МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Программной инженерии

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**КУРСОВОГО ПРОЕКТА:**

по дисциплине «Объектно-ориентированные технологии программирования»

Тема Программное средство «CoffeeCupBy»

Исполнитель

студент 2 курса 5 группы Савченко Владислав Юрьевич

(Ф.И.О.)

Руководитель

ассистент Северенчик Н.А.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Председатель Северенчик Н.А.

(подпись)

Минск 2021

**Содержание**

[**Введение** 3](#_Toc514788131)

[**1. Постановка задачи** 5](#_Toc514788132)

[**1.1 Актуальность решаемой задачи** 5](#_Toc514788133)

[**1.2 Обзор и анализ прототипов** 5](#_Toc514788134)

[**1.3 Средства разработки** 6](#_Toc514788135)

[**2. Разработка архитектуры проекта** 7](#_Toc514788136)

[**2.1 Структура проекта** 7](#_Toc514788137)

[**2.2 Проектирование классов приложения** 8](#_Toc514788138)

[**2.3 Проектирование базы данных** 10](#_Toc514788139)

[**3. Руководство пользователя** 12](#_Toc514788140)

[**4. Тестирование** 19](#_Toc514788141)

[**Список использованных источников** 22](#_Toc514788142)

[**Приложение А** 23](#_Toc514788143)

# **Введение**

Основная цель данного курсового проекта – это разработка настольного программного средства «CoffeeCupBy». Данное средство должно предоставлять возможность пользователю совершать покупки кофе со скидкой, а так же данное приложение будет служить компаниям вкачестве рекламной площадки. Все это должно быть реализовано в максимально простой и удобной форме, чтобы пользователь не испытывал дискомфорт при работе с программным средством.

Был произведен анализ требований к проекту, проектирование, дизайн, а затем тестирование. Любое программное средство должно быть конкурентоспособно. Именно для этого необходимо провести анализ аналогов и выявление их сильных, а также слабых сторон. При разработке данного приложения, должны быть учтены и исправлены слабые стороны приложений-аналогов, а также позаимствовано максимально возможное количество сильных сторон приложений-аналогов.

Одним из главных требований любого приложения – максимальное возможное привлечение аудитории на постоянной основе и получение прибыли. Для этого приложение должно быть удобным и полностью функциональным.

Для корректной реализации данного приложения, необходимо наладить работу базы данных, должна быть правильна спроектирована диаграмма классов, которая бы не нарушала законы логики и была максимально удобной для проектирования и разработки приложения.

Управление данной программой должно быть интуитивно понятным и удобным пользователю. При разработке были использованы несколько наиболее подходящих шаблонов проектирования ПО.

Структура приложения: приложение, написанное на С#. Оно устанавливается на ПК и представляет собой интерфейс.

ПК приложение для оптимальной работы требует:

* ОС Windows 10;

установленный .Net framework 5.0 или более поздней версии.

# **1. Постановка задачи**

## **1.1 Актуальность решаемой задачи**

Объектом разработки является ПО «CoffeeCupBy». Данное программное средство представляет из себя настольное приложение, где пользователь сможет активировать заказы, осуществлять поиск по карте, находить популярные кофейни, смотреть статистику своих заказов, смотреть список заказов, добавлять понравившуюся кофейню в избранные, для кофейнь доступны такие функции, как создать кофейню, просмореть свои кофейни, просмотр статистики и просмотр популярных кофейнь.

Данная тема актуальна, так как позволяет людям сэкономить на приобретении кофе и позволяет узнать о новых кофейнях.

В век информационных технологий достаточно иметь любой гаджет с доступом в интернет, чтобы иметь возможность зайти в приложение и активировать свой напиток. Данные приложения дают возможность просмотреть все доступные кофейни, которые стали на нашу площадку. Таким образом, приложение выступает не только в роли ресурса для активации скидки, но и рекламной платформой для новых кофейнь.

В ходе выполнения курсового проекта необходимо разработать приложение, в котором будет возможность просмотреть все доступные кофейни, выбрать подходящую, просмотреть информацию о напитках, которые есть у данной кофейни, производить поиск по карте, просмотр популярных кофейнь, просмотр своих заказов, добавление и просмотр своих любимых кофейнь, а так же для самих кофейнь: добавлять свои кофейни и их товары, просмотр своих кофейнь, просмотр популярных кофейнь и статистика по заказанным напиткам. Для этого необходимо выяснить и учесть требования к содержимому приложения, а также разработать и спроектировать удобный интерфейс для пользователей.

## **1.2 Обзор и анализ прототипов**

На сегодняшний день существуют приложения для предоставления скидок на кофе, но наш продукт отличается от аналогов со всех сторон и сейчас попробуем понять как.

В качестве объекта обзора и анализа взято приложение «Slivki.by» для платформы Android. Представление приложения из каталога Google Play изображено на рисунке 1.1.

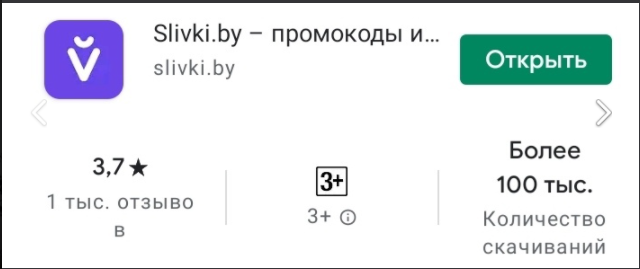


Рисунок 1.1 — Приложение «Slivki.by»

Принимая во внимание представленный на рисунке 1 прототип, а также учитывая то, что был дополнительно произведен анализ ряда программ для работы со скидками на кофе, можно сказать о следующих функциональных сходствах программных средств:

* возможность просмотра и контроля списка покупок,
* активация купонов,
* добавление в избранное,

Следует отметить, что в большинстве случаев визуальное представление списковых структур данных, которые используются для организации множества кофейных напитков, представлены на рисунке 1.2.

