|  |  |
| --- | --- |
| **Gerb-BMSTU_01** | **Министерство науки и высшего образования Российской Федерации**  Калужский филиал  федерального государственного бюджетного  образовательного учреждения высшего образования  ***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»***  ***(КФ МГТУ им. Н.Э. Баумана)*** |

**ФАКУЛЬТЕТ** \_ ***ИУ-КФ «Информатика и управление»\_\_\_\_\_\_\_\_\_***

**КАФЕДРА** \_\_ ***ИУ5-КФ «Системы обработки информации»***

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**к курсовой работе на тему:**

***Приложение-сервис по продаже домашней еды***

по дисциплине ***Базы данных***

Студент гр. ИУК5-51Б \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Шапина Ю.В. )

(подпись) (Ф.И.О.)

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ ( Кириллов В.Ю. )

(подпись) (Ф.И.О.)

Оценка руководителя \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка защиты \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_

30-50 (дата)

Оценка работы \_\_\_\_\_ баллов \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(оценка по пятибалльной шкале)

Комиссия: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

(подпись) (Ф.И.О.)

Калуга, 2020

Калужский филиал   
федерального государственного бюджетного образовательного учреждения высшего образования

***«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»   
(КФ МГТУ им. Н.Э. Баумана)***

УТВЕРЖДАЮ

Заведующий кафедрой **\_\_ИУК5\_\_\_**

\_\_\_\_\_\_\_\_\_\_\_(\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_)

«\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_\_г.

**ЗАДАНИЕ**

**на выполнение курсовой работы**

по дисциплине ***Базы данных***

Студент\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Шапина Ю.В., ИУК5-51Б\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы, индекс группы)

Руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Кириллов В.Ю. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(фамилия, инициалы)

График выполнения работы: 25% к\_4\_нед., 50% к\_7\_нед., 75% к\_10\_нед., 100% к\_14\_нед.

***1. Тема курсовой работы***

***Разработка приложения-сервиса по продаже домашней еды***

***2. Техническое задание***

*Разработать приложение-сервис по продаже домашней еды с использованием базы данных.*

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

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

***3. Оформление курсовой работы***

3.1. Расчетно-пояснительная записка на\_\_\_\_\_\_\_\_ листах формата А4.

3.2. Перечень графического материала КП (плакаты, схемы, чертежи и т.п.)\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

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

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

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

Дата выдачи задания «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20\_\_г.

Руководитель курсовой работы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/

(подпись) (Ф.И.О.)

Задание получил\_\_\_\_\_\_\_\_\_\_\_\_\_\_/\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_/ «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_20\_\_г.

(подпись) (Ф.И.О.)

Оглавление

[**1.** **ТЕХНИЧЕСКОЕ ЗАДАНИЕ** 5](#_Toc58958437)

[**1.1. Наименование.** 5](#_Toc58958438)

[**1.2. Основание для разработки.** 5](#_Toc58958439)

[**1.3. Исполнитель**. 5](#_Toc58958440)

[**1.4. Цель разработки.** 5](#_Toc58958441)

[**1.5. Содержание работы:** 5](#_Toc58958442)

[**1.5.1. Задачи, подлежащие решению.** 5](#_Toc58958443)

[**1.5.2. Требование к архитектуре АСОИ** 6](#_Toc58958444)

[**1.5.3. Требования к составу программных компонентов** 6](#_Toc58958445)

[**1.5.4. Требования к прикладным программам** 6](#_Toc58958446)

[**1.5.5. Требования к входным/выходным данным.** 6](#_Toc58958447)

[**1.5.6. Требования к временным характеристикам.** 6](#_Toc58958448)

[**1.5.7. Требования к составу технических средств.** 6](#_Toc58958449)

[**1.6. Этапы разработки.** 7](#_Toc58958450)

[**1.7. Техническая документация, предъявляемая по окончании работы.** 7](#_Toc58958451)

[**1.8. Дополнительные условия.** 8](#_Toc58958452)

[**2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ** 9](#_Toc58958453)

[**2.1 Постановка задачи** 9](#_Toc58958454)

[**2.2 Анализ существующих аналогов** 10](#_Toc58958455)

[**2.3. Выбор средств разработки** 15](#_Toc58958456)

[**2.4. Обоснование выбора среды разработки** 15](#_Toc58958457)

[**2.5 Перечень задач, подлежащих решению в процессе разработки** 16](#_Toc58958458)

[**2.6. Выводы** 17](#_Toc58958459)

[**3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ** 18](#_Toc58958460)

[**3.1 Общие сведения** 18](#_Toc58958461)

[**3.2 Структура приложения** 18](#_Toc58958462)

[**3.3. Структура данных** 18](#_Toc58958463)

[**3.3.1. Концептуальная модель данных** 18](#_Toc58958464)

[**3.3.2. Логическая модель данных** 19](#_Toc58958465)

[**3.3.3. Способ реализации** 21](#_Toc58958466)

[**3.4. Вывод** 22](#_Toc58958467)

[**4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ** 23](#_Toc58958468)

[**4.1 Требования к аппаратной платформе.** 23](#_Toc58958469)

[**4.2 Физическая модель данных.** 23](#_Toc58958470)

[**4.3. Тестирование приложения.** 29](#_Toc58958471)

[**4.4. Вывод** 38](#_Toc58958472)

[**Заключение** 39](#_Toc58958473)

[**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ** 40](#_Toc58958474)

1. **ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

### **1.1. Наименование.**

Сервис по продаже домашней еды «Домик в поселке городского типа».

### **1.2. Основание для разработки.**

Основанием для разработки сервиса является задание по курсовой работе в соответствии с учебным планом дисциплины «Базы данных».

### **1.3. Исполнитель**.

Студент группы ИУК5-51Б Шапина Юлия Вадимовна.

## **1.4. Цель разработки.**

Целью разработки является получение теоретических и практических знаний по работе с базами данных и создание сервиса по продаже домашних блюд.

## **1.5. Содержание работы:**

### **1.5.1. Задачи, подлежащие решению.**

1. Изучение предметной области, создание модели и структуры данных.
2. Реализация интерфейса.
3. Тестирование и оценка работоспособности программы.
4. Подведение итогов, анализ полученных результатов

### **1.5.2. Требование к архитектуре АСОИ**

Сервис «Домик в поселке городского типа» должен быть реализован на основе модульной архитектуры, где каждый модуль должен отвечать за различный функционал сервиса. Каждый модуль должен быть реализован в виде отдельного класса, а классы должны быть представлены в виде иерархической структуры.

### **1.5.3. Требования к составу программных компонентов**

Программа состоит из базы данных и ее программной оболочки.

### **1.5.4. Требования к прикладным программам**

Для работы системы необходимая операционная система Microsoft Windows 7/8/10 и минимальный набор драйверов.

### **1.5.5. Требования к входным/выходным данным.**

Требования к входным/выходным данным не предъявляются.

Входные данные:

* Пользовательские данные — логин, пароль, email, адрес
* Информация о блюдах

Выходные данные:

* Список доступных блюд
* Детальная информация

### **1.5.6. Требования к временным характеристикам.**

Требования к временным характеристикам не предъявляются.

### **1.5.7. Требования к составу технических средств.**

Для работы системы на компьютере необходимо, чтобы он обладал минимальными для своей операционной системы характеристиками:

* 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 ГГц или выше;
* 1 ГБ (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ;
* 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске;

## **1.6. Этапы разработки.**

1. Изучение предметной области
2. Создание модели данных
3. Реализация интерфейса, разработка программы
4. Тестирование программы

## **1.7. Техническая документация, предъявляемая по окончании работы.**

Должны быть разработаны следующие документы:

1. Расчетно-пояснительная записка:

– техническое задание;

– научно-исследовательская часть;

– проектно-конструкторская часть;

– проектно-технологическая часть;

– приложения.

2. Графическая часть – 2 листа формата А1 включающие в себя:

– демонстрационные чертежи;

– основные алгоритмы.

