



Министерство образования и науки России
Федеральное государственное бюджетное
образовательное учреждение
высшего образования
«Казанский национальный исследовательский
технологический университет»
(ФГБОУ ВО «КНИТУ»)

Кафедра «Интеллектуальных систем и управления информационными ресурсами»
Направление специальность 02.03.03 «Математическое обеспечение и администрирование
информационных систем»
Тема курсовой работы Прототип информационной системы «Школа танцев»

КУРСОВОЙ ПРОЕКТ

Заведующий кафедрой _____ (_____)

Руководитель проекта _____ (_____)

Студент _____ (_____)

Нормоконтролер _____ (_____)

Казань 2020г.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	2
ЗАДАНИЕ	4
ЛИСТ НОРМОКОНТРОЛЯ	5
1. РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ	6
1.1 Исследование предметной области	6
1.2 Функциональность системы	6
1.3 Действующие лица в системе и их функции	7
1.4 Диаграммы вариантов использования	8
1.5 Описание пользовательских историй	9
1.6 Макеты прототипа системы	16
1.7 ER-модель	21
1.8 Глоссарий	24
2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ	25
2.1 Описание средств разработки для реализации прототипа информационной системы	25
2.2 Проектирование прототипа информационной системы	27
2.2.1 Оконный интерфейс	27
2.2.2 Объектно-ориентированный подход для реализации прототипа информационной системы	29
ЗАКЛЮЧЕНИЕ	36
СПИСОК ЛИТЕРАТУРЫ	37

ВВЕДЕНИЕ

На сегодняшний день информационные системы являются одними из самых подходящих способов долговременного хранения и обработки информации.

Информационная система – это взаимосвязанная совокупность средств, методов и персонала, используемых для хранения, обработки и выдачи информации для достижения цели управления.

Практической значимостью проводимого исследования является формирование навыков практической работы, связанной с разработкой и реализацией информационной системы “Школа танцев”, а также закрепление и обобщение всех знаний, которые были получены во время изучения предмета и применение их в разработке прототипа информационной системы.

Целью курсовой работы является разработка прототипа информационной системы “Школа танцев”, повышение комфорта для пользователей благодаря системе, которая содержит всю необходимую информацию о школе и о тренировках клиента.

В процессе написания данной курсовой работы должны решаться следующие задачи:

1. Проанализировать предметную область и определить функциональные требования к системе;
2. Составить схему базы данных в виде ER-модели и создать базу данных школы танцев.
3. Разработать прототип информационной системы.

ЗАДАНИЕ

на курсовую работу (проект) студенту кафедры _____

Тема работы (проекта) _____

Исходные данные к проекту _____

Содержание расчетно-пояснительной записки (включая перечень подлежащих разработке вопросов, включая вопросы стандартизации и контроля качества)

Перечень графического материала (схемной документации) _____

Консультанты по проекту (с указанием относящихся к ним разделов)

Дата выдачи задания "_____" 20__ г.

Руководитель проекта _____ (_____)

(подпись)

(инициалы, фамилия)

ЛИСТ НОРМОКОНТРОЛЯ

1. Лист является обязательным приложением к пояснительной записке дипломного (курсового) проекта.

2. Нормоконтролер имеет право возвращать документацию без рассмотрения в случаях:

- нарушения установленной комплектности;
- отсутствие обязательных подписей;
- нечеткого выполнения текстового и графического материала.

3. Устранение ошибок, указанных нормоконтролером, обязательно.

ПЕРЕЧЕНЬ

замечаний и предложений нормоконтролера по курсовому проекту студента _____

(группа, инициалы, фамилия)

Лист (страница)	Условное обозначение (код ошибок)	Содержание замечаний и предложений со ссылкой на нормативный документ, стандарт или типовую документацию

Дата «__» _____ 20__ г.

Нормоконтролер _____

(подпись)

(инициалы, фамилия)

1. РАЗРАБОТКА ПРОЕКТА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ

1.1 Исследование предметной области

Система регистрирует клиентов, позволяет отслеживать расписание групповых занятий, а также предоставляет возможность записаться на разовое занятие, забронировать абонемент из различных вариантов по количеству тренировок на индивидуальные занятия или на групповые занятия. Так же система предоставляет одно пробное занятие для первого посещения. Система позволяет просматривать весь список направлений танцев. После того, как количество посещений тренировок по абонементу закончено, пользователь может снова совершить покупку. Благодаря системе администратор школы фиксирует посещения занятий клиентов. На главной странице школы система информирует, что все занятия ориентированы на людей старше 16 лет.

1.2 Функциональность системы

1. Система регистрирует пользователя- создает личный кабинет, в котором клиент будет видеть расписание групповых занятий, направления всех танцев школы, а также свои тренировки.

2. Система дает возможность записаться на одно бесплатное пробное занятие. Данное занятие может быть любого вида (индивидуальное или групповое).

3. Система предлагает выбор для покупки разового занятия, абонемента индивидуальных занятий или групповых. Виды абонементов: 4, 8, 12 или 24 занятий. Любой абонемент групповых занятий включает в себя все направления групповых занятий. При покупке абонемента на индивидуальные занятия направление танца определяется лично с тренером. Все абонементы бессрочны.

4. Система позволяет приобрести разовое занятие клиентам, которые на данный момент уже имеют абонементы.

5. Система позволяет просматривать весь список направлений танцев.

6. Система позволяет администратору школы просматривать информацию о пользователях и их тренировках: какой абонемент у клиента, вид данного абонемента, а также сколько занятий посещено и какой вид

занятий. Администратор может редактировать информацию о посещениях тренировок клиента.

Будущими пользователями системы являются клиенты и администратор школы.

1.3 Действующие лица в системе и их функции

- Роль гость

Гостем является пользователь, который находится на главной странице и еще не зарегистрирован в системе.

Функции:

- Просматривать расписание занятий
- Просматривать направления танцев
- Зарегистрироваться в системе

- Роль клиент

Клиентом является любой человек, кроме администратора данной школы, зарегистрированный в системе.

Функции:

- Забронировать абонемент
- Забронировать разовое занятие. Оплата происходит в реальной жизни.
- Записаться на одно бесплатное пробное занятие.
- Просматривать расписание занятий
- Просматривать направления танцев
- Просматривать количество занятий, которые были посещены.

- Роль администратор

Администратором является сотрудник школы, имея доступ к информации о клиентах и их тренировках.

Функции:

-Просматривать информацию о клиентах (ФИО, вид абонемента, количество посещений по абонементу, количество посещений разовых занятий, посещал ли пробное занятие)

- Редактировать информацию о клиентах: изменять количество посещений занятий и записей на пробные занятия.

1.4 Диаграммы вариантов использования

1) Диаграмма вариантов использования для гостя в соответствии с рисунком 1.1

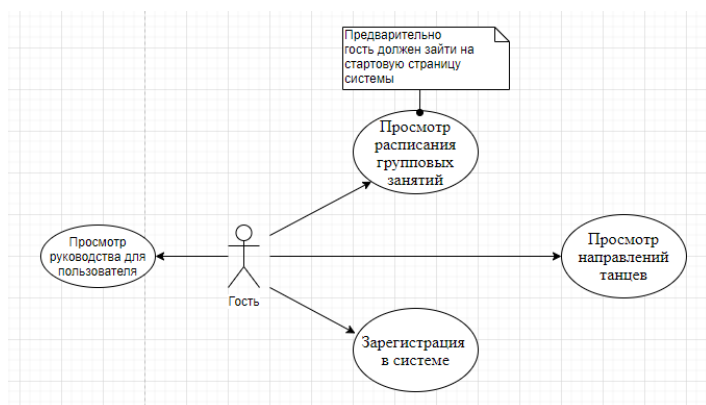


Рисунок 1.1 - Диаграмма вариантов использования для гостя.

