Образовательная автономная некоммерческая организация высшего образования

«МОСКОВСКИЙ ОТКРЫТЫЙ ИНСТИТУТ»

Направление подготовки: 09.03.03 Прикладная информатика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Разработка программного обеспечения оценки знаний студентов

Студент *Сокол Александр Александрович*

(Фамилия, имя, отчество) *(подпись)*

Руководитель *Карпов Александр Сергеевич*

(Фамилия, имя, отчество) *(подпись)*

Москва

2019

Содержание

Введение.....4
Глава 1. Анализ теоретических основ программного обеспечения при обучении студентов.....6
1.1. Анализ обучающего программного обеспечения с возможностью оценки знаний.....6
1.2. Обоснование использования программного обеспечения при обучении.....10
1.3. Анализ методик обработки результатов оценки знаний.....16
Глава 2. Проектирование концептуальной модели базы данных и логической схемы модели данных.....33
2.1. Разработка требований к программному обеспечению.....33
2.2. Проектирование концептуальной модели базы данных.....34
2.3. Проектирование логической схемы модели данных.....45
Глава 3. Разработка программных модулей и анализ их тестирования.....51
3.1. Обоснование выбора языка программирования и средств разработки.....51
3.2. Описание программных модулей.....54
3.3. Тестирование программного обеспечения.....57
Заключение.....75
Список использованной литературы.....78

Введение

Современные тенденции развития образовательной системы направлены к переходу от второго поколения образовательных стандартов к третьему поколению, что влечёт за собой значительные изменения в системе образования ВУЗов. По сравнению со стандартом второго поколения, стандарт третьего поколения дает более содержательное, структурированное описание задач профессиональной деятельности будущих специалистов и тех обязательных компетенций, владение которыми обеспечивает выполнение этих задач.

В этой связи наибольшую сложность для ВУЗов составляют программы непрерывного и дистанционного образования. Особенностью концепции непрерывного образования в России является направленность на постоянное обновление учебных программ для наиболее полного их соответствия современным достижениям науки. Дистанционное же образования должно, как и очное, соответствовать концепции третьего поколения, которая, помимо знаний, предполагает компетентностный подход. Из последнего следует, что студент по выпуску из учебного заведения должен не только обладать некоторыми знаниями, но и уметь променять их в профессиональной области, и весь образовательный процесс подчиняется этой цели.

В такой обстановке весьма актуальной становится разработка обучающих информационных систем, соответствующих новым стандартам образования. Данная область как отличается новизной, поскольку обучающие системы для стандартов третьего поколения ещё не разработаны, так и представляет собою научный интерес, т.к. разработка обучающей системы новой концепции требует нового подхода.

Существует довольно много программ, реализующих методы дистанционного образования. Однако далеко не все они позволяют

производить аналитику результатов тестирования с целью понять адекватность теста, его сложность, а также подвести статистические данные.

Объектом исследования является образовательный процесс. Предметом исследования является программное обеспечение оценки знаний студентов. В качестве информационной базы для исследований используются образовательные стандарты, требования к учебным программам, а также методики у указания к составлению обучающих систем.

Целью выпускной квалификационной работы является разработка программного обеспечения, позволяющего автоматизировать и оптимизировать процесс получения знаний студентами и проверки этих знаний.

Для достижения данной цели необходимо решить следующие задачи:

- Анализ основ построения обучающих систем;
- Анализ предметной области;
- Разработать концептуальную модель данных ERD;
- Разработать логическую модель данных IDEFIX;
- Разработать пользовательский интерфейс для программного обеспечения;
- Разработать базы данных;
- Разработать программные приложения для оценки знаний студентов.

На практике данная работа имеет большую значимость, поскольку позволяет автоматизировать элементы процесса обучения.

Отдельные компоненты разработанной информационной системы планируются к внедрению для обучения студентов.

В написании выпускной квалификационной работы использовались методы анализа, моделирования и формализации.

Глава1. Анализ теоретических основ программного обеспечения при обучении студентов

1.1 Анализ обучающего программного обеспечения с возможностью оценки знаний

Созданная Институтом образовательных технологий система дистанционного обучения «Прометей» позволяет организовать учебный процесс таким образом, что с одной стороны дает возможность преподавателю (тьютору), находясь в любом месте, где есть доступ к Интернет, — дома или на работе, всесторонне контролировать процесс обучения слушателя, сдачу экзаменов и прохождение тестов, с другой стороны слушателю — приобрести качественные знания и в конечном итоге – документ, подтверждающий получение высшего или дополнительного образования.

Обучающая система Moodle — автоматизированная система контроля обучающих курсов (система управления содержимым), также популярная как система управления учебным процессом или виртуальная лаборатория обучения. Является свободным (распространяется по лицензии GNU GPL) веб-приложением, которое предоставляет возможность создавать интерактивные курсы и тесты для обучения через локальную сеть или Интернет.

Система реализует концепцию «педагогики социального конструкционизма» и предназначена, прежде всего, для организации сообщений между преподавателем и студентами, хотя её можно применить и для создания традиционных удалённых курсов, а также дополнения к очному обучению.[5]

Moodle написана на PHP с использованием SQL-базы данных (MySQL, PostgreSQL, Microsoft SQL Server и др. БД — используется ADOdb XML). Moodle может работать с объектами SCO и

отвечает стандарту SCORM.

Благодаря развитой модульной архитектуре, возможности Moodle могут легко расширяться сторонними разработчиками. Помимо языковой поддержки и шаблонов оформления, Moodle позволяет подключать также следующие типы модулей:

- элементы курса;
- отчеты администратора;
- типы заданий;
- плагины аутентификации;
- блоки;
- форматы курсов;
- отчеты по курсам;
- поля базы данных (для элемента курса "база данных");
- плагины подписки на курсы;
- фильтры.

Преподаватель может по своему усмотрению использовать как тематическую, так календарную структуризацию курса. При тематической структуризации курс разделяется на секции по темам. При календарной структуризации каждая неделя изучения курса представляется отдельной секцией, такая структуризация удобна при дистанционной организации обучения и позволяет учащимся правильно планировать свою учебную работу.[17]

Редактирование содержания курса проводится автором курса в произвольном порядке и может легко осуществляться прямо в процессе обучения. Очень легко добавляются в электронный курс различные элементы: лекция, задание, форум, глоссарий, wiki, чат и т.д. Для каждого электронного курса существует удобная страница просмотра последних изменений в курсе.

Таким образом, LMS Moodle дает преподавателю обширный

инструментарий для представления учебно-методических материалов курса, проведения теоретических и практических занятий, организации учебной деятельности школьников как индивидуальной, так и групповой.

Администрирование учебного процесса достаточно хорошо продумано. Преподаватель, имеющий права администратора, может регистрировать других преподавателей и учащихся, назначая им соответствующие роли (создатель курса, преподаватель с правом редактирования и без него, студент, гость), распределять права, объединять учащихся в виртуальные группы, получать сводную информацию о работе каждого студента. С помощью встроенного календаря определять даты начала и окончания курса, сдачи определенных заданий, сроки тестирования. Используя инструмент Пояснение и Форум, публиковать информацию о курсе и новости.[24]

Ориентированная на дистанционное образование, система управления обучением Moodle обладает большим набором средств коммуникации. Это не только электронная почта и обмен вложенными файлами с преподавателем, но и форум (общий новостной на главной странице программы, а также различные частные форумы), чат, обмен личными сообщениями, ведение блогов.

Moodle имеет не только многофункциональный тестовый модуль, но и предоставляет возможность оценивания работы обучающихся в таких элементах курса как Задание, Форум, Wiki, Глоссарий и т.д., причем оценивание может происходить и по произвольным, созданным преподавателем, шкалам. Существует возможность оценивания статей Wiki, глоссария, ответов на форуме другими участниками курса. Все оценки могут быть просмотрены на странице оценок курса, которая имеет множество настроек по виду отображения и группировки оценок.

Одной из наиболее развитых программных систем дистанционного обучения является продукт LearningSpace фирмы Lotus. LearningSpace обеспечивает легкое внедрение в создаваемые курсы видео и аудио

фрагментов, позволяет реализовать весь диапазон услуг от самообучения до коллективного синхронного, дает преподавателям и студентам доступ ко всем своим функциям как из клиента Notes (в локальной сети учебной аудитории), так и из Web-броузера (дистанционное обучение). Базовая архитектура этого пакета основывается на пяти главных баз данных, которые используются для управления различными аспектами, как создания курсов, так и их сопровождения:

- Schedule – менеджер списков.
- Media Center – менеджер библиотеки курсовых мультимедийных материалов.
- The CourseRoom – инструмент электронного взаимодействия, позволяющий создавать как общие студенческие форумы, так и специализированные.
- The Profile Manager – содержит базовую информацию о участниках, наподобие домашней страницы или онлайн CV.
- The Assessment Manager – инструмент тьютора для проведения конфиденциальных тестирований, оценки их результатов и отсылки результатов испытуемым. Он также позволяет создавать журнал успеваемости всех обучаемых данного курса.
- Пакет LearningSpace включает инструменты для просмотра webдокументов и средства для включения мультимедиа материалов в документы данного пакета. Связи могут быть установлены из пакета LearningSpace к мультимедийному содержимому web-документов.

Одним из наиболее удобных инструментальных средств создания тестовых скриптов является пакет Hot Potatoes. Данный

продукт является бесплатным для личного использования и для применения его образовательными учреждениями в некоммерческих целях. Данный пакет предоставляет возможность создания различных видов тестов, имеет интуитивно понятный и весьма удобный интерфейс. Hot Potatoes содержит заготовки на HTML и JavaScript, при помощи которых легко создать тесты (с пятью вариантами ответов и комментариями к каждому из них), опросники, анкеты, обучающие кроссворды 20x20, задания по проверке правописания (орфография и синтаксис) и знания синонимов, антонимов и пр. Каждое из заданий преподаватель может сопроводить системой подсказок и построить по многовариантной схеме. [18]

Автоматизированная система Дистанционного обучающего центра (ДОЦЕНТ) системы асинхронного телеобучения, представляет собой комплекс программно-методических средств ДО, переподготовки и тестирования слушателей. Комплекс программно-методических средств основан на современных Интернет-технологиях и методиках образования на базе компьютерных обучающих программ и тестирующих систем.

1.2 Обоснование использования программного обеспечения при обучении

Дистанционное обучение (ДО) – это система обучения, основанная на взаимодействии преподавателя и учащихся, учащихся между собой на расстоянии, отражающая все присущие учебному процессу компоненты (цели, содержание, организационные формы, средства обучения) специфичными средствами ИКТ и Интернет-технологий[4].

В связи с тем, что дистанционная форма обучения не введена

в Закон об Образовании, существует много проблем при внедрении ДО в образовательный процесс:

- нет нормативной базы по поводу подготовки специалистов в области ДО;
- не разработано единых требований к созданию дистанционных курсов, электронных учебников;
- нет законодательных актов о нормировании труда преподавателей и охране их интеллектуальной собственности и т.д.;
- нет нормативов по оплате труда дистанционных преподавателей и авторов дистанционных курсов[5].

Поэтому каждое учебное заведение, желающее внедрить ДО, вынуждено решать все организационные и финансово-экономические вопросы самостоятельно.

Организация ДО в образовательном учреждении начинается с создания модели ДО, далее на её основе строится системы ДО, разрабатывается экономическая схема ДО, на курсах повышения квалификации готовятся кадры к созданию дистанционных курсов. [13]

Можно выделить следующие модели ДО, которые могут существовать сегодня в России в рамках действующего законодательства[6]:

1.Интеграция очных и дистанционных форм обучения.

2.Сетевое обучение (автономные сетевые курсы и информационно-образовательная среда).

3.Дистанционное обучение и кейс-технологии.

4.Интерактивное телевидение, видеоконференции.

Выбор модели ДО – стратегический ход в развитии организации. От модели будет зависеть набор средств ИКТ, педагогических

технологий и, следовательно, объём трудозатрат на её реализацию. В соответствии с выбранной моделью (моделями) необходимо разрабатывать систему оплаты труда.[17]

Для обучения студентов по программам заочного обучения и индивидуального заочного с применением ДОТ разработана модель для дистанционного обучения «Дистанционное обучение и кейс-технологии» с последующим построением модели «Сетевое обучение». Более наглядно её можно представить в виде схемы (рисунок 1) [4].

В модели ДО присутствуют две формы обучения:

1.Очное обучение.

2.Дистанционное обучение:

- кейс (УМК);
- дистанционная оболочка;
- сеть Интернет.

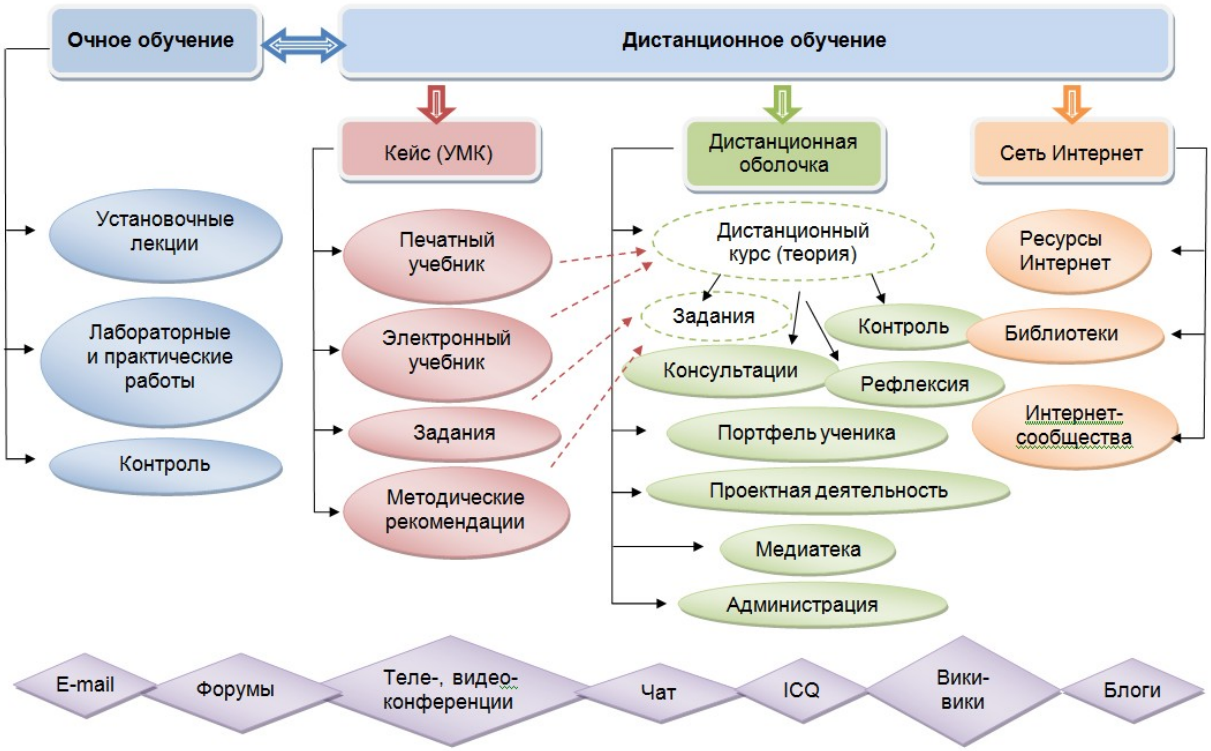


Рисунок 1 – Модель дистанционного обучения

В очной форме при обучении по модели «Дистанционное

обучение и кейс-технологии» предполагается проводить 1-2 раза в год сессии – установочные лекции, лабораторные и практические работы, контроль.

В дистанционной форме блок «Кейс (УМК)» предполагается использовать на начальном этапе, т.к. кейсы (УМК) – это уже имеющиеся ресурсы ДО, требующие небольшой доработки. В результате блок «Кейс (УМК)» планируется перестроить в блоки «Дистанционная оболочка» и «Сеть Интернет», с помощью которых можно будет реализовать модель «Сетевое обучение».

Блок «Кейс (УМК)». На основе составляющих кейса (Печатный учебник, Электронный учебник, Задания, Методические рекомендации) планируется создать дистанционные курсы (с заданиями, возможностью консультирования с преподавателем, контролем и рефлексией), которые будут находиться в дистанционной оболочке. Пунктирная линия на схеме показывает трансформацию кейса в дистанционный курс.

Блок «Дистанционная оболочка». Оболочка МФПА должна быть доработана для поддержания возможности реализации таких педагогических технологий, как «Портфель студента», «Проектная деятельность» и различных видов контроля. Для помощи студентам в оболочке уже существует «Медиатека» – справочные материалы, учебники, виртуальные тренажёры. В оболочке также должен размещаться административный ресурс для организации учебного процесса – знакомство (визитные карточки преподавателей, студентов), документация учебного процесса, доска объявлений [5].

Блок «Сеть Интернет». Ресурсы сети Интернет также могут быть вписаны в систему ДО: виртуальные библиотеки, профессиональные Интернет-сообщества для педагогов и студентов.

В модели также зафиксированы средства коммуникаций (для

индивидуального и группового обучения), которые могут применяться при различных педагогических технологиях ДО: e-mail, форум, теле-, видеоконференции, чат, ICQ, вики-вики, блоги.[17]

Дистанционное обучение – процесс значительно более трудоемкий и многоаспектный, чем очное. Существуют нормативы для оплаты труда преподавателей ВУЗов, школ при очном обучении. Но при ДО потребуется больше времени хотя бы потому, что информация вводится через клавиатуру, а не проговаривается устно. Следовательно, все обсуждения, дискуссии, ответы на вопросы при ДО растягиваются на часы и дни, тогда как при очном общении могут занимать минуты. Также комментирование заданий студентов преподавателем по e-mail, в чате или через форум требует значительных временных затрат [6].

При ДО наиболее эффективно можно организовать учебный процесс в том случае, если количество обучаемых в группе не будет превышать 15-20 человек. Эти данные выведены эмпирическим путём на основе анализа опыта различных курсов и программ ДО. Такое число обучаемых связано с необходимостью обеспечить постоянное общение преподавателя с каждым отдельным студентом и группой в целом, индивидуальный подход (что и в условиях очного обучения возможно лишь при той же пропорции - 1:20).

Требования обеспечения наглядности в случае использования электронного учебного материала могут и должны быть реализованы на принципиально новом, более высоком уровне. В перспективе, когда получат большее распространение системы виртуальной реальности, обеспечивающие воздействие на зрение, слух, осязание и даже обоняние, по-видимому, целесообразнее будет говорить не только о наглядности, но и о полисенсорности обучения, т.е. чем больше органов чувств участвует в процессе восприятия

информации, тем успешнее процесс познания. Вместе с тем, несмотря на всеобщее признание высокой значимости принципа наглядности обучения, в науке практически отсутствуют теоретические основы создания педагогически эффективных наглядно-образных представлений для электронных учебников[7].