**1.8. Дополнительные условия.**

1. язык программирования С#;
2. среда разработки Visual Studio 2019;
3. использование базы данных для хранения информации

# 2. НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ

## **2.1 Постановка задачи**

На сегодняшний день миллионы людей покупают различные товары в электронных магазинах, не выходя из дому. Во всем мире огромными темпами растет количество интернет-пользователей и как следствие количество потенциальных «электронных» покупателей. Электронные магазины существенно уменьшают издержки производителя, расширяют рынки сбыта, расширяют возможности покупателя: покупать любой товар в любое время в любой стране, в любое время суток и года. Это дает электронным магазинам неоспоримое преимущество перед обычными магазинами, что является существенным моментом при переходе производителей с «обычной» торговли на «электронную».

Кроме очевидных плюсов интернет-сервисов можно выделить и другую важную особенность: в условиях пандемии наличие возможности заказа товара по интернету, с возможностью избежать крупного скопления людей, как это бывает в обычных торговых центрах, является решающим фактором при выборе магазина, поставляющего необходимые услуги. Острую необходимость в интернет-аналогах ощущают на себе предприятия питания, предоставляющие одну из самых востребованных человечеством услуг.

Кроме того, подобный интернет-сервис — отличная возможность удаленного заработка не только крупным предприятиям, но и малому бизнесу, в том числе людям, занимающимся домашним хозяйством и людям, оставшимся без работы из-за пандемии.

Главной задачей курсовой работы является получение практических навыков по работе с базой данных и создание приложения-сервиса по продаже домашней еды с ее использованием.

## **2.2 Анализ существующих аналогов**

На сегодняшний день существует множество сервисов для заказа еды на дом, позволяющих выбрать блюда из ассортимента множества ресторанов, кафе, баров и других предприятий питания, либо свои фирменные блюда.

Среди таких сервисов можно выделить DeliveryClub, Яндекс.Еда, ВкусНаДом.

Проведем анализ некоторых из них и выделим преимущества и недостатки.

1) Delivery Club

У сервиса Delivery Club имеется и компьютерная версия, и приложение, устанавливаемое на мобильные устройства. Он сотрудничает с огромным количеством заведений общественного питания – сегодня их более пяти тысяч по всей России. При этом выбор можно делать только из ресторанов, входящих в область доставки для адреса, введенного пользователем.

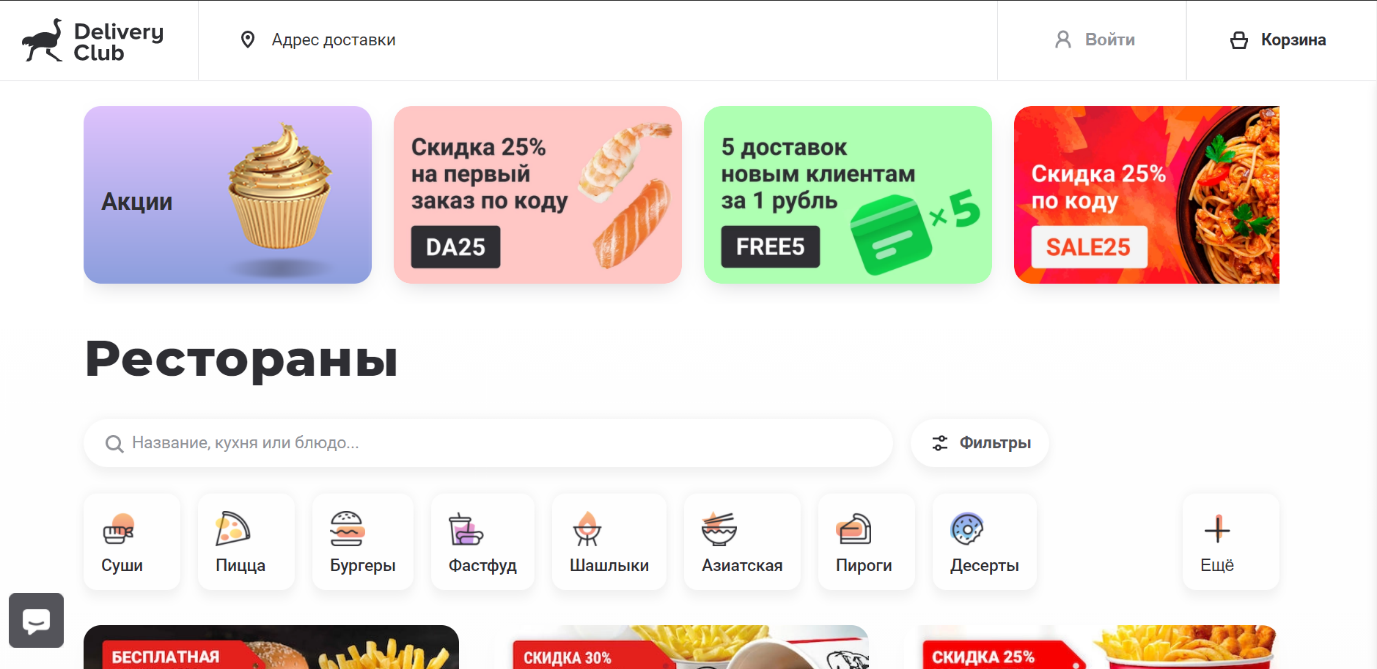


Рисунок 1 — начальная страница веб-сайта Delivery Club

Условия, касающиеся доставки и оплаты заказа, зависят от ресторана, который выбрал покупатель. Некоторые заведения устанавливают минимальную сумму заказа, и это можно назвать минусом – клиент не сможет заказать понравившуюся еду, если у него нет определенного количества денежных средств. Установленная сумма может находиться в пределах от трехсот до двух с половиной тысяч рублей (последнее часто встречается в Москве). Те же условия относятся и к стоимости доставки, при этом она может быть как платной, так и совершенно бесплатной (последнее привлекает большое количество покупателей). После оформления заказа в Деливери Клаб сотрудник компании оперативно звонит клиенту и уточняет все важные детали. Минус - сервис не предоставляет собственного меню, отсутствует возможность заказа домашней еды.

Подведем итог:

Плюсы:

* Мультиплатформенный сервис
* Большой ассортимент ресторанов
* Возможность указания города для уточнения доступных точек заказа

Минусы:

* Условия заказа разные для каждого ресторана-поставщика и ими же определяются

2) Яндекс.Еда

Пользуясь данной системой, можно выбрать и заказать любое понравившееся блюдо, после чего оно максимально оперативно будет доставлено сотрудником службы Yandex.Taxi.

