

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

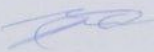
Направление подготовки: 09.03.01 «Информатика и вычислительная техника»
Профиль подготовки: «Программное обеспечение средств вычислительной
техники и автоматизированных систем»

БАКАЛАВРСКАЯ РАБОТА

на тему

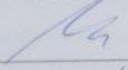
Обучающее Web-приложение для изучения
нейронных сетей

Студент  30.05.2025 (подпись, дата) Симонов Михаил Михайлович (ФИО полностью)

Руководитель 30.05.2025  С.А.Зинкин (подпись, дата) (фамилия, инициалы)

Нормоконтролёр  30.05.2025 (подпись, дата) А.С.Бычков (фамилия, инициалы)

Работа допущена к защите (протокол зас. кафедры от 28.05.2025 г. № 16)

Заведующий кафедрой  М.А. Митрохин (подпись) (фамилия, инициалы)

Работа защищена с отметкой  (протокол заседания ГЭК от

24 июня 2025 № 46)

Секретарь ГЭК  Е.И. Гурин (подпись) (фамилия, инициалы)

Пенза 2025

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Факультет
вычислительной техники

Кафедра
«Вычислительная техника»

«Утверждаю»
Заведующий кафедрой
М.А. Митрохин
«06» 05 2025г.

**ЗАДАНИЕ
НА БАКАЛАВРСКУЮ РАБОТУ**

1. Студент Симонов Михаил Михайлович гр.21ВВП2 факультета вычислительной техники.

2. Тема работы: «Обучающее Web-приложение для изучения нейронных сетей»

Тема утверждена приказом ПГУ №601/о от "06" мая 2025 г.

3. Руководитель бакалаврской работы д.т.н., профессор, каф. ВТ ПГУ Зинкин С.А.

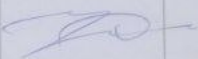
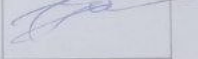
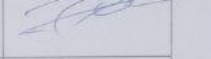
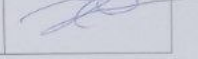
4. Задание на работу (назначение разработки, исходные данные и т.п.)

Разработать приложение «Обучающее Web-приложение для изучения нейронных сетей», позволяющее изучать основы нейронных сетей. Приложение должно обладать графическим и интуитивно понятным интерфейсом, для взаимодействия с пользователем. Веб-приложение должно предоставлять возможность изучения учебных материалов, прохождения тестирования и взаимодействия с личным кабинетом.

5. Перечень подлежащих разработке вопросов

Требуется произвести анализ предметных областей: изучить существующие обучающие веб-приложения, выделить их достоинства и недостатки. Выбрать язык программирования, который подойдет для создания интерфейса. Определить архитектуру приложения и вспомогательные средства разработки. Изучить функционал подходящего архитектурного паттерна.

6. Календарный график выполнения работы

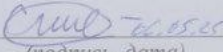
№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	Изучение предметной области	32	06.05.2025-08.05.2025	
2	Анализ требований	28	09.05.2025-10.05.2025	
3	Создание макета графического интерфейса	30	11.05.2025-13.05.2025	
4	Выбор технологий, применяемых при проектировании и разработке	40	14.05.2025-18.05.2025	
5	Реализация графического интерфейса	42	19.05.2025-22.05.2025	
6	Разработка основной части программы	70	23.05.2025-28.05.2025	
7	Оформление пояснительной записки	24	29.05.2025-30.05.2025	

Дата выдачи задания "06" мая 2025 г.

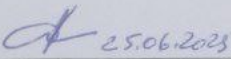
Руководитель бакалаврской работы _____

 06.05.2025 Зинкин С.А.
(подпись, дата)

Задание к исполнению принял студент _____

 06.05.2025 Симонов М.М.
(подпись, дата)

Работу к защите допустить
Декан факультета _____

 25.06.2025 Фионова Л.Р.
(подпись, дата)

Реферат

Бакалаврская работа содержит 145 листов, 51 рисунок, 3 источника, 3 приложения.

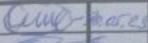
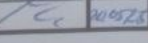
WEB - ПРИЛОЖЕНИЕ, ОБРАБОТКА СОСТОЯНИЙ, УПРАВЛЕНИЕ ПОЛЬЗОВАТЕЛЯМИ, ТЕСТИРОВАНИЕ, СИСТЕМА ТЕСТИРОВАНИЯ, БАЗА ДАННЫХ, LAMP — ОС GNU/Linux + AMP (Apache, MySQL, PHP)

Цель работы – разработать Web – приложение для изучения нейронных сетей, с большим уклоном в проверку полученных знаний.

Одним из путей достижения поставленной задачи является Web - приложение на основе системы тестирования, обеспечивающее выполнение следующих функций: авторизация пользователей, включая учеников и преподавателей, просмотр и редактирование личной информации учеников и преподавателей, добавление, изменение и удаление как тестов, так и вопросов в них, просмотр прогресса учеников, включая пройденные этапы обучения и результаты тестирования.

В конечном итоге разработано Web - приложение на основе системы тестирования, которая обеспечивает эффективную проверку знаний студентов. Информационная система полностью соответствующий цели данной выпускной квалификационной работы. В ходе достижения результата, были использованы следующие технологии: LAMP – ОС GNU/Linux + AMP (Apache, MySQL, PHP)

Разработка приложения закончена, приложение завершено и готово к эксплуатации.

<h3 style="margin: 0;">ПГУ1.090301.45.001 ПЗ</h3>				
Изм.	Лист	№ докум.	Подпись	Дата
Разраб.		Симонов М.М.		20.05.25
Провер.		Зинкин С.А.		20.05.25
Н. Контр		Бычков А.С.		20.05.25
Утверд.		Митрохин М.А.		20.05.25

Обучающее Web-приложение для изучения нейронных сетей. Пояснительная записка		
Лит.	Лист	Листов
1	2	67

ФВТ, каф. ВТ
гр. 21ВВП2

Содержание

Перечень сокращений и обозначений.....	4
1 Обучающее Web-приложение для изучения нейронных сетей.....	6
1.1 Анализ предметной области	6
1.2 Формы проведения компьютерного тестирования.....	7
1.3 Обзор существующих платформ для обучения	9
1.4 Анализ основных требований к системе	15
1.5 Сравнительный анализ СУБД и языков	17
2. Проектирование системы	20
2.1 Концептуальное проектирование	20
3 Разработка системы для онлайн тестирования	34
3.1 Архитектура программного обеспечения.....	34
3.2 Технологии и языки разработки системы.....	36
3.3 Структура программ (скриптов).....	37
3.4 База данных для системы онлайн тестирования	38
3.5 Основные алгоритмы работы системы	39
4. Администрирование системы тестирования	43
4.1 Администрирование базы данных.....	43
4.2 Администрирование сервера	45
4.3 Администрирование системы	48
5. Безопасность жизнедеятельности.....	58
5.1 Правила безопасности при использовании ПК.....	58
5.2 Правила безопасности при использовании смартфона	59
Заключение	61
Список использованных источников	63
Приложение А – UML - диаграммы.....	65
Приложение Б – Листинг программы	74
Приложение В - Презентация	117

Перечень сокращений и обозначений

В настоящей работе применяют следующие сокращения и обозначения

АИС – Автоматизированная информационная система

АС – Автоматизированная система

АРМ – Автоматизированное рабочее место

БД – База данных

ИС – Информационная система

ИО – Информационное обеспечение

ИТ – Информационные технологии

ОС – Операционная система

ПК – Персональный компьютер

ПЭВМ – Персональная электронно-вычислительная машина

ПО – Программное обеспечение

СУБД – Система управления базами данных

ЭВМ – Электронная вычислительная машина

ПК – Персональный компьютер

UML – (Unified Modeling Language) — унифицированный язык моделирования

IDEF0 – (Integration Definition for Function Modeling) – язык моделирования
бизнес-процессов

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		4

Введение

Актуальность темы данной выпускной квалификационной работы заключается в том, что в наше время технологии развиваются с невероятной скоростью, и интерес к таким передовым направлениям, как нейронные сети и искусственный интеллект, растет. Каждый день все больше людей стремятся изучать и осваивать эти новые технологии. В эпоху цифровой революции, когда постоянное самосовершенствование и обновление знаний становятся важной частью жизни, возникает необходимость в доступных и эффективных средствах обучения.

Разработка обучающего веб-приложения, посвященного нейронным сетям, является не только актуальной задачей, но и важным шагом к тому, чтобы дать широкой аудитории возможность освоить основные принципы работы этих сложных систем.

Это веб-приложение может стать полезным инструментом для студентов, изучающих информатику и смежные дисциплины, программистов, желающих расширить и улучшить свои профессиональные навыки, а также специалисты в других областях, таких как маркетинг, медицина или финансы, могут быть заинтересованы в использовании нейронных сетей для анализа данных и оптимизации процессов. Таким образом, создание такого приложения имеет значительный потенциал для внедрения в различные образовательные сферы.

Целью исследований выпускной квалификационной работы (ВКР), является разработка системы (инструментальной среды) для организации и проведения испытаний или, чаще всего говорят, тестирования знаний через сеть интернет.

Проблема исследования заключается в выборе и создании программного обеспечения, инструментальной среды для организации процессов удаленного тестирования знаний через сеть интернет - создания и проведения тестов, направленных на проверку знаний.

					ПГУ1.090301.45.001 ПЗ	Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

1 Обучающее Web-приложение для изучения нейронных сетей

1.1 Анализ предметной области

Для того чтобы разработать качественную и функциональную систему, необходимо проанализировать все аспекты работы обучающего Web – приложения и определить его потребности и требования. В рамках анализа были определены потребности и требования пользователей, а также выявлены задачи, которые могут быть оптимизированы с помощью обучающего приложения. На основе проведенного анализа были сформулированы требования к новой системе, которая должна быть удобной в использовании, безопасной и эффективной.

В следствии анализа было выявлено, что обучающее Web-приложение для изучения нейронных сетей представляет собой интерактивную платформу, предназначенную для повышения уровня знаний пользователей в области искусственного интеллекта и машинного обучения. В рамках данного приложения реализован функционал, позволяющий пользователям изучать теоретические аспекты нейронных сетей, а также применять полученные знания на практике через тестирование.

Анализ предметной области показывает, что нейронные сети становятся все более популярными в различных сферах, таких как обработка изображений, распознавание речи и анализ данных. С учетом быстрого развития технологий и увеличения объема информации, необходимы эффективные инструменты для обучения и проверки знаний. Обучающее Web-приложение решает эту задачу, предоставляя пользователям доступ к разнообразным учебным материалам, включая статьи, видеоуроки и интерактивные задания.

Для проверки полученных студентами знаний, можно использовать систему тестирования.

1.2 Формы проведения компьютерного тестирования

Системы проведения тестирований очень актуальны в настоящее время в связи с обучением по новым технологиям. Проведены исследования по методологиям и методам, используемым в тестировании знаний учащихся, исследователи и педагоги предлагают множество вариантов для технологии тестирования и оценки результата тестирования, которые могут зависеть от области знаний.

В этой связи перед разработкой инструментальных средств для организации тестирований, необходимо изучать множество теоретических источников и практических применений методов тестирования, методологии компьютерных тестирований знаний, как локального характера, так и основанных на использовании сети интернет, множество предлагаемых в сети программных продуктов, платформ для создания и проведения тестов одновременно для многих тестируемых (рисунок 1).



Рисунок 1 - Формы компьютерного тестирования

В первое множество тестов включаются тесты, для которых созданы специальные оболочки, поддерживающие ввод в систему тестирования определенных тестов в соответствии с определенными стандартами. Для функциональных оболочек таких систем характерно отличие в степени полноты и в интерфейсе тестируемых пользователей. Процедуры проведения тестирований, как правило, сводятся к заложенной последовательности при публикации вопросов тестов, получении ответов, их оценивания. При этом, как правило, каждый ответ оценивается по определенной бальной шкале, так «шкалируются» баллы за ответы переводом в одну из стандартных шкал. Результатом проведения тестов в любой системе она выдает заключение о полученных оценках и итоговых количествах баллов [2].

Во второй группе тестов относятся тесты и системы проведения тестирований для их поддержки, для которых по определенным алгоритмам генерируются версии тестов, которые создаются посредством инструментальных средств инструментальной системы тестирования. Текущая версия формируется, как на начальном этапе тестирования, так и в процессе проведения тестирований из заготовленных в базе данных набора стандартизированных тестов [2].

Тесты в третьей группе создаются на основе специальных адаптивных тестов. Здесь объем теста может сокращаться за счет удаления некоторых частей задания [2].

Технология проведения тестирований во многом зависит от той области знаний, в которой создается система тестирования и сами тесты. Например, для социологии и психологии, это могут быть тесты-опросы с множеством возможных ответов на каждый вопрос. Для данной работы выбран метод создания тестов и, соответственно, разрабатывается система тестирования, тестов опросов по первому способу.

1.3 Обзор существующих платформ для обучения

Перед началом разработки обучающего веб-приложения следует изучить и проанализировать существующие платформы.

Для рассмотрения аналогов были выбраны образовательные платформы «Stepik» и «Открытое образование».

Stepik - российская образовательная платформа и конструктор бесплатных и платных онлайн-курсов. Кроме того, каждый зарегистрированный участник имеет возможность разработать свои собственные обучающие уроки и онлайн-курсы.

Главная страница курса (рисунок 2). На главной странице собраны все основные сведения, касающиеся предмета изучения. Удобно структурированная информация помогает получить полное представление о том, что ждет пользователя в процессе обучения. Также на главной странице представлена программа курса, в которой описаны все этапы и модули, которые будет проходить пользователь. Помимо этого, можно также ознакомиться с отзывами участников, которые уже завершили данный курс.

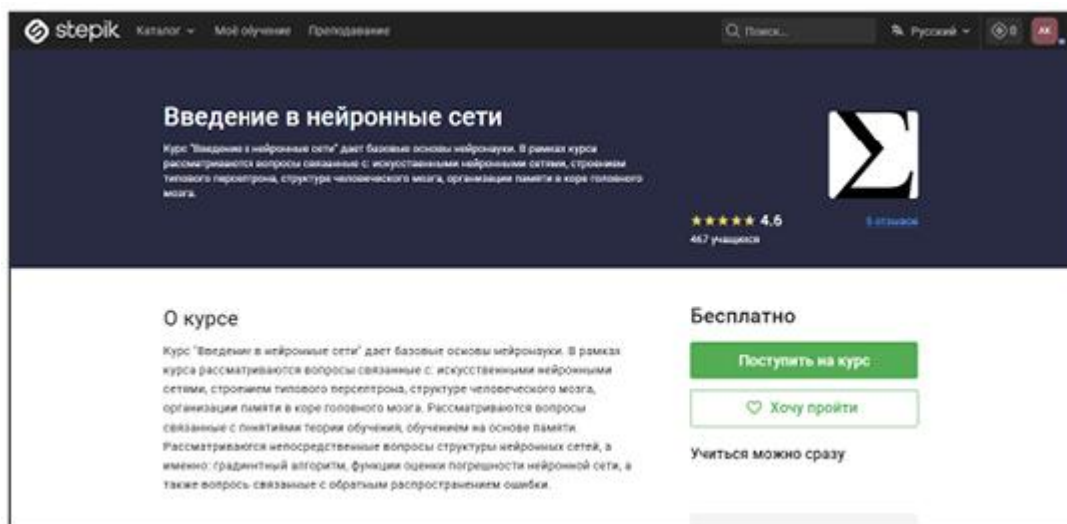


Рисунок 2 – Скриншот главной страницы

Страница с обучением (рисунок 3). Она имеет панель с разделами и темами, которая расположена слева. Остальную часть страницы занимает активное окно с самим процессом обучением (теория, видео, тесты).



Рисунок 3 – Страница с обучением

К плюсам данной платформы стоит отнести большой выбор курсов, удобный и понятный интерфейс, отслеживание прогресса, выдача сертификатов при окончании прохождения курсов, возможность создать и выложить свой курс. Из минусов следует выделить ограниченный выбор курсов на определенные темы и отсутствие возможности получить сертификат об окончании курса на бесплатной основе.

Открытое образование - современная образовательная платформа, предлагающая онлайн-курсы по дисциплинам, изучаемым в российских университетах.

Главная страница курса (рисунок 4). На данной странице представлены: актуальная информация о курсе, программа, которая позволяет оценить пользователем содержательность и структуру курса, отзывы, которые дают возможность получить объективное мнение о курсе от людей, уже прошедших его, а также требования, которые необходимы для комфортного освоения курса.

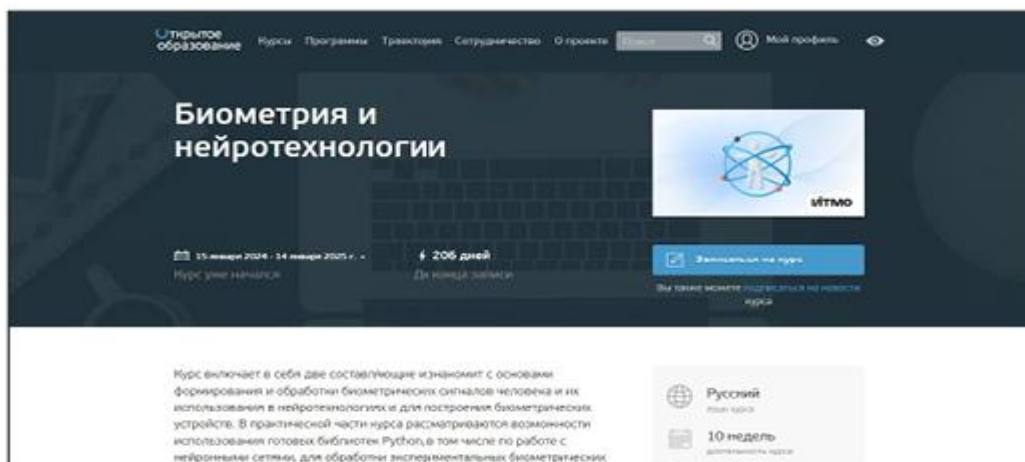


Рисунок 4 – Скриншот главной страницы курса на сайте

Страница с обучением (рисунок 5). На данной странице можно заметить, что практически весь доступный простор занят непосредственной учебным контентом, который является ключевым элементом страницы. Небольшой сегмент, расположенный в верхней части страницы, предназначен для навигационных элементов, которые служат для перемещения по разным разделам учебного материала. Эта область навигации играет важную роль, обеспечивая удобство и интуитивность использования сайта для пользователя.

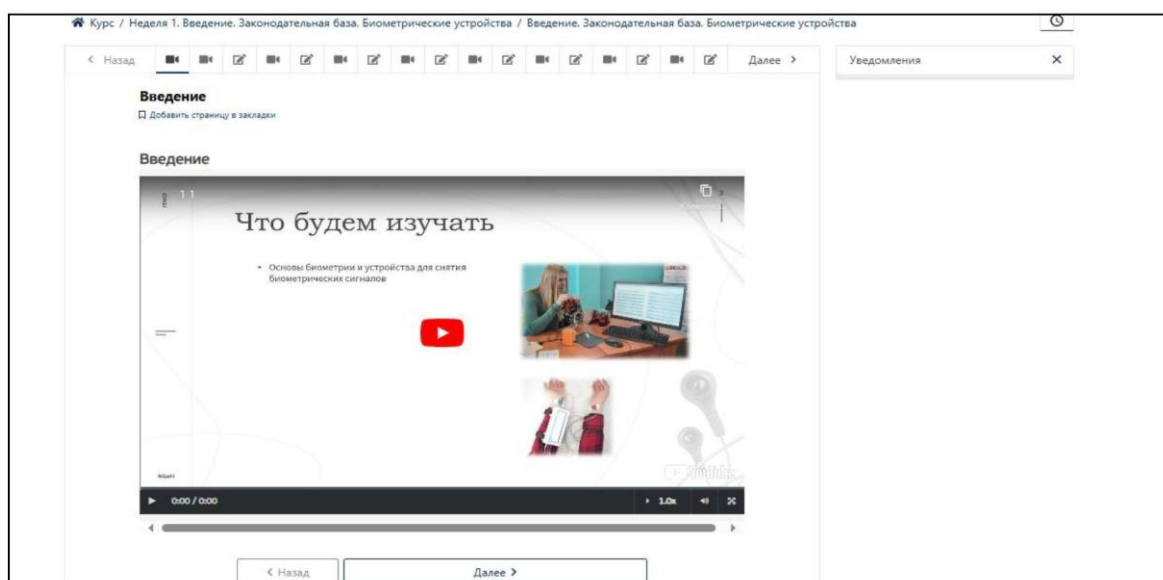


Рисунок 5 – Страница с обучением на сайте

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.45.001 ПЗ

Лист
11

Страница с тестами (рисунок 6). Практически после каждого теоретического материала или видео идет страница с тестом/опросом, который содержит один вопрос.

Опрос 1
[Добавить страницу в закладки](#)

Опрос 1
 1 point possible (ungraded)

К моторным проявлениям психических состояний относят:

☐ Мышечный тонус

☐ Мимику и жесты

☐ Голосовые характеристики

☐ Сердечно-сосудистый тонус

☐ Эмоциональные характеристики

Рисунок 6 – Тест на сайте

Страница с личным кабинетом (рисунок 7). На этой странице пользователь имеет возможность проводить необходимые изменения в своих персональных данных, тем самым поддерживая актуальность своих данных в системе, что способствует повышению уровня комфорта и удобства использования платформы.

Открытое образование

Курсы Программы Траектория Сотрудничество О проекте

Поиск

Мой профиль

Профиль

[Мои данные](#) [Пароль](#) [Аутентификатор](#) [Аккаунты соц.сетей](#)
[Сессии](#) [Приложения](#)

Имя пользователя

Пожалуйста, указывайте настоящие имя и фамилию, они будут вписаны в сертификат о прохождении курса

Фамилия *

Имя *

Отчество *

Рисунок 7 – Личный кабинет

Страница с курсами пользователя (рисунок 8). На данной странице пользователь может посмотреть текущие и завершённые курсы.

Открытое образование

Курсы Программы Траектория Сотрудничество О проекте

Поиск

Мой профиль

Анна Крылова

1 курс вы прошли на 3 курса вы записаны

Всего 3 курса

Текущие — 2 Завершённые — 1

Основы глубинного обучения

Рисунок 8 – Страница с курсами

К плюсам данного сайта стоит отнести удобный интерфейс, отслеживание прогресса и то, что курсы ведут профессора ведущих университетов страны и имеют широкий спектр различных направлений. К минусу можно отнести платный экзамен в конце обучения, чтобы получить документ о прохождении курса.

Учитывая, что одна из главных задач обучающих приложений, это проверить знания обучающихся, можно выделить ряд наиболее популярных и мощных основных платформ создания сценариев тестирования и проведения тестирований: GOOGLE формы, Quizlet, Proprofs, Kahoot!, ClassMarker, Plickers, Easy Test Maker.

Существуют специализированные платформы для создания и проведения тестирований для различных сфер знаний. Они, как правило, имеют и инструментарий для создания тестов, вопросов, ответов, в соответствии с определенными соглашениями систем, так и поддерживающие ряд различных технологий для проведения тестирований.

Для создания опросов, тестов можно использовать Google формы, которые входят в состав стандартного офисного инструментария Google Drive.

При использовании другого приложения - Quizlet, создаются тесты с возможностью выбрать правильные ответы из предлагаемых, заранее созданных, есть возможность сравнения рисунков и информации, давать свои варианты ответов.

Программа Proprofs является платным сервисом для создания онлайн тестов, который занимает лидирующую позицию среди сервисов такого рода.

Результаты анализа систем поддержки тестирования показывают, что большинство из них относятся к онлайн оболочкам тестов и развиваются в направлениях:

- предметно-ориентированные системы тестирования, разработанные в виде веб-приложений, сайтов;
- программные оболочки, настраиваемые на вопросы и ответы, связанные с любыми областями знаний.

Подавляющее число систем базируется на клиент-серверной архитектуре, с использованием серверов баз данных.

В большинстве инструментов и оболочек для тестирования знаний программное обеспечение строится на наборе очень сложных решений и

сложной программной реализации, требующей, в первую очередь, тщательного анализ и детальной проработки проблемы, и только затем, изучив сложную документацию, можно грамотно использовать ПО для создания тестов и проведения сеансов тестирования.

Как правило, задача заключается в наличие простых средств, инструментов для создания тестов и проведения тестирований, что является задачей настоящего проекта. Причем, большинство оболочек (инструментов) для создания тестовых сценариев и проведения тестирований основаны на клиент-серверной архитектуре программного обеспечения, что позволяет использовать систему одновременно несколькими пользователями.

1.4 Анализ основных требований к системе

Целью проекта является создание инструментальной среды (оболочки) для создания тестов и проведения компьютерных тестовых испытаний посредством использования web-технологий. Другими словами, необходимо разработать веб-ориентированную систему для создания тестов в соответствии с определенной концепцией тестирования и поддержки процесса тестирования знаний по определенной технологии тестирования, которая описана ниже.

Концепция тестирования состоит в следующем. Каждый тест состоит из множества вопросов, множества возможных ответов на каждый вопрос, где находится также и правильный ответ на поставленный вопрос. Для оценки результатов тестирования используется балльная шкала. За правильные ответы назначаются баллы и результат тестирования представляет сумму заработанных баллов и соответствующую этой сумме оценку по данному тесту.

Функциональные требования к системе состоят в следующем:

- разработанные с определенными соглашениями тесты должны гарантировать достоверную проверку знаний посредством использования системы, работа которой основана на этих соглашениях;

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		15

- возможность создания достоверных и адаптивных оценок результатов тестирования;
- простота создания тестов, удобство интерфейса для несложного создания и мобильного изменения;
- отсутствие зависимости программного обеспечения системы от определенной области знаний или отдельных предметов, а также зависимости от определенных аппаратных или программных возможностей использующих организаций;
- возможность одновременного прохождения тестирований по разным предметам, тестам (темам) неограниченного количества пользователей;
- легкость изменений количества и содержания предметов, тестов, вопросов и ответов;
- обеспечение высокой безопасности и сохранности данных - персональных данных тестируемых, предметам, тестам, результатам тестирования.

К первой группе пользователей системы относятся тестируемые - студенты. Ко второй группе относятся преподаватели, которые могут добавлять, обновлять, тесты, вопросы и ответы по своим тестам. Администратор, это ИТ-специалист, который занимается поддержкой системы, базы данных, консультацией специалистов по обновлению данных по тестам.

Преподаватель имеет доступ к результатам тестирования своих студентов. В форму на экране вводится каждый вопрос и несколько версий ответов, а также номер верного ответа и количество баллов, за верный ответ. Так система проходит и выдает все вопросы теста последовательно, но возможны и произвольные переходы между вопросами.

Система последовательно выводит на экран вопросы в виде формы в виде кнопок с ответами, либо в виде текстовой формы. Система должна определять тип выводимой на экран формы по данным базы данных и показывать тестируемому форму с каждым новым вопросом, вероятными

ответами и селекторами, если ответ предполагает выбор правильного ответа из множества. В форме тестируемый делает отметку напротив правильного, на взгляд тестируемого, ответа.

1.5 Сравнительный анализ СУБД и языков

Страницы в Интернете создаются с помощью языка разметки гипертекста — HTML. Недостатком таких страниц является то, что страницы получаются статическими. Значит в диалоге пользователь выбирает определенный раздел для получения информации со статическим контентом.

Для разработки веб-приложений используется множество разных технологий, которые отличаются площадками, на которой они исполняются: сервер приложений или компьютер клиента. К примеру, серверные технологии, это такие скрипты и языки программирования, как PHP, Perl, технологии ASP.NET, CGI-BIN. К клиентским относятся всевозможные встроенные скрипты, например, Javascript, VBScript, а Java Applets.

