НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КПІ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

**КУРСОВА РОБОТА**

***з дисципліни "Основи програмування"***

Виконав: Брусенцов Юрій Олексійович

Група: КП-61

Допущено до захисту

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

2 семестр 2016/2017

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КПІ ім. Ігоря Сікорського»

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра програмного забезпечення комп’ютерних систем

|  |  |
| --- | --- |
| Узгоджено    Керівник роботи    \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ | ЗАХИЩЕНА "\_\_"\_\_\_\_\_\_\_\_\_2017р.  з оцінкою\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_/Гадиняк Р.А./ |

**Програма-чат**

Виконавець роботи

Брусенцов Юрій Олексійович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017р.

**Вступний опис системи**

Програма-чат — клієнт-серверне програмне забезпечення з можливістю підключення і спілкування між собою двох або більше людей в режимі реального часу.

Чат - це відмінний спосіб спілкуватися зі своїми друзями. Створюйте свій сервер, або заходьте на вже створений та запрошуйте друзів або спілкуйтеся з незнайомцями.

Для спілкування вам всього лише потрібно зареєструватися у чат-кімнаті та відправити своє перше повідомлення.

Розроблений чат дозволяє підключатися до різних серверів та спілкуватися із користувачами в онлайн режимі.

При вході до чату перед вами з’явиться вікно реєстраціі та логіну. Коли ж ви зареєструєтесь або увійдете у вже створений аккаунт, ви потрапите на головне вікно, де й відбувається основне листування. Ви можете налаштувати свій власний профіль та переглядати профілі інших користувачів. Також ви маєте можливість додавати іншого користувача чату до ваших контактів. Повідомлення у чаті можна відправляти як всім користувачам, так і одному, певному юзеру, котрий знаходиться у вас в друзях, таке листування будете бачити лише ви вдвох.

Чат дозволяє змінити ip-адресу сервера кімнати-чату як із вікна логіну-реєстрації, так і з головного вікна листування. При вході до нового сервера вам потрібно заново зареєструватися або залогінитися , та налаштувати інформацію свого профілю, тощо.

Також ви можете запустити сервер на своєму комп’ютері та запросити друзів до свого чату через власну ip-адресу.

**Інструменти розробки**

Чат розроблений на мові програмування С++. Під час розробки використовувався фреймворк Qt, який надав можливість для створення віконних програм та реалізації клієнт-серверної архітектури.

Найголовніші класи фреймворку Qt, які були використані у розробці:

* QTcpSocket - дозволяє легко створити та використовувати ТСР сокет.
* QTcpServer - дозволяє створити сервер ТСР протоколу.
* QMainWindow, QDialog та інші - дозволяють створити та відмалювати графічний інтерфейс програми.

**Основний функціонал системи**

При запуску програми відкривається діалогове вікно для входу у чат. Вікно має поле для логіну та паролю. Дані поля дозволяють вводити лише певні символи та усі цифри та літери латинського алфавіту. При неправильному наборі даних логін-пароль вікно повідомляє про помилку. Якщо ви не зареєстровані, то ви можете поставити прапорець у полі “у мене ще немає аккаунту” та зареєструватися.

Під час написання роботи стало зрозуміло, що зробити реалізацію чату із кімнатами буде неможливо, тому було прийнято рішення зробити одне, головне вікно чату із можливістю відправляти приватні повідомлення іншим юзерам.

На головному вікні чату відображається історія листування, та повідомлення про підключення або відключення користувачів до чату, або певні повідомлення (про збереження налаштувань або неможливість відправлення приватного повідомлення).

Реалізовано бокове меню із кнопками:

* Змінити ip-адресу - дозволяє змінити адресу сервера не виходячи з програми.
* Контакти - виводить список всіх юзерів, які ви додали до контактів.
* Налаштування - дозволяє змінювати інформацію свого профілю та змінювати пароль.

При подвійному натисканні на ім’я користувача у списку юзерів онлайн, або у списку контактів (відкритий через кнопку у боковому меню) відкривається профіль користувача.

