



УНИВЕРЗИТЕТ У НОВОМ САДУ ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА



УНИВЕРЗИТЕТ У НОВОМ САДУ

ФАКУЛТЕТ ТЕХНИЧКИХ НАУКА

НОВИ САД

Департман за рачунарство и аутоматику

Одсек за рачунарску технику и рачунарске комуникације

ИСПИТНИ РАД

Кандидат: Никола Влашкалин

Број индекса: RA 187/2018

Предмет: Међурачунарске комуникације и рачунарске мреже 1

Тема рада: QT Framework - FTP client

Ментор рада: Проф. др Илија Башичевић

Нови Сад, јануар, 2023.

Садржај

1. Опис проблема	3
2. File Transfer Protocol	3
2.1 Историја протокола	3
2.2 Начин функционисања протокола	3
2.2.1 Команде протокола	4
2.2.2 Предности протокола	4
2.2.3 Мане протокола	4
3. Концепт решења	4
4. Опис решења	5
4.1 void FTPClient::connectToServer()	5
4.2 void FTPClient::login(const QString &myusername, const QString &password)	5
4.3 void FTPClient::download(const QString &file, int filetype)	5
4.4 void FTPClient::upload(const QString &file, int filetype)	6
4.5 void FTPClient::listAll()	6
4.6 void FTPClient::disconnect()	6
4.7 void FTPClient::wait()	6
4.7 SDL и MSC дијаграми	7
5.0 Тестирање	12
6.0 Закључак	12
7.0 Литература	13

Списак слика

- | | |
|--|-----------|
| 1. Слика 1. - Илустрација FTP протокола | страна 3 |
| 2. Слика 2. - SDL дијаграм 1 -Представља апликацију у идле стању, процес пријављивања и одјављивање | страна 7 |
| 3. Слика 3. - SDL дијаграм 2 - Представља команде које корисник може да унесе путем конзоле и тиме управља програмом | страна 8 |
| 4. Слика 4. - SDL дијаграм 3 - Представља излиставање датотека | страна 8 |
| 5. Слика 5. - SDL дијаграм 4 -Представља функцију за преузимање датотеке са сервера | страна 9 |
| 6. Слика 6. - SDL дијаграм 5 Представља функцију за учитавање датотеке на сервер | страна 9 |
| 7. Слика 7. - MSC дијаграм 1 - Представља успешно пријављивање, излиставање и преузимање фајла | страна 10 |
| 8. Слика 8. - MSC дијаграм 2 - Представља успешно пријављивање, логовање и неуспешно учитавање датотека на сервер | страна 11 |

1. Опис проблема

Потребно је коришћењем QT развојног окружења реализовати скуп основних команди FTP клијента (File Transfer Protocol (FTP) Client).

2. File Transfer Protocol

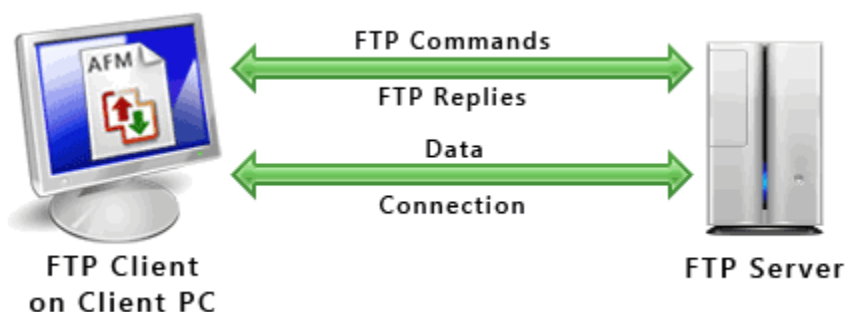
FTP (енгл. File transfer protocol – протокол за пренос датотека) је најчешће коришћен протокол за пренос података између два рачунара на мрежи. FTP користи TCP/IP за мрежну комуникацију, што омогућава да буде поуздан и сесијски оријентисан.