Сценарий (скрипт), это программный код, обеспечивающий работу интерактивных и динамических элементов Web-сайта.

Сценарии делятся на:

- серверный, это программа, выполняющаяся на Web-сервере, используются, например, для отображения в виде Web страниц содержимого баз данных, хранящихся на сервере;
- клиентский, это программа, выполняющаяся с помощью интерпретатора браузера и предоставляющая гораздо больше возможностей чем HTML по управлению интерфейсом посредством динамических страниц.

PHP был создан специально для использования в сетевых приложениях. На этом языке разрабатываются динамические сценарии, реализующие вместе логику работы web-приложения, например, форумы, счетчики посещений, поисковые системы.

В языке PHP реализована концепция объектно-ориентированного подхода (программирования, ООП), являющаяся актуально-востребованной концепцией индустрии современного программирования.

Базой ООП в языке PHP определяется совокупность объектов классов. Класс определяется, как закрытая совокупность функций (методов класса) и переменных, для определения шаблонов объектов с определенной логикой. Шаблоны используются для создания и использования в программах экземпляров этих шаблонов.

Предпочтение было отдано PHP, так как специалисты считают его основным языком для разработки систем, веб-приложений динамического, интерактивного характера.

Для веб-ориентированной системы тестирования знаний была обоснована и принята архитектура «клиент-сервер».

Если в коде есть SQL-запросы, то сервер приложений обращается к серверу баз данных для выполнения запросов, получения результата скриптами и дальнейшего использования в программном коде.

Трехуровневая архитектура представлена ниже (рис.9). В итоге ответ на запрос браузера может содержать только сформированный сервером HTTP, CSS, JavaScript код, интерпретируемый затем браузером.

Для разработки веб-ориентированной системы, выбран стек программных средств LAMP — ОС GNU/Linux + AMP (Apache, MySQL, PHP).

Веб-сервер Apache, СУБД MySQL и язык программирования PHP, используемый для создания веб-приложений, выбраны в связи с тем, что указанные программные средства являются бесплатными, широко распространёнными и всесторонне апробированными.

В качестве СУБД выбрана MySQL, которая обеспечивает гибкость в поиске, быструю настройку форматов вывода; хорошую скорость примногопользовательском режиме работы; обеспечивает быстроту удаленного доступа, высокую производительность и простоту использования.

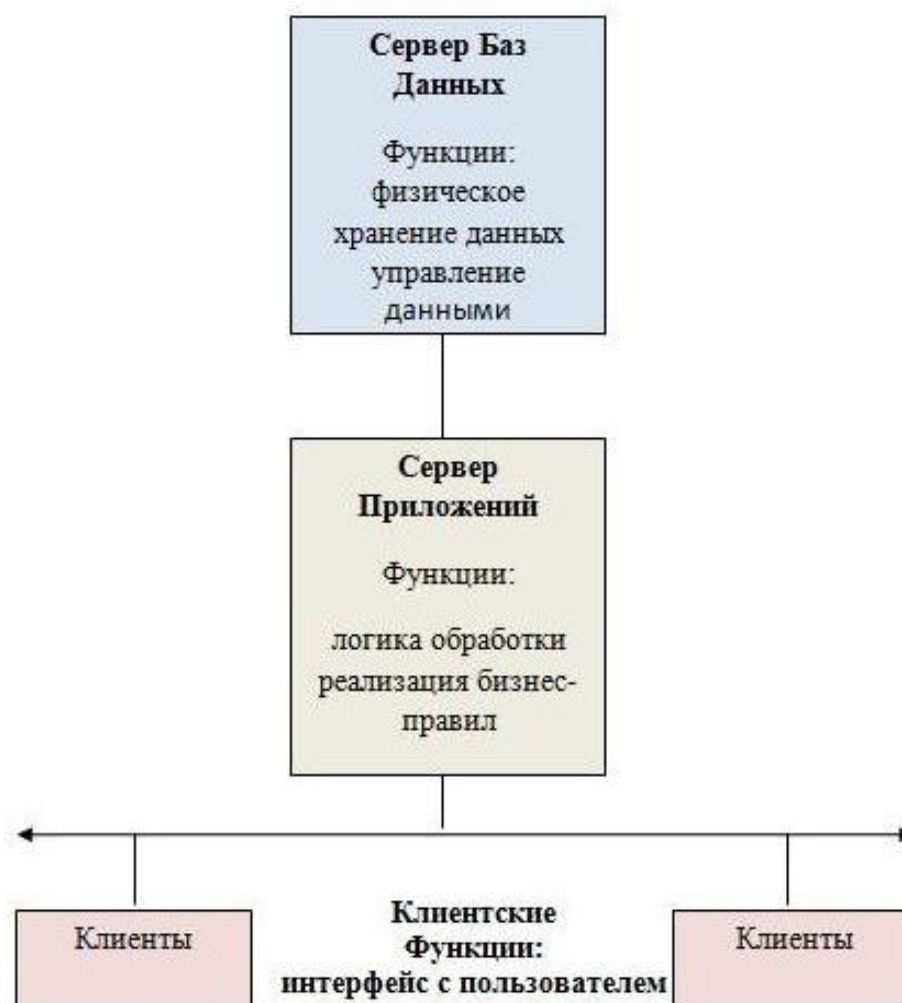


Рисунок 9 – трех уровневое клиент – серверное приложение

RHPRMyAdmin – веб-приложение с открытым кодом на языке PHP. Оно представляет разработчикам веб-интерфейс администрирования СУБД MySQL в виде таблиц и форм ввода данных.

2. Проектирование системы

2.1 Концептуальное проектирование

Целесообразно проектировать систему сверху вниз – от концепции тестирования до разработки базы данных и алгоритмов, реализующих концепцию.

В первую очередь необходимо принять определенную концепцию тестирования знаний. Что имеется ввиду. Есть перечень вопросов, на которые тестируемый должен отвечать. Анализ систем тестирования, проведенный в предыдущих разделах, показал, что вопросы могут задаваться последовательно, случайным образом, можно разрешить тестируемому возвращаться и давать новый ответ, вопросы могут задаваться в виде изображений.

Ответы тоже могут быть различного вида. Например, для сферы медицины и психологии ответы могут быть не однозначными, а предполагать сразу несколько вариантов ответов, ответ может состоять не в выборке одного предложенного варианта, а нескольких вариантов.

Технологические аспекты могут также заключаться в том, хранит ли система результаты тестов в базе данных или сохраняет их только на сеанс, только один вариант для каждого пользователя или хранит весь архив – кто и когда проходил тестирование и каковы результаты.

Все это определяет последующие алгоритмы и структуру базы данных.

Схему последовательности разработки системы можно представить в виде, данном на рисунке 10.



Рисунок 10 – Схема нисходящего проектирования системы тестирования

Из всех перечисленных концепций проведения тестирования выбраны наиболее общие и важные. В концепции тестирования в системе предусмотрены несколько ключевых аспектов. Система задает вопросы в текстовом виде, последовательно предлагая все вопросы из базы данных. Тестируемый выбирает один из представленных системой вариантов ответов. За верный ответ система начисляет один балл и информирует о правильности ответа, предоставляя также правильный ответ. Результаты тестирования фиксируются, и за верные ответы начисляются баллы, при этом система подсчитывает количество правильных ответов. Результаты тестирования

сохраняются на сеанс работы, и если начинается новый сеанс, результаты предыдущего сеанса удаляются системой. Что касается концепции самой системы тестирования, она имеет клиент-серверную архитектуру и функционирует в сети интернет, поддерживая независимую работу множества пользователей. Пользователи делятся на три группы. Первая группа — это администраторы системы, которые следят за работоспособностью системы, а также за сохранением и безопасностью базы данных. Вторая группа — преподаватели, которые регистрируются в системе и могут просматривать результаты всех тестов пользователей, а также добавлять, обновлять и удалять предметы, тесты по каждому предмету, вопросы и варианты ответов. Третья группа — тестируемые (студенты), которые могут регистрироваться в системе самостоятельно или через преподавателя, получая логин и пароль для входа. Они выбирают предмет и тест, проходят тестирование, отвечая на вопросы, выбирая ответы из предложенных системой вариантов. Результаты тестирования хранятся до следующего входа, и при начале нового тестирования результаты предыдущего сеанса удаляются системой.

Исходя из данной концепции проведения тестирования, были разработаны интерактивные формы пользователей системы.

2.2 Проектирование интерфейса пользователей

Варианты использования системы группами пользователей можно представить в виде UML диаграмм. Диаграммы представлены с учетом дальнейшего развития системы. В диаграммы добавлены те функции, которые планируется добавить при развитии системы и после ее опробования, когда будет понятно, какие функции реально необходимы для функционирования системы и для удобства различных групп пользователей.

Диаграмма для посетителя, незарегистрированного пользователя представлена на рисунке 11.

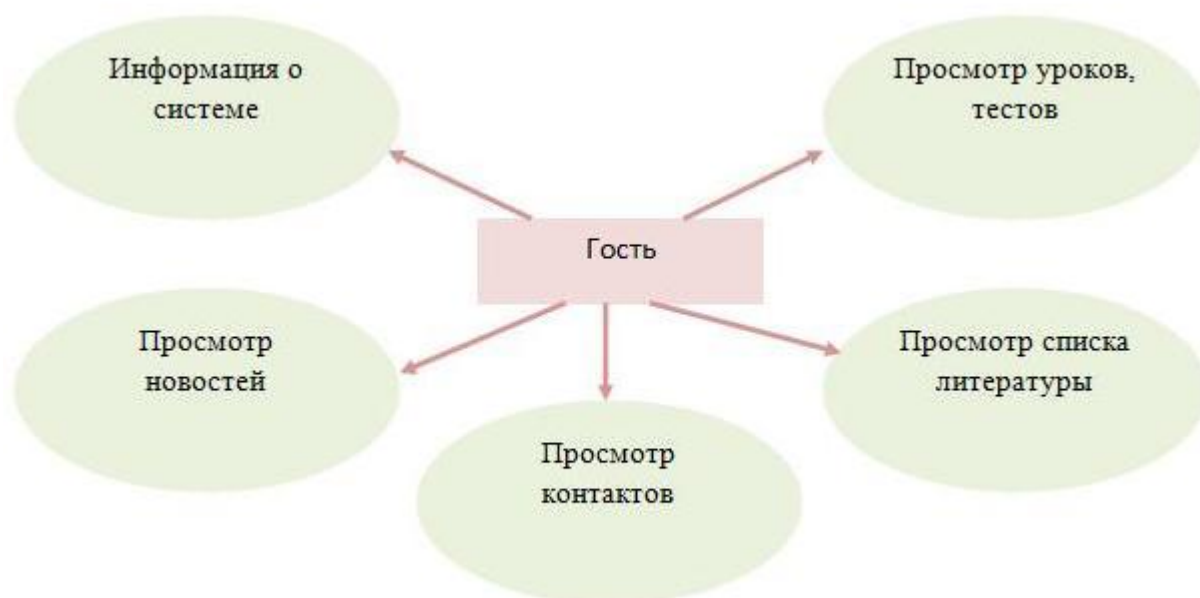


Рисунок 11 – UML – диаграмма вариантов использования системы посетителем

Диаграмма вариантов использования для зарегистрированных пользователей тестируемых студентов представлена на рисунке 12. Главными функциями при этом являются регистрация и авторизация пользователей, прохождение вопросов по тестам, просмотр результатов тестирования посредством личных кабинетов пользователей, просмотр личных данных в кабинетах пользователей.



Рисунок 12 – UML – диаграмма вариантов использования для тестируемых

Диаграмма вариантов использования для пользователей преподавателей представлена на рисунке 13. Главными из данных функций являются просмотр результатов тестирования студентов через личные кабинеты преподавателей.



Рисунок 13 – UML – диаграмма вариантов использования для преподавателя

Диаграмма вариантов использования для администратора системы представлена на рисунке 14.



Рисунок 14 – UML – диаграмма вариантов использования для администратора

Для интерактивности системы используются следующие формы, которые обрабатываются php-скриптами:

- форма для регистрации пользователей;
- форма для авторизации пользователей;
- формы для добавлений и обновлений данных таблиц базы данных системы, которые используются администратором и преподавателями;
- ряд форм, с помощью которых проводится тестирование пользователей-тестируемых;
- ряд интерактивных форм создан и используется для проведения тестирования в соответствии с концепцией тестирования, принятой в системе.

Исходя из концепции тестирования, разработаны интерактивные формы, посредством которых реализуется концепция проведения тестирования для

пользователя. Форма вопросов и ответов представлена кнопочной формой выбора ответа на рисунке 15.

| 1 | 2 | 3 | 4 |

ВОПРОС: Что такое нейронная сеть?

Перечень возможных ответов	Отметьте правильный ответ
а). Модель, имитирующая работу человеческого мозга для решения задач.	☐
б). Компьютерная программа, которая выполняет арифметические операции.	☐
в). Устройство для хранения данных	☒
не знаю	☐

Мой ответ

Рисунок 15 – Форма вопросов и ответов

Форма ответа системы с количеством заработанных баллов и правильным ответом на вопрос представлена на рисунке 16.

ВОПРОС: Что такое нейронная сеть?

Правильный ответ на вопрос:	Данный вами ответ на вопрос:
а). Модель, имитирующая работу человеческого мозга для решения задач.	а). Модель, имитирующая работу человеческого мозга для решения задач.
Количество очков за правильный ответ: 1	Количество очков за ваш ответ: 1

Ваш ответ занесен в базу данных

Следующий вопрос

Рисунок 16 – Форма ответа системы

Для регистрации пользователя, регистрации логина и пароля для входа в систему разработана форма на рисунке 17.

РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЕЙ

Регистрация пользователей подразумевает создание логина и пароля с целью входа в защищенные области сайта. Зарегистрированные пользователи-учащиеся, могут затем пройти тестирование и просмотреть результаты тестирования. Для этого они должны авторизоваться и зайти в свой личный кабинет пользователя. Зарегистрированные преподаватели могут просматривать результаты тестирования всех учащихся.

Наименования полей	Заносимые или обновляемые данные
Код учащегося	Simka
Пароль	Simka
ФИО	SimkaSimM
Факультет	SimkaMatika
Группа	SimkaGroup
Год Поступления	Simka2030
Специализация	Simka09
Наименование	SimkaIT
Телефон	+8963Simka
Электронный адрес	Simka@mail.ru
<input type="button" value="ДОБАВИТЬ"/>	

Рисунок 17 – Форма регистрации пользователя

В данном случае поля формы возможно избыточны, но далее при описании программного обеспечения, это можно изменить, синхронно меняя базу данных и табличные представления.

Многие входные и выходные данные системы представляются пользователям в виде таблиц с заголовками столбцов (полей таблицы) и данными внутри таблицы.

Данные результатов тестирования для тестируемого представляются в виде таблиц. В системе в виде таблиц представляются многие данные, например, списки преподавателей, ответов на вопрос, списки вопросов, ответов и пр.

Для администратора системы, преподавателей, результатов тестирования для студентов представление данных во многом базируется на представлении данных в виде таблиц определенного вида.

Таблица представления данных по результатам тестирования представлена на рисунке 18.

РЕЗУЛЬТАТ ПОИСКА									
Код ответа	№ предмета	Предмет	Тест	Код учащегося	№ вопроса	Наименование вопроса	Ответ тестируемого	Правильный ответ	Очки
211Sim	2	Продвинутый курс	1	Sim	1	Что такое переобучение (overfitting) в контексте нейронных сетей?	2) Когда модель слишком сложна и хорошо работает на обучающих данных, но плохо на тестовых.	2) Когда модель слишком сложна и хорошо работает на обучающих данных, но плохо на тестовых.	1
111Sim	1	Основы Нейронных Сетей	1	Sim	1	Что такое нейронная сеть?	а). Модель, имитирующая работу человеческого мозга для решения задач.	а). Модель, имитирующая работу человеческого мозга для решения задач.	1
212Sim	2	Продвинутый курс	1	Sim	2	Что такое сверточные нейронные сети (CNN)?	4) Нейронная сеть, использующая только один слой.	2) Тип нейронной сети, специально разработанный для обработки изображений и видео.	0

Рисунок 18 – Таблица результатов тестирования (ответов на вопросы)

Администратору доступны кнопки управления этими таблицами. Они располагаются внизу каждой таблицы, если администратор входит в личный кабинет и затем выбирает для просмотра любую таблицу базы данных.

В качестве примера представлена форма, с помощью которой добавляются или обновляются данные по вопросам, на рисунке 19.

Наименования полей	Заносимые или обновляемые данные
Код вопроса	212
№ вопроса	2
№ предмета	2
Предмет	Продвинутый курс
Тест	1
Наименование вопроса	Что такое сверточные нейронные сети (CNN)?
Первый ответ	1) Тип нейронной сети, используемый для обработки последовательностей данных.
Второй ответ	2) Тип нейронной сети, специально разработанный для обработки изображений и видео.
Третий ответ	3) Нейронная сеть, которая не требует обучения.
Четвертый ответ	4) Нейронная сеть, использующая только один слой.
№ верного	2
Очки	1
Добавить	Удалить
Изменить	Просмотр

Рисунок 19 - Форма добавления или обновления данных «ВОПРОСЫ»

Интерфейс пользователей предполагает определение последовательности работы пользователей с системой. Последовательность работы тестируемого представлена на рисунке 20.

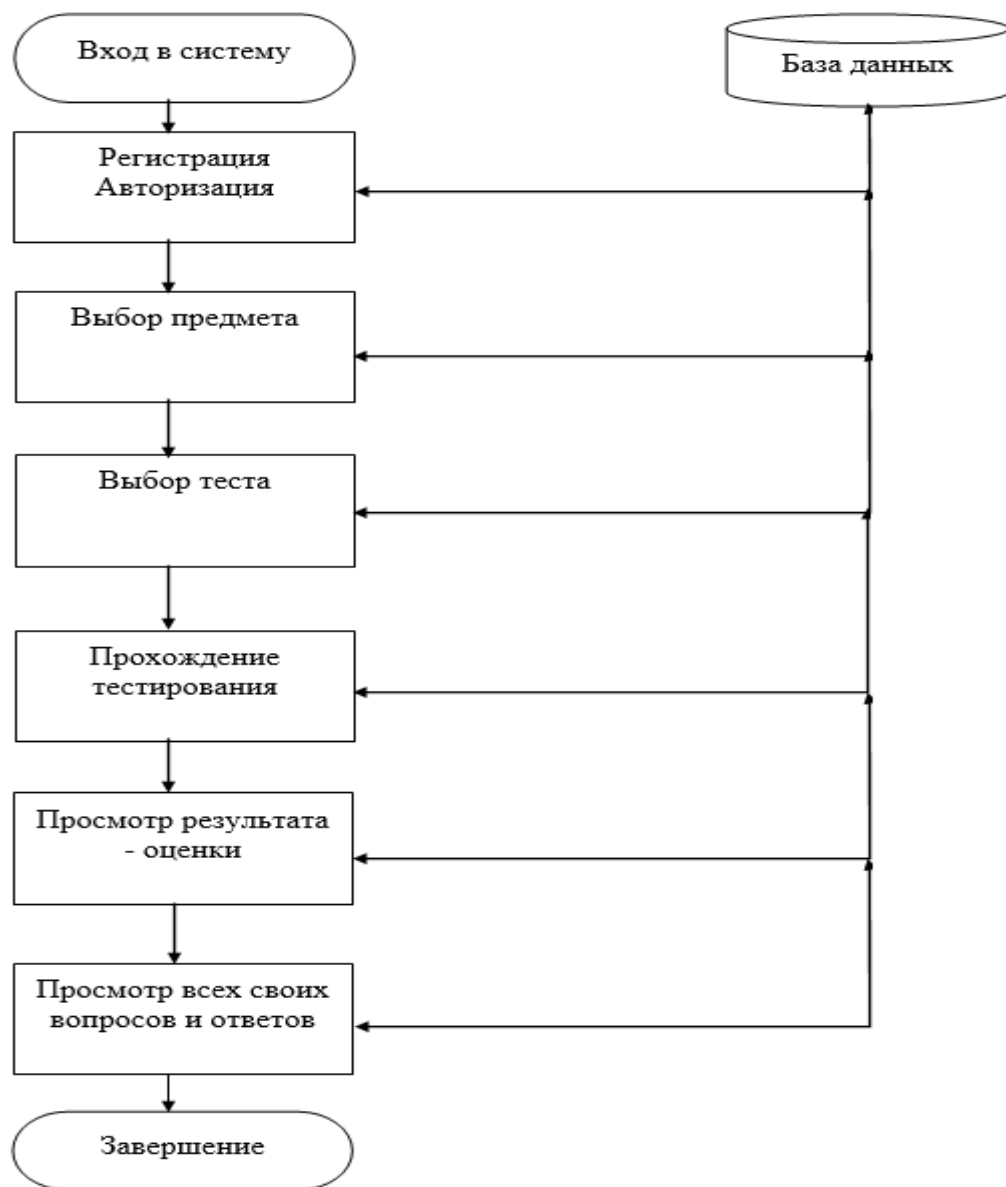


Рисунок 20 – UML - диаграмма последовательности работы тестируемого

Последовательность работы преподавателя представлена на рисунке 21.



Рисунок 21 – UML - диаграмма последовательности работы преподавателя

2.2 Проектирование базы данных системы

Как говорилось в первой главе, информация в используемой базе данных MySQL представляется с помощью таблиц. Для пользовательского интерфейса такое представление информации на его мониторе является важной составляющей.

Инструментом проектирования и создания базы данных, построения ER-диаграммы базы данных используется инструмент MySQL Workbench. Этот инструмент предоставляет средства проектирования, создания баз данных и их администрирования.

Физическая схема базы данных системы тестирования представлена на рисунке 22.

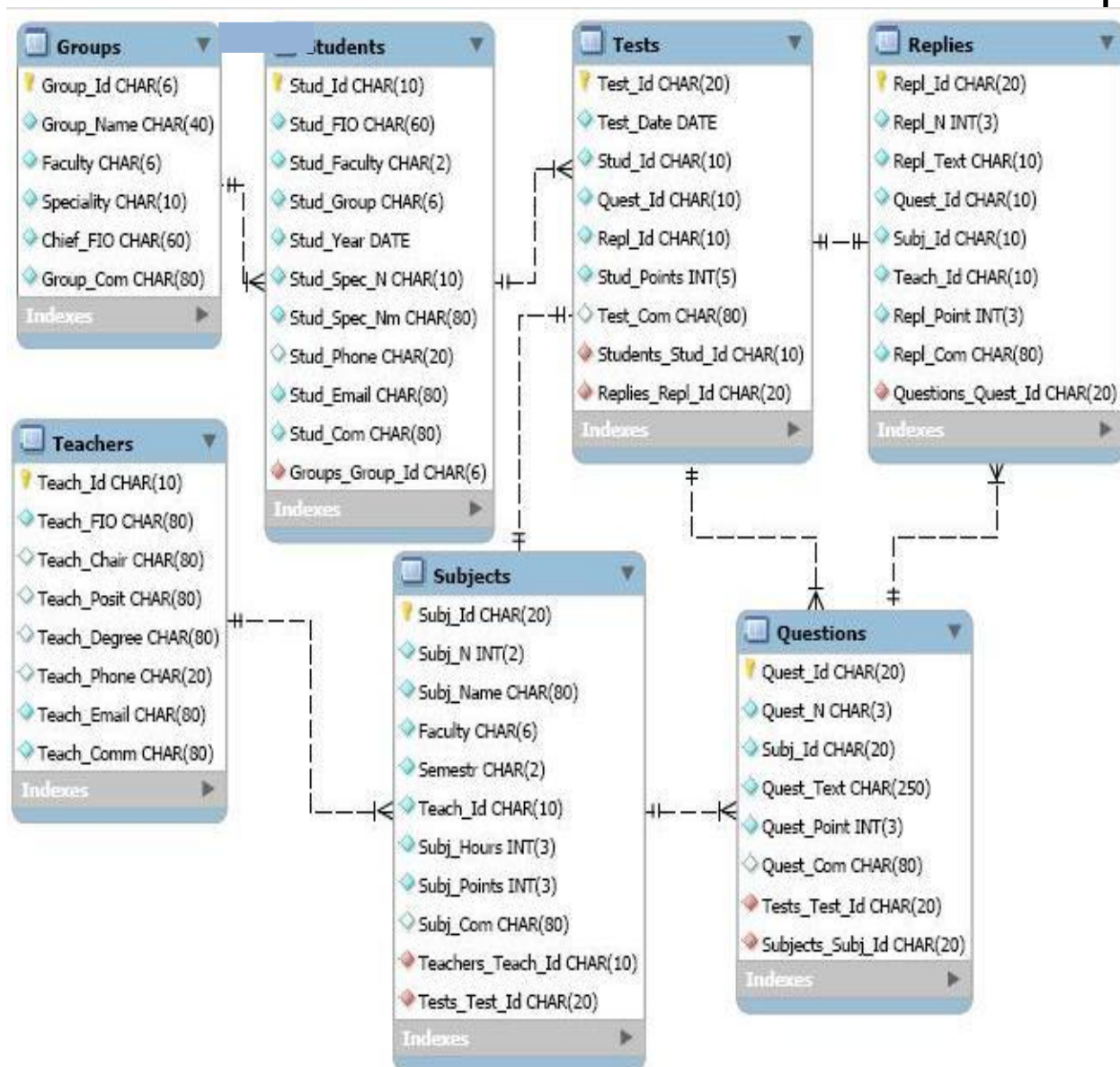


Рисунок 22 – Физическая схема БД

Доступ к таблицам базы данных администратор и преподаватель получают из своего кабинета. Кабинет преподавателя представлен на рисунке 23.

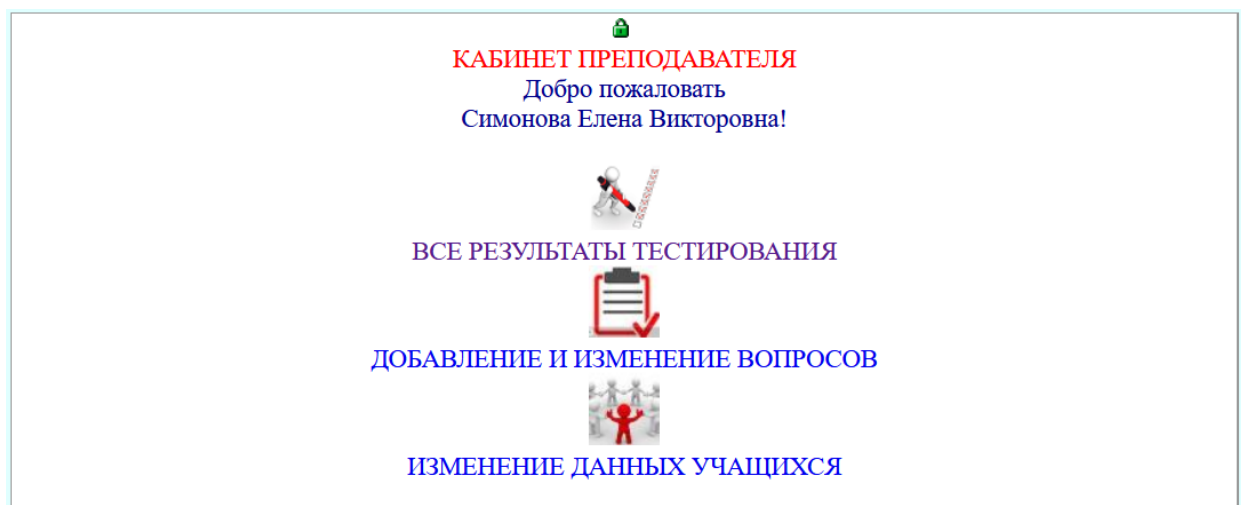


Рисунок 23 – Кабинет

3 Разработка системы для онлайн тестирования

3.1 Архитектура программного обеспечения

Система тестирования знаний, это веб-ориентированная система, веб-приложение с базой данных и веб-интерфейсом с «клиент-серверной архитектурой».

