[Введение 4](#_Toc183715569)

[1. Схема данных 4](#_Toc183715570)

[2. Схема пользователей 5](#_Toc183715571)

[3. Интерфейс приложения 5](#_Toc183715572)

[4. Диаграммы последовательностей 6](#_Toc183715573)

[5. Концептуальная модель данных 9](#_Toc183715574)

[6. Процесс нормализации 10](#_Toc183715575)

[7. Логическая модель данных 13](#_Toc183715576)

[8. Физическая модель данных 14](#_Toc183715577)

[9. Описание базы данных на сервере 15](#_Toc183715578)

[10. Вывод данных на экран 16](#_Toc183715579)

[11. Таблица тестов 22](#_Toc183715580)

[Заключение 29](#_Toc183715581)

# Введение

Курсовой проект заключается в проектировании базы данных, в которой будут храниться списки прокатов, клиентов, фильмов, киностудий и жанров.

В процессе работы будут получены знания и навыки по проектированию баз данных, написанию приложения, работающего напрямую с базами данных. Также будут улучшены навыки отладки приложений. Так как работа осуществляется средствами web, будут улучшены навыки работы с этой технологией. Будут улучшены навыки работы с SQL.

В качестве web-сервера будет использована кроссплатформенная сборка XAAMP, содержащая Apache и MySQL. Для создания как самой базы данных, так и таблиц в ней, будет использоваться СУБД phpMyAdmin, представляющая из себя web-приложение с открытым кодом. Так же в работе задействован сервер Express и Node.js.

Для выполнения данного проекта были поставлены такие задачи:

* составить объектную модель задачи;
* составить логическую и физическую модель данных;
* создать страницы с отображением данных из базы данных с возможностью добавления и удаления записей;
* провести тестирование реализованного WEB-приложения;
* составить отчёт по выполненному курсовому проекту.

# Схема данных

На рис. 1 показаны данные, которые будут представлены в базе данных.

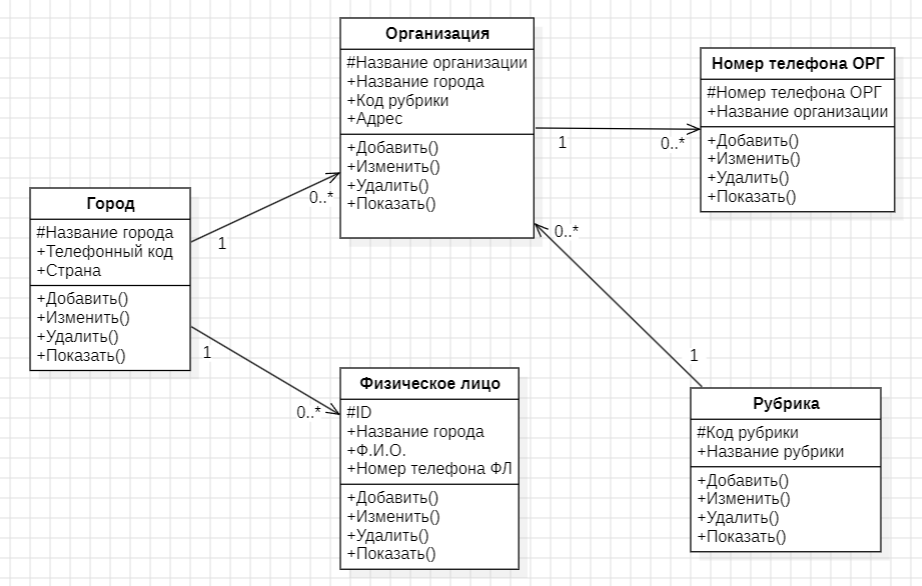


Рисунок 1 – Схема данных для телефонного справочника

# Схема пользователей

Рассмотрим пользователей БД, показанных на рис. 2. Ими являются Пользователь и Диспетчер справочника.

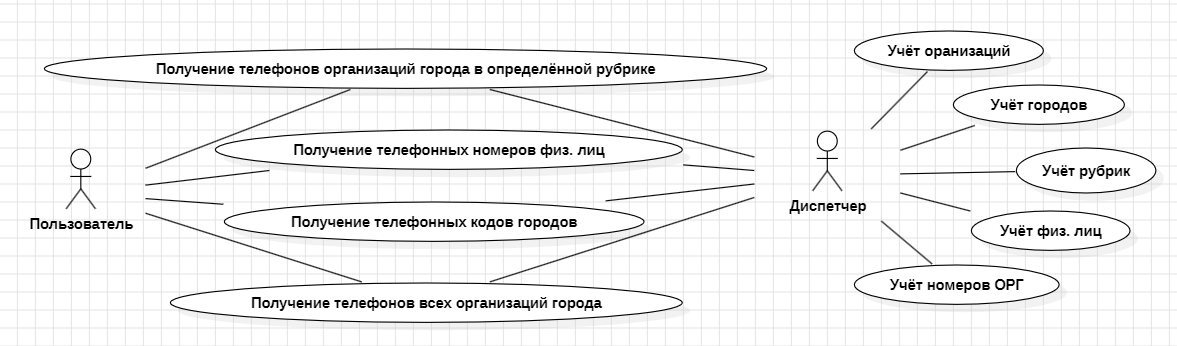


Рисунок 2 – Схема данных для телефонного справочника

# Интерфейс приложения

На рис. 3 представлен интерфейс WEB-приложения.

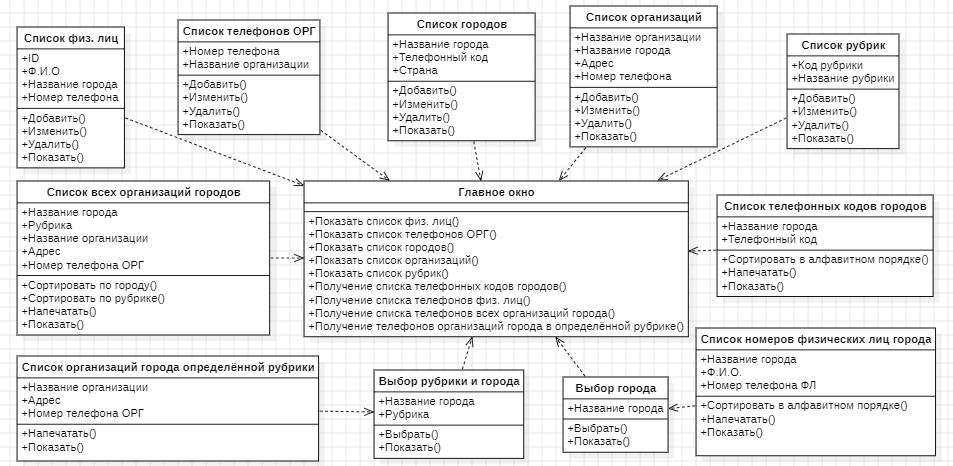


Рисунок 3 – Схема данных для телефонного справочника

# Диаграммы последовательностей

Необходимо сделать диаграммы последовательностей для более наглядного взаимодействия пользователей с интерфейсом и БД (рис. 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9).

