Содержание

[Введение 4](#_Toc9599228)

[1 Теоретическая часть 6](#_Toc9599229)

[1.1 Теория применяемых программных средств. Описание синтаксических конструкций 6](#_Toc9599230)

[1.2 Обоснование и выбор используемых средств проектирования информационной системы 7](#_Toc9599231)

[1.3 Принципы и инструменты разработки и программной реализации информационной системы 8](#_Toc9599232)

[1.4 Этапы и методы разработки информационной системы 10](#_Toc9599233)

[2 Аналитическая часть 14](#_Toc9599234)

[2.1 Анализ существующих языковых средств разработки информационной системы 14](#_Toc9599235)

[2.2 Анализ программных средств разработки информационной системы 15](#_Toc9599236)

[2.3 Анализ структуры и принципов функционирования баз данных 17](#_Toc9599237)

[3 Практическая часть 20](#_Toc9599238)

[3.1 Техническое задание на разработку информационной системы 20](#_Toc9599239)

[3.2 Описание процесса разработки информационной системы 23](#_Toc9599240)

[3.3 Описание функциональных возможностей информационной системы 35](#_Toc9599241)

[3.4 Описание защиты информационной системы 40](#_Toc9599242)

[3.5 Расчет затрат на внедрение и сопровождение информационной системы 43](#_Toc9599243)

[3.6 Охрана труда и техника безопасности при организации рабочего места 44](#_Toc9599244)

[Заключение 45](#_Toc9599245)

[Список использованных источников 46](#_Toc9599246)

# Введение

Задача накопления, обработки и распространения (обмена) информации стояла перед человечеством на всех этапах его развития. В течение долгого времени основными инструментами для ее решения были мозг, язык и слух человека. Первое кардинальное изменение произошло с приходом письменности, а затем изобретением книгопечатания. Поскольку в эпоху книгопечатания основным носителем информации стала бумага, то технологию накопления и распространения информации естественно называть «бумажной информатикой».

Положение в корне изменилось с появлением электронных вычислительных машин. Одним из примеров системного применения электронных вычислительных машин в мировой практике были так называемые административные системы обработки данных: автоматизация банковских операций, бухгалтерского учета, резервирования и оформления билетов. Решающее значение для эффективности систем подобного рода имеет то обстоятельство, что системы опираются на автоматизированные информационные базы. Это означает, что в памяти ЭВМ постоянно сохраняется информация, нужная для решения тех задач, на которые рассчитана система. Она и составляет содержимое информационной базы соответствующей системы.

При решении очередной задачи система нуждается во вводе только небольшой порции дополнительной информации, - остальное берется из информационной базы. Каждая порция вновь вводимой информации изменяет информационную базу системы. При переходе от автоматизации отдельных процессов предметной области к созданию автоматизированных информационных систем требуется не только взаимосвязь приложений, но и качественно новый подход к организации данных.

Актуальность исследуемой темы заключается в повсеместном использовании информационных систем на предприятиях различного типа.

Тема дипломного проекта «Разработка информационной системы».

Целью дипломного проекта является разработка информационной системы для учета заказов клиентов.

Для достижения цели был поставлен и решен ряд задач:

- выбран язык разработки;

- рассмотрены среды разработки;

- составлена база данных для информационной системы;

- разработан интерфейс информационной системы;

- разработан сайт для информационной системы как дополнение.

Предметом исследования является автоматизация работы организации, связанной с ведением учета заказов клиентов.

Объектом исследования является информационная система для автоматизации работы по ведению учета заказов клиентов.

Методы исследования:

- эмпирический, основанный на чувственном восприятии, а также на измерении при помощи приборов;

- теоретический, основанный на абстракциях и обобщенности;

- количественный, основан на количественных показателях;

- качественный, направлен на выявление качественных характеристик.

Результатом работы должна стать информационная система, которая должна облегчить работу, связанную с ведением учета заказов клиентов.

# 1 Теоретическая часть

# 1.1 Теория применяемых программных средств. Описание синтаксических конструкций

Во всех языках программирования определены способы организации данных и способы организации действий над данными. Кроме того, существует понятие «элементы языка», включающее в себя множество символов (алфавит), лексемы и другие изобразительные средства языка программирования. Несмотря на разнообразие указанных языков, их изучение происходит приблизительно по одной схеме. Это связано с общностью структуры различных языков программирования высокого уровня.

Синтаксис языка программирования - совокупность правил написания чисел, переменных, выражений, операторов, процедур и других элементов, а также синтаксических конструкций данного языка программирования.

Идентификаторы - имена объектов и конструкций программы (меток, констант, типов, переменных, типов, процедур, функций, объектов, модулей, программ, полей в записях). Имя может иметь любую длину, однако различимыми являются только первые 63 символа. Имя состоит из буквы, за которой могут следовать буквы, цифры или символ подчеркивания.

Константа - это фиксированное значение, которое не может быть изменено программой. Константа может относиться к любому базовому типу. Способ представления константы определяется ее типом.

Переменная - это область памяти, в которой находятся данные, которыми оперирует программа. Когда программа манипулирует с данными, она оперирует содержимым ячеек памяти, то есть переменными.

Тип данных - это описание, определяющее свойства и порядок обработки данных.  Типы данных могут быть различными, от байтов и цифр до массивов и объектов.

Любой современный язык программирования поддерживает различные операторы. Операторы могут быть условными, которые будут использоваться в условных конструкциях, могут быть операторами цикла, используемых для многократного выполнения определенного действия.

Языки программирования также поддерживают множество стандартных методов, которые используются для упрощения написания кода. Методы могут быть математическими, такими как возведение числа в степень, а некоторые методы могут использоваться для работы с объектами, так как все современные языки программирования являются объектно-ориентированными.

Каждый язык программирования имеет свой синтаксис, а значит и синтаксические конструкции, многие языки имеют одинаковые синтаксические конструкции, или же немного измененные, но некоторые языки могут иметь уникальные конструкции, которые могут повторяться очень редко и используются в конкретной предметной области, как правило, такие синтаксические конструкции подключаются из специальных библиотек, называемых фреймворками.

# 1.2 Обоснование и выбор используемых средств проектирования информационной системы

С помощью объектно-ориентированного программирования можно упростить задачу внесения изменений в системную архитектуру.

Язык C# может быть использован для написания кода информационной системы, а также создания интерфейса программы. Для создания приемлемого интерфейса и упрощения организации данного интерфейса используется специальный фреймворк - «MetroFraemwork». Данная библиотека содержит в себе элементы управления, настраивать которые намного проще.

Преимущества выбранного языка программирования описаны ниже.

Язык активно развивается, регулярно выходят новые версии C#, которые добавляют новые синтаксические конструкции в язык, а также увеличивают быстродействие и надежность.

Данный язык использует объектно-ориентированный подход к программированию во всем. Это означает, что нужно описывать абстрактные конструкции на основе предметной области, а потом реализовывать между ними взаимодействие.

Также в языке присутствует множество синтаксических конструкций, которые облегчают работу программиста. Вместо того, чтобы писать много строк кода, используется готовая конструкция, а компилятор выполняет остальную работу.

Одним из самых важных плюсов является наличие большого количества библиотек и шаблонов. Нужное решение скачивается из nuget, после чего решение можно использовать.

Немаловажно наличие хороших инструментов разработки. Абсолютное большинство разработчиков используют интегрированную среду разработки Visual Studio, которая предоставляет много возможностей.

При создании информационных систем также используются базы данных и системы управления базами данных для хранения большого объема данных. Для разработки информационной системы для дипломного проекта использовалась система управления базами данных «phpMyAdmin».

# 1.3 Принципы и инструменты разработки и программной реализации информационной системы

Разработка информационных систем осуществляется на основе основных принципов, которые описаны ниже.

Принцип системности требует рассматривать анализируемое предприятие, как единое целое, выделять типы связи между структурными элементами или бизнес-процессами, устанавливать направления информационных потоков, а также функции отдельных структурных подразделений. В результате все составные части объекта будут объединены и согласованы по целям функционирования по входным и результирующим документам и защите информации.

Принцип развития заключается в том, что информационная система должна создаваться с учетом постоянного изменения информационных потребностей пользователя. Данный принцип требует разработки средств, с помощью которых информационная система может адаптироваться или модернизироваться в соответствии с новыми требованиями. В настоящее время главное требование в том, что разработчики должны организовать постоянное сопровождение информационной системы в процессе эксплуатации.

Принцип совместимости - построение открытой информационной системы, ориентированной на максимальное использование стандартов программного, технического и иного обеспечения. Согласно теории систем, существует два вида систем: открытые и закрытые. Закрытая система имеет жесткие фиксированные границы, ее действия независимы от окружающей систему среды. Открытая система характеризуется взаимодействием с внешней средой. Принцип совместимости требует, чтобы предприятие рассматривалось в качестве открытой системы, так как оно зависит от поставщиков, конкурентов, налоговой и таможенной политики правительства, с учетом внешней информации в его деятельности.

