

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н. П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра)
общепрофессиональных и специальных (информационно-коммуникационных)
дисциплин

УТВЕРЖДАЮ
Зам. декана по учебной
работе, преподаватель ВКК
_____ В. А. Богатырская
(подпись)
«14» июня 2020г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К ДИПЛОМНОМУ ПРОЕКТУ

на тему Разработка обучающей программы по решению задач линейного
программирования

Автор проекта	30.05.2020	К. С. Горюнов
(подпись)	(дата)	

Специальность 09.02.03 Программирование в компьютерных системах

Обозначение дипломного проекта ДП–02069964– 09.02.03–04–20

Руководитель проекта

преподаватель ПКК	04.06.2020	Н. А. Базеева
(подпись)	(дата)	

Нормоконтролер

преподаватель ВКК	02.06.2020	М. Ф. Петянкин
(подпись)	(дата)	

Рецензент

канд. пед. наук, зав. каф.
ИТ и математики АНОО ВО ЦС РФ
«Российский университет кооперации»
Саранский кооперативный
институт (филиал)

	08.06.2020	С. С. Голяев
(подпись)	(дата)	

Саранск
2020

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ МОРДОВСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.П. ОГАРЁВА»

Факультет довузовской подготовки и среднего профессионального образования

Выпускающая предметная цикловая комиссия (кафедра)
общепрофессиональных и специальных (информационно-коммуникационных)
дисциплин

УТВЕРЖДАЮ
Зам. декана по учебной
работе, преподаватель ВКК
_____ В. А. Богатырская
(подпись)
«28» ноября 2019 г.

ЗАДАНИЕ НА ДИПЛОМНЫЙ ПРОЕКТ

Студент Горюнов Кирилл Сергеевич

1 Тема Разработка автоматизированного рабочего места администратора фитнес-клуба.

Утверждено приказом № 1605-с от 20.02.2020

2 Срок представления работы к защите 14.06.2020

3 Исходные данные для научного исследования (проектирования)

Реализовать автоматизированную информационную систему с интуитивно понятным интерфейсом пользователя, которая выполняет следующие функции:

- а) реализовать базу данных;
- б) создать формы для интерфейса;
- в) разработать дизайн информационной системы;
- д) реализовать добавление, редактирование, удаление и сохранение данных;
- е) добавить в приложение возможность отчетности.

4 Содержание выпускной квалификационной работы

4.1 Теоретическое обоснование проекта

- 4.2 Анализ предметной области
- 4.3 Методы и средства разработки и реализации
- 4.4 Составление технического задания
- 4.5 Проектирование базы данных
- 4.6 Реализация автоматизированной информационной системы
- 4.7 Описание инструмента моделирования StarUML
- 4.8 Диаграмма последовательности
- 4.9 Диаграмма компонентов
- 4.10 Диаграмма развертывания
- 4.11 Дизайн автоматизированной информационной системы
- 4.12 Тестирование программного продукта

5 Приложения:

5.1 ПРИЛОЖЕНИЕ А (обязательное) Листинг страницы «Окно авторизации пользователей»

5.2 ПРИЛОЖЕНИЕ Б (обязательное) Листинг страницы «Главное окно приложения»

5.3 ПРИЛОЖЕНИЕ В (обязательное) Листинг страницы «Тренажеры и снаряды»

5.4 ПРИЛОЖЕНИЕ Г (обязательное) Листинг страницы «Абонементы клиентов»

5.5 ПРИЛОЖЕНИЕ Д (обязательное) Листинг страницы «Клиенты»

5.6 ПРИЛОЖЕНИЕ Е (обязательное) Листинг страницы «Расписание»

5.7 ПРИЛОЖЕНИЕ Ж (обязательное) Листинг страницы «Инвентарь клуба»

5.8 ПРИЛОЖЕНИЕ И (обязательное) Техническое задание

5.9 ПРИЛОЖЕНИЕ К (обязательное) Руководство пользователя

Руководитель работы

28.11.19

подпись, дата

Н. А. Базеева

Задание принял к исполнению

28.11.19

подпись, дата

К. С. Горюнов

РЕФЕРАТ

Пояснительная записка по содержит 95 страниц, 14 рисунков, 5 использованных источников, 8 приложений.

ПРОЕКТИРОВАНИЕ, ДОКУМЕНТАЦИЯ, ДИАГРАММА,
РАЗРАБОТКА, ИНТЕРФЕЙС.

Объектом исследования является автоматизированное рабочего места администратора фитнес-клуба.

Цель работы — разработка автоматизированного рабочего места администратора фитнес-клуба.

В процессе работы проводился анализ предметной области, анализ автоматизированное рабочего места администратора фитнес-клуба, выбор методов и средств реализации.

Степень внедрения — частичная.

В результате проведенной работы были спроектированы база данных, даталогическая модель данных, ER-диаграммы и интерфейс.

Разработанное автоматизированное рабочего места администратора фитнес-клуба используется для учета, как абонементов и инвентаря, так и для составления отчетов.

					ДП-02069964-09.02.03-04-20				
Изм.	Лист	№ докум.	Подпись	Дата					
Разраб.		Горюнов			Разработка автоматизированного рабочего места администратора фитнес-клуба»		Лит.	Лист	Листов
Пров.		Базеева					Д	4	95
							ФДП и СПО, д/о, ПВКС, 409		
Н. контр.		Петянкин							
Утв.		Богатырская							

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	10
1 Теоретическое обоснование проекта	11
1.1 Анализ предметной области	11
1.2 Методы и средства разработки и реализации	12
2 Составление технического задания	19
3 Проектирование базы данных	21
4 Реализация автоматизированной информационной системы	25
4.1 Описание программного инструмента моделирования StarUML	25
4.2 Диаграмма последовательности	26
4.3 Диаграмма компонентов	27
4.4 Диаграмма развертывания	28
4.5 Дизайн автоматизированной информационной системы	29
4.6 Создание проекта в Visual Studio	30
4.7 Реализация интерфейса программного продукта	30
5 Тестирование программного продукта	38
ЗАКЛЮЧЕНИЕ	41
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	43
ПРИЛОЖЕНИЕ А (обязательное) Листинг страницы «Окно авторизации пользователей»	44
ПРИЛОЖЕНИЕ Б (обязательное) Листинг страницы «Главное окно приложения»	44
ПРИЛОЖЕНИЕ В (обязательное) Листинг страницы «Тренажеры и снаряды»	52
ПРИЛОЖЕНИЕ Г (обязательное) Листинг страницы «Абонементы клиентов»	54
ПРИЛОЖЕНИЕ Д (обязательное) Листинг страницы «Клиенты»	56
ПРИЛОЖЕНИЕ Е (обязательное) Листинг страницы «Расписание»	60
ПРИЛОЖЕНИЕ Ж (обязательное) Листинг страницы «Инвентарь клуба»	64

ПРИЛОЖЕНИЕ И (обязательное) Техническое задание

68

ПРИЛОЖЕНИЕ К (обязательное) Руководство пользователя

89

					ДП-02069964-09.02.03-04-20	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

ВВЕДЕНИЕ

Цель каждой компании — получение прибыли. Однако, если эта цель может быть достигнута путем улучшения производства и продукции, в настоящее время, помимо предоставления высококачественных продуктов или услуг, также требуются высокое качество обслуживания. Ведение бизнеса в соответствии с требованиями клиентов — это не желание, а необходимое условие для выживания бизнеса, поэтому клиенты, их потребности и требования стали главными руководящими принципами для руководства компаний в ведении бизнеса.

Компьютерные системы, в области коммуникаций и хранения данных, прогрессируют несколько стремительней, чем остальные, потому что им доводится идти в ногу с технологическими и социальными изменениями в мире. В связи с тем, что объемы информации, подлежащей хранению, растут высокими темпами, изготовители программного обеспечения обязаны разрабатывать новые эластичные подходы к управлению крупными объемами данных.

Поскольку базы данных, вне любых сомнений, занимают главенствующее положение в области информационных технологий, они становятся неотделимой частью жизни современного человека. Становление систем управления этими процессами идет по пути интеграции в цельное целое, и большим шагом к этому является применение эталона обработки данных с поддержкой структурированного языка запросов SQL. Программное обеспечение, работающее с применением SQL работает все больше неprimетно для пользователя, предоставляя ему нужную основу для манипуляции информацией. В реальное время ударение делается на применение сетевых информационных технологий, основывающихся на архитектуре клиент/сервер, и переходе от малопонятных внутренних процедур к обстоятельной работе с предметной областью, которую описывают хранимые данные.

					ДП-02069964-09.02.03-04-20	Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

Современные СУБД в основном являются приложениями Windows, так как данная среда разрешает больше полно применять вероятности персональной ЭВМ, нежели среда DOS. Снижение стоимости высокопроизводительных ПК обусловил не только широкий переход к среде Windows, где разработчик программного обеспечения может в поменьше степени заботиться о разделении источников, но также сделал программное обеспечение ПК в целом и СУБД в частности менее критичными к аппаратным источникам ЭВМ.

Среди особенно блестящих поверенных систем управления базами данных дозволено подметить: Lotus Approach, Microsoft Access, Borland dBase, Borland Paradox, Microsoft Visual FoxPro, Microsoft Visual Basic, а также баз данных Microsoft SQL Server и Oracle, используемые в приложениях, построенных по технологии «клиент-сервер». Реально, у всякий нынешней СУБД существует аналог, выпускаемый иной компанией, имеющий аналогичную область использования и вероятности, всякое приложение способно трудиться со многими форматами представления данных, осуществлять экспорт и импорт данных вследствие наличия большого числа конвертеров.

Таким образом, на сегодняшний день разработчик не связан рамками какого-либо определенного пакета, а в зависимости от поставленной задачи может применять самые различные приложения. Следственно, больше значимым представляется всеобщее направление становления СУБД и других средств разработки приложений в реальное время.

Цель дипломного работы — разработка автоматизированного рабочего места администратора фитнес-клуба.

В ходе работы над проектом будут найдены решения для следующего комплекса задач:

- провести анализ предметной области;
- провести анализ и выбор программного обеспечения;
- спроектировать даталогическую модель базы данных;
- спроектировать ER-диаграммы, выявить связи между сущностями;
- привести базу данных к Третьей нормальной форме;

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

- создать интерфейс автоматизированной информационной системы;
- сформулировать текст запросов к созданной базе данных;
- провести тестирование программного продукта.

На основании анализа поставленных задач были выбраны методы и инструментальные средства для разработки автоматизированной информационной системы. В частности, была выбрана система управления базами данных Microsoft SQL Server 2019, а также язык объектно-ориентированный и тип безопасный язык программирования C#.

Данный дипломный проект рассматривает под понятием объект исследования — фитнес клуб.

Предметом исследования является — автоматизация информационный системы фитнес-клуба.

1 Теоретическое обоснование проекта

1.1 Анализ предметной области

В связи с ростом востребованности получения фитнес услуг стала актуальной проблема улучшения производительности и качества работы фитнес-клубов.

Для успешной и продуктивной работы предприятия требуется комплексное внедрение информационно-коммуникационных технологий (ИКТ).

Основным действием перед введением в работу фитнес-клуб или какого-либо другого предприятия автоматизированной информационной системы является определение основных требований деятельности этой организации. Чаще всего решить данную проблему помогает исследование предметной области, на основе которой уже делаются все выводы.

Стоит прояснить значение понятия «Предметная область». В пределах какого-либо контекста рассматривается комплекс объектов настоящего или предполагаемого мира. Этот комплекс понимается как индивидуальное рассуждение, научная теория и т.д.

Определить область автоматизации какой-либо сферы деятельности некой организации (в этом случае — работа фитнес клуба) можно путем определения проблем и слабых сторон предметной области [1]. Достижение успешной разработки и внедрения базы данных можно обеспечить разработкой модели предметной области.

Фитнес-клуб — это объект спортивно-оздоровительного назначения, предоставляющий комплекс фитнес-услуг. Особенностью работы фитнес-клуба является использование системы клубных абонементов — так называемых карт членов клуба. Клубный абонемент, или членская карта, дает возможность

неограниченного времени пребывания в клубе в часы его работы и неограниченного количества посещений в период действия клубного абонементa.

Вся работа состоит из большого количества повторяющихся действий и однотипной информации, не исключено возникновение ошибок, замедление и усложнение всей деятельности предприятия.

Создаваемая база данных предназначена для структурирования и хранения данных, используемых в работе фитнес клубе. В базе данных ведется учет клиентов, абонементов, видов услуг, список сотрудников и продажа абонементов.

1.2 Методы и средства разработки и реализации

Когда появляется необходимость разработки программного модуля, остро встает вопрос выбора целевой платформы разработки, среды разработки и языка программирования, с помощью которого разрабатываемый модуль будет реализован.

В качестве целевой платформы разработки была выбрана десктоп-конфигурация с операционной системой (далее ОС) от компании Microsoft Windows, так как данная конфигурация является модульной и её при необходимости можно усовершенствовать, а не покупать новую; а ОС Windows является достаточно распространённой, имеет большой набор дополнительного инструментария и не требует специальных.

В качестве языка программирования для реализации поставленной задачи выбран объектно-ориентированный язык программирования от компании Microsoft C# и технология Windows Presentation Foundation (WPF) на платформе .NET Framework.

					ДП-02069964-09.02.03-04-20	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

C# — простой, современный объектно-ориентированный и типобезопасный язык программирования. C# относится к широко известному семейству языков C.

C# — это объектно-ориентированный язык, который также поддерживает компонентно-ориентированное программирование. Создание программных компонентов в форме независимых программных пакетов с самоописанием, которые реализуют различные функции, все больше привлекает разработку современных приложений. Важной функцией этого компонента является модель программирования, основанная на атрибутах, методах и событиях. Каждый компонент имеет атрибуты, которые предоставляют декларативную информацию о компоненте и встроенных элементах документа. C# предоставляет языковую конструкцию, которая напрямую поддерживает рабочую концепцию. Благодаря этому C# отлично подходит для создания и применения программных компонентов.

Конкуренцию данному языку в десктоп-разработке могут составить такие, как C++, Java, JavaScript.

Для разработки десктоп-приложения на языке программирования C++ необходим либо знания фреймворка Qt, либо знания WIN API. Разработка с использованием WIN API является долгим и сложным процессом, в отличие от разработки с применением Qt, так как он по сути является конструктором и является аналогом Windows Forms для C#. В отличие от приложения, разработанного с помощью C# Windows Forms, приложение C++ Qt является кроссплатформенным решением и не привязано к платформе .NET Framework, а, следовательно, и к операционной системе Windows. Однако сам язык программирования C++ является более сложным и более низкоуровневым в сравнении с C#. Примером этого является механизм указателей и ссылок для работы с памятью, а также необходимость «вручную» выделять и освобождать память от объектов, что в неумелых руках может привести к утечке памяти и снижению производительности.

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

Следующий конкурент, Java, является кроссплатформенным высокоуровневым языком программирования от компании Oracle. В отличие от C++ работа с памятью в данном языке происходит автоматически — выделение памяти под объекты и её освобождение автоматизированы — что частично решает проблему перерасхода памяти при разработке приложения новичком. Кроссплатформенность достигается при помощи использования виртуальной машины Java, преобразующей код Java в специальный байт-код на этапе компиляции. Для разработки графического пользовательского интерфейса с использованием Java существуют несколько фреймворков: Swing, AWT, JavaFX и так далее. Однако ни один из приведенных фреймворков не даёт нужного функционала для отдельной разработки пользовательского интерфейса и логики приложения, так как обе части пишутся на самом языке Java, что увеличивает время и сложность разработки.