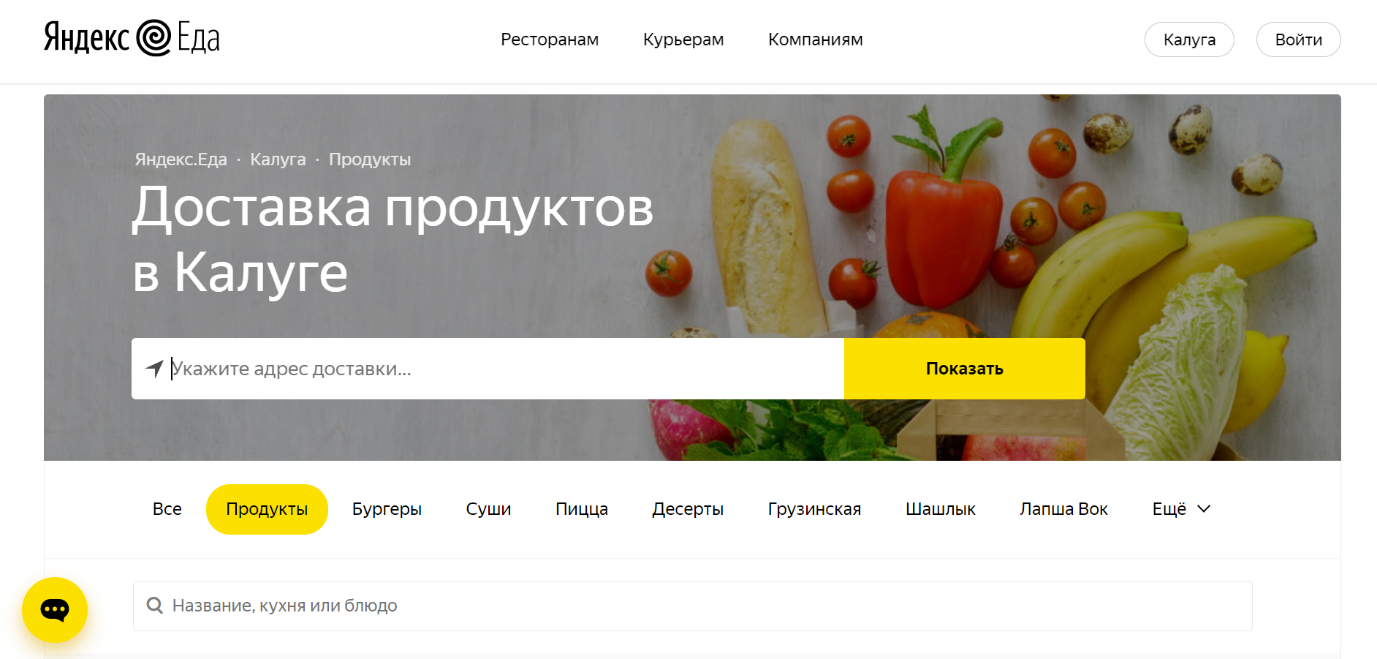


Рисунок 2 — главная страница веб-сайта сервиса Яндекс.Еда

Очевидный минус Яндекс.Еды, из-за которого он пока проигрывает в конкурентной борьбе с Delivery Club – сотрудничество с довольно небольшим количеством ресторанов. На сегодняшний день их чуть больше тысячи. Сервис сегодня работает как в Москве и Санкт-Петербурге, так и в большом количестве российских регионов, что очень удобно. Преимущество, привлекающее новых покупателей – отсутствие минимальной суммы заказа из конкретного заведения. Это означает, что купить еду можно на абсолютно любую сумму, заплатив дополнительно за доставку. Бесплатная доставка доступна тем, кто оформил заказ более чем на пятьсот рублей.

Подведем итоги:

Плюсы:

* Мультиплатформенный сервис
* Возможность указания города для уточнения доступных точек заказа
* Отсутствие минимальной суммы заказа

Минусы:

* Малый ассортимент ресторанов

3) ВкусНаДом

ВкусНаДом - один из сервисов, суть которого состоит в следующем: один раз в неделю клиенту доставляют свежие подготовленные продукты и рецепты на 5 ужинов на двоих, троих или четверых. Помимо ознакомительного промо-меню, у Вкус на дом на выбор есть специальные виды меню, например, вегетерианское или постное, также возможно выбрать блюда на свой вкус из предложенных на сайте.

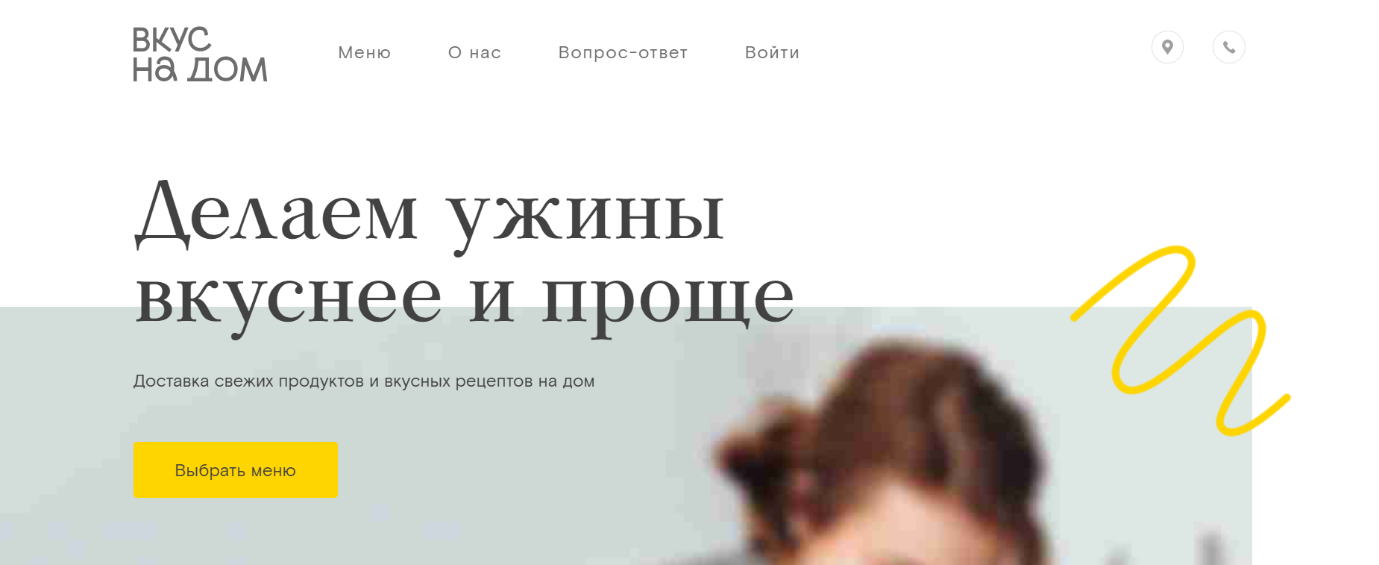


Рисунок 3 - главная страница веб-сайта сервиса ВкусНаДом

Доставка осуществляется только по Москве и Санкт-Петербургу, а также их областям.

Подведем итог:

Плюсы:

* Широкий ассортимент блюд
* Множество категорий на любой вкус

Минусы:

* Малая область доставки
* Блюда нужно готовить самому
* Не сотрудничают с другими предприятиями

Минус у всех перечисленных сервисов общий: нет возможности заказа домашней еды, как и нет возможности сотрудничества с данными сервисами, если вы не представляете крупное предприятие. Можно сделать вывод, что для поставленной задачи перечисленные приложения подходят не полностью, поэтому было принято решение разработать систему «Домик в поселке городского типа»

Сформируем представление об идеальном приложении, которое соответствует поставленной задаче:

* Мультиплатформенное
* Неограниченная область доставки
* Единые условия доставки
* Возможность пользователю приложения стать продавцом, из чего следует:
* Возможность добавлять и редактировать свои блюда
* Возможность пользователю-покупателю связываться с продавцом
* Получение уведомлений продавцам о поступающих заказах
* Возможность пользователям-покупателям оставлять комментарии, оценки к блюдам
* Возможность зайти в профиль продавца, чтобы увидеть все блюда, которые он предлагает

В рамках курсовой работы ввиду ограниченного времени будет спроектировано приложение, отвечающее следующим условиям:

* Оконное настольное приложение
* Есть возможность регистрации как покупатель либо как продавец
* Есть возможность покупателю сортировать блюда по категориям
* Есть возможность просмотреть полную информацию о блюде
* Есть возможность добавлять и удалять блюда из корзины, возможность оформить заказ
* Есть возможность продавцу добавлять и редактировать свои блюда

## **2.3. Выбор средств разработки**

Для решения поставленной задачи наилучшим образом подходит С#, потому что совмещает в себе свойства как высокоуровневых, так и низкоуровневых языков и широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования. Область его применения включает создание операционных систем, разнообразных прикладных программ, драйверов устройств, приложений для встраиваемых систем, высокопроизводительных серверов, а также игр.

C# поддерживает такие парадигмы программирования, как процедурное программирование, объектно-ориентированное программирование, обобщённое программирование. Язык имеет богатую стандартную библиотеку, которая включает в себя распространённые контейнеры и алгоритмы, ввод-вывод, регулярные выражения, поддержку многопоточности и другие возможности.

## **2.4. Обоснование выбора среды разработки**

Учитывая опыт разработки приложений на высокоуровневых языках, а также выбранный язык С#, идеальным вариантом для решения задачи является Microsoft Visual Studio – бесплатная (для личного пользования) среда разработки программного обеспечения для Windows, предназначенная для написания программ на множестве языков, включая С#. Visual Studio включает в себя: — Редактор исходного кода с возможностью простейшего рефакторинга. — Встроенный отладчик (может работать как отладчик уровня исходного кода, так и отладчик машинного уровня). — Редактор форм для упрощения создания графического интерфейса приложения — веб-редактор — дизайнер классов и дизайнер схемы базы данных. Visual Studio позволяет создавать и подключать сторонние дополнения для расширения функциональности практически на каждом уровне, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения.

