Logo

Description automatically generatedУниверзитет у Београду

Електротехнички факултет

Конкурентно и дистрибуирано програмирање

Дистрибуирани документи

Владимир Радомировиц 0474/2018

*Садржај*

[1. Текст задатка-Дистрибуирани документи 2](#_Toc63047453)

[2. Предложено решење 3](#_Toc63047454)

[3. UML - Дијаграми интеракције 4](#_Toc63047455)

[4. Упутство за коришћење програма 7](#_Toc63047456)

[4.1. Програм за клиенте 7](#_Toc63047457)

[4.2. Програм за подсервере 9](#_Toc63047458)

[4.3. Програм за главни сервер 10](#_Toc63047459)

[5. Примери рада програма у регуларним и ванредним ситуацијама 12](#_Toc63047460)

Прва верзија:11.1.2021.

Текућа верзиа:6.2.2021.

Место: Рачунски центар Електротехничког факултета

# **Текст задатка-Дистрибуирани документи**

Пројектовати дистрибуирани рачунарски систем који треба да омогући синхронизацију докумената на већем броју рачунара. Програм треба да ради у систему који се састоји од више рачунара повезаних у LAN (*Local Area Network*) или WAN (*Wide Area Network*). У систему постоји три типа програма:

1. Централни сервер који служи за чување оригиналних верзија докумената и информација о распрострањености копија докумената.
2. Подсервери на којима се чувају копије докумената, чији се оригинали налазе на централном серверу.
3. Клијентски програм преко кога клијент може да приступа документима на подсерверима.

Када клијентски рачунар пошаље захтев за неким документом неком од подсервера, тај подсервер проверава да ли се тражени документ налази код њега. Уколико се документ налази на подсерверу, тада подсервер кориснику враћа тражени документ. Уколико се документ не налази на подсерверу, онда подсервер од централног сервера тражи да му пошаље копију документа. Када на подсервер пристигне копија документа, она се памти на подсерверу, а такође се прослеђује и кориснику који је тражио документ.

Централи сервер периодично мења садржаје оригиналних докумената. Потребно је обезбедити одржање лабаве конзистентности на следећи начин. Када централни сервер утврди да неки документ треба променити он интерно провери на којим се све подсерверима налази тражени документ. Када је сервер утврдио листу подсервера, онда шаље свим подсерверима наредбу за ажурирање копије документа. Да се на серверу не би стално креирале нити које служе за слање захтева за ажурирањем докумената, на централном серверу постоји група нити која обавља овај посао. Број нити је параметар који се задаје приликом рада серверске апликације. Централни сервер не шаље захтеве за ажурирањем директно нитима већ их убацује у јединствени бафер захтева за ажурирањем докумената. Нити узимају захтев по захтев из бафера и прослеђују га одговарајућем подсерверу у редоследу у коме су и пристизали. Информација о успешном ажурирању свих докумената се записује у лог.

Уколико је подсервер недоступан у моменту када се упућује захтев за ажурирањем покушава се поново након *m/3* секунди. Уколико ажурирање свих подсервера на којима се налази копија документа не успе у интервалу од *m* секунди, тај захтев се проглашава необрађеним и то се уписује у лог. Притом се у лог уписују и подсервери на којима није успело ажурирање, али се поступак слања наставља све док се подсервер сматра исправним. Подсервер на коме од тренутка покретања није успело *k* захтева за ажурирањем проглашава се за неисправан и чека се његово поновно покретање. Уколико се неки подсервер не одазове извесно време (*x* секунди) такође се проглашава за неисправан. Неисправном подсерверу се не шаљу захтеви за ажурирањем, нити се одговара на захтеве за страницама док се он не јави централном серверу и потврди да је поновно покренут. Приликом покретања, подсервер брише све копије довучених докумената.

Водити рачуна да подсервери и сервер могу да приступају само документима који се налазе почев од специфицираног директоријума, односно да није дозвољено да се приступа документима који су изван специфицираног директоријума. Специфицирани директоријум може да има структуру произвољног нивоа дубине. Уколико тражени документ не постоји на подсерверима и серверу, кориснику треба вратити страницу/документ са стандардном поруком да се јавила грешка 404. Ову поруку подсервер треба да врати клијенту и у случају да је централни сервер недоступан, а да не поседује копију траженог документа.

