Частное учреждение образования

«Колледж бизнеса и права»

Программная реализация программного средства для онлайн-сервиса заказа услуг «Массаж на дом»

Пояснительная записка

к курсовому проекту по дисциплине

«Основы алгоритмизации и программирование»

КП Т.192015.401

Руководитель проекта (Дж. А. Миронова)

Учащийся (А. А. Сидоренко)

2023

Изм.

Лист

№ документа

Подпись

Дата

Лист

3

КП Т.192015.401

Разработал

Сидоренко А. А.

Проверили

Миронова Дж. А.

Реценз.

Н. Контр.

Утверд.

*Программное средство для онлайн-сервиса заказа услуг «Массаж на дом»*

Лит.

Листов

КБиП

у

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc6988)

[1 Описание задачи 5](#_Toc864)

[1.1 Анализы предметной области 5](#_Toc24095)

[1.2 Постановка задачи 5](#_Toc23325)

[2 Проектирование системы 7](#_Toc10718)

[2.1 Требование к оконному приложению 7](#_Toc25630)

[2.2 Проектирование модели 7](#_Toc6856)

[2.3 Концептуальный прототип 8](#_Toc31671)

[3 Описание реализации программного средства 13](#_Toc27283)

[3.1. Инструменты разработки и применяемые технологии 13](#_Toc18359)

[3.2. Порядок авторизации пользователей 13](#_Toc22291)

[3.3 Функции: логическая и физическая организация 14](#_Toc2264)

[3.4. Входные и выходные данные 17](#_Toc5996)

[3.5 Функциональное тестирование 18](#_Toc30137)

[3.6 Описание справочной системы 20](#_Toc7073)

[4 Применение 22](#_Toc2732)

[4.1 Назначение программного средства 22](#_Toc29109)

[4.2 Условия применения 22](#_Toc23764)

[Заключение 22](#_Toc17593)

[Список использованных источников 24](#_Toc13167)

[Приложение А (обязательное) Текст программных модулей 25](#_Toc19847)

# 

# **Введение**

В современном мире люди все больше и больше стремятся к удобству и доступности услуг прямо из комфорта своего дома. Онлайн-сервисы стали незаменимым инструментом для заказа различных услуг, и одной из таких услуг является массаж на дом. «Массаж на дом» предоставляет возможность людям получать профессиональные массажные услуги в уютной обстановке своего дома, без необходимости посещать спа-салоны или клиники.

Целью данного курсового проекта является программная реализация программного средства онлайн-сервиса для заказа услуг «Массаж на дом». Создание такого сервиса позволит клиентам удобно и легко заказывать массажиста прямо через веб-приложение, выбирать необходимую услугу, указывать предпочтительное время и место проведения массажа.

Для достижения данной цели были решены следующие задачи:

* выполнен анализ и проектирование системы;
* по модели приложения выполнено проектирование задачи;
* было сделано описание созданного программного средства;
* выбрана методика испытаний;
* описан весь процесс тестирования программного средства;
* приведены примеры мобильных приложений про отели, для области применения программы.

Актуальность приложения для онлайн-сервиса заказа услуг «Массаж на дом» состоит в том, что оно отвечает современным требованиям и потребностям людей, стремящихся к удобству и доступности услуг прямо из своего дома.

Пояснительная записка состоит из 4 разделов:

В разделе 1 «Описание задачи» описывается предметная область и задачи программного средства, которые должны быть решены.

В разделе 2 «Проектирование системы» включает в себя требования к программному средству для организации работы отеля «Массаж на дом»; проектирование модели программного средства; программные средства, необходимые для конкретной разработки программного средства.

В разделе 3 «Описание реализации программного средства» описываются инструменты разработки и применяемые технологии, функциональное тестирование; описание справочной системы приложения.

В разделе 4 «Применение» представлены все сведенья, функциональное назначение, описание разделов программного средства.

В заключении предоставлены все итоги при создании программного средства. По готовому программному средству, будет понятно соответствия требований поставленной задачи и выполненной работы в целом.

# **1 Описание задачи**

## Анализы предметной области

Целью данного курсового проекта является создание современного и интуитивно понятного программного средства, которое упростит процесс заказа услуг массажа на дом, предоставит информацию о доступных массажах и медицинских показаниях, а также позволит клиентам получать персонализированные рекомендации и предложения. Программное средство «Массаж на дом» будет представлять собой платформу, объединяющую все аспекты и взаимодействия. Рассмотрим основные аспекты приложения:

Заказ услуги массажа: Пользователь может выбрать тип массажа и указать предпочтительную дату для проведения массажа на дом. Он также сможет ознакомиться с информацией о доступных видах массажа и их медицинскими показаниями, чтобы выбрать наиболее подходящий массаж для себя.

Управление заказами: Клиент может просматривать и отменять свои заказы. Это позволит им иметь полный контроль над своими планами получения услуги массажа на дом и вносить изменения в дату проведения услуги массажа по своему усмотрению.

Информация о массажах: Приложение предоставит подробную информацию о каждом массаже, его медицинских показаниях. Это поможет клиенту сделать информированный выбор при заказе услуги.

Целевой аудиторией приложения являются люди, которые ценят удобство и комфорт получения массажа на дом. Это могут быть занятые профессионалы, люди, испытывающие физическую нагрузку и стресс, а также те, кто предпочитает индивидуальный подход и удобство заказа услуги на дом.

## Постановка задачи

При открытии программного средства «Массаж на дом» пользователя встречает приветственное окно с возможностью изменить цвет, язык, создать аккаунт, либо авторизоваться в аккаунте после того, как пользователь авторизовался, его ждёт окно с выбором различных функций. У пользователя есть возможность выбрать тип услуги, ознакомится с ею, узнать медицинские показания, так же есть кнопка «Оформить заказ», которая открывает окно, где пользователь может совершить оформление заказа на выбранный тип услуги, указав количество сеансов и дату проведения услуги массажа.

К аналогам приложения «Массаж на дом» относятся монополисты в этой сфере, а именно: «profi-bel», на рисунке 1.1 представлена главная страница сайта «profi-bel.by» [4].

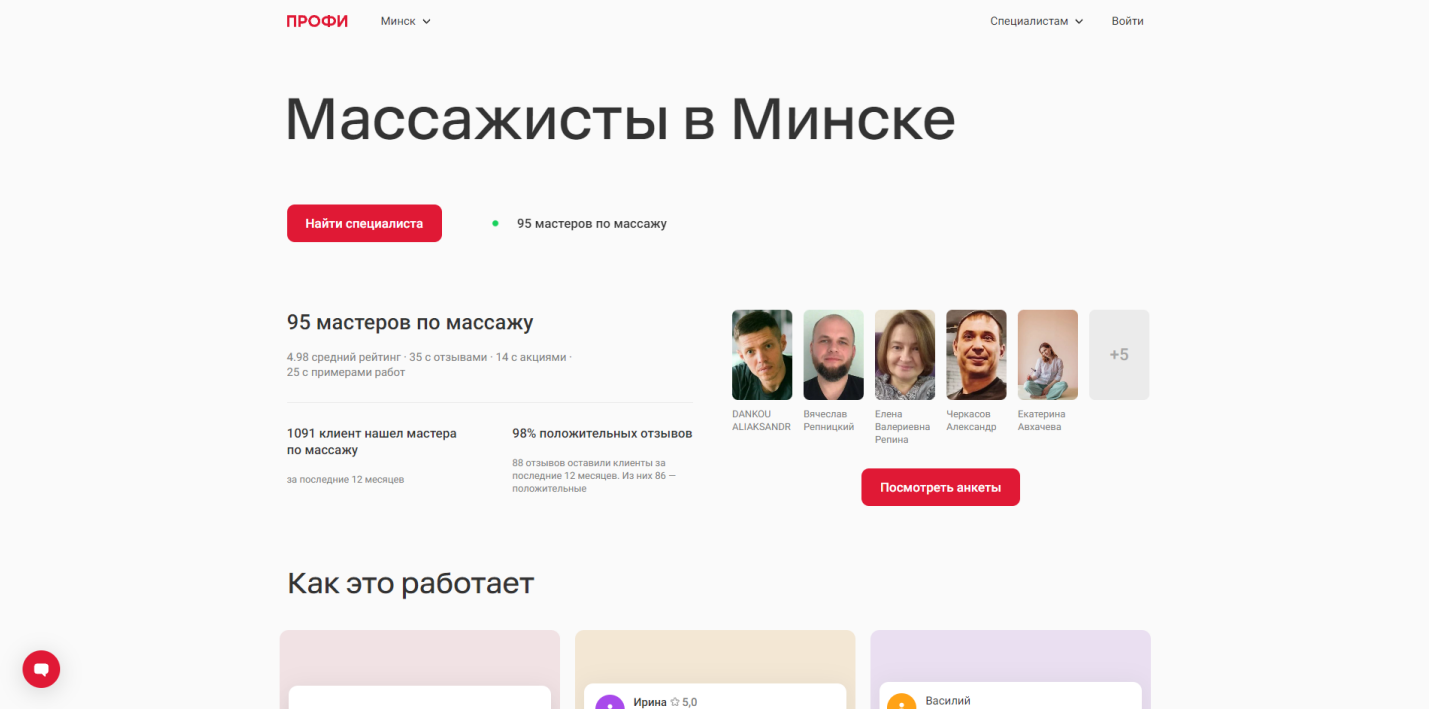


Рисунок 1.1 – Главная страница сайта «profi-bel.by»

Недостатками аналога является работа в онлайн режиме и невозможность работать без интернета, интуитивно непонятный интерфейс, отсутствие какого либо отношения к культуре или тематике массажа, не понятно сразу, что на сайте производится заказ услуг массажа на дом.

