|  |  |
| --- | --- |
| Изображение выглядит как текст  Автоматически созданное описание | Министерство науки и высшего образования Российской Федерации  Федеральное государственное бюджетное образовательное учреждение высшего образования  «Новгородский государственный университет имени Ярослава Мудрого»  **ПОЛИТЕХНИЧЕСКИЙ КОЛЛЕДЖ** |

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ ДЛЯ КЛИЕНТА СТОЛОВОЙ

Курсовой проект

ПМ.01. Разработка ПРОГРАММНЫХ МОДУЛЕЙ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ДЛЯ КОМПЬЮТЕРНЫХ СИСТЕМ

МДК.01.02 пРИКЛАДНОЕ ПРОГРАММИРОВАНИЕ

Пояснительная записка курсового проекта по специальности

09.02.03 Программирование в компьютерных системах

|  |  |
| --- | --- |
|  | Студент группы 8902  \_\_\_\_\_\_\_\_/ В.И. Томиленко  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_2020 г |
|  | Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Руководитель  \_\_\_\_\_\_\_\_\_/ Н.В. Сазонова /  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_2020 г |

Содержание

[Аннотация 4](#_Toc58828737)

[Введение 5](#_Toc58828738)

[1 Техническое задание 6](#_Toc58828739)

[1.1 Постановка задачи 6](#_Toc58828740)

[1.2 Требования интерфейса 6](#_Toc58828741)

[2.Внешние спецификации 7](#_Toc58828742)

[2.1 Наименование задачи 7](#_Toc58828743)

[2.2 Словесное описание задачи 7](#_Toc58828744)

[2.3 Внешняя спецификация данных 7](#_Toc58828745)

[2.4 Функциональная спецификация 8](#_Toc58828746)

[2.5 Спецификация интерфейса 8](#_Toc58828747)

[2.6 Спецификация тестирования 10](#_Toc58828748)

[3 Внутренние спецификации 12](#_Toc58828749)

[3.1 Наименование программы 12](#_Toc58828750)

[3.2 Внутренняя спецификация данных 12](#_Toc58828751)

[3.3 Спецификация процедур и функций 13](#_Toc58828752)

[3.4 Информационная модель 16](#_Toc58828753)

[3.5 Алгоритмы программы 19](#_Toc58828754)

[4 Руководство оператора 23](#_Toc58828755)

[4.1 Назначение программы 23](#_Toc58828756)

[4.2 Условия выполнения программы 23](#_Toc58828757)

[4.3 Выполнение программы 24](#_Toc58828758)

[4.4 Сообщения оператору 25](#_Toc58828759)

[Заключение 27](#_Toc58828760)

[Список литературы 28](#_Toc58828761)

[Приложение А (обязательное) 29](#_Toc58828762)

[Приложение Б (обязательное) 33](#_Toc58828763)

[Приложение В (обязательное) 38](#_Toc58828764)

[Приложение Г (обязательное) 42](#_Toc58828765)

# Аннотация

Документ представляет собой пояснительную записку курсового проекта на тему «Разработка информационной системы для клиента столовой». В состав пояснительной записки входят три раздела: «Внешние спецификации», «Внутренние спецификации», «Руководство пользователя». Также в пояснительной записке будут присутствовать блок-схемы алгоритмов, диаграмма Ганта, use-case diagram, IDEFO диаграмма, DFD диаграмма и диаграмма классов.

# Введение

В соответствии с заданием на курсовое проектирование требуется разработать приложение для клиента столовой.

Информационные технологии занимают все более значимую роль в человеческом обществе. Они проникли во все сферы деятельности, используются для обслуживания общественных потребностей в автоматизации труда, хранения данных, связи и др.

Столовая является неотъемлемой частью каждого учебного заведения. Во время обеда в ней выстраиваются огромные очереди. Каждый хочет быстрее сесть за стол и обедать и из-за этого многие нервничают, злятся, когда очередь медленно движется. А ведь еще надо изучить меню!

Разрабатываемое приложение предназначено для студентов, преподавателей и других работников Политехнического колледжа. Оно должно упростить выбор блюд на обед. Таким образом, задача данного курсового проектирования является актуальной.

Цель курсового проекта — создать приложение, которое будет содержать меню столовой Политехнического колледжа.

Задачи курсового проекта:

* знакомство с предметной областью исследуемой задачи;
* разработка модели базы данных (концептуальной или реляционной);
* создание структуры базы данных средствами СУБД;
* проектирование и создание приложения для работы с базой данных;
* разработка руководства пользователя.

# 1 Техническое задание

# Постановка задачи

Необходимо разработать приложение, которое будет хранить меню столовой на каждый день, а именно иметь данные о каждом блюде: название, состав, цена, вес. Также должна быть разработана реализация таких возможностей, как: выбор блюд и расчет итоговой стоимости выбранного заказа, использование карт постоянных покупателей и скидок. В том числе в конце месяца должен быть сформирован отчет об остатках продуктов на складе.

# Требования интерфейса

Должны присутствовать экраны:

* экран авторизации пользователя;
* главный экран программы;
* экран данных о блюдах;
* экран формирования заказа;
* экран справки.

# 2 Внешние спецификации

# 2.1 Наименование задачи

Главным назначением будущей программы будет являться просмотр меню столовой Политехнического колледжа, выбор блюд. В документации разработка программы называется комплексом работ по проектированию, созданию, тестированию, реализации проекта.

# 2.2 Словесное описание задачи

Для просмотра меню столовой и выбора блюд будет создано приложение, которое будет иметь следующие формы для решения поставленных задач:

* форма просмотра меню за всю неделю;
* формы выбора блюд и расчёт их итоговой суммы;
* форма справки.

Данное приложение направленно на возможное сокращение времени на просмотр меню и выбор блюд непосредственно во время выдачи блюд.

Для более эффективного решение поставленной задачи необходимо изучить предметную область, спроектировать и разработать программу для актуального рейтинга пользователей, используя язык программирования C#.

# 2.3 Внешняя спецификация данных

Данные, которые будут использоваться для создания базы данных можно увидеть в таблице 1.

Таблица 1 - Данные, как объекты внешнего мира

|  |  |  |  |
| --- | --- | --- | --- |
| Объект внешнего мира | Свойства объекта | Характеристики свойств | Связь между объектами и внутри объекта |
| Блюдо | Код\_блюда | Primary key | - |
| Название | Короткий текст | - |
| ID склада | Foreign key | Склад. Foreign key |
| Состав | Короткий текст | - |
| Вес | Число | - |
| Цена | Число | - |
| Пользователь | Код\_пользователя | Primary key | - |
| Логин | Короткий текст | - |
| Пароль | Короткий текст | - |
| Фамилия | Короткий текст | - |
| Имя | Короткий текст | - |
| Отчество | Короткий текст | - |
| Номер карты | Число | - |
| Склад | ID склада | Primary key | - |
| Количество продуктов в начале месяца | Число | - |
| Количество продуктов, раскупленных за неделю | Число | - |

Входные данные: Логин и пароль пользователя.

