

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Институт математики физики и информационных технологий
(наименование института полностью)

09.03.03 Прикладная информатика

(код и наименование направления подготовки / специальности)

Корпоративные информационные системы

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Автоматизация работы учебно-методического отдела образовательной
организации

Обучающийся

Я.А Полбицын

(Инициалы Фамилия)

(личная подпись)

Руководитель

канд. техн. наук, доцент, О.В Аникина

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

Т.С. Якушева

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Аннотация

Тема бакалаврской работы «Автоматизация работы учебно-методического отдела образовательной организации».

Бакалаврская работа посвящена разработке автоматизированной системы для решения основных задач сотрудниками учебно-методического отдела центра дополнительного образования детей.

Целью данной работы является разработка информационной системы, автоматизирующей работу сотрудников учебно-методического отдела учреждения дополнительного образования детей.

Объектом исследования бакалаврской работы является деятельность сотрудников учебно-методического отдела.

Предметом исследования является автоматизированная система, обеспечивающая выполнение основных функций сотрудников учебно-методического отдела.

Практическая значимость работы заключается в разработке информационной системы, автоматизирующей работу сотрудников учебно-методического отдела по основным видам деятельности.

Результатом выпускной квалификационной работы является разработанное приложение для автоматизации деятельности сотрудников учебно-методического отдела учреждения дополнительного образования детей.

Бакалаврская работа состоит из введения, 3 глав, заключения, списка используемой литературы.

Annotation

The theme of the bachelor's work is "Automation of the work of the educational and methodological department of an educational organization."

Bachelor's work is devoted to the development of an automated system for solving the main tasks by employees of the educational and methodological department of the center for additional education of children.

The purpose of this work is to develop an information system that automates the work of employees of the educational and methodological department of an institution of additional education for children.

The object of study of the bachelor's work is the activities of the staff of the educational and methodological department.

The subject of the study is an automated system that ensures the performance of the main functions of the employees of the educational and methodological department.

The practical significance of the work lies in the development of an information system that automates the work of the employees of the educational and methodological department for the main activities.

The result of the final qualifying work is a developed application for automating the activities of employees of the educational and methodological department of an institution of additional education for children.

Bachelor's work consists of introduction, 3 chapters, conclusion, list of used literature.

Оглавление

Введение	5
Глава 1 Анализ деятельности образовательной организации.....	7
1.1 Техничко-экономическая характеристика учреждения дополнительного образования.....	7
1.2 Концептуальное моделирование и анализ документооборота учебно-методического отдела.....	12
1.3 Постановка задачи на разработку	20
1.4 Анализ существующих разработок в области дополнительного образования.....	21
1.5 Разработка модели бизнес-процесса «КАК БУДЕТ»	23
1.6 Требования к разрабатываемой информационной системе	28
Глава 2 Разработка приложения для учебного отдела центра дополнительного образования	32
2.1 Концептуальное проектирование приложения для учебно- методического отдела	32
2.2 Логическое и физическое моделирование деятельности учебно- методического отдела	35
2.3 Реализация приложения для автоматизации деятельности учебного отдела.....	44
Глава 3 Тестирование реализованного приложения для учебно- методического отдела	54
3.1 Тестирование реализованного приложения	54
3.2 Пример работы реализованного приложения	56
Заключение	66
Список используемой литературы	67

Введение

В деятельности учреждений дополнительного образования детей необходимо вести систематический учет деятельности педагогов дополнительного образования и тренеров-преподавателей, формировать и вносить коррективы в списки обучающихся, составлять и анализировать планы проведения мероприятий и занятий по учебным программам, контролировать их реализацию и вести отчетность.

Актуальность темы данной работы обусловлена необходимостью автоматизации данного процесса в учебно-методическом отделе Центра дополнительного образования детей. Это приведет к рациональному управлению документацией по работе учреждения в современных условиях и позволит целенаправленно формировать информационные ресурсы организации, обеспечивать их эффективное функционирование.

База исследования: исследование будет проводиться на примере типового центра дополнительного образования детей (далее Центр).

Целью данной работы является разработка информационной системы, автоматизирующей работу сотрудников учебно-методического отдела учреждения дополнительного образования детей.

Объектом исследования бакалаврской работы является деятельность сотрудников учебно-методического отдела.

Предметом исследования является автоматизированная система, обеспечивающая выполнение основных функций сотрудников учебно-методического отдела.

Для достижения поставленной цели в ходе работы решаются следующие задачи:

- проанализировать организационную структуру, обеспеченность аппаратным и программным обеспечением учебно-методического отдела;

- проанализировать существующие бизнес-процессы в деятельности учебно-методического отдела для выявления «узких» мест и способов их устранения;
- сформулировать требования к разрабатываемой информационной системе;
- спроектировать приложение, реализующее все необходимые функции;
- выбрать средства реализации информационной системы;
- разработать базу данных в рамках выбранной СУБД и клиентские приложения для автоматизации учета деятельности педагогов, подготовки планов и отчетов об учебных программах и мероприятиях согласно установленным формам отчетности;
- протестировать разработанное приложение с целью выявления недостатков и их устранения.

Практическая значимость работы заключается в разработке информационной системы, автоматизирующей работу сотрудников учебно-методического отдела по основным видам деятельности.

Бакалаврская работа состоит из введения, 3 глав, заключения, списка используемой литературы.

Глава 1 Анализ деятельности образовательной организации

1.1 Техничко-экономическая характеристика учреждения дополнительного образования

Автоматизация деятельности учебного отдела будет проводиться на примере типового центра дополнительного образования для школьников.

Центр дополнительного образования для школьников (далее Центр) осуществляет свою деятельность в соответствии с законами и иными нормативными правовыми актами Российской Федерации, Самарской области, муниципальными правовыми актами городского округа Тольятти, а также Уставом.

Целями деятельности Центра являются:

- развитие мотивации личности к познанию и творчеству;
- реализация дополнительных образовательных программ и услуг в интересах личности, общества и государства;
- удовлетворение потребности детей в занятиях физической культурой, спортом и творчеством.

Для достижения указанных целей Центр осуществляет следующие основные виды деятельности:

- организация и проведение досуговой и внеурочной деятельности детей, массовых мероприятий с обучающимися, их родителями (законными представителями) и педагогами: конкурсы, олимпиады, соревнования, экскурсии;
- организация и проведение отдыха детей. В каникулярное время Центр может открывать в установленном порядке лагеря, создавать различные объединения с постоянными и переменными составами детей в лагерях (загородных или с дневным пребыванием) на своей базе, а также по месту жительства детей;

- оказание помощи педагогическим коллективам других образовательных учреждений в реализации дополнительных образовательных программ, организации досуговой и внеурочной деятельности детей, а также детским общественным объединениям и организациям по договору с ними;
- организация конкурсов, фестивалей, смотров различных уровней;
- организация и проведение соревнований, показательных выступлений, игровых, развлекательных и других праздников.

В Центре ведется методическая работа, направленная на совершенствование образовательного процесса, программ, форм и методов деятельности. Данной деятельностью занимается учебно-методический отдел.

Автоматизация деятельности учебно-методического отдела представляет собой изучение взаимодействий его как внешних, так и с внутренних факторов, которые необходимы для функционирования отдела, это нужно для того, чтобы предложить оптимальное решение одной из главных функций, которые выполняет данный отдел, это учёт результативности деятельности педагогов, направленной на развитие потенциала детей, посещающих центр. Учебно-методический отдел регулирует образовательный процесс детей, посещающих Центр, а также по месту жительства при проведении мероприятий во время каникул.

Также в отделе реализуются следующие задачи методической направленности:

- контроль состояния учебно-воспитательной работы Центра;
- повышение профессиональной компетентности тренеров-преподавателей, педагогов дополнительного образования (ДО);
- разработка критериев качества образования;
- развитие мотивационной сферы личности педагогическими средствами;
- научно-методическая работа.

Таким образом, деятельность учебно-методического отдела направлена на повышение результативности работы педагогов ДО и тренеров-

преподавателей, определение соответствия профессиональной компетентности сотрудников занимаемым должностям, тем самым, обеспечивая эффективную реализацию задач, стоящих перед Центром как учреждением дополнительного образования детей и школьников.

Организационная структура типового Центра представлена на рисунке 1. В учебно-методическом отделе работают следующие сотрудники: заместитель по учебно-воспитательной работе, руководитель структурного подразделения, методисты. Сотрудники учебно-методического отдела непосредственно подчиняются директору Центра.

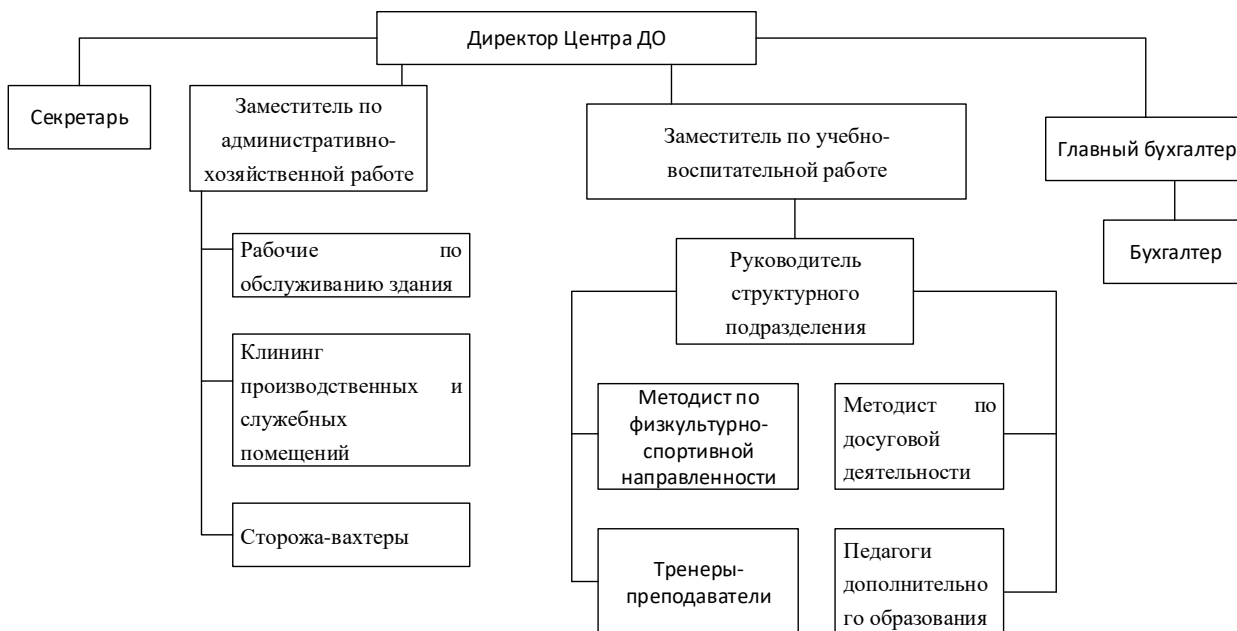


Рисунок 1 - Организационная структура Центра дополнительного образования школьников

Должностные обязанности заместителя по учебно-воспитательной работе:

- организует текущее и перспективное планирование деятельности педагогического коллектива;
- координирует работу педагогических работников по выполнению учебных планов и программ;

- организует и координирует разработку необходимой учебно-методической документации, руководит деятельностью методистов по подготовке учебно-методической документации;
- оказывает помощь педагогическим работникам в освоении и разработке инновационных программ и технологий;
- составляет план работы и других видов воспитательной (в том числе культурно-досуговой) деятельности;
- обеспечивает своевременное составление установленной отчетной документации, контролирует правильное и своевременное ведение тренерами-преподавателями и педагогами ДО журналов, другой документации;
- участвует в комплектовании организации, принимает меры по сохранению контингента воспитанников;
- оказывает помощь в проведении культурно-просветительских и оздоровительных мероприятиях;
- участвует в подборе и расстановке педагогических кадров, организует повышение их квалификации и профессионального мастерства;
- организует работу по соблюдению норм и правил охраны труда;
- организует разработку и периодический пересмотр не реже 1 раза в 5 лет инструкций по охране труда, а также разделов требований техники безопасности жизнедеятельности;
- контролирует своевременное проведение инструктажа воспитанников и его регистрацию в журнале.

Основные должностные обязанности руководителя структурного подразделения:

- организует работу спортивной школы;
- несет полную ответственность за все направления ее деятельности;
- определяет совместно с педагогическим советом основные направления развития спортивной школы;
- руководит работой педагогического совета;

- осуществляет контроль, за выполнением спортивных учебных программ, содержанием и эффективностью учебно-тренировочных занятий, соблюдением требований безопасности учебно-тренировочных занятий;

- обеспечивает рациональное использование выделяемых спортивной школе финансовых средств, определяет форму оплаты труда тренеров-преподавателей по спорту, распоряжается в соответствии с действующим законодательством имуществом и средствами;

- соблюдение правил санитарно-гигиенического режима, охраны труда, техники безопасности и противопожарной защиты;

- обеспечивает проведение внутри школьных, спортивных соревнований;

- осуществляет контроль за выполнением спортсменами-учащимися требований учебных программ, качеством знаний, умений и навыков, уровнем физического развития и подготовленности, своевременным прохождением ими углубленного медицинского осмотра;

- организует учебно-тренировочную, воспитательную и методическую работу в центре;

- несёт ответственность за организацию учебно-тренировочного процесса, комплектование, отбор в спортивную ориентацию занимающихся;

- осуществляет контроль за содержанием учебно-тренировочного процесса, выполнением спортсменами требований учебных программ, качеством знаний, умений и навыков, уровнем физического развития и подготовленности.

Должностные обязанности методистов:

- анализирует состояние учебно-методической и воспитательной работы, разрабатывает предложение по повышению её эффективности;

- оказывает помощь педагогическим работникам в определении в определении содержания форм, методов и средств обучения;

- разрабатывает предложения по повышению эффективности образовательного процесса и методической работы;

- разрабатывает методы прогнозирования планирования подготовки специалистов и учебные планы по конкретным направлениям обучения;
- организует разработку, рецензирование и подготовку к утверждению учебно-методической документации, пособий;
- анализирует и обобщает результаты работы центра;
- организует и разрабатывает необходимую документацию по проведению конкурсов.

Практической значимостью выпускной квалификационной работы является информационная система, автоматизирующая деятельность учебно-методического отдела. Наибольшее значение имеет в работе заместителя по учебно-воспитательной работе, т.к. в должностные обязанности входит ведение учета результативности деятельности педагогов дополнительного образования и тренеров-преподавателей, а также представление отчетности по установленным формам.

1.2 Концептуальное моделирование и анализ документооборота учебно-методического отдела

Основная задача учебно-методического отдела – организация и ведение учета результативности познавательно-развлекательной деятельности педагогов и тренеров-преподавателей (количество проведенных мероприятий, участие детей в соревнованиях, количество задействованных детей), а также представление установленной отчетности по мероприятиям и работе по учебным программам в Департамент образования. Для анализа документооборота данного отдела необходимо разделить потоки информации на входной и выходной, а также описать и проанализировать внутренний документооборот учебно-методического отдела [2].

Входная информация - это документы, данные, и их содержимое поступающие в анализируемый отдел извне, то есть из других взаимодействующих с отделом подразделений.

Входной информацией в данном случае можно считать следующие.

Список обучающихся для зачисления в Центр. Предоставляется педагогом или тренером-преподавателем в начале каждого учебного года. В список вносятся данные по всем детям, которые были записаны в учебную группу на основании предоставленных родителями документов (свидетельство о рождении или паспорт, медицинская справка), а также заявления. По каждой учебной группе предоставляется отдельный список [3].

Список обучающихся для отчисления. Предоставляется педагогом в течение учебного года в зависимости от изменения количества детей.

Расписание занятий отдельно по каждой программе.

Программа предоставляется педагогом на весь период обучения группы.

Журнал учета работы учебной группы ежемесячно предоставляется педагогами дополнительного образования и тренерами-преподавателями на проверку заместителю по учебно-воспитательной работе.

В начале каждого учебного года педагоги и тренеры-преподаватели предоставляют кроме списков воспитанников следующий пакет документов: план работы с родителями, план воспитательной работы с детьми, план деятельности по диагностике интересов и склонностей детей, план мероприятий в дни школьных каникул, план спортивно-массовой деятельности, план социально-значимой работы с детьми.

При внутреннем документообороте учебно-методического отдела формируются следующие документы: приказ «О комплектовании групп обучающихся», план внутриучрежденческого контроля и руководства учебно-воспитательным процессом в Центре, учебный план на учебный год, приказы об отчислении обучающихся из Центра, приказы о зачислении обучающихся в Центр и переводе их на следующие года обучения, приказ об оказании платных дополнительных образовательных услуг, программа «Каникулы», приказ об организации каникулярной занятости детей и подростков, приказы по выездным мероприятиям.

Выходная информация - это данные, документы, которые формируются после внутреннего документооборота самого подразделения для дальнейшего продвижения и оформления необходимых приказов.

Выходной информацией в данном случае можно считать следующую: сведения о дополнительных образовательных программах, отчет по воспитанникам и группам за учебный год, учебный план, отчет по проектам воспитанников, отчет по уровню реализации программ в учебном году (полностью реализованы, частично реализованы, не реализованы), отчет о работе Центра в период каникул (по месяцам), отчет о работе Центра в период летних каникул (сводный).

Заместитель директора по учебно-воспитательной работе (УВР) составляет статистические и списочные отчеты о работе Центра. В конце учебного года анализируется работа Центра по всем направлениям деятельности. На основе этого анализа составляется план работы на следующий учебный год. Таким образом, перечислив и проанализировав документы, участвующие непосредственно в функционировании учебно-методического отдела, приступим к рассмотрению построенной модели данных.

