



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ЮУрГГПУ»)

ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ

КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ

Автоматизация методик педагогической диагностики в оценке качества обучения  
школьников

Выпускная квалификационная работа  
по направлению 09.03.02 Информационные системы и технологии  
Направленность программы бакалавриата  
«Информационные системы и технологии»


Проверка на объем заимствований:

62,35 % авторского текста

Работа рекомендована к защите  
рекомендована/не рекомендована

« \_\_\_ » \_\_\_\_\_ 20\_\_ г.

и.о. зав. кафедрой И, ИТ и МОИ

 Рузаков А.А.

Выполнил :

Студент группы ЗФ404/095-4-1

Кузьмин Николай Михайлович

Научный руководитель:

доцент, кандидат педагогических наук,

 Леонова Елена Анатольевна

Челябинск

2017



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**Федеральное государственное бюджетное образовательное учреждение  
высшего образования**

**«ЮЖНО-УРАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНО-  
ПЕДАГОГИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ФГБОУ ВО «ЮУрГГПУ»)**

**ФИЗИКО-МАТЕМАТИЧЕСКИЙ ФАКУЛЬТЕТ**

**КАФЕДРА ИНФОРМАТИКИ, ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И  
МЕТОДИКИ ОБУЧЕНИЯ ИНФОРМАТИКЕ**

**Автоматизация методик педагогической диагностики в оценке качества обучения  
школьников**

**Выпускная квалификационная работа  
по направлению 09.03.02 Информационные системы и технологии  
Направленность программы бакалавриата**

**«Информационные системы и технологии»**

Проверка на объем заимствований:  
\_\_\_\_\_ % авторского текста

Работа \_\_\_\_\_ к защите  
рекомендована/не рекомендована

« \_\_\_ » \_\_\_\_\_ 20\_\_ г.  
и.о. зав. кафедрой И, ИТ и МОИ

\_\_\_\_\_ Рузаков А.А.

Выполнил :  
Студент группы ЗФ404/095-4-1  
Кузьмин Николай Михайлович

Научный руководитель:  
доцент, кандидат педагогических наук,  
\_\_\_\_\_ Леонова Елена Анатольевна

**Челябинск  
2017**

## Оглавление

Введение .....	3
<b>1 Описание предметной области и назначение системы оценки качества обучения .....</b>	<b>5</b>
<b>1.1 Характеристика Уйской средней общеобразовательной школы имени Александра Ивановича Тихонова .....</b>	<b>5</b>
1.1.1 Описание школы.....	5
1.1.2 Описание структуры МКОУ «Уйская СОШ имени А.И.Тихонова» и основных бизнес-процессов .....	7
1.1.3 Описание средств автоматизации работы МКОУ «Уйская СОШ имени А.И.Тихонова» .....	12
<b>1.2 Описание проектируемой системы оценки качества обучения.....</b>	<b>16</b>
1.2.1 Назначение системы оценки качества обучения.....	16
1.2.2 Цели создания системы оценки качества обучения\.....	19
1.2.3 Требования к структуре и функционированию системы оценки качества обучения .....	19
1.2.4 Требования к функциям, выполняемым системой оценки качества обучения .....	21
<b>1.3 Сравнительный анализ аналогичных программных продуктов .....</b>	<b>23</b>
Выводы по главе 1 .....	29
<b>2 Проектирование системы оценки качества обучения .....</b>	<b>30</b>
2.1 Выбор архитектуры информационной среды и инструментальных средств реализации.....	30
2.2 Описание модели данных .....	42
2.3 Описание реализации основных функций системы(подсистемы) .....	51
<b>3 Глава 3. Документирование готового программного продукта.....</b>	<b>63</b>
3.1 Испытания системы оценки качества обучения .....	63
3.2 Руководство пользователя системы оценки качества обучения .....	67
3.3 Руководство системного программиста .....	76
3.4 Технико-экономическое обоснование разработки системы оценки качества обучения .....	85
Выводы по Главе 3.....	90
Заключение.....	91
Библиографический список .....	92

## Введение

Неотъемлемый компонент образовательного процесса – это педагогическая диагностика, при помощи которой определяют достижение поставленной цели обучения. Без проведения диагностики невозможно эффективное управление дидактическими процессами. Диагностика – это точное определение результата дидактических процессов

В понятие диагностики вкладывают более широкий и глубокий смысл, чем в понятие проверка знания, умения и приобретенных навыков обучаемого. Последнее понятие используется лишь для констатации результатов, без объяснения их происхождения. Тогда как при диагностировании рассматривают результат с учетом способов его достижения, выявляют тенденции, динамику дидактических процессов.

В процессах педагогической диагностики можно выделить процессы контроля, проверки, оценивания (оценки), накопления статистических данных, проведение анализа, выявления динамики, тенденции, прогнозирования дидактических процессов.

Важный компонент осуществления диагностирования - это контроль. Контролем называют наблюдение за процессами усвоения знаний, умений и получением навыков. Составная часть контроля - проверка. Проверка – это упорядоченная система действий и операций для проведения контроля за усвоением знаний, умений и получением навыков.

Актуальность темы выпускной работы обусловлена тем, что по своей сути проведение контроля установление обратных связей, т.е. предоставляет сведения о результатах учебной деятельности учащихся. преподаватель выясняет, какие, в каком объеме знания усвоены обучаемыми, готовы ли они к восприятию новой информации. Он также получает сведения о характере самостоятельной учебной работе обучаемых. Контроль также необходим,

чтобы показать обучающему, насколько его собственная работа является плодотворной, удачно ли он применяет возможности педагогического процесса в учебных целях. Данный процесс требует большого количества ручного труда по обработке полученных данных. Поэтому осуществление автоматизации процесса контроля является весьма необходимым для эффективной работы.

Объект исследования: методика педагогической диагностики в оценке качества обучения школьников.

Предмет исследования: проведение автоматизации педагогической диагностики.

Цель работы состоит в разработке программного обеспечения для диагностики качества обучения.

Для достижения цели работы требуется выполнить следующие задачи.

1. Выявить методологические и психолого-педагогические основы и изучить состояние проблемы диагностики качества образования;
2. Разработать содержание, формы, методы-и средства диагностики качества обучения школьников.
3. Разработать программное обеспечение.
4. Проверить работоспособность созданной программы;
5. Определить эффективность предложенной модели диагностики качества обучения школьников по общим дисциплинам.

## **1 Описание предметной области и назначение системы оценки качества обучения**

### **1.1 Характеристика Уйской средней общеобразовательной школы имени Александра Ивановича Тихонова**

#### **1.1.1 Описание школы**

Муниципальное казенное общеобразовательное учреждение "Уйская средняя общеобразовательная школа имени Александра Ивановича Тихонова" (далее МКОУ «Уйская СОШ имени А.И.Тихонова») зарегистрировано постановлением Главы Уйского муниципального района Челябинской области (лицензия № 8984 от 01.02.2012г.; свидетельство о государственной аккредитации № 1850 от 23.04.2014г.) Учредитель ОУ Администрация Уйского муниципального района.

Школа расположена в районном центре Уйского района. Из 6 населенных пунктов (Яринка, Фомино, Воронино, Глазуновка, Горки, Брюхово) осуществляется подвоз детей на 4 автобусах и ГАЗ-322121.

В школе пятидневная учебная неделя. Занятия начинаются в 8:30 и проходят в одну смену. Продолжительность уроков 40 минут с тремя переменами по 10 минут, одна перемена - 15 минут и одна перемена - 20 минут.

Во вторую смену в школе проводятся занятия в кружках, секциях, внеклассная работа. В школе реализуется Концепция развития естественно-математического и технологического образования "ТЕМП". Организованы пункты консультаций по подготовке домашнего задания: русский язык, математика, компьютеры свободного доступа.

В школе обучается 565 учеников, 26 класс-комплектов.

Основная школа – 483 человека, 20 класс-комплектов.

Старшая школа – 82 человека, 4 класс-комплекта.

Цели деятельности Школы:

- формирование общей культуры личности обучающихся на основе усвоения обязательного минимума содержания общеобразовательных

программ, их адаптация к жизни в обществе, создание основы для осознанного выбора и последующего освоения профессиональных образовательных программ, воспитание гражданственности, трудолюбия, уважения к правам и свободам человека, любви к окружающей природе, Родине, семье, формирование здорового образа жизни;

- создание максимально благоприятных условий для умственного, нравственного, физического, эстетического развития личности ребенка;
- формирование у обучающихся адекватной современному уровню знаний и уровню ступени обучения целостной картины мира, адаптация личности к жизни в обществе;
- развитие и совершенствование образовательного процесса, материально-технической базы, осуществление дополнительных мер социальной поддержки обучающихся, воспитанников и работников Школы.

Предметом деятельности Школы является:

- реализация основных общеобразовательных программ начального общего, основного общего, среднего (полного) общего образования;
- реализация основных общеобразовательных программ дошкольного образования;
- реализация программ профессиональной подготовки;
- реализация программ дополнительного образования (научно-технической, спортивно-технической, культурологической, физкультурно-спортивной, туристско-краеведческой, эколого-биологической, военно-патриотической, социально-педагогической, социально-экономической, естественнонаучной, художественно-эстетической направленности и др.);
- организация работы по повышению квалификации педагогических работников Школы;
- разработка учебных планов, программ, учебных пособий, научной, методической, справочной литературы;

- проведение психологической диагностики, тестирования, консультаций учителя-логопеда, педагога-психолога;
- организация семинаров, конференций, конкурсов, олимпиад, в том числе международных;
- использование и совершенствование методик образовательного процесса и образовательных технологий, в том числе дистанционных образовательных технологий;
- организация концертов, выставок, выставок-продаж;

### 1.1.2 Описание структуры МКОУ «Уйская СОШ имени А.И.Тихонова» и основных бизнес-процессов

Организационно-штатная структура МКОУ «Уйская СОШ имени А.И.Тихонова» приведена на рисунке 1.

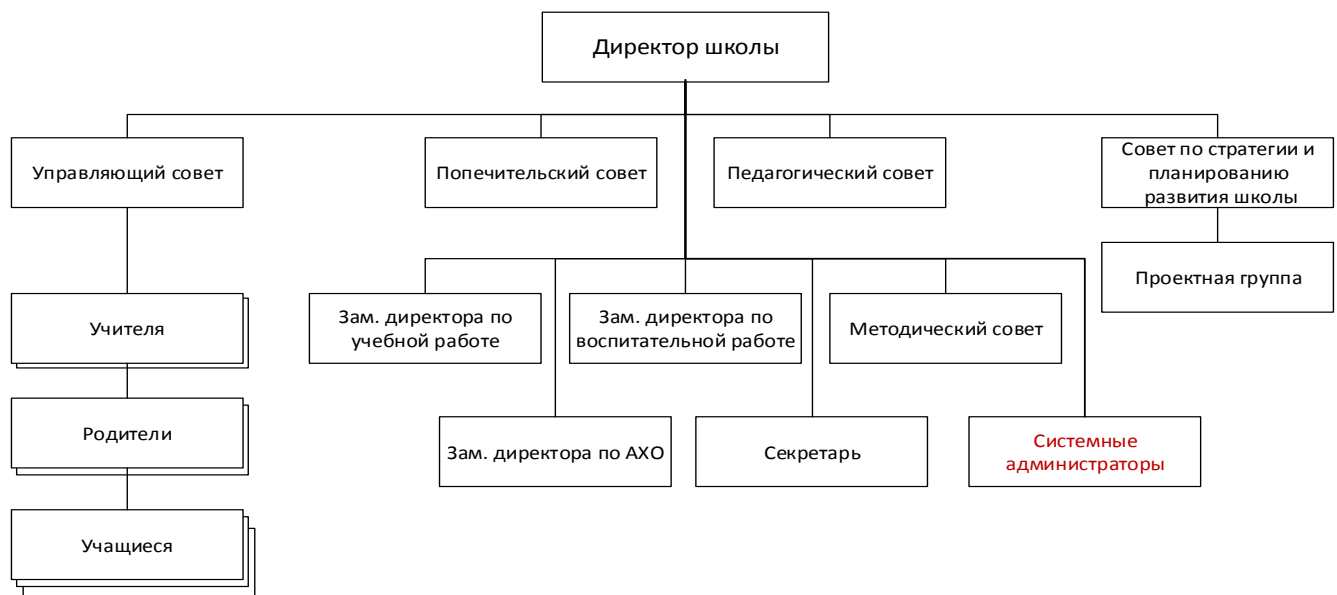


Рисунок 1.1 – Организационная структура МКОУ «Уйская СОШ имени А.И.Тихонова»

Организация управленческой деятельности в школе осуществляется на основе демократизации процессов управления, сочетания управления и самоуправления, исходя из задач, поставленных перед администрацией и



коллективом школы. Организационная структура управляющей системы школы состоит из четырех уровней управления.

Первый уровень:

Директор как главное административное лицо, осуществляющее непосредственное руководство школой и несущее персональную ответственность за все, что делается в школе всеми субъектами управления. На этом же уровне находятся коллегиальные и общественные органы управления:

Управляющий совет школы - коллегиальный орган управления муниципального общеобразовательного учреждения «Средняя общеобразовательная школа №289», состоящий из представителей обучающихся, их родителей (законных представителей) всех ступеней общего образования, педагогических работников школы и представителей общественности, осуществляющий общее руководство школой. Управляющий совет утверждает концепцию развития школы, долгосрочные образовательные программы, локальные акты учреждения и принимает решения по другим важным вопросам жизни школы, не отнесенным к компетенции директора. Кроме того, осуществляет общественный контроль за деятельностью администрации школы.

Педагогический совет – высший орган педагогического самоуправления, членами которого являются все учителя и воспитатели школы, а председателем – директор. На своих заседаниях педсовет рассматривает сложные педагогические и методические вопросы, проблемы организации учебно-воспитательного процесса, определяет порядок промежуточной и итоговой аттестации учащихся;

Общее собрание (конференция) родителей – высший орган родительского самоуправления, обсуждает кандидатуры от родителей в Попечительский совет, Управляющий совет, осуществляет контроль за целевым использованием средств Попечительским советом, разрабатывает рекомендации и предложения об изменении и дополнении Устава Школы,

локальных актов Школы, регламентирующих организацию образовательного процесса.

Попечительский совет – добровольное объединение спонсоров для содействия внебюджетному финансированию школы, помогает совершенствованию материально-технической базы, оказывает социальную поддержку всем участникам образовательного процесса и позволяет установить расширенное взаимодействие с внешкольной средой.

Общее собрание (конференция) обучающихся – высший орган ученического самоуправления. Исполнительным органом ученического самоуправления является Президентский Совет, который оказывает содействие администрации школы, педагогическому коллективу по включению каждого обучающегося в реализацию поставленной цели и подготовке их к преодолению жизненных трудностей, являясь инициатором коллективно творческих дел.

Второй уровень – заместители директора школы по УВР, воспитательной работе, АХО, педагоги-организаторы внеклассной работы, социальный педагог, психологи, организатор ОБЖ и органы, входящие в сферу влияния каждого из членов организации: методический совет, аттестационная комиссия, совет по профилактике правонарушений, комиссия по доплатам и надбавкам. Через этих членов школьной администрации директор осуществляет опосредованное руководство школьной системой.

Третий уровень – классные руководители, воспитатели педагоги дополнительного образования, руководители школьных методических объединений и творческих групп, которые, с одной стороны, выполняют организационно-управленческие функции, взаимодействие с органами общественного управления и самоуправления, а также с родителями и педагогами внешкольных институтов воспитания, а с другой стороны, осуществляют контроль и самоконтроль изменений в учебно-воспитательном процессе и формируют, и развивают деловые качества учащихся. Руководство

на этом уровне часто совпадает с лидерством, влияние которого шире по значению и богаче по содержанию, чем обычное управленческое влияние.

Четвертый уровень – учащиеся, органы ученического самоуправления в классах, члены кружков, студий, объединений не только в школе, но и в микрорайоне школы. Участие учащихся в управляющей системе школы и класса обеспечивает формирование и развитие организаторских способностей и деловых качеств личности.

Функциональные обязанности каждого члена администрации четко определены, что помогает им проявлять самостоятельность при принятии управленческих решений, повышает ответственность за свою деятельность.

Характеристика деятельности школы показана на рисунке 2.

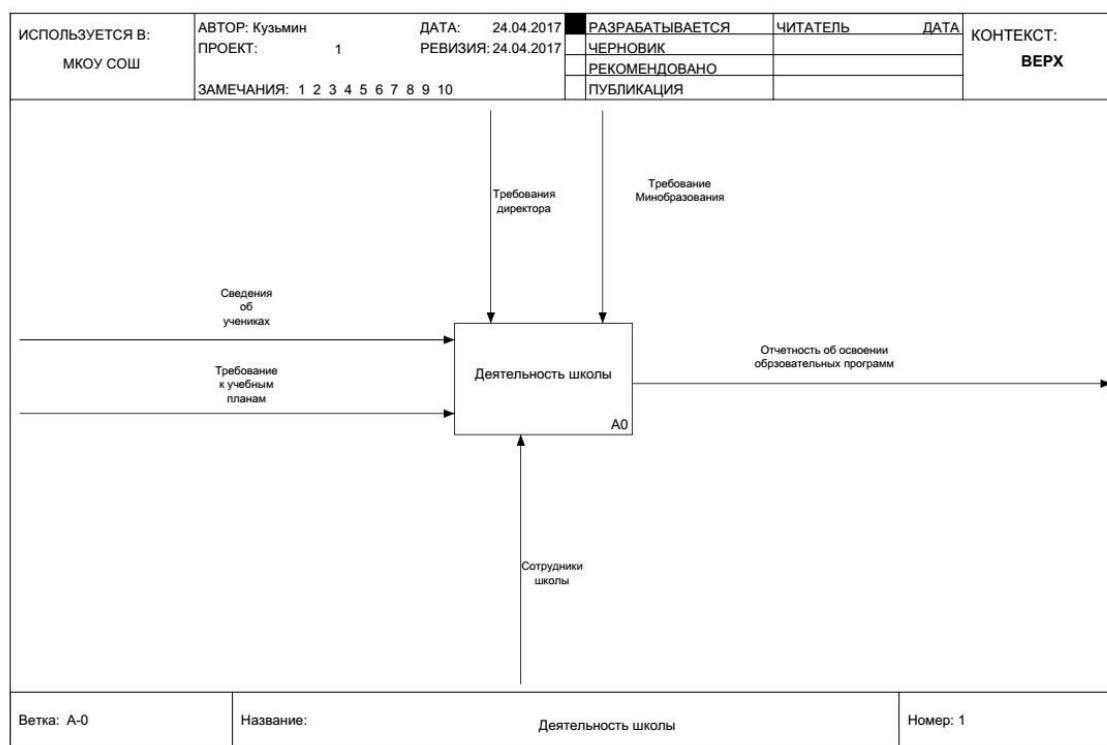


Рисунок 1.2 Характеристика деятельности школы

Для достижения целей деятельности, школа осуществляет следующие основные виды деятельности (рисунок 3):

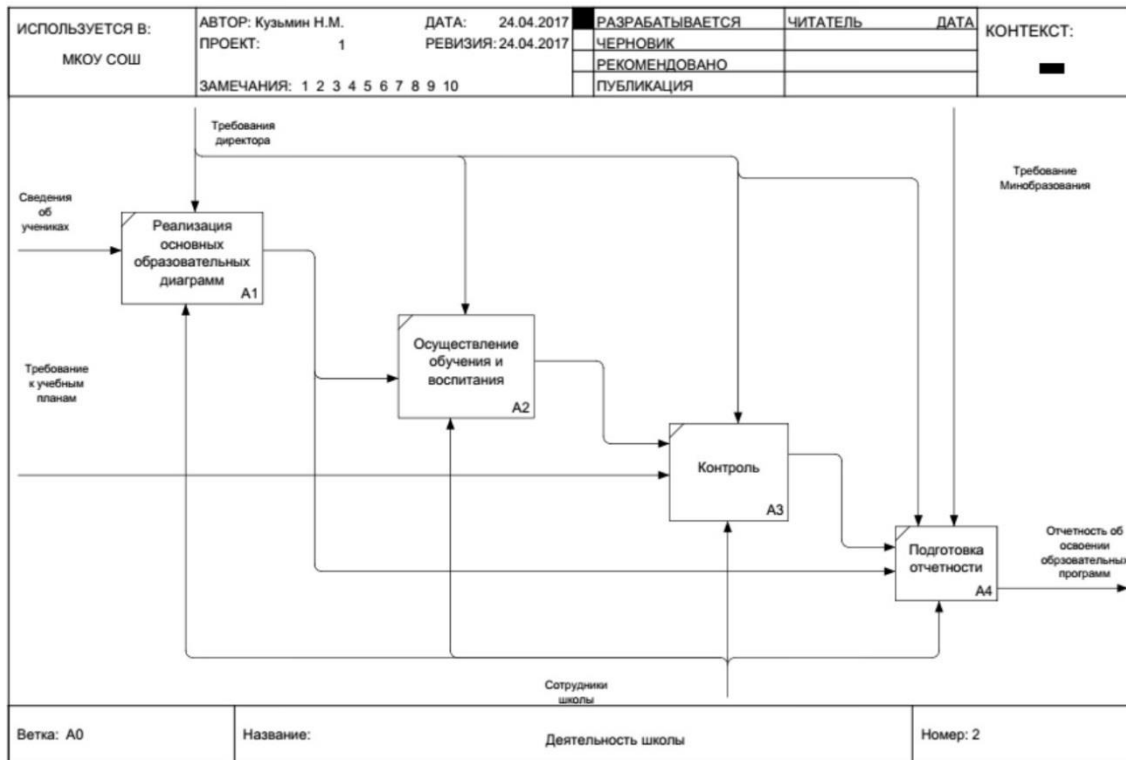


Рисунок 1.3 Декомпозиция деятельности школы

- реализация основных общеобразовательных программ начального общего, основного общего, среднего (полного) общего образования, обеспечивающих дополнительную (углубленную) подготовку обучающихся по одному или нескольким предметам;
- реализация основной общеобразовательной программы дошкольного образования;
- реализация образовательной программы профессиональной подготовки;
- использование и совершенствование методик образовательного процесса и образовательных технологий, в том числе с использованием дистанционных образовательных технологий;
- разработка и утверждение образовательных программ и учебных планов;
- разработка и утверждение рабочих программ учебных курсов, предметов, дисциплин (модулей);

- разработка и утверждение годовых календарных учебных графиков;
- выявление обучающихся, находящихся в социально-опасном положении, а также не посещающих или систематически пропускающих по неуважительным причинам занятия, принятие мер по их воспитанию, получению ими образования в рамках реализуемых образовательных программ;
- выявление семей, находящихся в социально-опасном положении, и оказание им содействия в обучении и воспитании детей;
- материально-техническое обеспечение и оснащение образовательного процесса, оборудование помещений в соответствии с государственными и местными нормами и требованиями;
- медицинская деятельность для реализации целей и задач Учреждения.

