**ПРАВОСЛАВНЫЙ СВЯТО-ТИХОНОВСКИЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ**

**Факультет информатики и прикладной математики**

**Кафедра Информатики**

**Отчет**

**по курсовой работе по теме:**

**«Программный комплекс «ВкусноЕшка» для ИП «Кормим как дома»»**

Выполнили: студенты группы ИПМ 1-01-08

Дедова Мария, Довнар Вацлав, Медведев Никита, Михайлова Мария

Руководитель: доцент Пуцко Николай Николаевич

Москва 2013

**Оглавление**

[1. Постановка задачи 4](#_Toc367899194)

[2. Оценка предварительной стоимости 5](#_Toc367899195)

[2.1 Расчет трудоемкости проекта 5](#_Toc367899196)

[2.2 Расчет длительности проекта 6](#_Toc367899197)

[2.3 Расчет числа разработчиков 7](#_Toc367899198)

[2.4 Расписание выполнения этапов проекта и вычисление общих затрат на персонал 8](#_Toc367899199)

[2.5 Вывод 9](#_Toc367899200)

[3. Техническое задание 10](#_Toc367899201)

[3.1 Цель работы 10](#_Toc367899202)

[3.2 Содержание работы 10](#_Toc367899203)

[3.3 Предполагаемые результаты работы 12](#_Toc367899204)

[4. Календарный план 13](#_Toc367899205)

[5. Описание программы 16](#_Toc367899206)

[5.1 Общие сведения 16](#_Toc367899207)

[5.2 Функциональное назначение 16](#_Toc367899208)

[5.3 Описание логической структуры 16](#_Toc367899209)

[2.4 Входные и выходные данные 16](#_Toc367899210)

[6. Пояснительная записка 17](#_Toc367899211)

[6.1 Введение 17](#_Toc367899212)

[6.2 Назначение и область применения 17](#_Toc367899213)

[6.3 Технические характеристики 17](#_Toc367899214)

[6.4 Ожидаемые технико-экономические показатели 17](#_Toc367899215)

[Источники, используемые при разработке АРМа 18](#_Toc367899216)

[7. Руководство программиста 19](#_Toc367899217)

[7.1 Необходимое техническое обеспечение 19](#_Toc367899218)

[7.2 Необходимое программное обеспечение для инсталляции 19](#_Toc367899219)

[7.3 Необходимое программное обеспечение для разработки 19](#_Toc367899220)

[СТРУКТУРА ПРОГРАММЫ 19](#_Toc367899221)

[Установка и настройка программы 19](#_Toc367899222)

[ПРОВЕРКА ПРОГРАММЫ 19](#_Toc367899223)

[Структура базы данных. 19](#_Toc367899225)

[Структура программы. 20](#_Toc367899226)

[Интрфейсы View 21](#_Toc367899227)

[View 22](#_Toc367899228)

[Controller 22](#_Toc367899229)

[Model 23](#_Toc367899230)

[8. Руководство пользователя 26](#_Toc367899231)

[8.1. Назначение программы 26](#_Toc367899232)

[8.2. Условия выполнения программы 26](#_Toc367899233)

[8.3. Выполнение программы 26](#_Toc367899234)

# Постановка задачи

Требуется разработать и реализовать программный комплекс для фирмы, занимающейся приготовлением и доставкой обедов в офисы, который позволит достичь следующих целей:

* + Сократить временные задержки в передаче актуальной информации, касающейся заказов, между всеми сотрудниками компании.
  + Упростить процесс обработки заказов от компании-клиента.
  + Эффективно управлять ценообразованием.
  + Упростить процесс создания новых блюд и наборов блюд по заданным критериям.
  + Оптимизировать расходы, закупая только необходимые продукты.
  + Получение финансовой статистики.

Для достижения поставленных целей требуется решить следующие задачи:

* Аутентификация и контроль прав доступа пользователей.
* Организация удобного и наглядного способа формирования, корректировки и выдачи заказов.
* Добавление, возможность редактирования и удаления данных о компании-клиенте, а так же их сотрудниках.
* Ведение истории заказов каждого сотрудника компании-клиента.
* Реализация формирования меню на несколько дней.
* Реализация возможности создания комплексного обеда на день.
* Возможность создания и изменения блюд в меню.
* Получение сведений о блюдах в меню:
  + количество продуктов, необходимое для приготовления блюда;
  + сколько уже есть приготовленных блюд;
  + сколько блюд можно приготовить из тех продуктов, что есть в данный момент на сладе.
* Получение сведений о количестве продуктов на складе
* Формирование списка продуктов для закупки на основании сделанных заказов
* Разработка графического интерфейса приложения администратора системы, позволяющего:
  + работать с учётными записями пользователей
  + просмотр статистики по следующим направлениям:
  + формирование цены и наценок

# Оценка предварительной стоимости

Целью данного раздела является обоснование затрат на персонал и отражение формул и показателей, которые были для этого использованы. Расчеты производились по стандартам методики COCOMO II, являющимися на сегодняшний день наиболее объективными для предварительной оценки проекта. С помощью данной методики были рассчитаны следующие показатели:

* Трудоемкость проекта
* Длительность проекта
* Количество разработчиков

Далее приводится распределение разработчиков по месяцам выполнения проекта.

## 2.1 Расчет трудоемкости проекта

Для оценки трудозатрат на базовом уровне модели COCOMO II применяется следующая формула:



После подстановки соответствующих коэффициентов было получено следующее число:

PM= 2,94\*22,75422\*0,62557785 = 41,84955159 [человеко-месяцев]

Человеко-месяц – это 19 рабочих дней в месяце (или 152 рабочих часа в месяц)

Далее приведено описание показателей, с помощью которых было получено это число:

A=2,94 и B = 0,91 – данные коэффициенты являются константами модели COCOMO II.

SIZE = 17

KSLOC – параметр, показывающий количество строк кода, которое будет написано в данном проекте (KSLOC – 1,000 Source Lines of Code)

Формула расчета степени для SIZE:



*Е* = 0,91+0,01\*19,29 = 1,1029;

В методике COCOMO II используются пять факторов масштаба SFj, которые определяются некоторыми характеристиками проекта. В следующей таблице указаны этих характеристики, приведен диапазон значений, которые принимают данные параметры. В поле «Значение» указано число из диапазона, которое применимо для данного проекта, и в поле «Примечание» обосновывается выбор этого числа:

*Табл 1.Детализация фактора масштаба*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Факторы масштаба ( SFj )*** | ***Диапазон изменений*** | ***Выбранное значение*** | ***Комментарий*** |
| ***Прецедентность (PREC)*** | 0 - 6,20 | 2,48 | Команда уже имела опыт подобных разработок |
| ***Гибкость процесса разработки (FLEX)*** | 0 - 5,07 | 3,04 | Процесс разработки не детерминирован |
| ***Архитектура и разрешение рисков (RESL)*** | 0 - 7,07 | 4,24 | Архитектура и разрешение рисков на обычном уровне |
| ***Сработанность команды (TEAM )*** | 0 - 5,48 | 3,29 | Хорошая сработанность команды |
| ***Зрелость процессов (PMAT)*** | 0 - 7,80 | 6,24 | Зрелость процессов низкая |

Для предварительной оценки трудоемкости программного проекта по методу COCOMO II были установлены семь множителей трудоемкости EMi:

*Табл 2.Множители трудоемкости*

|  |  |  |  |
| --- | --- | --- | --- |
| ***Множители трудоёмкости ( EMi )*** | ***Диапазон изменений*** | ***Выбранное значение*** | ***Комментарий*** |
| ***Квалификация персонала (PERS)*** | 0,5 - 2,12 | 1 | Квалификация персонала высокая |
| ***Сложность и надёжность (RCPX)*** | 0,49 - 2,72 | 1 | Сложность продукта обычная, но требуется высокая надёжность |
| ***Повторное использование (RUSE)*** | отсутствует либо 0,95 - 1,24 | 0,95 | Повторное использование не предполагается |
| ***Сложность платформы разработки (PDIF)*** | отсутствует либо 0,87 - 2,61 | 0,87 | Нет жёстких ограничений по памяти и быстродействию - сложность платформы низкая |
| ***Опыт персонала (PREX)*** | 0,62 - 1,59 | 0,87 | Опыт персонала в платформе и среде разработки высокий |
| ***Оборудование (FCIL)*** | 0,62 - 1,43 | 0,87 | Современные средства разработки, высокая степень коммуникации |
| ***Сжатие расписания (SCED)*** | отсутствует либо 1 - 1,43 | 1 | Сжатие расписание не предполагается |

## 2.2 Расчет длительности проекта

Длительность проекта в методике COCOMO II рассчитывается по формуле:

TDEV = C × (PMNS) D + 0,2 × 0,01 × ∑SFj × SCED / 100

TDEV = 3,67 × (41,84955159)0,28 + 0,2 \* 0,01 \* 19,29 = 12,05880638

С = 3,67, D = 0,28 – данные коэффициенты являются константами модели COCOMO II

PMns – расчет трудоемкости без учета множителя трудоемкости SCED. Но так как в нашем проекте сжатие расписания отсутствует, на что указывает SCED = 1, в процентном соотношении SCED =100% . Следовательно, PMns=PM.

