

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
(НИУ «БелГУ»)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ  
ТЕХНОЛОГИЙ

**РАЗРАБОТКА АВТОМАТИЗИРОВАННОЙ ПОДСИСТЕМЫ УЧЕТА  
ЗАКАЗОВ НА ПРИМЕРЕ КАФЕ «ТАЙМ»**

Выпускная квалификационная работа  
обучающегося по направлению подготовки 09.03.03 «Прикладная  
информатика»  
очной формы обучения, группы 07001304  
Ивановой Марии Николаевны

Научный руководитель  
ст. пр. Болгова Е.В.

БЕЛГОРОД 2017

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ .....	3
1 Аналитическая часть .....	5
1.1 Техничко-экономическая характеристика предприятия .....	5
1.2 Постановка задачи .....	8
1.2.1 Цель и назначение автоматизированной подсистемы учета заказов .....	8
1.2.2 Общая характеристика процесса создания подсистемы.....	9
1.2.3 Обоснование необходимости и цели создания автоматизированной подсистемы .....	10
1.2.4 Анализ существующих разработок.....	13
2 Проектная часть .....	16
2.1 Обоснование проектных решений .....	16
2.1.1 Техническое обеспечение .....	16
2.1.2 Информационное обеспечение.....	17
2.1.3 Программное обеспечение.....	18
2.1.4 Технологическое обеспечение.....	19
2.1.5 Выбор программных средств .....	20
2.2 Информационное обеспечение задачи .....	21
3 Программная реализация проектных решений .....	25
3.1 Разработка программного обеспечения задачи .....	25
3.1.1 Разработка базы данных.....	25
3.1.2 Разработка пользовательского интерфейса.....	30
3.2 Контрольный пример реализации проекта .....	37
3.3 Расчет экономической эффективности разработанной подсистемы.....	50
ЗАКЛЮЧЕНИЕ .....	55
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....	57
ПРИЛОЖЕНИЯ.....	61

## ВВЕДЕНИЕ

Предприятия общественного питания в наше время набирают большую популярность, всё больше людей предпочитают обедать, ужинать, просто перекусывать или отмечать различные мероприятия в каком-либо заведении по типу кафе, ресторана, пиццерии и так далее. Другие предпочитают сделать заказ на дом или в какое-то иное место [21].

Кафе «Тайм» является одним из главных заведений общественного питания в пгт. Волоконовка. Для его дальнейшего развития уместно создание приложения, которое сможет поспособствовать этому. Тема выпускной квалификационной работы является актуальной, так как на рассматриваемом предприятии нет автоматизированной системы, которая помогла бы сотрудникам наиболее удобно регистрировать заказы и вести их учет.

Объектом исследования выпускной квалификационной работы является деятельность кафе «Тайм».

Предметом исследования является алгоритм информационного представления процесса учета заказов в кафе «Тайм».

Цель выпускной квалификационной работы – повышение эффективности деятельности персонала занимающегося обслуживанием клиентов в кафе «Тайм».

Для достижения поставленной цели необходимо решить следующие задачи:

- спроектировать функциональную модель предприятия «Как есть» и «Как должно быть»;
- разработать логическую и физическую модель базы данных;
- разработать автоматизированную подсистему учета заказов в кафе «Тайм»;
- протестировать разработанную автоматизированную подсистему учета заказов в кафе «Тайм»;

– произвести оценку эффективности автоматизированной подсистемы учета заказов.

ВКР состоит из введения, трех разделов, заключения, списка использованных источников и приложений.

Во введении приведена актуальность темы, поставлена цель и необходимые для достижения поставленной цели задачи, а также приведен объект и предмет исследования.

В первом разделе рассмотрена предметная область, проанализирована деятельность кафе, разработана функциональная модель предприятия «Как есть» и «Как должно быть», определены решаемые задачи выпускной квалификационной работы и проведен анализ существующих систем автоматизации предприятий общественного питания.

Второй раздел ВКР включает в себя обоснование выбора проектных решений по техническому, информационному, программному и технологическому обеспечению. В данном разделе приведены схемы физического и логического уровня базы данных.

В третьем разделе описана программная реализация проектных решений. В этом разделе рассматривается разработка подсистемы, ее тестирование и описание экономической части выпускной квалификационной работы, в которой была рассчитана экономическая эффективность разработанной подсистемы.

В заключении приводятся выводы по данной выпускной квалификационной работе и перспективы разработанной подсистемы.

В приложениях представлен программный код, скриншоты работы подсистемы, функциональные модели деятельности предприятия, формулы расчета экономической эффективности, анализ существующих разработок и техническое задание.

Выпускная квалификационная работа написана на 58 страниц, содержит в себе 44 рисунка, 7 таблиц, 33 источника литературы и 6 приложений.

# 1 Аналитическая часть

## 1.1 Техничко-экономическая характеристика предприятия

Полное наименование организации: ИП Мартынов И.А. кафе «Тайм». Мартынов Игорь Александрович является индивидуальным предпринимателем, в соответствии с законом РФ «О государственной регистрации юридических лиц и индивидуальных предпринимателей» №129-ФЗ, Гражданским кодексом РФ и другими федеральными законами РФ, он в полной мере может заниматься предпринимательской деятельностью без специального юридического образования.

Кафе «Тайм» одно из главных заведений поселка Волоконовка предоставляющее услуги по организации питания, предоставляя своим клиентам широкий ассортимент блюд сложного приготовления, включая заказные и фирменные. Осуществляется бесплатная доставка еды на дом. Предварительный заказ доставки не предоставляется. Заказ банкета производится за несколько дней до необходимой даты (в пределах 7 дней). Руководителем предприятия является Мартынов Дмитрий Игоревич, он вправе выполнять все возложенные на него обязательства в соответствии с доверенностью 5/5354Д от 27.12.2016.

Основными видами деятельности кафе «Тайм» являются:

- торговая деятельность;
- производственная деятельность.

К основным услугам кафе «Тайм» относится:

- приготовление и выполнение заказа блюд и напитков в кафе;
- приготовление и выполнение заказа блюд, напитков и пиццы на дом, доставка осуществляется в пгт. Волоконовка и в пос. Пятницкое;
- бронирование столиков и залов.

Основной целью деятельности кафе является получение и увеличение прибыли. В таблице 1.1 представлены технико-экономические показатели кафе «Тайм», рассматриваемый период – один год.

Таблица 1.1 – Технико-экономические показатели кафе «Тайм»

№	Показатели	Единицы измерения	Декабрь 2015 г.	Декабрь 2016 г.
1)	Выручка от реализации	тыс. руб.	843670	874200
2)	Численность персонала	чел.	13	13
3)	Себестоимость реализации продукции	тыс. руб.	310459	325678
4)	Прибыль от реализации	тыс. руб.	176903	194870
5)	Рентабельность деятельности	%	21,40	23,12

На рисунке 1.1 представлена организационная структура предприятия.



Рисунок 1.1 – Организационная структура предприятия

Главную должность кафе «Тайм» занимает директор, он контролирует соблюдение работниками правил и норм по технике безопасности и по охране труда, принимает решения о назначении и освобождении сотрудников от занимаемых должностей, может наложить на сотрудника как штраф, так и назначить премию. Ведет переговоры от лица кафе, заключает договора. Осуществляет контроль, за правильным использованием финансовых, трудовых и материальных ресурсов, изучает спрос потребителей на продукцию кафе. Руководители среднего звена подчиняются непосредственно ему [10].

Бухгалтер ведет всю работу с документацией и ответственна за решение финансовых вопросов, в том числе за начисление заработной платы, больничных, определения графиков отпусков и ведение табеля учета, производит начисление и перечисление налогов. Составляет калькуляцию себестоимости продукции [23].

Администратор зала поддерживает комфорт и приятную атмосферу для гостей, управляет персоналом зала (составление графиков работы, контроль рабочего процесса), работает с документацией, занимается организацией и проведением банкетов, выполнением служебных поручений руководителя, также его обязанностями являются своевременная замена или ремонт испорченной мебели и оборудования. В подчинение администратора входят: официанты, бармен и уборщик [25].

Официанты подают меню, принимают заказы от посетителей, помогают в выборе блюд и напитков, сервируют столы, подают заказанные блюда, выводят окончательную сумму заказов и подают ее посетителю [22].

Бармен осуществляет изготовление и подачу напитков, а также исполняет роль кассира, то есть работает с кассой, выписывает чеки и предоставляет сдачу, принимает заявки на бронирование столиков, а также принимает заказы блюд на дом [9].

Уборщик поддерживает чистоту в зале в течение дня и убирает кафе после его закрытия (кухню, залы и другое) [15].

Следующим звеном кафе является шеф повар, который обучает поваров, разрабатывает новые блюда и меню в целом, контролирует качество блюд и поставляемых продуктов, осуществляет заказ необходимых продуктов, контролирует дисциплину кухни, соблюдение гигиены и контроль себестоимости блюд, также осуществляет приготовление и оформление блюд. В его подчинении: повар и мойщики посуды [29].

В обязанности повара входят приготовление блюд и его оформление [29].

В обязанности мойщика посуды входит своевременное мытье посуды и подготовка к ее использованию [21].

## **1.2 Постановка задачи**

### **1.2.1 Цель и назначение автоматизированной подсистемы учета заказов**

Для достижения поставленной цели выпускной квалификационной работы, необходимо разработать подсистему ведения учета заказов в кафе «Тайм». Данная автоматизированная подсистема должна включать в себя информацию о сотрудниках, занятости столиков, зарегистрированных заказах, обслуживаемых клиентах.

Задачи разрабатываемой автоматизированной подсистемы:

- сбор требований к подсистеме и определение решаемых подсистемой задач;
- обоснование необходимости создания подсистемы;
- проектирование базы данных;
- организация хранения, ввода и обработки данных в подсистеме;
- формирование отчетов по заданным параметрам;



- проведение анализа по некоторым наборам данным;
- разработка мер по защите от несанкционированного доступа к подсистеме;
- разработка автоматизированной подсистемы учета заказов в кафе.

### **1.2.2 Общая характеристика процесса создания подсистемы**

Входными данными в автоматизированной подсистеме будут сведения о сотрудниках, клиентах, столиках, заказах блюд и напитках, сведения о меню. Выходными: информация о занятости столиков и различные виды отчетов.

Этапы создания автоматизированной подсистемы учета заказов в кафе:

- выявить требования к подсистеме;
- выбрать используемые программные продукты;
- установить и настроить программные продукты для работы;
- спроектировать подсистему;
- разработать подсистему;
- заполнить подсистему необходимой информацией.

Автоматизированная подсистема позволит сотруднику скачать или распечатать отчет по заказам на определенную дату, а также просматривать проанализированные данные по регистрации заказов сотрудниками в виде диаграмм.

### 1.2.3 Обоснование необходимости и цели создания автоматизированной подсистемы

Проведя анализ предметной области, была разработана структурно-функциональная диаграмма организации бизнеса «Как есть». Такая диаграмма позволяет понять, как функционирует рассматриваемое предприятие [13].

На рисунке 1.2 контекстная диаграмма, которая представляет общее описание системы в целом [12]. На вход поступают: данные клиента, данные о сотруднике, предпочтения клиента, данные о номенклатуре. На выходе: чек, бронь, сформированный заказ, отказ от заказа, отказ от столика, напечатанный отчет.

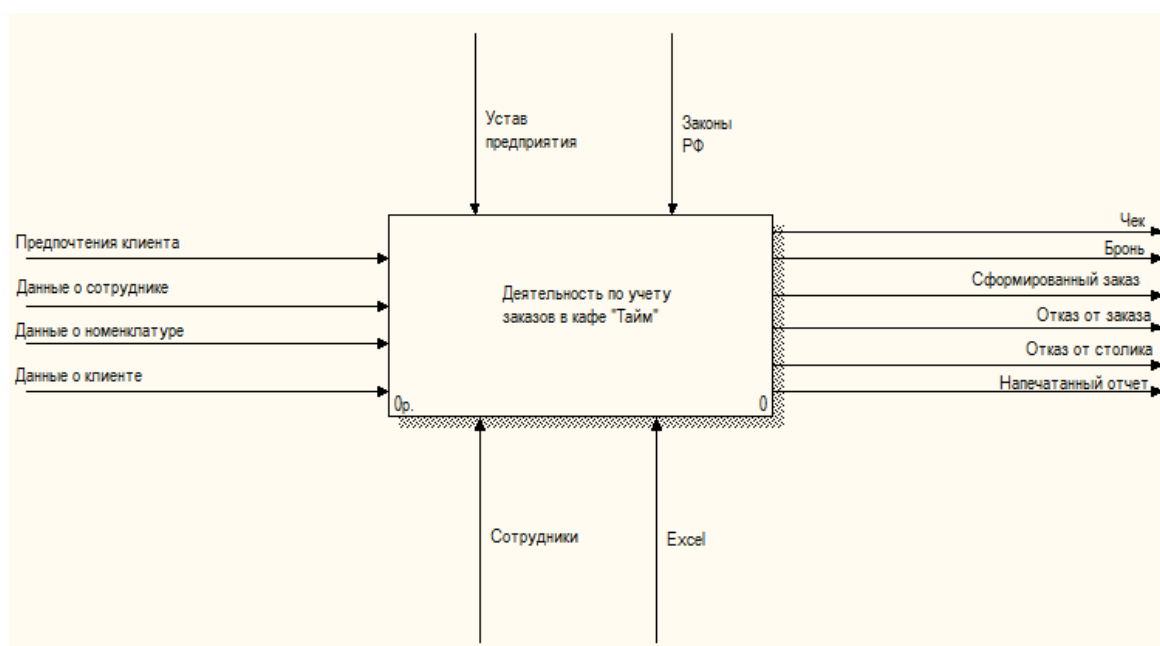


Рисунок 1.2 – Контекстная диаграмма

На рисунке 1.3 представлена декомпозиция контекстной диаграммы, то есть разбиение системы на подсистемы, для более наглядного описания работы предприятия. Данная декомпозиция имеет четыре работы: первоначальная получение и запись данных, бронирование, заказ блюд и напитков, составление отчетов. Работы имеют входные и выходные данные, а

также информацию управляющую действиями работы (стрелки управления) и ресурсы, выполняющие работу (стрелки механизма).

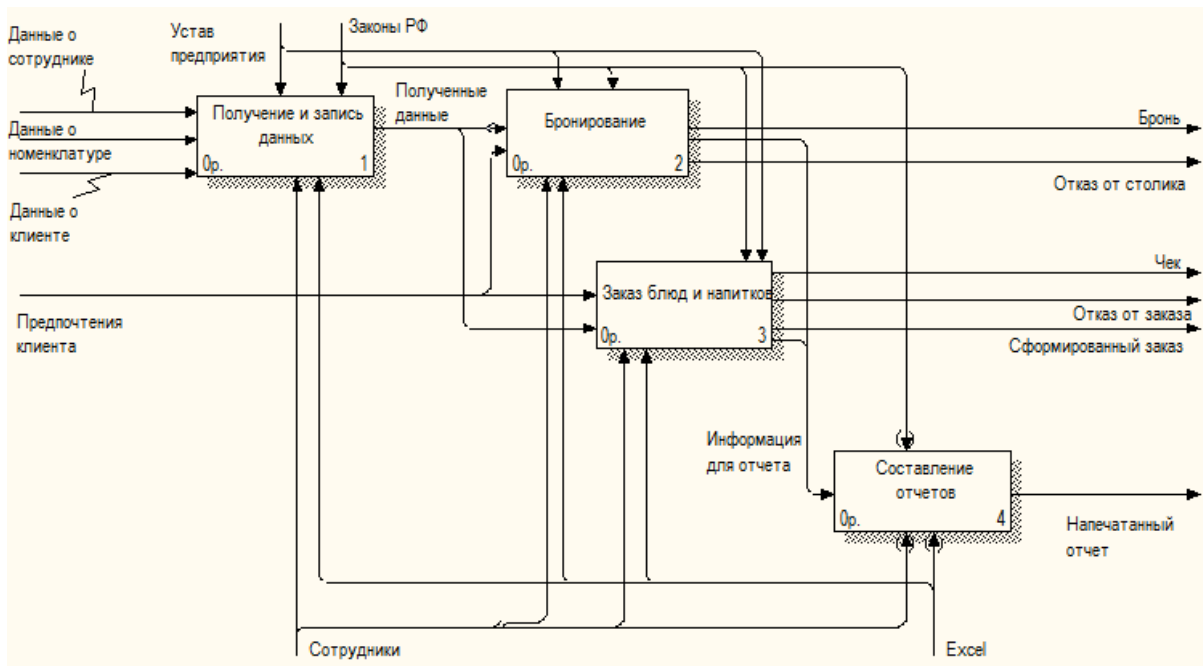


Рисунок 1.3 – Декомпозиция контекстной диаграммы

Декомпозиции отдельных блоков детально приведены в приложении А.

Главным недостатком кафе «Тайм», является работа сотрудников без автоматизированной системы или подсистемы. При использовании АИП работа предприятия будет наиболее эффективной, за счет создания базы данных, которая будет хранить в себе всю необходимую информацию и отчеты, а так же за счет сокращения времени работы сотрудников, так как при использовании подсистемы некоторые операции могут выполняться автоматически, например: создание отчетов.

Структурно-функциональная диаграмма организации бизнеса «Как должно быть» отражает состояние предприятия, устранив выявленные недостатки [1].

На рисунке 1.4 представлена декомпозиция контекстной диаграммы с внесенными изменениями.

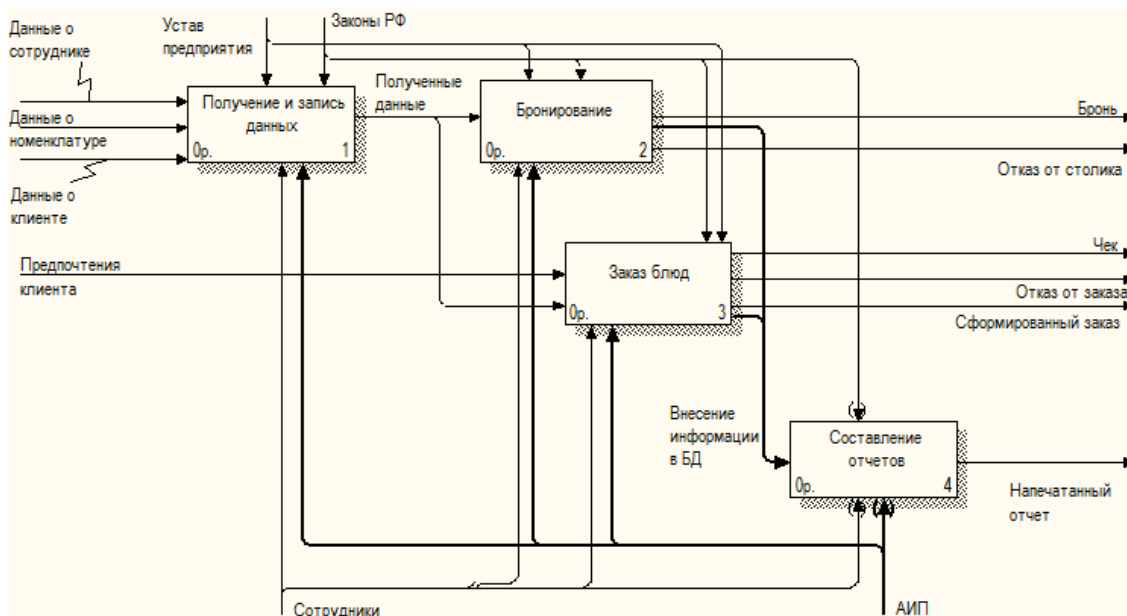


Рисунок 1.4 – Декомпозиция контекстной диаграммы

На рисунке 1.5 представлена декомпозиция контекстной диаграммы «как должно быть» «Составление отчетов». На данной диаграмме была добавлена база данных организации. Данная диаграмма позволяет посмотреть процесс обмена информацией между базой данных и АИП.

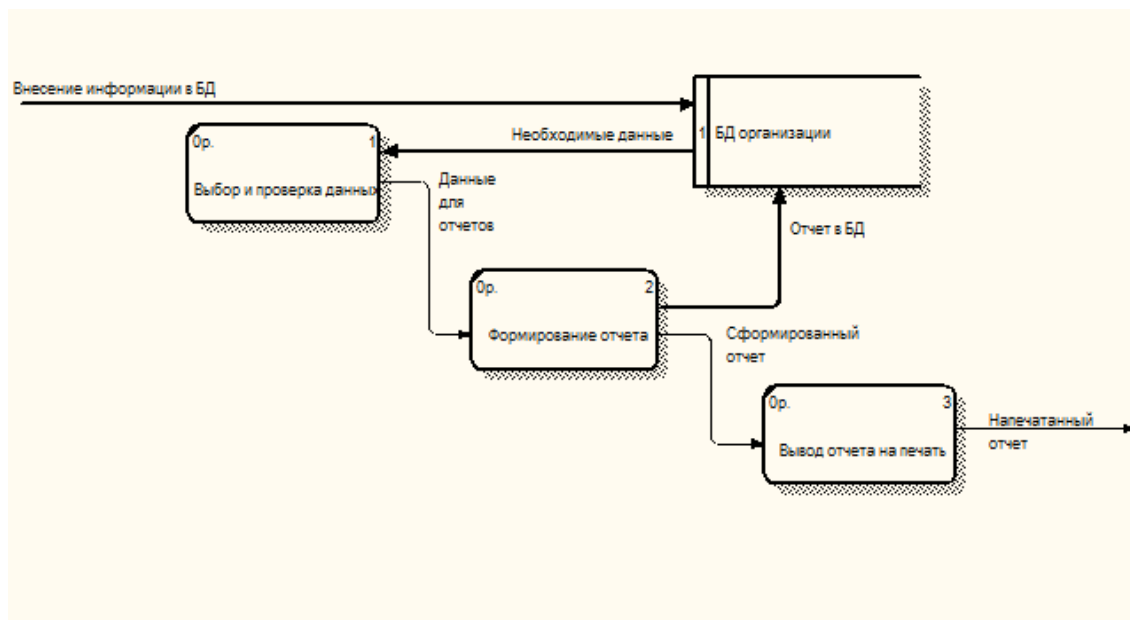


Рисунок 1.5 – Декомпозиция контекстной диаграммы «Составление отчетов»

## 1.2.4 Анализ существующих разработок

На сегодняшний день на рынке систем автоматизации предприятий общественного питания присутствует большое количество различных систем: 1С:Ресторан, R-keeper, ПКО, Парус, Трактирь и другие. Систем настолько много, что без проведения их анализа невозможно выбрать подходящую для ведения автоматизации бизнес-процессов. Для их успешного функционирования необходимо подобрать наиболее эффективную информационную систему, способную хранить и работать с большими объемами информации. Также система должна иметь если не большое количество, то обязательно основные функциональные возможности для автоматизации предприятия. цена такой системы не должна быть высокой [5].

Рассмотрев десять наиболее популярных систем: R-Keeper, ПКО, 1С-Ресторан, Трактир, Tillypad XL, Парус-Ресторан, Quick Resto, Poster, SmartTouch и Frontol (приложение Б). Были выделены основные признаки отбора, которые характеризуют систему в целом.

Позитивные признаки:

- возможность графического представление плана зала;
- наличие системы бронирования столиков;
- возможность распечатки заказа;
- наличие программы лояльности;
- наличие отчетов;
- наличие удобного интерфейса;
- возможность интеграции с системами;
- наличие разграничения прав доступа.

Негативные признаки:

- стоимость системы;
- ущерб.

Для всех признаков установлена градационная шкала, которая отражает степень их влияния на общую эффективность использования системы. При установке градационных шкал использовался метод дискретного разбиения. Суть данного метода заключается в присваивании каждому уровню разбиения числового номера от 1, в случае полного отсутствия влияния рассматриваемого признака на результат выбора, до некоторого значения  $n$ , в случае его наибольшего влияния [37]. Подробная установка градационных шкал, а также нахождение количественного значения каждого уровня шкалы представлена в приложении Б.

Для наглядного выбора наилучшей системы был сформирован лист экспертной оценки (таблица 1.2).

Таблица 1.2 – Лист экспертной оценки

Признак	Возможность графического представление плана зала			Оценка, %
1	2	3	4	5
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	
Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	
Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	
Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	
Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	
Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	
Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	

Продолжение таблицы 1.2

1	2	3	4	5
Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	
Общая суммарная оценка позитивных свойств $\sum P^+_i(s)$ в %				
Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	
Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	
Общая суммарная оценка негативных свойств $\sum P^-_j(s)$ в %				
Общая суммарная оценка эффективности использования в %				

С помощью разработанной таблицы была произведена оценка каждой рассматриваемой системы в отдельности (приложение Б).

Наиболее предпочтительна по итоговым данным, является система 1С-Ресторан, она имеет приемлемую цену, а большинство её признаков и возможностей имеют наивысшую оценку, которые были получены при проведении анализа. Следом за ней по собранным баллам идут системы R-keeper и ПКО, это самые дорогие системы из рассматриваемых программных продуктов, но функциональные возможности полноценнее остальных. Система Frontol является наихудшей, она имеет самую низкую цену из рассматриваемых систем, но количество возможностей у неё значительно меньше других рассматриваемых программных продуктов.

В первом разделе ВКР была проанализирована технико-экономическая характеристика предприятия, выделены конкретные задачи и цели создания автоматизированной подсистемы учета заказов. Также была проанализирована предметная область, на основе этого построена структурно-функциональные диаграммы организации бизнеса «Как есть» и «Как должно быть», выявлены недостатки работы предприятия. Проведен анализ имеющихся систем автоматизации предприятий общественного питания и выявлены наилучшие и наихудшие из рассматриваемых систем.

## 2 Проектная часть

### 2.1 Обоснование проектных решений

#### 2.1.1 Техническое обеспечение

К техническому обеспечению относят комплекс технических средств, которые предназначены для работы информационной системы или подсистемы, а также технологические процессы и соответствующая техническая документация [7].

Для нормального функционирования подсистемы подойдут следующие технические характеристики имеющегося на предприятии ноутбука MSI CX 61:

- процессор: Intel Core i5-4200M 2.5 GHz;
- материнская плата: Intel HM86;
- видео карта: NVIDIA GeForce GT 730M;
- оперативная память (Мб): 4096;
- дисплей: 15.6, HD 1366x768.

Технические средства и документация:

- модем: D-link dsl-2640u;
- мышь: Genius DX-165;
- колонки: Dialog w-203;
- принтер: HP Deskjet F300;
- бумага для печати, инструкция по эксплуатации, коврик для мыши;
- картриджи;
- USB Флеш-накопитель: SmartBay Crown 16 GB.



## 2.1.2 Информационное обеспечение

Информационное обеспечение – это совокупность информационных массивов данных, единая система кодирования и классификации информации, различные схемы информационных потоков, вращающиеся в организации, а также методология построения баз данных [33].

Информационное обеспечение подразделяется [7]:

- внутримашинное информационное обеспечение – это информационная база, которая представлена на машинном носителе, реализованная в режиме реального времени (база данных);
- немашинное информационное обеспечения – это документация на бумажных носителях (приказы, распоряжения, договора и другое).

Существуют также такие понятия как входная, выходная и управленческая информация. Входная – это информация, которая используется в работе, в дальнейшем может быть преобразована, для получения результата. Выходная – это проработанная или преобразованная информация, которая является результатом работы. Управленческая – это информация, которая несет в себе полезность для принятия человеком обоснованного решения [18].

Входными данными разрабатываемой автоматизированной подсистемы учета заказов являются сведения о наименовании, цене блюда и напитков, личные данные о сотрудниках кафе, сведения о клиентах, которые будут взаимодействовать с сотрудником кафе при заказе блюда на дом или при бронировании столиков.