2.1 Историја протокола

Протокол је много пута евакуирао кроз године. Први механизми слања докумената су настали 1971. и били су намењени серверима на М.И.Т. Универзитету. Занимљивост је та да је протокол настао пре TCP/IP протокола. Кроз ових 50 година, протокол се редовно унапређивао и дан данас је популаран.

2.2 Начин функционисања протокола

FTP веза се успоставља на захтев клијентског рачунара према серверском рачунару. Клијентски рачунар мора да поседује програм који имплементира FTP протокол (тзв. FTP клијент), а серверски рачунар мора да поседује програм који прихвата конекције на стандардном FTP порту и такође разуме команде протокола FTP (тзв. FTP сервер или FTP демон). Када се веза успостави, клијентски програм шаље корисничке команде серверском програму, који их обавља и шаље одговор. Тај одговор може бити порука да је команда успешно обављена, датотека која је захтевана или порука о грешци.



Слика 1. - Илустрација FTP протокола

2.2.1 Команде протокола

Протокол садржи велики број команди, неке од којих су најчешће. Овај пројекат је основна верзија клијента и садржи само основне команде. Те комане су:

- USER - Унос имена корисника који се пријављује на систем
- PASS - Унос шифре корисника који се пријављује на систем
- LIST - Излиставање свих датотека у директоријуму
- RETR - Преузимање датотеке
- STOR - Учитавање датотеке на сервер
- QUIT - Прекидање конекције са системом
- PASV - Говори систему да слуша на дата порту
- TYPE I - Представља која ће се тип датотека слати, у овом случају је бинарна датотека (.jpg, .bmp, .mp3, .zip..)
- TYPE A - Представља који ће се тип датотека слати, у овом случају су ASCII датотеке (.txt, .c, .html, .cpp..)

2.2.2 Предности протокола

Неке од предности протокола су:

- Дозвољава рад са више датотека
- Могућност настављања преноса у случају да је конекција нестала
- Могућност стављања датотека у ред чекања
- Могућност заказивања протокола
- Нема ограничења по питању величина датотека
- Бржи је од HTTP протокола

2.2.3 Мане протокола

Неке од мана протокола су:

- Датотеке и кориснички подаци нису енкриптовани при слању
- Нема могућност праћења процеса преноса, једини начин је да се мануелно провери статус
- Не испуњава захтеве усклађености безбедне везе

ФБИ је више пута напоменуо о слабостима овог протокола и против су његовог коришћења.

3. Концепт решења

Клијент покреће симулацију, аутоматски се шаље захтев за конектовање на сервер. Након тога, од стране корисника се очекују његови подаци тј корисничко име и шифра. Након успешног пријављивања, корисник добија приступ серверу. Корисник, путем конзолне апликације, има могућност да бира које ће команде користити и како. Команде корисник уноси преко тастатуре и формиране су на следећи начин:

1. Излиставање датотека
2. Преузимање фајла
3. Учитавање датотека на сервер
4. Одјављивање

Уколико се одлучи за излиставање, корисник добија листу свих датотека у директоријуму. Такође, има опцију и за преузимање и учитавање датотеке на сервер. Уколико је завршио са апликацијом, корисник уноси број 4 и тиме се одјављује са сервера.

4. Опис решења

Следеће функције које су реализовање врше интерпретацију основних команди FTP клијента. Уз опис истих, приложени су и MSC i SDL дијаграми.

Функције које су реализоване су:

- void FTPClient::connectToServer()
- void FTPClient::login(const QString &myusername, const QString &password)
- void FTPClient::download(const QString &file, int filetype)
- void FTPClient::upload(const QString &file, int filetype)
- void FTPClient::listAll()
- void FTPClient::disconnect()
- void FTPClient::wait()

4.1 void FTPClient::connectToServer()

Клијент се путем сокета повезује на сервер. Порт који се уноси као параметар је дефинисан у протоколу и то је 21.

4.2 void FTPClient::login(const QString &myusername, const QString &password)

Сервер након повезивања клијента захтева његове податке за пријављивање. Ова функција то реализује. Клијент у конзоли уноси његове податке и ти параметри су прослеђени функцијом. Након тога, параметри се стављају иза одговарајуће команде и путем сокета се шању на сервер који их проверава и одговара клијенту позитивним или негативним одговором. Након логовања, клијенту су омогућене остале функционалности.

