

Student Number: 215222721

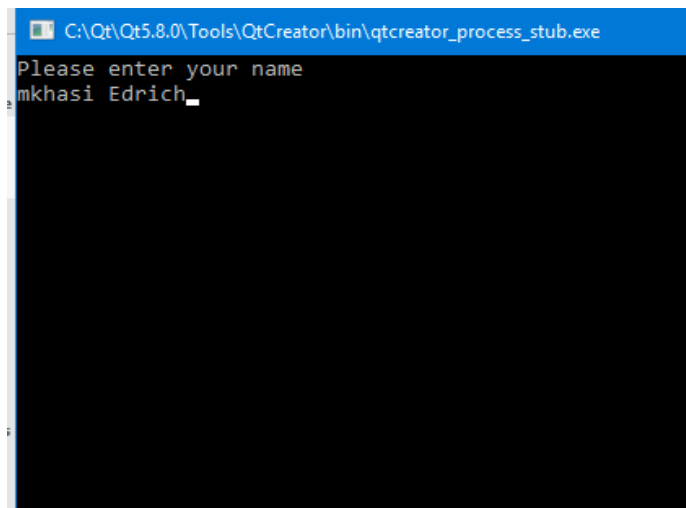
# Portfolio 1

## Description of what it does

Game 1 is a game that prompt a user or a player to enter his/her name at first. The game then prompts the user to enter ten names. At the end, the game displays the name of the player or user and the first five names entered and the last four names entered.

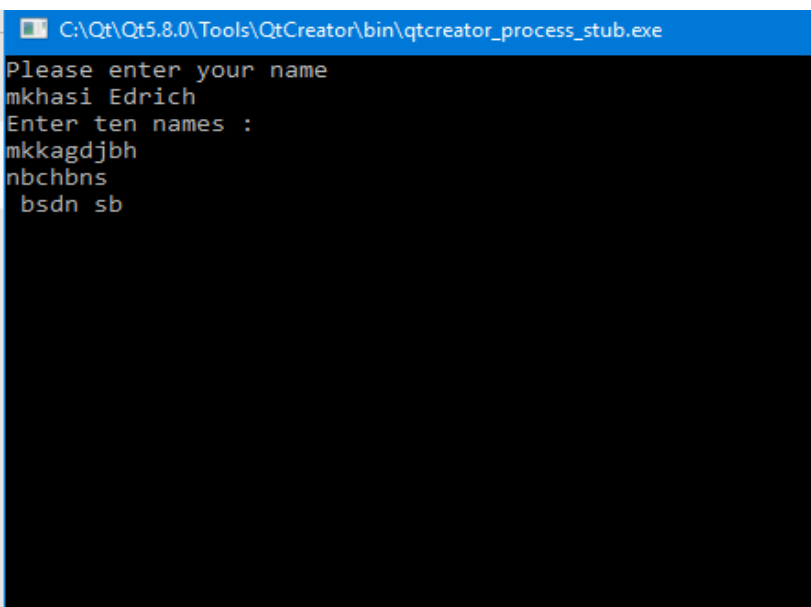
## Screenshot(s)

A user is prompted to for a name.

A screenshot of a Qt Creator console window. The title bar shows the path 'C:\Qt\Qt5.8.0\Tools\QtCreator\bin\qtcreator\_process\_stub.exe'. The console text reads: 'Please enter your name' followed by the user input 'mkhasi Edrich' with a cursor at the end.

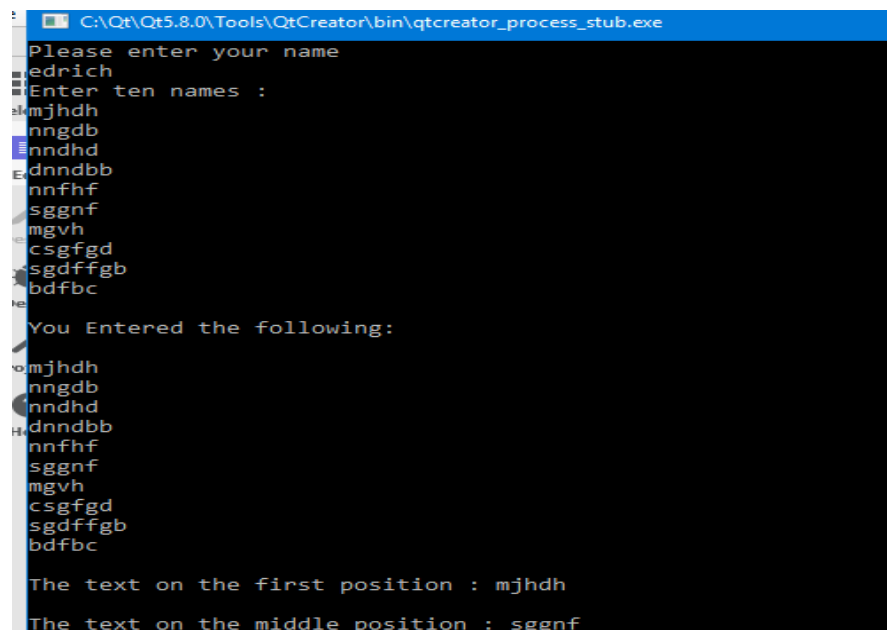
```
C:\Qt\Qt5.8.0\Tools\QtCreator\bin\qtcreator_process_stub.exe
Please enter your name
mkhasi Edrich_
```

A user enters a name and some few names.

A screenshot of a Qt Creator console window. The title bar shows the path 'C:\Qt\Qt5.8.0\Tools\QtCreator\bin\qtcreator\_process\_stub.exe'. The console text reads: 'Please enter your name' followed by 'mkhasi Edrich', then 'Enter ten names :' followed by four lines of input: 'mkkagdjbh', 'nbchbns', and 'bsdn sb'.

```
C:\Qt\Qt5.8.0\Tools\QtCreator\bin\qtcreator_process_stub.exe
Please enter your name
mkhasi Edrich
Enter ten names :
mkkagdjbh
nbchbns
bsdn sb
```

The application the displays the output.



```
C:\Qt\Qt5.8.0\Tools\QtCreator\bin\qtcreator_process_stub.exe
Please enter your name
edrich
Enter ten names :
mjhdh
nngdb
nndhd
dnndbb
nnfhf
sggnf
mgvh
csgfgd
sgdffgb
bdfbc
You Entered the following:
mjhdh
nngdb
nndhd
dnndbb
nnfhf
sggnf
mgvh
csgfgd
sgdffgb
bdfbc
The text on the first position : mjhdh
The text on the middle position : sggnf
```

## Source Code

```
//215222721
#include <QCoreApplication>
#include <QTextStream>
#include <QString>
#include <QList>

int main(int argc, char *argv[])
{
    QCoreApplication a(argc, argv);

    QTextStream out(stdout);
    QTextStream in(stdin);

    out << "Please enter your name\n";
    out.flush();

    QString text = in.readLine();

    QList<QString> list;

    QString name;

    //Remove this ! And get 10 items from input (loop)
    out<<"Enter ten names :"<<endl;

    for(int x=0;x<10;x++)
    {
        name=in.readLine();
        list.append(name);
    }

    out<<endl;
```

```

out<<"You Entered the following: "<<endl;
out<<endl;

for(int x=0;x<10;x++)
{
    out<<list.at(x)<<endl;
    out.flush();
}

//The name entered first
out<<endl;
out<<"The text on the first position : ";
out<<list.at(0);

out<<endl<<endl;
out<<"The text on the middle position : ";
out<<list.at(5);

//Display only the first 5 items in the list
out<<endl;
out<<"The first 5 items on your list..."<<endl;
for (int i = 0; i < 5; ++i)
{
    out << list.at(i) << endl;
    out.flush();
}

//Display the last 4 in the list reading from the back
out<<endl;
out<<"The last 4 items on your list..."<<endl;
for (int i = (list.size()-1); i >= 6; i=i-1)
{
    out << list.at(i) << endl;
    out.flush();
}

out << "Game over, You are smarter "+text+"\n";

out.flush();

return 0;

return a.exec();
}

```

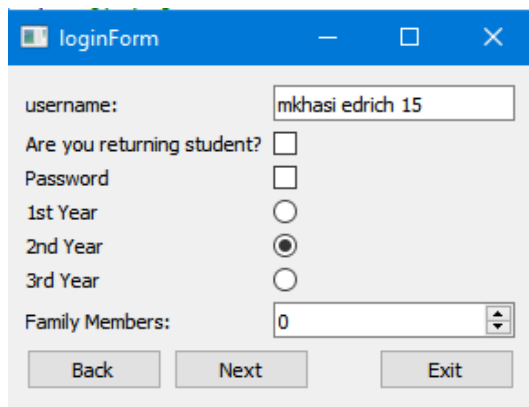
# Portfolio 2

## Description of what it does

A basic login interface form created using a back-end code. The application displays a form that include the QLabel class to create or display label, it also uses the QLineEdit class to allow a user to type any numbers or words as strings. The application also includes the Radio buttons and SpinBox as well as push buttons.

