

В работе представлена информационная система управления местами в школах. Приложение разработано на платформе Windows. Цель работы - рассмотреть принципы разработки информационных систем и разработать программу на основе этих принципов.

Для разработки были выбраны язык программирования – C# и среда разработки – Visual Studio. Для создания базы данных использовалась СУБД SQLite.

В разработанном приложении доступны возможность добавления, удаления, изменения и поиска записей в базе данных. Проведены тестирование и отладка программы.

The work presents an information system for managing places in schools. The application is developed on the Windows. The purpose of the work is to consider the principles of developing information systems and to develop a program based on these principles.

For development, the programming language C# and the development environment Visual Studio were selected. The SQLite DBMS was used to create the database.

The developed application provides the ability to add, delete, change and search for records in the database. Testing and debugging of the program were carried out.

Содержание

Введение.....	6
1. Анализ технического задания.....	8
1.1. Описание предметной области.....	10
1.2. Аналоги	11
2. Модели данных	Ошибка! Закладка не определена.
2.1. Концептуальная модель	13
2.2. Логическая модель	14
2.3. Приведение логической модели к физической.....	16
3. Разработка и реализация АИС	17
3.1. Создание SQL-запросов	17
3.2. Руководство пользователя	21
3.3. Руководство программиста.....	30
4. Тестирование АИС.....	Ошибка! Закладка не определена.
Заключение	400
Список литературы	411
Приложение А: Модели данных.....	422
Приложение Б: Текст кода	433

					МИВУ 10.03.01-19ПЗ		
Изм.	Лист	№ докум.	Подпись	Дата			
Разраб.		Смирнов В.Д.			АИС распределения детей по школах	Лит.	Лист
Провер.		Колпаков А. А.					Листов
Реценз.							5
Н. Контр.							53
Утверд.						МИВлГУ ИБ-122	

Введение.

Для анализа деятельности организации в целом, необходима четкая структурированная информация о происходящем в организации. Эту задачу можно решить с помощью автоматизированной информационной системы. Автоматизированная информационная система (АИС) — программно-аппаратный комплекс для авторизации деятельности организации, который обеспечивает хранение, передачу и обработку информации. АИС — это четкие структурированные процессы, которые можно полностью контролировать.

В автоматизированных информационных системах за хранение любой информации отвечают:

1. На физическом уровне:
 - а. внешние накопители;
 - б. встроенные устройства памяти (RAM);
 - в. массивы дисков;
2. На программном уровне:
 - а. СУБД;
 - б. файловая система ОС;
 - в. системы хранения мультимедиа, документов и т. д.

На сегодняшний день достаточно широко применяются разнообразные программные средства при работе с компьютером. В их числе находятся и автоматизированные информационные системы.

В современной вычислительной технике ИС представляет собой целый программный комплекс, который дает возможность надежно хранить данные в памяти, выполнять преобразования информации и производить вычисления с помощью удобного и легкого для пользователя интерфейса.

Исходя из вышесказанного, использование современных информационных систем позволяет нам:

1. Работать с огромными объемами данных;

					МИВУ 10.03.01-19ПЗ	Лист
						6
Изм.	Лист	№ докум.	Подпись	Дата		

2. Хранить какие-либо данные в течение довольно длительного временного периода;
3. Связать несколько компонентов, которые имеют свои определенные локальные цели, задачи и разнообразные приемы функционирования, в одну систему для работы с информацией;
4. Существенно снизить затраты на доступ и хранение к любым необходимым нам данным;
5. Довольно - таки быстро найти всю необходимую нам информацию и т. д.

В качестве классического примера современной информационной системы, стоит упомянуть банковские системы, АС управления предприятиями, системы резервирования железнодорожных или авиационных билетов и т. д.

Целью курсовой работы является создание информационной системы для добавления, удаления, изменения и поиска записей в базе данных.

Задачи, работающие для этой цели:

1. Проектирование логической и физической моделей базы данных.
2. Реализация физической модели в одной из современных СУБД.
3. Подключение БД к программе с помощью средств среды разработки.
4. Написание обработчиков событий и функций для добавления, удаления, изменения и поиска записей в БД.

1. Анализ технического задания.

В данной курсовой работе поставлена задача создания автоматизированной информационной системы (АИС) управления местами в школах. Для разработки такой системы потребуется СУБД для хранения данных и среда разработки для создания пользовательского интерфейса и функционала работы с базой данных.

Согласно заданию, в системе нужно учесть следующие особенности:

Работа с заявками: необходимо учитывать статусы заявок (например, в ожидании, отказано, распределено) и использовать предоставленные данные для их обработки.

Распределение учеников: система должна учитывать свободные места в школах и классы, а также наличие льгот для правильного распределения детей.

Фильтрация данных: система должна предоставлять возможности фильтрации данных для удобства работы пользователей, например, фильтры по школе, классу или статусу заявок.

Для реализации базы данных необходимо разработать структуру из нескольких взаимосвязанных таблиц для обеспечения целостности данных. Для разработки базы данных в рамках поставленной задачи можно использовать SQLite. SQLite — это быстрая и легкая встраиваемая однофайловая СУБД на языке C, которая не имеет сервера и позволяет хранить всю базу локально на одном устройстве. Для работы SQLite не нужны сторонние библиотеки или службы. Она обладает широкими возможностями для построения запросов, обеспечивает целостность данных и легко интегрируется с приложениями на C#.

Для создания программы, работающей с базой данных, будет использоваться язык программирования C# с использованием Windows Forms. Эти технологии позволяют реализовать современный интерфейс с интуитивно понятной навигацией и необходимой функциональностью.

					МИВУ 10.03.01-19ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		8

Программа должна быть разработана с учётом потребностей пользователей, таких как сотрудники отделов образования, заведующие учреждениями и родители. Интерфейс должен быть простым, интуитивным и обеспечивать доступ ко всем функциям без лишних сложностей. Приложение должно быстро реагировать на запросы пользователя, корректно обрабатывать данные и возвращать точные результаты. Это особенно важно для системы распределения мест в школах, где требуется точность.

					МИВУ 10.03.01-19ПЗ	Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

1.1. Описание предметной области

В данной курсовой работе рассматривается предметная область «Система распределения учащихся по школам». Основной задачей является проектирование и разработка приложения для автоматизации процесса распределения учащихся по школам.

Процесс распределения учащихся осуществляется следующим образом:

- 1) Регистрация новых учащихся. Сбор информации о потенциальных учениках.
- 2) Анализ школ. Оценка доступных мест в школах и их специализаций.
- 3) Распределение. Автоматическое распределение учащихся по школам с учетом различных критериев.

В системе есть несколько категорий пользователей: сотрудники и системные администраторы.

Основная задача курсовой работы - разработка приложения для автоматизации процесса распределения учащихся по школам.

В Системе можно выделить следующие сущности:

Школы:

- ID школы
- Название
- Адрес
- Количество свободных мест
- Специализация

Учащиеся:

- ФИО
- Дата рождения
- Адрес проживания
- Контактные данные родителей
- Предпочтительные школы

Администраторы:

					МИВУ 10.03.01-19ПЗ	Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

- ФИО
- Логин
- Пароль
- Роль в системе
- Контактные данные

Распределения:

- ID распределения
- Данные учащегося
- Школа назначения
- Дата распределения
- Статус

База данных будет содержать информацию об учащихся, школах, администраторах и распределениях. Это позволит автоматизировать все основные процессы работы системы распределения.

1.1 Аналогии

В настоящее время существует несколько систем автоматизированного распределения учащихся по школам: «Е-услуги» и АИС «Запись в школу»

Достоинства существующих систем:

- Автоматизация процесса распределения
- Учет различных критериев при распределении
- Прозрачность процесса для всех участников
- Возможность онлайн-мониторинга статуса распределения

Недостатки:

- Сложность настройки и внедрения
- Высокая стоимость готовых решений
- Недостаточная гибкость при учете местных особенностей
- Сложность интеграции с существующими системами учета

					МИВУ 10.03.01-19ПЗ	Лист
						11
Изм.	Лист	№ докум.	Подпись	Дата		

1. Модели данных

Данный этап является самым важным при создании АИС. Здесь выделяются сущности, атрибуты сущностей и связи между сущностями. На основе полученной диаграммы “Сущность – связь” или логической модели строятся функциональные модели системы и диаграмма потоков данных. Для создания базы данных, нужно логическую модель представить в виде физической.

Разработка моделей данных включает 3 этапа: [7]

1) Концептуальная модель

- определение сущностей (объектов, понятий) предметной области и их взаимосвязей;
- выявление атрибутов сущностей и описание их характеристик;
- построение диаграммы сущность-связь (ER-диаграммы) для визуального представления модели.