Клиентский процесс организуется интерфейсом пользователя посредством браузера. Браузер при взаимодействии с пользователем направляет сообщения (запросы) HTTP-сервер. Сервер получает и обрабатывает сообщения от браузера.

Динамические веб-страницы, обеспечивающие логику работы системы, разрабатывались на языке PHP с использованием SQL запросов к базе данных системы.

Совокупность всех программных средств, которые используются при создании и сопровождении системы, представлены на рисунке 24.

Используемые программные средства для создания и сопровождения системы представлены на рисунке 25.

Если программное обеспечение располагается на хостинге, то требования к хостингу следующие. Для функционирования системы необходимо следующее программное обеспечение:

веб-сервер – Apache версии не ниже 1.3.26;

СУБД – MySQL версии не ниже 3.23.

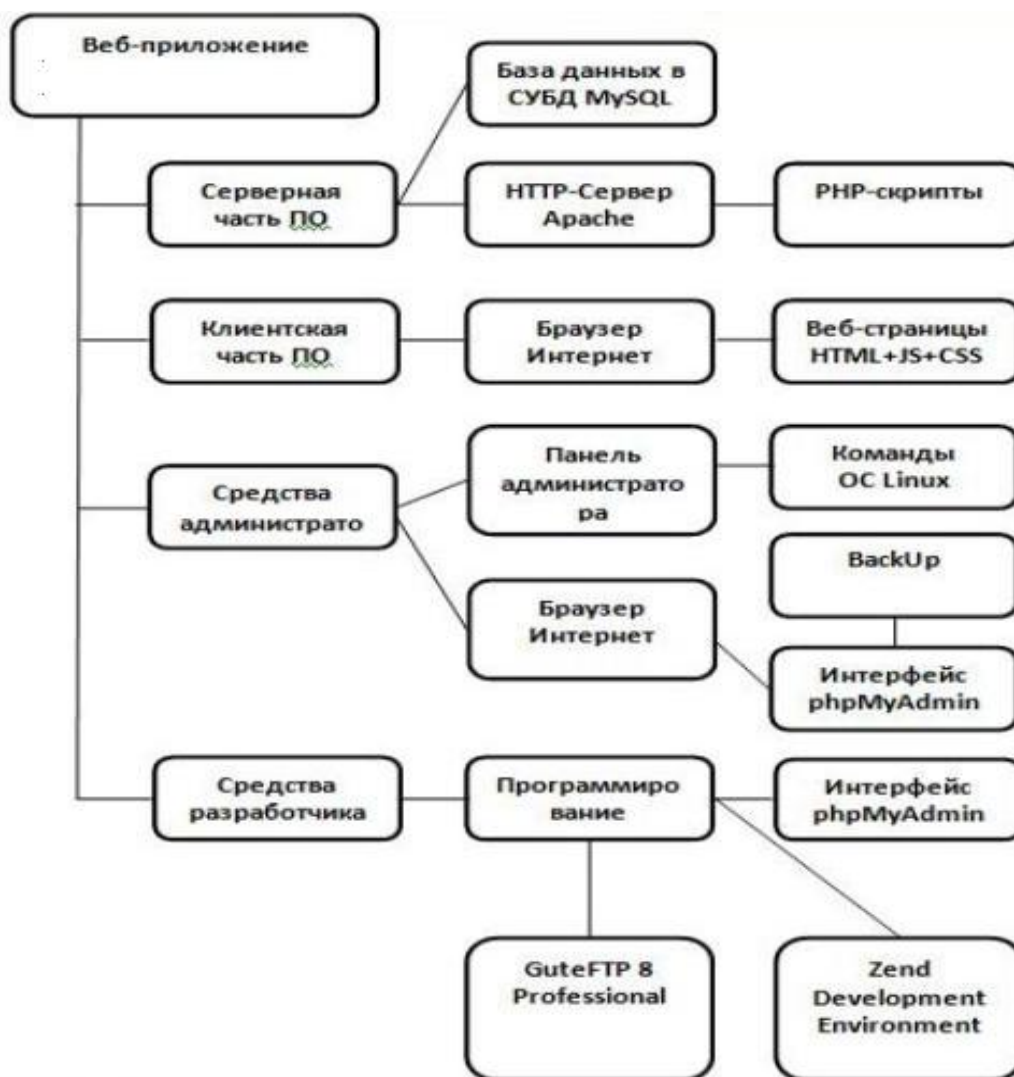


Рисунок 24 – Программные средства разработки и поддержки системы

Требования к клиентскому программному обеспечению

Веб-приложение должно быть доступно для полнофункционального просмотра с помощью любых браузеров:

- Google Chrome. Версия 49 и выше;
- Mozilla Firefox. Версия 49 и выше;
- Opera. Версия 40 и выше;
- Apple Safari. Версия 9 и выше;
- Яндекс.Браузер. Версия 16.9 и выше;
- Microsoft Internet Explorer. Версия 11 и выше;
- Microsoft Edge Browser. Версия 14.14 и выше;

Система должна быть работоспособна (информация, расположенная в системе, должна быть доступна) при отключении в браузере поддержки flash и JavaScript.

Требования к техническому обеспечению минимальные.

Управление программным обеспечением веб-приложений, расположенных на хостах общего пользования, производится с помощью FTP-клиентских приложений.

Для разработки модулей веб-приложения на компьютере разработчика использовалась программа Notepad++, которая позволяет не только редактировать разрабатываемый код, но и контролировать возможные ошибки разработчика программного обеспечения на языках html, php и других.

3.2 Технологии и языки разработки системы

В разработке не использовались CMS и фрейм-ворки, а только верстка html-css страниц и кодирование на языке PHP.

В кодировании на PHP при создании динамических страниц использовалось объектно ориентированное программирование (ООП). Были созданы классы ООП, которые использовались в разработке всех компонент программного обеспечения.

В основе работы класса табличного представления таблиц лежит понятие ассоциативного массива (модуль называется Any_Table_HTML_New.php).

В системе используется важная свойство сервера – создание защищенных областей с помощью управляющих файлов .htaccess, httpasswd. При обращении клиентской части приложения к защищенным с помощью .htaccess файлам папкам приложения, производится запрос на ввод логина и пароля пользователя.

Это свойство сервера использовалось для создания кабинетов зарегистрированного пользователя и администратора системы.

Для соединения с базой данных был использован класс DB_Class, реализующий основные запросы к базе данных.

3.3 Структура программ (скриптов)

На рисунке представлена структура взаимодействия программных модулей системы. Каждому уровню может соответствовать ряд программных модулей, представленных в крайней правой колонке схемы. Однако не все модули второго уровня делают запросы к базе данных и не все модули отображены в представленной схеме.

Рассмотрим более подробно All_Tables.php, который предназначен для отображения всех таблиц базы данных на экране пользователя.

Этот php-скрипт вызывает посредством команды `php_include` скрипт нижнего уровня DB_Tables_Fields.php. По сути, этот скрипт включается в тело All_Tables.php. DB_Tables_Fields.php сканирует текущую логику базы данных «на лету».

На рисунке 25 изображена структурная схема программного обеспечения. Для соединения с базой данных был использован класс DB_Class, реализующий основные запросы к базе данных.



Рисунок 25 – UML – диаграмма классов структурная схема программного обеспечения

3.4 База данных для системы онлайн тестирования

Физическая структура базы данных реализована с помощью пакета phpMyAdmin. Вход через браузер в локальную версию происходит при наборе адреса <http://localhost/tools/phpMyAdmin>. Система берет параметры соединения с базой данных из файла (файл DB_data.php).

В качестве примеров структуры таблиц базы данных приведем таблицу базы данных со всеми вопросами по тестам и таблицу базы данных с ответами по всем вопросам и всем тестам (рисунки 26-27).

MyDB\tests_psych\ - HeidiSQL 9.1.0

Файл Редактировать Поиск Инструменты Помощь

Фильтр базы Фильтр табл Хост: diofant.com База данных: tests_psych Запрос

MyDB	447,7 KiB
tests_psych	447,7 KiB
1_subjects	3,0 KiB
2_tests	3,5 KiB
3_quest	28,5 KiB
4_replies	147,6 KiB
5_staff_replies	231,7 KiB
6_staff	18,1 KiB
7_tests_results	5,6 KiB
8_conclusions	9,9 KiB

Имя	Строк	Размер	Создано	Обновлено	Движок	Комментарий
1_subjects	2	3,0 KiB	2020-04-...	2020-04-2...	MyISAM	Subject for testing...
2_tests	2	3,5 KiB	2020-05-...	2020-05-0...	MyISAM	Tests names
3_quest	26	28,5 KiB	2020-04-...	2020-05-0...	MyISAM	Subject questions,...
4_replies	96	147,6 K...	2020-04-...	2020-04-2...	MyISAM	Possible replies. P...
5_staff_replies	89	231,7 K...	2020-05-...	2020-05-0...	MyISAM	Staff replies for all ...
6_staff	7	18,1 KiB	2020-04-...	2020-05-0...	MyISAM	Registered Staff list
7_tests_results	2	5,6 KiB	2020-05-...	2020-05-0...	MyISAM	Tests Results Total
8_conclusions	6	9,9 KiB	2020-05-...	2020-05-0...	MyISAM	Tests names

Рисунок 26 – База данных системы

phpMyAdmin

127.0.0.1 » testsdill » questions "Subject questions, 999 questions for every Subject"

Обзор Структура SQL Поиск Вставить Экспорт Импорт Операции Слежение Триггеры

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	Quest_Id	char(4)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
2	Quest_N	int(5)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
3	Subj_Id	char(3)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
4	Subj_Nm	char(80)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
5	Test_Id	char(3)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
6	Quest_Text	char(250)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
7	Repl_1	char(250)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
8	Repl_2	char(250)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
9	Repl_3	char(250)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
10	Repl_4	char(250)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
11	Repl_N	char(1)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё
12	Repl_Point	int(1)	utf8_general_ci	Нет	Нет	Нет		Изменить Удалить Обзор уникальных значений Ещё

↑ Отметить все / Снять выделение С отмеченными: Обзор Изменить Удалить Первичный Уникальный Индекс

Пространственный Полнотекстовый

Версия для печати Связи Анализ структуры таблицы Отслеживать таблицу

Добавить 1 поле(я) В конец таблицы В начало таблицы После Quest_Id OK

Рисунок 27 – Структура таблицы базы Вопросов

3.5 Основные алгоритмы работы системы

Алгоритмы функционирования главных составляющих модулей системы можно представить в виде блок-схем. Блок-схема модуля (рисунок 28), регистрирует в системе новых пользователей-тестируемых. В этом случае, модуль выдает форму регистрации с полями логина и пароля пользователя. Пользователем вводятся выбранные им логин и пароль, а также вводится ряд данных самого пользователя.

Затем, если все данные соответствуют заложенным форматам для ввода данных, модулем проверяется наличие занесенного логина среди тех, которыми уже заполнена таблица базы данных пользователей. При этом открывается база данных и таблица данных пользователей. Данные связи с базой данных для открытия представлены в модуле DB_data.php.

Модуль производит поиск занесенного логина и если такой уже есть, то модуль выдает соответствующее сообщение на экран, что такой логин уже есть в базе данных. В противном случае пароль пользователя шифруется, логин, пароль и сопутствующие персональные данные заносятся в соответствующую таблицу пользователей базы данных системы.

Рассмотрим последовательность и составляющие алгоритма в процессе тестирования. Когда пользователем нажимается вкладка «Вход», система выдает форму, предназначенную для авторизации пользователей. Пользователем вводятся логин и пароль, зафиксированные системой ранее. Если системой находятся соответствующие значения логина и пароля, то система допускает пользователя к процессу тестирования.

На первом этапе пользователем выбирается предмет для тестирования из выпадающего списка предметов. На следующем этапе выдается форма для выбора собственно теста - имя теста. Системой используются во многом те же модули соединения и доступа к базе данных, определяются таблицы базы данных, которые будут применяться для работы алгоритмов тестирования. При этом система выбирает последовательно в базе последовательные вопросы и ответы. Сначала выбирается первый вопрос, и ряд возможных ответов на первый вопрос. Создается динамическая форма, представляющая эти ответы - форма переключателей для выбора одного варианта из всех возможных ответов. Когда пользователь выбирает ответ и нажимает кнопку «Следующий вопрос», системой заносится ответа в таблицу ответов базы данных. Система определяет количество баллов за данный ответ, если ответ верный. Баллы также заносятся в таблицу базы данных системы. Система определяет

следующий вопрос из таблицы базы данных вопросов и варианты возможных ответов к ним. Такие последовательные операции загрузки вопросов и ответов, генерации форм выдачи на экран происходят до тех пор, пока не закончатся все вопросы по тесту.

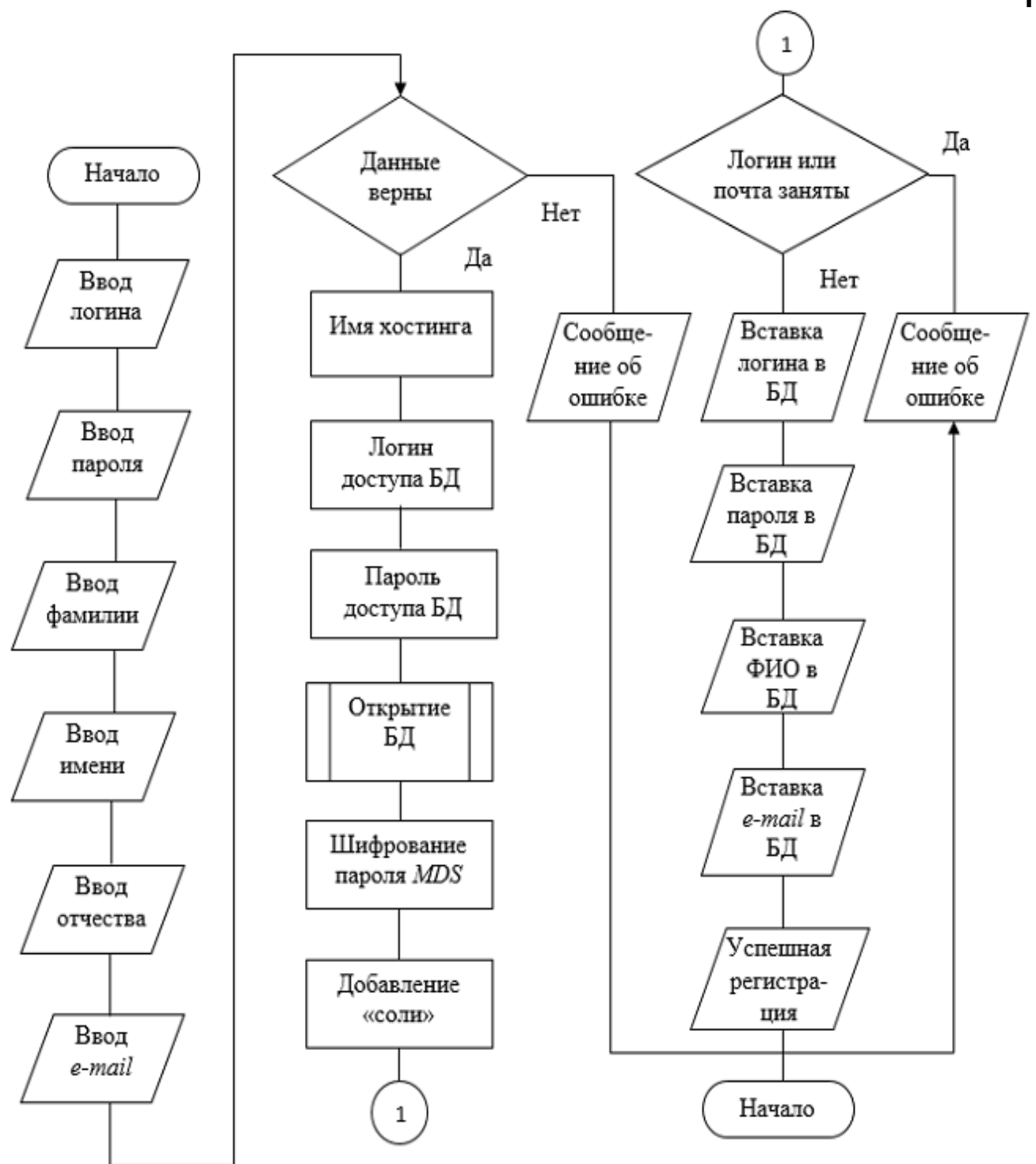


Рисунок 28 – Блок-схема алгоритма регистрации пользователей

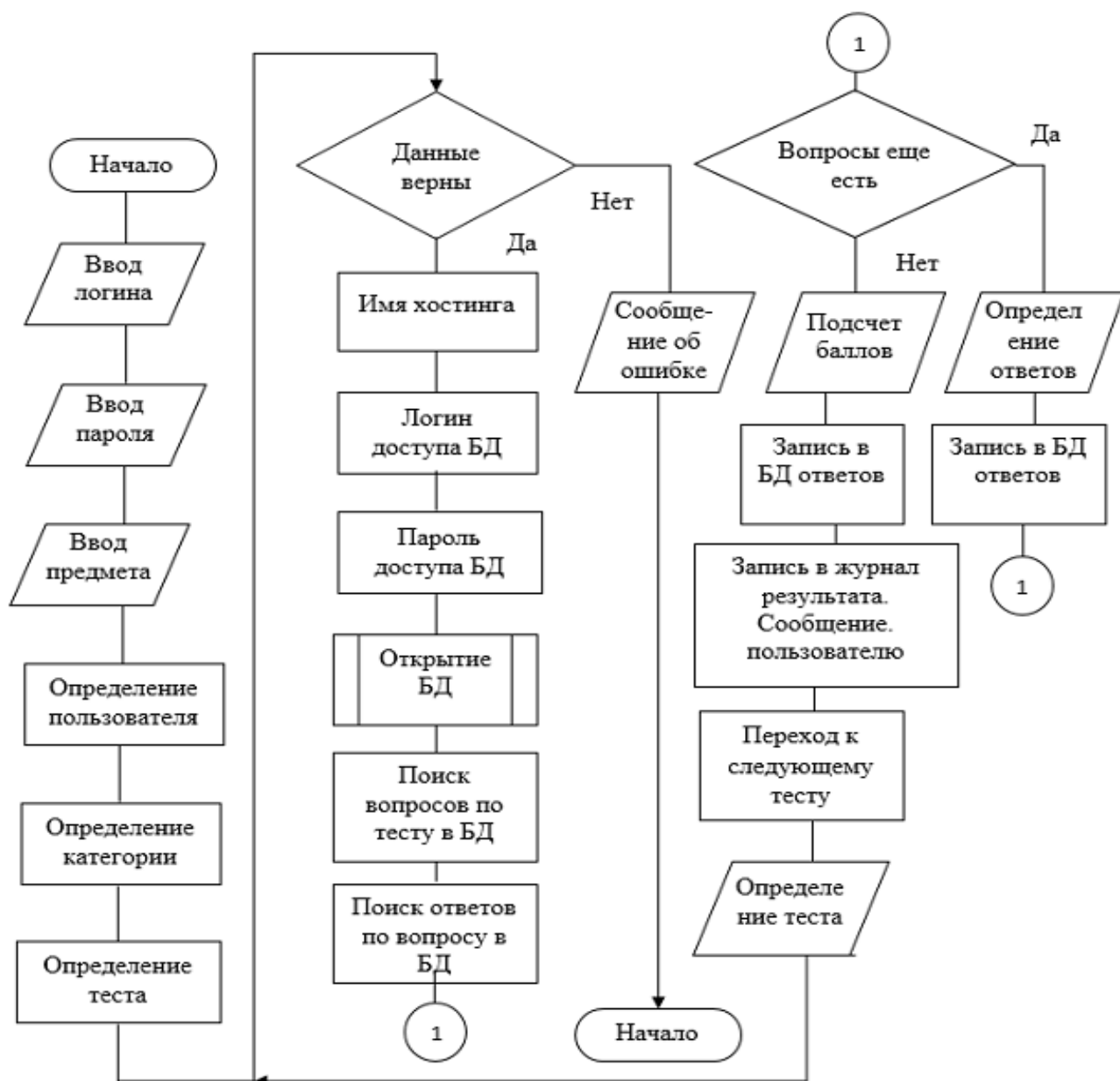


Рисунок 29 – Блок-схема алгоритма тестирования

Когда перечень вопросов по тесту закончивается, системой вычисляется сумма зачисленных баллов, вырабатывает текст заключения, который определяется набранным количеством баллов и этот текст вместе с количеством баллов выдается на экран тестируемого. Система предлагает перейти к следующему тесту.

Данные тестируемого, наименование тестов, вопросы и ответы, баллы и заключения сохраняются в журнале результатов тестирования. Результаты могут просматриваться тестируемым в своем личном кабинете.

4. Администрирование системы тестирования

4.1 Администрирование базы данных

Администрирование базы данных производится с помощью программы phpMyAdmin, которая расположена, как на локальном Денвере, так и на хостинге, куда загружается веб-приложение. Вид панели управления представлен на рисунке 30.

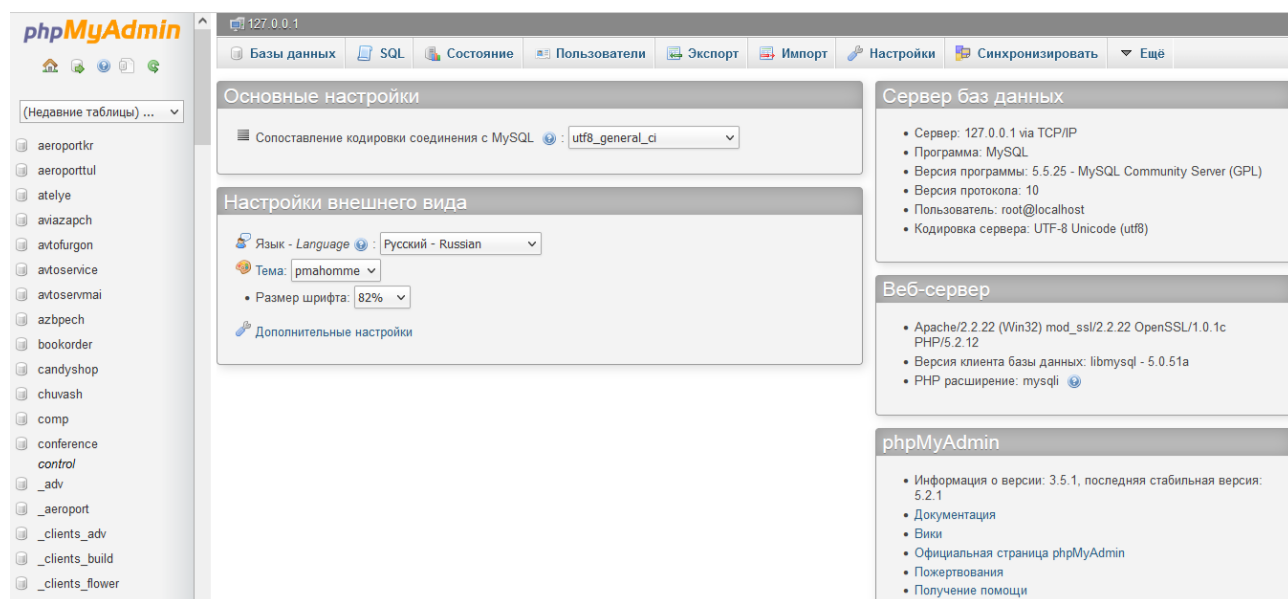


Рисунок 30 – Панель управления phpMyAdmin

Для администрирования базы данных выбирается база данных в левом фрейме панели управления. В результате этого появляется база данных с таблицами этой базы (рисунок 31)

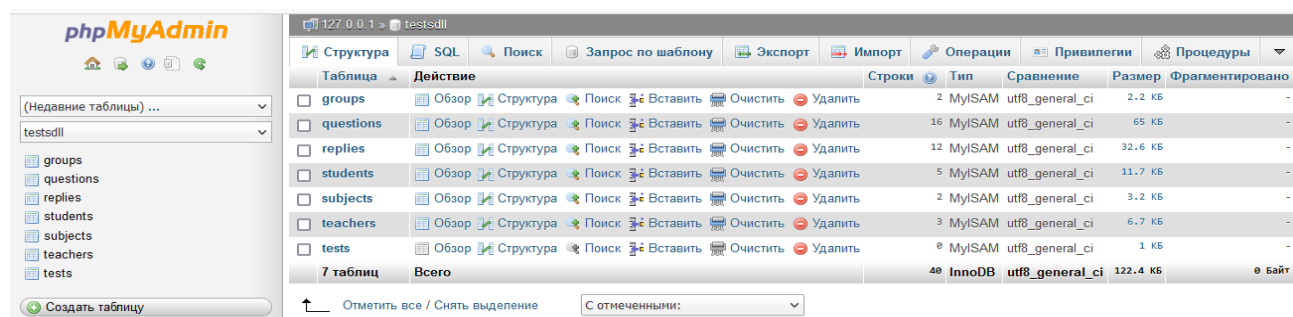


Рисунок 31 – Таблицы базы данных

Для управления содержимым таблиц базы данных, ставится флажок напротив соответствующей таблицы и определяется действие. Для изменения содержимого таблицы, необходимо кликнуть на наименование таблицы. Появится таблица со структурой, которое можно изменять с помощью панели управления (рисунок 32).

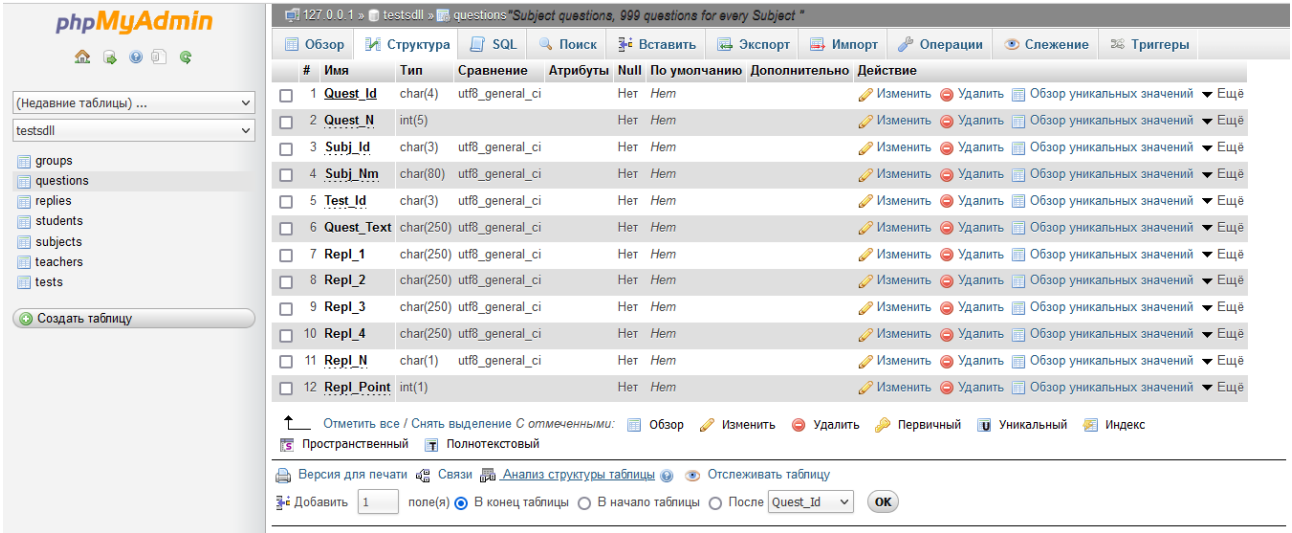


Рисунок 32 – Изменения содержимого таблицы «Вопросы»

А для того, чтобы изменять содержимое таблиц, необходимо кликнуть на вкладку панели управления «Обзор» (рисунок 33).