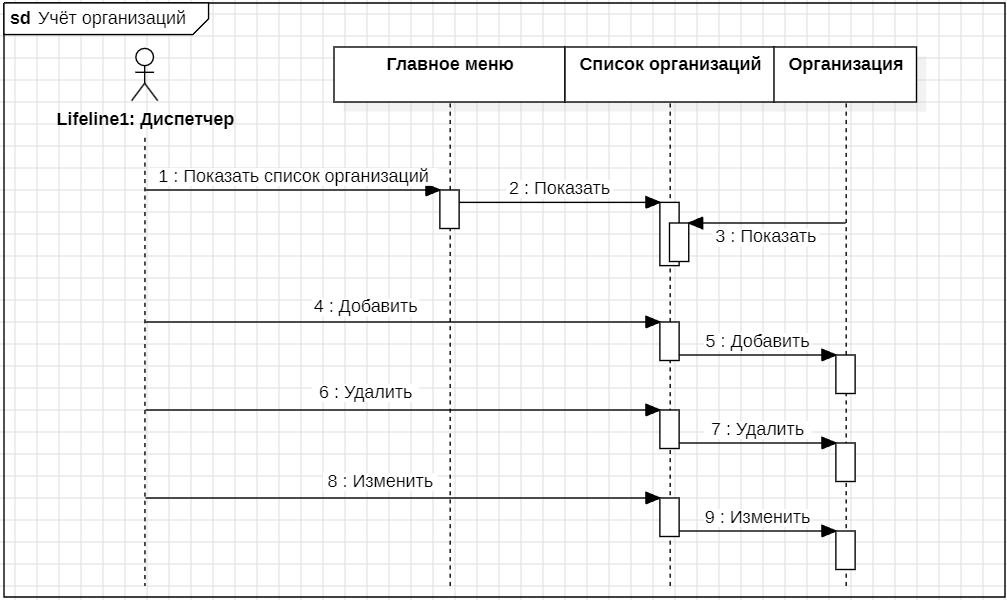


Рисунок 4.1 – Учёт организаций

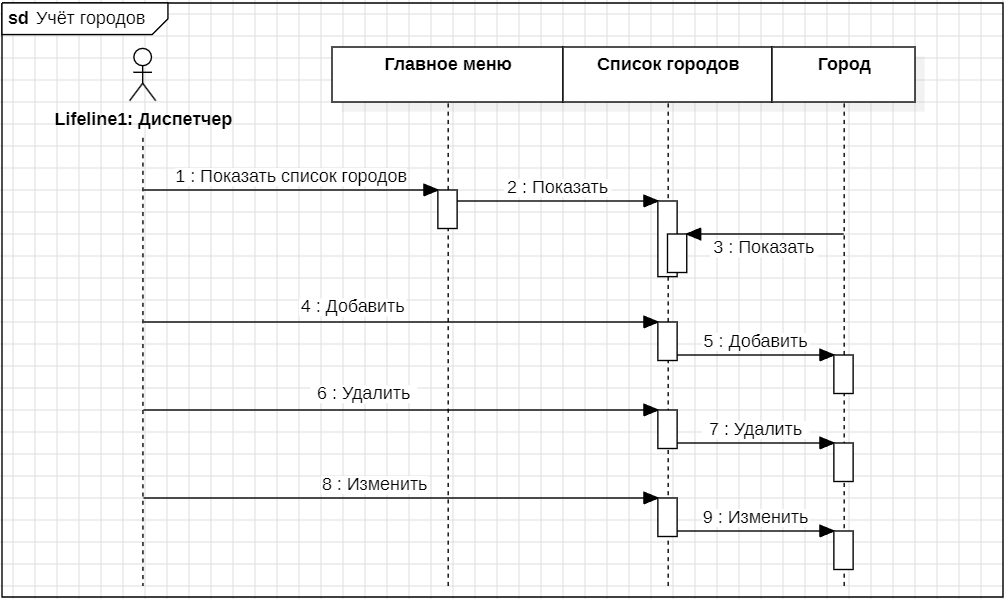


Рисунок 4.2 – Учёт городов

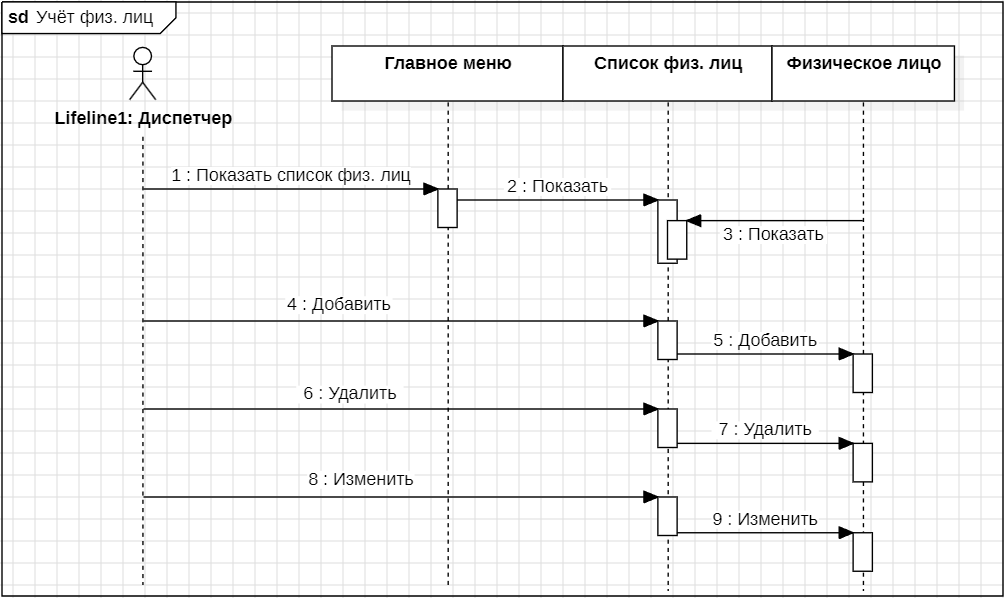


Рисунок 4.3 – Учёт физических лиц

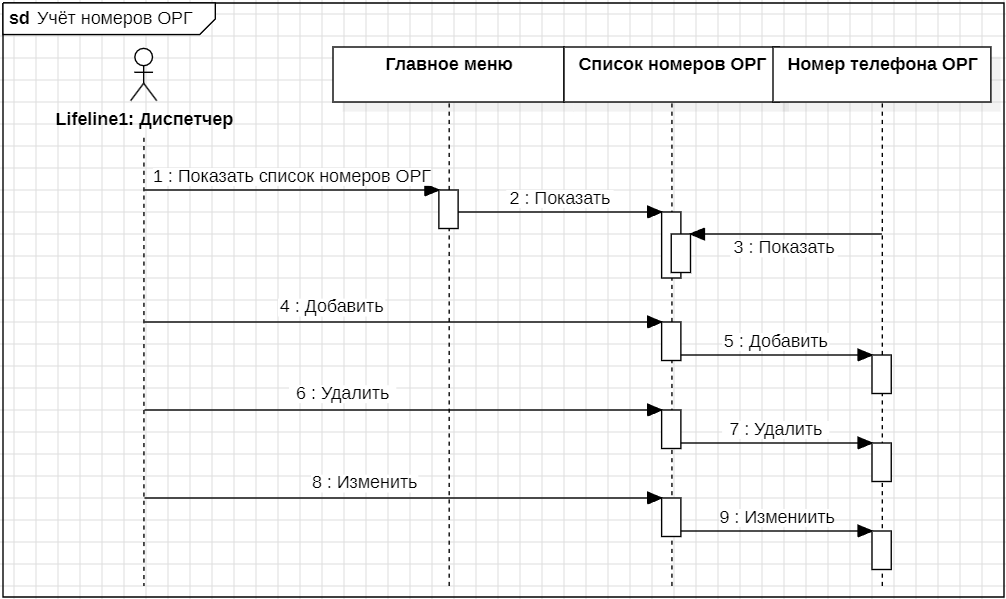


Рисунок 4.4 – Учёт номеров ОРГ

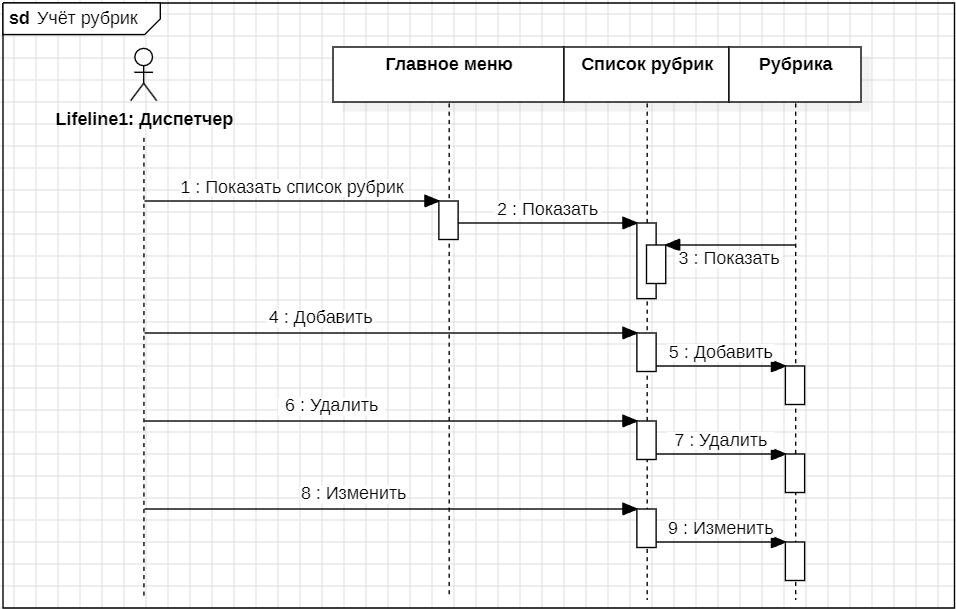


Рисунок 4.5 – Учёт рубрик

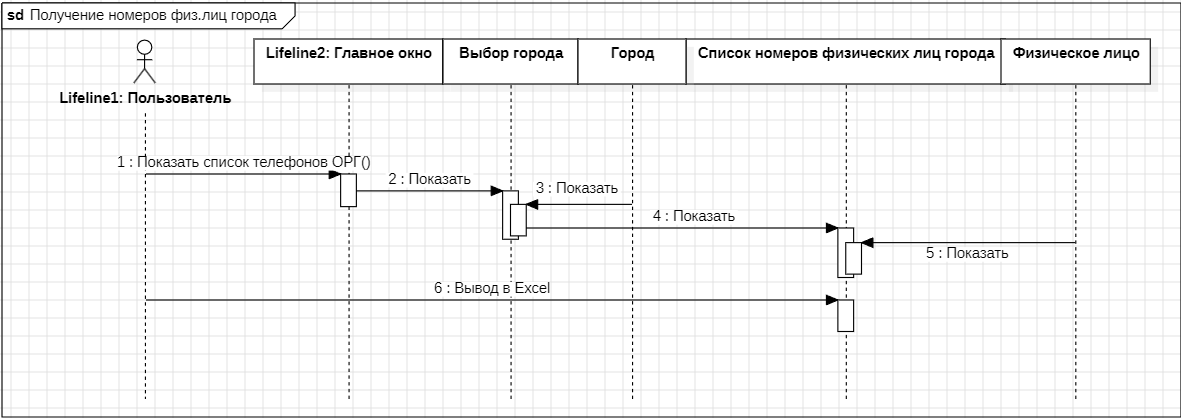


Рисунок 4.6 – Получение списка номеров физических лиц города

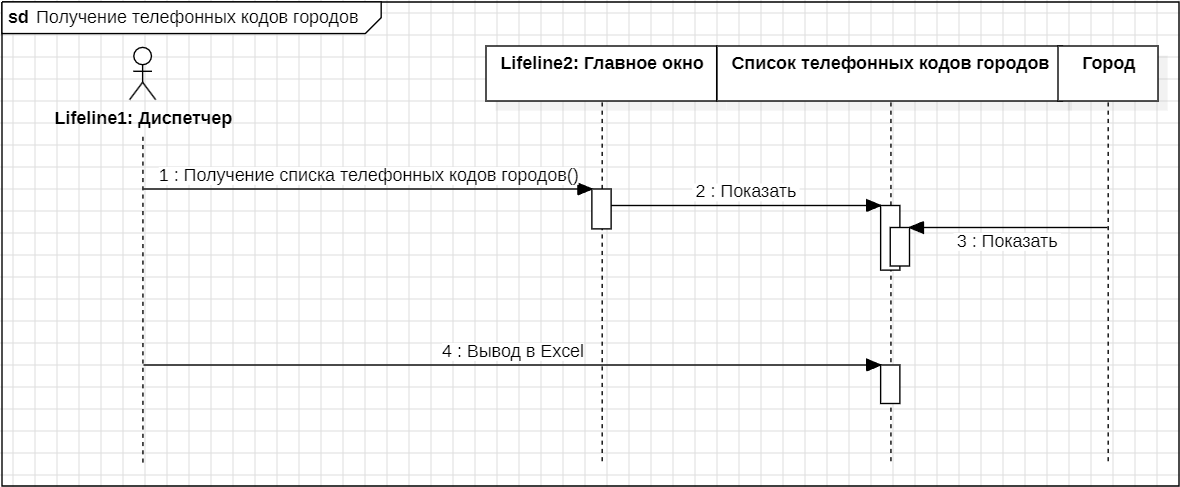


Рисунок 4.7 – Получение списка телефонных кодов городов

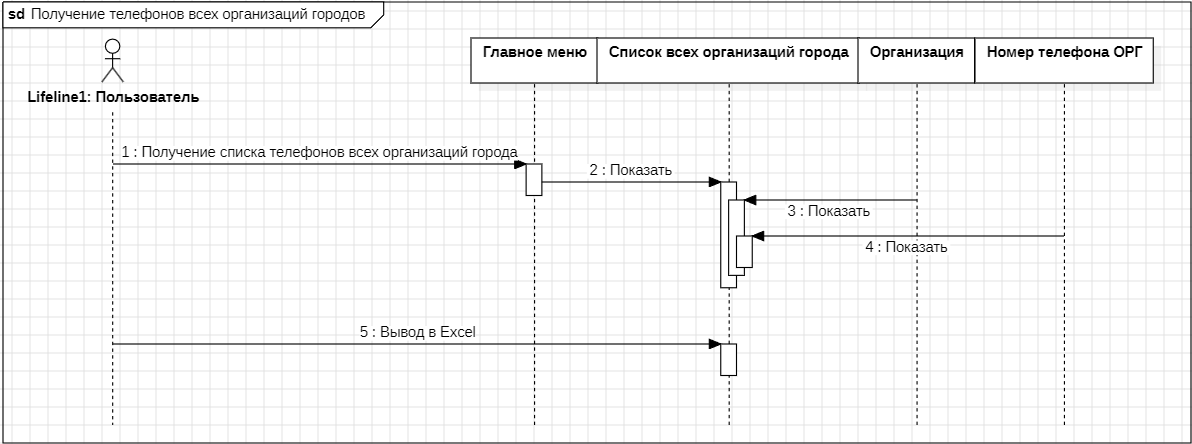


Рисунок 4.8 – Получение списка телефонов всех организаций городов

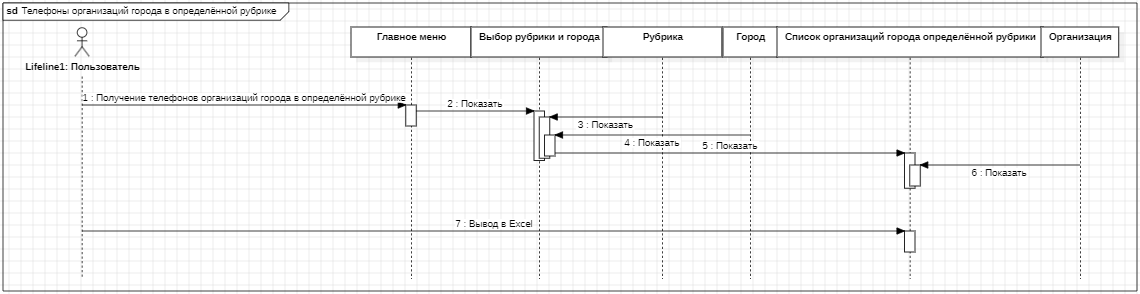


Рисунок 4.9 – Получения списка организаций города в определённой рубрике

# Концептуальная модель данных

Концептуальная модель методом «сущность-связь» (рис. 5).

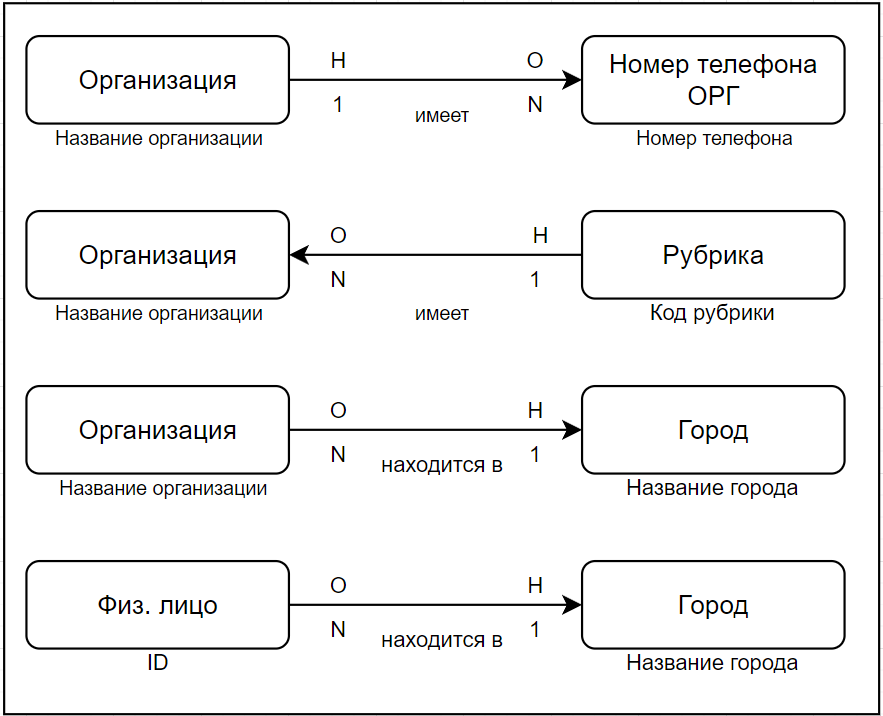


Рисунок 5 – Концептуальная модель данных

Мы получаем отношения по следующими правилами:

1. По 4-му правилу

Организация (Название организации)

Номер телефона ОРГ (Номер телефона ОРГ, Название организации)

1. По 4-му правилу

Организация (Название организации)

Рубрика (Код рубрики, Название организации)

1. По 4-му правилу

Организация (Название организации)

Город (Название города, Название организации)

1. По 4-му правилу

Физ. лицо (ID)

Город (Название города, ID)

Ключевые поля подчёркиваются.

