

Содержание

Введение	3
1 Назначение и цели разработки	4
2 Разработка технического проекта на основе анализа требований	6
2.1 Определение спецификаций программного обеспечения	6
2.2 Проектирование модели данных и диаграммы классов	12
2.3 Конструирование прототипа	15
3 Реализация	20
3.1 Обоснование выбора средств разработки	20
3.2 Разработка БД в среде СУБД	21
3.3 Описание программных модулей	22
4 Тестирование программных модулей	23
4.1 Модульное тестирование	23
4.2 Интеграционное тестирование	25
Заключение	26
Список использованных источников	27
Приложение А Техническое задание	30
Приложение Б Руководство по стилю	36
Приложение В Программный код	37
Приложение Г SQL-скрипт базы данных	45
Приложение Д Тест-кейсы	47

					ККЭП 09.02.07 023_19 ПЗ			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.		Лундин И.А.			Разработка модулей подсистем "Администратор" и "Личный кабинет контрагента" информационной системы "Профи"			
Провер.		Головки Р.А.						
						Лит.	Лист	Листов
						КП	2	49
						Гр. 23-Д9-ЗИСП		

Введение

«Профи» — это программная система, предназначенная для автоматизации добавления, изменения и удаления, а также поиска секций дополнительного образования, именуемые в дальнейшем СДО, и контрагентов.

Системой будут пользоваться специалисты по дополнительному образованию, для которых разработаны роли администратор, взаимодействующий с контрагентами, и контрагент, взаимодействующий с секциями дополнительного образования, в дальнейшем именуемые СДО.

Приложение на данный момент выполняет функции по разграничению доступа к приложению, работе с историей входа, контрагентами и СДО, фильтрация и поиск СДО и контрагентов, работа с печатной формой и экспорт в pdf.

					<i>ККЭП 09.02.07 023_11 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		3

1 Назначение и цели разработки

Техническим заданием, представленным в приложении А, предусмотрена разработка программной системы «Профи», предназначенная для автоматизации добавления, изменения и удаления, а также поиска СДО и контрагентов, для этого были разработаны модули:

- вход в систему;
- журнал входов;
- список СДО;
- список контрагентов;
- добавление и редактирование СДО;
- добавление и редактирование контрагентов.

Целью разработки является создание удобной системы добавления, редактирования и удаления СДО и контрагентов.

Назначением системы является улучшение качества жизни пользователей, предоставляя им удобный и надежный инструмент для принятия решений, связанных с СДО.

Таким образом, система «Профи» является универсальным инструментом, который поможет пользователям справляться с различными вызовами, связанными с СДО.

2 Разработка технического проекта на основе анализа требований

2.1 Определение спецификаций программного обеспечения

Рассмотрим определение вариантов использования (прецедентов).

Программная система требуется следующим заинтересованным лицам:

- менеджер центра дополнительного образования;
- желающий обучаться в СДО;
- обучающийся в СДО;
- специалисты по СДО (программисты ООО «Профи»).

На данный момент система ограничена двумя ролями действующих лиц: контрагент (менеджер центра дополнительного образования) и администратор (программисты ООО «Профи»), добавление в систему новых ролей и их функционала будет реализовано в будущем.

Соответственно основные прецеденты для разрабатываемой системы для роли контрагент:

- добавление и редактирование СДО;
- удаление СДО.

Основные варианты использования администратора:

- добавление и редактирование контрагентов;
- удаление контрагентов;
- просмотр журнала входов.

Также есть основные варианты, являющиеся общими для администратора и контрагента:

- просмотр СДО;
- просмотр контрагентов.

Представим диаграмму вариантов использования, созданную средствами StarUML для проектируемого ПО на рисунке 1.

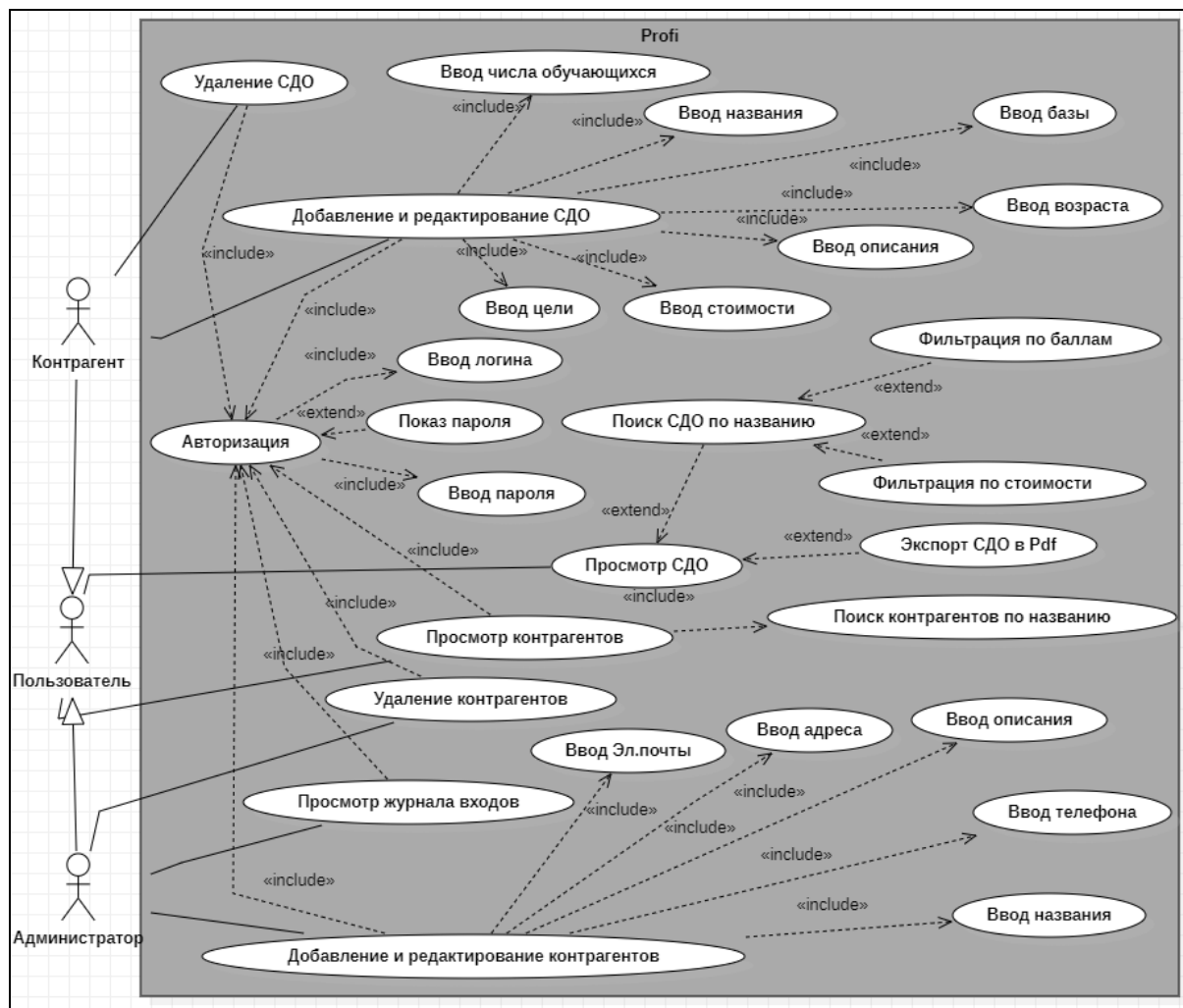


Рисунок 1 – Диаграмма вариантов использования системы «Профи»

Были сформированы требования для выполнения сценариев добавление СДО (таблицы 1-3), а к ним диаграмма деятельности и диаграмма последовательности (рисунки 2-3), редактирование контрагента (таблицы 4-6), удаление контрагента (таблицы 7-9).

Таблица 1 – Главный раздел сценария прецедента «Добавление СДО»

Вариант использования	Добавление СДО
Актеры	Контрагент
Краткое описание	Пользователь авторизуется как контрагент, использует модуль добавления и редактирования СДО, вводит данные о новом СДО
Цель:	Добавить новый СДО для дальнейшей работы
Тип	Базовый

Таблица 2 – Сценарий успешного выполнения варианта использования «Добавление СДО»

Действия актеров	Отклик системы
1. Пользователь авторизуется как контрагент, вводя логин и пароль Исключение 1. Такого пользователя нет в системе Исключение 2. Пользователь не является контрагентом 4. Пользователь вводит данные о новом СДО Исключение 3. Введенные данные некорректны или неполны	2. Система проверяет наличие пользователя в системе. 3. Система проверяет, является ли пользователь контрагентом. 5. Система проверяет корректность введенных данных 6. Система добавляет новое СДО

Таблица 3 – Обработка исключительных ситуаций для прецедента «Добавление СДО»

Действия актеров	Отклик системы
Исключение 1. Такого пользователя нет в системе	
1. Пользователь авторизуется как контрагент, вводя логин и пароль	2. Система оповещает пользователя о том, что такого пользователя нет в системе
Исключение 2. Пользователь не является контрагентом	
1. Пользователь авторизуется как контрагент, вводя логин и пароль	2. Система скрывает от пользователя интерфейс для добавления СДО
Исключение 3. Введенные данные некорректны или неполны	
1. Пользователь вводит данные о новом СДО	2. Система оповещает пользователя о том, что данные неполны или некорректны