### **1.1.3 Описание средств автоматизации работы МКОУ «Уйская СОШ имени А.И.Тихонова»**

К средствам технической архитектуры школы относятся сервера, персональные компьютеры работников школы, сетевое оборудование (кабель и маршрутизаторы).

Укрупненная схема технической архитектуры предприятия представлена на рисунке 4.

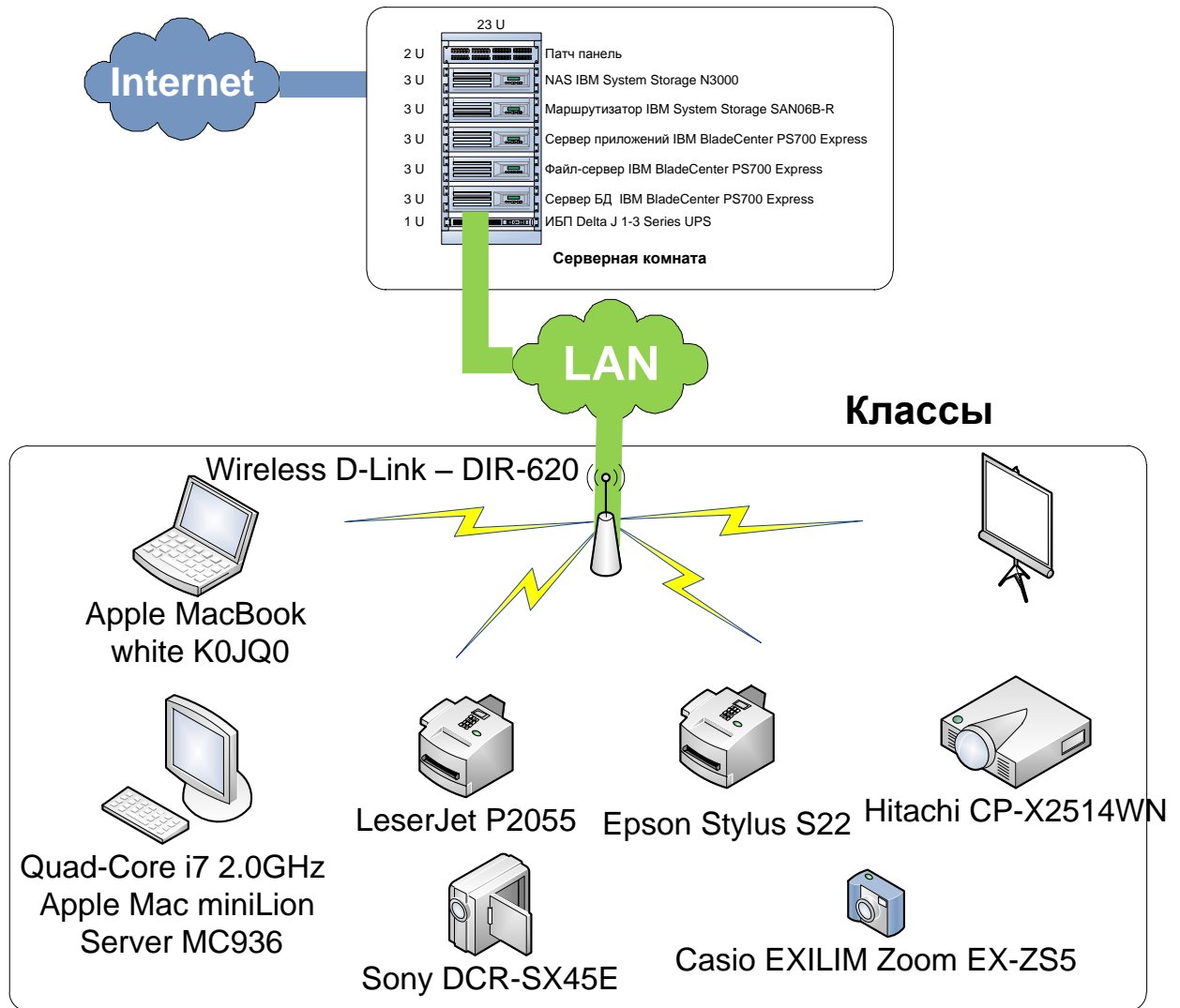


Рисунок 1.4 Структурная схема технической архитектуры школы

Локальная вычислительная сеть построена на основе многопротокольного маршрутизатора в стойке IBM System Storage SAN06B-R.

Данное сетевое устройство обеспечивает высокую производительность, имеет FC-порты со скоростью передачи данных до 8 Гбит/с и аппаратную обработку данных для работы портов 1 Gigabit Ethernet (GbE) IP со скоростью физической линии.

Кроме того, в серверной комнате в телекоммуникационной стойке расположены следующие устройства:

- blade-серверы IBM BladeCenter PS700 Express (для организации файлового сервера, сервера баз данных, сервера приложений);

- модульная дисковая система хранения IBM System Storage N3000
- ИБП Delta J 1-3 Series UPS
- патч-панель.

Перечень оборудования, используемого в классах школы, приведен в таблице 1.1

Таблица 1.1

## Перечень оборудования, используемого в школе

Наименование	Количество в классе
Ноутбук 13" Apple MacBook white K0JQ0	18 шт.
Рабочая станция Quad-Core i7 2.0GHz Apple Mac mini Lion Server MC936	1 шт.
Принтер лазерный ч/б HP LeserJet P2055	1 шт.
Принтер струйный цветной Epson Stylus S22	1 шт.
Адаптер Ethernet Apple – MC704 USB	1 шт.
Маршрутизатор Wireless D-Link – DIR-620	1 шт.
Внешний HDD 500	2 шт.
Сканер А4 HP Scanjet G3110	4 шт.
Планшет 6" Wacom Bamboo Pen A6 Wide (Small)	5 шт.
Наушники mini Philips – SHP1900	18 шт.
Микрофон USB VRN VRN-MIC3	3 шт.
Доска маркерная 120x240	1 шт.
Проектор мультимедийный Hitachi CP-X2511N	1 шт.
Стол для проектора PT9 Braun	1 шт.
Приставка для копи-режима Mimio Capture	1 шт.
Приставка интерактивная Mimio Tech	1 шт.
Копи-устройство Mimio Xi Mimio	1 шт.
Микроскоп цифровой Digital Blue	7 шт.
Видеокамера Sony DCR-SX45E	2 шт.
Сумка для видеокамеры	2 шт.
Фотокамера цифровая Casio EXILIM Zoom EX-ZS5	2 шт.
Штатив для видеокамеры	2 шт.
Штатив для фотокамеры	2 шт.

Существующая схема программной архитектуры информационной системы приведена на рисунке 5.

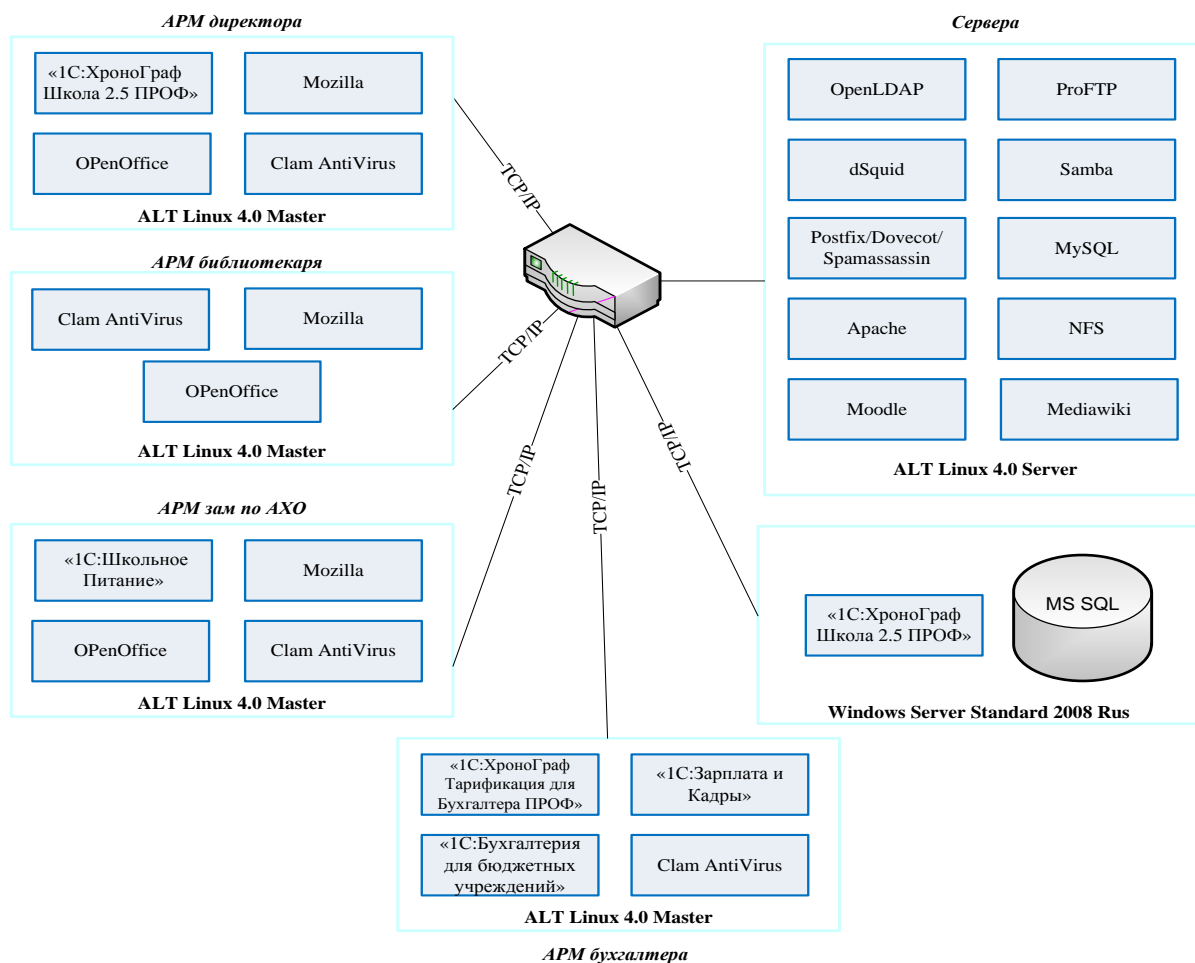


Рисунок 1.5 Схема программной архитектуры информационной системы

Для автоматизации администрирования управления предназначен программный комплекс «1С: Управление школой».

Программные продукты «1С:ХроноГраф Школа 2.5 ПРОФ» устанавливаются на выделенный сервер, остальные продукты – на сетевые рабочие станции.

На серверах также программное обеспечение, приведенное в таблице 2.

Таблица 1.2

#### Программное обеспечение серверов

№ пп	Наименование	Назначение
------	--------------	------------



1.	Базовая операционная система	Школьный Сервер
2.	ОС для IC	Windows Server 2003
3.	LDAP-сервер	OpenLDAP
4.	FTP-сервер	ProFTPd
5.	Прокси-сервер	Squid
6.	Samba-сервер	Samba
7.	почтовый сервер с элементами анализа содержимого писем	Postfix/Dovecot/Spam assassin
8.	СУБД с достаточными для функционирования платформы технологическими характеристиками	MySQL
9.	Веб-сервер с достаточными для функционирования платформы технологическими характеристиками	Apache
10.	Локальный репозиторий ПСПО со средствами синхронизации с внешним репозиторием ПСПО в сети Интернет для обновления и доустановки пакетов ПСПО на компьютерах ОУ	NFS
11.	Модульная объектно-ориентированная динамическая учебная среда	Moodle
12.	Управляемая веб-среда для организации взаимодействия участников учебного процесса	Mediawiki

Каждый пользователь имеет разный объем прав и может выполнять свою работу независимо от других пользователей.

## **1.2 Описание проектируемой системы оценки качества обучения**

### **1.2.1 Назначение системы оценки качества обучения**

Качество обучения в школе проверяется путем оценки и учета контрольных мероприятий. К таким относятся контрольные работы, а также различного рода тестирование.

При ведении бумажного каталога таких поиск работы в каталоге весьма затруднен. Поиск может понадобиться в том случае, если нужно сверить данные работы, обосновать верность предоставленной оценки, получить

статистику по выполнению контрольных мероприятий отдельным учеником, классом или по другому критерию.

Отсутствие связи между бизнес-процессами, материальными потоками и документами влечет за собой целый комплекс проблем:

- несоответствие действительного состояния материальных потоков документам;
- документально незафиксированные изменения в материальных потоках (в том числе изменения в процессах);
- мучительный поиск связанных между собой документов по различным классам;
- неточности и дублирования, высокая вероятность ошибок и несоответствий;
- отсутствие целостной картины состояния дел и проблематичность прогнозирования и принятия управленческих решений;
- непрозрачность бизнес-процессов учета контрольных мероприятий и подготовки отчетности;

Методист, как правило, тратит очень много времени на поиск необходимых данных по контрольным мероприятиям (по статистике на это уходит до 40% рабочего времени).

Проведем расчеты, позволяющие оценить возможную эффективность внедрения автоматизированного варианта учета контрольных мероприятий.

Допустим, что в течение рабочего дня методистом учитывается около 30 контрольных мероприятий.

Проведем расчет ожидаемого эффекта от внедрения средств автоматизации (таблица 1.3.). Время приведено с учетом за рабочий день.

Таблица 1.3

#### Расчет эффекта внедрения

Действие методиста	Время, затрачиваемое при базовой технологии/ мин.	Время, затрачиваемое по предлагаемому варианту/ мин.
--------------------	---	--

Учет темы контрольного мероприятия	55	14
Учет предмета контрольного мероприятия	40	10
Учет автора контрольного мероприятия	40	10
Учет проверяющего контрольного мероприятия	25	6
Учет оценки контрольного мероприятия	15	4
Подготовка списка контрольных мероприятий по критериям	25	6
Итого, минут:	200	50

Таким образом, при существующей системе на совершение указанных в таблице действий необходимо потратить более 3 часов. При внедрении предлагаемой системы это время сокращается до 50 минут, то есть в 4 раза.

Все эти задачи успешно могут быть выполнены в случае ведения автоматизированного учета при помощи специального приложения. При этом все данные о работах хранятся в базе данных, а приложение позволяет заносить информацию в базу данных и работать с ней.

Введение автоматизированного учета выполненных контрольных мероприятий повысит эффективность их поиска, ведения их каталога и отбора работ по определенным критериям.

В ходе дипломного проектирования должно быть разработано приложение, позволяющее автоматизировать ведение учета контрольных мероприятий. Учет контрольных мероприятий должен вестись в виде каталога выполненных работ, хранящего развернутый набор данных о каждой работе.

### **1.2.2 Цели создания системы оценки качества обучения\**

Целью разработки и внедрения информационной системы оценки качества обучения является снижение трудоемкости по учету контрольных мероприятий и их поиска.

Задачами разработки являются:

- автоматизация учет контрольных мероприятий;
- автоматизация построения отчетов по работам;
- автоматизация поиска по учтённым работам.

### **1.2.3 Требования к структуре и функционированию системы оценки качества обучения**

Использование информационной системы оценки качества обучения должно позволить упростить поиск работ в каталоге и их отбор по различным критериям за счет:

- автоматизации ведения каталога выполненных работ;
- ведения списков учеников, учителей, классов и выбор данных из них при внесении новых работ и поиске существующих работ;
- повышения оперативности доступа к информации о выполненных работах;
- автоматизации процесса поиска и отбора данных;
- обеспечения полноты и актуальности данных, необходимых для принятия управленческих решений;
- применения современных средств анализа информации;
- наглядного и удобного представления информации;
- печати списка работ и карточки каждой работы;
- получения развернутой информации о работе путем прикрепления к записям базы данных документов самой работы.

Программа ведения автоматизированного каталога выполненных работ должна иметь развитый и удобный пользовательский интерфейс. Современные пользователи привыкли, что Windows-программы имеют очень хорошие интерфейсы и соревнуются друг с другом в этом вопросе. При создании интерфейса разрабатываемой программы следует прежде всего изучить интерфейс самой операционной системы Windows, так как многие приложения, входящие в ее состав являются образцом для написания собственных приложений. Не следует пренебрегать такими элементами графического интерфейса как иконки и картинки. Это позволит повысить воспринимаемость данных.

Ведение списка связанных документов позволит повысить информативность каталога. Если информации в базе данных по выполненным работам недостаточно, пользователь всегда сможет получить развернутую информацию о работе из самих документов работы. Важно обеспечить открытие приложений, связанных с документами, непосредственно из каталога выполненных работ.

В результате выполнения дипломного проекта должно быть разработано Windows-приложение, выполняющее ведение автоматизированного каталога выполненных контрольных мероприятий, которое должно позволять выполнять следующие функции:

- ведение справочников классов, в которых выполнялись работы, обеспечение удобного поиска в этом справочнике класса по его наименованию;
- ведение списка учителей, обеспечение удобного поиска учителя в списке по его фамилии, имени или отчеству;
- ведение списка предметов обучения, обеспечение удобного поиска предмета по его названию;
- ведение списка авторов работ (учеников), обеспечение удобного поиска автора по его фамилии, имени или отчеству;

- ведение списка выполненных контрольных мероприятий, представление информации о работах в виде списка, в котором отображены основные данные о работе: наименование, дата сдачи, фамилия, имя, отчество, класс, дата сдачи, предмет, оценка;
- обеспечение удобного поиска работы в списке по любому из полей, хранящемуся в базе данных;
- обеспечение фильтрации и отбора работ по любому из полей, хранящихся в базе данных;
- ввод данных о новой работе и редактирование существующей в форме редактирования работы, в которой отображаются все поля, хранящиеся в базе данных работ.
- обеспечения хранения в базе данных путей к связанным документам, ведение списка таких документов, обеспечение просмотра списка связанных документов как в режиме просмотра списка работ, так и в режиме просмотра карточки документа;
- запуск приложений, обеспечивающих просмотр и редактирование связанных документов;
- обеспечение вывода на печатающее устройство списка выполненных работ с указанием основных данных работ;
- обеспечение вывода на печатающее устройство карточки работы с выводом всех данных, хранящихся в базе данных работ.

#### **1.2.4 Требования к функциям, выполняемым системой оценки качества обучения**

Для обеспечения ИС оценки качества обучения от несанкционированного доступа необходимо предусмотреть авторизацию пользователя при входе в систему.

Администратор при резервном копировании и архивировании информации базы данных должен защитить ее служебным паролем.

В системе требуется предусмотреть возможность разграничения полномочий пользователей. Необходимо, чтобы администратор мог распределять уровни доступа к информации каждого из пользователей и наделять им права доступа к системе.

В системе должны быть предусмотрены две группы пользователей:

- Администратор, обладающий всеми возможностями, по изменению данными в программе и получению всех отчетов;
- Методист, выполняющий функции по учету контрольных мероприятий, и получению отчетов за свою деятельность.

Сохранность информации при авариях необходимо обеспечивать путём резервного копирования данных. Резервное копирование данных должно осуществляться автоматически ежедневно либо в любой момент времени администратором системы по его инициативе.

Для нормального функционирования ИС сервер баз данных должен удовлетворять техническим требованиям, предъявляемым к аппаратному обеспечению, а именно:

- процессор не ниже Intel Core i3 не ниже 3.30 GHz. Рекомендуется использовать многопроцессорные сервера;
- оперативная память не менее 4096 Мбайт (рекомендуется 8128 Мбайт и выше). Компьютер конечного пользователя должен удовлетворять следующим техническим требованиям:
  - Intel Celeron Dual-Core, 2000Mhz, 1024 Мб RAM, 160 Гб HDD, VGA видеоадаптер, монитор LCD, клавиатура, мышь.

Для обеспечения сетевой работы ИС компьютеры возможных пользователей системы должны быть объединены в ЛВС. В качестве сетевого интерфейса необходимо применить сетевой протокол TCP/IP с пропускной способностью 10/100 Мбит/с.

Контроль и приемка разработки осуществляется на основании испытаний. В ходе контрольно-отладочного тестирования последовательно выполняются все функции программы.

В ходе контроля программы должны быть проведены следующие виды тестирования:

1. Тестирование в нормальных условиях
2. Тестирование в исключительных условиях, в том числе:
3. Тестирование нулевыми данными
4. Тестирование чужими данными
5. Тестирование избыточными данными
6. Оценка полноты проверки программы

### **1.3 Сравнительный анализ аналогичных программных продуктов**

**NetSchool** – это комплексная информационная система для современной школы. Ключевые достоинства NetSchool:

- электронный журнал и электронный дневник, тщательно апробированные с 2002 г.;
- проработанная аналитическая отчётность (более 40 автоматически формируемых отчётов для администрации школы, завучей, классных руководителей, учителей, учащихся, родителей);
- интеграция с другими программами (системы тестирования, учебные курсы, программы составления расписания, системы контроля доступа и др.);
- sms-сервис для родителей (не только sms-рассылки от классного руководителя, но и возможность для родителя получать информацию о своём ребёнке по sms-запросу).

Каждый пользователь имеет индивидуальные имя и пароль для входа в NetSchool. Права доступа к разным частям базы данных школы могут гибко настраиваться.



**Дневник.ру** - Единая образовательная сеть России, которая формирует уникальную электронную среду для учителей, учеников и их родителей. Разработка проекта началась в 2007 году и уже в 2009 проект был запущен под эгидой приоритетного национального проекта «Образование». Дневник.ру поддерживается Полномочным Представительством Президента РФ в СЗФО, региональными администрациями, министерствами, комитетами и департаментами образования.

Дневник.ру совмещает в себе три модуля:

- дистанционное обучение,
- управление школьным документооборотом,
- социальная сеть.

Базовый функционал Дневник.ру бесплатный для всех участников образовательного процесса. Пользователям доступны электронный классный журнал и электронный дневник учащегося, а также медиатека, библиотека образовательной литературы, онлайн-тренинг тестирования ЕГЭ, возможность пройти вступительные олимпиады в крупнейшие ВУЗы России.

