МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Программной инженерии

Специальность 1-47 01 02 «Дизайн электронных и веб-изданий»

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
КУРСОВОГО ПРОЕКТА:**

по дисциплине          «Объектно-ориентированное программирование»

Тема       «IT лаборатория»

Исполнитель

Студент(ка) 2 курса группы 9                    Ездакова Полина Вадимовна

(Ф.И.О.)

Руководитель работы    асс. каф. ПИ Р.И. Белькевич

(учен. степень, звание, должность, подпись, Ф.И.О.)  
  
  
Курсовой проект защищен с оценкой

Минск 2022

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc105088220)

[1 Постановка задачи 4](#_Toc105088221)

[1.1 Формулировка проблемы и алгоритма ее решения 4](#_Toc105088222)

[1.2 Анализ прототипов и аналогов 4](#_Toc105088223)

[2 Проектирование приложения 6](#_Toc105088224)

[2.1 Проектирование базы данных 6](#_Toc105088225)

[2.2 Проектирование дизайна приложения 7](#_Toc105088226)

[3 Разработка приложения 11](#_Toc105088227)

[3.1 Описание функций приложения 11](#_Toc105088228)

[3.2 Классы базы данных 11](#_Toc105088229)

[3.3 Основные окна и страницы приложения 12](#_Toc105088230)

[3.3.1 Окно авторизации 12](#_Toc105088231)

[3.3.2 Окно регистрации 12](#_Toc105088232)

[3.3.3 Окно «Кабинет пользователя» 13](#_Toc105088233)

[3.3.4 Диалоговое окно связи администратора с пользователем 14](#_Toc105088234)

[3.3.5 Окно администратора 15](#_Toc105088235)

[4 Руководство по установке и использованию 15](#_Toc105088236)

[5 Тестирование программного средства 17](#_Toc105088237)

[Заключение 19](#_Toc105088238)

[Список используемых источников 20](#_Toc105088239)

[Приложение А 21](#_Toc105088240)

ВВЕДЕНИЕ

В наше время компьютер стал неотъемлемой составляющей в жизни каждого человека, особенно если он IT-специалист. Невозможно представить себе жизнь без хорошего, грамотно собранного железа. Мое приложение позволяет быстро собрать хороший компьютер.

Целью курсового проекта является: развитие практических навыков вёрстки приложений на базе WPF (Windows Presentation Foundation) с использованием языка разметки — XAML (eXtensible Application Markup Language), и языка программирования — С#; разработка IU и UX дизайна.

Разработка приложения происходит в программной среде Microsoft Visual Studio 2022 на платформе WPF — это платформа пользовательского интерфейса для создания клиентских приложений для настольных систем.

Данные для работы хранятся в базе данных. Её создание происходит с помощью Microsoft SQL Server — это система управления реляционными базами данных, которая подключается к приложению.

Для работы с самим SQL Server используется Entity Framework, который представляет специальную объектно-ориентированную технологию на базе фреймворка .NET для работы с данными. Он позволяет работать напрямую с объектами, данные для которых хранятся в таблицах базы данных.

Курсовой проект включает в себя разделы: введение, содержание, обзор технический методов и программных средств разработки приложения, последовательность вёрстки приложения, составление его структуры, описание хода работы программного средства с копиями экрана пользователя, заключение, список использованной литературы.

# Постановка задачи

Основной задачей курсового проекта является разработка функционального приложения, позволяющего собрать компьютер.

## Формулировка проблемы и алгоритма ее решения

Основное наполнение приложения — фильмы. В приложении будет реализована возможность просмотра информации о фильме сохранения добавленного в библиотеку фильма себе на компьютер. Пользователь может осуществлять фильтрацию по жанру, а затем купить и добавить в библиотеку.

Основной функционал приложения имеет следующий вид:

* регистрация нового пользователя;
* авторизация уже существующих пользователей;
* авторизация администратора;
* поддержка работы с базой данных;
* добавление и удаление комплектующих для компьютера (для администратора);
* возможность связаться с администратором.

## Анализ прототипов и аналогов

Сайт «multimart.by» (Рисунок 1.1.)