Управленческой информацией необходимой при работе с подсистемой является: Устав предприятия и Законы РФ (Закон РФ от 07.02.1992 г. № 2300-1 «О защите прав потребителей», Правила оказания услуг общественного питания от 15.08.1997 г. № 1036 и другие).

Выходными данными разрабатываемой автоматизированной подсистемы учета заказов являются сведения о занятости столиков в кафе, зарегистрированный заказ и созданная различного вида отчетность.

### **2.1.3 Программное обеспечение**

Программное обеспечение – это совокупность программ, которые обеспечивают функционирование вычислительной системы, а также программ, направленных для решения конкретных задач пользователя [34].

Разрабатываемую автоматизированную подсистему планируется использовать на одной рабочей станции, имеющую операционную систему Windows 2007 Максимальная – это пользовательская операционная система, которая была выпущена после системы Windows Vista, но является предшественником системы Windows 8. По состоянию на 2017 год Windows 7 является самой популярной операционной системой в мире [41].

Windows 7 Максимальная предоставляет пользователю большой объем возможностей, в эту версию включен программный продукт, позволяющий менять язык системы. Таким образом, купив российскую версию, имеется возможность поменять интерфейс системы на английский язык. Кроме того, есть возможность осуществления федеративного поиска - это поиск данных, хранящихся за пределами ПК, к ним относятся данные хранящиеся на локальном компьютере или в корпоративных сетях [35].

Для реализации подсистемы в качестве среды программирования была выбрана кроссплатформенная система управления базами данных Firebird. Это является более удобным средством для программистов, чем использование алгоритмических языков, так как СУБД уже имеют специальные средства для работы с записями и файлами базы данных [31].

СУБД Firebird имеет единственный файл для хранения всех данных, который хранится в формате FDB . Имеется возможность резервного копирования всей базы, при помощи графических и консольных утилит, а так же восстановление всей базы данных. При создании резервной копии необходимо выбрать один важный параметр, который определяет формат резервной копии: платформозависимый или платформонезависимый. Это позволит администратору базы данных при необходимости перенести файл базы данных из одной операционной системы на другую [11].

#### **2.1.4 Технологическое обеспечение**

Технологическое обеспечение – это совокупность методов и средств сбора, обработки, защиты и хранения информации на базе вычислительной техники. Технологический процесс, в свою очередь – это последовательность технологических операций, направленных на выполнение определенного вида работ. Комплекс действий, выполняемых над информацией и её носителем – это технологическая операция [4].

В настоящий момент технологический процесс в кафе включает в себя: сбор данных, внесение их в программный продукт Microsoft Excel , обработку информации и выдачу информации пользователю. Следует отметить, что пользователи каждый раз вносят информацию вручную, при накоплении большого объема информации, нужную информацию в нужный момент времени сложно найти, некоторая информация просто теряется, нет возможности создавать аналитические и другие отчеты, а также не разработана системы скидок для клиентов.

Технологический процесс разрабатываемой подсистемы учета заказов в кафе «Тайм» состоит в следующем:

- сбор данных в процессе разговора с клиентом;

- внесение информации в БД пользователем, а также выбор необходимой информации из базы данных с помощью подсистемы;
- обработка информации пользователем, а также автоматически подсистемой;
- при выполнении заказа проверка системой на наличии скидки и расчет итоговой суммы заказа;
- выдача результирующей информации пользователю. Это может быть зарегистрированный заказ, необходимый отчет, выдача информации по запросу.

### **2.1.5 Выбор программных средств**

В данной выпускной квалификационной работе использовались следующие средства проектирования и разработки автоматизированной подсистемы:

1) AllFusion Process Modeler – это инструмент для описания предметной области, путем моделирования, документирования и оптимизации бизнес-процессов предприятия. Позволяет сформировать целостную картину деятельности предприятия, что позволит выявить недочеты работы предприятия и спроектировать их устранение. AllFusion Process Modeler доступный в приобретении и несложный в процессе работы [12].

2) AllFusion Erwin Data Modeler – специальное case-средство для проектирования БД, это бесплатный и простой в использовании инструмент. AllFusion Erwin Data Modeler позволяет создавать, сопровождать и документировать витрины данных и базы данных [11].

3) К используемым программным средствам относятся кроссплатформенная система управления базами данных Firebird, она работает на таких операционных системах как Microsoft, Unix, Linux.

Используется в различных промышленных системах, является независимым проектом С и С++ программистов. Эта бесплатная СУБД проста в установке и использовании, позволяет создавать сложные хранимые процедуры, использует язык SQL, позволяет создавать триггеры и генераторы, имеет большое количество различных функций [40].

4) IBExpert – это GUI-оболочка, которая предназначена для создания и администрирования баз данных Firebird и InterBase, также позволяет выбирать и изменять данные в базе. К достоинствам относится: возможность одновременной работы с несколькими БД, поиск метаданных, мощный SQL-редактор с возможностью фонового выполнения запросов и хранения их истории, автозавершение кода на языке SQL, отладчик процедур и триггеров, отдельный редактор для всех объектов БД с возможностью синтаксической подсветки и различные другие достоинства [16].

5) Для разработки программного обеспечения используется С++ Builder. Этот программный продукт позволяет объединить в себе комплекс библиотек, отладчик, компилятор, редактор кода и другие возможные компоненты. Цикл разработки схож с Delphi, большинство компонентов разработанных на этом языке, можно использовать и в С++ Builder без изменения, обратно это не работает [28].

6) Использовались также следующие программы Microsoft Office: Microsoft Word 2010, Microsoft Visio 2010.

## **2.2 Информационное обеспечение задачи**

Информационная модель содержит в себе данные об объектах и их свойствах. Некоторое количество таких моделей представляют собой базу данных, которая позволяет хранить всю информацию об объектах [20].

Информационная модель включает в себя логический и физический уровень [27]. На рисунке 2.1 показан логический уровень, он иллюстрирует сущности, а также взаимосвязь между ними.

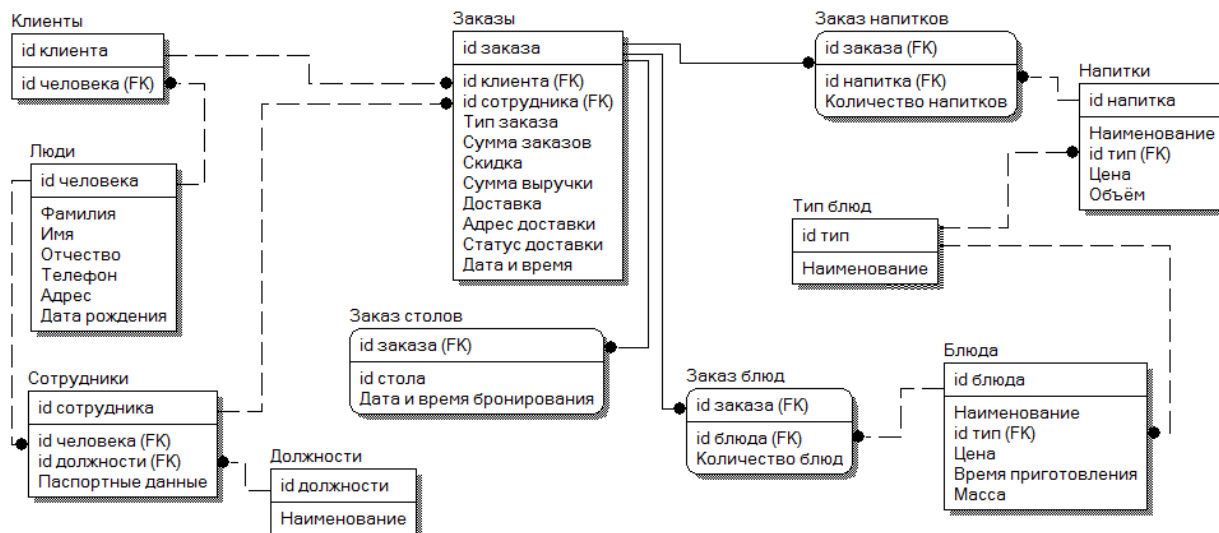


Рисунок 2.1 – Логический уровень

В данной выпускной квалификационной работе для работы подсистемы учета заказов в кафе были выделены следующие сущности:

1) Таблица «Тип блюд» - содержит информацию о возможных типах блюд в кафе. Так как один тип блюда может относиться к нескольким напиткам и в свою очередь к нескольким блюдам, были установлены связи «один ко многим» к таблице «Напитки» и отдельно к таблице «Блюда».

2) Таблица «Напитки» - содержит информацию о наименовании, цене и объеме напитка. Эта таблица связана с таблицей «Заказ напитков» связью «один ко многим», так как один напиток может быть выбран в нескольких заказах.

3) Таблица «Блюда» - содержит информацию о наименовании, цене, массе и время приготовления блюда. Данная таблица связана с таблицей «Заказ блюд» связью «один ко многим», так как одно блюдо может быть выбрано в нескольких заказах.

4) Таблица «Заказ напитков» - содержит информацию о идентификационном номере заказанного напитка и количестве заказанного за раз напитка.

5) Таблица «Заказ блюд» - содержит информацию об идентификационном номере заказанного блюда и количестве заказанного за раз блюда.

6) Таблица «Заказ столов» - содержит информацию об идентификационном номере заказанного стола и дате и времени его бронирования.

7) Таблица «Должности» - содержит информацию о наименовании должностей используемых на предприятии. Так как одна должность может относиться к нескольким сотрудникам, была установлена связь «один ко многим» к таблице «Сотрудники».

8) Таблица «Люди» - содержит личную информацию людей занесенных в подсистему.

Один и тот же человек может быть как барменом, так и в это же время администратором, поэтому была установлена связь «один ко многим» с таблицей «Сотрудники». Один и тот же человек может быть как сотрудником так и клиентом, поэтому была создана связь «один ко многим» с таблицей «Клиенты».

9) Таблица «Сотрудники» - содержит информацию об идентификационном номере человека, идентификационном номере должности и паспортные данные сотрудника.

10) Таблица «Клиенты» - содержит информацию об идентификационном номере человека в подсистеме.

11) Таблица «Заказы» - хранит информацию обо всех заказах, занесенных в подсистему. Здесь содержится такая информация как: тип заказа, сумма заказа, скидка, сумма выручки, доставка или нет, адрес доставки и её статус, дата и время внесенного заказа, идентификационные номера клиента и контактирующего с ним сотрудника.

Один и тот же заказ может содержать как заказ стола, так и заказ блюда и напитков, поэтому были установлены связи «один ко многим» с таблицами: «Заказ столов», «Заказ напитков» и «Заказ блюд». Но регистрируется заказ только на одного сотрудника и на одного клиента. Поэтому от таблиц «Клиенты» и «Сотрудники» были установлены связи «один ко многим» к таблице «Заказы».

Физический уровень представляет объекты логического уровня в том виде, в котором они будут размещаться, и использоваться в базе данных (рисунок 2.2) [36].

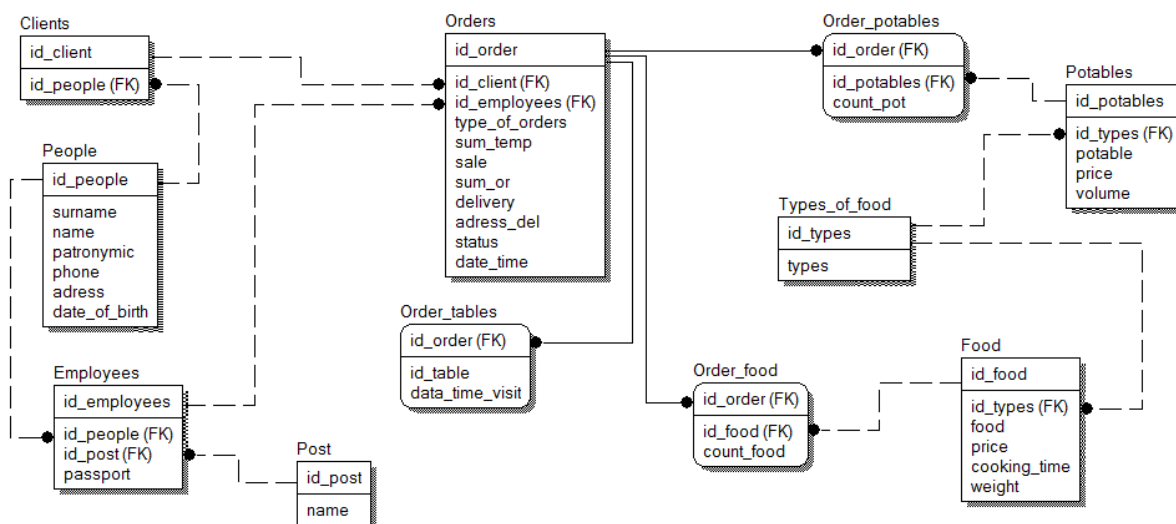


Рисунок 2.2 – Физический уровень

Во втором разделе выпускной квалификационной работы были обоснованы проектные решения по техническому, информационному, программному и технологическому обеспечению. Были выбраны программные средства, используемые при разработке подсистемы учета заказов. Также было создано информационное обеспечение задачи: спроектирован и описан логический, и физический уровень базы данных



### 3 Программная реализация проектных решений

#### 3.1 Разработка программного обеспечения задачи

##### 3.1.1 Разработка базы данных

С использованием утилиты IВExpert под управлением СУБД Firebird была создана и зарегистрирована база данных, на рисунке 3.1 представлена её регистрационная информация. Разработанная база данных хранится в файле KAFE.FDB.

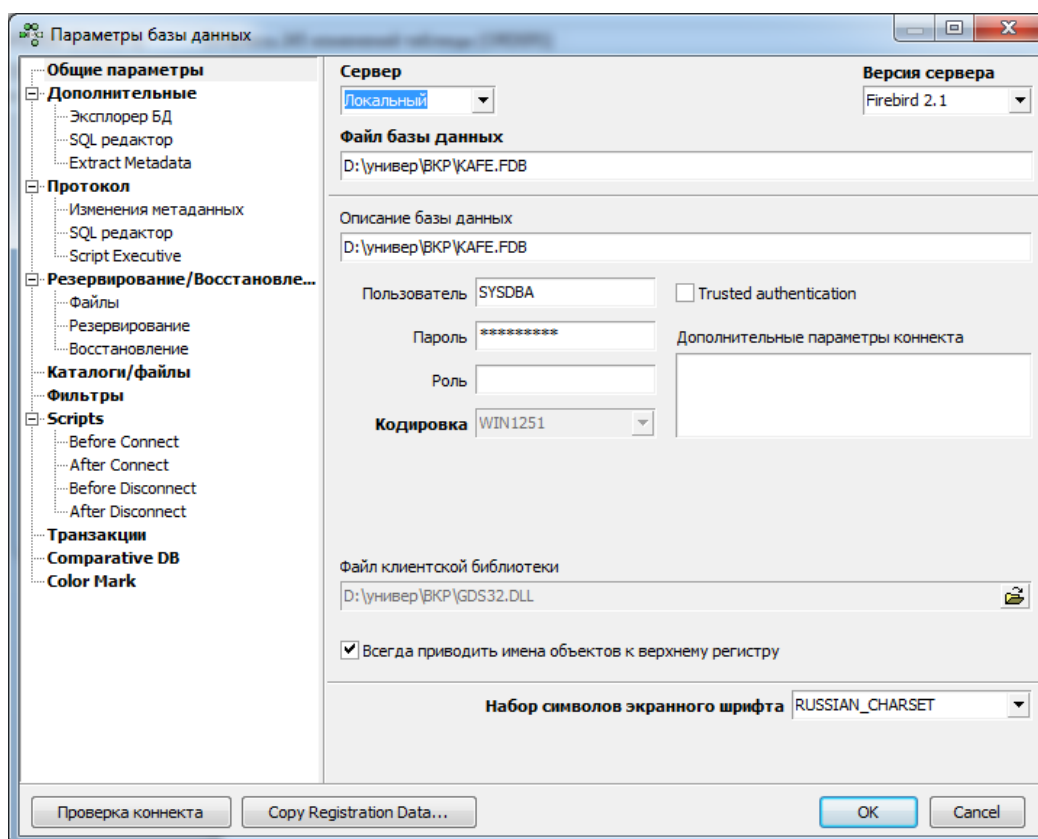


Рисунок 3.1 – Регистрационная информация базы данных

Перед созданием таблиц БД были созданы домены. Домен – это глобальное описание типа поля, которое доступно по всей базе данных [6].

Ниже представлен программный код разработки домена для поля Цена, который имеет вещественный тип данных.

Программный код: CREATE DOMAIN D\_PRICE AS FLOAT;

Подобным образом были созданы и другие домены базы данных.

Далее были разработаны таблицы БД. На рисунке 3.2 представлен программный код создания таблицы «Люди» с необходимыми ограничениями.

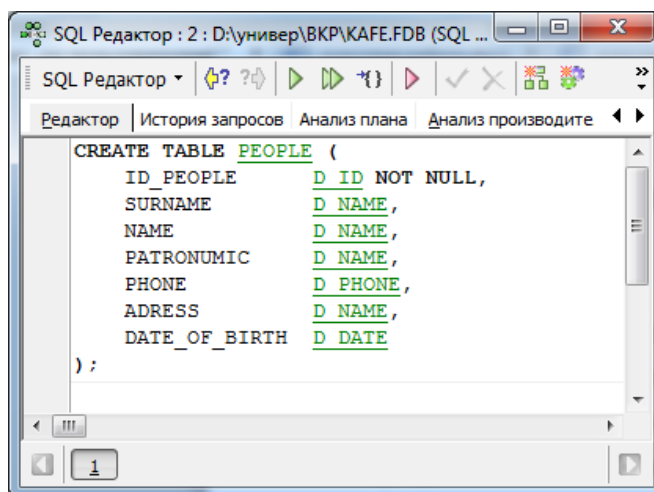


Рисунок 3.2 – Создание таблицы «Люди»

На рисунке 3.3 представлено создание таблицы «Заказы».

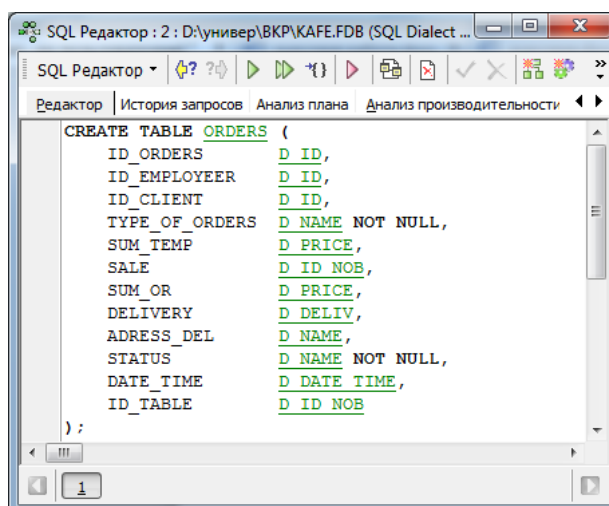


Рисунок 3.3 – Создание таблицы «Заказы»

Таким образом, были созданы оставшиеся девять таблиц.

Для автоматического наращивания и заполнения автоинкрементных полей были созданы триггеры и генераторы. На рисунке 3.4 представлено дерево разработанных компонентов.

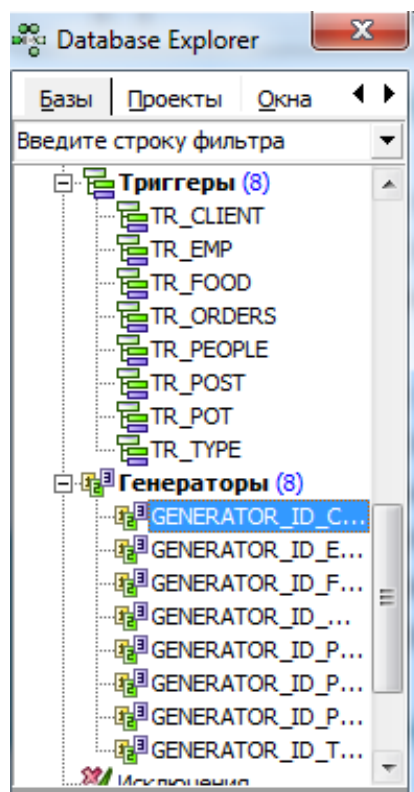


Рисунок 3.4 – Разработанные триггеры и генераторы

Далее были заполнены все таблицы базы данных. Заполнение может производиться как через программный код, так и через «Помощник» IVExpert [8]. При переходе на вкладку «Данные» можно добавлять, изменять и удалять записи из таблицы. На рисунке 3.5 представлена заполненная таблица «Блюда».

ID_FOOD	FOOD	ID_TYPES	PRICE	COOKING_TIME	WEIGHT
1	Французский поцелуй	1	270,500	00:30:00	200
2	Мясное оливье	1	150,000	00:25:00	200
3	Цезарь с курицей	1	280,000	00:25:00	250
4	Коктель	1	120,000	00:10:00	150
5	Дамский каприз	1	250,000	00:35:00	250
6	Ёка с зеленью	2	350,000	00:45:00	350
7	Гратен	2	300,000	00:30:00	300
8	Сырные крокеты	2	270,000	00:30:00	250
9	Овощи по азиатски	2	250,000	00:30:00	250
10	Фаршированные шампиньоны	2	300,000	00:45:00	350
11	Борщ украинский	3	150,000	00:15:00	200
12	Грибной суп	3	120,000	00:10:00	180
13	Куриный суп	3	100,000	00:05:00	180
15	Окрошка мясная	3	150,000	00:10:00	200
16	Картофель Фри	4	150,000	00:15:00	200
17	Цукини и помидоры	4	165,500	00:20:00	200

Рисунок 3.5 – Заполненная таблица «Блюда»

Далее были созданы представления БД. Представления – это виртуальная таблица, которая представляет собой результат разработанного запроса [2].

Ниже представлен программный код одного из разработанных представлений, который в итоге отразит фамилию сотрудников и сумму их выполненных заказов (рисунок 3.6).

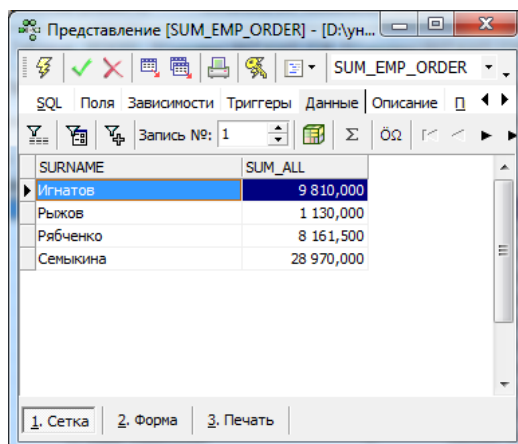
```

CREATE VIEW SUM_EMP_ORDER (
    SURNAME,
    SUM_ALL)
AS
select VIEW_ORDERS_SUM_3.surname_emp, sum(VIEW_ORDERS_SUM_3.sum_temp)
from VIEW_ORDERS_SUM_3
group by VIEW_ORDERS_SUM_3.surname_emp
;

```

Рисунок 3.6 – Программный код представления

На рисунке 3.7 представлена работа рассматриваемого выше представления.



SURNAME	SUM_ALL
Игнатов	9 810,000
Рыжков	1 130,000
Рябченко	8 161,500
Семькина	28 970,000

Рисунок 3.7 – Разработанное представление

Далее были разработаны хранимые процедуры на добавление, модификацию и удаление записей. На рисунке 3.8 представлен список разработанных процедур.

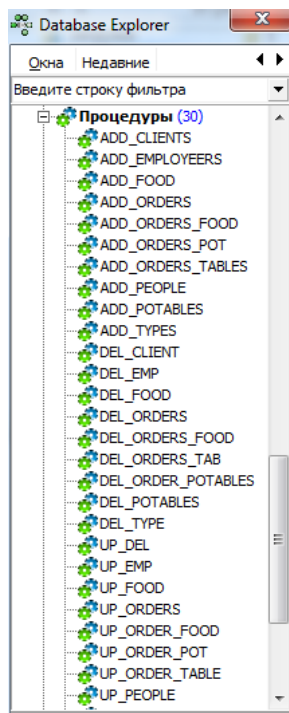


Рисунок 3.8 – Разработанные хранимые процедуры

Программный код, который использовался в создании компонентов базы данных, представлен в приложение В.

### 3.1.2 Разработка пользовательского интерфейса

Совокупность методов и средств, с помощью которых человек взаимодействует с аппаратурой, различными устройствами или машинами называют – интерфейсом [3].

На рисунке 3.9 представлена модульная структура разрабатываемой подсистемы учета заказов. Другими словами DataModule – это контейнер, для компонентов доступа к данным [24].

Он содержит в себе следующие компоненты:

- IVDatabase – это центральный компонент для соединения с базой данных;
- IVTransaction – предназначенный для управления транзакциями;
- DataSource – является посредником между компонентами IVTable, IVQuery и компонентами DBGrid, DataControls – являющимися элементами управления и представления данных;
- IVTable – предоставляет доступ к таблице;
- IVQuery – набор данных, который формируется в результате выполнения запроса на языке SQL;
- IVStoredProc – это компонент, который запускает одну из разработанных хранимых процедур.

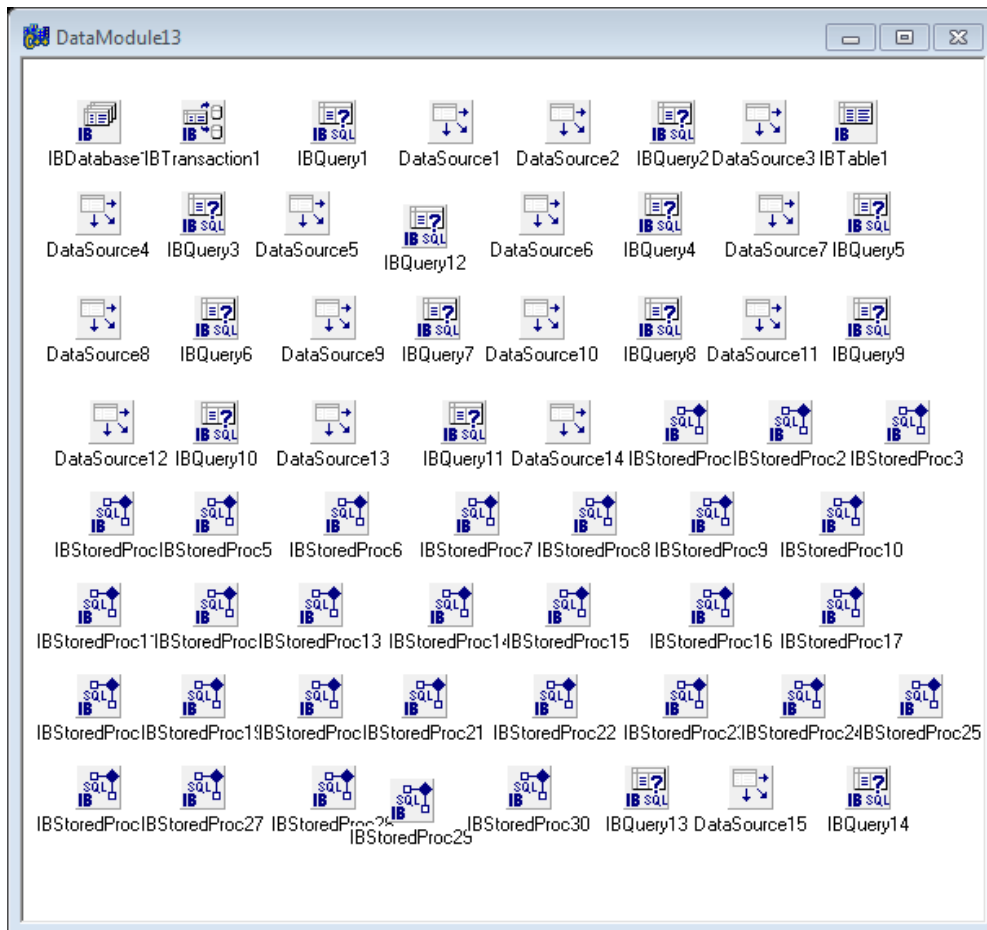


Рисунок 3.9 – Модульная структура разрабатываемой подсистемы

С помощью компонента IBDatabase был настроен доступ к базе данных (рисунок 3.10).