2) Диаграмма вариантов использования для клиента в соответствии с рисунком 1.2

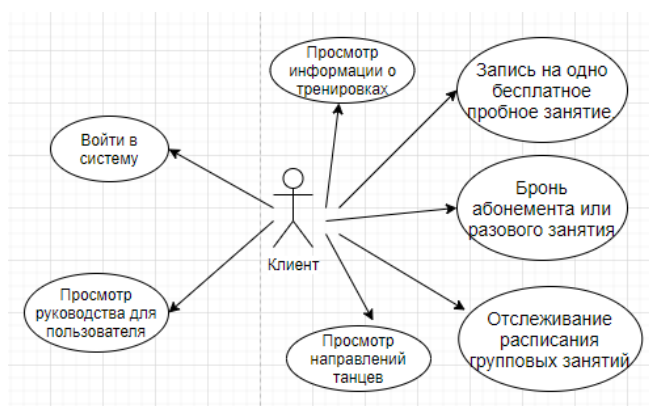


Рисунок 1.2 - Диаграмма вариантов использования для клиента.

3) Диаграмма вариантов использования для администратора в соответствии с рисунком 1.3

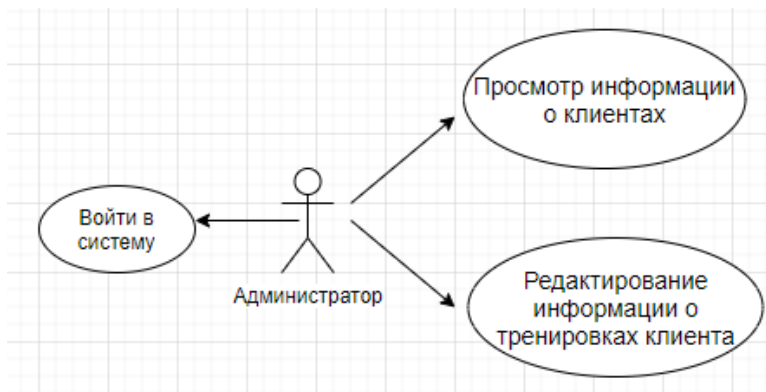


Рисунок 1.3 - Диаграмма вариантов использования для администратора.

1.5 Описание пользовательских историй

Исходя из потребностей действующих лиц данной системы, выделяются следующие варианты использования:

- Зарегистрироваться в системе.
- Просмотреть руководство по использованию.
- Войти в систему.
- Забронировать абонемент.
- Забронировать разовое занятие.
- Просмотреть расписание групповых занятий.
- Просмотреть направления танцев.
- Записаться на одно пробное бесплатное занятие.
- Просмотреть информацию о пользователях и их тренировках.
- Редактировать информацию о тренировках клиентов.

Вариант использования "Зарегистрироваться в системе"

Краткое описание:

Данный вариант использования описывает регистрацию пользователя в системе школы танцев.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет зарегистрироваться в системе школы танцев.

1. Система запрашивает имя, фамилию, логин и пароль.
2. Пользователь вводит имя, фамилию, логин и пароль.
3. Система открывает личный кабинет пользователя.

Альтернативные потоки:

1. Данный пользователь уже зарегистрирован: Если пользователь ввел логин, который уже есть в системе, то система выводит сообщение «Пользователь с таким логином уже есть». Пользователь может вернуться к началу основного потока или отказаться от регистрации в системе, при этом выполнение варианта использования завершается.

Вариант использования "Просмотреть руководство по использованию"

Краткое описание:

Данный вариант использования позволяет пользователю просмотреть руководство по пользованию. Данный вариант использования не обязывает входить в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет просмотреть руководство по пользованию системой.

1. Пользователь выбирает раздел Справка.
2. Система открывает выбранный раздел.
3. Пользователь просматривает руководство.

Вариант использования "Войти в систему"

Краткое описание:

Данный вариант использования описывает вход пользователя в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет войти в систему.

1. Система запрашивает логин и пароль.
2. Пользователь вводит имя и пароль.
3. Система подтверждает имя и пароль, после чего открывается личный кабинет пользователя.

Альтернативные потоки:

1. Неправильное логин/пароль: Если во время выполнения основного потока обнаружится, что пользователь ввел неправильное логин и/или пароль, система выводит сообщение об ошибке. Пользователь может вернуться к началу основного потока или отказаться от входа в систему, при этом выполнение варианта использования завершается.

Вариант использования "Забронировать абонемент"

Краткое описание:

Данный вариант использования позволяет клиенту забронировать абонемент. Система каталога абонементов предоставляет список всех существующих видов абонементов.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь вошел в систему и хочет забронировать абонемент.

1. Клиент переходит в раздел с абонементом.
2. Система открывает раздел.
3. Клиент выбирает вид абонемента, количество занятий в абонементе.
4. Отмеченный абонемент отмечается выбранным.
5. Система сохраняет выбранный абонемент.
6. Клиент бронирует абонемент.

7. Система сообщает о том, что бронь совершена успешно.

Вариант использования "Забронировать разовое занятие"

Краткое описание:

Данный вариант использования позволяет клиенту забронировать посещение разового индивидуального или группового занятия. Система предоставляет весь список направлений занятий.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь вошел в систему и хочет забронировать разовое занятие.

1. Клиент переходит в раздел “Записаться на разовое занятие”.
2. Система открывает раздел.
3. Клиент выбирает направление и вид занятия: групповое или индивидуальное. При выборе индивидуального занятия, не выбирается направление. Оно определяется лично с тренером. При выборе группового занятия не выбирается дата и время, доступно любое групповое занятие из расписания.
4. Отмеченный вид занятия отмечается выбранным.
5. Клиент бронирует занятие.
6. Система сообщает о том, что бронь совершена успешно.

Вариант использования "Просмотреть расписание групповых занятий"

Краткое описание:

Данный вариант использования позволяет пользователю просмотреть расписание групповых занятий всех направлений, а также осуществить поиск по направлению, дате, фамилии преподавателя или имени. Данный вариант использования не обязывает входить в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет просмотреть расписание групповых занятий.

1. Пользователь выбирает раздел “Расписание групповых занятий”.
2. Система открывает расписание групповых занятий.
3. Пользователь просматривает расписание групповых занятий.
4. Если пользователь осуществляет поиск, то система генерирует список по критерию, выбранному пользователем.

Альтернативные потоки:

1. Запись не найдена:

Если данных, введенные пользователем, нет в системе, то система выдает пустой список.

Вариант использования " Просмотр направлений танцев "

Краткое описание:

Данный вариант использования позволяет пользователю просмотреть все направления танцев данной школы и их стиль. Данный вариант использования не обязывает входить в систему.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь хочет просмотреть направления танцев.

1. Пользователь выбирает раздел “Направления”.
2. Система открывает выбранный раздел.
3. Пользователь просматривает все направления танцев данной школы и их стиль.

Вариант использования "Записаться на пробное занятие"

Краткое описание:

Данный вариант использования позволяет клиенту записаться на одно пробное бесплатное занятие любого вида и направления.

Основной поток событий:

Данный вариант использования начинает выполняться, когда пользователь вошел в систему и впервые хочет записаться на бесплатное пробное занятие.

