

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

К защите.

Зав. Кафедрой

_____ 201 г.

Выпускная квалификационная работа бакалавра

Мартыненко Анатолия

На тему:

РАЗРАБОТКА ИНФОРМАЦИОННОЙ СИСТЕМЫ «PROFI STAFF» -
РЕКРУТИНГОВОЕ АГЕНТСТВО

Выпускник _____
(подпись) (фамилия)

Руководитель _____
(подпись) (фамилия)

БЖД _____
(подпись) (фамилия)

Рецензент _____
(подпись) (фамилия)

ТАШКЕНТ-201

ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

Факультет ИИТ кафедра Информационные технологии.
Направление специальность) Информатика и информационные технологии

УТВЕРЖДАЮ

Зав кафедрой _____
« _____ » _____ 201 г.

ЗАДАНИЕ

на выпускную квалификационную работу

(фамилия, имя, отчество)

1. Тема работы Разработка информационной системы “Profi Staff” – рекрутинговое агентство
 2. Утверждена приказом по университету от « _____ » _____
 3. Срок сдачи законченной работы _____.
 4. Исходные данные к работе Литературные данные , входящие данные
 5. Содержание расчетно-пояснительной записи (перечень подлежащих разработке вопросов) Аннотация, Введение, Глава 1 – Анализ предметной области, Глава 2 – Проектирование информационной системы кадрового агентства, Глава 3 Реализация информационной системы кадрового агентства, Глава 4 – Безопасность жизнедеятельности, Заключение, Литература, Приложения
 6. Перечень графического материала Презентационные материалы
-
7. Дата выдачи задания _____

Руководитель _____ Задание принял _____
(подпись) (подпись)

8. Консультанты по отдельным разделам выпускной работы

Раздел	Ф.И.О руководителя	Подпись дата	
		Задание выдал	Задание получил
Основная часть			
Безопасность жизнедеятельности			

9. График выполнения работы

№	Наименование раздела работы	Срок выполнения	Отметка руководителя о выполнении
1	Введение		
2	Анализ предметной области		
3	Исследование информационной системы кадрового агентства		
4	Реализация информационной системы		
5	Безопасность жизнедеятельности		
6	Заключение		

Выпускник _____ « _____ » _____ 200__ г.
(подпись)

Руководитель _____ « _____ » _____ 200__ г.
(подпись)

АННОТАЦИЯ

В выпускной квалификационной работе проведён анализ деятельности кадровых агентств, и создана информационная система, выполняющая необходимые для этого процедуры. Также разработан дружественный интерфейс, облегчающий работу пользователя программы, создана база данных и созданы функции создания отчетов, которые осуществляют вывод данных на печать напрямую из информационной системы, функции импорта и экспорта в MS Word и MS Excel. Программа реализована на языке C# с использованием технологии ADO.NET в качестве инструмента взаимодействия с базой данных. В качестве СУБД в ВКР использовался Microsoft SQL Server 2005.

SUMMARY

In final qualifying work is organized analysis to activity trained agency, and created information system, and is created information system, executing required for this procedures. Is it Also designed friend interface, lightening functioning (working) the user of the program, is created database and are created functions of the making the reports, which realize the output given on print from information system stright, functions of the import and export in MS Word and MS Excel. The Program developed by with language C# and use of technologies ADO NET as instrument of the interaction with database. As DBMS in this qualifying work was used Microsoft SQL Server 2005.

MAZMUNNOMA

Bituruv malaka ishida kadrlar agentligining faoliyati o'rganilib, bu yo'nalish bo'yichaa informatsion tizim yaratilgan. Shuningdek, foydalanuvchi uchun tushunarli va qulay do'stona interfeys yaratilgan. Bundan tashqari ma'lumotlar ombori, hisobot yaratish va uni to'gridan to'gri informatsion tizimdan chop etish, hamda MS Word va MS Excel dasturlariga ma'lumotlarini import va eksport qilish funksiyalari yaratilgan. Dastur C# dasturlash tilida yozilgan. Ma'lumotlar ombori bilan o'zaro aloqa instrumenti sifatida ADO.NET texnologiyasidan foydalanilgan BMI da MOBT sifatida Microsoft SQL Server 2005 qo'llanilgan.

СОДЕРЖАНИЕ

АННОТАЦИЯ.....	4
SUMMARY	5
MAZMUNNOMA	6
СОДЕРЖАНИЕ	7
ВВЕДЕНИЕ.....	8
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	11
1.1 Изучение необходимости автоматизации рабочего места	11
1.2 Анализ существующих информационных систем	13
1.3 Постановка задачи	15
2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАДРОВОГО АГЕНТСТВА	17
2.1. Основные требования и функциональные компоненты.....	17
2.2. Разработка архитектуры информационной системы	18
2.3. Выбор средств реализации	23
2.4. Проектирование базы данных	32
Выводы ко второй главе.....	37
3. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАДРОВОГО АГЕНТСТВА	38
3.1. Реализация функциональных модулей.....	38
3.2. Организация экспорта и импорта в MS Excel.....	45
3.3. Создание отчетов	48
Выводы к третьей главе.	61
4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ	62
4.1. Производственный микроклимат.....	62
4.2. Пожарная безопасность.....	65
ЗАКЛЮЧЕНИЕ.	74
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	76

ВВЕДЕНИЕ

В настоящее время история развития систем, предназначенных для хранения и обработки информации с использованием ЭВМ, насчитывает уже более полувека. Еще относительно недавно в ходу были перфораторы в качестве устройств ввода данных, листинги в виде рулонов бумажной ленты длиной порою до нескольких метров - в качестве носителя результатов машинной обработки, недельные, либо месячные временные интервалы - в качестве нормативных сроков обработки информации[2].

Работа кадрового агентства нуждается в автоматизации ведения банка данных клиентов и заказчиков, поскольку спрос на вакансии и спрос на работников существует всегда. При огромном меняющемся потоке информации о вакансиях агентству очень сложно оперативно и точно предоставлять необходимые данные, оперируя только офисными принадлежностями. Поэтому, очень важным встает вопрос об автоматизации работы агентства, в частности специалиста по кадрам, который непосредственно общается с клиентами и подбирает для них вакансии. А также важным является подбор опытных специалистов для заказчика, который обращается в кадровое агентство[4].

Объектом данной выпускной работы является автоматизация работы кадрового агентства «Profі Staff», поэтому основной задачей является рассмотреть и спроектировать информационную систему которая будет иметь основные функции введения данных, как соискателя, так и заказчика, создавать события относящихся к сведению обеих сторон, организовывать поиск кандидатов по нужным критериям, возможность создания необходимых отчетов по проделанной работе, хранения резюме и возможность импорта и экспорта в другие приложения.

Для реализации поставленной цели необходимо выполнить следующие задачи:

✓ Изучить объект автоматизации и механизм учета клиентов кадрового агентства,

✓ Разработать систему автоматизации учета кадров: создание баз данных по работникам и работодателям, их ведение, создание входных и выходных форм учета клиентов агентства, выдача списков для специалиста кадровой службы,

✓ произвести пробный ввод в эксплуатацию и доработка по результатам тестирования,

✓ внедрение на предприятии.

Система предназначена для получения достоверной и оперативной информации, а также непрерывного функционирования в течение всего рабочего дня, в связи с этим к системе предъявляются следующие требования:

✓ удобство и простота работы с системой, ориентация на пользователя, не имеющего опыта работы на компьютере;

✓ оперативность полученных результатов;

✓ интегрированность системы, т.е. обеспечение тесной взаимосвязи между различными ее частями, однократность ввода данных и их многократное использование в дальнейшем;

Выпускная квалификационная работа состоит из введения, четырёх глав, заключения и приложений.

Глава 1. Анализ предметной области – дано понятие необходимости автоматизации рабочего места, анализ существующих решений и круг решаемых задач, создаваемого программного обеспечения, рассмотрена постановка задачи.

Глава 2. Проектирование информационной системы кадрового агентства - определён ход и порядок действий проектирования программы, а также её структура, определен инструментарий.

Глава 3. Реализация информационной системы - посвящена созданию программного обеспечения для кадрового агентства. Дана подробная инструкция к использованию.

Глава 4. Безопасность жизнедеятельности. – Проводится ознакомление с мерами безопасности при работе с персональным компьютером.

Заключение. – Приведены основные выводы выпускной квалификационной работы

Приложение. – Структура таблиц, входных форм. Код программ основных процедур.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1 Изучение необходимости автоматизации рабочего места

Стабильность деятельности компаний, их эффективность и конкурентоспособность в рыночных условиях во многом зависят от быстроты и степени радикальности мышления, как управляющего звена, так и всего персонала предприятия. Решение задач менеджмента напрямую зависит от персонала, работающего в компании. При этом важнейшей целью в деятельности руководителей является эффективное управление человеческим ресурсом. Эффективное управление начинается с профессионального поиска и подбора специалистов. В настоящее время в Узбекистане в связи с социально-экономическими преобразованиями особое значение для организаций приобретает проблема квалифицированного подбора персонала.

Подбор кадров один из самых ответственных этапов в управлении персоналом, поэтому на современном уровне развития бизнеса многие компании обращаются к услугам профессионалов кадровых агентств. В связи с этим квалифицированный подбор персонала, как один из важнейших аспектов HR-менеджмента, становится востребованной профессиональной деятельностью. Рынок подобных услуг выполняет социально важную функцию посредничества между организациями – работодателями и работниками, способствует оптимизации деятельности компаний.

Согласно международной классификации, кадровые агентства делятся на два вида: агентства по трудоустройству, которые подбирают работу соискателям и берут с них оплату за оказанные услуги, и рекрутинговые агентства, которые находят необходимые работодателям кадры и получают вознаграждение за подбор специалистов с компаний-заказчиков.

Субъектами рекрутинговой деятельности являются представители компании-заказчики (работодатели, клиенты), которые размещают информацию о вакантных местах в агентстве, рекрутеры и кандидаты

(соискатели), специалисты которые находятся в активном поиске работы, либо могут быть приглашены на новые места работы. Задачей рекрутера является организация поиска, отбора и представление высококвалифицированных специалистов в соответствии с потребностями компании-заказчика и оказание консультационно-информационных услуг по анализу рынка труда. Рекрутер – это специалист, профессионально занимающийся подбором персонала, кто использует современные технологии. Под технологией рекрутмента подразумеваются четкие правила и приемы выполнения последовательных операций, обеспечивающих успешный подбор нужных специалистов.

Основными инструментами найма персонала являются: отбор по резюме и анкетам, отбор в процессе интервью. В отличие от найма персонала, в рекрутинге после тщательного отбора, проведенного менеджерами по персоналу, клиенту предоставляется короткий список претендентов. Рекрутером оцениваются уровень квалификации кандидатов, соответствие корпоративной культуре компании-заказчика, изучаются мотивирующие факторы, которые могут повлиять на решение кандидата о переходе на другую работу, его успешный опыт и многое другое. В итоге заказчик проводит сравнительный анализ предложенных кандидатур и выявляется лучшая из них.[4]

Специфика Узбекского рынка рекрутинговых услуг заключается в том, что он изначально стал формироваться как самостоятельный вид бизнеса, а не сопутствующий управленческому и финансовому консультированию, как это было в других странах.

Раньше обслуживание клиентов велось ручным способом. С одной стороны, составлялась картотека анкет, заключался договор с работником на поиск вакансии. С другой стороны, велись карточки работодателей, заключались договора на поиск специалистов. Результат трудоустройства отслеживался специалистом кадровой службы.

Автоматизация данного процесса является важным, поскольку ввод и обработка анкет и карточек работодателей значительно сокращает время обслуживания одного клиента, при этом поиск и выдача результатов необходимой информации осуществляется точно и оперативно. Значительно упрощается процесс отслеживания выполнения обязательств по договорам.

В данном случае специалист агентства наиболее полно владеет информацией о клиентах, осуществляет подбор в кратчайшие сроки и с соответствием установленных требований.

Под проектированием автоматизированных экономических информационных систем понимается процесс разработки технической документации, связанный с организацией системы получения и преобразования исходной информации в результативную, т.е. с организацией автоматизированной информационной системы. Документ, полученный в процессе проектирования, носит название проект. Целью проектирования является подбор технического и формирование информационного, математического, программного и организационно-правового обеспечения.[2]

Успешная работа ИС в первую очередь определяется качеством проектирования, именно при проектировании создается система, способная функционировать при постоянном ее совершенствовании.

1.2 Анализ существующих информационных систем

На сегодняшний день рынок программного обеспечения демонстрирует богатый выбор тех или иных систем автоматизации рабочего места кадрового агентства. Таковы, например, конфигурация «АИСТ:Кадровое Агенство», «E-Staff Рекрутер», «Учет менеджеров и клиентов» - специализированная конфигурация для 1С:Предприятие.

Сравнительная характеристика вышеизложенных систем приведена в таблице 1.1.

Название ИС Функции	«Дельта КА»	«E-Staff Рекрутер»	«АИСТ:Кадровое Агентство»
Ввод и хранение заявок	ДА	ДА	ДА
Импорт и Экспорт данных	ДА	ДА	ДА
Прикрепление файлов	ДА	ДА	ДА
Поиск по критериям	ДА	ДА	ДА
Сканирование документов	ДА	НЕТ	НЕТ
Печать	ДА	ДА	ДА
Учет долгов	ДА	ДА	ДА
Тестирование	НЕТ	НЕТ	ДА
История контактов	ДА	ДА	ДА
Размещение информации в Интернете	НЕТ	ДА	НЕТ
Архивирование файлов	ДА	НЕТ	ДА
Администрирование	НЕТ	ДА	ДА

Табл.1.1.– Таблица функциональности ИС

Исходя из приведенных данных информационная система кадрового учета должна обладать следующими функциональными возможностями:

- ✓ Ведение анкет соискателей. Анкеты содержат всю необходимую информацию о соискателе.
- ✓ Ввод и хранение заявок на подбор персонала от клиента-работодателя кадровому агентству.
- ✓ Получение оперативного доступа к единой корпоративной базе анкет и заявок;
- ✓ Оперативный поиск анкет и заявок по любому заданному критерию, включая полнотекстовый поиск по ключевому слову.

- ✓ Автоматическая публикация объявлений о вакансиях в Интернет:
- ✓ Поиск кандидатов в Интернет.
- ✓ Импорт резюме кандидатов из почтовых ящиков и документов Microsoft Word

Данные системы представляют хороший пример продуктов, которые получили распространение на рынке систем кадровых агентств. Однако при всех основных возможностях основным недостатком является не дешевизна данных продуктов и не простота пользовательского интерфейса. Данные системы не обладают совместимостью со многими операционными системами, и требуют обучения рабочего персонала. Расширяемость таких систем требует дополнительной оплаты со стороны пользователя.

1.3 Постановка задачи

После того, как было выяснено основные направления работы кадрового агентства и определены объекты требуется сформировать основную задачу проектирования информационной системы

Таким образом, при создании информационной системы кадрового агентства необходимо решить следующие задачи:

- ✓ Продумать и спроектировать базу данных, которая предусматривает ввод большого количества записей, а также удовлетворяет современной информационной безопасности.
- ✓ Рассмотреть и разработать понятный и удобный интерфейс пользователя, который не составит труда в эксплуатации, предусматривающий некорректный ввод данных.
- ✓ Выбрать необходимую технологию доступа к данным применить функции ведения данных, поиска, фильтрации и выполнения корректных запросов.
- ✓ Применить в системе возможность работы с выходными отчетами по запросу пользователя, так как они представляют необходимую составляющую современных информационных систем.

✓ Организовать функции импорта и экспорта в табличный и текстовый редакторы, такими являются MS Excel MS и Word.

✓ Организовать хранилище файлов, которые при необходимости могут использоваться и редактироваться пользователем.

2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАДРОВОГО АГЕНТСТВА

2.1. Основные требования и функциональные компоненты

При создании индивидуальной конфигурации, необходимо собрать требования, которые бы помогли создать интерфейс таким образом, чтобы конечному пользователю было просто и понятно ею пользоваться.

Данный этап является важным, поскольку именно здесь заказчиком и разработчиком рассматривается первоначальное создание виртуального программного обеспечения исходя из представленных требований. Поэтому требования, которые предоставляет заказчик, должны быть сформулированы правильно, четко и корректно.

Главные требования к новому программному обеспечению были собраны при помощи рассмотрения некоторых аналогичных решений. Этот метод достаточно интересен тем что, при его использовании был рассмотрен некоторый предыдущий опыт в данной области.

Заказчики представили ряд главных требований, которые должны быть рассмотрены в первую очередь:

- ✓ эргономичность пользовательского интерфейса;
- ✓ ввод и редактирование различных видов данных;
- ✓ надежное хранение большого объема информации;
- ✓ вывод конечных результатов в удобном и наглядном для пользователя виде;
- ✓ возможности сравнения, сортировки, поиска;
- ✓ создание форм для печати бланков и результатов.

Результатом проектирования на данной стадии является разработка технико-экономического обоснования необходимости создания автоматизированной информационной системы для кадрового агентства.

2.2. Разработка архитектуры информационной системы

Современные программные приложения и информационные системы достигли такого уровня развития, что термин "архитектура" в применении к ним уже давно не удивляет. Грамотно построить информационную систему, эффективно и надежно функционирующую не простая задача. Существуют множество определений понятия архитектуры и архитектуры информационной системы.

Архитектура – это базовая организация системы, воплощенная в ее компонентах, их отношениях между собой и с окружением, а также принципы, определяющие проектирование и развитие системы.

Архитектура информационной системы – концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

Архитектура – это набор значимых решений по поводу организации системы программного обеспечения, набор структурных элементов и их интерфейсов, при помощи которых компоуется система, вместе с их поведением, определяемым во взаимодействии между этими элементами, компоновка элементов в постепенно укрупняющиеся подсистемы, а также стиль архитектуры, который направляет эту организацию – элементы и их интерфейсы, взаимодействия и компоновку.

Архитектура программы или компьютерной системы – это структура или структуры системы, которые включают элементы программы, видимые извне свойства этих элементов и связи между ними.

Архитектура программного обеспечения системы или набора систем состоит из всех важных проектных решений по поводу структур программы и взаимодействий между этими структурами, которые составляют системы. Проектные решения обеспечивают желаемый набор свойств, которые должна поддерживать система, чтобы быть успешной. Проектные решения предоставляют концептуальную основу для разработки системы, ее поддержки и обслуживания. Хотя определения несколько отличаются, можно

заметить немалую степень сходства. Например, большинство определений указывают на то, что архитектура связана со структурой и поведением, а также только со значимыми решениями, может соответствовать некоторому архитектурному стилю, на нее влияют заинтересованные в ней лица и ее окружение, она воплощает решения на основе логического обоснования.

Под архитектурой программных систем будем понимать совокупность решений относительно:

- ✓ организации программной системы;
- ✓ выбора структурных элементов, составляющих систему и их интерфейсов;
- ✓ поведения этих элементов во взаимодействии с другими элементами;
- ✓ объединение этих элементов в подсистемы;
- ✓ архитектурного стиля, определяющего логическую и физическую организацию системы: статические и динамические элементы, их интерфейсы и способы их объединения.

Архитектура программной системы охватывает не только ее структурные и поведенческие аспекты, но и правила ее использования и интеграции с другими системами, функциональность, производительность, гибкость, надежность, возможность повторного применения, полноту, экономические и технологические ограничения, а также вопрос пользовательского интерфейса.[2]

По мере развития программных систем все большее значение приобретает их интеграция друг с другом с целью построения единого информационного пространства предприятия. Как можно видеть из вышеприведенных определений интеграция является важнейшим элементом архитектуры. Для того чтобы построить правильную и надежную архитектуру и грамотно спроектировать интеграцию программных систем необходимо четко следовать современным стандартам в этих областях. Без

этого велика вероятность создать архитектуру, которая неспособна развиваться и удовлетворять растущим потребностям пользователей ИТ.

