**СодержанИЕ**

[ГЛАВА 1. Описание предметной области и обзор существующих информационных систем для дошкольных образовательных учреждений 7](#_Toc420327027)

[1.1. Описание предметной области. 7](#_Toc420327028)

[1.2. Функциональные особенности объекта управления. 8](#_Toc420327029)

[1.3. Исследования автоматизации процессов 9](#_Toc420327030)

[1.4. Описание существующих информационных систем 10](#_Toc420327031)

[1.4.1. Информационная система публикации и обработки данных о детях в ГОУ детский сад №176 «Белочка» 11](#_Toc420327032)

[1.3.2. Автоматизированная информационная система детского сада «Солнышко» 12](#_Toc420327033)

[1.3.3. Информационная система по учету кадров «Личная карточка» в дошкольном образовательном учреждении 14](#_Toc420327034)

[1.3.4. Сравнительный анализ 15](#_Toc420327035)

[ГЛАВА 2. ОПИСАНИЕ РАЗРАБОТАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ МБДОУ «ДЕТСКИЙ САД С.ЯКОВЛЕВКА» 18](#_Toc420327036)

[2.1. Функциональные возможности информационной системы МБДОУ «Детский сад с.Яковлевка» 18](#_Toc420327037)

[2.2. Входные и выходные данные в информационной системе МБДОУ «Детский сад с.Яковлевка» 19](#_Toc420327038)

[2.3. Информационные потоки 23](#_Toc420327039)

[2.4. Структура классов 26](#_Toc420327040)

[2.5. Варианты использования системы 28](#_Toc420327041)

[ГЛАВА 3. Описание практической части информационной системы МБДОУ «Детский сад с.Яковлевка» 30](#_Toc420327042)

[3.1. Выбор программного обеспечения 30](#_Toc420327043)

[3.1.1. Используемая платформа 30](#_Toc420327044)

[3.1.2. Используемая технология 33](#_Toc420327045)

[3.1.2.1. Физическая структура приложения 37](#_Toc420327046)

[3.1.3. Используемый язык программирования 39](#_Toc420327047)

[3.1.4. Средства проектирования клиентской части приложения 41](#_Toc420327048)

[3.1.5. Используемое средство разработки приложения 42](#_Toc420327049)

[3.1.6. Используемая СУБД 43](#_Toc420327050)

[3.1.6.1 Проектирование базы данных системы 45](#_Toc420327051)

[3.2. Описание веб-интерфейса приложения 51](#_Toc420327052)

[Заключение 52](#_Toc420327053)

[Список использованных источников 53](#_Toc420327054)

**РЕФЕРАТ**

Пояснительная записка 75 страниц, 3 части, 25 иллюстраций, 15 таблиц, 25 источников.

Ключевые слова: БАЗА ДАННЫХ, УПРАВЛЕНИЕ БАЗОЙ ДАННЫХ, МОДИФИКАЦИЯ ДАННЫХ, ИНФОРМАЦИОННАЯ СИСТЕМА.

Объект исследования – муниципальное бюджетное дошкольное образовательное учреждение «Детский сад с.Яковлевка Базарно-Карабулакского муниципального района Саратовской области».

Цель выпускной квалификационной работы – разработка специализированного веб-приложения для МБДОУ «Детский сад с.Яковлевка», которое позволит сэкономить массу времени при внесении данных, их обработке и получению результатов.

В работе приведен сравнительный анализ существующих аналогов информационных систем, предназначенных для управления процессами в дошкольных образовательных учреждениях.

В результате выполнения выпускной квалификационной работы было разработано управляемое веб-приложение «Информационная система для МБДОУ «Детский сад с.Яковлевка», которое позволит снизить трудоемкость бумажного документооборота и расширить список решаемых административных задач. В работе для построения приложения использовались платформа Microsoft .NET, технология ASP.NET MVC, язык программирования C#, средство разработки веб-приложений Microsoft Visual Studio Professional 2012, СУБД MS SQLServer 2008 Management Studio. Пояснительная записка оформлена в текстовом редакторе MS Word 2007.

Разработанная информационная система может использоваться сотрудниками дошкольного учреждения в служебных целях.

*Реферат.doc (MSWord 2007)*

**Введение**

Процесс информатизации играет значительную роль в развитии современного общества. В большинстве случаев деятельность организаций зависит от информативности и способности эффективно обрабатывать уже имеющуюся информацию.

Работа в детском саду, помимо обучения и воспитания, детей связана с накоплением большого объема разноплановой информации (о детях, родственниках, сотрудниках детского сада, различных мероприятиях, проводимых с детьми). Поэтому, очень много времени уходит на ручной труд, что является причиной снижения эффективности работы организации. Подобную проблему можно решить разработкой специализированной информационной системы для дошкольного учреждения. Использование информационных систем и технологий в деятельности дошкольных образовательных учреждений (ДОУ) позволяет облегчить трудоемкость выполнения каких-либо задач, связанных с управлением ДОУ.

В целом, процесс информатизации в дошкольных образовательных учреждениях развивается достаточно медленно.

Цель выпускной квалификационной работы заключается в разработке специализированной информационной системы с веб-интерфейсом для МБДОУ «Детский сад с.Яковлевка», которая позволит сэкономить массу времени при внесении данных, их обработке и получению результатов.

Для достижения поставленной цели необходимо спроектировать базу данных учреждения и снабдить приложение следующим функционалом:

* ввод, поиск, редактирование и удаление данных, содержащихся в базе данных с помощью веб-интерфейса;
* генерация отчетов;
* разграничение прав доступа;
* создание личной карточки ребенка;
* создание медицинской карточки ребенка.

Информатизация дошкольного учреждения позволит снизить трудоемкость бумажного документооборота, а также расширить список отчетов и решаемых административных задач.

В связи с тем, что в детских садах специального обучения информационным технологиям для сотрудников не производится, интерфейс приложения должен быть понятным и обеспечивать удобный доступ к основным функциям системы.

Объектом автоматизации является муниципальное бюджетное дошкольное образовательное учреждение Базарно-Карабулакского района Саратовской области «Детский сад с.Яковлевка».

Разработанная информационная система еще не использовалась в рамках рассматриваемого учреждения.

# **ГЛАВА 1. Описание предметной области и обзор существующих информационных систем для дошкольных образовательных учреждений**

### 1.1. Описание предметной области.

Данная работа посвящена разработке специализированной информационной системы с веб-интерфейсом, содержащая информацию о дошкольном образовательном учреждении. При проектировании информационной системы важнейшей задачей является получение информации об объекте управления на основе имеющихся данных. Рассмотрим организационную структуру МБДОУ «Детский сад с.Яковлевка» (рис. 1.1).

**Рис.1.1 Организационная структура МБДОУ «Детский сад с.Яковлевка».**

Организационная структура детского сада представлена в виде иерархической модели. Во главе стоит заведующий, который руководит всем дошкольным учреждением. Непосредственно ему подчиняются воспитатели, завхоз, музыкальный работник и медицинский работник. Третья ступень подчиняется завхозу. Рассмотрим основные функциональные особенности структуры объекта управления.

### 1.2. Функциональные особенности объекта управления.

В настоящее время одной из главных задач ДОУ является сбор необходимой информации о каждом ребенке для формирования необходимой отчетности, объективной оценки посещаемости детского сада и состояния здоровья воспитанников.

Данный процесс выполняет заведующий детским садом. Он решает следующие основные задачи:

* осуществление руководства всеми видами деятельности дошкольного учреждения;
* осуществление контроля над исполнением обязанностей каждого работника;
* получение всей необходимой информации о каждом ребенке и его родственниках;
* получение информации о сотрудниках детского сада;
* получение информации о мероприятиях, регулярно проводимых в детском саду;
* осуществление административно-хозяйственной деятельности учреждения;
* составление отчетности.

Работа по медицинскому обслуживанию осуществляется в специально оборудованном медицинском кабинете и обеспечивается медицинским персоналом. Медицинский работник детского сада выполняет следующие функции:

* получение объективной информации о каждом ребенке;
* осуществление медицинского наблюдения за физическим развитием и ростом ребенка;
* проведение регулярных антропометрических измерений у воспитанников дошкольного учреждения;
* оказание первой медицинской помощи при несчастных случаях;
* выявление заболеваний у ребенка и при необходимости его своевременная изоляция;
* проведение необходимых профилактических и медицинских мероприятий.

Сбор информации о группах, мероприятиях, ребенке и его родственниках также принадлежит воспитателям детского сада. В рассматриваемом учреждении их всего 6 человек.

### 1.3. Исследования автоматизации процессов.

Определим основные проблемы предметной области и выделим ту часть, которую можно частично или полностью информатизировать.

Выше были описаны процессы, выполняемые в дошкольном учреждении. Выявим ключевые проблемы, возникающие при обработке информации:

* сбор и анализ информации о ребенке и его родственниках, мероприятиях, а также медицинских мероприятиях производится сотрудниками детского сада ручным способом;
* личная и медицинская карточки ребенка хранятся в бумажном виде, что приводит к таким проблемам, как увеличение времени доступа к нужной информации (ручной поиск карточки в стопке документов), к неэффективному доступу к карточке ребенка (обычно хранятся в удаленных шкафах), нечитабельности информации (зависит от почерка сотрудника);
* отсутствие общего доступа к основной информации о детском учреждении;
* большие затраты времени для обработки уже имеющихся данных;
* сложность при составлении отчетов.

Для устранения рассмотренных проблем необходимо информатизировать следующие процессы:

* для решения главной проблемы предлагается разработать информационную систему с веб-интерфейсом, предоставляющему доступ к системе заведующей, медицинскому работнику и воспитателям дошкольного учреждения. Пользователям информационной системы предоставить доступ к информации, хранящейся в базе данных и возможность модифицирования данных;
* информационная система позволит создавать личную и медицинскую карточки ребенка в электронном виде, которые при необходимости можно будет распечатывать;
* для решения третьей проблемы будет разработан веб-сайт, который позволит получать доступ к основной информации о дошкольном учреждении через глобальную сеть Интернет;
* информатизация дошкольного учреждения позволит сэкономить массу времени при внесении данных, их обработке и получению результатов;
* информационная система позволит автоматически генерировать необходимые отчеты.

### 1.4. Описание существующих информационных систем.

При решении задач информатизации каких-либо учреждений или организаций всегда необходимо обращать внимание на уже разработанные системы. Рассмотрим несколько существующих информационных систем детских садов.

### 1.4.1. Информационная система публикации и обработки данных о детях в ГОУ детский сад №176 «Белочка».

Информационная система публикации и обработки данных о детях в ГОУ детский сад №176 «Белочка» используется для решения задач по двум основным направлениям:

1. образовательная часть - заключается в предоставлении образовательной программы по воспитанию ребенка и подготовке его в школу;
2. медицинская часть - заключается в обеспечении благоприятных условий для воспитания детей и слежении за их здоровьем.

В разработанной информационной системе имеется разграничение прав для пользователей, которые при входе в систему выполняют определенные функции.

Функции системы:

* ведение истории болезни детей;
* учет пройденных вакцинаций детьми;
* построение программы обучения детей;
* проведение у детей тестирования для определения способностей каждого из них;
* получение информации о заболеваемостях детей с целью принятия мер для дальнейшего предотвращения.

Информационная система обеспечивает высокую степень автоматизации процессов учета информации о детях дошкольного учреждения, повышает эффективность обработки отчетов по требованию, а также реализовывает дополнительные функции, такие как статистический анализ успеваемости ребенка и его заболеваемость в определенные сроки.

Технологии разработки: JEE, сервер базы данных - MySQL, сервер приложений - JBOSS Application Server.

### 1.3.2. Автоматизированная информационная система детского сада «Солнышко».

Разработанная информационная система «Солнышко» предполагает одно рабочее место и два типа пользователей - заведующий, имеющий полный доступ ко всей информации, хранящейся в базе данных, а также к модификации данных (добавлению, редактированию и удалению) и воспитатели, которые имеют возможность просмотра и печати информации. Разграничение прав доступа происходит в момент запуска базы данных.



**Рис. 1.2 Главная форма информационной системы детского сада «Солнышко».**

Функции системы:

* печать информации о сотрудниках;



**Рис. 1.3 Результаты поиска по сотрудникам.**

* печать списка группы: дети, воспитатели и младшие воспитатели;
* печать личной карточки ребенка, содержащую информацию непосредственно о ребенке;
* ввод, редактирование и удаление данных во всех таблицах;
* печать статистического распределения воспитанников детского сада по половому признаку;
* вызов формы редактирования данных о ребенке или сотрудниках из форм, отображающих эти данные.

Функция печати подразумевает наличие как экранных, так и печатных форм для отображения конечных результатов.

Таким образом, данная система содержит следующие формы:

* главную форму, в которой происходит выбор действия и запуск остальных форм, реализующих все функции;
* форма редактирования сотрудников;
* форма редактирования родственников ребенка, связанная с выбором детей;
* форма редактирования ребенка, связанная с выбором родственников ребенка;
* форма редактирования группы с возможностью выбора ребенка и сотрудников;
* форма печати статистики.

Технологии разработки: Microsoft Access.

### 1.3.3. Информационная система «Личная карточка» по учету кадров в дошкольном образовательном учреждении.

Информационная система «Личная карточка» предназначена для автоматизации личных данных сотрудников дошкольного образовательного учреждения. Необходимость создания информационной системы объясняется тем, что ведение учета личных данных о сотрудниках детского сада вручную является трудоемкой работой и длительным процессом. Разработанная система позволяет ускорить управление, повысить эффективность работы заведующего.

Автоматизированная информационная система «Личная карточка» отражает следующие личные данные работников детского учреждения:

- Фамилия, имя, отчество;

- Адрес проживания;

- Дата рождения;

- Гражданство;

- Начало работы;

- Стаж работы;

- Телефон;

- Табельный номер;

- ИНН;

- Номер страхового свидетельства;

- Номер трудового договора;

- Паспортные данные;

-Образование;

- Должность;

- Семейное положение.