**Проект «SmileS.Школьная карта»** - это уникальная система безопасности школы и коммуникаций между учениками, учителями и родителями.

Проект внедряет в школах программное обеспечение и специальное для его работы оборудование.

#### 1. «Электронный журнал» и «Электронный дневник»

Программное обеспечение для коммуникаций между учителями, учениками, родителями. Учитель на компьютере выставляет оценки, замечания, вносит домашнее задание в электронный журнал прямо на уроке – данные мгновенно отображаются в электронном дневнике школьника (Личном кабинете родителя). Здесь же, в Личном кабинете, доступно расписание школы, данные всех учителей и администрации, новости и

объявления класса и школы; можно оставить сообщение учителю или классному руководителю.

Родители могут заказать услугу "СМС-информирование" и получать на свой телефон оценки и замечания ребенка сразу после выставления их учителем в электронный журнал.

## 2. «Школьное питание»

Система (программное обеспечение+оборудование) для безналичной оплаты питания в школьной столовой и продуктов в буфете. Родители кладут определенную сумму денег на личный счет ребенка, привязанный к его персональной карте. Школьник с помощью этой карты оплачивает питание в школьной столовой и буфете.

Родители могут заказать на свой телефон услугу "СМС-информирование" и получать на свой телефон сообщения о том, на какие продукты и сколько денег потратил ребенок.

## 3. «Электронная библиотека»

Программное обеспечение, позволяющее вести учет книг, карточки учащихся, необходимую документацию в компьютере в электронном виде. Существенно упрощает работу библиотекаря.

Проект "SmileS.Школьная карта" направлен на родителей учеников, чтобы сделать процесс обучения более контролируемым и управляемым с их стороны.

**«АВЕРС: Электронный Классный Журнал»** - это новая система взаимодействия между школьным и родительским сообществом, которая позволяет образовательному учреждению автоматизировать составление школьной отчетности, оперативно получать различные статистические срезы с целью повышения качества управленческой деятельности, вовлекает образовательное учреждение в процесс электронизации хранения и обработки данных.

Каждый пользователь имеет свой логин и пароль. В зависимости от роли пользователя, система ограничивает доступ к определенным возможностям.

В программе предусмотрены следующие роли: Администратор, Директор, Учитель, Учащийся/Родитель.

Серверная часть системы устанавливается локально на сервере образовательного учреждения. Программа работает в рамках локальной сети учреждения.

Основываясь на проведенном обзоре, можно сделать следующие выводы о применимости данных систем для решения поставленной задачи разработки системы кафедры:

1. Современные тенденции развития рынка OpenSource LMS\LCMS направлены в сторону универсализации и увеличения функциональности систем.

2. Использование коммерческих систем управления электронным обучением не доступно кафедре по причине их высокой стоимости и необходимости продления лицензии на каждый учебный год.

3. Системы с открытым исходным кодом позволяют реализовать тот же набор возможностей, что и коммерческие с существенно меньшими затратами и большей эффективностью, но, тем не менее, не подходят из-за трудоемкости развертывания и излишней функциональности;

4. Общим недостатком систем является невозможность расширения без участия разработчиков.

Сравнение перечисленных систем приведено в таблице 4.

Сравнение систем автоматизации деятельности школы

		NetSchool	Дневник.ру	SmileS.Школьная Карта	Аверс
Функционал системы	Описание	<a href="http://www.net-school.ru">http://www.net-school.ru</a>	<a href="http://www.dnevnik.ru">http://www.dnevnik.ru</a>	<a href="http://www.shkolnaya-karta.ru">http://www.shkolnaya-karta.ru</a>	<a href="http://avers-journal.ru">http://avers-journal.ru</a>
Личные кабинеты	Раздел на сайте компании, вход в который осуществляется по личному логину и паролю	есть	есть	есть	есть
Личный кабинет учителя	Раздел для учителя	есть	есть	есть	есть
Личный кабинет ученика	Раздел для ученика	есть	есть	есть	нет
Личный кабинет родителя	Раздел для родителя	есть	есть	есть	есть
Социальная сеть	Сайт, объединяющий участников учебного процесса школ, в которых подключена система	нет	есть	есть	нет
Формирование отчетов	Отчеты для организаций сферы образования, внутришкольные отчеты, отчеты классного руководителя, отчеты успеваемости для родителей	есть	есть	есть	есть
Составление расписания	Генератор уроков для быстрого наполнения пустого расписания на основе схемы расписания	есть в интеграции с программой составления расписания	есть	есть	есть в интеграции с программой составления расписания
Портфолио ученика	Контактные и личные данные учеников и их родителей	есть	есть	есть	нет
Дистанционное обучение	Взаимодействие учителя и учеников на расстоянии	есть	есть	нет	нет

Основываясь на проведенном обзоре, можно сделать следующие выводы о применимости данных систем для решения поставленной задачи разработки ИС оценки качества обучения:

1. Современные тенденции развития рынка электронных дневников направлены в сторону универсализации и увеличения функциональности систем.

2. Использование коммерческих систем управления электронным обучением не доступно по причине их высокой стоимости и необходимости продления лицензии на каждый учебный год.

3. Системы с открытым исходным кодом позволяют реализовать тот же набор возможностей, что и коммерческие с существенно меньшими затратами и большей эффективностью, но, тем не менее, не подходят из-за трудоемкости развертывания и излишней функциональности;

4. Общим недостатком систем является невозможность расширения без участия разработчиков.

## **Выводы по главе 1**

В первом разделе работы была рассмотрена деятельность «Уйской средней общеобразовательной школы имени А.И.Тихонова», ее структура и используемые средства автоматизации. Была описана проектируемая система оценки качества обучения и ее назначение. В главе так же проведен расчет ожидаемого эффекта от внедрения средств автоматизации. Были сформулированы цели разработки и внедрения ИС, задачи разработки и требования к информационной системе оценки качества обучения, а также ее основные функции.

В завершении главы был проведен анализ аналогичных программных продуктов и составлена таблица сравнения.

## **2 Проектирование системы оценки качества обучения**

### **2.1 Выбор архитектуры информационной среды и инструментальных средств реализации**

Проектируемую систему можно разбить на отдельные составляющие подсистемы, функции которой заключаются в следующем:

- представление аутентификации и доступа для зарегистрированного пользователя;
- обработка введенной информации и запись ее в базу данных;
- тестирование учеников и получение их результатов;
- формирование отчетов.

Так как системой должно будет пользоваться несколько пользователей на различных компьютерах, оптимальной для системы будет двухуровневая архитектура «клиент-сервер», более конкретно – «тонкий клиент – толстый сервер». В этом случае все данные по распределениям будут храниться в общей базе данных, в то же время каждый пользователь будет работать за своим рабочим местом.

Технология клиент-сервер в отличие от файл-серверной позволяет реализовать большую безопасность, стабильность, согласованность, расширяемость, повышенную защищенность и надежность обработки и хранения данных. Внедрение систем с архитектурой клиент-сервер в своем большинстве обязано одному проверенному факту: использование недорогих серверов и недорогих клиентских ПК дает возможность получения оптимального соотношения производительности и итоговой стоимости.

Технология "клиент-сервер" делит ИС на два уровня: хранения и представления данных. На первом уровне располагаются клиентские компьютеры с прикладным ПО, при помощи которого пользователи работают

по сети с базой данных. Второй уровень включает в себя сервер с размещенной на нем базой данных.

В двухуровневой архитектуре присутствуют два звена: это клиентская ЭВМ и сервер СУБД.

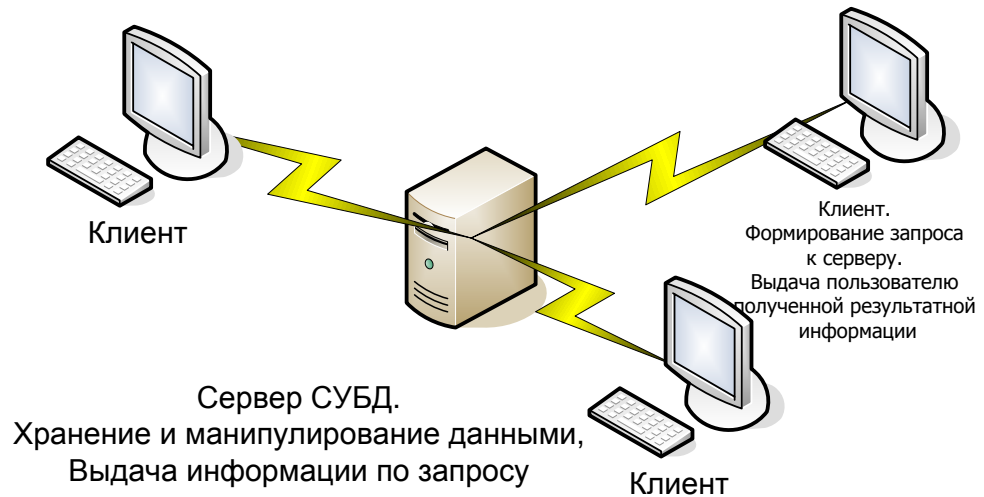


Рисунок 2.1 Двухуровневая архитектура «клиент-сервер»

В свою очередь, двухуровневая архитектура может быть устроена по-разному:

- по принципу «тонкий сервер»-«толстый клиент» (Рисунок 2.2);
- по принципу «толстый сервер»-«тонкий клиент» (Рисунок 2.3).

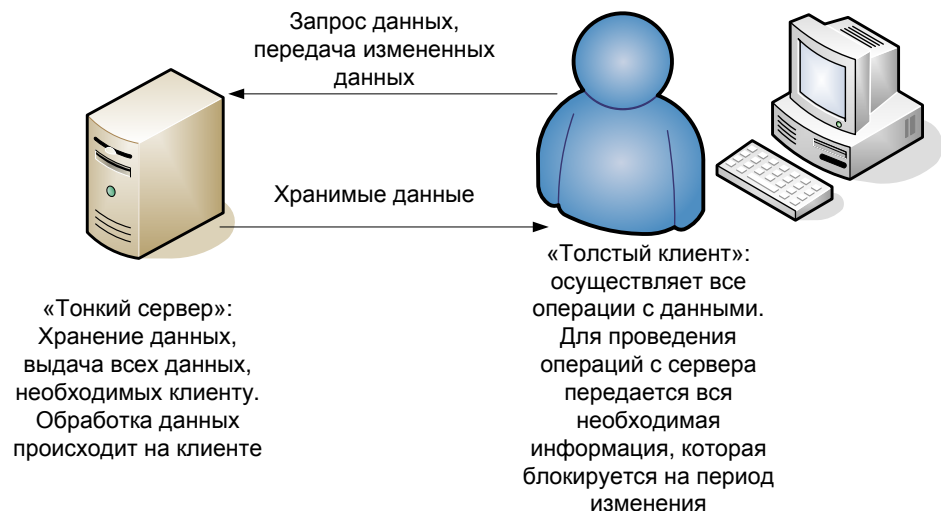


Рисунок 2.2. Двухуровневая архитектура «тонкий сервер»-«толстый клиент»



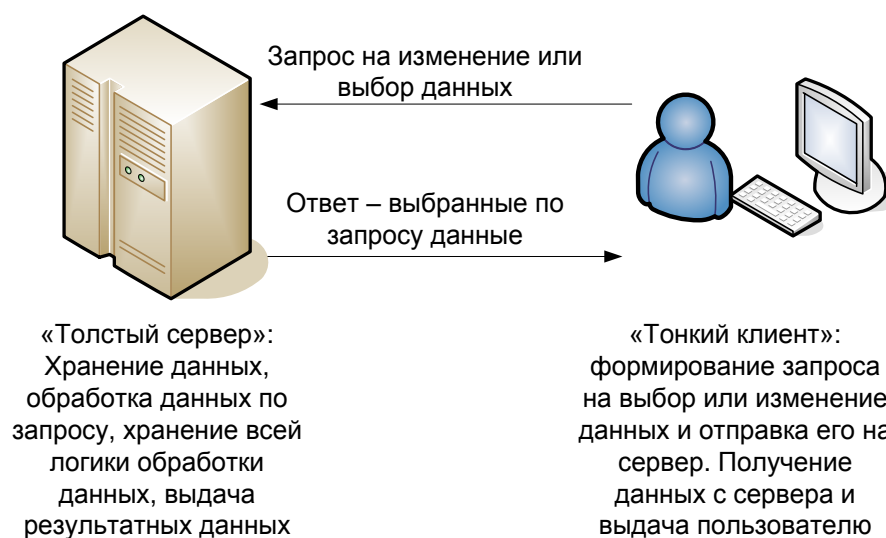


Рисунок 2.3. Двухуровневая архитектура «толстый сервер»-«тонкий клиент»

Архитектура «тонкий клиент»-«толстый сервер» является наиболее предпочтительной по ряду причин:

- большее быстродействие и производительность системы за счет того, что все операции манипулирования и выбора данных происходят на сервере, а клиенту передается только результат запроса;
- не происходит блокировок информации на период изменения её одним из клиентов;
- каналы передачи данных не перегружаются из-за необходимости передачи всего объема данных для модификации на клиенте;
- логику функционирования системы можно изменить на сервере без перекомпиляции клиентского ПО, что особенно актуально в условиях территориальной удаленности рабочих мест.

Отличительной особенностью трехзвенной, или трехуровневой, архитектуры, является окончательное разделение функций хранения данных, их обработки и представления пользователю[7].

Построение системы автоматизации по трехуровневой схеме "клиент-сервер" представлено на рисунке 2.5. Доступ к данным осуществляется через промежуточное звено - сервер приложений, дает значительные преимущества по сравнению с двухуровневой при построении крупномасштабных ИС на

разных аппаратных платформах в условиях большого количества клиентов и их территориальной распределенности. В частности, такая архитектура позволяет обеспечивать дополнительную надежность и отказоустойчивость системы, неограниченно наращивать мощность системы и количество клиентов за счет использования нескольких серверов приложений. Также становится возможным перенос БД с одной СУБД на другую и изменения логики изменения данных без необходимости обновления клиентских программ.

Сервер приложений, кроме выполнения предметных задач, еще и значительно уменьшает нагрузку на сервер базы данных.

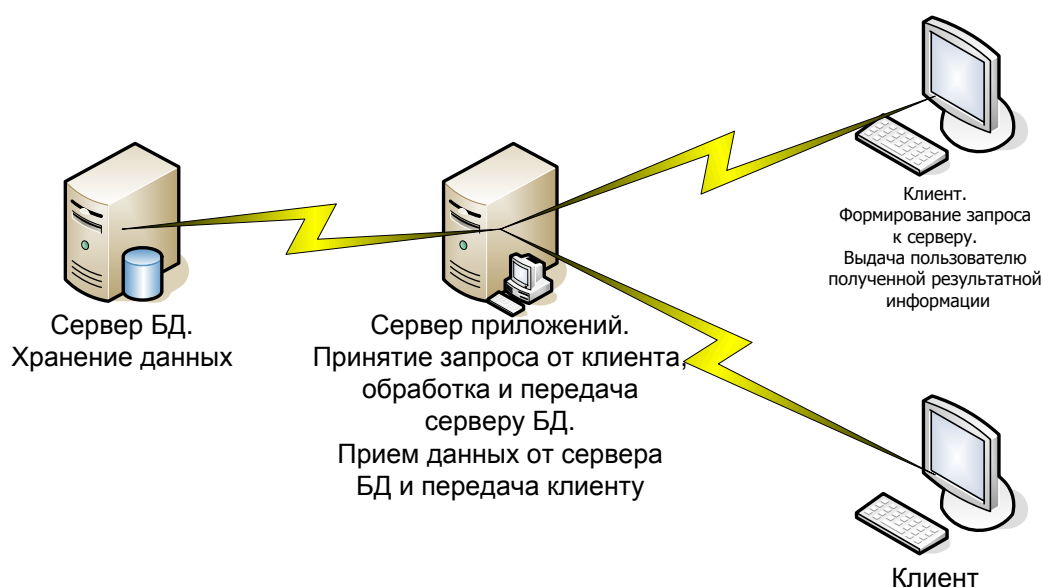


Рисунок 2.4. Трехуровневая архитектура «клиент-сервер».

Наиболее целесообразной для решения выбранной задачи является выбор двухуровневой архитектуры «тонкий клиент»-«толстый сервер», так как преимущества трехзвенной архитектуры не являются ценными для текущей задачи, а стоимость и сложность организации трехзвенной архитектуры существенно выше, чем двухуровневой. К тому же, существующая информационная инфраструктура отлично подходит именно для двухуровневой архитектуры.

Выбор программных средств, с помощью которых можно реализовать проектируемую АИС, основывается на следующих критериях[3]:

1. Возможность установки на ПК, не подключенном к сети.
2. Инструменты для работы непосредственно с объектом.
3. Набор средств для получения отчетов.
4. Возможность хранения первоначальных данных и результатов.
5. Графические возможности для построения статистических диаграмм.
6. Минимум затрат на приобретение и разработку.
7. Удобство использования и сопровождения.

Среди множества продуктов, предлагающих подобные возможности, можно выделить следующие направления:

- реализация через web-интерфейс, например с помощью php-программирования;
- реализация через интегрированные среды разработки программного обеспечения для Microsoft Windows на языках Delphi, VisualC++, VisualBasic и др.

Для выбора языка программирования сравним следующие языки – Java, C# и Visual Basic[3.3]. Сравнение языков программирования приведено в Таблице 2.1.

Таблица 2.1

## Сравнение языков программирования

Параметр	Степень соответствия, %		
	C#	Java	Visual Basic
Возможность компиляции	8	8	4
Многопоточная компиляция	8	0	8
Интерпретатор командной строки	6	4	0
Многомерные массивы	8	0	8
Динамические массивы	8	8	0
Ассоциативные массивы	4	0	0
Интерфейсы	8	0	4

Мультиметоды	8	0	0
Общая оценка	7,25	2,5	3

Для разработки и отладки программы применялась интегрированная система C#.

Выбор этого программного продукта обусловлен следующим [4]:

- исходный текст программы компилируется в промежуточный код, не зависящий от языка и платформы; этот код в дальнейшем выполняется в специальной управляемой среде.
- автоматический сбор мусора,
- отсутствие заголовочных файлов. Весь код помещается в пакеты (packages) и сборки (assemblies). Никаких проблем с порядком объявления классов в случае наличия перекрестных ссылок.
- многопоточность поддерживается путем блокирования объектов
- интерфейсы, множественная реализация интерфейсов классом, однократное наследование базового класса производным.
- внутренние классы
- отсутствие концепции наследования классов с заданным уровнем доступа.
- отсутствие глобальных функций и констант, все элементы должны принадлежать классам.
- массивы и строки со встроенной длиной и проверкой границ.
- не применяются операторы «->» , «::». Во всех случаях используется оператор «.».
- null и boolean / bool являются ключевыми словами.
- любая величина должна быть проинициализирована до того, как будет использована.
- нельзя использовать целые числа (integers) для управления операторами if.

В мире существует множество систем управления базами данных. Несмотря на то, что они могут по-разному работать с различными объектами

и предоставляют пользователю различные функции и средства, большинство СУБД опираются на единый устоявшийся комплекс основных функций.

В общем случае, перечень требований к СУБД, может изменяться в зависимости от поставленных целей. Тем не менее можно выделить несколько групп критериев:

- Структура данных;
- Функциональные возможности;
- Особенности разработки приложений;
- Производительность;
- Требования к рабочей среде.

Рассмотрим 5 различных реляционных СУБД. Согласно методу анализа иерархий, предложенному Т. Саати, проводится попарное сравнение всех СУБД по каждому критерию, в результате чего получается 5 матриц попарных сравнений альтернатив.

В качестве альтернатив рассмотрим следующие СУБД:

- 1) DB2;
- 2) Oracle;
- 3) Microsoft SQL Server;
- 4) MySQL;
- 5) PostgreSQL.

Сравним выбранные СУБД по критерию «Структура данных».

Все рассматриваемые альтернативы реализуют реляционную модель данных (РСУБД) или объектно-реляционную модель данных (ОРСУБД), следовательно, все рассматриваемые системы подходят для анализа и сравнения. Проводится анализ рассматриваемых альтернатив по предусмотренным типам данных. По результатам этого анализа можно построить матрицу попарных сравнений альтернатив по первому критерию (таблица 2.2), рассчитать вектор приоритетов, главное собственное значение и остальные показатели [3].

Таблица 2.2

Матрица попарных сравнений альтернатив по критерию «Структура данных»

	DB2	Oracle	MySQL	MS SQL	Postgre SQL
DB2	1	1	1	1	1
Oracle	1	1	1/4	1/5	1/3
MySQL	1	4	1	1/2	2
MS SQL	1	5	2	1	2
Postgre SQL	1	3	1/2	1/2	1

Вектор приоритетов: ОД 8 0,08 0,24 0,33 0,17

Главное собственное значение: 5,34. Индекс согласованности (ИС): 0,084. Отношение согласованности (ОС): 0,07. Как видно, ОС в пределах нормы.

Сравним выбранные СУБД по критерию «Функциональные возможности».

Пункт «Триггеры и хранимые процедуры» определяет наличие в некоторой СУБД класса процедур, функций. Проведем анализ альтернатив по данному пункту (Таблица 2.3).

Таблица 2.3

Анализ альтернатив по пункту «Триггеры и хранимые процедуры»

	Триггер	Функция	Процедура
DB2	+	+	+
Microsoft SQL Server	+	+	+
MySQL	+	+	+
Oracle	+	+	+
PostgreSQL	+	+	+

Пункт «Масштабируемость» предполагает возможности рассматриваемой СУБД по увеличению объема данных со временем и в случае необходимости [4]. Необходимо рассмотреть максимально возможный объем хранимых данных для каждой альтернативы (таблица 2.4).

Таблица 2.4

## Анализ альтернатив по пункту «Масштабируемость»

	Размер БД	Размер таблицы	Размер строки
DB2	512ТБ	512 ТБ	32677 В
Microsoft SQL Server	524258 ТБ	524258 ТБ	$\infty$
MySQL	$\infty$	256ТБ	64КВ
Oracle	$\infty$	4 Гб* Размер блока	8КВ
Postgre SQL	$\infty$	32 ТБ	1,6 ТБ

Таким образом, проведен анализ рассматриваемых альтернатив по пунктам критерия «Функциональные возможности». По результатам анализа можно построить матрицу попарных сравнений альтернатив по второму критерию (Таблица 2.5), рассчитать вектор приоритетов и основные показатели.