## 2.3 Расчет числа разработчиков

Для расчета числа разработчиков по COCOMO II используется следующая формула:

N=PM/TDEV, где

PM – трудоемкость проекта

TDEV – длительность проекта

N = 41,84955159/12,05880638 = 3,470455554 [человек]

## 2.4 Расписание выполнения этапов проекта и вычисление общих затрат на персонал

*Табл.3. Зарплата сотрудников*

|  |  |
| --- | --- |
| Менеджер/Lead | 40000 |
| Программист | 25000 |
| Документатор | 10000 |
| Тестировщик | 10000 |
| Дизайнер | 15000 |

*Табл. 4. Этапы разработки.*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Ставка за месяц | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | Всего за проект | Зарплата за проект |
| 40000 | 1,00 | 1,00 | 1,00 | 0,90 | 1,00 | 0,90 | 0,80 | 0,70 | 0,90 | 1,00 | 1,00 | 0,90 | 11,10 | 444000 |
| 25000 | 0,50 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 1,00 | 0,80 | 0,90 | 1,00 | 1,00 | 11,20 | 280000 |
| 10000 | 1,00 | 0,50 | 0,70 | 0,50 | 1,00 | 0,80 | 0,60 | 0,70 | 1,00 | 0,80 | 0,70 | 0,90 | 9,20 | 92000 |
| 10000 | 0,00 | 0,40 | 1,00 | 1,00 | 1,00 | 0,60 | 0,70 | 0,70 | 0,50 | 0,70 | 0,70 | 0,60 | 7,90 | 79000 |
| 15000 |  |  |  |  | 0,30 | 1,00 | 1,00 | 0,10 | 0,10 |  |  |  | 2,50 | 37500 |
|  | 2,50 | 2,90 | 3,70 | 3,40 | 4,00 | 3,30 | 3,10 | 3,10 | 3,20 | 3,40 | 3,40 | 3,40 | 41,90 |  |
|  | 1,00 | 0,60 | -0,20 | 0,10 | -0,50 | 0,20 | 0,40 | 0,40 | 0,30 | 0,10 | 0,10 | 0,10 | 0,00 |  |
|  | Анализ, разработка ТЗ и КП | Реализация прототипа, документирование, тестирование, взаимодействие с заказчиком | | Документирование, тестирование ядра программы | Показ прототипа заказчику | Доработка интерфейсов, дополнительный функционал, usability, закрытое тестирование | | Доработка интерфейсов, дополнительный функционал, usability, закрытое тестирование | | Предвыпускное тестирование, доработка продукта | | Сдача заказа |  |  |

## 2.5 Вывод

Проект может быть реализован за 12 месяцев коллективом из 4 разработчиков, занятых на полный рабочий день.

В реализации проекта будут задействованы 5 сотрудников. В вышеприведённой Таблице 4 суммарная нагрузка на сотрудников эквивалентна рассчётной ( то есть их труд эквивалентен труду 4 разработчиков ).

Стоимость проекта вычисляется на основе зарплат сотрудников и количества затраченных ими на проект человеко/месяцев.

Общая стоимость проекта 932 500 условных единиц.

# 3. Техническое задание

## 3.1 Цель работы

Разработать и реализовать программный комплекс, который позволит достичь следующих целей:

* + Сократить временные задержки в передаче актуальной информации, касающейся заказов, между всеми сотрудниками компании (менеджер, курьер, поставщик продуктов, повар).
  + Упростить процесс обработки заказов от компании-клиента (комплектация заказов у сотрудников, редактирование и выдача заказов).
  + Эффективно управлять ценообразованием (сравнение себестоимости и цен продуктов, контроль цен по времени)
  + Упростить процесс создания новых блюд и наборов блюд по заданным критериям (по цене и наличию ингредиентов).
  + Оптимизировать расходы, закупая только необходимые продукты (на основании сделанных заказов).
  + Получение финансовой статистики:
    - сведения о заказах компаний за период (что заказывали, количество замен и возвратов);
    - популярные блюда за период;
    - наиболее прибыльные блюда за период.

## 3.2 Содержание работы

Основные задачи, реализуемые в программных модулях, включают в себя:

Система аутентификации пользователей. Предусмотрены следующие роли:

* Администратор программного комплекса
* Курьер
* Повар
* Помощник повара (фасовщик)
* Менеджер по закупкам продуктов

Подсистема **“Курьер”** позволяет решать следующие задачи:

* организация удобного и наглядного способа формирования, корректировки и выдачи заказов
* добавление и редактирование данных о фирме (название, адрес и список сотрудников)

Подсистема **“Повар”** позволяет решать следующие задачи:

* формирование меню на день
* добавление нового блюда, с возможностью согласования его с администратором
* получение сведений о блюдах в меню:
  + необходимое количество продуктов
  + сколько приготовлено
  + сколько можно приготовить из остатков на складе

Подсистема **“Фасовщик”** позволяет решать следующие задачи:

* получение сведений о типе и количестве заказанных блюд для каждой компании
* о блюдах в виде ингредиентов удобного представления готовых блюд, сгруппированных по фирмам-заказчикам для дальнейшей фасовки и транспортировки

Подсистема **“Закупщик”** позволяет решать следующие задачи:

* получение сведений о количестве продуктов на складе
* формирование списка продуктов для закупки на основании сделанных заказов

Подсистема **“Администратор"** позволяет решать следующие задачи:

* управление пользователями и привилегиями:
  + добавление и удаление пользователей
  + назначение прав пользователя
* просмотр статистики по следующим направлениям:
  + наиболее часто заказываемые блюда в данной компании
  + компании, которые часто меняют свои заказы (после его выбора)
* формирование цены и наценки:
  + установка наценки на группы товаров
  + возможность установки отдельной цены на блюдо
  + просмотр текущей себестоимости каждого блюда
* Разработка необходимой документации к программному обеспечению.
* Проведение тестирования разработанного программного обеспечения.
* Внедрение разработанного программного продукта и его сопровождение.

## 3.3 Предполагаемые результаты работы

Предполагаемыми результатами выполнения работ являются:

* + Переносимое клиентское приложение.
  + Пакет документации к разработанному программному продукту.
  + Пакет исходных кодов.

# Календарный план

Наименование работ:

**ООО “Вкусноешка”**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№ п/п** | **Наименование этапов работ** | **Сроки выполнения начало-окончание** | **Расчетная цена этапа** | **Форма и вид отчетности** |
| 1 | I. Анализ.  II. Проектирование  III. Разработка пакета документации: ТЗ, Календарный план, описание технологических процессов.  IV. Разработка прототипа, включающего в себя:  1. Разработка структуры БД, для хранения информации о:   * продуктов (стоимость, наличие) * фирм и их сотрудников * блюд (состав, цена, себестоимость) * история заказов сотрудников компаний-клиентов   2. Создание ядра программы, костяк, который позволит вести независимою и совместную разработку программы разработчиками.  3. Защита доступа к системе. Разработка модуля аутентификация и контроля прав доступа пользователей.  4. Организация рабочего места курьера, которое включает следующий функционал:   * формирования, корректировки и выдачи заказов * добавление и редактирование данных о фирме (название, адрес и список сотрудников)   5. Организация рабочего места повара, которое включает следующий функционал:   * формирование меню на текущий рабочий день * добавление нового блюда с отображением себестоимости. * отображение количества блюд, которые повар может приготовить, исходя из наличия продуктов на складе.   6. Организация рабочего места фасовщика, которое включает следующий функционал:   * получение информации о типе и количестве заказанных блюд для каждой компании * информации о блюдах сгруппированных по фирмам-заказчикам для дальнейшей фасовки и транспортировки   7. Организация рабочего места закупщика, которое включает следующий функционал:   * получение сведений о количестве продуктов на складе * формирование списка продуктов для закупки на основании сделанных заказов   8. Организация рабочего места администратора, которое включает следующий функционал:   * управление пользователями и привилегиями:   + добавление и удаление пользователей   + назначение прав пользователя * просмотр статистики по следующим направлениям:   + наиболее часто заказываемые блюда в данной компании   + компании, которые часто меняют свои заказы (после его выбора) * формирование цены и наценки:   + установка наценки на группы товаров   + возможность установки отдельной цены на блюдо   + просмотр текущей себестоимости каждого блюда   9. Тестирование ядра  10. Отшлифовка проекта  11. Создание документации | 07.02.2013-  07.06.2013 | 384000 | Пакет документации.  Программное и информационное  обеспечение. |
| 2 | Разработка полного функционала:   1. Разработка календаря 2. Доработка функционала пользователей с учетом наличия календаря 3. Разработка возможности осуществления заказа не только на след. день, но и на любой день и месяц вперед. 4. Разработка функции позволяющей продумывать и предлагать разное меню на каждую неделю 5. Разработка функции позволяющей прогнозировать необходимые продукты до получения полного заказа 6. Реализация функции позволяющей просматривать блюда, которые не пользуются спросом 7. Реализация функции в интерфейсе администратора, позволяющей удалять все данные о фирме из БД по расторжении договора 8. Разработка возможность создания блюд с ориентировкой на продукты, которые есть на складе и нуждаются в использовании | 15.07.2013—  15.10.2014 | 472500 | Программное обеспечение. |
| 3 | Ограниченное тестирование:  сбор сведений от первых пользователей, дополнение продукта  коммерческое внедрение продукта  Сопровождение продукта | 15.01.2014—  15.02.2014 | 76000 | список с отзывами пользователей, список изменений |

# 5. Описание программы

## 5.1 Общие сведения

Программный комплекс «ВкусноЕшка» разработан для организации ИП «Кормим как дома» в рамках курсовой работы по ТРПО группой № 2 в средах разработки: Microsoft Visual Studio 2010, Microsoft Access.