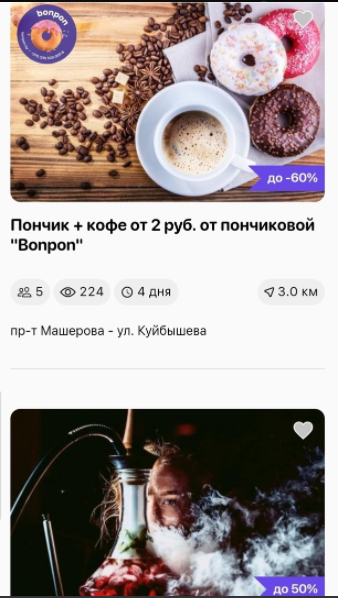


Рисунок 1.2 — Визуальная организация множества рецептов

Приложение «CoffeeCupBy» также функционально может выполнить основную задачу, которая ставится перед всеми программами данного класса, но оно более функционально и так же выступает в качестве рекламной площадки для кофейнь, это очень поможет только открывшимся заведениям получить приток новых клиентов и расширить свою клиентскую базу, им только нужно стать на наше приложение и дать скидки пользователям – это главный критерий, по которому они могут зарегистрироваться в нашем приложении и получить новый клиентов, а для уже “ гигантов” компаний, делать больше продаж за счёт нашего приложения.

## **1.3 Средства разработки**

При разработке приложения были использованы:

* интегрированная среда разработки Microsoft Visual Studio 2019;
* программная платформа .NET Framework 5.0;
* язык программирования C#;
* расширяемый язык разметки XAML;
* технология WPF;
* Entity Framework Core;
* Web Api;
* JWT;
* DI;
* Identity;

*C#* — объектно-ориентированный язык программирования. Разработан в 1998 — 2001 под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework.

*XAML* — это декларативный язык разметки. С точки зрения модели программирования .NET Framework язык XAML упрощает создание пользовательского интерфейса для приложения .NET Framework.

Entity Framework Core — представляет собой объектно-ориентированную, легковесную и расширяемую технологию от компании Microsoft для доступа к данным.

Web Api –– это программный интерфейс, состоящий из одной или нескольких общедоступных конечных точек для определенной системы сообщений запрос-ответ, обычно выраженной в JSON или XML.

JWT –– это открытый стандарт (RFC 7519) для создания токенов доступа, основанный на формате JSON. Как правило, используется для передачи данных для аутентификации в клиент-серверных приложениях. Токены создаются сервером, подписываются секретным ключом и передаются клиенту, который в дальнейшем использует данный токен для подтверждения своей личности.

DI –– это шаблон проектирования, используемый для реализации IoC. Он позволяет создавать зависимые объекты вне класса и предоставляет эти объекты классу различными способами. Используя DI, мы перемещаем создание и привязку зависимых объектов за пределы класса, который от них зависит.

Identity –– представляет встроенную в ASP.NET систему аутентификации и авторизации. Данная система позволяет пользователям создавать учетные записи, аутентифицироваться, управлять учетными записями

# **2. Разработка архитектуры проекта**

## **2.1 Структура проекта**

Структура проекта представлена на рисунке 2.1.

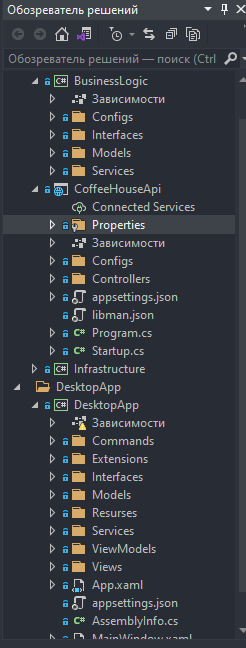


Рисунок 2.1 — Структура проекта

Приложение разделено на 3 части согласно архитектурному паттерну MVC(Model View Controller), за Model часть отвечает 2 проекта:

1. BusinessLogic – здесь находится вся бизнес логика на стороне сервера, только этот проект может отправлять запросы к Infrastructure .
2. Infrastructure – здесь находится вся логика работы с БД Sqlite,

За Controller отвечает проект CoffeeHouseApi, который получает запрос в виде HTTP запросов от DesktopApp(десктопное приложение на WPF) и обращается к проекту BusinessLogic, получает ответ и отправляет ответ обратно DesktopApp.

За View отвечает DesktopApp, с которым взаимодействует пользователь, приложение написано на WPF + C# с архитектурным паттерном MVVM(ModelViewViewModel). В папке Services находятся классы в которых прописаны методы для обращения к CoffeeHouseApi.

## **2.2 Проектирование классов приложения**

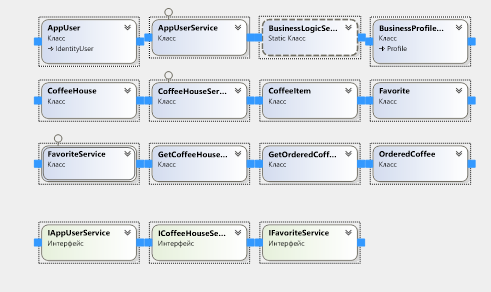


Рисунок 2.3 – Диаграмма классов BusinessLogic

Класс FavoriteService содержит методы для любимых кофейнь.