4.3 void FTPClient::download(const QString &file, int filetype)

Једна од тих функционалности је и функција за преузимање. Функција прима два параметра, први параметар представља локацију и име фајла док други параметар представља тип фајла.

Тип фајла може бити 0 (ако је бинарни) и 1 (ако је ASCII). Након тога, проверава се која је команда у питању и иста се шаље путем сокета. Затим се шаље команда PASV која

говори серверу да слуша порт за пренос података. Након слања исте, отвара се нови сокет за пренос података и повезује се на порт 1024. Када добијемо потврду да је отворен сокет за пренос података, путем сокета који служи за команде се шаље команда за преузимање и име датотеке. Затим се прави нови стринг за локацију и име датотеке где ће се сачувати, путем сокета за пренос података се читају подаци датотеке на серверу и уписују се у нашу локалну.

4.4 void FTPClient::upload(const QString &file, int filetype)

Начин функционисања ове функције је сличан као функција за преузимање.

Код овог случаја прослеђујемо име фајла које ће се слати. Затим се путем сокета уписује датотека у директоријум сервера.

4.5 void FTPClient::listAll()

Функција служи за излиставање свих датотека на страни сервера.

Првенствено се шаље команда PASV која му говори да слуша порт за пренос података. Након тога се шаље команда за излиставање и отвара се нови сокет за пренос података. Чита из сокета и уписује га у низ који се након тога исписује.

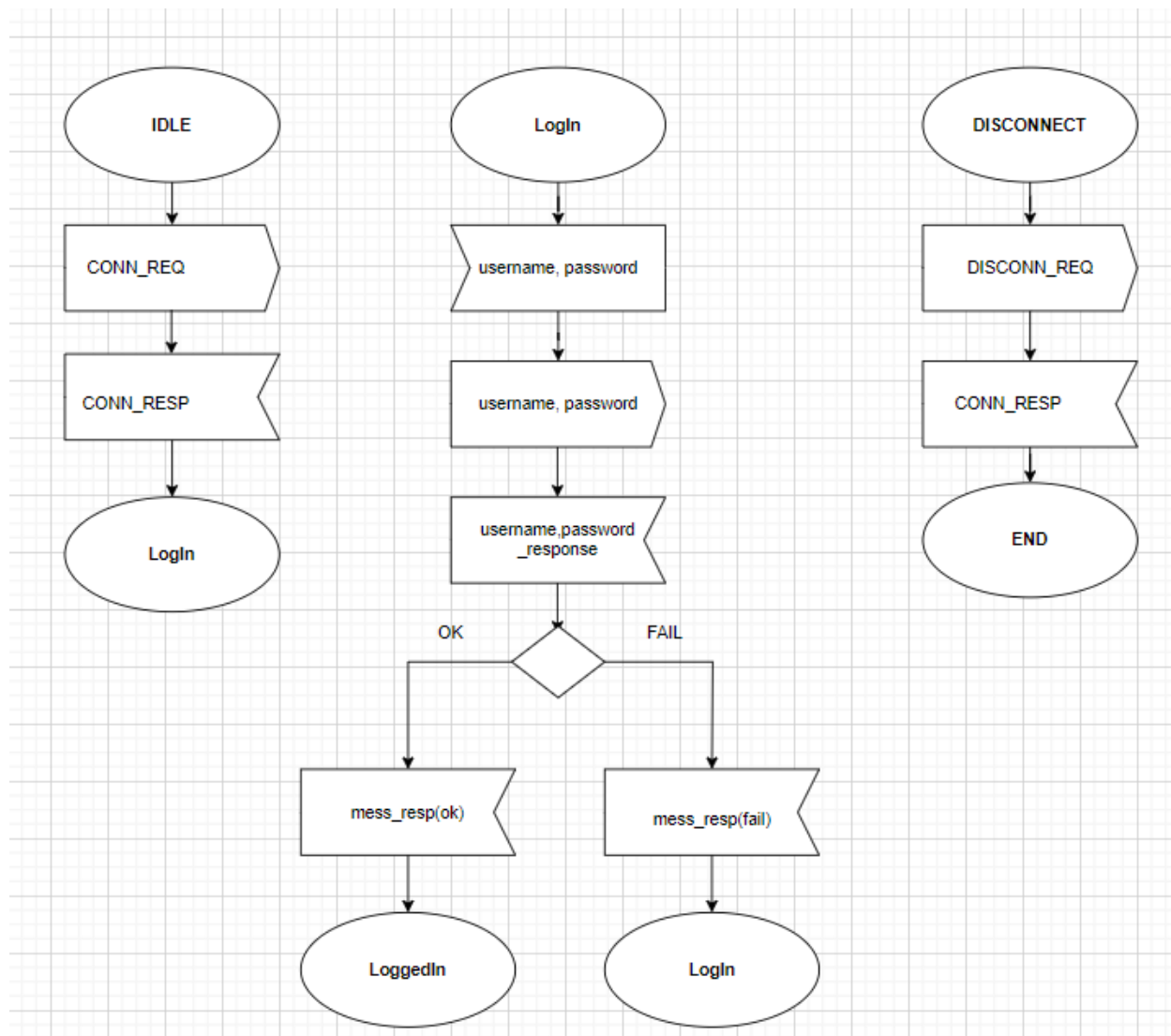
4.6 void FTPClient::disconnect()

