

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**
(Н И У « Б е л Г У »)

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК
КАФЕДРА ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**ИНФОРМАЦИОННОЕ СОПРОВОЖДЕНИЕ ПРОЦЕССА
ТЕСТИРОВАНИЯ. МОДУЛЬ ФОНДА ТЕСТОВЫХ ЗАДАНИЙ**

Выпускная квалификационная работа
обучающегося по направлению подготовки
09.03.03 «Прикладная информатика»
очной формы обучения, группы 07001405
Васильевой Александры Сергеевны

Научный руководитель:
доцент Петина М.А.

БЕЛГОРОД 2018

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Аналитическая часть.....	6
1.1 Техничко-экономическая характеристика предметной области.....	6
1.1.1 Характеристика НИУ «Белгородский государственный университет».....	6
1.1.2 Краткая характеристика отдела электронных образовательных ресурсов.....	7
1.2 Постановка задачи.....	9
1.2.1 Цель и назначение автоматизированного варианта решения задачи	9
1.2.2 Общая характеристика организации решения задачи на ЭВМ.....	10
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи	11
1.4 Анализ существующих разработок	13
2 Проектная часть.....	17
2.1 Обоснование проектных решений.....	17
2.1.1 Обоснование проектных решений по техническому обеспечению.....	17
2.1.2 Обоснование проектных решений по информационному обеспечению	18
2.1.3 Обоснование проектных решений по программному обеспечению	19
2.1.4 Обоснование выбора программных средств	20
2.2 Информационное обеспечение задачи.....	21
2.2.1 Информационная модель и ее описание.....	21
2.2.2 Используемые классификаторы и системы кодирования.....	25
2.2.3 Характеристика базы данных	26

2.2.3.1 Характеристика инфологической модели БД	26
2.2.3.2 Характеристика даталогической модели БД.....	28
2.2.4 Характеристика результатной информации.....	30
3 Программная реализация проектных решений.....	31
3.1 Программное обеспечение задачи (комплекса задач, АРМ)	31
3.1.1 Дерево функций и сценарий диалога.....	31
3.2. Организация технологии сбора, передачи, обработки и выдачи информации	32
3.3 Описание контрольного примера реализации проекта	36
3.4 Обоснование экономической эффективности разработки.....	44
ЗАКЛЮЧЕНИЕ	49
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	51
ПРИЛОЖЕНИЯ.....	54

ВВЕДЕНИЕ

Благодаря развитию IT-технологий, в течение последних нескольких лет, образовательная сфера начала активно внедрять современные средства в учебный процесс. Одной из первых ступеней модернизации стала автоматизация оценки знаний. Для отслеживания показателей учебного процесса существует необходимость использования технологических инструментов, позволяющих осуществлять проведение оценивания знаний и успеваемости учащихся посредством тестирования.

Традиционная система контроля обеспечивает недостаточную объективность в оценке знаний учащихся, в то время как тестирование решает данную проблему и обеспечивает унифицированность процедуры оценки, поскольку тестируемые находятся в идентичных условиях и на оценку знаний влияет лишь компетентность учащегося в изученном материале.

Актуальность выбранной темы выпускной квалификационной работы обусловлена трудоемкостью процесса создания тестовых заданий в существующих системах электронного тестирования. При отображении результатов отсутствует графическое представление данных, которое обеспечивает удобство восприятия информации. Также для работы с фондом тестовых заданий необходимо подключение к глобальной сети, что ограничивает работу при ее отсутствии.

Объектом исследования является фонд тестовых заданий.

Предметом исследования является процесс формирования фонда тестовых заданий.

Целью выпускной квалификационной работы является совершенствование фонда тестовых заданий путем модернизации.

В ходе исследования были поставлены следующие задачи:

– изучение предметной области и построение исходной модели «как есть»;

- разработка модели автоматизации тестирования «как должно быть»;
- проектирование модуля хранения тестовых заданий;
- программная реализация модуля фонда тестовых заданий;
- расчет эффективности модуля.

Структура выпускной квалификационной работы определена предметом, целью и задачами исследования. Работа состоит из введения, 3 глав и заключения.

Введение содержит в себе актуальность темы, объект и предмет исследования, цели работы, задачи, которые требуется решить в ходе выполнения.

Первая глава содержит в себе теоретические сведения о предметной области, характеристика организационной структуры предприятия, назначение автоматизации решения, обоснование необходимости создания системы, анализ существующих технологий и разработок, осуществляется проектирование модели «как есть».

Во второй главе описывается обоснование проектных решения по основным признакам, обосновывается выбор программных средств, осуществляется проектирование модели «как должно быть», анализ полученных моделей, инфологическая и даталогическая модели базы данных, создание макета интерфейса.

В третьей главе описывается программная реализация системы, включающая создание базы данных и разработку системы, а также проводится расчет показателей эффективности и тестирование модуля.

В заключении описываются выводы о проделанной работе.

Выпускная квалификационная работа изложена на 53 страницах, содержит 29 рисунков, 10 таблиц, 5 приложений и 32 библиографических источника.

1 Аналитическая часть

1.1 Технико-экономическая характеристика предметной области

1.1.1 Характеристика НИУ «Белгородский государственный университет»

Сегодня в БелГУ обучается 25 тысяч студентов из 85 регионов России и 76 стран мира. Университет готовит специалистов, бакалавров и магистров по 180 направлениям подготовки и специальностям, а также ведёт обучение по 26 направлениям подготовки и по 80 образовательным программам аспирантуры и докторантуры. В НИУ «БелГУ» 12 советов по защите докторских и кандидатских диссертаций. Ведутся фундаментальные и прикладные исследования по 50 направлениям. В вузе 9 учебно-научных инновационных комплексов; 50 научно-исследовательских центров и лабораторий, в том числе: 7 институтов: Институт инженерных технологий и естественных наук, Институт управления, Медицинский институт, Педагогический институт, Юридический институт, Институт межкультурной коммуникации и международных отношений, Институт экономики.

Основными целями работы университета являются:

- наиболее полное и адекватное удовлетворение потребностей населения в образовательных и иных, реализуемых в деятельности университета, услугах;
- развитие научного потенциала общества посредством проведения
- фундаментальных и прикладных научных исследований;
- повышение коэффициента интеллекта (IQ) нации посредством увеличения в занятом населении доли лиц, имеющих высшее профессиональное образование;

– формирование гражданского общества посредством управления
– гражданским воспитанием и поведением обучающихся и работников университета.

Основными задачами университета являются:

1) подготовка, переподготовка и повышение квалификации работников с высшим образованием, научно-педагогических работников высшей квалификации, руководящих работников и специалистов по профилю Университета;

2) развитие наук и искусств посредством научных исследований и творческой деятельности научно-педагогических работников и обучающихся, использование полученных результатов в образовательном процессе;

3) удовлетворение потребностей личности в интеллектуальном, культурном и нравственном развитии посредством получения высшего и (или) послевузовского профессионального образования, среднего профессионального образования, а также дополнительного профессионального образования;

4) сохранение и приумножение нравственных, культурных и научных ценностей общества;

5) формирование у обучающихся гражданской позиции, способности к труду и жизни в условиях современной цивилизации и демократии;

6) распространение знаний среди населения, повышение его образовательного и культурного уровня. [4]

1.1.2 Краткая характеристика отдела электронных образовательных ресурсов

Развитие информационных технологий обусловило появление новой формы образования – электронное образование, то есть обучение с использованием информационно-коммуникационных технологий. Основой электронного образования являются электронные образовательные ресурсы.

Под электронным образовательным ресурсом понимают образовательный ресурс, представленный в электронно-цифровой форме (ГОСТ 52653-2006), для использования которого необходимы средства вычислительной техники.

Задачи отдела:

- организация подготовки и консультирования авторов учебных курсов, преподавателей-консультантов и организаторов-методистов электронного обучения;
- разработка нормативно-правовой базы использования технологий электронного обучения в НИУ БелГУ;
- организация работы по созданию тестовой базы для контроля знаний студентов;
- организация и контроль работ по разработке и актуализации содержимого учебно-методических материалов;
- определение технических требований к учебно-методическим материалам для подготовки электронных образовательных ресурсов;
- внесение предложений по совершенствованию и развитию системы электронного обучения НИУ «БелГУ» в части учебной работы;
- ведение учета и анализа результатов тестирования.

Основные направления деятельности отдела:

- издание и регистрация электронных образовательных ресурсов;
- разработка и наполнение образовательного портала;
- реализация применения в учебном процессе электронного обучения, дистанционных образовательных технологий.
- сопровождение проведения интернет-тестирования на сайте. [30]

1.2 Постановка задачи

1.2.1 Цель и назначение автоматизированного варианта решения задачи

Система тестирования знаний предназначена для создания и редактирования тестов, проведения тестирований и анализа результатов.

В рамках проекта автоматизируется следующая деятельность:

1) Создание тестовых вопросов разных типов: вопрос с одним вариантом ответа, вопрос множественного выбора, вопрос на сопоставление, вопрос с ручным вводом ответа.

2) Формирование бланков тестов для тестирования по главам, промежуточному и итоговому тестированию.

3) Хранение информации об изменениях в системе.

Во время работы планируется создать автоматизированную систему фонда тестовых заданий, которая будет удовлетворять требованиям пользователя.

Система создается с целью:

- ускорения работы по подготовке тестов;
- создание единой системы тестовых заданий;
- обеспечение сбора и обработки информации, необходимой для подготовки тестов;
- обеспечение сбора и обработки информации для формирования отчетности об изменениях в системе;
- обеспечение удобства использования преподавателями.

В системе должны быть улучшены значения следующих показателей:

- время работы пользователя с системой для достижения итогового результата;
- сбор и обработка информации для отчетности и дельнейшего анализа.

Под автоматизацией системы фонда тестовых заданий подразумевается решение следующих задач: автоматизация добавления тестовых вопросов, автоматизация распределения вопросов по темам и подтемам, автоматизация составления бланков тестов, ручное формирование бланков тестов.

Таким образом, при автоматизации системы фонда тестовых заданий необходимо решить ряд задач, возникающих в системах составления тестов. В настоящее время АСФТЗ широко используются в учебных заведениях. Однако, в данный момент используются устаревшие методики составления тестов. Данная разработка ориентирована на современные технологии и обеспечивает удобство использования системы преподавателями. [9, 11, 14]

1.2.2 Общая характеристика организации решения задачи на ЭВМ

Использование ЭВМ для организации решения проблемы позволит решить следующие задачи:

- изменения в функциях подразделения связанных с работой входной и выходной информации: источники поступления информации являются независимыми, так как выходят за рамки системы.

- порядок ввода первичной информации: заполнение формы авторизации и аутентификации.

- краткая характеристика результатов: формирование фонда тестовых заданий и отчет о прохождении тестирования.

- периодичность решения задачи зависит от ее типа: формирование отчетов происходит автоматически.

Перечень экранных форм: окно авторизации и аутентификации, личный кабинет преподавателя, список дисциплин, список тем, добавление тем, удаление тем, редактирование тем, список вопросов, добавление вопросов,

удаление вопросов, редактирование вопросов, вывод списка оценок, диаграммы успеваемости студентов. [13, 22, 27]

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

Модель «КАК ЕСТЬ» была построена при помощи методологии «IDEF» и программы для моделирования процессов – AllFusion Process Modeler 7.

IDEF – методология, позволяющая моделировать большие системы, а также проводить анализ моделей деятельности сложных систем в разных разрезах. Широта декомпозиции исследования определяется самим разработчик, не перегружая систему излишними углублениями в процессы.

IDEF0 – методология функционального моделирования. С помощью наглядного графического языка IDEF0 изучаемая система предстает перед разработчиками и аналитиками в виде набора взаимосвязанных функций (функциональных блоков – в терминах IDEF0). [5]

Методология IDEF0 предписывает построение иерархии диаграмм. Для начала описывается функционирование системы в целом, а также ее взаимодействие с окружающими объектами. После этого система разбивается на подсистемы, которые далее так же разбиваются на более мелкие подсистемы. Пока подробность системы не будет исчерпана.

Каждая IDEF0-диаграмма содержит блоки и дуги. Блоки изображают функции моделируемой системы. Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.

Контекстная диаграмма «Система контроля знаний. Модуль фонда тестовых заданий» изображена на рисунке 1.1.