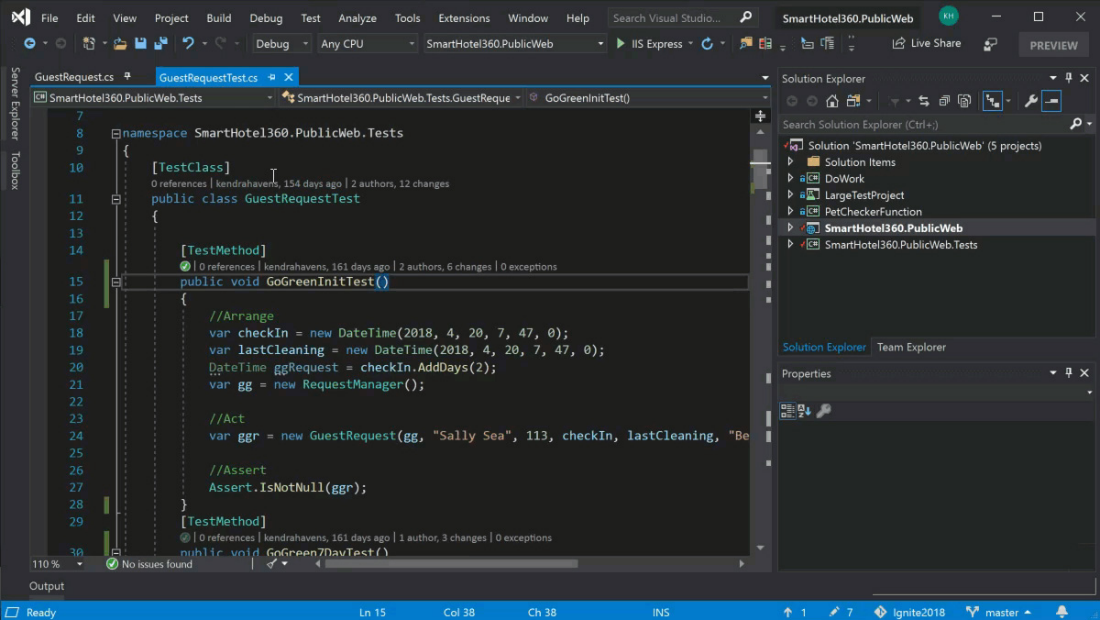


Рисунок 4 – Microsoft Visual Studio

## **2.5 Перечень задач, подлежащих решению в процессе разработки**

Для выполнения поставленной задачи необходимо решить следующие задачи:

* изучение предметной области
* разработка базы данных для хранения информации о пользователях и блюдах,
* проектировка и разработка приложения и его последующее тестирование.

## **2.6. Выводы**

Было принято решение использовать следующие инструменты: в качестве языка программирования был выбран С#. Для реализации сервиса была выбрана интегрированная среда разработки Microsoft Visual Studio.

# 3. ПРОЕКТНО-КОНСТРУКТОРСКАЯ ЧАСТЬ

**3.1 Общие сведения**

Данная курсовая работа представляет собой приложение-сервис «Домик в поселке городского типа», предназначенный для покупки и продажи домашней еды.

**3.2 Структура приложения**

Приложение «Домик в поселке городского типа» на базе ОС Windows представляет собой оконное приложение с возможностями регистрации, входа, просмотра информации о доступных товарах, отсортированных по категориям, добавления своих товаров, заказа товаров.

**3.3. Структура данных**

**3.3.1. Концептуальная модель данных**

При проектировании концептуальной модели были выделены следующие сущности: Пользователь, Поставщик, Товар, Корзина, Заказ, Категория.

Сущность Пользователь необходима для хранения данных о пользователе.

Сущность Поставщик необходима для хранения данных о продавце.

Сущность Товар необходима для хранения всей информации о блюде.

Сущность Категория является справочной для заполнения информации о блюде в базе. Эта сущность служит для сортировки блюд по категориям.

Сущность Корзина необходима для создания списка блюд, которые хочет заказать пользователь.

Сущность Заказ необходима для создания списка утвержденных для заказа блюд и передачи информации продавцу.

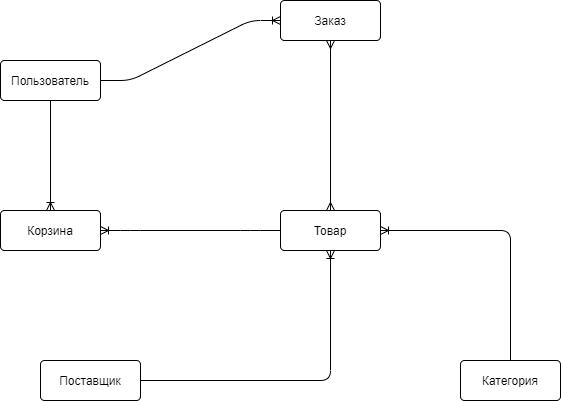


Рисунок 5 — Концептуальная модель данных

### **3.3.2. Логическая модель данных**

На этапе проектирования логической схемы данных были определены атрибуты сущностей, добавлена сущность ЗаказТовар.

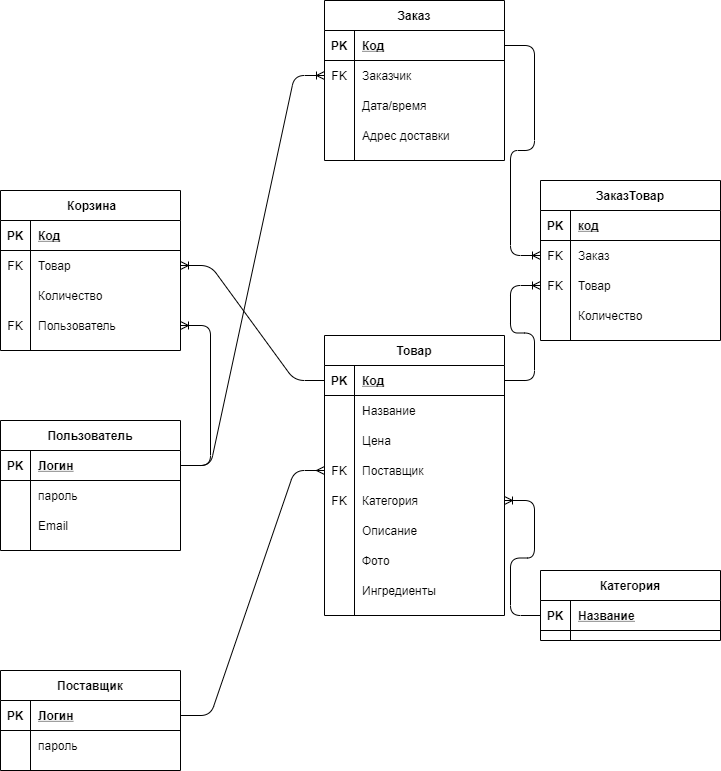


Рисунок 6 — Логическая модель данных

Атрибуты Логин и Пароль сущностей Пользователь и Поставщик необходимы для хранения логина и пароля пользователя или продавца для успешного входа в систему. Атрибут Email необходим для регистрации и для связи с пользователем или продавцом.

Сущность Товар содержит следующие атрибуты: Код, Название, Цена, Поставщик, Категория, Описание, Фото, Ингредиенты. Данная сущность имеет связь с сущностью ЗаказТовар для определения блюд в заказе. Атрибуты Поставщик, Категория и Код связываются с сущностями Поставщик, Категория и Корзина соответственно. Атрибут Цена содержит цену для каждого блюда. Атрибут Описание содержит описательную информацию о блюде, которую вводит продавец. Атрибут фото необходим для хранения пути к изображению книги, загруженному продавцом. Атрибут Ингредиенты содержит список составляющих блюдо продуктов.

Сущность Категория является справочной для заполнения информации о блюде в базе. Эта сущность служит для сортировки блюд по категориям.

Сущность Корзина содержит атрибуты Код, Товар, Количество, Пользователь и необходима для создания списка блюд, которые хочет заказать пользователь. Атрибуты Товар и Пользователь связываются с сущностями Товар и Пользователь соответственно.

