**Оглавление**

[Аннотация 4](#_Toc72446649)

[Введение 5](#_Toc72446650)

[1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ 6](#_Toc72446651)

[1.1 Основное функциональное назначение программного средства 6](#_Toc72446652)

[1.2 Полное наименование программного средства 6](#_Toc72446653)

[1.3 Условное обозначение программного средства 6](#_Toc72446654)

[1.4 Разработчики программного средства 6](#_Toc72446655)

[2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ 7](#_Toc72446656)

[2.1 Основание для разработки 7](#_Toc72446657)

[2.2 Назначение разработки 7](#_Toc72446658)

[2.3 Требование к программному средству 7](#_Toc72446659)

[2.4 Требования к программной документации 8](#_Toc72446660)

[2.5 Требования к эргономике и технической эстетике 8](#_Toc72446661)

[2.6 Стадии и этапы разработки 9](#_Toc72446662)

[2.7 Порядок контроля и приемки 10](#_Toc72446663)

[3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ 11](#_Toc72446664)

[3.1 Декомпозиция поставленной задачи 11](#_Toc72446665)

[3.2 Общая архитектура программного средства 12](#_Toc72446666)

[3.3 Разработка алгоритма решения задачи 15](#_Toc72446667)

[3.4 Реализация функционального назначения программного средства 16](#_Toc72446668)

[3.5 Структурная организация данных 17](#_Toc72446669)

[3.6 Разработка интерфейса ПС 22](#_Toc72446670)

[3.7 Описание структуры выходной информации 36](#_Toc72446671)

[4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ 39](#_Toc72446672)

[4.1 Назначение программного средства 39](#_Toc72446673)

[4.2 Условия выполнения программного средства 39](#_Toc72446674)

[4.3 Эксплуатация программного средства 39](#_Toc72446675)

[4.4 Сообщения пользователю 54](#_Toc72446676)

[5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА 55](#_Toc72446677)

[5.1 Определение верхней границы количества тестов 55](#_Toc72446678)

[5.2 Разработка тестовых вариантов 56](#_Toc72446679)

[5.3 Составление отчета о тестировании 56](#_Toc72446680)

[ЗАКЛЮЧЕНИЕ 64](#_Toc72446681)

[Оценка качества программного средства с помощью метрик 64](#_Toc72446682)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 67](#_Toc72446683)

[Приложение А Текст кода программного средства 69](#_Toc72446684)

[Приложение Б Спецификация 98](#_Toc72446685)

[Приложение B Диск с программным средством 100](#_Toc72446686)

**Аннотация**

На дипломную работу студента Мошкова Р.Д. на тему: «Служба доставки еды «Шеф»». Дипломная работа представлена пояснительной запиской на 92 листа машинописного текста. Состоит из введения, 5 глав, заключения и списка литературных источников.

В главе 1 содержится общие сведения о программном средстве. В главе 2 приведено техническое задание на проектирование БД согласно ГОСТ 19.201. Глава 3 содержит пояснительную записку к программному продукту согласно ГОСТ 19.404-79. В главе 4 содержится руководство пользователя согласно ГОСТ 19.504-79. Глава 5 содержит обоснование количества необходимых тестов и тестовые пакеты для оценки работоспособности программного продукта. В главе 6 приведены результаты ручного и автоматического тестирования программного продукта и выполнена оценка качества с помощью метрик.

В работе содержится общее описание программного средства, предназначенного для объекта проектирования – «Службы доставки еды «Шеф»», изложены требования к программному средству, описаны его архитектура, реализация функций, интерфейс, приведено руководство пользователя, обоснованы тестовые пакеты и выполнена оценка качества программного продукта с помощью метрик.

**Введение**

В дипломной работе была поставлена задача разработки приложения для службы доставки еды «Шеф», который позволяет автоматизировать доставку товаров.

Для достижения поставленной цели был составлен основной алгоритм работы программы, а также разработаны алгоритмы вспомогательных процедур.

Для реализации алгоритмов был выбран язык программирования C#. Выбор обусловлен тем, что возможностей этого языка достаточно для достижения поставленной цели моей дипломной работы.

При разработке интерфейса программы, я учитывал наиболее нужные и важные функциональные возможности, которые должна выполнять данная программа, а также тип пользователей, которые с ней будут работать.

**1 ОБЩИЕ СВЕДЕНИЯ О ПРОГРАММНОМ СРЕДСТВЕ**

**1.1 Основное функциональное назначение программного средства**

Программа предназначена для контроля и упрощения доставки еды. Программа не требует владения современными информационными технологиями для эффективного использования данного программного средства, т.е. подходит для любого пользователя.

**1.2 Полное наименование программного средства**

«Служба доставки еды «Шеф»».

**1.3 Условное обозначение программного средства**

«ChefApp 1.0»

**1.4 Разработчики программного средства**

Программное средство разработала студент Зеленодольского Механического колледжа группы 195 Мошков Р.Д.

**2 ТЕХНИЧЕСКОЕ ЗАДАНИЕ**

**2.1 Основание для разработки**

Работа ведётся на основании задания на дипломную работу по специальности 09.02.03 — «Программирование в компьютерных системах».

**2.2 Назначение разработки**

Программа «ChefApp 1.0» работает под управление операционной системы Windows 7, Windows 8, Windows 10. Программа предназначена для контроля и упрощения доставки еды и сопутствующих товаров. Цель создания программы для службы доставки еды. Программа позволяет:

* Создавать и редактировать базу товаров
* Регистрироваться в системе как покупатель
* Оформлять заказы
* Создавать отзывы

**2.3 Требование к программному средству**

«ChefApp 1.0» устанавливается на персональный компьютер, имеющий представленные минимальные системные требования:

Оперативная память: 4 Гб;

Процессор: Intel Core i / Phenom II X4 / AMD Athlon64 X2 4600 + 2.4 ГГц;

Видео карта: nVidia GeForce 8600 / ATI Radeon HD 2400;

Звуковая карта: Совместимая с DirectX;

Программный продукт «ChefApp 1.0» является кроссплатформенным продуктом, что позволяет не ограничиваться в работе с определенной версией ОС.

«ChefApp 1.0» написана на объектно-ориентированном языке программировании C# (си-шарп) в интегрированной среде разработки Visual Studio 2019. В качестве СУБД используется MS SQL. Программа не может подлежать сторонней оптимизации и редактированию программного кода.

**2.4 Требования к программной документации**

Состав программной документации должен включать в себя:

1. техническое задание;

2. сроки выполнения работ;

3. программу и методику испытаний;

4. эксплуатационные инструкции пользователю;

**2.5 Требования к эргономике и технической эстетике**

Система обеспечивает удобный для пользователей системный интерфейс, отвечающий следующим требованиям:

- единый стиль оформления для пользовательских интерфейсов;

- удобная, интуитивно понятная навигация в интерфейсе пользователя;

- взаимодействие пользователя с системой осуществляется на двух языках по выбору (русский и английский), для наибольшего удобства

Пользовательские интерфейсы системы спроектированы и разработаны с применением единых принципов графического представления информации и организации доступа к функциональным возможностям и сервисам. Разработан графический дизайн пользовательских интерфейсов, цветовые, шрифтовые и композиционные решения для отображения текстов, изображений, таблиц, гиперссылок, управляющих и навигационных элементов (меню, кнопок, форм и т.п.).

Экранные формы спроектированы с учетом требований унификации:

- все экранные формы пользовательского интерфейса выполнены в едином графическом дизайне, с одинаковым расположением основных элементов управления и навигации;

- для обозначения сходных операций использованы сходные графические значки, кнопки и другие управляющие (навигационные) элементы. Термины, используемые для обозначения типовых операций (добавление информационной сущности, редактирование поля данных), а также последовательности действий пользователя при их выполнении унифицированы;

- внешнее поведение сходных элементов интерфейса (реакция на наведение указателя «мыши», переключение фокуса, нажатие кнопки) реализованы одинаково для однотипных элементов.

**2.6 Стадии и этапы разработки**

Разработка должна быть проведена в три стадии:

1. разработка технического задания;

2. рабочее проектирование;

3. внедрение.

На стадии разработки технического задания должен быть выполнен этап разработки, согласования и утверждения настоящего технического задания.

На стадии рабочего проектирования должны быть выполнены перечисленные ниже этапы работ:

1. разработка программы;

2. разработка программной документации;

3. испытания программы.

На этапе разработки технического задания должны быть выполнены перечисленные ниже работы:

1. постановка задачи;

2. определение и уточнение требований к техническим средствам;

3. определение требований к программе;

4. определение стадий, этапов и сроков разработки программы и документации на нее;

5. выбор языков программирования;

6. согласование и утверждение технического задания.

На этапе разработки программы должна быть выполнена работа по программированию (кодированию) и отладке программы.

На этапе разработки программной документации должна быть выполнена разработка программных документов в соответствии с пунктом Предварительный состав программной документации настоящего технического задания.

На этапе испытаний программы должны быть выполнены перечисленные ниже виды работ:

1. разработка, согласование и утверждение порядка и методики испытаний;

2. проведение испытаний;

3. корректировка программы и программной документации по результатам испытаний.

**2.7 Порядок контроля и приемки**

Сдача-приёмка работ производится поэтапно, в соответствии с рабочей программой и календарным планом. Сдача-приемка осуществляется комиссией, в состав которой входят представители ГАПОУ и работодатель. По результатам приемки подписывается акт приемочной комиссии.

В указанном подразделе, согласно таблице «Сроки выполнения работ» технического задания, будет описаны перечень участвующих организаций, место и сроки проведения работ, согласно п. 2.8 ГОСТ 34.602-89.

Порядок согласования и утверждения приемочной документации должен регламентироваться организационно-распорядительной документацией организации, принимающей участие в создании программного продукта. Согласно разделу «Приемка результатов разработки» ГОСТ 15.001-88 для согласования и утверждения приемочной документации создается приемочная комиссия (приказом).

Статус приемочной комиссии определяется Заказчиком до проведения испытаний.

**3 ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ПРОГРАММНОМУ ПРОДУКТУ**

**3.1 Декомпозиция поставленной задачи**

Для достижения цели работы была поставлена следующая задача:

Создать модуль “CodeDatabase” отвечающий за сохранение БД. Результат работы модуля - сохранение БД.

Поставленные задачи наглядно отображаются на диаграмме деятельности.

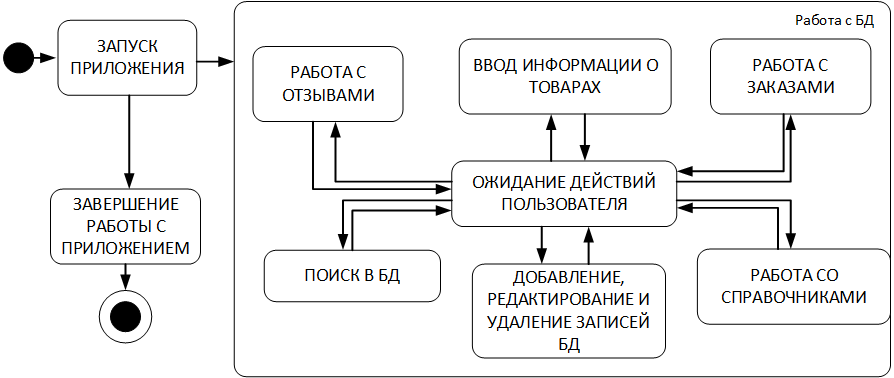


Рис.1 Декомпозиция поставленной задачи

**3.2 Общая архитектура программного средства**

Объектом моделирования является Служба доставки еды «Шеф».

Рассматриваемые процессы: поиск и просмотр сведений.

Объекты моделирования представлены на диаграмме классов, рис.2.

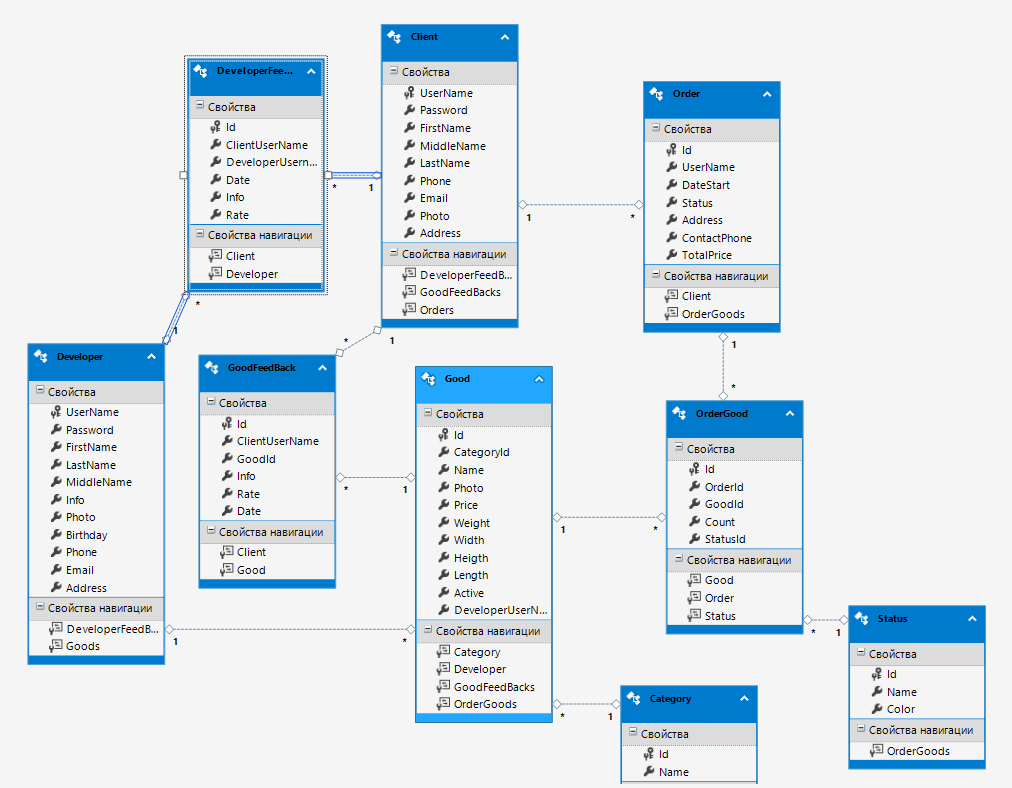


Рис.2 Диаграмма классов

По полученной декомпозиции задач была спроектирована следующая архитектура программного средства.

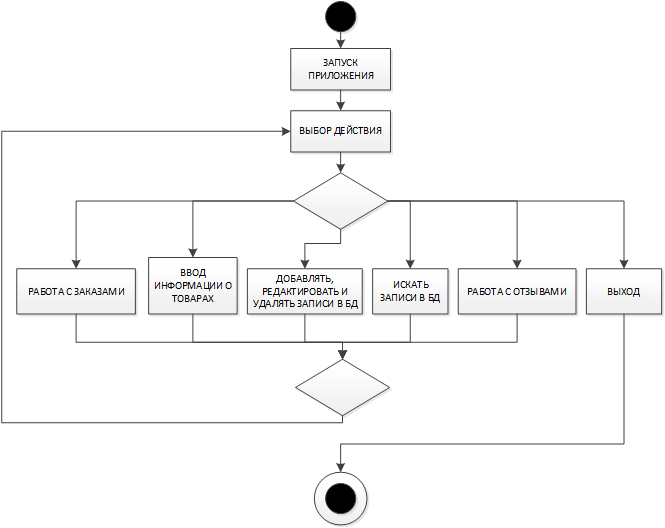


Рис.3 Схема событийно-ориентированной архитектуры

В «ChefApp 1.0» используется событийно-ориентированная архитектура.(Рис.3) Роль агента (источник событий) в программе выполняют: пользователи программы, за роль стоков (потребители событий) отвечают таблицы, входящие в базу данных. Например, когда пользователь выбирает определенное действие: добавить, удалить, редактировать, вывести в Word и т.д. система осуществит выбранные действия, и база отреагирует соответствующим образом: запись добавлена, удалена, отредактирована, либо отчет выведен в Excel. Системная архитектура пользователя рассматривает это изменение состояния как событие, создаваемое, публикуемое, определяемое и потребляемое различными приложениями в составе архитектуры.

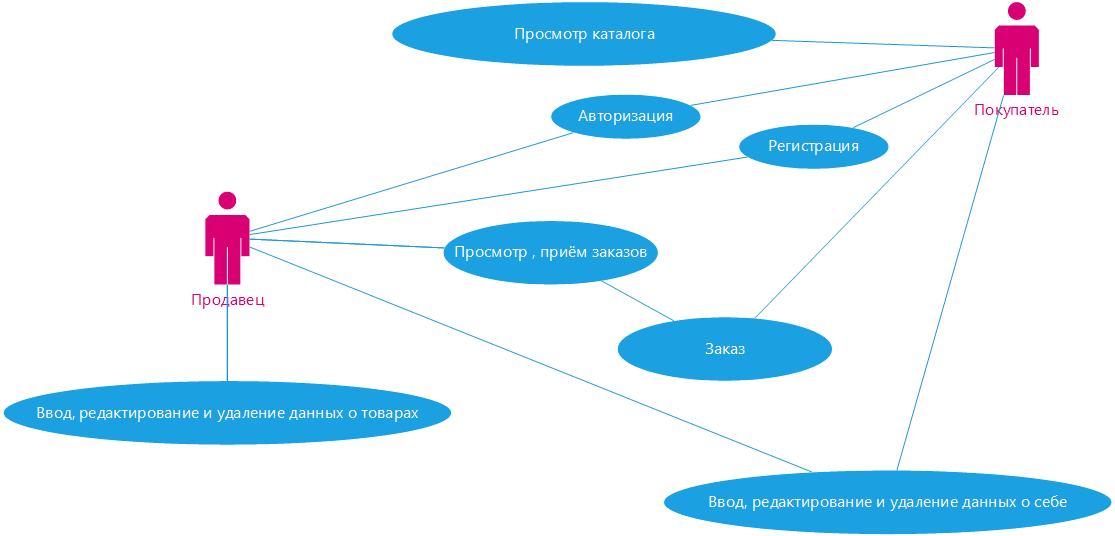
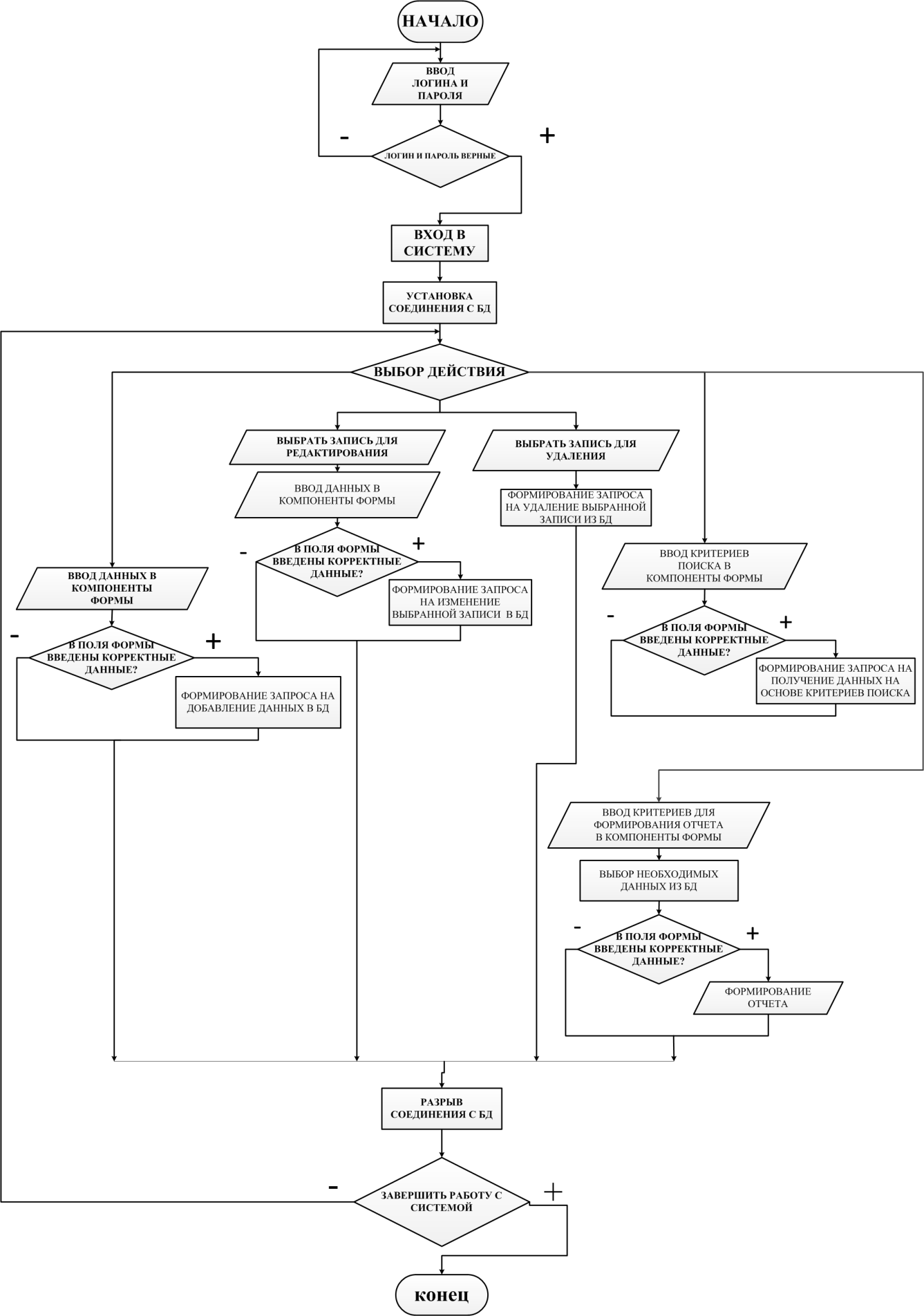


Рис.4 Диаграмма вариантов использования

**3.3 Разработка алгоритма решения задачи**

**Общий алгоритм**

****Рис.5 Алгоритм программы

**3.4 Реализация функционального назначения программного средства**

Программа «ChefApp 1.01» имеет следующий набор входных данных, такие как: товары, сведения о клиентах, заказы.

Данные вводятся администратором в соответствующие поля ввода, снабженные всплывающими подсказками.

Выходными данными являются: информация о доставке, квитанции.

Данные вводятся пользователем в соответствующие поля вывода, снабженные всплывающими подсказками.

Выходные данные редактировать вручную невозможно, т.к. они служат в качестве отчетной информации для пользователя.

**3.5 Структурная организация данных**

Для создания БД необходимо определиться с данными, которые необходимы для полноценного функционирования системы. Все эти данные указаны в реляционной модели «БД Служба доставки еды «Шеф»» представленная на рисунке. Любая реляционная база данных и называется реляционной, что характеризуется отношениями (relation) между таблицами. На рисунке изображены таблицы моей базы данных. При этом одна таблица является родительской (главной), а вторая – дочерней (подчиненной). Главной таблицей является «Заказ». Реляционная модель автоматизированной системы соответствует всем 12 правилам Кодда. (рис.6)

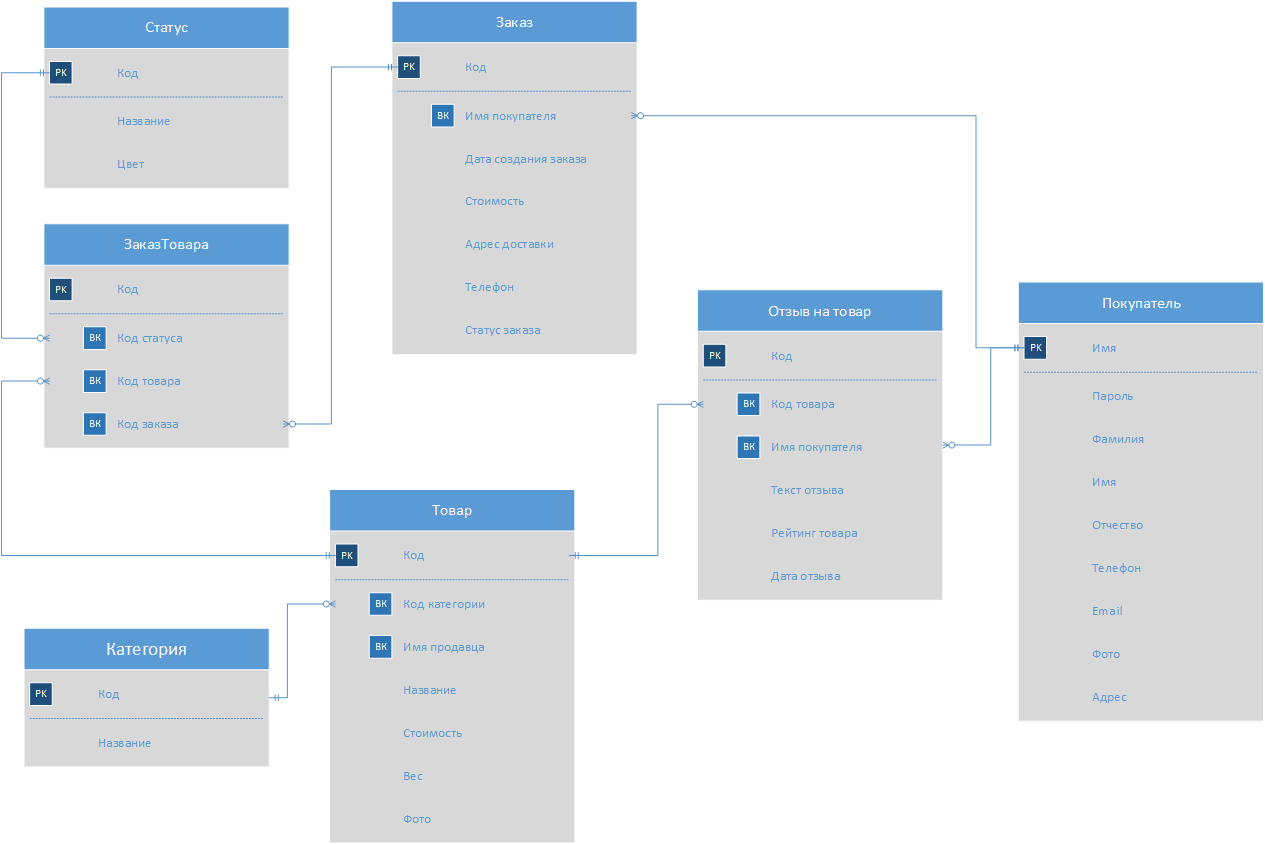


Рис.6 Реляционная модель

Первичный ключ в базе уникален, используется для организации отношений между таблицами, который не может иметь пустых и повторяющихся значений. Первичными ключами в базе являются поля: Код (таблица ЗаказТовара), Код (Таблица Заказ), Код (Таблица Товар), Код (таблица Отзыв на товар), Код (таблица Статус), Код (Таблица Заявка), Код (таблица Категория), Имя (таблица Покупатель). Остальные ID-ключи являются внешними ключами.