Принцип непосредственного участия работников предприятия в процессе обследования и разработки информационной системы. Непрерывный контроль со стороны заказчика позволяет избежать неправильных постановок задач и ускорить процесс внедрения системы.

Принцип безопасности - обеспечение безопасности всех информационных процессов.

Принцип эффективности - достижение рационального соотношения между затратами на создание информационной системы и результатами, получаемыми в процессе ее эксплуатации.

Для написания информационных систем используются среды разработки, при помощи которых разработчик может проверять работоспособность и выполнять сборку проекта в единое целое.

Информационные системы хранят большой объем информация, для таких целей используются базы данных и системы управления базами данных, при помощи которых можно создавать, изменять базы данных, а также выполнять добавление и удаление записей из таблиц.

# 1.4 Этапы и методы разработки информационной системы

Проектирование - это один из этапов жизненного цикла (далее - ЖЦ) информационной системы (далее - ИС). ЖЦ ИС состоит из последовательности состояний ИС, начиная от момента возникновения необходимости в данной ИС до момента прекращения ее использования. Применительно к программному обеспечению (далее ПО), являющемуся наиболее сложной и трудоемкой компонентой ИС, обычно выделяют следующие этапы ЖЦ:

1. анализ и формирование требований,
2. проектирование,
3. реализация,
4. эксплуатация (сопровождение),
5. снятие с эксплуатации.

Аналогичные стадии (этапы) составляют и жизненный цикл информационной системы. Первые три этапа ЖЦ (анализ и формирование требований, проектирование, реализация) объединяют общим понятием разработка ИС. Каждый этап разработки ИС можно представить в виде последовательности подэтапов. Так, этап анализ и формирование требований включает:

1. предпроектное обследование объекта информатизации (анализ существующей ИС),
2. разработка концепции ИС,
3. разработка технического задания.

Этот этап предшествует проектированию, иногда его называют предпроектным. Этап проектирование в свою очередь включает такие подэтапы:

1. эскизное проектирование,
2. разработка пилот-проекта,
3. техническое проектирование,
4. рабочее проектирование.

Этот этап завершается созданием проектной документации, достаточной для реализации ИС. Этап реализации включает:

1. создание ИС,
2. тестирование,
3. системная отладка,
4. ввод в эксплуатацию.

При предпроектном обследовании объекта информатизации осуществляется сбор и анализ данных (путем опроса специалистов, изучения документов и технических описаний) о структуре, функциях объекта информатизации. Описываются информационные потоки, реализуемые в объекте бизнес-процессы, существующие технологии управления. Выявляются недостатки существующей информационной системы и обосновывается целесообразность проведения работ по созданию новой (модернизации старой) ИС. Формируются требования пользователей к создаваемой ИС.

На подэтапе разработки концепции ИС осуществляется поиск путей удовлетворения требований пользователей на уровне концепции создаваемой системы (назначение, функции, программно-техническая платформа, режимы использования). Рассматриваются альтернативные варианты концепции построения системы, производится их анализ, выбирается лучшая концепция системы.

На стадии разработки технического задания формируется техническое задание (далее - ТЗ) на информационную систему. Состав и содержание ТЗ определены ГОСТ 34.602-89. Основой ТЗ являются требования к создаваемой системе.

На этапе эскизного проектирования осуществляется предварительная проработка проектных решений по системе в целом или ее частям. Необходимость этого этапа возникает при создании сложных систем, состоящих из большого числа взаимодействующих элементов, например, информационных систем крупных предприятий, отраслей.

Пилотные проекты создаются в случаях, когда необходимо проверить правильность тех или иных проектных решений. Как правило, пилот-проекты реализуются для отдельных элементов сложных информационных систем.

На этапе технического проектирования осуществляется разработка основных проектных решений по системе и ее частям: определение функциональной структуры, выбор комплекса технических средств, выбор СУБД, проектирование баз данных, входных и выходных форм; разработка технологии обработки информации, обеспечивающей выполнение требований, предъявляемых к данным; разработка алгоритмов обработки данных при выполнении различных функций.

На этапе рабочего проектирования проводится разработка программных средств системы, их автономная отладка, осуществляется адаптация приобретаемых программных продуктов, готовится проектная документация, ввода в действия и эксплуатации ИС.

Этап реализация включает выполнение строительно-монтажных работ, тестирование и системную отладку программных средств, организационную подготовку к вводу ИС в действие, обучение персонала, пусконаладочные работы, опытную эксплуатацию (с необходимой доработкой по ее результатам), приемочные испытания.

Основная идея структурного подхода заключается в декомпозиции, то есть разбиении системы на функциональные подсистемы. Структурный подход применим при проектировании ИС, где требуется получить представление о технологическом процессе (далее - ТП) производства изделия. Декомпозиция системы до атомарного уровня (элементарной операции) может нарушить принцип абстрагирования, то есть выделения существенных и отвлечения от несущественных аспектов системы.

Объектно-ориентированный подход предполагает оперирование «объектом», обладающим некоторыми атрибутами и способным выполнять определенные операции. При этом повышается унификация разработки и ее пригодность для повторного использования. ИС строится на основе стабильных промежуточных описаний, что упрощает внесение изменений.

# 2 Аналитическая часть

# 2.1 Анализ существующих языковых средств разработки информационной системы

Одними из самых распространенных языков программирования на сегодняшний день являются C++ и C#. Для написания информационной системы был выбран C#. Выбор обусловлен не только знанием данного языка программирования, но и достоинствами:

* в отличии от Java, которой не пошел на пользу переход в собственность Oracle, C# хорошо развивается благодаря усилиям Microsoft;
* в последнее время много совершенствуется, так как C# был создан позже, чем Java и другие языки, то требовалось очень много доработать. Также это касается популяризации и бесплатности - было обещано открыть исходный код, а инструменты (Visual Studio, Xamarin) стали бесплатными для частных лиц и небольших компаний;
* синтаксический сахар - это конструкции, которые созданы для облегчения написания и понимания кода (код другого программиста) и не играют роли при компиляции;
* синтаксис похожий на C, С++ или Java облегчает переход для других программистов;
* благодаря покупке Xamarin на C# теперь можно писать под Android и iOS. Это, несомненно, большой плюс, так как их собственная мобильная ОС (Windows Phone) не завоевала большой популярности.

С# позволяет стартовать разработку быстрее, а это позволяет быстрее получить прототип решения. Скорость разработки на С# на начальных этапах проекта значительно выше по сравнению с С++.

Однако, когда инфраструктура проекта создана, основные подходы и библиотеки выбраны, а билд настроен, скорость разработки на С++ и скорость разработки на С# становятся примерно одинаковыми.

Таким образом, в коротких малобюджетных проектах С# будет иметь преимущество по скорости разработки, но в длинных и относительно дорогих данное преимущество будет незначительным.

Сборка С++ проектов заметно сложнее сборки проектов С#. Однако стоит понимать, что большая сложность предоставляет и дополнительную гибкость, которая рано или поздно может стать полезной.

# 2.2 Анализ программных средств разработки информационной системы

Для написания информационной системы использовалась среда разработки Microsoft Visual Studio. Данная среда разработки была выбрана в связи с тем, что среда является самой популярной среди разработчиков, а также определяется как вне конкурентная среда разработки для настольного программного обеспечения.

Microsoft Visual Studio - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментальных средств. Данные продукты позволяют разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологии Windows Forms, а также веб-сайты, веб-приложения, веб-службы как в родном, так и в управляемом кодах для всех платформ, поддерживаемых Windows, Windows Mobile, Windows CE, .NET Framework, Xbox, Windows Phone .NET Compact Framework и Silverlight.

Visual Studio включает в себя редактор исходного кода с поддержкой технологии IntelliSense и возможностью простейшего рефакторинга кода. Встроенный отладчик может работать как отладчик уровня исходного кода, так и отладчик машинного уровня. Остальные встраиваемые инструменты включают в себя редактор форм для упрощения создания графического интерфейса приложения, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

Visual Studio позволяет создавать и подключать сторонние дополнения (плагины) для расширения функциональности практически на каждом уровне, включая добавление поддержки систем контроля версий исходного кода, добавление новых наборов инструментов или инструментов для прочих аспектов процесса разработки программного обеспечения.

Для хранения данных используется система управления базами данных phpMyAdmin. Выбор системы обусловлен тем что для дипломного проекта помимо информационной системы был разработан сайт, а для того чтобы сайту обращаться в базу данных проще всего использовать систему управления базами данных phpMyAdmin.

phpMyAdmin - это программа написанная на PHP и предназначенная для управления сервером MySQL через всемирную сеть. phpMyAdmin поддерживает широкий набор операций над MySQL. Наиболее часто используемые операции поддерживаются с помощью пользовательского интерфейса (управление базами данных, таблицами, полями, связями, индексами, пользователями, правами, одновременно можно напрямую выполнить любой SQL-запрос).

Для облегчения использования наибольшим количеством людей, phpMyAdmin переведен на 57 языков и поддерживает письменность как слева направо, так и справа налево.