2) Логическая модель:

- трансформация концептуальной модели в структуру, подходящую для конкретной СУБД;
- определение типов данных, ключей, индексов, ограничений целостности;
- нормализация данных для устранения избыточности и аномалий;
- спецификация отношений между сущностями (таблицами).

3) Физическая модель:

- оптимизация логической модели для конкретной СУБД и аппаратной платформы;
- определение физического хранения данных: файлов, томов, буферов и т.д.;
- результат - физическая схема базы данных, готовая для реализации.

					МИВУ 10.03.01-19ПЗ	Лист
						12
Изм.	Лист	№ докум.	Подпись	Дата		

2.1. Концептуальная модель

Концептуальная модель данных является первым и наиболее абстрактным уровнем проектирования базы данных. На этом этапе определяются ключевые сущности предметной области, их атрибуты и взаимосвязи. [7]

В системе выделены следующие базовые сущности и их атрибуты:

1) Школы

Атрибуты школы: ID, Название, Адрес, Количество мест, Количество мест в классе

2) Дети

Атрибуты учащегося: ID, ФИО, Дата рождения, Возраст, Пол

3) Пользователи

Атрибуты пользователей: ID_Пользователя, ФИО, Логин, Пароль, Роль

4) Распределения

Атрибуты распределения: ID, ID_Ребенка, ID_Школы, Дата распределения, Статус, Класс

5) Родители

Атрибуты родителей: ID_Родителя, ID_Пользователя, Телефон, Адрес

Все выделенные сущности объединены в общую концептуальную модель данных, которая будет представлена на рисунке 1.

					МИВУ 10.03.01-19ПЗ	Лист
						13
Изм.	Лист	№ докум.	Подпись	Дата		

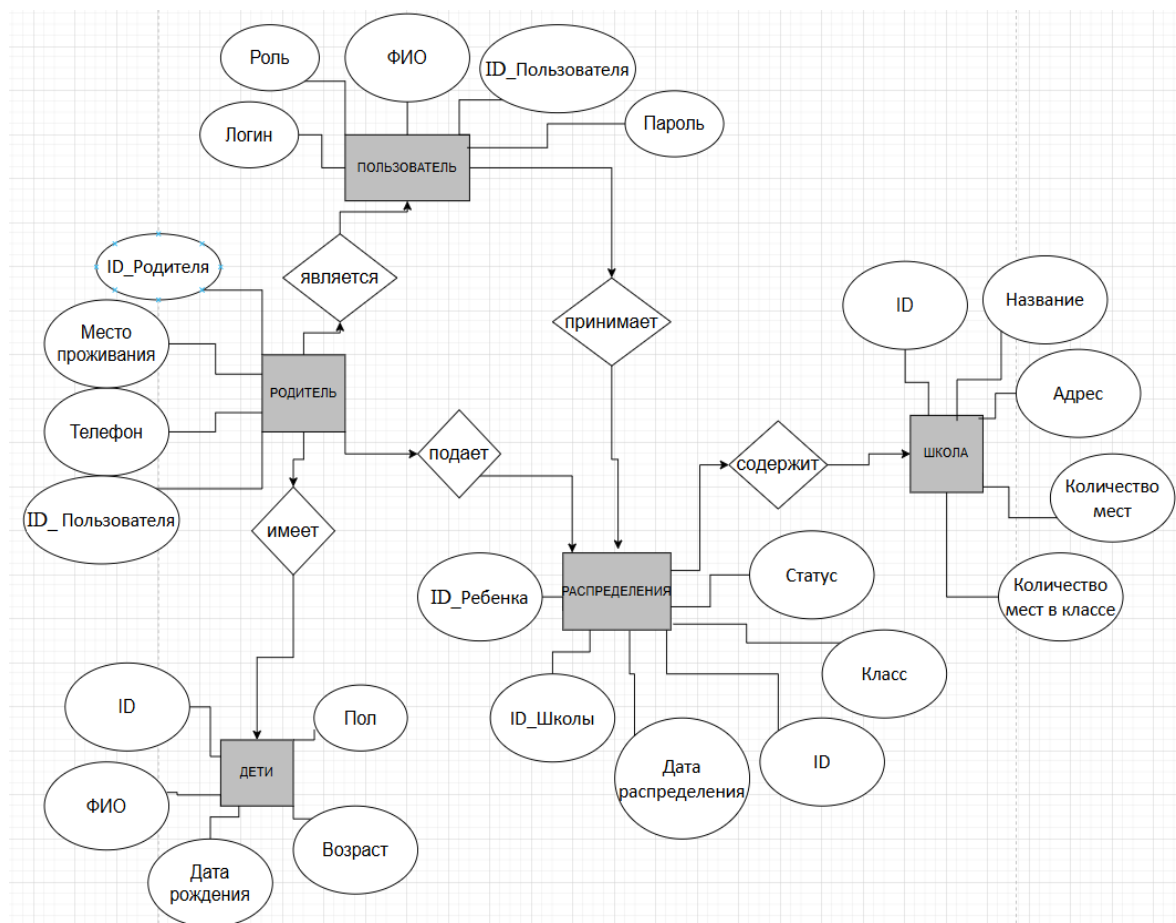


Рис. 1 – Концептуальная модель данных.

2.2. Логическая модель

Логическая модель данных — это абстрактное представление структуры данных, которое используется для планирования и проектирования баз данных. [7]

Ее создание начинается с анализа предметной области и выделения сущностей.

ПОЛЬЗОВАТЕЛЬ

- Идентификатор (РК)
- Имя пользователя
- Логин
- Пароль
- Роль

РОДИТЕЛЬ

- Идентификатор (РК)
- Телефон
- Адрес проживания
- Данные пользователя (FK)

РЕБЁНОК

- Идентификатор (РК)
- ФИО
- Дата рождения
- Возраст
- Пол
- Данные родителя (FK)

ШКОЛА

- Идентификатор (РК)
- Название
- Адрес
- Общее количество мест
- Места по классам

ЗАЯВЛЕНИЕ

- Идентификатор (РК)
- Дата подачи
- Статус
- Класс
- Льгота
- Данные ребёнка (FK)
- Выбранная школа (FK)

Связи между сущностями:

пользователь ——— родитель (1:1)

					МИВУ 10.03.01-19ПЗ	Лист
						15
Изм.	Лист	№ докум.	Подпись	Дата		

родитель — ребёнок (1:м)
ребёнок — заявление (1:м)
школа — заявление (1:м)

На основе этих данных была составлена логическая модель данных, представленная на рисунке 2.

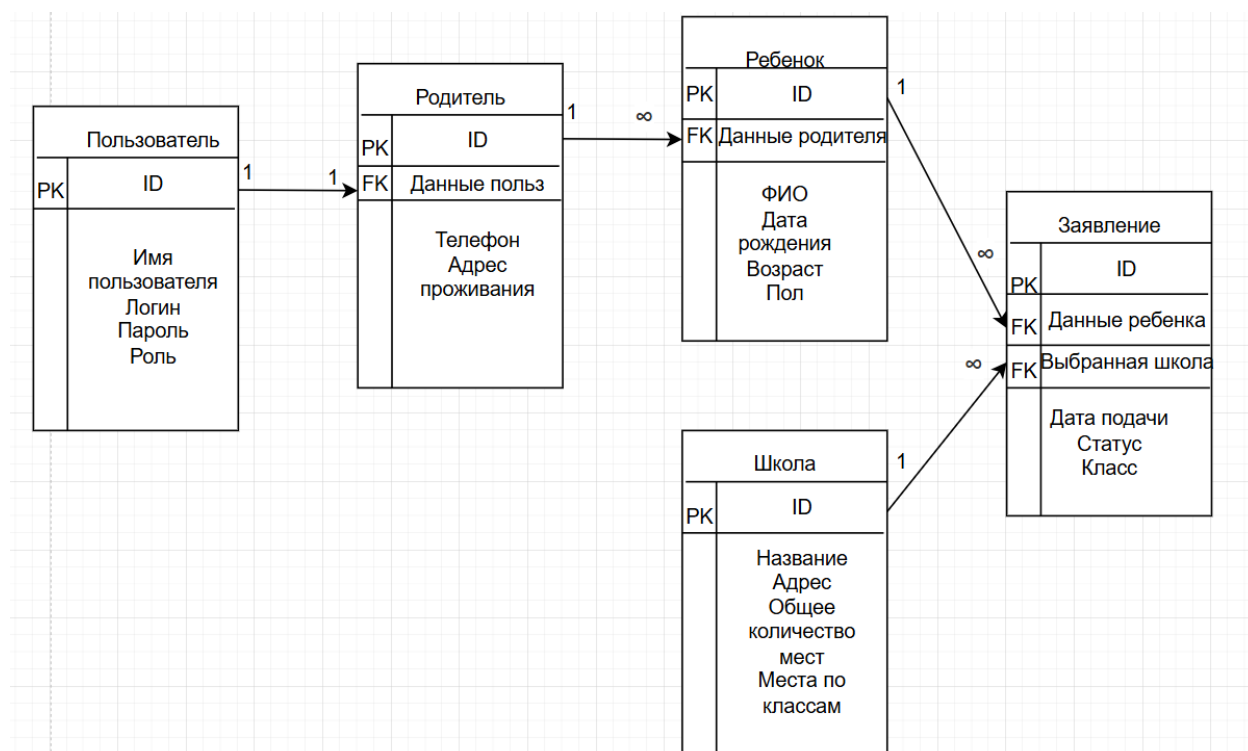


Рис. 2 – Логическая модель данных

2.3. Приведение логической модели к физической

Физическая модель данных — это заключительный шаг в процессе моделирования данных. [7] Она представляет фактические детали реализации в конкретной системе управления базами данных (СУБД).

