

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики, математики и электроники
Факультет информатики
Кафедра информационных систем и технологий

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

«АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ
ПОВЫШЕНИЯ ЭФФЕКТИВНОСТИ КОММУНИКАЦИОННЫХ ПРОЦЕССОВ
В ДЕТСКИХ ДОШКОЛЬНЫХ УЧРЕЖДЕНИЯХ»

по направлению подготовки 09.03.01 Информатика
и вычислительная техника
(уровень бакалавриата)
профиль «Информационные системы»

Обучающийся _____ В.А. Виханов
(подпись, дата)

Руководитель ВКР,
доцент кафедры ИСТ, к.т.н., доцент _____ И.А. Лёзин
(подпись, дата)

Нормоконтролер _____ Я.В. Соловьева
(подпись, дата)

Самара 2020

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное
образовательное учреждение высшего образования
«Самарский национальный исследовательский университет
имени академика С.П. Королева»
(Самарский университет)

Институт информатики, математики и электроники
Факультет информатики
Кафедра информационных систем и технологий

УТВЕРЖДАЮ

Заведующий кафедрой ИСТ

_____ С.А. Прохоров

« ____ » _____ 20__ г.

ЗАДАНИЕ НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

обучающемуся Виханову Владимиру Алексеевичу

группа № 6404-090301D

Тема работы: Автоматизированная информационная система для
повышения эффективности коммуникационных процессов в детских
дошкольных учреждениях

утверждена приказом по университету № 137-Т от «16» апреля 2020г.

Исходные данные к работе:

- тип операционной системы: Windows 7 и выше; _____
- язык программирования: C#; _____
- среда разработки: Visual Studio 2019; _____
- СУБД: MS SQL; _____

Перечень вопросов, подлежащих разработке в работе:

- анализ предметной области; _____

- аналитический обзор существующих систем-аналогов; _____
- разработка логического проекта системы; _____
- разработка логической и физической модели базы данных; _____
- разработка и отладка программного обеспечения; _____

Руководитель работы

доцент кафедры ИСТ, к.т.н., доцент _____ И.А. Лёзин
(подпись)

« ____ » _____ 20 ____ г.

Задание принял к исполнению _____ В.А. Виханов
(подпись)

« ____ » _____ 20 ____ г.

РЕФЕРАТ

Пояснительная записка содержит 82 страницы, 35 рисунков, 9 таблиц, 22 источника, 5 приложений.

Презентация: 18 слайдов Microsoft PowerPoint.

ДОШКОЛЬНЫЕ УЧРЕЖДЕНИЯ, ВОСПИТАТЕЛЬНО-ОБРАЗОВАТЕЛЬНЫЕ МЕРОПРИЯТИЯ, ВЗАИМОДЕЙСТВИЕ РОДИТЕЛЕЙ И ВОСПИТАТЕЛЕЙ, КОММУНИКАЦИОННЫЕ ПРОЦЕССЫ, АВТОМАТИЗИРОВАННАЯ ИНФОРМАЦИОННАЯ СИСТЕМА, БАЗА ДАННЫХ.

Объектом исследования являются коммуникационные процессы, связанные с взаимодействием родителей и воспитателей детского дошкольного учреждения.

Предметом исследования является автоматизированная информационная система, обеспечивающая взаимодействие родителей и воспитателей детского дошкольного учреждения.

Целью работы является повышение эффективности коммуникационных процессов посредством автоматизации процессов взаимодействия родителей и воспитателей детских дошкольных учреждений.

Актуальность данной работы заключается в необходимости повышения вовлеченности участников и эффективности коммуникационного процесса взаимодействия родителей и воспитателей. Необходимо спроектировать и разработать автоматизированную информационную систему, внедрение которой позволит улучшить качество воспитательных и образовательных процессов, а также повысить эффективность работы детских дошкольных учреждений в целом.

Сформулированные методические основы автоматизации коммуникационного процесса взаимодействия родителей и воспитателей детского дошкольного учреждения могут быть использованы коммерческими и некоммерческими организациями для разработки модели информационной архитектуры, реализующей переход к современной платформе управления.

СОДЕРЖАНИЕ

Перечень сокращений и условных обозначений	7
Введение.....	8
1 Анализ предметной области и формирование требований к автоматизированной информационной системе	11
1.1 Анализ существующей организации коммуникационного процесса взаимодействия родителей и воспитателей.....	11
1.1.1 Описание существующей организации коммуникационного процесса	11
1.1.2 Анализ недостатков существующей организации коммуникационного процесса	13
1.1.3 Обзор имеющихся решений выявленных недостатков	15
1.2 Формирование предложений по автоматизации коммуникационных процессов взаимодействия родителей и воспитателей	18
1.3 Разработка требований к проектируемой автоматизированной информационной системе	20
1.4 Обоснование выбора инструментов разработки автоматизированной информационной системы	24
1.4.1 Обоснование выбора системы управления базой данных	24
1.4.2 Обоснование выбора среды разработки	25
1.5 Обоснование выбора архитектуры автоматизированной информационной системы	27
1.6 Постановка задачи исследования	29
2 Проектирование автоматизированной информационной системы для детских дошкольных учреждений	30
2.1 Функциональная структура проектируемой автоматизированной информационной системы	30
2.2 Проектирование базы данных автоматизированной информационной системы	31
2.2.1 Логическое проектирование	31

2.2.2 Физическое проектирование.....	36
2.3 Проектирование автоматизированной информационной системы	37
2.3.1 Диаграмма прецедентов	37
2.3.2 Диаграмма последовательности	39
2.3.3 Диаграмма состояний	40
2.4 Проектирование структуры автоматизированной информационной системы	41
2.5 Основные алгоритмы, используемые в автоматизированной информационной системе	43
2.6 Выводы по главе.....	45
3 Разработка автоматизированной информационной системы для детских дошкольных учреждений	47
3.1 Контрольный пример.....	47
3.2 Проверка эффективности системы.....	49
3.3 Разработка пользовательской документации.....	51
3.4 Выводы по главе.....	51
Заключение	52
Список использованных источников	54
Приложение А	56
Приложение Б	57
Приложение В.....	61
Приложение Г	62
Г.1 Разработка руководства пользователя с ролью «Воспитатель»	63
Г.2 Разработка руководства пользователя с ролью «Родитель»	68
Приложение Д.....	70

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ И УСЛОВНЫХ ОБОЗНАЧЕНИЙ

IDEF – Методология функционального моделирования

MVC – Model view controller

SADT – Structured Analysis and Design Technique

SQL – Structured Query Language

TCP/IP – Transmission Control Protocol/Internet Protocol

UML – Unified Modeling Language

АИС – Автоматизированная информационная система

БД – База данных

ДОУ – Детское образовательное учреждение

ДДУ – Детское дошкольное учреждение

ОС – Операционная система

СУБД – Система управления базой данных

ДЗ – Домашнее задание

ПО – Программное обеспечение

ВВЕДЕНИЕ

Сегодня без современных информационных технологий трудно представить управление детскими дошкольными учреждениями на всех уровнях, а также организацию воспитательных и образовательных процессов в целом. Так, например, за счет внедрения информационных систем и ресурсов у воспитателей, медсестер и других специалистов появляется доступ к оперативной информации, знаниям, идеям и культурным ценностям, а также создание условий для самообразования сотрудников [1].

Кроме того, сегодня дошкольными учреждениями активно формируют каталоги ресурсов, в которых собраны обучающие материалы как для детей, так и для сотрудников, в том числе мультфильмы и другие аудио и видеоматериалы, сайты тематических интернет проектов и сетевых сообществ, а также тематические ресурсы по отдельным направлениям. Другими словами, в подавляющем большинстве детских учреждений определенные аспекты деятельности сопряжены с использованием современных информационных подходов [2,3,4].

Несмотря на этот факт одной из наиболее острых современных проблем на уровне детских дошкольных учреждений является недостаточный уровень осведомления родителей о воспитательной деятельности и проводимых дополнительно к основному расписанию мероприятиях в детском саду. В связи с тем, что дети не могут предоставить родителям исчерпывающую и достоверную информацию о тех или иных аспектах воспитательно-образовательных мероприятий, которые проводятся в дошкольных учреждениях на периодической основе, родителям необходимо предоставлять такую информацию [5,6].

Одним из решений описанной проблемы является оптимизация процессов, связанных с взаимодействием родителей и воспитателей детского дошкольного учреждения. Другими словами, актуальность данной работы заключается в необходимости повышения вовлеченности участников и эффективности коммуникационного процесса взаимодействия родителей и воспитателей.

Необходимо спроектировать и разработать автоматизированную информационную систему, внедрение которой позволит улучшить качество воспитательных и образовательных процессов, а также повысить эффективность работы детских дошкольных учреждений в целом [6,7].

Объектом исследования являются коммуникационные процессы, связанные с взаимодействием родителей и воспитателей детского дошкольного учреждения.

Предметом исследования является автоматизированная информационная система, обеспечивающая взаимодействие родителей и воспитателей детского дошкольного учреждения.

Целью работы является повышение эффективности коммуникационных процессов посредством автоматизации процессов взаимодействия родителей и воспитателей детских дошкольных учреждений.

1) Провести анализ существующей организации коммуникационного процесса взаимодействия родителей и воспитателей детского дошкольного учреждения;

2) Провести обзор имеющихся решений выявленных недостатков существующей организации коммуникационного процесса;

3) Составить спецификацию и провести обоснование функциональных и нефункциональных требований;

4) Спроектировать базу данных и автоматизированную информационную систему, обеспечивающую взаимодействие родителей и воспитателей детского дошкольного учреждения;

5) Разработать автоматизированную информационную систему, обеспечивающую взаимодействие родителей и воспитателей детского дошкольного учреждения;

6) Разработать пользовательскую документацию по порядку использования автоматизированной информационной системы, обеспечивающей взаимодействие родителей и воспитателей детского дошкольного учреждения.

В рамках данной работы применены следующие методики:

- 1) Унифицированный язык моделирования UML;
- 2) Объектно–ориентированная методика.

Практическая значимость работы обусловлена тем, что внедрение автоматизированной информационной системы позволит повысить эффективность работы детского дошкольного учреждения за счет повышения качества коммуникационного процесса взаимодействия родителей и воспитателей.

Сформулированные методические основы автоматизации коммуникационного процесса взаимодействия родителей и воспитателей детского дошкольного учреждения могут быть использованы коммерческими и некоммерческими организациями для разработки модели информационной архитектуры, реализующей переход к современной платформе управления.

В первой главе проведен анализ предметной области, обзор имеющихся решений выявленных недостатков, формирование требований к автоматизированной информационной системе, а также выбор инструментов разработки системы взаимодействия родителей и воспитателей детского дошкольного учреждения.

Во второй главе спроектирована база данных и представлены модели автоматизированной информационной системы взаимодействия родителей и воспитателей детского дошкольного учреждения.

В третьей главе проведена проверка эффективности автоматизированной информационной системы взаимодействия родителей и воспитателей детского дошкольного учреждения, а также разработана пользовательская документация для сотрудников с соответствующими ролями.

1 Анализ предметной области и формирование требований к автоматизированной информационной системе

1.1 Анализ существующей организации коммуникационного процесса взаимодействия родителей и воспитателей

1.1.1 Описание существующей организации коммуникационного процесса

В рамках данной работы предполагается посредством использования информационных технологий решить проблему, связанную с коммуникационной составляющей взаимодействия родителей и воспитателей в рамках реализации воспитательных и образовательных процессов дошкольного учреждения. С точки зрения детского дошкольного учреждения основные задачи взаимодействия воспитателей и родителей можно представить в следующем перечне:

- 1) Создать атмосферу взаимопонимания;
- 2) Установить партнерские отношения с семьей каждого воспитанника;
- 3) Развивать и воспитывать детей;
- 4) Использовать с родителями различные формы сотрудничества [8,6].

Главная цель всех видов форм коммуникаций воспитателей и родителей заключается в формировании доверительных отношений между детьми, родителями и педагогами. Сами формы взаимодействия можно классифицировать следующим образом:

- 1) Коллективные — родительские собрания, конференции, консультации;
- 2) Индивидуальные — индивидуальные консультации и беседы;
- 3) Наглядные — стенды, дни открытых дверей.

Большой популярностью у родителей воспитанников детских дошкольных учреждений пользуется следующий ряд форм взаимодействия:

1) Информационно-аналитическое взаимодействие, направленное на выявление интересов, запросов, установление контакта между детьми, родителями и педагогами, например, анкетирование или опрос. Эта форма используется воспитателем с целью изучения семьи, выяснение

образовательных потребностей родителей. Получив данные, педагог вырабатывает тактику общения с каждым родителем;

2) Наглядно-информационное взаимодействие, направленное на обогащение знаний и информационное просвещение родителей. Эту форму можно разделить на две подгруппы:

– информационно-ознакомительная, целью которой является ознакомление родителей с учреждением и педагогами, например, совместные выставки детских рисунков и коллажей;

– информационно-просветительская, целью которой является обогащение знаний родителей об развитии и воспитании детей. Например, информационные стенды, мини-библиотеки и т.д [5].

Существует также ряд форм взаимодействия, которые вызывают меньший интерес у родителей воспитанников детского дошкольного учреждения:

1) Познавательные формы - ознакомление родителей с возрастными и психологическими особенностями детей, обогащение педагогического опыта. В этой группе лидируют родительские собрания;

2) Родительские конференции — повышение педагогической культуры родителей. На ней выступают педагоги, представители медицинской службы, психологи;

3) Клубы для родителей — педагог оказывает помощь в возникших проблемах;

4) Досуговые — это неформальные, доверительные отношения, происходит контакт между педагогами и родителями, между родителями и детьми. К этой группе можно отнести праздники, утренники, совместный досуг;

5) Посещение семей — обследование условий проживания ребенка [9,6].

Таким образом, коммуникационный процесс взаимодействия родителей и воспитателей характеризуется широким спектром составляющих. В рамках данной работы ключевой интерес представляют наглядно-информационное и информационно-аналитическое взаимодействия, связанные с повышением осведомленности родителей о потоковых воспитательно-образовательных

мероприятиях, которые проводятся в дошкольных учреждениях на еженедельной основе, и о домашних заданиях, которые получают дети для выполнения дома. Основные задачи проведения коммуникации воспитателей и родителей можно представить следующим перечнем:

- 1) Предоставить родителям информацию по проводимым воспитательно-образовательным мероприятиям;
- 2) Предоставить родителям информацию о домашних заданиях;
- 3) Установить с родителями партнёрские отношения по взаимодействию;
- 4) Активизировать участие родителей в подобных воспитательно-образовательных мероприятиях;
- 5) Получать обратную связь в целях реализации единой программы воспитания и развития ребенка в ДООУ и семье [7].

Для обеспечения коммуникационного процесса взаимодействия родителей и воспитателей необходимо:

- 1) Информировать родителей о ключевых аспектах мероприятия;
- 2) Информировать родителей о домашних заданиях;
- 3) Уточнять неясности по мероприятиям и домашним заданиям;
- 4) Организовать взаимодействие «вопрос-ответ»;
- 5) Обеспечить обратную связь по результатам мероприятий и выполненных домашних заданий;
- 6) Сформировать задания для дальнейших мероприятий и домашних работ.

В рамках дальнейшей работы необходимо выявить проблемы, возникающие в ходе организации коммуникационного процесса.

1.1.2 Анализ недостатков существующей организации коммуникационного процесса

С целью повышения эффективности коммуникационного процесса взаимодействия родителей и воспитателей с родителями была проведена оценка критических факторов успеха по пятибалльной шкале. Оценка получена экспертным путем при помощи опроса по пятибалльной системе (оценка 1 –

недопустимо, оценка 5 - отлично) 300 родителей и воспитателей. Опрос отражает информацию об отношении к различным аспектам коммуникационного процесса информирования о потоковых воспитательно-образовательных мероприятиях (таблица 1.1).

Таблица 1.1 – Оценка критических факторов успеха

Компоненты коммуникационного процесса взаимодействия родителей и воспитателей	Эффективность	Вовлеченность	Удобство	Средний балл
Формат и полнота предоставляемой информации	4,1	4	3,6	3,9
Коммуникации формата «вопрос-ответ»	3,5	4,1	3,2	3,6
Реализация обратной связи по тематике мероприятий	4,83	4,69	3,9	4,47
Формирование плана работ для дальнейших мероприятий	4,72	4,75	4,84	4,78
Реализация обратной связи по домашним заданиям	4,2	4,5	3,7	4,1

Из приведенных данных в таблице 1.1 можно сделать выводы о недостатках коммуникационного процесса взаимодействия родителей и воспитателей в дошкольных учреждениях в части удобства формата предоставляемой информации, организации коммуникации формата «вопрос-ответ» и обратной связи по тематике мероприятий и домашних заданий.

После анализа полученных результатов опроса стало ясно, что большинству родителей не удобно использовать существующие коммуникационные каналы взаимодействия. По этой причине многие вопросы часто остаются при себе, хотя, как правило, получение ответов на них весьма важно для эффективности воспитательных процессов.

С целью повышения эффективности воспитательных процессов и качества коммуникационного процесса взаимодействия родителей и воспитателей было принято решение об автоматизации коммуникационного процесса путем внедрения автоматизированной информационной системы, способной организовать взаимодействие воспитателей и родителей в форме чата с привязкой к тематике проводимых мероприятий и домашних заданий. Такое

решение должно не только повысить осведомленность и вовлеченность родителей в воспитательно–образовательные процессы детских дошкольных учреждений, но и предоставить педагогическому составу обратную связь по различным аспектам проведения того или иного мероприятия в ДООУ.

1.1.3 Обзор имеющихся решений выявленных недостатков

На сегодняшний день практически не существует популярных специализированных решений вопроса автоматизации взаимодействия родителей и воспитателей в ходе коммуникационных процессов. Выявленная в работе проблема решается путем использования смежных по функционалу и назначению программных продуктов. Проведем анализ нескольких таких решений в области организации взаимосвязи родителей и воспитателей и оценим возможность использования существующих программных средств.

Программный продукт «Viber» — это быстрое и защищённое от злоумышленников приложение для обмена сообщениями. Скриншот программы представлен на рисунке 1.1.

