

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

Институт математики, физики и информационных технологий

Кафедра «Прикладная математика и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

ПРИКЛАДНАЯ ИНФОРМАТИКА В СОЦИАЛЬНОЙ СФЕРЕ

БАКАЛАВРСКАЯ РАБОТА

на тему Разработка информационно-развлекательной системы для обучения программированию детей младшего школьного возраста

Студент

В.О. Ерослаев

Руководитель

Е.А. Ерофеева

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

«_____» _____ 2017 г.

Тольятти 2017

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования

«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ

Зав. кафедрой «Прикладная
математика и информатика»

_____ А.В. Очеповский

« ____ » _____ 2017 г.

ЗАДАНИЕ
на выполнение бакалаврской работы

Студент Ерослаев Валерий Олегович

1. Тема «Разработка информационно-развлекательной системы для обучения программированию детей младшего школьного возраста»

2. Срок сдачи студентом законченной выпускной квалификационной работы
_____ 2017 г.

3. Исходные данные к выпускной квалификационной работе: нормативные документы и акты, требования на разработку системы

4. Содержание выпускной квалификационной работы: анализ литературы по проблеме обучения детей программированию; определение компонентов системы и требований к её разработке; обоснование выбора средств реализации; анализ аналогов; разработка системы для обучения программированию детей младшего школьного возраста.

5. Ориентировочный перечень графического и иллюстративного материала: логическая и физическая модели данных, диаграммы и скриншоты, приложения, презентация MS PowerPoint

6. Дата выдачи задания « ____ » _____ 2017 г.

Заказчик Директор ИП Фомин

_____ А.А.Фомин

Руководитель выпускной
квалификационной работы

_____ Е.А. Ерофеева

Задание принял к исполнению

_____ В.О. Ерослаев

Аннотация

Тема: «Разработка информационно-развлекательной системы для обучения программированию детей младшего школьного возраста».

Актуальность выбора темы работы обусловлена необходимостью разработки информационно-развлекательной системы для обучения программированию детей младшего школьного возраста, которая позволит объединить все пособия по обучению программирования в одном месте и оценивать успеваемость как отдельных учеников, так и общий результат всей группы в целом.

Объем бакалаврской работы 71 страница, из которых приложения составляют 8 страниц. В работе: рисунков - 19, таблиц - 15, формул – 16, список литературы состоит из 13 источников.

Целью работы является разработка информационно-развлекательной системы для обучения программированию детей мл. школьного возраста.

Бакалаврская работа состоит из введения, трех глав, заключения и приложений. Во введении обосновывается актуальность выбранной темы, формируется проблема, определяются объект и предмет исследования, выделяются цели и задачи.

Первая глава включает в себя анализ предметной области и формирование принципа построения информационно-развлекательных систем с описанием терминов, используемых в бакалаврской работе.

Вторая глава описывает проектные модули и пути решения основных задач информационно-развлекательной системы для обучения программированию детей младших классов.

В третьей главе описывается экономический анализ себестоимости программной продукции и трудозатраты на разработку .

Заключение содержит выводы, итоги о проделанной работе и дальнейшую оценку приложения.

В Приложение включены уникальные фрагменты кода, используемые при составлении информационно-развлекательной системы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
ГЛАВА 1. АНАЛИЗ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ ДЕТЕЙ МЛАДШЕГО ШКОЛЬНОГО ВОЗРАСТА	8
1.1 Система обучения программированию детей младшего школьного возраста	8
1.2 Обоснование автоматизации информационно-развлекательной системы	13
1.3 Концептуальное моделирование деятельности ИРС для обучения программированию школьников младших классов	15
1.4 Анализ существующих разработок и обоснование выбора технологии проектирования ИРС	23
1.5 Анализ существующего программного обеспечения	26
ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ	30
2.1 Информационное обеспечение задачи	30
2.1.1 Характеристика нормативно-справочной входной и оперативной информации	30
2.1.2 Концептуальная модель информационно-развлекательной системы по обучению программированию детей младшего школьного возраста	33
2.1.3 Характеристика результатной информации	35
2.2 Проектирование ИРС по обучению программированию детей младшего школьного возраста	36
2.2.1. Обоснование выбора средства реализации ИРС по обучению программированию детей младшего школьного возраста	36
2.2.2. Проектирование ИРС по обучению программированию детей младшего школьного возраста	39
2.3 Программное обеспечение задачи	40
2.3.1 Общие положения (дерево функций и сценарий диалога)	40

2.3.2 Логическая модель данных	40
2.3.3 Физическая модель данных	47
ГЛАВА 3. ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ИНФОРМАЦИОННО-РАЗВЛЕКАТЕЛЬНОЙ СИСТЕМЫ	51
3.1 Расчет себестоимости ИРС	51
ЗАКЛЮЧЕНИЕ	58
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	59
ПРИЛОЖЕНИЕ А	61

ВВЕДЕНИЕ

Учебный процесс очень сложный и емкий, поэтому в современном обществе первостепенной задачей учебных программ является всесторонняя помощь учителям и учащимся в облегчении многих рутинных процессов. Процесс сбора, регистрации и обработки информации должны взять на себя структурированное программное обеспечение. При решении задачи огромное неудобство испытывает Учитель, так как заполняются и обрабатываются документы вручную, что требует значительно больше времени, чем машинная обработка. При этом возникает большой риск появления ошибок как при заполнении, так при подсчете статистических данных и заполнении ведомостей. Наибольшую продолжительность времени для получения и обработки занимает сбор и составление таблицы результатов успеваемости учеников, а также составления рейтинга полезности материалов при выполнении заданий. Для решения задач ручным способом требуются внимательность при подборке нужных параметров, правильность оформления и трата бумажных ресурсов и/или трата времени учителя на проверку правильности решения задач. Так же при вычислении в отчетах и ведомости не стоит пренебрегать человеческим фактором. Велика вероятность допущения ошибок и для того, чтобы исправить их необходимо заново распечатывать весь материал и/или заменять файлы материалов в источнике и файлы заданий. При этом источник для хранения информации назначается хаотично, без единой структуры хранения информации.

Таким образом, актуальность данного проекта обусловлена необходимостью более удобной и быстрой обработки информации при создании и/или выкладывании материалов для обучения учеников, проверки выполнения заданий в развлекательной форме. Для этого анализируются работы и материалы учителей, и на основе результатов разрабатывается новая модель информационной системы уже с развлекательными для учеников элементами.

Автоматизация процесса позволит проходить ученикам задания с интересом, не выходя из дома, а так же обеспечит высвобождение личного и рабочего времени учителей от трудоемких работ, повышение качества получения выходных форм при минимальных затратах времени.

Применение современных технических средств позволит автоматизировать всю ручную работу, а новая модель информационной системы наведет порядок и позволит учителям и ученикам работать в ускоренном режиме в безопасно установленном порядке. Но главная первостепенная задача автоматизации – это уменьшение трудозатрат, то есть работа учителей будет заключаться только во вводе информации с первичных документов и проверке правильности выполнения работ учениками.

Рассмотренные нами проблемы автоматизации ИС крайне сложно реализовать без вмешательства Программного Обеспечения и без грамотно составленного Технического задания.

Объект исследования – обучение программированию детей младшего школьного возраста.

Предмет исследования - технология построения информационно-развлекательной системы (далее – «ИРС»).

Целью работы является разработка информационно-развлекательной системы для обучения программированию детей мл. школьного возраста.

Список задач, выполнение которых гарантирует достижение цели:

1. Проанализировать научную и учебно-методическую литературу, необходимую для разработки информационной системы и написания пояснительной записки.

2. Проанализировать бизнес-процессы схожих систем, методик обучения и существующие разработки с целью определения технологии проектирования и выявления функций, необходимых для реализации при разработке информационной системы.

3. Построить концептуальную модель предметной области для выявления процессов, требующих автоматизации.

4. Проанализировать существующие архитектуры информационных систем и выбрать наиболее подходящую из них для разработки информационной системы.

5. Спроектировать интерфейс информационно-развлекательной системы для обучения программированию.

6. Разработать информационно-развлекательную систему для обучения программированию детей мл. школьного возраста выбранными средствами реализации.

7. Просчитать эффективность разработки и внедрения информационной системы в общий доступ через интернет с возможностью использовать все доступные функции с помощью браузера.

Таким образом, практическая значимость проекта заключается в приобретении опыта по разработке ИРС с веб-интерфейсом, а также по проектированию и разработке модели БД информационной системы на основе анализа предметной области.

Бакалаврская работа состоит из введения, трех глав, заключения и библиографического списка и приложений.

Во **введении** дается обоснование актуальности выбранной темы, формулируется проблема, выделяются цели и задачи.

В **главе 1** освещаются вопросы анализа предметной области и дан обзор принципов построения информационно-развлекательных систем;

В **главе 2** описывается поэтапное проектирование модулей информационной системы;

В **главе 3** рассчитывается и дается экономическое обоснование актуальности предлагаемой информационной системы;

В **заключении** сделаны основные выводы и итоги по проделанной работе.

Приложения содержат интересные программные коды функций разрабатываемой системы.

ГЛАВА 1. АНАЛИЗ ОРГАНИЗАЦИИ ОБУЧЕНИЯ ПРОГРАММИРОВАНИЮ ДЕТЕЙ МЛАДШЕГО ШКОЛЬНОГО ВОЗРАСТА

1.1 Система обучения программированию детей младшего школьного возраста

Из психологии известно, что игра — это особый вид деятельности, при котором ребенок (и взрослый тоже) символически познает окружающий мир, осваивает знания и умения. [13]

Рассматривая пример с обучением маленьких детей, например, тому же программированию, важно понимать, что исходные данные – различны. То есть каждый ребенок имеет свой набор навыков, развитых в той или иной мере у детей одинаковой возрастной категории. Кто-то быстро читает, но медленно считает, у кого-то отлично развито воображение, а кто-то склонен к анализу. Нередко бывает так, что показывая азы программирования маленькому ребенку, который по всем видимым признакам имеет достаточный набор навыков для развития аналитических способностей, может даже сразу и не понять сути программирования. В этом случае можно сделать передышку на несколько месяцев и попробовать тот вариант подачи информации, которые любят и понимают дети, а именно – игра. Игровой подход отлично себя зарекомендовал и успешно применяется в обучении дошкольников и младших школьников сложным наукам, к которым относится искусство программирования.

Детям проще будет усвоить материал, если он будет предоставлен в игровой или соревновательной форме, где они смогут не только увидеть результат своих действий, но и сравнить его с другими учащимися, а также принять участие в совместных решениях поставленных задач.

Самостоятельное предоставление детям различных материалов для обучения программированию не является правильным решением для студента, не обладающего никаким образованием в сфере педагогики. Было решено разработать информационно-развлекательную систему для обучения

программированию детей младших классов, которая должна упростить учителям предоставление материала с созданием заданий, а детям предоставит эту информацию в соревновательно-развлекательной форме обучения.

Для определения понятия информационно-развлекательная система для обучения программированию детей младшего школьного возраста, выделим основные понятия, которые будем применять в исследовании:

Информационная система – это набор механизмов, методов и алгоритмов, направленных на поддержку жизненного цикла информации и включающих три основных процесса: обработку данных, управление информацией и управление знаниями (рис. 1.1.1). С точки зрения программных технологий, информационная система – это не один, и даже не несколько программных комплексов. Можно построить структурную модель информационной системы, выделив её основные компоненты, которые содержат программные модули определенного класса (рисунок 1.1.2)[7].



Рисунок 1.1.1 – Структура информационной системы



Рисунок 1.1.2 – Структурная модель информационной системы

Система Дистанционного Обучения (СДО) - способ передачи информации и учебного материала без личного присутствия и вмешательства учителя. Этот метод не нов и зародился задолго до появления вычислительной техники.

Автоматизированные Обучающие Системы (АОС) представляют собой программно-технические комплексы, включающие в себя методическую, учебную и организационную поддержку процесса обучения, проводимого на базе информационных технологий.

Автоматизированный Обучающий Курс (АОК) - структурированная последовательность занятий, отвечающая всем требованиям к АОС[12].

Любая информационная система состоит из базы данных или базы знаний. Баз данных, сокращенно БД - это совокупность структурированных данных, взаимосвязанных между собой. *Microsoft Excel* – самый распространенный пример простейшей БД. В последнее время наибольшее распространение получили реляционные базы данных. Реляционная база данных представляется пользователю как совокупность таблиц. Под базой знаний мы подразумеваем форму представления информации в экспертных системах, относящихся к классу систем искусственного интеллекта. Среди

основных требований, предъявляемых к информационным системам, следует отметить:

1. Отношение совокупности всех затрат на текущее мероприятие к результатам является *Эффективностью информационной системы*.

2. Степенью толерантности системы к выполнению заданных функций называют *Качество функционирования информационной системы*. Основные свойства качества функционирования информационной системы, по которым можно судить данную систему:

- правильность функционирования информационной системы в целом;
- взаимодействие, совершенствование и развитие информационной системы, как в программной части, так и в технической;
- своевременное представление информации;
- надежное и безопасное выполнение функциональных технологических операций;
- полнота представляемой информации;
- безошибочные вычислительные операции;
- актуальность данных
- конфиденциальность передаваемой информации.

3. Безотказность технических средств ее оснащения, а так же частота и количество ошибок исполнителей является *Надежностью информационной системы*.

4. Безопасность информационной системы включает в себя следующий функционал:

- защита информации, которая хранится, обрабатывается и используется в этой системе;
- защита пользователей информационной системы от шпионских атак и прочих недоброжелательных действий, направленных на причинение вреда этой системе и объектам самой системы;

- защита объектов информационной системы от несанкционированного доступа, редактирования ее заданных параметров и изменение режима эксплуатации[5].

При обучении детей программированию ИРС будут представлены учебные материалы и задания.