## Screenshots

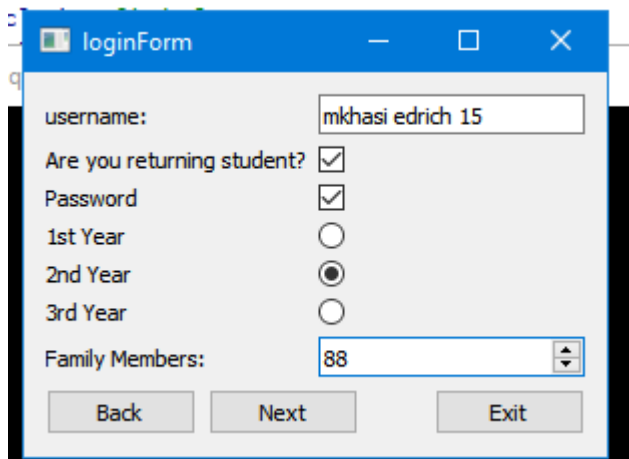
Only one Radio button is checked and the Line edit is filled with strings and integer values.



A screenshot of a Qt application window titled "loginForm". The window contains the following elements:

- A "username:" label followed by a QLineEdit containing the text "mkhasi edrich 15".
- A label "Are you returning student?" followed by an unchecked checkbox.
- A "Password" label followed by an unchecked checkbox.
- A group of three radio buttons labeled "1st Year", "2nd Year", and "3rd Year". The "2nd Year" radio button is selected.
- A "Family Members:" label followed by a SpinBox showing the value "0".
- Three push buttons at the bottom: "Back", "Next", and "Exit".

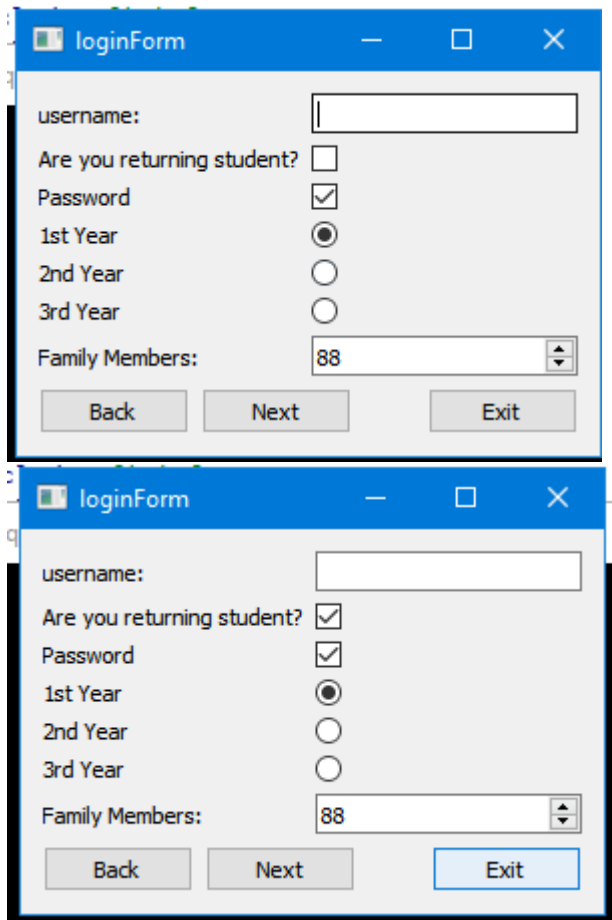
A radio button, checkbox and a line edit is filed with the values.



A screenshot of the same "loginForm" window, but with the following changes:

- The "Are you returning student?" checkbox is now checked.
- The "Password" checkbox is now checked.
- The "Family Members:" SpinBox now shows the value "88".
- The "2nd Year" radio button remains selected.
- The "username:" field still contains "mkhasi edrich 15".
- The "Back", "Next", and "Exit" buttons are still present at the bottom.

RadioButton, CheckBox is checked but the line edit is empty.



## Source Code

```
//215222721
#include <QApplication>
#include <QHBoxLayout>
#include <QLabel>
#include <QLineEdit>
#include <QPushButton>
#include <QGridLayout>
#include <QVBoxLayout>
#include <QRadioButton>
#include <QCheckBox>
#include <QSpinBox>
#include <QSlider>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;

    QVBoxLayout *mainLayout = new QVBoxLayout(&w);
    QGridLayout *editLayout = new QGridLayout;
    QHBoxLayout *buttonLayout = new QHBoxLayout;
    QHBoxLayout *radioLayout = new QHBoxLayout;
    QHBoxLayout *checkboxLayout = new QHBoxLayout;
    //QSlider * slider = new QSlider(Qt::Horizontal);
```

```

mainLayout->addLayout(editLayout);
mainLayout->addLayout(checkBoxLayout);
mainLayout->addLayout(radioLayout);
mainLayout->addStretch();
mainLayout->addLayout(buttonLayout);

QLabel *nameSurname = new QLabel("username:");
QLabel *firstYear= new QLabel("Password");
QLabel *returning = new QLabel("Are you returning student?");
QLabel * first = new QLabel("1st Year");
QLabel * sec = new QLabel("2nd Year");
QLabel * third = new QLabel("3rd Year");
QLabel *famMembers = new QLabel("Family Members:");

QLineEdit *details = new QLineEdit;
QCheckBox *check1 = new QCheckBox;
QCheckBox *check2 = new QCheckBox;
QRadioButton *btn1 = new QRadioButton;
QRadioButton *btn2 = new QRadioButton;
QRadioButton *btn3 = new QRadioButton;
QSpinBox *spinBox= new QSpinBox;

editLayout->addWidget(nameSurname,0,0);
editLayout->addWidget(details,0,1);
editLayout->addWidget(returning);
editLayout->addWidget(check2);
editLayout->addWidget(firstYear);
editLayout->addWidget(check1);

editLayout->addWidget(first);
editLayout->addWidget(btn1);
editLayout->addWidget(sec);
editLayout->addWidget(btn2);
editLayout->addWidget(third);
editLayout->addWidget(btn3);
editLayout->addWidget(famMembers);
editLayout->addWidget(spinBox);

//Button layout implemataion
QPushButton *btnQuit = new QPushButton("Exit");
QPushButton *btnNext = new QPushButton("Next");
QPushButton * btnBack = new QPushButton("Back");

buttonLayout->addWidget(btnBack);
buttonLayout->addWidget(btnNext);
buttonLayout->addStretch();
buttonLayout->addWidget(btnQuit);

w.show();
return a.exec();
}