JavaScript является интерпретируемым языком программирования, первоначально созданным для web-разработки. Со временем язык сильно развился и в настоящее время используется в большинстве сфер программирования, начиная web-программирование и заканчивая мобильной разработкой [5]. Одной из проблем в разработке может послужить динамическая типизация языка, что при недостатке знаний может приводить к фатальным ошибкам. Для разработки десктоп приложения на JavaScript требуется система Electron JS, позволяющая использовать в качестве пользовательского интерфейса разметку, созданную с помощью HTML и CSS. Однако, для работы приложений, созданных с помощью Electron JS необходим движок Chromium, используемый в таких браузерах как Google Chrome и Opera, и, входящий в саму систему Electron JS, что увеличивает объем памяти, занимаемый приложением. Так приложение, разработанное с помощью C# WPF .NET Framework, имеющее объем от единиц до двух десятков мегабайт, при разработке с помощью Electron JS будет занимать порядка 100 мегабайт. Также так как JavaScript является интерпретируемым языком программирования, то и

уровень производительности данного приложения будет сравнительно ниже.

При разработке десктоп приложения на C# есть несколько технологий, позволяющих разрабатывать пользовательский интерфейс приложения: Windows Forms (WF), Windows Presentation Foundation (WPF), Universal Windows Platform (UWP).

WF является самой простой и самой старой из выше перечисленных технологий. Разработка с помощью данной технологии ведется быстро, так как WF является конструктором окон. С помощью WF очень трудно разрабатывать сложный интерфейс, а при масштабировании окон приложение будет выглядеть не самым лучшим образом. На сегодняшний день данная технология является устаревшей.

WPF является технологией разработки более гибкого пользовательского интерфейса на платформе .NET. Данная технология использует декларативный язык разметки XAML, созданный на основе XML. Это позволяет отделить процесс разработки интерфейса от разработки логики приложения, однако из-за этого скорость разработки, в сравнении с WF, значительно ниже. WPF предназначена для разработки сугубо десктоп приложений, однако код таких приложений в дальнейшем можно перенести в проекты UWP и Xamarin, так как они также используют XAML и C#. Данная технология не поддерживается операционными системами, более ранними, чем Windows 7.

UWP является самой молодой из выше перечисленных технологий. Она также, как и WPF использует XAML для создания пользовательского интерфейса. [3] Отличием данной технологии от своего предшественника является то, что приложения пусть и разрабатываются лишь для Windows, однако теперь могут быть запущены и на мобильных устройствах с Windows Mobile в качестве ОС. Однако данная технология поддерживается лишь Windows 10 и Windows Mobile.

Для работы с базой данных используется технология Entity Framework 6, в силу отсутствия необходимости для программиста знания языка

					ДП-02069964-09.02.03-04-20	Лист
						14
Изм.	Лист	№ докум.	Подпись	Дата		

структурированных запросов SQL. Её аналогом является традиционная технология для работы с базами данных на платформе .NET Framework — ADO.NET.

ADO.NET предоставляет собой технологию работы с данными, которая основана на платформе .NET Framework. Эта технология предоставляет набор классов, через которые осуществляется механизм отправки запросов к базам данных, установления подключения, получения ответов от базы данных и осуществления ряда других операций.

Функционал ADO.NET построен таким образом, чтобы предоставить разработчикам унифицированный интерфейс для работы с самыми различными СУБД.

Основу интерфейса взаимодействия с базами данных в ADO.NET представляет ограниченный круг объектов:

- Connection — установление подключения к источнику данных;
- Command — выполнение операций с данными в базе данных;
- DataReader — считывание полученных в результате запроса данных;
- DataSet — предназначен для хранения данных из базы данных и позволяет работать с ними независимо от самой базы данных;
- DataAdapter — посредник между DataSet и источником данных.

Главным образом, через эти объекты и будет идти работа с базой данных.

Entity Framework представляет специальную объектно-ориентированную технологию на базе .NET Framework для работы с данными. Если вы напрямую используете инструменты ADO.NET для предоставления доступа к командам и другим объектам для взаимодействия с базой данных, Entity Framework представляет более высокий уровень абстракции, который позволяет абстрагироваться от всех баз данных независимо от типа хранилища. Entity Framework, работа с объектами, операции с таблицами, индексами, первичными ключами и внешними ключами.

Центральной концепцией Entity Framework является понятие сущности

(entity). Сущность представляет собой набор данных, ассоциированных с определенным объектом (класс). Поэтому данная технология предполагает работу не с таблицами, а с объектами и их наборами.

Важной особенностью Entity Framework является использование запросов, которые компилируются с использованием интегрированного языка запросов LINQ для выбора конкретных данных из базы данных [2].

2 Составление технического задания

Техническое задание на разработку и реализацию автоматизированной информационной системы фитнес-клуба составлено и оформлено в соответствии ГОСТ 34.602-89 [1] и содержит следующие разделы, которые могут быть разделены на подразделы:

- введение;
- общие сведения;
- назначение разработки;
- требования к программе или программному изделию;
- требования к программной документации;
- технико-экономические показатели;
- стадии и этапы разработки;
- порядок контроля и приемки.

В разделе «Введение» указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

В разделе «Основания для разработки» должны быть указаны:

- документ (документы), на основании которых ведется разработка;
- организация, утвердившая этот документ, и дата его утверждения;
- наименование и (или) условное обозначение темы разработки.

В разделе «Назначение разработки» должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

Раздел «Требования к программе или программному изделию» должен содержать следующие подразделы:

- требования к функциональным характеристикам;
- требования к надежности;

- требования к составу и параметрам технических средств;
- требования к информационной и программной совместимости;
- требования к маркировке и упаковке;
- требования к транспортированию и хранению;
- специальные требования.

В подразделе «Требования к функциональным характеристикам» должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам.

В подразделе «Требования к надежности» должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа).

В подразделе «Требования к составу и параметрам технических средств» указывают необходимый состав технических средств с указанием их основных технических характеристик.

В подразделе «Требования к информационной и программной совместимости» должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой.

В подразделе «Требования к маркировке и упаковке» в общем случае указывают требования к маркировке программного изделия, варианты и способы упаковки.

В подразделе «Требования к транспортированию и хранению» должны быть указаны для программного изделия условия транспортирования, места хранения, условия хранения, условия складирования, сроки хранения в различных условиях.

В разделе «Требования к программной документации» должны быть указаны предварительный состав программной документации и, при необходимости, специальные требования к ней.

В разделе «Технико-экономические показатели» должны быть указаны: ориентировочная экономическая эффективность, предполагаемая годовая потребность, экономические преимущества разработки по сравнению с лучшими отечественными и зарубежными образцами или аналогами.

В разделе «Стадии и этапы разработки» устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

В разделе «Порядок контроля и приемки» должны быть указаны виды испытаний и общие требования к приемке работы.

Составленное техническое задание представлено в приложении И.

					ДП-02069964-09.02.03-04-20	Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

3 Проектирование базы данных

Для проектирования базы данных первоначально необходимо определить сущности исходя из их реальных характеристик. После того как мы определили сущности и их атрибуты необходимо определить связи между ними. И построить даталогическую модель базы данных, перед этим убедившись, что все отношения приведены к 3НФ (3 нормальная форма). Даталогическая модель представлена на рисунке 1.

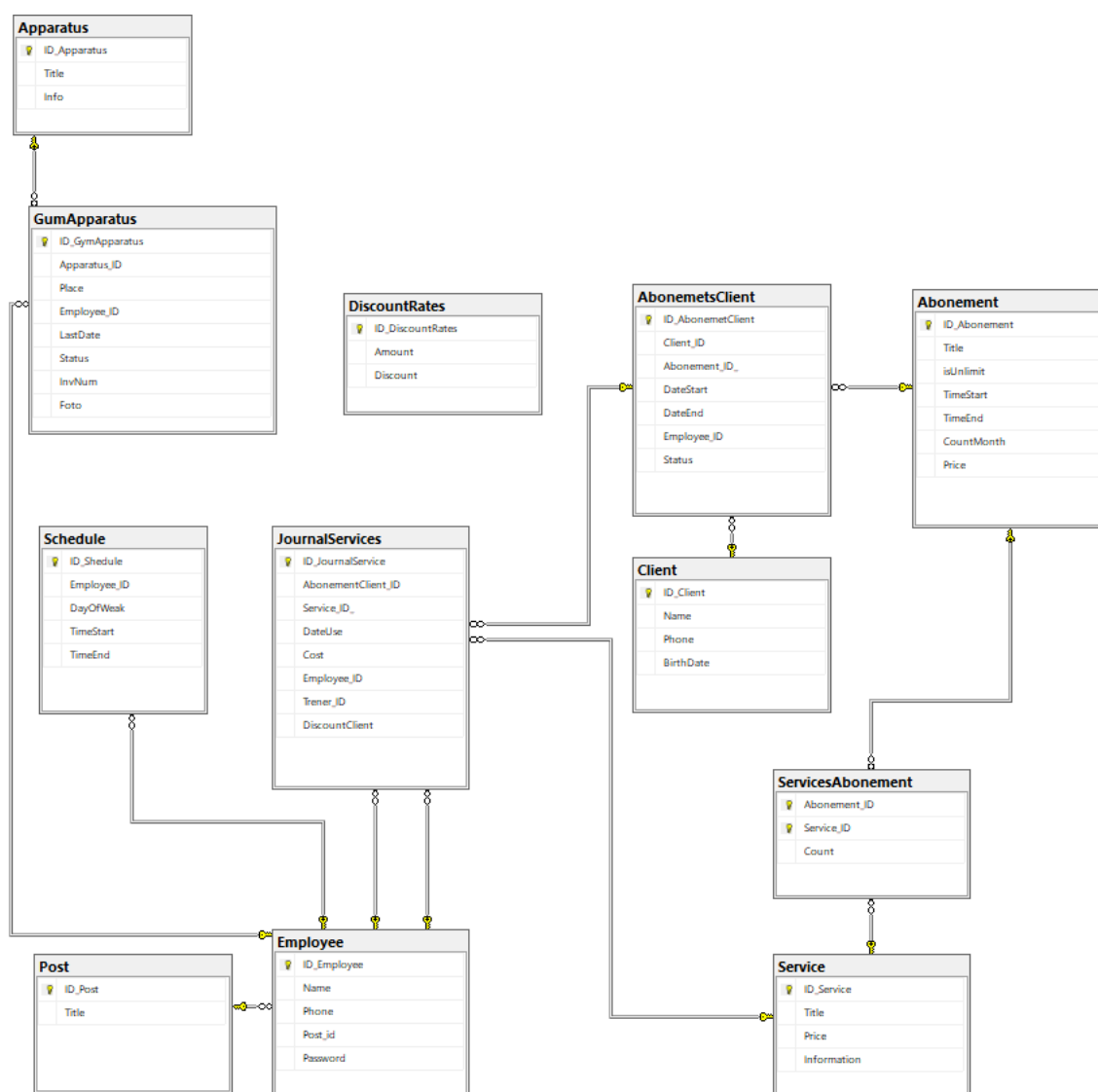


Рисунок 1 — Даталогическая модель

Ниже рассмотрим структуру каждой таблицы отдельно:

а) Таблица Apparatus:

- 1) ID_ Apparatus – первичный ключ;
- 2) Title –название аппарата;
- 3) Info информация об аппарате;

б) GumApparatus:

- 1) ID_GymApparatus – первичный ключ;
- 2) Apparatus_ID– номер аппарата;
- 3) Place – место;
- 4) Employee_ID – номер работника;
- 5) LastDate – последняя дата;
- 6) Status – статус аппарата;
- 7) InvNum индентифицирующий номер аппарата;
- 8) Foto – фотография аппарата;

в) Таблица DiscountRates:

- 1) ID_DiscountRates – первичный ключ;
- 2) Amount – сумма;
- 3) Discount – скидка;

г) Таблица AbonementsClient:

- 1) ID_AbonemetClient – первичный ключ;
- 2) Client_ID – номер клиента;
- 3) Abonement_ID_ – номер абонемент;
- 4) DateStart – начало действия;
- 5) DateEnd – окончания действия;
- 6) Employee_ID – номер работника;
- 7) Status поле, показывающее является ли пользователь администратором (1 – является, 0 – не является);

д) Таблица Abonement:

- 1) ID_Abonement – первичный ключ;

- 2) Title названия абонементов;
 - 3) isUnlimit;
 - 4) TimeStart – начало действия абонемента;
 - 5) TimeEnd – окончание действия абонемента;
 - 6) CountMonth – действие абонемента в месяцах;
 - 7) Price – цена;
- е) Таблица ID_Schedule:
- 1) ID_Schedule– первичный ключ;
 - 2) Employee_ID – номер работника;
 - 3) DayOfWeak;
 - 4) TimeStart – начало расписания;
 - 5) TimeEnd – окончание расписания;
- ж) Таблица JournalService:
- 1) ID_JournalService– первичный ключ;
 - 2) AbonementClient_ID – номер абонемента клиента;
 - 3) Service_ID_ – номер услуги;
 - 4) Cost – цена услуги;
 - 5) Employee_ID – номер работника;
 - 6) Trener_ID – номер тренера;
 - 7) DiscountClient – скидка клиента;
- з) Таблица Client:
- 1) ID_Client– первичный ключ;
 - 2) Name – ФИО клиента;
 - 3) Phone – номер клиента;
 - 4) BirthDate – дата рождения клиента;
- и) Таблица ServicesAbonement:
- 1) Abonement_ID – первичный ключ;
 - 2) Service_ID – номер услуги;
 - 3) Count – услуга;

- к) Таблица Post:
- 1) ID_Post – первичный ключ;
 - 2) Title – описание должности;
- л) Таблица Employee:
- 1) ID_Employee – первичный ключ;
 - 2) Name – имя работника;
 - 3) Phone – телефон работника;
 - 4) Post_id – номер должности;
 - 5) Password – пароль работника;
- м) Таблица Employee:
- 1) ID_Service– первичный ключ;
 - 2) Title – описание услуги;
 - 3) Price – цена услуги;
 - 4) Information – информация об услуги.

4 Реализация автоматизированной информационной системы

4.1 Описание программного инструмента моделирования StarUML

StarUML — это программа для разработки быстрой, гибкой, расширяемой, многофункциональной и бесплатной платформы UML, работающей на платформе Win32. Целью проекта StarUML является создание инструмента моделирования программного обеспечения, а также платформы, которая является убедительной заменой коммерческих инструментов UML, таких как Rational Rose, Together и так далее.

Главные особенности:

- точное соответствие стандарту UML;
- открытый формат программной модели;
- применимость методологий и платформ;
- превосходная расширяемость;
- программная функция проверки модели.

4.2 Диаграмма последовательности

Взаимодействие объектов в системе происходит через клиентский объект, принимающий и передающий сообщения и обрабатывающий сообщения через серверный объект. В то же время в разных ситуациях один и тот же объект может выступать в роли клиента или сервера. Этот тип графика позволяет вам отражать порядок сообщений, передаваемых между объектами, без необходимости фокусироваться на конкретных взаимодействиях, и предпочитает порядок получения / отправки сообщений.