Рассмотрим классификацию программных систем по их архитектуре:

Централизованная архитектура;

- ✓ Архитектура "файл-сервер";
- ✓ Двухзвенная архитектура "клиент-сервер";
- ✓ Многозвенная архитектура "клиент-сервер";
- ✓ Архитектура распределенных систем;
- ✓ Архитектура Веб-приложений;
- ✓ Сервис-ориентированная архитектура.

Следует заметить, что, как и любая классификация, данная классификация архитектур информационных систем не является абсолютно жесткой. В архитектуре любой конкретной информационной системы часто можно найти влияния нескольких общих архитектурных решений.

При проектировании информационной системы была выбрана архитектура "клиент-сервер". Клиент-сервер (Client-server) это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг (сервисов), называемых серверами, и заказчиками услуг, называемых клиентами. Нередко клиенты и серверы взаимодействуют через компьютерную сеть и могут быть как различными физическими устройствами, так и программным обеспечением.

Первоначально системы такого уровня базировались на классической двухуровневой клиент-серверной архитектуре (Two-tier architecture). Под клиент-серверным приложением в этом случае понимается информационная система, основанная на использовании серверов баз данных. (Рис 2.1)



Рис 2.1 Классическое представление архитектуры "клиент-сервер"

На стороне клиента выполняется код приложения, в который обязательно входят компоненты, поддерживающие интерфейс с конечным пользователем, производящие отчеты, выполняющие другие специфичные для приложения функции. Клиентская часть приложения взаимодействует с клиентской частью программного обеспечения управления базами данных, которая, фактически, является индивидуальным представителем СУБД для приложения. Интерфейс между клиентской частью приложения и клиентской частью сервера баз данных, как правило, основан на использовании языка SQL. Поэтому такие функции, как, например, предварительная обработка форм, предназначенных для запросов к базе данных, или формирование результирующих отчетов выполняются в коде приложения. Клиентская часть сервера баз данных, используя средства сетевого доступа, обращается к серверу баз данных, передавая ему текст оператора языка SQL.

Преимуществами данной архитектуры являются:

- ✓ возможность, в большинстве случаев, распределить функции вычислительной системы между несколькими независимыми компьютерами в сети;

✓ все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов, а также на сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа;

✓ поддержка многопользовательской работы;

✓ гарантия целостности данных.

✓ К недостаткам такой архитектуры можно отнести

✓ неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;

✓ администрирование данной системы требует квалифицированного профессионала;

✓ высокая стоимость оборудования;

✓ бизнес логика приложений осталась на клиентском ПО.

При проектировании информационной системы, основанной на архитектуре "клиент-сервер", большее внимание следует обращать на грамотность общих решений. Технические средства пилотной версии могут быть минимальными (например, в качестве аппаратной основы сервера баз данных может использоваться одна из рабочих станций). После создания пилотной версии нужно провести дополнительную исследовательскую работу, чтобы выявить узкие места системы. Только после этого необходимо принимать решение о выборе аппаратуры сервера, которая будет использоваться на практике. [5]

Увеличение масштабов информационной системы не порождает принципиальных проблем. Обычным решением является замена аппаратуры сервера (и, может быть, аппаратуры рабочих станций, если требуется переход к локальному кэшированию баз данных). В любом случае практически не затрагивается прикладная часть информационной системы. На схеме 2.1 представлена структура информационной системы «Profi Staff».

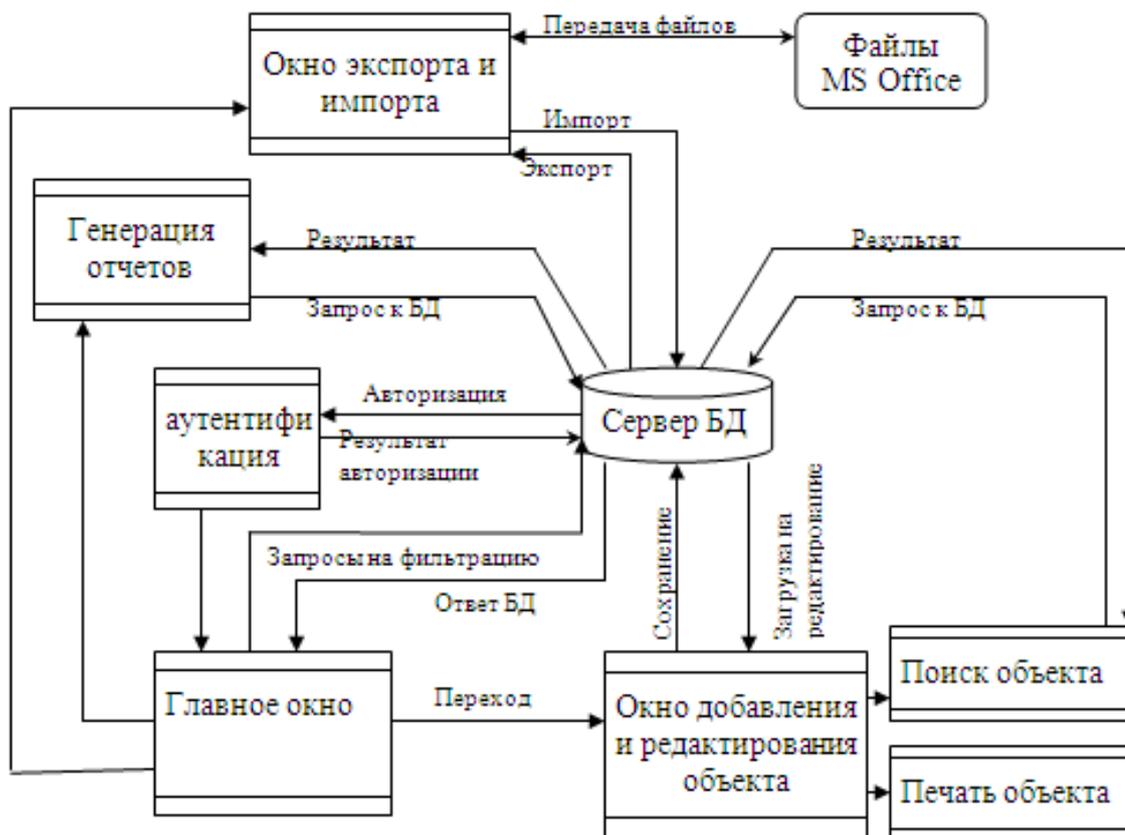


Схема 2.1 Структура информационной системы «Profi Staff»

2.3. Выбор средств реализации

При проектировании данной информационной системы особое внимание следует уделить средствам реализации, а именно обосновано подойти к выбору средств программирования и языка, а также выбору СУБД. При разработке данной информационной системы, основным языком реализации будет объектно-ориентированный язык программирования C# включенный в среду разработки Visual Studio .NET. Предпочитаемой СУБД для данной информационной системы является SQL Server 2005 так как данная СУБД является подходящей для организации клиент-серверной архитектуры системы.

Среда разработки Visual Studio .Net это проверенный временем программный продукт, являющийся седьмой версией Студии. Но новинки этой версии, связанные с идеей .Net, позволяют считать ее принципиально

новой разработкой, определяющей новый этап в создании программных продуктов. Выделю две важнейшие, на мой взгляд, идеи:

- ✓ открытость для языков программирования;
- ✓ принципиально новый подход к построению каркаса среды - Framework .Net.

Среда разработки теперь открытой языковой средой. Это означает, что наряду с языками программирования, включенными в среду фирмой Microsoft - Visual C++ .Net (с управляемыми расширениями), Visual C# .Net, J# .Net, Visual Basic .Net, - в среду могут добавляться любые языки программирования, компиляторы которых создаются другими фирмами-производителями. Таких расширений среды Visual Studio сделано уже достаточно много, практически они существуют для всех известных языков - Fortran и Cobol, RPG и Component Pascal, Oberon и SmallTalk. Я у себя на компьютере включил в среду компилятор одного из лучших объектных языков - языка Eiffel.

Открытость среды не означает полной свободы. Все разработчики компиляторов при включении нового языка в среду разработки должны следовать определенным ограничениям. Главное ограничение, которое можно считать и главным достоинством, состоит в том, что все языки, включаемые в среду разработки Visual Studio .Net, должны использовать единый каркас - Framework .Net. Благодаря этому достигаются многие желательные свойства: легкость использования компонентов, разработанных на различных языках; возможность разработки нескольких частей одного приложения на разных языках; возможность бесшовной отладки такого приложения; возможность написать класс на одном языке, а его потомков - на других языках. Единый каркас приводит к сближению языков программирования, позволяя вместе с тем сохранять их индивидуальность и имеющиеся у них достоинства. Преодоление языкового барьера - одна из важнейших задач современного мира. Благодаря единому каркасу, Visual Studio .Net в определенной мере решает эту задачу в мире программистов.

В каркасе Framework .Net можно выделить два основных компонента:

- ✓ статический - FCL (Framework Class Library) - библиотеку классов каркаса;
- ✓ динамический - CLR (Common Language Runtime) - общезыковую исполнительную среду.[6]

Язык C# является наиболее известной новинкой в области создания языков программирования. В отличие от 60-х годов XX века - периода бурного языкотворчества - в нынешнее время языки создаются крайне редко. За последние 15 лет большое влияние на теорию и практику программирования оказали лишь два языка: Eiffel, лучший, по моему мнению, объектно-ориентированный язык, и Java, ставший популярным во многом благодаря технологии его использования в Интернете и появления такого понятия как виртуальная Java-машина. Чтобы новый язык получил признание, он должен действительно обладать принципиально новыми качествами. Языку C# повезло с родителями. Явившись на свет в недрах Microsoft, будучи наследником C++, он с первых своих шагов получил мощную поддержку. Однако этого явно недостаточно для настоящего признания достоинств языка.

Создателем языка является сотрудник Microsoft Андреас Хейлсберг. Он стал известным в мире программистов задолго до того, как пришел в Microsoft. Хейлсберг входил в число ведущих разработчиков одной из самых популярных сред разработки - Delphi. В Microsoft он участвовал в создании версии Java - J++, так что опыта в написании языков и сред программирования ему не занимать. Как отмечал сам Андреас Хейлсберг, C# создавался как язык компонентного программирования, и в этом одно из главных достоинств языка, направленное на возможность повторного использования созданных компонентов. Из других объективных факторов отметим следующие:

- ✓ C# создавался параллельно с каркасом Framework .Net и в полной мере учитывает все его возможности - как FCL, так и CLR;

✓ C# является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;

✓ C# является мощным объектным языком с возможностями наследования и универсализации;

✓ C# является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход программистов от C++ к C#;

✓ сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как указатели, адресация, разыменование, адресная арифметика;

✓ благодаря каркасу Framework .Net, ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java. Эффективность кода даже повышается, поскольку исполнительная среда CLR представляет собой компилятор промежуточного языка, в то время как виртуальная Java-машина является интерпретатором байт-кода;

✓ мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;

✓ реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Microsoft SQL Server 2005 – это реляционная система управления базой данных (СУБД). В реляционных базах данных данные хранятся в таблицах. Взаимосвязанные данные могут группироваться в таблицы, кроме того, могут быть установлены также и взаимоотношения между таблицами. Отсюда и произошло название реляционные – от английского слова relational (родственный, связанный отношениями, взаимозависимый). Пользователи

получают доступ к данным на сервере через приложения, а администраторы, выполняя задачи конфигурирования, администрирования и поддержки базы данных, производят непосредственный доступ к серверу. SQL Server является масштабируемой базой данных, это значит, что она может хранить значительные объемы данных и поддерживать работу многих пользователей, осуществляющих одновременный доступ к базе данных.

СУБД SQL Server появилась в 1989 году и с тех пор значительно изменилась. Огромные изменения претерпели масштабируемость продукта, его целостность, удобство администрирования, производительность и функциональные возможности. Система SQL Server может быть реализована либо как клиент-серверная система, либо как автономная "настольная" система. Тип проектируемой вами системы зависит от количества пользователей, которые должны одновременно осуществлять доступ к базе данных, и от характера работ, которые должны выполняться. [9]

Клиент-серверная система SQL Server может иметь двухзвенную установку (two-tier setup) либо трехзвенную установку (three-tier setup). Независимо от варианта установки, программное обеспечение и базы данных SQL Server размещаются на центральном компьютере, который называется сервер базы данных(database server). Пользователи работают на отдельных компьютерах, которые называются клиенты (clients). Доступ пользователей к серверу базы данных производится при помощи приложений с их компьютеров-клиентов (в двухзвенных системах) либо при помощи приложений, выполняющихся на специально предназначенном для этой цели компьютере, который называется сервер приложений(application server) (в трехзвенных системах).

В частности, в двухзвенных системах клиенты исполняют приложения, осуществляющие доступ к серверу базы данных непосредственно через сеть. Таким образом, компьютеры-клиенты исполняют программный код, соответствующий нуждам предприятия, и код, отображающий для пользователя результаты доступа к базе данных. Такие клиенты называются

толстыми (thick client), потому что они выполняют два вида работы. Двухзвенная установка полезна при относительно небольшом количестве пользователей, потому что для соединения с каждым из пользователей расходуются системные ресурсы, такие как память и блокировки (locks). Чем больше будет количество соединений с пользователями, тем хуже будет производительность системы, из-за соперничества за ресурсы.

SQL Server может использоваться также и как автономный (stand-alone) сервер базы данных, работающий на настольном или на портативном компьютере. Обычно такие конфигурации называют настольными системами (desktop system). В них клиентские приложения исполняются на том же компьютере, на котором хранится программное обеспечение, реализующее механизм работы SQL Server и базы данных. В данной системе применяется только один компьютер, поэтому не устанавливаются никакие сетевые соединения от клиента к серверу – клиент устанавливает локальное соединение со своей локальной установкой SQL Server.

Для создания связи между программной частью и СУБД использовалась технология ADO .NET. ADO .NET (ActiveX Data Objects .NET) является набором классов, реализующих программные интерфейсы для облегчения подключения к базам данных из приложения независимо от особенностей реализации конкретной системы управления базами данных и от структуры самой базы данных, а также независимо от места расположения этой самой базы — в частности, в распределенной среде (клиент-серверное приложение) на стороне сервера.[13]

ADO .NET широко используется совместно с технологией web-программирования с использованием объектов ASP .NET для доступа к расположенным на сервере базам данных со стороны клиента.

Решение даже самой простой задачи, связанной с данными, предполагает использование множества разнообразных объектов – представителей классов ADO .NET, которые находятся между собой в достаточно сложных взаимоотношениях. Из-за этих отношений строго

последовательное описание элементов ADO .NET представляется весьма проблематичным. С какого бы элемента ни начиналось описание, всегда предполагается предварительное представление о множестве других элементов.

Работа с БД на уровне приложения .NET – это работа:

- ✓ с множествами объявлений классов, которые содержат объявления унаследованных методов и свойств, предназначенных для решения задачи извлечения информации из базы данных;

- ✓ с множеством объектов-представителей классов, которые обеспечивают работу с базами данных;

- ✓ с множеством значений и свойств конкретных объектов, отражающих специфику структуры конкретной базы данных.

Функциональные особенности этой сложной системы взаимодействующих классов обеспечивают ЕДИНООБРАЗНУЮ работу с базами данных независимо от системы управления базой и ее реализации.

Конечно же, при написании программы, взаимодействующей с базами данных, программист все может сделать своими руками. Правда, в силу сложности задачи, времени на разработку такого приложения может потребоваться достаточно много.

Для программиста – разработчика приложения принципиальной становится информация о логической организации (структуре таблиц, отношениях и ограничениях) данной конкретной базы данных — то есть о том, как эту базу видит приложение. Деятельность программиста – разработчика приложений для работы с базами данных в основе своей ничем не отличается от того, что было раньше. Те же объявления классов и интерфейсов. А потому желательно:

- ✓ понимать принципы организации и взаимодействия классов, которые обеспечивают работу с базой данных;

- ✓ представлять структуру, назначение и принципы работы соответствующих объектов;

- ✓ знать, для чего и как применять различные детали при организации взаимодействия с базами данных;

- ✓ уметь создавать компоненты ADO .NET заданной конфигурации с использованием вспомогательных средств (волшебников), предоставляемых в рамках Visual Studio .NET.

В ADO .NET используется модель доступа к отсоединенным данным. При этом соединение устанавливается лишь на то время, которое необходимо для проведения определенной операции над базой данных.

Объектная модель ADO .NET реализует отсоединенный доступ к данным. При этом в Visual Studio .NET существует множество встроенных мастеров и дизайнеров, которые позволяют реализовать механизмы доступа к БД еще на этапе разработки программного кода. С другой стороны, задача получения доступа к данным может быть решена непосредственно во время выполнения приложения.

Концепция доступа к данным в ADO .NET основана на использовании двух компонентов (рис. 2.2):

- ✓ НАБОРА ДАННЫХ (представляется объектом класса DataSet) со стороны клиента. Это локальное временное хранилище данных;

- ✓ ПРОВАЙДЕРА ДАННЫХ (представляется объектом класса DataProvider). Это посредник, обеспечивающий взаимодействие приложения и базы данных со стороны базы данных (в распределенных приложениях – со стороны сервера).

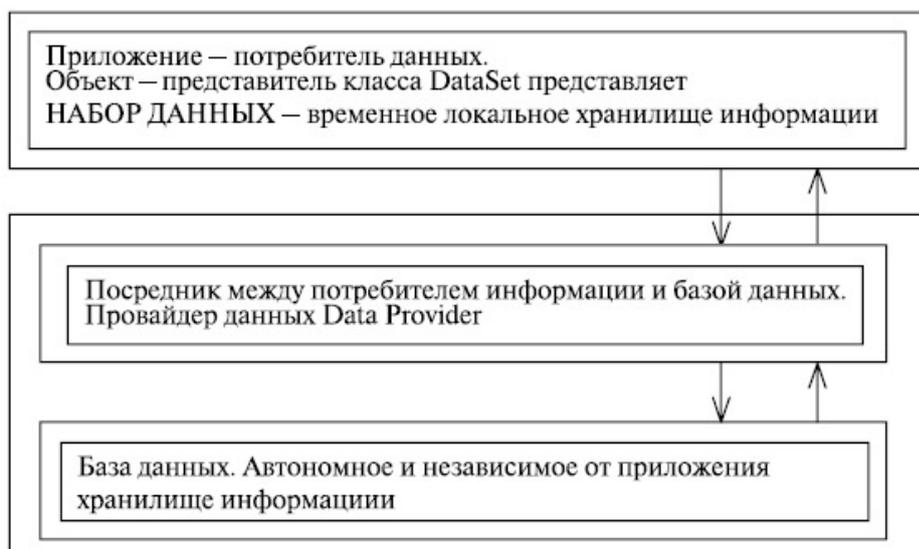


Рис. 2.2 Концепция доступа к данным в ADO .NET

Объектная модель ADO .NET предполагает существование (при написании приложения для работы с базой данных — использование) двух множеств классов, выполняющих четко определенные задачи при работе с базой данных рис. 2.3:

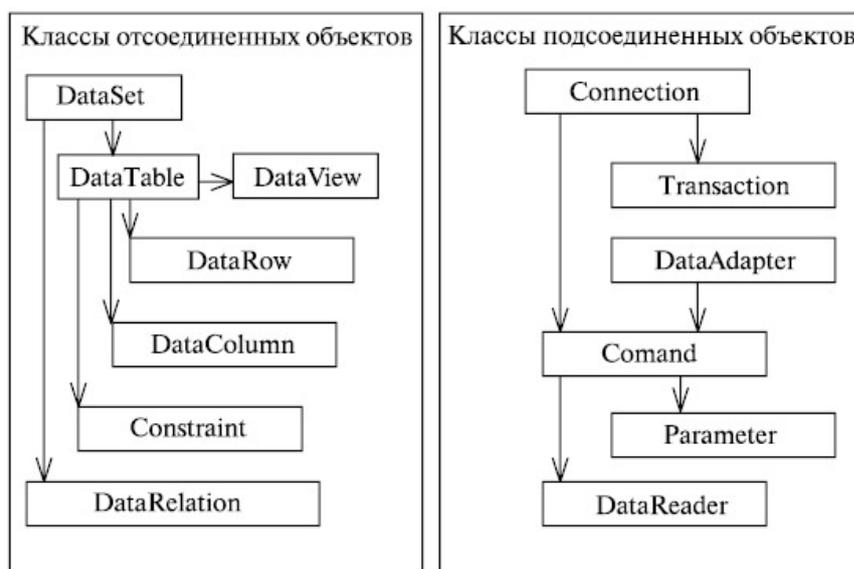


Рис. 2.3. Объектная модель ADO .NET

Классы подсоединенных объектов обеспечивают установление соединения с базой данных и управление базой со стороны приложения; классы отсоединенных объектов обеспечивают сохранение, использование и преобразование полученной от базы данных информации на стороне приложения.

