

Министерство образования Республики Беларусь

Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ  
УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования  
Кафедра инженерной психологии и эргономики  
Дисциплина: Конструирование программного  
обеспечения

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА  
к курсовой работе  
на тему

ВЕБ-СЛУЖБА КАФЕ «МАКДОНАЛЬДС»

БГУИР КП 6 - 05 - 06 12 01 006 ПЗ

Выполнил: студент группы 310901 Богущ А. В.  
Проверила: Болтак С. В.

Минск 2025





## СОДЕРЖАНИЕ

Введение.....	5
1 Анализ предметной области .....	6
1.1 Обзор аналогов .....	6
1.2 Постановка задачи.....	7
2 Проектирование программного средства .....	9
2.1 Структура программы.....	9
2.2 Проектирование интерфейса программного средства .....	9
2.3 Проектирование функционала программного средства.....	15
3 Разработка программного средства.....	16
3.1 Физическая структура.....	16
3.2 Описание разработанных модулей .....	16
4 Тестирование программного средства .....	18
5 Применение программы .....	19
5.1 Руководство пользователя.....	19
Заключение .....	26
Список использованных источников .....	28
Приложение А. Исходный код программы .....	29

## ВВЕДЕНИЕ

Курсовая работа «Веб-служба кафе «Макдональдс»» была предназначена для упрощения и ускорения процесса обслуживания клиентов. С ее помощью клиентам будет легче и удобнее сделать заказ, так как при использовании программы для клиента показывается все актуальное меню товаров, включая особые предложения и акции. Еще одним преимуществом использования приложения «Веб-служба кафе «Макдональдс»» является исключение социального взаимодействия клиентов с кассирами, так как пользователь может сделать заказ онлайн, что особенно актуально для тех, кто чувствует дискомфорт от общения с незнакомыми людьми.

При оплате через «Веб-службу кафе «Макдональдс»» информация о заказе автоматически передается по сети кассиру, который должен подтвердить или отклонить заказ. После подтверждения заказ переходит на кухню, и повара сразу приступают к готовке. Когда товар будет готов, кассир уведомляет об этом пользователя. Информация о наличии продуктов в программе всегда является актуальной, так как она редактируется администратором в режиме реального времени.

В сфере общественного питания первые веб-службы начали появляться в 1990-е годы. В Интернете нет доступной информации о том, какое кафе первым выпустило свою веб-службу, потому что в то время Интернет стал доступен более широкой публике, и многие заведения начали экспериментировать со своими веб-сайтами. Можно предположить, что одними из первых таких кафе являются «Старбакс», «Сабвей», «Бургер Кинг», «Макдональдс», «КФС», «Вендис». В 2000-х годах подобные веб-службы начали создавать и остальные существующие кафе из-за улучшения веб-технологий, а также всемирному распространению сети Интернет. Такие системы обслуживания клиентов позволяли существенно сократить очереди, оптимизировать процесс заказов и увеличить их количество [1].

Сегодня веб-службы для кафе стали неотъемлемой частью ресторанного бизнеса, особенно в ресторанах быстрого питания, где важны скорость и удобство. Они помогают снизить нагрузки на персонал, сократить очереди и повысить комфорт при заказе еды клиентами.

Целью курсовой работы «Веб-служба кафе «Макдональдс»» является облегчение и повышение качества обслуживания заказчиков, доступность, сокращение очереди в кафе, снизить загруженность кассиров, свести шанс допущения ошибки в заказе к минимуму.

Итогом курсовой работы должна получиться веб-служба, включающая актуальное меню товаров для заказа пользователя, оплату и историю заказов. После оплаты заказ пользователя должен подтвердить кассир кафе. После подтверждения пользователь должен дождаться, когда кассир отметит его товар готовым к выдаче. А затем уже клиент может забрать свой заказ.

# 1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## 1.1 Обзор аналогов

Подобных приложений существует множество. Самые популярные из них: «*Burger King*», «*KFC Беларусь*».

Веб-сервис «*Burger King*» (рисунок 1.1) – это удобный и быстрый способ заказать еду онлайн в ресторане, на машине или с доставкой на дом. Пользователь может выбрать любимые блюда из меню, заказать и оплатить их всего в несколько касаний. Выбрав заказ на вынос в ресторане, клиент может забрать его без очереди! Быстрая доставка привезет еду в течение 30 минут [2].

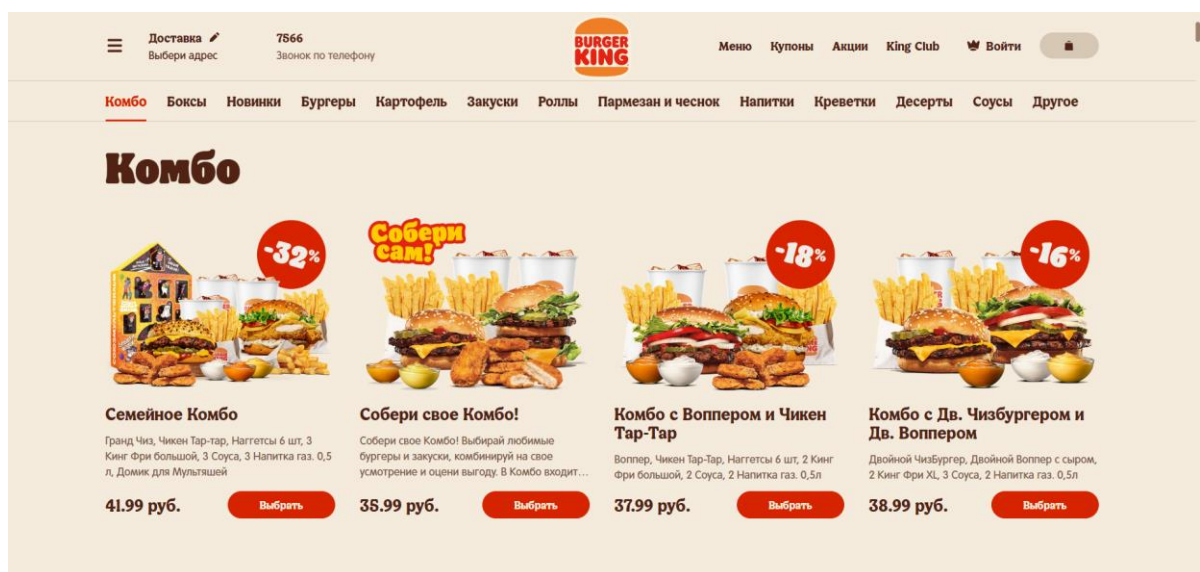


Рисунок 1.1 – Интерфейс веб-сервиса «*Burger King*».

При использовании данной программы я обнаружил много преимуществ:

- система бонусов за покупки, которыми можно оплачивать заказы (при достаточном их количестве);
- удобный сервис доставки заказов;
- современный и красивый дизайн, интуитивно понятный интерфейс.

Но есть и свои недостатки: «зависания» приложения, остановка работы, частые проблемы с оплатой заказа.

В веб-сервисе «*KFC Беларусь*» (рисунок 1.2) представлены меню с выгодными предложениями, возможность оформления заказа на самовывоз или с доставкой на дом, а также новинки и акции ресторана *KFC* в Беларуси [3].

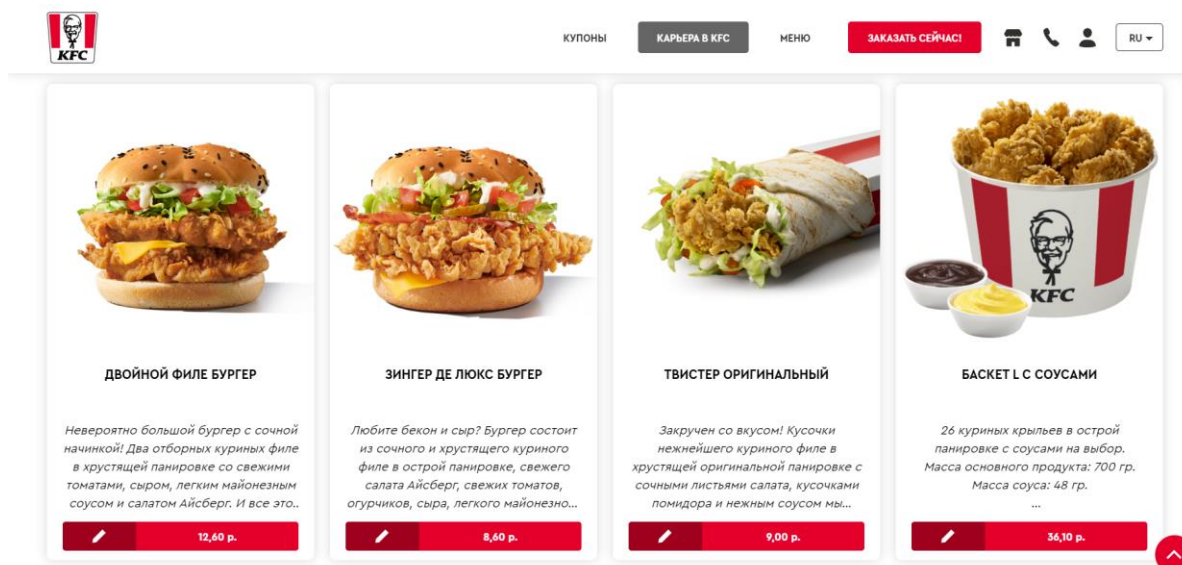


Рисунок 1.2 – Интерфейс веб-сервиса «KFC Беларусь».

В приложении есть возможность найти ближайший ресторан *KFC*. Основные преимущества в работе с приложением: актуальные акции и купоны (выгодные предложения), удобное меню. Из недостатков можно отметить довольно частые сбои в работе, непонятный интерфейс, проблемы с оплатой заказа.

## 1.2 Постановка задачи

При реализации курсового проекта «Веб-служба кафе «Макдональдс»» были поставлены следующие задачи:

- создание серверной части;
- создание клиентской части.

В серверной части необходимо:

- настроить взаимодействие с базой данных;
- обеспечить взаимодействие сервера и клиента с помощью HTTP-запросов;
- безопасное хранение паролей в базе данных.

В клиентской части:

- обеспечить понятный интерфейс;
- получение от сервера необходимых данных;
- настроить авторизацию пользователя;
- безопасная передача пароля при авторизации от клиента серверу.

Языком программирования был выбран C#, так как он имеет мощную экосистему, обеспечивает высокую производительность, сильную типизацию и безопасность, удобную интеграцию с базами данных, поддерживает микросервисы и облачные технологии.

Для создания серверной части использовался *ASP.NET Core*, а для клиентской части – *WPF*.

Разработка приложения производилась в среде разработки *Microsoft Visual Studio 2022*.



## 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

### 2.1 Структура программы

При разработке приложения будет использовано 10 модулей:

- *Admin* – модуль администратора, дающий доступ к редактированию таблиц базы данных;
- *App* – модуль, хранящий основные настройки программы, в частности оформление (стили);
- *Cart* – модуль, обеспечивающий отображение корзины для пользователя и работу с ней;
- *Cashier* – модуль кассира, обеспечивающий работу заказами клиентов;
- *History* – модуль, обеспечивающий просмотр истории заказов пользователя;
- *Item* – модуль, обеспечивающий просмотр подробной информации о товаре;
- *Menu* – модуль, обеспечивающий просмотр меню кафе по категориям;
- *Restaurants* – модуль, обеспечивающий просмотр расположения и информации о всех кафе «Макдональдс» в Беларуси;
- *SignIn* – модуль для входа в аккаунт пользователя;
- *SignUp* – модуль для регистрации нового пользователя.

### 2.2 Проектирование интерфейса программного средства

#### 2.2.1 Структура вкладки

Каждая вкладка должна иметь сверху панель вкладок, а снизу содержание текущей вкладки.



Рисунок 2.1 – Структура вкладки.

По нажатию на кнопки сверху должна открываться соответствующая вкладка.

### 2.2.2 Вкладка «Меню»

Вкладка «Меню» должна включать категории товаров и сами товары, подходящие под выбранную категорию. Должна быть возможность фильтрации по ценам.



Рисунок 2.2 – Категории товаров.

Сами товары должны отображаться сеткой по 3 товара на строку. Пользователь должен видеть фото товара, название и цену. По клику на товар откроется подробная информация о товаре, который он сможет добавить в корзину после авторизации. Рисунок сетки товаров представлен на следующей странице.



Цезарь Ролл  
10.00 руб.



Биф Тейсти Ролл  
11.00 руб.



Мак Бургер  
9.10 руб.



Гранд Чизбургер  
9.30 руб.



Биг Бургер  
13.80 руб.



Двойной Гранд Чизбургер  
13.80 руб.

Рисунок 2.3 – Сетка товаров во вкладке «Меню».

### 2.2.3 Подробная информация о товаре

В подробной информации о товаре должно быть представлено фото товара крупным планом, подробное описание о товаре, а также цены при различных размерах товара (например для латте – средняя порция стоит 5 руб., большая – 5.4 руб.).

#### Кантри Бургер

Сочный бифштекс из 100 % белорусской свинины, свежий лучок, маринованные огурчики и нежный сливочный соус с ароматом приправ. Попробуйте и насладитесь незабываемым вкусом!

3.30 руб.



Рисунок 2.4 – Подробная информация о товаре.

### 2.2.4 Авторизация пользователя

При регистрации пользователя нужно указать имя и пароль. Пароль нужно вписать дважды (чтобы они совпадали). При входе в созданный аккаунт пароль вписывается только один раз. Авторизация дает доступ к вкладкам «Корзина» и «История заказов».

**Регистрация**

Введите логин

Введите пароль

Повторите пароль

**Зарегистрироваться**

Рисунок 2.5 – Регистрация пользователя.

**С возвращением!**

Введите логин

Введите пароль

**Войти**

Ещё нет аккаунта? [Зарегистрироваться](#)

Рисунок 2.6 – Вход пользователя в созданный аккаунт.

### 2.2.5 Корзина и история заказов

Корзина должна отображать минимальную информацию о товаре: фото, цену, название, количество. Внизу должна отображаться итоговая стоимость заказа. При оплате корзины должно быть поле для указания ресторана, из которого пользователь собирается забрать свой заказ, а также способ оплаты (наличный или безналичный). В истории заказов та же информация, что и в корзине дополнительно с датой заказа. Все рисунки предоставлены на следующей странице.




	Цезарь Ролл	- + x5	10.00
	Маффин с яйцом и ветчиной	- + x1	5.50
	Американо (Средний)	- + x1	4.70
			<b>Итого:</b>
<div>Оплатить</div>			<b>60,2 руб.</b>

Рисунок 2.7 – Корзина пользователя.

# Оплата

Адрес ресторана...

ул. Шугаева, 2

---

Способ оплаты:

☒ Безналичная

☐ Наличная

Оплатить

Рисунок 2.8 – Выбор адреса ресторана и способа оплаты.

27.05.2025




	Цезарь Ролл (5 шт.)
	Маффин с яйцом и ветчиной (1 шт.)
	Американо (Средний) (1 шт.)
60,2 руб.	

Рисунок 2.9 – История заказов пользователя.

## 2.2.6 Меню управления администратора

Администратор может редактировать таблицы пользователей и товаров базы данных.

### Управление пользователями

### Управление меню

Рисунок 2.10 – Меню администратора.

В меню администратора должна быть возможность добавления нового товара, удаления товара или пользователя, а также изменения ячеек таблицы с последующим автоматическим сохранением изменений в базу данных.



id	Name	Image	Description
1	Цезарь Ролл	C:/Users/fona/source/repos/coursework3/Media/roll2.png	Сочные куриные стрипсы в хрустящей панировке, тертый сыр Пармезан, салат айс
2	Биф Тейсти Ролл	C:/Users/fona/source/repos/coursework3/Media/roll1.png	Два сочных бифштекса из 100 % говядины, свежие салат, помидор и лук, сыр Эмме
3	Мак Бургер	C:/Users/fona/source/repos/coursework3/Media/burger12.png	Бургер с двумя сочными бифштексами из натуральной говядины, сыром Чеддер, са
4	Гранд Чизбургер	C:/Users/fona/source/repos/coursework3/Media/burger11.png	Бургер с бифштексом из натуральной говядины, сыром Чеддер, маринованными о
5	Биг Бургер	C:/Users/fona/source/repos/coursework3/Media/burger10.png	Большой рубленый бифштекс из 100 % говядины, два слайса сочного помидора, св
6	Двойной Гранд Чизбургер	C:/Users/fona/source/repos/coursework3/Media/burger9.png	Бургер с двумя сочными бифштексами из натуральной говядины, сыром Чеддер, са
7	Чикен с пикантной горчицей	C:/Users/fona/source/repos/coursework3/Media/burger7.png	Хрустящая куриная котлета, свежий салат, маринованные огурчики, жареный лук и
8	Кантри Бургер	C:/Users/fona/source/repos/coursework3/Media/burger4.png	Сочный бифштекс из 100 % белорусской свинины, свежий лучок, маринованные о
9	Биг Бургер с курицей	C:/Users/fona/source/repos/coursework3/Media/burger5.png	Нежная куриная котлета в хрустящей золотистой панировке, два слайса сочного п
10	Биф с пикантной горчицей	C:/Users/fona/source/repos/coursework3/Media/burger6.png	Сочный бифштекс из 100 % говядины, свежий салат, маринованные огурчики, жар
11	Двойной Биг Бургер	C:/Users/fona/source/repos/coursework3/Media/burger8.png	Два больших рубленых бифштекса из 100 % говядины, два слайса сочного помид
12	Двойной Кантри Бургер	C:/Users/fona/source/repos/coursework3/Media/burger3.png	
13	Мазстро Биф Бургер	C:/Users/fona/source/repos/coursework3/Media/burger2.png	Премимальная новинка от мастер-шефа Антона Каленика! Ароматная булочка брио
14	Мазстро Чикен Бургер	C:/Users/fona/source/repos/coursework3/Media/burger1.png	Премимальная новинка от мастер-шефа Антона Каленика! Ароматная булочка брио
15	Стрипсы	C:/Users/fona/source/repos/coursework3/Media/chicken8.png	Нежная курочка в хрустящей панировке.

Удалить товар

Добавить товар

Рисунок 2.11 – Управление таблицей товаров в меню администратора.

## 2.2.7 Вид приложения для кассира

Кассир должен видеть весь заказ: фото товара, название, количество, сумму заказа и способ оплаты. Он должен подтвердить заказ или отклонить его нажатием по соответствующей кнопке. Последующее использование кнопок – по готовности товара. Рисунок вида приложения для кассира на следующей странице.



Мак Бургер (1 шт.)

9,1 р.  
Безналичная

Принять

Отклонить

Рисунок 2.12 – Вид приложения для кассира.

## 2.3 Проектирование функционала программного средства

Рассмотрим 3 функции:

- отображение товаров выбранной категории;
- авторизация пользователя;
- отображение заказов кассиру.

### 2.3.1 Отображение товаров выбранной категории

При выборе категории товара, отправляется запрос в базу данных. Она вернет список товаров, подходящих по категории. Полученный список товаров сортируется по выбранной фильтрации по цене. После этого список товаров выводится на экран пользователю в том порядке, в котором товары расположены в отсортированном списке (от первого элемента и до последнего).

### 2.3.2 Авторизация пользователя

При нажатии на клавишу «Войти» производится проверка, существует ли пользователь в базе данных с введенным именем. Если да, сервер получает зашифрованный пароль от клиента. Сервер расшифровывает данный пароль, шифрует по другой хеш-функции и сверяет с записанным паролем в базе данных. Если пароли совпали, пользователь успешно получает доступ в свой аккаунт.

### 2.3.3 Отображение заказов кассиру

В базе данных проверяется, какой пользователь указал в заказе адрес ресторана, в котором работает данный кассир. Получив такие заказы, проверяется статус заказа пользователя. Если он равен 1, то кассир выбирает отклонить или подтвердить заказ. Если статус равен 2 – кассир может указать, что заказ готов к получению. Если статус 3 – кассир может указать, что заказ забрали. При каждом нажатии на данные кнопки, кассир увеличивает статус заказа пользователя на 1. Если текущий статус равен 3 и кассир нажал на кнопку, статус становится равным 0 и заказ исчезает из панели кассира. При отклонении же заказа статус становится -1 и заказ тоже исчезает из панели кассира.

## 3 РАЗРАБОТКА ПРОГРАММНОГО СРЕДСТВА

### 3.1 Физическая структура

В приложении пользователь взаимодействует с базой данных через сервер, когда это необходимо. В меню используется информация из таблицы товаров, во всех остальных разделах (кроме «Рестораны» и раздела авторизации) используются обе таблицы (*users.db* и *products.db*). В разделе авторизации используется лишь таблица пользователей. Лишь в разделе «Рестораны» ни одна таблица не используется.

Все действия, связанные с корзиной (добавление товара, удаление, изменение количества, оплата корзины), передаются в таблицу пользователей.

В версии для кассира используются обе таблицы для получения информации. Любое действие кассира (отклонение, принятие заказа, отметка о готовности к выдаче, успешное получение заказа пользователем) записывается в таблицу пользователей.

Можно выделить два бизнес-процесса: «Процесс продаж» и «Процесс производства».

«Процесс продаж» описывает взаимодействие программы с пользователем. В него входит получение из базы данных информации о товарах пользователем, выбор товара, добавление его в корзину, что сразу же передается обратно в базу данных, последующая оплата корзины, итог которой тоже возвращается в базу данных.

«Процесс производства» выполняется только после прохождения «процесса продаж». Из базы данных в приложение к кассиру поступает информация о заказе. От него требуется подтверждение или отказ от заказа, последующая (после принятия) отметка о готовности, о получении товара. Ответ кассира возвращается обратно в базу данных. Каждый ответ из базы данных поступает в программу пользователя, ожидающего заказ [10].

### 3.2 Описание разработанных модулей

Для описания разработанных модулей выбраны три наиболее крупных: «Menu», «Cart» и «Authorization».

Модуль «Menu» включает в себя меню, а именно представление всех доступных для заказа товаров. Они разделены для удобства на категории. По каждому товару представлена краткая информация: название, стоимость, картинка. Эта информация берется из базы данных для товаров. Для более детального описания продукта и для его добавления в корзину (включая выбор количества товара для заказа) можно открыть его «личную» страницу, на которой будет помимо предоставленной ранее информации также показано описание товара. Также представлена возможность сортировки товаров по цене (как по возрастанию, так и по убыванию цены). Функция заказа товара доступна лишь зарегистрированным пользователям.



Модуль «*Cart*» реализует весь функционал корзины пользователя. Воспользоваться корзиной может только зарегистрированный пользователь. Если ранее в нее не было добавлено товаров, то есть корзина пуста, об этом пользователю будет сообщено во вкладке «Корзина» приложения. В противном случае там будут находиться все добавленные в корзину товары. Для удобства и избежания внешней нагруженности в ней представлены лишь самые основные данные о товаре – картинка, название, количество и цена. У пользователя есть возможность по нажатию на товар попасть на его «личную» страницу. Также клиент может изменить количество товара в корзине – увеличить его, уменьшить, а также убрать товар полностью из корзины. В нижней части окна написана итоговая цена весь заказ, а рядом расположена кнопка оплаты всего заказа. При оплате пользователю необходимо выбрать, какое кафе будет готовить его заказ (а значит в этом кафе данный заказ пользователь должен забрать) и способ оплаты: наличная или безналичная. Оплата производится только при получении клиентом товара. После выбора кафе и способа оплаты заказа информация о нем поступает кассирам кафе «Макдональдс», которые должны отклонить его или принять. После ответ кассира доходит до пользователя. Если заказ принят – работники кафе «Макдональдс» приступают к его приготовлению. В противном случае заказ отменяется. После приготовления кассир устанавливает отметку о готовности заказа к выдаче, о чем пользователь тоже уведомляется. После того, как заказчик забрал свой заказ, кассир отмечает это в приложении. После этого заказ исчезает из приложения кассира, корзина пользователя полностью очищается, а весь заказ переходит в раздел «История заказов» пользователя.

