Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

ИРКУТСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

Институт Информационных Технологий и Анализа Данных

Кафедра вычислительной техники

**Название работы** – “Web - приложение”

Отчет по лабораторной работе “Лабораторная работа №3”

по дисциплине Управление данными

Вариант Свой

Выполнил

Студент, номер группы ИСМб-19-1 \_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.А. Малиновцев

Дата: \_\_\_\_\_\_\_\_\_\_

Принял

Должность \_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.А.Харахинов

Дата: \_\_\_\_\_\_\_\_\_\_

Иркутск 2021 г.

Содержание

[Индивидуальное задание 3](#_Toc70136341)

[Клиентская часть приложения 4](#_Toc70136342)

[Интерфейс клиентской части приложения 4](#_Toc70136343)

[Обработка ошибок 9](#_Toc70136344)

[Исходный код 12](#_Toc70136345)

[Серверная часть приложения 23](#_Toc70136346)

[Реализация базы данных 23](#_Toc70136347)

[Взаимодействие сервера с базой данных 28](#_Toc70136348)

[Обработка ошибок 31](#_Toc70136349)

[Работа с файлами Excel 34](#_Toc70136350)

[Исходный код 38](#_Toc70136351)

Индивидуальное задание

**Задание**: Разработать клиент-серверное приложение, которое реализует решение задачи индивидуального варианта.

Клиентская часть приложения

Интерфейс клиентской части приложения

Для реализации клиентской части приложения был использован язык C# со средой разработки Visual Studio Community, с возможностью взаимодействия с базой данных, посредством веб-форм.

Интерфейс приложения предоставляет возможность взаимодействия с тремя таблицами: таблицей людей, профессий, компаний, работников и работы, а также с решением задач индивидуального варианта.

Для переключения между таблицами используются кнопки переходов.