Диаграмма классов DesktopApp представлена на рисунке 2.4.

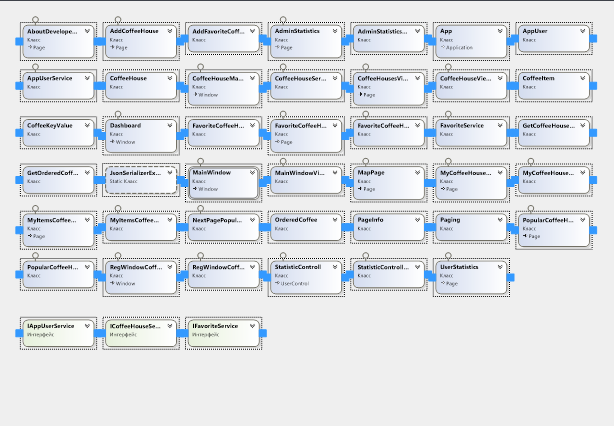


Рисунок 2.4 – Диаграмма классов DesktopApp

.

Диаграмма классов Infrastructure представлена на рисунке 2.5.

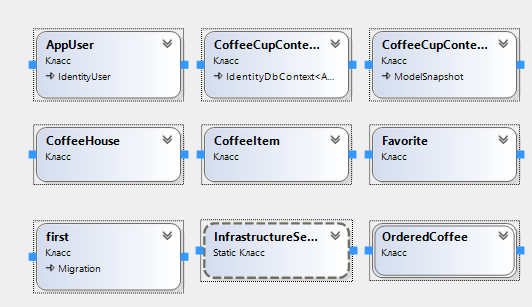


Рисунок 2.5 – Диаграмма классов для Infrastructure

## **2.3 Проектирование базы данных**

Диаграмма базы данных, спроектированной в ходе разработки приложения представлена на рисунке 2.7.

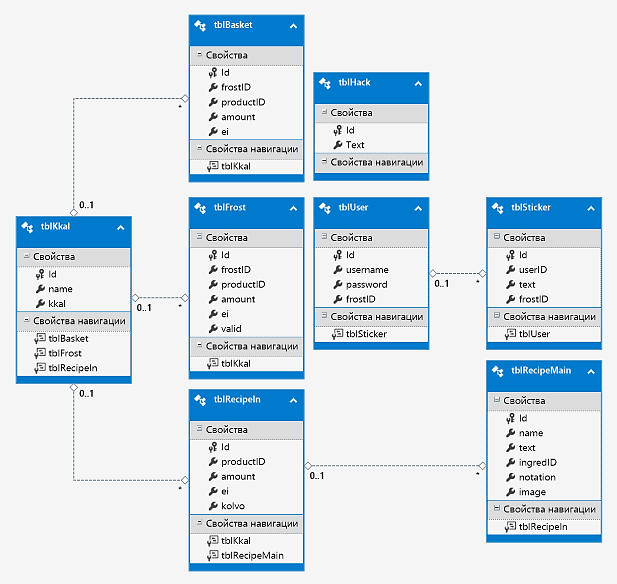


Рисунок 2.7 – Диаграмма базы данных

# **3. Руководство пользователя**

После запуска приложения пользователю представляется окно входа в приложение, представленное на рисунке 3.1.

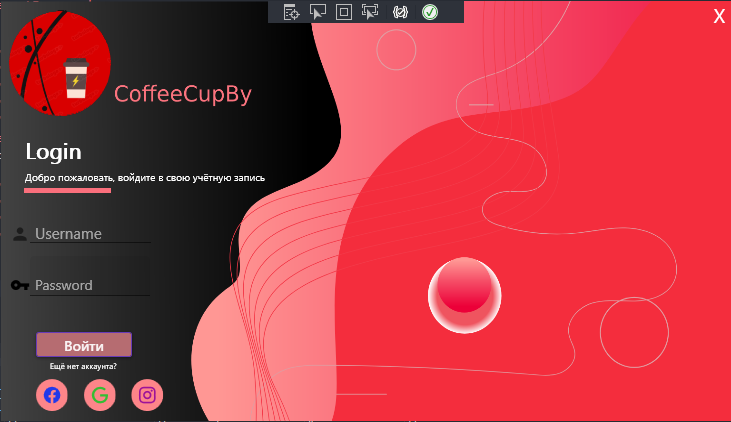


Рисунок 3.1 — Окно входа в аккаунт

Если пользователь ещё не имеет своего аккаунта, то по нажатию соответствующей кнопки можно перейти к окну регистрации (рис. 3.2).

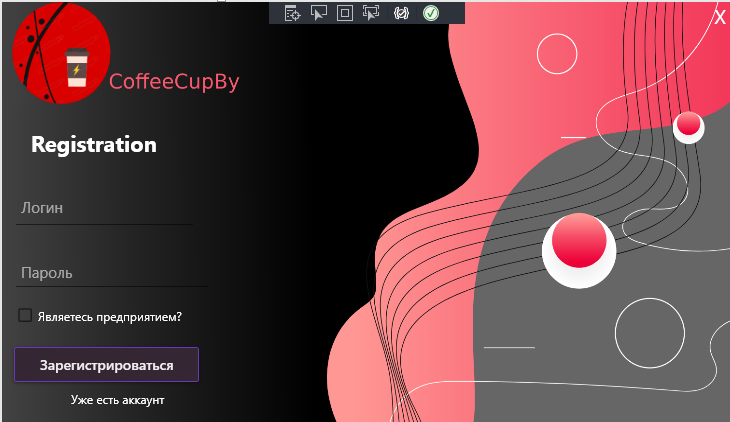


Рисунок 3.2 – Окно регистрации

При успешном входе в аккаунт появляется страница «Dashboard» или «CoffeeHouseMainWindow» в зависимости от того, являетесь ли вы кофейней или обычным пользователем. Данные окна представлены на рисунке 3.3.