Сущность Заказ содержит атрибуты Код, Заказчик, Дата/Время, Адрес доставки и необходима для создания списка утвержденных для заказа блюд и передачи информации продавцу. В атрибуте Дата/Время указывается дата, к которой блюдо должно быть готово и доставлено покупателю. В атрибуте Адрес доставки указывается адрес, по которому блюдо должно быть доставлено. Атрибут заказчик связывается с сущностью Пользователь.

Сущность ЗаказТовар содержит атрибуты Заказ, Товар и Количество. Атрибуты Заказ и Товар служат для связи таблиц Заказ и Товар соответственно.

### **3.3.3. Способ реализации**

Для реализации приложения была использована система для построения клиентских приложений WPF (Windows Presentation Foundation) — графическая подсистема в составе .NET Framework, использующая язык декларативного описания интерфейса XAML.

Для взаимодействия с базой данных используются пространство имен System.Data.SqlClient и следующие пользовательские классы на его основе:

ControlModule — класс подключает базу данных и содержит коллекции данных, является родительским для FoodControl и Registrative.

FoodControl — класс, управляющий данными о блюдах, включает в себя методы, переносящие информацию из приложения в БД и наоборот.

Registrative — класс с методами аутентификации.

**3.4. Вывод**

В ходе конструкторской части курсовой работы была определена структура приложения, которое будет выполнено в виде оконного приложения с возможностью купить или продать домашнюю еду.

Были разработаны концептуальная и логическая модели данных.

Был разработан интерфейс приложения, функционал реализован с помощью C# и WPF с использованием пространства имен System.Data.SqlClient.

**4. ПРОЕКТНО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ**

## **4.1 Требования к аппаратной платформе.**

Исходя из характеристик разработанного приложения и ресурсов, требуемых для его исправной работы, необходимо, чтобы компьютер обладал следующими минимальными характеристиками:

* 32-разрядный (x86) или 64-разрядный (x64) процессор с тактовой частотой 1 ГГц или выше;
* 1 ГБ (для 32-разрядного процессора) или 2 ГБ (для 64-разрядного процессора) ОЗУ;
* 16 ГБ (для 32-разрядной системы) или 20 ГБ (для 64-разрядной системы) свободного места на жестком диске;

## **4.2 Физическая модель данных.**

На основании разработанной логической модели была реализована физическая модель данных.

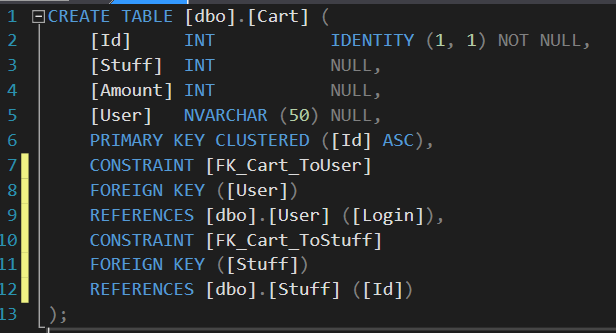


Рисунок 7 — Формирование таблицы Cart

Таблица Cart основана на сущности Корзина и имеет четыре колонки: Id — первичный ключ, Stuff и Amount типа INT для хранения айди заказа, айди заказанного блюда и количество соответственно, User типа NVARCHAR(50) для хранения логина пользователя (заказчика). User — вторичный ключ, связующий таблицу Cart с таблицей User. Stuff — вторичный ключ, связующий таблицу Cart с таблицей Stuff. Для сохранения целостности данные, хранящиеся в столбцах, должны соответствовать данным из связанных таблиц.

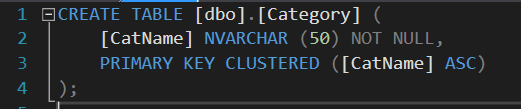


Рисунок 8 — Формирование таблицы Category

Таблица Category основана на сущности Категория и имеет только одну колонку: CatName типа NVARCHAR(50), предназначенную для хранения названий категорий блюд.

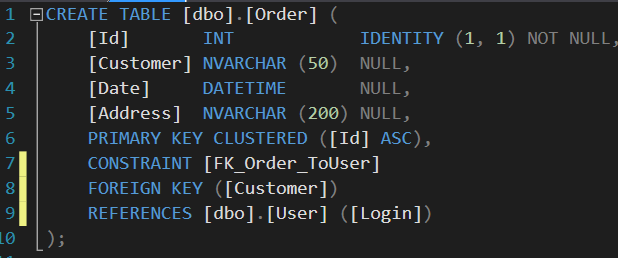


Рисунок 9 — Формирование таблицы Order

Таблица Order основана на сущности Заказ и имеет четыре колонки: Id — первичный ключ, предназначенный для хранения айди заказа, Customer типа NVARCHAR(50) для хранения логина заказчика, Date типа DATETIME для хранения даты и времени доставки, Address типа NVARCHAR(200) для хранения длинного адреса доставки. Customer — вторичный ключ, связующий таблицу Order с таблицей User. Для сохранения целостности данные, хранящиеся в столбцах, должны соответствовать данным из связанных таблиц.

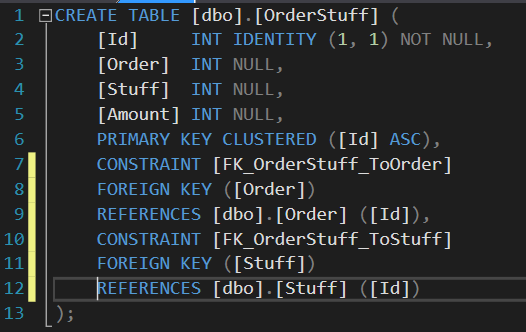


Рисунок 10 — Формирование таблицы OrderStuff

Таблица OrderStuff основана на сущности ЗаказТовар и имеет четыре колонки: Id — первичный ключ, Order, Stuff и Amount типа INT для хранения айди заказа, айди товара и количества товара соответственно. Order — вторичный ключ, связующий таблицу OrderStuff с таблицей Order. Stuff — вторичный ключ, связующий таблицу OrderStuff с таблицей Stuff. Для сохранения целостности данные, хранящиеся в столбцах, должны соответствовать данным из связанных таблиц.

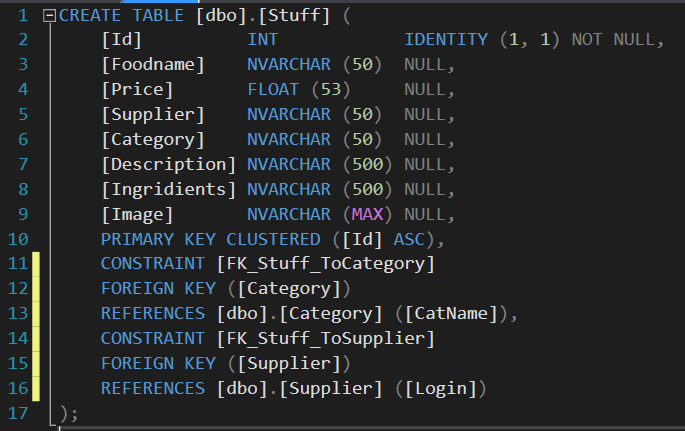


Рисунок 11 — Формирование таблицы Stuff

Таблица Stuff основана на сущности Товар и имеет восемь колонок: Id — первичный ключ, Foodname, Supplier, Category типа NVARCHAR(50) для хранения названия блюда, логина продавца и названия категории соответственно; Price типа FLOAT для хранения цены блюда; Description и Ingridients типа NVARCHAR(500) для хранения длинного описания блюда и списка ингредиентов соответственно; Image типа NVARCHAR(MAX) Category — вторичный ключ, связующий таблицу Stuff с таблицей Category. Supplier — вторичный ключ, связующий таблицу Stuff с таблицей Supplier. Для сохранения целостности данные, хранящиеся в столбцах, должны соответствовать данным из связанных таблиц.

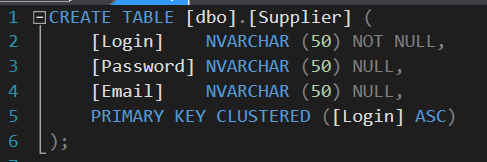


Рисунок 12 — Формирование таблицы Supplier

Таблица Supplier основана на сущности Поставщик и имеет три колонки: Login, Password и Email типа NVARCHAR(50) для хранения логина продавца, пароля продавца и его адрес электронной почты соответственно. Столбец Login является первичным ключом и не может быть пустым.