Дублирующие отношения исключаются, оставляем те, в которых получилось больше атрибутов.

И добавив не ключевые атрибуты получим:

* Рубрика (Код рубрики, Название рубрики);
* Номер телефона ОРГ (Номер телефона ОРГ, Название организации);
* Организация (Название организации, Название города, Код рубрики, Адрес);
* Город (Название города, Телефонный код, Страна);
* Физическое лицо (ID, Название города, ФИО, Номер телефона ФЛ).

# Процесс нормализации

Далее произведём нормализацию данных (см. рис. 6.1 и 6.2).

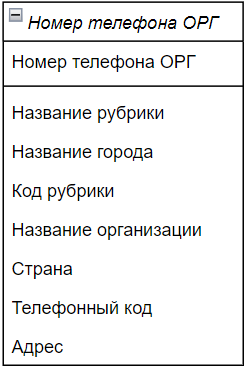


Рисунок 6.1 – Универсальное отношение для номеров телефона организаций

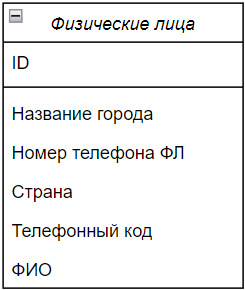


Рисунок 6.2 – Универсальное отношение для физических лиц

1NF требует атомарности атрибутов и отсутствия повторяющихся групп атрибутов.

Повторяющихся групп атрибутов нет.

Разделим неатомарные атрибуты (рис. 6.3 и 6.4).

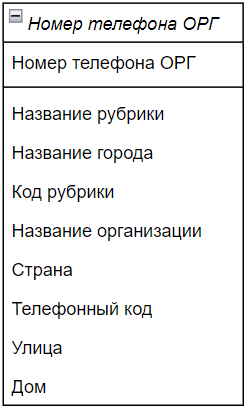


Рисунок 6.3 – Первая нормальная форма для номеров телефона организаций

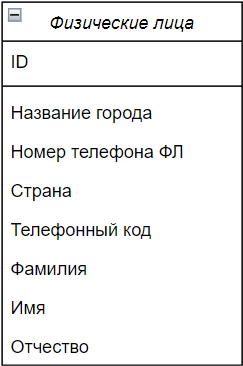


Рисунок 6.4 – Первая нормальная форма для физических лиц

2NF требует не избыточности первичного ключа и независимости не ключевых атрибутов от части составного первичного ключа. Т.к. в универсальном отношении по одному первичному ключу, то 2NF достигается автоматически.

3NF требует взаимной независимости не ключевых атрибутов (см. рис. 6.5 и 6.6).

Название организации → Название города

Название организации → Код рубрики

Название организации → Улица

Название организации → Дом

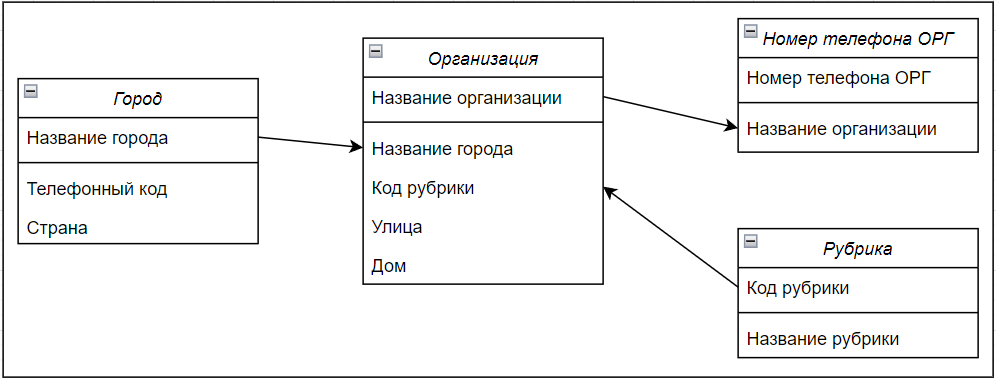


Рисунок 6.5 – Третья нормальная форма для номеров телефона организаций

ID → Название города

ID → Фамилия

ID → Имя

ID → Отчество

ID → Номер телефона ФЛ

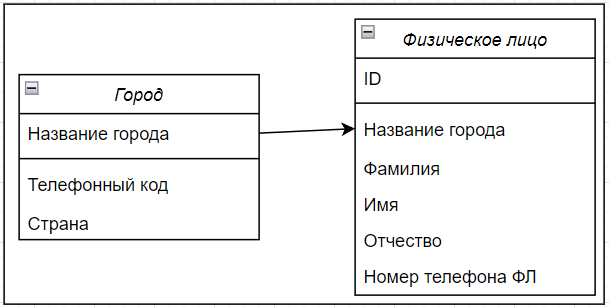


Рисунок 6.6 – Третья нормальная форма для физических лиц

# Логическая модель данных

Логическая модель предметной области иллюстрирует сущности, а также их взаимодействие между собой. Целью построения логической модели является получение графического представления логической структуры исследуемой предметной области. На рис. 7 представлена логическая модель базы данных.

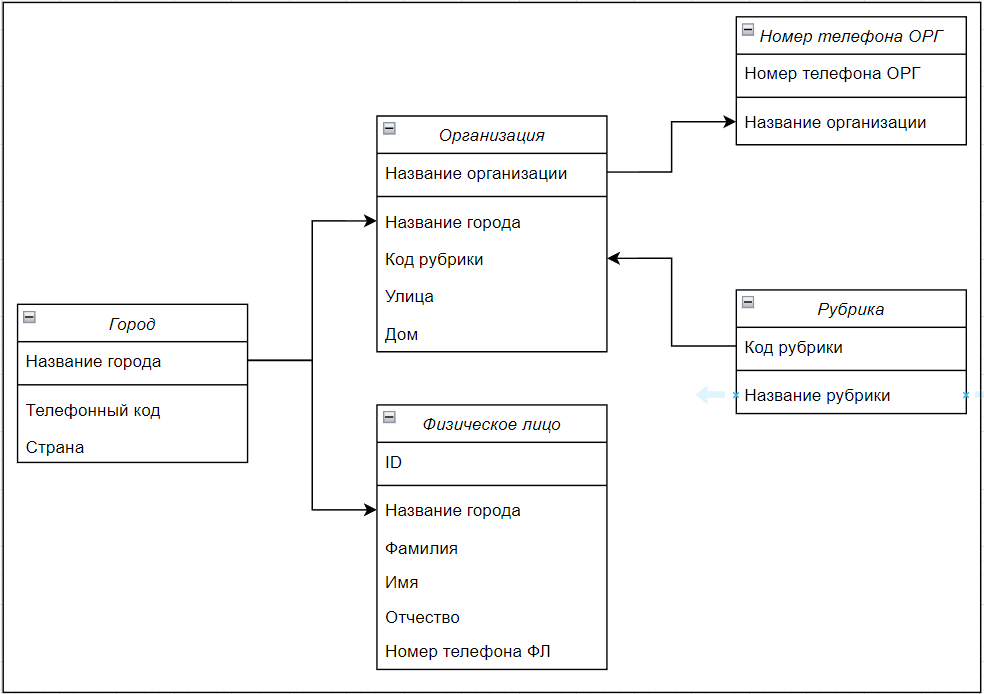


Рисунок 7 – Логическая модель данных

# Физическая модель данных

Физическая модель данных — это такая модель данных, которая определяет, каким образом представляются данные, и которая содержит все детали, необходимые СУБД для создания базы данных. На рис. 8 представлена физическая модель базы данных.

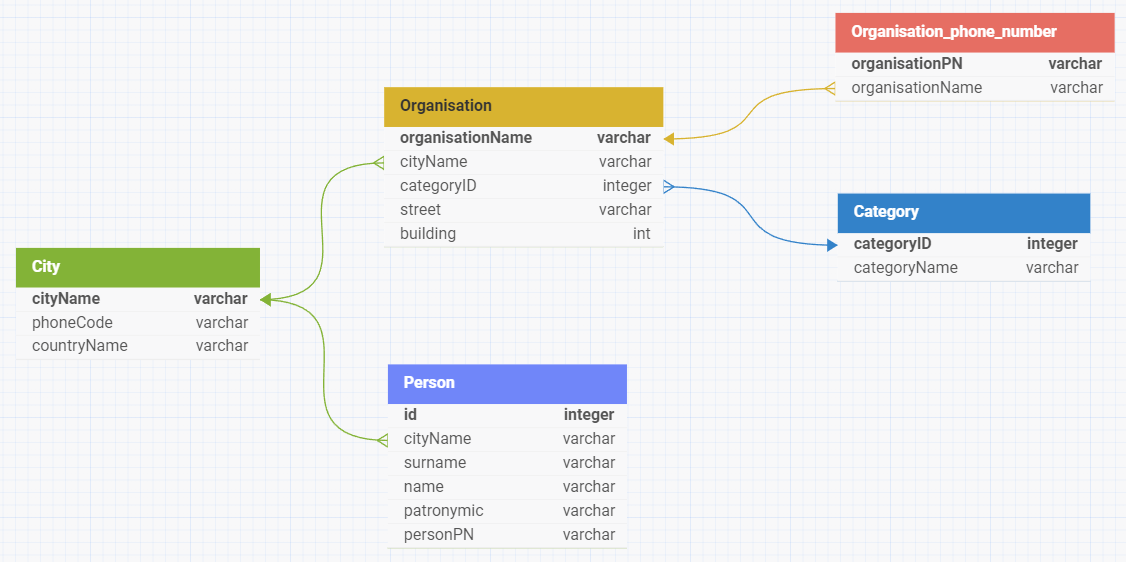


Рисунок 8 – Физическая модель данных

# Описание базы данных на сервере

Во время выполнения курсового проекта была использована XAMPP — кроссплатформенная сборка веб-сервера, содержащая Apache и MySQL.

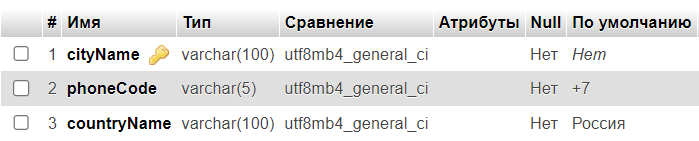


Рисунок 9.1 – Таблица city (Город)

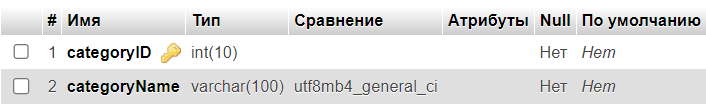


Рисунок 9.2 – Таблица category (Рубрика)

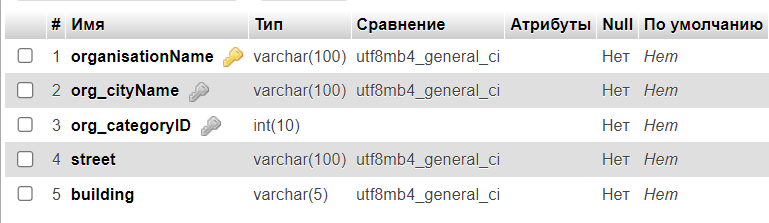


Рисунок 9.3 – Таблица organisation (Организация)

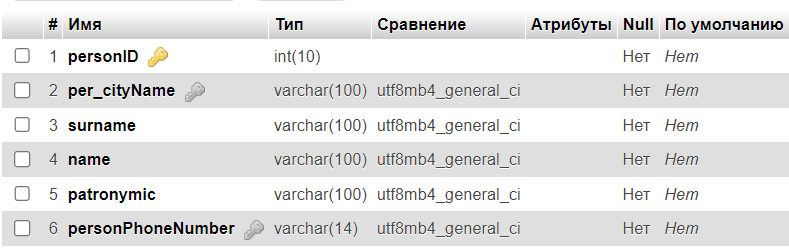


Рисунок 9.4 – Таблица person (Физическое лицо)

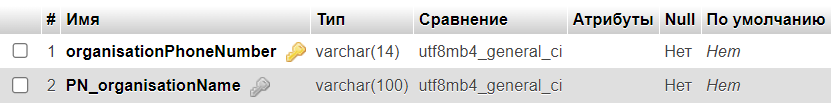


Рисунок 9.5 – Таблица organization\_phone\_number (Номера телефонов организации)

# Вывод данных на экран

Все таблицы оформлены в одном стиле. На каждой странице присутствуют все необходимые кнопки, подписанные русскими словами. Помимо функциональных кнопок, присутствуют и кнопки для маршрутизации по страницам.