					ДП-02069964-09.02.03-04-20	Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		

На диаграмме последовательности объект в основном представляет экземпляр класса или объекта, который имеет поведение. Этими объектами могут быть пользователь, который инициирует взаимодействие, класс, который играет роль в системе, программные компоненты или даже вся система. Диаграмма последовательности изображает только те объекты, которые непосредственно участвуют во взаимодействии, и не показывает возможные статические ассоциации с другими объектами. Для диаграмм последовательности ключевым моментом является динамическое взаимодействие объектов во времени.

Диаграмма последовательности предоставлена на рисунке 1.

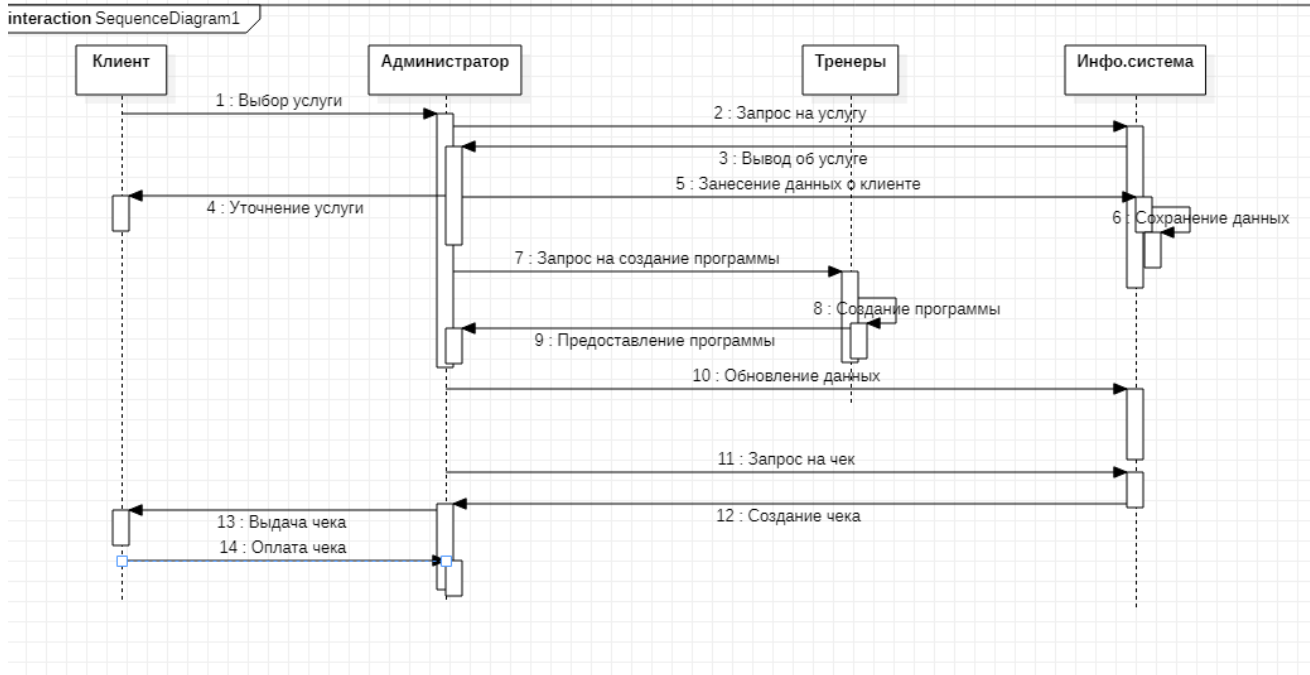


Рисунок 1 — Диаграмма последовательности

4.3 Диаграмма компонентов

Данная диаграмма используется для описания физического представления модели системы. С помощью диаграммы компонентов вы можете определить архитектуру разработанной системы, установив зависимости между программными компонентами, такими как исходный код, двоичный код и исполняемый код. Основными графическими элементами диаграммы компонентов являются компоненты, интерфейсы и их зависимости.

Компонент — физическая и заменяемая часть системы, которая соответствует набору интерфейсов и обеспечивает реализацию этого набора интерфейсов. Диаграмма компонентов предоставлена на рисунке 2.

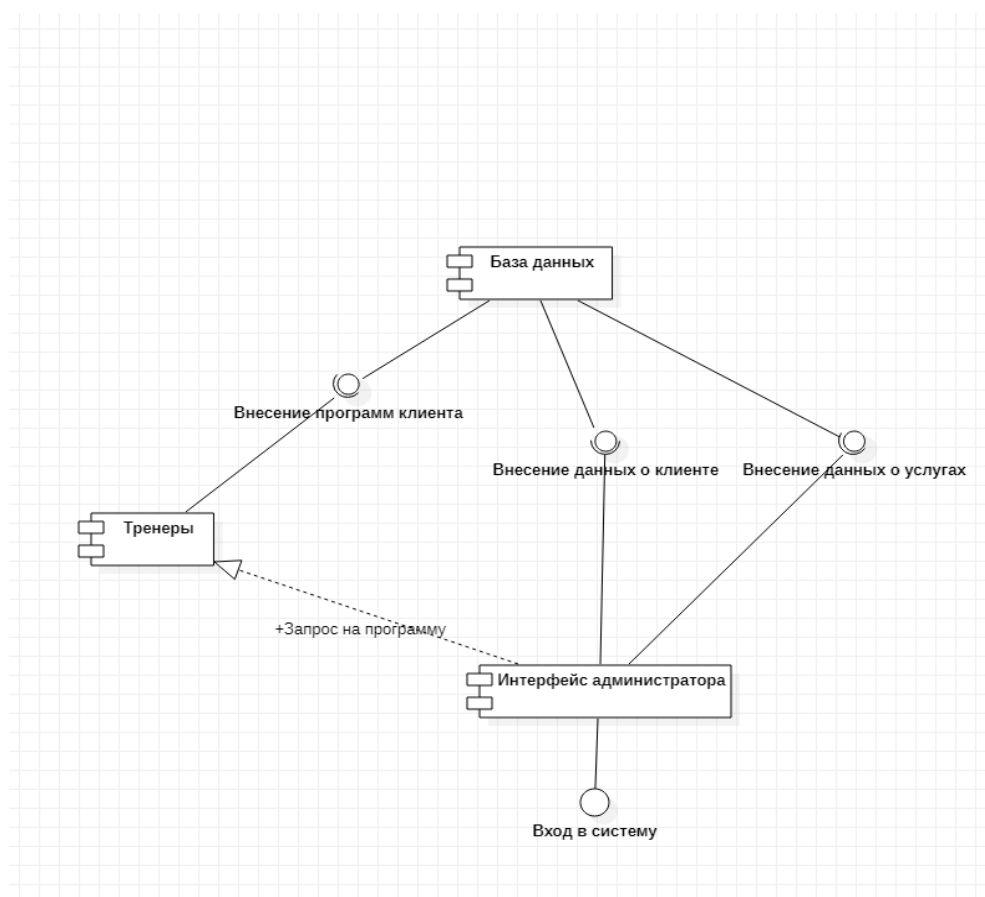


Рисунок 2 — Диаграмма компонентов

4.4 Диаграмма развертывания

Диаграмма развёртывания — один из доступных видов диаграмм.

Корпоративные приложения часто требуют некоторой ИТ-инфраструктуры для запуска, хранения информации в базе данных где-то на сервере компании, вызова веб-служб, использования общих ресурсов и так далее [4]. В этом случае полезно графически представить структуру, в которой будет развернута инфраструктура приложения. Для этого вам нужен сценарий развертывания, иногда называемый сценарием развертывания.

Такие диаграммы есть смысл строить только для аппаратно-программных систем, тогда как UML позволяет строить модели любых систем, не обязательно компьютерных.

Плюсы диаграммы развертывания:

- графическое представление ИТ-инфраструктуры может помочь более рационально распределить компоненты системы по узлам сети, от чего зависит в том числе и производительность системы;
- помощь в решении для множества вспомогательных задач, связанных, например, с обеспечением безопасности.

Диаграмма развертывания показывает топологию системы, распределение компонентов системы между ее узлами и путь передачи информации о соединении между аппаратными узлами. Это единственный граф, который использует «трехмерные» обозначения: узлы системы представлены многомерными наборами данных. Все остальные символы в UML — плоские фигуры. Диаграмма развертывания представлена на рисунке 3

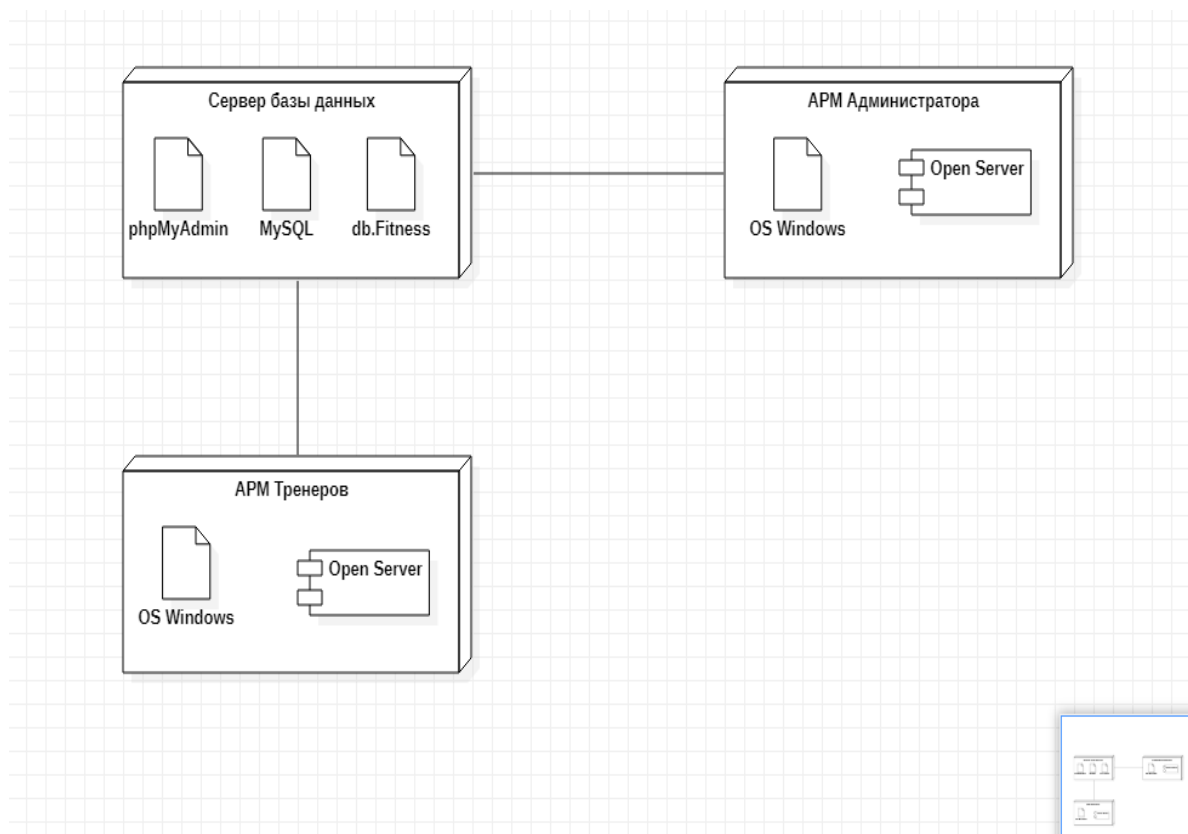


Рисунок 3 — Диаграмма развертывания

4.5 Дизайн автоматизированной информационной системы программы

Перед началом реализации было уделено внимание дизайну программы. Так как программа является информационной системы, то дизайн должен быть строгим, но простым в использовании, чтобы обучающий быстро мог понять принцип работы программы.

Реализация дизайна приложения происходила поэтапно, так как разработка и концепция дизайна приложения определена. Также нельзя в пользу дизайна урезать функциональность приложения.

Всем добавленным объектам в проект программы было дано наименование для удобства работы.

4.6 Создание проекта в Visual Studio

Реализация пользовательского интерфейса началась с создания проекта Windows Forms в Visual Studio 2019. Для создания проекта после запуска программы в правом меню нужно выбрать пункт «создание нового проекта», далее выбрать нужный тип проекта, в нашем случае это приложение Windows Forms. Создание проекта представлено на рисунке 4.

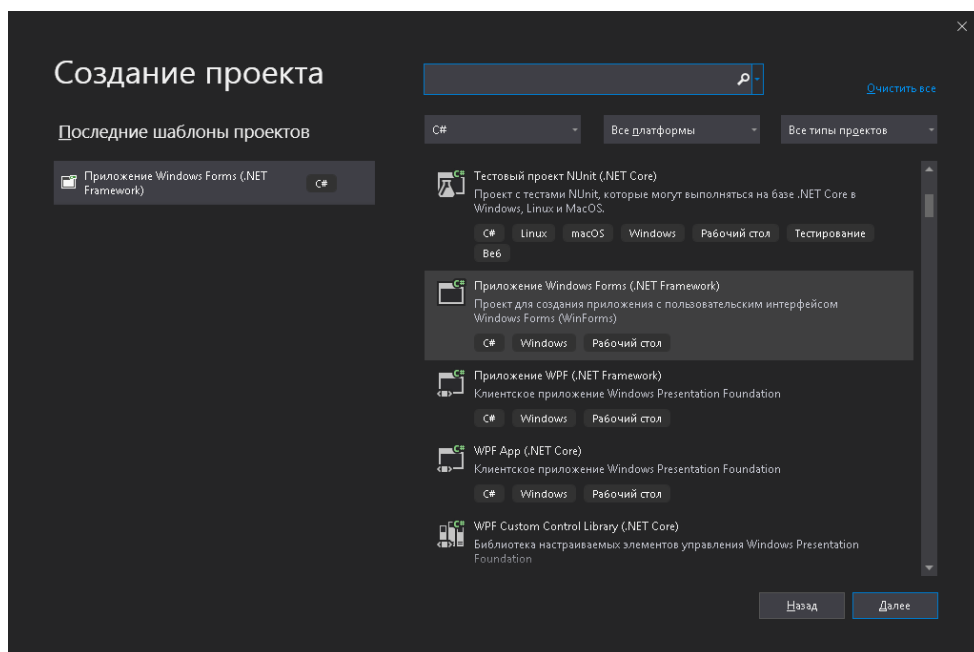


Рисунок 4 — Создание проекта

4.7 Реализация интерфейса программного продукта

Окно входа в программу представлено на Рисунке 5.

Каждый менеджер — это определенный человек. Для входа с ролью «Менеджер» необходимо выбрать свое ФИО из списка сотрудников и ввести пароль. Стандартный пароль для всех вновь созданных пользователей — password. После входа рекомендуется сменить свой пароль в меню программы.