Цель разработки учебного материала – сохранить не только все достоинства печатного учебного материала, но и, используя возможности компьютера, включить в него соответствующий арсенал наглядных средств.

Иллюстрации рекомендуется использовать в местах:

- трудных для понимания учебного текста, требующих дополнительных иллюстраций;
- для обобщений и систематизации тематических смысловых блоков;
- для общего оживления всего учебного материала и рассредоточенного по всему полю текста как печатного, так электронного (гипертекста).

Нецелесообразно предоставлять студенту "неограниченную" свободу передвижения (за счет средств навигации) по всему полю гипертекста к другим объектам ссылок, не сообразуясь с педагогическими методами и способами преподавания предмета, а также с правилами целесообразности и разумности. В результате таких бессмысленных и частых переходов: то на картинку, то на словарь, то на какой-то другой объект гипертекста – в голове учащегося остается лишь "гипертекстовый сумбур, каша", а вовсе не стройная система знаний. [12]

Студента, безусловно, нельзя лишать свободы навигации (маршрутов движения) по учебному материалу. Необходим разумный компромисс, основывающийся на таких ключевых педагогических

понятиях из области методики преподавания, как линейное прохождение материала, или линейно-концентрическое, или модульное (о чем прекрасно осведомлены преподаватели соответствующих дисциплин).

Нежелательно, когда в гипертексте рябит от подвижных картинок (анимация-изображение иллюстрации в динамике), которые часто расположены не к месту, но зато создают эффект красочного "фейерверка". Такая анимация ничего не дает для понимания учебного материала, в отличие от разумной развлекательности. Но та, которая работает на повышение эффективности изучения теоретического материала, – сокращает временные трудозатраты студента.

Так сколько должно быть иллюстраций, например, в одной теме? Все диктуется содержанием и характером теоретического учебного материала. Если автор в состоянии заменить, например, какой-то теоретический абзац соответствующим видом иллюстрации, то это хорошо; а если сможет переложить все словесно-теоретическое изложение курса на одни картинки, то еще лучше (если такое возможно). Перефразируя пословицу, можно сказать и так: "Лучше увидеть один раз иллюстрацию, чем долго читать скучный печатный или электронный текст".

Конкретное количество иллюстраций для страницы, темы или для всего курса лекций, таким образом, специально не устанавливается, а определяется следующими:

- содержанием и характером учебного текста;
- выбранной методикой обучения;
- возможностями организации (учреждения).

1.3 Анализ методик обработки результатов оценки знаний

Компьютерное тестирование - это средство, которое позволяет с минимальными затратами времени преподавателя объективно проверить знание большого количества студентов.

Хорошо спланированный график тестирований является хорошим стимулом, который побуждает студентов к систематической работе в течение семестра. Также большое значение имеет организация контроля над процессом обучения (рисунок 2).

Компьютерные тесты положительно воспринимаются студентами. Преимуществом компьютерного тестирования является автоматическая проверка результатов и исключение влияния человеческого фактора.

Еще одним преимуществом компьютерных тестов является то, что в процессе проверки знаний студенты видят в преподавателе не оппонента, а союзника.

Количественные характеристики теста:

1. надежность теста;
2. валидность;
3. дискриминативность (дифференцирующая способность);
4. надежность теста от угадывания;
5. статистическое распределение баллов (в зависимости от назначения теста);
6. сложность (трудность) теста;
7. нормирование теста.

Надежность теста показывает, насколько точно данный тест измеряет изучаемое явление.

Надежный тест - это тест, который внутренне согласован, или дает одни и те же результаты для каждого испытуемого при повторном тестировании.

Надежность характеризует точность теста как измерительного

инструмента, устойчивость его к действию помех.

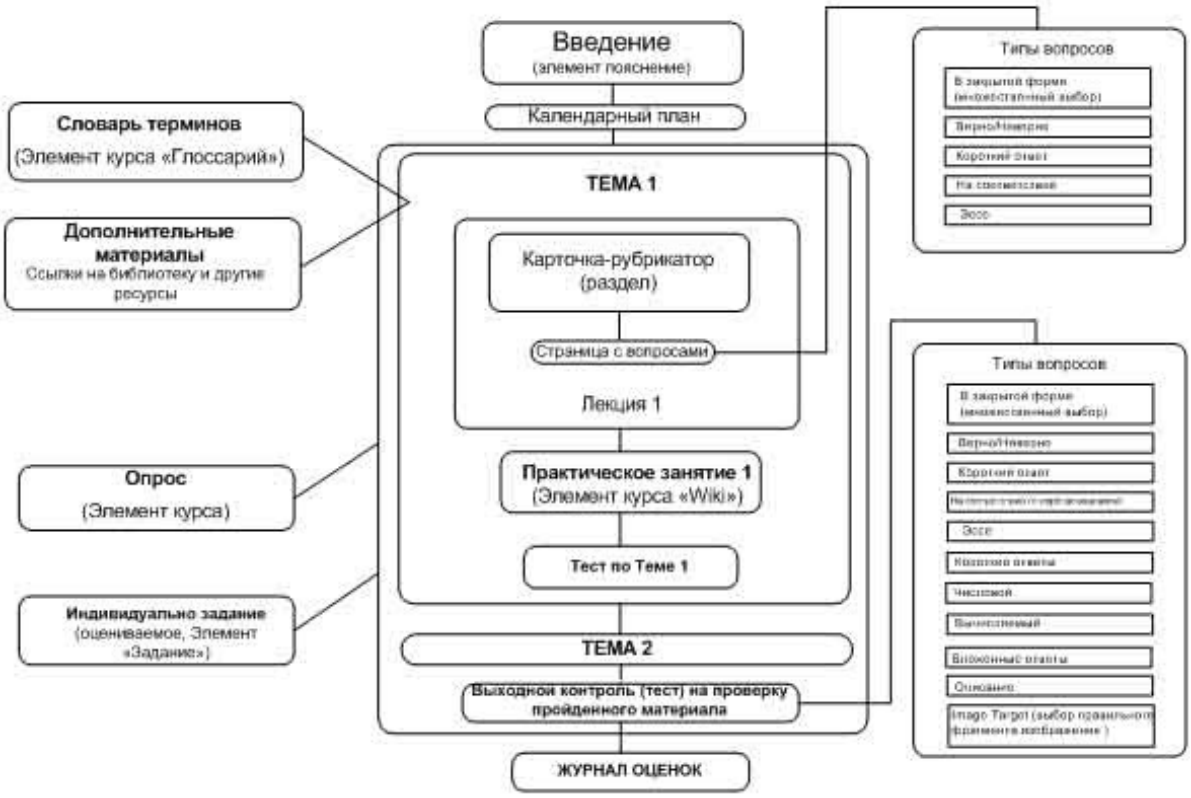


Рисунок 2 - Организация контроля над процессом обучения

На практике используются 3 основных метода оценки ретестовой надежности:

1. повторное тестирование (проводится через 2 недели по тому же тесту);

2. параллельное тестирование (даются 2 параллельные формы теста, измеряющие одно и то же качество);

3. расщепление (делят задания на четные и нечетные, так как внутри теста некоторые формы заданий являются параллельными, отсюда получается 2 теста).

Надежность по внутренней согласованности - это характеристика теста, указывающая на степень однородности состава заданий с точки зрения измеряемого качества.

Для ее определения применяется процедура установления

корреляции между результатами теста в целом и каждым отдельным заданием.

Для оценки ретестовой надежности применяется формула Спирмена-Брауна. Эта формула используется при расщеплении теста на 2 части, или если есть параллельные формы теста.

$$R'_{xx}=2r_{yy'} / (1 + r_{yy'}),$$

$$R'_{xx} = \frac{2 \cdot r_{yy'}}{1 + r_{yy'}} \quad (1)$$

где R - коэффициент надежности;

r - корреляция (теснота связей) между двумя половинками теста $r_{yy'}<1$.

Если $r_{yy'}$ близок к нулю, значит в тесте есть недочеты.

Если $r_{yy'}$ близок к минус единице, значит параллельные формы теста не связаны.

r должно стремится к 1.

Коэффициент Кронбаха:

$$k=R_{xx}=N/(N+1)*[1- E^{N}_{i=1} D_{yi}/D_x], \quad (2)$$

где N - количество заданий теста;

D_x - дисперсия результатов (мера рассеяния, разброса всех участников);

D_i - дисперсия результатов по заданию с N;

$R'_{xx} \geq 0,7$ – надежность.

Если хотя бы один коэффициент ≥ 0,7, то тест считается надежным.

Для оценки надежности теста по внутренней согласованности используется формула Пирсона:

$$r_{iE}= S_{xi}*(M_b-M_н)/S_x , \quad (3)$$

где r_{iE} - коэффициент корреляции между вектором i-го задания и общим показателем;

M_b - среднее арифметическое по всему тесту у испытуемых, успешно ответивших на задание i.

$$M_b=(1/N_6)* E \text{ баллов по тесту, (4)}$$

где N_b - количество испытуемых, успешно ответивших на задание;

$M_н$ - аналогично M_b, но у испытуемого, не справившегося с заданием i;

S_x - стандартное отклонение по всему тесту у всей группы испытуемых;

S_{xi} - стандартное отклонение по i-му заданию у всей группы испытуемых;

r_{iE} должно быть ≥ 0,3.

Эту величину считаем для каждого задания.

Валидный тест - это тест, который измеряет то, для измерения чего предназначен.

В тестологии выделяют следующие виды валидности:

1. очевидная;
2. содержательная;
3. конструктная;
4. критериальная;
5. конкурентная (сравнительная).

Очевидная валидность – это валидность с точки зрения испытуемого, т.е. впечатление, что тест измеряет именно то, для чего предназначен.

Содержательная валидность устанавливается экспериментами и

связана с отношением тестового материала к целям тестирования.

Конструктная валидность применяется в основном в социологии, в педагогике не применяется.

Критериальная валидность. Этот вид валидности является основным. Суть его заключается в способности теста служить индикатором строго определенных особенностей, форм поведения и состоит в сравнении результатов, полученных испытуемыми за решение теста с данными по критерию и вычисление корреляции между ними. В качестве критерия может выступать любой показатель, измеряющий ту же характеристику, что и валидизируемый тест. Например, оценки.[26]

Конкурентная валидность определяется по корреляции выполнения данного теста с результатами других тестов, которые доказательно предназначены для измерения той же самой величины.

Дискриминативность – это способность отделять испытуемых с высоким общим баллом по тесту от тех, кто получил низкий балл.

Дискриминативность означает различительную способность задачи.

Коэффициент дискриминативности рассчитывается с применением крайних групп (отличники и двоечники). Существует несколько подходов для выбора крайних групп:

1) количество испытуемых в крайних группах одинаково (берут по 27% от общего количества);
2) берут группы с высоким и низким показателем испытуемых, после чего считается количество студентов, попавших в группы.

$$D = N_{р\ верх} / N_{верх} - N_{р\ низ} / N_{низ}, \quad (5)$$

где D – коэффициент дискриминативности;

N_p – количество испытуемых, справившихся с заданием;

N – количество испытуемых в данной группе.

$$-1 \leq D \leq 1 \quad (6)$$

Если D близко к 1, значит задание хорошо разделяет учащихся на слабых и сильных.

Если D<0, удалить задание из теста. Если D близко к нулю, значит задание некорректно сформулировано. В идеале $D \geq 0,3$

Для простановки оценки вводят коэффициент усвоения.

Нормирование производят по социальным нормам, определяется процент двоечников (таблица 1 и 2).

Таблица 1 – Результат оценки знаний перед нормировкой

Курс	**Процент двоечников**
2	5%
3	10-12%
4	15-20%
5	10%

Таблица 2 – Пример нормировки после результата оценки знаний

Баллы	**Процент испытуемых, получивших данное количество баллов**	**Оценка**
20 19	E=10%	5

...		
Оставшиеся баллы делятся пополам	E=...%	4
	E=...%	3
... 1 0	E=10%	2

Способы при помощи которых определяется результативность учебно-познавательной деятельности обучаемого называются методами контроля знаний.

Как уже указывалось ранее, в практике обучения используются методы устного, письменного, машинного контроля и самоконтроля.

Устный контроль проходит путем устного опроса для учащихся и беседы со специалистом для сертифицируемого специалиста. Выявление знаний происходит не только на уровне конкретных понятий, а учитывается логика изложения и реакция обучаемого на различные ситуативные задачи

Письменный контроль проводится на базе заданий в широком смысле задания могут быть игрового, проектного и другого характера. В высших учебных заведениях это обычно (контрольные работы, сочинения, отчеты и т.д.). Это методика требует времени не меньше чем устный опрос, а иногда не позволяет объективно оценить работы без личной встречи, что реализуется с помощью технологий апелляций.

Практический контроль выявление практических навыков, реализуется проведением лабораторных, практических работ, а также участием в пилотных разработках и проектах.

Особую роль в последнее время стал играть тестовый контроль. Многие исследователи [Error: Reference source not found] отмечают несовершенство такого контроля, однако нельзя не признавать минимальные

трудозатраты на данный вид контроля знаний. Учитывая широкое применение этой технологии, возникли разные направления тестирования.

В работе [Error: Reference source not found] предложена методика оценки обученности учащихся по четырем уровням усвоения:

Первый уровень – уровень повторения или узнаваемости. Уровень соотнесения, выбора или распределения объектов, процессов. Возможность сформулировать их свойства или признаки, по которым выявляются эти объекты.

Второй уровень – уровень воспроизведения, повторения действий по решению задач или ситуаций на основе разобранных и изученных алгоритмов, демонстрация умения решать типовые задачи.

Третий уровень – уровень знаний, умений и навыков. Принципиальное применение методик для задач с измененными условиями и проявление умения синтезировать методы и выделять частные случаи. решение нестандартных задач выявление новых ролевых ситуаций, действия в деловых играх т.п.

Четвертый уровень – уровень творчества. Творческий процесс – это зарождение нового, в контурах уровней можно выделить как самостоятельные постановки заданий на основе ситуаций, выбор методик и планирование. Поиск нестандартных путей анализа и самостоятельное добывание новых знаний путем наблюдения, эксперимента, работа по проектам.

Каждому уровню знания соответствуют уровни тестирования [Error: Reference source not found].

Тесты первого уровня – тесты по выбору правильного ответа, обычно называются альтернативными или тестами на соответствие.

Тесты второго уровня умение без подсказок восстанавливать суть связей и явлений.

Тесты второго уровня можно разделить на три группы:

- тесты-подстановки,

- конструктивные тесты

- типовые задачи.

Тесты-подстановки состоят в требовании закончить мысль внести понятие и т.п., они могут реализовываться в символьной, графической или другой форме.

Конструктивные тесты состоят в требовании сформировать мысль полностью, без наведения тестируемого, например, дать развернутый ответ на вопрос.

Тесты – типовые задачи состоят в поиске решения, ответ на такой тест может быть заключен в конкретном результате решения такой задачи.

Тесты третьего уровня – нестандартные задачи, требующие от тестируемого эвристической деятельности, например, решения проблемных задач.

Тесты четвертого уровня состоят в требовании создания объектов и явлений, т.е. уровня творчества (исследование, конструирование, эксперимент и т.д.).

Количество выполненных правильно заданий определяет коэффициент усвоения**:** $K_y = n/m$, где m – общее число существующих операций, п – число правильно выполненных существенных операций. Для перевода коэффициента усвоения в балльную систему можно использовать следующую схему: $K_y < 6$ – неудовлетворительно (2), $6 \le K_y \le 7{,}5$ – (3), $7{,}5 \le K_y \le 9$ – (4),

$9 \le K_y \le 10$-(5).

Примерами для каждого уровня могут служить.

Уровень один

Опознавание

Соискатель на должность при наличии образования должен предъявить диплом?

Варианты ответов:

1) да;

2) нет;

3) не знаю.

Различение

Что является необходимым документом подтверждающим образование:

Варианты ответов:

1) диплом;

2) вкладыш к диплому;

3) сертификат.

Классификация

Какие документы подает соискатель если имеет:

1) среднее образование - аттестат

2) высшее образование - диплом

3) категорию специалиста - сертификат.

Уровень два

Тест подстановки

а) В перечень документов по проекту входит ___

б) Доход на физических лиц равен ____

Конструктивные тесты

а) В чем заключается оценка кредитоспособности?

б) Укажите виды кредитования для юридических лиц.

Типовые задачи

а) Определите тело кредита 200000 при процентной ставке 5 % на три года.

б) Оптимизировать выбор пути доставки груза по маршруту.

Уровень три. Проблемная ситуация

Как необходимо спланировать действия доставки в случае отсутствия возможности оптимального распределения, вследствие погодных условий.

Уровень четыре. Исследование

Провести анализ деятельности отдела маркетинга и выявить сотрудников достойных поощрения.

Система процедуры подтверждения соответствия и присвоения квалификации персонала (сертификация) строится на общепринятых стандартах и базируется на следующих **принципах**:

- сертификация призвана оценивать компетентность кандидата (соискателя) со стороны профессионального сообщества (ассоциаций, объединений) объективно и независимо;

- перед сертификацией кандидат должен быть ознакомлен с требованиями профессионального стандарта (ПС) по квалификационному уровню в отраслевой рамке квалификаций (ОРК) либо иными применяемыми законодательными и регламентирующими требованиями к компетенции персонала и схемой подтверждения соответствия (схемой сертификации);

- для получения документа, подтверждающего квалификацию – сертификата, кандидат должен пройти основные этапы процесса сертификации: идентификацию компетенций на соответствие требований ПС и квалификационный экзамен согласно соответствующей схеме сертификации.

В схеме подтверждения соответствия устанавливают порядок проведения сертификации персонала по каждому сертификационному направлению.

Квалификационный экзамен проводится Экзаменационной комиссией органа по подтверждения соответствия и присвоения квалификации (ОПС). В состав Экзаменационной комиссии включаются высококвалифицированные

специалисты соответствующей области деятельности сторонних организаций.

Экзаменационная комиссия должна выполнять требования ОПС, основанные на действующих профессиональных стандартах и других соответствующих документах.

Процесс формирования Экзаменационной комиссии должен гарантировать, что экзаменаторы, назначенные для проведения экзамена, или его части должны:

1) знать схемы сертификации персонала;

2) иметь знания и опыт в разработке экзаменационных методов и экзаменационной документации;

3) обладать соответствующими знаниями в экзаменуемой области;

4) быть независимым от любых интересов для беспристрастного и непредвзятого суждения (оценки).