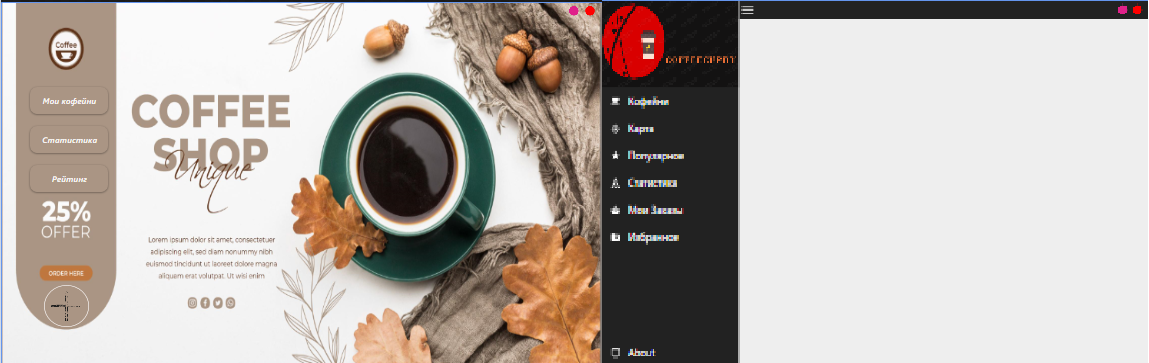


Рисунок 3.3 – Окно «CoffeeHouseMainWindow/ Dashboard»

В меню слева можно выбрать у обычного пользователя страницу «Кофейни», на которой представлены все кофейни, которые есть в приложении, функция активации купона и добавления в избранное, а так же наименование и адрес кофейни.

Данная страница представлена на рисунке 3.4.

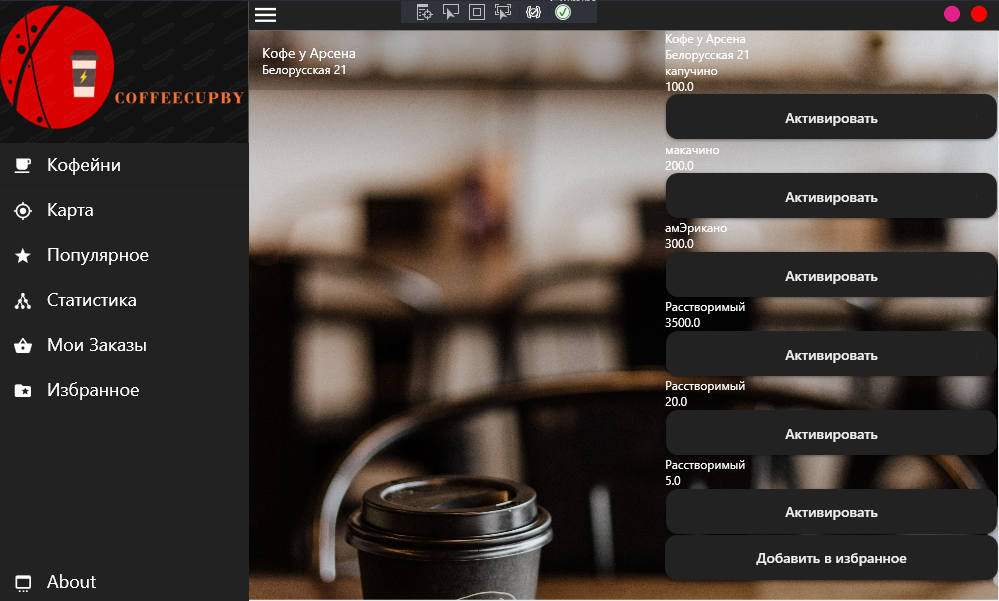


Рисунок 3.4 – страница «Кофейни»

Если нажать на вкладку Карта, откроется страница с картой и списком всех кофейнь, которые есть в приложении. Страница представлена на рисунке 3.5.