Учебный материал - это педагогически выверенная, целесообразная система знаний, которая подлежит усвоению учащимися. Содержание любого преподаваемого предмета в вузе и школе должны определять следующие принципы и требования с учётом примерных программ, утверждённых Департаментом образовательных программ и стандартов профессионального образования:

- Соответствие современному состоянию и перспективам развития науки, техники и технологическим процессам, применяемым в соответствующих отраслях, с учётом характера труда;

- Правильная постановка воспитательного процесса, развивающего обучение и подготовку квалифицированного специалиста;

- Реализация принципов общей и профессиональной дидактики, которые всецело являются научно-технической основой предмета;

- Наличие компонентов, развивающих техническое мышление студентов и способствующих интеллектуализации труда будущих специалистов согласно выбранной профессии[6].

Учебное задание — вид поручения учителя учащимся, в котором содержится требование выполнить какие -либо теоретические или практические учебные действия. Некоторые задания требуют активизации знаний и действий, другие - актуализации и успешного применения ранее усвоенного материала. Более сложные задания реализуют обе вышеуказанные функции. В педагогической и психологической литературе отражены общедидактические и частно-методические классификации учебных заданий, однако общепринятой единой классификации заданий пока не существует, и вряд ли возможна[3].

Далее рассмотрим структурную схему ИРС:



Рисунок 1.1.3 – Схема ИРС для обучения программированию детей мл. школьного возраста

1.2 Обоснование автоматизации информационно-развлекательной системы

Основные подходы концептуального моделирования основаны на разных принципах декомпозиции. Это:

1) *Структурный подход*. Это разделение системы на несколько функций, работа каждой из которых отдельно получает данные от другой и предоставляет конечный результат другой, решая конкретные задачи и выполняя соответствующие процедуры, для получения результата.

2) *Объектно-ориентированный подход*. Отдельно работающие элементы (объекты) этой системы представляют собой целостное решение конкретных задач.

Методологии структурного подхода основаны на двух базовых принципах:

- разбиения сложной задачи на несколько меньших независимых задач;
- составление задач в единую иерархическую древовидную структуру, с большей детализацией на каждом последующем уровне;

Объектно-ориентированный подход строится на такой категории объектной модели как класс, объединяющей данные и операции, относящиеся к

определенному объекту системы. Данные объекта в основном остаются неизменными, в отличие от динамических процессов, происходящих с объектами.

Для структурного анализа используются следующие методологии:

- SADT-методология, предназначенная для построения функциональной модели предметной области, отображающая функциональную структуру объекта моделирования, производимые системой операции и связи между ними. Является основой методологии IDEF0;

- DFD-методология – определяет модель системы как иерархию диаграмм потоков данных, которые описывают процессы преобразования входной информации в выходную информацию. Процесс представляет собой иерархию, состоящую из нескольких уровней, каждый верхний уровень детализируется при помощи диаграмм нижних уровней;

- IDEF3-методология представляет собой механизм документирования и сбора информации о процессах в системе.

Описание ИРС модели обучения программированию детей младшего школьного возраста будем выполнять с использованием IDEF0-модели, позволяющей выполнять всесторонний анализ бизнес-процессов, осуществляемых в ИРС. В отличие от нее, DFD-методология в основном используется для описания документооборота.

На практике использование какой-либо методологии осуществляется с помощью определенных программных средств, носящих название CASE-средства.

Программные средства, которые поддерживают процессы создания и сопровождения ИС (CASE-средства), в том числе формулировку требований к системе, проектирование прикладного программного обеспечения, анализ базы данных, шаблонное генерирование программного кода, тестирования и документирования, обеспечивают качественное конфигурационное управление системы и проекта в целом[1].

В России наиболее востребованы такие средства, как BPwin, ERwin, Rational Rose, Developer Suite, MS Visio.

Для описания бизнес-процессов ИРС требуется средство, предоставляющее функцию построения моделей предметной области. Поэтому для построения функциональных диаграмм «Как есть» и «Как будет» будем использовать CASE-средство MS Visio.

1.3 Концептуальное моделирование деятельности ИРС для обучения программированию школьников младших классов

Определение точки зрения, с которой будут рассматриваться бизнес-процессы:

Преподаватели:

- регистрация в информационной системе;
- разработка заданий для учеников;
- распределение категорий и условий для обучения;
- проверка результатов выполнения заданий;
- получение подробной отчетности успеваемости своих учеников;
- возможность дистанционного взаимодействия с учениками из других заведений;
- добавление индивидуальных заданий для групп учеников.

Ученики:

- регистрация в информационной системе;
- обучение взаимодействию с компонентами веб-интерфейса;
- прохождение курсов обучения программирования;
- выполнение заданий по ходу обучения;
- помощь и взаимодействие с другими участниками обучения;
- взаимодействие с учителями;
- участие в соревнованиях;
- закрепление знаний итоговыми тестами.

Используя программу BPWin из пакета AllFusion Process Modeler 7 построим концептуальную функциональную модель турагентства по методологии IDEF0 (рис. 1.3.1). Компания обозначена прямоугольником,

стрелками обозначены потоки входной и выходной информации.



Рисунок 1.3.1 – Концептуальная диаграмма деятельности ИРС

Входными потоками являются:

- ученик (данные ФИО, почта, информация);
- учитель(ФИО, информация о учебном заведении, предметы);
- методические пособия (учебные и задания, созданные учителями).

Выходными потоками являются:

- отчеты по успеваемости (табличная статистика, список оценок).

Потоками управления являются:

- руководство по эксплуатации оборудования (справочный материал);
- школьная программа (учрежденная Мин.Обр.РФ);
- списки учебных заведений (адрес, телефон);
- должностные инструкции (трудовое законодательство, договор);
- инструкции пользователя (взаимодействие с ИРС, браузером);
- ГОСТ-документация (подробнее в разделе литература).

Потоками механизмов системы являются:

- СУБД ИРС;
- браузерные отличительные библиотеки преобразования кода;

- учетная запись;
- проверяющий;
- серверные скрипты;
- печатное оборудование.

Выполним декомпозицию концептуальной модели 0-го уровня (Рис 1.3.2)

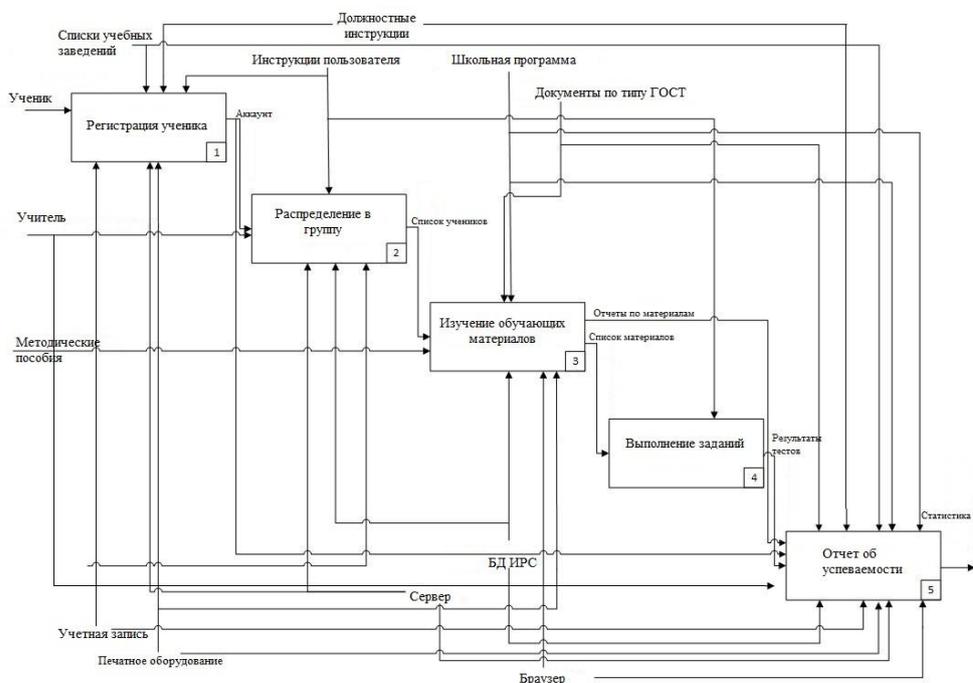


Рисунок 1.3.2 – Диаграмма A0 в виде декомпозиции деятельности ИРС

Диаграмма декомпозиции 1-го уровня представлена пятью блоками:

- регистрация ученика;
- распределение в группу;
- изучение обучающих материалов;
- выполнение заданий;
- отчет об успеваемости.

Каждая функция в ИРС важна и нуждается в более подробной детализации:

Регистрация ученика – важная функция ИРС. Во время регистрации создается аккаунт (учетная запись) и через неё осуществляется процесс обучения, соревнования и помощи между другими учениками. Детальное

рассмотрение функции «Регистрация ученика» представлена на рисунке 1.3.3:

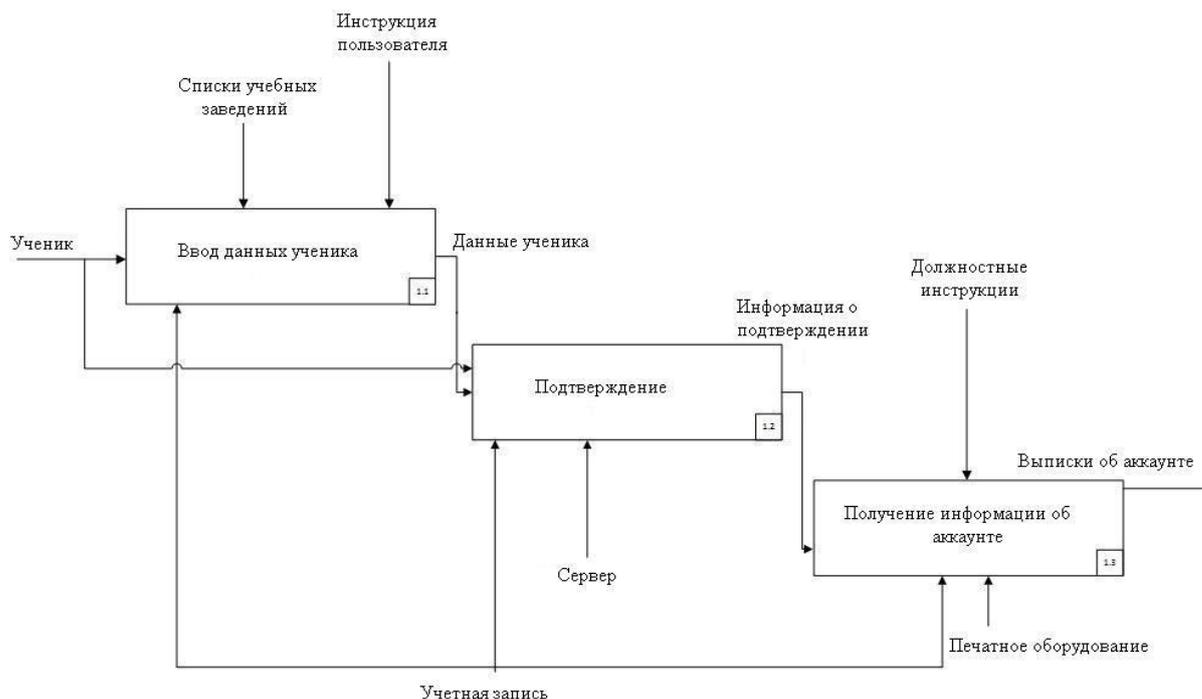


Рисунок 1.3.3 – Диаграмма бизнес-процесса в виде декомпозиции «Регистрация ученика»

Составными бизнес-процесса «Регистрация ученика» являются 3 блока:

- ввод данных ученика: здесь создается учетная запись и вводится ФИО;
- подтверждение: занесение информации с проверкой правильности;
- Получение информации об аккаунте: логин, пароль и группа.

При регистрации ученика, ученик вводит необходимые данные для учителя, чтобы подтвердить свою личность и вносит в Базу Данных информацию для выделения ему места в списке аккаунтов, а также получает при подтверждении необходимую информацию для дальнейшего взаимодействия с ИРС.

Далее рассмотрим более подробно функцию «Распределение в группу». Распределением учеников по группам занимаются учителя. Для этого им просто необходимо зайти под своей учетной записью в ИРС и получить список новых зарегистрированных учеников. Детальное рассмотрение функции «Распределение в группу» представлено на рисунке 1.3.4:

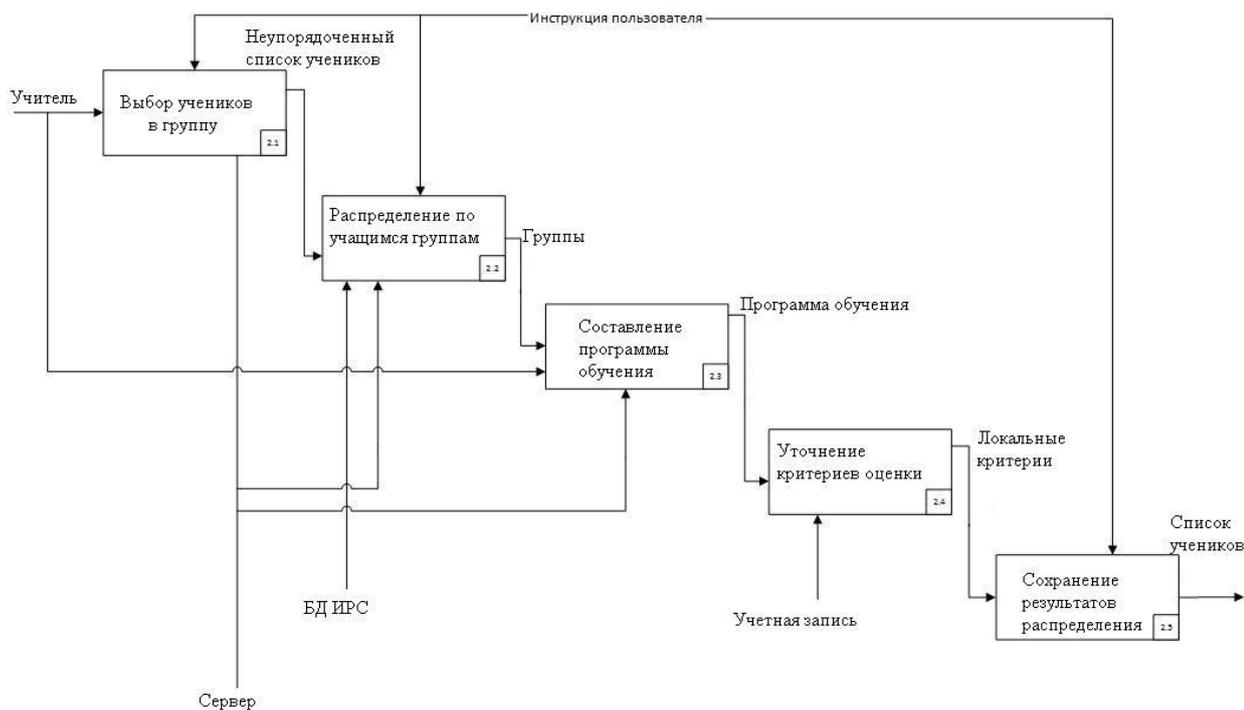


Рисунок 1.3.4 – Диаграмма бизнес-процесса в виде декомпозиции «Распределение в группу»

Составными бизнес-процесса «Распределение в группу» являются 5 блоков:

- выбор учеников в группу (выбор из списка именно своих учеников);
- распределение по учащимся группам (добавление учеников в группы);
- составление программы обучения (алгоритм прохождения заданий);
- уточнение критериев оценки (установить необходимый минимум);
- сохранение результатов распределения (ученики, группы, программы).