Уровень квалификации экзаменаторов, должен быть не ниже уровня квалификации кандидатов

При возникновения конфликта интересов членов Экзаменационной комиссии и кандидата, ОПС должен принять меры по обеспечению беспристрастности и конфиденциальности экзамена. Меры должны быть зафиксированы.

Соблюдение принципа оценки каждого кандидата каждым экспертом по каждому экзаменационному заданию [Error: Reference source not found].

Процедура оценивания включает в себя следующие этапы:

1) ОПС должен предоставлять соискателям и кандидатам описание процесса сертификации, соответствующего каждой схеме подтверждения соответствия (включая стоимость), а также документы, содержащие требования, предъявляемые к сертификации, права кандидата и обязанности сертифицированного специалиста, включая, при необходимости, соглашение о правилах поведения сертифицированного специалиста. ОПС осуществляет

информационную поддержку кандидата (процедуры, формы проведения оценки, критерии оценки, требования к кандидатам, роль экзаменаторов и т.д.) ;

2) Сбор и предоставление кандидатом установленного (требуемого) перечня документов (документов об образовании, обучении, дипломы, сертификаты, и т.д.);

3) Составление плана проведения оценки (схемы подтверждения соответствия).

Процедура оценки (сертификации персонала) включает следующие составляющие:

а) практический экзамен - проверка уровня владения практическими навыками и умениями по профессии в соответствии с требованиями ПС либо иным применяемым законодательными и регламентирующими требованиями к компетенции персонала. В процессе сертификации кандидат должен показать приоритетно практические умения и навыки на соответствие требованиям ПС по квалификационному уровню либо иным применяемым законодательными и регламентирующими требованиями к компетенции персонала. Практический экзамен должен проводиться на производственной базе (ресурсные центры, производственные подразделения, испытательные площадки и т.п.);

б) теоретический (устный, письменный, тестовый или другой) экзамен - проверка специальных профессиональных знаний в сертифицируемой области.

При идентификации компетенций соискателя оцениваются его **теоретические знания и уровень подготовки** на соответствие требованиям модели ПС по соответствующему квалификационному уровню согласно ОРК либо иным применяемым законодательными и регламентирующими требованиями к компетенции персонала на основе схем подтверждения соответствия [Error: Reference source not found].

Оценка на уровне идентификации компетенций производится посредством анализа. Для оценки компетентности соискателя при идентификации компетенций необходимо осуществить предварительную работу по формированию перечня контрольных вопросов (по функциональным обязанностям по ПС соответствующего квалификационного уровня).

Перечень контрольных вопросов может состоять как из полного списка компетенций/профессиональных функций (функциональных обязанностей по ПС), так и из его выборки, которая наиболее характеризует требования по профессии/квалификации/специализации и дает возможность **предварительно оценить компетентность кандидата.**

Соискатель получает заключение по результатам идентификации компетенций и допускается к квалификационному экзамену.

Замечание: основной оценкой при идентификации компетенций претендента является допуск или не допуск к квалификационному экзамену.

Сертификационный экзамен содержит две составляющие: практическую и теоретическую. Сертификационные требования могут быть разбиты на две группы [Error: Reference source not found,Error: Reference source not found]:

1) профессиональные, обуславливающие способность обеспечивать необходимое качество результатов своего труда и требуемый уровень производительности;

2) должностные, характеризующие способности и подготовленностью в соответствии ПС

Перечень экзаменационных заданий для практического и теоретического экзаменов составляется экспертами на основе сформированных компетенций, содержащихся в ПС определенного квалификационного уровня либо в иных применяемых законодательных и

регламентирующих требованиях к компетенции персонала [Error: Reference source not found].

Экзаменационные задания могут включать:

1) тесты и экзамены;

Экзамены и тесты являются способом оценки способности кандидата применять умения, знания и понимание в ситуации ограниченного времени и четко заданных условий.

Экзамены могут быть:

- устными, оценивающими способность сделать устное сообщение;

- в форме интервью, когда результаты обучения оцениваются посредством устных вопросов, задаваемых кандидату;

- в форме аудирования (для оценки умения слушать и интерпретировать);

- письменными (в виде тестов) для оценки понимания теоретических знаний и их использования.

Экзамены могут использоваться в дополнение к практическим заданиям по демонстрации компетенций.

2) наблюдение (получение доказательств наличия компетенций у определенного человека при наблюдении за его ежедневной работой). Данный метод заключается в определении доказательств наличия компетенций во время выполнения кандидатом непосредственной трудовой деятельности, за которой наблюдает третье лицо с целью оценки наличия компетенций требуемого уровня.

3) имитационные задания (доказательства, собранные в ситуации, максимально приближенной к трудовой среде). Имитация означает, что компетенции проверяются не в условиях реальной трудовой деятельности, т.е. кандидат помещается в условия, полностью схожие с условиями ежедневной практической работы для оценки его компетенций. Демонстрация выполнения практических заданий на рабочем месте или в

имитационной ситуации направлена на комплексную оценку как технических, так и ключевых компетенций.

Наблюдение за демонстрацией компетенций осуществляется индивидуально. При его организации необходимо обеспечить [Error: Reference source not found]:

- наличие ясных и точных инструкций по выполнению практического задания;

- четкое соответствие задания с компетенциями, подлежащими оценке;

- понимание кандидатом содержания и процедур выполнения задания и критериев оценки;

- необходимые материалы и ресурсы.

Оценка должна быть направлена на интегрированную диагностику теоретических знаний и практических умений.

Выбор методов оценки должен диктоваться содержанием тех трудовых функций, которые подлежат оценке по квалификационному уровню. Если оценивается компетенция к примеру, в области соблюдения правил санитарии и гигиены, оценочные задания должны содержать не пересказ содержания соответствующих регламентирующих, или нормативных документов, а способность использовать требуемые знания на практике.

При формировании оценки практического или теоретического экзаменов, а также экзаменов по оценке поведенческих навыков необходимо исходить из условий, что каждый практический и теоретический экзамен в совокупности составляет 100%.

При этом удельный вес заданий для:

1) практического экзамена составляет 70% от функциональных обязанностей ПС определенного квалификационного уровня.

2) теоретического экзамена и оценки поведенческих навыков составляет 20% от функциональных обязанностей ПС определенного

квалификационного уровня.

Таким образом, в результате проведённого анализа было выявлено, что существующие программы для обучения и оценки знаний студентов не в полной мере соответствуют современным требованиям. По этой причине было принято решение разработать указанное программное обеспечение самостоятельно.

Глава 2. Проектирование концептуальной модели базы данных и логической схемы модели данных

2.1 Разработка требований к программному обеспечению

Основное назначение разрабатываемого программного обеспечения – индивидуальное обучение пользователей любым учебным темам с возможностью контроля знаний. Система также предназначена для сопровождения учебного процесса и для организации самостоятельной работы студентов.

Программное обеспечение должно включать теоретический и практический материал с тестами и вопросами различной сложности. Также должны присутствовать практические задания на ввод ответа.

Весь учебный материал структурирован по темам. Темы снабжены иллюстрациями, примерами, вопросами, тестами, а также видеофрагментами, что позволит пользователю более эффективно познать изучаемый предмет.

Основными задачами, которые необходимо решить в ходе автоматизации являются:

- разграничение прав доступа к системе;
- определение уровня знаний пользователя и постепенное его повышение, путем предоставления необходимого дидактического материала;
- обеспечение самостоятельного изучения и закрепления материала, с помощью интуитивно-понятного интерфейса, удобного для восприятия представления теоретического материала, различного вида тестов, а также с помощью различных заданий, выполнение которых требует написания программного кода;
- хранение данных о пользователе, который изучает или преподает данный предмет;

- хранение истории активности каждого пользователя, т.е. какие темы пользователь изучил, какие темы необходимо повторить, какие тесты прошел и т.п.;
- предоставление возможности преподавателю добавлять новый учебный материал;
- хранение результатов тестирования каждого обучаемого, необходимых для формирования различных отчетов;
- формирование различных видов отчетности.

2.2 Проектирование концептуальной модели базы данных

На основе анализа предметной области и выделения информационных объектов и связей между ними была построена инфологическая модель базы данных. В качестве инструмента моделирования была использована диаграмма сущность-связь (ER - Entity-Relationship).

Цель инфологического моделирования – обеспечение наиболее естественных для человека способов сбора и представления той информации, которую предполагается хранить в создаваемой базе данных. Поэтому инфологическую модель данных пытаются строить по аналогии с естественным языком (последний не может быть использован в чистом виде из-за сложности компьютерной обработки текстов и неоднозначности любого естественного языка).

Сущность инфологического подхода к проектированию информационных систем заключается в установлении соответствия между состоянием предметной области и его представлением в базе данных. Основными конструктивными элементами инфологических моделей являются сущности, связи между ними и их свойства (атрибуты).

Сущность - это класс однотипных объектов, любой различимый

объект (объект, который мы можем отличить от другого), информация о которых должна быть учтена в модели. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе.

Экземпляр сущности - это конкретный представитель данной сущности. Экземпляры сущностей должны быть различимы, т.е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности [7].

Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности. Его наименование должно быть уникальным для конкретного типа сущности, но может быть одинаковым для различного типа сущностей. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Атрибуты используются для определения того, какая информация должна быть собрана о сущности. Абсолютное различие между типами сущностей и атрибутами отсутствует. Атрибут является таковым только в связи с типом сущности. В другом контексте атрибут может выступать как самостоятельная сущность [18].

Ключ сущности - это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность.

Сущность может иметь несколько различных ключей. Каждый экземпляр сущности должен быть уникален. Для этого вводят первичные и вторичные ключи [16].

Первичный ключ (Primary Key) – это атрибут (или группа атрибутов), однозначно идентифицирующий экземпляр сущности. При этом значение идентифицирующего атрибута находится во взаимно-

однозначном соответствии с экземплярами сущности. Первичный ключ должен удовлетворять ряду требований [2]:

- уникальность. Два экземпляра сущности не должны иметь одинаковых значений ключа.
- минимум атрибутов в ключе. Составной первичный ключ (группа атрибутов) не должен содержать ни одного атрибута, удаление которого не приводило бы к утрате уникальности.

Вторичный (или внешний) ключ (Foreign Key) – он появляется в процессе установления связей между сущностями. Это описательный атрибут, который назначается в дополнение к идентифицирующему атрибуту и служит не для идентификации этого уникального экземпляра, а для выделения из набора тех объектов, которые обладают интересующими нас свойствами. Этот ключ часто называют "ключом поиска". Ограничения на состав вторичных ключей определяются логикой запроса.

Различают зависимые и независимые сущности. Тип сущности определяется связью между сущностями [18].

Независимая сущность – обычная сущность, которая не зависит при идентификации от других объектов в модели, или просто независимая, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Зависимая сущность – это дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, или просто сущность, у которой однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности, и

сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации).

Связь - это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою. Связи позволяют по одной сущности находить другие сущности, связанные с нею. Каждая связь может иметь один из следующих типов связи [7]:

- связь «один-к-одному».
- связь «один-ко-многим».
- связь «много-ко-многим».

Связь типа один-к-одному означает, что один экземпляр первой сущности (левой) связан с одним экземпляром второй сущности (правой). Связь один-к-одному чаще всего свидетельствует о том, что на самом деле мы имеем всего одну сущность, неправильно разделенную на две [10].

Связь типа один-ко-многим означает, что один экземпляр первой сущности (левой) связан с несколькими экземплярами второй сущности (правой). Это наиболее часто используемый тип связи. Левая сущность (со стороны "один") называется родительской, правая (со стороны "много") - дочерней [14].

Связь типа много-ко-многим означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Тип связи много-ко-многим является временным типом связи, допустимым на ранних этапах разработки модели. В дальнейшем этот тип связи должен быть заменен двумя связями типа один-ко-многим путем создания промежуточной сущности [1].

Каждая связь может иметь одну из двух модальностей связи: «может» и «должен».

Модальность «может» означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром [11].

Модальность «должен» означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности.

Связь может иметь разную модальность с разных концов. Каждая связь может быть прочитана как слева направо, так и справа налево. [9]

Идентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым «должен», где подлежащим является имя дочерней таблицы. Она используется, когда требуется, чтобы внешний ключ передавался в дочернюю сущность (и, в результате, создавал зависимую сущность). Идентифицирующая связь устанавливается между независимой (родительской – начало связи) и зависимой (дочерней – конец связи) сущностями [6].

Очень важно обратить внимание на то, что при установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Происходит миграция атрибутов. В дочерней сущности новые атрибуты помечаются как внешний ключ (foreign key) [12].

Неидентифицирующая связь – это связь, которая характеризуется глагольной формой со сказуемым «может», где подлежащим является имя дочерней таблицы. Она используется для отображения другого типа передачи атрибутов внешних ключей – передача в область

данных дочерней сущности. Очень важно обратить внимание на то, что при установлении неидентифицирующей связи происходит миграция атрибутов первичного ключа родительской сущности не в состав первичного ключа дочерней сущности, а в состав неключевых атрибутов дочерней сущности [12].

На данном этапе были выявлены все сущности, необходимые для проектирования информационного обеспечения, а также связи между ними. На рисунке 3 наглядно представлена концептуальная модель базы данных ERD, где видно, что используется только сильный тип сущности, а также такие типы отношений как один ко многим (1:M) и многое ко многим (M:N).

Сильный тип сущности – тип сущности, существование которого не зависит какого-то другого типа сущности.

Отношение 1:M – отношения данного типа являются наиболее часто используемыми.

Отношение M:N – отношения данного типа обычно используются на ранних этапах проектирования с целью пояснения ситуации. В дальнейшем каждое из таких отношений должно быть преобразовано в комбинацию отношений типов 1:1 и 1:M (возможно с добавлением вспомогательных сущностей и введением новых отношений).

Описание сущностей и связей между ними представлено в словаре данных модели сущность – связь ERD.

Рисунок 3 – Концептуальная модель базы данных

В таблице 3 приведено описание сущностей.

Таблица 3 – Описание сущностей

Название сущности	**Краткое описание**	**Разновидность**
Пользователи	Содержит список всех пользователей, имеющих право входа в систему: администраторы, преподаватели, студенты.	Сильный
Избранное	Содержит список материалов, которые студент пометил как избранные.	Сильный
Учебный материал	Хранит все учебные материалы, создаваемые администраторами и преподавателями. В качестве текста материала используется текст в формате HTML.	Сильный
Видео	Содержит список всех видеофайлов, включенных в обучающую информационную систему. Видеофайлы хранятся в отдельном каталоге, а таблица содержит ссылки на них.	Сильный

Тест	Хранит список всех тестов вместе с их уровнями сложности.	Сильный

Продолжение таблицы 3

Название сущности	**Краткое описание**	**Разновидность**
Тестовый вопрос	В данной сущности хранятся все тестовые вопросы и практические задания. Следует отметить, что для упрощения решения задачи тестовые вопросы и практические задания хранятся единообразно, для их различия используется поле «Тип».	Сильный
Ответ	Ответ на тестовый вопрос или практическое задание.	Сильный
Результат теста	Хранит описание результата прохождения сеанса тестирования пользователем	Сильный
Статистика	Независимая сущность, хранящая описание всех действий. Независимость данной сущности обусловлена необходимостью хранить данные вне зависимости от изменения их в других таблицах. Например, при удалении пользователя из системы его статистика в любом случае сохраняется.	Сильный

В таблице 4 приведено описание типов связей между информационными сущностями.

Таблица 4 – Описание типов связей

Имя типа связи	**Краткое описание**	**Степень связи**	**Список типов сущностей**	
Включает	Один пользователь может встречаться в нескольких записях таблицы «Результат теста», однако одна запись о результате тестирования может содержать данные только об одном пользователе.	Бинарная	Пользователь	1
			Результат теста	M
Имеет	Один пользователь может встречаться в нескольких	Бинарная	Пользователь	1

	записях таблицы «избранное», однако одна запись об избранном материале может содержать данные только об одном пользователе.			
			Избранное	M

Продолжение таблицы 4

Имя типа связи	**Краткое описание**	**Степень связи**	**Список типов сущностей**	
Включает	Один учебный материал может быть в избранном у нескольких пользователей, однако каждая запись о занесении учебного материала в избранное может содержать данные только об одном материале	Бинарная	Учебный материал	1
			Избранное	M
Содержит	Один учебный материал может содержать ссылки на различные видео, а одно видео может быть включено в разные учебные материалы	Бинарная	Учебный материал	N
			Видео	M
Содержит	Один тестовый вопрос может содержать ссылки на различные видео, а одно видео может быть включено в разные тестовые вопросы	Бинарная	Тестовый вопрос	N
			Видео	M
Содержит	Один тест может содержать несколько тестовых вопросов, однако один тестовый вопрос может принадлежать только одному тесту.	Бинарная	Тест	1
			Тестовый вопрос	M
Содержит	Один тестовый вопрос может иметь несколько различных ответов, однако каждый ответ относится только к одному вопросу.	Бинарная	Тестовый вопрос	1
			Ответ	M
Включает	Один тест может содержать	Бинарная	Тест	1

	несколько записей о результатах тестирования, однако каждый результат тестирования относится только к одному тесту.		Результат теста	M

Рассмотрим описание атрибутов определенных выше информационных сущностей. Описание атрибутов сущности «Пользователь» приведено в таблице 5.

Таблица 5 – Описание сущности «Пользователь»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Логин	Текстовый	Простой	Логин пользователя
Пароль	Текстовый	Простой	Пароль пользователя
Имя	Текстовый	Простой	ФИО пользователя
Тип	Числовой	Простой	Код типа пользователя
Группа	Текстовый	Простой	Группа пользователя (студенческая группа, может быть пустым)

Описание атрибутов сущности «Учебный материал» приведено в таблице 6.

Таблица 6 – Описание сущности «Учебный материал»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Название	Текстовый	Простой	Название учебного материала
Текст	Текстовый	Простой	Текст учебного материала в формате HTML
Файл	Текстовый	Простой	Ссылка на файл с учебным материалом
Сложность	Числовой	Простой	Уровень сложности учебного материала

В таблице 7 приведено описание атрибутов сущности «Видео».

Таблица 7 – Описание сущности «Видео»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Название	Текстовый	Простой	Название видео
Имя файла	Текстовый	Простой	Ссылка на файл с видео

В таблице 8 приведено описание атрибутов сущности «Тест».

Таблица 8 – Описание сущности «Тест»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Название	Текстовый	Простой	Название теста
Сложность	Числовой	Простой	Уровень сложности теста

В таблице 9 приведено описание атрибутов сущности «Ответ».

