

**САНКТ-ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ  
БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ  
«КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ»**

**ОТЧЕТ ПО УЧЕБНОЙ ПРАКТИКЕ**

МДК 01.01. Разработка программных модулей

Специальность 09.02.07

Информационные системы и программирование

Квалификация «Программист»

Выполнил  
студент гр. 493  
Руководитель учебной практики

\_\_\_\_\_ И.О. Омелько  
\_\_\_\_\_ И.А. Матысик

Санкт-Петербург, 2023

## СОДЕРЖАНИЕ

Техническое задание на учебную практику .....	3
Введение.....	5
1. Общая часть.....	7
1.1. Цель разработки.....	7
1.2. Описание предметной области.....	7
1.3. Анализ средств и технологий.....	9
2. Проектная часть .....	12
2.1. Постановка задачи .....	12
2.2. Функционально-логическая структура проекта.....	14
2.2.1. Диаграмма вариантов использования .....	14
2.2.2. Диаграмма деятельности .....	15
2.3. Логическая модель данных.....	16
2.4. Разработка программных модулей .....	21
2.4.1. Пользовательский интерфейс.....	22
2.5. Руководство пользователя .....	22
2.5.1. Форма авторизации .....	22
2.5.2. Форма меню .....	23
2.5.3. Форма просмотра акции .....	25
2.5.4. Форма редактирования акции .....	26
2.5.5. Форма редактирования волонтера .....	32
2.5.6. Форма Просмотра акции для волонтеров .....	37
2.5.7. Выводы по учебной практике .....	39
Список используемых источников.....	41

## ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Требуется разработать информационную систему (ИС) «Дари добро» – систему управления волонтерским движением. Элементом в данной системе является волонтер. Для этого требуется разработать основные программные модули ИС для следующих пользователей:

- волонтер может просматривать акции помощи и записываться на акцию или выйти из состава акции, а также формировать отчет о завершенных акциях;

- организатор может просматривать список волонтеров, просматривать акций помощи, регистрировать новую волонтерскую акция, редактировать имеющуюся акцию и удалять акцию;

- менеджер волонтеров может просматривать список волонтеров, добавлять, удалять и редактировать волонтеров, а также формировать отчет о завершенных акциях;

- пользователь со статусом волонтер может быть зарегистрирован только менеджером волонтеров.

Выполнить анализ и проектирование бизнес-процессов разрабатываемой информационной системы путем составления диаграммы прецедентов (Use-Case) для пользователей системы.

Выполнить проектирование реляционного хранилища данных путем составления ER-диаграммы разрабатываемой информационной системы и ведение словаря данных.

Выполнить разработку базы данных информационной системы «Дари добро» на сервере баз данных, создать таблицы основных сущностей, атрибуты, отношения и необходимые ограничения. Выполнить импорт подготовленных данных в разработанную базу данных информационной системы.

Выполнить разработку desktop-приложения для автоматизации процесса организации волонтерских акций. Система должна функционировать в

многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему, т.е. приложение должно обеспечивать процесс авторизации пользователя в системе.

Организатор может создавать, редактировать и удалять волонтерские акции. Просматривать волонтерские акции. Организатор имеет возможность просматривать список волонтеров.

Менеджер волонтеров администрирует волонтеров. Менеджер волонтеров может создавать волонтеров, редактировать волонтеров и удалять данные о волонтерах. Менеджер волонтеров может просматривать список волонтеров.

Волонтер может просматривать списки волонтерских акций. Волонтер может просматривать данные о выбранной акции. Волонтер может присоединиться к составу акции или покинуть состав волонтерской акции.

В целом разрабатываемая система направлена на автоматизацию процессов организации и управления мероприятиями, организованными различными организациями.

## **ВВЕДЕНИЕ**

Учебная практика по МДК 01.01 «Разработка программных модулей» является обязательным учебным элементом для формирования общих и профессиональных компетенций по модулю ПМ 01 «Разработка модулей программного обеспечения для компьютерных систем».

Практика проходит в колледже информационных технологий, в компьютерной лаборатории №213 «Прикладное программирование». Период прохождения практики с 23.02.2023 по 15.03.2023. Объем учебной практики 30 часов. Тема учебной практики: «Разработка информационной системы магазина по продаже компьютерных комплектующих».

В отчете к учебной практике представлена краткая необходимая теоретическая справка, представлена четкая постановка задачи, представлены диаграммы, описывающие функциональную структуру программы и логическую модель данных, подробная инструкция пользователя и выводы по проделанной работе. Представлен программный код в приложении. В результате прохождения учебной практики по МДК 01.01 были подтверждены общие и профессиональные компетенции.

Перечень общих компетенций:

В результате прохождения учебной практики по МДК 01.01 «Разработка программных модулей», был получен практический опыт в следующих пунктах:

- в разработке кода программного продукта на основе готовой спецификации на уровне модуля;
- в использовании инструментальных средств на этапе отладки программного продукта.

В результате прохождения учебной практики по модулю, были получены знания по следующим понятиям:

- основные этапы разработки программного обеспечения;

- основные принципы технологии структурного и объектно-ориентированного программирования;
- основные возможности современных технологий.

# **1. ОБЩАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлена краткая теория по теме учебной практике, а также представлены результаты исследования проведенного анализа программных средств и технологий для разработки проекта учебной практики.

## **1.1. Цель разработки**

Целью проекта является проектирование и разработка информационной системы для организаторов волонтерских акций и волонтеров, которая поможет автоматизировать процесс организации и выполнения волонтерских акций, что поможет сократить затраты времени на организацию состава участников акции. В итоге разрабатываемая система должна привлечь больше потенциальных пользователей, обеспечив удобство участия в волонтерском движении, для целей которого разрабатывается данная информационная система.

## **1.2. Описание предметной области**

Информационная система "Дари добро" – прикладное программное обеспечение, использующееся для предоставления пользователям возможности мониторинга волонтерских акций и управления ими, а также ряда следующих функциональных особенностей:

- добавление новой волонтерской акции в базу данных и предоставление сведений о ней волонтерам;
- редактирование или удаление уже существующей волонтерской акции;
- добавление нового волонтера в базу данных и предоставление сведений о нем организаторам;
- редактирование или удаление уже существующего волонтера;

- добавление волонтера в состав акции и удаление волонтера из состава акции.

Основными целями внедрения системы являются:

- упрощение предоставления информации волонтерских акциях организаторов;
- автоматизация процесса создания волонтерской акции;
- минимизация человеческого фактора при ведении документов;
- снижение рутинной работы при работе с составом участников акций.

Рассмотрим определение прецедентов (вариантов использования).

Система требуется, прежде всего, следующим заинтересованным лицам:

- организатор;
- менеджер волонтеров;
- волонтер.

Функционал каждой роли:

Организатор может просматривать список волонтеров, просматривать акции помощи, регистрировать новую волонтерскую акция, редактировать имеющуюся акцию и удалять акцию.

Менеджер волонтеров может просматривать список волонтеров, добавлять, удалять и редактировать волонтеров.

Волонтер может просматривать список акций помощи и записываться на акцию или выйти из состава акции.

Система должна функционировать в многопользовательском режиме, поэтому каждый пользователь должен иметь свой пароль доступа в систему.

Проект предназначен для некоммерческих организаций, занимающихся волонтерской деятельностью, проводящих волонтерский акции в сотрудничестве с различными юридическими и физическими лицами.

Внедрение данного проекта поможет автоматизировать процесс добавления новых волонтеров, способствуя снижению рутинной работы с ведением документооборота и минимизации случаев человеческого фактора.



### 1.3. Анализ средств и технологий

В данном разделе описываются возможные пути и программные средства, с помощью которых можно реализовать приложение.

В данном подразделе пояснительной записки представлен материал проведенного анализа программных средств, которыми можно разработать приложение. В анализе присутствуют сравнительные характеристики программных технологий и методов, языков программирования и интегрированных сред.

Инструментальные средства:

- операционная система – Windows 11;
- язык программирования – "C#";
- среда программирования – "Microsoft Visual Studio 2019";
- СУБД – "SQL Server Management Studio 2019".

Для создания приложения будет использована интегрированная среда программирования "Microsoft Visual Studio 2019", язык программирования высокого уровня C#, систему построения клиентских приложений "Windows Presentation Foundation" (WPF), а также "SQL Server Management Studio 2019" для создания базы данных.

В качестве среды разработки была выбрана инструментальная среда разработки "Microsoft Visual Studio 2019", включающая в себя интегрированную среду разработки, редактор исходного кода, встроенный отладчик. Многие другие инструменты возможно получить благодаря подключению плагинов – сторонних расширений. Был выбран по причине поддержки языка программирования C#. Также это наиболее изученная разработчиком платформа, что позволяет быстрее разрабатывать приложения не тратя время на изучения нового материала.

В качестве языка разработки был выбран язык программирования C# – объектно-ориентированный язык программирования. Разработан в 1998 — 2001 годах группой инженеров компании Microsoft под руководством Андерса

Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы "Microsoft .NET Framework".

Достоинства C# по книге Биллига:

- создавался параллельно с каркасом ".Net Framework" и в полной мере учитывает все его возможности - как FCL, так и CLR;

- является полностью объектно-ориентированным языком, где даже типы, встроенные в язык, представлены классами;

- является мощным объектным языком с возможностями наследования и универсализации;

- является наследником языков C/C++, сохраняя лучшие черты этих популярных языков программирования. Общий с этими языками синтаксис, знакомые операторы языка облегчают переход от C++ к C#;

- сохранив основные черты своего великого родителя, язык стал проще и надежнее. Простота и надежность, главным образом, связаны с тем, что на C# хотя и допускаются, но не поощряются такие опасные свойства C++ как указатели, адресация, разыменование, адресная арифметика;

- благодаря каркасу ".Net", ставшему надстройкой над операционной системой, программисты C# получают те же преимущества работы с виртуальной машиной, что и программисты Java;

- мощная библиотека каркаса поддерживает удобство построения различных типов приложений на C#, позволяя легко строить Web-службы, другие виды компонентов, достаточно просто сохранять и получать информацию из базы данных и других хранилищ данных;

- реализация, сочетающая построение надежного и эффективного кода, является немаловажным фактором, способствующим успеху C#.

Выделение и объединение лучших идей современных языков программирования делает язык C# не просто суммой их достоинств, а языком программирования нового поколения.

"Windows Presentation Foundation" (WPF) — аналог WinForms, система для построения клиентских приложений Windows с визуально привлекательными возможностями взаимодействия с пользователем, графическая (презентационная) подсистема в составе ".NET Framework" (начиная с версии 3.0), использующая язык XAML.

"SQL Server Management Studio" – утилита из "Microsoft SQL Server 2019" и более поздних версий для конфигурирования, управления и администрирования всех компонентов "Microsoft SQL Server". Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

## **2. ПРОЕКТНАЯ ЧАСТЬ**

В данной части отчета по учебной практике представлено подробное описание разработанного приложения, соответствующего поставленной задаче в техническом задании к учебной практике.