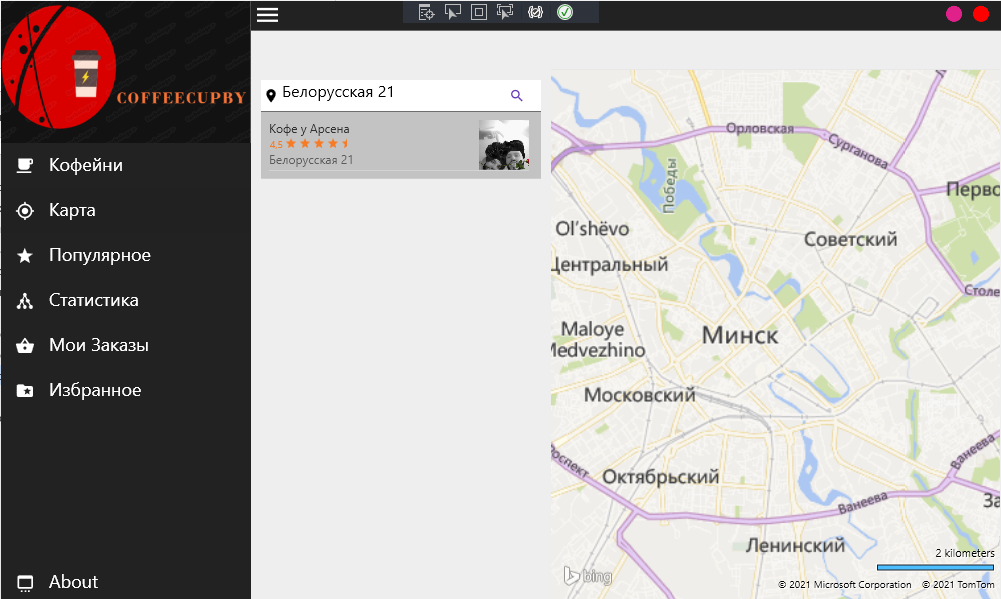


Рисунок 3.5 – страница с картой

Для просмотра общей информации о приложении реализована страница «About». Страница представлена на рисунке 3.6.

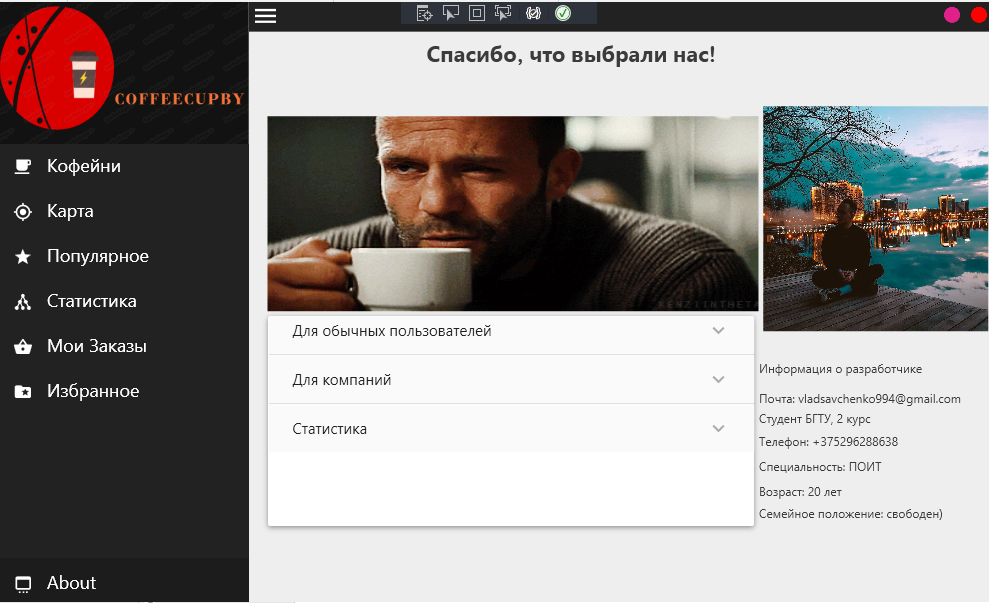


Рисунок 3.6 – О программе

А так же в панели «Меню» организован переход на следующие страницы:

* Популярное (рис. 3.7),
* Статистика (рис. 3.8),
* Мои заказы (рис. 3.9),
* Избранное (рис. 3.10),