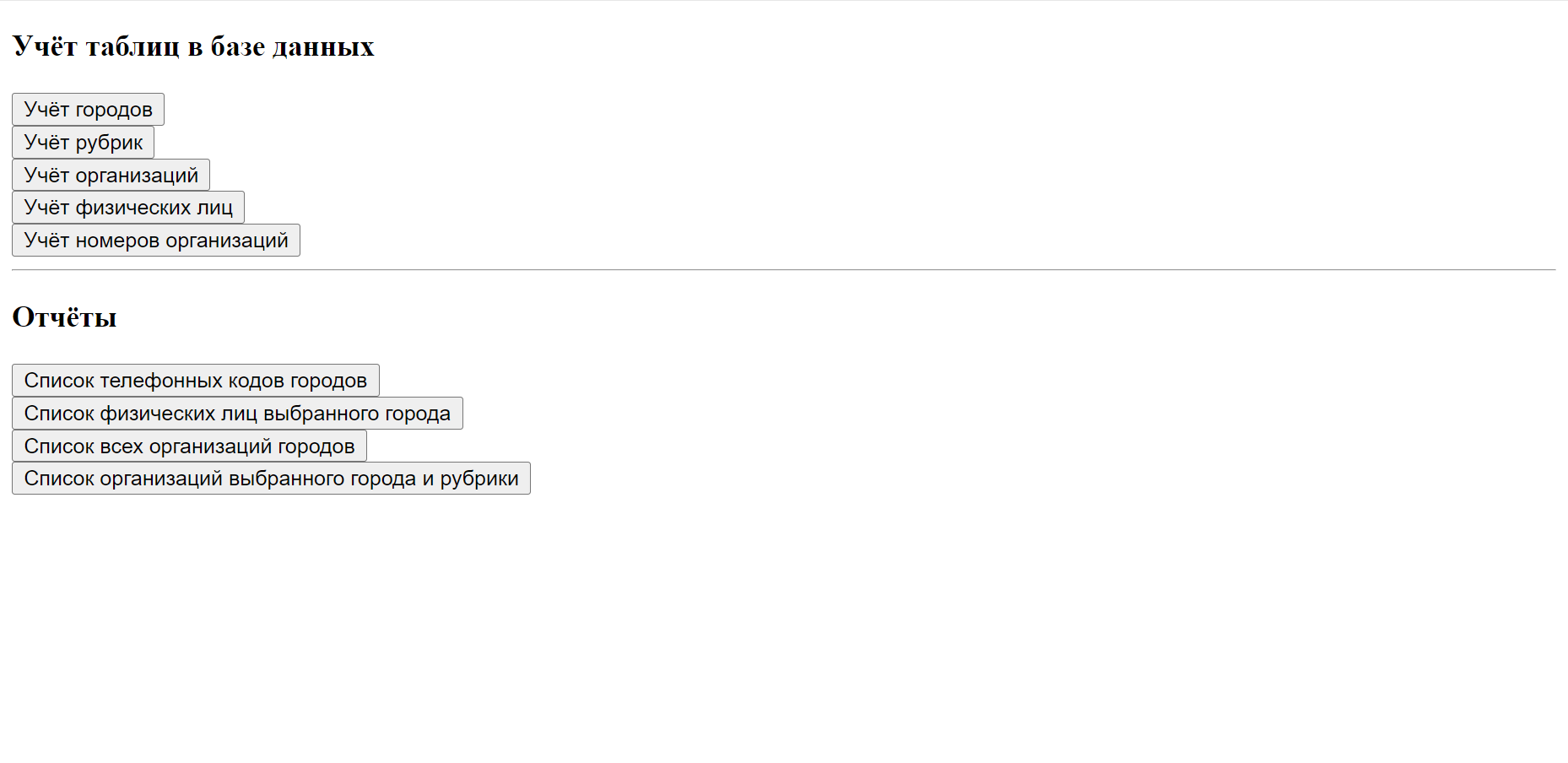


Рисунок 10.1 – Главная страница

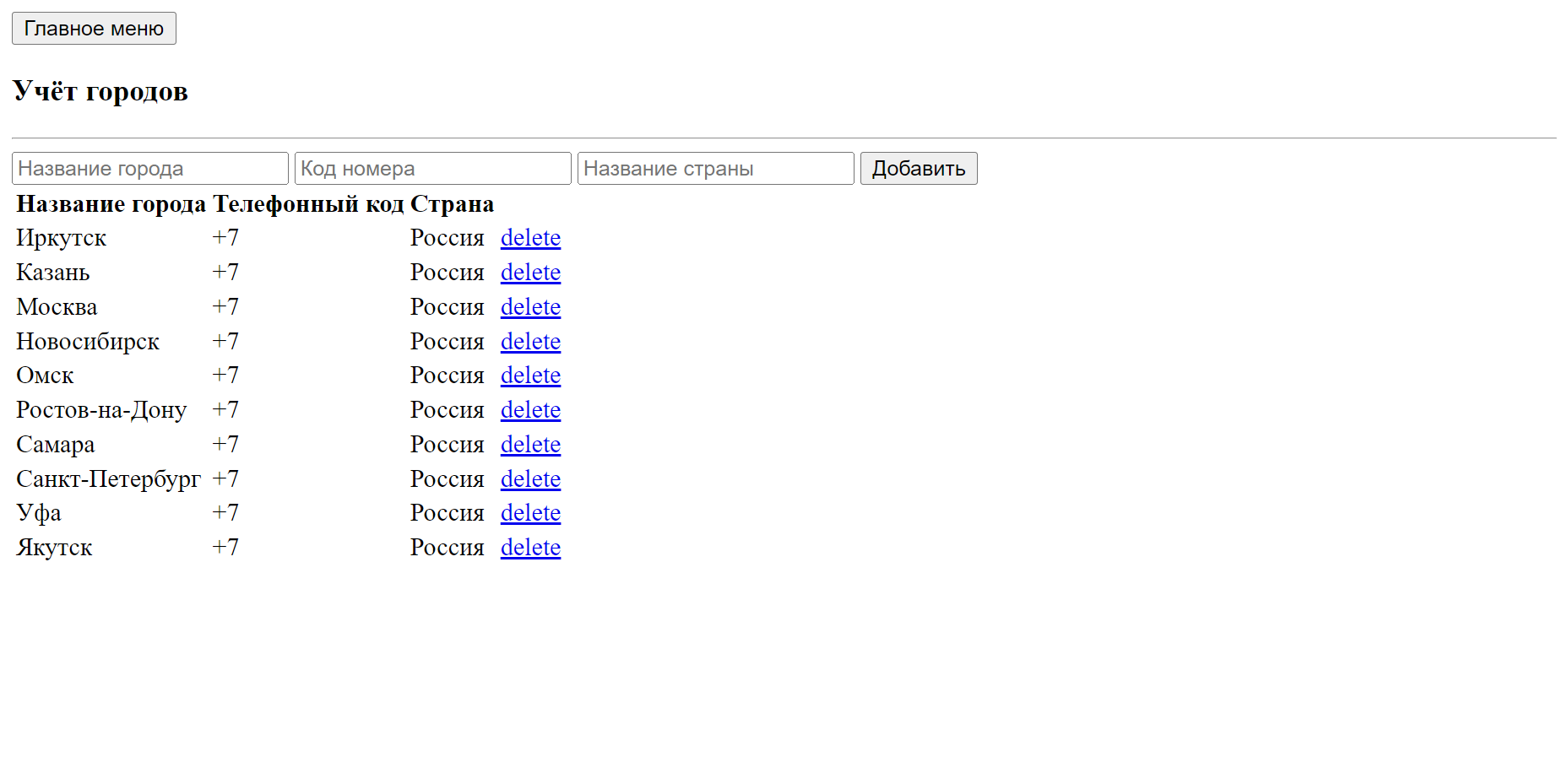


Рисунок 10.2 – Страница учёта городов

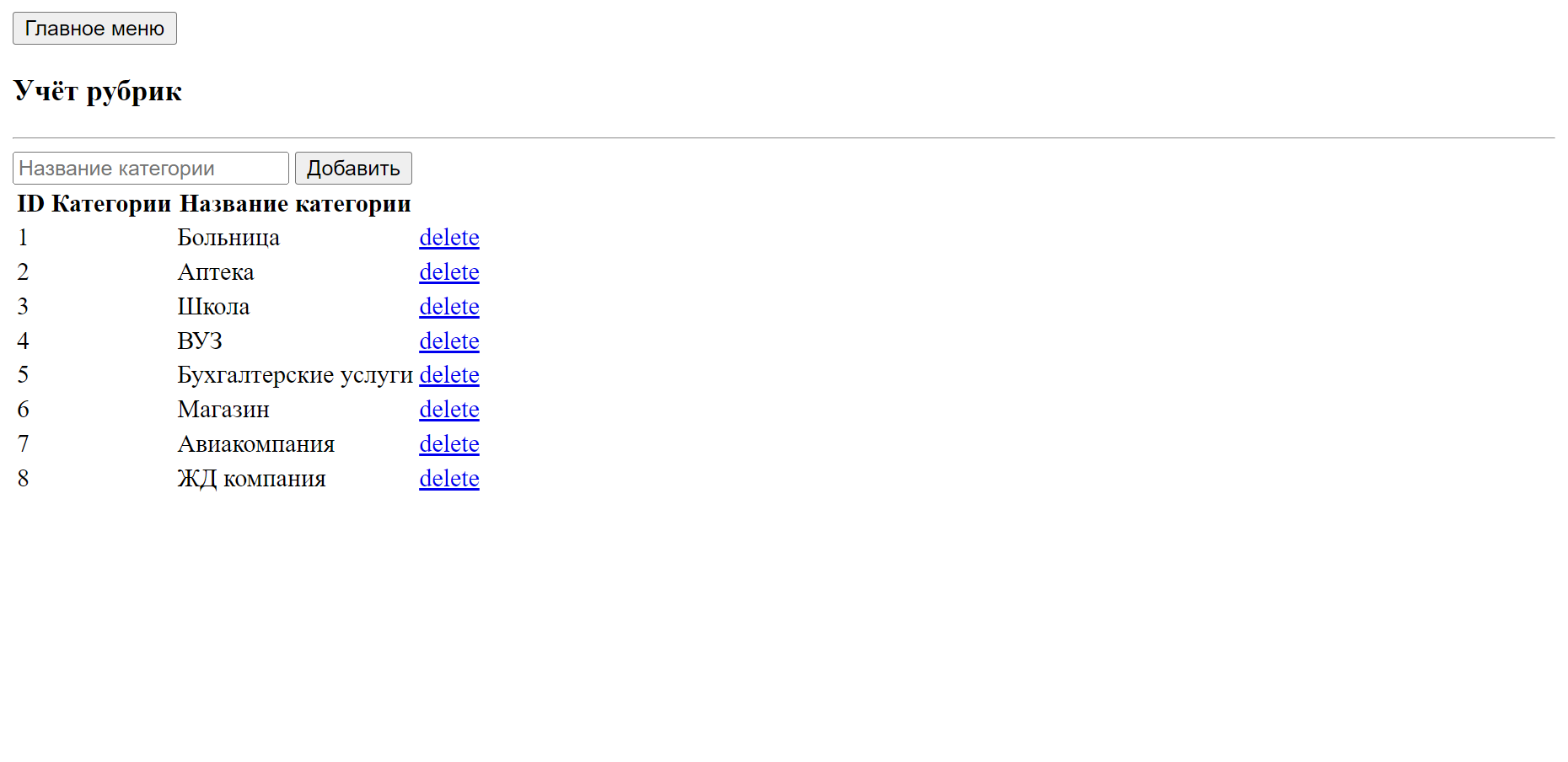


Рисунок 10.3 – Страница учёта рубрик

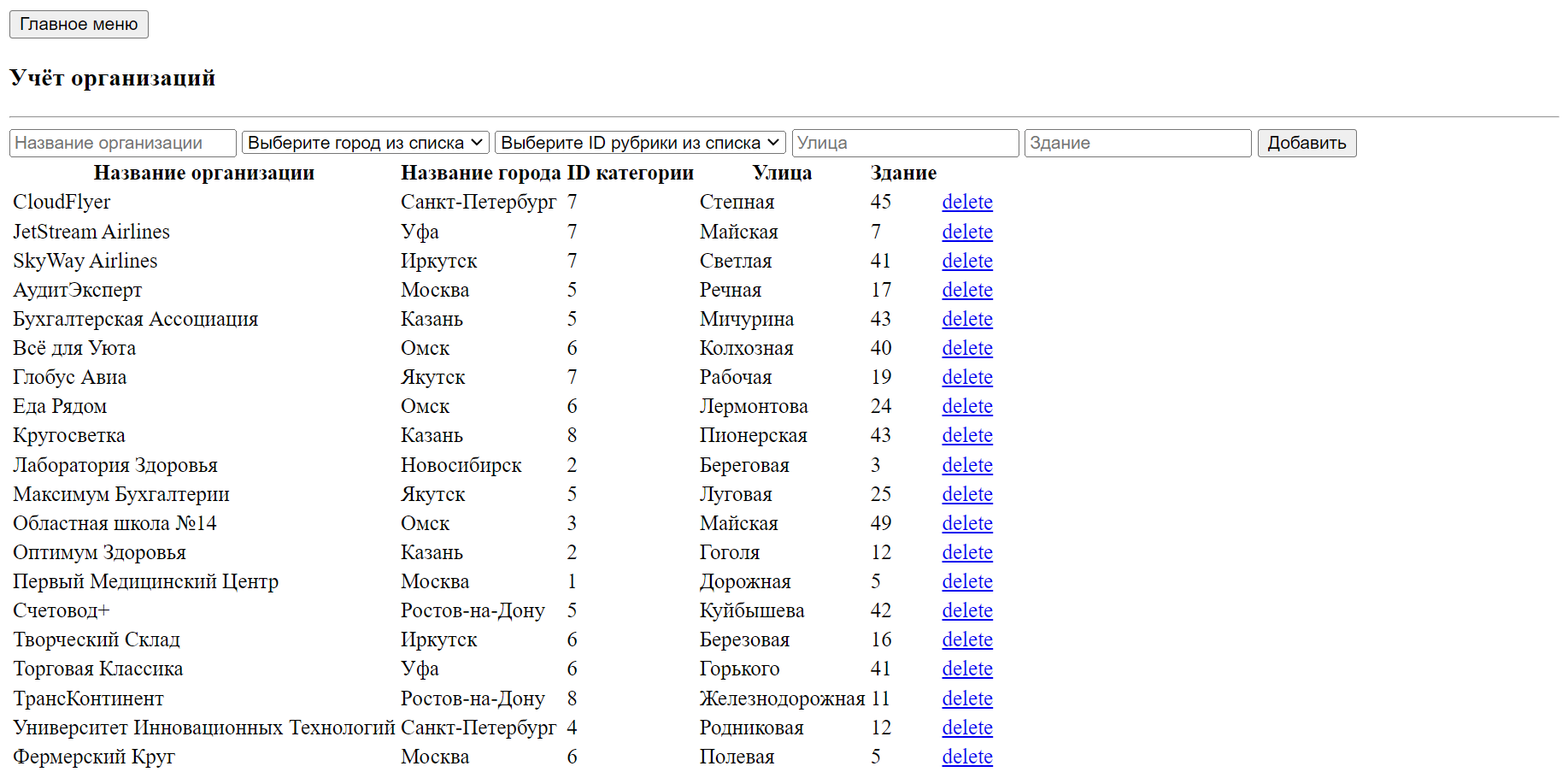


Рисунок 10.4 – Страница учёта организаций