2.4. Проектирование базы данных

Развитие компьютерных технологий, связанных с хранением и обработкой данных, привело к появлению в конце 60-х — начале 70-х годов специализированного программного-обеспечения, получившего название систем управления базами данных, или СУБД (DataBase Management Systems, DBMS). СУБД позволяют структурировать, систематизировать и организовывать данные для их компьютерного хранения и обработки. Именно системы управления базами данных являются основой практически любой информационной системы.

СУБД можно определить как некую систему управления данными, обладающую следующими свойствами:

- ✓ поддержанием логически согласованного набора файлов;
- ✓ предоставлением языка манипулирования данными;
- ✓ восстановлением информации после разного рода сбоев;
- ✓ обеспечением параллельной работы нескольких пользователей.

Основные функции систем управления базами данных.

К основным функциям, выполняемым системами управления базами данных, обычно относят:

- ✓ непосредственное управление данными во внешней памяти;
- ✓ управление буферами оперативной памяти;
- ✓ управление транзакциями;
- ✓ протоколирование;
- ✓ поддержку языков баз данных.

Широкое распространение реляционных СУБД и их использование в самых разных приложениях показывает, что реляционная модель данных достаточна для моделирования предметных областей. Однако проектирование реляционной базы данных в плане соответствия отношениям нормализации часто представляет собой очень сложный для проектировщика процесс. Это обусловлено некоторой ограниченностью реляционной модели данных, что особенно ярко проявляется в нескольких аспектах.[1]

Реляционная модель не предоставляет достаточных средств для раскрытия смысла данных. Проектировщик должен независимым от модели образом представлять семантику реальной предметной области. Примером может служить представление ограничений целостности.

В ряде случаев предметную область трудно моделировать на основе плоских таблиц. Сложности могут возникнуть на начальной стадии проектирования при описании предметной области в виде одной (возможно, даже ненормализованной) таблицы.

Хотя весь процесс проектирования происходит на основе учета зависимостей, реляционная модель не содержит никаких средств для представления этих зависимостей.

Процесс проектирования начинается с выделения некоторых объектов (сущностей) предметной области, существенных для приложения, и выявления связей между этими сущностями, реляционная модель данных не предлагает какого-либо аппарата для разделения сущностей и связей.

Для преодоления ограничений реляционной модели и обеспечения потребности проектировщиков баз данных в более удобных и мощных средствах моделирования предметной области проектирование баз данных обычно выполняется не в терминах реляционной модели, а с использованием концептуальных моделей предметной области.[12]

Обычно различают концептуальные модели двух видов:

- ✓ в объектно-ориентированных моделях сущности реального мира представляются в виде объектов, а не записей реляционных таблиц;
- ✓ семантические модели отражают значения реальных сущностей и отношений.

Объектно-ориентированную модель можно рассматривать как результат объединения семантической модели данных и объектно-ориентированного языка программирования.

Несмотря на то что в последнее время все большее распространение получают объектно-ориентированные модели, значение семантических моделей не снижается.

Процесс проектирования информационных систем является достаточно сложной задачей. Он начинается с построения инфологической модели данных, т.е. идентификации сущностей. Затем необходимо выполнить следующие шаги процедуры проектирования даталогической модели.

1. Представить каждый стержень (независимую сущность) таблицей базы данных (базовой таблицей) и специфицировать первичный ключ этой базовой таблицы.

2. Представить каждую ассоциацию (связь вида "многие-ко-многим" или "многие-ко-многим-ко-многим" и т.д. между сущностями) как базовую таблицу. Использовать в этой таблице внешние ключи для идентификации участников ассоциации и специфицировать ограничения, связанные с каждым из этих внешних ключей.

3. Представить каждую характеристику как базовую таблицу с внешним ключом, идентифицирующим сущность, описываемую этой характеристикой. Специфицировать ограничения на внешний ключ этой таблицы и ее первичный ключ – по всей вероятности, комбинации этого внешнего ключа и свойства, которое гарантирует "уникальность в рамках описываемой сущности".

4. Представить каждое обозначение, которое не рассматривалось в предыдущем пункте, как базовую таблицу с внешним ключом, идентифицирующим обозначаемую сущность. Специфицировать связанные с каждым таким внешним ключом ограничения.

5. Представить каждое свойство как поле в базовой таблице, представляющей сущность, которая непосредственно описывается этим свойством.

6. Для того чтобы исключить в проекте непреднамеренные нарушения каких-либо принципов нормализации, выполнить процедуру нормализации.

7. Если в процессе нормализации было произведено разделение каких-либо таблиц, то следует модифицировать инфологическую модель базы данных и повторить перечисленные шаги.

8. Указать ограничения целостности проектируемой базы данных и дать (если это необходимо) краткое описание полученных таблиц и их полей.

После того, как были определены основные объекты и произведена нормализация данных в СУБД SQL Server, были созданы следующие таблицы рис.2.3. и рис. 2.4.

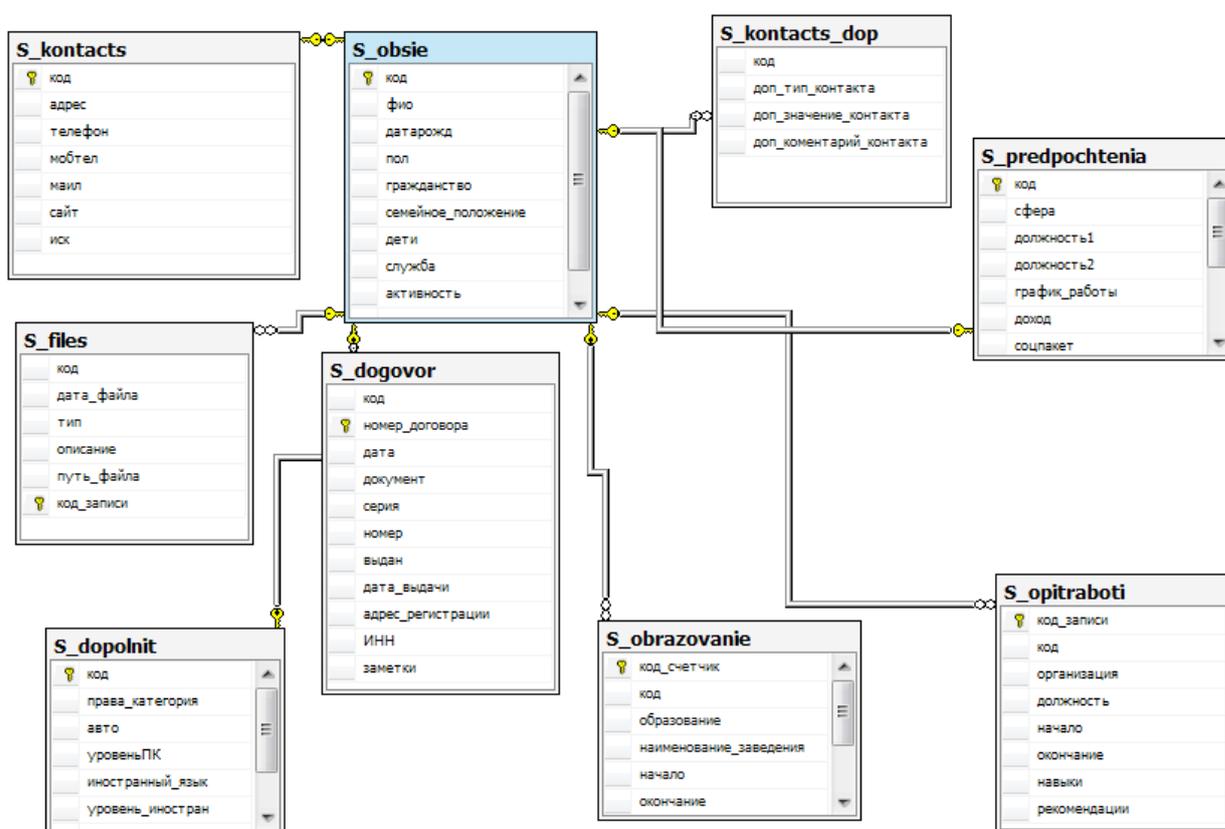


Рис 2.3. Физическая модель БД «Соискатель»

БД содержит следующие таблицы:

- ✓ S_obsie – содержит общую информацию о соискателе
- ✓ S_kontacts – содержит основные контакты соискателя
- ✓ S_kontacts_dop – дополнительные контакты и контактные лица
- ✓ S_obrazovanie – данные о полученном образовании и оконченных курсах

- ✓ K_contacts - содержит информацию о контактах и контактных лицах клиента
- ✓ Sobicie – таблица о событиях клиента и соискателя.

Выводы ко второй главе.

Во второй главе были рассмотрены следующие вопросы:

- Были рассмотрены требования к проектированию информационной системы
- Представлены функциональные компоненты ИС
- Произведен обзор средств реализации
- Показаны различные варианты проектирования базы данных и показана структура разработанной базы данных

3. РЕАЛИЗАЦИЯ ИНФОРМАЦИОННОЙ СИСТЕМЫ КАДРОВОГО АГЕНТСТВА

3.1. Реализация функциональных модулей

Данная выпускная работа реализована с применением компонентного и событийного программирования подхода с использованием СУБД. Так как выбор инструментария и важность средств был описан выше, то перейдем к непосредственному представлению реализации данной информационной системы. Среда разработки Visual Studio 2008 вобрала в себя лучшее и поэтому интерфейс и расположения инструментов сходно с такими средами как Borland Delphi Enterprise, Visual Basic 6.0 и др. При запуске среды выбирается тип создаваемого проекта и языка, в данном случае создается приложение под Windows рис.3.1.

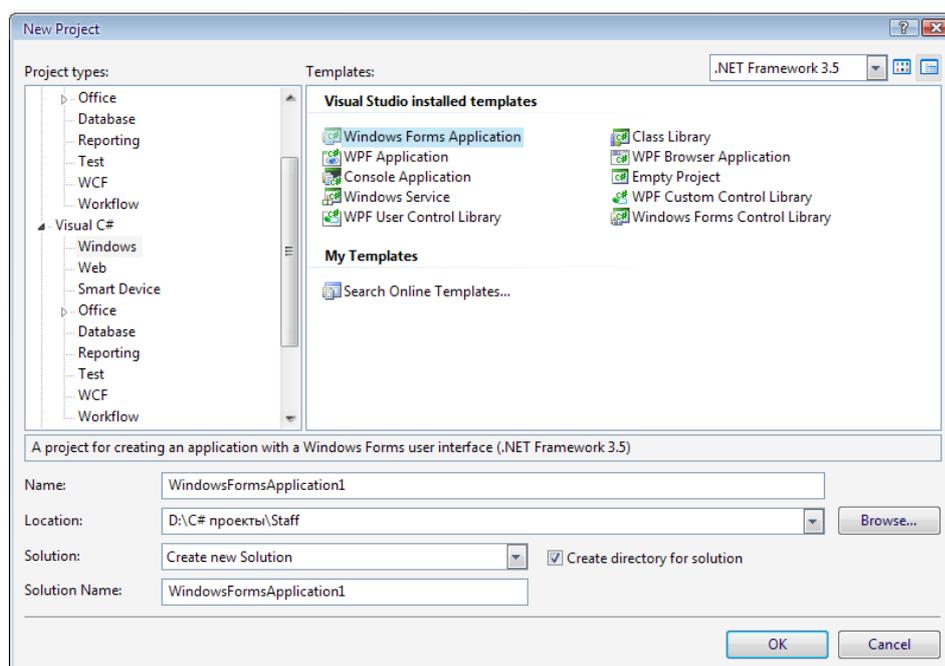


Рис.3.1. Окно выбора типа проекта и языка

После появляется стандартное окно с пустой формой и всем необходимым инструментарием. Слева находятся компоненты создания приложения, справа окно свойств и событий сверху панель управления проектом рис 3.2.

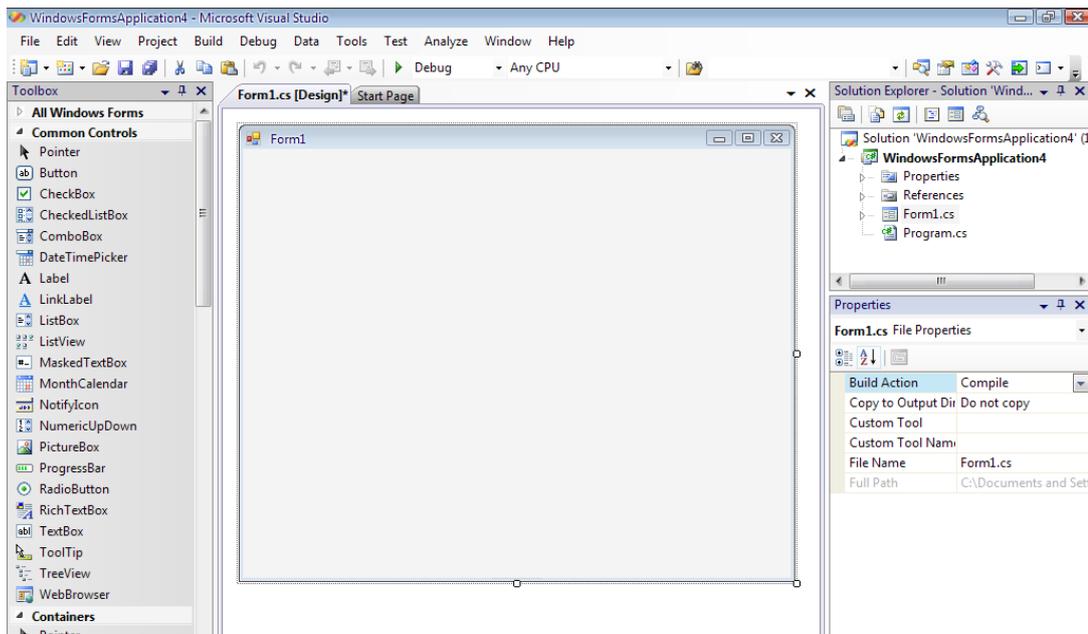


Рис 3.2. Главное окно работы с проектом

Реализованный проект на данный момент состоит из девяти созданных модулей. Первый модуль состоит из окна приветствия где осуществляется аутентификация пользователя т.е. выбирается сервер подключения и проверяется разрешение входа в систему рис. 3.3

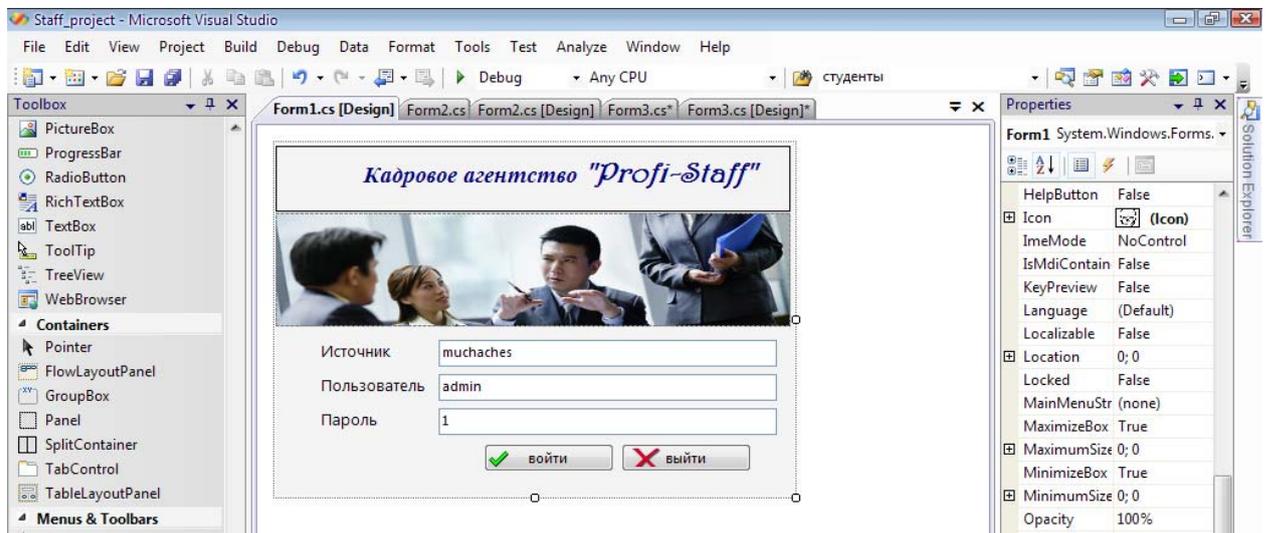


рис. 3.3 Аутентификация пользователя

Основными элементами управления на данной форме являются текстовые поля и кнопки. Следующий модуль содержит форму перехода для работы с объектами рис 3.4.