К основным задачи физической модели данных относится:

- 1) Определение физических характеристик таблиц и полей:
 - Типы данных для каждого атрибута (integer, varchar, date и т.д.)
 - Размеры полей (длина строк)

- Ограничения целостности данных (первичные/внешние ключи, уникальность, проверка значений)

2) Проектирование физической структуры базы данных:

- Определение физического хранения данных (файлы, тома, блоки)
- Настройка индексов для ускорения доступа к данным
- Решения по кластеризации таблиц
- Механизмы резервного копирования и восстановления данных

3) Оптимизация производительности:

- Анализ запросов и создание оптимальных индексов
- Денормализация данных для повышения скорости выборки
- Использование материализованных представлений

Физическая модель данных представлена на рисунке 3.

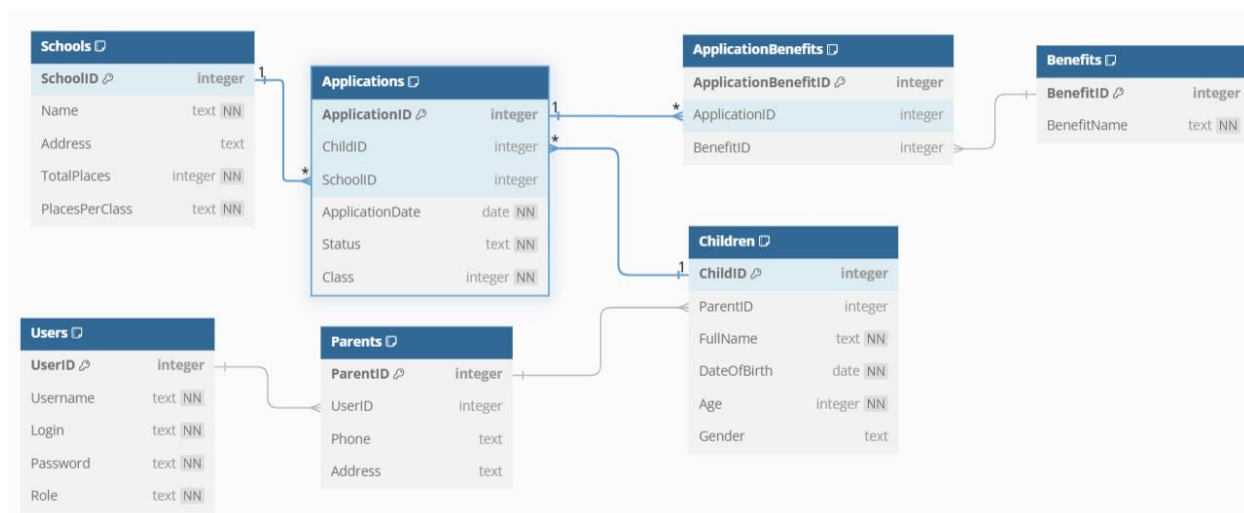


Рис. 3 – Физическая модель данных

2. Разработка и реализация АИС

3.1 Создание SQL-запросов

SQL-запросы являются основным механизмом взаимодействия с реляционными базами данных. Они позволяют эффективно управлять данными, обеспечивая их хранение, извлечение, модификацию и удаление.

[7]

Результатом запроса является представление — виртуальная таблица, представляющая собой поименованный запрос, который будет подставлен как подзапрос при использовании представления. [7] В отличие от обычных таблиц реляционных баз данных, представление не является самостоятельной частью набора данных, хранящегося в базе.

1.1.1 Запросы создания таблиц (CREATE TABLE):

```
CREATE TABLE IF NOT EXISTS ApplicationBenefits (
    ApplicationBenefitID INTEGER PRIMARY KEY AUTOINCREMENT,
    ApplicationID INTEGER NOT NULL,
    BenefitID INTEGER NOT NULL,
    FOREIGN KEY (ApplicationID) REFERENCES
Applications(ApplicationID) ON DELETE CASCADE,
    FOREIGN KEY (BenefitID) REFERENCES Benefits(BenefitID)
);

CREATE TABLE IF NOT EXISTS Applications (
    ApplicationID INTEGER PRIMARY KEY AUTOINCREMENT,
    ChildID INTEGER NOT NULL,
    SchoolID INTEGER NOT NULL,
    ApplicationDate DATE NOT NULL,
    Status TEXT NOT NULL CHECK (Status IN ('Pending', 'Approved',
'Rejected')),
    Class INTEGER NOT NULL,
    FOREIGN KEY (ChildID) REFERENCES Children(ChildID) ON
DELETE CASCADE,
    FOREIGN KEY (SchoolID) REFERENCES Schools(SchoolID)
);
```

1.1.2 Запросы для выборки данных (SELECT):

1) Запрос для получения всех заявлений с информацией о детях и школах:

SELECT

a.ApplicationID,
c.FullName as 'ФИО Ребенка',
s.Name as 'Школа',
a.ApplicationDate as 'Дата заявления',
a.Status as 'Статус',
a.Class as 'Класс'

FROM Applications a

LEFT JOIN Children c ON a.ChildID = c.ChildID

LEFT JOIN Schools s ON a.SchoolID = s.SchoolID;

	ApplicationID	ФИО Ребенка	Школа	Дата заявления	Статус	Класс
1	1	Иван Иванов	Школа №1	2024-07-01	Approved	2
2	2	Мария Иванова	Гимназия №2	2024-07-01	Approved	1
3	3	Алексей Петров	Школа №1	2024-07-02	Approved	4
4	4	Алексей Сергеев	Гимназия №2	2024-12-27 18:11:18.2068711	Approved	4

Рис. 4 – Результат выполнения запроса на таблицах Школы и Ученики

2) Запрос для получения информации о льготах по заявлению:

SELECT

a.ApplicationID,
c.FullName as 'ФИО Ребенка',
b.BenefitName as 'Льгота'

FROM Applications a

JOIN Children c ON a.ChildID = c.ChildID

JOIN ApplicationBenefits ab ON a.ApplicationID = ab.ApplicationID

JOIN Benefits b ON ab.BenefitID = b.BenefitID;

	ApplicationID	ФИО Ребенка	Льгота
1	1	Иван Иванов	Многодетная семья
2	1	Иван Иванов	Инвалидность
3	2	Мария Иванова	Мать-одиночка
4	3	Алексей Петров	Инвалидность
5	4	Алексей Сергеев	Многодетная семья
6	4	Алексей Сергеев	Инвалидность

Рис. 5 – Результат выполнения запроса на таблице Льготы

1.1.3 Запрос для вставки данных (INSERT):

```

INSERT INTO Applications (
    ChildID,
    SchoolID,
    ApplicationDate,
    Status,
    Class
) VALUES (
    1,
    1,
    '2024-07-01',
    'Pending',
    2
);

INSERT INTO ApplicationBenefits (
    ApplicationID,
    BenefitID
) VALUES (
    1,
    1
);

```

1.1.4 Запрос для обновления данных (UPDATE):

```
UPDATE Applications
SET Status = 'Approved',
    Class = 3
WHERE ApplicationID = 1;

UPDATE Children
SET FullName = 'Новое ФИО',
    Age = 9,
    Gender = 'Мужской'
WHERE ChildID = 1;
```

1.1.5 Запрос для удаления данных (DELETE):

```
DELETE FROM Applications
WHERE ApplicationID = 1;

DELETE FROM Children
WHERE ChildID = 1;
```

3.2 Руководство пользователя

При запуске приложения первым шагом для пользователя служит окно авторизации, показанное на рисунке 6. Оно служит для идентификации и аутентификации пользователей.