Преимуществом аналога выступают кроссплатформенность и работа с широкой базой пользователей и запросов. На сайте аналога также есть возможность предоставлять свои услуги массажа на дом другим пользователя.

# **2 Проектирование системы**

## 2.1 Требование к оконному приложению

Устройство для запуска разработанного программного средства должно соответствовать следующим минимальным требованиям:

* ОЗУ не менее 4 ГБ;
* ОС Windows 8 и выше.

Приложение должно иметь возможность менять языки, в данном случае «белорусский», «русский» и «английский». Приложение должно быть оформлено в красивых, не режущих глаза цветах. Главным требованием программного средства, является быстрая навигация пользователям с приложением, он не должен испытывать трудности в ознакомлении с нужной информации. Для этого были грамотно расставлены все кнопки, создан приятный интерфейс, чтобы он был наиболее понятным, а главное удобным для пользователя.

Дизайн несёт за собой успокаивающую тему, бамбук, который связан с культурной историей массажа. Все кнопки в программном средстве выполнены в лёгком округлённом стиле, чтобы пользователь быстро ориентировался, и чтобы он понимал их назначение.

Размер шрифта является чётким, для восприятия текста. Все элементы, представляющие текст, выглядят визуально приятно и читаемо.

В процессе анализа требований были сформированы основные функциональные требования программного средства.

Разрабатываемое программное средство предоставляет следующие функции:

* управление цветами приложения;
* возможность выбирать один из трёх языков;
* возможность пользователю пройти регистрацию;
* возможность пользователю выбрать нужный ему массаж и сделать заказ;
* возможность клиента просмотреть информацию о массаже;
* возможность админу посмотреть заказы и отредактировать их.

В результате пользователь получает удобный, а самое главное продуманный интерфейс, позволяющий легко ориентироваться в программном средстве.

## 2.2 Проектирование модели

Диаграмма работы алгоритма составления отчёта о выполненных заказах представляет собой поэтапное графическое отображение составление отчёта о выполненных заказах за выбранный промежуток времени. Диаграмма работы алгоритма составления отчёта выполненных заказах представлена в приложении Б.

Диаграмма работы программы помогает пользователю разобраться как работает программное средство и подробно описывает каждый шаг работы программного средства. Диаграмма работы программы представлена в приложении В.

Диаграмма классов позволяет пользователю получить подробное представление о структуре программного средства и о том, как различные классы взаимодействуют между собой, представленная в приложении Г.

## 2.3 Концептуальный прототип

Для разработки программного проекта одним из этапов является составление концептуального прототипа приложения. Концептуальный прототип приложения создаётся для представления идеи, концепции и основной функциональности приложения до его полноценной разработки. Он является ранней версией приложения, которая помогает лучше понять и визуализировать концепцию и потенциал приложения.

Концептуальный прототип начинается с начального окна, где пользователь может изменить язык и цветовую гамму программного средства, пройти авторизацию или регистрацию, прототип первого окна представлен на рисунке 2.1.

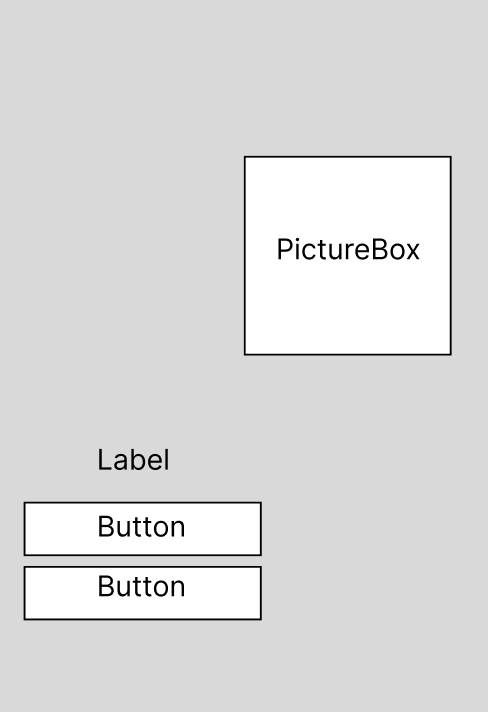


Рисунок 2.1 – Начальное окно

Графический макет для окна авторизации, представляет собой окно для входа в аккаунт. Окно авторизации представлено на рисунке 2.2.

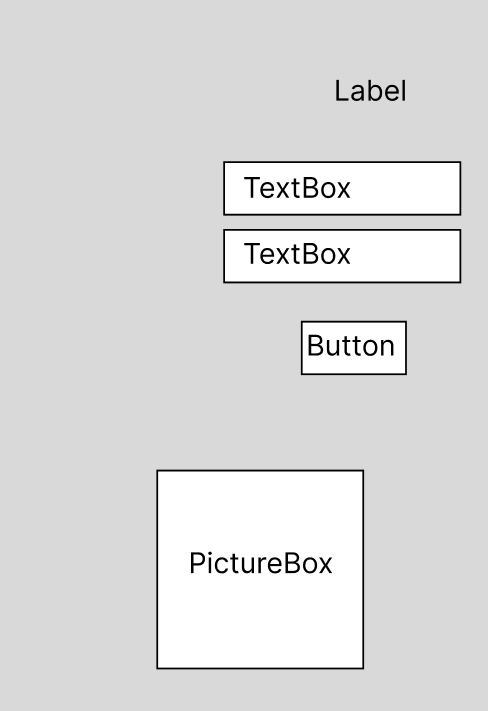


Рисунок 2.2 – Окно авторизации

Графический макет для окна регистрации содержит в себе 3 поля для заполнения, куда пользователю следует ввести свою электронную почту, пароль и мобильный номер телефона. Окно регистрации представлено на рисунке 2.3.

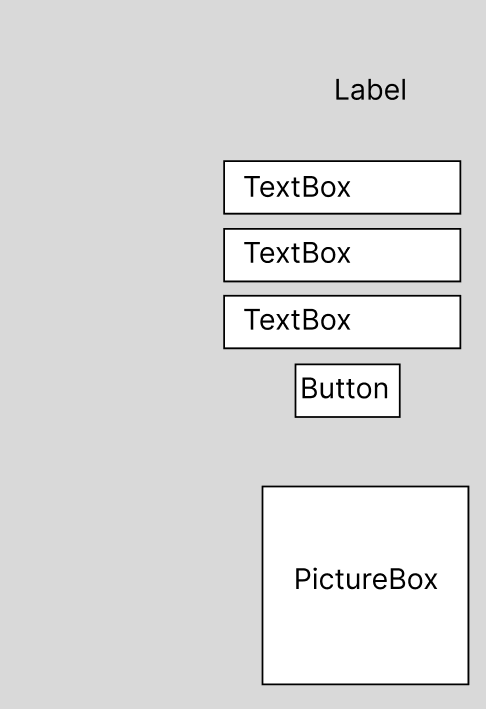


Рисунок 2.3 – Окно регистрации

Графический макет окна выбора массажа включает в себя список из видов массажа, при нажатии на который пользователь получает полную информацию о виде массажа. Так же имеются кнопки для получения минимальных медицинских показаний и оформления заказа. Окно выбора услуг изображено на рисунке 2.4.

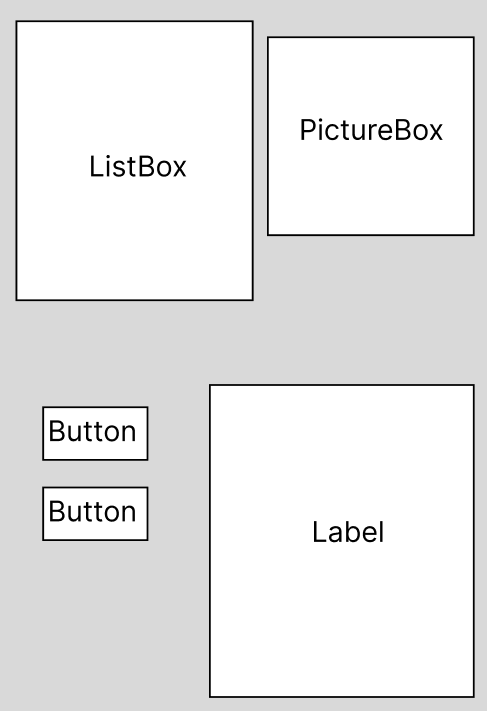


Рисунок 2.4 – Окно выбора услуг

# **3 Описание реализации программного средства**

## 3.1. Инструменты разработки и применяемые технологии

При создании программного средства использовались следующие инструменты разработки:

* операционная система Windows 10;
* среда разработки Visual Studio 2022;
* язык программирования C++;
* база данных SQLite;
* программа для создания документации Dr.Explain.

Для разработки программного продукта используются инструменты, которые включают операционную систему Windows 10, среду разработки Visual Studio 2022 и язык программирования C++.

Windows 10 – операционная система имеющая быструю скорость работы и улучшенную безопасность.

Язык программирование C++ позволяет разрабатывать большие и интересные проекты, такие как:

* игры;
* мобильные приложения;
* бизнес-приложения;
* научные приложения.

Среда разработки Visual Studio 2022 обеспечивает удобство для создания программного средства. Она снабжена множеством функций, обеспечивающая множество возможностей, которые упрощают процесс разработки [1].

SQLite – это компактная и встраиваемая реляционная база данных, которая предоставляет полнофункциональное хранилище данных внутри одного файла. Она отличается от большинства других систем управления базами данных (СУБД), так как не работает как отдельный сервер, а встраивается непосредственно в приложение [2].

