

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ



### **ЗВІТ**

**Про виконання лабораторної роботи № 4  
з дисципліни «Вступ до інженерії програмного забезпечення»**

**Лекторка:**

доцент Левус Є. В.

**Виконала:**

студ. групи ПЗ-16

Голомша О.Я.

**Прийняв:**

викладач Самбір А. А.

«\_\_\_» \_\_\_\_\_ 2022 р.

$\Sigma$  = \_\_\_\_\_

Львів – 2022

**Тема.** Підвищення зручності користування програмою як задача супроводу ПЗ.

**Мета.** Навчитися розробляти ескізи інтерфейсу користувача та створювати інструкцію користувача.

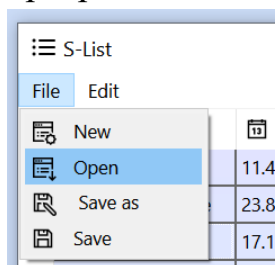
### Теоретичні відомості

2. У чому полягає етап експлуатації ПЗ? Що є вхідними даними для цього етапу?

Етап експлуатації полягає у тому що готове ПЗ передають у використання, визначають ефективність використання. Вхідними даними для цього етапу є протестована програма без серйозних помилок у роботі.

20. Наведіть типові приклади реалізації принципу «Сповіщення про поточний стан».

- Коли користувач наводить курсор миші на елемент графічного інтерфейсу чи натискає на нього, цей елемент змінює свій вигляд, даючи знати що відбувається взаємодія саме з цією частиною програмної системи



Наприклад, коли я наводжу курсор на одну з кнопок своєї програми, вона підсвічується синім кольором.

- Використання прогрес-бару. Найчастіше їх використовують під час маніпуляцій з об'ємними файлами, наприклад, під час їх завантаження чи копіювання.
- Повідомлення про те де перебуває користувач, наприклад, якщо програма має декілька вкладок, треба щоб активна вкладка якось виділялась.

19. Як можуть бути виміряні характеристики зручного інтерфейсу?

Вони можуть бути виміряні такими чинниками:

- легкість освоєння системи користувачем;

- ефективність роботи з системою після освоєння її графічного інтерфейсу;
- запам'ятовуваність стилю використання графічного інтерфейсу програми;
- частота допускання помилок користувачем;
- загальне задоволення від користування програмною системою.

### Постановка завдання

1. Розробити прототип UI до наступної версії програми, яка була протестована у попередній лабораторній роботі. Інтерфейс має забезпечити зручність користування програмою і проілюструвати виконання евристик побудови зручного інтерфейсу користувача.

Розробити дизайн повідомлень усіх вище перерахованих типів для своєї програми.

Для виконання цього завдання потрібно зобразити ескіз інтерфейсу програми та повідомлень набором рисунків. Рисунки мають у повній мірі демонструвати усі можливі варіанти взаємодії користувача з програмою.

2. Скласти інструкцію користувача для програми (з передбаченим інтерфейсом).

Для виконання цього завдання потрібно використати описані рекомендації для створення інструкції користувача зі скороченим змістом.

- 3\*. Імплементувати розробку GUI для нової версії програми.

### Отримані результати

The screenshot shows a web application titled "S-List". It features a header with "File" and "Edit" menus. Below the header is a table-like structure with the following columns: "Name", "Date of birth", "Grade 1", "Grade 2", "Grade 3", "Grade 4", "Grade 5", and "Rating". Each grade column has a dropdown menu with "A+" selected. The "Rating" column has a star icon. The main content area is a large light blue rectangle. At the bottom, there is a footer with a text input field containing "Holomsha Oksana...", a date dropdown set to "01.01.2004", five grade dropdowns, and a plus button.

