

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Обзор и анализ предметной области	5
1.1 Обзор деятельности ООО «Сандорини»	5
1.2 Обзор и анализ области внедрения	6
1.3 Обзор и анализ существующих разработок информационных подсистем управления персоналом и закупками	7
1.4 Достоинства и недостатки существующих разработок	12
2 Проектирование модуля информационной подсистемы управления персоналом и закупками для ООО «Сандорини»	17
2.1 Постановка задачи на разработку	17
2.2 Проектирование структуры программного обеспечения	17
2.3 Проектирование структуры базы данных	18
2.4 Описание основных элементов СУБД	21
2.5 Руководство пользователя и администратора	38
ЗАКЛЮЧЕНИЕ	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	42

ВВЕДЕНИЕ

Автоматизация процессов на различного рода предприятиях стала уже неотъемлемой частью, без которой нельзя представить работу ни одной компании. Ежегодно возрастает количество информации, используемой человеком в процессе его жизнедеятельности. И чем больше объём этой информации становится, тем более остро встаёт вопрос о способах хранения, обработки и автоматизации общества в целом.

Под автоматизацией предприятий общественного питания понимается процесс, в ходе которого происходит внедрение программных и аппаратных средств (комплексов), использующихся в автоматизированной работе всех или же некоторых процессов на данном предприятии.

Целью выпускной квалификационной работы является проектирование и разработка автоматизированной системы, позволяющей администратору управлять работой персонала и закупками на предприятии общественного питания ООО «Сандорини».

Для достижения цели были поставлены следующие задачи:

- осуществить сбор данных и анализ предметной области на предприятии общественного питания ООО «Сандорини»;
- осуществить обзор деятельности предприятия;
- осуществить обзор области внедрения предприятия;
- осуществить обзор и анализ существующих разработок информационных подсистем для управления работой персонала и закупками;
- спроектировать модуль информационной подсистемы для управления работой персонала и закупками для ООО «Санторини»;
- разработать «Руководство пользователя и администратора».

Объект исследования – организация работы администратора на предприятии общественного питания ООО «Сандорини», связанная с управлением персоналом и закупками.

Предмет исследования – автоматизированная система для управления персоналом и закупками на рассматриваемом предприятии.

Разрабатываемая автоматизированная информационная система должна избавить администратора от повторяющейся рутинной работы, упростить его работу с документацией на предприятии, повысить его эффективность в работе с персоналом, а также дать возможность контролировать процесс закупок. До этого момента вся эта работа проводилась в бумажном виде, что сильно усложняло работу администратору и занимало достаточно много времени.

Основными задачами по автоматизации предприятий общественного питания являются:

- повышение уровня прибыли и понижение издержек предприятия общественного питания;
- возможность контролировать, а так же оптимизировать деятельность предприятия общественного питания;
- оказание более усовершенствованного качества обслуживания посетителей;
- возможность вовремя заметить и предотвратить хищения, а так же прочие злоупотребления и нарушения со стороны работы персонала;
- повышение производительности работы персонала;
- возможность получить поддержку со стороны различных маркетинговых предприятий;
- улучшение или создание системы лояльности для постоянных клиентов (разработка различных систем скидок);
- облегчить анализ деятельности предприятия общественного питания и запланировать дальнейшее развитие и усовершенствование бизнеса.

1 Обзор и анализ предметной области

1.1 Обзор деятельности ООО «Сандорини»

ООО «Сандорини» является предприятием общественного питания. Оно предоставляет меню готовых блюд и напитков из разных кухонь мира, например, японская, итальянская, но в основном греческой кухни. В меню представлены не только блюда, приготовленные по классическим, традиционным, национальным рецептам, а также и собственные фирменные, разработанные шеф-поварами.

По организационно-правовой форме ООО «Сандорини» - общество с ограниченной ответственностью. Категория субъекта в едином реестре субъектов малого и среднего предпринимательства – микропредприятие. По ОКВЭД (в общероссийском классификаторе видов экономической деятельности) зарегистрированы всего 5 видов деятельности. Основным видом деятельности является – деятельность ресторанов и услуги по доставке продуктов питания.

Дополнительные виды деятельности:

- деятельность ресторанов и кафе с полным ресторанным обслуживанием, кафетериев, ресторанов быстрого питания и самообслуживания;
- деятельность ресторанов и баров по обеспечению питанием в железнодорожных вагонах-ресторанах и на судах;
- подача напитков;
- аренда и управление собственным или арендованным недвижимым имуществом.

На основе анализа и сбора данных можно сделать вывод, что предприятие ООО «Сандорини» имеет самый простой из всех существующих типов организационных структур – линейную организационную структуру,

представленную на рисунке 1. Все решения, кроме экстренных ситуаций, принимаются директором. Затем принятые решения переходят к линейным руководителям, в роли которых выступают администраторы и шеф-повара. Они распределяют поставленные задачи на выполнение к исполнителям в лице поваров, официантов и прочего персонала. Следовательно, ООО «Сандорини» принадлежит к типу линейных организационных структур.

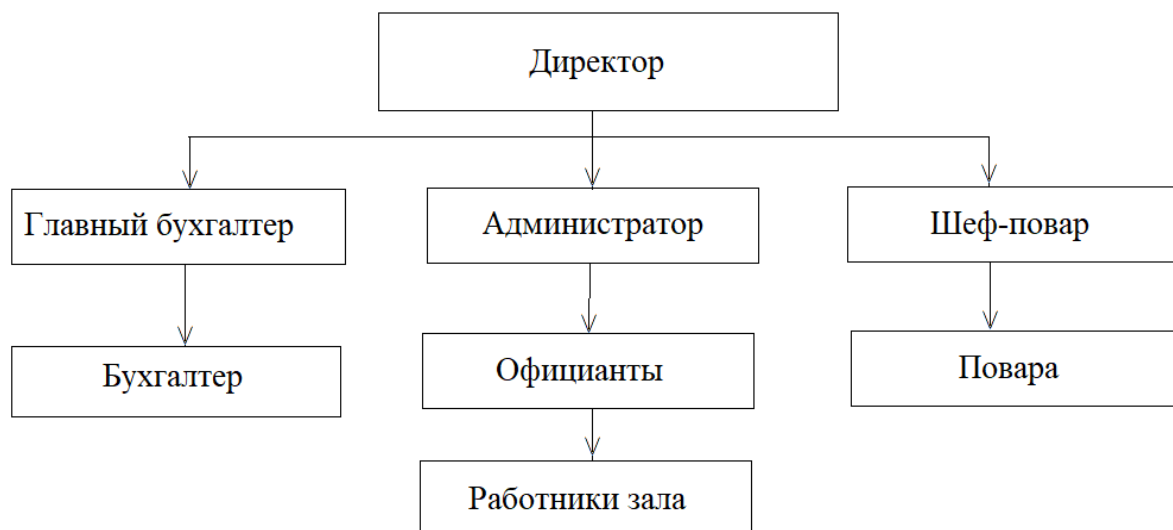


Рисунок 1 - Схема линейной организационной структуры ООО «Сандорини»

1.2 Обзор и анализ области внедрения

Всеми управленческими вопросами занимается директор. Он осуществляет контроль за всеми заведениями и за всеми работниками, а также регулирует и следит за деятельностью предприятия общественного питания ООО «Сандорини».

Также на предприятии имеется отдел бухгалтерии. Главный бухгалтер управляет деятельностью всего отдела, следит за финансами, занимается бухгалтерским учётом, считает все налоги.

За работой зала следит администратор, а за работу кухни отвечает шеф-повар. Администратор контролирует и регулирует деятельность всех официантов, уборщиков и поставщиков. А шеф-повар управляет работой сушефа, поваров всех цехов и заготовщиков.

Работа администратора заключается не только в осуществлении контроля за деятельностью зала, но он также обязан осуществлять набор персонала для работы, а также осуществление закупок. Это приводит к ежедневному рутинному заполнению одинаковых бумаг, на что тратится огромное количество времени и понижает эффективность его работы. Также, так как вся информация, информация о закупках и персонале хранятся в бумажном виде, иногда некоторые листы могут быть утеряны и приходится заполнять новые бумаги, что ещё больше отнимает времени.

Поэтому, в связи с большим документооборотом, было принято решение на проектирование и разработку автоматизированной системы для управления персоналом и закупками на предприятии общественного питания ООО «Сандорини».

1.3 Обзор и анализ существующих разработок информационных подсистем управления персоналом и закупками

Среди всех существующих приложений, разработанных для автоматизации управления персоналом и закупками основными разработками являются:

- «Трактирь»;
- «ШКО»;
- «R-keeper»;
- «Fusion POS».

Рассмотрим работу каждого из рассмотренных программных продуктов.

«Трактирь»

«Трактирь» – это целый программный комплекс, разработанный на базе 1С. Данная программа направлена на обеспечение постоянного контроля и управления деятельностью заведения. Пример окна программы представлен на рисунке 2.