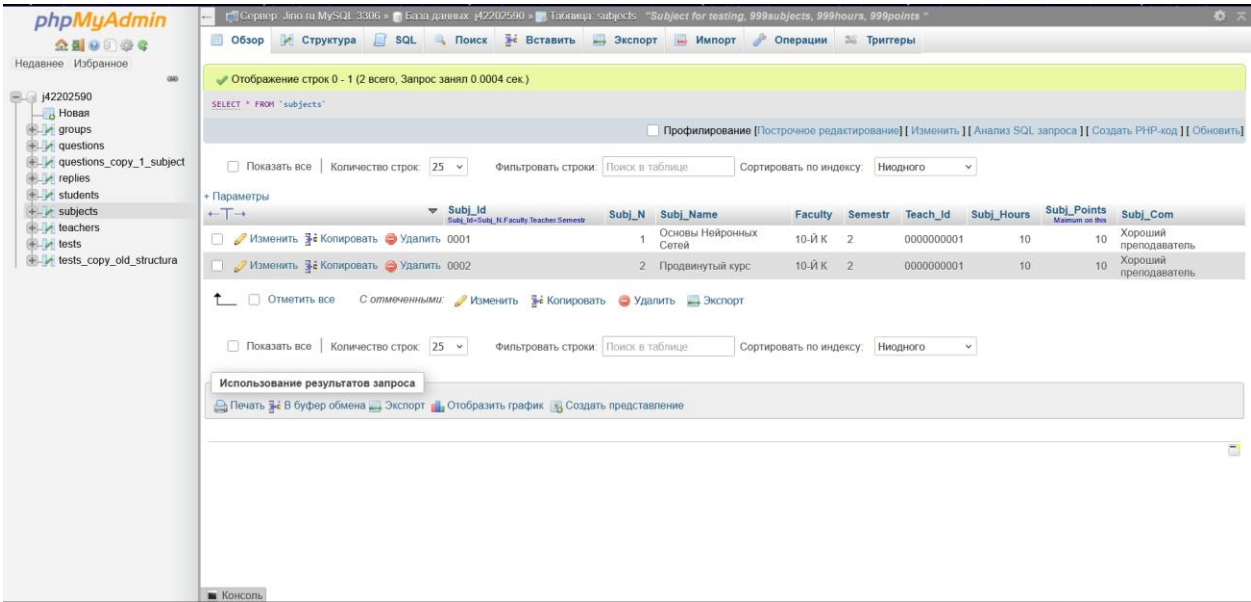


Рисунок 33 – Таблица вопросов для изменения, добавления данных

Администратор системы обязан делать периодически дампы базы данных и сохранять их в виде SQL кода для быстрого восстановления в случае сбоев системы и нарушения целостности базы данных. Для этого на панели управления имеется вкладка «Экспорт». Необходимо выбрать в правом фрейме базу данных и нажать на эту вкладку. Появится вкладка панели, с помощью которой создается дамп базы данных (рисунок 34).

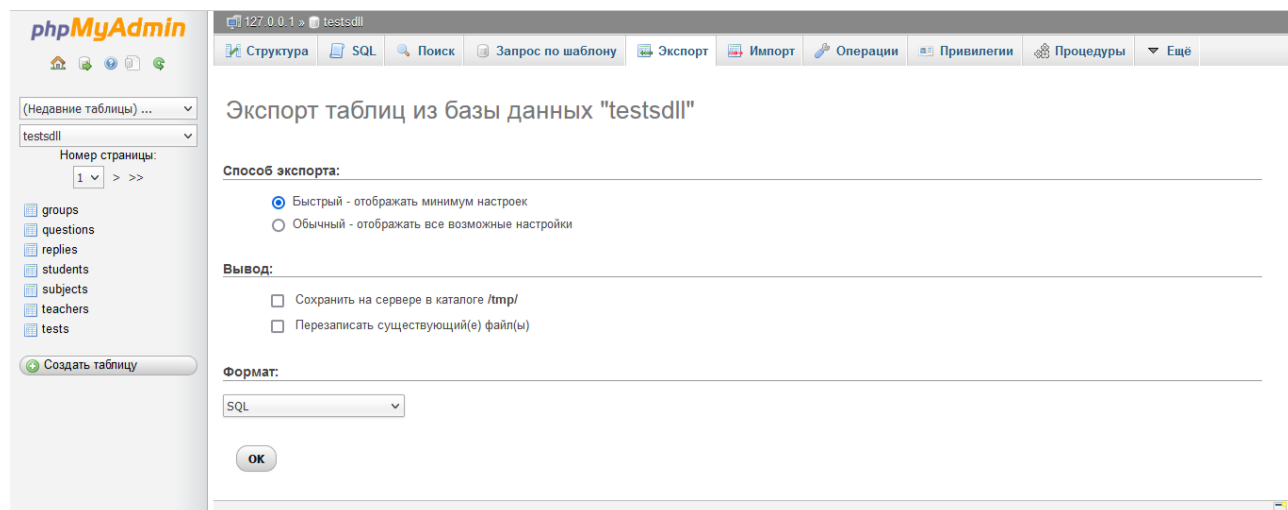


Рисунок 34 – Создание дампа базы данных

Все дампы необходимо хранить в отдельной папке приложения с именем SQL.

4.2 Администрирование сервера

Одним из важных критериев выбора хостинга является используемая операционная система, поскольку от этого зависит программное обеспечение, которое будет поддерживать функциональность тех или иных сервисов. Важным аспектом описания хостинга является наличие следующих возможностей:

- поддержка CGI: PHP;
- поддержка. htaccess/htpasswd (Apache);
- поддержка баз данных;
- установленные модули и фреймворки для каждой из возможностей.

Для размещения в сети Интернет системы критическими являются следующие возможности хостинга:

- поддержка PHP и MySQL;
- неограниченный входящий и исходящий трафик;
- отсутствие ограничений по посещаемости;
- отсутствие рекламы;
- дисковое пространство не менее 200 мегабайт;
- приемлемая стоимость.

Вход в систему администратора осуществляется по логину Sim и паролю S. В случае неудачной авторизации поля для ввода логина и пароля очищаются. Авторизация не происходит.

Для управления сервером и настройками имеется панель управления, в которой можно выбрать необходимые действия (рисунок 35).

					ПГУ1.090301.45.001 ПЗ	Лист
						46
Изм.	Лист	№ докум.	Подпись	Дата		

Упра

SQL

Базы данны

Для локальн

Справка о р

Все настрой

Файловый менеджер

FTP-аккаунты

Управление MySQL

Консоль

Задания по расписанию

Резервные копии

Настройки SSH

Настройки веб-сервера

Антивирус

Защита от сетевых атак

Техническая информация

Создать б

Имя БД

j42202590

зуйте имя сервера «localhost».

Пользователи

j42202590

Рисунок 35 – Панель управления и настройки сервера

Например, в настройках сервера можно выбрать версию PHP (рисунок 36).

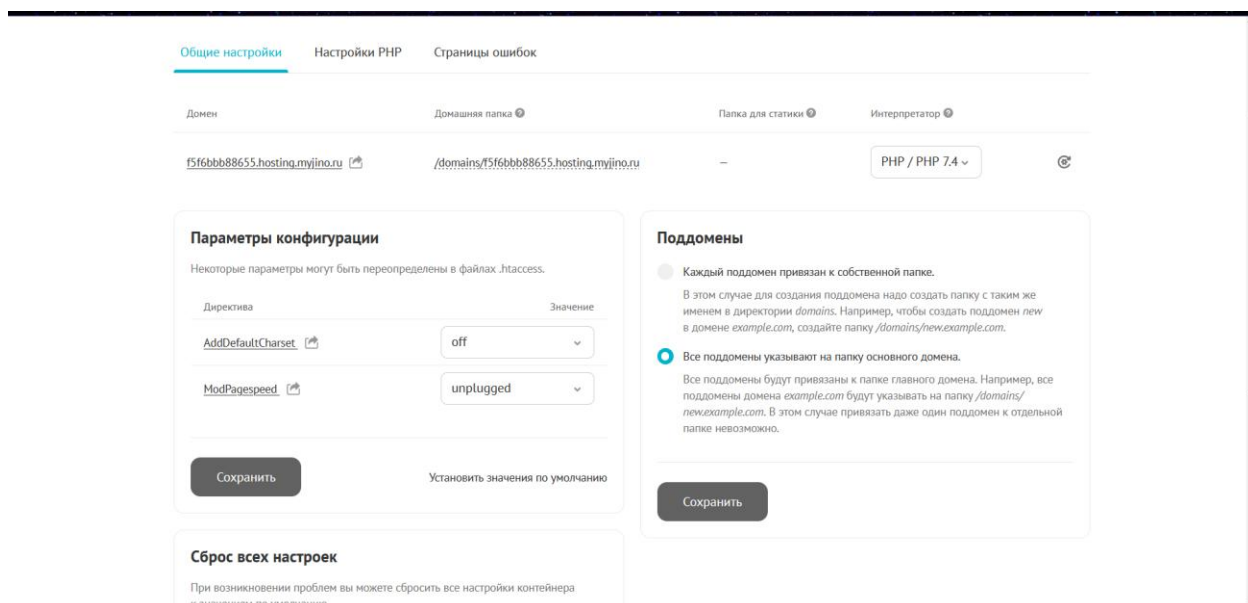


Рисунок 36 – Опции настроек сервера на панели управления

Возможны некоторые отличия в версиях PHP и директивах `php.ini` файле, которые можно адаптировать под приложение.

4.3 Администрирование системы

Рассмотрим последовательность действий при установке системы на хостинг без приобретения доменного имени для системы. Конкретное наименование хостинга не имеет значения. Важно, чтобы была поддержка стека программного обеспечения, которое фигурирует в проекте – LAMP (Linux, Apache, MySQL, PHP), что входит в пакет обслуживания любого хостинг провайдера. В данном случае используется хостинг `jino.ru`.

В этом случае для владельца хостинга создается корневой каталог, где можно хранить и использовать все приложения. Последовательность действий при установке системы на хостинг следующая.

Все программное обеспечение на локальном компьютере находится в папке `TestsStud`. Запаковываем все программное обеспечение. Получаем файл с именем `TestsStud.zip`.

Входим в аккаунт на хостинге, переходим к вкладке панели управления аккаунтом менеджер файлов (рисунок 37).

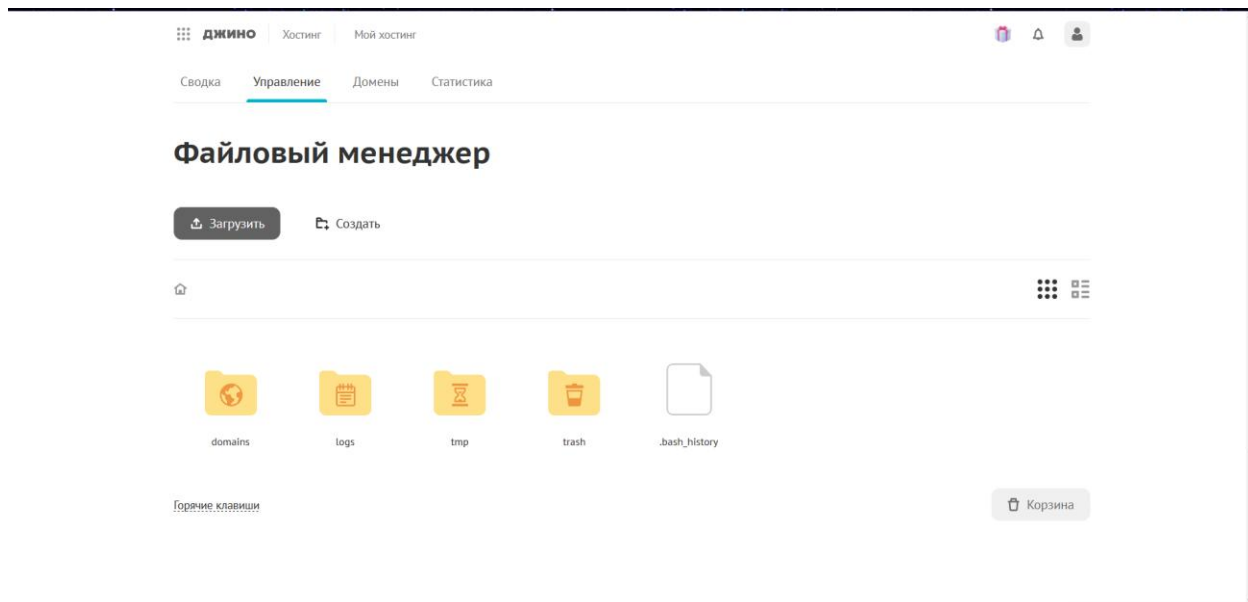


Рисунок 37 – Файловый менеджер

Затем открываем папку «domains», в котором находится корневой каталог для аккаунта со всеми приложениями, загружаем файл, используя кнопку «загрузить» и распаковываем. Структура папок представлена на рисунке (рисунок 38)

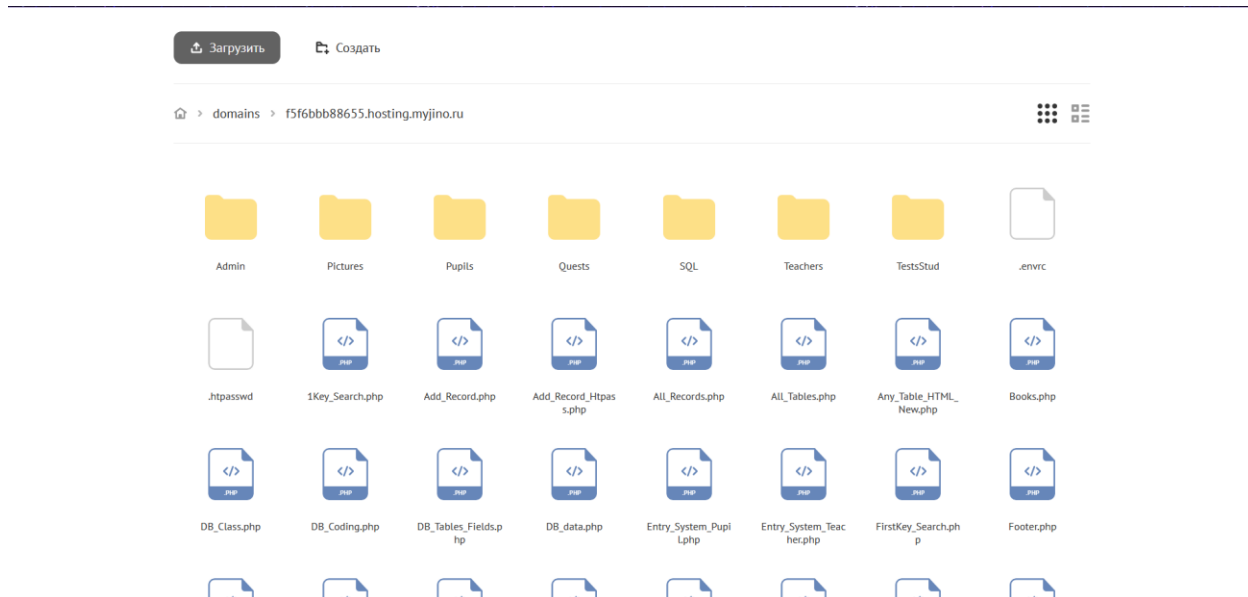


Рисунок 38 – Пример структуры папок и файлов на сервере

Теперь создадим базу данных с именем testsdll с помощью phpMyAdmin, перейдя в панели управления аккаунтом к вкладке «Управление MySQL». Так

как для данной системы не создано доменного имени, то база создается с префиксом и имеет название j42202590. Затем импортируем базу данных из папки TestsStud/SQL/testsdll.sql, находящуюся на локальном компьютере, в созданную на хостинге базу данных.

Перейдя во вкладку панели управления «Файловый менеджер», открыв папку TestsStud, редактируем данные соединения с базой данных в DB_data.php следующим образом (рисунок 39).

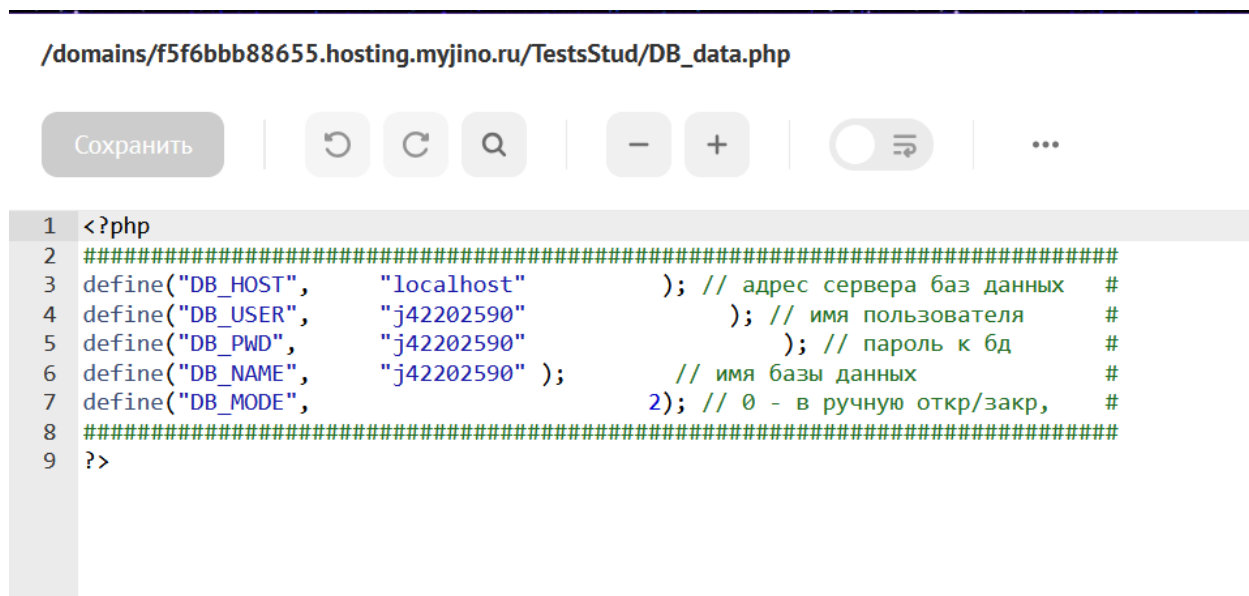


Рисунок 39 – Параметры соединения с базой данных

Теперь все данные для работы приложения готовы.

Еще один нюанс связан с авторизацией пользователей. В данном программном обеспечении используются свойства файла Apache .htaccess-httpasswd. И для разных групп пользователей свойства разные.

```

Allow from all

AuthName "Username and password required"
AuthUserFile
"/home/users/j/j55600682/domains/j55600682.myjino.ru/TestsStud/.ht
passwd"

AuthType      Basic
require user   Sim AdminRem
Allow from all

```

```

AuthName "Username and password required"
AuthUserFile
"/home/users/j/j55600682/domains/j55600682.myjino.ru/TestsStud/.ht
passwd"
AuthType      Basic
require valid-user

```

Для начала работы с сайтом набираем в адресной строке браузера адрес, связанный с аккаунтом на хостинге:

«<http://f5f6bbb88655.hosting.myjino.ru/>»

4.4 Тестирование системы

В первую очередь пользователь (тестируемый, студент) заходит в систему. Главная страница представлена на рисунке 40. Здесь он может познакомиться с концепцией приложения.



Рисунок 40 – Главная страница приложения

Здесь необходимо нажать вкладку «СТУДЕНТ» и отсюда переход к странице регистрации в системе для прохождения тестирования. Форма регистрации описана в предыдущем разделе. Затем через эту же вкладку он может войти в систему. При этом появляется форма авторизации пользователя (рисунок 41).

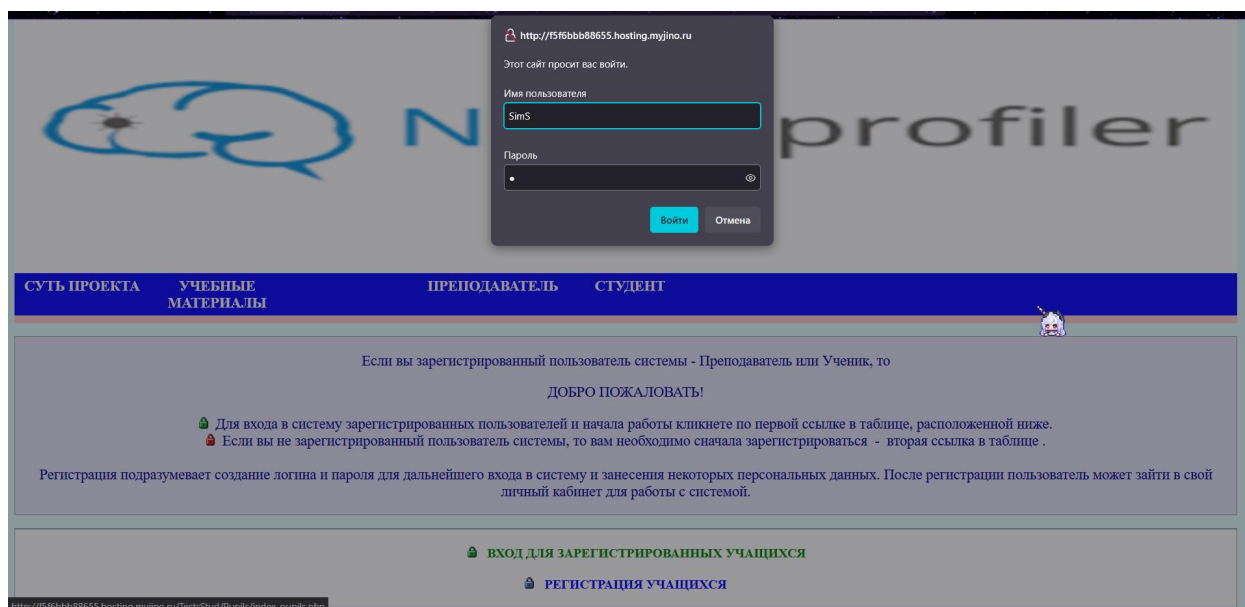


Рисунок 41 – Форма авторизации

Если данные логина и пароля правильные, то пользователь входит в личный кабинет пользователя (рисунок 42).



Рисунок 42 – Личный кабинет тестируемого пользователя

Далее пользователь нажимает на ссылку ПЕРЕХОД К ТЕСТИРОВАНИЮ. Появляется форма выбора предмета и теста (рисунок 43).



Рисунок 43 – Переход к форме выбора предмета и теста

После выбора пользователь переходит к тестированию, появляются вопросы с кнопочной формой (рисунок 44).

Перечень возможных ответов	Отметьте правильный ответ
а). Модель, имитирующая работу человеческого мозга для решения задач.	☒
б). Компьютерная программа, которая выполняет арифметические операции.	☐
в). Устройство для хранения данных	☐
не знаю	☐

Рисунок 44 – Вопросы с кнопочной формой

Система информирует о правильности ответа и заносит ответ в базу данных (рисунок 45).

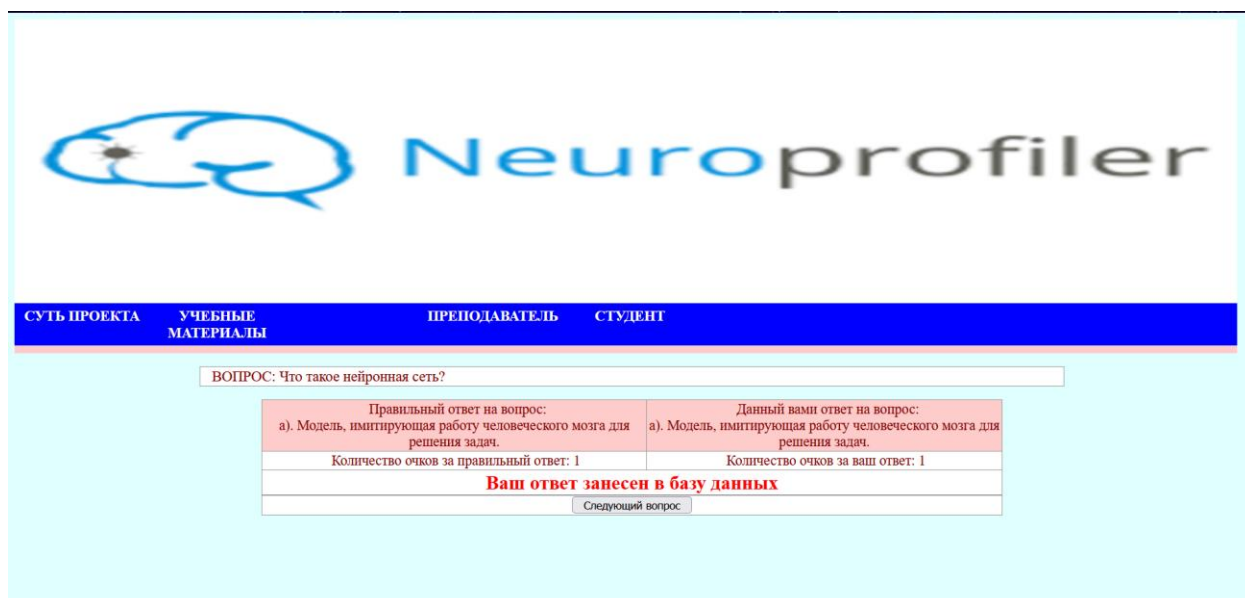


Рисунок 45 – Ответ занесен в базу данных

После прохождения всех вопросов система выдает оценку на основе той шкалы, которая заложена в систему (рисунок 46).

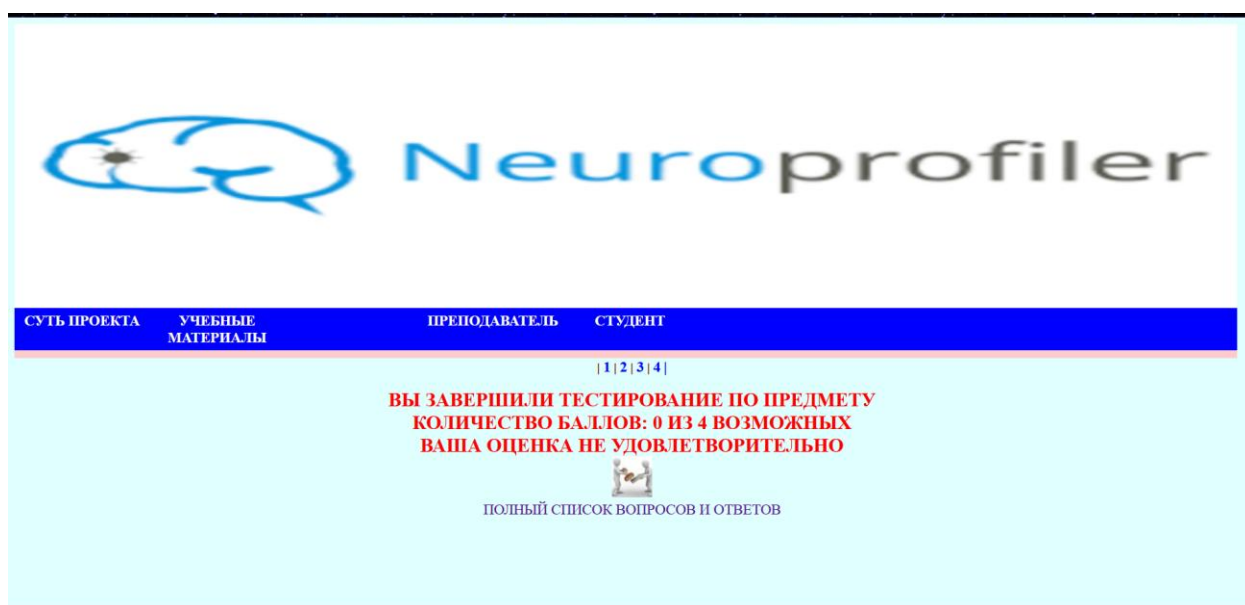


Рисунок 46 – Оценка ответов пользователя

Для начала работы пользователя преподавателя, необходимо нажать вкладку «ПРЕПОДАВАТЕЛЬ», затем «ВХОД ДЛЯ ЗАРЕГИСТРИРОВАННЫХ ПРЕПОДАВАТЕЛЕЙ». Появляется форма авторизации (рисунок 47).