S-List

File Edit

Name

Date of birth

Grade 1

Grade 2

Grade 3

Grade 4

Grade 5

Rating

←

липень 2004

→

Пн	Вт	Ср	Чт	Пт	Сб	Нд
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Harry Potter

21.07.2004

4

Grade 2

Grade 3

Grade 4

Grade 5

+

S-List

File Edit

Name

Date of birth

Grade 1

Grade 2

Grade 3

Grade 4

Grade 5

Rating

Grade 5

2

3

4

5

Harry Potter

21.07.2004

4

5

4

3

4

+

S-List

File

Edit

New

Open

Save as

Save

		Date of birth	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Rating
		11.4.2004	4	5	5	5	5	4.8
		23.8.2004	5	5	5	5	3	4.6
		17.1.2004	2	4	5	5	3	3.8
4	Lockart Gilderoy	27.12.2004	2	2	3	3	5	3
5	Moody Alastor	18.9.2004	3	3	3	4	5	3.6
6	McGonagall Minerva	7.4.2004	5	5	4	4	5	4.6
7	Umbridge Dolores	13.7.2004	2	2	2	2	3	2.2
8	Lestrangle Bellatrix	3.5.2004	4	4	4	4	4	4
9	Riddle Tom	14.5.2004	5	5	5	5	5	5
10	Dumbledore Albus	21.3.2004	2	3	2	2	3	2.4

Holomsha Oksana...

21.07.2004

Grade 1

Grade 2

Grade 3

Grade 4

Grade 5

+

S-List

File

Edit

+

 Add student

-

 Delete student
 Clear

		Date of birth	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Rating
1	Potter Harry	31.12.1980	4	5	5	5	5	4.8
2	Granger Hermione	19.9.1980	5	5	5	5	3	4.6
3	Weasley Ronald	17.1.2004	2	4	5	5	3	3.8
4	Lockart Gilderoy	27.12.2004	2	2	3	3	5	3
5	Moody Alastor	18.9.2004	3	3	3	4	5	3.6
6	McGonagall Minerva	7.4.2004	5	5	4	4	5	4.6
7	Umbridge Dolores	13.7.2004	2	2	2	2	3	2.2
8	Lestrangle Bellatrix	3.5.2004	4	4	4	4	4	4
9	Riddle Tom	14.5.2004	5	5	5	5	5	5
10	Dumbledore Albus	21.3.2004	2	3	2	2	3	2.4

Holomsha Oksana...

21.07.2004

Grade 1

Grade 2

Grade 3

Grade 4

Grade 5

+

S-List

File Edit

	Name	Date of birth	Grade 1	Grade 2	Grade 3	Grade 4	Grade 5	Rating
1	Potter Harry	11.4.2004	4	5	5	5	5	4.8
2	Granger Hermione	23.8.2004	5	5	5	5	3	4.6
3	Weasley Ronald	17.1.2004	2	4	5	5	3	3.8
4	Lockart Gilderoy	27.12.2004	2	2	3	3	5	3
5	Moody Alastor	18.9.2004	3	3	3	4	5	3.6
6	McGonagall Minerva	7.4.2004	5	5	4	4	5	4.6
7	Umbridge Dolores	13.7.2004	2	2	2	2	3	2.2
8	Lestrangle Bellatrix	3.5.2004	4	4	4	4	4	4
9	Riddle Tom	14.5.2004	5	5	5	5	5	5
10	Dumbledore Albus	21.3.2004	2	3	2	2	3	2.4
11	Holomsha Oksana	21.07.2004	3	3	4	5	2	3.4

Holomsha Oksana...

21.07.2004

Grade 1

Grade 2

Grade 3

Grade 4

Grade 5

Рис. 1-6. GUI нової версії програми

SList

Are you sure you want to add this student?

Student with this name is already in the view.