Таблица 2.5

## Матрица попарных сравнений альтернатив по критерию

## «Функциональные возможности»

	DB2	Oracle	MySQL	MS SQL	Postgre SQL
DB2	1	1/4	2	1/7	1/5
Oracle	4	1	1	1/4	1/2
MySQL	1/2	1	1	1/4	1/2
MS SQL	7	4	4	1	3
Postgre SQL	5	2	2	1/3	1

Вектор приоритетов: 0,07 0,13 0,09 0,49 0,22

Главное собственное значение: 5,45. Индекс согласованности (ИС): 0,11. Отношение согласованности (ОС): 0,09.

Рассмотрим критерий «Особенности разработки приложений». При рассмотрении этого критерия необходимо оценить трудозатраты на администрирование баз данных. Основные задачи такого администрирования: установка и конфигурирование базы данных, текущее администрирование базы данных, резервное копирование/восстановление [5].

Таким образом можно построить матрицу попарных сравнений альтернатив по третьему критерию (таблица 2.6), рассчитать вектор приоритетов и основные показатели.

Таблица 2.6

Матрица попарных сравнений альтернатив по критерию «Особенности разработки приложений»

	DB2	Oracle	MySQL	MS SQL	Postgre SQL
DB2	1	1	1	1/6	1
Oracle	1	1	1	1/4	1
MySQL	1	1	1	1/4	1
MS SQL	6	4	4	1	3
Postgre SQL	1	1	1	1/3	1

Главное собственное значение: 5,04. Индекс согласованности (ИС): 0,01.

Отношение согласованности (ОС): 0,01.

Сравним выбранные СУБД по критерию «Производительность».

Результаты теста производительности TPC-C представлены в таблице 2.7 [6].

Таблица 2.7

Результаты теста TPC

Название	Количество транзакций, tpmC	Стоимость транзакции, долл./tpmC	Монитор транзакций
Microsoft SQL Server 2005 x64	661,475	1.16USD	Microsoft COM+
Oracle Database Standard	631,766	1.08 USD	Microsoft COM+
IBM DB2 9.5	1,200,011	1.99 USD	Microsoft COM+

По имеющимся данным оценим рассматриваемые СУБД по критерию «Производительность», построим матрицу попарных сравнений альтернатив (Таблица 2.8).



Таблица 2.8

Матрица попарных сравнений альтернатив по критерию  
«Производительность»

по	DB2	Oracle	MySQL	MS SQL	Postgre SQL
DB2	1	4	5	3	5
Oracle	1/4	1	3	1/2	3
MySQL	1/5	1/3	1	1/4	1
MS SQL	1/3	2	4	1	4
Postgre SQL	1/5	1/3	1	1/4	1

Вектор приоритетов: 0,47 0,15 0,07 0,24 0,07

Главное собственное значение: 5,14. Индекс согласованности (ИС): 0,036. Отношение согласованности (ОС): 0,03.

Рассмотрим критерий «Требования к рабочей среде». В Таблице 2.9 приводятся результаты анализа альтернатив по критерию «Поддерживаемые операционные системы» [3].

Таблица 2.9

Поддерживаемые ОС рассматриваемых систем

	DB2	MS SQL Server	MySQL	Oracle	Postgre SQL
Windows	+	+	+	+	+
Mac OS	+	+	+	+	+
Linux	+	+	+	+	+
BSD	-	+	+	-	+
UNIX	+	+	+	+	+
AmigaOS	-	+	+	-	-
Symbian	-	+	+	-	-

Оценим рассматриваемые СУБД относительно критерия «Требования к рабочей среде», построим матрицу попарных сравнений альтернатив (Таблица 2.10).

Таблица 2.10

Матрица попарных сравнений альтернатив по критерию «Требования к рабочей среде»

	DB2	Oracle	MySQL	MS SQL	Postgre SQL
DB2	1	1	1/4	1/4	1/3
Oracle	1	1	1/4	1/4	1/2
MySQL	4	4	1	1	3
MS SQL	4	4	1	1	3
Postgre SQL	3	2	1/3	1/3	1

Пусть производительность имеет наибольшую важность по сравнению с другими критериями, требования к рабочей среде также важны, так как при выборе СУБД на ранних этапах поднимается вопрос о совместимости рассматриваемой системы с существующими программными и аппаратными средствами.

Построим матрицу попарных сравнений критериев (Таблица 2.11), для удобства пронумеруем критерии от 1 до 5.

Таблица 2.11

Матрица попарных сравнений критериев

	1	2	3	4	5
1	1	1	1/2	1/6	1/4
2	1	1	1/2	1/6	1/3
3	2	2	1	1/5	1/2
4	6	6	5	1	2
5	4	3	2	1/2	1

Вектор приоритетов альтернатив: 0,07 0,07 0,12 0,49 0,25

Главное собственное значение: 5,03. Индекс согласованности (ИС): 0,01.

Отношение согласованности (ОС): 0,01.

Таким образом, веса рассматриваемых СУБД распределены следующим образом: MySQL (0.32), DB2 (0.28), MS SQL Server (0.16), Oracle (0.13), PostgreSQL (0.11).

На основании данного сравнения выбираем для использования СУБД MySQL.

## 2.2 Описание модели данных

В ходе работы системы формируются входные и выходные потоки информации. Работа системы осуществляется на основании входных документов, а в результате работы системы формируются результатные документы.

Входной информацией для разрабатываемой системы являются данные пользователей, сведения об обучаемых, сведения о планируемых и проводимых занятиях, сведения об успеваемости каждого обучаемого. Эти данные поступают как в цифровом, так и в печатном виде.

В систему вносятся следующие данные об обучаемых:

- фамилия;
- имя;
- отчество;
- класс.

Также в систему вносятся данные о занятиях.

Результатными документами являются следующие:

- список обучаемых;
- список преподавателей;
- список предметов;
- результаты отчетности.

Список преподавателей содержит следующие реквизиты:

- фамилия
- имя и отчество
- дата рождения
- научное звание
- логин
- пароль

Список обучаемых содержит следующие реквизиты:

- фамилия, имя, отчество;
- дата рождения;
- класс;
- пароль и логин;
- виды и типы сданной отчетности.

Список планируемых занятий содержит следующие реквизиты:

- дата и время занятия;
- класс;
- предмет;
- руководитель;
- форма для регистрации результатов занятия.

Список выполненных работ включает в себя наименование типа работы, ее тему, дату и время проведения, количество вопросов (содержание) и результаты каждого обучаемого.

Информация, входящая в состав входных и выходных потоков информации, хранится в базе данных.

Логическая модель базы данных представляет собой сущности и связи между ними. В базе данных предусмотрены следующие сущности:

1) Сущность **Материал** включает следующие атрибуты:

- код\_материала
- код\_лекции
- номер
- наименование
- описание
- название\_файла
- дата\_файла
- размер\_файла
- код\_темы

- 2) Сущность Ответ включает следующие атрибуты:
  - код\_ответа
  - код\_вопроса
  - номер
  - наименование
  - правильный\_ответ
  - код\_типа\_вопроса
- 3) Сущность Ответы\_пользователя включает следующие атрибуты:
  - код\_вопроса\_пользователя
  - код\_ответа\_пользователя
  - код\_вопроса
  - код\_типа\_вопроса
- 4) Сущность Пользователи включает следующие атрибуты:
  - код\_пользователя
  - фамилия
  - имя
  - отчество
  - код\_типа\_пользователя
  - логин
  - пароль
  - блокировка
  - Email
  - регистрация
  - ключ\_регистрации
- 5) Сущность Вопросы включает следующие атрибуты:
  - код\_вопроса
  - вопрос
  - номер
  - описание

- код\_теста
- код\_типа\_вопроса
- правильный\_ответ
- статистика
- обязательный

б) Сущность Вопросы\_пользователя включает следующие атрибуты:

- код\_вопроса\_пользователя
- код\_записи
- код\_вопроса
- значение
- правильный\_ответ
- код\_типа\_вопроса
- код\_пользователя
- код\_типа\_пользователя

7) Сущность Группы включает следующие атрибуты:

- код\_группы
- наименование

8) Сущность Дневник включает следующие атрибуты:

- код\_дневника
- код\_группы
- дата\_публикации
- наименование
- информация

9) Сущность Лекция включает следующие атрибуты:

- код\_лекции
- код\_темы
- номер
- наименование

- 10) Сущность Тип\_пользователя включает следующие атрибуты:
  - код\_типа\_пользователя
  - наименование\_типа\_пользователя
- 11) Сущность Тип\_вопроса включает следующие атрибуты:
  - код\_типа\_вопроса
  - наименование
- 12) Сущность Тема включает следующие атрибуты:
  - код\_темы
  - наименование
- 13) Сущность Тест включает следующие атрибуты:
  - код\_теста
  - наименование
  - информация
  - доступна
  - ограничение\_по\_времени
  - код\_темы
- 14) Сущность Тесты\_пользователя включает следующие атрибуты:
  - код\_записи
  - код\_пользователя
  - код\_теста
  - дата\_и\_время
  - завершено
  - примечание
  - код\_типа\_пользователя
  - код\_темы

Концептуальная схема базы данных отображена на рисунке 2.5.

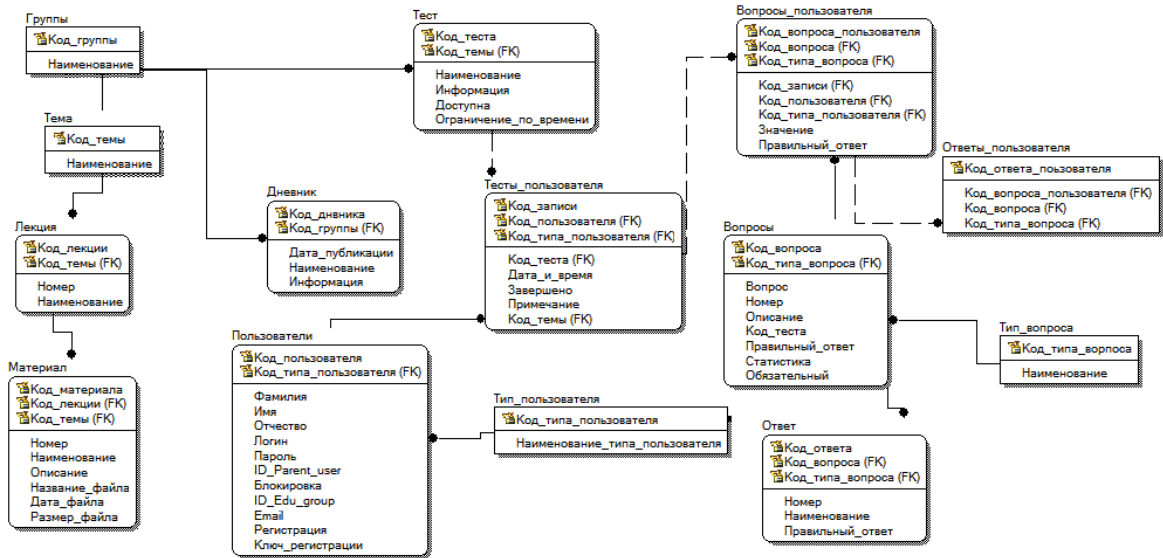


Рисунок 2.5 - Логическая модель базы данных

В соответствии с выбранной СУБД определим для каждой таблицы тип поля и формат содержащихся в нем данных.

Таблица 2.12

Структура таблицы answer

Поле	Тип
ID_Answer	bigint(20)
ID_Question	bigint(20)
Number	int(11)
Name	varchar(255)
Correct_answer	int(11)

Таблица 2.13

Структура таблицы edu\_group

Поле	Тип
ID_Edu_group	int(11)
Name	varchar(255)

Таблица 2.14

Структура таблицы edu\_group\_diary

Поле	Тип
ID_Edu_group_diary	int(11)



Поле	Тип
ID_Edu_group	int(11)
Publication_date	datetime
Name	varchar(255)
Info	longtext

Таблица 2.15

## Структура таблицы lecture

Поле	Тип
ID_Lecture	int(11)
ID_Subject	int(11)
Number	int(11)
Name	varchar(255)
ID_Questionary	int(11)

Таблица 2.16

## Структура таблицы material

Поле	Тип
ID_Material	int(11)
ID_Lecture	int(11)
Number	int(11)
Name	varchar(255)
Description	mediumtext
File_name	varchar(255)
File_data	mediumblob
File_size	int(11)

Таблица 2.17

## Структура таблицы question

Поле	Тип
ID_Question	bigint(20)
Name	varchar(255)
Number	int(11)
Description	longtext

Поле	Тип
ID_Questionary	int(11)
ID_Question_type	int(11)
IsRequired	int(11)
IsStatGroup	int(11)
Correct_answer	varchar(255)

Таблица 2.18

## Структура таблицы questionnaire

Поле	Тип
ID_Questionary	int(11)
Name	varchar(255)
IsAccessible	int(11)
TimeLimit	int(11)

Таблица 2.19

## Структура таблицы question\_type

Поле	Тип
ID_Question_type	int(11)
Name	varchar(255)

Таблица 2.20

## Структура таблицы subject

Поле	Тип
ID_Subject	int(11)
Name	varchar(255)

Таблица 2.21

## Структура таблицы user

Поле	Тип
ID_User	int(11)
Surname	varchar(255)
Name	varchar(255)
Patronymic	varchar(255)
ID_User_type	int(11)

Поле	Тип
Login	varchar(255)
Password	varchar(255)
ID_Parent_user	int(11)
Block	int(11)
ID_Edu_group	int(11)
Email	varchar(255)
Registred	varchar(255)
RegisterKey	varchar(255)

Таблица 2.22

## Структура таблицы user\_answer

Поле	Тип
ID_Question_result	bigint(20)
ID_Answer	bigint(20)

Таблица 2.23

## Структура таблицы user\_question

Поле	Тип
ID_Question_result	bigint(20)
ID_User_questionary	bigint(20)
ID_Question	bigint(20)
Value	varchar(255)
Correct_answer	int(11)

Таблица 2.24

## Структура таблицы user\_questionary

Поле	Тип
ID_User_questionary	bigint(20)
ID_User	int(11)
ID_Questionary	int(11)
CurrentDateTime	datetime
Completed	int(11)
Mark	double

Структура таблицы user\_type

Поле	Тип
ID_User_type	int(11)
Name	varchar(255)

Физическая модель базы данных представлена на рисунке 2.6

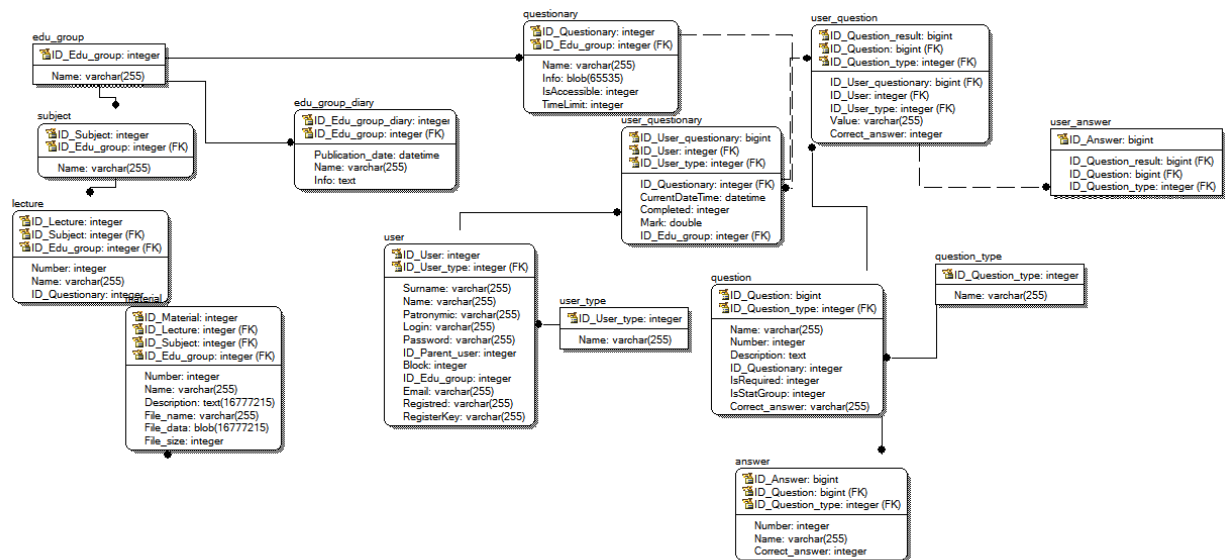


Рисунок 2.6 Физическая модель разработанной базы данных

### 2.3 Описание реализации основных функций системы

При разработке информационной системы ее функциональные возможности, то есть поведение, которым она должна обладать с точки зрения заказчика, документируются с помощью модели прецедентов. Модель прецедентов показывает, какие функции должна быть способна выполнять система, а также в какой среде она должна работать. Модель прецедентов разрабатывается на ранних этапах проектирования и дает возможность заказчику четко сформулировать свои требования, а инженеру-разработчику – понять, что нужно заказчику.

Основными элементами модели прецедентов являются актеры и прецеденты.

Субъекты - может только вводить информацию в систему, только получать информацию из системы или делать и то, и другое. В роли субъекта может выступать также и другая система, если она будет взаимодействовать с разрабатываемой системой, и не будет являться ее неотъемлемой частью.

Прецеденты - предназначены для моделирования диалога между системой и ее субъектами. Они представляют собой возможности, которые система может обеспечить конкретному субъекту.

Работа системы осуществляется следующим образом:

1. Администратор системы регистрирует ее пользователей и заполняет справочники.

2. Пользователь в роли преподаватель заполняет систему учебными материалами и тестовыми данными, а также получает отчеты по результатам тестирования обучаемых

3. Пользователь в роли Обучаемый изучает учебные материалы и проходит тестирование.

На рис. 2.7 представлена диаграмма прецедентов, которая представляет функции администратора по работе с системой.

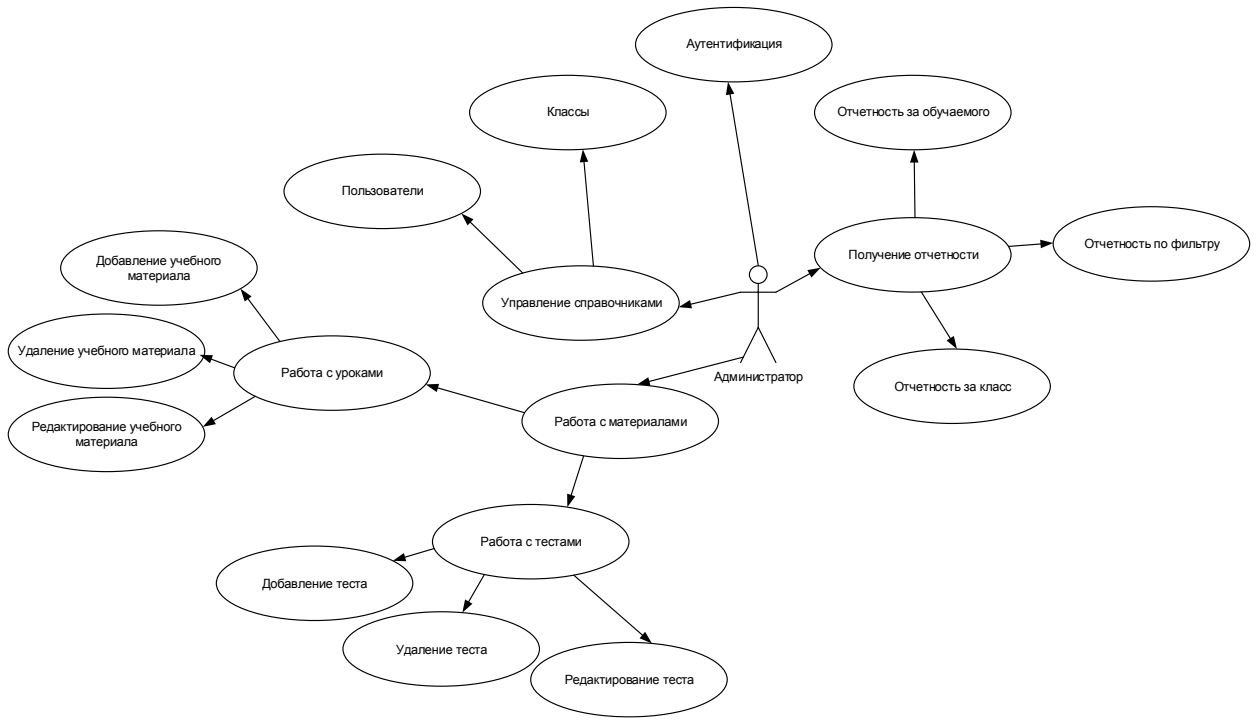


Рисунок 2.7 Диаграмма прецедентов администратора

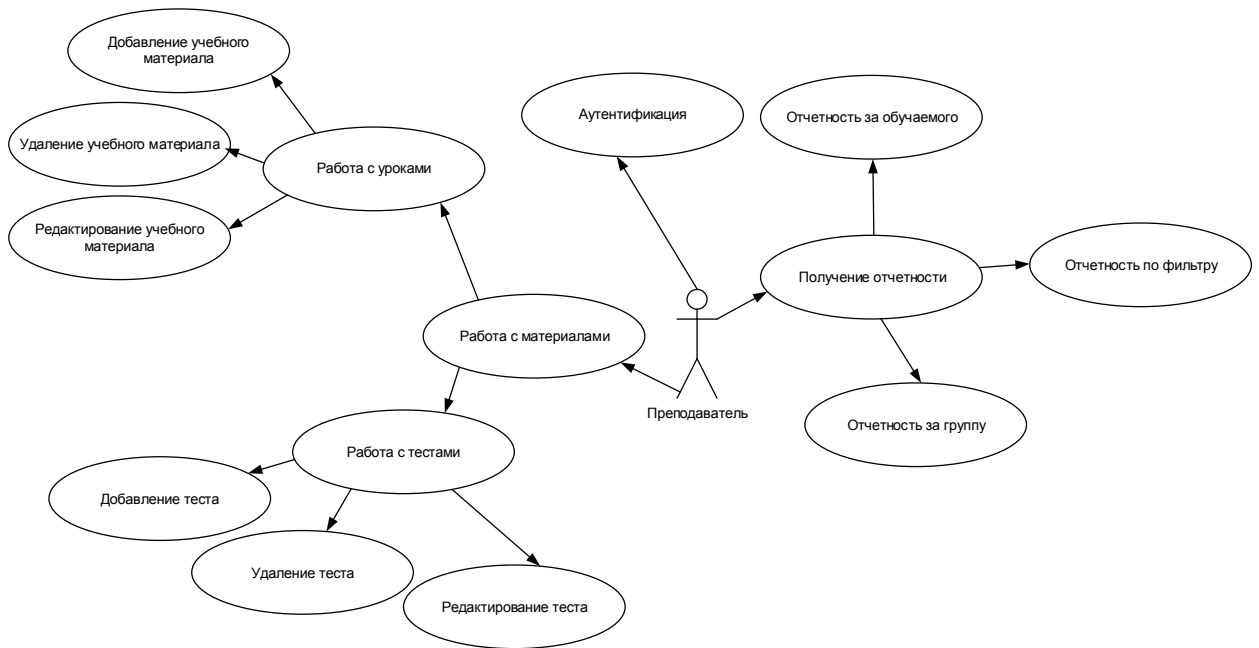


Рисунок 2.8 Диаграмма прецедентов преподавателя

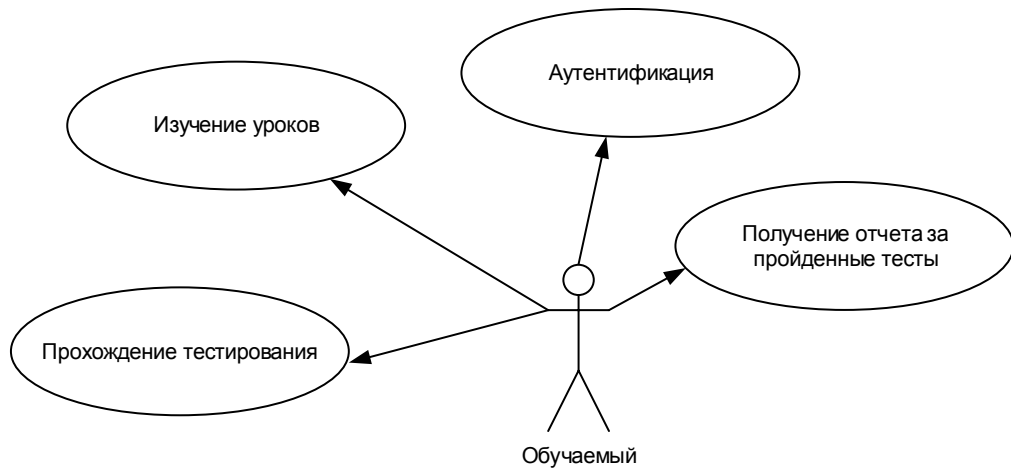


Рисунок 2.9 Диаграмма прецедентов обучаемого

Прецедент «Аутентификация пользователя» активизируется всеми субъектами ИС. Прежде чем пользователю начать работу с БД, система запрашивает его пароль и логин. Если пользователь не зарегистрирован или в пароле и/или логине допустил ошибку, то он не получает доступа к работе в ИС. После успешной проверки логина и пароля пользователя открывается главное окно программы. На рисунках представлены диаграмма последовательности и диаграмма действий этого прецедента.