Функции системы:

- вывод на экран личных данных сотрудников детского сада;

- добавление, удаление или редактирование данных о сотрудниках дошкольного учреждения;

- поиск информации о кадрах;

- печать личной карточки сотрудника детского сада.

Технологии разработки: серверная часть - база данных MySQL, клиентская часть - язык программирования - PHP.

### 1.3.4. Сравнительный анализ.

Для сравнительного анализа были выбраны три вышеописанные информационные системы, разработанные для дошкольных образовательных учреждений, которые имеют некоторые сходства с разрабатываемой в рамках выпускной квалификационной работы информационной системы для МБДОУ «Детский сад с.Яковлевка»: «Автоматизированная информационная система публикации и обработки данных о детях в ГОУ детский сад №176 «Белочка», «Автоматизированная информационная система «Солнышко», «Информационная система «Личная карточка» по учету кадров в дошкольном образовательном учреждении». Все эти системы схожи между собой и обладают своими преимуществами и недостатками.

АИС «Белочка» - достаточно полезная система, включающая автоматизацию процессов, связанных с направлениями по образовательной и медицинской части. К плюсам данной системы относится построение программы обучения детей и проведение тестирования для определения способности каждого ребенка. Однако здесь не предусмотрена обработка информации о ребенке, его родственниках, сотрудниках и группах. Она специализирована только для повышения эффективности выполнения отчетов по требованию и реализовывает такие функции, как статистический анализ успеваемости детей и их заболеваемость в определенные сроки.

Основное достоинство информационной системы «Солнышко» заключается в том, что она охватывает практически все необходимые объекты управления дошкольным образовательным учреждением и имеет разграничение прав пользователя. Однако работа по медицинской части в ней не предусмотрена.

К достоинствам ИС «Личный кабинет» относится то, что данная система направлена на повышение эффективности работы заведующего детским садом по учету кадров, что является немаловажной частью в управлении дошкольным учреждением. Недостаток системы: она разработана только для одного пользователя, включает узкую область применения и выполняет мало функций.

Сравнительная характеристика существующих информационных систем, разработанных для детских садов, приведена в таблице 1.

**Таблица 1**

Сравнительный анализ существующих ИС

| Название системы  Функции | АИС ДС «Белочка» | АИС "Солнышко" | АИС «Личная карточка» | ИС МБДОУ "Детский сад с.Яковлевка" |
| --- | --- | --- | --- | --- |
| 1 | 2 | 3 | 4 | 5 |
| Вывод на экран через web-интерфейс | Есть | Нет | Нет | Есть |
| Ввод, редактирование, удаление, поиск информации о детях, родителях, группах детского сада | Нет | Есть | Есть | Есть |
| Личная карточка ребенка | Нет | Есть | Нет | Есть |
| Построение программы обучения детей | Есть | Нет | Нет | Нет |
| Проведение у детей тестирования для определения способностей каждого из них | Есть | Нет | Нет | Нет |
| Получение информации о заболеваемости детей | Есть | Нет | Нет | Есть |
| Осуществление эффективной организационно-медицинской работы | Есть | Нет | Нет | Есть |
| Добавление, удаление или редактирование данных о сотрудниках дошкольного учреждения | Нет | Есть | Есть | Есть |
| Медицинская карточка ребенка | Есть | Нет | Нет | Есть |
| Ввод, редактирование, удаление, поиск информации по мероприятиям, проводимых в детском саду | Нет | Нет | Нет | Есть |
| Личная карточка сотрудника | Нет | Есть | Есть | Есть |
| Отчетные возможности | Есть | Нет | Нет | Есть |

Очевидно, что разработанные ранее системы не способны полностью обеспечить процесс автоматизации основных задач, с которыми ежедневно сталкиваются сотрудники ДОУ. Необходимо создание приложения с разграничением прав доступа, включающего не только обработку статистической информации о ребенке, его родственниках и сотрудниках учреждения, но и позволяющего автоматизировать процессы образовательной и медицинской части учреждения. В разработанной информационной системе МБДОУ «Детский сад с.Яковлевка» предусмотрены все недостатки, которыми обладают вышеперечисленные информационные системы:

* статистическая обработка информации о детях, их родственниках, сотрудниках детского сада;
* осуществление эффективной организационно-медицинской работы;
* получение информации о заболеваемостях ребенка;

# **ГЛАВА 2. ОПИСАНИЕ РАЗРАБОТАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ МБДОУ «ДЕТСКИЙ САД С.ЯКОВЛЕВКА»**

## 2.1. Функциональные возможности информационной системы МБДОУ «Детский сад с.Яковлевка».

В состав информационной системы МБДОУ «Детский сад с.Яковлевка» входят следующие подсистемы:

- подсистема хранения данных;

- подсистема ввода-вывода;

- подсистема обработки данных.

В информационной системе имеется разграничение прав пользователей, которые при входе в систему могут выполнять определенные функции.

Рассмотрим, какими функциональными возможностями обладает каждая из подсистем отдельно.

**1. Функции подсистемы хранения данных**

Базы данных позволяют хранить большой объем информации некоторым упорядоченным образом. Подсистема хранения данных предназначена для:

- хранения большого объема данных системы;

- добавления в БД нового ребенка, родственника, группы, сотрудника, должности, мероприятия, а также медицинского мероприятия посредством веб-интерфейса;

- удаление и редактирование имеющейся в БД информации;

- поиск необходимой информации по запросу пользователя системы.

**2. Функции подсистемы ввода-вывода**

Подсистема ввода-вывода предназначена для предоставления пользователям системы интерфейса для ввода и отображения информации:

- поиск информации по запросу пользователя;

- создание личной и медицинской карточки ребенка (в зависимости от прав пользователя системы);

- создание личной карточки сотрудника (в зависимости от прав пользователя системы);

- запрос на удаление и редактирование данных;

- ввод необходимой информации.

**3. Функции подсистемы обработки данных**

Подсистема обработки данных предназначена для реализации функций с полученными данными:

- генерация отчетности;

- опубликование информации для общего доступа.

Информационная система МБДОУ «Детский сад с.Яковлевка» должна обеспечивать публичный доступ гражданам Российской Федерации к открытой части информации через глобальную сеть Интернет.

## 2.2. Входные и выходные данные в информационной системе МБДОУ «Детский сад с.Яковлевка».

Определим входные и выходные данные разработанной информационной системы.

**Входные данные.**

*1. Аутентификация пользователя*

Данные, вводимые пользователем при входе в систему:

- имя пользователя;

- пароль.

*2. Создание личной карточки ребенка*

Для создания личной карточки ребенка пользователь вводит следующие данные:

- Фамилия;

- Имя;

- Отчество;

- Пол;

- Дата рождения;

- Дата принятия;

- Дата выпуска;

- Мать (выбирает из имеющегося списка);

- Отец (выбирает из имеющегося списка);

- Опекун (выбирает из имеющегося списка);

- Малообеспеченная семья;

- Дополнительная информация.

*2. Создание медицинской карточки ребенка*

Для создания медицинской карточки ребенка пользователь вводит следующие данные:

- Фамилия;

- Имя;

- Отчество;

- Пол;

- Дата рождения;

- Рост;

- Вес;

- Заболевания;

- Противопоказания.

*3. Добавление группы*

При добавлении новой группы пользователь вводит:

- Номер группы;

- Название;

- Сотрудников;

- Описание.

*4. Добавление родственников ребенка*

При добавлении родственников ребенка пользователь вводит:

- Фамилия;

- Имя;

- Отчество;

- Пол;

- Дата рождения;

- Название типа отношения;

- Адрес проживания;

- Место работы;

- Должность;

- Телефон

*5. Добавление должности*

При добавлении новой должности пользователь вводит:

- Название должности;

- Зарплата;

- Дополнительная информация.

*6. Создание личной карточки сотрудника*

Для создания личной карточки сотрудника пользователь вводит следующие данные:

- Фамилия;

- Имя;

- Отчество;

- Пол;

- Дата рождения;

- Адрес проживания;

- Телефон;

- Образование;

- Учебное заведение;

- Должность (выбирает из имеющегося списка);

- Группа (если принадлежит к какой-нибудь группе детского сада);

- Зарплата;

- Стаж;

- Надбавка;

- Дополнительная информация.

*7. Добавление мероприятия*

При добавлении нового мероприятия пользователь вводит следующие данные:

- Название;

- Дата проведения;

- Описание;

- Место проведения;

- Группа (принимающие участие в данном мероприятии);

- Сотрудники (ответственные за проведение мероприятия);

*8. Добавление медицинского мероприятия*

При добавлении нового медицинского мероприятия пользователь вводит следующие данные:

- Название;

- Тип;

- Описание;

- Дата начала;

- Дата конца;

- Сотрудники (ответственные за проведение медицинского мероприятия);

- Место проведения.

**Выходные данные:**

*1. Авторизация пользователя*

При авторизации пользователя происходит вход в главную страницу системы, где располагается основное меню. Функциональные возможности зависят от прав вошедшего в систему пользователя.

*2. Личная карточка ребенка*

В результате создается личная карточка с общей информацией о ребенке;

*3. Медицинская карточка ребенка*

В результате создается медицинская карточка с медицинской информацией о ребенке;

*4. Личная карточка сотрудника*

В результате создается личная карточка с общей информацией о сотруднике дошкольного учреждения;

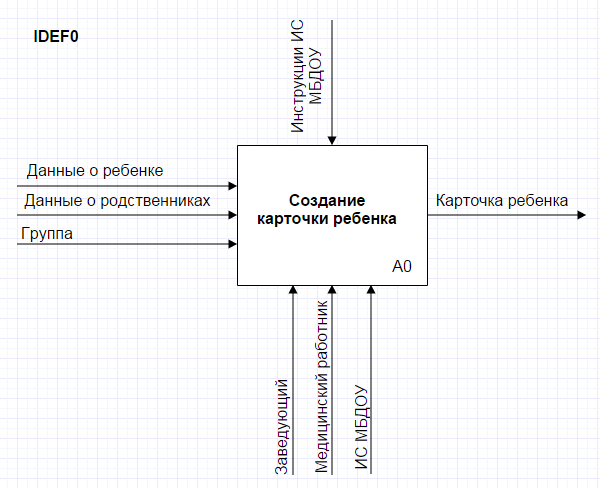
*5. Отчеты*

## 2.3. Информационные потоки.

На начальных этапах создания информационной системы необходимо понять, как работает организация, которую необходимо информатизировать. Наиболее удобным языком моделирования бизнес-процессов является IDEF0, где система представляется как совокупность взаимодействующих работ и функций. Для детального представления работы информационной системы «Детский сад с.Яковлевка» отразим на контекстной диаграмме функциональный блок, который соответствует одной из основных функций данной системы. При построении контекстной диаграммы IDEF0 выбирается точка зрения пользователя, ответственного за выполнение определенного процесса.

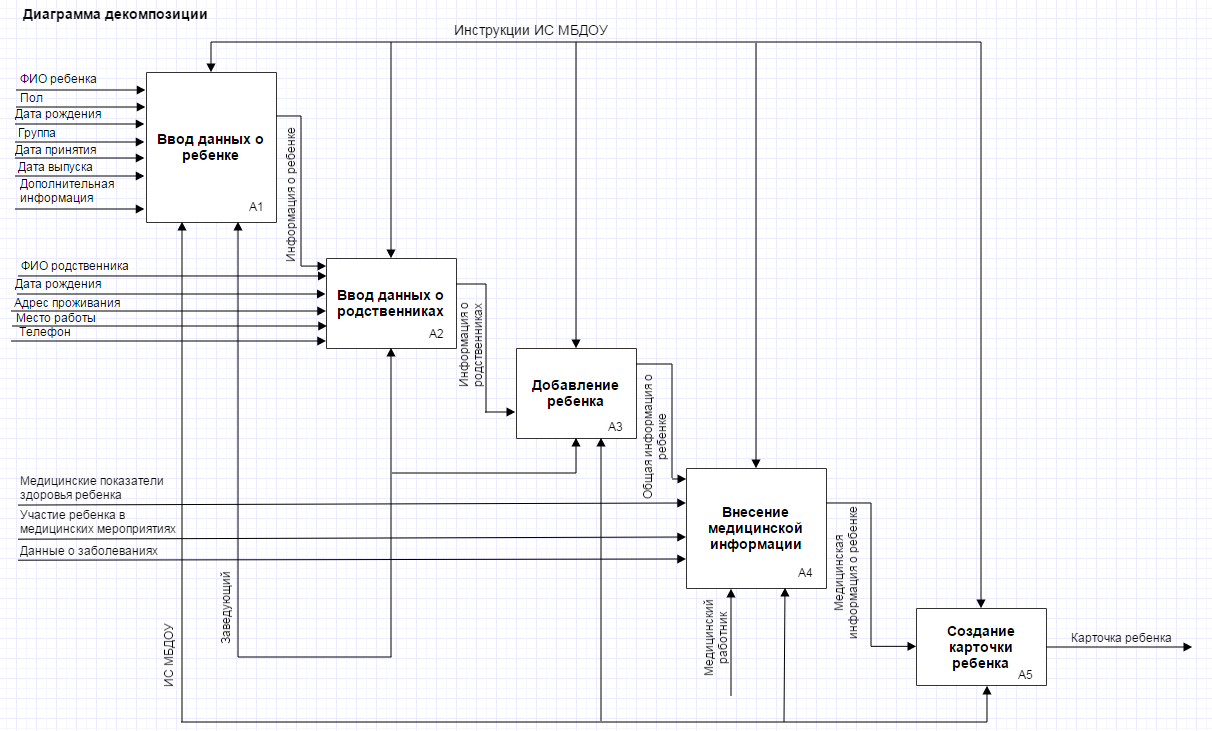
**Точка зрения:** пользователь

**Процесс:** добавление данных о ребенке в базу данных (личная информация)



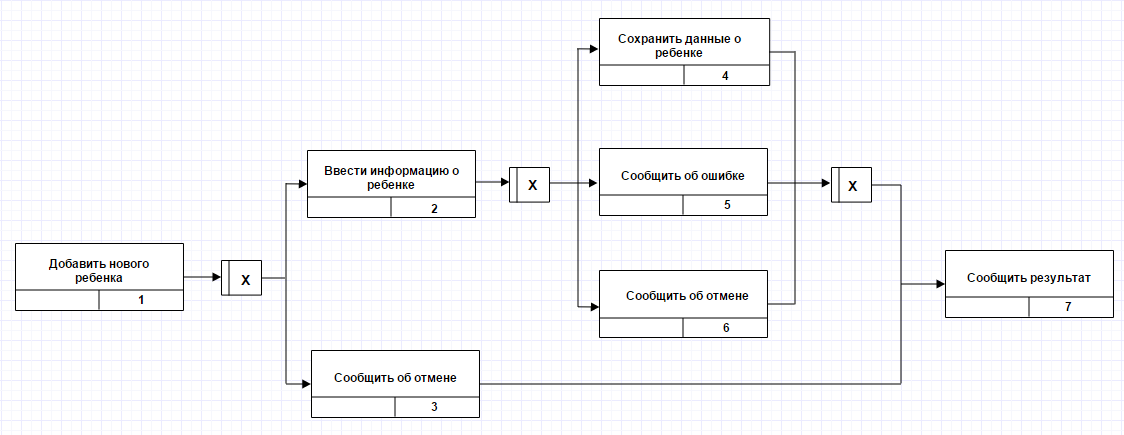
**Рис. 2.1. Контекстная диаграмма IDEF0 информационной системы.**

Функцию, приведенную на контекстной диаграмме можно разбить на более мелкие функции и описать их взаимодействие на диаграмме декомпозиции (рис 2.2).