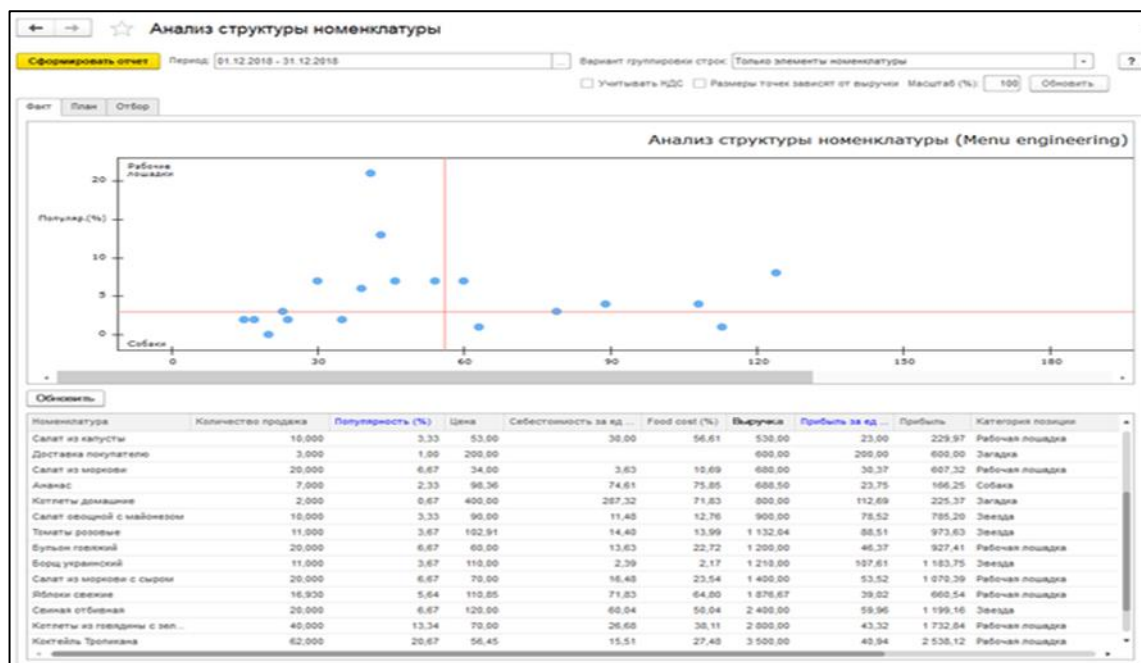


Рисунок 2 – Вид окна приложения «Трактор»

Приложение «Трактор» обеспечивает взаимодействие работы зала и работы кухни. Данное приложение позволяет работать практически со всеми наиболее популярными схемами работы предприятий общественного питания. Данный программный продукт также может использоваться для управления не только одним заведением, а целой сетью. Он обладает следующими функциями:

- учёт на складе;
- производственный учёт;
- управление закупками и финансами;
- управление персоналом.

Приложение «Трактор» среди пользователей обладает средней оценкой из-за своей сложности в использовании.

Приложение ИКО

Приложение ИКО предоставляет решение, которым могут воспользоваться заведения от самых небольших баров и кофеен до целых сетей ресторанов, что делает его довольно разноплановым. Вид окна программы представлен на рисунке 3.

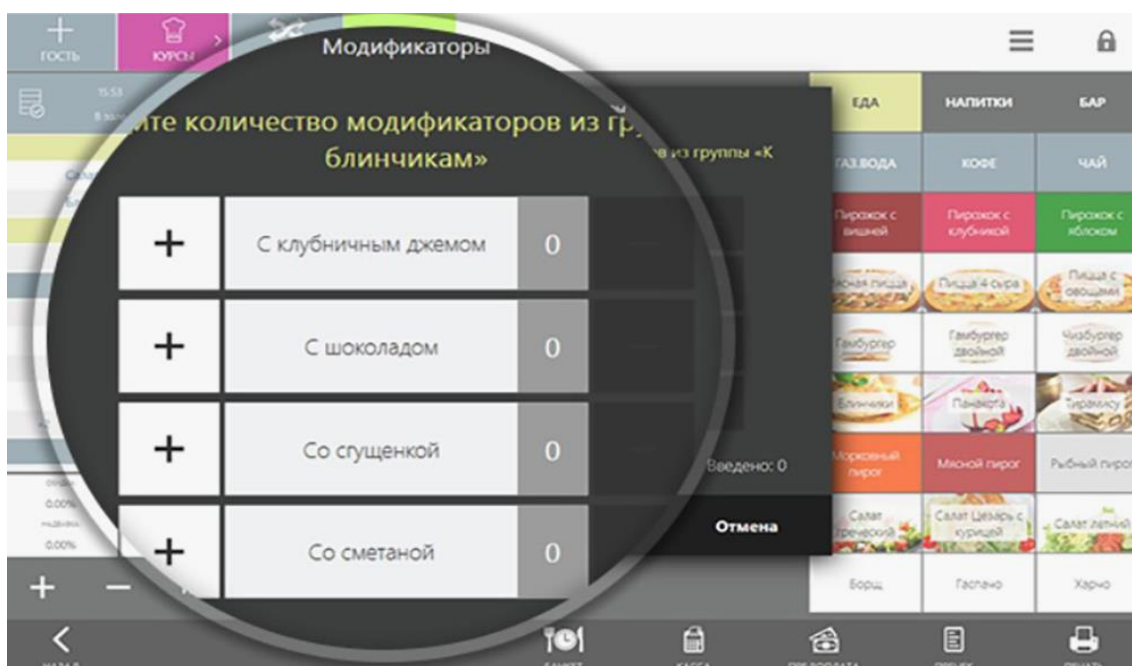


Рисунок 3 – Вид окна приложения «ПКО»

Приложение «ПКО» включает в себя такие функции:

- ведение учёта;
- есть возможность проведения аналитики и контроля;
- улучшение работы кухни.

В данном приложении есть возможность использовать специализированные модули, если есть необходимость, с их помощью можно настраивать приложение под работу конкретного заведения или сети ресторанов.

Общая оценка пользователей приложения «ПКО» - удовлетворительно. При работе «ПКО» является очень неудобным и малоэффективным решением, по мнению пользователей.

«R-keeper»

«R-keeper» – это приложение для удобного и эффективного управления ресторанным бизнесом. В ней есть несколько программных модулей, которые делают работу с данной программой более комфортной. Также R-keeper позволяет добиться максимально быстрой скорости для выполнения заказов при помощи сервиса.

Автоматизация ресторана с использованием данного приложения оптимизирует не только работу заведения, но и расходы. Приложение «R-keeper» обладает следующими функциями:

- автоматизация ведения кассы;
- деятельность, связанная с документацией;
- формирование бонусов, акций и скидок;
- учёт на складе;
- автоматизация доставки;
- формирование отчётности;
- управление персоналом.

Общая оценка пользователей программы «R-keeper»: простота использования, служба поддержки, функциональность – все эти критерии хорошо оцениваются пользователями. Также данное приложение является не очень дорогим, что делает его достаточно привлекательным решением по мнению пользователей. Пример окна программы представлен на рисунке 4.

показатель	период для сравнения 01.04.2020 - 30.04.2020 (30 дн.)	отчетный период 01.04.2021 - 30.04.2021 (30 дн.)	динамика (абсол. знач.)	динамика, %
Сумма	22 517 846,5	26 842 512	+4 324 665,5 ↑	+19,2% ↑
Выручка	23 007 054,5	27 071 529,5	+4 064 475 ↑	+17,7% ↑
Итого оплачено	23 007 054,5	27 071 529,5	+4 064 475 ↑	+17,7% ↑
Количество заказов	6 832	11 179	+4 347 ↑	+63,6% ↑
Среднее количество строк в чеке	4,8	3,5	-1,2 ↓	-26,1% ↓
Средний чек за заказ	3 367,5	2 421,6	-945,9 ↓	-28,1% ↓
Среднее количество гостей в заказе	1,8	1,3	-0,5 ↓	-26,4% ↓
Количество гостей	12 194	14 686	+2 492 ↑	+20,4% ↑
Средний чек на гостя	1 886,8	1 843,4	-43,4 ↓	-2,3% ↓
Длительность визита	01:06:56	00:48:59	-00:17:57	-26,8% ↓

Рисунок 4 – Вид окна приложения «R-keeper»

«Fusion POS»

Приложение «Fusion POS» подходит для автоматизации предприятий общественного питания любой направленности. «Fusion POS» была разработана специально для автоматизации работы малого ресторанного бизнеса и сетевых заведений.

The screenshot displays the 'Fusion POS' application interface. On the left is a dark sidebar menu with icons and labels for various functions: 'ТОЧКА ПРОДАЖ' (Point of Sale), 'Статистика точки' (Point statistics), 'Заказы' (Orders), 'Склад' (Warehouse), 'Отчеты' (Reports), 'Настройки точки' (Point settings), 'СЕТЬ ПРОДАЖ' (Network of sales), 'Статистика сети' (Network statistics), 'Справочники' (Reference), 'Финансы' (Finance), 'Система лояльности' (Loyalty system), and 'Настройки сети' (Network settings). The main area is titled 'Создание приходной накладной' (Creating an incoming receipt) and includes a breadcrumb trail: 'Fusion POS > Точка продаж > Склад > Накладные > Приходные'. A 'ВЫБРАТЬ ШАБЛОН' (Select template) button is in the top right. The form contains several input fields: 'Номер накладной' (Receipt number) with the value '1', 'Поставщик' (Supplier) with a dropdown menu, 'Получатель' (Receiver) with a dropdown menu, 'Сумма' (Sum) with the value '50000' and a ruble symbol, 'Дата накладной поставщика' (Supplier receipt date) with the value '07.03.2022 09:42', 'Дата прихода' (Receipt date) with the value '08.03.2022 09:42', and a 'Комментарий' (Comment) text area. At the bottom of the form are four buttons: 'Сохранить' (Save), 'Сохранить в черновик' (Save as draft), 'Сохранить как шаблон' (Save as template), and 'Удалить' (Delete). Below the form, the text 'ТОВАРЫ НАКЛАДНОЙ' (Receipt items) is visible, followed by a magnifying glass icon.

