Государственное АВТОНОМНОЕ профессиональное образовательное учреждение

свердловской области

«Каменск-Уральский политехнический колледж»

Специальность 09.02.07 Информационные системы и программирование

Группа ИСиП20-301

КУРСОВОЙ ПРОЕКТ

по МДК 05.02 разработка кода информационных систем

тема: Разработка информационной системы

фитнес-клуба

Выполнила:

Алексеева О.Ю.

Проверила:

Калмыкова И.М.

2023

Содержание

[ВВЕДЕНИЕ 3](#_Toc137074803)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 5](#_Toc137074804)

[1.1. Анализ предметной области 5](#_Toc137074805)

[1.2. Постановка задачи 8](#_Toc137074806)

[2. ПРОЕКТНАЯ ЧАСТЬ 10](#_Toc137074807)

[2.1. Проектирование и реализация базы данных системы 10](#_Toc137074808)

[2.2. Проектирование и реализация пользовательского интерфейса системы 12](#_Toc137074809)

[2.3. Структура приложения 13](#_Toc137074810)

[2.4. Реализация пользовательских форм 15](#_Toc137074811)

[2.5. Разработка запросов 19](#_Toc137074812)

[2.6. Разработка отчетов 25](#_Toc137074813)

[ЗАКЛЮЧЕНИЕ 26](#_Toc137074814)

[Список литературы 27](#_Toc137074815)

[Приложение А. Техническое задание на курсовой проект 28](#_Toc137074816)

## ВВЕДЕНИЕ

В наше время информационные технологии шагнули далеко вперёд. Для осуществления полноценной, удобной работы какой-либо организации люди зачастую прибегают к использованию информационных систем для автоматизирования основных бизнес-процессов.

Информационная система - [система](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0), предназначенная для хранения, поиска и обработки [информации](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), и соответствующие организационные ресурсы, которые обеспечивают и распространяют информацию.

В зависимости от специфики предметной области информационные системы могут отличаться друг от друга по функциям, реализации и архитектуре, но каждая из них сходится в том, что система ориентируется на конечного пользователя с минимальными знаниями в области применения вычислительной техники, т.е. само приложение должно быть удобным, простым и понятным для понимания и использования.

Объектом курсового проекта является фитнес-клуб.

Фитнес-клуб - это сооружение, которое имеет площадь для проведения оздоровительных и фитнес-тренировок при помощи силовых упражнений и оборудования для кардио-тренировок и которое открыто для свободного посещения за плату на основе платежей за разовое посещение или по членской системе. В тренажерном клубе можно нанять фитнес-тренера.

Фитнес-тренер – это инструктор, который руководит процессом тренировки и отвечает за правильность и безопасность выполнения упражнений. Его основная задача — раскрыть способности клиентов и научить их пользоваться своими ресурсами. Он обеспечивает общую физическую подготовку человека.

Ежедневно количество людей, занимающихся спортом, увеличивается, как и спрос на занятия в тренажерных залах. Необходимо вести учет посетителей, заносить их в базу данных.

Сотрудник может заносить клиентов вручную, но это непрактично, и в случае большого количества людей, неудобно. Работник фитнес-клуба может ошибиться расчетах и записях, вдобавок ещё и переводя большое количество бумаги. Тогда на помощь приходит информационная система, которая значительно снизит количество возможных ошибок, сведёт потребление бумаги к минимуму и освободит сотруднику фитнес-клуба время.

Целью курсового проекта является разработка информационной системы для фитнес-клуба, которая поможет вести учет клиентов и сотрудников.

Для достижения поставленной цели необходимо выполнить следующие задачи:

* 1. Провести анализ предметной области.
  2. Разработать техническое задание на информационную систему.
  3. Спроектировать БД системы.
  4. Разработать запросы в БД.
  5. Реализовать БД.
  6. Реализовать пользовательский интерфейс.
  7. Провести тестирование разработанной информационной системы.

Внедрение информационной системы поможет ускорить процесс получения и обработки информации о клиентах, видах услуг и сотрудников с возможностью их замены, удаления и добавления. Разработка данного ПО снижает затраты времени работы сотрудников во много раз.

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Анализ предметной области

Предметной областью курсового проекта является ведение учета клиентов фитнес-клуба.

Рынок фитнес-услуг в России является относительно молодым: появление первых классических фитнес-клубов в Москве и крупных городах России произошло в начале 90-х годов. До этого же в нашей стране укоренилось понятие «физическая культура».

Деятельность фитнес-клубов направлена на:

* Изменение своего внешнего вида;
* Поддержание внешнего вида;
* Укрепление здоровья;
* Коррекцию осанки.

В наши дни фитнес-центры активно пользуются спросом и не спроста: большинство людей ведут сидячий образ жизни, питаются нездоровой едой или просто перекусывают чем-то по пути, поэтому занятия спортом становятся актуальны как никогда раньше. Развитие спортивных клубов шагнуло вперёд. Появляется современное, удобное оборудование, позволяющее человеку как можно комфортнее заниматься спортом.

Фитнес-центры можно разделить на несколько основных типов:

1. По типу предлагаемых услуг:

* Большие залы с широким спектром услуг;
* Средние залы;
* Маленькие, узкоспециализированные залы

1. По гендерным различиям:

* Женские;
* Мужские;
* Смешанные

1. По возрасту

* Детские;
* Взрослые;
* Смешанные

1. По целевой аудитории:

* Фитнес-клубы премиум класса;
* Фитнес-клубы эконом-класса.