Профіль корисувача розподілений на три зони:

* Публічна інформація - відображається ім’я, вік, країна та поле із довільною інформацією (доступна будь-якому користувачу чату).
* Приватна інформація - містить телефон та електронну адресу користувача (доступна тільки для юзерів, які знаходяться у вашому списку контактів).
* Блок із двох кнопок: додати у контакти, та відправити приватне повідомлення.

Реалізовано відправлення приватних повідомлень одному юзеру: при натисканні на кнопку “відправити приватне повідомлення” у профілі вашого контакту відкривається вікно із полем для вводу повідомлення. Після натискання на кнопку відправлення, вікно автоматично закривається, і повідомлення відправляється до вибраного юзера.

При отриманні публічного повідомлення (повідомлення до всіх користувачів чату, що знаходяться онлайн) у лог головного вікна записується час, ім’я відправника та текст повідомлення.

При отриманні приватного повідомлення у лог головного вікна записується час, ім’я того хто відправив повідомлення та юзернейм того, кому воно було адресовано.

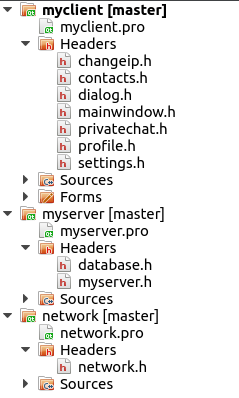
При підключенні або відключенні користувача чату, всі юзери сповіщуються про це відповідним повідомленням.

При зміні паролю або налаштувань вашого профілю у лог чату виводиться відповідне повідомлення.

Якщо сервер вимикається, то у лог чату записується відповідна помилка.

**Опис модулів програми**

Чат складається з серверу, клієнту та статичної бібліотеки, яка підключається і до клієнта і до сервера



**Модулі клієнта:**

**MainWindow**

Головне вікно чату, у якому можна відправити повідомлення та відображається історія листування.

|  |
| --- |
| #ifndef MAINWINDOW\_H #define MAINWINDOW\_H  #include <QTcpSocket> #include <QMainWindow> #include <QTextEdit> #include <QLineEdit> #include "network.h" #include "settings.h" #include "privatechat.h" #include <QListWidgetItem>  **namespace** Ui { **class** MainWindow; }  **class** MainWindow : **public** QMainWindow {  Q\_OBJECT  **public**:  MainWindow(**const** QString strHost, **int** nPort, QWidget \*parent = 0);  **void** setUser(User temp);  **void** sendCommandToServer(**int** command);  QString getUsername() { **return** user.username; }  ~MainWindow();  **public slots**:  **void** slotReadyRead();  **private slots**:   **void** slotError(QAbstractSocket::SocketError);  **void** slotConnected();   **void** on\_sendButton\_clicked();  **void** on\_messageLineEdit\_returnPressed();   **void** on\_settingsButton\_clicked();  **void** on\_settingsButton\_pressed();   **void** on\_usersOnlineWidget\_itemDoubleClicked(QListWidgetItem \*item);  **void** on\_usersOnlineWidget\_itemPressed(QListWidgetItem \*item);   **void** on\_contactsButton\_pressed();  **void** on\_contactsButton\_released();   **void** on\_messageLineEdit\_textChanged(**const** QString &text);   **void** on\_changeIpButton\_clicked();   **bool** isRequestedByMe(QString username);  **bool** isRequestedMe(QString username);  **private**:  Ui::MainWindow \*ui;  QTcpSocket\* thisSocket;  QTextEdit\* messageBox;  QLineEdit\* inputLine;  quint16 nextBlockSize;  QList<QString> usernamesList;  QList<QString> contactsList;  QList<QString> requestedContactsList;  QList<QString> requestedContactsByMeList;   User user;  UserInfo anotherUserInfo;  UserInfo userInfo;  QString deletedContact;  QString contact;  QString privateChatUser;   **int** serverPort;  QString serverIp; };  #endif // MAINWINDOW\_H |

