметре за рад система. У зависности од избора реализују се случајеви коришћења *Подешавање општих параметара*, *Подешавање начина уноса корисничког броја*, *Подешавање начина уноса инвентарног броја*, *Подешавање обавезних података за унос* или *Подешавање података за реверс*.

Изузеци: Нема.

Услови који морају бити задовољени после извршавања: Подешени су параметри система.

Случај коришћења: Подешавање општих параметара

Учесници Администратор

Услови који морају бити задовољени пре извршавања: У случају коришћења Подешавање параметара система изабрано је подешавање општих параметара.

Опис: Библиотекар подешава опште параметре. Ту спадају: локација на којој се рачунар налази, величина фонта и прозора апликације, изглед и боју апликације, предефинисана вредност за место и поштански број, да ли је дозвољено задуживање необрађених књига, да ли је дозвољено задуживање корисника са блокираном картицом и да ли се задужена књига аутоматски раздужује при поновном задужењу.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Унесене су вредности за подешавање општих параметара.

Случај коришћења: Подешавање начина уноса корисничког броја

Учесници: Администратор

Услови који морају бити задовољени пре извршавања: У случају коришћења Подешавање параметара система изабрано је подешавање начина уноса корисничког броја.

Опис: Администратор подешава начин уноса корисничког броја. Ту

спада: дужина корисничког броја, дужина и предефинисана вредност префикса одељења, да ли се број уноси у облику са сепаратором и у ком редоследу, ознака сепаратора.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Унесене су вредности за подешавање начина уноса корисничког броја.

Случај коришћења: Подешавање начина уноса инвентарног броја

Учесници: Администратор

Услови који морају бити задовољени пре извршавања: У случају коришћења Подешавање параметара система изабрано је подешавање начина уноса инвентарног броја.

Опис: Администратор подешава начин уноса инвентарног броја. Ту спада: дужина инвентарног броја, дужина и предефинисана вредност ознаке локације, дужина и предефинисана вредност ознаке инвентарне књиге, да ли се број уноси у облику са сепараторима и у ком редоследу, ознаке сепаратора.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Унесене су вредности за подешавање начина уноса инвентарног броја.

Случај коришћења: Подешавање обавезних података за унос

Учесници: Администратор

Услови који морају бити задовољени пре извршавања: У случају коришћења Подешавање параметара система изабрано је

подешавање обавезних података за унос.

Опис: Сви подаци који су дефинисани у случајевима коришћења Подаци о кориснику, Чланарина и Подаци о колективном кориснику могу бити обавезни или необавезни за унос. Администратор за сваки податак подешава да ли је обавезан или не.

Изузеци:

[погрешно унет податак] Поправити унос.

Услови који морају бити задовољени после извршавања: Унесене су вредности за подешавање обавезних података за унос.

Случај коришћења: Подешавање података за реверс

Учесници: Администратор

Услови који морају бити задовољени пре извршавања: У случају коришћења Подешавање параметара система изабрано је подешавање података за реверс.

Опис: Штампање реверса зависи од библиотеке. Администратор уноси назив и адресу библиотеке, као и висину и ширину папира на који се штампа реверс и број примерака који се штампају.

Изузеци:

[погрешно унет податак] Поправити унос.

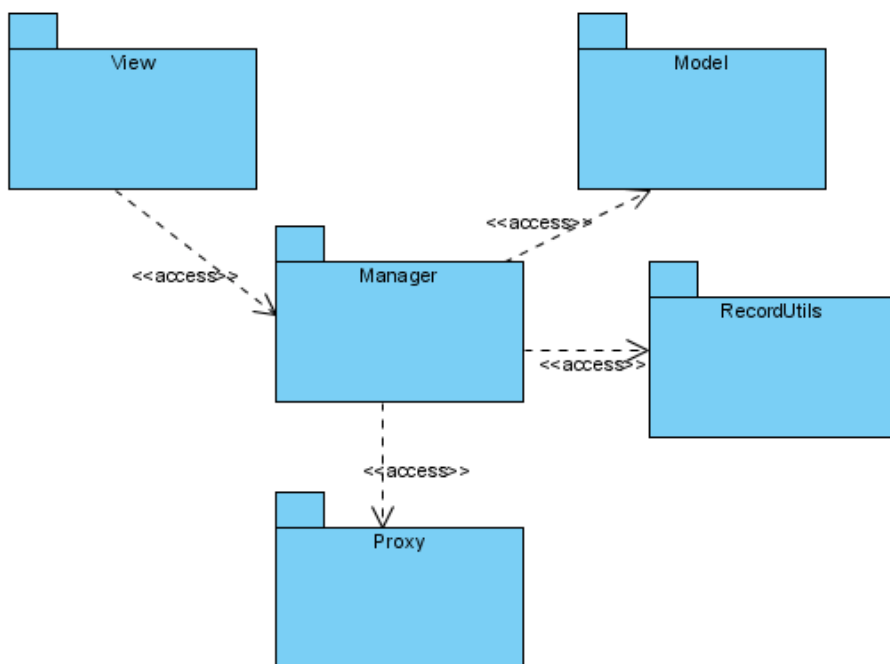
Услови који морају бити задовољени после извршавања: Унесене су вредности за подешавање реверса.

2.2 Статичка структура система

Статичка структура система је описана на једном дијаграму пакета и четири дијаграма класа. На дијаграму класа *Кориснички интерфејс* приказане су класе корисничког интерфејса тј. екранских форми преко којих корисник комуницира са системом. На дијаграму класа *Подаци* приказан је логички модел базе података и на основу тог модела се генерише физички модел базе података. На дијаграму класа *Proxy* су приказане класе које се користе за комуникацију са базом података, а на дијаграму класа *RecordUtils* су приказане класе које представљају модел библиотечког записа у систему за коришћење библиотечке грађе.

2.2.1 Дијаграм пакета система

На слици 2.4 је приказан дијаграм пакета система за коришћење библиотечке грађе којим је описана структура система.



Слика 2.4 Дијаграм пакета

Пакет *View* садржи класе корисничког интерфејса које су детаљније описане и приказане у одељку 2.2.2.

Пакет *Model* садржи класе које представљају објектни модел базе података и служе за привремено складиштење података с којима систем тренутно ради. Ове класе су описане и приказане у одељку 2.2.3.

Пакет *Manager* садржи класе које управљају током податка од корисничког интерфејса до базе података и обрнуто. Овај пакет је описан у одељку 2.2.4.

Пакет *Proxy* садржи интерфејс *Proxy* и његове две имплементације. Преко овог интерфејса класе из пакета *Manager* комуницирају са физичком базом податка. Овај пакет је описан у одељку 2.2.5.

Пакет *RecordUtils* садржи интерфејс који представља модел библиотечког записа у систему за коришћење библиотечке грађе. Овај интерфејс има две имплементације. Пакет је описан у одељку 2.2.6.

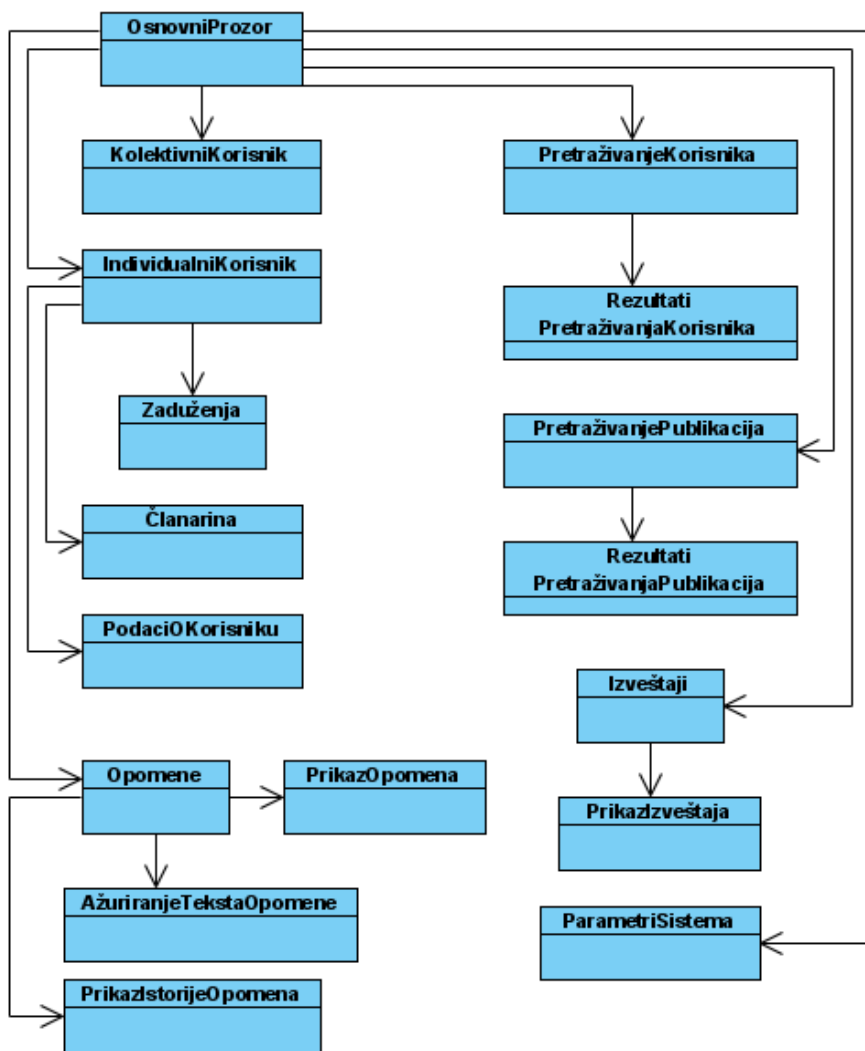
2.2.2 Класе пакета *View*

На слици 2.5 је дат дијаграм класа пакета *View*. На дијаграму су приказане основне класе корисничког интерфејса и везе између њих.

Преко класе *OsnovniProzor* корисник бира акцију коју жели да изврши.

За остваривање функционалности дефинисане на дијаграму случајева коришћења *Раđ са корисницима* користе се класе *IndividualniKorisnik*, *KolektivniKorisnik*, *PodaciOKorisniku*, *Članarina*, *Zaduženja*, *PretraživanjeKorisnika*, *RezultatiPretraživanjaKorisnika*, *PretraživanjePublikacija* и *RezultatiPretraživanjaPublikacija*. За остваривање функционалности дефинисане на дијаграму случајева коришћења *Извештавање* користе се класе *Izveštaji*, *PrikazIzveštaja*, *Opomene*, *PrikazOpomena*, *PrikazIstorijeOpomena* и *AžuriranjeTekstaOpomene*. А за функционалност дефинисану на

дијаграму случајева коришћења *Параметри система* користи се класа *ParametriSistema*.



Слика 2.5 Дијаграм класа *Кориснички интерфејс*

Преко класе *IndividualniKorisnik* и *KolektivniKorisnik* корисник система (библиотекар) има приступ подацима индивидуалних, односно колективних корисника библиотеке и може да иницира акције предвиђене за рад са тим подацима. Класа *IndividualniKorisnik*

користи класу *PodaciOKorisniku* за приступ основним подацима о кориснику, класу *Članarina* за приступ подацима о чланству корисника и класу *Zaduženja* за приступ подацима о задужењу корисника.

Преко класе *PretraživanjeKorisnika* корисник система (библиотекар) задаје критеријуме за претраживање корисника библиотеке. Резултати тог претраживања се приказују преко класе *RezultatiPretraživanjaKorisnika*. Слично томе преко класе *PretraživanjePublikacija* корисник система задаје критеријуме за претраживање публикација, а резултати претраживања се приказују преко класе *RezultatiPretraživanjaPublikacija*.

Преко класе *Izveštaji* се врши избор извештаја који систем треба да генерише кориснику. Генерисани извештај се кориснику приказује преко класе *PrikazIzveštaja*.

Преко класе *Opomene* корисник приступа генерисању опомена, прегледу историје генерисаних опомена и ажурирању текста опомене. Те акције корисник извршава преко класа *PrikazOpomena*, *PrikazIstorijeOpomena* и *AžuriranjeTekstaOpomene*.

Преко класе *ParametriSistema* администратор система врши подешавање параметара система.

2.2.3 Класе пакета *Model*

На слици 2.6 је дат дијаграм класа пакета *Model*. На њему је приказан објектни модел базе података. На основу модела се генерише и физички модел базе података.

Класа *Users* садржи податке о индивидуалном кориснику библиотеке. Има атрибуте: *sys_id* (јединствен системски идентификациони број корисника), *user_id* (број чланске карте корисника), *first_name* (име корисника), *last_name* (презиме), *parent_name* (име родитеља), *address* (адреса), *city* (место становања), *zip* (број поште), *phone* (телефон), *email* (e-mail адреса), *jmbg* (јединствени идентификациони број грађанина), *doc_id* (тип документа: лична карта, пасош, остало), *doc_no* (број документа), *doc_city* (место издавања документа), *country* (земља издавања документа), *gender* (пол), *age* (узраст), *sec_address* (додатна адреса),

sec_city (место становања додатне адресе), *sec_zip* (број поште додатне адресе), *sec_phone* (додатни телефон), *note* (напомена), *interests* (интересовања), *warning* (индикатор опомена), *occupation* (занимање), *title* (звање), *index_no* (број индекса), *class* (година школовања). На основу веза са класама *Groups*, *User_categs*, *Mmbr_types*, *Edu_lvl*, *Language* и *Organization* идентификују се колективни члан ком корисник припада, категорија корисника, врста учлањења, степен образовања корисника, матерњи језик и организација у којој је запослен.

Класа *Groups* садржи податке о колективном кориснику библиотеке. Има атрибуте: *sys_id* (јединствен системски идентификациони број корисника), *user_id* (број корисника), *inst_name* (назив институције), *sign_date* (датум уписа), *address* (адреса), *city* (место), *zip* (број поште), *phone* (телефон), *email* (e-mail адреса), *fax* (факс), *sec_address* (додатна адреса), *sec_city* (место додатне адресе), *sec_zip* (број поште додатне адресе), *sec_phone* (додатни телефон), *cont_name* (име контакт особе), *cont_lname* (презиме контакт особе), *cont_email* (e-mail адреса контакт особе).

Класа *Signing* садржи податке о чланарини индивидуалног корисника. Има атрибуте: *sign_date* (датум од кад важи чланарина), *until_date* (датум до кад важи чланарина), *cost* (цена), *receipt_id* (број признанице). На основу везе са класама *Users*, *Librarian* и *Location* идентификују се корисник библиотеке, библиотекар који је продужио чланарину и локација на којој је продужена чланарина.

Класа *Lending* садржи податке о задужењима и раздужењима корисника. Има атрибуте: *ctlg_no* (инвентарни број), *lend_date* (датум задужења), *return_date* (датум враћања), *resume_date* (датум продужења), *deadline* (рок враћања). На основу везе са класама *Users*, *Librarian* и *Location* идентификују се корисник библиотеке, библиотекар који је задужио публикацију, библиотекар који је продужио задужење, библиотекар који је раздужио публикацију и локација на којој је публикација задужена.

Класа *Librarian* садржи податке о библиотекарима који имају приступ систему. Има атрибуте: *username* (корисничко име), *password* (лозинка), *firs_name* (име), *last_name* (презиме), *address* (адреса становања), *city* (место), *zip* (број поште), *phone* (телефон),

email (e-mail адреса), *admin* (индикатор администраторских привилегија).

Класа *Location* је шифарник локација на којима се врши задуживање и учлањење. Има атрибуте: *id* (шифра локације), *name* (назив локације), *last_user_id* (бројач за аутоматску доделу броја новим корисницима).

Класа *User_categs* је шифарник категорија корисника. Има атрибуте: *id* (шифра категорије), *name* (назив категорије), *titles_no* (максималан број задужења корисника те категорије), *period* (временски период трајања задужења).

Класа *Mmbr_types* је шифарник врста учлањења. Има атрибуте: *id* (шифра врсте), *name* (назив врсте), *period* (временски период трајања чланарине).

Класа *Membership* је шифарник цена чланарине. Има атрибуте: *cost* (цена чланарине). На основу везе са класама *User_categs* и *Mmbr_types* идентификују се врста учлањења и категорија корисника.

Класа *Edu_lvl* је шифарник степена образовања. Има атрибуте: *id* (шифра степена образовања), *name* (назив степена).

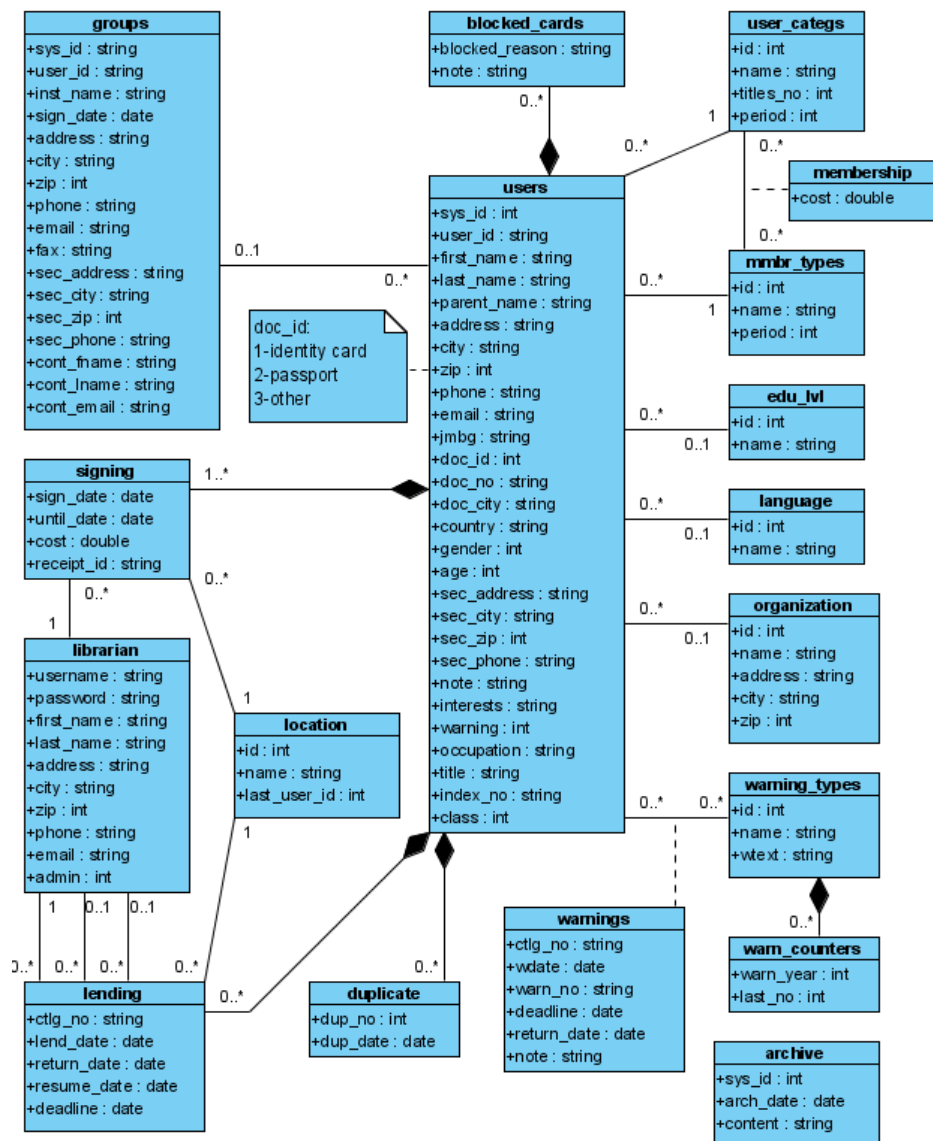
Класа *Language* је шифарник језика. Има атрибуте: *id* (шифра језика), *name* (назив језика).

Класа *Organization* је шифарник организација. Има атрибуте: *id* (шифра организације), *name* (назив организације), *address* (адреса), *city* (место), *zip* (број поште).

Класа *Blocked_cards* садржи податке о блокираним картицама и разлозима њиховог блокирања. Има атрибуте: *blocked_reason* (разлог блокирања) и *note* (напомена). На основу везе са класом *Users* идентификује се корисник библиотеке.

Класа *Duplicate* садржи податке о дупликатима картица. Има атрибуте: *dup_no* (број дупликата), *dup_date* (датум издавања дупликата). На основу везе са класом *Users* идентификује се корисник библиотеке.

Класа *Warning_types* је шифарник типова опомена. Има атрибуте: *id* (шифра опомене), *name* (назив опомене), *wtext* (текст опомене).



Слика 2.6 Дијаграм класа *Подаци*

Класа *Warn_counters* садржи податке о бројачима бројева опомена за сваку годину. Има атрибуте: *warn_year* (година), *last_no* (последњи број за ту годину). На основу везе са класом *Warning_types* идентификује се тип опомене.

Класа *Warnings* садржи податке о генерисаним опоменама. Има атрибуте: *ctlg_no* (инвентарни број), *wdate* (датум генерисања опомене), *warn_no* (број опомене), *deadline* (рок враћања публикације), *return_date* (датум враћања публикације), *note* (напомена). На основу везе са класама *Users* и *Warning_types* идентификују се корисник библиотеке и тип опомене.

Класа *Archive* представља архиву податка о корисницима. Има атрибуте: *sys_id* (број корисника), *arch_date* (датум архивирања података), *content* (садржај).

2.2.4. Пакет *Manager*

Пакет *Manager* садржи две класе: *UsersManager* и *RecordsManager*.

Класа *UsersManager* управља током податка о корисницима од корисничког интерфејса до базе података и обрнуто. У основи она обавља две врсте активности:

- на захтев класа корисничког интерфејса из пакета *View* класа *UsersManager* добавља и смешта податке из и у базу података преко класа пакета *Proxy*; ти подаци су на клијентској страни привремено смештени у објектима класа из пакета *Model*.

- пребацује податке из модела на кориснички интерфејс и обрнуто.

Класа *RecordsManager* врши комуникацију између система за коришћење библиотечке грађе и библиотечких записа. У систему за коришћење библиотечке грађе постоји потреба за приказом података из библиотечких записа, као и потреба за променом статуса појединачног примерка из задуженог у слободан и обрнуто. Класа *RecordsManager* врши претраживање и добављање библиотечких записа, као и привремено смештање записа у објекте класа пакета *RecordUtils*. Те објекте испоручује класама корисничког интерфејса које их користе за приказ податка из записа. Поред тога класа

RecordsManager врши проверу као и промену статуса примерка у процесу задуживања примерака.

2.2.5 Пакет *Proxy*

На слици 2.7 је приказан дијаграм класа пакета *Proxy*. Овај пакет служи за комуникацију клијентског дела апликације са базом података. Пакет се састоји од једног интерфејса и његове две имплементације.

Интерфејс *Proxy* представља скуп апстрактних операција које су потребне апликацији да би приступила подацима из базе података. То су следеће операције:

- *getUser(userID:String):Users* - за задати број корисника *userID* враћа податке о кориснику са свим задужењима корисника и продужењима чланарине. Ти подаци се смештају у објекте класа из пакета *Model*.

- *saveUser(user:Users):int* - смешта податке о новом или постојећем кориснику у базу података. Подаци се узимају из објекта класа из пакета *Model*, а враћа се број као индикатор успешности извршавања операције. На основу постојања или непостојања вредности атрибута *sys_id* класе *Users* из пакета *Model* се разликује постојећи од новог корисника.

- *getAll(class:Class):List* - за задату класу из пакета *Model* враћа листу свих појава те класе.

- *getGroup(userID):Group* - за задати број корисника *userID* враћа податке о колективном кориснику. Ти подаци се смештају у објекте класа из пакета *Model*.

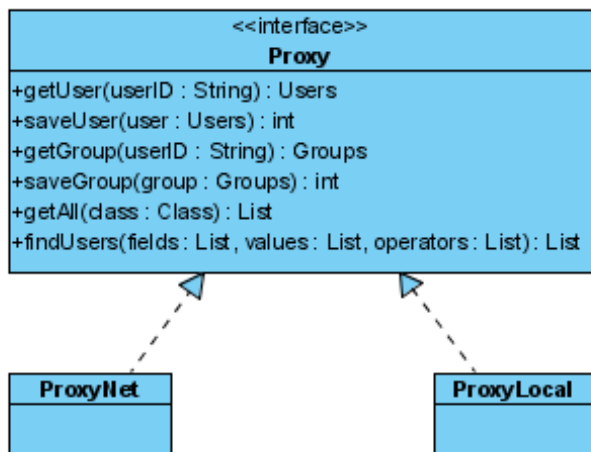
- *saveGroup(group:Group):int* - смешта податке о новом или постојећем колективном кориснику у базу података. Подаци се узимају из објекта класа из пакета *Model*, а враћа се број као индикатор успешности извршавања операције. На основу постојања или непостојања вредности атрибута *sys_id* класе *Group* из пакета *Model* се разликује постојећи од новог корисника.

- *findUsers(fields>List, values>List, operators>List):List* - врши претраживање корисника на основу задате листе атрибута

корисника, листе вредности тих атрибута и листе оператора *and*, *or* и *not* који се стављају између свака два операнда типа (атрибут = вредност).