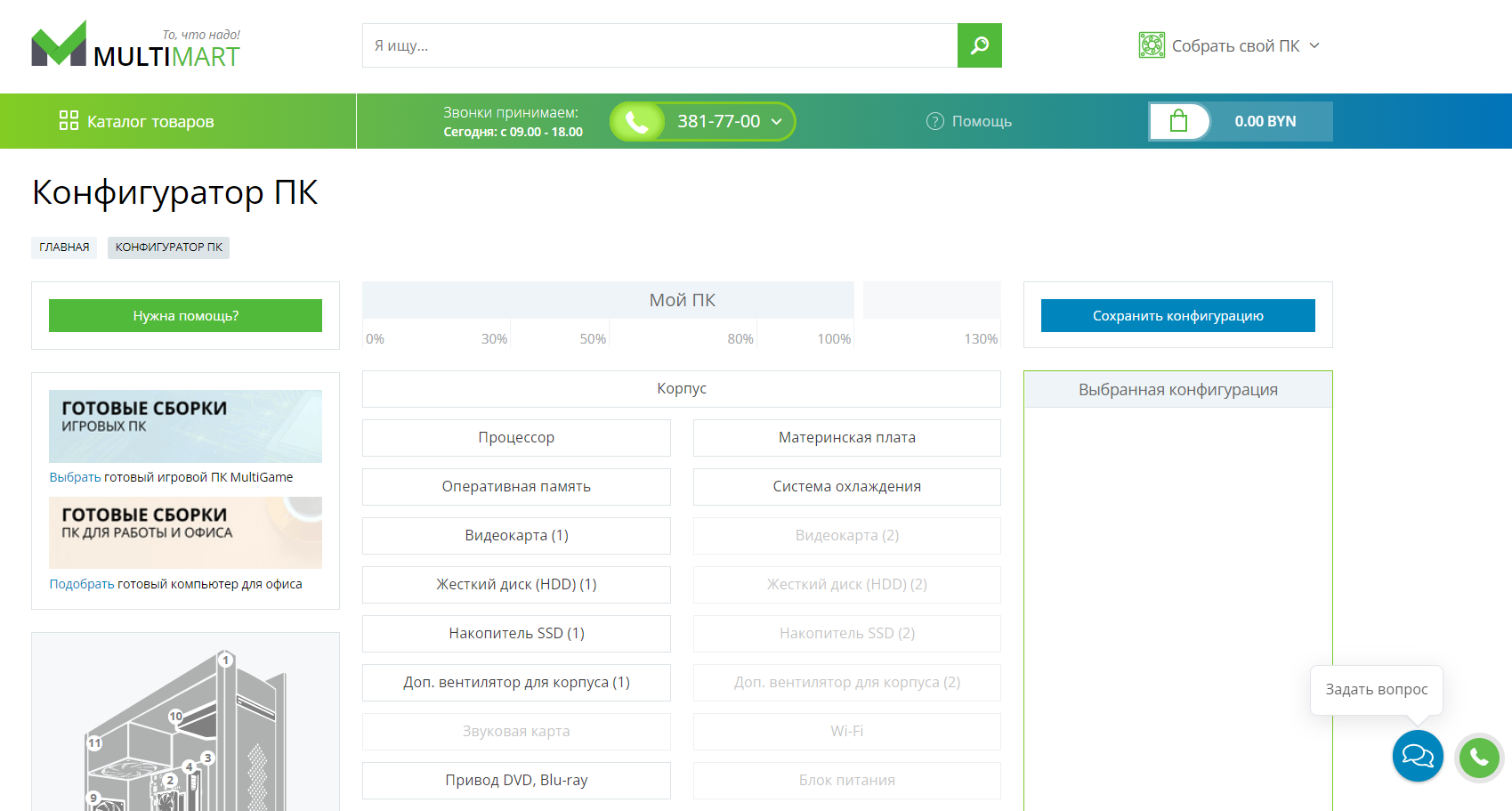


Рисунок 1.1 – Сайт «multimart.by».

Сайт дает возможность собрать свой собственный компьютер. Имеет много возможностей и позволяет быстро и качественно собрать полноценный ПК.

Сайт «edelws»

У сайта присутствует очень гармонический дизайн, благодаря которому собирать компьютер становится приятнее. Имеет много возможностей и комплектующих. (Рисунок 1.2.)

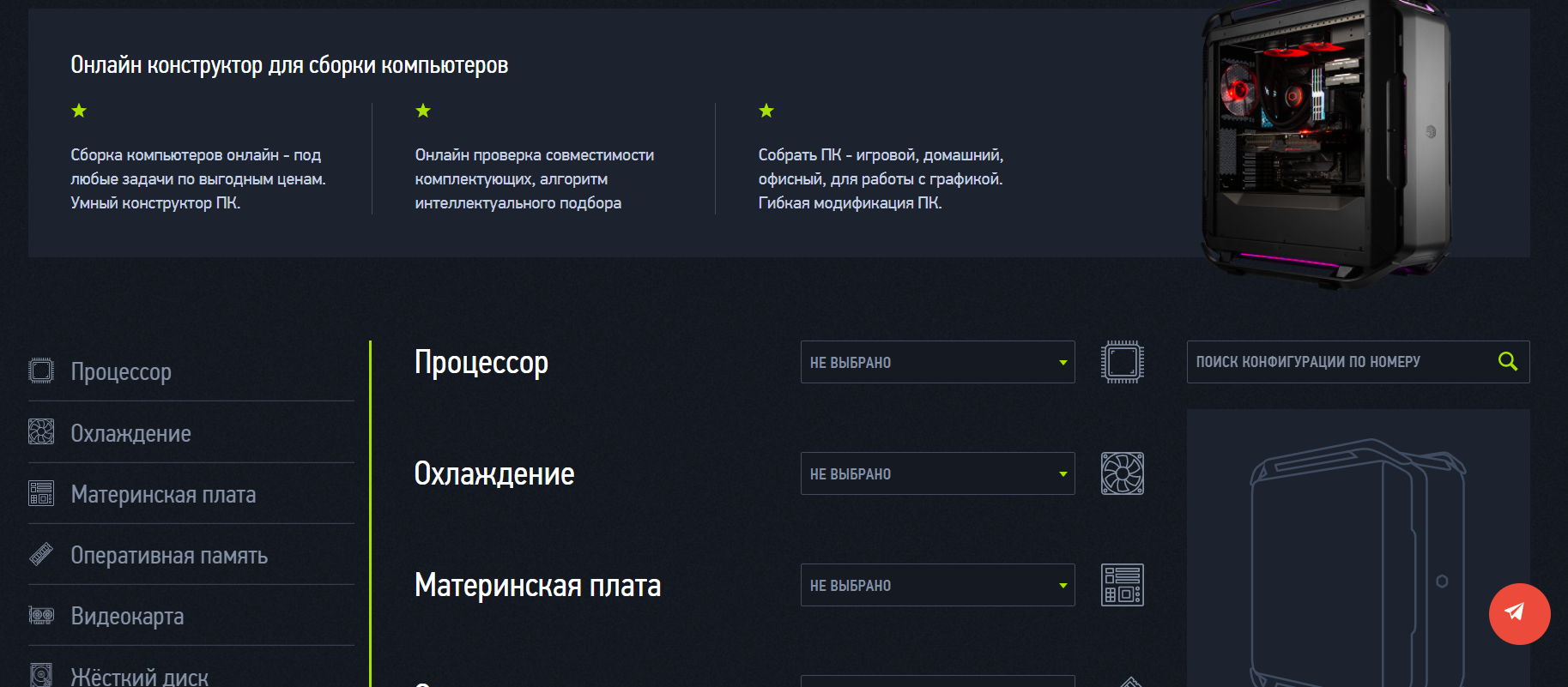


Рисунок 1.2 – Сайт «edelws».

Сайт «cooler».

