

Министерство цифрового развития, связи и массовых коммуникаций РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Сибирский государственный университет телекоммуникаций и информатики»
(СибГУТИ)

КУРСОВАЯ РАБОТА

Вариант 21

ТЕМА РАБОТЫ:

Администратор фитнес клуба

Выполнил студент группы ИП-112 Тихонов Н.А.

Проверил доцент кафедры ПМиК к.т.н Мерзлякова Е.Ю.

Новосибирск 2023

СОДЕРЖАНИЕ

СОДЕРЖАНИЕ	2
Введение	3
Анализ задач и пользователей.....	4
Выбор репрезентативных задач	5
Окна приложения.....	11
База данных.....	15
Анализ приложения	16
Список используемых источников.....	20
Приложение.....	21

Введение

Администратор фитнес клуба представляет собой приложение , в котором показана вся нужная информация для администратора фитнес клуба: база данных людей, купивших абонементы, цены на собственную продукцию и др.

Данная тема была выбрана мною в качестве курсовой работы с целью развития в области разработки UI-приложений, которые могут использоваться повсеместно в обычной жизни.

Очень много схожих приложений по типу администратор (Например администратор какого-нибудь магазина).

Анализ задач и пользователей

Задачи:

Необходимо разработать приложение, в котором у пользователя будет информация о покупателях, расписании тренировок, ценах в магазине.

Информация о покупателях должна сопровождаться именем, Фамилией, возрастом, а также ComboBox`ом, который показывает вид абонемента.

Пользователи:

Друг: возраст – 20 лет, студент, навыки владения компьютером – высокие

Родитель семьи: возраст – 37 лет, образование – высшее, навыки владения компьютером – средние

Пользователи программного продукта являются взрослыми людьми в возрасте от 18 до 50 лет. Навык владения компьютером обычно на уровне базового/среднего.

Выбор репрезентативных задач

- Добавление, удаление и редактирование покупателя
- Просмотр всех покупателей
- Просмотр магазина фитнес клуба
- Просмотр расписания тренировок

Прототип интерфейса

The image is a wireframe of a web application interface. It features a large central rectangular area for displaying a list of items, with the text "Покупатели и информация о них" (Buyers and information about them) at the top left. To the right of this area is a vertical column of three buttons: "Добавление" (Add), "Изменение" (Edit), and "Удаление" (Delete). Above these buttons is a small box labeled "Время" (Time). At the bottom of the interface, there are two horizontal boxes: "Немного про фитнес клуб" (A little about the fitness club) on the left and "Расписание" (Schedule) on the right.

Рис.1 Главное окно

Имя	Поле ввода имени
Фамилия	Поле введение фамилии
возраст	поле возраста
абонемент	выбор абонемента
сохранить	
Отменить	