Класа *ProxyLocal* представља имплементацију интерфејса *Proxy* која се користи у случају када се сервер базе налази у локалном интранету. Имплементација користи перманентну конекцију са базом података преко *JDBC* драјвера.

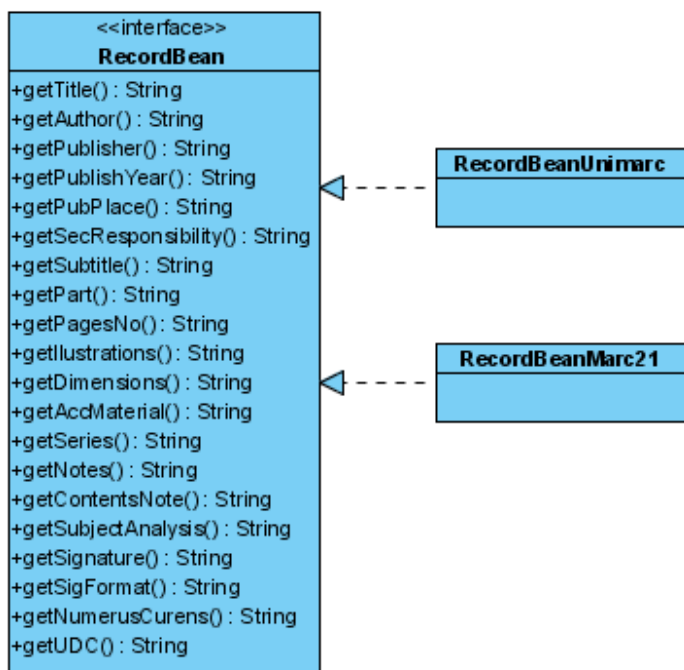
Класа *ProxyNet* представља имплементацију интерфејса *Proxy* која се користи у случају када се сервер базе налази на неком удаљеном серверу коме се приступа путем Интернета. У том случају се не користи перманентна конекција са базом података.



Слика 2.7 Дијаграм класа *Proxy*

2.2.6 Пакет *RecordUtils*

На слици 2.8 је приказан дијаграм класа пакета *RecordUtils*. Овај пакет служи остатку система за добијање одређених садржаја из библиотечког записа који се приказују у систему за коришћење библиотечке грађе. Пакет се састоји од једног интерфејса и његове две имплементације.

Слика 2.8 Дијаграм класа пакета *RecordUtils*

Интерфејс *RecordBean* представља скуп апстрактних операција које враћају одређене садржаје из библиотечког записа. То су:

- *getTitle()* - наслов дела
- *getAuthor()* - аутор дела
- *getPublisher()* - издавач
- *getPublishYear()* - година издавања
- *getPubPlace()* - место издавања
- *getSecResponsibility()* - секундарна одговорност
- *getSubtitle()* – поднаслов дела
- *getPart()* - том књиге
- *getPagesNo()* - број страна
- *getIllustrations()* - број илустрација
- *getDimensions()* - димензије књиге
- *getAccMaterial()* - пропратна грађа

- *getSeries()* – подаци о збирци
- *getNotes()* - напомене
- *getContentsNote()* - напомена о садржају дела
- *getSubjectAnalysis()* - предметне одреднице
- *getSignature()* - сигнатура
- *getSigFormat()* - формат сигнатуре
- *getNumerusCurens()* - *numerus curens*
- *getUDC()* - УДК број

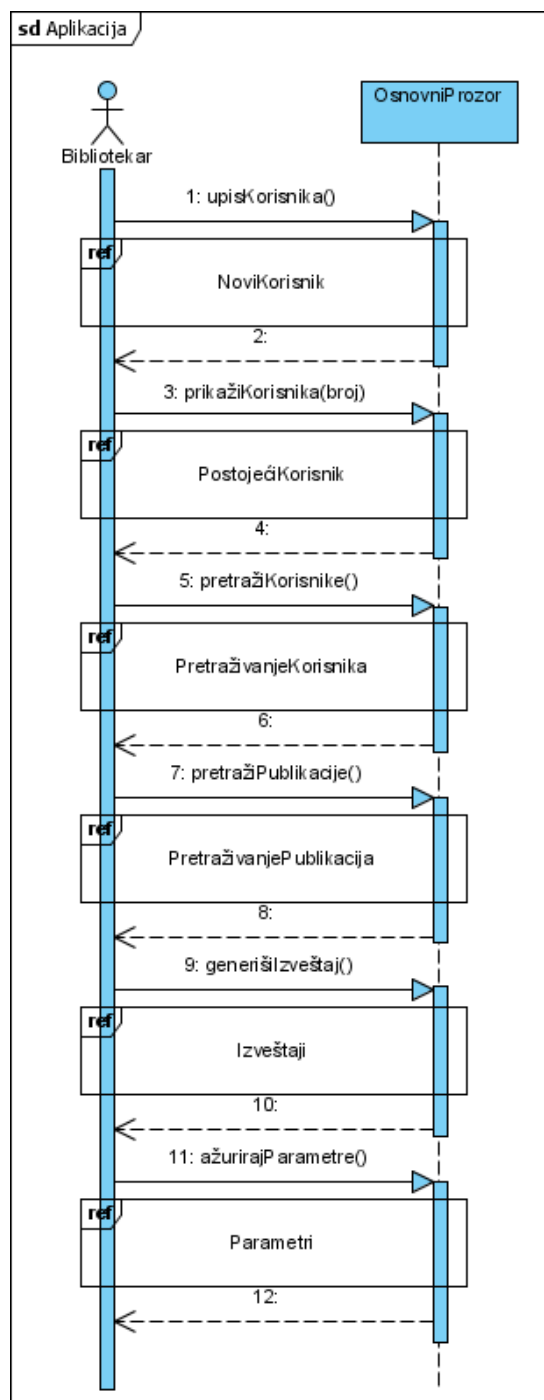
У зависности од библиографског формата ови садржаји су смештени у различитим пољима и потпољима записа. Класа *RecordBeanUnimarc* је имплементација интерфејса *RecordBean* која се користи у случају када се библиотечки записи налазе у *UNIMARC* [Marc] формату, а класа *RecordBeanMarc21* у случају када се записи налазе у формату *MARC21* [Marc].

2.3. Дијаграми секвенци

Динамичка структура система је приказана кроз неколико дијаграма секвенци. Овим дијаграмима нису обухваћени сви сценарији случајева коришћења који су описани у првом одељку овог поглавља, већ само један део. Дијаграми су креирани на основу случајева коришћења приказаних на дијаграму *Рад са корисницима* (слика 2.1) и на основу класа описаних у другом одељку овог поглавља. На сличан начин могу се описати и сценарији за остале случајеве коришћења.

2.3.1 Дијаграм секвенци *Апликација*

На слици 2.9 је приказан дијаграм секвенци *Апликација*. Дијаграм описује комуникацију између библиотекара и система. Објекти који учествују у интеракцији су појаве класа *Bibliotekar* и *OsnovniProzor*. Библиотекар има захтеве за уписом нових корисника, ажурирањем података и задуживањем постојећих корисника, претраживањем корисника и публикација, генерисањем извештаја и подешавањем параметара за рад система.



Слика 2.9 Дијаграм секвенци Арликација

У случају уписа новог корисника библиотекар шаље захтев *upisKorisnika()* објекту *OsnovniProzor*. Позива се дијаграм секвенци *NoviKorisnik*. У случају захтева за ажурирањем података или задуживањем постојећег корисника библиотекар шаље захтев *prikažiKorisnika(broj)* са параметром који представља број корисника, а затим се позива дијаграм *PostojećiKorisnik*.

У случају претраживања корисника или публикација, библиотекар шаље захтев *pretražiKorisnike()* односно *pretražiPublikacije()* који иницира позив дијаграма *PretraživanjeKorisnika* односно *PretraživanjePublikacija*.

У случају захтева за неким извештајем библиотекар шаље захтев *generišiIzveštaj()* који иницира позив дијаграма *Izveštaji*.

У случају захтева за подешавањем параметара система библиотекар шаље захтев *ažurirajParametre()* који иницира позив дијаграма *Parametri*.

У даљем тексту су описани само дијаграми *NoviKorisnik* и *PostojećiKorisnik*. На сличан начин се могу описати и остали дијаграми.

2.3.2 Дијаграм секвенци *NoviKorisnik*

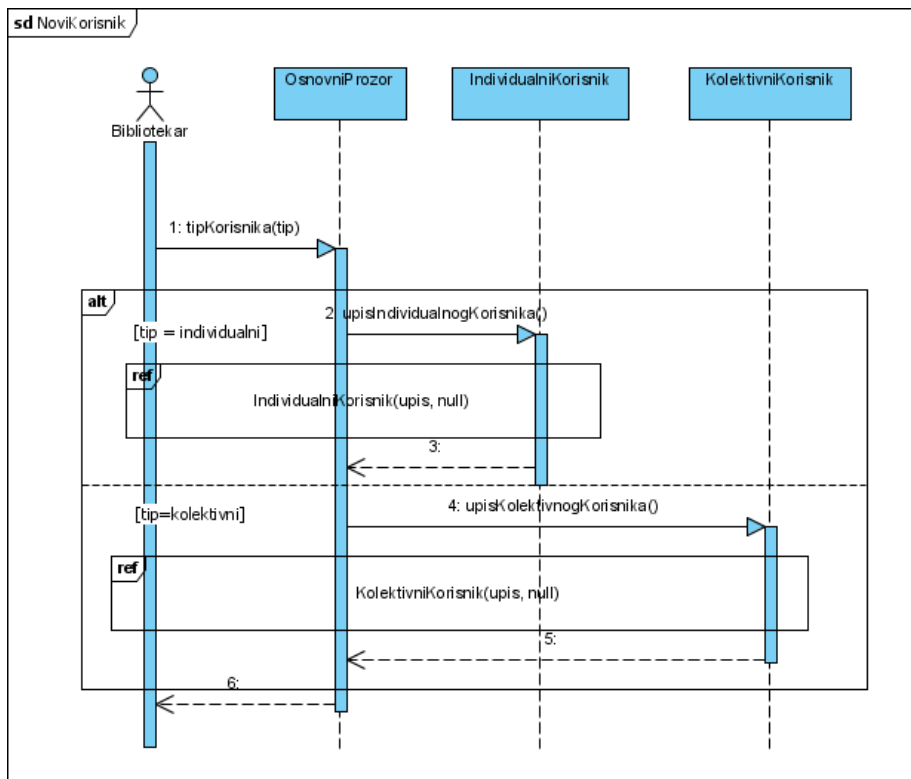
На слици 2.10 је приказан дијаграм секвенци *NoviKorisnik*. Дијаграм описује поступак уписа новог корисника. Објекти који учествују у интеракцији су појаве класа *Bibliotekar*, *OsnovniProzor*, *IndividualniKorisnik* и *KolektivniKorisnik*.

У поступку уписа новог корисника библиотекар прво шаље поруку *tipKorisnika(tip)* објекту *OsnovniProzor* у којој шаље информацију о типу корисника који се уписује. Тип корисника може бити индивидуални или колективни. У зависности од вредности параметра поруке *tip* извршава се једна или друга секвенца порука моделирана комбинованим фрагментом *alt* (алтернатива).

У случају вредности *tip = individualni* извршава се прва алтернатива комбинованог фрагмента. Објект *OsnovniProzor* шаље захтев *upisIndividualnogKorisnika()* објекту класе *IndividualniKorisnik* а

затим се позива дијаграм *IndividualniKorisnik* са параметрима *upis* и *null*.

У случају вредности *tip = kolektivni* извршава се друга алтернатива комбинованог фрагмента. Објекат *OsnovniProzor* шаље захтев *upisKolektivnogKorisnika()* објекту класе *KolektivniKorisnik* а затим се позива дијаграм *KolektivniKorisnik* са параметрима *upis* и *null*.



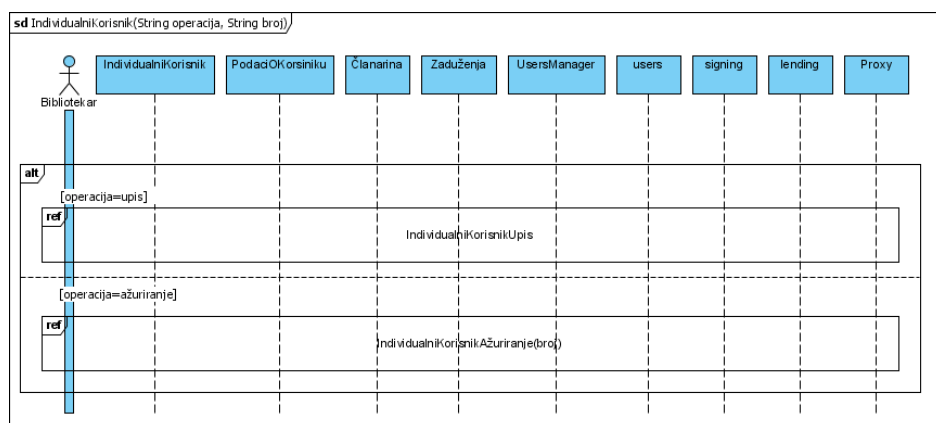
Слика 2.10 Дијаграм секвенци *NoviKorisnik*

2.3.3 Дијаграм секвенци *IndividualniKorisnik*

На слици 2.11 је приказан дијаграм секвенци *IndividualniKorisnik*. Дијаграм описује поступак рада са подацима индивидуалног корисника. Разликују се две основне операције у раду са подацима индивидуалног корисника. То су упис новог корисника и ажурирање података постојећег корисника. Због те две операције дијаграм је параметризован. Први параметар је операција која се

извршава, а други параметар је број корисника који је потребан при извршавању операције ажурирања података. Објекти који учествују у интеракцији су појаве класа *Bibliotekar*, *IndividualniKorisnik*, *PodaciOKorisniku*, *Članarina*, *Zaduženja*, *UsersManager*, *Users*, *Signing*, *Lending* и *Proxy*.

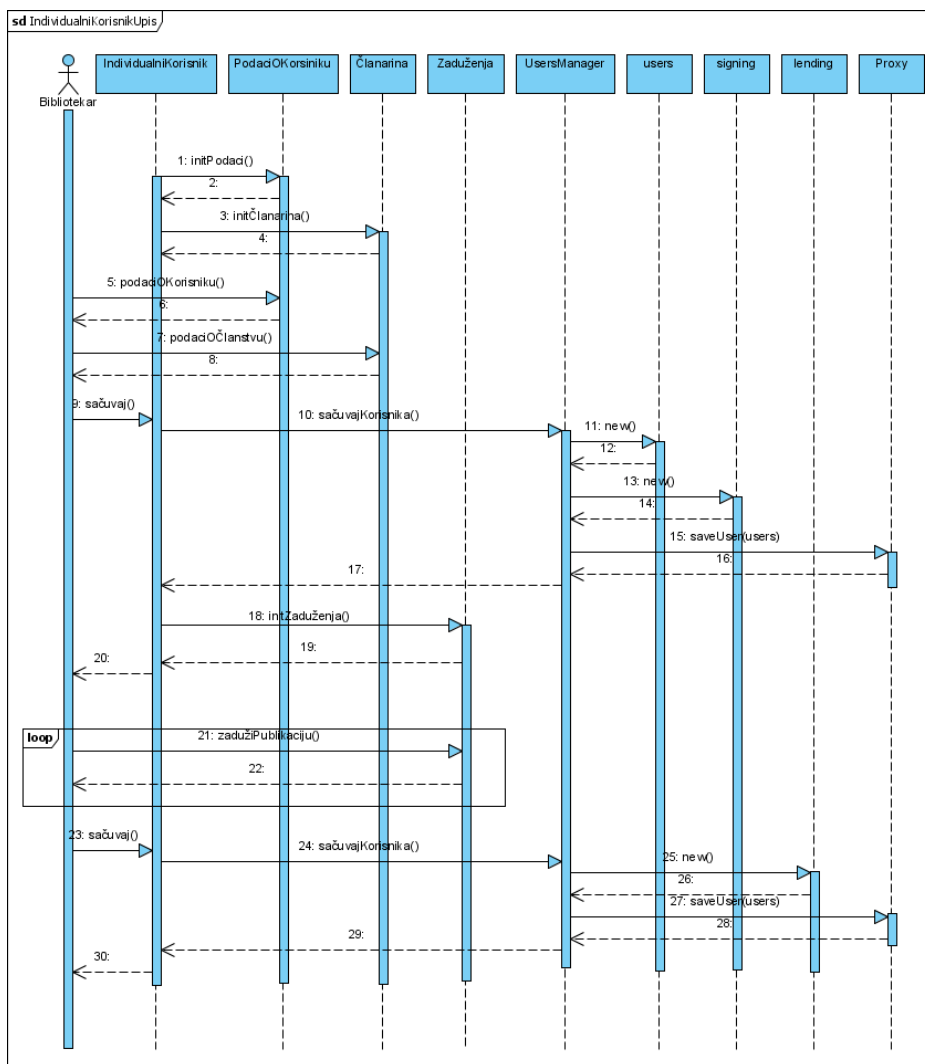
У зависности од вредности прослеђеног параметра *operacija* извршава се једна или друга секвенца порука моделирана комбинованим фрагментом *alt* (алтернатива). У случају вредности *operacija = upis* позива се дијаграм *IndividualniKorisnikUpis*, а у случају вредности *operacija = ažuriranje* позива се дијаграм *IndividualniKorisnikAžuriranje*.



Слика 2.11 Дијаграм секвенци *IndividualniKorisnik*

2.3.4 Дијаграм секвенци *IndividualniKorisnikUpis*

На слици 2.12 је приказан дијаграм *IndividualniKorisnikUpis* који описује поступак уписа новог корисника. Објекти који учествују у интеракцији су исти као и на претходном дијаграму.

Слика 2.12 Дијаграм секвенци *IndividualniKorisnikUpis*

Објекат *IndividualniKorisnik* шаље захтев *initPodaci()* објекту *PodaciOKorisniku* који представља захтев за иницијализацију објекта тј. припрему за упис података новог корисника. Исти такав захтев шаље објекту *Članarina* кроз захтев *initČlanarina()*. Након тога библиотекар у интеракцији са објектом *PodaciOKorisniku* уноси податке о кориснику кроз поруку *podaciOKorisniku()*. Након унесених података библиотекар уноси податке о чланству кроз поруку *podaciOČlanstvu()* у интеракцији са објектом *Članarina*. По завршетку

уноса података библиотекар шаље захтев за чувањем података објекту *IndividualniKorisnik* поруком *sačuvaj()*. Објекат *IndividualniKorisnik* шаље поруку *sačuvajKorisnika()* објекту *UsersManager*. Објекат *UsersManager* порукама *new()* креира објекте *Users* и *Signing* у које смешта податке о кориснику и податке о чланству и даље те објекте кроз референцу на објекат *Users* прослеђује објекту *Proxu* на перманентно чување. Тиме је поступак уписа новог корисника завршен. Након тога објекат *IndividualniKorisnik* шаље захтев за иницијализацију објекту *Zaduženja* у случају да се по упису корисника врши задуживање корисника публикацијама. Библиотекар може да задужи корисника са више публикација, а не мора ни са једном. Тај поступак је измоделиран комбинованим фрагментом *loop* (петља) у ком библиотекар шаље поруку *zadužiPublikaciju()* са подацима о задужењу објекту *Zaduženja*. По завршетку уноса свих задужења библиотекар поново шаље захтев за чувањем података објекту *IndividualniKorisnik* поруком *sačuvaj()* који тај захтев прослеђује објекту *UsersManager* који даље поруком *new()* креира објекат *Lending* у који смешта податке о задужењима и тај објекат прослеђује објекту *Proxu* на перманентно чување.

2.3.5 Дијаграм секвенци *IndividualniKorisnikAžuriranje*

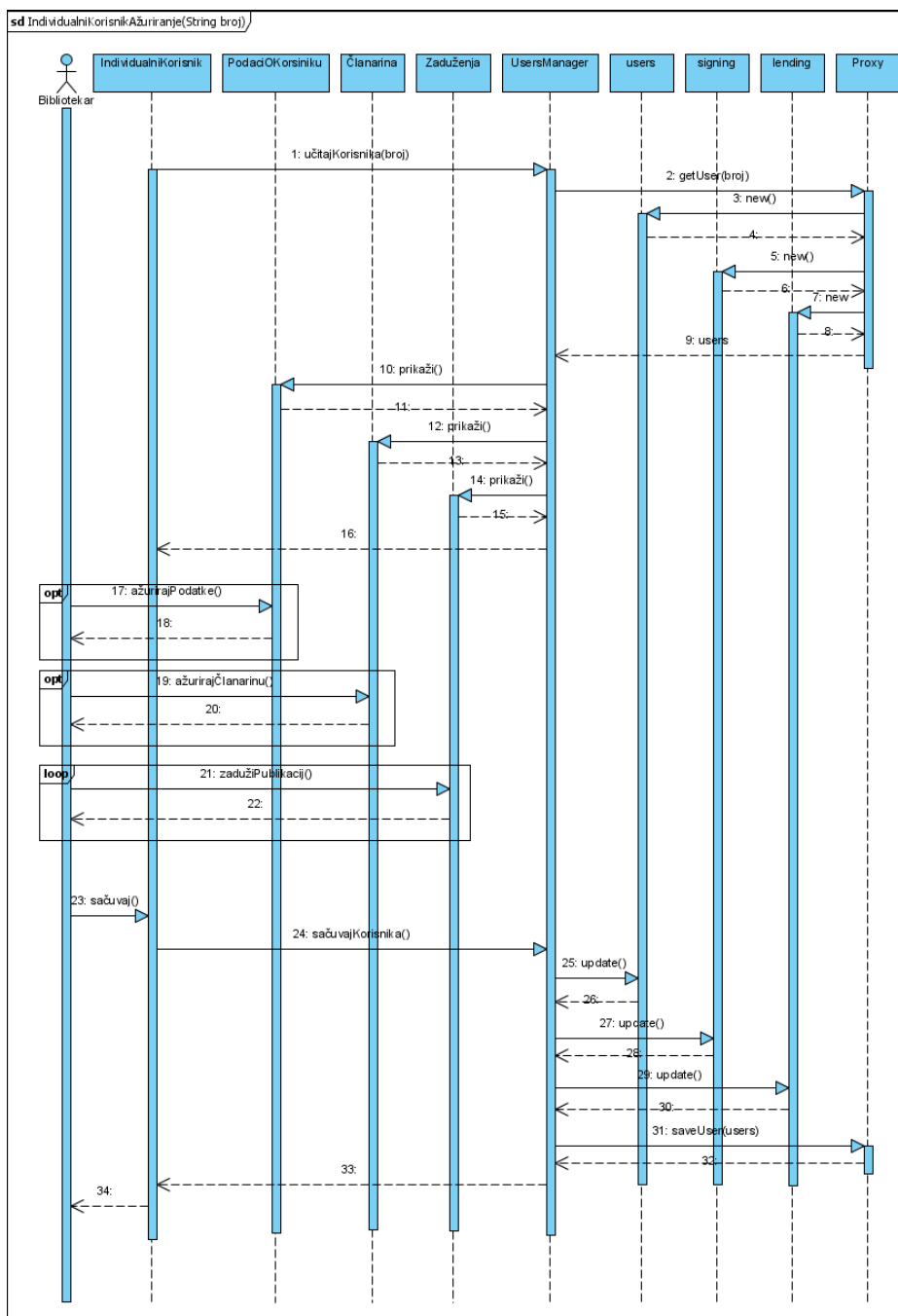
На слици 2.13 је приказан дијаграм *IndividualniKorisnikAžuriranje* који описује поступак ажурирања података постојећег корисника. Објекти који учествују у интеракцији су исти као и на претходна два дијаграма.

Објекат *IndividualniKorisnik* шаље захтев *učitajKorisnika(broj)* објекту *UsersManager* који представља захтев за читавањем података корисника који већ постоји у систему са бројем корисника *broj* који је дијаграму прослеђен као параметар. Објекат *UsersManager* прослеђује захтев објекту *Proxu* који читава податке из базе податка и порукама *new()* креира објекте *Users*, *Signing* и *Lending* у које смешта податке о кориснику, податке о чланству и податке о задужењима корисника, а затим те објекте кроз референцу на објекат *Users* прослеђује објекту *UsersManager*. Објекат *UsersManager* податке из тих објеката порукама *prikaži()* пребацује у објекте

PodaciOKorisniku, *Članarina* и *Zaduženja* чиме се подаци приказују библиотекару. Након тога библиотекар има могућности да ажурира податке о кориснику, да ажурира податке о чланству (продужује чланарину) и ажурира податке о задужењима. Поступак ажурирања података о кориснику је измоделиран комбинованим фрагментом *opt* (опција) где библиотекар у интеракцији са објектом *PodaciOKorisniku* ажурира податке корисника. Поступак ажурирања података о чланству је такође измоделиран комбинованим фрагментом *opt* (опција) где библиотекар у интеракцији са објектом *Članarina* ажурира податке о чланству тј. продужује чланарину кориснику. Ажурирање података о задужењима се одвија на исти начин како је описано претходно за дијаграм *IndividualniKorisnikUpis*. По завршетку ажурирања података библиотекар поново шаље захтев за чувањем података објекту *IndividualniKorisnik* поруком *sačuvaj()* који тај захтев прослеђује објекту *UsersManager*. Објект *UsersManager* порукама *update()* ажурира податке у објектима *Users*, *Signing* и *Lending* и те објекте кроз референцу на објект *Users* прослеђује објекту *Proxy* на перманентно чување.