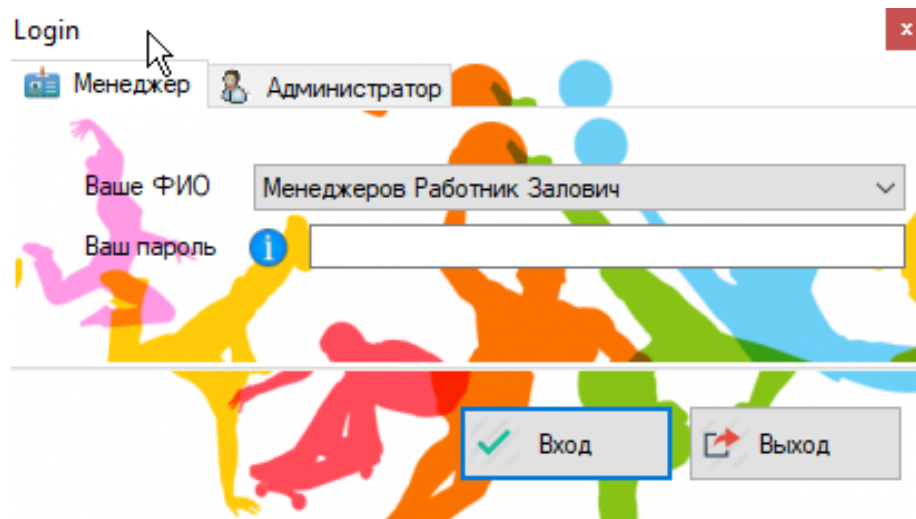


Рисунок 5 — Окно авторизации пользователей

Администратор — абстрактный пользователь без ФИО. Администратором может выступать любой пользователь, имеющий права на управление спортивным клубом. Администратора назначает директор спортивного клуба и выдает пароль. Стандартный пароль для администратора системы admin123.

Листинг страницы «Окно авторизации пользователей» представлен в приложении А.

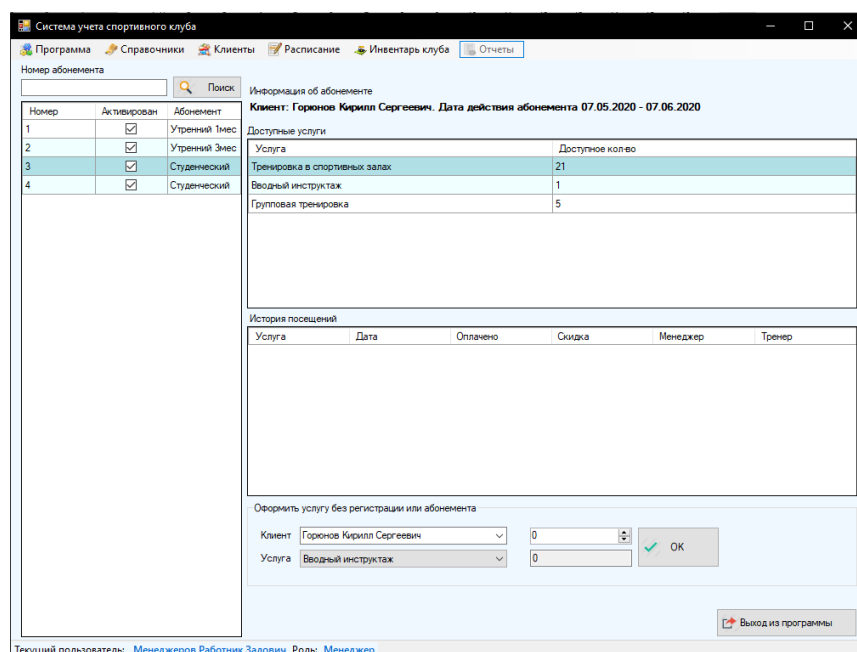


Рисунок 6 — Главное окно программы

Главное окно программы состоит из меню и главной рабочей области приложения для менеджера, где он может по приходу клиента зафиксировать его посещение. Главное окно программы представлено на рисунке 6

Листинг страницы «Главное окно приложения» представлен в приложении Б.

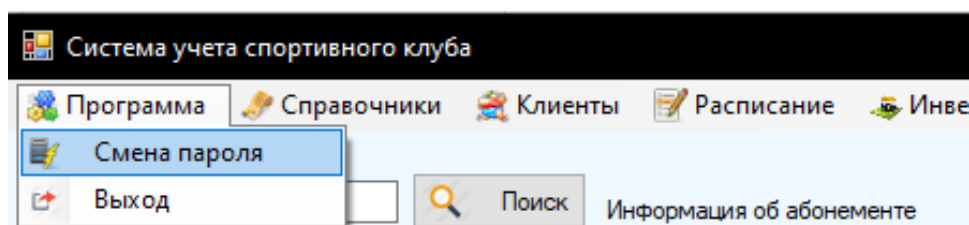


Рисунок 7 — Смена пароля

Каждому работнику рекомендуется сменить стандартный пароль на личный. Администратор системы обязан следить, чтобы каждый пользователь, имеющий доступ к системе имел личный пароль и не разглашал его третьим лицам. Сменить пароль можно после входа в программу в меню. Окно смены пароля представлено на рисунке 7

Меню «Справочники» — «Тренажеры и снаряды»

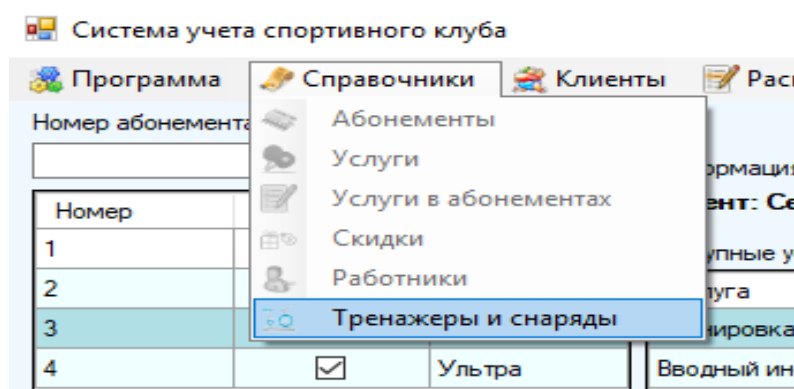


Рисунок 8 — Меню тренажеры

Менеджер не имеет доступ к редактированию справочников касающихся любых вопросов, связанных с финансовой стороной деятельности спортивного

клуба. Единственный доступный справочник для менеджера — «Тренажеры и снаряды». Окно Тренажеры и снаряды представлено на рисунке 9

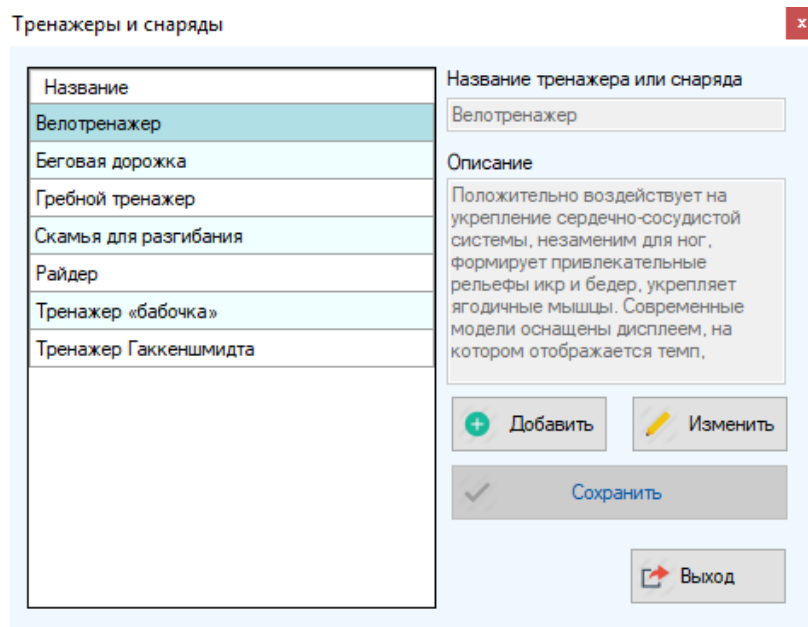


Рисунок 9 — Тренажеры и снаряды

Листинг страницы «Тренажеры и снаряды» представлен в приложении В.

Меню «Клиенты» — «Абонементы»

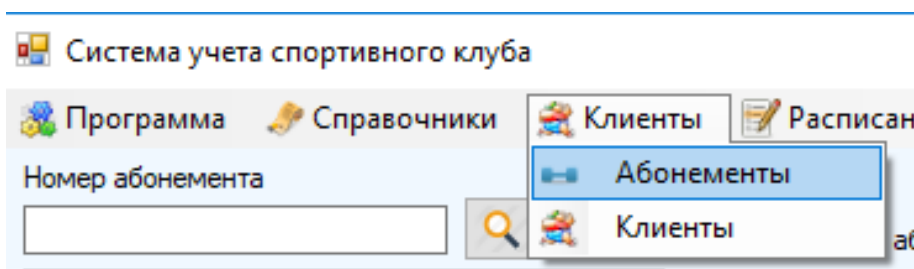


Рисунок 10 — Меню абонементов

Данная опция программы открывает окно работы с абонементом, зарегистрированными в системе. Опция Абонементы представлена на рисунке 10

Абонементы клиентов

Номер абонемента
4

Поиск

Абонемент: Ультра

Клиент: Сергеев Сергей Сергеевич

Статус: Активирован

Период действия:

22 мая 2018 г. 22 июня 2018 г.

Продлить период действия Заморозить

Активировать

Выход

Рисунок 11 — Абонементы клиентов

В этом окне менеджер может заморозить абонемент клиента в случае просьбы клиента, продлить действие абонемента после разморозки абонемента, активировать (в случае если абонемент подарочный или новый). Все новые абонементы имеют статус — «Не активирован». Работа с такими абонементами не возможна. После регистрации абонемента в системе, его необходимо активировать, либо активировать позже (если абонемент был подарочный). Окно Абонементы клиентов представлено на рисунке 11

Листинг страницы «Абонементы клиентов» представлен в приложении Г.

Меню «Клиенты» — «Клиенты»

ФИО клиента	Phone	Дата рождения
Не зарегистрированный клиент	(111) 111-1111	01.01.2000
Клиентов Тест Васильевич	(000) 000-0001	22.01.1990
Сергеев Сергей Сергеевич	(777) 777-7777	24.06.1985

Рисунок 12 — Клиенты

В данном окне происходит регистрация новых клиентов и оформление абонементов. Каждый оформленный абонемент учитывается в статистике менеджера, поэтому важно работать под своей личной учетной записью.

Для каждого конкретного выбранного клиента можно оформить абонемент, нажав на соответствующую кнопку. Окно клиенты представлено на рисунке 12

Листинг страницы «Клиенты» представлен в приложении Д.

Меню «Расписание»

Расписание сотрудников

ФИО	Должность	Понедельник	Вторник	Среда	Четверг	Пятница	Суббота	Воскресение
Менеджеров Ра...	Менеджер	07:00:00 - 19:00:00	07:00:00 - 15:00:00	15:00:00 - 23:00:00		07:00:00 - 13:00:00		10:00:00 - 16:00:00

☒ Понедельник
7:00:00 19:00:00
☒ Вторник
7:00:00 15:00:00
☒ Среда
15:00:00 23:00:00
☐ Четверг
0:00:00 0:00:00
☒ Пятница
7:00:00 13:00:00
☐ Суббота
0:00:00 0:00:00
☒ Воскресенье
10:00:00 16:00:00

Сохранить
Печать расписания

Всего часов за неделю: 40ч. 0 мин.

Выход

Рисунок 13 — Расписание

Менеджеру доступно окно с расписанием сотрудников, но не доступно редактирование информации. Составлять расписание может только администратор. Окно Расписание представлено на рисунке 13

Листинг страницы «Расписание» представлен в приложении Е.

Меню «Инвентарь клуба»

Менеджер имеет возможность составление списка инвентаря клуба, указывать место расположение оборудования, дату обслуживания и др. информацию. Менеджер обязан следить за исправностью оборудования, которое закреплено за ним. Окно Инвентарь клуба представлено на рисунке 14

Листинг страницы «Инвентарь клуба» представлен в приложении Ж.

Спортивный инвентарь клуба

Снаряд	Расположение	Дата посл. обслуж.	Состояние	Инв. номер	Отв. сотрудник
Велотренажер	Фитнес-зал 1	01.04.2018	Исправен	318544	Менеджеров Работни...
Беговая дорожка	Фитнес-зал 1	01.04.2018	Требуется ревизии	315113	Менеджеров Работни...
Райдер	Кроссфит зал 1	27.04.2018	На ремонте	K1P322	Менеджеров Работни...

Отв. сотрудник за состояние снаряда
Менеджеров Работник Залович

Статус снаряда
Исправен

Инв. номер

Поиск

Название тренажера или снаряда
Беговая дорожка

Место установки
Фитнес-зал 1

Отв. сотрудник за состояние снаряда
Менеджеров Работник Залович

Дата последнего обслуживания
1 апреля 2018 г.

Статус снаряда
Требуется ревизии

Инв. номер
315113

Выбрать фото

Добавить Изменить

Сохранить

Выход

Рисунок 14 — Инвентарь клуба

6 Тестирование программного продукта

Прежде всего, надо сказать, что вся поступающая информация подлежит проверке корректности данных, также как и выходные данные в соответствии с функциями, представленными в спецификациях требований. Весь этот процесс называется тестированием программного модуля, где происходит проверка корректности структуры модулей и примененных конструктивных элементов: блоков, переключателей, циклов и т.д.

Проверке могут подлежать также пути обработки информации, которые находятся в каждом модуле, и правильность их реализации в зависимости от изначальных исходных данных. При проверке полноты теста определяются критерии выделения маршрутов для тестирования вместе со степенью покрытия тестами требований спецификаций и возможных путей исполнения программы. Корректность выполняемых вычислений при определенно-фиксированных исходных данных должна проверяться на каждом выделенном пути. Одновременно с этим обнаруживаются ошибки неполного состава или некорректности условий при выполнении частных путей обработки данных. Для всякого выделенного пути по тексту программы (приложения) образовывается набор определенных условий, определяющих его реализацию и используемый при создании соответствующего теста. Подобное представлений путей позволяет упорядоченно-контролировать достигнутый уровень проверки путей и в какой-то степени предохраняет от случайного пропуска отдельных не протестированных маршрутов.

Ручное тестирование — это процесс поиска неисправностей в работе программы, когда тестер проверяет работоспособность всех составляющих программы, как если бы он был пользователем. Нередко, для точности проверки, тестер применяет заранее заготовленную схему тестирования, в котором отмечены наиболее важные моменты работы программы.