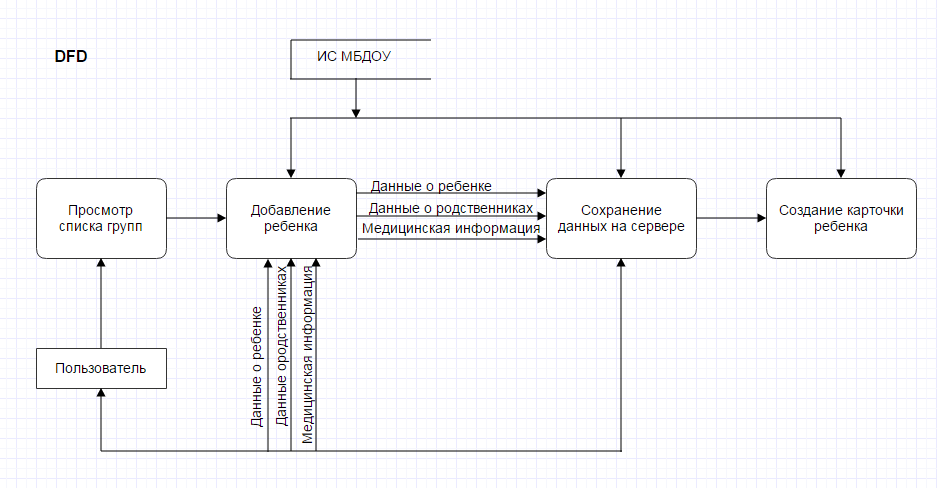
**Рис. 2.2. Декомпозиция контекстной диаграммы информационной системы.**

Для описания достаточно крупных задач применяются контекстная диаграмма IDEF0. Однако с их помощью трудно понять, как выполняются задачи: синхронно или асинхронно, для начала выполнения новой задачи обязательно ли завершение всех остальных и т.д. Поэтому, для подробного описания работы той или иной функции используется диаграмма стандарта IDEF3 (рис. 2.3).



**Рис. 2.3. IDEF3-диаграмма работы информационной системы.**

Контекстная диаграмма IDEF0 и диаграмма стандарта IDEF3 в основном применяются для описания функционирования системы и обходят стороной существующие в ней информационные потоки. Для описания потоков данных используются диаграммы потоков данных (Data Flow Diagrams – DFD), которые представляют собой иерархию функциональных процессов, связанных с информационными потоками.



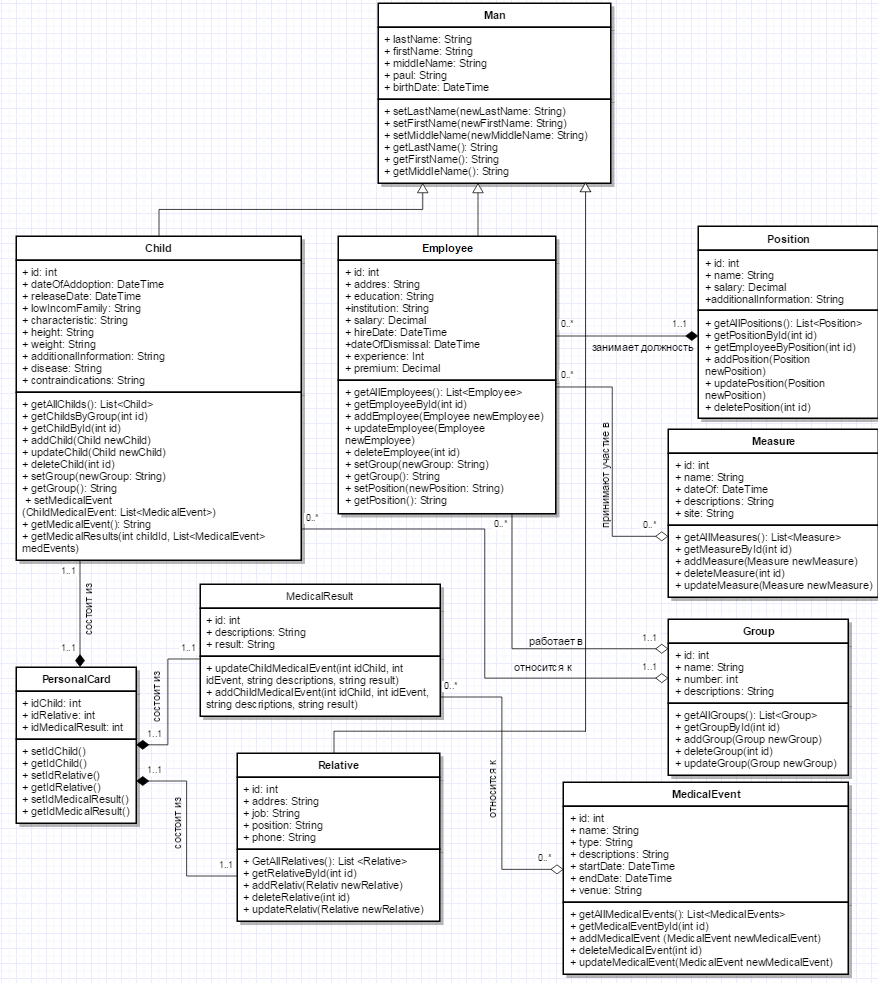
**Рис. 2.4. DFD-диаграмма потоков данных.**

Диаграмма потоков данных описывает процесс переработки информации от поступления в систему до выдачи пользователю. Информация поступает в систему от внешних сущностей и обрабатывается процессами, которые, в свою очередь, могут порождать новые информационные потоки. DFD представляет собой моделируемую систему как сеть связанных работ.

При разработке информационной системы необходимо провести объектно-ориентированный анализ, целью которого является создание объектной модели предметной области. UML (Unified Modeling Language) – унифицированный язык моделирования, являющийся языком широкого профиля, использующий графические обозначения для создания абстрактной модели системы, называемой UML-моделью.

## 2.4. Структура классов.

Центральное место в объектно-ориентированном программировании занимает разработка логической модели системы в виде диаграммы классов (class diagram), которая входит в состав языка моделирования UML. Она применяется для описания структуры системы, показывая ее классы, их атрибуты, операторы, а также взаимосвязи между классами. На рис. 2.5 представлена диаграмма классов разработанной информационной системы для МБДОУ «Детский сад с.Яковлевка».



**Рис. 2.5. Диаграмма классов.**

Диаграмма классов UML позволяет обозначать отношения между классами и их экземплярами.

## 2.5. Варианты использования системы.

Для наглядного представления функциональных возможностей групп пользователей системы часто используются диаграммы вариантов использования (use case diagram). На рис. 2.2 представлена диаграмма вариантов использования информационной системы для МБДОУ «Детский сад с.Яковлевка».



**Рис. 2.6. Диаграмма вариантов использования.**

В результате построения диаграммы вариантов использования было выявлено, что в системе существует четыре основные роли:

* гость;
* воспитатель;
* медицинский работник;
* заведующий;

Разграничение прав доступа пользователей системы реализовано в соответствии с разделением пользователей на группы. У каждой группы имеется свой функциональный набор. Рассмотрим варианты использования для каждой из этих ролей. Гость имеет право на просмотр общих сведений о детском саде. Пользователи, принадлежащие роли «Воспитатели» имеют следующие функциональные возможности:

* вход в систему по своим учетным данным (ввод логина и пароля);
* поиск информации о группах и родственниках детей;
* ведение данных о мероприятиях (расширяется возможностью добавления, удаления и редактирования)
* просмотр личной и медицинской карточки ребенка (расширяется возможностью их распечатывания).

Роли «Медицинский работник» принадлежит следующий набор функций:

* вход в систему по своим учетным данным (ввод логина и пароля);
* поиск информации о группах и родственниках детей;
* ведение данных о медицинских мероприятиях (расширяется возможностью добавления, удаления и редактирования);
* просмотр личной и медицинской карточки ребенка (расширяется возможностью их распечатывания);
* Создание медицинской карточки ребенка.

Пользователь, относящийся роли «Заведующий» наследует все варианты использования ролей «Гость», «Воспитатели» и «Медицинский работник», а также обладает дополнительными функциональными возможностями:

* создание личной карточки (расширяется возможностью создания личной и медицинской карточки ребенка, личной карточки сотрудника);
* поиск, добавление, удаление и редактирование данных о должностях, группах и родственниках детей;
* изменение прав доступа пользователей системы (расширяется возможностью добавления и удаления пользователя).

# **ГЛАВА 3. Описание практической части информационной системы МБДОУ «Детский сад с.Яковлевка»**

### 3.1. Выбор программного обеспечения.

Для реализации разработанного проекта в рамках выпускной квалификационной работы были выбраны платформа Microsoft .NET Framework с применением технологии ASP.NET MVC4, связанный с ней язык программирования C#, интегрированная среда разработки программного обеспечения Microsoft Visual Studio Professional 2012, СУБД Microsoft SQL Server 2008.

### 3.1.1. Используемая платформа.

Microsoft .NET Framework – это программная платформа, выпущенная в 2002 году компанией Microsoft.

NET Framework является одним из передовых технологий разработки программных продуктов на базе семейства операционных систем Windows, а также других операционных систем, таких как Mac OS X и различные дистрибутивы Unix и Linux, которая позволяет упростить объектно-ориентированное программирование и создавать различного рода приложения.

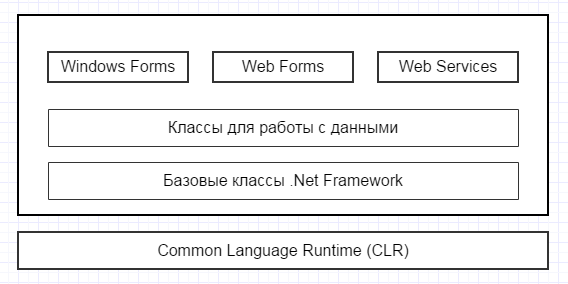
Выбор данной платформы в качестве платформы для разрабатываемого приложения обусловлена тем, что она пользуется популярностью и обладает большими преимуществами при создании программ:

* полное и абсолютное межъязыковое взаимодействие. Данная платформа поддерживает межъязыковое наследование, межъязыковую обработку исключений и межъязыковую отладку;
* общая среда выполнения для любых приложений .NET;
* наличие обширной библиотеки классов, позволяющей существенно облегчить решение задач, возникающих при создании автономных программ и веб-приложений. Эта библиотека насчитывает десятки тысяч классов, которые позволят использовать в своих разработках готовые модули;
* интеграция языков программирования. Платформа .NET Framework поддерживает интеграцию различных языков программирования, т.е. один язык может использовать типы, созданные на других языках;

.NET Framework состоит из двух основных компонентов:

* Common Language Runtime (CLR) – общеязыковая исполняющая среда, основной задачей которого является обнаружение, загрузка и управление типами .NET. Она обеспечивает исполнение программного кода и является основой платформы Microsoft .NET. Среда CLR активизирует исполняемый код, выполняет для него проверку безопасности, располагает этот код в памяти и выполняет его. Важной частью работы среды CLR является управление свободной памятью, автоматически обеспечивая использование освобождающейся при работе программ памяти (сборку мусора).
* Framework Class Library (FCL) представляет собой библиотеку классов, интерфейсов и типов значений, обеспечивающий доступ к функциональным возможностям системы. Она содержит в себе множество компонентов для работы с базами данных, вводом/выводом, сетью, пользовательским интерфейсом и т.д., позволяющий разработчику не заниматься низкоуровневым программированием, а использовать уже готовые классы.

На рис. 3.1 представлена архитектура платформы Microsoft .NET Framework.