Таблица 9 – Описание сущности «Ответ»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Текст	Текстовый	Простой	Текст ответа на вопрос
Верный	Числовой	Простой	Признак правильности ответа

В таблице 10 приведено описание атрибутов сущности «Тестовый вопрос».

Таблица 10 – Описание сущности «Тестовый вопрос»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Текст	Текстовый	Простой	Текст вопроса в формате HTML
Тип	Числовой	Простой	Тип вопроса (тестовый или практическое задание)
Название	Текстовый	Простой	Название вопроса в формате обычного текста

В таблице 11 приведено описание атрибутов сущности

«Статистика».

Таблица 11 – Описание сущности «Статистика»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Имя	Текстовый	Простой	Имя пользователя, совершившего действие
Дата	Дата	Простой	Дата совершения действия

Продолжение таблицы 11

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Название	Текстовый	Простой	Название материала, теста или вопроса, с которым было совершено действие
Действие	Текстовый	Простой	Наименование действия
Тип	Текстовый	Простой	Тип объекта, с которым было совершено действие

В таблице 12 приведено описание атрибутов сущности «Результат теста».

Таблица 12 – Описание сущности «Результат теста»

Имя атрибута	**Домен**	**Тип по составу**	**Описание**
Всего	Числовой	Простой	Общее количество вопросов в тесте
Верных	Числовой	Простой	Количество верных ответов на тестовые вопросы
Пропущено	Числовой	Простой	Количество пропущенных тестовых вопросов или практических заданий в сеансе тестирования
Дата	Дата	Простой	Дата тестирования

Описание доменов приведено в таблице 13.

Таблица 13 – Описание доменов

Наименование	**Описание**	**Диапазон значений**

Текстовый	Строка, состоящая из символов	[А-Я, а-я]
Дата	Формат дата ДД.ММ.ГГГГ	ДД - [01…31] ММ - [01…12] ГГГГ - [1900…+∞]
Числовой	Целое число	-2147483648…2147483648

2.3 Проектирование логической схемы модели данных

Все информационные ресурсы исследованы и решение о внедрении реляционной БД, как части корпоративной информационной системы было принято, поэтому была построена логическая структура БД с использованием метода IDEF1X.

На рисунке 4 наглядно представлена логическая модель данных, на которой представлено преобразование связей «многое ко многим» с добавлением вспомогательных сущностей.

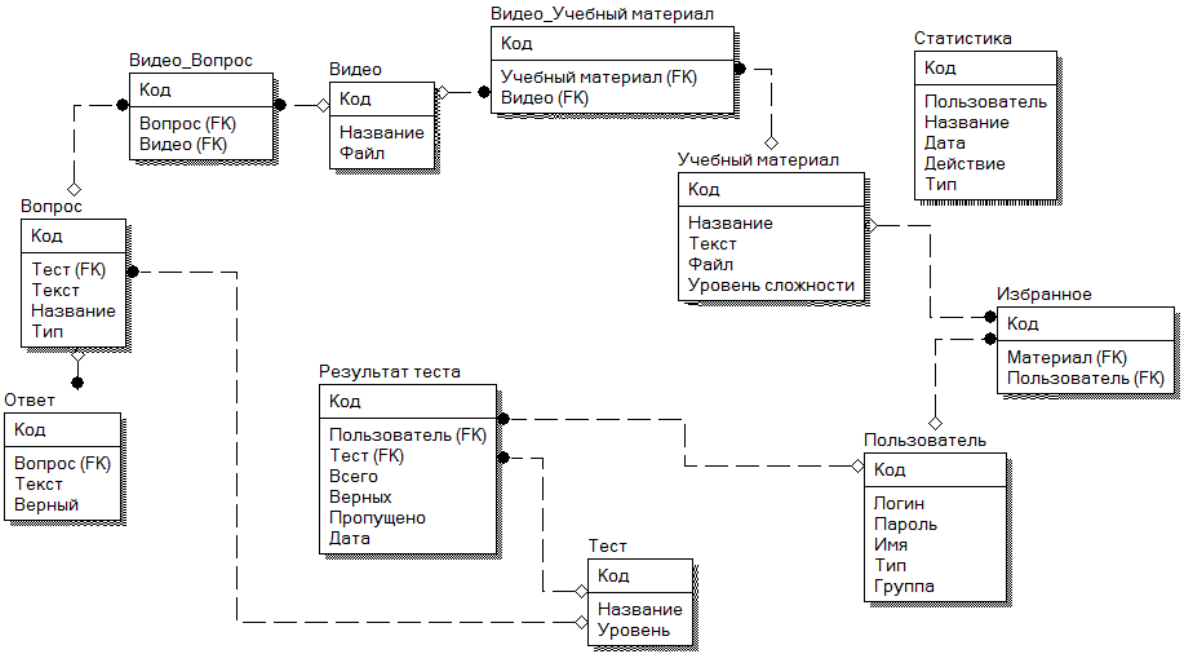


Рисунок 4 – Логическая модель базы данных

В таблицах 14 – 24 приведён словарь данных логической модели БД.

Таблица 14 – Структура таблицы «Вопрос»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код вопроса
-	Текст	Текстовый	NOT NULL	Текст вопроса
-	Тип вопроса	Числовой	NOT NULL	Тип вопроса (тестовый или с выполнением программного кода)
-	Название	Текстовый	NOT NULL	Заголовок вопроса
FK	Тест	Числовой	NOT NULL	Код теста, которому принадлежит вопроса

Таблица 15 – Структура таблицы «Ответ»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Уникальный идентификатор записи
FK	Вопрос	Числовой	NOT NULL	Код вопроса
-	Текст	Текстовый	NOT NULL	Текст вопроса
-	Верный	Числовой	NOT NULL	Верный/Неверный ответ

Таблица 16 – Структура таблицы «Видео»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код видео

-	Название	Текстовый	NOT NULL	Наименование видео
-	Файл	Текстовый	NOT NULL	Имя видео в папке, в которую сохраняется весь дидактический материал

Таблица 17 – Структура таблицы «Видео_Вопрос»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Уникальный идентификатор записи
FK	Вопрос	Числовой	NOT NULL	Код вопроса
FK	Видео	Числовой	NOT NULL	Код видео

Таблица 18 – Структура таблицы «Видео_Учебный материал»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Уникальный идентификатор записи
FK	Учебный материал	Числовой	NOT NULL	Код материала
FK	Видео	Числовой	NOT NULL	Код видео

Таблица 19 – Структура таблицы «Учебный материал»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код учебного материала

-	Название	Текстовый	NOT NULL	Наименование материала
-	Текст	Текстовый	NOT NULL	Текст учебного материала
-	Файл	Текстовый	NULL	Ссылка на файл с учебным материалом

Таблица 20 – Структура таблицы «Избранное»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Уникальный идентификатор записи
FK	Материал	Числовой	NOT NULL	Код материала
FK	Пользователь	Числовой	NOT NULL	Код пользователя

Таблица 21 – Структура таблицы «Пользователь»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код пользователя

Продолжение таблицы 21

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
-	Логин	Текстовый	NOT NULL	Логин пользователя
-	Пароль	Текстовый	NOT NULL	Пароль пользователя
-	Имя	Текстовый	NOT NULL	ФИО пользователя
-	Тип	Числовой	NOT NULL	Тип пользователя

-	Группа	Текстовый	NULL	Название группы

Таблица 22 – Структура таблицы «Статистика»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код записи в таблице «Статистика»
-	Наименование	Текстовый	NOT NULL	Наименование материала, над которым производили какое-либо действие
-	Дата	Дата	NOT NULL	Дата, когда было зафиксировано действие пользователя
-	Действие	Текстовый	NOT NULL	Наименование действия
-	Тип	Текстовый	NOT NULL	Тип материала, над которым производили какое-либо действие
-	Пользователь	Текстовый	NOT NULL	Имя пользователя

Таблица 23 – Структура таблицы «Тест»

Поля				
Ключ	**Наименование**	**Тип данных**	**Значение**	**Описание**
PK	Код	Числовой	NOT NULL	Код теста
-	Наименование	Текстовый	NOT NULL	Наименование теста
-	Сложность	Числовой	NOT NULL	Номер уровня сложности

Таблица 24 – Структура таблицы «Результат теста»

Поля

Ключ	Наименование	Тип данных	Значение	Описание
PK	Код	Числовой	NOT NULL	Код результата
FK	Пользователь	Числовой	NOT NULL	Код пользователя
FK	Тест	Числовой	NOT NULL	Код теста
-	Всего	Числовой	NOT NULL	Всего вопросов в сеансе тестирования
-	Верных	Числовой	NOT NULL	Верных ответов на вопросы в сеансе тестирования
-	Пропущено	Числовой	NOT NULL	Пропущенных вопросов в сеансе тестирования
-	Дата	Дата	NOT NULL	Дата проведения тестирования

В результате произведённого анализа предметной области были определены требования к разрабатываемой информационной системе. Кроме того, определены информационные сущности, разработаны отношения между ними. Сущности объединены в логическую схему базы данных, подготовленную для переноса в СУБД.

Глава3. Разработка программных модулей и анализ их тестирования

3.1 Обоснование выбора языка программирования и средств разработки

При выборе метода проектирования был выбрано структурное проектирование.

Обоснуем выбор:

1) Необходимо чёткое представление ИС.

2) Так же нам требуется декомпозиция системы, что позволит нам более быстро находить ошибки.

3) Иерархическое представление ИС.

Поскольку в настоящем дипломном проекте рассматривается разработка системы, которая будет работать в учебном учреждении, то, с точки зрения языка программирования, такая система должна обладать следующими свойствами:

- Возможность одинаковой работы приложения на разных версиях операционной системы Windows. Известно, что Windows API содержит как документированные, так и особые, недокументированные функции, использование которых может быть допустимо в рамках только одной версии ОС. Некоторые компиляторы по неизвестной причине добавляют в код программ вызов таких функций, тем самым накладывая значительное ограничение на количество операционных систем, под которыми скомпилированный код может быть исполнен. Следует по возможности избегать работы с подобными компиляторами [11].
- Выходная программа должна обладать высоким быстродействием. Поскольку предполагается, что программа

будет выполнять запросы на обработку значительных объёмов данных, а в требованиях к программе указаны временные ограничения на создание отчетов, данный пункт является обязательным [19].

- Возможности сетевой работы. Не все версии языков программирования поддерживают работу по сети должным образом. Поскольку все компьютеры в рамках одного учебного заведения, как правило, работают с одной БД, она находится на сервере, расположенном в сети. Поэтому реальному приложению обязательно нужна возможность подключения к сети и передачи данных по ней [13].

Итак, перечислены основные требования к языку. Из наиболее популярных удовлетворяют данным требованиям языки платформы .NET . Следует привести доводы, чтобы выбрать один из них.

NET Framework – это среда, в которой могут выполняться программно-независимые приложения. В данной среде поддерживается совместная работа в приложении разных языков программирования. NET Framework определяет библиотеки классов. Любая примитивная программа на NET платформе основана на использовании библиотек классов NET Framework.

NET Framework содержит в себе не зависящую от языка среду выполнения. Программы не компилируются в бинарный исполняемый файл, а компилируются в промежуточный язык (MS Int ermediat e Language, MSIL). MSIL можно назвать переносным ассемблером, но это псевдо-ассемблер. Среда переводит MSIL в машинный исполняемый код, который в последствии и выполняется. То есть, NET Framework содержит интерпретатор MSIL. Интерпретируемость – это идеальное решение проблем переносимости программ. В это же

время не возникает никаких ошибок при переносе

Visual C# .NET является следующим поколением языка Visual C#, спроектированным для быстрого и легкого создания приложений .NET .

Язык программирования Visual C# .NET получил широкое распространение и признание из-за своей простоты и наглядности. Он не только является мощным инструментом разработки современных приложений, но и идеальным средством обучения программированию. Он позволяет сочетать визуальные методы разработки графических приложений с непосредственным программированием объектов. А удобные и наглядные средства отладки ускорят поиск и разбор характерных ошибок.

Visual C# .NET обладает многими новыми и улучшенными характеристиками, такими как наследование, интерфейсы и перезагрузка, что превращает его в мощный объектно-ориентированный язык программирования. Другие новые возможности языка — обработка свободных потоков и структурированных исключений. Visual C# .NET полностью совместим с .NET Framework и общеязыковой средой выполнения, которые вместе обеспечивают возможность взаимодействия на разных языках, сборку мусора и усиленную безопасность.

Благодаря полной поддержке объектных технологий и наличию библиотек объектов в Visual C# .NET можно создавать приложения корпоративного уровня. Средства языка обеспечивают работу механизмов наследования, инкапсуляции и полиморфизма.

Итак, выбран язык Visual C# .NET .

В качестве среды разработки выберем Microsoft Visual Studio.

На серверах учебных заведений стоят различные СУБД, поэтому программа должна быть спроектирована так, чтобы после

небольших модификаций иметь возможность подключаться к любой из них. Это может быть Oracle, Microsoft SQL Server, Postgre SQL или иная СУБД.

Однако в рамках дипломного проекта необходимо спроектировать и создать рабочий прототип программы, с помощью которого можно было бы продемонстрировать работоспособность всех модулей программы и который не требовал бы при этом создания сервера и установки «тяжелой» БД на нём.

Поэтому было принято решение разработку вести с использованием драйвера Microsoft JET 4.0 в качестве драйвера провайдера базы данных. Этот драйвер установлен на всех операционных системах Windows класса NT с версией ядра 5 и старше (Windows 2000 и старше), поэтому программа, разработанная с его использованием, работоспособна на любом компьютере без необходимости установки дополнительного ПО. Кроме того, этот драйвер обеспечивает всю необходимую функциональность – хранение данных, SQL-запросы, синхронизация между различными приложениями.

Данный драйвер полностью поддерживается приложением Microsoft Access, поэтому база данных для работы приложения будет спроектирована именно в Access.

С учётом того факта, что база данных будет располагаться на сервере БД, перечислим требования к программному обеспечению компьютеров, на которых будет выполняться разрабатываемая программа:

- ОС не ниже Windows XP;

Наличие драйвера Microsoft Jet 4.0.

3.2 Описание программных модулей

Программное обеспечение представляет собой многоподульный проект – Study, рассмотрим его состав:

- Program – главный модуль приложения, предназначен для запуска программы, её инициализации и вывода на экран форм приложения;
- RegForm – модуль, выводящий на экран форму регистрации нового пользователя;
- Auth – модуль, отвечающий за аутентификацию пользователя;
- FormAddMaterial – модуль, выводящий на экран форму добавления нового учебного материала в базу данных информационной системы;
- FormAddTest – модуль, выводящий на экран форму добавления нового теста в базу данных информационной системы;
- FormAddTestAns – модуль, выводящий на экран форму добавления нового ответа на тестовый вопрос в базу данных информационной системы;
- FormEditMaterial – модуль, выводящий на экран форму редактирования учебного материала в базе данных информационной системы;
- FormEditTest – модуль, выводящий на экран форму редактирования теста в базе данных информационной системы;
- FormEditQTest – модуль, выводящий на экран форму для добавления и редактирования тестового вопроса;
- FormInputBox – модуль, выводящий на экран форму для ввода произвольного текстового значения;
- FormNewVideo – модуль, выводящий на экран форму для

добавления нового видеофайла;

- FormStudent – модуль, выводящий на экран интерфейс студента;
- AdminForm – модуль, выводящий на экран формы администратора и преподавателя;
- FormTestQ – модуль, выводящий на экран форму ответа студентом на тестовый вопрос;
- FormTestResult – модуль, выводящий на экран форму с результатами тестирования;
- FormVideoList – модуль, предназначенный для редактирования списка видео;
- FormViewContent – модуль, предназначенный для просмотра учебного материала студентом;
- StaticData – модуль, содержащий статический класс с различными функциями для работы с данными.

Рассмотрим подробнее модуль StaticData. Он содержит одноимённый класс, который реализует следующий богатый набор методов:

- public static void CreateRep(string Rep) – Создание отчёта об успеваемости;
- public static void Init() – инициализаций класса, создание подключения к базе данных;
- public static void ExecQ(string q) – выполнение произвольного текстового запроса к базе данных;
- public static string GetToday() – получить в текстовом виде строку с текущей датой;
- public static void SError(string Error) – отобразить окно с сообщением об ошибке;
- public static void SInfo(string Info) – отобразить окно с

информационным сообщением;

- public static bool Authorize(string Name, string Pass) – авторизовать пользователя;
- public static bool RegUser(string Login, string Name, string Pass, int Type, string Grp) – зарегистрировать нового пользователя в системе;
- public static void AddChange(string Name, string Do, string mType) – добавить информацию в таблицу «Статистика»
- public static void ShowUserWorkWindow() – показать на экране окно в зависимости от прав текущего пользователя;
- public static bool IsMaterialInFavs(string MId) – проверка, что материал находится в избранном у текущего пользователя;
- public static void AddMaterialToFavs(string MId) – добавление материала в избранное текущему пользователю;
- public static string GetTestIdByName(string TestName) – получить код теста по его названию;
- public static string GetQIdByName(string Name, string TestName) – получить код вопроса по его названию и названию теста;
- public static List<ListItem> GetAnsList(string QId) – получить список ответов на вопрос теста;
- public static string GetLastId(string TableName) – получить уникальный идентификатор последней добавленной в таблицу записи;
- public static List<ListItem> GetVideoList() – получить список видеофайлов;
- public static List<ListItem> GetTestList() – получить список тестов;

- public static void PlayVideo(string Id) – воспроизвести видеофайл;
- public static string InputBox(string Caption, string Default = "") – показать окно ввода произвольного текстового значения;
- public static bool TestExists(string Name) – проверка на существование теста;
- public static void StartTestingSession(string Id) – запуск сеанса тестирования.

3.3 Тестирование программного обеспечения

При запуске приложения на экране появляется окно авторизации пользователя (рисунок 5).

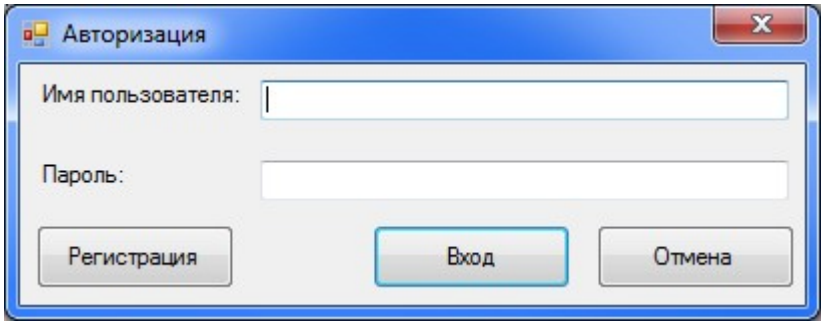


Рисунок 5 – Окно авторизации

При нажатии на кнопку «Регистрация» появляется окно регистрации, с помощью которого можно зарегистрировать в системе нового пользователя с правами доступа студента или преподавателя (рисунок 6).