Выходные данные: Информация о пользователе (ФИО), просмотр меню и расчёт итоговой стоимости выбранных блюд.

# 2.4 Функциональная спецификация

Для авторизации пользователя использовалась функция ввода и проверки данных.

Для просмотра меню использовалась функция соединения к базе данных.

Для выбора и подсчета блюд использовались функции ввода и проверки данных, выбор элементов.

# 2.5 Спецификация интерфейса

Дизайн программы выполнен желтых и зеленых тонах. Так как данные цвета вызывают у человека повышенный интерес к еде.

Спецификация интерфейса представлена в виде словесного описания и рисунков.

Основное взаимодействие: Пользователь – графический интерфейс.

Оборудование: ПК, клавиатура, компьютерная мышь, монитор.

Способы взаимодействия: Нажатие ЛКМ.

Для взаимодействия с программой будут выполняться один из перечисленных способов:

* ввод данных пользователя (логин, пароль, номер карты);
* нажатие на кнопки для перехода на другие формы.

Посмотреть примеры форм можно в приложении А на рисунках: А.1, А.2, А.3, А.4, А.5, А.6 и А.7.

# 2.6 Спецификация тестирования

Данные внешнего тестирования можно увидеть в таблице 2.

Таблица 2 – Данные внешнего тестирования

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | Входные данные | | | Ожидаемые выходные данные | | | Действия программы |
| Форма авторизации | Формы выбора блюд | Формы выбора блюд | Форма авторизации | Формы выбора блюд | Формы выбора блюд |
| Обработка сбойных ситуаций | Неправильно введены данные в поле «пользователь» и/или в поле «пароль» | Выбрано второе блюдо без гарнира | Не выбрано ни одно блюдо | Вывод сообщения об ошибке | Вывод сообщения об ошибке | Вывод сообщения об ошибке | Обработка исключение |
| Примеры корректной работы | Правильно введены данные в поле «пользователь» и/или в поле «пароль» | Выбрано второе блюдо с гарнира | Выбрано некое количество блюд | Переход на следующую форму | Расчет итоговой цены | Расчет итоговой цены | Выполнение sql-запроса |

Данные тестирования внешней спецификации можно увидеть в таблице 3.

Таблица 3 – Данные тестирования внешней спецификации

|  |  |  |
| --- | --- | --- |
| Этап работы программы | Действия пользователя | Ожидаемое поведение программы |
| Форма авторизации | Ввод данных в  поля «пользователь» и «пароль» | Переход на следующую форму |
| Панель клиента/администратора | Выбор/нажатие определенной кнопки | Переход на выбранную форму |
| Форма просмотра меню | 1. Выбор/нажатие вкладки «день недели» и выбор/нажатие нужного дня недели; 2. Выбор/нажатие вкладки «выбор блюд» и выбор/нажатие нужного дня недели; 3. Выбор/нажатие вкладки «действия» и выбор/нажатие «справки» или «выход». | 1. Вывод на форму меню выбранного дня; 2. Переход по выбранному дню недели на нужную форму расчета итоговой суммы; 3. Если выбрана «справка», то открытие формы справки. Если выбран «выход», то закрытие программы. |
| Форма дней недели для выбора блюд | Выбор/нажатие нужного дня неделе | Переход на выбранную форму |
| Форма выбора блюд | 1. Выбор блюд, ввод номера карты (не обязательно) и нажатие кнопки «рассчитать»; 2. Нажатие кнопки «назад»; 3. Нажатие кнопки «выход». | 1. Расчет итоговой суммы со (без) скидкой(-и); 2. Переход на форму «Форма дней недели для выбора блюд»; 3. Закрытие программы. |
| Форма отчета | 1. Нажатие кнопки «назад»; 2. Нажатие кнопки «выход». | 1. Возвращение на панель администратора; 2. Выход из программы. |

# 3 Внутренние спецификации

# Наименование программы

Данная программа носит название «IS\_Canteen», что расшифровывается как информационная система столовой.

# Внутренняя спецификация данных

Внутреннюю спецификацию данных программы можно увидеть в таблице 4.

Таблица 4 - Внутренняя спецификация данных

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Объект задачи | Имя в программе | Характеристики | | | Характер использования |
| Тип | Диапазон | Простая/структура |
| ФИО пользователя | lastName, firstName, secondName | Строковый | - | Простая | Выход |
| Подключение к базе данных | query | Строковый | - | Простая | Константа |
| Приветствие | txt | Строковый | - | Простая | Выход |
| Цена блюд | sal\_1, sal\_2, sal\_3, soup, secD\_1, secD\_2, secD\_3, g\_1, g\_2, n\_1, n\_2, n\_3, n\_4, n\_5, n\_6, n\_7, v\_1, v\_2, v\_3, v\_4, v\_5, v\_6 | Веществ. | [-∞;+∞] | Простая | Константа |
| Скидка | sk\_1, sk\_2 | Веществ. | [-∞;+∞] | Простая | Константа |
| Итоговая цена без скидки | sum | Веществ. | [-∞;+∞] | Простая | Выход |
| Итоговая цена со скидкой | card\_1 | Веществ. | [-∞;+∞] | Простая | Выход |
| Подсчет скидки | с | Веществ. | [-∞;+∞] | Простая | Константа |
| Получение данных из базы данных | adapter | SqlDataAdapter | - | Простая | Константа |
| Получение данных из базы данных | cmd | SqlCommand | - | Простая | Константа |
| Получение данных из базы данных | dataTable | DataTable | - | Простая | Константа |
| Получение данных из базы данных | dataReader | SqlDataReader | - | Простая | Константа |

# Спецификация процедур и функций

Панель авторизации:

* btnLog\_Click – функция, которая при нажатии проверяет введённые данные пользователем;
* btnExit\_Click – функция, которая закрывает программу.

Панель администратора:

* Exit\_Click – функция, которая закрывает программу;
* SeeMenu\_Click – функция, которая закрывает панель администратора и открывает панель просмотра меню;
* week\_Click – функция, которая закрывает панель администратора и открывает панель выбора блюд;
* Report\_Click – функция, которая закрывает панель администратора и открывает панель просмотра отчета.

Панель клиента:

* Exit\_Click – функция, которая закрывает программу;
* SeeMenu1\_Click – функция, которая закрывает панель администратора и открывает панель просмотра меню;
* week1\_Click – функция, которая закрывает панель администратора и открывает панель выбора блюд;
* Ref\_Click – функция, которая открывает панель просмотра справки.

Панель справки:

* Nazad\_Click – функция, которая закрывает панель справки.

Панель отчета:

* Exit\_Click – функция, которая закрывает программу;
* Nazad\_Click – функция, которая закрывает панель отчета и открывает панель администратора.