**Dialog**

Вікно для логіну або реєстрації, дане вікно викликається при запуску програми та дає доступ до головного вікна лише при авторизації або реєстрації.

|  |
| --- |
| #ifndef DIALOG\_H #define DIALOG\_H  #include <QDialog> #include "network.h" #include "mainwindow.h" #include "QLabel"  **namespace** Ui { **class** Dialog; }  **class** Dialog : **public** QDialog {  Q\_OBJECT  **public**:  **explicit** Dialog(**const** QString& strHost, **int** nPort, QWidget \*parent = 0);  **void** openDialog();  **void** closeDialog();  **bool** finished();  **bool** isLogined();  **void** sendToServer(**int** command);  **void** setUser(User temp);  User getUser();  **bool** checkUserUnique(QList<QString> users, QString user);  ~Dialog();  **private slots**:  **void** on\_loginButton\_clicked();  **void** slotConnected();  **void** slotError();  **void** readInfoFromServer();  **void** on\_checkBox\_clicked();   **void** on\_Dialog\_rejected();   **void** on\_loginButton\_pressed();  **private**:  Ui::Dialog \* ui;  **bool** \_finished;  **bool** isUserUnique;  QTcpSocket \* dialogSocket;  quint16 nextBlockSize;  **bool** loginStatus;  User user;  QLabel \* errorLabel;  QList<QString> usersOnline; };  #endif // DIALOG\_H |

**Contacts**

Вікно, яке відображає список усіх контактів, вашого користувача.

|  |
| --- |
| #ifndef CONTACTS\_H #define CONTACTS\_H  #include <QDialog> #include <QListWidgetItem> #include <QTcpSocket>  **namespace** Ui { **class** Contacts; }  **class** Contacts : **public** QDialog {  Q\_OBJECT  **public**:  **explicit** Contacts(QTcpSocket \* \_thisSocket, QWidget \*parent = 0);  **void** setContacts(QList<QString> contactsList);  QListWidgetItem \* getSelectedContact();  **bool** isAccepted();  ~Contacts();  **private slots**:  **void** on\_friendsListWidget\_itemDoubleClicked(QListWidgetItem \*item);   **void** on\_friendsListWidget\_itemPressed(QListWidgetItem \*item);   **void** on\_Contacts\_rejected();  **private**:  Ui::Contacts \*ui;  QList<QString> contactsList;  QListWidgetItem \* selectedContact;  QTcpSocket \* thisSocket;  **bool** accepted; };  #endif // CONTACTS\_H |

**Settings**

Вікно налаштувань. Користувач має можливість змінити дані свого профілю, та змінити пароль.

|  |
| --- |
| #ifndef SETTINGS\_H #define SETTINGS\_H  #include <QDialog> #include <QTextEdit> #include <QSpinBox> #include <QTcpSocket> #include "network.h"  **namespace** Ui { **class** Settings; }  **class** Settings : **public** QDialog {  Q\_OBJECT  **public**:  **explicit** Settings(QTcpSocket \* socket, QString \_username, QString \_password, QWidget \*parent = 0);  **void** setInfo(UserInfo \_userInfo);  ~Settings();  **private slots**:   **void** on\_buttonBox\_accepted();   **void** on\_Settings\_rejected();   **void** on\_changePassButton\_clicked();   **void** on\_changePassLine\_textChanged();  **public slots**:  **void** sendToServer(**int** command);  **private**:  Ui::Settings \*ui;  QTcpSocket \* thisSocket;  QTextEdit \* infoTextEdit;  QLineEdit \* nameLineEdit;  QLineEdit \* countryLineEdit;  QLineEdit \* phoneLineEdit;  QLineEdit \* emailLineEdit;  QSpinBox \* ageSpinBox;  QString username;  QString password;  QString oldPassword;  QString newPassword;  QString logMessage;  UserInfo userInfo;  quint16 nextBlockSize; };  #endif // SETTINGS\_H |