Рисунок 2.10 Диаграмма последовательности «Аутентификация пользователя»

Прецедент «Учет урока» активизируется субъектом ИС Преподаватель. Данный прецедент описывает процесс ввода новых данных в БД. Этим занимается пользователь с ролью заказчик. Данные вводятся в формы ввода на основании документов, а также вся справочная информация, которая нужна в процессе работы с системой. На рисунках представлены диаграмма последовательности и диаграмма действий данного прецедента.



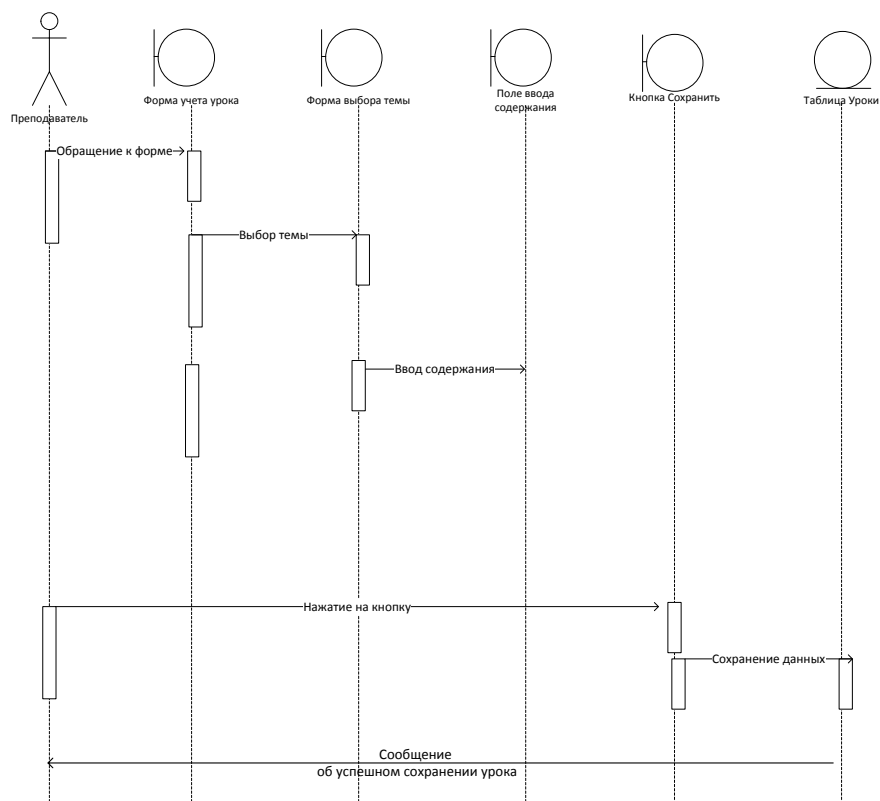


Рисунок 2.11 Диаграмма последовательности «Учет уроков»

Диаграмма последовательности «Прохождение теста» показывает порядок тестирования студентами. Диаграмма последовательности при прохождении тестирования приведена на рисунке 2.12.

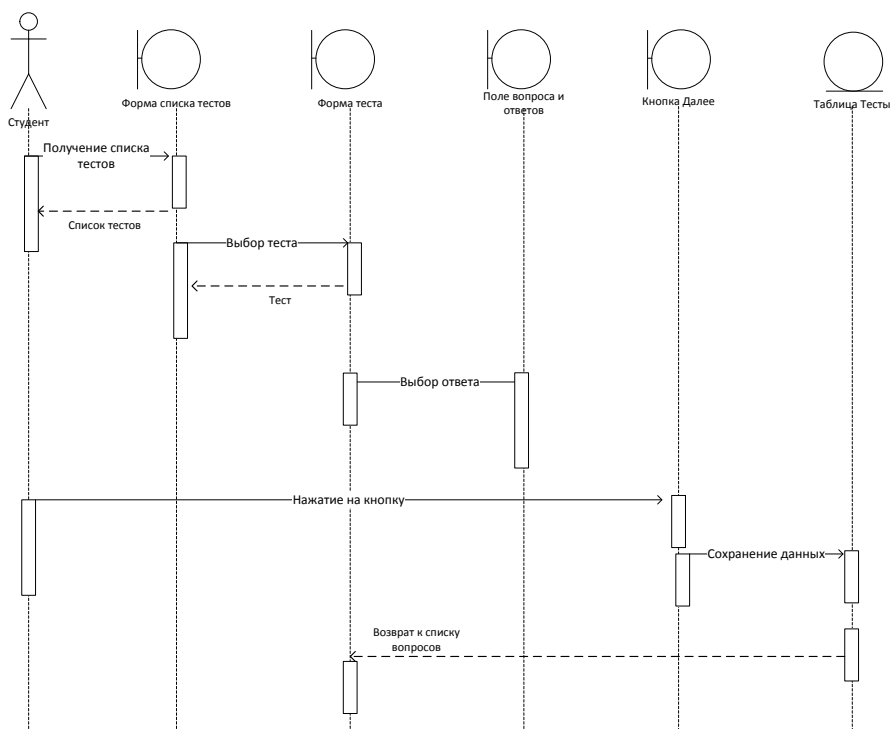


Рисунок 2.12 Диаграмма последовательности при тестировании

По окончании тестирования обучаемый получает его результаты.

Диаграмма компонентов показывает, из каких частей состоит проектируемая система и как эти части связаны друг с другом.

Компонент – совокупность описаний классов и других программных ресурсов, в том числе и самих пакетов.

Анализ концептуальной модели позволил выделить следующие компоненты:

- интерфейсные элементы - классы, реализующие интерфейсные компоненты;
- пользовательский интерфейс - классы, реализующие объекты интерфейса с пользователем;
- интерфейс с БД - классы, реализующие интерфейс с базой данных;
- база данных

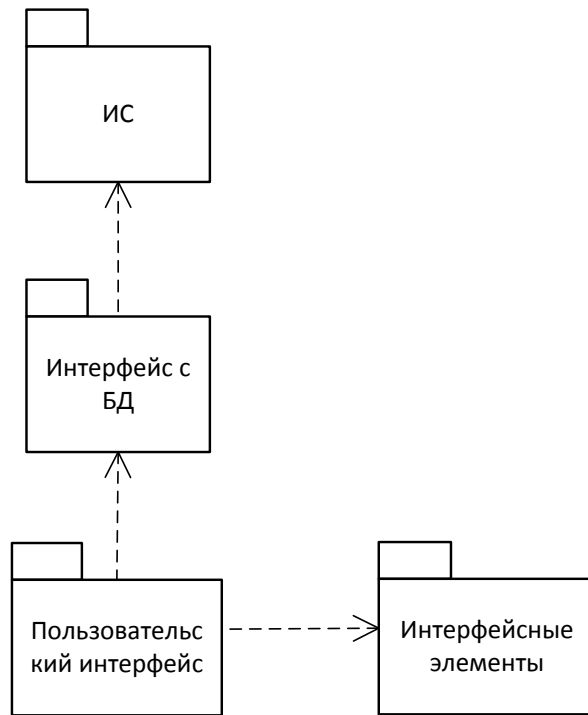


Рисунок 2.13 Диаграмма компонентов

Диаграмма развертывания системы представлена на рисунке 2.14.

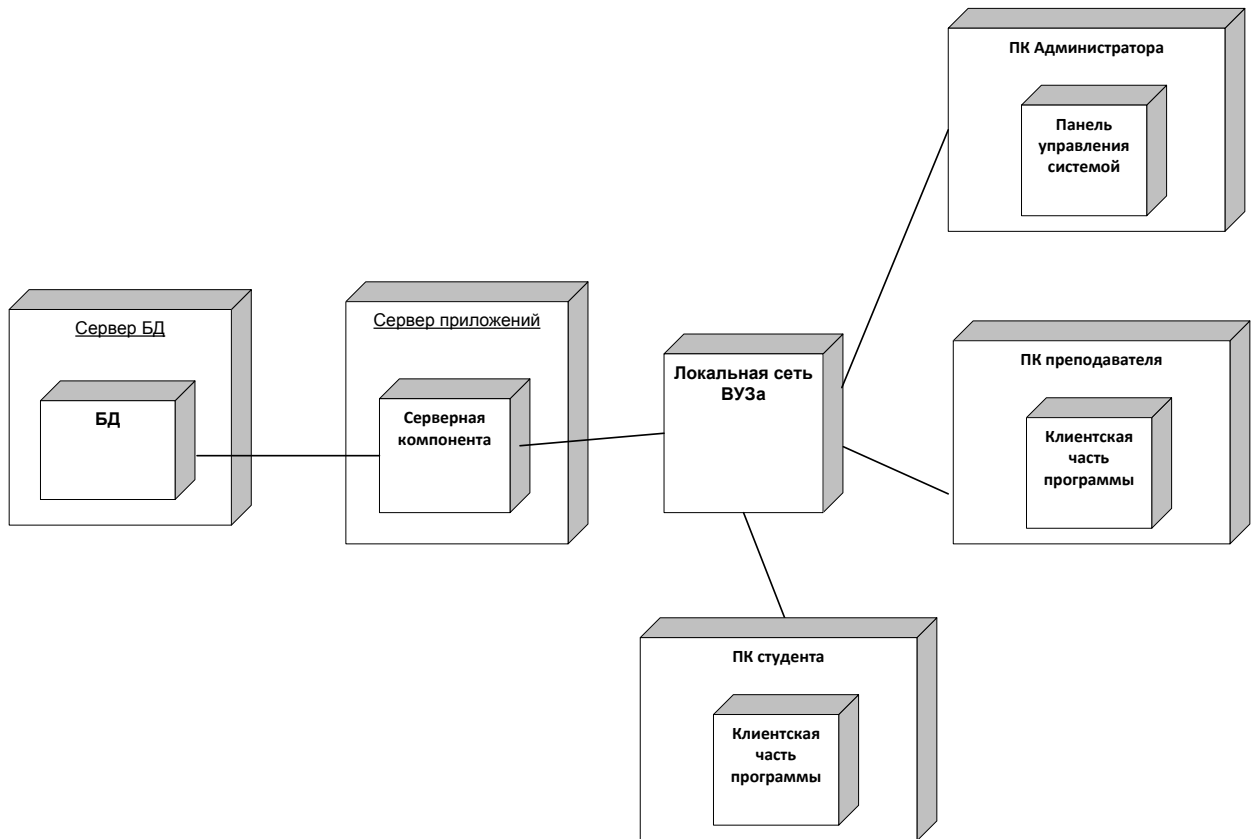


Рисунок 2.14 Схема развертывания

Как следует из приведенной диаграммы, для развертывания системы необходим сервер баз данных, сервер приложений, а также персональные

компьютеры пользователей системы с установленными на них клиентскими компонентами программы.

В соответствии с выбранной структурой приложения программное обеспечение включает следующие компоненты:

- операционная система на компьютере пользователя системы;
- операционная система на сервере;
- сервер баз данных;
- средство управления базой данных;
- средство разработки программного продукта.

Основная функция, разрабатываемой информационной системы, сложна и требует иерархического представления. Для более наглядного представления функции системы необходимо декомпозировать на составляющие его функции.

Представим основную функцию системы в дерева функций.

На рисунке 2.15. представлено дерево функций системы.



Рисунок 2.15 Дерево функций системы

Первый уровень представляет основную функцию разрабатываемой системы – учёт информации об обучаемых, занятиях, отчетности. Эту информацию можно условно разделить на две составляющие:

- данные об обучаемых;
- данные об отчетности.

Подфункция учета отчетности разделяется в соответствии с наименованиями типов отчетности, типов занятий и предметов . Подфункция работы со справочниками системы предполагает разделение на ввод данных, получение содержания справочника, редактирование и удаление записей в справочнике.

Служебные функции содержат функции настройки доступа к ИС, а также авторизационных данных.

Список входных данных для разработанной системы приведен в таблице 2.26.

Таблица 2.26

## Входные данные

Наименование	Тип данных	Допустимые значения	Источник
Класс	Общий	Алфавитно-цифровые данные (текст и числа)	Документ
Предмет	Текстовый	Алфавитные данные	Документ
Тема	Текстовый	Основная, дополнительная	Документ
ФИО обучаемого	Текстовый	Алфавитные данные	Документ
ФИО пользователя	Текстовый	Алфавитные данные	Документ
Класс	Текстовый	Алфавитно-цифровые и символьные данные (XX-1...XX-12)	Документ

Результатной информация в этом случае являются следующие документы (таблица 2.27).

## Перечень результатной информации

Наименование	Тип данных	Допустимые значения	Источник
ФИО обучаемого	Текстовый	Алфавитные данные	монитор
Список обучаемых	Текстовый	Алфавитные данные	Монитор, печатный документ
Список тем	Текстовый	Алфавитные данные	Монитор, печатный документ
Список тестов	Текстовый	Алфавитные данные	Монитор, печатный документ
Результаты тестов	Текстовый	Алфавитные данные	Монитор, печатный документ
Список пользователей	Текстовый	Алфавитные данные	Монитор, печатный документ

Таким образом, во второй главе работы выбрана архитектура разрабатываемой системы, описана модель данных, а также основные функции системы.

## **Выводы по Главе 2**

Во втором разделе работы произведено проектирование системы оценки качества обучения. Для этого выбрана двухуровневая архитектура «тонкий клиент»-«толстый сервер». Произведено сравнение языков программирования и систем управления базами данных, изучены их плюсы и минусы и составлены матрицы попарных сравнений.

В главе представлена физическая и логическая модель базы данных. Описана реализация основных функций системы, составлены диаграммы прецедентов для администратора, преподавателя и обучаемого. Составлено дерево функций системы и таблица входных данных.

### **3 Глава 3. Документирование готового программного продукта**

#### **3.1 Испытания системы оценки качества обучения**

Тестирование является важной и обязательной частью процесса разработки. Причем, в разработке программы необходимо не только финальное, обязательное тестирование, но и периодичное.

Процесс тестирования можно разделить на следующие этапы:

- проверка в нормальных условиях (правильность работы программы для характерных данных);
- проверка в исключительных ситуациях.
- тестирование нулевыми данными;
- тестирование чужими данными;
- тестирование избыточными данными.

Прикладное программное обеспечение будет тестироваться на локальной машине с использованием сервера MS SQL. Будет применяться комплексное тестирование имитирующее работу пользователя и администратора с системой. В ходе тестирования будут выполнены следующие действия: 1) регистрация нового пользователя, 2) подтверждение регистрации администратором, 3) задание прав модератора пользователю, 4) обработка заявки, 5) добавление новой заявки.

Тестирование в нормальных условиях предполагает тестирование на основе данных и условий, которые характерны для реального функционирования программы. В данном случае нормальные условия это: наличие неповрежденной базы данных, правильное указание параметров программы, ввод корректных данных.

При проверке в нормальных условиях все функции программы работают правильно, и выдается достоверная информация в соответствии с результатами контрольных расчетов.

Тестирование нулевыми данными



В разработанной системе обязательными являются все поля, содержание которых критично для вводимой информации.

Программа адекватно реагирует на нулевые и пустые данные и не позволяет закрыть окна, пока все соответствующие поля не будут заполнены.

#### Тестирование чужими данными

Тестирование чужими данными проводилось в некоторых экранных формах и при авторизации пользователя.

Исключительные данные представляют собой ввод данных в избыточном объеме. Программа определяет такие ситуации, выводит сообщение, что в данной ячейке вводится ограниченное количество знаков. Например, при добавлении телефона пользователя с количеством цифр более 11 программа выдаст ошибку.

#### Оценка полноты проверки программы

В ходе тестирования система эксплуатировалась в условиях, близких к реальным.

Система не позволяет пользователю вводить большинство недопустимых значений, что существенно упрощает процесс тестирования, минимизирует риск возникновения исключительных ситуаций и позволяет достаточно полно проверить работоспособность программного обеспечения в нормальных и экстремальных условиях. Вывод сформированных отчетных форм на печать работает без сбоев.

Оценку качества модуля (экспертизу) проводят на фазах жизненного цикла в соответствии с ГОСТ 28195-89 «Оценка качества программных средств». Она включает выбор номенклатуры показателей, их оценку и сопоставление значений показателей, полученных в результате сравнения с базовыми значениями.

Данное ПП относится к подклассу 599 — прочие ПС.

Формируем оценочную таблицу (приложение 1).

В процессе оценки качества ПС на каждом уровне (кроме уровня оценочных элементов) проводятся вычисления показателей качества ПС, т. е. определение количественных значений абсолютных показателей ( $P_{ij}$ , где  $P$  – порядковый номер показателя данного уровня для  $i$ -го показателя вышестоящего уровня) и относительных показателей ( $K_{ij}$ ), являющихся функцией показателя  $P_{ij}$  и базового значения  $P_{ij}^{\text{баз}}$ .

Каждый показатель качества 2-го и 3-го уровней (критерий и метрика) характеризуется двумя числовыми параметрами – количественным значением и весовыми коэффициентами ( $V_{ij}$ ).

Сумма весовых коэффициентов показателей уровня ( $i$ ) относящихся к  $i$ -му показателю вышестоящего уровня ( $i - 1$ ), есть величина постоянная. Сумма весовых коэффициентов ( $V_{ij}$ ) принимается равной 1.

$$\sum_{j=1}^n V_j = \text{const} = 1,$$

где  $j = 1 \text{--} n$ ,

$n$  – число показателей уровня ( $i$ ) относящихся к  $i$ -му показателю вышестоящего уровня ( $i-1$ ).

Общая оценка качества ПС в целом формируется экспертами по набору полученных значений оценок факторов качества.

Для оценки качества ПС различного назначения методом экспертного опроса составляется таблица значений базовых показателей качества ПС.

Определение усредненной оценки ( $m_{kq}$ ) оценочного элемента по нескольким его значениям ( $m_s$ ) проводится по формуле

$$m_{kq} = \frac{\sum_{s=1}^t m_s}{t}; \text{ где}$$

$t$  – число значений ОЭ (оценочного элемента);

$k$  – порядковый номер метрики;

$q$  – порядковый номер ОЭ.

Итоговая оценка  $k$ -й метрики  $j$ -го критерия ведется по формуле

$$P_{jk}^m = \frac{\sum_{i=1}^Q m_{kj}}{Q}; \quad (3)$$

где Q — число ОЭ в k-й метрике.

Абсолютные показатели критериев i-го фактора качества определяется по формуле

$$P_i = \sum_{k=1}^n (P_{jk}^M * V_{ji}^M) \quad (4)$$

где n — число метрик, относящихся к j-му критерию.

Относительный показатель /-го критерия i-го фактора качества вычисляется по формуле

$$K_{ij} = \frac{P_{ij}}{P_{ij}^{баз}} \quad (5)$$

Если значения базовых ПС принять идеальными, то есть равными единице, то получится следующее:

$$K_{ij} = P_{ij}^{баз} \quad (6)$$

Фактор качества ( $R_i^\phi$ ) вычисляется по формуле

$$R_i^\phi = \sum_{i=1}^N K_i * V_{ij}^k \quad (7)$$

где N — число критериев качества, относящихся к i-му фактору.

Итогом является таблица показателей качества (таблица 3.1).

Таблица 3.1

Показатели качества.

Фактор	Значение показателей
Надежность	0,90
Сопровождаемость	0,76
Удобство применения	0,92
Эффективность	0,85
Универсальность	0,79
Корректность	0,78

Показатели качества ПП в целом удовлетворяют предъявляемым требованиям.

Анализ результатов экспериментальных исследований, проведенной в трех точках, дает положительный результат работоспособности программы. Однако некоторые недоработки может определить только эксплуатация, но в рамках этих условий, руководствуясь результатами тестирования можно сказать об удовлетворительной работе программного продукта.