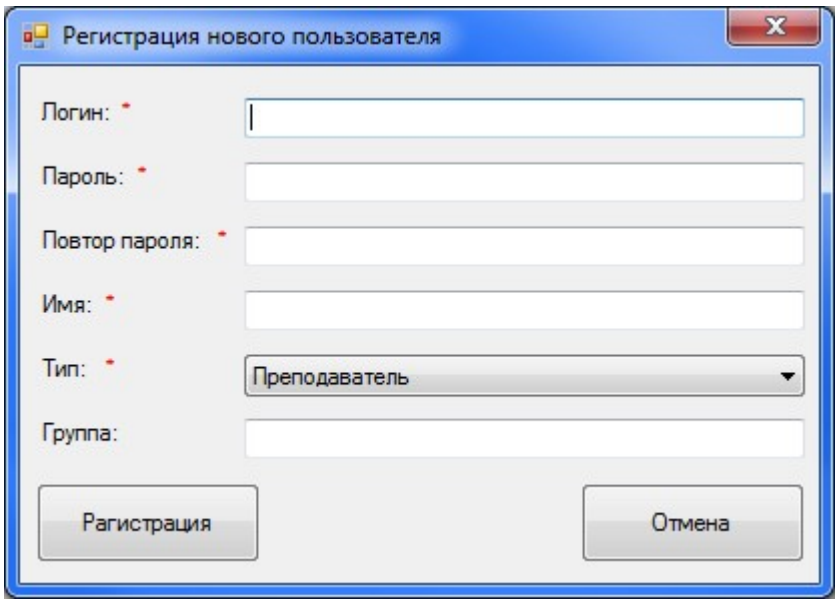


Рисунок 6 – Форма регистрации пользователя

Осуществим вход под аккаунтом администратора. Форма администратора приведена на рисунке 7.

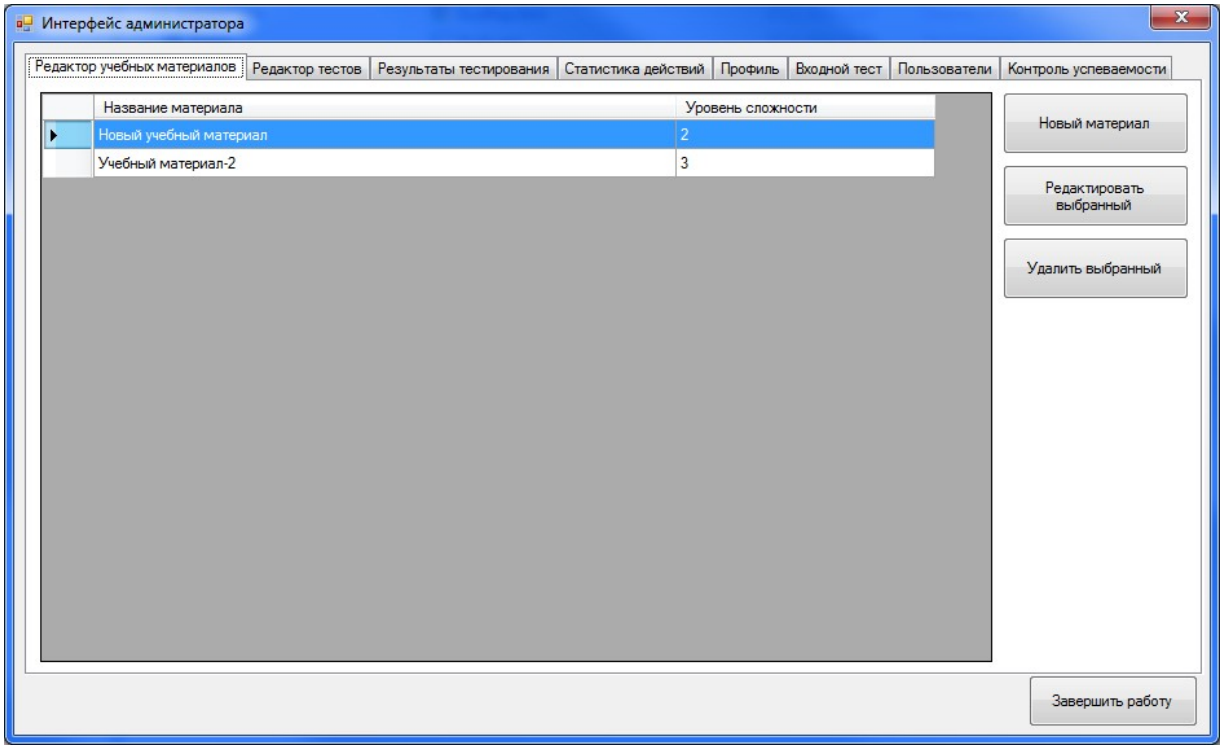


Рисунок 7 – Интерфейс администратора

Интерфейс администратора имеет следующие вкладки:

– редактор учебных материалов;

– редактор тестов;

– результаты тестирования;

– статистика действий;

– профиль;

– входной тест;

– пользователи;

– контроль успеваемости.

При запуске формы открывается вкладка «Редактор учебных материалов», изображённая на рисунке 7.

Для создания учебного материала используется окно, вид которого приведён на рисунке 8. Модуль изменения учебного материала приведён на рисунке 9.

В обоих модулях используется элемент управления для редактирования документов в формате HTML. Для удобства форматирования текста пользователю предложено как меню основных настроек, так и контекстное меню.

С помощью меню основных настроек пользователь может:

- вырезать фрагмент текста;
- вставить фрагмент текста;
- скопировать фрагмент текста;
- делать откат настроек;
- изменять шрифты;
- изменять выравнивание текста;
- создавать списки;
- вставлять различные элементы (разделяющие линии, таблицы, картинки, ссылки);
- осуществлять поиск по тексту;
- выводить данный теоретический материал на печать.

С помощью контекстного меню пользователь может проводить аналогичные манипуляции с текстом только уже расширенные.

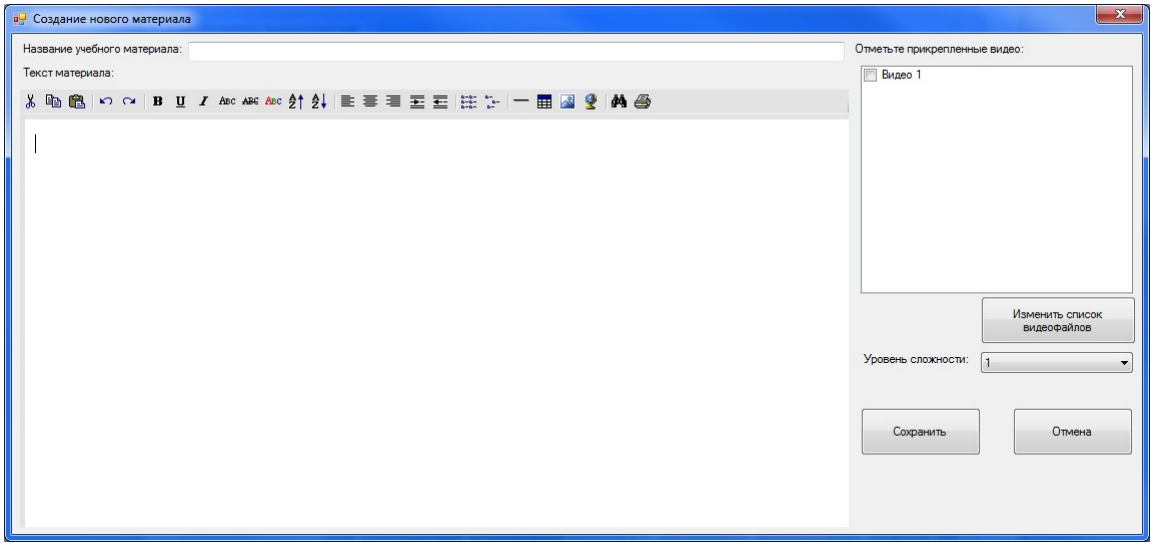


Рисунок 8 – Модуль для создания нового учебного материала

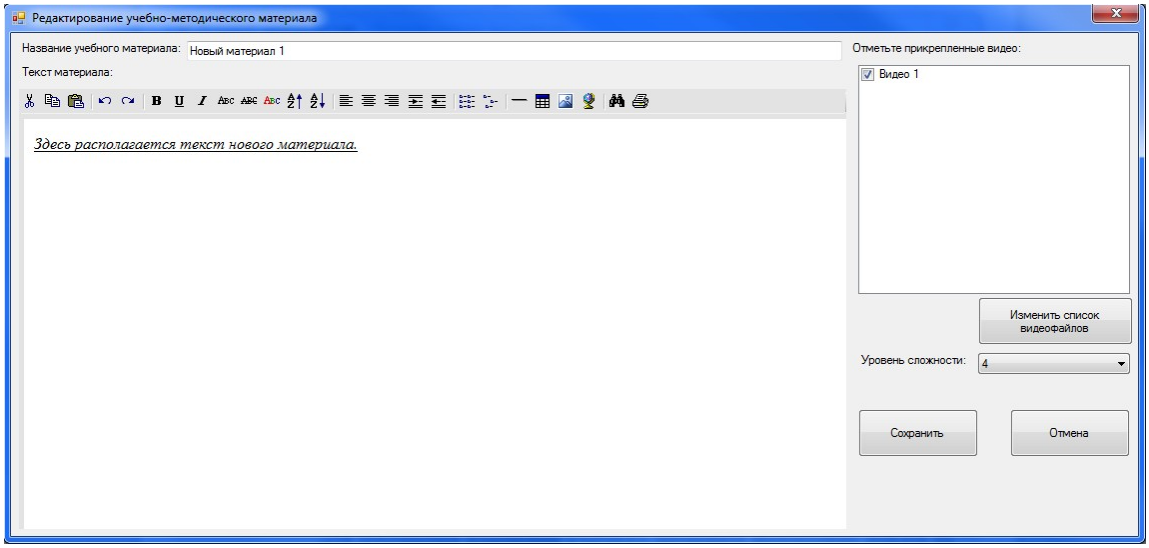


Рисунок 9 – Модуль для редактирования ранее созданного учебного материала

В обоих описанных интерфейсах присутствует кнопка «Изменить список видеофайлов». При нажатии на неё появляется окно редактирования видеофайлов (рисунок 10).

При нажатии на кнопку «Добавить» появляется окно добавления нового видеофайла (рисунок 11). При нажатии на кнопку «Изменить

название» появляется окно, в котором можно ввести новое название видеоролика (рисунок 12).

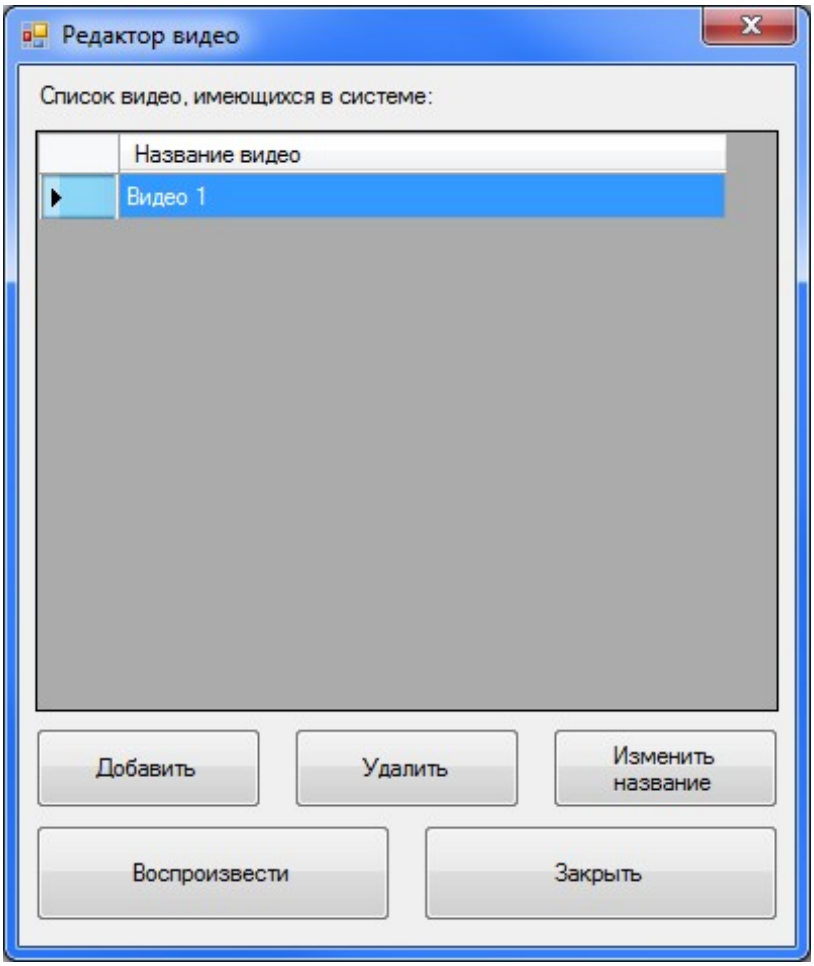


Рисунок 10 – Редактирование видеофайлов

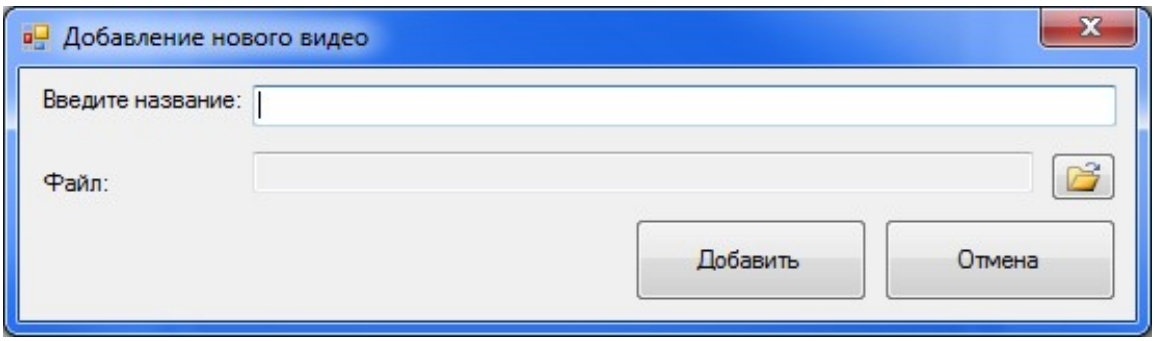


Рисунок 11 – Добавление нового видео

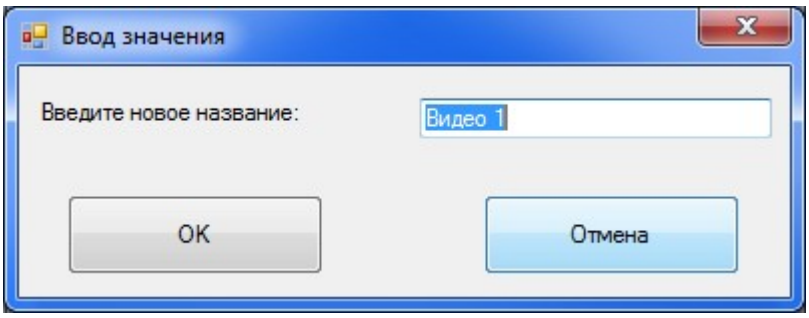


Рисунок 12 – Изменение название видео

На рисунке 13 приведено окно редактора тестов.

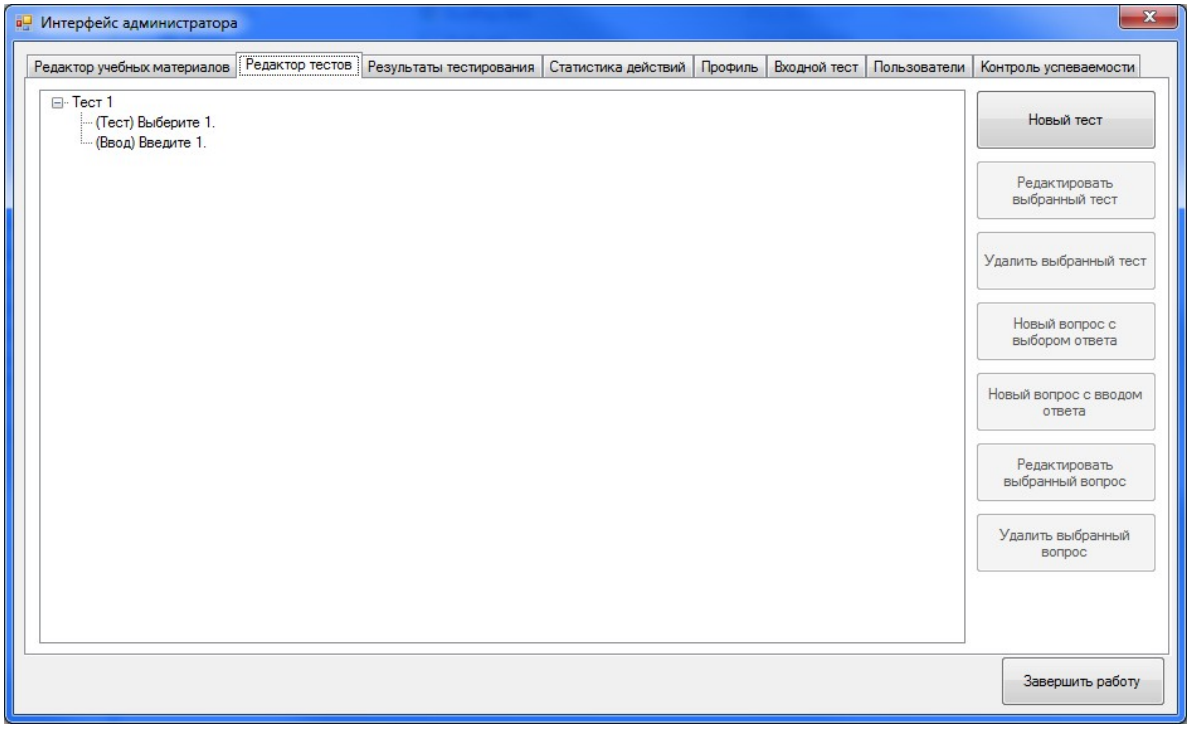


Рисунок 13 – Вкладка «Редактор тестов»

Кнопки в правой колонке этой вкладки активируются в зависимости от выбранного значения в дереве тестов. Если не выбрано ничего, то доступно только создание нового теста (рисунок 14). Если выбран тест, то доступно его редактирование (рисунок 15), создание нового вопроса с выбором ответа (рисунок 16) и создание нового вопроса с вводом ответа (рисунок 17).