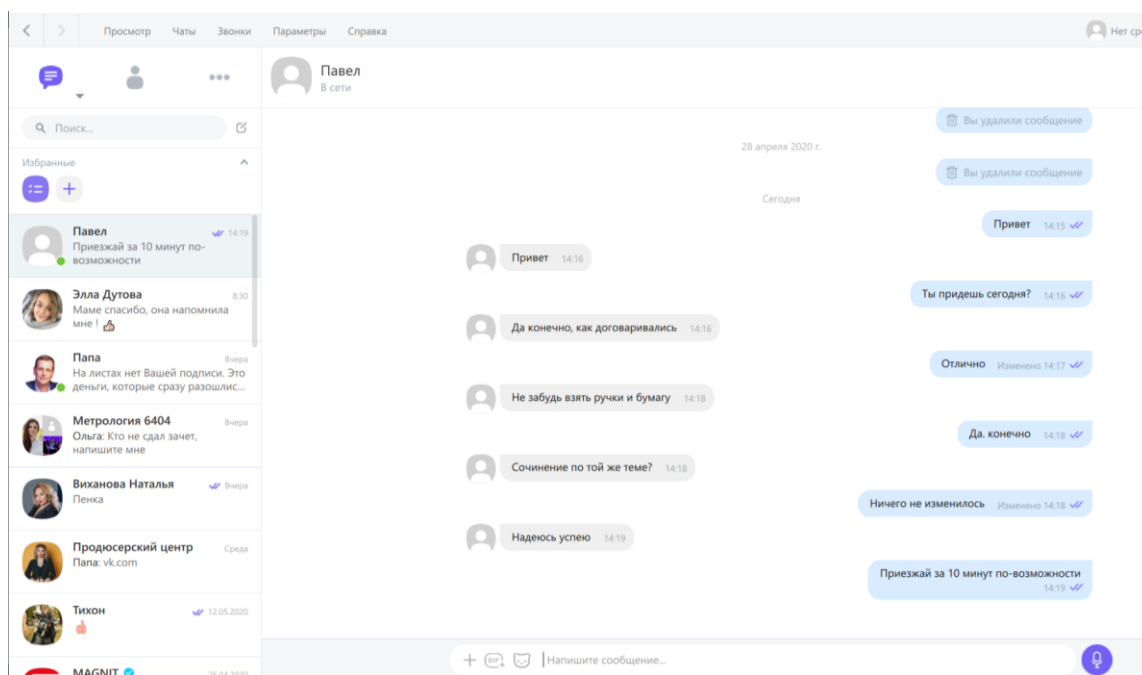


Рисунок 1.1 – Скриншот экрана с программным продуктом «Viber»

«Viber» создан для быстрого и надёжного обмена сообщениями. В рамках данной работы ключевой интерес представляют групповые чаты, в которых доступна пересылка сообщений, упоминания и хештеги — поэтому даже в

больших конференциях общение эффективно и упорядоченно. Так, для каждого воспитательно-образовательного мероприятия возможно создание отдельного чата, в котором пользователи смогут предоставлять обратную связь или задавать вопросы. Однако такой подход снижает вовлеченность участников по причине постоянной смены места обсуждения того или иного вопроса.

«Viber» можно установить на телефон, планшет, компьютер или использовать его онлайн. Поддерживаются устройства на iOS, Android, Windows Phone, Windows и многих других ОС. Необходимо также отметить, что «Viber» является бесплатной для пользователей.

Если «Viber» больше подходит для организации взаимодействия «вопрос-ответ», то сервис «Survio» позволяет создать анкету для обратной связи с профессиональным оформлением и содержанием. Результаты опроса можно просматривать в реальном времени в форме графиков, таблиц, PDF-отчетов и файлов данных в самых распространенных форматах. Скриншот программы представлен на рисунке 1.2.

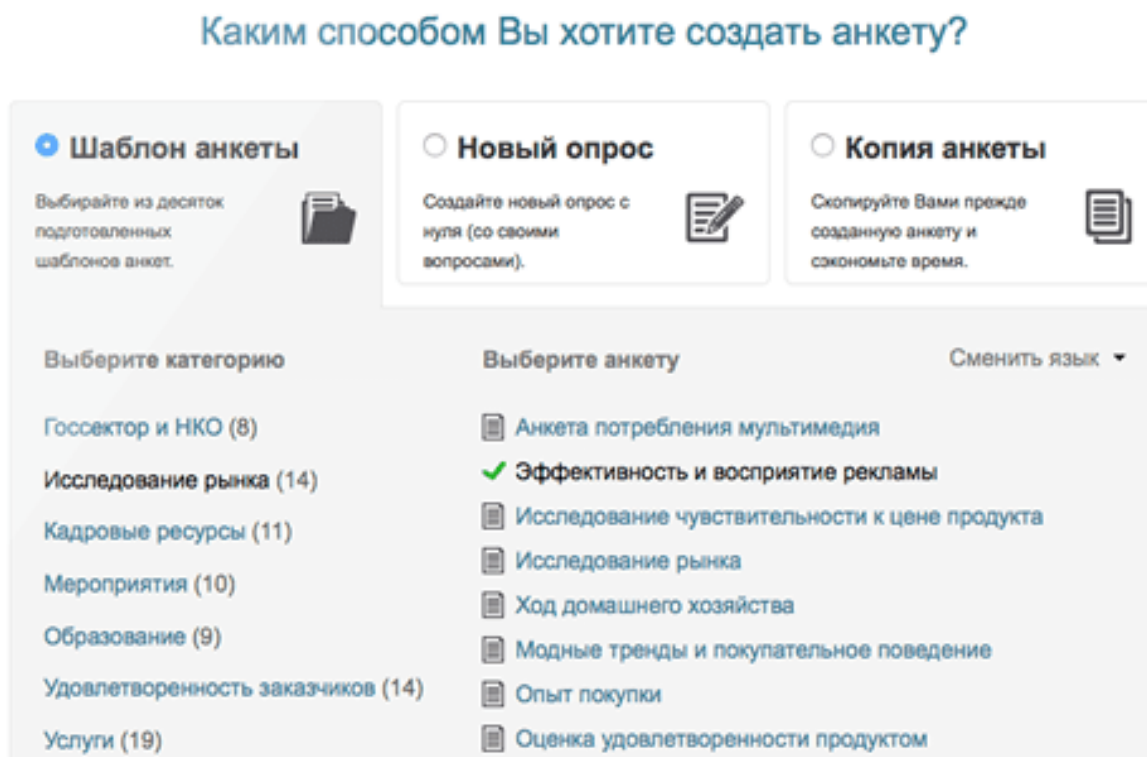


Рисунок 1.2 – экрана с программным продуктом «Survio»

По сути «Survio» - это программное обеспечение для проведения опросов, маркетинговых исследований, оценки удовлетворённости и получения обратной

связи. Система предоставляет более 100 готовых шаблонов для создания форм, а готовым опросом можно поделиться посредством URL-ссылки, в социальных сетях или через email.

Сфера применения возможностей сервиса достаточно обширна, в том числе с его помощью воспитатели могут оптимизировать коммуникационный процесс голосования или обсуждения.

Платформа позволяет копировать анкеты, добавлять любые из 19 типов вопросов, подключать тексты-подсказки и организовывать вопросы в случайном порядке. Софт предоставляет функции email-рассылки опросников, создания ссылки на заполнение, вставки анкеты на сайт в виде кнопки с возможностью изменения размера и цвета. Кроме того, доступна статистика в реальном времени с данными о количестве разосланных приглашений, создание анонимных анкет и импорт контактов из Windows Live, Yahoo и Gmail. Стоимость лицензии составляет 46000 рублей в год.

Подытожим проведенный анализ и представим в таблице 1.2 анализ ключевых особенностей предложенных решений в области совершенствования коммуникационного процесса взаимодействия родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий дошкольного учреждения.

Таблица 1.2 – Анализ ключевых особенностей предложенных решений в области совершенствования коммуникационного процесса взаимодействия родителей и воспитателей

Критерий сравнения	Наименование	
	Viber	Survio
Поддержка коммуникации формата «вопрос-ответ»	+	-
Поддержка групповой работы	+	+
Поддержка анонимности в группе	-	+
Поддержка коммуникации формата “обратная связь”	+	+
Создание отчетов для анализа качества проводимых мероприятий	-	+

Продолжение таблицы 1.2

Поддержка формата для мобильных устройств	+	+
Импорт данных	-	+
Экспорт данных	-	+
Стоимость лицензии, руб. в год	-	46000

Таким образом, предложенные программные продукты не позволяют полноценно решить вопрос совершенствования коммуникационного процесса взаимодействия родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий и домашних заданий дошкольного учреждения. В текущей ситуации наиболее оптимальным решением будет разработка автоматизированной информационной системы, в которой успешно реализованы принципы и особенности коммуникационного процесса взаимодействия родителей и воспитателей.

Вместе с тем, разработка индивидуального программного продукта не принесет дополнительных расходов на приобретение ежегодных лицензионных ключей, будет полностью учитывать все специфические особенности воспитательно-образовательного процесса, а вся информация будет храниться на рабочей станции ДООУ и контролироваться кругом лиц, имеющим к ней доступ [10,11].

1.2 Формирование предложений по автоматизации коммуникационных процессов взаимодействия родителей и воспитателей

В данной работе предлагается разработка программного продукта, предназначенного для совершенствования коммуникационного процесса взаимодействия родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий и в рамках домашних заданий дошкольного учреждения, что позволит повысить эффективность и результативность воспитательного процесса за счет:

1) Повышения осведомленности и вовлеченности родителей в воспитательно–образовательные процессы детских дошкольных учреждений;

2) Улучшения качества взаимодействия родителей и воспитателей через дополнительный канал общения;

3) Получения воспитателем обратной связи для анализа интереса детей/родителей в отношении того или иного мероприятия и домашней работы;

4) Получения руководителями и воспитателями (например, заведующим детским учреждением) обратной связи от родителей для анализа качества проводимых мероприятий и домашней работы в детском дошкольном учреждении;

5) Повышение репутации дошкольного учреждения в глазах родителей.

Таким образом, целью создания автоматизированной информационной системы, обеспечивающей взаимодействия родителей и воспитателей, является улучшение качества воспитательно-образовательного процесса и повышение эффективности работы детского дошкольного учреждения в целом.

Внедрение автоматизированной информационной системы позволит предоставить пользователям следующий функционал:

1) Обеспечение возможности публикации воспитателями информации о проводимых мероприятиях и домашних заданиях;

2) Обеспечение возможности изучения информации о проводимых мероприятиях и домашних заданиях родителями;

3) Организация переписки между родителями и воспитателями с привязкой к теме мероприятия или домашнего задания;

4) Обеспечение возможности оставлять комментарии воспитателю или давать обратную связь;

5) Обеспечение возможности формирования отчетов по ряду показателей.

Основываясь на приведенном в пункте 1.1 анализе и, проанализировав таблицу 1.1, можно сделать вывод о том, что необходимо оптимизировать коммуникационный процесс с целью автоматизации взаимодействия родителей и воспитателей. При оптимизации выявленных процессов, связанных с коммуникациями и взаимодействием родителей и воспитателей, использовались два метода:

1) Метод минимизации устной информации;

Устная информация имеет свойство сильно искажаться при передаче ее от одного участника бизнес-процесса к другому, в связи с чем необходимо сделать так, чтобы в рамках бизнес-процессов все информационные потоки были по возможности документированы.

2) Стандартизация форм сбора и передачи информации.

Повысить эффективность процессов позволяет метод стандартизации форм сбора и передачи информации. На текущий момент отсутствуют типовые формы документов и отчетов для анализа эффективности и результативности того или иного потокового мероприятия или домашнего задания, что усложняет их обработку [2,12,13].

Необходимость разработки и внедрения АИС, обеспечивающей взаимодействие родителей и воспитателей, обусловлена следующим перечнем преимуществ:

1) Узкая специализация разрабатываемой системы позволяет более точно организовать коммуникационный процесс взаимодействия;

2) В условиях возможного изменения принципов организации воспитательно-образовательных мероприятий настройка разрабатываемой системы (в том числе полная переработка/изменение) может производиться быстро и безболезненно для основной деятельности детского дошкольного учреждения, т.к. не затрагиваются централизованные системы.

3) Стоимость разработки и изменения много меньше стоимости ежегодных лицензионных ключей и обслуживания существующих решений [14].

1.3 Разработка требований к проектируемой автоматизированной информационной системе

Разработка АИС, обеспечивающей взаимодействие родителей и воспитателей, в части ввода/вывода данных должна соответствовать следующим функциональным требованиям:

- Верификация вводимых данных;
- Передача данных в модуль хранения.

На уровне взаимодействия с базой данных автоматизированная информационная система должна обеспечивать следующие функции:

- 1) Возможность ввода, изменения и удаления данных о воспитателях, родителях, детях, домашних заданиях и проводимых мероприятиях;
- 2) Поиск данных в базе данных по одному или нескольким параметрам;
- 3) Формирование отчетных форм.

Перед проектированием автоматизированной информационной системы, необходимо определиться с предъявляемыми к ней нефункциональными требованиями. Можно выделить следующие группы:

- Требования к программно-технической среде;
- Требования к информации, циркулирующей в АИС;
- Требования к квалификации персонала;
- Требования к надежности;
- Требования к обеспечению информационной безопасности;
- Требования к сохранности информации при авариях;
- Требования к эргономике и технической эстетике [15,14,16].

Предполагается, что разработка автоматизированной информационной системы будет реализована с использованием платформы ASP.NET MVC, в связи с чем специализированных требований к программно-технической среде предъявляться не будет, т.к. все действия пользователей будут происходить в браузере.

Входные данные, по возможности, имеют свободный формат, но при этом осуществляется проверка на корректность вводимых символов и команд. В случае некорректного ввода, выводится сообщение об ошибке, и просьба проверить вводимые данные.

Необходимо также отметить, что АИС должна обеспечивать возможность исторического хранения данных с глубиной не менее 1 года.

Персонал должен быть обучен работе в соответствии с функциями, выполняемыми в информационной системе в соответствии с предоставленной ролью. Разрабатываемая автоматизированная информационная система,

обеспечивающая взаимодействие родителей и воспитателей, будет доступна трем категориям пользователей. В таблице 1.3 приведены роли пользователей автоматизированной информационной системы.

В случае сбоя АИС, необходимо, чтобы она могла восстановить работоспособное состояние в небольшой промежуток времени, обеспечить сохранность и целостность данных, с которыми непосредственно работает система, а также достоверность и целостность данных, обрабатываемых во время обнаружения сбоя. Ошибки системы могут исправляться только разработчиком или администратором, обладающим требуемыми полномочиями. В случае обнаружения ошибки работа системы должна быть приостановлена для устранения дефекта. При возникновении сбоев работы программных или технических средств необходимо обеспечить достоверность данных, оставшихся после сбоя.

Устройство хранения данных должно быть защищено от внешних физических воздействий. Специализированного обслуживания технических средств системы не требуется.

Таблица 1.3 – Роли пользователей информационной системы

Пользователь	Роль пользователя	Задачи пользователя и доступность вкладок
Системный администратор	Администратор	Доступен полный функционал. Данная роль позволяет проводить обслуживание и настройку АИС, обеспечивать ее работоспособность и заводить пользователей в системе с ролью Администратор и Воспитатель.
Воспитатель	Воспитатель	Данная роль позволяет реализовывать информационный процесс взаимодействия с родителями, выгрузку отчетов, работа со списком групп и детей (добавление, изменение и удаление), а также заводить, редактировать и удалять пользователей в системе с ролью Родитель.

Продолжение таблицы 1.3

Родитель	Родитель	Доступен функционал по ознакомлению с общей информацией по проводимым мероприятиям, домашним заданиям, воспитателям, группам, а также формированию вопросов и получению ответов от пользователей с ролью Воспитатель и Родитель.
----------	----------	--

Требования к обеспечению информационной безопасности можно определить следующим перечнем:

1) Целостность – информация, на основе которой принимаются решения, должна быть достоверной и точной, защищенной от возможных непреднамеренных и злоумышленных искажений;

2) Доступность (готовность) – информация и соответствующие автоматизированные службы должны быть доступны, готовы к работе всегда, когда в них возникает необходимость;

3) Конфиденциальность – информация должна быть доступна только тому, кому она предназначена. Меры по обеспечению безопасности в АИС должны быть направлены на нейтрализацию актуальных угроз безопасности данных, связанных с получением несанкционированного доступа к информации.

Требования по сохранности информации включают перечень событий, при которых должна быть обеспечена сохранность информации в автоматизированной информационной системе:

1) Ошибки персонала в процессе эксплуатации автоматизированной информационной системы;

2) Искажения в каналах телекоммуникации информации, поступающей от внешних источников и передаваемой потребителям, а также недопустимые изменения характеристик потоков информации;

3) Сбои и отказы аппаратуры;

4) Изменения конфигурации автоматизированной информационной системы за пределы, проверенные при испытаниях.

Разрабатываемая АИС должна иметь удобный и простой интерфейс. Поля для ввода и отображения информации должны иметь четкое, понятное название, и расположены оптимальным образом для удобства работы с системой.

Требования, предъявляемые к эргономике и технической эстетике АИС, можно сформулировать следующим образом:

- 1) Удобство использования системы (user–friendly);
- 2) Наглядное отображение выходных данных и подсказок при возникновении ошибок;
- 3) Простота интерфейса.

1.4 Обоснование выбора инструментов разработки автоматизированной информационной системы

1.4.1 Обоснование выбора системы управления базой данных

Согласно поставленным задачам, можно сделать вывод о необходимости оперировать большим количеством информации. Следовательно, разрабатываемая АИС должна содержать базу данных, которая позволит хранить, резервировать и ограничивать права доступа к данной информации.

Для сравнения выберем такие СУБД, как: MS Access, MySQL, Oracle и MS SQL Server. Сравнительный анализ СУБД представлен в таблице А.1 Приложения А.

По результатам анализа таблицы можно сделать вывод, что в настоящее время на рынке представлен огромный выбор СУБД и каждая СУБД предназначена для решения определенного круга задач, которые она может выполнять [17].

Для реализации возможности накопления и редактирования информации в АИС, будет использован Microsoft SQL Server 2014. На рисунке 1.3 представлено окно администрирования базы данных MS SQL Server [18].