Рисунок 3.7 – Страница Популярное

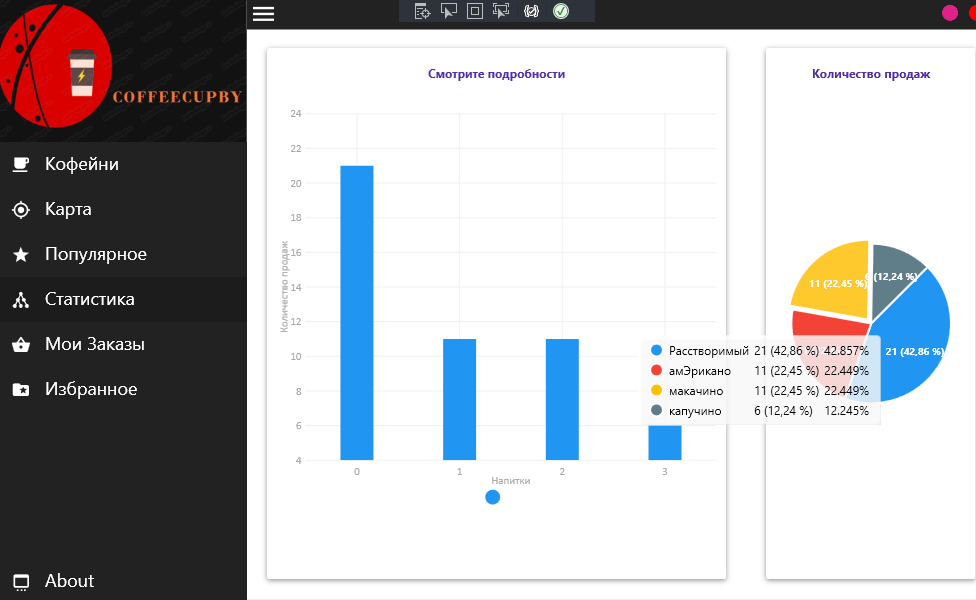


Рисунок 3.8 – Просмотр Статистики

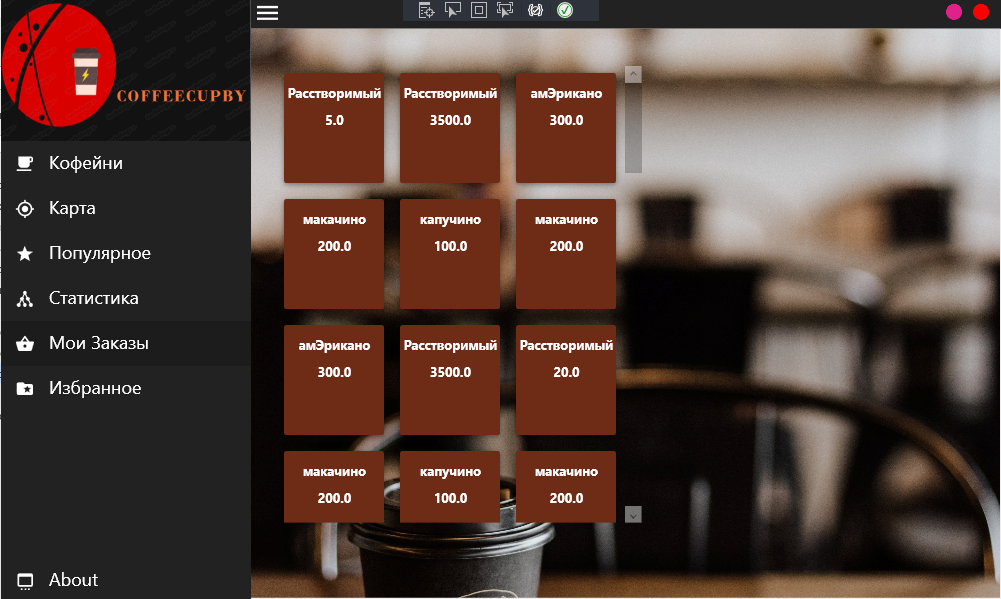


Рисунок 3.9 – Мои заказы

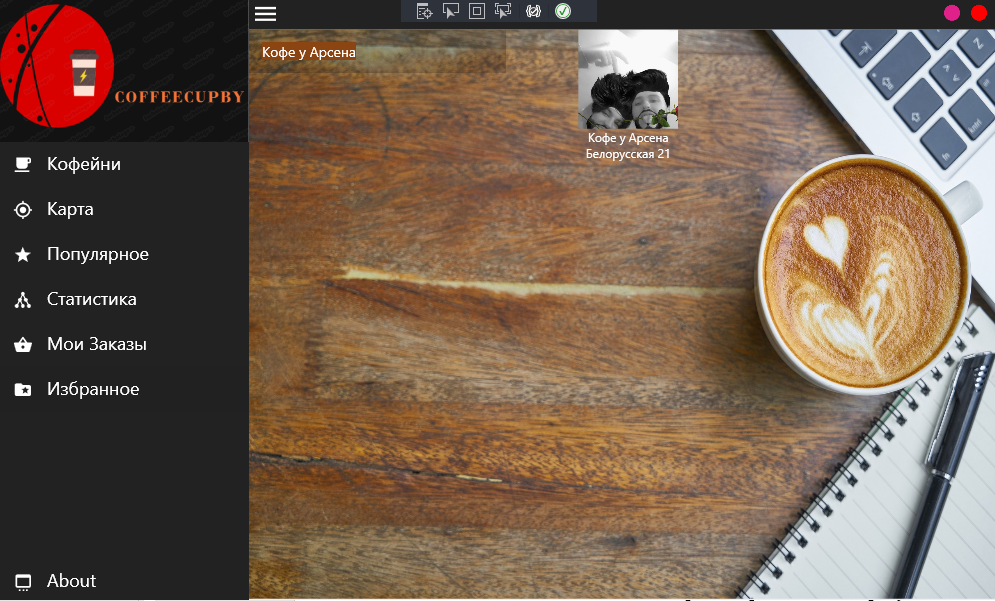


Рисунок 3.10 – Избранное

Для кофейнь отдельные окна:

* Мои кофейни (рис. 3.11)
* Статистика (рис. 3.12)
* Рейтинг (рис. 3.13)
* Добавить кофейню (рис. 3.14)