Далее рассмотрим более подробно функцию «Изучение обучающих материалов». Заранее созданные материалы становятся доступными для просмотра только тем ученикам, в программу которых входит изучение данного материала. После просмотра материалов открывается доступ к прохождению заданий. Детальное рассмотрение функции «Изучение обучающих материалов» представлено на рисунке 1.3.5:

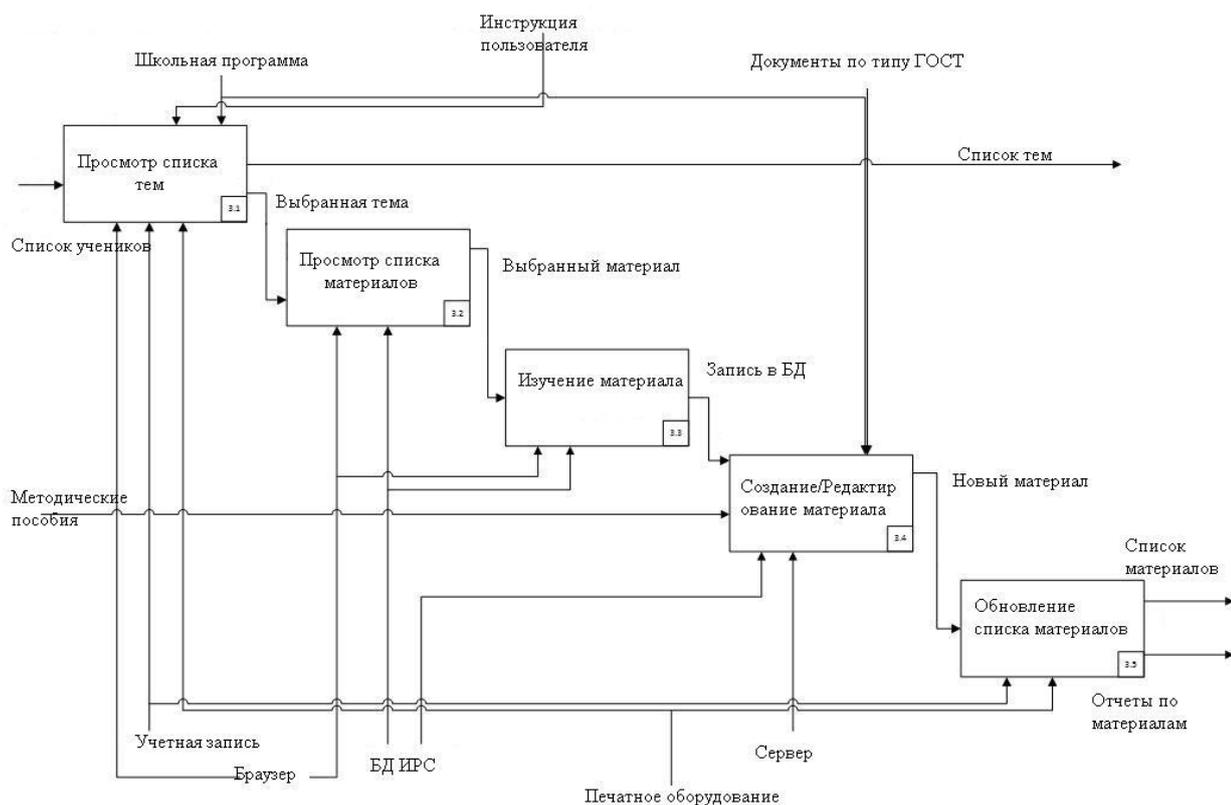


Рисунок 1.3.5 – Диаграмма бизнес-процесса в виде декомпозиции «Изучение обучающих материалов»

Составными бизнес-процесса «Изучение обучающих материалов» являются 5 блоков:

- просмотр списка тем (разделение по предмету, учителю, году обучения);
- просмотр списка материалов (материалы заранее создаются учителями);
- изучение материала (при просмотре заносится информация в БД);
- создание/редактирование материала (доступно только учителям);
- обновление списка материалов (после просмотра, редактирования).

Здесь наиболее важным блоком является «Создание/редактирование материала», так как именно в этой функции появляются новые материалы, необходимые для формирования процесса обучения в ИРС. Созданием материалов занимаются учителя в специальном редакторе создания материалов, где они могут как написать самостоятельно материал по обучению детей младшего школьного возраста, так и загрузить уже заранее созданные материалы поддерживаемого формата.

Детальное рассмотрение функции «Создание/редактирование материала» представлено на рисунке 1.3.6:

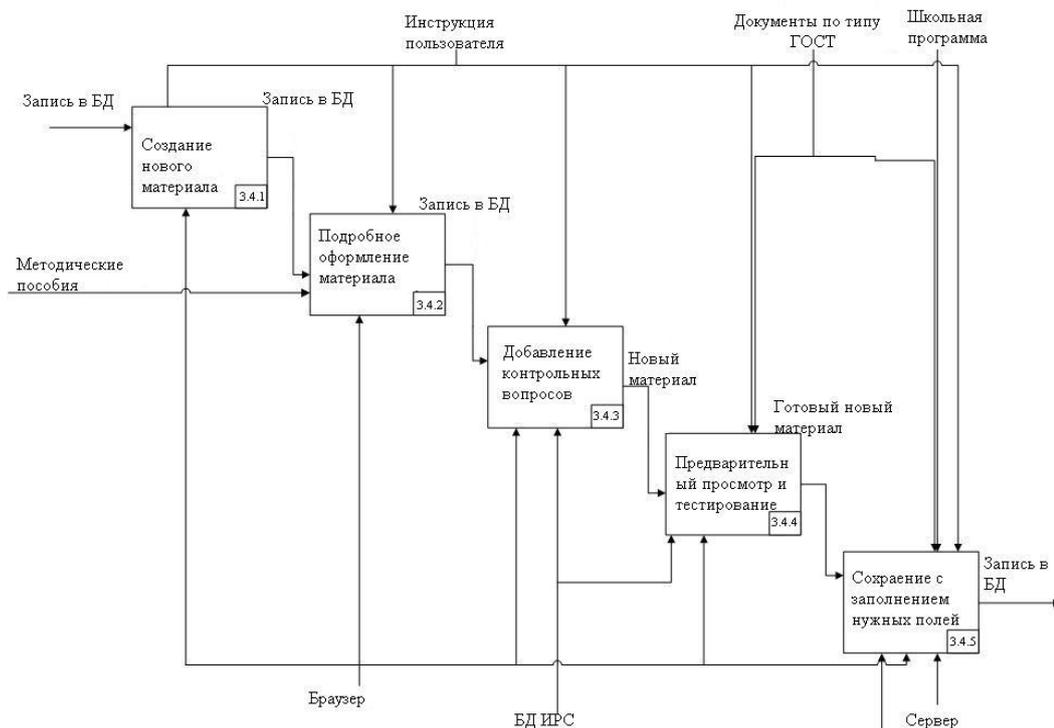


Рисунок 1.3.6 - Декомпозиция бизнес-процесса «Создание/редактирование материала»

Составными диаграммы декомпозиции 2-го уровня «Создание/редактирование материала» являются 5 блоков:

- создание нового материала (название, описание, загрузка изображений);
- подробное оформление материала (редактирование формата отображения);
- добавление контрольных вопросов (вставка пустых проверочных полей);
- предварительный просмотр и тестирование (доступно только учителю);
- сохранение с заполнением нужных полей (учитель, предмет, тема).

Выполнение заданий также является необходимой функцией ИРС. Задания создаются, также, учителями. При прохождении заданий, ученики получают результат в игровых очках (баллах). На таблице успеваемости будут видны самые успешные ученики. Соревнования могут проводиться не только между учениками одной группы, но и между учениками других групп

или между группами посредством распределения средних баллов учеников. Детальное рассмотрение функции «Выполнение заданий» представлено на рисунке 1.3.7:

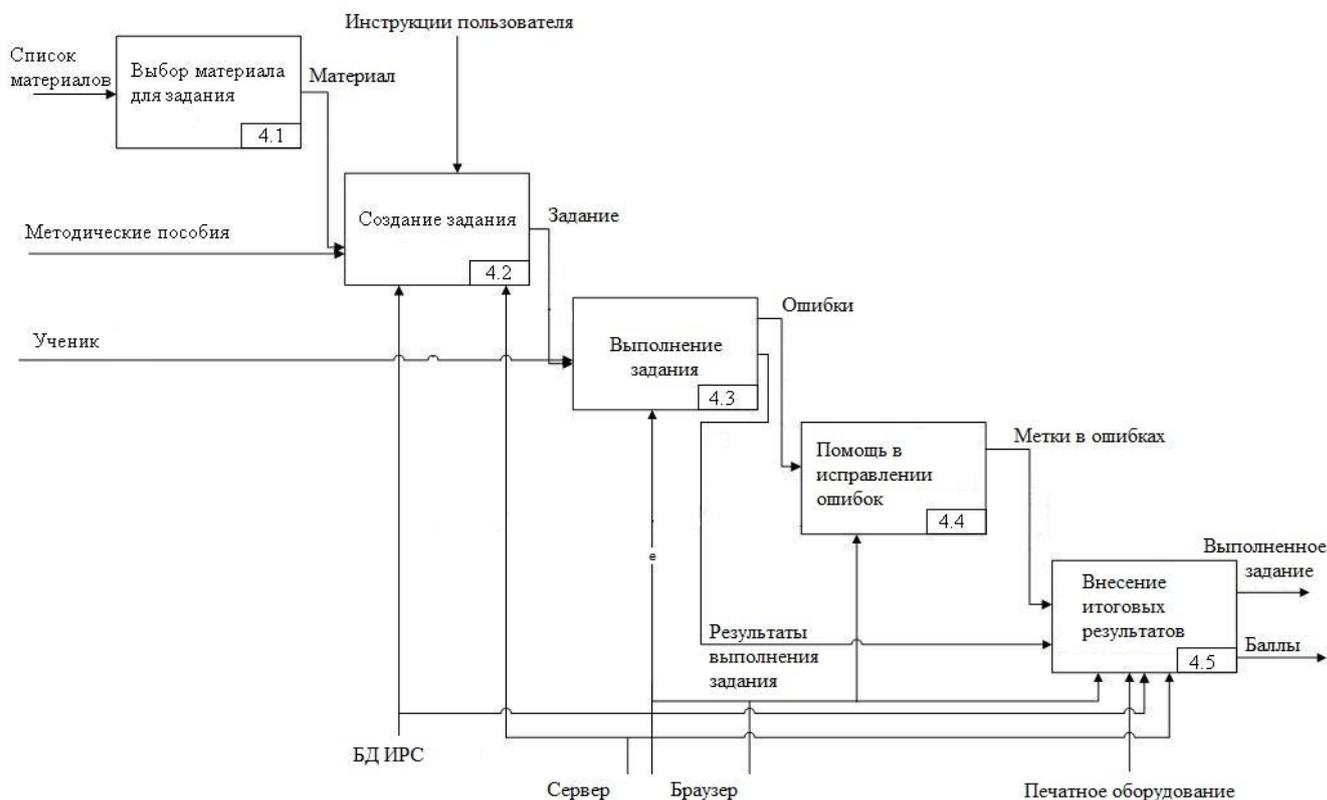


Рисунок 1.3.7 - Декомпозиция бизнес-процесса «Выполнение заданий»

На основе выполненного анализа, можно сделать выводы о том, что модель «Как должно быть» соответствует всем предъявленным выше требованиям и устраняет недостатки системы «Как есть», а именно:

- будет создана автоматизированная база данных учетных записей;
- будет создана база данных материалов, заданий и результатов;
- будет реализована возможность самостоятельной регистрации клиентов в базе данных ИРС;
- будет реализована регистрация оценок и автоматизация статистики;
- будет реализована возможность учителей создание и редактирование материалов, заданий и учебных программ (алгоритмов прохождения заданий);

- будет реализована игровая соревновательная система, процессы распределения очков при прохождении заданий и помощи другим ученикам.

1.4 Анализ существующих разработок и обоснование выбора технологии проектирования ИРС

Для информационно-развлекательных систем разработчиками предлагаются различные решения по автоматизации обучения.

Национальный открытый университет Intuit.ru: организация, предоставляющая с помощью собственного сайта услуги дистанционного обучения по нескольким образовательным программам, многие из которых касаются информационных технологий.