Несмотря на тип фитнес-клуба, каждый из них должен иметь свою систему управления и клиентскую базу, где будет указана вся необходимая информация о человеке.

Фитнес-клуб занимается предоставлением услуг клиенту. Человек может выбрать одну и более услуг и по необходимости оформить абонемент на них. Абонемент можно продлить, либо приостановить его действие. Также фитнес-клуб предоставляет личного тренера, с которым обговаривается график занятий. Клиент заносится сотрудником клуба в базу данных.

В базе данных отмечается следующая информация:

* Паспортные данные;
* Номер телефона;
* Дата посещений;
* При наличии абонемента его срок действия и на какие виды занятий распространяется;
* Оплаченные услуги;
* ФИО тренера, если человек покупал его услугу.

Основными ролями в организации работы фитнес-клуба являются:

* Старший администратор;
* Директор;
* Администратор.

Директор – должностное лицо, руководящее компанией.

Администратор – это должностное лицо, организующее деятельность в какой-либо компании, коллективе или же в учреждении.

В задачи администратора входит ведение учета клиентов, то есть занесение их в базу, обновление, изменение информации.

Старший администратор – лицо, имеющее полный доступ к работе с системой. В его задачи входят: задачи обычного администратора, операции удаления клиентов, ведение списка сотрудников фитнес-центра.

Учёт клиентов – это современная, простая и удобная система для управления отношениями с клиентами, которая обеспечивает ведение базы данных и учет клиентов (организаций и лиц), выставленных им счетов, заключенных с ними договоров (сделок), других документов, продуктов (товаров и услуг), сотрудников.

Основной бизнес-процесс имеет множество вспомогательных бизнес-процессов, например:

* Учет продаж;
* Учет предоставления услуг;
* Учет сотрудников;
* Учет клиентов.

В данном курсовом проекте будет рассмотрен вспомогательный

бизнес-процесс учета клиентов в фитнес-клубе.

Информационную систему, предназначенную для учета клиентов, можно отнести к классу Customer Relationship Management (CRM-система).

CRM-система – это система, в которой общение с клиентом является самой главной частью. CRM-система решает задачи, направленные на удовлетворение и удержание клиентов и служит оптимизации деятельности компании, сокращая издержки, связанные с поиском и обработкой информации, анализом данных, управлением продажами и т.д.

Без автоматизации процесса учета клиентов фитнес-клуба все сведения будут храниться в бумажном виде, а это в свою очередь неудобно и есть риск потерять необходимую информацию. Применение информационной системы позволит упростить работу сотрудникам, значительно сократив время работы и ускорив её и упростит работу с обобщением информации.

Для более детального рассмотрения процесса учета клиентов рассмотрим Рис. 1, представленный в виде диаграммы IDEF0, выполненной в программе Microsoft Visio.



Рисунок 1 – Функциональная диаграмма по учету клиентов

А также рассмотрим декомпозицию функциональной диаграммы по учету клиентов (Рис. 2).

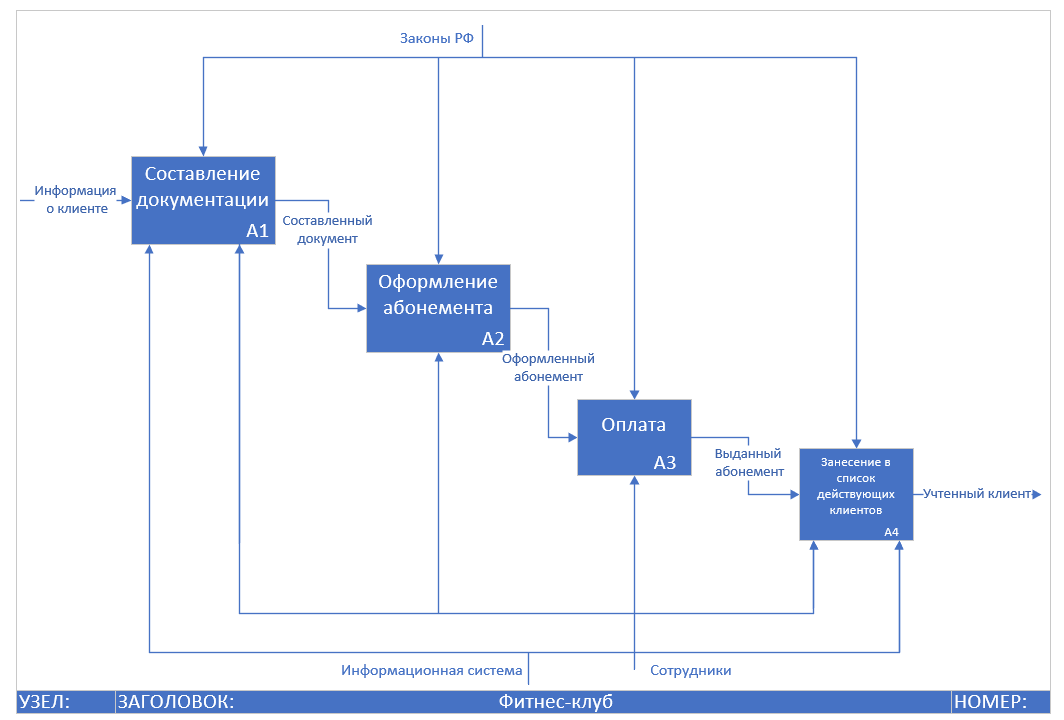


Рисунок 2 – Декомпозиция функциональной диаграммы по учету клиентов

Microsoft Visio — программа для создания различных видов схем. К их числу относятся блок-схемы, диаграммы DFD, схемы технологических процессов, модели бизнес-процессов, диаграммы плавательных дорожек, трехмерные карты и так далее.