Модуль «*Authorization*» включает в себя функционал регистрации и авторизации пользователя. Чтобы войти в аккаунт, он должен быть создан. Для создания аккаунта предлагается пройти регистрацию. От пользователя необходимо указать свое имя, под которым он хочет осуществлять заказы, а также придумать пароль, который он будет вводить при каждом входе в аккаунт. При регистрации пароль необходимо одинаково ввести дважды, а при входе в аккаунт ввод пароля всегда требуется лишь один раз. После регистрации введенный пароль проходит хеширование. В базе данных хранятся пароли, уже прошедшие через хеш-алгоритм. Также существует проверка на существующие имена аккаунтов. Если пользователь введет такое имя, то его попросят ввести другое. Это сделано для избежания конфликтов при работе с базами данных. При входе в аккаунт существуют различные валидации данных:

- проверка на существование аккаунта с введенным именем;
- проверка на совпадение введенного пароля с паролем аккаунта в базе данных.

После прохождения данного модуля у пользователя открывается весь доступный пользователю функционал.

#### 4 ТЕСТИРОВАНИЕ ПРОГРАММНОГО СРЕДСТВА

При первоначальной проверке схемы были обнаружены проблемы, связанные с оформлением заказа пользователями. Возможность заказа прошлых заказов из истории пользователя может привести к ошибкам: в истории отображается все, что покупал пользователь. В том числе продукты, которые больше не в наличии в кафе. В связи с этим при заказе прошлого заказа, который включает в себя товар, более не продаваемый, может возникнуть неопределенное поведение приложения. Поэтому функция повторного заказа из истории не была осуществлена.

Переходим к более детальному анализу, направленному на обеспечение надежности и безопасности и устранение рисков, связанных с нагрузками и задержками. При моделировании приложения планировалась передача от сервера клиенту сразу всех товаров из базы данных. Это оказалось нелогичным, так как в приложении не могут отображаться одновременно все товары (они разделены по категориям). Одновременное хранение всей таблицы товаров нагружает программу, что при длительном ее использовании может привести к долгим и неприятным для пользователя задержкам в работе. Решением оказалась передача только товаров, относящихся к той категории меню, которая сейчас открыта у пользователя в приложении. Также изначально при обращении к пользовательским параметрам – таким как корзина, история заказов – предоставлялась сразу вся информация о пользователе. Однако это небезопасный подход, поэтому было решено передавать только то, что необходимо в данный момент.

## 5 ПРИМЕНЕНИЕ ПРОГРАММЫ

### 5.1 Руководство пользователя

При первом запуске программы перед вами отобразится все доступное для заказа меню кафе «Макдональдс» (рисунок 5.1).

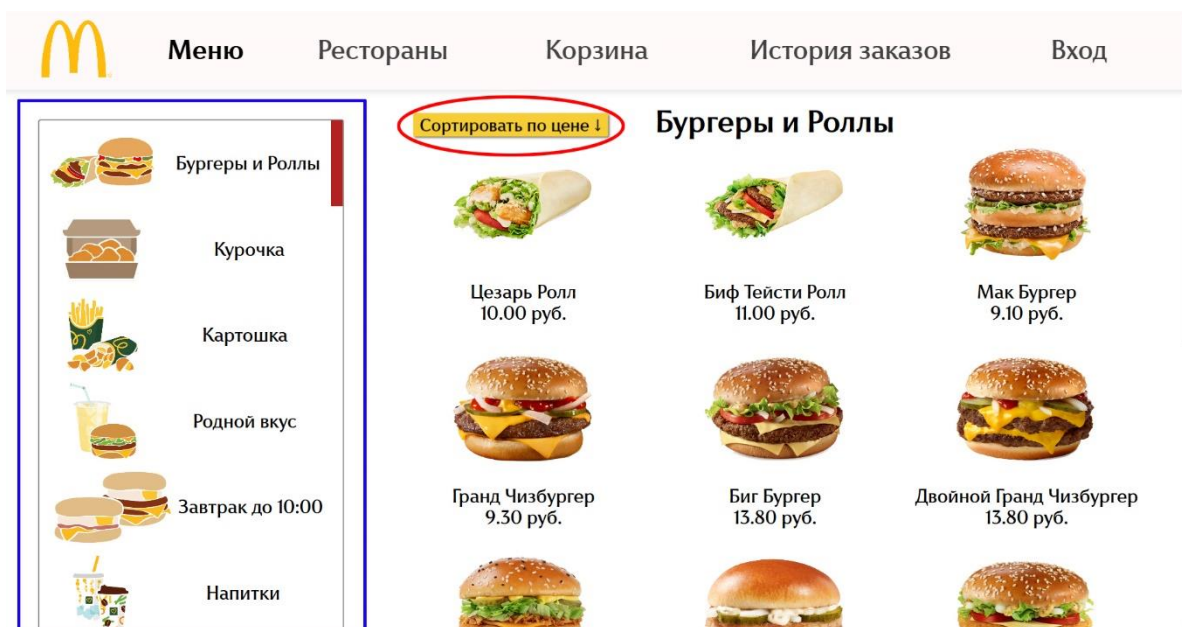


Рисунок 5.1 – Меню кафе.

Сверху расположена навигация по приложению, включающая в себя такие вкладки, как «Меню», «Рестораны», «Корзина», «История заказов», «Вход». Сейчас вы находитесь во вкладке «Меню». Рассмотрим ее.

Вкладка «Меню» состоит из двух частей: левой и правой. В левой части расположены категории товаров (рисунок 5.1, обведено синим цветом).

Вы можете переключаться между категориями по нажатию на необходимую. В правой части данной вкладки расположены непосредственно сами товары, входящие в данную вкладку. Для своего удобства вы можете использовать сортировку товаров текущего раздела по цене. Нажав на соответствующую кнопку первый раз (она обведена на рисунке 5.1, обведено красным цветом), вы включите сортировку по возрастанию цены. Если нажмете на нее второй раз, включится сортировка по убыванию [12]. Вы можете подробно просмотреть каждый товар с описанием. Для этого необходимо нажать на выбранный вами товар. После этого откроется подробное описание продукта, на который вы нажали (рисунок 5.2 на следующей странице). Однако добавить товар в корзину можно только после авторизации.

Для авторизации перейдите во вкладку «Вход». Сначала вам будет предложено войти в уже созданный аккаунт (рисунок 5.3 на следующей странице). Если вы не создали аккаунт, вам необходимо нажать на кнопку

«Еще нет аккаунта? Зарегистрироваться» (рисунок 5.3 на следующей странице, обведено красным цветом). Если же вы уже были зарегистрированы ранее, пропускайте этот абзац. При нажатии на кнопку откроется окно регистрации. В нем вам необходимо ввести ваш логин – имя, под которым вы будете делать заказы. Далее вам необходимо придумать и ввести пароль. Пароль нужно будет повторить. Запомните ваш логин и пароль – вы будете вводить их при каждом входе в свой аккаунт. После того, как вы заполните все поля, нажмите на кнопку «Зарегистрироваться» (рисунок 5.4 на следующей странице).

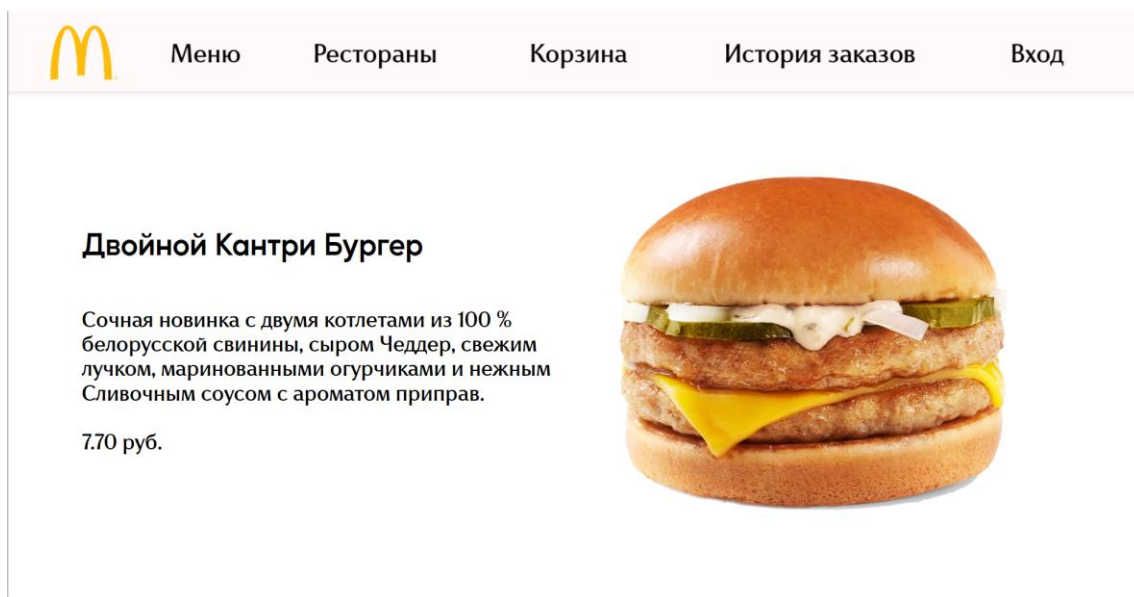


Рисунок 5.2 – Подробное описание выбранного продукта до авторизации.

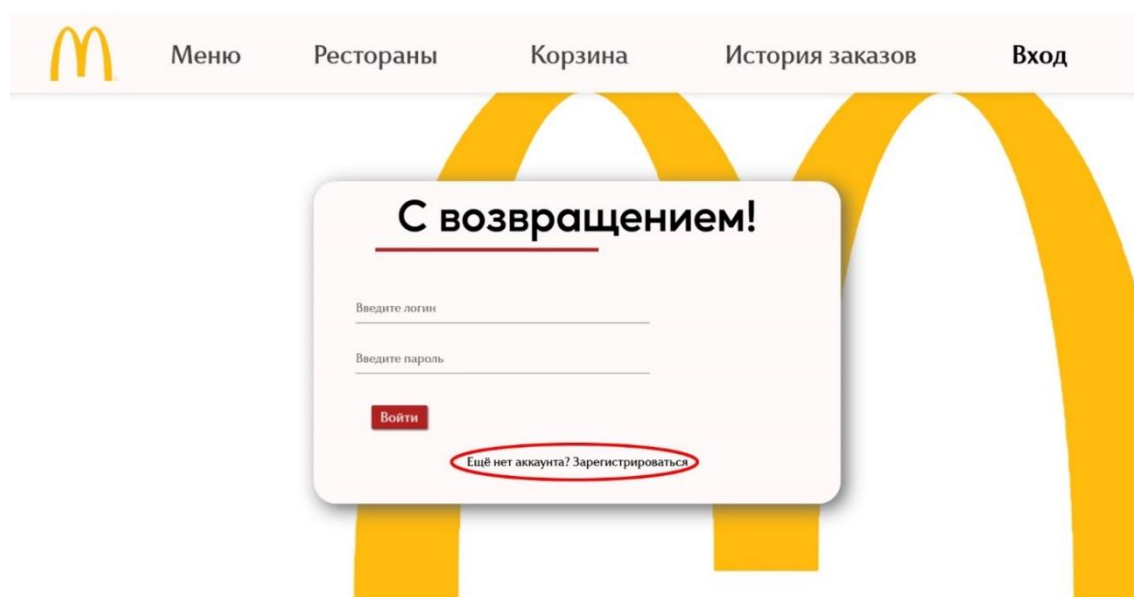


Рисунок 5.3 – Вход в аккаунт.

После регистрации вы должны войти в свой созданный аккаунт. Если вы только зарегистрировались, у вас автоматически откроется окно для входа. Вам нужно заполнить все поля – ввести ваш логин и пароль, которые вы указывали при регистрации. После того, как все поля будут заполнены, нажимайте кнопку «Войти».

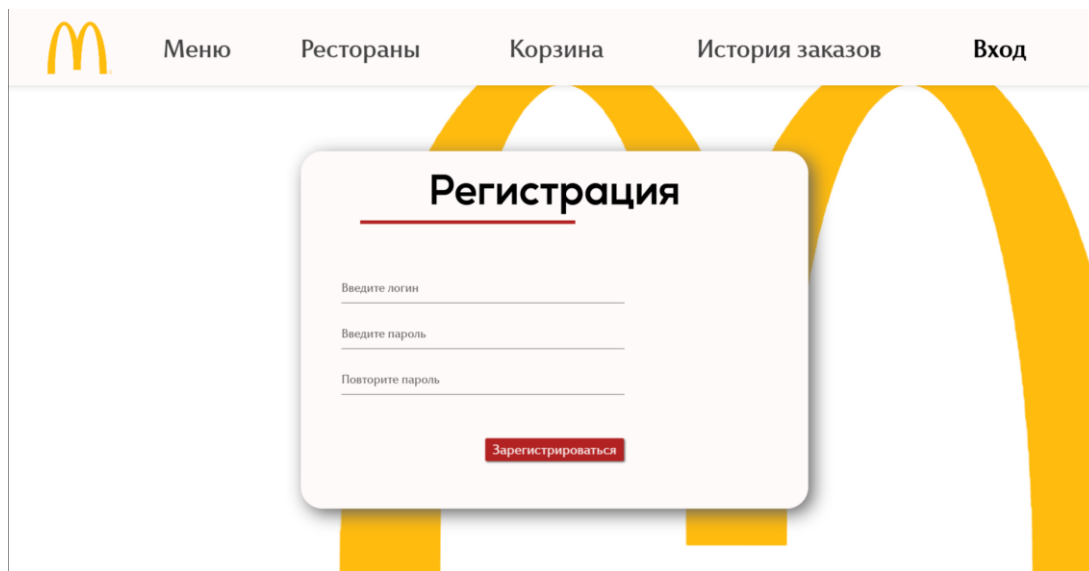
The image shows a web browser window with the McDonald's website. At the top, there is a navigation bar with the McDonald's logo on the left and links for 'Меню' (Menu), 'Рестораны' (Restaurants), 'Корзина' (Basket), 'История заказов' (Order History), and 'Вход' (Login) on the right. In the center of the page, there is a white registration modal box titled 'Регистрация' (Registration). Inside the box, there are three input fields labeled 'Введите логин' (Enter login), 'Введите пароль' (Enter password), and 'Повторите пароль' (Repeat password). Below these fields is a red button with the text 'Зарегистрироваться' (Register).

Рисунок 5.4 – Регистрация нового пользователя.

После входа в аккаунт у вас открываются возможности добавления товара в корзину, совершения заказа (вкладка «Корзина») и использования вкладки «История заказов». Без авторизации вы не сможете использовать эти функции.

После авторизации во вкладке «Меню», если выбрать какой-либо заинтересовавший вас товар, нажав на него, в открывшемся подробном описании товара появится кнопка «В корзину», а также функция изменения количества единиц товара, который вы хотите добавить в корзину (рисунок 5.5 на следующей странице). Для порционных товаров и напитков вы можете выбрать размер товара, который вы хотите добавить в корзину (рисунок 5.6 на следующей странице, обведено красным). Если вы хотите добавить выбранный товар в корзину, нажмите на кнопку «В корзину».

Вкладка «Корзина» не работает без авторизации. В ней отображена ваша корзина. Она может быть пустой – если вы не добавили в нее продуктов. Все добавленные вами продукты отображаются в корзине (рисунок 5.7 на следующей странице). Вы можете перейти к подробной информации о товаре в вашей корзине, нажав на название или картинку товара. Также вы можете изменить количество единиц товара в корзине, путем нажатия на «+» или «-» соответственно. Вы можете удалить товар из корзины, если при одной единице товара в корзине нажать на «-» в его ячейке. Если вы готовы совершить заказ, вам нужно нажать на кнопку «Оплатить» слева внизу окна. У вас откроется

окно оплаты корзины. В нем вам необходимо указать кафе, из которого вы хотите забрать заказ, и способ оплаты – безналичная или наличная.

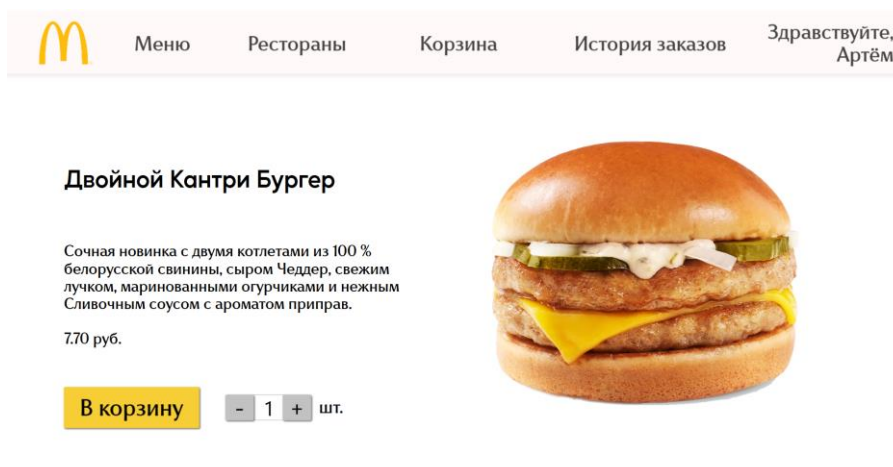


Рисунок 5.5 – Подробная информация о товаре после авторизации.

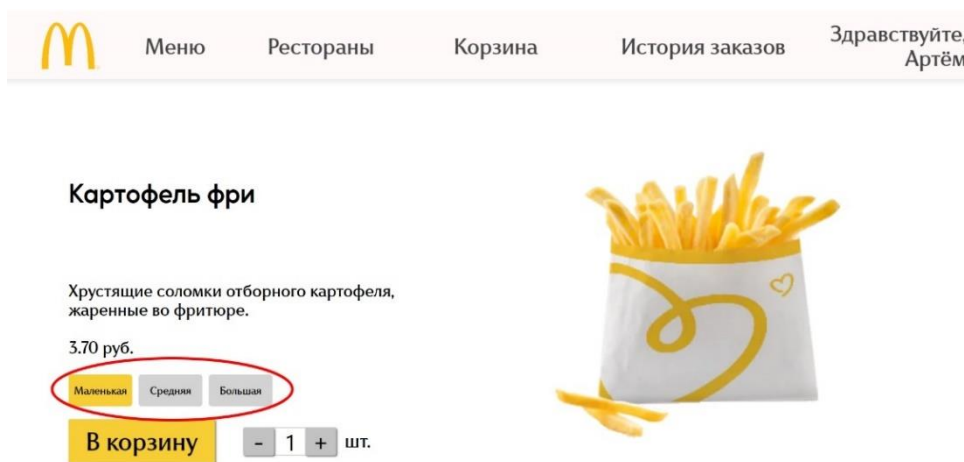


Рисунок 5.6 – Выбор размера продукта.

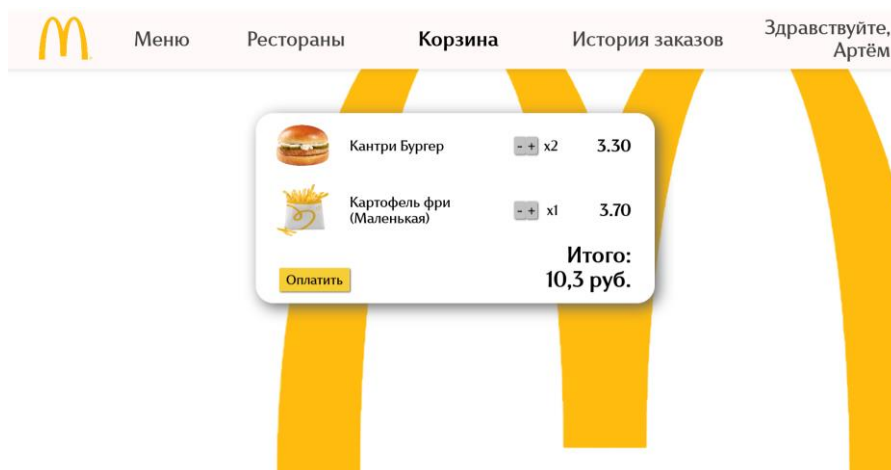


Рисунок 5.7 – Корзина пользователя.

Внешний вид окна оплаты представлен на рисунке 5.8. В поле «Адрес ресторана...» необходимо указать адрес кафе. Как только вы начнете вводить адрес, появится выпадающий список с предложенными адресами на выбор. Выберите нужный вам адрес путем нажатия на него. После выбора адреса ресторана и способа оплаты нажмите кнопку «Оплатить». Если вы передумали оплачивать корзину, вы можете отменить оплату, нажав еще раз на вкладку «Корзина». Обратите внимание, что денежная оплата будет производиться непосредственно в самом кафе, где вы будете забирать заказ.

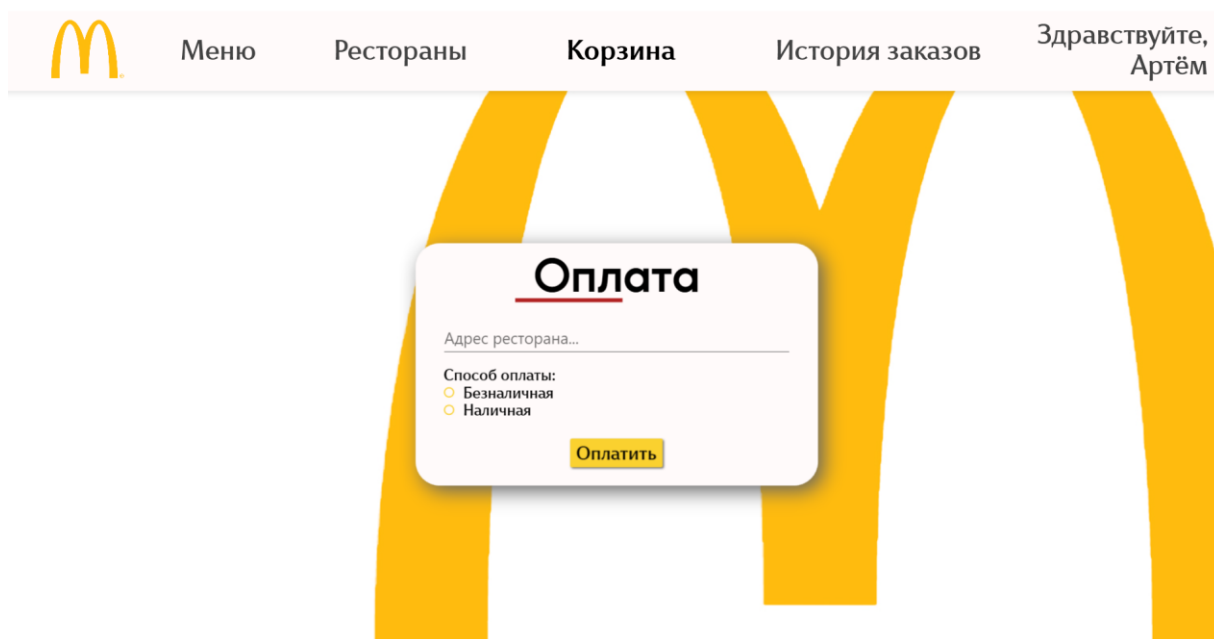


Рисунок 5.8 – Оплата корзины.

После нажатия на кнопку «Оплатить» вы должны дождаться одобрения вашего заказа кассиром. Об его ответе вы будете уведомлены во вкладке «Корзина». После подтверждения заказа повара начнут готовить ваш заказ. Ожидайте его приготовления. О готовности заказа к выдаче вы будете уведомлены во вкладке «Корзина», внизу окна (рисунок 5.9 на следующей странице, обведено красным). После выдачи заказа он исчезнет из вашей корзины и появится во вкладке «История заказов». Обратите внимание, что при ожидании подтверждения кассира вашего заказа, при его готовке и при ожидании его получения вы не можете изменять текущую корзину или добавлять в нее новые товары. В случае отказа от вашего заказа вы будете также уведомлены об этом соответствующим сообщением внизу окна корзины. После отказа вы можете свободно изменять свою корзину, а также оплатить ее снова.