**Рис. 3.1. Архитектура MS .NET Framework.**

На верхнем уровне выделены общеязыковая исполняющая среда (CLR) и библиотеки классов MS .NET Framework:

* набор базовых классов, которые обеспечивают работу со строками, ввод-вывод данных, многопоточность и т.д.
* набор классов для работы с данными, предоставляющих возможность использования SQL-запросов, ADO.Net и обработки XML данных;
* набор классов Windows Forms позволяют создавать обычные Windows-приложения, использующие стандартные элементы управления Windows;
* набор классов Web Forms обеспечивают возможность разработки веб-приложений, использующие стандартный графический интерфейс пользователя;
* набор классов Web Services поддерживают создание распределенных компонентов-сервисов, доступ к которым организован через сеть Интернет.

### 3.1.2. Используемая технология.

ASP.NET (Active Server Pages for .NET) - технология создания веб-приложений и веб-сервисов от компании Microsoft, являющейся составной частью платформы Microsoft .NET и развитием более старой технологии Microsoft ASP.

Хотя технология ASP.NET берет свое название от старой технологии Microsoft ASP и имеет с ней некоторые сходства, внутренне устройство ASP.NET существенно отличается, так как она основана на платформе .NET и использует все новые возможности, предоставляемые данной платформой. Основываясь на Common Language Runtime(CLR), Microsoft полностью перестроила ASP.NET. Разработчики могут писать код для ASP.NET, используя практически все языки программирования, которые входят в состав .NET Framework (C#, Visual Basic.NET, JScript .NET).

По сравнению с другими платформами для создания веб-приложений ASP.NET обладает целым рядом преимуществ. Ниже перечислены основные преимущества, которыми обладает данная технология:

* компиляция исполняемых частей Web-приложения, благодаря которой они работают быстрее, чем интерпретируемые сценарии;
* доступ к инфраструктуре .NET Framework, дополняющей возможности API Windows;
* использование известного языка программирования Visual Basic и нового языка Visual C#;
* автоматическое управление состоянием элементов управления Web-страниц (называемых серверными элементами управления - server controls). Благодаря этой особенности их поведение мало отличается от обычных элементов управления Windows;
* возможность создания новых, нестандартных серверных элементов управления на основе существующих;
* обеспечение безопасности Web-приложений с помощью встроенных защитных механизмов сервера Windows и других методов аутентификации и авторизации;
* интеграция с ADO.NET, позволяющая получать доступ к БД и применять инструменты для конструирования БД из Visual Studio .NET;
* полная поддержка языка Eextensible Markup Language (XML), каскадных таблиц стилей (CSS) и других новых стандартов Web;
* встроенные механизмы кэширования часто запрашиваемых Web -страниц на сервере, локализации содержимого (приспособления его для использования различных языков и региональных стандартов) и определения возможностей браузера.

В рамках платформы Microsoft .NET представлены две технологии создания веб-приложений: ASP.NET WebForms и ASP.NET MVC. Таким образом, одной из главных задач разработчика при создании приложений на платформе .NET является выбор между двумя этими технологиями.

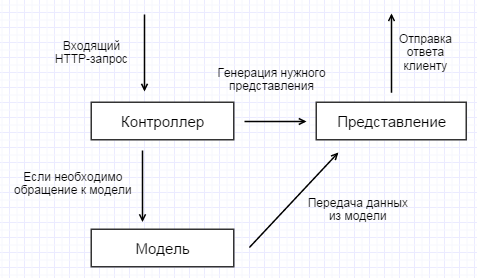
Более ранней технологией является ASP.NET WebForms, которая позволяет приблизить процесс разработки веб-приложений к процессу разработки настольных приложений. Данная технология имеет набор элементов управления, в которые входят как стандартные элементы, такие как кнопка, текстовое поле, список, так и специфические элементы управления (календарь, управление валидацией и т.д.). Данная технология позволяет размещать элементы управления на форме декларативно, при этом освобождая разработчика от необходимости писать большое количество HTML кода: элементы инкапсулируют HTML код, которыми они будут заменены в конечном итоге в HTML представлении. Элементы управления могут быть настроены при помощи свойств, которые доступны разработчику для изменения. Элементы управления для работы с данными позволяют быстро и удобно настроить элементы управления представлением для отображения, изменения и сохранения данных из БД практически без написания кода вручную.

ASP.NET MVC – это платформа, предназначенная для разработки сайтов и веб-приложений с использованием паттерна или шаблона MVC (Model - View - Controller). Данная платформа впервые появилась в 2009 году и в настоящее время обрела большую популярность по всему миру благодаря своей гибкости и адаптивности.

Шаблон Model - View - Controller (модель - представление - контроллер) подразумевает разделение приложения на три основных компонента, каждая из которых отвечает за свои задачи. Рассмотрим каждый компонент в отдельности:

* контроллер (controller) представляет собой класс, с которого и начинается работа приложения. Он является связующим звеном между моделью и представлением, обрабатывает данные, полученные от пользователя, и информирует модель и представление о необходимости соответствующей реакции;
* модель (molel) представляет собой набор классов и отвечает за бизнес-логику приложения. Он включает в себя классы-сущности данных, методы для получения, изменения и сохранения данных.
* представление (view) - визуальная часть приложения, отвечающая за отображение информации (пользовательский интерфейс).

На рис. 3.2 представлена общая схема взаимодействия вышеописанных компонентов.



**Рис. 3.2. Общая схема взаимодействия компонентов MVC.**

К написанию клиент-серверных приложений ASP.NET MVC предлагает классический подход «запрос-ответ», при этом обеспечивая прозрачный механизм обработки запросов. При запросе страницы сначала вызывается метод контроллера, который обрабатывает входные данные и отображает нужное представление. В отличие от WebForms, в котором код логики и представления смешаны между собой, в ASP.NET MVC идет четкое разделение кода логики от кода представления, вследствие чего приложение гораздо проще поддается модульному тестированию.

К основным преимуществам ASP.NET WebForms относятся:

* возможность быстрой разработки приложений;
* возможность декларативного создания страниц;
* простота разработки приложений, работающих с большими объемами данных;
* огромное количество библиотек;

К недостаткам ASP.NET WebForms относятся:

* логика представления тесно связана с кодом и, таким образом, сложно отделима;
* возникает сложность при модульном тестировании;
* отсутствует контроль над генерируемой HTML разметкой;
* большой размер страницы за счет использования сохранения состояния.

К основным преимуществам ASP.NET MVC относятся:

* четкое разделение между кодом логики и представления;
* полный контроль над генерируемой HTML разметкой;
* широкие возможности для модульного тестирования;
* небольшой размер страницы за счет отсутствия сохранения состояния.

К недостаткам ASP.NET MVC относятся:

* требуется больше времени при разработке приложения;
* количество библиотек в настоящее время невелико.

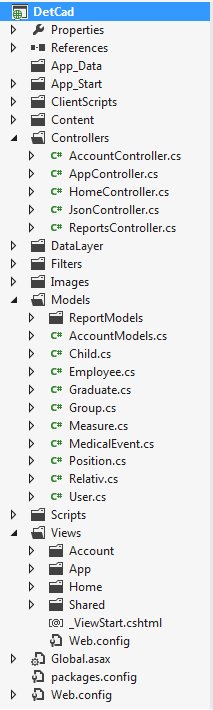
Таким образом, в результате сравнения двух технологий было выявлено, что каждая из них имеет свои достоинства и недостатки. Однако, технология ASP.NET MVC при разработке приложений обладает большим набором преимуществ по отношению к ASP.NET WebForms, вследствие чего и была выбрана для реализации проекта, разработанного в рамках выпускной квалификационной работы.

#### 3.1.2.1. Физическая структура приложения.

Основа программного обеспечения реализована в проекте DetCad, физическая структура которого представлена на рисунке 3.3.

Приложение разрабатывалось по принципам архитектуры MVC (Model-View-Controller), поэтому ответственность за различное функционирование здесь четко разделена:

* В папке Model содержатся классы, которые представляют модели данных (объектов). Например, класс Child описывает объект «ребенок», Employee – «сотрудник» и т.д.
* В папке View находятся все пользовательские представления.
* В папке Controller находятся контроллеры, которые содержат логику взаимодействия с пользователем и с объектами модели.



**Рис. 3.3. Физическая структура приложения DetCad.**

### 3.1.3. Используемый язык программирования.

C# - один из самых распространенных объектно-ориентированных языков программирования, являющийся специально созданным для работы в .NET. Таким образом, главная особенность языка C# заключается в том, что она ориентирована на платформу Microsoft .NET - создатели C# ставили своей целью предоставление разработчикам естественных средств доступа ко всем возможностям платформы .NET. Он был разработан группой инженеров под руководством Андерса Хейсберга в 1998 - 2001 годах.

Синтаксис языка программирования C# близок к С++ и Java. Язык поддерживает полиморфизм, указатели на функции-члены классов, события, атрибуты, свойства, имеет строгую типизацию. Переняв многое от своих предшественников - языков Java, C++, Delphi и опираясь на практику их использования, C# исключает при разработке программных систем некоторые проблематические модели. Например, в отличие от C++ не поддерживает множественное наследование классов.

Основные достоинства языка C#:

* является полностью объектно-ориентированным языком, в котором даже типы, встроенные в язык, представляются в виде классов;
* учитывает все возможности платформы Microsoft .NET Framework - как CLR, так и FCL;
* является наследником популярных языков программирования C/C++, сохраняя лучшие их черты. Знакомые операторы и общий синтаксис облегчают переход программистов от C++ к C#;
* благодаря мощной библиотеке каркаса Framework .Net поддерживает удобство построения разных типов приложений на C#, позволяя легко строить Web-службы и другие виды компонентов, достаточно просто сохранять и получать информацию из баз данных и других хранилищ данных;
* сочетает построение надежного и эффективного кода, что является немаловажной частью, способствующим успеху языка C#.

Еще одна особенность языка C# и фреймворка .NET - автоматическая сборка мусора. В большинстве случаев не возникает необходимость заботиться об освобождении памяти, в отличие от языка C++. Общеязыковая исполняющая среда CLR сама вызовет сборщика мусора и очистит память.

### 3.1.4. Средства проектирования клиентской части приложения.

Для написания клиентской части веб-приложения были выбраны язык гипертекстовой разметки HTML, каскадные таблицы стилей CSS, а также AngularJS, представляющий собой JavaScript-фреймворк, использующий шаблон MVC.

HTML (от англ. Hyper Text Language – «язык разметки гипертекста») – основной язык, с помощью которого создаются веб-страницы. Страница, которую пользователь видит в своем браузере, может состоять из различных файлов – изображений, сценариев JavaScript, анимационных роликов и т.д. – но практически всегда основой страницы является документ HTML. С помощью него можно создавать довольно простые, но красиво оформленные документы. Основным преимуществом использования HTML является то, что документ, созданный с помощью него, может быть просмотрен на Web-браузерах различных типов и на различных платформах.

Для описания внешнего вида любых HTML документов используется CSS (от англ. Cascading Style Sheets – «Таблицы Каскадных Стилей»). С его помощью разработчик имеет полный контроль над стилем и расположением каждого элемента веб-страницы, что гораздо проще и функциональнее использования обычных HTML-тегов.

При разработке приложения использовался AngularJS –JavaScript-фреймворк, разработанный специалистами Google. Он является основой, которая связывает HTML-код с JavaScript моделей и предоставляет разработчикам каркас будущего приложения, а также решение задач, встречающихся в большинстве проектов.

AngularJS включает в себя большой набор директив, которые позволяют связывать HTML-элементы моделей. Они представляют собой атрибуты, начинающихся с префикса ng-. Их можно добавлять к любому элементу. Одним из самых важных атрибутов при использовании Angular является ng-app, который нужно включить во все страница разрабатываемого сайта.

Важнейшим элементом AngularJS является область видимости (scope), которая ведет себя идентично области видимости в классических языках программирования. Область видимости – это обычный объект JavaScript. Работа с ней осуществляется простым добавлением или изменением свойств объектов. Он позволяет следить за изменениями собственных свойств, а также свойств родительских и дочерних областей видимости. Для работы с ними AngularJS предоставляет огромное количество функций.

Еще одной отличительной чертой данного фреймворка является двустороннее связывание, которое позволяет динамически изменять данные в одном месте интерфейса при изменении данных модели в другом. Таким образом, AngularJS синхронизирует модель и представление.

### 3.1.5. Используемое средство разработки приложения.

Microsoft Visual Studio - это универсальный продукт, разработанный компанией Microsoft, который позволяет создавать приложения, работающие на платформе .NET. Особенность данной платформы заключается в том, что она содержит широкий набор сервисов, доступных в различных языках программирования.

Возможности Microsoft Visual Studio:

* имеется возможность создавать приложения в Windows Forms и Web Forms путем простого перетаскивания мышью элементов пользовательского интерфейса;
* автоматическое выполнение всех шагов, необходимых для компиляции исходного кода;
* текстовый редактор интеллектуально может обнаруживать ошибки и делать подсказки в процессе ввода, какой код необходимо вставить;
* позволяет создавать многофункциональные веб-приложения, посредством использования данных баз ASP.NET;
* имеется большой набор мощных инструментов, с помощью которых можно просматривать отдельные элементы проекта или осуществлять в них поиск;
* поддерживает единую интегрированную среду разработки (IDE), независимо от языка программирования или типа проекта;
* упрощает разработку мощных веб-приложений, увеличивая эффективность работы разработчиков, предоставляя знакомую и интуитивную среду разработки.

Для разработки информационной системы МБДОУ «Детский сад с.Яковлевка» была выбрана среда Microsoft Visual Studio 2012, которая обладает высокой скоростью загрузки рабочей среды и позволяет управлять жизненным циклом приложения от этапа его разработки до стадии эксплуатации.

### 3.1.6. Используемая СУБД.

Microsoft SQL Server - это система управления базами данных (СУБД), разработанная корпорацией Microsoft, которая обеспечивает сетевой многопользовательский доступ. Он является одним из наиболее мощных СУБД архитектуры клиент-сервер и используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия. Основным используемым языком запросов является Transact-SQL, который был создан совместно Microsoft и Sybase.