IDEF0 (Integration Definition for Function Modeling) - методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Декомпозиция IDEF0 - это метод моделирования бизнес-процессов, который используется для детализации сложных функций.

## Постановка задачи

Используя инструментальные средства Microsoft Visual Studio и Microsoft SQL Server, необходимо разработать информационную систему для фитнес-клуба, а именно:

* Базу данных в SQL-сервере;
* Приложение в среде программирования;

Информационная система должна быть простой и удобной в использовании и иметь такие возможности, как:

* Ввод нового клиента;
* Удаление клиента;
* Обновление информации о клиенте;
* Поиск;
* Фильтрация и сортировка по необходимым критериям.

Шрифты в текстах должны быть легко читаемыми, а интерфейс приятный глазу.

# ПРОЕКТНАЯ ЧАСТЬ

## Проектирование и реализация базы данных системы

Начало разработки базы данных, как правило, происходит посредством построения ER-модели.

ER-модель – модель данных, позволяющая описывать концептуальные схемы предметной области и использующаяся при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-диаграмма представлена на Рис.3.



Рисунок 3 – ER-диаграмма

Разработанная ER-модель состоит из 8 сущностей: Сотрудник, клиенты, роль, виды занятий, абонемент, договор, тренер, категория.

Сущности клиенты, сотрудник, виды занятий, абонемент созданы для подстановки в сущность Договор.

Сущности сотрудник, клиенты, договор, тренер являются основными рабочими сущностями в БД.

Сущности «Сотрудник» и «Тренер» хранят в себе всю необходимую информацию о сотрудниках и тренерах. Они необходимы для учета ведения сотрудников в базе для дальнейшего занесения их в сущность «Договор».

Сущность «Клиент» хранит в себе всю необходимую информацию о клиенте.

Сущность «Договор» хранит в себе всю информацию о клиентах, сотрудниках, абонементах и занятиях.

После проектирования ER-модели был выполнен экспорт в MS SQL - server. Microsoft SQL Server - система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов - Transact-SQL, создан совместно Microsoftи Sybase. Transact - SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. В MS SQL Server были установлены типы данных для каждой таблицы.

На Рис. 4 представлена физическая схема БД, реализованная на платформе MS SQL Server. На модели отображены названия атрибутов, их типы данных на сервере и связи между таблицами.