Вкладка «История заказов» не работает без авторизации. В ней вы увидите все совершенные вами за все время заказы (рисунок 5.10 на следующей странице). Вам показана дата заказа, названия и картинки товаров, которые были в заказе, и общая сумма заказа. Каждый заказ отделен друг от друга черной горизонтальной полосой.



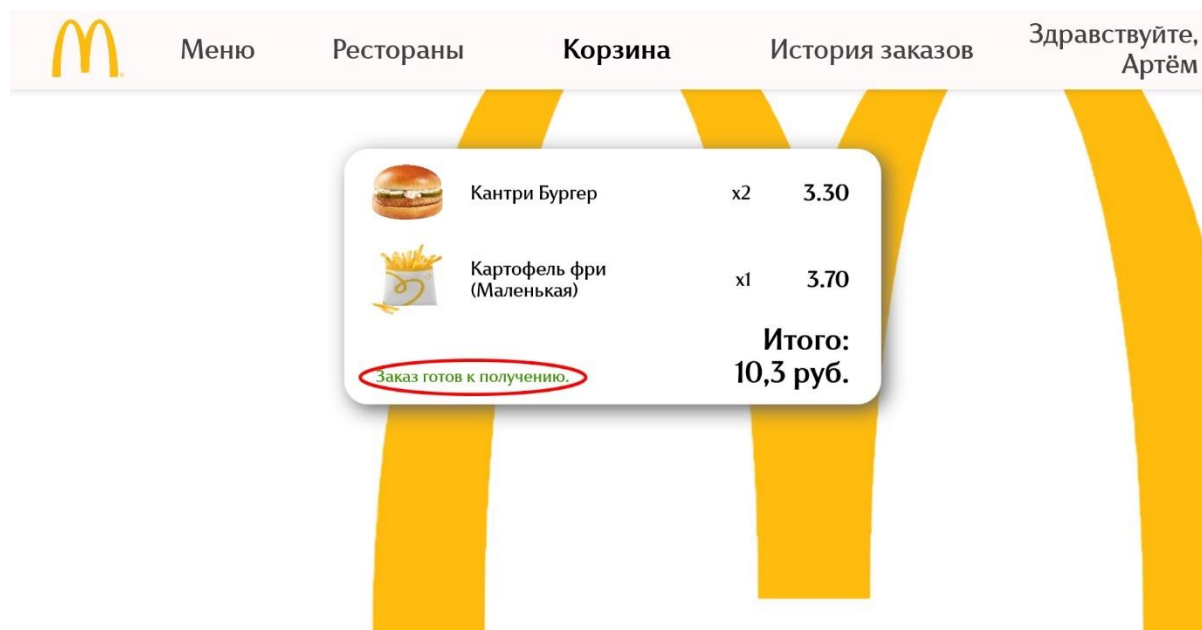


Рисунок 5.9 – Уведомление о готовности заказа к выдаче во вкладке «Корзина».

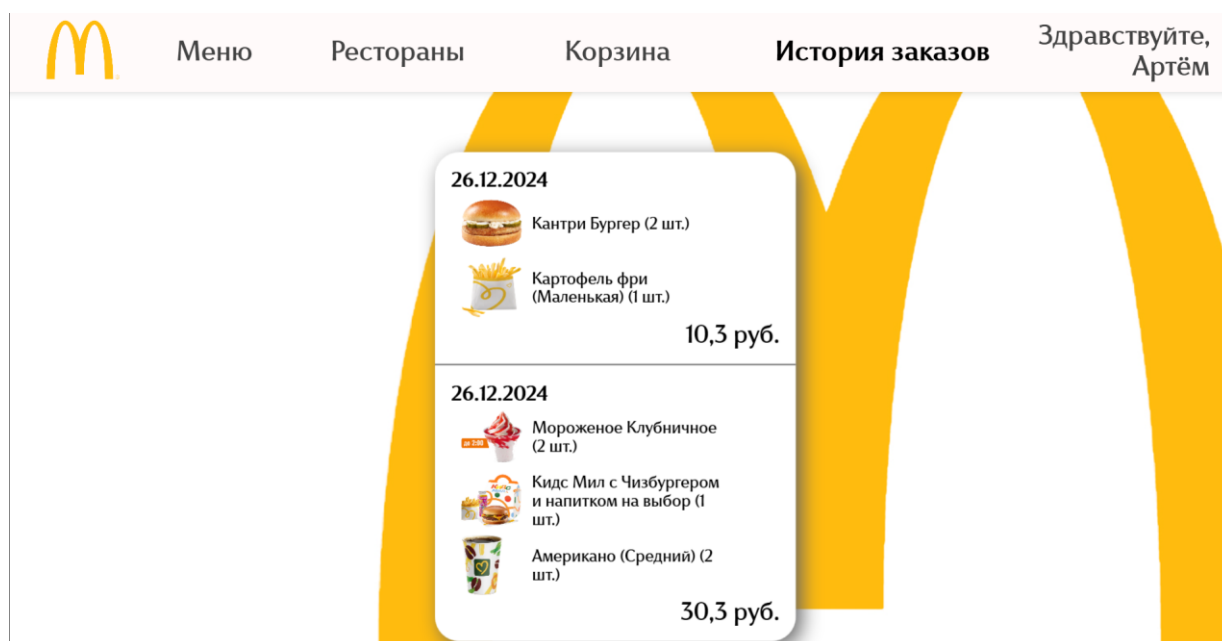


Рисунок 5.10 – История заказов.

Вкладка «Рестораны» – показывает все кафе «Макдональдс» в Беларуси на карте, а также информирует о дополнительных благах кафе (рисунок 5.11 на следующей странице) [13]. Она доступна как авторизованным, так и неавторизованным пользователям. Вы можете узнать о наличии определенных услуг у каждого кафе, таких как «Драйв» – возможность заказа на машине, «Завтрак» – возможность заказа товаров из категории «Завтрак до 10:00», «24/7» – круглосуточная работа кафе, «Парковка» – наличие парковки у кафе, «Wi-Fi» – наличие бесплатной беспроводной сети Интернет в кафе [14]. Для



этого необходимо просто нажать на интересующее вас кафе на карте. Как это выглядит – изображено на рисунке 5.12. Для изменения масштаба на карте используйте колесико мыши [15].

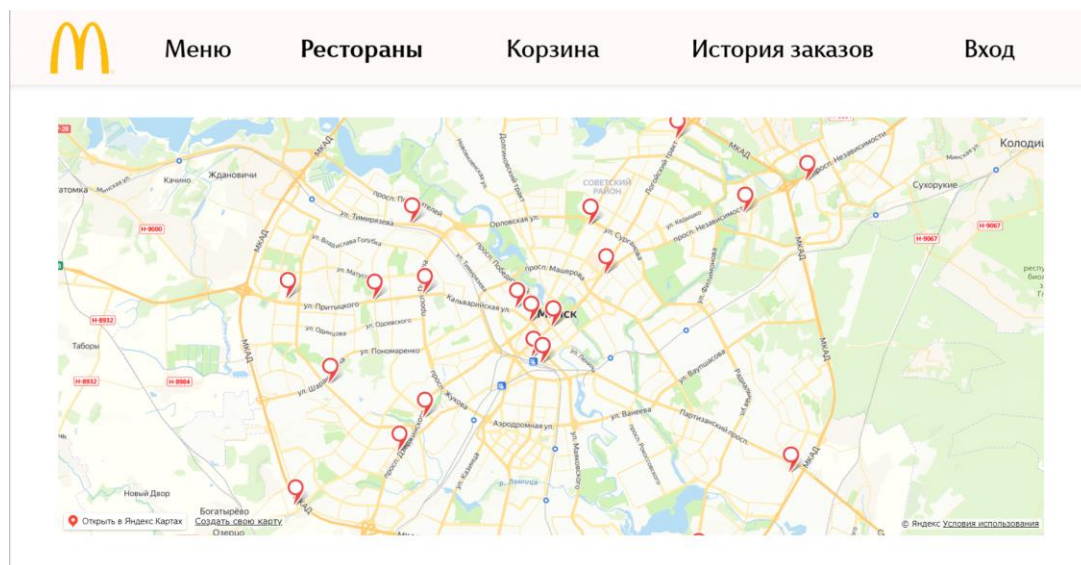


Рисунок 5.11 – Карта со всеми кафе «Макдональдс» в Минске.

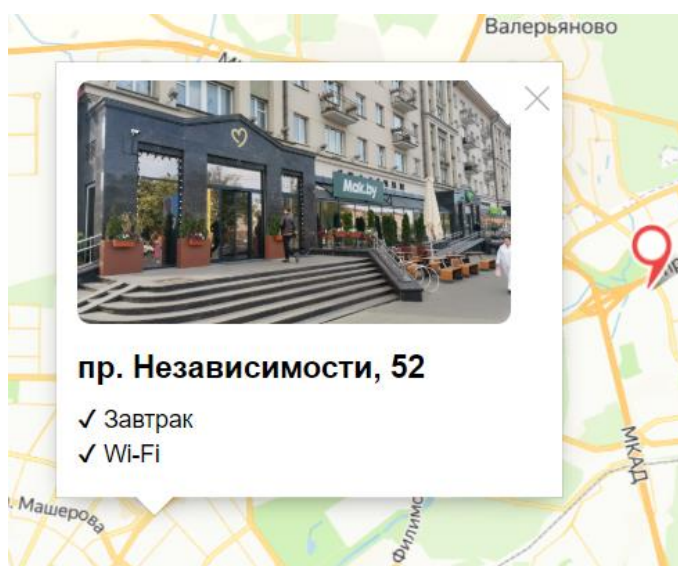


Рисунок 5.12 – Дополнительная информация о кафе «Макдональдс».

## ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы «Веб-служба кафе «Макдональдс»» была разработана программа, направленная на улучшение процесса обслуживания клиентов. Основной целью работы являлось упрощение и ускорение процесса заказа, а также минимизация ошибок, повышение качества взаимодействия с клиентами и безопасность работы сервера.

Реализованное приложение предоставляет пользователям доступ к актуальному меню с отображением акций и специальных предложений, что делает процесс заказа интуитивно понятным и удобным. Автоматизация передачи заказов от клиента к кассиру и поварам позволила исключить задержки, связанные с человеческим фактором, и сократить время ожидания.

Дополнительно был решен ряд важных задач, таких как создание своего алгоритма обработки заказов, а также реализация возможности работы администратора с базой данных в режиме реального времени. Это позволило достичь поставленных целей, включая снижение нагрузки на персонал, сокращение очередей и повышение общего уровня удовлетворенности клиентов, исключение социального взаимодействия, что также является преимуществом для некоторых клиентов.

Для обеспечения безопасности данных в приложении реализовано шифрование паролей, что является важным аспектом защиты пользовательской информации. Применение современных криптографических алгоритмов позволяет минимизировать риск утери аккаунта и его данных. Также для проверки подлинности пользователей используются *JWT*-токены для предотвращения несанкционированного доступа к системе со стороны злоумышленников.

Помимо всех перечисленных выше возможностей было решено добавить сортировку меню по цене (как по возрастанию, так и по убыванию). Она позволяет клиентам быстрее ориентироваться в предложениях, выбирая товары, которые соответствуют их бюджету.

Данная работа демонстрирует эффективность внедрения веб-сервисов в сфере общественного питания, где критически важны точность и комфорт. Она не только упрощает процесс заказа для клиентов, но и повышает эффективность работы персонала, что в конечном итоге приводит к увеличению продаж и улучшению общего качества обслуживания.

Программа может быть легко адаптирована под нужды других предприятий, что открывает перспективы для ее дальнейшего использования и совершенствования. Например, планируется развить карту ресторанов: добавить фильтрацию по наличию парковок, завтраков, круглосуточной работе. Также для повышения доступности и комфорта клиентов хорошим решением было бы добавление доставки заказа курьером по адресу. Рассматривается добавление личного кабинета пользователя, в котором он может изменить логин или пароль, привязать банковскую карту, а также

добавить контактные данные для доставки курьером. Дальнейшее развитие системы может включать интеграцию с мобильным приложением, улучшение интерфейса.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- [1] WSENS. История терминалов самообслуживания [Электронный ресурс]. – Режим доступа : <https://wsens.ru/company/articles/495/>.
- [2] БУРГЕР КИНГ – Доставка, купоны [Электронный ресурс]. – Режим доступа : <https://play.google.com/store/apps/details?id=ru.burgerking&hl=ru&pli=1>.
- [3] KFC Беларусь [Электронный ресурс]. – Режим доступа : <https://play.google.com/store/apps/details?id=com.by.kfc&hl=ru>.
- [4] Взаимодействие с базой данных [Электронный ресурс]. – Режим доступа : <https://metanit.com/sharp/wpf/21.1.php>.
- [5] Алгоритмы хэширования в .NET C# [Электронный ресурс]. – Режим доступа : [https://csharp.webdelphi.ru/algoritmy-xeshirovaniya-v-net-c/#\\_sha256\\_sha384\\_sha512](https://csharp.webdelphi.ru/algoritmy-xeshirovaniya-v-net-c/#_sha256_sha384_sha512).
- [6] Работа с SQLite в C# (.NET 5) [Электронный ресурс]. – Режим доступа : <https://csharp.webdelphi.ru/rabota-s-sqlite-v-c-net-5/>.
- [7] Mak.by. Меню [Электронный ресурс]. – Режим доступа : <https://mak.by/catalog/>.
- [8] Лучшие IDE для разработки на C# [Электронный ресурс]. – Режим доступа : <https://blog.geekbrains.by/luchshie-ide-dlja-razrabotki-na-c/>.
- [9] Обзор Visual Studio: возможности и особенности [Электронный ресурс]. – Режим доступа : <https://sky.pro/wiki/javascript/obzor-visual-studio-vozmozhnosti-i-osobennosti/>.
- [10] Что такое бизнес-процесс и описание бизнес процесса [Электронный ресурс]. – Режим доступа : <https://habr.com/ru/articles/342448/>.
- [11] Создание проекта модульного теста [Электронный ресурс]. – Режим доступа : <https://learn.microsoft.com/ru-ru/visualstudio/test/walkthrough-creating-and-running-unit-tests-for-managed-code?view=vs-2022#create-a-unit-test-project>.
- [12] Сортировка объектов. Интерфейс IComparable [Электронный ресурс]. – Режим доступа : <https://metanit.com/sharp/tutorial/3.23.php>.
- [13] Доступ к API [Электронный ресурс]. – Режим доступа : <https://yandex.ru/dev/site/api/ru/concepts/access>.
- [14] Mac.by. Рестораны [Электронный ресурс]. – Режим доступа : [https://mak.by/restorans\\_maps/](https://mak.by/restorans_maps/).
- [15] Предварительный стандарт предприятия [Электронный ресурс] : УО «Белорусский государственный университет информатики и радиоэлектроники», 2024. – Режим доступа : СТП\_2024.pdf.

## ПРИЛОЖЕНИЕ А. Исходный код программы

"Admin.xaml"

```
<Window x:Class="CourseWorkClient.Admin"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="800"
        WindowState="Maximized"
        Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <ScrollViewer VerticalScrollBarVisibility="Auto"
        HorizontalScrollBarVisibility="Auto">
        <Grid Width="1505" Height="784">
            <TextBox x:Name="IdTextBox" HorizontalAlignment="Left"
                VerticalAlignment="Top" Width="56" Margin="742,708,0,0" Height="48"
                Visibility="Collapsed" Style="{StaticResource TypeInTextBox}"
                FontSize="20" HorizontalContentAlignment="Center"
                VerticalContentAlignment="Center" TextWrapping="Wrap" CaretBrush="Black"
                SelectionBrush="#FF130C1F"/>
            <Button x:Name="delUserButton" Content="Удалить пользователя"
                HorizontalAlignment="Left" VerticalAlignment="Top" Width="257" Height="38"
                Margin="480,719,0,0" Click="delUserButton_Click" Style="{StaticResource DefaultButton}"
                FontSize="25" BorderThickness="1" BorderBrush="Black"
                Visibility="Collapsed"/>
            <Button x:Name="delMenuButton" Content="Удалить товар"
                HorizontalAlignment="Left" VerticalAlignment="Top" Width="167" Height="38"
                Margin="570,719,0,0" Click="delMenuButton_Click" Style="{StaticResource DefaultButton}"
                FontSize="25" BorderThickness="1" BorderBrush="Black"
                Visibility="Collapsed"/>
            <Button x:Name="CreateMenuButton" Content="Добавить товар"
                HorizontalAlignment="Left" VerticalAlignment="Top" Width="185" Height="38"
                Margin="1244,717,0,0" Click="CreateMenuButton_Click" Style="{StaticResource DefaultButton}"
                FontSize="25" BorderThickness="1" BorderBrush="Black"
                Visibility="Collapsed"/>
            <Button x:Name="BackToMain" Content="Вернуться в приложение"
                HorizontalAlignment="Left" VerticalAlignment="Top" Width="289" Height="36"
                Margin="1183,35,0,0" Click="BackToMain_Click" Style="{StaticResource DefaultButton}"
                FontSize="25" Visibility="Visible" Foreground="Blue"/>
            <PasswordBox x:Name="AdminPassword" materialDesign:HintAssist.Hint="Введите пароль"
                Style="{StaticResource TypeInPasswordBox}"
                SelectionBrush="#FF130C1F" CaretBrush="Black"
                Foreground="Black" FontSize="23" Margin="580,362,581,362" Background="#d9d9d9"/>
            <Button x:Name="confirmButton" Content="Подтвердить"
                HorizontalAlignment="Center" VerticalAlignment="Top" Width="124" Height="28"
                Margin="0,429,0,0" Click="confirmButton_Click" Style="{StaticResource DefaultButton}"
                FontSize="20" Visibility="Visible" BorderBrush="Black"
                BorderThickness="1"/>
            <Button x:Name="BackButton" Style="{StaticResource DefaultButton}"
                Margin="10,10,1420,712" Visibility="Collapsed" Click="BackButton_Click">
                <Button.Background>
                    <ImageBrush
                        ImageSource="C:\Users\fona\source\repos\CourseWorkClient\Media\back.png" />
                </Button.Background>
            </Button>
            <Border x:Name="MainMenu" Padding="5" HorizontalAlignment="Center"
                CornerRadius="30" Background="White" Width="474" VerticalAlignment="Center"
                Height="124" Visibility="Collapsed">
                <Border.Effect>
```

```

        <DropShadowEffect          BlurRadius="37"          Color="DarkGray"
ShadowDepth="10"/>
    </Border.Effect>
    <StackPanel>
        <Button Content="Управление пользователями" Style="{StaticResource
DefaultButton}"
                FontSize="24" HorizontalAlignment="Center" Margin="0 15 0
0" Click="Button_ControlUsers_Click"/>
        <Rectangle Fill="#363636" Height="1" Margin="0 10 0 0"/>
        <Button Content="Управление меню" Style="{StaticResource
DefaultButton}"
                FontSize="24" HorizontalAlignment="Center" Margin="0 15 0
0" Click="Button_ControlMenu_Click"/>
        <Rectangle Fill="#363636" Height="1" Margin="0 10 0 0"/>
    </StackPanel>
</Border>
<Border x:Name="ControlUsersBorder"          Padding="10"
HorizontalAlignment="Center" CornerRadius="30"
        Background="White"          Width="1353"          VerticalAlignment="Center"
Height="580" Visibility="Collapsed">
    <Border.Effect>
        <DropShadowEffect          BlurRadius="37"          Color="DarkGray"
ShadowDepth="10"/>
    </Border.Effect>
    <ScrollViewer          VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">
        <StackPanel>
            <DataGrid x:Name="usersDataGrid" AutoGenerateColumns="True"
HorizontalAlignment="Stretch" VerticalAlignment="Stretch" />
        </StackPanel>
    </ScrollViewer>
</Border>
<Border x:Name="ControlMenuBorder"          Padding="10"
HorizontalAlignment="Center" CornerRadius="30"
        Background="White"          Width="1353"          VerticalAlignment="Center"
Height="580" Visibility="Collapsed">
    <Border.Effect>
        <DropShadowEffect          BlurRadius="37"          Color="DarkGray"
ShadowDepth="10"/>
    </Border.Effect>
    <ScrollViewer          VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">
        <StackPanel>
            <DataGrid x:Name="menuDataGrid" AutoGenerateColumns="True"
HorizontalAlignment="Stretch" VerticalAlignment="Stretch" />
        </StackPanel>
    </ScrollViewer>
</Border>
</Grid>
</ScrollViewer>
</Window>

```

"Admin.xaml.cs"

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;

```

```

using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using static System.Net.Mime.MediaTypeNames;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Admin.xaml
    /// </summary>
    public partial class Admin : Window
    {
        public ObservableCollection<User> Users { get; set; }
        public ObservableCollection<Product> Products { get; set; }
        public Admin()
        {
            InitializeComponent();

            private async void Button_ControlUsers_Click(object sender, RoutedEventArgs e)
            {
                MainMenu.Visibility = Visibility.Collapsed;
                BackToMain.Visibility = Visibility.Collapsed;
                ControlUsersBorder.Visibility = Visibility.Visible;
                IdTextBox.Visibility = Visibility.Visible;
                delUserButton.Visibility = Visibility.Visible;
                BackButton.Visibility = Visibility.Visible;
                Users = await
                ApiClient.Instance.GetFromJsonAsync<ObservableCollection<User>>("Users") ?? new
                ObservableCollection<User>();
                usersDataGrid.ItemsSource = Users;
                usersDataGrid.CellEditEnding += async (s, ev) =>
                {
                    if (ev.Row.Item is User editedUser)
                    {
                        var json = System.Text.Json.JsonSerializer.Serialize(editedUser);
                        var content = new StringContent(json, Encoding.UTF8,
                        "application/json");
                        var response = await ApiClient.Instance.PutAsync("Users", content);
                    }
                };

            protected override void OnClosed(EventArgs e)
            {
                base.OnClosed(e);
            }

            private async void delUserButton_Click(object sender, RoutedEventArgs e)
            {
                int userId;
                if (int.TryParse(IdTextBox.Text, out userId))
                {
                    var response = await ApiClient.Instance.DeleteAsync($"Users/{userId}");
                    if (response.StatusCode == System.Net.HttpStatusCode.NoContent)
                        MessageBox.Show("Пользователь удален.");
                    else if (response.StatusCode == System.Net.HttpStatusCode.NotFound)
                        MessageBox.Show("Такого пользователя не найдено. Введите корректный ID.");
                }
            }

            private void BackButton_Click(object sender, RoutedEventArgs e)
            {
                MainMenu.Visibility = Visibility.Visible;
                BackToMain.Visibility = Visibility.Visible;
                ControlUsersBorder.Visibility = Visibility.Collapsed;
                IdTextBox.Visibility = Visibility.Collapsed;
                delUserButton.Visibility = Visibility.Collapsed;
                BackButton.Visibility = Visibility.Collapsed;
                ControlMenuBorder.Visibility = Visibility.Collapsed;
            }
        }
    }
}

```