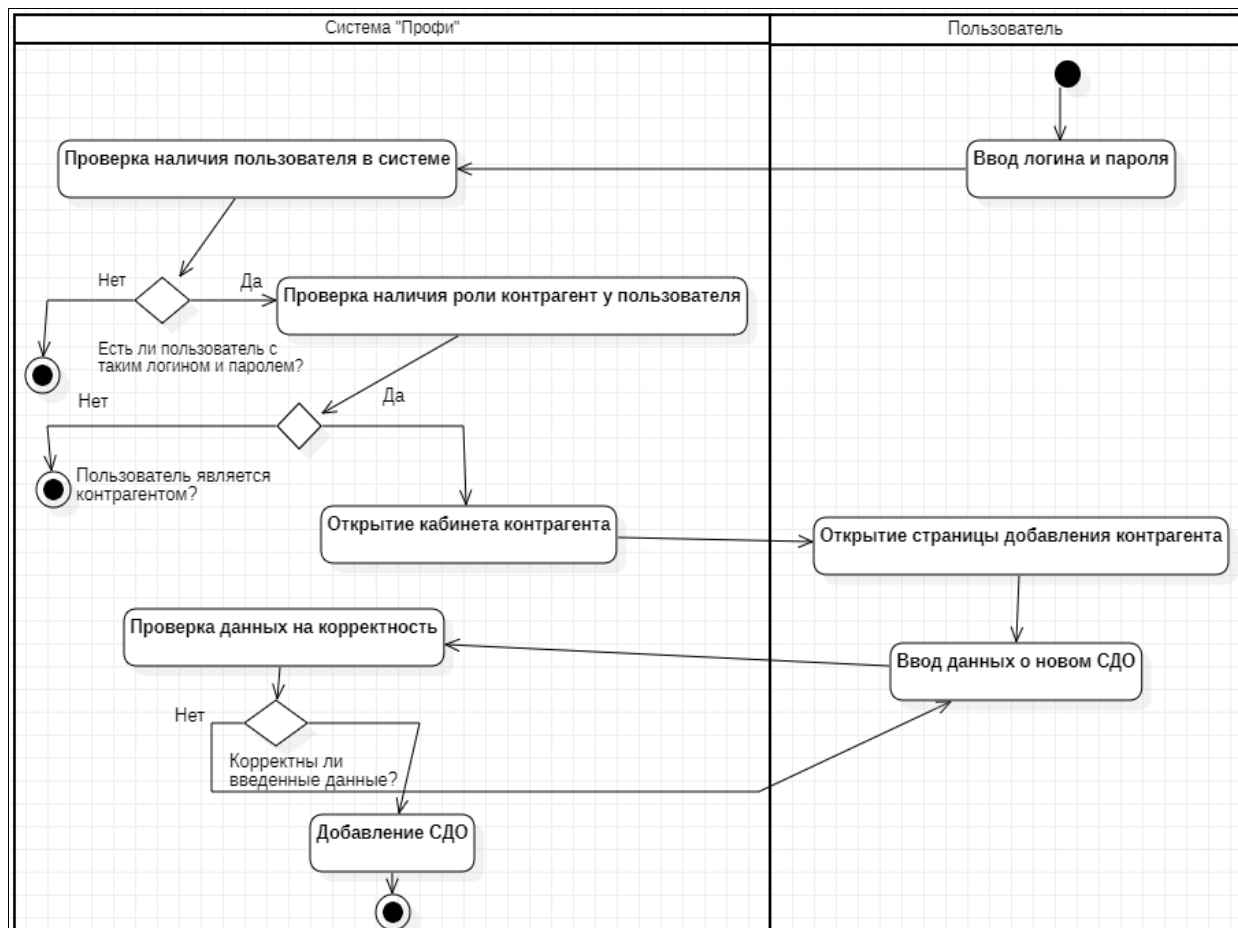


Рисунок 2 – Диаграмма деятельности к прецеденту «Добавление СДО»



Рисунок 3 – Диаграмма последовательности к прецеденту «Добавление СДО»

Таблица 4 – Главный раздел сценария прецедента «Редактирование контрагента»

Вариант использования	Редактирование контрагента
Актеры	Администратор
Краткое описание	Пользователь авторизуется как администратор, использует модуль добавления и редактирования контрагента, вводит новые данные о контрагенте
Цель:	Обновление данных о контрагенте
Тип	Базовый

Таблица 5 – Сценарий успешного выполнения прецедента «Редактирование контрагента»

Действия актеров	Отклик системы
<p>1. Пользователь авторизуется как администратор, вводя логин и пароль</p> <p>Исключение 1. Такого пользователя нет в системе</p> <p>Исключение 2. Пользователь не является администратором</p> <p>4. Пользователь вводит новые данные о контрагенте</p> <p>Исключение 3. Введенные данные некорректны или неполны</p>	<p>2. Система проверяет наличие пользователя в системе.</p> <p>3. Система проверяет, является ли пользователь администратором.</p> <p>5. Система проверяет корректность введенных данных.</p> <p>6. Система обновляет данные об контрагенте.</p>

Таблица 6 – Обработка исключительных ситуаций для прецедента
«Редактирование контрагента»

Действия актеров	Отклик системы
Исключение 1. Такого пользователя нет в системе	
1. Пользователь авторизуется как контрагент, вводя логин и пароль	2. Система оповещает пользователя о том, что такого пользователя нет в системе
Исключение 2. Пользователь не является контрагентом	
1. Пользователь авторизуется как администратор, вводя логин и пароль	2. Система скрывает от пользователя интерфейс для редактирования контрагентов
Исключение 3. Введенные данные некорректны или неполны	
1. Пользователь вводит новые данные о контрагенте	2. Система оповещает пользователя о том, что данные неполны или некорректны

Таблица 7 – Главный раздел сценария прецедента «Удаление контрагента»

Вариант использования	Удаление контрагента
Актеры	Администратор
Краткое описание	Пользователь авторизуется как администратор, использует модуль просмотра контрагентов и удаляет контрагента
Цель:	Удалить неактивного контрагента
Тип	Базовый

Таблица 8 – Сценарий успешного выполнения прецедента «Удаление контрагента»

Действия актеров	Отклик системы
1. Пользователь авторизуется как администратор, вводя логин и пароль	2. Система проверяет наличие пользователя в системе.
Исключение 1. Такого пользователя нет в системе	3. Система проверяет, является ли пользователь администратором.
Исключение 2. Пользователь не является администратором	5. Система удаляет контрагента
4. Пользователь удаляет контрагента	Исключение 3. В БД есть записи о контрагенте в связанных таблицах

Таблица 9 – Обработка исключительных ситуаций для прецедента «Удаление контрагента»

Действия актеров	Отклик системы
Исключение 1. Такого пользователя нет в системе	
1. Пользователь авторизуется как администратор, вводя логин и пароль	2. Система оповещает пользователя о том, что такого пользователя нет в системе
Исключение 2. Пользователь не является администратором	
1. Пользователь авторизуется как администратор, вводя логин и пароль	2. Система скрывает от пользователя интерфейс для добавления СДО
Исключение 3. В БД есть записи о контрагенте в связанных таблицах	
1. Пользователь удаляет контрагента	2. Система оповещает пользователя о том, что контрагента нельзя удалить, т.к. в БД есть записи о нем в связанных таблицах

Таким образом были проработаны основные варианты использования системы «Профи».

2.2 Проектирование модели данных и диаграммы классов

Для построения ER-диаграммы применялись средства среды StarUML (рисунок 4), характеристики диаграммы представлены в таблице 10. На основе ER-диаграммы была создана диаграмма со схожими сущностями (рисунок 5).

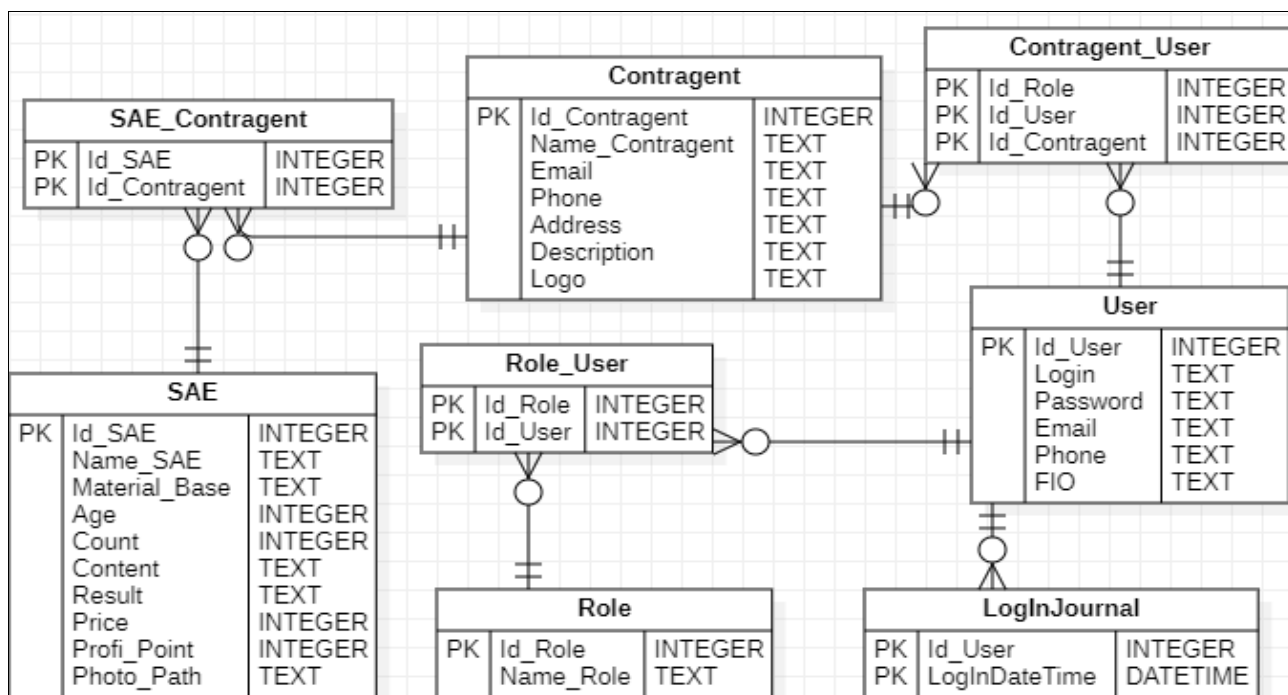


Рисунок 4 – ER-диаграмма БД для системы «Профи»

Таблица 10 – Характеристики сущностей и их связей в ER-диаграмме

Сущность	Назначение сущности	Ключ
Role	Данные о ролях в системе: — id роли; — название роли.	Id роли (PK)
LogInJournal	Данные о входе пользователей в систему: — id пользователя; — дата и время входа.	id пользователя, дата и время входа (PK)

Продолжение таблицы 10

Сущность	Назначение сущности	Ключ
User	Данные о пользователях в системе: — id пользователя; — логин; — пароль; — эл.почта; — телефон; — ФИО.	Id пользователя (PK)
Role_User	Служебная таблица для связи пользователей и ролей: — id роли; — id пользователя.	Id пользователя и Id роли (PK), Id пользователя (FK, User), Id роли (FK, Role)
Contragent	Данные о контрагентах: — id контрагента; — название контрагента; — эл.почта; — телефон; — адрес; — описание; — логотип.	Id контрагента (PK)
Contragent_User	Служебная таблица для связи пользователей роли с контрагентами: — id контрагента; — id пользователя; — id роли.	id контрагента, id пользователя, id роли (PK), id контрагента (FK, Contragent), id пользователя, id роли (FK, Role_User)
SAE	Данные о СДО: — id СДО; — название СДО; — мат.тех. база; — возраст обучающихся; — количество обучающихся.	Id СДО (PK)
SAE_Contragent	Служебная таблица для связи контрагентов и СДО: — id СДО; — id контрагента.	Id СДО, id пользователя (PK), Id СДО (FK, SAE), id пользователя (FK, Contragent)