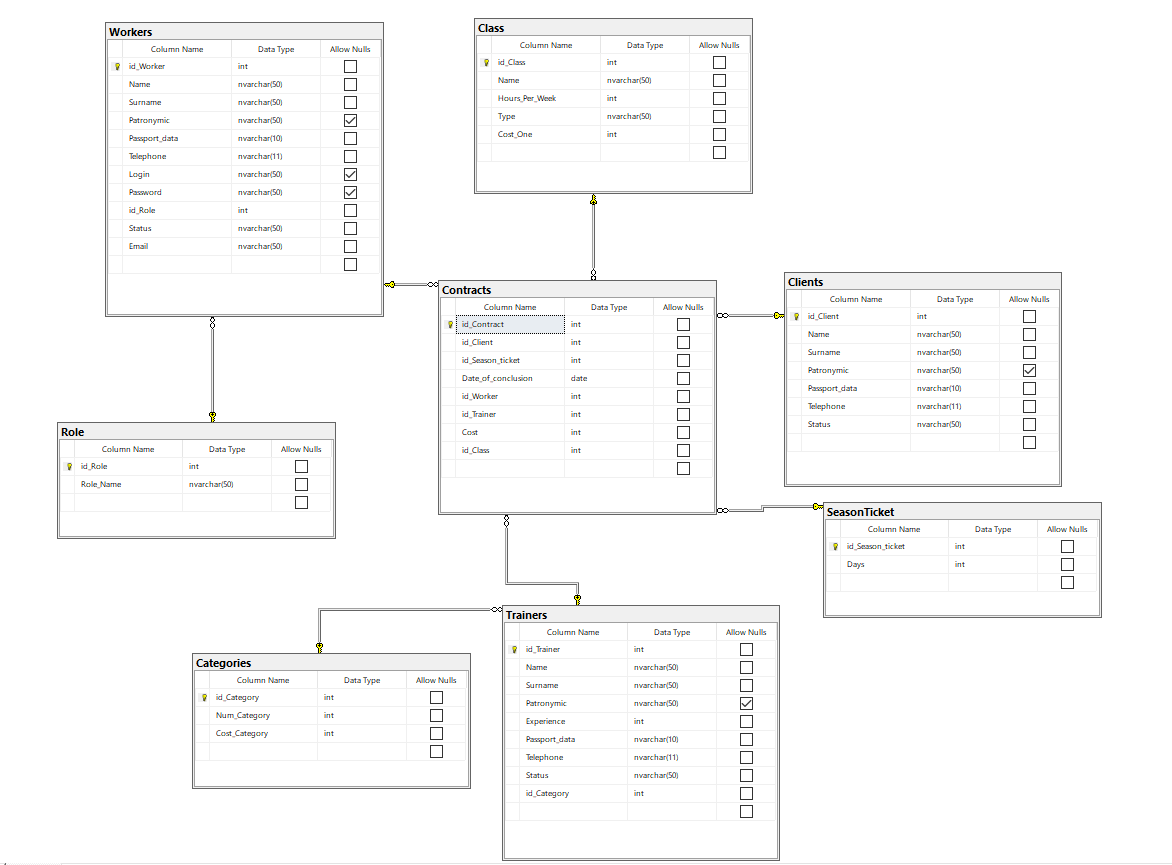


Рисунок 4 – Физическая модель БД

## Проектирование и реализация пользовательского интерфейса системы

Для реализации пользовательского интерфейса была выбрана интегрированная среда разработки Microsoft Visual Studio 2019.

Microsoft Visual Studio 2019 - линейка продуктов компании Microsoft, включающих интегрированную среду разработки программного обеспечения и ряд других инструментов. Данные продукты позволяют разрабатывать как консольные приложения, так и игры, и приложения с графическим интерфейсом.

Система программирования Visual Studio версии 2019 фирмы Microsoft предоставляет наиболее широкие возможности для программирования приложений ОС Windows.

Преимущества Visual Studio по сравнению с аналогичными программными продуктами это:

* Огромные наборы средств и возможностей: позволяет разрабатывать как консольные приложения, так и приложения с графическим интерфейсом, в том числе с поддержкой технологий WPF высокая производительность разработанного приложения;
* Низкие требования разработанного приложения к ресурсам компьютера;
* Удачная проработка иерархии объектов;
* Частично бесплатная;
* Удобна в использовании.
* Так же был использован язык программирования C# так как:
* Множество синтаксического сахара. Синтаксический сахар - это такие конструкции, которые используются для облегчения написания и понимания кода (особенно если это код другого программиста) и не играют роли при компиляции;
* Средний порог вхождения. Синтаксис похожий на C, С ++ или Java облегчает переход для других программистов. Для новичков это также один из самых перспективных языков для изучения;

## Структура приложения

Для реализации поставленных задач была выбрана клиент-серверная архитектура приложения.

Клиент-серверная архитектура (Рис. 5) - это вычислительная или сетевая архитектура, в которой задания или сетевая нагрузка распределены между поставщиками услуг, называемыми серверами, и заказчиками услуг, называемыми клиентами.

Клиент – локальный компьютер на стороне виртуального пользователя, который выполняет отправку запроса к серверу для возможности предоставления данных или выполнения определенной группы системных действий.

Сервер – очень мощный компьютер или специальное системное оборудование, которое предназначается для разрешения определенного круга задач по процессу выполнения программных кодов. Он выполняет работы сервисного обслуживания по клиентским запросам, предоставляет пользователям доступ к определенным системным ресурсам, сохраняет данные или БД.

Параметры, которые могут реализоваться на стороне сервера:

* Хранение, защита и доступ к данным;
* Работа с поступающими клиентскими запросами;
* Процесс отправки ответа клиенту.

Параметры, которые могут реализоваться на стороне клиента:

* Площадка по предоставлению пользовательского графического

интерфейса;

* Формулировка запроса к серверу и его последующая отправка;
* Получение итогов запроса и отправка дополнительной группы команд (запросы на добавление, обновление информации, удаление группы данных).

Рассмотрим преимущества клиент-серверной архитектуры:

* Отсутствие дублирования. Весь сайт или приложение хранится на одном компьютере-сервере. Это позволяет использовать его с разных устройств, будь то компьютер или мобильный телефон;
* Минимальные требования к пользователю. От него требуется только наличие программы-клиента. Для работы с сайтами достаточно иметь браузер;
* Безопасность. Данные хранятся в базе данных и пользователи не могут их просматривать. Это обеспечивает безопасность для персональных данных;
* Производительность. Серверы обычно производительнее, чем компьютеры пользователей. Это позволяет обрабатывать тысячи запросов от сотни разных пользователей одновременно.

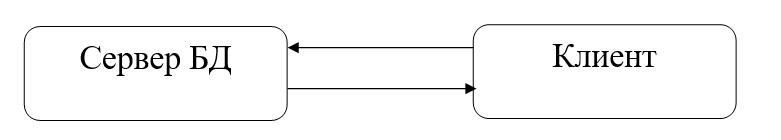


Рисунок 5 – Схема клиент-серверного приложения

Разработанное клиентское приложение выполняет все поставленные задачи и имеет сложную архитектуру, представленную на Рис. 6.



Рисунок 6 – Схема приложения

На схеме разными цветами выделены разграничения прав доступа.

Красным цветом отмечены функции, доступные администратору; Зелёным – старшему администратору; Чёрным – директору;

Синим отмечены функции, доступные для всех трёх групп; Жёлтым отмечены функции, доступные для администратора и старшего администратора.

## Реализация пользовательских форм

В ходе курсового проекта с помощью программного продукта Microsoft Visual Studio было создано приложение. Так же с помощью SQL Server Management Studio была создана база данных, которая была подключена к этому приложению. Приложение реализовано в TDI виде.