## 5.2 Функциональное назначение

Данное приложение предназначено для автоматизации процесса обработки приема и выдачи заказов от фирм-клиентов, а так же для упрощения и систематизации работы сотрудников данной организации.

## 5.3 Описание логической структуры

Программный комплекс представляет собой клиентское приложение с подключаемой к нему базой данных.

Приложение обеспечивает:

* Простоту и удобство эксплуатации
* Безопасность хранения данных
* … принцип контроля доступом

## 2.4 Входные и выходные данные

Для авторизации пользователя необходимо ввести:

* логин
* пароль

# 6. Пояснительная записка

## 6.1 Введение

Заказчик: индивидуальный предприниматель «Кормим как дома».

Название проекта: программный комплекс «ВкусноЕшка».

Разработчики: рабочая группа из четырех человек. Ведущий программист и дизайнер Медведев Н. А., менеджер и тестировщик Довнар В. ., документаторы Дедова М. А. и Михайлова М. А.

## 6.2 Назначение и область применения

Данный программный комплекс предназначен для систематизации процессов сбора и выдачи заказов от фирмы-клиента как непосредственно в офисе, так и вне его. Так же данное приложение позволит сократить время обмена необходимой информацией для сбора, приготовления и выдачи заказов между сотрудниками организации.

## 6.3 Технические характеристики

Объектом автоматизации является процесс приема и выдачи заказов.

Система должна обеспечивать:

* Корректное хранение информации в базе данных: о фирмах-клиентах, их сотрудниках и заказах каждого из них; о предоставляемых в заказе блюдах (состав, количество); и о сотрудниках данной организации (ФИО и должность).
* Проведение системы аутентификации для каждого сотрудника организации с назначениями прав доступа.

## 6.4 Ожидаемые технико-экономические показатели

- Система позволит оперативно выполнять различные операции с имеющейся базой данной (подробнее об операциях см. в Техническом задании).

- АРМ увеличивает мобильность предприятия в случае переезда, открытия нового филиала и т.п.

- АРМ увеличивает эффективность сотрудников, автоматизируя их повседневные и рутинные задачи передачи информации о выполненных задачах коллегам.

- АРМ автоматизирует сбор статистики о клиентах и количестве предоставляемых услуг.

- Система централизует хранение информации и переводит ее в электронный вид, увеличивая тем самым простоту архивирования и степень доступности

* 1. Система позволяет сократить временные задержки в передаче актуальной информации, касающейся заказов, между всеми сотрудниками компании (менеджер, курьер, поставщик продуктов, повар).
  2. Система позволяет упростить процесс обработки заказов от компании-клиента (комплектация заказов у сотрудников, редактирование и выдача заказов).
  3. Система позволяет эффективно управлять ценообразованием (сравнение себестоимости и цен продуктов, контроль цен по времени)
  4. Система позволяет упростить процесс создания новых блюд и наборов блюд по заданным критериям (по цене и наличию ингредиентов).
  5. Система позволяет оптимизировать расходы, закупая только необходимые продукты (на основании сделанных заказов).
  6. Система позволяет получить финансовую статистику:
     1. сведения о заказах компаний за период (что заказывали, количество замен и возвратов);
     2. популярные блюда за период;
     3. наиболее прибыльные блюда за период.

## Источники, используемые при разработке АРМа

1) *Для определения схемы хранения информации, объектов подлежащих программной реализации :* Групповые обсуждения бизнес-процессов компании – заказчика

2) *Для получения технической информации по средствам реализации программного комплекса:*

* <http://msdn.microsoft.com>,
* Джозеф Албахари, Бен Албахари ­– C# 3.0. Справочник. БХВ-Петербург. 2012
* <http://habrahabr.ru/hub/csharp/>
* лекции по курсу "Теория разработки программного обеспечения". Н.Н. Пуцко.
* <http://git-scm.com/book/ru>
* <http://ru.wikipedia.org/wiki/Категория:.NET_Framework>
* <http://ru.wikipedia.org/wiki/Категория:Ключевые_слова_SQL>
* <http://habrahabr.ru/post/60347/>

# 7. Руководство программиста

## 7.1 Необходимое ****техническое**** обеспечение

* 1 Гб оперативной памяти и выше
* Процессор 1.7 Ghz и выше
* 50 Мб свободного места на диске

## 7.2 Необходимое программное обеспечение для инсталляции

* ОС: MS Window 7, Windows 2008 Server 32/64-бит, MS Window 8, Windows Server 2012.
* .Net Framework 4.0 и выше

## 7.3 Необходимое программное обеспечение для разработки

* MS Visual Studio 2010
* MS Access 2010

## СТРУКТУРА ПРОГРАММЫ

Структурно система разделена на модули, отвечающие за выполнение конкретных функций.

Модуль авторизации отвечает за авторизацию и аутентификацию пользователей

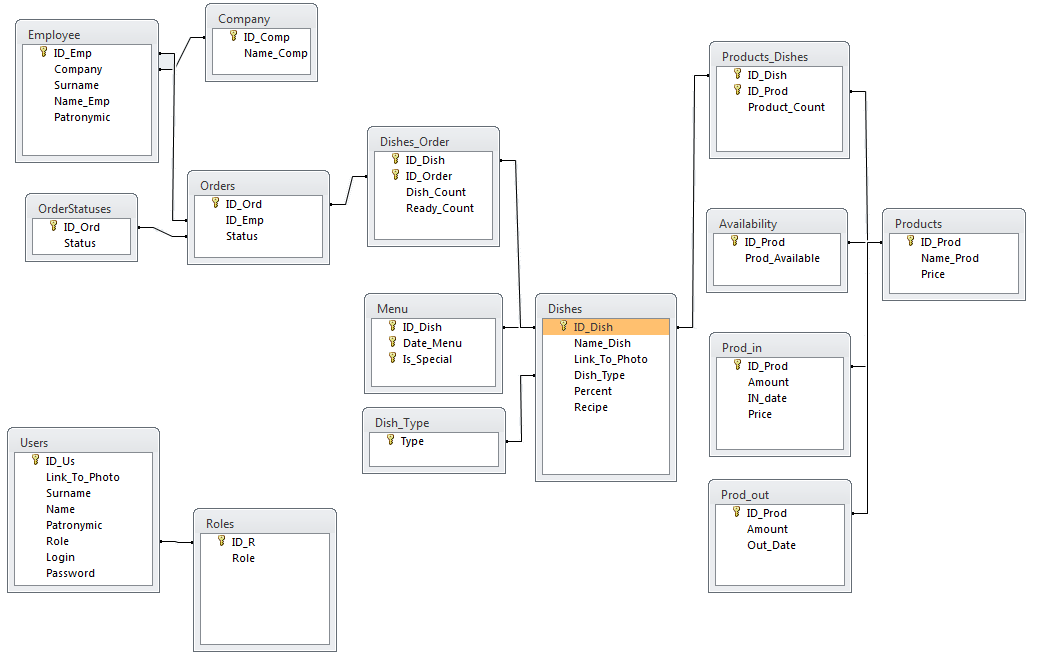
Остальные модули: Курьер, Закупщик, Повар, Администратор являются автоматизированными рабочими местами для сотрудников занимающих соответствующие должности в компании.

## Установка и настройка программы

Установка заключается в копировании директории с программой на компьютер пользователя. Дополнительная настройка не требуется.

## ПРОВЕРКА ПРОГРАММЫ

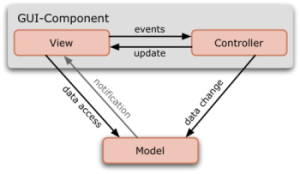
## Структура базы данных.



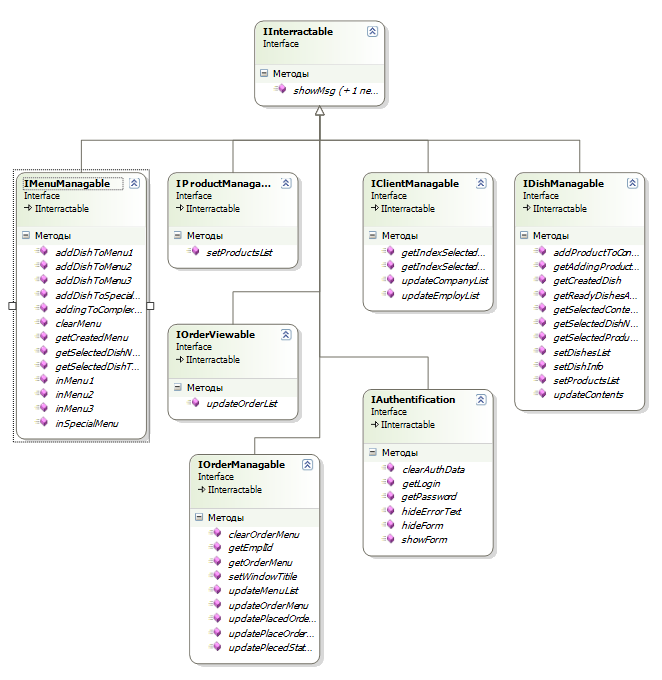
## Структура программы.