Язык SQL (Structured Query Language) был разработан в 1974 году фирмой IBM и в 1987 году стал международным стандартным языком баз данных.

Основные характеристики языка SQL:

* мощный компактный язык манипулирования реляционными данными с небольшим набором команд;
* предложения языка ориентированы не на процедуру этой обработки, а на конечный результат;
* язык SQL может использоваться как:

- интерактивный (для выполнения запросов);

- встроенный (для построения прикладных программ);

* ориентирован на работу с таблицами, не имеет достаточных средств для создания сложных прикладных программ.

SQL содержит 4 типа предложений (команд):

1. предложения определения данных;
2. запросы на выборку данных (инструкция SELECT);
3. команды модификации данных (INSERT, DELETE, UPDATE);
4. предложения управления данными (управление транзакциями и другие).

Microsoft SQL Server 2008 R2 - платформа, предназначенная для управления и работы с информацией, а также безопасного доступа к данным. Особенностью данного продукта является наличие широкого набора инструментов для подготовки отчетной документации и анализа бизнес-процессов.

Ниже перечислены достоинства Microsoft SQL Server 2008 R2:

* обеспечивает высокий уровень надежности и производительности систем управления базами данных и оборудования локальной сети;
* сокращает временные затраты на создание аналитической документации из-за высокой степени быстродействия, а также наличия необходимых инструментов при выполнении расчетов;
* простота использования функций программы. Позволяет работать с SQL Server разработчикам и администраторам любого уровня подготовки;
* Microsoft SQL Server 2008 R2 содержит в составе компонент Report Services, который обеспечивает генерирование комплексных систем корпоративной отчетности;
* осуществляет из огромного количества источников одновременный учет данных, обрабатывая при этом более 10000 записей в секунду;
* запросы пользователей и обслуживание базы данных выполняются с большой скоростью, вследствие чего повышается результативность и качество функционирования информационной системы организации;
* поддерживает 64-разрядные операционные системы, вследствие чего повышается скорость работы баз данных и приложений;
* дает возможность легко находить и повторно использовать необходимые сведения, обеспечивая при этом их целостность и доступность.

#### 3.1.6.1 Проектирование базы данных системы.

Одним из основных и важных этапов в проектировании информационной системы является проектирование базы данных.

База данных информационной системы МБДОУ «Детский сад с.Яковлевка» состоит из 13 таблиц и 60 хранимых процедур. Ниже приведено описание таблиц базы данных.

**Таблица 3.1**

Описание таблицы Child – таблица детей

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_child | Целочисленный | Идентификатор |
| ID\_group | Целочисленный | Код группы, к которому относится ребенок |
| LastName | Текстовый | Фамилия ребенка |
| FirstName | Текстовый | Имя ребенка |
| MiddleName | Текстовый | Отчество ребенка |
| Paul | Текстовый | Пол ребенка |
| BirthDate | Дата | Дата рождения ребенка |
| DateOfAdoption | Дата | Дата принятия ребенка |
| ReleaseDate | Дата | Дата выпуска ребенка |
| Characteristic | Текстовый | Характеристика ребенка |
| Height | Текстовый | Рост ребенка |
| Weights | Текстовый | Вес ребенка |
| Disease | Текстовый | Заболевания |
| Сontraindications | Текстовый | Противопоказания |
| LowIncomFamily | Текстовый | Малообеспеченность семьи |
| AdditionalInformation | Текстовый | Дополнительная информация |

**Таблица 3.2**

Описание таблицы Child\_MedicalEvents – таблица участия ребенка в медицинских мероприятиях

|  |  |  |
| --- | --- | --- |
| Наименование поля | Тип данных | Описание |
| ID\_child | Целочисленный | Идентификатор |
| ID\_medicalEvent | Целочисленный | Идентификатор |
| Descriptions | Текстовый | Описание проведенного медицинского мероприятия |
| Result | Текстовый | Результаты ребенка по проведенному медицинскому мероприятию |

**Таблица 3.3**

Описание таблицы Employee – таблица сотрудников детского сада

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_employee | Целочисленный | Идентификатор |
| ID\_group | Целочисленный | Код группы, к которому относится сотрудник |
| ID\_position | Целочисленный | Код должности сотрудника |
| LastName | Текстовый | Фамилия сотрудника |
| FirstName | Текстовый | Имя сотрудника |
| MiddleName | Текстовый | Отчество сотрудника |
| Paul | Текстовый | Пол сотрудника |
| BirthDate | Дата | Дата рождения сотрудника |
| Address | Текстовый | Адрес проживания сотрудника |
| Phone | Текстовый | Телефон |
| Education | Текстовый | Образование сотрудника |
| Institution | Текстовый | Учебное заведение |
| HireDate | Дата | Дата найма сотрудника |
| DateOfDismissal | Дата | Дата увольнения сотрудника |
| Salary | Денежный | Зарплата сотрудника |
| Experience | Целочисленный | Стаж сотрудника |
| Premium | Денежный | Надбавка сотрудника |
| Characteristic | Текстовый | Характеристика сотрудника |

**Таблица 3.4**

Описание таблицы Employee\_measures – таблица участия сотрудников в мероприятиях

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_employee | Целочисленный | Код сотрудника, участвующий в проведении мероприятия |
| ID\_measure | Целочисленный | Код мероприятия |
| Functions | Текстовый | Функция сотрудника в мероприятии |
| Descriptions | Текстовый | Описание проводимого мероприятия |

**Таблица 3.5**

Описание таблицы Graduate – таблица выпустившихся детей

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_graduate | Целочисленный | Идентификатор |
| ID\_ child | Целочисленный | Код выпустившегося ребенка |
| LastName | Текстовый | Фамилия ребенка |
| FirstName | Текстовый | Имя ребенка |
| MiddleName | Текстовый | Отчество ребенка |
| Paul | Текстовый | Пол ребенка |
| BirthDate | Дата | Дата рождения ребенка |
| DateOfAdoption | Дата | Дата принятия ребенка |
| ReleaseDate | Дата | Дата выпуска ребенка |
| Characteristic | Текстовый | Характеристика ребенка |
| AdditionalInformation | Текстовый | Дополнительная информация |

**Таблица 3.6**

Описание таблицы Groups – таблица групп детского сада

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_group | Целочисленный | Идентификатор |
| Name | Текстовый | Название группы |
| Number | Целочисленный | Номер группы |
| Descriptions | Текстовый | Описание группы |

**Таблица 3.7**

Описание таблицы Group\_measures – таблица проведения мероприятий в группах

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_group | Целочисленный | Код\_группы, в котором проводится мероприятие |
| ID\_measure | Целочисленный | Код\_мероприятия |

**Таблица 3.8**

Описание таблицы Measures – таблица мероприятий, проводимых в детском саду

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_measure | Целочисленный | Идентификатор |
| Name | Текстовый | Название мероприятия |
| DateOf | Дата | Дата проведения мероприятия |
| Descriptions | Текстовый | Описание мероприятия |
| Sites | Текстовый | Место проведения мероприятия |

**Таблица 3.9**

Описание таблицы MedicalEvents – таблица медицинских мероприятий, проводимых в детском саду

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_medicalEvents | Целочисленный | Идентификатор |
| Name | Текстовый | Название медицинского мероприятия |
| Type | Текстовый | Тип медицинского мероприятия |
| Descriptions | Текстовый | Описание медицинского мероприятия |
| StartDate | Дата | Дата начала медицинского мероприятия |
| EndDate | Дата | Дата конца медицинского мероприятия |
| ID\_responsible | Целочисленный | Код ответственного, проводившего медицинского мероприятие |
| Venue | Текстовый | Место проведения |

**Таблица 3.10**

Описание таблицы Position – таблица должностей сотрудников детского сада

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_position | Целочисленный | Идентификатор |
| Name | Текстовый | Название должности |
| Salary | Денежный | Зарплата |
| AdditionalInformation | Текстовый | Дополнительная информация |

**Таблица 3.11**

Описание таблицы Relations – таблица отношений между ребенком и родственником

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_child | Целочисленный | Код ребенка |
| ID\_relative | Целочисленный | Код родственника, относящийся к ребенку |
| Name\_relation | Текстовый | Название типа отношения |

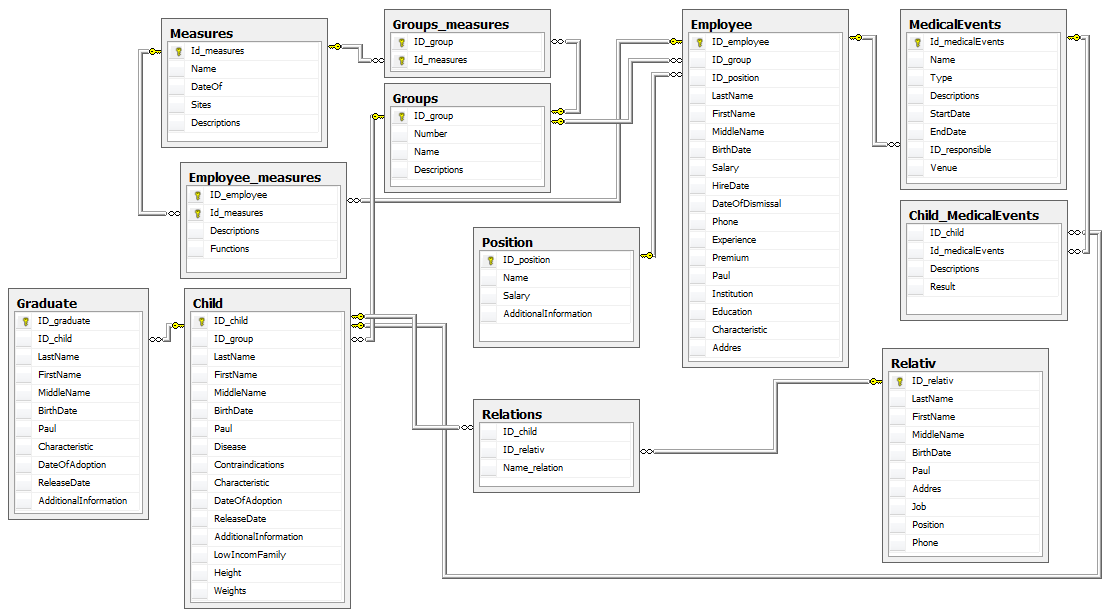
**Таблица 3.12**

Описание таблицы Relative – таблица родственников ребенка

| Наименование поля | Тип данных | Описание |
| --- | --- | --- |
| ID\_relative | Целочисленный | Идентификатор |
| LastName | Текстовый | Фамилия родственника |
| FirstName | Текстовый | Имя родственника |
| MiddleName | Текстовый | Отчество родственника |
| Paul | Текстовый | Пол родственника |
| Phone | Текстовый | Телефон родственника |
| Address | Текстовый | Адрес проживания родственника |
| Job | Текстовый | Место работы родственника |
| Position | Текстовый | Должность родственника |

На рис. 3.1 приведена структура базы данных детского сада, в котором реализуется два вида связи:

* «Один-к-одному»
* «Один-ко-многим»



**Рис. 3.1 Структура базы данных.**

Разработанное приложение и база данных взаимодействуют между собой посредством хранимых процедур.

### 3.2. Описание веб-интерфейса и примеры работы приложения.

Интерфейс приложения разработан в максимально простой и понятной для пользователя форме и обеспечивает удобный доступ к основным функциям системы.

### 3.2.1. Описание интерфейса внешней части информационной системы МБДОУ «Детский сад с.Яковлевка».

Данное веб-приложение подразумевает общий доступ к внешней части информационной системы, разработанной в виде веб-сайта. Для всех страниц имеется один общий шаблон (\_LayoutSite.cshtml). Навигация и структура на всех страницах одинакова. Поэтому, навигация по сайту проста и понятна для неподготовленного пользователя. Графическое оформление ненавязчиво и не мешает воспринимать информацию.

В связи с тем, что данное веб-приложение разрабатывалось для конкретного детского сада, то на «шапке» сайта размещены логотип Базарно-Карабулакского муниципального района Саратовской области, а также название дошкольного учреждения - Муниципальное бюджетное дошкольное образовательное учреждение «Детский сад с.Яковлевка». Нижняя часть сайта содержит информацию о создателе данного приложения и текущий год.

Ниже перечислены основные разделы сайта:

* Главная страница (Index.cshtml);
* О нас (Visit.cshtml);
* Фотогалерея (Gallery.cshtml);
* Наш коллектив (Collective.cshtml);
* Наши группы (Groups.cshtml);
* Родителям (Parents.cshtml);
* Контакты (Contact.cshtml);
* Карта сайта (Card.cshtml);
* Служебный вход (Login.cshtml).

Для неавторизованных пользователей имеется полный доступ к вышеперечисленным страницам сайта.