Панель просмотра меню:

* выходToolStripMenuItem\_Click – функция, которая закрывает программу;
* справкаToolStripMenuItem\_Click – функция, которая открывает панель просмотра справки;
* понедельникToolStripMenuItem\_Click – функция, которая выводит на текущую панель меню на понедельник;
* вторникToolStripMenuItem\_Click – функция, которая выводит на текущую панель меню на вторник;
* средаToolStripMenuItem\_Click – функция, которая выводит на текущую панель меню на среду;
* четвергToolStripMenuItem\_Click – функция, которая выводит на текущую панель меню на четверг;
* пятницаToolStripMenuItem\_Click – функция, которая выводит на текущую панель меню на пятницу;
* понедельникToolStripMenuItem1\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на понедельник;
* вторникToolStripMenuItem1\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на вторник;
* средаToolStripMenuItem1\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на среду;
* четвергToolStripMenuItem1\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на четверг;
* пятницаToolStripMenuItem1\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на пятницу.

Панель выбора блюд:

* Exit\_Click – функция, которая закрывает программу;
* btnMon\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на понедельник;
* btnTue\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на вторник;
* btnWen\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на среду;
* btnThu\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на четверг;
* btnFri\_Click – функция, которая закрывает текущую панель и открывает панель выбора блюд на пятницу.

Панель выбора блюд на понедельник:

* Exit\_Click – функция, которая закрывает программу;
* rass\_Click – функция, которая рассчитывает итоговою стоимость выбранных блюд;
* nazad\_Click – функция, которая закрывает текущую панель и переходит на панель выбора блюд.

Панель выбора блюд на вторник:

* Exit\_Click – функция, которая закрывает программу;
* rass\_Click – функция, которая рассчитывает итоговою стоимость выбранных блюд;
* nazad\_Click – функция, которая закрывает текущую панель и переходит на панель выбора блюд.

Панель выбора блюд на среду:

* Exit\_Click – функция, которая закрывает программу;
* rass\_Click – функция, которая рассчитывает итоговою стоимость выбранных блюд;
* nazad\_Click – функция, которая закрывает текущую панель и переходит на панель выбора блюд.

Панель выбора блюд на четверг:

* Exit\_Click – функция, которая закрывает программу;
* rass\_Click – функция, которая рассчитывает итоговою стоимость выбранных блюд;
* nazad\_Click – функция, которая закрывает текущую панель и переходит на панель выбора блюд.

Панель выбора блюд на пятницу:

* Exit\_Click – функция, которая закрывает программу;
* rass\_Click – функция, которая рассчитывает итоговою стоимость выбранных блюд;
* nazad\_Click – функция, которая закрывает текущую панель и переходит на панель выбора блюд.

# Информационная модель

Работа разрабатываемого приложения будет производиться с помощью связи классов, а в дальнейшем с помощью базы данных.

Работа началась с того, что была созданы модели базы данных MS SQL с помощью программы построения диаграмм Microsoft Visio.В соответствии с анализом задачи было создано три таблицы:

* dish (таблица с блюдами);
* sklad (таблица с данными склада);
* users (таблица с данными входа).

Одним из основных объектов является таблица dish, которая создана для хранения данных о блюдах. В информационной модели этот объект представляет следующий набор свойств:

* уникальный идентификатор;
* вид;
* название;
* состав;
* вес;
* цена;
* внешний ключ склада.

Таблица users создана для хранения информации для входа пользователя или администратора:

* уникальный идентификатор;
* логин;
* пароль;
* фамилия;
* имя;
* отчество;
* внешний ключ на номер карты.

Таблица sklad создана для хранения информации о количестве определенного блюда на складе:

* уникальный идентификатор;
* количество блюд в начале месяца;
* количество проданных блюд за неделю.

Также было создано пять представлений, содержащих в себе меню на один определённый день:

* Monday;
* Tuesday;
* Wednesday;
* Thursday;
* Friday.

Для создания отчета была создана хранимая процедура.

IDEF1X-диаграмма изображена на рисунках Б.1 и Б.2.

Увидеть примерное расписание последующих шагов можно на рисунках Б.3, Б.4, Б.5, Б.6.

Диаграмма потоков изображена на рисунке Б.7.

USE-CASE-диаграмма изображена на рисунке Б.8.

IDEF0-диаграмма изображена на рисунке Б.9.

Диаграмма классов изображена на рисунке Б.10.

# Алгоритмы программы

Алгоритм программы представлен в виде словесного описания, блок-схем и программного кода.

Основной алгоритм действия программы заключается в следующем.

Пользователь запускает программу, появляется первая форма – авторизация.

На панели авторизации находится две кнопки:

* «Войти»;
* «Выход».

При нажатии кнопки «Войти» происходит подключении к базе данных и проверка введенных данных.

Если введение данные верны, то открывается окно приветствия с ФИО пользователя, взятых из таблицы users и после осуществляется переход на клиентскую/администраторскую панель.

Если одно из введенных данных неверно, то открывается сообщение об ошибке и очищается поле для ввода пароля.

При нажатии кнопки «Выход» происходит закрытие программы.

Дальше в зависимости от введённых данных, программа перекинет пользователя на клиентскую или административную форму. Посмотреть блок-схему авторизации можно посмотреть в приложении В на рисунках В.1 и В.2. Посмотреть программный код можно в приложении Г.

Панель администратора состоит из четырех кнопок:

* «Просмотр меню»;
* «Выбор блюда»;
* «Отчет»;
* «Выход».

Кнопка «Просмотр меню» перенаправляет пользователя на форму, где пользователь можно посмотреть полное меню на всю неделю, а также перейти на формы «Выбора блюд».

Кнопка «Выбор блюд» перенаправляет на форму «Дни недели», где можно найти нужный день для выбора блюда и расчета итоговой суммы.

Кнопка «Отчет» открывает форму, где предоставлены данные о продуктах на конец месяца.

Кнопка «Выход» закрывает программу.

На панели клиента находится четыре кнопки:

* «Просмотр меню»;
* «Выбор блюда»;
* «Справка»;
* «Выход»;

Кнопки «Просмотр меню» и «Выбор блюд» выполняют туже функцию, что и на панели администратора.

Кнопка «Справка» открывает форму, где написаны основные моменты взаимодействия с программой.

Кнопка «Выход» закрывает программу.

Панель справки имеет одну кнопку «Назад», которая закрывает форму справки.

Панель отчета предоставляет информацию о блюдах администратору. Сколько блюд осталось на конец месяца.

На панели отчета находится две кнопки:

* «Выход»;
* «Назад».

Кнопка «Выход» закрывает программу.

Кнопка «Назад» закрывает панель отчета и открывает панель администратора.

На панели просмотра меню пользователь может посмотреть полное меню на каждый рабочий день недели. Перейти на выбор блюд по конкретному дню, просмотреть справку и выйти из программы.

Панель просмотра меню состоит элементов ToolStripMenu и DataGridView.

ToolStripMenu имеет три главные кнопки:

* «День недели»
* «Выбор блюд»;
* «Действия».

Кнопки «День недели» и «Выбор блюд» открывают вкладки, на которых можно выбрать нужный день недели.

Дни недели, выбранные на «День недели», подключаются к базе данных и выводят полное меню в DataGridView.

День недели, относящийся к «Выбор блюд», переносит пользователя на форму выбранного дня, где можно выбрать нужные блюда и рассчитать итоговую сумму.

Кнопка «Действие» дает возможность просмотреть справку или выйти из программы.