1. Клиент переходит в раздел “Записаться на пробное занятие”.
2. Система открывает раздел.
3. Клиент выбирает направление и вид занятия: групповое или индивидуальное.
4. Данный вид занятия отмечается выбранным. При выборе индивидуального занятия, не выбирается направление. Оно определяется лично с тренером. При выборе группового занятия не выбирается дата и время, будет доступно любое групповое занятие из расписания.
5. Система сохраняет введенные данные.
6. Клиент записывается на пробное занятие.
7. Система сообщает о том, что запись совершена успешно.

Альтернативные потоки:

1. *Клиент уже записывался на пробное занятие:*

Для клиента, который уже был записан на пробное занятие раздел “Записаться на пробное занятие” не будет доступным.

Вариант использования "Просмотреть информацию о пользователях"

Краткое описание:

Данный вариант использования позволяет администратору просмотреть информацию о пользователях и их тренировок.

Основной поток событий:

Данный вариант использования начинает выполняться, когда администратор вошел в систему и хочет просмотреть информацию о клиентах: ФИО, вид абонемента, сколько занятий посещено и какое направление (для разового группового) у конкретного клиента, посещал ли пробные занятия.

1. Система открывает раздел с информацией о клиентах.
2. Администратор смотрит информацию.

Альтернативные потоки:

1. Клиент не совершал покупки.

Если клиент зарегистрирован в системе, но ничего не покупал, то в информации о его тренировках будет пусто.

Вариант использования " Редактировать информацию о клиентах"

Краткое описание:

Данный вариант использования позволяет администратору редактировать информацию о клиентах: изменять количество посещений занятий и редактировать посещение на пробные занятия

Основной поток событий:

Данный вариант использования начинает выполняться, когда администратор вошел в систему и хочет отредактировать данные клиента (изменить количество посещений занятий и редактировать посещение на пробные занятия, добавить абонемент).

1. Система открывает раздел с информацией о клиентах.
2. Администратор выбирает клиента.
3. Система открывает информацию о выбранном клиенте.
4. Администратор редактирует данные клиента (изменяет количество посещений занятий и редактирует посещение на пробные занятия).

5. Для добавления нового абонемента клиенту, администратор выбирает в уже открытом разделе раздел “Добавить абонемент”.

1.6 Макеты прототипа системы

Для оконного интерфейса прототипа информационной системы была использована технология Windows Forms.

Интерфейс включает меню и интерактивные интерфейсные средства, а также содержит набор интерфейсных элементов (кнопки, поля ввода, диалоговые окна, графические элементы и т.д.).

Основным понятием в Windows Forms является форма. Чаще всего приложение Windows Forms строится с помощью помещения элементов управления на форму и написания кода для реагирования на действия пользователя, такие как щелчки мыши или нажатия клавиш. Элемент управления — это отдельный элемент пользовательского интерфейса, предназначенный для отображения или ввода данных.

Ниже представлены макеты информационной системы Школа танцев.

- Главная страница, в соответствии с рисунком 1.4

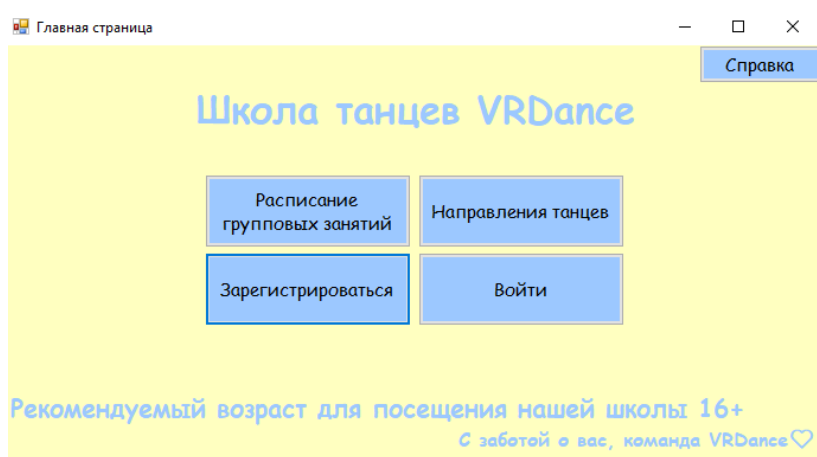


Рисунок 1.4 – Главная страница

- Кнопки главной страницы:

- 1) Кнопка расписание групповых занятий в соответствии с рисунком 1.5.

Расписание

Направление: Фамилия пр-ля: Имя пр-ля: Дата занятия: мая 2020 г.

Время	Направление	Фамилия преподавателя	Имя преподавателя	Отчество преподавателя
25.05.2020 10:00	Dancehall	Самсонова	Аделина	Матвеевна
25.05.2020 11:00	Contemporary	Кулакова	Милена	Никитовна
25.05.2020 12:00	Body Ballet	Пономарева	Стелла	Наумовна
25.05.2020 15:00	Hip Hop	Соколова	Евгения	Дмитриевна
25.05.2020 16:00	Street Dance	Нестерова	Амалия	Андреевна
25.05.2020 18:00	Twerk	Лукина	Дуня	Глебовна
25.05.2020 20:00	Vogue	Котова	Сабина	
26.05.2020 10:00	High Heels	Калашникова	Дарья	Даниловна
26.05.2020 12:00	Vogue	Котова	Сабина	
26.05.2020 14:00	Twerk	Лукина	Дуня	Глебовна
26.05.2020 17:00	Stretching	Митрофановна	Радослава	
26.05.2020 19:00	Dancehall	Самсонова	Аделина	Матвеевна
27.05.2020 10:00	Street Dance	Нестерова	Амалия	Андреевна
27.05.2020 11:00	Body Ballet	Пономарева	Стелла	Наумовна
27.05.2020 15:00	Zumba	Жукова	Ася	Онисимовна

Рисунок 1.5 – Расписание.

- 2) Кнопка Направления танцев в соответствии с рисунком 1.6

Направления

Направление	Стиль
Dancehall	Уличный
Hip Hop	Уличный
Street Dance	Уличный
Twerk	Женский
Vogue	Женский
Contemporary	Классический танец/джаз/модерн
High Heels	Экзотичный стиль на каблуках
Body Ballet	Танцевальная фитнес-программа
Zumba	Танцевальная фитнес-программа
Stretching	Растяжка

Рисунок 1.6 – Направления.

- 3) Кнопка Зарегистрироваться в соответствии с рисунком 1.7

Авторизация

Введите данные для регистрации.

Фамилия

Имя

Логин

Пароль

Рисунок 1.7 – Регистрация.

- 4) Кнопка Войти в соответствии с рисунком 1.8.

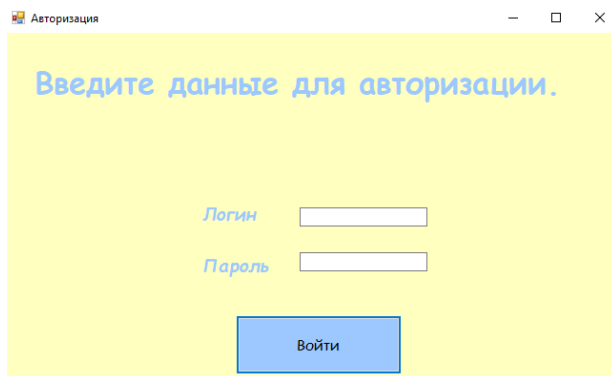


Рисунок 1.8 – Авторизация.