```

        delMenuButton.Visibility = Visibility.Collapsed;
        CreateMenuButton.Visibility = Visibility.Collapsed;
    }

    private async void Button_ControlMenu_Click(object sender, RoutedEventArgs e)
    {
        MainMenu.Visibility = Visibility.Collapsed;
        BackToMain.Visibility = Visibility.Collapsed;
        ControlMenuBorder.Visibility = Visibility.Visible;
        IdTextBox.Visibility = Visibility.Visible;
        delMenuButton.Visibility = Visibility.Visible;
        CreateMenuButton.Visibility = Visibility.Visible;
        BackButton.Visibility = Visibility.Visible;
        Products = await
        ApiClient.Instance.GetFromJsonAsync<ObservableCollection<Product>>("Products") ?? new
        ObservableCollection<Product>();
        menuDataGrid.ItemsSource = Products;
        menuDataGrid.CellEditEnding += async (s, ev) =>
        {
            if (ev.Row.Item is Product editedProd)
            {
                var json = System.Text.Json.JsonSerializer.Serialize(editedProd);
                var content = new StringContent(json, Encoding.UTF8,
                "application/json");
                var response = await ApiClient.Instance.PutAsync("Products/edit",
                content);
            }
        };
    }

    private async void delMenuButton_Click(object sender, RoutedEventArgs e)
    {
        int productId;
        if (int.TryParse(IdTextBox.Text, out productId))
        {
            var response = await
            ApiClient.Instance.DeleteAsync($"Products/{productId}");
            if (response.StatusCode == System.Net.HttpStatusCode.NoContent)
            MessageBox.Show("Товар удален.");
            else if (response.StatusCode == System.Net.HttpStatusCode.NotFound)
            MessageBox.Show("Такого товара не найдено. Введите корректный ID.");
        }
    }

    private async void CreateMenuButton_Click(object sender, RoutedEventArgs e)
    {
        var newProduct = new Product
        {
            Name = "Новый продукт",
            Image =
            "C:/Users/fona/source/repos/coursework3/Media/No_image_available.svg.png"
        };
        var json = JsonConvert.SerializeObject(newProduct);
        var content = new StringContent(json, Encoding.UTF8, "application/json");
        var response = await ApiClient.Instance.PostAsync($"Products", content);
        Products.Add(await
        ApiClient.Instance.GetFromJsonAsync<Product>($"Products/getlastprod"));
    }

    private void BackToMain_Click(object sender, RoutedEventArgs e)
    {
        Menu menu = new Menu();
        menu.Show();
        Close();
    }

    private void confirmButton_Click(object sender, RoutedEventArgs e)
    {
        if (AdminPassword.Password.Trim() == "22122005")
    }

```



```

        {
            confirmButton.Visibility = Visibility.Collapsed;
            AdminPassword.Visibility = Visibility.Collapsed;
            MainMenu.Visibility = Visibility.Visible;
        }
        else MessageBox.Show("Пароль неверный.\nЕсли вы попали сюда случайно, лучше
вернитесь в приложение :)");
    }
}
}

```

## "App.xaml"

```

<Application x:Class="CourseWorkClient.App"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:CourseWorkClient"
    xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
    StartupUri="Menu.xaml">
    <Application.Resources>
        <ResourceDictionary>
            <ResourceDictionary.MergedDictionaries>
                <ResourceDictionary
Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />
                <ResourceDictionary
Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesign3.Defaults.xaml" />
                <ResourceDictionary
Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.DeepPurple.xaml" />
                <ResourceDictionary
Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Secondary/MaterialDesignColor.Lime.xaml" />
            </ResourceDictionary.MergedDictionaries>
            <Style TargetType="GroupBox" x:Key="DefaultGroupBox">
                <Setter Property="Background" Value="Transparent"/>
                <Setter Property="BorderThickness" Value="0"/>
                <Setter Property="HeaderTemplate">
                    <Setter.Value>
                        <DataTemplate>
                            <TextBlock Text="{Binding}"
                                Background="Transparent"
                                Foreground="Black"
                                Padding="0"
                                HorizontalAlignment="Left"/>
                        </DataTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
            <Style x:Key="TypeInTextBox" TargetType="TextBox" BasedOn="{StaticResource MaterialDesignFloatingHintTextBox}">
                <Setter Property="local:TextBoxHelper.IsErrorState" Value="False" />
                <Style.Triggers>
                    <MultiTrigger>
                        <MultiTrigger.Conditions>
                            <Condition Property="IsKeyboardFocused" Value="True" />
                            <Condition Property="local:TextBoxHelper.IsErrorState"
Value="False" />
                        </MultiTrigger.Conditions>
                        <Setter Property="materialDesign:HintAssist.Foreground"
Value="Black" />
                        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="Black" />
                    </MultiTrigger>
                </Style.Triggers>
            </Style>
            <MultiTrigger>
                <MultiTrigger.Conditions>
                    <Condition Property="IsKeyboardFocused" Value="False" />

```

```

Value="False" />
        <Condition Property="local:TextBoxHelper.IsErrorState"
        <Condition Property="IsMouseOver" Value="True" />
    </MultiTrigger.Conditions>
    <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#3d3d3d" />
    <Setter Property="materialDesign:HintAssist.Foreground"
Value="#7F000000" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="False" />
            <Condition Property="local:TextBoxHelper.IsErrorState"
Value="True" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#bd4040" />
        <Setter Property="materialDesign:HintAssist.Foreground"
Value="#bd4040" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="True" />
            <Condition Property="local:TextBoxHelper.IsErrorState"
Value="True" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#C23030" />
        <Setter Property="materialDesign:HintAssist.Foreground"
Value="#C23030" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="False" />
            <Condition Property="local:TextBoxHelper.IsErrorState"
Value="True" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#bd4040" />
        <Setter Property="materialDesign:HintAssist.Foreground"
Value="#bd4040" />
    </MultiTrigger>
</Style.Triggers>
</Style>
<Style x:Key="TypeInPasswordBox" TargetType="PasswordBox"
BasedOn="{StaticResource MaterialDesignFloatingHintPasswordBox}">
    <Setter Property="local:PasswordBoxHelper.IsErrorState" Value="False"
/>
    <Style.Triggers>
        <MultiTrigger>
            <MultiTrigger.Conditions>
                <Condition Property="IsKeyboardFocused" Value="True" />
                <Condition Property="local:PasswordBoxHelper.IsErrorState"
Value="False" />
            </MultiTrigger.Conditions>
            <Setter
Property="materialDesign:HintAssist.Foreground"
Value="Black" />
        </MultiTrigger>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="Black" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="False" />
            <Condition Property="local:PasswordBoxHelper.IsErrorState"
Value="False" />
        </MultiTrigger.Conditions>
        <Condition Property="IsMouseOver" Value="True" />
    </MultiTrigger>

```

```

        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#3d3d3d" />
        <Setter
Property="materialDesign:HintAssist.Foreground"
Value="#7F000000" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="False" />
            <Condition Property="local:PasswordBoxHelper.IsErrorState"
Value="True" />
            <Condition Property="IsMouseOver" Value="True" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#bd4040" />
        <Setter
Property="materialDesign:HintAssist.Foreground"
Value="#bd4040" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="True" />
            <Condition Property="local:PasswordBoxHelper.IsErrorState"
Value="True" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#C23030" />
        <Setter
Property="materialDesign:HintAssist.Foreground"
Value="#C23030" />
    </MultiTrigger>
    <MultiTrigger>
        <MultiTrigger.Conditions>
            <Condition Property="IsKeyboardFocused" Value="False" />
            <Condition Property="local:PasswordBoxHelper.IsErrorState"
Value="True" />
            <Condition Property="IsMouseOver" Value="False" />
        </MultiTrigger.Conditions>
        <Setter
Property="materialDesign:TextFieldAssist.UnderlineBrush" Value="#bd4040" />
        <Setter
Property="materialDesign:HintAssist.Foreground"
Value="#bd4040" />
    </MultiTrigger>
</Style.Triggers>
</Style>
<Style TargetType="RadioButton" x:Key="DefaultRadioButton">
    <Setter Property="Foreground" Value="Transparent"/>
    <Setter Property="Cursor" Value="Hand"/>
    <Setter Property="Foreground" Value="Black" />
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="RadioButton">
                <ControlTemplate.Resources>
                    <BooleanToVisibilityConverter
x:Key="BooleanToVisibilityConverter" />
                </ControlTemplate.Resources>
                <StackPanel Orientation="Horizontal">
                    <Grid Width="20" Height="20">
                        <Ellipse x:Name="Ellipse" Width="13" Height="13"
Stroke="#fad12f" StrokeThickness="1.5">
                            <Ellipse.Fill>
                                <SolidColorBrush Color="Transparent"/>
                            </Ellipse.Fill>
                        </Ellipse>
                        <Ellipse Width="6.5" Height="6.5" Fill="#fad12f"
Visibility="{TemplateBinding IsChecked, Converter={StaticResource
BooleanToVisibilityConverter}}"/>
                    </Grid>
                    <TextBlock
Margin="8,0,0,0" VerticalAlignment="Center"/>
                </StackPanel>
            </ControlTemplate>
        </Setter.Value>
    </Setter>

```

```

        </Setter>
    </Style>
    <Style x:Key="RedButton" TargetType="Button">
        <Setter Property="Effect">
            <Setter.Value>
                <DropShadowEffect          Color="DarkGray"          BlurRadius="4"
ShadowDepth="2" />
            </Setter.Value>
        </Setter>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Grid>
                        <Border x:Name="border"
CornerRadius="2" Background="{TemplateBinding Background}"
BorderThickness="0" BorderBrush="Transparent">
                            <ContentPresenter          Content="{TemplateBinding
Content}"
                                HorizontalAlignment="Center" VerticalAlignment="Center"
                                Focusable="False" />
                        </Border>
                    </Grid>
                    <ControlTemplate.Triggers>
                        <Trigger Property="IsPressed" Value="True">
                            <Trigger.EnterActions>
                                <BeginStoryboard>
                                    <Storyboard>
                                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                                            To="#781717"
                                            Duration="0:0:0.0000001" />
                                    </Storyboard>
                                </BeginStoryboard>
                            </Trigger.EnterActions>
                            <Trigger.ExitActions>
                                <BeginStoryboard>
                                    <Storyboard>
                                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
                                            To="Firebrick"
                                            Duration="0:0:0.15" />
                                    </Storyboard>
                                </BeginStoryboard>
                            </Trigger.ExitActions>
                        </Trigger>
                        <Trigger Property="IsPressed" Value="True">
                            <Setter Property="Background" Value="Transparent"/>
                        </Trigger>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>
    <Style x:Key="GrayButton" TargetType="Button">
        <Setter Property="Effect">
            <Setter.Value>
                <DropShadowEffect          Color="DarkGray"          BlurRadius="4"
ShadowDepth="2" />
            </Setter.Value>
        </Setter>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Grid>
                        <Border x:Name="border"
CornerRadius="3" Background="{TemplateBinding Background}"

```

```

        BorderThickness="0" BorderBrush="Transparent">
            <ContentPresenter Content="{TemplateBinding
Content}"
        HorizontalAlignment="Center" VerticalAlignment="Center"
        Focusable="False" />
        </Border>
    </Grid>
    <ControlTemplate.Triggers>
        <Trigger Property="IsPressed" Value="True">
            <Trigger.EnterActions>
                <BeginStoryboard>
                    <Storyboard>
                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#8f8f8f"
Duration="0:0:0.0000001" />
                    </Storyboard>
                </BeginStoryboard>
            </Trigger.EnterActions>
            <Trigger.ExitActions>
                <BeginStoryboard>
                    <Storyboard>
                        <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#c2c2c2"
Duration="0:0:0.15" />
                    </Storyboard>
                </BeginStoryboard>
            </Trigger.ExitActions>
        </Trigger>
        <Trigger Property="IsPressed" Value="True">
            <Setter Property="Background" Value="Transparent"/>
        </Trigger>
    </ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
<Style x:Key="YellowButton" TargetType="Button">
    <Setter Property="Effect">
        <Setter.Value>
            <DropShadowEffect Color="DarkGray" BlurRadius="4"
ShadowDepth="2" />
        </Setter.Value>
    </Setter>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Grid>
                    <Border x:Name="border"
CornerRadius="2" Background="{TemplateBinding Background}"
BorderThickness="0" BorderBrush="Transparent">
                        <ContentPresenter Content="{TemplateBinding
Content}"
                    HorizontalAlignment="Center" VerticalAlignment="Center"
                    Focusable="False" />
                    </Border>
                </Grid>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsPressed" Value="True">
                        <Trigger.EnterActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#8f8f8f"
Duration="0:0:0.0000001" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.EnterActions>
                        <Trigger.ExitActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#c2c2c2"
Duration="0:0:0.15" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.ExitActions>
                    </Trigger>
                    <Trigger Property="IsPressed" Value="True">
                        <Setter Property="Background" Value="Transparent"/>
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>
<Style x:Key="YellowButton" TargetType="Button">
    <Setter Property="Effect">
        <Setter.Value>
            <DropShadowEffect Color="DarkGray" BlurRadius="4"
ShadowDepth="2" />
        </Setter.Value>
    </Setter>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Grid>
                    <Border x:Name="border"
CornerRadius="2" Background="{TemplateBinding Background}"
BorderThickness="0" BorderBrush="Transparent">
                        <ContentPresenter Content="{TemplateBinding
Content}"
                    HorizontalAlignment="Center" VerticalAlignment="Center"
                    Focusable="False" />
                    </Border>
                </Grid>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsPressed" Value="True">
                        <Trigger.EnterActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#8f8f8f"
Duration="0:0:0.0000001" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.EnterActions>
                        <Trigger.ExitActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#c2c2c2"
Duration="0:0:0.15" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.ExitActions>
                    </Trigger>
                    <Trigger Property="IsPressed" Value="True">
                        <Setter Property="Background" Value="Transparent"/>
                    </Trigger>
                </ControlTemplate.Triggers>
            </ControlTemplate>
        </Setter.Value>
    </Setter>
</Style>

```

```

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
    To="#ad9126"
    Duration="0:0:0.0000001" />
        </Storyboard>
    </BeginStoryboard>
</Trigger.EnterActions>
<Trigger.ExitActions>
    <BeginStoryboard>
        <Storyboard>
            <ColorAnimation
Storyboard.TargetName="border"

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
    To="#f7ce36"
    Duration="0:0:0.15" />
        </Storyboard>
    </BeginStoryboard>
    </Trigger.ExitActions>
</Trigger>
    <Trigger Property="IsPressed" Value="True">
        <Setter Property="Background" Value="Transparent"/>
    </Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
    </Setter.Value>
</Setter>
</Style>
<Style x:Key="GreenButton" TargetType="Button">
    <Setter Property="Effect">
        <Setter.Value>
            <DropShadowEffect          Color="DarkGray"          BlurRadius="4"
ShadowDepth="2" />
        </Setter.Value>
    </Setter>
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="Button">
                <Grid>
                    <Border x:Name="border"
CornerRadius="2" Background="{TemplateBinding Background}"
BorderThickness="0" BorderBrush="Transparent">
                        <ContentPresenter          Content="{TemplateBinding
Content}"
                            HorizontalAlignment="Center" VerticalAlignment="Center"
                            Focusable="False" />
                    </Border>
                </Grid>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsPressed" Value="True">
                        <Trigger.EnterActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
    To="#224d05"
    Duration="0:0:0.0000001" />
        </Storyboard>
    </BeginStoryboard>
    </Trigger.EnterActions>
    <Trigger.ExitActions>
        <BeginStoryboard>
            <Storyboard>
                <ColorAnimation
Storyboard.TargetName="border"

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
    To="#40910a"

```

```

        Duration="0:0:0.15" />
    </Storyboard>
</BeginStoryboard>
</Trigger.ExitActions>
</Trigger>
<Trigger Property="IsPressed" Value="True">
    <Setter Property="Background" Value="Transparent"/>
</Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
<Style x:Key="AmountChoice" TargetType="RadioButton">
    <Setter Property="Background" Value="LightGray"/>
    <Setter Property="Foreground" Value="Black"/>
    <Setter Property="Width" Value="100"/>
    <Setter Property="Height" Value="50"/>
    <Setter Property="Margin" Value="5"/>
    <Setter Property="FontFamily" Value="Faberge" />
    <Setter Property="FontSize" Value="17" />
    <Setter Property="Cursor" Value="Hand" />
    <Setter Property="Template">
        <Setter.Value>
            <ControlTemplate TargetType="RadioButton">
                <Border x:Name="border" Background="{TemplateBinding
Background}"
                    CornerRadius="5">
                    <ContentPresenter HorizontalAlignment="Center"
                        VerticalAlignment="Center" />
                </Border>
                <ControlTemplate.Triggers>
                    <Trigger Property="IsChecked" Value="True">
                        <Setter TargetName="border" Property="Background"
Value="#f7ce36"/>
                        <Setter Property="Foreground" Value="Black"/>
                    </Trigger>
                    <Trigger Property="IsPressed" Value="True">
                        <Trigger.EnterActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#ad9126"
Duration="0:0:0.0000001" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.EnterActions>
                        <Trigger.ExitActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <ColorAnimation
Storyboard.TargetName="border"
Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"
To="#f7ce36"
Duration="0:0:0.15" />
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.ExitActions>
                    </ControlTemplate.Triggers>
                </ControlTemplate>
            </Setter.Value>
        </Setter>
    </Style>
<Style x:Key="DefaultButton" TargetType="Button" BasedOn="{StaticResource
RedButton}">
    <Setter Property="Background" Value="Transparent"/>
    <Setter Property="Foreground" Value="Black"/>

```

```

        <Setter Property="BorderBrush" Value="Transparent"/>
        <Setter Property="BorderThickness" Value="0"/>
        <Setter Property="Cursor" Value="Hand"/>
        <Setter Property="RenderTransformOrigin" Value="0.5,0.5"/>
        <Setter Property="Effect" Value="{x:Null}" />
        <Setter Property="RenderTransform">
            <Setter.Value>
                <ScaleTransform ScaleX="1" ScaleY="1"/>
            </Setter.Value>
        </Setter>
        <Setter Property="Template">
            <Setter.Value>
                <ControlTemplate TargetType="Button">
                    <Border Background="{TemplateBinding Background}"
                        BorderBrush="{TemplateBinding BorderBrush}"
                        BorderThickness="{TemplateBinding
BorderThickness}">
                                <ContentPresenter Content="{TemplateBinding Content}"
/>
                            </Border>
                        </ControlTemplate>
                    </Setter.Value>
                </Setter>
            </Style>
            <Style
                x:Key="AnimButton"      TargetType="Button"      BasedOn="{StaticResource
DefaultButton}">
                <Style.Triggers>
                    <Trigger Property="IsMouseOver" Value="True">
                        <Setter Property="Background" Value="Transparent"/>
                        <Trigger.EnterActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <DoubleAnimation
Storyboard.TargetProperty="RenderTransform.(ScaleTransform.ScaleX)"
To="1.15" Duration="0:0:0.1"/>
                                    <DoubleAnimation
Storyboard.TargetProperty="RenderTransform.(ScaleTransform.ScaleY)"
To="1.15" Duration="0:0:0.1"/>
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.EnterActions>
                        <Trigger.ExitActions>
                            <BeginStoryboard>
                                <Storyboard>
                                    <DoubleAnimation
Storyboard.TargetProperty="RenderTransform.(ScaleTransform.ScaleX)"
To="1" Duration="0:0:0.1"/>
                                    <DoubleAnimation
Storyboard.TargetProperty="RenderTransform.(ScaleTransform.ScaleY)"
To="1" Duration="0:0:0.1"/>
                                </Storyboard>
                            </BeginStoryboard>
                        </Trigger.ExitActions>
                    </Trigger>
                </Style.Triggers>
            </Style>
        </ResourceDictionary>
    </Application.Resources>
</Application>

```

## "App.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Linq;
using System.Threading.Tasks;

```



```

using System.Windows.Controls;
using System.Windows;

namespace CourseWorkClient
{
    public static class AuthState
    {
        public static bool isAuthenticated { get; set; }
        public static int Id { get; set; }
    }

    public static class PasswordBoxHelper
    {
        public static readonly DependencyProperty IsErrorStateProperty =
            DependencyProperty.RegisterAttached(
                "IsErrorState",
                typeof(bool),
                typeof(PasswordBoxHelper),
                new PropertyMetadata(false));

        public static bool GetIsErrorState(DependencyObject obj)
        {
            return (bool)obj.GetValue(IsErrorStateProperty);
        }

        public static void SetIsErrorState(DependencyObject obj, bool value)
        {
            obj.SetValue(IsErrorStateProperty, value);
        }
    }

    public static class TextBoxHelper
    {
        public static readonly DependencyProperty IsErrorStateProperty =
            DependencyProperty.RegisterAttached(
                "IsErrorState",
                typeof(bool),
                typeof(TextBoxHelper),
                new PropertyMetadata(false));

        public static bool GetIsErrorState(DependencyObject obj)
        {
            return (bool)obj.GetValue(IsErrorStateProperty);
        }

        public static void SetIsErrorState(DependencyObject obj, bool value)
        {
            obj.SetValue(IsErrorStateProperty, value);
        }
    }

    /// <summary>
    /// Логика взаимодействия для App.xaml
    /// </summary>
    public partial class App : Application
    {
    }
}

```

"Cart.xaml"

```

<Window x:Class="CourseWorkClient.Cart"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"

```

```

MinHeight="200" MinWidth="270"
Width="1505" Height="2144"
WindowState="Maximized"
Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <Window.Background>
        <ImageBrush
            ImageSource="C:\Users\fona\source\repos\CourseWorkClient\Media\maclogo.jpg"
            TileMode="None" Stretch="None" AlignmentY="Top"/>
        </Window.Background>
        <ScrollViewer
            HorizontalScrollBarVisibility="Auto"
            VerticalScrollBarVisibility="Auto">
            <Border>
                <StackPanel x:Name="mainStackPanel">
                    <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top"
                        Margin="-28,-36,-30,0">
                        <Border.Effect>
                            <DropShadowEffect
                                BlurRadius="15" Opacity="0.15"
                                ShadowDepth="1"/>
                        </Border.Effect>
                        <StackPanel>
                            <Image
                                Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
                                Width="133" Margin="0,47,1330,0" Height="73"/>
                            <Button Content="Меню" FontSize="35" FontFamily="Faberge"
                                Width="98"
                                Margin="0,-60,1000,0" Height="50" FontWeight="Regular"
                                Foreground="#424242"
                                Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                            <Button Content="Рестораны" FontSize="35" FontFamily="Faberge"
                                Width="192"
                                Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
                                FontWeight="Regular"
                                Click="Button_Restaurants_Click" Foreground="#424242"/>
                            <Button Content="Корзина" FontSize="35" FontFamily="Faberge"
                                Width="148"
                                Margin="0,-60,-20,0" Height="50"
                                Style="{StaticResource DefaultButton}" FontWeight="Bold"
                                Click="Button_Cart_Click"/>
                            <Button Content="История заказов" FontSize="35"
                                FontFamily="Faberge" Width="283"
                                Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
                                Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
                            <Button x:Name="Button_Authorization" Content="Вход"
                                FontSize="35" FontFamily="Faberge" Width="91"
                                Margin="0,-60,-1260,0" Height="50"
                                Foreground="#424242" HorizontalContentAlignment="Center"
                                VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
                                Click="Button_Authorization_Click"/>
                            <TextBlock x:Name="TextBlock_Authorization" Text="Здравствуйте,
                                " FontSize="35" FontFamily="Faberge"
                                Width="221" Margin="0,-75,-1260,0" Height="90" TextAlignment="Right"
                                Foreground="#424242" HorizontalAlignment="Center" Visibility="Hidden"/>
                        </StackPanel>
                    </Border>
                    <Border x:Name="borderCart" Visibility="Hidden" Padding="30"
                        HorizontalAlignment="Center" CornerRadius="30"
                        Background="White" Margin="0,75,0,0">
                        <Border.Effect>
                            <DropShadowEffect
                                BlurRadius="37" Color="DarkGray"
                                ShadowDepth="10"/>
                        </Border.Effect>
                        <StackPanel x:Name="cartStackPanel" Margin="-30,-30,-30,-30">
                            </StackPanel>
                        </Border>
                        <TextBlock Visibility="Hidden" Text="Для красивого вида!"
                            Margin="0,75,0,0"/>
                    </StackPanel>
                </Border>
            </ScrollViewer>