**Profile**

Вікно профілю користувача. У даному вікні відображається публічна і приватна інформація (тільки для контактів) користувача чату. Також у вікні розміщенні кнопки для відправлення, прийняття або видаленя з контактів (залежно від стану).

|  |
| --- |
| #ifndef PROFILE\_H #define PROFILE\_H  #include <QDialog> #include <QTcpSocket> #include "network.h"  **namespace** Ui { **class** Profile; }  **class** Profile : **public** QDialog {  Q\_OBJECT  **public**:  **explicit** Profile(QString \_myUsername,  QString \_username,  QString name,  **int** age,  QString country,  QString info,  QString phone,  QString email,  QTcpSocket \* socket,  QList<QString> contactsList,  QString command,  QWidget \*parent = 0  );  **bool** isNotDeletedFromContacts();  **bool** isAccepted();  **void** sendCommandToServer(**int** command);  ~Profile();  **private slots**:  **void** on\_addToFriendsButton\_clicked();   **void** on\_privateChatButton\_clicked();   **void** on\_pushButton\_clicked();  **private**:  Ui::Profile \*ui;  UserInfo userInfo;  QString username;  QString myUsername;  QTcpSocket \* thisSocket;  **bool** removeFromContacts;  **bool** accepted;  **bool** isContact; };  #endif // PROFILE\_H |

**PrivateChat**

Вікно за допомогою якого виконується відправлення приватного повідомлення обраному користувачу.

|  |
| --- |
| #ifndef PRIVATECHAT\_H #define PRIVATECHAT\_H  #include <QDialog> #include <QTcpSocket>  **namespace** Ui { **class** PrivateChat; }  **class** PrivateChat : **public** QDialog {  Q\_OBJECT  **public**:  **explicit** PrivateChat(QString username, QString anotherUser, QTcpSocket \* socket, QWidget \*parent = 0);  **void** sendCommandToServer(**int** command);  ~PrivateChat();  **private slots**:  **void** on\_sendButton\_clicked();  **private**:  Ui::PrivateChat \*ui;  QTcpSocket \* thisSocket;  QString username;  QString anotherUsername;  quint16 nextBlockSize; };  #endif // PRIVATECHAT\_H |

Модулі сервера:

**MyServer**

Сервер, який обробляє запити клієнтів.

|  |
| --- |
| #ifndef MYSERVER\_H #define MYSERVER\_H  #include <QWidget> #include <QList> #include "network.h" #include "database.h"  **class** QTcpServer; **class** QTextEdit; **class** QTcpSocket;  **class** MyServer : **public** QWidget { Q\_OBJECT **private**:  QTcpServer \* thisServer;  QTextEdit \* textBox;  quint16 nextBlockSize;  QList<QTcpSocket\*> clientsList;  QList<QString> usernamesList;  QList<QString> contactsList;  QList<QString> requestedContactsList;  QList<QString> requestedContactsByMeList;  QString contact;  database db;  QString thisUser;  QString anotherUser;  QTcpSocket \* offlineSocket;  QString privateUser;   **void** sendToClient(QTcpSocket\* pSocket, **const** QString& str, **int** command);  **void** sendUserInfoToClient(QTcpSocket\* pSocket,  **int** command,  QString name,  **int** age,  QString country,  QString info,  QString phone,  QString email  );  **public**:  MyServer(**int** nPort, QWidget\* pwgt = 0);  **void** broadcast(**int** command, QString str);  QTcpSocket \* findSocketByUsername(QString username);  **bool** checkIsUserOnline(QString username);  QList<QString> getContactsList(QString username);  **public slots**:  **virtual** **void** slotNewConnection();  **void** slotReadClient();  **void** slotClientDisconnected(); }; #endif //MYSERVER\_H |

**Database**

Модуль який працює із базою даних, а саме, створює її з нуля, якщо її немає на сервері, записує та змінює нових та старих юзерів.