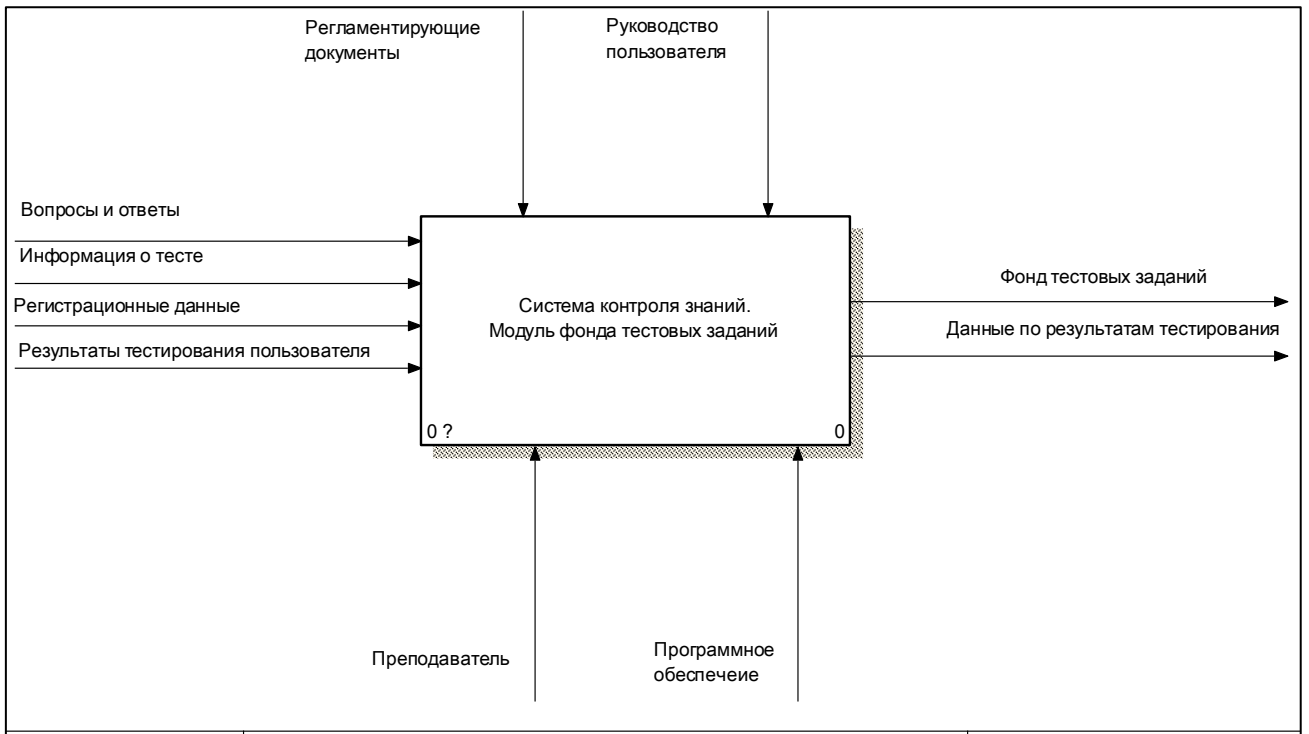


Рисунок 1.1 – Контекстная диаграмма

Диаграмма декомпозиции представляет собой детализацию процессов контекстной диаграммы.

На рисунке 1.2 отображена спроектированная диаграмма декомпозиции.

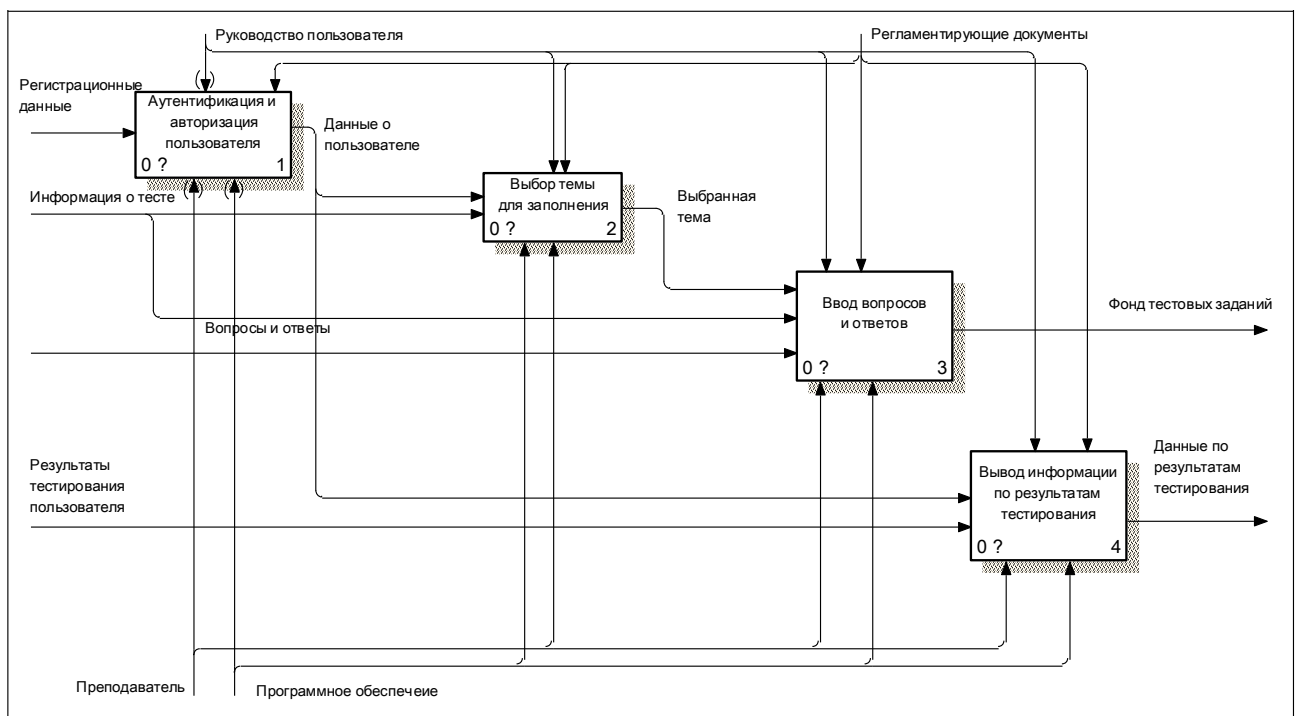


Рисунок 1.2 – Диаграмма декомпозиции

Диаграмма содержит 4 блока: «Аутентификация и авторизация пользователя», «Выбор темы для заполнения», «Ввод вопросов и ответов», «Вывод информации по результатам тестирования».

Управляющими документами выступают: руководство пользователя, регламентирующие документы.

Механизмом системы является: программное обеспечение и преподаватель.

1.4 Анализ существующих разработок

Тестирование PikaTest это утилита в свободном доступе для создания и проведения тестирования с двумя уровнями и любым количеством вопросов, таких как USE. Это интересно прежде всего потому, что это автономная система тестирования, которые на сегодняшний момент встречаются редко. Программа вышла на рынок в начале 2012 года.

PikaTest позволяет преподавателю создать полный тест с любым количеством вопросов разного типа. Можно добавлять вопросы с несколькими вариантами ответа и без них, а также обозначать каждый вопрос. Программа дает возможность создавать тесты, для которых установлено ограничение по времени прохождения. Сохранение файла тестов происходит в формате .pikatest, воспринимаемый только внутри программы.

Подробный отчет по тестированию хранится в формате *.txt и доступен в любое время.

Основные преимущества:

- 1) система не зависит от других программ и имеет маленький объем (всего 1761 КБ);
- 2) система бесплатна для использования;
- 3) программа проста в использовании и не требует особых знаний;

4) совместима со всеми ОС (XP 2003 Vista 7).

Недостатки:

1) система работает автономно, а помощь в программе доступна только через глобальную сеть;

2) отсутствует пользователь-администратор;

3) данные хранятся небезопасно;

4) отсутствуют многие полезные функции. [31]

Тестирование UniTest это программный комплекс для проведения компьютерного тестирования, созданное с использованием технологии SAM. Первая версия утилиты появилась в 2007 году.

Программа для создания фонда тестовых заданий и организации проверки знаний. Поддерживает протокол TCP/IP, а также тестирование внутри локальной сети.

Система UniTest это программное обеспечение, реализованное при использовании технологий Microsoft.NET 3.0, использующих алгоритмы, обрабатывающие информацию параллельно и средства защиты при помощи криптографии.

Программа UniTest имеет два модуля: UniTest TUTOR и UniTest CUSTOMER:

1) Модуль TUTOR работает с двумя категориями пользователей («Репетитор», «Администратор») и служит для обеспечения работы с банком тестовых задач, которые могут включать текстовую часть и различные мультимедийные компоненты. Модуль TUTOR позволяет организовать хранилище тестовых заданий по разделам, а также создать любое количество сценариев тестирования, предоставляет возможность составлять электронные журналы по результатам тестирования, чтобы структурировать студентов по отделам и группам.

2) Модуль CUSTOMER предназначен для пользователя «Студент», который предоставляет доступ к завершенным тестам системы UniTest и виртуальной учетной записи с котировками.

Система бесплатна и доступна для загрузки на официальном сайте.

Основные преимущества:

- 1) удобная работа для пользователя;
- 2) практичный интерфейс, объемные тесты занимают небольшой объем памяти (500-700 Кбайт);
- 3) поддержка вопросов основных и других типов;
- 4) высокий уровень защищённости данных (все данные зашифрованы BlowFish 448 бит);
- 5) возможность локального тестирования и по глобальной сети;
- 6) поддержка основных языков;
- 7) большое количество полезных функций.

Недостатки: высокие требования к техническим средствам; отсутствует совместимость с популярными версиями ОС (Windows 7/8); логин и пароль для пользователей назначается администратором системы. [7]

Система тестирования Indigo это программный пакет, позволяющий автоматизировать процесс проведения тестирования и обработку результатов. Продукт был разработан в 2010 году.

Система INDIGO является удобным инструментом, который может использоваться для решения многих задач:

- 1) Определение уровня подготовленности школьников к ГИА и ЕГЭ.
- 2) Тестирование и мониторинг уровня знаний студентов по учебным дисциплинам.
- 3) Определение уровня профессионализма сотрудников (в том числе при найме).
- 4) Автоматизация психологического тестирования, включая выбор профессии.
- 5) Проведение социологических и маркетинговых опросов.

Работа с INDIGO делится на два модуля: интерфейс администратора и пользовательский интерфейс.

Интерфейс администратора тестовой оболочки — это приложение Windows, которое реализует следующие функции: создавать и редактировать тесты; управление тестовой базой данных; управление пользовательской базой; управление веб-сервером и правилами тестирования; доступ к результатам тестирования. [6]

Пользовательский интерфейс тестовой оболочки — это веб-интерфейс, который реализует следующие функции:

- 1) Просмотреть доступных тестов.
- 2) Регистрация и авторизация пользователей в системе.
- 3) Выбор теста и производительности

В результате проведенного анализа всех выбранных систем, можно сказать, что если нет необходимости в организации всего обучающего процесса, то лучшей программой, для проведения интерактивного тестирования, послужит UniTest. Однако, отсутствие совместимости с Windows 7/8 является серьезным ограничением в работе. Следовательно, собственное решение – это наилучший вариант для построения системы, которая будет соответствовать предъявляемым требованиям, как техническим, так и функциональным.

Таким образом, эффективная система тестирования должна обладать следующими характеристиками:

- Возможность тестирования как локально, так и по сети.
- Использование современных криптографических средств для защиты информации.
- Наличие системы аутентификации пользователей.
- Система должна быть модульной, для удобства использования.
- Программа должна быть тесно интегрирована с АСУ ВУЗа.

2 Проектная часть

2.1 Обоснование проектных решений

2.1.1 Обоснование проектных решений по техническому обеспечению

Комплекс технических средств предназначен для автоматизации обработки данных.

Требования:

- минимизация издержек;
- надежность и защита;
- распределение по уровням обработки.

В комплексе технических средств выделяют:

1) Средства сбора и регистрации информации: датчики и счетчики, которые фиксируют какие-либо события и подсчитывают значения различных показателей; устройства измерения; персональные компьютеры; для считывания и преобразования данных с документов используются сканеры.

2) Комплекс средств передачи информации: GPS связь; локальные, глобальные и региональные компьютерные сети; телеграфная связь; радиосвязь; спутниковая связь.

3) Средства хранения данных: CD и DVD оптические диски; flash, и HDD USB-накопители; жесткие диски.

4) Средства обработки данных: суперкомпьютеры, ноутбуки, карманные компьютеры.

5) Средства вывода информации: мониторы, принтеры и плоттеры.

6) Средства организационной техники: работа с документами (изготовление, копирование, обработка и уничтожение); банкоматы, детекторы подсчета денежных купюр и проверки их на подлинность.

Техническое обеспечение описано в таблице 2.1.

Таблица 2.1 – Техническое обеспечение

№	Наименование	Характеристика
1	Материнская плата	Acer Aspire E1-531
2	Процессор	Intel(R) Pentium(R) CPU B960 @ 2.20GHz
3	Жесткие диски	WDC WD7500BPVT-22HXZT3 (698 ГБ)
4	Модули памяти. Flash-накопители	DDR-память: DIMM DDR 256Mb pc3200 (400MHz) Samsung; Flash-карты: SD Card 256Mb Kingmax
5	Видеокарты	Intel(R) HD Graphics
6	Звуковые карты	Realtek ALC269 @ Intel Panther Point PCH - High Definition Audio Controller [C-1] PCI
7	Приводы DVD- ROM/RW	MATSHITA DVD-RAM UJ8E1
8	Дисководы	Fdd 3.5" 1.44 Mb NEC black
9	Сетевой фильтр	Сетевой фильтр (5 розеток) Pilot GL 2м

2.1.2 Обоснование проектных решений по информационному обеспечению

Для создания рабочей программы дисциплины необходим утвержденный шаблон (правила оформления) и следующие документы:

1) учебные планы направления подготовки всех реализуемых форм обучения (из которых берётся информация о часах, о семестрах, о количестве зачетных единиц, о формах контроля и др.);

2) карты компетенций направления подготовки (из которых берётся информация о результатах освоения дисциплины);

3) ФГОС по направлению подготовки.