Начиная с версии 3.0.0, phpMyAdmin присоединился к инициативе GoPHP5 и прекратил поддержку написания кода под устаревшие версии PHP и MySQL; версия 3 и последующие требуют наличия PHP 5.2 и MySQL 5.

Для дипломного проекта, был разработан сайт, на котором пользователь просматривает статус своего заказа. Сайт разработан при помощи среды разработки «PhpStorm». Выбор обусловлен личным опытом использования среды разработки, а также многообразием функционала.

PhpStorm - коммерческая кросс-платформенная интегрированная среда разработки для PHP. Разрабатывается компанией JetBrains на основе платформы IntelliJ IDEA.

PhpStorm представляет собой интеллектуальный редактор для PHP, HTML и JavaScript с возможностями анализа кода, предотвращения ошибок в коде и автоматизированными средствами рефакторинга для PHP и JavaScript. Автодополнение кода в PhpStorm поддерживает спецификацию PHP 5.3, 5.4, 5.5, 5.6, 7.0, 7.1 и 7.2 (современные и традиционные проекты), включая генераторы, сопрограммы, пространства имен, замыкания, типажи и синтаксис коротких массивов. Имеется полноценный SQL-редактор с возможностью редактирования полученных результатов запросов.

PhpStorm разработан на основе платформы IntelliJ IDEA, написанной на Java. Пользователи могут расширить функциональность среды разработки за счет установки плагинов, разработанных для платформы IntelliJ, или написав собственные плагины.

# 2.3 Анализ структуры и принципов функционирования баз данных

База данных (далее - БД) - организованная в соответствии с определенными правилами и поддерживаемая в памяти компьютера совокупность данных, характеризующая актуальное состояние некоторой предметной области и используемая для удовлетворения информационных потребностей пользователей.

Данные в БД логически структурированы (систематизированы) с целью обеспечения возможности их эффективного поиска и обработки в вычислительной системе.

Структурированность подразумевает явное выделение составных частей (элементов), связей между ними, а также типизацию элементов и связей, при которой с типом элемента (связи) соотносится определенная семантика и допустимые операции.

БД включает метаданные, описывающие логическую структуру БД в формальном виде (в соответствии с некоторой метамоделью).

По модели данных БД выделяют следующие:

* иерархические,
* сетевые,
* реляционные,
* объектные и объектно-ориентированные,
* объектно-реляционные.

База данных определяется как совокупность взаимосвязанных данных, характеризующихся возможностью использования для большого количества приложений, возможностью быстрого получения и модификации необходимой информации, мини­мальной избыточностью информации, независимостью от при­кладных программ, общим управляемым способом поиска. Для работы с БД существует специальное программное обеспечение - системы управления базами данных (далее - СУБД).

СУБД выполняет функции:

- управление данными непосредственно в БД, находящейся на сервере;

- управление данными, находящимися в памяти компьютера - функция, связанная с тем, что СУБД работают с БД большого размера; в целях ускорения работы СУБД используется буферизация данных в памяти каждого компьютера;

- управление транзакциями - функция, которая производит ряд операций над БД как над единым целым. Транзакция - манипуляция над данными (добавление, изменение, удаление, запрос). Если транзакция успешно выполняется, то СУБД вносит соответствующие изменения в БД, в противном случае БД не изменяется;

- управление изменениями и протоколирование - функция, связанная с надежностью хранения данных, то есть возможностью СУБД восстанавливать состояние БД в аварийных ситуациях, например, при выключении питания, сбое носителя информации;

- поддержка языков БД - для работы с БД используются специальные языки, наиболее распространенным из которых является SQL.

Для работы с БД, как правило, используется архитектура «клиент - сервер». Основу СУБД составляет сервер БД - программа, осуществляющая комплекс действий по управлению данными. В качестве клиента БД может выступать другая программа, работающая на том же компьютере, что и сервер, либо на другом, связанном с компьютером-сервером посредством сети. Таким образом, данная архитектура включает, как минимум, три компонента:

- сервер БД, который, собственно, и является СУБД;

- клиенты БД - программы или компьютеры с соответствующими программами, выполняющие запросы к серверу;

- сеть или коммуникационное программное обеспечение.

# 3 Практическая часть

# 3.1 Техническое задание на разработку информационной системы

Для написания ИС и сайта, заказчиком и разработчиком было составлено техническое задание. В техническом задании были описаны требования, изложенные ниже.

Язык программирования для написания ИС не принципиален, разработчиком был выбран C#. Программа должна содержать следующие сущности:

- клиенты;

- заказы;

- сотрудники.

Сущность «Клиент» должна содержать следующую информацию:

- идентификатор;

- ФИО;

- номер телефона;

- электронная почта;

- дополнительная информация.

Сущность «Сотрудник» должна содержать следующую информацию:

- идентификатор;

- ФИО;

- дата рождения;

- электронная почта;

- дата приема на работу;

- роль в программе;

- логин для программы;

- пароль для программы;

- номер телефона;

- должность.

Такие поля как «Роль в программе» и «Должность» было решено вынести в отдельные сущности, которые имели предопределенные элементы, а сущность сотрудник имел лишь идентификатор сущности.

Сущность «Заказы» должна содержать следующую информацию:

- идентификатор;

- исполняющий сотрудник;

- заказавший клиент;

- тема заказа;

- дата начала проекта;

- дата закрытия проекта;

- статус заказа;

- цена заказа.

Статус заказа был выделен в отдельную сущность, с предопределенными элементами. У заказа хранится только идентификатор статуса.

Для каждого заказа должен формироваться уникальный номер заказа, при помощи которого пользователь, при помощи сайта может узнать информацию о своем заказе.

Программа должна иметь систему авторизации для пользователей. Администратор должен иметь права на добавление, изменение и удаление сущностей. Администратору также должен быть доступен функционал по формированию отчета по определенному фильтру. Отчет должен формироваться в документ Microsoft Word.

Было решено для работы с базой данных использовать «phpMyAdmin», для того чтобы сайт и программа имели единую БД.

Сайт должен содержать форму для указывания номера своего заказа. После того как пользователь указывает существующий заказ, ему должна отобразиться информация о его заказе, а именно:

- клиент;

- исполнитель;

- почта исполнителя;

- номер заказа;

- дата приема заказа;

- статус заказа.

Время, отведенное для разработки ИС и сайта три недели. Персональные данные в БД должны храниться в зашифрованном виде. Метод шифрования определяется разработччиком.

# 3.2 Описание процесса разработки информационной системы

Для разработки ИС использовался фреймворк. Данная библиотека позволяет стилизировать элементы управления. На рисунке 1 изображен конечный интерфейс информационной системы, стилизированный при помощи данного фраемворка.

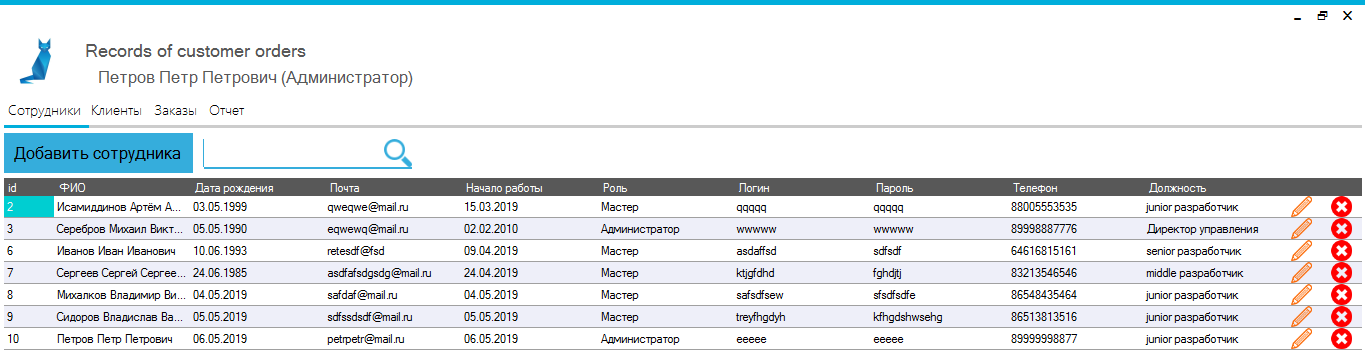


Рисунок 1 - Конченый интерфейс программы

Первой разрабатывалась часть администратора. Сначала был сформирован дизайн системы. Ключевым элементом системы выступает «metroTabControl», при помощи которого пользователь переключает основные страницы программы. Далее на каждой странице размещались таблицы, текстовые поля и кнопки для управления.

Для написания основных методов, которые будут доступны для программы отовсюду был создан класс «General». Данный класс описывает общие методы, которые в дальнейшем может вызвать каждая форма программы. На рисунке 2 изображена часть конечного варианта класса «General».