Yes

Replace

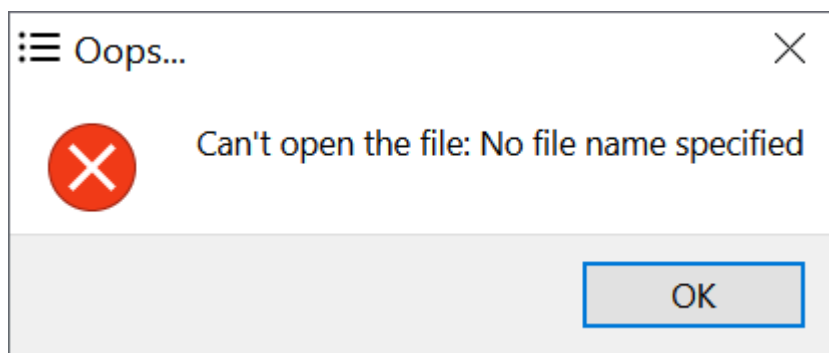
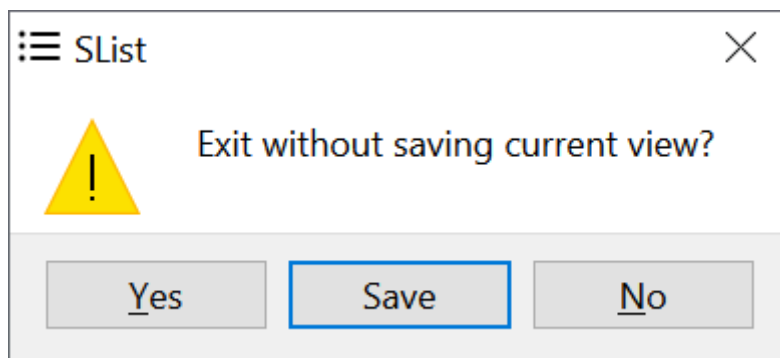
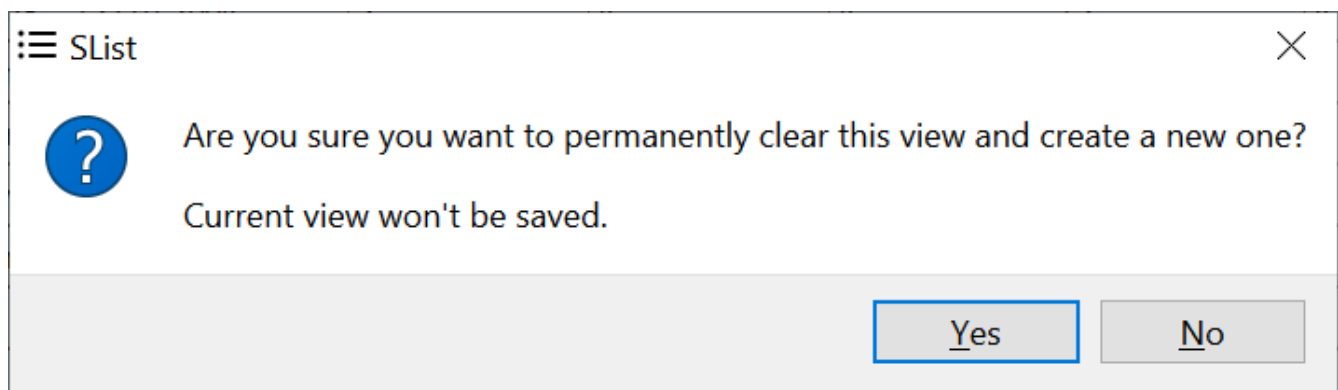
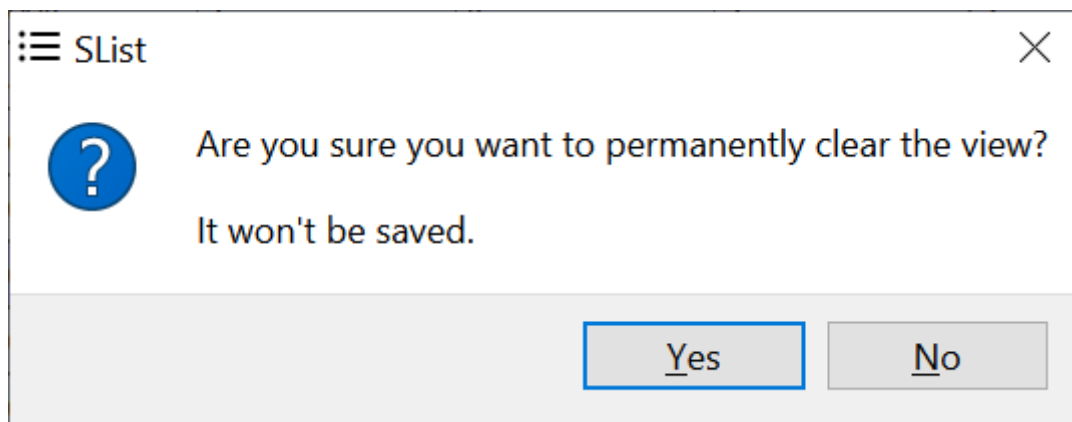
No

SList

Are you sure you want to remove selected row?

