МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ

УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЕТ**

о выполнении лабораторной работы № 3

по дисциплине “Кроссплатформенное Программирование”

Выполнил: ст. гр. ИС/б-20-2-о

Белик Г. М.

Проверил: доц. каф.

«Информационные системы»

Строганов В. А.

Севастополь

2023

**ЛАБОРАТОРНАЯ РАБОТА №3**

**«Исследование способов построения интерфейса пользователя на базе QtWidgets»**

**Цель работы**

Изучить основные методики создания графического пользовательского интерфейса с использованием виджетов QtWidgets. Приобрести навыки разработки интерфейса пользователя для приложений на основе фреймворка Qt.

**Задачи**

1. Создать проект Qt Gui Application.
2. Разработать графический интерфейс приложения согласно варианту задания
3. Используя механизм сигналов и слотов добавить функционал, согласно варианту задания.

**Вариант 4**

Тип формы – Форма добавления новости (заголовок (Line Edit), текст новости (Text Edit), дата (Date Edit)).

Функционал – Добавление на форму таблицы и добавление в неё введенных данных.

Метод разработки – Программно.

**Ход работы**

Ниже представлен интерфейс приложения, который видит пользователь при первом открытии программы.

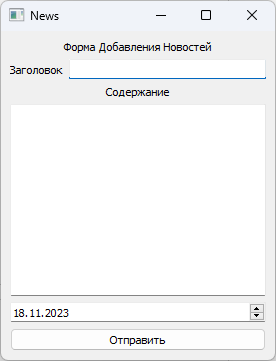


Рисунок 1 – Старт программы

После нажания кнопки “Отправить” появленися таблица с заголовками разделов и новая запись (Рисунок 2).

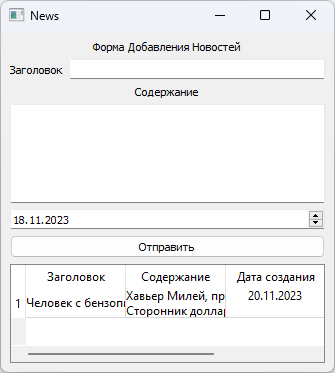


Рисунок 2 – Результат добавления записи в таблицу

В случае добавления новых записей они будут сохраняться и скроллиться в конец таблицы для удобства просмотра новых записей.

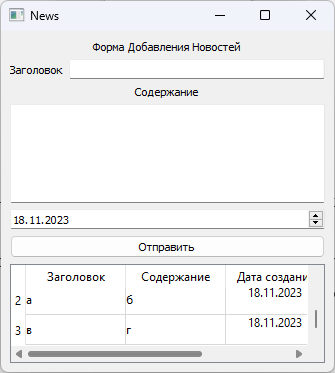


Рисунок 3 – Процесс добавления нескольких записей

Интерфейс явлется адаптивным, поэтому при изменении размера окна внешний вид остется удобным к использованию.

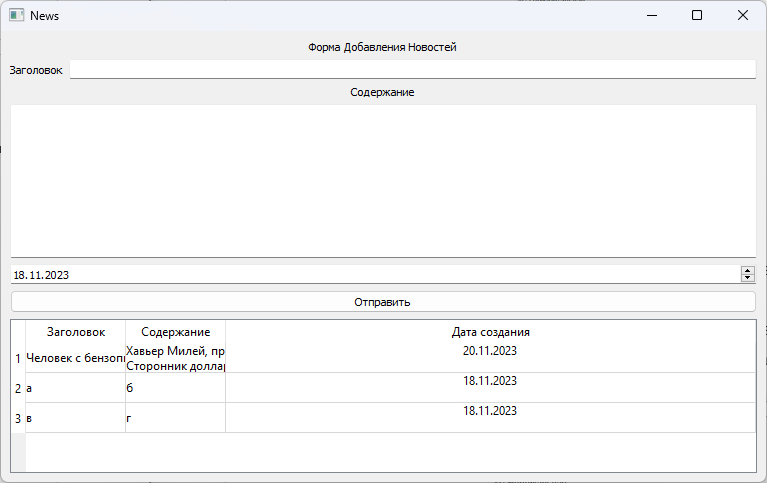


Рисунок 4 – Результат изменения размера окна

**Текст программы:**

**main.cpp**

#include "registrationwindow.h"

#include <QApplication>

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

{

QApplication a(*argc*, argv);

RegistrationWindow w;

w.show();

return a.exec();

}

**registrationwindow.h**

#ifndef REGISTRATIONWINDOW\_H

#define REGISTRATIONWINDOW\_H

#include "ui\_registrationwindow.h"

#include "QLabel"

#include "QLineEdit"

#include "QLayout"

#include "QPushButton"

#include "QTextEdit"

#include "QDateEdit"