Программа построена на основе паттерна VMC для настольных приложений.

Наглядно иллюстрирует поток управления следующий рисунок:



Концепция MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента:

* *Модель* (англ. *Model*). Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
* *Представление*, *вид* (англ. *View*). Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма (окно) с графическими элементами.
* *Контроллер* (англ. *Controller*). Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

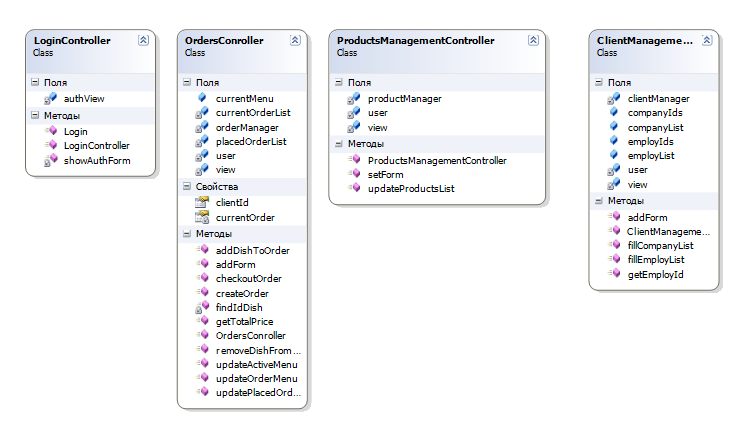
Интрфейсы View

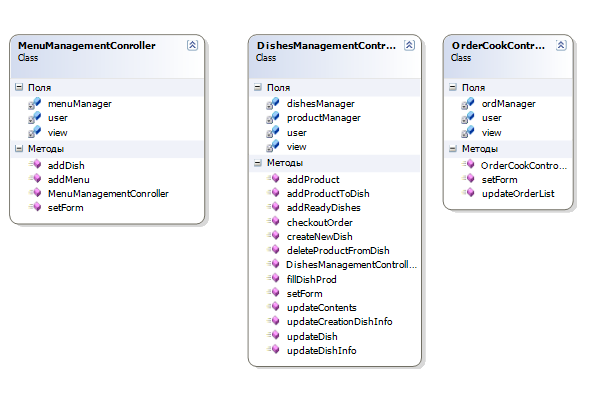
На данном рисунке показаны все интерфейсы, используемые в программе для модулей типа View. Т.е. все модули типа View пронаследованы от одного или более интерфейсов. Все классы типа Conroller могут обращаться к View только посредством интерфейсов. Таким образом любой квалифицированный программист может полностью изменить внешний вид программы правильно реализовав все описаны выше интерфейсы. Работоспособность в этом случае будет гарантирована архитектурой программы.

View

View – классы не приводятся в данном руководстве из-за свое громоздкости. Подробнее об их структуре вы можете узнать в исходном коде приложения в соотвуствующих файлах в директории …/src/View/

Controller





На данных рисунках приводятся схема controller-классов, которые содержат своими полями экземпляры типа View. Все методы котроллеров не имеют аргументов и имеют тип void.

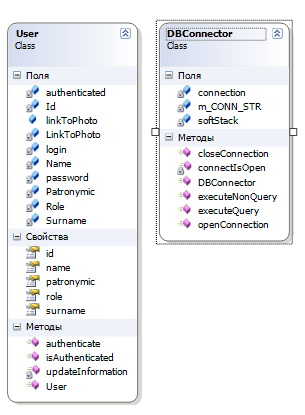
Model





Классы из раздела Model имеют как правило только одно поле – connector экземпляр класса DBConnector – предоставляющего непосредственный доступ к БД. Все методы классов реализуют какие-либо sql-запросы.

Отдельно стоит упоминать классы User и DBConnector:

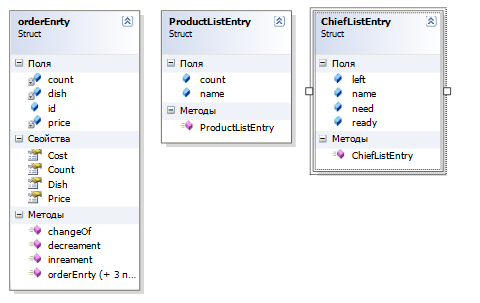


DBConnector реализует методы executeNoneQuery & ExecuteQuery позволяющие другим классам типа Model делать sql – запросы типа INSERT, DELETE, UPDATE, SELECT.

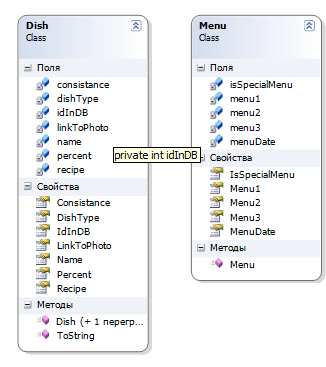
Класс User является исключением из общей архитектуры приложения и одновременно делает запросы к БД и предоставляет информацию во View.

Структуры, классы-типы данных, перечисления, классы общие для всех модулей.

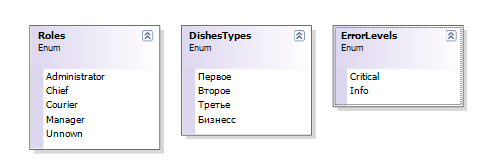
Структуры:



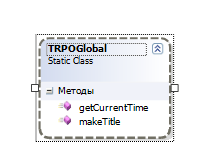
Классы, которые используются как типы данных:



Перечисления:



Класс содержащий глобальные методы, доступные всем остальным классам:



Обобщённый граф передачи потока управления в программе.

# 8. Руководство пользователя

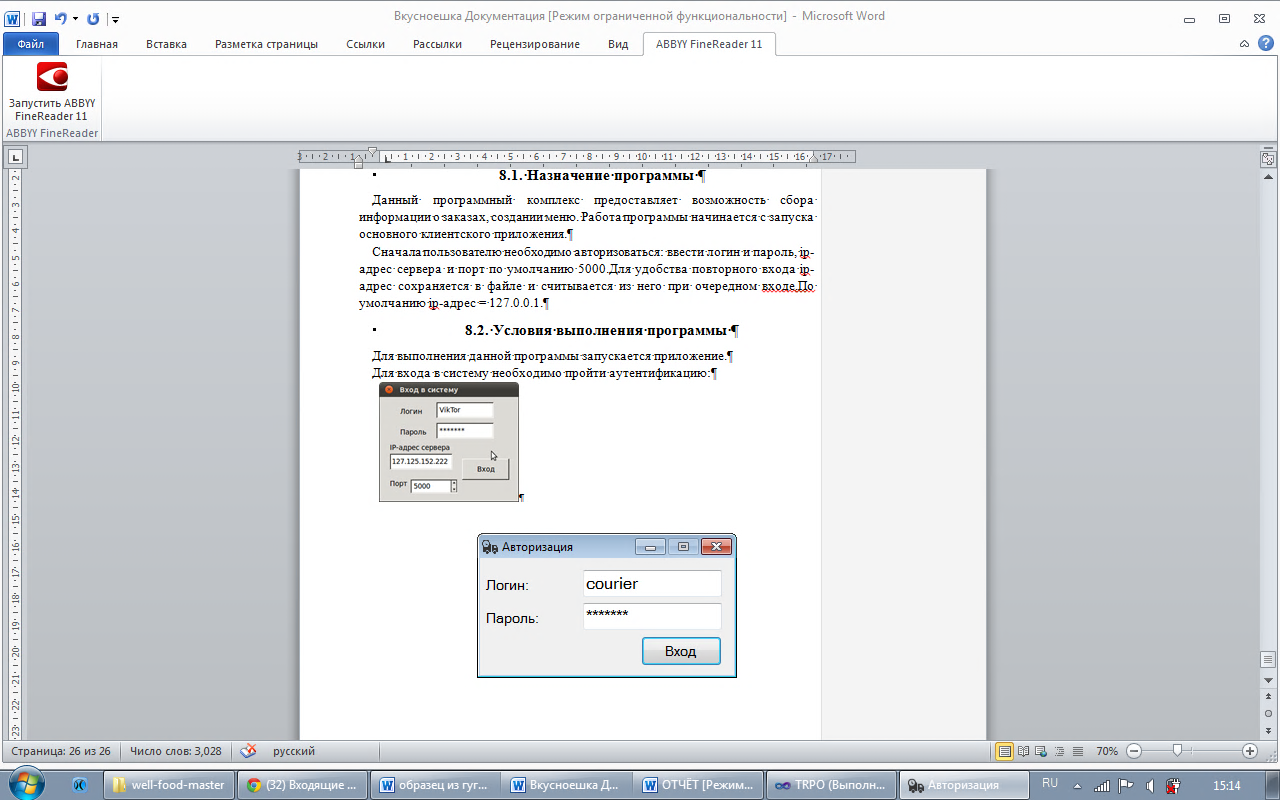
## 8.1. Назначение программы

Данный программный комплекс предоставляет возможность сбора информации о заказах, создании меню. Работа программы начинается с запуска основного клиентского приложения.