Yes

No



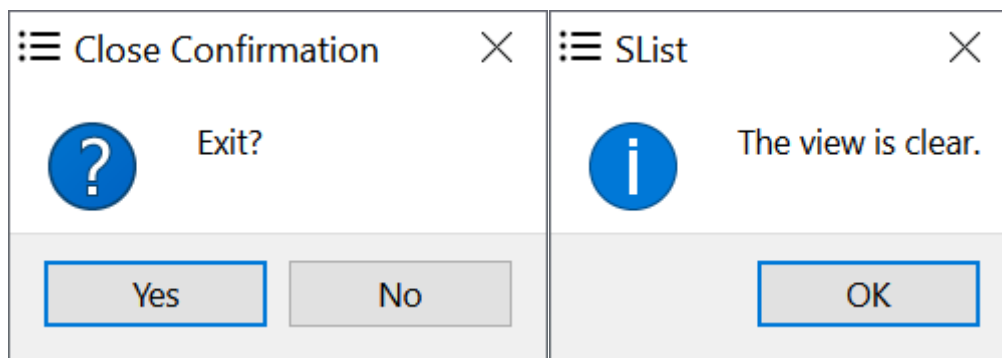


Рис. 7-14. Дизайн повідомлень

## Інструкція користувача

### 1. Компоненти ПЗ.

Програмну систему розроблено мовою Qt C++ 5.10 у середовищі розробки Qt Creator 7.0.0 і може експлуатуватися під управлінням ОС Windows.

Для коректної роботи програми необхідний комп'ютер з процесором не менше 200 МГц та оперативною пам'яттю не менше 256 Мб. Для експлуатації програми необхідно встановити стандартний пакет Qt Creator 7.0.0.

### 2. Встановлення ПЗ.

Для роботи пакету необхідно встановити пакет CMake 3.21.1 64-bit та Ninja 1.10.2.

### 3. Налаштування ПЗ.

Для коректної роботи програми непомірно проводити ніяких налаштувань.

### 4. Базові функції ПЗ.

#### 4.1. Створити нову таблицю:

Щоб створити нову таблицю, в меню "File" оберіть пункт "New".

#### 4.2. Відкрити файл:

Для того щоб вивести дані з файлу на таблицю, в меню "File" оберіть пункт "Open". За допомогою діалогу оберіть текстовий файл, який буде виведено у таблицю.

#### 4.3. Зберегти таблицю як:

Щоб зберегти таблицю як, в меню "File" оберіть пункт "Save as". За допомогою діалогу оберіть локацію збереження файлу та його назву.

#### 4.4. Зберегти таблицю в той самий файл:

Для збереження таблиці в той самий файл в меню "File" оберіть пункт "Save".

#### 4.5. Додати нового студента:



Щоб додати нового студента виконайте такі кроки:

- a) Уведіть прізвище та ім'я студента у графу з ім'ям під таблицею, встановіть його дату народження та 5 оцінок.
- b) Натисніть на кнопку “+” під таблицею або в меню “Edit”/”Add student”.

#### 4.6. Видалити студента:

Щоб видалити студента з таблиці виконайте такі послідовні кроки:

- a) Оберіть з таблиці 1 студента та натисніть на рядок у якому він знаходиться.
- b) У меню “Edit” натисніть на кнопку “Delete student”.

#### 4.7. Очистити таблицю:

Щоб очистити таблицю перейдіть в меню “Edit” та оберіть пункт “Clear”.

### 5. Аналіз можливих помилок.

У випадку зрою перегляньте файл log.txt.