```

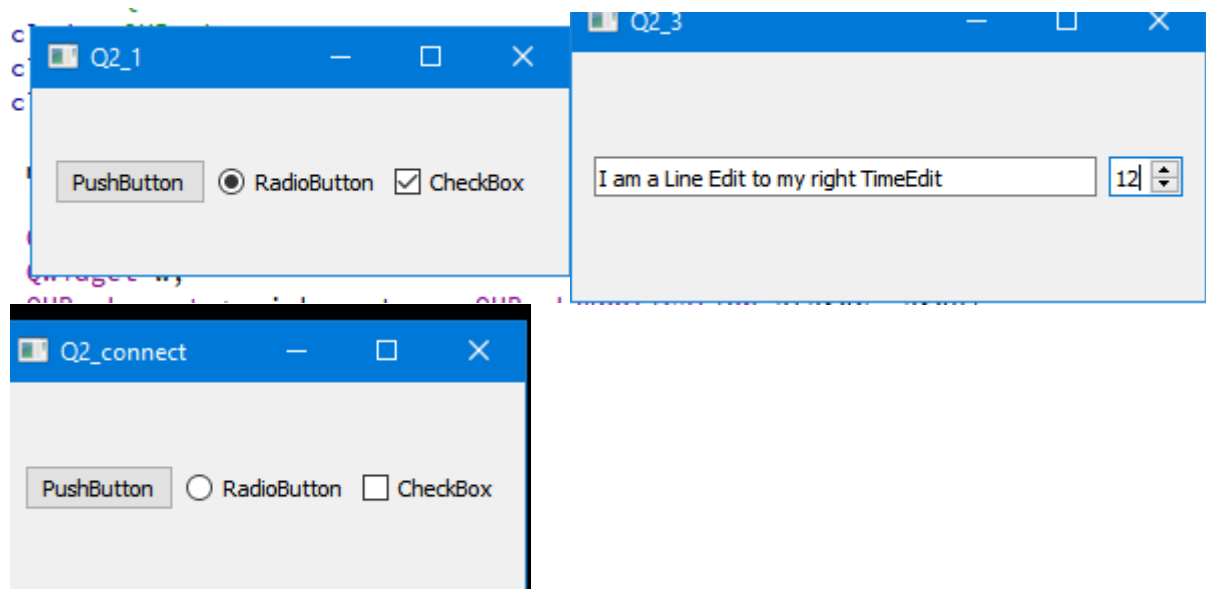
# Portfolio 3

## Description of what it does

The application is based on learning outcome 2. The application gives an introduction to the Qt Layouts where it uses a Horizontal, vertical and a Grid Layout to place the components. But still on the back end coding or console application.

## Screenshots

The PushButton, RadioButton and check Box is displayed on Horizontal Layout.



## Source Code

```
#include <QApplication>
#include <QPushButton>
#include <QHBoxLayout>
#include <QRadioButton>
#include <QCheckBox>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;
    QHBoxLayout *mainLayout=new QHBoxLayout (&w);

    QPushButton *btn=new QPushButton("PushButton");
    QRadioButton *rad=new QRadioButton("RadioButton");
```



```

        QCheckBox *check=new QCheckBox("CheckBox");

        mainLayout->addWidget(btn);
        mainLayout->addWidget(rad);
        mainLayout->addWidget(check);

        w.show();
        return a.exec();
    }

```

```

#include <QApplication>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QPushButton>
#include <QRadioButton>
#include <QCheckBox>
#include <QLineEdit>
#include <QSpinBox>

```

```

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;

    QVBoxLayout *vLay=new QVBoxLayout(&w);
    QHBoxLayout *mainLayout=new QHBoxLayout();
    QHBoxLayout *mainLayout2=new QHBoxLayout();

    QPushButton *pushButton=new QPushButton("PushButton");
    QRadioButton *rad=new QRadioButton("RadioButton");
    QCheckBox *check=new QCheckBox("CheckBox");
    QSpinBox *spn=new QSpinBox();
    QLineEdit *edtLine=new QLineEdit();

    mainLayout->addLayout(pushButton);
    mainLayout->addStretch();
    mainLayout->addLayout(rad);

    vLay->addWidget(mainLayout);
    vLay->addWidget(rad);

    vLay->addWidget(check);
    vLay->addWidget(spn);

    vLay->addWidget(mainLayout2);

    w.show();
    return a.exec();
}

```

```

#include <QApplication>
#include <QHBoxLayout>
#include <QVBoxLayout>
#include <QPushButton>
#include <QRadioButton>
#include <QCheckBox>
#include <QLineEdit>
#include <QSpinBox>

```

```

int main(int argc, char *argv[])

```

```

{
    QApplication a(argc, argv);
    QWidget q;

    QHBoxLayout *mainLayout=new QHBoxLayout(&q);

    QPushButton *btn=new QPushButton("PushButton");
    QRadioButton *rad=new QRadioButton("RadioButton");
    QCheckBox *check=new QCheckBox("CheckBox");

    mainLayout->addWidget(btn);
    mainLayout->addWidget(rad);
    mainLayout->addWidget(check);

    QObject::connect(btn,SIGNAL(clicked()),check,SLOT(toggle()));

    q.show();
    return a.exec();
}

```

## Portfolio 4

### Description of what it does

A register Form that allow a user to put in the registration details about him/herself. The application also uses the SIGNALS and SLOTS to connect the age spinBox and The Age slider to move or set same values as it changes at the other side.

### Screenshots

register\_login

REGISTRATION FORM

Username:

Password:

E-Mail:

Adress:

Age:

Male ☐ Female ☐ Other ☐

Register

register\_login

REGISTRATION FORM

Username:

Password:

E-Mail:

Adress:

Age:

Male ☐ Female ☒ Other ☒

Register

The image displays two screenshots of a Qt-based registration form titled "register\_login". The form contains the following fields and controls:

- Username:** mkgdbnndn
- Password:** nbcbhcj
- E-Mail:** nmsnnjd
- Address:** 1995
- Age:** 26 (with a slider control)
- Gender:** Male ☐ Female ☐ Other ☐
- Register Button:** A button labeled "Register".

The bottom screenshot shows the same form with the Age field updated to 46.

## Source Code

```
#include <QApplication>
#include <QLabel>
#include <QPushButton>
#include <QTextStream>
#include <QCheckBox>
#include <QVBoxLayout>
#include <QHBoxLayout>
#include <QLineEdit>
#include <QWidget>
#include <QSlider>
#include <QSpinBox>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;

    QVBoxLayout *mainLayout=new QVBoxLayout(&w);
    QLabel *lblheading=new QLabel("REGISTRATION FORM");
    QLabel *lblusername=new QLabel("Username: ");
    QLabel *lblpassword=new QLabel("Password: ");
    QLabel *lblemail=new QLabel("E-Mail: ");
    QLabel *lbladdress=new QLabel("Adress: ");
    QLabel *lblage=new QLabel("Age: ");
```

```

QLabel *lblmale=new QLabel("Male");
QLabel *lblfemale=new QLabel("Female");
QLabel *lblother=new QLabel("Other");

QLineEdit *edtusername=new QLineEdit();
QLineEdit *edtpassword=new QLineEdit();
QLineEdit *edtemail=new QLineEdit();
QLineEdit *edtadress=new QLineEdit();
QSpinBox *edtage=new QSpinBox();
QSlider *sldAge=new QSlider(Qt::Horizontal);

QCheckBox *male=new QCheckBox();
QCheckBox *female=new QCheckBox();
QCheckBox *other=new QCheckBox();

QPushButton *regBtn=new QPushButton("Register");

QHBoxLayout *horUser=new QHBoxLayout;
QHBoxLayout *horPass=new QHBoxLayout;
QHBoxLayout *horEmail=new QHBoxLayout;
QHBoxLayout *horAdress=new QHBoxLayout;
QHBoxLayout *horAge=new QHBoxLayout;
QHBoxLayout *horGen=new QHBoxLayout;

horUser->addWidget(lblusername);
horUser->addWidget(edtusername);

horPass->addWidget(lblpassword);
horPass->addWidget(edtpassword);

horEmail->addWidget(lblemail);
horEmail->addWidget(edtemail);

horAdress->addWidget(lbladress);
horAdress->addWidget(edtadress);

horAge->addWidget(lblage);
horAge->addWidget(edtage);
horAge->addWidget(sldAge);

horGen->addWidget(lblmale);
horGen->addWidget(male);
horGen->addWidget(lblfemale);
horGen->addWidget(female);
horGen->addWidget(lblother);
horGen->addWidget(other);

mainLayout->addWidget(lblheading,0,Qt::AlignCenter);
mainLayout->addLayout(horUser);
mainLayout->addLayout(horPass);
mainLayout->addLayout(horEmail);
mainLayout->addLayout(horAdress);
mainLayout->addLayout(horAge);
mainLayout->addLayout(horGen);

mainLayout->addWidget(regBtn,0,Qt::AlignRight);
edtage->setRange(16,60);
sldAge->setRange(16,60);