На основе анализа входной и выходной информации, организационной структуры отдела и нормативных документов, в т.ч. должностных инструкций, которыми руководствуются сотрудники учебно-методического отдела, построена функциональная модель предметной области. Диаграмма процесса «Работа учебно-методического отдела» представлена на рисунке 2. На данной диаграмме стандарта BPMN отображены главная функция исследуемой системы, входы и выходы ее деятельности, управляющее воздействие и механизмы реализации функции [5]. Основным входным информационным потоком будет являться информация об обучающихся, представленная в списках для зачисления, данные из списков для отчисления, приказы Департамента образования, связанные с выполнением определенных функций, а также программы, планы мероприятий, положения и сценарии,

расписание занятий, служебные записки, отчеты, журналы учета работы учебной группы и другие документы о деятельности педагогов дополнительного образования и тренеров-преподавателей.

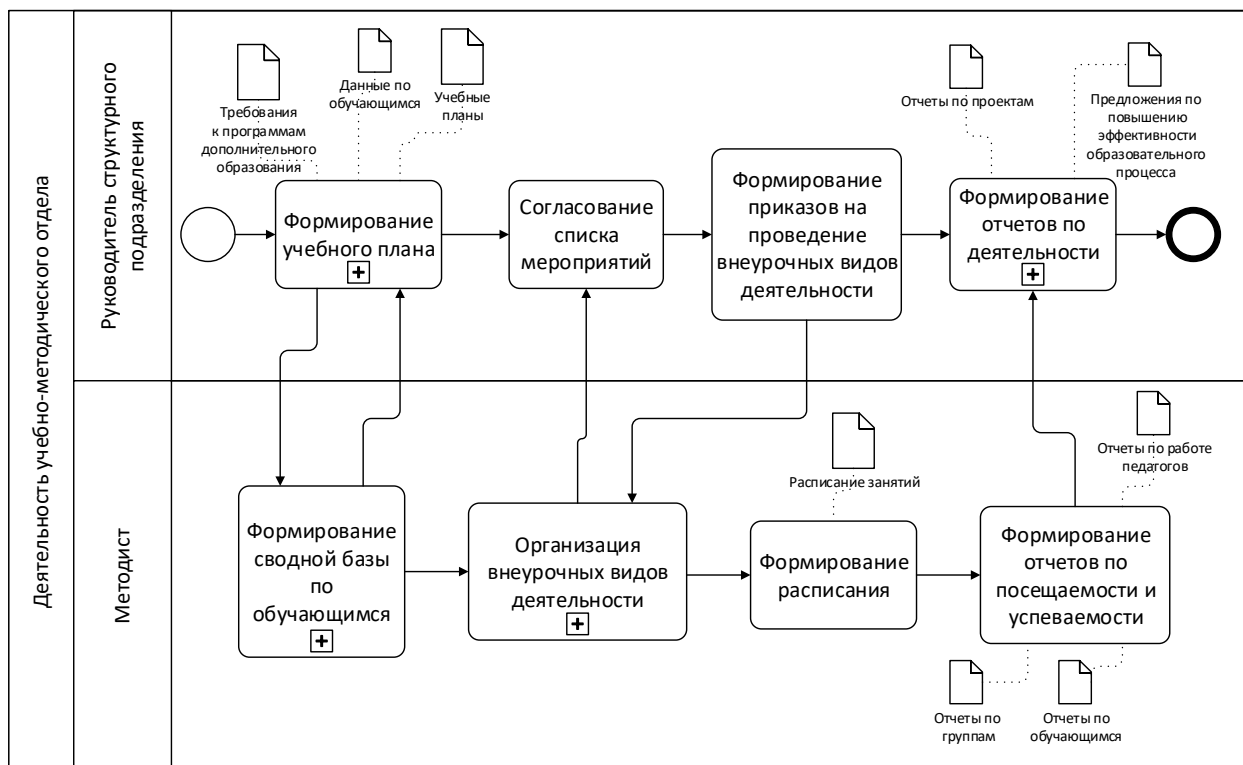


Рисунок 2 - Диаграмма процесса «Работа учебно-методического отдела» (AS-IS)

В результате обработки этих данных сотрудниками учебно-методического отдела, которые руководствуются в своей работе Уставом Центра, нормативами Департамента образования, должностными инструкциями, требованиями к программам дополнительного образования, получают документы учета результативности деятельности педагогов дополнительного образования и тренеров-преподавателей: сводная база обучающихся, планы мероприятий, сводные отчеты о занятости, учебный план.

Заместитель по учебно-воспитательной работе занимается подготовкой приказов по контролю деятельности педагогов, контролирует формирование планов и отчетов, и анализирует данные отчетов, на основе которых

впоследствии рассчитывает критерии результативности деятельности руководителей. Непосредственно подготовкой отчетов занимаются методисты, в зависимости от содержания отчета в его формировании может принимать участие заместитель по спортивно-массовой деятельности.

Рассмотрим поток информации, связанный с формированием и работой с базой данных обучающихся. Детализация контекстной диаграммы процесса «Формирование сводной базы по обучающимся» представлена на рисунке 3.

Изначально педагоги дополнительного образования и тренеры-преподаватели набирают обучающихся по объединениям, вносят данные по ним в списки на зачисление. Списки подаются в учебно-методический отдел. На основании списков и служебной записки педагога формируется Приказ о зачислении. После получения из Департамента образования приказа о предоставлении данных по обучающимся создается база данных. В нее вводится информация о детях, которые были зачислены в соответствии с Приказом о зачислении.

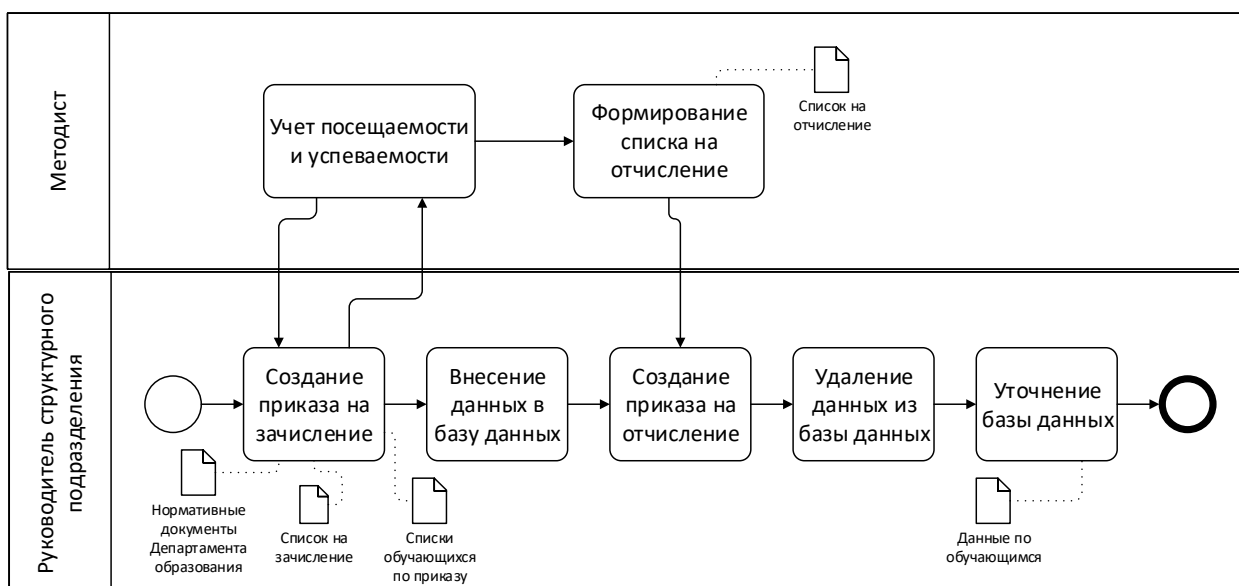


Рисунок 3 - Детализация процесса «Формирование сводной базы по обучающимся» (AS-IS)

В соответствии с Уставом Центра ребенок может быть отчислен из группы по ряду причин. Педагог или тренер-преподаватель в случае необходимости представляет списки на отчисление. В соответствии со списками формируется приказ об отчислении. На основании Приказа данные детей удаляются из базы. В случае удаления или ввода новых данных в базу выполняется проверка базы данных на наличие ошибок, уточняется содержимое базы данных.

Рассмотрим детализацию контекстной диаграммы процесса «Формирование учебного плана», представленную на рисунке 4. После идентификации обучающихся в базе данных поступает приказ о предоставлении данных по комплектованию отдельных групп. С учетом количества групп, а также учебных программ, расписаний занятий, поданных педагогами и тренерами-преподавателями, формируются учебные планы.

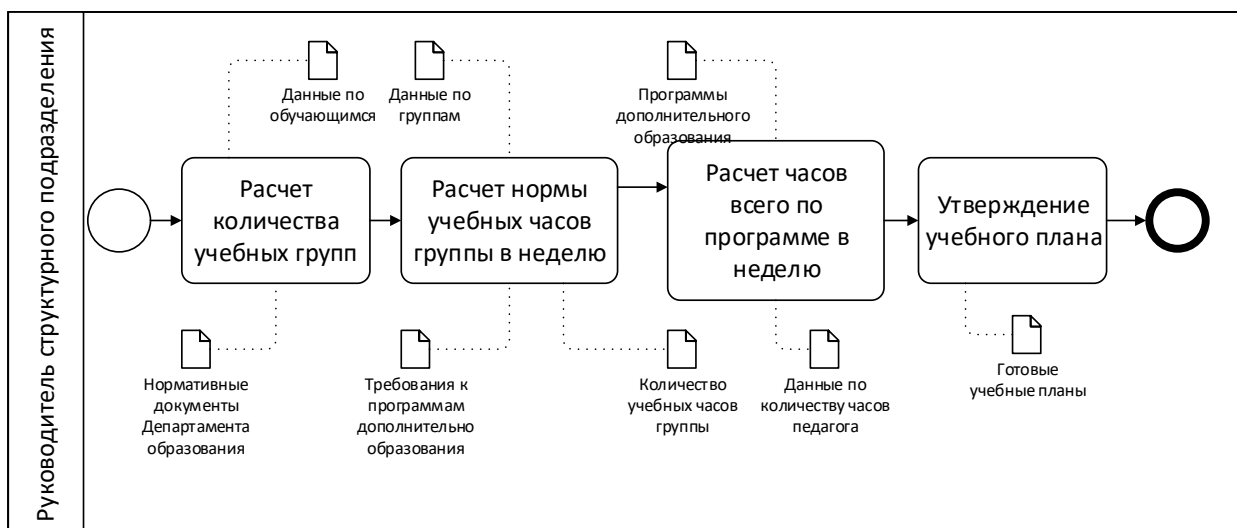


Рисунок 4 - Детализация процесса «Формирование учебного плана» (AS-IS)

Проанализируем детализацию контекстной диаграммы процесса «Организация внеурочных видов деятельности», представленную на рисунке 5. На основании планов мероприятий, положений или сценариев, а также

служебных записок педагогов и тренеров-преподавателей по отдельному мероприятию формируется сводный план мероприятий по месяцам. Сопоставляется расписание учебных занятий и график проведения мероприятий. Если необходимо, вносятся корректировки.

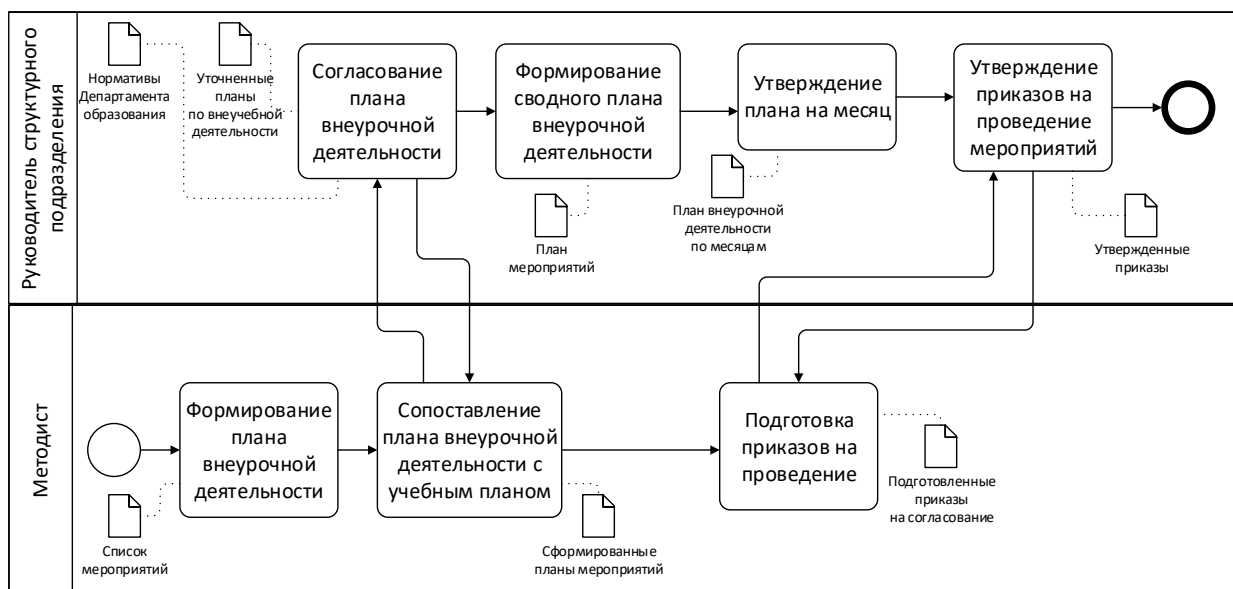


Рисунок 5 - Детализация контекстной диаграммы процесса «Организация внеурочных видов деятельности» (AS-IS)

Проанализировав этот блок, перейдем к следующей детализации контекстной диаграммы процесса «Формирование итоговых отчетов по деятельности», представленную на рисунке 6.

Сводные отчеты содержат собирательную информацию. Она поступает из отчетов педагогов и тренеров-преподавателей, журналов учета работы учебной группы, данных по обучающимся. Фактические показатели сравниваются с плановыми. Из имеющихся источников отбираются необходимые для анализа данных.

В соответствии с приказами отбираются данные для отчетов. С учетом параметров формируются списочные отчеты или осуществляется расчет количественных показателей для цифровых отчетов.

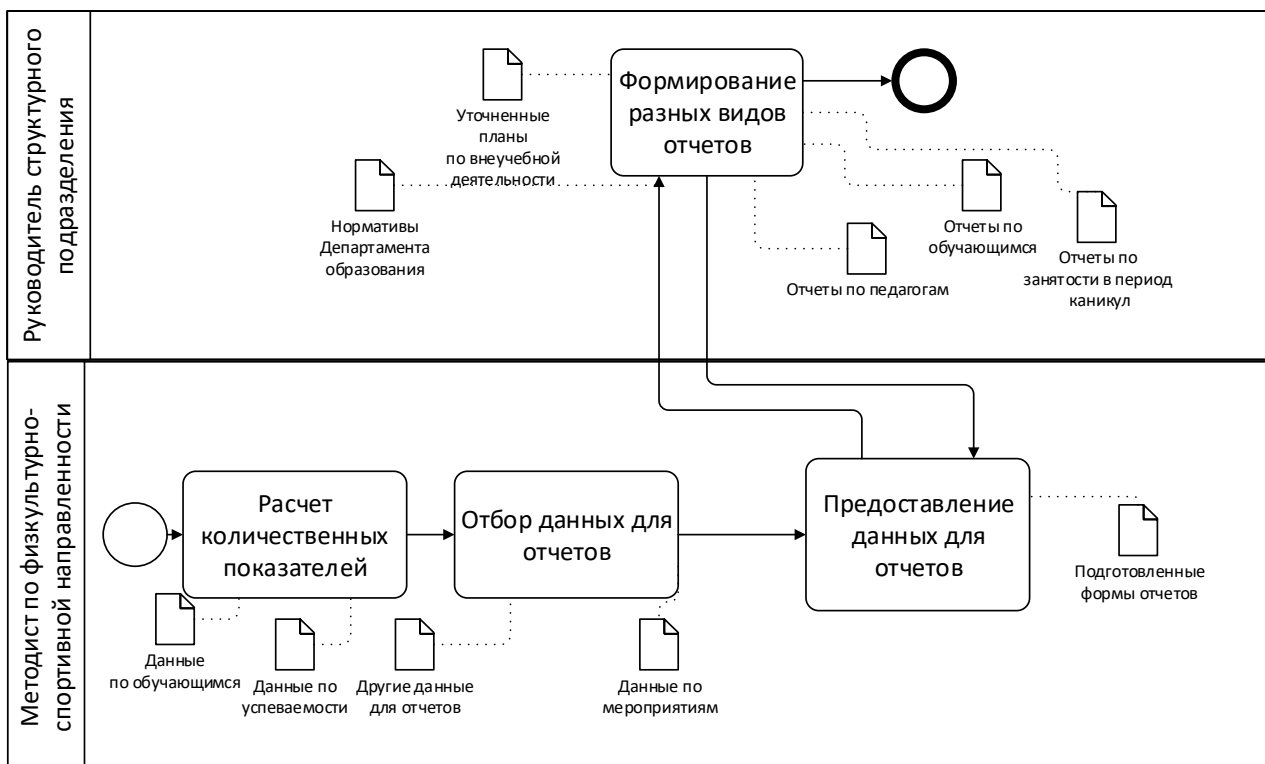


Рисунок 6 - Детализация контекстной диаграммы процесса «Формирование итоговых отчетов по спортивно-массовой работе» (AS-IS)

Существующая система документооборота не позволяет быстро и эффективно формировать сводные и статистические отчеты, поскольку эта работа в настоящее время предполагает ручную обработку информации. Автоматизация обеспечит сокращение времени на подготовку выходных документов, возможности появления ошибок в отчетах, наконец, позволит заместителю по учебно-воспитательной работе эффективнее организовать работу сотрудников отдела, позволит увеличить объем проектной деятельности методистов. Это, в конечном итоге, положительно повлияет на реализацию современных подходов в учебных программах дополнительного образования.