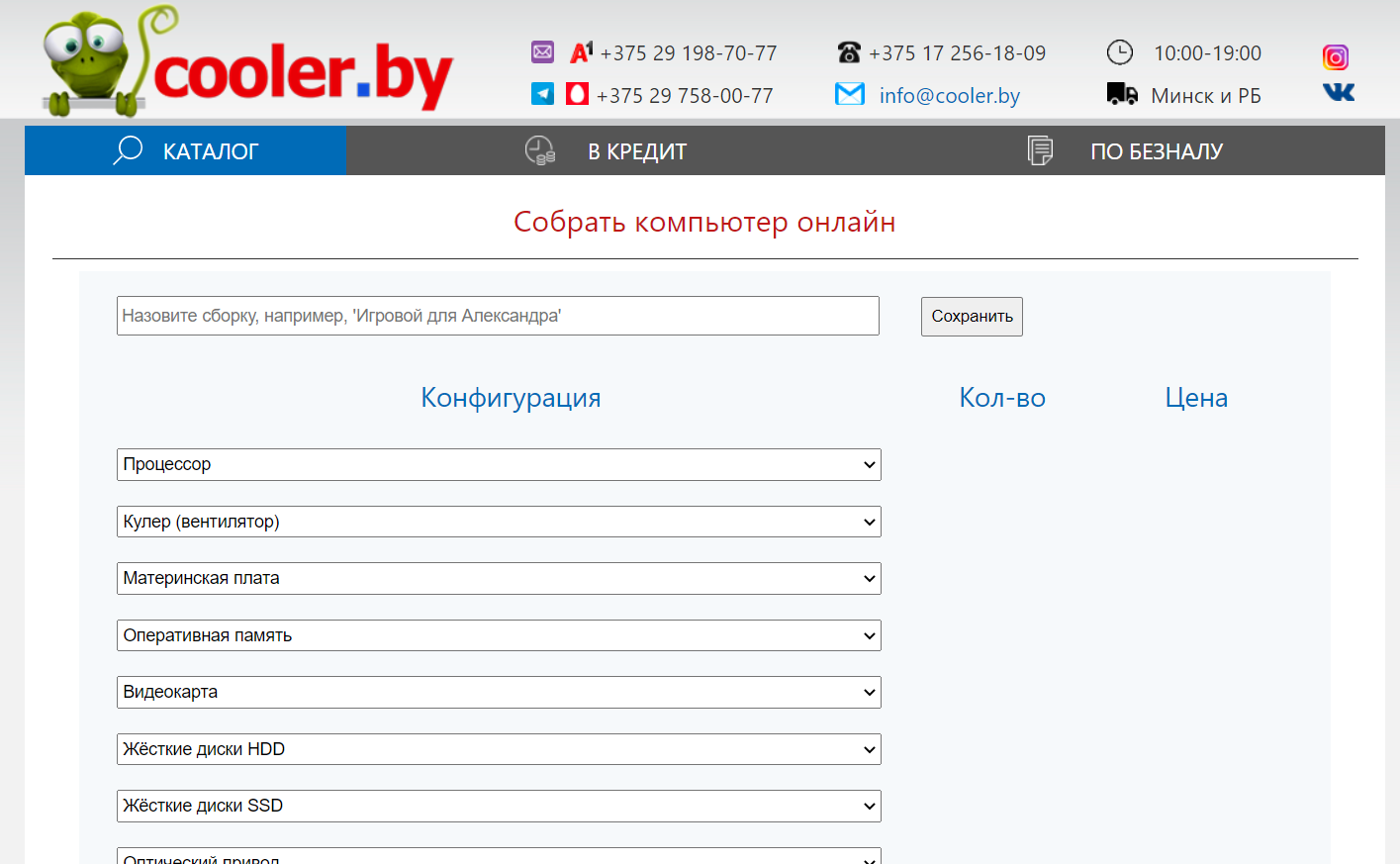


Рисунок 1.3 – Сайт «cooler».

Данный сайт выглядит не так привлекательно, как прошлый. Довольно прост в использовании. Принцип работы такой же, как и у других сайтов: сборка, расчет, заказ. (Рисунок 1.3.)

# Проектирование приложения

## Проектирование базы данных

При разработке приложения в первую очередь понадобилось создать базу данных так, чтобы она удовлетворяла потребностям приложения и не была избыточной.

База данных состоит из 3 таблиц, между которыми установлены связи (Рисунок 3.1.)

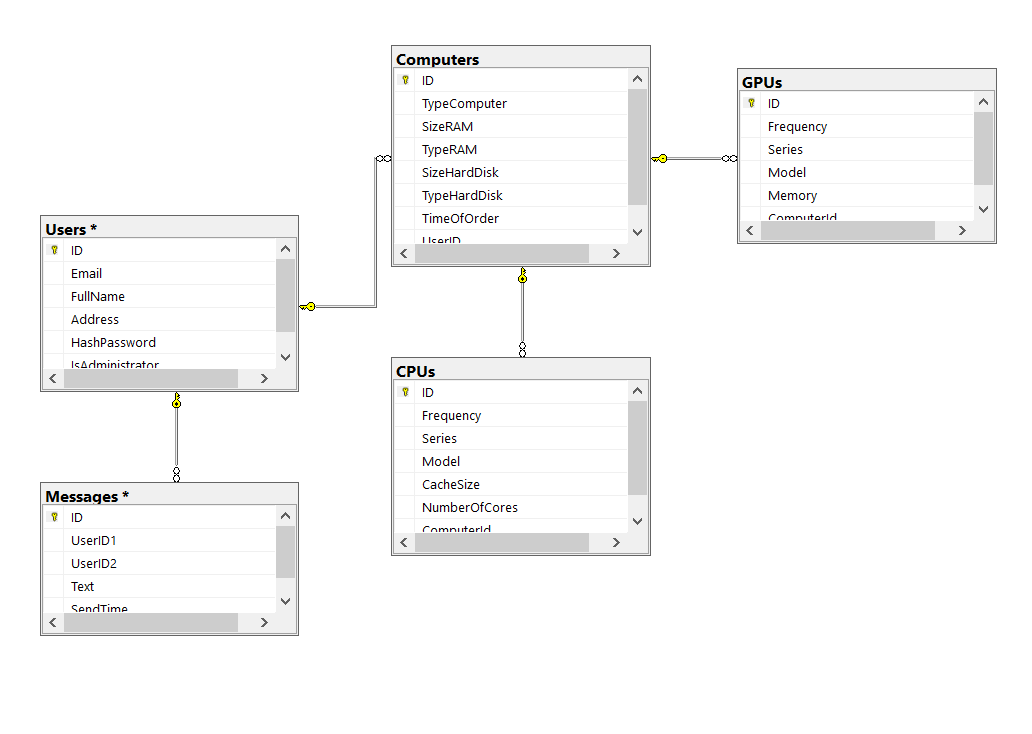


Рисунок 2.1 – Схема базы данных.