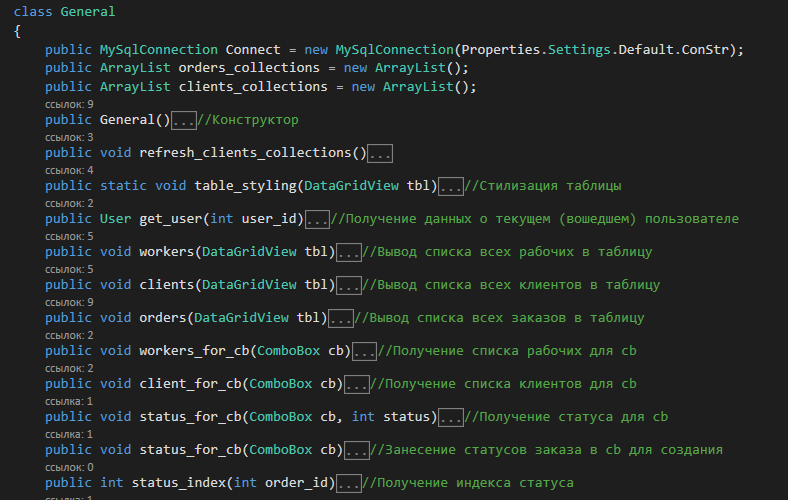


Рисунок 2 -Часть класса «General»

Далее были созданы классы сущностей, используемых в системе. На рисунке 3 изображена часть конечного варианта класса «User», данный класс описывает сущность сотрудники.

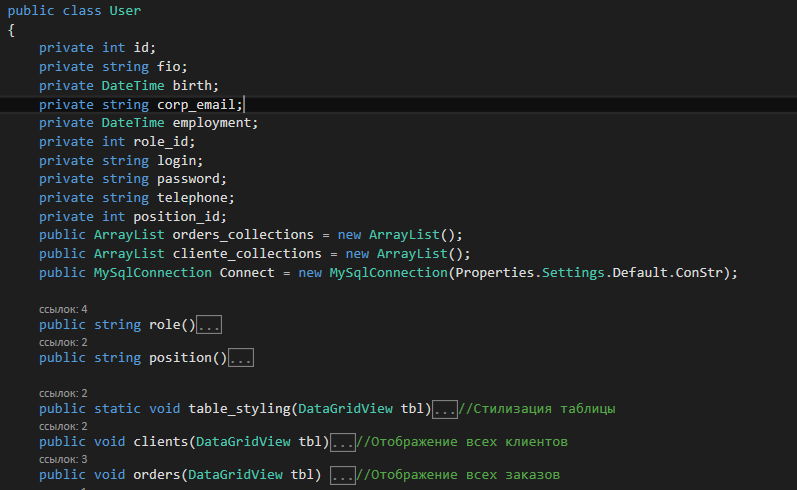


Рисунок 3 - Класс «User»

Далее был создан класс «Client», который описывал сущность клиент. На рисунке 4 изображен завершенный класс.

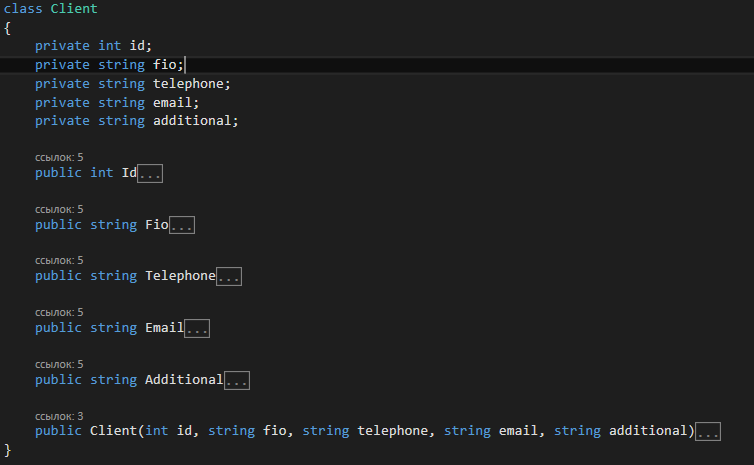


Рисунок 4 - Класс «Client»

Следующим был создан класс «Order», который описывал сущность заказы. На рисунке 5 изображена часть законченного класса «Order».

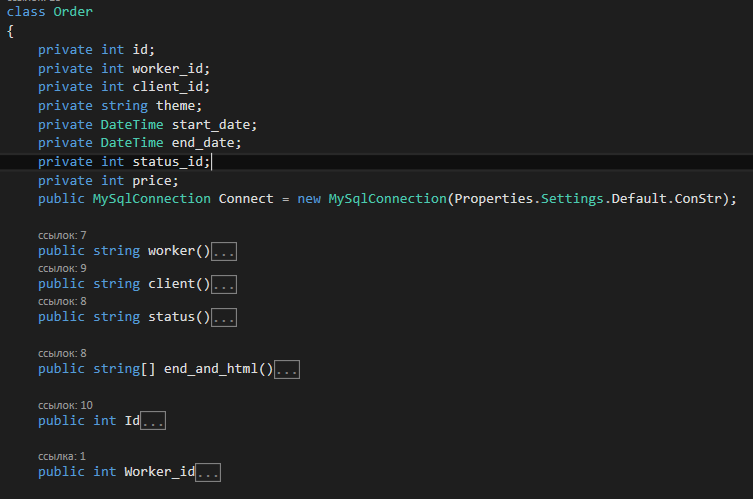


Рисунок 5 - Класс «Order»

После того как данные классы были созданы, основные сущности были заполненными полями, а также свойствами для обращения к ним.

Следующим этапом было написание методов для отображения списка записей в требуемые таблицы системы. Эти методы были описаны в классе «General». На рисунке 6 изображен метод вывода списка клиентов в таблицу.

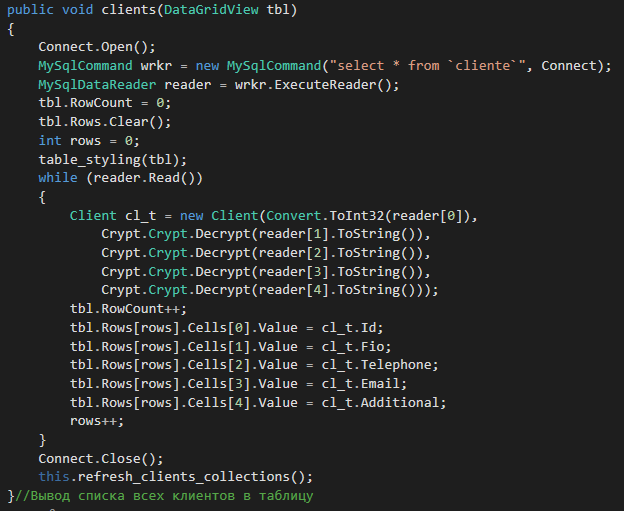


Рисунок 6 - Метод отображения списка клиентов

Затем создавались формы для добавления сущностей. На рисунке 7 изображена форма добавления клиента.

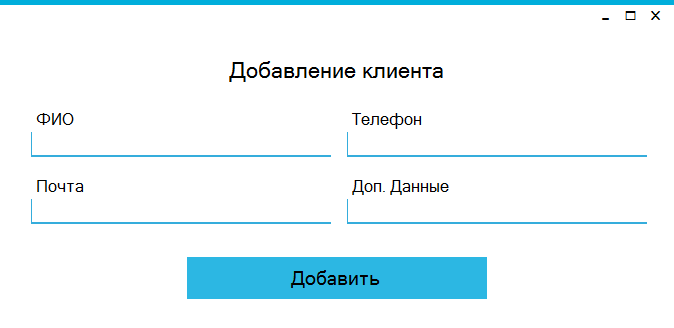


Рисунок 7 - Форма добавления клиента

По нажатию на кнопку «Добавить», программа выполняет запрос для добавления записи в БД, а также выполняет обновление списка записей в соответствующей таблице. На рисунке 8 изображена часть кода добавления новой записи о клиенте в БД.

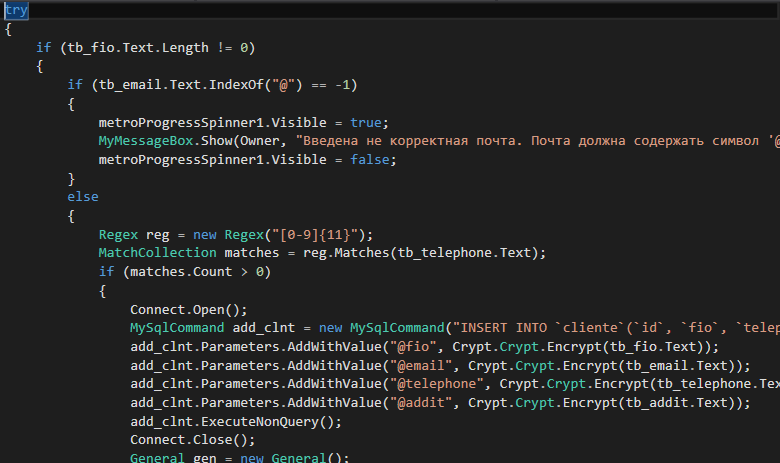


Рисунок 8 - Код добавления нового клиента в БД

Далее разрабатывались формы для изменения информации о сущности. На рисунке 9 изображена форма изменения информации о клиенте.