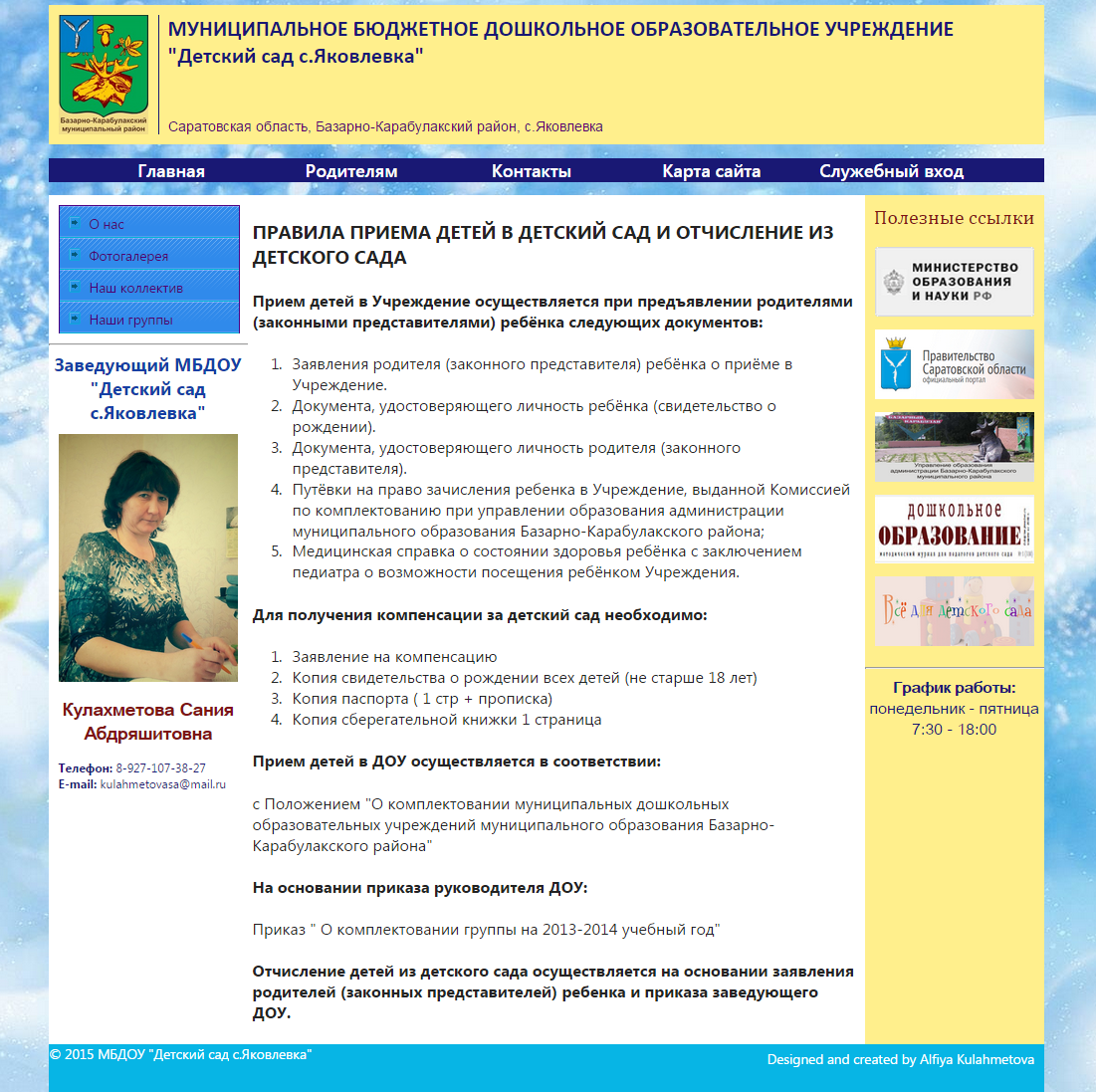
Начальной страницей, с которой начинается загрузка веб-приложения в браузере пользователя является Главная страница (Index.cshtml) (рис. 3.1), на которой находится меню сайта, размещены новости, а также имеется слайд-шоу с краткой информацией о детском саде. Для создания слайд-шоу использовался язык программирования JavaScript. Справа располагаются полезные ссылки, при нажатии на которые происходит переход на нужную страницу.



**Рис. 3.1. Главная страница веб-сайта.**

Главное меню сайта содержит пять разделов: «Главная», «Родителям», «Контакты», «Карта сайта», «Служебный вход».

В разделе «Родителям» размещена информация для родителей. В нем описывается правила приема детей в детский сад и отчисление из детского сада (рис.3.2).



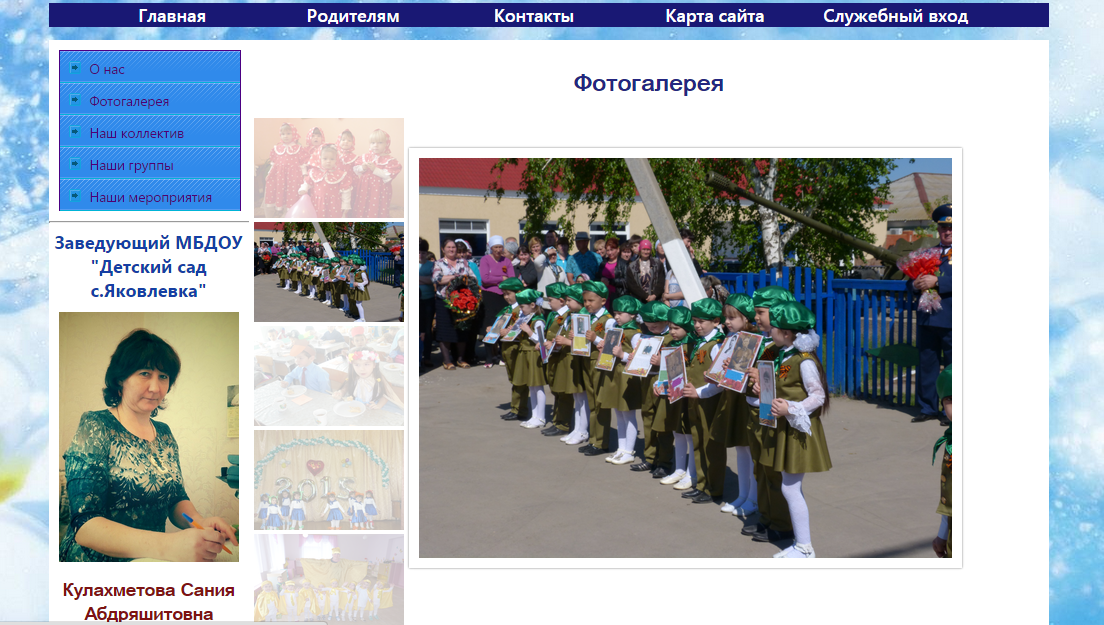
**Рис. 3.2. Страница раздела «Родителям».**

В разделе «Контакты» находятся контактные данные дошкольного образовательного учреждения. «Карта сайта» предоставляет быстрый доступ к необходимым страницам сайта.

Слева на сайте располагается дополнительное меню, содержащая следующие разделы: «О нас», «Фотогалерея», «Наш коллектив», «Наши группы».

В разделе «О нас» размещены основные сведения образовательного учреждения и приводится подробное описание данного учреждения.

В разделе «Фотогалерея» размещены фотографии детей из жизни детского сада.

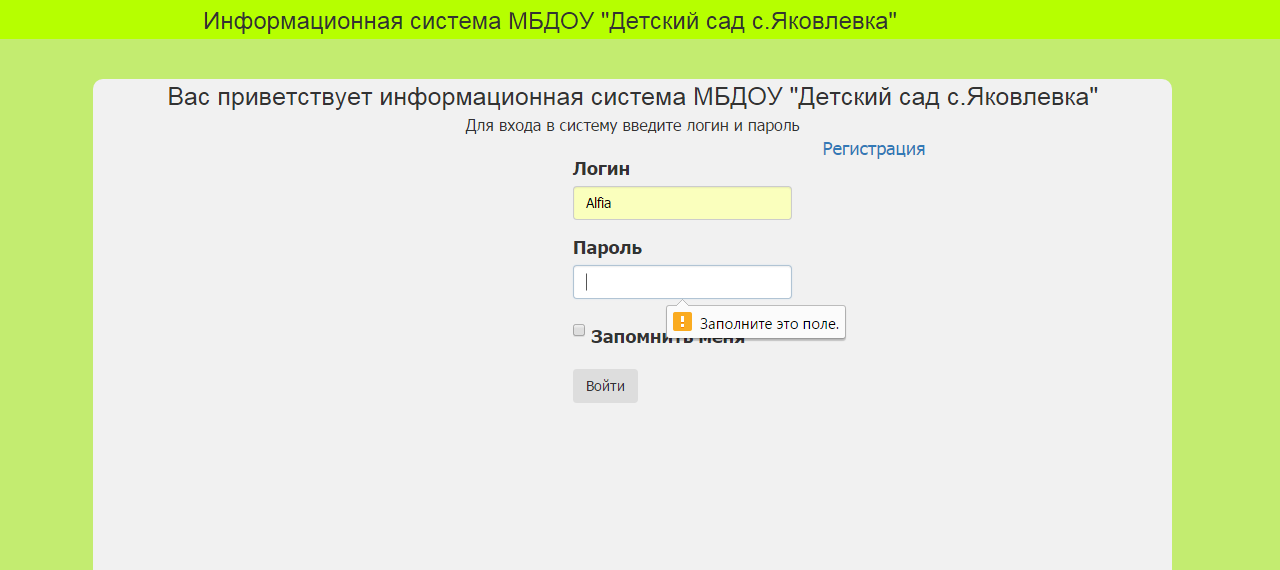


**Рис. 3.2. Страница раздела «Фотогалерея».**

Страница раздела «Наш коллектив» содержит краткую информацию о каждом сотруднике детского учреждения.

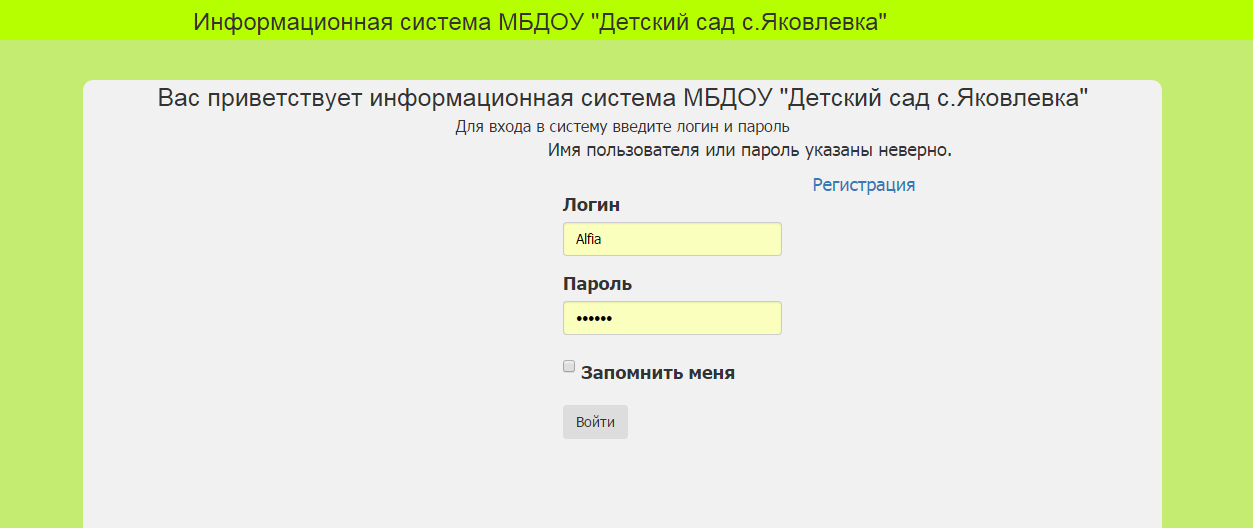
### 3.2.1. Описание интерфейса внутренней части информационной системы МБДОУ «Детский сад с.Яковлевка».

Внутренняя часть веб-приложения (информационная система) доступна только для сотрудников дошкольного учреждения, у которых имеются учетная запись для входа в систему. Для того, чтобы войти в информационную систему создано представление Login.cshtml (рис. 3.2), в котором нужно ввести логин и пароль. написать про роли



**Рис. 3.2. Авторизация пользователя.**

При неверном вводе данных выводится сообщение об ошибке (рис.3.3.).



**Рис. 3.3. Ошибка при авторизации пользователя.**

Для работы с пользователями и ролями в информационной системе МБДОУ «Детский сад с.Яковлевка» использовался провайдер SimpleMembershipProvider, который обладает большой гибкостью и расширяемостью. Он решает такие задачи как, авторизация и регистрация пользователей на сайте с использованием связки логина и пароля, хранения профиля в базе данных SQL Server и приспосабливается к современным требованиям обеспечения безопасности.

Если вход в систему пройдет успешно, то пользователь попадает на главную страницу, в котором находится главное меню с разделами данной информационной системы (рис. 3.2). Ниже перечислены разделы:

* Главная страница(Index.cshtml);
* Дети(Children.cshtml);
* Группы(Groups.cshtml);
* Родственники(Relative.cshtml);
* Сотрудники(Employee.cshtml);
* Должности(Position.cshtml);
* Мероприятия(Measures.cshtml);
* Медицинские мероприятия(MedicalEvents.cshtml);
* Отчеты(Reports.cshtml);
* Добавление пользователя(AddEmployee.cshtml).