Таблица **Users** предназначена для хранения информации о зарегистрированных пользователях, состоит из 10 столбцов:

* Уникальный ключ пользователя­– ключевое поле таблицы, которое содержит уникальный номер каждого пользователя;
* ФИО;
* Почта пользователя;
* Адрес пользователя;
* Права доступа;

Таблица **Computers** служит для хранения информации о составляющих компьютера, а также информацию об ID пользователей, кем данный компьютер был собран:

* Код компьютера;
* Тип компьютера;
* Размер ОЗУ;
* Тип ОЗУ;
* Размер жесткого диска;
* Тип жесткого диска;
* Дата;
* Код пользователя, собравшего компьютер.

Таблица **CPUs** хранит в себе информацию о процессоре, состоит из 10 столбцов:

* Код процессора – ключевое поле таблицы, уникальные номер;
* Производитель;
* Серия;
* Модель;
* Количество ядер процессора;
* Размер кэша;
* ID компьютера.

Таблица **GPUs** хранит в себе информацию о видеокарте и содержит почти все те столбцы, что есть у CPU:

* Код видеокарты – ключевое поле таблицы, уникальные номер;
* Серия;
* Модель;
* Частота;
* Объем памяти;
* ID компьютера.

Таблица **Messages** хранит в себе информацию о сообщениях, отправленных пользователем. Это нужно для вывода всех сообщений в таблицу Администратора:

* ID Первого юзера;
* ID Второго юзера;
* Сообщение;
* Время отправления.

## Проектирование дизайна приложения

Основные цвета (Рисунок 2.1.) Данные цвета были выбраны в связи с ассоциацией с компьютерами. Коричневатый оттенок – цвет стола, на котором стоит компьютер. Голубой цвет – серебристая поверхность ноутбуков.



Рисунок 2.2 – Цвета приложения.

Приложение представлено одним проектом «KursProj» (Рисунок 2.2)

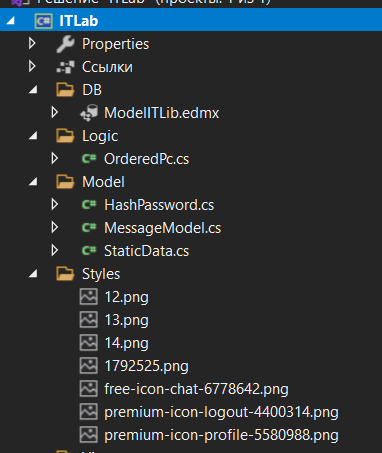
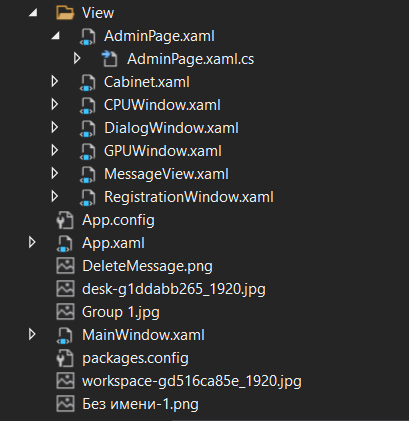
 

Рисунок 2.3 – Структура проекта

На данной диаграмме (Рисунок 2.3) представлена последовательность регистрации пользователя и заполнение формы для сбора компьютера.

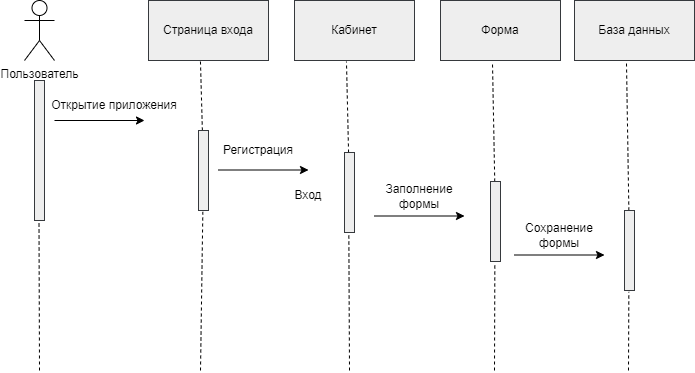


Рисунок 2.3 – Диаграмма последовательности

На данной диаграмме (Рисунок 2.4) находится наиболее общее представление функционального назначения системы.

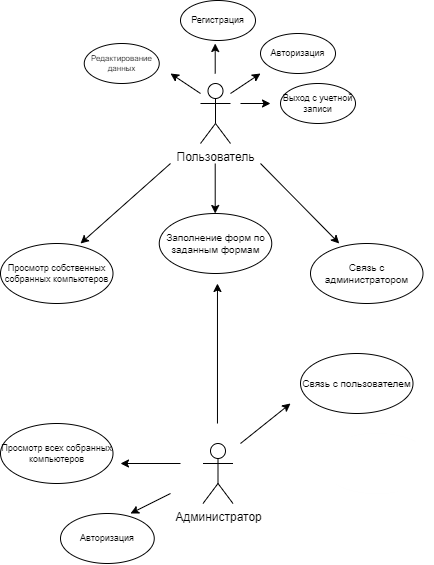


Рисунок 2.4 – Диаграмма использования

На блок-схеме (Рисунок 2.5) находится алгоритм входа пользователя в приложение с учетом авторизации и регистрации.