Рисунок 5 – Вид окна приложения «Fusion POS»

В приложении имеется два режима: режим кафе и режим фаст-фуда (fast-food), которые позволяют повысить эффективность работы заведений разных схем ведения ресторанного бизнеса. В приложении имеются такие функции как:

- работа в двух режимах;
- возможность проведения аналитики, ведения отчётности и просмотра графиков;
- управление меню;
- учёт склада;
- расчёт бонусов, скидок, акций.

Приложение «Fusion POS» пользователи оценивают лучше всего. Практически все оценки отличные, кроме службы поддержки. Также важной деталью является то, что данное приложение работает на многих известных платформах и может работать не только с использованием интернета, но и при его отсутствии. Эта программа выпущена сравнительно недавно по сравнению с остальными (2014 год), в ней отсутствуют некоторые функции, которые бы хотели видеть пользователи. Но, по сравнению с другими, она становится одним из самых выгодных решений по автоматизации предприятия общественного питания.

1.4 Достоинства и недостатки существующих разработок

У всех существующих разработок по управлению персоналом и закупками, рассмотренных выше есть свои преимущества и недостатки. На их основании можно сделать выводы, что требуется включить в автоматизированную систему для ООО «Сандорини».

Программа «Трактор»

Достоинства:

- есть возможность для обмена данными с ЕГАИС;
- можно заказать демодоступ (пробная версия, недолгосрочная) для любой интересующей функции через сайт;
- есть различные решения для бизнес-процессов на предприятии общественного питания;
- возможность сформировать приложение для конкретных целей, приобретая только необходимые для работы функции;
- система помогает наладить взаимодействие между кухней и залом;
- корректный подсчёт оплаты и начисления скидки.

Недостатки:

- у программного продукта плохая коммуникация со службой поддержки;

- данная программа довольно сложна для самостоятельной настройки;
- каждая функция и её установка приобретаются отдельно, программа стоит довольно дорого.

Программа «ПКО»

Достоинства:

- есть возможность работать с сетью ресторанов;
- можно установить программу не только локально, но и в облаке;
- присутствуют дополнительные функции для ресторанов с концепцией быстрого обслуживания и ресторанов с обслуживанием столов;
- используются технологии искусственного интеллекта.

Недостатки:

- плохая коммуникация со службой поддержки;
- реализуется через интернет и/или облачное хранилище, поэтому при отсутствии интернета (или, например, плохой погоде) система перестаёт работать или работает с большой задержкой, что вызывает трудности;
- неудобный интерфейс.

Программа «R-keeper»

Достоинства:

- позволяет оптимизировать все расходы и распределить затраты в заведениях любого типа;
- позволяет удобно осуществлять закупки;
- имеет четыре тарифа для разного рода заведений;
- увеличивает скорость выполнения заказов;
- имеет демонстрационную версию личного кабинета на сайте.

Недостатки:

- медленная техническая поддержка;
- для получения пробной версии нужно отправить запрос менеджеру на самостоятельное опробование или проведение презентации системы.

Программа «Fusion POS»

Достоинства:

- работах в нескольких доступных режимах;
- возможность проведения аналитики деятельности заведения;
- возможность работы со складом и с накладными;
- имеет мобильное приложение.

Недостатки:

- отсутствие надёжной безопасности и конфиденциальности данных;
- программа не входит в единый реестр российских программ;
- отсутствуют уведомления об изменениях в программе;
- отсутствует резервное копирование в нескольких местах, что грозит потерей данных.

Назначением данной выпускной квалификационной работы является написание приложения для автоматизации работы администратора на предприятии общественного питания ООО «Сандорини».

На данный момент отсутствуют какие-либо проектные решения для данной проблемы в заведении. Рассмотрев уже существующие на рынке разработки, было принято решение спроектировать и разработать автоматизированную систему для управления персоналом и закупками для ООО «Сандорини». Приложение должно быть интуитивно понятным и иметь удобный интерфейс.

Для разработки программного продукта используется Microsoft Visual Studio, с помощью платформы Windows Presentation Foundation (WPF). Microsoft использует Microsoft. Она включает в себя интегрированную среду для разработки программного продукта, а также большое количество других средств, которые могут понадобиться разработчику. Windows Presentation Foundation (WPF) помогает создавать пользовательские интерфейсы для клиентских приложений. Так же при помощи модели ADO.NET можно легко связать между собой базу данных из SQL Server Management Studio и программный продукт Windows Presentation Foundation (WPF).

База данных для разрабатываемого приложения будет храниться в системе управления базами данных SQL Server Management Studio, которая

является одной из самых распространённых и удобных СУБД (система управления базами данных).

Функциональные требования

Программный продукт должен реализовывать такие функции как:

- вход в программу должен осуществляться по фамилии администратора;
- вход в программу должен осуществляться по паролю, который будет выдаваться главным бухгалтером;
- отображение данных в виде таблиц: «Сотрудники», «Должность», «Заведение», «Продукты», «Телефон», «Адрес», «Поставщик»;
- добавление данных в основные таблицы: «Сотрудники», «Должность», «Заведение», «Продукты»;
- удаление данных из основных таблиц: «Сотрудники», «Должность», «Заведение», «Продукты»;
- изменение данных в основных таблицах: «Сотрудники», «Должность», «Заведение», «Продукты»;
- возможность построения графиков на основе уже существующих данных, взятых из таблиц;
- поиск в основных таблицах: «Сотрудники», «Должность», «Заведение», «Продукты».

Каждый программный продукт разрабатывается для выполнения определённых функций. При согласовании функциональных требований нужно учитывать, что они должны быть наиболее подробными, чтобы в дальнейшем можно было точно оценить работы по срокам и стоимости, которая должна быть произведена перед разработкой технического задания на создание программного продукта. Если на последующих этапах разработки программного продукта не возникнет корректировок и дополнений к изначально сформулированным требованиям к функциональности, то эта оценка получится наиболее точной.

Требования к аппаратным средствам

Для работы автоматизированной системы можно установить следующие минимальные требования к аппаратным средствам:

- наличие клавиатуры;
- наличие манипулятора типа «мышь»;
- объём жёсткого диска не менее 6 Гб;
- объём оперативной памяти не менее 2 Гб (рекомендуется 8 Гб);
- процессор с тактовой частотой не ниже 1,8 ГГц;
- видеоадаптер с минимальным разрешением 1280 на 720 пикселей (рекомендуется для оптимальной работы разрешение 1366 на 768 пикселей или более высокое);
- наличие SQL Server Managment Studio (рекомендуется для достижения оптимальной производительности не менее 4 Гб);
- наличие Microsoft Visual Studio (для установки потребуется от 20 до 50 Гб свободного места).

2 Проектирование модуля информационной подсистемы управления персоналом и закупками для ООО «Сандорини»

2.1 Постановка задачи на разработку

На этапе постановки задачи на разработку программного продукта главной целью является формулировка (постановка) назначения программного обеспечения и определение основных требований к нему. На данном этапе для программного решения должны быть описаны:

- функциональные требования (определяющие функции, которые должны выполняться программным продуктом);
- эксплуатационные требования (определяющие характеристики для функционирования программного продукта).

Для проектирования и разработки автоматизированной системы для управления персоналом и закупками на предприятии общественного питания ООО «Сандорини» будет использована однопользовательская архитектура.

2.2 Проектирование структуры программного обеспечения

Проектированием структуры программного обеспечения называется такое «разбиение» программного продукта на небольшие модули (подпрограммы или части программы), которые будут работать независимо друг от друга. Данное разбиение программы на подпрограммы производится с целью упрощения программы.

Модуль - это часть программы (подпрограмма), которая хранится в отдельном файле и может транслироваться отдельно от других частей программы. Три основных атрибута модуля:

первый – логика это внутренний алгоритм подпрограммы, описывает то, как модуль выполняет его функцию;

второй – функция – то, что делает этот модуль, какие действия реализуются, его внешнее описание;

- третий - контекст – то, как конкретно применяется эта часть подпрограммы.

В программном обеспечении для управления персоналом и закупками на предприятии общественного питания ООО «Сандорини» имеются два основных модуля:

- окно для авторизации;
- окно с наименованием «Главная».

В окне авторизации реализуется функция проверки пароля и логина из базы данных, что является основной его функцией. В окне с наименованием «Главная» расположены кнопки для перехода по страницам и объект, в котором они отображаются. Таким образом, его основной функцией является отображение пользователю страниц с данными, которые хранятся в базе данных и непосредственно в самом приложении.

2.3 Проектирование структуры базы данных

Для проектирования структуры базы данных необходимо определить, какие данные должны храниться в этой базе данных. В базе данных, для разрабатываемого модуля, должны храниться следующие данные:

- информация о всех сотрудниках на предприятии общественного питания;
- номера телефонов сотрудников, которые будут храниться в отдельной таблице, на случай, если у человека будет больше одного действующего номера телефона;
- должности (их наименование и зарплата);
- наименования кафе данной сети;
- отдельная таблица с адресами (для нормализации и корректного отображения данных);

- наименование поставщиков (чтобы можно было анализировать данные о поставляемой продукции, её качестве, цене);
- данные о продуктах (для осуществления своевременных закупок).

На рисунке 6 представлена диаграмма прецедентов, в которой отображены все виды возможных прецедентов для каждого пользователя.