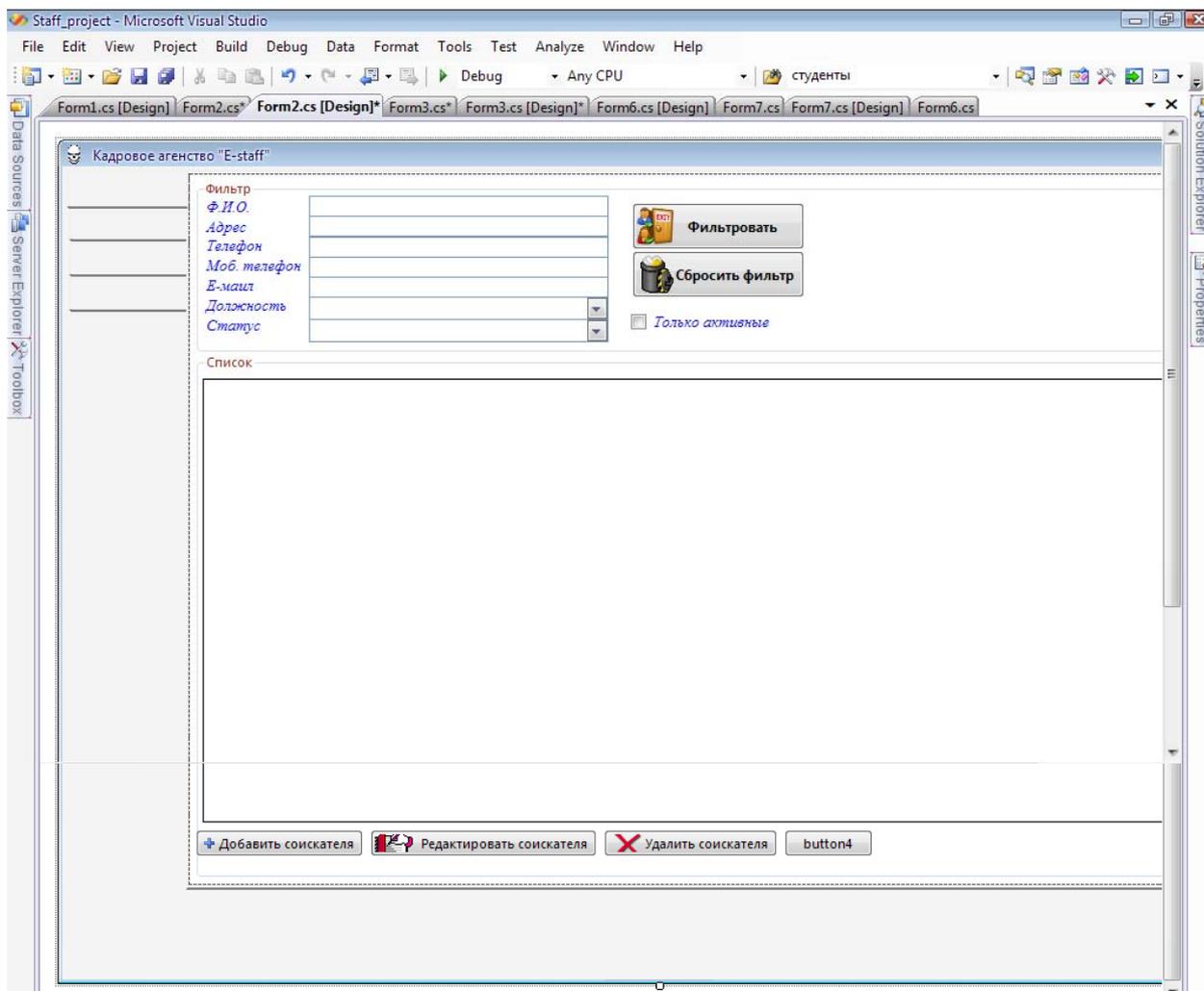


Рис 3.4. Форма для работы с объектами

На данной форме применены следующие элементы управления:

Элемент управления Windows Forms **TabControl** используется для отображения нескольких вкладок, аналогичных разделителям в записной книжке или меткам в наборе папок картотечного блока. Вкладки могут содержать рисунки и другие элементы управления.

Элемент управления **DataGridView** предоставляет мощный и гибкий способ отображения данных в табличном формате. Элемент управления **DataGridView** можно использовать для представления в режиме только чтения небольших объектов данных; можно расширить этот элемент для представления крупных объемов данных в режиме редактирования.

Функциональные возможности элемента управления **DataGridView** можно расширить несколькими способами, чтобы реализовать

пользовательское поведение в приложениях. Например, можно программно задать собственные алгоритмы сортировки, а также создать собственные типы ячеек. Внешний вид элемента управления **DataGridView** можно настроить, задав несколько свойств. В качестве источника данных могут использоваться различные типы хранилищ данных. Кроме того, элемент управления **DataGridView** может работать без связанных источников данных.

Для работы с соискателем был реализован следующий модуль рис 3.5.

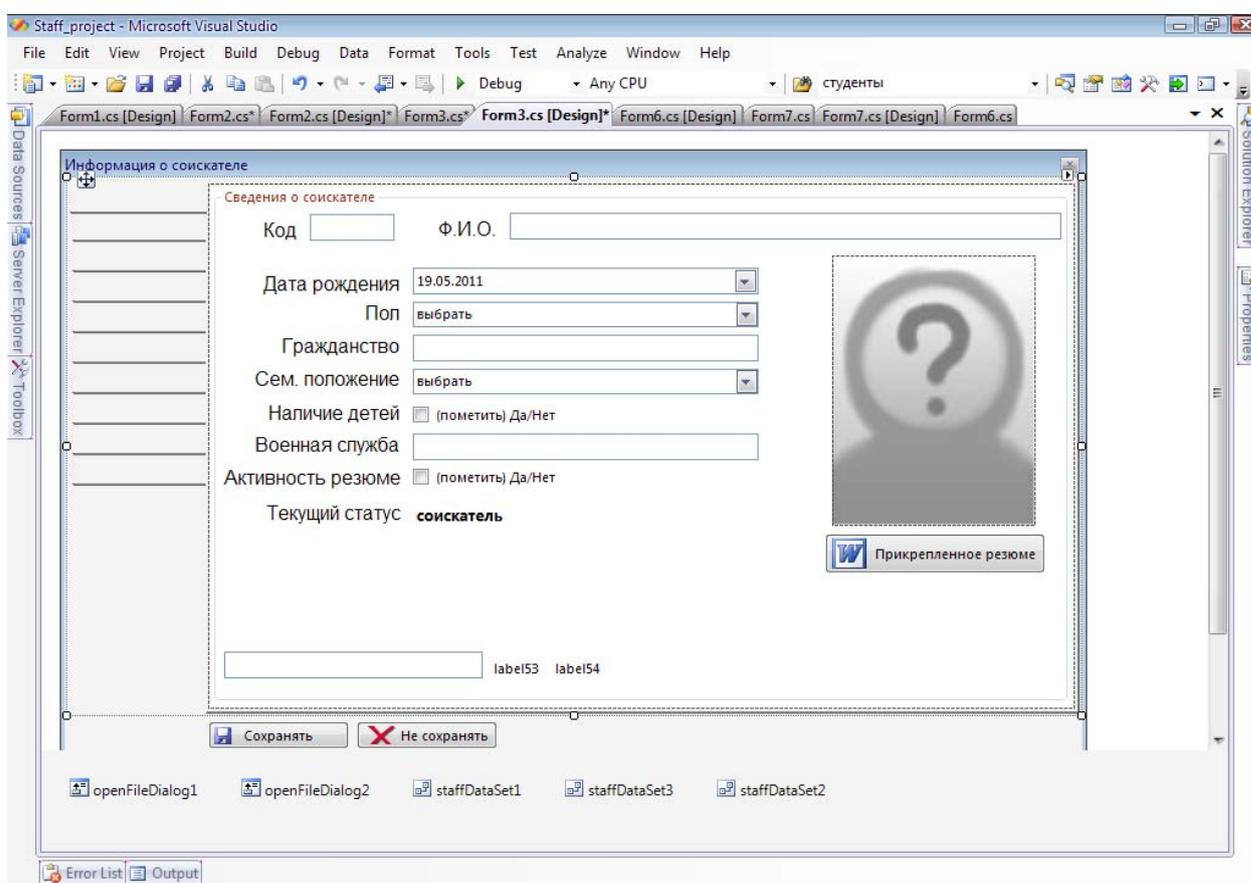


Рис 3.5. Модуль для работы с соискателем

На данной форме используется те же компоненты, что и при описанных выше только Компонент OpenFileDialog позволяет пользователям просматривать папки личного компьютера или любого компьютера в сети, а также выбирать файлы, которые требуется открыть. Диалоговое окно возвращает путь и имя файла, который был выбран пользователем.

После того, как пользователь выбирает файл, который требуется открыть, существует два подхода к открытию файла. Если разработчик предпочитает работать с потоками файлов, можно создать экземпляр класса `StreamReader`.

С другой стороны, для открытия выбранного файла можно использовать метод `OpenFile`, при использовании классов `OpenFileDialog` и `SaveFileDialog` в приложении их внешний вид и поведение зависит от используемой версии Windows. Если приложение, которое было создано в .NET Framework, версия 2.0 или более ранних версиях, отображается в Windows Vista, `OpenFileDialog` и `SaveFileDialog` автоматически отображаются с внешним видом и поведением Windows Vista. Начиная с .NET Framework 3.0 можно отказаться от автоматического обновления для отображения `OpenFileDialog` и `SaveFileDialog` в стиле Windows XP и с соответствующим поведением.

Объект `DataSet` является центральным элементом поддержки разьединенных распределенных сценариев данных в ADO.NET. Объект `DataSet` является находящимся в оперативной памяти представлением данных, обеспечивающим согласованную реляционную программную модель, независимо от источника данных. Он может использоваться с несколькими различными источниками данных, XML-данными или для управления данными, локальными по отношению к приложению. Объект `DataSet` представляет полный набор данных, включая связанные таблицы, ограничения и связи между таблицами. На следующей иллюстрации показана модель объектов `DataSet` рис 3.6

Методы и объекты `DataSet` согласованы с методами и объектами модели реляционной базы данных. Объект `DataSet` также может сохранять и повторно загружать свое содержимое в виде XML, а свою схему — в виде схемы XSD. [15]

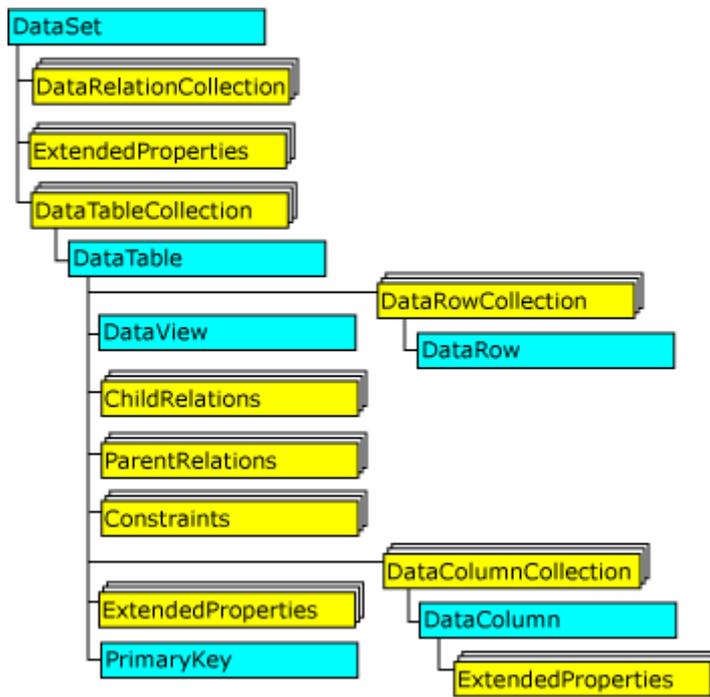


Рис 3.6 Модель объектов DataSet

Следующий модуль позволяет работать с клиентом с которым заключается договор на поиск работников рис 3.7.

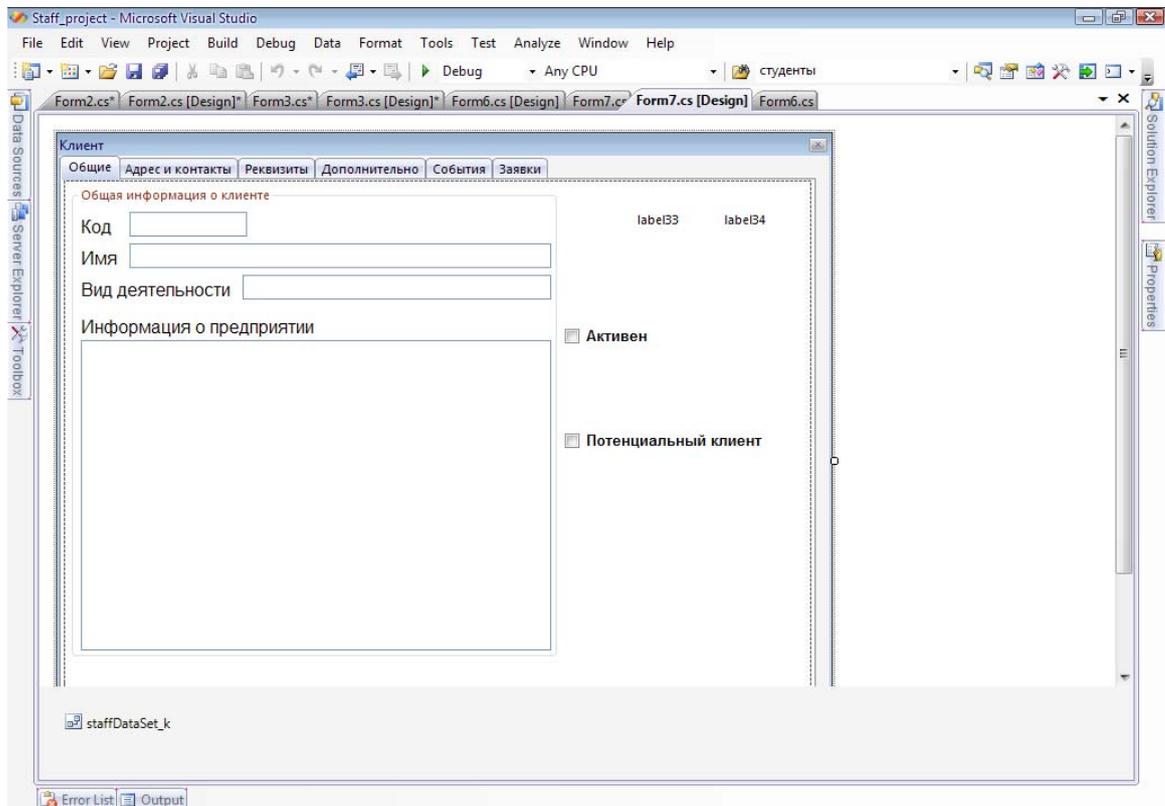


Рис 3.7. Форма для работы с клиентом

На рисунке 3.8. показан модуль, который будет организовывать события связанные с соискателями. Модуль событий для клиента аналогичен

модулю соискателя. Здесь также применены стандартные элементы управления.

Элемент управления Windows Forms **ComboBox** используется для вывода данных в раскрывающемся поле со списком. По умолчанию элемент управления **ComboBox** отображается в виде двух частей: верхняя часть представляет собой текстовое поле, в которое пользователь может ввести элемент списка. Вторая часть представляет собой список элементов, один из которых пользователь может выбрать.

Элемент управления Windows Forms **RadioButton** (переключатель) обеспечивает выбор из двух или более взаимоисключающих вариантов. Функции переключателей и флажков могут показаться схожими, но между ними есть важное отличие: в случае переключателя пользователь может выбрать лишь один вариант.

Элемент управления Windows Forms **DateTimePicker** позволяет пользователю выбрать отдельный элемент в списке дат или времени. Когда компонент используется для представления даты, он состоит из двух частей: раскрывающегося списка с датой, представленной в виде текста, и сетки, которая появляется при нажатии кнопки со стрелкой вниз, расположенной рядом со списком.

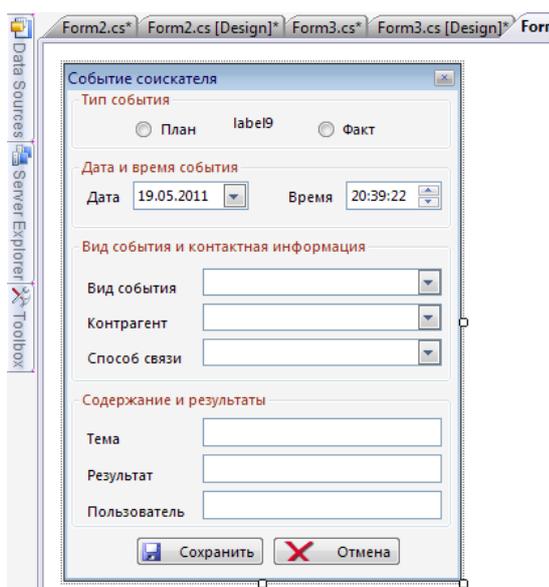


Рис 3.8. Форма для работы с событиями.

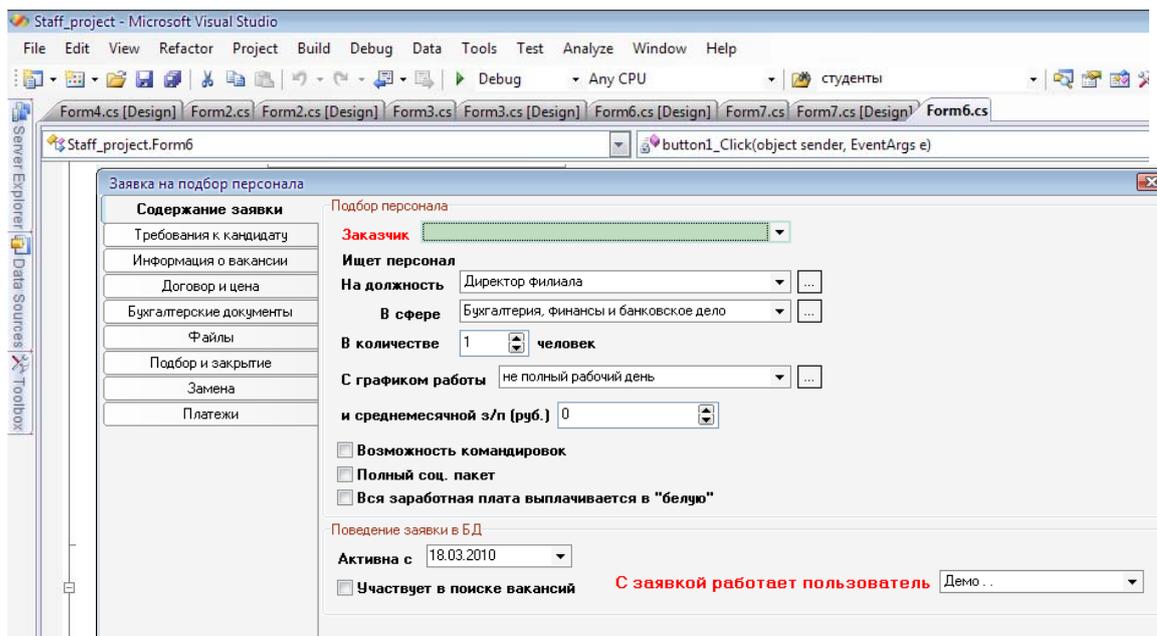


Рис 3.9. Форма для работы с заявками.

Данная форма предназначена для обработки запроса клиента на персонал, содержит стандартные элементы управления, назначение которых было описано выше.

3.2. Организация экспорта и импорта в MS Excel

С развитием информационных систем и программных продуктов, большое значение стало уделяться взаимопомощи таких систем как табличных и текстовых редакторов при работе с приложениями разработанных на заказ. С тех пор как офисные продукты компании Microsoft приобрели большую популярность, то во многих средах программирования стали использоваться библиотеки, которые позволяют организовать совместимость данных разрабатываемых информационных систем. К основным продуктам можно отнести MS Excel и MS Word.

Microsoft Excel это программа, предназначенная для организации данных в таблицы для документирования, сопоставления и графического представления информации. Например, можно использовать Excel для суммирования, вычисления среднего или максимального числа продаж за день; создания графика, показывающего определенный процент продаж,

сравнения общего объема продаж за день с тем же показателем других дней недели. Excel освобождает нас от проведения этих вычислений вручную.