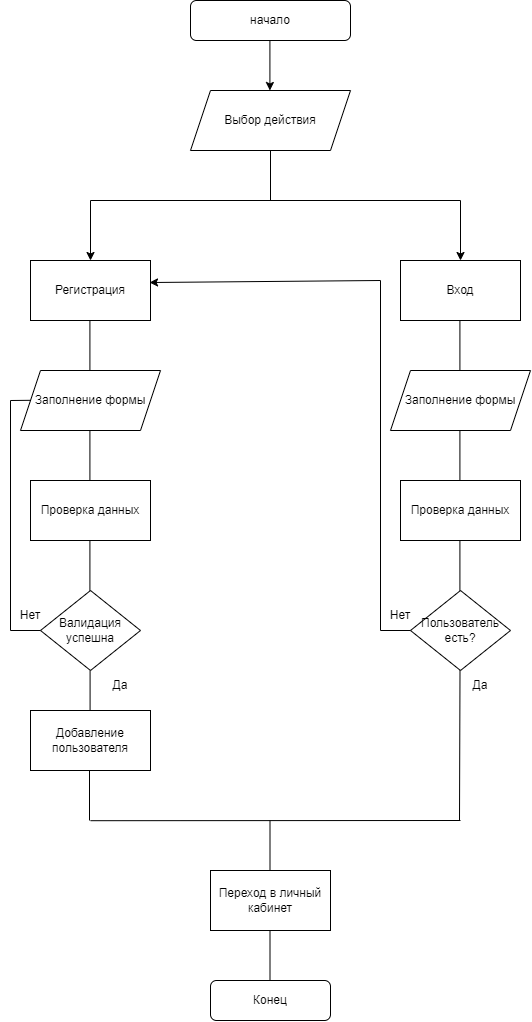


Рисунок 2.5 – Блок схема алгоритма регистрации.

На данной UML-диаграмме (Рисунок 2.6.) изображены классы программы, методы, поля и свойства классов.



Рисунок 2.6 – Диаграмма классов

# Разработка приложения

## Описание функций приложения

Данное приложение выполняет следующие функции:

* Авторизация администратора;
* Авторизация пользователей;
* Регистрация пользователей;
* Сбор компьютера по заданным формам;
* Возможность связаться с администратором;
* Просмотр всех компьютеров в кабинете администратора.

## Классы базы данных

База данных подключалась с помощью технологии Entity Framework. Этот фреймворк позволяет установить связь между проектом и базой данных.

При добавлении базы данных в проект в App.config в <connectionStrings> автоматически определяются строки подключения.

|  |
| --- |
| <connectionStrings>  <add name="ITLabDBEntities" connectionString="metadata=res://\*/DB.ModelITLib.csdl|res://\*/DB.ModelITLib.ssdl|res://\*/DB.ModelITLib.msl;provider=System.Data.SqlClient;provider connection string=&quot;data source= DESKTOP-US94KLG ;initial catalog=ITLabDB;integrated security=True;MultipleActiveResultSets=True;App=EntityFramework&quot;" providerName="System.Data.EntityClient" />  </connectionStrings> |

Листинг 3.1 – Подключение базы данных.

После подключения базы данных автоматически создаются сущности в виде классов.

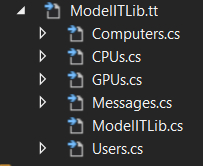


Рисунок 3.1 – Сущности базы данных.

В проекте используется подход DataBase First, так как у нас есть готовая база данных. Наша база данных подключается путем добавления в проект, как модель ADO.NET EDM.

После добавления будет создана модель по базе данных и добавлена в проект.

## Основные окна и страницы приложения

### Окно авторизации

В этом окне используется класс MainWindow. В этом окне пользователь может выполнять авторизацию. Если у пользователя нет аккаунта, он может нажать на кнопку регистрации и перейти в окно регистрации. Для того, чтобы пользователю авторизоваться, он должен корректно заполнить поля.

### Окно регистрации

Если у пользователя нет аккаунта в программе, на первоначальной странице он должен перейти на страницу регистрации.

На странице регистрации пользователь должен ввести следующие данные: ФИО, Email, Адрес, Пароль.

Поле ФИО должно состоять из трёх слов, каждое из которых начинается с заглавной буквы.

Поле Email должно содержать текст, соответствующий адресу электронной почты.

Пароль пользователя должен состоять из заглавных и строчных латинских символов, а также цифр и быть длиной 8-16 символов.

При корректном вводе данных эти данные записываются в базу данных.

Также, перед регистрацией при помощи LINQ to Entities выполняется проверка на существование такой же почты в базе данных.

После регистрации пользователь отправляется на окно входа в аккаунт.

|  |
| --- |
| Users newUser = new Users() { FullName = NameBox.Text, Email = EmailBox.Text, IsAdministrator = false, HashPassword = HashPassword.Hash(PasswordBox.Password) };  using (var db = new ITLabDBEntities())  {  db.Users.Add(newUser);  db.SaveChanges();  }  MessageBox.Show("Регистрация успешна!!"); |

Листинг 3.3 – Реализация кнопки регистрации

### Окно «Кабинет пользователя»

После успешной авторизации пользователь переходит в окно кабинет пользователя. В этом окне есть возможность смены пароля пользователя. Для этого пользователь должен ввести старый пароль и повторить новый пароль 2 раза.

Старый пароль сверяется с паролем в базе данных при помощи LINQ to Entities.

Если все поля заполнены корректно, то пароль пользователя обновляется в базе данных (листинг 3.4).

|  |
| --- |
| var passExp = new Regex(@"^(?=.{8,16}$)(?=.\*?[a-z])(?=.\*?[A-Z])(?=.\*?[0-9]).\*$");  if (!passExp.IsMatch(PasswordBox.Password)) { MessageBox.Show("В пароле должны быть: цифра, буквы нижнего и верхнего регистра, длина от 8 до 16 символов"); return; }  if (PasswordBox.Password != PasswordBoxRepeat.Password ) { MessageBox.Show("Пароли не совпадают"); return; }  user.Address = AddressBox.Text;    user.HashPassword = HashPassword.Hash(PasswordBox.Password);    user.FullName = FullnameBox.Text;  user.Email = EmailBox.Text;  db.SaveChanges();  StaticData.currentUser = user;  MessageBox.Show("Данные успешно изменены"); |

Листинг 3.4 – Смена пароля пользователя

Также пользователь может выйти из своего аккаунта и связаться с администратором (листинг 3.5).

|  |
| --- |
| using(var db = new ITLabDBEntities())  {  var admin = db.Users.FirstOrDefault(x => x.IsAdministrator == true);  var dialogWindow = new DialogWindow(admin);  dialogWindow.ShowDialog();  } |

Листинг 3.5 – Открытие окна связи с администратором

Для того чтобы пользователь смог собрать компьютер, нужно кликнуть по иконке компьютера сверху.

На это странице пользователь при помощи ComboBox выбирает необходимые комплектующие компьютера, выбирает дату сборки компьютера.

Для выбора процессора пользователь открывает окно CPU.

Для выбора видеокарты пользователь открывает окно GPU.