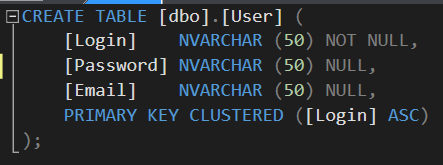


Рисунок 13 — Формирование таблицы User

Таблица User основана на сущности Пользователь и имеет три колонки: Login, Password и Email типа NVARCHAR(50) для хранения логина пользователя, пароля пользователя и его адрес электронной почты соответственно. Столбец Login является первичным ключом и не может быть пустым.

## **4.3. Тестирование приложения.**

Для тестирования был использован персональный компьютер.

При запуске приложения открывается начальная страница — страница входа в систему.

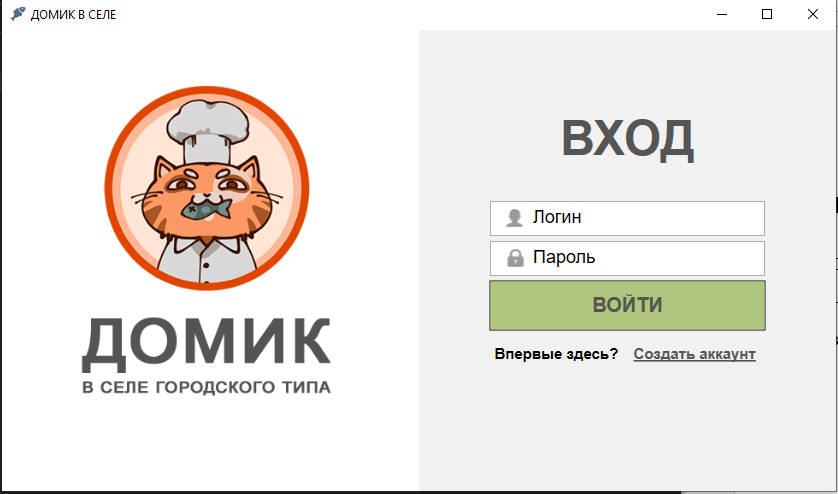


Рисунок 14 — Страница входа

При нажатии на «Создать аккаунт» открывается страница регистрации. Все поля должны быть корректно заполнены для успешной регистрации и дальнейшего использования приложения. Обязательным является выбор типа аккаунта: «Хочу кушать» как пользовательский аккаунт и «хочу готовить» как аккаунт продавца. Без выбора типа аккаунта регистрация не проходит даже при остальных заполненных полях.

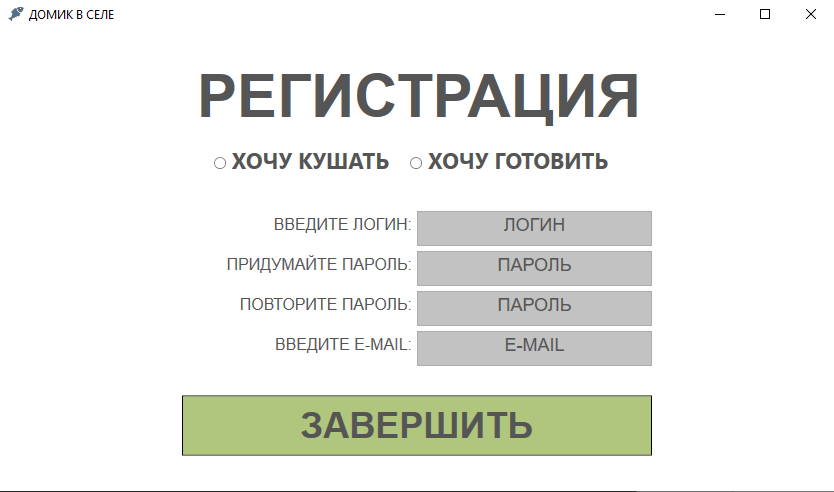


Рисунок 15 — Страница регистрации

Для тестирования был прописан логин и email пользователя, но поля пароля оставлены пустыми. При нажатии кнопки «завершить» на странице появляется сообщение об ошибке с просьбой заполнить пустые поля.

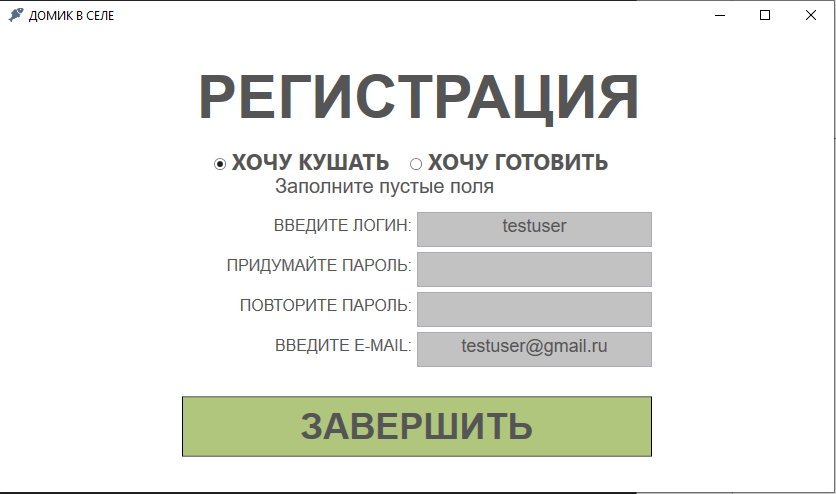


Рисунок 16 — сообщение об ошибке регистрации

Затем были заполнены все поля, но пароль намеренно не совпадал. Была выявлена ошибка, в ходе которой такая регистрация считалась успешной. Ошибка была исправлена корректировкой метода и добавлением соответствующего сообщения об ошибке.

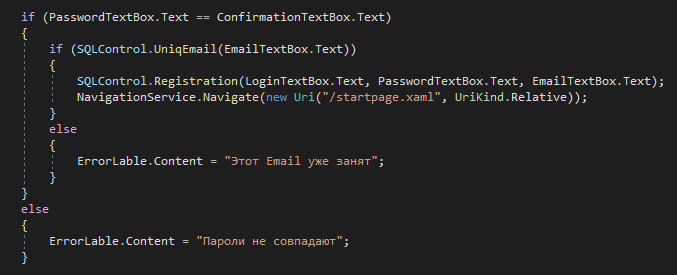


Рисунок 17 — скорректированный метод



Рисунок 18 — сообщение об ошибке (пароли не совпадают)

При тестировании регистрации в качестве продавца была обнаружена ошибка, при которой можно было зарегистрироваться под логином, существующим среди пользователей. При попытке входа система распознавала такой логин как логин пользователя, и на соответствующий аккаунт продавца зайти было невозможно. Ошибка была исправлена доработкой метода проверки.

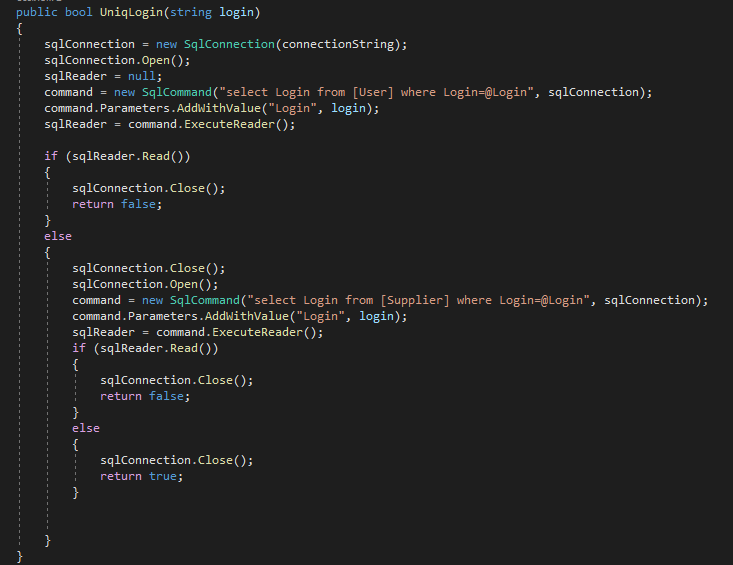


Рисунок 19 — доработанный метод проверки уникальности логина



Рисунок 20 — сообщение об ошибке (логин уже занят)

После завершения регистрации происходит перенаправление на страницу авторизации. Была добавлена надпись об ошибке, если авторизация не проходит успешно.