### 3.2 Руководство пользователя системы оценки качества обучения

Для начала работы системы необходимо разместить серверную составляющую на сервере, а клиентские части – на компьютерах пользователей. После запуска сервера и клиента каждый клиент устанавливает соединение с сервером в соответствии с настройками соединения (рисунок 3.1):

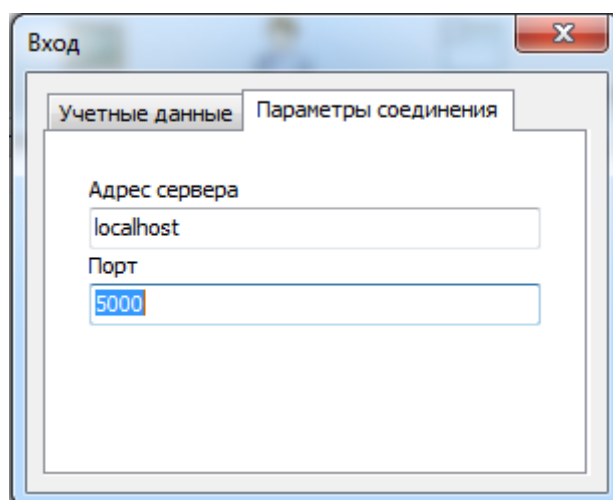


Рисунок 3.1 Настройки соединения с сервером

После установки соединения необходимо ввести пароль и логин для входа в программу (рисунок 3.2).

Рисунок 3.2 Ввод пароля и логина

Если пароль и логин введены верно, то пользователю откроется главная форма программы (рисунок 3.3).

Тип пользователя	Фамилия	Имя	Отчество	online
▶ Преподаватель	Сергеев	Алексей	Сергеевич	1

Рисунок 3.3 Главная форма программы для администратора

На главной форме расположены иконки доступа к пунктам меню программы. При нажатии иконки Уроки будут показаны имеющиеся в системе уроки (рисунок 3.4).

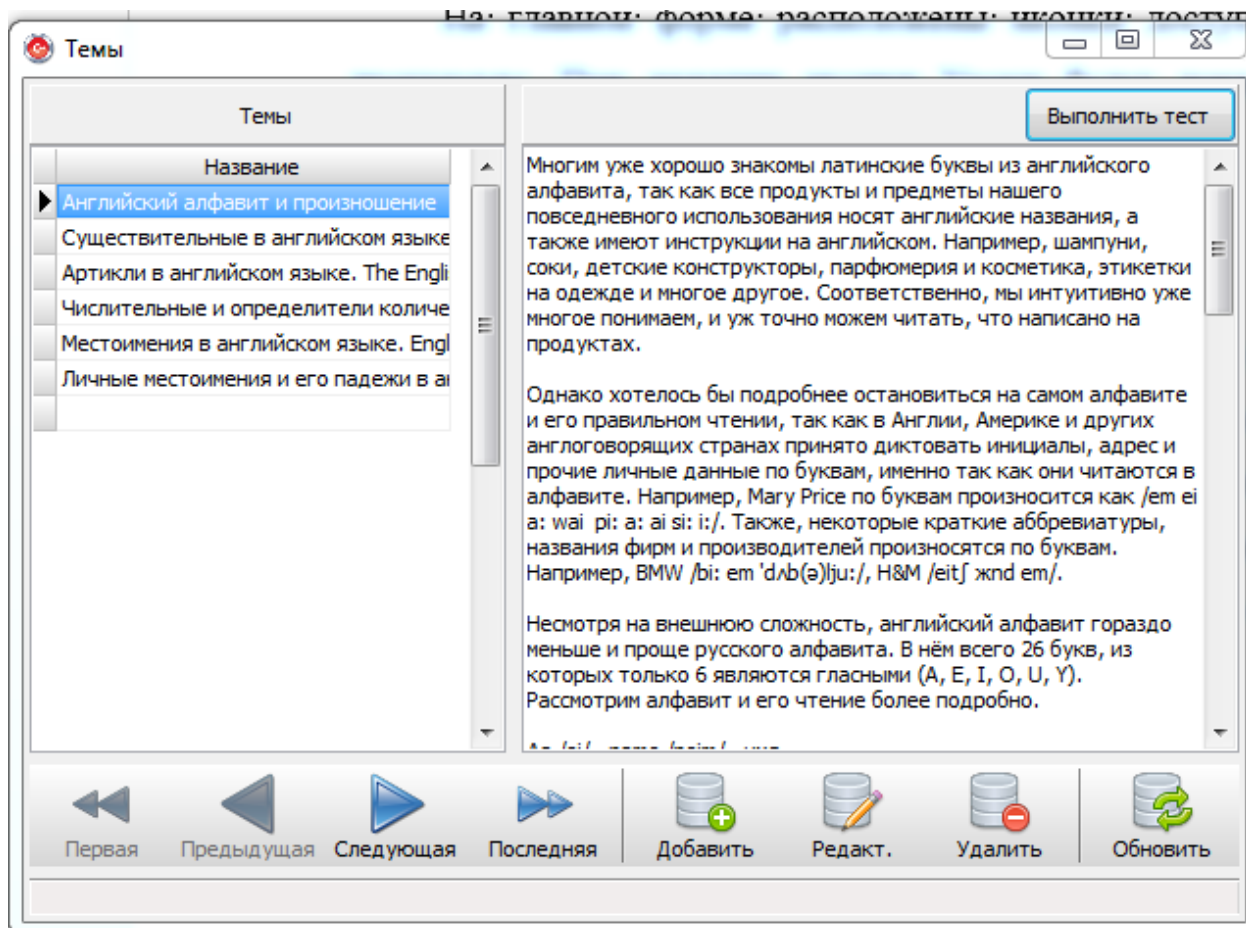


Рисунок 3.4 Уроки в программе

Данное окно состоит из трех частей – в левой части кона показан список уроков, в правой – содержание урока. В нижней части окна расположены стандартные кнопки интерфейса – перемещение по записям, а также кнопки добавления, редактирования, удаления и обновления информации. При нажатии на кнопку «Выполнить тест» открывается окно с тестом (рисунок 3.5).

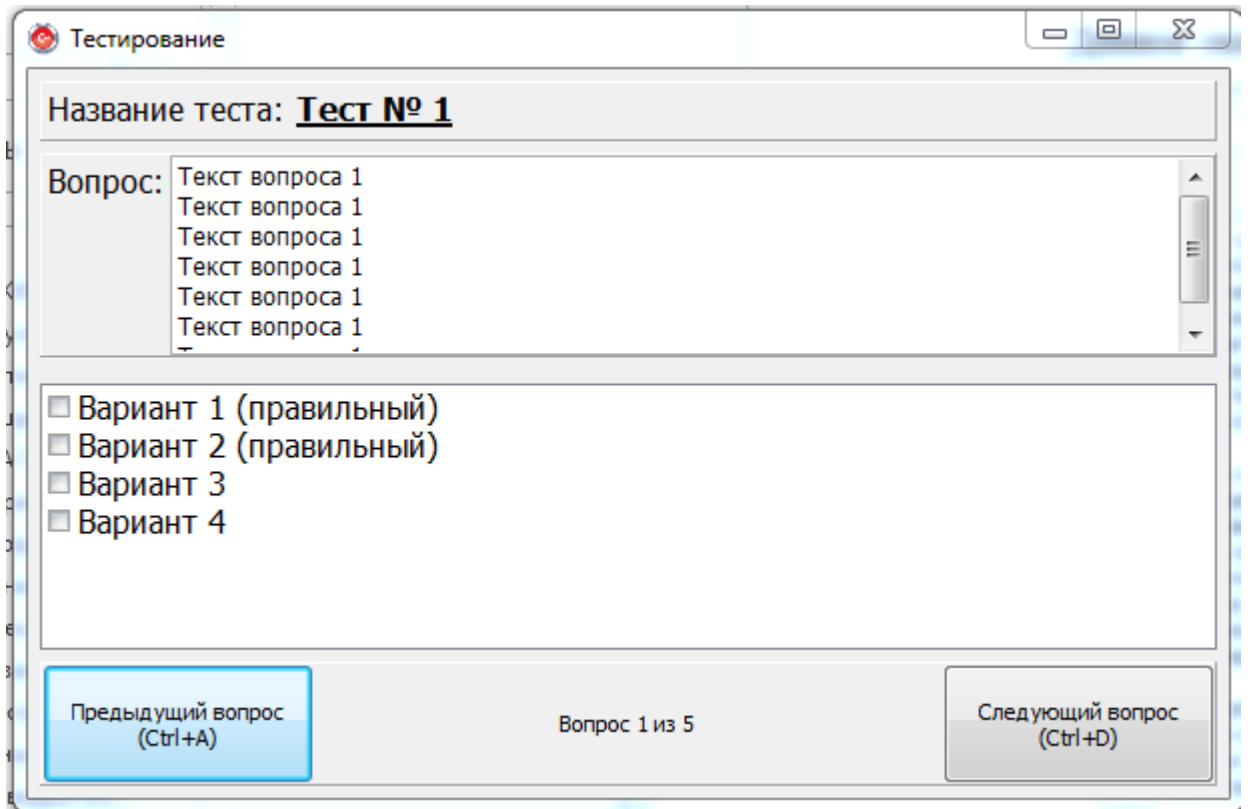
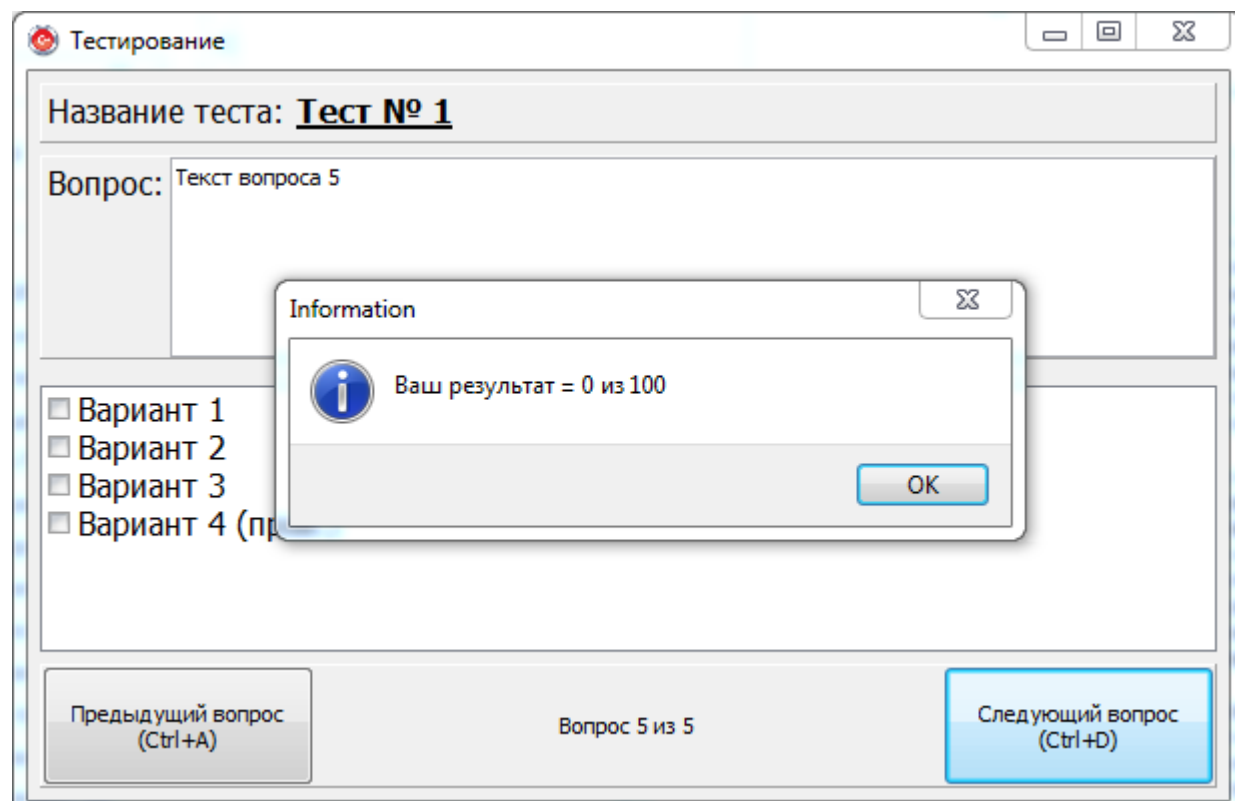


Рисунок 3.5 Окно прохождения теста

Тест проходит путем выбора верного варианта и перехода к следующему вопросу. При окончании теста показывается результат его прохождения (рисунок 3.6).



### Рисунок 3.6 Результат прохождения теста

Администратор имеет возможность просмотра результатов теста и вывода их на печать (рисунок 3.7).

Код	Дата	Тест	Группа	Фамилия	Имя	Отчество	Логин	online	Балл
57	07.05.2017 3:33:32	Тест № 1	Группа 2	Петров	Алексей	Петрович	user 1	0	
58	07.05.2017 3:35:40	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
59	07.05.2017 17:52:05	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
60	07.05.2017 11:38:43	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
61	07.05.2017 11:38:51	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
62	07.05.2017 10:46:48	Тест № 2	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
63	07.05.2017 10:51:00	Тест № 3	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
64	07.05.2017 12:57:28	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
65	07.05.2017 12:57:36	Тест № 2	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
66	07.05.2017 12:57:41	Тест № 3	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
67	07.05.2017 13:03:00	Тест № 3	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	
68	07.05.2017 17:31:21	Тест № 1	Группа 1	Сергеев	Алексей	Сергеевич	admin	1	

### Рисунок 3.7 Результаты тестов

Имеется возможность выбора как всех результатов, так и результата выбранного теста (рисунок 3.8).

Результаты тестирования					
№	Дата	ФИО	Группа	Название теста	Балл
1	07.05.2017 3:33:32	Петров Алексей Петрович	Группа 2	Тест № 1	80
2	07.05.2017 3:35:40	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	40
3	07.05.2017 17:52:05	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	0
4	07.05.2017 11:38:43	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	0
5	07.05.2017 11:38:51	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	100
6	07.05.2017 10:46:48	Сергеев Алексей Сергеевич	Группа 1	Тест № 2	0
7	07.05.2017 10:51:00	Сергеев Алексей Сергеевич	Группа 1	Тест № 3	0
8	07.05.2017 12:57:28	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	0
9	07.05.2017 12:57:36	Сергеев Алексей Сергеевич	Группа 1	Тест № 2	0
10	07.05.2017 12:57:41	Сергеев Алексей Сергеевич	Группа 1	Тест № 3	50
11	07.05.2017 13:03:00	Сергеев Алексей Сергеевич	Группа 1	Тест № 3	50
12	07.05.2017 17:31:21	Сергеев Алексей Сергеевич	Группа 1	Тест № 1	0

### Рисунок 3.8 Результаты тестирования для печати

Имеется возможность выбора как всех результатов, так и результата выбранного теста (рисунок 3.9).



### Результаты тестирования

Дата	07.05.2017 3:33:32
ФИО	Петров Алексей Петрович
Группа	Группа 2
Название теста	Тест № 1
Балл	80

Номер вопроса	Название вопроса	Вопросы	Ответы	Прав. ответ
1	Вопрос 1	Текст вопроса 1 Текст вопроса 1 Текст вопроса 1 Текст вопроса 1 Текст вопроса 1 Текст вопроса 1 Текст вопроса 1	Вариант 1 (правильный) (+) Вариант 2 (правильный) (+)	+
2	Вопрос 2	Текст вопроса 2	Вариант 1 (правильный) (+)	+
3	Вопрос 3	Текст вопроса 3	Вариант 1 (правильный) (+)	+
4	Вопрос 4	Текст вопроса 4	Вариант 1 (прав.) (+) Вариант 2 (-) Вариант 3 (-) Вариант 4 (-) Вариант 5 (прав.) (+) Вариант 6 (прав.) (+) Вариант 7 (прав.) (+)	
5	Вопрос 5	Текст вопроса 5	Вариант 4 (прав.) (+)	+

Рисунок 3.9 Результаты выбранного теста

Имеется также возможность выбора результатов теста с в соответствии с определёнными условиями фильтра (рисунок 3.10).

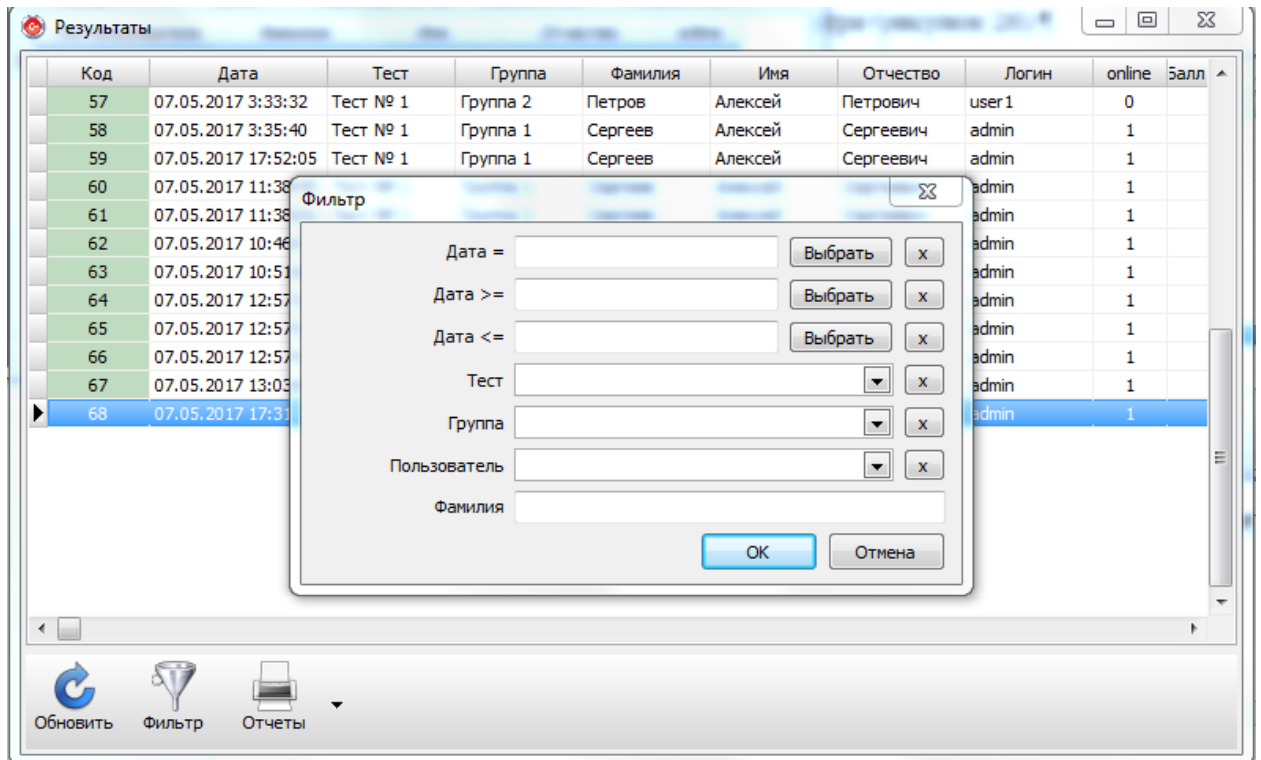


Рисунок 3.10 Фильтр для выбора получения отчета

Для добавления уроков предназначен редактор уроков (рисунок 3.11). При его использовании вначале добавляется тема урока, а затем его содержание.