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		37

Таблица 1 — Тестирование

Отображение	Наименование	Значение	Результат
Окно авторизации	Логин	ФИО пользователя	Успешная авторизация
	Пароль	password	
	Кнопка подсказки пароля	Клик	Успешное выполнение отображения подсказки
	Кнопка «Вход»	Клик	Выполнение действия на переход на главную страницу
	Кнопка «Выход»	Клик	Успешное завершение работы приложения
	Кнопка «Администратор»	Клик	Переход на окно авторизации Администратора
	Пароль	admin123	Успешная авторизация
Главное окно	Смена пароля	Клик	Переход в окно смены пароля
	Кнопка «Выход»	Клик	Выход из приложения
	Кнопка «Справочники»	Клик	Вывод выпадающего списка справочников
	Кнопка «Клиенты»	Клик	Вывод выпадающего списка
	Кнопка «Расписание»	Клик	Переход в окно Расписание
	Кнопка «Инвентарь»	Клик	Переход в окно Инвентарь
	Кнопка «Отчеты»	Клик	Вывод выпадающего списка Отчетов
	Кнопка «Выход из программы»	Клик	Успешное завершение работы приложения

Окончание таблицы 1

Тренажеры и снаряды	Кнопка «Добавить»	Клик	Успешное добавление новой записи
	Кнопка «Изменить»	Клик	Успешное изменение записи
	Кнопка «Сохранить»	Клик	Успешное сохранение записей
	Кнопка «Выход»	Клик	Успешный выход в главное окно приложения
Абонементы	Кнопка «Поиск»	Клик	Успешный поиск абонемента по его номеру
	Кнопка «Продлить абонемент»	Клик	Успешное продление абонемента
	Кнопка «Заморозить»	Клик	Успешное заморозка абонемента
	Кнопка «Активировать»	Клик	Успешное активация абонемента
	Кнопка «Выход»	Клик	Успешный выход в главное окно приложения
Клиенты	Кнопка «Добавить»	Клик	Успешное добавление новой записи
	Кнопка «Изменить»	Клик	Успешное изменение записи
	Кнопка «Сохранить»	Клик	Успешное сохранение записей
	Кнопка «Выход»	Клик	Успешный выход в главное окно приложения
	Кнопка «Поиск»	Клик	Успешный поиск клиента по ФИО и(или) его телефона

ЗАКЛЮЧЕНИЕ

Во время дипломной работы был разработан и реализован автоматизированное рабочее место администратора фитнес клуба. сформулированы требования к разрабатываемой системе, проведен анализ предметной области и, разработаны даталогическая схема базы данных, реализован интерфейс и логическая составляющая автоматизированного рабочего места администратора, разработан и реализован работающий программный продукт. Получен опыт практического использования современных объектно-ориентированных и типобезопасных языков программирования среды разработки программного продукта.

В результате выполнения дипломного проекта составлено техническое задание, описана предметная область, в соответствии с техническими требованиями, закреплены навыки по работе со средой C# и созданием автоматизированной информационной системы, проведено тестирование приложения.

В данном дипломном проекте разработано и реализовано автоматизированное рабочее места администратора фитнес-клуба. Назначение разработки заключается в следующем: облегчение деятельности сотрудников фитнес-клуба, а именно автоматизирование работы с данными клиента.

В итоге, достигнута цель и реализованы задачи, обозначенные в начале написания дипломного проекта.

- составить анализ предметной области;
- составить технического задание;
- спроектировать и разработать интерфейс;
- реализовать автоматизированную информационную систему:
 - а) создать даталогическую модель данных;
 - б) реализовать базу данных;
 - в) создать формы для интерфейса;

г) разработать дизайн информационной системы;
д) реализовать добавление, редактирование, удаление и сохранение данных;

е) добавить в приложение возможность отчетности.

– протестировать разработанное приложение.

В приложении выполняются все необходимые требования.

Главной перспективой развития данного приложения будет являться реализации в фитнес-клубе. Для этого дизайнерская составляющая будет улучшена. Также будут добавлены новые функции за счет доработки кода и внесения новых объектов в приложение.

					ДП-02069964-09.02.03-04-20	Лист
						41
Изм.	Лист	№ докум.	Подпись	Дата		

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 ГОСТ 34.602-89. Единая система программной документации. Техническое задание. Требования к содержанию и оформлению. – введ. 01–01–2018 – Москва : Государственный комитет СССР по стандартам от 18 декабря 1978 г. № 3351 ; М. : Изд-во стандартов, сор. 1978. – 59 с.

2 Федотова Е. Л. Информационные технологии в профессиональной деятельности: учеб. пособие / Е. Л. Федотова. — М. : ИД «ФОРУМ» : ИНФРА-М, 2018. – 367 с. – (Среднее профессиональное образование). – Режим доступа: <http://znanium.com/catalog/product/944899>.

3 Хлебников А. А. Информационные технологии: Учебник / А. А. Хлебников. – М.: КноРус, 2014. – 472 с. – Режим доступа: <https://static.my-shop.ru/product/pdf/148/1476911.pdf>.

4 Малышева Е. Н. Проектирование информационных систем. Объектно-ориентированная Case-технология проектирования информационных систем / Е. Н. Малышева – Кемерово: Кемеровский государственный институт культуры, 2009. – 70 с.

5 Гульятеев А. К. Проектирование и дизайн пользовательского интерфейса [Текст] : учебное пособие / А. К. Гульятеев, В. А. Машин. – СПб. : Корона-принт, 2000. – 349 с. : ил. – (Учитель и ученик). – ISBN 5-7931-0092-X : Б. ц. – Режим доступа: <https://spbib.ru/catalog/-/books/3146559-proektirovanie-i-dizajn-pol-zovatel-skogo-interfejsa>. – Загл. с экрана.

ПРИЛОЖЕНИЕ А

(обязательное)

Листинг страницы «Окно авторизации пользователей»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
namespace SportClub_Client
{
    public class LoginForm : Form
    {
        public bool isOK { get; set; }
        public object People { get; set; }
        public LoginForm()
        {
            this.InitializeComponent();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            switch (Role.role)
            {
                case RoleEnum.Администратор:
                    if (DataAccess.CheckPassword(Role.role, (object) null,
this.adminpass_txbx.Text))
                    {
                        this.People = new object();
                        this.isOK = true;
                        this.Close();
                        break;
                    }
                    int num1 = (int) MessageBox.Show("Пароль не верный!", "Ошибка",
MessageBoxButtons.OK, MessageBoxIcon.Hand);
                    break;
                case RoleEnum.Менеджер:
                    if (this.manager_cmbx.SelectedItem != null)
                    {
                        if (DataAccess.CheckPassword(Role.role,
this.manager_cmbx.SelectedItem, this.manager_pass_txbx.Text))
                        {
                            this.People = this.manager_cmbx.SelectedItem;
                        }
                    }
            }
        }
    }
}
```

Окончание ПРИЛОЖЕНИЯ А

```

        this.isOK = true;
        DataAccess.employee = this.manager_cmbx.SelectedItem as Employee;
        this.Close();
        break;
    }
    int num2 = (int) MessageBox.Show("Пароль не верный!", "Ошибка",
    MessageBoxButtons.OK, MessageBoxIcon.Hand);
    break;
}
    int num3 = (int) MessageBox.Show("Необходимо выбрать свое имя!",
    "Ошибка", MessageBoxButtons.OK, MessageBoxIcon.Hand);
    break;
}
}
private void LoginForm_Load(object sender, EventArgs e)
{
    for (int index = 0; index < this.tabControl1.TabPages.Count; ++index)
        this.tabControl1.TabPages[index].ImageIndex = index;
    this.manager_cmbx.DataSource = (object) DataAccess.GetMenagers();
    Role.role = RoleEnum.Менеджер;
}
private void pictureBox1_Click(object sender, EventArgs e)
{
    int num = (int) MessageBox.Show("Стандартный пароль системы - password.
    После входа в программу рекомендуется сменить пароль. \r\n
    Если ваш
    пароль не позволяет войти в программу, обратитесь к администратору системы",
    "Информация о паролях", MessageBoxButtons.OK, MessageBoxIcon.Asterisk);
}
private void exit_btn_Click(object sender, EventArgs e)
{
    this.Close();
}
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    string text = this.tabControl1.SelectedTab.Text;
    if (!(text == "Менеджер"))
    {
        if (!(text == "Администратор"))
            return;
        Role.role = RoleEnum.Администратор;
    }
    else
        Role.role = RoleEnum.Менеджер;
}
protected override void Dispose(bool disposing)
{
    if (disposing && this.components != null)
        this.components.Dispose();
    base.Dispose(disposing);
}
}

```

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг страницы «Главное окно приложения»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace SportClub_Client.Views
{
    public class MainForm : Form
    {

        public MainForm(object people)
        {
            this.InitializeComponent();
            this.People = people;
        }

        public object People { get; set; }

        private void MainForm_Load(object sender, EventArgs e)
        {
            switch (Role.role)
            {
                case RoleEnum.Администратор:
                    this.people_status.Text = "Администратор системы";
                    this.role_status.Text = Role.role.ToString();
                    this.использоватьУслугуToolStripMenuItem.Enabled = false;
            }
        }
    }
}
```

					ДП-02069964-09.02.03-04-20	Лист
						45
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ Б

```

        this.people_status.ForeColor = this.role_status.ForeColor =
Color.Red;
        this.ok_btn.Enabled = false;
        break;
    case RoleEnum.Менеджер:
        this.people_status.Text = (this.People as Employee).Name;
        this.role_status.Text = Role.role.ToString();
        this.отчетыToolStripMenuItem.Enabled = false;
        this.работникиToolStripMenuItem.Enabled = false;
        this.скидкиToolStripMenuItem.Enabled = false;
        this.абонементыToolStripMenuItem.Enabled =
this.услугиToolStripMenuItem.Enabled =
this.услугиВАбонементахToolStripMenuItem.Enabled = false;
        this.скидкиToolStripMenuItem.Enabled = false;
        break;
    }
    this.abon_dgv.DataSource = (object) DataAccess.GetAbonementsClients();
    this.clients_cmbx.DataSource = (object) DataAccess.GetClients();
    this.clients_cmbx.AutoCompleteCustomSource = new
AutoCompleteStringCollection();
    foreach (Client client in DataAccess.GetClients())
        this.clients_cmbx.AutoCompleteCustomSource.Add(client.Name);
    this.clients_cmbx.AutoCompleteMode = AutoCompleteMode.SuggestAppend;
    this.clients_cmbx.AutoCompleteSource = AutoCompleteSource.CustomSource;
    this.service_cmbx.DataSource = DataAccess.GetServices();
}

private void выходToolStripMenuItem_Click(object sender, EventArgs e)
{
    this.Close();
}

private void абонементыToolStripMenuItem_Click(object sender, EventArgs e)
{

```

Продолжение ПРИЛОЖЕНИЯ Б

```
        int num = (int) new AbonementForm().ShowDialog();
        this.abon_dgv.DataSource = (object) DataAccess.GetAbonementsClients();
    }
    private void услугиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        int num = (int) new ServiceForm().ShowDialog();
    }

    private void клиентыToolStripMenuItem_Click(object sender, EventArgs e)
    {
        int num = (int) new ClientsForm().ShowDialog();
        this.abon_dgv.DataSource = (object) DataAccess.GetAbonementsClients();
    }

    private void работникиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        int num = (int) new EmployeesForm().ShowDialog();
    }

    private void exit_btn_Click(object sender, EventArgs e)
    {
        this.Close();
    }

    private void скидкиToolStripMenuItem_Click(object sender, EventArgs e)
    {
        int num = (int) new DiscountRatesForm().ShowDialog();
        this.abon_dgv.DataSource = (object) DataAccess.GetAbonementsClients();
    }

    private void услугиВАбонементaxToolStripMenuItem_Click(object sender,
    EventArgs e)
    {
        int num = (int) new ServicesInAbonemtnsForm().ShowDialog();
    }
```

Продолжение ПРИЛОЖЕНИЯ Б

```
this.abon_dgv.DataSource = (object)
DataAccess.GetAbonementsClients().Where<AbonemetsClient>((Func<AbonemetsClient,
bool>) (x => x.Status)).ToList<AbonemetsClient>();
}
private void search_btn_Click(object sender, EventArgs e)
{
    this.abon_dgv.DataSource = (object)
DataAccess.GetAbonementsClients(this.textBox1.Text);
}

private void abon_dgv_DataBindingComplete(object sender,
DataGridViewBindingCompleteEventArgs e)
{
    if (this.abon_dgv.Columns["ID_AbonemetClient"] != null)
        this.abon_dgv.Columns["ID_AbonemetClient"].HeaderText = "Номер";
    if (this.abon_dgv.Columns["Abonement"] != null)
        this.abon_dgv.Columns["Abonement"].HeaderText = "Абонемент";
    if (this.abon_dgv.Columns["Client"] != null)
        this.abon_dgv.Columns["Client"].Visible = false;
    if (this.abon_dgv.Columns["Employee"] != null)
        this.abon_dgv.Columns["Employee"].Visible = false;
    if (this.abon_dgv.Columns["DateStart"] != null)
        this.abon_dgv.Columns["DateStart"].Visible = false;
    if (this.abon_dgv.Columns["DateEnd"] != null)
        this.abon_dgv.Columns["DateEnd"].Visible = false;
    if (this.abon_dgv.Columns["Employee_ID"] != null)
        this.abon_dgv.Columns["Employee_ID"].Visible = false;
    if (this.abon_dgv.Columns["JournalServices"] != null)
        this.abon_dgv.Columns["JournalServices"].Visible = false;
    if (this.abon_dgv.Columns["Status"] == null)
        return;
    this.abon_dgv.Columns["Status"].HeaderText = "Активирован";
}

private void abon_dgv_SelectionChanged(object sender, EventArgs e)
```


Продолжение ПРИЛОЖЕНИЯ Б

```

        if (this.abon_dgv.SelectedRows.Count > 0)
        {
            AbonemetsClient dataBoundItem =
this.abon_dgv.SelectedRows[0].DataBoundItem as AbonemetsClient;
            Label label4 = this.label4;
            string name = dataBoundItem.Client.Name;
            DateTime dateTime = dataBoundItem.DateStart;
            dateTime = dateTime.Date;
            string shortDateString1 = dateTime.ToShortDateString();
            dateTime = dataBoundItem.DateEnd;
            dateTime = dateTime.Date;
            string shortDateString2 = dateTime.ToShortDateString();
            string str = string.Format("Клиент: {0}. Дата действия абонемента {1} -
{2}", (object) name, (object) shortDateString1, (object) shortDateString2);
            label4.Text = str;
            DateTime dateEnd = dataBoundItem.DateEnd;
            dateTime = DateTime.Now;
            DateTime date = dateTime.Date;
            if (dateEnd < date)
            {
                this.label4.Text += " ПРОСПРОЧЕН!";
                this.label4.ForeColor = Color.IndianRed;
            }
            else
                this.label4.ForeColor = Color.Black;
            if (!dataBoundItem.Status)
            {
                this.label4.Text += " Заморожен!";
                this.label4.ForeColor = Color.IndianRed;
            }
            this.count_dgv.DataSource = (object)
DataAccess.GetAviableService(dataBoundItem);
            this.history_dgv.DataSource = (object)
DataAccess.GetJournal(dataBoundItem);
        }
    }
    else

```

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		49

Окончание ПРИЛОЖЕНИЯ Б

```

        this.label4.Text = "";
        this.count_dgv.DataSource = (object) null;
        this.history_dgv.DataSource = (object) null;
    }
}
private void count_dgv_DataBindingComplete(
    object sender,
    DataGridViewBindingCompleteEventArgs e)
{
    if (this.count_dgv.Columns["service"] != null)
        this.count_dgv.Columns["service"].HeaderText = "Услуга";
    if (this.count_dgv.Columns["count"] == null)
        return;
    this.count_dgv.Columns["count"].HeaderText = "Доступное кол-во";
}
private void history_dgv_DataBindingComplete(
    object sender,
    DataGridViewBindingCompleteEventArgs e)
{
    if (this.history_dgv.Columns["DateUse"] != null)
        this.history_dgv.Columns["DateUse"].HeaderText = "Дата";
    if (this.history_dgv.Columns["Cost"] != null)
        this.history_dgv.Columns["Cost"].HeaderText = "Оплачено";
    if (this.history_dgv.Columns["AbonemetsClient"] != null)
        this.history_dgv.Columns["AbonemetsClient"].Visible = false;
    if (this.history_dgv.Columns["ID_JournalService"] != null)
        this.history_dgv.Columns["ID_JournalService"].Visible = false;
    if (this.history_dgv.Columns["Employee"] != null)
        this.history_dgv.Columns["Employee"].HeaderText = "Менеджер";
    if (this.history_dgv.Columns["Employee1"] != null)
        this.history_dgv.Columns["Employee1"].HeaderText = "Тренер";
    if (this.history_dgv.Columns["Service"] != null)
    {
        this.history_dgv.Columns["Service"].HeaderText = "Услуга";
        this.history_dgv.Columns["Service"].DisplayIndex = 1;
    }
}