Шаље серверу команду за одјављивање.

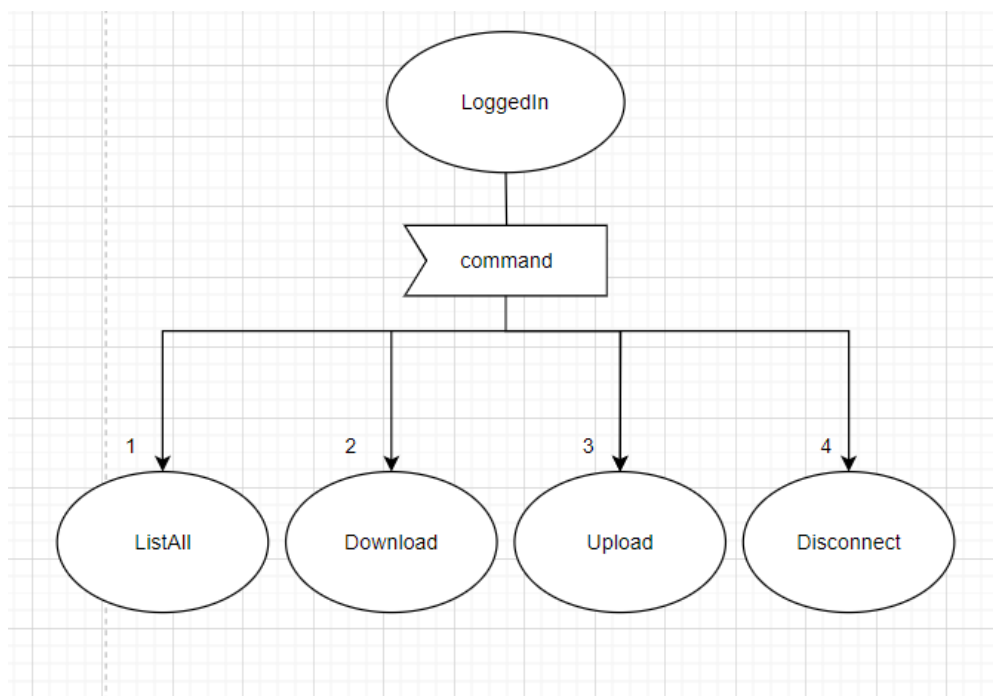
4.7 void FTPClient::wait()

Функција нема контакта са самим сервером, служи да се након сваког слања команде сачека једна секунда како би клијент добио одговор.