На сегодняшний день MS SQL Server – наиболее популярная СУБД, используемая для разработки настольных баз данных. MS SQL Server успешно применяется для создания многопользовательских приложений, где файлы базы данных являются разделяемыми ресурсами в сети. Кроме того, в MS SQL Server

реализована надёжная система защиты от несанкционированного доступа к файлам. Востребованность СУБД MS SQL Server обусловлена следующими причинами:

- 1) Система полностью русифицирована;
- 2) Благодаря идеологии Windows можно представить информацию наглядно и красочно;
- 3) Широкое и наглядное представление справочной системы;
- 4) Работает на различных платформах;
- 5) Полностью многопоточная;
- 6) Большое количество типов данных;
- 7) MS SQL Server может управлять очень большими базами данных [18].

Таким образом, использование данного программного обеспечения обусловлено наличием множества интегрированных служб, которые помогают расширить возможности применения разнообразной информации, например, осуществить поиск, выполнять синхронизацию и анализ, составлять запросы.

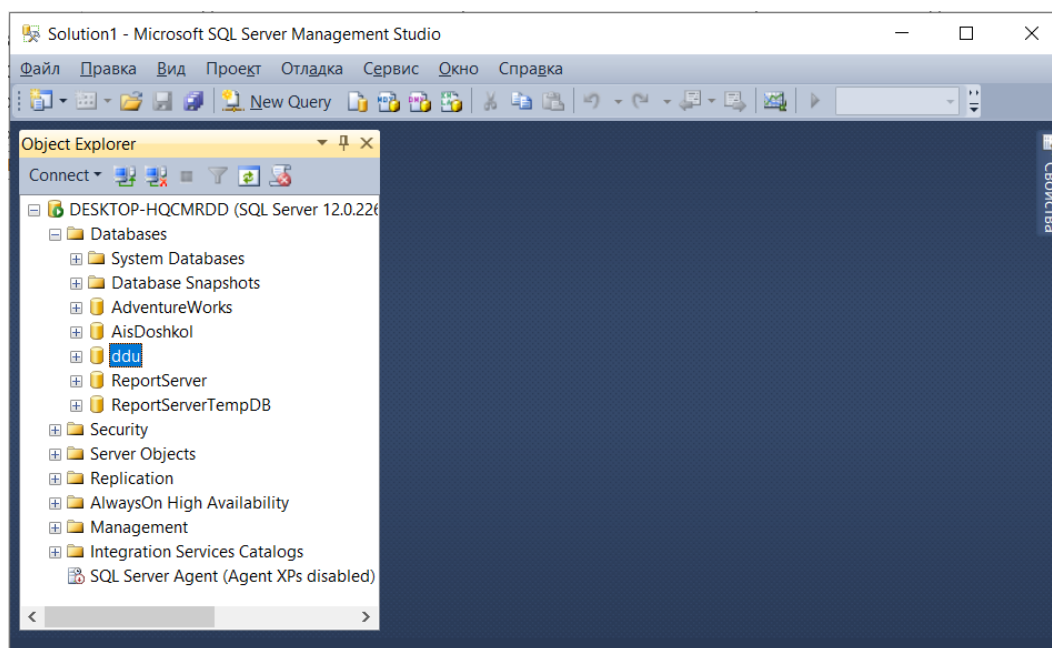


Рисунок 1.3 – Окно администрирования баз данных MS SQL Server 2014

1.4.2 Обоснование выбора среды разработки

Для выбора средства разработки АИС проведем сравнительный анализ следующих сред разработки:

- 1) Microsoft Visual Studio;

- 2) C++ Builder;
- 3) Borland Delphi [15].

Сравним характеристики языков программирования, используемых в описанных выше средах разработки. Результаты сравнения представим в таблице 1.4.

Таблица 1.4 – Сравнение характеристик языков программирования

	C	C++	C#	Delphi
Функциональный	-	+/-	+/-	+/-
Обобщенное программирование	-	+	+	+
Возможность компиляции	+	+	+	+
Интерпретатор командной строки	-/+	+/-	-	-
Ручное управления памятью	+	+	+	+
Поддержка try/catch	-	+	+	+
Алгебраические типы данных	-	-	-	-/+
Многомерные массивы	+	+	+	+
Целые числа с контролем границ	-	-	-	+
Интерфейсы	-	+	+	+
Макросы	-/+	-/+	-	-

В качестве средства реализации автоматизированной информационной системы, обеспечивающей взаимодействие родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий дошкольного учреждения, был выбран язык программирования C#, который разработан в 1998 - 2001 годах группой инженеров под руководством Андерса Хейлсберга в компании Microsoft как язык разработки приложений для платформы Microsoft .NET Framework [19].

C# является объектно-ориентированным языком и в этом плане много перенял у Java и C++. Использование C# в целях разработки автоматизированной информационной системы, обеспечивающей взаимодействие родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных

мероприятий дошкольного учреждения, обусловлен тремя важными характеристиками: простотой, эффективностью и гибкостью.

Перечислим принципиально важные характеристики, которые присущи языку программирования C#:

- Компонентно-ориентированный подход к программированию (который характерен и для идеологии Microsoft .NET в целом);
- Унифицированная система типизации (соответствует идеологии Microsoft .NET в целом);
- Индексаторы (indexer - операторы индекса для обращения к элементам класса - контейнера);
- Оператор foreach (обработка всех элементов классов-коллекций, аналог Visual Basic);
- Механизмы boxing и unboxing для преобразования типов;
- Атрибуты (средство оперирования метаданными в СОМ-модели) [19].

Главным фактором языка C# является практичность. C# предоставляет программисту средства для быстрого и эффективного решения поставленных задач и построению крупных, но в тоже время гибких, масштабируемых и расширяемых приложений.

Разработка программного обеспечения будет проводиться при помощи Visual Studio 2019. Использование Microsoft Visual Studio обусловлено удобством при написании кода, мощным функционалом для отладки, большим количеством дополнительных компонентов, упрощающих процесс написания ПО. В то же время, Microsoft Visual Studio 2019 является популярной средой разработки и используется во многих компаниях, что также послужило причиной выбора данной среды разработки [19].

1.5 Обоснование выбора архитектуры автоматизированной информационной системы

В связи с тем, что предложенный вариант организации взаимодействия родителей и воспитателей реализуется посредством использования территориально распределенных устройств обработки данных,

автоматизированная информационная система должна быть разработана как «клиент-серверное» приложение.

«Клиент-сервер» – сетевая архитектура, в которой нагрузка распределена между поставщиками услуг, называемых серверами, и заказчиками услуг, называемых клиентами. Использование архитектуры клиент-сервер позволяет создавать надежные (в смысле целостности данных) многопользовательские информационные системы с централизованной базой данных, независимые от аппаратной (а часто и программной) части сервера базы данных и поддерживающие графический интерфейс пользователя на клиентских станциях. Ключевые особенности использования архитектуры «Клиент-сервер» можно определить следующим перечнем:

- Клиентская программа работает с данными через запросы к серверному программному обеспечению;

- Базовые функции разделены между клиентом и сервером [2,14].

Преимущества использования архитектуры «Клиент-сервер» можно определить следующим перечнем:

- 1) Возможность распределения функции вычислительной системы между несколькими независимыми компьютерами в сети;

- 2) Безопасное хранение данных на сервере, имеющем более лучшую степень защиты по сравнению с клиентом;

- 3) Полная поддержка многопользовательской работы;

- 4) Гарантия целостности данных [2,14].

Необходимо также отметить, что АИС будет разработана с использованием платформы ASP.NET MVC, которая представляет собой фреймворк для создания сайтов и веб-приложений с помощью реализации паттерна MVC. Если предложенный проект покажет свою эффективность на практике, разработанный прототип информационной системы взаимодействия родителей и воспитателей может быть доработан в части оптимизации шаблона для различных мобильных устройств [19].

1.6 Постановка задачи исследования

Таким образом, в данном разделе проведено исследование коммуникационного процесса взаимодействия родителей и воспитателей. Был сделан вывод о том, что с целью повышения эффективности воспитательно-образовательного процесса в целом и коммуникационного процесса взаимодействия родителей и воспитателей в частности необходимо разработать и внедрить автоматизированную информационную систему, способную оказать поддержку родителям и педагогическому составу.

Вместе с тем, было сформировано предложение по разработке индивидуального программного продукта, которое будет полностью учитывать все требования нормативно-правовой документации и специфические особенности коммуникационного процесса взаимодействия родителей и воспитателей. Наряду с функциональным анализом коммуникационного процесса также был проведен выбор инструментов разработки автоматизированной информационной системы.

Исходя из полученных результатов, полученных в данном разделе, можно сделать вывод о том, что в дальнейшей работе следует решить следующие задачи:

- 1) Спроектировать базу данных и автоматизированную информационную систему, обеспечивающую взаимодействие родителей и воспитателей детского дошкольного учреждения;
- 2) Разработать автоматизированную информационную систему, обеспечивающую взаимодействие родителей и воспитателей детского дошкольного учреждения;
- 3) Разработать пользовательскую документацию по порядку использования автоматизированной информационной системы, обеспечивающей взаимодействие родителей и воспитателей детского дошкольного учреждения.

2 Проектирование автоматизированной информационной системы для детских дошкольных учреждений

2.1 Функциональная структура проектируемой автоматизированной информационной системы

Автоматизированная информационная система, обеспечивающая взаимодействие родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий дошкольного учреждения, состоит из четырех модулей:

- 1) Работа с базой данных;
- 2) Работа со справочными данными;
- 3) Организация обратной связи;
- 4) Анализ эффективности и результативности проведения потоковых мероприятий и домашних заданий посредством формирования отчетов.

Модуль работы с базой данных предусматривает ввод, хранение, редактирование и удаление данных о воспитателях, родителях, потоковых мероприятиях и переписке воспитателей и родителей. Результатом работы является база данных, которая используется во всех отдельных модулях.

Модуль работы со справочными данными использует базу данных для получения связей и дополнения существующих записей. В последствие такие данные также заносятся в базу данных и используются для формирования отчетов.

Модуль организации обратной связи использует базу данных для получения уже внесенной пользователями информации по переписке воспитателей и родителей. В результате на выходе мы получаем организацию процесса оперативной поддержки родителей по тематике потоковых воспитательно-образовательных мероприятий дошкольного учреждения.

Модуль анализа эффективности проведения потоковых мероприятий, домашних заданий и формирования отчетов использует базу данных с целью предоставления наглядной информации по числу возникающих вопросов и

проблем в разрезе тематик воспитательно-образовательных мероприятий и домашних заданий дошкольного учреждения.

2.2 Проектирование базы данных автоматизированной информационной системы

2.2.1 Логическое проектирование

В основе модели базы данных лежат такие основные понятия, как сущность, атрибут и связь. Сущности моделируют класс однотипных объектов, в системе существует несколько экземпляров каждой сущности. У сущности есть набор атрибутов – характеристик, которые определяют свойства представителя класса. Первичным ключом называют набор атрибутов, которые однозначно идентифицируют экземпляр сущности. Между разными сущностями могут быть определены связи, они показывают, как сущности взаимодействуют между собой [17].

База данных автоматизированной информационной системы, обеспечивающей взаимодействие родителей и воспитателей в рамках реализации потоковых воспитательно-образовательных мероприятий дошкольного учреждения, должна выполнять следующие функции:

- Хранение и обработка информации о воспитателях;
- Хранение и просмотр информации о родителях;
- Хранение и просмотр информации о детях;
- Хранение и просмотр информации о группах;
- Хранение и просмотр информации о домашних заданиях;
- Хранение и просмотр информации о проводимых мероприятиях;
- Хранение и просмотр информации о переписке между воспитателем и родителями (сообщениях в чатах).

Сформируем и проанализируем сущности базы данных АИС при помощи таблицы 2.1.

На этапе логического проектирования базы данных необходимо рассмотреть каждую связь между сущностями. Одно из основных требований к организации базы данных – это обеспечение возможности определения одних

сущностей по значениям других. Установим в проектируемой базе данных связи посредством модели «сущность-связь» [17].

Таблица 2.1 – Сущности и атрибуты базы данных АИС

Сущность	Назначение	Атрибуты
Мероприятие	Необходима для хранения и просмотра общих сведений о мероприятиях дошкольного учреждения	ID мероприятия, название, описание, дата проведения, количество лайков, количество дизлайков, ID типа мероприятия
Тип мероприятия	Содержит информацию о типах мероприятия	ID типа мероприятия, тип
Сообщение	Необходима для хранения и просмотра сведений о сообщениях в чатах	ID сообщения, текст, дата и время публикации, флаг «Аноним», ID типа сообщения, ID пользователя, ID мероприятия, ID домашнего задания
Тип сообщения	Содержит информацию о типах сообщения	ID типа сообщения, тип
Пользователь	Содержит информацию о пользователях, использующих автоматизированную информационную систему	ID пользователя, ФИО, E-mail, телефон, дата рождения, ID учетной записи
Тип учетной записи	Содержит информацию о типах учетной записи	ID типа учетной записи, тип
Учетная запись	Содержит информацию об учетных записях, для авторизации в системе	ID учетной записи, логин, пароль, ID типа учетной записи
Педагог	Содержит информацию о педагогах	ID пользователя, портфолио, стаж
Родитель	Содержит информацию о родителях	ID пользователя, флаг «Многодетная семья»
Ребенок	Содержит информацию о детях, учащихся в ДДУ	ID ребенка, ФИО, пол, дата рождения, ID матери, ID отца
Дети в группе	Содержит информацию о детях, состоящих в группах	ID ребенка, ID группы

Продолжение таблицы 2.1

Группа	Содержит информацию об обучающих группах	ID группы, название, время проведения занятий, описание, ID педагога
Домашнее задание	Содержит информацию о домашних заданиях	ID домашнего задания, название, описание, дата, количество лайков, количество дизлайков, ID типа домашнего задания
Тип домашнего задания	Содержит информацию о типах домашнего задания	ID типа домашнего задания, тип
Домашнее задание для группы	Содержит информацию о домашних заданиях для групп	ID домашнего задания, ID группы
Участники мероприятия	Содержит информацию о группах, участвующих в мероприятии	ID группы, ID мероприятия
Рейтинг мероприятия	Содержит информацию о проставленных лайках/дизлайках мероприятию от пользователей	ID мероприятия, ID пользователя, тип обратной связи(лайк/дизлайк)
Рейтинг домашнего задания	Содержит информацию о проставленных лайках/дизлайках домашнему заданию от пользователей	ID домашнего задания, ID пользователя, тип обратной связи(лайк/дизлайк)

Связь – это некоторая ассоциация между двумя сущностями. Связи позволяют по одной сущности находить другие, связанные с ней. Выберем ключевые связи по разработанным сущностям и отобразим их в таблице 2.2.

Таблица 2.2 – Взаимосвязь сущностей модели базы данных АИС

Сущность 1	Сущность 2	Описание
Мероприятие	Сообщение	Для каждого мероприятия существует множество сообщений, а для каждого сообщения может быть определен только один чат в рамках конкретного мероприятия

Продолжение таблицы 2.2

Мероприятие	Тип мероприятия	Для каждого мероприятия существует один тип, а для каждого типа множество мероприятий
Сообщение	Тип сообщения	Для каждого сообщения существует один тип, а для каждого типа множество сообщений
Тип учетной записи	Учетная запись	Для каждой учетной записи существует один тип, а для каждого типа множество учетных записей
Учетная запись	Пользователь	Для каждой учетной записи существует один пользователь, а для каждого пользователя одна учетная запись
Пользователь	Педагог	Для каждого пользователя существует один педагог, а для каждого педагога один пользователь
Пользователь	Родитель	Для каждого пользователя существует один родитель, а для каждого родителя один пользователь
Родитель	Ребенок	Роль – Мать. Для каждой матери существует множество детей, а для каждого ребенка одна мать
Родитель	Ребенок	Роль – Отец. Для каждого отца существует множество детей, а для каждого ребенка один отец
Ребенок	Группа	Каждый ребенок может состоять во множестве групп, а каждая группа может содержать множество детей. Для обеспечения связи «многие-ко-многим» используется сущность «Дети в группе»
Группа	Домашнее задание	Каждая группа может иметь множество домашних заданий, а каждое домашнее задание может быть задано для множества групп. Для обеспечения связи «многие-ко-многим» используется сущность «Домашнее задание для группы»

Продолжение таблицы 2.2

Группа	Мероприятие	Каждая группа может принимать участие во множестве мероприятий, а каждое мероприятие может быть предназначено для множества групп. Для обеспечения связи «многие-ко-многим» используется сущность «Участники мероприятия»
Домашнее задание	Тип домашнего задания	Для каждого домашнего задания существует один тип, а для каждого типа множество домашних заданий
Домашнее задание	Педагог	Каждое домашнее задание создает один педагог, а каждый педагог задает множество домашних заданий
Группа	Педагог	Каждая группа имеет одного педагога, а каждый педагог может обучать множество групп
Пользователь	Сообщение	Каждый пользователь может отправить множество сообщений, а каждое сообщение может быть отправлено одним пользователем
Сообщение	Домашнее задание	Для каждого домашнего задания существует множество сообщений, а для каждого сообщения может быть определен только один чат в рамках конкретного домашнего задания
Пользователь	Домашнее задание	Каждый пользователь может оценить множество домашних заданий, а каждое домашнее задание может быть оценено множеством пользователей. Для обеспечения связи «многие-ко-многим» используется сущность «Рейтинг домашнего задания»
Пользователь	Мероприятие	Каждый пользователь может оценить множество мероприятий, а каждое мероприятие может быть оценено множеством пользователей. Для обеспечения связи «многие-ко-многим» используется сущность «Рейтинг мероприятия»

Продолжение таблицы 2.2

Участники мероприятия	Содержит информацию о группах, участвующих в мероприятии	ID группы, ID мероприятия
Рейтинг мероприятия	Содержит информацию о предоставленных лайках/дизлайках мероприятию от пользователей	ID мероприятия, ID пользователя, тип обратной связи(лайк/дизлайк)
Рейтинг домашнего задания	Содержит информацию о предоставленных лайках/дизлайках домашнему заданию от пользователей	ID домашнего задания, ID пользователя, тип обратной связи(лайк/дизлайк)

Целью данного этапа проектирования является отображение концептуальной инфологической модели на логическую реляционную модель. Это производится путем совместного представления в отношениях ключевых атрибутов взаимосвязанных сущностей.