- Форма “Личный кабинет” в соответствии с рисунком 1.9.

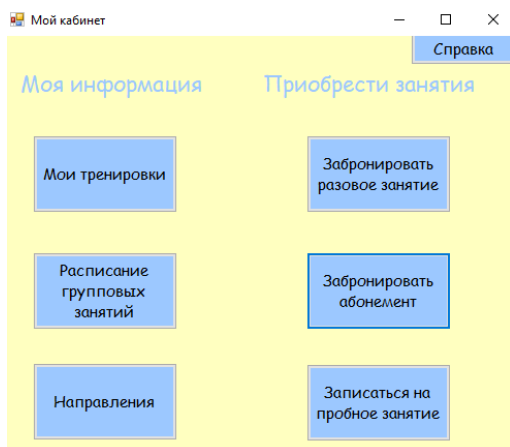


Рисунок 1.9 – Личный кабинет

- Кнопки личного кабинета
 - 1) Кнопка Мои тренировки в соответствии с рисунком 1.10.

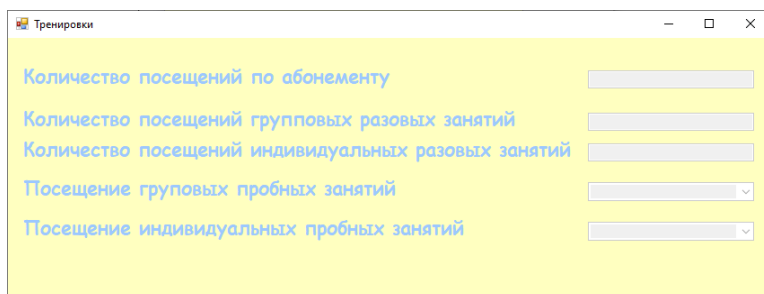


Рисунок 1.10 - Мои тренировки

- 2) Кнопка расписание групповых занятий в соответствии с рисунком 1.11.

Расписание

Направление: Фамилия пр-ля: Имя пр-ля: Дата занятия: мая 2020 г.

Время	Направление	Фамилия преподавателя	Имя преподавателя	Отчество преподавателя
25.05.2020 10:00	Dancehall	Самсонова	Аделина	Матвеевна
25.05.2020 11:00	Contemporary	Кулакова	Милена	Никитевна
25.05.2020 12:00	Body Ballet	Пономарева	Стелла	Наумовна
25.05.2020 15:00	Hip Hop	Сokolova	Евгения	Дмитриевна
25.05.2020 16:00	Street Dance	Нестерова	Амалия	Андреевна
25.05.2020 18:00	Twerk	Лукина	Дуня	Глебовна
25.05.2020 20:00	Vogue	Котова	Сабина	
26.05.2020 10:00	High Heels	Калашникова	Дарья	Даниловна
26.05.2020 12:00	Vogue	Котова	Сабина	
26.05.2020 14:00	Twerk	Лукина	Дуня	Глебовна
26.05.2020 17:00	Stretching	Митрофановна	Радослава	
26.05.2020 19:00	Dancehall	Самсонова	Аделина	Матвеевна
27.05.2020 10:00	Street Dance	Нестерова	Амалия	Андреевна
27.05.2020 11:00	Body Ballet	Пономарева	Стелла	Наумовна
27.05.2020 15:00	Zumba	Жукова	Ася	Онисимовна

Рисунок 1.11 – Расписание.

3) Кнопка Направления танцев в соответствии с рисунком 1.12

Направления

Направление	Стиль
Dancehall	Уличный
Hip Hop	Уличный
Street Dance	Уличный
Twerk	Женский
Vogue	Женский
Contemporary	Классический танец/джаз/модерн
High Heels	Экзотичный стиль на каблуках
Body Ballet	Танцевальная фитнес-программа
Zumba	Танцевальная фитнес-программа
Stretching	Растяжка

Рисунок 1.12 – Направления.

4) Кнопка забронировать разовое занятие в соответствии с рисунком 1.13

Запись на занятие

Выберите направление: Выберите дату занятия:

Выберите вид занятия: Выберите время занятия:

При выборе индивидуального занятия, не выбирается направлением. Оно определяется лично с тренером.
 При выборе группового занятия не выбирается дата и время. Вам будет

Рисунок 1.13 – Бронирование разового занятия.

5) Кнопка забронировать абонемент в соответствии с рисунком 1.14

Рисунок 1.14 – Бронирование абонемента.

6) Кнопка Записаться на пробное занятие в соответствии с рисунком 1.15

Рисунок 1.15 - Запись на пробное занятие.

• Панель администратора.

1) Администратор просматривает информацию о клиентах, в соответствии с рисунком 1.16

Фамилия клиента	Имя клиента	Отчество клиента	Вид абонемента	Количество занятий в абонементе	Количество посещений по абонементу	Количество индивидуальных занятий	Количество групповых занятий	Посещение индивидуального занятия	Посещение группового занятия	Направление
Парина	Алла	Игоревна	Индивиду...	8	0	5	4	Да	Да	Нр Нор
Ершова	Ульяна	Матвеевна				2	1	Да		Street Dance
Паткина	Милана	Александровна	Групповы...	8	3	3	2	Да		Twerk
Блохина	Лили	Германовна	Групповы...	12	11	5	3	Да		Vogue
Орлова	Марина	Венеславовна					1			Contempr...
Орлова	Марина	Венеславовна					2			Contempr...
Наумова	Ляра	Борисовна	Групповы...	24	3	1	2	Да		High Heels
Цветкова	Ирина		Индивиду...	8	0	2	1	Да	Да	Body Ballet
Соловьева	Олеся		Индивиду...	4	0		1			Zumba
Мартьянова	Галина		Групповы...	8	2	1	1	Да		Stretching

Рисунок 1.16 – Информация о клиентах.

2) Редактирование информации в соответствии с рисунком 1.17

Тренировки

Количество посещений по абонементу:

Количество посещений групповых разовых занятий:

Количество посещений индивидуальных разовых занятий:

Посещение групповых пробных занятий:

Посещение индивидуальных пробных занятий:

Рисунок 1.17 – Редактирование информации.

3) Кнопка Добавить абонемент в соответствии с рисунком 1.18

Бронирование абонемента

Пожалуйста, выберите абонемент на занятия

Выберите вид занятия:

☐ Абонемент на 4 занятия

☐ Абонемент на 8 занятий

☐ Абонемент на 12 занятий

☐ Абонемент на 24 занятия

Любой абонемент включает в себя все направления занятий.
Все абонементы бессрочны.

Рисунок 1.18 - Добавить абонемент

1.7 ER-модель

Модель “сущность-связь” (Entity-Relationshipmodel или ER – модель) представляет собой высокоуровневую концептуальную модель данных, которая описывает структуру базы данных в виде совокупности сущностей, атрибутов и связей [1].

Рассмотрим ER-модель базы данных информационной системы «Школа танцев» с ее сущностями, в соответствии с рисунком 1.16.