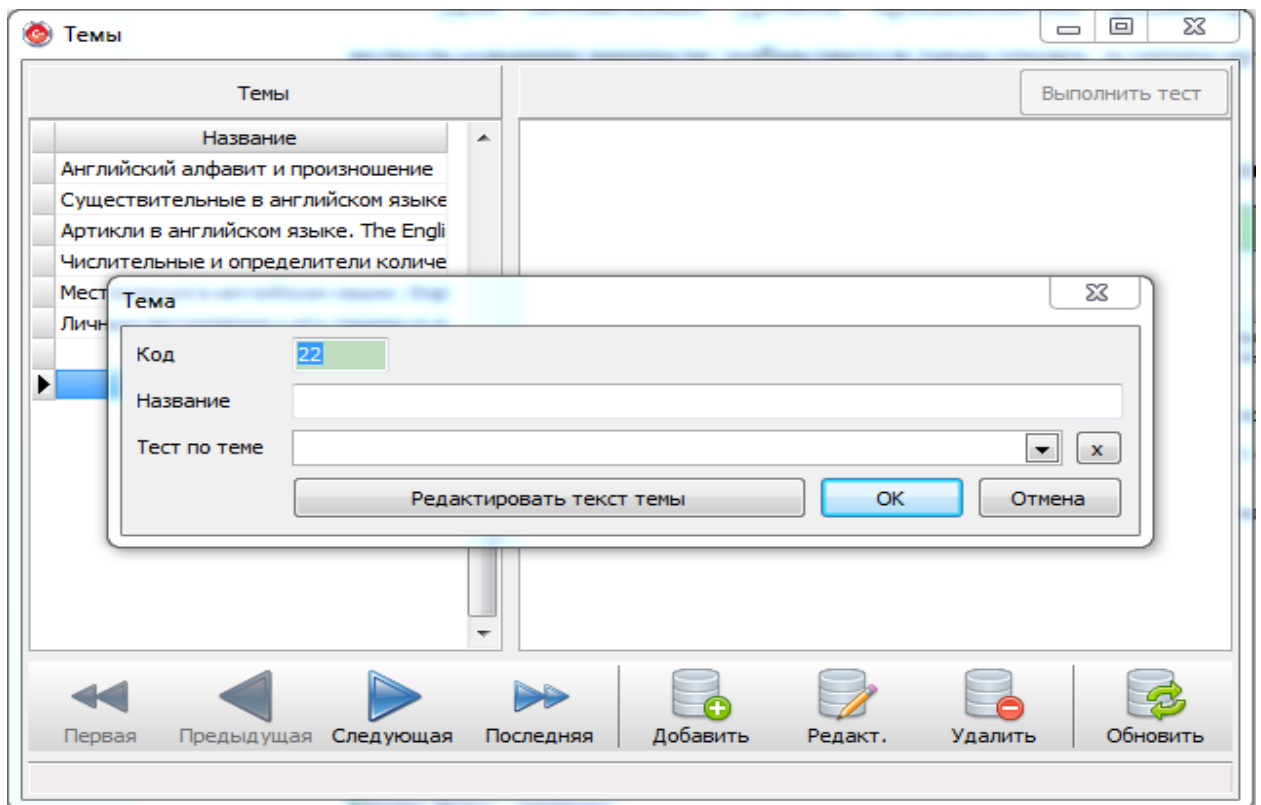


Рисунок 3.11 Добавление темы урока

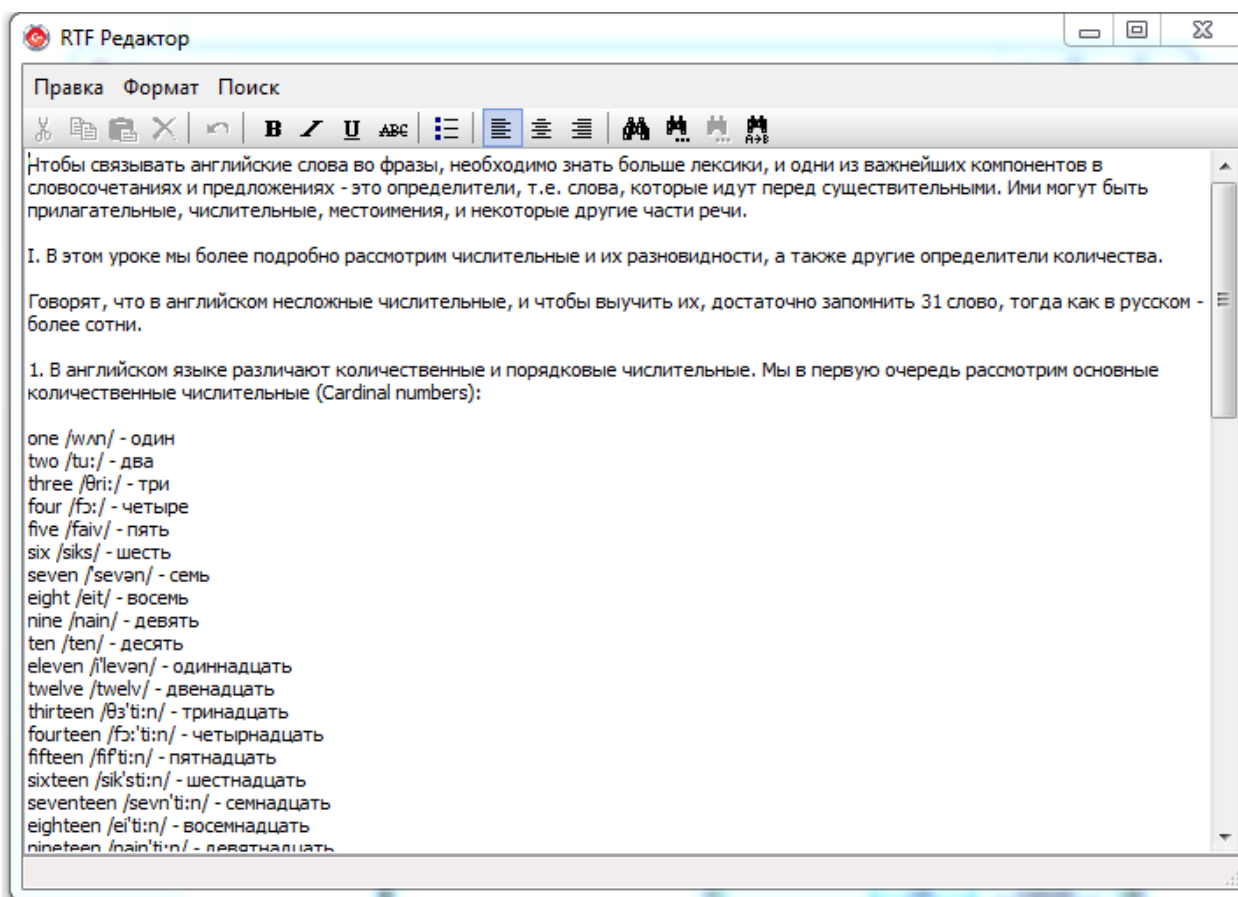


Рисунок 3.12 Добавление содержания урока

Для добавления тестов используется редактор тестов. При добавлении теста вначале необходимо ввести его название (рисунок 3.12).

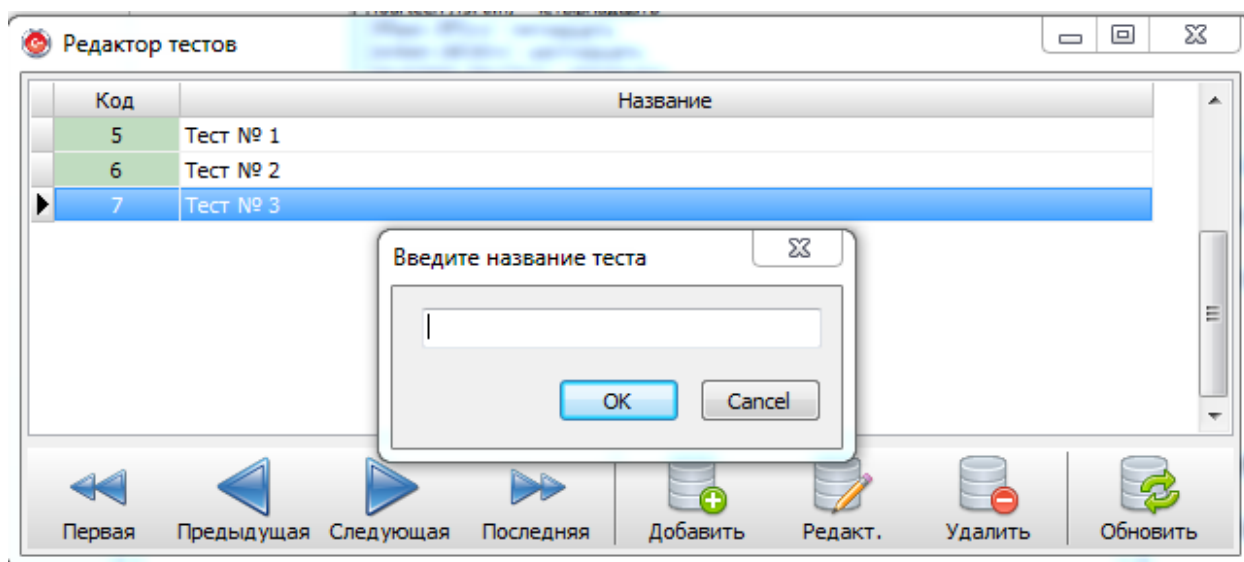


Рисунок 3.13 Добавление теста

После добавления названия последовательно добавляются вопросы теста и ответы к ним (рисунок 3.13).

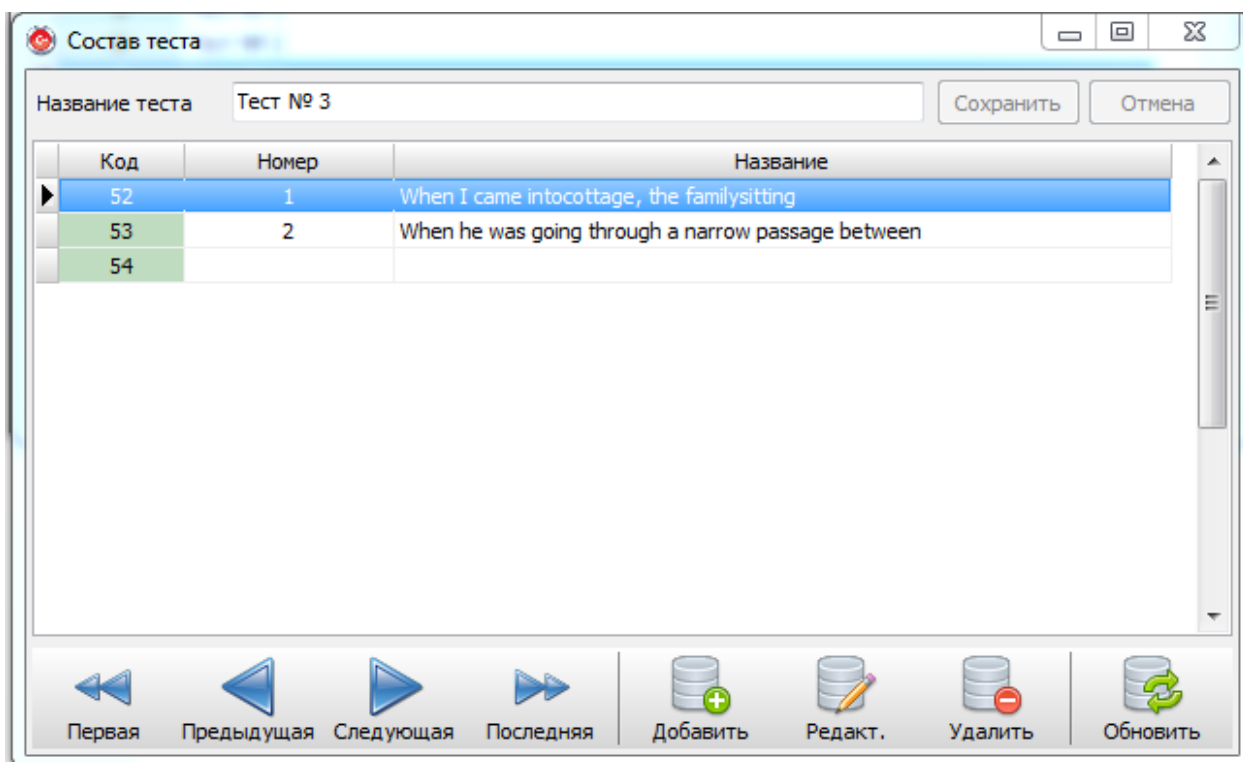


Рисунок 3.14 Список вопросов теста

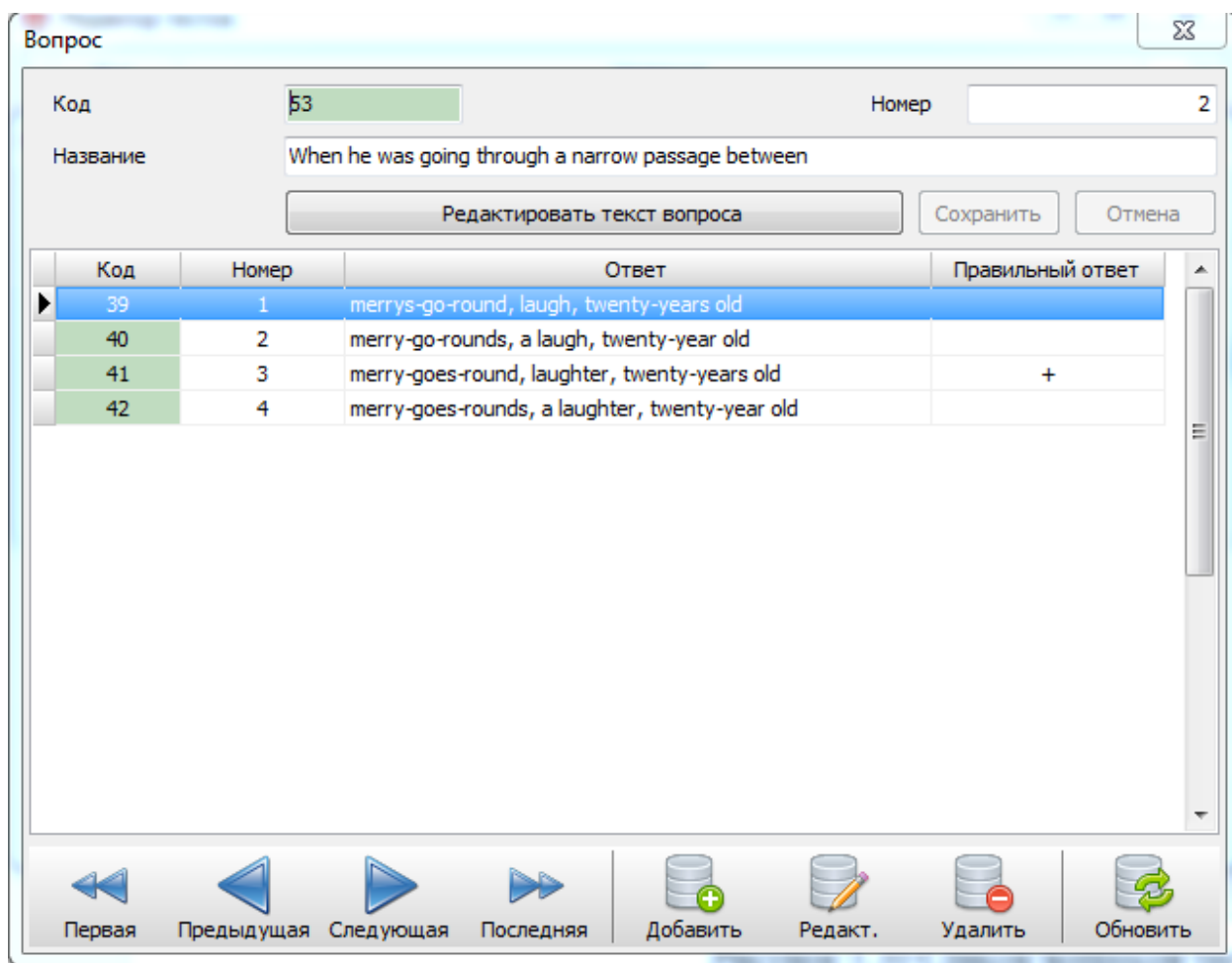


Рисунок 3.15 Добавление вопросов теста

Администратор также управляет пользователями, добавляя, удаляя их и редактируя их данные (рисунок 3.14).

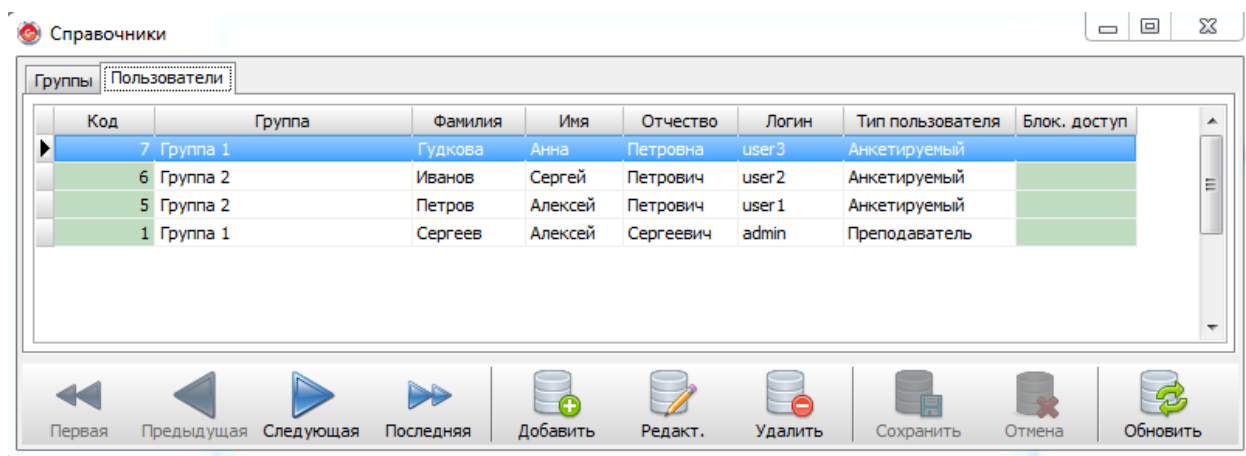


Рисунок 3.16 Добавление пользователей

### 3.3 Руководство системного программиста

Программная система состоит из следующих уровней: клиент; сервер приложений; сервер базы данных.

Клиент — это интерфейсный компонент, который представляет первый уровень, собственно приложение для конечного пользователя. Первый уровень не имеет прямых связей с базой данных и бизнес-логики.

Сервер приложений располагается на втором уровне. На втором уровне сосредоточена большая часть бизнес-логики.

Сервер базы данных обеспечивает хранение данных и выносится на третий уровень.

Сервер приложений разработан при помощи технологии Delphi XE2 DataSnap. Передача данных между клиентом и сервером осуществляется через протокол TCP.

Сервер приложений взаимодействует с базой данных через СУБД Microsoft Access. Подключение к базе данных выполняется через технологию ADO.

В структуре сервера можно выделить две основные части: Модуль управления сервером (TdmServer) и модуль предоставления данных (TdssmRemoteData). Описание модулей приведено ниже, в таблице 3.2.

Таблица 3.2

### Структура сервера приложений

№	Название модуля	Описание	Функции
1	TdmServer	Содержит компоненты для подключения к системе управления базами данных (через ADO) и компоненты для организации сервера приложений (передача данных выполняется через протокол TCP).	Подключение к СУБД. Управление сервером приложений (установка соединений с клиентскими приложениями, аутентификация пользователей, передача данных клиентам).
2	TdssmRemoteData	Модуль системы, который определяет доступные клиенту данные и функциональность системы. Экземпляр данного модуля создается для каждого подключенного клиента.	Авторизация. Предоставление данных; Предоставление функциональности;

При инициализации модуля управления сервером автоматически выполняется инициализация соединения с базой данных. Параметры соединения содержатся в файле Settings.ini. Если соединение с базой данных по каким-то причинам не удалось установить, то сервер выдаст сообщение о невозможности дальнейшей работы, т.е. успешное подключение к базе данных обязательное условия запуска сервера.

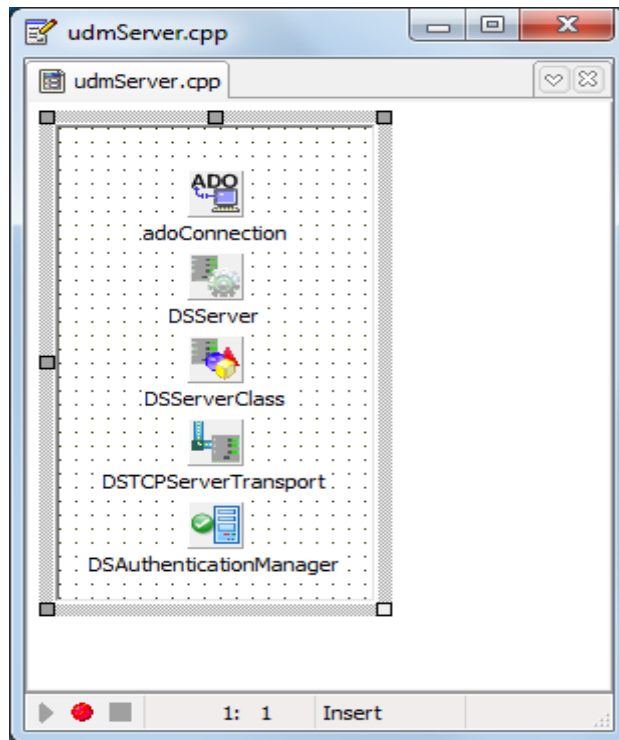


Рисунок 3.17 Состав модуля TdmServer

Как уже отмечалось ранее, сервер передает данные по протоколу TCP (за это отвечает компонент DSTCPServerTransport). Потому, для сервера должен быть задан порт прослушивания. В нашем случае он имеет значение 5000.

При подключении, на сервере создается отдельный поток. Клиент в обязательном порядке передает данные для выполнения аутентификации (логин и пароль), проверку которых выполняет компонент DSAuthenticationManager. В случае успешного прохождения процедуры аутентификации, для передачи данных между сервером и клиентом, DSServer создаст интерфейс взаимодействия. Этим интерфейсом является экземпляр класса TdssmRemoteData.

В случае ошибки аутентификации сервер вызывает исключение и уничтожает поток, который был создан для данного подключения (таблица 3.3).

Описание класса «TdmServer»

Область видимости	Поле	Описание
public	adoConnection: TADOCConnection;	Компонент, отмечающий за соединение с базой данных через технологию ADO.
	DSServer: TDSServer;	Компонент, организующий работу сервера DataSnap.
	DSTCPServerTransport: TDSTCPServerTransport;	Компонент передачи данных между сервером и клиентом.
	DSServerClass: TDSServerClass;	Компонент, определяющий класс, функциональность которого будет доступна подключенным к серверу клиентам.
	DSAAuthenticationManager: TDSAAuthenticationManager;	Менеджер аутентификации.

Как было сказано выше, экземпляр класса TdssmRemoteData создается после успешной аутентификации и служит для предоставления клиентскому приложению определенных данных и функциональности сервера.