2.3.6 Дијаграм секвенци *KolektivniKorisnik*

На слици 2.14 је приказан дијаграм секвенци *KolektivniKorisnik*. Дијаграм описује поступак рада са подацима колективног корисника. Као и у случају индивидуалног корисника разликују се две основне операције у раду са подацима колективног корисника: упис новог корисника и ажурирање података постојећег корисника. Због тога је и овај дијаграм параметризован. Први параметар је операција која се извршава, а други параметар је број корисника који је потребан при извршавању операције ажурирања података. Објекти који учествују у интеракцији су појаве класа *Bibliotekar*, *KolektivniKorisnik*, *UsersManager*, *Groups* и *Proxy*.


 Слика 2.13 Дијаграм секвенци *IndividualniKorisnikAžuriranje*

У случају вредности *operacija* = *upis* објекат *KolektivniKorisnik* се иницијализује тј. припрема за упис података новог корисника. Након тога библиотекар у интеракцији са објектом уноси податке о кориснику кроз поруку *podaci()*. По завршетку уноса података објекат податке које садржи шаље објекту *UsersManager* који даље управља процесом чувања података. Објекат *UsersManager* поруком *new()* креира објекат *Groups* у који смешта податке и даље тај објекат прослеђује објекту *Proxu* на перманентно чување.

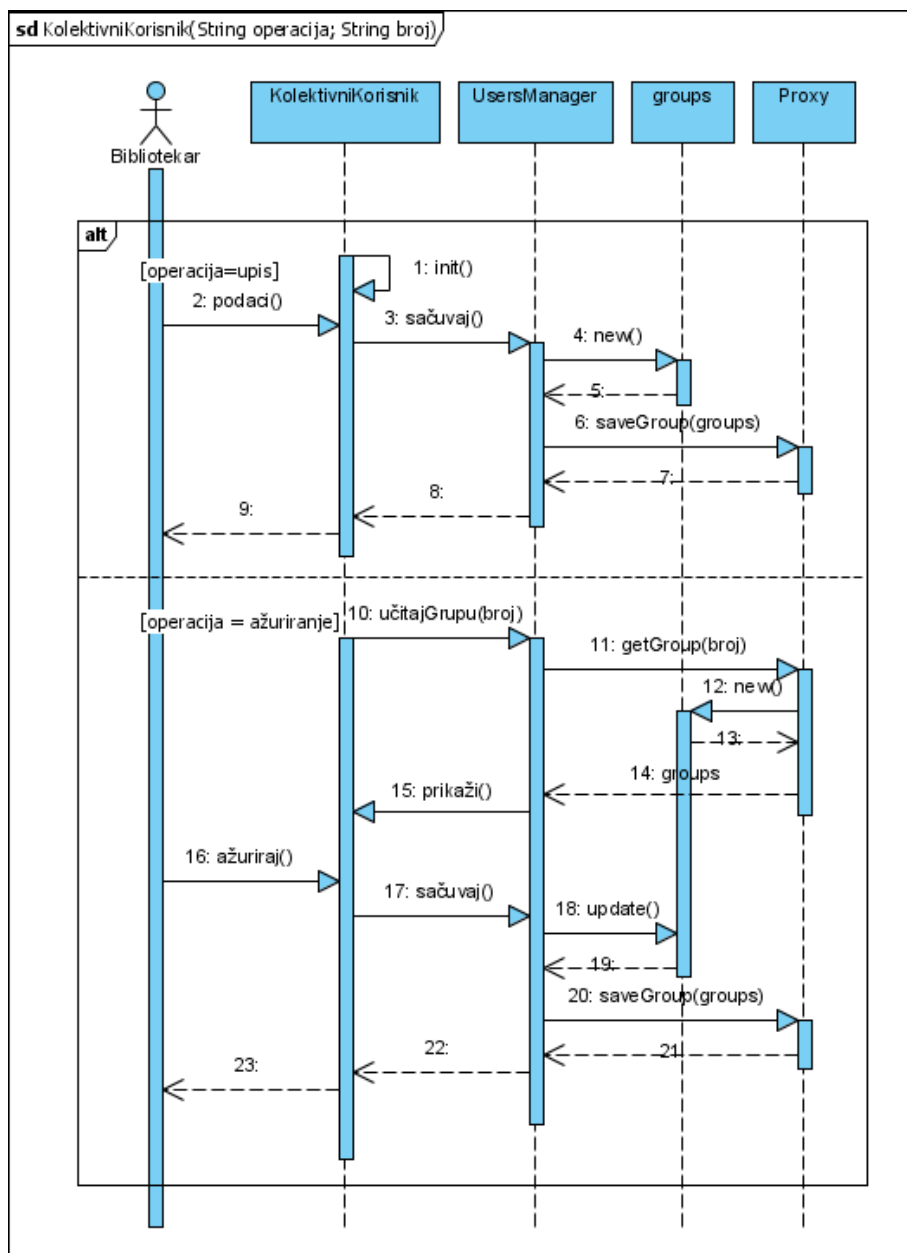
У случају вредности *operacija* = *ažuriranje* објекат *KolektivniKorisnik* учитава податке корисника који већ постоји у систему са бројем корисника *broj* који је дијаграму прослеђен као параметар. Објекат шаље захтев објекту *UsersManager* са бројем корисника чије податке захтева. Објекат *UsersManager* тај захтев прослеђује објекту *Proxu* који учитава податке из базе података, поруком *new()* креира објекат *Groups* у који смешта податке и враћа референцу на тај објекат објекту *UsersManager*. Објекат *UsersManager* те податке из објекта *Groups* поруком *prikaži()* пребацује у објекат *KolektivniKorisnik* чиме се подаци приказују библиотекар у интеракцији са објектом *KolektivniKorisnik* ажурира податке, а затим тај објекат шаље захтев за изменом података објекту *UsersManager* кроз поруку *sačuvaj()*. Објекат *UsersManager* поруком *update()* ажурира податке у објекту *Groups* и тај објекат прослеђује објекту *Proxu* на перманентно чување.

2.3.7 Дијаграм секвенци *PostojećiKorisnik*

На слици 2.15 је приказан дијаграм секвенци *PostojećiKorisnik*. Дијаграм описује поступак ажурирања података постојећег корисника. Објекти који учествују у интеракцији су појаве класа *Bibliotekar*, *OsnovniProzor*, *IndividualniKorisnik*, *KolektivniKorisnik*, *UsersManager* и *Proxu*.

У поступку ажурирања података постојећег корисника библиотекар прво шаље поруку *prikažiKorisnika(broj)* објекту *OsnovniProzor* у којој шаље број траженог корисника. Затим објекат *OsnovniProzor* прослеђује тај захтев објекту *UsersManager* који у интеракцији са објектом *Proxu* проверава ког је типа тражени корисник и у зависности од тога поставља вредности параметара

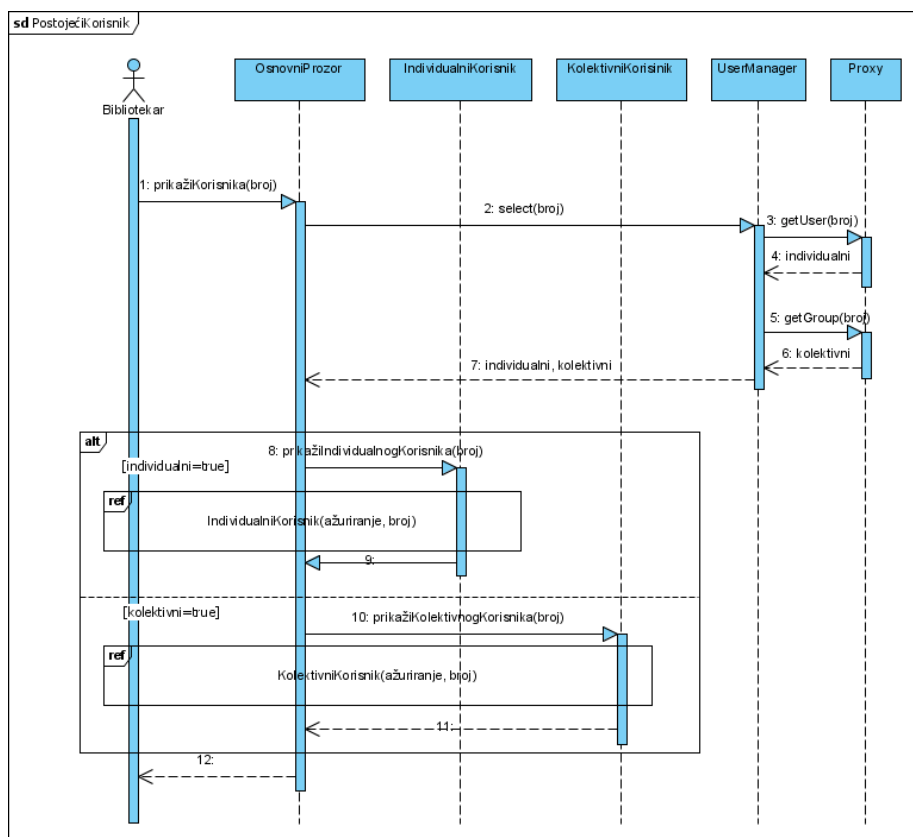
individualni = true или *kolektivni = true*. У зависности од вредности тих параметара извршава се једна или друга секвенца порука моделирана комбинованим фрагментом *alt* (алтернатива).



Слика 2.14 Дијаграм секвенци *KolektivniKorisnik*

У случају вредности *individualni* = *true* извршава се прва алтернатива комбинованог фрагмента. Објекат *OsnovniProzor* шаље захтев *prikažiIndividualnogKorisnika()* објекту класе *IndividualniKorisnik* а затим се позива дијаграм *IndividualniKorisnik* са параметрима *ažuriranje* и *broj*.

У случају вредности *kolektivni* = *true* извршава се друга алтернатива комбинованог фрагмента. Објекат *OsnovniProzor* шаље захтев *prikažiKolektivnogKorisnika()* објекту класе *KolektivniKorisnik* а затим се позива дијаграм *KolektivniKorisnik* са параметрима *ažuriranje* и *broj*.



Слика 2.15 Дијаграм секвенци *PostojećiKorisnik*

Имплементација система за коришћење библиотечке грађе

На основу спецификације описане у претходном поглављу урађена је имплементација система за коришћење библиотечке грађе. Систем је имплементиран у *Java* окружењу. Као развојни алат је коришћено окружење *Eclipse v3.2*. Систем се састоји из три дела: клијентске апликације, серверског дела апликације и базе података.

Клијентска апликација *Циркулација* је имплементирана као класична десктоп апликација чије извршавање се одвија у потпуности на клијенту. Такав начин имплементације је одабран првенствено због функционалности и брзине екранских форми што је био један од основних захтева библиотека. Екранске форме корисничког интерфејса су имплементирани помоћу *Swing* пакета. Валидација екранских форми је урађена помоћу *Jakarta Commons Validator* пакета.

Поред екранских форми апликација садржи и имплементацију објектног модела базе података преко кога остатак апликације приступа подацима из базе података. Део апликације који је задужен за комуникацију између тог објектног модела и базе података је реализован помоћу *ORM (Object/Relational Mapping)* технологије, а за имплементацију је коришћен *Hibernate* пакет.

Клијентска апликација има два начина комуникације са базом података. Један начин је комуникација преко *JDBC* драјвера и тај начин комуникације се користи у случају када се сервер базе налази у локалном интранету. Други начин комуникације се користи у случају када се сервер базе налази на удаљеном серверу и у том

случају клијентска апликација преко *HTTP* протокола шаље податке серверском делу апликације који даље комуницира са базом података. Оваква два начина комуникације омогућавају рад клијента и у случају кад се база података налази у локалном интранету и у случају кад се налази на удаљеном серверу, чиме систем постаје независан од мрежне архитектуре библиотеке.

Шема базе података је направљена на основу дијаграма класа *Подаци* приказаног на слици 2.6 у другом поглављу овог рада. Свака класа са дијаграма одговара једној табели шеме базе података. Сва обележја класа су пресликана у одговарајуће колоне табела. На основу веза међу класама су направљени одговарајући референцијални интегритети међу табелама. Као сервер базе података се користи *MySQL Enterprise Server 5.0*.

У даљем тексту је дат кратак опис коришћених технологија.

3.1 Софтверско окружење

Eclipse Foundation Inc. је непрофитна организација чији пројекти нуде *open source* платформу и *framework*-е за развој софтвера. Један од пројеката је ***Eclipse Platform*** који представља развојно окружење за моделирање и развој софтвера. Приликом израде система за коришћење библиотечке грађе је коришћења верзија окружења 3.2. Окружење је имплементирано у *Java* програмском језику тако да може да ради на различитим оперативним системима и нуди *plug-in* базиран *framework* који омогућава проширење основног пакета додатним алатима имплементираним као *plug-in*-и за платформу. Окружење служи за развој апликација у *Java*, *C++* и *PHP* програмским језицима. Остали пројекти *Eclipse Foundation Inc.* нуде различите додатне алате као *plug-in*-е за платформу. Приликом израде екранских форми система за коришћење библиотечке грађе је коришћен *plug-in Visual Editor* као графички едитор за *Swing* компоненте.

ORM (Object/Relational Mapping) технологија представља начин мапирања објекто-оријентисане логике апликације на релациони модел базе података. ***Hibernate*** пакет је једна од имплементација *ORM* технологије. *Hibernate* је тренутно најбоље решење које постоји у тој области и зато је изабран за израду дела система за

коришћење библиотечке грађе. За израду система је коришћена верзија *Hiberanate 3.2*. *Hiberanate* омогућава како мапирање класа апликације на табеле базе података тако и перзистенцију тих класа. Највећа предност *Hiberanate* пакета је у томе што су мапирање и перзистенција одвојени, а и они заједно могу да се одвоје од остатка апликације. Информације о мапирању стоје у одвојеним *XML* документима, а мапирају се чисте *Java* класе које не зависе од *Hiberanate* пакета. Перзистенција података се врши помоћу *Hiberanate* пакета, а при томе се користе информације о мапирању и мапиране *Java* класе. На тај начин се добија могућност одвајања дела логике апликације од дела који се бави перзистенцијом података.

Commons Validator је програмски пакет развијен у оквиру *The Apache Jakarta Project*-а. Служи за валидацију података како у документима тако и у објектима у оквиру апликације. У случају система за коришћење библиотечке грађе искоришћен је за валидацију уноса података преко екранских форми. Дефиниција података која се користи при валидацији је смештена у *XML* документу и самим тим погодна за измене. На тај начин се обавезност уноса података и врста података може одвојити од саме имплементације апликације и прилагодити конкретним потребама.

MySQL Enterprise Server је *open-source* база податка, тренутно врло заступљена у свету. Брза је, пружа добре перформансе у раду и лака за инсталацију и одржавање. Постоје верзије за *Linux*, *UNIX* и *Windows* оперативне системе.

3.2 Имплементација корисничког интерфејса

На дијаграму случајева коришћња *Рад са корисницима* описаном у одељку 2.1.1 су приказане све функционалности које библиотекар обавља у интеракцији са системом да би услужио корисника. Функционалности описане на том дијаграму су пренесене на екранске форме корисничког интерфејса преко којих библиотекар комуницира са системом.

На слици 4.2 из одељка 4.1 је приказана основна екранска форма апликације која се отвара приликом покретања апликације. Ова екранска форма представља имплементацију класе *OsnovniProzor* са дијаграма класа *Кориснички интерфејс* приказаног на слици 2.5 и

описаног у одељку 2.2.2. Основна екранска форма садржи главни мени апликације и *toolbar* апликације. Главни мени је подељен на пет делова: *Obrada*, *Korisnici*, *Pretraživanje*, *Izveštaji* и *Sistem*. Мени *Obrada* односи се на обраду библиографске грађе, што није предмет ове монографије. Мени *Korisnici* се састоји од два подменија: *Novi* и *Postojeći*. У подменију *Novi* се налазе акције везане за упис новог корисника. Подмени *Novi* има две акције *Individualni* и *Kolektivni*. Акција *Individualni* иницира функцију уписа новог индивидуалног корисника, а акција *Kolektivni* новог колективног корисника. Овај подмени представља реализацију финкционалности случаја коришћења *Упис новог корисника* приказаног на дијаграму *Рад са корисницима*. Подмени *Postojeći* садржи акције везане за рад са постојећим корисником у систему. То су: *Podaci*, *Članarina* и *Zaduženja*. Преко тих акција се иницирају функције промене података корисника, продужења чланарине и задуживања и раздуживања корисника. Овај подмени представља реализацију функционалности случаја коришћења *Постојећи корисник*. У менију *Pretraživanje* се налазе две акције: *Korisnici* и *Publikacije*. Акција *Korisnici* иницира функцију претраживања корисника, а акција *Publikacije* претраживање публикација. У менију *Izveštaji* се налазе акције којима се покреће генерисање извештаја који постоје у систему, а у менију *Sistem* акције везане за администрацију система и ажурирање шифарника. На *toolbar*-у апликације се налазе пречице до најчешће коришћених акција из менија (описане у следећем поглављу).

Избором акције уписа новог индивидуалног корисника отвара се екранска форма приказана на слици 4.6 у одељку 4.2.3 (са празним пољима за унос података). Ова екранска форма представља имплементацију класе *IndividualniKorisnik* са дијаграма класа *Кориснички интерфејс* и на њој се реализује финкционалност случаја коришћења *Индивидуални корисик* са дијаграма *Рад са корисницима*. Екранска форма садржи четири картице: *Osnovni podaci*, *Dodatni podaci*, *Članarina* и *Zaduženja*. Изглед картице *Osnovni podaci* је приказан на већ поменутој слици 4.6, а изглед осталих картица је приказан на сликама 4.7 (одељак 4.2.4), 4.8 (одељак 4.2.5) и 4.9 (одељак 4.2.6). Картице *Osnovni podaci* и *Dodatni podaci* представљају имплементацију класе *PodaciOKorisniku* са дијаграма класа *Кориснички интерфејс* и на њима је реализована

функционалност случаја коришћења *Подаци о кориснику*. Садржај картице *Članarina* представља имплементацију класе *Članarina* и реализује функционалност случаја коришћења *Подаци о чланству*, а садржај картице *Zaduženja* имплементацију класе *Zaduženja* и функционалност случаја коришћења *Задуживање и раздуживање*.

На картицама *Osnovni podaci* и *Dodatni podaci* се уносе подаци о кориснику. Дугме *Štampaj* на картици *Osnovni podaci* представља реализацију случаја коришћења *Штампање картице*. На картици *Članarina* се уносе број корисника и остали подаци о чланству у библиотеку. Пошто се иста екранска форма користи и за новог и за постојећег корисника на овој картици је могуће и продужити чланарину кориснику, као и одштампати признаницу. На картици *Zaduženja* се врши задуживање и раздуживање корисника. Видљива је табела са задуженим публикацијама. Над сваком публикацијом у табели је могуће урадити две акције: раздуживање и продуживање задужења. Те акције се иницирају са два дугмета испод табеле. Публикације је могуће задужити на два начина: уношењем инвентарног броја у текстуално поље изнад табеле и потврдом тог уноса и претраживањем библиотечког фонда које води на екранску форму на којој се реализује функционалност случаја коришћења *Претраживање публикација*. Дугме *Revers* реализује случај коришћења *Штампање реверса*, а дугме *Istorija* случај коришћења *Штампање историје задуживања*. Поред тога је видљив и кратак инфо о кориснику са упозорењима о истеку чланарине и посланим опоменама. При дну екранске форме се налазе дугме *Sačuvaj* које иницира акцију перманентног чувања података и дугме *Odustani* које занемарује унесене податке и затвара екранску форму за унос новог корисника.

Рад са подацима постојећег индивидуалног корисника је сличан као унос података за новог корисника. Користе се исте екранске форме као за унос новог корисника које су претходно описане. Избором у менију једне од три могуће акције за рад са постојећим корисником (*Podaci*, *Članarina* и *Zaduženja*) отвара се екранска форма за унос броја корисника приказана на слици 4.5 у одељку 4.2.2. Након уноса броја у текстуално поље и потврде тог уноса на дугме ОК отвара се одговарајућа екранска форма са слика 4.6, 4.8 или 4.9 у зависности од одабране акције и у случају да унесени број припада

индивидуалном кориснику. У случају да број припада колективном кориснику отвара се екранска форма за колективног корисника (описана даље у тексту) која представља реализацију случаја коришћења *Подаци о колективном кориснику*. Претходно описане акције представљају реализацију случаја коришћења *Постојећи корисник*. У случају да библиотекар не зна број корисника чије податке жели да види на екранској форми приказаној на слици 4.5 постоји дугме *Нађи* које води на екранску форму за претраживање корисника које је приказано касније у овом одељку.

Ако се у менију изабере акција уписа новог колективног корисника отвара се екранска форма на којој се реализују функционалности случајева коришћења *Колективни корисник* и *Подаци о колективном кориснику* и која представља имплементацију класе *KolektivniKorisnik*. Та екранска форма је приказана на слици 4.10 у одељку 4.3.1. На форми се уноси скуп података за колективног корисника. При дну екранске форме се налазе дугме *Sačivaј* које иницира акцију пермаментног чувања података и дугме *Odustani* које занемарује унесене податке и затвара екранску форму за унос новог корисника. Ова екранска форма се користи и за промену података колективног корисника.

Избором акције претраживања корисника отвара се екранска форма приказана на слици 4.11 у одељку 4.4. Ова екранска форма представља имплементацију класе *PretraživanjeKorisnika* и реализацију функционалности случаја коришћења *Претраживање корисника*. На форми се задају критеријуми претраживања. Може се претраживати по свим пољима која су дефинисана у класи *Users* која је приказана на дијаграму *Подаци* на слици 2.6 у одељку 2.2.3. Критеријуми се могу укрштати између себе са операторима *and*, *or* и *not*. Може да се зада до пет критеријума типа "поље = вредност поља". Дугме са знаком "..." иницира акцију избора поља по ком се претражује. Назив изабраног поља је видљив са леве стране дугмета. Са десне стране дугмета се налази текстуално поље у које се задаје тражена вредност изабраног поља класе *Users*. Уколико тражено поље узима вредности из неког од шифарника повезаних са класом *Users* уместо текстуалног поља за унос вредности појављује се падајућа листа из које се бира једна од постојећих вредности тог шифарника. Поред тога, на ове критеријуме је могуће додати и

временски период за датуме који постоје у класама *Signing* и *Lending* а то су: датум задужења, датум раздужења, рок враћања, датум продужења чланства и датум до кад важи чланство, као и локацију за сваку од тих акција извршених у том временском периоду.

На слици 4.12 у одељку 4.4 је дата екранска форма на којој се приказују резултати претраживања корисника. Форма представља имплементацију класе *RezultatiPretraživanjaKorisnika* и реализацију функционалности случаја коришћења *Приказ резултата претраживања корисника*. Пронађени корисници на основу задатог критеријума приказују се у табели. Дугме *Prikaži* отвара екранске форме за рад са подацима одабраног корисника које су већ претходно описане и представља реализацију случаја коришћења *Постојећи корисник*. Дугме *Štampaj* штампа листу пронађених корисника.

Избором акције претраживања публикација отвара се екранска форма приказана на слици 4.13 у одељку 4.5. Ова екранска форма представља имплементацију класе *PretraživanjePublikacija* и реализацију функционалности случаја коришћења *Претраживање публикација*. На форми се задају критеријуми претраживања. Претраживање се врши по понуђеним префиксима текст сервера и може да се зада до пет критеријума који међу собом могу да се укрсте операторима *and*, *or* и *not*. Ти префикси се односе на садржај библиотечког записа. Екранска форма за избор префикса се добија на тастер F9. На ове критеријуме је могуће додати и временски период за датум задуживања и датум раздуживања публикације, као и локацију за сваку од тих акција извршених у том временском периоду.

Екранска форма за приказ резултата претраживања је приказана на слици 4.14 у одељку 4.5. Екранска форма представља имплементацију класе *RezultatiPretraživanjaPublikacija* и реализацију функционалности случаја коришћења *Приказ резултата претраживања публикација*. На левој половини форме се приказује листа резултата. У резултате спадају библиотечки записи који су задовољили критеријуме претраживања. За сваки запис у листи се приказује аутор, наслов, издавач и година издања. Двокликом на сваки елемент листе се отвара подлиста за тај елемент која садржи инвентарне бројеве примерака који припадају том запису и статусе

тих примерака. Примерак који је задужен могуће је раздужити дугметом са сликом  и то представља реализацију случаја коришћења *Раздуживање публикације*. Такође за сваки задужени примерак се у десном горњем углу форме приказује број, име и презиме корисника код кога је задужен. Дугме *Prikaži* отвара екранске форме за рад са подацима тог корисника и представља реализацију случаја коришћења *Постојећи корисник*. Примерке који су задуживи је могуће задужити. Дугметом са сликом  се иницира реализација случаја коришћења *Задуживање и раздуживање* којој претходи унос броја корисника који се задужује. На десној страни екранске форме се приказују детаљи библиотечког записа одабраног у листи погодака. Детаљи се приказују на два начина: у облику кратког инфа као сто се види на слици 4.14. или у облику каталошког листића. Жељени начин приказа постаје видљив одабиром картице *Info* или *Listić*.