```

```

QObject::connect(edtage,SIGNAL(valueChanged(int)),sldAge,SLOT(setValue(int))
);

QObject::connect(sldAge,SIGNAL(valueChanged(int)),edtage,SLOT(setValue(int))
);

    w.resize(300,150);
    w.show();
    return a.exec();
}

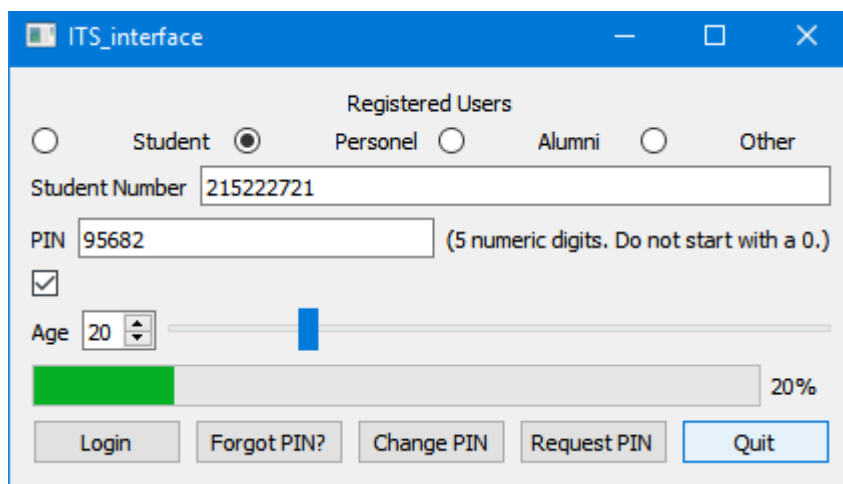
```

## Portfolio 5

### Description of what it does

The application resembles the interface of an old ITs interface that the students or users interact with to access their detailed stuff. The user is allowed to put in their details to login and check their stuff. The staus bar is also included on the interface to show thw process or number that is entered on the spin box.

### Screenshots



ITS\_interface

Registered Users

☐ Student
 ☒ Personnel
 ☐ Alumni
 ☐ Other

Student Number

PIN  (5 numeric digits. Do not start with a 0.)

☒

Age

50%

ITS\_interface

Registered Users

☐ Student
 ☒ Personnel
 ☐ Alumni
 ☐ Other

Student Number

PIN  (5 numeric digits. Do not start with a 0.)

☒

Age

50%

ITS\_interface

Registered Users

☐ Student
 ☒ Personnel
 ☐ Alumni
 ☐ Other

Student Number

PIN  (5 numeric digits. Do not start with a 0.)

☒

Age

80%

## Source Code

```
#include <QApplication>
#include <QLabel>
#include <QSpinBox>
#include <QPushButton>
#include <QHBoxLayout>
```

```

#include <QVBoxLayout>
#include <QLineEdit>
#include <QWidget>
#include <QRadioButton>
#include <QPixmap>
#include <QCheckBox>
#include <QString>
#include <QTextStream>
#include <QSlider>
#include <QProgressBar>
#include <QMessageBox>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QWidget w;

    QVBoxLayout *mainLayout=new QVBoxLayout(&w);
    QLabel *heading=new QLabel("Registered Users");

    QLabel *lblStudNum=new QLabel("Student Number");
    QLabel *lblPin=new QLabel("PIN");
    QLabel *lblPinInst=new QLabel("(5 numeric digits. Do not start with a
0.)");

    //Labels for radio buttons
    QLabel *lblStud=new QLabel("Student");
    QLabel *lblPers=new QLabel("Personel");
    QLabel *lblAlum=new QLabel("Alumni");
    QLabel *lblOther=new QLabel("Other");
    QLabel *lblAge=new QLabel("Age");

    //Line Edit
    QLineEdit *edtStudNum=new QLineEdit();
    QLineEdit *edtNumPin=new QLineEdit();

    //ProgressBar
    QProgressBar*studProg=new QProgressBar();

    //Sliders
    QSlider* sldAge=new QSlider(Qt::Horizontal);
    QSpinBox *spnAge=new QSpinBox();

    //Radio Buttons
    QRadioButton *radioStud=new QRadioButton();
    QRadioButton *radioPers=new QRadioButton();
    QRadioButton *radioAlum=new QRadioButton();
    QRadioButton *radioOther=new QRadioButton();

    //Push button
    QPushButton *login=new QPushButton("Login");
    QPushButton *fPin=new QPushButton("Forgot PIN?");
    QPushButton *chPin=new QPushButton("Change PIN");
    QPushButton *reqPin=new QPushButton("Request PIN");
    QPushButton *butQuit=new QPushButton("Quit");

    //CheckBoxes
    QCheckBox *logCheckBox=new QCheckBox;

    //Horizontal buttons
    QHBoxLayout *horButtons=new QHBoxLayout();

```



```

//Horizontal Radio Buttons
QHBoxLayout *horRadioBut=new QHBoxLayout();
QHBoxLayout *horStudNum=new QHBoxLayout();
QHBoxLayout *horPin=new QHBoxLayout();
QHBoxLayout*horSpnSld=new QHBoxLayout();

//Horizontal layout
horStudNum->addWidget(lblStudNum);
horStudNum->addWidget(edtStudNum);
horPin->addWidget(lblPin);
horPin->addWidget(edtNumPin);
horPin->addWidget(lblPinInst);

//Horizontal Radio Buttons
horRadioBut->addWidget(radioStud);
horRadioBut->addWidget(lblStud);
horRadioBut->addWidget(radioPers);
horRadioBut->addWidget(lblPers);
horRadioBut->addWidget(radioAlum);
horRadioBut->addWidget(lblAlum);
horRadioBut->addWidget(radioOther);
horRadioBut->addWidget(lblOther);

//Horizontal Buttons
horButtons->addWidget(login);
horButtons->addWidget(fPin);
horButtons->addWidget(chPin);
horButtons->addWidget(reqPin);
horButtons->addWidget(butQuit);

//Horizontal age SpinBox and Slider
horSpnSld->addWidget(lblAge);
horSpnSld->addWidget(spnAge,0,Qt::AlignRight);
horSpnSld->addWidget(sldAge);

QObject::connect(butQuit, SIGNAL(clicked()),&a,SLOT(quit()));

//main layout
mainLayout->addWidget(heading,0,Qt::AlignCenter);

//Horizontal Radio Buttons display
mainLayout->addLayout(horRadioBut);

//Horizontal layout Edit and labels display
mainLayout->addLayout(horStudNum);
mainLayout->addLayout(horPin);

//Check Boxes layout
mainLayout->addWidget(logCheckBox);

//Horizontal SpinBox And slider
mainLayout->addLayout(horSpnSld);

mainLayout->addWidget(studProg);
//Stretch
mainLayout->addStretch();

//Horizontal Push Buttons display
mainLayout->addLayout(horButtons);