Если выбран вопрос, то доступна возможность его редактирования. При редактировании автоматически открывается

редактор вопроса в зависимости от того, какой это был вопрос – с выбором ответа или с вводом ответа.

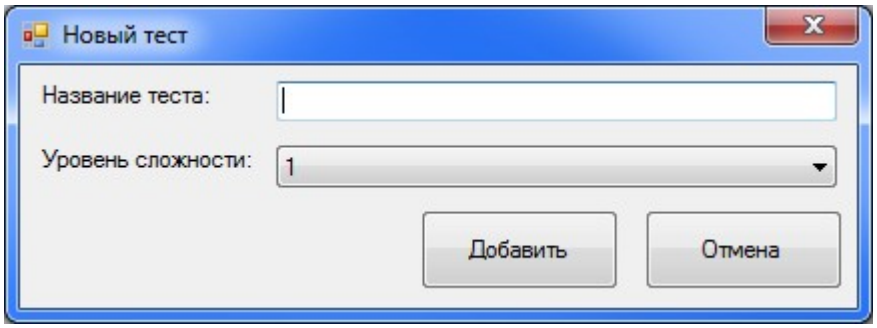


Рисунок 14 – Окно создания нового теста

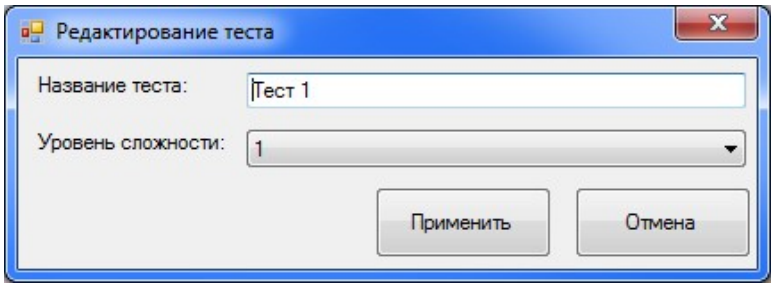


Рисунок 15– Окно редактирования ранее созданного теста

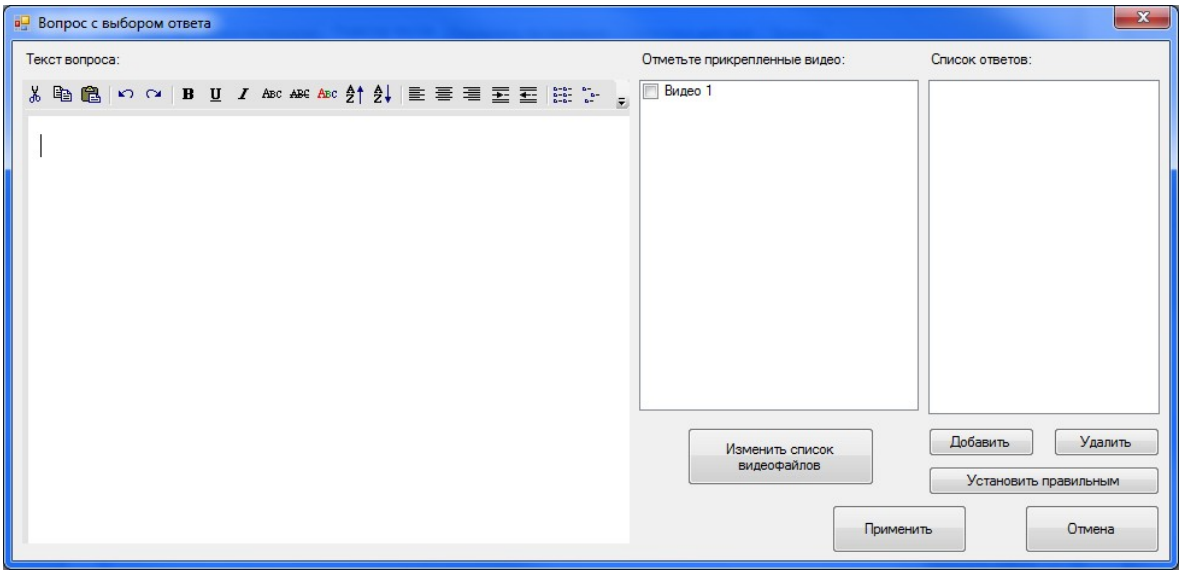


Рисунок 16 – Создание вопроса с выбором ответа

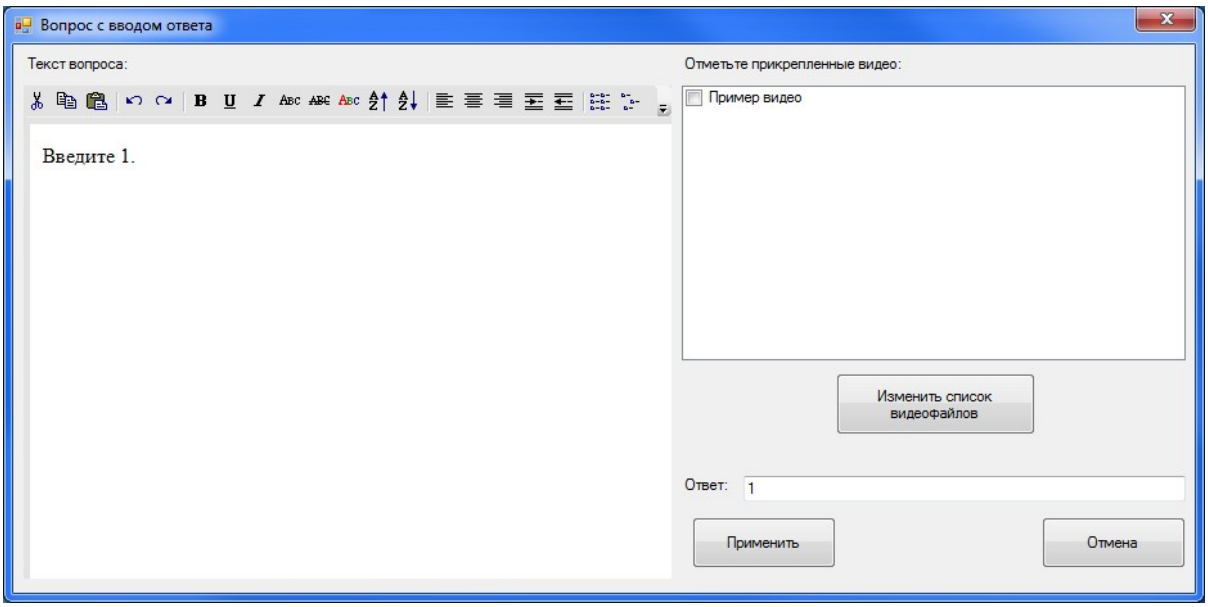


Рисунок 17 – Создание вопроса с вводом ответа

Вкладка «Результаты тестирования» позволяет администратору просматривать результаты тестирования студентов с выводом их на печать и возможностью фильтрации (рисунок 18).

Вкладка «Статистика действий» позволяет просматривать полную статистику: какие пользователи совершали действия с базой данных (рисунок 19).

Вкладка «Профиль» позволяет уточнить данные о текущем пользователе и сменить его при необходимости (рисунок 20).

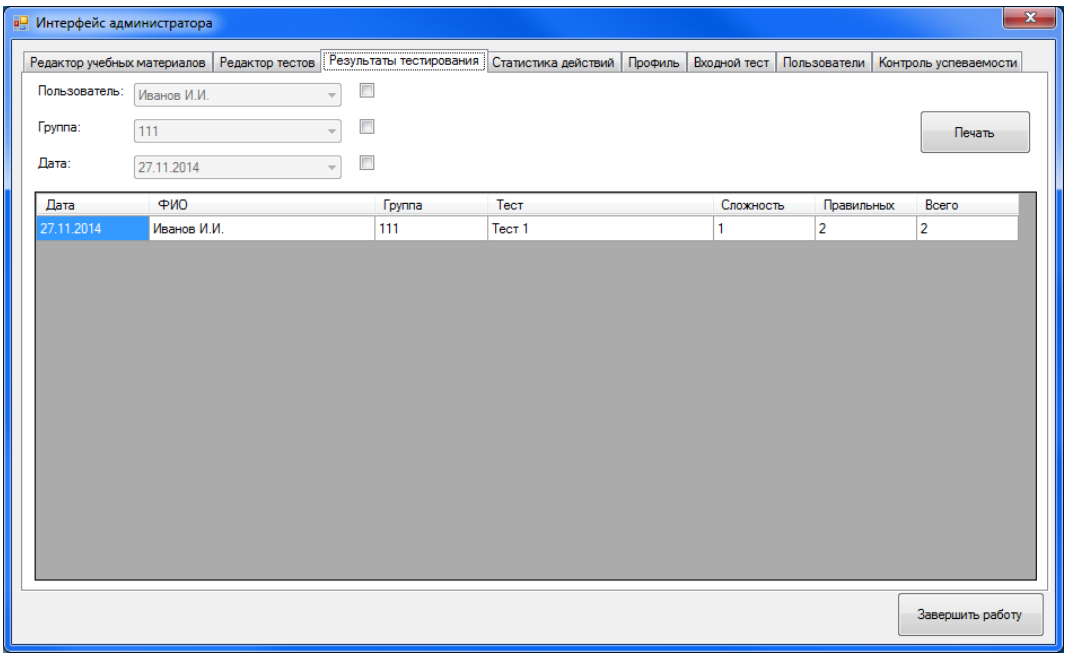


Рисунок 18 – Вкладка «Результаты тестирования»

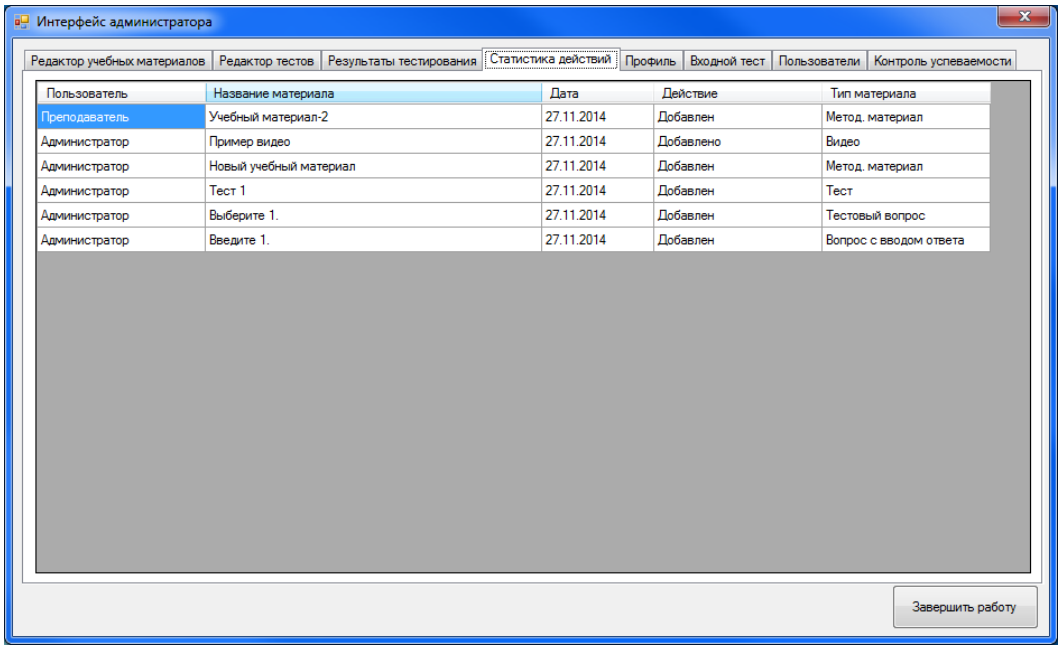


Рисунок 19 – Вкладка «Статистика действий»

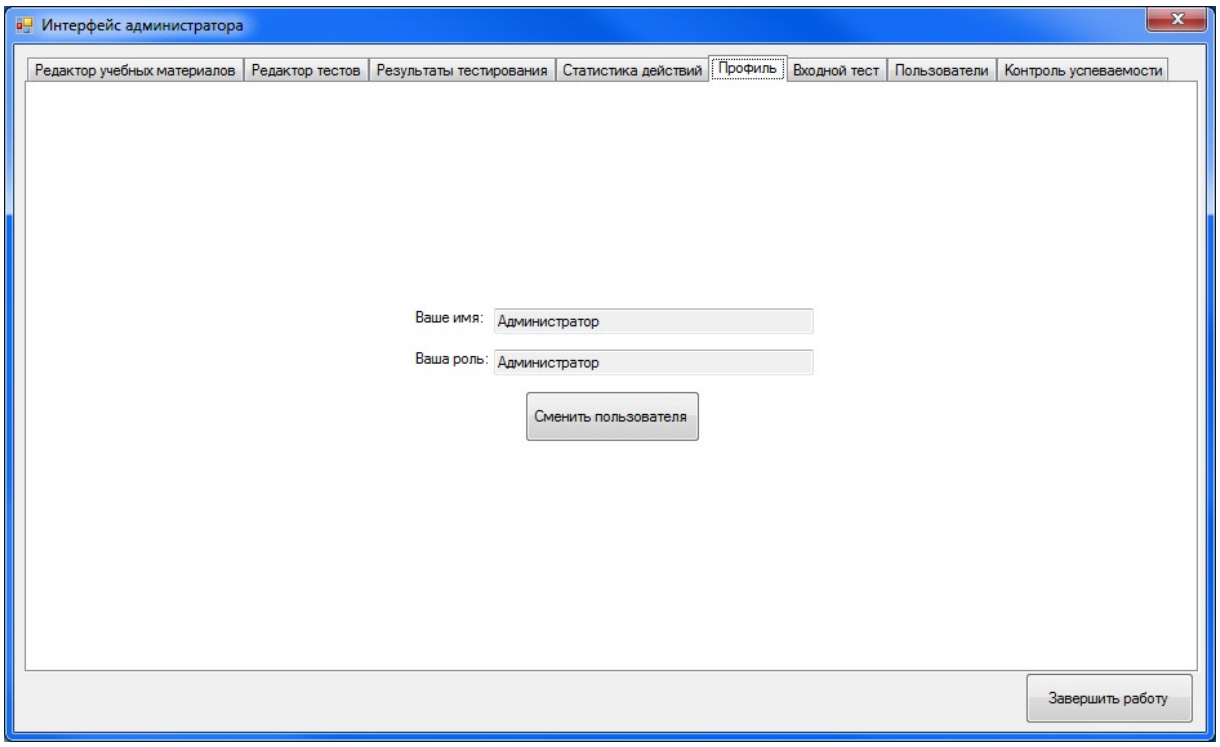


Рисунок 20 – Вкладка «Профиль»

Вкладка «Входной тест» позволяет установить, какой из тестов будет входным при регистрации нового студента, чтобы определить его степень подготовки (рисунок 21).

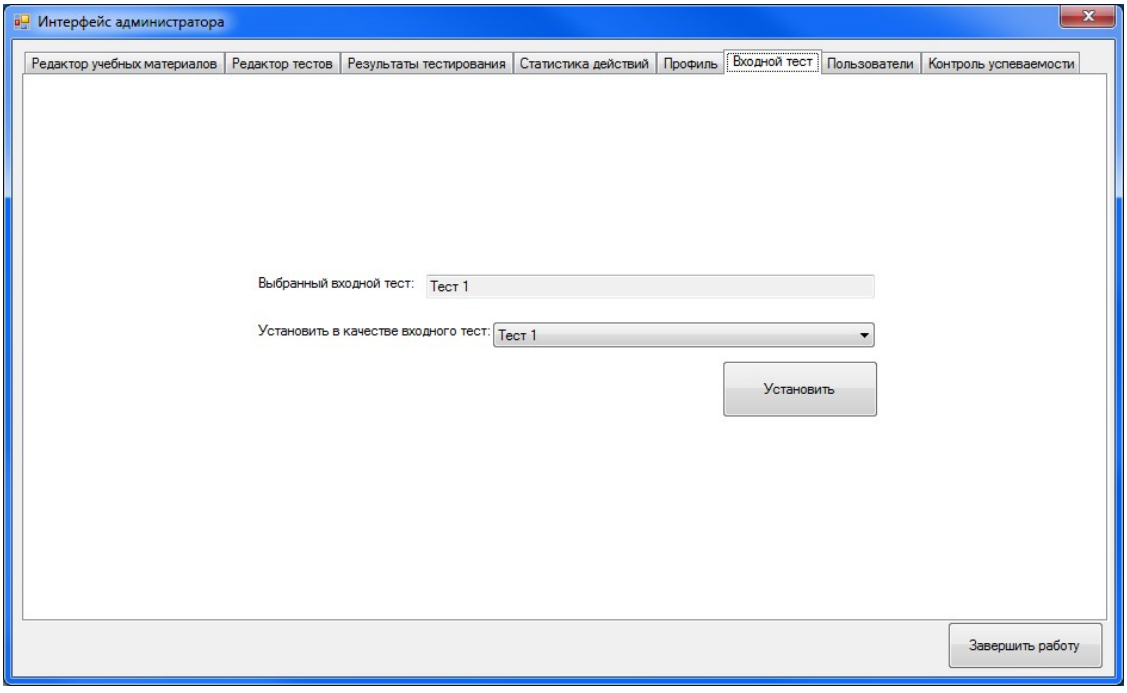


Рисунок 21 – Вкладка «Входной тест»

Вкладка «Пользователи» позволяет подтвердить регистрацию новых преподавателей (рисунок 22), а вкладка «Контроль успеваемости» позволяет выводить на экран отчёт об успеваемости студента и его прогрессе (рисунок 23).