На базе изученной информации, была разработана модель базы данных АИС. Данная модель не только отражает структуру данных, циркулирующих при проведении воспитательно-образовательных мероприятий, но и позволяет наглядно конкретизировать сущности, атрибуты и связи. Разработанная логическая модель базы данных АИС представлена на рисунке 2.1.

2.2.2 Физическое проектирование

С целью дальнейшего проектирования базы данных разработаем физическую модель базы данных, содержанием которой является определение модели данных. Модель данных - это набор соглашений по способам представления сущностей, связей, агрегатов, системы классификации. Каждая

модель данных определяет особенности выполнения основных операций над данными:

- Добавление;
- Удаление;
- Модификация;
- Выборка.

Результатом физического проектирования является описание логической структуры базы данных на языке описания. На этапе разработки физической модели определяется состав базы данных, например, принимается решение о хранении только исходных данных, а все производные могут быть получены расчетным путем в результате запроса [17].

Таким образом, особое внимание при построении модели уделяют целостности и отсутствию избыточности данных. Избыточность – это многократное повторение одних и тех же данных. В таблице 1 Приложения Б представлены атрибуты базы данных АИС для каждой сущности с описанием и типом данных.

2.3 Проектирование автоматизированной информационной системы

С целью проведения объектно-ориентированного проектирования АИС, была использована нотация UML. Язык UML представляет собой графический язык моделирования общего назначения, предназначенный для спецификации, визуализации, проектирования и документирования всех артефактов, создаваемых при разработке программных систем. При проектировании программного продукта были сформированы следующие UML -диаграммы:

- 1) Диаграмма прецедентов;
- 2) Диаграмма последовательности;
- 3) Диаграмма состояний.

2.3.1 Диаграмма прецедентов

В первую очередь необходимо спроектировать диаграмму прецедентов, отражающую отношения между актёрами и прецедентами. Прецеденты – это технология определения функциональных требований к системе. Работа

прецедентов заключается в описании типичных взаимодействий между пользователями системы и самой системой и предоставлении описания процесса ее функционирования. Диаграмма прецедентов ИС представлена на рисунке 1 Приложения В.

2.3.2 Диаграмма последовательности

Диаграмма последовательности отражает временные особенности передачи сообщений между частями программы, а также показывает время жизни классов, которые представлены в диаграмме классов. Одним из основных принципов объектно-ориентированного программирования является способ информационного обмена между элементами системы, выражающийся в отправке и получении сообщений друг от друга. Таким образом, основные понятия диаграммы последовательности связаны с понятием объект и сообщение.

На рисунке 2.2 – 2.4 изображены диаграммы последовательности объектов АИС в условиях реализации ключевых действий.

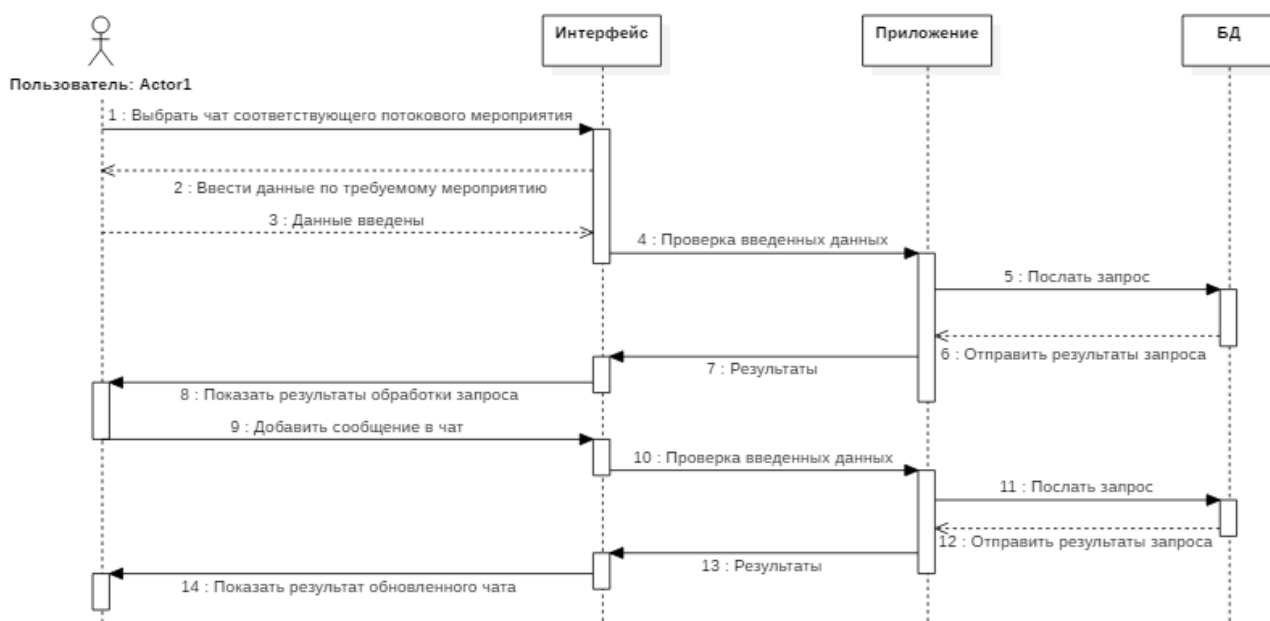


Рисунок 2.2 – Диаграмма последовательности добавления сообщения в чат потокового мероприятия



Рисунок 2.3 – Диаграмма последовательности добавления пользователя

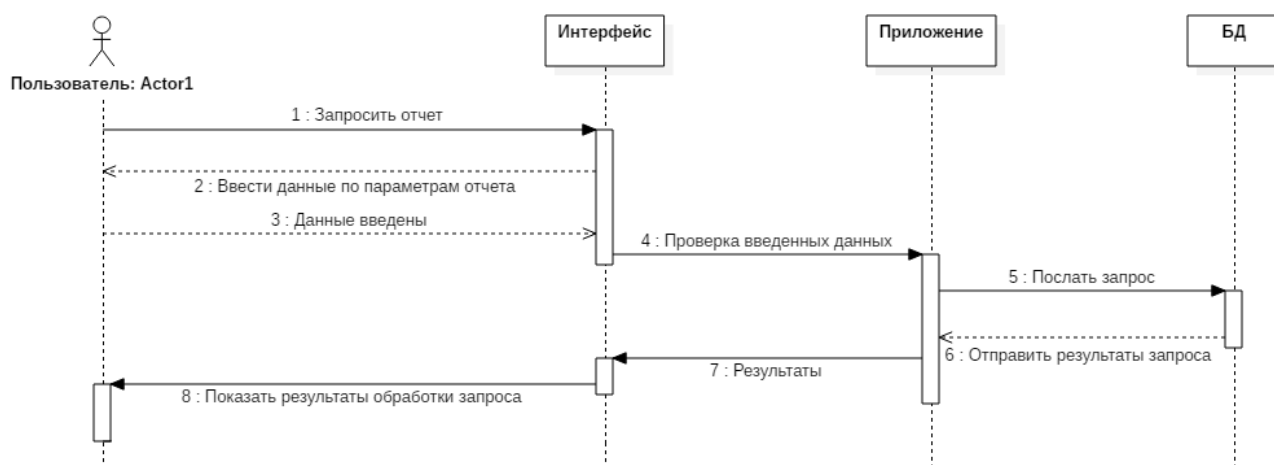


Рисунок 2.4 – Диаграмма последовательности формирования отчета

2.3.3 Диаграмма состояний

Диаграмма состояний показывает работу системы в динамике, характеризует жизненный цикл, который начинается при её запуске. Диаграмма представляет собой состояния, соединенные переходами.

Диаграммы состояний определяют все возможные состояния, в которых может находиться конкретный объект, а также процесс смены состояний объекта в результате наступления некоторых событий. В большинстве объектно-ориентированных методов диаграммы состояний строятся для единственного класса и отражают динамику поведения единственного объекта. На рисунке 2.5 представлена диаграмма состояний класса переписки между родителями и воспитателем.

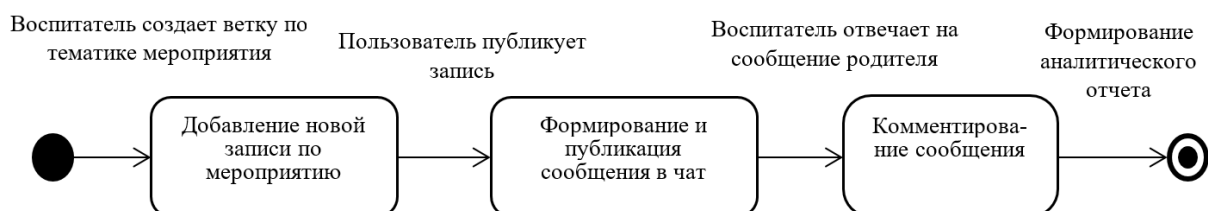


Рисунок 2.5 – Диаграмма состояний

2.4 Проектирование структуры автоматизированной информационной системы

Автоматизированная информационная система, обеспечивающая взаимодействие родителей и воспитателей, будет состоять из следующих модулей:

- 1) Пользователи;
- 2) Дети;
- 3) Группы;
- 4) Домашние задания;
- 5) Мероприятия;
- 6) Чат;
- 7) Отчеты;
- 8) Личный кабинет;
- 9) Настройка типов (Домашнее задание, сообщение, мероприятие).

Структура программного обеспечения представлена на рисунке 2.6, а описание модулей представлена в таблице 2.3.

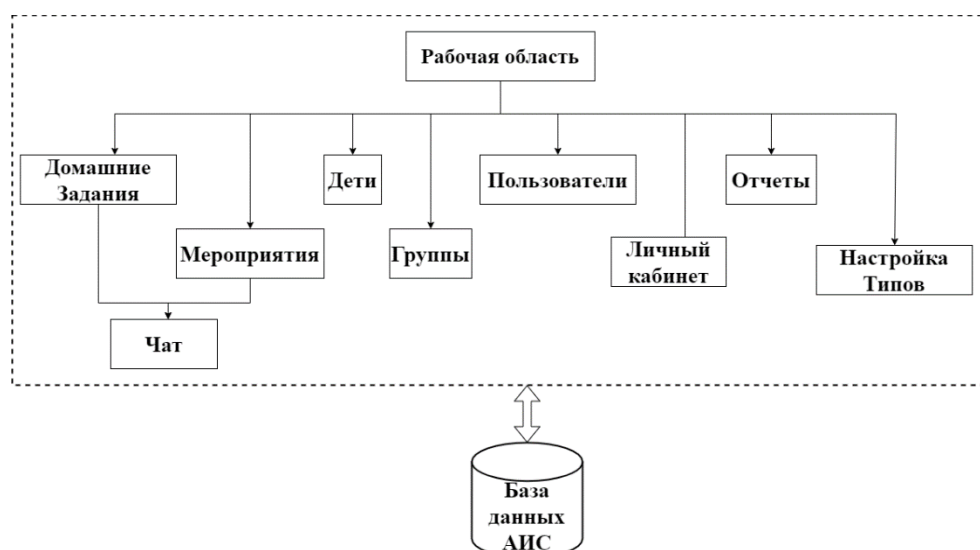


Рисунок 2.6 – Структура программного обеспечения

Таблица 2.3 – Описание модулей автоматизированной информационной системы взаимодействия родителей и воспитателей ДОУ

Наименование	Функциональное назначение	Входные данные	Выходные данные
Рабочая область	Навигация по автоматизированной информационной системе	Информация по мероприятиям, пользователям, группам, детям, домашним заданиям	Справочная информация по мероприятиям, пользователям, группам, детям, домашним заданиям
Группы	Управление информацией о группах: просмотр, добавление, удаление и изменение	Информация о детях	Таблица групп
Личный кабинет	Управление информацией о личных данных: просмотр и изменение	Информация о пользователе	Актуальная информация о пользователе
Настройка типов	Управление информацией о типах мероприятий, домашних заданий и сообщений: просмотр, добавление, удаление и изменение	Информация о типах	Таблица типов
Мероприятия	Управление информацией по мероприятиям: просмотр, добавление, удаление и изменение	Информация по мероприятиям	Таблица мероприятий

Продолжение таблицы 2.3

Чат	Обратная связь и взаимодействие формата «вопрос-ответ»	Информация по переписке родителей с воспитателем	Список сообщений для чата
Пользователи	Работа с пользователями АИС: просмотр, добавление, удаление и изменение	Информация по родителям и воспитателям, использующим систему	Таблица пользователей
Отчеты	Работа с отчетами	Статистическая информация, полученная из переписки родителей с воспитателем	Отчеты в различных разрезах
Домашние задания	Управление информацией по домашним заданиям: просмотр, добавление, удаление и изменение	Информация по домашним заданиям	Таблица домашних заданий
Дети	Управление информацией о детях: просмотр, добавление, удаление и изменение	Информация о детях	Таблица детей, обучающихся в ДДУ

2.5 Основные алгоритмы, используемые в автоматизированной информационной системе

При входе в систему пользователю необходимо пройти авторизацию для доступа к данным. Пользователь должен ввести логин и пароль, и при условии, что такой пользователь есть в системе, ему отобразится главная страница. В случае неправильного ввода логина или пароля система выдаст ошибку и попросит ввести данные еще раз. На рисунке 2.7 представлена блок-схема

алгоритма авторизации пользователя при входе в автоматизированную информационную систему взаимодействия родителей и воспитателей ДОУ.

После авторизации пользователю будет отображена начальная страница с выбором действий. При переходе на страницу проводимых мероприятий ему представится выбор из всех событий. При выборе определенного мероприятия данные будут отправлены на сервер, и если данное мероприятие существует, то пользователю отобразится страница с подробной информацией о выбранном мероприятии. В случае отсутствия в базе данных информации о выбранном мероприятии, будет выдана ошибка на клиенте о том, что данное мероприятие больше не существует. На рисунке 2.8 представлена блок-схема алгоритма выбора определенного мероприятия.

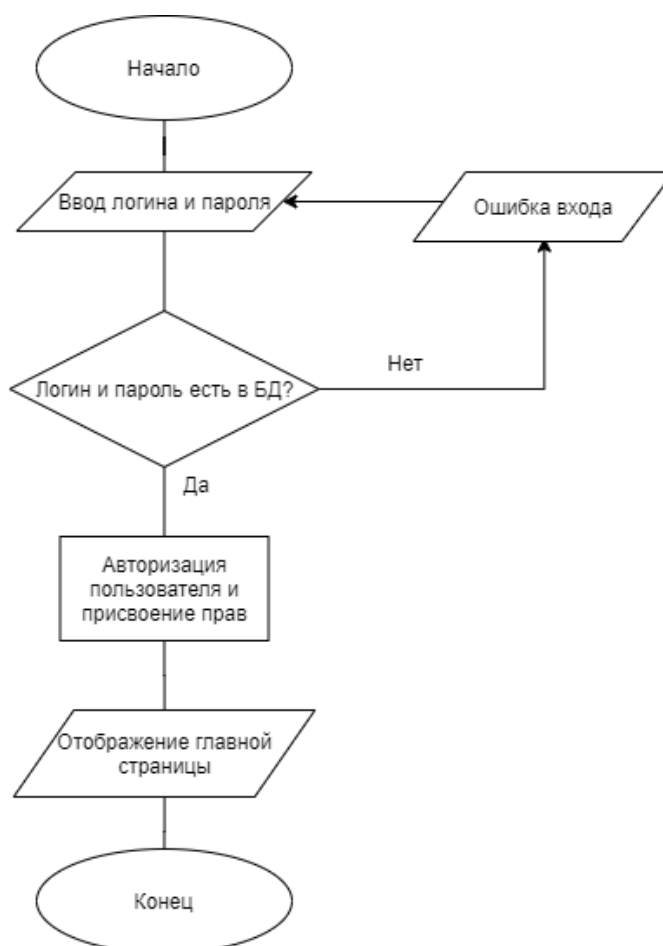


Рисунок 2.7 – Блок-схема алгоритма авторизации пользователя

Если пользователь перейдет в чат выбранного мероприятия, то перед ним отобразится окно чата и окно ввода нового сообщения, с возможностью выбора категории сообщения. После того как он введет сообщение и выберет его тип,

данные отправятся на сервер, и если они валидны, то сохранятся в базе данных. В случае некорректности данных, будет выведено сообщение об ошибке и просьба ввести данные повторно. После работы с чатом пользователь может перейти на другую вкладку или завершить работу с системой. На рисунке 2.9 представлена блок-схема алгоритма работы с чатом.

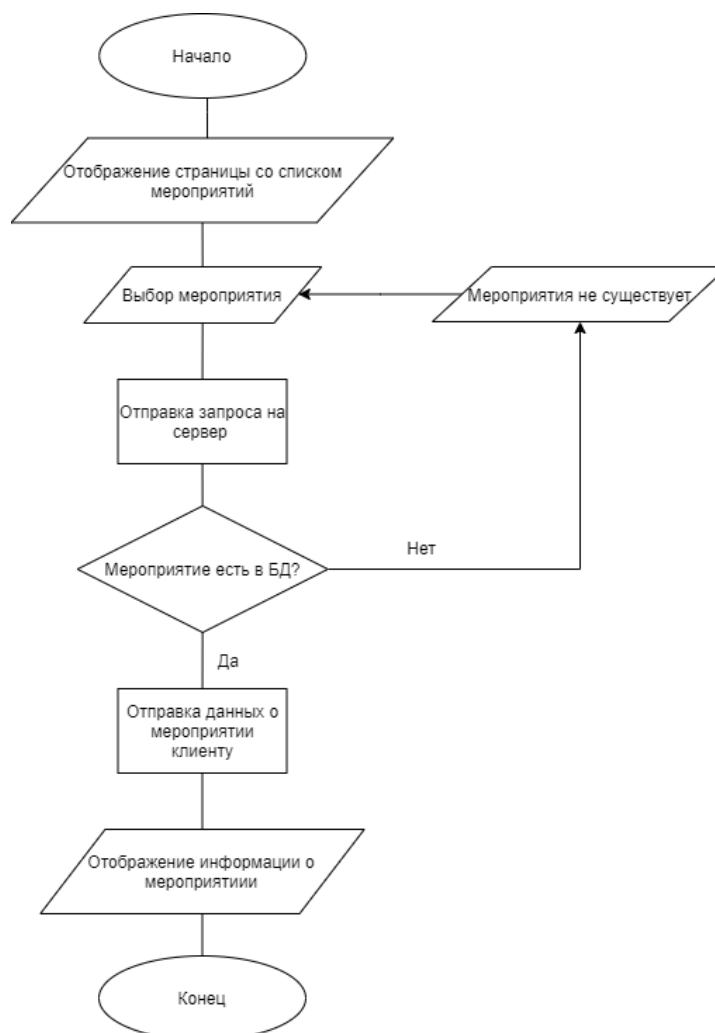


Рисунок 2.8 – Блок-схема алгоритма выбора определенного мероприятия

2.6 Выводы по главе

Таким образом в данном разделе проведено функциональное моделирование разрабатываемой АИС, и объектно-ориентированное проектирование с использованием нотации UML путем разработки диаграммы прецедентов, диаграммы последовательности, а также диаграммы состояний. Были спроектированы логическая и физическая модели базы данных, а также описан состав модулей разрабатываемого программного обеспечения в соответствие со спецификацией, представленной в первом разделе.