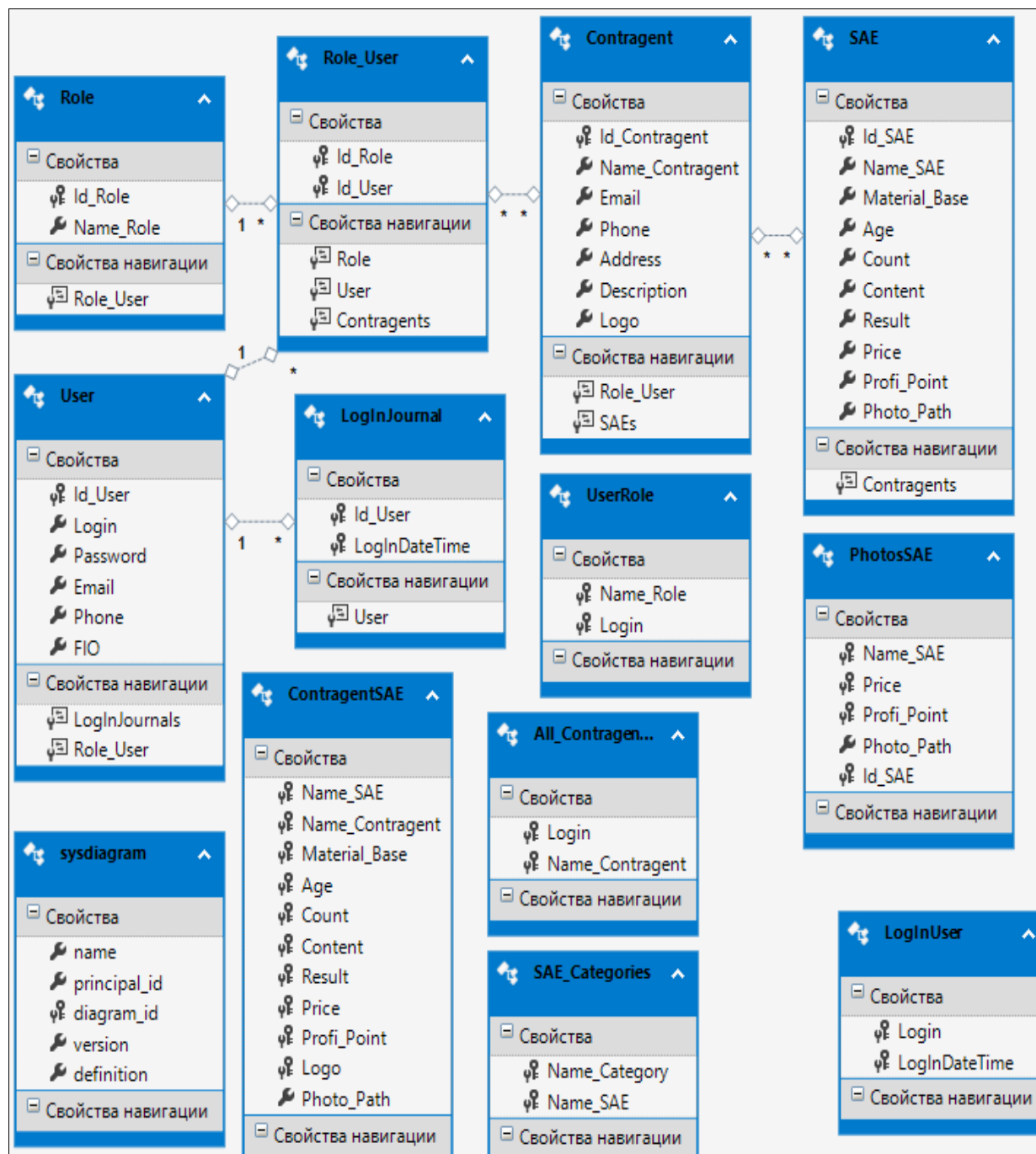


Рисунок 5 – Диаграмма классов, сформированная конструктором на основе ER-диаграммы.

Таким образом были сформированы необходимые диаграммы для реализации приложения.

2.3 Конструирование прототипа

Для построения прототипа первоначально были составлены требования к пользовательскому интерфейсу, подробно изложенные в руководстве по стилю в приложения Б. Основные экранные формы представлены на рисунках 6-14.

Рисунок 6 – Окно входа


Рисунок 7 – Страница журнала входа

Изм.	Лист	№ докум.	Подпись	Дата

ККЭП 09.02.07 023_11 ПЗ


Лист

15



Профи

Admin Понасенков Евгений Николаевич




Название: Школа бокса Тайсон

Стоимость: 4000

Профи-баллы: 100

Изменить

Открыть



Название: Клуб вязания крючком Юные рукодель

Стоимость: 1000

Профи-баллы: 30

Изменить

Открыть



Название: Кружок шахмат Юный Магнус Карлсен

Поиск по

Стоимости По умолчанию


Профи-баллам По умолчанию

Контрагенты

Выйти

Журнал

Рисунок 8 – Страница списка СДО



Название СДО: Школа бокса Тайсон

Название контрагента: Центр спортивного развития Богатырь

Десять груш, сто наборов гантель и перчаток

Материально-техническая база:

Возраст учащихся: 18

Количество учащихся в группе: 100

Нет данных

Описание программы:

Нет данных

Цели программы:

Редактировать

Экспорт PDF

Рисунок 9 – Страница конкретного СДО

Название СДО:	Школа бокса Тайсон
Материально-техническая база:	Десять груш, сто наборов гантель и перчаток
Возраст учащихся:	18
Количество учащихся в группе:	100
Описание программы:	Нет данных
Цели программы:	Нет данных
Цена:	4000
Профи-баллы:	100

Рисунок 10 – Страница добавления и редактирования СДО




	<p>Название: Центр спортивного развития Богатырь</p> <p>Телефон: 9991121127</p> <p>Эл.почта: bogatyr@mail.ru</p> <p>Адрес: Байбакова 20/1</p> <p> <input type="button" value="Изменить"/> <input type="button" value="Открыть"/> </p>
	<p>Название: Дворец культуры Горький</p> <p>Телефон: 9991918991</p> <p>Эл.почта: bitter1917@gmail.com</p> <p>Адрес: Ленина 1</p> <p> <input type="button" value="Изменить"/> <input type="button" value="Открыть"/> </p>

Рисунок 11 – Страница списка контрагентов



Название: Центр спортивного развития Богатырь
 Эл.почта: bogatyr@mail.ru
 Телефон: 9991121127
 Адрес: Байбакова 20/1
 Описание: Центр спортивного развития, растящий победителей соревнований по боксу с 2015

Рисунок 12 – Страница конкретного контрагента



Название: Центр спортивного развития Богатырь
 Эл.почта: bogatyr@mail.ru
 Телефон: 9991121127
 Адрес: Байбакова 20/1
 Описание: Центр спортивного развития, растящий победителей соревнований по боксу с 2015

Рисунок 13 – Страница добавления и редактирования контрагента

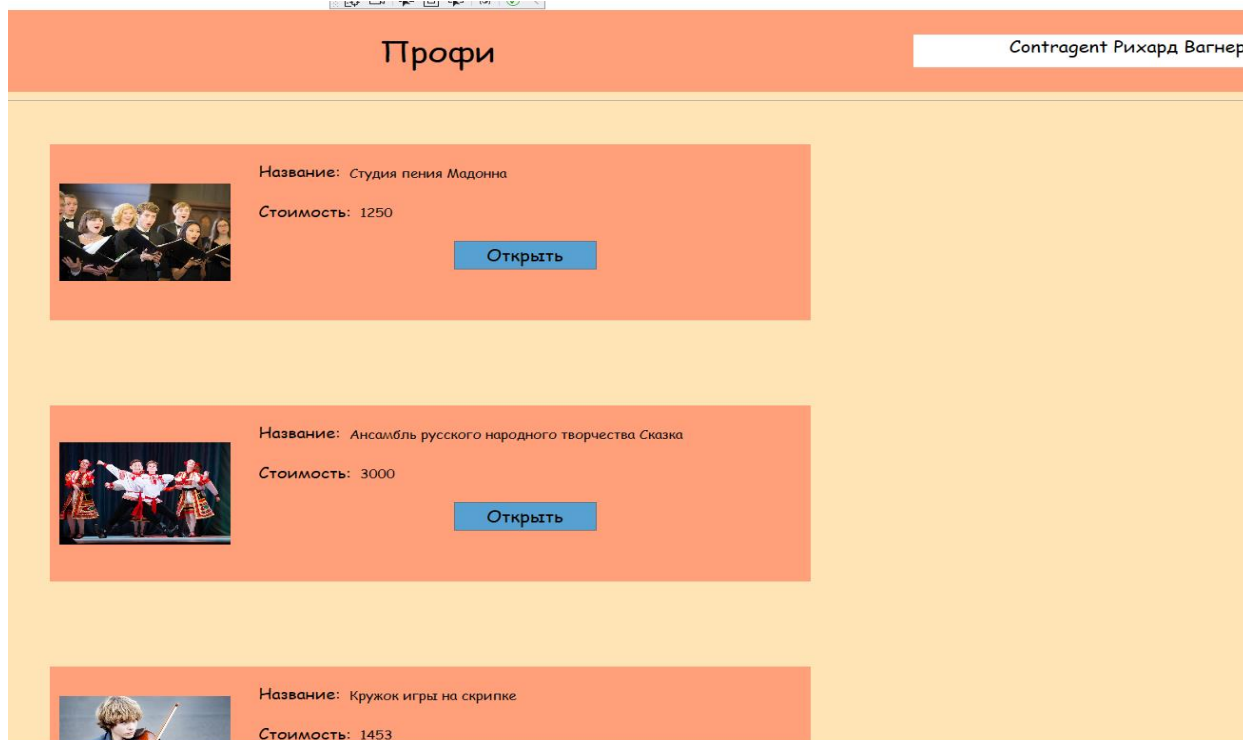


Рисунок 14 – Страница кабинета контрагента с его СДО

Таким образом был разработан прототип интерфейса для приложения, используемого в основных формах.

3 Реализация

Программный код представлен в приложении В.

3.1 Обоснование выбора средств разработки

В данном подразделе мы рассмотрим основные критерии выбора среды для разработки системы, а также системы для реализации базы данных.

Система состоит из следующих компонентов:

- Программный код: WPF;
- СУБД: SSMS
- среда разработки: Visual Studio 2022 Community Edition.

В качестве основной технологии была выбран WPF, обеспечивающая производительность, гибкость и возможность обработки больших объемов данных. WPF также удобна в работе при создании сложных пользовательских интерфейсов и архитектур, таких как приложения для настольных компьютеров.

IDE Visual Studio 2022 Community Edition используется для разработки приложения, что обеспечивает удобство в работе, обширный выбор полезных функций и настройку конфигураций для различных целей.

СУБД SQL Server Management Studio была выбрана за простоту освоения функционала и использование на большинстве компьютеров заказчика.

В общем, выбранные средства разработки обеспечивают необходимую надежность, производительность, и удобство в разработке, что делает их оптимальным выбором.