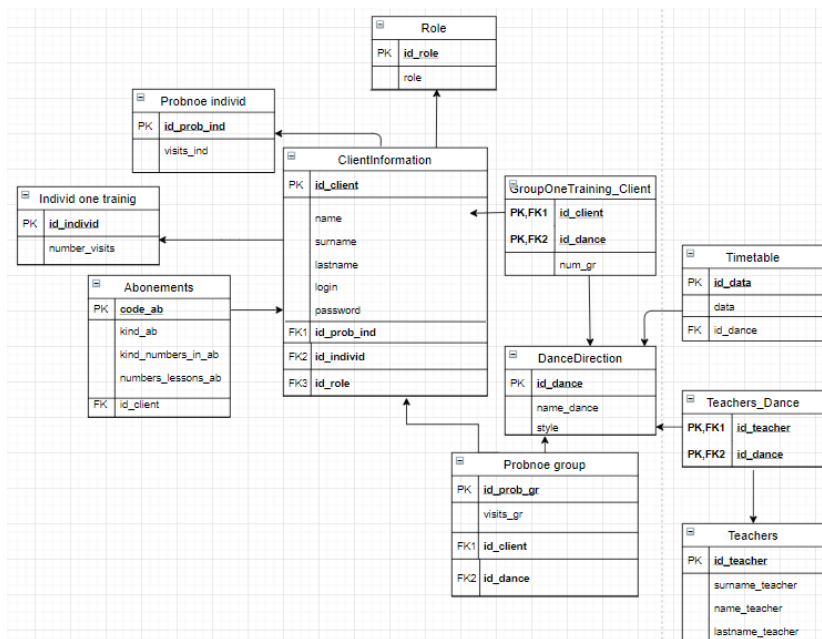


Рисунок 1.16 – ER-модель.

Описание сущностей и атрибутов:

- Таблица ClientInformation – Информация о клиентах

id_client- id клиента

name- имя клиента

surname- фамилия клиента

lastname- отчество клиента

- Таблица Abonements- Абонементы

code_ab- код абонемента

kind_ab- вид абонемента

kind_numbers_in_ab- кол-во занятий в абонменте

numbers_lessons_ab -кол-во посещений по абонементу

Таблица Group one trainig- Групповые разовые занятия

id_dance- id направления

id_gr-id занятия

num_gr- кол-во посещений разовых занятий

Таблица Individ one trainig

id_individ-id разового индивид-ого занятия

Таблица DanceDirection- Направления танцев

id_dance- id направления

name_dance – название направления

style – стиль танца

Таблица Probное group – Пробное групповое занятие

id_prob_gr – id пробного группового занятия

visits_gr– посещение

Таблица Probное individ- Пробное индивидуальное занятие

id_prob_ind- id пробного индивидуальное занятие

visits_ind- посещение

Таблица Teachers – Преподаватели

id_teacher- id преподавателя

surname_teacher – фамилия преподавателя

name_teacher – имя

lastname_teacher – отчество

Таблица Teachers_Dance- Преподаватели и танцы

id_teacher- id преподавателя

id_dance- id направления

Таблица TimeTable- Расписание

code_timetable- код

data- дата

Таблица Users

login- имя пользователя

password- пароль

role- роль

1.8 Глоссарий

1) Информационная система – система, предназначенная для обработки, поиска, хранения информации, и организационные ресурсы, обеспечивающие и распространяющие информацию.

2) Предметная область – совокупность объектов, свойства которых и отношения, между которыми представляют интерес для пользователей информационной системы.

3) Вариант использования— это описание поведения системы, с чем-то (или с кем-то) из внешней среды.

4) Windows Forms - интерфейс программирования приложений, отвечающий за графический интерфейс пользователя и являющийся частью Microsoft .NET Framework.

5) Форма - это визуальная поверхность, на которой выводится информация для пользователя.

6) Сущность - это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность должна иметь уникальное в пределах моделируемой системы наименование, выраженное существительным в единственном числе [2].

2. РАЗРАБОТКА ПРОТОТИПА ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Описание средств разработки для реализации прототипа информационной системы

Прототип информационной системы был реализован в среде разработки Visual Studio 2019 на платформе ADO.NET Entity Framework с использованием Windows Forms.

Entity Framework - программная модель, которая позволяет заполнить пробел между конструкциями базы данных и объектно-ориентированными конструкциями [4].

Так же были добавлены такие пакеты NuGet, как Entity Framework. Npsql, NpgsqlforEntityFramework6.

После добавления Entity Framework к проекту была подключена ранее созданная база данных SchoolDance.

В базе данных были созданы следующие таблицы с соответствующими атрибутами и внешними ключами, в соответствии с рисунком 2.1

```

Create table Probnoe_individ (id_prob_ind serial primary key, visits_ind text not null);
Create table Individ_one_trainig (id_individ serial primary key, number_visits int);
Create table DanceDirection (id_dance serial primary key, name_dance text not null, style text not null);
Create table Teachers (id_teacher serial primary key, surname_teacher varchar(100) not null, name_teacher
varchar(100) not null, lastname_teacher varchar(100));
Create table Probnoe_group (id_prob_gr serial primary key, visits_gr text, id_client int, id_dance int);
Create table TimeTable( data timestamp primary key, id_dance int );
Create table Roles (id_role serial primary key, role text not null);
Create table Teachers_Dance (id_teacher int REFERENCES teachers, id_dance int REFERENCES
dancedirection, PRIMARY KEY(id_teacher,id_dance) );
Create table ClientInformation (id_client serial primary key, surname VARCHAR (100) not null, name
VARCHAR (100) not null, lastname VARCHAR (100), id_prob_ind int, id_individ int, login varchar not null,
password varchar not null, id_role int REFERENCES roles);
Create table Abonements (id_ab serial primary key, kind_ab varchar (100), kind_numbers_in_ab int,
numbers_lessons_ab int, id_client int);
Create table Grouponetraining_client (id_client int REFERENCES clientinformation, id_dance int
REFERENCES dancedirection, num_gr int, PRIMARY KEY(id_client,id_dance) );

Alter table ClientInformation ADD foreign key (id_prob_ind)
References Probnoe_individ (id_prob_ind);
Alter table ClientInformation ADD foreign key (id_individ)
References Individ_one_trainig (id_individ);
Alter table Abonements ADD foreign key (id_client)
References ClientInformation(id_client);
Alter table Probnoe_group ADD foreign key (id_client)
References ClientInformation (id_client);
Alter table Probnoe_group ADD foreign key (id_dance)
References DanceDirection (id_dance);
Alter table Timetable ADD foreign key (id_dance)
References DanceDirection (id_dance);

```

Рисунок 2.1 – Создание таблиц в БД.

Заполнение таблиц данными представлено в электронном варианте.

После подключения к существующей базе данных SchoolDance, в обозревателе решений появилась созданная модель EDM, в соответствии с рисунком 2.2.

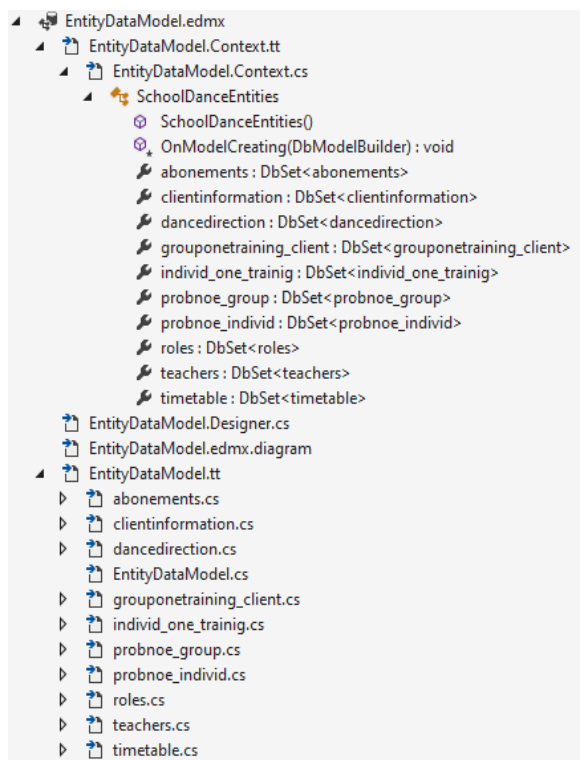


Рисунок 2.2 - Содержимое модели EDM.