### **2.1. Постановка задачи**

Информационная система «Дари добро» – прикладное программное обеспечение, использующееся для предоставления информации о волонтерских акциях, упрощения формирования списка участников волонтерской акции, а также предназначена для решения следующих задач:

- редактирование информации об акции: название акции, описание акции, тип акции, статус акции, район проведения акции, сроки проведения акции и количество участников акции;

- редактирование информации о волонтере: фамилия имя и отчество волонтера, электронная почта и номер телефона, район работы волонтера, дату рождения и фотография волонтера.

Основными целями внедрения системы являются:

- упрощение предоставления информации о волонтерских акциях организаторов;

- автоматизация процесса оформления волонтерской акции;

- автоматизация процессов добавления и удаления волонтера из состава акции;

- минимизация человеческого фактора при ведении документов.

Исходная информация об акциях в систему поступает из документов об утверждении проведения волонтерской акции со стороны заказчика.

Входными данными являются следующие:

- название акции;

- описание акции;

- тип и статус акции;
- район проведения акции;
- сроки проведения акции;
- количество участников акции.

Исходная информация о волонтерах в систему поступает из документов об утверждении состава волонтеров менеджера волонтеров. Входными данными являются следующие:

- фио волонтера;
- дата рождения волонтера;
- номер телефона волонтера;
- электронная почта волонтера;
- район работы волонтера;
- фотография волонтера.

Рассмотрим определение прецедентов (вариантов использования).

Система требуется прежде всего следующим заинтересованным лицам:

- организатор;
- менеджер волонтеров;
- волонтер.

Определяется перечень функциональных подсистем, их назначение и основные характеристики, требования к числу уровней иерархии и степени централизации системы.

Смежными системами для «Волонтеры» являются:

- информационные системы ввода, хранения, поиска и оперативной обработки информации о волонтерах;
- информационная система управления волонтерами, необходимая менеджерам волонтеров, содержащая следующие данные о волонтере: ФИО, номер телефона, электронную почту, дату рождения, район работы.

Источниками данных для системы должны быть:

- СУБД MS SQL;

- таблицы Excel.

Входными данными являются следующий набор данных:

- ФИО клиента;
- номер телефона волонтера;
- логин и пароль.

Система должна функционировать в многопользовательском режиме и давать возможность:

- просматривать записи базы данных, в том числе и с помощью различных фильтров;
- добавлять новые записи и изменять существующие;
- удалять записи.

## **2.2. Функционально-логическая структура проекта**

В данном разделе отчета по учебной практике представлены различные диаграммы на языке UML, описывающие функциональную и логическую структуру разрабатываемой программы учебной практики.

### **2.2.1. Диаграмма вариантов использования**

В этом разделе приведена UML-диаграмма типа вариантов использования или диаграмма прецедентов, представленная на рисунке 1. Диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Прецедент - возможность моделируемой системы, благодаря которой пользователь может получить конкретный, измеримый и нужный ему результат.

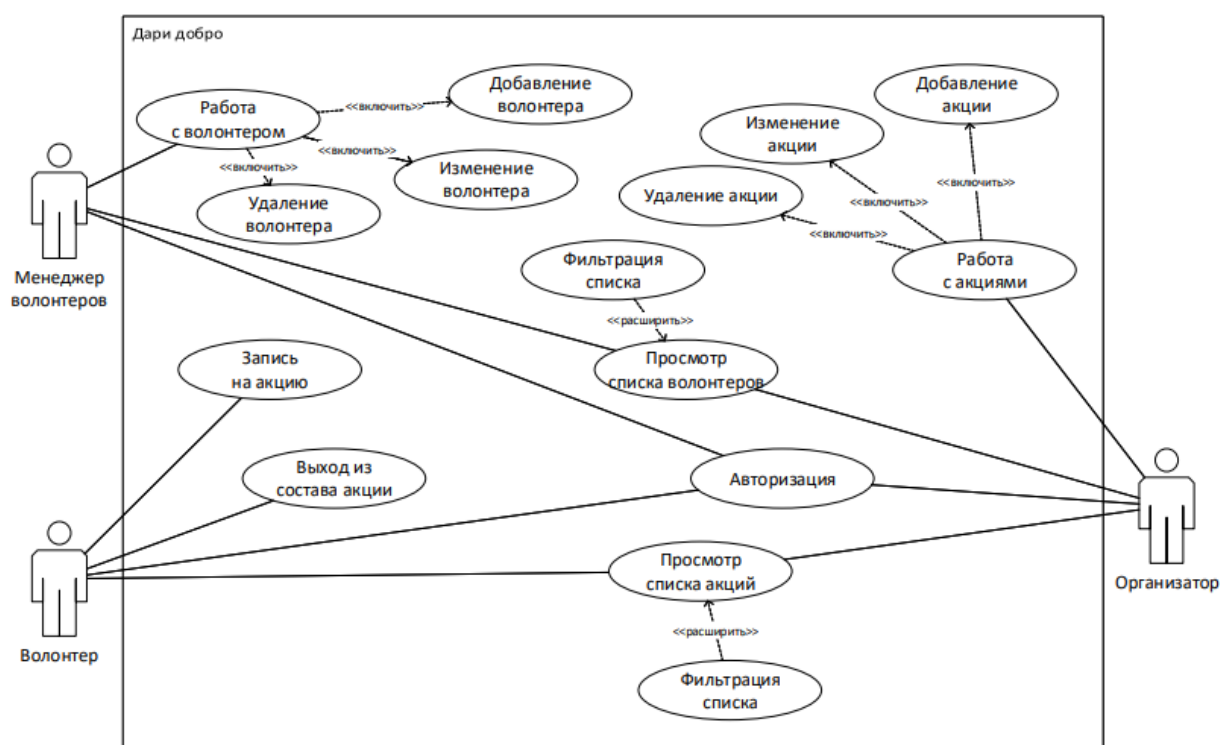


Рисунок 1 – Диаграмма вариантов использования

## 2.2.2. Диаграмма деятельности

Диаграмма деятельности (англ. "activity diagram") – UML-диаграмма, на которой показаны действия, состояния которых описаны на диаграмме состояний. Под деятельностью понимается спецификация исполняемого поведения в виде координированного последовательного и параллельного выполнения подчинённых элементов – вложенных видов деятельности и отдельных действий, соединённых между собой потоками, которые идут от выходов одного узла ко входам другого.

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

Была разработана диаграмма деятельности для прецедента «Редактирования акции», которая осуществляется ролью «Организатор». Данная диаграмма представлена на рисунке 2.

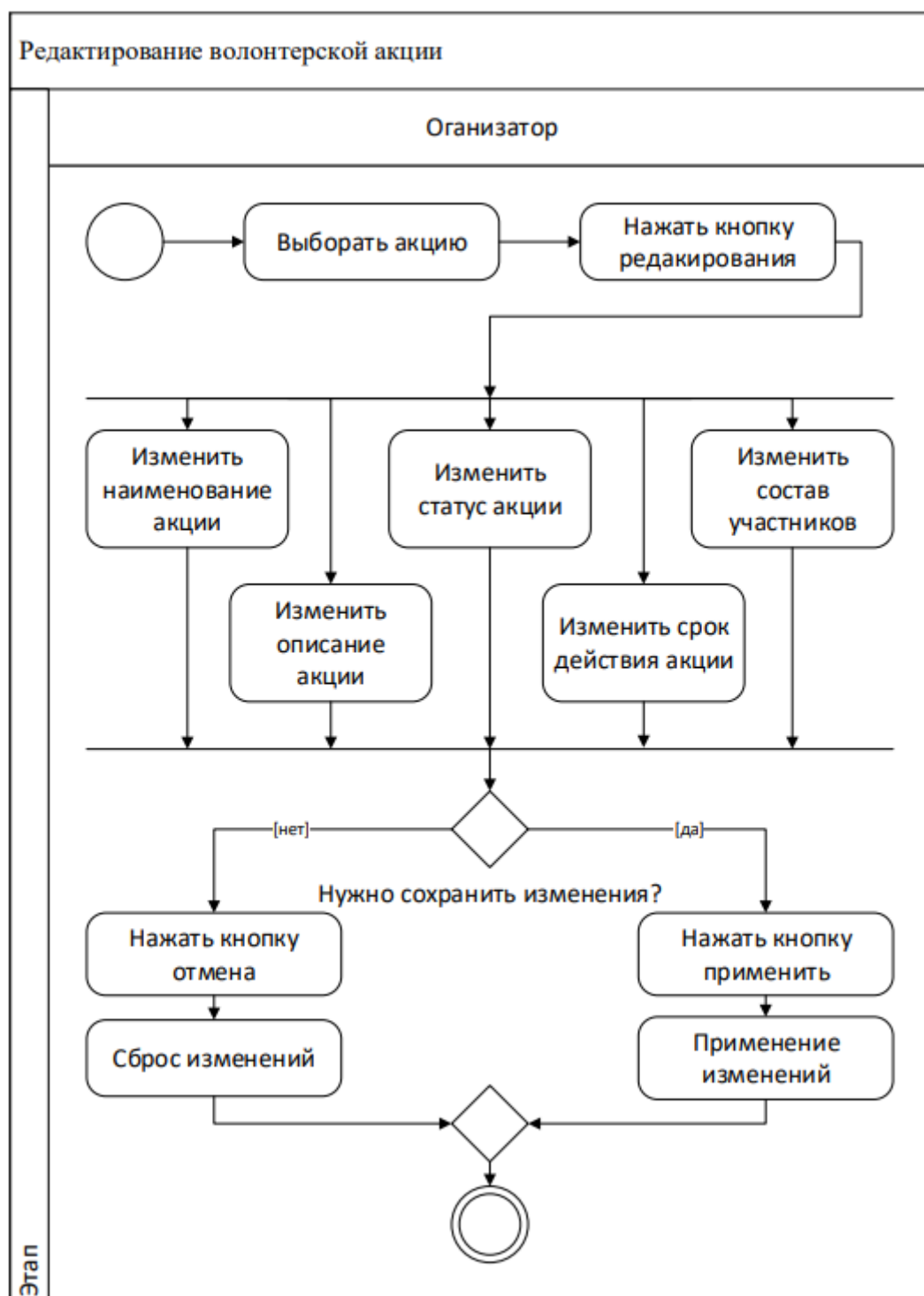


Рисунок 2 – Диаграмма деятельности

### 2.3. Логическая модель данных

В основе разрабатываемого приложения находится база данных, как структурированное хранилище информации разрабатываемой системы.



Удобно логическую модель данных описывать с применением UML-языка на основе ED-диаграммой.

В системе есть три сущности: «Пользователи», «Акции» и «Районы».