1.3 Постановка задачи на разработку

Учебно-методический отдел является одним из ключевых подразделений в Центре. Поэтому в отделе установлены современные средства компьютерной и оргтехники, в т.ч. три компьютера, один принтер, один ксерокс.

В отделе оборудовано три автоматизированных рабочих места:

- АРМ заместителя по учебно-воспитательной работе;
- АРМ методистов.

Каждое АРМ оснащено компьютером, подключенным к локальной сети.

Для обеспечения работы сотрудников отдела установлено следующее программное обеспечение:

- Операционная система Windows 8.
- Пакет офисных приложений Microsoft Office XP (MS Word, MS Excel, MS Access, MS PowerPoint, MS FrontPage).

Данный список является неполным, но включает основные программы, используемые сотрудниками учебно-методического отдела.

Большое количество рабочего времени сотрудники учебно-методического отдела занимаются ручной обработкой информации, содержащейся в списках на зачисление, на отчисление, служебных записках и отчетах педагогов по мероприятиям, ручному подсчету статистики, формированию отчетов. Это очень трудоемкая работа. Кроме того, существует высокая степень вероятности появления ошибок в статистических отчетах, поскольку вмешивается человеческий фактор.

Очевидно, что при наличии соответствующих программных продуктов, которые бы обеспечивали автоматизацию ввода, редактирования данных о деятельности сотрудников и возможность формирования выходных форм (отчетов), скорость обработки информации значительно бы повысилась. При этом появилась бы возможность обеспечить вывод на печать всех форм отчетов и сократить количество ошибок при подсчете статистических данных.

Наличие в учебно-методическом отделе современных персональных компьютеров и соответствующего программного обеспечения позволяет автоматизировать процессы обработки информации, сэкономив рабочее время заместителя по учебно-воспитательной работе и других сотрудников отдела, повысив достоверность выходной информации.

1.4 Анализ существующих разработок в области дополнительного образования

Рассмотрим некоторые существующие разработки, предоставляющие функционал для организаций дополнительного образования.

«Автоматизированная информационная система «КТЕ.Образование.Дополнительное образование» обеспечивает реализацию оказания в электронном виде муниципальной услуги «Прием заявлений на зачисление детей в учреждения дополнительного образования», входящую в перечень приоритетных в соответствии с распоряжением Правительства РФ №1993-р от 17.12.2009» [20].

Данная информационная система является коммерческой версией и предоставляет такие основные преимущества и функции, как:

- «прием электронного заявления на зачисление детей в учреждения дополнительного образования,
- ведение учета обучения и достижений детей, передача этих данных в смежные информационные системы, вывод услуг на РПГУ, ЕПГУ, использование в МФЦ,
- учет движения контингента учреждений дополнительного образования на основании данных приказов по контингенту,
- взаимодействие со смежными информационными системами по учету контингента образовательных учреждений в целях получения первичных данных о детях и их верификации,

- формирование аналитической и статистической отчетности любого уровня,
- ведение журнала юридически значимых событий по каждому заявлению, по каждому личному делу, и по системе в целом,
- интеграция с региональным сегментом единой федеральной межведомственной системы учета контингента обучающихся по основным образовательным программам и дополнительным общеобразовательным программам,
- функционирование на любой современной серверной платформе,
- использование платформо независимой СУБД с открытым кодом» [20].

«АИС «БАРС.Образование – Электронное дополнительное образование» предназначена для автоматизации и реализации оказания государственных и муниципальных услуг в сфере дополнительного образования в электронном виде.

Система предоставляет такие возможности, как:

- ведение данных по общеобразовательным организациям;
- ведение информации о сотрудниках;
- ведение информации об учащихя;
- ведение групп;
- ведение информации о родителях учащихя;
- ведение расписания занятий;
- ведение электронного журнала группы;
- импорт входных данных;
- предоставление выходных данных в печатной форме;
- ведение электронного дневника учащегося;
- предоставление отчетности деятельности ОДО» [21].

Сравнительная характеристика существующих разработок в области автоматизации дополнительного образования представлена в таблице 1.

Таблица 1 – Сравнительная характеристика существующих разработок

Критерии	КТЕ.Образование. Дополнительное образование	АИС «БАРС.Образование – Электронное дополнительное образование	Проектный вариант
Хранение данных обучающихся	+	+	+
Хранение данных образовательных программах	-	+	+
Ведение электронного журнала группы	+	+	+
Составление отчетов о деятельности	+	+	+
Удобный интерфейс	-	-	+
Наличие бесплатной версии	-	-	+
Итого	3 из 6	4 из 6	6 из 6

Таким образом, анализ существующих разработок помог определить основные критерии и функциональные возможности, которые будут положены в основу реализации приложения для автоматизации деятельности учебно-методического отдела.

1.5 Разработка модели бизнес-процесса «КАК БУДЕТ»

На диаграмме, отражающей основную функцию информационной системы учебно-методического отдела, согласно проекту, внесены изменения по сравнению с базовым вариантом. Контекстная диаграмма основной функции представлена на рисунке 7.

Вместо бланков документов (справок, отчетов, личных карточек) на бумажных носителях появились электронные формы документов. В качестве важнейшего механизма реализации функции отдела наряду с сотрудниками отдела выступает приложение, позволяющее автоматизировать ввод и редактирование данных в базу данных, а также обеспечить эффективное формирование установленных отчетов. В проектном варианте также

отсутствует функция проверки занятий, мероприятий и учебной документации, так как этот этап не автоматизируется [11].

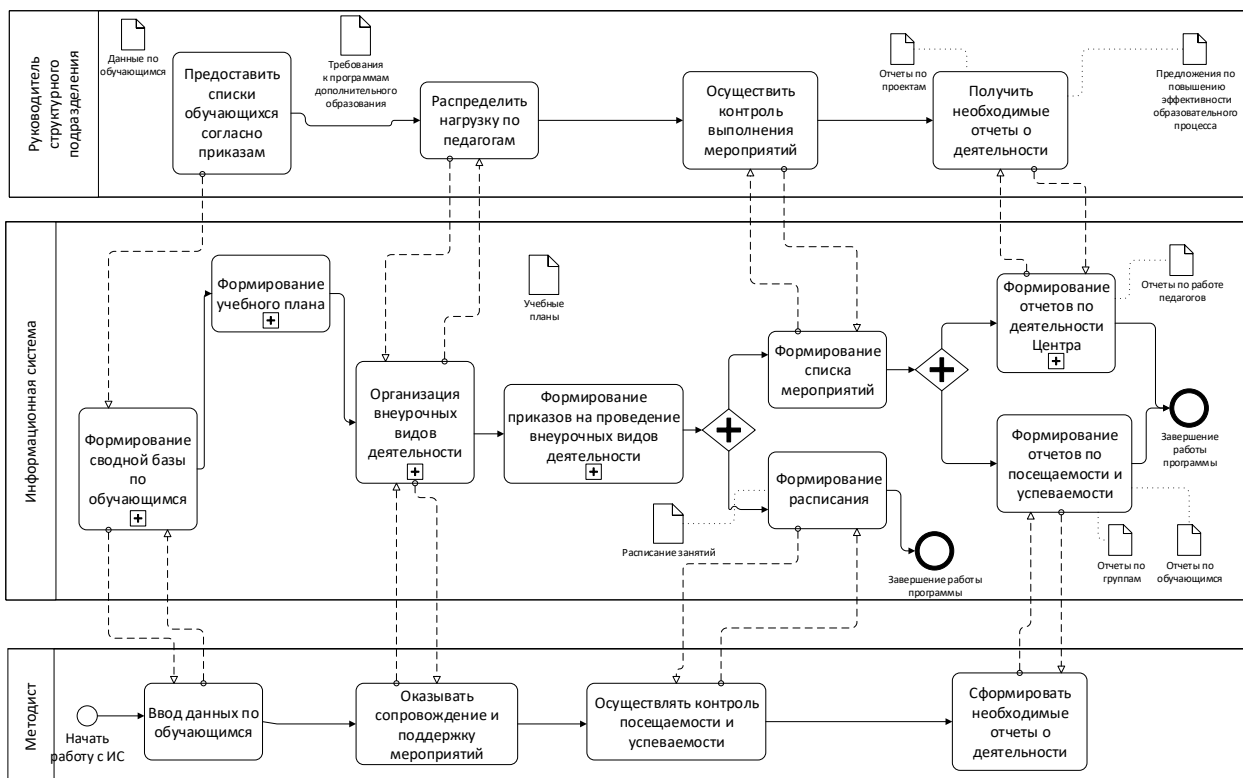


Рисунок 7 - Контекстная диаграмма процесса «Работа информационной системы учебно-методического отдела (ТО-ВЕ)»

Центральное место при автоматизированной обработке данных педагогов дополнительного образования и тренеров-преподавателей занимает электронная база данных [7]. Сотрудники учебно-методического отдела вводят информацию в пользовательские формы на основе списков обучающихся на зачисление и на отчисление, программ, планов мероприятий, положений и сценариев, отчетов педагогов дополнительного образования и т. д. После того, как электронная база данных находится в актуальном состоянии, пользователи, т.е. сотрудники учебно-методического отдела, могут формировать необходимые отчеты, применяя специальное приложение, автоматизирующие деятельность отдела [8]. На выходе получают такие документы, как электронная база данных воспитанников, учебный план,

отчеты, которые могут быть представлены как на электронных, так и на бумажных носителях.

При автоматизированной обработке данных сотрудниками учебно-методического отдела всю работу по сортировке и расчету данных для формирования учебного плана берет на себя автоматизированная система. Детализация контекстной диаграммы процесса «Формирование учебного плана» представлена на рисунке 8.

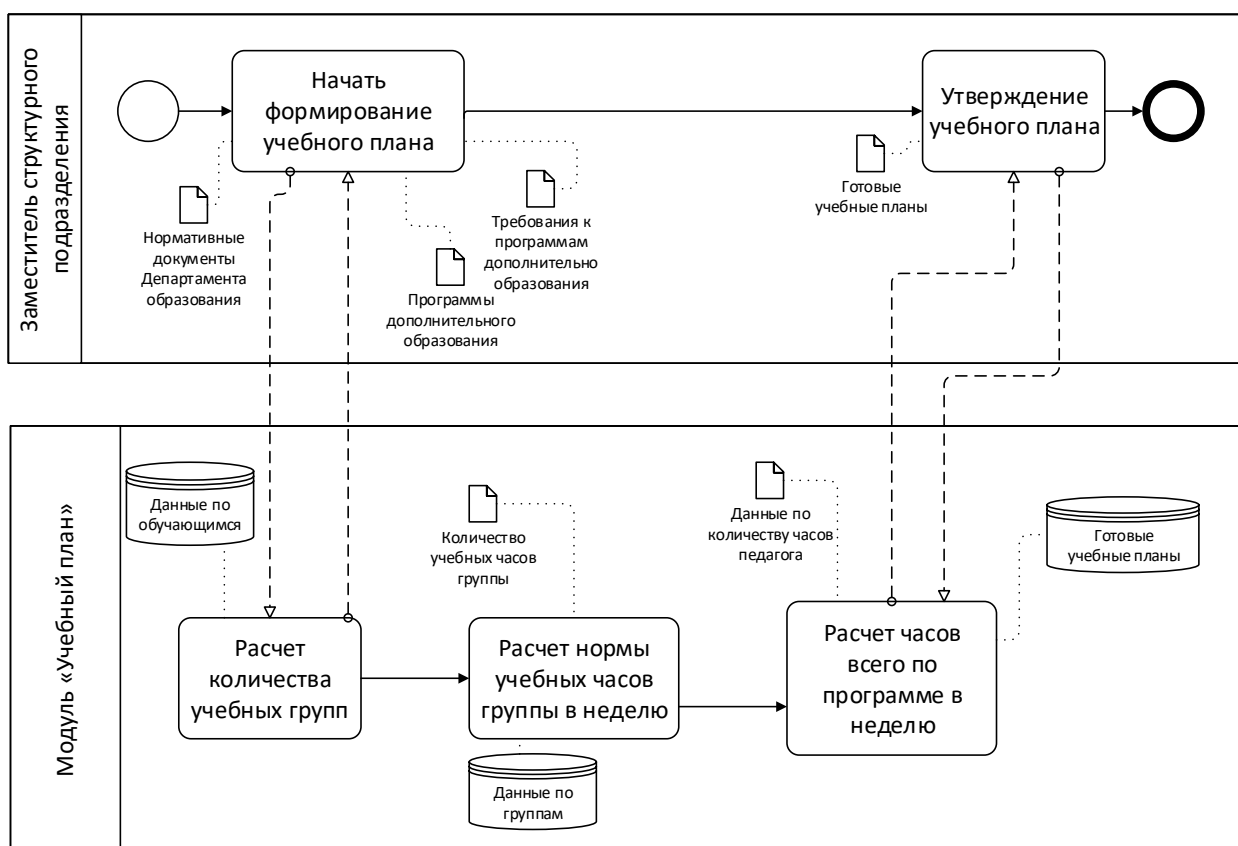


Рисунок 8 - Детализация контекстной диаграммы процесса «Формирование учебного плана» (ТО-ВЕ)

После ввода данных по учебным группам по запросу пользователя информационная система отбирает информацию о группах, которая содержится в базе данных. Осуществляется расчет общего количества учебных групп, а также количество по направленностям, программам и годам обучения. Определяется количество учебных часов в каждой из групп и

количество воспитанников. Данные суммируются по программам и годам обучения. После подготовки учебного плана его можно просмотреть в электронном виде, отформатировать или вывести на печать.

При планировании внеурочных видов деятельности после ввода данных из планов мероприятий, полученных от педагогов дополнительного образования и тренеров всю работу по обработке данных, содержащихся в базе, также выполняет ПК.

Как видно на рисунке 9, после получения запроса от пользователя формируется готовый план, который пользователь при необходимости выводит на печать.

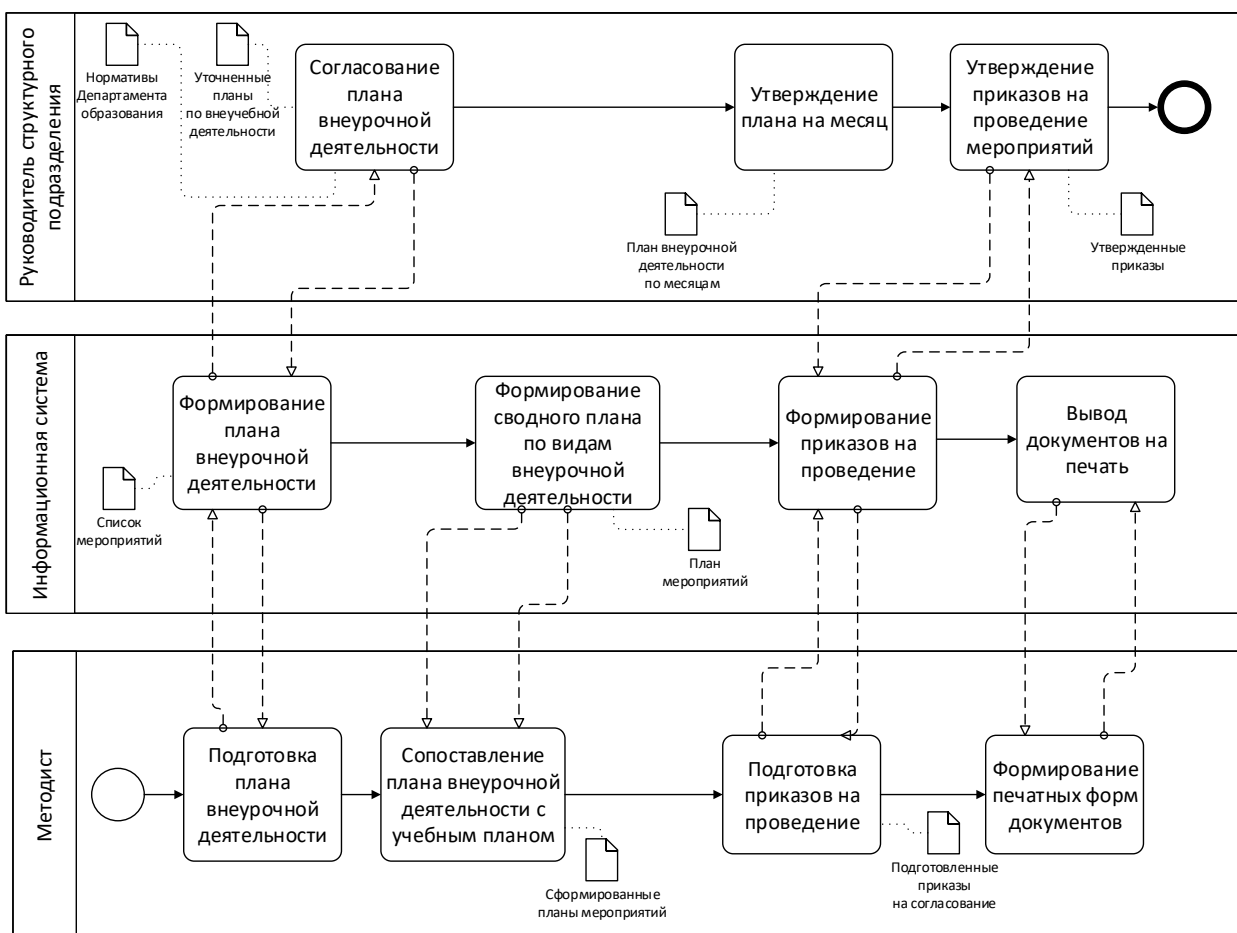


Рисунок 9 - Детализация контекстной диаграммы процесса «Организация внеурочных видов деятельности» (ТО-ВЕ)

Формирование итоговых отчетов по деятельности также осуществляется в основном силами информационной системы. Детализация диаграммы процесса представлена на рисунке 10. Для формирования итоговых отчетов пользователь задает информационной системе параметры отчетов. В соответствии с указанными параметрами осуществляется запрос к базе данных, из которой отбирается необходимая информация.