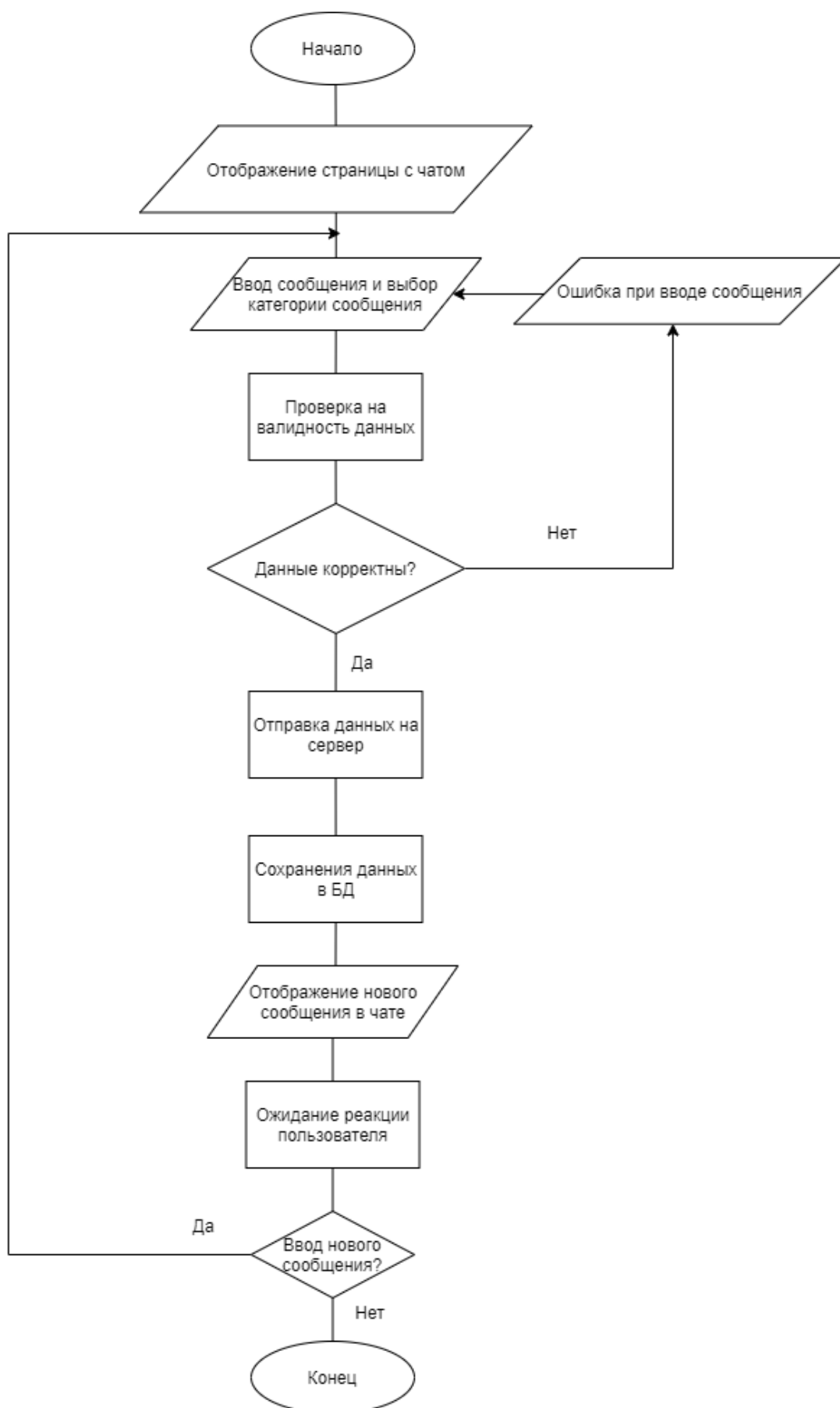
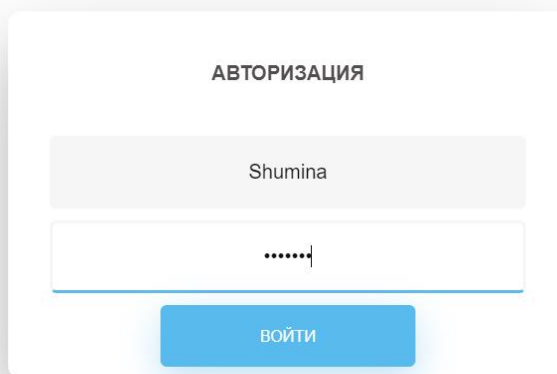


Рисунок 2.9 – Блок-схема алгоритма работы с чатом

3 Разработка автоматизированной информационной системы для детских дошкольных учреждений

3.1 Контрольный пример

Пусть контрольный пример будет заключаться в задаче родителя с логином «Shumina» по публикации предложения в рамках запланированного воспитательно-образовательного мероприятия по направлению «Выезд в зоопарк». В первую очередь пользователю необходимо ввести данные для авторизации в соответствующие поля АИС (рисунок 3.1).



The image shows a login form with the title "АВТОРИЗАЦИЯ" at the top. There are two input fields. The first field contains the text "Shumina". The second field contains masked characters ".....". Below the input fields is a blue button with the text "ВОЙТИ".

Рисунок 3.1 – Авторизация с логином «Shumina»

Родитель не знает точного названия мероприятия, но знает ориентировочный период, в который должно состояться мероприятие. Для того чтобы найти мероприятие родитель вбивает диапазон дат (рисунок 3.2).

В результате АИС предлагает родителю искомое мероприятие (рисунок 3.3).

Родитель просматривает информацию по данному мероприятию при помощи кнопки «Show», после чего переходит на страницу просмотра подробной информации о мероприятии (рисунок 3.4).

Группы Домашние задания Мероприятия Педагоги
Shumina Выйти

Мероприятия

Дата мероприятия от 01/05/2020 До 31/05/2020
Количество лайков от До
Группа:

Раннее развитие
Английский 1 группа
Фитнес-интеллект

Найти

Search

Название	Дата	Количество лайков	Тип
Спектакль	09.07.2019	30	Групповой выезд
Приглашение специалистов	09.12.2019	23	Лекция
День рождения клуба	30.06.2020	0	Праздник

Рисунок 3.2 – Поиск мероприятия по диапазону календарных дат

Группы Домашние задания Мероприятия Педагоги
Shumina Выйти

Мероприятия

Дата мероприятия от 01/05/2020 До 31/05/2020
Количество лайков от До
Группа:

Раннее развитие
Английский 1 группа
Фитнес-интеллект

Найти

Search

Название	Дата	Кол-во лайков	Кол-во дизлайков	Тип
Выезд в зоопарк	30.05.2020	0	0	Групповой выезд

Рисунок 3.3 – Результат поиска в рамках контрольного задания

После нажатия на кнопку «Перейти в чат» перед пользователем открывается окно с чатом (рисунок 3.5). Родитель выбирает тип сообщения, пишет текст сообщения в специальное окно ввода и нажимает кнопку «Отправить», после чего его сообщение отобразится в диалоговом окне чата (рисунок 3.6).

Таким образом, родитель с логином «Shumina» успешно выполнил задачу по поиску мероприятия и публикации в обсуждении требуемого информационного сообщения для других родителей и воспитателей.

The screenshot shows the 'Просмотр Мероприятия' (View Event) page. At the top is a blue navigation bar with links: 'Группы', 'Домашние задания', 'Мероприятия', and 'Педагоги'. On the right of the bar are two buttons: 'Shumina' and 'Выйти'. The main title 'Просмотр Мероприятия' is centered. Below it, the event details are listed: 'Название: Выезд в зоопарк', 'Описание: Выезд в зоопарк планируется в 8:30, машины заказали', 'Дата: 30.05.2020', 'Тип: Групповой выезд', and 'Группы: Раннее развитие' and 'Английский 1 группа'. A red button 'Перейти в чат' is located below the group selection.

Рисунок 3.4 – Просмотр искомого мероприятия

The screenshot shows the 'Чат' (Chat) page. It has the same blue navigation bar as Figure 3.4. The main title 'Чат' is centered. Below it, the chat details are listed: 'Название: Выезд в зоопарк', 'Дата: 30.05.2020', and 'Тип: Групповой выезд'. There is a dropdown menu for 'Тип сообщения:' with 'Обратная связь' selected. Below this is a text input field for the message, containing the text 'Мы завтра опоздаем, скажите пожалуйста, во сколько отходит машина?'. There is a checkbox for 'Анонимно' which is unchecked. Below the input field is a blue button 'Отправить'. At the bottom, there is a section labeled 'Чат' with a large empty text area for the chat history.

Рисунок 3.5 – Ввод сообщения в чат

3.2 Проверка эффективности системы

Разработанная АИС может использоваться для ведения учета клиентов в ДДУ, информирования родителей об организуемых воспитательно-образовательных мероприятиях и о домашних заданиях для их детей, получения

воспитателем обратной связи для анализа интереса детей/родителей в отношении того или иного мероприятия и домашней работы. Благодаря интуитивно понятному интерфейсу системы операции управления данными не требуют большого количества времени. Традиционный метод ведения учёта на бумаге не обладает достаточным удобством и простотой, к его недостаткам можно отнести возможность потери или порчи носителя, ошибки в записи. В отличие от способа ведения учета на бумаге, программный вариант намного быстрее, удобнее и надежнее, так как вероятность допущения ошибки намного меньше. Также, одним из главных преимуществ АИС является возможность редактировать данные в любое время.

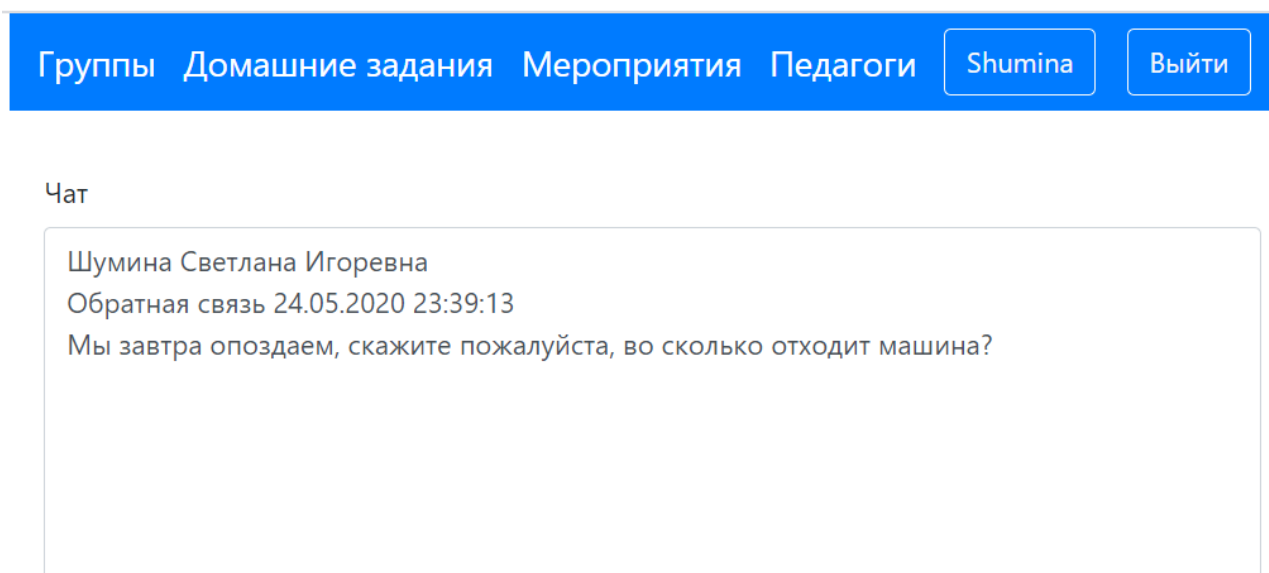


Рисунок 3.6 – Результат ввода сообщения в чат

Исходя из выше сказанного, можно сделать вывод о том, что внедрение разработанной АИС положительно повлияет на работу ДДУ. Это позволит повысить осведомленность и вовлеченность родителей в воспитательно–образовательные процессы, улучшить качество взаимодействия родителей и воспитателей через дополнительный канал общения, получать обратную связь от родителей для проведения последующего анализа качества проводимых мероприятий.

Так как АИС представляет из себя WEB-приложение, то имеется возможность использовать её на различных устройствах с различными операционными системами.

3.3 Разработка пользовательской документации

Пользовательская документация для АИС объясняет пользователям, как они должны действовать, чтобы применить разрабатываемое ПО. Подробное руководство пользователя для всех ролей представлено в Приложении Г.

3.4 Выводы по главе

Таким образом, в данной главе была реализована проверка работоспособности и эффективности системы, а также разработана документация для пользователей с различными ролями и представлен контрольный пример проверки работоспособности системы.

ЗАКЛЮЧЕНИЕ

Таким образом, в данной работе была исследована деятельность, связанная с взаимодействием родителей и воспитателей дошкольного учреждения, по результатам анализа которой был сделан вывод о необходимости повышения эффективности коммуникационного процесса взаимодействия и вовлеченности участников.

В рамках проведенного обоснования необходимости проектирования и разработки АИС были выявлены требования, предъявляемые к проектируемой системе. С целью формирования функционала проектируемой автоматизированной информационной системы было проведено функциональное моделирование, разработаны диаграмма прецедентов, диаграммы последовательностей и диаграмма состояний. Вместе с тем, были спроектированы логическая и физическая модели базы данных.

В результате при помощи Microsoft Visual Studio 2019 разработана АИС взаимодействия родителей и воспитателей, внедрение которой позволит обеспечить выполнение следующих положительных эффектов:

- 1) Повысится осведомленность и вовлеченность родителей в воспитательно–образовательных процессах детских дошкольных учреждений;
- 2) Улучшится качество взаимодействия родителей и воспитателей через дополнительный канал общения;
- 3) Воспитатели смогут получать обратную связь для анализа интереса детей/родителей в отношении того или иного мероприятия и того или иного домашнего задания;
- 4) Руководители и воспитатели смогут получать обратную связь от родителей для проведения последующего анализа качества проводимых мероприятий в детском дошкольном учреждении;
- 5) Повысится репутация дошкольного учреждения в глазах родителей.

Подытожив вышесказанное, внедрение разработанной автоматизированной информационной системы позволит повысить эффективность работы воспитательного учреждения за счет повышения качества

коммуникационного процесса взаимодействия родителей и воспитателей. Сформулированные методические основы автоматизации коммуникационного процесса могут быть использованы коммерческими и некоммерческими организациями для разработки модели информационной архитектуры, реализующей переход к современной платформе управления.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Киселев, Г.М. Информационные технологии в педагогическом образовании [Текст]/Г.М. Киселев, Р.В. Бочкова. – 2-е изд., перераб. и доп. – М.: Издательско-торговая корпорация «Дашков и К», 2014.-304 с.
- 2 Вдовенко, Л.А. Информационная система предприятия [Текст]/Л.А. Вдовенко. – М.: Вузовский учебник, Инфра – М, 2016.-240 с.
- 3 Громов, А.И. Управление бизнес-процессами: современные методы : монография [Текст]/А.И. Громов, А. Фляйшман, В. Шмидт. – Люберцы: Юрайт, 2016.-367 с.
- 4 Ежкова, Н.С. Дошкольное образование с позиции возрастных ценностей [Текст]/Н.С. Ежкова. – Москва: ИЛ, 2017.-357 с.
- 5 Козлова, А.М. Дошкольное образование [Текст]/А.М. Козлова, Е.В. Никифорова, Н.А. Скопинова. – М.: ЦГЛ, 2017.-958 с.
- 6 Кулаков, А.В. Модернизация образования и повышение квалификации педагогических кадров [Текст]/А.В. Кулаков, В.Н. Манюкова//Среднее профессиональное образование. - 2014. - №6. - С.3-6.
- 7 Майер, А.А. Интеграция основных компонентов дошкольного образования [Текст]/А.А. Майер. – М.: Сфера, 2015.-128 с.
- 8 Ивлев, В.А. Информационные системы на основе действий [Текст]/В.А. Ивлев, Т.В. Попова. – М.: 1С-Паблишинг, 2017.-248 с.
- 9 Кочкина, Н.А. Метод проектов в дошкольном образовании [Текст]/Н.А. Кочкина. – М.: Мозаика-Синтез, 2017.-542 с.
- 10 Нелис, Й. Управление бизнес-процессами: Практическое руководство по успешной реализации проектов [Текст]/Й. Нелис, Д. Джестон. – СПб.: Символ-плюс, 2015.-512 с.
- 11 Тельнов, Ю.Ф. Реинжиниринг бизнес-процессов. Компонентная методология [Текст]/Ю. Ф. Тельнов. – Изд. 2-е, перераб. и доп. – М.: Финансы и статистика, 2014.-320 с.
- 12 Репин, В.В. Бизнес-процессы. Моделирование, внедрение, управление [Текст]. – М.: Манн, Иванов и Фербер, 2013.-512 с.