Таблица "User" состоит из полей идентификатора, Фамилия, Имя, Отчество, email-адрес, логин, пароль, фотография, дата рождения и ID роли (связанная таблица, описана ниже).

Таблица "UserDistrict" состоит из полей идентификатора пользователя (связанная таблица "User") и идентификатора района (связанная таблица "District").

Таблица "Role" состоит из полей идентификатора записи и названия роли.

Таблица "Action" состоит из полей: идентификатор, название акции, описание акции, тип акции (связанная таблица "ActionType"), статус акции (связанная таблица "ActionStatus"), максимальное количество участников, дата регистрации, дата начала, дата окончания.

Таблица "ActionType" состоит из полей идентификатора записи и названия типа.

Таблица "ActionStatus" состоит из полей идентификатора записи и названия статуса.

Таблица "District" состоит из полей идентификатора записи и названия района.

Таблица "ActionDistrict" состоит из полей идентификатора акции (связанная таблица "Action") и идентификатора района (связанная таблица "District").

Таблица "ActionOrganizer" состоит из полей идентификатора акции (связанная таблица "Action") и идентификатора пользователя (связанная таблица "User").

Таблица "ActionParticipation" состоит из полей идентификатора акции (связанная таблица "Action") и идентификатора пользователя (связанная таблица "User").

Все таблицы базы данных находятся в третьей нормальной форме. Третья нормальная форма (3НФ) – это стандартная форма нормализации в базах данных, которая помогает избежать избыточности данных и устранить проблемы, связанные с неоднозначностью данных. Третья нормальная форма достигается тогда, когда все неключевые атрибуты в каждой таблице зависят только от первичного ключа и не зависят от других неключевых атрибутов. В более простых терминах, каждый неключевой атрибут должен зависеть только от первичного ключа таблицы и никак не зависеть от других атрибутов. На рисунке 3 представлена ER-диаграмма разработанной базы данных.

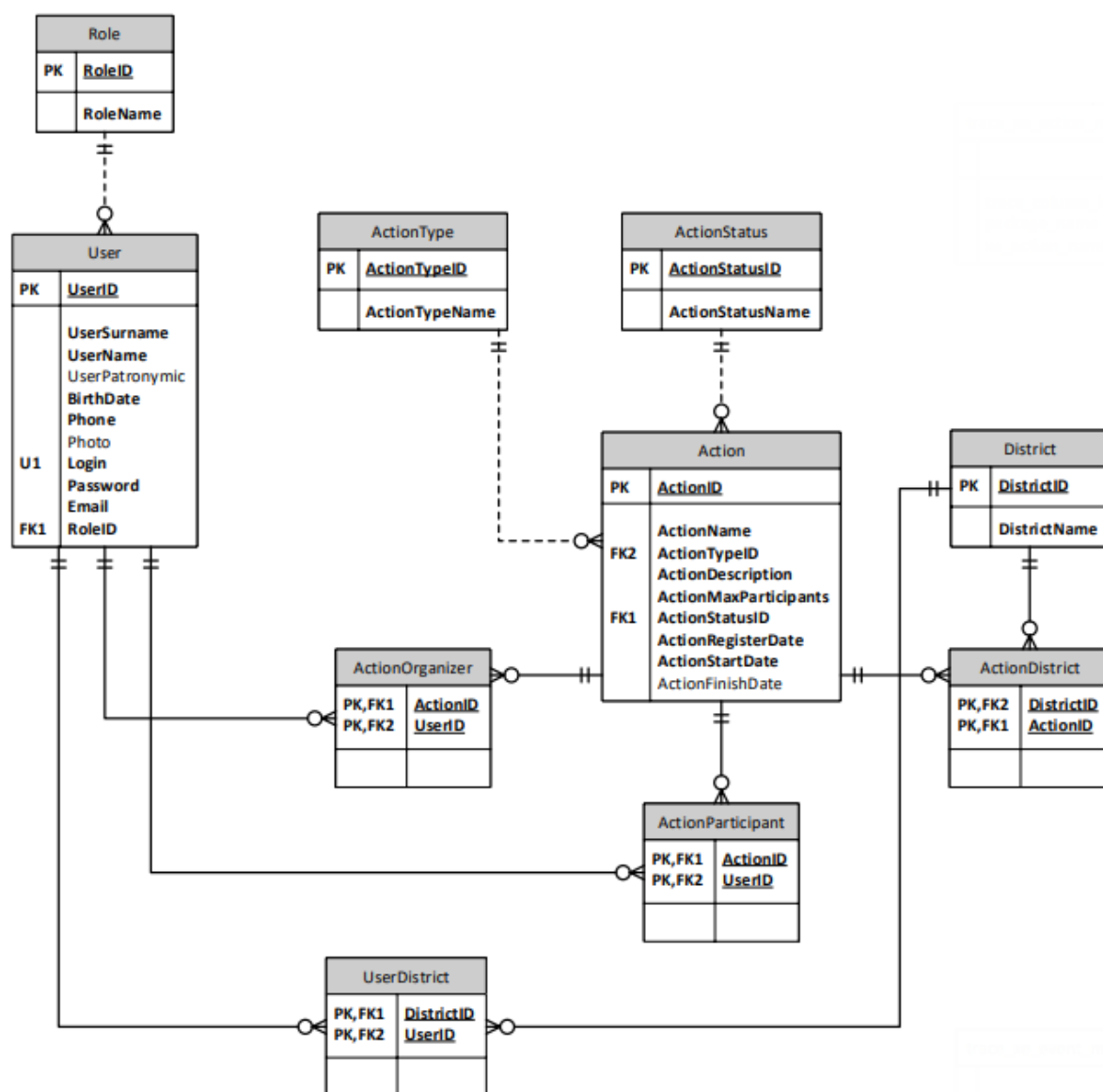


Рисунок 3 – ER диаграмма

Словарь данных таблицы "Action" представлен в таблице 1.

**Таблица 1 – словарь данных таблицы "Action"**

Key	FieldName	Data type	Required?	Notes
PK	ActionID	int	Y	
	ActionName	nvarchar(100)	Y	
FK	ActionTypeID	int	Y	
	ActionDescription	nvarchar(MAX)	Y	
	ActionMAXParticipations	int	Y	
FK	ActionStatusID	int	Y	
	ActionRegisterDate	date	Y	

Словарь данных таблицы "User" представлен в таблице 2.

**Таблица 2 – словарь данных таблицы "User"**

Key	FieldName	Data type	Required?	Notes
PK	UserID	int	Y	
	UserSurname	nvarchar(50)	Y	
	UserName	nvarchar(50)	Y	
	UserPatronymic	nvarchar(50)	N	
	BirthDate	date	Y	
	Phone	nvarchar(16)	Y	
	Photo	nvarchar(MAX)	N	
	Login	varchar(50)	Y	Unique field
	Password	varchar(50)	Y	
	Email	varchar(MAX)	Y	
FK	RoleID	int	Y	

Словарь данных таблицы "UserDistrict" представлен в таблице 3.

**Таблица 3 – словарь данных таблицы "UserDistrict "**

Key	FieldName	Data type	Required?	Notes
PK, FK	DistrictID	int	Y	PK, FK
PK, FK	UserID	int	Y	PK, FK

Словарь данных таблицы "ActionOrganizer" представлен в таблице 4.

**Таблица 4 – словарь данных таблицы "ActionOrganizer "**

Key	FieldName	Data type	Required?	Notes
PK, FK	ActionID	int	Y	
PK, FK	UserID	int	Y	

Словарь данных таблицы "ActionParticipant" представлен в таблице 5.

**Таблица 5 – словарь данных таблицы "ActionParticipant"**

Key	FieldName	Data type	Required?	Notes
PK, FK	ActionID	int	Y	
PK, FK	UserID	int	Y	

Словарь данных таблицы "District" представлен в таблице 6.

**Таблица 6 – словарь данных таблицы "District"**

Key	FieldName	Data type	Required?	Notes
PK	DistrictID	int	Y	
	DistrictName	nvarchar(100)	Y	

Словарь данных таблицы "ActionDistrict" представлен в таблице 7.

**Таблица 7 – словарь данных таблицы "ActionDistrict"**

Key	FieldName	Data type	Required?	Notes
PK, FK	DistrictID	int	Y	
PK, FK	ActionID	int	Y	

Словарь данных таблицы "Role" представлен в таблице 8.

**Таблица 8 – словарь данных таблицы "Role"**

Key	FieldName	Data type	Required?	Notes
PK	RoleID	int	Y	
	RoleName	nvarchar(100)	Y	

Словарь данных таблицы "ActionStatus" представлен в таблице 9.

**Таблица 9 – словарь данных таблицы "ActionStatus"**

Key	FieldName	Data type	Required?	Notes
-----	-----------	-----------	-----------	-------

PK	ActionStatusID	int	Y	PK
	ActionStatusName	nvarchar(100)	Y	

Словарь данных таблицы "ActionType" представлен в таблице 10.

**Таблица 10 – словарь данных таблицы "ActionType"**

Key	FieldName	Data type	Required?	Notes
PK	ActionTypeID	int	Y	
	ActionTypeName	nvarchar(100)	Y	

Как продемонстрировано в таблицах словарей данных, все таблицы связаны между собой при помощи внешних ключей. Внешний ключ – это атрибут (или набор атрибутов) в таблице, который связывает данные в этой таблице с данными в другой таблице. Он используется для установления связи между двумя таблицами в базе данных. Внешний ключ определяет связь между двумя таблицами на уровне данных, и обычно он ссылается на первичный ключ в другой таблице. Таким образом, внешний ключ обеспечивает целостность ссылочной целевой таблицы, что означает, что ссылочная целевая таблица не может содержать записи, которые не связаны с другой таблицей через ее первичный ключ.

Словарь данных (data dictionary) – это коллекция данных, описывающая структуру, содержание, характеристики и связи между различными элементами базы данных. Он представляет собой централизованное хранилище информации о всех объектах, используемых в базе данных, включая таблицы, представления, индексы, ограничения, процедуры, функции и многое другое.

## **2.4. Разработка программных модулей**

В данном разделе отчета представлены действия, необходимые для программной реализации разрабатываемого приложения, а также требования к разработке пользовательского интерфейса.

### **2.4.1. Пользовательский интерфейс**

Не допускаются орфографические и грамматические ошибки.

Все экранные формы пользовательского интерфейса должны иметь заголовок с логотипом logo.png (расположен в ресурсах). Нельзя искажать логотип (не изменяйте изображение, его пропорции, цвет).

Также для приложений должна быть установлена иконка.

В случае возникновения ошибок система оповещает пользователя всплывающими окнами с информацией об ошибке.

## **2.5. Руководство пользователя**

Для начала работы необходимо запустить exe файл – GiveGood.exe.