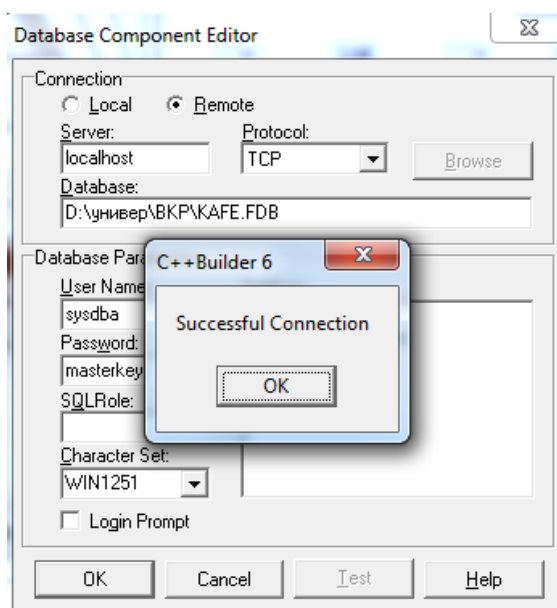


Рисунок 3.10 – Подсоединение базы данных к подсистеме

Далее было разработано главное окно подсистемы, на фоне которого расположены рисунки, а сверху с помощью компонента MainMenu на форму была добавлена навигационная панель, которая содержит переход на вкладки: клиенты, сотрудники, заказы, меню, доставка и выход. Главное окно представлено на рисунке 3.11.

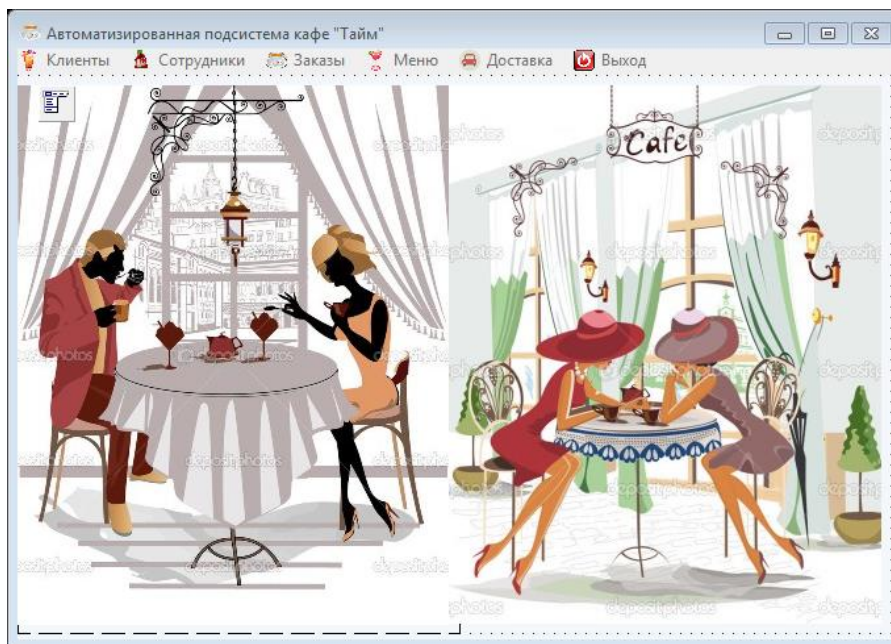


Рисунок 3.11 – Главное окно разрабатываемой подсистемы

На следующей форме представлена разработанная вкладка «Список клиентов» (рисунок 3.12), она содержит следующие компоненты: Label – для отображения текста на форме; компонент DBGrid – для отображения таблицы базы данных; Button – кнопки, которые выполняют заданные им функции; Edit – поля ввода данных; DateTimePicker – компонент, который используется для ввода даты и времени; GroupBox – предназначен для объединения компонентов; ComboBox – это выпадающий список элементов [26].



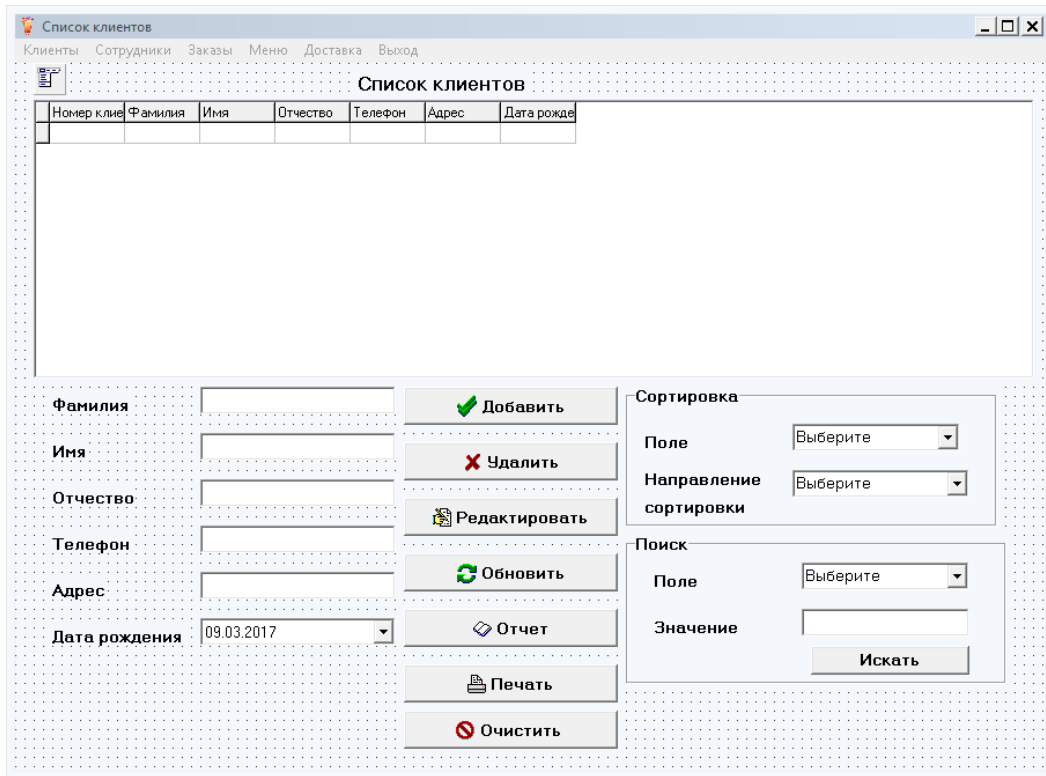


Рисунок 3.12 – Разрабатываемая вкладка «Список клиентов»

Следующая форма содержит аналогичные компоненты. На рисунке 3.13 представлена вкладка «Список сотрудников».

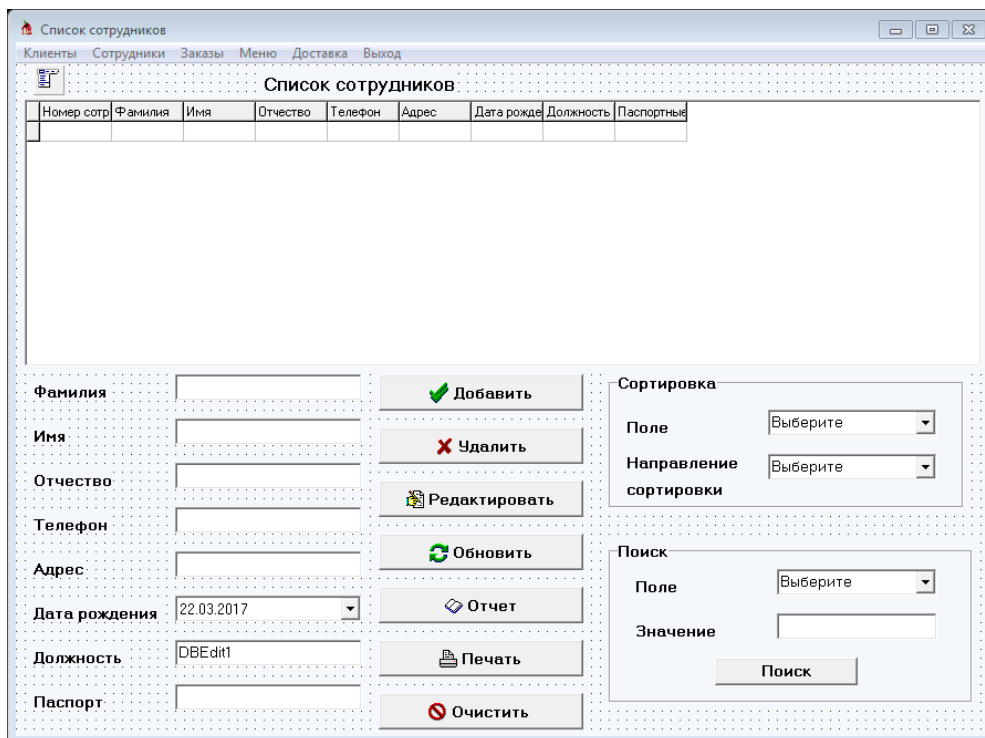


Рисунок 3.13 – Разрабатываемая вкладка «Список сотрудников»

На следующей форме представлена вкладка «Заказ столов», которая содержит аналогичные предыдущей вкладки компоненты: поля ввода, кнопки, выпадающие списки, поле для отображения таблицы базы данных (рисунок 3.14).

Номер заказа	Номер стола	Дата и время визита
--------------	-------------	---------------------

Сортировка

Поле: Выбрать

Направление сортировки: Выбрать

Поиск

Поле: Выбрать

Значение:

Поиск

Номер заказа: DBEdit1

Номер стола:

Дата визита: 26.03.2017

Время визита: 20:20:44

Добавить

Удалить

Редактировать

Очистить

Обновить

Отчет

Печать

Рисунок 3.14 – Разрабатываемая вкладка «Заказ столов»

Последующие формы «Заказ блюд», «Заказ напитков», «Список блюд», «Список напитков» были выполнены аналогично рассмотренным выше формам.

Следующая форма схожа с предыдущими, за исключением добавления компонента DBChart, этот компонент позволяет строить эффективные диаграммы и графики. На рисунке 3.15 представлена форма «Заказы» вкладка «Учет заказов».

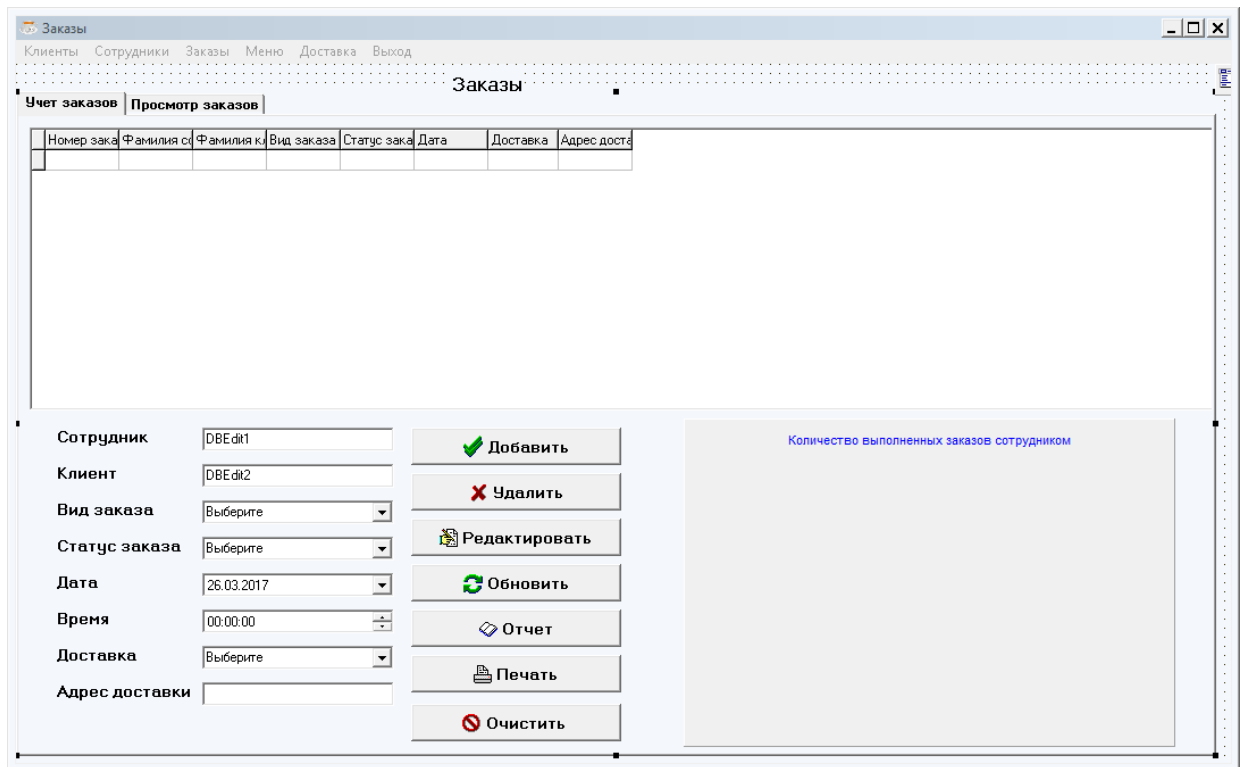


Рисунок 3.15 – Разрабатываемая форма «Заказы»

На форме «Отчеты» представлены следующие компоненты: QuickRep – это основа для создания отчета; IBTable – это компонент, который предназначен для выбора информации из таблиц в отчеты; DataSource – этот компонент необходим для установления связи между компонентами на форме и таблицей базы данных [17]. На рисунке 3.16 представлена форма «Отчеты», на которой находятся разработанные в подсистеме отчеты: о клиентах, напитках, блюдах, заказах напитков, блюд и столов, отчет о доставке, о заказе столов и так далее.

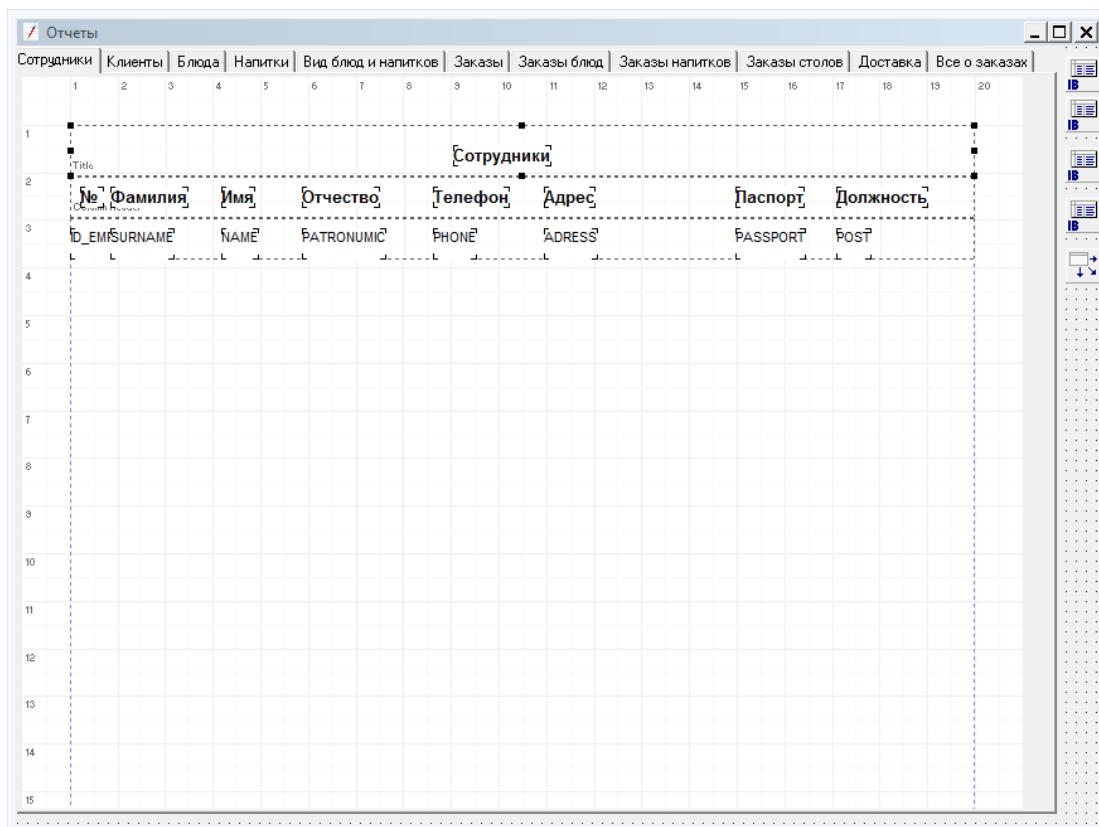


Рисунок 3.16 – Разрабатываемая форма «Отчеты»

Последняя разрабатываемая вкладка, это «Авторизация», она включает в себя следующие компоненты: Label, ComboBox, Edit и Button (рисунок 3.17).

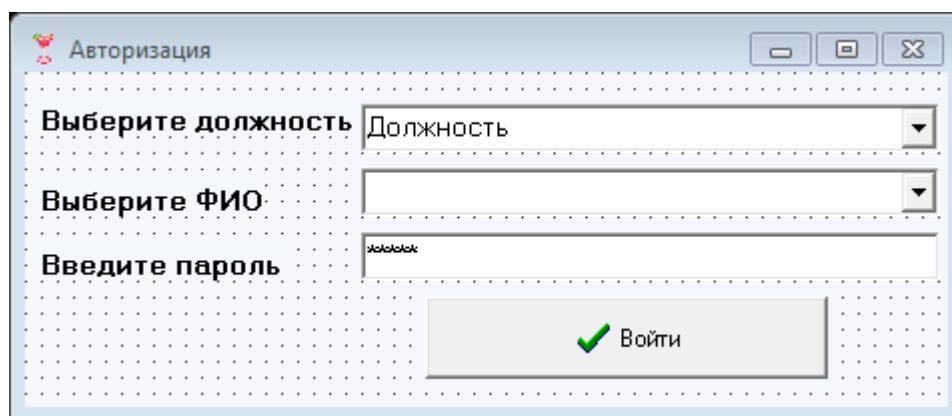


Рисунок 3.17 – Разрабатываемая форма «Авторизация»

Программный код разрабатываемой подсистемы и скриншоты разработки, которые не были отображены в данном разделе, представлены в приложение В.

## 3.2 Контрольный пример реализации проекта

Для начала работы с подсистемой необходимо открыть файл Project1.exe, появится окно авторизации, выбрать занимаемую должность, ФИО и ввести пароль (рисунок 3.17).

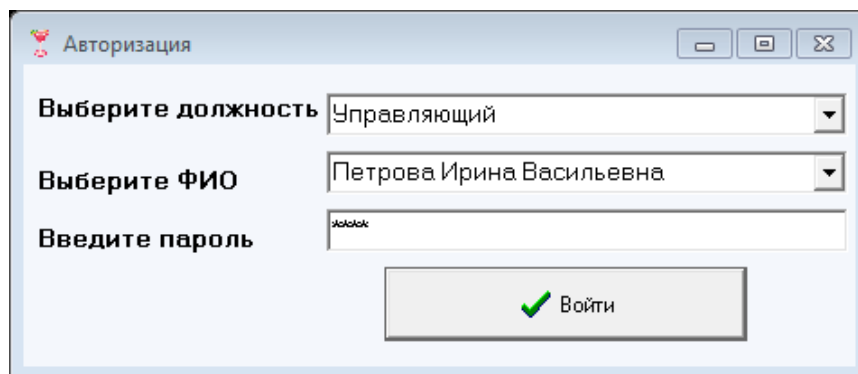


Рисунок 3.17 – Окно «Авторизация»

Далее попадаем в главное меню автоматизированной подсистемы учета заказов кафе «Тайм», навигационная панель содержит такие кнопки перехода как «Клиенты», «Сотрудники», «Заказы», «Меню», «Доставка» и «Выход» (рисунок 3.18).

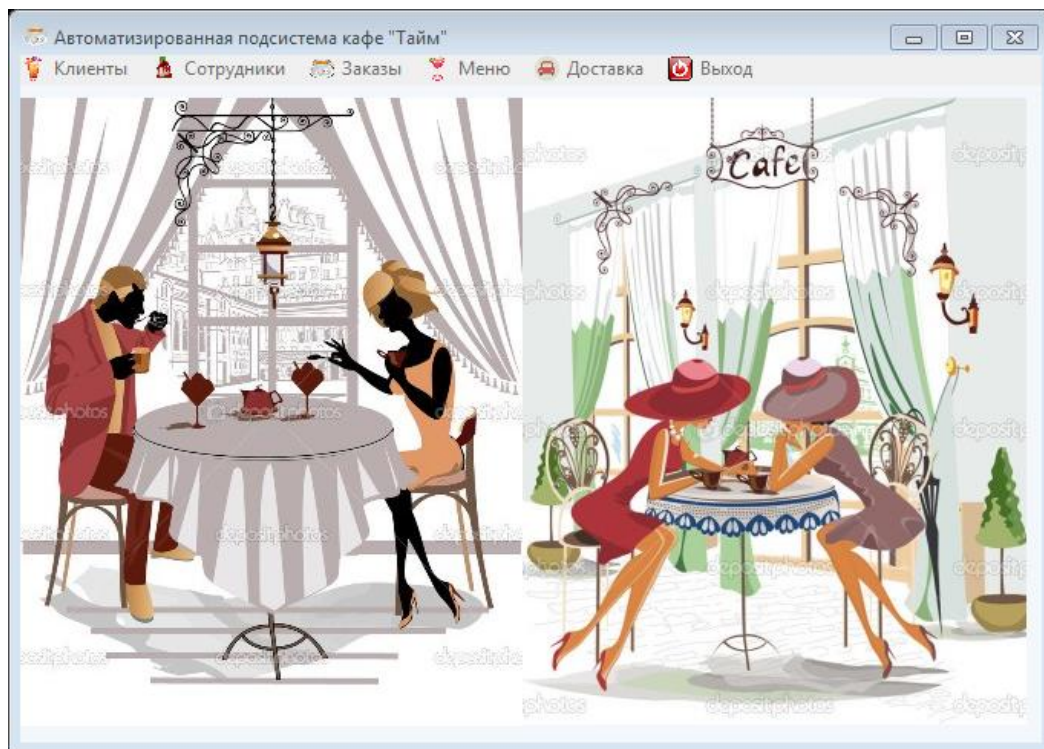


Рисунок 3.18 – Главное окно Автоматизированной подсистемы кафе «Тайм»

Кнопка «Выход» возвращает пользователя обратно на авторизацию, выход происходит только из учетной записи.

Вход выполнен через Администратора, он имеет право работать со всей информацией, хранящейся в базе данных, а так же просматривать все виды отчетов.

Перейдем на вкладку «Сотрудники». На данной вкладке выведен список сотрудников, имеется возможность добавлять, удалять, обновлять и редактировать записи, очистить вводимую в поля информацию, печатать и просматривать отчет по таблице, а также выполнить поиск по необходимому значению и произвести сортировку по определенному полю.

На рисунке 3.19 и 3.20 представлена работа кнопки «Очистить».

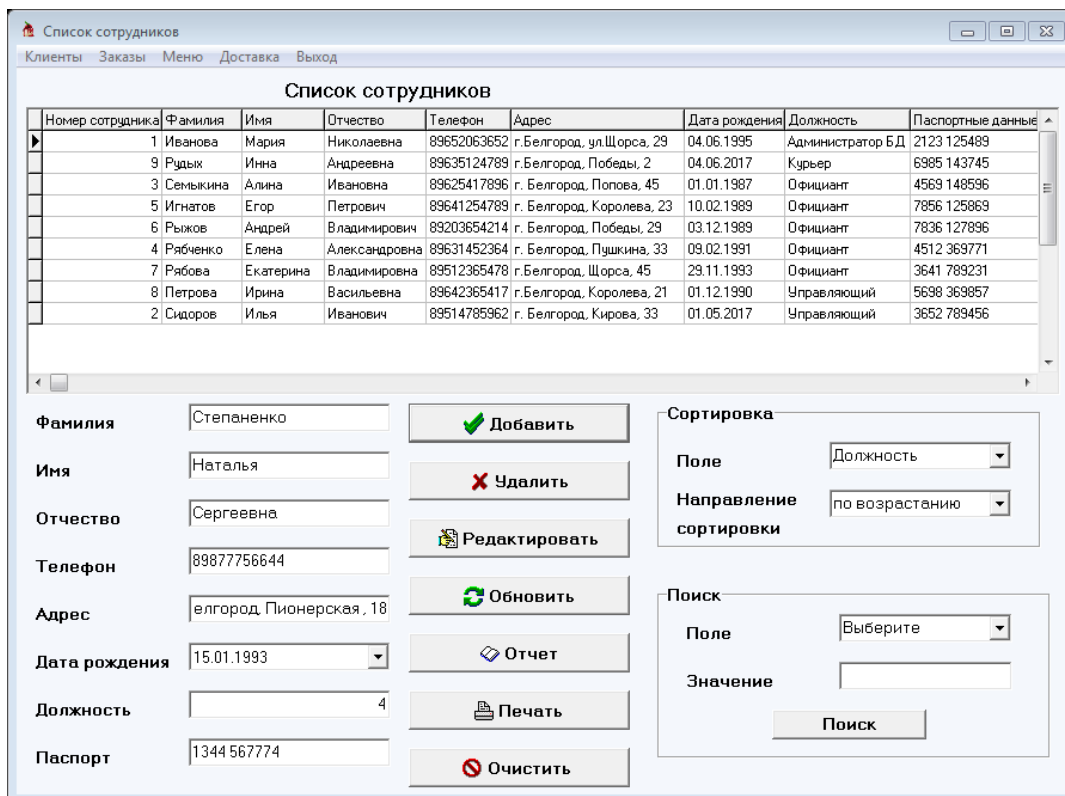


Рисунок 3.19 – Ввод информации

Нажав на кнопку «Очистить» поля, вводимые вручную, очищаются (рисунок 3.20).