Система Intuit.ru позволяет:

- вести учет поисковых запросов на сайте еще до выбора обучающего курса;
- вести учет прохождения обучения, результаты тестов и оформление сертификатов по обучающим курсам зарегистрированных пользователей;
- посчитать эффективность выполнения заданий;
- вести учет задания теста любой сложности;
- вести учет индивидуальных заданий для корпоративных лиц;
- проверка срока действия выполненного задания студента на дату начала теста;
- вложение фотографий и видеоматериалов сразу в задание или в обучающий материал;
- установка напоминаний по курсам и заданиям, отправка напоминаний по электронной почте студентам, а не только на сайте;
- раздача заданий сотрудникам через сайт, также как это делается в MS outlook с контролем исполнения;
- сохранение всей истории работы с курсом и легкая передача последних результатов тестирования в случае подтверждения;
- получать и отправлять email письма студентам и учителям из сайта;

- вести учет материалов с заданиями и напоминания о выпуске новых;
- учет оплат не только вам студентами, а еще расчет части прибыли учителям;
- уже наполненный справочник категорий и ключевых фраз;
- обновляемый справочник категорий со всеми реквизитами по оформлению курсов;
- оформить курс на Юридическое лицо, распечатать материал и/или учебник на юридическое лицо;
- печать именного сертификата, подтверждающего прохождение курса повышения квалификации и любых бланков которые настраиваются под пользователя;
- обеспечить свободное перемещение студентов по сайту, сортировка по дате добавления курсов и рейтингу;
- вести полный учет доходов сайта, включая проценты учителям и рекламу;
- выгружать автоматически курсы и задания в студенческую программу;
- импортировать в систему данные из документов офисных приложений и сохранить тем самым всю целостность работы и стилистику предоставления материала.

Программное обеспечение, которое было рассмотрено ранее, не предоставляют необходимый перечень решения для нашей задачи, но обладают следующими достоинствами:

- удобный поиск курсов по нескольким ключевым фразам;
- ведение рассылок и генерация автоматических смс-сообщений по различным событиям;
- возможность добавления графической и документальной информации.

Эти достоинства необходимо учесть при разработке ИРС.

Выполненный анализ модели «Как должно быть» и анализ предложений разработчиков программного обеспечения на рынке для обучения школьников младших классов, а также сформулированные задачи автоматизации ИРС,

позволяют сформулировать принципиальное техническое задание на разработку ИРС обучения программированию школьников младших классов.

Разработка предназначена для автоматизации сбора, обработки и хранения информации о школьниках, группах и учителях, загруженных, полученных и выполненных материалов, заданий и расчета обучаемой эффективности в виде игрового соревнования с результатом.

Должна быть разработана система защиты базы данных. Доступ к редактированию материалов и заданий предоставляется только учителям или администратору сайта.

ИРС необходимо снабдить удобным редактором материалов и заданий, который предоставит возможность учителям самостоятельно добавлять как текст материалов, так и файлы, содержащие этот материал.

Требования к ИРС:

1. Функциональные характеристики.

Обеспечение следующих необходимых функций:

- формирование задания и полей для ввода информации исходя из синтаксиса редактора заданий;
- учет школьников, учителей, материалов, заданий, результатов их выполнений;
- поиск учебных материалов и заданий по группе;
- формирование отчета по успеваемости школьников;
- формирование списка материалов по запросам пользователей;
- отправка электронных уведомлений на события.

Выходными данными в ИРС являются:

- просмотр и печать выбранных материалов;
- просмотр и печать результата выполненных задач;
- просмотр и печать информации по выбранному школьнику;
- просмотр и печать отчета успеваемости.

2. Требования к надежности.

Говоря о функционировании любой ИРС, во главу угла всегда ставится строгий контроль входной и выходной информации. Интересует, прежде всего, данные, вводимые пользователем и данные, которые уже содержатся в базе.

Контроль должен быть отлажен, в случае ошибки ввода неверной информации необходима подача соответствующих сообщений и подсказок для предотвращения ошибок, а так же должна предоставляться возможность безотказного продолжения работы.

Удаление данных или отмена операции не должна вызывать сбоев программы и потери информации из базы данных. Так же всегда должна быть предоставлена возможность отменить любое действие.

1.5 Анализ существующего программного обеспечения

Рассматриваемая задача автоматизации может быть выполнена с помощью специального программного обеспечения, разрабатываемого web-средствами.

Рассмотрим некоторые из предлагаемых на рынке программных продуктов, учитывая, что необходимо наличие следующих основных функций:

- учет учащихся;
- учет материалов, заданий и наличие поиска по ним;
- формирование предварительной программы обучения и алгоритма прохождения для группы школьников.

Одной из наиболее подходящих является Информационная система студентов Тольяттинского Государственного Университета.

В данной программе реализованы перечисленные ниже функции.

Работа со студентами:

- автоматически подставляются данные студентов, ранее зарегистрированных в ИС;
- автоматически сохраняются результаты прохождения тестирования (если студент зашел и прошел тест под своей учетной записью);

- автоматически в материалы для студентов могут быть подставлены документы офисных приложений и таблицы.

Работа с учителями:

- информация обо всех студентах сохраняется в базе данных;
- при оформлении успеваемости все его данные автоматически подставляются в единую БД, таблицу успеваемости и другие необходимые документы;

- с помощью поиска можно найти нужного студента. Фильтр в поиске можно настроить на фамилию, имя и отчество, контактную информацию, а также группу;

- своевременное получение необходимой информации и уведомлений.

Есть еще некоторые функции, которые не были рассмотрены, относящиеся к контролю действий образовательной системы в заданном учреждении.

В глобальной сети интернета особой популярностью пользуется онлайн институт «Интуит». Данный сайт предоставляет разнообразные курсы и выполняет следующие основные функции:

- учет созданных курсов с возможностью создания списка «похожих» материалов (по специальности);

- печать официально зарегистрированного документа, свидетельствующего о прохождении курса "Сертификат";

- информирование администраторов и учителей о покупке курсов по заявкам;

- онлайн поиск и предпросмотр курсов.

- контроль доступа пользователей к различным модулям системы.

Таким образом, использование подобной информационной системы смогло бы решить все поставленные задачи автоматизации, однако, здесь обнаружены следующие недостатки:

- данная система не имеет открытого исходного кода и запрещена к распространению с учетом авторских прав;

- функции, необходимые для автоматизации рассматриваемой задачи ограничиваются CMS, на которой написана Информационная система, а самостоятельное изменение и/или добавление модулей в ИС влечет за собой последствия, отследить которые, не зная MVC-системы CMS очень сложно.

Следующий программный продукт создан в Тольяттинском Политехническом Техникуме и носит такое же название. Данное приложение распространено только в учебных заведениях и позволяет работать только по локальной сети. Для функционирования данного ПО необходима СУБД MS Access.

Она выполняет следующие функции:

- сбор статистики и контроль базы данных студентов;
- добавление графической информации к тестам;
- оформление тестируемого материала по разделам (группы, учителя, программы и др.);
- добавление графических или других объектов к материалам.

Главными недостатками вышеописанной системы является ее стоимость, а так же неоднократные удачные попытки взлома со стороны студентов (проблемы безопасности). Недостаточное количество поддерживаемых и выполняемых функций, а также проблемы с безопасностью, служат отказом от применения данной программы.

При получении задания, было принято решение о создании своей собственной системы в более простом стиле изначально и с возможностью расширения дополнительных модулей в дальнейшем, по мере необходимости. Для подтверждения учительских аккаунтов используется ручной способ – самостоятельное подтверждение личности через директора учебного заведения. Подтверждать учеников могут сами учителя.

Помимо упрощения, функционал нашей ИРС имеет ряд своих особенностей и отличий в сравнение с рассматриваемыми, а это означало либо изменение исходного кода, либо запрос на расширение.

Результат сравнения аналогичных программ приведен в таблице 1.5.1.

Таблица 1.5.1 - Сравнение программного обеспечения

	Система студентов ТГУ	Интуит	Система студентов ТПТ
Функционал	Меньшая часть	Большая часть	Большая часть
Работа с удаленного места	Учетная запись	Учетная запись	Локальная запись
Необходимость обучения со стороны учителя	Справочный материал	Нет данных	Справочный материал
Структура приложения	Веб-интерфес	Веб-интерфейс	Windows-приложение
Ценовая политика	Неизвестно	Только сотрудничество	На основании договора

На основании анализа можно сделать вывод, что покупка такой системы не будет экономически выгодна в рамках решения поставленной задачи автоматизации.

Поэтому встает вопрос о самостоятельной разработке такого программного обеспечения, которое будет, во-первых, экономически целесообразно, во-вторых, учитывать все особенности автоматизируемых бизнес-процессов, в-третьих, охватывать все существующие потребности.

Выводы по первой главе

В данной главе были рассмотрены приоритетные вопросы и стороны решения задачи ИРС с использованием АИС, а именно, упорядоченное хранение справочной информации о школьниках и динамических расчетах внутри ИРС, информации об учителях, заданиях, материалах, результатах, успеваемости. Это позволяет проводить мероприятия по обучению школьников младших классов программированию в форме соревновательной игры. Рассмотрены аналогичные системы. Выполнено обоснование и постановка задачи на разработку ИРС.

ГЛАВА 2. ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ

2.1 Информационное обеспечение задачи

2.1.1 Характеристика нормативно-справочной входной и оперативной информации

Объекты информационной системы, которые описывают подмножества данной системы, называют *Справочниками*. В большинстве случаев, к справочникам следует относить данные, которые вносятся однократно и затем динамически обновляются. Данные справочников используются в системе обычно как базовые[8].

Заполнение справочников можно производить как во время создания ИРС, так и уже после создания, подготовив соответствующие модули. Добавление информации произойдет само собой в течение времени.

Таблица 2.1.1 – Перечень используемых справочников

№	Название	Приблизительный объем справочной информации	Приблизительная частота обновления актуальности	Приблизительный объем обновляемой информации, %
	Учебные заведения	40	1 раз в 2 недели	< 10%
	Группы	130	2 раза в год	< 10%
	Материалы	150	5 раз в месяц	< 10%
	Задания	160	4 раза в месяц	< 10%
	Аккаунты	1500	4 раза в год	< 10%
	Изображения	100	3 раза в месяц	< 10%

Учебные заведения – список учебных заведений для регистрации учителей и групп учеников, одной группе принадлежит одно уч. Заведение.

Группы – список учебных групп, содержащих информацию об учебном заведении и учителе. Один ученик может быть только в одной группе.

Материалы – список материалов, предоставляемых группам учеников для изучения предметной области. Может содержать файлы и изображения.

Задания – список заданий, созданных учителями для проверки знаний учеников. Содержит пустые поля для заполнения и количество очков.

Аккаунты – список учителей и учеников. Учителя могут создавать группы, материалы и задания, а также принимать учеников в группы.

Изображения – список изображений, хранящихся на сервере и использующиеся для материалов. Проверяется пересечение имен.

Каждый справочник может быть заполнен или исправлен лицами, получивший доступ к добавлению, редактированию и изменению информации, кроме некоторых частных случаев:

- редактирование справочника «Аккаунты» возможно либо владельцами этих аккаунтов, либо учителями при присвоении новым ученикам групп, а также исключением из списка группы ученика;

- редактирование материала или задания осуществляет только учитель, опубликовавший это задание. В случае, если учитель удален или заблокирован в Базе Данных, задания и материалы становятся недоступны для редактирования, пока учитель не разблокирует свою учетную запись;

- изображения редактировать или удалять - не позволено;

- редактировать группу может только тот учитель, кто создал группу;

- редактировать информацию об учебном заведении может только администратор сайта, получив соответствующее уведомление на электронную почту во избежание дезинформации или прочих неудобств;

- задания могут быть созданы только при прикреплении к какому-либо материалу и открыты учениками только в случае записи о его прохождении материала, который прикреплен к данному заданию.

2.1.2 Концептуальная модель информационно-развлекательной системы по обучению программированию детей младшего школьного возраста

Модель — образ или прообраз какого-либо объекта или системы объектов, используемый при определённых условиях в качестве их «заместителя» или «представителя»[11].

Известно, что система состоит из множества отдельных элементов, соединённых между собой определёнными связями. Значит и модель системы должна абсолютно точно описывать и воспроизводить все отношения и связи внутри исследуемого объекта. Это справедливо для взаимоотношений всех элементов или выделяемых групп элементов, рассматриваемых как подсистемы. Главным итогом процесса моделирования системы считается установление влияния и действия одних элементов на другие. Так же необходимо установить последствия этих взаимодействий.

С помощью процесса моделирования многие предприятия определяют план создания информационных систем, удовлетворяющих его ближайшие и перспективные информационные потребности.

Моделирование систем удобно тем, что в ходе самого процесса отрабатываются и прогоняются различные сценарии и выбирается наиболее подходящий вариант модели.

Черный ящик — это система, имеющая определенный вход и выход, но устройство которой неизвестно и неважно. Черный ящик выполняет одну хорошо понимаемую функцию и связи между черными ящиками должны быть максимально просты.

Иерархия подразумевает, что систему можно разбить на несколько логических уровней и изобразить их в виде пирамиды. Каждый нижний уровень является более подробной копией предыдущего.

DFD — это структурированная иерархия черных ящиков. На каждом уровне абстракции модель представляет собой набор черных ящиков, связанных потоками данных.

Построение модели DFD начинается с определения внешних сущностей, источников и приемников данных. В итоге, у нас есть все необходимое, чтобы построить контекстную диаграмму, которая представляет нашу систему как один большой черный ящик[10].