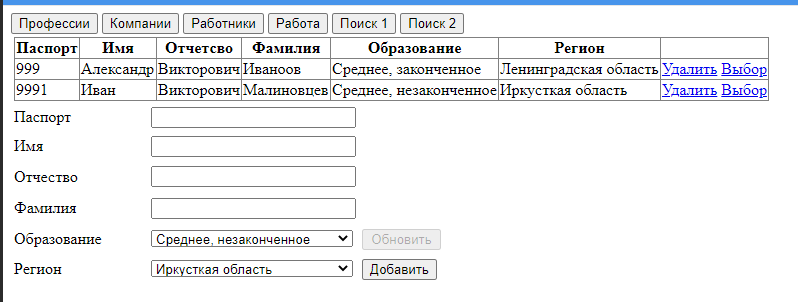


Рисунок 1 – Интерфейс для взаимодействия с таблицей людей

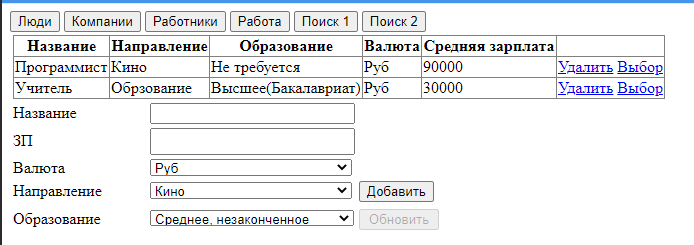


Рисунок 2 – Интерфейс для взаимодействия с таблицей профессий

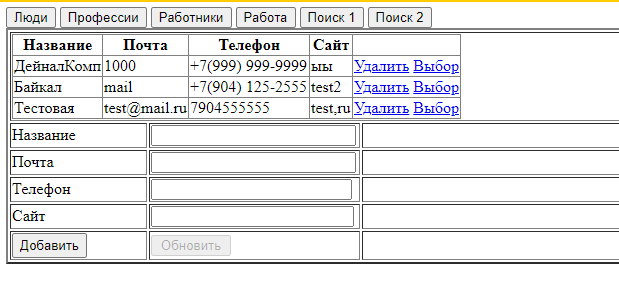


Рисунок 3 – Интерфейс взаимодействия с таблицей компаний

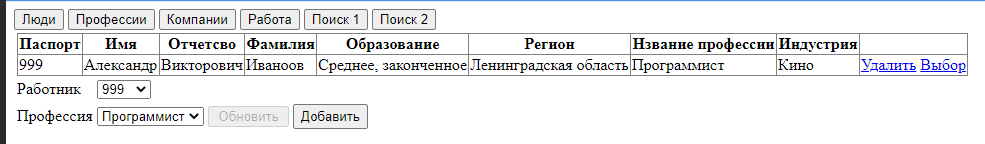


Рисунок 4 – Интерфейс взаимодействия с таблицей работников

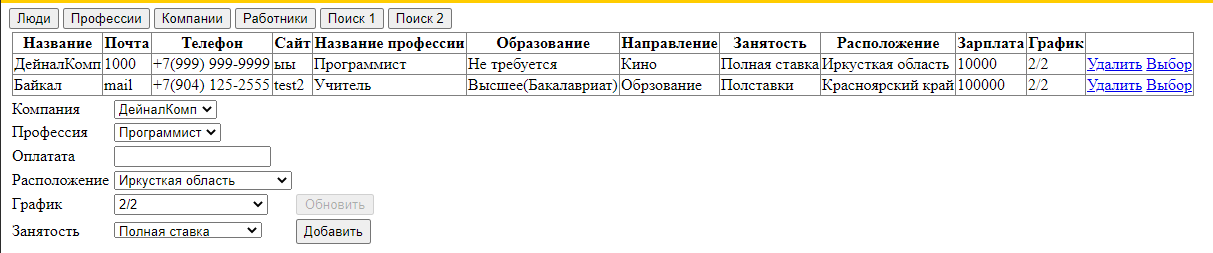


Рисунок 5– Интерфейс взаимодействия с таблицей работы

С помощью каждой веб-формы можно манипулировать поведением Insert, Delete, Update и Select каждой таблицы. Каждая веб-форма реализует свою логику взаимодействия с той таблицей базы данных, к которой она привязана. Данные из базы данных привязываются посредством указания в DataSource каждого элемента управления GridView хранимой процедуры, осуществляющей возврат данных из соответствующей таблицы (Select). Имеется возможность добавления, правки и удаления каждой записи соответствующей таблицы.

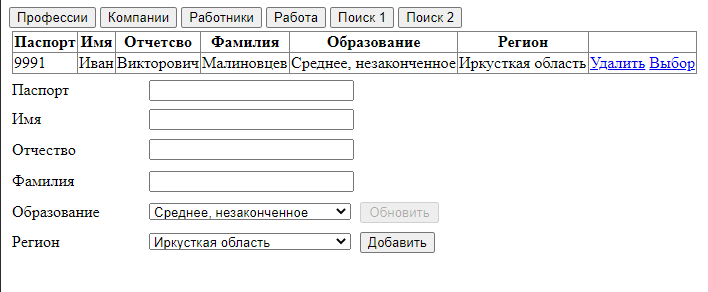


Рисунок 6 – Таблица Людей, после удаление записи с паспортными данными 999 (сравните с рисунком 1)

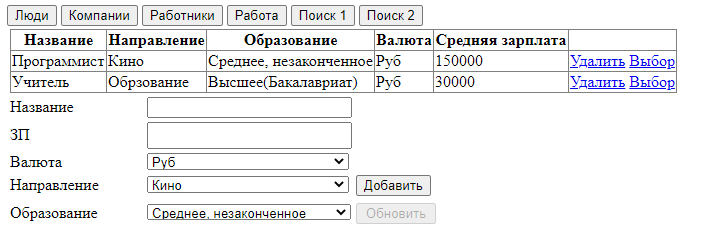


Рисунок 7 – Таблица профессий после изменения записи с именем Программист (сравните с рисунком 2)