```

</Window>

## "Cart.xaml.cs"

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Globalization;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Security.Policy;
using System.Text;
using System.Text.Json;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Media.Effects;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.Xml.Linq;
using static MaterialDesignThemes.Wpf.Theme.ToolBar;
using static System.Net.Mime.MediaTypeNames;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Cart.xaml
    /// </summary>
    public partial class Cart : Window
    {
        private List<string> suggestions;

        public Cart()
        {
            InitializeComponent();
            suggestions = new List<string>
            {
                "Логойский тракт, 35", "ул. Шугаева, 2", "ул.П.Мстиславца, 11-4, пом.68",
                "ул. Сурганова, 63", "пр. Независимости, 52", "ул. Тимирязева, 76",
                "ТЦ Green City (ул. Притыцкого, 156-9)", "ул. Притыцкого, 58", "ул. Притыцкого, 28",
                "ул. Шаранговича, 23", "пр-т Дзержинского, 59", "пр. Дзержинского, 96",
                "ТРЦ DiaMond city (Щомыслицкий с/с, 32/4, пом. 209, Минская область, Минский район)",
                "ТЦ МОМО (пр. Партизанский, 150А)", "ул. Уборевича, 93", "ул. Максима Танка, 2",
                "ул. Немига, 12Б", "пр. Независимости, 23", "ул. Кирова, 1", "ул. Бобруйская, 6 (ТРЦ Galileo)",
                "г. Могилев, пр. Пушкинский, 12 (ТЦ «Арбат»)", "г. Витесбк, ул. Терешковой, 1", "г. Гомель, ул. Советская, 68",
                "г. Брест, ул. Октябрьской революции, д. 1а.", "г. Гродно, ул. Горновых, 9", "г. Гродно, ТРК TRINITY (проспект Янки Купалы 87, 3 этаж)"
            };

            if (AuthState.isAuthenticated)
            {
                Button_Authorization.Visibility = Visibility.Hidden;
                TextBox_Authorization.Visibility = Visibility.Visible;
                borderCart.Visibility = Visibility.Visible;
                showProducts();
            }
            if (!AuthState.isAuthenticated)
```

```

        {
            TextBlock error = new TextBlock
            {
                Text = "Вы не вошли в аккаунт!",
                FontSize = 35,
                FontFamily = new FontFamily("Faberger"),
                HorizontalAlignment = HorizontalAlignment.Center,
                Background = new SolidColorBrush(Color.FromRgb(255, 255, 255)),
                Width = 400,
                Height = 60,
                TextAlignment = TextAlignment.Center,
                Padding = new Thickness(0, 10, 0, 0)
            };
            error.Margin = new Thickness(0, 100, 0, 0);
            mainStackPanel.Children.Add(error);
        }

        if (AuthState.IsAuthenticated) Loaded += async (s, e) => await
MakeNiceName();
    }
    private string restaurantName = null;
    private string PaymentMethod = null;
    private async Task MakeNiceName()
    {
        var username = await
ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getname");
        if (username.Length > 13) TextBlock_Authorization.Text =
$"Здравствуйте, \n{username.Substring(0, 11)}...";
        else TextBlock_Authorization.Text = $"Здравствуйте, \n{username}";
    }
    private async void showProducts()
    {
        var cart = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Users/{AuthState.Id}/getcart");
        double result = 0;

        if (cart != "" && cart != null)
        {
            string[] prods = cart.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
            foreach (string prod in prods)
            {
                if (prod != null)
                {
                    string[] parts = prod.Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                    Product product = new Product();
                    string temp = parts[0];
                    if (parts.Length == 3) temp = Regex.Replace(temp, @"\"s*(.*?)\"",
""");
                    var response = await
ApiClient.Instance.GetAsync($"Products/{temp}");
                    if (response.IsSuccessStatusCode)
                    {
                        product = await
response.Content.ReadFromJsonAsync<Product>();
                        System.Windows.Controls.Image image = new
System.Windows.Controls.Image
                        {
                            Source = new BitmapImage(new Uri($"{product.Image}")),
                            Width = 90,
                            Height = 90,
                            HorizontalAlignment = HorizontalAlignment.Left,
                            VerticalAlignment = VerticalAlignment.Center
                        };
                        image.Margin = new Thickness(35, 0, 0, 0);
                        TextBlock textblock = new TextBlock
                        {
                            Text = $"{parts[0]}",
                            FontSize = 25,

```

```

        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Center,
        TextAlignment = TextAlignment.Left,
        TextWrapping = TextWrapping.Wrap,
        Width = 250
    };
    textblock.Margin = new Thickness(160, 0, 0, 0);
    TextBlock amount = new TextBlock
    {
        Text = $"x{parts[1]}",
        FontSize = 25,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Right,
        VerticalAlignment = VerticalAlignment.Center,
        TextAlignment = TextAlignment.Center
    };
    amount.Margin = new Thickness(0, 0, 165, 0);
    TextBlock cost = new TextBlock
    {
        FontSize = 30,
        FontWeight = FontWeights.Bold,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Right,
        VerticalAlignment = VerticalAlignment.Center,
        TextAlignment = TextAlignment.Center
    };
    if (product.Cost.Contains("or")) cost.Text = $"{parts[2]}";
    else cost.Text = $"{product.Cost}";
    cost.Margin = new Thickness(0, 0, 40, 0);
    Button openItemInfo = new Button
    {
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Center,
        Margin = new Thickness(35, 0, 0, 0),
        Width = 340,
        Height = 90,
        Style = (Style)FindResource("DefaultButton"),
        Background = Brushes.Transparent,
        BorderThickness = new Thickness(0),
        Foreground = Brushes.Transparent
    };
    openItemInfo.Click += (sender, e) =>
    {
        Item item = new Item(product);
        item.Show();
        Close();
    };
    Grid itemCounter = new Grid
    {
        VerticalAlignment = VerticalAlignment.Center,
        HorizontalAlignment = HorizontalAlignment.Right,
        Margin = new Thickness(0, 0, 200, 0),
        Width = 40,
        Height = 30
    };
    itemCounter.ColumnDefinitions.Add(new ColumnDefinition());
    itemCounter.ColumnDefinitions.Add(new ColumnDefinition());
    Button decrementButton = new Button
    {
        Style = (Style)FindResource("GrayButton"),
        Background = new SolidColorBrush(Color.FromRgb(194,
194, 194)),
        Content = "-",
        FontSize = 30,
        Cursor = Cursors.Hand,
        FontFamily = new FontFamily("Faberger"),
        VerticalAlignment = VerticalAlignment.Center
    };
    decrementButton.Click += async (sender, e) =>

```

```

        {
            int indexToChange = Array.IndexOf(prods,
prods.FirstOrDefault(n => n.Contains(temp)));
            if (Convert.ToInt16(parts[1]) != 1)
            {
                parts[1] = (Convert.ToInt16(parts[1]) -
1).ToString();
                if (parts.Length == 3) prods[indexToChange] =
$"{{parts[0]}} - {{parts[1]}} - {{parts[2]}}";
                else prods[indexToChange] = $"{{parts[0]}} -
{{parts[1]}}";
                cart = null;
                foreach (string newprod in prods)
                {
                    cart += $"{{newprod}}\n";
                }
            }
            else
            {
                string[] newProds = prods.Where((val, idx) => idx
!= indexToChange).ToArray();
                cart = null;
                foreach (string newprod in newProds)
                {
                    cart += $"{{newprod}}\n";
                }
            }
            var json = JsonSerializer.Serialize(cart);
            var content = new StringContent(json, Encoding.UTF8,
"application/json");
            await
ApiClient.Instance.PutAsync($"Users/{AuthState.Id}/cart/add", content);
            cartStackPanel.Children.Clear();
            showProducts();
        };
        Grid.SetColumn(decrementButton, 0);
        itemCounter.Children.Add(decrementButton);
        Button incrementButton = new Button
        {
            Style = (Style)FindResource("GrayButton"),
            Background = new SolidColorBrush(Color.FromRgb(194,
194, 194)),
            Content = "+",
            FontSize = 30,
            Cursor = Cursors.Hand,
            FontFamily = new FontFamily("Faberger"),
            VerticalAlignment = VerticalAlignment.Center
        };
        incrementButton.Click += async (sender, e) =>
        {
            if (Convert.ToInt16(parts[1]) != 99)
            {
                int indexToChange = Array.IndexOf(prods,
prods.FirstOrDefault(n => n.Contains(temp)));
                parts[1] = (Convert.ToInt16(parts[1]) +
1).ToString();
                if (parts.Length == 3) prods[indexToChange] =
$"{{parts[0]}} - {{parts[1]}} - {{parts[2]}}";
                else prods[indexToChange] = $"{{parts[0]}} -
{{parts[1]}}";
                cart = null;
                foreach (string newprod in prods)
                {
                    cart += $"{{newprod}}\n";
                }
                var json = JsonSerializer.Serialize(cart);
                var content = new StringContent(json,
Encoding.UTF8, "application/json");
                await
ApiClient.Instance.PutAsync($"Users/{AuthState.Id}/cart/add", content);

```

```

        cartStackPanel.Children.Clear();
        showProducts();
    }
};
Grid.SetColumn(incrementButton, 1);
itemCounter.Children.Add(incrementButton);

Grid newgrid = new Grid
{
    Width = 680,
    Height = 110
};

newgrid.Children.Add(image);
newgrid.Children.Add(textblock);
newgrid.Children.Add(amount);
newgrid.Children.Add(cost);
var purchasestatus = await
ApiClient.Instance.GetFromJsonAsync<int>($"Users/{AuthState.Id}/getpurchasestatus");
if (purchasestatus != 1 && purchasestatus != 2 &&
purchasestatus != 3) newgrid.Children.Add(itemCounter);
newgrid.Children.Add(openItemInfo);
cartStackPanel.Children.Add(newgrid);

if (product.Cost.Contains("ot")) result +=
Convert.ToInt16(parts[1]) * Convert.ToDouble(parts[2], CultureInfo.InvariantCulture);
else result += Convert.ToInt16(parts[1]) *
Convert.ToDouble(product.Cost, CultureInfo.InvariantCulture);
}
}
if (result != 0)
{
    TextBlock sum = new TextBlock
    {
        Text = $"Итого:\n{result} pyб.",
        FontSize = 38,
        FontWeight = FontWeights.Bold,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Right,
        VerticalAlignment = VerticalAlignment.Bottom,
        TextAlignment = TextAlignment.Right
    };
    sum.Margin = new Thickness(0, 0, 40, 20);
    Button purchaseButton = new Button
    {
        Width = 120,
        Height = 40,
        Content = "Оплатить",
        FontFamily = new FontFamily("Faberger"),
        FontSize = 23,
        Cursor = Cursors.Hand,
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Bottom,
        Margin = new Thickness(40, 0, 0, 20),
        Background = new SolidColorBrush(Color.FromRgb(247, 206, 54))
    };
    purchaseButton.Click += PurchaseButton_Click;
    Grid purchaseGrid = new Grid();
    purchaseButton.Style = (Style)FindResource("YellowButton");
    TextBlock declined = new TextBlock // -1
    {
        Text = "Заказ отклонен.",
        Foreground = new SolidColorBrush(Color.FromRgb(194, 48, 48)),
        FontSize = 22,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Bottom,
        Margin = new Thickness(170, 0, 0, 28)
    };
};

```

```

        TextBlock cooking = new TextBlock // 2
        {
            Text = "Готовим ваш заказ :)",
            Foreground = new SolidColorBrush(Color.FromRgb(64, 145, 10)),
            FontSize = 22,
            FontFamily = new FontFamily("Faberger"),
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Bottom,
            Margin = new Thickness(40, 0, 0, 24)
        };
        TextBlock ready = new TextBlock // 3
        {
            Text = "Заказ готов к получению.",
            Foreground = new SolidColorBrush(Color.FromRgb(64, 145, 10)),
            FontSize = 22,
            FontFamily = new FontFamily("Faberger"),
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Bottom,
            Margin = new Thickness(40, 0, 0, 24)
        };
        TextBlock waiting = new TextBlock // 1
        {
            Text = "Ожидание ответа ресторана...",
            Foreground = new SolidColorBrush(Color.FromRgb(112, 112, 112)),
            FontSize = 22,
            FontFamily = new FontFamily("Faberger"),
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Bottom,
            Margin = new Thickness(40, 0, 0, 24)
        };

        var purchasestatus = await
        ApiClient.Instance.GetFromJsonAsync<int>($"Users/{AuthState.Id}/getpurchasestatus");
        if (purchasestatus == -1)
        {
            purchaseGrid.Children.Add(declined);
        }
        purchaseGrid.Children.Add(sum);
        if (purchasestatus == 2)
        {
            purchaseGrid.Children.Add(cooking);
        }
        else if (purchasestatus == 3)
        {
            purchaseGrid.Children.Add(ready);
        }
        else if (purchasestatus == 1)
        {
            purchaseGrid.Children.Add(waiting);
        }
        else purchaseGrid.Children.Add(purchaseButton);
        cartStackPanel.Children.Add(purchaseGrid);
    }
}
if (cart == "" || cart == null || result == 0)
{
    TextBlock nullCartText = new TextBlock
    {
        Text = "Корзина пуста.",
        FontSize = 35,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Center,
        Width = 400,
        Height = 50,
        TextAlignment = TextAlignment.Center,
        Padding = new Thickness(0, 10, 0, 0)
    };
    cartStackPanel.Children.Add(nullCartText);
}

```



```

    }

    protected override void OnClosed(EventArgs e)
    {
        base.OnClosed(e);
    }

    private void PurchaseButton_Click(object sender, RoutedEventArgs e)
    {
        mainStackPanel.Children.Remove(borderCart);
        Border border = new Border
        {
            HorizontalAlignment = HorizontalAlignment.Center,
            Margin = new Thickness(0, 100, 0, 0),
            CornerRadius = new CornerRadius(30),
            Width = 505,
            Height = 304,
            Background = new SolidColorBrush(Color.FromRgb(255, 250, 250)),
            Effect = new DropShadowEffect
            {
                BlurRadius = 37,
                Color = Colors.DarkGray,
                ShadowDepth = 10
            }
        };
        Grid grid = new Grid();
        TextBlock textBlock = new TextBlock
        {
            Text = "Оплата",
            FontSize = 58,
            FontFamily = new FontFamily("Mustica Pro SemBd"),
            HorizontalAlignment = HorizontalAlignment.Center,
            Margin = new Thickness(0, -3, 0, 228)
        };
        grid.Children.Add(textBlock);

        Rectangle animRectangle = new Rectangle
        {
            Width = 135,
            Height = 5,
            Fill = Brushes.Firebrick,
            HorizontalAlignment = HorizontalAlignment.Left,
            Margin = new Thickness(125, 69, 0, 230)
        };

        DoubleAnimation widthAnimation = new DoubleAnimation
        {
            From = 0.0,
            To = 135.0,
            Duration = new Duration(TimeSpan.FromSeconds(1.65)),
            AccelerationRatio = 0.1,
            DecelerationRatio = 0.9,
            AutoReverse = false
        };
        Storyboard.SetTarget(widthAnimation, animRectangle);
        Storyboard.SetTargetProperty(widthAnimation,
PropertyPath(Rectangle.WidthProperty));
new

        Storyboard storyboard = new Storyboard();
        storyboard.Children.Add(widthAnimation);

        animRectangle.Loaded += (s, ev) => storyboard.Begin();

        grid.Children.Add(animRectangle);

        TextBox searchBox = new TextBox
        {
            Style = (Style)FindResource("TypeInTextBox"),
            Name = "SearchBox",
            VerticalAlignment = VerticalAlignment.Top,

```

```

        FontSize = 20,
        Margin = new Thickness(36, 93, 36, 0),
        Foreground = Brushes.Black,
        CaretBrush = Brushes.Black,
        SelectionBrush = new SolidColorBrush(Color.FromRgb(19, 12, 31)),
        Text = string.Empty
    };
    MaterialDesignThemes.Wpf.HintAssist.SetHint(searchBox, "Адрес
ресторана...");
    grid.Children.Add(searchBox);

    GroupBox groupBox = new GroupBox
    {
        Style = (Style)FindResource("DefaultGroupBox"),
        Header = "Способ оплаты:",
        FontSize = 20,
        FontFamily = new FontFamily("Faberger"),
        Margin = new Thickness(26, 153, 310, 78)
    };

    StackPanel stackPanel = new StackPanel();
    RadioButton radioButton1 = new RadioButton
    {
        Style = (Style)FindResource("DefaultRadioButton"),
        Content = "Безналичная",
        GroupName = "OptionGroup"
    };
    radioButton1.Checked += RadioButton_Checked;

    RadioButton radioButton2 = new RadioButton
    {
        Style = (Style)FindResource("DefaultRadioButton"),
        Content = "Наличная",
        GroupName = "OptionGroup"
    };
    radioButton2.Checked += RadioButton_Checked;

    stackPanel.Children.Add(radioButton1);
    stackPanel.Children.Add(radioButton2);
    groupBox.Content = stackPanel;

    Button button = new Button
    {
        Style = (Style)FindResource("YellowButton"),
        Content = "Оплатить",
        FontSize = 24,
        Height = 36,
        VerticalAlignment = VerticalAlignment.Bottom,
        FontFamily = new FontFamily("Faberger"),
        Cursor = System.Windows.Input.Cursors.Hand,
        Background = new SolidColorBrush(Color.FromRgb(250, 209, 47)),
        Margin = new Thickness(194, 0, 195, 23)
    };

    ListBox suggestionsBox = new ListBox
    {
        Margin = new Thickness(36, 131, 36, 26),
        FontSize = 15,
        Visibility = Visibility.Collapsed,
        Background = new SolidColorBrush(Color.FromRgb(255, 250, 250))
    };
    searchBox.TextChanged += (send, E) =>
    {
        string query = searchBox.Text;

        if (string.IsNullOrWhiteSpace(query))
        {
            suggestionsBox.Visibility = Visibility.Collapsed;
            suggestionsBox.ItemsSource = null;
            restaurantName = null;
        }
    }

```

```

        return;
    }

    var filteredSuggestions = suggestions
        .Where(s => s.StartsWith(query,
StringComparison.OrdinalIgnoreCase))
        .ToList();

    if (filteredSuggestions.Any())
    {
        suggestionsBox.Visibility = Visibility.Visible;
        suggestionsBox.ItemsSource = filteredSuggestions;
    }
    else
    {
        suggestionsBox.Visibility = Visibility.Collapsed;
        suggestionsBox.ItemsSource = null;
    }

    restaurantName = query;
};
suggestionsBox.SelectionChanged += (sendr, Ev) =>
{
    if (suggestionsBox.SelectedItem != null)
    {
        searchBox.Text = suggestionsBox.SelectedItem.ToString();
        suggestionsBox.Visibility = Visibility.Collapsed;
    }
};

TextBlock errorRestaurantName = new TextBlock
{
    Text = "Укажите ресторан.",
    Foreground = new SolidColorBrush(Color.FromRgb(194, 48, 48)),
    FontSize = 14,
    FontFamily = new FontFamily("Faberger"),
    Margin = new Thickness(36, 136, 36, 0),
    Visibility = Visibility.Collapsed
};
TextBlock errorRadioButton = new TextBlock
{
    Text = "Укажите способ оплаты.",
    Foreground = new SolidColorBrush(Color.FromRgb(194, 48, 48)),
    FontSize = 14,
    FontFamily = new FontFamily("Faberger"),
    Margin = new Thickness(36, 226, 310, 0),
    Visibility = Visibility.Collapsed
};
grid.Children.Add(errorRestaurantName);
grid.Children.Add(errorRadioButton);
grid.Children.Add(groupBox);
grid.Children.Add(button);
grid.Children.Add(suggestionsBox);
button.Click += async (Sender, Event) => {
    if (restaurantName == null || restaurantName == "")
    {
        errorRestaurantName.Visibility = Visibility.Visible;
        TextBoxHelper.SetIsErrorState(searchBox, true);
    }
    else
    {
        restaurantName = restaurantName.Trim();
        errorRestaurantName.Visibility = Visibility.Collapsed;
        TextBoxHelper.SetIsErrorState(searchBox, false);
    }
    if (radioButton1.IsChecked == false && radioButton2.IsChecked == false)
    {
        errorRadioButton.Visibility = Visibility.Visible;
        groupBox.BorderBrush = new SolidColorBrush(Color.FromRgb(194, 48,
48));
    }
};

```

```

        groupBox.BorderThickness = new Thickness(1);
    } else
    {
        errorRadioButton.Visibility = Visibility.Collapsed;
        groupBox.BorderThickness = new Thickness(0);
    }
    if (restaurantName != null && restaurantName != "" &&
        (radioButton1.IsChecked != false || radioButton2.IsChecked != false))
    {
        MessageBox.Show("Вы можете отслеживать статус готовности вашего
        заказа в корзине. Не забывайте обновлять информацию (путем повторного нажатия на вкладку
        \"Корзина\")!");
        var json =
        JsonSerializer.Serialize($"{restaurantName}\n{PaymentMethod}");
        var content = new StringContent(json, Encoding.UTF8,
        "application/json");
        await
        ApiClient.Instance.PutAsync($"Users/{AuthState.Id}/purchase", content);
        Cart cart = new Cart();
        cart.Show();
        Close();
    }
    };
    border.Child = grid;
    mainStackPanel.Children.Add(border);
}

private void RadioButton_Checked(object sender, RoutedEventArgs e)
{
    RadioButton selectedRadioButton = sender as RadioButton;
    PaymentMethod = selectedRadioButton.Content as string;
}

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
    Close();
}
private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
{
    Restaurants rest = new Restaurants();
    rest.Show();
    Close();
}
}
}

```

"Cashier.xaml"

```

<Window x:Class="CourseWorkClient.Cashier"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»" WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="2144"
        WindowState="Maximized"
Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <ScrollViewer
        VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">
        <Border>
            <StackPanel>
                <Border x:Name="borderCashier" Visibility="Hidden" Padding="30"
HorizontalAlignment="Center" CornerRadius="30"
                Background="White" Margin="0,75,0,0">
                    <Border.Effect>
                        <DropShadowEffect BlurRadius="37" Color="DarkGray"
ShadowDepth="10"/>
                    </Border.Effect>
                    <StackPanel x:Name="cashierStackPanel" Margin="-30,-30, -30,-30">

                        </StackPanel>
                    </Border>
                    <TextBlock Visibility="Hidden" Text="Для красивого вида!"
Margin="0,75,0,0"/>
                </StackPanel>
            </Border>
        </ScrollViewer>
    </Window>

```

## "Cashier.xaml.cs"

```

using MaterialDesignThemes.Wpf;
using System;
using System.Collections.Generic;
using System.Collections.ObjectModel;
using System.Globalization;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Text;
using System.Text.Json;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Cashier.xaml
    /// </summary>
    public partial class Cashier : Window
    {

        public Cashier()
        {
            InitializeComponent();
        }
    }
}

```

```

public Cashier(string place)
{
    InitializeComponent();
    borderCashier.Visibility = Visibility.Visible;
    showOrders(place);
}

private async void showOrders(string place)
{
    int usersCount = 0;
    var Users = await
ApiClient.Instance.GetFromJsonAsync<ObservableCollection<User>>("Users") ?? new
ObservableCollection<User>();
    foreach (User user in Users)
    {
        string[] placeParts;
        if (user.Place != null)
        {
            placeParts = user.Place.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        } else
        {
            continue;
        }
        double result = 0;
        if ((user.PurchaseStatus == 1 || user.PurchaseStatus == 2 ||
user.PurchaseStatus == 3) && placeParts[0].Trim() == place)
        {
            string[] prods = user.Cart.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
            foreach (string prod in prods)
            {
                if (prod != null)
                {
                    string[] cartParts = prod.Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                    string cartName = cartParts[0].Trim();
                    if (cartParts.Length == 3) cartName =
Regex.Replace(cartName, @"\"s*(.*?)\"", "");
                    var response = await
ApiClient.Instance.GetAsync($"Products/{cartName}/getcost");
                    if (response.IsSuccessStatusCode)
                    {
                        var cost = await
response.Content.ReadFromJsonAsync<string?>();
                        var image_request = await
ApiClient.Instance.GetFromJsonAsync<string>($"Products/{cartName}/getimage");
                        string amount = cartParts[1];
                        if (cartParts.Length == 3) result +=
Convert.ToInt16(cartParts[1]) * Convert.ToDouble(cartParts[2],
CultureInfo.InvariantCulture);
                        else result += Convert.ToInt16(cartParts[1]) *
Convert.ToDouble(cost, CultureInfo.InvariantCulture);

                        TextBlock textblock = new TextBlock
                        {
                            Text = $"{cartParts[0]} ({cartParts[1]} шт.)",
                            FontSize = 21,
                            FontFamily = new FontFamily("Faberger"),
                            HorizontalAlignment = HorizontalAlignment.Left,
                            VerticalAlignment = VerticalAlignment.Center,
                            TextAlignment = TextAlignment.Left,
                            TextWrapping = TextWrapping.Wrap,
                            Width = 335,
                            Margin = new Thickness(108, 0, 0, 0)
                        };
                        System.Windows.Controls.Image image = new
System.Windows.Controls.Image
                    {

```