Для организации более эффективной обработки данных применяется нормализация. Таблицы моей БД находятся в 3НФ:

* БД находится в форме -1НФ потому, что
  + Таблица не имеет повторяющихся записей;
  + Каждый атрибут отношения хранит одно-единственное значение и не является списком, ни множеством значений;
  + Таблица не имеет повторяющихся групп полей.
    - Вторая нормальная форма(2НФ):
      * Устранены атрибуты, зависящие только от части уникального (первичного) идентификатора, т.е. ID.
        + Третья нормальная форма(3НФ):

Отсутствуют атрибуты, зависящие от атрибутов, не входящих в уникальный (первичный) ключ.

На основе реляционной модели базы данных мною в MS SQL была построена следующая база данных. (Рис.7) В ней первичным ключом является поле таблицы “Order” – “Id”. Связь таблиц “Good” и “OrderGood” по полю “GoodId” один ко многим (1:М). Связь таблиц “Category” и “Good” по полю “CategoryId” один ко многим (1:М). Связь таблиц “Order” и “OrderGood” по полю “OrderId” один ко многим (1:М). Связь таблиц “Status” и “OrderGood” по полю “StatusId” один ко многим (1:М). Связь таблиц “Client” и “GoodFeedBack” по полю “ClientUserName” один ко многим (1:М). Связь таблиц “Good” и “GoodFeedBack” по полю “GoodId” один ко многим (1:М). Связь таблиц “Client” и “Good” по полю “ClientUserName” один ко многим (1:М). Все связи между таблицами базы данных типа один ко многим.



Рис.7 Схема базы данных

Таблица “Status”(Статус)

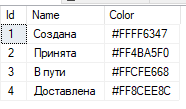


Рис.8 Таблица Status

Таблица “Category”(Категории)

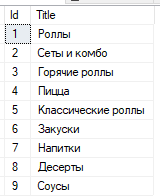


Рис.9 Таблица Category

Таблица “Good”(Товар)

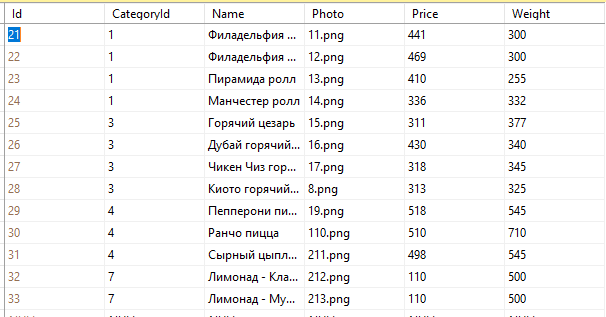


Рис.10 Таблица Good

Таблица “GoodFeedBack”(Отзывы на товары)

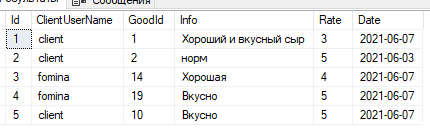


Рис.11 Таблица GoodFeedBack

Таблица “Order”(Заявка)

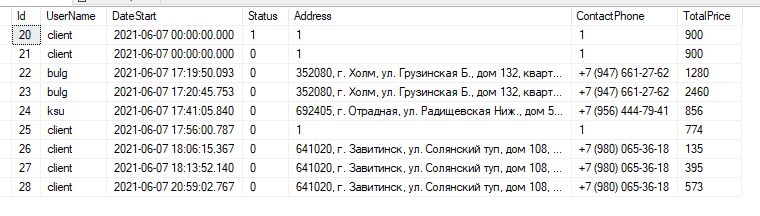


Рис.12 Таблица Order

Таблица “OrderGood”(Заявки на товары)

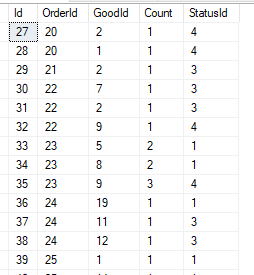


Рис.13 Таблица OrderGood

**3.6 Разработка интерфейса ПС**

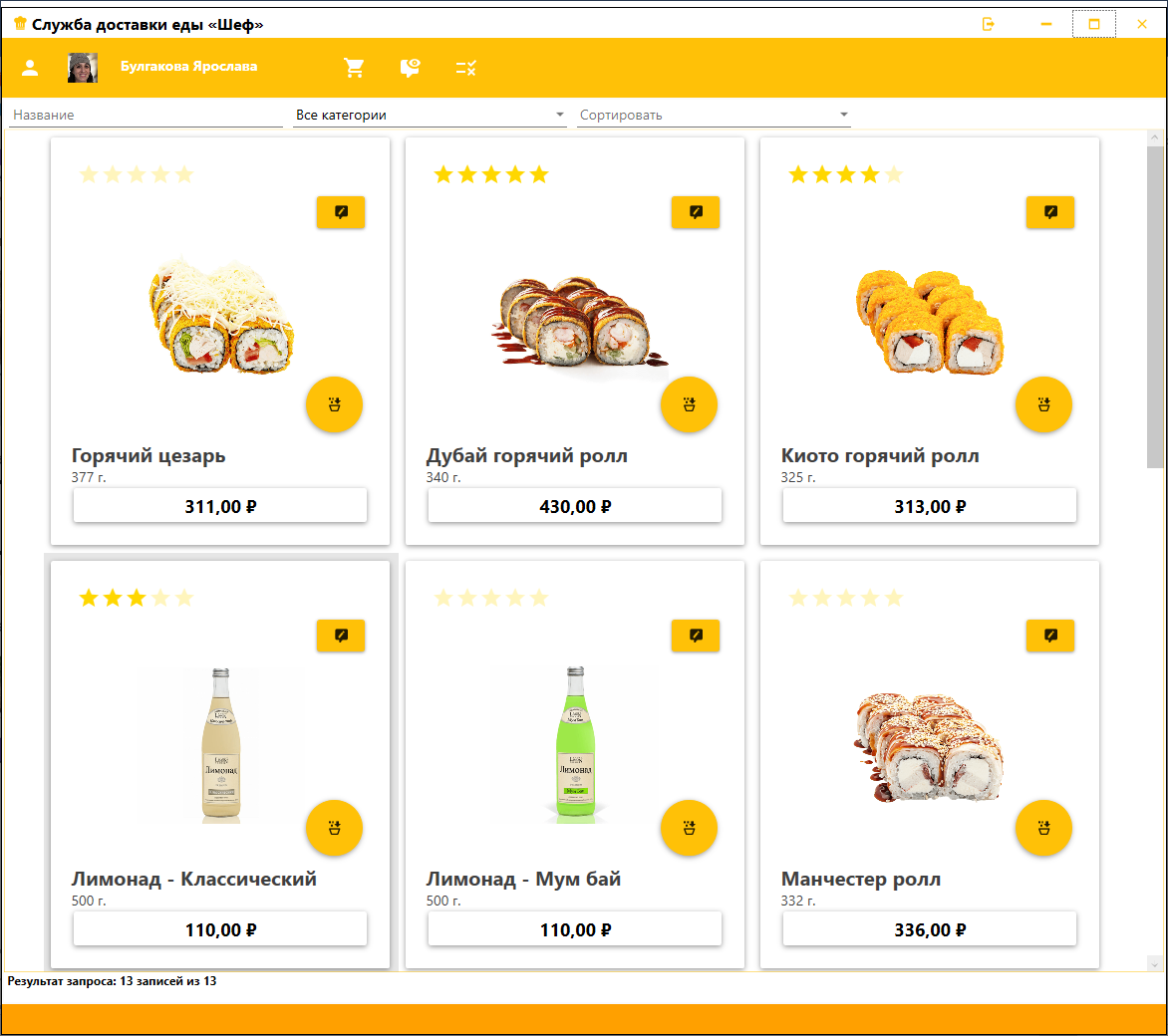
Приложение предназначено для двух типов пользователей: покупателя и администратора. Для входа в систему для полного функционала необходимо ввести свои учётные данные. Для запуска приложения достаточно щелкнуть дважды левой кнопкой мыши по ярлыку.

После запуска программы пользователь приступит к работе с главной формой программы, на которой в виде плиточного интерфейса отображаются товары. «ChefApp 1.01» обладает удобным интерфейсом, так что даже неопытному в обращение с пользовательскими компьютерными программами пользователю не составит труда быстро привыкнуть к работе с данным программным средством.

На рабочем экране расположены несколько кнопок для комфортной работы с приложением.

**Форма Служба доставки еды «Шеф»**(Рис.14). Основная форма приложения. При разработке интерфейса этой формы было использовано множество различны компонентов:

1. ListView – отображает список товароы в плиточном интерфейсе;
2. ComboBox – выпадающий списки, используемые для фильтрации;
3. ListBox – списки щитов и стоимости;
4. RatingBar – отображает рейтинг товра в виде звёздочек;
5. Buttons – кнопки для переход и достпуа к различным частям системы;
6. TextBlock – текстовые метки;
7. TextBox – поле для ввода текста;
8. Image – отображают изображения



Button

ComboBox

TextBlock

RatingBar

Image

TextBox

ListView

Рис.16 Служба доставки еды «Шеф»

**Форма Подробнее**. Интерфейс данной формы состоит из компонентов: Button – кнопка ОК, Grid – сетка для аккуратного размещения компонентов, Image – изображение, TextBlock – текстовые метки и materialDesign:RatingBar – отображение рейтинга в виде звездочек. (Рис.17)



Кнопка ОК

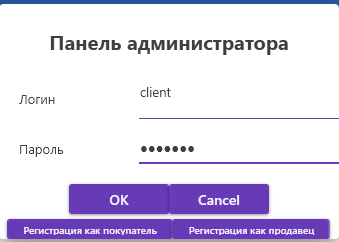
Image

materialDesign:RatingBar

Grid

Рис.17 Форма Подробнее

**Форма Авторизация**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК и Cancel, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода логина, PasswordBox – поле для ввода пароля и TextBlock – текстовые метки. (Рис.18)



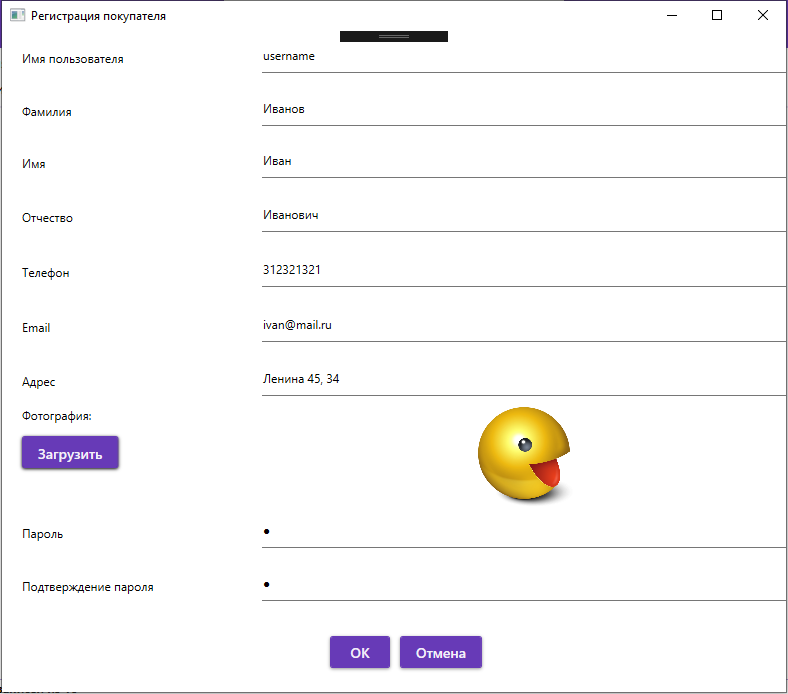
Кнопки ОК и Cancel

TextBox

PasswordBox

Рис.18 Форма Авторизация

**Форма Регистрация покупателя**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК, Загрузить и Отмена, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода данных, PasswordBox – поле для ввода пароля, Image – фотография покупателя и TextBlock – текстовые метки. (Рис.19)



Buttons

Grid

Image

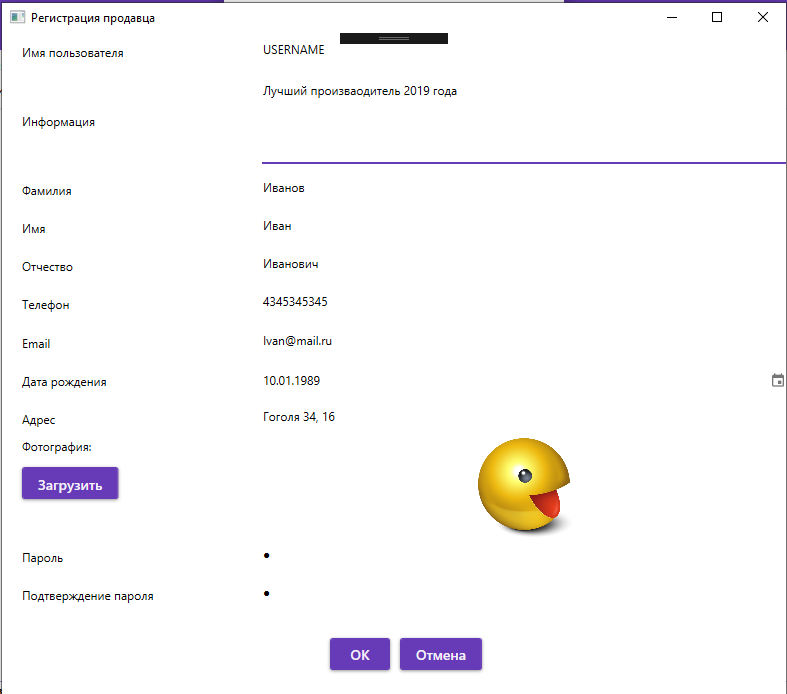
TextBlock

TexBox

PasswordBox

Рис.19 Форма Регистрация покупателя

**Форма Регистрация продавца**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК, Загрузить и Отмена, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода данных, PasswordBox – поле для ввода пароля, Image – фотография покупателя, DatePicker – поле выбора даты рождения и TextBlock – текстовые метки. (Рис.20)



Buttons

Grid

Image

TextBlock

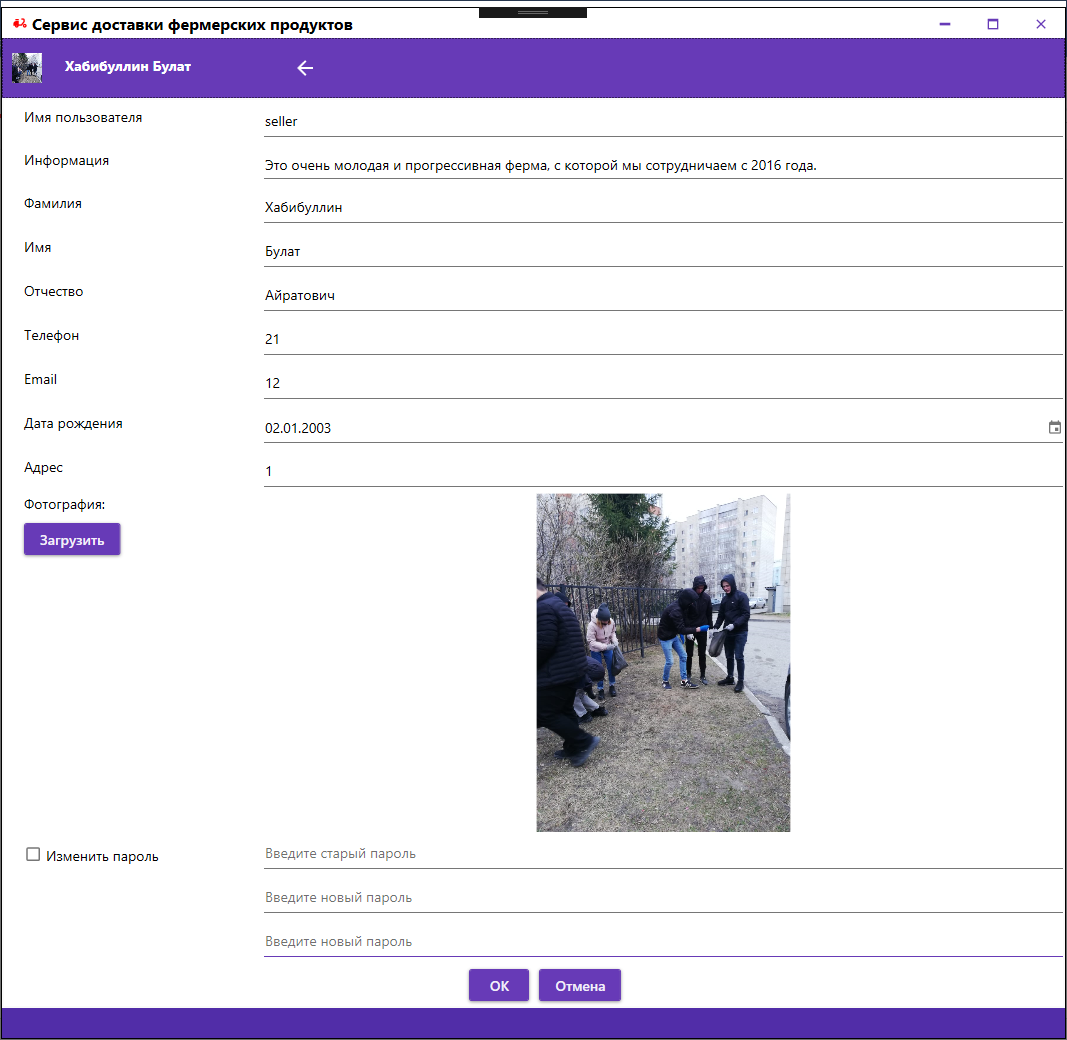
TexBox

PasswordBox

DatePicker

Рис.20 Форма Регистрация продавца

**Страница редактирование профиля продавца**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК, Загрузить и Отмена, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода данных, PasswordBox – поле для ввода пароля, Image – фотография покупателя, DatePicker – поле выбора даты рождения и TextBlock – текстовые метки. (Рис.21)



Buttons

Grid

Image

TextBlock

TexBox

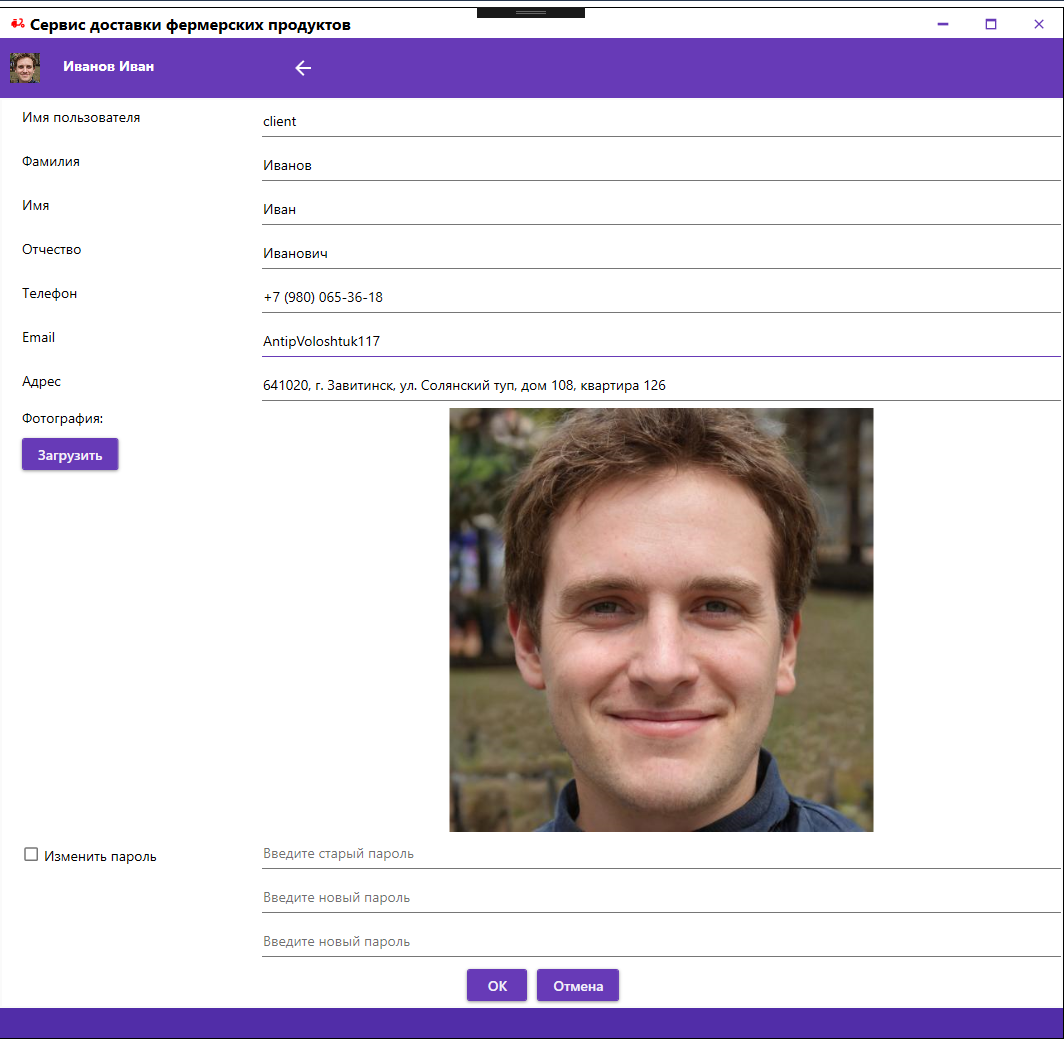
PasswordBox

DatePicker

CheckBox

Рис.21 Страница редактирование профиля продавца

**Страница редактирование профиля покупателя**. Интерфейс данной формы состоит из компонентов: Button – кнопки ОК, Загрузить и Отмена, Grid – сетка для аккуратного размещения компонентов, TextBox – поле для ввода данных, PasswordBox – поле для ввода пароля, Image – фотография покупателя, DatePicker – поле выбора даты рождения и TextBlock – текстовые метки. (Рис.22)