Для того чтобы организовать процедуры экспорта и импорта из базы данных в MS Excel необходимо подключить библиотеку пример на рис 3.10

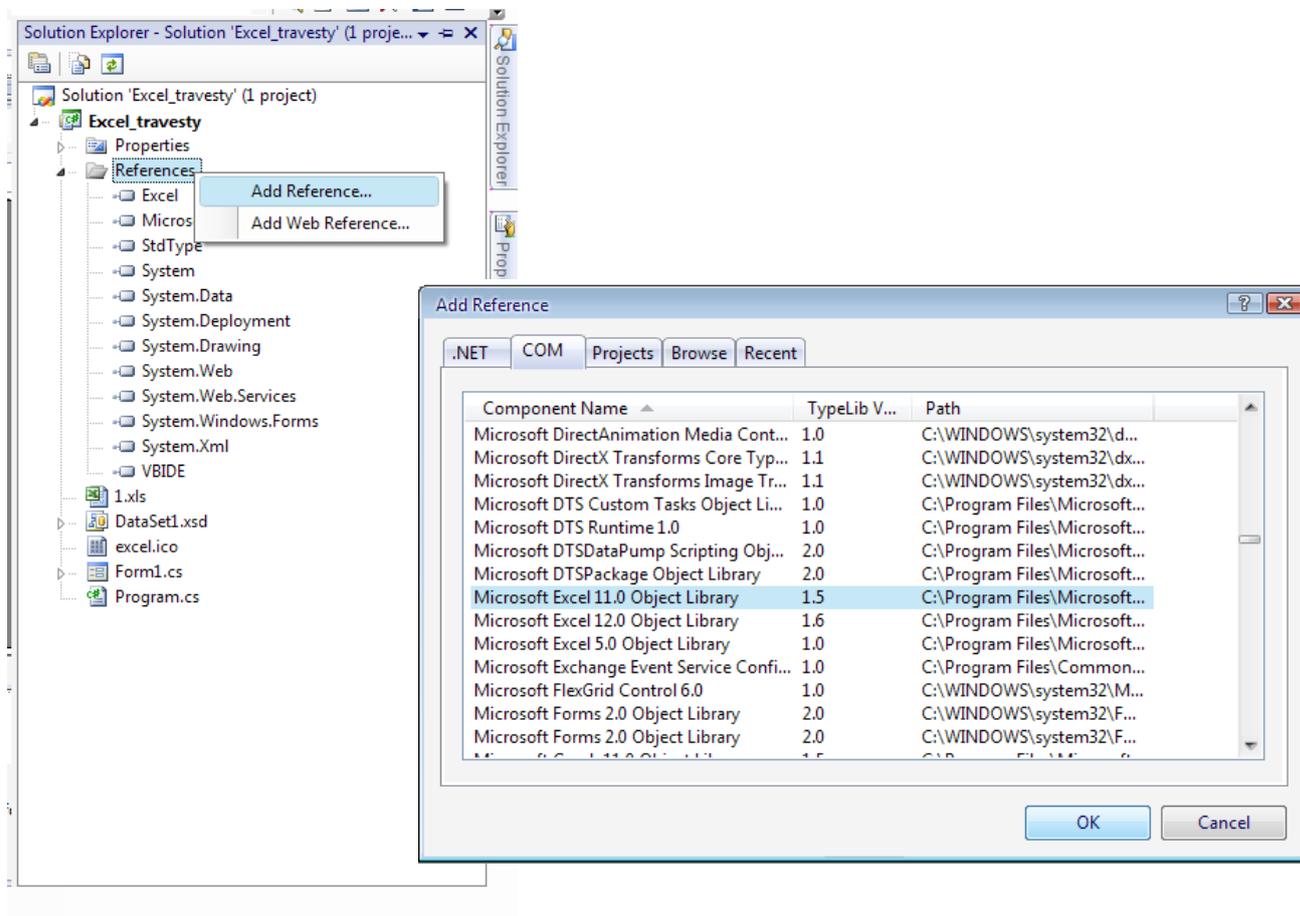


Рис 3.10 Подключение библиотеки Microsoft Excel 11.0

При подключении библиотеки открывается возможность использовать классы из Microsoft.Office.Interop.Excel здесь есть все необходимое чтобы обеспечит свой проект функциональностью работы с Excel, данные обычно берутся из компонента DataGridView рис 3.11.

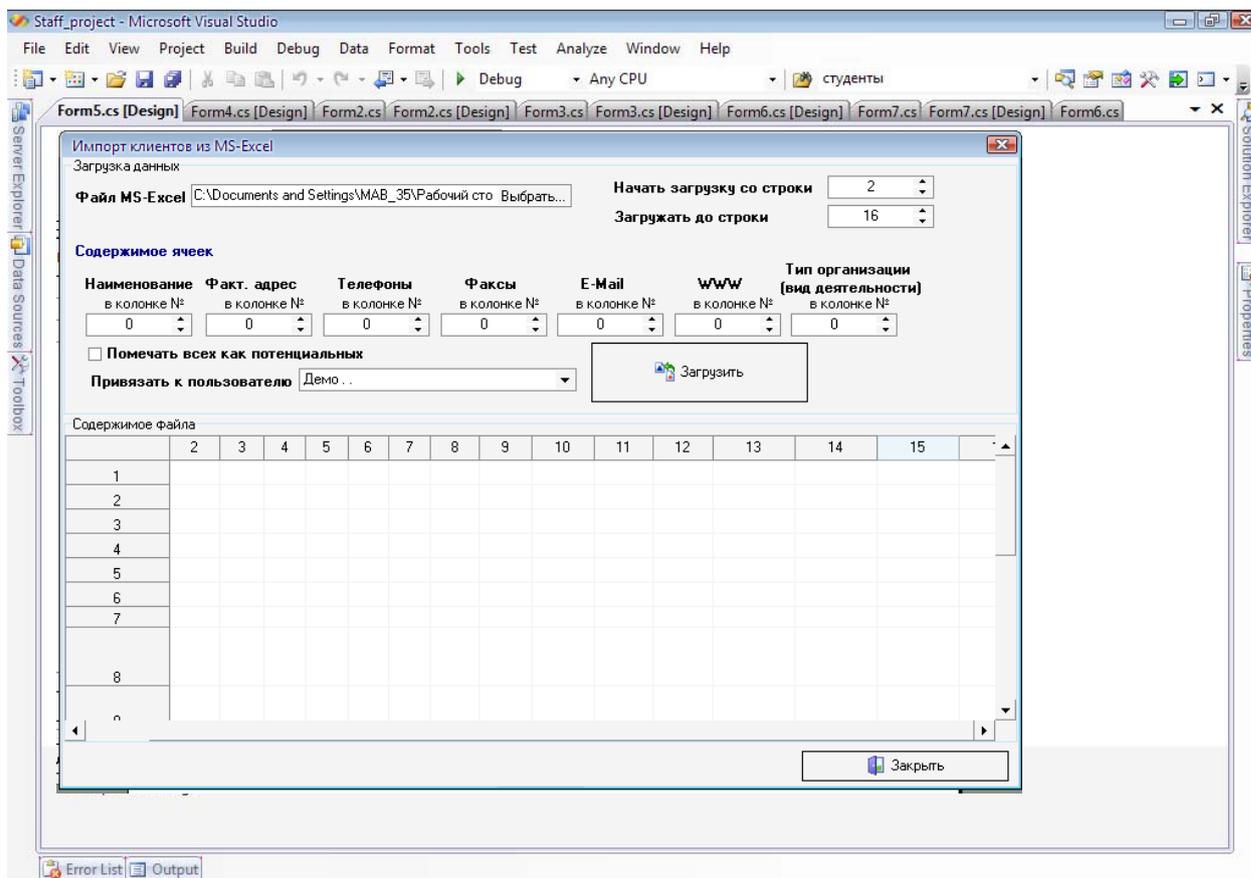


Рис 3.11 Форма импорта клиента из MS Excel

Процедура экспорта осуществляется с использованием той же библиотеки и тех же компонентов. Для того чтобы осуществить работу с классами этой библиотеки необходимо привязать к переменной библиотеку `using excel=Microsoft.Office.Interop.Excel` после можно работать с кодом программы.

```

Для импорта можно создать запрос используя ADO. Net
filename = openFileDialog1.FileName;
OleDbConnection olcon = new
OleDbConnection("Provider=Microsoft.
Jet.OLEDB.4.0;data source=" + filename + ";Extended
Properties=
\"Excel 8.0;HDR=NO;IMEX=1;\");
olcon.Open();
OleDbDataAdapter da = new OleDbDataAdapter("SELECT *
FROM [лист1$]", olcon);

```

```
DataTable dt = new DataTable();  
da.Fill(dt);
```

Для экспорта создаем следующие функции.

```
excel.Application app = new excel.Application();  
excel.Workbook wbook =  
app.Workbooks.Add(excel.XlWBATemplate.xlWBATWorksheet);  
excel.Worksheet wsheet =  
(excel.Worksheet)wbook.Worksheets[1];  
excel.Range range = wsheet.get_Range("A1", "Z" +  
rowcount.ToString());  
for (int i = 0; i < rowcount; i++)  
    for (int j = 0; j < columcount; j++)  
        masiv[i, j] = dg[tabControl1.SelectedIndex][j,  
i].Value;  
  
        range.Value2 = masiv;  
        wbook.Saved = true;  
        wbook.SaveCopyAs(filename);
```

3.3. Создание отчетов

Создание отчетов является неотъемлемой частью любой современной информационной системы. Отчеты позволяют сократить построение вручную необходимых шаблонов и тем самым снизить труд, затрачиваемый при помощи обычных офисных программ. При проектировании данной информационной системы, были использованы следующие компоненты –это ReportViewer и Report.

ReportViewer – отличный инструмент создания отчетов для Web и Windows проектов, с помощью которого вы можете сделать следующее. Подключится к существующей базе данных и сформировать отчет на основе таблиц содержащихся в ней. Дизайнер отчетов ReportViewer контекстно-зависимый, поэтому отчеты изменяются в зависимости от типа используемого проекта. Для того, чтобы начать использовать компонент

ReportViewer необходимо его перенести из панели инструментов и с помощью мастера настроить его, также к проекту необходимо добавить компонент Report рис 3.12 и рис 3.13.

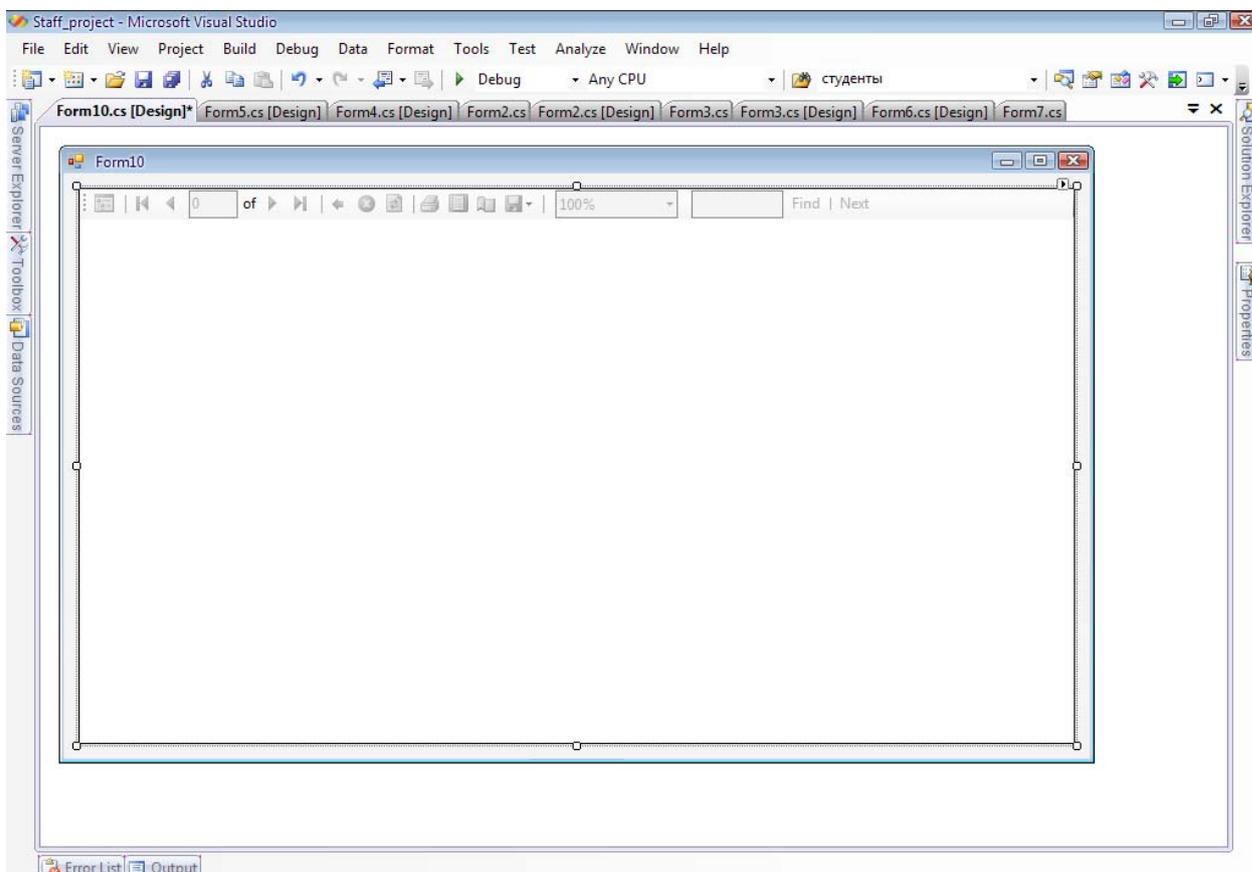


Рис 3.12 Добавление к форме компонента *ReportViewer*

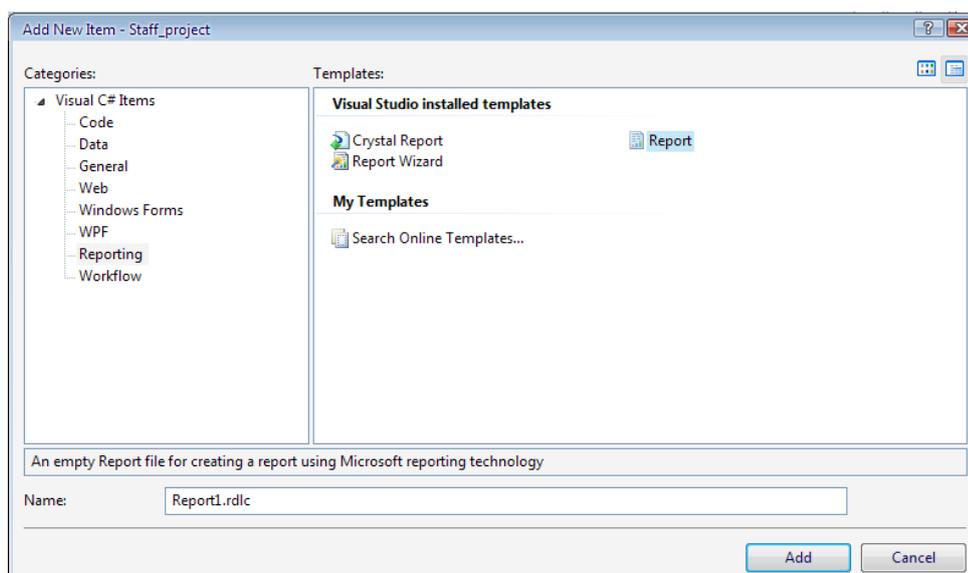


Рис 3.13 Добавление компонента *Report*

После того как основные компоненты построения были добавлены, с помощью мастера необходимо организовать отчет рис 3.13 .

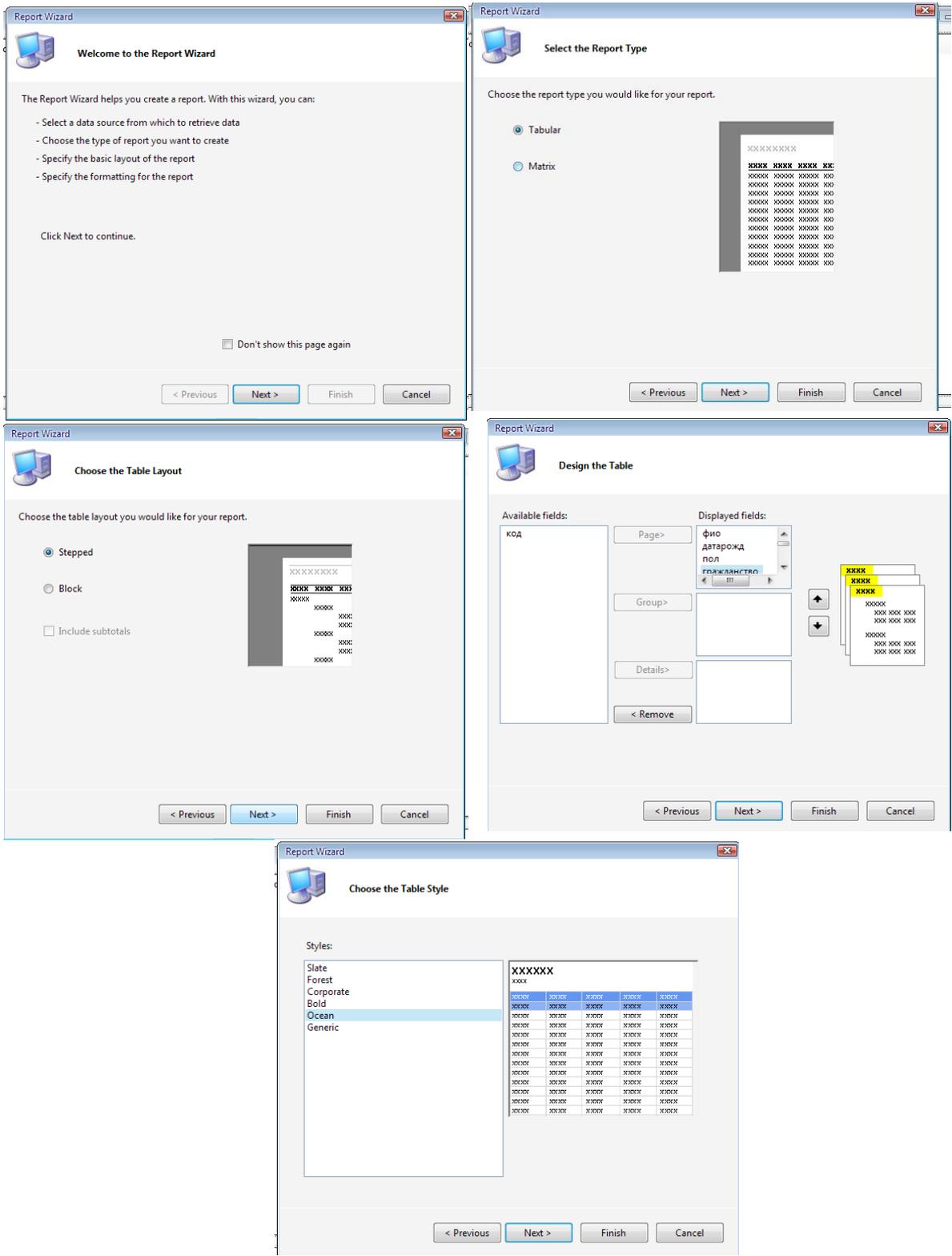


Рис 3.13 Порядок создания отчета

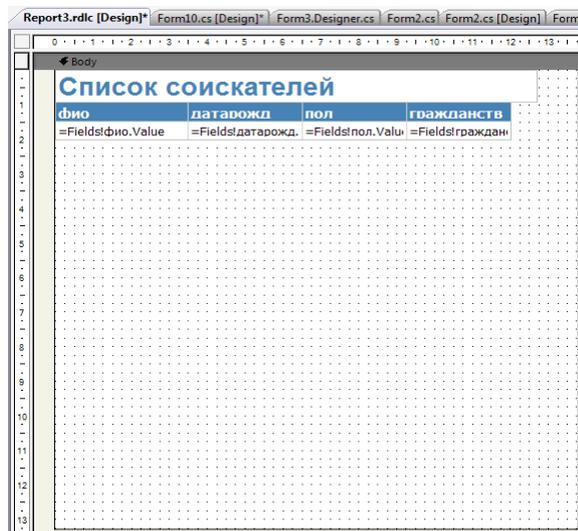


Рис 3.13 Дизайнер шаблона

После работы мастера открывается дизайнер шаблона, который позволяет дополнять и изменять печатную форму рис 3.13. Результат при загрузке выглядит следующим образом рис 3.14.