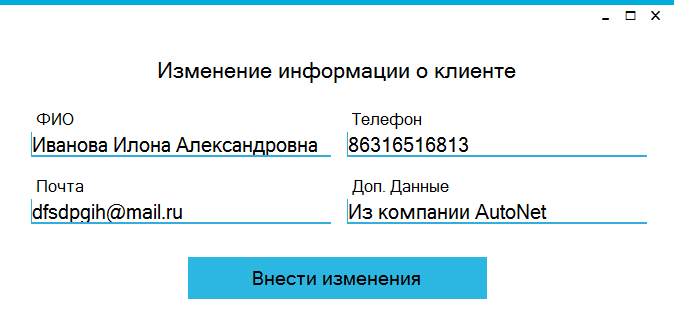


Рисунок 9 - Форма изменения информации о клиенте

По нажатию на кнопку «Внести изменения», программа выполняет код для обновления записи в таблице БД, а также выводе нового списка в соответствующую таблицу системы. Код формы имеет валидаторы, при помощи которых проверяется корректность введенных данных. На рисунке 10 изображена часть кода обновления данных о клиенте.

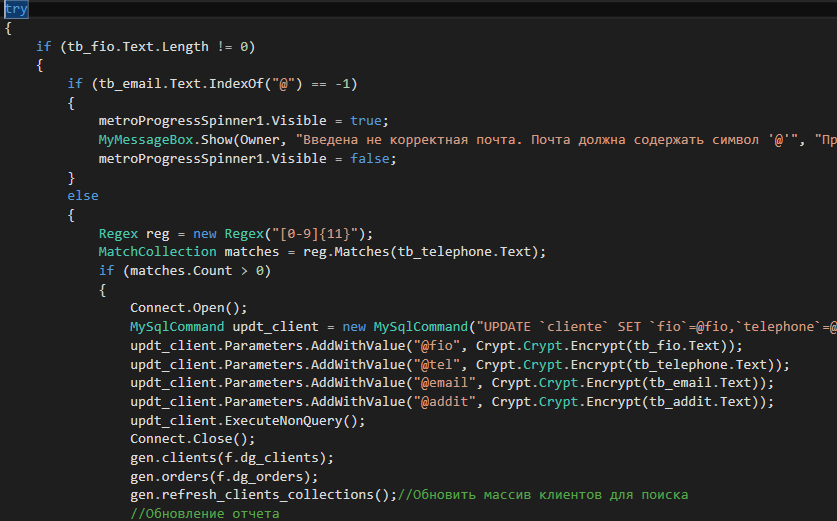


Рисунок 10 - Код обновления данных о клиенте

Далее разрабатывалась возможность удаления сущности. При нажатии на кнопку удаления пользователю отображается подтверждение удаления, после чего требуемая запись удаляется из БД и обновляется список записей в системе. Часть кода удаления клиента изображена на рисунке 11.

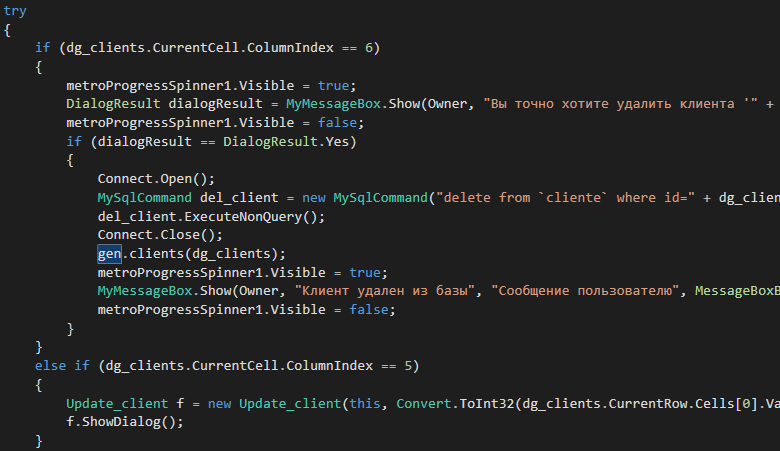


Рисунок 11 - Код удаления клиента

Для программы также был создан класс для шифрования и дешифрования данных «Crypt». Код класса представлен не будет в связи политикой о конфиденциальности организации.

Для отображения разного вида сообщений пользователю был создан класс «MyMessageBox». На рисунке 12 изображено диалоговое окно, вызываемое этим классом.

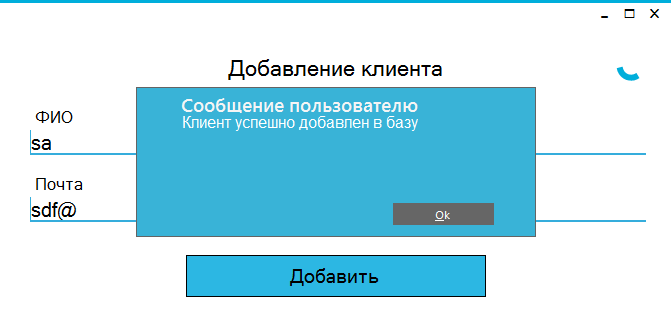


Рисунок 12 - Диалоговое окно программы

Код для данного окна был взят из интернета. На рисунке 13 изображена часть кода класса «MyMessageBox».

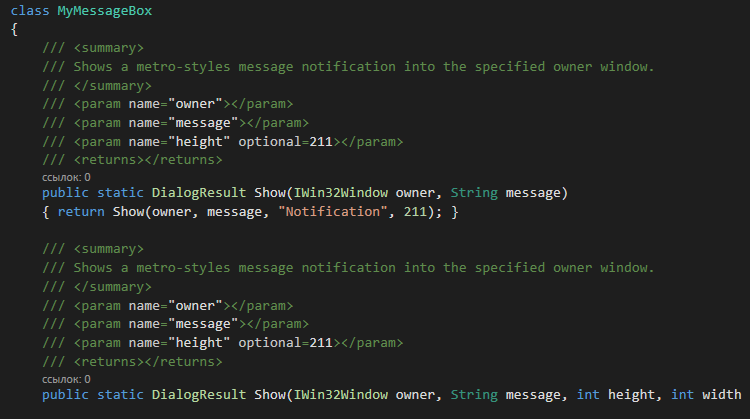


Рисунок 13 - Код класса «MyMessageBox»

Далее разрабатывался функционал создания отчета. Была создана форма для настраивания отчета, а также фильтров отображения данных. На рисунке 14 изображена форма настроек отчета.

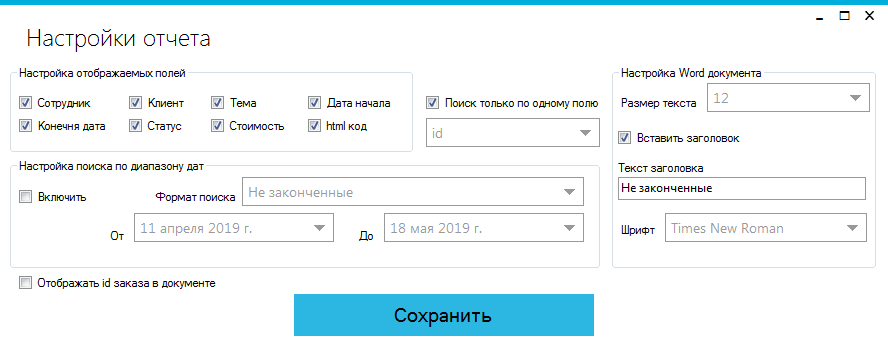


Рисунок 14 - Форма настроек отчета

Далее был написан код для формирования отчета. На рисунке 15 изображена часть кода формирования отчета.

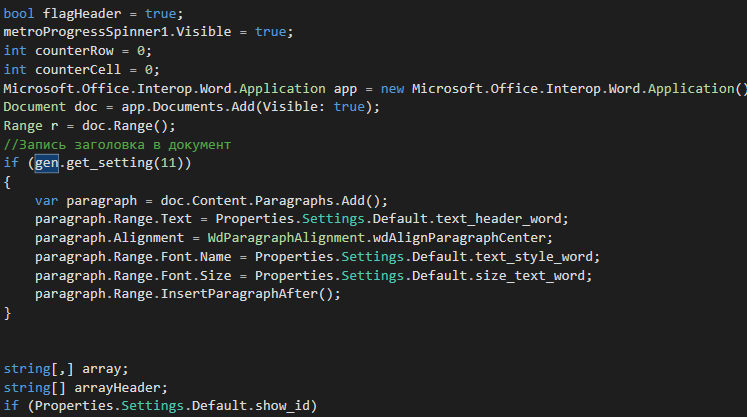


Рисунок 15 - Код формирования отчета.