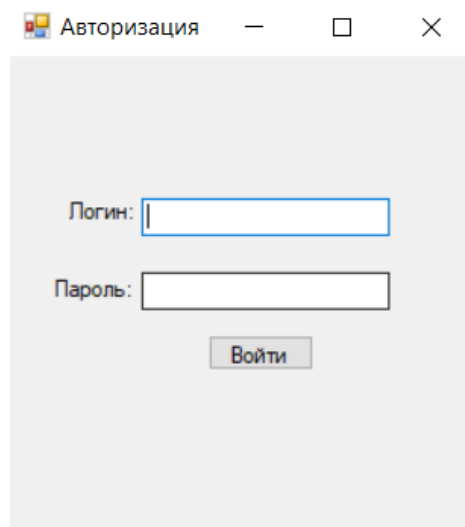


Рис. 6 – Окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, что показано на рисунках 7 и 8. Если логин и пароль правильные, то он попадает в рабочее окно, что показано рисунках 10-13.

The screenshot shows a window titled 'Авторизация' (Authorization) with standard Windows window controls (minimize, maximize, close). Inside the window, there are two input fields. The first is labeled 'Логин:' (Login) and contains the text 'admin@test.ru'. The second is labeled 'Пароль:' (Password) and contains a series of asterisks to mask the password. Below these fields is a button labeled 'Войти' (Log In).

Рис. 7 – Ввод данных от учетной записи администратора

This screenshot is similar to the previous one, showing the 'Авторизация' (Authorization) window. In this instance, the 'Логин:' (Login) field contains the text 'staff1@test.ru'. The 'Пароль:' (Password) field is also masked with asterisks. The 'Войти' (Log In) button is visible at the bottom.

Рис. 8 – Ввод данных от учетной записи сотрудника

Авторизация

Логин: parent1@test.ru

Пароль:

Войти

Рис. 9 – Ввод данных от учетной записи родителя

Поиск Выход

	FullName	DateOfBirth	Age	Gender	Benefits	School	Status
▶	Иван Иванов	10.05.2015	8	Мужской	Многодетная с...	Школа №1	Одобрено
	Мария Иванова	20.11.2017	6	Женский	Мать-одиночка	Гимназия №2	Одобрено

Имя ребенка Школа Дата рождения Льготы

Школа №1 28.12.2024

Пол Мужской

Подать заявку Редактировать Удалить

☐ Многодетная семья
☐ Инвалидность
☐ Мать-одиночка

Рис. 10 – Рабочее окно родителя

Панель администратора

Пользователи Родители

	Username	Login	Role
	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent

Имя пользователя Login Пароль Роль

Добавить Редактировать Удалить Просмотр отчетов

Рис. 11 – Рабочее окно администратора

Панель администратора

Пользователи Родители

	FullName	Email	Phone	Address
▶	Родитель Один	parent1@test.ru	89123456788	ул. Ленина, д. 10, кв. 25
	Родитель Два	parent2@test.ru	89876543210	ул. Мира, д. 5, кв. 12

Имя Email Телефон Адрес

Добавить Редактировать Удалить Выход

Рис. 12 – Рабочее окно администратора

Панель сотрудника

	ФИО ребенка	Возраст	Пол	Дата подачи	Школа	Статус	Класс	Льготы
▶	Иван Иванов	8	Мужской	01.07.2024	Школа №1	Одобрено	2	Многодетная ...
	Мария Ивано...	6	Женский	01.07.2024	Гимназия №2	Одобрено	1	Мать-одиночка
	Алексей Пет...	10	Мужской	02.07.2024	Школа №1	Одобрено	4	Инвалидность
	Алексей Серг...	10	Мужской	27.12.2024 18...	Гимназия №2	Одобрено	4	Многодетная ...

Имя

Школа

Льготы

Школа №1

Поиск

Создать отчет

Распределить

Сохранить изменения

Выход

Рис. 13 – Рабочее окно сотрудника

Рабочее окно родителя, показанное на рисунке 9 имеет следующие функции:

- 1) Текстовые поля для ввода данных заявки.
- 2) Поиск по имени ребенка.

Форма администратора, представленная на рисунке 11-12 состоит из нескольких страниц для каждой из таблиц в базе. Она имеет следующие функции:

- 1) Возможность просмотра необходимых данных, они показаны на рисунках 14 – 16.

Панель администратора

Пользователи Родители

	Username	Login	Role
	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent

Имя пользователя Login Пароль Роль

Добавить Редактировать Удалить Просмотр отчетов

Рис. 14 – Данные из таблицы Пользователи

Панель администратора

Пользователи Родители

	FullName	Email	Phone	Address
▶	Родитель Один	parent1@test.ru	89123456788	ул. Пенина, д. 10, кв. 25
	Родитель Два	parent2@test.ru	89876543210	ул. Мира, д. 5, кв. 12

	FullName	DateOfBirth	Age	Gender
	Иван Иванов	10.05.2015	8	Мужской
	Мария Иванова	20.11.2017	6	Женский

Имя Email Телефон Адрес

Добавить Редактировать Удалить Выход

Рис. 15 – Данные из таблицы Дети

Панель администратора

Пользователи Родители

	FullName	Email	Phone	Address
	Родитель Один	parent1@test.ru	89123456788	ул. Ленина, д. 10, кв. 25
▶	Родитель Два	parent2@test.ru	89876543210	ул. Мира, д. 5, кв. 12

	FullName	DateOfBirth	Age	Gender
▶	Алексей Петров	15.01.2014	10	Мужской
	Алексей Сергеев	15.01.2014	10	Мужской

Имя

Email

Телефон

Адрес

Добавить Редактировать Удалить

Выход

Рис. 16 – Данные из таблицы Дети

2) Каждая из страниц имеет текстовые поля для ввода данных, а также кнопки для их добавления, обновления или удаления. Для добавления данных в таблицу необходимо заполнить все поля, как показано на рисунке 17 и нажать на кнопку. Результат показан на рисунке 18.

Имя пользователя

Логин

Пароль

Роль

Добавить Редактировать Удалить Просмотр отчетов

Рис. 17 – Заполнение полей

Панель администратора

Пользователи Родители

	Username	Login	Role
▶	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent
	Родитель Три	parent3@test.ru	Parent

Имя пользователя: Логин: Пароль: Роль:

Добавить Редактировать Удалить Просмотр отчетов

Рис. 18 – Запись добавлена в таблицу

3) Для изменения данных необходимо выбрать запись, изменить данные и нажать на кнопку. Результат показан на рисунках 19-20.

Панель администратора

Пользователи Родители

	Username	Login	Role
	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent
▶	Родитель Три	parent3@test.ru	Parent

Имя пользователя: Логин: Пароль: Роль:

Добавить Редактировать Удалить Просмотр отчетов

Рис. 19 – Изменение данных

Панель администратора

Пользователи Родители

	Username	Login	Role
▶	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent
	Родитель Три	parent345@test.ru	Parent

Имя пользователя Логин Пароль Роль

Добавить Редактировать Удалить Просмотр отчетов

Рис. 20 – Запись обновляется

4) Для удаления данных необходимо также выбрать запись и нажать на кнопку удаления. Результат представлен на рисунках 21-22.

Панель администратора

Пользователи Родители

	Username	Login	Role
	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent
▶	Родитель Три	parent345@test.ru	Parent

Имя пользователя Логин Пароль Роль

Родитель Три parent345@test.ru parentpass3 Parent

Добавить Редактировать Удалить Просмотр отчетов

Рис. 21– Выбор записи

Панель администратора

Пользователи Родители

Username	Login	Role
Админ Админов	admin@test.ru	Admin
Сотрудник	staff1@test.ru	Staff
Родитель Один	parent1@test.ru	Parent
Родитель Два	parent2@test.ru	Parent

Имя пользователя: Логин: Пароль: Роль:

Добавить Редактировать Удалить Просмотр отчетов

Рис. 22 - Запись удалена

3.3 Руководство программиста

3.3.1 Введение

Приложение разработано на языке программирования C# в среде Visual Studio с использованием СУБД SQLite. Программа включает в себя несколько форм: форма авторизации, рабочая форма администратора, рабочая форма студента.