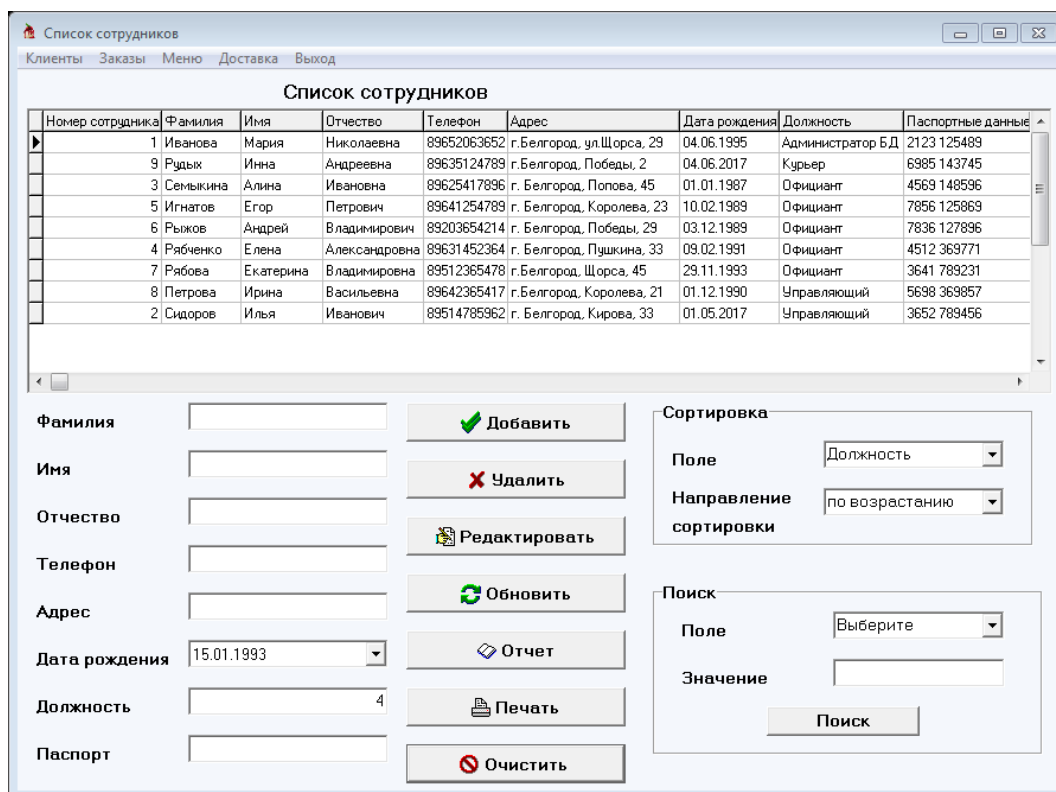


Рисунок 3.20 – Работа кнопки «Очистить»

Рассмотрим подробнее удаление записи на вкладке «Список клиентов». Для добавления необходимо ввести полную информацию о клиенте в поля ввода (рисунок 3.21).

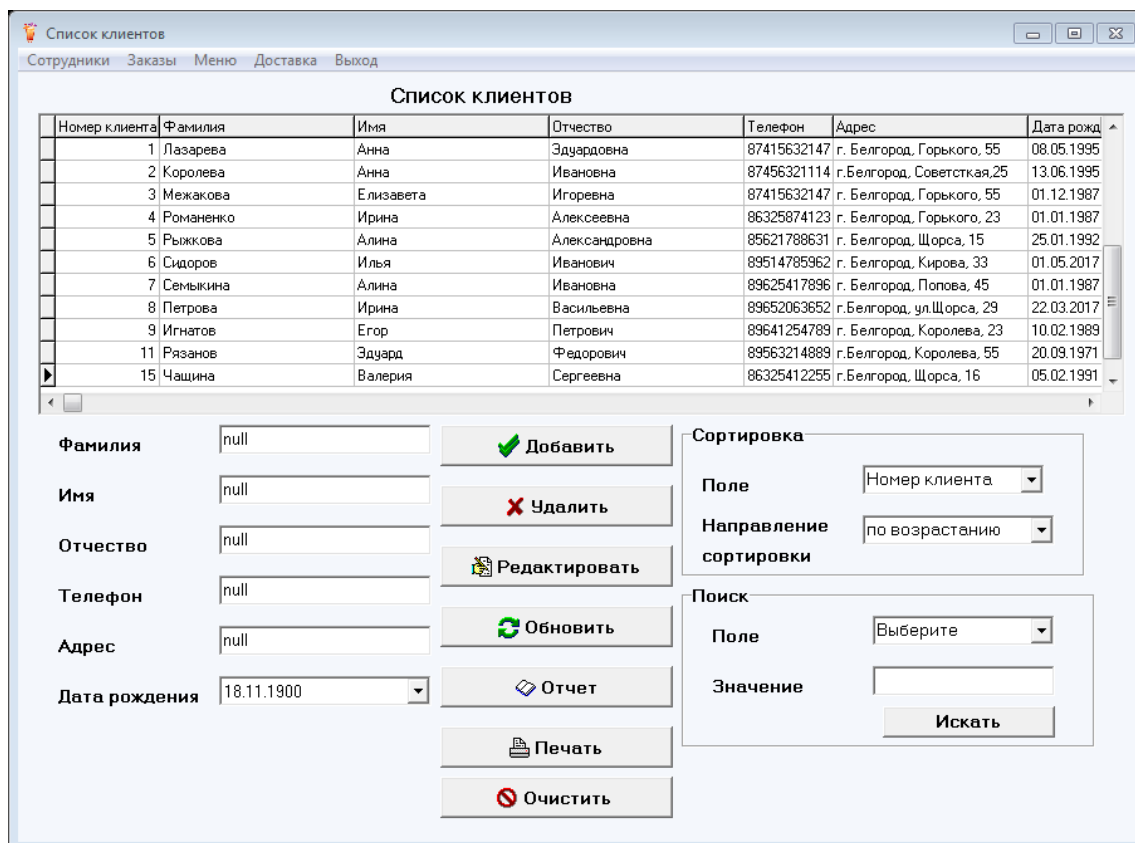


Рисунок 3.21 – Ввод информации о новом клиенте

Далее нажав на кнопку «Добавить», в списке появится новая запись (рисунок 3.22).



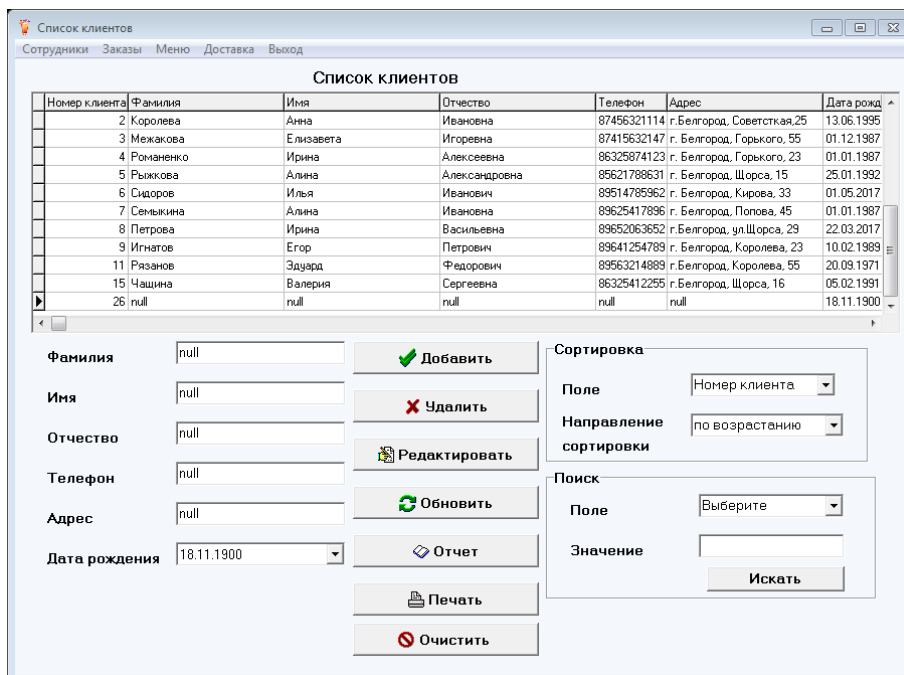


Рисунок 3.22 – Работы кнопки «Добавить»

На примере окна «Список блюд» рассмотрим работу сортировки. На рисунке 3.23 представлен список блюд, отсортированный автоматически при выводе таблицы по номеру блюда.

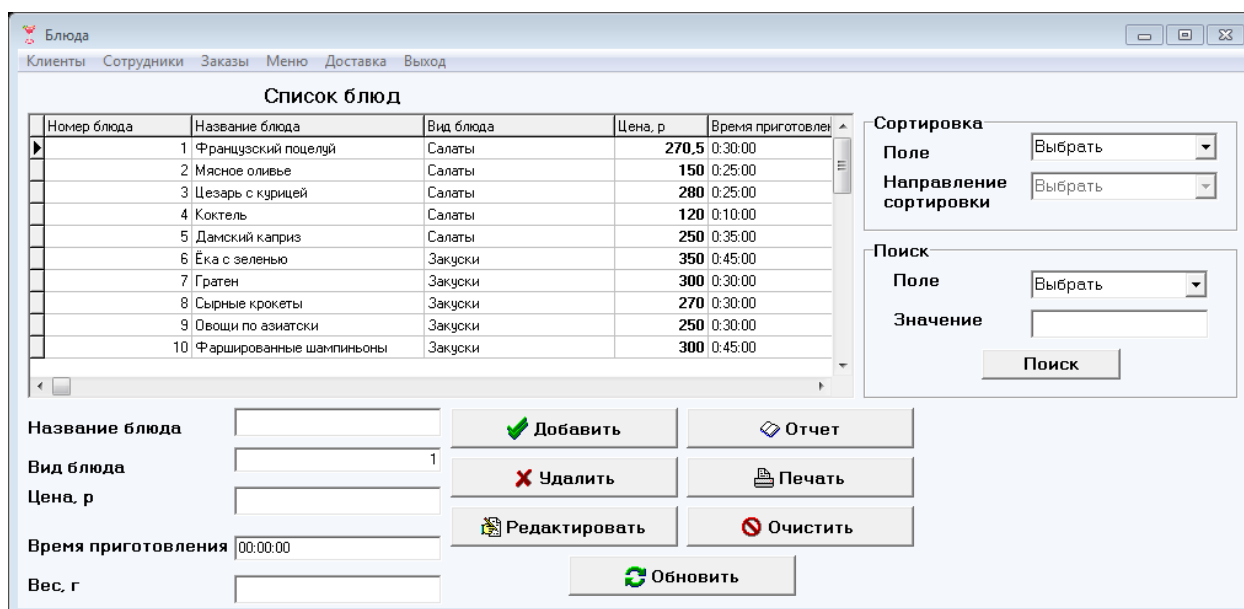


Рисунок 3.23 – Разработанное окно «Список блюд»

Выберем критерии сортировки: поле – по цене; направление – по возрастанию, полученный результат представлен на рисунке 3.24.

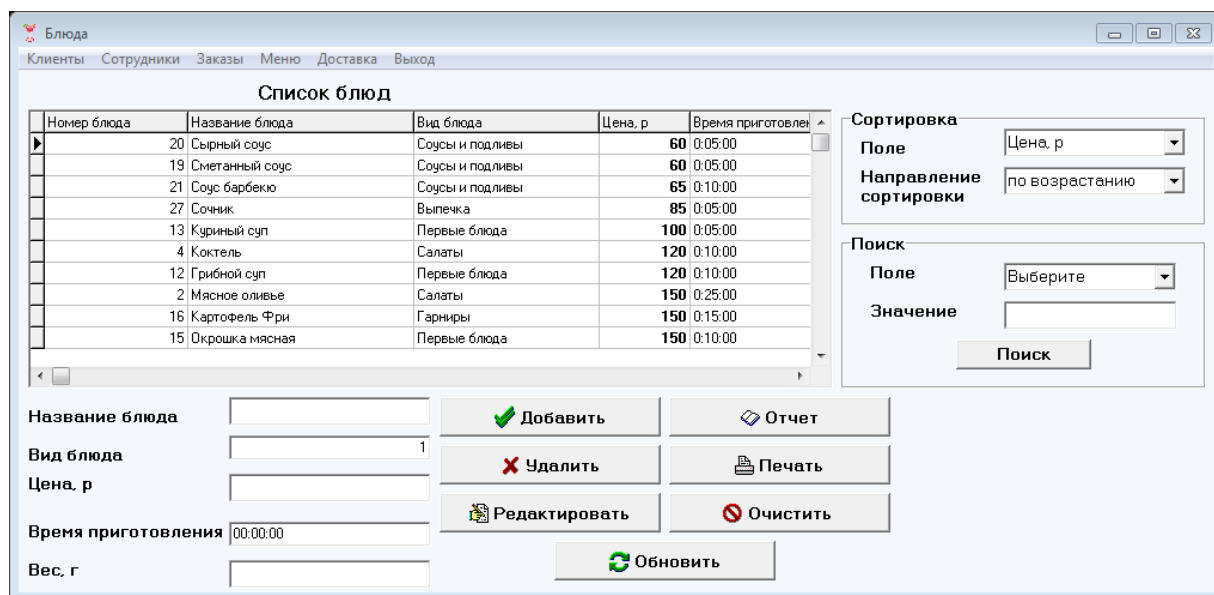


Рисунок 3.24 – Выполнение сортировки

На примере окна «Вид напитков и блюд» рассмотрим работу поиска. Для начала введем значение для поиска в таблице (рисунок 3.25)

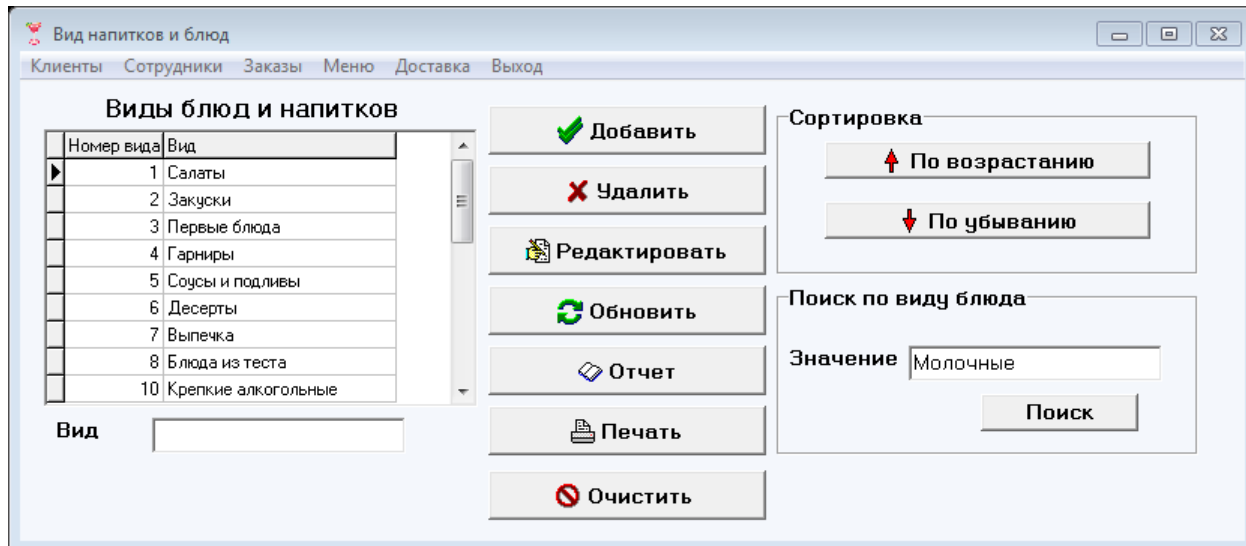


Рисунок 3.25 – Ввод значения для поиска

Далее нажав на кнопку «Поиск» выполнится разработанный запрос и в окне вывода таблицы появится значение, которое искали (рисунок 3.26).

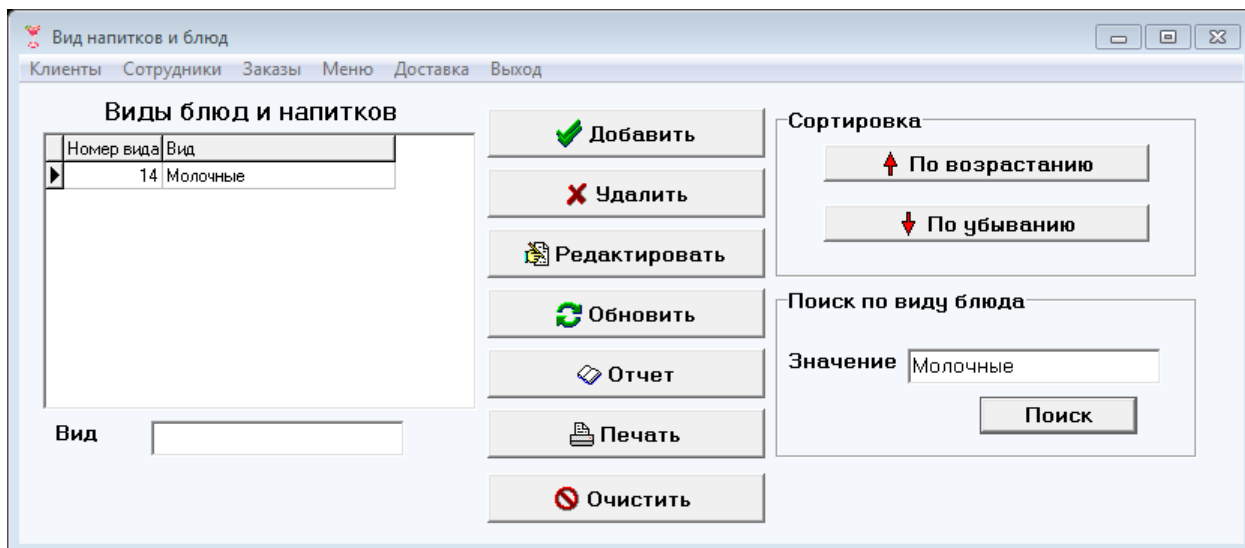


Рисунок 3.26 – Работа кнопки «Поиск»

На рисунке 3.27 представлена работы кнопки «Отчет» на вкладке заказы. Данный отчет позволяет просмотреть всю информацию о заказах, занесенных в подсистему.

Заказы								
№	Клиент	Сотрудник	Доставка	Стоимость	Скидка	Итого	Статус	Вид заказа
1	Межакова	Семькина	да	8955	10	8059,5	выполнен	Заказ блюд
2	Лазарева	Семькина	нет	19380	10	17442	принят	Заказ блюд
3	Королева	Игнатов	да	7710	10	6939	принят	Заказ блюд
4	Рыжкова	Рябченко	нет	3196,5	10	2876,85	приготовление	Заказ блюд
5	Лазарева	Игнатов	нет	2100	10	1890	выполнен	Заказ блюд
6	Межакова	Рябченко	нет				принят	Заказ столов
7	Рыжкова	Рыжов	нет				выполнен	Заказ столов
8	Рыжкова	Семькина	нет				принят	Заказ блюд
10	Петрова	Рыжов	нет	1130			выполнен	Заказ блюд

Рисунок 3.27 – Отчет о заказах

Таким образом, были рассмотрены некоторые из возможностей пользователя «Администратор». Рассмотрим возможности следующего пользователя. Нажав на кнопку «Выход», высвечивается смена пользователя,

устанавливаем пароль для выбранной должности и попадаем на главное окно подсистемы (рисунок 3.28), интерфейс администратора от официанта отличается возможностью работать со списком сотрудников, официантам такое право не отведено.

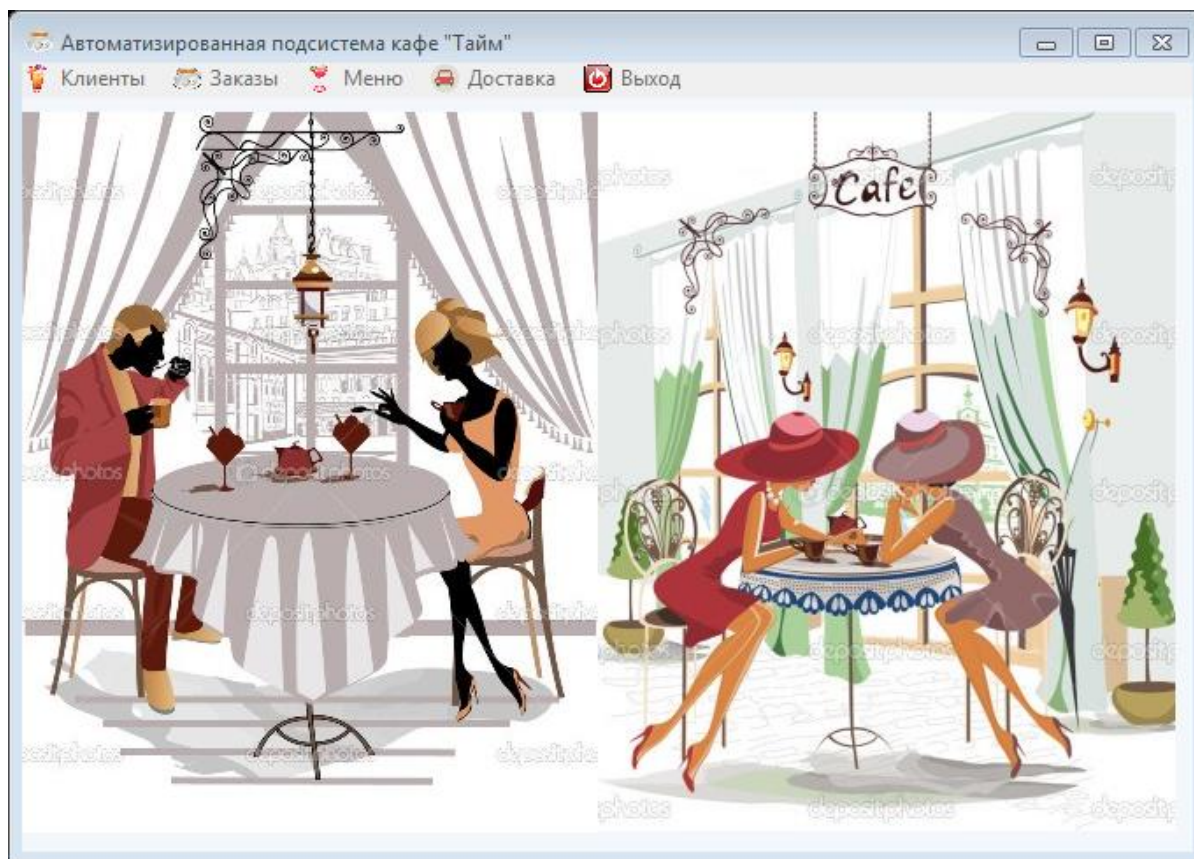


Рисунок 3.28 – Главное окно подсистемы кафе «Тайм»

Рассмотрим, как происходит проведение заказа. Для начала необходимо создать заказ на вкладке «Учет заказов». Для этого заполняем все поля ввода, заполнение поля «Сотрудник» и «Клиент» происходит с помощью выбора необходимой записи из разработанных справочников (рисунок 3.29).

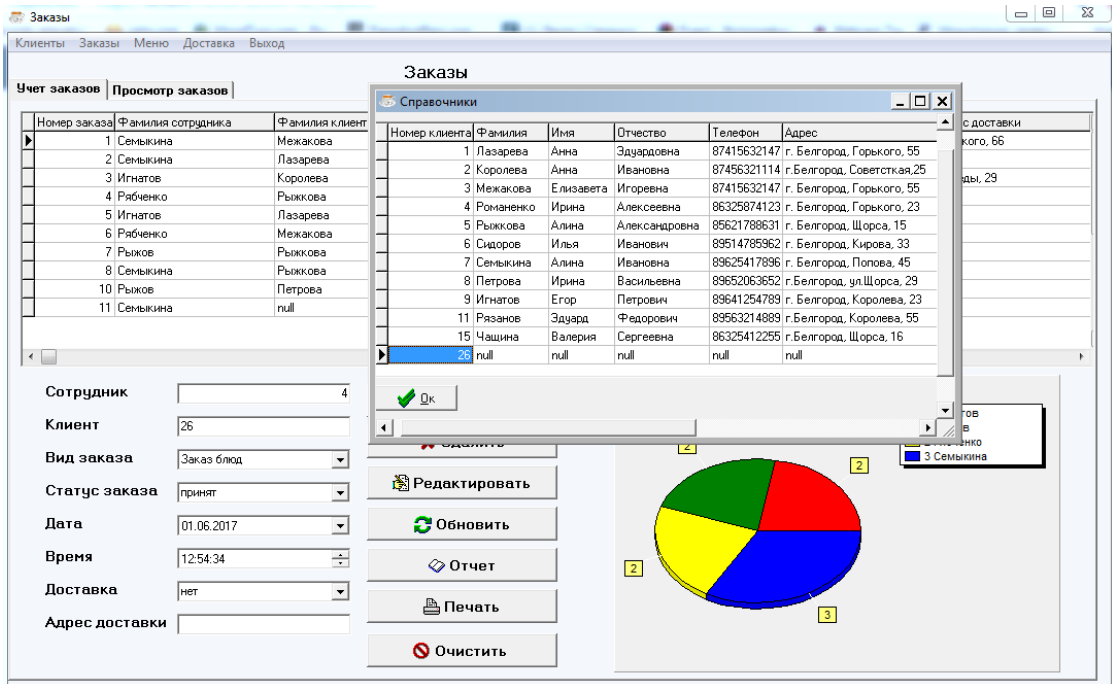


Рисунок 3.29 – Выбор клиента из справочника

Так как клиенты в момент посещения кафе не распространяют свою личную информацию, это значит, что их не добавляют в базу данных, была специально создана пустая запись. На рисунке 3.30 представлены все заполненные поля.

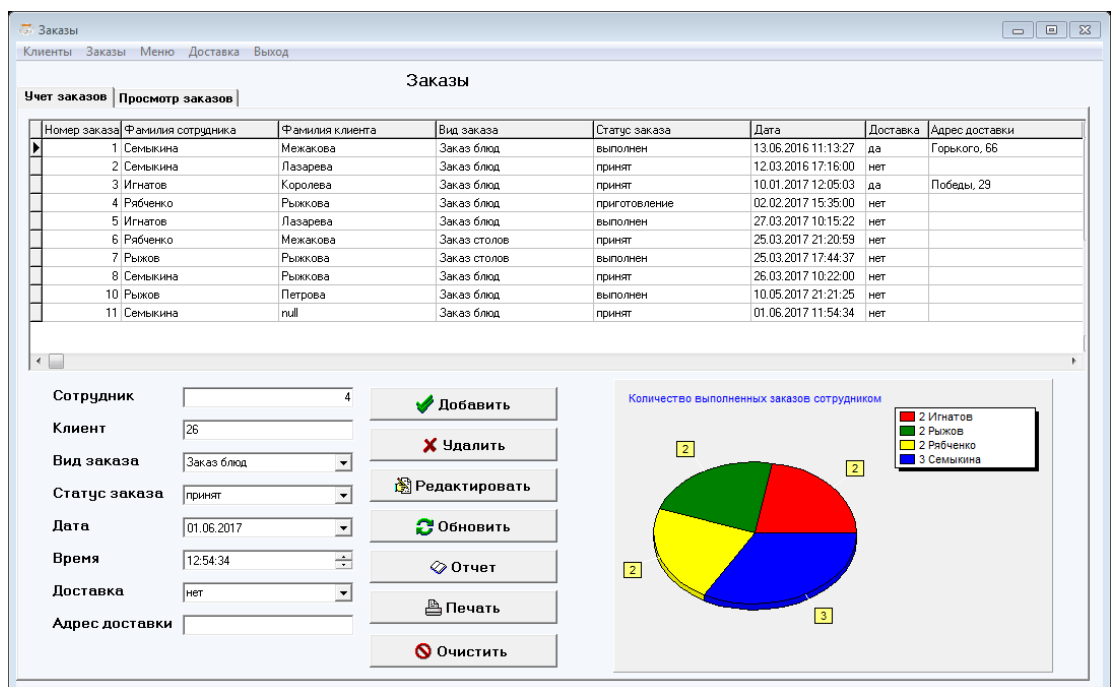


Рисунок 3.30 – Заполненные поля для добавления заказа

Добавив заказ в базу, переходим на вкладку «Заказ блюд». Создаем новый заказ блюда: из появившихся справочников необходимо выбрать нужный номер заказа и номер блюда, которое заказал клиент, далее поставить необходимое количество и нажав на кнопку добавить, заказ будет сформирован (рисунок 3.31).