```

```

QObject::connect (spnAge, SIGNAL (valueChanged (int)) , sldAge, SLOT (setValue (int)
));

QObject::connect (spnAge, SIGNAL (valueChanged (int)) , studProg, SLOT (setValue (in
t)));

QObject::connect (sldAge, SIGNAL (valueChanged (int)) , studProg, SLOT (setValue (in
t)));

QObject::connect (sldAge, SIGNAL (valueChanged (int)) , spnAge, SLOT (setValue (int)
));
    QObject::connect (login, SIGNAL (clicked ()) , &a, SLOT (quit ()) );

    if (butQuit->isChecked ()==1)
    {
        QMessageBox::question (butQuit, "Verification", "Click (OK) to Quit");
    }
    w.show ();
    return a.exec ();
}

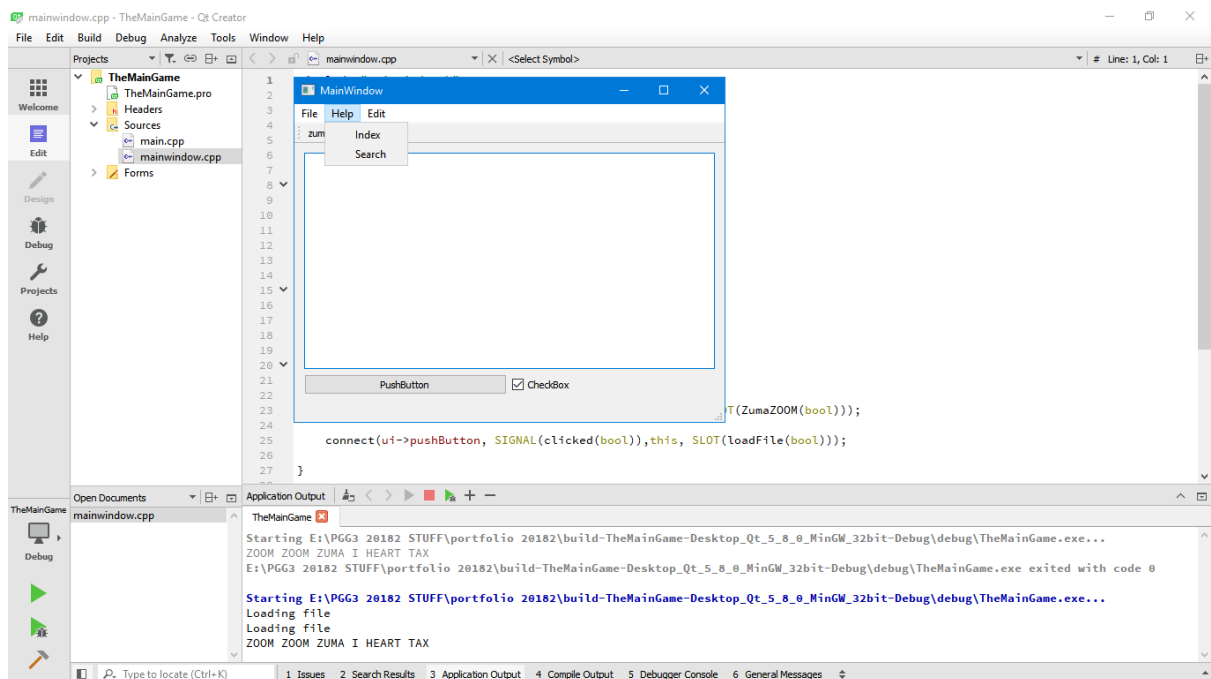
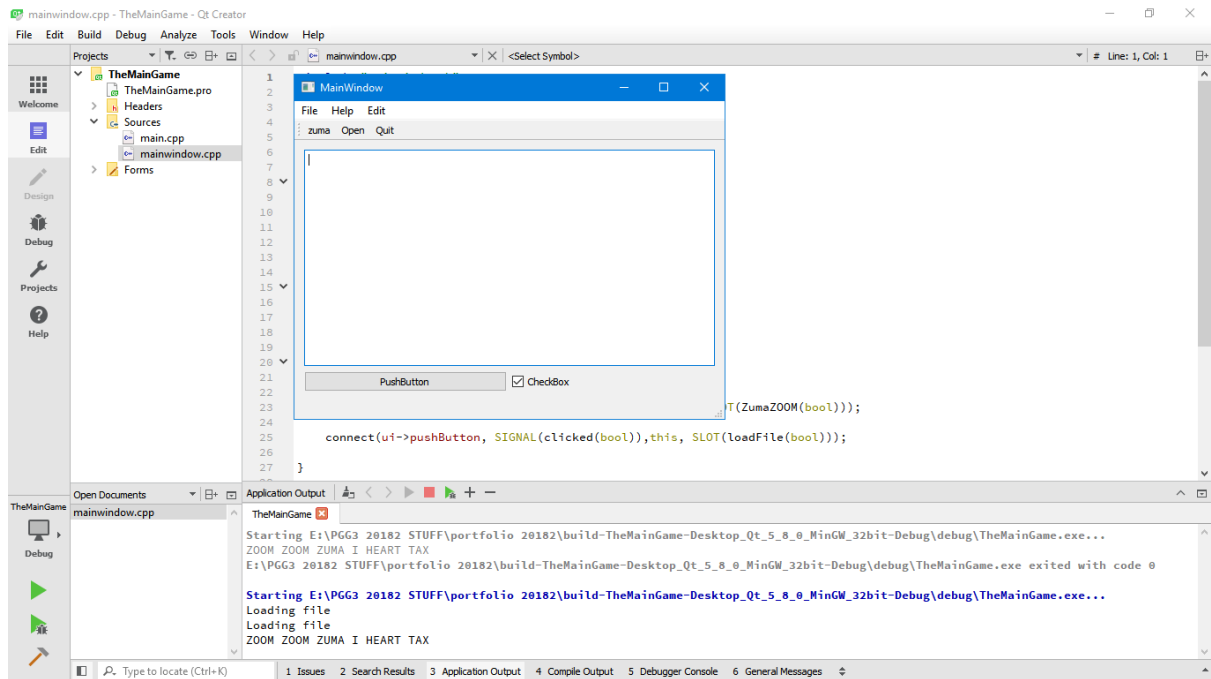
```