3.3.2 Установка и настройка

Необходимые компоненты:

- Visual Studio 2019 или новее
- .NET Framework 4.7.2 или новее
- СУБД SQLite

Установка:

- 1) Откройте Visual Studio

- 2) Запустите проект
- 3) Установите пакет SQLite через пункт управление NuGet пакетами

3.3.3 Структура приложения

Проект состоит из следующих файлов:

- LoginForm.cs – Форма авторизации пользователей
- AdminForm.cs – Форма администратора для управления

пользователями и родителями

- ParentForm.cs – Форма родителя для управления заявками детей
- ParentForm.cs – Форма сотрудника для управления распределением

детей

3.3.4 Описание форм

1) LoginForm.cs

- Конструктор LoginForm: инициализирует форму.
- Метод loginButton_Click: обрабатывает нажатие кнопки входа, проверяет введенные логин и пароль, и открывает соответствующую форму (AdminForm, StaffForm или ParentForm) в зависимости от роли пользователя.

2) AdminForm.cs

- Конструктор AdminForm: инициализирует форму и элементы управления, включая заполнение roleComboBox ролями пользователей.
- Метод AdminForm_Load: загружает данные о пользователях и родителях при открытии формы.
- Метод LoadUsers: загружает список пользователей в usersDataGridView.
- Метод LoadParents: загружает список родителей в parentsDataGridView.
- Метод addUserButton_Click: обрабатывает добавление нового пользователя.

					МИВУ 10.03.01-19ПЗ	Лист
						31
Изм.	Лист	№ докум.	Подпись	Дата		

- Метод editUserButton_Click: обрабатывает редактирование существующего пользователя.
- Метод deleteUserButton_Click: обрабатывает удаление пользователя.
- Метод addParentButton_Click: обрабатывает добавление нового родителя.
- Метод editParentButton_Click: обрабатывает редактирование данных родителя.
- Метод deleteParentButton_Click: обрабатывает удаление родителя.
- Метод viewReportsButton_Click: обрабатывает просмотр отчетов.
- Метод backButton_Click: возвращает к форме авторизации.

3) ParentForm.cs

- Конструктор ParentForm: инициализирует форму и получает ID родителя.
- Метод ParentForm_Load: загружает данные о детях, школах и льготах при открытии формы.
- Метод LoadChildren: загружает список детей родителя.
- Метод LoadSchools: загружает список доступных школ.
- Метод LoadBenefits: загружает список доступных льгот.
- Метод applyButton_Click: обрабатывает подачу новой заявки на распределение ребенка.
- Метод editButton_Click: обрабатывает редактирование данных ребенка и его заявки.
- Метод deleteButton_Click: обрабатывает удаление данных о ребенке и его заявки.
- Метод searchButton_Click: обрабатывает поиск детей по имени.
- Метод ValidateChildAge: проверяет корректность возраста ребенка.
- Метод TranslateStatus: переводит статус заявки с английского на русский.
- Метод backButton_Click: возвращает к форме авторизации.

- Метод childrenDataGridView_SelectionChanged: обрабатывает выбор ребенка в таблице и заполняет поля формы его данными.

4) StaffForm.cs

- Конструктор StaffForm: инициализирует форму для сотрудника школы.
- Метод StaffForm_Load: загружает данные при открытии формы
- Метод LoadApplications: загружает список всех заявок с информацией

о детях и школах

- Метод LoadSchools: загружает список всех школ
- Метод LoadClasses: загружает доступные классы для распределения
- Метод approveButton_Click: обрабатывает одобрение заявки
- Метод rejectButton_Click: обрабатывает отклонение заявки
- Метод filterApplications_Click: применяет выбранные фильтры к

списку заявок:

- Метод checkAvailablePlaces: проверяет наличие свободных мест в выбранной школе
- Метод generateReport_Click: формирует отчет о распределении детей

по школам

- Метод searchApplications_TextChanged: осуществляет поиск по заявкам в реальном времени

- Метод applicationGridView_SelectionChanged: обрабатывает выбор заявки в таблице

- Метод updateApplicationStatus: обновляет статус заявки в базе данных
- Метод validateDistribution: проверяет корректность распределения
- Метод backButton_Click: возвращает к форме авторизации
- Метод refreshData: обновляет данные на форме

3. Тестирование АИС

Тестирование автоматизированной информационной системы является неотъемлемой частью процесса разработки, направленной на обеспечение ее корректной работы, надежности и безопасности. При открытии приложения первым шагом для пользователя является окно авторизации, показанное на рисунке 23, которое служит для идентификации и аутентификации пользователей. Это окно играет важную роль в защите данных и управлении доступом к функционалу системы, что делает его критически важным элементом для тестирования.

Рис. 23 – Окно авторизации

В окне авторизации пользователь вводит логин и пароль от своей учетной записи, если логин и пароль правильные, то пользователь попадает в рабочее окно, а если логин или пароль, который был введен не корректен, то приложение выдаст ошибку, что логин и пароль не правильный, это показано на рисунке 24.

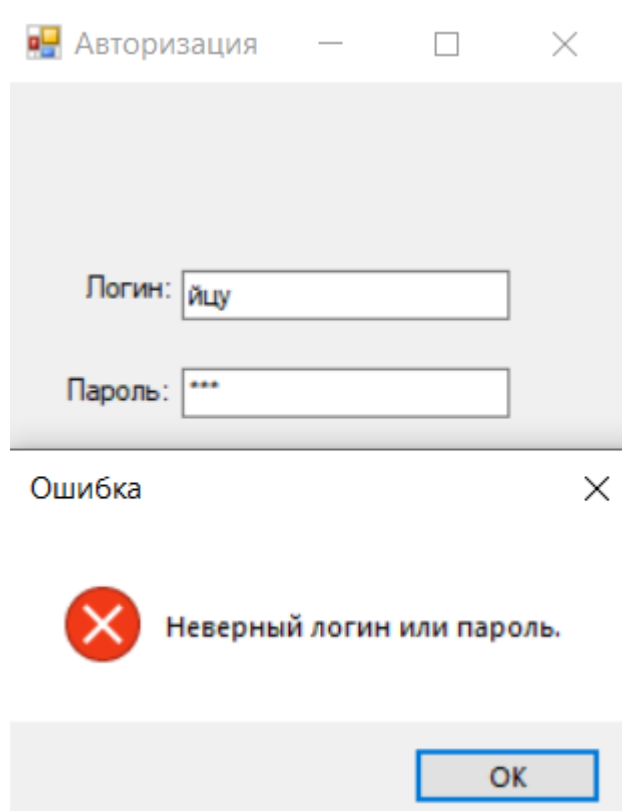


Рис. 24 – Ошибка входа при неправильном логине или пароле

Если логин и пароль правильные, то пользователь попадает в одно из рабочих окон. Начнем с формы администратора. Здесь администратор может управлять другими пользователями и более подробно управлять родителями и просматривать отчеты о распределении.

Если администратор попытается удалить сам себя, то будет выведено сообщение. Результат представлен на рисунках 25-26