- 13 Ротер, М. Учись видеть бизнес-процессы: Построение карт потоков создания ценности [Текст]/М. Ротер. – М.: Альпина Паблишер, 2015.-136 с.
- 14 Мезенцев, К.Н. Автоматизированные информационные системы [Текст]/ К.Н. Мезенцев. – М.: Академия, 2017.-176 с.
- 15 Гвоздева, В.А. Информатика, автоматизированные информационные технологии и системы [Текст]/В.А. Гвоздева. – М.: Форум, Инфра-М, 2017.-544 с.
- 16 Репин, В.В. Процессный подход к управлению. Моделирование бизнес-процессов [Текст]/В.В. Репин. – М.: Манн, Иванов и Фербер, 2013.-544 с.
- 17 Редько, В.Н. Базы данных и информационные системы [Текст]/В.Н. Редько, И.А. Басараб. – М.: Знание, 2016.-667 с.
- 18 Википедия. Microsoft SQL Server [Электронный ресурс]. – <https://ru.wikipedia.org/wiki/MicrosoftSQLServer>
- 19 Википедия. Microsoft Visual Studio [Электронный ресурс]. – <https://ru.wikipedia.org/wiki/MicrosoftVisualStudio>

ПРИЛОЖЕНИЕ А

Таблица А.1 - Сравнительный анализ СУБД

Критерии оценки	MS Access	MySQL	MS SQL Server	Oracle
Размер базы данных				
несколько Мб	+	+		
до 100Мб	+	+		
несколько ГБ		+	+	+
100Гб и более			+	+
Количество одновременных пользователей, работающих с базой данных напрямую				
1 пользователь	+	+		
до 10	+	+		
более 10		+	+	+
более 100		+	+	+
более 1000			+	+
Стоимость лицензии				
бесплатно (только лицензия на пакет)	+	+		
дешево				
дорого			+	+
очень дорого				
Платформа, на которой способна работать СУБД				
Windows	+		+	+
Unix/Linux				+
Windows+Linux		+		+
Кластеры			+	
Встроенный язык программирования				
Microsoft	+		+	
Языки для Linux			+	
Тип программы				
Локальное приложение	+			+
WEB		+		
Сложная система			+	+
Защита данных				
низкая				
слабая	+			
сильная		+	+	+
Требования к техническим характеристикам ПК				
неприхотливые	+	+		
чувствительные				
необходимы мощные сервера			+	+
Сложность настройки и администрирования				
никаких сложностей	+			
небольшие сложности				

ПРИЛОЖЕНИЕ Б

Таблица Б.1 – Описание сущностей модели базы данных АИС взаимодействия родителей и воспитателей ДОУ

Наименование атрибута	Описание	Тип данных MS SQL
Сущность МЕРОПРИЯТИЕ		
ID Мероприятия	Идентификационный номер Мероприятия	uniqueidentifier
Название	Название Мероприятия	nvarchar (50)
Описание	Подробное описание Мероприятия	nvarchar (4000)
Дата проведения	Дата проведения Мероприятия	datetime
Число лайков	Отражает отношение пользователя к проведенному Мероприятия	int
Число дизлайков	Отражает отношение пользователя к проведенному Мероприятия	int
ID Типа Мероприятия	Идентификационный номер Типа Мероприятия	uniqueidentifier
Сущность ТИП МЕРОПРИЯТИЯ		
ID Типа Мероприятия	Идентификационный номер Типа Мероприятия	uniqueidentifier
Тип	Тип Мероприятия	nvarchar (50)
Сущность СООБЩЕНИЕ		
ID Сообщения	Идентификационный номер Сообщения	uniqueidentifier
Текст Сообщения	Текст Сообщения	nvarchar (4000)
Дата и время создания Сообщения	Дата и время публикации Сообщения	datetime
Флаг «Аноним»	Отражает возможность публикации анонимного Сообщения	bit
ID Типа Сообщения	Идентификационный номер Типа Сообщения	uniqueidentifier
ID Пользователя	Идентификационный номер Пользователя	uniqueidentifier
ID Мероприятия	Идентификационный номер Мероприятия	uniqueidentifier
ID Домашнего Задания	Идентификационный номер Домашнего Задания	uniqueidentifier

Продолжение таблицы Б.1

Сущность ТИП СООБЩЕНИЯ		
ID Типа Сообщения	Идентификационный номер Типа Сообщения	uniqueidentifier
Тип	Тип Сообщения	nvarchar (50)
Сущность ТИП УЧЕТНОЙ ЗАПИСИ		
ID Типа Учетной записи	Идентификационный номер Типа Учетной записи	uniqueidentifier
Тип	Тип Учетной записи	nvarchar (50)
Сущность УЧЕТНАЯ ЗАПИСЬ		
ID УЧЕТНОЙ ЗАПИСИ	Идентификационный номер Учетной записи	uniqueidentifier
Логин	Логин Учетной записи в системе	nvarchar (50)
Пароль	Пароль Учетной записи в системе	nvarchar (50)
ID Типа Учетной записи	Идентификационный номер Типа Учетной записи	uniqueidentifier
Сущность ПОЛЬЗОВАТЕЛЬ		
ID Пользователя	Идентификационный номер Пользователя	uniqueidentifier
ФИО	Фамилия, имя и отчество Пользователя	nvarchar (50)
Е-mail	Е-mail Пользователя	nvarchar (50)
Телефон	Телефонный номер Пользователя	nvarchar (12)
Дата рождения	Дата рождения Пользователя	datetime
ID Учетной записи	Идентификационный номер Учетной записи	uniqueidentifier
Сущность ПЕДАГОГ		
ID Пользователя	Идентификационный номер Пользователя	uniqueidentifier
Портфолио	Портфолио Педагога	nvarchar (4000)
Стаж	Стаж работы Педагога	nvarchar (12)
Сущность РОДИТЕЛЬ		
ID Родителя	Идентификационный номер Родителя	uniqueidentifier
Флаг «Многодетная семья»	Отражает наличие в семье больше двух детей	bit
Сущность РЕБЕНОК		
ID Ребенка	Идентификационный номер Ребенка	uniqueidentifier
Дата рождения	Дата рождения Ребенка	datetime
ФИО	Фамилия, имя и отчество Ребенка	nvarchar (50)
Пол	Отражает пол ребенка	nvarchar (1)

Продолжение таблицы Б.1

Мать	Идентификационный номер Родителя с ролью - Мать	uniqueidentifier
Отец	Идентификационный номер Родителя с ролью - Отец	uniqueidentifier
Сущность ДЕТИ В ГРУППЕ		
ID Ребенка	Идентификационный номер Ребенка	uniqueidentifier
ID Группы	Идентификационный номер Группы	uniqueidentifier
Сущность ГРУППА		
ID Группы	Идентификационный номер Группы	uniqueidentifier
Название	Название Группы	nvarchar (50)
Описание	Подробное описание Группы	nvarchar (4000)
Время проведения занятий	Расписание проводимых занятий в Группе	nvarchar (100)
ID Педагога	Идентификационный номер Педагога	uniqueidentifier
Сущность ДЗ ДЛЯ ГРУППЫ		
ID Домашнего Задания	Идентификационный номер Домашнего Задания	uniqueidentifier
ID Группы	Идентификационный номер Группы	uniqueidentifier
Сущность ДОМАШНЕЕ ЗАДАНИЕ		
ID Домашнего Задания	Идентификационный номер Домашнего Задания	uniqueidentifier
Название	Название Домашнего Задания	nvarchar (50)
Описание	Подробное описание Домашнего Задания	nvarchar (4000)
Дата	Дата выполнения Домашнего Задания	datetime
Количество лайков	Количество лайков, поставленных пользователями, конкретному Домашнему Заданию	int
Количество дизлайков	Количество дизлайков, поставленных пользователями, конкретному Домашнему Заданию	int
Ссылка на дополнительные материалы	Ссылка на дополнительные материалы для выполнения ДЗ	nvarchar (4000)

Продолжение таблицы Б.1

ID Педагога	Идентификационный номер педагога	uniqueidentifier
ID Типа Домашнего Задания	Идентификационный номер Типа Домашнего Задания	uniqueidentifier
Сущность ТИП ДОМАШНЕГО ЗАДАНИЯ		
ID Типа Домашнего Задания	Идентификационный номер Типа Домашнего Задания	uniqueidentifier
Тип	Тип Домашнего Задания	nvarchar (50)
Сущность УЧАСТНИКИ МЕРОПРИЯТИЯ		
ID Группы	Идентификационный номер Группы	uniqueidentifier
ID Мероприятия	Идентификационный номер Мероприятия	uniqueidentifier
Сущность РЕЙТИНГ ДОМАШНЕГО ЗАДАНИЯ		
ID Домашнего Задания	Идентификационный номер Домашнего Задания	uniqueidentifier
ID Пользователя	Идентификационный номер Пользователя	uniqueidentifier
Тип обратной связи	Определяет тип оценки Домашнему Заданию от Пользователя(лайк/дизлайк)	bit
Сущность РЕЙТИНГ МЕРОПРИЯТИЯ		
ID Мероприятия	Идентификационный номер Мероприятия	uniqueidentifier
ID Пользователя	Идентификационный номер Пользователя	uniqueidentifier
Тип обратной связи	Определяет тип оценки Мероприятию от Пользователя(лайк/дизлайк)	bit

ПРИЛОЖЕНИЕ В

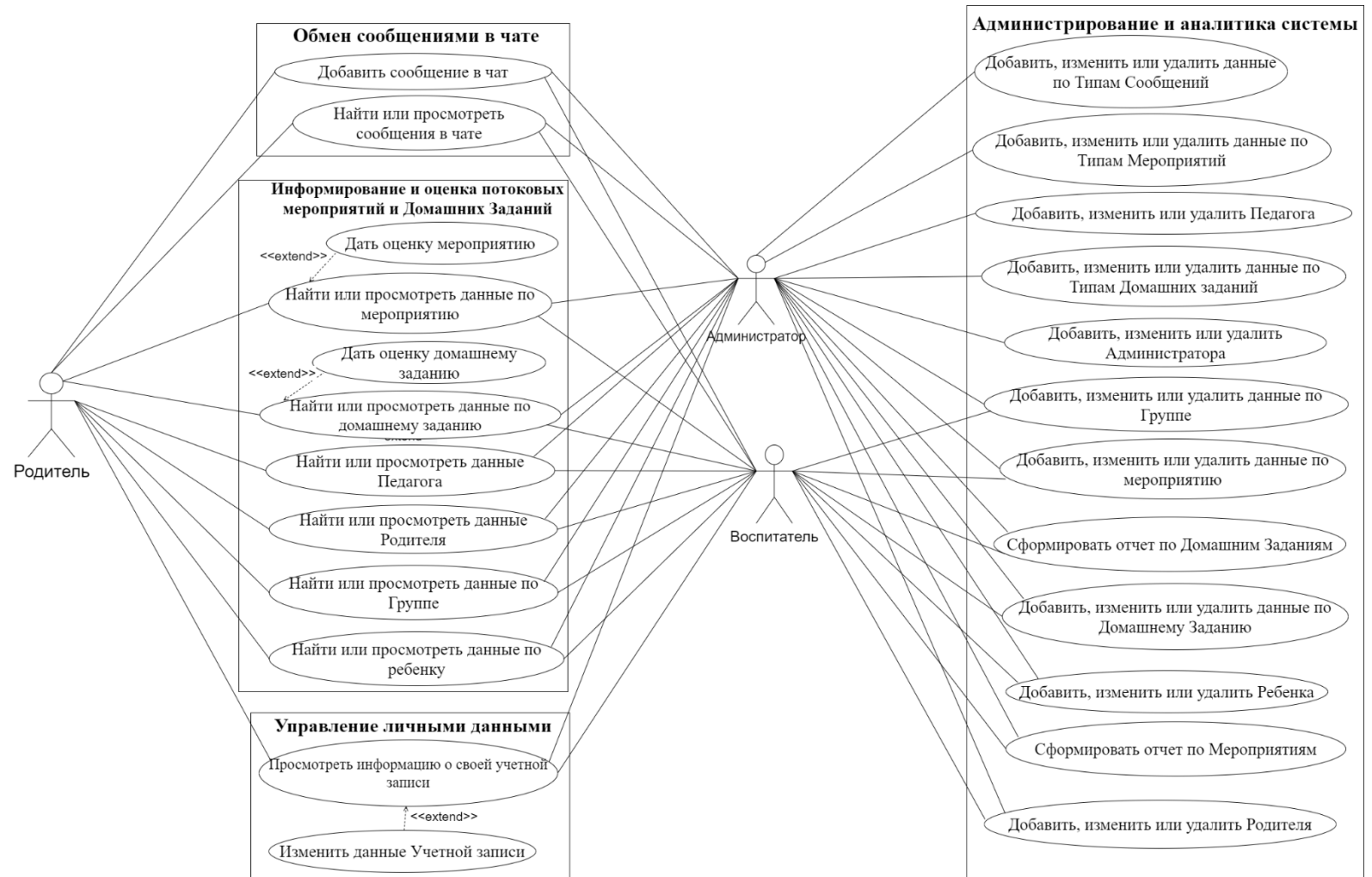


Рисунок Б.1 – Диаграмма прецедентов АИС

ПРИЛОЖЕНИЕ Г

Разработка пользовательской документации

Для работы с АИС в первую очередь необходимо авторизоваться. На рисунке Г.1 представлены формы авторизации, отражающие процесс ввода данных и информационное сообщение о неправильно введенных данных.

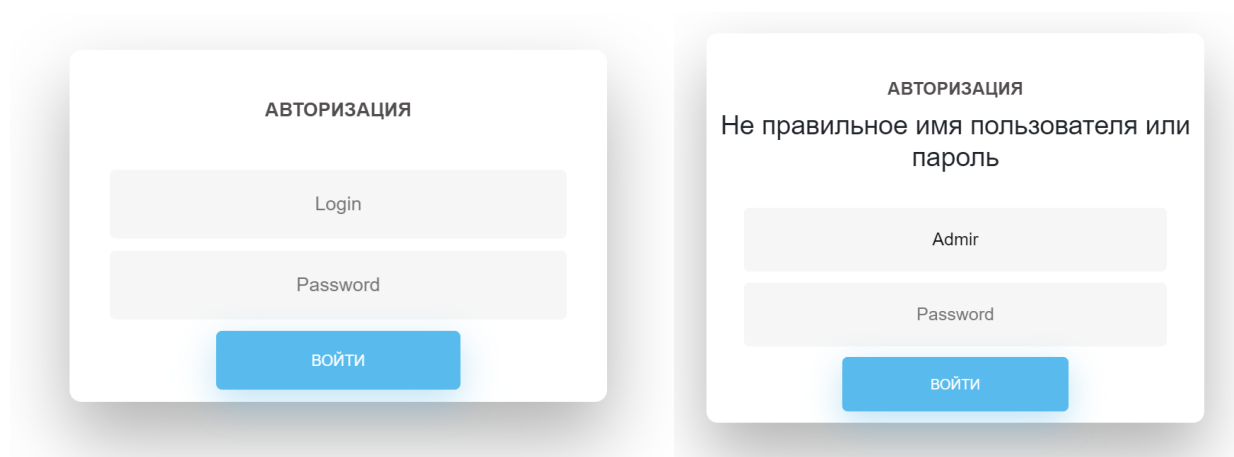


Рисунок Г.1 – Формы авторизации

После успешной авторизации появляется главная форма АИС. При этом отображение вкладок зависит от роли авторизованного пользователя. На рисунке Г.2 представлена рабочая область информационной системы для пользователя с ролью «Администратор», которому доступен полный функционал.

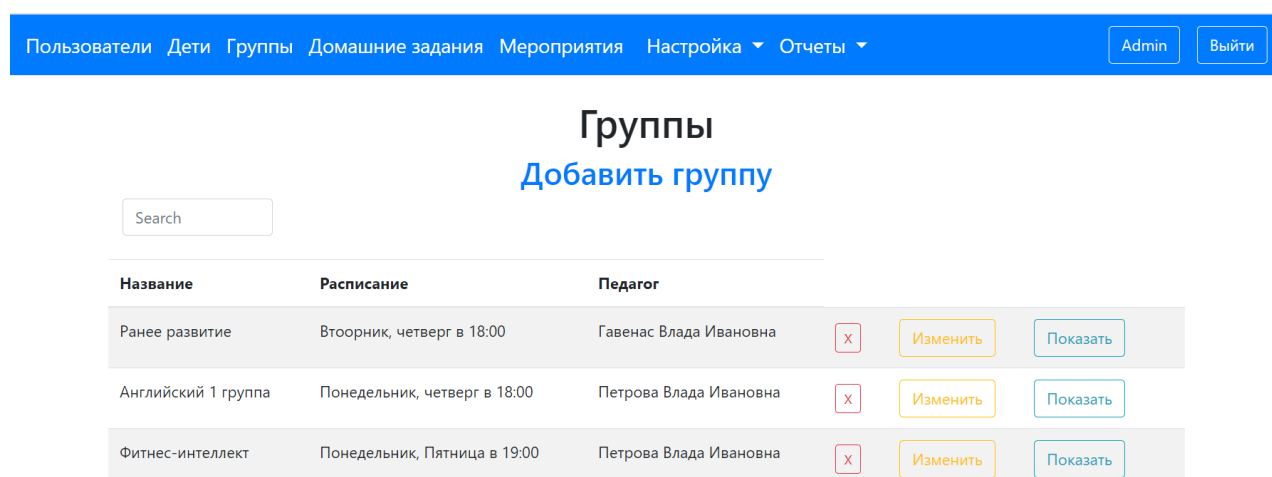
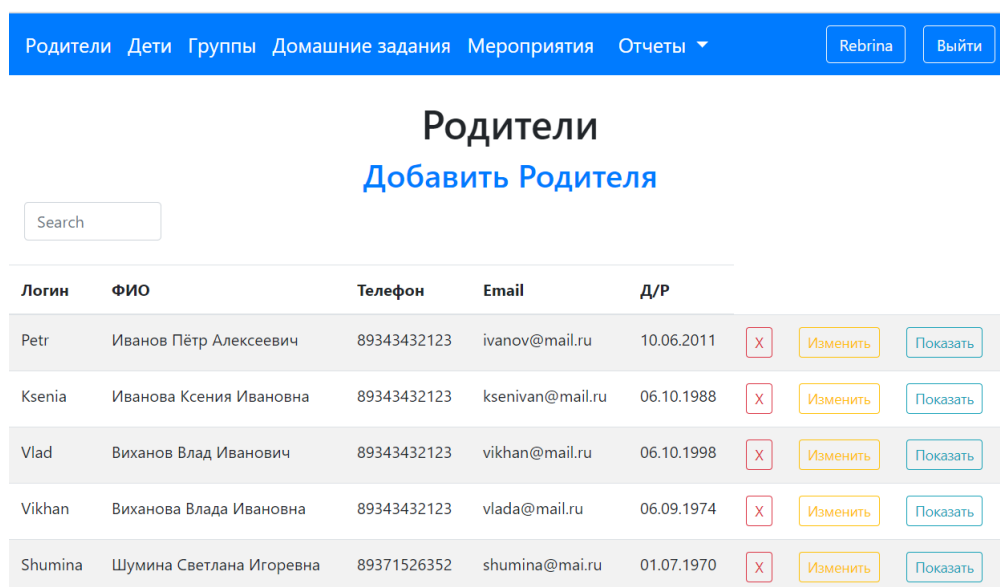


Рисунок Г.2 – Рабочая область для пользователя с ролью «Администратор»

Г.1 Разработка руководства пользователя с ролью «Воспитатель»

Ключевой особенностью функционала пользователя с ролью «Воспитатель» является возможность добавления, изменения и удаления родителей, работающих с АИС. На рисунке Г.3 представлена рабочая область вкладки «Родители», на которой представлен перечень пользователей.