```

Uri("${image_request}")),

Source = new BitmapImage(new

Width = 75,
Height = 75,
HorizontalAlignment = HorizontalAlignment.Left,
VerticalAlignment = VerticalAlignment.Center,
Margin = new Thickness(20, 0, 0, 0)
});
Grid newgrid = new Grid
{
Width = 450,
Height = 80,
Margin = new Thickness(0, 10, 0, 0)
};
newgrid.Children.Add(image);
newgrid.Children.Add(textblock);
cashierStackPanel.Children.Add(newgrid);
}
}
if (result != 0)
{
usersCount++;
TextBlock sum = new TextBlock
{
Text = "${result} p.",
FontSize = 32,
FontFamily = new FontFamily("Faberger"),
HorizontalAlignment = HorizontalAlignment.Right,
VerticalAlignment = VerticalAlignment.Top,
TextAlignment = TextAlignment.Right,
};
TextBlock paymentMethod = new TextBlock
{
Text = "${placeParts[1]}",
FontSize = 17,
FontFamily = new FontFamily("Faberger"),
HorizontalAlignment = HorizontalAlignment.Right,
VerticalAlignment = VerticalAlignment.Bottom,
};
Grid payGrid = new Grid
{
Width = 100,
Height = 50,
HorizontalAlignment = HorizontalAlignment.Right,
Margin = new Thickness(0, 0, 20, 0)
};
payGrid.Children.Add(paymentMethod);
payGrid.Children.Add(sum);
cashierStackPanel.Children.Add(payGrid);

if (user.PurchaseStatus == 1)
{
Button confirmButton = new Button
{
Width = 90,
Height = 30,
Content = "Принять",
FontFamily = new FontFamily("Faberger"),
FontSize = 19,
BorderThickness = new Thickness(0),
Cursor = Cursors.Hand,
Foreground = Brushes.White,
Background = new SolidColorBrush(Color.FromRgb(64, 145,
10)),

HorizontalAlignment = HorizontalAlignment.Left
};
confirmButton.Style = (Style)FindResource("GreenButton");
Button declineButton = new Button
{

```

```

        Width = 110,
        Height = 30,
        Content = "Отклонить",
        FontFamily = new FontFamily("Faberge"),
        FontSize = 19,
        Foreground = Brushes.White,
        BorderThickness = new Thickness(0),
        Cursor = Cursors.Hand,
        Margin = new Thickness(95, 0, 0, 0),
        Background = Brushes.Firebrick,
        HorizontalAlignment = HorizontalAlignment.Right
    };
    declineButton.Style = (Style)FindResource("RedButton");
    confirmButton.Click += async (sender, e) =>
    {
        user.PurchaseStatus = 2;
        await
ApiClient.Instance.PutAsync($"Users/{user.id}/purchasestatus", null);
        cashierStackPanel.Children.Clear();
        showOrders(place);
    };
    declineButton.Click += async (sender, e) =>
    {
        user.PurchaseStatus = -1;
        var json = JsonSerializer.Serialize(user);
        var content = new StringContent(json, Encoding.UTF8,
"application/json");

        await ApiClient.Instance.PutAsync("Users", content);
        cashierStackPanel.Children.Clear();
        showOrders(place);
    };
    Grid answerButtons = new Grid()
    {
        Width = 210,
        Height = 30,
        HorizontalAlignment = HorizontalAlignment.Left,
        Margin = new Thickness(20, 0, 0, 20)
    };
    answerButtons.Children.Add(declineButton);
    answerButtons.Children.Add(confirmButton);
    cashierStackPanel.Children.Add(answerButtons);
}
if (user.PurchaseStatus == 2)
{
    Button readyButton = new Button
    {
        Width = 90,
        Height = 30,
        Content = "Готов",
        FontFamily = new FontFamily("Faberge"),
        FontSize = 19,
        Foreground = Brushes.White,
        Cursor = Cursors.Hand,
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Bottom,
        Margin = new Thickness(20, 0, 0, 20),
        Background = new SolidColorBrush(Color.FromRgb(64, 145,
10))

    };
    readyButton.Style = (Style)FindResource("GreenButton");
    readyButton.Click += async (sender, e) =>
    {
        user.PurchaseStatus = 3;
        await
ApiClient.Instance.PutAsync($"Users/{user.id}/purchasestatus", null);
        cashierStackPanel.Children.Clear();
        showOrders(place);
    };
    cashierStackPanel.Children.Add(readyButton);
}
}

```



```

        if (user.PurchaseStatus == 3)
        {
            Button overButton = new Button
            {
                Width = 90,
                Height = 30,
                Content = "Отдан",
                FontFamily = new FontFamily("Faberger"),
                FontSize = 19,
                Foreground = Brushes.White,
                Cursor = Cursors.Hand,
                HorizontalAlignment = HorizontalAlignment.Left,
                VerticalAlignment = VerticalAlignment.Bottom,
                Margin = new Thickness(20, 0, 0, 20),
                Background = new SolidColorBrush(Color.FromRgb(64, 145,
10))
            };
            overButton.Style = (Style)FindResource("GreenButton");
            overButton.Click += async (sender, e) =>
            {
                user.PurchaseStatus = 0;
                DateTime today = DateTime.Today;
                string formattedDate = today.ToString("dd.MM.yyyy");
                user.History += $"{formattedDate}\n";
                foreach (string prod in prods)
                {
                    string[] cartParts = prod.Trim().Split(new[] { " -
" }, StringSplitOptions.RemoveEmptyEntries);
                    string cartName = cartParts[0];
                    if (cartParts.Length == 3) cartName =
Regex.Replace(cartName, @"\"s*(.*?)\"", "");
                    var response = await
ApiClient.Instance.GetAsync($"Products/{cartName}/getimage");
                    if (response.IsSuccessStatusCode)
                    {
                        var image_request = await
response.Content.ReadFromJsonAsync<string>();
                        user.History += $"{cartParts[0]}
({cartParts[1]} мт.)\n{image_request}\n";
                    }
                    user.History += $"{result} py6.\n\n\n";
                    user.Cart = null;
                    var json = JsonSerializer.Serialize(user);
                    var content = new StringContent(json, Encoding.UTF8,
"application/json");

                    await ApiClient.Instance.PutAsync("Users", content);
                    cashierStackPanel.Children.Clear();
                    showOrders(place);
                }
            };
            cashierStackPanel.Children.Add(overButton);
        }

        Rectangle splitLine = new Rectangle
        {
            Height = 1,
            Width = 390,
            HorizontalAlignment = HorizontalAlignment.Center,
            Fill = Brushes.DarkGray,
            VerticalAlignment = VerticalAlignment.Bottom,
        };
        cashierStackPanel.Children.Add(splitLine);
    }
}
if (usersCount == 0)
{
    TextBlock nullCashierText = new TextBlock
    {
        Text = "Пока нет заказов.",
    }
}

```

```

        FontSize = 35,
        FontFamily = new FontFamily("Faberger"),
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Center,
        Width = 400,
        Height = 50,
        TextAlignment = TextAlignment.Center,
        Padding = new Thickness(0, 10, 0, 0)
    };
    cashierStackPanel.Children.Add(nullCashierText);
}
}

protected override void OnClosed(EventArgs e)
{
    base.OnClosed(e);
}
}
}

```

## "History.xaml"

```

<Window x:Class="CourseWorkClient.History"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="2144"
        WindowState="Maximized"
        Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <Window.Background>
        <ImageBrush
            ImageSource="C:\Users\fona\source\repos\CourseWorkClient\Media\maclogo.jpg"
            TileMode="None" Stretch="None" AlignmentY="Top"/>
    </Window.Background>
    <ScrollViewer
        HorizontalScrollBarVisibility="Auto"
        VerticalScrollBarVisibility="Auto">
        <Border>
            <StackPanel x:Name="mainStackPanel">
                <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top"
                    Margin="-28,-36,-30,0">
                    <Border.Effect>
                        <DropShadowEffect
                            BlurRadius="15" Opacity="0.15"
                            ShadowDepth="1"/>
                    </Border.Effect>
                    <StackPanel>
                        <Image
                            Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
                            Width="133" Margin="0,47,1330,0" Height="73"/>
                        <Button Content="Меню" FontSize="35" FontFamily="Faberger"
                            Width="98"
                            Margin="0,-60,1000,0" Height="50" FontWeight="Regular"
                            Foreground="#424242"
                            Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                        <Button Content="Рестораны" FontSize="35" FontFamily="Faberger"
                            Width="192"
                            Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
                            FontWeight="Regular" Foreground="#424242"
                            Click="Button_Restaurants_Click"/>
                        <Button Content="Корзина" FontSize="35" FontFamily="Faberger"
                            Width="148"
                            Margin="0,-60,-20,0" Height="50" Foreground="#424242"
                            Style="{StaticResource DefaultButton}" FontWeight="Regular"
                            Click="Button_Cart_Click"/>
                    </StackPanel>
                </Border>
            </StackPanel>
        </ScrollViewer>
    </Window>

```

```

<Button Content="История заказов" FontSize="35"
FontFamily="Faberge" Width="283"
Margin="0,-60,-680,0" Height="50" FontWeight="Bold"
Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
<Button x:Name="Button_Authorization" Content="Вход"
FontSize="35" FontFamily="Faberge" Width="91"
Margin="0,-60,-1260,0" Height="50"
Foreground="#424242" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
Click="Button_Authorization_Click"/>
<TextBlock x:Name="TextBlock_Authorization" Text="Здравствуйте,
" FontSize="35" FontFamily="Faberge"
Width="221" Margin="0,-75,-1260,0" Height="90" TextAlignment="Right"
Foreground="#424242" HorizontalAlignment="Center" Visibility="Hidden"/>
</StackPanel>
</Border>
<Border x:Name="borderHistory" Visibility="Hidden" Padding="30"
HorizontalAlignment="Center" CornerRadius="30"
Background="White" Margin="0,75,0,0">
<Border.Effect>
<DropShadowEffect BlurRadius="37" Color="DarkGray"
ShadowDepth="10"/>
</Border.Effect>
<StackPanel x:Name="historyStackPanel" Margin="-30,-30,-30,-30">

</StackPanel>
</Border>
<TextBlock Visibility="Hidden" Text="Для красивого вида!"
Margin="0,75,0,0"/>
</StackPanel>
</Border>
</ScrollView>
</Window>

```

## "History.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.IO;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Markup;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для History.xaml
    /// </summary>
    public partial class History : Window
    {
        public History()
        {
            InitializeComponent();
            if (AuthState.IsAuthenticated)
            {
                Button_Authorization.Visibility = Visibility.Hidden;
                TextBlock_Authorization.Visibility = Visibility.Visible;
                borderHistory.Visibility = Visibility.Visible;
            }
        }
    }
}

```

```

        showProducts();
    }
    if (!AuthState.isAuthenticated)
    {
        TextBlock error = new TextBlock
        {
            Text = "Вы не вошли в аккаунт!",
            FontSize = 35,
            FontFamily = new FontFamily("Faberger"),
            HorizontalAlignment = HorizontalAlignment.Center,
            Background = new SolidColorBrush(Color.FromRgb(255, 255, 255)),
            Width = 400,
            Height = 60,
            TextAlignment = TextAlignment.Center,
            Padding = new Thickness(0, 10, 0, 0)
        };
        error.Margin = new Thickness(0, 100, 0, 0);
        mainStackPanel.Children.Add(error);
    }

    if (AuthState.isAuthenticated) Loaded += async (s, e) => await
MakeNiceName();
    }
    private async Task MakeNiceName()
    {
        var username = await
ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getname");
        if (username.Length > 13) TextBlock_Authorization.Text =
$"Здравствуйте,\n{username.Substring(0, 11)}...";
        else TextBlock_Authorization.Text = $"Здравствуйте,\n{username}";
    }

    private async void showProducts()
    {
        var history = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Users/{AuthState.Id}/gethistory");
        if (history != "" && history != null)
        {
            string[] blocks = history.Split(new[] { "\n\n\n" },
StringSplitOptions.RemoveEmptyEntries);
            for (int j = 0; j < blocks.Length; j++)
            {
                if (blocks[j] != null)
                {
                    string[] lines = blocks[j].Split(new[] { '\n' },
StringSplitOptions.None);
                    string data = lines[0].Trim();
                    string total = lines[lines.Length - 1].Trim();

                    TextBlock datablock = new TextBlock
                    {
                        Text = $"{data}",
                        FontSize = 26,
                        FontFamily = new FontFamily("Faberger"),
                        HorizontalAlignment = HorizontalAlignment.Left,
                        VerticalAlignment = VerticalAlignment.Top,
                        FontWeight = FontWeights.Bold,
                        Width = 250,
                        Margin = new Thickness(20, 20, 0, 0)
                    };
                    historyStackPanel.Children.Add(datablock);
                    TextBlock sum = new TextBlock
                    {
                        Text = $"{total}",
                        FontSize = 32,
                        FontFamily = new FontFamily("Faberger"),
                        HorizontalAlignment = HorizontalAlignment.Right,
                        VerticalAlignment = VerticalAlignment.Bottom,
                        TextAlignment = TextAlignment.Right,

```

```

        Margin = new Thickness(0, 0, 20, 20)
    };

    for (int i = 1; i < lines.Length - 2; i += 2)
    {
        string productLine = lines[i].Trim();
        string imagePath = lines[i + 1].Trim();

        try
        {
            if (!File.Exists(imagePath))
            {
                throw new FileNotFoundException();
            }
        }
        catch (FileNotFoundException)
        {
            imagePath =
"C:/Users/fona/source/repos/coursework3/Media/No_image_available.svg.png";
        }

        System.Windows.Controls.Image image = new
System.Windows.Controls.Image
        {
            Source = new BitmapImage(new Uri($"{imagePath}")),
            Width = 75,
            Height = 75,
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Center,
            Margin = new Thickness(33, 0, 0, 0)
        };
        TextBlock textblock = new TextBlock
        {
            Text = $"{productLine}",
            FontSize = 21,
            FontFamily = new FontFamily("Faberge"),
            HorizontalAlignment = HorizontalAlignment.Left,
            VerticalAlignment = VerticalAlignment.Center,
            TextAlignment = TextAlignment.Left,
            TextWrapping = TextWrapping.Wrap,
            Width = 250,
            Margin = new Thickness(121, 0, 0, 0)
        };

        Grid newgrid = new Grid
        {
            Width = 450,
            Height = 80
        };
        newgrid.Children.Add(image);
        newgrid.Children.Add(textblock);
        historyStackPanel.Children.Add(newgrid);
    }

    historyStackPanel.Children.Add(sum);

    if (j < blocks.Length - 1)
    {
        Rectangle splitLine = new Rectangle
        {
            Height = 2,
            Fill = new SolidColorBrush(Color.FromRgb(130, 130,
130)),
            VerticalAlignment = VerticalAlignment.Bottom,
        };
        historyStackPanel.Children.Add(splitLine);
    }
    }
}

```

```

else
{
    TextBlock nullHistoryText = new TextBlock
    {
        Text = "История заказов пуста.",
        FontSize = 35,
        FontFamily = new FontFamily("Faberge"),
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Center,
        Width = 500,
        Height = 50,
        TextAlignment = TextAlignment.Center,
        Padding = new Thickness(0, 10, 0, 0)
    };
    historyStackPanel.Children.Add(nullHistoryText);
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
    Close();
}

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
{
    Restaurants rest = new Restaurants();
    rest.Show();
    Close();
}
}
}

```

## "Item.xaml"

```

<Window x:Class="CourseWorkClient.Item"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="800"
        WindowState="Maximized"
        Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">

```

```

        <ScrollViewer
HorizontalScrollBarVisibility="Auto">
            <Grid Width="1505" Height="784">
                <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top" Margin="-
28,-36,-30,0">
                    <Border.Effect>
                        <DropShadowEffect BlurRadius="15" Opacity="0.15" ShadowDepth="1"/>
                    </Border.Effect>
                    <StackPanel>
                        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden
Width="133" Margin="0,47,1330,0" Height="73"/>
                        <Button Content="Меню" FontSize="35" FontFamily="Faberger"
Width="98"
Margin="0,-60,1000,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                        <Button Content="Рестораны" FontSize="35" FontFamily="Faberger"
Width="192"
Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
FontWeight="Regular" Foreground="#424242"
Click="Button_Restaurants_Click"/>
                        <Button Content="Корзина" FontSize="35" FontFamily="Faberger"
Width="148"
Margin="0,-60,-20,0" Height="50" Foreground="#424242"
Style="{StaticResource DefaultButton}" FontWeight="Regular"
Click="Button_Cart_Click"/>
                        <Button Content="История заказов" FontSize="35"
FontFamily="Faberger" Width="283"
Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
                        <Button x:Name="Button_Authorization" Content="Вход" FontSize="35"
FontFamily="Faberger" Width="91"
Margin="0,-60,-1260,0" Height="50"
Foreground="#424242" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
Click="Button_Authorization_Click"/>
                        <TextBlock x:Name="TextBlock_Authorization" Text="Здравствуй, "
FontSize="35" FontFamily="Faberger"
Width="221" Margin="0,-75,-1260,0" Height="90" TextAlignment="Right"
Foreground="#424242" HorizontalAlignment="Center" Visibility="Hidden"/>
                    </StackPanel>
                </Border>
                <Grid x:Name="ItemGrid" Margin="85,191,781,138" >
                    <Grid.RowDefinitions>
                        <RowDefinition Height="52*" />
                        <RowDefinition Height="157*" />
                    </Grid.RowDefinitions>
                    <Grid.ColumnDefinitions>
                        <ColumnDefinition Width="169*" />
                    </Grid.ColumnDefinitions>
                    <Button x:Name="AddToCartButton" Style="{StaticResource YellowButton}"
Background="#f7ce36"
Content="В корзину" FontSize="40" FontFamily="Faberger"
Width="230" Height="70" Margin="0,314,409,-70" Grid.Row="1"
Click="AddToCartButton_Click"
Cursor="Hand" Visibility="Hidden"/>
                    <TextBlock x:Name="ActivePurchaseStatusText" Text="У вас есть активный
заказ." Visibility="Collapsed" FontFamily="Faberger"
FontSize="32" Foreground="#696969" Margin="0,314,0,-70"
Grid.Row="1"/>
                    <TextBlock x:Name="textBlockDesc" FontFamily="Faberger" FontSize="27"
TextWrapping="Wrap" HorizontalAlignment="Left" VerticalAlignment="Center"
Grid.Row="1"/>
                </Grid>
                <StackPanel Orientation="Horizontal" HorizontalAlignment="Left"
VerticalAlignment="Top" Margin="80,557,0,0">
                    <RadioButton x:Name="RadioButton1" Visibility="Collapsed"
GroupName="Quantity" Style="{StaticResource AmountChoice}"
Checked="RadioButton1_Checked"/>

```

```

        <RadioButton x:Name="RadioButton2" Visibility="Collapsed"
GroupName="Quantity" Style="{StaticResource AmountChoice}"
Checked="RadioButton2_Checked"/>
        <RadioButton x:Name="RadioButton3" Visibility="Collapsed"
GroupName="Quantity" Style="{StaticResource AmountChoice}"
Checked="RadioButton3_Checked"/>
        <RadioButton x:Name="RadioButton4" Visibility="Collapsed"
GroupName="Quantity" Style="{StaticResource AmountChoice}"
Checked="RadioButton4_Checked"/>
    </StackPanel>
    <Grid x:Name="ItemCounter" Margin="359,645,1000,95" Visibility="Hidden">
        <Grid.ColumnDefinitions>
            <ColumnDefinition />
            <ColumnDefinition />
            <ColumnDefinition />
        </Grid.ColumnDefinitions>
        <Border BorderBrush="#c2c2c2" BorderThickness="1" Grid.ColumnSpan="3"
Margin="46,0,44,0">
            <TextBlock x:Name="NumberTextBlock" Grid.Column="1" Text="1"
FontSize="35" VerticalAlignment="Center" HorizontalAlignment="Center"/>
        </Border>
        <Button Style="{StaticResource GrayButton}" Background="#c2c2c2"
Content="-" FontSize="45" Cursor="Hand" Grid.Column="0"
FontFamily="Faberge"
Click="DecrementButton_Click"/>
        <Button Style="{StaticResource GrayButton}" Background="#c2c2c2"
Content="+" FontSize="45" Cursor="Hand" Grid.Column="2" FontFamily="Faberge"
Click="IncrementButton_Click"/>
    </Grid>
    <TextBlock x:Name="Amount" Text="шт." FontSize="30" FontFamily="Faberge"
VerticalAlignment="Top" Margin="518,649,940,0"
Visibility="Hidden"/>
    <Grid x:Name="ItemImageGrid" Margin="724,211,86,121" />
</Grid>
</ScrollViewer>
</Window>

```

## "Item.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Security.Policy;
using System.Text;
using System.Text.Json;
using System.Text.RegularExpressions;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Item.xaml
    /// </summary>
    public partial class Item : Window
    {
        public Item()
        {
            InitializeComponent();
        }
    }
}

```



```

private Product prodForCart = new Product();
private int switchcase = 0;
public Item(Product product)
{
    prodForCart = product;
    InitializeComponent();
    if (AuthState.isAuthenticated)
    {
        Button_Authorization.Visibility = Visibility.Hidden;
        TextBlock_Authorization.Visibility = Visibility.Visible;
    }
    Image image = new Image
    {
        Source = new BitmapImage(new Uri($"{product.Image}")),
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Center
    };

    TextBlock textBlockName = new TextBlock
    {
        Text = product.Name,
        FontFamily = new FontFamily("Mustica Pro SemBd"),
        FontSize = 40,
        TextWrapping = TextWrapping.Wrap,
        HorizontalAlignment = HorizontalAlignment.Left,
        VerticalAlignment = VerticalAlignment.Bottom
    };

    textBlockDesc.Text = $"{product.Description}\n\n";

    if (product.Cost.Contains("or"))
    {
        RadioButton1.IsChecked = true;
        string[] amounts = product.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        for (int i = 1; i < amounts.Length; i++)
        {
            string[] parts = amounts[i].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
            switch (i)
            {
                case 1:
                    RadioButton1.Content = parts[0];
                    RadioButton1.Visibility = Visibility.Visible;
                    break;
                case 2:
                    RadioButton2.Content = parts[0];
                    RadioButton2.Visibility = Visibility.Visible;
                    break;
                case 3:
                    RadioButton3.Content = parts[0];
                    RadioButton3.Visibility = Visibility.Visible;
                    break;
                case 4:
                    RadioButton4.Content = parts[0];
                    RadioButton4.Visibility = Visibility.Visible;
                    break;
            }
        }
    }
    else textBlockDesc.Text += $"{product.Cost} py6.";

    ItemImageGrid.Children.Add(image);

    ItemGrid.Children.Add(textBlockName);
    Grid.SetRow(textBlockName, 0);

    if (AuthState.isAuthenticated) Loaded += async (s, e) => await MakeNiceUI();
}

private async Task MakeNiceUI()

```

```

        {
            var username = await
ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getname");
            if (username.Length > 13) TextBlock_Authorization.Text =
$"Здравствуйте, \n{username.Substring(0, 11)}...";
            else TextBlock_Authorization.Text = $"Здравствуйте, \n{username}";

            var purchasestatus = await
ApiClient.Instance.GetFromJsonAsync<int>($"Users/{AuthState.Id}/getpurchasestatus");
            if (purchasestatus == 1 || purchasestatus == 2 || purchasestatus == 3)
            {
                ActivePurchaseStatusText.Visibility = Visibility.Visible;
            }
            else
            {
                AddToCartButton.Visibility = Visibility.Visible;
                ItemCounter.Visibility = Visibility.Visible;
                Amount.Visibility = Visibility.Visible;
            }
        }