TDI (Tabbed Document Interface) - это интерфейс, где каждый документ или окно представлено в виде вкладки в одном основном окне приложения. Пользователь может переключаться между открытыми вкладками для доступа к разным документам или окнам. TDI интерфейс широко используется в веб-браузерах и редакторах кода.

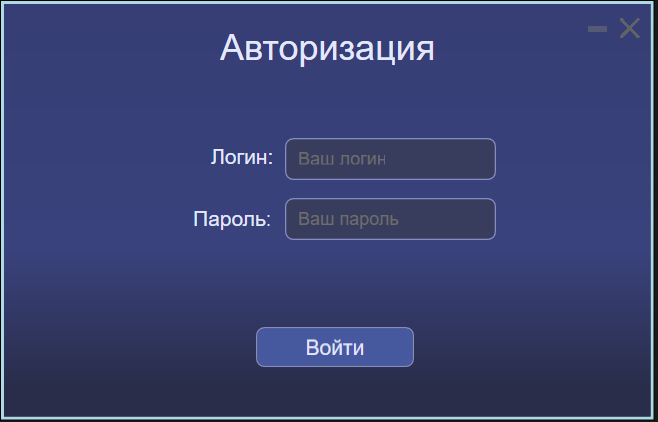
 В начале приложения было разработано два основных окна, а также дополнительные. Первым было разработано окно авторизации, представленное на Рис. 7.

Рисунок 7 – Окно авторизации

Авторизоваться в системе можно по одной из трёх ролей: старшего администратора, администратора, директора.

Код авторизации:

private void btn\_Entry\_Click(object sender, RoutedEventArgs e)

{

kol++;

FitnessApp f = new FitnessApp();

var employees = FitnesEntities.GetContext().Workers;

bool check = false;

string output;

MD5 mD5 = MD5.Create();

byte[] inputBytes = Encoding.UTF8.GetBytes(tb\_Password.Text);

byte[] hash = mD5.ComputeHash(inputBytes);

output = BitConverter.ToString(hash).Replace("-", "");

foreach (var p in employees){

if (tb\_Login.Text == p.Login && tb\_Password.Text == p.Password)

{

kol = 0;

switch (p.id\_Role)

{

case 1:

MessageBox.Show("Администратор" + p.Role.Role\_Name + " " + p.Surname + " " + p.Name + " " + p.Patronymic);

f.tci\_Workers.Visibility = Visibility.Collapsed;

f.btn\_DelClient.Visibility = Visibility.Collapsed;

f.btn\_DelContr.Visibility = Visibility.Collapsed;

f.Show();

break;

case 3:

MessageBox.Show("Директор" + p.Role.Role\_Name + " " + p.Surname + " " + p.Name + " " + p.Patronymic);

f.dg\_Contracts.Margin = new Thickness(-100, 50, 20, 20);

f.dg\_Clients.Margin = new Thickness(-100, 40, 20, 20);

f.dg\_Workwers.Margin = new Thickness(-120, 60, 20, 220);

f.dg\_Trainers.Margin = new Thickness(-120, 260, 20, 20);

f.lbl\_Sot.Margin = new Thickness(-120, 40, 0, 240);

//Margin="56,248,0,0"

f.lbl\_Train.Margin = new Thickness(-120, 240, 0, 0);

f.Show();

f.btn\_AddClient.Visibility = Visibility.Collapsed;

f.btn\_AddContract.Visibility = Visibility.Collapsed;

f.btn\_AddWorker.Visibility = Visibility.Collapsed;

f.btn\_DelClient.Visibility = Visibility.Collapsed;

f.btn\_DelContr.Visibility = Visibility.Collapsed;

f.btn\_DelWorker.Visibility = Visibility.Collapsed;

f.btn\_RedClient.Visibility = Visibility.Collapsed;

f.btn\_RedWorker.Visibility = Visibility.Collapsed;

f.btn\_RedContr.Visibility = Visibility.Collapsed;

break;

case 2:

MessageBox.Show("Старший администратор" + p.Role.Role\_Name + " " + p.Surname + " " + p.Name + " " + p.Patronymic);

f.Show();

break;

}

check = true;

break;

}

}

if (!check )

{

if (kol < 3)

{

MyMessageBox.Show("Ошибка!", "Неверный логин и/или пароль!", MessageBoxButton.OK);

}

}

if (kol >= 3)

{

if( MyMessageBox.Show("Ошибка входа!", "Вы ввели пароль неверно 3/более раз. Отправить запрос на смену пароля главному администратору: ", MessageBoxButton.YesNo) == MessageBoxResult.Yes)

{

infoToMail inf = new infoToMail();

inf.Show();

}

}

}

После авторизации происходит переход на главную форму, откуда уже и осуществляется вся основная работа в зависимости от роли.

Переход между формами реализован следующим образом:

«Название формы» «название переменной» = new «Название формы»();

«название переменной».Show();

На главной форме располагаются вкладки TabItem для перехода по форме на «Договоры», «Клиенты», «Сотрудники», «Отчеты».

У старшего администратора имеется полный доступ к системе, без ограничений (Рис. 8).



Рисунок 8 – Главное окно старшего администратора

У администратора имеется доступ к изменению, добавлению клиентов, договоров и формированию отчетов (Рис. 9).



Рисунок 9 – Главное окно администратора

У директора имеется доступ к просмотру всех клиентов, договоров, сотрудников и формированию отчетов, поиску и сортировке таблиц (Рис. 10).