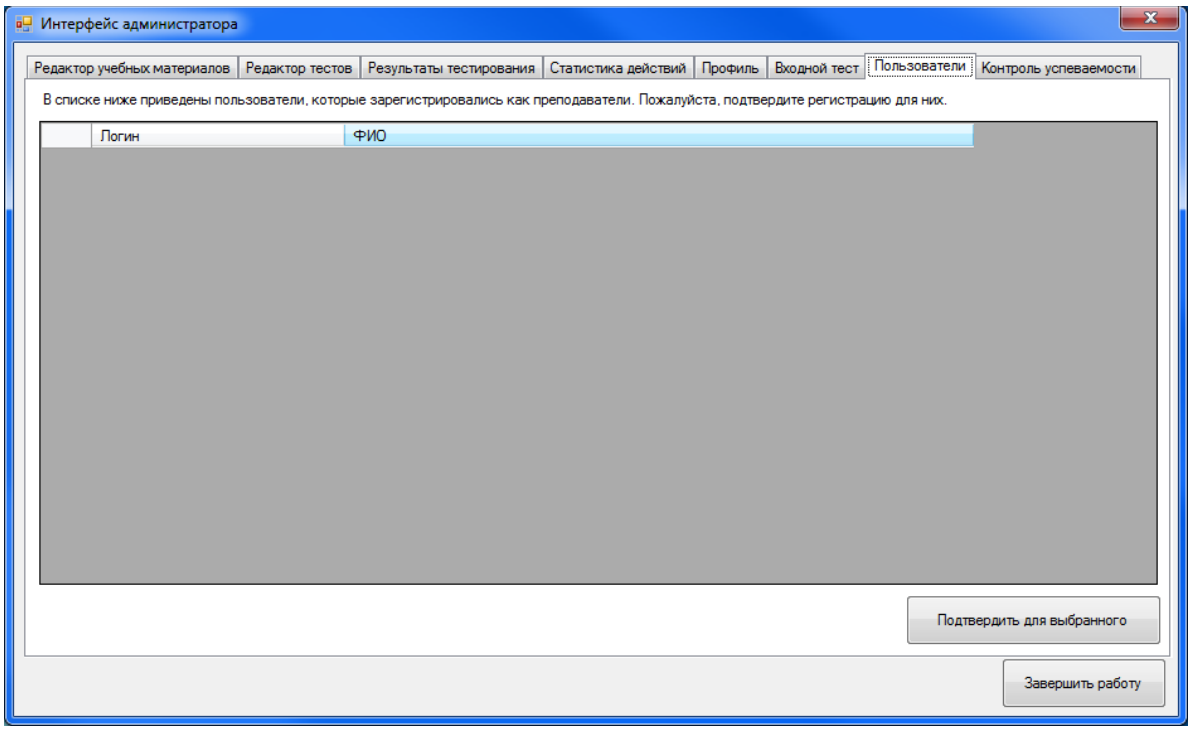


Рисунок 22 – Вкладка «Пользователи»

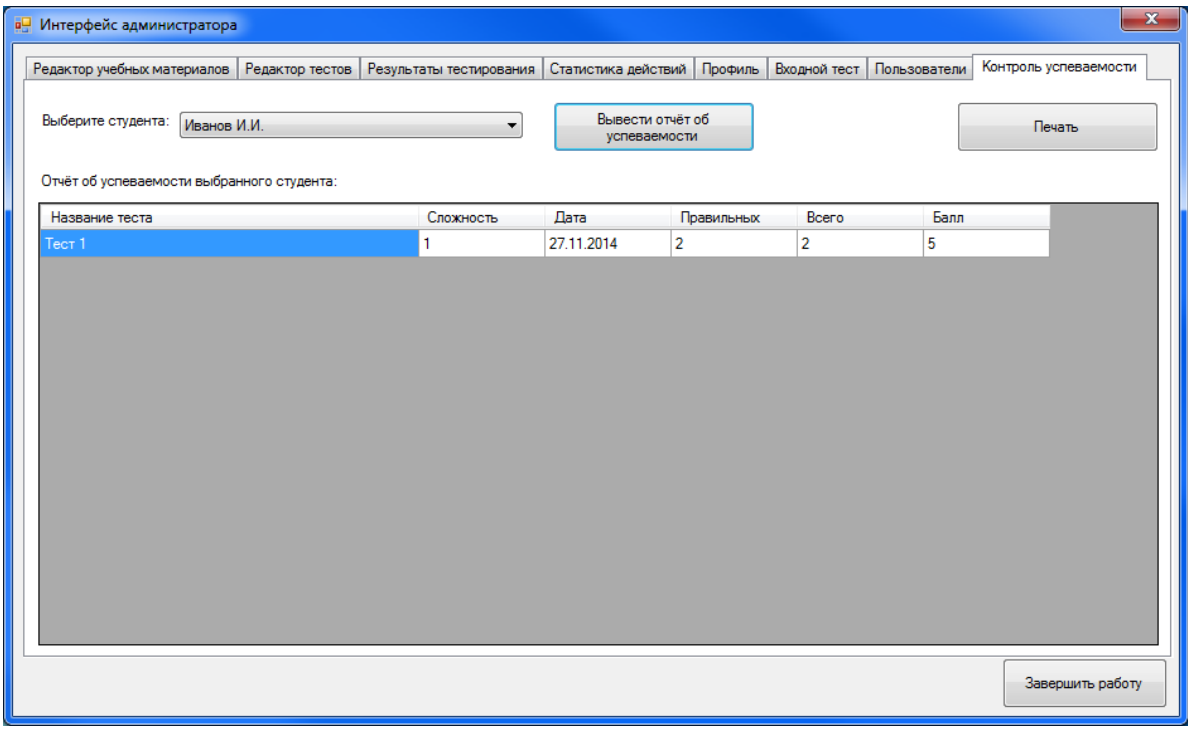


Рисунок 23 – Вкладка «Контроль успеваемости»

Интерфейс преподавателя полностью аналогичен интерфейсу администратора за единственным иключением: отсутствует вкладка «Статистика действий», вкладка «Пользователи». Рассмотрим интерфейс студента. Первая вкладка, которая открывается при входе – «Учебные материалы» (рисунок 24). Вкладка позволяет просматривать учебные материалы (рисунок 25) и добавлять их в избранное (рисунок 26).

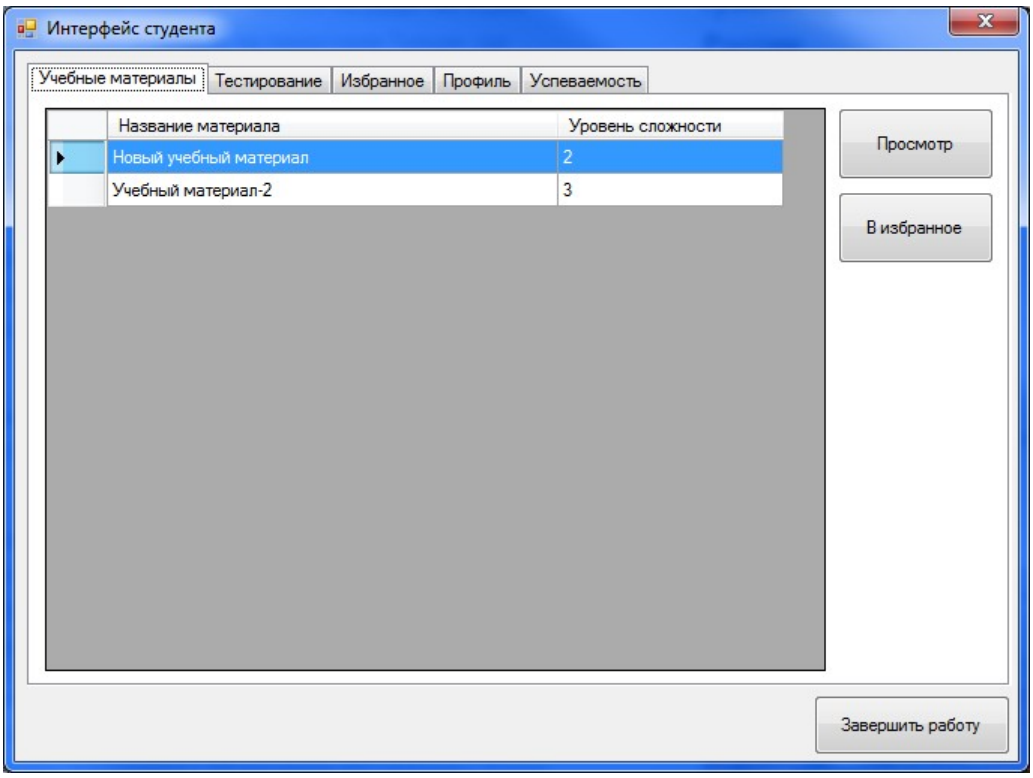


Рисунок 24 – Интерфейс студента

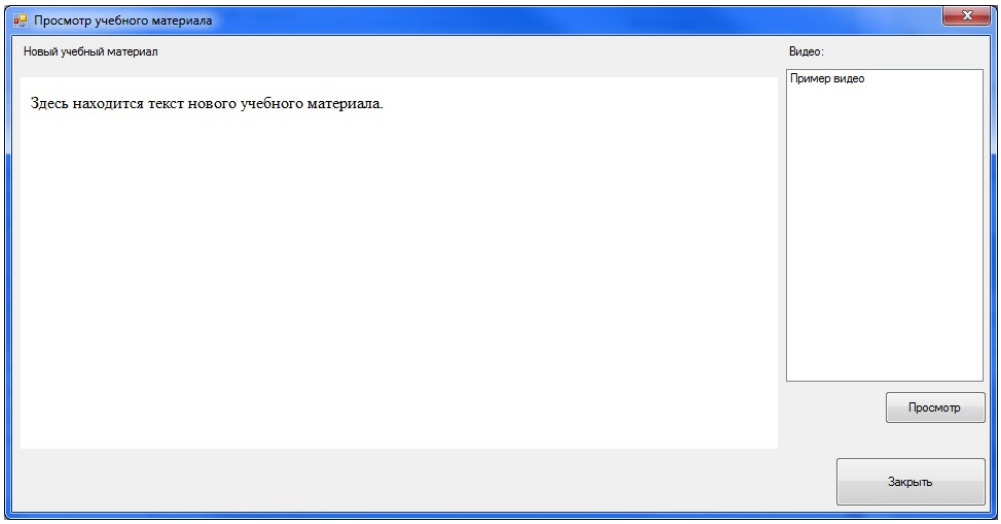


Рисунок 25 – Окно просмотра учебного материала

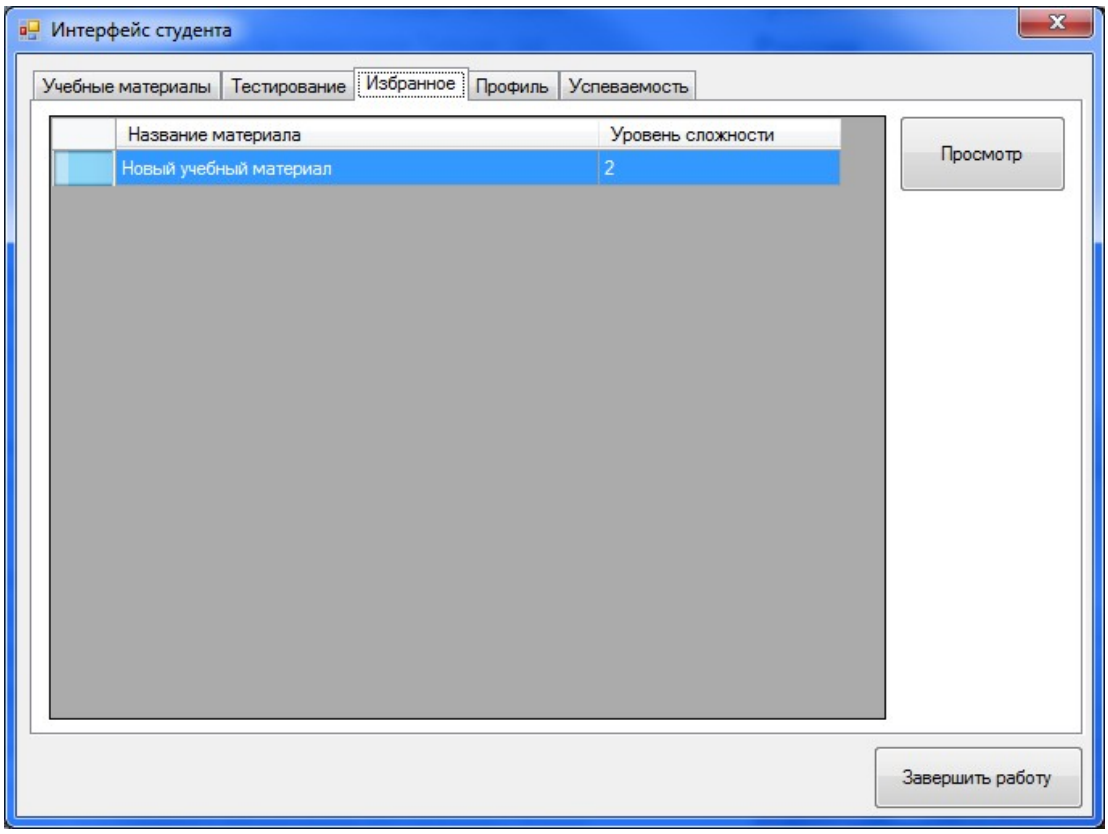


Рисунок 26 – Вкладка «Избранное»

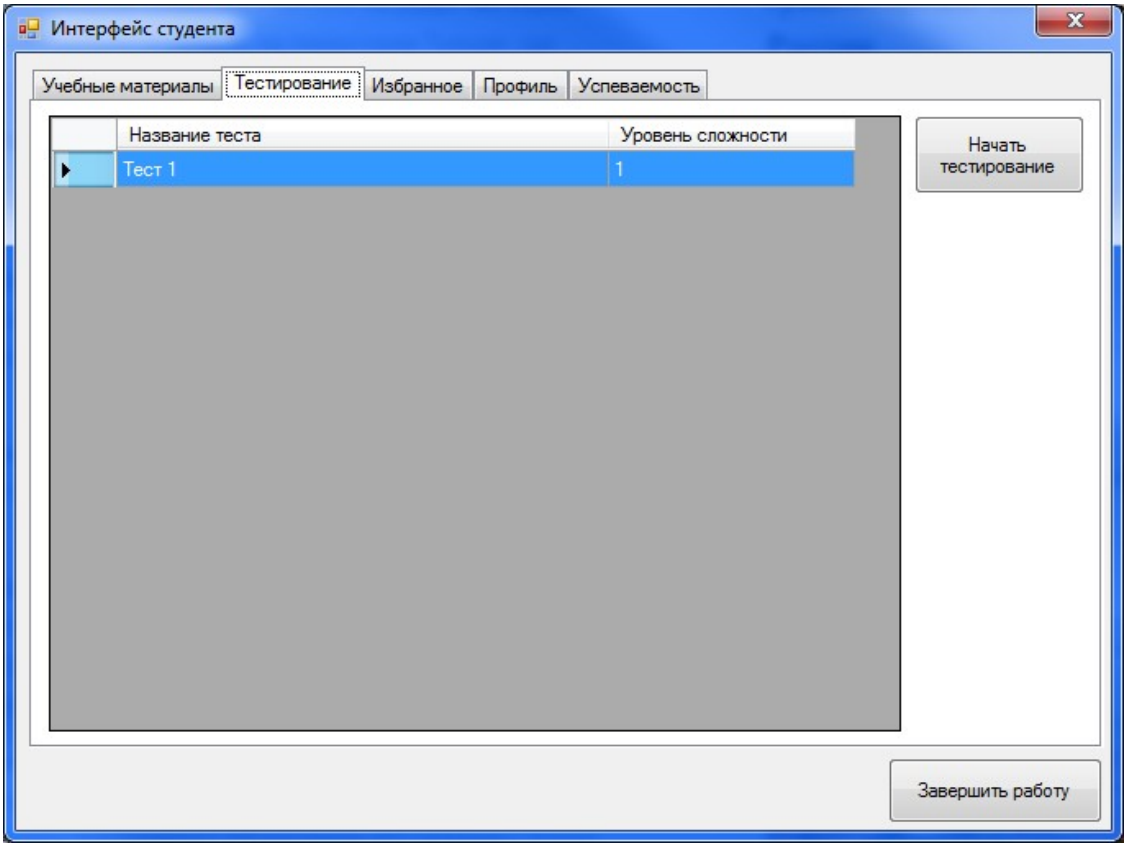


Рисунок 27 – Вкладка «Тестирование»

С помощью этой вкладки можно выбрать тест и начать сеанс тестирования. В процессе тестирования студент отвечает на вопросы с выбором вариантов ответа (рисунок 28) или на вопросы с вводом ответа (рисунок 29). После завершения сеанса тестирования на экран выводится форма с результатами (рисунок 30).