Клијенти имају могућност додавања докумената и мењања докумената и њиховог садржаја. Све измене треба да буду видљиве свим клијентима који су затражили документ или га тренутно мењају. Дозвољено је да више клијената истовремено мења документ, при чему клијенти могу да раде неометано (одзив мора бити задовољавајући). Уколико подсервер у међувремену постане недоступан, треба урадити балансирање свих клијената на све доступне подсервере (*load balancing*).

Проблем решити користећи мрежну комуникацију у програмском језику Јава. За сваки од ова три типа рачунара треба да постоји одговарајући графички кориснички интерфејс (GUI треба да буде развијен користећи Java SWING компоненте или JavaFX). Графички кориснички интерфејс мора бити интуитиван. За сваки од могућих случајева коришћења система поруке и обавештења морају бити прецизно приказани. Такође, потребно је да се кроз графички кориснички интерфејс прикажу детаљни логови, како за централни сервер, тако и за подсервере. Сервери треба да имају могућност покретања и без корисничког интерфејса. Такође, IP адресе и портови потребни за функционисање програма (као и остали подаци који се могу динамички мењати) морају се уносити преко графичког интерфејса (не смеју бити унапред дефинисани у изворном коду).

# **Предложено решење**

Идеја за решавање овог проблема свела се на писање 3 групе програма које би међусобно комуницирале као и додатну групу бафера који би служили за складиштењем података, конекција између било које групе би била флексибила тако да престанак рада 2 од 3(изузетак је главни сервер који служи са складиштење свих података, али и даље могу комуницирати подсервери и клијенти) групе не би проузроковала никакве проблеме и сметње даљем раду осталих компоненти.

Групе програма су:

1. Група клијена
2. Група подсервера
3. Главни сервер

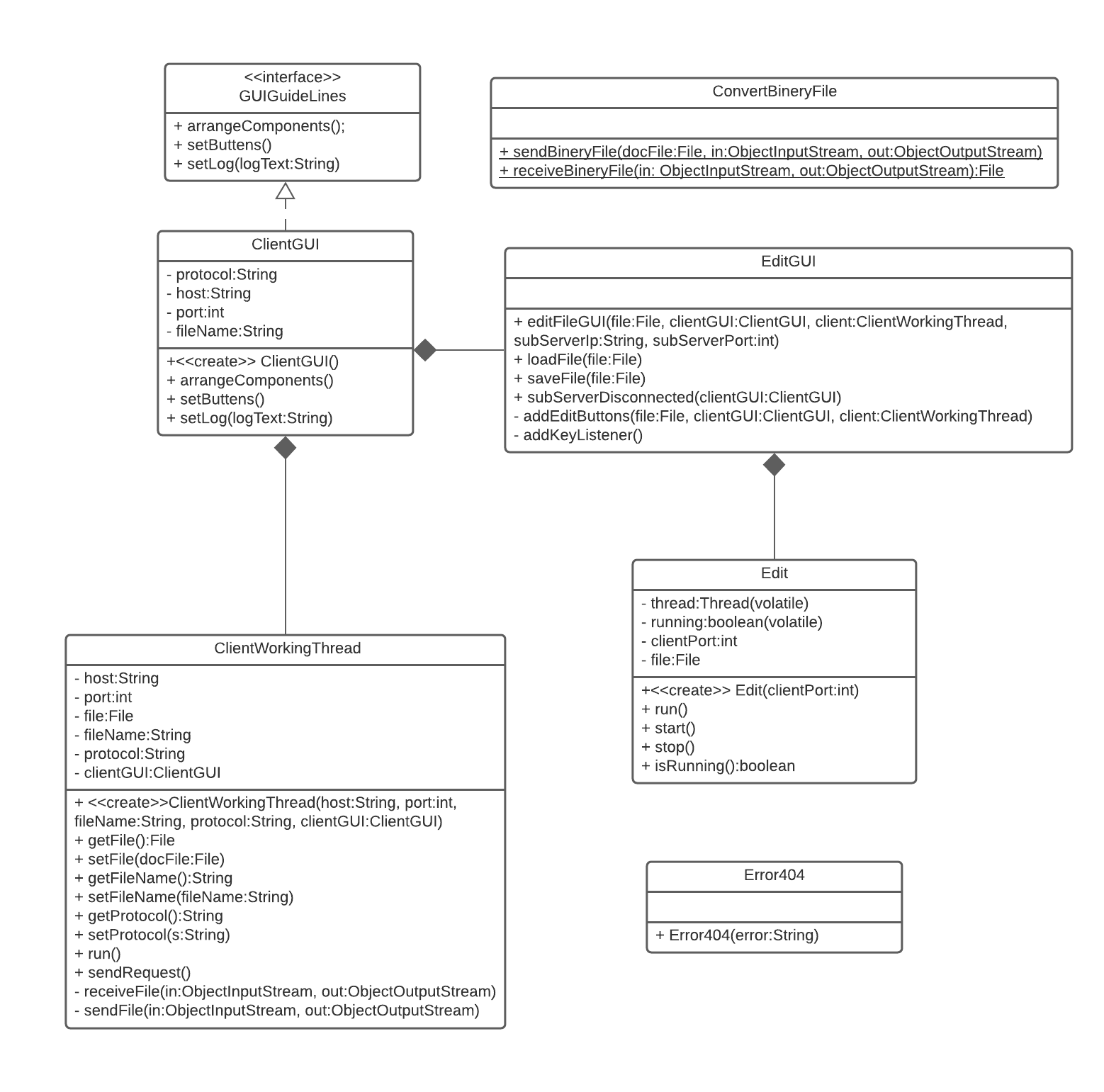
Група клијнта – клиенти би представљали најнижу групу програма, они би слали захтеве серверима за читањем података и слање самих података на подсервере и главни сервер. Клиенти ће се повезивати са подсерверима и слати им захтеве за слањем података или за довлачењем података са сервера. Приликом слања података клиентима ће се исписати обавештења да ли је послати податак успесно сачуван или није, такође у случају да податак под истим именом већ постоји на главном серверу тај податак неће бити сачуван и грешка ће бити исписана. Приликом слања захтева за неким податком сервер ће пријавити поруку да ли се тражени податак налази на неком од сервера, у слуцају да се не налази грешка ће бити исписана, у случају да се налази, податак ће бити довучен са сервера и би ће отворен за обраду, након обраде од стране клиента измењени податак ће поново бити послат на сервер и сачуван.