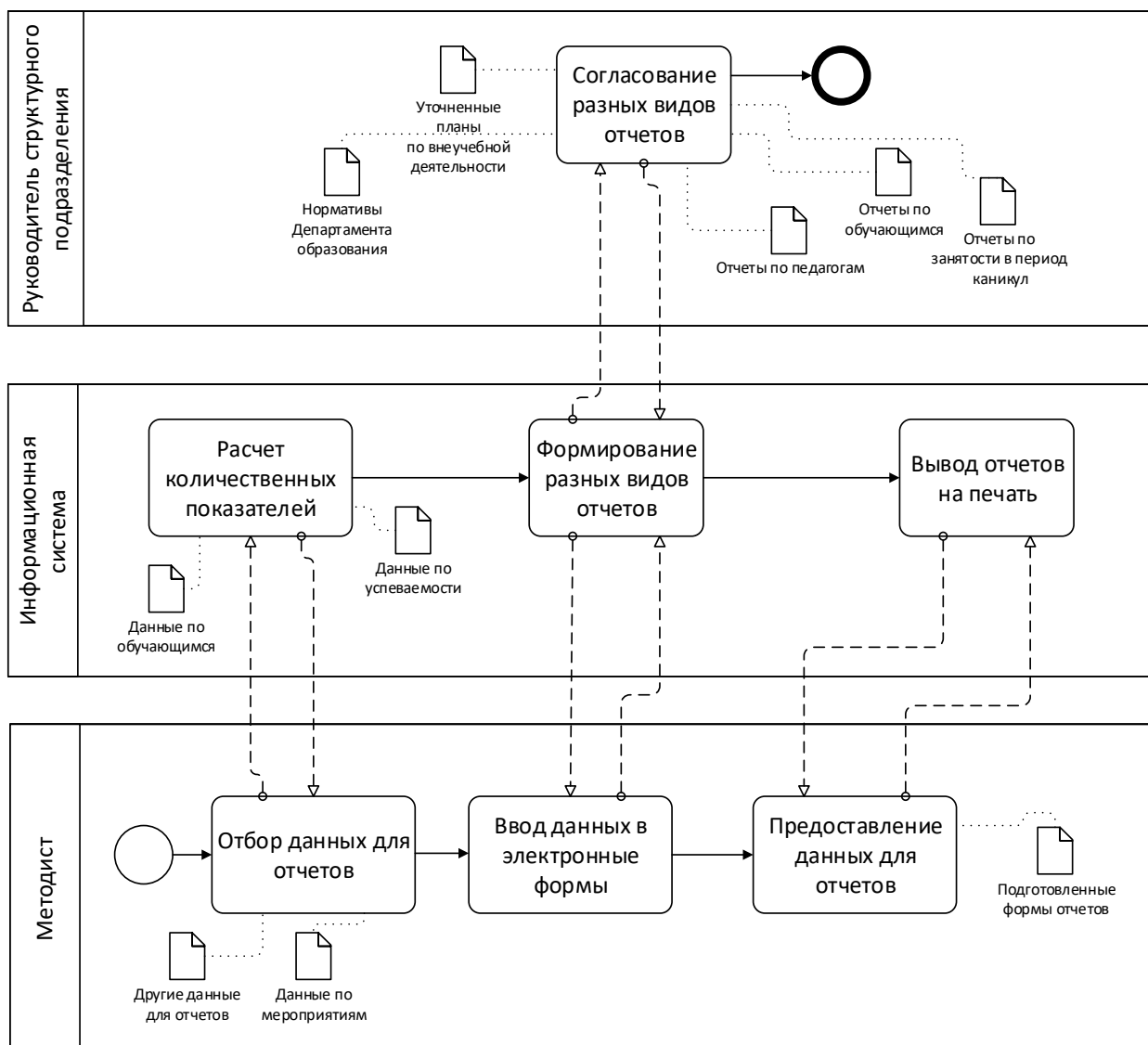


Рисунок 10 - Детализация контекстной диаграммы процесса «Формирование итоговых отчетов по деятельности» (ТО-ВЕ)

Система автоматически заполняет электронные формы рассчитанными количественными показателями или списочными данными, которые пользователь впоследствии выводит на печать.

Таким образом, были подробно рассмотрены основные бизнес-процессы учебно-методического отдела с использованием информационной системы.

1.6 Требования к разрабатываемой информационной системе

Программные модули информационной системы должны соответствовать бизнес-процессам, а функции автоматизированных рабочих мест – должностным обязанностям сотрудников. При выборе программно-аппаратных платформ и отдельных приложений должны применяться непротиворечивые, согласующиеся технологии, соблюдаться единая технология эксплуатации и обслуживания системы [14].

Помимо этих ключевых требований есть ряд общих технических требований для любой информационной системы:

- быстродействие, т.е. достаточно малое время реакции системы (единицы секунды) при вводе, поиске и обработке информации;
- надёжная защита от несанкционированного доступа к данным и регистрации действий персонала;
- удобный пользовательский интерфейс рабочих мест;
- возможность масштабирования и развития системы;
- интеграция с модулями системы передачи данных;
- возможность конвертации данных из использовавшихся в прошлом приложений в новую систему;
- высокая надёжность работы.

Функциональные требования к информационной системе, которые описываются, в том числе, и с помощью моделей процессов и структур данных, являются только частью общих требований. Требования к информационной системе могут включать также:

- требования к функциональным характеристикам;
- требования к надежности;
- требования к эксплуатации;
- требования к информационной и программной совместимости;
- требования к документации.

Требования к функциональным характеристикам указывают требования к составу выполняемых функций, организации входных и выходных данных. При выборе между объектными и структурными методами следует использовать принцип концептуальной общности, который предполагает следование единой философии на всех этапах жизненного цикла. Если предполагается использовать структурное программирование, то и на этапе анализа следует использовать структурный подход, а в случае использования объектно-ориентированных языков разработки - объектный анализ и объектное проектирование. При необходимости структурный и объектный подходы могут использоваться одновременно.

Требования к надежности определяют требования к обеспечению надежного функционирования: контроль входной и выходной информации, время и механизмы восстановления после программных и аппаратных отказов, а также описывают организацию системы безопасности, включая подсистемы контроля доступа, шифрования и т. п.

Формулирование условий эксплуатации предполагает описание необходимого обслуживания, которое требуется для работы системы, например создание резервных копий, реиндексирование баз и т. п., а также требования к квалификации персонала (пользователей и обслуживающего персонала).

Требования к составу и параметрам технических средств указывают необходимый состав технических средств с указанием их основных технических характеристик. Могут указываться требования к помещениям, в которых будет находиться оборудование. В этом разделе указываются требования к переносимости системы.

Требования к информационной и программной совместимости – это требования к информационным структурам на входе и выходе, методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой.

Требования к программной документации предполагают указание предварительного состава программной документации, и при необходимости, специальных требований к ней.

Следующим важным пунктом требований к информационной системе является степень охвата направлений деятельности, характерных для данного подразделения. Следует оценить, насколько глубоко автоматизированы процессы деятельности учебно-методического отдела, насколько информационная система обеспечивает удобный ввод, обработку и представление информации.

Следует особо остановиться и на соответствии масштабов информационной системы масштабам задач, актуальных для конкретного подразделения. Информационная система для учебно-методического отдела не должна быть слишком большой, включая лишние функции, однако должна в полной мере обеспечивать автоматизацию основных процессов в подразделении, включая ведение автоматизированного учета результативности деятельности педагогов и тренеров-преподавателей, формирование установленных отчетов, обеспечение быстрого поиска информации из базы данных.

Исходя из рассмотренных выше общих требований к информационной системе, были сформулированы требования к системе автоматизации деятельности учебно-методического отдела. Информационная система должна:

- быть быстродействующей, т.е. достаточно малое время реакции системы (единицы секунды) при вводе, поиске и обработке информации;
- обеспечивать надёжную защиту от несанкционированного доступа к данным и неправильных действий персонала;

- включать удобный пользовательский интерфейс приложений;
- обеспечивать возможность масштабирования и развития системы;
- соблюдать требования к функциональным характеристикам, т.е. к составу выполняемых функций, организации входных и выходных данных (формирование приказов об отчислении и зачислении обучающихся, автоматическая подготовка отчетов установленных форм, критериев деятельности руководителей, обеспечение поиска информации о детях из списка обучающихся по направленности, по группе, к которой он относится);
- обеспечивать надежность работы (все возможные ошибки пользователя должны быть предусмотрены и предупреждены, система при сбое не должна повреждать данные, хранящиеся в базе данных);
- включать документацию, поясняющую пользователям порядок работы с приложениями.

Степень соответствия разработанной информационной системы (ИС) указанным требованиям можно частично оценить, проведя опрос пользователей системы. На основании рекомендаций пользователей вносятся коррективы в ИС.

Выводы по первой главе

Были определены основные бизнес-процессы учебно-методического отдела центра дополнительного образования детей. Выделены недостатки существующего процесса, сделан вывод о необходимости автоматизации ряда функций по организации внеучебной деятельности.

Рассмотрены программные продукты, предназначенные для автоматизации дополнительного образования, рассмотрены их достоинства и недостатки, сделан вывод о необходимости разработки собственного ИТ решения.

Глава 2 Разработка приложения для учебного отдела центра дополнительного образования

2.1 Концептуальное проектирование приложения для учебно-методического отдела

Для автоматизации деятельности учебно-методического отдела наиболее эффективным представляется разработка собственного программного комплекса. Это позволит с минимальными затратами создать программу, отвечающую требованиям сотрудников учебно-методического отдела, обеспечивающую эффективный учет результативности деятельности педагогов дополнительного образования, а также формирование отчетов.

На рисунке 11 представлена диаграмма вариантов использования информационной системы сотрудниками учебного отдела.

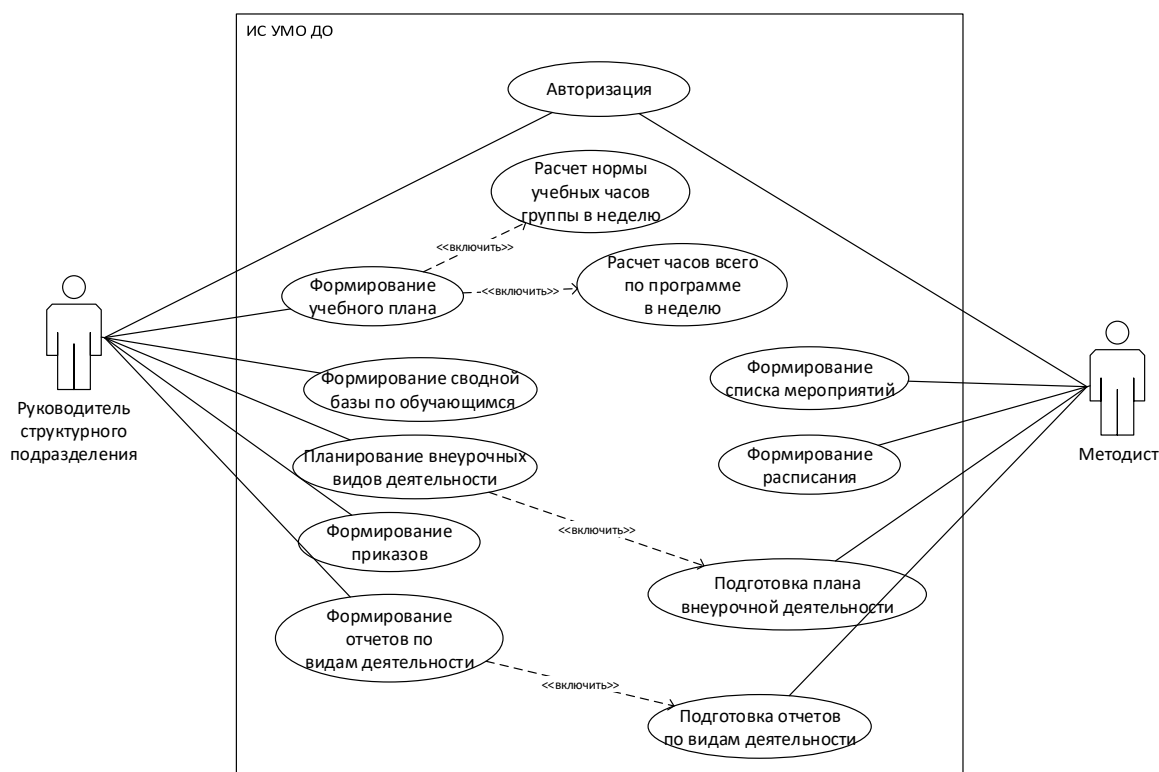


Рисунок 11 – Диаграмма вариантов использования информационной системы сотрудниками учебного отдела

На диаграмме присутствуют две роли: руководитель структурного подразделения и методист. В зависимости от роли после авторизации в системе пользователю предоставляются различные функциональные возможности.

Представленная диаграмма отображает основные функции и пользователей информационной системы для автоматизации основных видов деятельности учебно-методического отдела.

Для реализации приложения была выбрана трехзвенная архитектура.

Трехуровневая архитектура представляет собой средний уровень, который позволяет снизить нагрузку на обработку данных на двух других концах. Созданные на нескольких логических уровнях или уровнях, эти уровни не обязательно соответствуют физическим расположениям на различных компьютерах в сети, а скорее логическим уровням приложения.

В простейшей конфигурации физически сервер приложений может быть совмещён с сервером базы данных на одном компьютере, к которому по сети подключается один или несколько терминалов (рисунок 12).

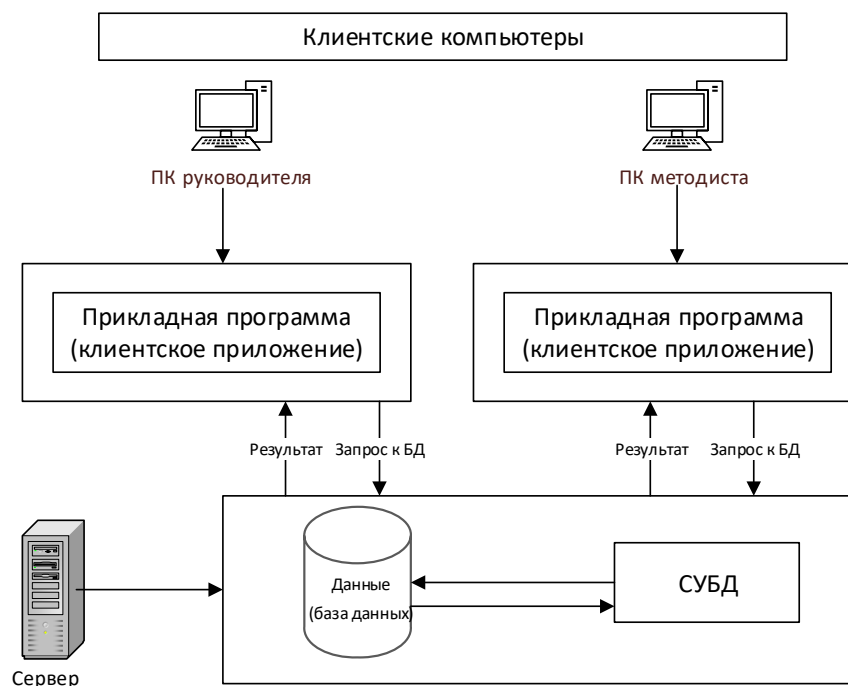


Рисунок 12 – Архитектура «клиент-сервер»

Трехуровневая архитектура состоит из клиентов, серверов приложений и баз данных. Серверы приложений на среднем уровне, состоящие из проверок данных и обеспечения соблюдения бизнес-правил и правил обработки данных, реализуют большинство бизнес-сервисов. Например, этот уровень может просто открывать соединение с базой данных, вызывать запросы SQL и возвращать результаты клиентам [4].

Принцип интеграции предполагает организацию хранения информации в виде базы данных (БД), где все данные собраны в едином интегрированном хранилище и к информации, как важнейшему ресурсу, обеспечен защищенный доступ различных пользователей [1].

Разработка приложения будет осуществляться средствами Qt. Qt мощный и стабильный комплект разработки как для разработчиков программного обеспечения, так и для системных инженеров. Qt используется инженерами и разработчиками программного обеспечения более двух десятилетий для создания кроссплатформенных приложений. Qt многофункциональный и поддерживает большое количество платформ. Отлично подходит не только для разработки настольных приложений, но и для мобильных и встраиваемых систем.

«Qt имеет огромный пакет SDK (комплект для разработки программного обеспечения), который содержит широкий спектр инструментов и библиотек, помогающих разработчикам выполнять свою работу, не беспокоясь о технических проблемах, связанных с конкретной платформой. Qt решает множество проблем с интеграцией и совместимостью. Это не только повышает эффективность, но и снижает затраты на разработку, особенно при разработке кросс-платформенных приложений, рассчитанных на более широкий круг пользователей» [19]. Одно из самых важных преимуществ Qt среди всех других существующих на рынке SDK, это кроссплатформенность. Этот комплект разработки поддерживает большое количество платформ без написания разных наборов кода для каждой платформы [6].