Претходно описане екранске форме представљају интерфејс за реализацију функционалности случајева коришћења са дијаграма *Рад са корисницима*. Поред ових у систему су имплементиране и екранске форме које представљају интерфејсе за функционалности дефинисане случајевима коришћења приказаним на два преостала дијаграма описана у другом поглављу, а то су *Извештавање* и *Параметри система*. Те форме такође представљају и имплементацију преосталих класа са дијаграма *Кориснички интерфејс*.

3.2 Имплементација ажурирања базе података

Део апликације који је задужен за рад са подацима из базе података је имплементиран на основу спецификације пакета *Model* (одељак 2.2.3), пакета *Proxy* (одељак 2.2.5) и класе *UsersManager* (одељак 2.2.4). Имплементацију спецификације пакета *Model* и *Proxy* представљају *Java* пакети *Model* и *Proxy* са својим класама, а имплементацију класе *UsersManager Java* класа *UsersManager* која се налази унутар *Java* пакета *Manager*.

3.2.1 Имплементација објектног модела базе података

Имплементацију објектног модела базе података представљају *Java* класе пакета *Model*. Имплементација је урађена на основу спецификације класа описаних у одељку 2.2.3 и приказаних на дијаграму класа *Подаци* на слици 2.6. Свакој класи са дијаграма одговара једна *Java* класа која има исто име, нпр. класи *Users* са дијаграма одговара *Java* класа *Users.java*.

Сви атрибути специфицирани у класама на дијаграму постоје и у имплементацији. *Java* класе су имплементирани по *JavaBeans* стандарду што значи да за сваки атрибут *xxx* постоје методе *setXxx()* и *getXxx()*. Методе *setXxx()* постављају вредности атрибута и као параметар им се прослеђује та вредност, а *getXxx()* враћају вредности атрибута.

Објекти који су појаве класа, поред вредности атрибута имају и референце на друге објекте с којима су у вези. Тим референцама су имплементирани асоцијације које постоје на дијаграму класа *Подаци*. Тако на пример, асоцијација 1:N између класа *Users* и *Lending* је имплементирана тако да објекат класе *Users* поседује листу референци на објекте класе *Lending* с којима је у вези, а објекти класе *Lending* поседују референцу на одговарајући објекат класе *Users*. Класа *Users* има методе: *getLendings()*, која враћа листу референци на објекте класе *Lending*, *setLendings()*, којом се поставља вредност за листу референци и *addLending()*, којом се додаје један објекат класе *Lending* у већ постојећу листу. Слично, класа *Lending* има методе: *getUsers()* која враћа референцу на одговарајући објекат класе *Users* и *setUsers()* која поставља вредност референце. На исти начин су имплементирани и остале асоцијације са дијаграма *Подаци*.

3.2.2 Имплементација комуникације са базом података

Део апликације који је задужен за комуникацију између претходно описаног објектног модела базе и физичке базе података је реализован помоћу *ORM (Object/Relational Mapping)* технологије, а за имплементацију је коришћен *Hibernate* пакет. Овај део апликације представља имплементацију пакета *Proxy* описаног у одељку 2.2.5 и спакован је у *Java* пакет који се такође зове *Proxy*.

Класе и интерфејс приказани на дијаграму класа пакета *Proxy* на слици 2.7 у одељку 2.2.5 имају своју имплементацију у облику *Java* класа и интерфејса и то су: интерфејс *Proxy.java*, класа *ProxyLocal.java* и класа *ProxyNet.java*. Поред тога у *Java* пакету се налазе и фајлови потребни *Hibernate* пакету за чување информација о мапирању објектног модела базе на физичку базу података.

3.2.2.1 Мапирање базе података

Hibernate пакет захтева да се информације о мапирању класа и атрибута класа објектног модела на табеле и поља базе података чувају у *XML* документима. За сваку класу пакета *Model* постоји један *XML* документ са именом *ImeKlase.hbm.xml* у ком се налазе информације о мапирању те класе на одговарајућу табелу базе података. Пример једног таквог документа за класу *Signing* је дат на листингу 3.1.

Елементом *class* је дефинисано да се класа *Signing* мапира на табелу базе *signing*. Елементима *property* је за сваки атрибут класе дефинисано на које поље табеле се мапира, која је дужина поља у табели и ког је типа. Елементима *many-to-one* су везе између класа мапиране на референцијалне интегритете. Елементом *id* се дефинише који атрибут се мапира на примарни кључ табеле. Слично овом примеру су измапиране и остале класе пакета *Model*.

3.2.2.2 Приступ бази података у локалном интранету

Класа *ProxyLocal.java* представља имплементацију интерфејса *Proxy.java* која се користи за приступ бази података у случају када се база налази у локалном интранету. Класа имплементира све методе интерфејса које су дефинисане и описане у одељку 2.2.5, као и још неке помоћне за којима се јавила потреба приликом имплементације, а које нису од суштинског значаја. Имплементација за комуникацију са базом користи *JDBC* драјвер, што значи да постоји перманентна конекција са базом податка. Из тог разлога ова имплементација се користи за локални интранет.

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//
    Hibernate/Hibernate Mapping DTD 3.0//EN"
    "http://hibernate.sourceforge.net/hibernate-
    mapping-3.0.dtd">
<hibernate-mapping
    package="com.gint.app.bisis35.client.circ.model">
  <class name="Signing" table="signing">
    <id name="id" type="int">
      <column name="id" />
      <generator class="native" />
    </id>
    <many-to-one name="location" class="Location">
      <column name="location" not-null="true">
      </column>
    </many-to-one>
    <many-to-one name="users" class="Users">
      <column name="sys_id" not-null="true">
      </column>
    </many-to-one>
    <many-to-one name="librarian"
      class="Librarian">
      <column name="librarian" not-null="true">
      </column>
    </many-to-one>
    <property name="signDate" type="timestamp">
      <column name="sign_date" length="19" not-
        null="true">
      </column>
    </property>
    <property name="untilDate" type="timestamp">
      <column name="until_date" length="19">
      </column>
    </property>
    <property name="cost" type="double">
      <column name="cost" precision="10"
        scale="0">
      </column>
    </property>
    <property name="receiptId" type="string">
      <column name="receipt_id">
      </column>
    </property>
  </class>
</hibernate-mapping>
```

Листинг 3.1 Мапирање класе *Signing* на табелу *signing*

На листингу 3.2 је дат пример имплементације методе *getUser* класе *ProxyLocal.java*. Овај пример илуструје употребу *Hibernate* пакета за имплементацију метода класе *ProxyLocal.java*.

```
...
private SessionFactory sessions;
...
static{
    sessions = new
        Configuration().buildSessionFactory();
}
...
public Users getUser(String userID){
    Session session = sessions.openSession();
    Transaction tx = session.beginTransaction();
    Query q = session.createQuery("from Users u where
        u.userId = :userid");
    q.setString("userid", userID);
    Users user = (Users)q.uniqueResult();
    tx.commit();
    session.close();
    return user;
}
```

Листинг 3.2 Имплементације методе *getUser*

Приликом иницијализације класе креира се објекат типа *SessionFactory*. *SessionFactory* је интерфејс *Hibernate* пакета који управља конекцијама са базом података и кешом који се може користити у више сесија. Објекат се креира на основу конфигурационог фајла. У случају овог локалног интерфејса у конфигурационом фајлу су наведена подешавања за креирање конекције са базом података преко *JDBC* драјвера. Приликом позива методе *getUser* за задати број корисника, отвара се нова сесија помоћу постојећег објекта типа *SessionFactory*, а затим и трансакција. Унутар једне сесије може да се изврши више трансакција. Унутар трансакције се креирају и извршавају упити. Упити се могу креирати у *HQL* језику над објектним моделом базе података (како је и приказано у примеру на листингу 3.2), у *SQL* језику над табелама базе података или помоћу интерфејса *Hibernate* пакета *Criteria*. Након извршавања упита затварају се трансакција и сесија.

На сличан начин се ради и ажурирање базе података. Као што се у примеру види креирање упита се врши позивањем методе `createQuery` над објектом типа `Session`, а извршавање упита методом `uniqueResult` (ако резултат извршавања није јединствен користи се метода `list`). Ажурирање базе података се врши позивањем метода `save`, `saveOrUpdate` и `update` над објектом типа `Session`. Тим методама се као параметар прослеђују објекти класа пакета `Model`. Поступак ажурирања је сличан као у наведеном примеру: отвара се сесија, затим трансакција, позивају се наведене методе за ажурирање података, а затим се затварају трансакција и сесија.

Приликом извршавања упита као резултат се добија објекат одговарајуће класе пакета `Model`. Такође, приликом ажурирања базе података методама се прослеђују објекти класа пакета `Model`. *Hibernate*, на основу информација о мапирању објектног модела на базу података, које је описано у претходном одељку, зна над којим табелама да извршава упите и који атрибути објеката одговарају којим пољима табела базе података.

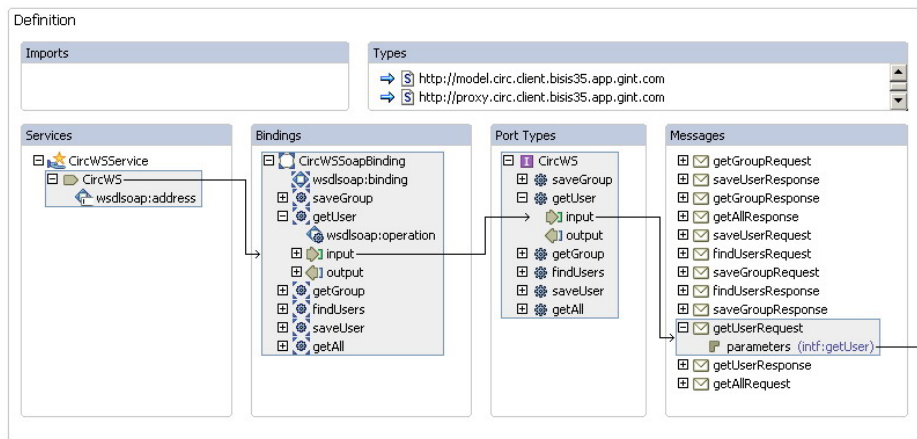
Овде је на примеру методе `getUser` илустрована употреба *Hibernate* пакета за имплементацију метода класе `ProxyLocal.java`. На сличан начин су имплементирани и остале методе ове класе.

3.2.2.3 Приступ бази података преко Интернета

Класа `ProxyNet.java` представља имплементацију интерфејса `Proxy.java` која се користи за приступ бази података у случају када се база налази на удаљеном серверу. Класа имплементира све методе интерфејса које су дефинисане и описане у одељку 2.2.5 Имплементација комуникацију са базом података остварује преко серверског дела апликације који се налази на истом или неком другом удаљеном серверу. Серверски део апликације је имплементиран као *web* сервис да би се комуникација између сервера и клијента одвијала преко *HTTP* протокола.

На слици 3.1 је дат графички приказ *WSDL* документа *web* сервиса *CircWS*. Приказане су операције које *web* сервис нуди. Те операције су еквивалентне методама које поседује интерфејс `Proxy.java`, а имплементира класа `ProxyNet.java`. Методе класе

ProxyNet.java позивају одговарајуће операције *web* сервиса и прослеђују им објекте које су добиле, а даље операције *web* сервиса извршавају функционалност метода која је дефинисана интерфејсом *Proxy.java*. Овај *web* сервис није јаван и служи искључиво за употребу од стране клијентске апликације система за коришћење библиотечке грађе.



Слика 3.1 Графички приказ WSDL документа *web* сервиса *CircWS*

На листингу 3.3 је дат пример имплементације методе *getUser* класе *ProxyNet.java*. Пре имплементације саме методе је наведен део иницијализације класе у коме се креира објекат клијента *web* сервиса преко кога се приступа *web* сервису. Сама метода *getUser* позива одговарајућу операцију *web* сервиса *getUser* и прослеђује јој одговарајући параметар. На исти начин су имплементирани и остале методе класе *ProxyNet.java*.

Серверска страна апликације садржи имплементацију *web* сервиса и објектни модел базе података имплементиран већ описаним пакетом *Model*. За имплементацију операција *web* сервиса је искоришћен *Hibernate* пакет и имплементација ових операција је иста или врло слична имплементацији метода класе *ProxyLocal.java* које су описане у претходном одељку. Разлике у односу на локални интерфејс постоје највећим делом у начину на који *Hibernate* приступа бази података. У случају локалног интерфејса *Hibernate* је

подешен да бази приступа преко *JDBC* драјвера. Тај начин приступа је изабран због једноставности одржавања система јер у том случају систему на серверској страни не треба ништа друго осим базе података. У случају интерфејса за удаљени приступ већ само постојање *web* сервиса условљава постојање серверске стране апликације која се извршава на неком *web* серверу. Из тог разлога је изабрано да се за комуникацију са базом података искористе погодности које пружа *web* сервер уместо *JDBC* драјвера, па је у овом случају *Hibernate* подешен да користи *JTA* приступ бази података кроз *web* сервер уместо *JDBC* драјвера. Због транспарентности *Hibernate* интерфејса задуженог за трансакције са базом података и у случају када се користи *JDBC* и у случају *JTA* приступа измене у коду које користе тај интерфејс су незнатне. Због тога је имплементација операција *web* сервиса врло слична или иста имплементацији метода класе *ProxyLocal.java*.

```
...
private CircWSServiceLocator locator;
private CircWS service;
...
static{
    locator = new CircWSServiceLocator();
    service = locator.getCircWS();
}
...
public Users getUser(String userID){
    return service.getUser(userID);
}
```

Листинг 3.3 Имплементације методе *getUser*

3.2.3 Имплементација менаџера

Класа *UsersManager.java* која се налази унутар *Java* пакета *Manager* имплементира функционалности које управљају током података о корисницима од корисничког интерфејса до базе података и обрнуто. Ова класа својим функционалностима остварује везу између пакета *View*, *Model* и *Proxy*.

Те функционалности обухватају:

- добављање података из базе података преко класа пакета *Proxy* који се привремено смештају у објекте класа из пакета *Model*
- чување података из објектног модела у базу података преко класа пакета *Proxy*
- пребацивање података из објектног модела на кориснички интерфејс и обрнуто.

Извршавање метода ове класе се иницира из корисничког интерфејса на акције корисника. Методе које ова класа поседује су следеће:

- *loadUser(User userForm, Users userModel)* – из објектног модела *userModel* пребацује податке о кориснику на екранску форму за рад са корисницима коју представља параметар *userForm*.

- *loadGroup(Group groupForm, Groups groupModel)* – из објектног модела *groupModel* пребацује податке о колективном кориснику на екранску форму за рад са колективним корисницима коју представља параметар *groupForm*.

- *getUser(User userForm, Group groupForm, String userID)* – за задати број корисника *userID* преко класа пакета *Proxy* добавља податке о кориснику из базе података који су смештени у класама пакета *Model*, а затим позивањем једне од две претходно описане две методе (у зависности од тога да ли је корисник индивидуални или колективни) пребацује податке из објектног модела на одговарајућу екранску форму *userForm* или *groupForm*.

- *toObjectModel(User userForm, Users userModel)* – податке са екранске форме *userForm* пребацује у објектни модел *userModel*.

- *saveUser(User userForm)* – позивом претходно описане методе податке о индивидуалном кориснику са екранске форме пребацује у објектни модел а затим иницира акцију чувања података преко класа пакета *Proxy*.

- *toObjectModel(Group groupForm, Groups groupModel)* – податке са екранске форме *groupForm* пребацује у објектни модел *groupModel*.

- *saveGroup(Group groupForm)* – позивом претходно описане методе податке о колективном кориснику са екранске форме пребацује у објектни модел а затим иницира акцију чувања података преко класа пакета *Proxy*.

- *initialiseUser(User userForm)* – иницијализује екранску форму *userForm*, као и објектни модел, за упис новог корисника.
- *loadCombos(User userForm)* – учитава вредности из шифарника преко класа пакета *Proxy* у падајуће листе приликом иницијализације екранске форме *userForm*.
- *addSigning(Signing sig)* – у објектни модел корисника додаје нови објекат типа *Signing* приликом продужења чланарине.
- *removeSigning(Signing sig)* – брише објекат *sig* из објектног модела корисника.
- *addLending(Lending lend)* – дадаје ново задужење у објектни модел корисника.
- *getMembership(MmbrTypes mt, UserCategs uc)* – за задату категорију и врсту чланства преко класа пакета *Proxy* проналази висину чланарине коју корисник треба да плати.
- *getUserId(String location)* – враћа следећи слободан број корисника за задату локацију и позива се приком аутоматске доделе броја корисника.
- *existsUser(String userID)* – преко класа пакета *Proxy* проверава да ли постоји корисник са задатим бројем *userID*.
- *getChargedUser(String ctlgno)* – преко класа пакета *Proxy* враћа корисника код кога је задужена публикација са задатим инвентарним бројем *ctlgno*.
- *dischargeUser(String ctlgno)* – преко класа пакета *Proxy* раздужује корисника код кога је задужена публикација са задатим инвентарним бројем *ctlgno*.

3.3 Имплементација рада са библиотечким записима

Део апликације који је задужен за рад са библиотечким записима је имплементиран на основу спецификације пакета *RecordUtils* (одељак 2.2.6) и класе *RecordsManager* (одељак 2.2.4). Имплементацију спецификације пакета *RecordUtils* представљају *Java* пакет *RecordUtils* са својим класама, а имплементацију класе

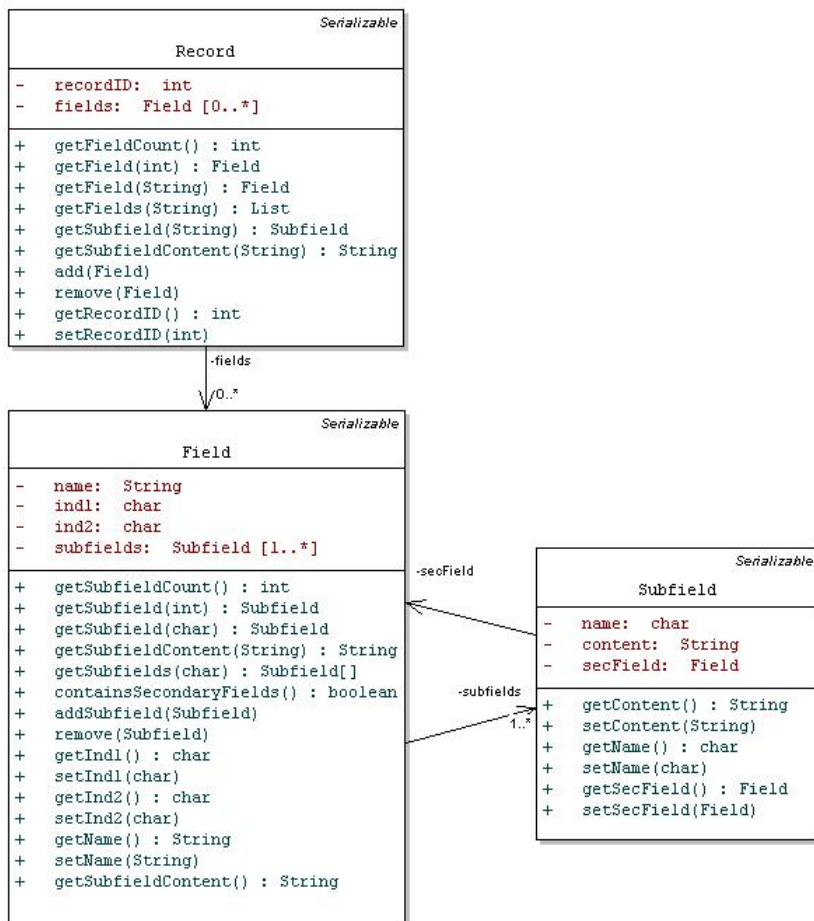
RecordsManager Java класа *RecordsManager.java* која се налази унутар Java пакета *Manager*.

3.3.1 Имплементација интерфејса библиотечког записа

Систем за коришћење библиотечке грађе садржаје из библиотечких записа добија преко имплементације класа пакета *RecordUtils* чија је спецификација описана у одељку 2.2.6. Тај пакет садржи један интерфејс и две класе који су приказани на слици 2.8. Класе и интерфејс приказани на дијаграму имају своју имплементацију у облику Java класа и интерфејса и то су: интерфејс *RecrodBean.java*, класа *RecordBeanUnimarc.java* и класа *RecordBeanMarc21.java*. Ове класе и интерфејс имплементирају све методе описане у одељку 2.2.6. Класа *RecordBeanUnimarc.java* је имплементација интерфејса *RecordBean.java* која се користи у случају када се библиотечки записи налазе у UNIMARC формату, а класа *RecordBeanMarc21* у случају када се записи налазе у формату MARC21.

У оквиру система БИСИС библиотечки записи се налазе у YUMARC формату. Због тога је направљена имплементација интерфејса *RecordBeanYumarc.java* која се користи за рад са записима у YUMARC формату какав је случај у БИСИС систему. И ова класа имплементира све методе интерфејса *RecrodBean.java*. Конструктору класе се прослеђује објектни модел записа система БИСИС. Методе класе из тог објектног модела записа узимају потребне податке да би извршиле своју функционалност.

Објектни модел записа система БИСИС је имплементиран пакетом *Record* који се налази у оквиру БИСИС система. На слици 3.2 је приказан дијаграм класа пакета *Record*. Основна класа пакета која представља библиотечки запис је класа *Record*. Садржај поља и потпоља библиотечког записа су смештени и објектима класа *Field* и *SubField*. Детаљан опис овог пакета и класа се налази у мографији [Dimic07b].

Слика 3.2 Дијаграм класа пакета *Record*

На листингу 3.4 је дат пример имплементације методе *getPublishYear* класе *RecordBeanYumarc.java* као илустрација начина имплементирања метода ове класе. Пре саме имплементације методе је приказан конструктор класе коме се прослеђује објектни модел записа *Record*. Метода *getPublishYear* по спецификацији из одељка 2.2.6 враћа годину издавања публикације. Метода је имплементирана тако да из записа враћа вредност потпоља *100c* у које је по *YUMARC* стандарду формата записа смешта вредност за годину издавања публикације. Остале методе ове класе на сличан начин враћају одговарајуће вредности из библиотечког записа. У

табели 3.1 је дат преглед свих метода класе и одговарајућих поља и потпоља записа из којих се узимају вредности.