Група подсервера – подсервери би представљали средњу групу која би повезивала клијенте са главним сервером, примали би захтеве од клиента и комуницирали са главним сервером у случају да је то потребно. Сваки подсервер би имао сопствени бафер за складиштење података и додатних информација потребних за несметану комуникацију са клиентима и главним сервером. Подсервери би примали 2 групе упита, прва група би представљала захтеве које шаљу клијенти, а друга група би представљала ажурирања која би се примала од стране главног сервера. Даље захтеви би се делили у на захтеве за довлачењем података и захтеве за чувањем података. Приликом чувања података податак би се чувао на подсерверу а и послао на главни сервер у случају да главни сервер није доступан или да потатак већ постоји податак не ће био сачуван, приликом захтева података од сервера прво би се проверавало да ли се податак налази на подсерверу у случају да се не налази на подсерверу послао би се упит главном серверу, у случају да нигде није пронађен податак исписала би се грешка, подсервер би примао ажурирања периодично од главног сервера и ажурирања би слао клиентима који у том тренутку поседују тај податак.

Главни сервер – главни сервер би представљао највишу групу програма, он би примао податке од свих подсервера чувао их и главни бафер и преусмеравао те податке оним серверима који га затраже, у баферу би се чували довучени податци и додатне информације везане за довучени податак, такође би имао потребну групу нити која би служила за ажурирање података, додатни бафер у којем ће се гурати потдатци које треба ажурирати и додатна нит која би периодично проверавала и гурала те податке. Главни сервер би примао захтеве од подсервера и по потребио слао одговарајуће податке или поруке.