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
    Close();
}

private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
{
    Restaurants rest = new Restaurants();
    rest.Show();
    Close();
}

private int count = 1;
private async void AddToCartButton_Click(object sender, RoutedEventArgs e)
{
    switch (switchcase)
    {
        case 0:
            break;
        case 1:
            prodForCart.Name = Regex.Replace(prodForCart.Name, @"\"s*\"(.*)\"",
""");
            prodForCart.Name += $" ({RadioButton1.Content})";
            break;
        case 2:

```

```

        prodForCart.Name = Regex.Replace(prodForCart.Name, @"\s*(.*?)",
""");
        prodForCart.Name += $" ({RadioButton2.Content})";
        break;
    case 3:
        prodForCart.Name = Regex.Replace(prodForCart.Name, @"\s*(.*?)",
""");
        prodForCart.Name += $" ({RadioButton3.Content})";
        break;
    case 4:
        prodForCart.Name = Regex.Replace(prodForCart.Name, @"\s*(.*?)",
""");
        prodForCart.Name += $" ({RadioButton4.Content})";
        break;
    }

    var cart = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Users/{AuthState.Id}/getcart");
    if (cart != null && cart.Contains(prodForCart.Name) && cart != "")
    {
        string[] prods = cart.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] parts = prods.FirstOrDefault(n =>
n.Contains(prodForCart.Name)).Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
        count += Convert.ToInt16(parts[1]);
        int indexToRemove = Array.IndexOf(prods, prods.FirstOrDefault(n =>
n.Contains(prodForCart.Name)));
        string[] newProds = prods.Where((val, idx) => idx !=
indexToRemove).ToArray();
        cart = null;
        foreach (string newprod in newProds) {
            cart += $"{newprod}\n";
        }
    }
    if (prodForCart.Cost.Contains("OT"))
    {
        string[] amounts = prodForCart.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] diffparts;
        switch (switchcase)
        {
            case 1:
                diffparts = amounts[1].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                cart += $"{prodForCart.Name} - {count} - {diffparts[1]}\n";
                break;
            case 2:
                diffparts = amounts[2].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                cart += $"{prodForCart.Name} - {count} - {diffparts[1]}\n";
                break;
            case 3:
                diffparts = amounts[3].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                cart += $"{prodForCart.Name} - {count} - {diffparts[1]}\n";
                break;
            case 4:
                diffparts = amounts[4].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
                cart += $"{prodForCart.Name} - {count} - {diffparts[1]}\n";
                break;
        }
    }
    else cart += $"{prodForCart.Name} - {count}\n";
    count = 1;
    NumberTextBlock.Text = count.ToString();
    var json = JsonSerializer.Serialize(cart);
    var content = new StringContent(json, Encoding.UTF8, "application/json");
    await ApiClient.Instance.PutAsync($"Users/{AuthState.Id}/cart/add",
content);

```

```

        MessageBox.Show("Товар добавлен в корзину.");
    }

    private void DecrementButton_Click(object sender, RoutedEventArgs e)
    {
        if (count != 1) count--;
        NumberTextBlock.Text = count.ToString();
    }

    private void IncrementButton_Click(object sender, RoutedEventArgs e)
    {
        if (count != 99) count++;
        NumberTextBlock.Text = count.ToString();
    }

    private void RadioButton1_Checked(object sender, RoutedEventArgs e)
    {
        string[] amounts = prodForCart.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] parts = amounts[1].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
        textBlockDesc.Text = $"{prodForCart.Description}\n\n{parts[1]} руб.";
        switchcase = 1;
    }

    private void RadioButton2_Checked(object sender, RoutedEventArgs e)
    {
        string[] amounts = prodForCart.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] parts = amounts[2].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
        textBlockDesc.Text = $"{prodForCart.Description}\n\n{parts[1]} руб.";
        switchcase = 2;
    }

    private void RadioButton3_Checked(object sender, RoutedEventArgs e)
    {
        string[] amounts = prodForCart.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] parts = amounts[3].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
        textBlockDesc.Text = $"{prodForCart.Description}\n\n{parts[1]} руб.";
        switchcase = 3;
    }

    private void RadioButton4_Checked(object sender, RoutedEventArgs e)
    {
        string[] amounts = prodForCart.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries);
        string[] parts = amounts[4].Trim().Split(new[] { " - " },
StringSplitOptions.RemoveEmptyEntries);
        textBlockDesc.Text = $"{prodForCart.Description}\n\n{parts[1]} руб.";
        switchcase = 4;
    }
}
}

```

## “Menu.xaml”

```

<Window x:Class="CourseWorkClient.Menu"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"

```

```

        Width="1505" Height="1800"
        WindowState="Maximized"
Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <ScrollView
        VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">
        <Grid Width="1505" Height="1503">
            <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top" Margin="-
28,-36,-30,0">
                <Border.Effect>
                    <DropShadowEffect BlurRadius="15" Opacity="0.15" ShadowDepth="1"/>
                </Border.Effect>
                <StackPanel>
                    <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
Width="133" Margin="0,47,1330,0" Height="73"/>
                    <Button Content="Меню" FontSize="35" FontFamily="Faberger"
Width="98"
                        Margin="0,-60,1000,0" Height="50" FontWeight="Bold"
                        Style="{StaticResource DefaultButton}"
Click="Button_Menu_Click"/>
                    <Button Content="Рестораны" FontSize="35" FontFamily="Faberger"
Width="192"
Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
FontWeight="Regular" Foreground="#424242"
Click="Button_Restaurants_Click"/>
                    <Button Content="Корзина" FontSize="35" FontFamily="Faberger"
Width="148"
Margin="0,-60,-20,0" Height="50" Foreground="#424242"
Style="{StaticResource DefaultButton}" FontWeight="Regular"
Click="Button_Cart_Click"/>
                    <Button Content="История заказов" FontSize="35"
FontFamily="Faberger" Width="283"
Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
                    <Button x:Name="Button_Authorization" Content="Вход" FontSize="35"
FontFamily="Faberger" Width="91"
                        Margin="0,-60,-1260,0" Height="50"
                        Foreground="#424242" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
Click="Button_Authorization_Click"/>
                    <TextBlock x:Name="TextBlock_Authorization" Text="Здравствуйете, "
FontSize="35" FontFamily="Faberger"
Width="221" Margin="0,-75,-1260,0" Height="90"
TextAlignment="Right"
                        Foreground="#424242" HorizontalAlignment="Center"
Visibility="Hidden"/>
                </StackPanel>
            </Border>
            <Border Background="White" BorderBrush="Gray" BorderThickness="1.5"
Margin="39,138,1074,369"
                CornerRadius="2">
                <Border Height="110" Margin="0,-2,0,756">
                    <StackPanel Margin="0,-63,0,-820">
                        <Border Height="110">
                            <Grid>
                                <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\avzd0x7ig4kymzpgxjpsh6wf90ut
ogjl.png" HorizontalAlignment="Left" Margin="12,12,0,13" Width="137"/>
                                <TextBlock Text="Бургеры и Роллы" FontSize="25"
FontFamily="Faberger" Margin="175,41,26,37"/>
                                <Rectangle x:Name="rect1" Fill="Firebrick" Width="15"
HorizontalAlignment="Right"/>
                                <Button Height="110" Background="Transparent"
BorderThickness="0" Foreground="Transparent"
                                    Style="{StaticResource DefaultButton}"
Click="ButtonBurger_Click"/>
                            </Grid>
                        </Border>
                    </StackPanel>
                </Border>
            </Grid>

```

```

        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\chicken.png"
HorizontalAlignment="Left" Margin="10,-16,0,-9" Width="144"/>
        <TextBlock          Text="Курочка"          FontSize="25"
FontFamily="Faberge" Margin="224,41,73,37"/>
        <Rectangle x:Name="rect2" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
        <Button          Height="110"          Background="Transparent"
BorderThickness="0" Foreground="Transparent"
                        Style="{StaticResource          DefaultButton}"
Click="ButtonChicken_Click"/>
    </Grid>
</Border>
<Border Height="110">
    <Grid>
        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\fries.png"
HorizontalAlignment="Left" Margin="28,5,0,5" Width="100"/>
        <TextBlock          Text="Картошка"          FontSize="25"
FontFamily="Faberge" Margin="209,41,55,37"/>
        <Rectangle x:Name="rect3" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
        <Button          Height="110"          Background="Transparent"
BorderThickness="0" Foreground="Transparent"
                        Style="{StaticResource          DefaultButton}"
Click="ButtonFries_Click"/>
    </Grid>
</Border>
<Border Height="110">
    <Grid>
        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\rodnoy-vkus.png"
HorizontalAlignment="Left" Margin="32,5,0,5" Width="100"/>
        <TextBlock          Text="Родной        вкус"          FontSize="25"
FontFamily="Faberge" Margin="197,41,55,37"/>
        <Rectangle x:Name="rect4" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
        <Button          Height="110"          Background="Transparent"
BorderThickness="0" Foreground="Transparent"
                        Style="{StaticResource          DefaultButton}"
Click="ButtonRodnoyVkus_Click"/>
    </Grid>
</Border>
<Border Height="110">
    <Grid>
        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\breakfast.png"
HorizontalAlignment="Left" Margin="17,8,0,7" Width="178"/>
        <TextBlock          Text="Завтрак    до    10:00"          FontSize="25"
FontFamily="Faberge" Margin="178,41,20,37"/>
        <Rectangle x:Name="rect5" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
        <Button          Height="110"          Background="Transparent"
BorderThickness="0" Foreground="Transparent"
                        Style="{StaticResource          DefaultButton}"
Click="ButtonBreakfast_Click"/>
    </Grid>
</Border>
<Border Height="110">
    <Grid>
        <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\drinks.png"
HorizontalAlignment="Left" Margin="28,5,0,5" Width="100"/>
        <TextBlock          Text="Напитки"          FontSize="25"
FontFamily="Faberge" Margin="214,41,77,37"/>
        <Rectangle x:Name="rect6" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
        <Button          Height="110"          Background="Transparent"
BorderThickness="0" Foreground="Transparent"

```

```

Click="ButtonDrinks_Click"/>
        </Grid>
    </Border>
    <Border Height="110">
        <Grid>
            <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\desert.png"
HorizontalAlignment="Left" Margin="23,0,0,5" Width="161"/>
            <TextBlock Text="Десерты" FontSize="25"
FontFamily="Faberge" Margin="214,41,74,37"/>
            <Rectangle x:Name="rect7" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
            <Button Height="110" Background="Transparent"
BorderThickness="0" Foreground="Transparent"
Style="{StaticResource DefaultButton}"
Click="ButtonDeserts_Click"/>
        </Grid>
    </Border>
    <Border Height="110">
        <Grid>
            <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\kidsmeal.png"
HorizontalAlignment="Left" Margin="36,10,0,10" Width="81"/>
            <TextBlock Text="Кидс Мил" FontSize="25"
FontFamily="Faberge" Margin="209,41,70,37"/>
            <Rectangle x:Name="rect8" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
            <Button Height="110" Background="Transparent"
BorderThickness="0" Foreground="Transparent"
Style="{StaticResource DefaultButton}"
Click="ButtonKidsMeal_Click"/>
        </Grid>
    </Border>
    <Border Height="110">
        <Grid>
            <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\sets.png"
HorizontalAlignment="Left" Margin="3,-4,0,0" Width="162"/>
            <TextBlock Text="Сеты" FontSize="25"
FontFamily="Faberge" Margin="230,41,101,37"/>
            <Rectangle x:Name="rect9" Fill="Firebrick" Width="15"
HorizontalAlignment="Right" Visibility="Hidden"/>
            <Button Height="110" Background="Transparent"
BorderThickness="0" Foreground="Transparent"
Style="{StaticResource DefaultButton}"
Click="ButtonSets_Click"/>
        </Grid>
    </Border>
</StackPanel>
</Border>
</Border>
<Grid x:Name="menuList" Margin="500,138,37,57">
    <Grid.RowDefinitions>
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
        <RowDefinition />
    </Grid.RowDefinitions>
    <Grid.ColumnDefinitions>
        <ColumnDefinition />
        <ColumnDefinition />
        <ColumnDefinition />
    </Grid.ColumnDefinitions>
    </Grid>
    <Button x:Name="Sort1Button" Content="Сортировать по цене ↓" Width="250"
Height="30" Cursor="Hand"
Style="{StaticResource YellowButton}" Background="#f7ce36"
FontFamily="Faberge"

```

```

        FontSize="23"
        Margin="520,130,735,1343"
Click="Button_Sort1_Click"/>
        <Button x:Name="Sort2Button" Content="Сортировать по цене ↑" Width="250"
Height="30" Cursor="Hand"
        Style="{StaticResource YellowButton}" Background="#f7ce36" FontFamily="Faberge"
        FontSize="23"
        Margin="520,130,735,1343"
        Click="Button_Sort2_Click"
Visibility="Collapsed"/>
    </Grid>
</ScrollView>
</Window>

```

## "Menu.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.Globalization;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Json;
using System.Security.Policy;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Controls.Primitives;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Animation;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Menu.xaml
    /// </summary>
    public partial class Menu : Window
    {
        private string category = "Бургеры и Роллы";
        public Menu()
        {
            InitializeComponent();
            if (AuthState.IsAuthenticated)
            {
                Button_Authorization.Visibility = Visibility.Hidden;
                TextBlock_Authorization.Visibility = Visibility.Visible;
            }
            MakeCategoryName("Бургеры и Роллы");
            PerformProduct("burgers", 0);

            if (AuthState.IsAuthenticated) Loaded += async (s, e) => await
            MakeNiceName();
        }

        private async Task MakeNiceName()
        {
            var username = await
            ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getname");
            if (username.Length > 13) TextBlock_Authorization.Text =
            $"Здравствуйте, \n{username.Substring(0, 11)}...";
            else TextBlock_Authorization.Text = $"Здравствуйте, \n{username}";
        }

        private void MakeCategoryName(string name)
        {
            TextBlock categoryName = new TextBlock
            {
                Text = name,
            }
        }
    }
}

```



```

        FontSize = 40,
        FontFamily = new FontFamily("Faberger"),
        FontWeight = FontWeights.Bold,
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Top,
        TextAlignment = TextAlignment.Center
    };
    categoryName.Margin = new Thickness(0, -20, 0, 0);

    menuList.Children.Add(categoryName);
    Grid.SetColumn(categoryName, 1);
}
private void ClearGrid()
{
    int row = 0;
    int column = 0;
    while (row < menuList.RowDefinitions.Count && column <
menuList.ColumnDefinitions.Count)
    {
        if (menuList.Children.Count != 0)
        {
            menuList.Children.Clear();
            column++;
            if (column >= menuList.ColumnDefinitions.Count)
            {
                column = 0;
                row++;
            }
        }
        else break;
    }
}
private async void PerformProduct(string prodType, int numSort)
{
    var prod = await
ApiClient.Instance.GetFromJsonAsync<List<Product>>($"Products/get/{prodType}");
    if (numSort != 0)
    {
        foreach (var one in prod)
        {
            if (one.Cost.Contains("ot"))
            {
                one.Cost = one.Cost.Split(new[] { '\n' },
StringSplitOptions.RemoveEmptyEntries)[0].Split(' ')[1];
            }
        }
    }
    if (numSort == 1)
    {
        prod.Sort((x, y) => Convert.ToDouble(x.Cost,
CultureInfo.InvariantCulture).CompareTo(Convert.ToDouble(y.Cost,
CultureInfo.InvariantCulture)));
        foreach (var one in prod)
        {
            one.Cost = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Products/{one.Name}/getcost");
        }
    } else if (numSort == 2)
    {
        prod.Sort((x, y) => Convert.ToDouble(y.Cost,
CultureInfo.InvariantCulture).CompareTo(Convert.ToDouble(x.Cost,
CultureInfo.InvariantCulture)));
        foreach (var one in prod)
        {
            one.Cost = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Products/{one.Name}/getcost");
        }
    }
    int row = 0;
    int column = 0;

```

```

foreach (var one in prod)
{
    Button button = new Button
    {
        Style = (Style)FindResource("AnimButton")
    };
    button.Background = new ImageBrush
    {
        ImageSource = new BitmapImage(new Uri($"{one.Image}")),
        Stretch = Stretch.Uniform,
        Viewport = new Rect(0.22, 0.14, 0.55, 0.55)
    };
    TextBlock textblock = new TextBlock
    {
        Text = $"{one.Name}\n{one.Cost.Split('\n')[0].Trim()} pyб.",
        FontSize = 25,
        FontFamily = new FontFamily("Faberge"),
        HorizontalAlignment = HorizontalAlignment.Center,
        VerticalAlignment = VerticalAlignment.Bottom,
        TextAlignment = TextAlignment.Center,
        TextWrapping = TextWrapping.Wrap
    };
    button.Content = textblock;
    button.Click += (sender, e) =>
    {
        Item item = new Item(one);
        item.Show();
        Close();
    };

    menuList.Children.Add(button);
    Grid.SetRow(button, row);
    Grid.SetColumn(button, column);

    column++;
    if (column >= menuList.ColumnDefinitions.Count)
    {
        column = 0;
        row++;
    }
}

private void RectChange(int num)
{
    rect1.Visibility = Visibility.Hidden;
    rect2.Visibility = Visibility.Hidden;
    rect3.Visibility = Visibility.Hidden;
    rect4.Visibility = Visibility.Hidden;
    rect5.Visibility = Visibility.Hidden;
    rect6.Visibility = Visibility.Hidden;
    rect7.Visibility = Visibility.Hidden;
    rect8.Visibility = Visibility.Hidden;
    rect9.Visibility = Visibility.Hidden;
    switch(num)
    {
        case 1:
            rect1.Visibility = Visibility.Visible;
            break;
        case 2:
            rect2.Visibility = Visibility.Visible;
            break;
        case 3:
            rect3.Visibility = Visibility.Visible;
            break;
        case 4:
            rect4.Visibility = Visibility.Visible;
            break;
        case 5:
            rect5.Visibility = Visibility.Visible;

```

```

        break;
    case 6:
        rect6.Visibility = Visibility.Visible;
        break;
    case 7:
        rect7.Visibility = Visibility.Visible;
        break;
    case 8:
        rect8.Visibility = Visibility.Visible;
        break;
    case 9:
        rect9.Visibility = Visibility.Visible;
        break;
    }
}

private void ButtonBurger_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Бургеры и Роллы";
    MakeCategoryName("Бургеры и Роллы");
    PerformProduct("burgers", 0);
    RectChange(1);
}

private void ButtonChicken_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Курочка";
    MakeCategoryName("Курочка");
    PerformProduct("chicken", 0);
    RectChange(2);
}

private void ButtonFries_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Картошка";
    MakeCategoryName("Картошка");
    PerformProduct("fries", 0);
    RectChange(3);
}

private void ButtonRodnoyVkus_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Родной вкус";
    MakeCategoryName("Родной вкус");
    PerformProduct("rodnoyvkus", 0);
    RectChange(4);
}

private void ButtonBreakfast_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Завтрак до 10:00";
    MakeCategoryName("Завтрак до 10:00");
    PerformProduct("breakfast", 0);
    RectChange(5);
}

private void ButtonDrinks_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Напитки";
    MakeCategoryName("Напитки");
    PerformProduct("drinks", 0);
    RectChange(6);
}

```

```

private void ButtonDeserts_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Десерты";
    MakeCategoryName("Десерты");
    PerformProduct("deserts", 0);
    RectChange(7);
}

private void ButtonKidsMeal_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Кидс Мил";
    MakeCategoryName("Кидс Мил");
    PerformProduct("kidsmeal", 0);
    RectChange(8);
}

private void ButtonSets_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    category = "Сеты";
    MakeCategoryName("Сеты");
    PerformProduct("sets", 0);
    RectChange(9);
}

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
    Close();
}

private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
{
    Restaurants rest = new Restaurants();
    rest.Show();
    Close();
}

private void Button_Sort1_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    switch (category)
    {
        case "Бургеры и Роллы":
            category = "Бургеры и Роллы";
            MakeCategoryName("Бургеры и Роллы");
            PerformProduct("burgers", 1);
    }
}

```

```

        break;
    case "Курочка":
        category = "Курочка";
        MakeCategoryName("Курочка");
        PerformProduct("chicken", 1);
        break;
    case "Картошка":
        category = "Картошка";
        MakeCategoryName("Картошка");
        PerformProduct("fries", 1);
        break;
    case "Родной вкус":
        category = "Родной вкус";
        MakeCategoryName("Родной вкус");
        PerformProduct("rodnoyvкус", 1);
        break;
    case "Завтрак до 10:00":
        category = "Завтрак до 10:00";
        MakeCategoryName("Завтрак до 10:00");
        PerformProduct("breakfast", 1);
        break;
    case "Напитки":
        category = "Напитки";
        MakeCategoryName("Напитки");
        PerformProduct("drinks", 1);
        break;
    case "Десерты":
        category = "Десерты";
        MakeCategoryName("Десерты");
        PerformProduct("deserts", 1);
        break;
    case "Кидс Мил":
        category = "Кидс Мил";
        MakeCategoryName("Кидс Мил");
        PerformProduct("kidsmeal", 1);
        break;
    case "Сеты":
        category = "Сеты";
        MakeCategoryName("Сеты");
        PerformProduct("sets", 1);
        break;
    }
    Sort1Button.Visibility = Visibility.Collapsed;
    Sort2Button.Visibility = Visibility.Visible;
}

private void Button_Sort2_Click(object sender, RoutedEventArgs e)
{
    ClearGrid();
    switch (category)
    {
        case "Бургеры и Роллы":
            category = "Бургеры и Роллы";
            MakeCategoryName("Бургеры и Роллы");
            PerformProduct("burgers", 2);
            break;
        case "Курочка":
            category = "Курочка";
            MakeCategoryName("Курочка");
            PerformProduct("chicken", 2);
            break;
        case "Картошка":
            category = "Картошка";
            MakeCategoryName("Картошка");
            PerformProduct("fries", 2);
            break;
        case "Родной вкус":
            category = "Родной вкус";
            MakeCategoryName("Родной вкус");
            PerformProduct("rodnoyvкус", 2);
    }
}

```

```

        break;
    case "Завтрак до 10:00":
        category = "Завтрак до 10:00";
        MakeCategoryName("Завтрак до 10:00");
        PerformProduct("breakfast", 2);
        break;
    case "Напитки":
        category = "Напитки";
        MakeCategoryName("Напитки");
        PerformProduct("drinks", 2);
        break;
    case "Десерты":
        category = "Десерты";
        MakeCategoryName("Десерты");
        PerformProduct("deserts", 2);
        break;
    case "Кидс Мил":
        category = "Кидс Мил";
        MakeCategoryName("Кидс Мил");
        PerformProduct("kidsmeal", 2);
        break;
    case "Сеты":
        category = "Сеты";
        MakeCategoryName("Сеты");
        PerformProduct("sets", 2);
        break;
    }
    Sort1Button.Visibility = Visibility.Visible;
    Sort2Button.Visibility = Visibility.Collapsed;
}
}
}

```

## "Restaurants.xaml"

```

<Window x:Class="CourseWorkClient.Restaurants"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        xmlns:wv2="clr-namespace:Microsoft.Web.WebView2.Wpf;assembly=Microsoft.Web.WebView2.Wpf"
        mc:Ignorable="d"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="800"
        WindowState="Maximized"
        Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <ScrollViewer
        HorizontalScrollBarVisibility="Auto"
        VerticalScrollBarVisibility="Auto">
        <Grid Width="1505" Height="784">
            <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top" Margin="-28,-36,-30,0">
                <Border.Effect>
                    <DropShadowEffect BlurRadius="15" Opacity="0.15" ShadowDepth="1"/>
                </Border.Effect>
                <StackPanel>
                    <Image
                        Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
                        Width="133" Margin="0,47,1330,0" Height="73"/>
                    <Button Content="Меню" FontSize="35" FontFamily="Faberger"
                        Width="98"
                        Margin="0,-60,1000,0" Height="50" FontWeight="Regular" Foreground="#424242"
                        Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                    <Button Content="Рестораны" FontSize="35" FontFamily="Faberger"
                        Width="192"
                        Margin="0,-60,520,0" Height="50"
                        Style="{StaticResource DefaultButton}" FontWeight="Bold"