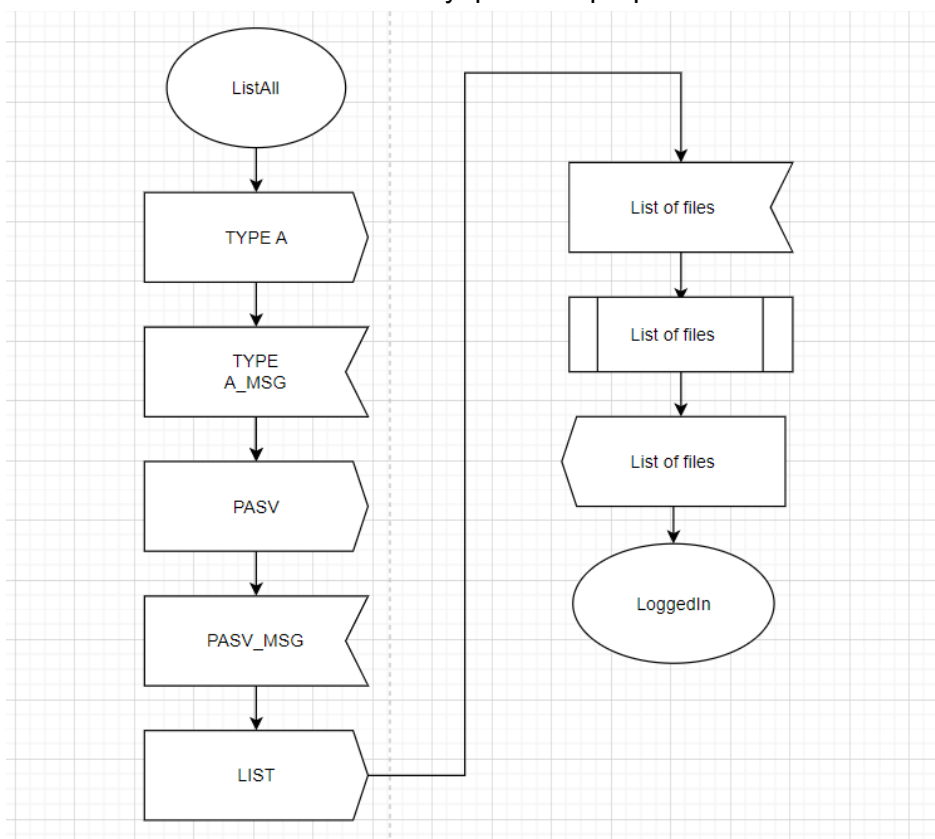
4.7 SDL и MSC дијаграми



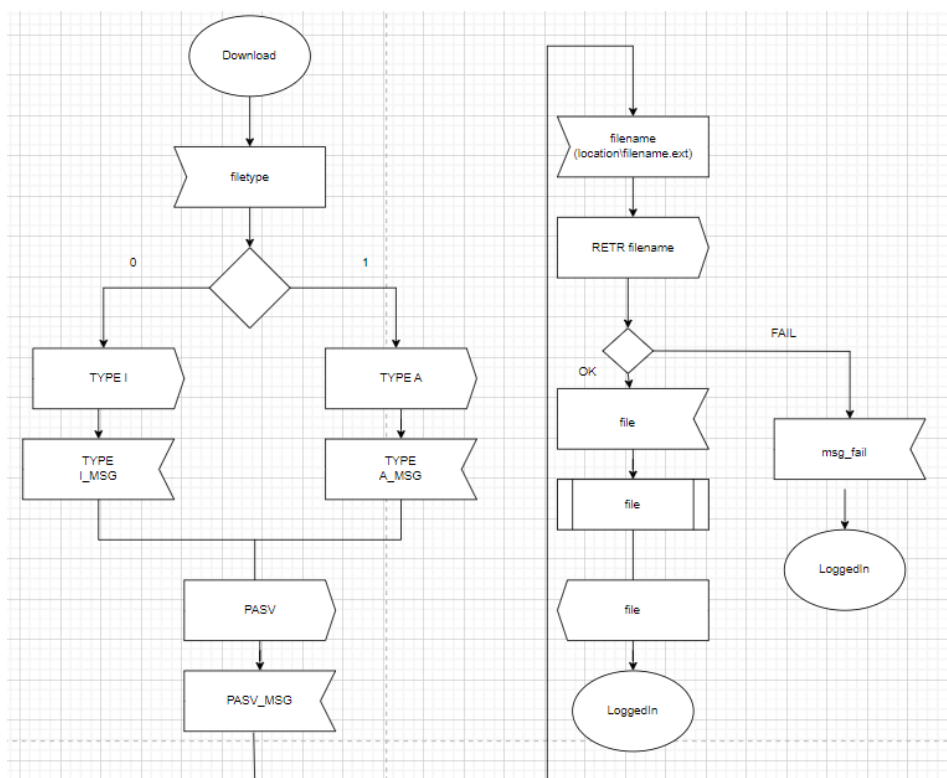
Слика 2. SDL дијаграм 1 који представља апликацију у идле стању, процес пријављивања и одјављивање