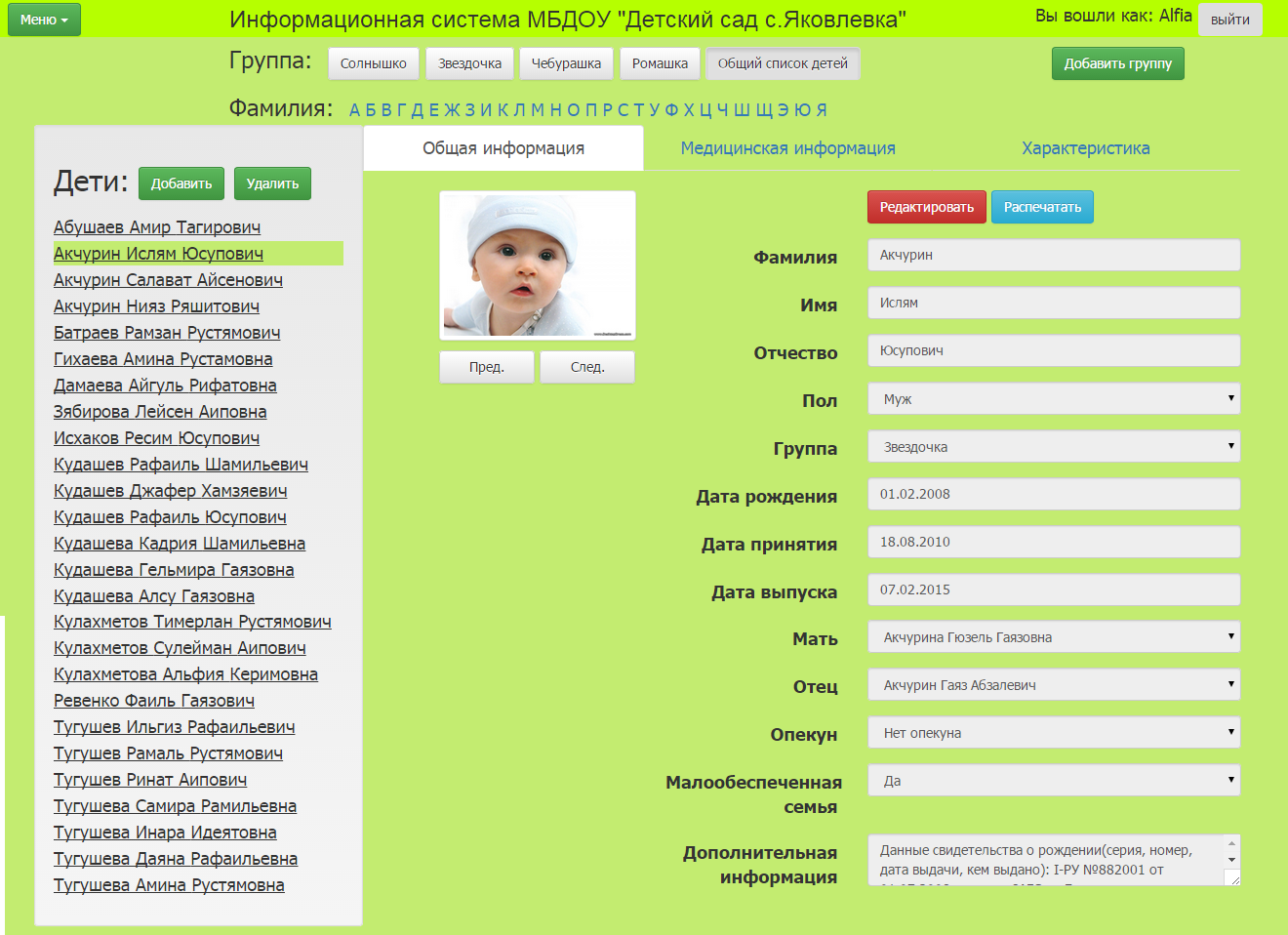
При наведении курсора на каждый из этих разделов появляется всплывающая подсказка с краткой информацией о функциональном наборе.



**Рис. 3.4. Главная страница внутренней части информационной системы МБДОУ Детский сад с.Яковлевка.**

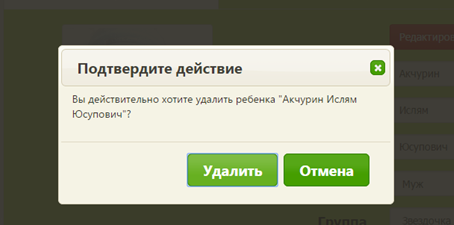
Все страницы внутренней части веб-приложения имеют единый шаблон (\_LayoutApp.cshtml), на котором в верхней части размещены быстрое меню приложения, название информационной системы и имя вошедшего пользователя, а также кнопка «Выйти» для того, чтобы выйти из системы. При нажатии на кнопку «Выйти» пользователь перенаправляется на главную страницу сайта внешней части информационной системы.

При выборе раздела «Дети» пользователь переходит на представление Children.cshtml (рис.3.5), в котором представлена вся информация о детях детского сада, хранящаяся в БД. По умолчанию слева располагается общий список детей дошкольного учреждения, отсортированных по фамилии ребенка. При нажатии на конкретную группу появляется список детей выбранной группы. Выбрав ребенка, пользователь может редактировать данные по нему, а также нажатием на кнопку «Распечатать» отправить данные на печать. Находясь в режиме редактирования или добавления ребенка возможно добавление нового родственника с общей информацией и типом отношения к ребенку. Помимо этого, можно выбрать родственника из имеющегося списка (рис. 3….).



**Рис. 3.5. Страница с информацией о детях.**

Также, имеется возможность добавления нового ребенка при нажатии на кнопку «Добавить», а также удаления ребенка из имеющегося списка. Выбрав ребенка и нажав на кнопку «Удалить», осуществляется проверка на подтверждение действия (рис. 3.6). При подтверждении пользователем действия происходит удаление ребенка, в противном случае - возвращение в исходное состояние.

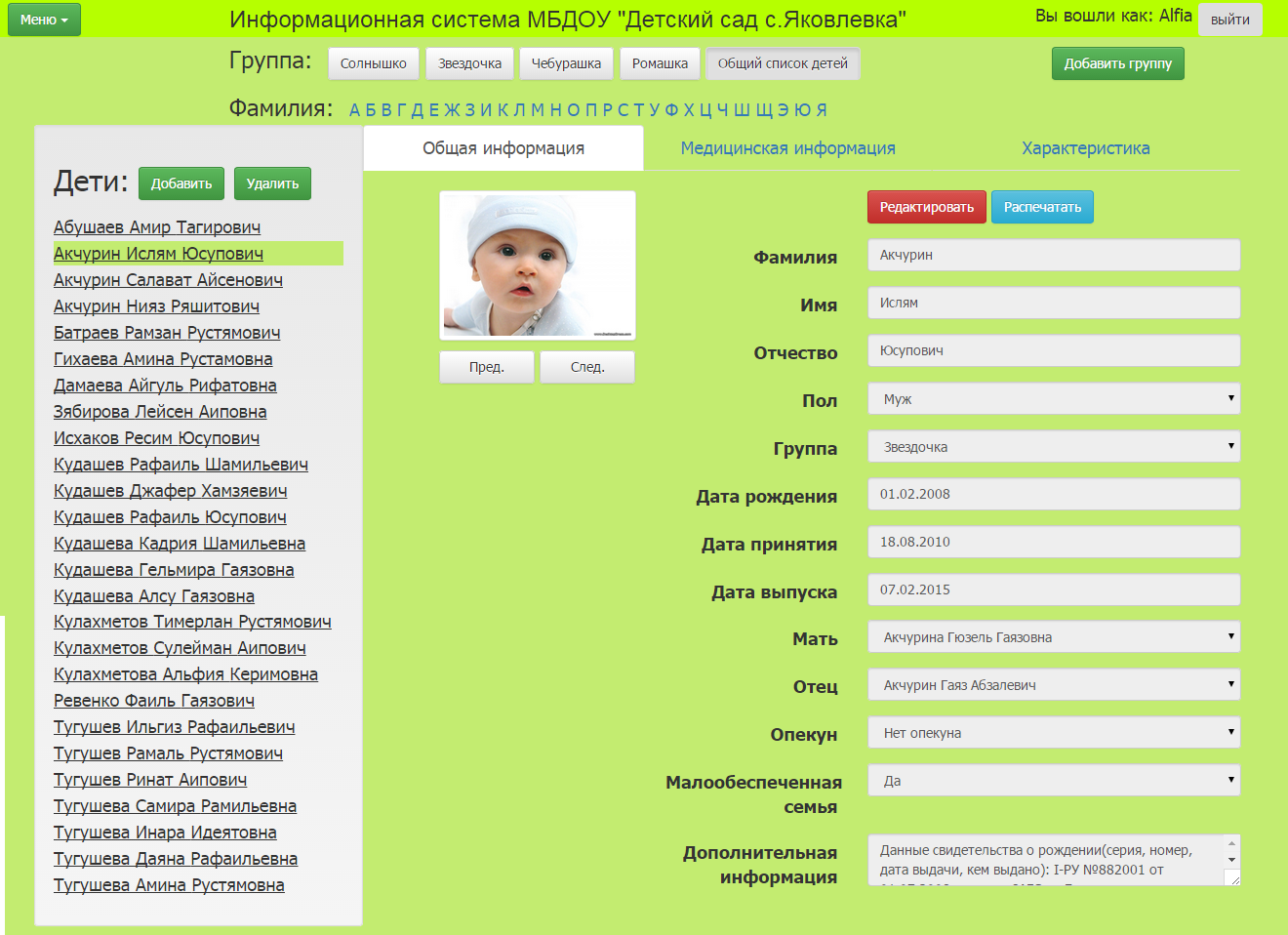


**Рис. 3.6. Проверка на удаление ребенка.**

Страница Children.cshtml включает в себя следующие данные:

* личная информация о ребенке (рис.3.7)

Личная карточка ребенка содержит общую информацию о ребенке.



**Рис. 3.7. Личная карточка ребенка.**

* медицинская информация о ребенке (рис.3.6)

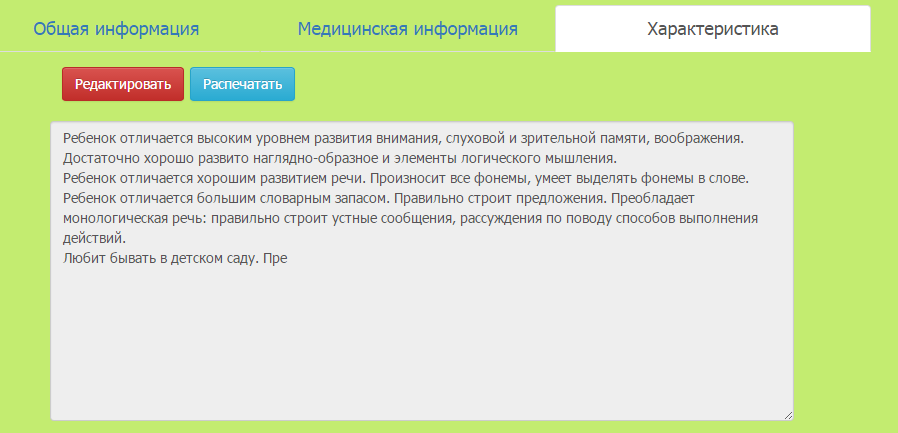
Медицинская карточка содержит медицинскую информацию о ребенке, а также результаты участия ребенка в медицинских мероприятиях.



**Рис. 3.8. Медицинская карточка ребенка.**

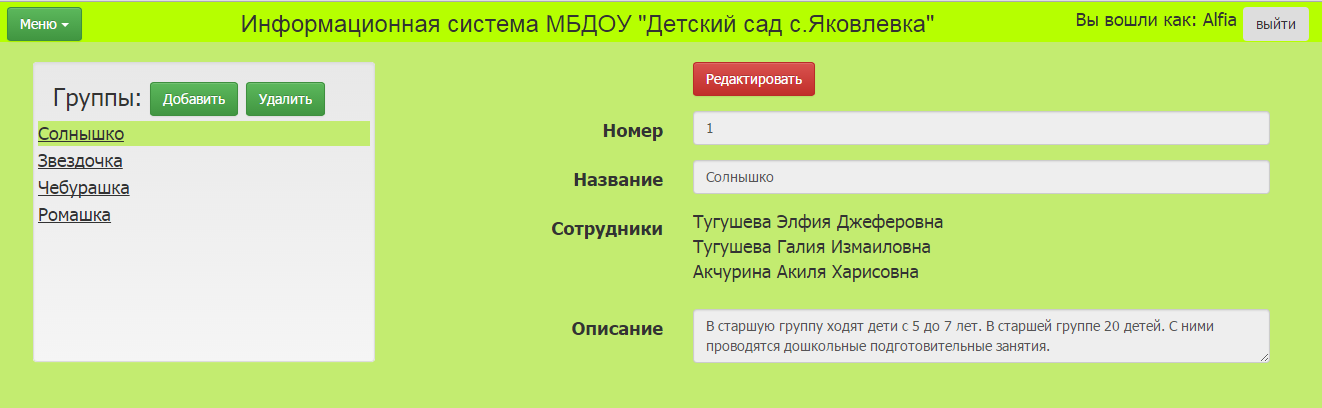
* характеристика ребенка (рис.3.8)

Данная вкладка содержит характеристику ребенка.