```

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		50

ПРИЛОЖЕНИЕ В

(обязательное)

Листинг страницы «Тренажеры и снаряды»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;
namespace SportClub_Client.Views
{
    public class ApparatusForm : Form
    {
        public ApparatusForm()
        {
            this.InitializeComponent();
        }
        public bool isEdit { get; set; }

        private void add_btn_Click(object sender, EventArgs e)
        {
            this.EnableControls();
            this.txbx.Text = string.Empty;
            this.info_txbx.Text = string.Empty;
            this.isEdit = false;
            this.edit_btn.Enabled = false;
        }
        private void edit_btn_Click(object sender, EventArgs e)
        {
            this.EnableControls();
            this.isEdit = true;
            this.add_btn.Enabled = false;
        }
        private void DisableControls()
        {
            this.save_btn.Enabled = this.txbx.Enabled = this.info_txbx.Enabled =
false;
            this.add_btn.Enabled = this.edit_btn.Enabled = true;
        }

        private void EnableControls()
        {
            this.save_btn.Enabled = this.txbx.Enabled = this.info_txbx.Enabled = true

```

Окончание ПРИЛОЖЕНИЯ В

```

        Apparatus dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem as
Apparatus;
        dataBoundItem.Title = this.txbx.Text;
        dataBoundItem.Info = this.info_txbx.Text;
        DataAccess.UpdateApparatus(dataBoundItem);
    }
}
else
    DataAccess.InsertApparatus(new Apparatus()
    {
        Title = this.txbx.Text,
        Info = this.info_txbx.Text
    });
    this.DisableControls();
    this.dgv.DataSource = (object) DataAccess.GetApparatusList();
}
private void dgv_DataBindingComplete(object sender,
DataGridViewBindingCompleteEventArgs e)
{
    if (this.dgv.Columns["Title"] != null)
        this.dgv.Columns["Title"].HeaderText = "Название";
    if (this.dgv.Columns["ID_Apparatus"] != null)
        this.dgv.Columns["ID_Apparatus"].Visible = false;
    if (this.dgv.Columns["GumApparatus"] != null)
        this.dgv.Columns["GumApparatus"].Visible = false;
    if (this.dgv.Columns["Info"] == null)
        return;
    this.dgv.Columns["Info"].Visible = false;
}
private void dgv_SelectionChanged(object sender, EventArgs e)
{
    if (this.dgv.SelectedRows.Count <= 0)
        return;
    Apparatus dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem as
Apparatus;
    this.txbx.Text = dataBoundItem.Title;
    this.info_txbx.Text = dataBoundItem.Info;
    this.DisableControls();
    this.isEdit = false;
}
private void Form_Load(object sender, EventArgs e)
{
    this.dgv.DataSource = (object) DataAccess.GetApparatusList();
}
private void exit_btn_Click(object sender, EventArgs e)
{

```

ПРИЛОЖЕНИЕ Г

(обязательное)

Листинг страницы «Абонементы клиентов»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Drawing;
using System.Linq;
using System.Windows.Forms;
namespace SportClub_Client.Views
{
    public class AbonementsForm : Form
    {
        private IContainer components = (IContainer) null;
        private AbonemetsClient abon;
        private Button search_btn;
        private Label label3;
        private TextBox textBox1;
        private Label label1;
        private DateTimePicker dateTimePicker1;
        private DateTimePicker dateTimePicker2;
        private Button button1;
        private Button button2;
        private Button button3;
        private Button button4;
        private Label name_lbl;
        private Label label2;
        private Label label4;
        private Label abon_lbl;
        private Label status_lbl;
        private Label label6;
        public AbonementsForm()
        {
            this.InitializeComponent();
            this.name_lbl.Text = this.abon_lbl.Text = this.status_lbl.Text =
string.Empty;
        }
        private void search_btn_Click(object sender, EventArgs e)
        {
            this.SearchAbon();
        }
        private void button2_Click(object sender, EventArgs e)
        {
            if (this.abon == null)
                return;
            DataAccess.ChangeStatusAbonement(this.abon);
        }
    }
}
```

Окончание ПРИЛОЖЕНИЯ Г

```

        this.SearchAbon();
    }

    private void button4_Click(object sender, EventArgs e)
    {
        if (this.abon == null)
            return;
        this.abon.Status = true;
        DataAccess.ChangeStatusAbonement(this.abon);
        this.SearchAbon();
    }

    private void SearchAbon()
    {
        if (!int.TryParse(this.textBox1.Text, out int _))
            return;
        this.abon =
        DataAccess.GetAbonementsClients(this.textBox1.Text).FirstOrDefault<AbonemetsCli
        ent>();
        if (this.abon != null)
        {
            this.name_lbl.Text = this.abon.Client.Name;
            this.abon_lbl.Text = this.abon.Abonement.Title;
            this.dateTimePicker1.Value = this.abon.DateStart;
            this.dateTimePicker2.Value = this.abon.DateEnd;
            this.status_lbl.Text = this.abon.Status ? "Активирован" : "Заморожен";
        }
        else
        {
            int num = (int) MessageBox.Show("Такой абонемент не найден!", "Не
            найден", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
            this.name_lbl.Text = this.abon_lbl.Text = this.status_lbl.Text =
            string.Empty;
            DateTimePicker dateTimePicker1 = this.dateTimePicker1;
            DateTimePicker dateTimePicker2 = this.dateTimePicker2;
            DateTime now = DateTime.Now;
            DateTime date;
            DateTime dateTime1 = date = now.Date;
            dateTimePicker2.Value = date;
            DateTime dateTime2 = dateTime1;
            dateTimePicker1.Value = dateTime2;
        }
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (this.abon == null)
            return;
        this.abon.DateEnd = this.dateTimePicker2.Value.Date;
        DataAccess.ChangeStatusAbonement(this.abon);
        int num = (int) MessageBox.Show("Срок действия продлен", "Продлен",
        MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
        this.SearchAbon();
    }

```

ПРИЛОЖЕНИЕ Д

(обязательное)

Листинг страницы «Клиенты»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Drawing;
using System.Windows.Forms;

namespace SportClub_Client.Views
{
    public class ClientsForm : Form
    {
        private IContainer components = (IContainer) null;
        private Label label1;
        private Label label2;
        private Button save_btn;
        private Button exit_btn;
        private Button edit_btn;
        private Button add_btn;
        private TextBox txbx;
        private DataGridView dgv;
        private MaskedTextBox phone_txbx;
        private TextBox textBox1;
        private Label label3;
        private Label label4;
        private MaskedTextBox maskedTextBox1;
        private Button search_btn;
        private DateTimePicker birth_dtpc;
        private Label label5;
        private ContextMenuStrip contextMenuStrip1;
        private ToolStripMenuItem добавитьToolStripMenuItem;
        private ToolStripMenuItem изменитьToolStripMenuItem;
        private ToolStripMenuItem удалитьToolStripMenuItem;
        private ToolStripSeparator toolStripSeparator1;
        private ToolStripMenuItem обновитьДанныкToolStripMenuItem;
        private Button button1;
        public ClientsForm()
        {
            this.InitializeComponent();
        }
        public bool isEdit { get; set; }
        private void add_btn_Click(object sender, EventArgs e)
        {
            this.EnableControls();
            this.txbx.Text = string.Empty;
        }
    }
}
```

Продолжение ПРИЛОЖЕНИЯ Д

```

        this.phone_txbx.Text = string.Empty;
        this.birth_dtpc.Value = DateTime.Now;
        this.isEdit = false;
        this.edit_btn.Enabled = false;
    }

    private void edit_btn_Click(object sender, EventArgs e)
    {
        this.EnableControls();
        this.isEdit = true;
        this.add_btn.Enabled = false;
    }

    private void DisableControls()
    {
        this.save_btn.Enabled = this.txbx.Enabled = this.phone_txbx.Enabled =
this.birth_dtpc.Enabled = false;
        this.add_btn.Enabled = this.edit_btn.Enabled = true;
    }

    private void EnableControls()
    {
        this.save_btn.Enabled = this.txbx.Enabled = this.phone_txbx.Enabled =
this.birth_dtpc.Enabled = true;
    }

    private void обновитьДанныеToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        this.dgv.DataSource = (object) null;
        this.dgv.DataSource = (object) DataAccess.GetClients();
    }

    private void save_btn_Click(object sender, EventArgs e)
    {
        if (this.isEdit)
        {
            if (this.dgv.SelectedRows.Count > 0)
            {
                Client dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem as
Client;
                dataBoundItem.Name = this.txbx.Text;
                dataBoundItem.Phone = this.phone_txbx.Text;
                dataBoundItem.BirthDate = this.birth_dtpc.Value;
                DataAccess.UpdateClient(dataBoundItem);
            }
        }
        else
        {
            DataAccess.InsertClient(new Client()
            {
                Name = this.txbx.Text,
                Phone = this.phone_txbx.Text,
                BirthDate = this.birth_dtpc.Value
            });
        }
    }

```


Продолжение ПРИЛОЖЕНИЯ Д

```

        this.DisableControls();
        this.dgv.DataSource = (object) DataAccess.GetClients();
    }

    private void dgv_DataBindingComplete(object sender,
DataGridViewBindingCompleteEventArgs e)
    {
        if (this.dgv.Columns["Name"] != null)
            this.dgv.Columns["Name"].HeaderText = "ФИО клиента";
        if (this.dgv.Columns["ID_Client"] != null)
            this.dgv.Columns["ID_Client"].Visible = false;
        if (this.dgv.Columns["BirthDate"] != null)
            this.dgv.Columns["BirthDate"].HeaderText = "Дата рождения";
        if (this.dgv.Columns["ServicesAbonement"] != null)
            this.dgv.Columns["ServicesAbonement"].Visible = false;
        if (this.dgv.Columns["AbonemetsClient"] == null)
            return;
        this.dgv.Columns["AbonemetsClient"].Visible = false;
    }
    private void dgv_SelectionChanged(object sender, EventArgs e)
    {
        if (this.dgv.SelectedRows.Count > 0)
        {
            Client dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem as
Client;
            this.txbx.Text = dataBoundItem.Name;
            this.phone_txbx.Text = dataBoundItem.Phone;
            this.birth_dtpc.Value = dataBoundItem.BirthDate;
            this.DisableControls();
            this.isEdit = false;
        }

        else
        {
            this.txbx.Text = this.phone_txbx.Text = "";
            this.birth_dtpc.Value = DateTime.Now;
        }
    }
    private void Form_Load(object sender, EventArgs e)
    {
        this.dgv.DataSource = (object) DataAccess.GetClients();
        if (DataAccess.employee != null)
            return;
        this.button1.Enabled = false;
    }
    private void exit_btn_Click(object sender, EventArgs e)
    {
        this.Close();
    }
    private void удалитьToolStripMenuItem_Click_1(object sender, EventArgs e)
    {

```

Окончание ПРИЛОЖЕНИЯ Д

```

        if (this.dgv.SelectedRows.Count <= 0)
            return;
        DataAccess.DeleteClient(this.dgv.SelectedRows[0].DataBoundItem as
Client);
        this.dgv.DataSource = (object) DataAccess.GetClients();
    }

    private void обновитьДанныкToolStripMenuItem_Click(object sender, EventArgs
e)
    {
        this.dgv.DataSource = (object) null;
        this.dgv.DataSource = (object) DataAccess.GetClients();
    }

    private void search_btn_Click(object sender, EventArgs e)
    {
        if (this.maskedTextBox1.MaskCompleted)
            this.dgv.DataSource = (object)
DataAccess.GetClients(this.textBox1.Text, this.maskedTextBox1.Text);
        else
            this.dgv.DataSource = (object)
DataAccess.GetClients(this.textBox1.Text, "");
    }

    private void button1_Click(object sender, EventArgs e)
    {
        if (this.dgv.SelectedRows.Count <= 0)
            return;
        int num = (int) new
AddAbonementForm(this.dgv.SelectedRows[0].DataBoundItem as
Client).ShowDialog();
    }

    protected override void Dispose(bool disposing)
    {
        if (disposing && this.components != null)
            this.components.Dispose();
        base.Dispose(disposing);
    }

```

ПРИЛОЖЕНИЕ Е

(обязательное)

Листинг страницы «Расписание»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Windows.Forms;

namespace SportClub_Client.Views
{
    public class ScheduleForm : Form
    {
        private IContainer components = (IContainer) null;
        private TableLayoutPanel tableLayoutPanel1;
        private Button button3;
        private DataGridView schedule_dgv;
        private Panel panel1;
        private CheckBox monday_check;
        private CheckBox tuesday_check;
        private CheckBox wednesday_check;
        private CheckBox thursday_check;
        private CheckBox friday_check;
        private CheckBox saturday_check;
        private CheckBox sunday_check;
        private Button button2;
        private DateTimePicker sunEnd;
        private DateTimePicker sunStart;
        private DateTimePicker satEnd;
        private DateTimePicker satStart;
        private DateTimePicker friEnd;
        private DateTimePicker friStart;
        private DateTimePicker thuEnd;
        private DateTimePicker thuStart;
        private Label lbl;
        private DateTimePicker wenEnd;
        private Label label6;
        private DateTimePicker wenStart;
        private Label label5;
        private DateTimePicker tueEnd;
        private Label label4;
        private DateTimePicker tueStart;
        private Label label3;
        private DateTimePicker monEnd;
        private Label label2;
        private DateTimePicker monStart;
```

Продолжение ПРИЛОЖЕНИЯ Е

```

private Button button1;

public ScheduleForm()
{
    this.InitializeComponent();
}
private void ScheduleForm_Load(object sender, EventArgs e)
{
    this.schedule_dgv.DataSource = (object) ScheduleBuilder.Scedule;
}

private void FillDateTimePickers()
{
    if (this.schedule_dgv.SelectedRows.Count <= 0)
        return;
    DataRow row = (this.schedule_dgv.SelectedRows[0].DataBoundItem as
DataRowView).Row;
    for (int index = 2; index < row.Table.Columns.Count; ++index)
        this.SetTime(row.Table.Columns[index].ColumnName, row[index] as
ScheduleTime);
}
private void schedule_dgv_SelectionChanged(object sender, EventArgs e)
{
    this.FillDateTimePickers();
    if (this.schedule_dgv.SelectedRows.Count <= 0)
        return;
    DataRow row = (this.schedule_dgv.SelectedRows[0].DataBoundItem as
DataRowView).Row;
    TimeSpan timeSpan = new TimeSpan();
    for (int index = 2; index < row.Table.Columns.Count; ++index)
    {
        if (row[index] is ScheduleTime scheduleTime)
            timeSpan = timeSpan.Add(scheduleTime.end.Value -
scheduleTime.start.Value);
    }
    this.hours_lbl.Text = ((int) timeSpan.TotalHours).ToString() + "ч. " +
timeSpan.Minutes.ToString() + " мин.";
}
private void SetTime(string DayOfWeek, ScheduleTime time)
{
    string s = DayOfWeek;
    // ISSUE: reference to a compiler-generated method
    switch (\u003CPrivateImplementationDetails\u003E.ComputeStringHash(s))
    {
        case 449991711:
            if (!(s == "Среда"))
                break;
            if (time != null)
            {
                this.wednesday_check.Checked = true;
                this.wenStart.Value = DateTime.Parse(time.start.ToString());
            }
        }
    }