Панель выбора блюд состоит из кнопок, которые дают возможность выбрать нужный день недели. Всего на панели имеется шесть кнопок:

* «Понедельник»;
* «Вторник»;
* «Среда»;
* «Четверг»;
* «Пятница»;
* «Выход».

В программе присутствует пять практически идентичных форм выбора блюд. Поэтому их общие элементы будут рассматриваться совместно. На каждой панели есть по три кнопки:

* «Назад»;
* «Рассчитать»;
* «Выход».

Кнопка «Выход» закрывает программу.

Кнопка «Назад» возвращает пользователя на форму выбора нужного дня недели.

Кнопка «Рассчитать» проверяет какие блюда выбраны, суммирует их общую стоимость и выводит их на экран.

Если кнопка была нажата, но никакое блюдо не было выбрано, то появляется сообщение об ошибке.

Также будет выводиться ошибка, если пользователь выбрал второе блюдо, но не выбрал к нему гарнир.

Также пользователь может ввести последние четыре цифры своей скидочной карты, чтобы активизировать скидку. В этом случае при нажатии кнопки отправляется запрос в базу данных для проверки наличия введенного номера карты. Если номер не существует, то выводится сообщение об ошибке. Если номер есть, то ищется выбранное блюдо с наибольшей скидкой. После с найденной скидкой рассчитывается итоговая сумма. Посмотреть блок-схему расчета цена можно посмотреть в приложении В на рисунках В.3 и В.4. Посмотреть программный код можно в приложении Г.

# 4 Руководство оператора

# 4.1 Назначение программы

Программа предназначена для общего пользования, входит в возрастной рейтинг 16+. Используется для получения нужной информации о блюдах, их составе, весе, цене. Программа дает быстрый доступ, как и обычному пользователю, так и администратору.

Программа со стороны администратора предназначена для удобной работы с базой данных, поэтому было решено разработать программный продукт, который в форме приложения был полностью защищен от некорректных действий пользователя. Был разработан простой и понятный интерфейс.

Программа со стороны клиента предназначена для передачи нужной информации, при этом защищая пользователя от некорректных действий.

# 4.2 Условия выполнения программы

Программа «IS\_Canteen» работает на компьютере стандартной конфигурации, на котором установлена операционная система Windows.

Минимальные требования:

* монитор с разрешением 800Х600;
* оперативная память 2 ГБ;
* процессор Pentium G3258;
* видеокарта GeForce 7800 GTX;
* материнская плата ECS Z97-PK;
* постоянное запоминающее устройство 60 Мб;
* операционная система Windows 7 Home.

Рекомендуемые требования:

* монитор с разрешением 1280Х760;
* оперативная память 8 ГБ;
* процессор intel core i3;
* видеокарта GeForce GTX 265;
* материнская плата Gigabyte 365 Ultra Durable;
* постоянное запоминающее устройство 80 Мб;
* операционная система Windows 8.

Максимальные требования:

* монитор с разрешением 1920Х1080;
* оперативная память 16 ГБ;
* процессор AMD Ryzen 5 3600 Processor;
* видеокарта GeForce GTX 1050;
* материнская плата Gigabyte 365 Ultra Durable;
* постоянное запоминающее устройство 100 Мб;
* операционная система Windows 10.

# 4.3 Выполнение программы

При запуске приложения открывается экран авторизации. После ввода логина и пароля пользователи и нажатия на кнопку “Войти” осуществляется приветствие пользователя и переадресация на главный экран программы.

Для выхода из программы нужно на экране нажать кнопку «Выход» или нажать на крестик в правом верхнем углу окна приложения.

Получить информацию о программе можно на форме «Справка», которая находится на двух формах:

* клиентская панель;
* форма «Просмотра меню».

Для получения справки на клиентской панели, надо нажать кнопку «Справка» на главном экране. На экране появляется окно с необходимой информацией.

Для получения справки на форме «Просмотра блюд», надо выбрать вкладку «Действия» и нажать на «Справка».

Для просмотра полного меню следует нажимать кнопку «Просмотр меню» на главной панели.

Для открытия форм с выбором блюд можно использовать два способа:

* Нажать на главной панели кнопку «Выбор блюд», перейти на форму «Дни недели» и выбрать кнопку, где написан нужный день. После произойдет переход на нужную форму;
* Нажать на главной панели «Просмотр блюд», после перехода зайти на вкладку «Выбор блюд» и выбрать интересующий день.

# 4.4 Сообщения оператору

Пользователь, во время использования программы, будет получать два вида сообщений:

* информационную;
* об ошибке.

Информационное сообщение пользователь получает при правильном вводе свои логина и пароля. В сообщение содержится приветствие с обращением к пользователю: «Здравствуйте, …!».

Сообщение об ошибке пользователь будет получать каждый раз при совершение каких-либо некорректных действий. Такие сообщения можно получить на формах:

* авторизации;
* понедельник;
* вторник;
* среда;
* четверг;
* пятница.

На форме авторизации сообщение будет появляться, если пользователь будет вводить не правильные логин или пароль.

Сообщение будет содержать данную информацию: «Введенные данные не верны. Или не введены не полностью».

На формах выбора блюд и расчета их общей стоимости по всем рабочим дням недели будут выводиться одинаковые сообщения об ошибке.

Сообщения будут о:

* том, что не выбрано ни одно блюдо и расчет не возможен;
* выбрано второе блюдо, но не выбран гарнир;
* введенный номер карты не верен.

# Заключение

В соответствии с заданием на курсовом проектирование было разработано приложение для клиента столовой.

При проектировании были улучшены знания для работы со специализированными программами:

* Visio 2016;
* Microsoft SQL Server Management Studio 18;
* Microsoft Visual Studio 2017.

Также, улучшены навыки программирования, такие как:

* создание баз данных в Microsoft SQL Server Management Studio 18;
* создание, заполнение и изменение таблиц;
* подключение базы данных к программе и дальнейшая работа с ней через программный код;
* работа с элементами форм.

Разработан программный продукт, соответствующий требованиям индивидуального задания.

В данной работе были решены все поставленные во введении задачи, начиная со знакомства с предметной областью и заканчивая разработкой приложения.

На этапе проектирования программы использовались такие программы как: Microsoft Visual Studio 2017 и Microsoft SQL Server Management Studio 18. Microsoft Visual Studio 2017 использовался для создания самой программы, начиная внешним видом заканчивая действиями, которые она выполняет. В Microsoft SQL Server Management Studio 18 были созданы все таблицы и хранимая процедура. Также для создания программы использовался объектно-ориентированный язык программирования C#.

Поэтому, можно сделать вывод, что задание на курсовое проектирование выполнено в полном объёме.