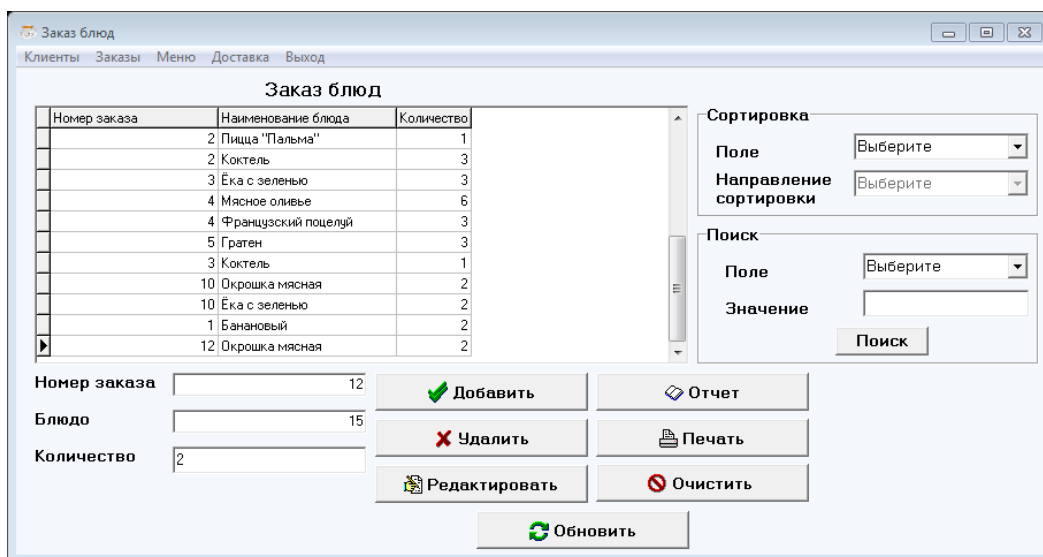


Рисунок 3.31 – Добавление блюда в заказ

Аналогичным образом добавляем в заказ напиток (рисунок 3.32).

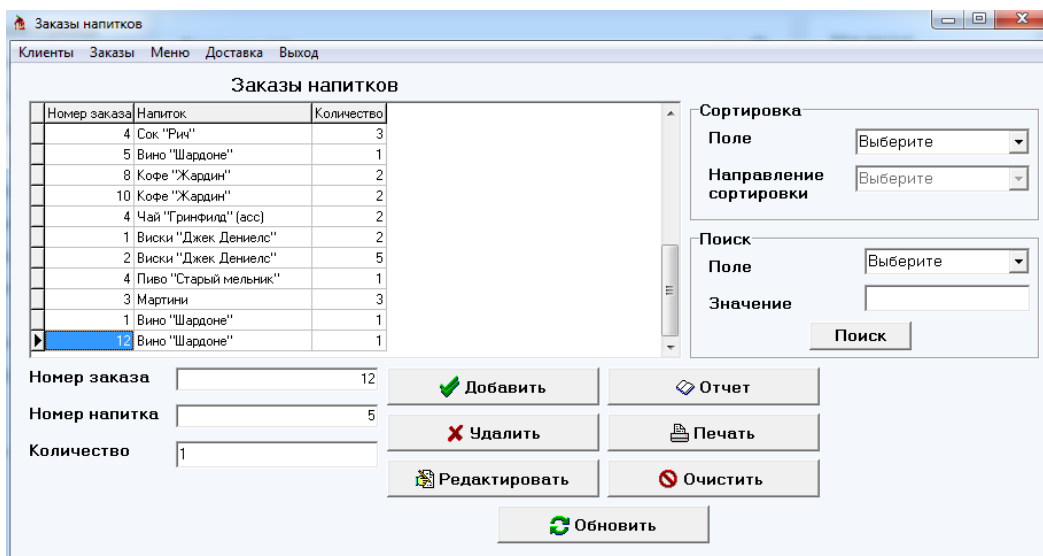


Рисунок 3.32 – Добавление напитка в заказ

Теперь перейдем на вкладку просмотр заказов, здесь можно просмотреть итоговую сумму созданного заказа (номер 12), а также наличие или отсутствие скидки (рисунок 3.33).

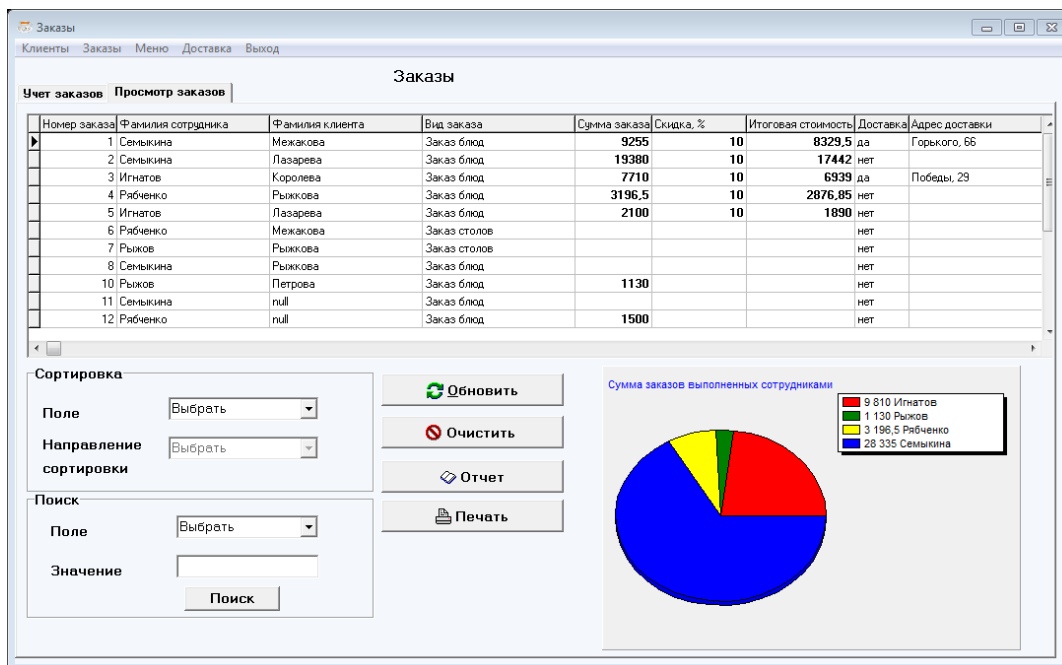


Рисунок 3.33 – вкладка «Просмотр заказов»

Помимо простых операций на добавление, удаление, модификацию, сортировку, поиск и так далее. Существуют аналитические отчеты, представленные на вкладках «Просмотр заказов» и «Учет заказов». Они позволяют следить за работой и успеваемостью сотрудников. На рисунке 3.34 представлена диаграмма, которая показывает сумму заказов выполненных сотрудниками.

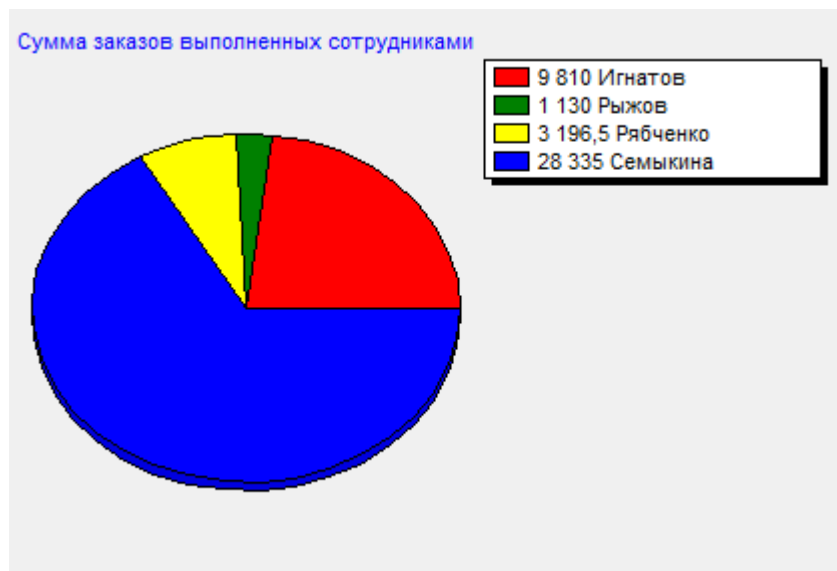


Рисунок 3.34 – Аналитический отчет «Сумма заказов выполненных сотрудниками»

На рисунке 3.35 представлена диаграмма, на которой отображено количество выполненных сотрудниками заказов.

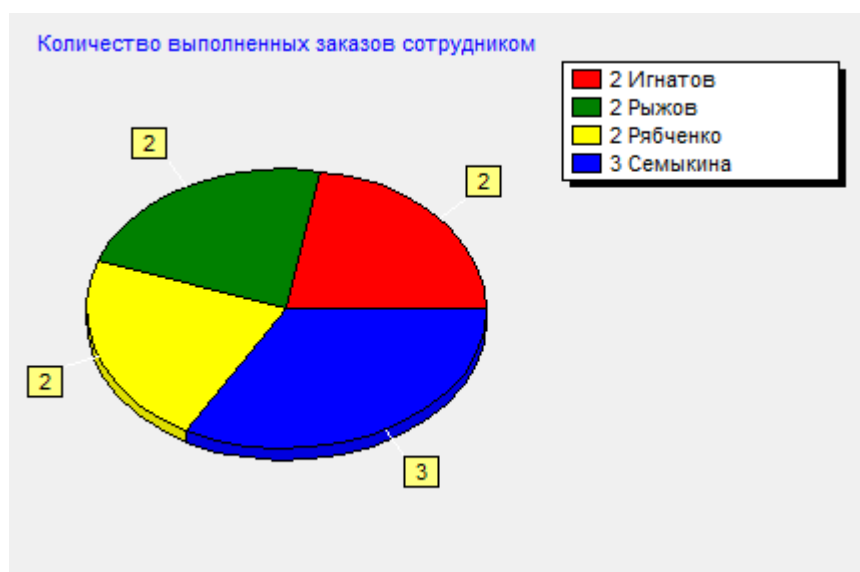


Рисунок 3.35 – Аналитический отчет «Количество выполненных заказов сотрудником»

На рисунке 3.36 представлен отчет «Доставка заказов», который отображает все внесенные в подсистему заказы на дом.



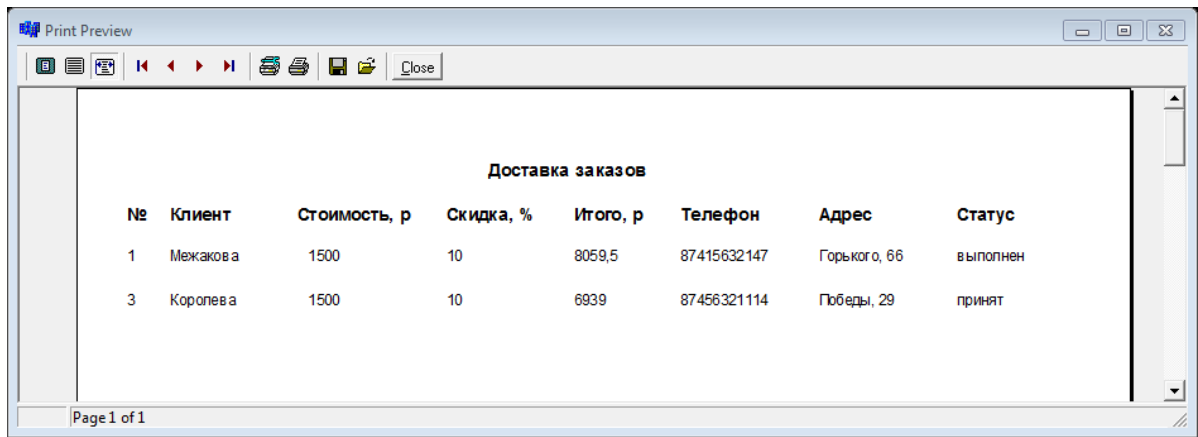


Рисунок 3.36 – Отчет «Доставка заказов»

Снова переходим в окно «Авторизация», здесь также имеется возможность зайти как «Системный администратор» и «Курьер».

Войдя в подсистему как «Курьер» главное окно имеет только две кнопки на панели инструментов: «Доставка» и «Выход». То есть курьер имеет право только просматривать информацию о доставке (рисунок 3.37).

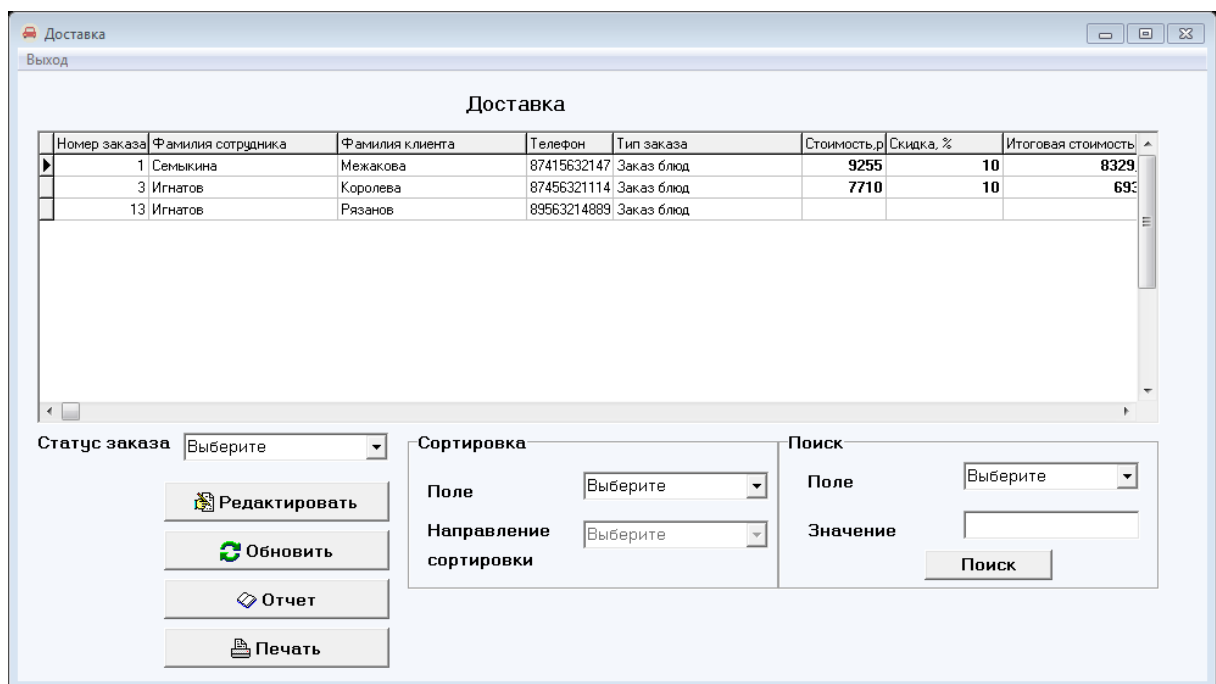


Рисунок 3.37 – Окно «Доставка»

Таким образом, были рассмотрены все основные функциональные возможности разработанной автоматизированной подсистемы учета заказов.

Скриншоты работы подсистемы, которые не были представлены в главе три, отображены в приложение Г.

### **3.3 Расчет экономической эффективности разработанной подсистемы**

Для расчета экономической эффективности разработанной подсистемы подсчитаем время, затраченное на её разработку. Для этого определим показатели трудоёмкости выполнения выпускной квалификационной работы (таблица 3.1).

Таблица 3.1 – Показатели трудоёмкости выполнения ВКР

Этапы	Содержание этапов	Доля этапа трудоёмкости
1) Подготовительный этап	Анализ и изучение научно-технической литературы по заданной предметной области.	30 дней
2) Выполнение теоретических разработок	Разработка схем и теоретических обоснований, систематизация теоретических разработок и составление расчетов и проектов.	31 день
3) Проектирование подсистемы	Разработка схем и диаграмм. Обоснование проектных решений.	45 дней
4) Создание базы данных	Создание и заполнение базы данных	30 дней
5) Создание интерфейса	Разработка интерфейса подсистемы	50 дней
6) Программирование подсистемы	Программирование разработанного интерфейса	60 дней
7) Техническая отчетность	Определение экономического эффекта и составление отчета	14 дней
8) Заключительный этап	Оформление и утверждение результатов работы	10 дней
	<b>ИТОГО:</b>	<b>270 дней</b>

Рассчитав показатели трудоёмкости выполнения ВКР получили – девять месяцев, это время, затрачиваемое на разработку подсистемы учета заказов. Далее рассчитываем эксплуатационные расходы до разработки и внедрения подсистемы. Основные формулы, используемые в расчетах, представлены в приложение Д.

В таблице 3.2 приведены исходные данные для расчетов.

Таблица 3.2 – Исходные данные

№ п/п	Наименование показателя	Ед. измер.	Значения показателей	
			до создания	после создания
1)	Коэффициент приведенных затрат к единому году	%	-	25
2)	Коэффициент отчислений	%	30	30
3)	Стоимость электроэнергии	руб.	2,53	2,53
4)	Заработная плата программиста	руб.	10000	10000
5)	Заработная плата инженера	руб.	5000	-
6)	Заработная плата инженера	руб.	5000	-
7)	Заработная плата инженера	руб.	5000	-
8)	Заработная плата системного администратора	руб.	10000	10000
9)	Время на разработку	мес.	9	9

Рассчитаем эксплуатационные расходы на функционирование подсистемы за один месяц потребление электроэнергии – 200 кВт/ч.

$$P_{\text{эн}} = 200 * 9 * 2.53 = 4554 \text{ руб.}$$

Расходы на материалы до внедрения ИС приведены в таблице 3.3.

Таблица 3.3 – Расходы на материалы до внедрения ИС

Материалы	Ед. измер.	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	шт	200	2	200
Канцтовары	-		-	100
USB Флеш-накопитель	шт.	1	500	700
Итого				800

Расходы на материалы за девять месяцев составили:

$$P_M = (200+100)*9+700= 3400\text{руб.}$$

Поскольку  $P_{\text{общ}} + P_{\text{пр}} = 4554+3400=7954.00$  руб., то

$$P_{\text{пр}} = 0.60*7954.00=4772.40 \text{руб.},$$

Следовательно, общепроизводственные расходы равны:

$$P_{\text{общ}} = 7954.00+4772.40= 12726.40 \text{руб.}$$

До разработки подсистемы работали три инженера, программист и системный администратор. Зарботная плата основного персонала составила:

$$P_{\text{осн}} = (10000.00+5000.00+5000.00+5000.00+10000.00)*9*0.30=94500.00 \text{руб.}$$

Отсюда, зарботная плата административно-управленческого персонала составила:

$$P_{\text{ауп}} = 0.25*94500.00=23625.00 \text{руб.}$$

Таким образом, эксплуатационные расходы на решение задачи до внедрения системы составляют:

$$P_0 = 94500.00+23625.00+12756.40=130881.40 \text{руб.}$$

Далее рассчитываем затраты после разработки автоматизированной подсистемы.

Разработкой подсистемы занимается программист, в месяц он получает 10000 руб. Фактическое время, затраченное на разработку подсистемы – 9 месяцев. Поскольку разработка программного обеспечения осуществлялась на уже имеющемся компьютере, затраты на основное оборудование в расчет не включаются.

Капитальные затраты на проектирование равны:

$$K_{\text{пр}} = 10000.00 * 9 * 0.30$$

$$K = 27000.00 \text{ руб.}$$

Далее рассчитываются расходы за девять месяцев после внедрения подсистемы:

Расходы на электроэнергию – 300 кВт/ч.

$$P_{\text{эн}} = 200 * 9 * 2.53 = 4554.00 \text{ руб.}$$

В таблице 3.4 приведены расходы на материалы после внедрения подсистемы.

Таблица 3.4 – Расходы на материалы после внедрения подсистемы

Материалы	Ед. измер.	Кол-во	Цена за ед. (руб.)	Стоимость (руб.)
Бумага	шт.	200	2	400
Канцтовары	-		-	70
Итого				470

Расходы на материалы за девять месяцев составили:

$$P_{\text{м}} = 4230.00 \text{ руб.}$$

$$P_{\text{общ}} + P_{\text{пр}} = 4554.00 + 4230.00 = 8784.00 \text{ руб.}$$

$$P_{\text{пр}} = 0.60 * 8784.00 = 5270.40 \text{ руб.}$$

$$P_{\text{общ}} = 8784.00 + 5270.40 = 14054.40 \text{ руб.}$$

После разработки и внедрения подсистемы остались работать программист и системный администратор. Получаем, что расходы за расчетный период (9 месяцев) после разработки и внедрения подсистемы составили:

$$P_{\text{осн}} = (10000 + 10000) * 9 * 0.30 = 54000.00 \text{ руб.}$$

$$P_{\text{ауп}} = 0.25 * 54000.00 = 13500.00 \text{ руб.}$$

Таким образом, эксплуатационные расходы на решение задачи после внедрения системы составляют:

$$P_1 = 54000.00 + 13500.00 + 14054.40 = 81554.40 \text{ руб.}$$

Далее находим приведенные затраты:

$$З = 81554.40 + 0.25 * 27000.00 = 88304.40$$

Условный экономический эффект:

$$130881.40 - 81554.40 = 49327.00 \text{ руб.}$$

Срок окупаемости:

$$T_p = 27000.00 / 49327.00 = 0.55$$

Расчетный коэффициент:

$$E_p = 49327.00 / 27000.00 = 1.82$$

Результаты расчетов представлены в таблице 3.5.

Таблица 3.5 – Результаты расчетов

№ п/п	Наименование показателей	Условные обозначения	Значение показателей
1)	Приведенные затраты	руб.	88304.40
2)	Расходы до внедрения	руб.	130881.40
3)	Расходы после внедрения	руб.	81554.40
4)	Экономический эффект	руб.	49327.00
5)	Срок окупаемости	мес.	6
6)	Расчетный коэффициент	%	1.83

В результате расчетов, можно заметить, что произошло сокращение затрат. При внедрении подсистемы сократилось число сотрудников, поэтому снизились расходы на заработную плату и общепроизводственные расходы. Срок окупаемости данной разработки составляет 6 месяцев. Расчетный коэффициент  $E_p \leq E_p = 1.7 \leq 1.83$ , это доказывает целесообразность разработки и внедрения подсистемы.

В третьем разделе выпускной квалификационной работы была разработана база данных, разработан и запрограммирован пользовательский интерфейс подсистемы учета заказов в кафе «Тайм». Выполнен и описан контрольный пример реализации проекта. Рассчитана экономическая эффективность разрабатываемой подсистемы учета заказов.

## ЗАКЛЮЧЕНИЕ

В результате выполнения выпускной квалификационной работы была разработана автоматизированная подсистема учета заказов на примере кафе «Тайм». Использование данной подсистемы способствует повышению эффективности деятельности персонала, занимающегося обслуживанием клиентов в кафе «Тайм», следовательно, цель выпускной квалификационной работы выполнена.

В первом разделе выпускной квалификационной работы была проанализирована технико-экономическая характеристика предприятия, выделены конкретные задачи и цели создания автоматизированной подсистемы учета заказов. Далее была проанализирована предметная область, на основе этого построены структурно-функциональные диаграммы организации бизнеса «Как есть» и «Как должно быть», выявлены недостатки работы предприятия. Проведен анализ имеющихся систем автоматизации предприятий общественного питания и выявлены наилучшие и наихудшие из рассматриваемых систем.

Во втором разделе выпускной квалификационной работы были обоснованы проектные решения по техническому, информационному, программному и технологическому обеспечению. Были выбраны программные средства, используемые при разработке подсистемы учета заказов. Также было создано информационное обеспечение задачи: спроектирован и описан логический, и физический уровень базы данных

В третьем разделе выпускной квалификационной работы была разработана база данных, разработан и запрограммирован пользовательский интерфейс подсистемы учета заказов в кафе «Тайм». Выполнен и описан контрольный пример реализации проекта. Рассчитана экономическая эффективность разрабатываемой подсистемы учета заказов.

Таким образом, в выпускной квалификационной работе были решены все поставленные задачи:

- изучена необходимая литература по данной теме;
- проанализирована технико-экономическая характеристика предприятия;
- разработаны функциональные модели деятельности предприятия «Как есть» и «Как должно быть»;
- разработана логическая и физическая модель базы данных;
- разработана структура базы данных;
- разработана автоматизированная подсистема учета заказов на примере кафе «Тайм»;
- выполнен контрольный пример реализации разработанной подсистемы;
- рассчитана экономическая эффективность разработанной подсистемы учета заказов.

Разработанная подсистема позволяет хранить большие объёмы информации, модифицировать, удалять и добавлять данные в подсистему, позволяет выполнять различные запросы на поиск и сортировку данных, создавать отчеты, в том числе и аналитические, также были установлены права доступа сотрудникам, для обеспечения защиты от несанкционированного доступа к системе.

Расчет экономической эффективности показал, что срок окупаемости разработанной подсистемы равен шесть месяцев. Расчетный коэффициент равен 1.83 – это больше 1.7, что доказывает целесообразность разработки и внедрения подсистемы

Разработанная подсистема будет введена в эксплуатацию в кафе «Тайм», а в дальнейшем планируется разработать полноценную систему автоматизации предприятия общественного питания.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем [Текст] / А.М. Вендров. – Москва: Вильямс, 2008. – 298 с.
- 2) Груббер, М. Введение в язык SQL [Текст] / М. Груббер. – Москва: ЛОРИ, 2012. – 644 с.
- 3) Гультаев, А.К. Проектирование и дизайн пользовательского интерфейса [Текст] / А.К. Гультаев, В.А. Машин. – СПб.: КОРОНАПринт, 2015. – 349 с.
- 4) Жданов, С.А. Информационные системы [Текст] / С.А. Жданов, А.С. Алфимова, М.Л. Соболева. – Москва: Прометей, 2015. – 304 с.
- 5) Иванова, М.Н. Проведение сравнительного анализа систем автоматизации предприятий общественного питания [Текст] / М.Н. Иванова, К.С. Шохина // Естественнонаучные, инженерные и экономические исследования в технике, промышленности, медицине и сельском хозяйстве: сб. статей. – Белгород, 2017 – с. 171-173.
- 6) Илюшечкин, В.М. Основы использования и проектирования баз данных [Текст] / В.М. Илюшечкин. – Москва: Юрайт, 2017. – 214 с.
- 7) Информационно управленческий портал [Электронный ресурс] Режим доступа: <http://portal-u.ru/informatsionnye-tekhnologii-upravleniya>
- 8) Карпова, Т.С. Базы данных: модели, разработка, реализация [Текст] / Т.С. Карпова. – Санкт-Петербург: Питер, 2011. – 304 с.
- 9) Кондратьев, К.П. Организация производства на предприятиях общественного питания [Текст] / К.П. Кондратьев. – Улан-Удэ: ВСГТУ, 2012. – 108 с.
- 10) Коршунов, Н.В. Организация обслуживания в ресторанах [Текст] / Н.В. Коршунов. – Москва: Высшая школа, 2010. – 223 с.