Рисунок 10 – Главное окно директора

## Разработка запросов

Вся работа с запросами работает с помощью ComboBox. Элемент ComboBox образует выпадающий список и совмещает функциональность компонентов ListBox и TextBox. Для хранения элементов списка в ComboBox также предназначено свойство Items.

В системе есть следующие запросы:

1. В окне «Авторизация»

* Запрос на восстановление пароля (Рис. 11)

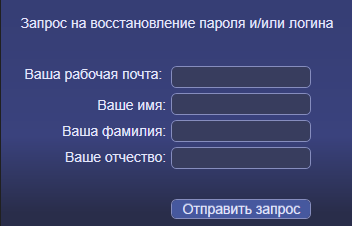


Рисунок 11 – Окно запроса на восстановление

1. В окне «Договоры»

* Запрос на удаление договора;
* Запрос на добавление договора (Рис. 12);
* Запрос на редактирование договора.

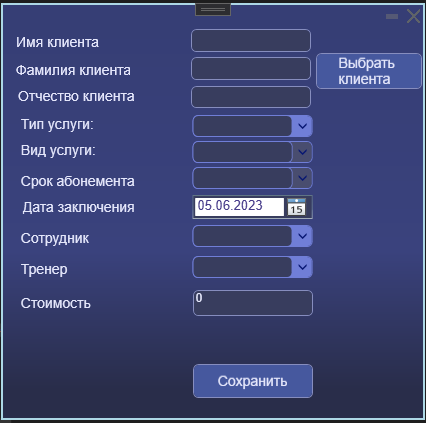


Рисунок 12 – Окно добавления договора

1. В окне «Клиенты»

* Запрос на удаление клиента;
* Запрос на добавление клиента (Рис. 13).
* Запрос на редактирование клиента.

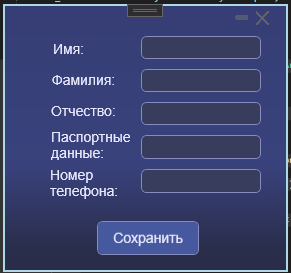


Рисунок 13 –Окно добавления клиента

1. В окне «Сотрудники»

* Запрос на удаление сотрудника/тренера;
* Запрос на добавление сотрудника/тренера; (Рис. 14-15)
* Запрос на редактирование сотрудника/тренера.

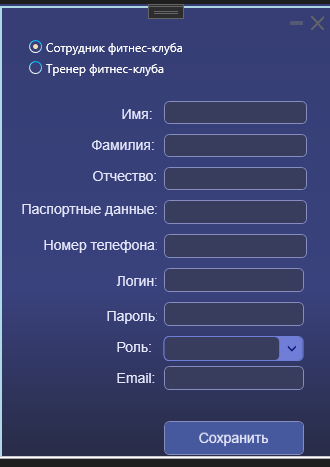


Рисунок 14 – Окно добавления сотрудника фитнес-клуба

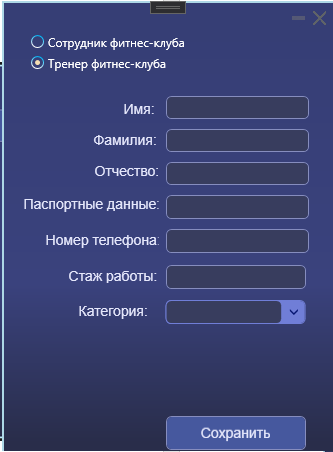


Рисунок 15 – Окно добавления тренера

1. В окне «Отчёт»

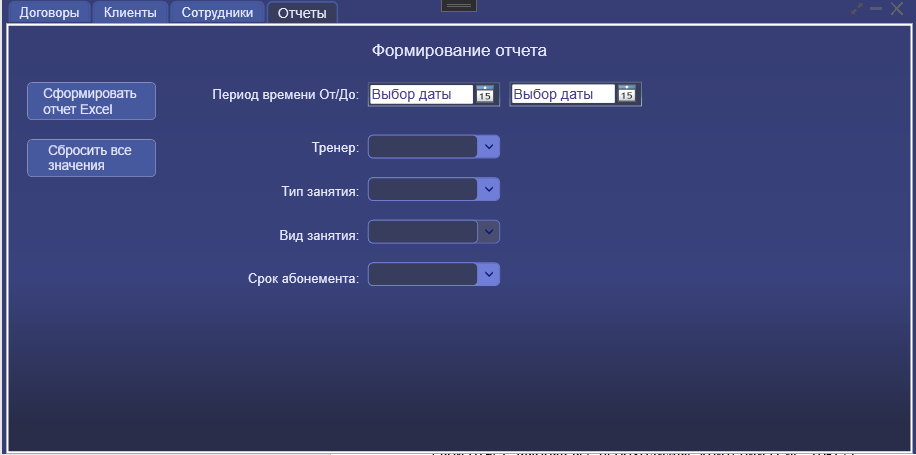
* Запрос на подстановку значений;
* Запрос на формирование отчета (Рис. 16).