«Qt обычно использует C++, компилируемый язык, генерирующий небольшой и эффективный код. Он также хорошо документирован и следует очень последовательному набору соглашений об именах, что сокращает траекторию обучения для разработчика. Следует иметь в виду, что Qt включает небольшое количество функций, которые работают только на определенных платформах. Однако они минимальны и часто предназначены для особых случаев использования, таких как датчики Qt, которые работают только на мобильных платформах, Qt Web Engine, который работает только на десктопах, Qt NFC, только для Android и Linux, и так далее. Это некоторые очень специфические функции, которые существуют только на определенных платформах, которые их поддерживают. Помимо этого, общие функции обычно поддерживаются на всех платформах» [19].

MySQL 5 поддерживает хранимые процедуры на уровне SQL, но не имеет API для управления параметрами IN, OUT и INOUT. Поэтому параметры необходимо задавать и считывать с помощью команд SQL, а не `QSqlQuery::bindValue()`. Встроенный сервер MySQL является простой заменой обычной клиентской библиотеки. Со встроенным сервером MySQL сервер MySQL не требуется для использования функций MySQL. Чтобы использовать встроенный сервер MySQL, достаточно связать подключаемый модуль Qt `libmysqldc libmysqlclient`. Это можно сделать, добавив `-DMySQL_LIBRARY=<path/to/mysqld/>libmysqld.<so|lib|dylib>` в командную строку `configure`.

2.2 Логическое и физическое моделирование деятельности учебно-методического отдела

Создание эффективной и надежной базы данных это основа для стабильной и долговечной системы сбора и хранения данных [9]. Система реляционных баз данных имеет простую логическую структуру с прочной теоретической основой. Реляционная модель основана на основной концепции

отношений. В реляционной модели все данные логически структурированы в рамках отношений [10]. Модель базы данных определяет логическую структуру базы данных и способ хранения, организации, просмотра, представления и обработки данных, а также как данные могут быть взаимосвязаны. Таким образом, моделирование базы данных - это процесс выбора структуры и создания дизайна и схемы, которые будут определять, как строится фактическая база данных [15].

Для проектирования базы данных сотрудников использовалась модель «сущность – связь» (Entity – Relationship model, или ER – модель). ER – модель представляет собой высокоуровневую концептуальную модель данных, цель которой - упрощение задачи проектирования баз данных [16]. Данная модель данных включает в себя набор концепций, которые описывают структуру базы данных и связанные с ней транзакции обновления и выборки данных. Следует подчеркнуть, что концептуальная модель данных не зависит от конкретной СУБД или аппаратной платформы, которая используется для физической реализации базы данных [17].

На рисунке 13 отображена логическая модель базы данных учебно-методического отдела, представляющая собой совокупность взаимосвязанных сущностей и их атрибутов [18].

Разработанная логическая модель базы данных сотрудников позволяет реализовать следующие транзакции на выборку и изменение данных:

- получить общую информацию о любом воспитаннике центра, определенную в атрибутах сущности «Обучающиеся»;
- получить список всех групп центра и программ, по которым они занимаются с указанием педагога, работающего в данной группе;
- получить список воспитанников любой учебной группы Центра;
- определить время и место проведения занятий группы для организации контроля ее деятельности;
- определить какие внеурочные виды работы использует педагог дополнительного образования или тренер-преподаватель;

- получить список мероприятий для подготовки приказов на их проведение, проверить наличие отчета педагога по мероприятию;
- изменить данные педагогов, воспитанников, программ, групп, мероприятий, проектов, форм работы с родителями, добавить новые данные или удалить существующие;
- получить сведения о нормативах для групп различного года и этапа обучения, занимающегося по определенной программе;
- отследить дату утверждения дополнительных образовательных программ;
- подготовить необходимые отчеты с помощью запросов к базе данных и вывести их на печать.

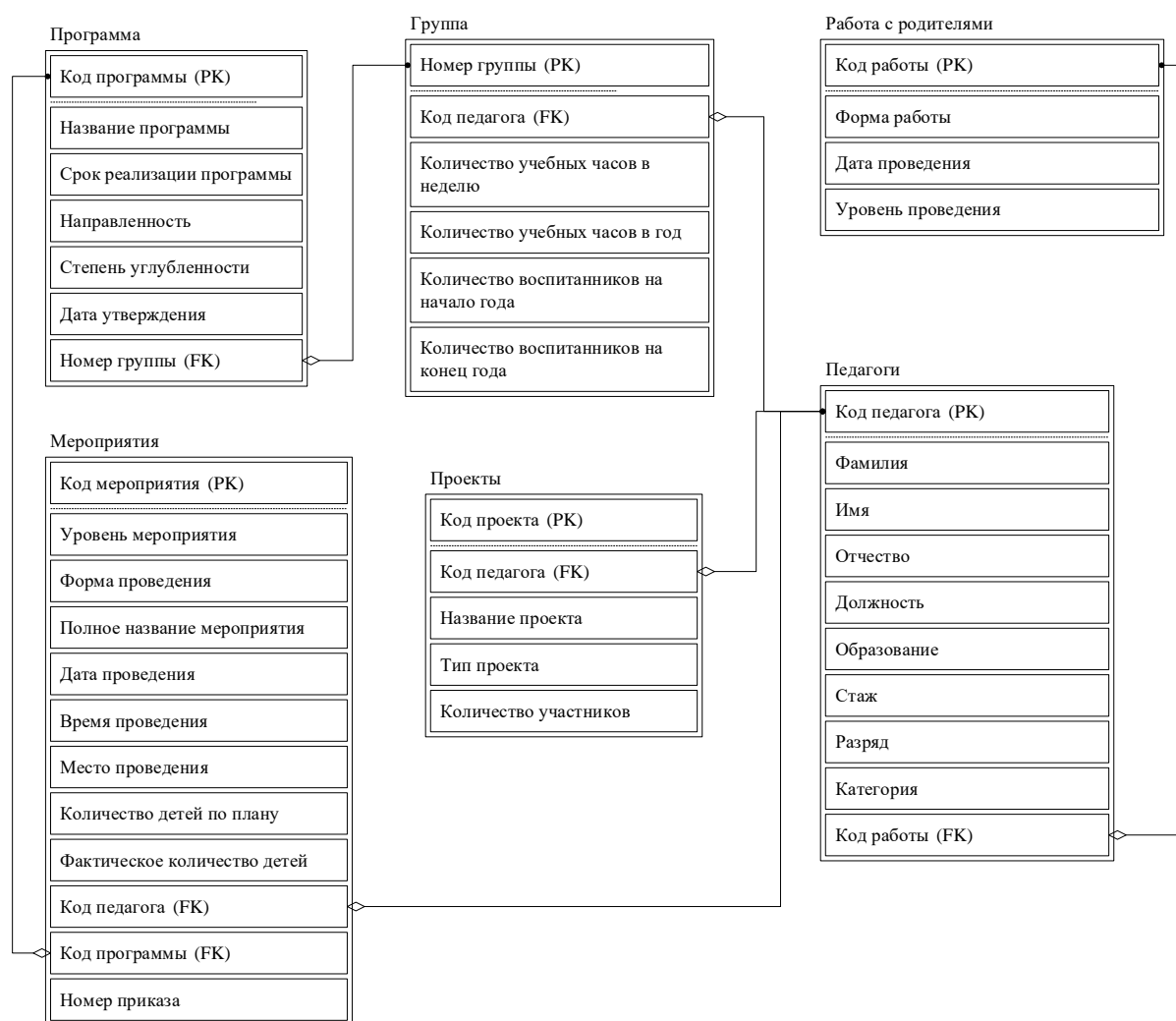


Рисунок 13 - Логическая модель данных

Физическая модель базы данных учебно-методического отдела составлена с учетом логической модели (рисунок 14).

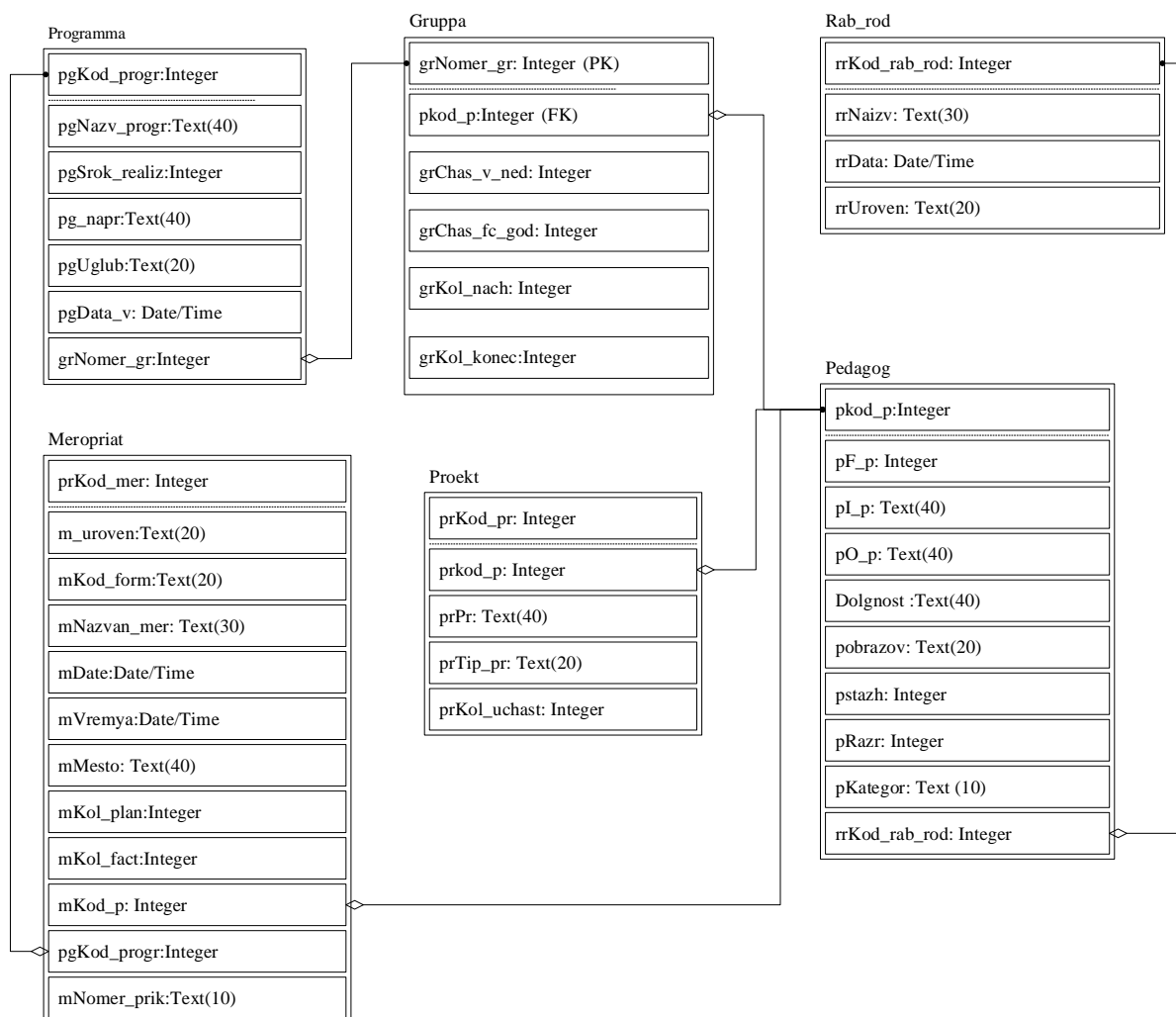


Рисунок 14 - Физическая модель данных

В каждой таблице отмечен первичный ключ. Между таблицами базы данных установлена связь «один-ко-многим».

Схема физической модели базы данных представляет, как данные хранятся в дисковом хранилище или целевом объекте данных. Это самая низкая форма абстракции по отношению к схеме, которая действует как основа для других типов схем для создания отношений и индексов.

Таблица Pedagog представляет собой сущность Педагог логической модели базы данных, содержит данные о педагогах ДО и тренерах-

преподавателях. Имеет следующие подчиненные таблицы: Proekt, Rab_rod, Meropriat, Gruppa, главных нет. Структура таблицы Pedagog представлена в таблице 2. Знаком «*» отмечены ключевые поля.

Таблица 2 - Структура таблицы Pedagog

Наименование	Описание	Тип данных
pkod_p*	Содержит код педагога	Integer
pFIO_p	Содержит фамилию и инициалы педагога	Text(40)
pobrazov	Данные об образовании педагога	Text(20)
pstazh	Число лет стажа педагога	Integer
pRazr	Разряд ЕТС педагога	Integer
pKategor	Данные о категории педагога	Text (10)
pDolzhn	Точное наименование должности	Text(40)
pMesto	Данные о месте работы педагога	Text(40)

Таблица Proekt представляет собой сущность Проект, содержит данные о проектах педагогов дополнительного образования и тренеров-преподавателей. Главная таблица Pedagog, подчиненных нет. Структура таблицы представлена в таблице 3.

Таблица 3 - Структура таблицы Proekt

Наименование	Описание	Тип данных
prKod_pr*	Содержит код проекта	Integer
prkod_p	Содержит код педагога, используется для связи с главной таблицей	Integer
prPr	Название проекта педагога	Text(40)
prTip_pr	Тип проекта	Text(20)
prKol_uchast	Количество детей, участвующих в проекте	Integer

Таблица Rab_rod представляет собой сущность Работа с родителями, содержит данные о работе с родителями педагогов дополнительного образования и тренеров-преподавателей. Главная таблица Pedagog, подчиненных нет. Структура таблицы представлена в таблице 4.

Таблица 4 - Структура таблицы Rab_rod

Наименование	Описание	Тип данных
rrKod_rab_rod*	Содержит код работы с родителями	Integer
rrkod_p	Содержит код педагога, используется для связи с главной таблицей	Integer
rrNaizv	Название формы работы с родителями	Text(30)
rrData	Дата проведения формы работы	Date/Time
rrUroven	Уровень проведения (группа, объединение)	Text(20)

Таблица Мероприат представляет собой сущность Мероприятие, содержит данные о мероприятиях педагогов ДО и тренеров-преподавателей. Главные таблицы Pedagog, Forma, Urov_meropr, подчиненных нет. Структура представлена в таблице 5.

Таблица 5 - Структура таблицы Мероприат

Наименование	Описание	Тип данных
prKod_mer*	Содержит код мероприятия	Integer
mKod_uroven	Содержит код уровня, используется для связи с главной таблицей Urov_meropr	Integer
mKod_form	Код формы, используется для связи с главной таблицей Forma	Integer
mKod_p	Код педагога, используется для связи с главной таблицей Pedagog	Integer
mNazvan_mer	Полное название мероприятия	Text(30)
mVremya	Дата и время проведения мероприятия	Date/Time
mMesto	Школа и кабинет проведения мероприятия	Text(40)
mKol_plan	Количество детей по плану	Integer
mKol_fact	Фактическое количество детей	Integer
mOsnoy	Основание для проведения (положение, сценарий)	Text(20)
mNomer_prik	Номер приказа	Text(10)

Таблица Программа представляет собой сущность Программа, содержит данные об учебных программах. Структура представлена в таблице 6.

Таблица 6 - Структура таблицы Programma

Наименование	Описание	Тип данных
pgKod_progr*	Содержит код программы	Integer
pgNazv_progr	Содержит полное название программы	Text(40)
pgSrok_realiz	Срок реализации программы, лет	Integer
pgKod_napr	Направленность программы	Integer
pgUglub	Степень углубленности изучения	Text(20)
pgData_v	Дата утверждения программы	Date/Time

Таблица Gruppa представляет собой сущность Группа, содержит данные об учебных группах. Структура представлена в таблице 7.