## Код нової версії програми з імплементованим GUI

### Mainwindow.h

```
#ifndef MAINWINDOW_H
#define MAINWINDOW_H

#include <QMainWindow>

QT_BEGIN_NAMESPACE
namespace Ui { class MainWindow; }
QT_END_NAMESPACE

class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private:
    Ui::MainWindow *ui;
    QString m_filename;
    bool m_save_checker;

private slots:
    void open();
    void add();
    void delete_selected();
    void action_clear();
    void new_view();
}
```

```

void save_as();
void save();
void on_actionAdd_element_triggered();
void closeEvent(QCloseEvent *event);
};
#endif // MAINWINDOW_H

```

## Mainwindow.cpp

```

#include "mainwindow.h"
#include "../ui_mainwindow.h"
#include <QFile>
#include <QFileDialog>
#include <QMessageBox>
#include <QDesktopServices>
#include <QStatusBar>
#include <QCloseEvent>
// #include <QHBoxLayout>

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

    ui->tableWidget->horizontalHeader()->setSectionResizeMode(0,
QHeaderView::ResizeToContents);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(1,
QHeaderView::ResizeToContents);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(2,
QHeaderView::Stretch);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(3,
QHeaderView::Stretch);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(4,
QHeaderView::Stretch);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(5,
QHeaderView::Stretch);
    ui->tableWidget->horizontalHeader()->setSectionResizeMode(6,
QHeaderView::Stretch);
    // ui->tableWidget->horizontalHeader()->setSectionResizeMode(7,
QHeaderView::Stretch);

    connect(ui->actionOpen, &QAction::triggered, this, &MainWindow::open);
    connect(ui->actionDelete_student, &QAction::triggered, this,
&MainWindow::delete_selected);
    connect(ui->actionAdd_element, &QAction::triggered, this, &MainWindow::add);
    connect(ui->actionClear, &QAction::triggered, this, &MainWindow::action_clear);
    connect(ui->actionNew, &QAction::triggered, this, &MainWindow::new_view);
    connect(ui->actionSaveAs, &QAction::triggered, this, &MainWindow::save_as);
    connect(ui->actionSave, &QAction::triggered, this, &MainWindow::save);
    connect(ui->btn_add, SIGNAL(clicked(bool)), this, SLOT(add()));
}

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

```

```

void MainWindow::open() {
    QString filename = QFileDialog::getOpenFileName(this, "Open the file");
    m_filename = filename;
    QFile file(filename);
    if(!file.open(QFile::ReadOnly|QFile::Text)){
        QMessageBox::critical(this, "Oops...", "Can't open the file:
"+file.errorString());
        return;
    }
    //file.open(QIODevice::ReadOnly);
    QTextStream in(&file);
    QList< QStringList > lists;
    QString line;
    int count = 0;
    do{
        line = in.readLine();
        lists << line.split(" ; ");
        ++count;
    }while(!line.isNull());

    ui->tableWidget->setRowCount(count - 1);
    for ( int row = 0; row < ui->tableWidget->rowCount(); ++row ) {
        for ( int column = 0; column < ui->tableWidget->columnCount(); ++column ) {
            if(column == 0)
                ui->tableWidget->setVerticalHeaderItem(row, new
QTableWidgetItem(QString::number(row + 1)));
            if(column == ui->tableWidget->columnCount() - 1){
                int mark1 = ui->tableWidget->item(row, 2)->text().toInt();
                int mark2 = ui->tableWidget->item(row, 3)->text().toInt();
                int mark3 = ui->tableWidget->item(row, 4)->text().toInt();
                int mark4 = ui->tableWidget->item(row, 5)->text().toInt();
                int mark5 = ui->tableWidget->item(row, 6)->text().toInt();
                float avgMark = (mark1 + mark2 + mark3 + mark4 + mark5) / 5.0;
                ui->tableWidget->setItem(row, column, new
QTableWidgetItem(QString::number(avgMark)));
            }
            /*else if(column == ui->tableWidget->columnCount() - 1){
                QWidget* pWidget = new QWidget();
                QPushButton* btn_delete = new QPushButton();
                btn_delete->setText("Delete");
                QHBoxLayout* pLayout = new QHBoxLayout(pWidget);
                pLayout->addWidget(btn_delete);
                pLayout->setAlignment(Qt::AlignCenter);
                pLayout->setContentsMargins(0, 0, 0, 0);
                pWidget->setLayout(pLayout);
                ui->tableWidget->setCellWidget(row, column, pWidget);
            }*/
            else
                ui->tableWidget->setItem(row, column, new
QTableWidgetItem(lists[row][column]));
        }
    }
    file.close();
    m_save_checker = false;
}

void MainWindow::add() {
    if(ui->cbox_1->currentText() != "Grade 1") ui->cbox_1->setStyleSheet("");
    if(ui->cbox_2->currentText() != "Grade 2") ui->cbox_2->setStyleSheet("");
}