### **2.5.1. Форма авторизации**

Первое окно, открывающееся при запуске приложения это окно авторизации. Для авторизации необходимо ввести логин в поле ввода логина, ввести пароль в поле ввода пароля и нажать кнопку входа. При успешной авторизации откроется окно меню. При неуспешной авторизации появится окно сообщения о неуспешной авторизации. Процесс авторизации представлен на рисунке 4.

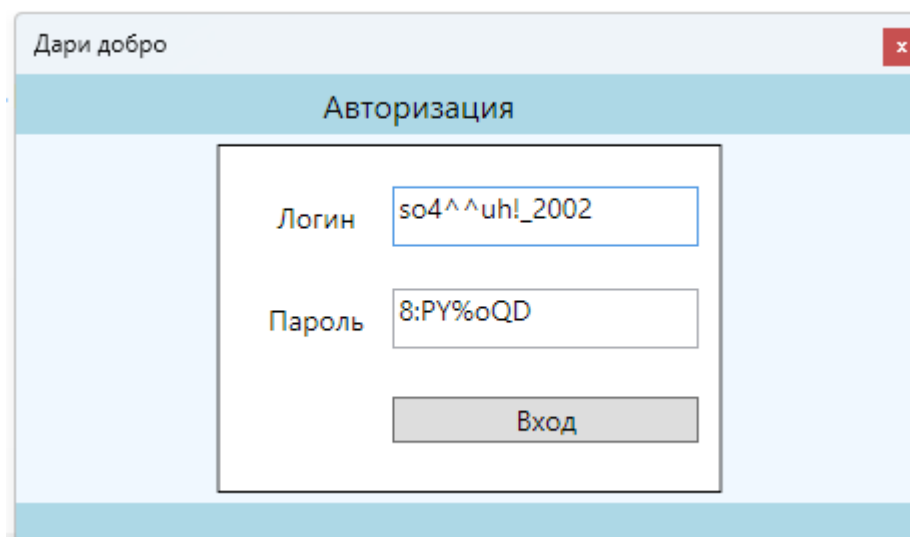


Рисунок 4 – Окно авторизации

Результат авторизации представлен на рисунке 5.

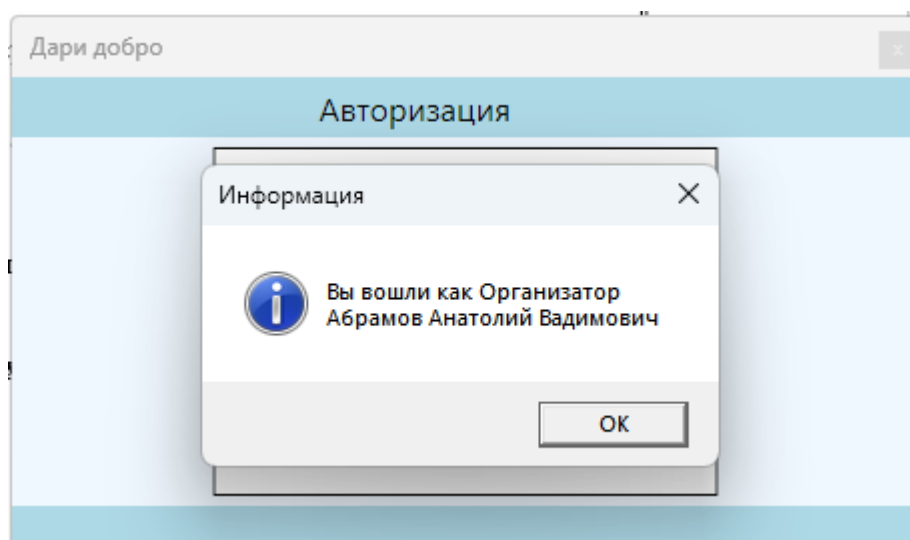


Рисунок 5 – Результат авторизации

### 2.5.2. Форма меню

При успешной авторизации открывается форма меню. При открытии формы меню отображается профиль пользователя. Профиль пользователя представлен на рисунке 6.

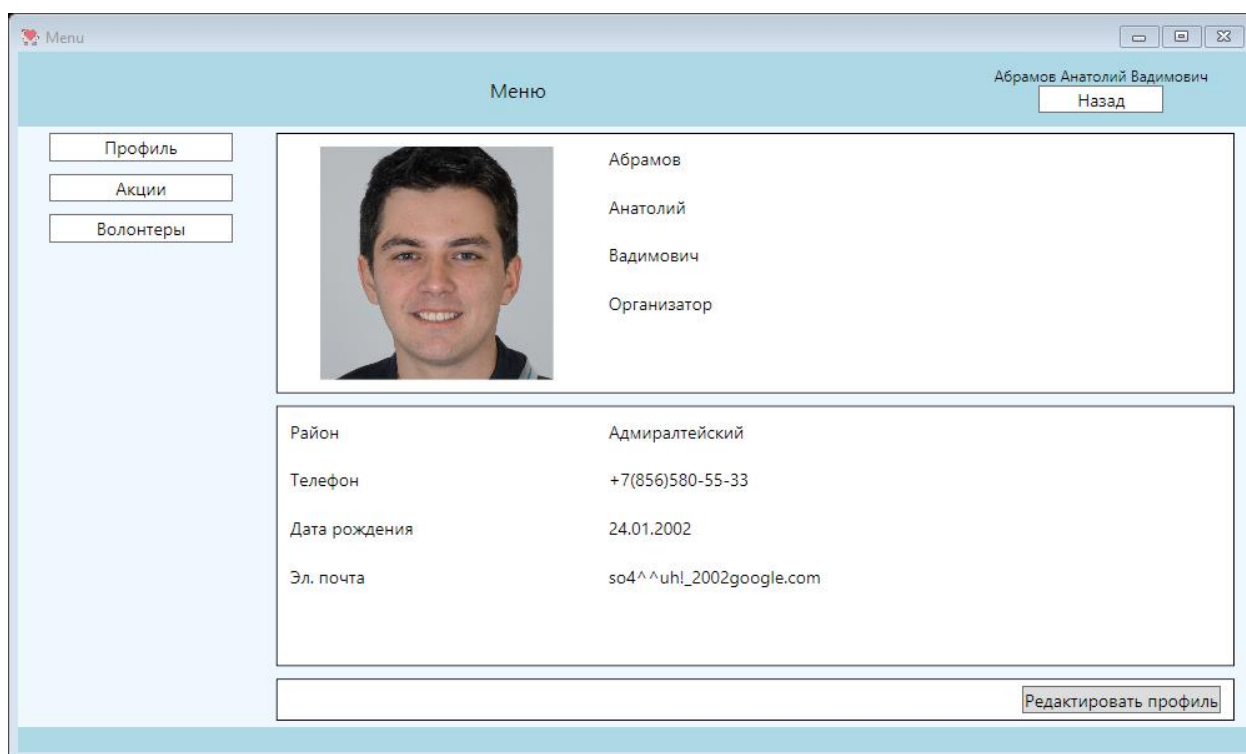


Рисунок 6 – Профиль пользователя

При нажатии на кнопку акции отображается форма списка акций, представленная на рисунке 7.

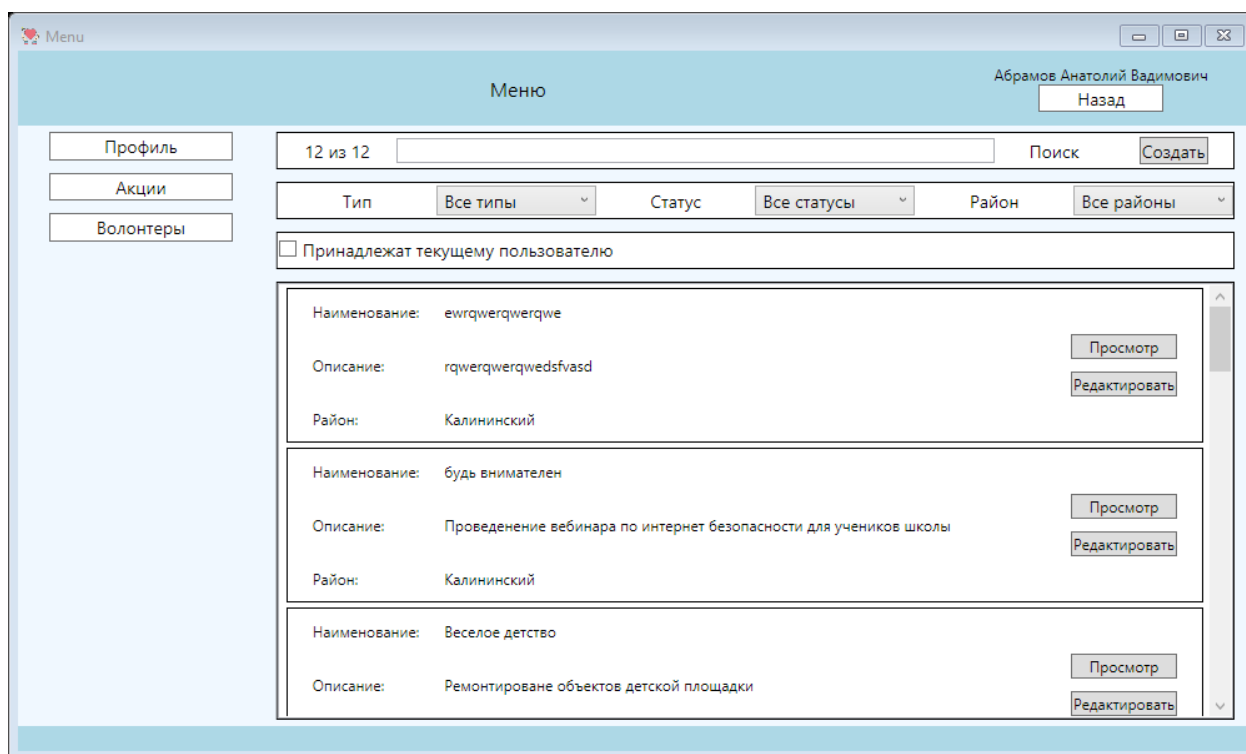


Рисунок 7 – Список акций



При нажатии на кнопку создания новой акции или кнопку редактирования акции откроется форма редактирования акции. При нажатии на кнопку просмотра акции откроется окно просмотра акции. При нажатии на кнопку «Волонтеры» отображается форма списка волонтеров, представленная на рисунке 8.