```

Продолжение ПРИЛОЖЕНИЯ Е

```

        this.wenEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.wednesday_check.Checked = false;
    this.wenStart.Value = DateTime.Parse("00:00:00");
    this.wenEnd.Value = DateTime.Parse("00:00:00");
    break;
case 695876427:
    if (!(s == "Понедельник"))
        break;
    if (time != null)
    {
        this.monday_check.Checked = true;
        this.monStart.Value = DateTime.Parse(time.start.ToString());
        this.monEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.monday_check.Checked = false;
    this.monStart.Value = DateTime.Parse("00:00:00");
    this.monEnd.Value = DateTime.Parse("00:00:00");
    break;
case 795394900:
    if (!(s == "Вторник"))
        break;
    if (time != null)
    {
        this.tuesday_check.Checked = true;
        this.tueStart.Value = DateTime.Parse(time.start.ToString());
        this.tueEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.tuesday_check.Checked = false;
    this.tueStart.Value = DateTime.Parse("00:00:00");
    this.tueEnd.Value = DateTime.Parse("00:00:00");
    break;
case 1145421657:
    if (!(s == "Воскресение"))
        break;
    if (time != null)
    {
        this.sunday_check.Checked = true;
        this.sunStart.Value = DateTime.Parse(time.start.ToString());
        this.sunEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.sunday_check.Checked = false;
    this.sunStart.Value = DateTime.Parse("00:00:00");
    this.sunEnd.Value = DateTime.Parse("00:00:00");
    break;
case 1546339181:
    if (!(s == "Четверг"))
        break;
    if (time != null)
    {

```

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

Окончание ПРИЛОЖЕНИЯ Е

```

        this.thursday_check.Checked = true;
        this.thuStart.Value = DateTime.Parse(time.start.ToString());
        this.thuEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.thursday_check.Checked = false;
    this.thuStart.Value = DateTime.Parse("00:00:00");
    this.thuEnd.Value = DateTime.Parse("00:00:00");
    break;
case 1957370608:
    if (!(s == "ПЯТНИЦА"))
        break;
    if (time != null)
    {
        this.friday_check.Checked = true;
        this.friStart.Value = DateTime.Parse(time.start.ToString());
        this.friEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.friStart.Value = DateTime.Parse("00:00:00");
    this.friEnd.Value = DateTime.Parse("00:00:00");
    this.friday_check.Checked = false;
    break;
case 2215241657:
    if (!(s == "Суббота"))
        break;
    if (time != null)
    {
        this.saturday_check.Checked = true;
        this.satStart.Value = DateTime.Parse(time.start.ToString());
        this.satEnd.Value = DateTime.Parse(time.end.ToString());
        break;
    }
    this.satStart.Value = DateTime.Parse("00:00:00");
    this.satEnd.Value = DateTime.Parse("00:00:00");
    this.saturday_check.Checked = false;
    break;
}

}
    }

```

ПРИЛОЖЕНИЕ Ж

(обязательное)

Листинг страницы «Инвентарь клуба»

```
using SportClub_Client.Data;
using SportClub_Client.Properties;
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Windows.Forms;

namespace SportClub_Client.Views
{
    public class GymApparatusForm : Form
    {
        private List<string> statusList = new List<string>()
        {
            "Исправен",
            "Требует ревизии",
            "Не исправен",
            "На ремонте"
        };

        public GymApparatusForm()
        {
            this.InitializeComponent();
        }

        private void GymApparatusForm_Load(object sender, EventArgs e)
        {
            this.dgv.DataSource = (object) DataAccess.GetGymAparatus();
            this.apparatus_cmbx.DataSource = (object) DataAccess.GetApparatusList();
            this.apparatus_cmbx.SelectedItem = (object) null;
            this.empl_cmbx.DataSource = (object) DataAccess.GetEmployees();
            this.empl_cmbx.SelectedItem = (object) null;
            this.status_cmbx.DataSource = (object) this.statusList;
            this.status_cmbx.SelectedItem = (object) null;
            this.sStatus_cmbx.DataSource = (object) this.statusList.ToList<string>();
            this.sEmpl_cmbx.DataSource = (object) DataAccess.GetEmployees();
        }

        private void select_btn_Click(object sender, EventArgs e)
        {
            OpenFileDialog openFileDialog = (FileDialog) new OpenFileDialog();
            openFileDialog.Filter = "Image files| *.jpeg;*.jpg;*.bmp;";
            openFileDialog.FileOk += new CancelEventHandler(this.fd_FileOk);
            int num = (int) openFileDialog.ShowDialog();
        }

        private void fd_FileOk(object sender, CancelEventArgs
```

					ДП-02069964-09.02.03-04-20	Лист
Изм.	Лист	№ докум.	Подпись	Дата		63

Продолжение ПРИЛОЖЕНИЯ Ж

```

{
    this.fotoBytes = File.ReadAllBytes((sender as OpenFileDialog).FileName);
    if (this.fotoBytes != null)
        this.pictureBox1.Image = Image.FromStream((Stream) new
MemoryStream(this.fotoBytes));
    else
        this.pictureBox1.Image = (Image) null;
}

private void exit_btn_Click(object sender, EventArgs e)
{
    this.Close();
}

private void dgv_DataBindingComplete(object sender,
DataGridViewBindingCompleteEventArgs e)
{
    if (this.dgv.Columns["Status"] != null)
        this.dgv.Columns["Status"].HeaderText = "Состояние";
    if (this.dgv.Columns["LastDate"] != null)
        this.dgv.Columns["LastDate"].HeaderText = "Дата посл. обслуж.";
    if (this.dgv.Columns["InvNum"] != null)
        this.dgv.Columns["InvNum"].HeaderText = "Инв. номер";
    if (this.dgv.Columns["Foto"] != null)
        this.dgv.Columns["Foto"].Visible = false;
    if (this.dgv.Columns["Apparatus"] != null)
    {
        this.dgv.Columns["Apparatus"].HeaderText = "Снаряд";
        this.dgv.Columns["Apparatus"].DisplayIndex = 1;
    }
    if (this.dgv.Columns["Employee"] != null)
        this.dgv.Columns["Employee"].HeaderText = "Отв. сотрудник";
    if (this.dgv.Columns["Place"] != null)
        this.dgv.Columns["Place"].HeaderText = "Расположение";
    if (this.dgv.Columns["ID_GymApparatus"] == null)
        return;
    this.dgv.Columns["ID_GymApparatus"].Visible = false;
}

public bool isEdit { get; set; }
private void add_btn_Click(object sender, EventArgs e)
{
    this.EnableControls();
    this.INV_txbx.Text = this.place_txbx.Text = string.Empty;
    this.status_cmbx.SelectedItem = this.apparatus_cmbx.SelectedItem =
this.empl_cmbx.SelectedItem = (object) null;
    this.date_dtpc.Value = DateTime.Now.Date;
    this.fotoBytes = (byte[]) null;
    this.pictureBox1.Image = (Image) null;
    this.isEdit = false;
    this.edit_btn.Enabled = false;
}

```


Продолжение ПРИЛОЖЕНИЯ Ж

```

private void edit_btn_Click(object sender, EventArgs e)
{
    this.EnableControls();
    this.isEdit = true;
    this.add_btn.Enabled = false;
}
private void DisableControls()
{
    this.save_btn.Enabled = this.INV_txbx.Enabled = this.place_txbx.Enabled =
this.status_cmbx.Enabled = this.apparatus_cmbx.Enabled =
this.status_cmbx.Enabled = this.date_dtpc.Enabled = this.select_btn.Enabled =
false;
    this.add_btn.Enabled = this.edit_btn.Enabled = true;
}
private void EnableControls()
{
    this.save_btn.Enabled = this.INV_txbx.Enabled = this.place_txbx.Enabled =
this.status_cmbx.Enabled = this.apparatus_cmbx.Enabled =
this.status_cmbx.Enabled = this.date_dtpc.Enabled = this.select_btn.Enabled =
true;
}
private void обновитьДанныеToolStripMenuItem_Click(object sender, EventArgs
e)
{
    this.dgv.DataSource = (object) null;
    this.dgv.DataSource = (object) DataAccess.GetGymAparatus();
}
private void save_btn_Click(object sender, EventArgs e)
{
    if (this.dgv.SelectedRows.Count > 0 && this.apparatus_cmbx.SelectedItem
!= null && this.empl_cmbx.SelectedItem != null && this.status_cmbx.SelectedItem
!= null)
    {
        if (this.isEdit)
        {
            GumApparatus dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem
as GumApparatus;
            dataBoundItem.Apparatus = this.apparatus_cmbx.SelectedItem as
Apparatus;
            dataBoundItem.Employee = this.empl_cmbx.SelectedItem as Employee;
            dataBoundItem.Foto = this.fotoBytes;
            dataBoundItem.InvNum = this.INV_txbx.Text;
            dataBoundItem.LastDate = this.date_dtpc.Value.Date;
            dataBoundItem.Place = this.place_txbx.Text;
            dataBoundItem.Status = this.status_cmbx.Text;
            DataAccess.UpdateGymApparatus(dataBoundItem);
        }
        else
        {
            DataAccess.InsertGumApparatus(new GumApparatus())
            {

```

Окончание ПРИЛОЖЕНИЯ Ж

```

        Apparatus = this.apparatus_cmbx.SelectedItem as Apparatus,
        Employee = this.empl_cmbx.SelectedItem as Employee,
        Foto = this.fotoBytes,
        InvNum = this.INV_txbx.Text,
        LastDate = this.date_dtpc.Value.Date,
        Place = this.place_txbx.Text,
        Status = this.status_cmbx.Text
    });
}
else
{
    int num = (int) MessageBox.Show("Пожалуйста, заполните все поля!",
"Отмена операции", MessageBoxButtons.OK, MessageBoxIcon.Exclamation);
}
this.DisableControls();
this.dgv.DataSource = (object) DataAccess.GetGymAparatus();
}

private void dgv_SelectionChanged(object sender, EventArgs e)
{
    if (this.dgv.SelectedRows.Count <= 0)
        return;
    GumApparatus dataBoundItem = this.dgv.SelectedRows[0].DataBoundItem as
GumApparatus;
    this.apparatus_cmbx.SelectedItem = (object) dataBoundItem.Apparatus;
    this.empl_cmbx.SelectedItem = (object) dataBoundItem.Employee;
    this.fotoBytes = dataBoundItem.Foto;
    this.INV_txbx.Text = dataBoundItem.InvNum;
    this.date_dtpc.Value = dataBoundItem.LastDate;
    this.place_txbx.Text = dataBoundItem.Place;
    this.status_cmbx.Text = dataBoundItem.Status;
    this.pictureBox1.Image = dataBoundItem.Foto == null ? (Image) null :
Image.FromStream((Stream) new MemoryStream(this.fotoBytes));
    this.DisableControls();
    this.isEdit = false;
}
}

```

ПРИЛОЖЕНИЕ И

(обязательное)

Техническое задание

ИП ««Silver's gym»»

УТВЕРЖДАЮ

Руководитель ИП«Silver's gym »

УТВЕРЖДАЮ

Начальник управления

Личная подпись Расшифровка подписи

Дата

Личная подпись

Расшифровка подписи

Дата _____

ТЕХНИЧЕСКОЕ ЗАДАНИЕ

На 17 листах

Действует с 09.04.2020

СОГЛАСОВАНО

Руководитель ВЦ

Личная подпись

Расшифровка подписи

Дата _____

Саранск 2020

					ДП-02069964-09.02.03-04-20	Лист
						67
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

1. Общие сведения

1.1. Наименование системы

1.1.1. Полное наименование системы

1.1.2. Краткое наименование системы

1.2. Основания для проведения работ

1.3. Наименование организаций – Заказчика и Разработчика

1.3.1. Заказчик

1.3.2. Разработчик

1.4. Плановые сроки начала и окончания работы

1.5. Источники и порядок финансирования

1.6. Порядок оформления и предъявления заказчику результатов работ

1.7. Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

2. Назначение и цели создания системы

2.1. Назначение системы

2.2. Цели создания системы

3. Характеристика объектов автоматизации

4. Требования к системе

4.1. Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы

4.1.1.1. Требования к способам и средствам связи для информационного обмена между компонентами системы

4.1.1.2. Требования к характеристикам взаимосвязей создаваемой системы со смежными системами

4.1.2. Требования к численности и квалификации персонала системы

4.1.2.1. Требования к численности персонала

4.1.2.2. Требования к квалификации персонала

4.1.3. Показатели назначения

					ДП-02069964-09.02.03-04-20	Лист
						68
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

4.1.4. Требования к надежности

4.1.5. Требования к эргономике и технической эстетике

4.1.6. Требования к эксплуатации, техническому
обслуживанию, ремонту и хранению компонентов системы

4.1.7. Требования к защите информации от
несанкционированного доступа

4.1.8. Требования по сохранности информации при авариях

4.2. Требования к функциям (задачам), выполняемым системой

4.3. Требования к видам обеспечения

5. Состав и содержание работ по созданию (развитию) системы

5.1. Перечень стадий и этапов работы

6. Порядок контроля и приемки системы

7. Требования к составу и содержанию работ по подготовке объекта
автоматизации к вводу системы в действие

8. Требования к документированию

					ДП-02069964-09.02.03-04-20	Лист
						69
Изм.	Лист	№ докум.	Подпись	Дата		

1. Общие сведения

1.1. Наименование системы

1.1.1. Полное наименование системы

Полное наименование: Автоматизированная информационная система «Silver's gym» для комплексной автоматизации деятельности администратора в фитнес клубе.

1.1.2. Краткое наименование системы

Краткое наименование: АИС «Silver's gym»

1.2. Основания для проведения работ

Работа выполняется на основании договора.

1.3. Наименование организаций – Заказчика и Разработчика

1.3.1. Заказчик

Заказчик: ОАО «Silver's gym»

Адрес фактический: г. Саранск, ул. Полежаева, д.9, кв.11

					ДП-02069964-09.02.03-04-20	Лист
						70
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

Телефон / Факс: +7 (495) 2222222

1.3.2. Разработчик

Разработчик: ИП Горюнов

Адрес фактический: г. Саранск, ул. Ярославская, д.14, кв.41

Телефон / Факс: +7 (8342) 560200

1.4. Плановые сроки начала и окончания работы

Плановый срок начала работ по созданию АИС «Silver's gym» — 30 марта 2020 года.

Плановый срок окончания работ по — 11 апреля 2020 года.

1.5. Источники и порядок финансирования

Источником финансирования является бюджет ОАО «Silver's gym».

Порядок финансирования определяется условиями Договора.

1.6. Порядок оформления и предъявления заказчику результатов работ

Работы по созданию Фитнес-клуба сдаются Разработчиком поэтапно в соответствии с календарным планом Проекта. По окончании каждого из этапов работ Разработчик сдает Заказчику соответствующие отчетные документы этапа, состав которых определены Договором.

					ДП-02069964-09.02.03-04-20	Лист
						71
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

1.7. Перечень нормативно-технических документов, методических материалов, использованных при разработке ТЗ

При разработке автоматизированной системы и создании проектно-эксплуатационной документации Исполнитель должен руководствоваться требованиями следующих нормативных документов:

- ГОСТ 19.201-78. Техническое задание. Требования к содержанию и оформлению;
- ГОСТ 34.601-90. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания;
- ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплексность и обозначение документов при создании автоматизированных систем.