```

```

if(ui->cbox_3->currentText() != "Grade 3") ui->cbox_3->setStyleSheet("");
if(ui->cbox_4->currentText() != "Grade 4") ui->cbox_4->setStyleSheet("");
if(ui->cbox_5->currentText() != "Grade 5") ui->cbox_5->setStyleSheet("");
if(ui->txt_name->text() != "") ui->txt_name->setStyleSheet("");

bool name_checker = false;
for(int i = 0; i < 10; ++i){
    if(ui->txt_name->text().contains(QString::number(i))) name_checker = true;
    if(ui->txt_name->text().length() < 4) name_checker = true;
    if(!ui->txt_name->text().contains(" ")) name_checker = true;
}

int t = 0;
if(ui->txt_name->text() == "" || name_checker == true){
    ui->txt_name->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(ui->cbox_1->currentText() == "Grade 1"){
    ui->cbox_1->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(ui->cbox_2->currentText() == "Grade 2"){
    ui->cbox_2->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(ui->cbox_3->currentText() == "Grade 3"){
    ui->cbox_3->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(ui->cbox_4->currentText() == "Grade 4"){
    ui->cbox_4->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(ui->cbox_5->currentText() == "Grade 5"){
    ui->cbox_5->setStyleSheet("border: 1px solid #7575ff");
    t++;
}
if(t > 0) return;

QMessageBox msg;
msg.setWindowIcon(this->>windowIcon());
msg.setText("Are you sure you want to add this student?");
msg.setInformativeText("Student with this name is already in the view.");
msg.setIcon(QMessageBox::Question);
QPushButton *btn_yes = msg.addButton(QMessageBox::Yes);
QPushButton *btn_no = msg.addButton(QMessageBox::No);
QPushButton *btn_replace = msg.addButton(tr("Replace"), QMessageBox::YesRole);
for(int i = 0; i < ui->tableWidget->rowCount(); ++i){
    if(ui->tableWidget->item(i, 0)->text() == ui->txt_name->text()) msg.exec();
}
if(msg.clickedButton() == btn_no) return;
if(msg.clickedButton() == btn_replace){
    for(int i = 0; i < ui->tableWidget->rowCount(); ++i){
        if(ui->tableWidget->item(i, 0)->text() == ui->txt_name->text()){
            std::string name = ui->txt_name->text().toStdString();
            name[0] = std::toupper(name[0]);
            int txt_i;
            for(int i = 0; i < name.length(); ++i){