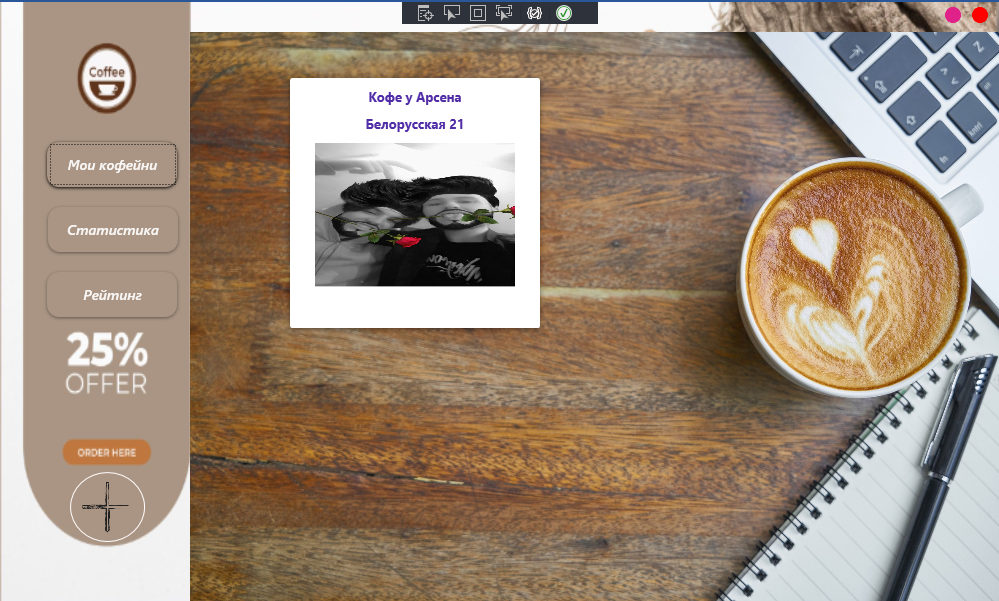


Рисунок 3.11 – Мои кофейни

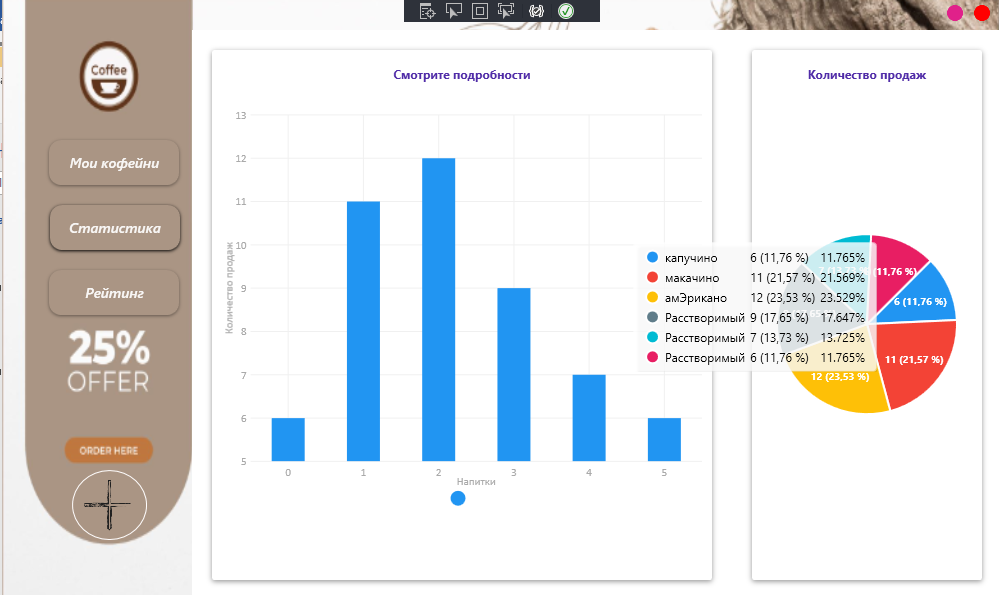


Рисунок 3.12 – Статистика



Рисунок 3.13 – Рейтинг

Для добавления кофейни открываем соответствующую страницу (рис. 3.14). В поле ввода текста необходимо ввести набор символов.

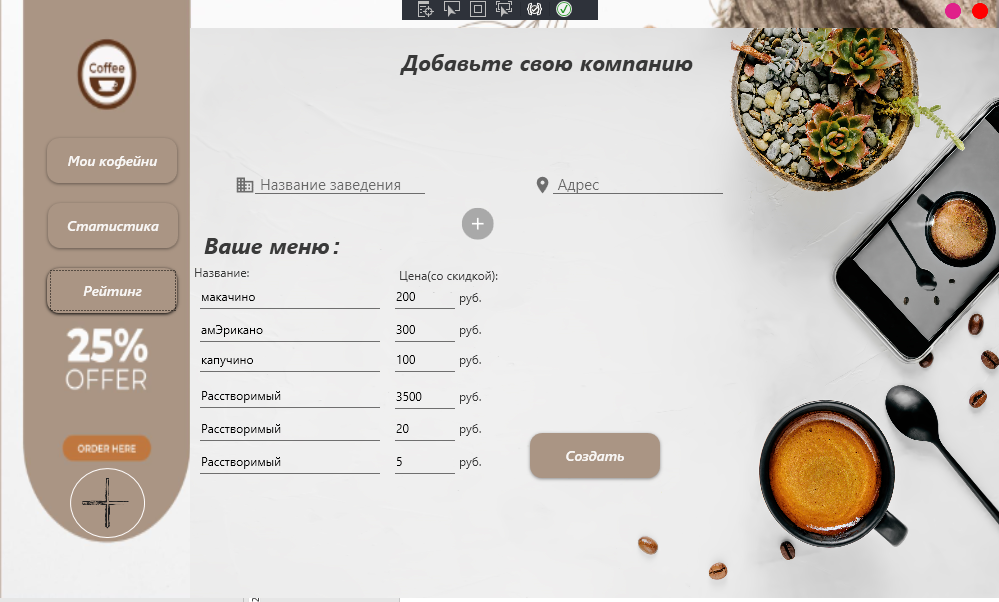


Рисунок 3.14 – Добавление кофейни

# **4. Тестирование**

При тестировании данного приложения были проведены следующие тесты на ситуации, которые могли бы привести к ошибке:

* тесты авторизации и регистрации;
* тест изменения данных;

В момент регистрации возможна ситуация, когда пользователь вводит недостаточной безопасности пароль. Обработка данного исключения продемонстрирована на рисунке 4.1.