Рис.2 Окно Добавления и редактирования покупателя

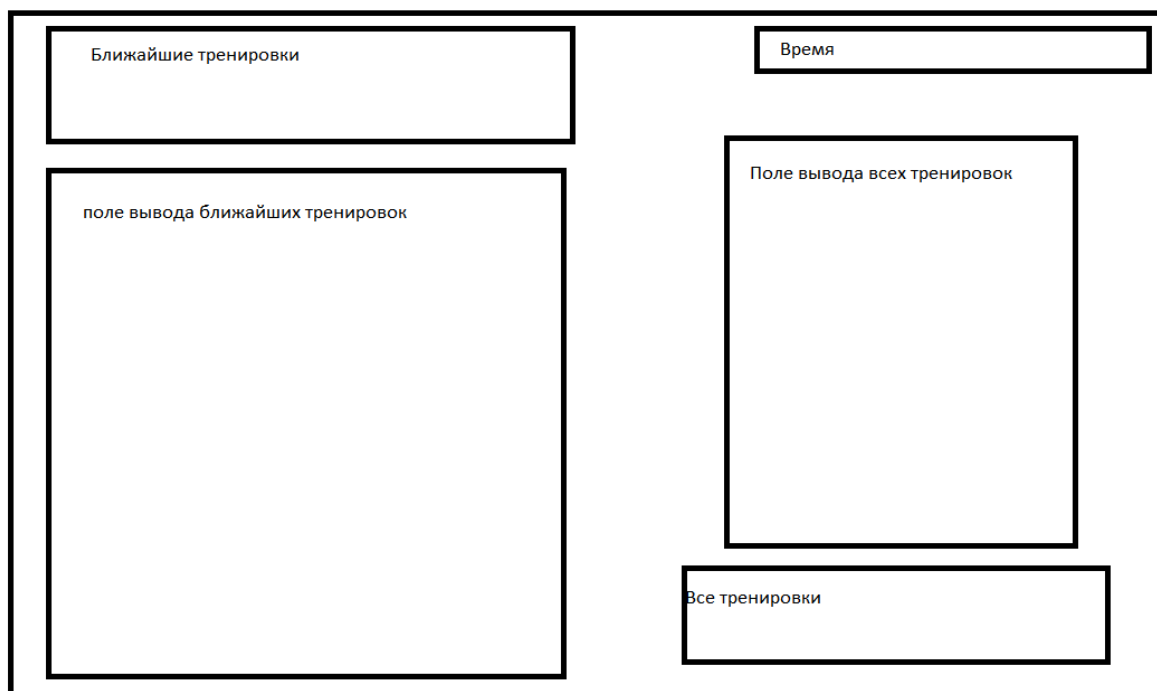


Рис.3 Окно просмотра расписания тренировок

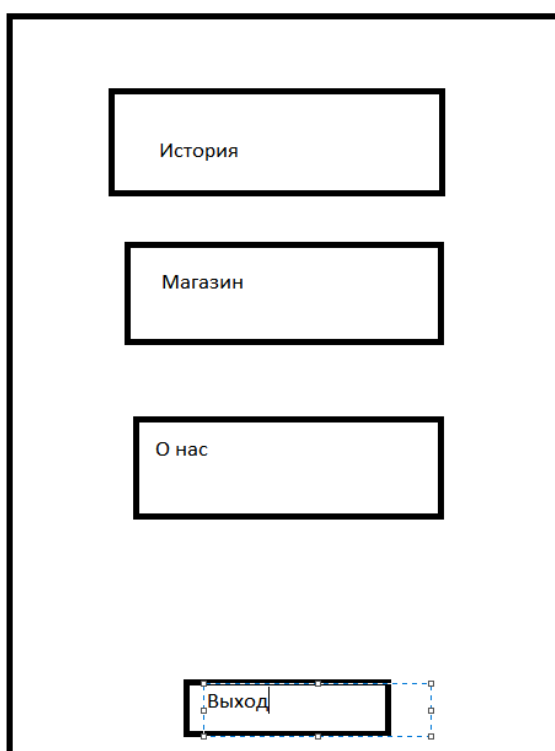