Схема инфологической модели данных представлена на рисунке 2.1.1

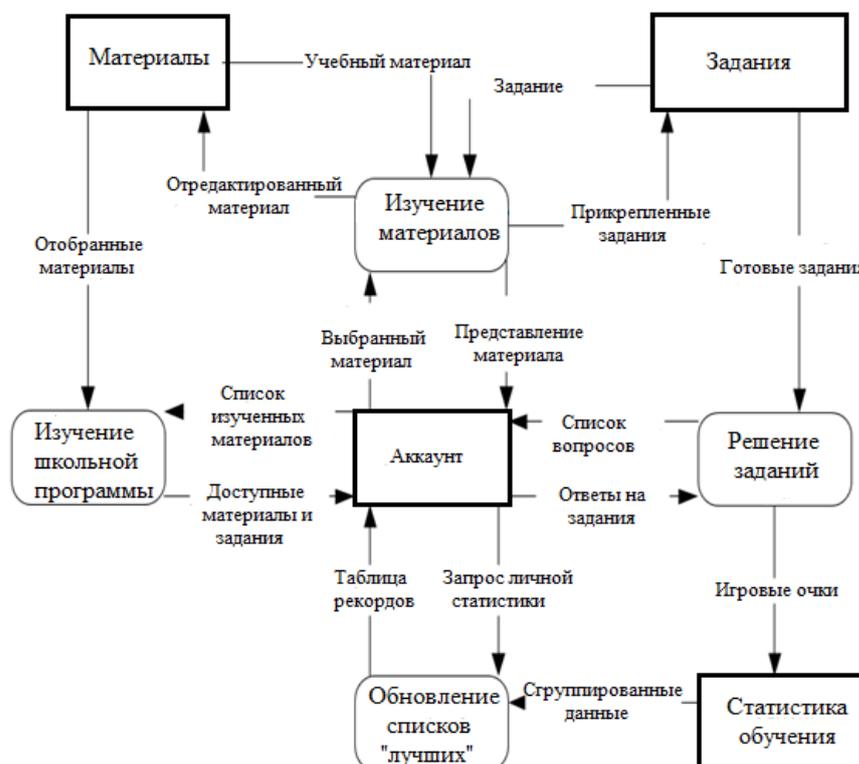


Рисунок 2.1.1 - Инфологическая схема ИРС

Рассмотрим данную модель:

В данной модели у аккаунта есть выбор:

1. В режиме учителя:
 1. Создать/отредактировать учебные материалы;
 2. Создать/отредактировать задания;
 3. Выполнить анализ результатов выполнения заданий.
2. В режиме учника:
 1. Изучить учебные материалы.
 2. Выполнить задания.

Далее рассмотрим подробнее несколько процессов:

Контекстная диаграмма уровня системы

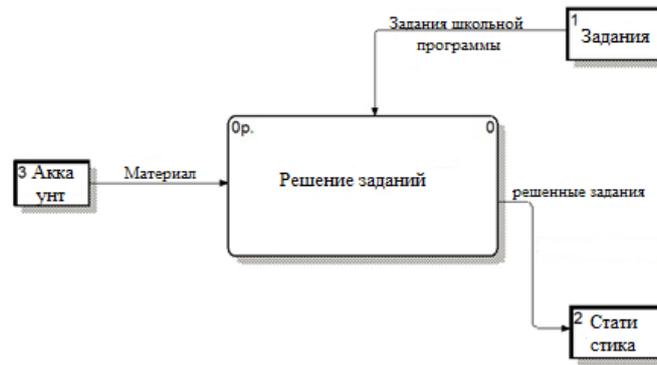


Рисунок 2.1.2 - Контекстная диаграмма системы «Решение заданий»

Диаграмма уровня подсистемы

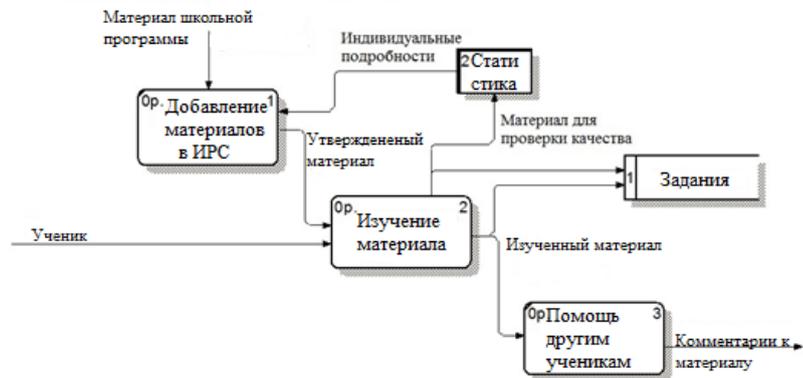


Рисунок 2.1.3 – Диаграмма уровня подсистемы «Изучение материала»

Диаграмма уровня процесса

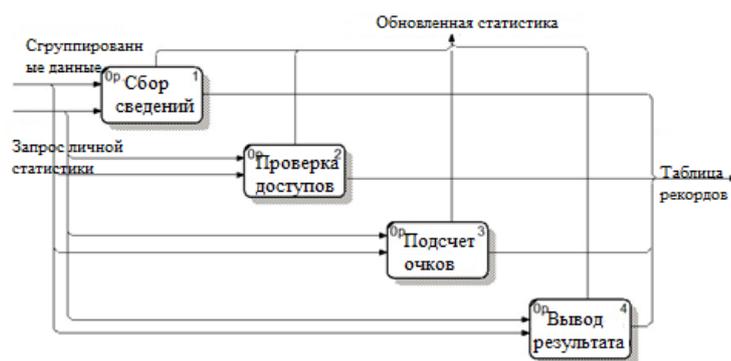


Рисунок 2.1.4 – Диаграмма уровня процесса «Обновление списков»

2.1.3 Характеристика результатной информации

К результатной информации в разработанной ИРС относятся следующие документы:

- таблица рекордов по заданию;
- список успеваемости групп;
- список успеваемости учеников в группе;
- статистика выполнения выбранного задания;
- список успеваемости учебных заведений;
- личная статистика ученика;
- отчет выполнения заданий за учебный год.

Таблица 2.1.2 – Реквизиты результатных документов

Наименование результатного документа	Реквизиты	Пользователь
Таблица рекордов по заданию	- ФИО ученика - Группа - Дата - Кол-во баллов	Ученик, учитель
Список успеваемости групп	- Наименование группы - Учебное заведение - Учитель - Дата регистрации - Средний балл - Количество учеников	Ученик, учитель
Список успеваемости учеников в группе	- Наименование группы - ФИО ученика - Пройденные материалы - Пройденные задания - Сумма баллов - Средний балл - Пропущенные задания	Ученик, учитель
Статистика выполнения выбранного задания	- Название задания - Создатель задания - ФИО ученика - Группа ученика - Набранный балл - Дата - Средний балл по заданию	Учитель

Наименование результатного документа	Реквизиты	Пользователь
Список успеваемости учебных заведений	<ul style="list-style-type: none"> - Город - Название учебного заведения - Список групп - Список учителей - Средний балл группы - Количество материала учителя - Количество заданий учителя - Средний балл учебного заведения 	Учитель, ученик
Личная статистика ученика	<ul style="list-style-type: none"> - ФИО ученика - Группа - Название пройденного материала - Название задания - Балл за задание - Сумма баллов 	Учитель, ученик
Отчет выполнения заданий за учебный год	<ul style="list-style-type: none"> - Название группы - ФИО ученика - Количество выполненных заданий - Средний балл 	Учитель

2.2 Проектирование ИРС по обучению программированию детей младшего школьного возраста

2.2.1. Обоснование выбора средства реализации ИРС по обучению программированию детей младшего школьного возраста

На основании анализа существующих информационных систем, в качестве среды для будущей программы, была выбрана WEB-среда.

Как мы хорошо знаем, HTML исключительно удобен для создания веб-страниц. Во многом проблема веб-сервера при обработке статичных HTML-страниц заключается в том, что в этом случае он выступает в роли простого передаточного механизма, и не более того. Браузер запрашивает страницу, сервер предоставляет ее, и на этом все заканчивается. Чтобы превратить сайт в диалоговое веб-приложение, веб-сервер должен выступить в новой, более динамичной роли, которую ему может обеспечить PHP.

PHP позволяет вам манипулировать содержанием веб-страницы на сервере непосредственно перед тем, как она будет отправлена браузеру. Это происходит следующим образом: в процессе интерпретации PHP-сценария сервер может изменить или создать любой HTML-код. После этого HTML-страница отправляется браузеру, который не знает и не придает значения тому, что в модификации кода HTML на сервере принимал участие PHP.

PHP-код интерпретируется на сервере. Он сохраняется в PHP-сценариях, которые чаще всего имеют расширение имени файла PHP. Эти сценарии часто выглядят так же, как и обычный HTML- и CSS-код. Фактически результатом процесса интерпретации сервером PHP-кода является чистый HTML- и CSS-код. Поэтому каждый PHP-сценарий в конечном счете после окончания исполнения его на сервере преобразуется в HTML и CSS.

PHP - это аббревиатура, которая вначале означала Personal Home Pages (персональные домашние страницы). Позднее значением этой аббревиатуры стало PHP: Hypertext Preprocessor (PHP: обработчик гипертекста). Последняя аббревиатура рассматривается как рекурсивная, так как она ссылается на саму себя - аббревиатура в аббревиатуре[9].

Проанализировав существующие Среды программирования веб-приложений, самым подходящим решением является:

- сервер на системе Apache 2;
- серверный язык программирования PHP 5.6;
- сервер Баз Данных MySQL;
- каскадная библиотека таблиц стилей bootstrap;
- язык программирования JavaScript ES5;
- язык разметки HTML;
- каскадные таблицы стилей CSS.

Данное решение имеет следующие преимущества:

1) Язык разметки HTML позволяет задать местоположение элементов Web-страницы в окне Web-браузера. У языка HTML довольно обширные возможности: вставка изображения, таблиц и гиперссылок, форматирование

отдельных символов или целых фрагментов текста, создание панели навигации с помощью карты, разделение окна Web-браузера на несколько и многое другое.

2) Оформление интерфейса со стороны пользователей осуществляется за счет каскадных таблиц стилей, сокращенно CSS. Необходимый внешний вид страницы достигается за счет заданных элементов стилистики и их параметров.

3) Однако, виртуальные страницы, созданные с помощью инструментов HTML и CSS, имеют важный недостаток, который проявляется в их статичности, то есть они не в состоянии изменяться, отвечая действиям пользователя. Исправить этот недостаток сможет внедрение в HTML программ на языке JavaScript. Такие страницы станут интерактивными, а интернет-пользователям станет проще и интереснее с ними работать. Язык JavaScript позволяет делать огромное количество действий с Web-страницами: можно обрабатывать формы еще до отправки на сервер, получать информацию о Web-браузере пользователя и его мониторе и изменять форматирование страницы в зависимости от необходимости. Кроме того система позволит редактировать элементы HTML-документа при выполнении различных скриптов, например, создавать всплывающие подсказки, окна или обновление данных таблицы, скрывать и отображать элементы Web-страницы.

4) Нельзя не отметить, что огромные возможности еще дает использование серверных технологий, среди которых популярен язык программирования PHP. В настоящее время он является наиболее популярным серверным языком, несмотря на появление новых, более оптимизированных аналогов. Благодаря большому количеству готовых продуктов и расширений он все еще на плаву. С помощью него можно управлять полученными данными и распределять информацию в браузере, в зависимости от типа и версии используемого браузер. PHP позволяет работать с файлами и каталогами, обрабатывая данные формы прямо на сервере. В список его возможностей входит: рассылка e-mail писем, загрузка различных файлов на сервер, создание

данной диаграммой пользователи системы имеют различные варианты использования данной системы:

1. Пользователь “Ученик”:

- может просматривать учебные материалы;
- может выполнять задания;
- помогать другим ученикам с выполнением заданий.

2. Пользователь “Учитель”:

- может просматривать учебные материалы;
- может управлять группами учеников;
- может помогать ученикам с заданиями;
- может создавать учебные материалы и задания;
- может создавать алгоритм по школьной программе.

На основе анализа данной диаграммы использования проектируемой системы была построена функциональная схема ИРС (рис. 2.2.2).

На рисунке изображена мобильная система поддержки образовательного процесса, которая была поделена на 2 модуля: Учитель и Ученик.

2.3 Программное обеспечение задачи

2.3.1 Общие положения (дерево функций и сценарий диалога)

Основным действующим лицом в разработанной системе является учитель. Дерево функций для учителя представлено на рисунке 2.3.1

2.3.2 Логическая модель данных

Система баз данных — это, по сути, не что иное, как компьютеризированная система хранения однотипных записей. Современную базу данных можно рассматривать с точки зрения компьютерной картотеки - некоего хранилища для определенного количества файлов данных, занесенных в компьютер. Пользователи базы могут выполнять большое количество разнообразных операций над такими файлами сами либо передавать команды на обработку файлов системе, например:

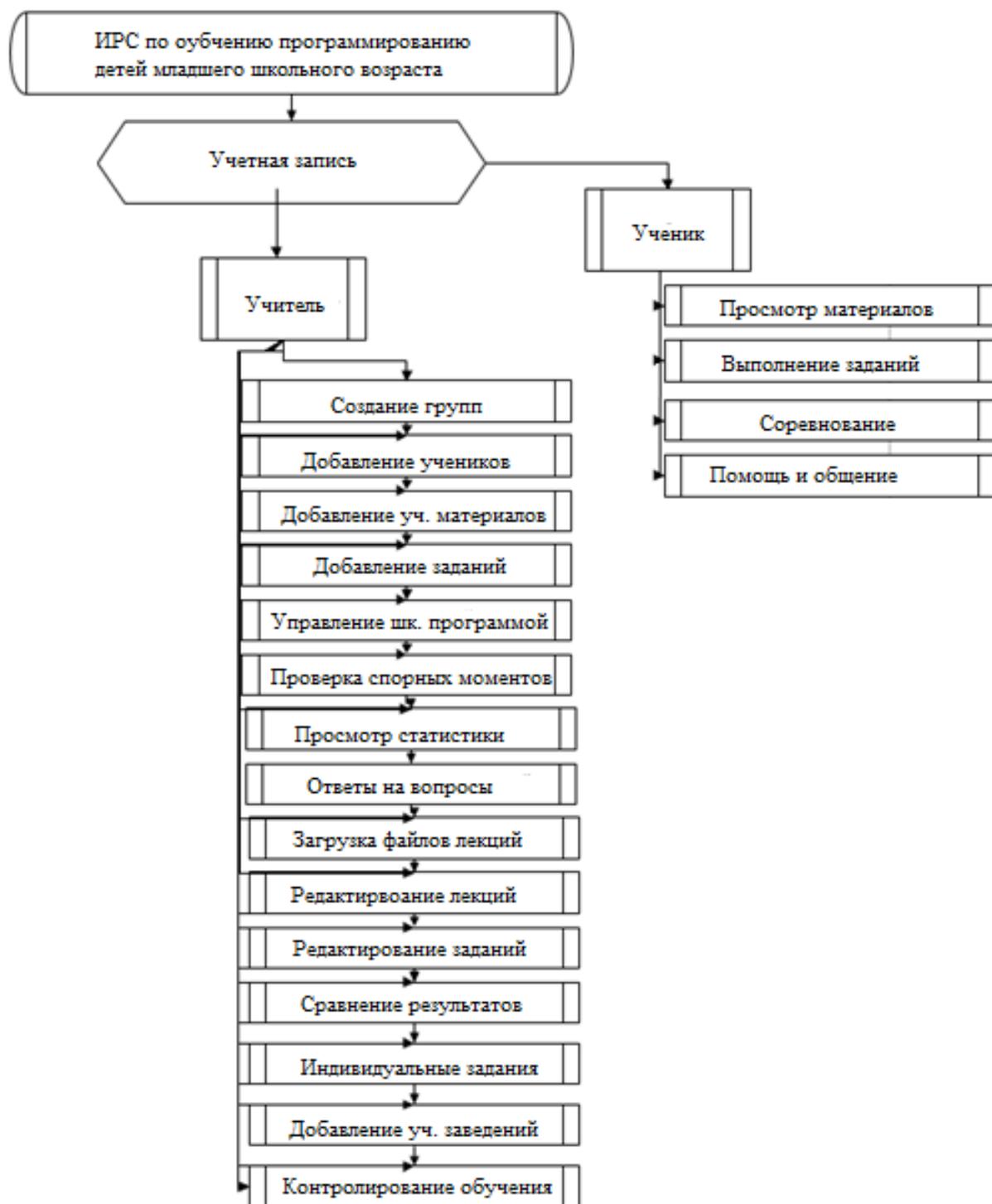


Рисунок 2.2.2 – Функциональная схема ИРС

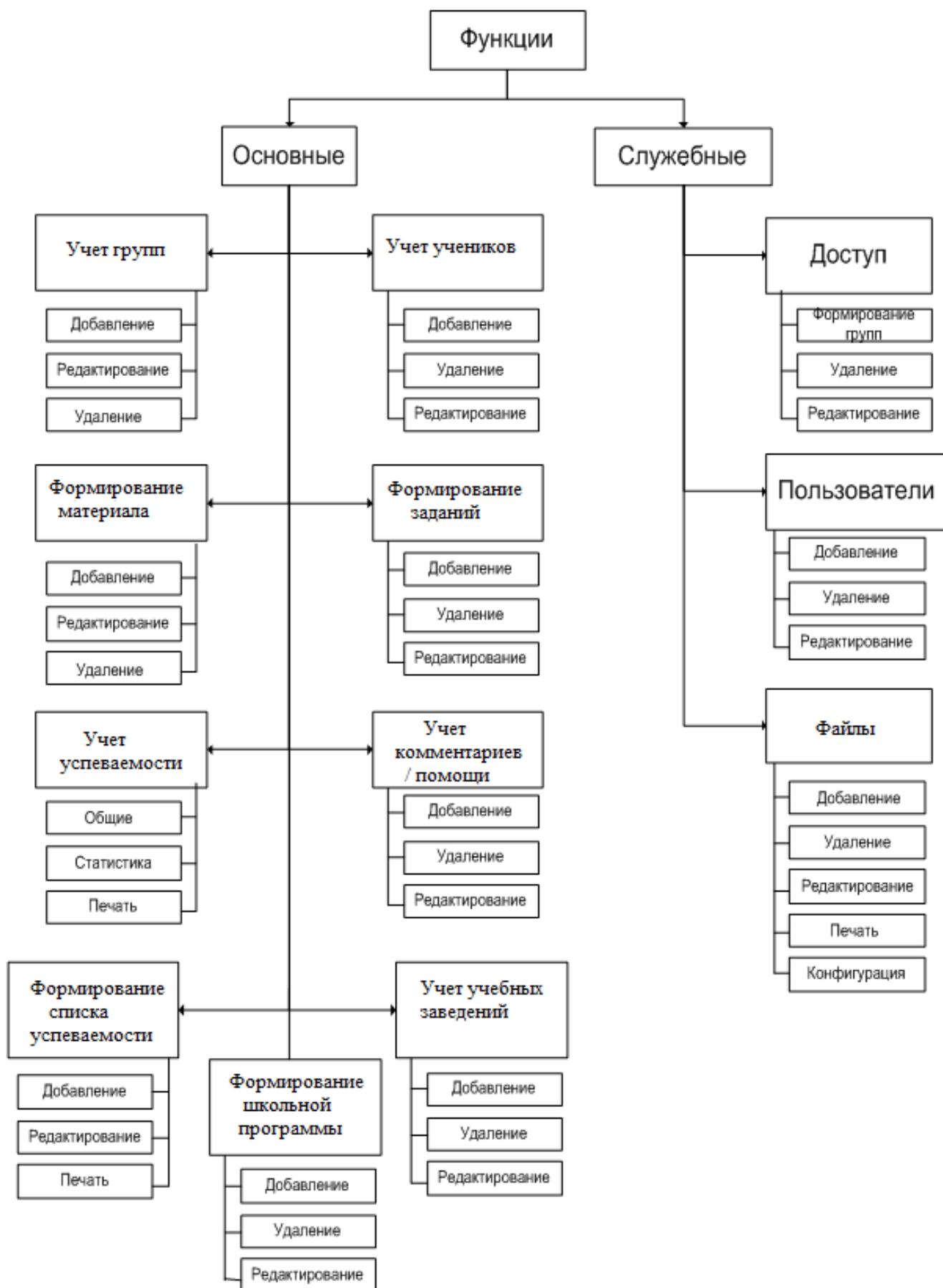


Рисунок 2.3.1 - Дерево функций системы для учителя

- добавлять новые пустые файлы в базу данных;
- изменять данные в созданных ранее файлах;
- получать или удалять данные из существующих и ранее созданных файлов;
- вводить новые данные в существующие файлы;
- удалять созданные файлы из базы данных.

База данных — это некоторый набор перманентных (постоянно хранимых) данных, используемых прикладными программными системами какого-либо предприятия.

Предприятием могут выступать любые научные, технические или коммерческие организации.

Концептуальное представление — когда описание всей информации, хранящейся в базе данных, носит более абстрактное представление (как и в случае внешнего представления) по сравнению с описанием физического способа хранения данных. Однако такой тип представления данных значительно отличается от представления данных какого-либо отдельного пользователя.

Концептуальное представление определяется с помощью концептуальной схемы, включающей определения для каждого существующего типа концептуальных записей.

Самый популярный метод семантического моделирования называется *ER-модель*, по-другому модель «сущность-связь». Работа начинается с определения четко идентифицированного понятия сущности (entity). При этом сущности подразделяются на обычные и слабые. Слабая сущность всегда зависима от другой, более сильной сущности и не может существовать без функционирования другой сущности.

Сущности и их связи обладают некоторым набором свойств (property). Для сущностей или связей одного и того же типа характерно обладание общими свойствами. Рассмотрим некоторые характеристики и особенности свойств сущностей:

Простое или составное свойство. Составное свойство – совокупность простых. Например, "имя работника" будет считаться составным, если под его значением подразумевается совокупность значений простых свойств "имя", "отчество" и "фамилия".

Ключевое свойство – это свойство, которое будет считаться уникальным только при рассмотрении его в определенном контексте. Например, имя человека, находящегося на иждивении у определенного сотрудника, может быть уникальным только в контексте данных об этом сотруднике.

Однозначное или многозначное свойство - в этой модели допускаются повторяющиеся группы.

Отсутствующее свойство - неизвестное или неприменимое свойство.

Базовое или производное свойство – вычисляемый элемент.

Связь (relationship) - система, объединяющая несколько сущностей.

Сущности, объединенные в общую связь, выступают ее участниками, а количество участников связи прямо указывают на ее степень[4].

При проектировании информационной системы были выявлены следующие сущности:

1. Учебные заведения – хранит информацию об учебных заведениях, зарегистрированных в ИРС и содержит следующие атрибуты:

- номер учебного заведения (идентификационный номер заведения);
- наименование (официальное название учебного заведения);
- город (наименование города, где зарегистрировано заведение);
- описание (дополнительная информация об учебном заведении).

2. Группы – хранит информацию о группах учеников для предоставления им учебного материала и ведением статистики усвоения группой материала и прохождения заданий. Содержит следующие атрибуты:

- номер группы (идентификационный номер группы учеников);
- имя группы (наименование группы внутри учебного заведения);
- номер учебного заведения (идентификатор учебного заведения, где располагается или проходит обучение данная группа учеников);

- описание (дополнительная необходимая информация о группе);
- номер учителя (номер пользователя, создавшего эту группу).

3. Пользователи – хранит информацию о всех зарегистрированных в ИРС пользователях и информацию, необходимую для авторизации, определяющую уровни доступа и возможности. Содержит атрибуты:

- номер пользователя (идентификатор пользователя в ИРС);
- логин (имя пользователя, необходимое для авторизации в ИРС);
- ФИО (фамилия, имя и отчество пользователя ИРС);
- пароль (данные, необходимые для авторизации в ИРС);
- электронная почта (адрес электронной почты для отправления уведомлений, восстановления доступа или напоминаний);
- обо мне (дополнительная информация о пользователе);
- Город и заведение (дополнительная информация при добавлении);
- номер группы (идентификатор группы, которой принадлежит пользователь или определяющий права доступа как учитель).

4. Учебные материалы – хранит информацию об учебных материалах, использующихся при обучении учеников с возможностью прикрепления к ним заданий, файлов и комментариев. Содержат следующие атрибуты:

- номер материала (идентификатор учебного материала в ИРС);
- номер учителя (идентификатор учителя, добавившего материал);
- название материала (тема или наименование учебного материала);
- текст материала (содержательная часть учебного материала, заполненная с помощью встроенного редактора материалов);
- документ материала (название прикрепленного файла-документа);
- дата материала (дата размещения или обновления материала);
- количество заданий (число прикрепленных к материалу заданий).

5. Задания – хранит информацию о заданиях, прикрепленных к учебным материалам, созданных с помощью внутреннего редактора заданий:

- номер (идентификатор прикрепленного к материалу задания);
- номер материала (материал, к которому прикреплено задание);

- текст задания (содержательная часть задания с пустыми полями в тексте, созданная с помощью внутреннего редактора заданий);

- текст ответов (правильные ответы на вопросы в задании, расположенные в порядке полей ввода и разделенные символами);

6. Пройденные материалы – хранит информацию об изученных учениками материалах для проверки доступа к заданиям. Содержит:

- номер (идентификационный номер просмотра уч. материала);
- номер материала (идентификационный номер уч. Материала);
- номер ученика (идентификационный номер обученного ученика);
- дата (дата освидетельствования нажатия на кнопки завершения).

7. Школьная программа – хранит информацию об алгоритме прохождения определенной группы определенных материалов. Содержит:

- номер программы (идентификационный номер шага обучения);
- номер шага (номер в последовательности изучения материалов);
- номер группы (идентификационный номер группы для обучения);
- номер материала (идентификационный номер материала, необходимого для изучения, чтобы перейти к следующему шагу).

8. Изображения – хранит информацию о путях изображений, использующихся в материалах или заданиях. Содержит атрибуты:

- номер изображения (идентификационный номер изображения);
- номер материала (идентификационный номер материала, содержащего графический элемент с файлом данного изображения);

- номер задания (идентификационный номер задания, содержащего графический элемент файла, использующего данное изображение);

- название изображения (наименование для использования к формированию пути файла для вставки изображения в содержательную часть задания или материала, где он используется);

- номер в очереди (число, какое место в очереди файлов изображений в этом материале или задании занимает данный файл).

9. Результат заданий – хранит информацию о результатах прохождения учениками заданий и начислении результатных очков. Содержит атрибуты:

- номер результата (идентификационный номер прохождения);
- Номер задания (идентификационный номер выполненного учеником задания, записывается только одобренная им попытка);
- номер ученика (идентификационный номер пользователя, выполняющего задание, прикрепленного к пройденному материалу);
- результат (количество очков, полученное за выполнение задания).

На рисунке 2.3.2 мы видим ER-модель разрабатываемой ИРС, составленную из выделенных сущностей.