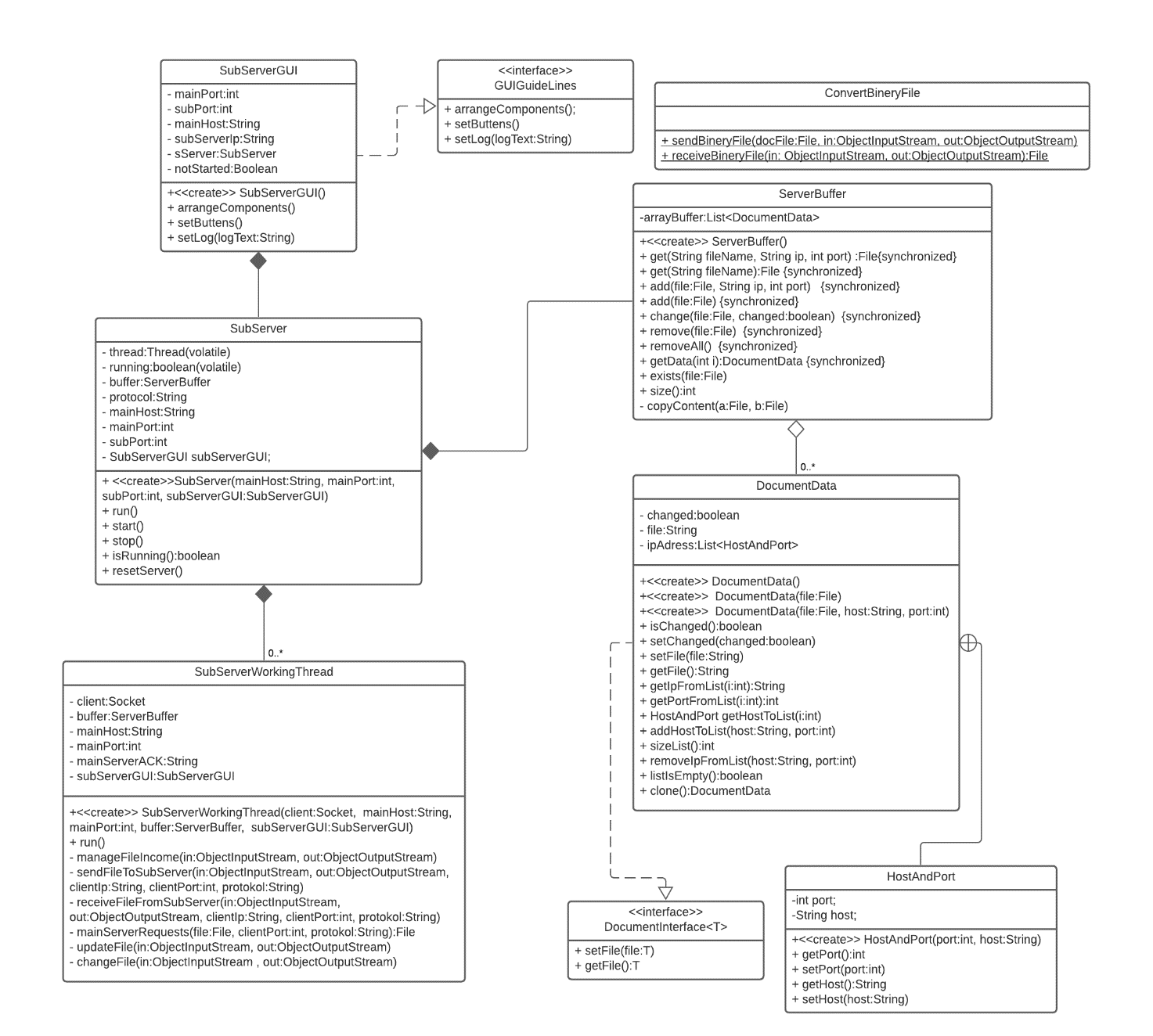
# **UML - Дијаграми интеракције**

UML Дијаграм 1 - Представа групе програма за клијента



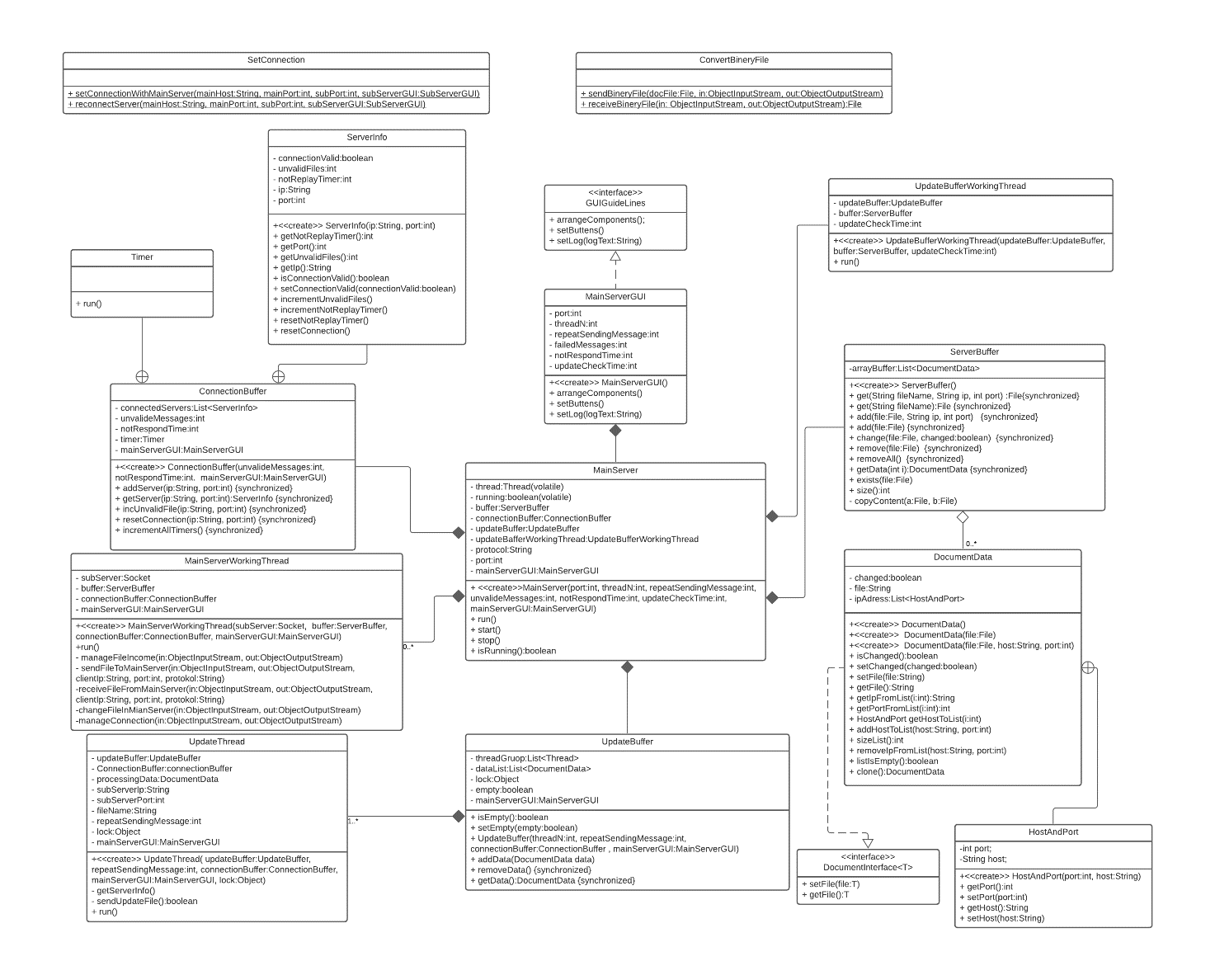
Клијентска група програма састоји се од ClientGUI класе која имплементира интерфејс GUIGuideLines која служи за исцртавање клиентског прозора који се користи за комуникацију са серверима, приликом слања захтева серверу прави се и покреће ClientWorkingThread нит која комуницира са сервером прима и шаље податке, у слуцају да је послат захтев за довлачењем податка са сервера покрећу се команде из EditGUI класе које служе за креирање посебног прозора за обраду података, такође се покреће и Edit нит која ослускује да ли је досло до ажурирања податка који се тренутно обрађује. Класа ConverBineryFile служи само за слање и примање податка од и ка северу, такођ је могуће доћи до греске AlreadyExists у случају да податак који се саље на сервер већ постоји на серверу и греске Error404 ако тражени податак не постоји или није могуће успоставити везу са сервером.