В окне выбора процессора пользователь заполняет данные при помощи ComboBox. Аналогичным методом заполняются данные окна видеокарты.

После заполнения всех данных и нажатии кнопки заказа компьютера, все данные записываются в базу данных (листинг 3.6).

|  |
| --- |
| using (var db = new ITLabDBEntities())  {  var newComputer = new Computers() { UserID = StaticData.currentUser.ID, TypeComputer = TypeCombo.Text, SizeRAM = RamSizeCombo.Text, TypeRAM = RamTypeCombo.Text, SizeHardDisk = Convert.ToInt32(DiskSizeCombj.Text), TypeHardDisk = DiskTypeCombo.Text, TimeOfOrder = DateTime.Parse(DateCombo.Text) };  db.Computers.Add(newComputer);  db.SaveChanges();  var selComputer = db.Computers.FirstOrDefault(x => x.ID == newComputer.ID);  cpu.ComputerId = selComputer.ID;  gpu.ComputerId = selComputer.ID;  db.GPUs.Add(gpu);  db.CPUs.Add(cpu);  db.SaveChanges();  MessageBox.Show("Компьютер успешно заказан");  } |

Листинг 3.6 – Запись данных сборки в базу данных

Если же пользователь до занесения я компьютера в базу данных решил отказаться от заказа, то для очистки полей он может нажать на иконку мусорки. После нажатия все поля будут очищены. Реализация этой кнопки представлена на листинге 3.7.

|  |
| --- |
| TypeCombo.Text = "";  RamTypeCombo.Text = "";  RamSizeCombo.Text = "";  DiskSizeCombj.Text = "";  DiskTypeCombo.Text = "";  DateCombo.Text = "";  cpu = new CPUs();  gpu = new GPUs();  cpu.Series = "";  gpu.Series = ""; |

Листинг 3.7 – Очистка заполненных полей

После выполнения заказа, на главной странице кабинета, в ListView отображаются данные о всех сборках пользователя.

### Диалоговое окно связи администратора с пользователем

Для связи с администратором пользователь может с кабинета открыть диалоговое окно связи. После его открытия окна инициализируются все сообщения, существующие в базе данных. Для отправления сообщения требуется ввести его в TextBox, после чего нажать кнопку отправки сообщения (листинг 3.4). После этого сообщение запишется в базе данных и отобразится при открытии у другого участника переписки.

|  |
| --- |
| if (TextBoxMessage.Text == null || TextBoxMessage.Text == "") { return; }  Messages message = new Messages()  {  UserID1 = StaticData.currentUser.ID,  SendTime = DateTime.Now,  Text = TextBoxMessage.Text,  UserID2 = user.ID,  };  using (var db = new ITLabDBEntities())  {  db.Messages.Add(message);  db.SaveChanges();  }  TextBoxMessage.Text = "";  MessagesList.Add(new MessageModel(message)); |

Листинг 3.4 – Кнопка отправки сообщения

### Окно администратора

В этом окне производится вывод заказов и клиентов, существующих в базе данных при помощи ListView. Также реализована аналогичная возможность обратной связи с клиентом. Для этого администратор должен выбрать необходимого клиента, а после нажать кнопку «Написать сообщение клиенту».

Кроме этого, у администратора реализована кнопка выхода из аккаунта, после нажатия которой будет открыто окно авторизации (листинг 3.5).

|  |
| --- |
| StaticData.currentUser = null;  App.Current.MainWindow.Hide();  App.Current.MainWindow = new MainWindow();  App.Current.MainWindow.Show(); |

Листинг 3.5 – Кнопка выхода из аккаунта

# Руководство по установке и использованию

При запуске приложения открывается окно авторизации (Рисунок 5.1.). Для продолжения требуется ввести почту и пароль. Для входа нужно нажать на кнопку «Войти», если данные верные, то открывается страница каталога, если нет, выводятся надписи с указанием ошибки.

Если пользователь не зарегистрирован, то требуется заполнить поля имя и пароль и нажать кнопку «Зарегистрировать».

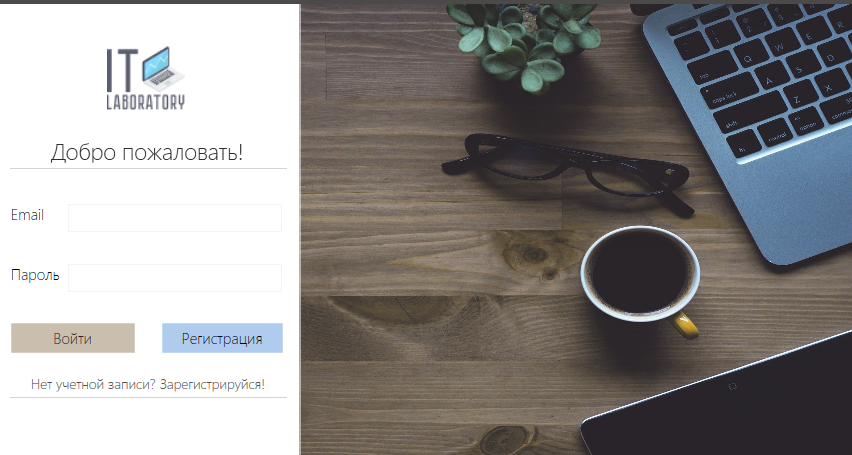


Рисунок 4.1 – Окно авторизации.

После входа пользователь попадает на главную страницу, на которой видит основные навигационные кнопки и сам каталог (Рисунок 5.2).