**Рис. 3.8. Характеристика ребенка.**

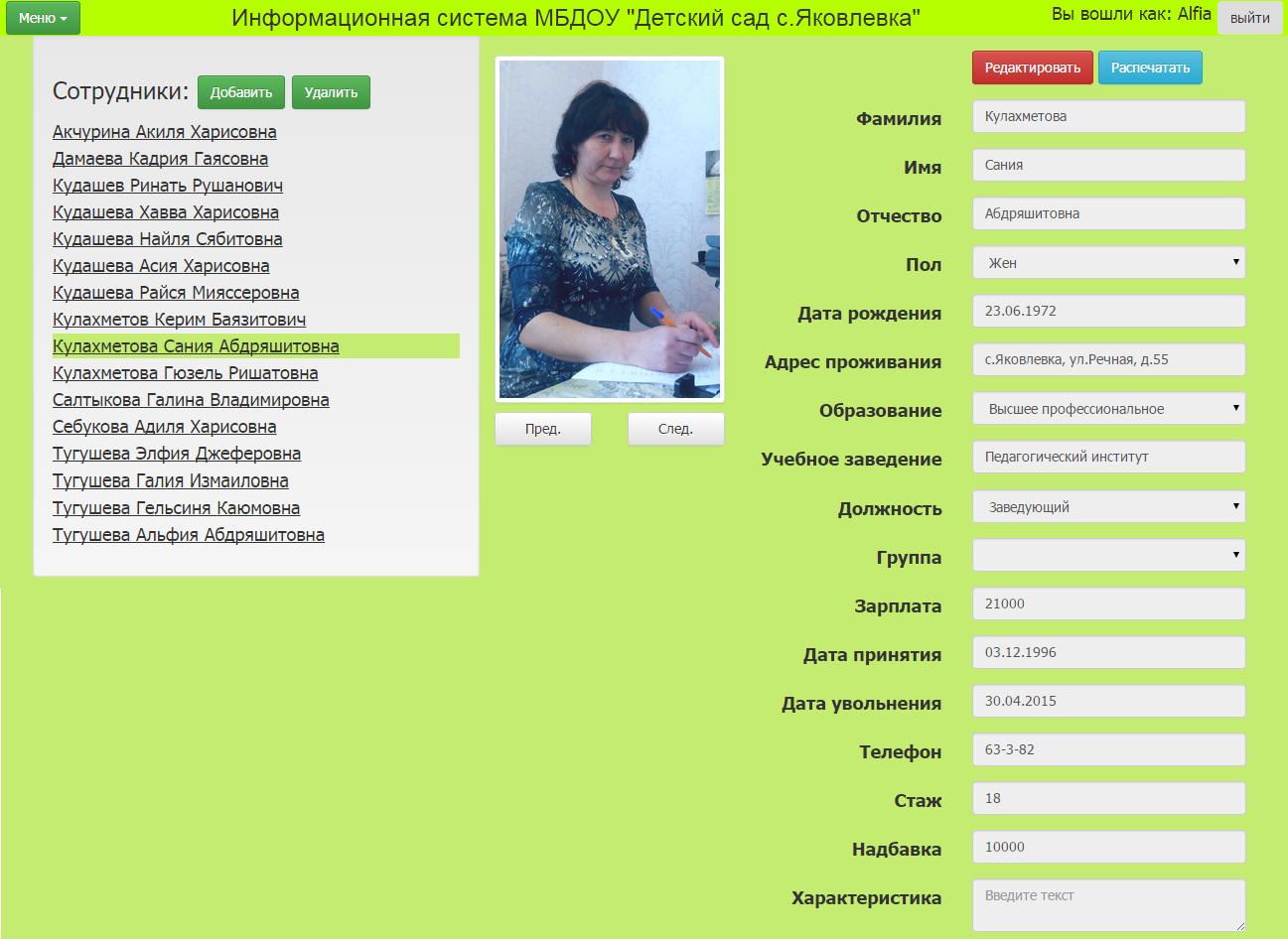
При нажатии на кнопку «Добавить группу» происходит переход на представление Groups.cshtml (рис.3.9), в котором располагается информация обо всех группах детского сада. На данной странице имеется возможность добавления и удаления группы, а также редактирования информации о группах.



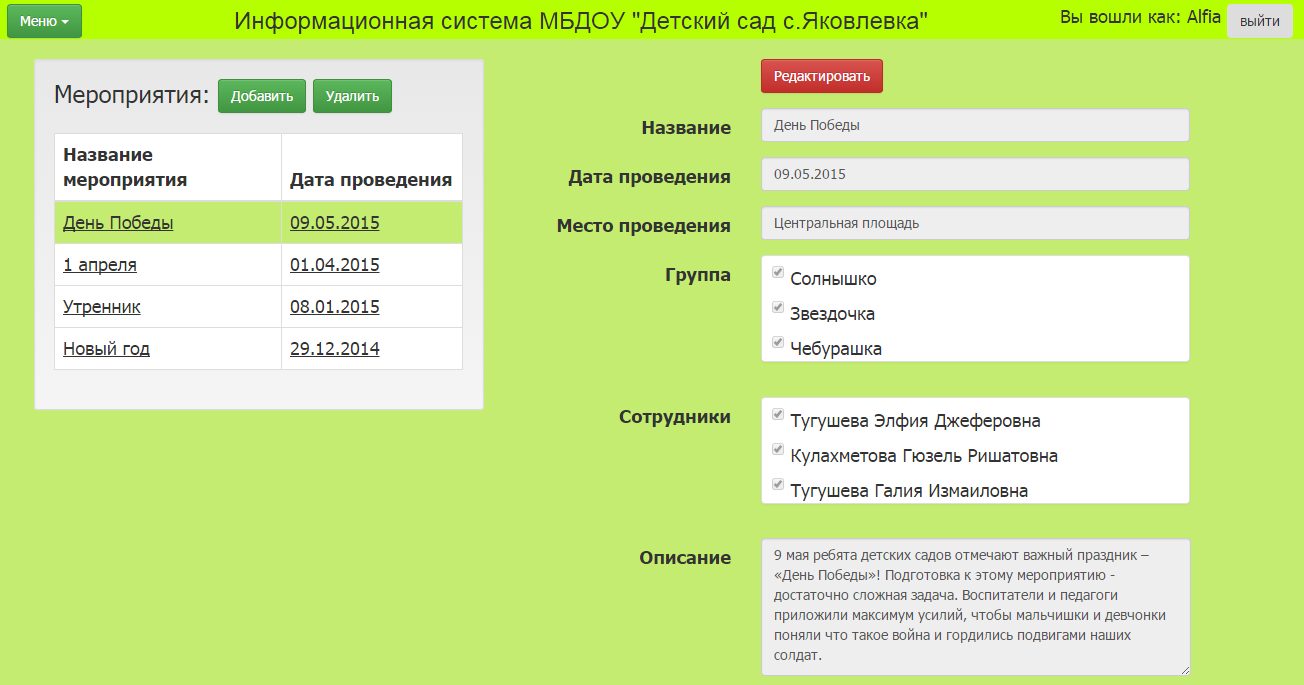
**Рис. 3.9. Страница с информацией о группах.**

Представление Employee.cshtml (рис. 3.10) содержит информацию о сотрудниках детского сада. Слева располагается общий список сотрудников, отсортированных по фамилии в алфавитном порядке. При нажатии на определенного сотрудника справа появляется его фотография, а также информация, которую можно редактировать или отправить на печать.

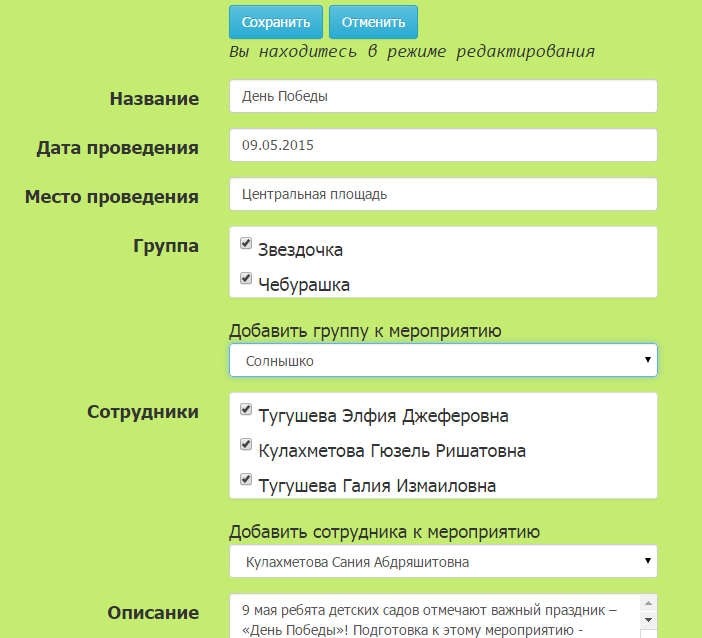
Помимо этого имеется возможность добавления нового сотрудника, либо удаление из имеющегося списка.



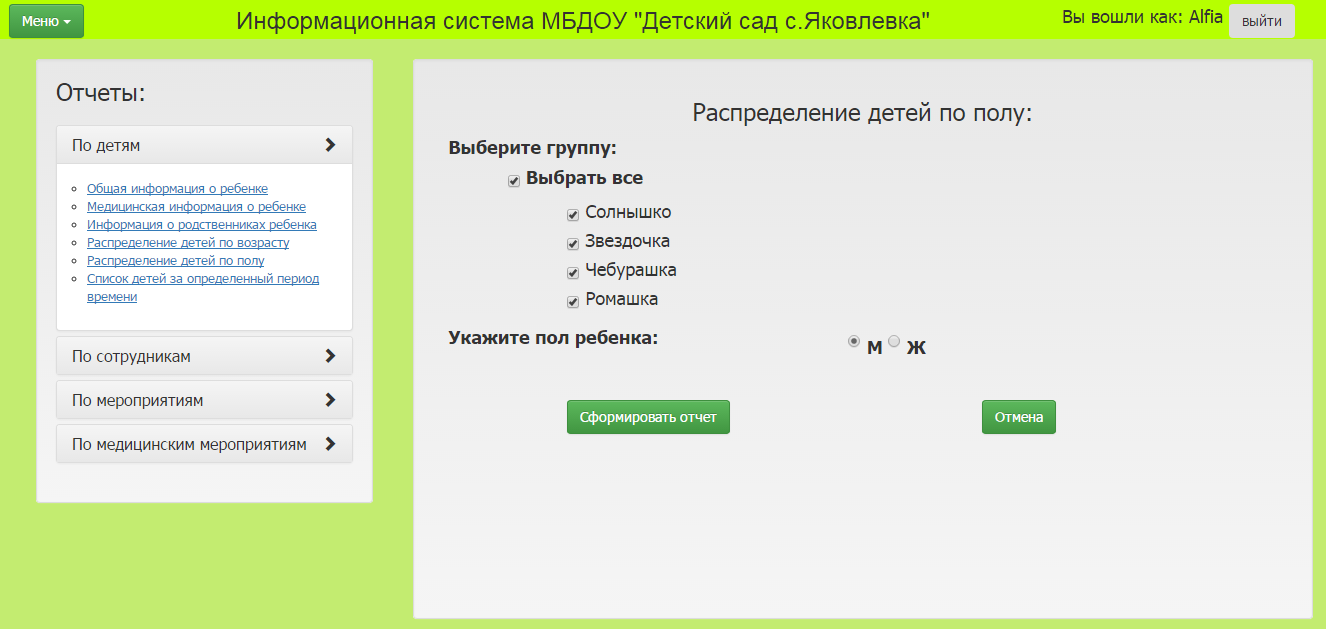
**Рис. 3.9. Страница с информацией о сотрудниках.**



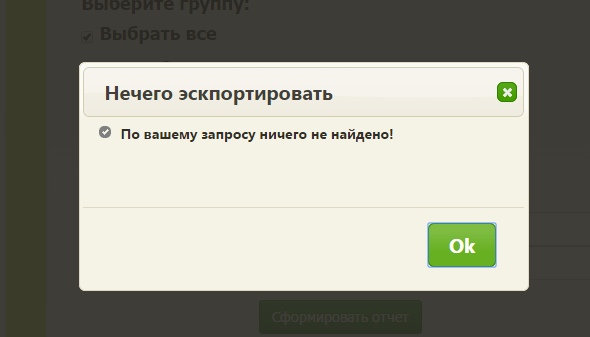
**Рис. 3.9. Страница с информацией о мероприятиях.**



**Рис. 3.9. Режим редактирования мероприятий**



**Рис. 3.10. Форма с выбором отчета.**

****

**Рис. 3.10. Проверка на экспорт отчета.**

Регистрация происходит на странице «Добавление пользователя» (AddEmployee.cshtml) (рис. 3.3). Регистрировать новых пользователей данной информационной системы может только заведующий детским садом. При регистрации необходимо заполнить следующие поля (фамилия, имя, отчество, дата рождения, пароль) и нажать на кнопку «Зарегистрировать».

# **Заключение**

# **Список использованных источников**

1. О принципах построения модели IDEF0 - [Электронный ресурс]. - 2015 - URL: http://www.intuit.ru/studies/courses/2195/55/lecture/1630 (Дата обращения: 10.05.2015).
2. Отношения классов от UML к коду – [Электронный ресурс]. – 2012 - URL: <http://habrahabr.ru/post/150041/> (Дата обращения: 12.04.2015).
3. Методологии моделирования предметной области – [Электронный ресурс]. – 2011 – URL: <http://www.intuit.ru/studies/courses/2195/55/lecture/1628?page=5> (Дата обращения: 12.04.2015).
4. Борисова, С.В., Крюков, А.Ф., Шелехин, А.Г. Сравнение технологий ASP.NET Web Forms и ASP.NET MVC для создания WEB-приложений // Труды XIX Международной научно-технической конференции "Информационные средства и технологии", т.1 — М.: Издательский дом МЭИ, 2011, — с. 255-258
5. Магдануров, Г. И. ASP.NET MVC Framework [Текст] / Г. И. Магдануров, В. А. Юнев. — СПб.: БХВ-Петербург, 2010. — 320 с.
6. Иванов А.В. Управление данными: учеб. пособие для студентов специальностей "Информационные системы и технологии", "Прикладная информатика в экономике", направлений "Информационные системы", "Информационные технологии", "Информатика и вычислительная техника", "Программная инженерия", "Информационное и программное обеспечение автоматизированных систем"// М-во образования и науки Рос. Федерации, Cарат. гос. техн. ун-т. - Саратов: СГТУ, 2009. - 88 с.
7. Шилдт Г. Полный справочник по C#: [пер. с англ.] / Г. Шилдт. – М.: Издательский дом “Вильямс”, 2004. – 752 с. : ил.
8. Эспозито Д. Microsoft ASP.NET 2.0. Углубленное изучение. [пер. с англ.] / Д. Эспозито – М.: Издательство «Русская Редакция»; СПб.: Питер, 2008. – 592 с.: ил.
9. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. Мастер\_класс: [пер. с англ.] / Дж. Рихтер – 2-е изд., исправ. М.: Издательство «Русская Редакция»; СПб: Питер , 2008. – 656 стр.: ил.
10. Нейгел К. С# 2008 и платформа .NET 3.5 для профессионалов: [пер. с англ.] / К. Нейгел, Б. Ивьен, Дж. Глинн, К. Уотсон, М. Скиннер – М.: ООО "И.Д. Вильяме", 2009. – 1392 с.: ил.