Таким образом, структура документов методического отдела в рамках деятельности по разработке рабочих программ дисциплин имеет 4 группы: карты компетенций, учебные планы, ФГОС и непосредственно рабочие программы дисциплин.

Необходимое информационное обеспечение отображено в таблице 2.2.

Таблица 2.2 – Информационное обеспечение

№	Наименование	Характеристика
1	Паспорт фонд тестовых заданий	Содержит в себе тестовые вопросы и ответы
2	Учебный план	Содержит количество тестирований для каждого предмета

2.1.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение – это множество программ, которые обеспечивают работу компьютеров и решение с их помощью задач различных предметных областей. Программное обеспечение – это неотъемлемая часть компьютерной системы, которая является логическим продолжением технических средств, а также определяет сферу применения компьютера.

Для реализации системы было использовано программное обеспечение, приведенное в таблице 2.3.

Таблица 2.3 – Программное обеспечение

№	Наименование	Характеристика
1	2	3
1	Операционная система	Microsoft Windows 10 Pro: Стандартные программы. Служебные программы.
2	Пакет прикладных программ	Microsoft Office: Microsoft Office Word 2016.
3	Файловые менеджеры	Windows Commander
4	Графические редакторы	Corel Draw. Adobe Illustrator. Adobe Photoshop.

Продолжение таблицы 2.3

1	2	3
5	Информационно-поисковые системы	Google
6	Текстовый редактор	Sublime Text 3
7	Архиваторы	WinRAR.
8	Веб-сервер	Apache 2.4
9	Система управления базами данных	PHPMyAdmin 4.6.6
10	Текстовый редактор	Sublime Text 3

2.1.4 Обоснование выбора программных средств

Apache HTTP-сервер – это свободный веб-сервер. [19]

Данный веб-сервер поддерживает следующие операционные системы: Linux, BSD, Mac OS, Microsoft Windows, Novell NetWare, BeOS.

Apache имеет надежную и гибкую конфигурацию. Для предоставления данных он позволяет подключать внешние модули и для аутентификации пользователей использовать СУБД, проводить модификацию сообщений об ошибках.[24]

Sublime Text – кроссплатформенный текстовый редактор. Данная программа является бесплатной и не имеет ограничений на время использования.

Изменения в новой версии: быстрее запускается программа; улучшился интерфейс и работа поиска; использование Python 3 для плагинов; индексирование файлов для поиска объявления переменной, функции или класса; перемещение плагинов в отдельный процесс для предотвращения задержек загрузки и краха Sublime Text. [20]

PhpMyAdmin – система управления базами данных, имеет открытый код, который написан на PHP и представляет собой интерфейс

администрирования СУБД MySQL. PHPMyAdmin позволяет через браузер и не только осуществлять администрирование сервера MySQL, выполнять команды SQL и просматривать базы данных. Приложение пользуется большой популярностью у веб-разработчиков, так как позволяет управлять СУБД MySQL без непосредственного ввода SQL команд, предоставляя дружелюбный интерфейс. [21]

2.2 Информационное обеспечение задачи

2.2.1 Информационная модель и ее описание

При описании модели «КАК ДОЛЖНО БЫТЬ» контекстная диаграмма была спроектирована в нотации IDEF0 (рисунок 2.1).

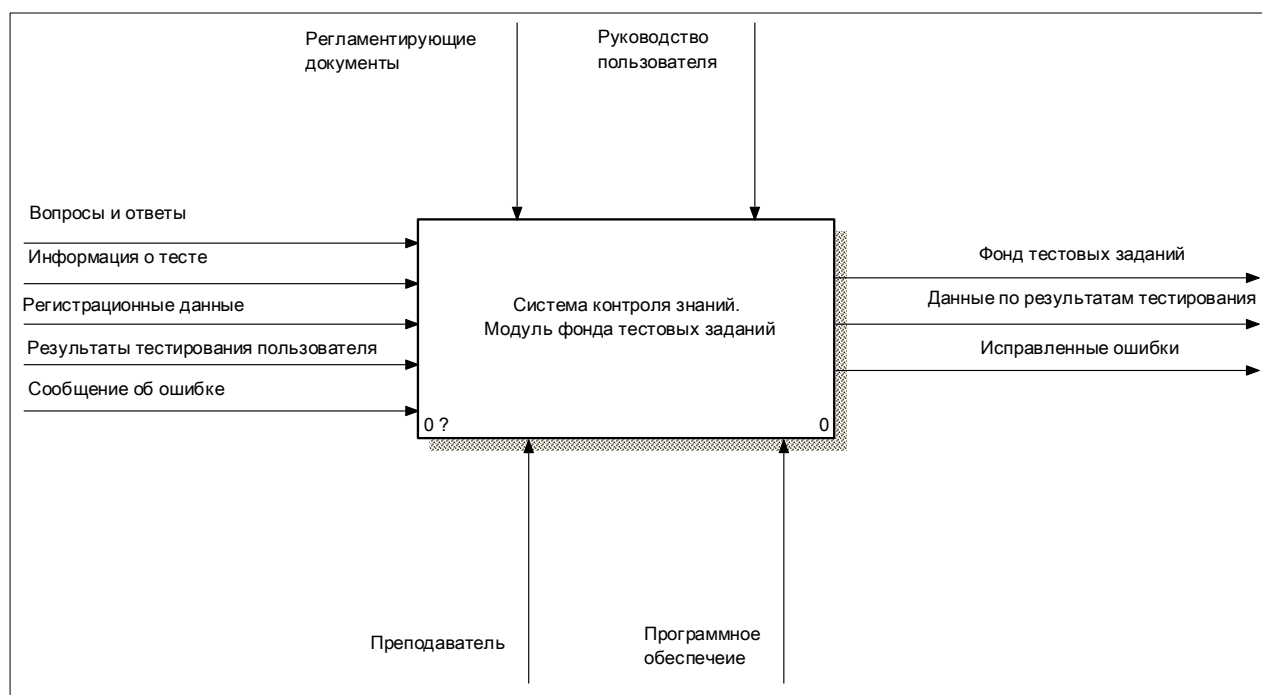


Рисунок 2.1 – Контекстная диаграмма

В отличие от модели «КАК ЕСТЬ» выходной информацией является возможность получить обратную связь от студента во время просмотра вопросов (рисунок 2.2).

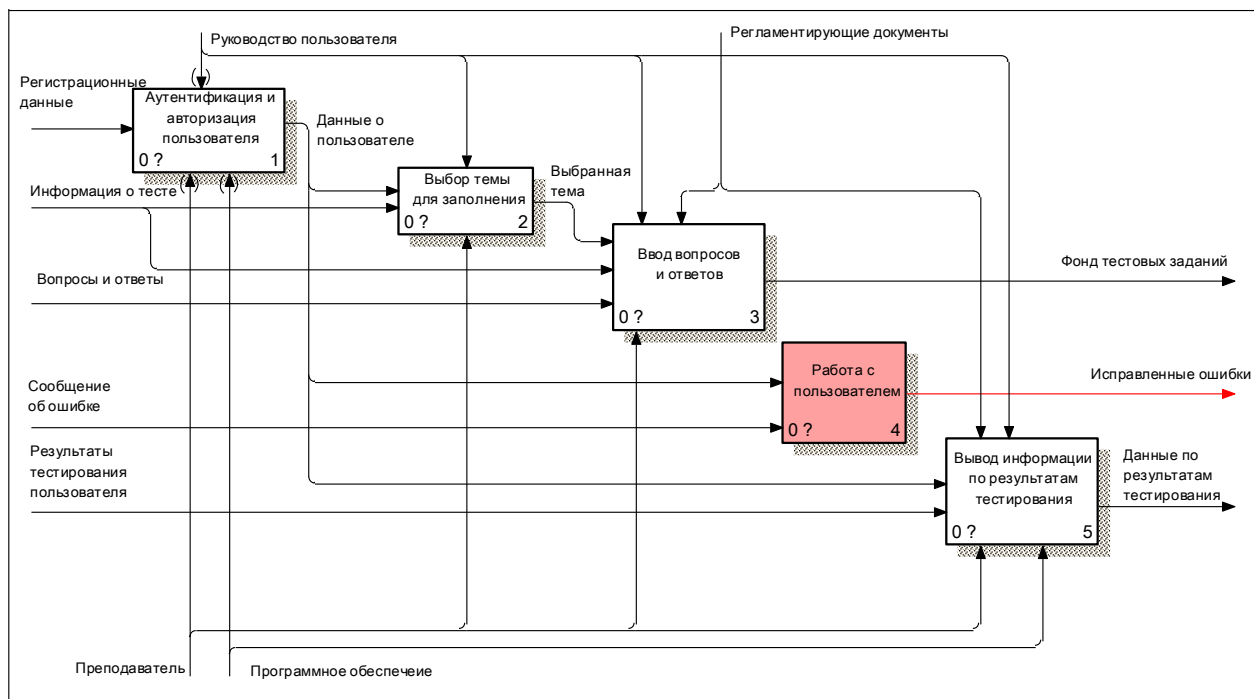


Рисунок 2.2 – Диаграмма декомпозиции

Функциональный блок «Работа с пользователем» включает в себя выполнение следующих работ: просмотр жалоб, редактирование вопросов, редактирование ответов и отправку результата (рисунок 2.3). Оператор «асинхронное или» означает, что одна или несколько работ запускаются, то есть преподаватель может редактировать вопросы или редактировать ответы в разном порядке и затем сохранить результат.

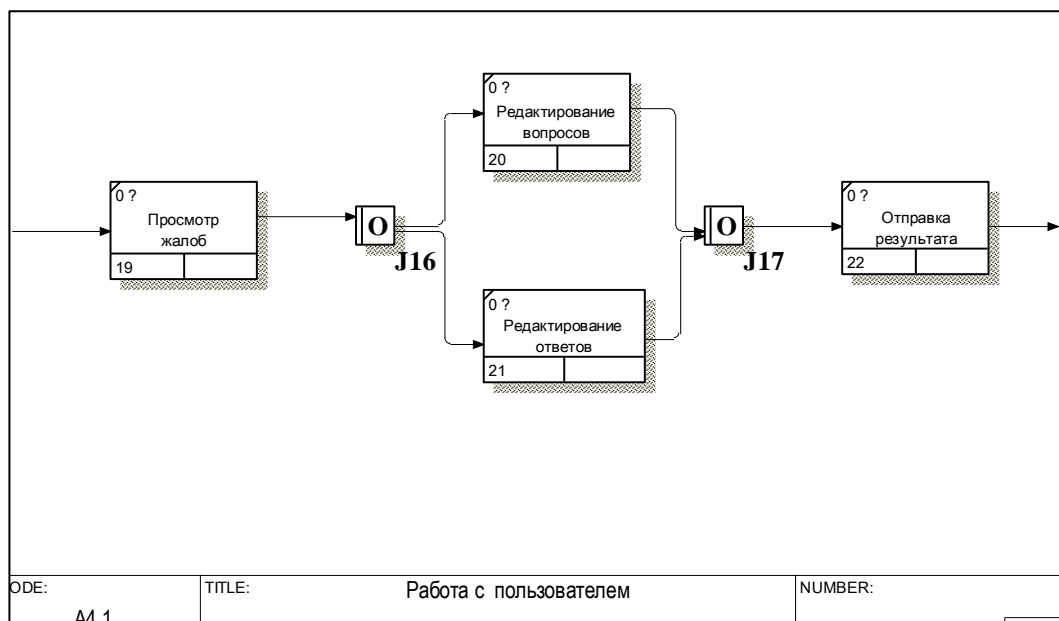


Рисунок 2.3 – Диаграмма декомпозиции «Работа с пользователем»

Общий вид системы контроля знаний изображен на рисунке 2.4.

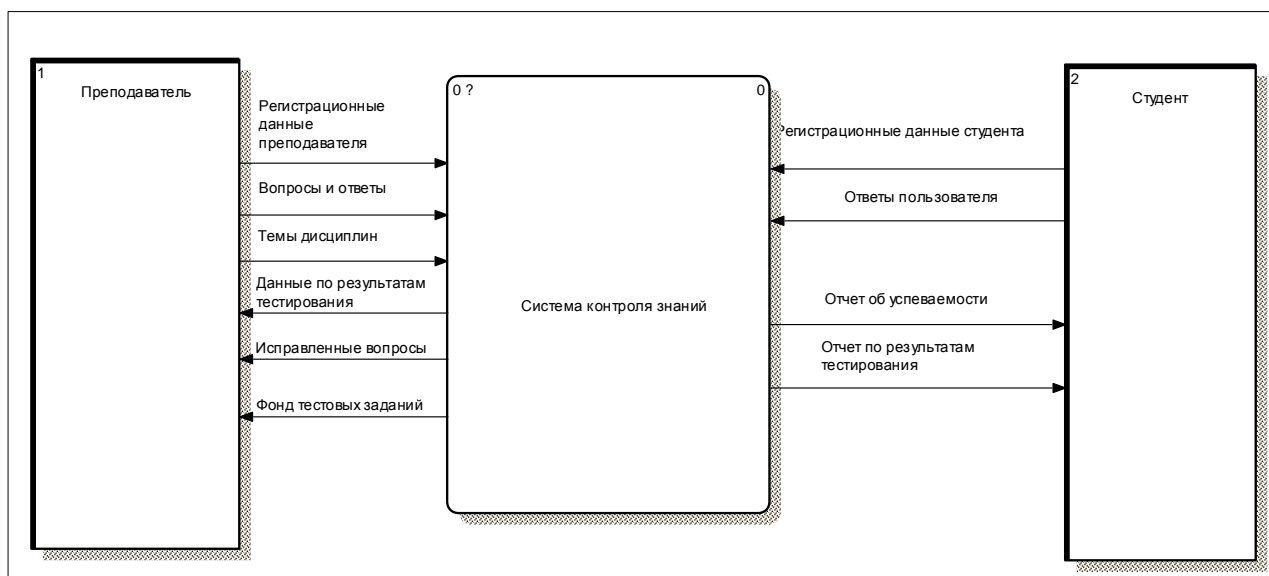


Рисунок 2.4 – Контекстная диаграмма системы

Система контроля знаний взаимодействует с внешними сущностями: преподаватель и студент. Преподаватель передает: регистрационную информацию, вводит вопросы и ответы в созданные ранее темы дисциплин.