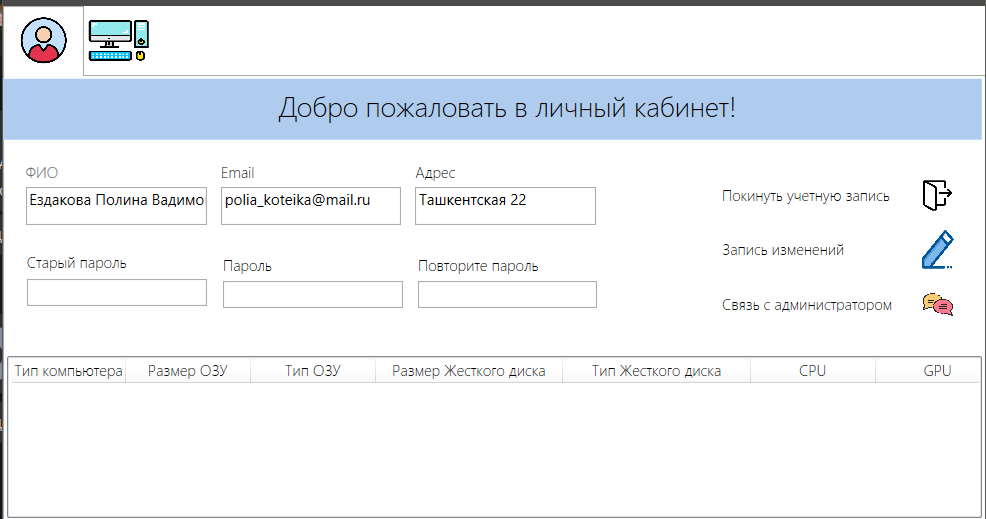


Рисунок 4.2 – Главная страница (кабинет пользователя).

Далее идет окно с формами, где пользователь и собирает свой компьютер.

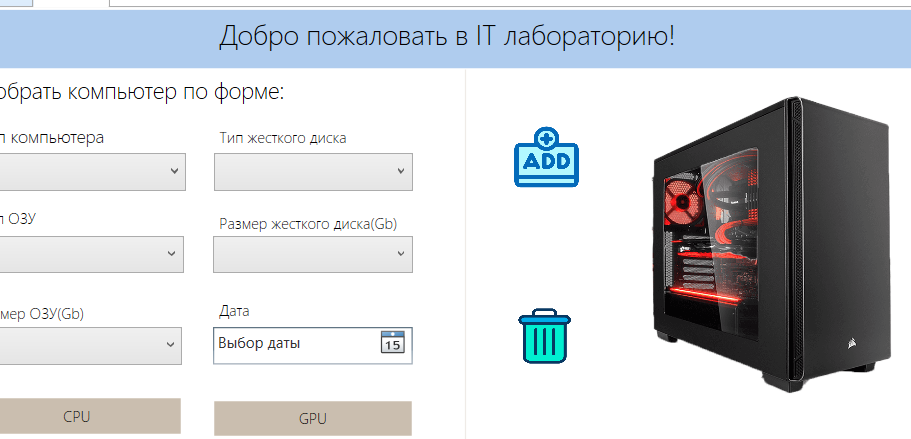


Рисунок 4.3 – Страница с формой.

Пользователь должен заполнить все поля. Без этого форма не сохранится и вылетит ошибка. Если пользователь передумал собирать компьютер, он может воспользоваться кнопкой «Удалить». Тогда все поля очистятся.

Когда человек полностью заполнит все формы, он нажимает на добавление и форма отправляется прямиком в личный кабинет пользователя в таблицу.

Также в личном кабинете предусмотрена кнопка связи с администратором. Это нужно для того, чтобы пользователь задавал интересующие его вопросы, а также имел возможность проконсультироваться со специалистом.

# Тестирование программного средства

Для того, чтобы удостовериться, что программное средство работает как нужно, произведем некоторые действия. Например, ввод не валидных данных в строки ввода в окне Авторизации и Регистрации.

Сперва, возьмем окно регистрации. В поле email введем некорректные данные и посмотрим, как отреагирует программа.

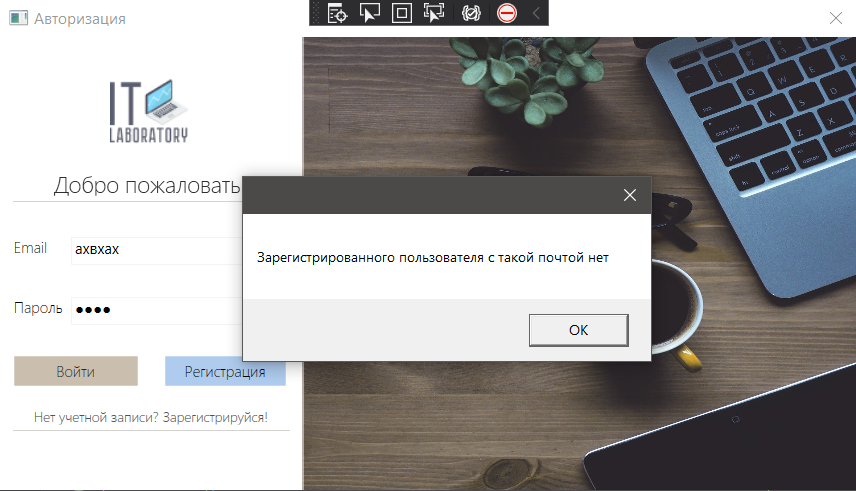


Рисунок 5.1 – Ввод некорректных данных в поля ввода.

Теперь, введем правильную почту, но неправильный пароль и посмотрим, как отреагирует программа.

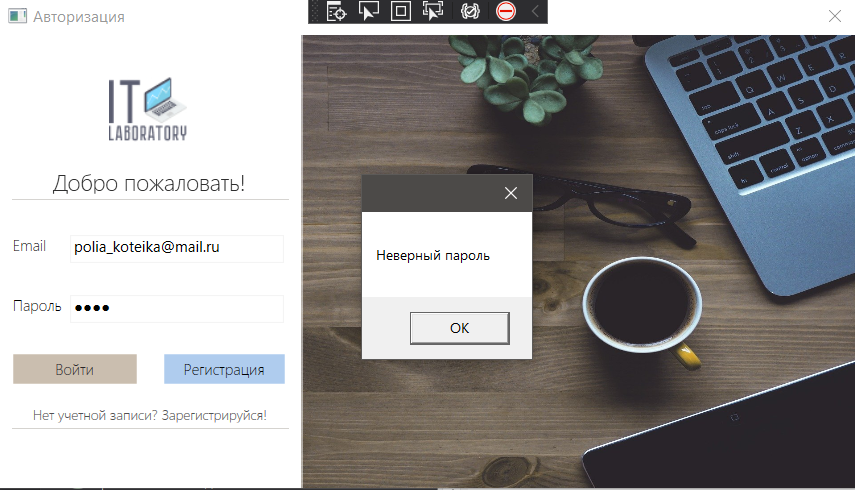


Рисунок 5.2 – Ввод некорректных данных в поле пароля.