Кнопка «X» позволяет воспитателю удалить запись о родителе, кнопка «Изменить» позволяет изменять информацию о родителе, кнопка «Show» отображает подробную информацию о родителе.



Родители	Дети	Группы	Домашние задания	Мероприятия	Отчеты ▾	Rebrina	Выйти
Родители							
Добавить Родителя							
<input type="text" value="Search"/>							
Логин	ФИО	Телефон	Email	Д/Р			
Petr	Иванов Пётр Алексеевич	89343432123	ivanov@mail.ru	10.06.2011	X	Изменить	Показать
Ksenia	Иванова Ксения Ивановна	89343432123	ksenivan@mail.ru	06.10.1988	X	Изменить	Показать
Vlad	Виханов Влад Иванович	89343432123	vikhan@mail.ru	06.10.1998	X	Изменить	Показать
Vikhan	Виханова Влада Ивановна	89343432123	vlada@mail.ru	06.09.1974	X	Изменить	Показать
Shumina	Шумина Светлана Игоревна	89371526352	shumina@mail.ru	01.07.1970	X	Изменить	Показать

Рисунок Г.3 – Рабочая область вкладки «Родители»

На рисунке Г.4 представлена рабочая область по добавлению и изменению существующей записи о родителе.

На рисунке Г.5 представлена страница отображения подробной информации о родителе.

Кнопка «Показать детей» позволяет просмотреть всех детей выбранного родителя. На рисунке Г.6 представлена страница отображения всех детей Иванова Петра Алексеевича.

Для удобства работы пользователей на каждой вкладке расположена поисковая строка по таблице. На рисунке Г.7 представлены фильтры на вкладке «Родители».

Добавление Родителя Редактирование Родителя

Логин:	<input type="text"/>	Логин:	<input type="text" value="Petr"/>
Пароль:	<input type="password"/>	Пароль:	<input type="password" value="parent"/>
ФИО:	<input type="text"/>	ФИО:	<input type="text" value="Иванов Пётр Алексеевич"/>
Телефон:	<input type="text"/>	Телефон:	<input type="text" value="89343432123"/>
Email:	<input type="text"/>	Email:	<input type="text" value="ivanov@mail.ru"/>
Дата Рождения:	<input type="text" value=""/>	Дата Рождения:	<input type="text" value="06/10/2011"/>
Многодетная семья:	<input type="checkbox"/>	Многодетная семья:	<input checked="" type="checkbox"/>
<input type="button" value="Добавить"/>		<input type="button" value="Изменить"/>	

Рисунок Г.4 – Формы добавления и изменения записи о пользователе

Родители Дети Группы Домашние задания Мероприятия Отчеты ▾

Просмотр Родителя

Логин:	Petr
Пароль:	parent
ФИО:	Иванов Пётр Алексеевич
Телефон:	89343432123
Email:	ivanov@mail.ru
Дата Рождения:	10.06.2011
Многодетная семья:	Да
<input type="button" value="Показать детей"/>	

Рисунок Г.5 – Форма отображения подробной информации о родителе

На вкладке «Мероприятия» пользователь с ролью «Воспитатель» имеет возможность добавления, изменения, просмотра подробной информации о мероприятии и удаления мероприятий. На рисунке Г.8 представлена рабочая область вкладки «Мероприятия», на которой представлен перечень заведенных в системе мероприятий.

Дети

Добавить ребенка

Search

ФИО	Пол	Д/Р	Мать	Отец		
Иванова Дарья Ивановна	ж	09.06.2015	Иванова Ксения Ивановна	Иванов Пётр Алексеевич	X	Изменить
Иванов Влад Иванович	м	09.06.2016	Иванова Ксения Ивановна	Иванов Пётр Алексеевич	X	Изменить

Рисунок Г.6 – Страница отображения всех детей выбранного родителя

Поиск

Логин	ФИО	Телефон	Email	Д/Р			
Petr	Иванов Пётр Алексеевич	89343432123	ivanov@mail.ru	10.06.2011	X	Изменить	Показать
Ksenia	Иванова Ксения Ивановна	89343432123	ksenivan@mail.ru	06.10.1988	X	Изменить	Показать

Рисунок Г.7 – Поисковая строка на вкладке «Родители»

Родители
Дети
Группы
Домашние задания
Мероприятия
Отчеты

Rebrina
Выйти

Мероприятия

Добавить Мероприятие

Дата мероприятия от

До

Количество лайков от

До

Группа:

Раннее развитие

Английский 1 группа

Фитнес-интеллект

Найти

Search

Название	Дата	Кол-во лайков	Кол-во дизлайков	Тип					
Выезд в зоопарк	30.05.2020	0	0	Групповой выезд	X	Изменить	Показать	Лайк	Дизлайк
Приглашение специалистов	09.12.2019	23	5	Лекция	X	Изменить	Показать	Лайк	Дизлайк
Спектакль	09.07.2019	30	8	Групповой выезд	X	Изменить	Показать	Лайк	Дизлайк
День рождение клуба	30.06.2020	0	0	Праздник	X	Изменить	Показать	Лайк	Дизлайк

Рисунок Г.8 – Рабочая область вкладки «Мероприятия»

Пользователь может изменить информацию о мероприятии с использованием формы, представленной на рисунке Г.9.

Редактирование Мероприятия

Название:	Выезд в зоопарк
Описание:	Выезд в зоопарк планируется в 8:30, машины заказали
Дата:	30.05.2020
Тип:	Групповой выезд
Группы:	<div> <div>Ранее развитие</div> <div>Английский 1 группа</div> <div>Фитнес-интеллект</div> </div>

Изменить

Рисунок Г.9 – Форма изменения существующей записи о проводимом мероприятии

Пользователь также может воспользоваться поиском необходимых мероприятий при помощи фильтров, представленных на рисунке Г.10.

Дата мероприятия от До

Количество лайков от До

Группа:

Ранее развитие

Английский 1 группа

Фитнес-интеллект

Найти

Рисунок Г.10 – Фильтры на вкладке «Проводимые мероприятия»

На рисунке Г.11 представлена страница отображения подробной информации о мероприятии.

Кнопка «Перейти в чат» позволяет перейти в чат выбранного мероприятия. На рисунке Г.12 представлена страница с отображением чата.

Необходимо отметить, что пользователи имеют возможность:

- Публиковать анонимные сообщения в обсуждении проводимого мероприятия с использованием кнопки «Анонимно»;
- Выбирать категорию сообщения для упрощения организации коммуникаций между воспитателями и родителями (рисунок Г.13).

Просмотр Мероприятия

Название: Выезд в зоопарк

Описание: Выезд в зоопарк планируется в 8:30, машины заказали

Дата: 30.05.2020

Тип: Групповой выезд

Группы: Раннее развитие
Английский 1 группа

[Перейти в чат](#)

Рисунок Г.11 – Форма отображения подробной информации о мероприятии

Чат

Название: Выезд в зоопарк

Дата: 30.05.2020

Тип: Групповой выезд

Тип сообщения: Обратная связь ▾

Сообщение

☐ Анонимно

[Отправить](#)

Чат

Шумина Светлана Игоревна
Обратная связь 24.05.2020 23:39:13
Мы завтра опоздаем, скажите пожалуйста, во сколько отходит машина?

Рисунок Г.12 – Работа в чате мероприятия

Описанные выше функциональные возможности пользователей в части публикации дополнительной информации в процессе переписки родителей и воспитателей позволяют в последствии воспитателю формировать сводные отчеты в разрезе диапазона дат и тем проводимых мероприятий (рисунок Г.14). По результатам анализа таких отчетов воспитатель или вышестоящее руководство могут сделать выводы о качестве проведённого воспитательно-образовательного мероприятия и при необходимости уделить дополнительное внимание изучению того или иного направления.

Рисунок Г.13 – Категорирование сообщений в чате

Мероприятие	Лайки	Дизлайки	Вопросы	Ответы	Комментарии	Обратная связь	Всего сообщений в чате
Выезд в зоопарк	0	0	0	0	0	1	1
Приглашение специалистов	23	5	2	2	1	1	6
Спектакль	30	8	3	2	3	2	10

Рисунок Г.14 – Формирование отчета

Г.2 Разработка руководства пользователя с ролью «Родитель»

Пользователю с ролью «Родитель» доступны только вкладки «Группы», «Домашние задания», «Мероприятия» и «Педагоги» с урезанным функционалом в части изменения. На рисунке Г.15 представлена рабочая область вкладки «Мероприятия» для пользователя с ролью «Родитель».

Пользователь с ролью «Родитель» имеет возможность просмотра информации о проводимых мероприятиях (рисунок Г.16).

На рисунке Г.17 представлена вкладка «Педагоги», на которой пользователь имеет возможность ознакомиться со списком педагогов, работающих в ДДУ.

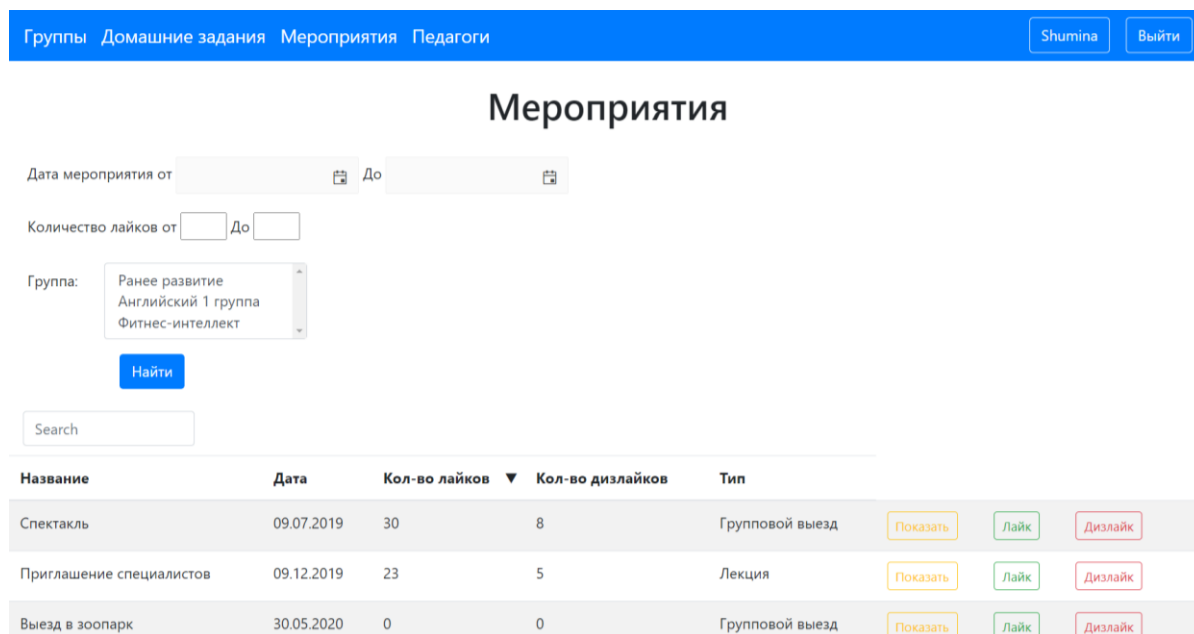


Рисунок Г.15 – Рабочая область вкладки «Мероприятия» для пользователя с ролью «Родитель»

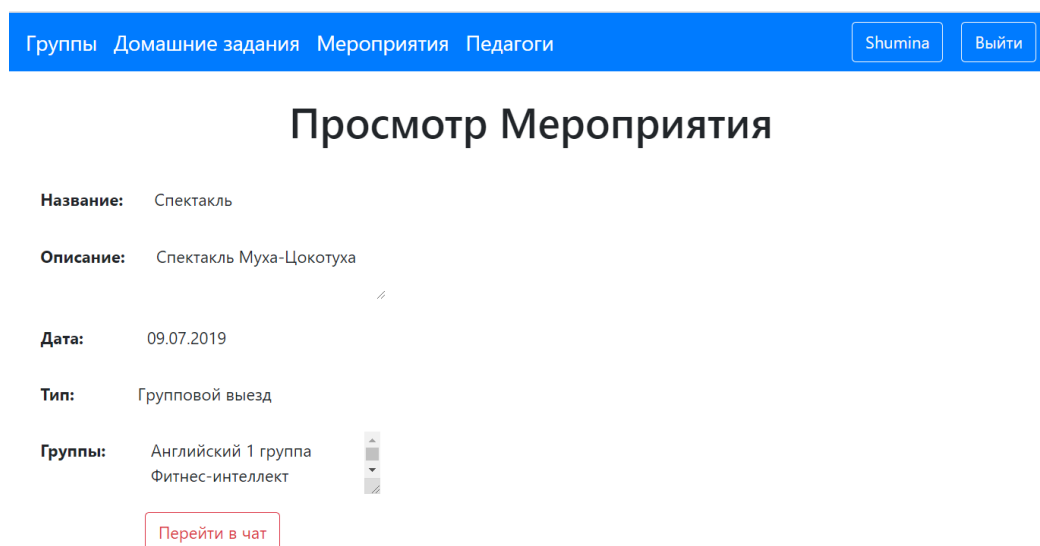


Рисунок Г.16 – Форма просмотра информации о мероприятиях для пользователя с ролью «Родитель»

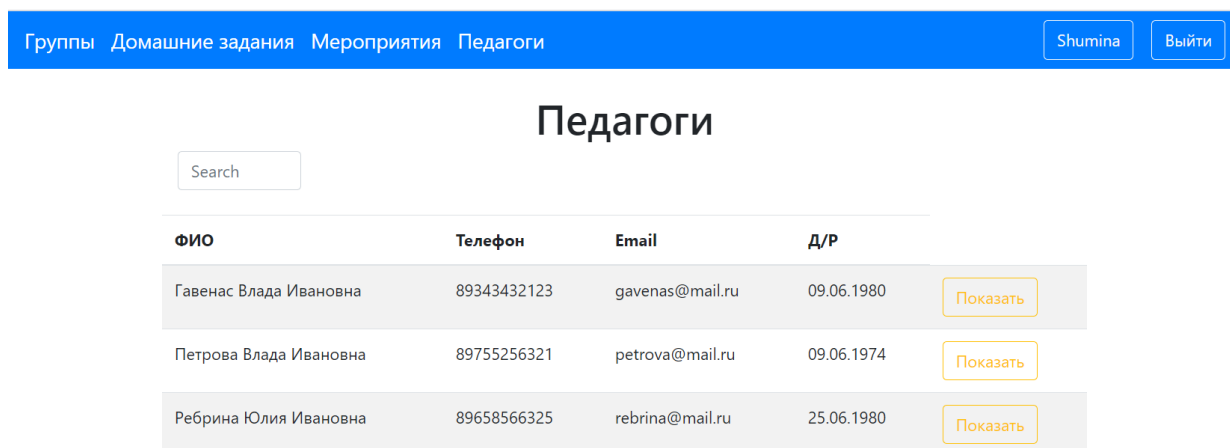


Рисунок Г.17 – Вкладка «Педагоги»

ПРИЛОЖЕНИЕ Д

Листинг программы

Серверная часть

Account.cs

```
using System;

namespace Server.Models
{
    public partial class Account
    {
        public Guid Id { get; set; }
        public string Login { get; set; }
        public string Password { get; set; }
        public Guid AccountTypeId { get; set; }

        public AccountType AccountType { get; set; }
        public User User { get; set; }
    }
}
```

AccountRepository.cs

```
using Microsoft.EntityFrameworkCore;
using Server.Models;
using System;
using System.Collections.Generic;
using System.Linq;

namespace Server.DataBase.Repository
{
    public class AccountRepository : IRepository<Account>
    {
        private DduContext _dduContext; // контекст базы данных

        public AccountRepository(DduContext dduContext)
        {
            _dduContext = dduContext;
        }

        public Account Get(Guid id)
        {
            return _dduContext.Accounts.Find(id);
        }

        public List<Account> GetAll()
        {
            return _dduContext.Accounts.Include(x => x.AccountType).Include(x => x.User).ToList();
        }

        public Guid Add(Account Account)
        {
            _dduContext.Accounts.Add(Account);
            _dduContext.SaveChanges();
            return Account.Id;
        }

        public void Delete(Guid id)
        {
            Account Account = _dduContext.Accounts.Find(id);

            if (Account != null)
            {
                _dduContext.Accounts.Remove(Account);
                _dduContext.SaveChanges();
            }
        }

        public void Update(Account Account)
        {
            _dduContext.Entry(Account).State = EntityState.Modified;
            _dduContext.SaveChanges();
        }
    }
}
```

```
}
```