## 8.2. Условия выполнения программы

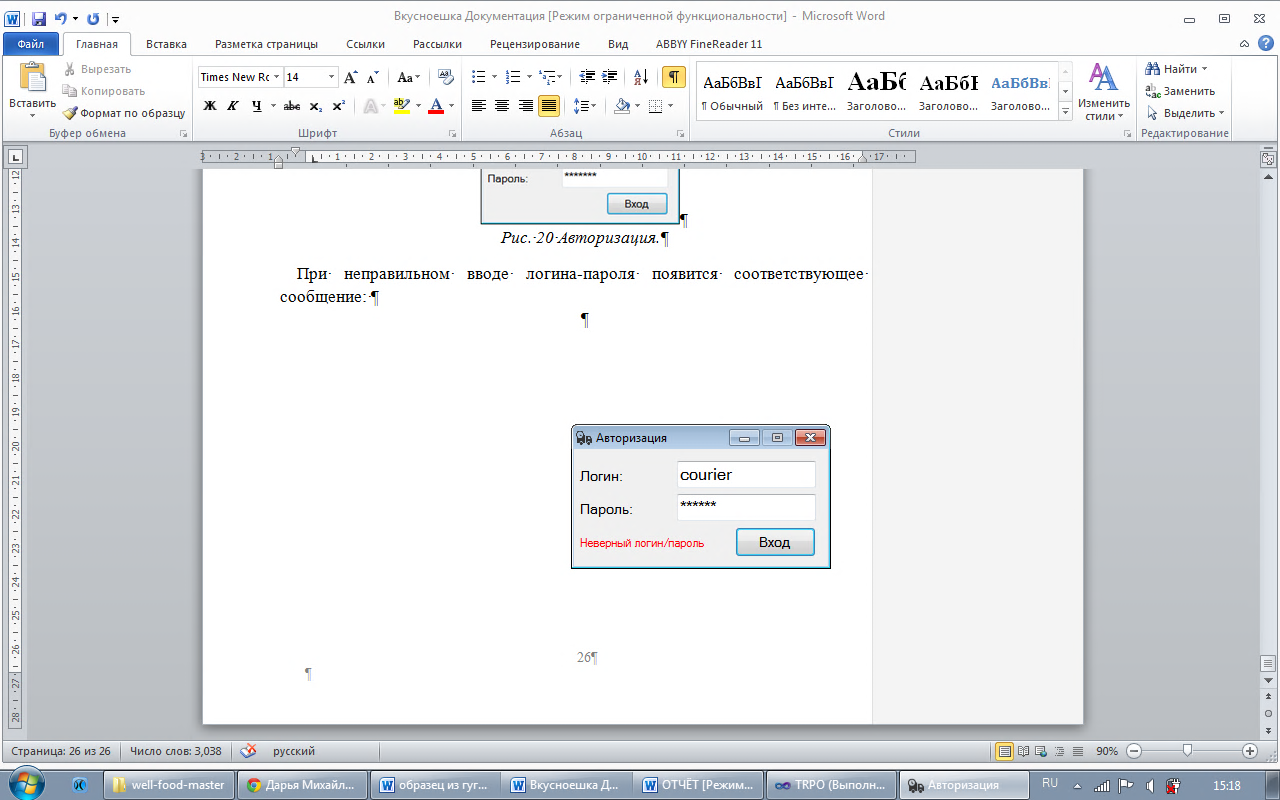
Для выполнения данной программы запускается приложение.

Для входа в систему необходимо пройти аутентификацию:



*Рис. 20 Авторизация.*

При неправильном вводе логина-пароля появится соответствующее сообщение:



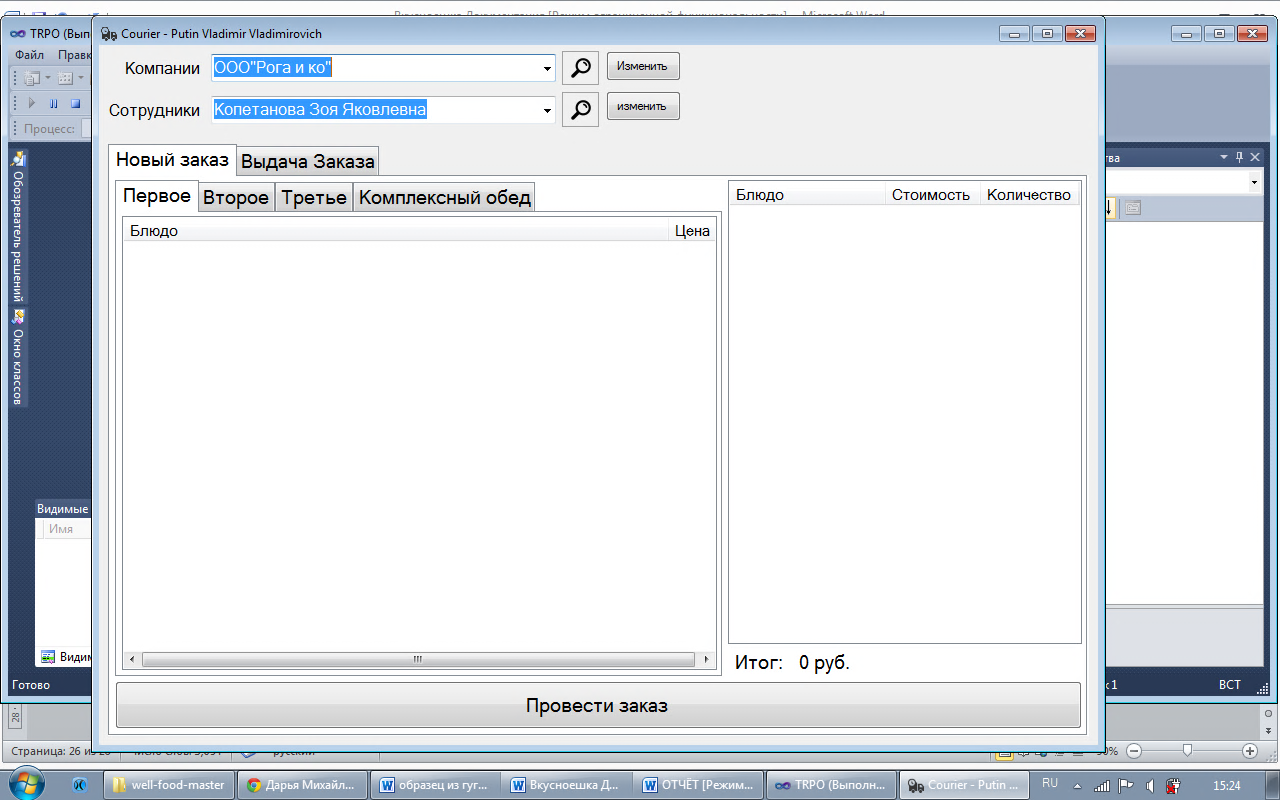
*Рис. 21 Ошибка авторизации.*

## 8.3. Выполнение программы

8.3.1 Пользователь «Курьер»:

* Запускаем приложение
* Проходим процедуру аутентификации, как курьер.

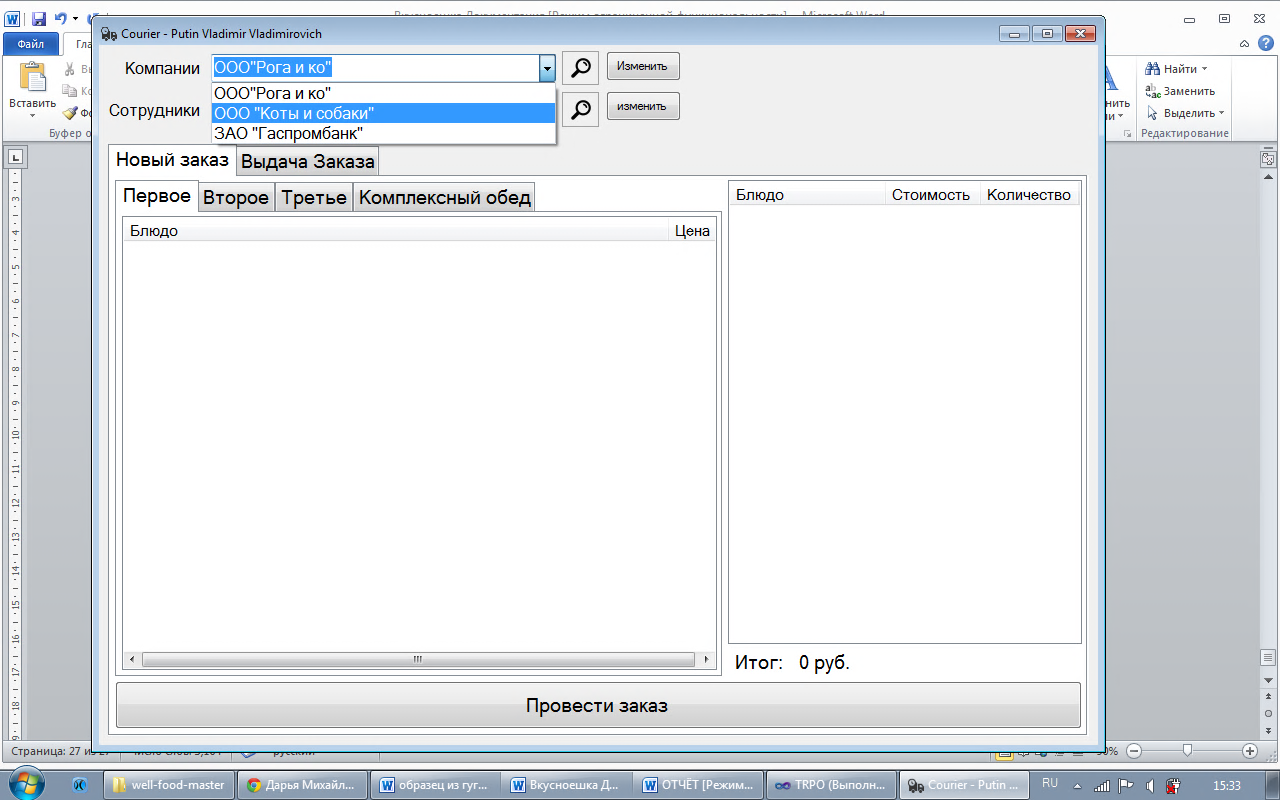
Если процедура аутентификации прошла успешно, открывается рабочее окно пользователя.