Детализация контекстной диаграммы изображена на рисунке 2.5. Данная декомпозиция отображает работу двух модулей одновременно: модуль фонда тестовых заданий и модуль самого тестирования.

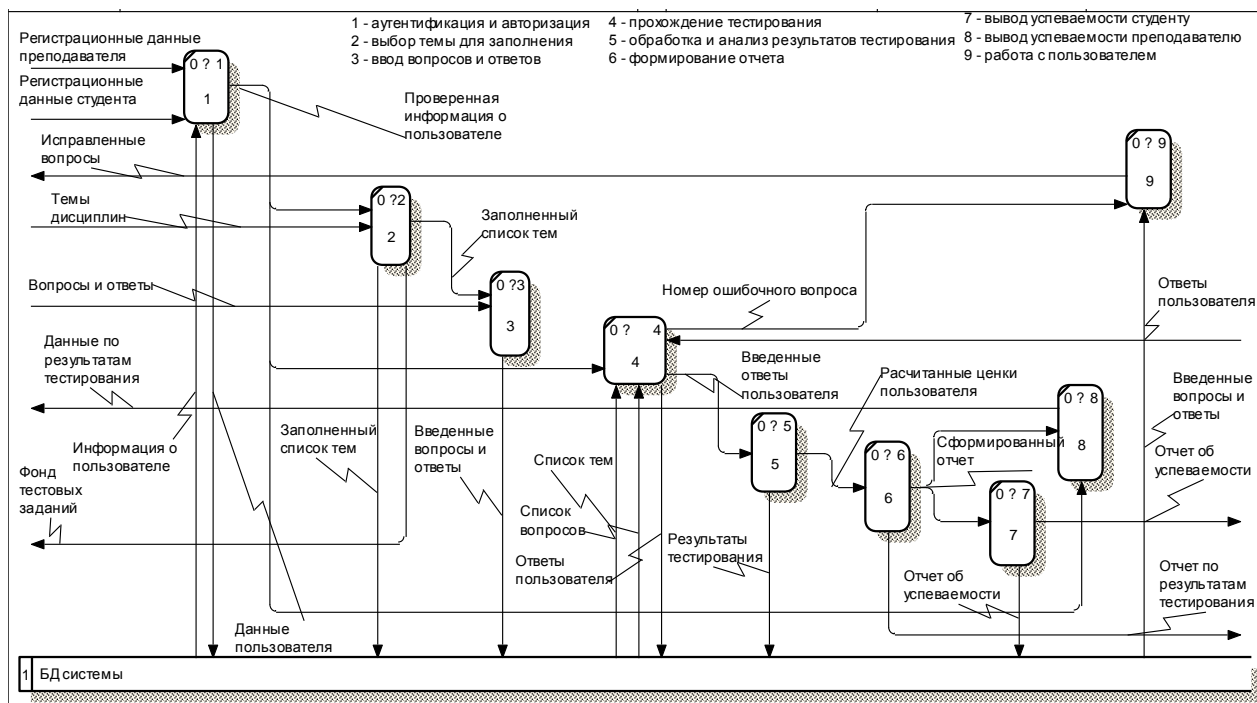


Рисунок 2.5 – Диаграмма декомпозиции «Система контроля знаний»

Преподаватель проходит авторизацию и заполняет темы дисциплин. После выполнения данных действий переходит к вводу вопросов и ответов. Затем студент проходит тестирование и формируется отчет об его успеваемости, эти данные перенаправляются преподавателю в личный кабинет.

Так же во время тестирования студент может сделать пометку «ошибка в тесте», преподаватель увидит данное сообщение у себя в личном кабинете и внесет правку.

Разработанная модель системы «КАК ДОЛЖНО БЫТЬ» отображает все процессы работы системы. Поставленные задачи к разработке и реализации были учтены в моделировании процессов. [10, 12, 25, 1]

2.2.2 Используемые классификаторы и системы кодирования

Автоматизация обработки позволяет составить сводки/таблицы/ведомости, т.е. сгруппированную информацию по каким-либо признакам или реквизитам, например, по функционирующим подразделениям.

Для того, чтобы сгруппировать данные необходимо присвоить кодовые обозначения, которые использует система классификаций и кодирования. С помощью данной системы информация приобретает читабельную форму для работы с машиной.

Кодированию подлежит информация, которую необходимо сгруппировать по реквизитам-признакам.

Классификатор – это систематизированный свод однородных наименований, т. е. классифицируемых объектов и их кодовых обозначений.

Код представляет собой условное обозначение объекта знаком или группой знаков по определенным правилам, установленным системами кодирования.

Коды могут быть разных типов: цифровыми, буквенными, комбинированными. При обработке экономической информации часто применяют мнемокоды, штрих-коды; в ряде случаев машина сама может кодировать заносимые в нее объекты.

Кодирование – это процесс присвоения объектам кодовых обозначений.

Основной задачей кодирования является однозначность обозначения объектов и обеспечение достоверной кодируемой информации.

Организация кодирования позволяет выполнять основные функции, которые связаны с обработкой различной экономической информации: уменьшает объем данных при вводе в вычислительную систему и при передаче по каналам связи, облегчает поиск и сортировку по различным признакам, облегчает работу разработки сводной отчетности.

Структура кодовых обозначений объектов приведена в таблице 2.4.

Таблица 2.4 – Кодовые обозначения

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Вид классификатора
ID пользователя	11	Порядковая	Локальный
ID ответа	11	Порядковая	Локальный
ID дисциплины	11	Порядковая	Локальный
ID ошибки	11	Порядковая	Локальный
ID избранного	11	Порядковая	Локальный
ID группы	11	Порядковая	Локальный
ID вопроса	11	Порядковая	Локальный
ID оценки	11	Порядковая	Локальный
ID темы	11	Порядковая	Локальный
ID типа вопроса	11	Порядковая	Локальный
ID типа пользователя	11	Порядковая	Локальный

Представленные коды использованы при создании базы данных.

2.2.3 Характеристика базы данных

2.2.3.1 Характеристика инфологической модели БД

Под инфологической моделью – понимают описание предметной области, выполненное с использованием специальных языковых средств, не зависящих от используемых в дальнейшем программных средств.

Инфологическая модель строит в независимости от типа используемой СУБД.

На рисунке 2.6 представлена взаимосвязь основных этапов проектирования БД. Инфологическая модель предметной области строится первой. Предварительная инфологическая модель строится еще на предпроектной стадии и затем уточняется на более поздних стадиях проектирования. Затем на основе инфологической модели строится даталогическая модель, а затем идет физическая модель (используется для конкретной привязки даталогической модели к среде хранения).

Требования, предъявляемые к инфологическим моделям:

1) Адекватное отображение предметной области. В связи с этим язык для представления ИЛМ должен обладать даталогическими выразительными возможностями для отображения явлений предметной области.

2) Инфологическая модель должна быть непротиворечива. Она является единым интегрированным описанием предметной области и отражает взгляды и потребности всех пользователей систем. (не должно быть неоднозначной трактовки модели)

3) Модель инфологическая конечна. Модель должна удовлетворять свойству конечности, т. к. предметная область ограничена.

4) Инфологическая модель должна обладать свойством расширяемости. Должен обеспечиваться ввод новых данных и удаление старых данных.

5) Инфологическая модель должна обеспечивать свойство композиции и декомпозиции. Реальные СУБД очень большие и модель должна обеспечивать возможности композиции и декомпозиции.

6) Инфологическая модель должна легко восприниматься разными категориями пользователей. [14]

Продолжение таблицы 2.5

1	2	3	4
Errors	<i>id</i>	int(11)	NO
	question_id	int(11)	NO
Favourites	<i>id</i>	int(11)	NO
	discipline_id	int(11)	NO
	id_user	int(11)	NO
Group	<i>id</i>	int(11)	NO
	number	varchar(10)	NO
Type_question	<i>id</i>	int(11)	NO
	name	varchar(30)	NO
Question	<i>id</i>	int(11)	NO
	text_quest	varchar(500)	NO
	id_theme	int(11)	NO
	id_type	int(11)	NO
Score	<i>id</i>	int(11)	NO
	id_stud	int(11)	NO
	discipline_id	int(11)	YES
	theme_id	int(11)	YES
	number_score	float	NO
	date	datetime	YES
Student_group	<i>id</i>	int(11)	NO
	id_stud	int(11)	NO
	id_group	int(11)	NO
Theme	<i>id</i>	int(11)	NO
	name	varchar(100)	NO
	id_disc	int(11)	NO
	pass	varchar(30)	NO
	name	varchar(20)	NO
	surname	varchar(20)	NO
	type_id	int(11)	NO
User_discip	<i>id</i>	int(11)	NO
	user_id	int(11)	NO
	discip_id	int(11)	NO

На основании информации из таблицы строятся запросы для создания базы данных.

2.2.4 Характеристика результатной информации

Результирующей информацией данной выпускной квалификационной работы является:

1) Исправленные ошибки в тестах. Учащиеся во время тестирования имеют возможность сообщить преподавателю о том, что вопрос содержит неверную формулировку или ответы на вопросы содержат ошибку. За данный процесс отвечают таблицы: discipline, theme, error, question.

2) Отчет о результатах прохождения тестирования студентов. После прохождения тестирования пользователем преподавателю приходят оценки в личный кабинет. За данный процесс отвечают таблицы: user, discipline, theme, score.

3) Список тем. Преподаватель имеет возможность внести в дисциплину, которую он преподает, список тем. За данный процесс отвечают таблицы: user, theme.

4) Список вопросов. Преподавателю предоставляется возможность внести список вопросов в созданные ранее темы. За данный процесс отвечают таблицы: type_question, user, discipline, theme, question.

3 Программная реализация проектных решений

3.1 Программное обеспечение задачи (комплекса задач, АРМ)

3.1.1 Дерево функций и сценарий диалога

Взаимодействие с пользователем на сайте происходит через нажатие ссылок в меню, на странице, либо кнопок формы. По нажатию срабатывает контроллер и вызывает функцию в файле-модели. Каждая функция отвечает за разные действия с данными.

Для иерархического представления функций, выполняющихся во время работы системы используется модель дерева функций. Функции делятся на основные и вспомогательные. Основные отвечают за прямое взаимодействие с пользователем. Вспомогательные – за внутреннюю обработку данных. Дерево функций представлено на рисунке 3.1.

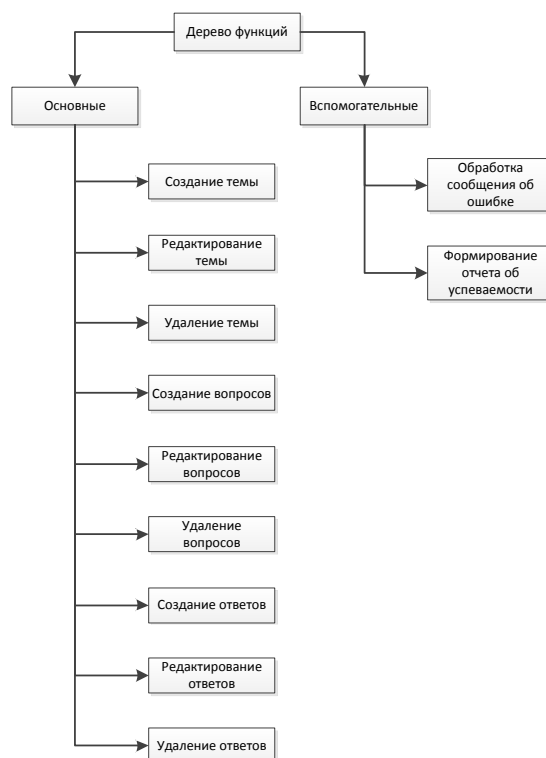


Рисунок 3.1 – Дерево функций

Структура основной ветки состоит из 9 функций:

- создание темы – позволяет пользователю создать новую тему в базе данных;
- редактирование темы – позволяет изменить существующую тему;
- удаление темы – отвечает за удаление темы из базы данных;
- создание вопросов – позволяет пользователю создать новый вопрос в базе данных;
- редактирование вопросов – позволяет изменить существующий вопрос;
- удаление вопросов – отвечает за удаление вопроса из базы данных;
- создание ответов – позволяет создать ответы на вопросы;
- редактирование ответов – позволяет редактировать текст ответа;
- удаление ответов – позволяет удалить ответ из базы данных.