Buttons

Grid

Image

TextBlock

TexBox

PasswordBox

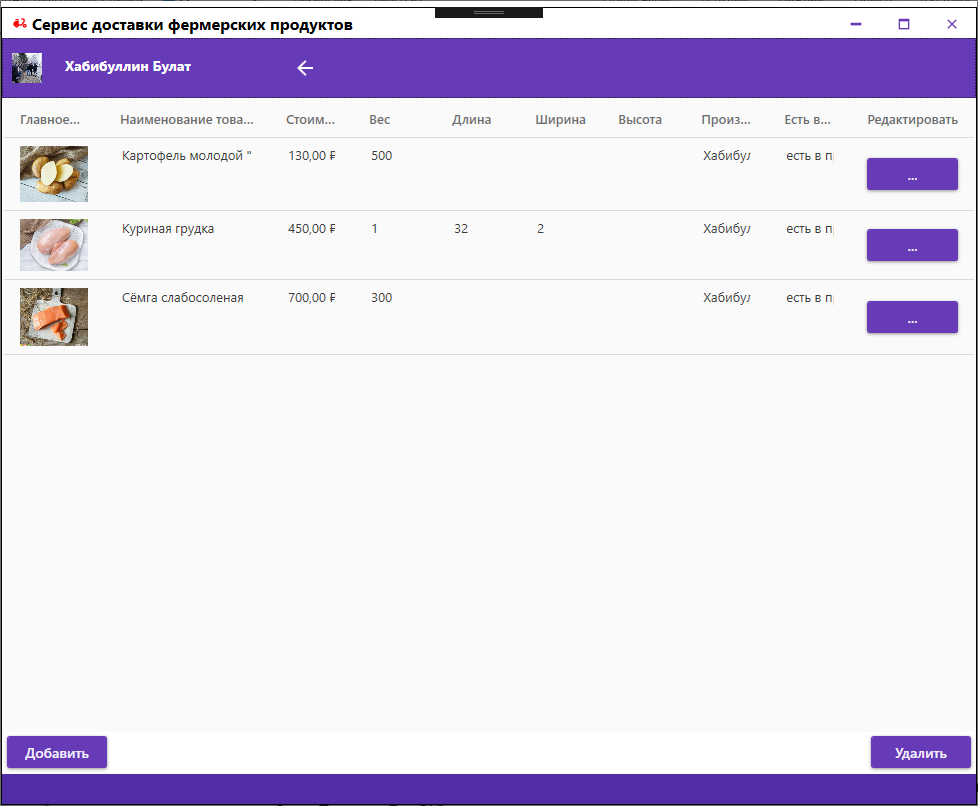
DatePicker

CheckBox

Рис.22 Страница редактирование профиля покупателя

**Страница товары(**Рис.23**)**. При разработке интерфейса этой формы были использованы компоненты:

1. DataGrid – отображение списка в виде таблицы;
2. Image – для отображения изображения;
3. Buttons – кнопки для добавления, редактирования и удаления записей;
4. TextBlock – текстовая метка;



Buttons

DataGrid

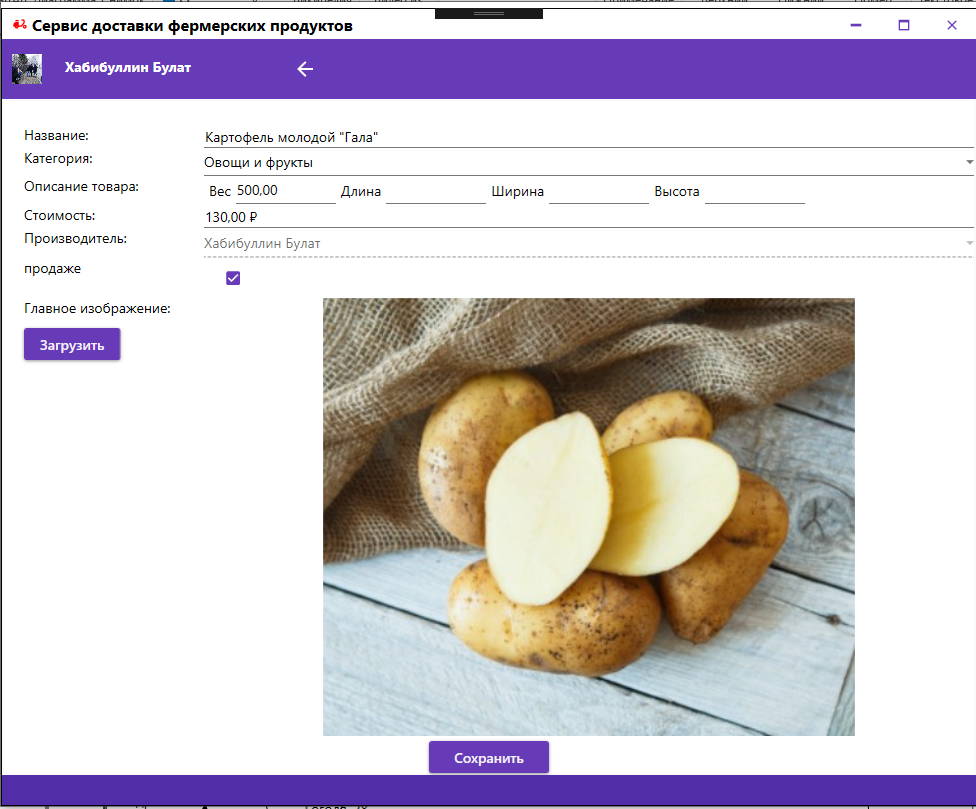
Image

TextBlock

Рис.23 Страница товары

**Страница добавления и редактирования(**Рис.24**)**. При разработке интерфейса этой формы были использованы компоненты:

1. ComboBox – выпадающий список групп товаров;
2. Buttons – кнопки Сохранения, Загрузки и перехода назад;
3. CheckComboBox – для установки логических полей;
4. TextBox – поля для ввода
5. Image – для отображения изображения;
6. TextBlock – текстовые метки;



Grid

Buttons

Image

TextBlock

materialDesign:RatingBar

CheckBox

ComboBox

TextBox

Button

Рис.24 Страница добавления и редактирования

**Страница Заказы**. Интерфейс данной формы состоит из компонентов: DataGrid, шести кнопок: Сохранить, Изменить, Удалить, Найти, Отмена поиска и вывод в Excel, combobox с выбором способа поиска, TextBox – поле для ключевых символов для поиска. (Рис.25)



Button – кнопки Изменения статуса заказа

Button – кнопки Найти, Отмена поиска и Вывод в Excel

Combobox – выбор способа поиска

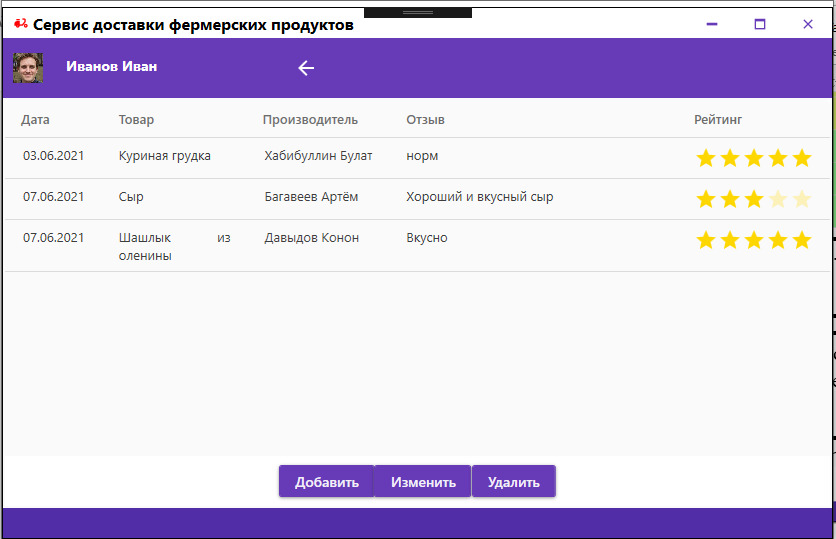
TextBox – ключ для поиска

Компонент DataGrid - таблица

Button - назад

Рис.25 Страница Заявки

**Страница Отзывы на товары**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.26)



Button – кнопки Добавить, Изменить и Удалить

Button – кнопка Назад

Компонент DataGrid - таблица

Рис.26 Страница Отзывы на товары

**Форма Добавления и редактирования отзыва**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.27)

Button – кнопки Ок и Отмена

TextBox

ComboBox

DateTimePicker

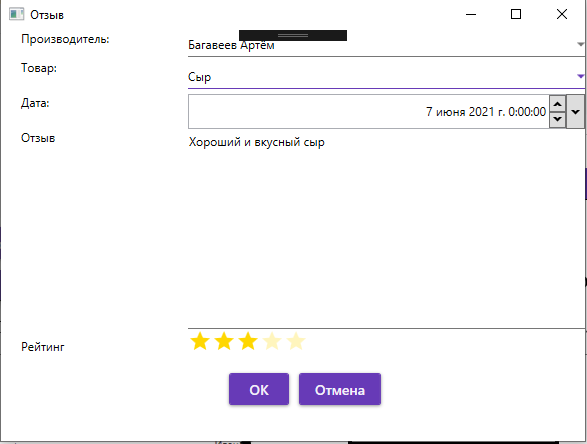
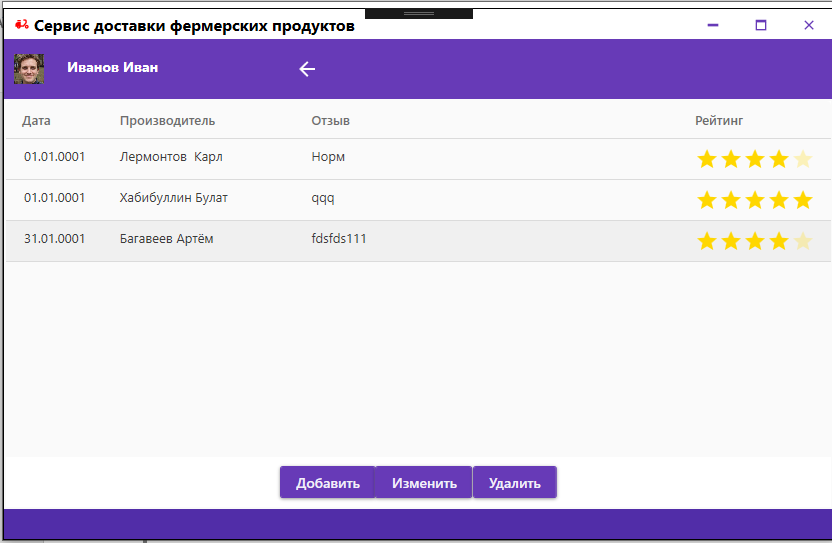


Рис.27 Форма добавления и редактирования отзыва

**Страница Отзывы на продавцов**. Интерфейс данной формы состоит из компонентов: DataGrid и кнопок: Добавить, Изменить, Удалить, Назад. (Рис.28)



Button – кнопки Добавить, Изменить и Удалить

Button – кнопка Назад

Компонент DataGrid - таблица

Рис.28 Страница Отзывы на продавцов

**Форма Добавления и редактирования отзыва**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода и кнопок Button. (Рис.29)

Button – кнопки Ок и Отмена

TextBox

ComboBox

DateTimePicker

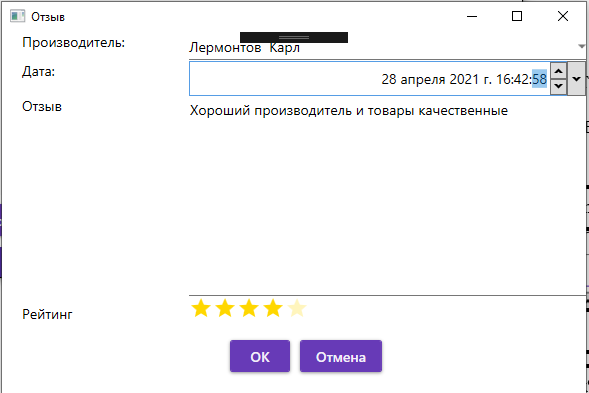
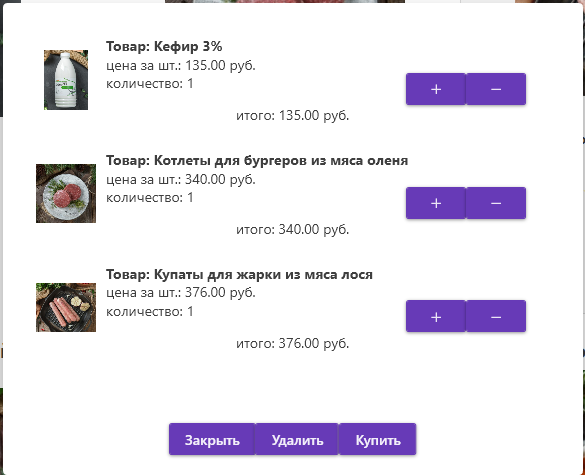


Рис.29 Форма добавления и редактирования отзыва

**Форма Корзина**. Интерфейс данной формы состоит из компонентов:Listview, TextBlock, Image и кнопок: Добавить, Удалить, Закрыть и Купить. (Рис.30)



Buttons

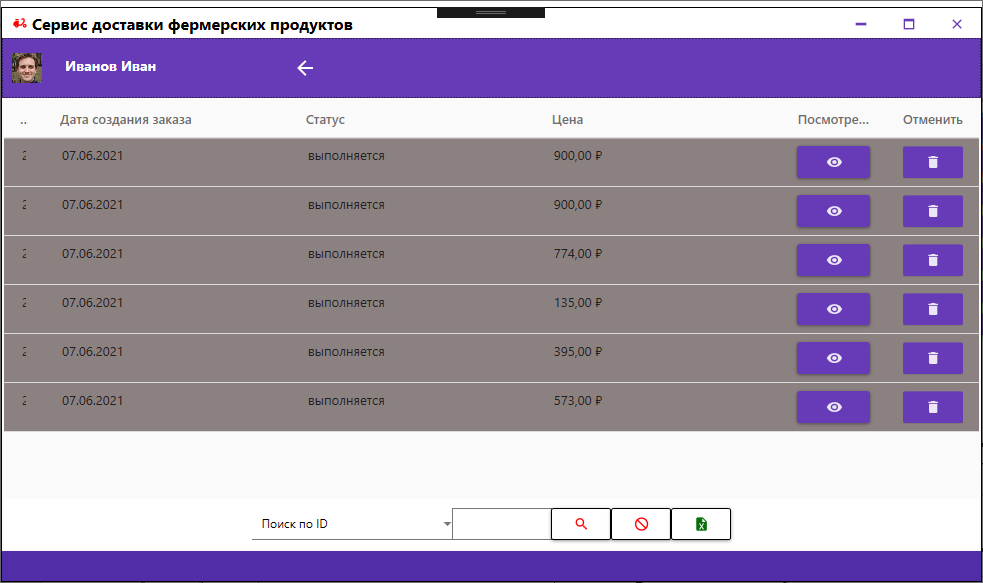
Image

TextBlock

Компонент ListView - Список

Рис.30 Форма Корзина

**Страница Заказы**. Интерфейс данной формы состоит из компонентов: DataGrid, шести кнопок: Поиск, Отмена поиска, Отменить заказ, Посмотреть заказ, и вывод в Excel, combobox с выбором способа поиска, TextBox – поле для ключевых символов для поиска. (Рис.31)



Button – кнопки Изменения статуса заказа

Button – кнопки Найти, Отмена поиска и Вывод в Excel

Combobox – выбор способа поиска

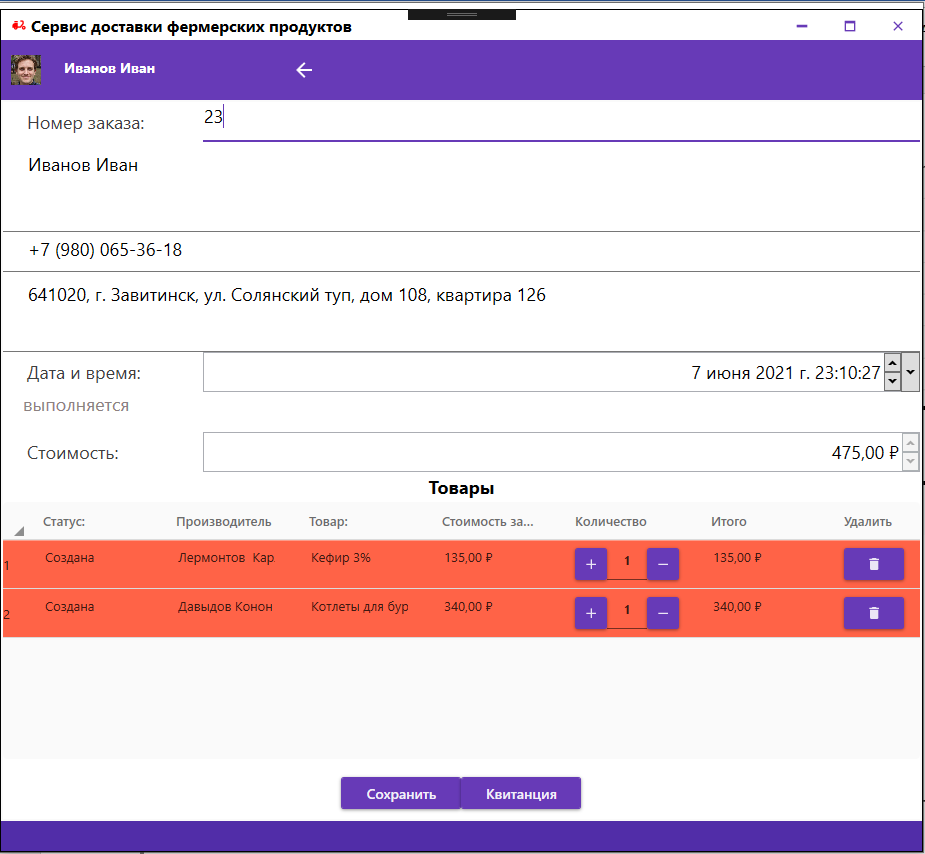
TextBox – ключ для поиска

Компонент DataGrid - таблица

Button - назад

Рис.31 Страница Заявки

**Окно добавления и редактирования Заказа**. Интерфейс данной формы состоит из компонентов: TextBox – текстовое поле ввода, DateTimePicker, DataGrid и кнопок Button. (Рис.32)



DateTimePicker

DoubleUpDown

Button – кнопки Сохранить и Квитанция, Удалить позицию

TextBox

Рис.32 Окно добавления и редактирования

**3.7 Описание структуры выходной информации**

Выходными данными являются:

1. Списки заказов покупателей(рис. 33).

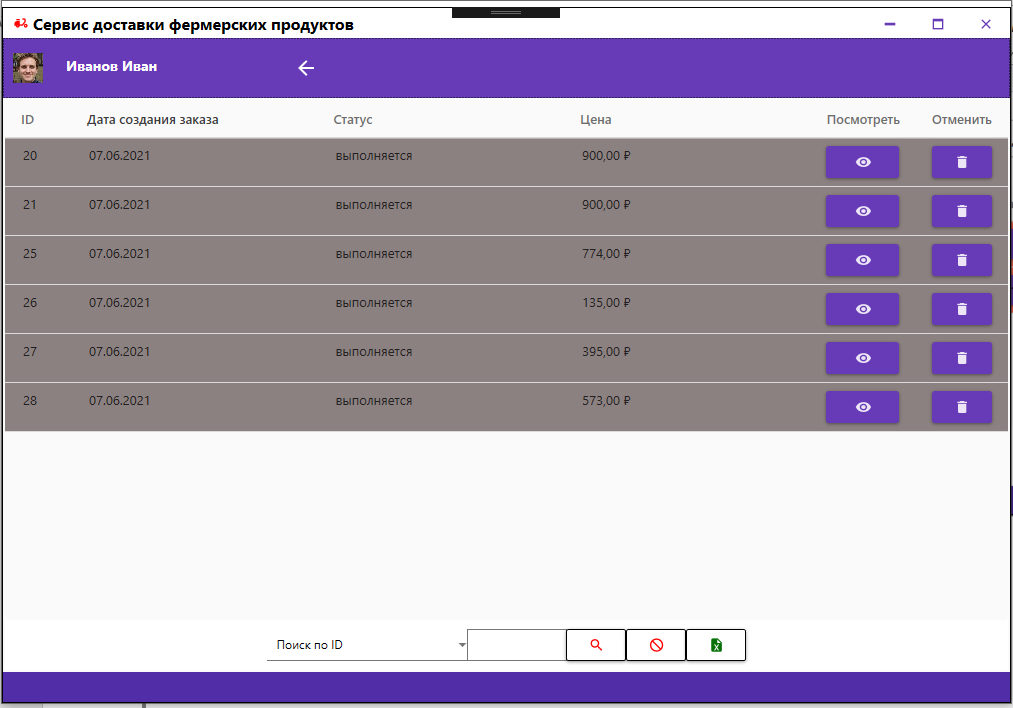


Рис.33 Список заказов покупателей

1. Список заказов продавцов(рис. 34).

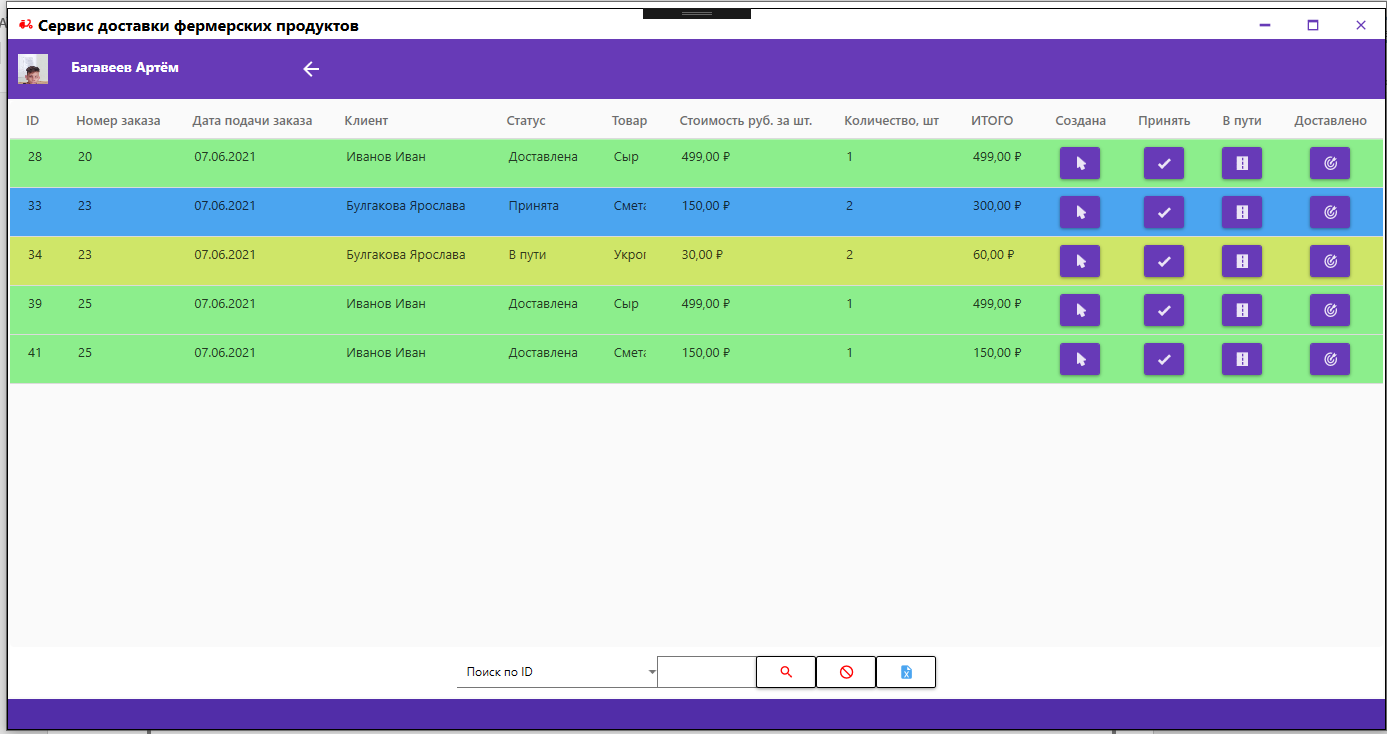


Рис.34 Список заказов продавцов

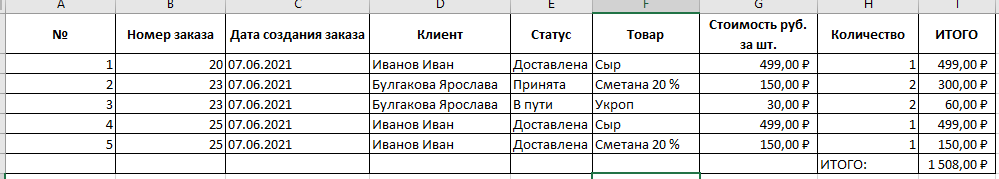


Рис.35 Список заказов продавцов

1. Квитанция(рис. 36).

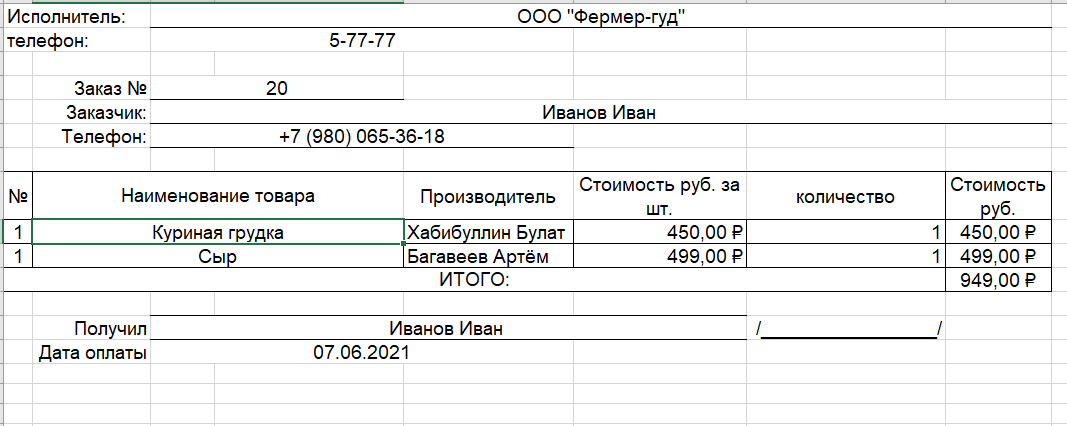


Рис.36 Квитанция

**4 РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**4.1 Назначение программного средства**

Назначением программного средства является контроль и упрощение доставки продутов питания и сопутствующих товаров.