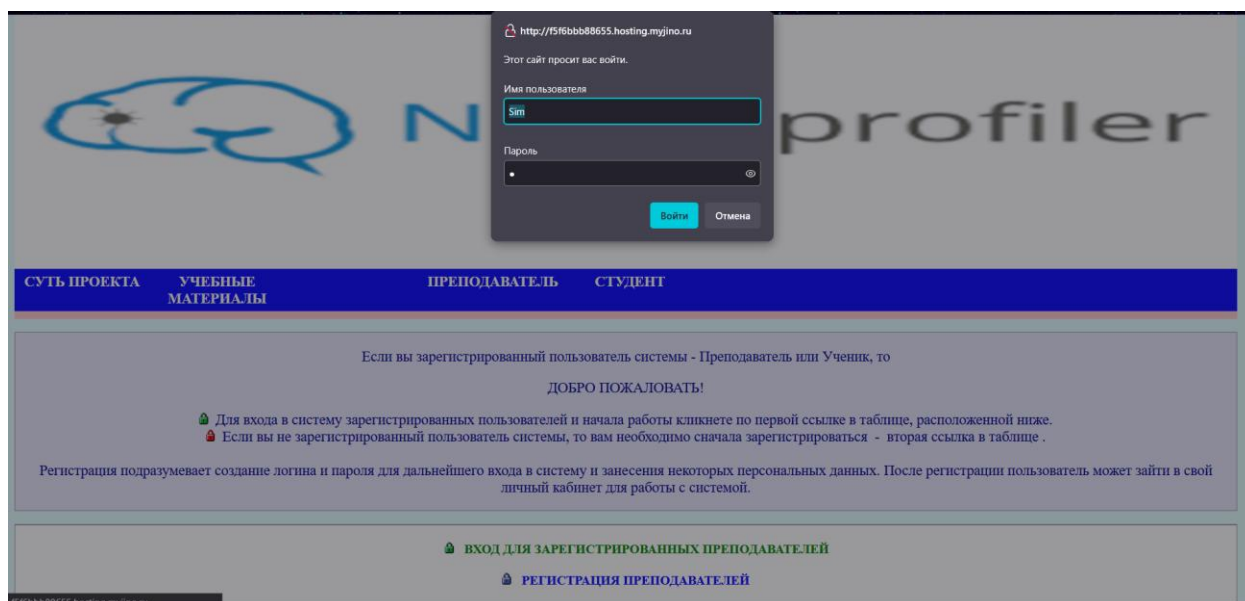


Рисунок 47 – Авторизация преподавателя

Логин и пароль верные, то преподаватель входит в свой личный кабинет. В данном случае в системе зарегистрирован преподаватель Teacher001 (рисунок 48)



Рисунок 48 – Кабинет преподавателя

Преподаватель может просматривать все данные по тестированию студентов (рисунок 49).



Neuroprofiler

СУТЬ ПРОЕКТА		УЧЕБНЫЕ МАТЕРИАЛЫ		ПРЕПОДАВАТЕЛЬ		СТУДЕНТ	
--------------	--	-------------------	--	---------------	--	---------	--

Студенты

Код ответа	№ предмета	Предмет	Тест	Код учащегося	№ вопроса	Наименование вопроса	Ответ тестируемого	Правильный ответ	Очки
211Sim	2	Продвинутый курс	1	Sim	1	Что такое переобучение (overfitting) в контексте нейронных сетей?	2) Когда модель слишком сложна и хорошо работает на обучающих данных, но плохо на тестовых.	2) Когда модель слишком сложна и хорошо работает на обучающих данных, но плохо на тестовых.	1
212Sim	2	Продвинутый курс	1	Sim	2	Что такое сверточные нейронные сети (CNN)?	4) Нейронная сеть, использующая только один слой.	2) Тип нейронной сети, специально разработанный для обработки изображений и видео.	0
111SimS	1	Основы Нейронных Сетей	1	SimS	1	Что такое нейронная сеть?	а). Модель, имитирующая работу человеческого мозга для решения задач.	а). Модель, имитирующая работу человеческого мозга для решения задач.	1
111Sim	1	Основы Нейронных Сетей	1	Sim	1	Что такое нейронная сеть?	а). Модель, имитирующая работу человеческого мозга для решения задач.	а). Модель, имитирующая работу человеческого мозга для решения задач.	1

Введите ID код для удаления, поиска или обновления:

[Просмотр структуры всех таблиц](#) [Добавить новую запись](#) [Свернуть все окна](#)

Рисунок 49 – Данные по тестированию студентов

Важной функцией преподавателя является занесение данных в базу данных по новым предметам и тестам.

Форма для занесения данных представлена на рисунке 50.



Neuroprofiler

СУТЬ ПРОЕКТА		УЧЕБНЫЕ МАТЕРИАЛЫ		ПРЕПОДАВАТЕЛЬ		СТУДЕНТ	
--------------	--	-------------------	--	---------------	--	---------	--

Наименования полей	Заносимые или обновляемые данные
Код ответа	<input type="text" value="111Sim"/>
№ предмета	<input type="text" value="1"/>
Предмет	<input type="text" value="Основы Нейронных Сетей"/>
Тест	<input type="text" value="1"/>
Код учащегося	<input type="text" value="Sim"/>
№ вопроса	<input type="text" value="1"/>
Наименование вопроса	<input type="text" value="Что такое нейронная сеть?"/>
Ответ тестируемого	<input type="text" value="а). Модель, имитирующая работу человеческого мозга для решения задач."/>
Правильный ответ	<input type="text" value="а). Модель, имитирующая работу человеческого мозга для решения задач."/>
Очки	<input type="text" value="1"/>
<input type="button" value="Добавить"/>	<input type="button" value="Удалить"/>
<input type="button" value="Изменить"/>	<input type="button" value="Просмотр"/>

Рисунок 50 – Форма занесения данных по тестам

Преподаватель может изменить данные студента, кликнув на ссылку в личном кабинете «ИЗМЕНЕНИЕ ДАННЫХ УЧАЩИХСЯ». Появляется форма для обновления данных (рисунок 51)



Neuroprofiler

СУТЬ ПРОЕКТА	УЧЕБНЫЕ МАТЕРИАЛЫ	ПРЕПОДАВАТЕЛЬ	СТУДЕНТ
Наименования полей	Заносимые или обновляемые данные		
Код учащегося	<input type="text"/>		
Пароль	<input type="text"/>		
ФИО	<input type="text"/>		
Факультет	<input type="text"/>		
Группа	<input type="text"/>		
Год Поступления	<input type="text"/>		
Специализация	<input type="text"/>		
Наименование	<input type="text"/>		
Телефон	<input type="text"/>		
Электронный адрес	<input type="text"/>		
<input type="button" value="Добавить"/>	<input type="button" value="Удалить"/>		
<input type="button" value="Изменить"/>	<input type="button" value="Просмотр"/>		

Рисунок 51 – Форма для обновления данных студента

5. Безопасность жизнедеятельности

5.1 Правила безопасности при использовании ПК

В данном разделе необходимо учесть риски, перечисленные в ГОСТ 12.0.003-2015 "Система стандартов безопасности труда. Электронно-вычислительная техника. Общие требования безопасности". Ниже приведен перечень рисков, которым следует придать особое значение:

- риск повреждения зрения и утомления глаз:
 - 1) длительное использование ПК без перерывов и отдыха глаз;
 - 2) плохая эргономика рабочего места, включая неправильное расположение монитора, отражение света на экране и недостаточное освещение;
- риск травм и заболеваний опорно-двигательной системы:
 - 1) неправильное положение тела при работе с ПК, например, неправильная осанка, неправильная поза или неправильное расположение рук и ног;
 - 2) отсутствие эргономической мебели и принадлежностей, таких как стулья с подлокотниками и подставки для рук;
- риск воздействия электромагнитных полей:
 - 1) использование устаревшего или неисправного оборудования, которое может создавать сильные электромагнитные поля;
 - 2) неправильное расположение оборудования, например, слишком близко к рабочему месту или другим электронным устройствам;
- риск пожара и электрического удара:
 - 1) неправильное подключение и использование электрооборудования, включая перегрузку розеток и использование поврежденных кабелей;

2) нарушение правил пожарной безопасности, включая использование запрещенных электроприборов вблизи ПК и наличие горючих материалов рядом с оборудованием;

– риск нарушения информационной безопасности:

1) небезопасное использование паролей и учетных записей, включая слабые пароли или их совместное использование;

2) несанкционированный доступ к компьютерной информации, включая утечку конфиденциальных данных;

– риск воздействия шума:

1) использование шумных компьютерных компонентов или работа в шумной среде.

5.2 Правила безопасности при использовании смартфона

В этом разделе следует уделить внимание рискам, связанным с безопасностью при использовании данного устройства. А также разработать свод правил, чтобы гарантировать предотвратить следующие риски:

– риск зависимости и негативного воздействия на психологическое и физическое здоровье;

– риск отрицательного влияния на сон и здоровье:

– риск потери или кражи смартфона;

– риск воздействия радиочастотного излучения;

– риск отвлечения водителя.

С учетом указанных рисков, следует разработать и рекомендовать следующие правила безопасности при использовании смартфона:

– ограничьте время использования смартфона, особенно перед сном;

– соблюдайте эргономику и правильное положение тела при работе со смартфоном;

– используйте оригинальные зарядные устройства и кабели, избегайте перегрева и короткого замыкания;

- устанавливайте надежные пароли и используйте функции блокировки экрана для защиты данных;
- храните смартфон в безопасном месте и будьте бдительны в общественных местах;
- регулярно делайте резервные копии данных и используйте антивирусное программное обеспечение для защиты от вредоносных программ;
- ограничьте использование смартфона во время вождения и соблюдайте правила безопасности на дороге.

Правильное соблюдение этих правил поможет минимизировать риски и обеспечить безопасное использование смартфона.

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		60

Заключение

В рамках выпускной квалификационной работ была спроектирована и разработана инструментальная среда для проведения компьютерных испытаний, тестирований знаний. Разработанная система является веб-ориентированной системой, поддерживающей процесс создания и проведения тестов для проверки знаний через сеть интернет.

Программное обеспечение системы разработано с использованием современных веб-технологий, современных языков разработки веб-приложений с применением серверных систем управления базами данных. Разработанный прототип системы включает такие компоненты, как база данных системы, веб-интерфейсы групп пользователей – студентов и преподавателей.

Для достижения цели выпускной квалификационной работы были решены следующие задачи:

разработана концепция системы тестирования, методология проведения тестов и оценки результатов тестирования;

система тестирования ориентирована на текстовые вопросы, возможные ответы на которые предлагаются в виде формы выбора возможного ответа;

тестируемый выбирает правильный, с его точки зрения ответ, отмечая его в предлагаемой системой форме выбора;

выбрана клиент-серверная архитектура системы тестирования, ориентированная на использование интернет технологий;

проведен анализ программных средств, инструментов для разработки системы;

разработана база данных системы, которая реализована в среде СУБД MySQL;

разработана веб-ориентированная система онлайн тестирования, проведено тестирование системы на основе специально составленных тематических вопросов и ответов.

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		61

Система онлайн тестирования знаний реализована на основе клиент-серверной архитектуры. Клиентская часть реализуется посредством браузера пользователя. Логическая часть функционирования системы, реализуемой на стороне сервера, представлена взаимосвязанными PHP-скриптами, выполняющими все операции по обработке процесса тестирования пользователей, также выполняющими все функции административной части по управлению данными тестирования. Скрипты направляют SQL-запросы, взаимодействуя с базой данных системы, которая содержит все данные, связанные с тестированием – данные зарегистрированных тестируемых пользователей, перечни вопросов и ответов по тестам, правильные ответы пользователей, начисляемые баллы. В системе используется сервер базы данных СУБД MySQL.

Универсальность системы означает, что ее можно использовать для настройки на любые тесты по любым предметам. В настоящей версии заложены тесты по тестированию только некоторых направлений веб-технологий.

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		62

Список использованных источников

1. Новиков А. М. Образовательный проект [Электронный ресурс]: (методология образовательной деятельности): пособие для работников образования, участвующих в инновационной деятельности / А. М. Новиков, Д. А. Новиков. - Москва: Эгвес, 2004. - 119 с. ISBN 5-85009-551-9.
2. Симонов В. П. Педагогический менеджмент: ноу-хау в образовании: учебное пособие по курсу "Управление педагогическими системами": рек. УМО вузов РФ / В. П. Симонов. - Москва: Высшее образование, 2009. 357 с. табл. - (Основы наук) (Высшее образование). - Библиогр.: с. 352-357. - ISBN 978-5-9692-0481-2.
3. Информационные системы/О.Л. Голицына, Н.В. -М.: ММИЭИФП, 2004.- 329 с.
4. Грэхем И. Объектно-ориентированные методы. Принципы и практика. –М.: Вильямс, 2004.–880 с.
5. Акперов, И. Г. Управление проектами: учебно-методический комплекс / И. Г. Акперов, В. А. Долятовский. – Ростов-на-Дону: ИУБиП, 1999.
6. Дрогобыцкий, И.Н. Проектирование автоматизированных информационных систем. Учебник /И.Н. Дрогобыцкий/ – М.: Финансы и статистика, 1992.
7. Дейт, К. Дж. Введение в системы баз данных.: Пер. с англ. / К. Дж. Дейт. – М.: Издательский дом «Вильямс», 2002.
8. Конноли Т., Бегг К., Страчан А. База данных: проектирование, реализация и сопровождение. Теория и практика, 2-е изд.: Пер. с англ.: Уч. пос. – М.: Издательский дом «Вильямс», 2002. – 1120с.: ил. – Парал. тит. англ.
9. Вендров А.М. Современные технологии создания программного обеспечения / [Электронное издание]. (<http://www.jetinfo.ru>). Проверено 29.03.2006.

10. Зиндер Е.З. Проектирование баз данных: новые требования, новые подходы / СУБД. – 1996.
11. Зиндер, Е.З. Новое Системное Проектирование: Информационные Технологии и Бизнес-реинжиниринг. / Системы Управления Базами
12. Коберн А. Быстрая разработка программного обеспечения.: Пер. с англ. – М.: ЛОРИ, 2002.
13. Коберн А. Современные методы описания функциональных требований к системам.: Пер. с англ. – М.: ЛОРИ, 2002.
14. Водолазский В. Эффективная работа: РНР4. -СПб.: "Издательство Питер", 2002.
15. Конноли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. – М.: «Вильямс», 2001.
16. Шафер, Дональд, Ф., Управление программными проектами: достижение оптимального качества при минимуме затрат.: Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 1136 с.
17. Олифер В.Г., Олифер Н.А. Основы сетей передачи данных // [Электронное издание]. (<http://www.intuit.ru>). Проверено 11.02.2008.
18. Грофф, Дж. Р. Энциклопедия SQL. / Дж. Р. Грофф, П. Н. Вайнберг.: Пер. с англ. – СПб: «Питер», 2003. – 896 с.
19. Петров Ю.К. JAM - инструментальное средство разработки приложений в информационных системах архитектуры «клиент/сервер», построенных на базе РСУБД. «СУБД», 1995.
20. Тамре Л. Введение в тестирование программного обеспечения : Пер. с англ. – М.: Издательский дом «Вильямс», 2003.

Приложение А

(обязательное)

Обучающее Web-приложение для изучения нейронных сетей

UML диаграммы

					ПГУ1.090301.45.001 ПЗ	Лист
						65
Изм.	Лист	№ докум.	Подпись	Дата		

посетителем



Приложение А.2 – UML – диаграмма вариантов использования для тестируемых



				ПГУ 1.090301.17.005 Д19			
				Оперативное, системное администрирование работы различных информационных ресурсов			
				1М, категория «разработка»			

Приложение А.3 – UML – диаграмма вариантов использования для преподавателя

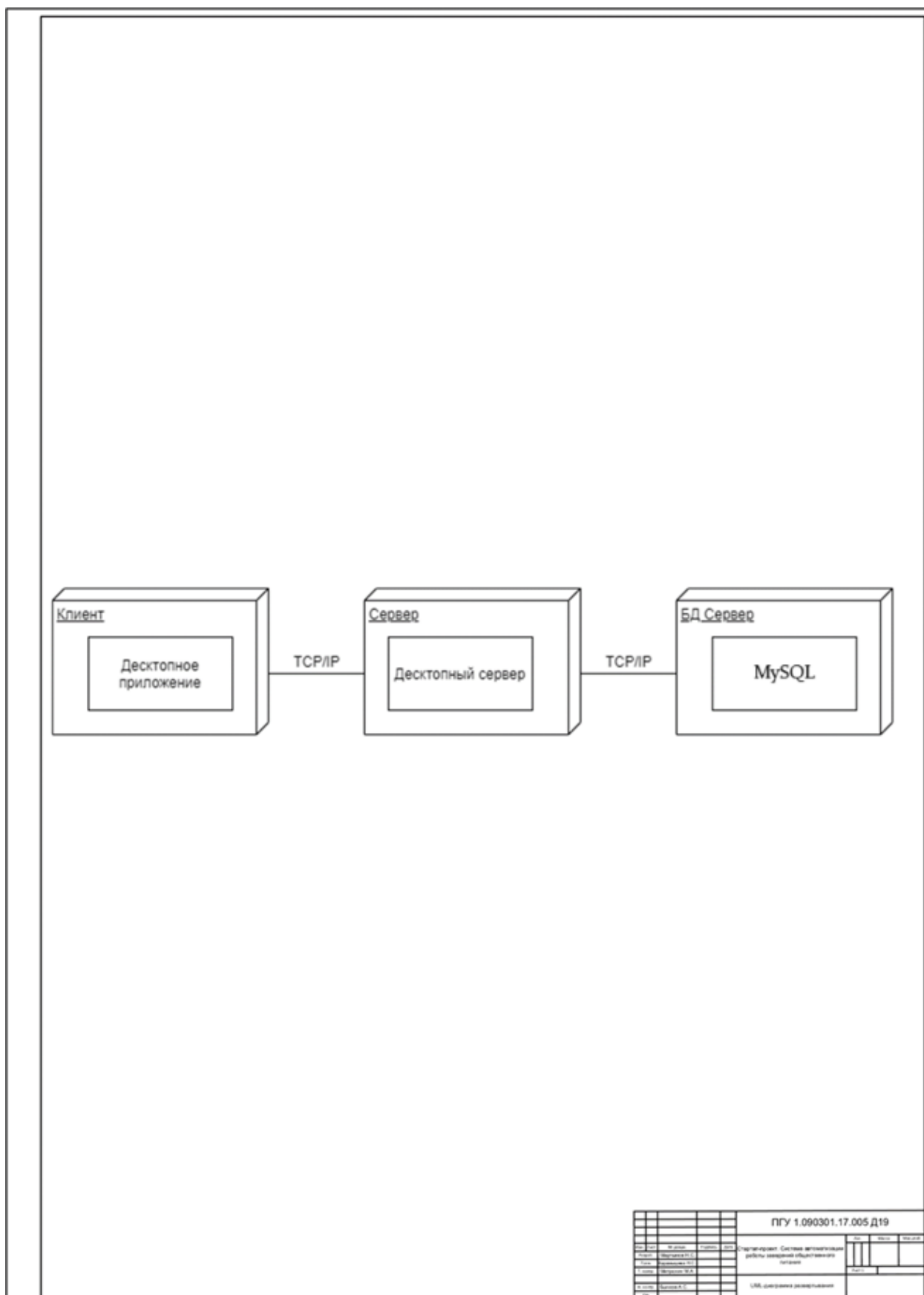
[illegible]

Приложение А.4– UML – диаграмма вариантов использования для администратора

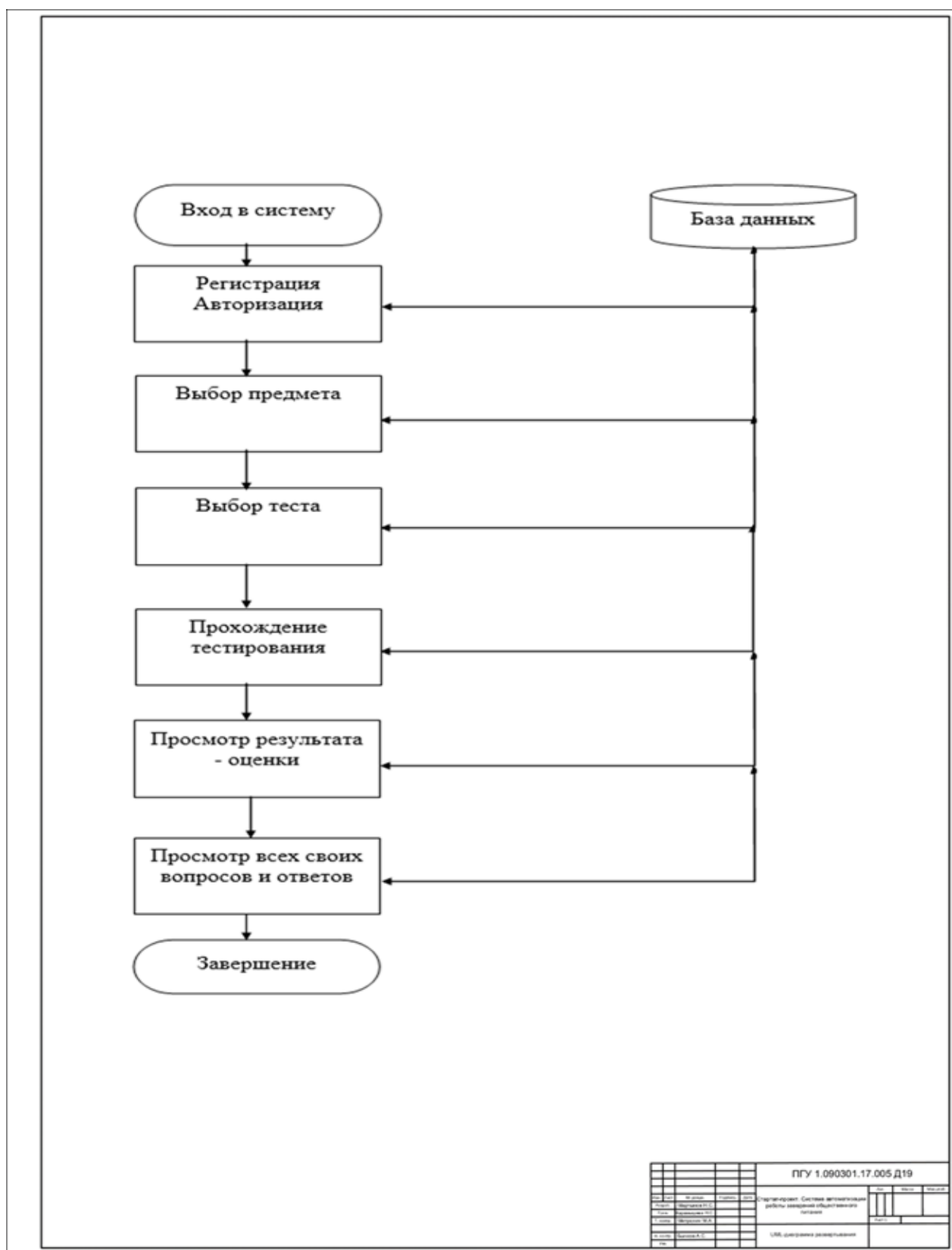


				ПГУ 1.090301.17.005 Д19			
№ п/п	№ докум.	Подпись	Дата	№ п/п	№ докум.	Подпись	Дата
1	Исходный №			2	Исходный №		
3	Исходный №			4	Исходный №		
5	Исходный №			6	Исходный №		
7	Исходный №			8	Исходный №		
9	Исходный №			10	Исходный №		
11	Исходный №			12	Исходный №		
13	Исходный №			14	Исходный №		
15	Исходный №			16	Исходный №		
17	Исходный №			18	Исходный №		
19	Исходный №			20	Исходный №		
21	Исходный №			22	Исходный №		
23	Исходный №			24	Исходный №		
25	Исходный №			26	Исходный №		
27	Исходный №			28	Исходный №		
29	Исходный №			30	Исходный №		
31	Исходный №			32	Исходный №		
33	Исходный №			34	Исходный №		
35	Исходный №			36	Исходный №		
37	Исходный №			38	Исходный №		
39	Исходный №			40	Исходный №		
41	Исходный №			42	Исходный №		
43	Исходный №			44	Исходный №		
45	Исходный №			46	Исходный №		
47	Исходный №			48	Исходный №		
49	Исходный №			50	Исходный №		
51	Исходный №			52	Исходный №		
53	Исходный №			54	Исходный №		
55	Исходный №			56	Исходный №		
57	Исходный №			58	Исходный №		
59	Исходный №			60	Исходный №		
61	Исходный №			62	Исходный №		
63	Исходный №			64	Исходный №		
65	Исходный №			66	Исходный №		
67	Исходный №			68	Исходный №		
69	Исходный №			70	Исходный №		
71	Исходный №			72	Исходный №		
73	Исходный №			74	Исходный №		
75	Исходный №			76	Исходный №		
77	Исходный №			78	Исходный №		
79	Исходный №			80	Исходный №		
81	Исходный №			82	Исходный №		
83	Исходный №			84	Исходный №		
85	Исходный №			86	Исходный №		
87	Исходный №			88	Исходный №		
89	Исходный №			90	Исходный №		
91	Исходный №			92	Исходный №		
93	Исходный №			94	Исходный №		
95	Исходный №			96	Исходный №		
97	Исходный №			98	Исходный №		
99	Исходный №			100	Исходный №		
101	Исходный №			102	Исходный №		
103	Исходный №			104	Исходный №		
105	Исходный №			106	Исходный №		
107	Исходный №			108	Исходный №		
109	Исходный №			110	Исходный №		
111	Исходный №			112	Исходный №		
113	Исходный №			114	Исходный №		
115	Исходный №			116	Исходный №		
117	Исходный №			118	Исходный №		
119	Исходный №			120	Исходный №		
121	Исходный №			122	Исходный №		