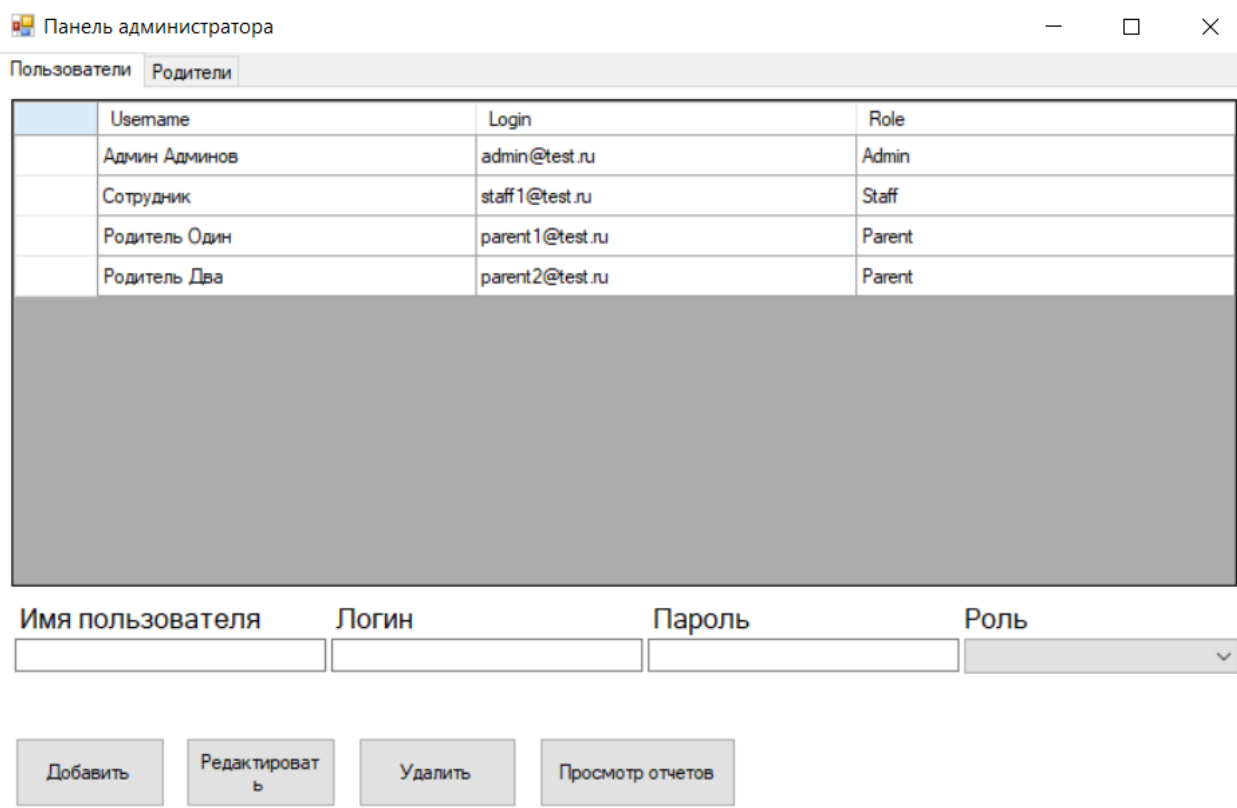


Рис. 25 – Корректное удаление

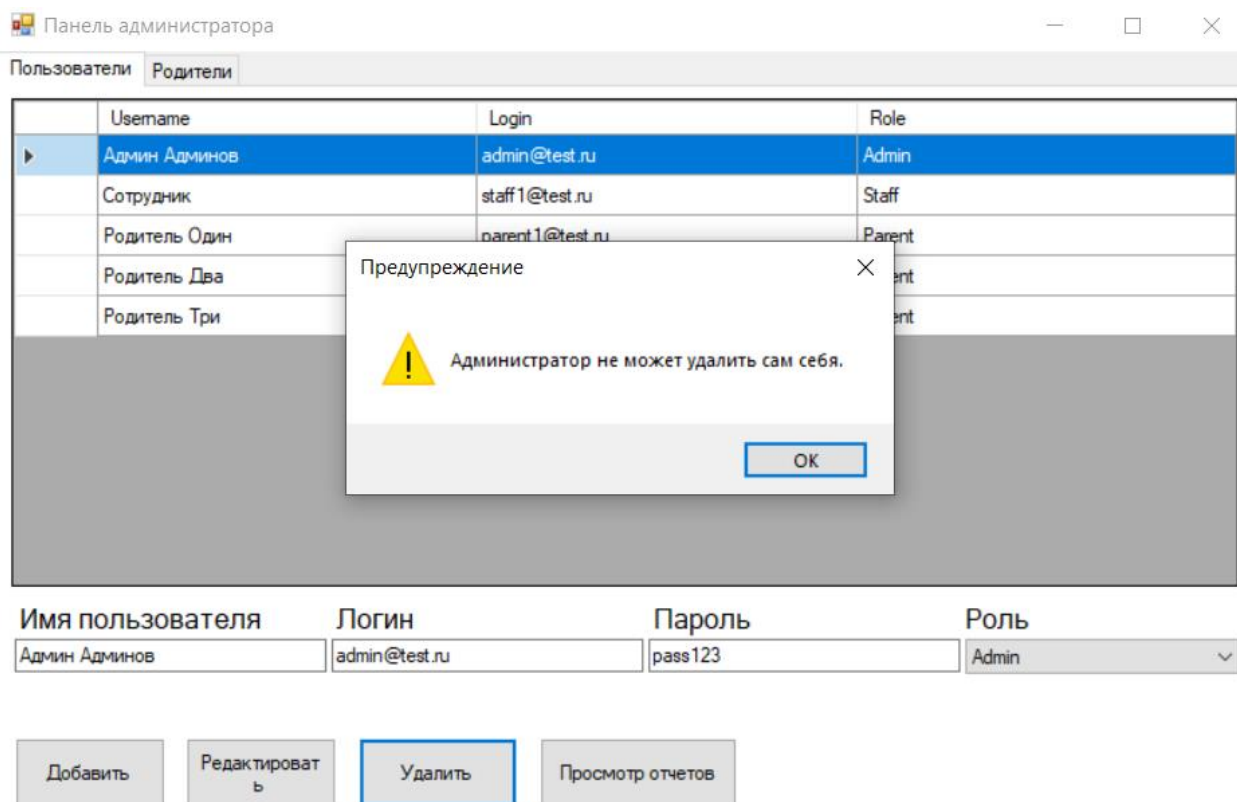


Рис. 26 – Попытка удалить администратора

Если администратор попытается добавить пользователя, не заполнив поля, появится сообщение, результат на рисунке 27.

Панель администратора

Пользователи Родители

	Username	Login	Role
	Админ Админов	admin@test.ru	Admin
	Сотрудник	staff1@test.ru	Staff
	Родитель Один	parent1@test.ru	Parent
	Родитель Два	parent2@test.ru	Parent
▶	Родитель Три		

Предупреждение

⚠ Заполните все обязательные поля: Логин, Пароль, Роль, Имя.

ОК

Имя пользователя Логин Пароль Роль

Добавить Редактировать Удалить Просмотр отчетов

Рис. 27 – Попытка заполнения с пустыми полями.

Теперь перейдем к форме родителя. Он может добавить заявку на распределение своего ребенка в школу. Если он не заполнит обязательные поля, появится сообщение, результат приведен на рисунках 28-29.

ParentForm

Поиск Выход

	FullName	DateOfBirth	Age	Gender	Benefits	School	Status
▶	Алексей Петров	15.01.2014	10	Мужской	Инвалидность	Школа №1	Одобрено
	Алексей Сергеев	15.01.2014	10	Мужской	Многодетная с...	Гимназия №2	Одобрено
	Антон Сибирский	15.01.2014	10	Мужской	Многодетная с...	Гимназия №2	На рассмотрен...

Имя ребенка Школа Дата рождения Льготы

Пол

Подать заявку Редактировать Удалить

☐ Многодетная семья
☐ Инвалидность
☐ Мать-одиночка

Рис. 28 – Успешная попытка заполнения.

ParentForm

Поиск Выход

	FullName	DateOfBirth	Age	Gender	Benefits	School	Status
▶	Алексей Петров	15.01.2014	10	Мужской	Инвалидность	Школа №1	Одобрено
	Алексей Сергеев	15.01.2014	10	Мужской	Многодетная с...	Гимназия №2	Одобрено

Предупреждение

⚠ Заполните все обязательные поля

OK

Имя ребенка Школа

Гимназия №2 15.01.2014

Пол Мужской

Подать заявку Редактировать Удалить

- ☒ Многодетная семья
- ☐ Инвалидность
- ☐ Мать-одиночка

Рис. 29 – Попытка заполнения с неполным количеством полей.

Теперь перейдем к панели сотрудника. Сотрудник может управлять распределением детей по школам. При наличии заявок на распределение будет выведено сообщение о корректном распределении. Но если заявок нет, будет выведено сообщение о отсутствии заявок для распределения, результат представлен на рисунках 30-31.