- 11) Култыгин, О.П. Администрирование баз данных СУБД MS SQL Server [Текст] / О.П. Култыгин. – Москва: Московская финансово-промышленная академия, 2012. – 232 с.
- 12) Маклаков, С.В. Моделирование бизнес-процессов с BPwin 4.0 [Текст] / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2012. – 224 с.
- 13) Маторин, С.И. Теория систем и системный анализ [Текст] / С.И. Маторин, О.А. Зимовец. - Белгород: ИД Белгород, 2012. – 154 с.
- 14) Михелев, В.М. Базы данных и СУБД [Текст] / В.М. Михелев. – Белгород: Издательство БелГУ, 2007. – 200 с.
- 15) Мрыхина, Е.Б. Организация производства на предприятиях общественного питания [Текст] / Е.Б. Мрыхина. – Москва: Инфра-М, 2017. – 416 с.
- 16) О работе с InterBase/Firebird. [Электронный ресурс] - Режим доступа: <http://www.ibase.ru/devinfo/ibrusfaq.htm>, свободный.
- 17) Павловская, Т.А. С/С++ Программирование на языке высокого уровня [Текст] / Т.А. Павловская. – Санкт-Петербург: Питер Спб, 2017. – 464 с.
- 18) Петров, В.Н. Информационные системы [Текст] / В.Н. Петров. – Санкт-Петербург: Питер, 2014. – 688 с.
- 19) Пирогов, В.Ю. Информационные системы и базы данных: организация и проектирование [Текст] / В.Ю. Пирогов. – Москва: Юрайт, 2014. – 529 с.
- 20) Построение инфологической модели [Электронный ресурс] - Режим доступа: <http://citforum.ru/database/dbguide/5-2.shtml>, свободный.
- 21) Радченко, Л.А. Организация производства и обслуживания на предприятиях общественного питания [Текст] / Л.А. Радченко. – Москва: КноРус, 2017. – 322 с.
- 22) Радченко, Л.А. Обслуживание на предприятиях общественного питания [Текст] / Л.А. Радченко. – Ростов-на-Дону: Феникс, 2013. – 320 с.
- 23) Романенко, И.В. Экономика предприятия [Текст] / И.В. Романенко. – Москва: Финансы и статистика, 2014. – 532 с.

- 24) Саттер, Г. Стандарты программирования на С++ [Текст] / Г. Саттер, А. Александреску. – Москва: Вильямс, 2015. – 224 с.
- 25) Сологубова, Г.С. Организация обслуживания на предприятиях общественного питания / Г.С. Сологубова. – Москва: Юрайт, 2016. – 58 с.
- 26) Страуструп, Б. Язык программирования С++. Стандарт С++ [Текст] / Б. Страуструп. – Москва: Бином, 2017. – 176 с.
- 27) Стружкин, Н.П. Базы данных: Проектирование [Текст] / Н.П. Стружкин, В.В. Годин. – Москва: Юрайт, 2017. – 292 с.
- 28) Тимофеев, В.В. С++ как он есть [Текст] / В.В. Тимофеев. – Москва: Бином, 2014. – 336 с.
- 29) Усов, В.В. Организация производства и обслуживания на предприятиях общественного питания [Текст] / В.В. Усов. – Москва: Академия, 2014. – 416с.
- 30) Усов, В.В. Организация обслуживания в ресторанах [Текст] / В.В. Усов. – Москва: Высшая школа, 2010 – 208 с.
- 31) Фуфаев, Э. В. Базы данных [Текст] / Э.В. Фуфаев, Д.Э. Фуфаев. – Москва: Академия, 2015. – 320 с.
- 32) Холингворт, Д. BorlandC++ Builder 6. Руководство разработчика [Текст] / Д. Холингворт, Б. Сворт. – Москва: Вильямс, 2014. – 976 с.
- 33) Центр управления финансами. Информационное обеспечение. [Электронный ресурс] – Режим доступа: <http://center-yf.ru/data/stat/Informacionnoe-obespechenie.php>
- 34) Чакон, С. Git для профессионального программиста [Текст] / С. Чакон, Б. Штрауб. – Санкт-Петербург: Питер, 2016. – 496 с.
- 35) Чекмарев, А. Microsoft Windows 7 для пользователей [Текст] / А. Чекмарев. – Санкт-Петербург: БХВ-Петербург, 2013. – 560 с.
- 36) Чистов, Д.В. Проектирование информационных систем. Учебник и практикум [Текст] / Д.В. Чистов, П.В. Мельников. – Москва: Юрайт, 2017, – 216 с.

37) Шелементьева, В.А. Сравнительный анализ систем [Электронный ресурс] Режим доступа: <https://www.scienceforum.ru/2017/2271/27257>, свободный.

38) Шень, А. Программирование: теоремы и задачи [Текст] / А. Шень. – Москва: МЦНМО, 2015. – 296 с.

39) Шугарева, В.Г. Программное обеспечение и операционные системы ПК [Текст] / В.Г. Шугарева. – Ростов-на-Дону: Феникс, 2013. -382 с.

40) Firebird. Руководство разработчика баз данных. [Электронный ресурс] – Режим доступа: <http://coollib.com/b/240145/read>, свободный.

41) Microsoft windows [Электронный ресурс] – Режим доступа: <https://support.microsoft.com/ru-ru/products/windows?os=windows-7>, свободный.

## ПРИЛОЖЕНИЕ А

### Структурно-функциональные модели деятельности предприятия «Как есть» и «Как должно быть»

На рисунке А.1 представлена декомпозиция контекстной диаграммы «Получение и запись данных», разработанная с помощью методологии IDEF3 – графическое моделирование, предназначенное для описания и документирования информационных потоков в системе. Данная диаграмма содержит следующие блоки работ: проверка данных, оповещение об ошибке, запись данных.

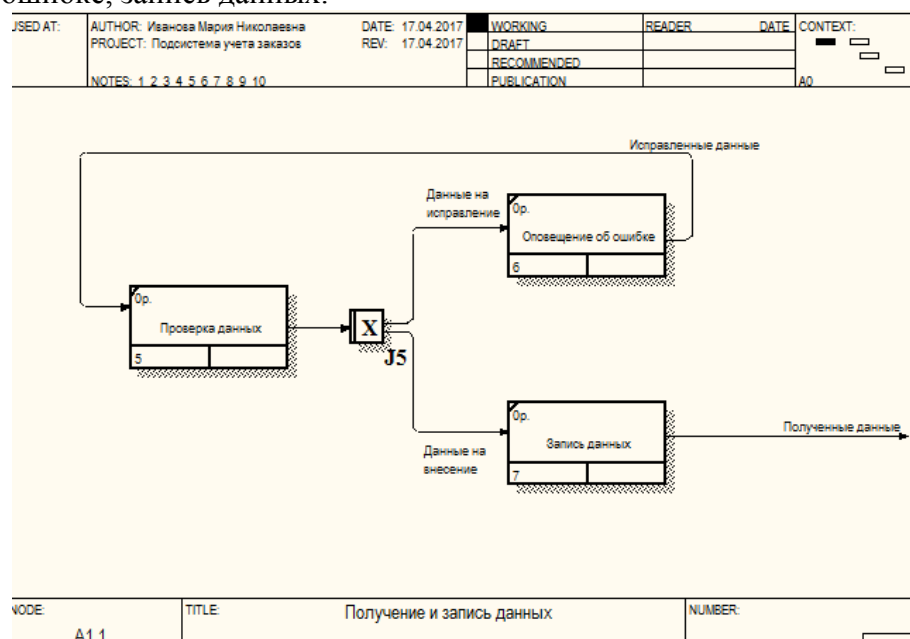


Рисунок А.1 – Декомпозиция контекстной диаграммы «Получение и запись данных»

На рисунке А.2 представлена декомпозиция контекстной диаграммы «Бронирование», разработанная с помощью методологии IDEF3. На данной диаграмме представлено как происходит бронирование столика в кафе, она содержит следующие блоки работ: выбор столика, проверка занятости и запись брони.

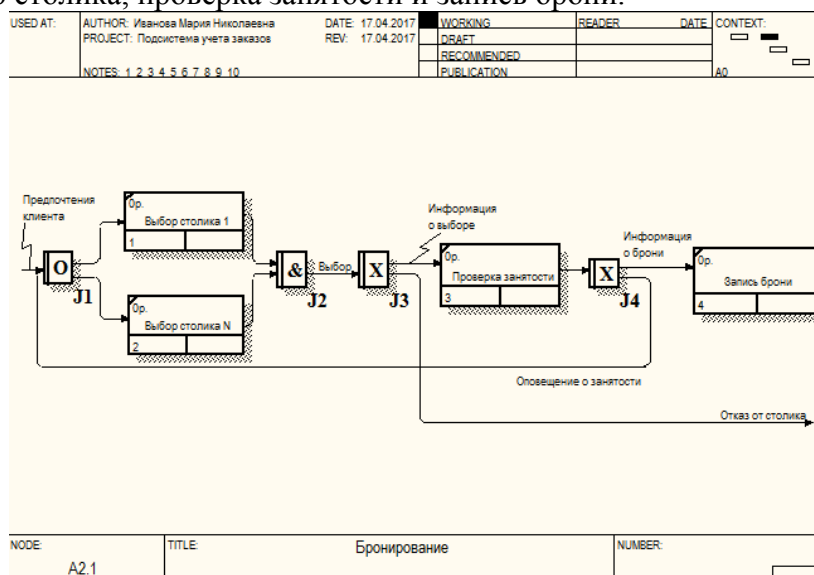


Рисунок А.2 – Декомпозиция контекстной диаграммы «Бронирование»

На рисунке А.3 представлена декомпозиция контекстной диаграммы «Заказ блюд и напитков», построена с помощью методологии IDEF0. на данной диаграмме представлены следующие блоки работ: выбор блюд и напитков, запись заказа, отправка клиенту.

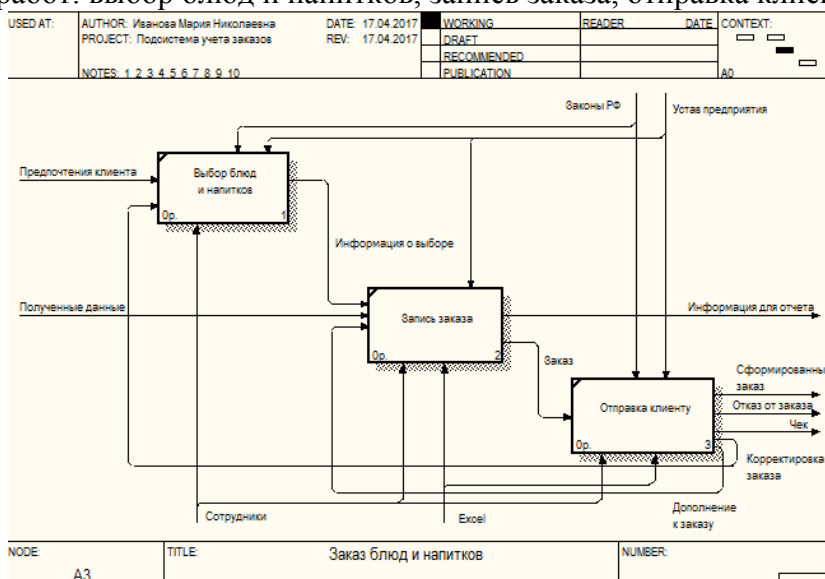


Рисунок А.3 – Декомпозиция контекстной диаграммы «Заказ блюд и напитков»

На рисунке А.4 представлена декомпозиция контекстной диаграммы «Составление отчетов», представленная в методологии DFD – диаграмма потоков данных, описывает процесс преобразования информации от её входа в систему до выдачи пользователю.

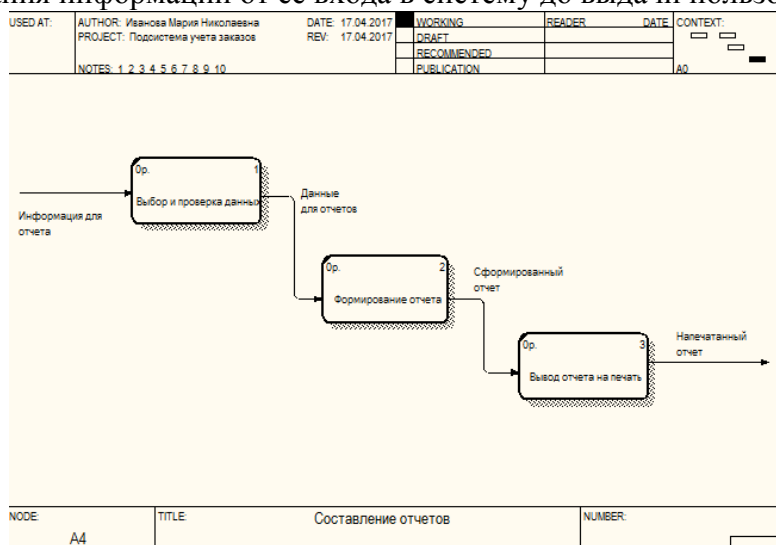


Рисунок А.4 – Декомпозиция контекстной диаграммы «Составление отчетов»

Для выявления присутствующих на предприятии недостатков была создана модель организации бизнеса «как должно быть».

Главным недостатком деятельности кафе «Тайм», является работа сотрудников без автоматизированных систем или подсистем. При использовании АИП работа предприятия будет наиболее эффективной, за счет создания базы данных, которая будет хранить в себе всю необходимую информацию и отчеты, а так же за счет сокращения времени работы сотрудников, так как при использовании подсистемы некоторые операции могут выполняться автоматически, например: создание отчетов.

На рисунке А.5 представлена декомпозиция контекстной диаграммы с внесенными изменениями.

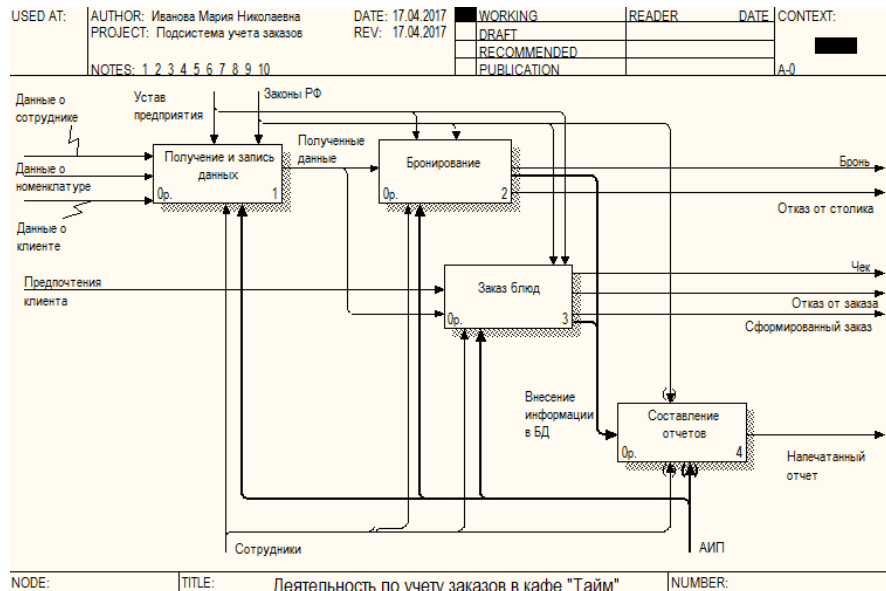


Рисунок А.5 – Декомпозиция контекстной диаграммы

На рисунке А.6 представлена декомпозиция контекстной диаграммы «как должно быть» «Составление отчетов». На данной диаграмме была добавлена база данных организации. Данная диаграмма позволяет посмотреть процесс обмена информацией между базой данных и АИП.

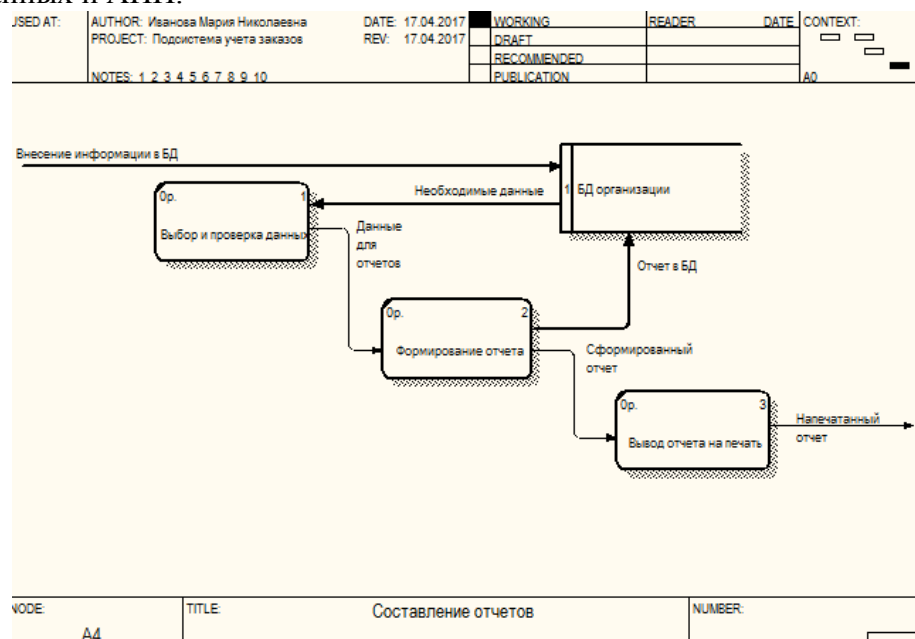


Рисунок А.6 - Декомпозиция контекстной диаграммы «Составление отчетов»

## ПРИЛОЖЕНИЕ Б

### Анализ существующих разработок

Для проведения анализа были выбраны наиболее популярные на рынке предприятий общественного питания системы:

1. R-Keererg на российском рынке впервые была представлена в 1992 году под брендом. В настоящее время считается самой популярной в России и в некоторых странах за рубежом. На 2016 год эта система была внедрена в 47 странах, более чем в 42000 ресторанах. R-Keererg – система автоматизации предприятий общественного питания: кафе, ресторана, бары, столовой.

Главная её особенность – это модульность, то есть заказчик может приобрести необходимый модуль в отдельности, соответствующий его стратегии в ведении бизнеса, а также его предпочтениям, в итоге он не будет переплачивать за аппаратуру, которую он не собирался приобретать. Стоимость такой системы от 54000 рублей.

2. Система ПКО в 2015 году была введена в 10000 ресторане, на данный момент она обслуживает более 16000 заведений, в 28 странах мира. Главная особенность – возможность быстрого распределения заказов между свободными сотрудниками. До 4 раза в год выходит новая версия системы с обновленными возможностями и добавлением новых функций. Существуют прикладное решение как для небольших заведений, так и для сети ресторанов, для фастфуда, доставки, автоматизации столовой и другие. Средняя стоимость такой системы от 54000 рублей.

3. 1С:Ресторан – разработчиком данного программного продукта автоматизации процессов обслуживания клиентов на предприятиях общественного питания является компания 1С-Парус. Данная система может использоваться как с бэк-офисом, так и без бэк-офиса. Главной особенностью является платформа 1С:Предприятие 8.2, что позволит легко администрировать систему и дорабатывать её под особенности своего предприятия. По последним данным эта система обслуживается в более чем 12500 тыс. заведений. Цена программного продукта от 40000 рублей, также можно заказать демо-версию бесплатно.

4. Система Трактирь, её разработчиком является компания СофтБаланс, автоматизация ресторанов, кафе и баров с помощью данной системы началась с 1998 года, на сегодняшний день внедрено более 10000 экземпляров данной системы. Отдел разработки СофтБаланс обладает статусом 1С:Центр разработки, так же эта компания является официальным партнером 1С.

Особенностью является то, что для работы Трактирь:Nano требуется наличие платформы 1С:Предприятие 8.2., так как она не является самостоятельным программным продуктом. Стоимость данной разработки равна 30000 рублей.

5. Tillyrad XL на рынке автоматизации ресторанов 20 лет, используется в более чем 12000 ресторанов России, СНГ, Великобритании и Германии. Имеется эффективная система скидок, программы лояльности, полностью настроенный под конкретное предприятие интерфейс, возможность работы с депозитами и предоплатой и другое. Цена данной системы составляет в около 27500 рублей.

6. Для автоматизации кафе также используется система Парус-Ресторан, её использует более 10000 заведений. Имеется возможность заказать презентацию данной системы бесплатно. Также на канале youtube можно просмотреть видео обзоры системы Парус-Ресторан. Эта система может использоваться как для полной автоматизации бизнес-процессов, так и для ведения учета отдельных отделов на предприятии. Её стоимость равна 30000 рублей.

7. Компания Quick Resto создает свои продукты нового поколения, предназначенная для автоматизации кафе. У них есть своя система автоматизации основанная на 4 шагах. Система Quick Resto также оснащена специальным приложением на iPhone, с помощью которого руководитель сможет в любой момент найти самую точную



информацию о ключевых показателях работы предприятия: клиенты, деньги, заказы и другое. На сегодняшний день внедрено около 4000 систем. Цена данной продукции равно 42000 рублей в год.

8. Smart Touch используется в более чем 5000 заведений, предназначена для автоматизации предприятий общественного питания. Имеется возможность принимать заказы и осуществлять предзаказы, с помощью специального бесплатного мобильного приложения BonusMe. Стоимость данной системы равна 20000 рублей в год.

9. Poster имеет облачную кассу для кафе, предназначена для автоматизации бизнес-процессов на предприятии. Данная система установлена в более чем 6000 заведений. За дополнительную плату можно заказать сайт-витрину для кафе или ресторана, которой будет автоматически импортировать данные из системы на сайт. Цена данного программного продукта равна около 32500 рублей в год.

10. Frontol – программное обеспечение так называемого класса front-office, используется для обеспечения автоматизации торговых залов предприятий розничной торговли и предприятий общественного питания. В настоящее время установлено около 4000 экземпляров данного ПО. Такие системы имеют понятный интерфейс, любая система лояльности настраивается с помощью удобного визуального конструктора, в котором необходимо задать только основные параметры: условия, действия и сроки акции. Его стоимость в среднем равна 16000 рублей.

Рассмотрев выбранные системы и выделив необходимые признаки, влияющие на выбор системы, была установлена градационная шкала.

Все позитивные признаки можно разбить на три градации, где:

- 1 – отсутствует данная возможность;
- 2 – частично присутствует;
- 3 – присутствует полноценно.

Негативные признаки так же можно разбить на три градации.

Стоимость:

- 1 – низкая стоимость системы – до 30000 рублей;
- 2 – средняя стоимость системы – от 30000 до 45000 рублей;
- 3 – высокая стоимость системы – от 45000 рублей.

Ущерб:

- 1 – не наносит ущерб предприятию;
- 2 – наносит незначительный ущерб;
- 3 – наносит большой ущерб.

Для каждого уровня шкалы градационного разбиения определены количественные значения отдельных признаков качества, с учетом назначенных коэффициентов весомости каждого признака.

Коэффициенты весомости для позитивных признаков:

1. Возможность графического представление плана зала ( $P_{\text{граф}}$ )= $k_{\text{граф}}=2$
2. Наличие системы бронирования столиков ( $P_{\text{брон}}$ )= $k_{\text{брон}}=3$
3. Возможность распечатки заказа ( $P_{\text{расп}}$ )= $k_{\text{расп}}=2$
4. Наличие программы лояльности ( $P_{\text{лоял}}$ )= $k_{\text{лоял}}=2$
5. Наличие отчетов ( $P_{\text{отч}}$ )= $k_{\text{отч}}=3$
6. Наличие простого и удобного интерфейса ( $P_{\text{интерф}}$ )= $k_{\text{интерф}}=1$
7. Возможность интеграции с системами ( $P_{\text{интегр}}$ )= $k_{\text{интегр}}=3$
8. Наличие разграничения прав доступа ( $P_{\text{доступ}}$ )= $k_{\text{доступ}}=5$

Коэффициенты весомости для негативных признаков:

1. Стоимость ( $P_{\text{стоим}}$ )= $k_{\text{стоим}}=5$
2. Ущерб ( $P_{\text{ущ}}$ )= $k_{\text{ущ}}=2$

Исходя, из полученных данных рассчитываются коэффициенты нормирования для каждого класса признаков:

Для положительных:

$$N^+ = 100 / \sum_{i=1}^n k_i^+ = 100 / (2+3+2+2+3+1+3+5) = 100 / 21 = 4,7619$$

Для отрицательных:

$$N^- = 100 / \sum_{j=1}^m k_j^- = 100 / (5+2) = 100 / 7 = 14,2857$$

Определяются ранги для каждого признака.

Для первого класса признаков:

1.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{2 * 4,7619\} = 9,5238$
2.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{3 * 4,7619\} = 14,2857$
3.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{2 * 4,7619\} = 9,5238$
4.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{2 * 4,7619\} = 9,5238$
5.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{3 * 4,7619\} = 14,2857$
6.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{1 * 4,7619\} = 4,7619$
7.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{3 * 4,7619\} = 14,2857$
8.  $r_i^+ = 66 \text{ntire} \{k_i^+ * N^+\} = 66 \text{ntire} \{5 * 4,7619\} = 23,8095$

Для второго класса признаков:

$$r_j^- = \text{entier} \{k_j^- * N^-\} = \text{entier} \{5 * 14,2857\} = 71,4285$$

$$2. r_j^- = \text{entier} \{k_j^- * N^-\} = \text{entier} \{2 * 14,2857\} = 28,5714$$

Округлив сумму рангов всех признаков в каждой группе, можно заметить, что она приравнивается к 100:

$$\sum_{i=1}^n r_i^+ = 9,5238 + 14,2857 + 9,5238 + 9,5238 + 14,2857 + 4,7619 + 14,2857 + 23,8095 = 100 -$$

верно

$$\sum_{j=1}^m r_j^- = 71,4285 + 28,5714 = 100 - \text{верно}$$

Рассчитывается шаг градации для каждого признака.

Для первого класса признаков:

$$H_{\text{граф}} = \text{entier} \{r_{\text{граф}} / (q_{\text{граф}} - 1)\} = \text{entier} \{9,5238 / (3 - 1)\} = 4,7619;$$

$$H_{\text{брон}} = \text{entier} \{r_{\text{брон}} / (q_{\text{брон}} - 1)\} = \text{entier} \{14,2857 / (3 - 1)\} = 7,1428;$$

$$H_{\text{расп}} = \text{entier} \{r_{\text{расп}} / (q_{\text{расп}} - 1)\} = \text{entier} \{9,5238 / (3 - 1)\} = 4,7619;$$

$$H_{\text{лоял}} = \text{entier} \{r_{\text{лоял}} / (q_{\text{лоял}} - 1)\} = \text{entier} \{9,5238 / (3 - 1)\} = 4,7619;$$

$$H_{\text{отч}} = \text{entier} \{r_{\text{отч}} / (q_{\text{отч}} - 1)\} = \text{entier} \{14,2857 / (3 - 1)\} = 7,1428;$$

$$H_{\text{интерф}} = \text{entier} \{r_{\text{интерф}} / (q_{\text{интерф}} - 1)\} = \text{entier} \{4,7619 / (3 - 1)\} = 2,3809;$$

$$H_{\text{интегр}} = \text{entier} \{r_{\text{интегр}} / (q_{\text{интегр}} - 1)\} = \text{entier} \{14,2857 / (3 - 1)\} = 7,1428;$$

$$H_{\text{доступ}} = \text{entier} \{r_{\text{доступ}} / (q_{\text{доступ}} - 1)\} = \text{entier} \{23,8095 / (3 - 1)\} = 11,9047;$$

Для второго класса признаков:

$$H_{\text{стоим}} = \text{entier} \{r_{\text{стоим}} / (q_{\text{стоим}} - 1)\} = \text{entier} \{71,4285 / (3 - 1)\} = 35,7142;$$

$$H_{\text{ущ}} = \text{entier} \{r_{\text{ущ}} / (q_{\text{ущ}} - 1)\} = \text{entier} \{28,5714 / (3 - 1)\} = 14,2857.$$

Рассчитываются значения уровней градации для каждого признака.