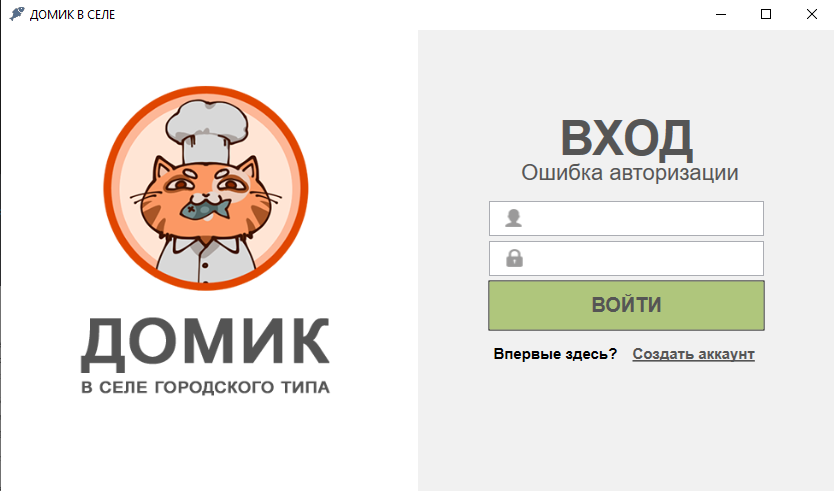


Рисунок 21 — сообщение об ошибке авторизации

На странице продавца есть возможность добавить блюдо. В ходе тестирования этой функции была устранена ошибка некорректного отображения категорий. Так же была добавлена возможность редактирования уже добавленного блюда щелчком по нему. Кнопка «Добавить» меняется на «Изменить»

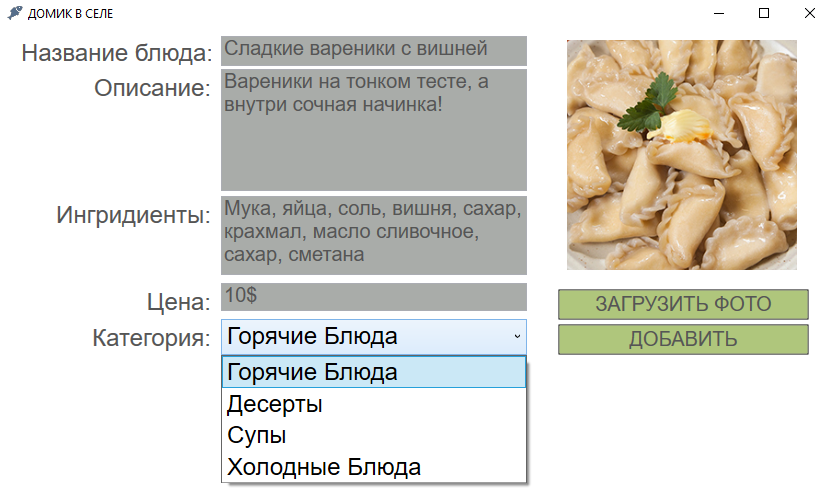


Рисунок 22 — корректное отображение страницы добавления блюда

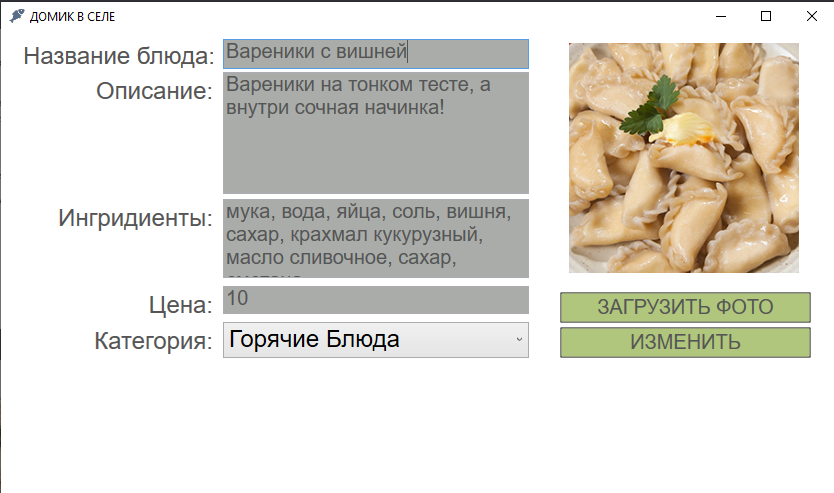


Рисунок 23 — Функция изменения

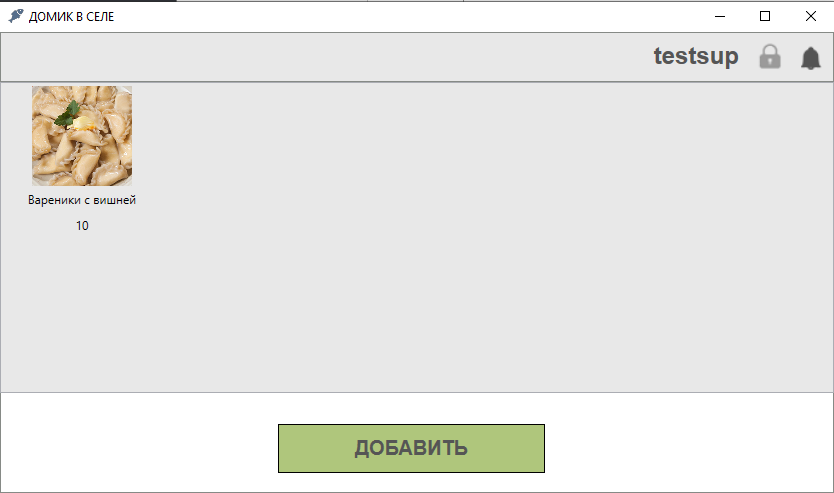


Рисунок 24 — Корректное отображение измененного блюда

В ходе тестирования функционала аккаунта пользователя была выявлена ошибка, при которой не работали кнопки быстрого добавления блюд в корзину, и от них было решено отказаться.