Dr.Explain – программное обеспечение, предназначенное для создания профессиональной документации и справочных материалов для программного обеспечения. Оно обладает функциональностью для создания детальных инструкций, руководств пользователя, справочных руководств, помощи в виде онлайн-документации и других видов технической документации [3].

## 3.2. Порядок авторизации пользователей

В разработанном программном средстве определённый две роли:

администратор и пользователь.

Администратор может выполнять следующие функции:

* вход в программное средство;
* редактирование заказов;
* составление отчёта.

Пользователь может выполнять следующие функции:

* авторизация и регистрация;
* ознакомиться с видами массажа;
* заказать услугу массажа на дом.

## 3.3 Функции: логическая и физическая организация

Логическая организация функций:

* регистрация новых пользователей, код функции представлен ниже:

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ password = textBox1->Text;

String^ email = textBox2->Text;

String^ phone = textBox3->Text;

if (String::IsNullOrWhiteSpace(password) || password == "Пароль") {

MessageBox::Show("Введите пароль", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email) || email == "Почта") {

MessageBox::Show("Введите адрес электронной почты", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone) || phone == "Номер телефона") {

MessageBox::Show("Введите номер телефона", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(password)|| password->Contains(" ")) {

MessageBox::Show("Введите пароль без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email)|| email->Contains(" ")) {

MessageBox::Show("Введите адрес электронной почты без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone)|| phone->Contains(" ")) {

MessageBox::Show("Введите номер телефона без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

String^ databasePath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "new.db");

SQLiteConnection^ connection = gcnew SQLiteConnection("Data Source=" + databasePath + ";Version=3;");

connection->Open();

SQLiteCommand^ command = connection->CreateCommand();

command->CommandText = "INSERT INTO Users (email, password, phone) VALUES (@email, @password, @phone)";

command->Parameters->Add("@email", System::Data::DbType::String)->Value = email;

command->Parameters->Add("@password", System::Data::DbType::String)->Value = password;

command->Parameters->Add("@phone", System::Data::DbType::String)->Value = phone;

command->ExecuteNonQuery();

connection->Close();

MessageBox::Show("User added to the database\r\nPlease proceed with authentication", "Successful!", MessageBoxButtons::OK);

Log2Form^ r = gcnew Log2Form(SignSelectedLanguage, SignSelectedColor);

this->Close();

r->Show();

}

* заказ услуг массажа, код функции представлен ниже:

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ count = textBox1->Text;

String^ date = textBox2->Text;

String^ databasePath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "new.db");

SQLiteConnection^ connection = gcnew SQLiteConnection("Data Source=" + databasePath + ";Version=3;");

connection->Open();

SQLiteCommand^ command = connection->CreateCommand();

command->CommandText = "INSERT INTO Offers (count, date, type) VALUES (@count, @date, @type)";

command->Parameters->Add("@count", System::Data::DbType::String)->Value = count;

command->Parameters->Add("@date", System::Data::DbType::String)->Value = date;

command->Parameters->Add("@type", System::Data::DbType::String)->Value = TypeOfOffer;

command->ExecuteNonQuery();

connection->Close();

MessageBox::Show("Order has been created successfully", "Success!", MessageBoxButtons::OK);

this->Close();

}

Физическая организация функций:

* функции для работы с базой данных (добавление/удаление пользователя, бронирование/отмена бронирования), код функции представлен ниже:

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e)

{

String^ password = textBox1->Text;

String^ email = textBox2->Text;

String^ phone = textBox3->Text;

if (String::IsNullOrWhiteSpace(password) || password == "Пароль") {

MessageBox::Show("Введите пароль", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email) || email == "Почта") {

MessageBox::Show("Введите адрес электронной почты", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone) || phone == "Номер телефона") {

MessageBox::Show("Введите номер телефона", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(password)|| password->Contains(" ")) {

MessageBox::Show("Введите пароль без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email)|| email->Contains(" ")) {

MessageBox::Show("Введите адрес электронной почты без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone)|| phone->Contains(" ")) {

MessageBox::Show("Введите номер телефона без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

String^ databasePath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "new.db");

SQLiteConnection^ connection = gcnew SQLiteConnection("Data Source=" + databasePath + ";Version=3;");

connection->Open();

SQLiteCommand^ command = connection->CreateCommand();

command->CommandText = "INSERT INTO Users (email, password, phone) VALUES (@email, @password, @phone)";

command->Parameters->Add("@email", System::Data::DbType::String)->Value = email;

command->Parameters->Add("@password", System::Data::DbType::String)->Value = password;

command->Parameters->Add("@phone", System::Data::DbType::String)->Value = phone;

command->ExecuteNonQuery();

connection->Close();

MessageBox::Show("User added to the database\r\nPlease proceed with authentication", "Successful!", MessageBoxButtons::OK);

Log2Form^ r = gcnew Log2Form(SignSelectedLanguage, SignSelectedColor);

this->Close();

r->Show();

}

* функции для работы с графическим интерфейсом пользователя, код функции представлен ниже:

private: System::Void оливковыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FC = "Оливковый";

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "MainFormOlive.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

}

Основные функции:

* регистрация новых пользователей. Функция получает от пользователя данные для регистрации (Почту, пароль, номер телефона). Входные параметры: почта, пароль, номер телефона;
* заказ услуг массажа на дом. Функция получает от пользователя данные для услуги (название услуги, дата и время, количество сеансов). Входные параметры: название услуги, дата и время, количество сеансов.

## 3.4. Входные и выходные данные

Входными данными для администратора являются:

* электронная почта;
* пароль;
* информация о заказах.

Выходными данными для администратора является отчет о заказанных услугах за выбранный промежуток времени.

Входными данными для пользователя являются:

* электронная почта;
* пароль;
* мобильый номер телефона;
* количество сеансов;
* дата проведения услуги массажа на дом;
* дополнительная информация об заказе.

Выходными данными для пользователя является окно об успешном заказе выбранной услуги.

## 3.5 Функциональное тестирование

Данное тестирование проводится для выявления неполадок и недочётов программы на этапе её сдачи в эксплуатацию.

Функциональное тестирование – это тестирование функций приложения на соответствие требованиям. Оценка производится в соответствии с ожидаемыми и полученными результатами (на основании функциональной спецификации), при условии, что функции отрабатывали на различных значениях.

Функциональное тестирование предполагает проверку выполнения определённых на этапе проектирования функций.

Протестируем функцию авторизации. Тест кейсы для функции авторизации приведены в таблице 3.1.

Таблица 3.1 – Тест-кейсы для функции «Авторизация»

|  |  |  |  |
| --- | --- | --- | --- |
| %№ | Модуль/ функция | Шаги воспроизведения | Результат |
| 11 | Авторизация | 1. Заполняем поля данными:   * Электронная почта: не заполняем * Пароль: «123»   2. Нажать на кнопку «Авторизоваться» | Ожидаемый результат: отображение предупреждения. |
| Фактический результат: соответствует ожидаемому, представленного на рисунке 3.1 |
| 22 | Авторизация | 1. Заполнить поля данными:   * Электронная почта: «Example@gmail.com» * Пароль: «123»   2. Нажать на кнопку «Авторизоваться» | Ожидаемый результат: отображение предупреждения. |
| Фактический результат: соответствует ожидаемому, представленного на рисунке 3.2 |

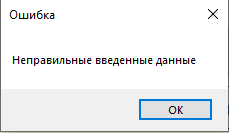


Рисунок 3.1 – Результат неполного ввода данных при авторизации

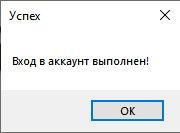


Рисунок 3.2 – Результат правильного ввода данных при авторизации

После авторизации у пользователя появляется главное окно, где он может перейти к функции оформления заказа. Протестируем функцию оформления заказа. Тест кейсы для функции оформления заказа приведены в таблице 3.2.

Таблица 3.2 – Тест-кейсы функции «Оформления заказа»

|  |  |  |  |
| --- | --- | --- | --- |
| № | Модуль/функция | Шаги воспроизведения | Результат |
| 1 | Оформления заказа | 1. Заполнить данные полей:   * Количество сеансов: «10» * Дата: «26.06.2023» * Комментарии к заказу: «Массаж будет проведён для ребёнка»   2. Нажать на кнопку «Оформить заказ» | Ожидаемый результат: отображение предупреждения. |
| Фактический результат: соответствует ожидаемому, представленного на рисунке 3.3 |
| 2 | Оформления заказа | 1. Заполнить данные полей:   * Количество сеансов: «-» * Дата: «-» * Комментарии к заказу: «-»   2. Нажать на кнопку «Оформить заказ» | Ожидаемый результат: отображение предупреждения. |
| Фактический результат: соответствует ожидаемому, представленного на рисунке 3.4 |

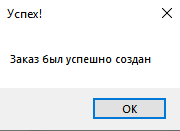


Рисунок 3.3 – Результат успешного добавления заказа

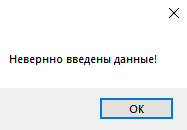


Рисунок 3.4 – Результат неверного добавления заказа

## 3.6 Описание справочной системы

Справочная система – подробный рассказ о структуре и средствах разработки использованных для создания программного средства. Цель справочной системы – представить пользователям необходимую информацию о программном средстве. Справочная система разработана в виде всплывающего окна.

Справочная система была разработана с помощью Dr.Explain. Для открытия справочной системы необходимо нажать на кнопку «Справка» на первом окне.

Dr.Explain – это программное средство для создания профессиональной документации и справочных материалов для программного обеспечения. Оно предоставляет возможность разработчикам и техническим писателям создавать наглядные и понятные инструкции, помощь пользователю и документацию к приложениям [3].