Таблица 7 - Структура таблицы Gruppa

Наименование	Описание	Тип данных
grNomer_gr*	Содержит номер группы	Integer
grKod_p	Код педагога, используется для связи с главной таблицей Pedagog	Integer
grChas_v_ned	Количество учебных часов в неделю по учебному плану	Integer
grChas_fc_god	Фактическое количество учебных часов в год	Integer
grKol_nach	Количество воспитанников на начало года	Integer
grKol_konec	Количество воспитанников на конец года	Integer

Для отображения данных были созданы следующие SQL-запросы. Представление «Все воспитанники» - обOsp представлено на рисунке 15.

```

1 SELECT Obuch.obkod_obuch AS Код, Obuch.obfamily AS Фамилия, Obuch.obImia AS Имя, Obuch.obotchestvo
2 AS Отчество, Obuch.obdata_rozden AS [Дата рождения], Obuch.obtip_doc
3 AS [Тип документа], Obuch.obser_doc AS Серия, Obuch.obnomer_doc AS Номер, klass.klGod_obuch
4 AS Параллель, klass.klLitera AS Литера, Tip_OU.tSokr_tip AS [Тип ОУ], OU.oNomer_OU
5 AS [№ ОУ], Obuch.obAdres AS Адрес, Obuch.obTelefon AS Телефон, Obuch.obFIO_rod
6 AS [ФИО родителей], Obuch.obPol AS Пол, Obuch.obData_med
7 AS [Дата медосмотра], Obuch.obGr_risk AS [Группа риска]
8 FROM Tip_OU INNER JOIN (OU INNER JOIN (klass INNER JOIN Obuch
9 ON klass.klKod_Klass = Obuch.obkod_klass)
10 ON OU.oNomer_OU = klass.klNomer_OU) ON Tip_OU.tSokr_tip = OU.oSokr_tip
11 ORDER BY Obuch.obfamily

```

Рисунок 15 – SQL-запрос на создание таблицы «Все воспитанники»

Представление «Все группы» - grOsp представлено на рисунке 16.

```
1 SELECT Gruppya.grNomer_gr AS [Номер группы], Napravl.nNazv_napravl
2 AS Направленность, Programma.pgNazv_progr AS Программа, pedagog.pFIO_p
3 AS Педагог, God_obuch.goGod_obuch AS [Год обучения], Etap_obuch.etSokr_etap
4 AS Этап, Gruppya.grChas_v_ned AS [Количество часов]
5 FROM (Napravl INNER JOIN Programma ON Napravl.nKod_napravl = Programma.pgKod_napr)
6 INNER JOIN (pedagog INNER JOIN ((God_obuch INNER JOIN (Etap_obuch
7 INNER JOIN Normativ ON Etap_obuch.etSokr_etap = Normativ.nmSokr_etap)
8 ON God_obuch.goGod_obuch = Normativ.nmGod_obuch)
9 INNER JOIN Gruppya ON Normativ.nmKod_norm = Gruppya.grKod_norm)
10 ON pedagog.pKod_p = Gruppya.grKod_p) ON Programma.pgKod_progr = Normativ.nmKod_progr
11 ORDER BY Gruppya.grNomer_gr
```

Рисунок 16 - SQL-запрос на создание таблицы «Все группы»

Представление «Учащиеся группы» представлено на рисунке 17.

```
1 SELECT Uch_v_gr.ugKod_uch_v_gr AS [№ ЛКУ], Obuch.obfamily
2 AS Фамилия, Obuch.obImia AS Имя, Obuch.obotchestvo AS Отчество, Obuch.obdata_rozden
3 AS [Дата рождения], Obuch.obtip_doc AS Тип, Obuch.obser_doc
4 AS Серия, Obuch.obnomer_doc AS Номер, Paralel.Paralel_kl AS Параллель, Litera.litKlass
5 AS Класс, OU.oNomer_OU AS [Номер ОУ], Gruppya.grNomer_gr AS Группа
6 FROM Tip_OU INNER JOIN (Paralel INNER JOIN (OU INNER JOIN ((Litera
7 INNER JOIN Obuch ON Litera.litKod = Obuch.obKod_liter)
8 INNER JOIN (Gruppya INNER JOIN Uch_v_gr ON Gruppya.grNomer_gr = Uch_v_gr.ugNomer_gr)
9 ON Obuch.obkod_obuch = Uch_v_gr.ugKod_obuch) ON OU.oNomer_OU = Obuch.obNomer_OU)
10 ON Paralel.Paralel_kl = Obuch.obParal) ON Tip_OU.tSokr_tip = OU.oSokr_tip
11 ORDER BY Obuch.obfamily
```

Рисунок 17 - SQL-запрос на создание таблицы «Учащиеся группы»

Представление «Педагоги ДО и тренеры-преподаватели» представлено на рисунке 18.

```

1 SELECT pedagog.pFIO_p AS ФИО, pedagog.pDolzhn AS Должность, pedagog.pObrazov
2 AS Образование, pedagog.pStazh AS Стаж, pedagog.pRazr AS Разряд, pedagog.pKategor
3 AS Категория, pedagog.pMesto AS Место, pedagog.pKod_p, COUNT(Gruppa.grNomer_gr)
4 AS [Количество групп], Sum(Gruppa.grChas_v_ned) AS [Количество часов]
5 FROM pedagog INNER JOIN Gruppa ON pedagog.pKod_p = Gruppa.grKod_p
6 GROUP BY pedagog.pFIO_p, pedagog.pDolzhn, pedagog.pObrazov, pedagog.pStazh, pedagog.pRazr,
7 pedagog.pKategor, pedagog.pMesto, pedagog.pKod_p
8 ORDER BY pedagog.pFIO_p
9

```

Рисунок 18 - SQL-запрос на создание таблицы «Педагоги ДО и тренеры-преподаватели»

Пример создания объекта подключения к базе данных представлен на рисунке 19.

```

1 Dim strSQL AS String      'объявляется переменная строки запроса
2 'объявляется переменная объекта ADODB.Connection (подключение к базе данных)
3 Dim Cn AS ADODB.Connection
4 'объявляется переменная объекта ADODB.Recordset (набор данных)
5 Dim Rs AS ADODB.Recordset
6 SET Cn = New ADODB.Connection 'объявляется переменная объекта ADODB.Connection
7 'создается строка подключения
8 Cn.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;DataSource=dbGranit.mdb; Persist Security Info=False"
9 Cn.OPEN      'открывается подключение к базе данных
10 'создается набор данных
11 SET Rs = New ADODB.Recordset
12 'создается строка SQL-запроса к базе данных
13 strSQL = "SELECT * FROM Uch_v_gr"
14 'выполняется SQL-запрос
15 SET Rs = Cn.Execute(strSQL)

```

Рисунок 19 - Пример создания объекта подключения к базе данных

При выборе средств для разработки приложения следует учитывать три основных фактора: ресурсы компьютера, особенности приложения (потребность в модификации функций программы, время на разработку, необходимость контроля доступа и поддержание целостности информации) и цель разработки (отчуждаемый программный продукт или система автоматизации своей повседневной деятельности).

2.3 Реализация приложения для автоматизации деятельности учебного отдела

Для автоматизации деятельности учебно-методического отдела было принято решение о создании трех модулей. Структура управления ИС представлена на рисунке 20.



Рисунок 20 - Структура управления ИС

Модуль «Деятельность педагогов ДО и тренеров-преподавателей» предназначен для ввода и редактирования информации о работе педагогов дополнительного образования и тренеров-преподавателей в базе данных [24].

Модуль «Работа с группами» предназначен для ввода и редактирования информации о работе учебных групп в базе данных. Работа центров оценивается в первую очередь по эффективности организации учебно-

познавательной групповой деятельности. Данный модуль является наиболее объемным с информационной точки зрения.

Модуль «Подготовка отчетов» предназначен для создания отчетных форм. Предполагается вывод основных форм отчетов как списочных, так и статистических. Списочные отчеты предполагают выборку данных из базы данных по определенным критериям и представление ее в виде списков. Статистические отчеты включают подсчет количественных показателей. Для заполнения и редактирования информации о работе учебных групп и их воспитанниках используется модуль «Работа с группами» [12].

«Сценарий, реализующий соединение «клиент-сервер» выглядит следующим образом: сервер занимает определенный порт, по которому он предоставляет свои услуги, и начинает ожидать поступления запросов от клиентов через этот порт. Чтобы подключиться к серверу, клиент должен знать адрес сервера и номер порта. Для соединения с сервером клиент должен создать сокет» [19].

«Для реализации TCP-сервера Qt предоставляет класс QTcpServer, который предназначен для управления входящими TCP-соединениями. Реализация сервера, принимающего и подтверждающего получение запросов клиентов представлена на рисунке 21. Сначала необходимо создать объект сервера. Чтобы запустить сервер, необходимо создать объект класса MyServer, передав в конструктор номер порта, по которому должен осуществляться нужный сервис» [19].

```
#include <QtWidgets>
#include "MyServer.h"

int main(int argc, char** argv)
{
    QApplication app(argc, argv);
    MyServer server (2323);
    server.show();
    return app.exec();
}
```

Рисунок 21 – Реализация сервера

«В классе MyServer (рисунок 22) объявлен атрибут m_ptcpServer, который является основой управления сервером в данном случае. Атрибут m_nNextBlockSize служит для хранения длинных следующего полученного от сокета блока. Многострочное текстовое поле m_ptxt предназначено для информирования о происходящих соединениях. Определение класса MyServer представлено на рисунке 22» [19].

```
#pragma once
#include <QWidget>
class QTcpServer;
class QTextEdit;
class QTcpSocket;

class MyServer : public QWidget {
    Q_OBJECT
private:
    QTcpServer* m_ptcpServer;
    QTextEdit* m_ptxt;
    quint16 m_nNextBlockSize;
private:
    void sendToClient(QTcpSocket* pSocket, const QString& str);
public:
    MyServer(int nPort, QWidget* pwgt = 0);
public slots:
    virtual void slotNewConnection();
    void slotReadClient ();
};
```

Рисунок 22 - Определение класса MyServer

«Для запуска сервера необходимо вызвать в конструкторе метод listen(), реализация которого представлена на рисунке 23. В этот метод нужно передать номер порта, который был получен в конструкторе. При возникновении ошибочных ситуаций, например невозможности захвата порта, этот метод вернет значение false, в ответ на которое будет выведено сообщение об ошибке. Если ошибки не произошло, произойдет соединение (рисунок 24) слота slotNewConnection() с сигналом newConnection(), который отправляется при каждом присоединении нового клиента» [19].

```

MyServer::MyServer(int nPort, QWidget* pWgt /*=0*/) : QWidget(pWgt)
                                                    , m_nNextBlockSize(0)
{
    m_ptcpServer = new QTcpServer(this);
    if (!m_ptcpServer->listen(QHostAddress::Any, nPort)) {
        QMessageBox: :critical (0,
                                "Server Error",
                                "Unable to start the server:"
                                + m_ptcpServer->errorString()
                                );
        m_ptcpServer->close();
        return;
    }
    connect(m_ptcpServer, SIGNAL(newConnection()),
            this, SLOT(slotNewConnection())
            );
    m_ptxt = new QTextEdit;
    m_ptxt->setReadOnly(true);

    QVBoxLayout* pvbLayout = new QVBoxLayout;
    pvbLayout->addWidget(new QLabel("<H1>Server</H1>"));
    pvbLayout->addWidget(m_ptxt);
    setLayout(pvbLayout);
}

```

Рисунок 23 – Реализация конструктора класса MyServer

Для отображения информации создан виджет многострочного текстового поля (указатель m_ptxt) и организован вызов метода setReadOnly(), в котором установлен режим только просмотра информации.

«Метод slotNewConnection(), реализация которого показана на рисунке 24, вызывается каждый раз при соединении с новым клиентом. Для подтверждения соединения с клиентом необходимо вызвать метод nextPendingConnection(), который возвратит сокет, посредством которого можно осуществлять дальнейшую связь с клиентом. Последующая работа сокета обеспечивается сигнально-слотовыми связями. В данном случае произойдет соединение сигнала disconnected(), отправляемого сокетом при отсоединении клиента, со стандартным слотом QObject::deleteLater(), предназначенным для его последующего уничтожения. При поступлении

запросов от клиентов отправляется сигнал `readyToRead()`, который имеет соединение со слотом `slotReadClient()`» [19].

```
/*virtual*/ void MyServer::slotNewConnection()
{
    QTcpSocket* pClientSocket = m_ptcpServer->nextPendingConnection();
    connect(pClientSocket, SIGNAL(disconnected()),
           pClientSocket, SLOT(deleteLater())
           );
    connect(pClientSocket, SIGNAL(readyRead()),
           this, SLOT(slotReadClient())
           );
    sendToClient(pClientSocket, "Server Response: Connected!");
}
```

Рисунок 24 - Метод `slotNewConnection()` (файл `MyServer.cpp`)

В методе `sendToClient()` формируются данные, которые будут отосланы клиенту. Реализация данного метода показана на рисунке 25 [13].

```
void MyServer: :sendToClient(QTcpSocket* pSocket, const QString& str)
{
    QByteArray arrBlock;
    QDataStream out(&arrBlock, QIODevice::WriteOnly);
    out.setVersion(QDataStream::Qt_5_3);
    out << quint16(0) << QTime: :currentTime() << str;
    out.device()->seek(0);
    out << quint16(arrBlock.size() - sizeof(quint16) );
    pSocket->write(arrBlock);
}
```

Рисунок 25 - Метод `sendToClient` (файл `MyServer.cpp`)

«Для реализации клиента нужно создать объект класса `QTcpSocket`, а затем вызвать метод `connectToHost()`, передав в него первым параметром имя компьютера (или его IP-адрес), а вторым - номер порта сервера. Объект класса `QTcpSocket` самостоятельно попытается установить связь с сервером и, в случае успеха, вышлет сигнал `connected()`. В противном случае будет выслан

сигнал `error(int)` с кодом ошибки, определенным в перечислении `QAbstractSocket::SocketError`» [19]. Это может произойти, например, в том случае, если на указанном компьютере не запущен сервер или не соответствует номер порта. После установления соединения объект класса `QTcpSocket` может высылать данные серверу или считывать их с него. Взаимодействие клиента с сервером продемонстрировано на рисунках 26 - 29. Пересылка на сервер информации, введенной в однострочном текстовом поле окна клиента, осуществляется после нажатия на кнопку `Send`.

В функции `main()`, приведенной на рисунке 26, создаем объект клиента. Если сервер и клиент запускаются на одном компьютере, то в качестве имени компьютера можно передать строку `"localhost"`.

```
#include <QApplication>
#include "MyClient.h"

int main(int argc, char** argv)
{
    QApplication app (argc, argv);
    MyClient client("localhost", 2323);
    client.show();
    return app.exec();
}
```

Рисунок 26 – Реализация взаимодействия клиента с сервером

«В классе `MyClient`, представленном на рисунке 27, объявлен атрибут `m_pTcpSocket`, который нужен для управления клиентом, и атрибут `m_nNextBlockSize`, необходимый для хранения длины следующего полученного от сокета блока. Остальные два атрибута: `m_txtinfo` и `m_txtinput` - служат для отображения и ввода информации соответственно» [19].

```

#pragma once

#include <QWidget>
#include <QTcpSocket>

class QTextEdit;
class QLineEdit;

class MyClient : public QWidget (
Q_OBJECT
private:
    QTcpSocket* m_pTcpSocket;
    QTextEdit* m_ptxtinfo;
    QLineEdit* m_ptxtinput;
    quint16 m_nNextBlockSize;
public:
    MyClient(const QString& strHost, int nPort, QWidget* pwgt = 0)
    private slots:
    void slotReadyRead();
    void slotError (QAbstractSocket::SocketError) ;
    void slotSendToServer();
    void slotConnected();
};

```

Рисунок 27 - Определение класса MyClient

«На рисунке 28 представлена реализация конструктора, в котором создается объект сокета (указатель `m_pTcpSocket`), затем вызывается метод `connectToHost()` этого объекта, устанавливающий связь с сервером. Первым параметром в этот метод передается имя компьютера, а вторым - номер порта. Коммуникация сокетов асинхронна. Сокет отправляет сигнал `connected()`, как только будет создано соединение, а также сигнал `readyRead()` – при готовности предоставить данные для чтения. Эти сигналы соединяются со слотами `slotconnected()` и `slotReadyRead()`. В случае возникновения ошибок сокет отправляет сигнал `error()`, который будет соединяться со слотом `slotError()`, предназначенным для отображения ошибок» [19].

Затем создается пользовательский интерфейс программы, состоящий из надписи, кнопки, однострочного и многострочного текстовых полей. Сигнал `clicked()` виджета кнопки нажатия соединяется со слотом `slotSendToServer()`

класса MyClient, ответственным за отправку данных на сервер. Для того, чтобы к аналогичному действию приводило и нажатие клавиши <Enter>, соединяем сигнал returnPressed() виджета текстового поля (m_ptxtInput) с тем же слотом slotSendToServer().

```
MyClient::MyClient(const QString& strHost,
                  int nPort,
                  QWidget* pwtg /*=0*/
                  ) : QWidget(pwtg)
                    , m_nNextBlockSize(0)
{
    m_pTcpSocket = new QTcpSocket(this);
    m_pTcpSocket->connectToHost(strHost, nPort);
    connect(m_pTcpSocket, SIGNAL(connected()), SLOT(slotConnected()));
    connect(m_pTcpSocket, SIGNAL(readyRead()), SLOT(slotReadyRead()));
    connect(m_pTcpSocket, SIGNAL(error(QAbstractSocket::SocketError)),
            this, SLOT(slotError(QAbstractSocket::SocketError))
            );
    m_ptxtInfo = new QTextEdit;
    m_ptxtInput = new QLineEdit;
    m_ptxtInfo->setReadOnly(true);

    QPushButton* pcmd = new QPushButton("&Send");
    connect(pcmd, SIGNAL(clicked()), SLOT(slotSendToServer()));
    connect(m_ptxtInput, SIGNAL(returnPressed()),
            this, SLOT(slotSendToServer())
            );
    QVBoxLayout* pvbLayout = new QVBoxLayout;
    pvbLayout->addWidget(new QLabel("<H1>Client</H1>"));
    pvbLayout->addWidget(m_ptxtInfo);
    pvbLayout->addWidget(m_ptxtInput);
    pvbLayout->addWidget(pcmd);
    setLayout(pvbLayout);
}
```

Рисунок 28 - Реализация конструктора класса MyClient

В программном коде, представленном на рисунке 29 приведен слот slotReadyToRead(), который вызывается при поступлении данных от сервера.

Цикл for нужен потому, что с сервера не все данные могут прийти одновременно. Поэтому клиент, как и сервер, должен быть в состоянии получать как весь блок целиком, так и только часть блока или даже все блоки

сразу. Каждый принятый блок начинается полем размера блока. Чтение данных из сокета осуществляется при помощи объекта потока данных. Полученная информация добавляется в виджет многострочного текстового поля (указатель `m_ptxtInfo`) с помощью метода `append()`.

```
void MyClient: :slotReadyRead()
{
    QDataStream in(m_pTcpSocket);
    in.setVersion(QDataStream::Qt_5_3);
    for (;;) {
        if ( !m_nNextBlockSize) {
            if (m_pTcpSocket->bytesAvailable() < sizeof(quint16)) {
                break;
            }
            in >> m_nNextBlockSize;
            if (m_pTcpSocket->bytesAvailable() < m_nNextBlockSize)
                break;
        }
        QTime time;
        QString str;
        in >> time >> str;
        m_ptxtInfo->append(time.toString() + " " + str);
        m_nNextBlockSize = 0;
    }
}
```

Рисунок 29 - Реализация метода `slotReadyRead()`

В завершение анализа блока данных атрибуту `m_nNextBlockSize` присваиваем значение 0, которое говорит о том, что размер очередного блока данных неизвестен.