|  |
| --- |
| #ifndef DATABASE\_H #define DATABASE\_H  #include <string> #include <QString> #include "network.h"  **class** database { **public**:  **bool** driverAvailable();  **bool** newDatabase();   **bool** createUserTable();  **bool** createInfoTable();  **bool** createContactsTable();   **bool** setUsernameAndPassword(QString username, QString password);  **bool** checkUsernameUnique(QString username, QString password);  **bool** singIn(QString username, QString password);   **bool** addContact(QString username, QString contact);  QList<QString> getContacts(QString username);  QList<QString> getRequests(QString contact);  **bool** isContacts(QString username, QString contact);  **bool** deleteContact(QString username, QString contact);   **bool** addUserInfo(QString username, QString name, **int** age, QString country, QString info, QString phone, QString email);  **bool** getUserInfo(QString username, UserInfo \* usrInf);   **bool** changePassword(QString username, QString newPassword); };  #endif // DATABASE\_H |

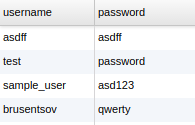
Модуль статичної бібліотеки:  
  
**Network.h**

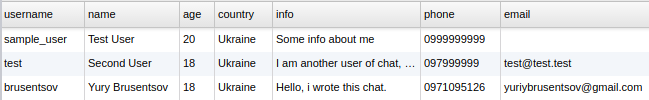
|  |
| --- |
| #ifndef NETWORK\_H #define NETWORK\_H  #include <QString> #include <QtCore/QCoreApplication> #include <QDataStream> #include <QByteArray> #include <QFile> #include <QString> #include <QHash> #include <QDebug>  **enum** A : **int** {  sendMessage,  setName,  userConnected,  setUsernameToWidget,  checkInDatabase,  sendUserToServer,  sendUserToServerSingin,  database\_logined,  database\_wrongUsername,  database\_wrongPassword,  sendUserInfo,  getMyInfoFromDatabase,  getUserInfoFromDatabase,  addContactToDatabase,  getContactsFromDatabase,  deleteContactFromDatabase,  getOnlineUsers,  sendPrivateMessage,  receivePrivateMessage,  openPrivateChat,  privateChatUserIsOffline,  changePassword,  setLogMessage,  sendContactRequest  //another comands };  **struct** User {  QString username;  QString password; };  **struct** UserInfo {  QString name;  **int** age;  QString country;  QString info;  QString phone;  QString email; };  **void** fillStreamClient(**int** command, QDataStream \* out, User user); QString readIpFromFile(); **void** writeIpToFile(QString ipText);  #endif // NETWORK\_H |

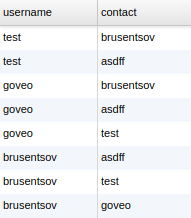
**Організація даних**

Для збереження даних про клієнтів було обрано використовувати базу даних типу sqlite3.

Таблиця паролів :



Таблиця інформації про користувачів:  


Таблиця контактів:  


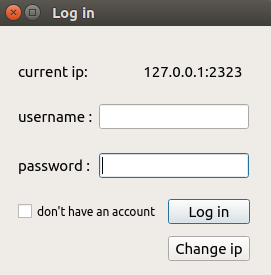
У таблиці контактів користувачі які знаходяться у колонці username відправили запит на додавання у друзі тому користувачу, який записаній у колонці contact. Якщо ж користувачі відправили запити один одному, вони додаються до контактів.

**Опис найважливіших алгоритмів**

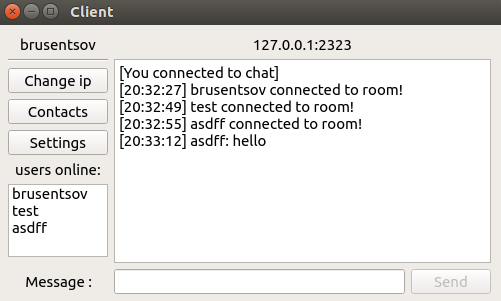
Функція для відправки повідомлень клієнту:

|  |
| --- |
| **void** MyServer::sendToClient(QTcpSocket\* pSocket, **const** QString& str, **int** command) {  QByteArray arrBlock;  QDataStream out(&arrBlock, QIODevice::WriteOnly);  out.setVersion(QDataStream::Qt\_4\_2);   out << quint16(0) << command;   **if** (command == setUsernameToWidget) {  out << usernamesList;  } **else** **if** (command == getContactsFromDatabase) {  out << contactsList << requestedContactsByMeList << requestedContactsList;  } **else** **if** (command == getOnlineUsers) {  out << str << usernamesList;  } **else** **if** (command == receivePrivateMessage) {  out << str << privateUser;  } **else** **if** (command == openPrivateChat) {  out << str << privateUser;  } **else** **if** (command == privateChatUserIsOffline) {  out << anotherUser;  } **else** **if** (command == changePassword) {  out << str;  } **else** {  out << str;  }   out.device()->seek(0);  out << quint16(arrBlock.size() - **sizeof**(quint16));  pSocket->write(arrBlock); } |

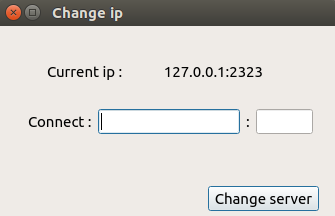
**Результати роботи**



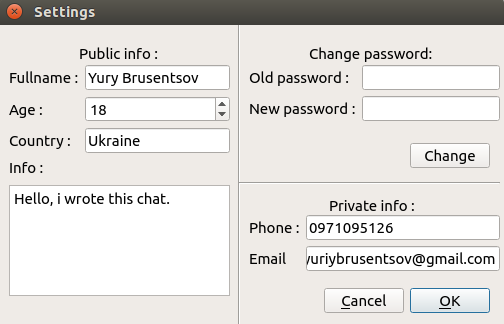
(рис. 1 - вікно реєстрації)



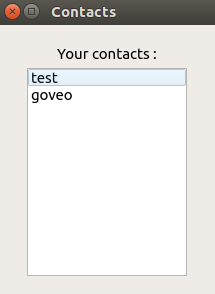
(рис.2 - головне вікно чату)



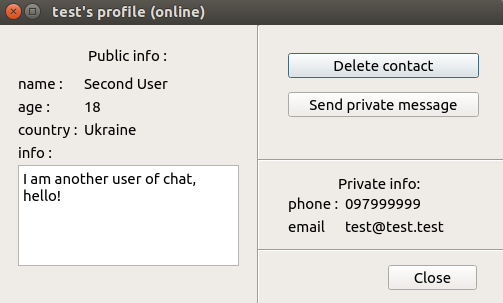
(рис.3 - вікно зміни ip-адреси)



(рис.4 - вікно налаштувань)



(рис.5 - вікно контактів)



(рис.6 - вікно профілю іншого користувача)

**Мануал для користувача**

При першому вході до серверу вам необхідно зареєструватися, це можна зробити натиснувши прапорець біля напису “Ще не маю аккаунту”, якщо ж ви вже маєте профіль, то для входу в кімнату потрібно правильно ввести логін та пароль. Коли увійдете у свій аккаунт, ви потрапите на головне вікно чату.

Щоб змінити сервер за ip-адресою вам потрібно натиснути на кнопку “Змінити ip” та у відкритому вікні ввести потрібні дані.

Щоб налаштувати свій профіль або змінити пароль - натисніть на кнопку “Налаштування” та у відповідних полях замініть дані на бажані.

Щоб переглянути профіль іншого користувача натисніть на його ім’я подвійним кліком.

Щоб додати іншого користувача до контактів потрібно відкрити його профіль та натиснути на кнопку “додати до контактів”, лише після того як користувач прийме ваш запит він додасться до вашого списку контактів.

Щоб переглянути список усіх ваших контактів потрібно натиснути на кнопку “Контакти”.

Також ви можете запустити сервер на своєму комп’ютері та запросити друзів до свого чату через власну ip-адресу.