Рис.4 Окно навигации по фитнес клубу

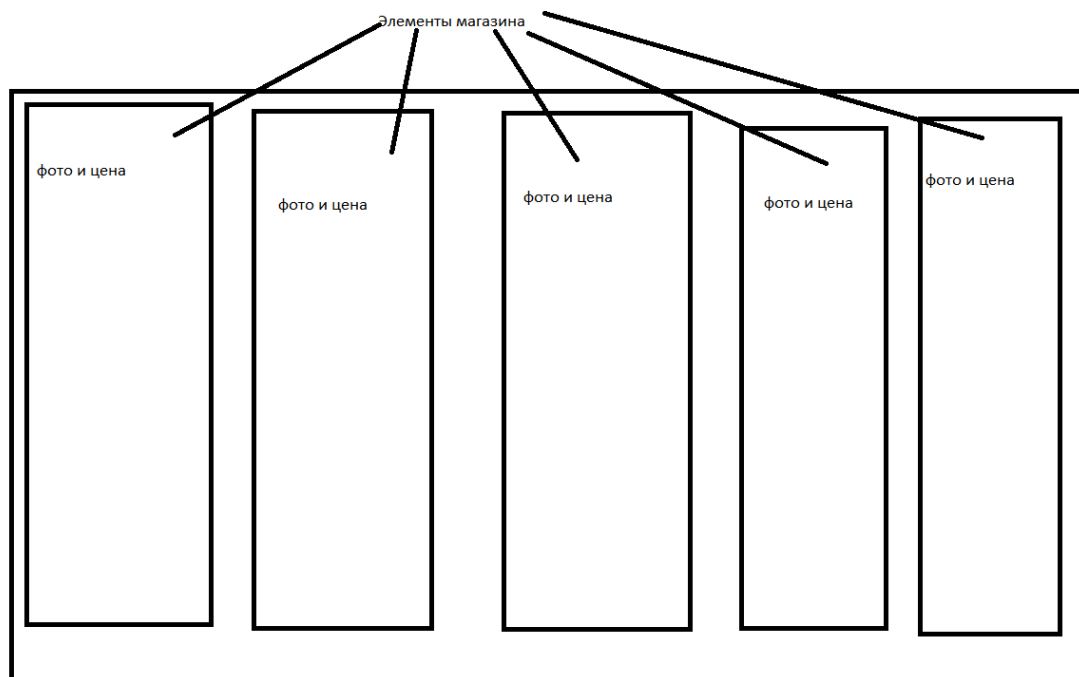


Рис.5 Окно просмотра магазина

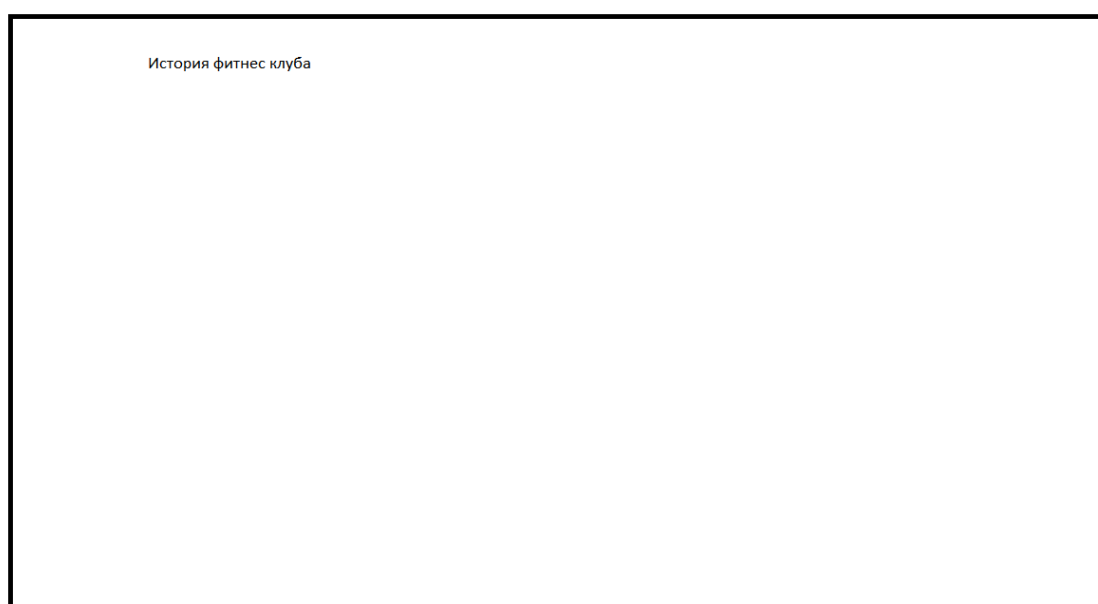


Рис.6 История фитнес клуба

Реализация

Функционал:

1) Сохранение данных покупателя

- 2) Редактирование данных покупателя
- 3) Удаление покупателя
- 4) Просмотр всех покупателей
- 5) Просмотр магазина
- 6) Просмотр Расписания тренировок
- 7) Просмотр окна с ссылками

Классы:

- 1) database – класс базы данных(создание, соединение и др.)
- 2)price – класс просмотра магазина
- 3) history– класс просмотра истории о фитнес клубе
- 4) about– класс просмотра ссылок
- 5) addabonement– класс создания нового покупателя
- 6) editabonement– класс редактирования покупателя
- 7)deleteabonement – класс удаления покупателя
- 8)schedule – класс просмотра расписания тренировок

Окна приложения

Приложение состоит из окон:

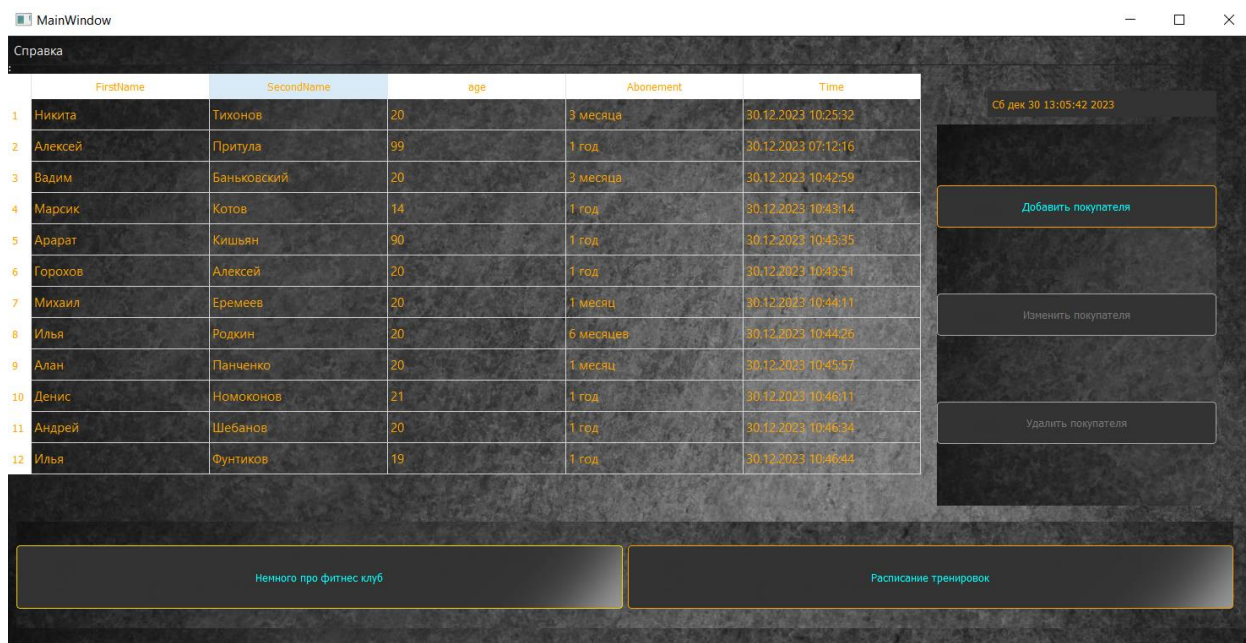


Рис.7 Главное окно

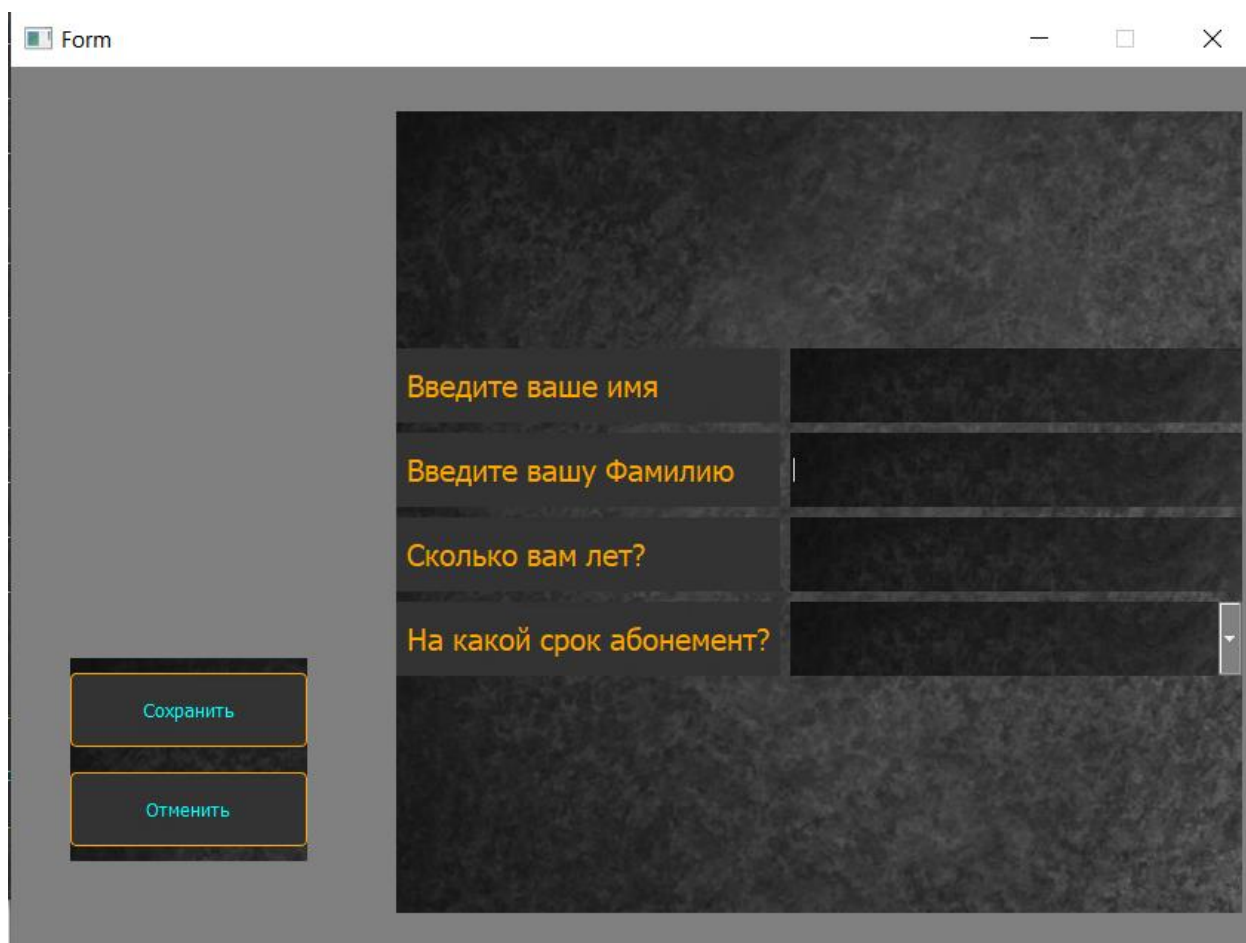


Рис.8 Окно добавления покупателя

Form

Введите ваше имя: Илья

Введите вашу Фамилию: Родкин

Сколько вам лет?: 20

На какой срок абонемент?: 6 месяцев

Сохранить

Отменить

Рис.9 Окно редактирования покупателя

Form

Дата и время: Сб дек 30 13:07:00 2023

Ближайшие тренировки

Тренер: kfajsfjasl;fk
Тренировка: Растяжка
Время начала тренировки: 14 : 00

Тренер: Tikhonov Nikita
Тренировка: Тренировка на Выносливость
Время начала тренировки: 15 : 30

Тренер: Pritula Aleksei
Тренировка: Медитация
Время начала тренировки: 17 : 00

Тренер: Motivator Atnon
Тренировка: Мотивация
Время начала тренировки: 18 : 30

Тренер: Uncle Vasia
Тренировка: Силовая тренировка
Время начала тренировки: 20 : 00