**4.2 Условия выполнения программного средства**

Условия, при соблюдении которых возможно выполнение работы с программным продуктом:

-Компьютер с установленной на нём «ChefApp 1.0».

Подготовительные действия:

-На иконке «ChefApp 1.01» рабочего стола произвести двойной щелчок левой кнопкой мыши.

**4.3 Эксплуатация программного средства**

Этот раздел содержит обзор приложений «ChefApp 1.0» и ее основных параметров.

***Активация продукта***

Активировать копию программы ChefApp 1.0 можно сразу после её установки, когда появится окно подтверждения подлинности Вашей версии. Ключ активации расположен внутри упаковки программного продукта. Для успешного выполнения активации ключ активации необходимо вводить именно в том виде, в котором он предоставлен.

Вводные сведения об интерфейсе пользователя

После запуска приложения отображается стартовое окно приложения (Рис.37). Стартовое окно разделено на две части. В верхней части расположен блок фильтрации и поиска. Внизу в виде плиток отображается список товаров.

В правом верхнем углу расположена кнопка для авторизации.

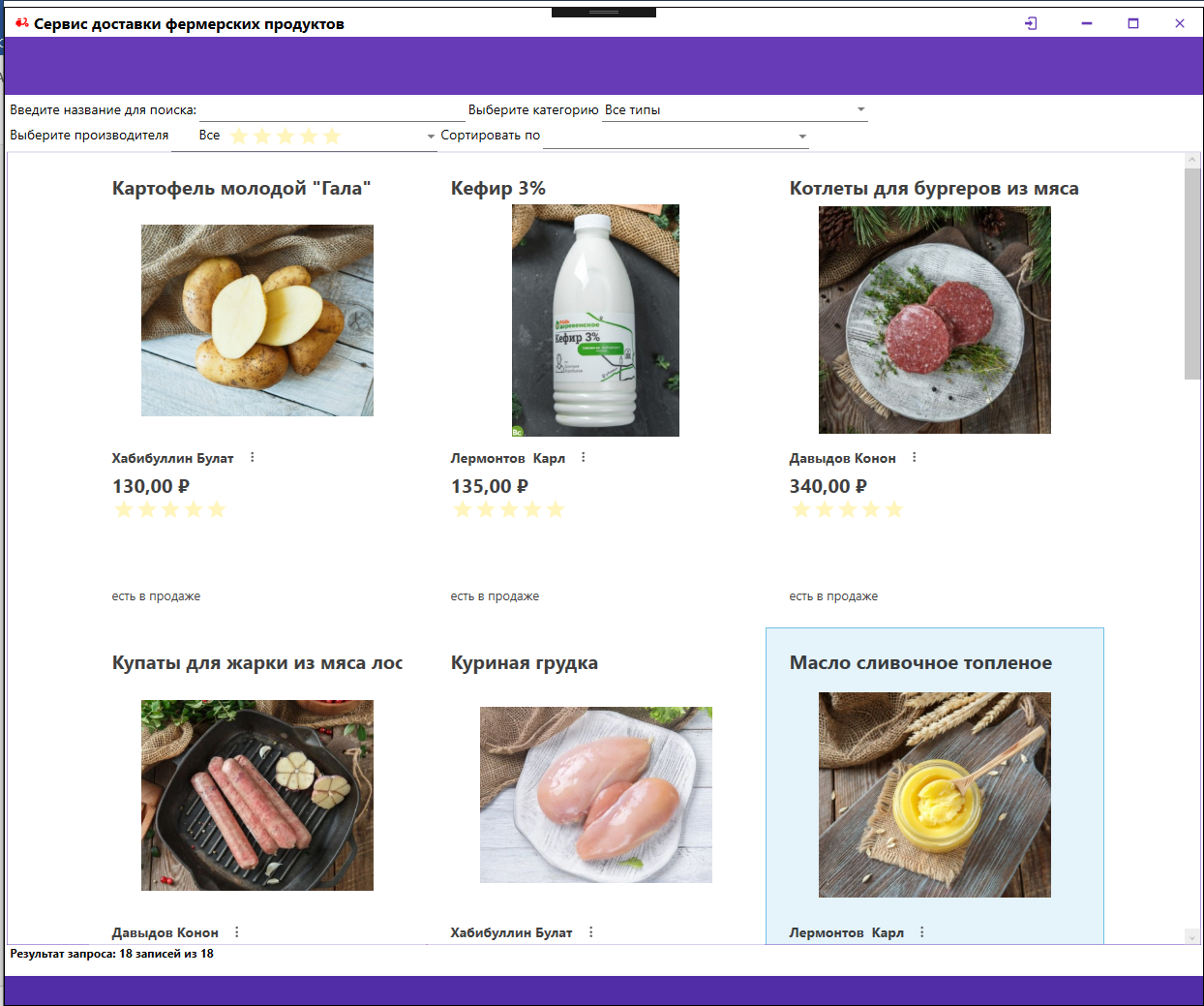


Рис.37 Окно Служба доставки еды «Шеф»

В списке товаров в виде карточек отображаются краткие сведения. В карточке имеются:

* - название товара;
* - изображение товара;
* - рейтинг
* - продавец(производитель товара);
* - кнопка для открытия окна Подробнее о производителе.

Кликнув по кнопке «Закрыть» программа потребует подтверждение. (Рис.38)

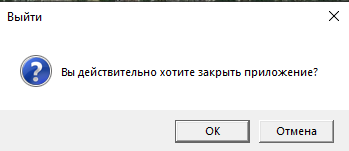


Рис.38 Сообщение программы

Для авторизации необходимо кликнуть на кнопку . Ввести логин и пароль пользователя в появившейся форме и нажать OK.

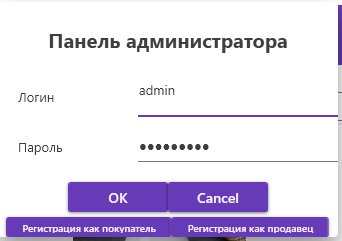


Рис.39 Форма входа в систему

Под заголовком появится панель кнопок для переход к другим страницам и частям приложениия.

Панель инструментов покупателя(Рис. 40) представляет собой пять кнопок(Переход к профилю, Корзина, Отзывы на товары, Отзывы на продавцов, Мои заказы), изображение - фото покупателя из профиля и фамилия имя.



Рис.40 Панель инструментов покупателя

Панель инструментов продавца(Рис. 41) представляет собой три кнопки (Переход к профилю, Товары и Мои заказы), изображение - фото продавца из профиля и фамилия имя.



Рис.41 Панель инструментов покупателя

Для того чтобы выйти из системы нажмите на кнопку , программа потребует подтверждения(Рис.42)

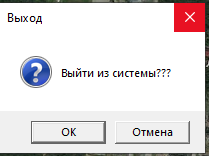


Рис.42 Сообщение программы

Если вы являетесь новым пользователем в системе, то необходимо нажать на кнопку Регистрация как покупатель или Регистрация как продавец. В появившейся форме требуется заполнить поля. При регистрации роль нового пользователя продавец или покупатель(Рис. 43).

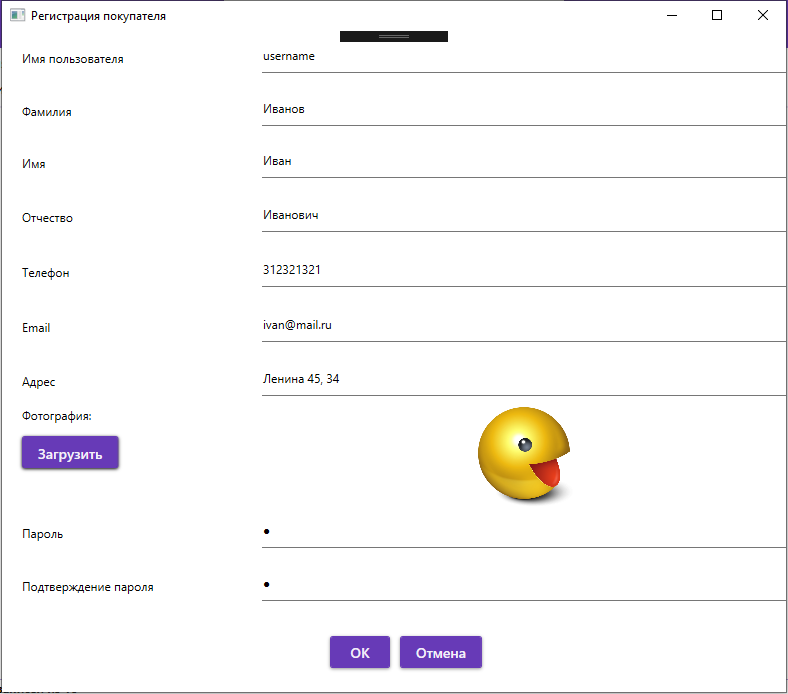


Рис.43 Форма регистрации покупателя

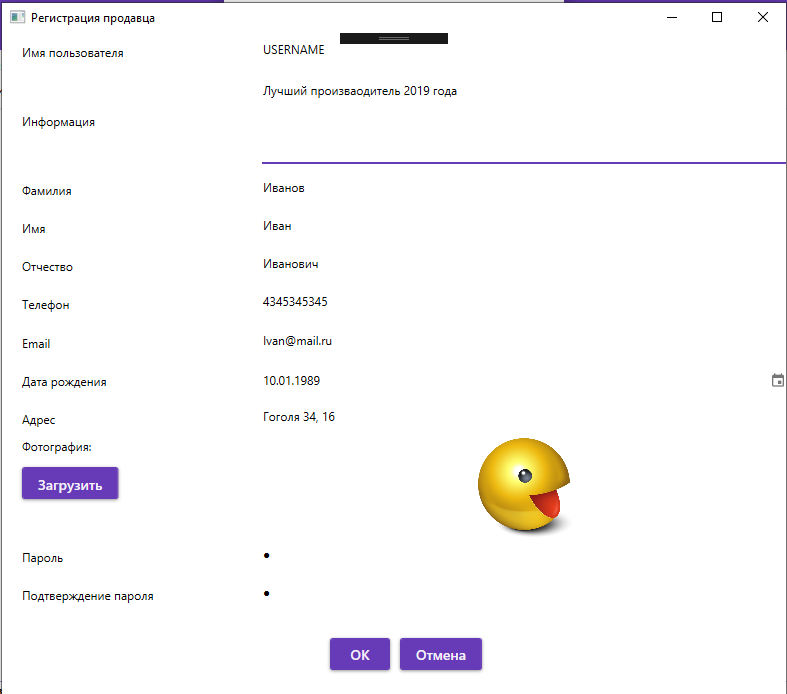


Рис.44 Форма регистрации продавца

Ниже описаны пункты меню управления и информации.

Форма Подробнее(Рис. 45). Появляется после нажатия на кнопку  в карточке товара. Это окно показывает сведения о производителе товара в новом окне.



Рис.45 Форма Подробнее

Страница профиль Продавца(Рис. 46). Появляется после нажатия на кнопку  на главной форме приложения. Это окно позволяет отредактировать информацию о продавце. При необходимости можно сменить пароль.

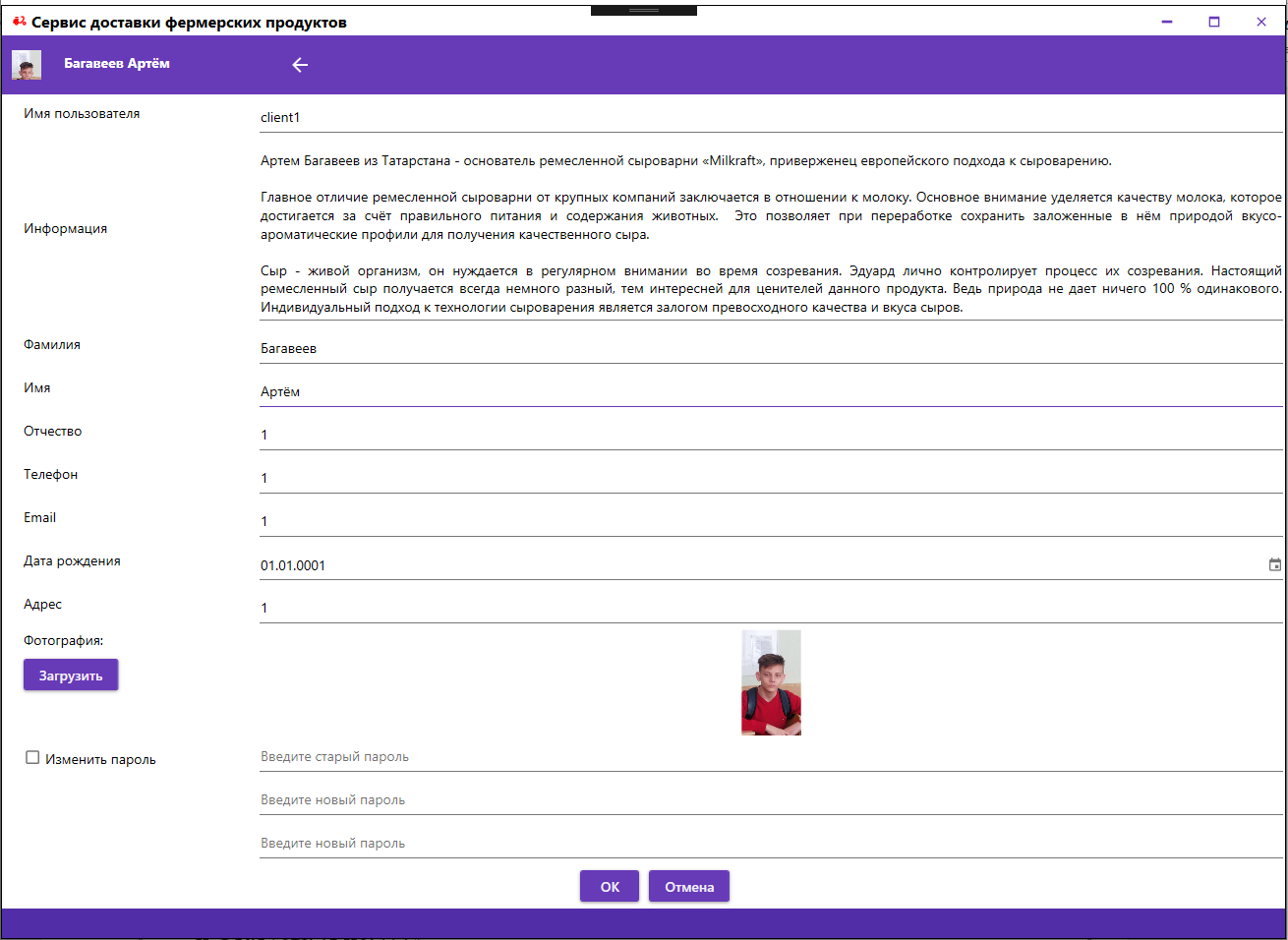


Рис.46 Страница профиль Продавца

Страница «Товары». Эта форма открывается при нажатии на кнопку  для пользователя с ролью продавец. Данная страница предназначена для отображения информации в табличном виде, открытия страницы редактирования и добавления. (Рис.47)

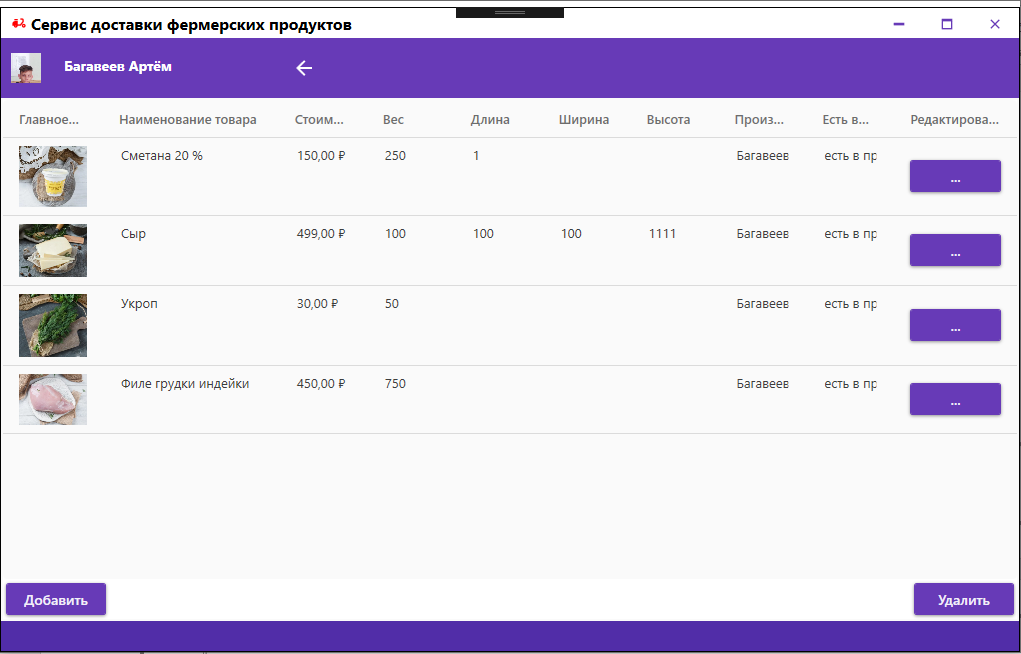


Рис.47 Страница Товары

Страница Добавление и редактирование. (Рис.48) Для открытия этой страницы нужно на странице товары выбрать запись и нажать на кнопку Редактировать или нажать на кнопку  выбранного товара на странице каталога. Также данную страницу можно открыть в режиме добавления, нажав на соответствующую кнопку на предыдущей странице. Эта страница предназначена для добавления или редактирования сведений о товаре.

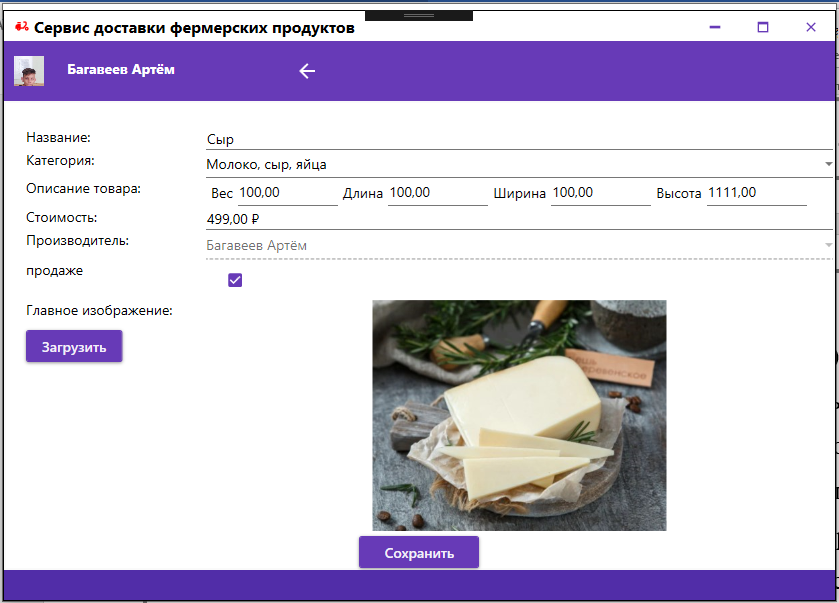


Рис.48 Страница Добавление и редактирование

«Заявки продавцов»: одна из основных страниц. На данной странице отображаются заказы на товары данного продавца.(Рис.49) Статусы заявок меняются на этой странице. Список товаров можно фильтровать по разным критериям, а также выводить данные в Excel

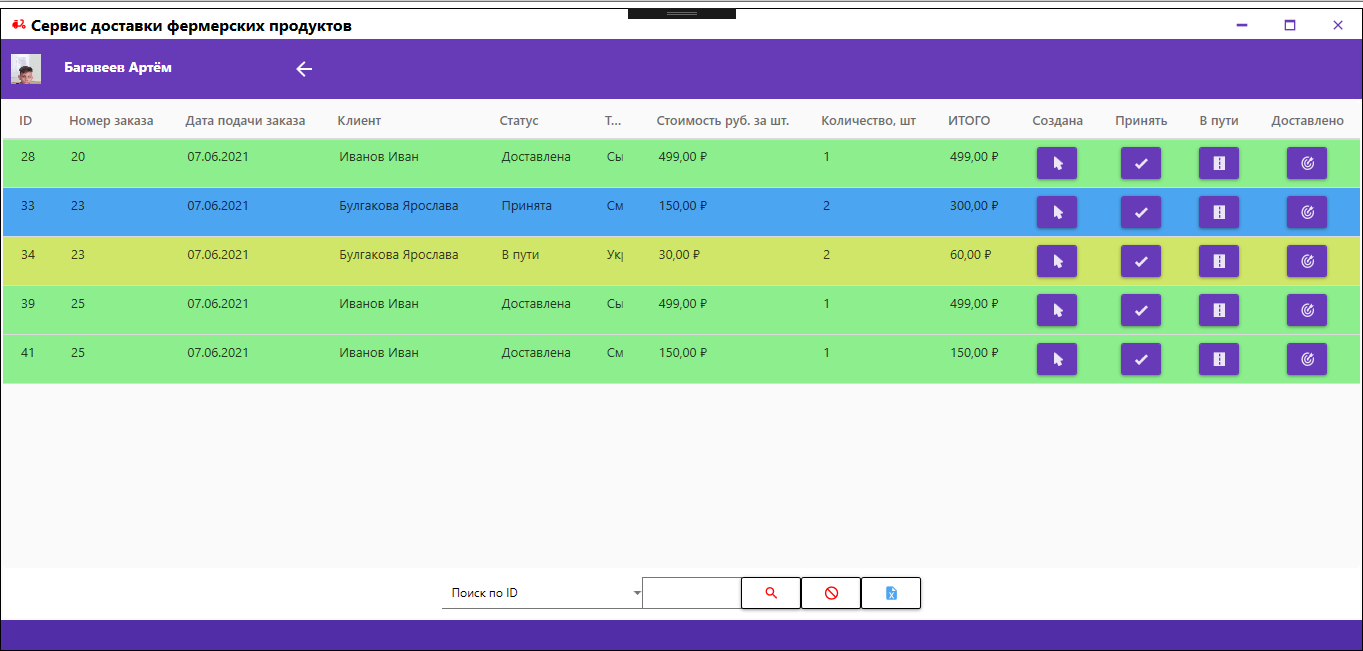


Рис.49 Страница Заявки

Страница профиль Покупателя(Рис. 50). Появляется после нажатия на кнопку  на главной форме приложения. Это окно позволяет отредактировать информацию о покупателе. При необходимости можно сменить пароль.