3.2 Разработка базы данных в среде СУБД

На основе ER-диаграммы на рисунке 4 в SSMS была разработана база данных на рисунке 15, в таблице 11 описаны таблицы БД.

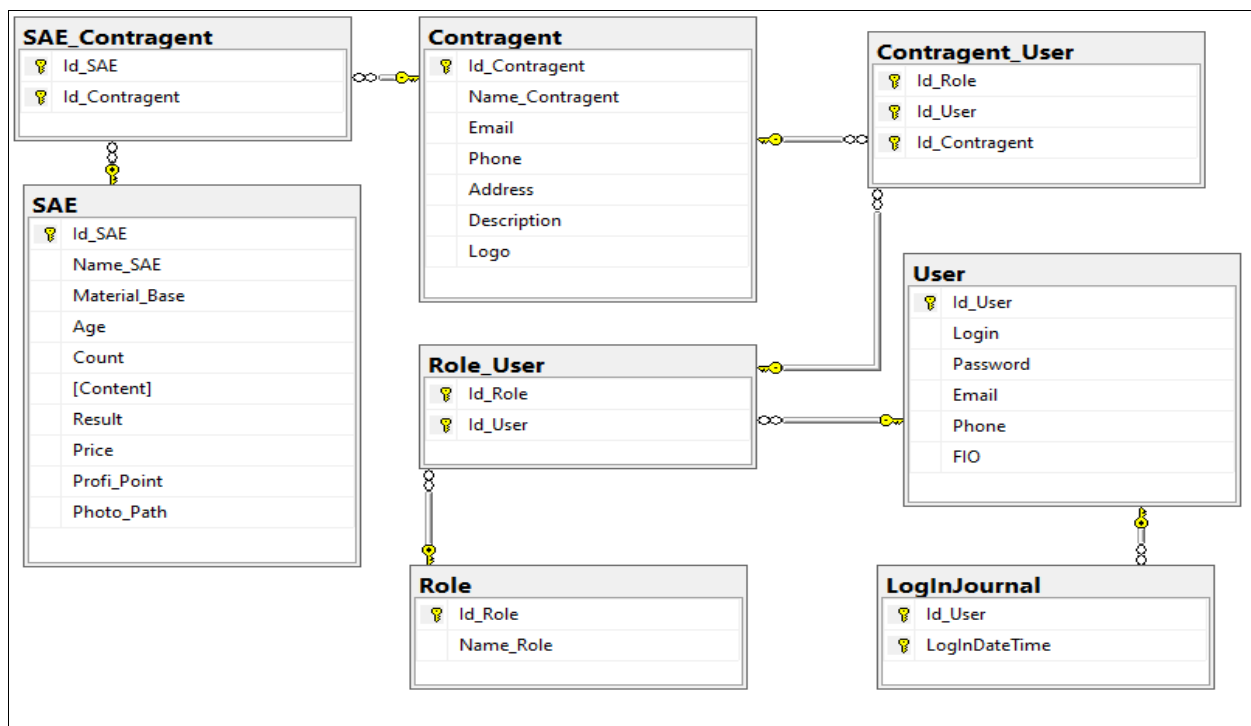


Рисунок 15 – БД системы «Профи»

Таблица 11 – Краткое описание таблиц

Название	Краткое описание
Role	Данные о ролях в системе
User	Данные о пользователях в системе
Role_User	Служебная таблица для связи пользователей и их ролей
LogInJournal	Данные о входе пользователей в систему
SAE	Данные о СДО
Contragent	Данные о контрагентах
SAE_Contragent	Служебная таблица для связи СДО и контрагентов
Contragent_Upser	Служебная таблица для связи контрагентов и пользователей с ролью контрагент

Таким образом была разработана БД на основе ER-диаграммы.

3.3 Описание программных модулей

Структура проекта (обозреватель решения) представлена на рисунке 16, описание папок и файлов представлены в таблице 12.

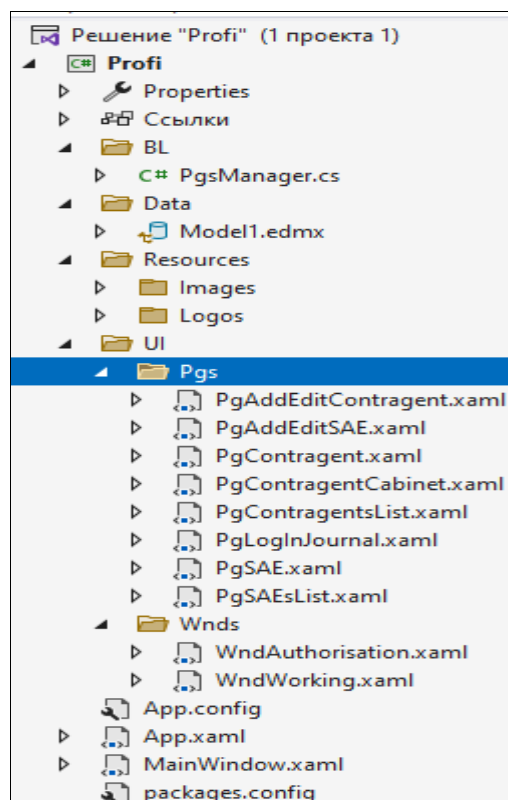


Рисунок 16 – Обзорщик решения со всеми файлами

Таблица 12 – Описание структуры проекта

Модуль	Назначение
BL	Бизнес-логика, хранится класс для смены страниц внутри окна
Data	Модель для связи БД с приложением
Resources	Хранятся ресурсы приложения: Images – изображения СДО, иконки и служебные картинки, Logos – логотипы контрагентов
UI	Хранятся интерфейсы с кодом обработчиков: Wnds – окна и Pgs – страницы

Экранные формы уже были представлены на рисунках 6-14.

4 Тестирование

4.1 Модульное тестирование

При планировании стратегии модульного тестирования были разработаны следующие задачи:

- Функциональное тестирование для проверки соответствия системы требованиям;
- Тестирование функционала на обнаружение ошибок и необработанных исключений.

Для выполнения модульных тестов были разработаны unit-тесты в Visual Studio 2022. В решении был создан проект для модульного тестирования «ProfiTests», в котором впоследствии были разработаны тесты для проверки функциональности системы. На рисунках 17-19 представлены разработанные unit-тесты.