Тренер: kfajsfjasl;fk
Тренировка: Растяжка
Время начала тренировки: 21 : 30

Посмотреть все тренировки

Назад на главную страницу

Рис.10 Окно просмотра расписания тренировок

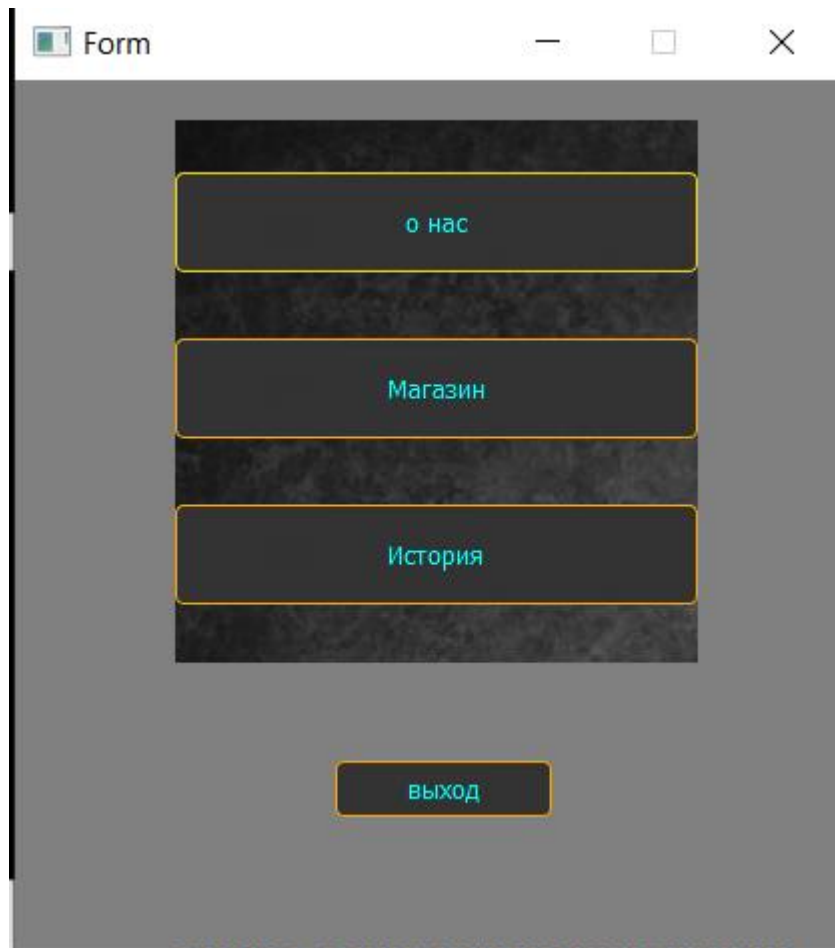


Рис.11 Окно навигатора по фитнес клубу



Рис.12 Окно ссылок

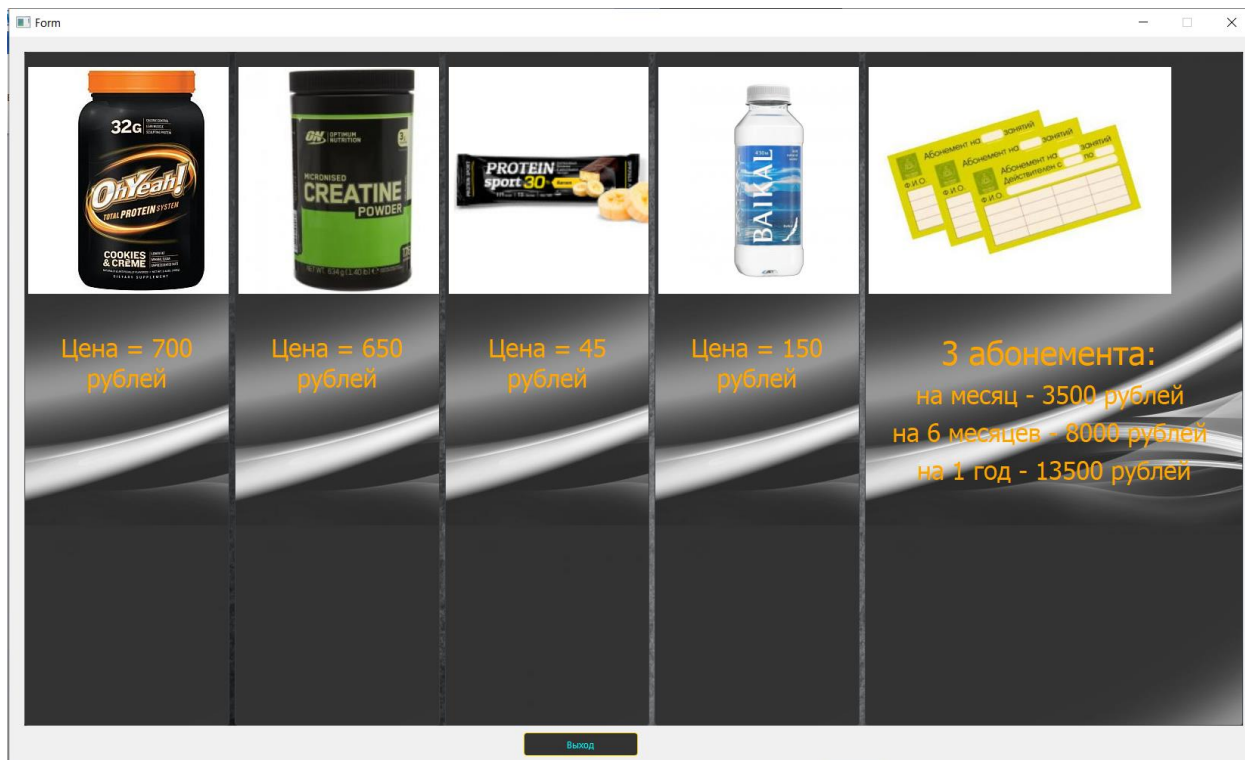


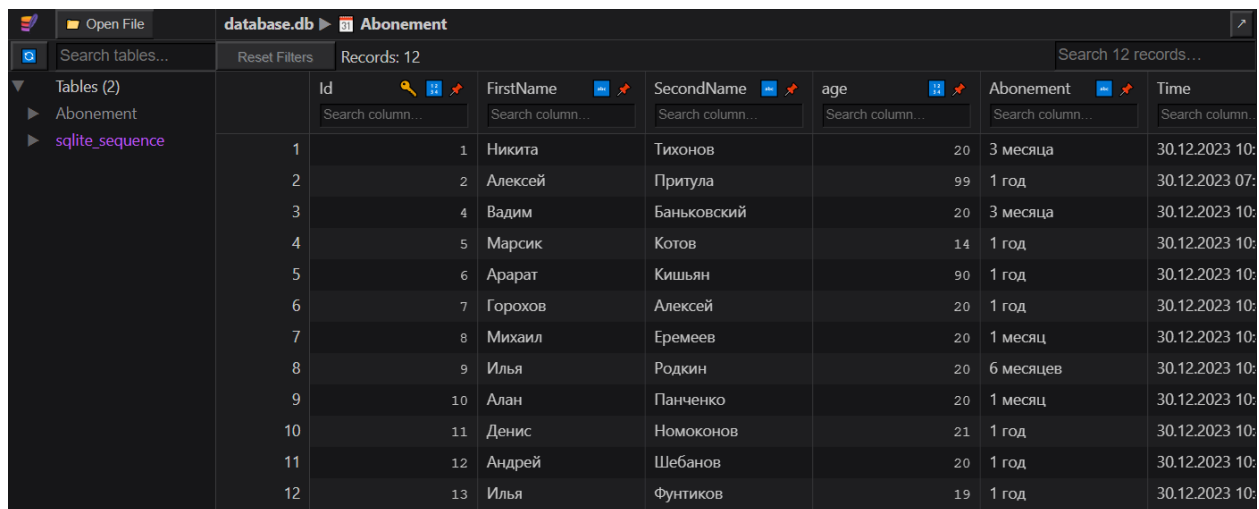
Рис.13 Окно магазина



Рис.12 Окно истории фитнес клуба

База данных

Описание базы данных:



The screenshot shows a database viewer interface with a dark theme. The top bar indicates the database is 'database.db' and the table is 'Abonement'. Below the top bar, there are search and filter options. The main area displays a table with 8 columns: Id, FirstName, SecondName, age, Abonement, and Time. The table contains 12 records. The 'Id' column is highlighted in blue. The 'Abonement' column shows various durations like '3 месяца', '1 год', and '6 месяцев'. The 'Time' column shows timestamps like '30.12.2023 10:00'.