Пример работы с информационной системой представлен на диаграмме последовательности (рисунок 30).

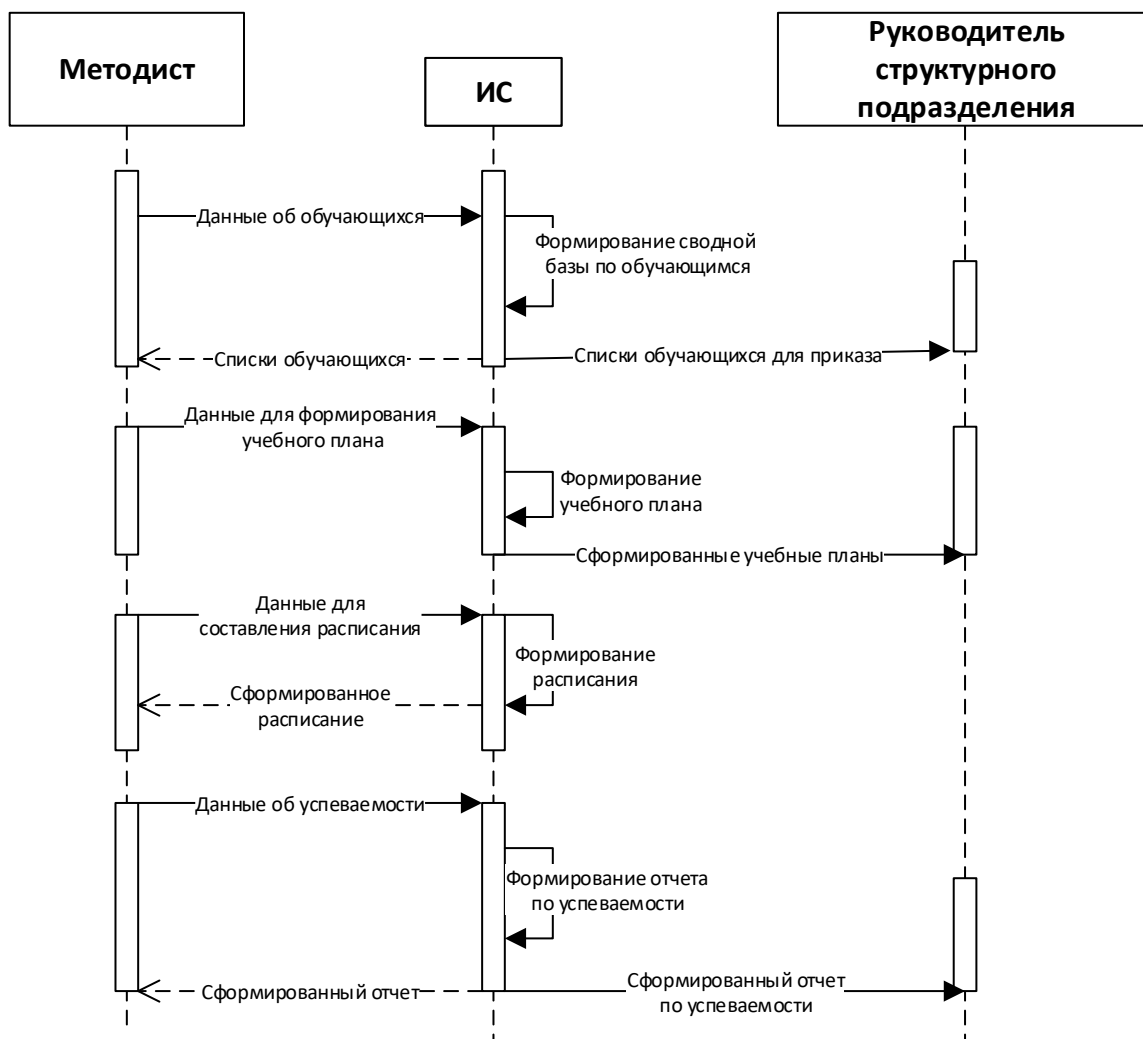


Рисунок 30 – Диаграмма последовательности работы учебно-методического отдела с информационной системой

Выводы по второй главе

Проведено концептуальное проектирование предметной области. Определены роли для работы с информационной системой. Определены основные функции информационной системы для автоматизации основных видов деятельности учебно-методического отдела. Спроектированы логическая и физическая модели данных. Осуществлен выбор средств реализации приложения.

Глава 3 Тестирование реализованного приложения для учебно-методического отдела

3.1 Тестирование реализованного приложения

Понятие «ошибка» в терминах программирования означает ошибку или дефект в компьютерной программе, который препятствует правильной работе программного обеспечения, что часто приводит к неправильному выводу или даже к сбою. Технические проблемы возникают постоянно в процессе разработки. Чтобы решить эти проблемы, нужно выяснить все эти проблемы и решить их, прежде чем выпустить приложение для пользователей. Метод, используемый для поиска технических проблем, называется отладкой [22].

В процессе отладки программы, независимо от языка программирования или платформы, необходимо знать, какая часть кода вызывает проблему [23]. Есть несколько способов определить проблемный код [25]:

- спросить пользователя, в какой момент произошла ошибка, например, какая кнопка была нажата, какие шаги привели к сбою и так далее;
- закомментировать часть кода, затем собрать и снова запустить программу, чтобы проверить, возникает ли проблема по-прежнему. Продолжать комментировать код, пока не будет найдена проблемная строка кода;
- использовать встроенный отладчик, чтобы проверить изменения переменных в целевой функции, установив точку останова данных. Таким способом можно легко определить, изменилась ли одна из переменных на неожиданное значение или указатель объекта стал неопределенным указателем [26];
- убедиться, что все библиотеки, включенные в установщик для пользователей, имеют номера версий, совпадающие с теми, которые используются в текущем проекте.

Qt поставляется со встроенным модулем модульного тестирования, который мы можем использовать, добавив ключевое слово `testlib` в файл проекта (.pro): `QT += core gui testlib`

После этого добавим в исходный код следующий заголовок:
`#include<QtTest/QtTest>`

Затем можем начать тестировать код. Нужно объявить тестовые функции как частные слоты. Помимо этого, класс также должен наследоваться от класса [27]. Например, создадим две текстовые функции с именами `testString()` и `testGui()`. Определения функций представлены на рисунке 31.

```
void MainWindow :: testString()
{
    QString text = "Testing";
    QVERIFY (text.toUpper() == "TESTING");
}
void MainWindow :: testGui()
{
    QTest :: keyClicks (ui -> lineEdit, "testing gui");
    QCOMPARE (ui -> lineEdit -> text(), QString("testing gui"));
}
```

Рисунок 31 - Определения функций `testString()` и `testGui()`

Для оценки выражения, переданного в качестве его аргумента, используются макросы, предоставляемые классом `QTest`, такие как `QVERIFY`, `QCOMPARE`. Если выражение оценивается как истинное, выполнение тестовой функции продолжается. В противном случае в журнал тестирования добавляется сообщение с описанием сбоя, и выполнение функции тестирования прекращается. Также использовали `QTest::keyClicks()` для имитации щелчка мышью в приложении.

Чтобы начать тестирование класса `MainWindow`, используем функцию `QTEST_MAIN()`. Результат unit-теста представлен на рисунке 32.

```
***** Start testing of MainWindow *****
Config: Using QTest library 5.9.1 (i386-little-endian-ilp32
shared (dynamic) debug build; by GCC 5.3.0)
PASS      : MainWindow::initTestCase()
PASS      : MainWindow::_q_showIfNotHidden()
PASS      : MainWindow::testString()
PASS      : MainWindow::testGui()
PASS      : MainWindow::cleanupTestCase()
Totals: 5 passed, 0 failed, 0 skipped, 0 blacklisted, 880ms
***** Finished testing of MainWindow *****
```

Рисунок 32 – Результат unit-теста

Результаты модульного тестирования показали работоспособность реализованного кода.

3.2 Пример работы реализованного приложения

Приложение для автоматизации работы учебно-методического отдела состоит из трех модулей: «Деятельность педагогов ДО и тренеров-преподавателей», «Работа с группами», «Подготовка отчетов». При работе с модулем «Деятельность педагогов ДО и тренеров-преподавателей» при запуске открывается форма, представленная на рисунке 33. На ней можно просмотреть список педагогов и тренеров, работающих в центре. Кнопка «Обновить» позволяет вывести список педагогов из базы повторно. После щелчка по кнопке «Удалить педагога» выбранный педагог удаляется из списка при подтверждении пользователем удаления. При щелчке по кнопке «Редактировать» вызывается пользовательская форма, в которой осуществляется редактирование данных педагога. После выбора педагога в списке становятся доступными кнопки «Проекты», «Родители», «Мероприятия». Они позволяют перейти на формы с соответствующими названиями и просмотреть данные или внести изменения для выбранного педагога.

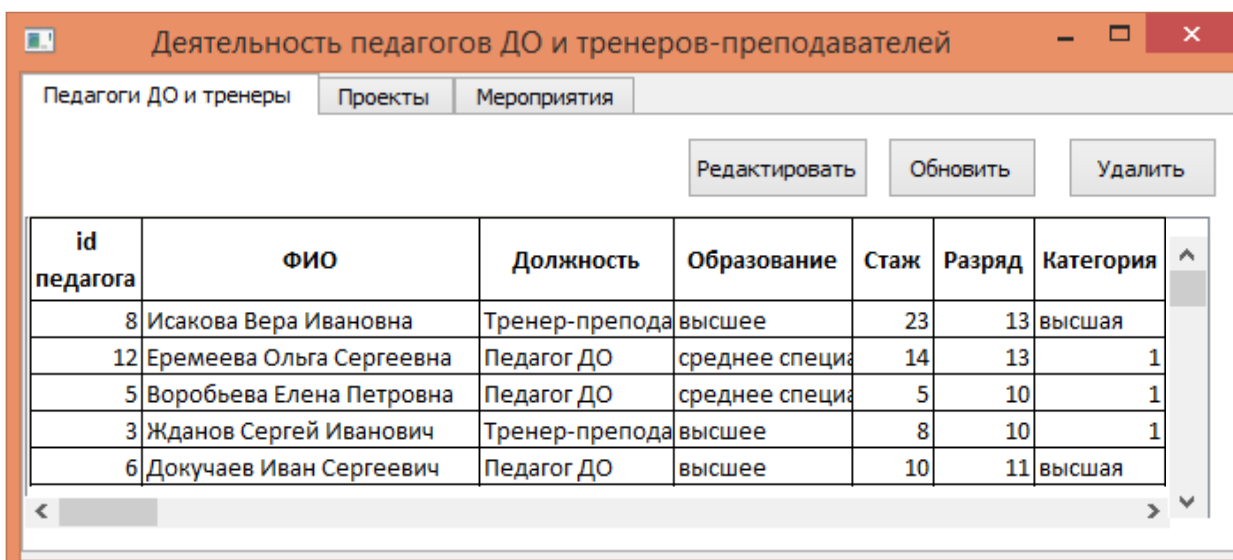


Рисунок 33 - Пользовательская форма «Педагоги ДО и тренеры» модуля «Деятельность педагогов ДО и тренеров-преподавателей»

На рисунке 33 представлена форма «Проекты», которая содержит информацию о проектах конкретного педагога. Кнопка «Добавить» позволяет пользователю вызвать форму «Проекты. Добавление нового проекта» (рисунок 34).

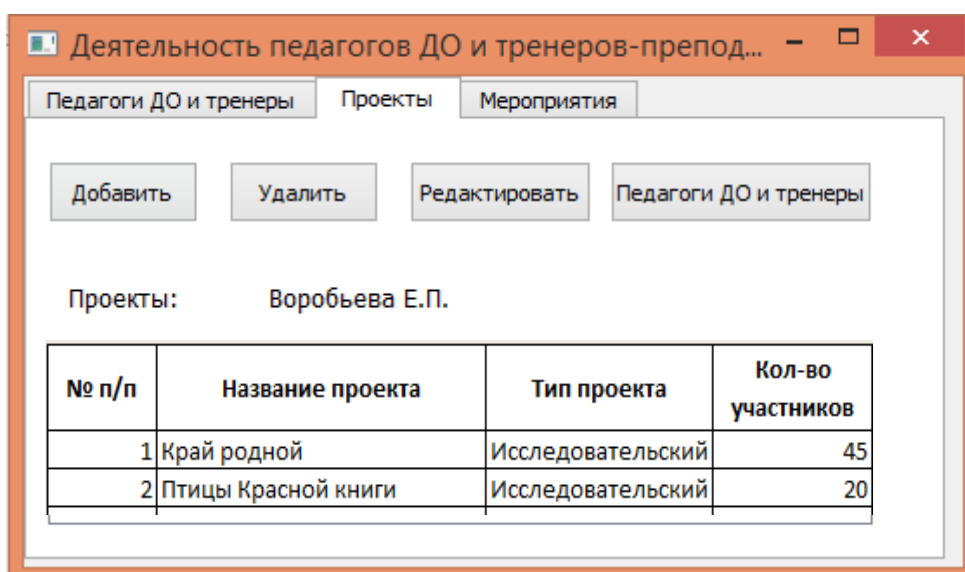


Рисунок 34 - Пользовательская форма «Проекты» модуля «Деятельность педагогов ДО и тренеров-преподавателей»

При щелчке по кнопке «Редактировать» вызывается пользовательская форма для редактирования, интерфейс которой аналогичен форме добавления. При щелчке по кнопке «Удалить» осуществляется удаление проекта из БД, после подтверждения пользователем удаления. При щелчке по кнопке «Педагоги ДО и тренеры» осуществляется переход к соответствующей форме.

После ввода данных в пользовательскую форму «Проект. Добавление нового проекта» при щелчке по кнопке «Добавить» осуществляется добавление проекта в базу для выбранного педагога (рисунок 35). При щелчке по кнопке «Отменить» форма закрывается.

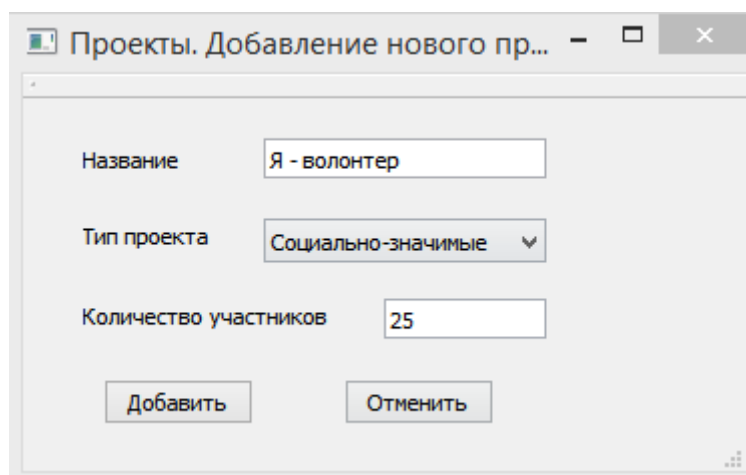


Рисунок 35 - Пользовательская форма «Проекты. Добавление нового проекта»

На рисунке 36 представлена форма «Мероприятия», которая содержит информацию о мероприятиях конкретного педагога. Кнопка «Добавить» позволяет пользователю вызвать форму «Мероприятия. Добавление нового мероприятия» (рисунок 37). При щелчке по кнопке «Редактировать» вызывается пользовательская форма для редактирования, интерфейс которой аналогичен форме добавления. При щелчке по кнопке «Удалить» осуществляется удаление мероприятия из базы данных. При щелчке по кнопке «Педагоги ДО и тренеры» осуществляется переход к соответствующей форме.

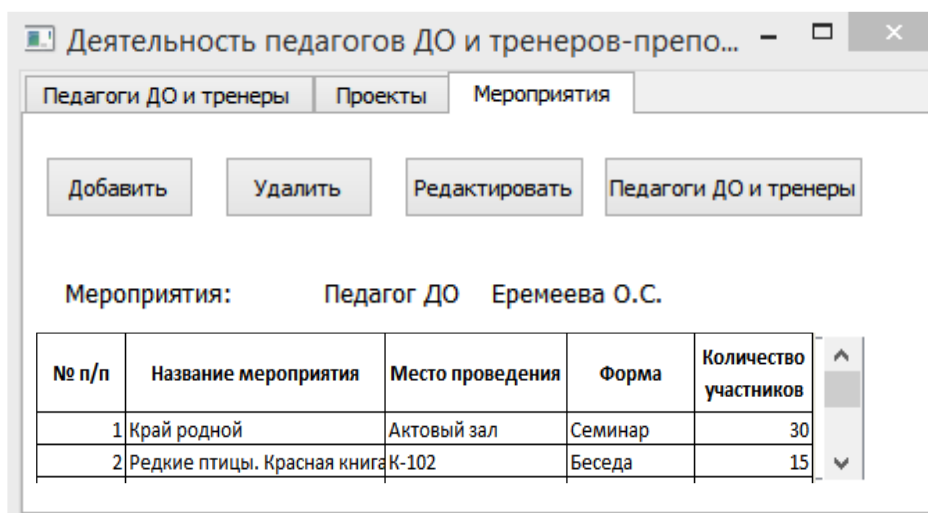


Рисунок 36 - Пользовательская форма «Мероприятия» модуля «Деятельность педагогов ДО и тренеров-преподавателей»

После ввода данных в пользовательскую форму «Мероприятия. Добавление нового мероприятия» при щелчке по кнопке «Добавить» осуществляется добавление нового мероприятия в базу для выбранного педагога. При щелчке по кнопке «Отменить» форма закрывается.