Вспомогательная часть дерева состоит из 2 функций:

- обработка сообщения об ошибке – обрабатывает данные о полученной ошибке;
- формирование отчета об успеваемости – выводит пользователю данные об успеваемости.

3.2. Организация технологии сбора, передачи, обработки и выдачи информации

В системе использованы PHP, СУБД PhpMyAdmin и язык запросов SQL, которые обеспечивают работу технологий сбора, передачи, обработки и выдачи информации.

Для организации логичного проектирования системы была использована схема разделения данных MVC. Такой шаблон проектирования позволяет отделить бизнес-логику от визуального представления системы. MVC

расшифровывается как: Model (Модель), View (Представление), Controller (Контроллер) – это основные компоненты схемы.

Компонент «Модель» получает запросы от приложения о его состоянии через «Представление», после чего изменяет его по инструкции, полученной из «Контроллера».

«Представление» отвечает за вывод информации на экран и получение данных от пользователя.

«Контроллер» обрабатывает действия пользователя и отправляет инструкции модели или представлению по их изменению. [29]

Рисунок 3.2 содержит схему взаимодействия компонентов в шаблоне MVC.

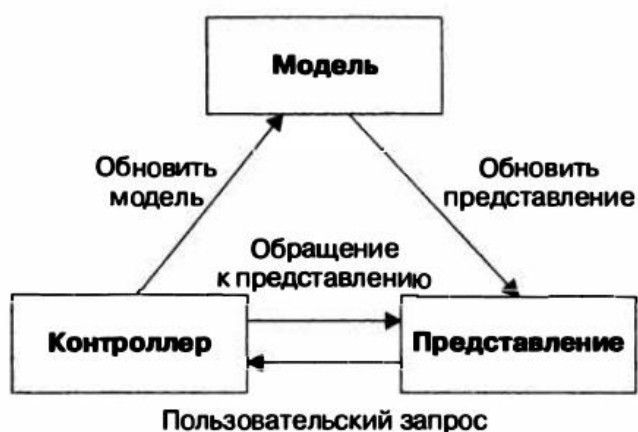


Рисунок 3.2 – Схема работы шаблона MVC

Работа шаблона контролируется файлом-роутером, содержащим в себе функцию по определению вызова необходимого контроллера, функции в нем и передачи параметров из строки запроса. Возможные варианты содержимого адресной строки содержатся в подключаемом файле с маршрутами, который показан на листинге 3.1.

Листинг 3.1 – Файл с маршрутами

```
return array(  
    'alltest/([0-9]+)course' => 'test/runAll/$1',  
    'runtest/theme([0-9]+)' => 'test/run/$1',  
    'test/([0-9]+)/theme([0-9]+)' => 'test/view/$1/$2',  
    'warning([0-9]+)' => 'test/warning/$1',  
);
```

Левая часть массива маршрутов содержит возможный текст адресной строки, а правая – инструкции для роутера, который пользуется шаблоном: контроллер/экшн/параметр. Экшн – функция, которая находится внутри контроллера и может принимать переменные из адресной строки, а также вызывать модели. Пример файла контроллера и вызываемого экшена представлен на листинге 3.2.

Листинг 3.2 – Файл-контроллер

```
<?php  
  
include_once ROOT. '/models/Blog.php';  
include_once ROOT. '/models/Course.php';  
include_once ROOT. '/models/User.php';  
  
class CabinetController  
{  
    public function actionIndex()  
    {  
        $id_user=$_SESSION["user"];  
        $blogList=array();  
        $blogList=Blog::getBlogList();  
        $courseList=array();  
        $courseList=Course::getCourseList();  
  
        require_once(ROOT . '/views/cabinet/index.php');  
    }  
}
```

Модель, вызываемая из экшена, требует подключение файла, который содержит название класса модели с функциями. В конце функции вызывается представление, которое отвечает за вывод данных на страницу.

При помощи модели происходит подключение базы данных через PDO – расширение языка PHP. Оно обеспечивает защиту от sql-инъекций, которые могут произойти в результате приема данных от пользователя через метод POST и GET посредством работы с формами. Для этого функция PDO

подготавливает sql-запрос, после чего подставляет параметры, если требуется и выполняет запрос. Параметры для подключения к базе данных хранятся в файле, представленном на листинге 3.3.

Листинг 3.3 – Массив параметров для подключения БД

```
<?php

// Массив с параметрами подключения к базе данных
return array(
    'host' => '172.23.64.64',
    'dbname' => 'db_14759954',
    'user' => '14759954',
    'password' => 'Onlybsu1!',
);
```

Параметры запрашиваются в модели, работающей с базой данных, после чего происходит их подстановка в запрос на подключение. Функция подключения отображена на листинге 3.4.

Листинг 3.4 – Функция подключения БД

```
<?php

class Db
{
    public static function getConnection()
    {
        $paramsPath = ROOT . '/config/db_params.php';
        $params = include($paramsPath);

        $dsn = "mysql:host={$params['host']};dbname={$params['dbname']};charset=cp1251";
        $db = new PDO($dsn, $params['user'], $params['password']);

        return $db;
    }
}
```

Для работы с данными, введенными пользователем на страницах, отображаемых представлением используются формы, которые содержат в себе поля для ввода. Обработка формы происходит в контроллере через метод POST, позволяющий передавать данные, не отображая их в адресной строке. После передачи данные хранятся в глобальной переменной и могут быть вызваны при обращении к ней.

Для ввода ответов на вопросы в системе используется разделение вопросов на типы: с одним ответом, множественным выбором и вводом вручную. Для первых двух требуется выводить поле формы типа select со значением radio или checkbox соответственно. После того как нажата кнопка отправки формы – в переменную POST записываются значения выбранных переключателей в виде логических данных.

При авторизации на сайте используется метод сессий. После ввода данных в форму, они проверяются на правильность. Если пользователь с такой парой логина и пароля существует, то в глобальную переменную SESSION помещается идентификатор пользователя, который в дальнейшем используется на всех страницах. [15, 23, 26,28]

3.3 Описание контрольного примера реализации проекта

Для начала работы с системой – требуется пройти авторизацию, введя логин и пароль, выданный администратором. Окно авторизации представлено на рисунке 3.3.

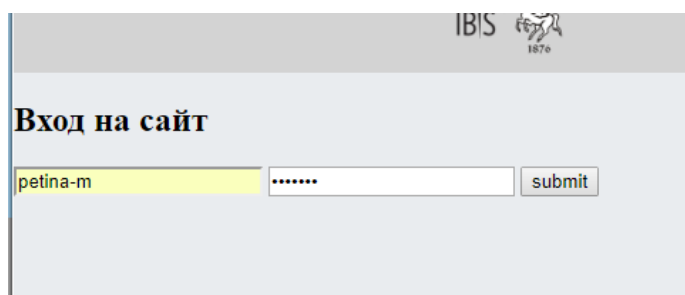


Рисунок 3.3 – Окно авторизации

После успешной авторизации пользователь попадает в кабинет, который имеет меню, представленное на рисунке 3.4.

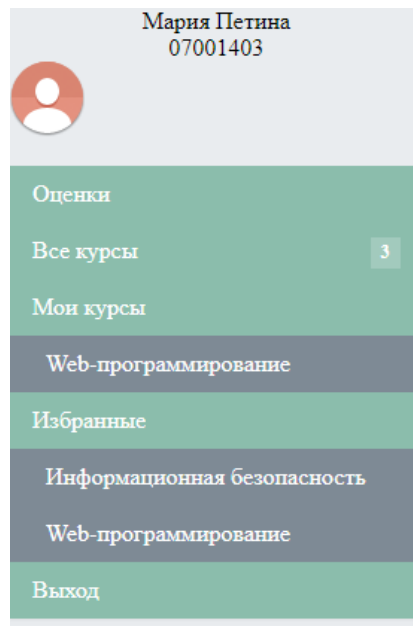


Рисунок 3.4 – Меню сайта

Меню имеет 5 пунктов. В оценках содержится информация об оценках студентов, переход на все курсы содержит в себе список всех доступных курсов на сайте. Страница всех курсов представлена на рисунке 3.5.

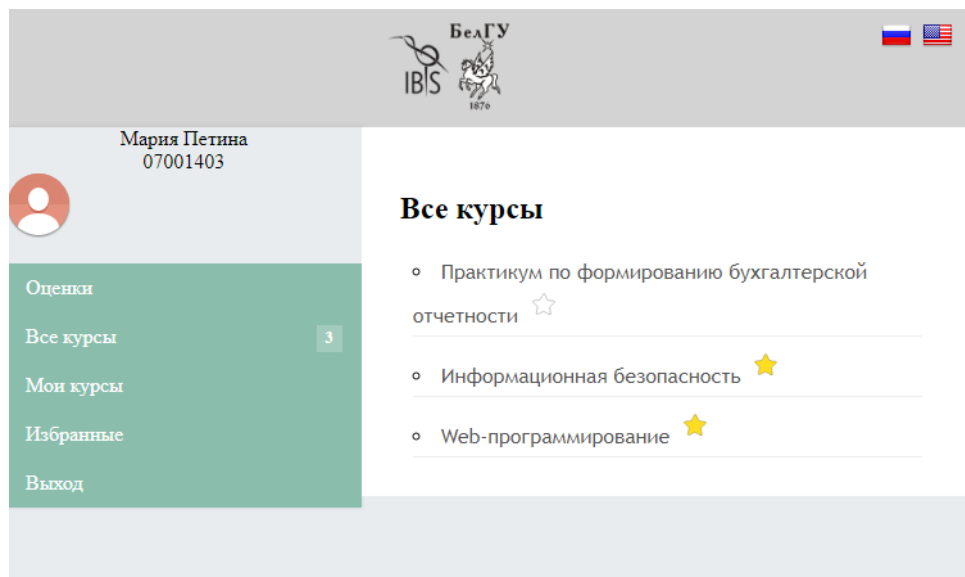


Рисунок 3.5 – Окно страницы всех курсов

Пункты «Мои курсы» и «Избранное» представляют собой выпадающие меню со списком курсов, привязанных к пользователю и курсы, находящиеся в избранном, соответственно. Для того, чтобы добавить или удалить курс из

избранного – требуется нажать на иконку возле названия, изображенную на рисунке 3.6.

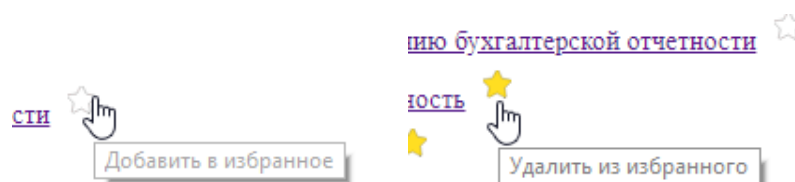


Рисунок 3.6 – Работа кнопки добавления в избранное

После перехода на страницу курса отображается список, в котором можно проводить работу с темами. Данная страница представлена на рисунке 3.7.

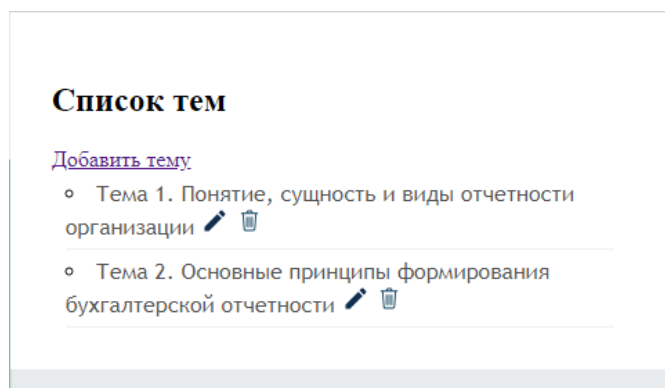


Рисунок 3.7– Окно страницы курса

На странице курса имеется 3 кнопки. Кнопка для добавления темы, нажатие на которую позволяет перейти на страницу добавления темы, изображенную на рисунке 3.8.

Добавить тему

Название темы:

Рисунок 3.8 – Окно страницы добавления темы

Форма добавления содержит поле для ввода текста и кнопку создания. Кнопка с иконкой карандаша возле названия темы позволяет перейти на страницу редактирования темы, которая представлена на рисунке 3.9.

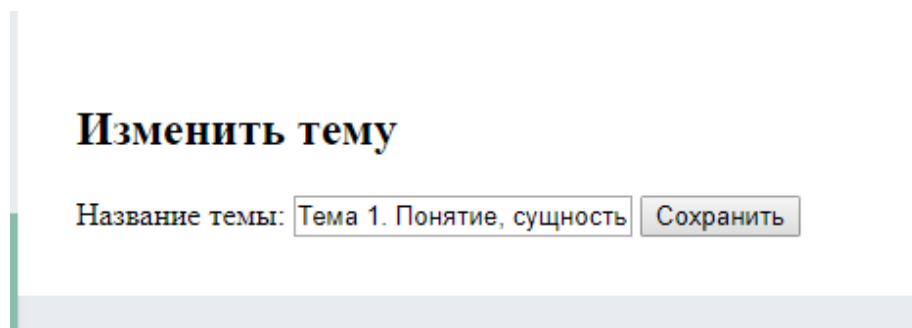


Рисунок 3.9 – Окно страницы изменения темы

Форма изменения содержит поле с существующим текстом темы, доступным для изменения и кнопка сохранения. Кнопка удаления возле названия темы позволяет сделать переход на страницу для удаления темы, изображенную на рисунке 3.10.