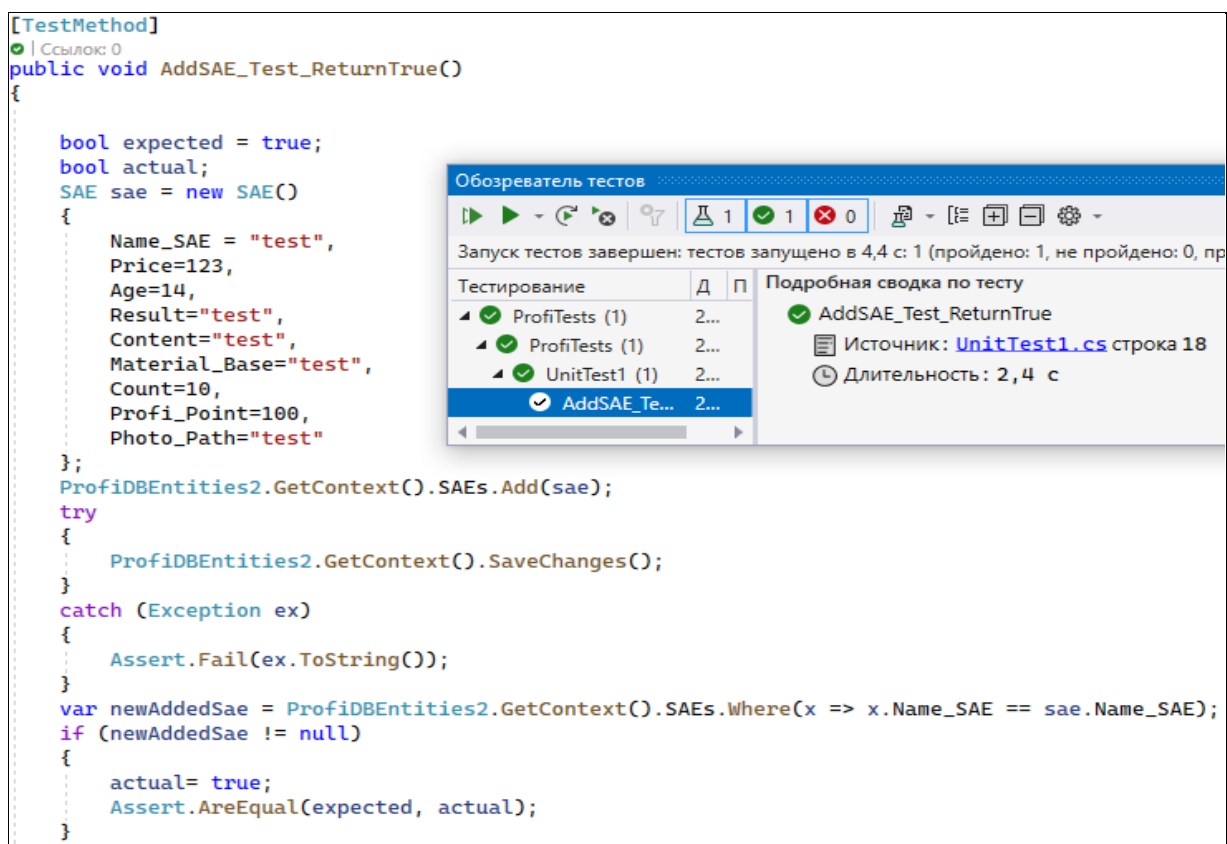


Рисунок 17 – Unit-тест добавления СДО

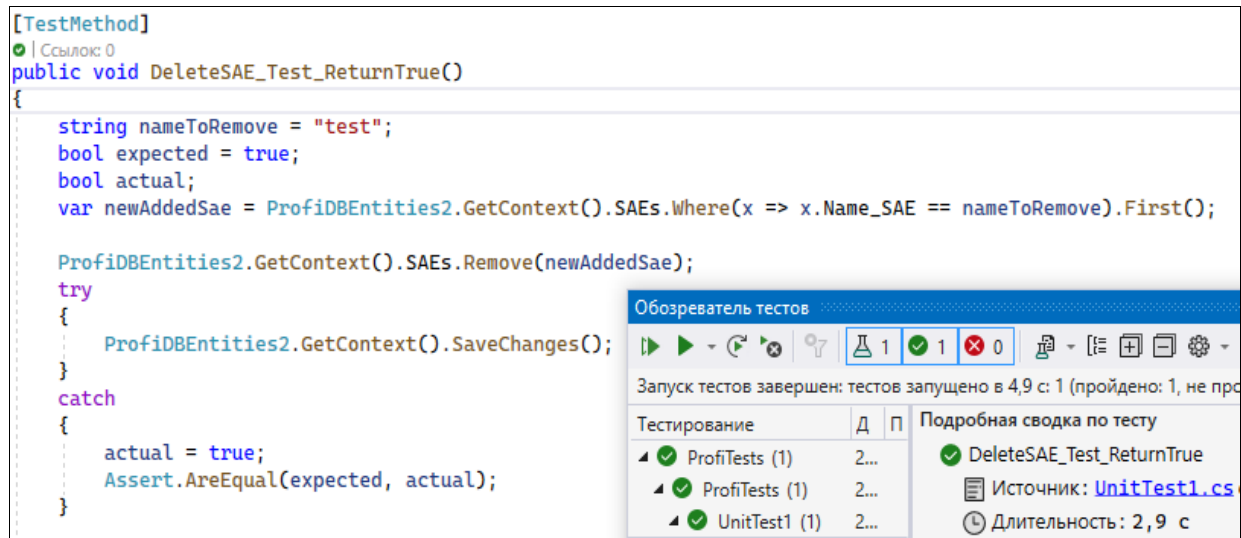


Рисунок 18 – Unit-тест удаления СДО

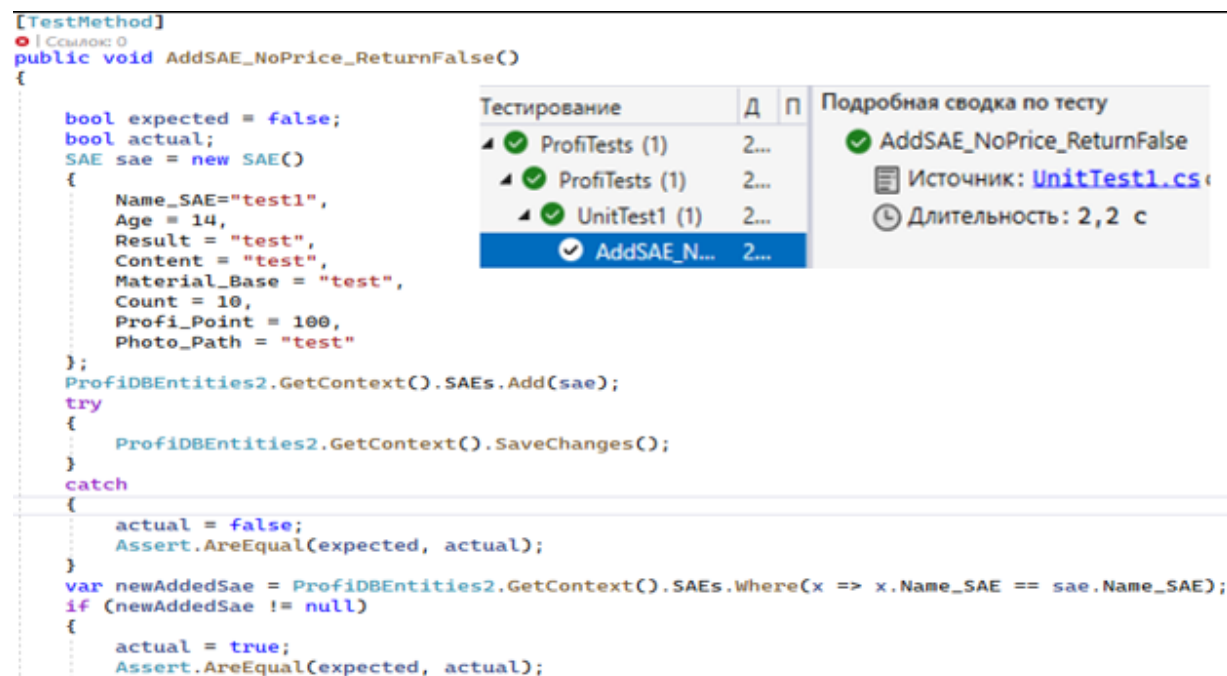


Рисунок 19 – Unit-тест отрицательный тест добавления СДО

Таким образом было проведено модульное тестирование, не выявившее ошибок и необработанных исключений в функционале.

4.2 Интеграционное тестирование

Интеграционное тестирование - это проверка совместной работы модулей и компонентов системы, проводимая с целью выявления проблем при взаимодействии. Этот тип тестирования проверяет работу различных частей системы в едином контексте, что позволяет выявить проблемы раньше, чем при системном тестировании. В ходе интеграционного тестирования проверяется совместимость разных элементов системы, обмен данными между ними, работа в целом. Тест-кейсы представлены в приложении Д. Общая информация о тестировании представлена в таблице 13.

Таблица 13 – Общая информация о тестировании

Название проекта	Profi
Номер версии	1.0.0
Имя тестировщика	Лундин Илья
Даты тестирования	26.06.2023 - 30.06.2023

В таблицах Д.1 и Д.2 представлены тест-кейсы №1 и №2 для проверки работы модуля добавления СДО, результат тестирования тест-кейса №1 и №2: «Пройдено».

В таблице Д.3 представлен тест-кейс №3 для проверки функционала удаления СДО, результат тестирования тест-кейса №3: «Пройдено».

Заключение

В результате проделанной работы в рамках курсового проекта была разработана согласно требованиям ГОСТ 19.102-77 «Единая система программной документации (ЕСПД). Стадии разработки». При разработке программной системы были выполнены следующие задачи:

1. Техническое задание:

- Разработаны функциональные требования к программной системе;
- Определена основная бизнес-логика приложения
- Спроектированы базовые функции для каждой из требуемых функциональных областей.

2. Технический проект:

- Созданы диаграммы классов и базы данных;
- Определена архитектура программной системы;
- Утверждена структура базы данных;
- Согласованы интерфейсы системы.

3. Рабочий проект:

- Разработаны и протестированы все модули приложения;
- Созданы механизмы валидации входных данных, обеспечивающие корректность работы приложения.

4. Внедрение:

- Установлена и настроена база данных приложения;
- Произведена настройка конфигурации программной системы на сервере;
- Произведено тестирование функциональности системы.

Таким образом, программная система «Профи» была разработана в соответствии с требованиями заказчика. Была определена основная бизнес-логика, спроектирована архитектура и созданы механизмы валидации данных. Система является важным инструментом для людей, нуждающихся в автоматизации процессов добавления, редактирования и поиска СДО.

Список использованных источников

1. ГОСТ 2.105 – 2019. Единая система конструкторской документации (ЕСКД). Общие требования к текстовым документам (Издание с Изменением N 1) = Unified system for design documentation. General requirements for textual documents: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 29 апреля 2019 г. № 175-ст: введен впервые: дата введения 2021-02-01 / Разработан Федеральным государственным унитарным предприятием «Российский научно-технический центр информации по стандартизации, метрологии и оценке соответствия» (ФГУП «СТАНДАРТИНФОРМ»). – Москва.: Стандартинформ, 2021. – 35 с. – Текст непосредственный.

2. ГОСТ 7.0.100 – 2018. Система стандартов по информации, библиотечному и издательскому делу (СИБИД). Библиографическая запись. Библиографическое описание. Общие требования и правила составления (с Поправкой) = System of standards on information, librarianship and publishing. Bibliographic record. Bibliographic description. General requirements and rules: национальный стандарт Российской Федерации: издание официальное: утвержден и введен в действие Приказом Федерального агентства по техническому регулированию и метрологии от 3 декабря 2018 года № 1050-ст: введен впервые: дата введения 2019-07-01 / Разработан Федеральным государственным унитарным предприятием "Информационное телеграфное агентство России (ИТАР-ТАСС)", филиал "Российская книжная палата", Федеральным государственным бюджетным учреждением "Российская государственная библиотека", Федеральным государственным бюджетным учреждением "Российская национальная библиотека". – Москва: Стандартинформ, 2018. – 128 с. – Текст непосредственный.

3. ГОСТ 19.201-78. Единая система программной документации (ЕСПД). Техническое задание. Требования к содержанию и оформлению (с Изменением N 1) = Unified system for program documentation. Technical specifications for development. Requirements to contents and form of presentation: межгосударственный стандарт: издание официальное: утвержден и введен в действие Постановлением Государственного комитета СССР по стандартам от 18 декабря 1978 г. № 3351: введен впервые: дата введения 1980-01-01. – Москва: Стандартинформ, 2010. – 4 с. – Текст непосредственный.

4. РД 50-34.698-90. Методические указания. Информационная технология. Комплекс стандартов и руководящих документов на автоматизированные системы. Автоматизированные системы. Требования к содержанию документов: руководящий документ по стандартизации: издание официальное: утверждены и введены в действие Постановлением Государственного комитета СССР по управлению качеством продукции и стандартами от 27 декабря 1990 г. № 3380: дата введения 1992-01-01 / Разработан Министерством электротехнической промышленности и приборостроения СССР. – Москва.: ИПК Издательство стандартов, 2002 г. – 27 с. –Текст непосредственный.

5. Перлова, О.Н. Соадминистрирование баз данных и серверов: учебник для студ. учреждений среднего профессионального образования / О.Н. Перлова, О.П. Ляпина. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-7075-2. – Текст: непосредственный.

6. Семакин, И.Г. Основы алгоритмизации и программирования: учебник для студ. учреждений среднего профессионального образования / И.Г. Семакин, А.П. Шестаков. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-6228-3. – Текст: непосредственный.

7. Федорова, Г.Н. Разработка модулей программного обеспечения компьютерных систем: учебник для студ. учреждений среднего

профессионального образования / Г.Н. Федорова. – М.: Издательский центр «Академия», 2016. – 336 с. – (Профессиональное образование). – ISBN 978-5-4468-1585-2. – Текст: непосредственный.

8. Федорова, Г.Н. Участие в интеграции программных модулей: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – М.: Издательский центр «Академия», 2016. – 304 с. – (Профессиональное образование). – ISBN 978-5-4468-2374-1. – Текст: непосредственный.

9. METANIT.COM. Сайт о программировании. Полное руководство по языку программирования C# 11 и платформе .NET 7: [сайт] – Текст: электронный. – URL: <https://metanit.com/sharp> (дата обращения 22.06.2023);

10. Microsoft Learn. Техническая документация. Документация по ASP.NET: [сайт] – Текст: электронный. – URL: <https://learn.microsoft.com/ru-ru/aspnet/core/introduction-to-aspnet-core?view=aspnetcore-7.0> (дата обращения 22.06.2023);

11. Тестирование ПО. Ulearn.me: [сайт] – Текст. Видеоматериалы: электронные. – URL https://ulearn.me/Course/Testing/Vvedenie_5656d8a3-1269-4834-bdfe-8fbc1f1c8f30 (дата обращения 22.06.2023);

12. Федорова, Г.Н. Разработка, администрирование и защита баз данных: учебник для студ. учреждений среднего профессионального образования / Г.Н. Федорова. – 2-е изд., стер. – М.: Издательский центр «Академия», 2018. – 288 с. – (Профессиональное образование). – ISBN 978-5-4468-6994-7. – Текст: непосредственный.