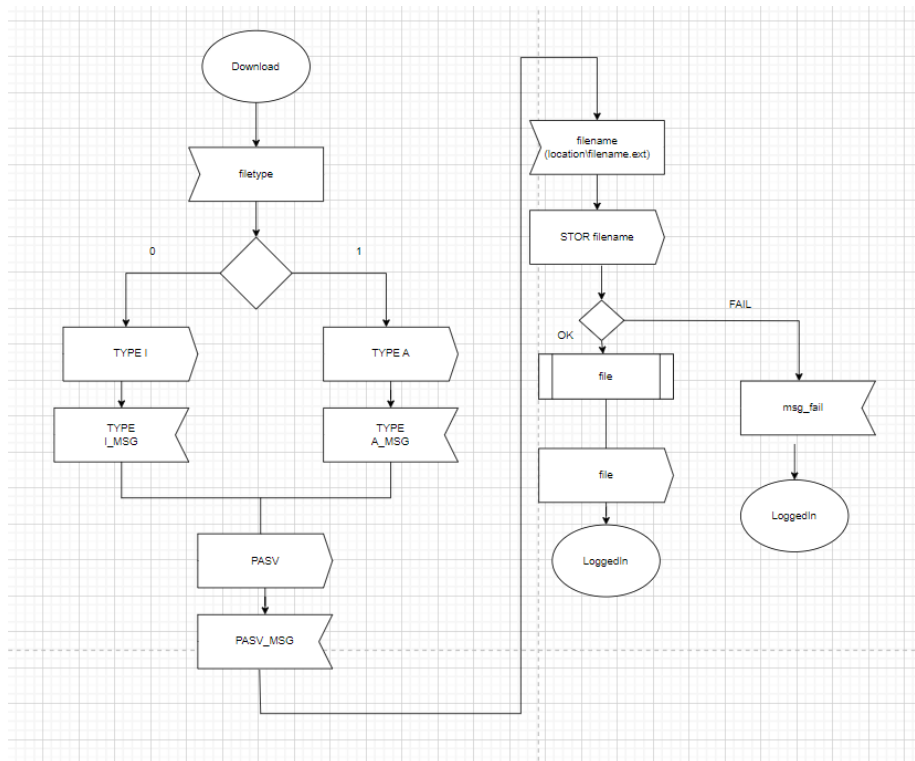
Слика 3. SDL дијаграм 2 - Представља команде које корисник може да унесе путем конзоле и тиме управља програмом