Для первого класса признаков:

$$P_{\Pi}(1) = (s-1) * h_{\Pi}$$

$$P_{\text{граф}}(1) = 0 * 4,7619 = 0$$

$$P_{\text{граф}}(2) = 1 * 4,7619 = 4,7619$$

$$P_{\text{граф}}(3) = 2 * 4,7619 = 9,5238$$

$$P_{\text{брон}}(1) = 0 * 7,1428 = 0$$

$$P_{\text{брон}}(2) = 1 * 7,1428 = 7,1428$$

$$P_{\text{брон}}(3) = 2 * 7,1428 = 14,2857$$

$$P_{\text{расп}}(1) = 0 * 4,7619 = 0$$

$$P_{\text{расп}}(2) = 1 * 4,7619 = 4,7619$$

$$P_{\text{расп}}(3) = 2 * 4,7619 = 9,5238$$

$$P_{\text{лоял}}(1) = 0 * 4,7619 = 0$$

$$P_{\text{лоял}}(2) = 1 * 4,7619 = 4,7619$$

$$P_{\text{лоял}}(3) = 2 * 4,7619 = 9,5238$$

$$P_{\text{отч}}(1) = 0 * 7,1428 = 0$$

$$P_{\text{отч}}(2) = 1 * 7,1428 = 7,1428$$

$$P_{\text{отч}}(3) = 2 * 7,1428 = 14,2857$$

$$P_{\text{интерф}}(1) = 0 * 2,3809 = 0$$

$$P_{\text{И}}(2) = 1 * 2,3809 = 2,3809$$

$$P_{\text{И}}(3) = 2 * 2,3809 = 4,7619$$

$$P_{\text{интегр}}(1) = 0 * 7,1428 = 0$$

$$P_{\text{интегр}}(2) = 1 * 7,1428 = 7,1428$$

$$P_{\text{интегр}}(3) = 2 * 7,1428 = 14,2857$$

$$P_{\text{доступ}}(1) = 0 * 11,9047 = 0$$

$$P_{\text{доступ}}(2) = 1 * 11,9047 = 11,9047$$

$$P_{\text{доступ}}(3) = 2 * 11,9047 = 23,8095$$

Для второго класса признаков:

$$P_{\text{стоим}}(1) = 0 * 35,7142 = 0$$

$$P_{\text{стоим}}(2) = 1 * 35,7142 = 35,7142$$

$$P_{\text{стоим}}(3) = 2 * 35,7142 = 71,4285$$

$$P_{\text{ущ}}(1) = 0 * 14,2857 = 0$$

$$P_{\text{ущ}}(2) = 1 * 14,2857 = 14,2857$$

$$P_{\text{ущ}}(3) = 2 * 14,2857 = 28,5714$$

Таблица Б.1 – Оценка системы: R-Кеерер

Признак	Возможность графического представление плана зала			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	9,5238

Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	14,2857

Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	9,5238

Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	9,5238

Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	14,2857

Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	4,7619

Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	14,2857

Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	23,8095

<b>Общая суммарная оценка позитивных свойств <math>\Sigma P^+(s)</math> в %</b>	<b>99,9999</b>
---	----------------

Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	71,4285

Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	0

<b>Общая суммарная оценка негативных свойств <math>\Sigma P^-(s)</math> в %</b>	<b>71,4285</b>
---	----------------

<b>Общая суммарная оценка эффективности использования Информационной системы <math>\Theta = \Sigma P^+(s) - \Sigma P^-(s)</math> в %</b>	<b>28,5714</b>
--	----------------

Таблица Б.2 – Оценка системы: ПКО

Признак	Возможность графического представление плана зала			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>14,2857</b>

Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>7,1428</b>

Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	<b>4,7619</b>

Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>14,2857</b>

Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	<b>23,8095</b>

<b>Общая суммарная оценка позитивных свойств <math>\Sigma P^+(s)</math> в %</b>	<b>92,857</b>
---	---------------

Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	<b>71,4285</b>

Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	<b>0</b>

<b>Общая суммарная оценка негативных свойств <math>\Sigma P^-(s)</math> в %</b>	<b>71,4285</b>
---	----------------

<b>Общая суммарная оценка эффективности использования Информационной системы <math>\Theta = \Sigma P^+(s) - \Sigma P^-(s)</math> в %</b>	<b>21,4285</b>
--	----------------

Таблица Б.3 – Оценка системы: 1С-Ресторан

Признак	Возможность графического представление плана зала			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>14,2857</b>

Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	<b>9,5238</b>

Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>7,1428</b>

Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	<b>2,3809</b>

Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	<b>14,2857</b>

Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	<b>23,8095</b>

<b>Общая суммарная оценка позитивных свойств <math>\Sigma P^+(s)</math> в %</b>	<b>90,476</b>
---	---------------

Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	<b>35,7142</b>

Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	<b>0</b>

<b>Общая суммарная оценка негативных свойств <math>\Sigma P^-(s)</math> в %</b>	<b>71,4285</b>
---	----------------

<b>Общая суммарная оценка эффективности использования Информационной системы <math>\Theta = \Sigma P^+(s) - \Sigma P^-(s)</math> в %</b>	<b>54,7618</b>
--	----------------

Таблица Б.4 – Оценка системы: Трактирь

Признак	Возможность графического представление плана зала			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	9,5238

Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	14,2857

Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	0

Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	0

Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	7,1428

Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	2,3809

Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	0

Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	23,8095

<b>Общая суммарная оценка позитивных свойств <math>\Sigma P^+_i(s)</math> в %</b>	<b>57,1427</b>
---	----------------

Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	35,7142

Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	0

<b>Общая суммарная оценка негативных свойств <math>\Sigma P^-_j(s)</math> в %</b>	<b>35,7142</b>
---	----------------

<b>Общая суммарная оценка эффективности использования Информационной системы <math>\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)</math> в %</b>	<b>21,4285</b>
--	----------------

Таблица Б.5 – Оценка системы: Frontol

Признак	Возможность графического представление плана зала			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	0

Признак	Наличие системы бронирования столиков			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	0

Признак	Возможность распечатки заказа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	4,7619

Признак	Наличие программы лояльности			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	4,7619	9,5238	0

Признак	Наличие отчетов			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	0

Признак	Наличие простого и удобного интерфейса			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	2,3809	4,7619	2,3809

Признак	Возможность интеграции с системами			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	7,1428	14,2857	0

Признак	Наличие разграничения прав доступа			Оценка, %
Градация	Отсутствует данная возможность	Частично присутствует	Присутствует полностью	
Значение	0	11,9047	23,8095	23,8095

<b>Общая суммарная оценка позитивных свойств <math>\Sigma P^+_i(s)</math> в %</b>	<b>30,9543</b>
---	----------------

Признак	Стоимость			Оценка, %
Градация	Низкая стоимость	Средняя стоимость	Высокая стоимость	
Значение	0	35,7142	71,4285	0

Признак	Ущерб			Оценка, %
Градация	Не наносит ущерба	Наносит незначительно	Наносит большой ущерб	
Значение	0	14,2857	28,5714	14,2857

<b>Общая суммарная оценка негативных свойств <math>\Sigma P^-_j(s)</math> в %</b>	<b>14,2857</b>
---	----------------

<b>Общая суммарная оценка эффективности использования Информационной системы <math>\Theta = \Sigma P^+_i(s) - \Sigma P^-_j(s)</math> в %</b>	<b>16,6686</b>
--	----------------

## ПРИЛОЖЕНИЕ В

### Программный код

```
CREATE TABLE CLIENTS (
  ID_CLIENT D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  ID_PEOPLE D_ID /* D_ID = INTEGER NOT NULL */
);
CREATE GENERATOR GENERATOR_ID_EMPL;
CREATE TABLE EMPLOYEES (
  ID_EMPLOYEES D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  ID_PEOPLE D_ID /* D_ID = INTEGER NOT NULL */,
  ID_POST D_ID /* D_ID = INTEGER NOT NULL */,
  PASSPORT D_PHONE /* D_PHONE = CHAR(11) */
);
CREATE GENERATOR GENERATOR_ID_FOOD;
CREATE TABLE FOOD (
  ID_FOOD D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  FOOD D_NAME /* D_NAME = CHAR(25) */,
  ID_TYPES D_ID /* D_ID = INTEGER NOT NULL */,
  PRICE D_PRICE /* D_PRICE = FLOAT */,
  COOKING_TIME D_TIME /* D_TIME = TIME */,
  WEIGHT D_ID /* D_ID = INTEGER NOT NULL */
);
CREATE GENERATOR GENERATOR_ID_ORDERS;
CREATE TABLE ORDERS (
  ID_ORDERS D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  ID_EMPLOYEEER D_ID /* D_ID = INTEGER NOT NULL */,
  ID_CLIENT D_ID /* D_ID = INTEGER NOT NULL */,
  TYPE_OF_ORDERS D_NAME NOT NULL /* D_NAME = CHAR(25) */,
  SUM_TEMP D_PRICE /* D_PRICE = FLOAT */,
  SALE D_ID_NOB /* D_ID_NOB = SMALLINT */,
  SUM_OR D_PRICE /* D_PRICE = FLOAT */,
  DELIVERY D_DELIV /* D_DELIV = CHAR(3) */,
  ADRESS_DEL D_NAME /* D_NAME = CHAR(25) */,
  STATUS D_NAME NOT NULL /* D_NAME = CHAR(25) */,
  DATE_TIME D_DATE_TIME /* D_DATE_TIME = TIMESTAMP NOT NULL */,
  ID_TABLE D_ID_NOB /* D_ID_NOB = SMALLINT */
);
CREATE TABLE ORDERS_OF_TABLES (
  ID_ORDERS D_ID /* D_ID = INTEGER NOT NULL */,
  ID_TABLE D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  DATA_TIME_VISIT D_DATE_TIME /* D_DATE_TIME = TIMESTAMP NOT NULL */
);
CREATE TABLE ORDER_FOOD (
  ID_ORDER D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  ID_FOOD D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  COUNT_FOOD D_ID /* D_ID = INTEGER NOT NULL */
);
CREATE TABLE ORDER_POTABLES (
  ID_ORDERS D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  ID_POTABLES D_ID /* D_ID = INTEGER NOT NULL */,
  COUNT_POT D_ID /* D_ID = INTEGER NOT NULL */
);
CREATE GENERATOR GENERATOR_ID_PEOPLE;
CREATE VIEW COUNT_EMP_ORDER(
  SURNAME,
  COUNT_ALL)
AS
select VIEW_ORDERS_SUM_3.surname_emp, count(VIEW_ORDERS_SUM_3.id_orders) from
VIEW_ORDERS_SUM_3 group by VIEW_ORDERS_SUM_3.surname_emp
;
```



```

;
CREATE GENERATOR GENERATOR_ID_POT;
CREATE TABLE POTABLES (
  ID_POTABLES D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  POTABLE     D_NAME /* D_NAME = CHAR(25) */,
  ID_TYPES   D_ID /* D_ID = INTEGER NOT NULL */,
  PRICE      D_PRICE /* D_PRICE = FLOAT */,
  VOLUME     D_ID /* D_ID = INTEGER NOT NULL */
);
CREATE TABLE PEOPLE (
  ID_PEOPLE   D_ID NOT NULL /* D_ID = INTEGER NOT NULL */,
  SURNAME     D_NAME /* D_NAME = CHAR(25) */,
  NAME        D_NAME /* D_NAME = CHAR(25) */,
  PATRONUMIC  D_NAME /* D_NAME = CHAR(25) */,
  PHONE       D_PHONE /* D_PHONE = CHAR(11) */,
  ADDRESS     D_NAME /* D_NAME = CHAR(25) */,
  DATE_OF_BIRTH D_DATE /* D_DATE = DATE */
);
CREATE VIEW CREATE_ORDERS(
  ID_ORDERS,
  SURNAME_EMP,
  SURNAME_CL,
  TYPE_OF_ORDERS,
  STATUS,
  DATE_TIME,
  DELIVERY,
  ADRESS_DEL)
AS
select id_orders, surname_emp, surname_cl, type_of_orders, status, date_time, DELIVERY, ADRESS_DEL from
view_orders
;
CREATE VIEW VIEW_FOOD_PRICE_COUNT(
  ID_ORDER,
  ID_FOOD,
  COUNT_FOOD,
  PRICE)
AS
select DISTINCT order_food.id_order, order_food.id_food, order_food.count_food, price from orders,
order_food, orders_of_tables, food where
order_food.id_order = orders.id_orders and order_food.id_food = food.id_food
;
CREATE VIEW VIEW_EMPLOYEERS(
  ID_EMPLOYEES,
  SURNAME,
  NAME,
  PATRONUMIC,
  PHONE,
  ADDRESS,
  DATE_OF_BIRTH,
  POST,
  PASSPORT)
AS
select id_employees, surname, people.name, patronumic, phone, adress, date_of_birth, post.name, passport from
employees, people, post where
employees.id_people = people.id_people and employees.id_post=post.id_post
;
CREATE VIEW VIEW_SALE_SUM_ALL(
  ID_ORDERS,
  SALE,
  SUMM_ALL)
AS
select id_orders, sale, summ_all - summ_all * sale / 100 as summ_all
from view_sale_sum

```

```

;

#include <vcl.h>
#pragma hdrstop
#include "Unit3.h"
#include "Unit2.h"
#include "Unit1.h"
#include "Unit13.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit12.h"
#include "Unit9.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit11.h"
#include "File1.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
int flag;
void __fastcall TForm3::ComboBox1Change(TObject *Sender)
{
if (ComboBox1->ItemIndex == 0){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Петрова Ирина Васильевна");
    ComboBox2->Items->Add("Сидоров Илья Иванович");
}if (ComboBox1->ItemIndex == 1){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Семыкина Алина Ивановна");
    ComboBox2->Items->Add("Рябченко Елена Александровна");
    ComboBox2->Items->Add("Игнатов Егор Петрович");
    ComboBox2->Items->Add("Рыжов Андрей Владимирович");
    ComboBox2->Items->Add("Рябова Екатерина Владимировна");
}if (ComboBox1->ItemIndex == 2){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Иванова Мария Николаевна");
}if (ComboBox1->ItemIndex == 3){
    ComboBox2->Items->Clear();
    ComboBox2->Enabled = true;
    ComboBox2->Items->Add("Абрамова Арина Семеновна");
}
ComboBox2->Text="ФИО";
}
void __fastcall TForm3::ComboBox2Change(TObject *Sender)
{
Edit1->Enabled = true;
}else{
//ShowMessage("Ошибка, отсутствует введенный сотрудник! Выберите сотрудника из предложенного выпадающего списка");
//}
}
void __fastcall TForm3::BitBtn1Click(TObject *Sender)
{
if ((ComboBox1->ItemIndex==0)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0001")
|| (ComboBox1->ItemIndex==0)&&(ComboBox2->ItemIndex==1)&& (Edit1->Text == "0001")){
    Form3->Visible=false;
    Form1->Show();
    flag = 1; //если управляющий
}
}

```

```

}
else if ((ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0002")
|| (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==1)&& (Edit1->Text == "0002")
|| (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==2)&& (Edit1->Text == "0002")
|| (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==3)&& (Edit1->Text == "0002")
|| (ComboBox1->ItemIndex==1)&&(ComboBox2->ItemIndex==4)&& (Edit1->Text == "0002")){

    Form1->N2->Visible=false;
    Form3->Visible=false;
    Form1->Show();
    flag = 2; //если официант
}else if ((ComboBox1->ItemIndex==2)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0003")){
    Form3->Visible=false;
    Form1->Show();
    flag = 3; //если администратор
}else if ((ComboBox1->ItemIndex==3)&&(ComboBox2->ItemIndex==0)&& (Edit1->Text == "0004")){
    Form1->N1->Visible=false;
    Form1->N2->Visible=false;
    Form1->N3->Visible=false;
    Form1->N4->Visible=false;
    Form3->Visible=false;
    Form1->Show();
    flag = 4;
}else{
    ShowMessage("Ошибка,проверте выбранную должность, ФИО или пароль!");
}
}
void __fastcall TForm1::N12Click(TObject *Sender)
{

    Form3->Show();
    Form1->Visible=false;
}
void __fastcall TForm1::N1Click(TObject *Sender)
{

if (flag == 2){
    Form2->N2->Visible=false;
}
Form2->N1->Visible=false;
Form2->Show();
Form1->Visible=false;
}
void __fastcall TForm1::N2Click(TObject *Sender)
{
if (flag == 2){
    Form4->N2->Visible=false;
}
Form4->Show();
Form1->Visible=false;
Form4->N2->Visible=false;
}
void __fastcall TForm1::FormClose(TObject *Sender, TCloseAction &Action)
{
Form3->Close();
}
void __fastcall TForm1::N10Click(TObject *Sender)
{
if (flag == 2){
    Form12->N2->Visible=false;
}
Form12->Show();
Form1->Visible=false;
}

```

```

Form12->N10->Visible=false;
}
void __fastcall TForm1::N5Click(TObject *Sender)
{
if (flag == 2){
    Form5->N2->Visible=false;
}
Form5->Show();
Form1->Visible=false;
Form5->N5->Visible=false;
}
void __fastcall TForm1::N6Click(TObject *Sender)
{
if (flag == 2){
    Form6->N2->Visible=false;
}
Form6->Show();
Form1->Visible=false;
Form6->N6->Visible=false;
}
void __fastcall TForm1::N8Click(TObject *Sender)
{
if (flag == 2){
    Form8->N2->Visible=false;
}
Form8->Show();
Form1->Visible=false;
Form8->N8->Visible=false;
}
void __fastcall TForm1::N9Click(TObject *Sender)
{
if (flag == 2){
    Form9->N2->Visible=false;
}
Form9->Show();
Form1->Visible=false;
Form9->N9->Visible=false;
}
void __fastcall TForm1::N7Click(TObject *Sender)
{
if (flag == 2){
    Form10->N2->Visible=false;
}
Form10->Show();
Form1->Visible=false;
Form10->N7->Visible=false;
}
void __fastcall TForm1::N11Click(TObject *Sender)
{
if (flag == 2){
    Form11->N2->Visible=false;
}else if (flag == 4) {
Form11->N1->Visible=false;
Form11->N2->Visible=false;
Form11->N3->Visible=false;
Form11->N4->Visible=false;
Form11->N5->Visible=false;
Form11->N6->Visible=false;
Form11->N7->Visible=false;
Form11->N8->Visible=false;
Form11->N9->Visible=false;
Form11->N10->Visible=false;
}
}

```

```

}
Form11->Show();
Form1->Visible=false;
Form11->N11->Visible=false;
}
void __fastcall TForm1::N13Click(TObject *Sender)
{
if (flag == 2){
    Form7->N2->Visible=false;
}
Form7->Show();
Form1->Visible=false;
Form7->N13->Visible=false;
}
void __fastcall TForm2::FormClose(TObject *Sender, TCloseAction &Action)
{
Form3->Close();
}
void __fastcall TForm2::N2Click(TObject *Sender)
{
if (flag == 2){
    Form4->N2->Visible=false;
}
Form4->Show();
Form2->Visible=false;
Form4->N2->Visible=false;
}
void __fastcall TForm2::N5Click(TObject *Sender)
{
if (flag == 2){
    Form5->N2->Visible=false;
}
Form5->Show();
Form2->Visible=false;
Form5->N5->Visible=false;
}
void __fastcall TForm2::N6Click(TObject *Sender)
{
if (flag == 2){
    Form6->N2->Visible=false;
}
Form6->Show();
Form2->Visible=false;
Form6->N6->Visible=false;
}
void __fastcall TForm2::N10Click(TObject *Sender)
{
if (flag == 2){
    Form12->N2->Visible=false;
}
Form12->Show();
Form2->Visible=false;
Form12->N10->Visible=false;
}
void __fastcall TForm2::N7Click(TObject *Sender)
{
if (flag == 2){
    Form10->N2->Visible=false;
}
Form10->Show();
Form2->Visible=false;
Form10->N7->Visible=false;
}
}

```

```

void __fastcall TForm2::N8Click(TObject *Sender)
{
if (flag == 2){
    Form8->N2->Visible=false;
}
Form8->Show();
Form2->Visible=false;
Form8->N8->Visible=false;
}
void __fastcall TForm2::N9Click(TObject *Sender)
{
if (flag == 2){
    Form9->N2->Visible=false;
}
Form9->Show();
Form2->Visible=false;
Form9->N9->Visible=false;
}
void __fastcall TForm2::N11Click(TObject *Sender)
{
if (flag == 2){
    Form11->N2->Visible=false;
}
Form11->Show();
Form2->Visible=false;
Form11->N11->Visible=false;
}
void __fastcall TForm2::N12Click(TObject *Sender)
{
Form3->Show();
Form2->Visible=false;
}
void __fastcall TForm2::N1Click(TObject *Sender)
{
Form4->Show();
Form2->Visible=false;
Form4->N11->Visible=false;
}
void __fastcall TForm2::BitBtn3Click(TObject *Sender)
{
//C:\Program Files\Common Files\Borland Shared\Images\Buttons
DataModule13->IBQuery1->Close();
DataModule13->IBQuery1->SQL->Clear();
DataModule13->IBQuery1->SQL->Add("select * from view_client");
DataModule13->IBQuery1->Open();
ComboBox1->ItemIndex= -1;
ComboBox1->Text= "Выберите";
ComboBox2->ItemIndex= -1;
ComboBox2->Text= "Выберите";
ComboBox3->ItemIndex= -1;
ComboBox3->Text= "Выберите";
Edit7->Clear();
DateTimePicker1->Date = Now().DateString();
flag_one_up=false;
}
void __fastcall TForm2::ComboBox1Change(TObject *Sender)
{
ComboBox2->Enabled=true;
ComboBox3->ItemIndex=-1;
ComboBox3->Text= "Выберите";
ComboBox2->ItemIndex=-1;
ComboBox2->Text= "Выберите";
Edit7->Clear();
}

```

```

}
void __fastcall TForm2::ComboBox2Change(TObject *Sender)
{
AnsiString d;
switch (ComboBox2->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule13->IBQuery1->Close();
DataModule13->IBQuery1->SQL->Clear();
DataModule13->IBQuery1->SQL->Add("select * from view_client");
switch (ComboBox1->ItemIndex)
{case 0: d= "ORDER BY ID_CLIENT"+ d; break;
case 1: d= "ORDER BY SURNAME"+ d; break;
case 2: d= "ORDER BY NAME"+ d; break;
case 3: d= "ORDER BY PATRONUMIC"+ d; break;
case 4: d= "ORDER BY PHONE"+ d; break;
case 5: d= "ORDER BY ADRESS "+ d; break;
case 6: d= "ORDER BY DATE_OF_BIRTH"+ d;
}
DataModule13->IBQuery1->SQL->Add(d);
DataModule13->IBQuery1->Open();
}
void __fastcall TForm2::ComboBox3Change(TObject *Sender)
{
Edit7->Enabled=true;
ComboBox1->ItemIndex= -1;
ComboBox1->Text= "Выберите";
ComboBox2->ItemIndex= -1;
ComboBox2->Text= "Выберите";
}
void __fastcall TForm2::Button1Click(TObject *Sender)
{
DataModule13->IBQuery1->Close();
DataModule13->IBQuery1->SQL->Clear();

switch (ComboBox3->ItemIndex)
{
case 0:
DataModule13->IBQuery1->SQL->Add("select * from view_client where ID_CLIENT like '%" + Edit7->Text
+ "%' ");
break;
case 1:
DataModule13->IBQuery1->SQL->Add("select * from view_client where SURNAME like '%" + Edit7->Text
+ "%' ");
break;
case 2:
DataModule13->IBQuery1->SQL->Add("select * from view_client where NAME like '%" + Edit7->Text +
"%' ");
break;
case 3:
DataModule13->IBQuery1->SQL->Add("select * from view_client where PATRONUMIC like '%" + Edit7-
>Text + "%' ");
break;
case 4:
DataModule13->IBQuery1->SQL->Add("select * from view_client where PHONE like '%" + Edit7->Text +
"%' ");
break;
case 5:
DataModule13->IBQuery1->SQL->Add("select * from view_client where ADRESS like '%" + Edit7->Text +
"%' ");
break;
case 6:

```

```

        DataModule13->IBQuery1->SQL->Add("select * from view_client where DATE_OF_BIRTH = " + Edit7-
>Text + " ");
        break;
    }
    DataModule13->IBQuery1->Open();
    if ((DataModule13->IBQuery1->RecordCount==0) || (Edit7->Text == ""))
    {
        ShowMessage("He найдено");
    }
}
void __fastcall TForm2::N13Click(TObject *Sender)
{
    if (flag == 2){
        Form7->N2->Visible=false;
    }
    Form7->Show();
    Form2->Visible=false;
    Form7->N13->Visible=false;
}
void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
    if ((Form2->Edit1->Text != "") && (Form2->Edit2->Text != "")&& (Form2->Edit3->Text != "")
&& (Form2->Edit4->Text != "") && (Form2->Edit5->Text != "")){
        DataModule13->IBStoredProc1->ParamByName ("SURNAME")->AsString=Edit1->Text;
        DataModule13->IBStoredProc1->ParamByName ("NAME")->AsString=Edit2->Text;
        DataModule13->IBStoredProc1->ParamByName ("PATRONUMIC")->AsString=Edit3->Text;
        DataModule13->IBStoredProc1->ParamByName ("PHONE")->AsString=Edit4->Text;
        DataModule13->IBStoredProc1->ParamByName ("ADDRESS")->AsString=Edit5->Text;
        DataModule13->IBStoredProc1->ParamByName ("DATE_OF_BIRTH")->AsString=DateTimePicker1->Date;
        DataModule13->IBStoredProc1->Prepare();
        DataModule13->IBStoredProc1->ExecProc();
        DataModule13->IBStoredProc2->Prepare();
        DataModule13->IBStoredProc2->ExecProc();
        DataModule13->IBQuery1->Close();
        DataModule13->IBQuery1->Open();
    }else{
        ShowMessage("He все поля заполнены данными!");
    }
}
void __fastcall TForm2::BitBtn2Click(TObject *Sender)
{
    DataModule13->IBStoredProc11->ParamByName ("id_client")->AsString=DBEdit1->Text;
    DataModule13->IBStoredProc11->Prepare();
    DataModule13->IBStoredProc11->ExecProc();
    DataModule13->IBQuery1->Close();
    DataModule13->IBQuery1->Open();
}
void __fastcall TForm2::BitBtn7Click(TObject *Sender)
{
    Edit1->Clear();
    Edit2->Clear();
    Edit3->Clear();
    Edit4->Clear();
    Edit5->Clear();
    Edit7->Clear();
}
void __fastcall TForm2::BitBtn4Click(TObject *Sender)
{
    if (flag_one_up == true){
        DataModule13->IBStoredProc20->ParamByName ("id_people")->AsString=DBEdit1->Text;
        if (Edit1->Text != ""){
            DataModule13->IBStoredProc20->ParamByName ("SURNAME")->AsString=Edit1->Text;
        }else{