```

```

        if(std::isspace(name[i])) txt_i = i + 1;
    }
    name[txt_i] = std::toupper(name[txt_i]);
    ui->tableWidget->setItem(i, 0, new
QTableWidgetItem(QString::fromStdString(name)));
    ui->tableWidget->setItem(i, 1, new QTableWidgetItem(ui->dateEdit-
>text()));
    ui->tableWidget->setItem(i, 2, new QTableWidgetItem(ui->cbox_1-
>currentText()));
    ui->tableWidget->setItem(i, 3, new QTableWidgetItem(ui->cbox_2-
>currentText()));
    ui->tableWidget->setItem(i, 4, new QTableWidgetItem(ui->cbox_3-
>currentText()));
    ui->tableWidget->setItem(i, 5, new QTableWidgetItem(ui->cbox_4-
>currentText()));
    ui->tableWidget->setItem(i, 6, new QTableWidgetItem(ui->cbox_5-
>currentText()));
    int mark1 = ui->cbox_1->currentText().toInt();
    int mark2 = ui->cbox_2->currentText().toInt();
    int mark3 = ui->cbox_3->currentText().toInt();
    int mark4 = ui->cbox_4->currentText().toInt();
    int mark5 = ui->cbox_5->currentText().toInt();
    float avgMark = (mark1 + mark2 + mark3 + mark4 + mark5) / 5.0;
    ui->tableWidget->setItem(i, 7, new
QTableWidgetItem(QString::number(avgMark)));
    m_save_checker = false;
    return;
}

}

std::string name = ui->txt_name->text().toStdString();
name[0] = std::toupper(name[0]);
int txt_i;
for(int i = 0; i < name.length(); ++i){
    if(std::isspace(name[i])) txt_i = i + 1;
}
name[txt_i] = std::toupper(name[txt_i]);
ui->tableWidget->setRowCount(ui->tableWidget->rowCount() + 1);
ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 0, new
QTableWidgetItem(QString::fromStdString(name)));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 1, new
QTableWidgetItem(ui->dateEdit->text()));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 2, new
QTableWidgetItem(ui->cbox_1->currentText()));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 3, new
QTableWidgetItem(ui->cbox_2->currentText()));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 4, new
QTableWidgetItem(ui->cbox_3->currentText()));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 5, new
QTableWidgetItem(ui->cbox_4->currentText()));
    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 6, new
QTableWidgetItem(ui->cbox_5->currentText()));
    int mark1 = ui->cbox_1->currentText().toInt();
    int mark2 = ui->cbox_2->currentText().toInt();
    int mark3 = ui->cbox_3->currentText().toInt();
    int mark4 = ui->cbox_4->currentText().toInt();
    int mark5 = ui->cbox_5->currentText().toInt();
    float avgMark = (mark1 + mark2 + mark3 + mark4 + mark5) / 5.0;

```

```

    ui->tableWidget->setItem(ui->tableWidget->rowCount() - 1, 7, new
QTableWidgetItem(QString::number(avgMark)));
    m_save_checker = false;
    //????????????????
    //QMessageBox::information(this, "SList", "The student has been added
sucessfully!");
}

void MainWindow::delete_selected() {
    QMessageBox msg;
    msg.setWindowIcon(this->windowIcon());
    msg.setText("Are you sure you want to remove selected row?");
    msg.setIcon(QMessageBox::Question);
    QPushButton *btn_yes = msg.addButton(QMessageBox::Yes);
    QPushButton *btn_no = msg.addButton(QMessageBox::No);
    msg.exec();
    if(msg.clickedButton() == btn_yes){
        ui->tableWidget->removeRow(ui->tableWidget->currentRow());
        for(int i = 0; i < ui->tableWidget->rowCount(); ++i)
            ui->tableWidget->setVerticalHeaderItem(i, new
QTableWidgetItem(QString::number(i + 1)));
        m_save_checker = false;
    }
    else if(msg.clickedButton() == btn_no)
        return;
}

void MainWindow::action_clear() {
    QMessageBox msg;
    msg.setWindowIcon(this->windowIcon());
    msg.setIcon(QMessageBox::Question);
    msg.setText("Are you sure you want to permanently clear the view?");
    msg.setInformativeText("It won't be saved.");
    QPushButton *btn_yes = msg.addButton(QMessageBox::Yes);
    QPushButton *btn_no = msg.addButton(QMessageBox::No);
    msg.exec();
    if(msg.clickedButton() == btn_yes){
        ui->tableWidget->clearContents();
        ui->tableWidget->setRowCount(0);
        m_save_checker = true;
    }
    else if(msg.clickedButton() == btn_no)
        return;
}

void MainWindow::new_view() {
    if(ui->tableWidget->rowCount() != 0){
        QMessageBox msg;
        msg.setWindowIcon(this->windowIcon());
        msg.setIcon(QMessageBox::Question);
        msg.setText("Are you sure you want to permanently clear this view and
create a new one?");
        msg.setInformativeText("Current view won't be saved.");
        QPushButton *btn_yes = msg.addButton(QMessageBox::Yes);
        QPushButton *btn_no = msg.addButton(QMessageBox::No);
        msg.exec();
        if(msg.clickedButton() == btn_yes) {
            ui->tableWidget->clearContents();
            ui->tableWidget->setRowCount(0);