# Список литературы

1. СТО 1.505 – 2010.
2. СТО 1.701-2010.
3. Казанский, А. А. Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва: Издательство Юрайт, 2020. — 192 с.
4. Джепикс Филипп, Троелсен Эндрю, Язык программирования C# 7 и платформы .NET и .NET Core/ Джепикс Филипп, Троелсен Эндрю – 8-е изд. — Москва: Издательство Вильямс, 2018 – 1328 с.
5. Форта Бен, Язык T-SQL для Microsoft SQL Server за 10 минут/ Форта Бен — Москва: Издательство Вильямс, 2017 – 384 с.
6. Мюллер Джон Поль, Семпф Билл, Сфер Чак C# для чайников/ Мюллер Джон Поль, Семпф Билл, Сфер Чак — Москва: Издательство Диалектика, 2019 – 609 с.
7. <https://docs.microsoft.com/ru-ru/sql/relational-databases/database-engine-tutorials?view=sql-server-2017> – 09.12.2020.
8. <https://metanit.com/sql/sqlserver/1.3.php> – 09.12.2020.
9. <http://www.t-sql.ru> – 09.12.2020.
10. <https://www.cyberforum.ru> – 09.12.2020.

# Приложение А

# (обязательное)

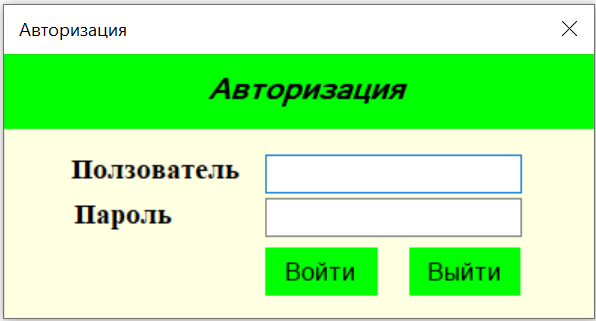


Рисунок А.1 – Форма авторизации пользователя

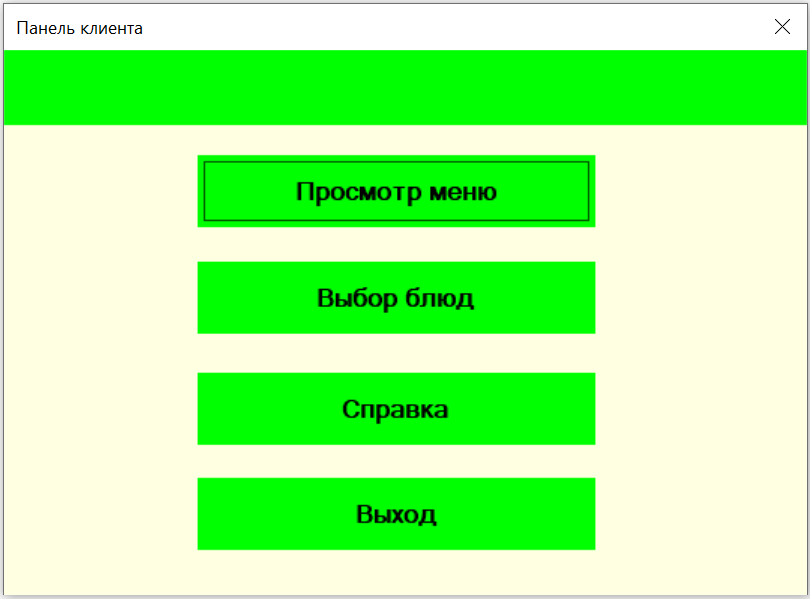


Рисунок А.2 – Форма панели клиента

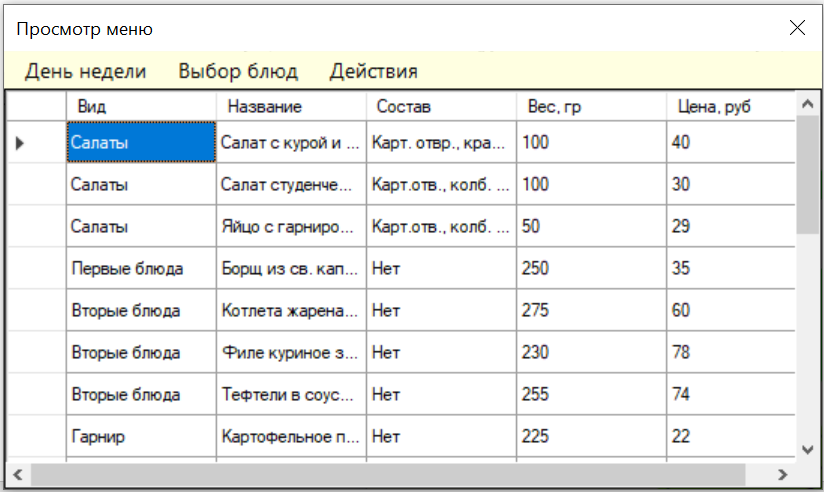


Рисунок А.3 – Форма просмотра меню



Рисунок А.4 – Форма дней недели для выбора блюд

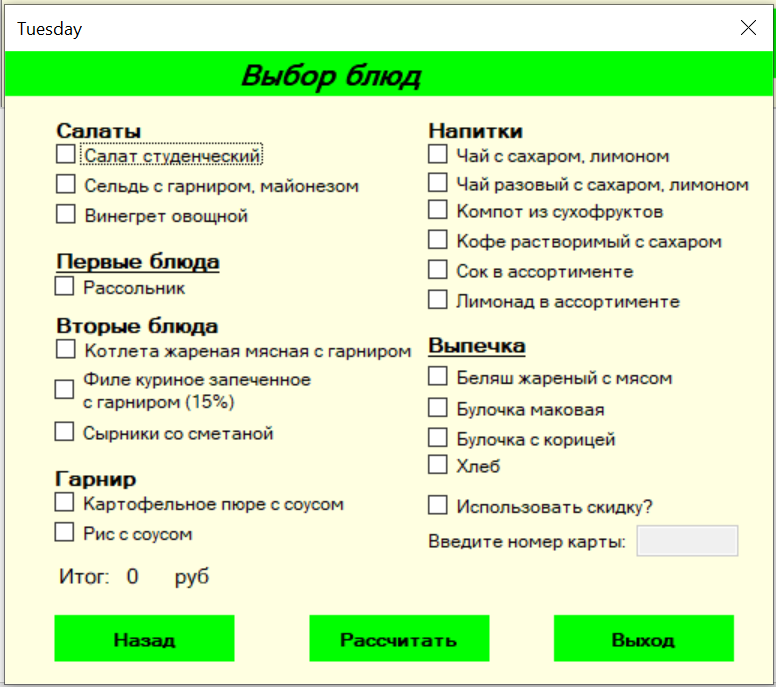


Рисунок А.5 – Форма выбора блюд

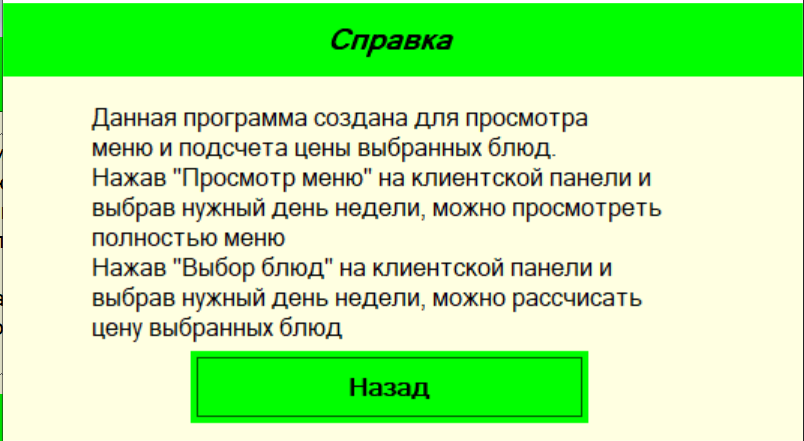


Рисунок А.6 – Форма справки



Рисунок А.6 – Форма отчета

# Приложение Б

# (обязательное)



Рисунок Б.1 – IDEFX-диаграмма меню

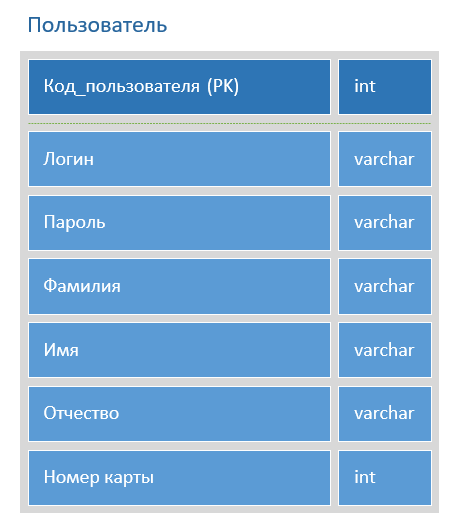


Рисунок Б.2 – IDEFX-диаграмма пользователя



Рисунок Б.3 – Диаграмма Ганта



Рисунок Б.4 – Диаграмма Ганта

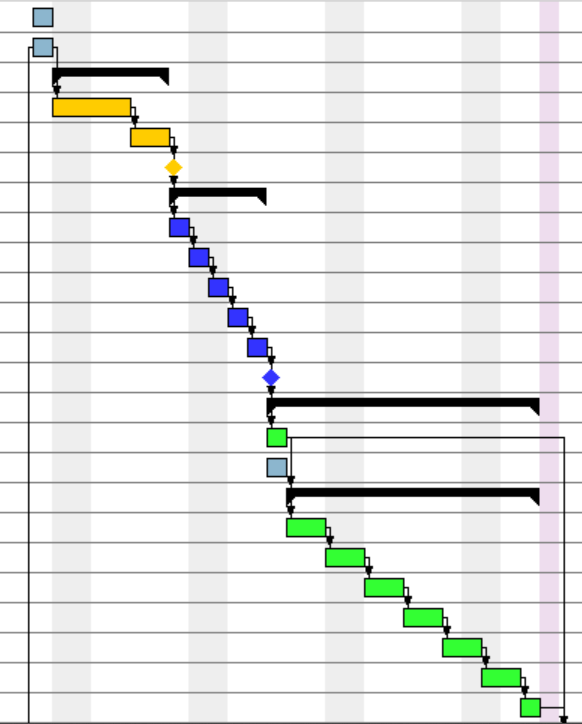


Рисунок Б.5 – Диаграмма Ганта

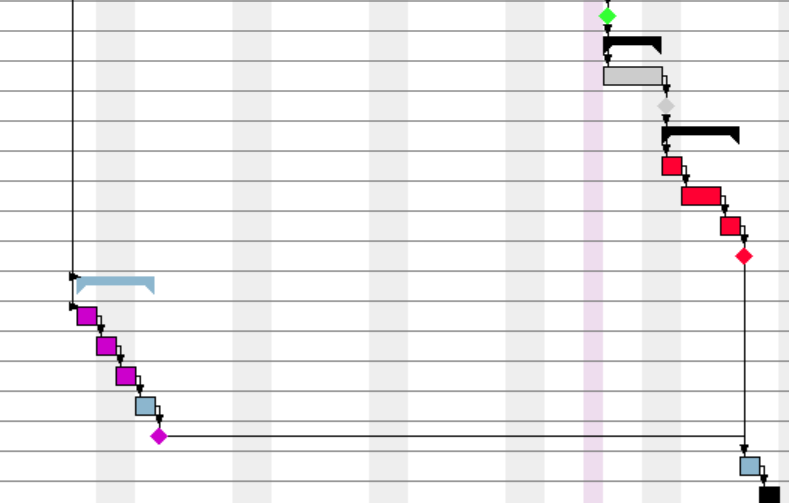


Рисунок Б.6 – Диаграмма Ганта

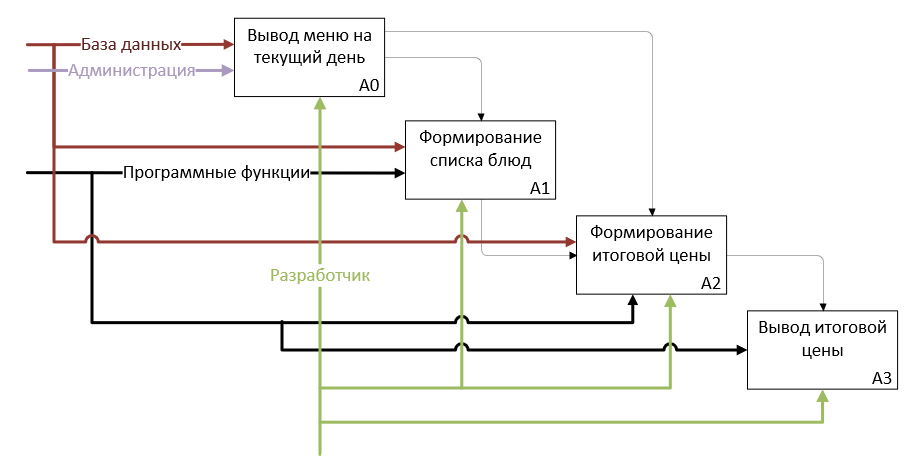