В целом, справочная система является важным компонентом любого программного приложения, цель которого – предоставить пользователям понятный и доступный источник информации о функциях и возможностях программы. Структура справочной системы и средства разработки обеспечивают пользователям доступ к информации, необходимой для эффективного и результативного использования программного продукта.

# **4 Применение**

## 4.1 Назначение программного средства

Программное средство для онлайн-сервиса заказа услуг массажа на дом предназначено для упрощения процесса заказа массажных услуг и обеспечения удобства для клиентов.

Программное средство предоставляет пользователю удобный интерфейс, позволяющий легко и быстро ознакомиться с доступными услугами массажа, выбрать удобное время и место проведения массажа, а также осуществить онлайн-заказ. Это снижает необходимость личного контакта и упрощает процесс оформления заказа для клиентов.

Программное средство позволяет клиентам ознакомиться с различными видами массажа и выбрать оптимальный для себя вариант. Клиенты могут ознакомиться с описанием каждого вида массажа, его преимуществами и рекомендациями специалистов, что помогает им сделать информированный выбор.

## 4.2 Условия применения

Для корректной работы прог средства необходимы минимальные требования:

* мышка;
* клавиатура;
* монитор;
* процессор;
* видеокарта.

Для работы программного средства на другом компьютере, необходимо перенести папку проекта и запустить файл с расширением .ex

# **Заключение**

В результате курсового проектирования была выполнена цель – разработка программного средства онлайн-сервиса заказа услуг «Массаж на дом». Для достижения данной цели были решены следующие задачи:

* выполнен анализ и проектирование системы;
* по модели приложения выполнено проектирование задачи;
* было сделано описание созданного программного средства;
* выбрана методика испытаний;
* описан весь процесс тестирования программного средства;
* приведены примеры мобильных приложений про отели, для области применения программы.

Основные функции разработанного программного средства:

* управление цветами приложения;
* возможность выбирать один из трёх языков;
* возможность пользователю пройти регистрацию;
* возможность пользователю выбрать нужный ему массаж и сделать заказ;
* возможность клиента просмотреть информацию о массаже;
* возможность админу посмотреть заказы и отредактировать их.

При разработке программного средства возникли трудности с подключением базы данных. Для решения данной трудности были использованы различные интернет-материалы.

Преимуществами разработанного программного средства являются:

* низкие системные требования;
* интуитивно понятный интерфейс;
* удобное оформление услуг массажа на дом.

Недостатком программного средства является отсутствие кроссплатформенности, что в будущем будет решено с помощью необходимых технологий.

# **Список использованных источников**

1. Visual Studio 2022 [Электронный ресурс]. - Microsoft, 2023. - Режим доступа : https://docs.microsoft.com/ru-ru/visualstudio/get-started/visual-studio-ide?view=vs-2022. - Дата доступа : 19.06.2023.
2. SQLite Documentation [Электронный ресурс] - sqlite.org, 2023. - Режим доступа : https://www.sqlite.org/docs.html/. - Дата доступа : 22.06.2023.
3. Dr.Explain [Электронный ресурс]. - drexplain.ru, 2023. - Режим доступа : https://www.drexplain.ru/. - Дата доступа : 22.06.2023.
4. profi-bel.by [Электронный ресурс]. - profi-bel.by, 2023. - Режим доступа : <https://profi-bel.by/.> - Дата доступа 25.06.2023.

# **Приложение А**

# **(обязательное)**

# **Текст программных модулей**

«MainForm.cpp»

#include "MainForm.h"

#include<Windows.h>

using namespace System;

using namespace System::Windows::Forms;

[STAThread]

void main(array<String^>^ arg)

{

Application::EnableVisualStyles();

Application::SetCompatibleTextRenderingDefault(false);

KurProject::MainForm form;

Application::Run(% form);

}

«MainForm.h»

#pragma once

#include "LogForm.h"

#include "SignForm.h"

#include "Log2Form.h"

#include "Offers.h"

#include "FirstForm.h"

namespace KurProject {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для MainForm

/// </summary>

public ref class MainForm : public System::Windows::Forms::Form

{

public:

MainForm(void)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~MainForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::Label^ label1;

protected:

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ изменитьЦветToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ оливковыйToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ мятныйToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ лавандовыйToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ изменитьЯзыкToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ русскийToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ английскийToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ белорусскийToolStripMenuItem;

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(MainForm::typeid));

this->label1 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->изменитьЦветToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->оливковыйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->мятныйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->лавандовыйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->изменитьЯзыкToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->русскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->английскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->белорусскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// label1

//

this->label1->AutoSize = true;

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label1->Cursor = System::Windows::Forms::Cursors::Hand;

this->label1->Font = (gcnew System::Drawing::Font(L"Segoe UI", 18, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->label1->ForeColor = System::Drawing::SystemColors::ControlLightLight;

this->label1->Location = System::Drawing::Point(68, 546);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(205, 32);

this->label1->TabIndex = 0;

this->label1->Text = L"Авторизоваться";

this->label1->Click += gcnew System::EventHandler(this, &MainForm::label1\_Click);

this->label1->MouseEnter += gcnew System::EventHandler(this, &MainForm::label1\_MouseEnter);

this->label1->MouseLeave += gcnew System::EventHandler(this, &MainForm::label1\_MouseLeave);

//

// label3

//

this->label3->AutoSize = true;

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label3->Cursor = System::Windows::Forms::Cursors::Hand;

this->label3->Font = (gcnew System::Drawing::Font(L"Segoe UI", 16, System::Drawing::FontStyle::Bold));

this->label3->ForeColor = System::Drawing::Color::White;

this->label3->Location = System::Drawing::Point(54, 632);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(233, 30);

this->label3->TabIndex = 2;

this->label3->Text = L"Зарегистрироваться";

this->label3->TextAlign = System::Drawing::ContentAlignment::MiddleCenter;

this->label3->Click += gcnew System::EventHandler(this, &MainForm::label3\_Click);

this->label3->MouseEnter += gcnew System::EventHandler(this, &MainForm::label3\_MouseEnter);

this->label3->MouseLeave += gcnew System::EventHandler(this, &MainForm::label3\_MouseLeave);

//

// menuStrip1

//

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(2) {

this->изменитьЦветToolStripMenuItem,

this->изменитьЯзыкToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(534, 24);

this->menuStrip1->TabIndex = 3;

this->menuStrip1->Text = L"menuStrip1";

//

// изменитьЦветToolStripMenuItem

//

this->изменитьЦветToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->оливковыйToolStripMenuItem,

this->мятныйToolStripMenuItem, this->лавандовыйToolStripMenuItem

});

this->изменитьЦветToolStripMenuItem->Font = (gcnew System::Drawing::Font(L"Segoe UI", 9, System::Drawing::FontStyle::Bold));

this->изменитьЦветToolStripMenuItem->Name = L"изменитьЦветToolStripMenuItem";

this->изменитьЦветToolStripMenuItem->Size = System::Drawing::Size(107, 20);

this->изменитьЦветToolStripMenuItem->Text = L"Изменить цвет";

//

// оливковыйToolStripMenuItem

//

this->оливковыйToolStripMenuItem->Name = L"оливковыйToolStripMenuItem";

this->оливковыйToolStripMenuItem->Size = System::Drawing::Size(148, 22);

this->оливковыйToolStripMenuItem->Text = L"Оливковый";

this->оливковыйToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::оливковыйToolStripMenuItem\_Click);

//

// мятныйToolStripMenuItem

//

this->мятныйToolStripMenuItem->Name = L"мятныйToolStripMenuItem";

this->мятныйToolStripMenuItem->Size = System::Drawing::Size(148, 22);

this->мятныйToolStripMenuItem->Text = L"Мятный";

this->мятныйToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::мятныйToolStripMenuItem\_Click);

//

// лавандовыйToolStripMenuItem

//

this->лавандовыйToolStripMenuItem->Name = L"лавандовыйToolStripMenuItem";

this->лавандовыйToolStripMenuItem->Size = System::Drawing::Size(148, 22);

this->лавандовыйToolStripMenuItem->Text = L"Лавандовый";

this->лавандовыйToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::лавандовыйToolStripMenuItem\_Click);

//

// изменитьЯзыкToolStripMenuItem

//

this->изменитьЯзыкToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->русскийToolStripMenuItem,

this->английскийToolStripMenuItem, this->белорусскийToolStripMenuItem

});

this->изменитьЯзыкToolStripMenuItem->Font = (gcnew System::Drawing::Font(L"Segoe UI", 9, System::Drawing::FontStyle::Bold));

this->изменитьЯзыкToolStripMenuItem->Name = L"изменитьЯзыкToolStripMenuItem";

this->изменитьЯзыкToolStripMenuItem->Size = System::Drawing::Size(110, 20);

this->изменитьЯзыкToolStripMenuItem->Text = L"Изменить язык";

//

// русскийToolStripMenuItem

//

this->русскийToolStripMenuItem->Name = L"русскийToolStripMenuItem";

this->русскийToolStripMenuItem->Size = System::Drawing::Size(151, 22);

this->русскийToolStripMenuItem->Text = L"Русский";

this->русскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::русскийToolStripMenuItem\_Click);

//

// английскийToolStripMenuItem

//

this->английскийToolStripMenuItem->Name = L"английскийToolStripMenuItem";

this->английскийToolStripMenuItem->Size = System::Drawing::Size(151, 22);

this->английскийToolStripMenuItem->Text = L"Английский";

this->английскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::английскийToolStripMenuItem\_Click);