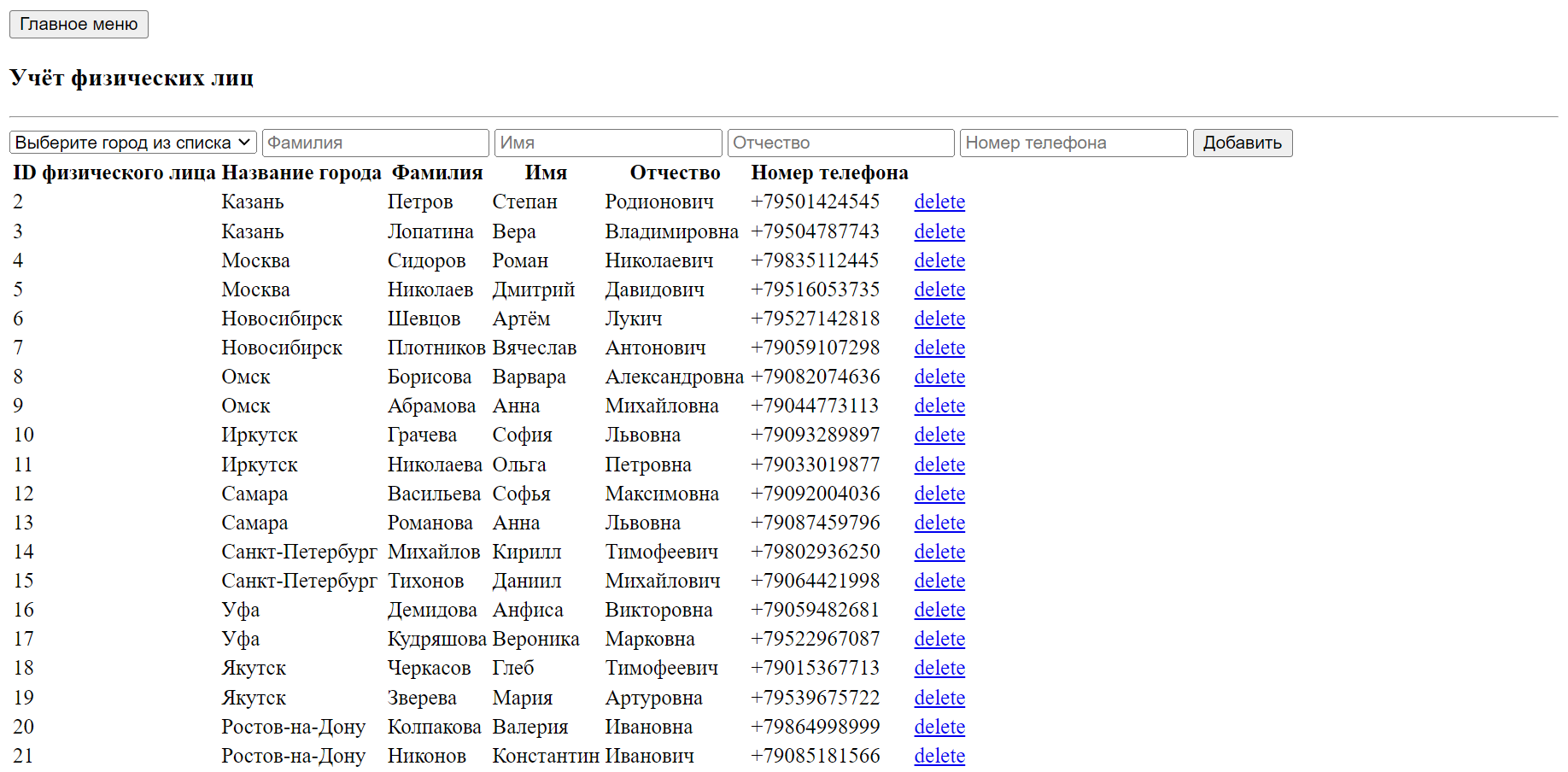


Рисунок 10.5 – Страница учёта физических лиц

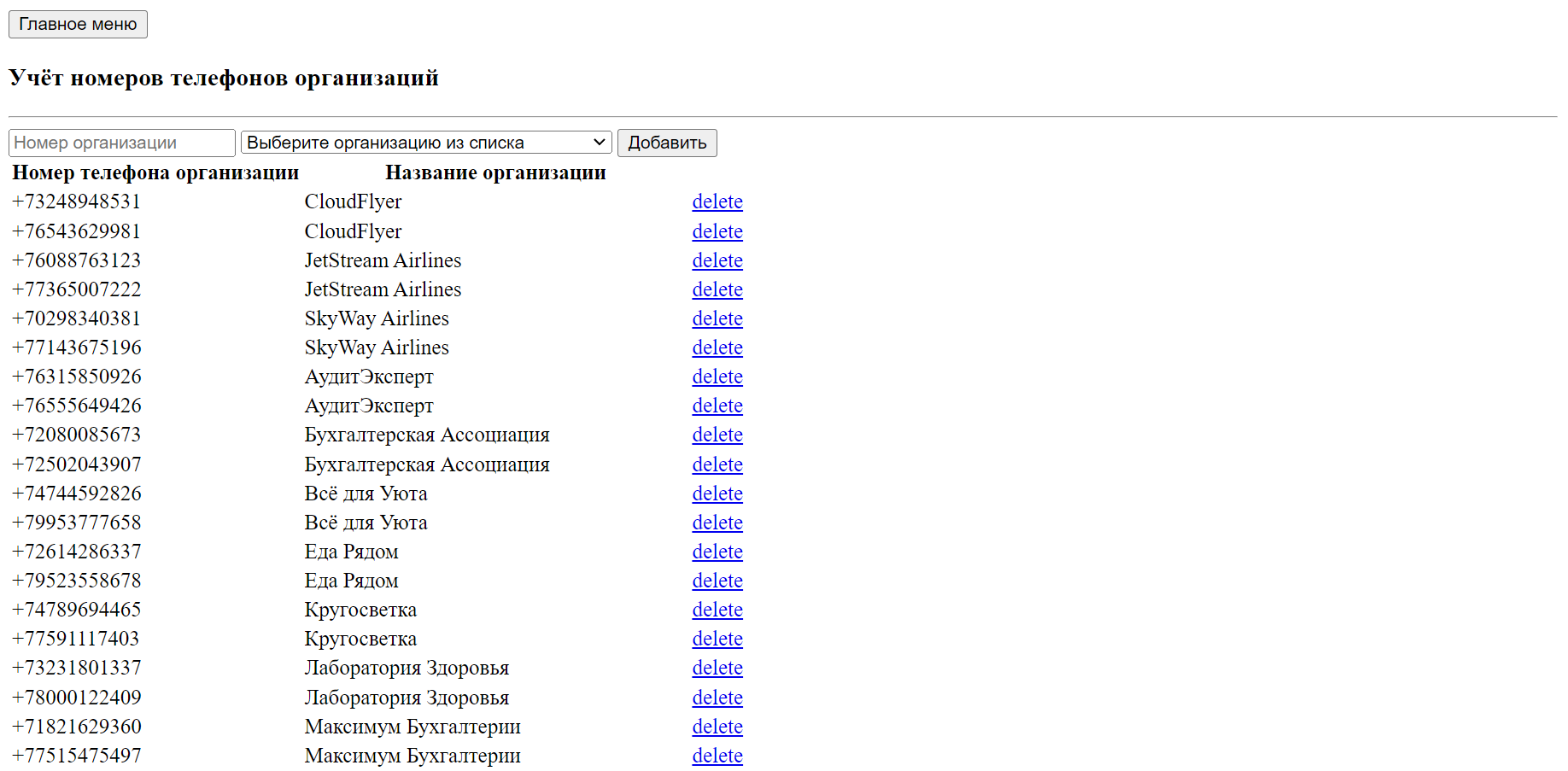


Рисунок 10.6 – Страница учёта номеров организаций

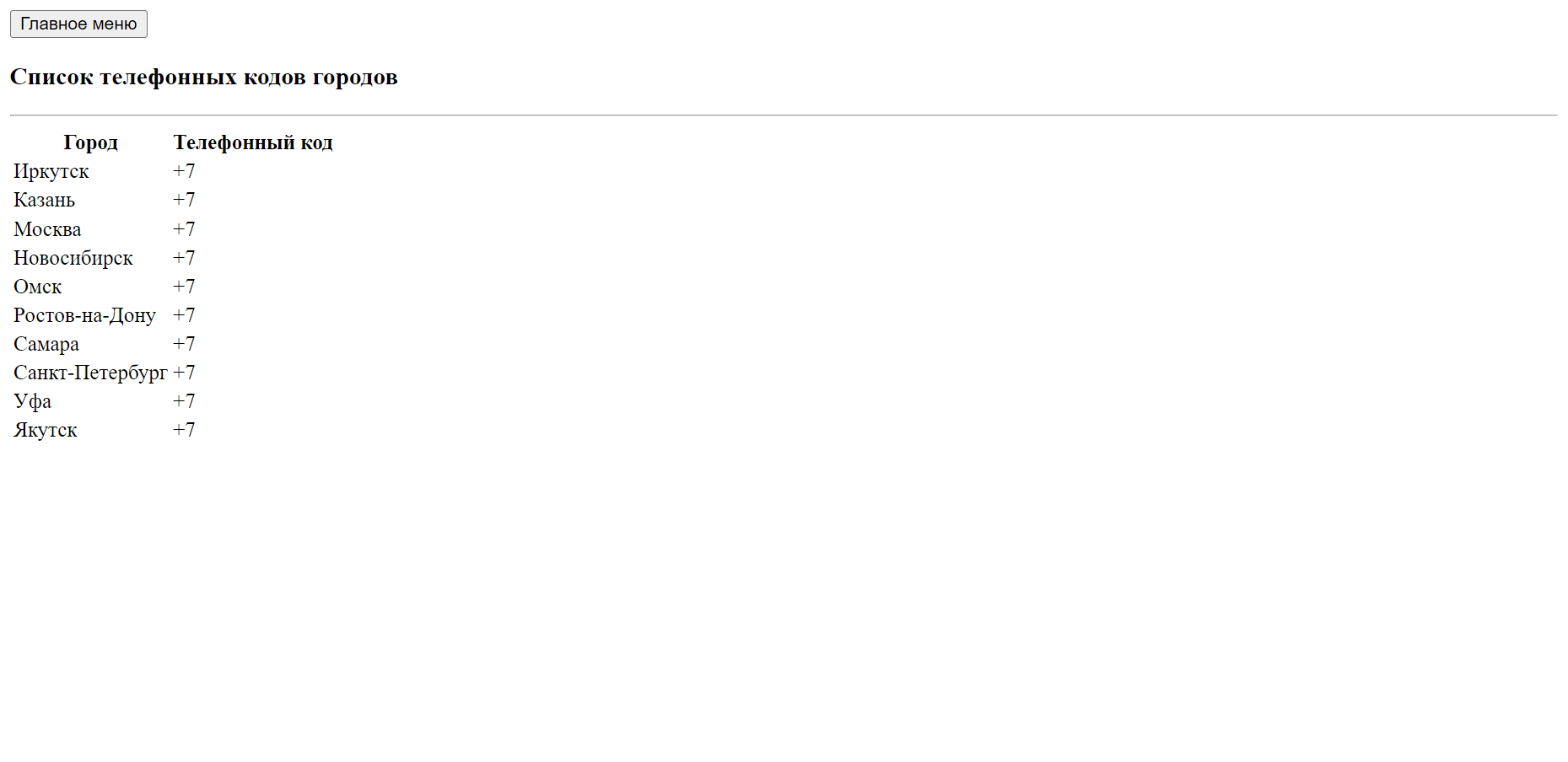


Рисунок 10.7 – Страница отчёта «Телефонные коды городов»