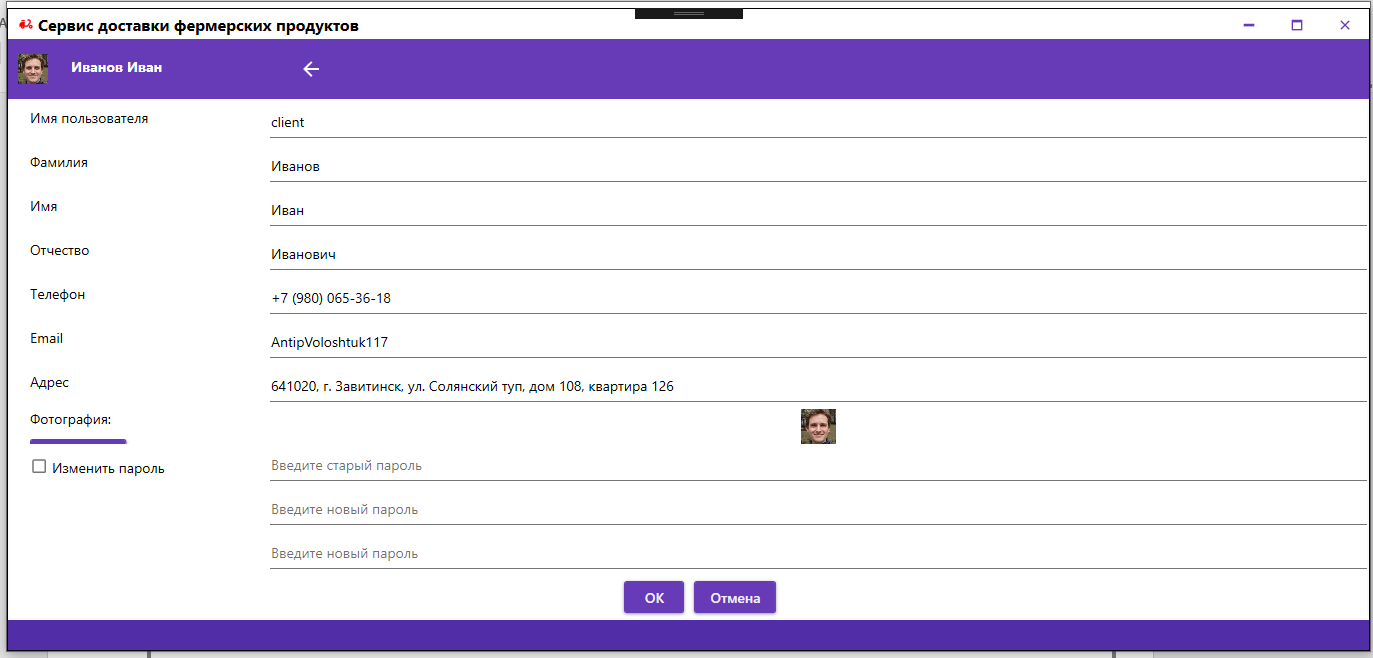


Рис.50 Страница профиль Покупателя

«Корзина»: эта форма предназначена для работы с корзиной товаров. Перед тем как добавить товар, покупатель на странице каталога выбирает товар и нажимает по кнопке  - в корзину. Элементы корзины можно редактировать, добавляя или удаляя товар. Для перехода к оформлению заказа покупателю нужно нажать на кнопку Купить. (Рис.51)

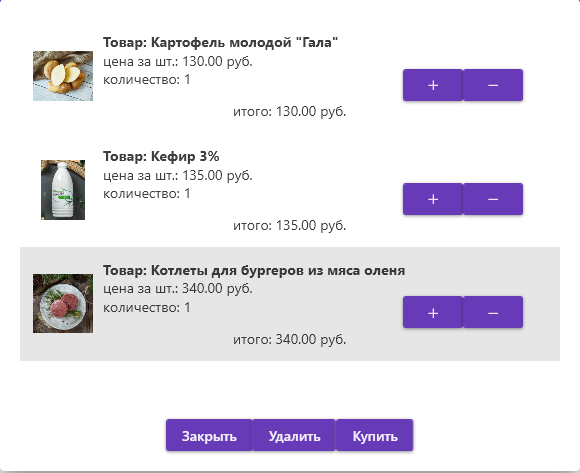


Рис.51 Корзина

«Заказ»: эта страница предназначена для оформления покупки и просмотра подробных сведений о заказе. Прежде чем сохранить товар, покупатель имеет возможность редактировать содержимое заказа. После нажатия на кнопку Сохранить пользователю разрешено отменить заказ и вывести квитанцию.(Рис.52)

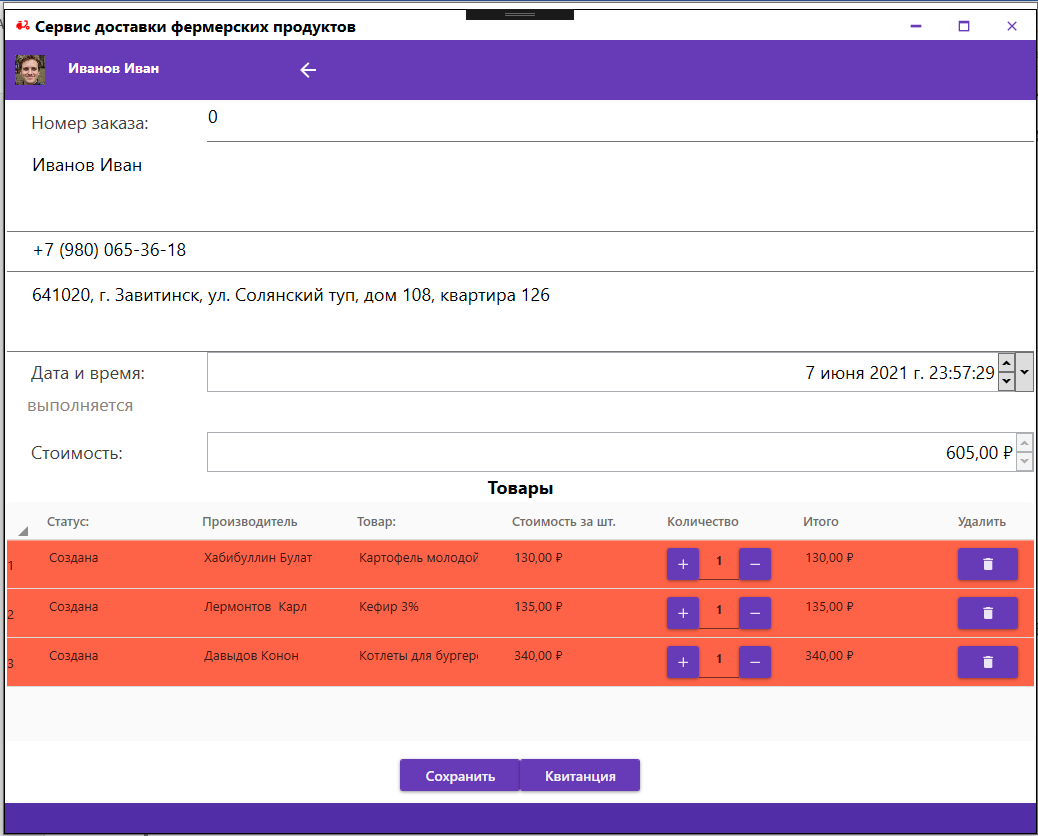


Рис.52 Страница Заказ

«Отзывы на товары»: эта страница предназначена для просмотра и удаления отзывов о товарах. Добавление и редактирование осуществляется через дополнительную форму. (Рис.53)

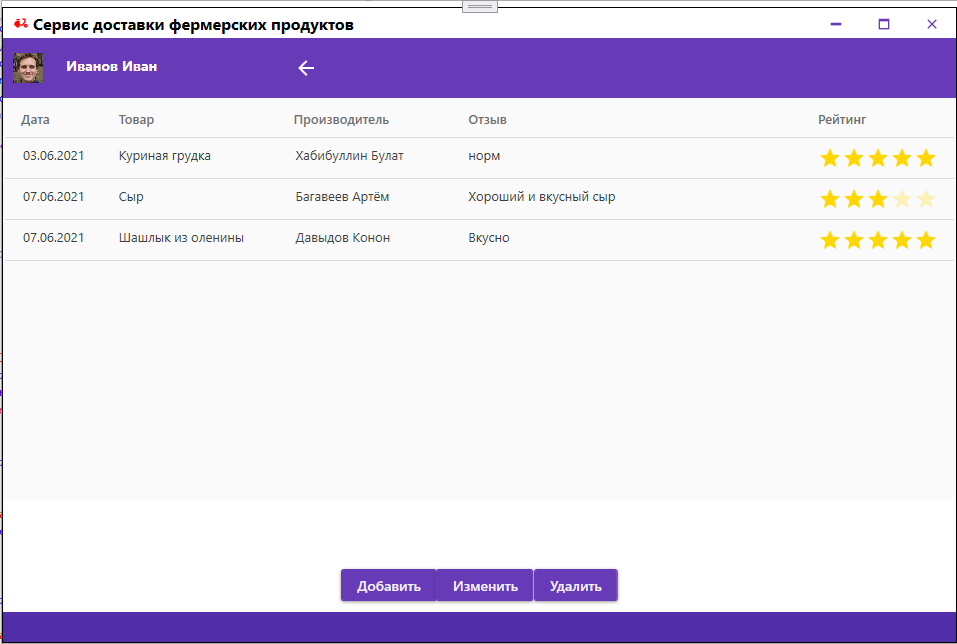


Рис.53 Страница Отзывы на товары

«Отзыв на товар»: эта форма предназначена для добавления и редактирования записи о городе. (Рис.54)

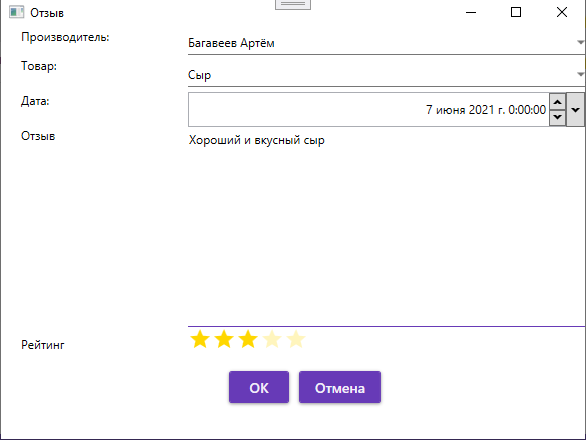


Рис.54 Форма Отзыв на товар

«Отзывы о продавцах»: эта страница предназначена для просмотра и удаления отзывов о продавцах. Добавление и редактирование осуществляется через дополнительную форму. (Рис.55)

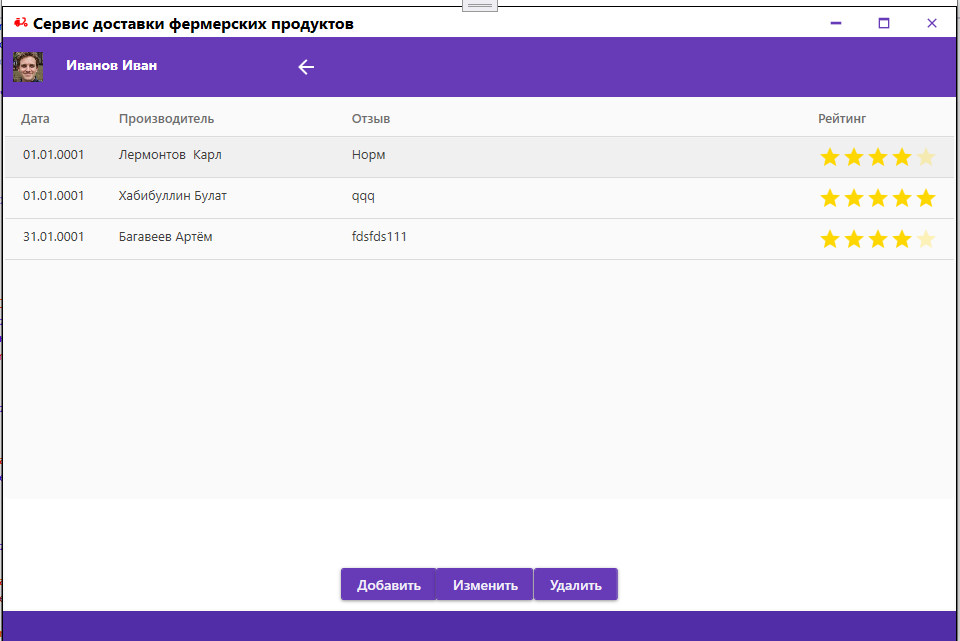


Рис.55 Страница Отзывы о продавцах

«Отзыв о продавце»: эта форма предназначена для добавления и редактирования записи о городе. (Рис.56)

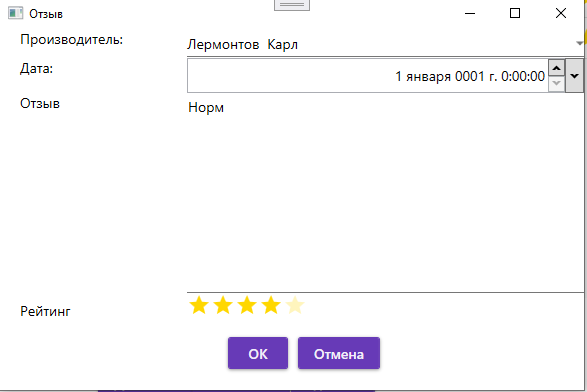


Рис.56 Форма отзыв о продавце

«Мои заказы»: одна из основных страниц. На данной странице отображаются заказы покупателя.(Рис.57) С этой страницы есть возможность перейти к информации о заказе. Список товаров можно фильтровать по разным критериям.

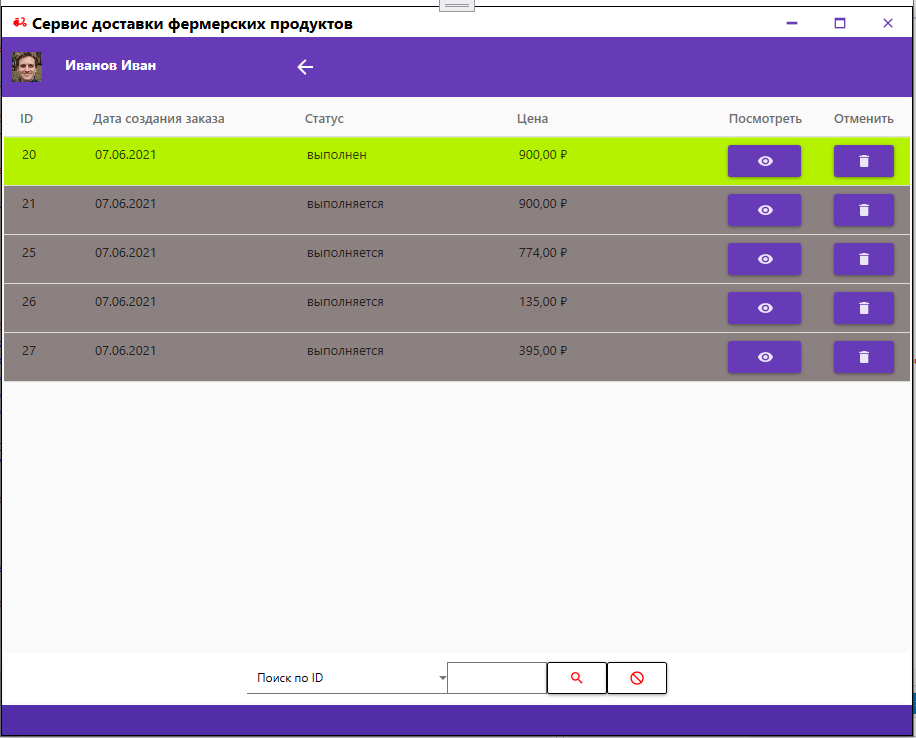


Рис.57 Страница Мои заказы

Работа с ChefApp 1.01

После запуска программы Вы перейдете на главную форму программы. В формах «Товары», «Профили», «Заказы», «Каталог» заполняем все необходимые для работы пункты.

После этого кликаем по кнопке «Добавить запись».

Если Вы желаете удалить любую введенную ранее информацию, Вам необходимо кликнуть по нужной кнопке.

Для того чтобы изменить запись, введите в редактируемые поля данные и нажмите кнопку сохранить. Данные формы редактированию и управлению со стороны пользователя не подлежат.

После того, как Вы закончите сеанс работы с программой, если Вы хотите выйти из ChefApp 1.01 кликните на соответствующей кнопке «Выход».

Основные кнопки ChefApp 1.01

Кнопки управления «Добавить запись», «Удалить запись», «Сохранить» расположены на каждой конкретной форме в базе данных.

Действия этих кнопок приведены в таблице 1:

Таблица 1 – Действие кнопок

|  |  |
| --- | --- |
| Добавить запись | После нажатия на эту кнопку выйдет вспомогательная форма для добавления записей |
| Удалить запись | Кнопка «Удалить запись» позволяет удалить выделенную запись. |
| Сохранить данные | Кнопка «Сохранить данные» добавляет запись введенную пользователем в БД, либо обновляет существующую. |
| Редактировать | Открывает форму в режиме редактирования информации |
|  | Позволяет Загрузить файл |
|  | Переход на предыдущую страницу |
|  | Выгрузить информацию в Excel |
|  | Показать сведения о продавце в дополнительном окне |
|  | Поместить выбранный товар в корзину |

**4.4 Сообщения пользователю**

Для уверенной и комфортной работы пользователю важно понимать поведение программы, а программе, в свою очередь, поведение пользователя. Для информирования и уточнения намерений пользователей в программах используются сообщения.

При попытке удалить запись, пользователю выводится сообщение с запросом на подтверждение или отмену действия. (Рис.58)

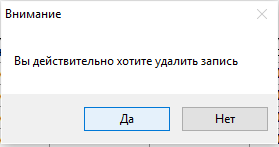


Рис.58 Сообщение Подтверждение удаления

После добавления новой записи, пользователю выводится сообщение. (Рис.59)

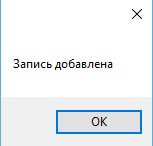


Рис.59 Сообщение Запись добавлена

После обновления записи, пользователю выводится сообщение. (Рис.60)

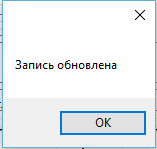


Рис.60 Сообщение Запись обновлена

При попытке удалить связанные записи, программа выводит пользователю сообщение о невозможности действия. (Рис.61)

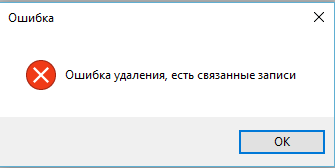


Рис.61 Сообщение Ошибка удаления

**5 РАЗРАБОТКА ТЕСТОВОГО НАБОРА**

**5.1 Определение верхней границы количества тестов**

Для разработанного алгоритма был построен граф.

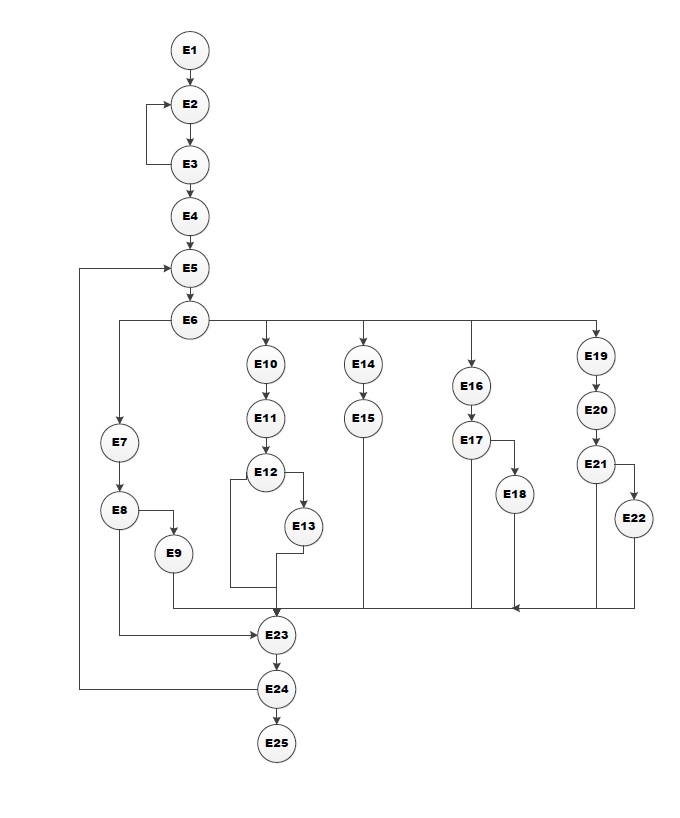
****

Рис.62 Потоковый граф

Необходимое количество тестов определяется по формуле V(G) = E-N+2, где E – это число ребер, а N–это число вершин данного графа. Согласно представленному графу E=34, N=25. Далее по формуле рассчитаем необходимое количество тестовых сценариев (тест-кейсов).

V(G)=34-25+2=11.

Для данного программного средства необходимо разработать не менее 11 тестовых сценариев.

Чтобы определить количество тестовых прогонов для программного продукта ChefApp 1.0 необходимо выделить требования. Из требований к приложению выделим поддержку 2-х операционных систем с одним основным языком локализации и выполнения 6-ти основных функций. Кроме того, приложение позволяет выполнять 5 функций по взаимодействию с системой (запуск приложения, выход из приложения, сохранение результатов в файл, работа с буфером и т.п.).

Таким образом, полное покрытие требований задаёт набор из 2\*1\*(6+5) =22 тестовых прогонов.

**5.2 Разработка тестовых вариантов**

Таблица 2 – Тестирование корректного добавления информации в Статусы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Статусы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Статус = создана  Цвет = Зелёный | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Статусы открыта | | Отображается форма Статусы |  |

Таблица 3 – Тестирование некорректного добавления информации в таблицу Статусы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Статусы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Статус =  Цвет = Зелёный | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, так как поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Статусы открыта | | Отображается форма Статусы |  |

Таблица 4 – Тестирование корректного редактирования информации в таблице Статусы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Статусы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Статус = создана  Цвет = Зелёный | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Статусы  открыта | | Отображается форма Статусы |  |

Таблица 5 – Тестирование некорректного редактирования информации в таблице Статусы

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Статусы | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Статусы | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Статус =  Цвет = Зелёный | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Статусы  открыта | | Отображается форма Статусы |  |

Таблица 6 – Тестирование корректного добавления информации в таблицу Товар

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Товар | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Товар | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фото= загружено  Название = Сметана 20 %  Категория = Молоко, сыр, яйца  Вес = 250 г  Стоимость = 150 р.  Производитель = Багавеев Артём  В продаже = Да | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Товар открыта | | Отображается форма Товар |  |

Таблица 7 – Тестирование некорректного добавления информации в таблицу Товар

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Товар | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Товар | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Фото=  Название =  Категория = Молоко, сыр, яйца  Вес = 250 г  Стоимость = 150 р.  Производитель = Багавеев Артём  В продаже = Да | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Товар  открыта | | Отображается форма Товар |  |

Таблица 8 – Тестирование корректного редактирования информации в таблице Товар

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Товар | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Товар | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Фото= загружено  Название = Сметана 20 %  Категория = Молоко, сыр, яйца  Вес = 250 г  Стоимость = 150 р.  Производитель = Багавеев Артём  В продаже = Да | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Товар открыта | | Отображается форма Товар |  |

Таблица 9 – Тестирование некорректного редактирования информации в таблице Товар

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Товар | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Товар | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Фото=  Название =  Категория = Молоко, сыр, яйца  Вес = 250 г  Стоимость = 150 р.  Производитель = Багавеев Артём  В продаже = Да | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Товар открыта | | Отображается форма Товар |  |

Таблица 10 – Тестирование корректного добавления информации в Категории товаров

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного добавления информации в таблицу Категории товаров | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категории товаров | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Категории товаров= Молоко, сыр, яйца | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категории товаров открыта | | Отображается форма Категории товаров |  |

Таблица 11 – Тестирование некорректного добавления информации в таблицу Категории товаров

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного добавления информации в таблицу Категории товаров | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категории товаров | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Нажмите кнопку добавить  Заполните форму  Категории товаров= | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, так как поле пустое | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категории товаров открыта | | Отображается форма Категории товаров |  |

Таблица 12 – Тестирование корректного редактирования информации в таблице Категории товаров

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование корректного редактирования информации в таблице Категории товаров | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категории товаров | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Категории товаров= Молоко, сыр, яйца | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Данные успешно сохранены | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категории товаров открыта | | Отображается форма Категории товаров |  |

Таблица 13 – Тестирование некорректного редактирования информации в таблице Категории товаров

|  |  |  |  |
| --- | --- | --- | --- |
| **Название:** | Тестирование некорректного редактирования информации в таблице Категории товаров | | |
| **Функция:** | Циклы | | |
| **Действие** | | **Ожидаемый результат** | **Результат теста:**   * пройден * провален * заблокирован |
| **Предусловие:** | |  |  |
| Открываем форму Категории товаров | | Форма открыта |  |
| **Шаги теста:** | |  |  |
| Выберите запись для редактирования и измените данные,  Категории товаров= | | Данные успешно введены |  |
| Нажмите кнопку «Сохранить» | | Кнопка «Сохранить» заблокирована, поля пустые | **Тест пройден** |
| **Постусловие:** | |  |  |
| Форма Категории товаров открыта | | Отображается форма Категории товаров |  |

**5.3 Составление отчета о тестировании**

Для форм проекта были созданы и проведены тесты, результаты сведены в таблицу:

Таблица 14 – Результаты тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Прохождение | | |
| Форма | Число ТП | пройдено | | не пройдено |
| Главная | 4 | 4 | | 0 |
| Товары | 4 | 4 | | 0 |
| Статусы | 4 | 4 | | 0 |
| Категории товаров | 4 | 4 | | 0 |
| Профиль продавца | 4 | 4 | | 0 |
| Заказы | 4 | 4 | | 0 |
| Профиль покупателя | 4 | 4 | | 0 |
| Отзыв на товар | 4 | 4 | | 0 |
| Отзыв на продавца | 4 | 4 | | 0 |
| ИТОГО | 36 | 36 | 0 | |

По результатам тестирования были выявлены и устранены следующие проблемы:

При выполнении теста таблица №8. Тестирование формы «Товар», Некорректный запрос

**Ошибка**

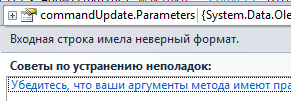


Рис.68 Ошибка теста №6

**Причина**

Поле вес имеет тип вещественное число, было введено строковое значение.

**Принятые меры**

Ошибка устранена. В поле широта можно ввести только цифры и запятую

**ЗАКЛЮЧЕНИЕ**

**Оценка качества программного средства с помощью метрик**

Оценка качества программного средства проводится исходя из стандарта оценки качества ISО 9126.

На компьютере следующей конфигурации:

Установленная операционная система – Windows 10 Professional.

Процессор: Intel Pentium G2030 3.0 GHz.