По рисункам 2.1 и 2.2 видно, что каждой таблице, созданной в базе данных, соответствует свой сущностный класс.

2.2 Проектирование прототипа информационной системы

В данном разделе описывается реализация прототипа информационной системы с помощью объектно-ориентированного подхода. Прототип имеет оконный интерфейс, включающий меню и интерактивные интерфейсные средства, а также содержит набор интерфейсных элементов.

2.2.1 Оконный интерфейс

Весь интерфейс программы представлен в формах.

Пользователь взаимодействует с программой с помощью таких элементов, как кнопки, списки, поля ввода, диалоговые окна.

Приложение SchoolDance интуитивно понятно для использования за счет удобного интерфейса.

Ниже представлен оконный интерфейс для данной системы.

1) Форма “Вход” реализует вариант использования Войти с систему, в соответствии с рисунком 2.3.

Рисунок 2.3 - Форма “Вход”.

2) Форма “Расписание” реализует вариант использования “Просмотреть расписание групповых занятий”, в соответствии с рисунком 2.4.

Время	Направление	Фамилия преподавателя	Имя преподавателя	Отчество преподавателя
25.05.2020 10:00	Dancehall	Самсонова	Аделина	Матвеевна
25.05.2020 11:00	Contemporary	Кулакова	Милена	Никитевна
25.05.2020 12:00	Body Ballet	Пономарёва	Стелла	Наумовна
25.05.2020 15:00	Hip Hop	Соколова	Евгения	Дмитриевна
25.05.2020 16:00	Street Dance	Нестерова	Амалия	Андреевна
25.05.2020 18:00	Twerk	Лукина	Дуня	Глебовна
25.05.2020 20:00	Vogue	Котова	Сабина	
26.05.2020 10:00	High Heels	Калашникова	Дарья	Даниловна
26.05.2020 12:00	Vogue	Котова	Сабина	
26.05.2020 14:00	Twerk	Лукина	Дуня	Глебовна
26.05.2020 17:00	Stretching	Митрофановна	Радослава	
26.05.2020 19:00	Dancehall	Самсонова	Аделина	Матвеевна
27.05.2020 10:00	Street Dance	Нестерова	Амалия	Андреевна
27.05.2020 11:00	Body Ballet	Пономарёва	Стелла	Наумовна
27.05.2020 15:00	Zumba	Жукова	Ася	Онисимовна

Рисунок 2.4 – Форма “Расписание”.

1) 3 Форма “Запись на пробное занятие” реализует вариант использования “Записаться на пробное занятие” в соответствии с рисунком 2.5.

Рисунок 2.5 - Форма “Запись на пробное занятие”.

2.2.2 Объектно-ориентированный подход для реализации прототипа информационной системы

Объектно-ориентированно программирование (ООП) — методология программирования, которая основана на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования [6].

Основные понятия ООП:

Класс - это абстракция множества сущностей реального мира, объединенных общностью структуры и поведения [7].

Объект - это элемент класса, то есть абстракция определенной сущности [8].

Наследование - построение новых классов, на основе существующих. Наследование дает возможность добавления или переопределения данных и методов. Наследование помогает избежать размножение сущностей без необходимости. Информация не дублируется, указывается только то, что меняется. При этом класс -потомок помнит о своих "корнях"[9].

Информационная система «Школа танцев» реализована с помощью объектно-ориентированного подхода, поэтому вся логика программы прописана в классах.

Рассмотрим основные классы форм и сущностей данной системы.

А. Главная страница

1) Класс формы “Главная страница” в соответствии с рисунком 2.6

```
{
    Ссылка: 3
    public partial class MainMenuView : Form
    {
        private MainMenuViewModel context;

        Ссылка: 1
        public MainMenuView()
        {
            context = new MainMenuViewModel();
            StartPosition = FormStartPosition.CenterScreen;
            InitializeComponent();
        }

        Ссылка: 1
        private void registrationButton_Click(object sender, EventArgs e)
        {
            context.OpenAuthCommand.Execute(AuthType.SignUp);
        }

        Ссылка: 1
        private void authButton_Click(object sender, EventArgs e)
        {
            context.OpenAuthCommand.Execute(AuthType.SignIn);
        }

        Ссылка: 1
        private void openTimetableButton_Click(object sender, EventArgs e)
        {
            context.OpenTimetableCommand.Execute(null);
        }

        Ссылка: 1
        private void openDirectionsButton_Click(object sender, EventArgs e)
        {
            context.OpenDirectionsCommand.Execute(null);
        }

        Ссылка: 1
        private void button5_Click(object sender, EventArgs e) // Справка
        {
            String openPDFFile = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + @"\HelpDoc.pdf";
            System.IO.File.WriteAllBytes(openPDFFile, global::SchoolDanceManageApp.Properties.Resources.Справка);
            System.Diagnostics.Process.Start(openPDFFile);
        }
    }
}
```

Рисунок 2.6 – Класс формы “Главная страница”.

2) Класс сущности “Главная страница” в соответствии с рисунком 2.7.

В данном фрагменте кода реализован метод для открытия соответствующих форм.

```

namespace SchoolDanceManageApp.ViewModels
{
    class MainMenuViewModel : BindableBase
    {
        public Action RequestClose { get; set; }

        public ICommand OpenAuthCommand { get; private set; }

        public ICommand OpenTimetableCommand { get; private set; }

        public ICommand OpenDirectionsCommand { get; private set; }

        public MainMenuViewModel()
        {
            OpenAuthCommand = new RelayCommand((arg) =>
            {
                Navigation.Open(new AuthorizationView((AuthType)arg));
                RequestClose?.Invoke();
            });

            OpenTimetableCommand = new RelayCommand((arg) =>
            {
                Navigation.Open(new TimetableView());
                RequestClose?.Invoke();
            });

            OpenDirectionsCommand = new RelayCommand((arg) =>
            {
                Navigation.Open(new DirectionsView());
                RequestClose?.Invoke();
            });
        }
    }
}

```

Рисунок 2.7 – Класс сущности “Главная страница”.

Б. Личный кабинет.

1) Класс формы “ Личный кабинет клиента” в соответствии с рисунком 2.8.

```

PrivateOfficeViewModel context;
public PrivateOfficeView(int clientId)
{
    context = new PrivateOfficeViewModel();
    context.ClientId = clientId;
    StartPosition = FormStartPosition.CenterParent;
    InitializeComponent();

    button7.DataBindings.Add(new Binding(nameof(button7.Enabled), context, nameof(context.ClientHasNotTest), false, DataSourceUpdateMode.Never));
}

private void button5_Click(object sender, EventArgs e) // Расписание
{
    context.OpenTimetableCommand.Execute(false);
}

private void button4_Click(object sender, EventArgs e) // Направления
{
    context.OpenDirectionsCommand.Execute(null);
}

private void button3_Click(object sender, EventArgs e)
{
    context.OpenClientInfoCommand.Execute(true);
}

private void button1_Click(object sender, EventArgs e) // Забронировать абонемент
{
    context.OpenBookingCommand.Execute(null);
}

private void button2_Click(object sender, EventArgs e) // Забронировать разовое занятие
{
    context.OpenSignUpForClassCommand.Execute(ClassType.Single);
}

private void button7_Click(object sender, EventArgs e) // Пробоное
{
    context.OpenSignUpForClassCommand.Execute(ClassType.Test);
}

private void button6_Click(object sender, EventArgs e) // Справка
{
    String openPDFFile = Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments) + @"HelpDoc.pdf";
    System.IO.File.WriteAllBytes(openPDFFile, global::SchoolDanceManageApp.Properties.Resources.Cnpaska2);
    System.Diagnostics.Process.Start(openPDFFile);
}
}

```

Рисунок 2.8 - Класс формы “ Личный кабинет”.