Id	FirstName	SecondName	age	Abonement	Time
1	Никита	Тихонов	20	3 месяца	30.12.2023 10:00
2	Алексей	Притула	99	1 год	30.12.2023 07:00
3	Вадим	Баньковский	20	3 месяца	30.12.2023 10:00
4	Марсик	Котов	14	1 год	30.12.2023 10:00
5	Арагат	Кишьян	90	1 год	30.12.2023 10:00
6	Горохов	Алексей	20	1 год	30.12.2023 10:00
7	Михаил	Еремеев	20	1 месяц	30.12.2023 10:00
8	Илья	Родкин	20	6 месяцев	30.12.2023 10:00
9	Алан	Панченко	20	1 месяц	30.12.2023 10:00
10	Денис	Номоконов	21	1 год	30.12.2023 10:00
11	Андрей	Шебанов	20	1 год	30.12.2023 10:00
12	Илья	Фунтиков	19	1 год	30.12.2023 10:00

Рис.13 Таблица abonement

В базе данных хранятся имя (FirstName), фамилия (SecondName), возраст(Age), вид абонемента(Abonement) покупателя, а также уникальный идентификатор(id) и время создания покупателя(time), которые генерируются автоматически

Анализ приложения

CWT-анализ

Задача 1. Просмотр магазина

№	Действие	История	Исправление
1	Запуск приложения	Приложение успешно запустилось	
2	Поиск кнопки с переходом в магазин	кнопка не найдена	Добавить Кнопку Магазина на начальное окно
3	Нажатие на кнопку Магазин	После нажатия на кнопку , открывается окно с магазином фитнес клуба	

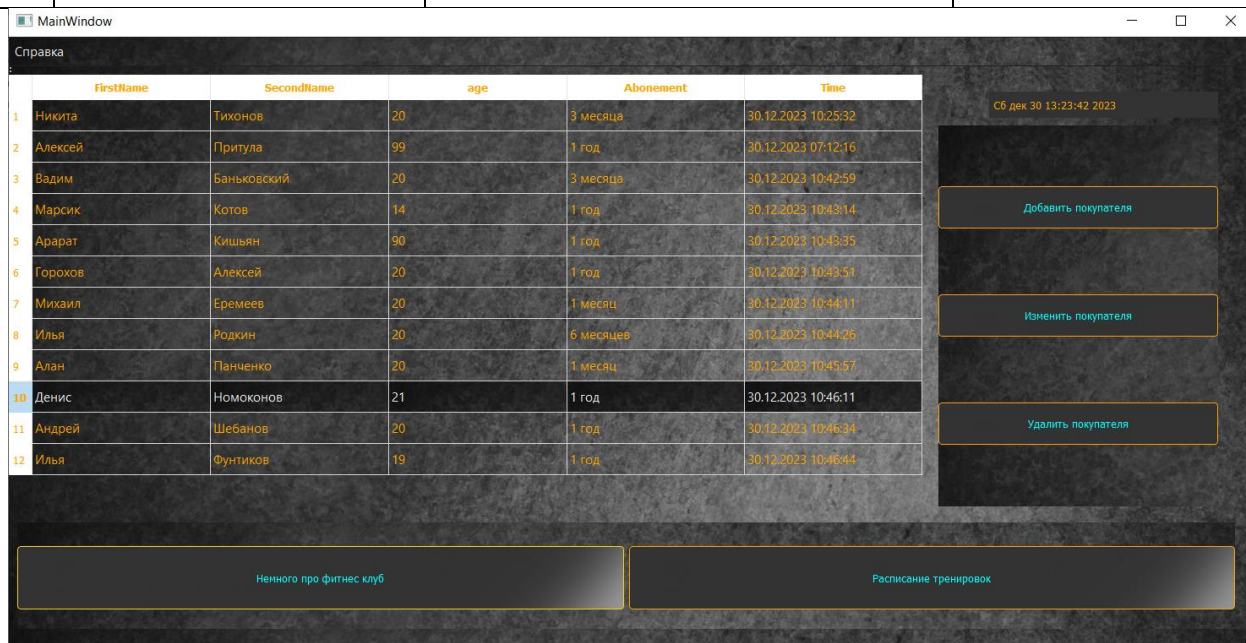


Рис 14 показывает, что нету кнопки магазина на начальном окне

Задача 2. Удаление записи(будем думать, что база данных пуста)

№	Действие	История	Исправление
1	Запуск приложения	Приложение успешно запустилось	
2	Поиск кнопки с удалением элемента	кнопка найдена , но она не активна	Сделать подсказку, что база данных пуста и нужно добавить покупателя

3	Поиск кнопки с добавлением покупателя	Кнопка успешно найдена	
4	Нажатие на кнопку с добавлением покупателя	После нажатия на кнопку открывается окно , в котором можно добавить покупателя	
5	Заполнение пустых ячеек	Пользователь вводит имя, фамилию, возраст, вид абонента в пустых ячейках	
6	Поиск кнопки с сохранением данных	Кнопка легко найдена	
7	Нажатие на кнопку сохранения	Появляется окошко , что сохранение прошло успешно, после чего окно с добавлением покупателя закрывается и открывается главное окно	
8	Нажатие на кнопку удаления	После нажатия элемент удаляется	Нужно добавить дополнительное окно для перепроверки действий (вдруг случайно не того покупателя хочет удалить)

A screenshot of a software window titled "Form". The window has a dark gray background. On the right side, there are four input fields stacked vertically. Each field has a yellow text prompt: "Введите ваше имя", "Введите вашу Фамилию", "Сколько вам лет?", and "На какой срок абонемент?". To the left of these fields, there are two buttons stacked vertically: "Сохранить" (Save) and "Отменить" (Cancel). The window has standard Windows-style window controls (minimize, maximize, close) in the top right corner.