*Рис.22 Интерфейс курьера.*

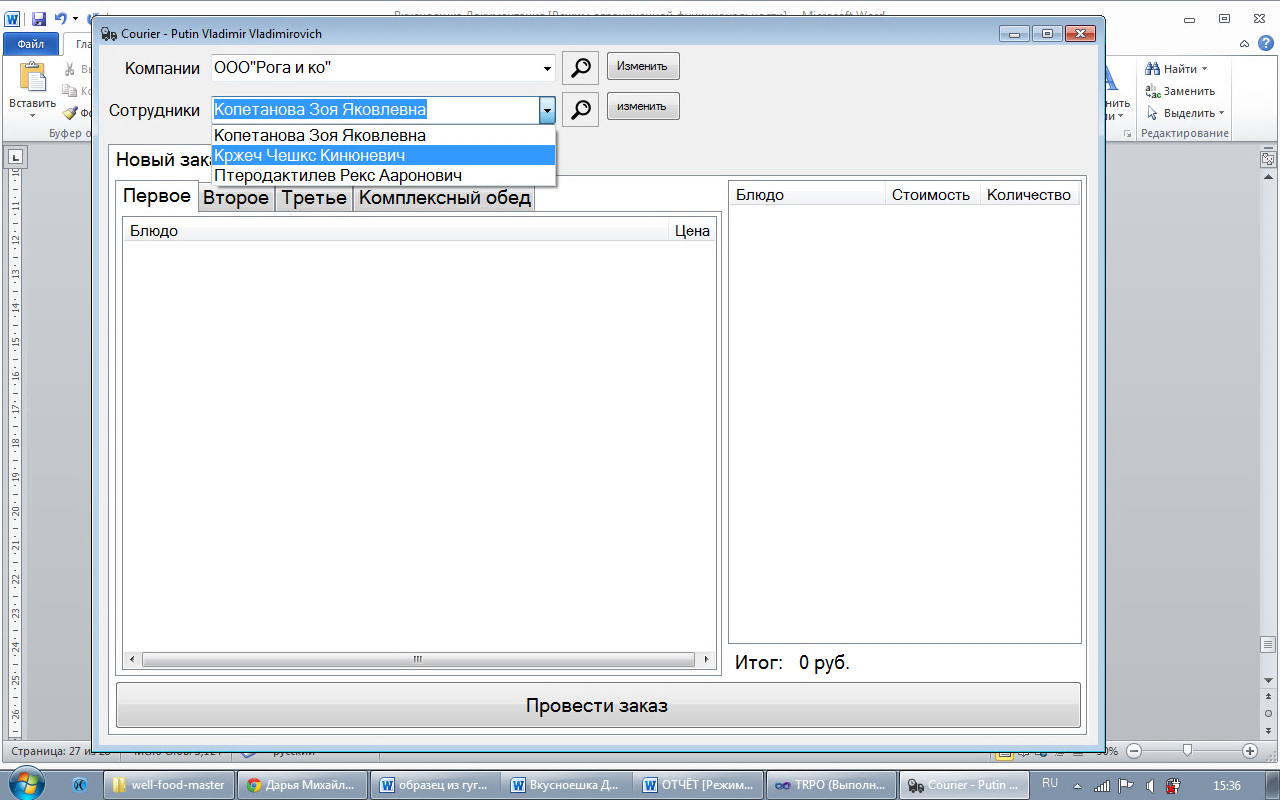
Данный пользователь должен формировать заказы для сотрудников фирм-клиентов.

Для этой процедуры во вкладке «Компании» пользователь может выбрать интересующую его фирму-клиента:



*Рис.23 Выбор компании в выпадающем списке.*

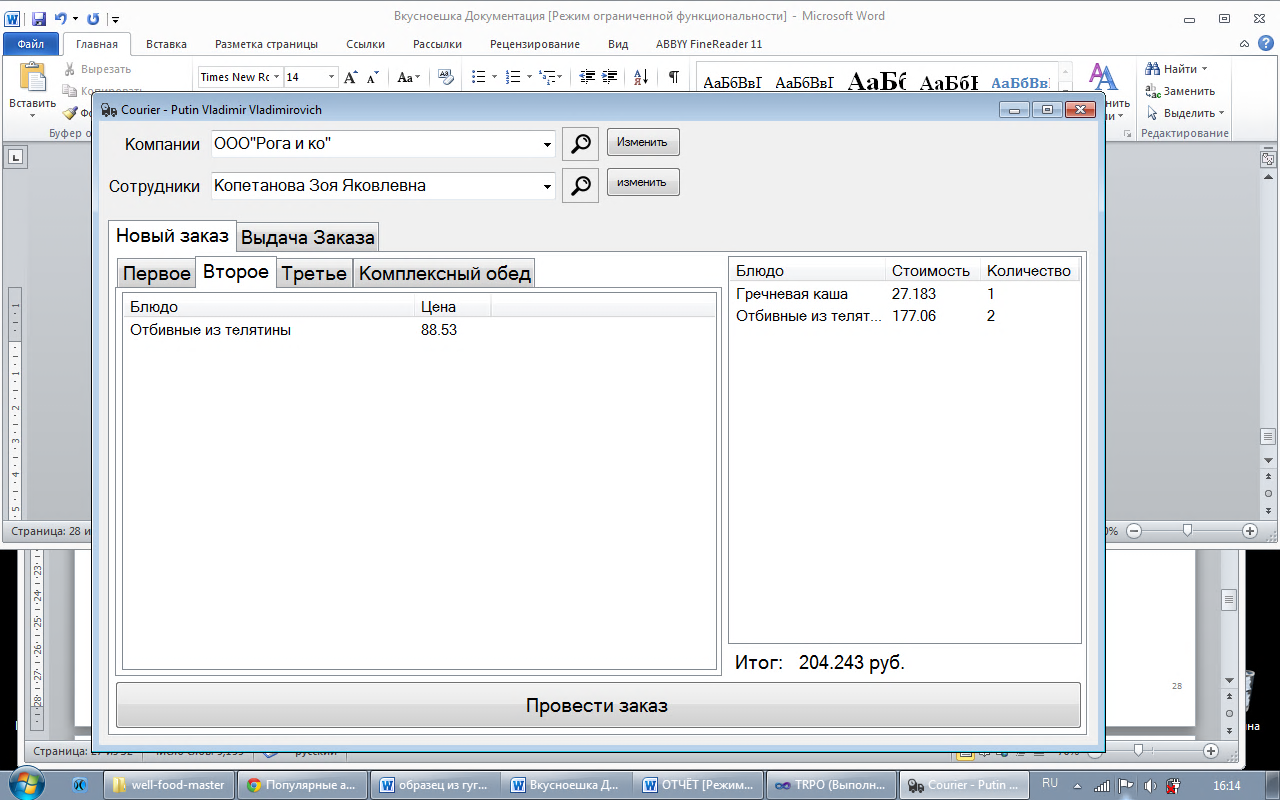
И у выбранной компании найти интересующего сотрудника для формирования нового или выдачи выполненного заказа:



*Рис.24 Выбор сотрудника компании.*

После выбора конкретного сотрудника можно сформировать на него заказ путем выбора вкладки «Новый заказ». Для добавления блюд в заказ их можно найти во вкладках «Первое», «Второе» и «Третье» и добавить двойным щелчком мыши. После чего блюдо появится в правом списке состава заказа. Для добавления второй или более порций необходимо еще раз дважды щелкнуть по выбранному блюду.