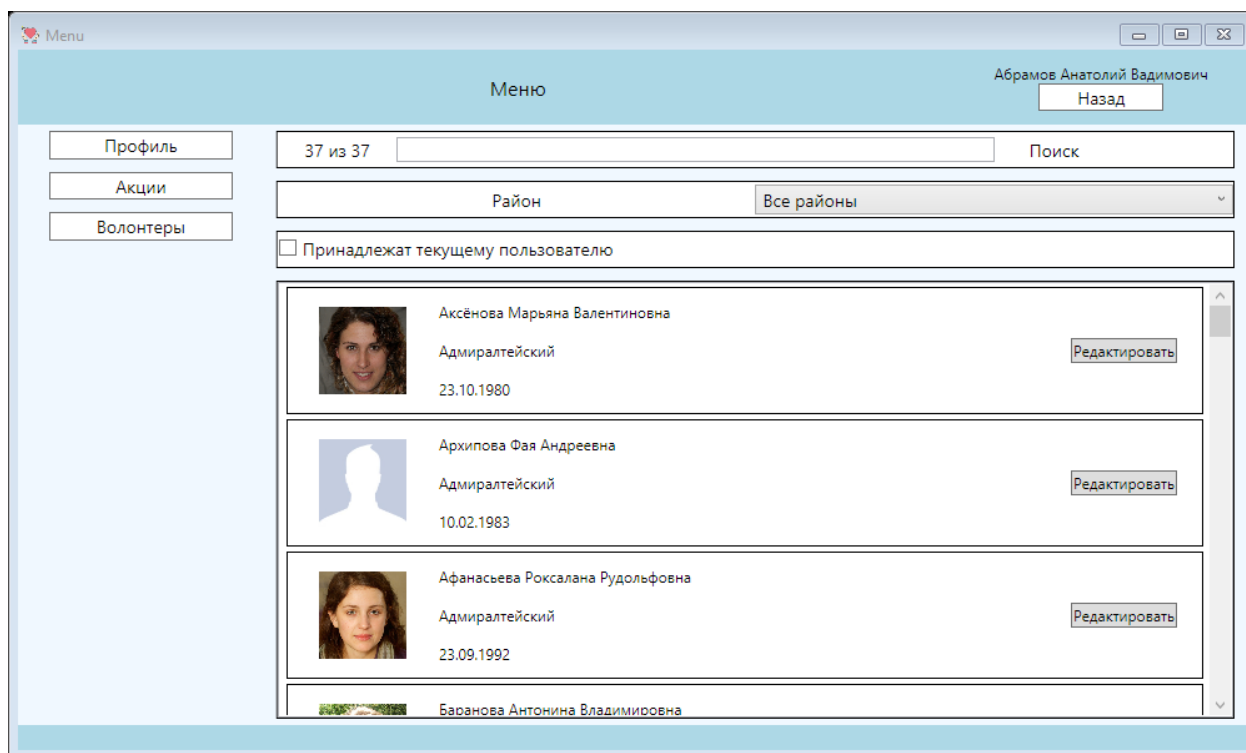


Рисунок 8 – Список волонтеров

При нажатии кнопки «редактировать профиль» откроется форма редактирования профиля пользователя. При нажатии кнопки «назад» произойдет выход из учетной записи и откроется форма авторизации.

### 2.5.3. Форма просмотра акции

Действия описываемые далее требуют авторизации пользователя с ролью организатор. Для просмотра акции необходимо нажать на кнопку просмотра акции. Форма просмотра акции представлена на рисунке 9.

ActionView

Просмотр акции

Абрамов Анатолий Вадимович

Назад

Идентификатор	7		
Наименование	Проект "чистота 4"		
Описание	Убрать мусор на территории парка		
Район	Невский	Дата регистрации	10.03.2023
Тип	Уборка мусора	Дата начала	13.03.2023
Статус	Действует	Дата окончания	
Максимальное количество участников	10	Количество участников	5

Лыткин	Тимофей	Семенович
Нестеров	Осип	Эдуардович
Орехова	Луиза	Васильевна
Панова	Дания	Мироновна
Шарапова	Айлин	Ростиславовна

Рисунок 9 – Форма просмотра акции

#### 2.5.4. Форма редактирования акции

Действия описываемые далее требуют авторизации пользователя с ролью организатор. Для добавления новой акции необходимо нажать на кнопку создания новой акции, затем ввести данные об акции и нажать на кнопку сохранения. Форма редактирования акции представлена на рисунке 10.

The screenshot shows a software window titled 'ActionEdit' with a light blue header. The header contains the text 'Изменение акции' (Edit Action) and the user's name 'Абрамов Анатолий Вадимович'. A 'Назад' (Back) button is located in the top right of the header. The main form area is divided into several sections. The first section contains 'Идентификатор' (Identifier) with the value '16' and 'Наименование' (Name) with the value 'Новая акция'. The second section is a large text area for 'Описание' (Description) with the text 'Описание новой акции'. The third section contains several input fields: 'Район' (District) with 'Адмиралтейский', 'Тип' (Type) with 'Уборка мусора', 'Статус' (Status) with 'Сбор участников', and 'Максимальное количество участников' (Maximum number of participants) with '10'. To the right of these are date pickers for 'Дата регистрации' (Registration date) set to '24.03.2023' and 'Дата начала' (Start date) set to '25.03.2023'. There is also an unchecked checkbox for 'Дата окончания' (End date). A 'Сохранить' (Save) button is positioned at the bottom right of the form area.

Идентификатор	16		
Наименование	Новая акция		
Описание	Описание новой акции		
Район	Адмиралтейский	Дата регистрации	24.03.2023
Тип	Уборка мусора	Дата начала	25.03.2023
Статус	Сбор участников	<input type="checkbox"/> Дата окончания	
Максимальное количество участников	10	Сохранить	

Рисунок 10 – Форма редактирования акции

Новая акция отобразится в списке акций. Список акций представлен на рисунке 11.

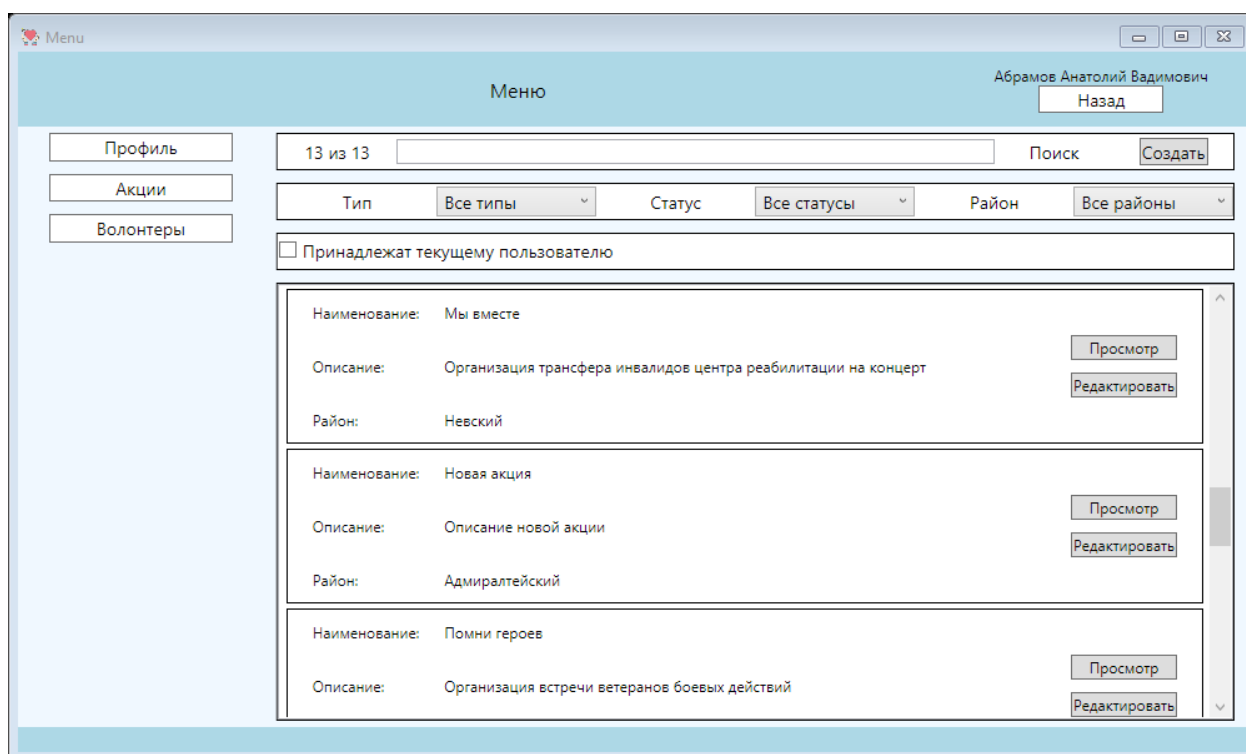
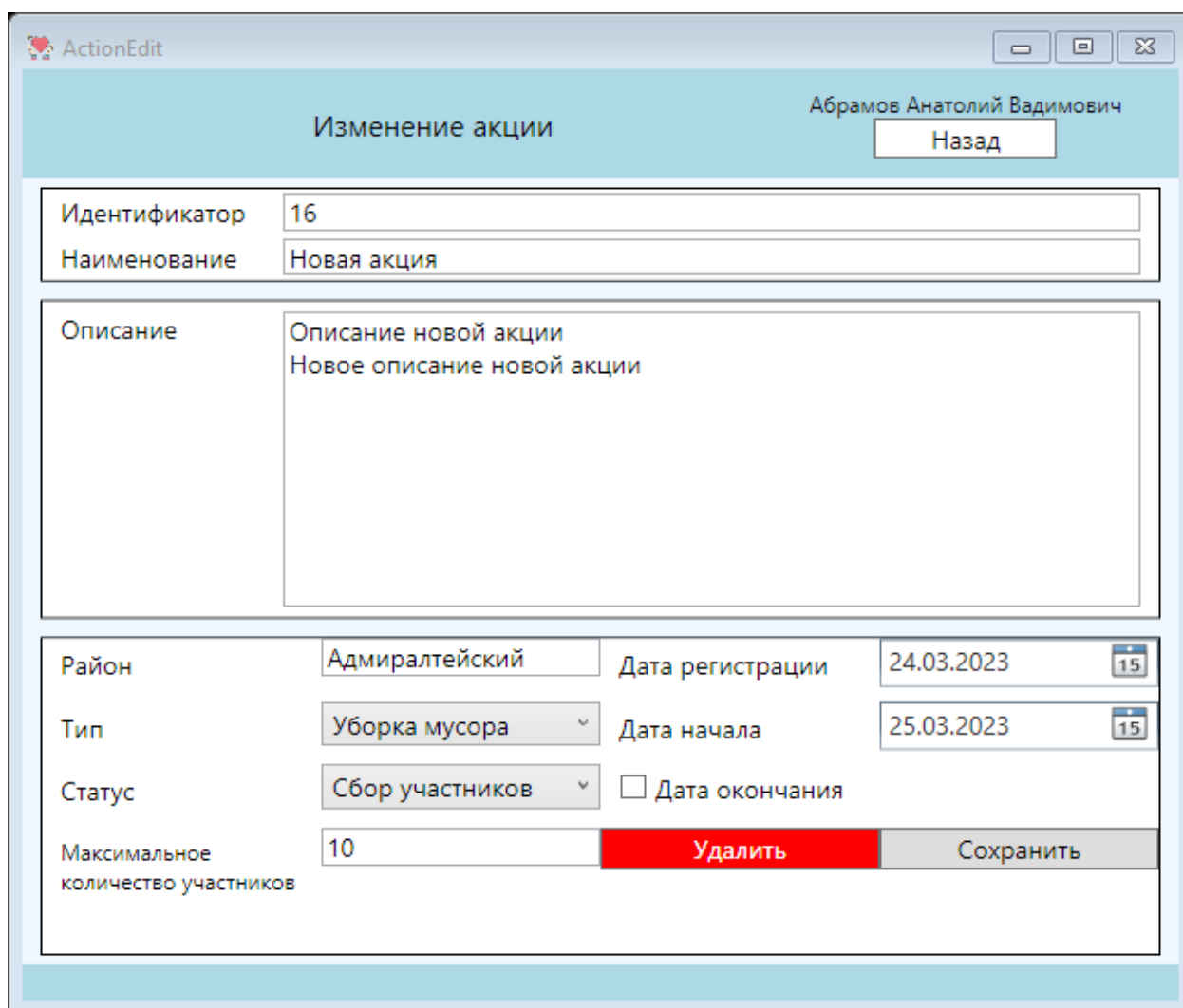


Рисунок 11 – Список акций

Для редактирования акции необходимо нажать на кнопку редактирования акции, затем изменить данные об акции и нажать на кнопку сохранения. Редактирование акции продемонстрировано на рисунке 12.