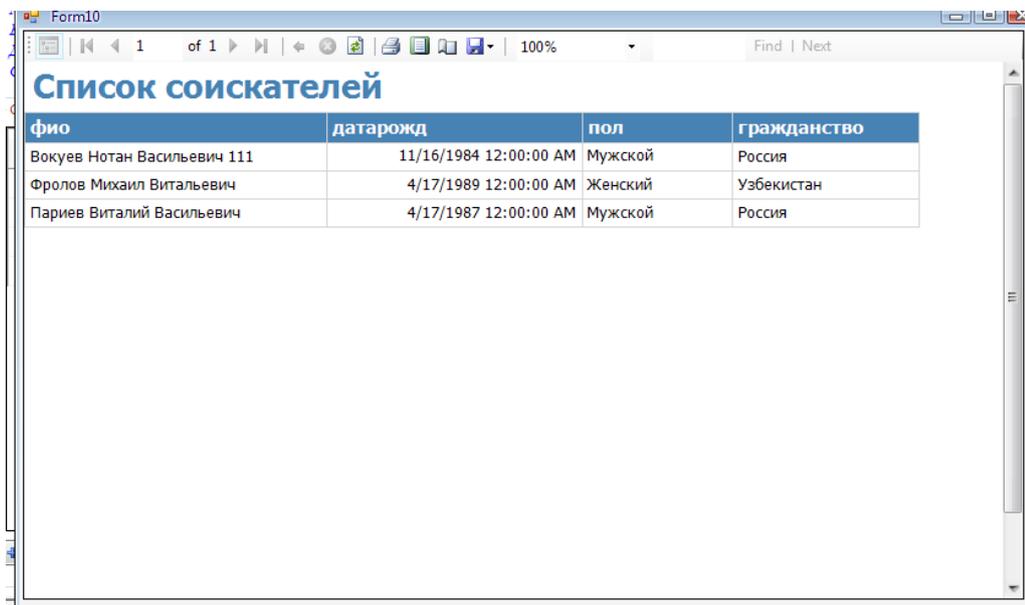


Рис 3.14 Печатная форма

3.4. Описание пользовательского интерфейса

Работа программы начинается с главной формы, которая представляет собой форму аутентификации и подключения к базе данных. Здесь указывается название сервера, пользователь и пароль рис 3.4.1.

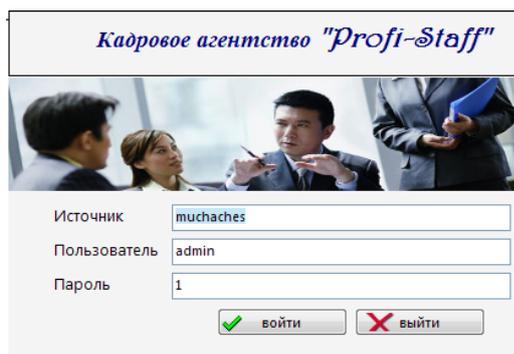


Рис 3.4.1 Стартовое окно программы

После загрузки открывается главное рабочее окно, которое отображает категории объектов и основной функционал рис.3.4.2.

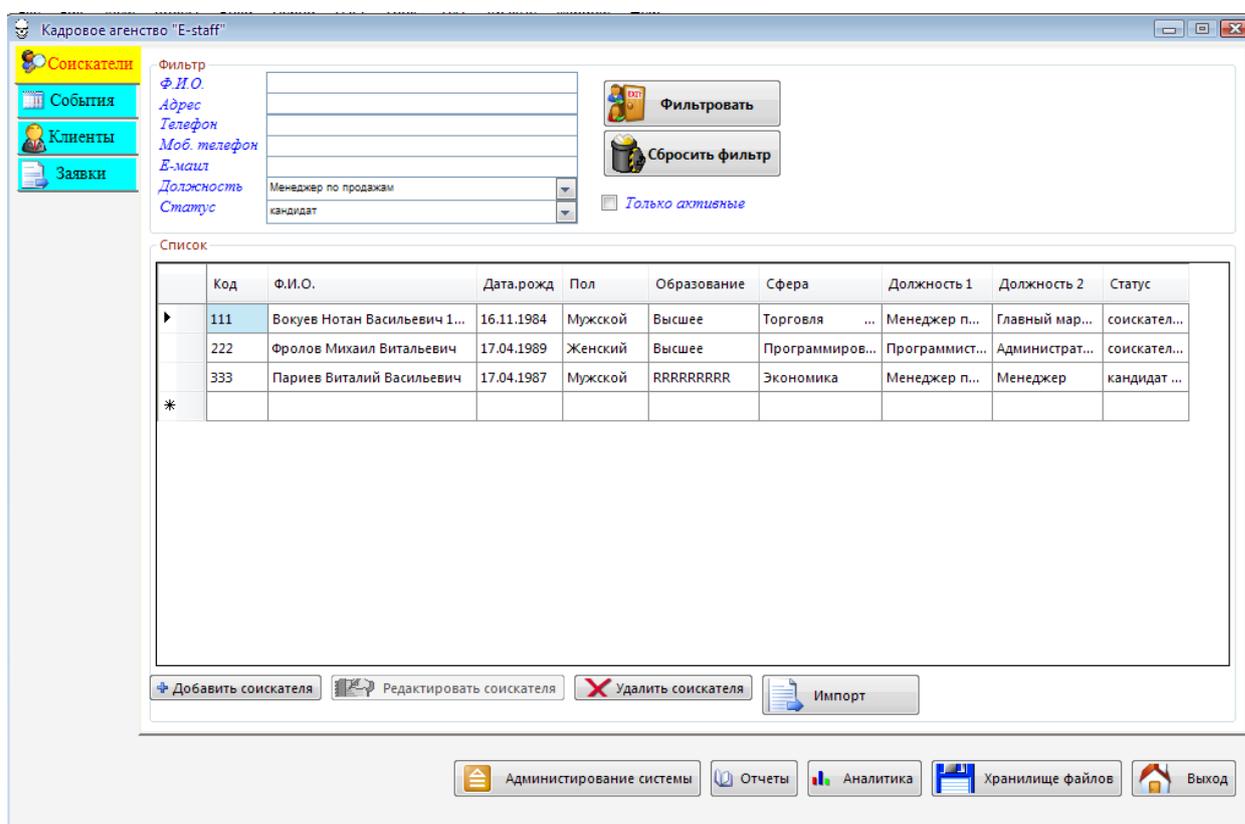


Рис.3.4.2.Главное рабочее окно

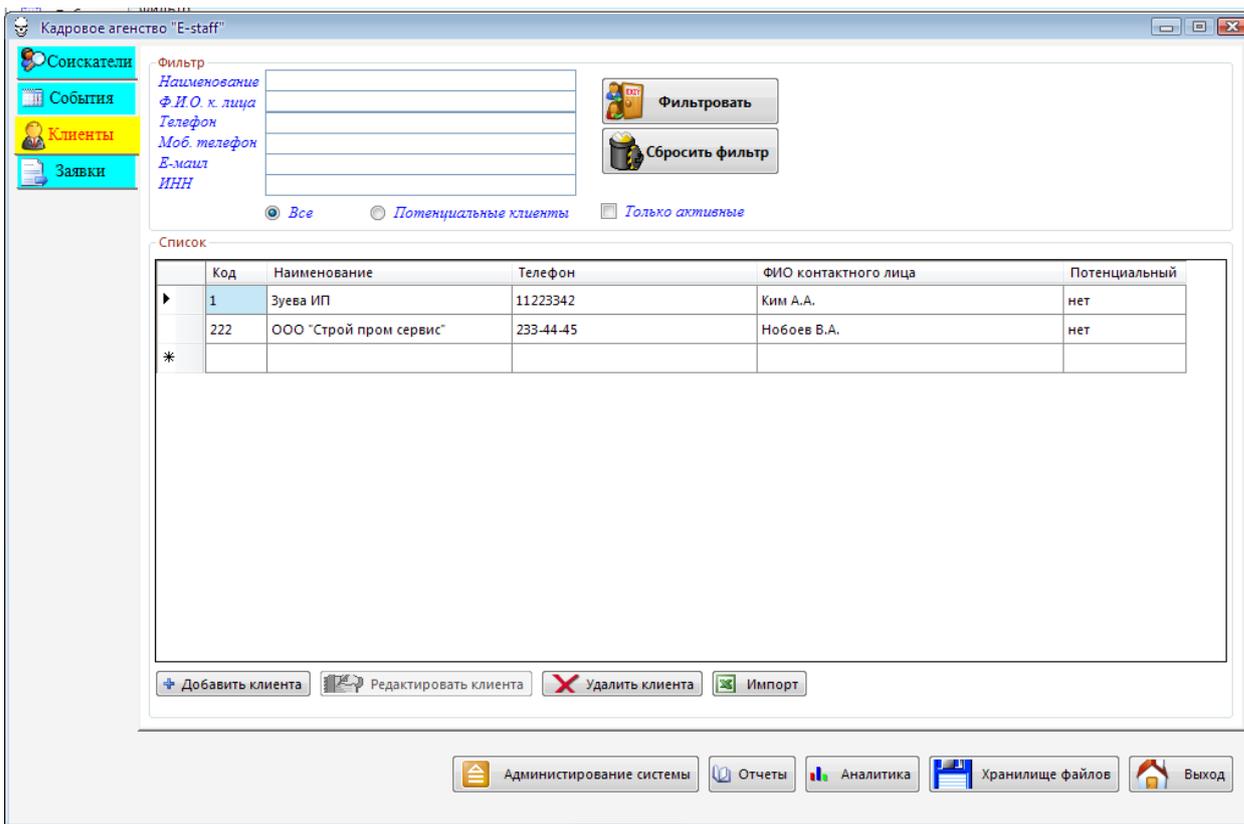


Рис.3.4.3.Главное рабочее окно

Для добавления нового объекта необходимо выбрать одну из вкладок и нажать кнопку «Добавить...» после чего открывается форма для работы с данными причем в ряде форм необходимо указать код объекта иначе другие функции будут не доступны рис 3.4.4.

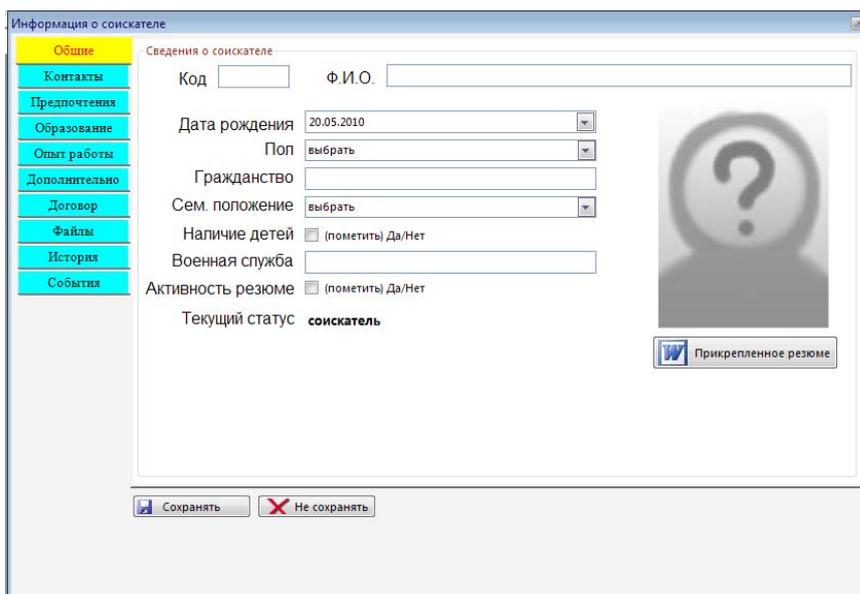


Рис 3.4.4 (а) Добавление данных о соискателе

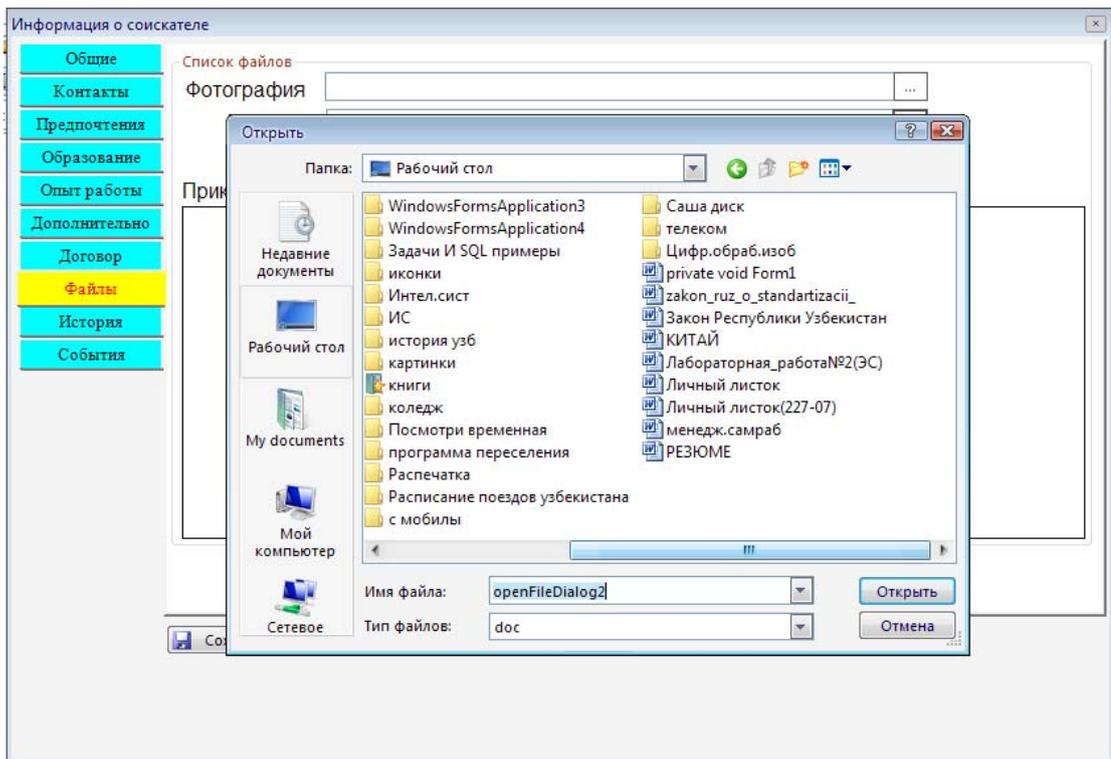


Рис 3.4.4 (б) Добавление данных о соискателе

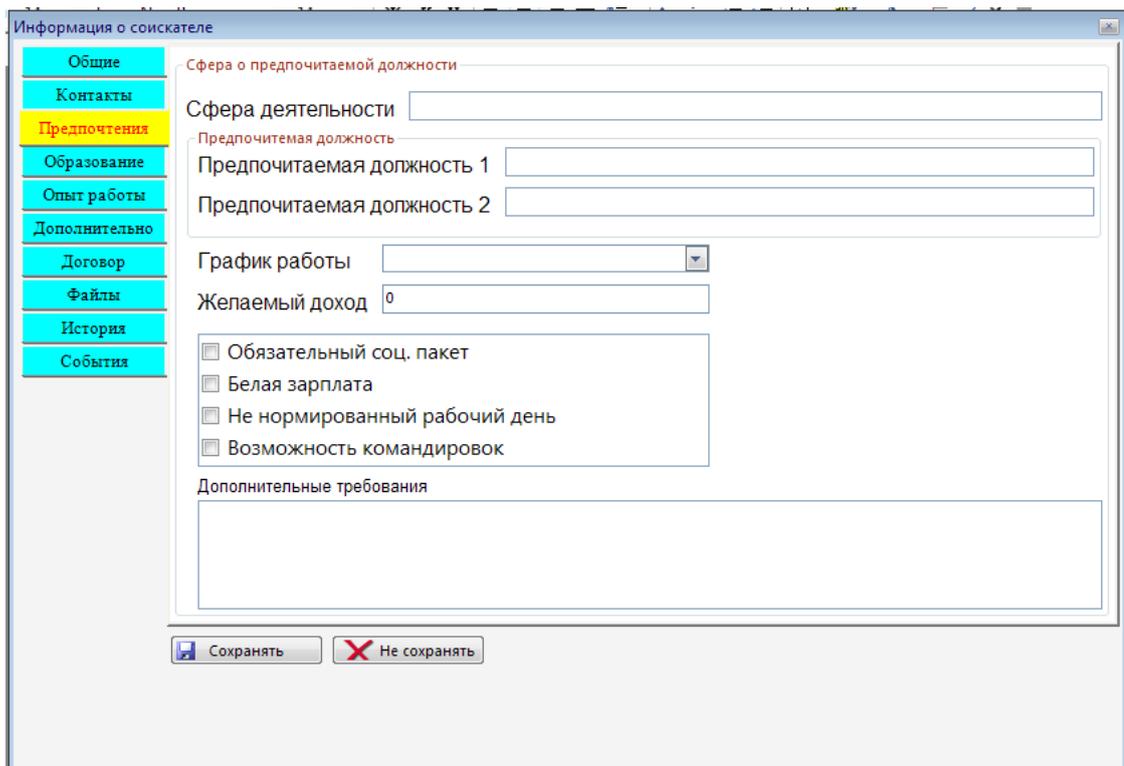


Рис 3.4.4 (в) Добавление данных о соискателе

Клиент

Общие Адрес и контакты Реквизиты Дополнительно События Заявки

Общая информация о клиенте

Код

Имя

Вид деятельности

Информация о предприятии

Активен

Потенциальный клиент

Рис 3.4.4 (г) Добавление данных о клиенте

Событие соискателя

Тип события

План label9 Факт

Дата и время события

Дата 20.05.2010 Время 23:18:44

Вид события и контактная информация

Вид события

Контрагент Вокуев Нотан Васильевич 111

Способ связи

Содержание и результаты

Тема

Результат

Пользователь

Рис 3.4.4 (д) События клиенту

Для редактирования данных какого либо объекта необходимо выбрать запись в главной форме и нажать на кнопку «Редактировать ...» после чего откроется форма с загруженными для редактирования данными рис 3.4.5.

The screenshot shows a window titled "Информация о соискателе" (Information about the applicant). On the left is a vertical menu with tabs: "Общие" (General), "Контакты" (Contacts), "Предпочтения" (Preferences), "Образование" (Education), "Опыт работы" (Work Experience), "Дополнительно" (Additional), "Договор" (Contract), "Файлы" (Files), "История" (History), and "События" (Events). The "Общие" tab is selected. The main area is titled "Сведения о соискателе" (Applicant Information) and contains the following fields:

- Код (Code): 111
- Ф.И.О. (Full Name): Вокуюев Нотан Васильевич 111
- Дата рождения (Date of Birth): 16.11.1984
- Пол (Gender): Мужской (Male)
- Гражданство (Citizenship): Россия (Russia)
- Сем. положение (Marital Status): женат/замужем (Married)
- Наличие детей (Children): (позначить) Да/Нет (Mark Yes/No)
- Военная служба (Military Service): Майор воздушных сил в запасе (Major of the Air Force Reserve)
- Активность резюме (Resume Activity): (позначить) Да/Нет (Mark Yes/No)
- Текущий статус (Current Status): соискатель (Applicant)

On the right side, there is a photo of a man and a button labeled "Прикрепленное резюме" (Attached Resume). At the bottom, there are two buttons: "Сохранять" (Save) and "Не сохранять" (Do not save).