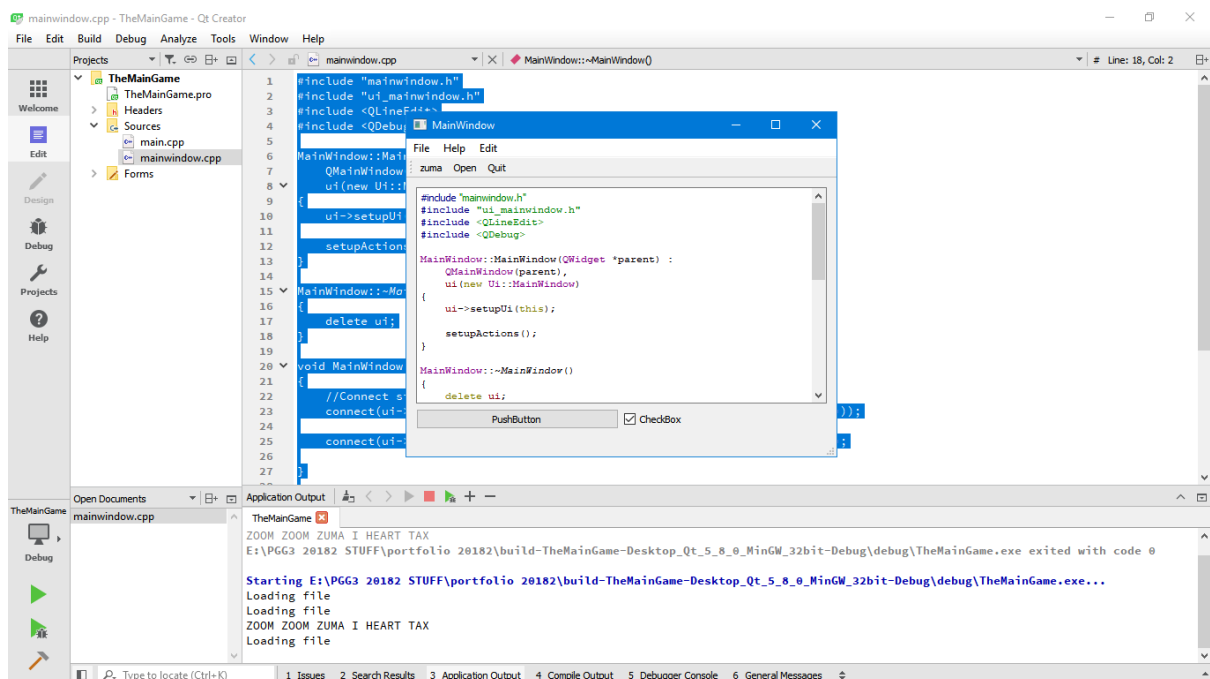
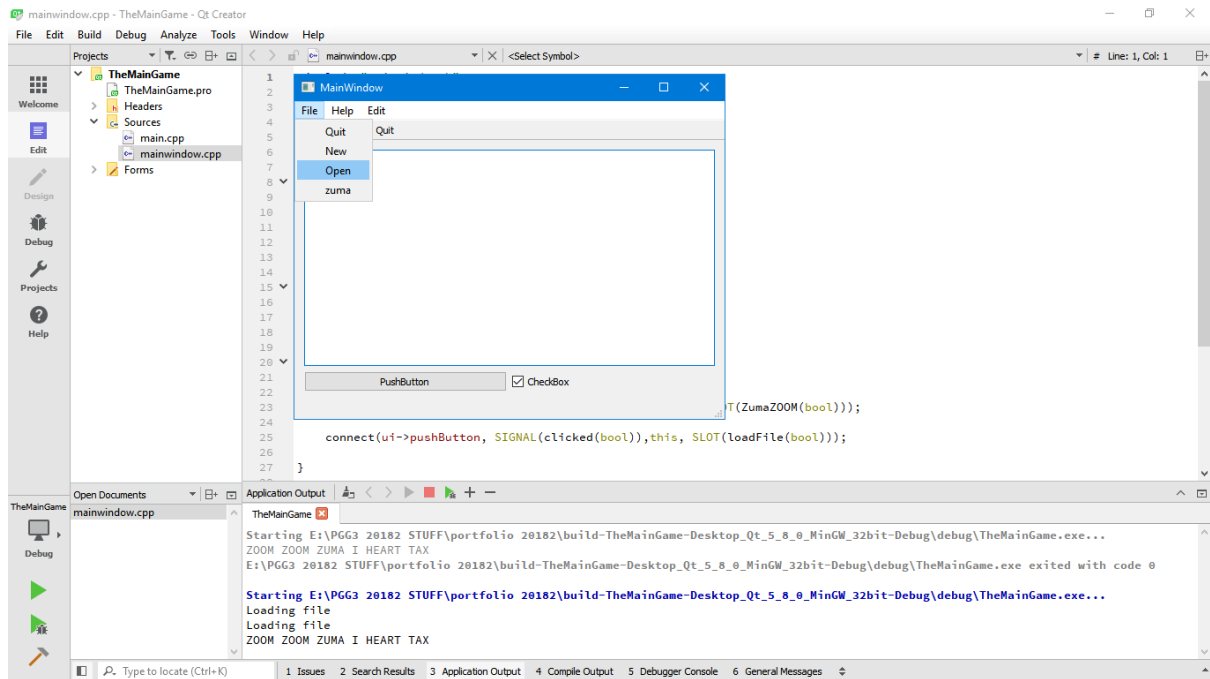
## Portfolio 6

### Description of what it does

The application is a game that works as a word editor thst slows a user to type any amount of words or copy and paste strings of word on the plain document. When the push button is pressed, the string "Load file" is written om the application output. It also has a link to the help desk.

### Screenshots





## Source Code

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QLineEdit>
```

```

#include <QDebug>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    setupActions();
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::setupActions()
{
    //Connect signals to slots
    connect(ui->actionzuma, SIGNAL(triggered(bool)),this,
    SLOT(ZumaZOOM(bool)));
    connect(ui->pushButton, SIGNAL(clicked(bool)),this,
    SLOT(loadFile(bool)));
}

void MainWindow::loadFile(bool ok)
{
    qDebug() << "Loading file";
}

void MainWindow::ZumaZOOM(bool checked)
{
    qDebug() << "ZOOM ZOOM ZUMA I HEART TAX";
}