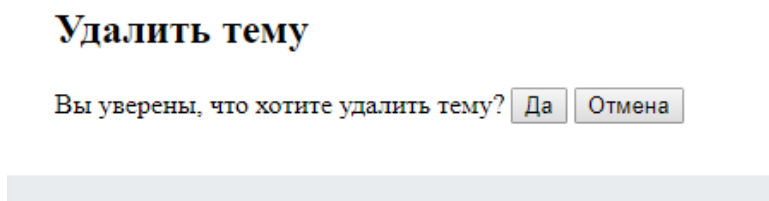


Рисунок 3.10 – Окно страницы удаления темы

При переходе на название темы – открывается страница темы. Данная страница, представленная на рисунке 3.11, содержит список вопросов.

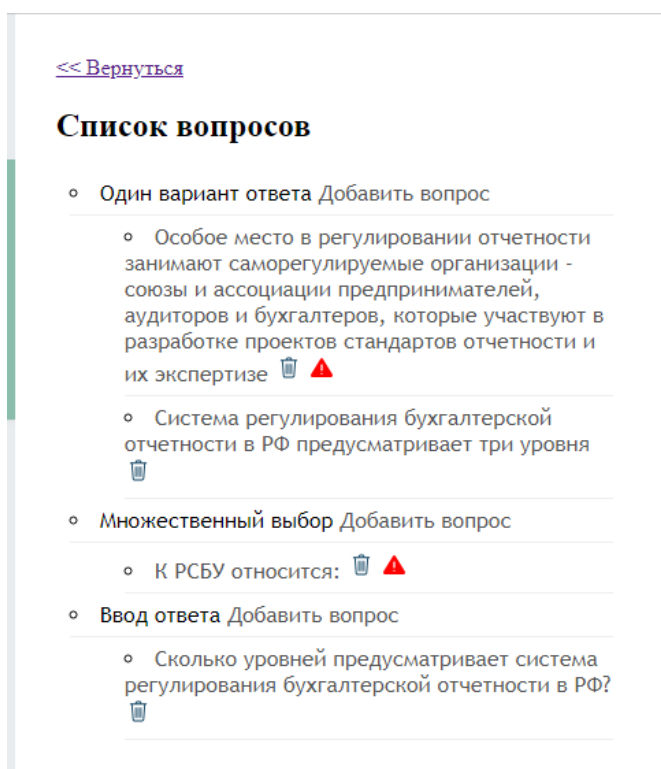


Рисунок 3.11 – Окно страницы темы

Список вопросов разделен на типы вопросов. Это позволяет совершать удобную навигацию по вопросам. При нажатии на кнопку добавления вопроса пользователь переходит на страницу добавления вопроса, изображенную на рисунке 3.12.

Добавить вопрос

Количество ответов:

Текст вопроса:

Введите текст...

Рисунок 3.12 – Окно страницы добавления вопроса

Страница содержит две формы. Первая позволяет ввести количество ответов для вопроса. После этого во второй форме появляется список с полями

для ответа. Так как вопрос может быть разного типа, то список переключателей может отличаться. Различные переключатели представлены на рисунке 3.13.

Добавить вопрос

Количество ответов:

Текст вопроса:

Ответ: Правильный:

Ответ: Правильный:

Ответ: Правильный:

Ответ: Правильный:

Добавить вопрос

Количество ответов:

Текст вопроса:

Ответ: Правильный:

Ответ: Правильный:

Ответ: Правильный:

Рисунок 3.13 – Разновидность типов вопросов

Для вопросов со вводом ответа добавляются поля для записи правильных вариантов ответа, которые представлены на рисунке 3.14.

Добавить вопрос

Количество ответов:

Текст вопроса:

Ответ:

Ответ:

Ответ:

Рисунок 3.14 – Вопрос с ручным вводом ответа

Для удобства работы с обратной связью, на странице темы, возле вопросов расположена иконка с пометкой об ошибке в вопросе. Вид иконки представлен на рисунке 3.15.


-
- Особое место в регулировании отчетности занимают предпринимателей, аудиторов и бухгалтеров, которые у и их экспертизе 
 - Система регулирования бухгалтерской отчетности в F
- Ошибка в вопросе

Рисунок 3.15 – Пометка об ошибке в вопросе

При нажатии на вопрос открывается страница изменения вопроса, содержащая форму для редактирования текста вопроса, текста ответов и выбор правильного. Вид страницы изображен на рисунке 3.16.

Изменить вопрос

Текст вопроса:

Особое место в регулировании отчетности занимают саморегулируемые организации - союзы и ассоциации предпринимателей, аудиторов и бухгалтеров, которые участвуют в разработке проектов стандартов отчетности и их экспертизе

Ответ: Верно

Ответ: Неверно

Submit

Рисунок 3.16 – Окно страницы изменения вопроса

Для перехода на страницу удаления вопроса, представленную на рисунке 3.17, требуется нажать на иконку с изображением урны, расположенную возле вопроса.

Удалить вопрос

Вы уверены, что хотите удалить вопрос?

Рисунок 3.17 – Окно страницы удаления вопроса

При переходе в меню оценок пользователь попадает на страницу выбора курса, представленную на рисунке 3.18.

Оценки студентов

- [Практикум по формированию бухгалтерской отчетности](#) ☆
- [Информационная безопасность](#) ★
- [Web-программирование](#) ★

Рисунок 3.18 – Окно страницы выбора курса студентов

На странице расположена форма, содержащая выпадающий список. В нем содержится номер группы и имя студента. Данный список изображен на рисунке 3.19.

Оценки студентов

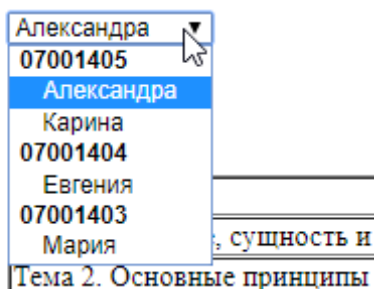


Рисунок 3.19 – Просмотр доступных студентов

После того, как пользователь выбрал студента, он получает список оценок студента по темам. Выбирается наибольшая оценка, набранная студентом. Визуальное представление списка изображено на рисунке 3.20.

Оценки студентов

Александра ▾

Успеваемость студентов

Тема	Дата и время	Балл
Тема 1. Понятие, сущность и виды отчетности организации	2018-05-21 00:24:09	95
Тема 2. Основные принципы формирования бухгалтерской отчетности	2018-05-24 03:53:16	57
Итоговое тестирование	2018-05-21 01:20:10	83.3333

Рисунок 3.20 – Окно страницы визуального представления оценок

При нажатии на кнопку «Выход» пользователь попадает на страницу авторизации, где можно снова ввести данные и войти на страницу под другим логином. [16-18]

3.4 Обоснование экономической эффективности разработки

Для того, чтобы дать оценку эффективности разработки, требуется провести расчет затрат на разработку модуля фонда тестовых заданий, а также оценить временную эффективность системы.

Для расчета затрат была использована программа для управления ресурсами и задачами проекта – Microsoft Project. Она позволяет внести задачи, которые должны быть выполнены в ходе разработки проекта, и требуемый промежуток времени, а также ресурсы, используемые каждой задачей. После этого производится автоматический подсчет затраченных ресурсов.

При разработке информационной системы был создан календарный план, дающий возможность оценить продуктивность и степень готовности системы ко вводу в эксплуатацию. Создание информационных систем построено на следующих этапах:

- обоснование необходимости создания АС и изучение предметной области;
- формирование требований к АС;
- разработка и утверждение технического задания;
- разработка проектных решений;
- разработка документации на АИС;
- разработка документации на поставку комплектующих изделия;
- адаптация АИС и её частей;
- подготовка объекта автоматизации;
- подготовка персонала;
- комплектация АИС;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

На основе этих этапов были созданы задачи в Microsoft Project. После этого, каждой задаче был назначен срок. Список задач проекта представлен на таблице 3.1.

Таблица 3.1 – Список задач проекта

▲ Разработка системы информационного сопровождения процесса тестирования. Модуль фонда тестовых заданий	3 204 ч	120 дней	Пн 15/01/18	Пт 08/06/18
▷ Обоснование необходимости создания АС и изучение предметной области	120 ч	5 дней	Пн 15/01/18	Пт 19/01/18
▷ Формирования требований к АС	64 ч	4 дней	Сб 20/01/18	Чт 25/01/18
▷ Разработка и утверждение ТЗ	24 ч	3 дней	Чт 25/01/18	Вт 30/01/18
▷ Разработка модуля фонда тестовых заданий	1 120 ч	20 дней	Вт 30/01/18	Пт 23/02/18
▷ Разработка рабочей документации	240 ч	15 дней	Пт 23/02/18	Ср 14/03/18
▷ Адаптация АИС и её частей	288 ч	12 дней	Ср 14/03/18	Чт 29/03/18
▷ Подготовка объекта автоматизации к вводу АС в действие	128 ч	8 дней	Чт 29/03/18	Пн 09/04/18
▷ Подготовка персонала	280 ч	10 дней	Пн 09/04/18	Сб 21/04/18
▷ Комплектация АИС	32 ч	4 дней	Сб 21/04/18	Чт 26/04/18
▷ Пусконаладочные работы	224 ч	7 дней	Пт 27/04/18	Сб 05/05/18
▷ Проведение предварительных испытаний	324 ч	13,5 дней	Сб 05/05/18	Пн 21/05/18
▷ Проведение опытной эксплуатации	240 ч	10 дней	Пн 21/05/18	Сб 02/06/18
▷ Проведение приемочных испытаний	120 ч	5 дней	Сб 02/06/18	Пт 08/06/18

Исходя из данных, выполнение проекта займет 120 дней. Далее для расчета экономической составляющей разработки были внесены ресурсы, используемые во время работы над проектом. Список ресурсов представлен на таблице 3.2.

Таблица 3.2 – Список ресурсов

Разработчики	Трудовой	100%	200,00р./ч
Компьютеры	Трудовой	100%	30,00р./ч
Данные учебных планов	Трудовой	100%	2,00р./ч
Данные карт компетенций	Трудовой	100%	2,00р./ч
Данные ФГОС	Трудовой	100%	2,00р./ч
Проект интерфейса	Трудовой	100%	10,00р./ч
База данных	Трудовой	100%	10,00р./ч
Преподаватели	Трудовой	100%	100,00р./ч
CD диск	Материальный		40,00р.
Переплет	Материальный		200,00р.
Печать	Материальный		60,00р.

После ресурсы были распределены по задачам. Задачи и используемые ими ресурсы представлены в таблице 3.3.

Таблица 3.3 – Таблица ресурсов и задач

Название	Названия ресурсов
Обоснование необходимости создания АС и изучение предметной области	Разработчики; Преподаватели
Формирования требований к АС	Разработчики; Преподаватели
Разработка и утверждение ТЗ	Разработчики
Разработка модуля тестирования и обработки данных по результатам тестирования	База данных; Данные карт компетенций; Данные учебных планов; Данные ФГОС; Компьютеры; Проект интерфейса; Разработчики
Разработка рабочей документации	Разработчики; Компьютеры
Адаптация АИС и её частей	Разработчики; База данных; Компьютеры
Подготовка объекта автоматизации к вводу АС в действие	Разработчики; Компьютеры
Подготовка персонала	Разработчики; Проект интерфейса; База данных; Компьютеры; Преподаватели
Комплектация АИС	Печать; Переплет; CD диск; Разработчики
Пусконаладочные работы	База данных; Разработчики; Компьютеры; Проект интерфейса
Проведение предварительных испытаний	База данных; Преподаватели; Компьютеры
Проведение опытной эксплуатации	База данных; Преподаватели; Компьютеры
Проведение приемочных испытаний	База данных; Компьютеры; Преподаватели

Так как каждый ресурс имеет стандартную ставку в час, либо за единицу, то для расчета итогового значения стоимости разработки требуется

умножить ресурсы на количество времени для задачи, к которой они привязаны и сложить получившиеся результаты. Общие затраты на проект составили 176 240 рублей. Затраты на каждый этап разработки и итоговая стоимость представлены на таблице 3.4.

Таблица 3.4 – Поэтапные затраты

4 Разработка системы информационного сопровождения процесса тестирования. Модуль фонда тестовых заданий	176 240,00р.
Обоснование необходимости создания А	12 000,00р.
Формирования требований к АС	6 400,00р.
Разработка и утверждение ТЗ	4 800,00р.
Разработка модуля фонда тестовых заданий	40 960,00р.
Разработка рабочей документации	27 600,00р.
Адаптация АИС и её частей	23 040,00р.
Подготовка объекта автоматизации к вводу АС в действие	14 720,00р.
Подготовка персонала	11 200,00р.
Комплектация АИС	6 400,00р.
Пусконаладочные работы	14 000,00р.
Проведение предварительных испытаний	4 320,00р.
Проведение опытной эксплуатации	3 200,00р.
Проведение приемочных испытаний	1 600,00р.

Таким образом, длительность проекта составляет 120 дней, а общая стоимость – 176 240 рублей.