2) Класс сущности “ Личный кабинет клиента”.

2. В данном фрагменте кода, в соответствии с рисунком 2.9, проверяется записывался ли клиент на пробное занятие. Если запись была, то кнопка Запись на пробное занятие больше не активна.

```
using System.Windows.Input;

namespace SchoolDanceManageApp.View.Models
{
    class PrivateOfficeViewModel : BindableBase
    {
        private int id_client;

        public int ClientId
        {
            get
            {
                return id_client;
            }
            set
            {
                id_client = value;
                OnPropertyChanged(nameof(ClientId));
            }
        }

        /// У клиента были пробные занятия
        public bool ClientHadNotTest
        {
            get
            {
                var query = Globals.Db.clientinformation
                    .Where(x => x.id_client == ClientId)
                    .Join(Globals.Db.probnoe_group, x => x.id_client, y => y.id_client, (client, probnoe_group) => probnoe_group);

                var query2 = Globals.Db.clientinformation
                    .Where(x => x.id_client == ClientId)
                    .Join(Globals.Db.probnoe_individ, x => x.id_client, y => y.id_client, (client, probnoe_individ) => probnoe_individ);

                return !(query.Count() != 0 && query.First().visits_gr == "Да") || (query2.Count() != 0 && query2.First().visits_ind == "Да");
            }
        }

        public ICommand OpenTmetableCommand { get; private set; }
        public ICommand OpenDirectionsCommand { get; private set; }
        public ICommand OpenBookingCommand { get; private set; }
        public ICommand OpenSignUpForClassCommand { get; private set; }
        public ICommand OpenClientInfoCommand { get; private set; }
    }
}
```

Рисунок 2.9 - Класс сущности “Личный кабинет”.

3. В данном коде, в соответствии с рисунком 2.10, создается метод для открытия других форм из личного кабинета при нажатии на соответствующие кнопки.

```
public PrivateOfficeViewModel()
{
    OpenTmetableCommand = new RelayCommand((arg) =>
    {
        Navigation.OpenDialog(new TimetableView());
    });

    OpenDirectionsCommand = new RelayCommand((arg) =>
    {
        Navigation.OpenDialog(new DirectionsView());
    });

    OpenBookingCommand = new RelayCommand((arg) =>
    {
        Navigation.OpenDialog(new BookAbonementView(ClientId));
    });

    OpenSignUpForClassCommand = new RelayCommand((arg) =>
    {
        Navigation.OpenDialog(new SignUpForClassView((ClassType)arg, ClientId));
        OnPropertyChanged(nameof(ClientHadNotTest));
    });

    OpenClientInfoCommand = new RelayCommand((arg) =>
    {
        Navigation.OpenDialog(new ClientInformationView(new Tuple<int, bool>(ClientId, (bool)arg)));
    });
}
```

Рисунок 2.10 – Продолжение класса сущности “ Личный кабинет”.

В. Панель администратора

1) Класс формы “Панель администратора” в соответствии с рисунком 2.11

```
using System.Windows.Forms;

namespace SchoolDanceManageApp.Views
{
    Ссылка: 3
    public partial class AdminPanelView : Form
    {
        AdminPanelViewModel context;

        Ссылка: 1
        public AdminPanelView()
        {
            context = new AdminPanelViewModel();
            StartPosition = FormStartPosition.CenterParent;
            InitializeComponent();

            dataGridView1.ReadOnly = true;
            dataGridView1.DataBindingComplete += (sender, e) =>
            {
                dataGridView1.Columns[0].HeaderText = "Фамилия клиента";
                dataGridView1.Columns[1].HeaderText = "Имя клиента";
                dataGridView1.Columns[2].HeaderText = "Отчество клиента";
                dataGridView1.Columns[3].HeaderText = "Вид абонемента";
                dataGridView1.Columns[4].HeaderText = "Количество занятий в абонементе";
                dataGridView1.Columns[5].HeaderText = "Количество посещений по абонементу";
                dataGridView1.Columns[6].HeaderText = "Количество индивидуальных разовых занятий";
                dataGridView1.Columns[7].HeaderText = "Количество групповых разовых занятий";
                dataGridView1.Columns[8].HeaderText = "Посещение индивидуальных пробных занятий";
                dataGridView1.Columns[9].HeaderText = "Посещение групповых пробных занятий";
                dataGridView1.Columns[10].HeaderText = "Направление разовых групповых занятий";

                //Скрытие id
                dataGridView1.Columns[11].Visible = false;
            };
            dataGridView1.DataBindings.Add(new Binding(nameof(dataGridView1.DataSource), context, nameof(context.DataSource), false, DataSourceUpdateMode.Never));
        }

        Ссылка: 1
        private void button1_Click(object sender, EventArgs e) // Обновить
        {
            context.OnPropertyChanged(nameof(context.DataSource));
        }

        Ссылка: 1
        private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
        {
            if (e.RowIndex != -1)
            {
                context.OpenClientInformationCommand.Execute(new Tuple<int, bool>((int)dataGridView1[11, e.RowIndex].Value, false));
            }
        }
    }
}
```

Рисунок 2.11 – Класс формы “Панель администратора”.

4. Класс сущности “Панель администратора” в соответствии с рисунком 2.12.

```
Ссылка: 3
class AdminPanelViewModel : BindableBase
{
    Ссылка: 1
    // Данные из БД для таблицы во View
    public object DataSource
    {
        get
        {
            var clientinfo = Globals.Db.clientinformation.ToList();

            var query = (from info in clientinfo
                join a in Globals.Db.abonements on info.id_client equals a.id_client
                join b in Globals.Db.individ_one_trainig on info.id_individ equals b.id_individ
                join c in Globals.Db.probnoe_individ on info.id_prob_ind equals c.id_prob_ind
                join pr in Globals.Db.probnoe_group on info.id_client equals pr.id_client
                join raz in Globals.Db.grouponetraining_client on info.id_client equals raz.id_client
                join d in Globals.Db.dancedirection on raz.id_client equals d.id_dance
                orderby info.id_client
                select new
                {
                    info.surname,
                    info.name,
                    info.lastname,
                    a.kind_ab,
                    a.kind_numbers_in_ab,
                    a.numbers_lessons_ab,
                    b.number_visits,
                    raz.num_gr,
                    c.visits_ind,
                    pr.visits_gr,
                    d.name_dance,
                    info.id_client
                }).Distinct().ToList();

            return query;
        }
    }

    Ссылка: 2
    // Команда открытия формы ClientInformation
    public ICommand OpenClientInformationCommand { get; set; }

    Ссылка: 1
    public AdminPanelViewModel()
    {
        OpenClientInformationCommand = new RelayCommand((arg) =>
        {
            Navigation.OpenDialog(new ClientInformationView((Tuple<int, bool>)arg));
        });
    }
}
```

Рисунок 2.12 – Класс сущности “Панель администратора”.