#include "QTableWidget"

#include <QMainWindow>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **RegistrationWindow**; }

QT\_END\_NAMESPACE

class **RegistrationWindow** : public QMainWindow

{

Q\_OBJECT

public:

**RegistrationWindow**(QWidget \*parent = nullptr);

~***RegistrationWindow***();

private:

Ui::RegistrationWindow \*ui;

QVBoxLayout \*centralLayout;

QLineEdit \*zagolovokEdit;

QTextEdit \*soderzhanieText;

QDateEdit \*dateEdit;

QTableWidget \*FullTable;

QDate today;

int flag = 0;

int tableheight = 55;

private slots:

void **createTable**();

};

#endif

**registrationwindow.cpp**

#include "registrationwindow.h"

#include "ui\_registrationwindow.h"

#include "QLabel"

#include "QLineEdit"

#include "QLayout"

#include "QPushButton"

#include "QTextEdit"

#include "QDateEdit"

#include "QDate"

#include "QTableWidget"

#include "QHeaderView"

RegistrationWindow::**RegistrationWindow**(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::RegistrationWindow)

{

this->setWindowTitle("News");

QLabel \*zagolovokLabel = new QLabel("Заголовок");

zagolovokEdit = new QLineEdit();

QLayout \*zagolovokLayout = new QHBoxLayout();

zagolovokLayout->addWidget(zagolovokLabel);

zagolovokLayout->addWidget(zagolovokEdit);

QLabel \*soderzhanieLabel = new QLabel("Содержание");

// soderzhanieLabel->setMinimumWidth(330);

soderzhanieLabel->setAlignment(Qt::AlignHCenter);

soderzhanieText = new QTextEdit();

QLayout \*soderzhanieLayout = new QVBoxLayout();

soderzhanieLayout->addWidget(soderzhanieLabel);

soderzhanieLayout->addWidget(soderzhanieText);

QLabel \*FormTitleLabel = new QLabel("Форма Добавления Новостей");

FormTitleLabel->setAlignment(Qt::AlignHCenter);

dateEdit = new QDateEdit();

today= dateEdit->date();

today.setDate(2023,11,18);

dateEdit->setDate(today);

QPushButton \*sendButton = new QPushButton("Отправить");

centralLayout = new QVBoxLayout();

centralLayout->setAlignment(Qt::AlignHCenter | Qt::AlignVCenter);

centralLayout->addWidget(FormTitleLabel);

centralLayout->addLayout(zagolovokLayout);

centralLayout->addLayout(soderzhanieLayout);

centralLayout->addWidget(dateEdit);

centralLayout->addWidget(sendButton);

QWidget \*centralWidget = new QWidget();

centralWidget->setLayout(centralLayout);

this->setCentralWidget(centralWidget);

connect(sendButton, SIGNAL(clicked()), this, SLOT(createTable()));

}

RegistrationWindow::~***RegistrationWindow***()

{

//delete ui;

}

void RegistrationWindow::**createTable**()

{

if (flag == 0) {

FullTable = new QTableWidget(0, 3, nullptr);

FullTable->setHorizontalHeaderLabels({"Заголовок", "Содержание", "Дата создания"});

centralLayout->addWidget(FullTable);

flag = 1;

}

QString zagolovok = zagolovokEdit->text();

QString soderzhanie = soderzhanieText->toPlainText();

QDate date = dateEdit->date();

int row = FullTable->rowCount();

FullTable->insertRow(row);

//FullTable->setMinimumSize(340,tableheight);

FullTable->setMinimumWidth(340);

tableheight+=30;

QLabel \*zagolovokLabel = new QLabel(zagolovok);

QLabel \*soderzhanieLabel = new QLabel(soderzhanie);

QLabel \*dateLabel = new QLabel(date.toString("dd.MM.yyyy"));

FullTable->setCellWidget(row, 0, zagolovokLabel);

FullTable->setCellWidget(row, 1, soderzhanieLabel);

FullTable->setCellWidget(row, 2, dateLabel);

dateLabel->setAlignment(Qt::AlignHCenter);

FullTable->horizontalHeader()->setStretchLastSection(true);

zagolovokEdit->clear();

soderzhanieText->clear();

today= dateEdit->date();

today.setDate(2023,11,18);

dateEdit->setDate(today);

FullTable->scrollToBottom();

}

**Выводы**

В ходе лабораторной работы были исследованы способы разработки интерфейса, а именно с помощью Qt designer и программно в cpp файле. Программный способ позволяет немного расширить функциональность, однако требует более глубоких знаний и большего количества времени. С использованием различных виджетов были скомпанованы горизонтальные слои, которые в последующем были вертикально раположены друг за другом, что позволило создать адаптивый интерфейс под размер окна при изменении его длины и ширины.