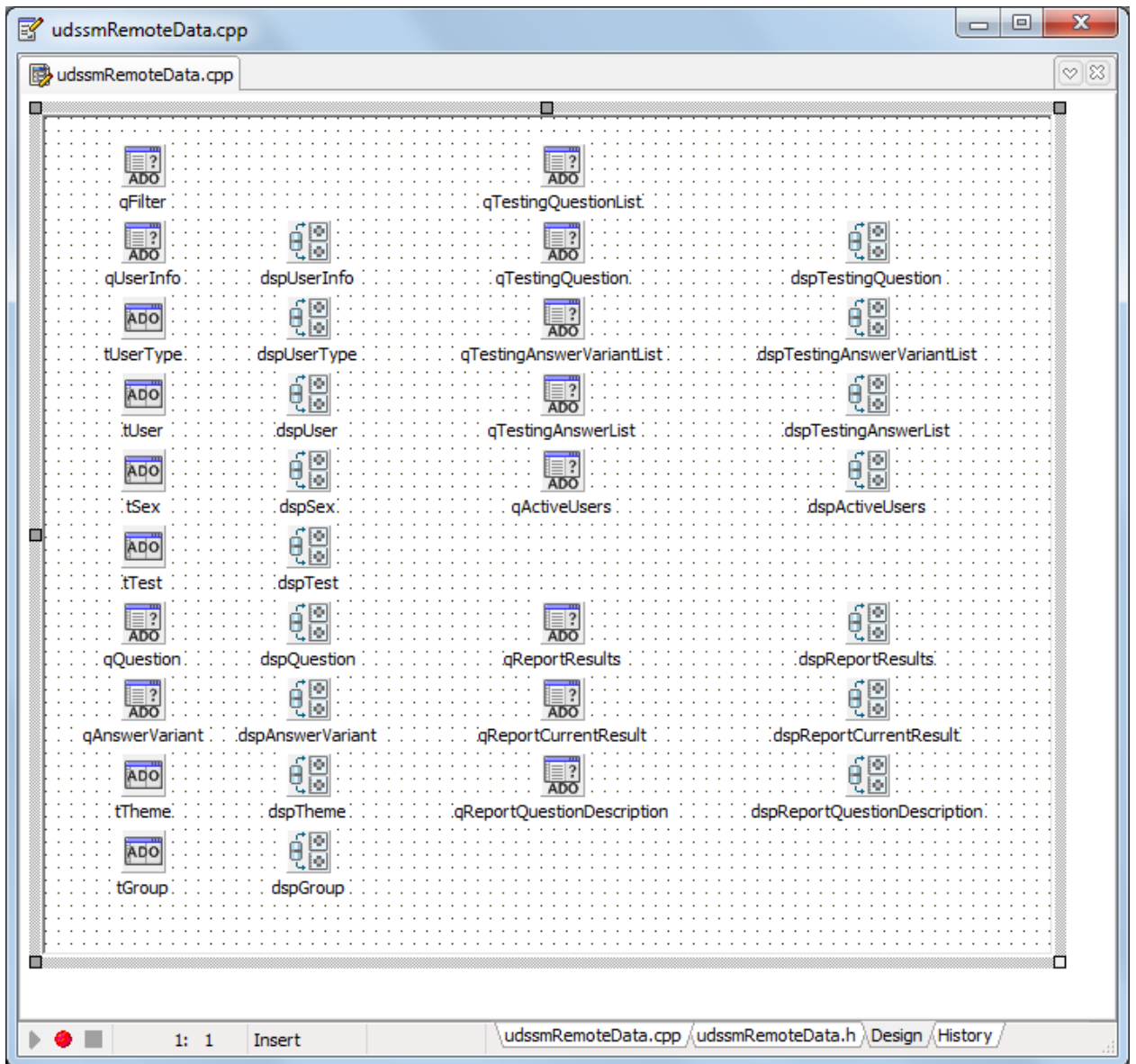


Рисунок 3.18 Состав модуля TdssmRemoteData

Цепочка передачи данных между клиентом и сервером выглядит так:

1. Компонент подключения к БД (TADOConnection);
2. Набор данных (TADOTable или TADOQuery);
3. Провайдер набора данных (TDataSetProvider);
4. Компонент клиентского соединения (TSQLConnection);
5. Клиентский компонент соединения с провайдером данных (TDSProviderConnection);
6. Клиентский набор данных (TclientDataSet).

Пункты 1, 2, 3 – выполняются на стороне сервера, а пункты 4, 5, 6 – на стороне клиента. Компоненты класса `TDataSetProvider` служат для определения прав доступа и настройки параметров передачи данных клиенту. В его настройках можно указать такие параметры как: доступность набора данных для клиента, режим «только чтение», запрет редактирования, запрет вставки, запрет удаления и т.д.

Отдельно хочется отметить реализацию механизма тестирования. Все операции связанные с обработкой результатов тестирования реализованы в интерфейсе взаимодействия и выполняются сервером. Клиентское приложение лишь использует определенный набор методов (через проекцию серверного класса), который и выполняет управление процессом тестирования. Все поля и методы, отвечающие за управление процессом тестирования, описаны ниже, в таблице 3.4.

Таблица 3.4

## Поля серверного класса (реализация механизма тестирования)

Область видимости	Определение	Описание
private	int FcurrentUserKey;	Идентификатор текущего пользователя (прошедшего аутентификацию и работающего в системе). Данный идентификатор сохраняется в таблицу результатов, с целью привязки результатов тестирования к конкретному пользователю.
private	int FtestingTestKey;	Идентификатор теста, который выполняет пользователь (Передается через метод <code>TestingStart</code> ).
private	int FtestingResultKey;	Идентификатор результатов тестирования.
private	TADOQuery *qTestingQuestionList;	Набор данных «Список вопросов теста». Содержит список всех вопросов теста. Между элементами этого списка и выполняются переходы ( <code>TestingNextQuestion</code> , <code>TestingPriorQuestion</code> ).
public	TADOQuery *qTestingAnswerVariantList;	Набор данных «Список вариантов ответа». Содержит возможные варианты ответа на текущий вопрос.
public	TDataSetProvider *dspTestingAnswerVariantList ;	Провайдер набора данных «Список вариантов ответа».
public	TADOQuery *qTestingQuestion;	Набор данных «Вопрос теста». Через него выводится текущий вопрос для клиента.

public	TdataProvider *dspTestingQuestion;	Провайдер набора данных «Вопрос теста».
public	TADOQuery *qTestingAnswerList;	Набор данных «Список ответов». Содержит в себе список ответов, которые пользователь выбрал для текущего вопроса.
public	TdataProvider *dspTestingAnswerList;	Провайдер набора данных «Список ответов».

Таблица 3.5

## Методы серверного класса (реализация механизма тестирования)

Область видимости	Определение	Описание
private	int __fastcall GetTestingQuestionKey ();	Функция для получения идентификатора (ключа) текущего вопроса.
private	void __fastcall TestTestingActive ();	Проверка активности тестирования. Проверяет, выполнен ли запуск тестирования через функцию TestingStart.
private	int __fastcall AddResult (int TestKey);	Функция для добавления записи к таблице результатов. Возвращаемо значение данной функции сохраняется в переменную класса FtestingResultKey (см. поля серверного класса)
private	int __fastcall ComputeBall ();	Функция вычисления балла по результатам тестирования.
private	void __fastcall SaveBall (int Ball);	Функция сохранения балла по тестированию.
private	bool __fastcall TestingTestAnswerAccessory (int AnswerKey);	Функция проверки принадлежности варианта ответа к текущему вопросу.
private	void __fastcall TestingAddAnswerDescription (int ResultKey, int QuestionKey, bool TrueAnswer, UnicodeString Answer);	Функция записи в базу данных истории ответов пользователя.
public	void __fastcall TestingStart (int TestKey);	Функция запуска тестирования. Параметр TestKey идентификатор теста.
public	int __fastcall TestingStop ();	Функция завершения тестирования. Возвращает количество набранных баллов (от 0 до 100).
Public	bool __fastcall TestingNextQuestion ();	Функция перехода к следующему вопросу. Возвращаемое значение функции – это осуществление перехода. Если true – то переход был выполнен, false – если переход не удалось осуществить.
public	bool __fastcall TestingPriorQuestion ();	Функция перехода к предыдущему вопросу. Возвращаемое значение функции – это осуществление перехода. Если true – то переход к предыдущему вопросу был

		выполнен, false – если переход не удалось осуществить.
public	void __fastcall TestingClearCurrentQuestionAnswerList ();	Функция очистки списка ответов для текущего вопроса.
public	void __fastcall TestingAddAnswerInCurrentQuestion (int AnswerKey);	Функция добавления ответа к текущему вопросу.
public	int __fastcall TestingQuestionCount ();	Функция возвращает количество вопросов теста.
public	int __fastcall TestingQuestionNumber ();	Функция возвращает номер текущего вопроса.

Для начал тестирования пользователь должен вызвать метод «TestingStart». Данный метод выполняет начальную подготовку тестирования: делает запись в таблицу результатов (метод «AddResult»), подготавливает список вопросов теста и переводит указатель на первый вопрос тестирования (набор данных «qTestingQuestionList»). После этого пользователь дает ответы по вопросам (методы «TestingClearCurrentQuestionAnswerList» и «TestingAddAnswerInCurrentQuestion»). И выполняет переход к следующему вопросу (метод «TestingNextQuestion»). У пользователя всегда имеется возможность вернуться к предыдущему вопросу (метод «TestingPriorQuestion»).

Для завершения тестирования пользователь должен вызвать метод TestingStop. При вызове данного метода выполняется вычисление балла по результатам тестирования (метод «ComputeBall»), также выполняется документирование ответов с целью их последующего вывода при составлении отчета по результатам тестирования. И в завершении выполняется сохранение балла в таблицу результатов (метод «SaveBall»).

Основой клиентской части является модуль TdmData. Модуль выполняет следующие функции: соединение с сервером; управление предоставленными сервером наборами данных; получение доступа к функциональности сервера (создание проекции интерфейса взаимодействия «TdssmRemoteData»).

Для подключения к серверу используется компонент `SqlConnection`. В качестве параметров подключения мы указываем: тип подключения (`DataSnap`); IP-адрес сервера; порт подключения (5000); протокол взаимодействия (в нашем случае TCP); логин и пароль пользователя.

Для доступа к функциональности и данным сервера используется специальный компонент `DSProviderConnection`, который необходим для подключения клиентского набора данных (`TclientDataSet`) к провайдеру набора данных (`TdataSetDrovider`).

После успешного подключения к серверу выполняется активация доступных наборов данных (открытие набора данных).

После открытия набора пользователь может редактировать набор данных через визуальные компоненты (такие как `TDBGrid`, `TDBEdit`, `TDBMemo`, `TDBImage`, `TDBNavigator` и т.д.), разработанные специально для работы с базами данных. Но визуальные компоненты подключаются не напрямую к клиентскому набору данных, а используют специальный «посредник» (объект класса `TdataSource`) который призваны унифицировать использование визуальных компонентов с различными типами наборов данных.

Клиент также получает доступ к интерфейсу взаимодействия сервера `TdssmRemoteData`. На стороне клиента, через компонент `DSProviderConnection`, создается «проекция» класса `TdssmRemoteData`, в которую входят все методы серверного класса со спецификатором `public`. Класс «проекция» инициализируется сразу после подключения клиента к серверу (рисунок 3.19).

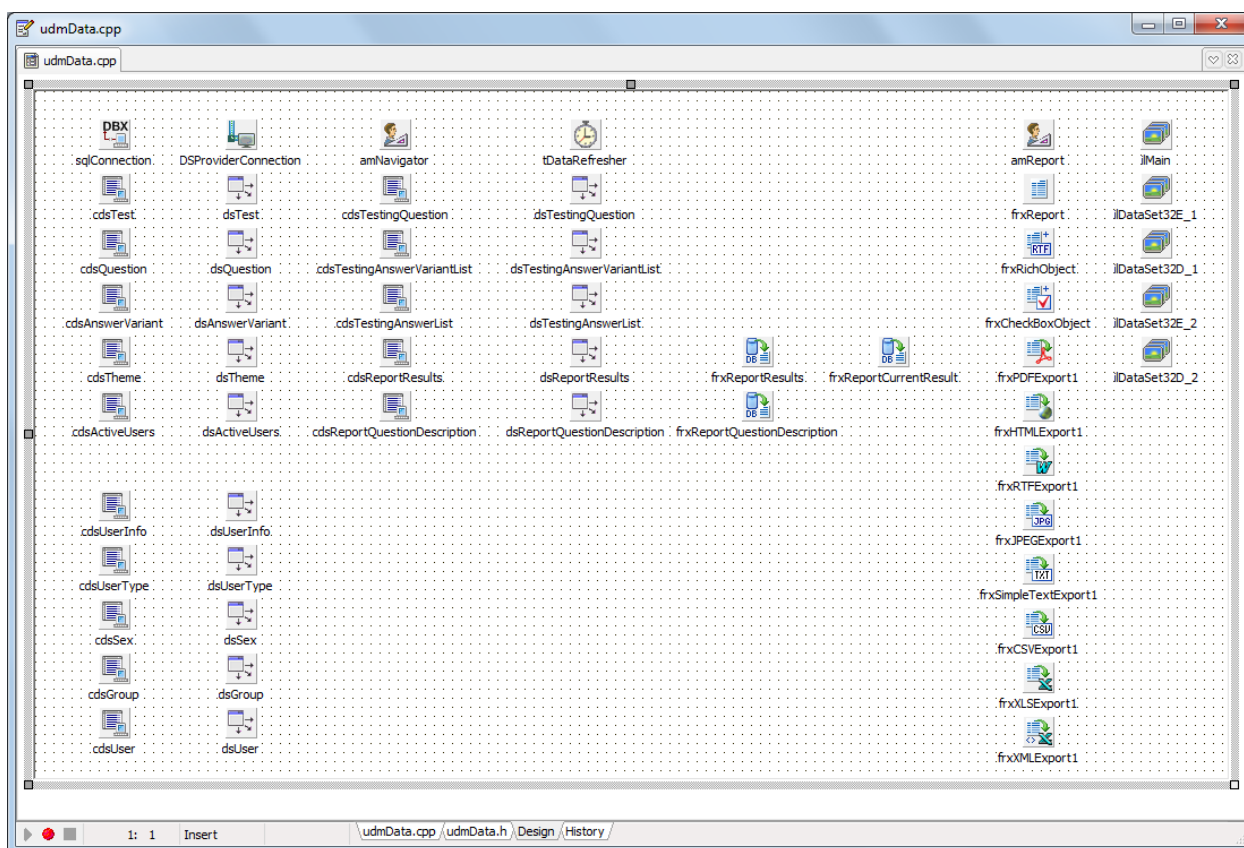


Рисунок 3.19 Схема модуля TdmData

Все отчеты системы сконцентрированы в модуле TdmReport и формируются при помощи генератора отчетов «FastReports». Простота в использовании, высокая производительность и надежность генератора отчетов делают его непревзойденным лидером в своей области. Концентрация всех отчетов в одном модуле, а также использование компонента TactionManager позволяет с легкостью вызывать отчеты из различных модулей интерфейса системы. Ядро Fast Reports позволяет экспортировать созданные отчеты в различные форматы: PDF, XLS, ODT, RTF и т.д.

### 3.4 Технико-экономическое обоснование разработки системы оценки качества обучения

Произведем сравнение суммы затрат для базового варианта и программы автоматизации. Допустим, что средняя зарплата сотрудника

составляет 35000, что в перерасчете на час при 21 восьмичасовом рабочем дне составляет 208 рублей в час.

Если допустить, что объем одной записи составляет около 10 документострок, объем сведений об обучаемых – 10 документострок, запись, содержащая сведения о выполненных работах – 20 документострок, а объем отчета - 200 документострок, объем отчетов, получаемых из архива – 400 строк.

Тогда за год получаем:

Всего записей –  $10 \cdot 21 \cdot 12 = 2520$  документострок;

Всего строк при регистрации пользователя –  $10 \cdot 21 \cdot 12 = 2520$  документострок;

Всего сведений о работах –  $20 \cdot 21 \cdot 12 = 5040$  документострок;

Всего отчетов –  $200 \cdot 4 \cdot 12 = 9600$  документострок.

Всего отчетов из архива –  $400 \cdot 4 \cdot 12 = 19200$  документострок.

Тогда операции технологического процесса при базовом и проектном варианте за год и их характеристики представлены в таблице 3.6 и таблице 3.7.

Таблица 3.6

Базовый вариант

№ п/п	Наименование операций технологического процесса решения комплекса задач	Единица измерения	Объем работы в год	Норма выработки (опер/в час.)	Трудоемкость (Ч)	Среднечасовая зарплата специалиста (руб.)	Стоимостные затраты для ручных операций (руб.)
1	Учет пользователей	документострока	2520	80	31,5	208,3333	6562,5
2	Учет преподавателей	работа	2520	50	50,4	208,3333	10500
3	Учет выполненных	работа	5040	100	50,4	208,3333	10500

	обучаемыми работ						
4	Формирование отчетов	работа	9600	120	80	208,3333	16666,66667
5	Поиск информации в архиве	работа	19200	100	192	208,3333	40000
	Итого:				404,30		84229,17

Таблица 3.7

## Проектный вариант

№ п/п	Наименование операций технологического процесса решения комплекса задач	Оборудование	Ед. Изм.	Объем работы в год	Норма выработки / производительность устройств ЭВМ (опер/в час.)	Трудоемкость (Ч)	Среднемесячная зарплата специалиста (руб.)	Стоимостные затраты (руб.)
1	Учет пользователей	ЭВМ	работа	2520	300	8,4	208,3333	1750,00
2	Учет преподавателей	ЭВМ	работа	2520	250	10,08	208,3333	2100,00
3	Учет выполненных обучаемым и работ	ЭВМ	работа	5040	300	16,8	208,3333	3500,00
4	Формирование отчетов	ЭВМ	работа	9600	350	27,429	208,3333	5714,29
5	Поиск информации в архиве	ЭВМ	работа	19200	300	64	208,3333	13333,33
	Итого:					129,60		26397,62

Абсолютный показатель снижения трудовых затрат на обработку информации:

$$\Delta T = 404,30 - 126,71 = 277,59 \text{ часов}$$



Коэффициент снижения трудовых затрат

$$K_T = (126,71/404,30) * 100\% = 31\%$$

Показатель снижения стоимостных затрат

$$\Delta C = 84229,17 - 26397,62 = 57831,55 \text{ рублей}$$

Рассчитаем затраты на внедрение и проектирование системы. Для этого учтем заработную плату разработчиков. При средней заработной плате 35000, привлечении двух разработчиков и общем сроке разработки 21 рабочий день (месяц), принимаем стоимость разработки – 70000 рублей.

Срок окупаемости затрат на внедрение проекта машинной обработки информации:

$$T_{ок} = 70000/57831,55 = 1,21 \text{ года}$$

Окупаемость затрат на внедрение проекта составляет около 15 месяцев.

Рассчитаем расчетный коэффициент эффективности капитальных затрат:

$$E_p = 1 / T_{ок} = 1/1,21 = 0,82$$

На рис. 3.20. приведена диаграмма сравнения базового и проектного вариантов трудовых затрат, на рис. 3.21 – стоимостных затрат.



Рисунок 3.20 Диаграмма сравнения базового и проектного варианта трудовых затрат обработки информации.

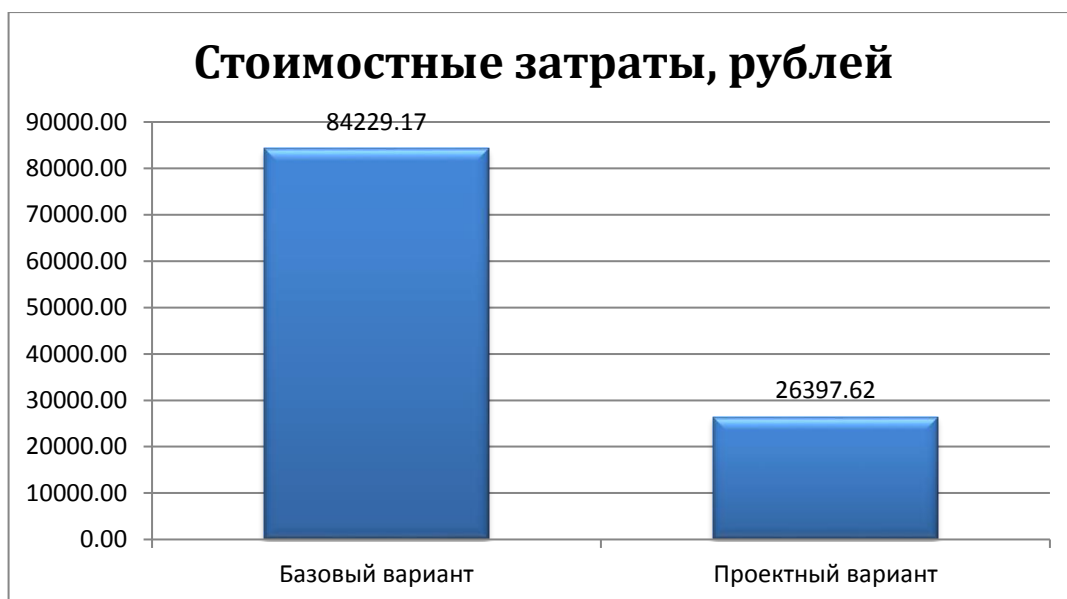


Рисунок 3.21 Диаграмма сравнения базового и проектного варианта стоимостных затрат обработки информации.

### Выводы по Главе 3

В третьем разделе приведено испытание системы, в том числе описаны руководства пользователя, системного программиста. Приведено технико-экономическое обоснование разработки системы оценки качества обучения.

## Заключение

В настоящей выпускной квалификационной работе рассмотрены вопросы использования информационных технологий в образовании на примере преподавания иностранного языка. В работе спроектирована и разработана информационная система тестирования знаний обучаемых, причем для наполнения программы использовались тестовые задания по английскому языку, но для работы программы можно использовать любой другой предмет школьной программы.

Работа состоит из 3 разделов. В первом разделе описана предметная область и обоснована необходимость автоматизации системы оценки качества обучения.

Во втором разделе проведено проектирование системы, в том числе обоснована информационное обеспечение системы, произведен выбор средств управления данными (СУБД), проведено проектирование базы данных, дано описание каждой таблицы и ее полей, описана организация сбора и обработки информации.

В третьем разделе приведено испытание системы, в том числе описаны руководства пользователя, системного программиста, оценена экономическая эффективность.

В данной выпускной работе подтверждена актуальность темы исследования. Создание систем автоматизации процессов педагогической диагностики является необходимым фактором повышения эффективности. Цель работы достигнута, система автоматизации разработана.

## Библиографический список

1. А. Васильев: Самоучитель С++ с примерами и задачами, Москва, Издательство: Наука и Техника, 2015 г., 480 стр.
2. Арнольд Виллемер: Программирование на С++, Москва, Издательство: Эксмо, 2013 г., 528 стр.
3. Белл, Киндал, Талманн: Обеспечение высокой доступности систем на основе MySQL, Издательство: BHV, М., 2012 г., 624 стр.
4. В. Тимофеев: Самоучитель С++ как он есть, Москва, Издательство: Бином, 2009 г., 336 стр.
5. Викрам Васвани: MySQL: использование и администрирование, Издательство: Питер, СПб, 2011 г., 368 стр.
6. Герберт Шилдт: С++ для начинающих, Москва, Издательство: Эком, 2010 г., 640 стр.
7. Морган, Тернстрем: Проектирование и оптимизация доступа к базам данных Microsoft SQL Server 2005, Издательство: BHV, М., 2008 г., 480 стр.
8. Скотт Мэйерс: Эффективное использование С++. 55 верных способов улучшить структуру и код ваших программ, Москва, Издательство: ДМК-Пресс, 2014 г., 300 стр.
9. Тахагхогхи, Вильямс: Руководство по MySQL, Издательство: BHV, М., 2011 г., 544 стр.
10. Ховард, Лебланк, Виэга: Как написать безопасный код на С++, Java, Perl, PHP, ASP.NET, - Москва, Москва, Издательство: ДМК-Пресс, 2014 г., 288 стр.
11. Энтони Уильямс: Параллельное программирование на С++ в действии. Практика разработки многопоточных программ, Москва, Издательство: ДМК-Пресс, 2014 г., 672 стр.