//

// белорусскийToolStripMenuItem

//

this->белорусскийToolStripMenuItem->Name = L"белорусскийToolStripMenuItem";

this->белорусскийToolStripMenuItem->Size = System::Drawing::Size(151, 22);

this->белорусскийToolStripMenuItem->Text = L"Белорусский";

this->белорусскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &MainForm::белорусскийToolStripMenuItem\_Click);

//

// MainForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->AutoSizeMode = System::Windows::Forms::AutoSizeMode::GrowAndShrink;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(534, 761);

this->Controls->Add(this->label3);

this->Controls->Add(this->label1);

this->Controls->Add(this->menuStrip1);

this->DoubleBuffered = true;

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MainMenuStrip = this->menuStrip1;

this->MaximumSize = System::Drawing::Size(550, 800);

this->MinimumSize = System::Drawing::Size(550, 800);

this->Name = L"MainForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"MainForm";

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

public:

String^ FL;

String^ FC;

private: System::Void label1\_Click(System::Object^ sender, System::EventArgs^ e)/\*Log in\*/ {

LogForm^ logform = gcnew LogForm(FL,FC);

logform->Show();

}

private: System::Void label1\_MouseEnter(System::Object^ sender, System::EventArgs^ e) {

label1->ForeColor = Color::Lavender;

}

private: System::Void label1\_MouseLeave(System::Object^ sender, System::EventArgs^ e) {

label1->ForeColor = Color::White;

}

private: System::Void label3\_Click(System::Object^ sender, System::EventArgs^ e) {

SignForm^ s = gcnew SignForm(FL,FC);

s->Show();

}

private: System::Void label3\_MouseEnter(System::Object^ sender, System::EventArgs^ e) {

label3->ForeColor = Color::Lavender;

}

private: System::Void label3\_MouseLeave(System::Object^ sender, System::EventArgs^ e) {

label3->ForeColor = Color::White;

}

private: System::Void оливковыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FC = "Оливковый";

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "MainFormOlive.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

//162; 166; 141

}

private: System::Void мятныйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FC = "Мятный";

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "MainFormMint.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

//160; 212; 200

}

private: System::Void лавандовыйToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FC = "Лавандовый";

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "MainFormLavanda.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

//103; 89; 121

}

private: System::Void русскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FL = "Русский";

this->белорусскийToolStripMenuItem->Text = "Белорусский";

this->английскийToolStripMenuItem->Text = "Английский";

this->русскийToolStripMenuItem->Text = "Русский";

this->изменитьЯзыкToolStripMenuItem->Text = "Изменить язык";

this->изменитьЦветToolStripMenuItem->Text = "Изменить цвет";

this->мятныйToolStripMenuItem->Text = "Мятный";

this->оливковыйToolStripMenuItem->Text = "Оливковый";

this->лавандовыйToolStripMenuItem->Text = "Лавандовый";

this->label1->Text = "Авторизоваться";

this->label3->Text = "Зарегистрироваться";

this->label3->Location = System::Drawing::Point(54, 632);

this->label1->Location = System::Drawing::Point(68, 546);

}

private: System::Void английскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FL = "Английский";

this->белорусскийToolStripMenuItem->Text = "Belarusian";

this->английскийToolStripMenuItem->Text = "English";

this->русскийToolStripMenuItem->Text = "Russian";

this->изменитьЯзыкToolStripMenuItem->Text = "Change Language";

this->изменитьЦветToolStripMenuItem->Text = "Change Color";

this->мятныйToolStripMenuItem->Text = "Mint";

this->оливковыйToolStripMenuItem->Text = "Olive";

this->лавандовыйToolStripMenuItem->Text = "Lavender";

this->label1->Text = "Authenticate";

this->label3->Text = "Register";

this->label1->Location = System::Drawing::Point(96, 547);

this->label3->Location = System::Drawing::Point(120, 633);

}

private: System::Void белорусскийToolStripMenuItem\_Click(System::Object^ sender, System::EventArgs^ e) {

FL = "Белорусский";

this->белорусскийToolStripMenuItem->Text = "Беларускі";

this->английскийToolStripMenuItem->Text = "Англійскi";

this->русскийToolStripMenuItem->Text = "Рускі";

this->изменитьЯзыкToolStripMenuItem->Text = "Змяненне мовы";

this->изменитьЦветToolStripMenuItem->Text = "Змяненне колеру";

this->мятныйToolStripMenuItem->Text = "Мятны";

this->оливковыйToolStripMenuItem->Text = "Аліўковы";

this->лавандовыйToolStripMenuItem->Text = "Лавандовы";

this->label1->Text = "Аўтарызавацца";

this->label3->Text = "Зарэгістравацца";

this->label3->Location = System::Drawing::Point(68, 632);

this->label1->Location = System::Drawing::Point(68, 546);

}

};

}

«LogForm.h»

#pragma once

#include "SignForm.h"

#include "FirstForm.h"

#include "AdminForm.h"

namespace KurProject {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для LogForm

/// </summary>

public ref class LogForm : public System::Windows::Forms::Form

{

public:

LogForm(String^ value1, String^ value2)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

SetLogSelectedValues(value1, value2);

if (value1 == "Английский")

{

this->textBox1->Text = L"Enter your email\r\n\r\n";

this->textBox2->Text = L"Enter your password";

this->button1->Text = L"Continue";

this->button2->Text = L"Don't have an account yet? Create an account";

}

if (value1 == "Белорусский")

{

this->textBox1->Text = L"Увядзіце свой адрас электроннай пошты\r\n\r\n";

this->textBox2->Text = L"Увядзіце свой пароль";

this->button1->Text = L"Працягнуць";

this->button2->Text = L"У вас яшчэ няма уліковага запісу? Стварыце уліковы запіс";

}

if (value1 == "Русский")

{

this->textBox1->Text = L"Введите свою почту\r\n\r\n";

this->textBox2->Text = L"Введите свой пароль";

this->button1->Text = L"Продолжить";

this->button2->Text = L"У вас ещё нет аккаунта\? Создать аккаунт";

}

if (value2 == "Мятный")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "LogFormMint.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));//кнопка

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));//кнопка

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(226)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(240)));//фон

this->button2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(226)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(240)));//фон

//226;243;240

}

if (value2 == "Лавандовый")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "LogFormLavanda.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));//кнопка

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));//кнопка

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(216)), static\_cast<System::Int32>(static\_cast<System::Byte>(209)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));//фон

this->button2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(216)), static\_cast<System::Int32>(static\_cast<System::Byte>(209)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));//фон

//216;209;226

}

if (value2 == "Оливковый")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "LogFormOlive.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));//кнопка

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));//кнопка

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));//фон

this->button2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));//фон

//239;243;226

}

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~LogForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

protected:

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::Button^ button1;

private: System::Windows::Forms::Button^ button2;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(LogForm::typeid));

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->button2 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// textBox1

//

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox1->Font = (gcnew System::Drawing::Font(L"Segoe UI", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox1->ForeColor = System::Drawing::Color::White;

this->textBox1->Location = System::Drawing::Point(233, 287);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(238, 22);

this->textBox1->TabIndex = 0;

this->textBox1->Text = L"Введите почту\r\n\r\n";

this->textBox1->Click += gcnew System::EventHandler(this, &LogForm::textBox1\_Click);

//

// textBox2

//

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox2->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox2->Font = (gcnew System::Drawing::Font(L"Segoe UI", 12, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->textBox2->ForeColor = System::Drawing::Color::White;

this->textBox2->Location = System::Drawing::Point(233, 372);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(238, 22);

this->textBox2->TabIndex = 1;

this->textBox2->Text = L"Введите пароль";

this->textBox2->Click += gcnew System::EventHandler(this, &LogForm::textBox2\_Click);

//

// button1

//

this->button1->BackColor = System::Drawing::Color::Transparent;

this->button1->BackgroundImageLayout = System::Windows::Forms::ImageLayout::None;

this->button1->Cursor = System::Windows::Forms::Cursors::Hand;

this->button1->FlatAppearance->BorderSize = 0;

this->button1->FlatAppearance->MouseDownBackColor = System::Drawing::Color::Transparent;

this->button1->FlatAppearance->MouseOverBackColor = System::Drawing::Color::Transparent;

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Segoe UI", 14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->ForeColor = System::Drawing::Color::Black;

this->button1->Location = System::Drawing::Point(272, 417);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(160, 43);

this->button1->TabIndex = 2;

this->button1->Text = L"Продолжить";

this->button1->UseVisualStyleBackColor = false;

this->button1->Click += gcnew System::EventHandler(this, &LogForm::button1\_Click);

this->button1->MouseEnter += gcnew System::EventHandler(this, &LogForm::button1\_MouseEnter);

this->button1->MouseLeave += gcnew System::EventHandler(this, &LogForm::button1\_MouseLeave);

//

// button2

//

this->button2->BackColor = System::Drawing::Color::Transparent;

this->button2->BackgroundImageLayout = System::Windows::Forms::ImageLayout::None;

this->button2->Cursor = System::Windows::Forms::Cursors::Hand;

this->button2->FlatAppearance->BorderSize = 0;

this->button2->FlatAppearance->MouseDownBackColor = System::Drawing::Color::Transparent;

this->button2->FlatAppearance->MouseOverBackColor = System::Drawing::Color::Transparent;

this->button2->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button2->Font = (gcnew System::Drawing::Font(L"Segoe UI", 10, System::Drawing::FontStyle::Regular, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button2->Location = System::Drawing::Point(177, 466);

this->button2->Name = L"button2";

this->button2->Size = System::Drawing::Size(345, 27);

this->button2->TabIndex = 3;

this->button2->Text = L"У вас ещё нет аккаунта\? Создать аккаунт";

this->button2->UseVisualStyleBackColor = false;

this->button2->Click += gcnew System::EventHandler(this, &LogForm::button2\_Click);

this->button2->MouseEnter += gcnew System::EventHandler(this, &LogForm::button2\_MouseEnter);

this->button2->MouseLeave += gcnew System::EventHandler(this, &LogForm::button2\_MouseLeave);