Рисунок Б.7 – Диаграмма потоков

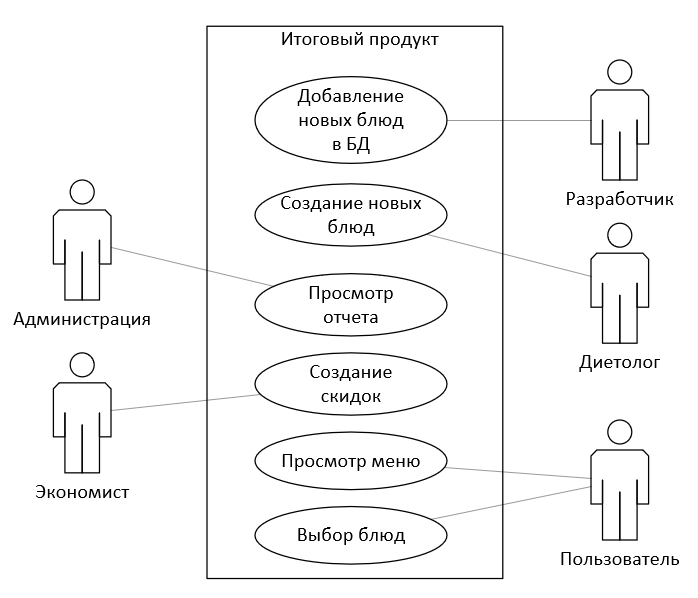


Рисунок Б.8 – USE-CASE диаграмма

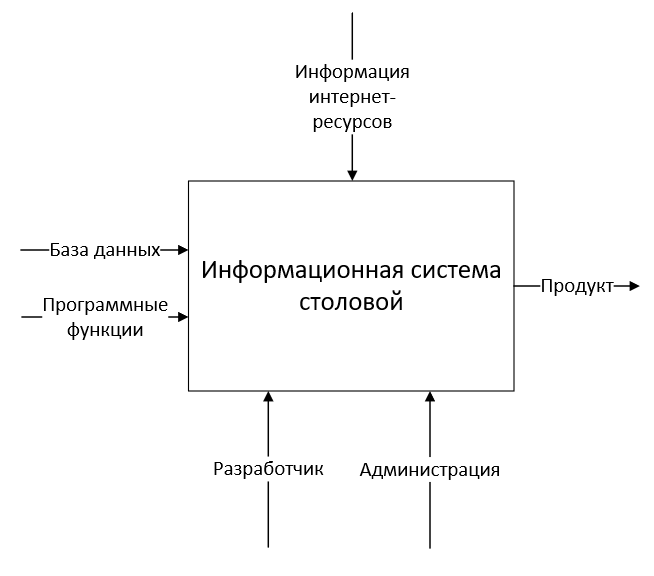


Рисунок Б.9 – IDEF0-диаграмма

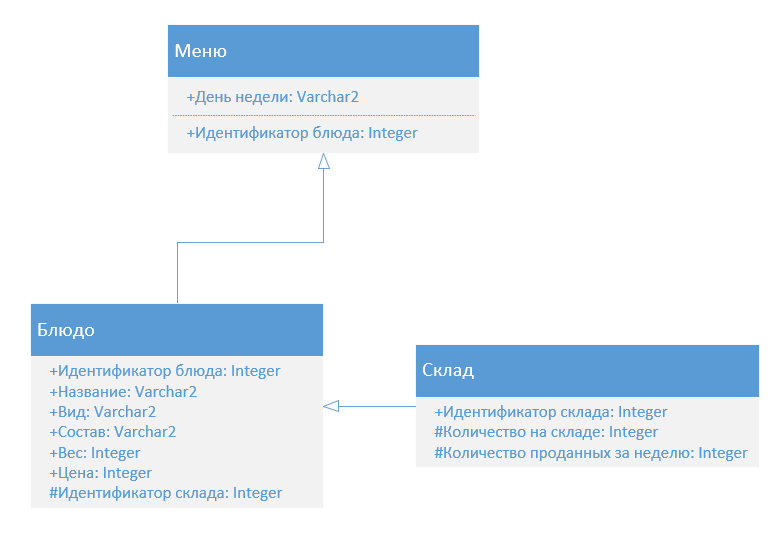


Рисунок Б.10 – Диаграмма классов

# Приложение В

# (обязательное)

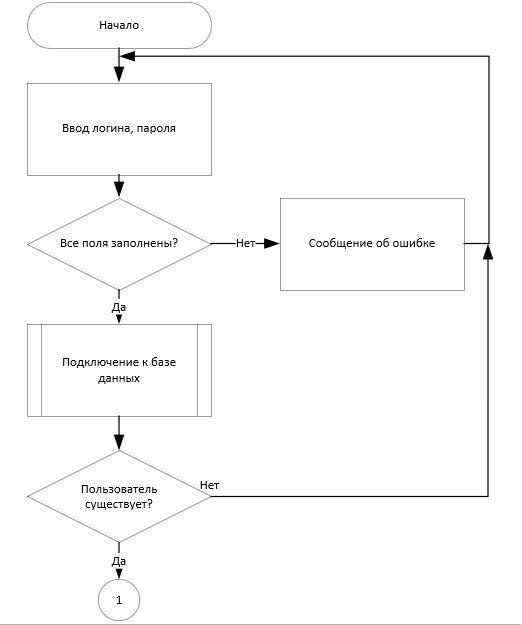


Рисунок В.1 – Блок-схема авторизация

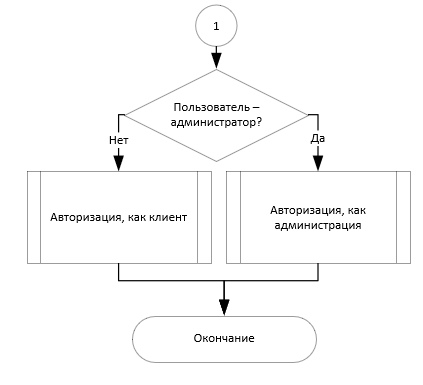


Рисунок В.2 – Блок-схема авторизация

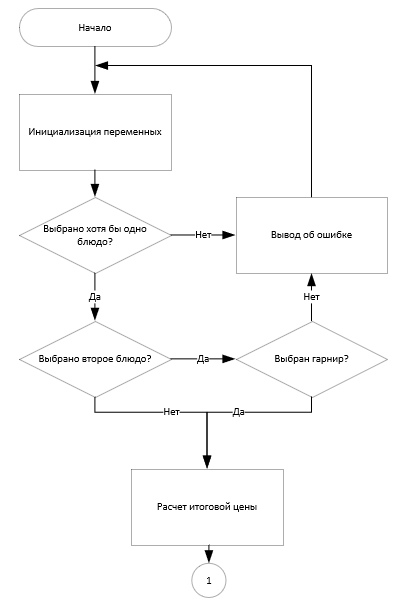


Рисунок В.3 – Блок-схема расчета итоговой цены

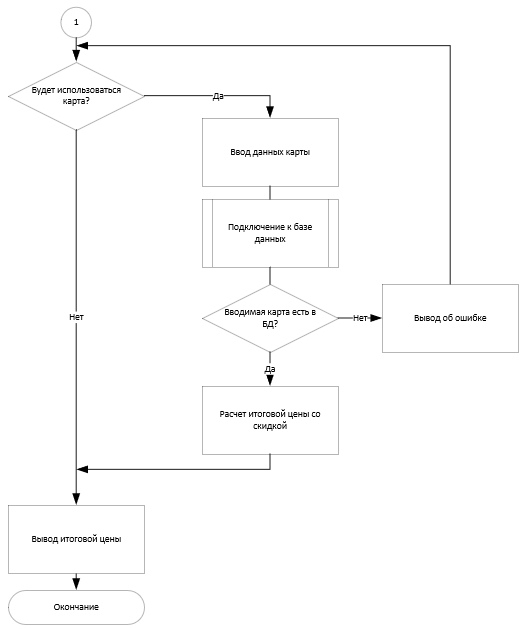


Рисунок В.4 – Блок-схема расчета итоговой цены

# Приложение Г

# (обязательное)

Г.1 Панель авторизации

SqlConnection connection = new SqlConnection

(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=canteen;Integrated Security=True");

connection.Open();

string query = "SELECT \* FROM users WHERE login = '"