Рис 3.4.5(а). Форма для редактирования соискателя

The screenshot shows a window titled "Клиент" (Client). It has several tabs: "Общие" (General), "Адрес и контакты" (Address and contacts), "Реквизиты" (Details), "Дополнительно" (Additional), "События" (Events), and "Заявки" (Requests). The "Общие" tab is selected. The main area is titled "Общая информация о клиенте" (General information about the client) and contains the following fields:

- Код (Code): 1
- Имя (Name): Зуева ИП
- Вид деятельности (Type of activity): Производство и продажа мягкой мебели (Production and sale of soft furniture)
- Информация о предприятии (Company information): Организация основана в 1990 г. Количество персонала более 30 человек. (Organization founded in 1990. Number of staff more than 30 people.)

On the right side, there are two checkboxes:

- Активен (Active)
- Потенциальный клиент (Potential client)

At the bottom, there are two buttons: "Сохранять" (Save) and "Не сохранять" (Do not save).

Рис 3.4.5(б). Форма для редактирования клиента

Также в программе предоставляется возможность фильтрации данных рис 3.4.6. для этого необходимо ввести искомые данные в область «Фильтр» и нажать кнопку «Фильтровать»

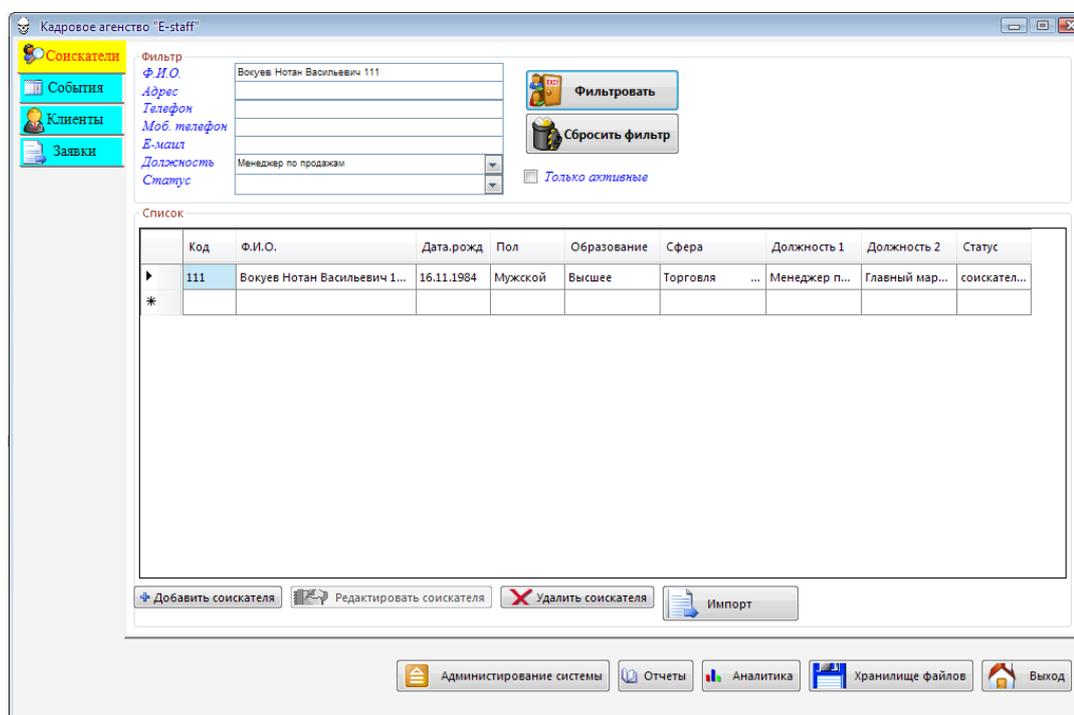


Рис 3.4.6. Фильтрация по выбранным полям

Для того чтобы организовать процедуру переноса данных из файлов других программ не обходимо воспользоваться кнопкой «Импорт», после чего появится форма в которой нужно выбрать нужный файл и указать ячейки с данными для импорта Рис 3.4.7

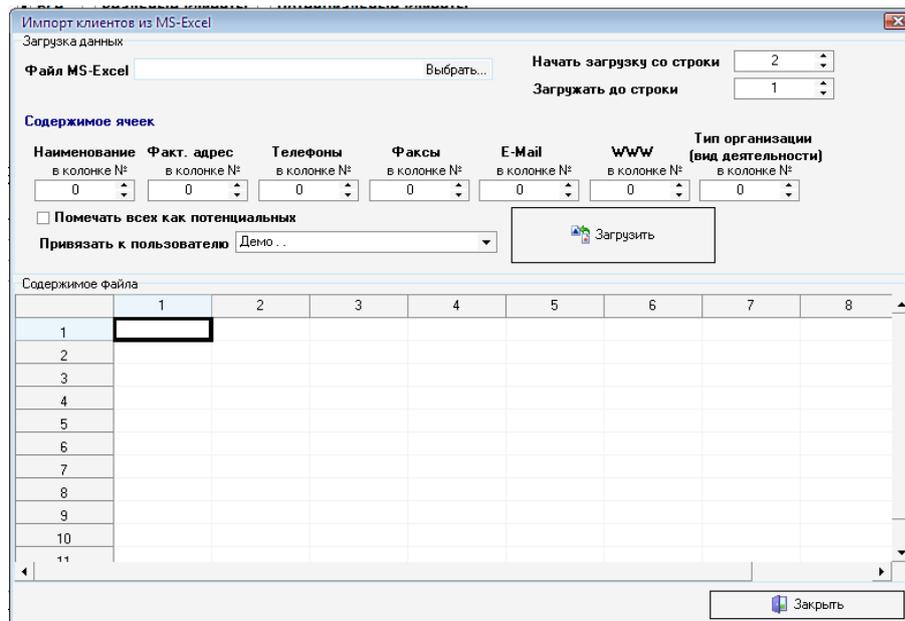


Рис 3.4.7. Форма импорта данных из внешних файлов

Для создания отчетов необходимо нажать кнопку «Отчеты» и выбрать один из вариантов создания отчетов рис. 3.4.8. далее откроется печатная форма которая позволяет напечатать документ

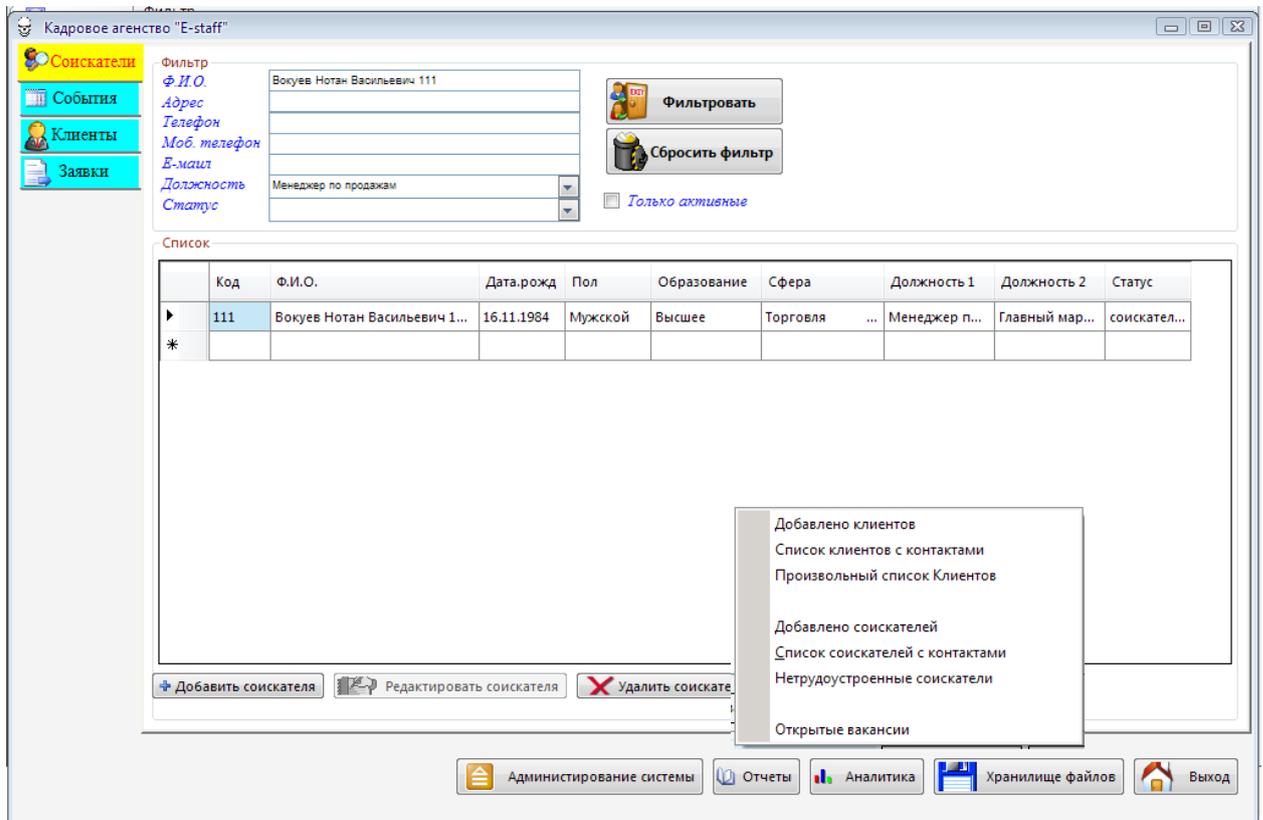


Рис. 3.4.8.(а)Выбор отчета

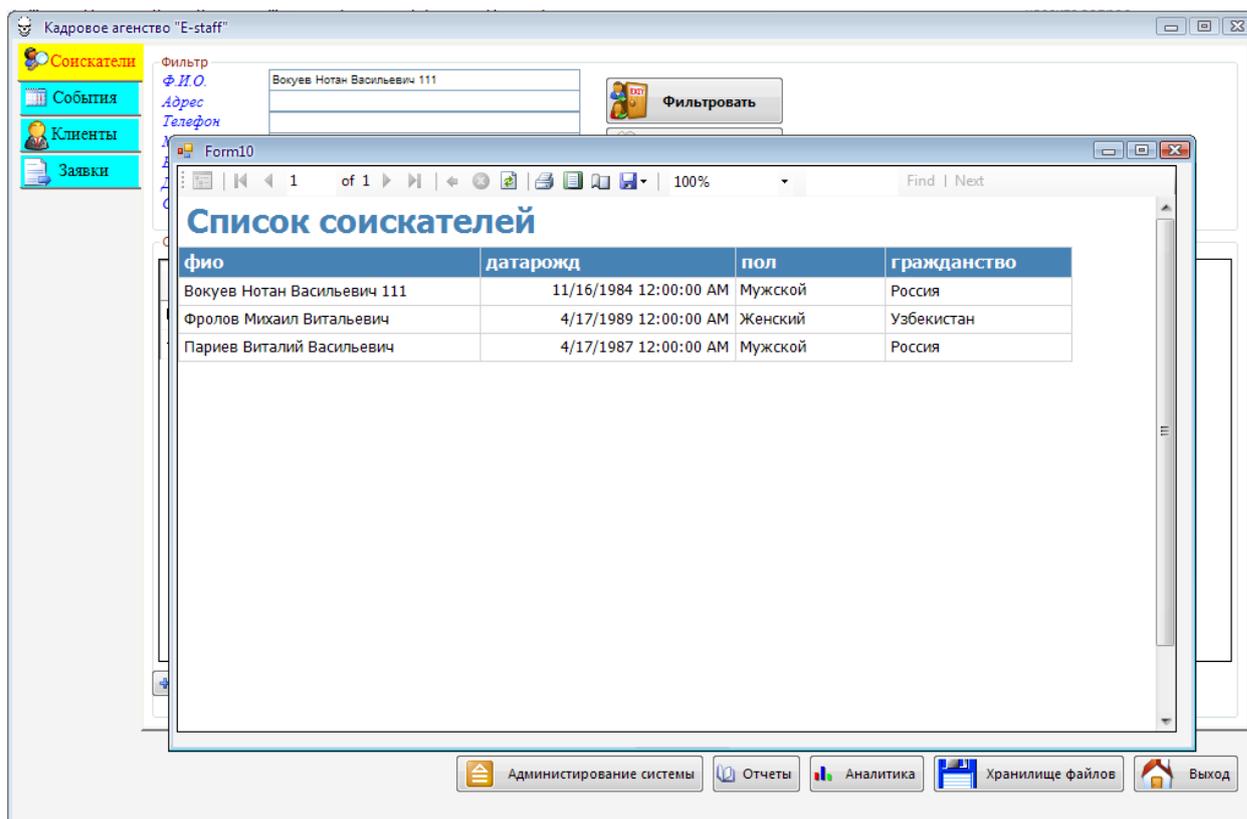
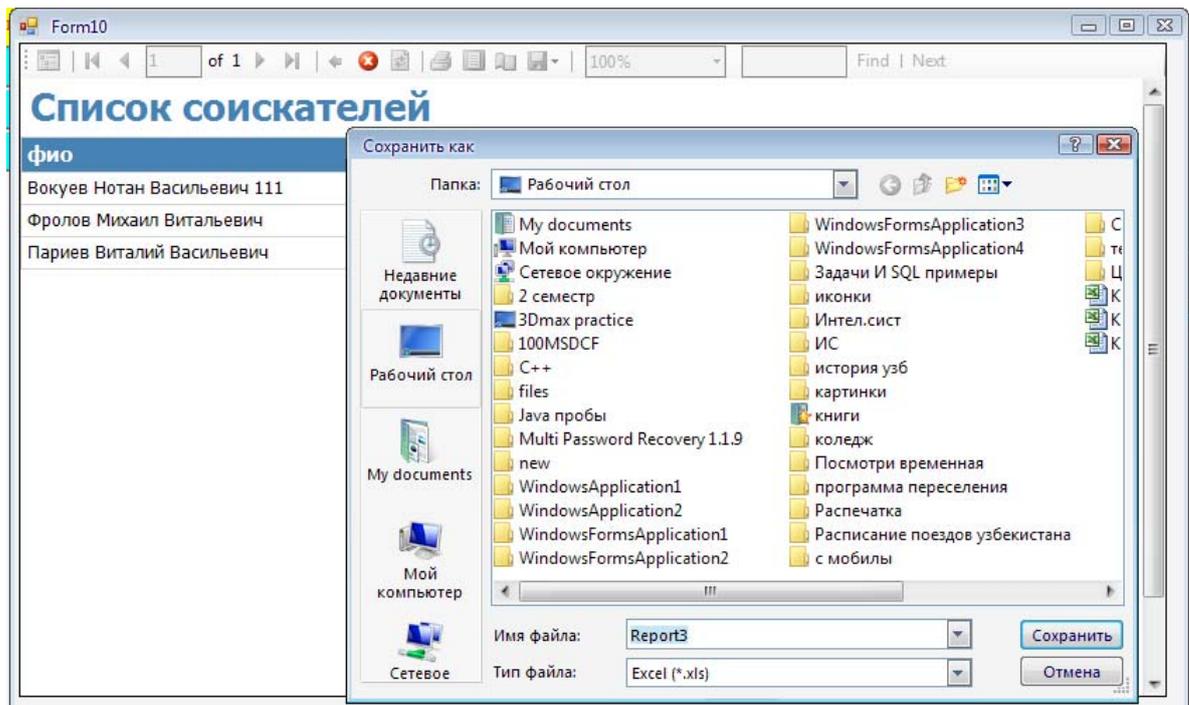


Рис. 3.4.8.(б) Печатная форма

Печатная форма обладает необходимыми функциями для работы с печатью, а именно настройка принтера, и параметров страницы, также экспорта в MS Excel и Acrobat Reader рис 3.4.9.



The screenshot shows Microsoft Excel with the data from the web page. The table is located in the range A1:J5 and contains the following information:

фио	датарожд	пол	гражданство
Вокуев Нотан Васильевич 111	11.16.1984 12:00:00 AM	Мужской	Россия
Фролов Михаил Витальевич	4.17.1989 12:00:00 AM	Женский	Узбекистан
Париев Виталий Васильевич	4.17.1987 12:00:00 AM	Мужской	Россия

The screenshot shows Adobe Reader displaying the printed version of the data table. The table is titled 'Список соискателей' and contains the following information:

фио	датарожд	пол
Вокуев Нотан Васильевич 111	11/16/1984 12:00:00 AM	Мужской
Фролов Михаил Витальевич	4/17/1989 12:00:00 AM	Женский
Париев Виталий Васильевич	4/17/1987 12:00:00 AM	Мужской

3.4.9. Функции печатной формы

Выводы к третьей главе.

В данной главе были рассмотрены следующие вопросы:

- Описана реализация основных программных модулей и описаны основные компоненты, которые применялись при проектировании системы.
- Били представлены функции экспорта и импорта из MS Excel, подробно описано создание отчетов и их применение .
- Было дано подробное описание работы с информационной системой пользователю.

4. БЕЗОПАСНОСТЬ ЖИЗНЕДЕЯТЕЛЬНОСТИ

4.1. Производственный микроклимат

Нормы производственного микроклимата определяют оптимальные условия для рабочей зоны и нормируются. Контроль параметров воздушной среды осуществляется соответствующими специалистами с помощью следующих приборов:

- ✓ термометр (температура воздуха);
- ✓ психрометр (относительная влажность);
- ✓ анемометр (скорость движения воздуха);
- ✓ актинометр (интенсивность теплового излучения);
- ✓ газоанализатор (концентрация вредных веществ).

Для рассматриваемых помещений характерны следующие показатели:

1. Комната системного администратора:

- ✓ температура воздуха 16-18 С - в холодное время года, 18-20 - в теплое;
- ✓ относительная влажность воздуха 40-60% - в холодное время года, 55-75% - в теплое;
- ✓ скорость движения воздуха 0.2 м/с.

2. Кабинет директора предприятия:

- ✓ температура воздуха 18-20 С - в холодное время года, 20-22 - в теплое;
- ✓ относительная влажность воздуха 40-60% - в холодное время года,
- ✓ 55-75%
- ✓ в теплое;
- ✓ скорость движения воздуха 0.2 м/с.

3. Основные производственные помещения, в которых находятся операторы ПЭВМ однотипны и обладают следующими климатическими показателями:

✓ температура воздуха 20-22 С - в холодное время года, 22-24 - в теплое;

✓ относительная влажность воздуха 40-60% - в холодное время года, 55-75% - в теплое;

✓ скорость движения воздуха 0.2 м/с.

С целью создания комфортных условий труда, для поддержания влажности и оптимальной температуры в помещениях установлены кондиционеры («Electra», вентиляция, охлаждение, увлажнение воздуха). Система кондиционирования воздуха обеспечивает поддержание необходимых параметров микроклимата, осуществляет очистку воздуха от пыли и вредных веществ.