```

...
private Record rec;
...
public RecordBeanUnimarc(Record rec){
    this.rec = rec;
}
...
public String getPublishYear(){
    return rec.getSubfieldContent("100c");
}

```

Листинг 3.4 Имплементација методе *getPublishYear*

На сличан начин су имплементиране и класе *RecordBeanUnimarc* и *RecordBeanMarc21* у складу са форматом записа. Оваквим начином имплементације систем за коришћење библиотечке грађе постаје независан од MARC формата у коме се налазе библиотечки записи јер је интерфејс преко кога се приступа садржајима из записа увек исти.

3.3.2 Имплементација менаџера записа

Класа *RecordsManager.java* имплементира функционалности везане за комуникацију између система за коришћење библиотечке грађе и библиотечких записа. У оквиру БИСИС система записима се приступа преко текст сервера имплементираног унутар система. Класа *RecordsManager.java* је задужена за комуникацију са текст сервером.

Прва функционалност за коју је класа задужена је промена статуса појединачног примерка. Приликом задуживања примерка потребно је проверити да ли је примерак у неком од задуживих статуса да би се дозволило задуживање тог примерка. Такође при задуживању је потребно променити статус примерка у задужен, као што је потребно при раздуживању статус променити у слободан. Ова функционалност је реализована кроз следеће методе класе *RecordsManager.java*:

- *lendBook(String ctlgno)* – проверава статус примерка са инвентарним бројем *ctlgno* и ако је примерак задужив мења га у заузет.

- *returnBook(String ctlgno)* – статус примерка са инвентарним бројем *ctlgno* мења у слободан.

Табела 3.1 Методе класе *RecordBeanYumarc.java*

Метода	Опис методе	Поље/потпоље
<i>getTitle()</i>	наслов дела	200a, 200i, 540a
<i>getAuthor()</i>	аутор дела	700a, 700b, 710a
<i>getPublisher()</i>	издавач	210c
<i>getPublishYear()</i>	година издавања	100c
<i>getPubPlace()</i>	место издавања	210a
<i>getSecResponsibility()</i>	секундарна одговорност	702a, 702b, 702d
<i>getSubtitle()</i>	поднаслов дела	200e
<i>getPart()</i>	том књиге	200h, 200v
<i>getPagesNo()</i>	број страна	215a
<i>getIllustrations()</i>	број илустрација	215c
<i>getDimensions()</i>	димензије књиге	215d
<i>getAccMaterial()</i>	пропратна грађа	215e
<i>getSeries()</i>	подаци о збирци	225a, 225i
<i>getNotes()</i>	напомене	300a
<i>getContentsNote()</i>	напомена о садржају дела	327a
<i>getSubjectAnalysis()</i>	предметне одреднице	6xx
<i>getSignature()</i>	сигнатура	996d
<i>getSigFormat()</i>	формат сигнатуре	996dl
<i>getNumerusCurens()</i>	numerus curens	996dn
<i>getUDC()</i>	УДК број	675a

Друга функционалност за коју је класа задужена је претраживање и добављање библиотечких записа. Претраживање се врши креирањем упита на језику текст сервера и прослеђивањем

тих упита текст серверу. Текст сервер као поготке враћа записе у објектном моделу записа који је такође имплементиран у оквиру система БИСИС. Класа има задатак да те записе остатку система за коришћење библиотечке грађе испоручи у облику објектата класа пакета *RecordUtils* чија је имплементација описана у даљем тексту. Систем даље те објекте користи за приказ података из библиотечких записа. Методе класе *RecordsManager.java* којима се комуницира са текст сервером су следеће:

- *getRecord(String ctlgno)* – добавља запис који садржи примерак са задатим инвентарним бројем *ctlgno*.
- *getRecords(Query q)* – добавља записе који задовољавају упит *q*; упит се креира помоћном методом *makeLuceneAPIQuery()* која од одговарајућих параметара креира упит на језику текст сервера.

На листингу 3.5 је дата имплементација методе *getRecord* као илустрација коришћења текст сервера из система за коришћење библиотечке грађе. Текст серверу се приступа позивањем методе *getRecordManager()* класе *BisisApp*. Прво се текст серверу прослеђује упит методом *select2* која враћа листу бројева записа који задовољавају упит, а затим се методом *getRecord* којој се прослеђује број записа добија запис у објектном моделу. Од тог објектног модела се креира објекат класе *RecordBeanYumarc.java* преко кога остатак система за коришћење библиотечке грађе добија садржаје из записа. Упит на језику текст сервера се креира позивом методе *makeQueryTerm* помоћне класе *QueryUtils*. На сличан начин су имплементирани и остале наведене методе класе *RecordsManager.java* које користе текст сервер.