Рис 15 показывает окно добавления

GOMS-анализ

Описание анализа

Практически все интерфейсные взаимодействия можно описать следующими операциями:

К – нажатие клавиши;

В – клик кнопкой мыши;

Р – наведение указателя мыши;

Р – ожидание ответной реакции компьютера;

Н – перенос руки с клавиатуры на мышь или наоборот

Д – проведение с помощью мыши прямой линии (например, выделение или прокрутка текста);

М – мыслительная подготовка (к осуществлению одной из перечисленных операций).

Разные пользователи выполняют указанные операции за разное время.

Однако, GOMS исследует работу опытного пользователя. Многочисленные исследования выявили средние значения времени операций, выполняемых опытными пользователями.

К 0.2 с

В 0.2 с

Р 1.1 с

Н 0.4 с

М 1.35 с

Цель 1 – Посмотреть все тренировки:

1) Нажать на кнопку “Расписание тренировок”

- Р – наведение указателя мыши на поле логина
- В – нажатие на это поле

Пояснение: Мы перешли в окно с тренировками

2) Нажать на поле “Посмотреть все тренировки”

- Р – наведение указателя мыши на поле логина
- В – нажатие на это поле

Итого: $2 \cdot Р + 2 \cdot В = 2.2 + 0.4 = 2.6 \text{ с.}$

Данный пример показывает, что программа оптимизирована под это действие.

Цель 2 – Добавление нового пользователя

(Никита Тихонов 20 1 год абонемент)

1) Нажать на кнопку Добавления покупателя

- Р – наведение указателя мыши на поле логина
- В – нажатие на это поле

2) Нажатие на поле Введите ваше имя и ввод имени Никита(РНКККККК)

- Р – наведение указателя мыши на поле введите ваше имя
- В- нажатие на поле Ваше имя
- Н – перенос руки с клавиатуры на мышь
- 6К – ввод имени Никита(КККККК)

- 3) Нажатие на поле Введите вашу фамилию и ввод фамилии Тихонов(НРВНKKKKKKK)
- Н – перенос руки с клавиатуры на мышь
 - Р – наведение указателя мыши на кнопку изменения пароля
 - В - нажатие на это поле
 - Н – перенос руки с мыши на клавиатуру
 - 7К – Нажатие клавиши для ввода символа нового пароля.
- 4) Нажатие на поле возраст и ввод 20(НРВНKK)
- Н-перенос руки с клавиатуры на мышь
 - Р-наведение указателя мыши на поле возраста
 - В – нажатие на это поле
 - Н – перенос руки с мыши на клавиатуру
 - 2К – ввод возраста 20
- 5) Нажатие на поле абонемент и выбор нужного абонемент и сохранение (НРВРВРВ)
- Н – перенос руки с клавиатуры на мышь
 - Р-наведение указателя мыши на поле абонемент
 - В – нажатие на поле
 - Р-наведение указателя мыши на
 - В – выбор нужного абонемент
 - Р – наведение указателя мыши на кнопку сохранить
 - В – нажатие на кнопку

Итого: $7*Р+7*В+6*Н+19*К = 7.7+1.4+2.4+3.8=15.3с.$

Список используемых источников

- 1)<http://qt-doc.ru/potoki-vvoda-vyvoda-klass-qtextstream.html>
- 2) Лекции по визуальному программированию
- 3)<https://doc.qt.io/qt-6/qfile.html>