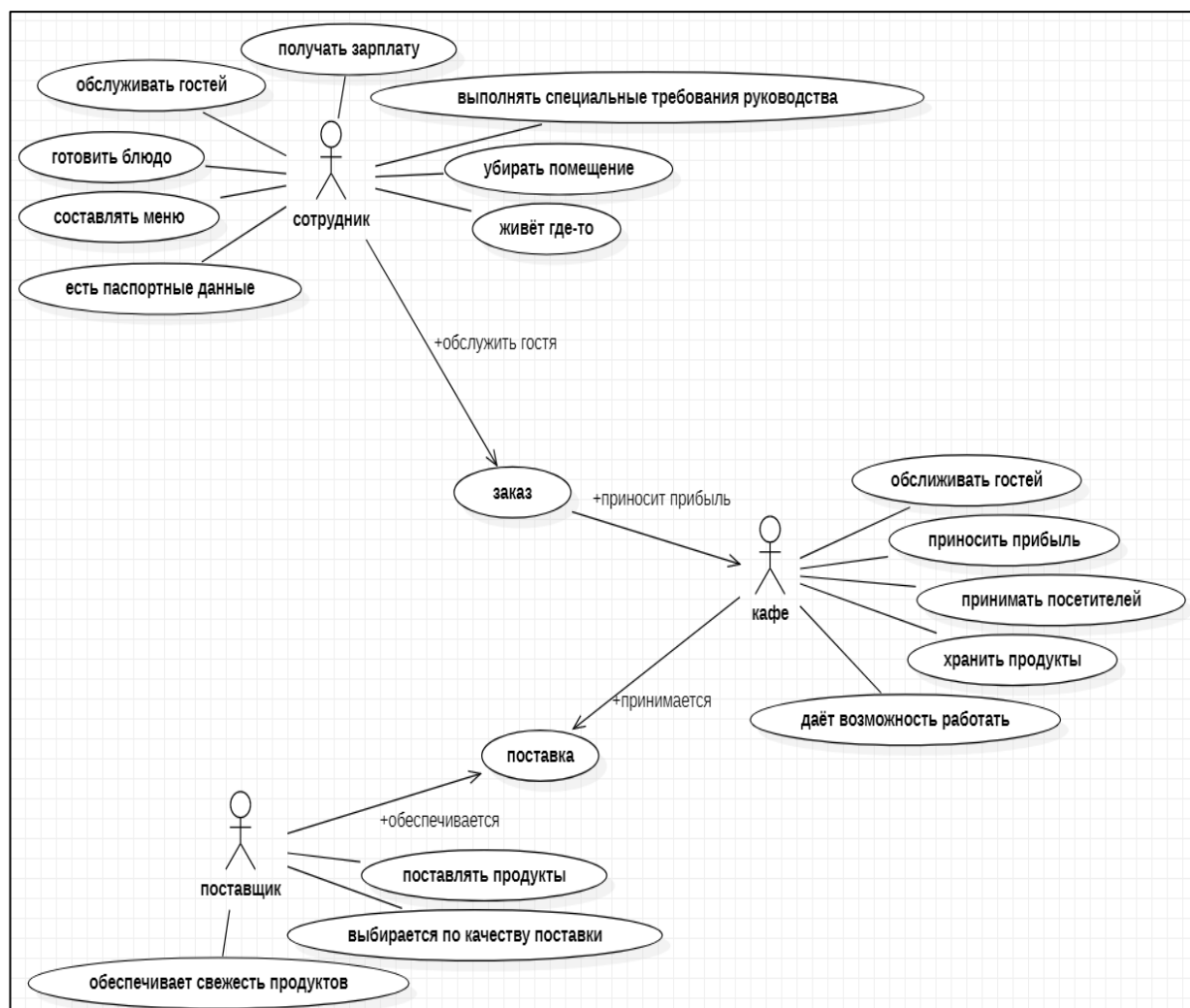


Рисунок 6 – Диаграмма прецедентов разрабатываемого программного модуля

Дале была разработана логическая схема базы данных, представленная на рисунке 7, которая описывает необходимые сущности (таблицы) их атрибуты и ограничения, уточняется состав сущностей (таблиц), а также то, как они будут взаимосвязаны между собой.

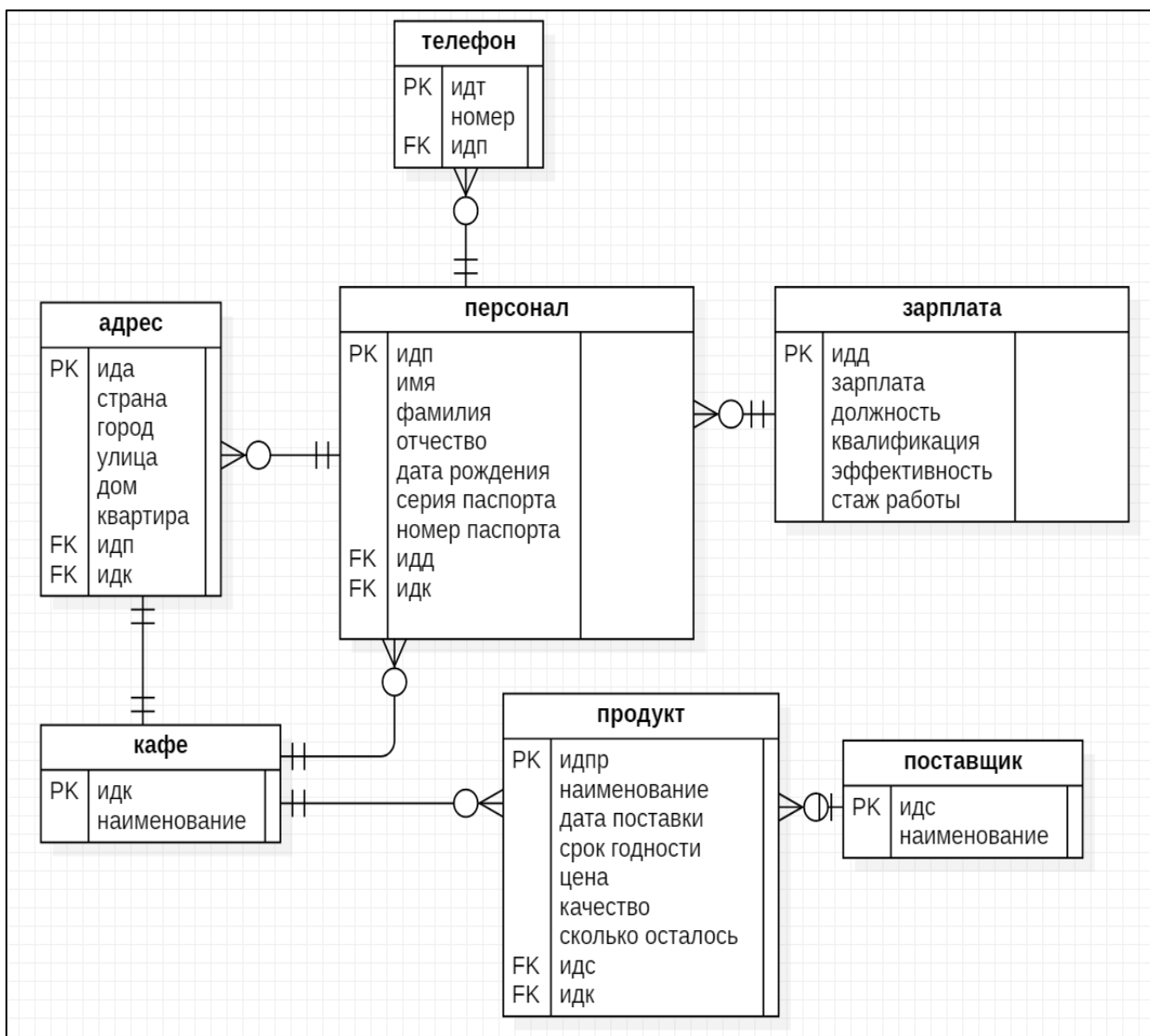


Рисунок 7 - Схема логической модели базы данных

После логического проектирования была разработана физическая модель базы данных в среде MS Management Studio, представленная на рисунке 8.