Для расчета эффективности разработки по временным показателям были проведены опытные испытания для определения времени, затрачиваемого на поиск вопросов в фонде тестовых заданий и создание новых вопросов в существующей системе и в системе информационного сопровождения тестирования. Результаты сравнения представлены в таблице 3.5.

Таблица 3.5 – Результаты сравнения временных затрат

Название процесса	Затраченное время в существующей системе (сек.)	Затраченное время в разработанной системе (сек.)
Поиск вопроса, требующего исправления	15	5
Работа с интерфейсом	7	5
Выбор типа вопроса	13	6
Заполнение полей текста	45	30

Таким образом, поиск одного вопроса в разработанной системе займет 5 секунд, вместо 15 секунд в существующей системе, а суммарное время создания вопроса – 41 секунда и 65 секунд, соответственно.

Так как продукт разработан для внутреннего пользования, то оценить эффективность довольно сложно, как и вычислить ожидаемую прибыль. [32]

ЗАКЛЮЧЕНИЕ

В процессе выполнения выпускной квалификационной работы изучена структура НИУ «БелГУ», а также работа отдела информатизации, определено назначение создания системы, изучены существующие разработки, выявлены их узкие места. После чего был сформирован список требований к создаваемому модулю фонда тестовых заданий. Для разработки данного модуля были выбраны программные средства, позволяющие достигнуть поставленной цели.

В ходе выполнения работы была достигнута цель по совершенствованию фонда тестовых заданий путем модернизации и был создан модуль фонда тестовых заданий системы информационного сопровождения процесса тестирования.

Созданная разработка может быть использована как в НИУ «БелГУ», так и в других учебных заведениях. Система является универсальной, с легкозаменяемыми данными. Работа с модулем интуитивно понятна и не требует предварительной подготовки и обучения.

Разработанный модуль значительно ускоряет время работы с фондом тестовых заданий. На внесение изменений в систему затрачивается в два раза меньше времени, чем в существующей системе, что повышает эффективность работы преподавателя.

Система и модуль используют подключение к локальной сети, что дает возможность работать с ними при отсутствии соединения с глобальной сетью.

Интерфейс системы состоит из блоков, которые имеют удобное расположение и позволяют перейти на необходимые страницы, выполнив минимальное количество действий. Также интерфейс имеет возможность перехода на основной сайт НИУ «БелГУ».

Для достижения цели были поставлены следующие задачи:

- изучение предметной области и построение исходной модели «как есть»;
- разработка модели автоматизации тестирования «как должно быть»;
- проектирование модуля хранения тестовых заданий;
- программная реализация модуля фонда тестовых заданий;
- расчет эффективности модуля.

Поставленные цели и задачи выпускной квалификационной работы выполнены в полном объеме.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) Управление данными [Электронный ресурс] – Режим доступа: <http://s-nov.narod.ru/07.Upravlenie/07.Upravlenie.htm> (дата обращения 15.05.18)
- 2) DFD [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/DFD> (дата обращения 15.05.18)
- 3) IDEF3 [Электронный ресурс] – Режим доступа: <https://ru.wikipedia.org/wiki/IDEF3> (дата обращения 15.05.18)
- 4) НИУ «БелГУ» [Электронный ресурс] – Режим доступа: <http://bsu.edu.ru/> (дата обращения 15.05.18)
- 5) Стандарт функционального моделирования IDEF0 [Электронный ресурс] – Режим доступа: http://studopedia.ru/9_182395_standart-funktsionalnogo-modelirovaniya-IDEF.html (дата обращения 15.05.18)
- 6) INDIGO – Программа для создания тестов и онлайн тестирования [Электронный ресурс] – Режим доступа: <https://indigotech.ru/> (дата обращения 15.05.18)
- 7) UniTest [Электронный ресурс] – Режим доступа: <https://sight2k.com/rus/unitest/qtour/> (дата обращения 15.05.18)
- 8) Калашян, А.Н. Структурные модели бизнеса: DFD-технологии [Текст] / А.Н. Калашян. – Москва: Прикладные информационные технологии, 2009 – 249 с.
- 9) Гагарина, Л.Г. Технология разработки программного обеспечения [Текст] / Л.Г. Гагарина, Е.В. Кокорева – Москва: Форум, 2016 – 401 с.
- 10) Силич, В.А. Моделирование и анализ бизнес-процессов [Текст] / В.А. Силич, М.П. Силич. – Москва: Перспектива, 2013 – 385 с.
- 11) Брауде, Э.Дж. Технология разработки программного обеспечения [Текст] / Э.Дж. Брауде. – Санкт-Петербург: Издательский дом Питер, 2014 – 655 с.

- 12) Маторин, С.И. Теория систем и системный анализ. Учебное пособие [Текст] / С.И. Маторин, О.А. Зимовец – Белгород: Белгородский государственный университет, 2012 – 288 с.
- 13) Симонович, С.В. Информатика. Базовый курс. Учебник для Вузов [Текст] / С.В. Симонович. – Санкт-Петербург: Издательский дом Питер, 2015 – 640 с.
- 14) Орлов, С. Технологии разработки программного обеспечения [Текст] / С.А. Орлов. – Санкт-Петербург: Издательский дом Питер, 2013 – 464 с.
- 15) Форта, Б. Освой самостоятельно SQL за 10 минут [Текст] / Б. Форта. – Москва: Издательский дом Вильямс, 2017 – 288 с.
- 16) Макфарланд, Д.С. Большая книга CSS3 [Текст] / Д.С. Макфарланд. – Санкт-Петербург: Издательский дом Питер, 2016 – 608 с.
- 17) Фельке-Моррис, Т. Большая книга веб-дизайна [Текст] / Т. Фельке-Моррис. – Москва: Эксмо, 2013 – 608с.
- 18) Роббинс, Дж. Н. HTML5, CSS3 и JavaScript. Исчерпывающее руководство [Текст] / Дж. Н. Роббинс. – Москва: Эксмо, 2014 – 528 с.
- 19) Apache HTTP Server [Электронный ресурс] – <https://httpd.apache.org/>
(дата обращения 15.05.18)
- 20) Sublime Text [Электронный ресурс] – <https://www.sublimetext.com/>
(дата обращения 15.05.18)
- 21) PhpMyAdmin [Электронный ресурс] – <https://www.phpmyadmin.net/>
(дата обращения 15.05.18)
- 22) Коннолли, Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика [Текст] / Т. Коннолли, К. Бегг. – Москва: Издательский дом Вильямс, 2017 – 1440 с.
- 23) Грабер, М. SQL для простых смертных [Текст] / М. Грабер. – Москва: Лори, 2014 – 378 с.
- 24) Арнольд, М. Администрирование Apache [Текст] / М. Арнольд, Дж. Д. Алмейда, К. Миллер. – Москва: Лори, 2016 – 418 с.

- 25) Антонов, А. Системный анализ. Учебник [Текст] / А. Антонов. – Москва: Инфра-М, 2017 – 368 с.
- 26) Зандстра, М. PHP. Объекты, шаблоны и методики программирования [Текст] / М. Зандстра. – Москва: Издательский дом Вильямс, 2016 – 576 с.
- 27) Чистов, Д.В. Проектирование информационных систем. Учебник и практикум. [Текст] / Д.В. Чистов – Москва: Юрайт, 2016. – 258 с.
- 28) Никсон, Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 [Текст] / Р. Никсон. – Санкт-Петербург: Издательский дом Питер, 2017 – 768 с.
- 29) Чедвик, Дж. ASP.NET MVC 4: разработка реальных веб-приложений с помощью ASP.NET MVC [Текст] / Дж. Чедвик, Т. Снайдер, Х. Панда. – Москва: Вильямс, 2013 – 432 с.
- 30) Управление электронных образовательных технологий [Электронный ресурс] – Режим доступа: <https://www.bsu.edu.ru/bsu/structure/> (дата обращения 15.05.18)
- 31) PikaTest [Электронный ресурс] – Режим доступа: <http://pedsovet.su/forum/78-5830-1> (дата обращения 15.05.18)
- 32) Куперштейн, В. Microsoft Project 2013 в управлении проектами [Текст] / В. Куперштейн. – Санкт-Петербург: БХВ-Петербург, 2013 – 432 с.

ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ А

Исходный код модуля подключения к локальной базе данных

```
<?php
return array(
    'host' => '172.23.64.64',
    'dbname' => 'db_14759954',
    'user' => '14759954',
    'password' => 'Onlybsu1!',
);
<?php

class Db
{
    public static function getConnection()
    {
        $paramsPath = ROOT . '/config/db_params.php';
        $params = include($paramsPath);

        $dsn
        "mysql:host={$params['host']};dbname={$params['dbname']};charset=cp1251";
        $db = new PDO($dsn, $params['user'], $params['password']);

        return $db;
    }
}
```

ПРИЛОЖЕНИЕ Б

Исходный код модуля кабинета пользователя

```
<?php
include_once ROOT. '/models/Blog.php';
include_once ROOT. '/models/Course.php';
include_once ROOT. '/models/User.php';
class CabinetController
{
    public function actionIndex()
    {
        $id_user=$_SESSION["user"];
        $blogList=array();
        $blogList=Blog::getBlogList();
        $courseList=array();
        $courseList=Course::getCourseList();
        require_once(ROOT . '/views/cabinet/index.php');
    }
    public function actionLogout()
    {
        session_start();
        unset($_SESSION["user"]);
        require_once(ROOT . '/views/user/login.php');
    }
    public function actionCourseList()
    {
        $id_user=$_SESSION["user"];
        $hname="Все курсы";
        $courseList=array();
    }
}
```



```

$courseList=Course::getCourseList();
User::getGroup($id_user);
require_once(ROOT . '/views/cabinet/courselist.php');
}
public function actionAddFave($course_id)
{
    $id_user=$_SESSION["user"];
    Course::addFave($course_id,$id_user);
    header('Location: /allcourses');
}

public function actionDelFave($course_id)
{
    $id_user=$_SESSION["user"];
    Course::delFave($course_id,$id_user);
    header('Location: /allcourses');
}

public function actionStats()
{
    $hname="Успеваемость";
    $themeStats=array();
    $id_user=$_SESSION["user"];
    $user=User::getUserById($id_user);
    if ($user["type_id"]==2:
        $hname="Оценки студентов";
        $courseList=Course::getCourseList();
        if (isset($_POST['submit'])) {
            $chosenStats=Course::getStatsList($theme_id);
            echo "<pre>";
            print_r($chosenStats);echo "</pre>";
        }
    }
}

```

```

        header('Location: /stats');
    }

    endif;

    $disciplineStats=Cabinet::disciplineStats($id_user);
    require_once(ROOT . '/views/cabinet/stats.php');
}

public function actionCoursestats($id_discipline)
{
    $id_user=$_SESSION["user"];
    $user=User::getUserById($id_user);
    $themeList=array();
    $hname="Оценки студентов";
    $grup=Course::getGroups();
    $disciplineStats=array();
    if (isset($_POST['Submit'])) {
        $stud_id=$_POST['student'];
        $themeList=Course::getThemeList($id_discipline);
        $disciplineStats=Course::getMaxDisciplineScore($id_discipline,$stud_id);
    }
    require_once(ROOT . '/views/cabinet/coursestats.php');
}
}

```

ПРИЛОЖЕНИЕ В

Исходный код модуля работы с темой

```
<?php
include_once ROOT. '/models/Course.php';
include_once ROOT. '/models/Theme.php';
include_once ROOT. '/models/User.php';
class ThemeController
{
    public function actionView($discip_id,$theme_id)
    {
        $id_user=$_SESSION["user"];
        $user=User::getUserById($id_user);
        $hname="Список вопросов";
        $typeList=array();
        $typeList=Theme::getTypeList();
        $theme=$theme_id;
        $discip=$discip_id;
        Theme::isError(2);
        require_once(ROOT . '/views/theme/view.php');
    }
    public function actionAdd($discip_id)
    {
        $hname="Добавить тему";
        $question="";
        if (isset($_POST['Submit'])) {
            Theme::createTheme($discip_id);
        }
    }
}
```

```

        require_once(ROOT . '/views/theme/add.php');
    }
    public function actionDelete($theme_id, $discip_id)
    {
        $hname="УДАЛИТЬ ТЕМУ";
        $question="";
        if (isset($_POST['Submit'])) {
            Theme::deleteTheme($theme_id);
            header('Location: /course-'. $discip_id);
        }

        if (isset($_POST['Cancel'])) {
            header('Location: /course-'. $discip_id);
        }
        require_once(ROOT . '/views/theme/delete.php');
    }
    public function actionEdit($theme_id)
    {
        $hname="ИЗМЕНИТЬ ТЕМУ";
        $theme="";
        $isUpdate=0;
        $theme=Theme::getTheme($theme_id);
        if (isset($_POST['Submit'])) {
            $isUpdate=Theme::updateTheme($theme_id);
        }
        require_once(ROOT . '/views/theme/edit.php');
    }

```

```

    }
}
<?php
class Theme
{
    public static function getTypelist ()
    {
        $db = Db::getConnection();
        $typeList=array();
        $result=$db->query('SELECT * FROM type_question');
        $i=0;
        while ($row=$result->fetch()) {
            $typeList[$i]['id']=$row['id'];
            $typeList[$i]['name']=$row['name'];
            $i++;
        }
        return $typeList;
    }
    public static function getQuestionlist ($id_theme, $id_type)
    {
        $db = Db::getConnection();
        $questionList=array();
        $result = $db->prepare('SELECT * FROM question where
id_theme=:id_theme AND id_type=:id_type');
        $result->bindParam(':id_theme', $id_theme, PDO::PARAM_INT);
        $result->bindParam(':id_type', $id_type, PDO::PARAM_INT);
        $result->execute();

        $i=0;
        while ($row=$result->fetch()) {