Приложение

```
#include "mainwindow.h"
#include <QApplication>
#include <QApplication>
#include <QFile>
#include <QTextStream>
#include <QTranslator>
#include <QLibraryInfo>
#include <QSplashScreen>
#include <QTime>
#include <QPainter>

void LoadModules(QSplashScreen* psplash) {
    QTime time;
    time.start();

    for (int i = 0; i < 100; ) {
        if (time.elapsed() > 40) {
            time.start();
            ++i;
        }

        // Отображение сообщения о загрузке на экране SplashScreen
        psplash->showMessage("Загрузка модулей: " + QString::number(i) + "%",
Qt::AlignRight | Qt::AlignTop, Qt::gray);
        QApplication->processEvents();
    }
}

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

    QFile f(":/style/style.qss");
    if (!f.exists()) {
        // Вывод сообщения об ошибке, если файл стилей не найден
        printf("Невозможно установить таблицу стилей, файл не найден\n");
    } else {
        f.open(QFile::ReadOnly | QFile::Text);
        QTextStream ts(&f);
        QApplication->setStyleSheet(ts.readAll());
    }

    QTranslator qtTranslator;
    qtTranslator.load("qt_ru",
QLibraryInfo::location(QLibraryInfo::TranslationsPath));
    a.installTranslator(&qtTranslator);

    MainWindow w;

    QSplashScreen splash(QPixmap(":/images/load.jpg"));
    splash.show();

    LoadModules(&splash);

    // Закрытие SplashScreen и открытие основного окна
    splash.finish(&w);
}
```

```

        w.show();

        return a.exec();
    }

#include "mainwindow.h"
#include "ui_mainwindow.h"

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

    secondWindow = new Schedule();
    connect(secondWindow, &Schedule::firstWindow, this, &MainWindow::show);

    thirdWindow = new Fitness();
    connect(thirdWindow, &Fitness::firstWindow, this, &MainWindow::show);

    settings = new QSettings("geometry", "SettingsWindow", this);
    LoadSettings();

    QTimer *timer = new QTimer(this);
    connect(timer, SIGNAL(timeout()), this, SLOT>ShowTime());
    timer->start();

    db = new database();
    db->connectToDataBase();

    model = new QSqlTableModel;
    model->setTable("abonement");
    model->setHeaderData(0, Qt::Horizontal, "id");

    ui->tableView->setModel(model);
    ui->tableView->setColumnHidden(0, true);
    ui->tableView->setSelectionBehavior(QAbstractItemView::SelectRows);
    ui->tableView->setSelectionMode(QAbstractItemView::SingleSelection);
    ui->edit_button->setEnabled(true);
    ui->delete_button->setEnabled(true);

    ui->tableView->horizontalHeader()->setSectionResizeMode( 1,
    QHeaderView::Stretch );
    ui->tableView->horizontalHeader()->setSectionResizeMode( 2,
    QHeaderView::Stretch );
    ui->tableView->horizontalHeader()->setSectionResizeMode( 3,
    QHeaderView::Stretch );
    ui->tableView->horizontalHeader()->setSectionResizeMode( 4,
    QHeaderView::Stretch );
    ui->tableView->horizontalHeader()->setSectionResizeMode( 5,
    QHeaderView::Stretch );
    updateTable();

    editForm = new editabonement();
    editForm->setWindowModality(Qt::ApplicationModal);
    addForm = new addabonement();
    addForm->setWindowModality(Qt::ApplicationModal);

```

```

        connect(addForm, SIGNAL(sendToWidget(QString, QString, QString, QString,
QString)),
                this, SLOT(addRecordDataBase(QString, QString, QString, QString,
QString))));
        connect(this, SIGNAL(sendForEdit(int, QString, QString, QString,
QString, QString)),
                editForm, SLOT(catchInfo(int, QString, QString, QString, QString,
QString))));
        connect(editForm, SIGNAL(sendToWidgetUpdate(int, QString, QString,
QString, QString, QString)),
                this, SLOT(editRecordDataBase(int, QString, QString, QString,
QString, QString))));

        connect(ui->add_button, SIGNAL(clicked()), this, SLOT(slotAdd()));
        connect(ui->edit_button, SIGNAL(clicked()), this, SLOT(slotEdit()));
        connect(ui->delete_button, SIGNAL(clicked()), this, SLOT(slotDel()));
        connect(ui->action, SIGNAL(triggered()), this, SLOT(slotInfo()));
    }

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

void MainWindow::updateTable()
{
    currID = -1;
    model->select();
    model->setSort(5, Qt::AscendingOrder);

    if (currRow == -1) {
        ui->edit_button->setEnabled(false);
        ui->delete_button->setEnabled(false);
    } else {
        ui->edit_button->setEnabled(true);
        ui->delete_button->setEnabled(true);
    }

    ui->tableView->show();
}

void MainWindow::slotAdd()
{
    addForm->show();
}

void MainWindow::addRecordDataBase(const QString &fname, const QString
&sname, const QString &age ,const QString &abonement, const QString &time)
{
    db->insertIntoTable(fname, sname, age, abonement, time);
    updateTable();
    currRow = -1;
    currID = -1;
}

void MainWindow::slotEdit()

```

```

{
    if (currRow != -1) {
        QString name = model->data(model->index(currRow, 1)).toString();
        QString sname = model->data(model->index(currRow, 2)).toString();
        QString age = model->data(model->index(currRow, 3)).toString();
        QString abonement = model->data(model->index(currRow, 4)).toString();
        QString time = model->data(model->index(currRow, 5)).toString();
        emit sendForEdit(currID, name, sname, age, abonement, time);
        editForm->show();
    }
}

void MainWindow::editRecordDataBase(const int &id, const QString &fname, const
QString &sname, const QString &age, const QString &abonement, const QString
&time)
{
    currRow = -1;
    currID = -1;
    db->editInTable(id, fname, sname, age, abonement, time);
    updateTable();
}

void MainWindow::slotDel()
{
    if (currID != -1){
        db->removeRecord(currID);
        updateTable();
        QMessageBox::information(nullptr, "Уведомление", "Запись успешно
удалена");
        currRow = -1;
        currID = -1;
    }
}

void MainWindow::on_tableView_clicked(const QModelIndex &index)
{
    {
        ui->edit_button->setEnabled(true);
        ui->delete_button->setEnabled(true);
        currID = model->data(model->index(index.row(), 0)).toInt();
        currRow = index.row();
    }
}

void MainWindow::on_pushButton_3_clicked()
{
    secondWindow->show();
    this->close();
}

void MainWindow::ShowTime()
{
    QDateTime dateTime = QDateTime::currentDateTime();
    QString datetime_text = dateTime.toString();
    ui->label_5->setText(datetime_text);
}

void MainWindow::SlotInfo()
{
    helpinformation* form = new helpinformation();
    form->setWindowModality(Qt::ApplicationModal);
}

```



```
        form->show();
    }

    void MainWindow::on_pushButton_2_clicked()
    {
        thirdWindow->show();
        this->close();
    }

    void MainWindow::SaveSettings()
    {
        settings->setValue("geometry", geometry());
    }

    void MainWindow::LoadSettings()
    {
        setGeometry(settings->value("geometry", QRect(200, 200, 300,
300)).toRect());
    }
}
```