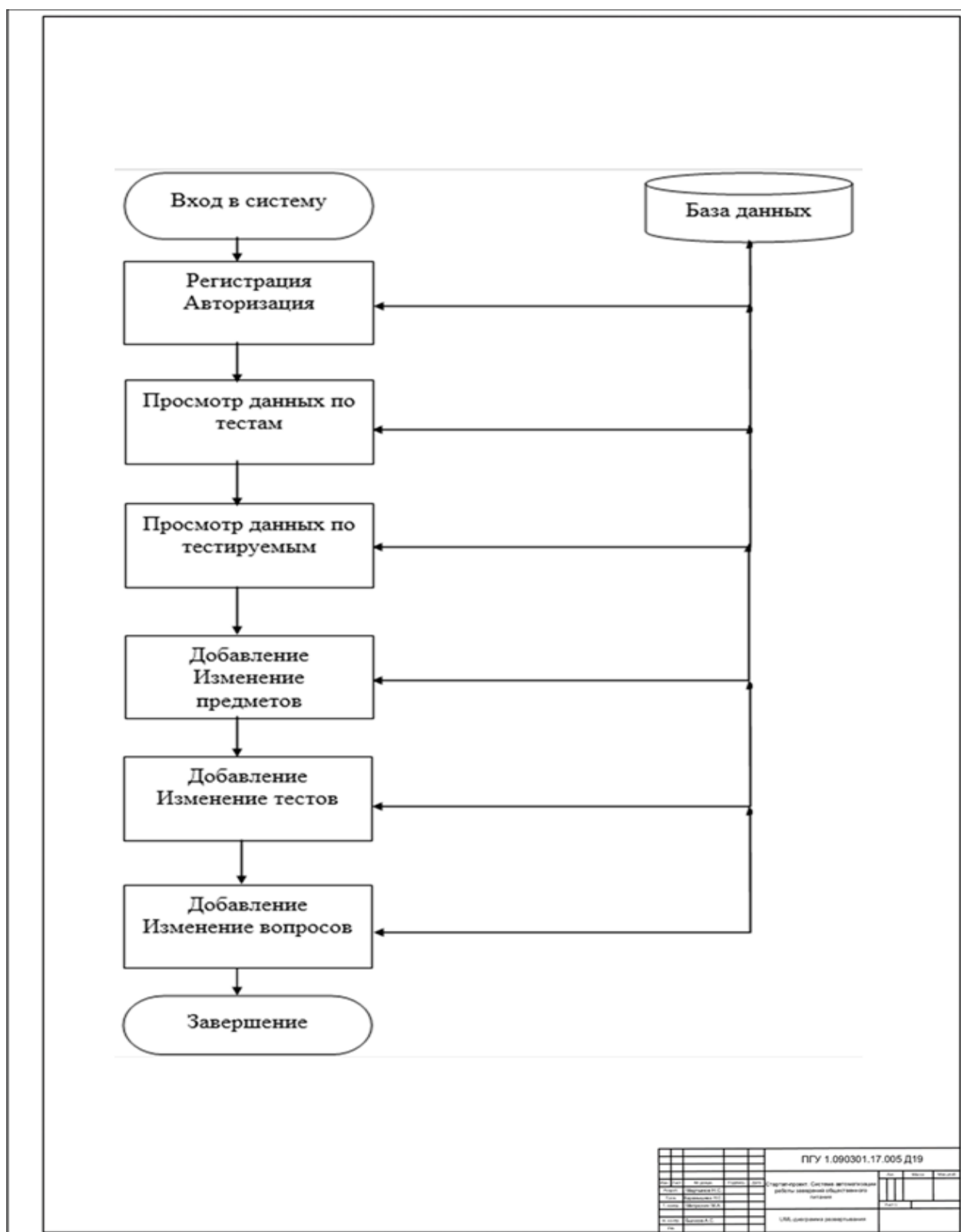
Приложение А.5 – UML – диаграмма развертывания



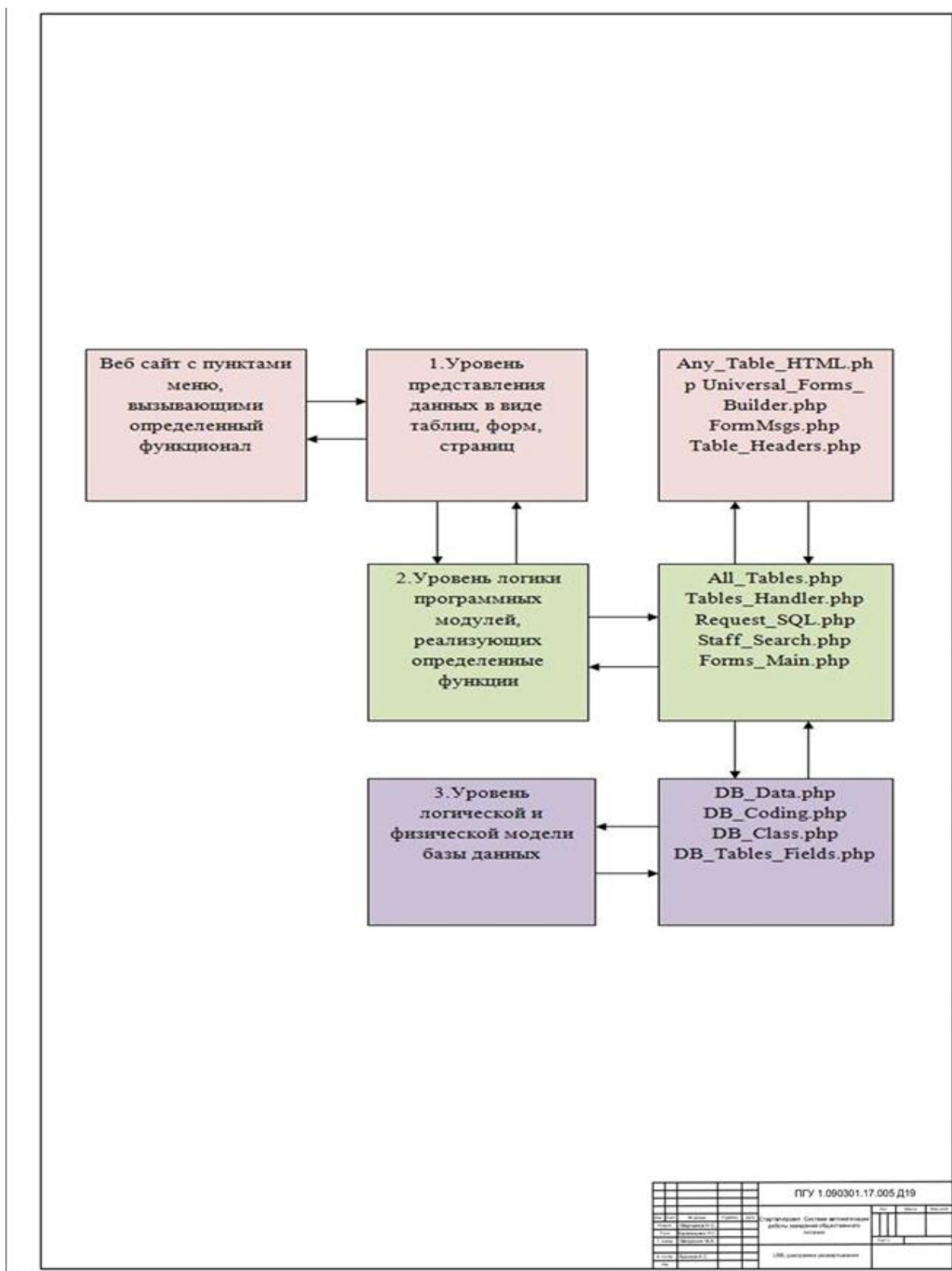
Приложение А.6 – UML – диаграмма последовательности работы
тестируемого



Приложение А.7 – UML – диаграмма последовательности работы
преподавателя



Приложение А.8 – UML – диаграмма классов структурная схема
программного обеспечения



ПГУ 1.090301.17.005 Д19			
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель
Исполнитель	Исполнитель	Исполнитель	Исполнитель

Приложение Б

(обязательное)

Обучающее Web-приложение для изучения нейронных сетей

Листинг программы

					ПГУ1.090301.45.001 ПЗ	Лист
						74
Изм.	Лист	№ докум.	Подпись	Дата		

Файл «index_admin.php»

```

<?php
session_start();
include($_SERVER["DOCUMENT_ROOT"]."/TestsStud/Header.php");
?>
<p></p>
<table bgcolor="#999999" border="0" height=8 width="660"
cellpadding="1" cellspacing="1" align="center">
  <tr>
    <td align="center" bgcolor="#FFFFFF">
      <p>
<img src = '../..TestsStud/Pictures/lock_red.gif'>&nbsp;
<br>
<?php
$Status      ='Преподаватель';
$username    =$_SERVER['REMOTE_USER'];

print "<br><strong>Добро пожаловать ".$username."<br>Вы успешно
прошли авторизацию!</strong>";

?>
<br><a
href="/TestsStud/Tables_Handler.php?FormType=replies&username=<?
php print $username;?>&Status=<?php print $Status;?>">
      Кликните по ссылке для просмотра результатов
тестирования.
    </a>
<br><a
href="/TestsStud/Tables_Handler.php?FormType=questions&username=
<?php print $username;?>&Status=<?php print $Status;?>">
      Кликните по ссылке для изменения или добавления вопросов
по тестам.
    </a>
<br><a
href="/TestsStud/Tables_Handler.php?FormType=students&username=<
?php print $username;?>&Status=<?php print $Status;?>">
      Кликните по ссылке для изменения или добавления данных
учащихся.
    </a>
<p></p>

<p>
<img src = '../..TestsStud/Pictures/lock_gre.gif'>&nbsp;
<br>
</p>
<br>
</p>
</td>
</tr>
</table>

```

```
<?php
include($_SERVER["DOCUMENT_ROOT"]."/TestsStud/Footer.php"); ?>
```

Файл «index-member.php»

```
<?php
session_start();
include($_SERVER["DOCUMENT_ROOT"]."/TestsStud/Header.php");

$Status      ='Студент';
$FormType     ='students';
$username     =$_SERVER['REMOTE_USER'];
$_SESSION['username'] = $username;
$_SESSION['Status'  ] = $Status;
$IndSearch    = '2';
$KeyValue     = $username;
include
($_SERVER["DOCUMENT_ROOT"]."/TestsStud/FirstKey_Search.php");
$FIO          = $Nm;
$_SESSION['FIO'      ] = $FIO;
print '
    <p></p>
    <p></p>
    <table bgcolor="#999999" border="0" height=8 width="860"
cellpadding="1" cellspacing="1" align="center">
        <tr>
            <td align="left" bgcolor="#FFFFFF">
';
print '
    <center>

                                <img          src          =
"../../TestsStud/Pictures/lock_gre.gif">&nbsp;
    </center>
';
print "
    <center>

                                <font    color='Red'    size='4'>КАБИНЕТ
ТЕСТИРУЕМОГО<br></font>
                                <font    color='darkblue'    size='4'>Добро
пожаловать <br> ".$FIO."</font>
                                </font>
';
?>
<p></p>
<font color="darkblue" size="4">
<img src = ' ../../TestsStud/Pictures/Chelovecheck50.png'>
<br>
<a
href="/TestsStud/Test_Select1.php?username=<?=$username?>&Status
=<?=$Status?>">
```

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		76

```

        ПЕРЕХОД К ТЕСТИРОВАНИЮ
</a>

<!--    //-->
<br>
<font color="Maroon" size="4">
<img src = '../..TestsStud/Pictures/OneClientSmall.jpg'>
<br>
<a
                                href
                                =
"../..TestsStud/1Key_Search.php?FormType=replies&IndSearch=4&Key
Value=?=$username?">
        РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ
</a>

</font>
</p>
</center>
<br>
<br>
</td>
</tr>
</table>
<?php
include($_SERVER["DOCUMENT_ROOT"]."/TestsStud/Footer.php"); ?>

```

Файл «1Key-Search.php»

```

<?php
#####
#####
# Модуль производит поиск по полю - Id
#
#
#
#####
#####
#
session_start();
$Status      = $_SESSION['Status'  ];
$username    = $_SESSION['username'];
$FIO         = $_SESSION['FIO'      ]; // FIO влвдельца
кабинеты
#
$FormType    = $_GET [FormType  ];
$IndSearch   = $_GET [IndSearch ];
$KeyValue    = $_GET [KeyValue  ];
if            (!$_GET [KeyValue  ])
$KeyValue    = $_POST[KeyValue  ]; // это если после
кнопки
/*

```

```

print          "<br>FormType    =$FormType";
print          "<br>IndSearch  =$IndSearch";
print          "<br>KeyValue    =$KeyValue";
*/
#
$Arr_acc       = array();
$Row           = array();
$dbtable       = $FormType;
#
# в ранних версиях был файл require_once 'Table_Headers_DB.php';
# Заголовки всех таблиц БД проекта, на английском формируются
динамически
require_once 'DB_Tables_Fields.php';
#
# Класс для построения таблиц по заголовкам и содержанию таблиц
require_once 'Any_Table_HTML_New.php';

# Класс для работы с базой данных MySQL, описание в теле модуля

require_once 'DB_Class.php';
#
# Заголовки всех таблиц веб проекта, которые появляются на
экране
include_once 'Table_Headers.php';
#
# Так мы делаем софт более независимым от конкретных таблиц БД
#
$headers       = $Table_Headers    [$FormType]; // наименования
столбцов таблице
$headers_db    = $Table_Headers_DB[$FormType]; // наименования
доменов в таблице
#
#                                     // БД и <input name's
в формах
# Создаем объекты класса для вывода таблиц, вызываем конструктор
#$handler     = $FormType.'_Handler';          // обработчик
настоящей формы

#
# Объект класса работы с базой данных создается внутри модуля
DB_Class.php
// $dbObj      = new MySQL_db(DB_HOST, DB_USER,
DB_PWD, DB_NAME);
#
#include 'Client_form_valid.php // проверяет введенные данные

*****                                     Поиск                                     ключу
*****

#
$KeyName       = $headers_db[$IndSearch];
$queryResult   = $dbObj->query("SELECT * FROM $dbtable WHERE
$KeyName      ='$KeyValue' OR $KeyName      ='$FIO' ORDER BY
$headers_db[5] ");

```

```

// $QueryResult = $dbObj->query("SELECT * FROM $dbtable WHERE
$KeyName ='$KeyValue' OR $KeyName ='$FIO' ORDER BY '$KeyName'
");
if ($QueryResult==false)
{
    $msg1 = 'Запись не найдена в таблице';
    // $QueryResult = $dbObj->query("SELECT * FROM $dbtable ORDER
    BY '$KeyName' ");
}
else
{
    /* print '<pre>';
    print_r($QueryResult);
    print '</pre>'; */
}
$head_color = 'silver';
$Table_view = new HTML_Table($headers , $head_color);
foreach ($QueryResult as $row => $Arr_acc)
{
    $i=0;
    foreach ($headers_db as $element)
    {
        $Row = array_merge($Row, array($headers[$i] =>
$Arr_acc[$headers_db[$i]]) );
        $$headers_db[$i] = $Arr_acc[$headers_db[$i]];
        $i++;
    }
    $Table_view -> AddRowAssArr($Row);
}
include'Header.php'; // общий заголовок для всех
страниц проекта
print "<p></p><font color='blue'><strong>РЕЗУЛЬТАТ
ПОИСКА</strong>&nbsp;$msg1&nbsp;</font>";
$Table_view -> PrintArr();
#
?>

```

Файл «Add-Record.php»

```

<?php
#####
#####
# Добавление записи в любую таблицу БД
#
# Входным параметром является имя таблицы БД FormType
#
# (т.к.каждой таблице соответствует форма)
#
# Параметр передается по GET, или непосредственным заданием
значения
#

```

```

# Определяются все поля таблицы. Формируется строка через
запятую. #
# Так мы делаем софт более независимым от конкретных таблиц БД
#
#####
#####
#
session_start();
$Status = $_SESSION['Status'];
$username = $_SESSION['username'];
$FIO = $_SESSION['username'];
#
if ($_GET[FormType])
$FormType = $_GET[FormType];
#
require_once 'DB_Tables_Fields.php'; //
Считывание всех таблиц БД
#
$dbtable = $FormType;
#
# Класс для работы с базой данных MySQL, описание в теле модуля
'DB_Tables_Fields.php';
#
# Заголовки всех столбцов таблиц, которые появляются на экране.
Формируем сами.
include_once 'Table_Headers.php';
#
# Так мы делаем софт более независимым от конкретных таблиц БД
#
$headers = $Table_Headers [$FormType]; //
Наименования столбцов в таблице на русском
$headers_db = $Table_Headers_DB[$FormType]; //
Наименования доменов в таблице БД
# // как
результат DB_Tables_Fields.php
# Объект класса работы с базой данных создается внутри модуля
DB_Class.php
#
#include 'Client_form_valid.php // проверяет введенные данные
#
# Добавление записи в таблицу БД. Данные передаются методом
POST из
# Forms_Main.php. Нажата кнопка Добавить. Или из другой
формы.
#
$i=0;
foreach ($headers_db as $element) //
{
    $headers_db[$i]-значения полей, данные
    $$headers_db[$i] = $_POST[$headers_db[$i]];
    $Str2.=",".".$$headers_db[$i]."";
    $i++;
}

```

Изм.	Лист	№ докум.	Подпись	Дата

```

    }
    $Str2=substr($Str2,1); // Это
    список данных для занесения в БД через запятую
    $$headers_db[0] = substr($$headers_db[0],0,10); // в первой
    колонке всегда первичный ключ ???
    # // или ключ
    формируется (или нет) в форме
    foreach ($FormTypes as $current) //
    FormTypes[] из DB_Tables_Fields.php
    {
        if ($FormType==$current){
            include 'Request_SQL.php'; //The result in
$QueryResult
        } //
    }
    if ($QueryResult==1)
    { $msg1 = 'Новые данные успешно добавлены ';
        if (($FormType=='students')||($FormType=='teachers'))// 19-
10-2017 !!!
            include 'Add_Record_Htpass.php'; //
Добавляется Погин:Пароль в htpass
        }
    else
        $msg1 = "При вводе данных произошла
ошибка!
Данные уже есть в таблице ";
    //echo $msg1;
    //exit;
    if (($FormType=='students')||($FormType=='teachers'))
    {
        include 'Header.php';
        echo $msg1;

        //header("Location:".$DOCUMENT_ROOT."/TestsStud/Result.php?msg1=
$msg1&FormType=$FormType");
        exit;
    }
    else
    header("Location:".$DOCUMENT_ROOT."/TestsStud/Tables_Handler_Fra
me.php?msg1=$msg1&FormType=$FormType");
    exit;
    //include 'Result.php'; все портит
?>

```

Файл «Add-Record-Htpass.php»

```

<?php
if (($FormType=='students')||($FormType=='teachers'))
{
    $filename = "$_SERVER[DOCUMENT_ROOT]/TestsStud/htpass";
    include 'Pass_Encrypt.php';
}

```

```

$Str_LP    = $$headers_db[0].':'. $$headers_db[1];
$fp        = fopen($filename,"a");
if ($fp    ==false)
    $fp     = fopen($filename,"w");
fputs ($fp,"$Str_LP\n");
fclose($fp);
//print "Str_LP=$Str_LP";
}
?>

```

Файл «All-Tables.php»

```

<?php
#####
#####
# Модуль выводит на экран содержимое базы данных по-таблично.
#
# Полностью все данные не выводятся на экран, а только первые
две строки
#
#####
#####
#
require_once          'DB_Tables_Fields.php';  //!!!!//
$Arr_acc              = array();
$Row                  = array();
$dbtable              = $FormTypes;
//$head_color          = 'silver';
# Класс для построения таблиц по заголовкам и содержанию таблиц
require_once 'Any_Table_HTML_New.php';

# Класс для работы с базой данных MySQL, описание в теле модуля

# require_once 'DB_Class.php';
# Объект класса работы с базой данных создается внутри модуля
DB_Class.php
# $dbObj              = new MySQL_DB(DB_HOST, DB_USER,
DB_PWD, DB_NAME);
###
# Так мы делаем софт более независимым от конкретных таблиц БД
#
# Заголовки всех таблиц веб проекта, которые появляются на
экране
include_once 'Table_Headers.php';                //!!!!// если уже
созданы русские заголовки
#
# Заголовки всех таблиц БД проекта, на английском
# require_once 'Table_Headers_DB.php';            //!!!! В этой
версии загружаются из БД
#
#
// на лету
программой DB_Tables_Fields.php

```

```

require_once 'Header.php'; // заголовок для
страниц проекта
require_once 'FormMsgs.php'; // все сообщения
программ проекта
#
#
$f=0; // номер элемента $FormType
foreach ($dbtable as $value) {
#
$headers = $Table_Headers [$FormTypes[$f]]; // наименования
столбцов таблицы
$headers_db = $Table_Headers_DB[$FormTypes[$f]]; // наименования
доменов в таблице
# // БД и <input
name's в формах
###
# Создаем объекты класса для вывода таблиц, вызываем конструктор
# $handler = $FormType.'_Handler'; // обработчик настоящей
формы

$Table_view = new HTML_Table($headers , $heard_color);
#
#запрос на выдачу всей таблицы

$QueryResult = $dbObj->query("SELECT * FROM $dbtable[$f] LIMIT
0,2");
# возвращает массив ассоциативных массивов (см. класс
DB_Class.php)
# по сути двумерный массив, поэтому используем конструкцию
foreach
# это можно увидеть убрав знак # перед print см. ниже. Таким
образом
# формируются массивы для вывода на экран для каждой таблицы
#
if ($QueryResult==false)
{
$msg = 'Нет записей в базе данных!';
}
else
{
$NA = count($QueryResult);
//print '$NA='.$NA;
//print '<pre>'; print_r($QueryResult); print '</pre>';
}
foreach ($QueryResult as $row => $Arr_acc)
{
$i=0;
foreach ($headers_db as $element)
{
$Row = array_merge($Row, array($headers[$i] =>
$Arr_acc[$headers_db[$i]]) );
$i++;

```

```

    }
    $Table_view -> AddRowAssArr($Row);
}
print "<p></p><font color= \"blue\"
><strong>$All_Tables_Names[$f]</strong></font>"
    ".$Comment_to_Table_up[$f]";
    $Table_view -> PrintArr();
print "<a href=\"Tables_Handler.php?FormType=$FormTypes[$f]\">
    Все данные Таблицы&nbsp;\"$All_Tables_Names[$f]\"</a>"
    .'<p></p>';
//print "
href=\"\"$FormType[$f]_Handler.php?FormType=$FormType[$f]\">
// Все данные Таблицы&nbsp;\"$All_Tables_Names[$f]\"</a>"

#
$f++; // новое значение $FormType,
новая таблица
}
#
include 'Footer.php'; // стандартная нижняя строка
каждой страницы
#
?>

```

Файл «Any-Table-HTML-New.php»

```

<?php
class Any_Table
{
    var $table_array = array();
    var $headers      = array();
    var $domns;
    function Any_Table ($headers)
    {
        $this->headers = $headers;
        $this->domns   = count($headers);
    }

    function AddRow ($row)
    {
        If ( count ($row) != $this->domns)
            return false;
        array_push( $this -> table_array,
$row);
        return true;
    }
    function SetArr($Arr)
    {
        $this -> table_array = $Arr;
        return true;
    }
}

```

```

    }

    function AddRowAssArr ($row_ass)
    {
        $row=array();
        foreach ($this -> headers as
$header)
        { if (!isset($row_ass[$header]))
            $row_ass[$header]='';
            $row[ ]=$row_ass[$header];
        }
        array_push( $this -> table_array,
$row);
        return true;
    }
    function PrintArr ()
    {
        print "<pre>";
        foreach ($this -> headers as
$header)
            print '
            <strong>$header</strong>'. ' ';
        print "\n";
        foreach ($this -> table_array as
$string)
        {
            foreach ($string as
$elemnt)
                print
                '    <strong>$elemnt</strong>'. ' ';
            print "\n";
        }
    }
}
class HTML_Table extends Any_Table
{
    var $bgcolor;
    var $bgcolor_h = "#FFCCCC";
    var $bgcolor_c = "#FFFFFF";
    var $cellpadding = "1";
    function HTML_Table($headers, $bg='#ffffff' )
    {
        Any_Table::Any_Table($headers);
        $this->bgcolor=$bg;
    }
    function SetCellPadding ($padding)
    {
        $this -> cellpadding = $padding;
    }
    function PrintArr()
    {

```

```

                                print "<table width=\"99%\"
bgcolor=\"#999999\"    cellspacing=\"1\"    cellpadding=\">cellpadding\" border=0>";
                                print "<tr>";
                                foreach ($this -> headers as $header)
                                    print "<td align=\"center\"
bgcolor=\">$this->bgcolor_h\">$header</td>";
                                print "</tr>";
                                foreach ($this -> table_array as $row=>$cells)
                                {
                                    print "<tr>";
                                    foreach ($cells as $elemnt)
                                        print "<td align='center'
bgcolor=\">$this->bgcolor_c\" >$elemnt</td>";
                                    print "</tr>";
                                }
                                print '</table>';
                            }
                            function OutForm($handler)
                            {
                                $width_h = array(170, 360);
                                print "<form name='uniform'
action=\">$handler\" method=post>";
                                print "<table width=\"99%\"
bgcolor=\"#999999\"    cellspacing=\"1\"    cellpadding=\">$this-
>cellpadding\" border=0>";
                                print "<tr>";
                                $i=0;
                                foreach ($this -> headers as $header){
                                    print "<td
width=\".$width_h[$i].\"    align=\"left\"    bgcolor=\">$this-
>bgcolor_h\">&nbsp;$header</td>";
                                    $i++;
                                }
                                print "</tr>";
                                $i=0;
                                foreach ($this -> table_array as $row=>$cells)
                                {
                                    print "<tr>";
                                    foreach ($cells as $elemnt)
                                    {
                                        print "<td
width=\".$width_h[$i].\"    align='left'    bgcolor=\">$this-
>bgcolor_c\" >$elemnt</td>";
                                        if ($i==0) $i++; else $i-
-;
                                    }
                                    print "</tr>";
                                }
                                print '</table>';
                                print '</form>';
                            }

```

}

?>

Файл «DB-Class.php»

```
<?php
class MySQL_DB
{
    private $dbObj = null;
    private $result = null;

    /* Для конструктора - адрес, имя пользователя, пароль, имя базы
    данных, порт,
    /*а также кодировку для соединения.
    /* По умолчанию используется utf8 */

    public function __construct($host, $user, $password,
    $base, $port = null, $charset = 'utf8')
    {
        $this->dbObj = new mysqli($host, $user,
    $password, $base, $port);
        $this->dbObj->set_charset($charset);
    }

    /*основная и единственная функция, которая выполняет запрос и
    возвращает результат его работы*/

    public function query($query)
    {
        if(!$this->dbObj)
            return false;

        /*очищаем предыдущий результат*/

        if(is_object($this->result))
            $this->result->free();

        /*выполняем запрос*/

        $this->result = $this->dbObj->query($query);

        /*если есть ошибки - выводим их*/

        if($this->dbObj->errno)
        #!!!#
            die("mysqli error #".$this->dbObj->errno." :
        ".$this->dbObj->error);
        return $this->dbObj->error;
    }
}
```

```

/*если в результате выполнения запроса (например SELECT...)
получены данные - возвращаем их.
/* данные всегда возвращаются в массиве, даже если запрос
возвращает одну запись.*/

        if(is_object($this->result))
        {
                                while($row = $this->result-
>fetch_assoc())
                                $data[] = $row;

                                return $data;
        }

/*если результат отрицательный - возвращаем false */

        else if($this->result == FALSE)
                return false;

/*если запрос (например UPDATE или INSERT) затронул какие-либо
строки - возвращаем их количество*/

        else return $this->dbObj->affected_rows;
    }
}
include 'DB_data.php';
$dbObj = new MySQL_DB(DB_HOST, DB_USER, DB_PWD, DB_NAME);
include 'DB_Coding.php';
?>

```

Файл «DB-data.php»

```

<?php
#####
#####
define("DB_HOST",          "localhost"          ); // адрес сервера
баз данных      #
define("DB_USER",          "root"                ); // имя
пользователя    #
define("DB_PWD",           ""                    ); // пароль к бд
#
define("DB_NAME",          "j55600682_testsdll" ); // имя базы данных
#
define("DB_MODE",          2); // 0 - в ручную
откр/закр,      #
#####
#####
?>