//

// LogForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(534, 761);

this->Controls->Add(this->button2);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->DoubleBuffered = true;

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MaximumSize = System::Drawing::Size(550, 800);

this->MinimumSize = System::Drawing::Size(550, 800);

this->Name = L"LogForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"LogForm";

this->FormClosed += gcnew System::Windows::Forms::FormClosedEventHandler(this, &LogForm::LogForm\_FormClosed);

this->Load += gcnew System::EventHandler(this, &LogForm::LogForm\_Load);

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private:

String^ LogSelectedLanguage;

String^ LogSelectedColor;

public:

void SetLogSelectedValues(String^ value1, String^ value2)

{

LogSelectedLanguage = value1;

LogSelectedColor = value2;

}

public: System::Void LogForm\_FormClosed(System::Object^ sender, System::Windows::Forms::FormClosedEventArgs^ e) {

}

private: System::Void LogForm\_Load(System::Object^ sender, System::EventArgs^ e) {

}

private: System::Void textBox2\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox2->Text = String::Empty;

}

private: System::Void textBox1\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = String::Empty;

}

private: System::Void button2\_Click(System::Object^ sender, System::EventArgs^ e) {

SignForm^ signform = gcnew SignForm(LogSelectedLanguage,LogSelectedColor);

this->Close();

signform->Show();

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ databasePath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "new.db");

SQLiteConnection^ connection = gcnew SQLiteConnection("Data Source=" + databasePath + ";Version=3;");

connection->Open();

String^ emailToCheck = textBox1->Text;

String^ passwordToCheck = textBox2->Text;

if (passwordToCheck == "Admin" && emailToCheck == "Admin@gmail.com")

{

MessageBox::Show("Вход в аккаунт выполнен!", "Успех");

AdminForm^ r = gcnew AdminForm(LogSelectedLanguage, LogSelectedColor);

this->Close();

r->Show();

return;

}

String^ query = "SELECT COUNT(\*) FROM Users WHERE email = @email AND password = @password";

SQLiteCommand^ command = gcnew SQLiteCommand(query, connection);

command->Parameters->AddWithValue("@email", emailToCheck);

command->Parameters->AddWithValue("@password", passwordToCheck);

int result = Convert::ToInt32(command->ExecuteScalar());

if (LogSelectedLanguage == nullptr) {

if (result > 0)

{

MessageBox::Show("Вход в аккаунт выполнен!", "Успех");

FirstForm^ r = gcnew FirstForm(LogSelectedLanguage, LogSelectedColor);

this->Close();

r->Show();

}

else

{

MessageBox::Show("Неправильные введенные данные", "Ошибка");

}

}

if (LogSelectedLanguage == "Английский") {

if (result > 0)

{

MessageBox::Show("Logged in successfully!", "Success");

FirstForm^ r = gcnew FirstForm(LogSelectedLanguage, LogSelectedColor);

this->Close();

r->Show();

}

else

{

MessageBox::Show("Incorrect data entered", "Error");

}

}

if (LogSelectedLanguage == "Русский") {

if (result > 0)

{

MessageBox::Show("Вход в аккаунт выполнен!", "Успех");

FirstForm^ r = gcnew FirstForm(LogSelectedLanguage, LogSelectedColor);

this->Close();

r->Show();

}

else

{

MessageBox::Show("Неправильные введенные данные", "Ошибка");

}

}

if (LogSelectedLanguage == "Белорусский") {

if (result > 0)

{

MessageBox::Show("Уваход у акаўнт паспяховы!", "Паспяхова");

FirstForm^ r = gcnew FirstForm(LogSelectedLanguage, LogSelectedColor);

this->Close();

r->Show();

}

else

{

MessageBox::Show("Няправільна ўведзеныя дадзеныя", "Памылка");

}

}

connection->Close();

//FirstForm^ r = gcnew FirstForm(LogSelectedLanguage, LogSelectedColor);

//this->Close();

//r->Show();

}

private: System::Void button2\_MouseEnter(System::Object^ sender, System::EventArgs^ e) {

button2->ForeColor = Color::White;

}

private: System::Void button2\_MouseLeave(System::Object^ sender, System::EventArgs^ e) {

button2->ForeColor = Color::Black;

}

private: System::Void button1\_MouseEnter(System::Object^ sender, System::EventArgs^ e) {

button1->ForeColor = Color::White;

}

private: System::Void button1\_MouseLeave(System::Object^ sender, System::EventArgs^ e) {

button1->ForeColor = Color::Black;

}

};

}

«SignForm.h»

#pragma once

#include "Log2Form.h"

#include <string>

namespace KurProject {

using namespace System::Data::SQLite;

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для SignForm

/// </summary>

public ref class SignForm : public System::Windows::Forms::Form

{

public:

SignForm(String^ value1, String^ value2)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

SetSignSelectedValues(value1, value2);

if (value1 == "Английский")

{

this->textBox1->Text = L"Password";

this->textBox2->Text = L"Email";

this->textBox3->Text = L"Phone Number";

this->button1->Text = L"Continue";

}

if (value1 == "Русский")

{

this->textBox1->Text = L"Пароль";

this->textBox2->Text = L"Почта";

this->textBox3->Text = L"Номер телефона";

this->button1->Text = L"Продолжить";

}

if (value1 == "Белорусский")

{

this->textBox1->Text = L"Пароль";

this->textBox2->Text = L"Адрас электроннай пошты";

this->textBox3->Text = L"Нумар тэлефона";

this->button1->Text = L"Працягнуць";

}

if (value2 == "Мятный")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "RegFormMint.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

this->textBox3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(226)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(240)));

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(160)), static\_cast<System::Int32>(static\_cast<System::Byte>(212)),

static\_cast<System::Int32>(static\_cast<System::Byte>(200)));\*///кнопка

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(226)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(240)));\*///фон

}

if (value2 == "Лавандовый")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "RegFormLavanda.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

this->textBox3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(216)), static\_cast<System::Int32>(static\_cast<System::Byte>(209)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(103)), static\_cast<System::Int32>(static\_cast<System::Byte>(89)),

static\_cast<System::Int32>(static\_cast<System::Byte>(121)));\*///кнопка

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(216)), static\_cast<System::Int32>(static\_cast<System::Byte>(209)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));\*///фон

}

if (value2 == "Оливковый")

{

String^ PngPath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "RegFormOlive.png");

this->BackgroundImage = Image::FromFile(PngPath);

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->button1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));\*///кнопка

/\*System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));\*///фон

}

}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~SignForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::TextBox^ textBox1;

protected:

private: System::Windows::Forms::TextBox^ textBox2;

private: System::Windows::Forms::TextBox^ textBox3;

private: System::Windows::Forms::Button^ button1;

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(SignForm::typeid));

this->textBox1 = (gcnew System::Windows::Forms::TextBox());

this->textBox2 = (gcnew System::Windows::Forms::TextBox());

this->textBox3 = (gcnew System::Windows::Forms::TextBox());

this->button1 = (gcnew System::Windows::Forms::Button());

this->SuspendLayout();

//

// textBox1

//

this->textBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox1->Font = (gcnew System::Drawing::Font(L"Segoe UI", 12));

this->textBox1->ForeColor = System::Drawing::Color::White;

this->textBox1->Location = System::Drawing::Point(95, 247);

this->textBox1->Name = L"textBox1";

this->textBox1->Size = System::Drawing::Size(243, 22);

this->textBox1->TabIndex = 0;

this->textBox1->Text = L"Пароль";

this->textBox1->Click += gcnew System::EventHandler(this, &SignForm::textBox1\_Click);

//

// textBox2

//

this->textBox2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox2->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox2->Font = (gcnew System::Drawing::Font(L"Segoe UI", 12));

this->textBox2->ForeColor = System::Drawing::Color::White;

this->textBox2->Location = System::Drawing::Point(95, 332);

this->textBox2->Name = L"textBox2";

this->textBox2->Size = System::Drawing::Size(243, 22);

this->textBox2->TabIndex = 1;

this->textBox2->Text = L"Почта";

this->textBox2->Click += gcnew System::EventHandler(this, &SignForm::textBox2\_Click);

//

// textBox3

//

this->textBox3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->textBox3->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->textBox3->Font = (gcnew System::Drawing::Font(L"Segoe UI", 12));

this->textBox3->ForeColor = System::Drawing::Color::White;

this->textBox3->Location = System::Drawing::Point(95, 415);

this->textBox3->Name = L"textBox3";

this->textBox3->Size = System::Drawing::Size(243, 22);

this->textBox3->TabIndex = 2;

this->textBox3->Text = L"Номер телефона";

this->textBox3->Click += gcnew System::EventHandler(this, &SignForm::textBox3\_Click);