Производственное освещение

Освещение служит одним из важнейших факторов влияющих на благоприятные условия труда. Рационально устроенное освещение на рабочих местах работников, обеспечивает высокий уровень работоспособности и оказывает положительное психологическое воздействие на работающих, способствует повышению производительности труда

Вся информация подается через зрительный анализатор. Вред. воздействие на глаза человека оказывают следующие опасные и вред. производственные факторы:

- ✓ Недостаточное освещение рабочей зоны;
- ✓ Отсутствие/недостаток естественного света;
- ✓ Повышенная яркость;
- ✓ Перенапряжение анализаторов (в т.ч. зрительных)
- ✓ По данным ВОЗ на зрение влияет:
- ✓ УФИ;
- ✓ яркий видимый свет;
- ✓ мерцание;
- ✓ блики и отраженный свет.

Воздействие шума

Шум - всякий нежелательный для человека звук, мешающий восприятию полезных сигналов.

Допустимые уровни звукового давления, уровни звука и эквивалентные им уровни шума на рабочем месте не должны превышать 20 дБ, что является областью звукового комфорта.

Источниками шума в рассматриваемых помещениях являются кондиционер, компьютер, принтер. Уровень шума - порядка 20 дБ, что не превышает допустимых уровней.

Внешний шум и вибрации в рассматриваемом помещении отсутствуют практически полностью, так как отделка выполнена с учетом требований звукоизоляции.

Электромагнитные излучения

Мониторы являются основным источником различных видов излучений (электромагнитного, ионизирующего, неионизирующего) и статического электричества. Электронно-лучевая трубка (ЭЛТ) монитора является потенциальным источником рентгеновского излучения.

В данном случае на рабочем месте установлены ПЭВМ типа IBM-PC с монитором LG Flatron типа SVGA со следующими техническими характеристиками:

- ✓ Размер экрана 17 дюймов;
- ✓ Величина зерна (dot/pitch) трубки по горизонтали, т. е. минимальный размер точки на экране монитора входит в норму от 0,22 до 0,41 мм;
- ✓ Частота регенерации изображения входит в норму от 75 Гц;
- ✓ Максимальное разрешение монитора для 17 дюймового монитора составляет 1268 на 1024 точек;
- ✓ монитор удовлетворяет стандарту TCO 92.95.99 и MPR – II.

Тем не менее, в течение рабочего дня необходимо равномерно распределять и чередовать различную по степени напряженности нагрузку (ввод данных, редактирование программ, печать документов или чтение

информации с экрана). При этом непрерывная работа за монитором не должна превышать четырех часов при 8 часовом рабочем дне, а количество обрабатываемых символов (знаков) 30 тыс. за 4 часа работы.

Таким образом, при использовании вышеуказанной аппаратуры и соблюдении изложенных требований условия работы за дисплеем выполнены в соответствии с основными требованиями санитарных норм и правил.

4.2. Пожарная безопасность

Возникновение пожара.

Пожар - это горение вне специального очага, которое не контролируется и может привести к массовому поражению и гибели людей, а также к нанесению экологического, материального и другого вреда.

Горение - это химическая реакция окисления, сопровождающаяся выделением теплоты и света. Для возникновения горения требуется наличие трех факторов: горючего вещества, окислителя и источника загорания. Окислителями могут быть кислород, хлор, фтор, бром, йод, окиси азота и другие. Кроме того, необходимо чтобы горючее вещество было нагрето до определенной температуры и находилось в определенном количественном соотношении с окислителем, а источник загорания имел определенную энергию.

Наибольшая скорость горения наблюдается в чистом кислороде. При уменьшении содержания кислорода в воздухе горение прекращается. Горение при достаточной и надмерной концентрации окислителя называется полным, а при его нехватке - неполным.

Выделяют три основных вида самоускорения химической реакции при горении: тепловой, цепной и цепочно-тепловой. Тепловой механизм связан с экзотермичностью процесса окисления и возрастанием скорости химической реакции с повышением температуры. Цепное ускорение реакции связано с катализом превращений, которое осуществляют промежуточные продукты

превращений. Реальные процессы горения осуществляются, как правило, по комбинированному (цепочно-тепловой) механизму.

Процесс возникновения горения подразделяется на несколько видов.

Вспышка - быстрое сгорание горючей смеси, не сопровождающееся образованием сжатых газов.

Возгорание - возникновение горения под воздействием источника зажигания.

Воспламенение - возгорание, сопровождающееся появлением пламени.

Самовозгорание - явление резкого увеличения скорости экзотермических реакций, приводящее к возникновению горения вещества при отсутствии источника зажигания.

Самовоспламенение - самовозгорание, сопровождается появлением пламени.

Взрыв - чрезвычайно быстрое (взрывчатое) превращение, сопровождающееся выделением энергии с образованием сжатых газов.

Основными показателями пожарной опасности являются температура самовоспламенения и концентрационные пределы воспламенения.

Температура самовоспламенения характеризует минимальную температуру вещества, при которой происходит резкое увеличение скорости экзотермических реакций, заканчивающееся возникновением пламенного горения.

Температура вспышки - самая низкая (в условиях специальных испытаний) температура горючего вещества, при которой над поверхностью образуются пары и газы, способные вспыхивать в воздухе от источника зажигания, но скорость их образования еще недостаточна для последующего горения.

Горючими называются вещества, способные самостоятельно гореть после изъятия источника загорания.

По степени горючести вещества делятся на: горючие (сгораемые), трудногорючие (трудносгораемые) и негорючие (несгораемые).

К трудногорючим относятся такие вещества, которые не способны распространять пламя и горят лишь в месте воздействия источника зажигания.

Негорючими являются вещества, не воспламеняющиеся даже при воздействии достаточно мощных источников зажигания (импульсов).

Горючие вещества могут быть в трех агрегатных состояниях: жидком, твердом и газообразном. Большинство горючих веществ независимо от агрегатного состояния при нагревании образует газообразные продукты, которые при смешении с воздухом, содержащим определенное количество кислорода, образуют горючую среду. Горючая среда может образоваться при тонкодисперсном распылении твердых и жидких веществ.

Из горючих газов и пыли образуются горючие смеси при любой температуре, в то время как твердые вещества и жидкости могут образовать горючие смеси только при определенных температурах.

В производственных условиях может иметь место образование смесей горючих газов или паров в любых количественных соотношениях. Однако взрывоопасными эти смеси могут быть только тогда, когда концентрация горючего газа или пара находится между границами воспламеняемых концентраций.

Минимальная концентрация горючих газов и паров в воздухе, при которой они способны загораться и распространять пламя, называемое *нижним концентрационным пределом воспламенения*.

Максимальная концентрация горючих газов и паров, при которой еще возможно распространение пламени, называется *верхним концентрационным пределом воспламенения*.

Указанные пределы зависят от температуры газов и паров: при увеличении температуры на 100°C величины нижних пределов

воспламенения уменьшаются на 8 -10 %, верхних - увеличиваются на 12 - 15 %.

Пожарная опасность вещества тем больше, чем ниже нижний и выше верхний пределы воспламенения и чем ниже температура самовоспламенения.

Пыли горючих и некоторых не горючих веществ (например алюминий, цинк) могут в смеси с воздухом образовать горючие концентрации.

Наибольшую опасность по взрыву представляет взвешенная в воздухе пыль. Однако и осевшая на конструкциях пыль представляет опасность не только с точки зрения возникновения пожара, но и вторичного взрыва, вызываемого в результате взвихривания пыли при первичном взрыве.

Минимальная концентрация пыли в воздухе, при которой происходит ее загорание, называется *нижним пределом воспламенения пыли* .

Поскольку достижение очень больших концентраций пыли во взвешенном состоянии практически нереально, термин "верхний предел воспламенения" к пылям не применяется.

Воспламенение жидкости может произойти только в том случае, если над ее поверхностью имеется смесь паров с воздухом в определенном количественном соотношении, соответствующим нижнему температурному пределу воспламенения.

Меры по пожарной профилактике.

Мероприятия по пожарной профилактике разделяются на организационные, технические, режимные и эксплуатационные.

Организационные мероприятия: предусматривают правильную эксплуатацию машин и внутризаводского транспорта, правильное содержание зданий, территории, противопожарный инструктаж и тому подобное.

Технические мероприятия: соблюдение противопожарных правил и норм при проектировании зданий, при устройстве электропроводов и

оборудования, отопления, вентиляции, освещения, правильное размещение оборудования.

Режимные мероприятия - запрещение курения в неустановленных местах, запрещение сварочных и других огневых работ в пожароопасных помещениях и тому подобное.

Эксплуатационные мероприятия – своевременная профилактика, осмотры, ремонты и практика тушения пожаров наибольшее распространение получили следующие принципы прекращения горения:

1. изоляция очага горения от воздуха или снижение концентрации кислорода путем разбавления воздуха негорючими газами (углеводороды $CO_i < 12 - 14 \%$).
2. охлаждение очага горения ниже определенных температур;
3. интенсивное торможение (ингибирование) скорости химической реакции в пламени;
4. механический срыв пламени струей газа или воды;
5. создание условий огнепреграждения (условий, когда пламя распространяется через узкие каналы).

Средства тушения пожара.

Вещества, которые создают условия при которых прекращается горение называются огнегасящими. Они должны быть дешевыми и безопасными в эксплуатации не приносить вреда материалам и объектам.

Вода является хорошим огнегасящим средством, обладающим следующими достоинствами: охлаждающее действие, разбавление горючей смеси паром (при испарении воды ее объем увеличивается в 1700 раз), механическое воздействие на пламя, доступность и низкая стоимость, химическая нейтральность.

Недостатки: нефтепродукты всплывают и продолжают гореть на поверхности воды; вода обладает высокой электропроводностью, поэтому ее нельзя применять для тушения пожаров на электроустановках под напряжением.

Тушение пожаров водой производят установками водяного пожаротушения, пожарными автомашинами и водяными стволами. Для подачи воды в эти установки используют водопроводы.

К установкам водяного пожаротушения относят спринклерные и дренчерные установки.

Спринклерная установка представляет собой разветвленную систему труб, заполненную водой и оборудованную спринклерными головками. Выходные отверстия спринклерных головок закрываются легкоплавкими замками, которые расплавляются при воздействии определенных температур (345, 366, 414 и 455 К). Вода из системы под давлением выходит из отверстия головки и орошает конструкции помещения и оборудование.

Дренчерные установки представляют собой систему трубопроводов, на которых расположены специальные головки - дренчеры с открытыми выходными отверстиями диаметром 8, 10 и 12, 7 мм лопастного или розеточного типа, рассчитанные на орошение до 12 м² площади пола.

Дренчерные установки могут быть ручного и автоматического действия. После приведения в действие вода заполняет систему и выливается через отверстия в дренчерных головках.

Пар применяют в условиях ограниченного воздухообмена, а также в закрытых помещениях с наиболее опасными технологическими процессами. Гашение пожара паром осуществляется за счет изоляции поверхности горения от окружающей среды. При гашении необходимо создать концентрацию пара приблизительно 35 %.

Пены применяют для тушения твердых и жидких веществ, не вступающих во взаимодействие с водой. Огнегасящий эффект при этом достигается за счет изоляции поверхности горючего вещества от окружающего воздуха. Огнетушащие свойства пены определяются ее кратностью - отношением объема пены к объему ее жидкой фазы,

стойкостью дисперсностью, вязкостью. В зависимости от способа получения пены делят на химические и воздушно-механические.

Химическая пена образуется при взаимодействии растворов кислот и щелочей в присутствии пенообразующего вещества и представляет собой концентрированную эмульсию двуокиси углерода в водном реакторе минеральных солей. Применение химических солей сложно и дорого, поэтому их применение сокращается.

Воздушно-механическую пену низкой (до 20), средней (до 200) и высокой (свыше 200) кратности получают с помощью специальной аппаратуры и пенообразователей ПО-1, ПО-1Д, ПО-6К и т. д.

Инертные газообразные разбавители : двуокись углерода, азот, дымовые и отработавшие газы, пар, аргон и другие.

Ингибиторы - на основе предельных углеводородов, в которых один или несколько атомов водорода замещены атомами галлоидов (фтор, хлор, бром). Галоидоуглеводороды плохо растворяются в воде, но хорошо смешиваются со многими органическими веществами:

- ✓ тетрафтордибромэтан (хладон 114В2),
- ✓ бромистый метилен
- ✓ трифторбромметан (хладон 13В1)
- ✓ 3, 5, 7, 4НД, СЖБ, БФ (на основе бромистого этила)

Порошковые составы несмотря на их высокую стоимость, сложность в эксплуатации и хранении, широко применяют для прекращения горения твердых, жидких и газообразных горючих материалов. Они являются единственным средством гашения пожаров щелочных металлов и металлоорганических соединений. Для гашения пожаров используется также песок, грунт, флюсы. Порошковые составы не обладают электропроводимостью, не корродируют металлы и практически не токсичны. Широко используются составы на основе карбонатов и бикарбонатов натрия и калия.

Аппараты пожаротушения: передвижные (пожарные автомобили), стационарные установки, огнетушители.

Автомобили предназначены для изготовления огнегасящих веществ, используются для ликвидации пожаров на значительном расстоянии от их дислокации и подразделяются на :

- ✓ автоцистерны (вода, воздушно-механическая пена) АЦ - 40 2, 1 - 5м³ воды;
- ✓ специальные - АП - 3, порошок ПС и ПСБ - 3 3, 2т.
- ✓ аэродромные ; вода, хладон.

Стационарные установки предназначены для тушения пожаров в начальной стадии их возникновения без участия человека. Подразделяются на водяные, пенные, газовые, порошковые, паровые. Могут быть автоматическими и ручными с дистанционным управлением.

Огнетушители - устройства для гашения пожаров огнегасящим веществом, которое он выпускает после приведения его в действие, используется для ликвидации небольших пожаров. Как огнетушащие вещества в них используют химическую или воздухомеханическую пену, диоксид углерода (жидком состоянии), аэрозоли и порошки, в состав которых входит бром.

Подразделяются: по подвижности:

- ✓ ручные до 10 литров
- ✓ передвижные
- ✓ стационарные
- ✓ по огнетушащему составу:
 - ✓ - жидкостные; (заряд состоит из воды или воды с добавками)
 - ✓ углекислотные; (СО₂)
 - ✓ химпенные (водные растворы кислот и щелочей)
 - ✓ воздушно-пенные;
 - ✓ хладоновые; (хладоны 114В2 и 13В1)
 - ✓ порошковые; (ПС, ПСБ-3, ПФ, П-1А, СИ-2)

- ✓ комбинированные

Огнетушители маркируются буквами (вид огнетушителя по разряду) и цифровой (объем). Ручной пожарный инструмент - это инструмент для раскрывания и разбирания конструкций и проведения аварийно-спасательных работ при гашении пожара. К ним относятся : крюки, ломы, топоры, ведра, лопаты, ножницы для резания металла. Инструмент размещается на видном и доступном месте на стендах и щитах.

Средства оповещения. Сигнализация.

К системам сигнализации предъявляются следующие технические требования: они должны иметь минимальную инерционность сработки, обеспечивать заданную достоверность информации, отсутствие ошибочной сработки; быть надежными в работе при всех условиях эксплуатации, обеспечивать автономное включение сигнала тревоги.

Основными элементами пожарной сигнализации являются:

- ✓ датчики пожарной сигнализации, которые размещаются в наиболее пожаро- и взрывоопасных местах;
- ✓ электронно-усилительный блок, который обеспечивает дистанционный контроль за состоянием датчиков;
- ✓ исполнительный блок, с помощью которого включается первый рубеж противопожарной системы и блок сигнализации.

Датчики - наиболее важный элемент системы сигнализации, который в основном определяет возможности и характеристики системы в целом. В зависимости от физической сути, заложенной в основу работы датчика, системы подразделяются на: тепловые, ионизационные, радиационные и т.п. Тепловые системы реагируют на повышение температуры либо стенок конструкции, либо окружающей среды, ионизационные и радиационные срабатывают при наличии огня, принцип их работы основан на том, что под влиянием высокой температуры ионизируются продукты горения, а также приблизительно 20 % всей энергии - излучение.

ЗАКЛЮЧЕНИЕ.

Результатом представленной дипломной работы стало разработка информационной системы «Profi Staff» рекрутинговое агентство. В данной программе, вся работа кадрового агентства осуществляется в рамках этой программы: ввод первичных данных о клиентах, обслуживание и корректировка необходимой информации, поиск по необходимым параметрам, а также вывода на печать списков вакансий и работников, экспорт и импорт в табличные и текстовые редакторы.

Эффективное практическое применение информационная система получит при работе связанной с обслуживанием клиентов, оперативностью выдачи данных. При ручном учете этой задачи затрачивается огромное количество времени на принятие и оформление документов, поиск необходимой вакансии, а также составление списков, которые необходимы для агентства в виде прайс-листа.

Автоматизация этой задачи позволит не только устранить выше перечисленные задачи, но также позволит менеджеру отслеживать выполненные обязательства по договорам, получить эффективность, которая в конечном итоге положительно влияет на получение прибыли предприятия, увеличение производительности труда менеджера по персоналу.

Программа автоматизации учета работы кадрового агентства позволяет менеджеру по кадрам работать без дополнительного обучения, поскольку интерфейс довольно прост, и рассчитан на пользователя знающего только основные офисные программы компьютера.

Многие кадровые агентства используют ручной труд менеджера по персоналу, поскольку работа в данном направлении является новой и не изученной сферой деятельности. Автоматизация учета позволяет повысить уровень обслуживания клиентов, снизить затраты на работу кадрового агентства, сэкономить время как менеджера по персоналу, так и работника, ускоряет процесс работы и дает возможность обслужить большее количество

клиентов. Применение данной программы значительно повышает эффективность, производительность труда, снижает уровень безработицы.

Поскольку многие кадровые агентства не используют автоматизированный учет, либо базы ведутся в EXCEL в виде таблиц, то данная программа может быть установлена в других агентствах, не имеющих специализированного программного обеспечения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Ходиев Б.Ю., Бегалов Б.А., Абдуллаев А., Пилипенко Е.Ф., Рузметова Х.Н. Введение в базы данных и знаний. Ташкент, 2003.-293с.
2. Ходиев Б.Ю., Мусалиев А.А., Бегалов Б.А. Введение в информационные системы и технологии, Аккад, 2004.-250с.
3. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений, КОРОНАПринт, 2002.-457с.
4. Шураков В.В. Автоматизированное рабочее место для статической обработки данных, Професионал,1990.-273с.
5. Константайн Л., Локвуд Л. Разработка программного обеспечения. СПб.: Питер, 2004.-345с.
6. Троэлсен Э. С# и платформа .NET». СПб.: Питер, 2004.-450с.
7. Тай Т., Лэм Х. К. Платформа .NET. Основы. Символ, 2003.-500с.
8. Рихтер Д. Программирование на платформе Microsoft .NET Framework. Москва 2003г.-470с.
9. Браст Э. Дж., Форте С. Разработка приложений на основе Microsoft SQL Server 2005. Русская редакция. 2007г.-670с
10. Либерти Д. Создание .NET приложений Программирование на С#. Русская редакция, 2005.-340с.
11. Пол Н. SQL Server 2005 Библия пользователя. Москва 2008г.- 762с.
12. Харрингтон Д. Л. Проектирование реляционных баз данных. Лори.2005.-344с.
13. Малик С.. Microsoft ADO.NET 2.0 ДЛЯ ПРОФЕССИОНАЛОВ Москва. 2006 -322с.
14. Дунаев В. БАЗЫ ДАННЫХ ЯЗЫК SQL ДЛЯ СТУДЕНТА Санкт-Петербург -БХВ-Петербург. 2006-345с
15. Сеппа Д. Microsoft ADO.NET. Москва 2003г.-657с.