Теперь перейдем к окну Регистрации и посмотрим, как оно отреагирует на неправильно введенные данные.

Сразу вводим в строку ФИО не валидные данные.

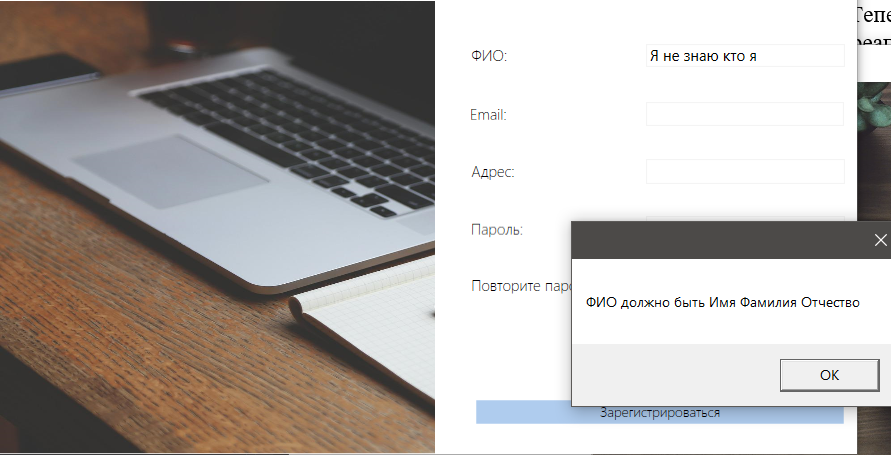


Рисунок 5.3 – Ввод некорректных данных в поле ФИО.

Реакция приложения положительная – ошибка не вылетает, так как она обработана. Далее перейдем к почте.

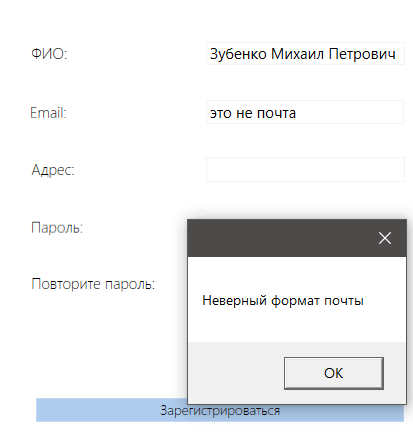


Рисунок 5.4 – Ввод некорректных данных в поле почты.

Ошибок выявлено не было, так как все ошибки были исправлены во время написания кода.

Сделаем вывод: Регистрация и Авторизация оказались полностью валидными. Это значит, что ошибки выявить не удалось.

# Заключение

Итогом выполнения курсового проекта стало приложение «IT лабораторияпо». В ходе выполнения работы были улучшены и закреплены навыки работы с языком разметки XAML и языком программирования С#. Также были закреплены знания по разработке и использованию баз данных MS SQL и фреймворка Entity. Для создания и реализации проекта были использованы не только знания, полученные на предмете «Объектно-ориентированное программирование», но и навыки, полученные в ходе подготовки в области дизайна. Благодаря использованию WPF, удалось реализовать приложение, которое в большей части соответствует изначальному макету.

В ходе разработки были пройдены основные этапы создания приложения, такие как макетирование, анализ, реализация и тестирование. Разработанный проект имеет перспективы для внедрения дополнительных функций и возможностей, что позволяет ему сохранить свою актуальность на протяжении более длительного времени.

# Список используемых источников

1. Professor WEB [Электронный ресурс]. – ЭУ. – Режим доступа: <https://professorweb.ru/my/WPF/UI_WPF/level6/UI_WPF_index.php>.

2. Stackowerflow [Электронный ресурс]. – DispatcherTimer. – Режим доступа: https://ru.stackoverflow.com/questions/693150/%D0%A0.

3. WPF tutorial [Электронный ресурс]. – Режим доступа: <https://www.wpf-tutorial.com/>.

5. Metanit [Электронный ресурс]. – Работа с EF. – Режим доступа: <https://metanit.com/sharp/wpf/19.3.php>.

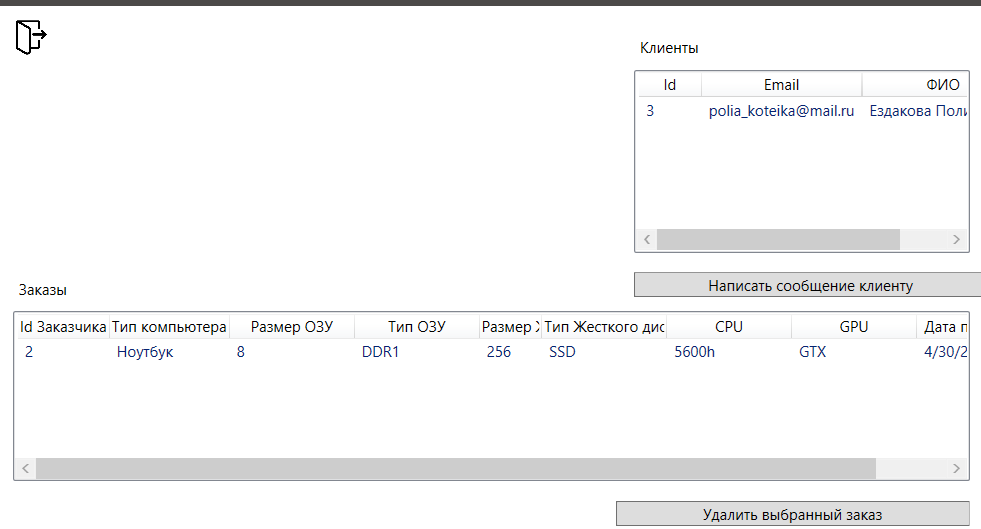
6. «Руководство по Entity Framework 6» [Электронный ресурс] - <https://metanit.com/sharp/entityframework/>. Дата доступа: 15.05.2022

7.Документация Microsoft SQL [Электронный ресурс] - <https://docs.microsoft.com/en-us/sql/?view=sql-server-ver16>. Дата доступа: 02.05.2022

8. Metanit [Электронный ресурс]. – Руководство по WPF. – Режим доступа: <https://metanit.com/sharp/wpf/>.

# Приложение А

Страница администратора

****