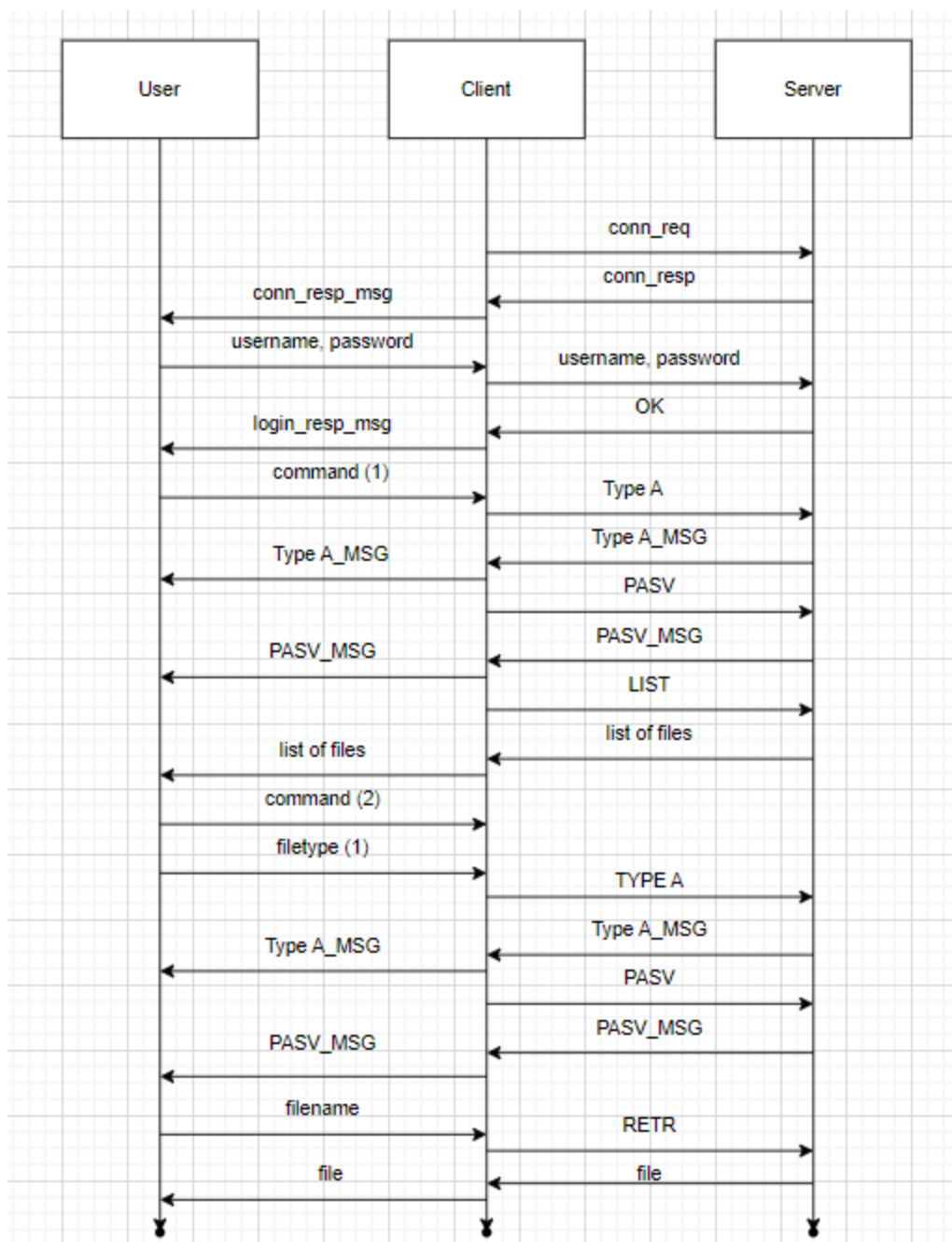
Слика 4. SDL дијаграм 3 - Представља излиставање датотека