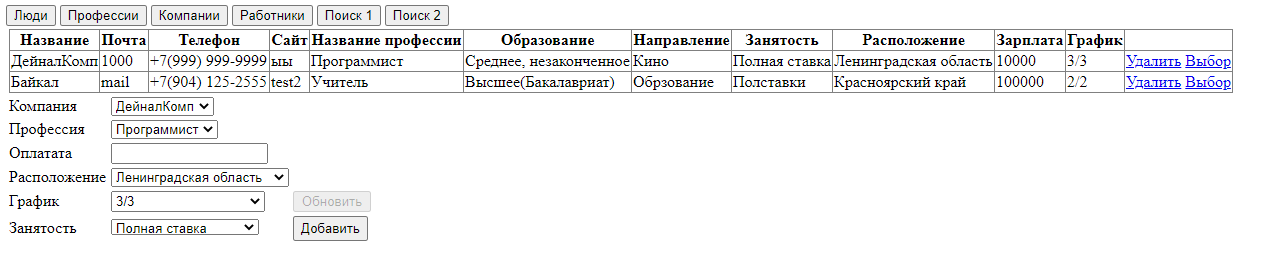


Рисунок 8 – Таблица Работы после изменения записи с названием Программист и Компанией ДэйналКомп(сравните с рисунком 3)

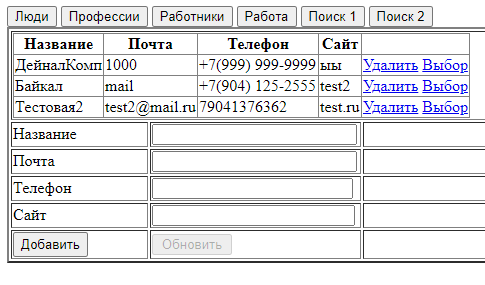


Рисунок 8 – Добавление данных в таблицу Компаний

Для решения задач индивидуального варианта, необходимо перейти в форму таблицы Поиск 1 или Поиск 2 и осуществить ввод данных в поля ввода.

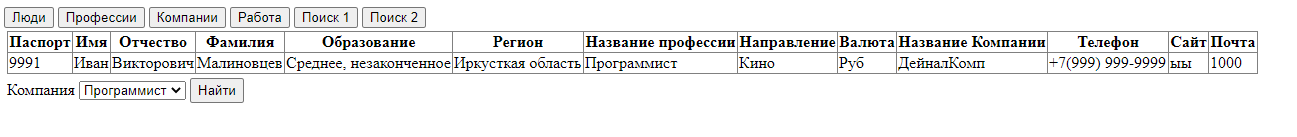


Рисунок 10 – Решение задачи индивидуального варианта

Обработка ошибок

Для контролирования процесса ввода и проверки входных значений были использованы такие средства, как RangeValidator, RequiredFieldValidator, RegularExpressionValidator и вставки скрипта на языке программирования JavaScript.

Вставки кода на JavaScript, не позволяют вводить неправильные наборы данных, посредством запрета ввода определённых символов (например, ввод только цифр, без символов). Валидаторы ASP.NET работают другим образом: они позволяют вводить любые символы, однако при изменении фокуса мышки или нажатии на одну из кнопок управления пользователь получает сообщение об ошибке и происходит игнорирование текущей операции (добавление или обновление данных).

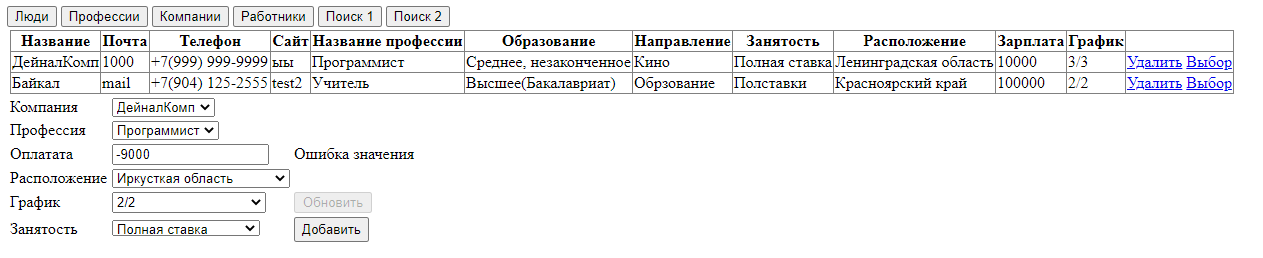


Рисунок 11 – Некорректный ввод данных и попытка добавления (не успешная)