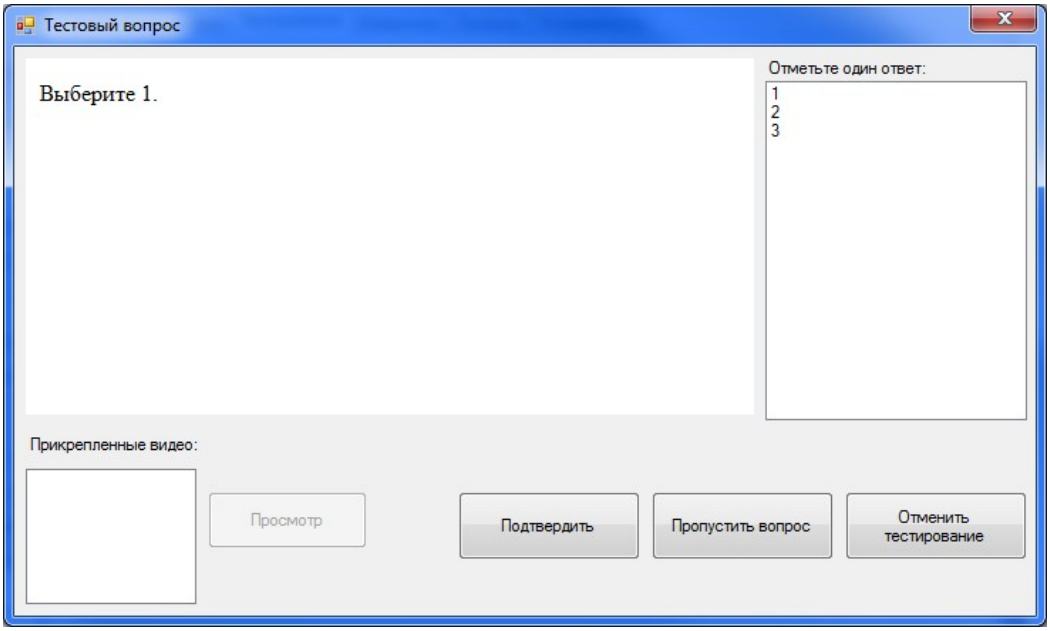


Рисунок 28 – Окно сеанса тестирования с вводом варианта ответа

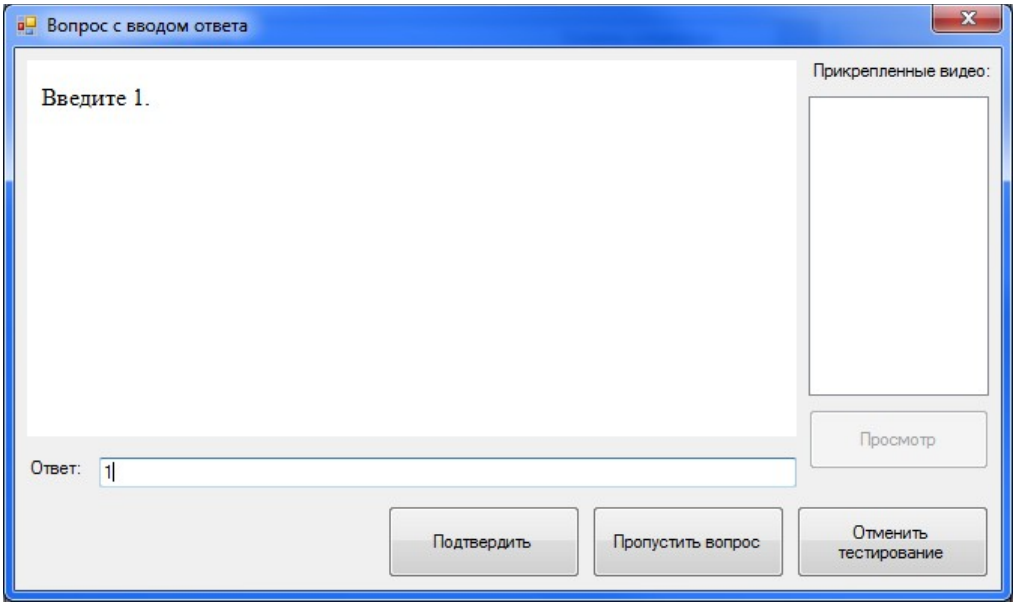


Рисунок 29 – Окно сеанса тестирования с вводом ответа

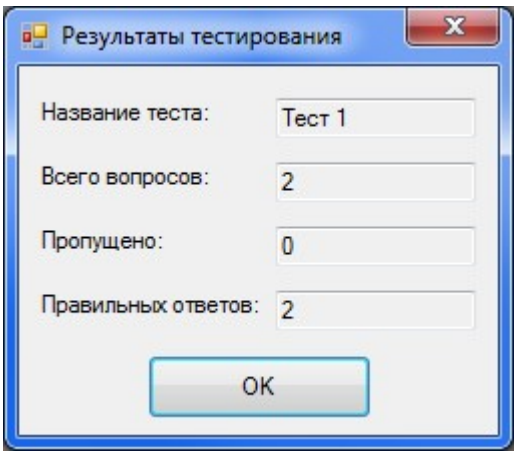


Рисунок 30 – Окно с результатами тестирования

На рисунке 31 изображена вкладка «Профиль» интерфейса студента.

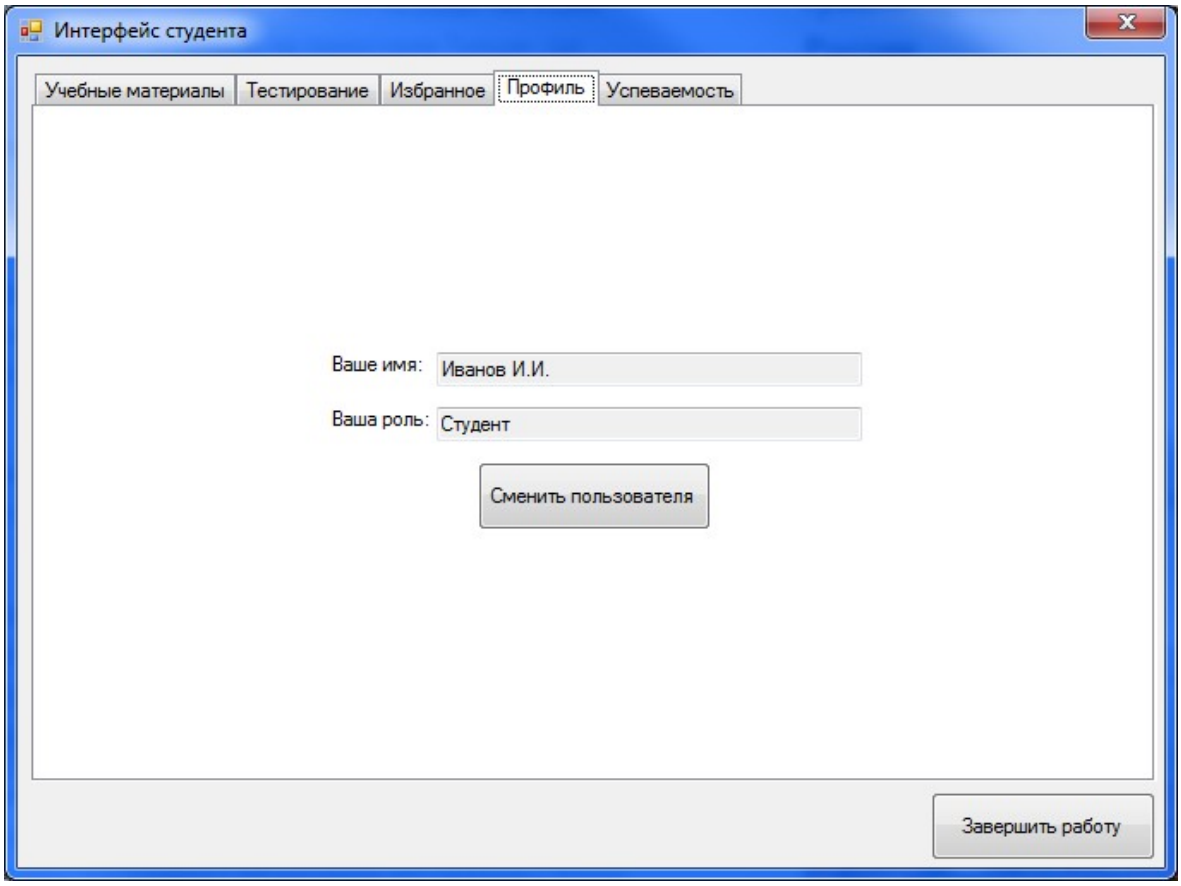


Рисунок 31 – Вкладка «Профиль»

Студент может просматривать свою успеваемость, вид соответствующей вкладки приведён на рисунке 32.

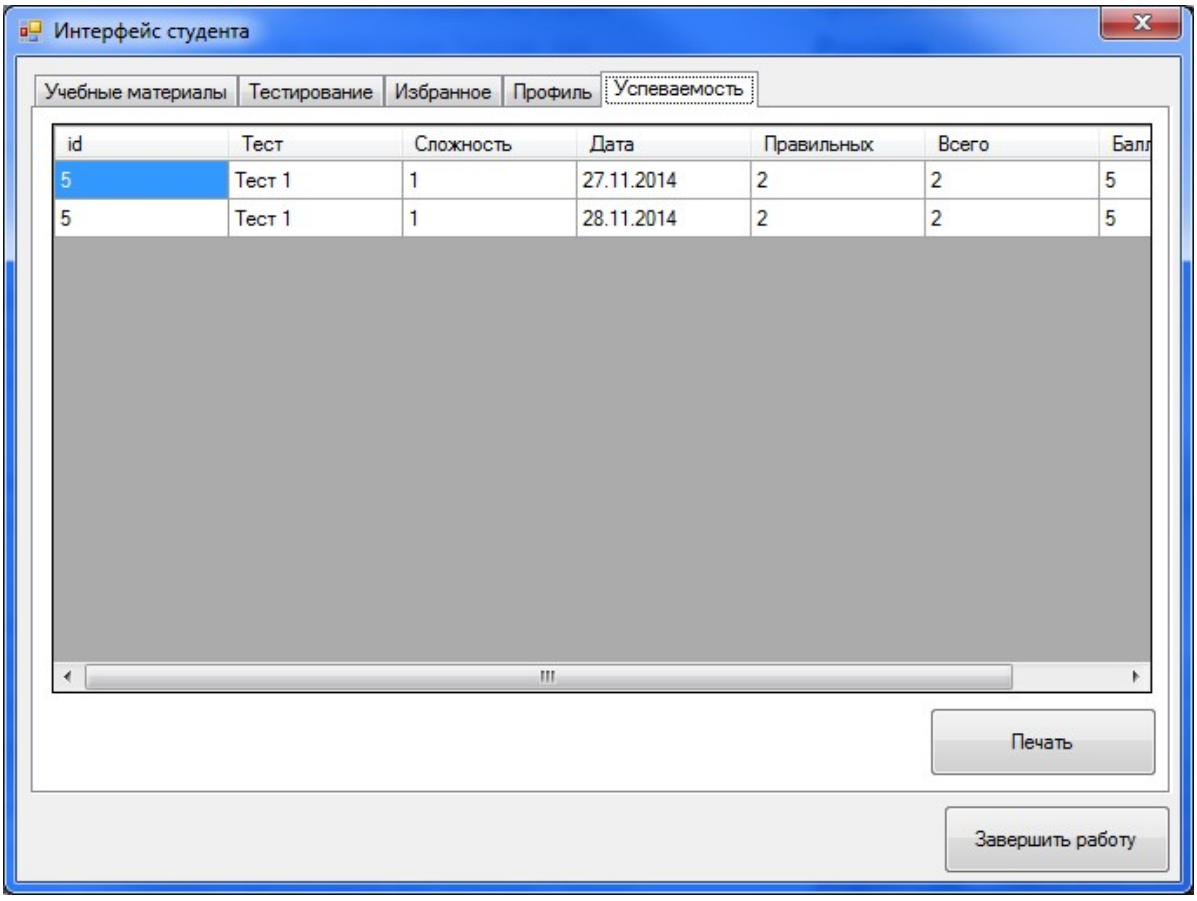


Рисунок 32 – Отчёт по успеваемости

Таким образом, определён и обоснован набор программного обеспечения, необходимый для разработки. Кроме того, разработано и представлено описание разработанного программного обеспечения: описаны все модули, подробно описан процесс работы.

Заключение

В первой главе выпускной квалификационной работы рассмотрены следующие из существующих приложений для обучения и оценки знаний студентов:

- LMS «Moodle»;
- Система «Learning Space»;
- Система «Hot Potatoes»;
- Система дистанционного обучения «Доцент»;
- Система дистанционного обучения «Прометей»;

В результате проведённого анализа было выявлено, что существующие программы для обучения и оценки знаний студентов не в полной мере соответствуют современным требованиям. По этой причине было принято решение разработать указанное программное обеспечение самостоятельно.

Кроме того, в первой главе рассмотрены методы оценки результатов тестирования учащихся.

Во второй главе выпускной квалификационной работы описан процесс разработки базы данных, предназначенной для информационного обеспечения разрабатываемого приложения. Сначала разработан проект концептуальной модели базы данных, включающий в себя 9 информационных сущностей, затем был разработан проект логической модели базы данных, состав которого насчитывает 11 сущностей. Приведены составы таблиц и связей между ними, подготовленные для физической их реализации с использованием СУБД.

В третьей главе рассмотрены различные системы программирования, с помощью которых можно разработать указанное программное обеспечение.

В данном случае пишется программа, предназначенная для оптимизации учебного процесса. В работе программы задействована сетевая СУБД и встроенный компилятор исходного кода, позволяющий «на лету» компилировать, запускать и определять правильность работы программы. Поэтому основное требование к такой программе – быстродействие, поскольку нельзя допустить, чтобы действия приложения выполнялись слишком долго, и минимальный объём используемых ресурсов, поскольку на учебных местах студентов в организации стоят далеко не суперкомпьютеры. Следовательно, нужно выбирать тот язык программирования, который обеспечивает наибольшую производительность. По указанным причинам выбран язык программирования C#.

Также в третьей главе выпускной квалификационной работы приведен процесс тестирования программного обеспечения. В результате было определено, что в программе успешно реализованы и работают следующие функции:

- разграничение прав доступа к системе;
- определение уровня знаний пользователя и постепенное его повышение, путем предоставления необходимого дидактического материала;
- обеспечение самостоятельного изучения и закрепления материала, с помощью интуитивно-понятного интерфейса, удобного для восприятия представления теоретического материала, различного вида тестов, а также с помощью различных заданий, выполнение которых требует написания программного кода;
- хранение данных о пользователе, который изучает или преподает данный предмет;
- хранение истории активности каждого пользователя, т.е. какие темы пользователь изучил, какие темы необходимо повторить, какие тесты прошел и т.п.;
- предоставление возможности преподавателю добавлять новый

учебный материал;

- хранение результатов тестирования каждого обучаемого, необходимых для формирования различных отчетов;
- формирование различных видов отчетности.

Разработанная система, предназначенная для обучения студентов, включает в себя множество функций необходимых организации процесса обучения, манипулирования дидактическим материалом, а также для вывода необходимых отчетов.

Таким образом, поставленные задачи решены, цель выпускной квалификационной работы достигнута.

Список использованной литературы

1. Архангельский А.Я. Программирование в C#/ А. Я. Архангельский. – Москва: ООО «Бином-Пресс», 2015. – 1152 с.
2. Ахо А.В. Структуры данных и алгоритмы М. : Изд. дом «Вильямс», 2015. – 384 с
3. Бойко В.В., Савинов В.М. Проектирование баз данных информационных систем. - М.: Финансы и статистика, 2015.
4. Бэбб Р. Программирование на параллельных вычислительных системах. – М.: Мир, 2016, - 376 с.
5. Вейцман В. М. Проектирование экономических информационных систем: Учебное пособие/ В. М. Вейцман; Международный университет бизнеса и новых технологий; Ярославский филиал Московского государственного университета экономики, статистики и информатики. – Ярославль: МУБИНТ, 2018.-163 с.
6. Вирт Н. Алгоритмы и структуры данных М. : Мир, 2009. – 360 с
7. Гаспариан М.С. Информационные системы/ М. С. Гаспариан. – Москва: МЭСИ, 2017. – 33 с.
8. Голощапов А.Л. Microsoft Visual Studio 2010 (+ CD-ROM). — Москва, БХВ-Петербург, 2016 г.- 544 с.
9. Горев А., Ахаян Р., Макашарипов С. Эффективная работа с СУБД. – СПб.: Питер, 2018. – 704 с.
10. Дейт К. Введение в системы баз данных. - К.: Диалектика, 2015.
11. Джексон Г. Проектирование реляционных баз данных для использования с микро - ЭВМ. - М.: Мир, 2016.
12. Дорот В.Л. Толковый словарь современной компьютерной лексики СПб. : БХВ-Петербург, 2016. – 512 с.
13. Зиборов В. «Visual C# 2010 на примерах» Издательство: БХВ-Петербур, 2018 г.

14. Карпова Т.С. Базы данных: модели, разработка. - СПб.: Питер, 2016, 304 с.
15. Карпова Т.С. Базы данных: модели, разработка, реализация. - СПб.: Питер, 2015. - 304с.
16. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика: Пер. с англ.: Учеб. пособие.- 2-е изд. – М.: Издательский дом «Вильямс», 2015. – 1120 с.
17. Котов В.Э. Элементы параллельного программирования. – М.: Радио и связь, 2017. – 192 с.
18. Кренке Д. Теория и практика построения баз данных: [пер.с англ] / Д. Кренке. - 9 - е изд. - СПб.: Питер, 2016. - 858 с.
19. Майерс Г. Архитектура современных ЭВМ: В 2-кн. Кн.1. – М.: Мир, 2016. – 364 с. Кн.2. – М.: Мир, 1985. – 312 с.
20. Мейер Б. Методы программирования: в 2-х томах. Т. 2 М. : Мир, 2017. – 368 с
21. Мейер Д. Теория реляционных баз данных. - М.: Мир, 2014. - 608с.
22. Михеева В.Д., Харитонова И.А. Microsoft Access 2015. – СПб.: БХВ – Санкт-Петербург, 2017. – 1088 с.
23. Назаров С.В. Программирование на MS VB: учебное пособие. -М.: Финансы и статистика, 2016. -320с.
24. Новицка Ю.В. Основы логического и функционального программирования Новосибирск: Изд-во НГТУ, 2016. – 60 с
25. Петров В.Н. Информационные системы. – СПб.: Питер, 2017.
26. Порублев И.Н. Алгоритмы и программы. Решение олимпиадных задач М. : Издат. дом «Вильямс», 2014. – 480 с
27. Робинсон С. Microsoft Access 2000: Учебный курс – СПб.: Питер, 2015. – 512 с.
28. Трахтенгерц Э.А. Введение в теорию анализа и распараллеливания программ ЭВМ в процессе трансляции. – М.: Наука, 2016. – 256 с.

29. Трусов М.А. "Visual C#.NET Практическое руководство для начинающего программиста" Издательство: НТ Пресс: 2016 г.
30. Ульман Дж., Уидом Дж. Введение в системы баз данных. - М.: Лори, 2017. - 374с.
31. Фаронов В.В. Программирование баз данных в C#: Учебный курс/ В.В. Фаронов. – Санкт Петербург: Питер, 2017. – 459 с.
32. Фаронов В.В. Программирование баз данных в C#: Учебный курс. – СПб.: Питер, 2014 - 464 с.
33. Хансен Г., Хансен Д. Базы данных: разработка и управление: Пер. с англ. – М.:ЗАО «Издательство БИНОМ», 2015. – 704 с.
34. Харп Г. Транспьютеры. Архитектура и программное обеспечение. – М.: Радио и связь, 2015. – 304 с.
35. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. - СПб.: КОРОНА принт, 2016. - 416с.
36. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – СПб.: КОРОНА принт, 2017. – 416 с.
37. Хомоненко А.Д., Цыганков В.М., Мальцев М.Г. Базы данных: Учебник для высших учебных заведений / Под ред. проф. А.Д. Хомоненко. – Издание 2-е, доп. и перераб. – СПб.: КОРОНА принт, 2016. – 672 с.
38. Хэзвилд Ричард: Искусство программирования на Basic. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста К. : Издательство «ДиаСофт», 2016. – 736 с
39. Чекалов А. Базы данных: от проектирования до разработки приложений. – СанктПетербург: БХВ-Петербург, 2017. – 384 с.
40. Четвериков В.Н. и др. Базы и банки данных. - М.: Высш.шк., 2015.
41. Шеннон Р. Имитационное моделирование систем – искусство и наука

М.: Изд-во «Мир», 2017. – 418 с

42. Шилдт Г. Программирование на C и C++ для Windows. – Киев: Торгово-изд. Бюро BHV, 2069. – 400 с.