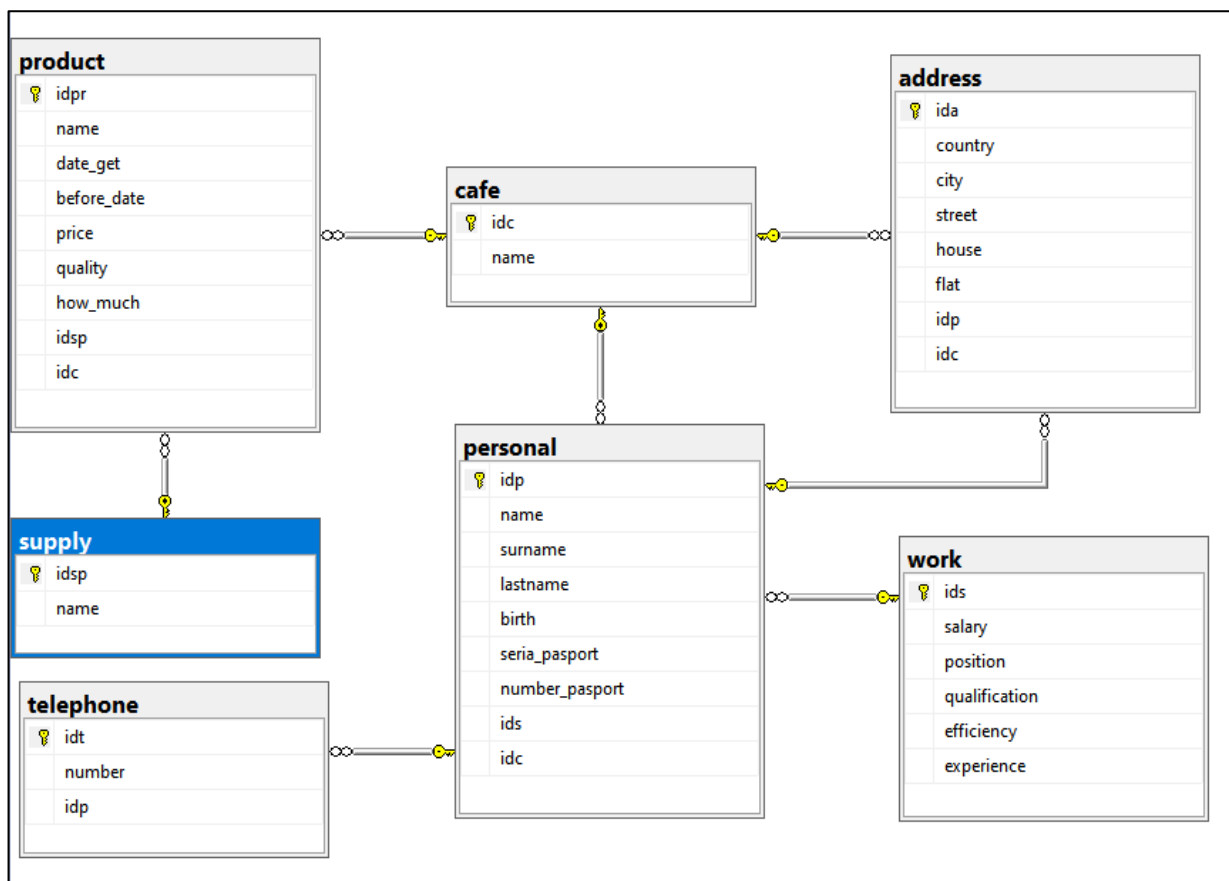


Рисунок 8 – Диаграмма разработанной базы данных

2.4 Описание основных элементов СУБД

После разработки базы данных было спроектировано и разработано, непосредственно, само приложение для автоматизированного управления персоналом и закупками на предприятии общественного питания ООО «Сандорини», описание работы которого представлено ниже.

При запуске приложения открывается окно авторизации, представленное на рисунке 9. Для того, чтобы осуществить авторизацию и перейти на следующее окно в приложении необходимо в поле для логина ввести свою фамилию, а в поле ниже ввести пароль.



Рисунок 9 – Вид окна «Авторизация»

После ввода фамилии и пароля пользователю необходимо нажать кнопку «Войти» (рисунок 10), метод в программе сверит полученные данные, введенные в текстовые поля (рисунок 11) с данными, которые хранятся в базе данных.



Рисунок 10 - Кнопка для перехода на следующее окно программы

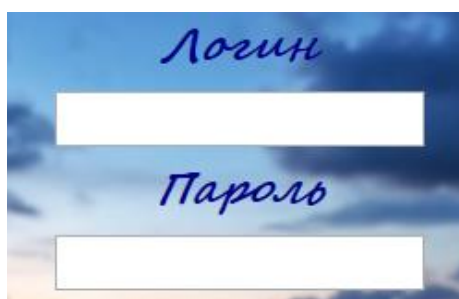


Рисунок 11 - Текстовые блоки для ввода данных и поля для отображения текста

На рисунке 12 представлен метод для обработчика кнопки перехода на следующее окно. Данные из одного текстового поля для ввода логина сравниваются с фамилиями, которые находятся в таблице персонал. А данные из текстового поля для ввода пароля сравнивает полученные данные с правильным паролем.

```
private void Enter_Click(object sender, RoutedEventArgs e)
{
    //осуществляем вход в приложение по логину и паролю, которые получаем из базы данных
    string log = login.Text;
    string pass = password.Password;
    if ((context.personal.Where(z => z.surname == log).Count() != 0) && (pass == "admin") && (context
    {
        //переход на окно главная
        Window z = new главная();
        ... this.Close();
        ... z.ShowDialog();
    }
    else
    {
        //если логин или пароль неверные, то переход не осуществится и вылетит сообщение об ошибке
        MessageBox.Show("Данные введены неверно", "OK", MessageBoxButton.OK, MessageBoxImage.Warning);
    }
}
```

Рисунок 12 - Метод для обработчика кнопки перехода на следующее окно

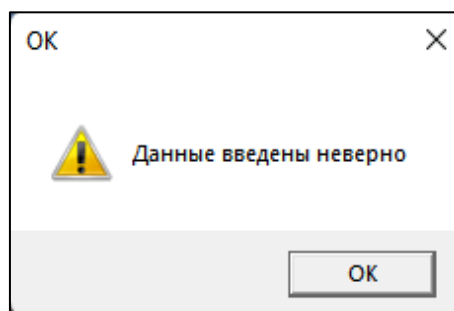


Рисунок 13 - Диалоговое окно с сообщением об ошибке при вводе данных

Если пароль или фамилия введены неверно, программа выдаст обрабатываемую ошибку: «Данные введены неверно», как показано на рисунке 13. Если в программу попытается войти человек, который не является администратором, также система выдаст сообщение об ошибке.

На рисунке 14 приведен пример разметки окна авторизации. Фоновым изображением на окне «Авторизация» выступает фотография, которая хранится в файлах приложения.

```
<Grid>
  <Image x:Name="photo" Source="/photos/grace-santorini-hotel-44-2048x1536.jpg" Stretch
    <Image.Effect>
      <BlurEffect/>
    </Image.Effect>
  </Image>
  <Label HorizontalAlignment="Center" VerticalAlignment="Top" Margin="20">Добро пожало
  <StackPanel HorizontalAlignment="Center">
    <Label Margin="0 80 0 0" HorizontalContentAlignment="Center">Логин</Label>
    <TextBox x:Name="login" Height="30" Width="200" Margin="0 0 0 0" HorizontalConte
    <Label Margin="0 0 0 0" HorizontalContentAlignment="Center">Пароль</Label>
    <PasswordBox x:Name="password" Height="30" Width="200" HorizontalContentAlignmen
    <Button x:Name="enter" Click="Enter_Click" Width="200" Margin="0 20 0 0">Войти</
  </StackPanel>
  <Image HorizontalAlignment="Left" ToolTip="Santorini" VerticalAlignment="Bottom" Wid
  <Button x:Name="about" FontSize="18" HorizontalAlignment="Right" VerticalAlignment="
</Grid>
</Window>
```

Рисунок 14 - Пример разметки окна авторизации

Содержимое окна авторизации находится в специальном контейнере «Grid», но также, можно увидеть, что для удобства расположения некоторых элементов в вертикальный столбец был применён контейнер «StackPanel», расположенный по центру окна. Также в данной разметке можно увидеть, что в окне добавлены такие элементы как:

- кнопка для входа;
- два поля для отображения текста;
- два поля для ввода пароля и логина;
- две фотографии, одна из которых является фоном данного окна.

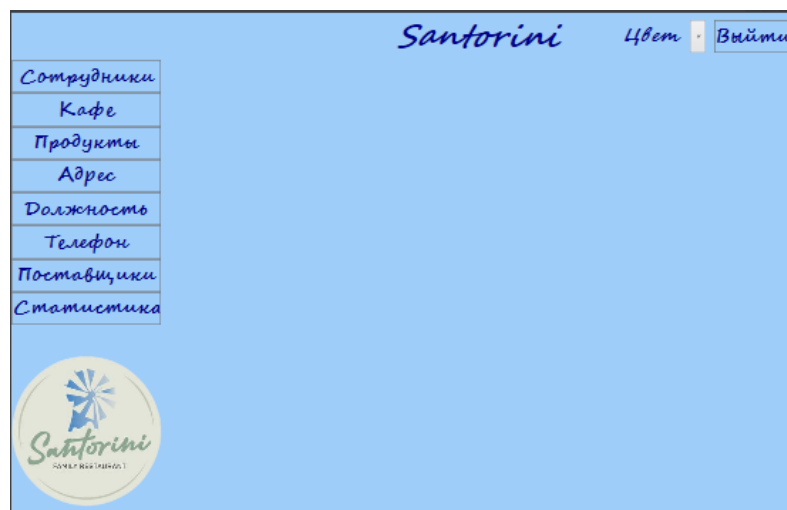


Рисунок 15 – Вид окна «Главная»

Если данные при авторизации были введены верно, то окно авторизации закроется и откроется следующее окно – «Главная», вид которого представлен на рисунке 15.

Данное окно разделено на две строки и два столбца. Большую часть окна занимает пустая панель, на которую будут выводиться страницы с таблицами. Также в этом окне расположены:

- фотография с логотипом предприятия общественного питания ооо «сандорини», представленная на рисунке 16;



Рисунок 16 - Логотип ООО «Сандорини»

- поле для отображения текстовой информации, представленная на рисунке 17, где отображено название заведений «santorini»;

Santorini

Рисунок 17 - Поле для отображения текстовой информации

– кнопки для перехода по страницам с таблицами, представленные на рисунке 18;



Рисунок 18 - Кнопки для перехода по страницам

– кнопка для возврата обратно на окно с авторизацией, представленная на рисунке 19.

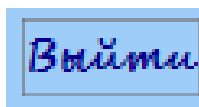


Рисунок 19 - Кнопка для возврата на окно «Авторизация»

Каждому элементу присвоено имя, что позволяет обращаться к конкретному объекту из кода программы. Главным и центральным элементом этого окна является «Frame», в котором и будут отображаться страницы с таблицами.