```

```

    }
    else if(msg.clickedButton() == btn_no) return;
}

void MainWindow::save_as(){
    if(ui->tableWidget->rowCount() == 0){
        QMessageBox::information(this,"SList","The view is clear.");
        return;
    }

    QString filename = QFileDialog::getSaveFileName(nullptr, "test", ".", "Text
files (*.txt)");
    m_filename = filename;
    QFile file(filename);
    if (file.open(QFile::WriteOnly | QFile::Truncate))
    {
        QTextStream data( &file );
        QStringList strList;
        for( int r = 0; r < ui->tableWidget->rowCount(); ++r ){
            strList.clear();
            for( int c = 0; c < ui->tableWidget->columnCount(); ++c ){
                QTableWidgetItem* item = ui->tableWidget->item(r,c); //Load
items
                if (!item || item->text().isEmpty()){ //Test
if there is something at item(r,c)
                    ui->tableWidget->setItem(r,c,new QTableWidgetItem("0")); //IF
there is nothing write 0
                }
                strList << ui->tableWidget->item( r, c )->text()+" ";
            }
            data << strList.join( "; " )+"\n";
        }
        file.close();
        m_save_checker = true;
        QDesktopServices::openUrl(QString(filename)

        .arg(QCoreApplication::applicationDirPath()));
    }
}

void MainWindow::save(){
    if(ui->tableWidget->rowCount() == 0){
        QMessageBox::information(this,"SList","The view is clear.");
        return;
    }
    QString filename = m_filename;
    QFile file(filename);
    if (file.open(QFile::WriteOnly | QFile::Truncate))
    {
        QTextStream data( &file );
        QStringList strList;
        for( int r = 0; r < ui->tableWidget->rowCount(); ++r ){
            strList.clear();
            for( int c = 0; c < ui->tableWidget->columnCount(); ++c ){
                QTableWidgetItem* item = ui->tableWidget->item(r,c); //Load
items
                if (!item || item->text().isEmpty()){ //Test
if there is something at item(r,c)

```

```

        ui->tableWidget->setItem(r,c,new QTableWidgetItem("0")); //IF
there is nothing write 0
    }
    strList << ui->tableWidget->item( r, c )->text()+" ";
}
data << strList.join( "; " )+"\n";
}
file.close();
m_save_checker = true;
QDesktopServices::openUrl(QString(filename)

.arg(QCoreApplication::applicationDirPath()));
}
}

void MainWindow::closeEvent(QCloseEvent *event){
    event->ignore();
    if(m_save_checker == false){
        QMessageBox msg;
        msg.setWindowIcon(this->windowIcon());
        msg.setIcon(QMessageBox::Warning);
        msg.setText("Exit without saving current view?");
        //msg.setInformativeText("Current view won't be saved.");
        QPushButton *btn_yes = msg.addButton(QMessageBox::Yes);
        QPushButton *btn_no = msg.addButton(QMessageBox::No);
        QPushButton *btn_save = msg.addButton(QMessageBox::Save);
        msg.exec();
        if(msg.clickedButton() == btn_yes) event->accept();
        else if(msg.clickedButton() == btn_no) return;
        else if(msg.clickedButton() == btn_save){
            ui->actionSave->trigger();
            event->accept();
        }
    }
    else if (QMessageBox::Yes == QMessageBox::question(this, "Close Confirmation",
"Exit?", QMessageBox::Yes | QMessageBox::No))
        event->accept();
}

```

Main.cpp

```

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    MainWindow w;
    w.show();
    return a.exec();
}

```

## Висновки

Під час виконання лабораторної роботи я навчилася розробляти ескізи інтерфейсу користувача, створювати інструкцію користувача та імплементувати розробку GUI для нової версії програми.