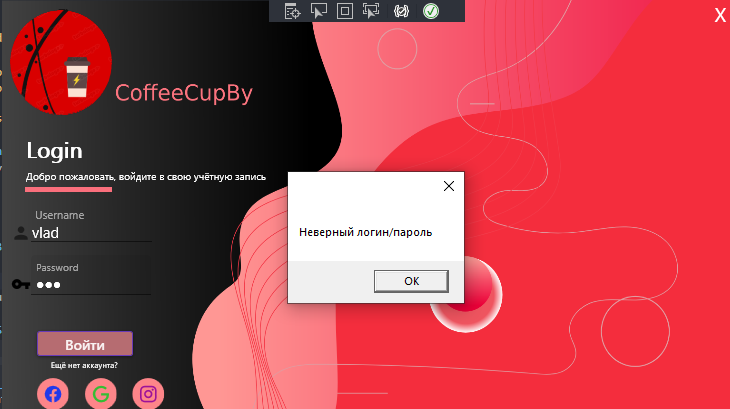
 

Рисунок 4.1 – Тестируем регистрацию и вход

В момент авторизации возможна ситуация, когда пользователь вводит неверные данные пользователя или пользователя, который уже зарегистрирован. Обработка данного исключения продемонстрирована на рисунке 4.1.

Когда мы создаём кофейню, пользователь может оставить поле для ввода названия или адреса пустым. Обработка данного исключения продемонстрирована на рисунке 4.2.

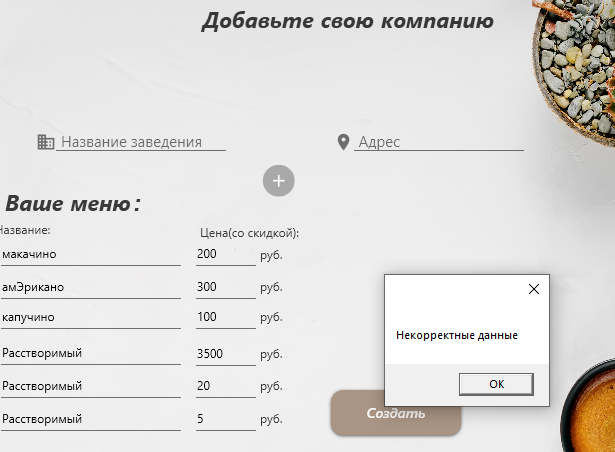


Рисунок 4.2 – Тестируем изменения записей

Если пользователь при добавлении продукта в холодильник или базу данных не заполнит все поля и нажмет кнопку, то система сообщит о некорректном значении (рис. 4.3).

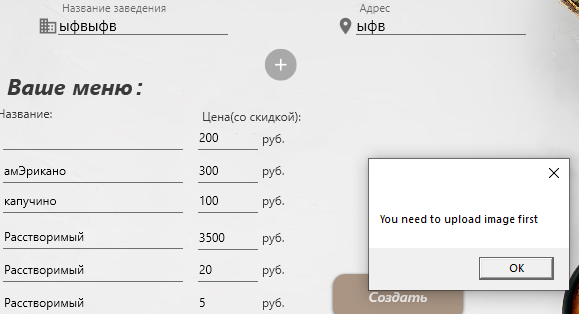


Рисунок 4.3 – Тестируем добавление кофейни

**Заключение**

В ходе выполнения курсовой работы было разработано приложение «CoffeeCupBy», являющееся для обычных пользователей приложением для скидок на кофе, а для кофейнь рекламной площадкой. Во время выполнения данной курсовой работы было изучено немало теоретического материала, а также просмотрено и разобрано большое количество уже готовых решений тех или иных задач.

В ходе выполнения курсовой работы было выполнено следующее:

- разработка базы данных для хранения информации;

- разработка архитектуры приложения;

- разработка функциональных части приложения;

- разработка пользовательского интерфейса;

- написание исходного кода приложения;

- тестирование приложения.

В соответствии с полученным результатом работы программы можно сделать вывод, что разработанная программа работает, верно, а требования технического задания выполнены в полном объеме.

**Список использованных источников**

1. Пацей, Н.В. Курс лекций по языку программирования С# / Н.В. Пацей. – Минск: БГТУ, 2016. – 175 с.
2. Пацей, Н.В. Технология разработки программного обеспечения / Н.В. Пацей. – Минск: БГТУ, 2016. – 129 с.
3. MSDN сеть разработчиков в Microsoft [Электронный ресурс]
4. METANIT.COM Сайт о программировании [Электронный ресурс]
5. ProfessorWeb .NET & Web Programming [Электронный ресурс]

# **Приложение А**

|  |
| --- |
| public static IServiceCollection AddBusinessLogicServiceCollection(this IServiceCollection services)  {  services.AddScoped(typeof(IAppUserService), typeof(AppUserService));  services.AddScoped(typeof(ICoffeeHouseService), typeof(CoffeeHouseService));  services.AddScoped(typeof(IFavoriteService), typeof(FavoriteService));  services.AddAutoMapper(typeof(BusinessProfileMappingProfile));  return services;  } |

Листинг 1 — Стиль использования DI на стороне сервера

|  |
| --- |
| public string CreateToken(IEnumerable<Claim> claims)  {  var secretsBytes = Encoding.UTF8.GetBytes(\_configuration["TokenSecret"]);  var key = new SymmetricSecurityKey(secretsBytes);  var algorithm = SecurityAlgorithms.HmacSha256;  var signingCredentials = new SigningCredentials(key, algorithm);  var token = new JwtSecurityToken(  \_configuration["TokenIssuer"],  \_configuration["TokenAudience"],  claims,  notBefore: DateTime.Now,  expires: DateTime.Now.AddDays(7),  signingCredentials);  return new JwtSecurityTokenHandler().WriteToken(token);  } |

Листинг 2 — Создание токена доступа