Кнопки находятся на специальной панели, которая позволяет расположить кнопки вертикально друг за другом. Это оптимизирует код, позволяя сократить его. Разметка данной панели с кнопками представлена на рисунке 20.

```

<Button x:Name="exit" Grid.Row="0" Grid.Column="1" HorizontalAlignment="Right" Click="Exit_Click">Выход</Button>
<StackPanel x:Name="panel" Grid.Row="1" Grid.Column="0">
    <Button x:Name="personal" Click="Personal_Click">Сотрудники</Button>
    <Button x:Name="cafe" Click="Cafe_Click">Кафе</Button>
    <Button x:Name="product" Click="Product_Click">Продукты</Button>
    <Button x:Name="adress" Click="Adress_Click">Адрес</Button>
    <Button x:Name="work" Click="Work_Click">Должность</Button>
    <Button x:Name="telephone" Click="Telephone_Click">Телефон</Button>
    <Button x:Name="postavka" Click="Postavka_Click">Поставщики</Button>
    <Button x:Name="statistic" Click="Statistic_Click">Статистика</Button>

```

Рисунок 20 – Разметка панели, в которой расположены кнопки

При нажатии кнопки «Выйти» приложение закроет текущее окно, а затем снова откроет окно для авторизации, реализация данного метода представлена на рисунке 21.

```

private void exit_Click(object sender, RoutedEventArgs e)
{
    //возвращаемся на окно авторизации
    var z = new MainWindow();
    this.Close();
    z.ShowDialog();
}

```

Рисунок 21 – Реализация метода для кнопки «Выйти»

В окне «Главная» находится пустая панель «Frame», которая используется для отображения в ней страниц с таблицами. Методы переходов на страницы «Кафе», «Продукты», «Адрес», «Сотрудники», «Телефон» по нажатию на соответствующие кнопки в окне приложения представлен на рисунке 22. В коде осуществляется переход по страницам «Frame», позволяющим просматривать таблицы базы данных. Каждая кнопка отвечает за открытие отдельной страницы.

```

ссылка: 1
private void Cafe_Click(object sender, RoutedEventArgs e)
{
    frame.NavigationService.Navigate(new Cafe());
}

ссылка: 1
private void Product_Click(object sender, RoutedEventArgs e)
{
    frame.NavigationService.Navigate(new Product());
}

ссылка: 1
private void Address_Click(object sender, RoutedEventArgs e)
{
    frame.NavigationService.Navigate(new Address());
}

ссылка: 1
private void Work_Click(object sender, RoutedEventArgs e)
{
    frame.NavigationService.Navigate(new Work());
}

ссылка: 1
private void Telephone_Click(object sender, RoutedEventArgs e)
{
    frame.NavigationService.Navigate(new Telephone());
}

```

Рисунок 22 - Методы переходов на страницы приложения

На рисунке 23 представлена схема модели, которая обеспечивает связь программного продукта Windows Presentation Foundation (WPF) в Microsoft Visual Studio с базой данных в SQL Server Managment Studio. Этот процесс называется интеграцией данных. В этом случае две среды обмениваются между собой данными благодаря модели данных ADO.NET, которая обеспечивает эту интеграцию.

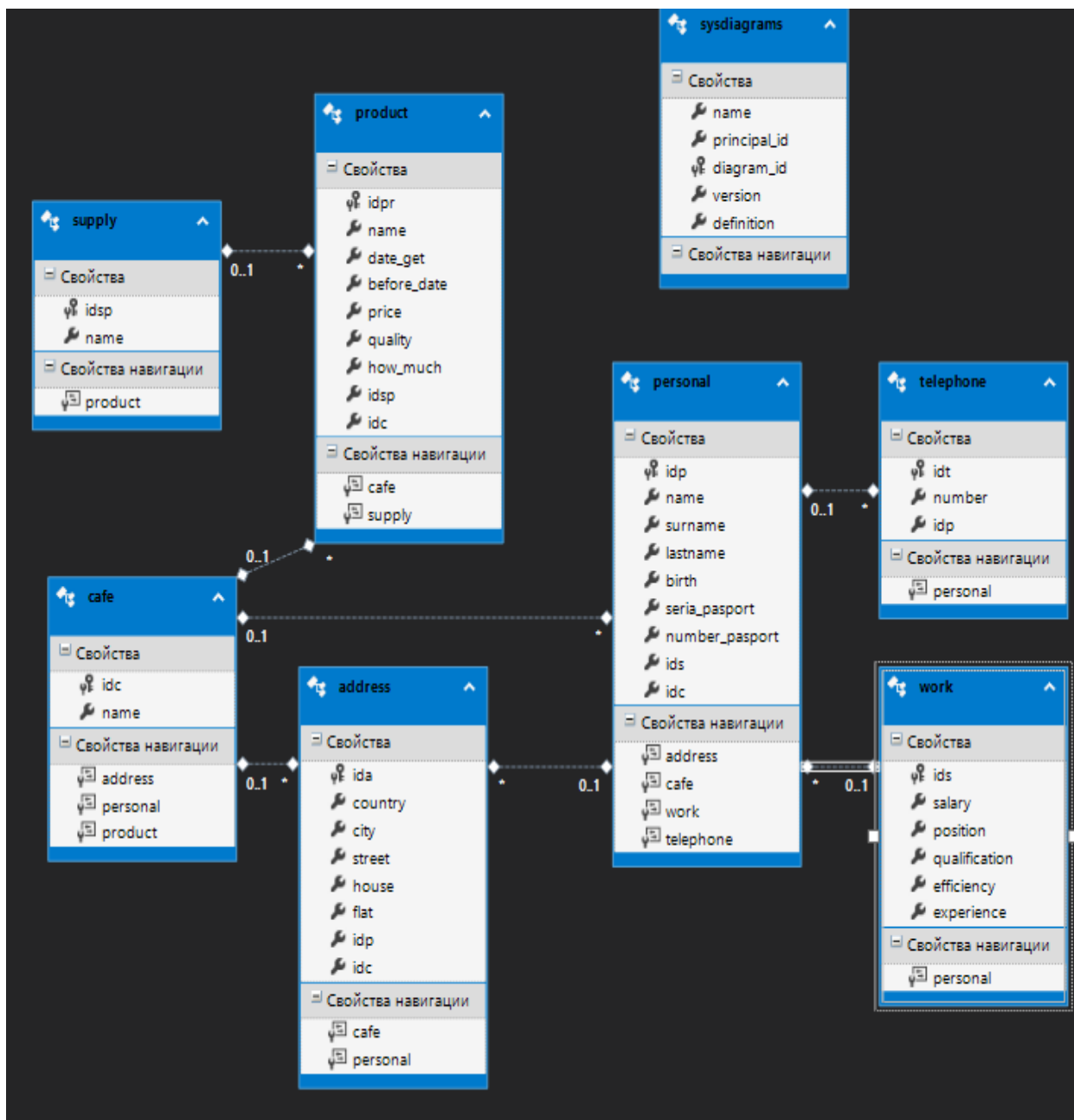


Рисунок 23 – Схема модели ADO.NET подключенной к приложению

В целях оптимизации и упрощения исходного кода приложения добавлено несколько стилей. Приложение типа WPF позволяет добавить необходимое количество стилей вместо того, чтобы прописывать их каждый раз заново, что значительно упрощает работу с кодом и, соответственно, работу с элементами управления.

Для примера один из стилей, который значительно упростил работу с элементом управления «Label», представлен на рисунке 24.

```

<Style TargetType="Label">
  <Setter Property="FontSize" Value="24"></Setter>
  <Setter Property="FontStyle" Value="Italic"></Setter>
  <Setter Property="FontWeight" Value="DemiBold"></Setter>
  <Setter Property="FontFamily" Value="Segoe Script"></Setter>
  <Setter Property="Foreground" Value="■"DarkBlue"></Setter>
  <Setter Property="Background" Value="■"Transparent"></Setter>
  <Setter Property="VerticalContentAlignment" Value="Center"></Setter>
  <Setter Property="HorizontalContentAlignment" Value="Center"></Setter>
</Style>

```

Рисунок 24 – Пример описания стиля для полей отображения текста

Объекту «Label», полю для отображения текстовой информации, присваиваются значения для размера шрифта, его стиля, жирности, цвета шрифта, изменяется начертание шрифта, а также фон самого поля – заливки. Размер шрифта «FontSize» – 24, «FontStyle» со значением «Italic» означает, что текст отображается под наклоном (курсивом), «FontWeight» – шрифт текста полужирный, свойство «FontFamily» – тип шрифта «Segoe Script», свойство «Foreground» – цвет для текста тёмно-синий, а свойство «Background» – заливка поля прозрачная.

Написание таких стилей помогает сэкономить время написания кода для объектов, так как больше не требуется прописывать их каждый раз заново. Достаточно прописать стиль один раз в App.xaml, где хранятся все стили, и он автоматически будет применяться для всех объектов этого типа, например, «Label», что значительно ускоряет и упрощает работу с кодом приложения. Такие стили также были прописаны для кнопок, текстовых полей для ввода данных, полей с выпадающими списками и таблиц.