```
public RecordBean getRecord(String ctlgno){
    Record record = null;
    int[] hits =BisisApp.getRecordManager()
        .select2(QueryUtils.makeQueryTerm
            ("IN", ctlgno, "", null), null);
    if (hits!=null && hits.length != 0){
        record = BisisApp.getRecordManager()
            .getRecord(hits[0]);
    }
    return new RecordBeanYumarc(record);
}
```

Листинг 3.5 Имплементација методе *getRecord*

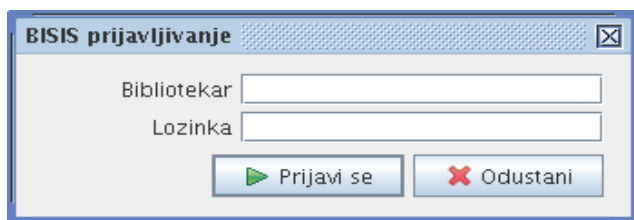
Опис коришћења апликације

У овом поглављу дат је кратак опис за коришћење и администрацију апликације од стране библиотекара. Приказане су и описане екранске форме за рад са корисницима библиотеке, за претраживање корисника, претраживање библиотечког фонда, генерисање извештаја и опомена, као и администрацију система. Описане су како изглед, тако и функционалност и међусобна повезаност екранских форми. Садржај екранских форми илустрован је са подацима из библиотеке Департмана за математику и информатику Природно – математичког факултета у Новом Саду.

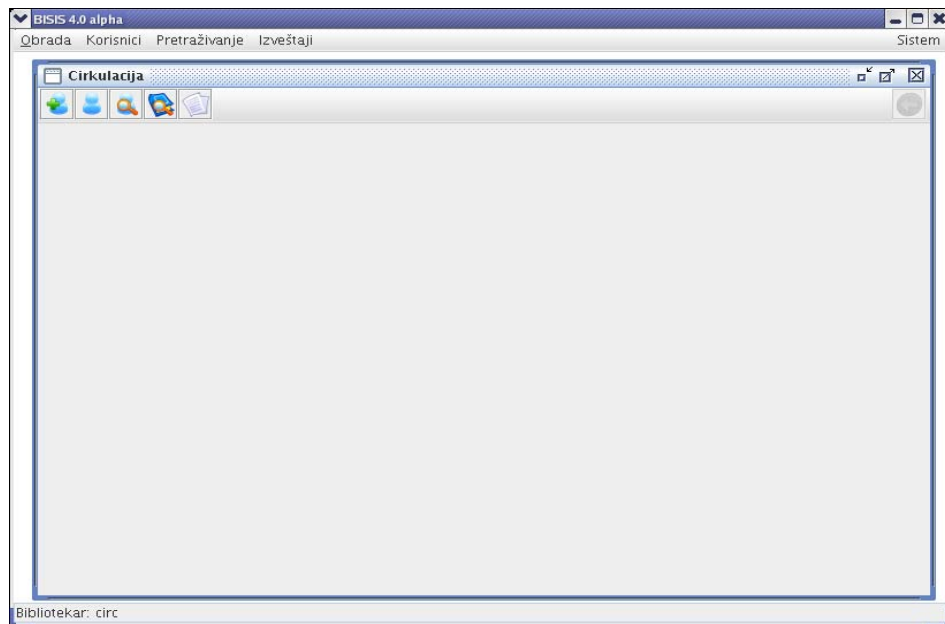
4.1 Основна екранска форма

Приликом покретања апликације прво се корисник пријављује на систем својим корисничким именом и лозинком. Зато се пре отварања основне екранске форме система појављује екранска форма за пријављивање на систем приказана на слици 4.1. На тој екранској форми корисник система уноси корисничко име и лозинку и на дугме *Prijavi se* потврђује унос. У случају успешног пријављивања отвара се основна екранска форма апликације. У супротном исписује се порука о грешци. Такође, корисник може да одустане од пријављивања на систем (дугме *Odustani*).

Основна екранска форма апликације која се отвара након успешне пријаве на систем је приказана на слици 4.2. Ова екранска форма садржи главни мени апликације и *toolbar* апликације.



Слика 4.1 Пријава корисника на систем

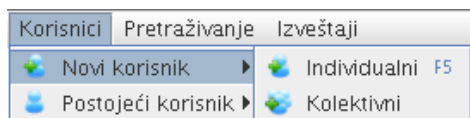


Слика 4.2 Основна екранска форма апликације

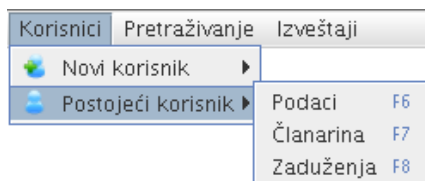
Главни мени је подељен на пет делова: *Obrada*, *Korisnici*, *Pretraživanje*, *Izveštaji* и *Sistem*. Садржај менија зависи од привилегија које има корисник који се пријавио на систем. Тако за кориснике који имају привилегије за циркулацију у менијима нису доступне ставке везане за обраду и администрацију. Слично је са корисницима који имају привилегије за обраду библиографске грађе или администрацију. Овде ће бити описане само акције везане за циркулацију библиотечке грађе. Те акције се налазе у менијима *Korisnici*, *Pretraživanje*, *Izveštaji* и *Sistem*. Мени *Korisnici* се састоји од два подменија: *Novi korisnik* и *Postojeći korisnik*. Изглед ових подменија је приказан на сликама 4.3а и 4.3б. У подменију *Novi*

korisnik се налазе акције везане за упис новог корисника. Подмени *Novi korisnik* има две акције *Individualni* и *Kolektivni*. Акција *Individualni* иницира функцију уписа новог индивидуалног корисника, а акција *Kolektivni* новог колективног корисника. Подмени *Postojeći korisnik* садржи акције везане за рад са постојећим корисником у систему. То су: *Podaci*, *Članarina* и *Zaduženja*. Преко тих акција се иницирају функције промене података корисника, продуживања чланарине и задуживања и раздуживања корисника. У менију *Pretraživanje* (слика 4.4) се налазе две акције: *Korisnici* и *Publikacije*. Акција *Korisnici* иницира функцију претраживања корисника, а акција *Publikacije* претраживање публикација. У менију *Izveštaji* се налазе акције којима се покреће генерисање извештаја који постоје у систему, а у менију *Sistem* акције везане за администрацију система и ажурирање шифарника. Све наведене функције су даље описане у тексту.

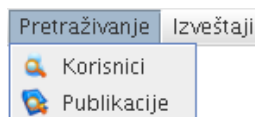
За чешће коришћене акције постоје и пречице преко функцијских тастера на тастатури. Ако је за неку акцију дефинисана пречица преко функцијског тастера, у менију поред имена акције стоји и ознака тог функцијског тастера. Те пречице омогућавају иницирање одговарајуће акције без уласка у меније, што убрзава и олакшава рад корисницима система.



4.3a



4.3b

Слика 4.3 Садржај менија *Korisnici*Слика 4.4 Садржај менија *Pretraživanje*

На *toolbar*-у апликације који је видљив на слици 4.2 налазе се пречице до најчешће коришћених акција из менија. Свака акција је представљена одговарајућом сликом као и у менију, а када се показивач миша доведе изнад слике отвара се и мали текст бокс у коме стоји опис акције. На *toolbar*-у постоји пет пречица. То су:

-  - иницира акцију уписа новог индивидуалног корисника;
-  - иницира акцију рада са подацима постојећег корисника;
-  - иницира акцију претраживања корисника;
-  - иницира акцију претраживања публикација;
-  - иницира акцију генерисања извештаја.

4.2 Подаци индивидуалних корисника

Постоје две акције које се односе на податке индивидуалних корисника и то су:

- упис новог индивидуалног корисника,
- ажурирање података постојећег корисника.

Избором неке од ове две акције отварају се екранске форме са подацима индивидуалног корисника. Те екранске форме су исте и за новог и за постојећег корисника, тако да се од тог момента рад са подацима новог и постојећег корисника не разликују са становишта корисника система. Разлика постоји једино у томе да ли ће се на крају унети подаци сачувати као подаци новог корисника или ће тим подацима да се ажурирају подаци већ постојећег корисника.

4.2.1 Упис новог корисника

Акција уписа новог корисника може да се иницира на три начина: избором опције уписа новог индивидуалног корисника из менија *Korisnici*, пречицом преко функцијског тастера *F5* на тастатури или преко *toolbar*-а избором пречице са сликом .

Иницирањем акције уписа новог индивидуалног корисника отвара се екранска форма са подацима индивидуалног корисника. Екранска форма (слике 4.6 – 4.9) садржи четири картице: *Osnovni*

podaci, *Dodatni podaci*, *Članarina* и *Zaduženja*. Свака од ове четири картице је приказана и описана у даљем тексту. Приликом уписа новог корисника екранска форма се отвара са празним пољима за унос. Поља која су означена са ознаком * су обавезна за унос, а остала су опциона. На дну екранске форме се налазе два дугмета: *Sačuvaj* и *Odustani*. На дугме *Sačuvaj* се иницира акција чувања података корисника, а на дугме *Odustani* се прекида унос и затвара се екранска форма са подацима индивидуалног корисника. Приликом акције чувања података врши се валидација података. Проверава се да ли су попуњена сва обавезна поља и да ли су попуњена поља у одговарајућем формату. У случају да је све исправно приказује се порука о успешно извршеној акцији чувања података, а у супротном приказује се порука са информацијом која поља нису прошла валидацију и зашто.

4.2.2 Ажурирање података постојећег корисника

Као и код уписа новог корисника акција ажурирања података постојећег корисника може да се иницира на три начина: избором одговарајуће опције из менија *Korisnici*, преко функцијских тастера и преко пречице на *toolbar*-у. Иницирањем акције на било који од три наведена начина отвара се форма са подацима индивидуалног корисника, као и у случају уписа новог корисника.

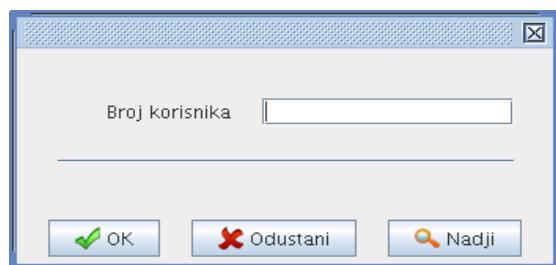
У случају иницирања акције кроз мени *Korisnici* постоје три опције: *Podaci*, *Članarina* и *Zaduženja*. У случају избора опције *Podaci* на форми која се отвара је видљива картица *Osnovni podaci*. У случају избора опције *Članarina* видљива је картица *Članarina*, а у случају избора опције *Zaduženja* картица *Zaduženja*. Независно од начина на који је форма отворена, тј. која картица је видљива, кориснику система су увек доступне и све остале картице. Тако, на пример, и у случају да је форма отворена опцијом *Podaci* и видљива је картица *Osnovni podaci*, без затварања екранске форме може се избрати и картица *Zaduženja*.

За сваку од три опције из менија постоји и пречица преко функцијских тастера на тастатури. Тастер *F6* је пречица за опцију *Podaci*, тастер *F7* је пречица за опцију *Članarina*, а тастер *F8* за опцију *Zaduženja*. Који тастер је дефинисан за коју опцију је видљиво и у

менију. Поред имена сваке опције стоји ознака тастера који је пречица за ту опцију.

На *toolbar*-у такође постоји пречица за иницирање акције ажурирања података постојећег корисника. То је пречица са сликом . Ова пречица отвара екранску форму за рад са корисницима на којој је видљива картица *Osnovni podaci*.

Након иницирања акције ажурирања података корисника на један од три описана начина, а пре отварања екранских форми за рад са подацима, отвара се екранска форма за унос броја корисника приказана на слици 4.5. На тој форми се уноси број корисника чији се подаци желе ажурирати. Након уноса броја у текстуално поље и потврде тог уноса на дугме *OK*, у случају да унесени број припада индивидуалном кориснику, отвара се екранска форма за рад са подацима индивидуалног корисника. У случају да број припада колективном кориснику, отвара се екранска форма за колективног корисника која ће бити описана у даљем тексту. Поред дугмета *OK* на екранској форми се још налазе и дугме *Odustani* којим се одустаје од акције и затвара форма и дугме *Nadji* које води на екранску форму за претраживање корисника, која ће такође бити описана у даљем тексту. Ову акцију корисник система користи у случају када не зна број корисника чије податке жели ажурирати.



Слика 4.5 Унос броја корисника

Након отварања екранске форме за рад са подацима индивидуалног корисника на којој су приказани подаци траженог корисника врши се ажурирање података. Као што је већ објашњено у случају уписа новог корисника, поља која су означена са ознаком * су обавезна, а остала су опциона. Акција чувања података се

иницира на дугме *Sačuvaj*, а на дугме *Odustani* се прекида ажурирање и затвара се екранска форма за рад са подацима индивидуалног корисника. Приликом акције чувања података се на исти начин врши валидација података као што је већ описано у случају уписа новог корисника.

The screenshot shows a software window titled 'BISIS 4.0 alpha' with a menu bar containing 'Obrada', 'Korisnici', 'Pretraživanje', 'Izveštaji', and 'Sistem'. The main area is titled 'Cirkulacija' and has four tabs: 'Osnovni podaci' (selected), 'Dodatni podaci', 'Članarina', and 'Zaduženja'. The 'Osnovni podaci' tab contains two columns of input fields. The left column includes: 'Ime*' (Daniijela), 'Prezime*' (Tešendić), 'Ime roditelja*' (Drago), 'JMBG*' (3004979128009), 'Dokument' (Licna karta), 'Broj dokumenta*' (844/97), 'Mesto izdavanja*' (Modriča), and 'Zemlja izdavanja*'. The right column includes: 'Adresa*' (Svetosavska 26), 'Broj pošte*' (74480), 'Grad*' (Modriča), 'Telefon' (053/811-259), 'E-mail', 'Pol' (radio buttons for Muško and Žensko, with Žensko selected), 'Uzrast' (radio buttons for Odrasli and Dete, with Odrasli selected), and 'Indikator opomena' (checkbox checked). At the bottom right is a 'Štampaj' button. At the bottom of the window are two large buttons: 'Sačuvaj' (with a green checkmark) and 'Odustani' (with a red X). The status bar at the very bottom reads 'Bibliotekar: circ'.

Слика 4.6 Картица *Osnovni podaci*

4.2.3 Картица *Osnovni podaci*

На слици 4.6 је приказана екранска форма за рад са подацима индивидуалног корисника на којој је видљива картица *Osnovni podaci*. На овој картици се уносе наведени подаци о кориснику приказани на слици 4.6. Поред овога могуће је корисника индикатором опомена укључити или искључити из система за штампање опомена. Дугме *Štampaj* служи за штампање чланске карте корисника. На постојећој екранској форми ово дугме иницира отварање пете картице на којој се приказује чланска карта и одатле може да се шаље на штампу. У пољима *Broj pošte* и *Grad* функцијски тастер *F12* иницира отварање екранске форме са листом градова и њихових бројева. Листа може да се сортира по

називу града или броју. Избором неког града из листе подаци се преносе у поља *Broj pošte* и *Grad* на картици.

The screenshot shows the 'BISIS 4.0 alpha' window with the 'Dodatni podaci' tab selected. The form is divided into several sections:

- Top Bar:** 'BISIS 4.0 alpha' title bar, menu bar (Obrada, Korisnici, Pretraživanje, Izveštaji), and 'Sistem' button.
- Navigation:** 'Cirkulacija' tab, sub-tabs: 'Osnovni podaci', 'Dodatni podaci' (active), 'Članarina', 'Zaduženja'.
- Form Fields:**
 - Dodatna adresa:** Adresa (Gogoljeva 7), Broj pošte (21000), Grad (Novi Sad), Telefon (Novi Sad).
 - Maternji jezik:** Dropdown menu.
 - Napomena:** Text area.
 - Interesovanja:** Text area.
 - Organizacija:** Dropdown menu.
 - Obrazovanje:** Dropdown menu.
 - Zanimanje:** Text area.
 - Zvanje:** Text area.
 - Broj indeksa:** 454/98.
 - Godina:** Dropdown menu.
- Duplikati Table:**

Datum	Broj duplikata
- Buttons:** 'Sačuvaj' (green checkmark), 'Odustani' (red X), 'Blokiraj' (orange X).
- Footer:** 'Bibliotekar: circ'.

Слика 4.7 Картица *Dodatni podaci*

4.2.4 Картица *Dodatni podaci*

На слици 4.7 је приказана екранска форма за рад са подацима индивидуалног корисника на којој је видљива картица *Dodatni podaci*. На овој картици се уносе наведени подаци о кориснику приказани на слици 4.7. Подаци о организацији, степену образовања, години студија и матерњем језику су шифрирани па се они уносе избором једне од опција из падајуће листе. Ако жељена опција није у листи потребно ју је додати у шифарник што се ради у делу за администрацију система. На овој картици се води и евиденција о издавању дупликата чланских карата кориснику. У табели *Duplikati* се приказује редни број издатог дупликата и датум издавања дупликата. На дугме са сликом  се додаје нови ред у табелу у случају издавања новог дупликата, а на дугме са сликом  се брише ред из тебеле у случају поништавања дупликата. Дугме *Blokiraj* служи за блокирање налога корисника. Иницирањем акције се

отвара текст бокс у који се уноси разлог блокирања. Након уноса текста тастером *Enter* на тастатури се потврђује унос. Текст на дугмету *Blokiraj* се мења у *Aktiviraj* и даље то дугме служи за поновно активирање налога корисника. Индикација да је картица блокирана и разлог блокирања се приказују на картици *Zaduženja* у кратком инфо-у о кориснику. Слично као на картици *Osnovni podaci* у пољима *Broj pošte* и *Grad* функцијски тастер *F12* иницира отварање екранске форме са листом градова и њихових бројева.

4.2.5 Картица *Članarina*

На слици 4.8 је приказана екранска форма за рад са подацима индивидуалног корисника на којој је видљива картица *Članarina*. На овој картици се уносе број корисника и остали подаци о чланству у библиотеку.

У делу за администрацију система дефинише се колико број корисника има цифара и да ли број има или нема префикс шифре одељења коме корисник припада. На основу тих подешавања се формира број корисника. У случају да је у систему дефинисано да број корисника има префикс одељења, у делу за унос броја корисника на овој картици активна је падајућа листа. Из те листе се бира одељење коме корисник припада. На основу избора, у првом текстуалном пољу за број корисника се појављује шифра изабраног одељења. У друго текстуално поље за број корисника се уноси број корисника (за изабрано одељење). Број се може унети ручно са тастатуре или се може аутоматски генерисати. Аутоматско генерисање броја се иницира на дугме *Auto*. Број се додељује на основу дефинисаног бројача бројева корисника у систему. Додељује се први следећи слободан број у односу на вредност бројача. Свако одељење има одвојено дефинисан бројач.

Поред броја корисника на овој картици се уноси информација о томе да ли је индивидуални корисник учлањен преко неког колективног корисника. У том случају у делу картице за колективног корисника се бира опција *Da* и из падајуће листе, која тад постаје активна, се бира колективни корисник преко кога је индивидуални корисник учлањен. Поред овога постоје још и падајуће листе за избор врсте учлањења и категорије корисника.

Одељења, колективни корисници, врсте учлањења и категорије корисника су шифрирани подаци па се они уносе избором једне од опција из падајуће листе. Ако жељена опција није у листи потребно ју је додати у шифарник што се ради у делу за администрацију система.

BISIS 4.0 alpha

Qbrada Korisnici Pretraživanje Izveštaji Sistem

Cirkulacija

Osnovni podaci Dodatni podaci **Članarina** Zaduženja

Broj korisnika _____ Kolektivni ☐ Da ☒ Ne

Odeljenje _____

Broj korisnika* 000001369 Auto

Kategorija* Student Vrsta učlanjenja* Redovan

Članarina

Odeljenje*	Datum*	Važi do*	Cena	Priznanica
Biblioteka	Oct 6, 1998	Oct 6, 2003	0	

Sačuvaj Odustani

Bibliotekar: circ

Слика 4.8 Картица *Članarina*

У табели на доњем делу картице се води евиденција о чланству корисника. Један ред табеле носи информацију о периоду чланства, одељењу на које је корисник учлањен у том периоду, висини надокнаде коју је платио и броју признанице која му је издата. У случају продуживања чланства на дугме са сликом  се додаје нови ред у табелу. При томе се аутоматски попуњавају датум продужења са тренутним датумом, датум до кад важи чланство и цена, а тај датум и цена се аутоматски рачунају на основу претходно изабране врсте учлањења и категорије корисника. Вредност за одељење се уноси кроз падајућу листу као и у случају броја корисника. Аутоматски додељени датуми и цена се могу мењати. Ради лакшег уноса података приликом измене датума се отвара графички

приказ календара из кога се бира одговарајући датум. На дугме са сликом  се брише ред из табеле у случају поништавања чланста, а на дугме са сликом  се иницира акција штампања признанице кориснику за продужено чланство.

4.2.6 Картица *Zaduženja*

На слици 4.9 је приказана екранска форма за рад са подацима индивидуалног корисника на којој је видљива картица *Zaduženja*. На овој картици се врши задуживање и раздуживање корисника.

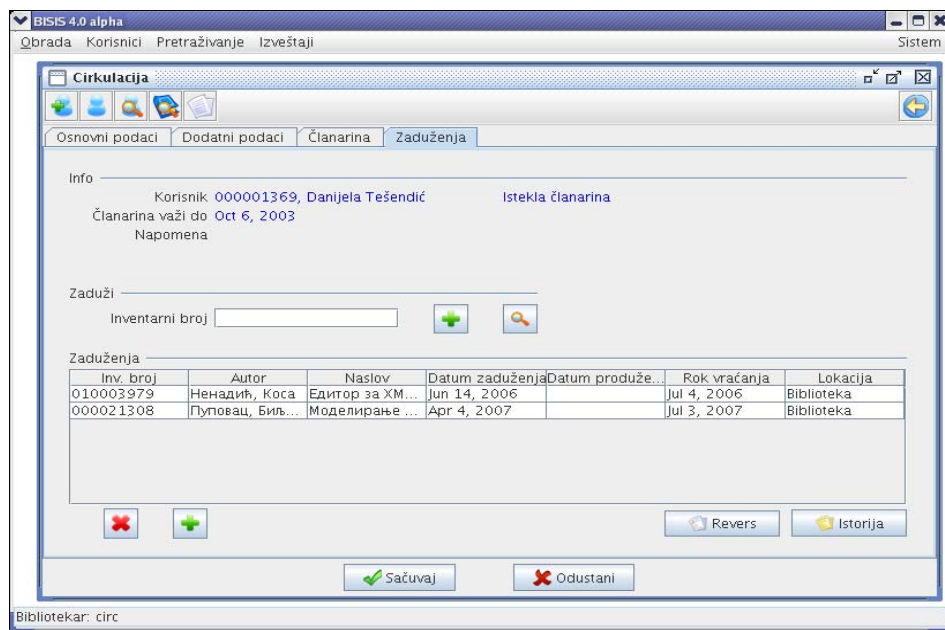
На горњем делу картице се налази кратак инфо о кориснику. Ту се приказују *Korisnik* (број корисника, име и презиме), *Članarina važi do* (датум до кад важи чланство) и *Napomena*. Поред тога, са десне стране стоје упозорења о истицању чланарине ако је истекла, посланим опоменама кориснику и блокираном налогу корисника. Такође стоји и индикатор о издатом дупликату чланске карте кориснику.

На доњем делу картице се налази табела са задуженим публикацијама. За сваку публикацију је видљив инвентарни број, аутор дела, наслов дела, датум задуживања, датум продуживања и рок враћања публикације, као и одељење на ком је извршено задуживање. Над сваком публикацијом у табели је могуће урадити две акције: раздуживање и продуживање задужења. Те акције се иницирају са два дугмета испод табеле. Дугме са сликом  служи за раздуживање публикација, а дугме са сликом  за продуживање задужења.

Публикације је могуће задужити на два начина: уношењем инвентарног броја у текстуално поље изнад табеле и потврдом уноса на дугме поред поља са сликом  ; претраживањем библиотечког фонда које се иницира дугметом са сликом  које води на екранску форму за претраживање библиотечког фонда која ће бити описана у даљем тексту.

Дугме *Revers* иницира акцију штампања реверса кориснику. На реверс се штампа садржај табеле са задужењима. Дугме *Istorija* иницира штампање историје задуживања корисника. То је један од извештаја који постоје у систему и ово дугме води на екранску

форму за генерисање извештаја која ће бити описана у даљем тексту.



Слика 4.9 Картица *Zaduženja*

4.3 Подаци колективних корисника

Као и код индивидуалних корисника постоје две акције које се односе на податке колективних корисника и то су:

- упис новог колективног корисника,
- ажурирање података постојећег корисника.

Избором неке од ове две акције отвара се екранска форма за рад са подацима колективног корисника приказана на слици 4.10. Та екранска форма је иста и за новог и за постојећег корисника. Разлика постоји једино у томе да ли ће се на крају унети подаци сачувати као подаци новог корисника или ће тим подацима да се ажурирају подаци већ постојећег корисника.

4.3.1 Упис новог корисника

Акција уписа новог колективног корисника се иницира избором опције уписа новог колективног корисника из менија *Korisnici*.