UML Дијаграм 2 - Представа групе програма за подсервер



Подсерверска група програма састаоји се од SubServerGUI класе која имплементира интерфејс GUIGuideLines која служи за исцртавање подсерверског прозора, он служи са само покретање подсервера. Подсервер се покреће креирањем и покретањем SubServer нити, та нит ослушкује билокакав упит који би тренао да јој стигне, у случају да је стигао неки упит креира се нит SubServerWorkingThread која надаље обрађује пристигли податак, такође класа SubServer садржи бафер за складиштење података ServerBuffer који складисти податке у виду DocumentData класа, DocumentData класа имплементира интерфејс DocumentInterface и садржи угнежђену класу HostAndPort. Класа ConverBineryFile служи само за слање и примање податка. Команде из SetConnection класе слже за слање информације главном серверу да је упаљео овај подсервер.

UML Дијаграм 3 - Представа групе програма за глевни сервер



Група програма за главни сервер састоји се од MainServerGUI класе која имплементира интерфејс GUIGuideLines која служи за исцртавање прозора главног сервера који слузи за покретање главног сервера. Приликом покретања главног сервера ствара се и покреће нит MainServer која ослускује захтеве који стижу од сервера. У случају да је пристиго неки захтев ствара се и покреће нит MainServerWorkingThread који обрађује пристигле захтеве. Приликом стварања нити MainServer ствара се и покреће нит UpdateBufferWorkingThread која периодично проверава да ли је дошло до ажурирања некод од података у случају да јесте, тај податак се саље у UpdateBuffer класу која се такодје креира приликом стварања нити MainServer. Та класа садржи одређен број нити UpdateThread које узимају за слање ажурираних података до подсервера који садрже тај податак. Приликом стварања нити MainServer такође се ствара и ConnectionBuffer класа која служи за чување потребних информација о свим подсерверима који су повезани са главним сервером, та класа у себи садржи угнежђену класу ServerInfo која служи за чување свих потребних података о прикљученим подсерверика, као и угнежђену класу Timer која константо откуцава време и проверава колико дуго су посервери неактивни. Такође у главном серверу као и у свим подсерверима постаји ServerBuffer који служи за чување свих пристиглих података.

# **Упутство за коришћење програма**

## **Програм за клиенте**

Као клијент приликом учитавања програма појавиће вам се прозор као приказан на *слици 1 - Клиентски прозор* приликом отварања програма имате избор да ли желите да додате документ на сервере или желите да затражите документ од сервера. У случају да желите да додате документ на сервере потребно је да ознаците радио дугме Dodaj Dokument у пољу host потребно је уписати ip адресу жељеног подсервера на који хоћете да посаљете документ, у пољу port потребно је уписати порт подсерверс на који желите да пошаљете документ и у пољу Upisi lokaciju dokumenta потребно је уписати локацију документа који желите да посаљете серверу.

Graphical user interface, application

Description automatically generated

слика 1 - Клиентски прозор

Приликом додавања документа на сервер у случају да је документ успесно додат на сервер исписује се порука као на  *слици 2 - успешно сачуван податак*, међутим може се десити да документ под истим именом већ постои на серверу онда се исписује поруја као на  *слици 3 - податак већ постои*.

Graphical user interface, text, application

Description automatically generated

слика 2 - успешно сачуван податак

Graphical user interface, text, application, email

Description automatically generated

слика 3 - податак већ постои

У случају да желите да затражите документ од сервера потребно не означити радио дугме Zatrazi Dokument, даље слично као и у претходном случају у пољу host потребно је уписати ip адресу жељеног подсервера са кода хоћете да захтевате документ, у пољу port потребно је уписати порт подсерверс са кога желите да примите документ и у пољу ime zahtevanog dokumenta потребно је уписати име траженог документа.

У случају да сервер није доступан или да податак под уписаним именом не постоји на траженим серверима исписује се греска као на *слици 4 - пријева грешке.*

Shape

Description automatically generated

слика 4 - пријева грешке

У случају да је успешно примљен податак отвара се нови прозор као на *слиci 5 - прозор за измену* који служи за измену документа. У случају да сте задовољни са изменама притиском на дугме exit завршавате измену документа, враћате се на оригинални прозор за клијента и исписује вам се порука *слика 6 - успешно измењен фајл*.