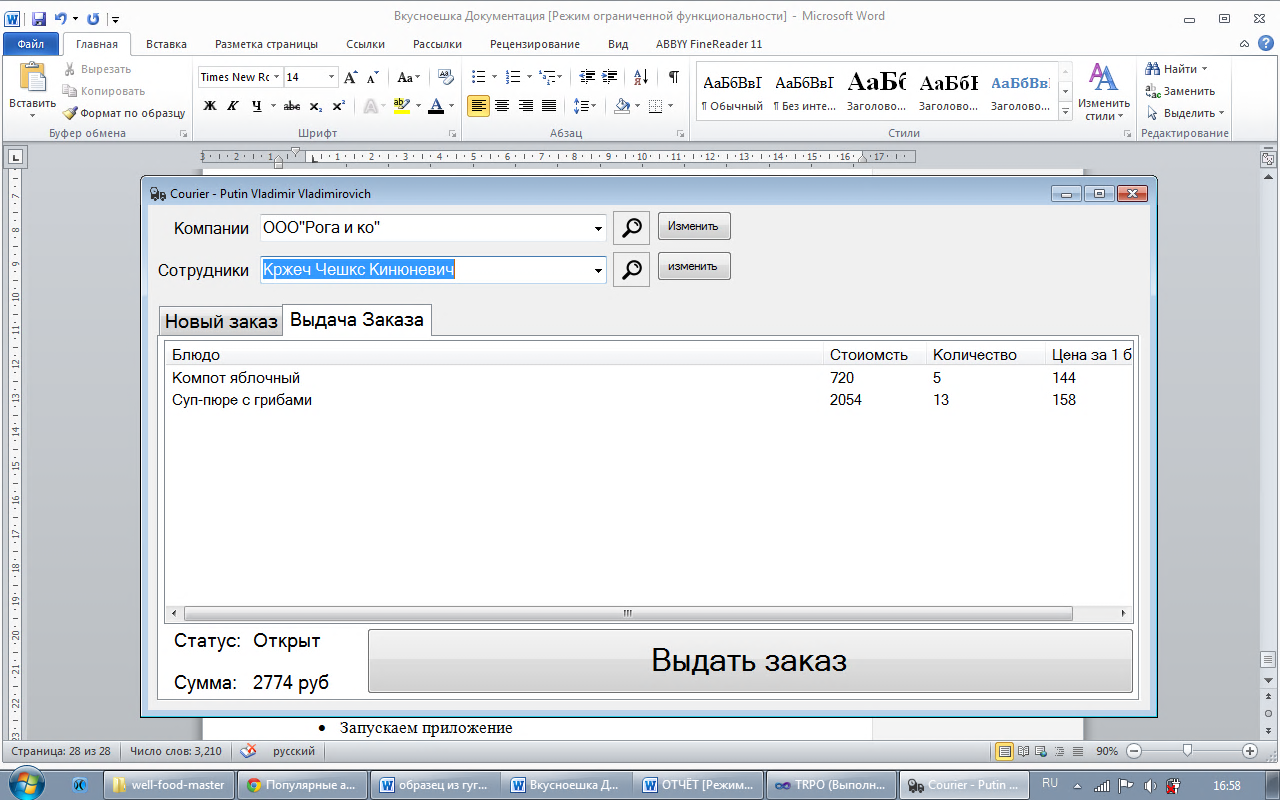
Если появилась необходимость удаления блюда из заказа, который формируется, или уменьшить количество порций можно просто дважды нажать на них в правом списке предварительного заказа.



*Рис.25 Формирование заказа*

После того как заказ полностью сформирован пользователь нажимает «Провести заказ». Заказ сохраняется в базе данных и попадает в очередь на обработку пользователю повар.

Выдача заказа осуществляется по переходу во вкладку «Выдача заказа»

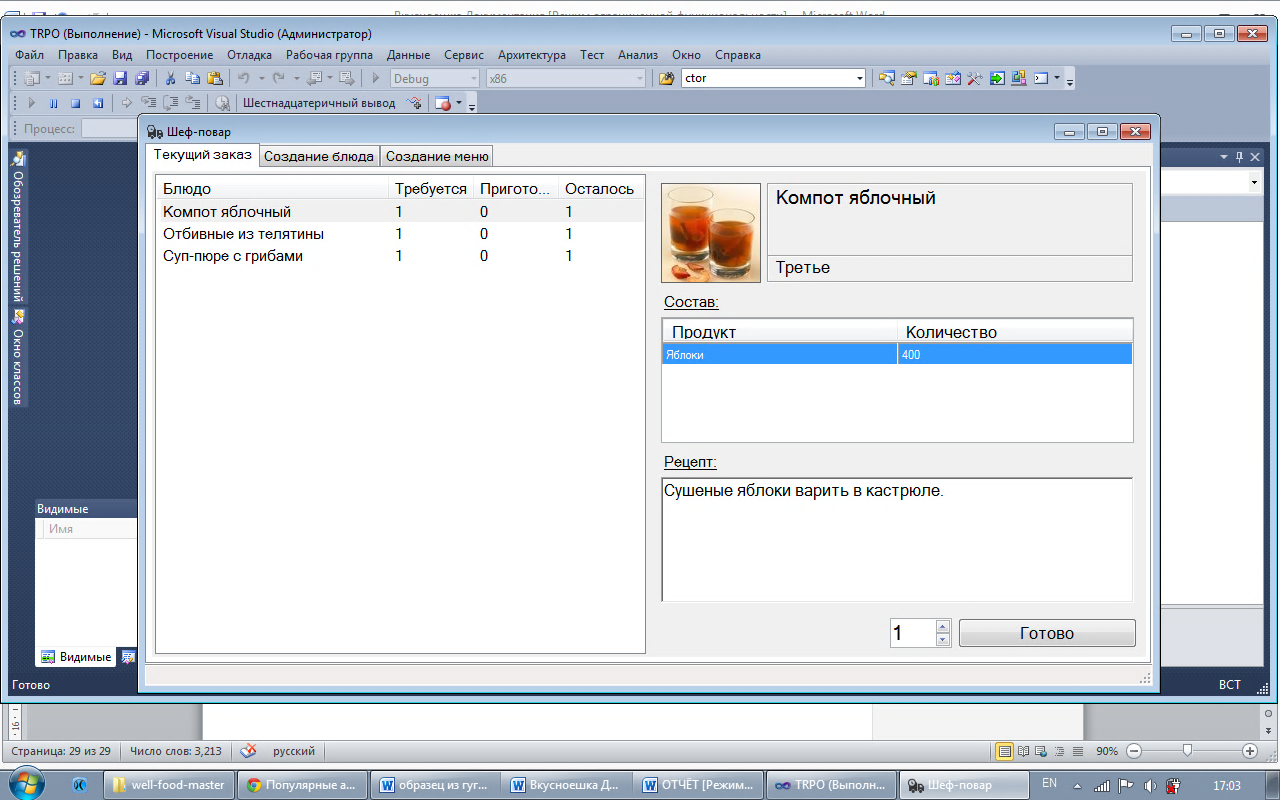


*Рис.26 Выдача заказа*

8.3.2 Пользователь «Повар»:

* Запускаем приложение
* Проходим процедуру аутентификации, как повар.

Если процедура аутентификации прошла успешно, открывается рабочее окно пользователя.

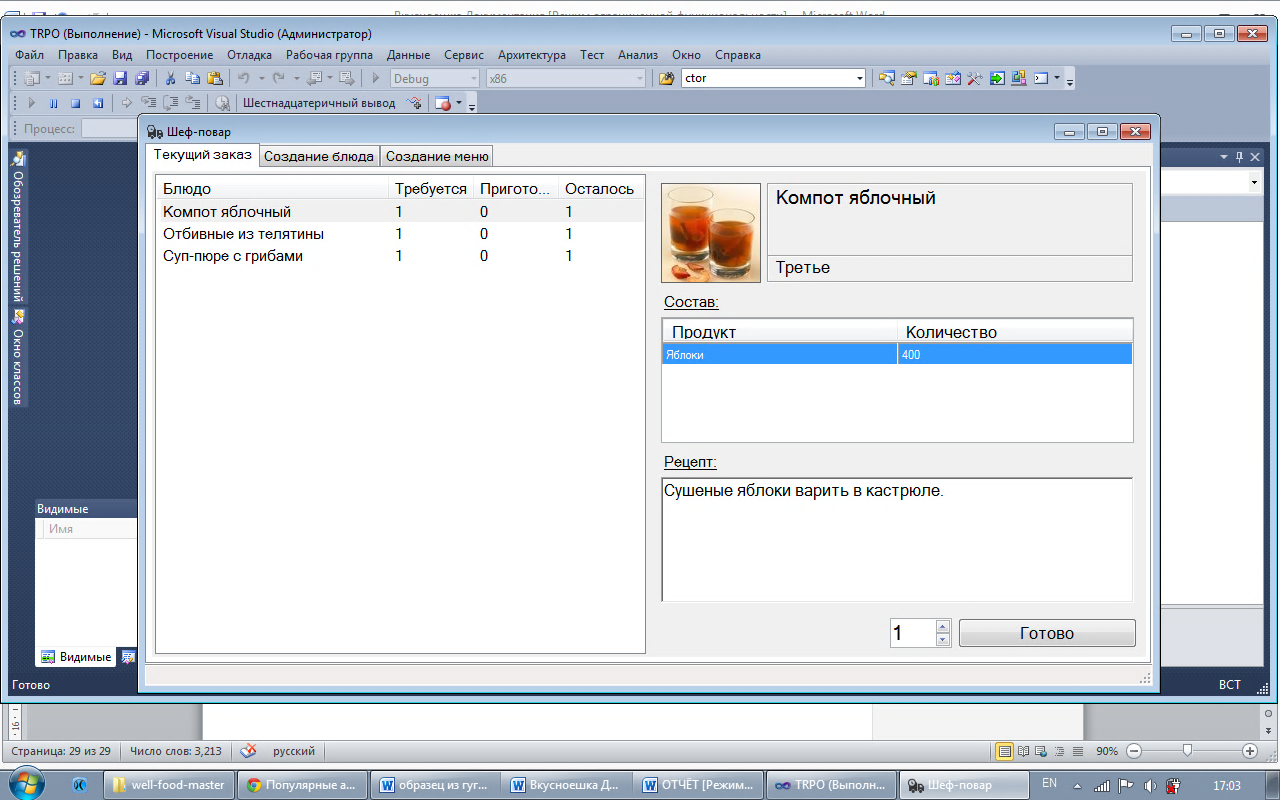


*Рис.27 Интерфейс повара*

Данный пользователь должен готовить блюда из списка заказов, а так же создавать новые блюда и формировать меню на день.