Иницирањем акције уписа новог колективног корисника отвара се екранска форма са подацима колективног корисника приказана на слици 4.10. На екранској форми се уноси скуп података за колективног корисника. Приликом уписа новог корисника екранска форма се отвара са празним пољима за унос. Поља која су означена са ознаком * су обавезна за унос, а остала су опциона. На дну екранске форме се налазе два дугмета: *Sačuvaj* и *Odustani*. На дугме *Sačuvaj* се иницира акција чувања података корисника, а на дугме *Odustani* се прекида унос и затвара се екранска форма. Приликом акције чувања података врши се валидација података. Проверава се да ли су попуњена сва обавезна поља и да ли су попуњена поља у одговарајућем формату. У случају да је све исправно приказује се порука о успешно извршеној акцији чувања података, а у супротном приказује се порука са информацијом која поља нису прошла валидацију и зашто.

The screenshot shows a web application window titled 'BISIS 4.0 alpha' with a menu bar containing 'Obrada', 'Korisnici', 'Pretraživanje', 'Izveštaji', and 'Sistem'. The main content area is titled 'Cirkulacija' and contains the following fields and controls:

- Naziv organizacije***: Text input field containing 'FTN'.
- Adresa***: Text input field containing 'Trg Dositeja Obradovića 6'.
- Broj pošte***: Text input field containing '21000'.
- Grad***: Text input field containing 'Novi Sad'.
- Telefon**: Empty text input field.
- Telefax**: Empty text input field.
- E-mail**: Empty text input field.
- Broj korisnika**: Text input field containing '498'.
- Odeljenje**: Dropdown menu.
- Datum upisa***: Text input field containing 'Oct 16, 2003'.
- Kontakt**: Section containing:
 - Ime***: Text input field containing 'Dragan'.
 - Prezime***: Text input field containing 'Simić'.
 - E-mail***: Text input field containing 'simic@uns.ns.ac.yu'.
- Dodatna adresa**: Section containing:
 - Adresa**: Empty text input field.
 - Broj pošte**: Empty text input field.
 - Grad**: Empty text input field.
 - Telefon**: Empty text input field.

At the bottom of the form are two buttons: a green 'Sačuvaj' button and a red 'Odustani' button. The status bar at the bottom left reads 'Bibliotekar: circ'.

Слика 4.10 Подаци колективног корисника

4.3.2 Ажурирање података постојећег корисника

Акција ажурирања података постојећег колективног корисника се иницира избором опције *Podaci* из менија *Korisnici*. Иницирањем акције отвара се претходно описана екранска форма за рад са подацима колективног корисника. Пре отварања ове екранске форме, отвара се екранска форма за унос броја корисника приказана на слици 4.5. Рад са овом екранском формом је претходно описан у одељку за ажурирање података постојећег индивидуалног корисника.

Након отварања екранске форме са подацима колективног корисника на којој су приказани подаци траженог корисника врши се ажурирање података. Рад са овом екранском формом је описан у претходном одељку.

4.4 Претраживање корисника

Акција претраживања корисника може да се иницира на два начина: избором опције претраживања корисника из менија *Pretraživanje* и преко *toolbar*-а избором пречице са сликом .

Иницирањем акције претраживања корисника отвара се екранска форма приказана на слици 4.11. На екранској форми се задају критеријуми претраживања. Претраживање се може вршити по свим подацима о кориснику који су дефинисани у систему.

Може да се зада до пет критеријума типа “назив податка = вредност податка”. Један критеријум претраживања представља један ред екранске форме дефинисан називом податка по ком се претражује и текстуалним пољем у које се уноси тражена вредност. Дугме са знаком “...” иницира акцију избора податка по ком се претражује. Назив изабраног податка је видљив са леве стране дугмета. Са десне стране дугмета се налази текстуално поље у које се задаје тражена вредност изабраног податка. Уколико тражени податак узима вредности из неког од шифарника уместо текстуалног поља за унос вредности се појављује падајућа листа из које се бира једна од постојећих вредности тог шифарника. Критеријуми се могу укрштати између себе са операторима *and*, *or* и *not*. Избор оператора се врши из падајуће листе поред

текстуалног поља за унос вредности. Тај оператор се односи на критеријум дефинисан тим редом и први следећи задати критеријум у редовима испод.

Поред тога, на ове критеријуме је могуће додати и временски период за датуме: датум задуживања, датум раздуживања, рок враћања, датум продуживања чланства и датум до кад важи чланство, као и локацију за сваку од тих акција извршену у том временском периоду. Слично као у случају података о кориснику, датум по ком се претражује се бира на дугме са знаком "...". Приликом задавања вредности датума ради лакшег уноса података могуће је отворити графички приказ календара из кога се бира одговарајући датум. Локација се бира из падајуће листе поред текстуалних поља за датуме.

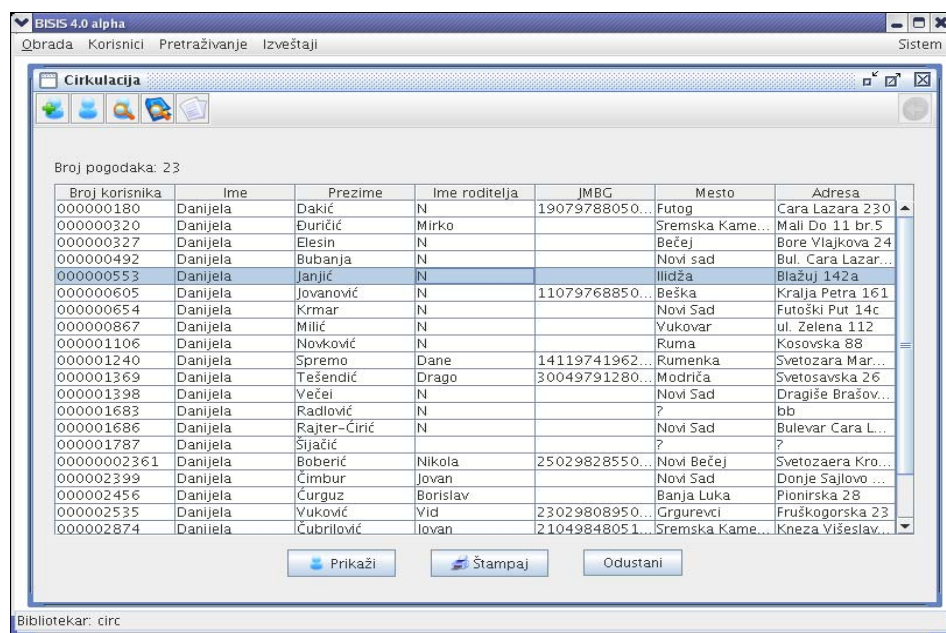
Укрштањем података о кориснику и одговарајућих датума могу се добити разни статистички подаци. На пример, укрштајући категорију корисника и период задуживања на одређеној локацији добиће се корисници који припадају задатој категорији, а задуживали су публикације у задатом периоду на задатој локацији.

The screenshot displays the 'Cirkulacija' window within the BISIS 4.0 alpha application. The window title bar includes 'BISIS 4.0 alpha' and standard OS controls. The menu bar shows 'Obrada', 'Korisnici', 'Pretraživanje', and 'Izveštaji'. The 'Cirkulacija' window has a toolbar with icons for home, search, and document management. The main area contains search criteria for circulation, organized into two sections. The first section for user identification includes fields for 'Broj korisnika', 'Ime' (containing 'daniijela'), 'Prezime', 'JMBG', and 'Adresa', each followed by a dropdown menu with an 'and' operator. The second section for loan dates includes 'Datum zaduženja' and 'Datum razduženja', each with a date picker icon and a dropdown menu. At the bottom of the form are 'Pretraži' and 'Odustani' buttons. The status bar at the bottom left indicates 'Bibliotekar: circ'.

Слика 4.11 Претраживање корисника

При дну екранске форме се налазе два дугмета *Pretraži* и *Odustani*. Дугме *Pretraži* иницира претрагу по задатим критеријумима, а дугме *Odustani* прекида претрагу и затвара екранску форму.

На слици 4.12 је дата екранска форма на којој се приказују резултати претраживања корисника. Пронађени корисници на основу задатог критеријума су приказани у табели. За сваког корисника су приказани број корисника, име и презиме, име родитеља, ЈМБГ, место и адреса становања. Изнад табеле стоји информација о укупном броју погодака. Испод табеле се налазе три дугмета: *Prikaži*, *Štampaj* и *Odustani*. Дугме *Prikaži* отвара екранске форме за рад са подацима одабраног корисника које су већ претходно описане у одељку у ком се описује рад са подацима индивидуалних корисника. Дугме *Štampaj* штампа садржај табеле, тј. листу пронађених корисника по задатом критеријуму. У заглављу се штампа критеријум по ком се претраживало, а испод листа корисника са подацима који су приказани у табели. Дугме *Odustani* затвара екранску форму.



Слика 4.12 Приказ резултата претраживања корисника

4.5 Претраживање публикација

Акција претраживања публикација може да се иницира на два начина: избором опције претраживања публикација из менија *Pretraživanje* и преко *toolbar*-а избором пречице са сликом .

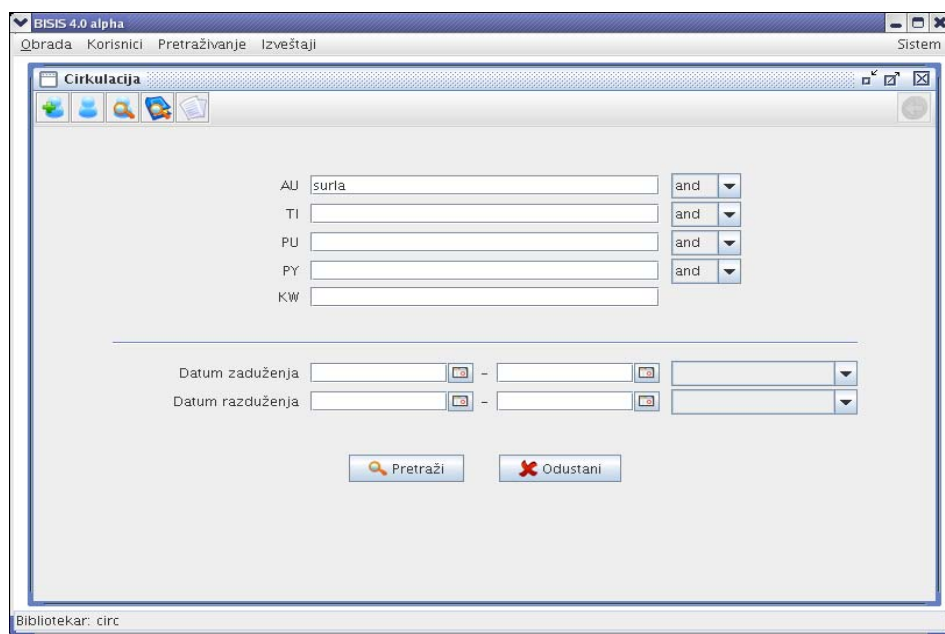
Иницирањем акције претраживања публикација отвара се екранска форма приказана на слици 4.13. На екранској форми се задају критеријуми претраживања. Претраживање се врши по понуђеним префиксима текст сервера БИСИС система. Ти префикси се односе на садржај библиотечког записа.

Може да се зада до пет критеријума типа “префикс = трежена вредност”. Један критеријум претраживања представља један ред екранске форме дефинисан називом префикса по ком се претражује и текстуалним пољем у које се уноси тражена вредност. Екранска форма за избор префикса се добија на тастер F9. Критеријуми се могу укрштати између себе са операторима *and*, *or* и *not*. Избор оператора се врши из падајуће листе поред текстуалног поља за унос вредности. Тај оператор се односи на критеријум дефинисан тим редом и први следећи задати критеријум у редовима испод.

Поред тога, на ове критеријуме је могуће додати и временски период за датум задуживања и датум раздуживања публикације, као и локацију за сваку од тих акција извршену у том временском периоду. Приликом задавања вредности датума ради лакшег уноса података могуће је отворити графички приказ календара из кога се бира одговарајући датум. Локација се бира из падајуће листе поред текстуалних поља за датуме.

Укрштањем неких префикса са периодима задуживања и раздуживања могу се добити разни статистички подаци о коришћењу библиотечког фонда. На пример, укрштајући УДК број и период задуживања на одређеној локацији добиће се примерци (број примерака) који имају задати УДК број, а задуживани су у задатом периоду на задатој локацији.

При дну екранске форме се налазе два дугмета *Pretraži* и *Odustani*. Дугме *Pretraži* иницира претрагу по задатим критеријумима, а дугме *Odustani* прекида претрагу и затвара екранску форму.

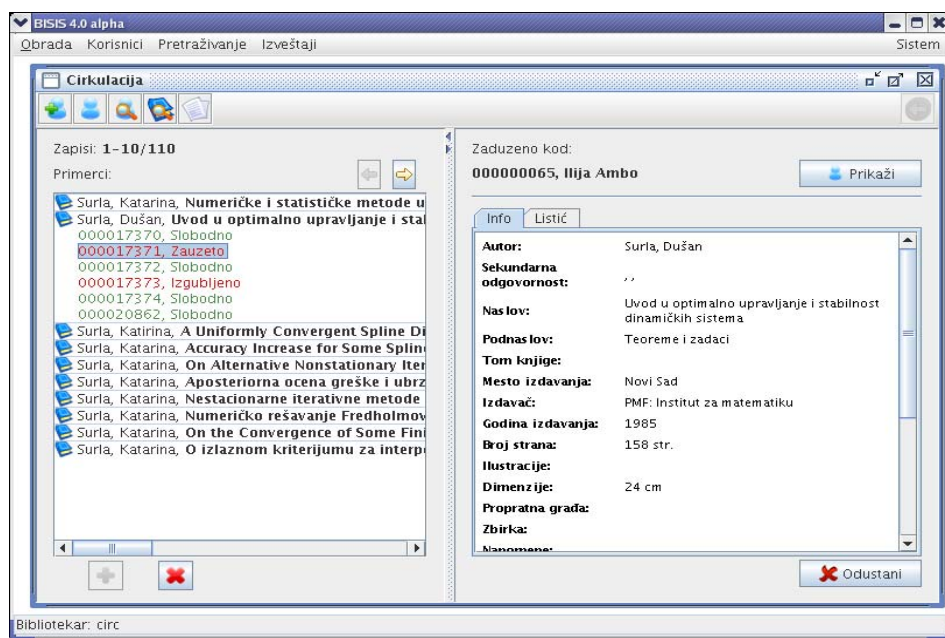


Слика 4.13 Претраживање публикација

Екранска форма за приказ резултата претраживања је приказана на слици 4.14. На левој половини екранске форме се приказује листа резултата. У резултате спадају библиотечки записи који су задовољили критеријуме претраживања. За сваки запис у листи су приказани аутор, наслов, издавач и година издања. Двокликом на сваки елемент листе се отвара подлиста за тај елемент која садржи инвентарне бројеве примерака који припадају том запису и статусе тих примерака. Зеленом бојом су приказани инвентарни бројеви који имају неки од статуса у ком је примерак задужив, а црвеном они који имају статус у ком примерак није задужив. Поновним двокликом на елемент листе затвара се подлиста инвентарних бројева.

Резултати који се приказују у листи су подељени у блокове од по двадесет записа, тј. у једном тренутку се у листи налази један блок од највише двадесет погодака. Дугме са сликом ➡ учитава следећи блок од двадесет записа у листу, а дугме са сликом ⬅ даје претходно приказани блок.

Укупан број погодатака је приказан изнад листе резултата. У делу *Zapisi* се приказује укупан број записа који су задовољили критеријуме претраживања, као и који од тих записа су тренутно приказани у листи. У делу *Primerci* се приказује број примерака који се налазе у записима. У случају да су приликом задавања критеријума претраживања задати и период задуживања или период раздуживања примерака, број који је наведен у делу *Primerci* укључује само оне примере који задовољавају тај критеријум. У случају да у критеријуме претраживања нису укључени датум задуживања или раздуживања у делу *Primerci* се приказује укупан број примерака из свих записа који су погоди.



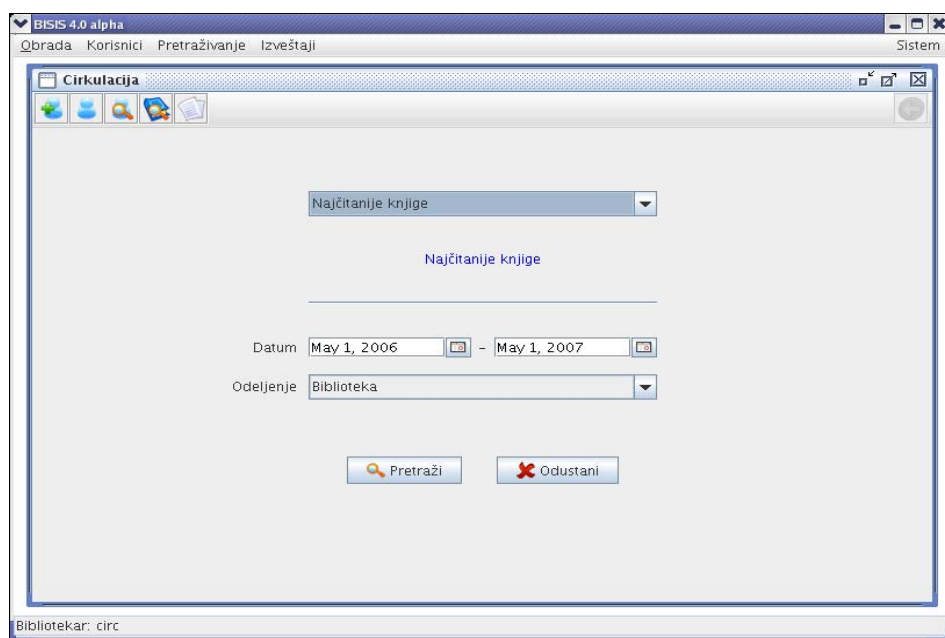
Слика 4.14 Приказ резултата претраживања публикација

Примерак у подлисти који је задужен могуће је раздужити дугметом са сликом . Такође за сваки задужени примерак се у десном горњем углу форме приказује број, име и презиме корисника код кога је задужен. Дугме *Prikaži* отвара екранске форме за рад са подацима тог корисника које су већ претходно описане у одељку у ком се описује рад са подацима индивидуалних

корисника. Примерке који су задуживи (означени зеленом бојом) је могуће задужити. Дугметом са сликом  се иницира акција задуживања корисника публикацијом. Отвара се екранска форма за унос броја корисника који се задужује, а затим се отвара екранска форма за рад са подацима индивидуалног корисника на којој је видљива картица са већ постојећим задужењима корисника којима се додаје и ово ново задужење. Екранске форме за унос броја корисника и рад са подацима индивидуалних корисника су већ претходно описане.

На десној страни екранске форме за приказ резултата претраживања се приказују детаљи библиотечког записа одабраног у листи погодака. Детаљи се приказују на два начина: у облику кратког инфо-а као што се види на слици 4.14 или у облику каталогског листића. Жељени начин приказа постаје видљив одабиром картице *Info* или *Listić*.

На дугме *Odustani* се затвара екранска форма.



Слика 4.15 Генерисање извештаја

4.6 Генерисање извештаја

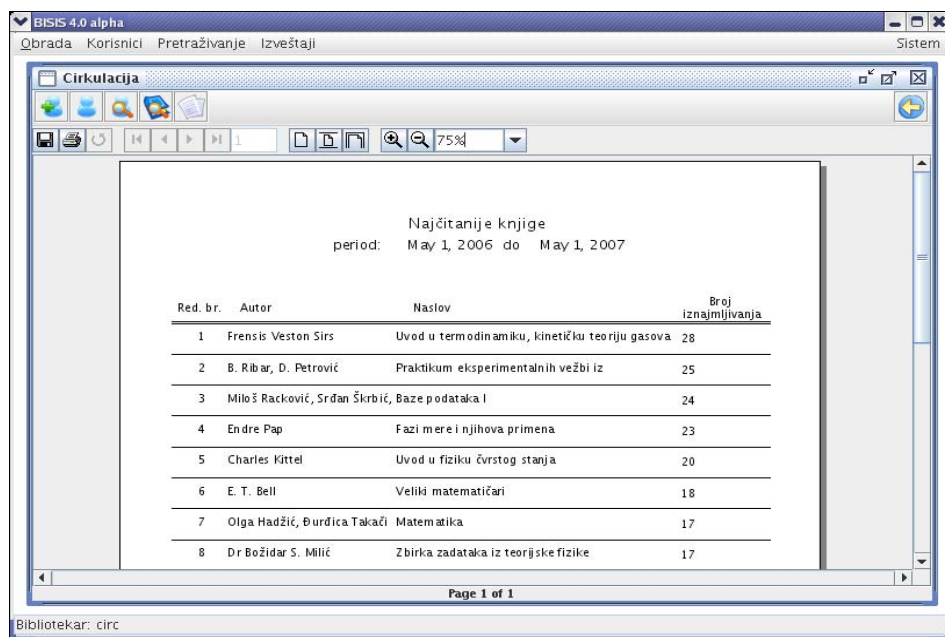
Акција генерисања извештаја може да се иницира на два начина: избором опције извештаја за циркулацију из менија *Izveštaji* и преко *toolbar*-а избором пречице са сликом .

Иницирањем акције отвара се екранска форма приказана на слици 4.15. На форми се из падајуће листе бира извештај који жели да се генерише. Избором извештаја испод падајуће листе појављује се опис извештаја. Задаје се период за који се жели генерисати извештај. Приликом задавања вредности датума ради лакшег уноса података могуће је отворити графички приказ календара из кога се бира одговарајући датум. Након задавања датума из падајуће листе се бира одељење за које се генерише извештај. На дугме *Pretraži* се иницира акција генерисања извештаја, а на дугме *Odustani* се прекида генерисање извештаја и затвара се екранска форма.

Извештаји које је могуће генерисати су:

- Структура уписаних корисника (даје дневну и периодичну статистику о структури уписаних корисника по категорији корисника, по врсти учлањења и по полу).
- Збирни извештај (даје дневну и периодичну статистику о броју задужених и раздужених публикација, броју задужених и раздужених корисника, броју активних и пасивних корисника као и броју уписаних корисника).
- Књига уписа (дневни извештај који садржи податке о уписаним корисницима библиотеке у току једног дана као и финансијски биланс наплаћене чланарине, постоје четири варијанте овог извештаја: корисници сортирани по абecedном реду, корисници груписани по библиотекару који је вршио учлањење, корисници груписани по категорији и корисници груписани по врсти учлањења).
- Структура посетилаца (даје дневну и периодичну статистику о структури корисника библиотеке по категорији корисника, по врсти учлањења и по полу).
- Посетиоци (дневни извештај који садржи податке о корисницима библиотеке који су посетили библиотеку тог дана).

- Листа колективних корисника.
- Листа индивидуалних корисника који припадају датом колективном кориснику.
- Листа корисника са блокираним налогом.
- Историји задуживања члана.
- Историји задуживања публикације – Картица књиге.
- Најчитаније књиге.
- Најчитаније књиге из одређене УДК групе.
- Највреднији читаоци.
- Издате и враћене књиге по УДК групама.
- Издате и враћене књиге по језику.
- Читаност по категорији корисника и УДК групама.



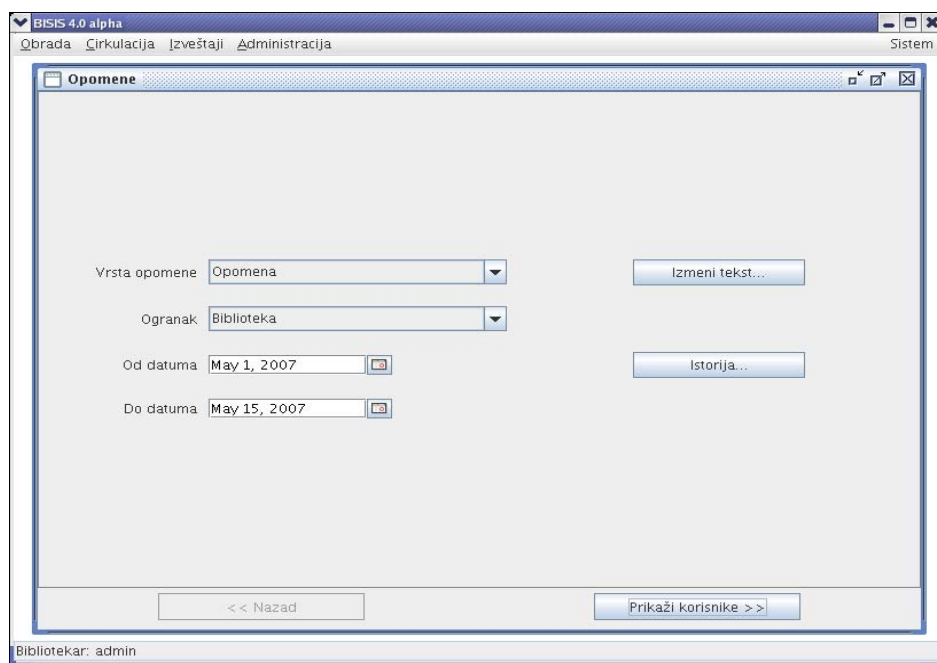
Слика 4.16 Приказ генерисаног извештаја

На слици 4.16 је приказана форма која садржи изгенерисан извештај *Најчитаније књиге*. У дну форме се види укупан број страна извештаја и број тренутно видљиве стране. У врху екрансе форме се налазе акције којима се могу листати стране извештаја, повећавати или смањивати приказ, сачувати извештај у фајл или га одштампати.

Иста екранска форма се користи и за приказ осталих извештаја које је могуће генерисати у систему.

4.7 Генерисање опомена

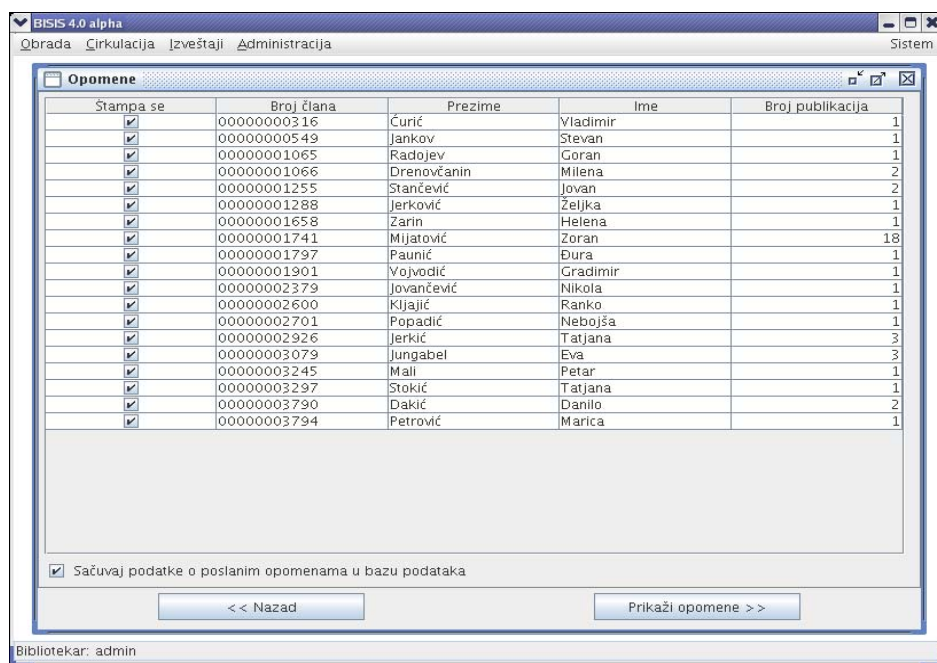
Акција генерисања опомена се иницира из менија *Administracija* који је доступан само оним библиотекарима који имају администраторске привилегије на систему. Опомена је штампани докуменат који се шаље кориснику који је прекорачио рок за враћање публикација. Постоје две врсте опомена: Опомена и Опомена пред утужење. Опомене се генеришу периодично у односу на рок враћања публикације.



Слика 4.17 Генерисање опомена

Иницирањем акције из менија се отвара основна екранска форма за генерисање опомена који је приказан на слици 4.17. Са ове екранске форме могу да се генеришу опомене, мењају текстови опомена и прегледа и штампа историја изгенерисаних опомена.

За генерисање опомена потребно је задати врсту опомене, одељење и период за које се опомене генеришу. Врста опомене и одељење се бирају из падајућих листи, а приликом задавања вредности датума ради лакшег уноса података могуће је отворити графички приказ календара из кога се бира одговарајући датум. Дугме *Prikaži korisnike* иницира акцију проналажења корисника којима треба генерисати опомену у односу на задате параметре.



Слика 4.18 Корисници којима се генеришу опомене

Листа корисника којима се генеришу опомене се приказује у табели на следећој екранској форми која је приказана на слици 4.18. У табели се за сваког корисника приказују број, име, презиме, број публикација за које је прекорачен рок враћања. Поред тога, за сваког корисника у колони *Štampa se* библиотекар може да изабере

да ли ће се за тог корисника генерисати опомена или не. Испод табеле се налази опција којом библиотекар бира да ли ће се информације о послатим опоменама сачувати или не. Дугме *Prikaži opomene* иницира акцију генерисања опомена, а дугме *Nazad* враћа на почетну екранску форму приказану на слици 4.17.

На слици 4.19 је приказана екранска форма са изгенерисаним опоменама. Са ове екранске форме опомене се могу листати, штампати или сачувати у фајл систему. Дугме *Nazad* враћа на почетну екранску форму приказану на слици 4.17, а дугме *Spisak* даје списак изгенерисаних опомена тј. корисника којима су опомене изгенерисане. Изглед екранске форме са тим списком опомена је дат на слици 4.20. Са те екранске форме са дугметом *Opomene* се враћа на екранску форму са изгенерисаним опоменама приказану на слици 4.19, а са дугметом *Nazad* се враћа на почетну екранску форму.

BIBLIOTEKA Departmana za matematiku i **OPOMENA**
 Departman za matematiku i informatiku rok vraćanja: 10.05.2007.
 Prirodno - matematički fakultet

Trg Dositeja Obradovića 3
 21000 Novi Sad

Uvidom u našu evidenciju ustanovili smo da niste u predviđenom roku vratili pozajmljene knjige-e:

R. br.	Naslov	Autor	Inv.broj	Signatura	Br dana prekoračenja
1	Zbirka zadataka iz više	Miličić, Pavle M.	00000018226	M-18226	183

Molimo vas da svoju obavezu ispunite u roku od 3 (tri) dana kako ne bi došlo do primene drugih predviđenih mera, kao i da prilikom vraćanja knjiga u Biblioteku Departmana za matematiku i informatiku, Trg Dositeja Obradovića 3, izvršite obavezu nadoknade troškova u iznosu:

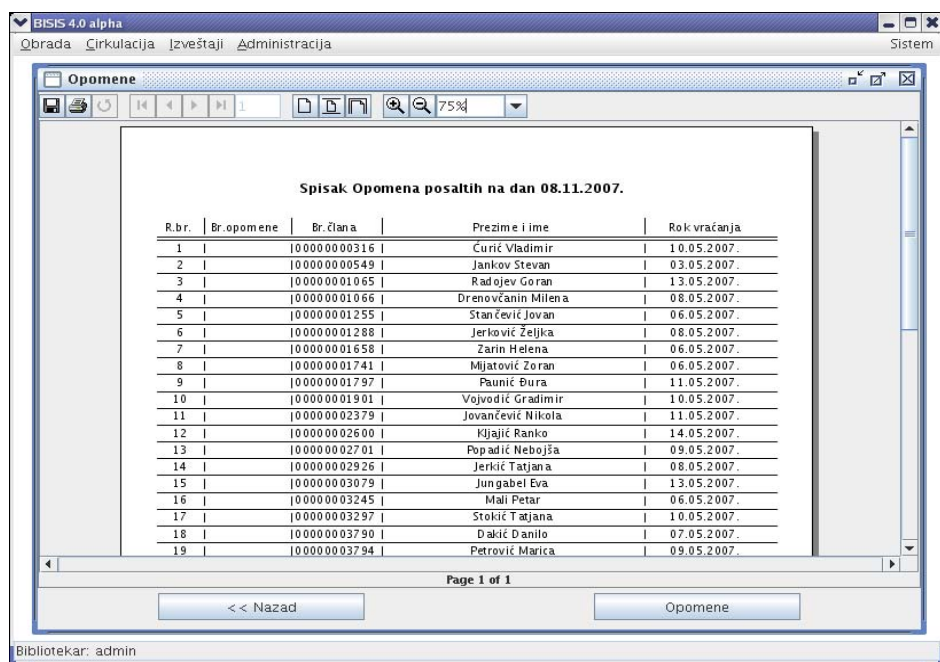
- null	0.00
- nadoknada za zakašnjenje: 50 din. za svaki mesec	0.00

Page 1 of 20

<< Nazad Spisak

Bibliotekar: admin

Слика 4.19 Изгенерисане опомене



Слика 4.20 Списак опомена

4.7.1 Измена текста опомене

На основној екранској форми за генерисање опомена приказаној на слици 4.17 дугметом *Izmeni tekst* се иницира акција измене текста претходно изабране врсте опомене из падајуће листе. Иницирањем акције се отвара екранска форма приказана на слици 4.21. У листи на левој страни екранске форме су наведени делови опомене. Избором неког дела из листе на левој страни, на десној се приказују текстуална поља са текстовима који се налазе на том делу опомене. На слици 4.21 су приказана текстуална поља са текстовима који се налазе у заглављу опомене, а на слици 4.22 текстуална поља са текстовима који се налазе изнад и испод табеле са публикацијама. Слично изгледају и текстуална поља за остале делове опомене. Поред измене текста опомене постоји и опција за избор писма на ком ће се штампати подаци о корисницима и публикацијама из базе података. Дугме *Sačuvaj* иницира акцију чувања измена, а дугме *Nazad* враћа на почетну екранску форму приказану на слици 4.17.

BIBIS 4.0 alpha

Obrada Cirkulacija Izveštaji Administracija Sistem

Opomene

- zaglavlje
- tekst iznad i ispod tabele
- zaglavlje tabele
- nadoknade i vrednost knji
- podaci o korisniku
- direktor

Biblioteka Departmana za matematiku i informatiku

Departman za matematiku i informatiku

Prirodno – matematički fakultet

Trg Dositeja Obradovića 3

21000 Novi Sad

OPOMENA

rok vraćanja:

Podaci iz baze se prikazuju na:

☒ latinici

☐ ćirilici

< Nazad Sačuvaj

Bibliotekar: admin

Слика 4.21 Измена текста заглавља опомене

BIBIS 4.0 alpha

Obrada Cirkulacija Izveštaji Administracija Sistem

Opomene

- zaglavlje
- tekst iznad i ispod tabele
- zaglavlje tabele
- nadoknade i vrednost knji
- podaci o korisniku
- direktor

Uvidom u našu evidenciju ustanovili smo da niste u predviđenom roku vratili pozajmljene knjigu-e:

Molimo vas da svoju obavezu ispunite u roku od 3 (tri) dana kako ne bi došlo do primene drugih predviđenih mera, kao i da prilikom vraćanja knjiga u Biblioteku Departmana za matematiku i informatiku, Trg Dositeja Obradovića 3, izvršite obavezu nadoknade troškova u iznosu:

Napomena: Obavezno poneti i ovu opomenu.

Podaci iz baze se prikazuju na:

☒ latinici

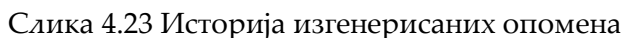
☐ ćirilici

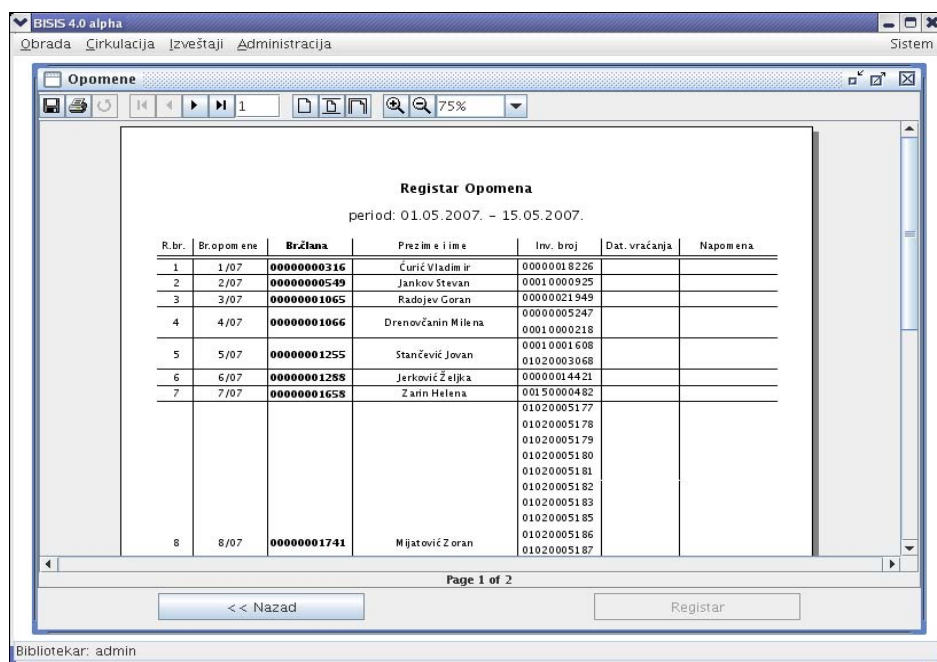