Приложение А

Техническое задание. Требования к программным модулям

1 Введение

1.1 Наименование программы

Наименование программной системы «Профи».

1.2 Краткая характеристика области применения

Программная система «Профи» предназначена для автоматизации добавления, изменения и удаления, а также поиска секций дополнительного образования для занятий. С приложением будут работать специалисты по дополнительному образованию, для которых создана роль администратор, ведущий работу с контрагентами, секций дополнительного образования, в дальнейшем именуемые СДО, и представитель организации по ведению СДО, для которого предусмотрена в системе роль контрагент, ведущий работу с СДО.

					<i>ККЭП 09.02.07 023_11 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		30

2 Основания для разработки

Основанием для разработки является Договор 1917 от 21.03.2023. Договор 1917 утвержден Директором ООО «Профи» Лундиным Алексеем Владимировичем, именуемым в дальнейшем Заказчиком, и Лундиным Ильёй Алексеевичем (само занятый), именуемым в дальнейшем Исполнителем, 21.03.2023.

Согласно договору, исполнитель обязан разработать и установить систему «Профи» на оборудовании заказчика не позднее 13.06.2023, предоставить исходные коды и документацию к разработанной системе не позднее 28.06.2023.

Наименование темы разработки – «Разработка модулей подсистем «Администратор» и «Личный кабинет контрагента» информационной системы «Профи».

					<i>ККЭП 09.02.07 023_11 ПЗ</i>	Лист
Изм.	Лист	№ докум.	Подпись	Дата		31

3 Назначение разработки

3.1 Функциональное назначение

Программа должна обеспечивать выполнение следующих функций:

- разграничение доступа;
- работа с историей входа;
- работа с контрагентами;
- работа с СДО;
- поиск и фильтрация;
- работа с печатными формами и экспорт в pdf.

3.2 Эксплуатационное назначение

Программная система должна эксплуатироваться на всех рабочих местах пользователей, представляющих собой персональные компьютеры и на отдельном рабочем месте, который будет использоваться как сервер.

4 Требования к программе и программному изделию

4.1 Требования к функциональным характеристикам

4.1.1 Требования к составу выполняемых функций

Функция разграничения доступа включает в себя вход в приложение через логин и пароль, а также разделение пользователей на роли администратора и контрагента.

Функция работы с историей входа предполагает создание журнала, в котором администратор сможет увидеть дату, время, логин каждого вошедшего пользователя.

Функция работы с контрагентами включает добавление, изменение и удаление контрагента и следующих сведений о них: наименование, электронная почта, телефон, адрес, описание. Все вышеперечисленные задачи в системе будут выполняться администратором.

Функция работы с СДО подразумевает добавление, изменение и удаление СДО и данных о них: наименование, возраст обучающихся, количество обучающихся, расписание, содержание программы, результат обучения, материально-техническая база, стоимость обучения, данные задачи находятся в ведении контрагента.

Функция поиска и фильтрации означает поиск СДО по их названию, а также их фильтрацию по стоимости и профи-баллам. Поиск и фильтрацию всех вышеперечисленных объектов по их свойствам могут осуществлять как администратор, так и контрагент.

Функция работы с печатными формами и экспорт в pdf предполагает, что как администратор, так и контрагент могут выбрать СДО, который можно экспортировать в pdf или выбрать принтер для печати.

4.1.2 Требования к организации входных и выходных данных

Входные данные, которые будут вставлены в БД, представлены ниже:

Сведения для добавления контрагента:

- название (nvarchar(100));
- электронная почта (nvarchar(20));
- телефон (nvarchar(10));
- адрес (nvarchar(200));
- описание (nvarchar(500));

Сведения для добавления СДО:

- название (nvarchar(100));
- материально-техническая база (описание помещения и иных предметов, необходимых при обучении, nvarchar(500));
- возраст (int);
- количество обучающихся (int);
- содержание программы (nvarchar(500)) ;
- результат обучения (nvarchar(200)) ;
- стоимость обучения (int).

Выходные данные:

Печатная форма, представляющая из себя заполненную страницу конкретного СДО.

4.1.3 Требования к временным характеристикам

После изменения, добавления или удаления любых данных, находящихся в базе данных, обновленная информация отображается не позднее, чем через пять секунд.

4.2 Требования к надежности и безопасности

Для обеспечения безопасности необходимо реализовать вход в систему по логину и паролю, более того требуется разграничить доступ к функционалу, согласно выделенным ролям в системе.

4.3 Условия эксплуатации

Программа запускается на всех компьютерах компании. База данных находится на отдельной станции. Должна существовать устойчивая связь по сети между клиентами и базой данных.

4.3.1 Требования к составу и параметрам технических средств

Состав технических средств предусматривает наличие компьютеров – рабочих станций и компьютер-сервер. Компьютер-рабочая станция (для администратора) включает:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб;
- видеокарту, монитор, мышь, клавиатура.

Для компьютера – сервера (для функционирования СУБД) (основной и резервный), включают в себя:

- процессор x86 с тактовой частотой, не менее 1 ГГц;
- оперативную память объемом, не менее 1 Гб; видеокарту, монитор, мышь.

4.4 Требования к информационной и программной совместимости

На серверах должны быть установлены ОС Windows Server 2019, сервер БД Microsoft SQL Server Express 2019, SQL Server Management Studio 18.8.

Приложение В

Руководство по стилю

Общие требования:

- в приложении не допускаются орфографические и грамматические ошибки;
- все экранные формы пользовательского интерфейса имеют заголовок с логотипом. Цвет, размер и пропорции логотипа не должны изменяться при использовании программы. Так же для приложения должна быть установлена иконка, соответствующая логотипу;
- тип шрифта – «Comic Sans MS».

Цветовая схема представлена в таблице В.1.

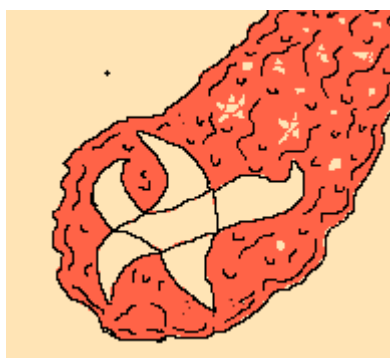


Рисунок 20 – Иконка приложения

Таблица В.1 – Цветовая схема приложения

Основной фон	Дополнительный фон	Акцентирование внимания
hex(#ffe4b5) rgb(255, 228, 181)	hex(#ffa07a) rgb(255, 160, 122)	hex(#57a0d2) rgb(87, 160, 210)

Программный код

В ходе выполнения работы были написаны код верстки и код бизнес-логики на рисунках 21-34.

```
<Style x:Key="GridMain" TargetType="Grid">
    <Setter Property="Background" Value="#ffe4b5"></Setter>
</Style>
<Style x:Key="BorderMain" TargetType="TextBlock">
    <Setter Property="Background" Value="#ffa07a"></Setter>
    <Setter Property="FontFamily" Value="Comic Sans MS"></Setter>
    <Setter Property="FontSize" Value="20"></Setter>
</Style>
<Style x:Key="ButtonMain" TargetType="Button">
    <Setter Property="Height" Value="35"></Setter>
    <Setter Property="Width" Value="150"></Setter>
    <Setter Property="Background" Value="#57a0d2"></Setter>
    <Setter Property="FontFamily" Value="Comic Sans MS"></Setter>
    <Setter Property="FontSize" Value="20"></Setter>
    <Setter Property="Margin" Value="5"></Setter>
</Style>
<Style x:Key="TextBlockMain" TargetType="TextBlock">
    <Setter Property="FontFamily" Value="Comic Sans MS"></Setter>
    <Setter Property="FontSize" Value="18"></Setter>
    <Setter Property="TextAlignment" Value="Center"></Setter>
    <Setter Property="VerticalAlignment" Value="Center"></Setter>
    <Setter Property="HorizontalAlignment" Value="Center"></Setter>
</Style>
<Style x:Key="TextBoxMain" TargetType="TextBox">
    <Setter Property="FontFamily" Value="Comic Sans MS"></Setter>
    <Setter Property="FontSize" Value="14"></Setter>
    <Setter Property="TextAlignment" Value="Left"></Setter>
    <Setter Property="Height" Value="30"></Setter>
    <Setter Property="Margin" Value="10"></Setter>
</Style>
<Style x:Key="ComboBoxMain" TargetType="ComboBox">
    <Setter Property="FontFamily" Value="Comic Sans MS"></Setter>
    <Setter Property="FontSize" Value="20"></Setter>
    <Setter Property="Height" Value="40"></Setter>
</Style>
<Style x:Key="TextBlockAutho" TargetType="TextBlock" BasedOn="{StaticResource TextBlockMain}">
    <Setter Property="FontSize" Value="20"></Setter>
    <Setter Property="HorizontalAlignment" Value="Left"></Setter>
    <Setter Property="Margin" Value="20"></Setter>
</Style>
```

Рисунок 21 – Стили кнопок и блоков текста в App.xaml

```

<Style x:Key="TextBoxAuth" TargetType="TextBox" BasedOn="{StaticResource TextBoxMain}">
    <Setter Property="Height" Value="30" />
    <Setter Property="Width" Value="150" />
    <Setter Property="HorizontalAlignment" Value="Left" />
    <Setter Property="Margin" Value="10" />
    <Setter Property="VerticalAlignment" Value="Bottom" />
</Style>
<Style x:Key="TextBlockLabel" TargetType="TextBlock" BasedOn="{StaticResource TextBlockMain}">
    <Setter Property="Height" Value="40" />
    <Setter Property="TextAlignment" Value="Right" />
    <Setter Property="Background" Value="#ffa07a" />
    <Setter Property="Width" Value="400" />
    <Setter Property="HorizontalAlignment" Value="Right" />
    <Setter Property="FontFamily" Value="Comic Sans MS" />
    <Setter Property="FontSize" Value="21" />
    <Setter Property="Margin" Value="10" />
</Style>
<Style x:Key="TextBoxLabel" TargetType="TextBox">
    <Setter Property="Height" Value="40" />
    <Setter Property="Width" Value="410" />
    <Setter Property="Margin" Value="10" />
    <Setter Property="HorizontalAlignment" Value="Right" />
    <Setter Property="FontFamily" Value="Comic Sans MS" />
    <Setter Property="FontSize" Value="20" />
</Style>
<Style x:Key="PasswordBoxMain" TargetType="PasswordBox">
    <Setter Property="FontFamily" Value="Comic Sans MS" />
    <Setter Property="FontSize" Value="14" />
    <Setter Property="Height" Value="30" />
    <Setter Property="Width" Value="150" />
    <Setter Property="HorizontalAlignment" Value="Left" />
    <Setter Property="Margin" Value="10" />
    <Setter Property="VerticalAlignment" Value="Bottom" />
</Style>