ActionEdit

Изменение акции

Абрамов Анатолий Вадимович

Назад

Идентификатор: 16

Наименование: Новая акция

Описание: Описание новой акции  
Новое описание новой акции

Район: Адмиралтейский

Дата регистрации: 24.03.2023

Тип: Уборка мусора

Дата начала: 25.03.2023

Статус: Сбор участников

Дата окончания: ☐

Максимальное количество участников: 10

Удалить

Сохранить

Рисунок 12 – Редактирование акции

После сохранения изменений, для возврата к списку акций необходимо нажать кнопку «назад». Измененная акция отобразится в списке акций. Список акций представлен на рисунке 13.

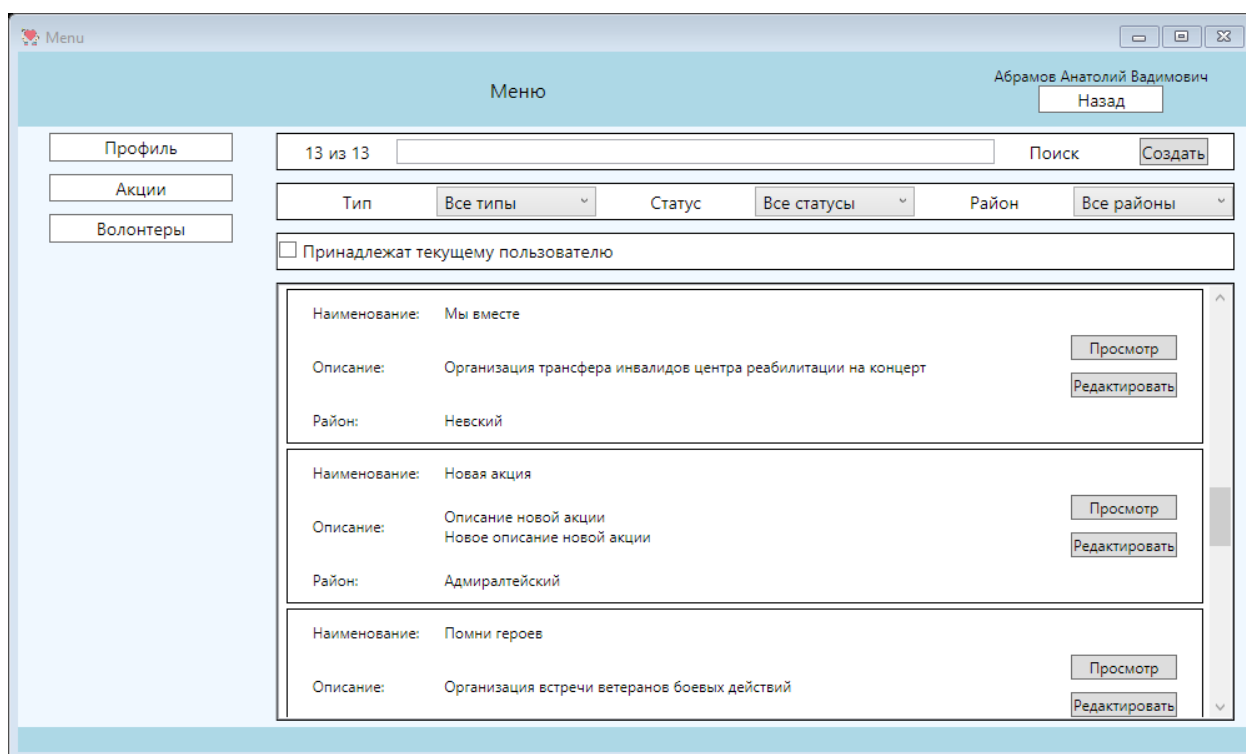


Рисунок 13 – Список акций

Для удаления акции необходимо нажать на кнопку редактирования акции, затем нажать кнопку удаления акции и подтвердить действие. Удаление акции продемонстрировано на рисунке 14.

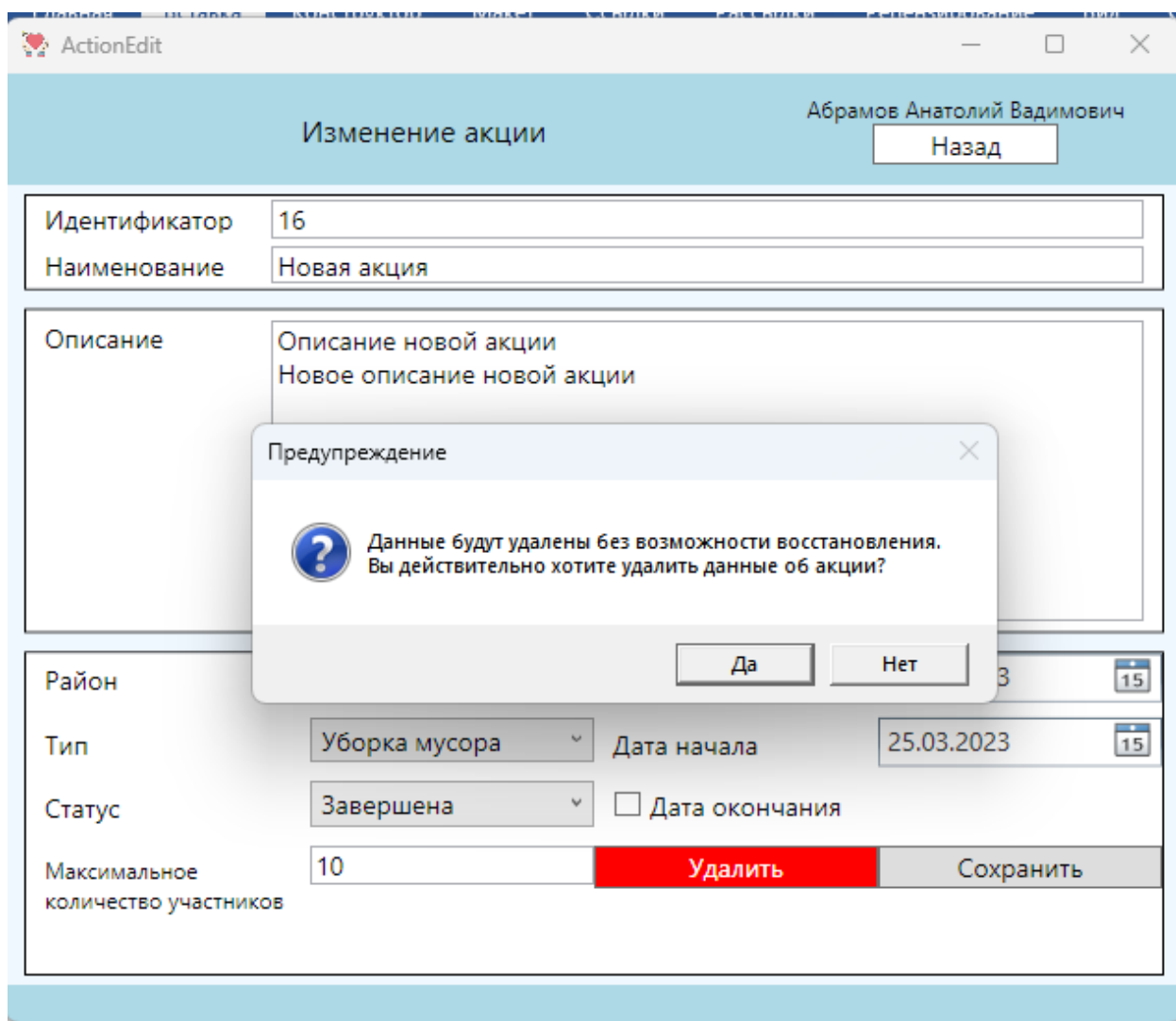


Рисунок 14 – Удаление акции

После удаления, акция исчезнет из списка акций. Список акций представлен на рисунке 15.