```



```

        DataModule13->IBStoredProc20->ParamByName ("SURNAME")->AsString=DBEdit2->Text;
    }
    if (Edit2->Text != ""){
        DataModule13->IBStoredProc20->ParamByName ("NAME")->AsString=Edit2->Text;
    }else{
        DataModule13->IBStoredProc20->ParamByName ("NAME")->AsString=DBEdit3->Text;
    }
    if (Edit3->Text != ""){
        DataModule13->IBStoredProc20->ParamByName ("PATRONUMIC")->AsString=Edit3->Text;
    }else{
        DataModule13->IBStoredProc20->ParamByName ("PATRONUMIC")->AsString=DBEdit4->Text;
    }
    if (Edit4->Text != ""){
        DataModule13->IBStoredProc20->ParamByName ("PHONE")->AsString=Edit4->Text;
    }else{
        DataModule13->IBStoredProc20->ParamByName ("PHONE")->AsString=DBEdit5->Text;
    }
    if (Edit5->Text != ""){
        DataModule13->IBStoredProc20->ParamByName ("ADRESS")->AsString=Edit5->Text;
    }else{
        DataModule13->IBStoredProc20->ParamByName ("ADRESS")->AsString=DBEdit6->Text;
    }
    if (DateTimePicker1->Date != Now()){
        DataModule13->IBStoredProc20->ParamByName ("DATE_OF_BIRTH")->AsString=DateTimePicker1-
>Date;
    }else{
        DataModule13->IBStoredProc20->ParamByName ("DATE_OF_BIRTH")->AsString=DBEdit7->Text;
    }
    DataModule13->IBStoredProc20->Prepare();
    DataModule13->IBStoredProc20->ExecProc();
    DataModule13->IBQuery1->Close();
    DataModule13->IBQuery1->Open();
    DateTimePicker1->Date = Now().DateString();
}else if (flag_one_up == false){
    ShowMessage("Проверьте правильность ИЗМЕНЯЕМЫХ полей! После изменения восстановление
данных будет невозможно!");
    flag_one_up = true;
}
}
}
void __fastcall TForm2::FormCreate(TObject *Sender)
{
    DateTimePicker1->Date=Now();
}
void __fastcall TForm2::BitBtn5Click(TObject *Sender)
{
    Form15->QuickRep2->Preview();
}
void __fastcall TForm2::BitBtn6Click(TObject *Sender)
{
    Form15->QuickRep2->Print();
}
void __fastcall TForm5::N1Click(TObject *Sender)
{
    if (flag == 2){
        Form2->N2->Visible=false;
    }
    Form2->N1->Visible=false;
    Form2->Show();
    Form5->Visible=false;
}
void __fastcall TForm5::N2Click(TObject *Sender)
{
    if (flag == 2){

```

```

        Form4->N2->Visible=false;
    }
    Form4->Show();
    Form5->Visible=false;
    Form4->N2->Visible=false;
    }
    void __fastcall TForm5::N10Click(TObject *Sender)
    {
        if (flag == 2){
            Form12->N2->Visible=false;
        }
        Form12->Show();
        Form5->Visible=false;
        Form12->N10->Visible=false;
    }
    void __fastcall TForm5::N7Click(TObject *Sender)
    {
        if (flag == 2){
            Form10->N2->Visible=false;
        }
        Form10->Show();
        Form5->Visible=false;
        Form10->N7->Visible=false;
    }
    void __fastcall TForm5::N8Click(TObject *Sender)
    {
        if (flag == 2){
            Form8->N2->Visible=false;
        }
        Form8->Show();
        Form5->Visible=false;
        Form8->N8->Visible=false;
    }
    void __fastcall TForm5::N9Click(TObject *Sender)
    {
        if (flag == 2){
            Form9->N2->Visible=false;
        }
        Form9->Show();
        Form5->Visible=false;
        Form9->N9->Visible=false;
    }
    void __fastcall TForm5::N11Click(TObject *Sender)
    {
        if (flag == 2){
            Form11->N2->Visible=false;
        }
        Form11->Show();
        Form5->Visible=false;
        Form11->N11->Visible=false;
    }
    //-----
    void __fastcall TForm5::N12Click(TObject *Sender)
    {
        Form3->Show();
        Form5->Visible=false;
    }
    void __fastcall TForm5::FormClose(TObject *Sender, TCloseAction &Action)
    {
        Form3->Close();
    }
    void __fastcall TForm5::N6Click(TObject *Sender)
    {

```

```

if (flag == 2){
    Form6->N2->Visible=false;
}
Form6->Show();
Form5->Visible=false;
Form6->N6->Visible=false;
}
void __fastcall TForm5::N13Click(TObject *Sender)
{
if (flag == 2){
    Form7->N2->Visible=false;
}
Form7->Show();
Form5->Visible=false;
Form7->N13->Visible=false;
}
void __fastcall TForm5::DBEdit2Click(TObject *Sender)
{
Form14->Show();
Form14->TabSheet7->Show();
Form14->Width = 1440;
Form14->Height = 810;
}
void __fastcall TForm5::Edit1Click(TObject *Sender)
{
Form14->Show();
Form14->TabSheet7->Show();
Form14->Width = 980;
Form14->Height = 450;
}
void __fastcall TForm5::BitBtn1Click(TObject *Sender)
{
if (Form5->Edit1->Text != "") {
DataModule13->IBStoredProc9->ParamByName ("ID_ORDERS")->AsString=DBEdit1->Text;
DataModule13->IBStoredProc9->ParamByName ("ID_TABLE")->AsString=Edit1->Text;
DataModule13->IBStoredProc9->ParamByName ("DATA_TIME_VISIT")-
>AsString=DateToStr(DateTimePicker1->Date)+TimeToStr(DateTimePicker2->Time);
DataModule13->IBStoredProc9->Prepare();
DataModule13->IBStoredProc9->ExecProc();
DataModule13->IBQuery7->Close();
DataModule13->IBQuery7->Open();
DataModule13->IBQuery5->Close();
DataModule13->IBQuery5->Open();
}else{
    ShowMessage("Не все поля заполнены данными!");
}
}
void __fastcall TForm5::DBEdit1Click(TObject *Sender)
{
Form14->Show();
Form14->TabSheet8->Show();
}
void __fastcall TForm5::ComboBox2Change(TObject *Sender)
{
AnsiString d;
switch (ComboBox2->ItemIndex)
{case 0: d = " "; break;
case 1: d = " DESC";
}
DataModule13->IBQuery7->Close();
DataModule13->IBQuery7->SQL->Clear();
DataModule13->IBQuery7->SQL->Add("select * from ORDERS_OF_TABLES");
}

```

```

switch (ComboBox1->ItemIndex)
{case 0: d= "ORDER BY ID_ORDERS " + d; break;
case 1: d= "ORDER BY ID_TABLE " + d; break;
case 2: d= "ORDER BY DATA_TIME_VISIT " + d;
}
DataModule13->IBQuery7->SQL->Add(d);
DataModule13->IBQuery7->Open();
}
void __fastcall TForm5::ComboBox1Change(TObject *Sender)
{
ComboBox2->Enabled=true;
ComboBox3->ItemIndex=-1;
ComboBox3->Text= "Выберите";
ComboBox2->ItemIndex=-1;
ComboBox2->Text= "Выберите";
Edit7->Clear();
}
void __fastcall TForm5::ComboBox3Change(TObject *Sender)
{
Edit7->Enabled=true;
ComboBox1->ItemIndex= -1;
ComboBox1->Text= "Выберите";
ComboBox2->ItemIndex= -1;
ComboBox2->Text= "Выберите";
}
void __fastcall TForm5::Button1Click(TObject *Sender)
{
DataModule13->IBQuery7->Close();
DataModule13->IBQuery7->SQL->Clear();

switch (ComboBox3->ItemIndex)
{
case 0:
DataModule13->IBQuery7->SQL->Add("select * from ORDERS_OF_TABLES where ID_ORDERS like '%"
+ Edit7->Text + "%' ");
break;
case 1:
DataModule13->IBQuery7->SQL->Add("select * from ORDERS_OF_TABLES where ID_TABLE like '%" +
Edit7->Text + "%' ");
break;
case 2:
DataModule13->IBQuery7->SQL->Add("select * from ORDERS_OF_TABLES where DATA_TIME_VISIT
like '%" + Edit7->Text + "%' ");
break;
}
DataModule13->IBQuery7->Open();
if ((DataModule13->IBQuery7->RecordCount==0) || (Form5->Edit7->Text == ""))
{
ShowMessage("Не найдено");
}
}
void __fastcall TForm5::BitBtn3Click(TObject *Sender)
{
DataModule13->IBQuery7->Close();
DataModule13->IBQuery7->SQL->Clear();
DataModule13->IBQuery7->SQL->Add("select * from ORDERS_OF_TABLES");
DataModule13->IBQuery7->Open();
ComboBox1->ItemIndex= -1;
ComboBox1->Text= "Выберите";
ComboBox2->ItemIndex= -1;
ComboBox2->Text= "Выберите";
ComboBox3->ItemIndex= -1;

```

```

ComboBox3->Text= "Выберите";
Edit7->Clear();
DateTimePicker1->Color=clWindow;
DateTimePicker2->Color=clWindow;
flag_one_up=false;
}
void __fastcall TForm5::BitBtn2Click(TObject *Sender)
{
    DataModule13->IBStoredProc13->ParamByName ("id_orders")->AsString=DBEdit2->Text;
    DataModule13->IBStoredProc13->ParamByName ("id_table")->AsString=DBEdit3->Text;
    DataModule13->IBStoredProc13->ParamByName ("data_time_visit")->AsString=DBEdit4->Text;
        DataModule13->IBStoredProc13->Prepare();
        DataModule13->IBStoredProc13->ExecProc();
        DataModule13->IBQuery7->Close();
        DataModule13->IBQuery7->Open();
        DataModule13->IBQuery5->Close();
DataModule13->IBQuery5->Open();
}
void __fastcall TForm5::BitBtn7Click(TObject *Sender)
{
    Edit1->Clear();
    Edit7->Clear();
}
void __fastcall TForm5::BitBtn4Click(TObject *Sender)
{
    if (flag_one_up == true){
        DataModule13->IBStoredProc23->ParamByName ("ID_ORDERS")->AsString=DBEdit2->Text;
        DataModule13->IBStoredProc23->ParamByName ("ID_TABLE_from")->AsString=DBEdit3->Text;
        DataModule13->IBStoredProc23->ParamByName ("DATA_TIME_VISIT_from")->AsString=DBEdit4->Text;
        if (Edit1->Text!=""){
            DataModule13->IBStoredProc23->ParamByName ("ID_TABLE_to")->AsString=Edit1->Text;
        }else{
            DataModule13->IBStoredProc23->ParamByName ("ID_table_to")->AsString=DBEdit3->Text;
        }
        DataModule13->IBStoredProc23->ParamByName ("DATA_TIME_VISIT_to")->
        >AsString=DateToStr(DateTimePicker1->Date)+TimeToStr(DateTimePicker2->Time);
        DataModule13->IBStoredProc23->Prepare();
        DataModule13->IBStoredProc23->ExecProc();
        DataModule13->IBQuery7->Close();
        DataModule13->IBQuery7->Open();
        DataModule13->IBQuery5->Close();
        DataModule13->IBQuery5->Open();
        flag_one_up = false;
        DateTimePicker1->Color=clWindow;
        DateTimePicker2->Color=clWindow;
    }else{
        ShowMessage("Проверьте правильность ИЗМЕНЯЕМЫХ обязательных полей (дата и время визита)! После
        изменения восстановление данных будет невозможно!");
        flag_one_up = true;
        DateTimePicker1->Color=clMoneyGreen;
        DateTimePicker2->Color=clMoneyGreen;
    }
}
}

```

## ПРИЛОЖЕНИЕ Г

### Скриншоты работы подсистемы

Номер	Фамилия	Имя	Отчество	Телефон	Адрес	Дата рождения	Должность	Паспортный
2	Сидоров	Илья	Иванович	89514785962	г. Белгород, Кирова, 33	01.05.2017	Управляющий	3652 7894
3	Семькина	Алина	Ивановна	89625417896	г. Белгород, Попова, 45	01.01.1987	Официант	4563 1485
4	Рябенко	Елена	Александровна	89631452364	г. Белгород, Пушкина, 33	09.02.1991	Официант	4512 3697
5	Игнатов	Егор	Петрович	89641254789	г. Белгород, Королева, 23	10.02.1989	Официант	7856 1258
6	Рыжов	Андрей	Владимирович	89203654214	г. Белгород, Победы, 29	03.12.1989	Официант	7836 1278
7	Рябова	Екатерина	Владимировна	89512365478	г. Белгород, Щорса, 45	29.11.1993	Официант	3641 7892
1	Петрова	Ирина	Васильевна	89652063652	г. Белгород, ул. Щорса, 29	22.03.2017	Администратор БД	2123 1254
8	Абрамова	Арина	Семеновна	89642365417	г. Белгород, Королева, 21	15.02.1988	Администратор БД	5698 3698
9	Рудых	Инна	Андреевна	89635124789	г. Белгород, Победы, 2	12.02.1987	Администратор БД	6985 1437
10	Рябова	Екатерина	Владимировна	89512365478	г. Белгород, Щорса, 45	29.11.1993	Курьер	5698 1236

Рисунок Г.1 – Справочник «Сотрудники»

Номер	Фамилия сотрудника	Фамилия клиента	Вид заказа	Статус заказа	Дата	Доставка	Адрес доставки
1	Семькина	Межакова	Заказ блюд	выполнен	13.06.2016 11:13:27	да	Горького, 66
2	Семькина	Лазарева	Заказ блюд	принят	12.03.2016 17:16:00	нет	
3	Игнатов	Королева	Заказ блюд	принят	10.01.2017 12:05:03	да	Победы, 29
4	Рябенко	Рыжкова	Заказ блюд	приготовление	02.02.2017 15:35:00	нет	
5	Игнатов	Лазарева	Заказ блюд	выполнен	27.03.2017 10:15:22	нет	
6	Рябенко	Межакова	Заказ столов	принят	25.03.2017 21:20:59	нет	
7	Рыжов	Рыжкова	Заказ столов	выполнен	25.03.2017 17:44:37	нет	
8	Семькина	Рыжкова	Заказ блюд	принят	26.03.2017 10:22:00	нет	
10	Рыжов	Петрова	Заказ блюд	выполнен	10.05.2017 21:21:25	нет	
11	Семькина	null	Заказ блюд	принят	01.06.2017 11:54:34	нет	
12	Рябенко	null	Заказ блюд	принят	01.06.2017 12:54:34	нет	
13	Игнатов	Рязанов	Заказ блюд	выполнен	25.05.2017 23:30:35	да	Мира, 54

Рисунок Г.2 – Справочник «Заказы»

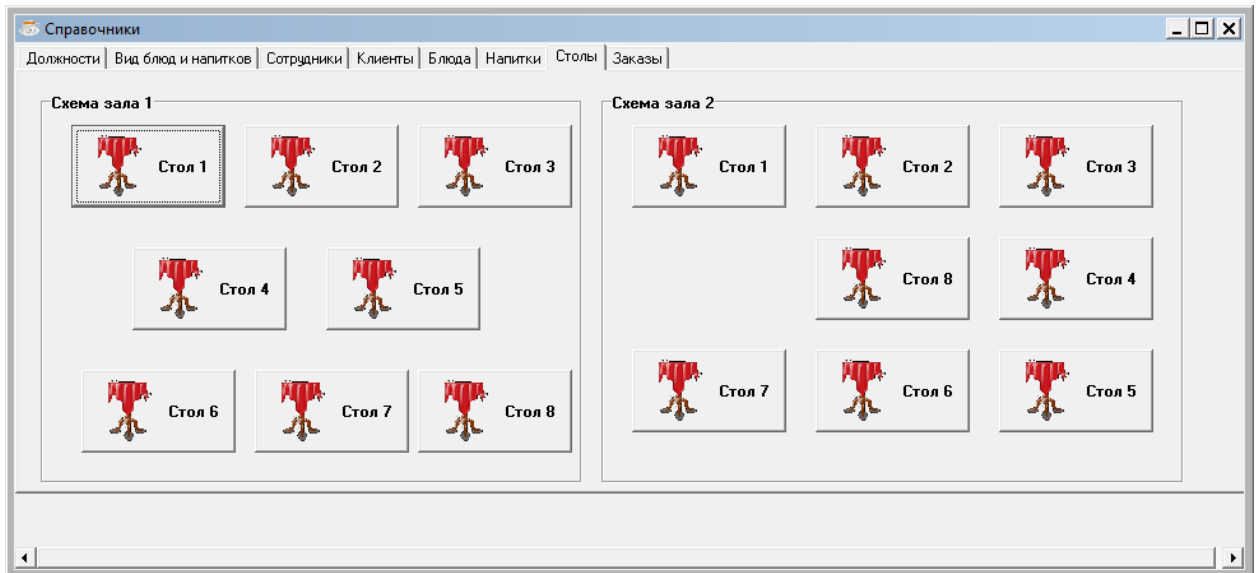


Рисунок Г.3 – Справочник «Стол»

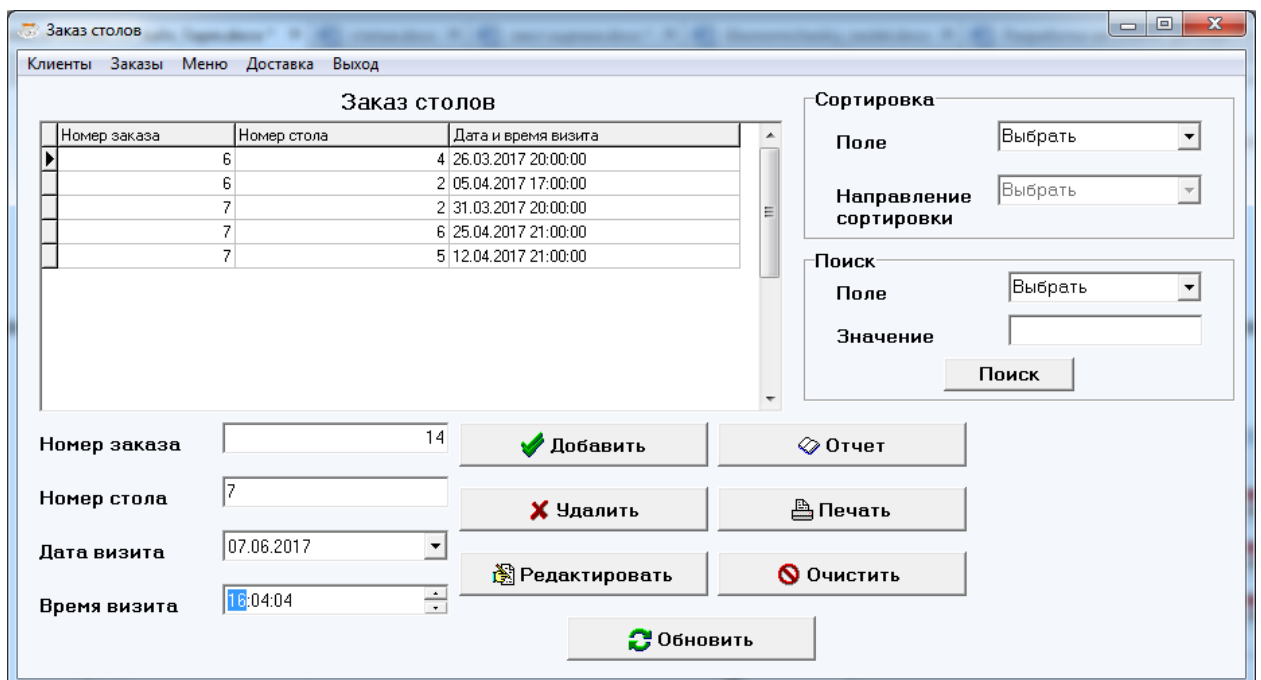


Рисунок Г.4 – Вкладка «Заказ столов»

Print Preview

Полный отчет о заказах

Номер заказа	Клиент	Сотрудник	Стоимость, р	Блюдо/напиток/стол	Количество/дата визита
1	Межакова	Семыкина	8329,5		
				Цезарь с курицей	2
				Банановый	2
				Кофе "Жардин"	3
				Виски "Джек Дениелс"	2

1200% Page 1 of 5

Рисунок Г.5 – Полный отчет о заказах

Print Preview

Клиенты

№	Фамилия	Имя	Отчество	Телефон	Адрес	Дата рождения
1	Лазарева	Елена	Эдуардовна	87415632147	г. Белгород, Горького, 55	08.05.1995
2	Королева	Елена	Ивановна	87456321114	г. Белгород, Советская, 25	13.06.1995
3	Межакова	Елена	Игоревна	87415632147	г. Белгород, Горького, 55	01.12.1987
4	Романенко	Елена	Алексеевна	86325874123	г. Белгород, Горького, 23	01.01.1987
5	Рыжкова	Елена	Александровна	85621788631	г. Белгород, Щорса, 15	25.01.1992
6	Сидоров	Елена	Иванович	89514785962	г. Белгород, Кирова, 33	01.05.2017
7	Семыкина	Елена	Ивановна	89625417896	г. Белгород, Попова, 45	01.01.1987
8	Петрова	Елена	Васильевна	89652063652	г. Белгород, ул. Щорса, 29	22.03.2017
9	Игнатов	Елена	Петрович	89641254789	г. Белгород, Королева, 23	10.02.1989
11	Рязанов	Елена	Федорович	89563214889	г. Белгород, Королева, 55	20.09.1971
15	Чащина	Елена	Сергеевна	86325412255	г. Белгород, Щорса, 16	05.02.1991

Page 1 of 1

Рисунок Г.6 – Отчет о клиентах



Print Preview

160% Page 1 of 1

Напитки				
№	Наименование напитка	Вид напитка	Цена, р	Объем
11	Ирландский ликер	Крепкие алкогольные	1500	700
12	Мартини	Крепкие алкогольные	2000	1000
13	Виски "Джек Дениелс"	Крепкие алкогольные	3500	1000
4	Вино "Изабелла"	Слабые алкогольные	1000	700
14	Пиво "Эфест"	Слабые алкогольные	120	500
15	Пиво "Клинское"	Слабые алкогольные	95	500
17	Пиво "Старый мельник"	Слабые алкогольные	120	500
1	Чай "Гринфилд" (асс)	Горячие	25	150
2	Кофе "Жардин"	Горячие	65	150
3	Кофе "Якобс"	Горячие	85	200
5	Вино "Шардоне"	Горячие	1200	750

Рисунок Г.7 – Отчет о напитках

Print Preview

233% Page 1 of 1

Блюда					
№	Наименование блюда	Вид блюда	Цена, р	Вес	Время приготовления
1	Французский поцелуй	Салаты	270,5	200	0:30:00
2	Мясное оливье	Салаты	150	200	0:25:00
3	Цезарь с курицей	Салаты	280	250	0:25:00
4	Коктейль	Салаты	120	150	0:10:00
5	Дамский каприз	Салаты	250	250	0:35:00
6	Ёка с зеленью	Закуски	350	350	0:45:00
7	Гратен	Закуски	300	300	0:30:00
8	Сырные крокеты	Закуски	270	250	0:30:00
9	Овощи по азиатски	Закуски	250	250	0:30:00
10	Фаршированные шампиньоны	Закуски	300	350	0:45:00
11	Борщ украинский	Первые блюда	150	200	0:15:00

Рисунок Г.8 – Отчет о блюдах

## ПРИЛОЖЕНИЕ Д

### Формулы расчета экономической эффективности

Формула расчета капитальных затрат:

$$K = K_{\text{гр}} + K_{\text{об}} + K_{\text{вс}} + K_{\text{тм}} + K_{\text{мг}}$$

Формула расчета собственных затрат:

$$З = P + E_n K$$

Эксплуатационные расходы на функционирование системы:

$$P = P_{\text{осн}} + P_{\text{ауп}} + P_{\text{общ}}$$

Общепроизводственные расходы:

$$P_{\text{общ}} = P_{\text{эн}} + P_{\text{м}} + P_{\text{рем}} + P_{\text{зап}} + P_{\text{кс}} + P_{\text{пр}}$$

Экономическая эффективность:

$$\mathcal{E} = P_0 - P_1$$

Срок окупаемости разработанной системы рассчитывается как отношение капитальных затрат к экономической эффективности:

$$T_p = K / \mathcal{E}$$

Обратная величина будет представлять расчетный коэффициент приведения:

$$E_p = \mathcal{E} / K$$

$E_p$  – расчетный коэффициент приведения.

# **ПРИЛОЖЕНИЕ Е**

## **Техническое задание**

### **1 Введение**

Подсистема учета заказов в кафе «Тайм» предназначена для: работы с внесенной в базу данных информацией, регистрации заказанных клиентами блюд или столиков, хранения информации, создания отчетов.

Данная подсистема позволит повысить эффективность деятельности персонала занимающегося обслуживанием клиентов в кафе «Тайм».

### **2 Основание для разработки**

Договор №768844 от 15.09.16 на разработку и внедрение подсистемы учета заказов.

### **3 Назначение разработки**

Автоматизированная подсистема учета предназначена для решения следующих задач:

- хранение и работа с информацией в базе данных о сотрудниках, занимающихся ведением учета, о клиентах, заказах, меню и столиках;
- выбор необходимого столика, блюда, напитка;
- разграничение прав доступа по должностям;
- оформление и ведение учета заказов столиков, напитков и блюд;
- составление отчетов: о заказах, о клиентах, о сотрудниках, о напитках и блюдах;
- составление графического отчета на выявление количества выполненных сотрудниками заказов и на выявление суммы выполненных заказов.

### **4 Требование к программному изделию**

#### **а. Требования к функциональным характеристикам**

Подсистема должна обеспечивать следующие функции:

- 1) Ввод, вывод, модификация, хранение, печать, информации о клиентах:

- ФИО;
- номер телефона;
- адрес;
- дата рождения.

2) Ввод, вывод, модификация, хранение, печать, информации о сотрудниках занимающихся заказами:

- ФИО;
- адрес;
- номер телефона;
- должность
- паспортные данные.

3) Ввод, вывод, модификация, хранение, печать, информации о заказах:

- тип заказа;
- скидка;
- сумма заказа;
- адрес доставки;
- статус доставки;
- дата и время.

4) Вход в подсистему осуществляется с помощью авторизации пользователем.

*Входная информация подсистемы:* регистрационная информация о заказах и о клиентах, предпочтения клиента.

*Выходная информация подсистемы:* отчеты о заказах, о клиентах, о сотрудниках, о напитках и блюдах, а также графические отчеты на выявление количества выполненных сотрудниками заказах и на выявление суммы выполненных заказов.

#### **в. Требования к надежности**

Данная подсистема должна:

- контролировать корректность входных данных;

– в случае возникновения ошибок, выдавать сообщение для пользователя.

### **с. Условия эксплуатации**

Подсистема может эксплуатироваться в помещениях, пригодных для постоянного наличия людей: температура 18-24 градуса С, влажность 60-80%. Интерфейс максимально понятен. Использовать подсистему будут люди различных квалификаций.

### **д. Требования к составу и параметрам технических средств**

Настоящая подсистема должна работать на ноутбуке MSI CX 61. Оперативная память на ноутбуке не должна быть меньше 5.89 Гб и достаточно 10 Гб свободного места на жестком диске.

### **е. Требования к информационной и программной совместимости**

Подсистеме необходимо работать под управлением операционной системы семейства Windows. Выбор другого программного обеспечения разработчиком основывается на низкой стоимости.

### **ф. Требования к маркировке и упаковке**

Разработанная подсистема поставляется на диске CD-R или CD-RW.

### **г. Требования к транспортированию и хранению**

Требования к транспортированию и хранению подсистемы аналогичны тем требованиям, которые предъявляются при хранении компакт-дисков.

## **5 Требования к программной документации**

Требования к программной документации не предъявляются.

Администратор

Кафе «Тайм»

\_\_\_\_\_

Мартынов Д.И.

МП

Выпускная квалификационная работа выполнена мной совершенно самостоятельно. Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

« » \_\_\_\_\_ 2017 г.

Студент \_\_\_\_\_ Иванова М.Н.