Г. Расписание

1) Класс формы “Расписание” в соответствии с рисунком 2.13.

```
namespace SchoolDanceManageApp.Views
{
    class TimestableView : Form
    {
        TimestableViewModel context;
        public TimestableView()
        {
            context = new TimestableViewModel();
            StartPosition = FormStartPosition.CenterParent;
            InitializeComponent();

            dataGridView1.ReadOnly = true;
            dataGridView1.DataBindingComplete += (sender, e) =>
            {
                dataGridView1.Columns[0].HeaderText = "Время";
                dataGridView1.Columns[1].HeaderText = "Направление";
                dataGridView1.Columns[2].HeaderText = "Имя преподавателя";
                dataGridView1.Columns[3].HeaderText = "Нов преподаватель";
                dataGridView1.Columns[4].HeaderText = "Отчество преподавателя";
            };

            dataGridView1.DataBindings.Add(new Binding(nameof(dataGridView1.DataSource), context, nameof(context.DataSource), false, DataSourceUpdateMode.Never));
            textBox1.DataBindings.Add(new Binding(nameof(textBox1.Text), context, nameof(context.DirectionFilter), false, DataSourceUpdateMode.OnPropertyChanged));
            textBox2.DataBindings.Add(new Binding(nameof(textBox2.Text), context, nameof(context.SurnameFilter), false, DataSourceUpdateMode.OnPropertyChanged));
            textBox3.DataBindings.Add(new Binding(nameof(textBox3.Text), context, nameof(context.FirstNameFilter), false, DataSourceUpdateMode.OnPropertyChanged));
            checkBox1.DataBindings.Add(new Binding(nameof(checkBox1.Checked), context, nameof(context.DateFilterEnabled), false, DataSourceUpdateMode.OnPropertyChanged));
            dateTimePicker1.DataBindings.Add(new Binding(nameof(dateTimePicker1.Enabled), context, nameof(context.DateFilterEnabled), false, DataSourceUpdateMode.Never));
            dateTimePicker1.DataBindings.Add(new Binding(nameof(dateTimePicker1.Value), context, nameof(context.DateFilter), false, DataSourceUpdateMode.OnPropertyChanged));
        }
    }
}
```

Рисунок 2.13 - Класс формы “Расписание”.

2) Класс сущности “Расписание”.

- В данном фрагменте кода, в соответствии с рисунком 2.14, выполняется поиск данных по фамилии преподавателя, а также осуществляется проверка на применение поиска по дате.

```
namespace SchoolDanceManageApp.Models
{
    class TimestableViewModel : BindableBase
    {
        private string surnameFilter = string.Empty;
        private string firstNameFilter = string.Empty;
        private string directionFilter = string.Empty;
        private DateTime dateFilter = DateTime.Now;
        private bool dateFilterEnabled = false;

        // Применяется ли поиск по дате
        public bool DateFilterEnabled
        {
            get
            {
                return dateFilterEnabled;
            }
            set
            {
                dateFilterEnabled = value;
                OnPropertyChanged(nameof(DateFilterEnabled));
                OnPropertyChanged(nameof(DataSource));
            }
        }

        // Поиск по фамилии
        public string SurnameFilter
        {
            get
            {
                return surnameFilter;
            }
            set
            {
                surnameFilter = value;
                OnPropertyChanged(nameof(SurnameFilter));
                OnPropertyChanged(nameof(DataSource));
            }
        }
    }
}
```

Рисунок 2.14 - Класс сущности “Расписание”.

- В данном фрагменте кода, в соответствии с рисунком 2.15, выполняется поиск данных по имени преподавателя, по направлению и по дате.

```

// Поиск по имени
Ссылка: 2
public string FirstNameFilter
{
    get
    {
        return firstNameFilter;
    }
    set
    {
        firstNameFilter = value;
        OnPropertyChanged(nameof(FirstNameFilter));
        OnPropertyChanged(nameof(DataSource));
    }
}

//Поиск по направлениям
Ссылка: 2
public string DirectionFilter
{
    get
    {
        return directionFilter;
    }
    set
    {
        directionFilter = value;
        OnPropertyChanged(nameof(DirectionFilter));
        OnPropertyChanged(nameof(DataSource));
    }
}

// Поиск по дате
Ссылка: 2
public DateTime DateFilter
{
    get
    {
        return dateFilter;
    }
    set
    {
        dateFilter = value;
        OnPropertyChanged(nameof(DateFilter));
        OnPropertyChanged(nameof(DataSource));
    }
}

```

Рисунок 2.15 – Поиск по дате, имени и направлению.

- В данном фрагменте кода, в соответствии с рисунком 2.16, определяются данные для вывода в таблицу (DataGridView).

```

public object DataSource
{
    get
    {
        var timesheet = (from time in Globals.Db.timetable
                        select time).ToList();

        var query = (from time in timesheet
                    join d in Globals.Db.dancedirection on time.id_dance equals d.id_dance
                    join tt in Globals.Db.teachers on d.id_dance equals tt.id_teacher
                    orderby time.data
                    select new
                    {
                        time.data,
                        d.name_dance,
                        tt.surname_teacher,
                        tt.name_teacher,
                        tt.lastname_teacher
                    })
                    .Where(x =>
                        x.name_dance.Contains(directionFilter)
                        && x.surname_teacher.Contains(surnameFilter)
                        && x.name_teacher.Contains(firstNameFilter)
                        && (dateFilterEnabled ? x.data.Date == dateFilter.Date : true))
                    .ToList();

        return query;
    }
}

```

Рисунок 2.16 – Данные для таблицы.

ЗАКЛЮЧЕНИЕ

В результате выполнения курсовой работы был разработан прототип информационной системы «Школа танцев», а также были проведены проверка и тестирование данной системы.

В результате проектирования разработанная система позволяет оперативно получать информацию о школе танцев для клиентов: расписание занятий, список направлений, информация о тренировках клиента, а также позволяет администратору просматривать всю информацию о клиентах и вносить изменения о количестве тренировок клиентов. Все выше перечисленное является необходимым для качественного пользования школой танцев.

Информационная система «Школа танцев», является учебной системой для получения опыта в информационных технологиях, но она так же может быть полностью доработана и видоизменена при необходимости.

СПИСОК ЛИТЕРАТУРЫ

- 1) Дунаев, В. В. Базы данных. Язык SQL для студента / В.В. Дунаев. - М.: БХВ-Петербург, 2017. - 288 с.
- 2) PostgreSQL[электронный ресурс].URL:<https://www.postgresql.org/>
- 3) Гутман, Г. Н. Объектно-реляционная СУБД PostgreSQL: учебное пособие / Г. Н. Гутман. – Самара: АСИ СамГТУ, 2016. — 204с.
- 4) Введение в Entity Framework[электронный ресурс].
<https://metanit.com/sharp/entityframework/1.1.php>
- 5) Rob Miles. C# Programming Yellow Book, 8.2 November 2016
- 6) Мартин Роберт К.. Чистый код: Питер, 2018 г.
- 7) Фримен Эрик, Бейтс Берт, Сьерра Кэтти. Паттерны проектирования: Питер, 2017 г.
- 8) Албахари Дж., Албахари Б. C# 7.0. Карманный справочник. Скорая помощь для программистов на C# 7.0: Артеменко Ю.Н, 2017.
- 9) Ошероув Р. Искусство автономного тестирования с примерами на C#: Слинкин А.А.: 2016г.
- 10) 10. Джон Шарп: Microsoft Visual C#. Подробное руководство: Питер, 2017.