//

// button1

//

this->button1->BackColor = System::Drawing::Color::Transparent;

this->button1->BackgroundImageLayout = System::Windows::Forms::ImageLayout::None;

this->button1->Cursor = System::Windows::Forms::Cursors::Hand;

this->button1->FlatAppearance->BorderSize = 0;

this->button1->FlatAppearance->MouseDownBackColor = System::Drawing::Color::Transparent;

this->button1->FlatAppearance->MouseOverBackColor = System::Drawing::Color::Transparent;

this->button1->FlatStyle = System::Windows::Forms::FlatStyle::Flat;

this->button1->Font = (gcnew System::Drawing::Font(L"Segoe UI", 14, System::Drawing::FontStyle::Bold, System::Drawing::GraphicsUnit::Point,

static\_cast<System::Byte>(204)));

this->button1->ForeColor = System::Drawing::Color::Black;

this->button1->Location = System::Drawing::Point(145, 467);

this->button1->Name = L"button1";

this->button1->Size = System::Drawing::Size(144, 35);

this->button1->TabIndex = 3;

this->button1->Text = L"Продолжить";

this->button1->UseVisualStyleBackColor = false;

this->button1->Click += gcnew System::EventHandler(this, &SignForm::button1\_Click);

this->button1->MouseEnter += gcnew System::EventHandler(this, &SignForm::button1\_MouseEnter);

this->button1->MouseLeave += gcnew System::EventHandler(this, &SignForm::button1\_MouseLeave);

//

// SignForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(534, 761);

this->Controls->Add(this->button1);

this->Controls->Add(this->textBox3);

this->Controls->Add(this->textBox2);

this->Controls->Add(this->textBox1);

this->DoubleBuffered = true;

this->FormBorderStyle = System::Windows::Forms::FormBorderStyle::FixedSingle;

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MaximumSize = System::Drawing::Size(550, 800);

this->MinimumSize = System::Drawing::Size(550, 800);

this->Name = L"SignForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"SignForm";

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private:

String^ SignSelectedLanguage;

String^ SignSelectedColor;

public:

void SetSignSelectedValues(String^ value1, String^ value2)

{

SignSelectedLanguage = value1;

SignSelectedColor = value2;

}

private: System::Void textBox1\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox1->Text = String::Empty;

}

private: System::Void textBox2\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox2->Text = String::Empty;

}

private: System::Void textBox3\_Click(System::Object^ sender, System::EventArgs^ e) {

textBox3->Text = String::Empty;

}

private: System::Void button1\_Click(System::Object^ sender, System::EventArgs^ e) {

String^ password = textBox1->Text;

String^ email = textBox2->Text;

String^ phone = textBox3->Text;

///////////////////// Проверки на всё /////////////////////

if (String::IsNullOrWhiteSpace(password) || password == "Пароль") {

MessageBox::Show("Введите пароль", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email) || email == "Почта") {

MessageBox::Show("Введите адрес электронной почты", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone) || phone == "Номер телефона") {

MessageBox::Show("Введите номер телефона", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(password)|| password->Contains(" ")) {

MessageBox::Show("Введите пароль без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(email)|| email->Contains(" ")) {

MessageBox::Show("Введите адрес электронной почты без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

if (String::IsNullOrWhiteSpace(phone)|| phone->Contains(" ")) {

MessageBox::Show("Введите номер телефона без пробелов", "Ошибка", MessageBoxButtons::OK, MessageBoxIcon::Error);

return;

}

///////////////////// Проверки на всё /////////////////////

String^ databasePath = System::IO::Path::Combine(AppDomain::CurrentDomain->BaseDirectory, "new.db");

SQLiteConnection^ connection = gcnew SQLiteConnection("Data Source=" + databasePath + ";Version=3;");

connection->Open();

SQLiteCommand^ command = connection->CreateCommand();

command->CommandText = "INSERT INTO Users (email, password, phone) VALUES (@email, @password, @phone)";

command->Parameters->Add("@email", System::Data::DbType::String)->Value = email;

command->Parameters->Add("@password", System::Data::DbType::String)->Value = password;

command->Parameters->Add("@phone", System::Data::DbType::String)->Value = phone;

command->ExecuteNonQuery();

connection->Close();

if(SignSelectedLanguage == "Русский")

{

MessageBox::Show("Пользователь добавлен в базу данных\r\nПройдите авторизацию","Успех!", MessageBoxButtons::OK);

}

if (SignSelectedLanguage == "Английский")

{

MessageBox::Show("User added to the database\r\nPlease proceed with authentication", "Successful!", MessageBoxButtons::OK);

}

if (SignSelectedLanguage == "Белорусский")

{

MessageBox::Show("Карыстальнік дададзены ў базу даных\r\nКалі ласка, правядзіце аўтэнтыфікацыю", "Паспяхова!", MessageBoxButtons::OK);

}

Log2Form^ r = gcnew Log2Form(SignSelectedLanguage, SignSelectedColor);

this->Close();

r->Show();

}

private: System::Void button1\_MouseEnter(System::Object^ sender, System::EventArgs^ e) {

button1->ForeColor = Color::DarkSlateGray;

}

private: System::Void button1\_MouseLeave(System::Object^ sender, System::EventArgs^ e) {

button1->ForeColor = Color::Black;

}

};

}

«FirstForm.h»

#pragma once

#include "Offers.h"

namespace KurProject {

using namespace System;

using namespace System::ComponentModel;

using namespace System::Collections;

using namespace System::Windows::Forms;

using namespace System::Data;

using namespace System::Drawing;

/// <summary>

/// Сводка для FirstForm

/// </summary>

public ref class FirstForm : public System::Windows::Forms::Form

{

private:

bool aOpened; // Ссылка на форму O

Offers^ o; // Поле класса для хранения ссылки на форму "a"

private: System::Windows::Forms::ToolStripMenuItem^ изменениеЯзыкаToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ русскийToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ английскийToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ белорусскийToolStripMenuItem;

bool oOpened; // Ссылка на форму O

public:

FirstForm(String^ value1, String^ value2)

{

InitializeComponent();

//

//TODO: добавьте код конструктора

//

SetFirstSelectedValues(value1, value2);

o = nullptr;

oOpened = false;

}

//void ShowForm2() {

// Form2^ form2 = gcnew Form2();

// form2->Show();

// this->Hide();

//}

//void ShowForm1() {

// this->Show();

//}

protected:

/// <summary>

/// Освободить все используемые ресурсы.

/// </summary>

~FirstForm()

{

if (components)

{

delete components;

}

}

private: System::Windows::Forms::ListBox^ listBox1;

private: System::Windows::Forms::Label^ label1;

private: System::Windows::Forms::Label^ label2;

private: System::Windows::Forms::Label^ label3;

private: System::Windows::Forms::Label^ label4;

private: System::Windows::Forms::MenuStrip^ menuStrip1;

private: System::Windows::Forms::ToolStripMenuItem^ изменениеЦветаToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ лавандовыйToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ оливковыйToolStripMenuItem;

private: System::Windows::Forms::ToolStripMenuItem^ мятныйToolStripMenuItem;

protected:

protected:

private:

/// <summary>

/// Обязательная переменная конструктора.

/// </summary>

System::ComponentModel::Container^ components;

#pragma region Windows Form Designer generated code

/// <summary>

/// Требуемый метод для поддержки конструктора — не изменяйте

/// содержимое этого метода с помощью редактора кода.

/// </summary>

void InitializeComponent(void)

{

System::ComponentModel::ComponentResourceManager^ resources = (gcnew System::ComponentModel::ComponentResourceManager(FirstForm::typeid));

this->listBox1 = (gcnew System::Windows::Forms::ListBox());

this->label1 = (gcnew System::Windows::Forms::Label());

this->label2 = (gcnew System::Windows::Forms::Label());

this->label3 = (gcnew System::Windows::Forms::Label());

this->label4 = (gcnew System::Windows::Forms::Label());

this->menuStrip1 = (gcnew System::Windows::Forms::MenuStrip());

this->изменениеЦветаToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->лавандовыйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->оливковыйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->мятныйToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->изменениеЯзыкаToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->русскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->английскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->белорусскийToolStripMenuItem = (gcnew System::Windows::Forms::ToolStripMenuItem());

this->menuStrip1->SuspendLayout();

this->SuspendLayout();

//

// listBox1

//

this->listBox1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->listBox1->BorderStyle = System::Windows::Forms::BorderStyle::None;

this->listBox1->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 14));

this->listBox1->FormattingEnabled = true;

this->listBox1->ItemHeight = 25;

this->listBox1->Items->AddRange(gcnew cli::array< System::Object^ >(10) {

L"Классический", L"Детский", L"Лечебный", L"Косметологический",

L"Медовый", L"Спортивный", L"Точечный", L"Аппаратный", L"Аромамассаж", L"Стоун-терапия"

});

this->listBox1->Location = System::Drawing::Point(0, 2);

this->listBox1->Name = L"listBox1";

this->listBox1->Size = System::Drawing::Size(283, 300);

this->listBox1->TabIndex = 0;

this->listBox1->SelectedIndexChanged += gcnew System::EventHandler(this, &FirstForm::listBox1\_SelectedIndexChanged);

//

// label1

//

this->label1->AutoSize = true;

this->label1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label1->Cursor = System::Windows::Forms::Cursors::Hand;

this->label1->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 10));

this->label1->ForeColor = System::Drawing::Color::White;

this->label1->Location = System::Drawing::Point(28, 449);

this->label1->Name = L"label1";

this->label1->Size = System::Drawing::Size(99, 38);

this->label1->TabIndex = 1;

this->label1->Text = L"Медицинские\r\n показания\r\n";

this->label1->Click += gcnew System::EventHandler(this, &FirstForm::label1\_Click);

this->label1->MouseEnter += gcnew System::EventHandler(this, &FirstForm::label1\_MouseEnter);

this->label1->MouseLeave += gcnew System::EventHandler(this, &FirstForm::label1\_MouseLeave);