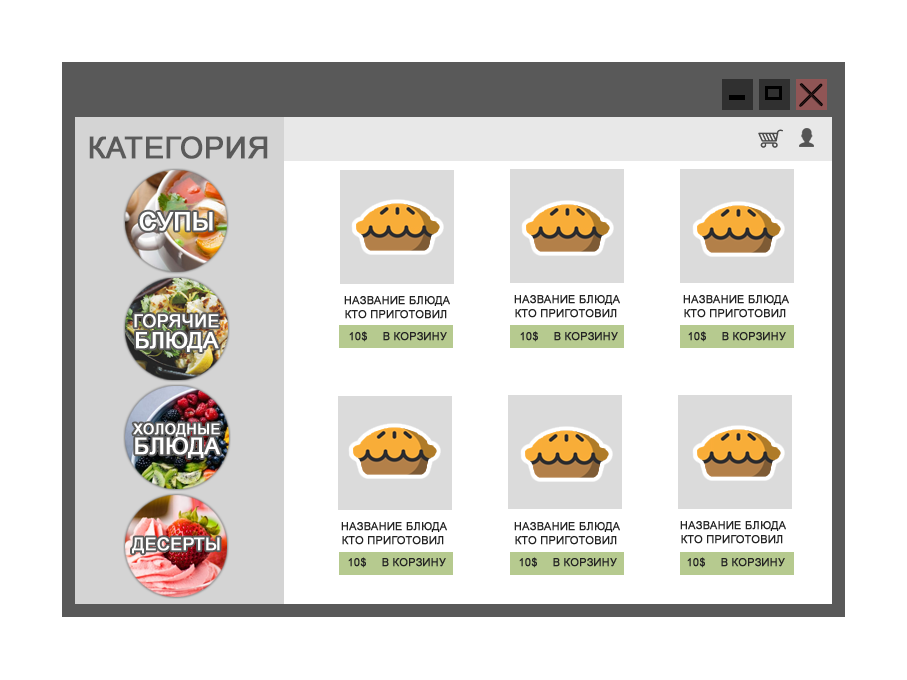


Рисунок 25 — Изначальный концепт

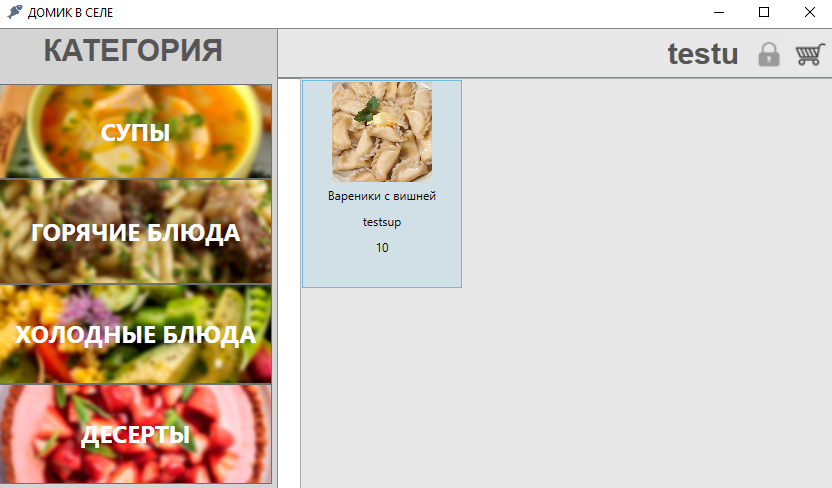


Рисунок 26 — измененный интерфейс

Была выявлена ошибка, из-за которой в заказе неправильно рассчитывалось время желаемой доставки. Ошибка была исправлена доработкой метода. Была исправлена ошибка, из-за которой кол-во символов поле ввода адреса было ограничено.

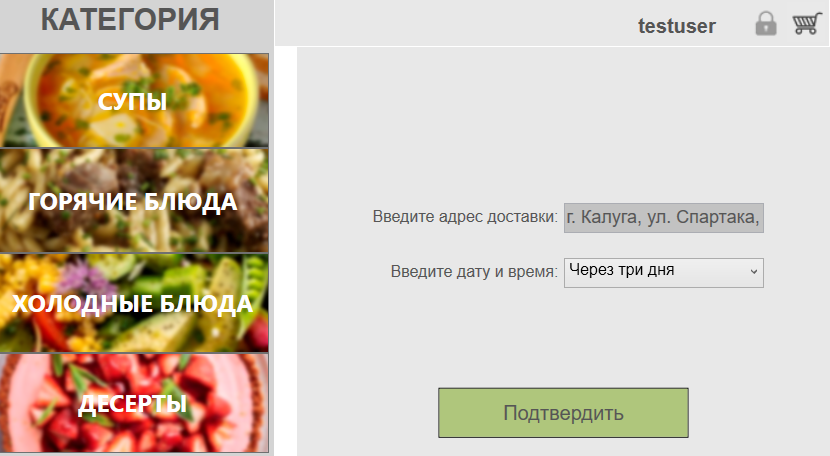


Рисунок 27 — исправленное поле ввода

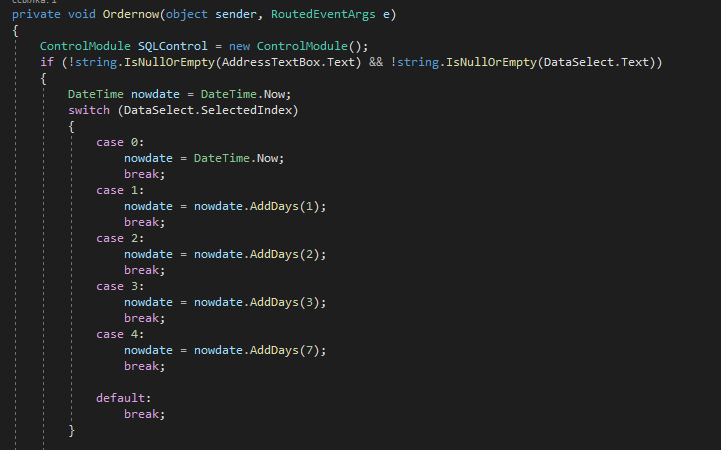


Рисунок 28 — исправленный метод

**4.4. Вывод**

В ходе проектно-технологической части работы были определены системные требования для разработанного приложения.

Было проведено тестирование, в ходе которого были найдены и исправлены ошибки, и затем подтверждена корректная работа приложения.

# Заключение

В результате выполнение курсовой работы было разработано приложение-сервис «Домик в поселке городского типа», предназначенное для покупки и продажи домашней еды.

В ходе работы были проанализированы существующие программы-аналоги. По результатам исследования было принято решение разработать свое приложение и определен функционал.

В ходе работы были достигнуты все поставленные задачи, такие как:

1. Изучение предметной области, проектирование концептуальной и логической моделей данных.
2. Разработка базы данных для хранения информации о пользователях и блюдах.
3. Разработка и тестирование приложения.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абрамян, А.В. Разработка пользовательского интерфейса на основе технологии Windows Presentation Foundation: учебник по курсу «Основы разработки пользовательского интерфейса» для студентов направления 02.03.02 «Фундаментальная информатика и информационные технологии» (бакалавриат) / А.В. Абрамян, М.Э. Абрамян ; Южный федеральный университет. – Ростов-на-Дону ; Таганрог : Южный федеральный университет, 2018. – 302 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=499453 (дата обращения: 15.12.2020).
2. Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования/ Э. Гамма, Р. Джонсон, Р. Хелм – Спб.: Питер, 2019. – 368 с.
3. Горелов, С.В. Современные технологии программирования: разработка Windows-приложений на языке С#: учебник для студентов, обучающихся по дисциплине «Современные технологии программирования», направление «Прикладная информатика» (09.03.03 — для бакалавров, 09.04.03 — для магистров) : в 2 томах : [16+] / С.В. Горелов ; под науч. ред. П.Б. Лукьянова ; Финансовый университет при Правительстве Российской Федерации. – Москва : Прометей, 2019. – Том 1. – 363 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=576037 (дата обращения: 15.12.2020).
4. Документация по С# [Электронный ресурс]/ Microsoft – URL: https://docs.microsoft.com/ru-ru/dotnet/csharp//(дата обращения 15.12.2020).
5. Зайцев, М.Г. Объектно-ориентированный анализ и программирование : учебное пособие : [16+] / М.Г. Зайцев ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2017. – 84 с. : ил., табл. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=576800 (дата обращения: 15.12.2020).
6. Кудрявцев, К. Я. Создание баз данных : учебное пособие / К. Я. Кудрявцев. — Москва : НИЯУ МИФИ, 2010. — 155 с. — ISBN 978-5-7262-1302-6. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/75822 (дата обращения: 15.12.2020).
7. Рояк, М.Э. Программирование под Windows графических интерфейсов пользователя : учебное пособие : [16+] / М.Э. Рояк, И.М. Ступаков ; Новосибирский государственный технический университет. – Новосибирск : Новосибирский государственный технический университет, 2018. – 72 с. : ил. – Режим доступа: по подписке. – URL: https://biblioclub.ru/index.php?page=book&id=575018 (дата обращения: 15.12.2020).
8. Тарасов, С. В. СУБД для программиста. Базы данных изнутри / С. В. Тарасов. — Москва : СОЛОН-Пресс, 2015. — 320 с. — ISBN 978-2-7466-7383-0. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/64959 (дата обращения: 15.12.2020)
9. Шнырёв, С. Л. Базы данных : учебное пособие / С. Л. Шнырёв. — Москва : НИЯУ МИФИ, 2011. — 224 с. — ISBN 978-5-7262-1483-2. — Текст : электронный // Лань : электронно-библиотечная система. — URL: https://e.lanbook.com/book/75809 (дата обращения: 15.12.2020).
10. Microsoft Visual Studio [Электронный ресурс]/ Википедия – Режим доступа: https://ru.wikipedia.org/wiki/Microsoft\_Visual\_Studio(дата обращения 15.12.2020).