ОЗУ: 4 Gb

Таблица 15 – Требования к количественным характеристикам качества программного средства

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Надежность** |  | | |
| *Завершенность:*  *-* наработка на отказ при отсутствии рестарта; | Часы | 10 | 10 |
| *Устойчивость:*  - относительные ресурсы на обеспечение надежности и рестарта. | % | 0,5 | 0,5 |
| *Восстанавливаемость:*  -длительность восстановления. | Минуты | 2 | 2 |
| *Доступность-готовность:*  -относительное время работоспособного функционирования. | Вероятность | 0,998 | 0,998 |
| **Эффективность** |  | | |
| *Временная эффективность:*  *-*время отклика - получение результатов на типовое задание;  - пропускная способность число типовых заданий, исполняемых в единицу времени. | Секунды  Число в минуту | 3  60 | 3  60 |
| *Используемость ресурсов:*  *-* относительная величина использования ресурсов ЭВМ при нормальном функционировании программного средства. |  | 0,001 | 0,001 |

Таблица 16 – Требования к качественным характеристикам программного средства.

|  |  |  |  |
| --- | --- | --- | --- |
| **Характеристики качества** | **Мера** | **Требуемое значение** | **Реальное значение** |
| **Практичность** |  | | |
| *Простота использования:*  **-**среднее время ввода заданий;  -среднее время отклика на задание. | Секунды  Секунды | 10  3 | 10  3 |
| *Изучаемость :*  - трудоемкость изучения применения ПС;  -продолжительность изучения;  -объем эксплуатационной документации; | Чел.-часы  Часы  Страницы | 5  2  15 | 5  2  15 |
| **Сопровождаемость** |  | | |
| *Изменяемость:*  -трудоемкость подготовки изменений;  -длительность подготовки изменений. | Чел.-часы  Часы | 5  5 | 5  5 |
| *Тестируемость:*  *-трудоемкость тестирования изменений;*  *- длительность тестирования изменений.* | Чел.-часы  Часы | 3  3 | 3  3 |
| **Мобильность** |  | | |
| *Адаптируемость:*  - трудоемкость адаптации;  -длительность адаптации. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Простота установки:*  -трудоемкость инсталляции;  -длительность инсталляции. | Чел.-часы  Часы | 0,2  1 | 0,2  1 |
| *Замещаемость:*  -трудоемкость замены компонентов;  - длительность замены компонентов. | Чел.-часы  Часы | 0,2  0,2 | 0,2  0,2 |

Таблица 17 – Сложность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Тип объекта | Количество | Вес | | | Итого |
|  |  | Простой | Средний | Сложный |  |
| Экран | 17 | x1 | x2 | **x3** | =51 |
| Отчет | 3 | x2 | x5 | **x8** | =24 |

Таблица 18 – Сложность экрана

|  |  |  |  |
| --- | --- | --- | --- |
| Экраны | Количество клиентских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | **Всего >8** |
|  | (<2 срв, <3 клт) | (2-3 срв, 3-5 клт) | (>3 срв, >5 клт) |
| <3 | Простой | Простой | Средний |
| 3-7 | Простой | Средний | Сложный |
| **>8** | Средний | Сложный | **Сложный** |

Таблица 19 – Сложность отчета

|  |  |  |  |
| --- | --- | --- | --- |
| Отчеты | Количество Возвратских (клт) таблиц данных | | |
| Количество представлений | Всего <4 | Всего <8 | Всего >8 |
|  | (<2 срв, <3 клт) | (2-3 срв, 3-5 клт) | **(>3 срв, >5 клт)** |
| 0 или 1 | Простой | Простой | Средний |
| **2 или 3** | Простой | Средний | **Сложный** |
| >4 | Средний | Сложный | Сложный |

Производительность = 5000 тысяч строк кода/1 месяца=5000 тысяч строк кода/месяц

Качество = 25 ошибок /5 тысяч строк кода=5 ошибок/тысяч строк кода

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Г. Шилдт C#: Учебный курс. - Спб.: Питер, 2019. - 512с

2. Либерти, Д. Программирование на Си Шарп. - М.: Символ-плюс, 2018. - 684 с.

3. Ватсон К. Си Шарп/К.Ватсон.-М.:Лори, 2019.-862 с

4. Балена Ф., Димауро Д. Современная практика программирования на Visual Basic и Visual C#.-М.:Русская редакция,2019,604с.

5. Енин А., Енин Н. Локальная СУБД своими руками. Учимся на примерах.-М.:СОЛОН-ПРЕСС,2019,464с.,СД

6. Лабор В.В. Си Шарп: СоВозврат приложений для Windows/-Мн.: Харвест, 2018-384с.

7. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2019-512с.

8. ГОСТ 19.201-78.ЕСПД. Техническое задание. Требования к содержанию и оформлению

9. ГОСТ 19.102-77.ЕСПД. Стадии разработки

10. ГОСТ 19.505-79.ЕСПД. Руководство оператора. Требования к содержанию и оформлению

11. ГОСТ 19.002-80.ЕСПД. Схемы алгоритмов и программ. Правила выполнения

12. ГОСТ 19.404-79.ЕСПД. Пояснительная записка. Требования к содержанию и оформлению

13. Владимир Биллинг. Основы программирования на С#. 2018 г.

[Электронный ресурс http://mexalib.com/view/42671]

14. Никита Культин. Основы программирования в MicrosoftVisualC# 2019. 2019 г.

[Электронный ресурс http://www.litres.ru/nikita-kultin/osnovy-programmirovaniya-v-microsoft-visual-c-2010-2]

15. Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#.-М.:КУДИЦ-ОБРАЗ, 2019.

[Электронный ресурс http://www.twirpx.com/file/27852]

16. Йен Гриффитс. Программирование на С#. 2018 г.

[Электронный ресурс http://rutracker.org/forum/viewtopic.php?t=4772853]

**Приложение А**

«Служба доставки еды «Шеф»»

Текст кода программного средства

460.ЗМК.00008-01 12 01

**Текст кода программного средства**

<Application x:Class="ChefApp.App"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:local="clr-namespace:ChefApp"

StartupUri="Windows/MainWindow.xaml">

<Application.Resources>

<ResourceDictionary>

<ResourceDictionary.MergedDictionaries>

<!--<ResourceDictionary Source="Dictionary1.xaml"/>-->

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Light.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Defaults.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Button.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.RadioButton.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.Menu.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.ToolTip.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignThemes.Wpf;component/Themes/MaterialDesignTheme.PopupBox.xaml" />

<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Primary/MaterialDesignColor.DeepPurple.xaml" />

<!--<ResourceDictionary Source="pack://application:,,,/MaterialDesignColors;component/Themes/Recommended/Accent/MaterialDesignColor.LightBlue.xaml" />-->

</ResourceDictionary.MergedDictionaries>

</ResourceDictionary>

</Application.Resources>

</Application>

using MaterialDesignThemes.Wpf;

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Controls;

namespace ChefApp.Models

{

public class Manager

{ public static Frame MainFrame { get; set; }

public static TextBlock TbCount { get; set; }

public static Badged BadgeCount { get; set; }

public static Client currentClient { get; set; }

public static Developer currentDeveloper { get; set; }

public struct BuyItem

{ public int Count { get; set; }

public double Total { get; set; }

public string StatusName { get; set; }

public string StatusColor { get; set; }

}

public class ItemOf

{ public Good Good { get; set; }

public BuyItem BuyItem { get; set; }

}

public static Dictionary<Good, BuyItem> buyGoods = new Dictionary<Good, BuyItem>();

}}

<Window x:Class="ChefApp.MainWindow"

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:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"

xmlns:local="clr-namespace:ChefApp"

mc:Ignorable="d"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

Title="" Height="800" Width="1080"

WindowStartupLocation="CenterScreen" Closing="Window\_Closing"

BorderThickness="1" WindowStyle="None" ContentRendered="Window\_ContentRendered" >

<Grid> <Grid.RowDefinitions>

<RowDefinition Height="30"/>

<RowDefinition Height="60"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions> <Rectangle Grid.Row="0" Grid.Column="0" Fill="White"/> <StackPanel Orientation="Horizontal" Grid.Column="1" Grid.Row="0" HorizontalAlignment="Right" >

<Button x:Name="BtnAdmin" Margin="5 0 10 0"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnAdminClick">

<materialDesign:PackIcon x:Name="IconBtnKey" Kind="Login"/>

</Button>

<Button x:Name="BtnMinimize" Content="{materialDesign:PackIcon Kind=WindowMinimize}"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMinimize\_Click"/>

<Button x:Name="BtnMaximizeMin"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMaximizeMin\_Click">

<materialDesign:PackIcon x:Name="IconMaximize" Kind="WindowMaximize"/>

</Button>

<Button x:Name="BtnClose" Content="{materialDesign:PackIcon Kind=Close}"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnCloseClick"/>

</StackPanel>

<materialDesign:ColorZone Grid.Column="1" Grid.Row="1" VerticalAlignment="Stretch"

Mode="PrimaryMid" Foreground="White"

Padding="0">

<StackPanel Orientation="Horizontal" VerticalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Left" >

<Button x:Name="BtnMyAccount" Content="{materialDesign:PackIcon Kind=Account, Size=24}" Foreground="White"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMyAccount\_Click" Visibility="Collapsed"/>

<Image Height="30" Width="30" x:Name="ImgUserPhoto" Stretch="UniformToFill" Margin="10 0 10 0"/>

<TextBlock Height="30" Width="200" x:Name="TbUserInfo" Margin="10 0 10 0" FontSize="14" Padding="3" FontWeight="Bold" />

<materialDesign:Badged x:Name="BadgeCount" BadgePlacementMode="Left" Padding="0" VerticalContentAlignment="Center">

<Button VerticalAlignment="Center" x:Name="BtnBuy" Foreground="White" Style="{StaticResource MaterialDesignFlatButton}" Content="{materialDesign:PackIcon Kind=Cart, Size=24}"

ToolTip="Корзина" Click="BtnBuyClick" Height="40" />

</materialDesign:Badged>

<Button x:Name="BtnMyFeedBacks" Content="{materialDesign:PackIcon Kind=CommentEye, Size=24}" Foreground="White" ToolTip="Отзывы на товары"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMyFeedBacks\_Click" Visibility="Collapsed"/>

<Button x:Name="BtnMyDevFeedBacks" Content="{materialDesign:PackIcon Kind=CommentAccount, Size=24}" Foreground="White" ToolTip="Отзывы на производителей"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMyDevFeedBacks\_Click" Visibility="Collapsed"/>

<Button x:Name="BtnMyOrders" Content="{materialDesign:PackIcon Kind=ListStatus, Size=24}" Foreground="White" ToolTip="Мои заказы"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnMyOrders\_Click" Visibility="Collapsed"/>

<Button x:Name="BtnEdit" Content="{materialDesign:PackIcon Kind=Edit, Size=24}"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnEditClick" Foreground="White" Visibility="Collapsed"/>

<Button x:Name="BtnSellerOrders" Content="{materialDesign:PackIcon Kind=ListStatus, Size=24}" Foreground="White" ToolTip="Мои заказы"

Style="{StaticResource MaterialDesignFlatButton}" Click="BtnSellerOrders\_Click" Visibility="Collapsed"/>

</StackPanel>

<Button x:Name="BtnBack" Content="{materialDesign:PackIcon Kind=ArrowBack, Size=24}" Foreground="White" HorizontalAlignment="Right" Margin="5 0 10 0"

Style="{StaticResource MaterialDesignFlatButton}" Visibility="Collapsed" Click="BtnBack\_Click"/>

</StackPanel >

</materialDesign:ColorZone>

<Frame x:Name="MainFrame" Grid.Row="2"

NavigationUIVisibility="Hidden" BorderBrush="{StaticResource MaterialDesignPaper}"

BorderThickness="2" ContentRendered="MainFrameContentRendered"/>

<TextBlock Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="2" Text="Сервис доставки фермерских продуктов"

Style="{StaticResource MaterialDesignHeadline1TextBlock}"

FontSize="16" VerticalAlignment="Center" FontWeight="Bold" Padding="30 0" Margin="0 0 400 0"

MouseDown="TextBlock\_MouseDown"/>

<materialDesign:PackIcon Grid.Row="0" Grid.Column="0" Kind="DeliveryDining"

Foreground="Red"

VerticalAlignment="Center"

Margin="10 0"/>

<TextBlock Grid.Row="3" Grid.Column="0" x:Name="TextBlockCount" Foreground="White" Background="{StaticResource PrimaryHueDarkBrush}" />

<materialDesign:DialogHost Grid.Row="0" Grid.Column="0" Grid.ColumnSpan="4" x:Name="AccessWindow"

BorderThickness="1"

BorderBrush="Aqua"

HorizontalAlignment="Center"

VerticalAlignment="Center" CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<Grid>

<Grid.ColumnDefinitions>

<ColumnDefinition />

</Grid.ColumnDefinitions>

<Grid.RowDefinitions>

<RowDefinition Height="75"/>

<RowDefinition Height="40"/>

<RowDefinition Height="60"/>

<RowDefinition Height="40"/>

<RowDefinition/>

</Grid.RowDefinitions>

<TextBlock Grid.Row="0" Text="Панель администратора"

HorizontalAlignment="Center"

VerticalAlignment="Center"

FontSize="20"

FontWeight="Bold" />

<StackPanel Orientation="Horizontal" Grid.Row="1">

<TextBlock

Grid.Row="1"

Text="Логин"

Margin="20 0"

VerticalAlignment="Center" Width="100"/>

<TextBox x:Name="TbLogin" Text="client"

Grid.Column="1"

Width="200"/>

</StackPanel>

<StackPanel Orientation="Horizontal" Grid.Row="2">

<TextBlock

Grid.Row="2" Grid.Column="0"

Text="Пароль" Margin="20 0"

VerticalAlignment="Center" Width="100"/>

<PasswordBox x:Name="TbPass" Password="1" Height="30" Width="200" />

</StackPanel >

<StackPanel Orientation="Horizontal" Grid.Row="3" HorizontalAlignment="Center" Margin="5">

<Button Width="100" Height="30"

Content="OK"

x:Name="BtnOk"

Click="BtnOkClick"/>

<Button Width="100"

Height="30"

Content="Cancel"

x:Name="BtnCancel"

Click="BtnCancelClick"/>

</StackPanel>

<StackPanel Grid.Row="4 " Orientation="Horizontal" HorizontalAlignment="Center">

<Button FontSize="10"

Height="20"

Content="Регистрация как покупатель"

x:Name="BtnRegsBuyer"

Click="BtnRegsBuyer\_Click"/> <Button FontSize="10"

Height="20"

Content="Регистрация как продавец"

x:Name="BtnRegsSeller"

Click="BtnRegsSeller\_Click"/>

</StackPanel>

</Grid>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

<materialDesign:DialogHost Grid.Row="0" x:Name="BuysWindow" Grid.RowSpan="4"

BorderThickness="1"

BorderBrush="Aqua"

HorizontalAlignment="Center"

VerticalAlignment="Center" Padding="0"

CloseOnClickAway="True">

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<ListBox x:Name="LbBuy"

ScrollViewer.HorizontalScrollBarVisibility="Auto"

Height="400" Width="540" SelectedValuePath="Key">

<ListBox.ItemTemplate>

<DataTemplate>

<StackPanel Margin="5" Orientation="Horizontal">

<Image Width="60" Height="60" Source="{Binding Path=Key.GetPhoto}" />

<StackPanel Margin="10 0 0 0">

<TextBlock FontSize="14" Text="{Binding Path=Key.Name, StringFormat={}Товар: {0}}" Width="350" HorizontalAlignment="Left" FontWeight="Bold" />

<TextBlock FontSize="14" Text="{Binding Path=Key.Price, StringFormat={}цена за шт.: {0:F2} руб.}" Width="300" HorizontalAlignment="Left" />

<StackPanel Orientation="Horizontal" Margin="0 0 10 0">

<TextBlock FontSize="14" Text="{Binding Path=Value.Count, StringFormat={}количество: {0}}" Width="300" HorizontalAlignment="Left" />

<Button x:Name="BtnAdd" Content="{materialDesign:PackIcon Kind=Plus}" Width="60" DataContext="{Binding Key}" Click="BtnAdd\_Click"/>

<Button x:Name="BtnDelete" Content="{materialDesign:PackIcon Kind=Minus}" Width="60" DataContext="{Binding Key}" Click="BtnDelete\_Click\_1"/>

</StackPanel>

<TextBlock FontSize="14" Text="{Binding Path=Value.Total, StringFormat={}итого: {0:F2} руб.}" Width="300" HorizontalAlignment="Right" />

</StackPanel>

</StackPanel>

</DataTemplate>

</ListBox.ItemTemplate>

</ListBox>

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<Button x:Name="BtnOkBuy"

Content="Закрыть" Click="BtnOkBuy\_Click" />

<Button x:Name="BtnDelete" Content="Удалить"

Click="BtnDelete\_Click" />

<Button x:Name="BtnBuyItem" Click="BtnBuyItemClick"

Content="Купить" />

</StackPanel>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Window>

using ChefApp.Models;

using ChefApp.Pages;

using ChefApp.Windows;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using static ChefApp.Models.Manager;

using Word = Microsoft.Office.Interop.Word;

namespace ChefApp

{ /// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{ bool \_login = false;

int \_itemcount = 0;

public MainWindow()

{ InitializeComponent();

MainFrame.Navigate(new CatalogPage());

BtnEdit.Visibility = Visibility.Collapsed;

BtnSellerOrders.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnMyFeedBacks.Visibility = Visibility.Collapsed;

BtnMyDevFeedBacks.Visibility = Visibility.Collapsed;

BtnMyOrders.Visibility = Visibility.Collapsed; // TbCount.Visibility = Visibility.Collapsed;

//BtnBuy.Visibility = Visibility.Collapsed;

BadgeCount.Visibility = Visibility.Collapsed;

Manager.MainFrame = MainFrame;

// Manager.TbCount = TbCount;

Manager.BadgeCount = BadgeCount;

}

private void BtnCloseClick(object sender, RoutedEventArgs e)

{ Close();

}

private void Window\_Closing(object sender, System.ComponentModel.CancelEventArgs e)

{ // на экране отображается форма с двумя кнопками

MessageBoxResult x = MessageBox.Show("Вы действительно хотите закрыть приложение?",

"Выйти", MessageBoxButton.OKCancel, MessageBoxImage.Question);

if (x == MessageBoxResult.Cancel)

e.Cancel = true;

}

private void BtnAdminClick(object sender, RoutedEventArgs e)

{ if (\_login)

{ MessageBoxResult messageBoxResult = MessageBox.Show($"Выйти из системы??? ", "Выход", MessageBoxButton.OKCancel,

MessageBoxImage.Question);

if (messageBoxResult == MessageBoxResult.OK)

{ IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Login;

\_login = false;

BtnEdit.Visibility = Visibility.Collapsed;

BtnSellerOrders.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnMyFeedBacks.Visibility = Visibility.Collapsed;

BtnMyDevFeedBacks.Visibility = Visibility.Collapsed;

BtnMyOrders.Visibility = Visibility.Collapsed;

// TbCount.Visibility = Visibility.Collapsed;

// BtnBuy.Visibility = Visibility.Collapsed;

BadgeCount.Visibility = Visibility.Collapsed;

Manager.currentDeveloper = null;

Manager.currentClient = null;

BadgeCount.Badge = null;

ImgUserPhoto.Source = null;

ImgUserPhoto.Visibility = Visibility.Collapsed;

//TbUserInfo.Visibility = Visibility.Collapsed;

TbUserInfo.Text = "";

Manager.buyGoods.Clear();

MainFrame.NavigationService.Refresh();

//MainFrame.Navigate(new CatalogPage());

TbPass.Password = "";

TbLogin.Text = "";

return;

}

}

else

AccessWindow.IsOpen = true;

}

private void BtnCancelClick(object sender, RoutedEventArgs e)

{ AccessWindow.IsOpen = false;

}

private void BtnOkClick(object sender, RoutedEventArgs e)

{ try

{ //загрузка всех пользователей из БД в список

List<Client> clients = FermerBdEntities.GetContext().Clients.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

Client u = clients.FirstOrDefault(p => p.Password == TbPass.Password && p.UserName == TbLogin.Text);

List<Developer> developers = FermerBdEntities.GetContext().Developers.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

Developer d = developers.FirstOrDefault(p => p.Password == TbPass.Password && p.UserName == TbLogin.Text);

if (u != null)

{ // логин и пароль корректные запускаем главную форму приложения

IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Logout;

\_login = true;

Manager.currentClient = u;

BtnMyFeedBacks.Visibility = Visibility.Visible;

BtnMyDevFeedBacks.Visibility = Visibility.Visible;

BtnMyAccount.Visibility = Visibility.Visible;

BtnMyOrders.Visibility = Visibility.Visible;

// TbCount.Visibility = Visibility.Visible;

// BtnBuy.Visibility = Visibility.Visible;

BadgeCount.Visibility = Visibility.Visible;

ImgUserPhoto.Visibility = Visibility.Visible;

ImgUserPhoto.Source = new BitmapImage(new Uri(u.GetPhoto));

TbUserInfo.Text = Manager.currentClient.GetFio;

AccessWindow.IsOpen = false;

MessageBox.Show("Вы вошли в систему как покупатель");

//MainFrame.NavigationService.RemoveBackEntry();

//MainFrame.Navigate(new CatalogPage());

MainFrame.NavigationService.Refresh();

return;

}

if (d != null)

{ // логин и пароль корректные запускаем главную форму приложения

IconBtnKey.Kind = MaterialDesignThemes.Wpf.PackIconKind.Logout;

\_login = true;

Manager.currentDeveloper = d;

BtnMyAccount.Visibility = Visibility.Visible;

ImgUserPhoto.Visibility = Visibility.Visible;

ImgUserPhoto.Source = new BitmapImage(new Uri(d.GetPhoto));

AccessWindow.IsOpen = false;

BtnEdit.Visibility = Visibility.Visible;

BtnSellerOrders.Visibility = Visibility.Visible;

TbUserInfo.Text = Manager.currentDeveloper.GetFio;

MessageBox.Show("Вы вошли в систему как продавец");

//MainFrame.NavigationService.RemoveBackEntry();

//MainFrame.Navigate(new CatalogPage());

MainFrame.NavigationService.Refresh();

return;

}

MessageBox.Show("Не верный логин или пароль");

}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString());

}

}

private void MainFrameContentRendered(object sender, EventArgs e)

{ if (MainFrame.CanGoBack)

{ BtnBack.Visibility = Visibility.Visible;

BtnAdmin.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnMyFeedBacks.Visibility = Visibility.Collapsed;

// TbCount.Visibility = Visibility.Collapsed;

//BtnBuy.Visibility = Visibility.Collapsed;

BtnMyOrders.Visibility = Visibility.Collapsed;

BadgeCount.Visibility = Visibility.Collapsed;

BtnMyDevFeedBacks.Visibility = Visibility.Collapsed;

BtnEdit.Visibility = Visibility.Collapsed;

BtnSellerOrders.Visibility = Visibility.Collapsed;

}

else

{ BtnBack.Visibility = Visibility.Collapsed;

BtnAdmin.Visibility = Visibility.Visible;

if ((Manager.currentDeveloper == null) && (Manager.currentClient == null))

{ BtnMyOrders.Visibility = Visibility.Collapsed;

BtnMyAccount.Visibility = Visibility.Collapsed;

BtnMyFeedBacks.Visibility = Visibility.Collapsed;

BtnMyDevFeedBacks.Visibility = Visibility.Collapsed;

// TbCount.Visibility = Visibility.Collapsed;

//BtnBuy.Visibility = Visibility.Collapsed;

BadgeCount.Visibility = Visibility.Collapsed;

}

else

{ BtnMyAccount.Visibility = Visibility.Visible;

if (Manager.currentDeveloper != null)

{ BtnEdit.Visibility = Visibility.Visible;

BtnSellerOrders.Visibility = Visibility.Visible;

BtnMyFeedBacks.Visibility = Visibility.Collapsed;

BtnMyDevFeedBacks.Visibility = Visibility.Collapsed; // TbCount.Visibility = Visibility.Collapsed;

//BtnBuy.Visibility = Visibility.Collapsed;

BtnMyOrders.Visibility = Visibility.Collapsed;

BadgeCount.Visibility = Visibility.Collapsed;

TbUserInfo.Text = Manager.currentDeveloper.GetFio;

ImgUserPhoto.Source = null;

ImgUserPhoto.Source = new BitmapImage(new Uri(Manager.currentDeveloper.GetPhoto));

}

else

{ BtnEdit.Visibility = Visibility.Collapsed;

BtnSellerOrders.Visibility = Visibility.Collapsed;

BtnMyFeedBacks.Visibility = Visibility.Visible;

BtnMyDevFeedBacks.Visibility = Visibility.Visible;

BtnMyOrders.Visibility = Visibility.Visible;

// TbCount.Visibility = Visibility.Visible;

//BtnBuy.Visibility = Visibility.Visible;

BadgeCount.Visibility = Visibility.Visible;

if (Manager.currentClient != null)

{ ImgUserPhoto.Source = null;

TbUserInfo.Text = Manager.currentClient.GetFio;

ImgUserPhoto.Source = new BitmapImage(new Uri(Manager.currentClient.GetPhoto));

} } } } }

private void BtnEditClick(object sender, RoutedEventArgs e)

{MainFrame.Navigate(new GoodPage());

}

private void BtnMaximizeMin\_Click(object sender, RoutedEventArgs e)

{ if (this.WindowState == WindowState.Normal)