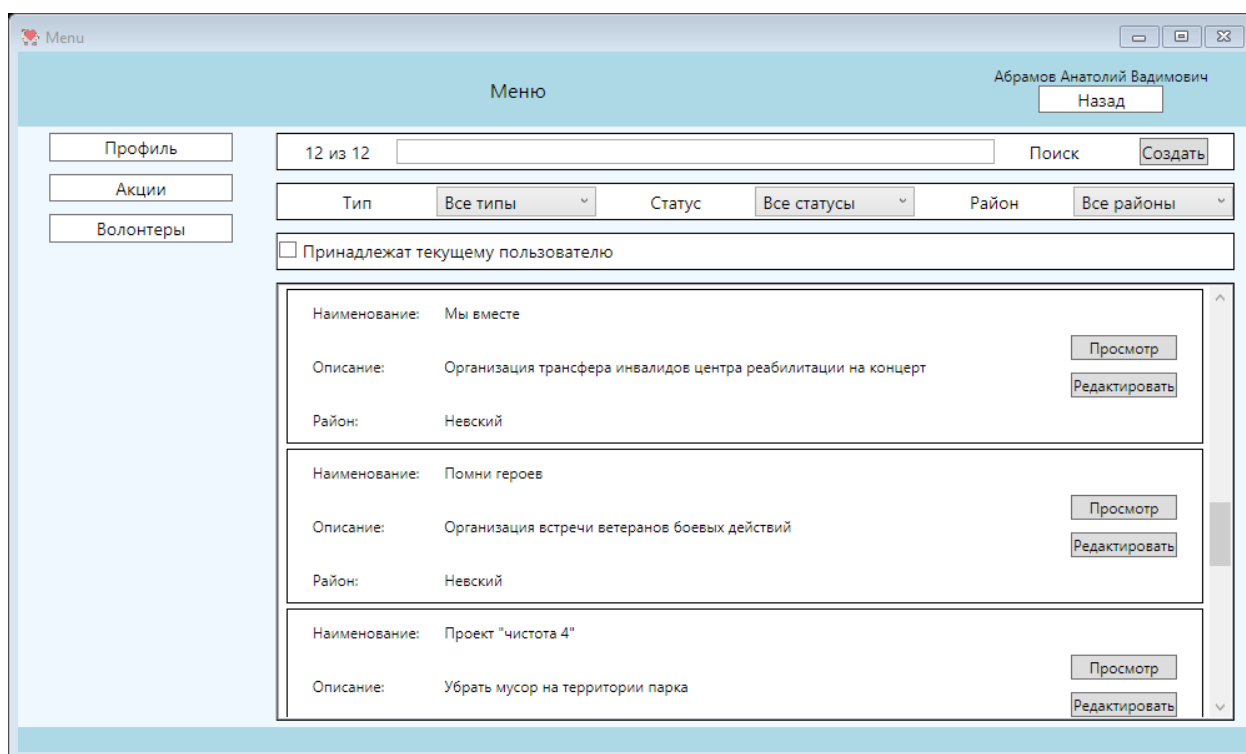


Рисунок 15 – Список акций

### 2.5.5. Форма редактирования волонтера

Действия описываемые далее требуют авторизации пользователя с ролью менеджер волонтеров. Для просмотра волонтера, на форме списка волонтеров необходимо нажать на кнопку редактирования волонтера. Просмотр волонтера продемонстрирован на рисунке 16.



ProfileEdit

Редактирование профиля

Королёв Гавриил Данилович

Назад

Выбрать фото

Фамилия: Аксёнова

Имя: Марьяна

Отчество: Валентиновна

Роль: Волонтер

Логин: jgmwhn&!\_1980

Пароль: Ag:Y)\*Q"

Район: Адмиралтейский

Телефон: +7(076)868-81-18

Дата рождения: 23.10.1980

Эл. почта: jgmwhn&!\_1980yandex.ru

Сохранить Удалить

Рисунок 16 – Просмотр волонтера

Для добавления нового волонтера необходимо нажать на кнопку создания нового волонтера, затем ввести данные волонтера и нажать на кнопку сохранения. Добавление волонтера продемонстрировано на рисунке 17.

ProfileEdit

Редактирование профиля

Королёв Гавриил Данилович

Назад

Выбрать фото

Фамилия: Алексейев  
 Имя: Иван  
 Отчество: Дмитриевич  
 Роль: Волонтер  
 Логин: Megalvan449  
 Пароль: YaKrut\_999

Район: Адмиралтейский  
 Телефон: +7(343)996-65-22  
 Дата рождения: 16.06.1995  
 Эл. почта: Ivan44@mail.ru

Сохранить

Рисунок 17 – Добавление волонтера

Новая волонтер отобразится в списке волонтеров. Список волонтеров представлен на рисунке 18.

Menu

Меню

Королёв Гавриил Данилович

Назад

Профиль

38 из 38

Поиск

Создать

Волонтеры

Район: Все районы

☐ Принадлежат текущему пользователю

	Аксёнова Марияна Валентиновна Адмиралтейский 23.10.1980	Редактировать
	Алексеев Иван Дмитриевич Адмиралтейский 16.06.1995	Редактировать
	Архипова Фая Андреевна Адмиралтейский 10.02.1983	Редактировать
	Афанасьева Роксалана Рудольфовна	

Рисунок 18 – Список волонтеров

Для редактирования волонтера необходимо нажать на кнопку редактирования волонтера, затем изменить данные о волонтере и нажать на кнопку сохранения. Редактирование волонтера продемонстрировано на рисунке 19.

The screenshot shows a web application window titled 'ProfileEdit'. The main heading is 'Редактирование профиля' (Profile Editing). In the top right corner, the user's name 'Королёв Гавриил Данилович' is displayed, along with a 'Назад' (Back) button. The form is divided into two main sections. The first section on the left contains a profile picture of a man with glasses and a 'Выбрать фото' (Choose photo) button. To the right of the photo are input fields for: 'Фамилия' (Last name) with value 'Алексеев', 'Имя' (First name) with value 'Иван', 'Отчество' (Patronymic) with value 'Дмитриевич', 'Роль' (Role) with value 'Волонтер' (Volunteer), 'Логин' (Login) with value 'Megalvan449', and 'Пароль' (Password) with value 'YaKrut\_999'. The second section below contains fields for: 'Район' (District) with value 'Адмиралтейский', 'Телефон' (Phone) with value '+7(343)996-65-22', 'Дата рождения' (Date of birth) with value '16.06.1995' and a calendar icon, and 'Эл. почта' (Email) with value 'Ivan44@mail.ru'. At the bottom right of the form are two buttons: 'Сохранить' (Save) and 'Удалить' (Delete).

Рисунок 19 – Редактирование волонтера

После сохранения изменений, для возврата к списку волонтеров необходимо нажать кнопку «назад». Измененный волонтер отобразится в списке волонтеров. Список волонтеров представлен на рисунке 20.

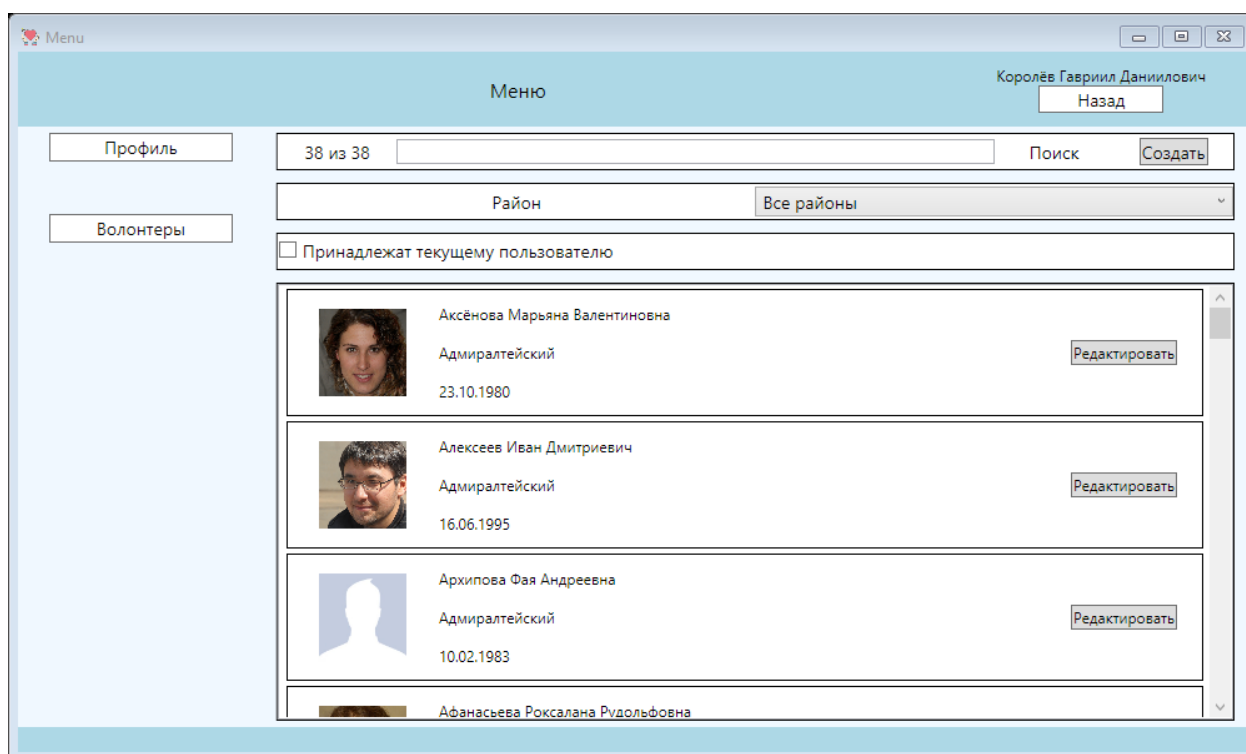


Рисунок 20 – Список волонтеров

Для удаления волонтера необходимо нажать на кнопку редактирования акции, затем нажать кнопку удаления и подтвердить действие. Удаление волонтера продемонстрировано на рисунке 21.

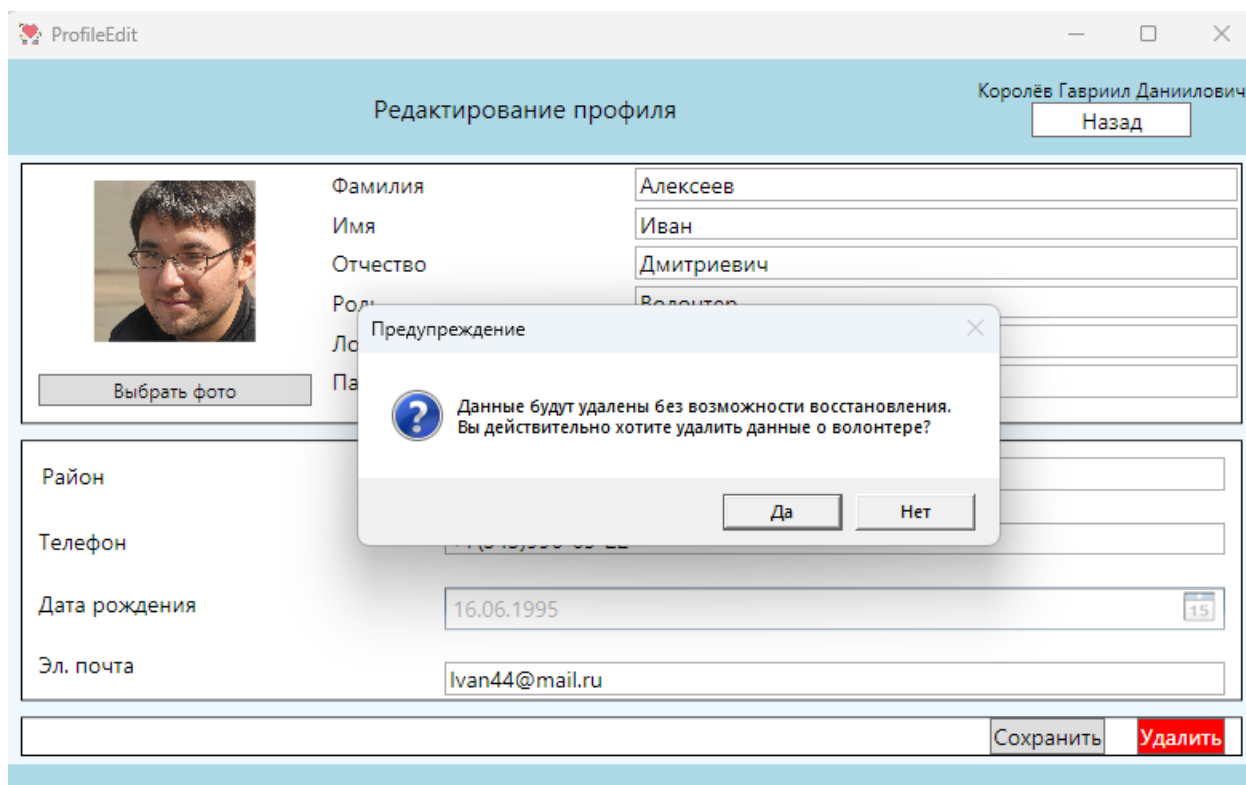


Рисунок 21 – Удаление волонтера

После удаления, волонтер исчезнет из списка волонтеров. Список волонтеров представлен на рисунке 22.