Graphical user interface

Description automatically generated

слика 5 - прозор за измену

Graphical user interface, application

Description automatically generated

слика 6 - успешно измењен фајл

## **Програм за подсервере**

Као подсервер приликом учитавања програма појављује се прозор као на *слика 7 - Главни прозор подсервера*. У пољу Port za ulaz уписује се порт на коме ће подсервер да ослушкује све захтеве, у пољу Port glavnog servera потребно је уписати порт главног сервера преко кога ће подсервер слати податке, а у пољу ip adresa glavnog servera потребно је уписати ип адресу главног сервера са којим се треба повезати. Када је све уписано потребно је покренути сервер сервер притиском на дугме Pokreni server. У случају да је све лепо унето и да је успесно успостављена конекција са главним сервером прозор ће се променити и исписаће се порука да је сервер успешно конектован на главни сервер *слика 8 - Успешно остварена конекција*, дугме Pokreni server ће се променити у дугме Restartuj server које служи за поновно покретање сервера приликом чега неће узимати у обзир нове промене у пољиме.

Table

Description automatically generated

слика 7 - Главни прозор подсервера



слика 8 - Успешно остварена конекција

Приликом доласка и одласка одговарајућих података и измене докумената на исписиваће се одговарајуће поруке везане се рад подсервера *слика 9 - Порука приликом успешно примљеног документа, успесно измењеног документа и неуспешно затраженог документа , слика 10 - Порука приликом слања захтева главном серверу за прослеђивање документа са потврдним одговором*.

Graphical user interface, text, application, email

Description automatically generated

слика 9 - Порука приликом успешно примљеног документа, успесно измењеног документа и неуспешно затраженог документа

Graphical user interface, text, application, email

Description automatically generated

слика 10 - Порука приликом слања захтева главном серверу за прослеђивање документа са потврдним одговором

## **Програм за главни сервер**

Као главни сервер приликом учитавања програма појављује се прозор као на *слици 11 - прозор главног сервера.* У пољу Port za ulaz уписује се порт на коме ће главни сервер да ослушкује захтеве који му стижу од подсервера, у пољу Br. niti уписије се броји нити које се стварају и служе за ажурирање података, пољу Ponovi upit уписује се време након чијег нејављања се неки податак проглашава невалидним, у пољу Br. greski се уписије после којико невалидних података се прогласава неки сервер невалидним, у пољу Gasi Server се уписује после које времена нејављања некод сервера тај сервер се прогласава невалидним, у пољу menjaj sadrzaj се уписује време на које се периодично ажурирају податци у свим серверима. Главни сервер се покреће притиском на дугме Pokreni server. Као и код подсервера и код главног сервера константно се исписују све промене које се дешавају у оквиру главног сервера сви захтеви, поруке и ажурирања података *слика 12 - Порука приликом успешне конекције са подсервером, успешно сачуваног податка,успешно измеженог податка, успешно ажурираног податка,неуспешно пронађеног податка*.