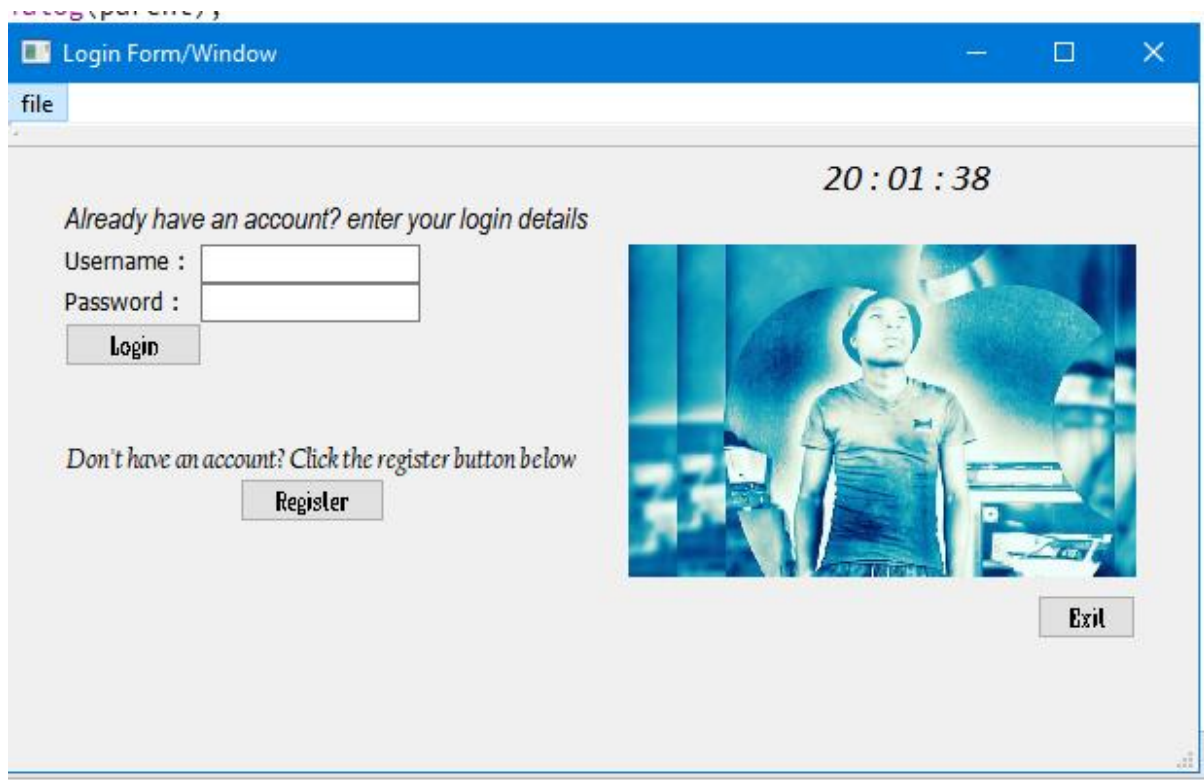
```

## Portfolio 7

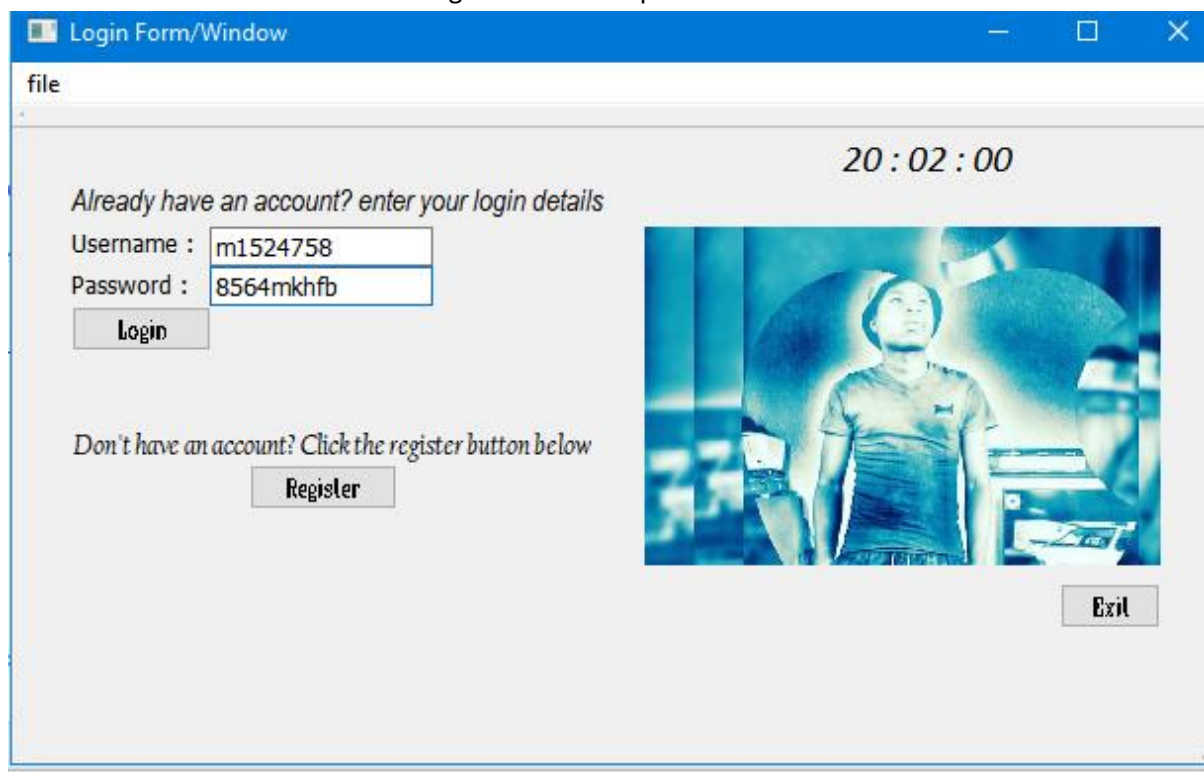
### Description of what it does

The application is an online registration. Where users or people interact with the system on the front end interface. It provides a space for a username and password if the user already owns an account. The application also shows a running system time for the user to check as the process goes on. If the user does not have an account he/she can click the register button to go to the window where a user can have a registration form to register or create account. The application also gives the message "You are successfully registered" when the register form is fully filled in.

### Screenshots



The main window where there is a login details and spaces



Show the running time as it changes

Login Form/Window

file


20:02:05

*Already have an account? enter your login details*

Username :

Password :

*Don't have an account? Click the register button below*



Dialog

*Enter your details below to register*

First Name(s) :

Last Name :

Birth Date :

Id Number :

Age :

Phone Number :

Enter your details below to register

First Name(s) : john

Last Name : doe

Birth Date : 1/1/2000

Id Number : 968545284545125

Age : 21

Phone Number : 0767833354

Register Back

## Source Code

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <regdialog.h>
#include <QDebug>
#include <QDateTime>
#include <QMessageBox>
#include <QPixmap>

MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);

    setWindowTitle("Login Form/Window");
    timer=new QTimer(this);
    connect(timer, SIGNAL(timeout()),this,SLOT(myTime()));
    timer->start(1000);

    QPixmap pix("E:/PGG3 20182 STUFF/portfolio 20182/digitalClock/EDRICH
LOOK-UP APPLE.jpg");
    ui->lblPix->setPixmap(pix.scaled(350,300,Qt::KeepAspectRatio));
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::myTime()
{
    QTime time=QTime::currentTime();
    QString timeDisplay=time.toString("hh : mm : ss");
    ui->lblTimeDate->setText(timeDisplay);
}
```



```

void MainWindow::on_btnExit_clicked()
{
    this->close();
}

void MainWindow::on_btnReg_clicked()
{
    hide();
    regDialog secForm;
    secForm.setModal(true);
    secForm.exec();
}

#include "regdialog.h"
#include "ui_regdialog.h"
#include <QMessageBox>

regDialog::regDialog(QWidget *parent) :
    QDialog(parent),
    ui(new Ui::regDialog)
{
    ui->setupUi(this);
}

regDialog::~regDialog()
{
    delete ui;
}

void regDialog::on_btnBack_clicked()
{
    QMessageBox::about(this, "Back", "You are about to go back..");
    connect(btnBack, SIGNAL(clicked()), this, SLOT(quit()));
}

void regDialog::on_btnRegister_clicked()
{
    QMessageBox::about(this, "Register Message", "You are successfully
registered..");
}

```

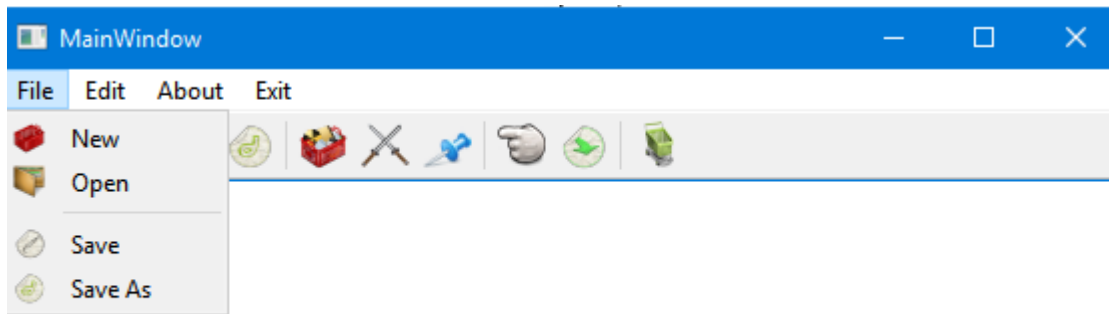
## Portfolio 8

### Description of what it does

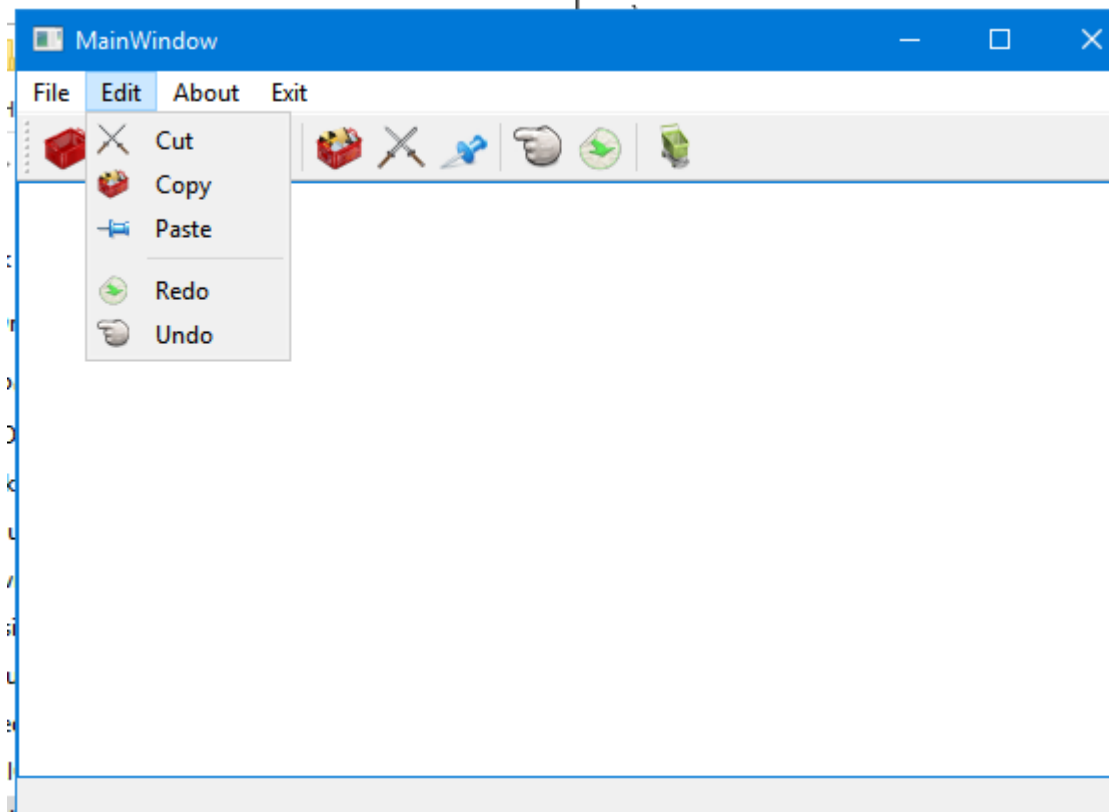
The application is designed to function as a notepad application, the most interesting features about the application is that it contains the icons that describe or shows the actions in the toolbar. The user can use the application to write an editable information. The application also includes the following feature actions: New, Open, Save, Save As, Copy, Paste, Cut, About Notepad and Exit.