```

Файл «DB-Tables-Fields.php»

```
<?php
#####
# Пример использования SHOW COLUMNS и SHOW COLUMNS #
#####
#
require_once      'DB_Class.php';
$Arr_acc_tables  = array();
$Arr_acc_fields  = array();
$Tables          = array();
$fields          = array();
$fields_1        = array();
$Work            = array();
$Table_Headers_DB = array();
$SQL_String1     = array();
$SQL_String2     = array();
#
$Tables_DB_Name   = 'Tables_in_'.DB_NAME;
#
$QueryResult = $dbObj->query("SHOW TABLES");
# ассоциированный массив с именами таблиц
# Array
# (
#     [0] => Array
#         (
#             [Tables_in_control_clients_adv] => Clients
#         )
#     ...
#
#     print '<pre>';
#     print_r($QueryResult);
#     print '</pre>';
#
$i=0;
foreach ($QueryResult as $row_tables => $Arr_acc_tables)
{
    $Tables[$i] = $Arr_acc_tables[$Tables_DB_Name];
    $QueryResult = $dbObj->query("SHOW COLUMNS FROM
$Tables[$i]");
#
# Array
# (
#     [0] => Array
#         (
#             [Field] => Id_Client
#             [Type] => char(10)
#             [Null] => NO
#             [Key] => PRI
#             [Default] =>
```

```

#           [Extra] =>
#           )
           $j=0;
           $string = '';
           $string1='';
           $string2='';
           foreach ($QueryResult as $row_fields =>
$Arr_acc_fields)
           {
               $Fields[$i][$j] = $Arr_acc_fields[Field];
               $Fields_1 [$j] = $Arr_acc_fields[Field];
               $string.= ',' . $Fields[$i][$j];
               $string1.="," . $Fields[$i][$j];
               $string2.=",$" . $Fields[$i][$j];
               $j++;
           }
//           print '<br>'. $Tables[$i]. $string;
//           print '<br>'. $Tables[$i]. $string1;
//           print '<br>'. $Tables[$i]. $string2;

//           print '<pre>';
//           print_r($QueryResult);
//           print '</pre>';
           $string1 = substr($string1,1);
           $string2 = substr($string2,1);
           $Table_Headers_DB =
array_merge($Table_Headers_DB,array($Tables[$i] => $Fields_1));
           $SQL_String1 = array_merge($SQL_String1
, array($Tables[$i] => $string1 ));
           $SQL_String2 = array_merge($SQL_String2
, array($Tables[$i] => $string2 ));
           $Fields_1 = $Work;
           $i++;
       }
$FormTypes=$Tables;
### Эту часть только при первом запуске, пока не созданы
заголовки Table_Headers.php
include_once 'Table_Headers.php';
if (!$Table_Headers)
{
    $Table_Headers = $Table_Headers_DB;
    print '<br>***** $Table_Headers *****';
    print '<pre>';
    print_r($Table_Headers);
    print '</pre>';
}
//print '<br>***** $Fields *****';
//print '<pre>';
//print_r($Fields);
//print '</pre>';
//print '<br>***** $Table_Headers_DB *****';
//print '<pre>';

```

```
//print_r($Table_Headers_DB);
//print '</pre>';
//print '<pre>';
//print_r($SQL_String1);
//print '</pre>';
//print '<pre>';
//print_r($SQL_String2);
//print '</pre>';
#
?>
```

Файл «FirstKey-Search.php»

```
<?php
#####
#####
# Модуль производит поиск по полю - Id
#
#
#
#####
#####
#
$Arr_w          = array();
$Row_w          = array();
$dbtable        = $FormType;
#
# в ранних версиях был файл require_once 'Table_Headers_DB.php';
# Заголовки всех таблиц БД проекта, на английском формируются
динамически
require_once 'DB_Tables_Fields.php';
#
# Класс для построения таблиц по заголовкам и содержанию таблиц
require_once 'Any_Table_HTML_New.php';

# Класс для работы с базой данных MySQL, описание в теле модуля

require_once 'DB_Class.php';
#
# Заголовки всех таблиц веб проекта, которые появляются на
экране
include_once 'Table_Headers.php';
#
# Так мы делаем софт более независимым от конкретных таблиц БД
#
$headers        = $Table_Headers      [$FormType]; // наименования
столбцов таблице
$headers_db     = $Table_Headers_DB[$FormType]; // наименования
доменов в таблице
```

Изм.	Лист	№ докум.	Подпись	Дата

```

#                                                                 // ВД и <input name's
в формах
# Создаем объекты класса для вывода таблиц, вызываем конструктор
#$handler          = $FormType.'_Handler';          // обработчик
настоящей формы

#
# Объект класса работы с базой данных создается внутри модуля
DB_Class.php
//$dbObj              = new MySQL_db(DB_HOST, DB_USER,
DB_PWD, DB_NAME);
#
#include 'Client_form_valid.php // проверяет введенные данные

*****                               Поиск                               ключу
*****
$KeyName            = $headers_db[0];
$queryResult         = $dbObj->query("SELECT * FROM $dbtable WHERE
$KeyName='$KeyValue' ORDER BY '$KeyName' ");
if ($queryResult==false)
{
    $msg1            = 'Запись не найдена в таблице';
    // $queryResult = $dbObj->query("SELECT * FROM $dbtable ORDER
BY '$KeyName' ");
}
else
{
    // print '<pre>';
    // print_r($queryResult);
    // print '</pre>';
    // $msg1         = '<br>Запись найдена в базе данных';
    $Nm              = $queryResult[0][$headers_db[$IndSearch]];
}
print $msg1;
?>

```

Файл «Form-Buttons.php»

```

<!--без таблиц //-->
<p></p>
    <form      action="Tables_Handler.php?FormType=<?php      print
$FormType?>" method=post>
    <!--form  action="<?php      //print      $FormType;?>_Handler.php"
method=post/-->
    <strong>Введите Id код для удаления, поиска или
обновления&nbsp;  </strong>
    <input type="text"      name      ="<?php print      $headers_db[0];?>"
size=10 maxlength=10
                                                                value      ="<?php      print
$$headers_db[0];?>">

```

```

        <INPUT TYPE="submit" NAME="Delete" VALUE="Удалить">
        &nbsp;
        <INPUT TYPE="submit" NAME="Search" VALUE="Найти">
        &nbsp;
        <INPUT TYPE="submit" NAME="Update" VALUE="Обновить">
        &nbsp;
    </form>

```

Файл «Forms-Main.php»

```

<?php
#####
#####
# Модуль выводит на экран форму для добавления данных в любую
таблицу #
# Если модуль вызывается по ссылке
href="Forms_Main.php?formtype=Staff", в этом #
# случае работает Get.
#
# Однако модуль может вызываться и в случае обновления данных в
таблице, тогда #
# см. например Staff_Handler.php ...if
(isset($_POST["Update"]))... строка ~147 #
# в полях формы появляются данные для обновления
#
#
#
#####
#####
if ($_GET['formtype']) //!!!!//
    $FormType = $_GET['formtype'];
require_once 'DB_Tables_Fields.php'; //!!!!//
# В этой версии формируется программой 'DB_Tables_Fields.php'
//require_once 'FormTypes.php'; //!!!!//
#
require_once 'FormMsgs.php';
#
# Форма для добавления, удаления, обновления данных по
подразделениям,
# сотрудникам, приборам, изделиям, результатам испытаний и пр.
# Формируются исходные данные для формы и вызывается
универсальный
# построитель форм такого вида Universal_Forms_Builder.php
###
# В случае вызова модуля для заполнения включаются эти include-ы
include_once 'Header.php'; // Общий заголовок
веб-страниц проекта
include_once 'Table_Headers.php'; // Заголовки таблиц
и форм таблиц

```

```

# В этой версии Table_Headers_DB формируется программой
'DB_Tables_Fields.php'
# include_once 'Table_Headers_DB.php';           // Заголовки таблиц
базы данных
include_once 'Any_Table_HTML_New.php';           // Класс для
построения таблиц по                                     // заголовкам и
содержанию
#
$handler      = 'Tables_Handler.php?FormType='.$FormType;
                                                    // обработчик
настоящей формы
#$handler      = $FormType.'_Handler';           // обработчик
настоящей формы
$headers       = $Table_Headers [$FormType]; // наименования
столбцов таблиц
$headers_db    = $Table_Headers_DB[$FormType]; // наименования
доменов в таблице
$head_color   = "silver";                       // Staff БД и input
name's в формах
$form_headers = array(' Наименования полей',
                      ' Заносимые или обновляемые данные'
                      );
# Создаем объекты класса для вывода таблиц, вызываем конструктор
#
$Form_view    = new HTML_Table($form_headers , $head_color);
# Заголовок формы страницы
print ' <p><font color= "red" ><strong>'
      . $FormMsgs[$FormType][0].
      '</strong></font>'
      </p>'; // Заголовок формы
страницы
#
#           построитель           простых           форм           такого           вида
Universal_Forms_Builder.php
#
include 'Universal_Forms_Builder.php';
?>
<p></p>
</body>
</html>

```

Файл «Header.php»

```

<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-
8">
<title>Обучающее Web приложение

```

					ПГУ1.090301.45.001 ПЗ	Лист
						94
Изм.	Лист	№ докум.	Подпись	Дата		

```

</title>
<style>
    a {
        text-decoration: none; /* Отменяем подчеркивание у
ССЫЛКИ */
    }
    A IMG
    {
        border: none; /* Убираем рамку в изображениях-
ССЫЛКАХ */
    }
    body
    {
        background: #E0FFFF; /* Цвет фона
        */
        color: Maroon; /* Цвет текста
        */
    }
    td
    {
        vertical-align: top; /* Выравнивание по верхнему краю
ячеек */
        font-size: 13pt; /* Размер шрифта в пунктах
        */
    }
    .required
    {
        color: red;
    }
</style>
</head>
<body>
<table bgcolor="#FFFFFF" border="0" height=8 width="99%"
cellpadding="5" cellspacing="0">
<tr>
<td colspan="9">
        
</td>
</tr>
<tr height=20><td width="11%" align="center" bgcolor="blue">
        <a href="../../../TestsStud/index.php"><font color="#FFFFFF"
size="4"><strong>СУТЬ ПРОЕКТА</strong></font></a>
</td>

        <td width="11%" align="center" bgcolor="blue">
                <a href="../../../TestsStud/Instruction.php"><font
color="#FFFFFF" size="4"><strong>УЧЕБНЫЕ
МАТЕРИАЛЫ</strong></font></a>
</td>

        <td width="11%" align="center" bgcolor="blue">

```

| [ПРЕПОДАВАТЕЛЬ](../../TestsStud/Entry_System_Teacher.php) |

CTYДЕНТ
--

```
<td width="11%" align="center" bgcolor="blue">
  <a href="#../..TestsStud/Books.php"><font color="#FFFFFF"
size="4"> <strong></strong></font></a>
</td>
```

 | | |

--	--

--

[illegible]

```

</strong></font>
</p>!-->
</td>
</tr>
</table>
</body>
</html>

```

Файл «Index.php»

[illegible]

Файл «Req-Frame.php»

```
<?php
header("Content-Type: text/html; charset=utf-8");
require_once 'Header.php';
?>
<p></p>
```

```

<p></p>
<p></p>
<table bgcolor="#999999" height=20 width="99%" cellpadding="1" cellspacing="1" border="0">
</table>
<p></p>
<p></p>
<p></p>
<p></p>
<p></p>
<table bgcolor="#999999" height=20 width="60%" cellpadding="1" cellspacing="1" border="0">
<tr>
<td align="left" valign="top" bgcolor="#E6E6FA">

<font color="darkblue" size="4">
<h3>РЕГИСТРАЦИЯ ПОЛЬЗОВАТЕЛЕЙ</h3>
Регистрация пользователей подразумевает создание логина и пароля
с целью входа в защищенные области сайта.
Зарегистрированные пользователи-учащиеся, могут затем пройти
тестирование и просмотреть результаты тестирования.
Для этого они должны авторизоваться и зайти в свой личный
кабинет пользователя.
Зарегистрированные преподаватели могут просматривать результаты
тестирования всех учащихся.

<?php
$operation='ADD';
include 'Forms_Main_Reg.php';
?>

</td>
</tr>
</table>
<p></p>
<p></p>
<p></p>

<?php require_once 'Footer.php'; ?>

```

Файл «regection.htm.php»

```

<html>
<head>
<title>Авторизация не прошла </title>
<style>
a {
text-decoration: none; /* Отменяем подчеркивание у ссылки
*/
}

```

```

A IMG
{
border: none; /* Убираем рамку в изображениях-ссылках
*/
}
body
{
background: #E0FFFF; /* Цвет фона
*/
color: Maroon; /* Цвет текста
*/
}
td
{
vertical-align: top; /* Выравнивание по верхнему краю ячеек
*/
font-size: 13pt; /* Размер шрифта в пунктах
*/
}
.required
{
color: red;
}
</style>
</head>
<body>
<table bgcolor="#FFFFFF" border="0" height=8 width="99%"
cellpadding="0" cellspacing="0">
<tr>
<td colspan="9">

</td>
</tr>
<tr><td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/index.php"><font color="Gold"
size="4"><strong>О Проекте</strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/Instruction.php"><font color="Gold"
size="4"> <strong>Инструкция</strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/Register_System.php"><font
color="Gold" size="4"> <strong>Регистрация</strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/Entry_System_Teacher.php"><font
color="Gold" size="4"> <strong>Преподаватель</strong></font></a>

```

```

</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/Entry_System_Pupil.php"><font
color="Gold" size="4"> <strong> Студент </strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/Books.php"><font color="Gold"
size="4"> <strong>Лекции </strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/News.php"><font color="Gold" size="4">
<strong>Новости</strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href=""><font color="Gold"
size="4"><strong></strong></font></a>
</td>

<td width="11%" align="center" bgcolor="blue">
<a href="../../../TestsStud/All_Tables.php" ><font color="Gold"
size="4"> <strong></strong></font></a>
</td>
</tr>
</table>
<p><p>
<table bgcolor="#999999" border="0" height=8 cellpadding="1"
cellspacing="1" align="center">
<tr>
<td bgcolor="#FFFFFF">
<center>
<img src = "../../../TestsStud/Pictures/lock_red.gif">&nbsp;
<br>
<br>
<strong><font color="red">Вам не удалось пройти авторизацию!
</font></strong>
<br>
<p></p>
Только зарегистрированные пользователи имеют доступ к системе.
<br>Просим вас зарегистрироваться в системе!
<br>
<br>
</center>
</td>
</tr>
</table>
</body>
</html>

```

Файл «Request-SQL.php»

```

<?php
#   require_once 'DB_Tables_Fields.php' формирует первую строку
запроса $Str1
#   в зависимости от таблицы БД ($FormType). Значения $Str2
формируются
#   динамически в вызывающей программе
#
    $Str1=$SQL_String1[$FormType];
//   $Str2=$SQL_String2[$FormType];
    $QueryResult = $dbObj->query("INSERT INTO $dbtable
                                (
                                    $Str1
                                )
                                values
                                (
                                    $Str2
                                )"
                                );

?>

```

Файл «Table-Headers.php»

```

<?php
$Table_Headers =
Array
(
    'groups' => Array                                     # Данные
заполняет преподаватель #
    (
        'Группа (Класс)'                                ,      # [0] =>
Group_Id
        'Наименование'                                  ,      # [1] =>
Group_Name
        'Факультет'                                     ,      # [2] =>
Faculty
        'Ориентация '                                   ,      # [3] =>
Speciality
        'Руководитель'                                  ,      # [4] =>
Chief_FIO
        'Примечания'                                    ,      # [5] =>
Group_Com
    ),

    'questions' => Array                                  # Данные
заполняет преподаватель #

```

```

(
    'Код вопроса' , # [0] =>
    Quest_Id
    '№ вопроса' , # [1] =>
    Quest_N
    '№ предмета' , # [2] =>
    Subj_Id
    'Предмет' , # [3] =>
    Subj_Nm
    'Тест' , # [4] =>
    Test_Id
    'Наименование вопроса' , # [5] =>
    Quest_Text
    'Первый ответ' , # [6] =>
    Repl_1
    'Второй ответ' , # [7] =>
    Repl_1
    'Третий ответ' , # [8] =>
    Repl_1
    'Четвертый ответ' , # [9] =>
    Repl_1
    '№ верного' , # [10]=>
    Repl_N
    'Очки' # [11]=>
    Repl_Point
),

```

```

'replies' => Array # Данные
берутся из БД Questions #
( # Вносится
номер правильного ответа #
    'Код ответа' , # [0] =>
    Repl_Id
    '№ предмета' , # [1] =>
    Subj_Id
    'Предмет' , # [2] =>
    Subj_Nm
    'Тест' , # [3] =>
    Test_Id
    'Код учащегося' , # [4] =>
    Stud_Id
    '№ вопроса' , # [5] =>
    Quest_N
    'Наименование вопроса' , # [6] =>
    Quest_Text
    'Ответ тестируемого' , # [7] =>
    Repl_U
    'Правильный ответ' , # [8] =>
    Repl
    'Очки' # [9] =>
    Repl_Point

```

```

    ),

    'students' => Array                                # Данные
заносятся студентом #
    (
        'Код учащегося'                                , # [0] =>
Stud_Id
        'Пароль'                                        , # [1] =>
Stud_Pass
        'ФИО'                                           , # [2] =>
Stud_FIO
        'Факультет'                                    , # [3] =>
Stud_Faculty
        'Группа'                                        , # [4] =>
Stud_Group
        'Год Поступления'                              , # [5] =>
Stud_Year
        'Специализация'                                , # [6] =>
Stud_Spec_N
        'Наименование'                                , # [7] =>
Stud_Spec_Nm
        'Телефон'                                       , # [8] =>
Stud_Phone
        'Электронный адрес'                            , # [9] =>
Stud_Email
    ),

    'subjects' => Array                                # Данные
заносятся пре-лем #
    (
        'Идентификатор'                                , # [0] =>
Subj_Id
        'Номер или Код'                                , # [1] =>
Subj_N
        'Наименование Предмета'                        , # [2] =>
Subj_Nm
        'Факультет (отделение)'                        , # [3] =>
Faculty
        'Четверть'                                      , # [4] =>
Semestr
        'Идентификатор Преподавателя'                  , # [5] =>
Teach_Id
        'Количество Часов'                              , # [6] =>
Subj_Hours
        'Максимальный Балл'                            , # [7] =>
Subj_Points
        'Примечания'                                    , # [8] =>
Subj_Com
    ),

    'teachers' => Array                                # Данные
заносятся админом при идентификации #

```

```

(
    'Логин' , # [0] =>
Teach_Id
    'Пароль' , # [1] =>
Teach_Comm
    'ФИО' , # [2] =>
Teach_FIO
    'Предметы' , # [3] =>
Teach_Chair
    'Должность' , # [4] =>
Teach_Posit
    'Ученая Степень' , # [5] =>
Teach_Degree
    'Телефон' , # [6] =>
Teach_Phone
    'Электронный Адрес' , # [7] =>
Teach_Email
),

```

```

    'tests' => Array # Результаты
всех тестов всех экзаменуемых #
    ( # Можно
только просматривать при авторизации #
        'Номер' , # [0] =>
Test_Id
        'Наименование' , # [1] =>
Test_Nm
        'Предмет' , # [2] =>
Stud_Id
        'Количество вопросов' , # [3] =>
Quest_Id
        'Количество ответов' , # [4] =>
Repl_Id
        'Количество Очков' , # [5] =>
Stud_Points
        'Примечания' # [6] =>
Test_Com
    )

```

```

);
#
#
# This is my advanced idea for the next version:
# Should be loop for the $Table_Headers_DB like this
# $Table_Headers_DB ($FormTypes[$i], headers_DB[$i])
# and headers_DB[$i] - two dim array
#
# This is not correct array construction !!!
# PHP5 print_r print in this view (without "" and ",")
#
#$A=array
#(

```

```
# [0]=>array
# (
# [0]='a0',
# [1]='b0'
# ),
# [1]=>array
# (
# [0]='a1',
# [1]='b1'
# )
#);
?>
```

Файл «Test-Select.php»

```
<?php
#####
#####
# Модуль выдает форму для занесения наименования предмета,
номера теста, номера #
# ответа - определяет данные для первичного входа в предмет
тестирования #
#####
#####
#
include_once 'Header.php'; // Общий заголовок
веб-страниц проекта
$Stud_Id = $_GET['username']; // значение после
авторизации студента

// Далее в select

Ajax
include_once 'DB_Tables_Fields.php'; // Все таблицы и
поля БД
include_once 'DB_Class.php'; // Класс работы с
БД - открытие, запросы
include_once 'Header.php'; // Общий заголовок
веб-страниц проекта
#
### Все вопросы по текущему тесту, по центру BAR номеров
вопросов ###
#
$QueryResult = $dbObj->query("SELECT * FROM subjects");
//print '<pre>';
//print_r($QueryResult);
//print '</pre>';
?>
<p></p>
<p></p>
<p></p>
<center><p><font color= "Blue" size="3"><strong>Выберете предмет
и номер теста для начала тестирования</strong></font>
```

```

</p></center>
<p></p>
<form action="Forms_Main_Repl.php" method=post>
<table align="center" width="60%" bgcolor="#999999"
cellspacing="1" cellpadding="1" border=0>
  <tr>
    <td width='130' align="center" bgcolor="#FFCCCC">
      &nbsp; Выберите наименование предмета
    </td>
    <td width='120' align="center" bgcolor="#FFCCCC">
      <p>
        <select id="Subj" name="Subj_Id"
onchange="main_func();">
          <?php
            $i=0;
            foreach ($QueryResult as $row => $Arr_acc){
              $Subj_N[$i] = $QueryResult[$i]['Subj_N'];
              $Subj_Name[$i] = $QueryResult[$i]['Subj_Name'];
              print "<option value =
'$Subj_N[$i]'>$Subj_Name[$i]</option>";
              $i++;
            }
          ?>
        <!--
          <option value="1">Библиотеки динамической
загрузки</option>
          <option value="2">Сетевые приложения
        </option>
        //-->
      </select>
    </p>
    </td>
  </tr>
  <tr>
    <td width='130' align='center' bgcolor="#FFFFFF" >
      &nbsp; Выберите номер теста
    </td>
    <td width='120' align='center' bgcolor="#FFFFFF" >
      <p>
        <select id="Test" name="Test_Id"
onchange="main_func();">
          <option value="1"> Тест 1</option>
        </select>
      </p>
    </td>
  </tr>
  <!--
  <tr>
    <td width='130' align='center' bgcolor="#FFFFFF" >
      &nbsp; Выберите номер вопроса
    </td>
    <td width='120' align='center' bgcolor="#FFFFFF" >

```

```

        <p>
                <select
onchange="main_func();"
                id="Test"
                name="Quest_N"
                <option value="1">1</option>
                <option value="2">2</option>
                <option value="3">3</option>
                <option value="4">4</option>
                <option value="5">5</option>
                <option value="6">6</option>
                <option value="7">7</option>
                </select>
        </p>
    </td>
</tr>
//-->
    <tr>
        <td
            colspan="2"
            width='120'
            align='center'
            bgcolor="#FFFFFF" >
                <p></p>
                <font color= "Red" size="5"><strong><?php print
$msg1; ?></strong></font>
                <br>
            </td>
        </tr>
        <tr>
            <td colspan="2" width='130' align='center' bgcolor="#FFFFFF"
>
                <INPUT TYPE='submit' NAME='Add' STYLE='font-size: 10 pt;
                background: font-family: arial,verdana,helvetica,sansserif'
                VALUE=' Начать тестирование ' >
            </td>
        </tr>
    </table>
<!--***** начинаем с первого вопроса
*****/-->
    <input type='hidden' name ='Stud_Id' value='<?php print
$Stud_Id; ?>'>
    <input type='hidden' name ='Quest_N' value='1'>
</form>
<p></p>
<p></p>
</body>
</html>

```

Файл «Reply-Handler.php»

```

<?php
#####
#####

```

```

# Модуль обрабатывает ответы по тестам - номер теста, определяет
ответ и готовит #
# данные для занесения в базу данных
#
#####
#####
#
$FormType      = 'replies';                                // Это
таблица БД с ответами
$Arr_acc       = array();
$Row           = array();
$dbtable       = $FormType;
#
include_once   'DB_Tables_Fields.php';                      //
определитель таблиц и полей БД на лету
include_once   'Header.php';                                // Общий
заголовок веб-страниц проекта
include_once   'Table_Headers.php';                          // заголовки
столбцов таблиц на экране
#
# Так мы делаем софт более независимым от конкретных таблиц БД
#
$headers       = $Table_Headers [$FormType];                //
наименования столбцов таблицы на экране
$headers_db    = $Table_Headers_DB[$FormType];              //
наименования доменов в таблице БД

#                                                            // БД и
<input name's в формах
#
# Repl_N - это номер правильного ответа в БД, Repl_U - номер
# ответа экзаменуемого
#
# Определяем все входные данные, переданные по методу POST для
БД Replies
#                                                            //
$headers_db это имена доменов в таблице БД
if ($_POST["Add"]||$_POST["UpdPro"])
{                                                            //
    $$headers_db[$i] - значение $i-го поля
        $i=0;
        foreach ($headers_db as $element)
        {
            ${$headers_db[$i]} = $_POST[$headers_db[$i]]; // На
хостинге должно быть так, с {}
#         print '<br>'.$headers_db[$i].'='.${$headers_db[$i]};
            $i++;
        }
    }
if ($Quest_N==1){

```

```

$QueryResult = $dbObj->query("DELETE FROM replies WHERE
Subj_Id='$Subj_Id' AND Test_Id='$Test_Id' AND Stud_Id='$Stud_Id'
");
}
#
# определяем правильность ответа
#
$Quest_N = $_POST['Quest_N'];
$Repl_Correct = $_POST['Repl_Correct'];
$Repl_U = $_POST['Repl_U'];
$Repl_Point_Max = $Repl_Point;
$Repl_Correct = trim ($Repl_Correct);
$Repl_U = trim ($Repl_U);
If ($Repl_Correct!=$Repl_U)
{
    $Repl_Point=0;
}
${$headers_db[7]} = $Repl_U; // На хостинге
должно быть так
${$headers_db[8]} = $Repl_Correct;
${$headers_db[9]} = $Repl_Point;
#
# Занесение в базу данных 'replies'
# Формируем строку запроса для добавления в БД
#
$i=0;
foreach ($headers_db as $element)
{
    $Str2.=","."${$headers_db[$i]}".""; // На хостинге
должно быть так !!!
    $i++;
}
$Str2 = substr($Str2,1); // Убираем первую
запятую
include 'Request_SQL.php';
if ($QueryResult==1) $msg1 = 'Ваш ответ занесен в базу
данных';
else {
    $msg1 = "Произошла ошибка! Данные уже есть в БД! ";
    include 'Update_SQL.php';
    $msg1 = 'Ваш ответ успешно обновлен';
}
$Str2='';
$Quest_N++; // Переход к
следующему вопросу
# // без перескока

//print '<pre>';
//print_r($headers_db);
//print '</pre>';
#
?>