```

```

Click="Button_Restaurants_Click"/>
<Button Content="Корзина" FontSize="35" FontFamily="Faberger"
Width="148"
Margin="0,-60,-20,0" Height="50" Foreground="#424242"
Style="{StaticResource DefaultButton}" FontWeight="Regular"
Click="Button_Cart_Click"/>
<Button Content="История заказов" FontSize="35"
FontFamily="Faberger" Width="283"
Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
<Button x:Name="Button_Authorization" Content="Вход" FontSize="35"
FontFamily="Faberger" Width="91"
Margin="0,-60,-1260,0" Height="50"
Foreground="#424242" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
Click="Button_Authorization_Click"/>
<TextBlock x:Name="TextBlock_Authorization" Text="Здравствуйте, "
FontSize="35" FontFamily="Faberger"
Width="221" Margin="0,-75,-1260,0" Height="90" TextAlignment="Right"
Foreground="#424242" HorizontalAlignment="Center" Visibility="Hidden"/>
</StackPanel>
</Border>
<Grid Margin="54,146,54,46">
<wv2:WebView2 x:Name="webView" />
</Grid>
</Grid>
</ScrollViewer>
</Window>

```

## "Restaurants.xaml.cs"

```

using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Text;
using Newtonsoft.Json;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
// using System.Windows.Shapes;
using Microsoft.Web.WebView2.Core;
using Microsoft.Web.WebView2.Wpf;
using System.Net.Http;
using System.Net.Http.Json;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для Restaurants.xaml
    /// </summary>
    public partial class Restaurants : Window
    {
        public Restaurants()
        {
            InitializeComponent();
            if (AuthState.IsAuthenticated)
            {
                Button_Authorization.Visibility = Visibility.Hidden;
                TextBlock_Authorization.Visibility = Visibility.Visible;
            }
            this.PreviewKeyDown += AdminWindow_PreviewKeyDown;

            Loaded += async (s, e) => await UILoad();
        }
    }
}

```

```

    }

    public class ApiKeyResponse
    {
        public string ApiKey { get; set; }
    }

    private async Task UIload()
    {
        var keyResponse = await ApiClient.Instance.GetAsync("Map/api-key");
        keyResponse.EnsureSuccessStatusCode();
        var keyResult = await keyResponse.Content.ReadFromJsonAsync<ApiKeyResponse>();
        string apiKey = keyResult.ApiKey;
        string htmlPath = Path.Combine(AppDomain.CurrentDomain.BaseDirectory,
"Map.html"); ;
        string html = File.ReadAllText(htmlPath);
        html = html.Replace("{API_KEY}", apiKey);

        await webView.EnsureCoreWebView2Async(null);
        webView.NavigateToString(html);

        if (AuthState.IsAuthenticated)
        {
            var username = await ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getname");
            if (username.Length > 13) TextBlock_Authorization.Text =
$"Здравствуйте, \n{username.Substring(0, 11)}...";
            else TextBlock_Authorization.Text = $"Здравствуйте, \n{username}";
        }
    }

    private void Button_Authorization_Click(object sender, RoutedEventArgs e)
    {
        SignIn signin = new SignIn();
        signin.Show();
        Close();
    }

    private void Button_Menu_Click(object sender, RoutedEventArgs e)
    {
        Menu menu = new Menu();
        menu.Show();
        Close();
    }

    private void Button_Cart_Click(object sender, RoutedEventArgs e)
    {
        Cart cart = new Cart();
        cart.Show();
        Close();
    }

    private void Button_History_Click(object sender, RoutedEventArgs e)
    {
        History history = new History();
        history.Show();
        Close();
    }

    private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
    {
        Restaurants rest = new Restaurants();
        rest.Show();
        Close();
    }

    private void AdminWindow_PreviewKeyDown(object sender, KeyEventArgs e)
    {

```



```

        if (Keyboard.IsKeyDown(Key.LeftCtrl) && Keyboard.IsKeyDown(Key.LeftAlt) &&
Keyboard.IsKeyDown(Key.A) && e.Key == Key.D0)
        {
            Admin admin = new Admin();
            admin.Show();
            Close();
        }
    }
}
}

```

## "SignIn.xaml"

```

<Window x:Class="CourseWorkClient.SignIn"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервиса кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="800"
        WindowState="Maximized"
Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <Window.Background>
        <ImageBrush
ImageSource="C:\Users\fona\source\repos\CourseWorkClient\Media\maclogo.jpg"
TileMode="None" ViewportUnits="Absolute" Viewport="0,80,1535,720"/>
    </Window.Background>
    <ScrollViewer VerticalScrollBarVisibility="Auto"
HorizontalScrollBarVisibility="Auto">
        <Grid Width="1505" Height="784">
            <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top" Margin="-
28,-36,-30,0">
                <Border.Effect>
                    <DropShadowEffect BlurRadius="15" Opacity="0.15" ShadowDepth="1"/>
                </Border.Effect>
                <StackPanel>
                    <Image
Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
Width="133" Margin="0,47,1330,0" Height="73"/>
                    <Button Content="Меню" FontSize="35" FontFamily="Faberger"
Width="98"
Margin="0,-60,1000,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                    <Button Content="Рестораны" FontSize="35" FontFamily="Faberger"
Width="192"
Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
FontWeight="Regular" Foreground="#424242"
Click="Button_Restaurants_Click"/>
                    <Button Content="Корзина" FontSize="35" FontFamily="Faberger"
Width="148"
Margin="0,-60,-20,0" Height="50" Foreground="#424242"
Style="{StaticResource DefaultButton}" FontWeight="Regular"
Click="Button_Cart_Click"/>
                    <Button Content="История заказов" FontSize="35"
FontFamily="Faberger" Width="283"
Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
                    <Button Content="Вход" FontSize="35" FontFamily="Faberger"
Width="91"
Margin="0,-60,-1260,0" Height="50" FontWeight="Bold"
Foreground="Black" HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
Click="Button_Authorization_Click"/>
                </StackPanel>
            </Border>

```

```

        <Border Padding="30" HorizontalAlignment="Center" CornerRadius="30"
            Width="710" Background="#FFFAFA" Height="434"
VerticalAlignment="Top" Margin="0,219,0,0">
            <Border.Effect>
                <DropShadowEffect BlurRadius="37" Color="DarkGray"
ShadowDepth="10"/>
            </Border.Effect>
            <StackPanel Margin="-30,-31,-30,-30">
                <TextBlock Text="С возвращением!" FontSize="58" FontFamily="Mustica
Pro SemBd"
                    HorizontalAlignment="Center" Margin="0,5,0,0" />
                <Rectangle Name="animRectangle"
                    Width="301" Height="5"
                    Fill="Firebrick"
                    HorizontalAlignment="Left" Margin="83,6,0,0">
                    <Rectangle.Triggers>
                        <EventTrigger RoutedEvent="Rectangle.Loaded">
                            <BeginStoryboard>
                                <Storyboard>
                                    <DoubleAnimation
Storyboard.TargetName="animRectangle" Storyboard.TargetProperty="Width"
                                        From="0.0" To="301.0"
Duration="0:0:1.65" AccelerationRatio="0.1" DecelerationRatio="0.9"
                                        AutoReverse="False" />
                                </Storyboard>
                            </BeginStoryboard>
                        </EventTrigger>
                    </Rectangle.Triggers>
                </Rectangle>
                <Grid Margin="0,60,0,0">
                    <Button Content="Ещё нет аккаунта? Зарегистрироваться"
                        Margin="198,207,197,173"
                        Width="315" Height="21"
                        Background="Transparent" Foreground="Black"
                        HorizontalContentAlignment="Center"
VerticalContentAlignment="Center" Cursor="Hand"
                        FontSize="17" FontFamily="Faberge"
                        Click="Button_SignUpWindow_Click" >
                    <Button.Style>
                        <Style TargetType="Button">
                            <Setter Property="Template">
                                <Setter.Value>
                                    <ControlTemplate TargetType="Button">
                                        <Grid>
                                            <Border x:Name="border"
                                                CornerRadius="1"
Background="{TemplateBinding Background}"
                                                BorderThickness="0"
BorderBrush="Transparent">
                                            </Grid>
                                        <ContentPresenter
Content="{TemplateBinding Content}"
HorizontalAlignment="Center" VerticalAlignment="Center"
Focusable="False" />
                                    </ControlTemplate>
                                </Setter.Value>
                            </Setter>
                        </Style>
                    </Button>
                </Grid>
            </StackPanel>
        </Border>
    </Grid>
    <ControlTemplate.Triggers>
        <Trigger Property="IsMouseOver"
            <Setter TargetName="border"
                <Setter.Value>
                    <SolidColorBrush
Color="Gray" Opacity="0.4" />
                </Setter.Value>
            </Setter>
        </Trigger>
        <Trigger Property="IsPressed"
            Value="True">

```

```

        <Trigger.EnterActions>
            <BeginStoryboard>
                <Storyboard>
                    <ColorAnimation
Storyboard.TargetName="border"

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"

To="#212121"

Duration="0:0:0.0000001" />

                </Storyboard>
            </BeginStoryboard>
        </Trigger.EnterActions>
        <Trigger.ExitActions>
            <BeginStoryboard>
                <Storyboard>
                    <ColorAnimation

Storyboard.TargetName="border"

Storyboard.TargetProperty="(Border.Background).(SolidColorBrush.Color)"

To="Gray"

Duration="0:0:0.15" />

                </Storyboard>
            </BeginStoryboard>
        </Trigger.ExitActions>
    </Trigger>
</ControlTemplate.Triggers>
</ControlTemplate>
</Setter.Value>
</Setter>
</Style>
</Button.Style>
</Button>
<Button Content="Войти"
Margin="78,142,557,227"
Width="75" Height="32"
Cursor="Hand"
HorizontalAlignment="Center"
VerticalContentAlignment="Center"
FontSize="19" FontFamily="Faberge"
Background="Firebrick" Foreground="#DDFFFFFF"
Style="{StaticResource RedButton}"
Click="Button_SignIn_Click"/>
    <TextBox Name="loginTextBox"
materialDesign:HintAssist.Hint="Введите логин" Style="{StaticResource TypeInTextBox}"
Height="50" Margin="57,-18,257,369"
FontFamily="Faberge" FontSize="17"
Foreground="Black" CaretBrush="Black"
SelectionBrush="#FF130C1F"/>
    <PasswordBox Name="passPasswordBox"
materialDesign:HintAssist.Hint="Введите пароль"
Style="{StaticResource TypeInPasswordBox}"
Height="50" Margin="57,50,257,301"
FontFamily="Faberge" FontSize="17"
Foreground="Black" CaretBrush="Black"
SelectionBrush="#FF130C1F" />
    <TextBlock x:Name="ErrorTextBlock"
Text="Заполните все поля." FontSize="14"
FontFamily="Faberge"
Foreground="#c23030"
Margin="57,110,511,285"
Visibility="Hidden"/>
    <TextBlock x:Name="ErrorLoginOrPassTextBlock"
Text="Неверный логин или пароль." FontSize="14"
FontFamily="Faberge"
Foreground="#c23030"
Margin="57,110,469,285"

```

```

Visibility="Hidden"/>
        </Grid>
    </StackPanel>
</Border>
</Grid>
</ScrollView>
</Window>

```

## "SignIn.xaml.cs"

```

using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Net.Http.Json;
using System.Security.Cryptography;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для SignIn.xaml
    /// </summary>
    public partial class SignIn : Window
    {
        public SignIn()
        {
            InitializeComponent();
        }

        public class LoginResponse
        {
            public string Token { get; set; }
            public int id { get; set; }
        }

        private async void Button_SignIn_Click(object sender, RoutedEventArgs e)
        {
            string login = loginTextBox.Text.Trim();
            string password = passPasswordBox.Password.Trim();

            ErrorTextBlock.Visibility = Visibility.Hidden;
            ErrorLoginOrPassTextBlock.Visibility = Visibility.Hidden;
            TextBoxHelper.SetIsErrorState(loginTextBox, false);
            PasswordBoxHelper.SetIsErrorState(passPasswordBox, false);

            if (login == "")
            {
                ErrorTextBlock.Visibility = Visibility.Visible;
                TextBoxHelper.SetIsErrorState(loginTextBox, true);
            }
            else if (password == "")
            {
                ErrorTextBlock.Visibility = Visibility.Visible;
                PasswordBoxHelper.SetIsErrorState(passPasswordBox, true);
            }
            else

```

```

        {
            var requestData = new { Login = login, Password =
ApiClient.Encrypt(password) };
            var json = JsonSerializer.Serialize(requestData);
            var content = new StringContent(json, Encoding.UTF8,
"application/json");
            var response = await ApiClient.Instance.PostAsync("Users/authenticate",
content);
            if (response.StatusCode == System.Net.HttpStatusCode.NotFound ||
response.StatusCode == System.Net.HttpStatusCode.Unauthorized)
            {
                ErrorLoginOrPassTextBlock.Visibility = Visibility.Visible;
                TextBoxHelper.SetIsErrorState(loginTextBox, true);
                PasswordBoxHelper.SetIsErrorState(passPasswordBox, true);
                return;
            }
            var result = await response.Content.ReadFromJsonAsync<LoginResponse>();
            AuthState.Id = result.id;
            ApiClient.SetJwtToken(result.Token);
            if (await
ApiClient.Instance.GetFromJsonAsync<string>($"Users/{AuthState.Id}/getiscashier") ==
"false")
            {
                AuthState.isAuthenticated = true;
                Menu menu = new Menu();
                menu.Show();
                Close();
            } else
            {
                var place = await
ApiClient.Instance.GetFromJsonAsync<string?>($"Users/{AuthState.Id}/getplace");
                Cashier cashier = new Cashier(place);
                cashier.Show();
                Close();
            }
        }
    }

private void Button_SignUpWindow_Click(object sender, RoutedEventArgs e)
{
    SignUp signUp = new SignUp();
    signUp.Show();
    Close();
}

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
}

```

```

        Close();
    }
    private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
    {
        Restaurants rest = new Restaurants();
        rest.Show();
        Close();
    }
}
}

```

## "SignUp.xaml"

```

<Window x:Class="CourseWorkClient.SignUp"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
        xmlns:local="clr-namespace:CourseWorkClient"
        mc:Ignorable="d"
        xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"
        Title="Веб-сервис кафе «Макдональдс»"
        WindowStyle="ThreeDBorderWindow"
        MinHeight="200" MinWidth="270"
        Width="1505" Height="800"
        WindowState="Maximized"
        Icon="C:\Users\fona\source\repos\CourseWorkClient\McDonald's_logo.png">
    <Window.Background>
        <ImageBrush
            ImageSource="C:\Users\fona\source\repos\CourseWorkClient\Media\maclogo.jpg"
            TileMode="None" ViewportUnits="Absolute" Viewport="0,80,1535,720"/>
    </Window.Background>
    <ScrollViewer VerticalScrollBarVisibility="Auto"
        HorizontalScrollBarVisibility="Auto">
        <Grid Width="1505" Height="784">
            <Border Background="#FFFAFA" Height="136" VerticalAlignment="Top" Margin="-
28,-36,-30,0">
                <Border.Effect>
                    <DropShadowEffect BlurRadius="15" Opacity="0.15" ShadowDepth="1"/>
                </Border.Effect>
                <StackPanel>
                    <Image
                        Source="C:\Users\fona\source\repos\CourseWorkClient\Media\Golden Arches.png"
                        Width="133" Margin="0,47,1330,0" Height="73"/>
                    <Button Content="Меню" FontSize="35" FontFamily="Faberge"
                        Width="98"
                        Margin="0,-60,1000,0" Height="50" FontWeight="Regular" Foreground="#424242"
                        Style="{StaticResource DefaultButton}" Click="Button_Menu_Click"/>
                    <Button Content="Рестораны" FontSize="35" FontFamily="Faberge"
                        Width="192"
                        Margin="0,-60,520,0" Height="50" Style="{StaticResource DefaultButton}"
                        FontWeight="Regular" Foreground="#424242"
                        Click="Button_Restaurants_Click"/>
                    <Button Content="Корзина" FontSize="35" FontFamily="Faberge"
                        Width="148"
                        Margin="0,-60,-20,0" Height="50" Foreground="#424242"
                        Style="{StaticResource DefaultButton}" FontWeight="Regular"
                        Click="Button_Cart_Click"/>
                    <Button Content="История заказов" FontSize="35"
                        FontFamily="Faberge" Width="283"
                        Margin="0,-60,-680,0" Height="50" FontWeight="Regular" Foreground="#424242"
                        Style="{StaticResource DefaultButton}" Click="Button_History_Click"/>
                    <Button Content="Вход" FontSize="35" FontFamily="Faberge"
                        Width="91"
                        Margin="0,-60,-1260,0" Height="50" FontWeight="Bold"
                        Foreground="Black" HorizontalContentAlignment="Center"
                        VerticalContentAlignment="Center" Style="{StaticResource DefaultButton}"
                        Click="Button_Authorization_Click"/>
                </StackPanel>
            </Border>
        </Grid>
    </ScrollViewer>
</Window>

```

```

        </Border>
        <Border Padding="30" HorizontalAlignment="Center" CornerRadius="30"
            Width="710" Background="#FFFAFA" Height="500" VerticalAlignment="Top"
Margin="0,192,0,0">
            <Border.Effect>
                <DropShadowEffect BlurRadius="37" Color="DarkGray"
ShadowDepth="10"/>
            </Border.Effect>
            <StackPanel Margin="-30,-31,-30,-30">
                <TextBlock Text="Регистрация" FontSize="58" FontFamily="Mustica Pro
SemBd"
                    HorizontalAlignment="Center" Margin="0,12,0,0" />
                <Rectangle Name="animRectangle"
                    Width="301" Height="5"
                    Fill="Firebrick"
                    HorizontalAlignment="Left" Margin="83,6,0,0">
                    <Rectangle.Triggers>
                        <EventTrigger RoutedEvent="Rectangle.Loaded">
                            <BeginStoryboard>
                                <Storyboard>
                                    <DoubleAnimation
Storyboard.TargetName="animRectangle" Storyboard.TargetProperty="Width"
                                        From="0.0" To="301.0"
Duration="0:0:1.65" AccelerationRatio="0.1" DecelerationRatio="0.9"
                                        AutoReverse="False" />
                                </Storyboard>
                            </BeginStoryboard>
                        </EventTrigger>
                    </Rectangle.Triggers>
                </Rectangle>
                <Grid Margin="0,60,0,0">
                    <Button Content="Зарегистрироваться"
                        Margin="0,235,0,134" Height="32"
                        Cursor="Hand"
                        HorizontalAlignment="Center"
                        HorizontalContentAlignment="Center"
VerticalContentAlignment="Center"
                        FontSize="19" FontFamily="Faberge"
Background="Firebrick" Foreground="#DDFFFFFF"
                        Width="194" Style="{StaticResource RedButton}"
                        Click="ButtonReg_Click"/>
                    <TextBox x:Name="loginTextBox"
materialDesign:HintAssist.Hint="Введите логин" Style="{StaticResource TypeInTextBox}"
                        Height="50" Margin="57,-4,257,355"
                        FontFamily="Faberge" FontSize="17"
                        Foreground="Black" CaretBrush="Black"
SelectionBrush="#FF130C1F"/>
                    <PasswordBox x:Name="passwordPassBox"
materialDesign:HintAssist.Hint="Введите пароль" Style="{StaticResource
TypeInPasswordBox}"
                        Height="50" Margin="57,60,257,291"
                        FontFamily="Faberge" FontSize="17"
                        CaretBrush="Black" SelectionBrush="#FF130C1F" />
                    <PasswordBox x:Name="repeatPasswordPassBox"
materialDesign:HintAssist.Hint="Повторите пароль" Style="{StaticResource
TypeInPasswordBox}"
                        Height="50" Margin="57,124,257,227"
                        FontFamily="Faberge" FontSize="17"
                        CaretBrush="Black" SelectionBrush="#FF130C1F" />
                    <TextBlock x:Name="passwordErrorTextBlock"
                        Text="Пароли не совпадают!" FontSize="14"
FontFamily="Faberge"
                        Foreground="#c23030"
                        Margin="57,184,511,211"
                        Visibility="Hidden"/>
                    <TextBlock x:Name="ErrorTextBlock"
                        Text="Заполните все поля." FontSize="14"
FontFamily="Faberge"
                        Foreground="#c23030"
                        Margin="57,184,511,211"

```

```

                                Visibility="Hidden"/>
                                <TextBlock x:Name="ErrorLoginTextBlock"
Text="Такой логин занят." FontSize="14" FontFamily="Faberger"
Foreground="#c23030"
Margin="57,184,511,211"
Visibility="Hidden"/>
                                </Grid>
                                </StackPanel>
                                </Border>
                                </Grid>
                                </ScrollView>
</Window>

```

## "SignUp.xaml.cs"

```

using MaterialDesignColors;
using MaterialDesignThemes.Wpf;
using System;
using System.Collections.Generic;
using System.Net.Http.Json;
using System.Net.Http;
using System.Linq;
using System.Text;
using System.Text.Json;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Shapes;
using System.Security.Cryptography;
using System.IO;

namespace CourseWorkClient
{
    /// <summary>
    /// Логика взаимодействия для SignUp.xaml
    /// </summary>
    public partial class SignUp : Window
    {
        public SignUp()
        {
            InitializeComponent();
        }

        private async void ButtonReg_Click(object sender, RoutedEventArgs e)
        {
            string login = loginTextBox.Text.Trim();
            string password = passwordPassBox.Password.Trim();
            string repeatPassword = repeatPasswordPassBox.Password.Trim();

            passwordErrorTextBlock.Visibility = Visibility.Hidden;
            ErrorTextBlock.Visibility = Visibility.Hidden;
            ErrorLoginTextBlock.Visibility = Visibility.Hidden;
            TextBoxHelper.SetIsErrorState(loginTextBox, false);
            PasswordBoxHelper.SetIsErrorState(passwordPassBox, false);
            PasswordBoxHelper.SetIsErrorState(repeatPasswordPassBox, false);

            if (login == "")
            {
                ErrorTextBlock.Visibility = Visibility.Visible;
                TextBoxHelper.SetIsErrorState(loginTextBox, true);
            }
            else if (password == "")
            {
                ErrorTextBlock.Visibility = Visibility.Visible;
                PasswordBoxHelper.SetIsErrorState(passwordPassBox, true);
            }
        }
    }
}

```



```

    }
    else if (repeatPassword == "")
    {
        ErrorTextBlock.Visibility = Visibility.Visible;
        PasswordBoxHelper.SetIsErrorState(repeatPasswordPassBox, true);
    }
    else
    {
        if (await
ApiClient.Instance.GetFromJsonAsync<bool>($"Users/{login}/signup"))
        {
            ErrorLoginTextBlock.Visibility = Visibility.Visible;
            TextBoxHelper.SetIsErrorState(loginTextBox, true);
        }
        else if (password != repeatPassword)
        {
            passwordErrorTextBlock.Visibility = Visibility.Visible;
            PasswordBoxHelper.SetIsErrorState(repeatPasswordPassBox, true);
        }
        else
        {
            var requestData = new { Login = login, Password =
ApiClient.Encrypt(password) };
            var json = JsonSerializer.Serialize(requestData);
            var content = new StringContent(json, Encoding.UTF8,
"application/json");
            var response = await ApiClient.Instance.PostAsync($"Users/add",
content);

            SignIn signIn = new SignIn();
            signIn.Show();
            Close();
        }
    }

}

private void Button_Authorization_Click(object sender, RoutedEventArgs e)
{
    SignIn signin = new SignIn();
    signin.Show();
    Close();
}

private void Button_Menu_Click(object sender, RoutedEventArgs e)
{
    Menu menu = new Menu();
    menu.Show();
    Close();
}

private void Button_Cart_Click(object sender, RoutedEventArgs e)
{
    Cart cart = new Cart();
    cart.Show();
    Close();
}

private void Button_History_Click(object sender, RoutedEventArgs e)
{
    History history = new History();
    history.Show();
    Close();
}

private void Button_Restaurants_Click(object sender, RoutedEventArgs e)
{
    Restaurants rest = new Restaurants();
    rest.Show();
    Close();
}

}
}

```