Рисунок 16 – Окно формирования отчета

Пример кода добавления:

StringBuilder error = new StringBuilder();

if (cb\_SeasonTicket.SelectedIndex == -1) { error.AppendLine("Укажите продолжительность абонемента"); }

if (cb\_Type.SelectedIndex == -1) { error.AppendLine("Укажите тип занятия"); }

if (cb\_View.SelectedIndex == -1) { error.AppendLine("Укажите вид занятия"); }

if (dp\_Date.SelectedDate==null) { error.AppendLine("Укажите дату заключения"); }

if (cb\_Trainer.SelectedIndex == -1) { error.AppendLine("Укажите тренера"); }

if (cb\_Worker.SelectedIndex == -1) { error.AppendLine("Укажите сотрудника"); }

if (tb\_Name.Text==null) { error.AppendLine("Укажите имя клиента"); }

if (tb\_Surname.Text == null) { error.AppendLine("Укажите фамилию клиента"); }

if (tb\_Patronymic.Text == null) { error.AppendLine("Укажите отчество клиента"); }

FitnessApp app = new FitnessApp();

Clients clients = app.dg\_Clients.SelectedItem as Clients;

try

{

\_contr.id\_Client = Convert.ToInt32(tbl.Text);

\_contr.Date\_of\_conclusion = Convert.ToDateTime(dp\_Date.SelectedDate);

\_contr.Cost = Convert.ToInt32(tb\_Cost.Text);

}

catch { }

if (error.Length > 0) { MyMessageBox.Show("Ошибка сохранения",error.ToString(), MessageBoxButton.OK); return; }

if (\_contr.id\_Contract == 0) { FitnesEntities.GetContext().Contracts.Add(\_contr); }

try

{

if (\_contr.Clients.Status == "Неактивный") { \_contr.Clients.Status = "Активный"; }

app.tci\_Contract.Visibility = Visibility.Visible;

app.tci\_Contract.IsEnabled = true;

FitnesEntities.GetContext().SaveChanges();

MyMessageBox.Show("Уведомление о сохранении","Запись добавлена", MessageBoxButton.OK);

Close();

app.tc.SelectedItem = app.tci\_Contract;

app.Show();

}

catch (Exception ex)

{

MyMessageBox.Show("Уведомление об ошибке сохранения",ex.Message.ToString(), MessageBoxButton.OK);

}

Вне зависимости от роли или выбранной вкладки, пользователь может совершать поиск по таблицам и их сортировку.

Пример кода поиска по таблице «Договоры»:

var con = FitnesEntities.GetContext().Contracts.ToList();

con = con.Where(

p => p.Clients.Name.ToLower().Contains(tb\_SearchCon.Text.ToLower())

|| p.Clients.Surname.ToLower().Contains(tb\_SearchCon.Text.ToLower())

|| p.Clients.Patronymic.ToLower().Contains(tb\_SearchCon.Text.ToLower())

|| p.Clients.Telephone.ToLower().Contains(tb\_SearchCon.Text.ToLower())

|| p.Class.Name.ToLower().Contains(tb\_SearchCon.Text.ToLower())

||p.Class.Type.ToLower().Contains(tb\_SearchCon.Text.ToLower())).ToList();

dg\_Contracts.ItemsSource = con;

Пример кода сортировки таблицы «Договоры»:

switch (cb\_SeasonTicketCon.Text)

{

case "1":

con = con.Where(p => p.SeasonTicket.Days == 1).ToList();

dg\_Contracts.ItemsSource = con;

break;

case "30":

con = con.Where(p => p.SeasonTicket.Days == 30).ToList();

dg\_Contracts.ItemsSource = con;

break;

case "90":

con = con.Where(p => p.SeasonTicket.Days == 90).ToList();

dg\_Contracts.ItemsSource = con;

break;

case "120":

con = con.Where(p => p.SeasonTicket.Days == 120).ToList();

dg\_Contracts.ItemsSource = con;

break;

case "360":

con = con.Where(p => p.SeasonTicket.Days == 360).ToList();

dg\_Contracts.ItemsSource = con;

break; }

## Разработка отчетов

Для разработки отчетов был использован метод работы с программой MS Excel для вывода данных из таблицы в Excel (Рис. 17).

В системе предоставляется возможность «самостоятельно» составить свой отчет, выбрав все необходимые критерии: Дата от/до, тренер, тип занятия, вид занятия, срок абонемента.

Часть кода формирования отчета:

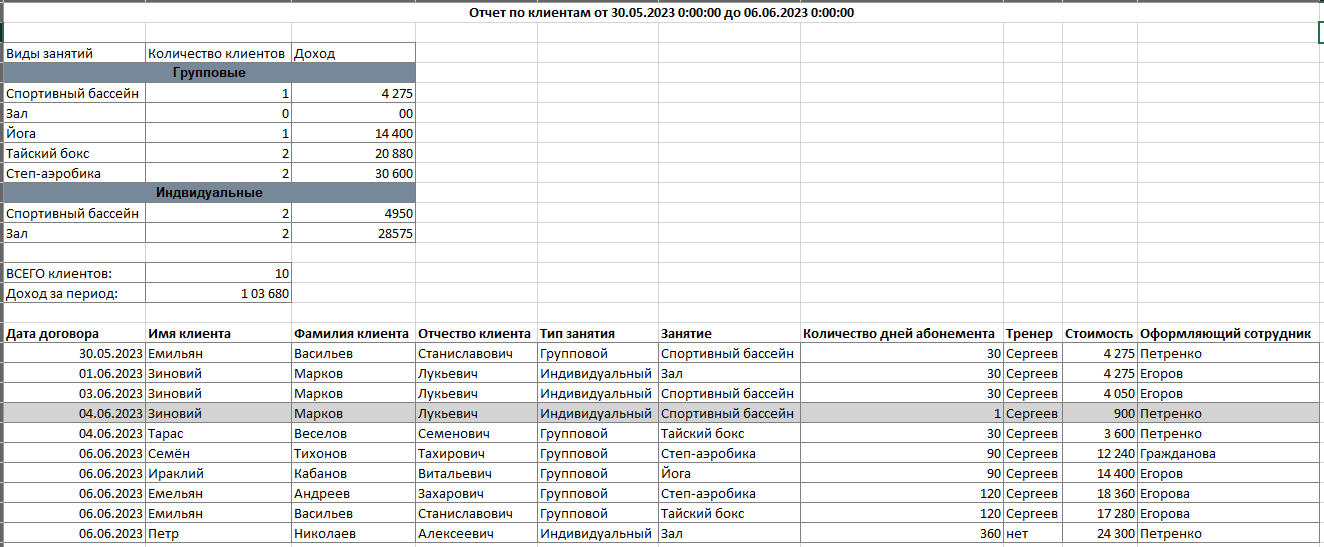
****

Рисунок 17 – Пример отчета

# ЗАКЛЮЧЕНИЕ

Цели поставленные в начале разработки были выполнены полностью, приложение имеет удобный и простой интерфейс, а также выполняет все необходимые действия.

Разработанное приложение может быть использовано в фитнес-клубах.

Целью данной информационной системы является своевременное обеспечение надлежащих людей надлежащей [информацией](https://ru.wikipedia.org/wiki/%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D1%8F), то есть для удовлетворения конкретных информационных потребностей в рамках определенной [предметной области](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0%D1%8F_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C), при этом результатом функционирования информационных систем является информационная продукция - документы, [информационные массивы](https://ru.wikipedia.org/w/index.php?title=%D0%98%D0%BD%D1%84%D0%BE%D1%80%D0%BC%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D1%8B%D0%B9_%D0%BC%D0%B0%D1%81%D1%81%D0%B8%D0%B2&action=edit&redlink=1), [базы данных](https://ru.wikipedia.org/wiki/%D0%91%D0%B0%D0%B7%D0%B0_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) и информационные услуги, и снижение рутиной работы с бумагами во время работы с курсантами.

Во время выполнения курсового проекта были использованы следующие программные продукты:

* Visual Studio 2022
* MS SQL Server
* SQL Server Management Studio
* MS Visio

Приложение может:

* Формировать отчеты;
* Осуществлять учет клиентов;
* Добавлять новых сотрудников;

Выполняя данный курсовой проект, была более углубленно изучена работа с Visual Studio 2022 с MS SQL Server.

# Список литературы

1. [Белов В.В.](http://www.academia-moscow.ru/authors/detail/47801/) , [Чистякова В. И.](http://www.academia-moscow.ru/authors/detail/47802/) [Проектирование информационных систем](http://www.academia-moscow.ru/catalogue/4889/168066/). - Под редакцией: [Белов В. В.](http://www.academia-moscow.ru/authors/detail/47801/)М.: Академия, 2019.
2. Рыбальченко М.В. Архитектура информационных систем: учеб. пособие для СПО. – М.: Издательство Юрайт,2019.- 91с.
3. Чистов Д.В. Проектирование информационных систем: учебник и практикум для СПО. – М.: Издательство Юрайт, 2019 – 258 с.
4. [Федорова Г.Н.](http://www.academia-moscow.ru/authors/detail/46229/) [Разработка программных модулей программного обеспечения для компьютерных систем](http://www.academia-moscow.ru/catalogue/4889/195591/) – М.: Академия, 2019
5. Грекул, В. И. Проектирование информационных систем: учебник и практикум для среднего профессионального образования / В. И. Грекул, Н. Л. Коровкина, Г. А. Левочкина. — Москва: Издательство Юрайт, 2020. — 385 с.
6. Игнатьев А. В.Методы и средства проектирования информационных систем и технологий. Учебное пособие - Волгоград: ВолгГАСУ, 2019
7. Казанский, А. А.  Программирование на Visual C# : учебное пособие для среднего профессионального образования / А. А. Казанский. — 2-е изд., перераб. и доп. — Москва : Издательство Юрайт, 2023. — 192 с. — (Профессиональное образование). — ISBN 978-5-534-14130-6. Образовательная платформа Юрайт [сайт].
8. КоцюбаИ.Ю., Чунаев А.В., Шиков А.Н. Основы проектирования информационных систем. Учебное пособие. - Университет ИТМО, 2019
9. Подбельский, В. В.  Программирование. Базовый курс С# : учебник для среднего профессионального образования / В. В. Подбельский. — Москва: Издательство Юрайт, 2023. — 369 с. — (Профессиональное образование). — ISBN 978-5-534-11467-6. — Текст: электронный // Образовательная платформа Юрайт [сайт].
10. Проектирование информационных систем: курс лекций: учебное пособие : [16+] / авт.- сост. Т.В. Киселева ; Ставрополь : СКФУ, 2019. – Ч. Часть 1. – 150 с.: ил. – Режим доступа: по подписке. – URL: http://biblioclub.ru/index.php?page=book&id=563326– Текст: электронный.

# Приложение А. Техническое задание на курсовой проект