2.3.3 Физическая модель данных

MySQL Community Server – наиболее известная и распространенная версия одной из самых популярных и известных СУБД с открытым кодом, которая находится в свободном доступе. Она сочетает в себе скорость, компактность, стабильность и удобство эксплуатации. Физическая структура на основе MySQL данных представлена в таблицах 2.3.1 – 2.3.9

Таблица 2.3.1 – «Учебные материалы»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
teach_id	Номер учителя	Внешний	Integer (11)	NOT NULL
material_name	Название материала	-	Varchar (120)	NOT NULL
material_text	Текст материала		LONGTEXT	NULL
material_doc	Вложенный файл	-	Varchar (180)	NULL
material_date	Дата обновления	-	Date	NOT NULL
cnt_quest	Количество заданий	-	Integer (11)	NULL

Таблица 2.3.2 – «Учебное заведение»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
build_name	Название заведения	-	Varchar (150)	NOT NULL
build_city	Название города	-	Varchar (150)	NOT NULL
build_desc	Описание заведения	-	TEXT	NULL

Таблица 2.3.3 – «Группы»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
group_name	Название группы	-	Varchar (100)	NOT NULL
group_build	Номер заведения	Внешний	Integer (11)	NOT NULL
group_desc	Описание группы	-	TEXT	NULL
teach_id	Номер учителя	Внешний	Integer (11)	NOT NULL

Таблица 2.3.4 – «Задания»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
material_id	Номер материала	Внешний	Integer (11)	NOT NULL
quest_text	Содержание задания	-	TEXT	NOT NULL
quest_answer	Ответы на задания		TEXT	NOT NULL

Таблица 2.3.5 – «Пользователи»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
login	Имя пользователя	-	Varchar (100)	NOT NULL
fiо	Фамилия И.О.	-	Varchar (120)	NOT NULL
pass	Пароль для входа	-	Varchar (50)	NOT NULL
email	Номер учителя	-	Varchar (120)	NOT NULL
aboutme	Описание себя	-	TEXT	NULL
cityandbuild	Город и заведение	-	Varchar (120)	NULL
who_id	Номер группы	Внешний	Integer (11)	NOT NULL

Таблица 2.3.6 – «Пройденные материалы»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
material_id	Номер материала	Внешний	Integer (11)	NOT NULL
user_id	Номер пользователя	Внешний	Integer (11)	NOT NULL
date	Дата прохождения		Date	NOT NULL

Таблица 2.3.7 – «Школьная программа»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
numb	Номер в очереди	-	Integer (11)	NOT NULL
group_id	Номер группы	Внешний	Integer (11)	NOT NULL
material_id	Номер материала	Внешний	Integer (11)	NOT NULL

Таблица 2.3.8 – «Результаты заданий»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
quest_id	Номер задания	Внешний	Integer (11)	NOT NULL
user_id	Номер пользователя	Внешний	Integer (11)	NOT NULL
result	Количество баллов	-	Integer (11)	NOT NULL

Таблица 2.3.9 – «Изображения»

Название поля	Описание поля	Ключ	Тип поля	Null значения
id	Идентификатор	Первичный	Integer (11)	NOT NULL
material_id	Номер материала	Внешний	Integer (11)	NOT NULL
quest_id	Номер задания	Внешний	Integer (11)	NOT NULL
name_pic	Название файла	-	Varchar (180)	NOT NULL
numb_pic	Номер в очереди	-	Integer (11)	NOT NULL

Схематично физическая модель данных представлена на рисунке 2.3.3. При разработке схемы, была использована программа «phpMyAdmin». Это делает возможным просмотреть типы данных для каждой таблицы.

Связи между выявленными информационными объектами определяются реальными отношениями между парами объектов, показанные в таблице 1.9.

№ Связи	Ключи связи	Главный объект	Подчиненный объект	Тип
1	Название группы	Учебное заведение	Группы	1:М
2	Номер группы	Группы	Пользователи	1:М
3	Номер учителя	Пользователи	Группы	1:М
4	Номер пользователя	Пользователи	Результаты заданий	1:М
5	Номер учителя	Пользователи	Материалы	1:М
6	Номер пользователя	Пользователи	Пройденные материалы	1:М
7	Номер материала	Материалы	Задания	1:М
8	Номер материала	Материалы	Пройденные материалы	1:М
9	Номер материала	Материалы	Школьная программа	1:М
10	Номер материала	Материалы	Изображения	1:М
11	Номер задания	Задания	Результаты заданий	1:М
12	Номер задания	Задания	Изображения	1:М
13	Номер группы	Группы	Школьная программа	1:М

Выводы по второй главе

Вторая глава бакалаврской работы включает в себя проектирование компонентов предлагаемой информационно-развлекательной системы с помощью построения концептуальной, логической и физической модели данных. Все это в совокупности позволяет выявить и оценить специфику моделируемой предметной области. Так же во второй главе были сформированы диаграммы, функциональная схема, дерево функций и проанализирована структура документов. Был выбран комплекс технических и программных средств реализации данной информационно-развлекательной системы для обучения программированию детей младших классов. Спроектирована функциональная схема информационной системы и дано описание основных функциональных модулей системы.

ГЛАВА 3. ЭКОНОМИЧЕСКАЯ ХАРАКТЕРИСТИКА ИНФОРМАЦИОННО-РАЗВЛЕКАТЕЛЬНОЙ СИСТЕМЫ

3.1 Расчет себестоимости ИРС

Расчет трудоемкости разработки ИРС:

Для быстрой, приближенной оценки трудоемкости и длительности разработки программы используется базовая модель, состоящая из формулы.

Трудоемкость разработки программного изделия (t) определяется по формуле (1), чел.-мес.

$$t=3,6*(\eta_{т.и.к})^{1,2}, \quad (1)$$

$$t=3,6*(1,164)^{1,2} = 5,4 \text{ (чел.-мес)}$$

$\eta_{т.и.к}$ - исходные команды (число, тысяч).

Численность персонала исполнителей:

Число исполнителей (среднее) рассчитывается из о заданных характеристик трудоемкости и длительности разработки ПИ по формуле (2), чел.:

$$Ч_{и}=t/T, \quad (2)$$

$$Ч_{и}=5,4/ 4,3 = 1,3$$

Условно будем использовать для разработки двух людей.

Длительность разработки ПИ (Т) рассчитывается по формуле (3), мес.:

$$T=3,6*t^{0,32}, \quad (3)$$

$$T=2,5* 5,4^{0,32} = 4,3 \text{ (мес.)}$$

Производительность труда группы разработчиков программного изделия (Π_p), исходных команд/чел.-мес. определяется по формуле (4):

$$\Pi_p=1000* \eta_{т.и.к} /t \quad (4)$$

$$\Pi_p=1000* 1,164/5,4 = 259,2 \text{ (команд/чел.-мес)}$$

В таблице 4.1 приведены характеристики программного изделия, входящего в ПО АС, или ПО АС в целом при оценке его как единого программного изделия, рассчитанные с использованием базовой модели.

Таблица 3.1 – Характеристики программного изделия ПО АС

Характеристики программного изделия
1
Количество символов ПИ, Т.И.К. – 1164
Трудоемкость t, чел.-мес. – 5,4
Необходимая производительность Пр, исх.ком/чел.-мес. – 259,2
Продолжительность работ Т, мес. – 4,3
Количество исполнителей, чел. – 2

Расчет затрат разработки ИРС

Затраты на разработку ИРС производится по формуле (5):

$$K_n = \Phi_{з/н} * [(1 + \beta_d)(1 + \beta_c) + \beta_u + \beta_{np}] + t_{\text{отладка}} * C_{\text{м-ч}} \quad (5)$$

где $\Phi_{з/н}$ - Зарплата работников (Фонд)

β_d - Дополнительная зарплата работников (коэффициент), 0,10..0,15

β_c - Отчисления на социальные нужды (коэффициент), 0,3

β_u - Накладные расходы (коэффициент), 0,6..0,8

β_{np} - Прочие расходы (коэффициент), 0,1..0,2

$t_{\text{отладка}}$ - Отладка ПИ (время)

$C_{\text{м-ч}}$ - стоимость машино-часа

$$K_n = 141845,6 * [(1 + 0,1)(1 + 0,3) + 0,6 + 0,1] + 153,6 * 31,2 = 306930,47 \text{ (руб)}$$

Зарплата других исполнителей

Официальная зарплата инженера-программиста третьей категории

16000руб. в месяц

Зарплата работников рассчитывается по формуле (6):

$$\Phi_{з/н} = \sum_{ij} t_{ij} * C \quad (6)$$

где $\sum_{ij} t_{ij}$ - трудоемкость работ по заданной ИС

$$\Phi_{з/н} = (16000 * 5,4) + (10000 * 5,4) = 68841,6 + 43026 = 141845,6$$

Таблица 3.2 – Должностной оклад исполнителей

Категория работника	Количество человек	Должностной оклад	Категория	Годовой фонд заработной платы
Web-мастер	1	16000	3	192000
Дизайнер	1	10000	2	120000

Расчет машинного времени на отладку программного обеспечения АС принимается равным 16% от времени разработки:

$$t_{эвм} = 8 * 22 * 5,4 * 0,16 = 153,6(\text{час})$$

Расчет себестоимости машино-часа работы ЭВМ

Стоимость машино-часа работы ЭВМ или комплекса средств автоматизации $C_{м-ч}$ рассчитывается на основе данных той организации, где выполнялась разработка программного продукты. Себестоимость машино-часа рассчитывается по формуле (7):

$$C_{м-ч} = \frac{Z_n + A + Z_э + Z_p + Z_n + Z_m}{\Phi_о} \quad (7)$$

где

Z_n - затраты на заработную плату обслуживающего персонала

A - годовая сумма амортизации

$Z_э$ - затраты на силовую электроэнергию

Z_p - затраты на ремонт и обслуживание оборудования

Z_n - накладные расходы

Z_m - затраты на материалы

$\Phi_о$ - действительный годовой фонд времени работы КСА

$$C_{м-ч} = \frac{6864 + 7320 + 6024,5 + 3660 + 3050 + 85107,4}{1793 * 2} = 31,24(\text{руб})$$

Расчет затрат на заработную плату обслуживающего персонала

Расчет затрат на заработную плату обслуживающего персонала рассчитывается по формуле (8):

$$З_n = \sum_{i=1}^n l_i * 12 * K_o * K_{с.н} \quad (8)$$

где

n - количество работников

l_i - месячный оклад

K_o - коэффициент дополнительной заработной платы, 1,1..1,2

$K_{с.н}$ - коэффициент отчислений на социальные нужды 1,26

Возьмем месячный оклад обслуживающего персонала равный 6000руб.

$$З_n = 8000 * 12 * 1,1 * 1,3 * 0,05 = 6864(\text{руб})$$

Расчет годовой суммы амортизации КСА

Годовые амортизационные отчисления по КСА рассчитываются по формуле (9):

$$A = \frac{C_{ксн} * H_a}{100} \quad (9)$$

где

$C_{ксн}$ - стоимость ЭВМ и прочего оборудования

H_a - норма амортизации 12%

$$A = \frac{61000 * 12}{100} = 7320(\text{руб})$$

Таблица 3.3 – Используемое оборудование

Наименование оборудования	Модель, серия	Количество	Стоимость, руб	
			Ед	Всего
1	2	3	4	5
Компьютер	Intel Core i5	2	23000	46000
Монитор	Samsung SyncMas.	2	5000	10000
ИБП	Samsung ElShield	1	2500	2500
Wi-Fi роутер	D-Link DIR-300	1	2500	2500
Всего				61000

Годовые затраты на электроэнергию (10):

$$Z_{\text{э}} = W_{\text{у}} * C_{\text{э}} * T_{\text{в}} \quad (10)$$

где

$W_{\text{у}}$ - установленная мощность

$C_{\text{э}}$ - стоимость силовой энергии

$T_{\text{в}}$ - время, в течение года, когда КСА потребляет энергию

$$Z_{\text{э}} = 0,4 * 4,2 * (1793 * 2) = 6024,5(\text{руб})$$

Расчет затрат на ремонт и обслуживание оборудования и материалы

Затраты на текущий ремонт и на материалы в год рассчитываются по формулам (11, 12) с коэффициентами затрат 4,5-10% от стоимости КСА.

Примем коэффициенты на ремонт равный 6,5%, а на материалы 7%

$$Z_{\text{р}} = 61000 * 0,06 = 3660(\text{Руб}) \quad (11)$$

$$Z_{\text{м}} = 61000 * 0,05 = 3050(\text{Руб}) \quad (12)$$

где

$Z_{\text{р}}$ -затраты на ремонт

$Z_{\text{м}}$ - затраты на материалы

Расчет накладных расходов

В накладные расходы включены затраты на оплату труда АУП, затраты на амортизацию и содержание площадей, затраты на отопление, освещение и

прочие. Коэффициент принимают равным 0,6..0,9 от затрат на основную заработную плату.

Накладные расходы рассчитываются по формуле (13):

$$Z_n = 141845,6 * 0,6 = 85107,4 \text{ (Руб)} \quad (13)$$

Расчет действительного годового фонда времени работы КСА

Годовой фонд времени устанавливается, исходя из номинального времени и времени профилактики оборудования и ремонтов. Рассчитывается по формуле (14):

$$\Phi_o = S * h * D * T_{np} \quad (14)$$

где

S - продолжительность смены

H - количество смен

D – число рабочих дней в году

T_{np} - время ремонтов и профилактики в год

$$\Phi_o = 8 * 1 * 249 * 0,9 = 1793 \text{ (ч.)}$$

Расчет цены разработки ПО АС

Минимальная цена разработки АС складывается из полных затрат на разработку K_n и минимально необходимой суммы прибыли, размер которой позволял бы на минимальном уровне осуществления самофинансирования организации-разработчика после всех обязательных платежей и выплаты налогов.

Рассчитывается по формуле (15):

$$Z_{\min} = K_n + \Pi_{\min} \quad (15)$$

$$Z_{\min} = 306930,47 + 61386,09 = 368316,57 \text{ (руб)}$$

Сумма прибыли рассчитывается, исходя из планируемого минимального уровня рентабельности затрат организации-разработчика по формуле (16):

$$\Pi_{\min} = K_n * \frac{R_{\min}}{100} \quad (16)$$

где

R_{\min} -минимальный уровень рентабельности, % (20-30%)

Примем уровень рентабельности равный 20%

$$П_{\min} = 30693048 * \frac{20}{100} = 61386,09 \text{ (руб)}$$

Вывод по третьей главе

В результате изучения экономической характеристики информационно-развлекательной системы, были рассчитаны основные затраты на разработку, включая затраты на заработную плату обслуживающего персонала, сумму амортизации, затраты на силовую электроэнергию, затраты на ремонт и обслуживание оборудования, накладные расходы, затраты на материалы и прочие расходы. При рекомендуемых двух разработчиках ИРС, с задачей справился лишь один, сэкономив некоторые затраты на разработку системы и уменьшив себестоимость системы.

Срок окупаемости разработанного ПО зависит от динамики популярности приложения и рынка рекламы.

По экономическим отчетам, расходы на разработку должны были составить более трехсот тысяч рублей, однако с использованием одного специалиста при его заработной плате, сумму удалось сократить более чем в два раза, что является положительным экономическим эффектом для заказчика.