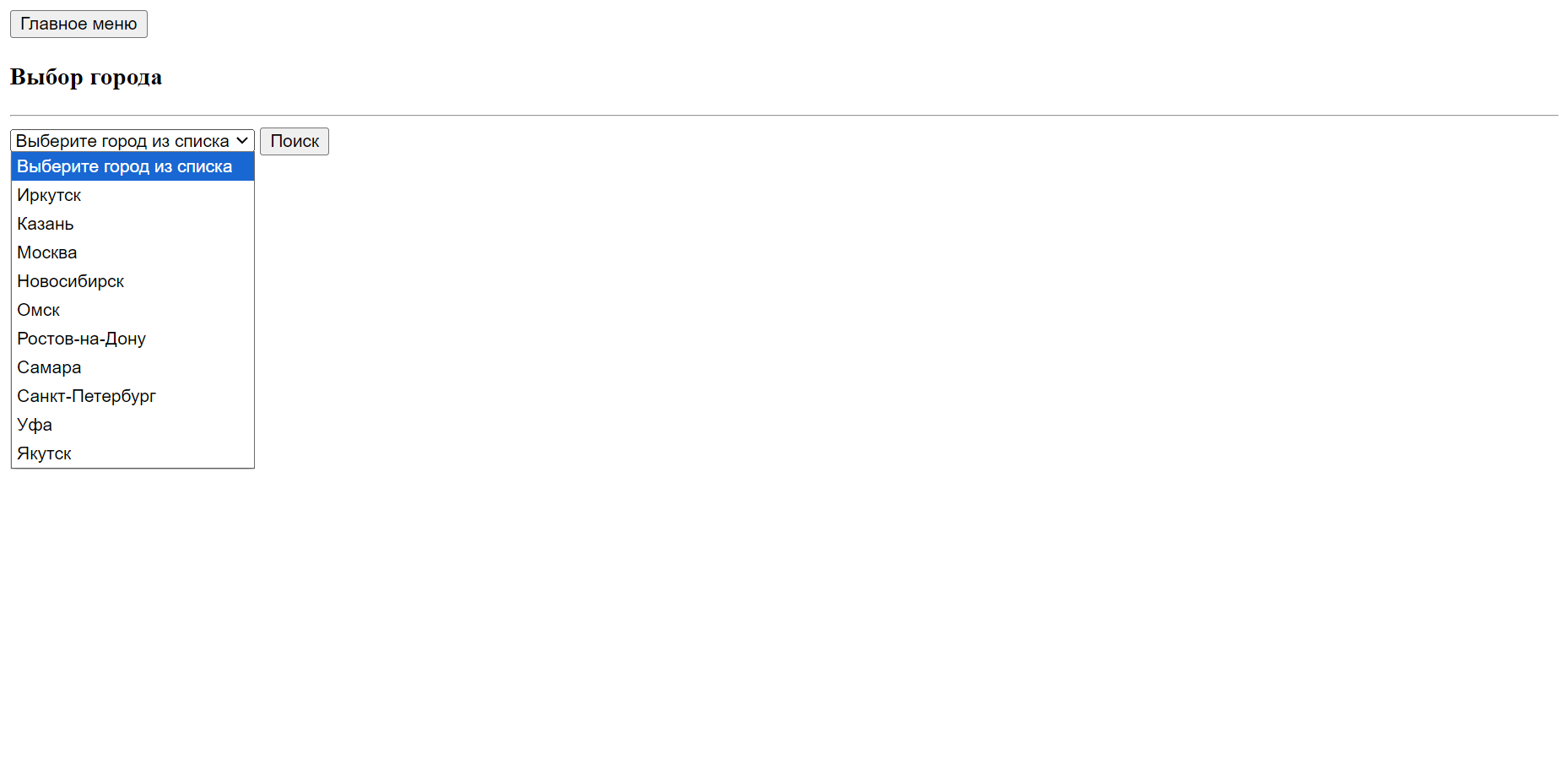


Рисунок 10.8 – Страница выбора города для отчёта

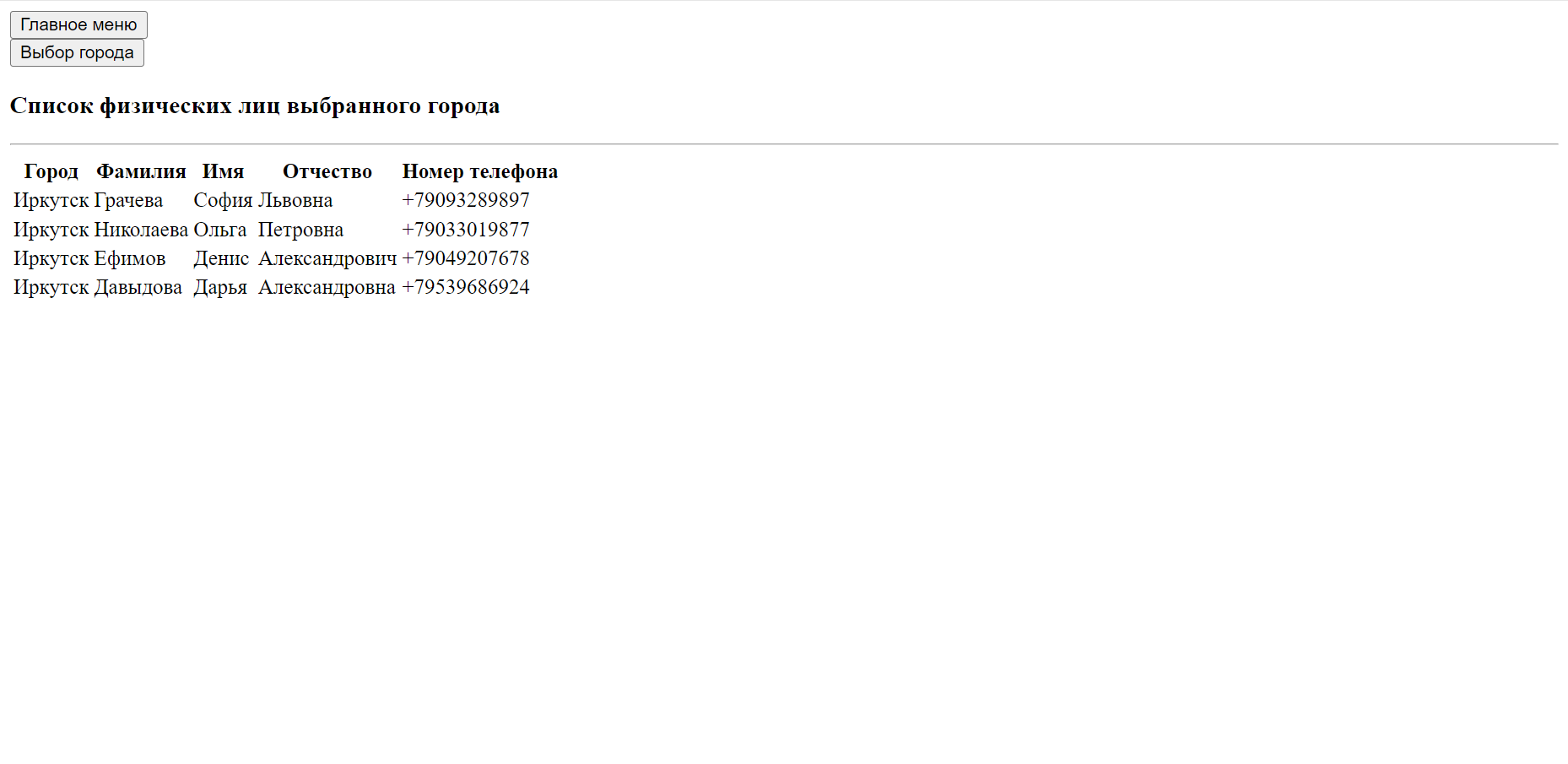


Рисунок 10.9 – Страница отчёта «Телефоны физических лиц города»

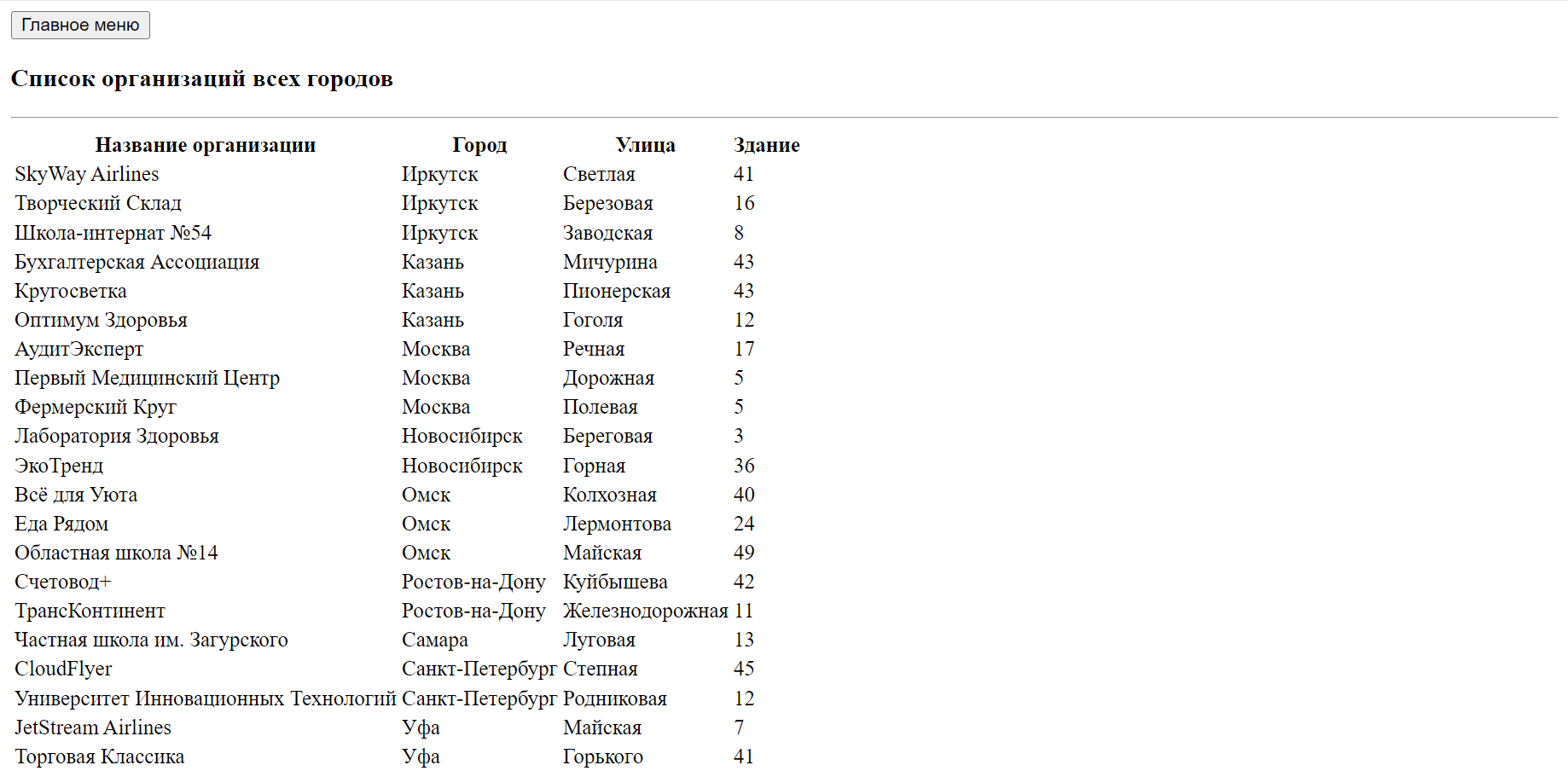


Рисунок 10.10 – Страница отчёта «Телефоны всех организаций городов»