Для того чтобы увидеть список тех блюд, которые необходимо приготовить для заказов необходимо перейти во вкладку «Текущий заказ». В данной вкладке пользователь может ознакомиться не только со списком блюд, но так же и с количеством того сколько требуется, сколько уже приготовленного и сколько осталось ему приготовить. Кроме всего вышеперечисленного в правой части вкладки можно ознакомиться с тем, какие продукты необходимы, и в каком количестве, для приготовления единицы блюда, а так же с рецептом.

По мере приготовления пользователь может увеличивать количество приготовленных блюд:



*Рис.28 Увеличение количество приготовленных блюд*

Для создания нового или редактирования существующего блюда пользователю необходимо перейти во вкладку «Создание блюда».

Создание блюда. Для этого необходимо поставить в правом нижнем углу флажок «Создание нового блюда».



*Рис.29 Создание блюда*

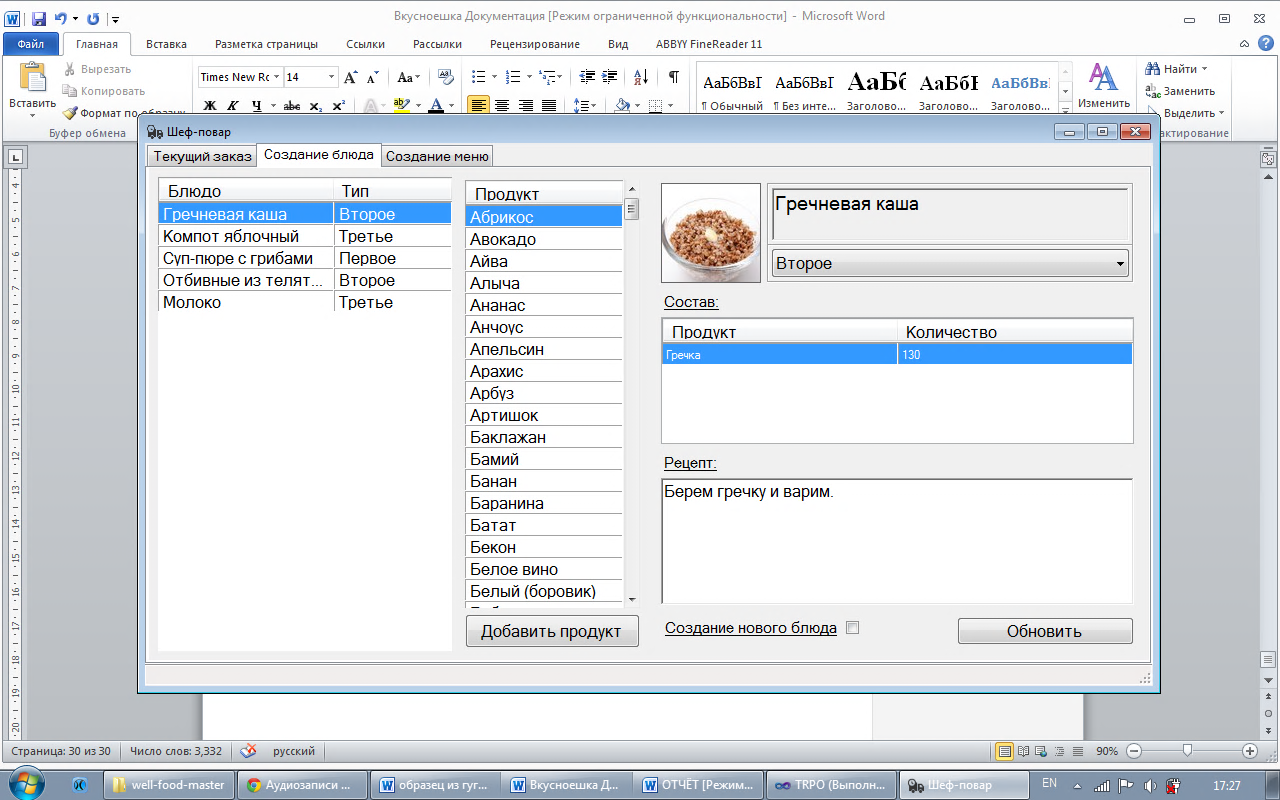
На этом этапе пользователю необходимо ввести название нового блюда, загрузить изображение, выбрать тип блюда. Для формирования состава блюда необходимо выбрать продукты из списка «Продукты» и подкорректировать их количество. Если вдруг блюда не оказалось в списке то его можно добавить нажав на кнопку «Добавить продукт»:



*Рис.30 Добавление продукта*

После чего продукт появиться в списке и его можно добавить в состав. Потом следует описать рецепт приготовления данного блюда, все еще раз проверить и, по нажатии кнопки «Создать», блюдо появиться в левом списке всех существующих в приложении блюд.

Для редактирования какого-то блюда надо выбрать его из списка слева и в правой части окна изменить какие-либо его данные, будь то название, изображение, состав или рецепт.

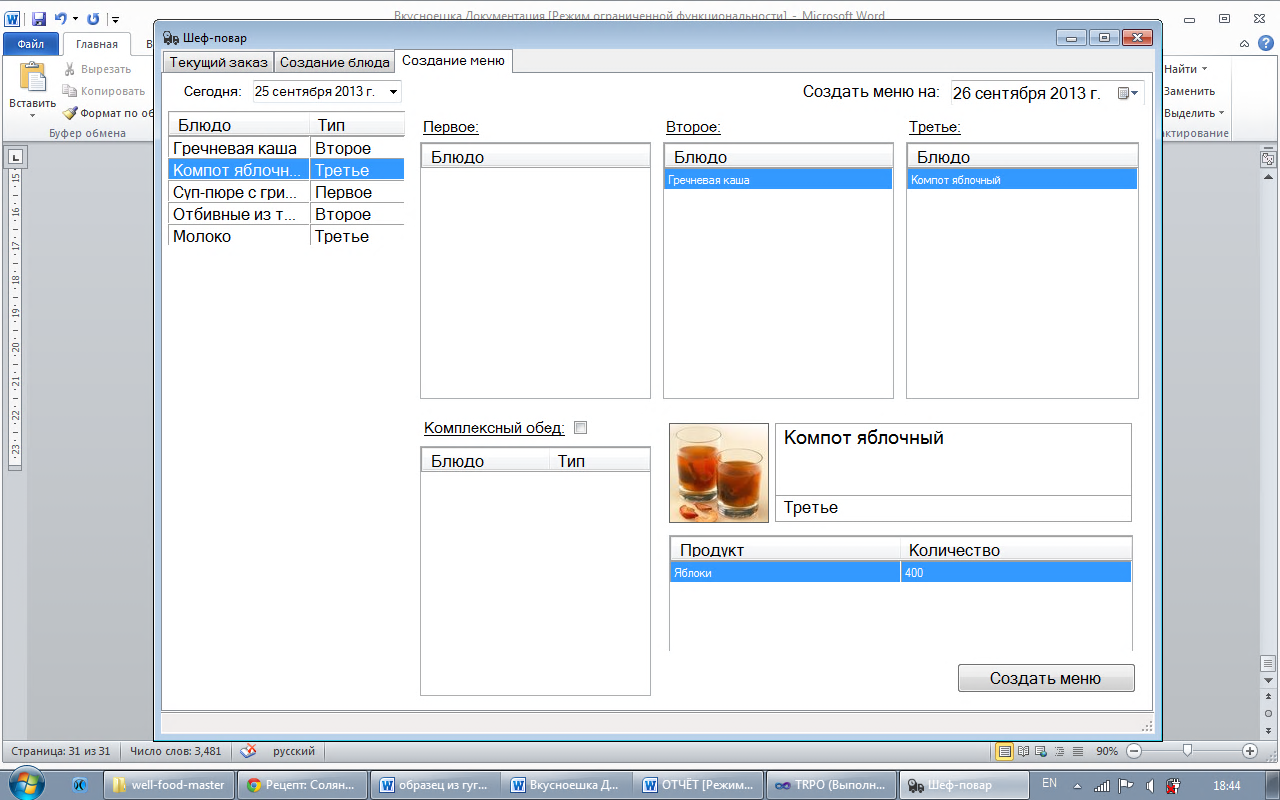


*Рис.31 Редактирование блюда*

Для того чтобы создать меню пользователю необходимо перейти на третью вкладку «Создание меню" и из левого списка всех блюд существующих в приложении двойным кликом выбирать те, которые будут в меню на следующий день.

Так же можно увидеть краткий состав каждого из блюд в левом нижнем углу приложения. И для того чтобы убрать блюдо из меню нужно дважды по нему кликнуть.

Также есть возможность создания комплексного обеда при выборе флажка «Комплексный обед».



*Рис. 32 Создание меню*