Рисунок 37 - Пользовательская форма «Мероприятия. Добавление нового мероприятия»

При работе с модулем «Работа с группами» при запуске открывается форма, представленная на рисунке 38.

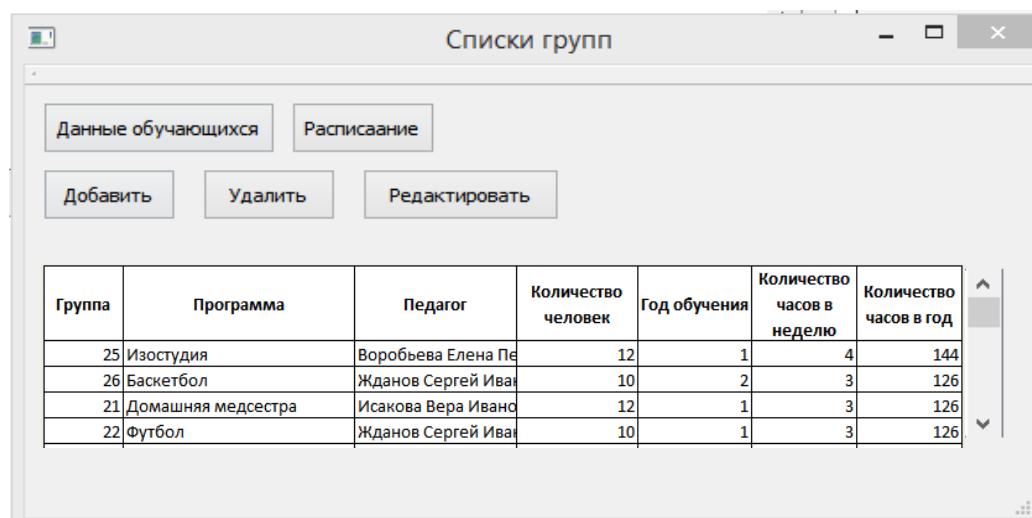


Рисунок 38 - Пользовательская форма «Список групп» модуля «Работа с группами»

Она позволяет работать с данными по учебным группам (редактировать, вносить новые данные, удалять записи). После выбора группы становятся доступными кнопки «Учащиеся групп» и «Расписание», после щелчка по ним осуществляется переход к соответствующим формам. Щелчок по кнопке «Добавить/Справочники» позволяет вызвать форму «Группы. Добавление новой группы». Кнопки «Редактировать» и «Удалить группу» используются для выполнения соответствующих действий. При щелчке по кнопке «Данные обучающихся» осуществляется переход к форме «Список обучающихся».

Пользовательская форма «Группы. Добавление новой группы» (рисунок 39) позволяет не только вносить новые данные по группам, но и позволяет осуществить доступ к справочным таблицам «Педагоги» и «Тип группы», путем щелчка по соответствующим кнопкам на форме.

Перейти к форме списка воспитанников учебной группы можно только после выбора группы в форме «Список групп». С помощью соответствующих кнопок можно удалить или обновить данные. Можно перейти к формам

«Список групп» и «Данные обучающихся». При щелчке по кнопкам «Редактировать» и «Добавить/Справочники» вызываются соответствующие формы.

Группы. Добавление новой группы

Группа 27 Количество учащихся на начало года 18

Педагог: Исакова Вера Иванов

Тип группы: Код 22

Добавить педагога Выбрать тип

Оплата

Добавить группу Отменить

Рисунок 39 - Пользовательская форма «Группы. Добавление новой группы»

На рисунке 40 представлено расписание учебной группы. Перейти к этой форме можно только после выбора группы в форме «Список групп».

Расписание

Список групп Обновить

Добавить Удалить Редактировать

Группа 25

Программа Изостудия

№ п/п	День	Время	Место
1	вторник	18.00 - 19.00	К-104
2	четверг	18.00 - 19.00	К-104
3	суббота	12.00 - 13.00	К-104

Рисунок 40 - Пользовательская форма «Расписание» модуля «Работа с группами»

С помощью соответствующих кнопок можно удалить или обновить данные. Можно перейти к форме «Список групп». При щелчке по кнопкам «Редактировать» и «Добавить» вызываются соответствующие формы.

Пользовательские формы редактирования данных аналогичны формам для добавления данных.

Приложение «Подготовка отчетов» трудоемкое и функционально насыщенное, включает множество SQL-запросов к базе данных. Вкладка «Настройка параметров отчета по количеству учащихся и групп» представлена на рисунке 41.

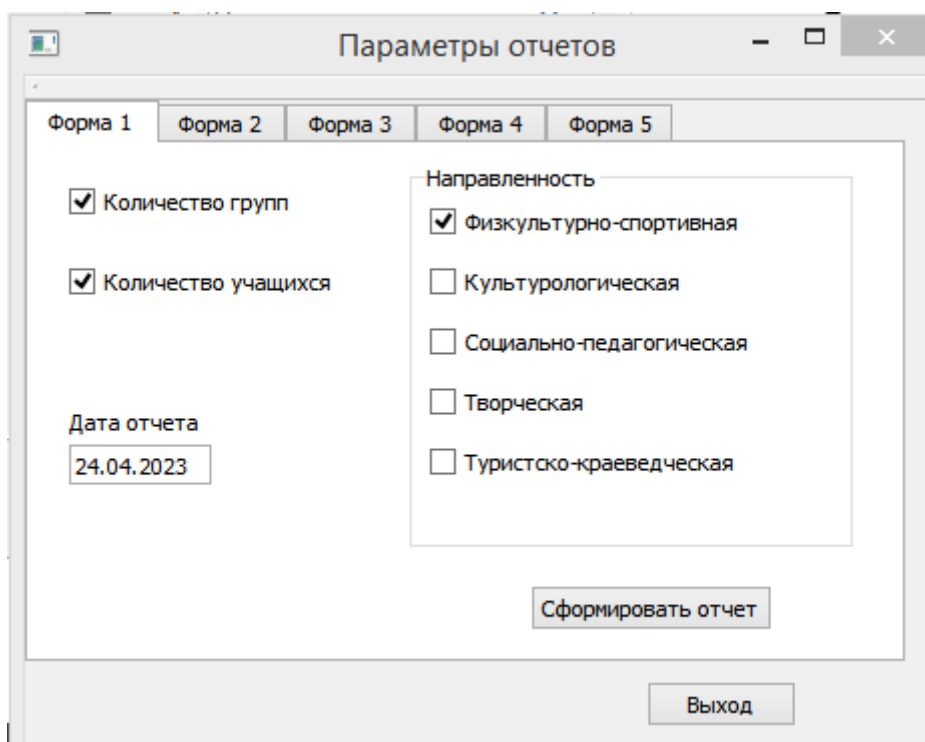


Рисунок 41 - Вкладка «Настройка параметров отчета по количеству учащихся и групп»

В зависимости от выбранной пользователем формы по щелчке по кнопке «Параметры» осуществляется переход к вкладке для настройки параметров выбранного отчета. Процесс формирования необходимого отчета осуществляется после выбора полей отчета для списочных отчетов и ввода

даты. Достаточно нажать кнопку «Сформировать отчет», чтобы загрузить готовый отчет на текущую дату. Отчет по количеству учащихся и групп представлен на рисунке 42.

Отчет по количеству учащихся и групп 24.04.2023			
№ п/п	Программа	Количество групп	Количество обучающихся
1	Физкультурно-спортивная		
	Футбол	2	41
	Баскетбол	1	18

Рисунок 42 - Отчет по количеству учащихся и групп

Вкладка «Настройка параметров отчета по проектной деятельности» представлена на рисунке 43.

The screenshot shows a window titled "Параметры отчетов" (Parameters of reports). It has five tabs: "Форма 1", "Форма 2", "Форма 3", "Форма 4", and "Форма 5". The "Форма 1" tab is active. Inside the window, there is a section titled "Вид проекта" (Project type) with three checkboxes: "Социально-значимые" (Socially significant), "Исследовательские" (Research), and "Все виды" (All types). At the bottom right of the window, there is a button labeled "Сформировать отчет" (Generate report), and at the bottom center, there is a button labeled "Выход" (Exit).

Рисунок 43 - Вкладка «Настройка параметров отчета по проектной деятельности»

Отчет по проектной деятельности представлен на рисунке 44.

**Отчет по проектной деятельности
за октябрь 2022**

№	Проект	Руководитель	Количество обучающихся
Социально-значимые проекты			
1	Я - волонтер	Исакова Вера Ивановна	20
2	Домашняя медсестра	Исакова Вера Ивановна	25
Исследовательские			
3	Край родной	Еремеева Ольга Сергеевна	25
4	Редкие птицы. Красная книга	Еремеева Ольга Сергеевна	24

Рисунок 44 - Отчет по проектной деятельности

Отчет по мероприятиям за период представлен на рисунке 45.

**Отчет по мероприятиям
в период каникул
с 21.03.2023 по 29.03.2023**

Форма мероприятия	Количество мероприятий	Количество участников по плану	Фактическое количество участников
Групповые и индивидуальные учебные занятия			
Учебно-тренировочные сборы			
Оздоровительно-образовательные загородные лагеря			
Оздоровительно-образовательные лагеря дневного пребывания			
Участие в соревнованиях и конкурсах	8	122	116
Спортивные и культурно-массовые мероприятия	4	82	64
Социально-значимая и общественно-полезная деятельность			
Беседа	2	25	26
Экскурсия	1	30	30
Всего:	15	259	236

Рисунок 45 - Отчет по мероприятиям за период

На вкладке «Настройка параметров отчета по дополнительным образовательным программам» имеется возможность выбора направленности (физкультурно-спортивная, культурологическая, социально-педагогическая, туристско-краеведческая), а также возраст обучающихся, срок реализации, степень реализации, уровень углубления проектной деятельности.

Электронные отчеты полностью соответствуют установленным формам отчетности, применяемым сотрудниками учебно-методического отдела.

Выводы по третьей главе

В данной главе представлены возможности выявления технических проблемы в проекте Qt, используя несколько методов отладки. Продемонстрированы возможности автоматизации некоторых этапов отладки с помощью модульного тестирования. Представлен пример работы с формами приложения.

Таким образом, можно сделать вывод, что разработанное приложение представляет весь необходимый функционал для выполнения основных функций учебно-методического отдела по организации внеучебной деятельности детей.

Заключение

В результате выполнения выпускной квалификационной работы разработана информационная система, автоматизирующая деятельность учебно-методического отдела муниципального образовательного учреждения дополнительного образования детей.

Основные результаты работы сводятся к следующему:

- проведен анализ деятельности типового учебно-методического отдела и сформулированы требования к разрабатываемой информационной системе,
- проанализирован документооборот по учету результативности деятельности педагогов дополнительного образования, составлена схема информационных потоков в учебно-методическом отделе,
- спроектированы модели предметной области базового и проектного вариантов, построена логическая модель базы данных для автоматизации деятельности учебно-методического отдела,
- проведен выбор оптимального решения задач автоматизации,
- разработано приложение средствами фреймворка Qt. Приложение состоит из трех модулей: «Деятельность педагогов ДО и тренеров-преподавателей», «Работа с группами», «Подготовка отчетов»,
- приложение разработано с учетом того, что все электронные документы могут быть выведены на печать,
- проведено тестирование реализованного приложения.

Таким образом, разработанная информационная система обеспечивает сокращение времени на подготовку выходных документов, возможности появления ошибок в отчетах, наконец, позволяет заместителю по учебно-воспитательной работе эффективнее организовать работу сотрудников отдела, а также увеличить объем проектной деятельности методистов. Это, в конечном итоге, положительно повлияет на реализацию современных подходов в учебных программах дополнительного образования.

Список используемой литературы

1. Базаржапова Т. Ж. Базы данных : учебное пособие для обучающихся по направлению подготовки 09.03.03 «Прикладная информатика» / составители Т. Ж. Базаржапова, О. А. Гармаева, А. Ю. Хаптахаяев. — Улан-Удэ : Бурятская государственная сельскохозяйственная академия им. В.Р. Филиппова, 2022. — 84 с.
2. Бугаев Ю. В. Исследование и моделирование информационных процессов и систем : учебное пособие / Ю. В. Бугаев, Л. А. Коробова, С. Н. Черняева. — Воронеж : Воронежский государственный университет инженерных технологий, 2022. — 108 с.
3. Бурков, А. В. Проектирование информационных систем в Microsoft SQL Server 2008 и Visual Studio 2008 : учебное пособие / А. В. Бурков. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 310 с.
4. Вакорин М. П. Архитектура предприятий и информационных систем : учебное пособие / М. П. Вакорин, Д. Н. Достовалов. — Новосибирск : Новосибирский государственный технический университет, 2022. — 64 с.
5. Грекул, В. И. Проектирование информационных систем : учебное пособие / В. И. Грекул, Г. Н. Денищенко, Н. Л. Коровкина. — 3-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 299 с.
6. Дерябкин, В. П. Проектирование информационных систем по методологии UML с использованием Qt-технологии программирования : учебное пособие / В. П. Дерябкин, В. В. Козлов. — Самара : Самарский государственный технический университет, ЭБС АСВ, 2017. — 156 с.
7. Евдошенко О. И. Проектирование информационных систем : учебно-методическое пособие / составители О. И. Евдошенко, Ю. С. Андрианова, А. А. Морозова. — Астрахань : Астраханский государственный архитектурно-строительный университет, ЭБС АСВ, 2022. — 70 с.

8. Ильин И. В. Архитектура предприятия и цифровая трансформация : учебное пособие / И. В. Ильин, А. А. Лепехин, А. Д. Борреманс [и др.]. — Санкт-Петербург : Санкт-Петербургский политехнический университет Петра Великого, 2022. — 74 с.

9. Ильин, И. В. Базы данных : учебное пособие / И. В. Ильин, О. Ю. Ильяшенко. — Санкт-Петербург : Санкт-Петербургский политехнический университет Петра Великого, 2020. — 96 с.

10. Киренберг А. Г. Информационная безопасность современных операционных систем : учебное пособие / А. Г. Киренберг. — Кемерово : Кузбасский государственный технический университет имени Т.Ф. Горбачева, 2022. — 138 с.

11. Кугаевских, А. В. Проектирование информационных систем. Системная и бизнес-аналитика : учебное пособие / А. В. Кугаевских. — Новосибирск : Новосибирский государственный технический университет, 2018. — 256 с.

12. Минин Ю. В. Разработка графического интерфейса пользователя информационной системы с использованием библиотеки Qt : учебное пособие / Ю. В. Минин, А. И. Елисеев, В. В. Алексеев, Ю. А. Губсков. — Тамбов : Тамбовский государственный технический университет, ЭБС АСВ, 2021. — 80 с.

13. Моренкова, О. И. Программирование на C++ в среде разработки Qt Creator : учебно-методическое пособие / О. И. Моренкова, А. Ю. Голошубов. — Новосибирск : Сибирский государственный университет телекоммуникаций и информатики, 2021. — 90 с.

14. Морозова, В. И. Функциональное и объектное проектирование информационных систем : учебно-методическое пособие к выполнению практических работ / В. И. Морозова, К. Э. Врублевский. — Москва : Российский университет транспорта (МИИТ), 2021. — 58 с.

15. Пиляй, А. И. Базы данных и операционные системы : учебно-методическое пособие / А. И. Пиляй, А. М. Якубович. — Москва : МИСИ-МГСУ, ЭБС АСВ, 2021. — 46 с.

16. Сафонов В. О. Основы современных операционных систем : учебное пособие / В. О. Сафонов. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 826 с.

17. Тимофеев А. В. Проектирование и разработка информационных систем : учебное пособие для СПО / А. В. Тимофеев, З. Ф. Камальдинова, Н. С. Агафонова. — Саратов : Профобразование, 2022. — 91 с.

18. Токмаков, Г. П. Базы данных: модели и структуры данных, язык SQL, программирование баз данных : учебное пособие / Г. П. Токмаков. — Ульяновск : Ульяновский государственный технический университет, 2021. — 362 с.

19. Шлее М. Qt 5.10. Профессиональное программирование на C++. - СПб.: БХВ-Петербург, 2018. - 1072 с.

20. АИС «КТЕ.ОБРАЗОВАНИЕ.ДОПОЛНИТЕЛЬНОЕ ОБРАЗОВАНИЕ» [Электронный ресурс]. – Режим доступа: <https://ktelabs.ru/product-o.html> (дата обращения 11.03.2023).

21. АИС «БАРС. Электронное дополнительное образование». - Электронный ресурс. – Режим доступа: <https://zatocdod.ru> (дата обращения 11.03.2023).

22. Djirdeh H., Murray N., Lerner A.: Fullstack Vue: The Complete Guide to Vue.js., 2018. – 442с.

23. G. Heineman, G. Pollice and S. Selkow Algorithms in a Nutshell 2E [Текст]: O'Reilly Media 2015. – 435 с.: - Sorting Algorithms: с. 81 – 87. – Searching: с. 120 – 121. - ISBN: 063-6-920-03288-5

24. Introduction to C++/CLI Programming [Электронный ресурс]: Ajay Yadav. — Электрон. текстовые данные. — Codeguru 2015. — Режим доступа: <https://www.codeguru.com/cpp/cpp/introduction-to-ccli-programming.html>.

25. Lee Zhi Eng: Hands-On GUI Programming with C++ and Qt5 – First published: April 2018. – 393 p

26. Joseph Howse. Learning OpenCV 4 Computer Vision with Python 3: Get to grips, techniques, and algorithms for computer vision and machine learning 3rd Edition / Joseph Howse, Joe Minichino. – USA : Packt Publishing, 2020. – 372 p.

27. Raman R., Dewailly L.: Building RESTful Web Services with Spring 5 - Second Edition: Leverage the power of Spring 5.0, Java SE 9, and Spring Boot 2.0, 2018. — 228 p