Добавить	Удалить	Изменить			
зарплата	название	квалификация	эффективность	стаж (в годах)	
26000.0000	waiter	high qualification	high efficiency	5	

Рисунок 25 – Вид страницы с отображением таблицы «Должность»

На рисунке 25 показана страница с таблицей «Должность» из базы данных. Кроме неё на странице также имеются три кнопки, одно текстовое поле и одно поле с выпадающим списком. Добавлены кнопки отвечают за добавление, внесение изменений и удаления данных из базы данных. Также на данной странице реализован активный поиск данных. На рисунке 26 представлена разметка рассмотренной страницы.

```

<StackPanel Orientation="Horizontal">
  <Button x:Name="ButtonAdd" Click="ButtonAdd_click" Margin="0 0 5 0">Добавить</Button>
  <Button x:Name="ButtonDel" Click="ButtonDel_click" Margin="0 0 5 0">Удалить</Button>
  <Button x:Name="ButtonCh" Click="ButtonCh_Click" Margin="0 0 5 0">Изменить</Button>
  <TextBox x:Name="search" TextChanged="search_TextChanged" Margin="0 0 5 0" Width="150" Height="30">
  <ComboBox x:Name="choose" SelectionChanged="choose_SelectionChanged" Width="120" Height="30">
</StackPanel>
<DataGrid CanUserAddRows="False" Grid.Row="1" x:Name="DataGridRegistration" ColumnWidth="*" Auto
  <DataGrid.Columns>
    <DataGridTextColumn Header="зарплата" Binding="{Binding salary}"></DataGridTextColumn>
    <DataGridComboBoxColumn Header="название" x:Name="name" SelectedItemBinding="{Binding Pa
    <DataGridComboBoxColumn Header="квалификация" x:Name="qua" SelectedItemBinding="{Binding
    <DataGridComboBoxColumn Header="эффективность" x:Name="ef" SelectedItemBinding="{Binding
    <DataGridTextColumn Header="стаж (в годах)" Binding="{Binding experience}"></DataGridTex
  </DataGrid.Columns>

```

Рисунок 26 – Пример разметки страницы «Должность»

На странице с таблицей можно просматривать непосредственно сами данные таблицы, удалять строки или вносить изменения. Для удаления необходимо просто выбрать строку и нажать на кнопку удалить. Если перед удалением не выбрать строку, приложение выдаст ошибку в виде диалогового окна с сообщением об ошибке (рисунок 27).

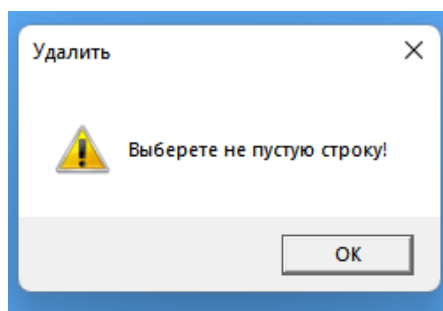


Рисунок 27 - Диалоговое окно с сообщением об ошибке при удалении пустой строки

Если строка была выбрана, то перед удалением выйдет появится диалоговое окно с подтверждением удаления, действительно ли нужно удалять строку, на тот случай, если строка была выбрана неправильно или случайно (рисунок 28).

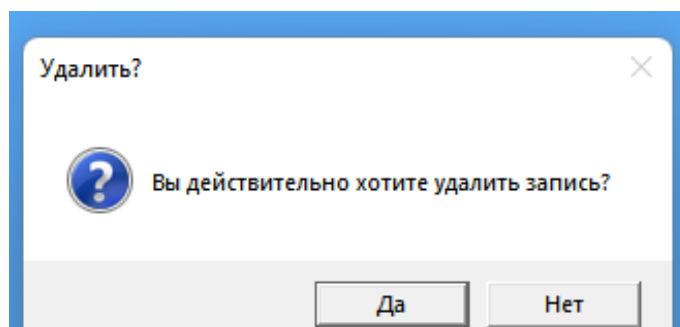


Рисунок 28 – Диалоговое окно с подтверждением удаления

На рисунке 29 представлен обработчик ошибки при выборе пустой строки, с предупреждением об удалении.

```
//проверка на то, выбрали ли строку
var r = DataGridRegistration.SelectedItem as product;
if (r == null)
{
    MessageBox.Show("Выберете не пустую строку!", "Удалить",
    return;
}
```

Рисунок 29 – Пример кода кнопки удаления пустой строки

Если выбранные данные ещё используются, связаны с другими таблицами, в которых используются, то появится диалоговое окно с сообщением об ошибке при удалении данных, которые ещё необходимы. Так на рисунке 29 показан пример попытки удаления данных о должности.

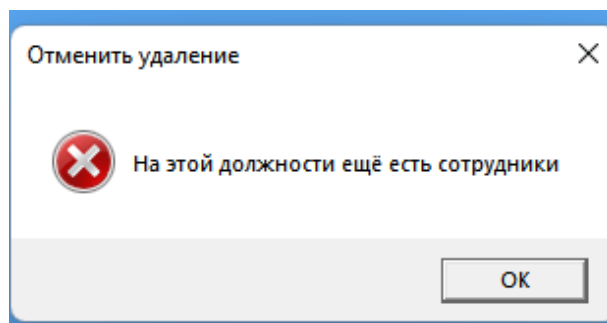


Рисунок 30 – Пример попытки удаления данных

Обработчик ошибки при удалении ещё используемых данных, представленные на рисунке 31.

```
//удаление строки
MessageBoxResult result = MessageBox.Show("Вы действительно хотите удалить",
if (result == MessageBoxResult.Yes)
{
    try
    {
        context.product.Remove(r);
        context.SaveChanges();
        DataGridRegistration.ItemsSource = context.product.ToList();
    }
    catch
    {
        MessageBox.Show("Этот продукт ещё используется в приготовлении",
    }
}
```

Рисунок 31 - Пример кода кнопки удаления

Для изменения данных таблицы необходимо сначала выбрать строку, внести изменения и потом уже нажать на кнопку, чтобы сохранить изменения. Если при нажатии кнопки не были изменены данные таблицы, то появится сообщение об ошибке (рисунок 32).

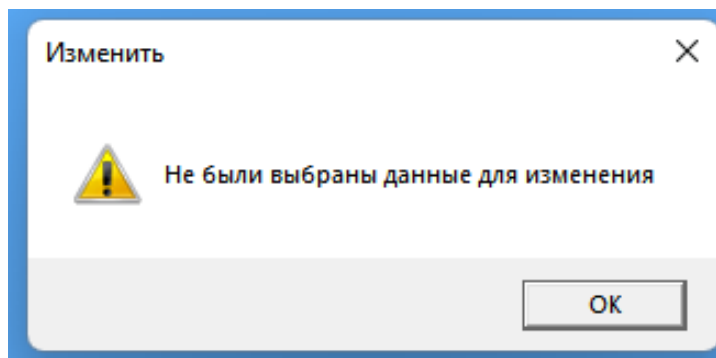


Рисунок 32 - Диалоговое окно с сообщением об ошибке при изменении

При изменении данных в таблице должностей было предусмотрено, чтобы пользователь сам постоянно не вводил одинаковые и повторяющиеся записи, в таблице, кроме обычных столбцов, есть столбцы, в которых можно осуществлять выбор данных из уже представленных или, например, столбцы, в которых происходит привязка данных из другой таблицы, осуществляется выбор из данных привязанного столбца из другой таблицы (рисунок 33).



Рисунок 33 – Пример выбора данных в столбце для изменения

Если данные берутся из другой таблицы, то в списке указываются данные привязанного столбца. Чтобы избежать постоянного рутинного ввода данных в ручную, данные берутся из массива (рисунок 34).

```
InitializeComponent();
context = new santoriniEntities();
//вывод данных таблицы официант в таблицу на странице
DataGridRegistration.ItemsSource = context.work.ToList();
choose.ItemsSource = context.work.ToList();
string[] mas = { "low qualification", "low-middle qualification",
qua.ItemsSource = mas;
string[] mas1 = { "cleaner", "waiter", "cooker", "administrator",
name.ItemsSource = mas1;
string[] mas2 = { "low efficiency", "low-middle efficiency", "midd
ef.ItemsSource = mas2;
```

Рисунок 34 – Пример кода с массивом данных для отображения в таблице

При нажатии на кнопку добавления откроется другая страница для добавления строк в соответствующую таблицу (рисунок 35).

Рисунок 35 – Страница для добавления данных в таблицу

На данной странице расположены следующие управляющие элементы:

- кнопка для сохранения данных в таблицу;
- кнопка для возвращения обратно на страницу с отображением таблицы без внесения изменений;

- поля для отображения текста;
- поле для ввода текстовой информации;
- поля с выбором информации из предложенных вариантов.

Разметка данной страницы в XAML, представлена на рисунке 36.

```

Grid Background="#FF9ECD9F">
  <Grid.ColumnDefinitions>
    <ColumnDefinition/>
    <ColumnDefinition/>
  </Grid.ColumnDefinitions>
  <Grid.RowDefinitions>
    <RowDefinition/>
    <RowDefinition Height="60"/>
  </Grid.RowDefinitions>
  <StackPanel Orientation="Vertical" Grid.Column="0">
    <Label HorizontalAlignment="Center" Width="300" Margin="0 20 0 0">
    <Label HorizontalAlignment="Center" Width="300" Margin="0 35 0 0">
    <Label HorizontalAlignment="Center" Width="300" Margin="0 40 0 35">
    <Label HorizontalAlignment="Center" Width="200" HorizontalContentA
  </StackPanel>
  <StackPanel Orientation="Vertical" Grid.Column="1">
    <ComboBox x:Name="name" Margin="0 30 0 30" HorizontalAlignment="Ce
    <ComboBox x:Name="qua" Margin="0 30 0 30" HorizontalAlignment="Cent
    <ComboBox x:Name="ef" Margin="0 30 0 30" HorizontalAlignment="Cent
    <TextBox x:Name="exp" Text="{Binding experience}" Margin="0 30 0
  </StackPanel>
  <Button Grid.Row="1" Grid.ColumnSpan="2" HorizontalAlignment="Center"
  <Button Grid.Row="1" HorizontalAlignment="Right" Grid.ColumnSpan="2" x
Grid>

```

Рисунок 36 – Пример разметки страницы для добавления

Поля с выбором данных реализованы через массивы. Чтобы пользователю не приходилось каждый раз вводить одни и те же данные, было принято решение добавить вместо обычных текстовых полей для ввода данных поля с выпадающим списком (рисунок 37).

```

context = new santoriniEntities();
string[] mas = { "low qualification", "low-middle qualification", "middle qualifi
qua.ItemsSource = mas;
string[] mas1 = { "cleaner", "waiter", "cooker", "administrator", "deliveryman",
name.ItemsSource = mas1;
string[] mas2 = { "low efficiency", "low-middle efficiency", "middle efficiency",
ef.ItemsSource = mas2;

```

Рисунок 37 – Пример кода массива данных для полей с выпадающим списком на странице с добавлением

Для того, чтобы администратору на предприятии общественного питания ООО «Сандорини» не приходилось заниматься вычислением

заработной платы, она автоматически вычисляется при добавлении строки с должностью. Каждая должность имеет минимальный оклад за работу, надбавки получаются при выборе пунктов в выпадающих списках. Надбавки начисляются за квалификацию сотрудника и его эффективность в работе.