{ this.WindowState = WindowState.Maximized;

IconMaximize.Kind = MaterialDesignThemes.Wpf.PackIconKind.WindowRestore;

}

else

{ this.WindowState = WindowState.Normal;

IconMaximize.Kind = MaterialDesignThemes.Wpf.PackIconKind.WindowMaximize;

}

}

private void BtnMinimize\_Click(object sender, RoutedEventArgs e)

{ this.WindowState = WindowState.Minimized;

}

private void TextBlock\_MouseDown(object sender, MouseButtonEventArgs e)

{ if (e.ChangedButton == MouseButton.Left)

this.DragMove();

}

private void BtnBack\_Click(object sender, RoutedEventArgs e)

{

if (Manager.buyGoods.Count == 0)

BadgeCount.Badge = null;

Manager.MainFrame.GoBack();

}

private void BtnRegsSeller\_Click(object sender, RoutedEventArgs e)

{ RegsWindow regsWindow = new RegsWindow();

regsWindow.ShowDialog();

}

private void BtnRegsBuyer\_Click(object sender, RoutedEventArgs e)

{ RegsBuyerWindow regsWindow = new RegsBuyerWindow();

regsWindow.ShowDialog();

}

private void BtnMyAccount\_Click(object sender, RoutedEventArgs e)

{ if (Manager.currentClient != null)

{ MainFrame.Navigate(new EditClientPage(Manager.currentClient));

}

if (Manager.currentDeveloper != null)

{ MainFrame.Navigate(new EditDeveloperPage(Manager.currentDeveloper)); }

}

private void BtnBuyClick(object sender, RoutedEventArgs e)

{ LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

BuysWindow.IsOpen = true;

}

private void BtnOkBuy\_Click(object sender, RoutedEventArgs e)

{ BuysWindow.IsOpen = false;

}

private void BtnDelete\_Click(object sender, RoutedEventArgs e)

{ // вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить товар из корзины???", "Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{ if (LbBuy.SelectedIndex >= 0)

{ var x = (LbBuy.SelectedValue as Good);

Manager.buyGoods.Remove(x);

LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

BadgeCount.Badge = Manager.buyGoods.Count;

// TbCount.Text = $"В корзине {Manager.buyGoods.Count} товаров";

}

}

if (Manager.buyGoods.Count == 0)

BadgeCount.Badge = null;

}

private void BtnMyFeedBacks\_Click(object sender, RoutedEventArgs e)

{ MainFrame.Navigate(new FeedBackPage());

}

private void BtnMyDevFeedBacks\_Click(object sender, RoutedEventArgs e)

{ MainFrame.Navigate(new DevFeedBackPage());

}

private void BtnAdd\_Click(object sender, RoutedEventArgs e)

{ var x = (sender as Button).DataContext as Good;

//Good g = FermerBdEntities.GetContext().Goods.Find(x.Good.Id);

//MessageBox.Show(x.Name);

if (Manager.buyGoods.ContainsKey(x))

{ int k = Manager.buyGoods[x].Count + 1;

double p = x.Price \* k;

Manager.buyGoods[x] = new Manager.BuyItem { Count = k, Total = p };

LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

}

}

private void BtnDelete\_Click\_1(object sender, RoutedEventArgs e)

{ var x = (sender as Button).DataContext as Good;

//Good g = FermerBdEntities.GetContext().Goods.Find(x.Good.Id);

//MessageBox.Show(x.Name);

if (Manager.buyGoods.ContainsKey(x))

{ int k = Manager.buyGoods[x].Count;

if (k > 0) k--;

if (k == 0)

{ MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить товар из корзины???", "Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{ Manager.buyGoods.Remove(x);

LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

BadgeCount.Badge = Manager.buyGoods.Count;

// TbCount.Text = $"В корзине {Manager.buyGoods.Count} товаров";

}

else

{ k = 1;

BadgeCount.Badge = Manager.buyGoods.Count;

// TbCount.Text = $"В корзине {Manager.buyGoods.Count} товаров";

double p = x.Price \* k;

Manager.buyGoods[x] = new Manager.BuyItem { Count = k, Total = p };

LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

}

}

else

{ BadgeCount.Badge = Manager.buyGoods.Count;

// TbCount.Text = $"В корзине {Manager.buyGoods.Count} товаров";

double p = x.Price \* k;

Manager.buyGoods[x] = new Manager.BuyItem { Count = k, Total = p };

LbBuy.ItemsSource = null;

LbBuy.ItemsSource = Manager.buyGoods;

}

}

if (Manager.buyGoods.Count == 0)

BadgeCount.Badge = null;

}

private void BtnBuyItemClick(object sender, RoutedEventArgs e)

{ BuysWindow.IsOpen = false;

if (Manager.buyGoods.Count == 0)

return;

MainFrame.Navigate(new AddOrderPage(null));

}

private void Window\_ContentRendered(object sender, EventArgs e)

{

}

private void BtnMyOrders\_Click(object sender, RoutedEventArgs e)

{ MainFrame.Navigate(new MyOrdersPage());

}

private void BtnSellerOrders\_Click(object sender, RoutedEventArgs e)

{ MainFrame.Navigate(new SellerOrdersPage());

}

}}

<Page x:Class="ChefApp.Pages.CatalogPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:ChefApp.Pages" xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Каталог"

IsVisibleChanged="Page\_IsVisibleChanged"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="\*"/>

<RowDefinition Height="30"/>

</Grid.RowDefinitions>

<WrapPanel Grid.Row="0" HorizontalAlignment="Stretch" >

<StackPanel Orientation="Horizontal">

<TextBlock Text="Введите название для поиска:" FontSize="14" Padding="3"/>

<TextBox Width="275" x:Name="TBoxSearch" FontSize="14" Padding="3"

TextChanged="TBoxSearchTextChanged"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Выберите категорию" FontSize="14" Padding="3"

/>

<ComboBox Width="275" x:Name="ComboCategory" FontSize="14" Padding="3"

SelectionChanged="ComboTypeSelectionChanged"

DisplayMemberPath="Name"/>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Выберите производителя" FontSize="14" Padding="3" />

<ComboBox Width="275" x:Name="ComboDeveloper" FontSize="14" Padding="3" SelectionChanged="ComboDeveloper\_SelectionChanged\_1">

<ComboBox.ItemTemplate>

<DataTemplate>

<StackPanel Orientation="Horizontal">

<Image Source="{Binding GetPhoto}" Width="15" Margin="5 0 5 0" />

<TextBlock Text="{Binding GetFio}" />

<materialDesign:RatingBar

Value="{Binding Rate}"

Orientation="Horizontal"

Foreground="Gold"

Margin="0" IsReadOnly="True"/>

</StackPanel>

</DataTemplate>

</ComboBox.ItemTemplate>

</ComboBox>

</StackPanel>

<StackPanel Orientation="Horizontal">

<TextBlock Text="Сортировать по" FontSize="14" Padding="3"

/>

<ComboBox Width="275" FontSize="14" Padding="3"

x:Name="ComboSort"

SelectionChanged="ComboSortSelectionChanged" >

<ComboBoxItem Content="По возрастанию цены"/>

<ComboBoxItem Content="По убыванию цены"/>

</ComboBox>

</StackPanel>

</WrapPanel>

<Border

BorderThickness="0.5"

BorderBrush="{DynamicResource PrimaryHueMidBrush}"

MinWidth="256"

MinHeight="256" Grid.Row="1"

ClipToBounds="True">

<ListView x:Name="LViewGoods"

ScrollViewer.HorizontalScrollBarVisibility="Disabled"

>

<ListView.ItemContainerStyle>

<Style TargetType="{x:Type ListViewItem}">

<Setter Property="Background" Value="{Binding GetColor}" />

</Style>

</ListView.ItemContainerStyle>

<ListView.ItemsPanel >

<ItemsPanelTemplate>

<WrapPanel Orientation="Horizontal" HorizontalAlignment="Center"/>

</ItemsPanelTemplate>

</ListView.ItemsPanel>

<ListView.ItemTemplate>

<DataTemplate>

<Grid Margin="20" Width="300">

<Grid.RowDefinitions>

<RowDefinition Height="auto" />

<RowDefinition Height="250" />

<RowDefinition Height="auto" />

<RowDefinition Height="20" />

<RowDefinition Height="auto" />

</Grid.RowDefinitions>

<TextBlock Text="{Binding Name}" Height="Auto" VerticalAlignment="Center" FontWeight="Bold" FontSize="20"/>

<Image Width="240" Grid.Row="1" Stretch="Uniform"

HorizontalAlignment="Center" Margin="5"

Source="{Binding Path=GetPhoto}"/>

<StackPanel Grid.Row="2" Height="150">

<StackPanel Orientation="Horizontal">

<TextBlock Text="{Binding Developer.GetFio}" Height="Auto" VerticalAlignment="Center" FontWeight="Bold" FontSize="14" />

<Button Style="{StaticResource MaterialDesignToolButton}" Width="30" ToolTip="Подробнее" x:Name="BtnMoreInfo" Padding="2 0 2 0"

materialDesign:RippleAssist.IsCentered="True" Click="BtnMoreInfo\_Click" >

<materialDesign:PackIcon Kind="DotsVertical" />

</Button>

</StackPanel>

<TextBlock Text="{Binding Price, StringFormat='c', ConverterCulture='ru-RU'}" Height="Auto" VerticalAlignment="Center" FontWeight="Bold" FontSize="20"/>

<materialDesign:RatingBar Value="{Binding Rate}" Orientation="Horizontal" Foreground="Gold" Margin="0" IsReadOnly="True"/>

</StackPanel>

<TextBlock Text="{Binding GetStatus}" Height="Auto" Grid.Row="3"/>

<StackPanel HorizontalAlignment="Right" VerticalAlignment="Top" Orientation="Horizontal" Margin="8" Grid.Row="2">

<Button x:Name="BtnBuy" Style="{StaticResource MaterialDesignFloatingActionAccentButton}" ToolTip="В корзину" Visibility="{Binding GetVisibilityBuy}" Click="BtnBuy\_Click">

<materialDesign:PackIcon Kind="BasketFill" />

</Button>

<Button x:Name="BtnEditGood" Style="{StaticResource MaterialDesignFloatingActionAccentButton}" ToolTip="Редактировать" Visibility="{Binding GetVisibilityEdit}" Click="BtnEditGood\_Click" >

<materialDesign:PackIcon Kind="Edit" />

</Button>

</StackPanel>

</Grid>

</DataTemplate>

</ListView.ItemTemplate>

</ListView>

</Border>

<TextBlock Grid.Row="2" x:Name="TextBlockCount" FontSize="12" FontWeight="Bold" />

<materialDesign:DialogHost x:Name="DialogHostMoreInformation"

Background="{DynamicResource PrimaryHueDarkBrush }" MinWidth="800" MaxWidth="800"

MinHeight="400" MaxHeight="400" BorderThickness="1" HorizontalAlignment="Center" VerticalAlignment="Center" Grid.RowSpan="3"

>

<materialDesign:DialogHost.DialogContent>

<StackPanel Margin="20" >

<Image Source="{Binding Path=Developer.GetPhoto}" Height="240" Width="320" Stretch="Uniform" />

<Grid >

<Grid.RowDefinitions>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

<RowDefinition/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="100"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<TextBlock Text="ФИО:" TextWrapping="Wrap" Grid.Row="0" Grid.Column="0"/>

<TextBlock FontWeight="Bold" Text="{Binding Developer.GetFio}" TextWrapping="Wrap" Grid.Row="0" Grid.Column="1" />

<TextBlock Text="Информация:" TextWrapping="Wrap" Grid.Row="1" Grid.Column="0"/>

<TextBox TextWrapping="Wrap" VerticalAlignment="Center" Grid.Row="1" Grid.Column="1" IsReadOnly="True" Text="{Binding Developer.Info}" Width="300" Height="200" VerticalScrollBarVisibility="Auto" /> <TextBlock Text="Рейтинг:" TextWrapping="Wrap" Grid.Row="2" Grid.Column="0"/>

<materialDesign:RatingBar Grid.Row="2" Grid.Column="1"

Value="{Binding Developer.Rate}"

Orientation="Horizontal"

Foreground="Gold"

Margin="0" />

</Grid>

<Button x:Name="btnOk" Click="BtnOk\_Click" Width="100" HorizontalAlignment="Center">

<StackPanel Orientation="Horizontal" HorizontalAlignment="Center">

<materialDesign:PackIcon Kind="Check" Foreground="#FFF" Margin="5" VerticalAlignment="Center"/>

<TextBlock Text="OK" VerticalAlignment="Center"/>

</StackPanel>

</Button>

</StackPanel>

</materialDesign:DialogHost.DialogContent>

</materialDesign:DialogHost>

</Grid>

</Page>

using ChefApp.Models;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using Word = Microsoft.Office.Interop.Word;

namespace ChefApp.Pages

{ /// <summary>

/// Логика взаимодействия для CatalogPage.xaml

/// </summary>

public partial class CatalogPage : Page

{ int \_itemcount = 0;

public CatalogPage()

{ InitializeComponent();

LoadData();

LoadCategories();

//if (Manager.currentClient != null)

//{

//}

}

void LoadData()

{ if (Manager.currentDeveloper != null)

{ ComboDeveloper.IsEnabled = false;

LViewGoods.ItemsSource = FermerBdEntities.GetContext().Goods.Where(p => p.DeveloperUserName == Manager.currentDeveloper.UserName).OrderBy(p => p.Name).ThenBy(p => p.Price).ToList();

}

else

{ LViewGoods.ItemsSource = FermerBdEntities.GetContext().Goods.OrderBy(p => p.Name).ThenBy(p => p.Price).ToList();

ComboDeveloper.IsEnabled = true;

}

\_itemcount = LViewGoods.Items.Count;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {\_itemcount} записей из {\_itemcount}";

}

void LoadCategories()

{ var categories = FermerBdEntities.GetContext().Categories.OrderBy(p =>p.Name).ToList();

categories.Insert(0, new Category

{ Name = "Все типы"

}

);

ComboCategory.ItemsSource = categories;

ComboCategory.SelectedIndex = 0;

var developers = FermerBdEntities.GetContext().Developers.OrderBy(p => p.LastName).ToList();

developers.Insert(0, new Developer

{ LastName = "Все"

}

);

ComboDeveloper.ItemsSource = developers;

ComboDeveloper.SelectedIndex = 0;

}

// Поиск товаров, которые содержат данную поисковую строку

private void TBoxSearchTextChanged(object sender, TextChangedEventArgs e)

{ UpdateData();

}

// Поиск товаров конкретного производителя

private void ComboTypeSelectionChanged(object sender, SelectionChangedEventArgs e)

{ UpdateData();

}

/// <summary>

/// Метод для фильтрации и сортировки данных

/// </summary>

private void UpdateData()

{ LViewGoods.ItemsSource = null;

// получаем текущие данные из бд

List<Good> currentGoods;

if (Manager.currentDeveloper != null)

{ currentGoods = FermerBdEntities.GetContext().Goods.Where(p => p.DeveloperUserName == Manager.currentDeveloper.UserName).OrderBy(p => p.Name).ThenBy(p => p.Price).ToList();

}

else

{ currentGoods = FermerBdEntities.GetContext().Goods.OrderBy(p => p.Name).ThenBy(p => p.Price).ToList();

if (ComboDeveloper.SelectedIndex > 0)

currentGoods = currentGoods.Where(p => p.DeveloperUserName == (ComboDeveloper.SelectedItem as Developer).UserName).ToList();

}

// выбор только тех товаров, которые принадлежат данному производителю

if (ComboCategory.SelectedIndex > 0)

currentGoods = currentGoods.Where(p => p.CategoryId == (ComboCategory.SelectedItem as Category).Id).ToList();

// сортировка

if (ComboSort.SelectedIndex >= 0)

{ // сортировка по возрастанию цены

if (ComboSort.SelectedIndex == 0)

currentGoods = currentGoods.OrderBy(p => p.Price).ToList();

// сортировка по убыванию цены

if (ComboSort.SelectedIndex == 1)

currentGoods = currentGoods.OrderByDescending(p => p.Price).ToList();

}

// выбор тех товаров, в названии которых есть поисковая строка

currentGoods = currentGoods.Where(p => p.Name.ToLower().Contains(TBoxSearch.Text.ToLower())).ToList();

// В качестве источника данных присваиваем список данных

LViewGoods.ItemsSource = currentGoods;

// отображение количества записей

TextBlockCount.Text = $" Результат запроса: {currentGoods.Count} записей из {\_itemcount}";

}

// сортировка товаров

private void ComboSortSelectionChanged(object sender, SelectionChangedEventArgs e)

{ UpdateData();

}

void ExportWord(Good selected)

{ string fileName = Directory.GetCurrentDirectory() + "\\" + "AboutBanner" + ".dotx";

Word.Application wrdApp = new Word.Application();

try

{ Word.Document document = wrdApp.Documents.Add(fileName);

document.Bookmarks["Name"].Range.Text = selected.Name;

document.Bookmarks["Price"].Range.Text = selected.Price.ToString();

document.Bookmarks["Category"].Range.Text = selected.Category.Name;

document.Bookmarks["UserName"].Range.Text = $"{ selected.Developer.FirstName} { selected.Developer.LastName} { selected.Developer.MiddleName}";

object oRange = document.Bookmarks["Photo"].Range;

object saveWithDocument = true;

object missing = Type.Missing;

string pictureName = selected.GetPhoto;

document.InlineShapes.AddPicture(pictureName, ref missing, ref saveWithDocument, ref oRange);

//document.SaveAs("В");

}

catch (Exception ex)

{ MessageBox.Show(ex.ToString());

wrdApp.Quit();

}

finally

{ wrdApp.Visible = true;

wrdApp.ScreenUpdating = true;

}

}

private void Page\_IsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{ //обновление данных после каждой активации окна

if (Visibility == Visibility.Visible)

{ //FermerBdEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

//LViewGoods.ItemsSource = FermerBdEntities.GetContext().Goods.OrderBy(p =>

//p.Name).ToList();

LoadData();

LoadCategories();

if (Manager.buyGoods.Count == 0)

Manager.BadgeCount.Badge = null;

}

}

private void ComboDeveloper\_SelectionChanged(object sender, SelectionChangedEventArgs e)

{ }

private void ComboDeveloper\_SelectionChanged\_1(object sender, SelectionChangedEventArgs e)

{ UpdateData();

}

private void BtnMoreInfo\_Click(object sender, RoutedEventArgs e)

{ Good selected = (sender as Button).DataContext as Good;

DialogHostMoreInformation.DataContext = selected;

DialogHostMoreInformation.IsOpen = true;

}

private void BtnOk\_Click(object sender, RoutedEventArgs e)

{ DialogHostMoreInformation.IsOpen = false;

}

private void BtnBuy\_Click(object sender, RoutedEventArgs e)

{ var x = (sender as Button).DataContext as Good;

Status status = FermerBdEntities.GetContext().Status.Find(1);

if (Manager.buyGoods.ContainsKey(x))

{ int k = Manager.buyGoods[x].Count + 1;

double p = x.Price \* k;

Manager.buyGoods[x] = new Manager.BuyItem { Count = k, Total = p, StatusName = status.Name, StatusColor = status.Color };

}

else

{ int k = 1;

double p = x.Price \* k;

Manager.buyGoods[x] = new Manager.BuyItem { Count = k, Total = p, StatusName = status.Name, StatusColor = status.Color };

}

MessageBox.Show("Товар добавлен в корзину " + Manager.buyGoods.Count.ToString());

// Manager.TbCount.Text = $"В корзине {Manager.buyGoods.Count} товаров";

Manager.BadgeCount.Badge = Manager.buyGoods.Count;

}

private void BtnEditGood\_Click(object sender, RoutedEventArgs e)

{ Manager.MainFrame.Navigate(new AddGoodPage((sender as Button).DataContext as Good));

}

}}

<Window x:Class="ChefApp.Windows.RegsBuyerWindow"

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:ChefApp.Windows"

mc:Ignorable="d"

Title="Регистрация покупателя" Height="700" Width="800" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="55\*" />

<RowDefinition Height="53\*" />

<RowDefinition Height="53\*" />

<RowDefinition Height="56\*" />

<RowDefinition Height="56\*" />

<RowDefinition Height="56\*" />

<RowDefinition Height="53\*" />

<RowDefinition Height="100"/>

<RowDefinition Height="54\*" />

<RowDefinition Height="53\*" />

<RowDefinition Height="80"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="260" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="Имя пользователя" VerticalAlignment="Center" Grid.Row="0" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox x:Name="TbUserName" Text="{Binding UserName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="0" Height="30" Margin="0,7" />

<TextBlock Text="Фамилия" VerticalAlignment="Center" Grid.Row="1" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding LastName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="1" Height="30" Margin="0,7" />

<TextBlock Text="Имя" VerticalAlignment="Center" Grid.Row="2" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding FirstName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="2" Height="30" Margin="0,7" />

<TextBlock Text="Отчество" VerticalAlignment="Center" Grid.Row="3" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding MiddleName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="3" Height="30" Margin="0,7" />

<TextBlock Text="Телефон" VerticalAlignment="Center" Grid.Row="4" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Phone}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="4" Height="30" Margin="0,7" />

<TextBlock Text="Email" VerticalAlignment="Center" Grid.Row="5" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Email}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="5" Height="30" Margin="0,7" />

<TextBlock Text="Адрес" VerticalAlignment="Center" Grid.Row="6" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Address}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="6" Height="30" Margin="0,7" />

<TextBlock Grid.Row="7" Grid.Column="0" Text="Фотография:"

Margin="20 0" Height="30" VerticalAlignment="Top"/>

<Image x:Name="ImagePhoto" Source="{Binding GetPhoto}" Grid.Row="7" Grid.Column="1"

VerticalAlignment="Stretch" HorizontalAlignment="Stretch"/>

<Button x:Name="BtnLoad" Grid.Row="7" Grid.Column="0" Margin="20,30,0,0"

HorizontalAlignment="Left" VerticalAlignment="Top"

Click="BtnLoadClick" Content="Загрузить"/>

<TextBlock Text="Пароль" VerticalAlignment="Center" Grid.Row="8" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<PasswordBox Grid.Column="1" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword1" Grid.Row="8" Height="30" Margin="0,7" />

<TextBlock Text="Подтверждение пароля" VerticalAlignment="Center" Grid.Row="9" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<PasswordBox Grid.Column="1" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword2" Grid.Row="9" Height="30" Margin="0,7" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="10" Grid.ColumnSpan="2" Margin="127,0,103,0" Width="162">

<Button IsDefault="True" Click="Accept\_Click" MinWidth="60" Margin="5">OK</Button>

<Button IsCancel="True" MinWidth="60" Margin="5">Отмена</Button>

</StackPanel>

</Grid>

</Window>

using ChefApp.Models;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace ChefApp.Windows

{ /// <summary>

/// Логика взаимодействия для RegsBuyerWindow.xaml

/// </summary>

public partial class RegsBuyerWindow : Window

{ private Client \_currentItem = new Client();

// путь к файлу

private string \_filePath = null;

// название текущей главной фотографии

private string \_photoName = null;

// текущая папка приложения

private static string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\";

public RegsBuyerWindow()

{ InitializeComponent();

\_currentItem = new Client();

DataContext = \_currentItem;

}

// загрузка фото

private void BtnLoadClick(object sender, RoutedEventArgs e)

{ try

{ //Диалог открытия файла

OpenFileDialog op = new OpenFileDialog();

op.Title = "Select a picture";

op.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg|GIF Files (\*.gif)|\*.gif";

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{ // проверка размера файла

// по условию файл дожен быть не более 2Мб.