//

// label2

//

this->label2->AutoSize = true;

this->label2->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label2->Cursor = System::Windows::Forms::Cursors::Hand;

this->label2->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 10));

this->label2->ForeColor = System::Drawing::Color::White;

this->label2->Location = System::Drawing::Point(38, 510);

this->label2->Name = L"label2";

this->label2->Size = System::Drawing::Size(77, 38);

this->label2->TabIndex = 2;

this->label2->Text = L"Оформить\r\n заказ";

this->label2->Click += gcnew System::EventHandler(this, &FirstForm::label2\_Click);

this->label2->MouseEnter += gcnew System::EventHandler(this, &FirstForm::label2\_MouseEnter);

this->label2->MouseLeave += gcnew System::EventHandler(this, &FirstForm::label2\_MouseLeave);

//

// label3

//

this->label3->AutoSize = true;

this->label3->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));

this->label3->Cursor = System::Windows::Forms::Cursors::Hand;

this->label3->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 32));

this->label3->ForeColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(239)), static\_cast<System::Int32>(static\_cast<System::Byte>(243)),

static\_cast<System::Int32>(static\_cast<System::Byte>(226)));

this->label3->Location = System::Drawing::Point(-39, 558);

this->label3->Name = L"label3";

this->label3->Size = System::Drawing::Size(188, 59);

this->label3->TabIndex = 3;

this->label3->Text = L"Аккаунт\r\n";

//

// label4

//

this->label4->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->label4->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 11.5F));

this->label4->Location = System::Drawing::Point(163, 393);

this->label4->Name = L"label4";

this->label4->Size = System::Drawing::Size(359, 317);

this->label4->TabIndex = 4;

this->label4->Text = L"Выберите нужный вам массаж\r\n";

//

// menuStrip1

//

this->menuStrip1->BackColor = System::Drawing::Color::FromArgb(static\_cast<System::Int32>(static\_cast<System::Byte>(162)), static\_cast<System::Int32>(static\_cast<System::Byte>(166)),

static\_cast<System::Int32>(static\_cast<System::Byte>(141)));

this->menuStrip1->Items->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(2) {

this->изменениеЦветаToolStripMenuItem,

this->изменениеЯзыкаToolStripMenuItem

});

this->menuStrip1->Location = System::Drawing::Point(0, 0);

this->menuStrip1->Name = L"menuStrip1";

this->menuStrip1->Size = System::Drawing::Size(534, 29);

this->menuStrip1->TabIndex = 5;

this->menuStrip1->Text = L"menuStrip1";

this->menuStrip1->Visible = false;

//

// изменениеЦветаToolStripMenuItem

//

this->изменениеЦветаToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->лавандовыйToolStripMenuItem,

this->оливковыйToolStripMenuItem, this->мятныйToolStripMenuItem

});

this->изменениеЦветаToolStripMenuItem->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 11.5F));

this->изменениеЦветаToolStripMenuItem->Name = L"изменениеЦветаToolStripMenuItem";

this->изменениеЦветаToolStripMenuItem->Size = System::Drawing::Size(155, 25);

this->изменениеЦветаToolStripMenuItem->Text = L"Изменение цвета";

//

// лавандовыйToolStripMenuItem

//

this->лавандовыйToolStripMenuItem->Name = L"лавандовыйToolStripMenuItem";

this->лавандовыйToolStripMenuItem->Size = System::Drawing::Size(175, 26);

this->лавандовыйToolStripMenuItem->Text = L"Лавандовый";

this->лавандовыйToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::лавандовыйToolStripMenuItem\_Click);

//

// оливковыйToolStripMenuItem

//

this->оливковыйToolStripMenuItem->Name = L"оливковыйToolStripMenuItem";

this->оливковыйToolStripMenuItem->Size = System::Drawing::Size(175, 26);

this->оливковыйToolStripMenuItem->Text = L"Оливковый";

this->оливковыйToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::оливковыйToolStripMenuItem\_Click);

//

// мятныйToolStripMenuItem

//

this->мятныйToolStripMenuItem->Name = L"мятныйToolStripMenuItem";

this->мятныйToolStripMenuItem->Size = System::Drawing::Size(175, 26);

this->мятныйToolStripMenuItem->Text = L"Мятный";

this->мятныйToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::мятныйToolStripMenuItem\_Click);

//

// изменениеЯзыкаToolStripMenuItem

//

this->изменениеЯзыкаToolStripMenuItem->DropDownItems->AddRange(gcnew cli::array< System::Windows::Forms::ToolStripItem^ >(3) {

this->русскийToolStripMenuItem,

this->английскийToolStripMenuItem, this->белорусскийToolStripMenuItem

});

this->изменениеЯзыкаToolStripMenuItem->Font = (gcnew System::Drawing::Font(L"Segoe UI Semibold", 11.5F));

this->изменениеЯзыкаToolStripMenuItem->Name = L"изменениеЯзыкаToolStripMenuItem";

this->изменениеЯзыкаToolStripMenuItem->Size = System::Drawing::Size(156, 25);

this->изменениеЯзыкаToolStripMenuItem->Text = L"Изменение языка";

//

// русскийToolStripMenuItem

//

this->русскийToolStripMenuItem->Name = L"русскийToolStripMenuItem";

this->русскийToolStripMenuItem->Size = System::Drawing::Size(180, 26);

this->русскийToolStripMenuItem->Text = L"Русский";

this->русскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::русскийToolStripMenuItem\_Click);

//

// английскийToolStripMenuItem

//

this->английскийToolStripMenuItem->Name = L"английскийToolStripMenuItem";

this->английскийToolStripMenuItem->Size = System::Drawing::Size(180, 26);

this->английскийToolStripMenuItem->Text = L"Английский";

this->английскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::английскийToolStripMenuItem\_Click);

//

// белорусскийToolStripMenuItem

//

this->белорусскийToolStripMenuItem->Name = L"белорусскийToolStripMenuItem";

this->белорусскийToolStripMenuItem->Size = System::Drawing::Size(180, 26);

this->белорусскийToolStripMenuItem->Text = L"Белорусский";

this->белорусскийToolStripMenuItem->Click += gcnew System::EventHandler(this, &FirstForm::белорусскийToolStripMenuItem\_Click);

//

// FirstForm

//

this->AutoScaleDimensions = System::Drawing::SizeF(6, 13);

this->AutoScaleMode = System::Windows::Forms::AutoScaleMode::Font;

this->BackgroundImage = (cli::safe\_cast<System::Drawing::Image^>(resources->GetObject(L"$this.BackgroundImage")));

this->BackgroundImageLayout = System::Windows::Forms::ImageLayout::Stretch;

this->ClientSize = System::Drawing::Size(534, 761);

this->Controls->Add(this->label4);

this->Controls->Add(this->label3);

this->Controls->Add(this->label2);

this->Controls->Add(this->label1);

this->Controls->Add(this->listBox1);

this->Controls->Add(this->menuStrip1);

this->DoubleBuffered = true;

this->Icon = (cli::safe\_cast<System::Drawing::Icon^>(resources->GetObject(L"$this.Icon")));

this->MainMenuStrip = this->menuStrip1;

this->MaximumSize = System::Drawing::Size(550, 800);

this->MinimumSize = System::Drawing::Size(550, 800);

this->Name = L"FirstForm";

this->StartPosition = System::Windows::Forms::FormStartPosition::CenterScreen;

this->Text = L"Type of massage";

this->menuStrip1->ResumeLayout(false);

this->menuStrip1->PerformLayout();

this->ResumeLayout(false);

this->PerformLayout();

}

#pragma endregion

private:

String^ FirstSelectedLanguage;

String^ FirstSelectedColor;

public:

void SetFirstSelectedValues(String^ value1, String^ value2)

{

FirstSelectedLanguage = value1;

FirstSelectedColor = value2;

}

**Этикетка**

для курсовых проектов

**Курсовой проект**

Тема \_\_\_\_\_\_ Программное средство для онлайн-сервиса заказа услуг «Массаж на дом»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

КП Т.192015.401

Разработан \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Утверждён\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Разработчик: \_\_\_\_\_ Сидоренко А. А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Руководитель: \_\_\_\_ Дж. А. Миронова\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Технические средства \_\_\_Процессор Intel Core i7 7600U 2800 МГц, оперативная память DDR4 8 ГБ, встроенная видеокарта Intel HD Graphics 620, винчестер SSD 512 ГБ, клавиатура и мышь.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Программные средства: \_\_\_\_\_\_\_\_\_Visual Studio 2022, SQLite, операционная система Windows 10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Состав документа:**

Пояснительная записка – файл («ПЗ\_Сидоренко\_КП.docx»)

Программные документы – файлы («Курсовой проект»)

Графическая часть – файлы («ГЧ1.docx»), («ГЧ2.docx»), («ГЧ3.docx»).

Сведения о защите информации: нет

**Удостоверяющий лист**

электронного документа – курсовой проект

Тема КП Программное средство для онлайн-сервиса заказа услуг «Массаж на дом»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Обозначение КП \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ КП Т.192015.401\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Разработчик \_Сидоренко А. А. Руководитель \_Дж. А. Миронова\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.) (Ф.И.О.)

Председатель цикловой комиссии \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Ф.И.О.)

Подписи лиц, ответственных за разработку электронного документа

|  |  |  |
| --- | --- | --- |
| Состав ЭД | Разработчик | Руководитель |
| Пояснительная записка (на  бумажном носителе формата  А4), файл ПЗ\_Сидоренко\_КП.docx |  |  |
| ГЧ №1, файл «ГЧ1.docx» |  |  |
| ГЧ №2, файл «ГЧ2.docx» |  |  |
| ГЧ №3, файл «ГЧ3.docx» |  |  |
| Программная документация: папка с проектом «Курсовой проект» |  |  |
| Тип носителя: магнитный диск |  |  |