< Nazad Sačuvaj

Bibliotekar: admin

Слика 4.22 Измена текста изнад и испод табеле

На основној екранској форми за генерисање опомена приказаној на слици 4.17 дугметом *Istorija* се приказује листа изгенерисаних опомена за претходно изабране врсту опомене, одељење и период. Опомене се приказују у табели приказаној на слици 4.23. Сваки ред табеле представља једну публикацију са опомене. За сваку публикацију се приказује број, име и презиме корисника код ког је публикација задужена, број и врста опомене на којој се публикација налази и датум слања опомене, као и рок враћања публикације, датум враћања ако је публикација враћена и напомена коју библиотекар уноси приликом раздуживања корисника. Унос напомене је описан даље у тексту. Дугме *Registar* иницира акцију генерисања регистра опомена који је приказан на слици 4.24, а дугме *Nazad* враћа на почетну екранску форму приказану на слици 4.17. Регистар опомена је документ који садржи податке о послатим опоменама у неком периоду.





Слика 4.24 Регистар опомена

4.7.3 Раздуживање корисника којима је послата опомена

У случају када је кориснику послата опомена и када је информација о томе сачувана, на екранској форми за задуживање корисника се приказује индикација о томе. На слици 4.25 је приказан изглед екранске форме за рад са корисницима са отвореном картицом *Zaduženja* на којој се види индикација о послатој опомени кориснику. На дугме *Opomene* се отвара додатна картица *Opomene* на којој су у табели приказане публикације за које је кориснику послата опомена (слика 4.26). За сваку публикацију библиотекар има могућност да унесе напомену која је после видљива у регисту опомена.

BISIS 4.0 alpha

Обрада Циркулација Извештаји Администрација Sistem

Циркулација

Osnovni podaci Dodatni podaci Članarina **Zaduženja**

Info

Korisnik 00000001066, Milena Drenovčanin Istekla članarina

Članarina važi do Feb 24, 1999

Napomena

Zaduži

Inventarni broj  

Poslate opomene!

 Opomene

Zaduženja

Inv. broj	Autor	Naslov	Signatura	Datum zaduže...	Datum produže...	Rok vraćanja
00000022156	Pap, Endre	Fazi mere i nji...	M-22156	Jun 6, 2007		Jun 26, 2007
00000022252	Hecceg, Drago...	Numeričke i st...	M-22252	Dec 28, 2005		Jan 17, 2006
00010000218	Đurić, Branko	Fizika	M1-218	Apr 18, 2007		May 8, 2007
00000015959	Hecceg, Drago...	Numeričke me...	M-15959	Nov 3, 2005		Nov 23, 2005
00000005247	Sears, Francis W.	Mehanika, tala...	M-5247	Apr 18, 2007		May 8, 2007

 Revers  Istorija

 Sačuvaj  Odustani

Bibliotekar: admin

Слика 4.25 Картица *Zaduženja*

BISIS 4.0 alpha

Обрада Циркулација Извештаји Администрација Sistem

Циркулација

Osnovni podaci Dodatni podaci Članarina **Zaduženja** **Opomene**

Broj opomene	Tip opomene	Datum slanja	Rok vraćanja	Inventarni broj	Napomena
23/07	Opomena	Nov 8, 2007	May 8, 2007	00010000218	
4/07	Opomena	Nov 8, 2007	May 8, 2007	00010000218	
23/07	Opomena	Nov 8, 2007	May 8, 2007	00000005247	
4/07	Opomena	Nov 8, 2007	May 8, 2007	00000005247	

 Sačuvaj  Odustani

Bibliotekar: admin

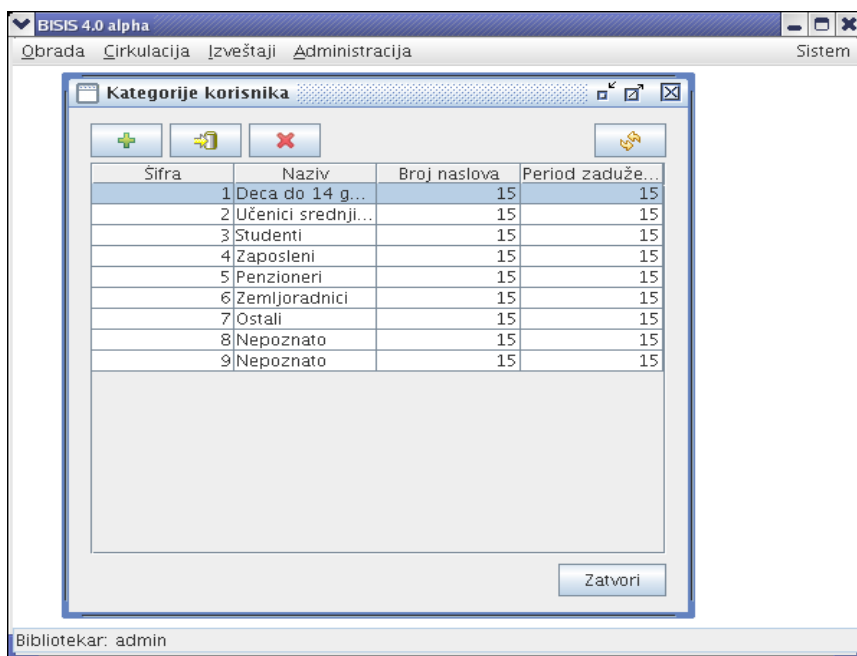
Слика 4.26 Картица *Opomene*

4.8 Администрација система

Акције везане за администрацију система се иницирају из менија *Administracija* који је доступан само оним библиотекарима који имају администраторске привилегије на систему. У те акције спадају администрација шифарника и администрација параметара система.

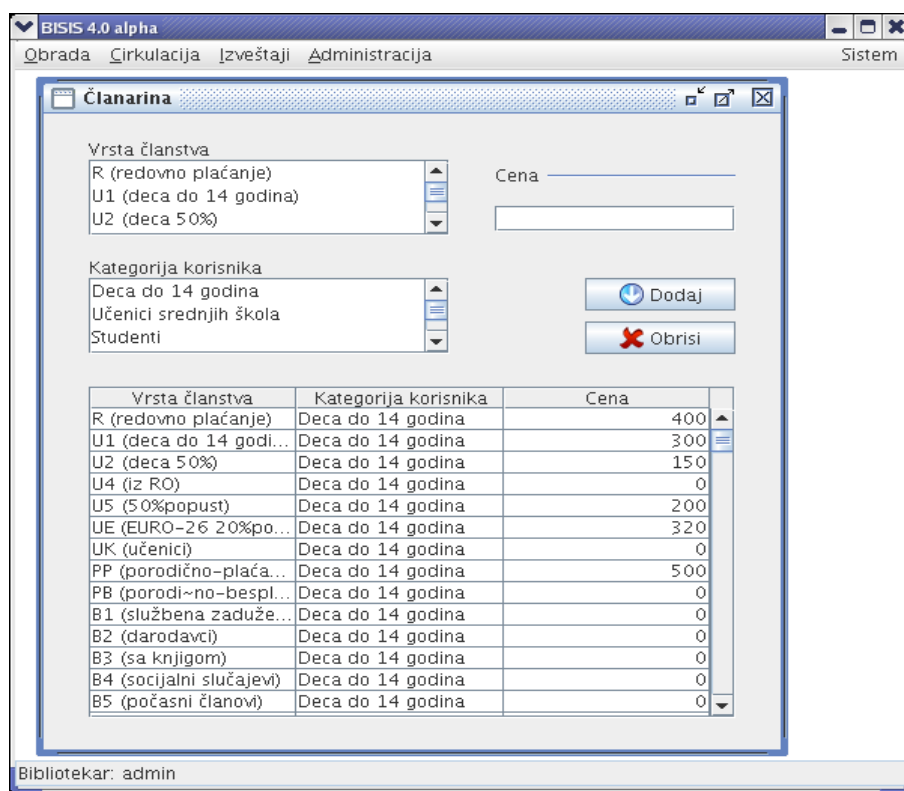
4.8.1 Администрација шифарника

У делу *Šifarnici* менија *Administracija* за сваки шифарник се налази по једна акција која отвара екранску форму за администрацију шифарника. Постоји осам шифарника које је могуће администрирати. То су шифарници за категорију корисника, врсту чланства, чланарину, организацију, степен образовања, матерњи језик, одељење, место. Шифарник за чланарину зависи од категорије корисника и врсте чланства, па се за администрацију тог шифарника користи посебна екранска форма у односу на администрацију других шифарника.



Слика 4.27 Администрација шифарника

На слици 4.27 приказана је екранска форма за администрацију шифарника за категорију корисника. У табели се приказује садржај шифарника. На дугме са сликом  се додаје нови ред у табелу и приликом тог процеса се од корисника очекује да унесе податке. На дугме са сликом  се иницира акција чувања података, а на дугме са сликом  се брише селектовани ред из табеле. Преко ове екранске форме се врши и администрација свих осталих шифарника сем шифарника за чланарину.



Vrsta članstva	Kategorija korisnika	Cena
R (redovno plaćanje)	Deca do 14 godina	400
U1 (deca do 14 godina)	Deca do 14 godina	300
U2 (deca 50%)	Deca do 14 godina	150
U4 (iz RO)	Deca do 14 godina	0
U5 (50%popust)	Deca do 14 godina	200
UE (EURO-26 20%po...	Deca do 14 godina	320
UK (učenici)	Deca do 14 godina	0
PP (porodično-plaća...	Deca do 14 godina	500
PB (porodično-bespl...	Deca do 14 godina	0
B1 (službena zaduže...	Deca do 14 godina	0
B2 (darodavci)	Deca do 14 godina	0
B3 (sa knjigom)	Deca do 14 godina	0
B4 (socijalni slučajevi)	Deca do 14 godina	0
B5 (počasni članovi)	Deca do 14 godina	0

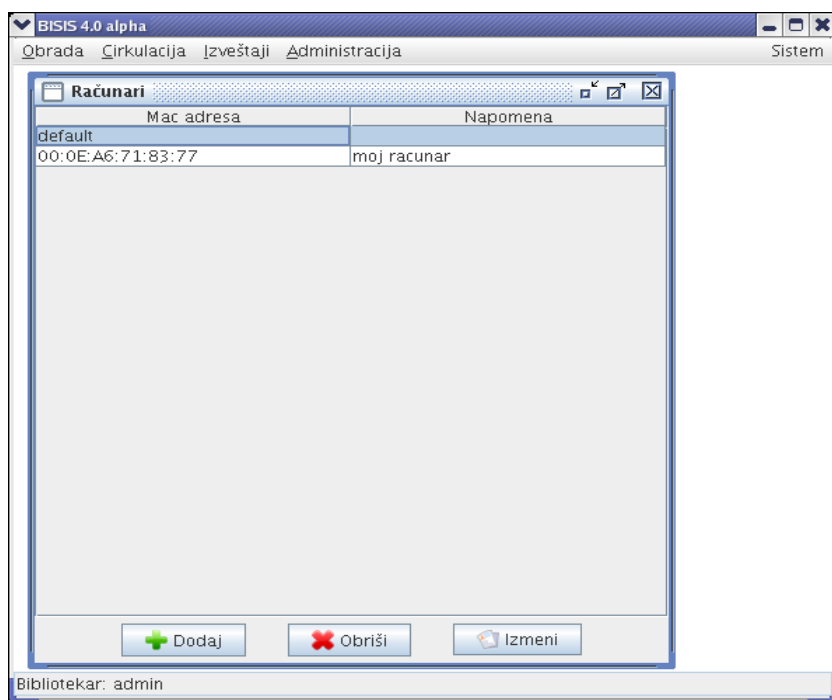
Слика 4.28 Администрација шифарника за чланарину

На слици 4.28 приказана је екранска форма за администрацију шифарника за чланарину. Како чланарина зависи од категорије корисника и врсте чланства да би се дефинисала нова висина чланарине потребно је из одговарајућих листи селектовати категорију и врсту чланства за чланарину која се дефинише. У

текстуално поље цена се уноси висина чланарине, а на дугме *Dodaj* се иницира акција чувања података. У табели на доњем делу форме приказује се преглед дефинисаних чланарина. Селектован ред у табели је могуће обрисати на дугме *Obriši*.

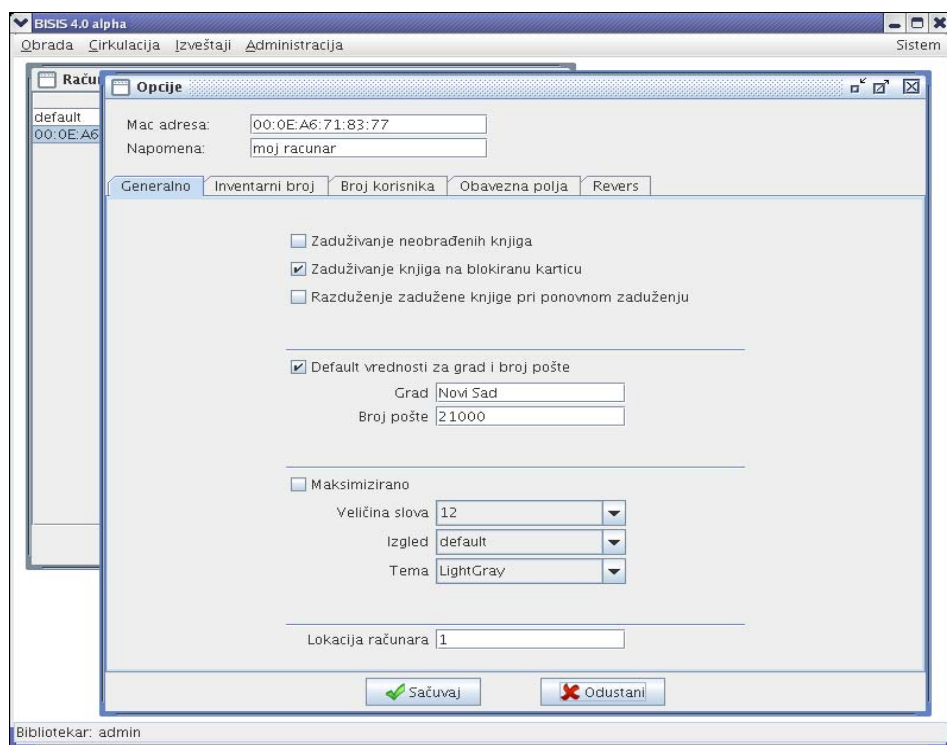
4.8.2 Параметри система

Подешавање параметара система се иницира акцијом *Opције* из менија *Administracija*. Неки параметри се дефинишу за сваког клијента појединачно, а неки важе за све клијенте. Клијенти се идентификују својом MAC адресом. Иницирањем акције отвара се екранска форма приказана на слици 4.29. У табели на форми приказане су адресе клијената за које постоје подешавања. Уколико у систему постоји клијент за кога не постоје подешавања, тај клијент користи подешавања дефинисана за адресу *default*. На дугме *Izmeni* се отвара екранска форма за подешавање параметара за одабраног клијента из табеле.



Слика 4.29 Адресе клијената

На слици 4.30 приказана је екранска форма за подешавање параметара система. У заглављу форме се налазе МАС адреса рачунара и напомена. Подешавања су распоређена на пет картица. На картици *Generalno* се налазе параметри која се односе на разне акције система. На картици *Inventarni broj* се налазе параметри везани за формат и унос инвентарног броја, а на картици *Broj korisnika* параметри везани за формат и унос броја корисника. Картица *Obavezna polja* садржи информације о обавезности уноса података о кориснику. Картица *Revers* садржи параметре везане за реверс који се штампа при задуживању корисника. Сви параметри сем параметара на картици *Obavezna polja* су везани за појединачног клијента, док се параметри са картице *Obavezna polja* односе на све клијенте у систему.



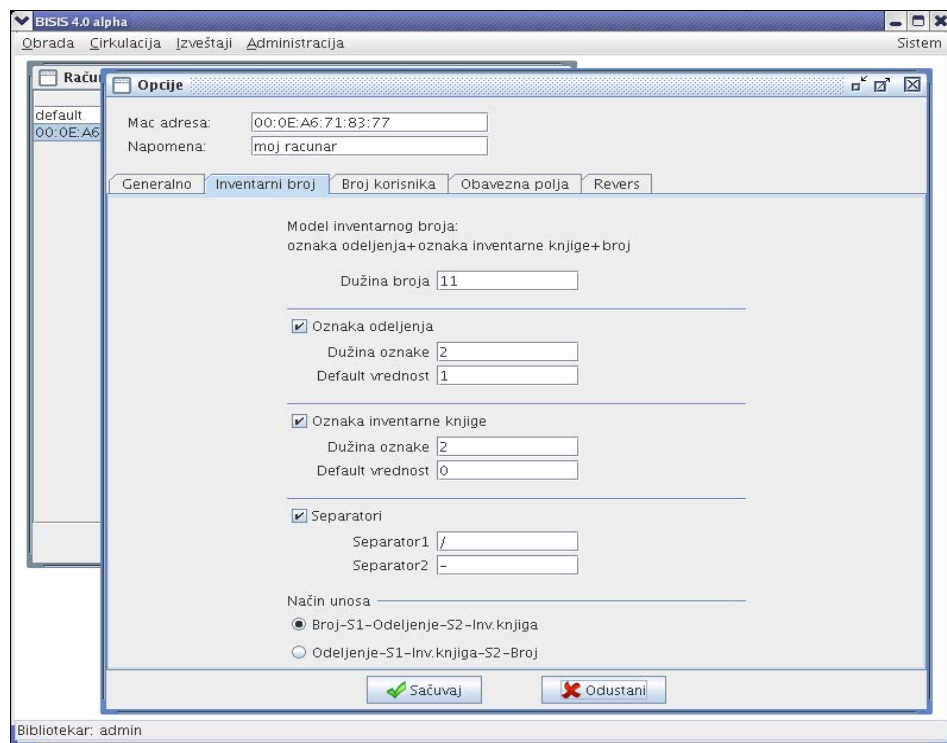
Слика 4.30 Подешавање параметара

На слици 4.30 приказана је екранска форма за подешавање параметара на којој је видљива картица *Generalno*. Параметар *Zaduživanje neobrađenih knjiga* омогућава задуживање књига које не постоје у бази записа. При коришћењу ове опције не постоји могућност било какве провере инвентарног броја која иначе постоји и због тога није препоручљиво да се користи. Ову опцију користе библиотеке које немају потпуно обрађен фонд, а издају књиге. Параметар *Zaduživanje knjiga na blokiranu kartici* омогућава да се блокирање картице користи само као индикатор и да се корисник коме је блокирана картица може даље задуживати и раздуживати. Параметар *Razduženje zadužene knjige pri ponovnom zaduženju* омогућава да се библиотекару приликом задуживања већ задужене књиге понуди опција да се та књига аутоматски раздужи од корисника код кога је задужена, а задужи на новог корисника. У супротном библиотекар мора прво пронаћи корисника код кога је књига задужена и раздужити га да би могао књигу задужити другом кориснику. Параметри везани за град и број поште омогућавају да при упису новог корисника аутоматски буду попуњена поља за град и број поште вредностима дефинисаним овим параметрима. Параметар *Maksimizirano* основну екранску форму апликације отвара максимизирану, а параметри *Veličina slova*, *Izgled* и *Tema* одређују величину слова и дизајн апликације. Параметар *Lokacija računara* садржи шифру одељења на коме се налази клијент и та шифра се користи при задуживању књига као податак на ком се одељењу врши задуживање.

На слици 4.31 је приказана екранска форма за подешавање параметара на којој је видљива картица *Inventarni broj*. На овој картици се уносе параметри везани за формат инвентарног броја који се уноси при задуживању. Инвентарни број може да се састоји од ознаке одељења, ознаке инвентарне књиге и самог инвентарног броја публикације. На овој картици се подешавају дужина инвентарног броја, да ли се користе ознаке одељења и инвентарне књиге и ако се користе које су дужине и која им је *default* вредност, као и редослед којим се уносе и сепаратори који се користе при том уносу.

На слици 4.32 је приказана екранска форма за подешавање параметара на којој је видљива картица *Broj korisnika*. На овој

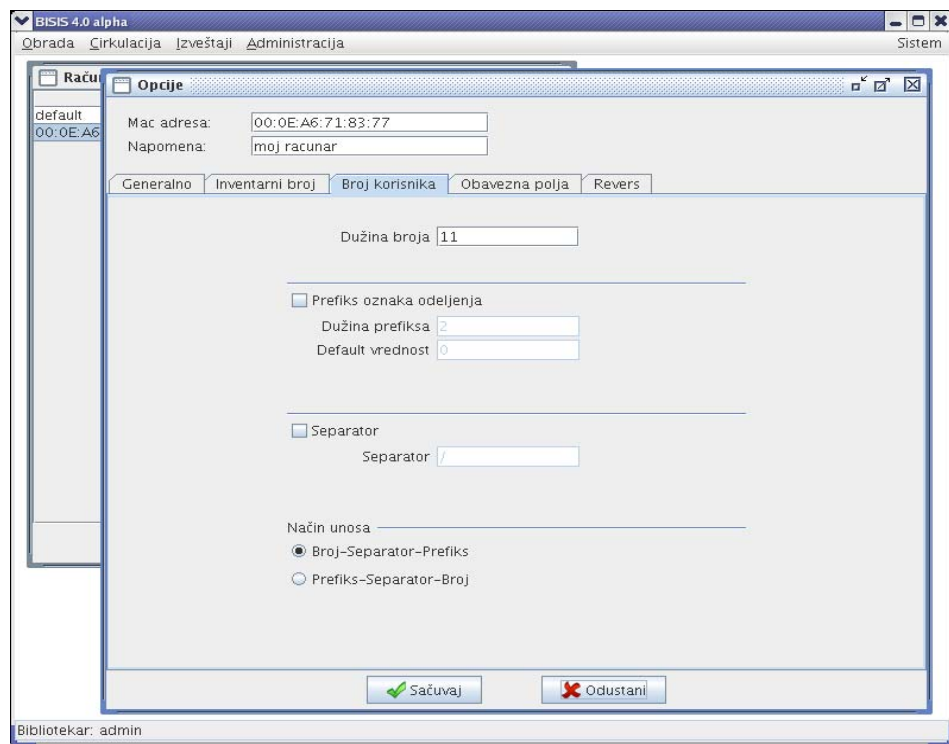
картици се уносе параметри везани за формат и начин уноса броја корисника. Број корисника се састоји од ознаке одељења и броја. На овој картици се подешавају дужина броја корисника, да ли се користи ознака одељења и ако се користи које је дужине и која јој је *default* вредност. Такође, дефинише се и редослед којим се уносе одељење и број, као и сепаратор који се користи при том уносу.



Слика 4.31 Подешавање параметара – картица *Inventarni broj*

На картици *Obavezna polja* се дефинише која поља са форме за рад са корисницима су обавезна за унос. На основу ове дефиниције се врши валидација података о кориснику при чувању података, као и обележавање имена поља са * као индикације обавезности уноса. Ова подешавања су јединствена за све клијенте и измена подешавања на једном клијенту се рефлектује и на све остале.

На картици *Revers* се налазе параметри везани за реверс који се штампа при задуживању корисника. Ту спадају назив и адреса библиотеке, као и формат папира на који се штампа.



Слика 4.32 Подешавање параметара – картица *Broj korisnika*

Литература

[Aleph] ALEPH 500 homepage,
<http://www.exlibrisgroup.com/aleph.htm> (14. јануар 2007)

[Atrium] Atrium homepage,
<http://www.booksys.com/v2/products/atrium/> (17. јануар 2007)

[Budimir04] Будимир, Г., Сурла Д., *Систем за контролу квалитета XML библиографских записа*, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2004.

[Concourse] Concourse homepage,
<http://www.booksys.com/v2/products/concourse/> (17. јануар 2007)

[CyberTools] Cybertools for Libraries homepage,
<http://www.cybertoolsforlibraries.com/> (21. јануар 2007)

[Dimic07a] Димић, Б., *XML едитор за обраду библиографске грађе*, магистарска теза, Природно-математички факултет, Нови Сад

[Dimic07b] Димић, Б., Сурла, Д., *Монографија, Обрада библиографске грађе у софтверском систему БИСИС*, Природно – математички факултет, Нови Сад, 2007

[Eclipse] Eclipse SDK v3.2.1, Eclipse homepage,
<http://www.eclipse.org/> (15. октобар 2006)

[EOS] EOS.Web homepage,
[http://www.eosintl.com /](http://www.eosintl.com/) (21. јануар 2007)

[Evergreen] Evergreen Integrated Library System homepage,
<http://open-ils.org/> (25. јануар 2007)

[Hibernate] Hibernate v3.2.2, Hibernate homepage,
<http://www.hibernate.org/> (12. децембар 2006)

[Java] Java homepage, <http://java.sun.com/> (23. јул 2006)

[Koha] Koha Integrated Library System homepage,
<http://koha.org/> (30. јануар 2007)

[Lazarevic96] Paunović, Đ., Lazarević, B., Surla, D., Ivković, M., Korunović, D., *Formiranje i pretraživanje baza podataka u sistemu naučnih i tehnoloških informacija Srbije*, editor Branislav Lazarević, Ministarstvo za nauku i tehnologiju Republike Srbije, Beograd, 1996.

[LibLime] LibLime homepage,
<http://liblime.com/> (30. јануар 2007)

[Mandarin] Mandarin Library Automation homepage,
<http://mlasolutions.com/> (25. јануар 2007)

[Marc] Marc Standards, <http://www.loc.gov/marc/> (14. јануар 2007)

[Mijic03] В. Мијић. *XML едитор за опис UNIMARC формата*, магистарска теза, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2003.

[Millenium] Millenium homepage,
<http://www.iii.com/mill/index.shtml/> (21. јануар 2007)

[Milosavljevic07] Milosavljević B., Dimić B., *XML schema of UNIMARC format variant and bibliographic record in BISIS software system*, NSJOM, Vol. 37, No.1, 2007, pp 115-128.

[Milosavljevic97] Milosavljević, B., Surla, D., *Implementacija tekst servera za indeksiranje i pretraživanje bibliotečke građe u Oracle okruženju*, Zbornik radova Info-Teh'97, Vrnjačka Banja, 1997. str. 452-456.

[MySQL] MySQL homepage,
<http://www.mysql.com/> (25. септембар 2006)

[Polaris] Polaris homepage,
<http://www.polarislibrary.com/> (25. јануар 2007)

[Pupovac] Pupovac, B., Surla, D., *Model slučajeve korišćenja komandi bibliotečkog informacionog sistema*, CD-ROM sa Simpozijuma o računarskim naukama i informacionim tehnologijama YU INFO'01, Kopaonik, 2001.

[Radjenovic06a] Rađenović J., Milosavljević B., Surla D., *Generating catalog cards by the BISIS library software*, INFOTEKA, 7(2006)1-2, str. 61-74

[Radjenovic06b] Рађеновић Ј., Сурла, Д., Милосављевић Б., *Софтверски пакет за генерисање библиотечких каталошких листића* (монографија), Иновациони центар за електронске библиотеке и архиве, Природно-математички факултет, Департман за математику и информатику, Нови Сад, 2006.

[RFID] Radio Frequency Identification,
<http://www.rfid.org/> (14. јануар 2007)

[Sagebrush] Sagebrush Corporation homepage,
<http://www.sagebrushcorp.com/> (25. јануар 2007)

[SIP] 3M Standard Interchange Protocol ver.2,
http://solutions.3m.com/wps/portal/3M/en_GB/library/home/documents/protocols/ (14. јануар 2007)

[Skrbic04] Škrbić, S., Surla D., *Upravljanje XML bibliografskim zapisima u XML tehnologiji*, XXXI Simpozijum o operacionim istraživanjima SYM-OP-IS 2004, Iriški Venac 2004. pp133-136

[Skrbic05] Škrbić, S., Surla D., *Obrada bibliografske građe u XML native tehnologiji*, PMF, Univerzitet u Novom Sadu, 2005.

[Surla00] D. Surla, Z. Konjović, B. Milosavljević, M. Vidaković. *Bibliotečki informacioni sistem BISIS ver. 3.01*, Deveta međunarodna konferencija "Informatika u obrazovanju, kvalitet i nove informacione tehnologije", Zrenjanin 2000., str 494-504.

[Surla03] Д. Сурла, З. Коњовић, Б. Пуповац, Б. Милосављевић, М. Видаковић, Т. Тошић, Т. Зубић. *Упутство за коришћење библиотечног софтверског система БИСИС вер. 3*, Група за Информационе технологије, Нови Сад, 2003.

[Surla04a] Монографија *Дистрибуирани библиотечки информациони систем БИСИС*, уредници Д. Сурла, З. Коњовић, Група за Информационе технологије, Нови Сад, 2004.

[Surla04b] Surla, D., Konjović, Z., *Overwiev of the Development of the Library Information System BISIS*, Proc. of International Conference on Distributed Library Information Systems, Ohrid, FYROM, 2004, pp.13-18.

[Surla04c] Сурла, Д., Коњовић, З., Милосављевић, Б., Зарић, М., Сладић, Г., Протић, З., Комазец, С., Окановић, Д., *Приказ реализације мрежне дигиталне библиотеке докторских, магистарских и дипломских радова*, Инфотека, 5 (1-2), 2004.

[Surla96] Surla, D., Todoric, D., *Presentation of Searching Report of the Documents Database in World Wide Web*, East-West International Conference Information Technology in Design, EWITD'96, Moscow, 1996. pp. 11-16.

[Surla98] Сурла, Д., Тодорић, Д., *Библиотека на Интернету*, поглавље 7, (191-216) у монографији *"Интернет и савремено пословање"*, редактори др Миодраг Ивковић и др Божа Раденковић, Технички факултет "Михајло Пупин", Зрењанин, 1998

[Tesendic] Tešendić, D., Surla, D., *XML Schema for members' data*, Zbornik sažetaka PRIM 2006, Kragujevac, str. 22.

[Todoric] Тодорић, Д., Сурла, Д., *Функције сервера за узајамну каталогизацију библиографских записа*, Симпозијум о рачунарским наукама у информатици YU INFO'98, Копаоник, 1998. стр. 267-272.

[Tosic] Тошић, Т., Surla, D., *Korišćenje bibliotečke građe*, Monografija: Distribuirani bibliotečki informacioni sistem BISIS (Poglavlje 10), Grupa za informacione tehnologije, Novi Sad, 2004.

[Validator] Commons Validator homepage,
<http://jakarta.apache.org/commons/validator/> (13. септембар 2006)

[VisParadigm] Visual Paradigm SDE for Eclipse, v.4.0,
<http://www.visual-paradigm.com/product/sde/ec/> (8. новембар 2006)

[Voyager] Voyager homepage,
<http://www.exlibris-usa.com/voyager.htm> (17. јануар 2007)

[Zaric03] Зарић, М. Сурла, Д., Java implementation of the Protocol for Metadata Harvesting. In *Proceedings of the 3d International Conference on Informatics and Information Technology*, Bitola 2003.

[Zaric04a] Зарић, М. Сурла, Д., Metadata Dissemination using OAI-PMH. In *Proceedings of the International Conference on Distributed Library Information Systems*, Охрид, 2004.

[Zaric04b] Зарић, М. Сурла, Имплементација протокола за прикупљање метаподатака у мрежи дигиталних библиотека. *Инфотека* 5 (1-2):nn. 99-112

[Zeremski01a] Zeremski, M., Surla, D., *Modeliranje UNIMARC formata u XML – Data jeziku*, Zbornik radova sa Simpozijuma o računarskim naukama i informacionim tehnologijama YU INFO 2001 (na CD-ROM-u), Kopaonik, 2001.

[Zeremski01b] Zeremski, M., Surla, D., *Modeliranje UNIMARC formata pomoću XML Schema jezika*, Zbornik radova sa XXVIII Jugoslovenskog simpozijuma o operacionim istraživanjima SYM-OP-IS 2001, Beograd, 2001. str. 261-264.

[Zeremski01c] Зеремски, М., Сурла, Д., *Моделирање библиографских записа у облику XML докумената*, ИНФОТЕКА: Часопис за информатику и библиотекарство, Београд, 2(2001) 1-2, стр. 93-97.

[Zeremski01d] Zeremski, M., Surla, D., *Modeliranje bibliografskih zapisa pomoću XML Schema jezika*, Zbornik radova sa XXVIII Jugoslovenskog simpozijuma o operacionim istraživanjima SYM-OP-IS 2001, Beograd, 2001. str. 265-268.

[Zeremski02] Zeremski, M., Surla, D., *Validacija bibliografskih zapisa pomoću XML Schema jezika*, Zbornik radova sa Simpozijuma o računarskim naukama i informacionim tehnologijama YU INFO 2002 (na CD-ROM-u), Kopaonik, 2002.

[Zeremski04] Zeremski, M., Surla, D., *Validation of XML bibliographic records in Java environment*, Novi Sad Journal of Mathematics, 2004.

[Zubic] Зубић, Т., Сурла, Д., *Објектно оријентисано моделирање извештавања у библиотечком информационом систему*, CD ROM Зборник радова YU INFO'2000, Копаоник, 2000.

CIP – Каталогизација у публикацији
Библиотека Матице српске, Нови Сад

025.6:004.42BISIS

ТЕШЕНДИЋ, Данијела

Коришћење библиотечке грађе у софтверском систему
БИСИС / Данијела Тешендић, Душан Сурла. – Нови Сад :
Природно-математички факултет, Департман за математику и
информатику, 2007 (Нови Сад : АБМ економик). - 130 стр. :
илустр. ; 24 cm

Тираж 150. - Библиографија. - Abstrakt.

ISBN 978-86-7031-135-0

1. Сурла, Душан

а) Библиотечко-информациони систем БИСИС – Библиотечки
фондови - Позајмница

COBISS.SR-ID 226790407

Мр Данијела Тешендић је истраживач приправник Природно-математичког факултета у Новом Саду. Рођена је (1979) у Градачцу, БиХ. Дипломирала је (2004) на Природно-математичком факултету у Новом Саду, где је (2007) одбранила и магистарску тезу. Од децембра 2005. године запослена је на Природно-математичком факултету у Новом Саду и ангажована је на пројекту Апстрактни модели и примена у рачунарским наукама (бр. 144017) који финансира Министарство науке и заштите животне средине Републике Србије. Има три објављена научна рада који припадају области пројектовања библиотечких информационих система.

Др Душан Сурла је редовни професор Природно-математичког факултета у Новом Саду. Рођен је (1942) у Омсици, Република Хрватска. Дипломирао је (1969) на Филозофском факултету у Новом Саду. Магистрирао је (1976) на Машинском факултету у Новом Саду, а докторирао (1980) на новосадском Факултету техничких наука. На Природно-математичком факултету у Новом Саду ради од 1976. а за редовног професора изабран је (1991) за научну област информатика. Објавио је око 200 библиографских јединица из области роботике и области информациони системи.

ISBN 978-86-7031-135-0



9 788670 311350