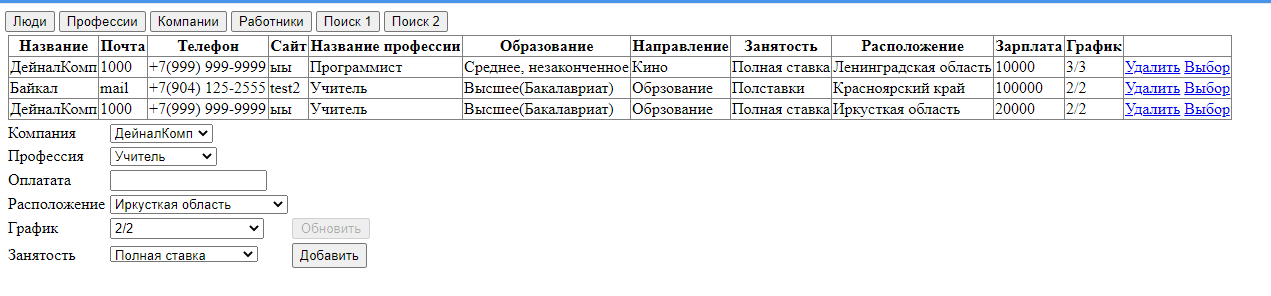


Рисунок 12 – Корректный ввод данных и попытка добавления данных (успешная попытка)

Взаимодействие сервера с базой данных

Для взаимодействия клиентской части с базой данных, хранящейся на сервере, был использован элемент управления GridView, который позволяет обрабатывать основные команды для взаимодействия с базой данных: Select, Insert, Delete и Update.

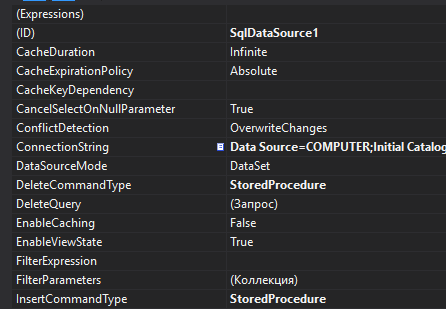


Рисунок 13 – Свойства экземпляра объекта SqlDataSource, для веб-формы WebFormReader

Для контроля передачи корректных значений в хранимые процедуры INSERT, UPDATE и DELETE, был использован метод вмешательства в передаваемые параметры хранимым процедурам, автоматизированной технологией ASP.NET, посредством внедрения кода перед событием вызова команды Insert(), Update() и Delete() соответственно.

Обработка ошибок

Хранимые процедуры возвращают значение и поведение приложение в зависимости от этих возвращаемых значений должно быть разным, поскольку учтены различные ситуации. Например, контроль невозможности добавления в таблицу читателей записи с паспортными данными, которые в этой таблице уже есть.

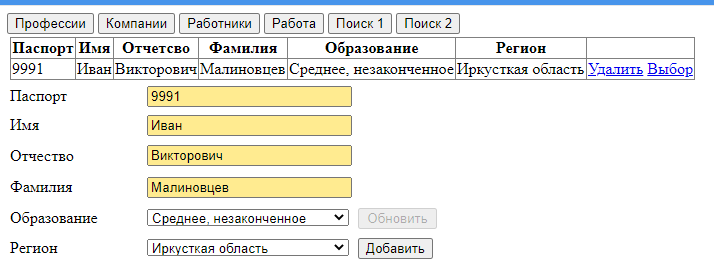


Рисунок 13 – Форма Default (до попытки добавления записи)

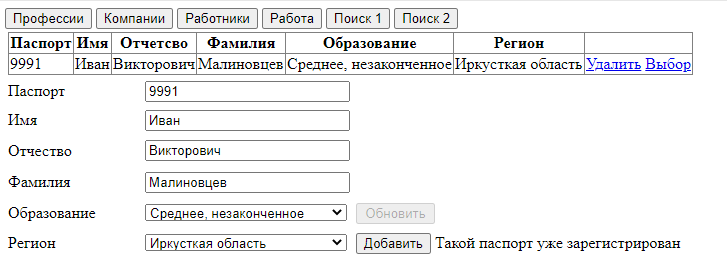


Рисунок 14 – Форма с выводом ошибки на экран