The application allows the user to open a new file while the current file is running and edit.

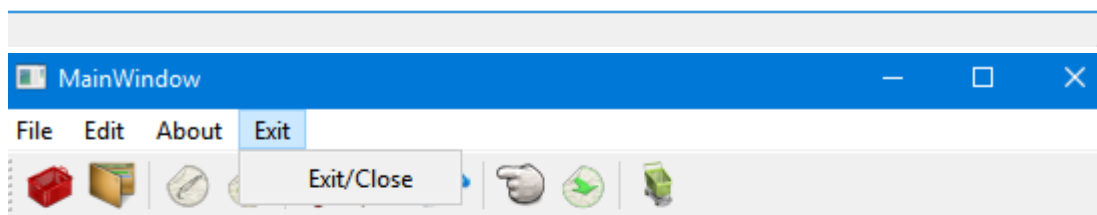
# Screenshots

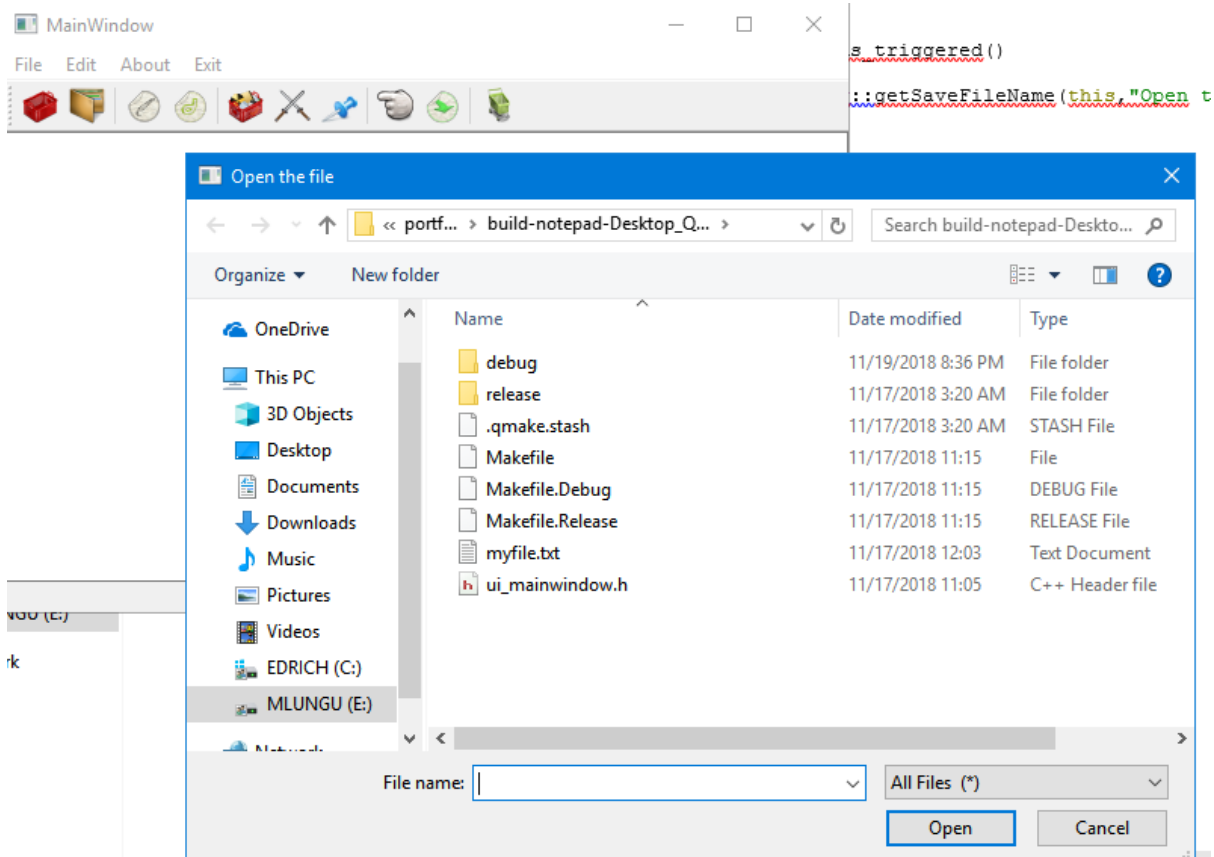


application shows the edit actions .

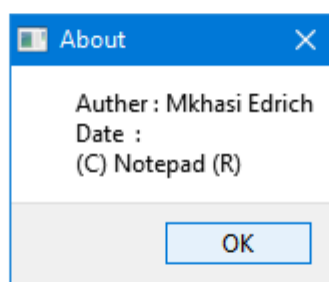


application shows the screenshot for an about action where it shows the the co piled date





the screenshot shows the action open, where it gives the user an option to open another emptyfile.



# Source Code

```
#include "mainwindow.h"
#include "ui_mainwindow.h"
#include <QFile>
#include <QFileDialog>
#include <QTextStream>
#include <QMessageBox>
MainWindow::MainWindow(QWidget *parent) :
    QMainWindow(parent),
    ui(new Ui::MainWindow)
{
    ui->setupUi(this);
    this->setCentralWidget(ui->textEdit);
}

MainWindow::~MainWindow()
{
    delete ui;
}

void MainWindow::on_actionNew_triggered()
{
    filePath="";
    ui->textEdit->setText("");
}

void MainWindow::on_actionOpen_triggered()
{
    QString fileName=QFileDialog::getOpenFileName(this,"Open the file");
    QFile file(fileName);
    filePath=fileName;
    if(!file.open(QFile::ReadOnly | QFile::Text))
    {
        QMessageBox::warning(this,"..","File not open");
        return;
    }
    QTextStream in(&file);
    QString text = in.readAll();
    ui->textEdit->setText(text);
    file.close();
}

void MainWindow::on_actionSave_triggered()
{
    QFile file(filePath);
    if(!file.open(QFile::WriteOnly | QFile::Text))
    {
        QMessageBox::warning(this,"..","File Saved");
        return;
    }
    QTextStream out(&file);
    QString text = ui->textEdit->toPlainText();
    out<<text;
    file.flush();
    file.close();
}

void MainWindow::on_actionSave_As_triggered()
{

```

```

        QString fileName=QFileDialog::getSaveFileName(this,"Open the file");
        QFile file(fileName);
        filePath=fileName;
        if(!file.open(QFile::WriteOnly | QFile::Text))
        {
            return;
        }
        QTextStream out(&file);
        QString text = ui->textEdit->toPlainText();
        out<<text;
        file.flush();
        file.close();
    }

void MainWindow::on_actionCut_triggered()
{
    ui->textEdit->cut();
}

void MainWindow::on_actionCopy_triggered()
{
    ui->textEdit->copy();
}

void MainWindow::on_actionPaste_triggered()
{
    ui->textEdit->paste();
}

void MainWindow::on_actionRedo_triggered()
{
    ui->textEdit->redo();
}

void MainWindow::on_actionUndo_triggered()
{
    ui->textEdit->undo();
}

void MainWindow::on_actionAbout_Notepad_triggered()
{
    QString aboutText;
    aboutText="Auther : Mkhasi Edrich\n";
    aboutText+="Date : \n";
    aboutText+="(C) Notepad (R)";
    QMessageBox::about(this,"About",aboutText);
}

void MainWindow::on_actionExit_Close_triggered()
{
    QApplication::quit();
}

```