После завершения работы над административной частью была начата работа над частью работника. Интерфейс программы работника изображен на рисунке 16.

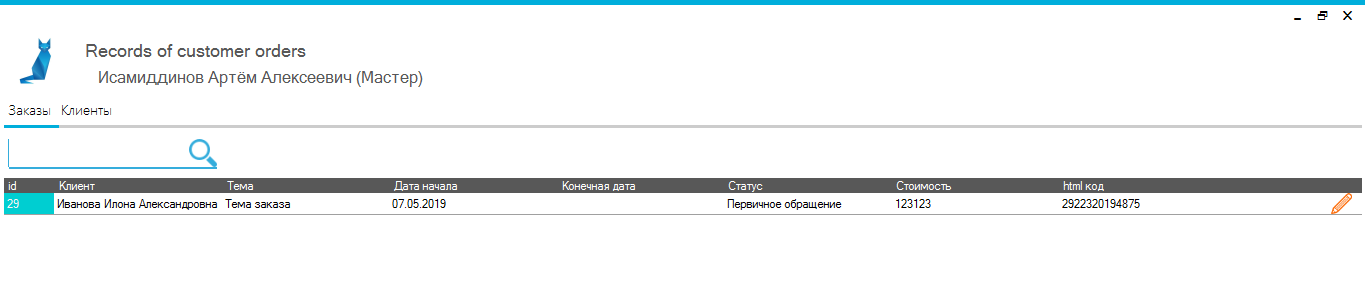


Рисунок 16 - Интерфейс программы обычного рабочего

Интерфейс работника имеет на много меньше функционала чем администратора. Для работника отображается только его заказы, а также только те клиенты, с которыми он работал.

Методы для отображения этих данных описаны в классе «User», так как имеют привязку к конкретному рабочему. На рисунке 17 изображен код отображения клиентов для сотрудника.

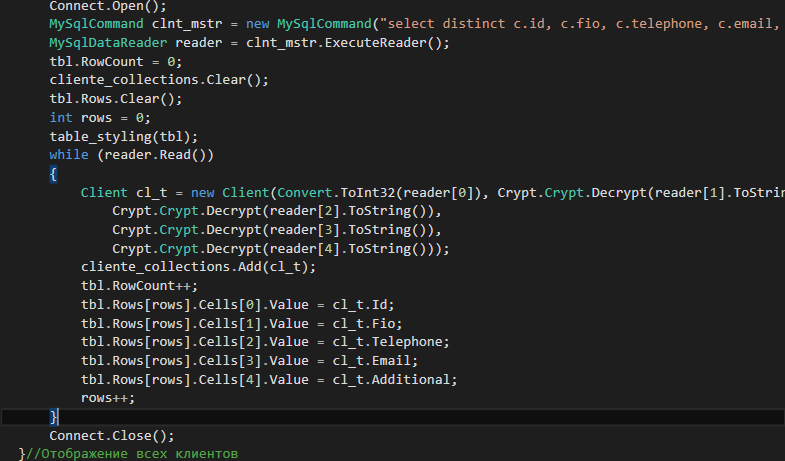


Рисунок 17 - Отображение клиентов сотрудника

Сотрудник не может выполнять удаление, его права ограничиваются только изменением статуса заказа. Поэтому для сотрудника отображается та же форма что и для администратора, но только с одним функционирующим полем.

Также в программе предусмотрен функционал по смене данных для подключения к БД. Для отображения данной формы нужно выполнить определенные действия, которые описываться не будут, в целях безопасности. О данных действиях знает только администратор. На рисунке 18 изображена форма для смены данных подключения к БД.

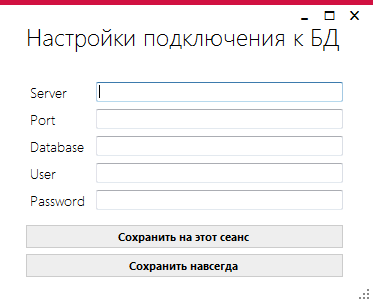


Рисунок 18 - Окно смены данных подключения к БД

После разработки информационной системы был разработан сайт. Сайт имеет одностраничную структуру. Изначально пользователю предоставляется возможность указать код своего заказа, который выдается администратором. На рисунке 19 изображен сайт с формой ввода кода заказа.

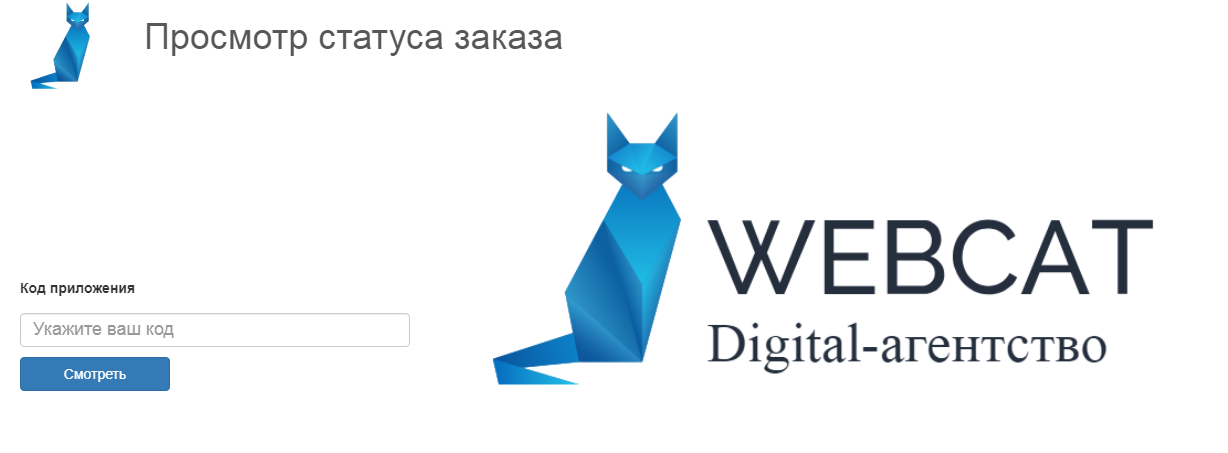


Рисунок 19 - Форма ввода заказа на сайте

После того как пользователь укажет свой код, для него отображается информация о его заказе. На рисунке 20 изображен сайт с информацией о заказе.

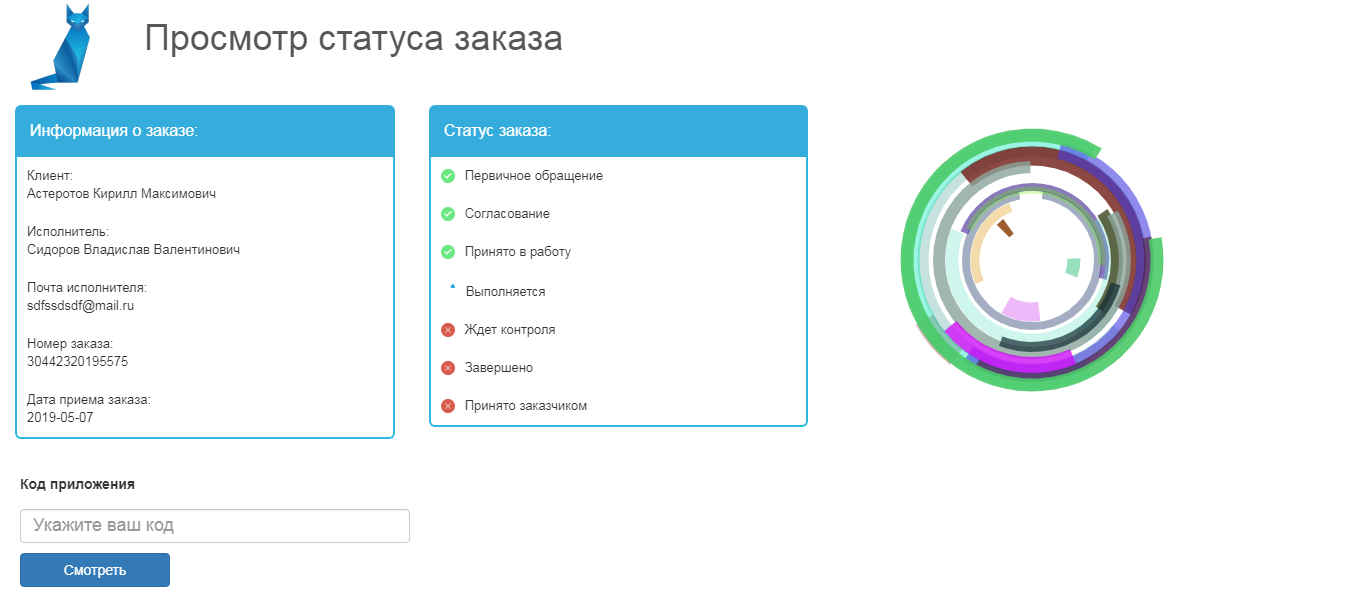


Рисунок 20 - Сайт с информацией о заказе

После того как были разработаны информационная система и сайт, проводилось тестирование, на наличие различных ошибок. Были исправлены все найденные ошибки, самыми значимыми из которых были ошибки при шифровании, а также проблемы при отмене сохранения сформированного отчета в окне выбора места сохранения отчета.