FileInfo fileInfo = new FileInfo(op.FileName);

if (fileInfo.Length > (1024 \* 1024 \* 2))

{ // размер файла меньше 2Мб. Поэтому выбрасывается новое исключение

throw new Exception("Размер файла должен быть меньше 2Мб");

}

ImagePhoto.Source = new BitmapImage(new Uri(op.FileName));

\_photoName = op.SafeFileName;

\_filePath = op.FileName;

}

}

catch (Exception ex)

{ MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

\_filePath = null;

}

}

//подбор имени файла

string ChangePhotoName()

{ string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{ while (File.Exists(x))

{ i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

private void Accept\_Click(object sender, RoutedEventArgs e)

{ StringBuilder \_error = CheckFields();

if (\_error.Length > 0)

{ MessageBox.Show(\_error.ToString());

return;

}

try

{ // формируем новое название файла картинки,

string photo = ChangePhotoName();

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentItem.Photo = photo;

\_currentItem.Password = PasswordBoxNewPassword1.Password;

FermerBdEntities.GetContext().Clients.Add(\_currentItem);

FermerBdEntities.GetContext().SaveChanges();

MessageBox.Show("Регистрация прошла успешно");

this.DialogResult = true;

}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString());

return;

}

}

private StringBuilder CheckFields()

{ StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentItem.UserName))

s.AppendLine("Задайте имя пользователя");

if (string.IsNullOrWhiteSpace(\_currentItem.FirstName))

s.AppendLine("Поле имя пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.LastName))

s.AppendLine("Поле фамилия пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Phone))

s.AppendLine("Поле телефон пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Email))

s.AppendLine("Поле email пустое");

if (string.IsNullOrWhiteSpace(\_photoName))

s.AppendLine("фото не выбрано пустое");

if (string.IsNullOrWhiteSpace(PasswordBoxNewPassword1.Password))

s.AppendLine("Введите пароль");

if (PasswordBoxNewPassword1.Password != PasswordBoxNewPassword2.Password)

s.AppendLine("Пароли не совпадают");

List<Developer> users = FermerBdEntities.GetContext().Developers.ToList();

Developer u = users.FirstOrDefault(p => p.UserName == TbUserName.Text);

if (u != null)

{ s.AppendLine("Данный логин занят, выберите другой логин");

}

List<Client> clients = FermerBdEntities.GetContext().Clients.ToList();

Client c = clients.FirstOrDefault(p => p.UserName == TbUserName.Text);

if (c != null)

{ s.AppendLine("Данный логин занят, выберите другой логин");

}

return s;

}

}}

<Window x:Class="ChefApp.Windows.RegsWindow"

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:ChefApp.Windows"

xmlns:materialDesign="http://materialdesigninxaml.net/winfx/xaml/themes"

mc:Ignorable="d"

Title="Регистрация продавца" Height="700" Width="800" WindowStartupLocation="CenterScreen">

<Grid>

<Grid.RowDefinitions>

<RowDefinition />

<RowDefinition Height="100"/>

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition />

<RowDefinition Height="auto"/>

<RowDefinition />

<RowDefinition />

<RowDefinition Height="100"/>

<RowDefinition />

<RowDefinition />

<RowDefinition Height="80"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="260" />

<ColumnDefinition />

</Grid.ColumnDefinitions>

<TextBlock Text="Имя пользователя" VerticalAlignment="Center" Grid.Row="0" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox x:Name="TbUserName" Text="{Binding UserName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="0" Height="30" Margin="0,7" />

<TextBlock Text="Информация" VerticalAlignment="Center" Grid.Row="1" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox x:Name="TbInfo" Text="{Binding Info}" Grid.Column="1" VerticalAlignment="Stretch" HorizontalAlignment="Stretch" Grid.Row="1" Margin="0,7" TextWrapping="Wrap" />

<TextBlock Text="Фамилия" VerticalAlignment="Center" Grid.Row="2" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding LastName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="2" Height="30" Margin="0,7" />

<TextBlock Text="Имя" VerticalAlignment="Center" Grid.Row="3" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding FirstName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="3" Height="30" Margin="0,7" />

<TextBlock Text="Отчество" VerticalAlignment="Center" Grid.Row="4" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding MiddleName}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="4" Height="30" Margin="0,7" />

<TextBlock Text="Телефон" VerticalAlignment="Center" Grid.Row="5" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Phone}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="5" Height="30" Margin="0,7" />

<TextBlock Text="Email" VerticalAlignment="Center" Grid.Row="6" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Email}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="6" Height="30" Margin="0,7" /> <TextBlock Text="Дата рождения" VerticalAlignment="Center" Grid.Row="8" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<DatePicker Text="{Binding Birthday, StringFormat='d', ConverterCulture='ru-RU'}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="8" Height="30" Margin="0,7" />

<TextBlock Text="Адрес" VerticalAlignment="Center" Grid.Row="9" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<TextBox Text="{Binding Address}" Grid.Column="1" VerticalAlignment="Center" Grid.Row="9" Height="30" Margin="0,7" />

<TextBlock Grid.Row="10" Grid.Column="0" Text="Фотография:"

Margin="20 0" Height="30" VerticalAlignment="Top"/>

<Image x:Name="ImagePhoto" Source="{Binding GetPhoto}" Grid.Row="10" Grid.Column="1"

VerticalAlignment="Stretch" HorizontalAlignment="Stretch"/>

<Button x:Name="BtnLoad" Grid.Row="10" Grid.Column="0" Margin="20,30,0,0"

HorizontalAlignment="Left" VerticalAlignment="Top"

Click="BtnLoadClick" Content="Загрузить"/>

<TextBlock Text="Пароль" VerticalAlignment="Center" Grid.Row="11" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<PasswordBox Grid.Column="1" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword1" Grid.Row="11" Height="30" Margin="0,7" />

<TextBlock Text="Подтверждение пароля" VerticalAlignment="Center" Grid.Row="12" Margin="20 0 0 0" Grid.Column="0" Height="16" />

<PasswordBox Grid.Column="1" VerticalAlignment="Center" x:Name="PasswordBoxNewPassword2" Grid.Row="12" Height="30" Margin="0,7" />

<StackPanel HorizontalAlignment="Center" Orientation="Horizontal" Grid.Row="13" Grid.ColumnSpan="2" Margin="127,0,103,0" Width="162">

<Button IsDefault="True" Click="Accept\_Click" MinWidth="60" Margin="5">OK</Button>

<Button IsCancel="True" MinWidth="60" Margin="5">Отмена</Button>

</StackPanel>

</Grid>

</Window>

using ChefApp.Models;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace ChefApp.Windows

{ /// <summary>

/// Логика взаимодействия для RegsWindow.xaml

/// </summary>

/// public partial class RegsWindow : Window

{ //текущий товар

private Developer \_currentItem = new Developer();

// путь к файлу

private string \_filePath = null;

// название текущей главной фотографии

private string \_photoName = null;

// текущая папка приложения

private static string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\";

public RegsWindow()

{ InitializeComponent();

\_currentItem = new Developer();

DataContext = \_currentItem; }

// загрузка фото

private void BtnLoadClick(object sender, RoutedEventArgs e)

{ try

{ //Диалог открытия файла

OpenFileDialog op = new OpenFileDialog();

op.Title = "Select a picture";

op.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg|GIF Files (\*.gif)|\*.gif";

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{ // проверка размера файла

// по условию файл дожен быть не более 2Мб.

FileInfo fileInfo = new FileInfo(op.FileName);

if (fileInfo.Length > (1024 \* 1024 \* 2))

{ // размер файла меньше 2Мб. Поэтому выбрасывается новое исключение

throw new Exception("Размер файла должен быть меньше 2Мб");

}

ImagePhoto.Source = new BitmapImage(new Uri(op.FileName));

\_photoName = op.SafeFileName;

\_filePath = op.FileName;

}

}

catch (Exception ex)

{ MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

\_filePath = null;

}

}

string ChangePhotoName()

{ string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{ while (File.Exists(x))

{ i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

private void Accept\_Click(object sender, RoutedEventArgs e)

{ StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки в MessageBox

// и прерываем выполнение

if (\_error.Length > 0)

{ MessageBox.Show(\_error.ToString());

return;

}

try

{ string photo = ChangePhotoName();

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentItem.Photo = photo;

\_currentItem.Password = PasswordBoxNewPassword1.Password;

FermerBdEntities.GetContext().Developers.Add(\_currentItem);

FermerBdEntities.GetContext().SaveChanges();

MessageBox.Show("Регистрация прошла успешно");

this.DialogResult = true;

}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString());

return; } }

private StringBuilder CheckFields()

{ StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentItem.UserName))

s.AppendLine("Задайте имя пользователя");

if (string.IsNullOrWhiteSpace(\_currentItem.FirstName))

s.AppendLine("Поле имя пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Info))

s.AppendLine("Поле информация пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.LastName))

s.AppendLine("Поле фамилия пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Phone))

s.AppendLine("Поле телефон пустое");

if (string.IsNullOrWhiteSpace(\_currentItem.Email))

s.AppendLine("Поле email пустое");

if (string.IsNullOrWhiteSpace(\_photoName))

s.AppendLine("фото не выбрано пустое");

if (string.IsNullOrWhiteSpace(PasswordBoxNewPassword1.Password))

s.AppendLine("Введите пароль");

if (PasswordBoxNewPassword1.Password != PasswordBoxNewPassword2.Password)

s.AppendLine("Пароли не совпадают");

List<Developer> users = FermerBdEntities.GetContext().Developers.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

Developer u = users.FirstOrDefault(p => p.UserName == TbUserName.Text);

if (u != null)

{ s.AppendLine("Данный логин занят, выберите другой логин");

}

List<Client> clients = FermerBdEntities.GetContext().Clients.ToList();

//попытка найти пользователя с указанным паролем и логином

//если такого пользователя не будет обнаружено то переменная u будет равна null

Client c = clients.FirstOrDefault(p => p.UserName == TbUserName.Text);

if (c != null)

{ s.AppendLine("Данный логин занят, выберите другой логин");

}

return s; } }}

<Page x:Class="ChefApp.Pages.GoodPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:ChefApp.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Товары" IsVisibleChanged="PageIsVisibleChanged">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="\*"/>

<RowDefinition Height="40"/>

</Grid.RowDefinitions>

<DataGrid x:Name="DataGridGood" Grid.Row="0"

AutoGenerateColumns="False"

IsReadOnly="True"

SelectionMode="Single"

LoadingRow="DataGridGoodLoadingRow">

<DataGrid.Columns>

<DataGridTemplateColumn Width="100" Header="Главное изображение">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Image Source="{Binding Path=GetPhoto}" />

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn>

<DataGridTextColumn Binding="{Binding Name}" Header="Наименование товара"

Width="2\*"/>

<DataGridTextColumn Binding="{Binding Price, StringFormat='c', ConverterCulture='ru-RU'}"

Header="Стоимость" Width="\*"/>

<DataGridTextColumn Binding="{Binding Weight}" Header="Вес" Width="\*"/>

<DataGridTextColumn Binding="{Binding Length}" Header="Длина" Width="\*"/>

<DataGridTextColumn Binding="{Binding Width}" Header="Ширина" Width="\*"/>

<DataGridTextColumn Binding="{Binding Heigth}" Header="Высота" Width="\*"/>

<DataGridTextColumn Binding="{Binding Developer.GetFio}" Header="Производитель"

Width="\*"/>

<DataGridTextColumn Binding="{Binding GetStatus}" Header="Есть в наличии" Width="\*"/>

<DataGridTemplateColumn Width="auto" Header="Редактировать">

<DataGridTemplateColumn.CellTemplate>

<DataTemplate>

<Button x:Name="BtnEdit" Content="..." Click="ButtonClick" Margin="0"/>

</DataTemplate>

</DataGridTemplateColumn.CellTemplate>

</DataGridTemplateColumn> </DataGrid.Columns>

</DataGrid>

<Button x:Name="BtnAdd" Grid.Row="1" Width="100" HorizontalAlignment="Left" Margin="3"

Content="Добавить" Click="BtnAddClick"/>

<Button x:Name="BtnDelete" Grid.Row="1" Width="100" HorizontalAlignment="Right" Margin="3"

Content="Удалить" Click="BtnDeleteClick"/>

</Grid>

</Page>

using ChefApp.Models;

using System;

using System.Collections.Generic;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace ChefApp.Pages

{ /// <summary>

/// Логика взаимодействия для GoodPage.xaml

/// </summary>

public partial class GoodPage : Page

{ public GoodPage()

{ InitializeComponent();

}

private void ButtonClick(object sender, RoutedEventArgs e)

{ // открытие редактирования товара

// передача выбранного товара в AddGoodPage

Manager.MainFrame.Navigate(new AddGoodPage((sender as Button).DataContext as Good));

}

private void PageIsVisibleChanged(object sender, DependencyPropertyChangedEventArgs e)

{ //событие отображения данного Page

// обновляем данные каждый раз когда активируется этот Page

if (Visibility == Visibility.Visible)

{ DataGridGood.ItemsSource = null;

//загрузка обновленных данных

FermerBdEntities.GetContext().ChangeTracker.Entries().ToList().ForEach(p => p.Reload());

List<Good> goods = FermerBdEntities.GetContext().Goods.Where(p => p.DeveloperUserName == Manager.currentDeveloper.UserName).OrderBy(p => p.Name).ToList();

DataGridGood.ItemsSource = goods;

}

}

private void BtnAddClick(object sender, RoutedEventArgs e)

{ // открытие AddGoodPage для добавления новой записи

Manager.MainFrame.Navigate(new AddGoodPage(null));

}

private void BtnDeleteClick(object sender, RoutedEventArgs e)

{ // удаление выбранного товара из таблицы

//получаем все выделенные товары

var selectedGoods = DataGridGood.SelectedItems.Cast<Good>().ToList();

// вывод сообщения с вопросом Удалить запись?

MessageBoxResult messageBoxResult = MessageBox.Show($"Удалить {selectedGoods.Count()} записей ??? ",

"Удаление", MessageBoxButton.OKCancel, MessageBoxImage.Question);

//если пользователь нажал ОК пытаемся удалить запись

if (messageBoxResult == MessageBoxResult.OK)

{ try

{ // берем из списка удаляемых товаров один элемент

Good x = selectedGoods[0];

// проверка, есть ли у товара в таблице о продажах связанные записи

// если да, то выбрасывается исключение и удаление прерывается

if (x.OrderGoods.Count > 0)

throw new Exception("Есть записи в продажах");

//ищем записи в таблице Complect, с которой связан этот товар // удаляем товара

FermerBdEntities.GetContext().Goods.Remove(x);

//сохраняем изменения

FermerBdEntities.GetContext().SaveChanges();

MessageBox.Show("Записи удалены");

List<Good> goods = FermerBdEntities.GetContext().Goods.Where(p => p.DeveloperUserName == Manager.currentDeveloper.UserName).OrderBy(p => p.Name).ToList();

DataGridGood.ItemsSource = null;

DataGridGood.ItemsSource = goods;

}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString(), "Ошибка удаления", MessageBoxButton.OK,

MessageBoxImage.Error); } } }

private void BtnSellClick(object sender, RoutedEventArgs e)

{ // открытие страницы о продажах SellGoodsPage

// передача в него выбранного товара

//Manager.MainFrame.Navigate(new SellGoodsPage((sender as Button).DataContext as Good));

}

// отображение номеров строк в DataGrid

private void DataGridGoodLoadingRow(object sender, DataGridRowEventArgs e)

{ e.Row.Header = (e.Row.GetIndex() + 1).ToString();

} } }

<Page x:Class="ChefApp.Pages.AddGoodPage"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:local="clr-namespace:ChefApp.Pages"

mc:Ignorable="d"

d:DesignHeight="450" d:DesignWidth="800"

Title="Добавление и редактирование" FontSize="14" Loaded="Page\_Loaded"

>

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="auto"/>

<RowDefinition Height="30"/>

<RowDefinition Height="40"/>

<RowDefinition/>

<RowDefinition Height="auto"/>

</Grid.RowDefinitions>

<Grid.ColumnDefinitions>

<ColumnDefinition Width="200"/>

<ColumnDefinition/>

</Grid.ColumnDefinitions>

<Rectangle Grid.Row="0" Grid.ColumnSpan="2" />

<ComboBox x:Name="ComboCategory" SelectedItem="{Binding Category}"

DisplayMemberPath="Name" Grid.Row="2" Grid.Column="1"

SelectedValuePath="Id" />

<StackPanel Grid.Row="3" Orientation="Horizontal" Grid.Column="1">

<TextBlock Text="Вес" VerticalAlignment="Center" Padding="5" />

<TextBox Width="100" x:Name="TextBoxWeight"

Text="{Binding Weight, StringFormat={}{0:f2}, ConverterCulture='ru-RU'}"/>

<TextBlock Padding="5" Text="Длина"

VerticalAlignment="Center"

/>

<TextBox Width="100" x:Name="TextBoxLength"

Text="{Binding Length, StringFormat={}{0:f2}, ConverterCulture='ru-RU'}"/>

<TextBlock Padding="5" Text="Ширина" VerticalAlignment="Center" />

<TextBox Width="100" x:Name="TextBoxWidth"

Text="{Binding Width, StringFormat={}{0:f2}, ConverterCulture='ru-RU'}"/>

<TextBlock Padding="5" Text="Высота" VerticalAlignment="Center" />

<TextBox Width="100" x:Name="TextBoxHeight"

Text="{Binding Heigth, StringFormat={}{0:f2}, ConverterCulture='ru-RU'}"/>

</StackPanel>

<Button x:Name="BtnSave" Grid.Row="8" Content="Сохранить" HorizontalAlignment="Center"

Margin="140,5,340,0" VerticalAlignment="Top" Width="120"

Click="BtnSaveClick" Grid.Column="1"/>

<TextBlock x:Name="TextBlockGoodId" Grid.Row="0" Grid.Column="0" Text="ID:" Margin="20 0"/>

<TextBlock Grid.Row="1" Grid.Column="0" Text="Название:" Margin="20 0"/>

<TextBlock Grid.Row="2" Grid.Column="0" Text="Категория:" Margin="20 0"/>

<TextBlock Grid.Row="3" Grid.Column="0" Text="Описание товара:" Margin="20 0"/>

<TextBlock Grid.Row="4" Grid.Column="0" Text="Стоимость:" Margin="20 0"/>

<TextBlock Grid.Row="5" Grid.Column="0" Text="Производитель:" Margin="20 0"/>

<TextBlock Grid.Row="6" Grid.Column="0" Text="продаже" Margin="20 0"/>

<TextBlock Grid.Row="7" Grid.Column="0" Text="Главное изображение:"

Margin="20 0" Height="30" VerticalAlignment="Top"/>

<TextBox x:Name="TextBoxGoodId" Text="{Binding GoodId}" Grid.Row="0" Grid.Column="1"

Padding="0,2" IsReadOnly="True"/>

<TextBox x:Name="TextBoxGoodName" Text="{Binding Name}" Grid.Row="1"

Grid.Column="1" Padding="0,2"/>

<ComboBox x:Name="ComboDeveloper" SelectedItem="{Binding Developer}" Grid.Row="5"

Grid.Column="1" DisplayMemberPath="GetFio"

SelectedValuePath="UserName" />

<Image x:Name="ImagePhoto" Source="{Binding GetPhoto}" Grid.Row="7" Grid.Column="1"

VerticalAlignment="Stretch" HorizontalAlignment="Stretch"/>

<Button x:Name="BtnLoad" Grid.Row="7" Grid.Column="0" Margin="20,30,0,0"

HorizontalAlignment="Left" VerticalAlignment="Top"

Click="BtnLoadClick" Content="Загрузить"/>

<TextBox x:Name="TextBoxPrice" Text="{Binding Price, StringFormat='c', ConverterCulture='ru-RU'}"

Grid.Row="4" Grid.Column="1" Padding="0,2"/>

<CheckBox x:Name="CheckBoxActive" Grid.Row="6" Grid.Column="1"

VerticalAlignment="Center" Margin="20,12,0,12" IsChecked="{Binding Active}"/>

</Grid>

</Page>

using ChefApp.Models;

using Microsoft.Win32;

using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

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.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace ChefApp.Pages

{ /// <summary>

/// Логика взаимодействия для AddGoodPage.xaml

/// </summary>

public partial class AddGoodPage : Page

{ //текущий товар

private Good \_currentGood = new Good();

// путь к файлу

private string \_filePath = null;

// название текущей главной фотографии

private string \_photoName = null;

// текущая папка приложения

private static string \_currentDirectory = Directory.GetCurrentDirectory() + @"\Images\";

// передача в AddGoodPage товара

public AddGoodPage(Good selectedGood)

{ InitializeComponent();

// если передано null, то мы добавляем новый товар

if (selectedGood != null)

{ \_currentGood = selectedGood;

TextBoxGoodId.Visibility = Visibility.Hidden;

TextBlockGoodId.Visibility = Visibility.Hidden;

int x = selectedGood.Id;

// загрузка комплементарных товаров \_filePath = \_currentDirectory + \_currentGood.Photo;

}

ComboDeveloper.IsEnabled = false;

DataContext = \_currentGood;

\_photoName = \_currentGood.Photo;

//загрузка производителей

ComboDeveloper.ItemsSource = FermerBdEntities.GetContext().Developers.ToList();

ComboCategory.ItemsSource = FermerBdEntities.GetContext().Categories.ToList();

} // проверка полей private StringBuilder CheckFields()

{ StringBuilder s = new StringBuilder();

if (string.IsNullOrWhiteSpace(\_currentGood.Name))

s.AppendLine("Поле название пустое");

if (\_currentGood.Developer == null)

s.AppendLine("Выберите производителя");

if (\_currentGood.Price < 0)

s.AppendLine("Цена не может быть отрицательной");

if (!string.IsNullOrWhiteSpace(TextBoxHeight.Text))

{ double x = 0;

if (!double.TryParse(TextBoxHeight.Text, out x))

s.AppendLine("Высота только число");

else if (x < 0)

{ s.AppendLine("Высота не может быть отрицательной"); } }

if (!string.IsNullOrWhiteSpace(TextBoxLength.Text))

{ double x = 0;

if (!double.TryParse(TextBoxLength.Text, out x))

s.AppendLine("Длина только число");

else if (x < 0)

{ s.AppendLine("Длина не может быть отрицательной");

}

} if (!string.IsNullOrWhiteSpace(TextBoxWidth.Text))

{ double x = 0;

if (!double.TryParse(TextBoxWidth.Text, out x))

s.AppendLine("Ширина только число");

else if (x < 0)

{ s.AppendLine("Ширина не может быть отрицательной");

}

}

if (!string.IsNullOrWhiteSpace(TextBoxWeight.Text))

{ double x = 0;

if (!double.TryParse(TextBoxWeight.Text, out x))

s.AppendLine("Вес только число");

else if (x < 0)

{ s.AppendLine("Вес не может быть отрицательным");

}

}

if

(string.IsNullOrWhiteSpace(\_photoName))

s.AppendLine( "фото не выбрано пустое ");

return s;

}

// сохранение

private void BtnSaveClick(object sender, RoutedEventArgs e)

{ StringBuilder \_error = CheckFields();

// если ошибки есть, то выводим ошибки

// и прерываем выполнение

if (\_error.Length > 0)

{ MessageBox.Show(\_error.ToString());

return

;

}

\_currentGood.DeveloperUserName = Manager.currentDeveloper.UserName;

// проверка полей прошла успешно

if(\_currentGood.Id == 0)

{ // добавление нового товара

// формируем новое название файла картинки,

// так как в папке может быть файл с тем же именем

string photo = ChangePhotoName();

// путь куда нужно скопировать файл

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentGood.Photo = photo;

// добавляем товар FermerBdEntities.GetContext().Goods.Add(\_currentGood);

}

try

{ if (\_filePath != null)

{ string photo = ChangePhotoName();

string dest = \_currentDirectory + photo;

File.Copy(\_filePath, dest);

\_currentGood.Photo = photo;

}

// Сохраняем изменения в БД

FermerBdEntities.GetContext().SaveChanges();

MessageBox.Show("Запись Изменена");

// Возвращаемся на предыдущую форму

Manager.MainFrame.GoBack();

}

catch (Exception ex)

{ MessageBox.Show(ex.Message.ToString());

}

} // загрузка фото

private void BtnLoadClick(object sender, RoutedEventArgs e)

{ try

{ //Диалог открытия файла

OpenFileDialog op = new OpenFileDialog();

op.Title = "Select a picture";

op.Filter = "JPEG Files (\*.jpeg)|\*.jpeg|PNG Files (\*.png)|\*.png|JPG Files (\*.jpg)|\*.jpg|GIF Files(\*.gif) | \*.gif";

// диалог вернет true, если файл был открыт

if (op.ShowDialog() == true)

{ // проверка размера файла

// по условию файл дожен быть не более 2Мб.

FileInfo fileInfo = new FileInfo(op.FileName);

if (fileInfo.Length > (1024 \* 1024 \* 2))

{ // размер файла меньше 2Мб. Поэтому выбрасывается новое исключение

throw new Exception("Размер файла должен быть меньше 2Мб");

}

ImagePhoto.Source = new BitmapImage(new Uri(op.FileName));

\_photoName = op.SafeFileName;

\_filePath = op.FileName;

}

}

catch (Exception ex)

{ MessageBox.Show(ex.Message, "Ошибка", MessageBoxButton.OK, MessageBoxImage.Error);

\_filePath = null;

}

}

//подбор имени файла

string ChangePhotoName()

{ string x = \_currentDirectory + \_photoName;

string photoname = \_photoName;

int i = 0;

if (File.Exists(x))

{ while (File.Exists(x))

{ i++;

x = \_currentDirectory + i.ToString() + photoname;

}

photoname = i.ToString() + photoname;

}

return photoname;

}

private void Page\_Loaded(object sender, RoutedEventArgs e)

{ ComboDeveloper.SelectedValue = Manager.currentDeveloper.UserName;

}

}}

}

**Приложение Б**

«Служба доставки еды «Шеф»»

Спецификация

460.ЗМК.00008-01 01

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| |  | | --- | | **Обозначение** | | |  | | --- | | **Наименование** | | |  | | --- | | **Примечание** | |
|  | |  | | --- | | Документация | |  |
| 460.ЗМК.00008-01 12 01 | «Служба доставки еды «Шеф»» |  |
|  | Текст программы |  |
| 460.ЗМК.00008-01 13 01 | «Служба доставки еды «Шеф»» |  |
|  | Описание программы |  |
| 460.ЗМК.00008-01 34 01 | «Служба доставки еды «Шеф»» |  |
|  | Руководство пользователя |  |
| 460.ЗМК.00008-01 81 01 | «Служба доставки еды «Шеф»» |  |
|  | Пояснительная записка |  |

**Приложение B**

**Диск с программным средством**