Если при добавлении новой должности совпадут рассчитываемая заработная плата и название должности, то появиться сообщение об ошибке, как представлено на рисунке 38.

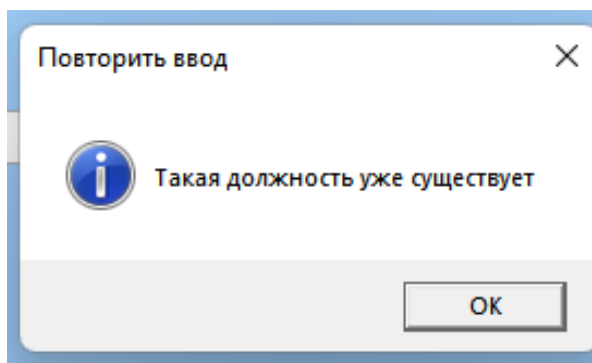


Рисунок 38 - Диалоговое окно с сообщением об ошибке при неверном добавлении

Код данной ошибки представлен на рисунке 39. Если же ошибок не возникнет и все данные были введены корректно, то добавится новая строка и приложение сразу перейдет на страницу с отображением таблицы с помощью метода указанного на рисунке 40.

```
//проверка на существование такой строки
if (context.work.Any(o => o.position == name.Text) &&
{
    ...
    MessageBox.Show("Такая должность уже существует",
}
else
{
```

Рисунок 39 – Пример кода об ошибке

```

context.work.Add(new work //добавление строки в таблицу
{
    position = Convert.ToString(name.SelectedValue),
    qualification=Convert.ToString(qua.SelectedValue),
    efficiency= Convert.ToString(ef.SelectedValue),
    experience=exp.Text,
    salary=sa,
});
context.SaveChanges();//сохранение новой строки
NavigationService.Navigate(new Work());

```

Рисунок 40 – Пример кода с методом сохранения и перехода на страницу с отображением таблицы

2.5 Руководство пользователя и администратора

Руководство пользователя

Данное приложение предназначено для использования в качестве информационно-поисковой системы, используемой администраторами предприятия общественного питания ООО «Сандорини».

Приложение состоит из нескольких логически самостоятельных частей взаимосвязанных между собой частей, каждая из которых разработана в соответствии с требованиями и условиями, удовлетворяющими запросам каждого из видов работ.

Составлена база данных, в которой будут храниться все данные, связанные с работой приложения.

После запуска программы перед пользователем появляется форма авторизации, на которой необходимо ввести данные для входа в приложение, что позволит начать работу с программой. Необходимо ввести фамилию администратора и пароль, который выдается администратору главным бухгалтером. Он предназначен для защиты базы данных от посторонних пользователей.

Следующим этапом является либо добавление/изменение текущих данных в базе данных, либо просмотр таблиц с данными.

На форме просмотра таблиц расположены кнопки для выбора необходимой таблицы.

На форме добавления строк в таблицу расположены компоненты для ввода, выбора данных. Также на форме расположено несколько кнопок для работы с программой.

Кнопка «Выход» осуществляет выход из программы.

Кнопка «О программе» позволяет осуществить запуск текстового документа, который содержит в себе информацию о предназначении данной программы.

Руководство администратора

Данное приложение предназначено для управления персоналом и закупками и должно реализовывать такие функции как:

- вход в программу должен осуществляться на фамилии администратора;
- вход в программу должен осуществляться по паролю, который является общим и будет выдаваться главным бухгалтером;
- отображение данных в виде таблиц: «Сотрудники», «Должность», «Заведение», «Продукты», «Телефон», «Адрес», «Поставщик»;
- добавление данных в основные таблицы: «Сотрудники», «Должность», «Заведение», «Продукты». И в другие таблицы: «Телефон», «Адрес», «Поставщик»;
- удаление данных из основных таблиц: «Сотрудники», «Должность», «Заведение», «Продукты». И в других таблиц: «Телефон», «Адрес», «Поставщик»;
- изменение данных в основных таблицах: «Сотрудники», «Должность», «Заведение», «Продукты». И в других таблицах: «Телефон», «Адрес», «Поставщик».

Приложение предназначено для сотрудников предприятия, в должностные обязанности которых входят функции по администрированию программным модулем информационной подсистемы на предприятии общественного питания ООО «Сандорини».

ЗАКЛЮЧЕНИЕ

На предприятии общественного питания ООО «Сандорини» администратор вынужден ежедневно выполнять повторяющуюся, рутинную работу с бумажными документами. Этот подход к ведению документооборота является устаревшим и неудобным, снижает эффективность его трудовой деятельности.

Для решения данной проблемы, вызванной большим ежедневным документооборотом была поставлена задача на проектирование и разработку автоматизированной системы для управления персоналом и закупками на предприятии общественного питания ООО «Сандорини».

В первой главе выпускной квалификационной работы, название которой «Обзор и анализ предметной области» подробно описан анализ деятельности предприятия, рассмотрены примеры существующих программных решений, проанализированы их достоинства и недостатки, описаны функциональные и аппаратные требования к будущему программному продукту.

Во второй главе выпускной квалификационной работы, название которой «Проектирование модуля информационной подсистемы управление персоналом и закупками для ООО «Сандорини»» описан выбор программной среды для создания информационной подсистемы и система управления базами данных, подробно описаны процессы проектирования и разработки информационной системы и базы данных.

Все поставленные задачи и цели были полностью реализованы в автоматизированной системе для управления персоналом и закупками на предприятии общественного питания ООО «Сандорини».

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Албахари, Джозеф, Албахари, Бен. А45 С# 7.0. Справочник. Полное описание языка.: Пер. с англ. — СПб.: ООО “Альфакнига”, 2018. — 1024 с. : ил. — Парал. тит. англ.
2. Евдокимов П. В., С# на примерах. 4-е издание (переработанное и обновленное) – СПб.: Наука и Техника, 2019. — 320 с., ил.
3. Краткое руководство. Подключение к экземпляру SQL Server и выполнение запросов с помощью SQL Server Management Studio (SSMS) - <https://docs.microsoft.com/ru-ru/sql/ssms/quickstarts/ssms-connect-query-sql-server?view=sql-server-2017>
4. Учебник. Создание первого приложения универсальной платформы Windows в Visual Studio с помощью XAML и С# - <https://docs.microsoft.com/ru-ru/visualstudio/get-started/csharp/tutorial-uwp?view=vs-2022>
5. Хайруллин Р.С., Программирование на С#: учебное пособие. – Хайруллин Р.С. – Казань: Изд-во Казан.гос.архитект.-строит.ун-та, 2017 – 159 с.
6. <http://analyst.by/articles/rukovodstvo-polzovatelya-sovetyi-dlya-sostavleniya>
7. http://mycsharp.ru/post/37/2014_02_04_peregruzka_metodov_v_si-sharp.html
8. <http://techwrconsult.com/rukovodstvo-polzovatelya>
9. http://tehprog.ru/index.php_page%3Dlecture35.html
10. <https://blog.iteam.ru/vidy-organizatsionnoj-struktury-predpriyatiya/>
11. <https://c-sharp.pro/?p=59>
12. <https://devpractice.ru/c-sharp-lesson-6-if-and-for/#p11>
13. <https://dic.academic.ru/dic.nsf/ruwiki/45728>
14. <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/keywords/try-catch>

15. <https://gospodaretsva.com/komponent-tradiogroup-prilozhenie-s-neskolkimi-formami.html>
16. <https://habr.com/ru/company/mailru/blog/325492/>
17. <https://habr.com/ru/company/ruvds/blog/461661/>
18. <https://habr.com/ru/company/trinion/blog/245615/>
19. <https://infopedia.su/24xacf.html>
20. <https://intuit.ru/studies/courses/629/485/lecture/11015>
21. https://knowledge.allbest.ru/programming/3c0b65625a3bc79a5d53a88521306d37_0.html
22. https://professorweb.ru/my/csharp/charp_theory/level6/6_1.php
23. <https://sprintinvest.ru/organizacionnaya-struktura-predpriyatiya-ponyatie-i-tipy>
24. https://studbooks.net/1884022/informatika/otladka_programmnyh_moduly_ispolzovaniem_spetsializirovannyh_programmnyh_sredstv
25. https://studbooks.net/2067027/informatika/vybor_strategii_testirovaniya_razrabotka_testov
26. https://studwood.ru/1880553/informatika/vybor_strategii_testirovaniya_razrabotka_testov
27. https://www.prj-exp.ru/dwh/dwh_model_types.php
28. <https://www.software-testing.ru/library/5-testing/207-2008-10-06-10-05->