```

```

        QuestionList[$i]['id']=$row['id'];
        QuestionList[$i]['text_quest']=$row['text_quest'];
        $i++;
    }
    return QuestionList;
}

public static function createTheme ($discip_id)
{
    $theme = $_POST['name'];
    $db = Db::getConnection();
    $result = $db->prepare('INSERT INTO theme SET name=:name,
id_disc=:id_disc');
    $result->bindParam(':name', $theme, PDO::PARAM_STR);
    $result->bindParam(':id_disc', $discip_id, PDO::PARAM_INT);
    $result->execute();
}

public static function getTheme ($theme_id)
{
    $theme=array();
    $db = Db::getConnection();
    $result = $db->prepare('SELECT * FROM theme where id=:id');
    $result->bindParam(':id', $theme_id, PDO::PARAM_INT);
    $result->execute();

    $i=0;
    while ($row=$result->fetch()) {
        $theme['id']=$row['id'];
        $theme['name']=$row['name'];
        $i++;
    }
}

```

```

        return $theme;
    }

    public static function updateTheme ($theme_id)
    {
        $theme = $_POST['name'];
        $db = Db::getConnection();
        $result = $db->prepare('UPDATE theme SET name=:name
WHERE id=:theme_id');
        $result->bindParam(':name', $theme, PDO::PARAM_STR);
        $result->bindParam(':theme_id', $theme_id,
PDO::PARAM_STR);
        $result->execute();
    }

    public static function deleteTheme ($theme_id)
    {
        $theme_id=$_POST['id'];

        $db = Db::getConnection();
        $result = $db->prepare('DELETE FROM theme WHERE
id=:theme_id');
        $result->bindParam(':theme_id', $theme_id, PDO::PARAM_STR);
        $result->execute();

    }

    public static function isError ($question_id)
    {
        $db = Db::getConnection();

```

```

        $result = $db->prepare('SELECT * FROM errors WHERE
question_id=:question_id');
        $result->bindParam(':question_id',           $question_id,
PDO::PARAM_INT);
        $result->execute();
        $error=$result->fetch();
        return $error;
    }
}

```

```

<?php include ROOT . '/layouts/header.php'; ?>
<?php include ROOT . '/layouts/sidebar.php'; ?>

```

```

<div class="content">
    <div class="course-list">
        <a href="/course-<?php echo $discip; ?>"><< Вернуться</a>
        <h2><?php echo $hname; ?></h2>
        <div id="blog-post">
            <ul class="category-list">
                <?php foreach ($typeList as $typeItem):?>
                    <li><?php echo $typeItem['name']; ?></span> <a
href="/question/add/theme-<?php echo $theme; ?>/<?php echo
$typeItem['id']; ?>">Добавить вопрос</a> </li>
                    <?php $id_type=$typeItem['id'];
                    $questionList=array();
                    $error="";
                    $questionList=Theme::getQuestionlist($theme, $id_type);
                    ?>
                <ul class="sublist">
                    <?php foreach ($questionList as $questionItem):?>

```



```

                <li><a href="/question-<?php echo
$questionItem['id']; ?>/edit/theme-<?php echo $theme;?>"><?php echo
$questionItem['text_quest']; ?></a>
                <a href="/question/del/theme-
<?php echo $theme;?>/<?php echo $questionItem['id'];?>/course<?php echo
$discip; ?>">
                </a>
                <?php
$error=Theme::isError($questionItem['id']); if($error):?>
                </a>
                <?php endif;?>
                </li>
                <?php endforeach; ?>
            </ul>
        <?php endforeach; ?>
    </div>
</div>
</div>
<?php include ROOT . '/layouts/footer.php'; ?>

```

ПРИЛОЖЕНИЕ Г

Исходный код модуля работы с вопросами

```
<?php
include_once ROOT. '/models/Course.php';
class QuestionController
{
    public function actionEdit($question_id, $theme_id)
    {
        $isUpdate=0;
        $answer=Question::getAnswer($question_id);
        if (isset($_POST['Submit'])) {
            $isUpdate=Question::updateQuestion($question_id, $answer);
        }
        $hname="Изменить вопрос";
        $question=Question::getQuestion($question_id);
        $typeInput="";
        switch ($question['type']) {
            case '1':
                {$typeInput="radio";}
                break;
            case '2':
                {$typeInput="checkbox";}
                break;
            case '3':
                {$typeInput="";}
                break;
            default:
```

```

        echo "ERROR";
        break;
    }
    $answer=Question::getAnswer($question_id);
    require_once(ROOT . '/views/question/edit.php');
}
public function actionAdd($theme_id,$quest_type)
{
    $hname="Добавить вопрос";
    $typeInput="";
    switch ($quest_type) {
        case '1':
            {$typeInput="radio";}
            break;
        case '2':
            {$typeInput="checkbox";}
            break;
        case '3':
            {$typeInput="";}
            break;
        default:
            echo "ERROR";
            break;
    }
    $question="";
    $ans=array();
    $ansNum="";
    if (isset($_POST['Submit'])) {
        Question::createQuestion($theme_id,$quest_type);
    }
}

```

```

    }
    if (isset($_POST['formAns'])) {
        $ansNum=$_POST['ansNum'];
    }
    require_once(ROOT . '/views/question/add.php');
}
public function actionDelete($theme_id,$quest_id, $discip_id)
{
    $hname="УДАЛИТЬ ВОПРОС";
    if (isset($_POST['Submit'])) {
        Question::deleteQuestion($quest_id);
        header('Location: /course-'. $discip_id);

    }
    if (isset($_POST['Cancel'])) {
        header('Location: /course-'. $discip_id);
    }
    require_once(ROOT . '/views/question/delete.php');
}
}
<?php
class Question
{
    public static function createQuestion ($theme_id,$quest_type)
    {
        $text = $_POST['text'];
        print_r($text);
        $db = Db::getConnection();
        $result = $db->prepare('INSERT INTO question SET id="",
text_quest=:text, id_theme=:theme, id_type=:type');

```

```

$result->bindParam(':text', $text, PDO::PARAM_STR);
$result->bindParam(':theme', $theme_id, PDO::PARAM_INT);
$result->bindParam(':type', $quest_type, PDO::PARAM_INT);
$result->execute();
print_r($result);
$result=$db->query('SELECT max(id) FROM question');
$id_question=$result->fetch();
$i=0;
foreach ($_POST['ans'] as $ansItem) {
    $flag=0;
    $ans[$i]=$ansItem;
    $a=0;
    foreach ($_POST['isTrue'] as $isTrue) {
        if ($i==$isTrue[$a]) {
            $flag=1;
        }
        $result = $db->prepare('INSERT INTO answer SET id="",
text_answer=:text, flag=:flag, id_question=:id_question');
        $result->bindParam(':text', $ans[$i], PDO::PARAM_STR);
        $result->bindParam(':flag', $flag, PDO::PARAM_INT);
        $result->bindParam(':id_question',          $id_question[0],
PDO::PARAM_INT);
        $result->execute();
    }
    $i++;
}
}
public static function getQuestion ($id)
{
    $db = Db::getConnection();

```

```

$question=array();
$result = $db->prepare('SELECT * FROM question where id=:id');
$result->bindParam(':id', $id, PDO::PARAM_INT);
$result->execute();
$i=0;
while ($row=$result->fetch()) {
    $question['id']=$row['id'];
    $question['text_quest']=$row['text_quest'];
    $question['type']=$row['id_type'];
    $i++;
}
return $question;
}

public static function getAnswer ($id_question)
{
    $db = Db::getConnection();
    $answer=array();
    $answer[0]['N']=0;
    $result = $db->prepare('SELECT * FROM answer where
id_question=:id');
    $result->bindParam(':id', $id_question, PDO::PARAM_INT);
    $result->execute();
    $i=1;
    while ($row=$result->fetch()) {
        $text="";
        $answer[$i]['id']=$row['id'];
        $answer[$i]['text_answer']=$row['text_answer'];
        if ($row['flag']==1) {
            $text="checked";
        }
    }
}

```

```

        $answer[$i]['flag']=$text;
        $i++;
        $answer[0]['N']++;
    }
    return $answer;
}

public static function updateQuestion ($question_id, $answer)
{
    $question = $_POST['text_quest'];
    $db = Db::getConnection();
    $result = $db->prepare('UPDATE question SET text_quest=:text
WHERE id=:id_question');
    $result->bindParam(':text', $question, PDO::PARAM_STR);
    $result->bindParam(':id_question', $question_id, PDO::PARAM_STR);
    $result->execute();
    $i=0;
    foreach ($_POST['ans'] as $ansItem) {
        $flag=0;
        $ans[$i]=$ansItem;
        $a=0;
        foreach ($_POST['isTrue'] as $isTrue) {
            if ($i==$isTrue[$a]) {
                $flag=1;
            }
        }
        $result = $db->prepare('UPDATE answer SET
text_answer=:text, flag=:flag WHERE id_question=:id_question AND
id=:id_answer');
        $result->bindParam(':text', $ans[$i], PDO::PARAM_STR);
        $result->bindParam(':flag', $flag, PDO::PARAM_INT);
    }
}

```

```

        $result->bindParam(':id_question',          $question_id,
PDO::PARAM_INT);

        $result->bindParam(':id_answer',          $answer[$i+1]["id"],
PDO::PARAM_INT);

        $result->execute();
    }
    $i++;
}
return true;
}

public static function deleteQuestion ($question_id)
{
    $db = Db::getConnection();
    $result = $db->prepare('DELETE FROM question WHERE
id=:question_id');
    $result->bindParam(':question_id', $question_id, PDO::PARAM_STR);
    $result->execute();
}
}

<?php include ROOT . '/layouts/header.php'; ?>
<?php include ROOT . '/layouts/sidebar.php'; ?>

<div class="content">
    <h3><?php echo $hname;?></h3>
    <form action="#" method="post">
        <label>Количество ответов:</label>
        <input type="text" name="ansNum" value="" placeholder="Введите
цифру...">
        <input type="submit" name="formAns" value="Submit" />
    </form>

```



```

<form action="#" method="post">
  <br>
  <label>Текст вопроса:</label>
  <br>
<textarea name="text" placeholder="Введите текст..." ></textarea>
  <br />
  <?php
  $N=$ansNum;
  while ($ansNum) { ?>
    <label>Ответ:</label>
    <input type="text" name="ans[]" value="" placeholder="Введите
текст...">
    <?php if ($typeInput!=NULL){ ?>
      <label>Правильный:</label> <input type="<?php echo
$typeInput;?>" name="isTrue[]" value="<?php echo ($N-$ansNum);?>" />
      <br />
    <?php } ?>
    <?php $ansNum--;} ?>
    <input type="submit" name="Submit" value="Submit" />
  </form>
</div>
<?php include ROOT . '/layouts/footer.php'; ?>

```

ПРИЛОЖЕНИЕ Д

Исходный код модуля работы с успеваемостью

```
<?php include ROOT . '/layouts/header.php'; ?>
<?php include ROOT . '/layouts/sidebar.php'; ?>
<div class="content">
<?php if ($user["type_id"]==1): ?>
    <h2><?php echo $hname; ?></h2>
<canvas id="myChart" width="100" height="30"></canvas>
<script>
var ctx = document.getElementById("myChart").getContext('2d');
var myChart = new Chart(ctx, {
    type: 'horizontalBar',
    data: {
        labels: [
            <?php foreach ($disciplineStats as $disciplineItem): ?>
                "<?php echo
                Cabinet::getDisciplineName($disciplineItem['discipline_id'])?>", <?php
endforeach;?>],
        datasets: [{
            label: 'Успеваемость по дисциплине',
            barPercentage: 10,
            data: [<?php foreach ($disciplineStats as $disciplineItem): ?>
                "<?php echo $disciplineItem['number_score']?>", <?php endforeach;?>],
            backgroundColor: [
                'rgba(255, 99, 132, 0.2)',
                'rgba(54, 162, 235, 0.2)',
                'rgba(255, 206, 86, 0.2)'
```

```

        ],
        borderColor: [
            'rgba(255,99,132,1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)'
        ],
        borderWidth: 1
    }}
},
options: {
    scales: {
        xAxes: [{
            ticks: {
                min: 0,
                max: 100
            }
        }]
    }
}
});
</script>
<?php endif; ?>
<?php if ($user["type_id"]==2): ?>
<h2><?php echo $hname; ?></h2>
<?php foreach ($courseList as $courseItem): ?>
    <li><a href="/score/<?php echo $courseItem['id']; ?>"><?php echo
$courseItem['name']; ?></a></span>
    <?php if(Course::isFave($id_user,$courseItem['id'])) {?>

```

```
<a href="/delFave/course/<?php echo $courseItem['id']; ?>">
</a></li> <?php } else {?> <a href="/addFave/course/<?php echo $courseItem['id'];
?>"> </a></li> <?php }?>
<?php endforeach; ?>
<?php endif; ?>
</div>
<?php include ROOT . '/layouts/footer.php'; ?>
```