					ДП-02069964-09.02.03-04-20	Лист
						72
Изм.	Лист	№ докум.	Подпись	Дата		

2. Назначение и цели создания системы

2.1. Назначение системы

АИС «Silver's gym» предназначена для повышения оперативности и качества обслуживания клиентов и обновления услуг.

Основным назначением АИС «Silver's gym» является автоматизация процессов фитнес клуба Заказчика.

2.2. Цели создания системы

Целью создания АИС «Silver's gym» является улучшение качества предоставляемых фитнес услуг за счет улучшения качества выполнения управленческих, расчетных, аналитических, распределительных и других задач, а также улучшение прибыли.

3. Характеристика объектов автоматизации

Объектом автоматизации для разрабатываемой системы выступает фитнес клуб, в рамках которой возникает потребность в автоматизации различных функций: учёт клиентов, оформление абонементов, учёта спортивного инвентаря, составление отчета за месяц об оказанных услугах.

					ДП-02069964-09.02.03-04-20	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

4. Требования к системе

4.1. Требования к системе в целом

4.1.1. Требования к структуре и функционированию системы

Система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище.

В системе предлагается выделить следующие функциональные подсистемы:

- подсистема сбора;
- подсистема хранения данных;
- подсистема формирования и визуализации отчетности;
- подсистема оптимизации данных.

4.1.2. Требования к численности и квалификации персонала системы и режиму его работы

4.1.2.1. Требования к численности персонала

В состав персонала, необходимого для обеспечения эксплуатации АИС «Silver's gym» в рамках соответствующих подразделений, необходимо выделить следующих ответственных лиц:

- Руководитель IT-отдела – 1 человек.

Продолжение ПРИЛОЖЕНИЯ И

– Администратор – 1 человека.

Данные лица должны выполнять следующие функциональные обязанности.

– Руководитель IT-отдела – на всем протяжении функционирования АИС обеспечивает общее руководство группой сопровождения;

– Администратор — на всем протяжении функционирования АИС обеспечивает распределение дискового пространства, модификацию структур базы данных, оптимизацию производительности.

4.1.2.2. Требования к квалификации персонала

К квалификации персонала, эксплуатирующего АИС «Silver's gym», предъявляются следующие требования.

– Конечный пользователь – знание соответствующей предметной области; знание основ работы с ПК; знания и навыки работы с аналитическими приложениями.

– Администратор – понимание принципов многомерного анализа; знание методологии проектирования хранилищ данных; знание и навыки администрирования приложения.

4.1.3. Показатели назначения

АИС должна обеспечивать возможность исторического хранения данных с глубиной не менее 5 лет. Система должна предусматривать возможность

					ДП-02069964-09.02.03-04-20	Лист
						76
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

масштабирования по производительности и объему обрабатываемой информации без модификации ее программного обеспечения путем модернизации используемого комплекса технических средств. Возможности масштабирования должны обеспечиваться средствами используемого базового программного обеспечения.

4.1.4. Требования к надежности

Уровень надежности должен достигаться согласованным применением организационных, организационно-технических мероприятий и программно-аппаратных средств.

Надежность должна обеспечиваться за счет:

- применения технических средств, системного и базового программного обеспечения, соответствующих классу решаемых задач;
- своевременного выполнения процессов администрирования;
- соблюдения правил эксплуатации и технического обслуживания программно-аппаратных средств;
- предварительного обучения пользователей и обслуживающего персонала.

4.1.5. Требования к эргономике и технической эстетике

Подсистема формирования и визуализации отчетности данных должна обеспечивать удобный для конечного пользователя интерфейс, отвечающий требованиям.

					ДП-02069964-09.02.03-04-20	Лист
						77
Изм.	Лист	№ докум.	Подпись	Дата		

4.1.6. Требования к эксплуатации, техническому обслуживанию, ремонту и хранению компонентов системы

Система должна быть рассчитана на эксплуатацию в составе программно–технического комплекса заказчика. Техническая и физическая защита аппаратных компонентов системы, носителей данных, бесперебойное энергоснабжение, резервирование ресурсов, текущее обслуживание реализуется техническими и организационными средствами, предусмотренными в инфраструктуре заказчика.

Для нормальной эксплуатации разрабатываемой системы должно быть обеспечено бесперебойное питание ПЭВМ. При эксплуатации система должна быть обеспечена соответствующая стандартам хранения носителей и эксплуатации ПЭВМ температура и влажность воздуха.

АИС должна обладать функцией мягкой остановки своей работы с закрытием активных подключений, сохранением информации, остановкой соответствующих служб.

Переходу в режим ожидания выключения или перезагрузки. По завершению регламентных работ, АИС с помощью предусмотренной функции должна возвращаться к штатной работе, восстанавливая работу своих служб и компонентов.

4.1.7. Требования к защите информации от несанкционированного доступа

Обеспечение информационное безопасности Системы КХД должно удовлетворять следующим требованиям:

Продолжение ПРИЛОЖЕНИЯ И

- Защита АИС должна обеспечиваться комплексом программно-технических средств и поддерживающих их организационных мер.
- Защита должна обеспечиваться на всех технологических этапах обработки информации и во всех режимах функционирования, в том числе при проведении ремонтных и регламентных работ.
- Программно-технические средства защиты не должны существенно ухудшать основные функциональные характеристики (надежность, быстродействие, возможность изменения конфигурации).
- Разграничение прав доступа пользователей и администраторов должно строиться по принципу «что не разрешено, то запрещено».

4.1.8. Требования по сохранности информации при авариях

В АИС должно быть обеспечено резервное копирование данных. Выход из строя трех жестких дисков дискового массива не должен сказываться на работоспособности подсистемы хранения данных.

					ДП-02069964-09.02.03-04-20	Лист
						79
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

4.2. Требования к функциям (задачам), выполняемым системой

Таблица 1 – Перечень функций, задач подлежащей автоматизации

Функция	Задача
Управление процессами сбора, обработки и загрузки данных	Создание, редактирование и удаление процессов сбора, обработки и загрузки данных
	Формирование последовательности выполнения процессов сбора, обработки и загрузки данных (регламентов загрузки данных)
	Определение и изменение расписания процессов сбора, обработки и загрузки данных
Выполнение процессов сбора, обработки и загрузки данных из источников в ХД	Запуск процедур сбора данных из систем источников, загрузка данных в область временного, постоянного хранения, обработка и преобразование извлечённых данных, поддержка медленно меняющихся измерений
Протокол результатов сбора, обработки и загрузки данных	Ведение журналов результатов сбора, обработки и загрузки данных
	Оперативное извещение пользователей о всех нештатных ситуациях в процессе работы подсистемы

4.3. Требования к видам обеспечения

Программа должна обеспечивать переносимость в рамках операционных систем семейства Windows.

Программа должна работать по архитектуре «клиент-сервер», поддерживать несколько одновременных соединений.

Программа должна вести журнал своей деятельности, куда будут сохраняться все сообщения об ошибках, нарушениях передачи и прочих проблемах.

Минимальная аппаратная конфигурация системы, обеспечивающей нормальное функционирование обучающей системы должна быть не ниже следующей:

- ПЭВМ, стандартной конфигурации;
- операционная система;
- объем свободного пространства на жестком диске: 690 Мб;
- объем оперативной памяти: для более эффективной работы необходимо 512 Мб.

При использовании автоматизированной информационной системы в локальной сети, на одном из компьютеров должен быть установлен и запущен сервер с предварительно установленной на нем базой данных. На остальных машинах необходимо установить клиентское приложение.

5. Состав и содержание работ по созданию (развитию) системы

5.1. Перечень стадий и этапов работы

Состав и содержание работ по созданию системы включают следующие этапы:

1. Предпроектное обследование, сбор необходимой информации.
Результат: - определение целей, задач системы, которые в дальнейшем должны быть решены;

2. Анализ предметной области. Результат: подробный анализ системы и введение организационных требований к решению задач и целей;

3. Разработка ТЗ. Результат- документация на разрабатываемую систему, в которой указаны сроки реализации, кем будет реализована, для кого, описаны все необходимые организационные требования к разработке;

4. Разработка модели программы. Результат – описание спецификаций данных, определение связей между сущностями, построение концептуальной модели базы данных, построение логической модели базы данных;

5. Разработка технического продукта;

6. Разработка рабочего проекта, состоящего из:

- написания программы;
- отладка программы;
- корректировка программы.

7. Проведение тестирования и доработка информационного программного обеспечения по замечаниям и предложениям;

8. Сдача системы в эксплуатацию с выпуском описания алгоритмов и технологической документации.

6. Порядок контроля и приемки системы

Разработанная система принимается в опытную эксплуатацию. Готовые компоненты системы могут передаваться поочередно. Сдача и приемка автоматизированной информационной системы осуществляется на основе результатов тестирования, проводимого представителями Заказчика и Исполнителя в соответствии с программой испытания, которая формируется совместно. В программе испытания должны быть указаны виды, состав, объем и методы проверки правильности получения выходных данных и соответствия системы требованиям данного технического задания.

Для проверки работоспособности системы проводится выполнение контрольных примеров. Составление контрольных примеров с последующей их передачей комиссии, производится эксплуатационной службой и разработчиками совместно. Для выполнения контрольного примера должен быть предоставлен программно-аппаратный комплекс, удовлетворяющий требованиям, изложенным в подразделе «Требования к видам обеспечения» настоящего документа. По результатам выполнения тестов комиссией составляется и рассматривается в течение 3 дней перечень замечаний.

Продолжение ПРИЛОЖЕНИЯ И

7. Требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие

АИС «Silver's gym» должна быть проста в использовании и не требовать сложной специальной подготовки пользователей.

Основные мероприятия по вводу системы в действие включают: создание условий функционирования объекта автоматизации, при которых гарантируется соответствие создаваемой системы требованиям, содержащимся в ТЗ.

					ДП-02069964-09.02.03-04-20	Лист
						84
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

8. Требования к документированию

Для системы на различных стадиях создания должны быть выпущены следующие документы:

- ведомость технического проекта;
- пояснительная записка к техническому проекту;
- руководство пользователя;
- общее описание системы;
- листинг кода;
- результаты тестирования.

					ДП-02069964-09.02.03-04-20	Лист
						85
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ И

Перечень принятых сокращений

КХД	Корпоративное хранилище данных
ТЗ	Техническое задание
АИС	Автоматизированная информационная система
ПЭВМ	Персональная электронно-вычислительная машина

Окончание ПРИЛОЖЕНИЯ И

СОСТАВИЛИ

Наименование организации, предприятия	Должность исполнителя	Фамилия имя, отчество	Подпись	Дата

СОГЛАСОВАНО

Наименование организации, предприятия	Должность исполнителя	Фамилия имя, отчество	Подпись	Дата

ПРИЛОЖЕНИЕ К

(обязательное)

Руководство пользователя

1. Введение

1.1. Область применения

АИС «Silver's gym» предназначен для повышения оперативности и качества выполняемых услуг и обновления отчётности.

Основным назначением АИС «Silver's gym» является автоматизация процессов предоставления услуг фитнес клуба Заказчика.

1.2. Краткое описание возможностей

Целью создания АИС «Silver's gym» является улучшение качества предоставляемых фитнес клуба услуг за счет улучшения качества выполнения управленческих, расчетных, аналитических, распределительных и других задач, а также улучшение прибыли.

1.3. Уровень подготовки пользователя

К квалификации персонала, эксплуатирующего АИС «Silver's gym», предъявляются следующие требования.

– Конечный пользователь – знание соответствующей предметной области; знание основ работы с ПК; знания и навыки работы с аналитическими приложениями.

					ДП-02069964-09.02.03-04-20	Лист
						88
Изм.	Лист	№ докум.	Подпись	Дата		

Продолжение ПРИЛОЖЕНИЯ К

– Администратор – понимание принципов многомерного анализа; знание методологии проектирования хранилищ данных; знание и навыки администрирования приложения.

Продолжение ПРИЛОЖЕНИЯ К

1.4. Перечень эксплуатационной документации

Перечень эксплуатационных документов, с которым необходимо ознакомиться:

- «Руководство администратора»;
- «Руководство пользователя»;
- Т.д.

					ДП-02069964-09.02.03-04-20	Лист
						90
Изм.	Лист	№ докум.	Подпись	Дата		

2. Назначение и условия применения

2.1. Виды деятельности, функции

Руководство администратора предназначена для автоматизации следующих видов деятельности:

– Улучшение качества предоставляемых фитнес клубом услуг за счет улучшения качества выполнения управленческих, расчетных, аналитических, распределительных и других задач, а также улучшение прибыли;

Объектом автоматизации для разрабатываемой системы выступает фитнес клуб, в рамках которой возникает потребность в автоматизации различных функций: учёт клиентов, оформление абонементов, учёта спортивного инвентаря, составление отчета за месяц об оказанных услугах.

2.2. Программные и аппаратные требования к системе

Система должна быть централизованной, т.е. все данные должны располагаться в центральном хранилище.

В системе предлагается выделить следующие функциональные подсистемы:

- подсистема сбора;
- подсистема хранения данных;
- подсистема формирования и визуализации отчетности;
- подсистема оптимизации данных.

3. Подготовка к работе

3.1. Состав дистрибутива

- клиентская часть windows;
- MySQL.

3.2. Запуск системы

Запуск клиентского приложения предоставленного для работы АИС «Silver's gym».

3.3. Проверка работоспособности системы

АИС «Silver's gym» считается работоспособным, если в результате действий пользователя, изложенных в п.п.3.2, на экране монитора отобразилось главная экрана интерфейса без выдачи пользователю сообщений о сбое в работе.

4. Описание операций

АИС «Silver's gym» позволяет администратору осуществлять оформление абонементов. Следить за учетом посещения и спортивного инвентаря.

4.1. Оформление абонемента

Оформление абонемента осуществляется после выбора пользователем необходимого вида абонемента. После выбора товара пользователь должен нажать на «Оформление абонемента», далее ввести данные клиента и активировать абонемент.

4.2. Вход в личный кабинет

Если пользователь зарегистрирован, то необходимо ввести ФИО пользователя и пароль.

5. Аварийные ситуации

Потеря данных потерять в результате аварийного сбоя, ошибочных действий персонала или намеренных действий злоумышленников.

Доступ к данным может быть нарушен в результате аварии или сбоя компьютерной сети. Также он может прекратиться, если перестали работать серверы, обеспечивающие пользовательский интерфейс информационной системы.

Конфиденциальность данных может быть утрачена, если в информационную систему попал кто-то посторонний. Кроме того, имеется потенциальная возможность похищения данных на уровне файлов или их носителей (дисков и т. п.).

АИС «Silver's gym» предусматривает резервное копирование по определенному расписанию для обеспечения устойчивости к сбоям различного характера.

6. Рекомендации по освоению

Для успешного освоения приложения АИС «Silver's gym» со стороны администратора необходимо иметь навыки работы с ПК и изучить следующее:

- Нормативно-правовую базу по вопросам управления государственными кадрами;
- Раздел «Описание процесса деятельности» документа «Пояснительная записка (Технический проект)»;
- Раздел «Описание автоматизируемых функций» документа «Пояснительная записка (Технический проект)»;
- Настоящее «Руководство пользователя».