Graphical user interface

Description automatically generated

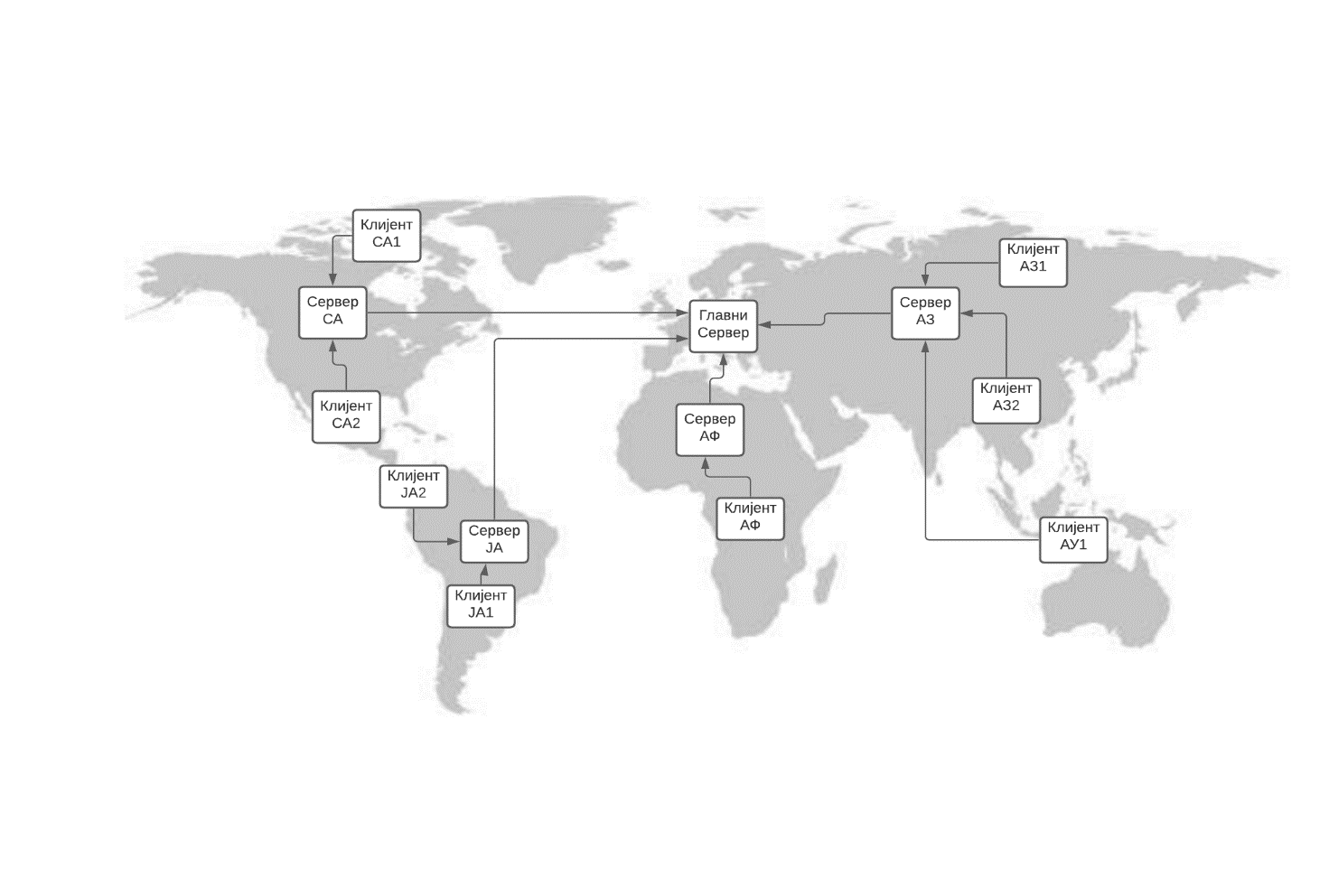
слика 11 - прозор главног сервера

Graphical user interface, text, application

Description automatically generated

слика 12 - Порука приликом успешне конекције са подсервером, успешно сачуваног податка,успешно измеженог податка, успешно ажурираног податка,неуспешно пронађеног податка

# **Примери рада програма у регуларним и ванредним ситуацијама**



слика 13 - пример распореда сервера

Предпоставимо да имамо систем као систем приказан на *слика 13 - пример распореда сервера*. У овом систе су сви подсервери лепо повезани са главним сервером и сви клијенти првенствено комуницирају са означеним подсерверима. У рекуларним ситуацијама све би требало савршено да ради, међутим рецимо да подсервер који се налази у Азији константно добија упите од клијента па због тога је често заузет, може се десити да када од главног сервера дође захтев за ажурирањем тај подсервер је сваки пут заузет и велика је вероватноћа да ће тај сервер бити проглашен не валидним због тога сто велики број података није било ажурирано. Насупрот томе рецимо да подсервер који се налази у африци прима премало захтев код њега се може десити да ниједан податак неће стићи до галвног сервера дуже време па ће због тода тај сервер исто да се прогласи не валидним. Такође претпоставимо да је сервер у јужној америци остао без напајања у томе случају може се десити било кока од ових грески ако довољно документа треба ажурирати сервер ће бити искључен ако не због тога онда ће сервер бити искључен ако се не јави главном серверу дужи временски период.