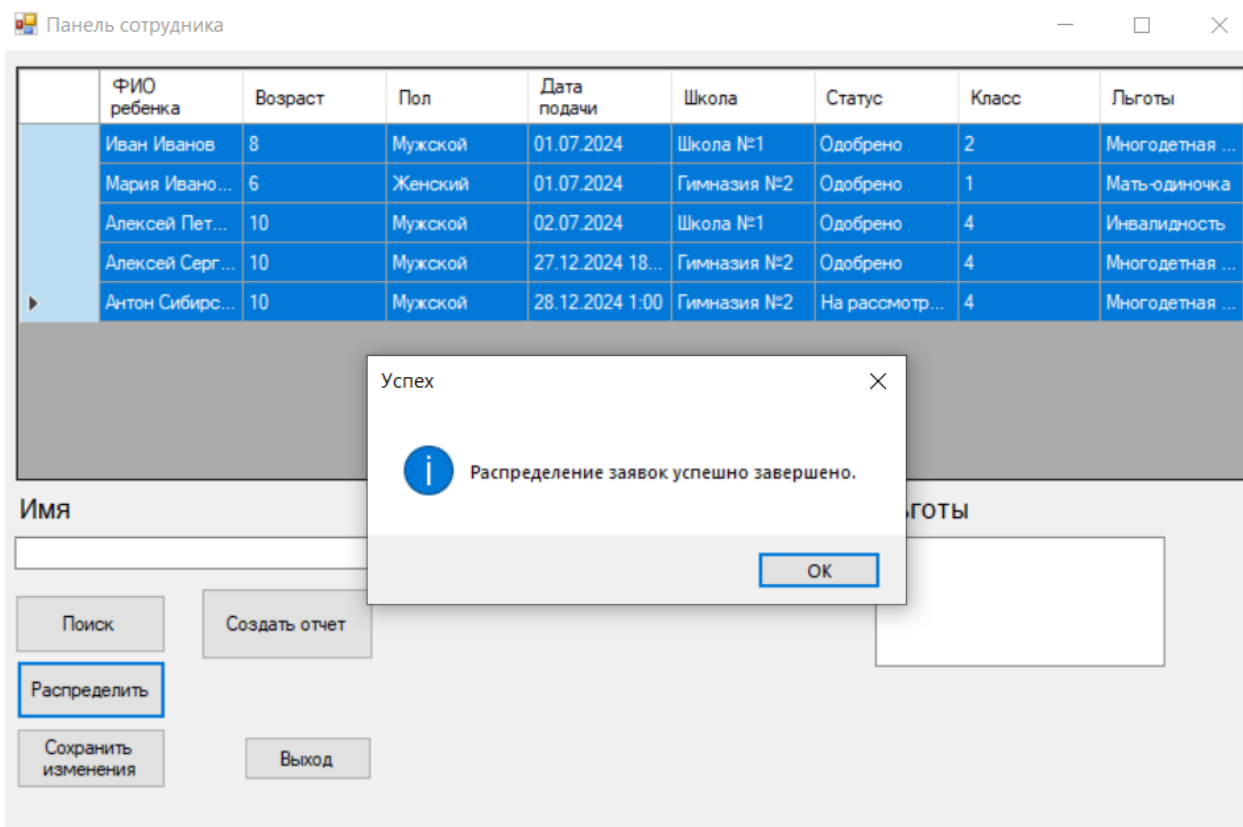


Рис. 29 – Успешное распределение.

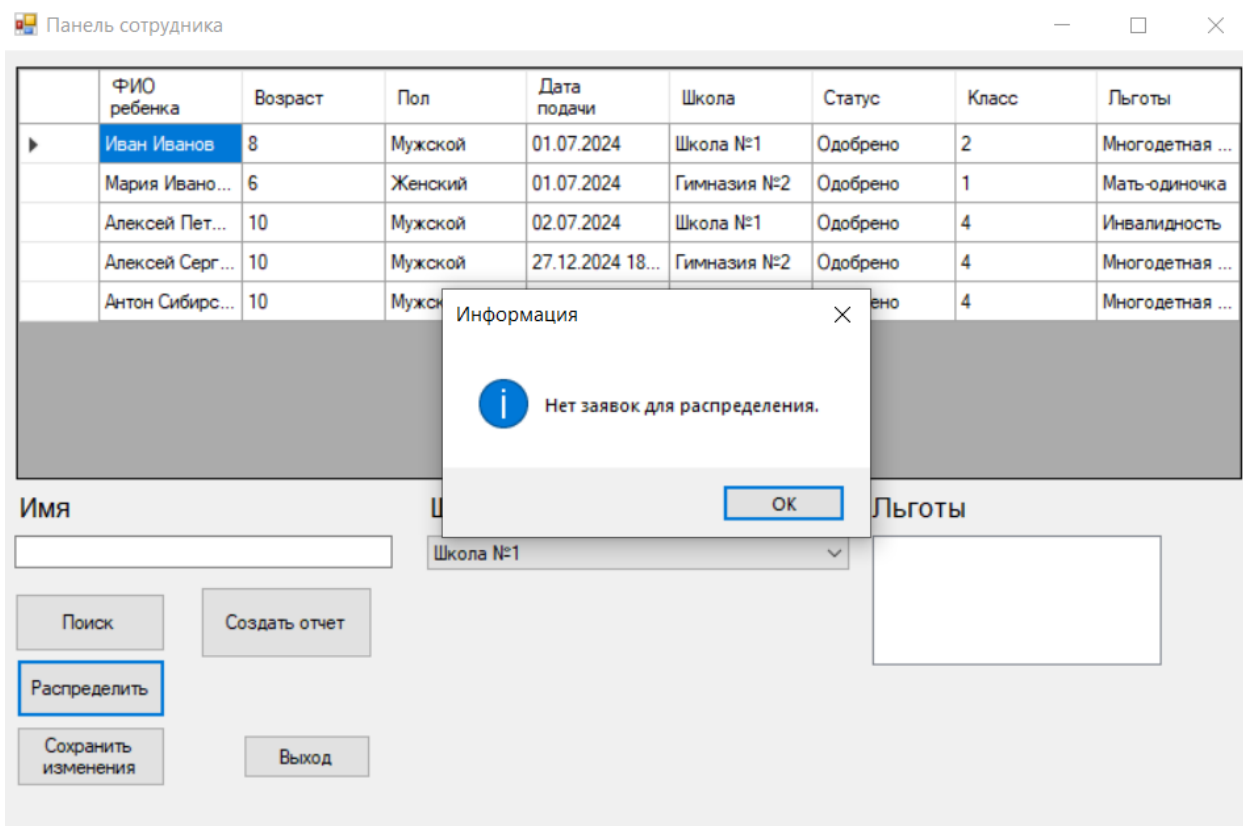


Рис. 29 – Попытка распределения без наличия заявок.

Заключение

В ходе выполнения данной курсовой работы были успешно решены поставленные задачи по созданию Автоматизированной Информационной Системы (АИС) распределения детей в школы. Основной акцент был сделан на разработке эффективной базы данных и удобного пользовательского интерфейса, обеспечивающих комплексное управление процессом распределения учащихся.

Разработанная система позволяет автоматизировать следующие ключевые процессы:

- подача и обработка заявлений на зачисление в школы
- учет льготных категорий учащихся
- контроль количества свободных мест в школах
- управление данными о детях, родителях и школах

Использование современных технологий, таких как Entity Framework Core и SQLite, в сочетании с Windows Forms, позволило создать надежное и производительное решение. Многопользовательский интерфейс с разграничением прав доступа (администратор, сотрудник, родитель) обеспечивает удобную работу всех категорий пользователей системы.

Разработанная АИС существенно оптимизирует процесс распределения детей в школы, делая его более прозрачным и эффективным. Система готова к масштабированию и может быть дополнена новыми функциями в соответствии с растущими потребностями образовательных учреждений.

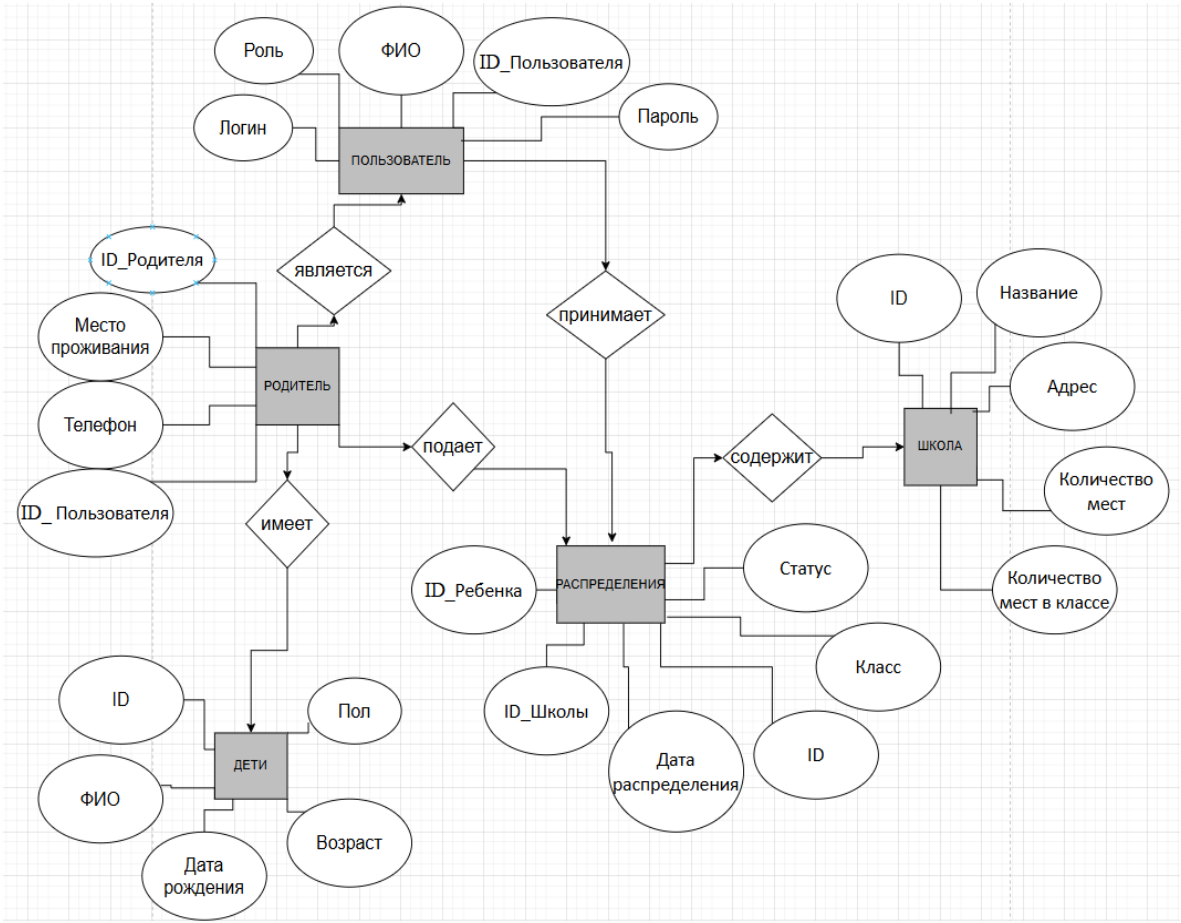
					МИВУ 10.03.01-19ПЗ	Лист
						40
Изм.	Лист	№ докум.	Подпись	Дата		