```

Рисунок 22 – Стили блока ввода пароля и ввода текста в App.xaml

```

Title="Профи - Окно входа" Height="550" Width="310" ResizeMode="NoResize" Topmost="True" WindowStartupLocation="CenterScreen">
<Grid Style="{StaticResource GridMain}">
    <Grid.RowDefinitions>
        <RowDefinition Height="100" />
        <RowDefinition Height="150" />
        <RowDefinition Height="150" />
        <RowDefinition />
        <RowDefinition Height="50" />
    </Grid.RowDefinitions>
    <TextBlock Height="100" Style="{StaticResource BorderMain}" TextAlignment="Center"
        Padding="25" Text="" />
    <TextBlock Style="{StaticResource TextBlockAuth}" VerticalAlignment="Top"
        Text="Вход в систему" FontSize="25" Margin="95,21,0,0" />
    <Image Height="70" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10" Source="/Resources/Images/SuperStar.png" />
    <TextBlock Style="{StaticResource TextBlockAuth}" VerticalAlignment="Top" Text="Логин" Grid.Row="1" Margin="35,21,0,0" />
    <TextBox Padding="3" Style="{StaticResource TextBoxAuth}" Grid.Row="1" Name="TbxLogin" TextChanged="TbxLogin_TextChanged"
        Margin="20,0,0,34" />
    <TextBlock Style="{StaticResource TextBlockAuth}" Visibility="Hidden" VerticalAlignment="Top" Text="Пароль" Grid.Row="2" Name="TbxPassword"
        RenderTransformOrigin="0.828,0.468" Margin="35,27,0,0" />
    <TextBox Padding="3" Style="{StaticResource TextBoxAuth}" Grid.Row="2" Name="TbxPassword" TextChanged="TbxPassword_TextChanged"
        Visibility="Hidden" Margin="20,0,0,35" />
    <PasswordBox Style="{StaticResource PasswordBoxMain}" Grid.Row="2" Name="PasswordBox" PasswordChanged="PasswordBox_PasswordChanged"
        Visibility="Hidden" Margin="20,0,0,35" />
    <CheckBox Padding="5" Grid.Row="2" VerticalAlignment="Bottom" HorizontalAlignment="Right" Margin="20" Height="25"
        Width="25" Name="OpenPassword" Checked="OpenPassword_Checked" Unchecked="OpenPassword_Unchecked" Visibility="Hidden"
        ToolTip="Показать пароль" IsChecked="False" />
    <Button Style="{StaticResource ButtonMain}" Grid.Row="3" Name="BtnLogIn" Content="Войти" Click="BtnLogIn_Click"
        Visibility="Hidden" HorizontalAlignment="Center" />
    <TextBlock Grid.Row="4" Style="{StaticResource BorderMain}" />
</Grid>

```

Рисунок 23 – Верстка окна входа

```

private void OpenPassword_Checked(object sender, RoutedEventArgs e)
{
    TbxPassword.Visibility = Visibility.Visible;
    PasswordBox.Visibility = Visibility.Hidden;
}
Ссылка: 1
private void OpenPassword_Unchecked(object sender, RoutedEventArgs e)
{
    TbxPassword.Visibility = Visibility.Hidden;
    PasswordBox.Visibility = Visibility.Visible;
}
public static string login;
public static string password;
public static string fio;
public static string roleName;
Ссылка: 1
private void BtnLogin_Click(object sender, RoutedEventArgs e)
{
    var userObj = ProfiDBEntities1.GetContext().Users.FirstOrDefault(x => x.Login == TbxLogin.Text);
    if (userObj == null)
    {
        MessageBox.Show("Такого пользователя нет в системе!", "Ошибка входа в систему", MessageBoxButton.OK, MessageBoxImage.Error);
        TbxLogin.Text = "";
        TbxPassword.Text = "";
    }
}

```

Рисунок 24 – Логика определения наличия пользователя в системе

```

else
{
    password = userObj.Password;
    login = userObj.Login;
    fio = userObj.FIO;
    if (password == TbxPassword.Text)
    {
        MessageBox.Show($"Успешный вход в систему! {fio}", "Информация", MessageBoxButton.OK, MessageBoxImage.Information);
        var role = ProfiDBEntities1.GetContext().UserRoles.Where(x => x.Login == TbxLogin.Text).FirstOrDefault();
        roleName = role.Name_Role;
        WndWorking mw = new WndWorking();
        mw.Show();
        //Блок записи в журнал входа
        LogInJournal _currentLogIn = new LogInJournal();
        _currentLogIn.Id_User = userObj.Id_User;
        _currentLogIn.LogInDateTime = DateTime.Now;
        ProfiDBEntities1.GetContext().ChangeTracker.Entries().ToList().Clear();
        ProfiDBEntities1.GetContext().LogInJournals.Add(_currentLogIn);
        ProfiDBEntities1.GetContext().SaveChanges();
        this.Close();
    }
    else
    {
        MessageBox.Show("Неверный пароль!", "Ошибка входа в систему", MessageBoxButton.OK, MessageBoxImage.Error);
        TbxPassword.Text = "";
    }
}
}

```

Рисунок 25 – Логика входа и записи его в журнал

```

Ссылка: 1
private void TbxLogin_TextChanged(object sender, TextChangedEventArgs e)
{
    if (TbxLogin.Text != "")
    {
        TblPassword.Visibility = Visibility.Visible;
        PasswordBox.Visibility = Visibility.Visible;
    }
    else
    {
        TblPassword.Visibility = Visibility.Hidden;
        PasswordBox.Visibility = Visibility.Hidden;
        PasswordBox.Password = "";
        TbxPassword.Text = "";
    }
}

Ссылка: 1
private void TbxPassword_TextChanged(object sender, TextChangedEventArgs e)
{
    if (PasswordBox.Password != "")
    {
        OpenPassword.Visibility = Visibility.Visible;
        BtnLogIn.Visibility = Visibility.Visible;
        PasswordBox.Password = TbxPassword.Text;
    }
    else
    {
        OpenPassword.Visibility = Visibility.Hidden;
        BtnLogIn.Visibility = Visibility.Hidden;
        PasswordBox.Password = TbxPassword.Text;
    }
}

```

Рисунок 26 – Логика показа пароля

```

<Grid Grid.Row="0" Style="{StaticResource GridMain}" Background="#ffa07a" Grid.ColumnSpan="2">
    <Image HorizontalAlignment="Left" Margin="10" Source="/Resources/Images/SuperStar.png"/>
    <TextBlock Text="Профи" FontSize="35" Style="{StaticResource TextBlockMain}"/>
</Grid>
<TextBlock FontFamily="Comic Sans MS" FontSize="15" Grid.Row="3" Grid.ColumnSpan="2" TextAlign="Center">
<Button Name="BtnExit" Grid.Row="3" Content="Выйти" Style="{StaticResource ButtonMain}" Click="BtnExit_Click" />
<TextBlock Name="UserName" Width="450" Height="40" Background="White" Grid.Column="0" Grid.Row="4" Grid.ColumnSpan="2">
    <TextBlock FontFamily="Comic Sans MS" FontSize="20" TextAlignment="Center"/>
</TextBlock>
<Frame Grid.Row="1" Name="MainFrame" ContentRendered="MainFrame_ContentRendered" NavigationUIShown="MainFrame_NavigationUIShown">
<TextBlock Grid.Row="1" Grid.ColumnSpan="2" Background="#ffe4b5" Panel.ZIndex="-1"></TextBlock>
<TextBlock Grid.Row="2" Grid.ColumnSpan="2" Background="#ffe4b5" Panel.ZIndex="-1"></TextBlock>
<Button Content="Назад" Style="{StaticResource ButtonMain}" HorizontalAlignment="Center" Name="BtnBack" />
<Button Content="Журнал" Style="{StaticResource ButtonMain}" HorizontalAlignment="Right" Name="BtnJournal" />
<Button Content="Кабинет" Style="{StaticResource ButtonMain}" HorizontalAlignment="Right" Name="BtnCabinet" />

```

Рисунок 27 – Верстка рабочего окна


```

UserName.Text = WndAuthorisation.roleName + " " + WndAuthorisation.fio;

if (WndAuthorisation.roleName == "Admin")
{
    MainFrame.Navigate(new PgSAEsList());
    PgsManager.MainFrame = MainFrame;
    BtnJournal.Visibility = Visibility.Visible;
}
if (WndAuthorisation.roleName == "Contragent")
{
    BtnCabinet.Visibility = Visibility.Visible;
    var currentContragent = ProfiDBEntities1.GetContext().All_Contragents_User.Where(x => x.Login == WndAuthorisation.login).FirstOrDefault();
    PgsManager.MainFrame = MainFrame;
    MainFrame.Navigate(new PgContragentCabinet(currentContragent));
}

```

Рисунок 28 – Отображение и получение страниц в зависимости от роли

```

private void BtnExit_Click(object sender, RoutedEventArgs e)
{
    WndAuthorisation wA = new WndAuthorisation();
    wA.Show();
    this.Close();
}
Ссылка: 1
private void MainFrame_ContentRendered(object sender, EventArgs e)
{
    if (MainFrame.CanGoBack)
    {
        BtnBack.Visibility = Visibility.Visible;
    }
    else
    {
        BtnBack.Visibility = Visibility.Hidden;
    }
}
Ссылка: 1
private void BtnBack_Click(object sender, RoutedEventArgs e)
{
    MainFrame.GoBack();
}
Ссылка: 1
private void BtnJournal_Click(object sender, RoutedEventArgs e)
{
    MainFrame.Navigate(new PgLogInJournal());
}
Ссылка: 1
private void BtnCabinet_Click(object sender, RoutedEventArgs e)
{
    var currentContragent = ProfiDBEntities1.GetContext().All_Contragent
    PgsManager.MainFrame = MainFrame;
    MainFrame.Navigate(new PgContragentCabinet(currentContragent));
}

```

Рисунок 29 – Логика навигации по страницам


```

<ScrollView>
  <Canvas Name="PdfCanvas" Width="1400" Height="1200">
    <StackPanel Background="#ffe4b5" Height="1200" Width="1750">
      <StackPanel Background="#ffa07a" HorizontalAlignment="Center" Margin="0,50,350,0">
        <Grid Background="#ffa07a" Height="210" Width="800">
          <Image Margin="10" Height="200" Width="200">
            <Image.Source>
              <Binding Path="Photo_Path">
                <Binding.TargetNullValue>
                  <ImageSource>
                    /Resources/Images/nophoto.png
                  </ImageSource>
                </Binding.TargetNullValue>
              </Binding>
            </Image.Source>
          </Image>
        </Grid>
      </StackPanel>
    </Canvas>
  </ScrollView>