# 3.3 Описание функциональных возможностей информационной системы

Разработанная ИС имеет функционал по авторизации пользователей в системе, благодаря чему увеличивается безопасность информации. Авторизация происходит при помощи логина и пароля выдаваемых сотруднику администратором. На рисунке 21 изображено окно входа в программу.

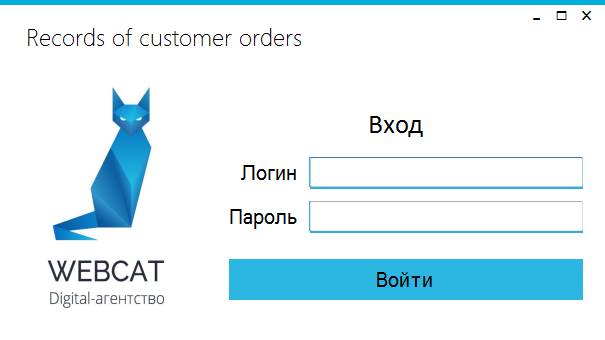


Рисунок 21 - Окно входа в программу

После того как пользователь авторизовался, в зависимости от его роли, будет запущена основная часть ИС. Для пользователя, имеющего роль «Мастер», доступна возможность просмотра его заказов. На рисунке 22 изображено окно со списком заказов рабочего.

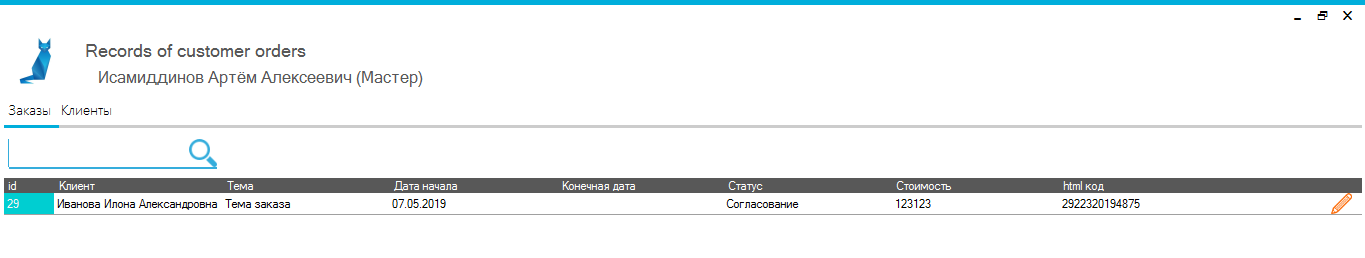


Рисунок 22 - Список заказов рабочего

Для рабочего отображаются только не законченные заказы, как только заказ получает статус «Принято заказчик», заказ удаляется из списка. Рабочий также может менять статус заказа до «Ждет контроля». Это сделано для того чтобы рабочий сам не смог закрыть заказ. Полностью закрыть заказ может только администратор. Форма изменения статуса заказа изображена на рисунке 23.

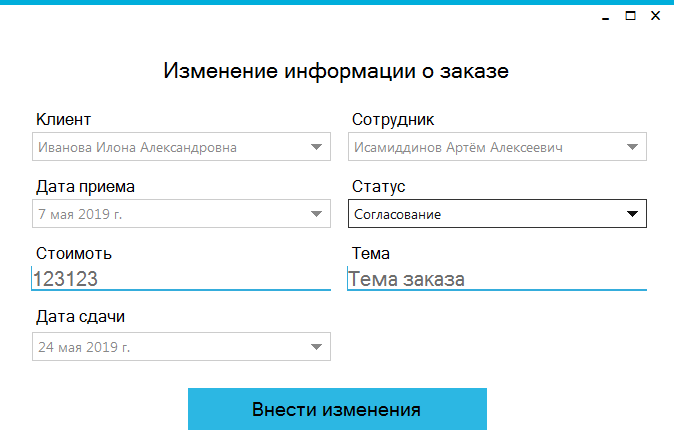


Рисунок 23 - Форма изменения статуса заказа

Для рабочего доступно только поле «Статус», остальные поля не работают, данные поля доступны только администратору.

Пользователь с ролью «Мастер», также имеет возможность посмотреть список клиентов, но только тех. с которыми он уже работал. На рисунке 24 изображена программа со списком клиентов рабочего.

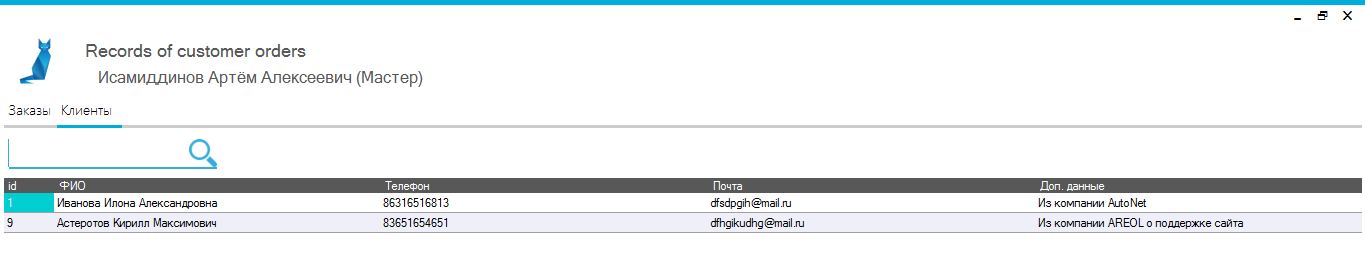


Рисунок 24 - Список клиентов рабочего

Если пользователь авторизовался под ролью «Администратор», то ему доступен функционал просмотра, полного изменения и удаления записей. Все сущности программы имеют схожие действия поэтому далее будет описываться функционал на основе сотрудников.

Администратору доступен список всех сотрудников организации. На рисунке 25 изображен список сотрудников организации.

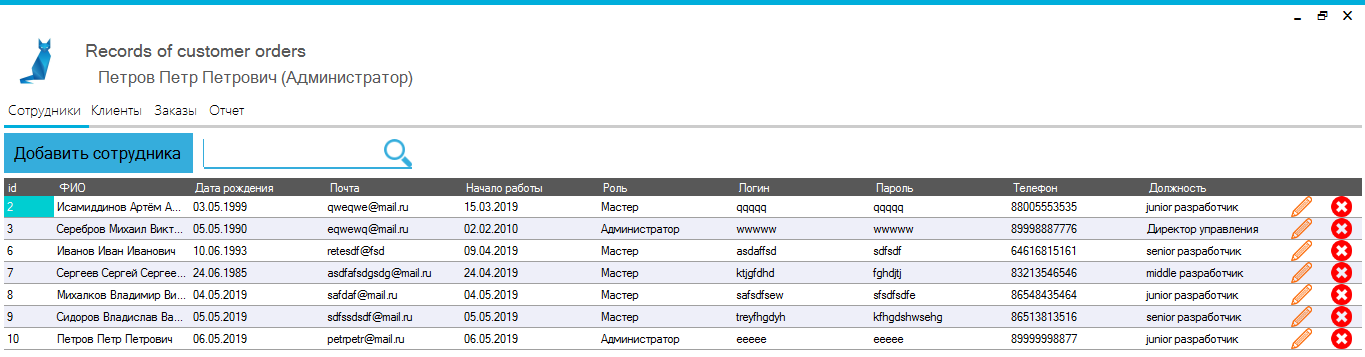


Рисунок 25 - Список сотрудников организации

Каждую запись администратор может изменить, нажав на карандаш в строке записи или удалить, нажав на крест рядом с карандашом той записи. При изменении записи, администратору открывается форма где можно вносить изменения. На рисунке 26 изображена форма изменения информации о сотруднике.

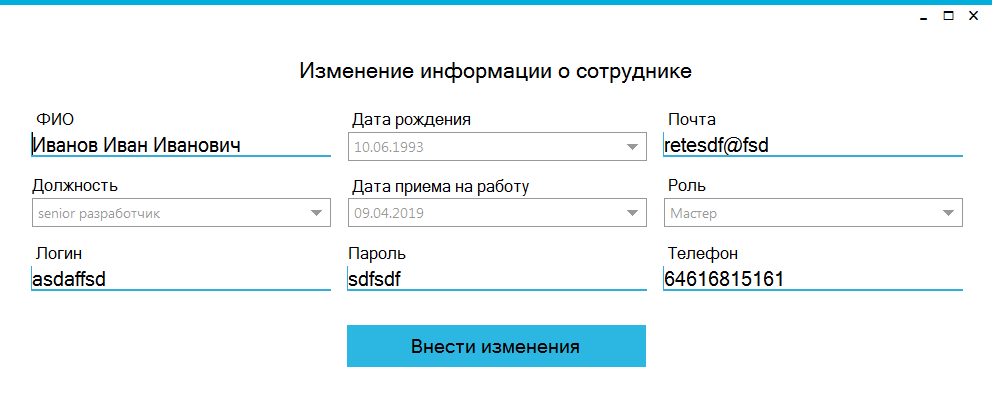


Рисунок 26 - Форма изменения информации о сотруднике

После внесения изменений ИС система оповестит администратора о успешном внесении изменений, если что-то пойдет не так, то появится соответствующее предупреждение.