Список литературы

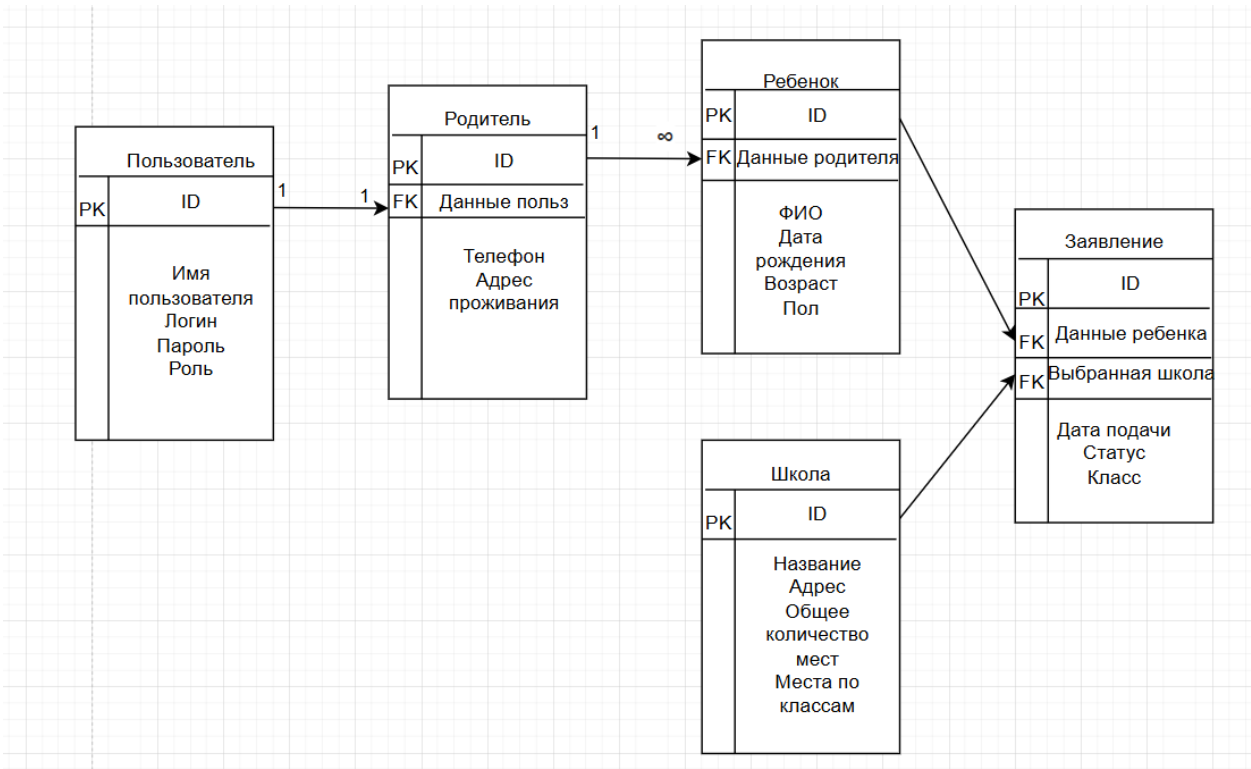
- 1) Бьерн Страуструп, Язык программирования C++. Специальное издание. Пер. с англ. – М.: Издательство Бином, 2011 г. – 1136 с.: ил.
- 2) Казарин С.А., Клишин А.П. К 143 Среда разработки Java-приложений Eclipse: (ПО для объектно-ориентированного программирования и разработки приложений на языке Java): Учебное пособие. Москва 2008. — 77 с.
- 3) Медведев, М. А. М42 Программирование на СИ# : учеб. пособие / М. А. Медведев, А. Н. Медведев. — Екатеринбург : Изд-во Урал. ун-та, 2015. — 64 с.
- 4) Мотев А. А. М85 Уроки MySQL. Самоучитель. — СПб.: БХВ-Петербург, 2006. — 208 с.: ил.
- 5) Рындина С. В.. – Пенза : Изд-во ПГУ, 2023. – 82 с. , Цифровые технологии управления получением, хранением, передачей и обработкой больших данных: SQLite : учеб.-метод. пособие.
- 6) Сапаров А.Ю., Разработка Windows Forms приложений на языке программирования C#: учебно-методическое пособие / Сост.: А.Ю. Сапаров, Ижевск, 2020. 61 с.
- 7) Сергеева Т.И. Базы данных: модели данных, проектирование, язык SQL: учеб. пособие / Т.И. Сергеева, М.Ю. Сергеев. Воронеж: ФГБОУ ВПО «Воронежский государственный технический университет», 2012. 233 с.

Приложение А: Модели данных

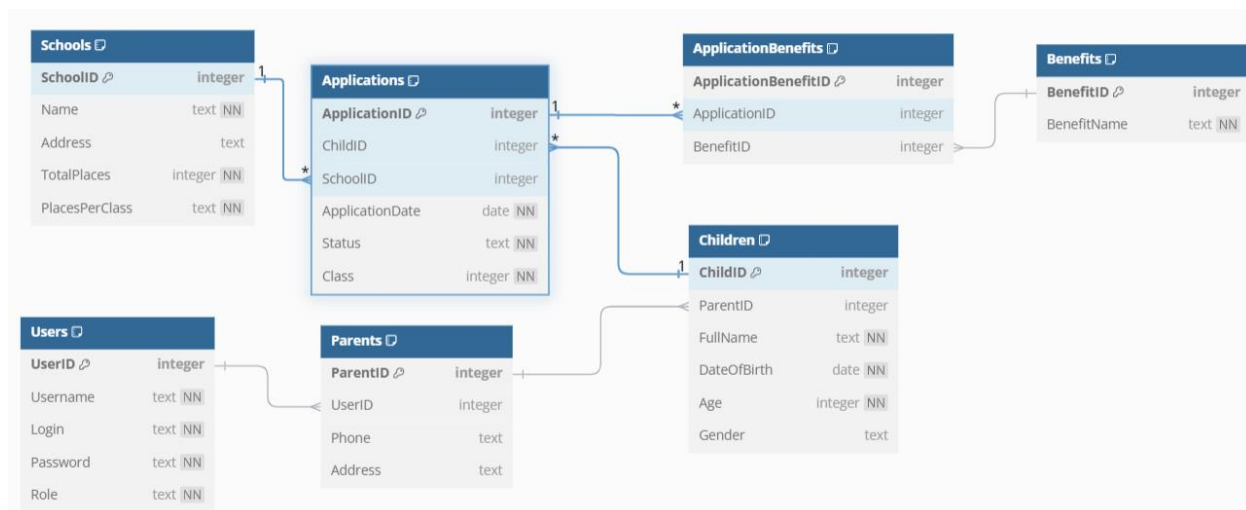
Концептуальная модель.



Логическая модель.



Физическая модель.



Приложение Б: Текст кода

Весь исходный код представлен по ссылке:

<https://github.com/FastFoodHero/Kursovaya-BD>