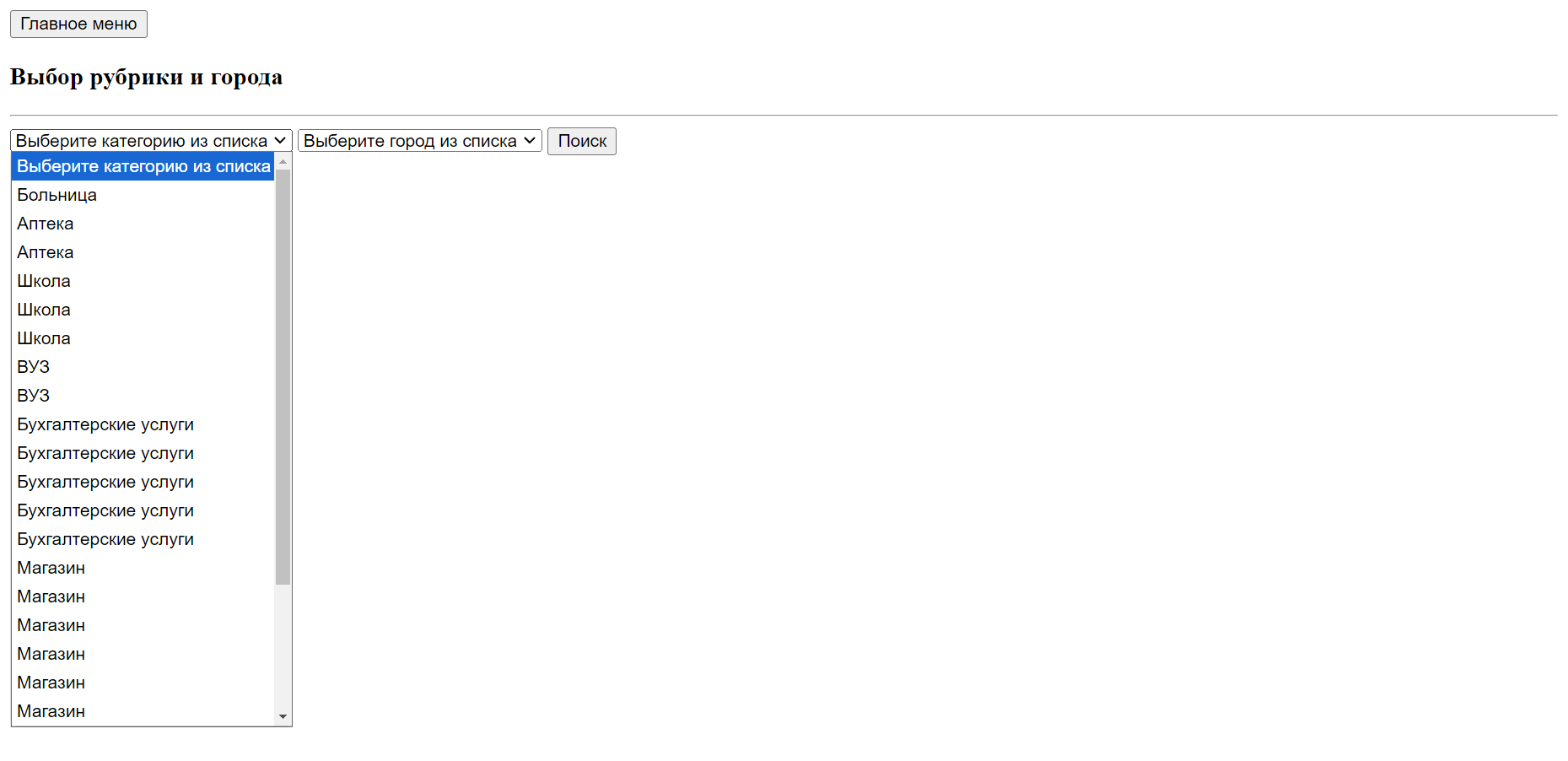


Рисунок 10.11 – Страница выбора рубрики и города для отчёта

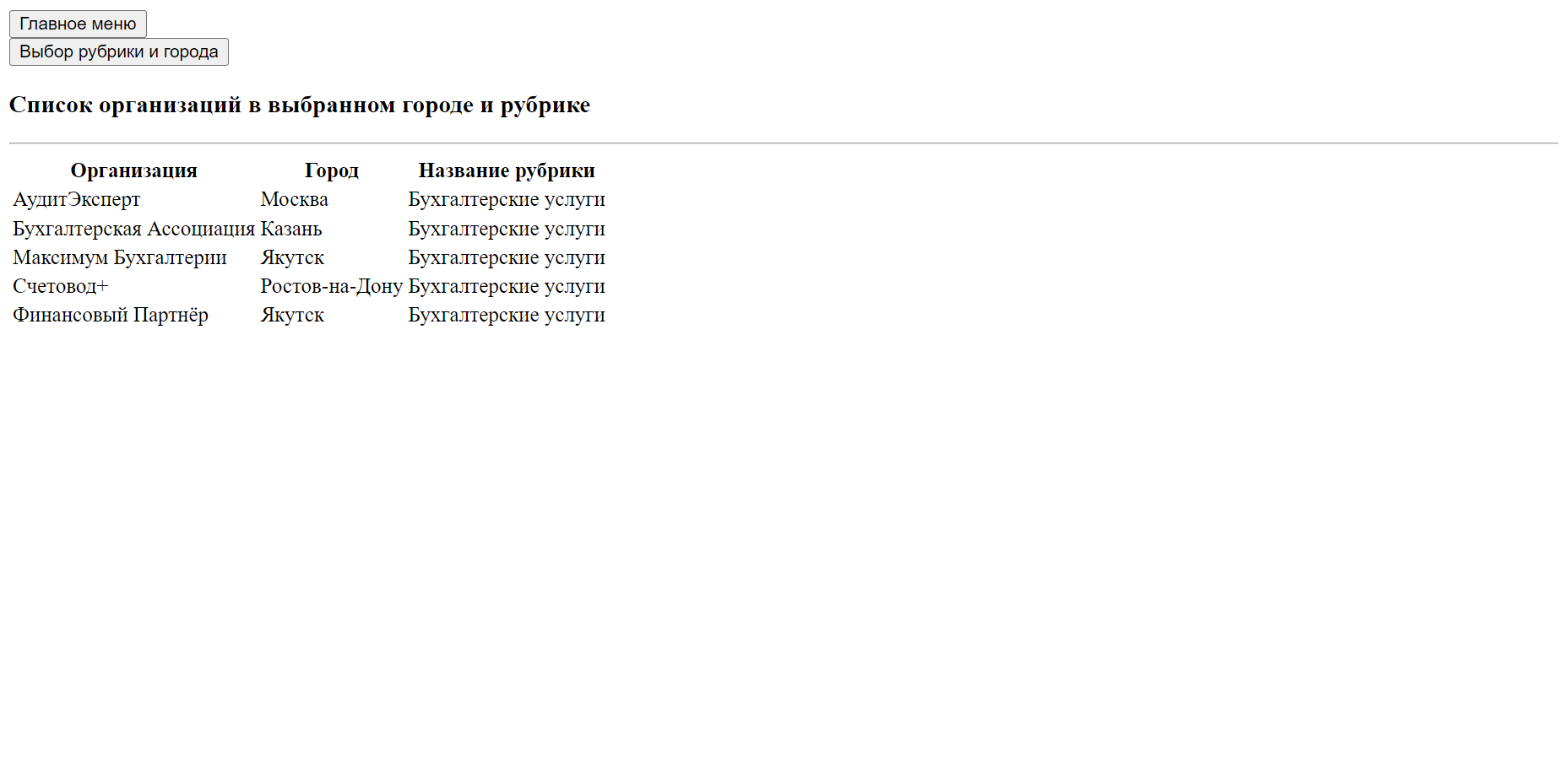


Рисунок 10.12 – Страница отчёта «Телефоны организаций города определенной рубрики»

# Таблица тестов

Проведём тестирование основных функций веб приложения для удостоверения в работоспособности.

Таблица 1 – Проверка работоспособности

|  |  |  |
| --- | --- | --- |
| № | Тест | Результат |
| 1 | Ввод данных в таблицу городов | Рисунок 11.1 |
| 2 | Удаление данных из таблицы городов | Рисунок 11.2 |
| 3 | Ввод данных в таблицу рубрик | Рисунок 11.3 |
| 4 | Удаление данных из таблицы рубрик | Рисунок 11.4 |
| 5 | Ввод данных в таблицу организаций | Рисунок 11.5 |
| 6 | Удаление данных из таблицы организаций | Рисунок 11.6 |
| 7 | Ввод данных в таблицу физических лиц | Рисунок 11.7 |
| 8 | Удаление данных из таблицы физических лиц | Рисунок 11.8 |
| 9 | Ввод данных в таблицу номеров организаций | Рисунок 11.9 |
| 10 | Удаление данных из таблицы номеров организаций | Рисунок 11.10 |



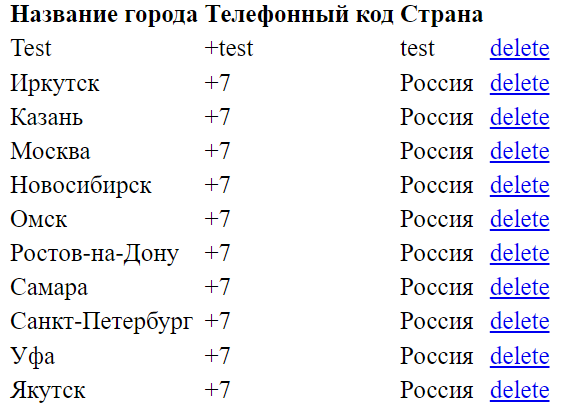


Рисунок 11.1 – Ввод данных в таблицу городов

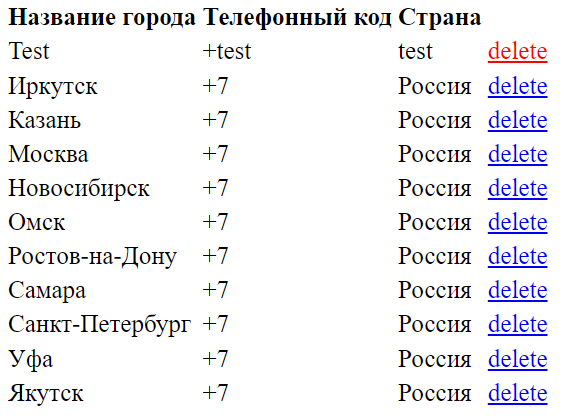
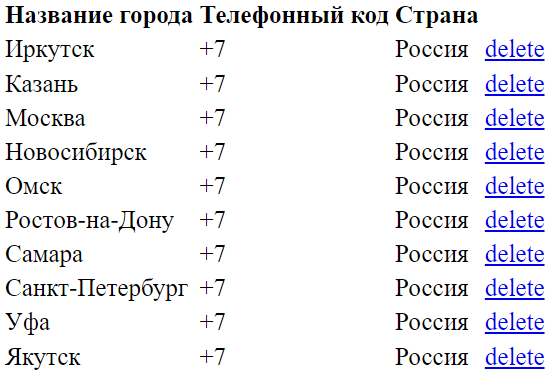
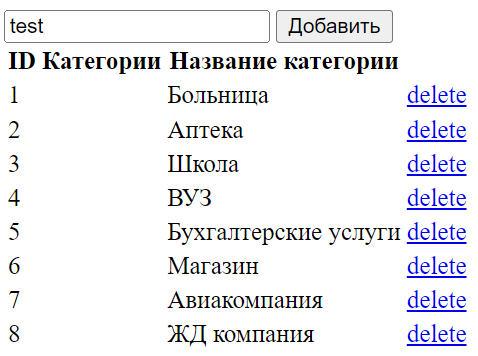
 

Рисунок 11.2 – Удаление данных из таблицы городов



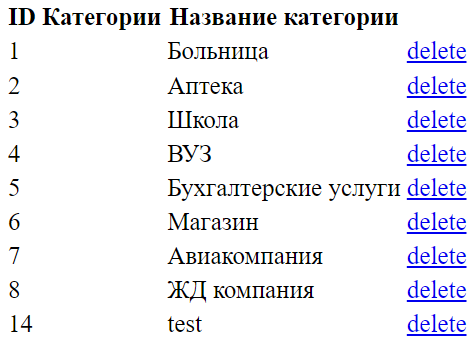
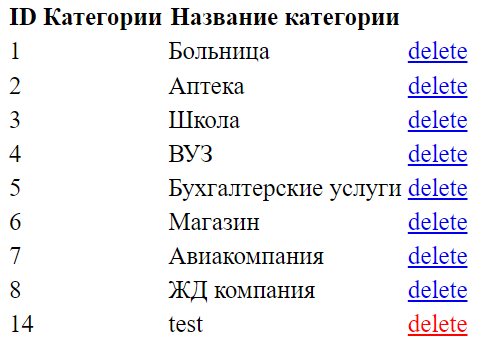