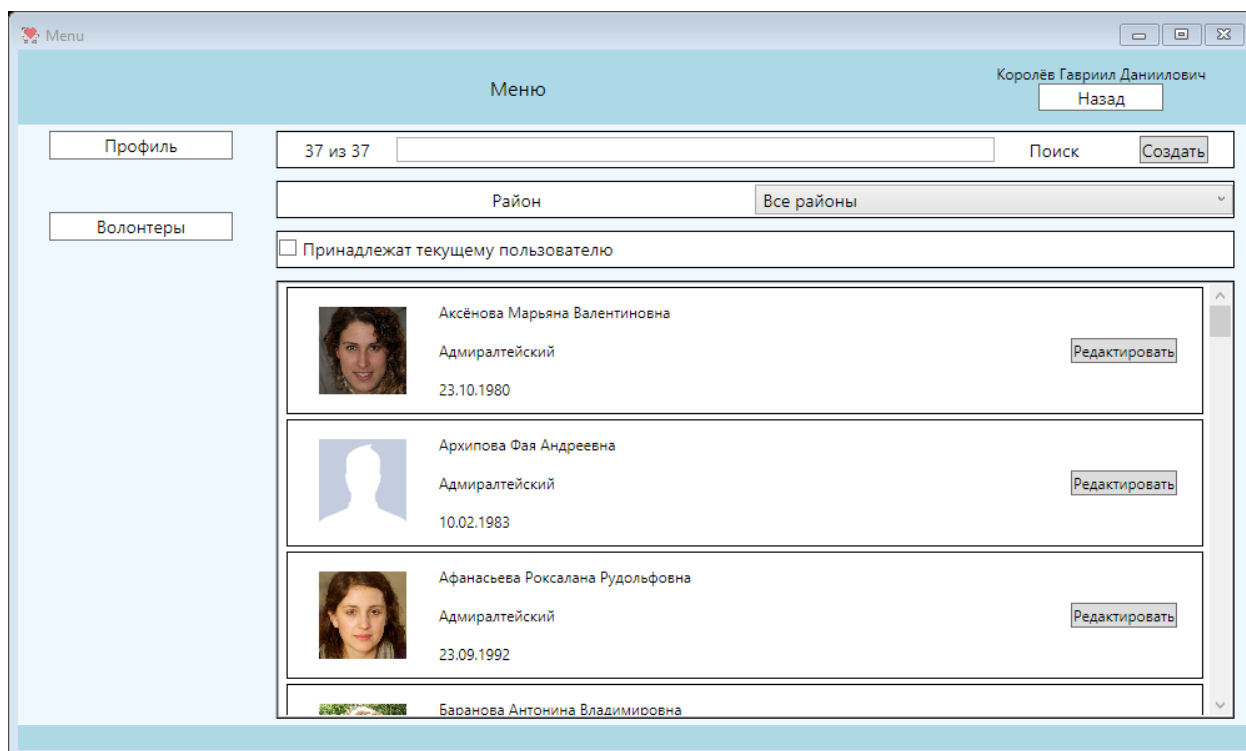


Рисунок 22 – Список волонтеров

### 2.5.6. Форма Просмотра акции для волонтеров

Действия описываемые далее требуют авторизации пользователя с ролью волонтер. Для просмотра акции необходимо нажать на кнопку просмотра акции. Форма просмотра акции представлена на рисунке 23.

ActionView

Просмотр акции

Большакова Светлана Тимофеевна

Назад

Идентификатор	1		
Наименование	Проект "чистота 5"		
Описание	Убрать мусор на территории парка		
Район	Курортный	Дата регистрации	10.03.2023
Тип	Уборка мусора	Дата начала	20.03.2023
Статус	Сбор участников	Дата окончания	
Максимальное количество участников	10	Количество участников	0
Присоединиться			

Рисунок 23 – Просмотр акции волонтером

Для добавления волонтера в состав акции необходимо нажать на кнопку «Присоединиться». Добавление волонтера в состав акции продемонстрировано на рисунке 24.

ActionView

Просмотр акции

Большакова Светлана Тимофеевна

Назад

Идентификатор	1		
Наименование	Проект "чистота 5"		
Описание	Убрать мусор на территории парка		
Район	Курортный	Дата регистрации	10.03.2023
Тип	Уборка мусора	Дата начала	20.03.2023
Статус	Сбор участников	Дата окончания	
Максимальное количество участников	10	Количество участников	1
Покинуть акцию			

Большаякова Светлана Тимофеевна

Рисунок 24 – Добавление волонтера в состав акции

Для удаления волонтера из состава акции необходимо нажать на кнопку «Покинуть акцию». Удаление волонтера из состава акции продемонстрировано на рисунке 25.

The screenshot shows a web application window titled 'ActionView'. The main heading is 'Просмотр акции' (View Action). In the top right corner, the user's name 'Большакова Светлана Тимофеевна' is displayed, along with a 'Назад' (Back) button. The form contains the following fields:

Идентификатор	1		
Наименование	Проект "чистота 5"		
Описание	Убрать мусор на территории парка		
Район	Курортный	Дата регистрации	10.03.2023
Тип	Уборка мусора	Дата начала	20.03.2023
Статус	Сбор участников	Дата окончания	
Максимальное количество участников	10	Количество участников	0

At the bottom of the form, there is a button labeled 'Присоединиться' (Join).

Рисунок 25 – Удаление волонтера из состава акции

### 2.5.7. Выводы по учебной практике

Темой учебной практики связана с разработкой автоматизированной информационной системы для организации проведения волонтерских акций.

Было выполнено:

- анализ предметной области;
- выбор технологий и средств;
- создана функционально-логическая структура проекта;
- разработан пользовательский интерфейс;
- были построены диаграммы проекта;
- реализованы некоторые программные модули, такие как «Авторизация», «Список волонтерских акций», «Список волонтеров», «Работа с волонтерскими акциями», «Работа с волонтерами».

Учебная практика проходила в конце обучения на четвертом курсе и подтвердила приобретенные в ходе учебного процесса профессиональные и общие компетенции по МДК 01.01 «Разработка программных модулей».

В ходе выполнения учебной практике разрабатывалось прикладное

приложение для автоматизации проведения волонтерских акций.

Достоинством данного приложения является его простота использования пользователем, т.к. оно имеет интуитивно понятный стандартизированный интерфейс.

Благодаря данной учебной практике были закреплены полученные знания и практические умения при изучении МДК 01.01 «Разработка программных модулей», более глобально изучена среда программирования Visual Studio, язык программирования C#, технология разработки и программирования баз данных.

В итоге был, получен опыт в работе по данной теме, так же были освоены основные приемы работы с базами данных в среде программирования Visual Studio 19, применена технология Entity Framework, работающая с базами данных. Важным аспектом, было изучение новых компонентов, с помощью которых осуществлялась связь и отображение информации базы данных.

Были реализованы основные способы манипуляции с базами данных, а именно: просмотр, редактирование, поиск и фильтрация по заданным критериям.

База данных, кроме числовых и строковых полей, позволяет хранить и использовать двоичные поля, которые представляются изображениями отдельных записей. Программа снабжена простым интерфейсом, но, тем не менее, были реализованы все необходимые операции с таблицами баз данных.



## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Г. Н. Федорова Разработка, внедрение и адаптация программного обеспечения отраслевой направленности: учебное пособие / Г. Н. Федорова. – Москва : КУРС : ИНФРА-М, 2020 – 336 с. (Среднее профессиональное образование).
2. Гагарина Л. Г. Технология разработки программного обеспечения : учебное пособие / Л. Г. Гагарина, Е. В. Кокорина, Б. Д. Сидорова-Виснадул : под ред. Л. Г. Гагариной. – Москва : ИД «ФОРУМ» : ИНФРА-М, 2021. – 400 с. – (Среднее профессиональное образование).
3. Федорова Г. Н. Осуществление интеграции программных модулей: учебник для студ. учреждений сред. проф. образования / Г. Н. Федорова. 2-е изд., стер. – М. : Издательский центр «Академия», 2018. – 288 с.
4. Перлова О. Н. Проектирование и разработка информационных систем : учебник для студ. учреждений сред. проф. образования / О. П. Ляпина, А. В. Гусева. - 3-е изд., испр. – М. : Издательский центр «Академия», 2020. – 256 с.
5. Зверева В.П. Сопровождение и обслуживание программного обеспечения компьютерных систем : учебник для студ. учреждений сред. проф. образования / В. П. Зверева, А. В. Назаров. – 2-е изд., испр. – М. : Издательский центр «Академия», 2020. -256 с.
6. Голицына О.Л. Основы алгоритмизации и программирования : учебное пособие / О.Л. Голицына, И.И. Попов. — 4-е изд., испр. и доп. — Москва : ФОРУМ : ИНФРА-М, 2021. — 431 с. — (Среднее профессиональное образование).
7. Хорев П.Б. Объектно-ориентированное программирование с примерами на C# : учебное пособие / П.Б. Хорев. — Москва : ФОРУМ : ИНФРА-М, 2021. — 200 с. — (Среднее профессиональное образование).
8. Гуриков С. Р. Введение в программирование на языке Visual C#

: учеб. пособие / С.Р. Гуриков. — М. : ФОРУМ : ИНФРА-М, 2019. — 447 с.  
— (Среднее профессиональное образование).

9. 3      Зубкова Т. М. Федорова технология разработки программного обеспечения: учебное пособие / Т. М. Зубкова. — Санкт-Петербург: Лань, 2019. - 324 с.: ил.
10. <https://metanit.com/> — метанит
11. <https://docs.microsoft.com/ru-ru/> — техническая документация от компании Майкрософт
12. <https://habr.com> — харб
13. <https://nationalteam.worldskills.ru/skills/programmnye-resheniya-dlya-biznesa/> — сайт WorldSkills.