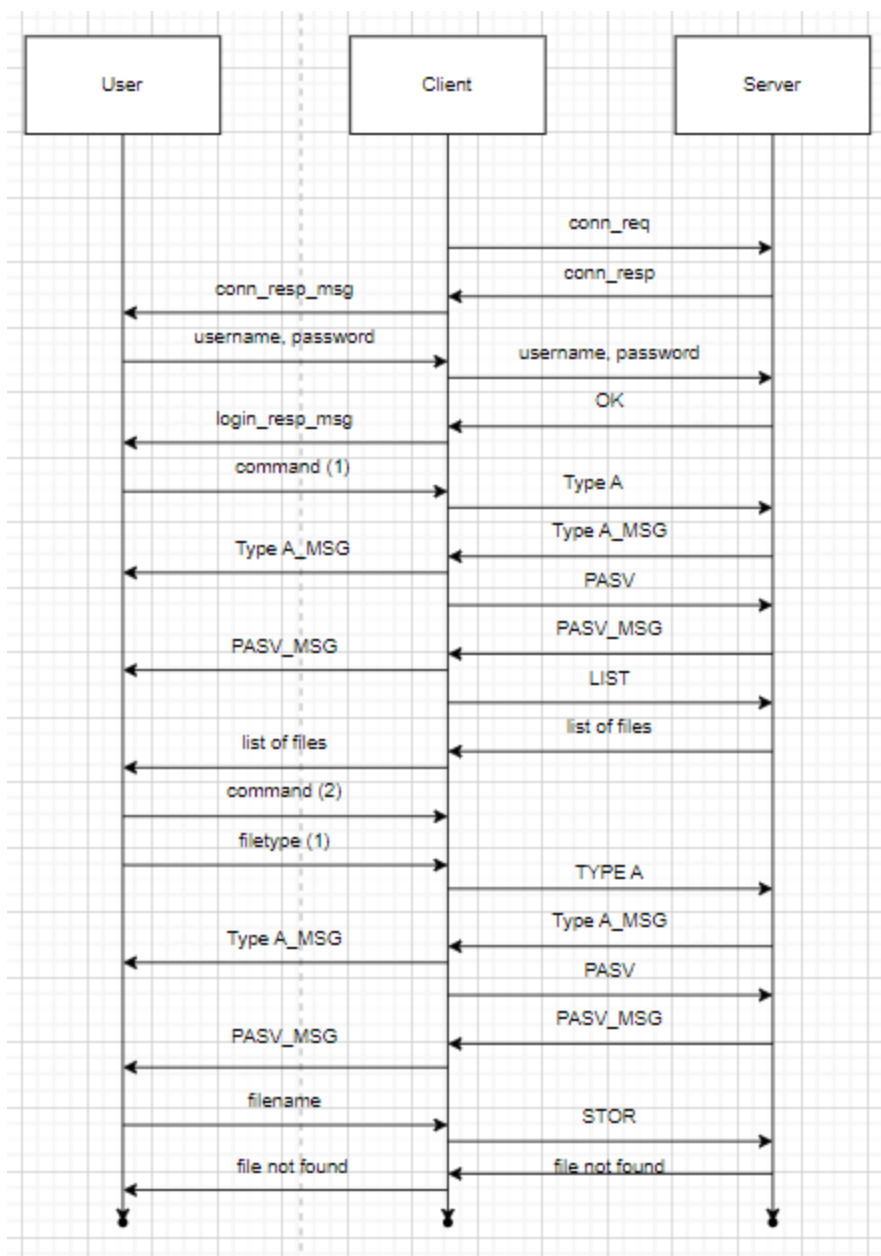
Слика 5. SDL дијаграм 4 - Представља функцију за преузимање датотеке са сервера



Слика 6. SDL дијаграм 5 - Представља функцију за учитавање датотеке на сервер



Слика 7. MSC дијаграм 1 - Слика представља успешно пријављивање, излиставање и преузимање фајла



Слика 8. MSC дијаграм 2 - Представља успешно пријављивање, логовање и неуспешно учитавање датотека на сервер

5.0 Тестирање

Тестирање је обављено са два типа датотека: .jpg и .txt. Свака датотека је успешно била преузета и учитана на сервер. Излиставање такође функционише. Покушани су случајеви где датотека није постојећа, што даје очекивани излаз. Такође, покушани су погрешни кориснички подаци где је излаз такође очекиван.

6.0 Закључак

Овај пројекат ми је био јако занимљив. Приближио ми је окружење QT Creator и увиђам његову једноставност и одличну прилику за примену у мрежама. Сама апликација је доста проста и примитивна. Увиђам свакако начин за проширење као што су додатне команде и можда покриће неких погрешних случајева. Такође, ова апликација би била идеална прилика да се у њој користе widget-i. На ово бих се свакако вратио у некој ближој будућности и покушао да је унапредим.

7.0 Литература

[1] STD 9 RF 959, File Transfer Protocol, J. Postel, J. Reynolds, 1985.

[2] Computer Networking A Top-Down Approach 6th Edition, James F. Kurose, Keith W. Ross, Pearson, 2013.