AccountController.cs

```
using System;
using System.Collections.Generic;
using Microsoft.AspNetCore.Mvc;
using Server.DataBase.Repository;
using Server.Models;

namespace Server.Controllers
{
    [Route("api/[controller]")]
    [ApiController]
    public class AccountController : ControllerBase
    {
        private AccountRepository _AccountRepository; //репозиторий для операций с базой

        public AccountController(AccountRepository AccountRepository)
        {
            _AccountRepository = AccountRepository;
        }

        [Route("{id}", Name = "GetAccount"), HttpGet]
        public ActionResult<Account> GetAccount(Guid id)
        {
            return _AccountRepository.Get(id);
        }

        [Route("", Name = "GetAccounts"), HttpGet]
        public ActionResult<IEnumerable<Account>> GetAllAccounts()
        {
            return _AccountRepository.GetAll();
        }

        [Route("", Name = "AddAccount"), HttpPost]
        public ActionResult<Guid> AddAccount(Account Account)
        {
            return _AccountRepository.Add(Account);
        }

        [Route("", Name = "UpdateAccount"), HttpPut]
        public ActionResult UpdateAccount(Account Account)
        {
            _AccountRepository.Update(Account);
            return Ok();
        }

        [Route("{id}", Name = "DeleteAccount"), HttpDelete]
        public ActionResult DeleteAccount(Guid id)
        {
            _AccountRepository.Delete(id);
            return Ok();
        }
    }
}
```

Клиентская часть

AccountController.cs

```
using Client.Models;
using Client.Models.ServerModels;
using Client.Services;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;

namespace Client.Controllers
```

```

{
    public class AccountController : Controller
    {
        private AccountService _accountService;//сервис для вызова функций сервера
        private AccountTypeService _accountTypeService;
        private ParentService _parentService;
        private TeacherService _teacherService;
        private UserService _userService;
        private ChildService _childService;
        public AccountController()//конструктор
        {
            _parentService = new ParentService();
            _teacherService = new TeacherService();
            _accountService = new AccountService();
            _accountTypeService = new AccountTypeService();
            _userService = new UserService();
            _childService = new ChildService();
        }

        public ActionResult Index()
        {
            if (Session["userId"] == null)
            {
                return View("~/Views/Login/LoginView.cshtml", new LoginModel());
            }

            List<Parent> parents = _parentService.GetAllParents();
            List<Teacher> teachers = _teacherService.GetAllTeachers();
            IEnumerable<SelectListItem> accountTypes = from item in
_accountTypeService.GetAllAccountTypes()
                                                    select new SelectListItem { Text = item.Type, Value
= item.Id.ToString() };

            IEnumerable<AccountModel> parentList = from item in parents
                                                    select new AccountModel
                                                    {
                UserId = item.User.Id,
                ParentId = item.Id,
                AccountId = item.User.AccountId,
                AccountTypeId = item.User.Account.AccountTypeId,
                Login = item.User.Account.Login,
                AccountType = item.User.Account.AccountType.Type,
                Fio = item.User.Fio,
                Email = item.User.Email,
                Birthsday = item.User.Birthsday,
                BirthsdayStr = item.User.Birthsday != null ?
item.User.Birthsday.Value.ToString("dd/MM/yyyy") : "n/a",
                LargeFamilieStr = item.LargeFamilie ? "Yes" : "No",
                LargeFamilie = item.LargeFamilie,
                Phone = item.User.Phone,
                Password = item.User.Account.Password
            };

```



```

IEnumerable<AccountModel> teacherList = from item in teachers
                                         select new AccountModel
                                         {
                                             TeacherId = item.Id,
                                             UserId = item.User.Id,
                                             AccountId = item.User.AccountId,
                                             AccountTypeId = item.User.Account.AccountTypeId,
                                             Login = item.User.Account.Login,
                                             AccountType = item.User.Account.AccountType.Type,
                                             Fio = item.User.Fio,
                                             Email = item.User.Email,
                                             Birthsday = item.User.Birthsday,
                                             BirthsdayStr = item.User.Birthsday != null ?
item.User.Birthsday.Value.ToString("dd/MM/yyyy") : "n/a",
                                             Phone = item.User.Phone,
                                             Password = item.User.Account.Password,
                                             Portfolio = item.Portfolio,
                                             Experience = item.Experience
                                         };

IEnumerable<AccountModel> accountModels = parentList.Concat(teacherList);

return View("ListAccountView", accountModels);
}

public ActionResult ShowAccount(AccountModel accountModel)//просмотр информации о пользователе
{
    if (Session["userId"] == null)
    {
        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }

    return View("ShowAccountView", accountModel);
}

public ActionResult EditAccount(AccountModel accountModel)//вызов страницы изменения информации о
пользователе
{
    if (Session["userId"] == null)
    {
        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }
    return View("EditAccountView", accountModel);
}

[HttpPost]
public ActionResult UpdateAccount(AccountModel accountModel)//изменение информации о пользователе
{
    if (Session["userId"] == null)
    {

```

```

        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }
    try
    {
        _accountService.UpdateAccount(new Account()
        {
            Id = accountModel.AccountId,
            Login = accountModel.Login,
            Password = accountModel.Password,
            AccountTypeId = accountModel.AccountTypeId
        });

        _userService.UpdateUser(new User()
        {
            Id = accountModel.UserId,
            Fio = accountModel.Fio,
            Phone = accountModel.Phone,
            Email = accountModel.Email,
            Birthsday = accountModel.Birthsday,
            AccountId = accountModel.AccountId
        });

        if (accountModel.AccountType == "Parent")
        {
            _parentService.UpdateParent(new Parent()
            {
                Id = accountModel.ParentId,
                LargeFamilie = accountModel.LargeFamilie,
            });
        }

        if (accountModel.AccountType == "Teacher")
        {
            _teacherService.UpdateTeacher(new Teacher()
            {
                Id = accountModel.TeacherId,
                Portfolio = accountModel.Portfolio,
                Experience = accountModel.Experience
            });
        }
    }
    catch (Exception)
    {
        return View("EditAccountView", accountModel);
    }

    return RedirectToAction("Index");
}

```

```

public ActionResult DeleteAccount(AccountModel accountModel)
{
    if (Session["userId"] == null)
    {
        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }

    if (accountModel.AccountType == "Parent")
    {
        List<Child> children = _childService.GetAllChildren();
        foreach (var child in children)
        {
            if (child.FatherId == accountModel.ParentId && child.MotherId == null ||
child.MotherId == accountModel.ParentId && child.FatherId == null)
            {
                _childService.DeleteChild(child.Id);
            }
            else
            {
                if (accountModel.ParentId == child.MotherId)
                {
                    child.MotherId = null;
                    _childService.UpdateChild(child);
                }

                if (accountModel.ParentId == child.FatherId)
                {
                    child.FatherId = null;
                    _childService.UpdateChild(child);
                }
            }
        }

        _parentService.DeleteParent(accountModel.ParentId);
        _userService.DeleteUser(accountModel.ParentId);
        _accountService.DeleteAccount(accountModel.AccountId);
    }

    if (accountModel.AccountType == "Teacher")
    {
        _teacherService.DeleteTeacher(accountModel.TeacherId);
    }

    return RedirectToAction("Index");
}

[HttpGet]
public ActionResult AddAccount()//вызов страницы добавления аккаунта
{
    if (Session["userId"] == null)
    {
        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }

```

```

    }

    IEnumerable<SelectListItem> accountTypes = from item in
        _accountTypeService.GetAllAccountTypes()
        select new SelectListItem { Text = item.Type, Value
        = item.Id.ToString() };

    AccountModel accountModel = new AccountModel
    {
        AccountTypes = accountTypes
    };

    return View("AddAccountView", accountModel);
}

[HttpPost]
public ActionResult AddAccount(AccountModel accountModel)//добавление пользователя
{
    if (Session["userId"] == null)
    {
        return View("~/Views/Login/LoginView.cshtml", new LoginModel());
    }

    try
    {
        string role = _accountTypeService.GetAccountType(accountModel.AccountTypeId).Type;
        Guid acc = _accountService.AddAccount(new Account()
        {
            Login = accountModel.Login,
            Password = accountModel.Password,
            AccountTypeId = accountModel.AccountTypeId,
        });

        Guid us = _userService.AddUser(new User()
        {
            Fio = accountModel.Fio,
            Phone = accountModel.Phone,
            Email = accountModel.Email,
            Birthsday = accountModel.Birthsday,
            AccountId = acc
        });

        if (role == "Parent")
        {
            _parentService.AddParent(new Parent()
            {
                LargeFamilie = accountModel.LargeFamilie,
                Id = us
            });
        }

        if (role == "Teacher")
        {

```

```

        _teacherService.AddTeacher(new Teacher()
        {
            Portfolio = accountModel.Portfolio,
            Experience = accountModel.Experience,
            Id = us
        });
    }
}
catch (Exception)
{
    IEnumerable<SelectListItem> accountTypes = from item in
_accountTypeService.GetAllAccountTypes()
                                                select new SelectListItem { Text = item.Type,
Value = item.Id.ToString() };

    accountModel.AccountTypes = accountTypes;

    return View("AddAccountView", accountModel);
}

return RedirectToAction("Index");
}
}
}
}

```

AccountService.cs

```

using Client.Models.ServerModels;
using System;
using System.Collections.Generic;
using System.Net.Http;

namespace Client.Services
{
    public class AccountService : BaseService
    {
        public AccountService() : base() { }
        public List<Account> GetAllAccounts()
        {
            string url = "account";
            var response = Client.GetAsync(url).Result;

            response.EnsureSuccessStatusCode();

            return response.Content.ReadAsAsync<List<Account>>().Result;
        }

        public Guid AddAccount(Account account)//добавление пользователя
        {
            var content = GetContent(account);
            string url = "account";
            var response = Client.PostAsync(url, content).Result;

            response.EnsureSuccessStatusCode();

            return response.Content.ReadAsAsync<Guid>().Result;
        }
        public Guid UpdateAccount(Account account)//обновление информации о ползователе
        {
            var content = GetContent(account);
            string url = "account";
            var response = Client.PutAsync(url, content).Result;

```

```

        response.EnsureSuccessStatusCode();

        return response.Content.ReadAsAsync<Guid>().Result;
    }
    public void DeleteAccount(Guid id)//удаление пользователя
    {
        string url = "account/" + id.ToString();
        var response = Client.DeleteAsync(url).Result;

        response.EnsureSuccessStatusCode();
    }
}
}

```

ListAccountView.cshtml

```

@using Client.Models;
@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@model IEnumerable<AccountModel>

<div class="jumbotron">
    <br>
    <center>
        <h1>Пользователи</h1>
        <h2>
            <a href="@Url.Action("AddAccount", "Account")">Добавить пользователя</a>
        </h2>
    </center>
    <div id="container">
        <div class="col-2" align="left">
            <input class="form-control" type="text" placeholder="Search" id="search-text"
onkeyup="tableSearch()">
        </div>
        <br>
        <table class="table table-striped" id="info-table">
            <thead>
                <tr>
                    <th>Тип</th>
                    <th>Логин</th>
                    <th>ФИО</th>
                    <th>Телефон</th>
                    <th>Email</th>
                    <th>Д/Р</th>
                </tr>
            </thead>
            <tbody>
                @foreach (AccountModel item in @Model)
                {
                    <tr>
                        <td>@item.AccountType</td>
                        <td>@item.Login</td>
                        <td>@item.Fio</td>
                        <td>@item.Phone</td>
                        <td>@item.Email</td>
                        <td>@item.BirthdayStr</td>
                        <td>
                            @Html.ActionLink("X", "DeleteAccount", "Account", item, new { @class = "btn
btn-sm btn-outline-danger" })
                            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
                            @Html.ActionLink("Edit", "EditAccount", "Account", item, new { @class = "btn
btn-outline-warning" })
                            &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&
                            @Html.ActionLink("Show", "ShowAccount", "Account", item, new { @class = "btn
btn-outline-warning" })
                        </td>
                    </tr>
                </tr>
                }
            </tbody>
        </table>
    </div>
</div>

```

AddAccountView.cshtml

```

@{

```

```

        Layout = "~/Views/Shared/_Layout.cshtml";
    }
    @model Client.Models.AccountModel

    <div class="jumbotron">
        <br>
        <center>
            <h1>Добавление нового пользователя</h1>
        </center>
        <br>
        <div id="addForm">
            @using (Html.BeginForm("AddAccount", "Account", FormMethod.Post, new { enctype = "multipart/form-
data" }))
            {
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-2 col-form-label">Логин</label>
                    <div class="col-sm-3">
                        @Html.TextBoxFor(m => m.Login, new { @class = "form-control", placeholder = "Enter
Login" })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-2 col-form-label">Пароль</label>
                    <div class="col-sm-3">
                        @Html.TextBoxFor(m => m.Password, new { @class = "form-control", placeholder = "Enter
Password" })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-3 col-form-label">Тип учетной записи</label>
                    <div class="col-sm-3">
                        @Html.DropDownListFor(model => model.AccountTypeId, Model.AccountTypes, new { @class =
"custom-select", id = "CategoryTrees" })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-2 col-form-label">ФИО</label>
                    <div class="col-sm-4">
                        @Html.TextBoxFor(m => m.Fio, new { @class = "form-control", placeholder = "Enter FIO"
                    })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-2 col-form-label">Телефон</label>
                    <div class="col-sm-3">
                        @Html.TextBoxFor(m => m.Phone, new { @class = "form-control", placeholder = "Enter
Phone" })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-2 col-form-label">Email</label>
                    <div class="col-sm-3">
                        @Html.TextBoxFor(m => m.Email, new { @class = "form-control", placeholder = "Enter
Email" })
                    </div>
                </div>
                <div class="form-group row">
                    <label for="inputEmail3" class="col-sm-3 col-form-label">Дата Рождения</label>
                    <div class="col-sm-9">
                        @Html.TextBoxFor(m => m.Birthday, new { @class = "form-control", id = "Birthday",
placeholder = "Enter Birthday" })
                    </div>
                </div>

                <div style="display:none" id="blokone">
                    <div class="form-group row">
                        <div class="col-sm-3">Многодетная семья</div>
                        <div class="col-sm-1">
                            <div class="form-check">
                                @Html.CheckBoxFor(m => m.LargeFamilie, new { @class = "form-check-input" })
                            </div>
                        </div>
                    </div>
                </div>

                <div style="display:none" id="bloktwo">
                    <div class="form-group row">

```

```

        <label for="inputEmail3" class="col-sm-2 col-form-label">Стаж</label>
        <div class="col-sm-3">
            @Html.TextBoxFor(m => m.Experience, new { @class = "form-control", placeholder =
"Enter Experience" })
        </div>
    </div>
    <div class="form-group row">
        <label for="inputEmail3" class="col-sm-2 col-form-label">Портфолио</label>
        <div class="col-sm-5">
            @Html.TextAreaFor(m => m.Portfolio, new { @class = "form-control", placeholder =
"Enter Portfolio" })
        </div>
    </div>
    </div>
    <input type="submit" class="btn btn-primary" value="Добавить" style="margin-left: 100px;" />
    @Html.HiddenFor(m => m.AccountId)
    @Html.HiddenFor(m => m.AccountType)
    @Html.HiddenFor(m => m.ParentId)
    @Html.HiddenFor(m => m.UserId)
    }
</div>
</div>
<script>
    $(document).ready(function () {
        $("#Birthsday").kendoDatePicker({
            dateInput: false,
            format: 'dd/MM/yyyy'
        });
    });
</script>
<script>
    document.getElementById("CategoryTrees").onchange = function () {
        var b = {
            Parent: "blokone",
            Teacher: "bloktwo"
        }, c = this.options[this.selectedIndex].text,
        a;
        for (a in b) document.getElementById(b[a])
            .style.display = 0 == c || c == a ? "block" : "none"
    };
</script>

```

ShowAccountView.cshtml

```

@{
    Layout = "~/Views/Shared/_Layout.cshtml";
}
@model Client.Models.AccountModel

<div class="jumbotron">
    <br>
    <center>
        <h1>Просмотр пользователя</h1>
    </center>
    <br>
    <form style="margin-left: 30px">
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-2 col-form-label">Логин</label>
            <div class="col-sm-2">
                @Html.TextBoxFor(m => m.Login, new { @readonly = true, @class = "form-
control-plaintext" })
            </div>
        </div>
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-2 col-form-label">Пароль</label>
            <div class="col-sm-2">
                @Html.TextBoxFor(m => m.Password, new { @readonly = true, @class = "form-
control-plaintext" })
            </div>
        </div>
    </form>
    <div class="form-group row">

```



```

        <label for="staticEmail" class="col-sm-2 col-form-label">Тип учетной
записи</label>
        <div class="col-sm-2">
            @Html.TextBoxFor(m => m.AccountType, new { @readonly = true, @class =
"form-control-plaintext" })
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-2 col-form-label">ФИО</label>
        <div class="col-sm-3">
            @Html.TextBoxFor(m => m.Fio, new { @readonly = true, @class = "form-
control-plaintext" })
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-2 col-form-label">Телефон</label>
        <div class="col-sm-2">
            @Html.TextBoxFor(m => m.Phone, new { @readonly = true, @class = "form-
control-plaintext" })
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-2 col-form-label">Email</label>
        <div class="col-sm-2">
            @Html.TextBoxFor(m => m.Email, new { @readonly = true, @class = "form-
control-plaintext" })
        </div>
    </div>
    <div class="form-group row">
        <label for="staticEmail" class="col-sm-2 col-form-label">Дата
Рождения</label>
        <div class="col-sm-2">
            @Html.TextBoxFor(m => m.BirthdayStr, new { @readonly = true, @class =
"form-control-plaintext" })
        </div>
    </div>
    @if (Model.AccountType == "Parent")
    {
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-2 col-form-label">Многодетная
семья</label>
            <div class="col-sm-2">
                @Html.TextBoxFor(m => m.LargeFamilieStr, new { @readonly = true,
@class = "form-control-plaintext" })
            </div>
        </div>
    }
    @if (Model.AccountType == "Teacher")
    {
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-2 col-form-
label">Портфолио</label>
            <div class="col-sm-3">
                @Html.TextAreaFor(m => m.Portfolio, new { @readonly = true, @class =
"form-control-plaintext" })
            </div>
        </div>
        <div class="form-group row">
            <label for="staticEmail" class="col-sm-2 col-form-label">Стаж</label>
            <div class="col-sm-2">
                @Html.TextBoxFor(m => m.Experience, new { @readonly = true, @class =
"form-control-plaintext" })
            </div>
        </div>
    }
}

```

```
        @if (Model.AccountType == "Parent")
        {
            @Html.ActionLink("Показать детей", "ShowChildrenFromParent", "Child", new {
                parentId = Model.ParentId }, new { @class = "btn btn-sm btn-outline-danger" })
        }
    </form>
</div>
```