Если администратор попытается удалить запись ИС попросит подтверждения. При подтверждении запись будет удалена из базы, а также из списка в программе. На рисунке 27 изображено предупреждение удаления.

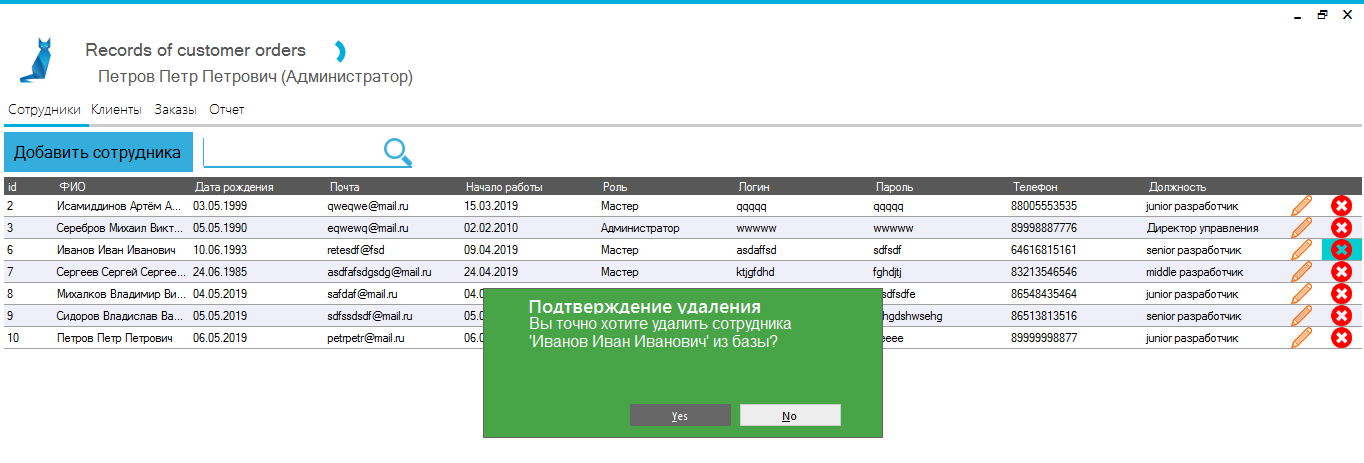


Рисунок 27 - Предупреждение удаления

Администратор также может добавлять новую запись по нажатию на кнопку «Добавить сотрудника». После чего откроется форма добавления сотрудника, изображенная на рисунке 28.

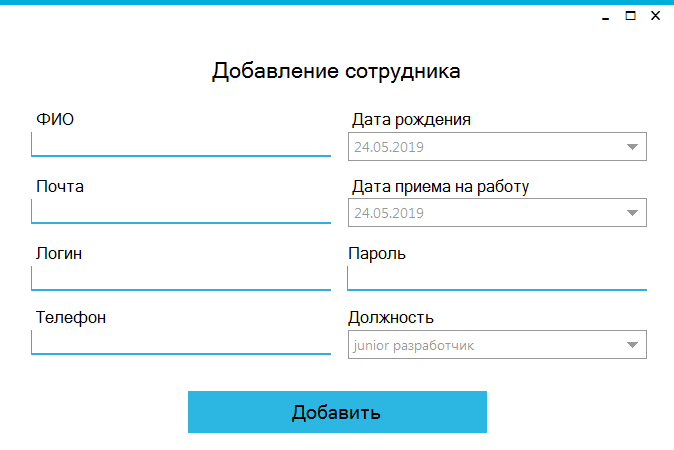


Рисунок 28 - Форма добавления сотрудника

Администратору также доступен функционал для формирования отчета при помощи определенных настроек. Настройка осуществляется по нажатию на шестеренку во вкладке «Отчет». На рисунке 29 изображена форма настройки отчета.

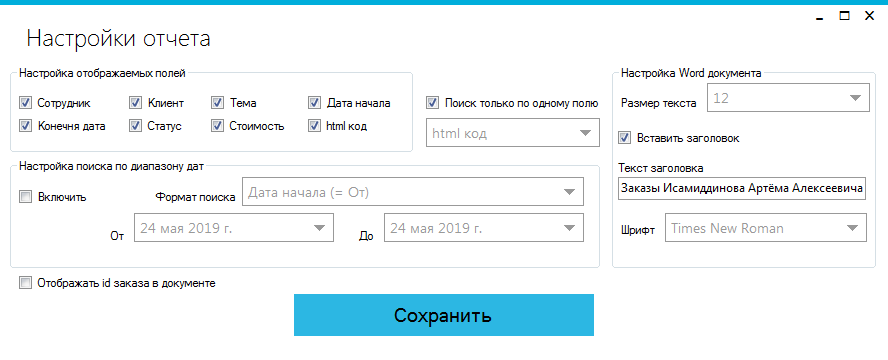


Рисунок 29 - Форма настройки отчета

После настройки во вкладе «Отчет» будет заново сформированная таблица в зависимости от примененных настроек, при помощи кнопки «Экспорт в Word», администратор может сформировать отчет в документ Microsoft Word.

# 3.4 Описание защиты информационной системы

Как говорилось ранее ИС имеет возможность авторизации при помощи логина и пароля, который выдается пользователю администратором. Логин и пароль администратор создает сам. Если при входе будет указан неверный логин или пароль, то программа отобразить соответствующую надпись на форме. На рисунке 30 изображен результат неправильного ввода логина или пароля.

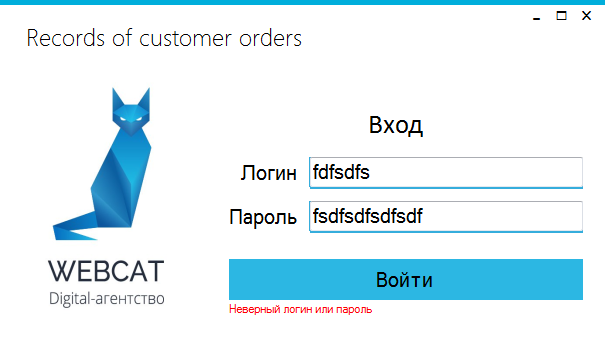


Рисунок 30 - Результат неправильного ввода логина или пароля

Также ИС имеет при работе с БД шифрует персональные данные. При добавлении, изменении записей данные в БД записываются в зашифрованном в виде. При извлечении данных из БД данные расшифровываются обратно. Шифрование происходит при помощи методов класса «Crypt». В данном классе существует два метода «Encrypt» и «Decrypt». Первый метод получает строку и возвращает ее в зашифрованном виде. Второй получает зашифрованную строку и возвращает ее в расшифрованном виде.

Шифрование происходи при помощи специального ключа. Данный ключ содержит в себе сопоставление обычному символу зашифрованный. Шифрование применяется только на буквы русского и латинского алфавита разных регистров, а также на цифры. Ключ был сгенерирован при помощи написанной программы, которая формирует зашифрованные символы и генерирует файл в котором содержится «таблица» шифрованных значений. На рисунке 31 изображена программа генерирования ключа.

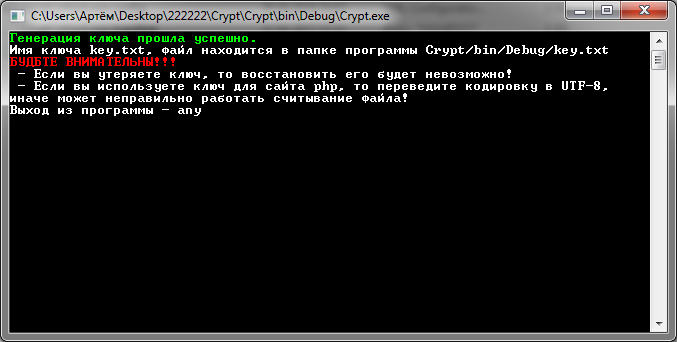


Рисунок 31 - Программа генерирования ключа

Так как ключ генерируется случайным образом подделать его практически невозможно. Нои из-за этого при потере ключа нельзя будет расшифровать данные. В связи с этим создано несколько копий ключа. Ключ используется также сайтом для дешифрования данных.

# 3.5 Расчет затрат на внедрение и сопровождение информационной системы

# 3.6 Охрана труда и техника безопасности при организации рабочего места

# Заключение

# Список использованных источников