ЗАКЛЮЧЕНИЕ

Основные задачи выпускной квалификационной работы — анализ существующих систем обучения детей программированию, выявление их недостатков, а так же разработка информационно-развлекательной системы для обучения программированию детей младшего школьного возраста выполнены. Проведенный анализ системы обучения позволил выделить четкие цели, назначение и задачи предлагаемой автоматизированной информационной системы. Так же было проведено сравнение разработанной ИРС с функционирующими на данный момент и структурированы результаты исследования. После выявления задач, которые должна разрешать предлагаемая ИРС, была определена технология проектирования системы. Далее проект заключался в поиске путей решений задач по информационному, техническому, программному и технологическому обеспечению. Для проектируемой системы была построена концептуальная информационная модель, а так же физическая и логическая схемы баз данных. Важным аспектом выпускной работы служит экономическое обоснование актуальности предлагаемого проекта. Рассчитанный ожидаемый экономический эффект от внедрения проекта позволяет предположить его рентабельность и экономическую выгоду. Оптимизация позволила сократить затраты на разработку ИРС. Срок окупаемости затрат на разработку и внедрение предлагаемой информационной системы составит немного меньше полутора лет. Разработанная информационно-развлекательная система обучения может быть введена в эксплуатацию со следующего учебного года, кроме того, может быть использована и на других, аналогичных системах за счет своей универсальности с минимальными изменениями или внедрением новых модулей в текущую разработку.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Государственные стандарты

1. ГОСТ 4.071.030. АСУП. Создание системы. Нормативы рабочего времени.

Научная и методическая литература

2. Бейли Линн, Моррисон Майкл. Изучаем PHP и MySQL / [пер. с англ.]. - М. : Эксмо, 2010. - 800 с. : ил. - (Мировой компьютерный бестселлер). с. 38-51

3. Бим-Бад Б.М. Педагогический энциклопедический словарь. М., 2002 г. с.89-90

4. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с.: ил. — Парал. тит. Англ с.536-538

5. Горюнов В. С. Информационные системы в образовании // Молодой ученый. — 2010. — №5. Т.2. — С. 159-161

6. Коростелёв А.И., Коростелёва О.Н. Процесс обучения, содержание учебного материала и методы обучения в сельскохозяйственном ВУЗе // Успехи современного естествознания. – 2007. – № 3. – С. 50-53

7. Матяш С. А. Корпоративные информационные системы с.110-111

8. Шахгельдян К.И., Крюков В.В., Гмарь Д.В. Система автоматического управления доступом к информационным ресурсам вуза//Информационные технологии. 2006.-№2.-с.19-29

9. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. — 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2010. — 912 с.: ил. + Видеокурс (на CD-ROM) — (Профессиональное программирование) с.1-3

Электронные ресурсы

10. <http://growandmanage.com/data-flow-diagrams-in-russian/>

11. <http://studepedia.org/index.php?vol=1&post=2122> - Проектирование модели системы программного обеспечения. ВВЕДЕНИЕ
12. <http://www.ssl.obninsk.ru/web/002/index.nsf/index/aos> - ЭНИМЦ Моделирующие Системы
13. Parag. D. Writing Effective Test Cases for Any Application, 2015 // uTest – The Professional Network for Testers [Электронный ресурс]: сайт о тестировании программного обеспечения: <https://www.utest.com/articles/writing-effective-test-cases-for-any-application>
14. Ладик Ю.В. Обзор IDE средств для программирования на Java, 2013 // JavaPortal.ru – Все о Java и JavaScript [Электронный ресурс]: сайт о разработке программного обеспечения: <http://www.javaportal.ru/projects/taidej/results.html>
15. Репин, В.В. Сравнительный анализ нотаций ARIS/IDEF и поддерживающих их продуктов, 2014 // iTeam.Ru – технологии корпоративного управления [Электронный ресурс]: сайт консалтинговой компании: https://iteam.ru/publications/it/section_51/article_2518

Литература на иностранном языке

16. Bailis, P. Readings in Database Systems – 5th edition, MIT Press, 2015.
17. Esposito, D. Architecting Applications for the Enterprise – 2nd edition, Microsoft Press, 2014.
18. Fusco, M. Java 8 in action – 2st edition, Pearson PLC, 2014.
19. Haridi, S. Concepts, Techniques, and Models of Computer Programming – 3rd edition, MIT Press, 2014.
20. Kalloniatis C. Innovative Information Systems Modelling Techniques – InTech Press, 2016.

Приложение А

Отдельные фрагменты кода для решения уникальных задач в ИРС

Файл конфигурации config/config.ini:

```
[database]
dbhost      = "localhost"
dbuser      = "root"
dbpass      = "sch001pr0g"
dbport      = ""
dbbase      = "schoolprog"
```

```
[tables]
table_users = "users"
table_group = "groups"
table_materials = "materials"
table_quests = "quests"
```

```
[folders]
f_images = "assets/images/"
```

Файл подключения к БД config/db_connect.php

```
<?
```

```
$aConfig = parse_ini_file('config.ini');
```

```
// Connect to Database
```

```
$dbhost      = $aConfig['dbhost'];
```

```
$dbuser      = $aConfig['dbuser'];
```

```
$dbpass      = $aConfig['dbpass']; //Необходимые данные БД
```

```
$dbport      = $aConfig['dbport'];
```

```
$dbbase      = $aConfig['dbbase'];
```

```
$tUser              = $aConfig['table_users'];
```

```
// Подключение к БД с помощью системы PDO
```

```
$oDb = new PDO("mysql:host=$dbhost;port=$dbport;dbname=$dbbase",
$dbuser, $dbpass);
```

```
$mysite = "http://projectwebwolf.ru/";
```

```
?>
```

Функция распределения студента в группы (фронтэнд-часть)

```
<script>
    function savestudent(who)
    {
        $.ajax
        ({ /*Используем ajax-запрос с обращением к php*/
            type : 'POST',
            url: 'functions/savestudent.php',
            data:                                {'who':who,
'value':document.getElementById("opt"+who).value},
            async: true, /* Асинхронный запрос */
            success: function(data)
            {

                document.getElementById("opt"+who).parentNode.parentNode.style="display
:none"; /* Устанавливаем невидимость выбранному ученику */
            }
        });
    }
</script>
```

Функция распределения студента в группы (бэкэнд-часть)

```
<?php
$who = $_POST["who"];
$value = $_POST["value"];
if($value > 0)
{ // Проверка на пришедшие значения из метода POST
    include "../config/db_connect.php";
    $prep = $oDb->prepare("UPDATE users set who_id = :value where id =
:who");
    $prep->execute(array(":who" => $who, ":value" => $value));
} //Устанавливаем необходимую группу выбранному студенту
?>
```

Вывод списка доступных материалов для изучения materials.php

```
<h1>Обучающие материалы</h1>
```

```
<?php if($_SESSION['who_id'] == "Учитель")  
{ ?>
```

```
<a href="add_material.php" class="btn btn-lg btn-  
warning">Добавить новый</a>
```

```
<?php }?>
```

```
<hr>
```

```
<?php
```

```
$stakematerials = $oDb->query("SELECT a.id as id,  
a.material_name as material_name, a.material_text as material_text, b.name_pic as  
name_pic, c.fio from material a inner join pics b on a.id = b.material_id inner join  
users c on a.teach_id = c.id where numb_pic = 1");
```

```
$rowMat = $stakematerials->fetchAll();
```

```
foreach ($rowMat as $row)
```

```
{ ?>
```

```
<div class="row">
```

```
<div class="well clearfix">
```

```
<div class="col-lg-3 col-md-2 text-center">
```

```
">
```

```
<p><?php print_r([".$row["fio"]."  
".$row["material_name"]]); ?></p>
```

```
</div>
```

```
<div class="col-lg-9 col-md-10">
```

```
<p>
```

```
<?php //Выводим отрывок
```

```
$thistext = substr($row['material_text'], 0,  
100) . '...';
```

```
if(stripos($thistext, "[img1]") !== false )
```

```
{
```

```
$takepic = $oDb->prepare("SELECT  
name_pic from pics where material_id = :id && numb_pic = 1");
```

```

$takepic->execute(array(":id" =>
$row['id']));

$pic = $takepic->fetch();
$thistext = str_replace("[img1]", "<img
class='pic' src='pics/' . $pic['name_pic'] . ' />", $thistext);
    } // Создаем путь к файлу картинки
    print_r($thistext);
    ?>
</p>
</div>

<div class="col-lg-12 col-md-12">
    <a href="seematerial.php?id=<?=$row['id']?>"
class="btn btn-lg btn-warning pull-right">подробнее</a>
</div>

</div>

</div>
<?>
?>

```

Вывод заданий у выбранной группы group.php

```
<?php /* Задания группы*/
```

```

$allmy = $oDb->prepare("SELECT a.id
as id, p.material_id as mId, a.material_name as mName, b.fio as fio from program p
inner join material a on p.material_id = a.id inner join users b on a.teach_id = b.id
where p.group_id = :group order by p.numb");
$allmy->execute(array(":group"=>$_GET['id']));
$allgroups = $allmy->fetchAll();
$tmp_n = 0;
foreach($allgroups as $gro)
{
    $tmp_n++;
    $cnt = $oDb->prepare("SELECT
count(*) as cnt from finish_material where material_id = :material and user_id in
(select id from users where who_id = :gro)");

```

```

        $cnt-
>execute(array(":material"=>$gro["id"], ":gro" => $_GET["id"]));
        $need_cnt = $cnt->fetch();

        ?>
        <tr>
        <td class="col-lg-1 col-md-1
col-xs-2"><?=$tmp_n?></td>
        <td class="vert-align"><a
href="show.php"><?=[".$gro["fio"]."] ".$gro["mName"]?></a></td>
        <td class="text-center vert-
align"><?=$need_cnt["cnt"]."/". $cnt_student?></td>
        <td class="text-center vert-
align">
        <select id="tictac"
name="takeme<?=$gro["mId"]?>" class="form-control">
        <?
        for($x=1;
        $x<=count($allgroups); $x++)
        {
                ?><option id="<?php
if($x == $need_cnt){print_r("selected");}
value="<?=$x?>"><?=$x?></option><?
        }
        ?>
        </select></td>
        </tr>
        <?
        }
    }
    ?>

```

Редактирование школьной программы учителем (бэкэнд-часть)

```

<?php
include('config/db_connect.php');
$cnt = count($_POST);
$i = 0;
$newarray = array(); // Создаем новый массив последовательности

```

```

foreach ($_POST as $ps)
{
    $newarray[$i] = $ps;
    $i++;
} //Вносим в новый массив последовательность просмотра
$i=1;
$serr = '0';
foreach($_POST as $takeme)
{
    if($i <= $cnt) //Пошаговая проверка значений со следующим
    {
        for($x = $i; $x<$cnt; $x++)
        {
            if ($takeme == $newarray[$x]){ $serr = '1';}
        } // Если значения последовательности совпадают
        $i++;
    }
}
if($serr == '1'){echo "<h3>Уникальность нарушена! Выберите правильный
порядок очереди материалов!</h3>";} // Вывод ошибки
else
{
    foreach ($_POST as $key=>$text)
    {
        $key_fin = str_replace('takeme','$key');
        $query = $odb->prepare("Update program set numb = :numb
where group_id = :groupid && material_id = :mId");
        $query->execute(array(":numb"=>
                                                                    $text*(-1),
":groupid"=>$_GET["group"], ":mId"=>$key_fin));
    }
    foreach ($_POST as $key=>$text)
    {
        $key_fin = str_replace('takeme','$key');
        $query = $odb->prepare("Update program set numb = :numb
where group_id = :groupid && material_id = :mId"); //Обновление
        $query->execute(array(":numb"=>
                                                                    $text,
":groupid"=>$_GET["group"], ":mId"=>$key_fin));
    }
}
}

```

Функция добавления картинки при создании материала

```
<script>
var pica = 1;
function addpic()
{
    var parent = document.getElementById("allpics");
    var pic = document.createElement("input");
    pic.setAttribute("type", "file");
    pic.id = "pic"+pica;
    pic.name = "pic"+pica;
    pica++;
    parent.appendChild(pic);
}
</script>
```

Подготовка данных для вывода задания во фронтэнде seequest.php

```
<?php session_start();
include('config/db_connect.php');
$id = $_GET["id"];
$itake_material = $oDb->prepare("SELECT now() as nw, a.* from quests a
where id = :id"); //берем необходимые данные и время
$itake_material->execute(array(":id" => $id));
$itake = $itake_material->fetch();
$whoami = $oDb->prepare("SELECT concat(a.fio,' [' ,b.group_name,']') as fio
from users a inner join groups b on a.who_id = b.id where a.id = :id"); // Берем
данные о текущем пользователе
$whoami->execute(array(":id" => $_SESSION["my_id"]));
$iFio = $whoami->fetch();
$temptext = explode("/", $itake['quest_text']);
$result_temp = "";
foreach($temptext as $tmp_key => $temp) //Преобразование текста
{
    if($tmp_key != 0)
    {
        $temp_text = explode("/", $temp);
        $result_temp .= " <input type='text' name=answer".$tmp_key.">
".$temp_text[1];
    }
}
```

```

        else
        {
            $result_temp = $temp;
        }
    }
}

```

Проверка выполненного задания quest_result.php

```

<?php
session_start();
include("config/db_connect.php");
$quest_id = $_GET['id'];
$my_id = $_SESSION['my_id']; //Берем ид пользователя сессии
$stakequest = $oDb->prepare("select quest_answer from quests where id =
:id");
$stakequest->execute(array(":id" => $quest_id));
$stakeanswer = $stakequest->fetch();
$answers = explode("###", $stakeanswer['quest_answer']); //разделяем
$i = 0;
$points = 0;
foreach ($_POST as $key => $pst)
{
    if(trim($pst) == $answers[$i]){ $points += 10;} //баллы++
    $i++;
}
print_r($points); exit;
$down = $oDb->prepare("Insert into material (teach_id, material_name,
material_text, material_doc, material_date) values (:teach_id, :material_name,
:material_text, :material_doc, now())");
$down->execute(array(":teach_id" => $teach_id, ":material_name" =>
$material_name, ":material_text" => $material_text, ":material_doc" =>
$docname_main));

?>

```