Рисунок 11.3 – Ввод данных в таблицу рубрик



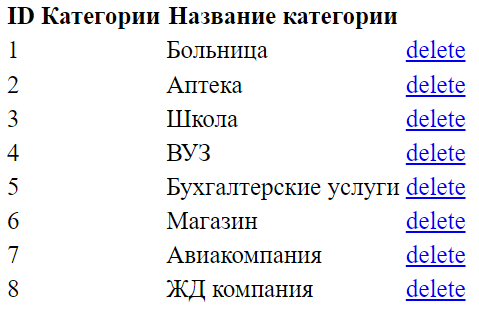
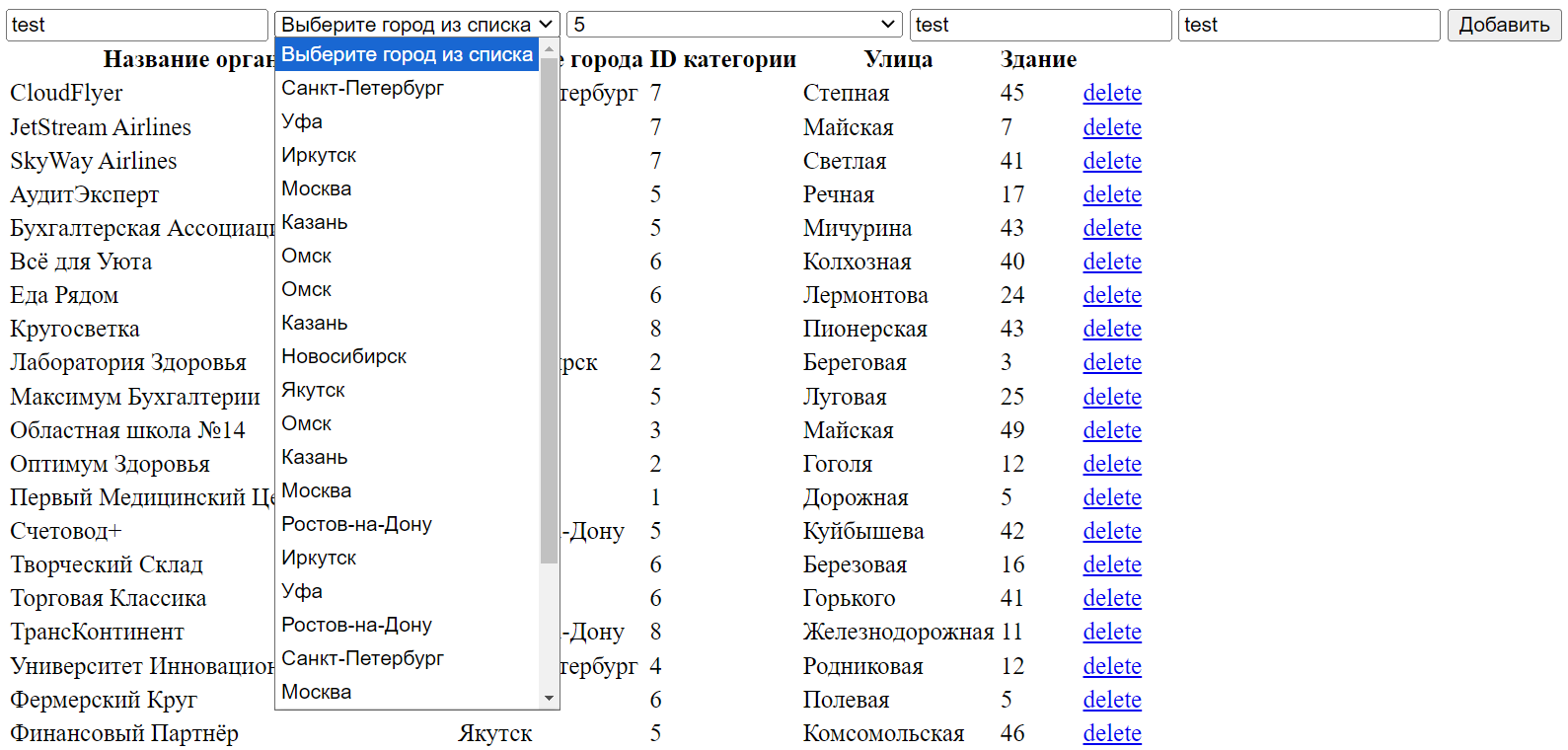


Рисунок 11.4 – Удаление данных из таблицы рубрик



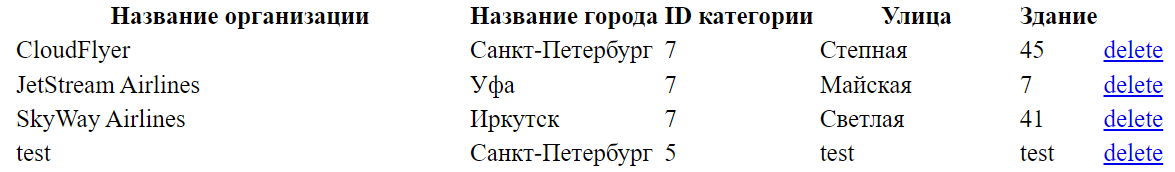


Рисунок 11.5 – Ввод данных в таблицу организаций

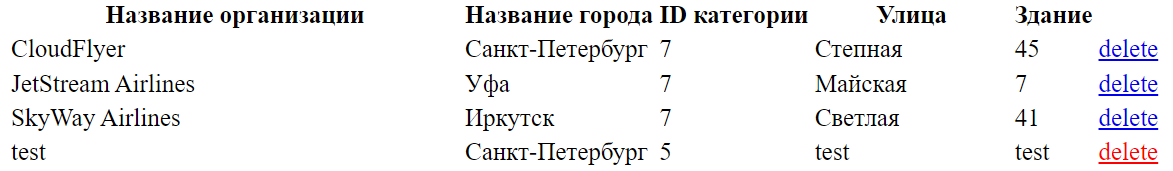




Рисунок 11.6 Удаление данных из таблицы организаций



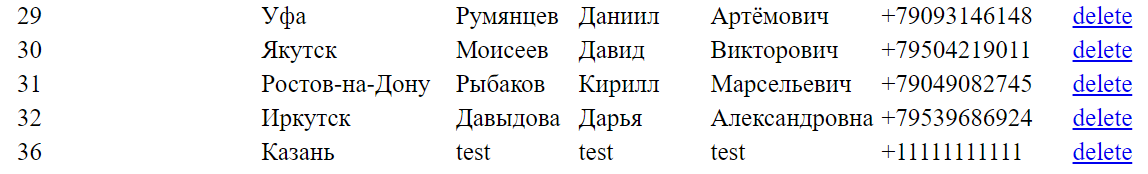
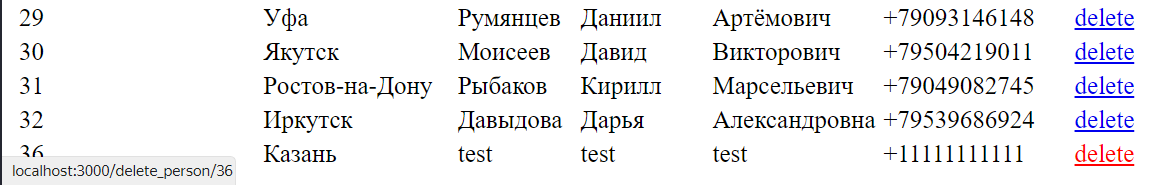


Рисунок 11.7 – Ввод данных в таблицу физических лиц



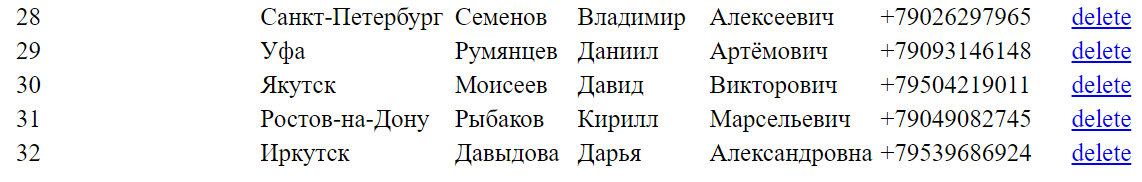
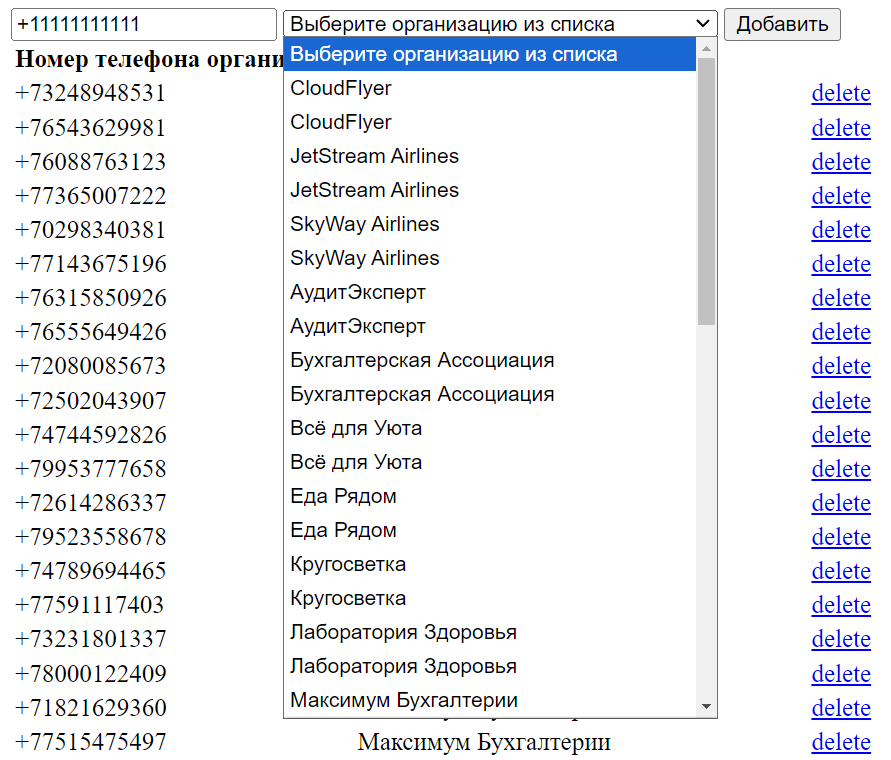


Рисунок 11.8 – Удаление данных из таблицы физических лиц



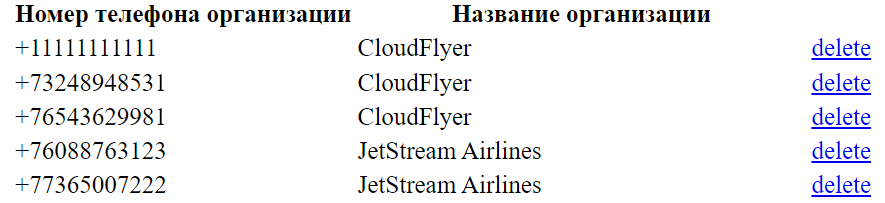
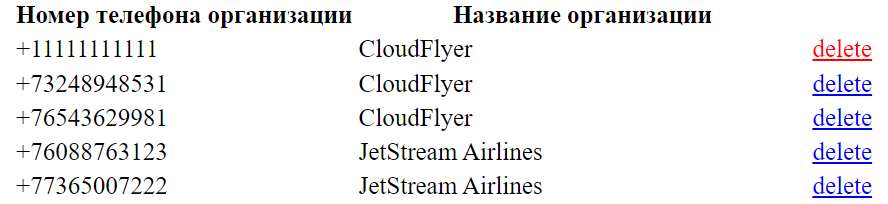


Рисунок 11.9 – Ввод данных в таблицу номеров организаций



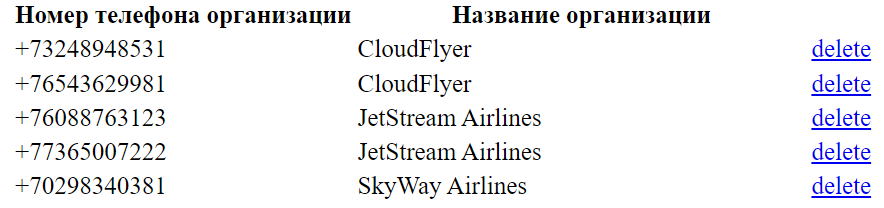


Рисунок 11.10 – Удаление данных из таблицы номеров организаций

Исходя из результатов тестирования можно сделать вывод, что все функции веб-приложения работают корректно.

# Заключение

В результате выполнения курсового проекта была спроектирована и реализована база данных «Телефонный справочник» в WEB-приложении. В базе данных представлены таблицы, связанные между собой внешними ключами.

Для каждой таблицы были реализованы функции добавления, просмотра и удаления. Была реализована функция экспорта выходного документа в MS Excel.

Была реализована функция авторизации, которая позволяет разграничивать права доступа для разных ролей.

В результате выполнения курсового проекта были выполнены такие задачи, как:

* составлена объектная модель задачи;
* составлена логическая и физическая модель данных;
* созданы страницы с отображением данных из базы данных;
* проведено тестирование WEB-приложения;
* составлен отчёт по выполненному курсовому проекту.