+ tBUser.Text.Trim() + "' and password = '" + tBPass.Text.Trim() + "'";

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

SqlCommand cmd = new SqlCommand(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

SqlDataReader dataReader = cmd.ExecuteReader();

if (dataTable.Rows.Count == 1)

{

if (dataReader.Read())

{

firstName = (string)dataReader["first\_name"];

lastName = (string)dataReader["last\_name"];

secondName = (string)dataReader["second\_name"];

}

connection.Close();

firstName = firstName.Replace(" ", "");

lastName = lastName.Replace(" ", "");

secondName = secondName.Replace(" ", "");

string txt = "Здравствуйте, " + lastName + " " + firstName + " " + secondName + "!";

MessageBox.Show(txt);

if ((tBUser.Text == "admin") && (tBPass.Text == "admin"))

{

adfmMain adfm = new adfmMain();

adfm.Show();

this.Hide();

}

else

{

fmMain fmMain = new fmMain();

this.Hide();

fmMain.Show();

}

}

else

{

MessageBox.Show("Введенные данные не верны\nИли не введены не полностью","Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

tBPass.Clear();

}

Г.2 Расчет итоговой цены

SqlConnection connection = new SqlConnection(@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=canteen;Integrated Security=True");

string query = "SELECT \* FROM users WHERE card\_number = '" + textBox1.Text.Trim()+"'";

SqlDataAdapter adapter = new SqlDataAdapter(query, connection);

DataTable dataTable = new DataTable();

adapter.Fill(dataTable);

double sal\_1 = 0;

double sal\_2 = 0;

double sal\_3 = 0;

double soup = 0;

double secD\_1 = 0;

double secD\_2 = 0;

double secD\_3 = 0;

double g\_1 = 0;

double g\_2 = 0;

double n\_1 = 0;

double n\_2 = 0;

double n\_3 = 0;

double n\_4 = 0;

double n\_5 = 0;

double n\_6 = 0;

double n\_7 = 0;

double v\_1 = 0;

double v\_2 = 0;

double v\_3 = 0;

double v\_4 = 0;

double v\_5 = 0;

double v\_6 = 0;

double sk\_1 = 0;

double sk\_2 = 0;

//Салат с курой и ветчиной

if (checkBox1.Checked == true)

{

sal\_1 = 40.00;

}

//Салат студенческий

if (checkBox2.Checked == true)

{

sal\_2 = 30.00;

sk\_1 = 5;

}

//Яйцо с гарниром, майонезом

if (checkBox3.Checked == true)

{

sal\_3 = 29.00;

}

//Борщ из св. капусты со сметаной

if (checkBox4.Checked == true)

{

soup = 35.00;

}

//Котлета жареная мясная с гарниром

if (checkBox5.Checked == true)

{

if ((checkBox8.Checked == true) || (checkBox9.Checked == true))

{

secD\_1 = 60.00;

}

else

{

MessageBox.Show("Вы не выбрали гарнир");

}

}

//Филе куриное запеченное с гарниром

if (checkBox6.Checked == true)

{

if ((checkBox8.Checked == true) || (checkBox9.Checked == true))

{

secD\_2 = 78.00;

}

else

{

MessageBox.Show("Вы не выбрали гарнир");

}

}

//Тефтели в соусе с гарниром

if (checkBox7.Checked == true)

{

if ((checkBox8.Checked == true) || (checkBox9.Checked == true))

{

secD\_3 = 74.00;

}

else

{

MessageBox.Show("Вы не выбрали гарнир");

}

}

//Картофельное пюре с соусом

if (checkBox8.Checked == true)

{

if ((checkBox5.Checked == true) || (checkBox6.Checked == true) || (checkBox7.Checked == true))

{

g\_1 = 0;

}

else

{

g\_1 = 22.00;

}

}

//Макароны с соусом

if (checkBox9.Checked == true)

{

if ((checkBox5.Checked == true) || (checkBox6.Checked == true) || (checkBox7.Checked == true))

{

g\_2 = 0;

}

else

{

g\_2 = 22.00;

}

}

//Чай с сахаром, лимоном

if (checkBox10.Checked == true)

{

n\_1 = 6.00;

}

//Чай разовый с сахаром, лимоном

if (checkBox11.Checked == true)

{

n\_2 = 12.00;

}

//Компот ягодный

if (checkBox12.Checked == true)

{

n\_3 = 15.00;

}

//Компот из сухофруктов

if (checkBox13.Checked == true)

{

n\_4 = 15.00;

}

//Кофе растворимый с сахаром

if (checkBox14.Checked == true)

{

n\_5 = 20.00;

}

//Сок в ассортименте

if (checkBox15.Checked == true)

{

n\_6 = 20.00;

}

//Лимонад в ассортименте

if (checkBox16.Checked == true)

{

n\_7 = 50.00;

}

//Сосиска в тесте

if (checkBox17.Checked == true)

{

v\_1 = 35.00;

}

//Беляш жареный с мясом

if (checkBox18.Checked == true)

{

v\_2 = 30.00;

}

//Булочка маковая

if (checkBox19.Checked == true)

{

v\_3 = 20.00;

}

//Булочка с корицей

if (checkBox20.Checked == true)

{

v\_4 = 20.00;

}

//Пицца с колбасой

if (checkBox21.Checked == true)

{

v\_5 = 40.00;

sk\_2 = 10;

}

//Хлеб

if (checkBox22.Checked == true)

{

v\_6 = 3;

}

double sum = sal\_1 + sal\_2 + sal\_3 + soup + secD\_1 + secD\_2 + secD\_3 + g\_1 + g\_2 + n\_1 + n\_2 + n\_3 + n\_4 + n\_5 + n\_6 + n\_7 + v\_1 + v\_2 + v\_3 + v\_4 + v\_5 + v\_6;

label8.Text = sum.ToString();

Boolean checkboxFlag = false;

foreach (var control in this.Controls)

{

if (control is CheckBox)

{

if (((CheckBox)control).Checked)

{

checkboxFlag = true;;

}

}

}

if (!checkboxFlag)

{

DialogResult uncheckederror = MessageBox.Show("Вы не выбрали ни одного блюда, расчет не возможен",

"Ошибка!", MessageBoxButtons.OK, MessageBoxIcon.Exclamation,

MessageBoxDefaultButton.Button1);

}

if (textBox1.Enabled == true)

{

if (dataTable.Rows.Count == 1)

{

if((sk\_1 != 0) && (sk\_2 != 0))

{

double c = sk\_2 \* (sum / 100);

double card\_1 = sum - c;

label8.Text = card\_1.ToString();

}

else if ((sk\_1 != 0) && (sk\_2 == 0))

{

double c = sk\_1 \* (sum / 100);

double card\_1 = sum - c;

label8.Text = card\_1.ToString();

}

else if ((sk\_1 == 0) && (sk\_2 != 0))

{

double c = sk\_2 \* (sum / 100);

double card\_1 = sum - c;

label8.Text = card\_1.ToString();

}

}

else

{

MessageBox.Show("Error");

}

}