```

Рисунок 30 – Верстка начала страницы СДО

```

<StackPanel Orientation="Horizontal" >
  <TextBlock Style="{StaticResource TextBlockLabel}" Text="Название СДО: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Name="TblNameSAE" Text="{Binding Name_SAE}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal" >
  <TextBlock Style="{StaticResource TextBlockLabel}" Text="Название контрагента: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Text="{Binding Name_Contragent}" Name="TblNameContr"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal" >
  <TextBlock Style="{StaticResource TextBlockLabel}" Padding="0,40,0,0" Height="120" Text="Материально-техническая база: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Height="120" Name="TblMaterialBase" Text="{Binding MaterialBase}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBlock Style="{StaticResource TextBlockLabel}" Text="Возраст учащихся: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Name="TblAge" Text="{Binding Age}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBlock Style="{StaticResource TextBlockLabel}" Text="Количество учащихся в группе: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Name="TblCount" Text="{Binding Count}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBlock Style="{StaticResource TextBlockLabel}" Padding="0,40,0,0" Height="120" Text="Описание программы: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Height="120" Name="TblContent" Text="{Binding Content}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBlock Style="{StaticResource TextBlockLabel}" Padding="0,40,0,0" Height="120" Text="Цели программы: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Height="120" Name="TblResult" Text="{Binding Result}"></TextBlock>
</StackPanel>
<StackPanel Orientation="Horizontal">
  <TextBlock Style="{StaticResource TextBlockLabel}" Text="Цена: "></TextBlock>
  <TextBlock Style="{StaticResource TextBlockLabel}" Width="500" FontSize="18" TextAlignment="Left" Name="TblPrice" Text="{Binding Price}"></TextBlock>
</StackPanel>

```

Рисунок 31 – Верстка полей СДО

```

internal Data.SAE _currentSAE = new Data.SAE();
Ссылка 2
public PgSAE(Data.SAE selectedSAE)
{
    InitializeComponent();
    if (selectedSAE != null)
        _currentSAE = selectedSAE;
    DataContext = _currentSAE;
    var name = ProfiDBEntities1.GetContext().ContragentSAEs.Where(x => x.Name_SAE == _currentSAE.Name_SAE).FirstOrDefault();
    TblNameContragent.Text = name.Name_Contragent;
}
Ссылка 1
private void BtnEditSAE_Click(object sender, RoutedEventArgs e)
{
    PgsManager.MainFrame.Navigate(new PgAddEditSAE((sender as Button).DataContext as SAE));
}
Ссылка 1
private void BtnPdf_Click(object sender, RoutedEventArgs e)
{
    //Открытие диалога для сохранения файла в pdf и для печати
    PrintDialog pD = new PrintDialog();
    if (pD.ShowDialog() == true)
    {
        pD.PrintVisual(PdfCanvas, "Вывод");
    }
}

```

Рисунок 32 – Логика вывода печатной формы и загрузки данных о СДО

```

InitializeComponent();
if (selectedContragent != null)
    _currentContragent = selectedContragent;
DataContext = _currentContragent;
}
Ссылка 1
private void BtnSaveContragent_Click(object sender, RoutedEventArgs e)
{
    StringBuilder errors = new StringBuilder();
    if (string.IsNullOrEmpty(_currentContragent.Name_Contragent))
    {
        errors.AppendLine("Введите название контрагента");
    }
    if (string.IsNullOrEmpty(_currentContragent.Description))
    {
        errors.AppendLine("Введите описание");
    }
    if (string.IsNullOrEmpty(_currentContragent.Phone))
    {
        errors.AppendLine("Введите телефон");
    }
    if (string.IsNullOrEmpty(_currentContragent.Address))
    {
        errors.AppendLine("Введите адрес");
    }
    if (errors.Length > 0)
    {
        MessageBox.Show(errors.ToString());
        return;
    }
    if (_currentContragent.Id_Contragent == 0)
    {
        _currentContragent.Logo = "/Resources/Images/nophoto.png";
        ProfiDBEntities1.GetContext().Contragents.Add(_currentContragent);
    }
    try
    {
        ProfiDBEntities1.GetContext().SaveChanges();
        MessageBox.Show("Информация сохранена!");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}

```

Рисунок 33 – Логика проверки заполнения полей для добавления и редактирования контрагента

```

private void BtnSaveSAE_Click(object sender, RoutedEventArgs e)
{
    if (string.IsNullOrEmpty(_currentSAE.Name_SAE) || _currentSAE.Price < 1 || _currentSAE.Age < 0)
    {
        MessageBox.Show("Заполнены не все поля или некорректные данные!", "Ошибка");
        return;
    }
    _currentSAE.Profi_Point = 0;
    if (_currentSAE.Id_SAE == 0)
        ProfiDBEntities1.GetContext().SAEs.Add(_currentSAE);

    try
    {
        ProfiDBEntities1.GetContext().SaveChanges();
        MessageBox.Show("Информация сохранена!");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString());
    }
}

```

Рисунок 34 – Логика добавления и редактирования СДО

Приложение Г

SQL-скрипт базы данных

Скрипт БД представлен на рисунках 32-33.

```
CREATE TABLE [dbo].[Contragent](
    [Id_Contragent] [int] IDENTITY(1,1) NOT NULL,
    [Name_Contragent] [nvarchar](100) NOT NULL,
    [Email] [nvarchar](20) NOT NULL,
    [Phone] [nvarchar](10) NOT NULL,
    [Address] [nvarchar](200) NOT NULL,
    [Description] [nvarchar](500) NOT NULL,
    [Logo] [nvarchar](200) NOT NULL,
    PRIMARY KEY CLUSTERED
    (
        [Id_Contragent] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE
    ) ON [PRIMARY]
GO
/***** Object: Table [dbo].[SAE_Contragent]    Script Date
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[SAE_Contragent](
    [Id_SAE] [int] NOT NULL,
    [Id_Contragent] [int] NOT NULL,
    CONSTRAINT [PK_SC] PRIMARY KEY CLUSTERED
    (
        [Id_SAE] ASC,
        [Id_Contragent] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE
    ) ON [PRIMARY]
GO
/***** Object: View [dbo].[ContragentSAEs]    Script Date:
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE VIEW [dbo].[ContragentSAEs]
AS
SELECT Name_SAE, Name_Contragent, Material_Base, Age, Count,
FROM SAE s JOIN SAE_Contragent sc ON s.Id_SAE = sc.Id_SAE
JOIN Contragent c ON sc.Id_Contragent = c.Id_Contragent
GO
```

Рисунок 35 – Строки создания таблиц контрагент, СДО_контрагент и представления СДО в контрагентах

```

CREATE TABLE [dbo].[Role](
    [Id_Role] [int] IDENTITY(1,1) NOT NULL,
    [Name_Role] [nvarchar](20) NOT NULL,
    PRIMARY KEY CLUSTERED
(
    [Id_Role] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[User]    Script D
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[User](
    [Id_User] [int] IDENTITY(1,1) NOT NULL,
    [Login] [nvarchar](20) NOT NULL,
    [Password] [nvarchar](20) NOT NULL,
    [Email] [nvarchar](20) NOT NULL,
    [Phone] [nvarchar](10) NOT NULL,
    [FIO] [nvarchar](100) NULL,
    PRIMARY KEY CLUSTERED
(
    [Id_User] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE
) ON [PRIMARY]
GO
/***** Object: Table [dbo].[Role_User]    Scr
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Role_User](
    [Id_Role] [int] NOT NULL,
    [Id_User] [int] NOT NULL,
    CONSTRAINT [PK] PRIMARY KEY CLUSTERED
(
    [Id_User] ASC,
    [Id_Role] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE
) ON [PRIMARY]
GO

```

Рисунок 36 – Создание таблиц роль, пользователь, роль_пользователь

Приложение Д

Тест-кейсы

Тест- кейсы представлены в таблицах Д.1, Д.2 и Д.3.

Таблица Д.2 – Тест-кейс №1 ДобавлениеСДОПозитивное_1

Тест-кейс №1	ТК ДобавлениеСДОПозитивное_1
Приоритет теста	Высокий
Описание тестирования/Имя	Проверка работы функционала по добавлению СДО.
Резюме испытания	В результате тестирования необходимо удостовериться в корректной работе приложения при добавлении СДО с корректными данными.
Шаги тестирования	1. Вход в систему за контрагента. 2. Нажатие на кнопку «Добавить». 3. Добавление корректных данных в поля ввода. 4. Нажатие на кнопку «Сохранить».
Данные тестирования	Во все текстовые поля введено «test», во все числовые поля введено «120».
Ожидаемый результат	Система информирует пользователя о успешном сохранении информации.
Фактический результат	Система информирует пользователя о успешном сохранении информации.
Предпосылки	Авторизация в системе.
Постусловия	При вводе корректных данных приложение сохраняет изменения и отображает новый СДО на странице списка СДО.
Статус (Пройден/Провален)	Пройден.
Комментарии	Отсутствуют.

Таблица Д.2 – Тест-кейс №2 ДобавлениеСДОНегативное_1

Тест-кейс №2	ТК РедактированиеСДОНегативное_1
Приоритет теста	Высокий.
Описание тестирования/Имя	Проверка работы функционала по добавлению СДО.
Резюме испытания	В результате тестирования необходимо удостовериться в корректной работе приложения при добавлении СДО с некорректными данными.
Шаги тестирования	1. Вход в систему за контрагента. 2. Нажатие на кнопку «Добавить». 3. Добавление некорректных данных в поля ввода. 4. Нажатие на кнопку «Сохранить».
Данные тестирования	Во все поля введено «test».
Ожидаемый результат	Система информирует пользователя о том, что возраст учащихся, цена и количество учащихся в группе не введены.
Фактический результат	Система информирует пользователя о том, что возраст учащихся, цена и количество учащихся в группе не введены.
Предпосылки	Авторизация в системе за контрагента.
Постусловия	Отсутствуют.
Статус (Пройден/Провален)	Пройден.
Комментарии	Отсутствуют.

Таблица Д.3 – Тест-кейс №3 УдалениеСДО_1

Тест-кейс №2	ТК УдалениеСДО_1
Приоритет теста	Высокий.
Описание тестирования/Имя	Проверка работы функционала по удалению СДО.
Резюме испытания	В результате тестирования необходимо удостовериться в корректной работе приложения при удалении СДО с отсутствующими записями в других таблицах БД.
Шаги тестирования	1. Вход в систему за контрагента. 2. Нажатие на СДО. 3. Нажатие на кнопку «Удалить» 4. Нажатие на кнопку «ОК» в всплывающем сообщении.
Данные тестирования	СДО test, созданный в Тест-кейсе №1
Ожидаемый результат	Система информирует пользователя о том, что данные удалены.
Фактический результат	Система информирует пользователя о том, что данные удалены.
Предпосылки	Авторизация в системе за контрагента.
Постусловия	СДО удаляется со страницы списка СДО и из БД.
Статус (Пройден/Провален)	Пройден.
Комментарии	Отсутствуют.