```

[illegible]

```

        <p></p>
        <font color= "Red" size="5"><strong><?=$msg1
?></strong></font>
        <br>
    </td>
</tr>
<tr>
    <td colspan="2" width='130' align='center' bgcolor="#FFFFFF"
>
        <INPUT TYPE='submit' NAME='Add' STYLE='font-size: 10 pt;
        background: font-family: arial,verdana,helvetica,sansserif'
VALUE=' Следующий вопрос ' >
    </td>
</tr>
</table>
<!--
*****
*****/-->
        <input type='hidden' name = 'Repl_Id'
value='<?=$Repl_Id ?>'>
        <input type='hidden' name = 'Subj_Id'
value='<?=$Subj_Id ?>'>
        <input type='hidden' name = 'Subj_Nm'
value='<?=$Subj_Nm ?>'>
        <input type='hidden' name = 'Test_Id'
value='<?=$Test_Id ?>'>
        <input type='hidden' name = 'Stud_Id'
value='<?=$Stud_Id ?>'>
        <input type='hidden' name = 'Quest_N'
value='<?=$Quest_N ?>'>
        <input type='hidden' name = 'Quest_Text'
value='<?=$Quest_Text?>'>
<!--***** Это то что в БД
*****/-->
        <input type='hidden' name = 'Repl_N'
value='<?=$Repl_N ?>'>
        <input type='hidden' name = 'Repl_Point'
value='<?=$Repl_Point?>'>
<!--***** Это то что в вычислено
*****/-->
        <input type='hidden' name = 'Repl_Correct'
value='<?=$Repl_Correct?>'>
</form>

```

Файл «Tables-Handler.php»

```

<?php
$Table_Headers =
Array
(

```

```

        'groups' => Array                                # Данные
заполняет преподаватель #
        (
            'Группа (Класс)'                                ,      # [0] =>
Group_Id
            'Наименование'                                ,      # [1] =>
Group_Name
            'Факультет'                                    ,      # [2] =>
Faculty
            'Ориентация '                                  ,      # [3] =>
Speciality
            'Руководитель'                                  ,      # [4] =>
Chief_FIO
            'Примечания'                                    ,      # [5] =>
Group_Com
        ),

        'questions' => Array                                # Данные
заполняет преподаватель #
        (
            'Код вопроса'                                    ,      # [0] =>
Quest_Id
            '№ вопроса'                                      ,      # [1] =>
Quest_N
            '№ предмета'                                    ,      # [2] =>
Subj_Id
            'Предмет'                                        ,      # [3] =>
Subj_Nm
            'Тест'                                          ,      # [4] =>
Test_Id
            'Наименование вопроса'                          ,      # [5] =>
Quest_Text
            'Первый ответ'                                  ,      # [6] =>
Repl_1
            'Второй ответ'                                  ,      # [7] =>
Repl_1
            'Третий ответ'                                  ,      # [8] =>
Repl_1
            'Четвертый ответ'                              ,      # [9] =>
Repl_1
            '№ верного'                                      ,      # [10]=>
Repl_N
            'Очки'                                          ,      # [11]=>
Repl_Point
        ),

        'replies' => Array                                # Данные
берутся из БД Questions #
        (                                                    # Вносится
номер правильного ответа #

```

```

Repl_Id      'Код ответа'      ,      # [0] =>
Subj_Id      '№ предмета'      ,      # [1] =>
Subj_Nm      'Предмет'         ,      # [2] =>
Test_Id      'Тест'            ,      # [3] =>
Stud_Id      'Код учащегося'   ,      # [4] =>
Quest_N      '№ вопроса'       ,      # [5] =>
Quest_Text   'Наименование вопроса' ,      # [6] =>
Repl_U       'Ответ тестируемого' ,      # [7] =>
Repl         'Правильный ответ'   ,      # [8] =>
Repl_Point   'Очки'            ,      # [9] =>
),

```

```

'students' => Array      # Данные
заносятся студентом #
(

```

```

Stud_Id      'Код учащегося'   ,      # [0] =>
Stud_Pass    'Пароль'          ,      # [1] =>
Stud_FIO     'ФИО'             ,      # [2] =>
Stud_Faculty 'Факультет'       ,      # [3] =>
Stud_Group   'Группа'          ,      # [4] =>
Stud_Year    'Год Поступления' ,      # [5] =>
Stud_Spec_N   'Специализация'  ,      # [6] =>
Stud_Spec_Nm  'Наименование'   ,      # [7] =>
Stud_Phone    'Телефон'        ,      # [8] =>
Stud_Email    'Электронный адрес' ,      # [9] =>
),

```

```

'subjects' => Array      # Данные
заносятся пре-лем #
(

```

```

Subj_Id      'Идентификатор'   ,      # [0] =>

```

```

Subj_N      'Номер или Код' , # [1] =>
Subj_Nm     'Наименование Предмета' , # [2] =>
Faculty     'Факультет (отделение)' , # [3] =>
Semestr     'Четверть' , # [4] =>
Teach_Id    'Идентификатор Преподавателя' , # [5] =>
Subj_Hours  'Количество Часов' , # [6] =>
Subj_Points 'Максимальный Балл' , # [7] =>
Subj_Com    'Примечания' , # [8] =>
),

'teachers' => Array # Данные
заносятся админом при идентификации #
(
Teach_Id    'Логин' , # [0] =>
Teach_Comm  'Пароль' , # [1] =>
Teach_FIO   'ФИО' , # [2] =>
Teach_Chair 'Предметы' , # [3] =>
Teach_Posit 'Должность' , # [4] =>
Teach_Degree 'Ученая Степень' , # [5] =>
Teach_Phone 'Телефон' , # [6] =>
Teach_Email 'Электронный Адрес' , # [7] =>
),

'tests' => Array # Результаты
всех тестов всех экзаменуемых #
( # Можно
только просматривать при авторизации #
Test_Id     'Номер' , # [0] =>
Test_Nm     'Наименование' , # [1] =>
Test_Points 'Предмет' , # [2] =>
Stud_Id     'Количество вопросов' , # [3] =>
Quest_Id

```

```

        'Количество ответов' , # [4] =>
Repl_Id
        'Количество Очков' , # [5] =>
Stud_Points
        'Примечания' # [6] =>
Test_Com
    )

);
#
#
# This is my advanced idea for the next version:
# Should be loop for the $Table_Headers_DB like this
# $Table_Headers_DB ($FormTypes[$i], headers_DB[$i])
# and headers_DB[$i] - two dim array
#
# This is not correct array construction !!!
# PHP5 print_r print in this view (without "" and ", "
#
#$A=array
#(
# [0]=>array
# (
# [0]=>'a0',
# [1]=>'b0'
# ),
# [1]=>array
# (
# [0]=>'a1',
# [1]=>'b1'
# )
#);
?>

```

Файл «Universal-Forms-Builder.php»

```

<?php
#
# построитель форм такого вида Universal_Forms_Builder.php
#
### Кнопки тоже вписываются в таблицы; Особенность в
NAME='UpdPro'. Первый раз
# пользователь нажимает кнопку Update, под списком всех строк
таблицы,
# например, сотрудников. Предполагается обновление только одной
записи.

```

```

# В этой форме происходит повторное нажатие "изменить", а
# обрабатывает
# один и тот же _Handler. Он "должен знать" откуда запрос на
# обработку.
#
$button_add = "<INPUT TYPE='submit' NAME='Add'
STYLE='font-size: 10 pt;
background: font-family: arial,verdana,helvetica,sansserif'
VALUE=' Добавить '>";
$button_del = "<INPUT TYPE='submit' NAME='Delete'
STYLE='font-size: 10 pt;
background: font-family: arial,verdana,helvetica,sansserif'
VALUE='Удалить '>";
$button_upd = "<INPUT TYPE='submit' NAME='UpdPro'
STYLE='font-size: 10 pt;
background: font-family: arial,verdana,helvetica,sansserif'
VALUE=' Изменить '>";
$button_view = "<INPUT TYPE='submit' NAME='View'
STYLE='font-size: 10 pt;
background: font-family: arial,verdana,helvetica,sansserif'
VALUE='Просмотр '>";
###
# Создаем объекты класса для вывода таблиц, вызываем
# конструктор
#$handler = $FormType.'_Handler'; // обработчик
# настоящей формы
#$headers = $Table_Headers [$FormType]; // наименования
# столбцов таблице
#$headers_db = $Table_Headers_DB[$FormType]; // наименования
# доменов в таблице // Staff БД и
input name's в формах
$i=0;
foreach ($headers as $element)
{
    $element1 = "<input type='text' name='".
    $headers_db[$i].
    "'size=140 maxlength=250
value='".$$headers_db[$i]."'>";
    // наименования
    доменов в таблице
    # // Staff БД и
    input name's в формах
    $Form_view -> AddRowAssArr
    (array($form_headers [0] =>
    $element,
    $form_headers [1] =>
    $element1
    )
    );
    $i++;
}

```

```

//buttons Add && Delete
$Form_view -> AddRowAssArr
    (array($form_headers [0] => $button_add,
           $form_headers [1] => $button_del
    )
    );
//buttons Update && View
$Form_view -> AddRowAssArr
    (array($form_headers [0] => $button_upd ,
           $form_headers [1] => $button_view
    )
    );
// вывод формы на экран
$Form_view -> OutForm($handler);
?>

```

Приложение В

(обязательное)

Обучающее Web-приложение для изучения нейронных сетей

Презентация

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		117

Обучающее Web-прилож изучения нейронн

ы: студент группы 21ВВП2
А.
ководитель: Д.Т.Н., профессор кафедры

се разработки должны быть решены следующие задачи:

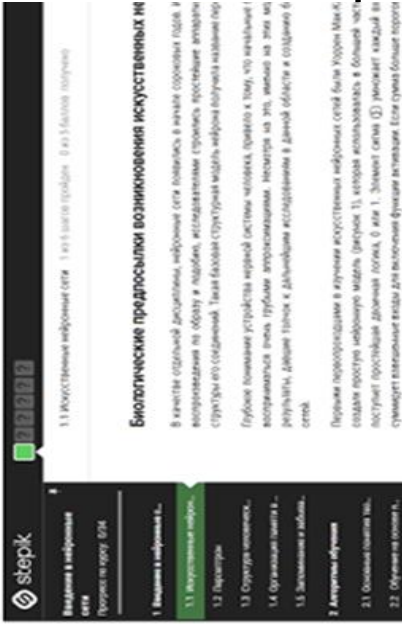
отанные с определенными соглашениями тесты должны
овать достоверную проверку знаний посредством исполъс
, работа которой основана на этих соглашениях;
«ность создания достоверных и адаптивных оценок результ
зания;
та создания тестов, удобство интерфейса для несложного с
ного изменения;
твие зависимости программного обеспечения системы от
енной области знаний или отдельных предметов, а также
ости от определенных аппаратных или программных возмож
ющих организаций;
«ность одновременного прохождения тестирований по разн
ам, тестам (темам) неограниченного количества пользова

Аналог

- Образовательная платформа «[Stepik](#)»
- Образовательная платформа «Открытое образование»
- Различные платформы для тестирования: [GOOGLE форм](#), [Quizlet](#), [Proprofs](#), [Kahoot!](#), [ClassMarker](#), [Plickers](#), [Easy Maker](#)

Изм.	Лист	№ докум.	Подпись	Дата

Образовательная платформа «Stepik»



азовательная платформа «Открытое образование»

Опрос 1

☐ Добавить страницу в закладки

Опрос 1

1 point possible (ungraded)

К какому проявлению психоэмоционального состояния относятся:

☐ Мышечный тонус
 ☐ Мышечная жесткость
 ☐ Голосовые характеристики
 ☐ Сердечно-сосудистый тонус
 ☐ Эмоциональные характеристики

Отправить

Программы

Тренировки

Сотрудничество

О проекте

Поиск

Мой профиль

Профиль

Мои данные

Пароль

Активизация

Аналитика

Сообщество

Сессии

Платформа

Мои данные

Имя пользователя

Имя

Поздравляем! Ваше имя и фамилия, они будут вписаны в сертификат о прохождении курса

Фамилия *

Имя *

Поздравляем!

.et



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.45.001 ПЗ

Лист
123

Стек технологий

PHP



Система управления базами данных

аммирования



Note

Среда разг

Apache
сервер



изработки графического интерфейса были

использованы:

. (HyperText Markup Language)
Cascading Style Sheets)
JavaScript

Изм.	Лист	№ докум.	Подпись	Дата

Основные php модули программы (скрипты):

in — Вход администратора
 iber — Вход студента
 :h — Поиск по ключу
 исание концепции
 d — Добавление записи
 d-Htpass — Шифрование пароля
 — Вывод таблиц
 .HTML-New — Представление таблиц
 - Работа с БД
 Данные соединения
 -Fields — Сканирование таблиц
 earch — Поиск по ключу

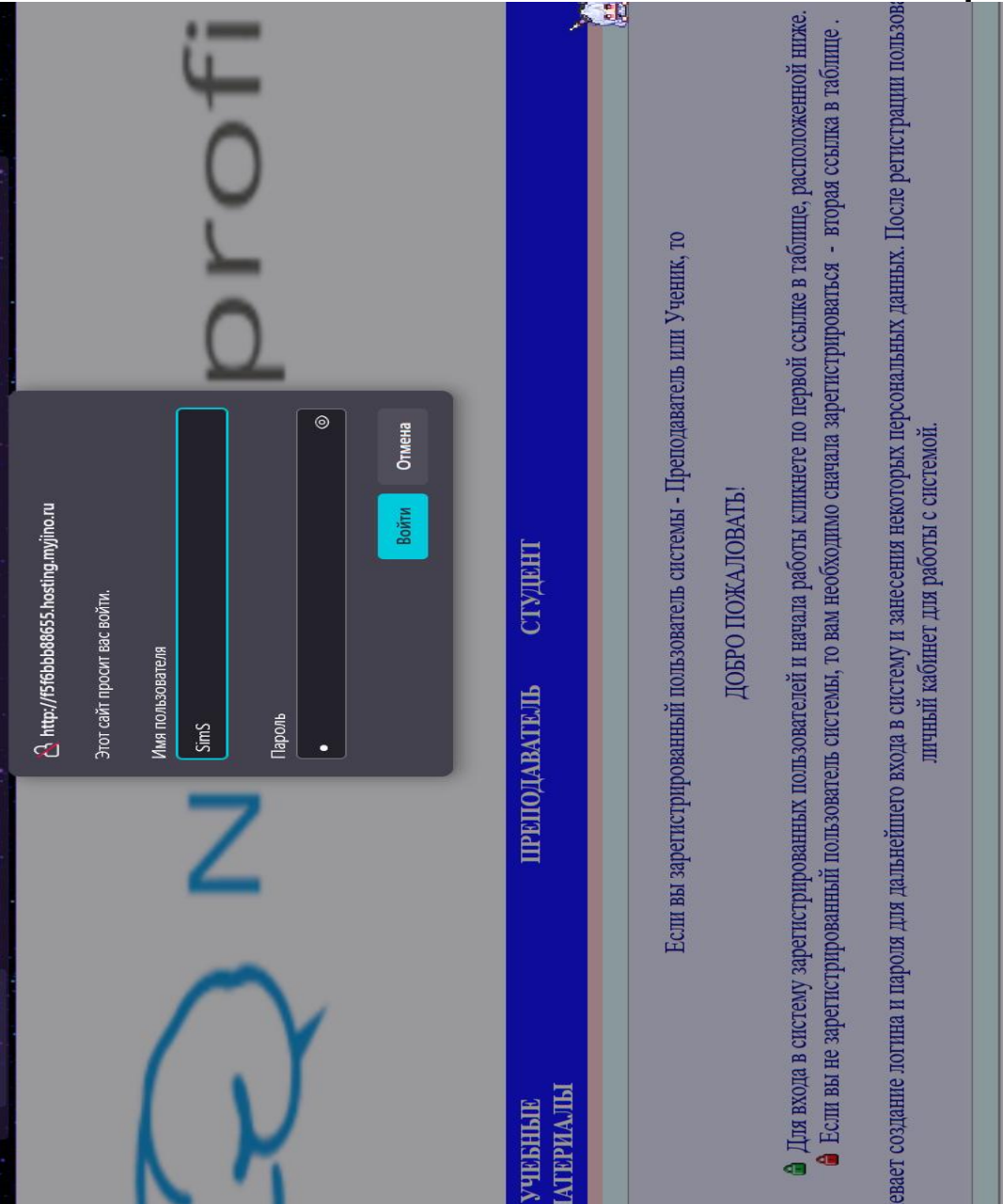
Form-Buttons — Стандартные
 Forms-Main — Построение фор
 Header — Шапка страниц
 index — Начальная страница
 Reg-Frame — Модуль регистра
 rejection.htm — Ошибка автори
 Request-SQL — Запрос на доба
 Table-Headers — Заголовки таб
 Test-Select — Выбор теста
 Reply-Handler — Обработка оте
 View-Table — Просмотр данных
 Tables-Handler — Обработка та
 Universal-Forms-Builder — Пос

Главная страница приложения



УЧЕБНЫЕ МАТЕРИАЛЫ ПРЕПОДАВАТЕЛЬ СТУДЕНТ

проекта является создание платформы, которая предоставляет студентам возможность проходить тесты по теме нейросетей. Проект включает регулярные помогут углубить понимание материала, а также рекомендации по подходящей литературе для самостоятельного изучения. Преподаватели смогут тесты, адаптированные под уровень подготовки студентов, что позволит более эффективно оценивать их знания и способствовать лучшему усвоению материала на интернет-технологиях и клиент-серверной архитектуре





ЧЕБНЫЕ ТЕРИАЛЫ	ПРЕПОДАВАТЕЛЬ	СТУДЕНТ
Лекция 1. Введение в нейронные сети Смотреть лекцию на Rutube		
<p>Список книг:</p> <ol style="list-style-type: none"> 1. Глубокое обучение - Иэн Гудфеллоу, Йошуа Бенджио, Аарон Курвилль 2. Нейронные сети и глубокое обучение - Майкл Нильсен 		

Neuroprofi

УЧЕБНЫЕ
МАТЕРИАЛЫ

ПРЕПОДАВАТЕЛЬ СТУДЕНТ



КАБИНЕТ ТЕСТИРУЕМОГО

Добро пожаловать

SimS!



ПЕРЕХОД К ТЕСТИРОВАНИЮ



Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.45.001 ПЗ

Лист

130

Neuroprofi

УЧЕБНЫЕ МАТЕРИАЛЫ

ПРЕПОДАВАТЕЛЬ

СТУДЕНТ

Выберите наименование предмета

Выберите наименование предмета

Основы Нейронных Сетей

Выберите ПРЕДМЕТ

Изм.	Лист	№ докум.	Подпись	Дата

ПГУ1.090301.45.001 ПЗ

Лист

131



УЧЕБНЫЕ МАТЕРИАЛЫ	ПРЕПОДАВАТЕЛЬ	СТУДЕНТ
<div>1 2 3 4 </div> <p>ВОПРОС: Что такое нейронная сеть?</p>		
<p>Перечень возможных ответов</p> <p>а). Модель, имитирующая работу человеческого мозга для решения задач.</p> <p>б). Компьютерная программа, которая выполняет арифметические операции.</p> <p>в). Устройство для хранения данных</p>		<p>Отметьте правильный ответ</p> <p>☒</p> <p>☐</p> <p>☐</p>



УЧЕБНЫЕ
МАТЕРИАЛЫ

ПРЕПОДАВАТЕЛЬ СТУДЕНТ

| 1 | 2 | 3 | 4 |

**ВЫ ЗАВЕРШИЛИ ТЕСТИРОВАНИЕ ПО ПРЕДМЕТУ
КОЛИЧЕСТВО БАЛЛОВ: 0 ИЗ 4 ВОЗМОЖНЫХ
ВАША ОЦЕНКА НЕ УДОВЛЕТВОРИТЕЛЬНО**



ПОЛНЫЙ СПИСОК ВОПРОСОВ И ОТВЕТОВ

Neuroprofi

УЧЕБНЫЕ
МАТЕРИАЛЫ

ПРЕПОДАВАТЕЛЬ СТУДЕНТ

КАБИНЕТ ПРЕПОДАВАТЕЛЯ

Добро пожаловать
Sim!



ВСЕ РЕЗУЛЬТАТЫ ТЕСТИРОВАНИЯ



ДОБАВЛЕНИЕ И ИЗМЕНЕНИЕ ВОПРОСОВ

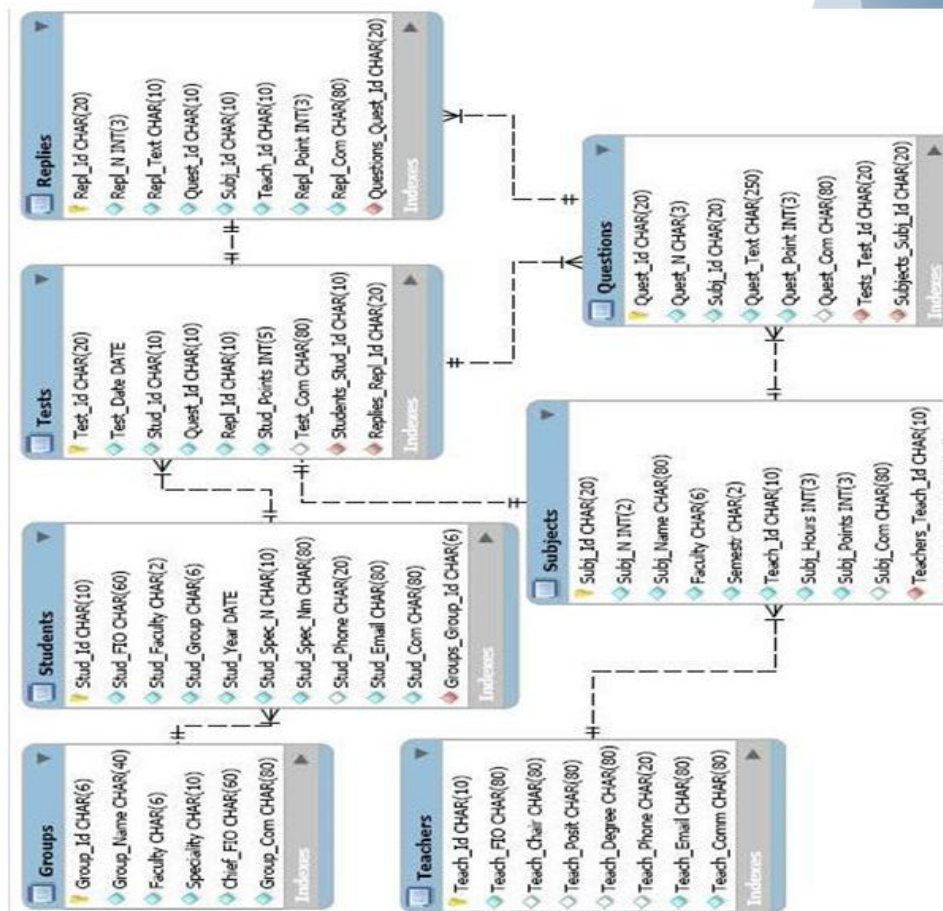


УЧЕБНЫЕ МАТЕРИАЛЫ		ПРЕПОДАВАТЕЛЬ		СТУДЕНТ		
Предмет	Тест	Код учащегося	№ вопроса	Наименование вопроса	Ответ тестируемого	Правильн
Введенный курс	1	Sim	1	Что такое переобучение (overfitting) в контексте нейронных сетей?	2) Когда модель слишком работает на обучающих данных, но плохо на тестовых.	2) Когда модель слишком работает на обучающ тест
Введенный курс	1	Sim	2	Что такое сверточные нейронные сети (CNN)?	4) Нейронная сеть, использующая только один слой.	2) Тип нейронно разработанный для об
Введенный курс	1	SimS	1	Что такое нейронная сеть?	а). Модель, имитирующая работу человеческого мозга для решения задач.	а). Модель, имити
Введенный курс	1	Sim	1	Что такое нейронная сеть?	а). Модель, имитирующая работу	а). Модель, имити



ЧЕБНЫЕ ТЕРИАЛЫ		ПРЕПОДАВАТЕЛЬ	СТУДЕНТ
Заносимые или обновляемые данные			
111Sim			
1			
Основы Нейронных Сетей			
1			
Sim			
1			
Что такое нейронная сеть?			
а). Модель, имитирующая работу человеческого мозга для решения задач.			

Схема БД



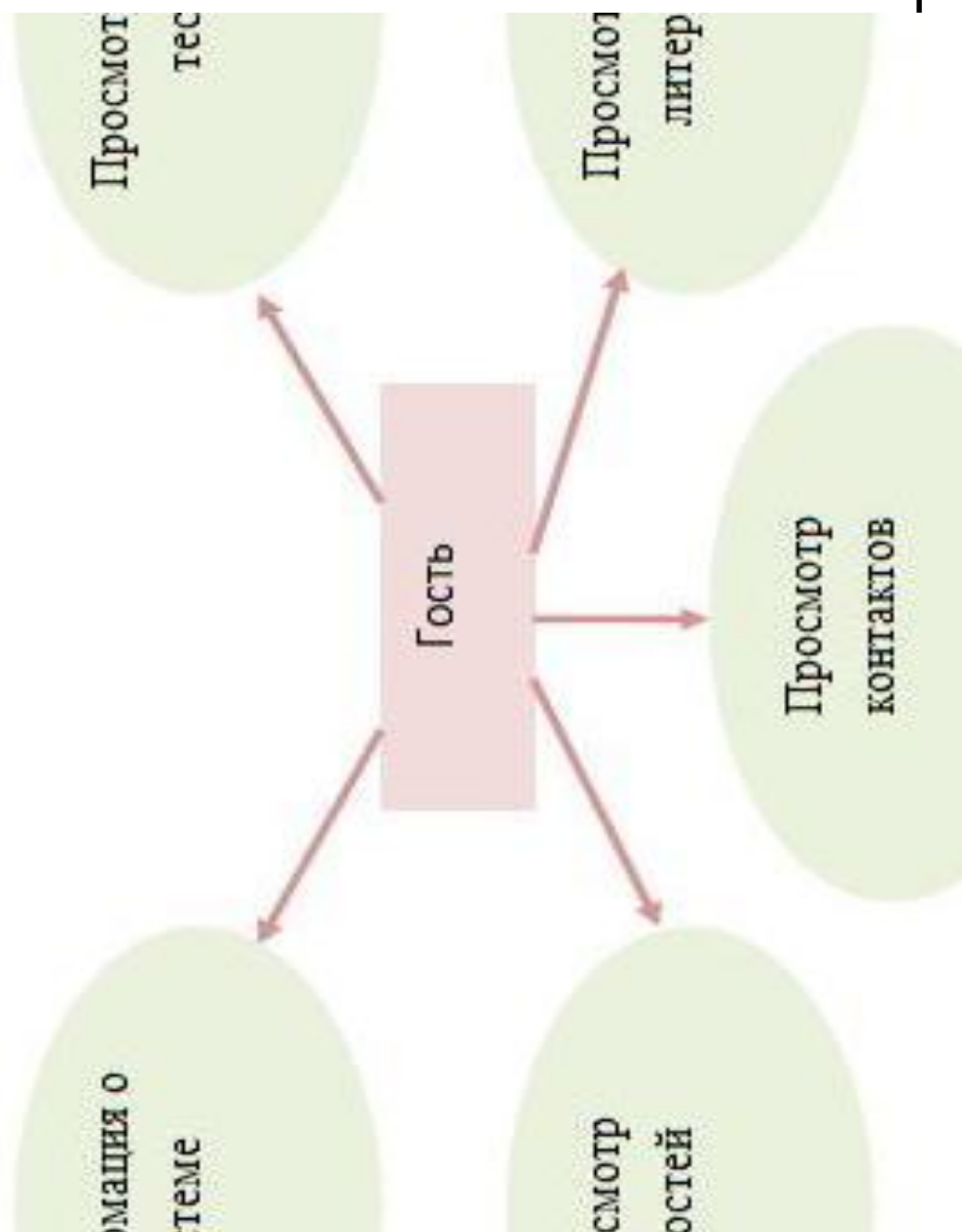


Диаграмма вариантов использования студентом

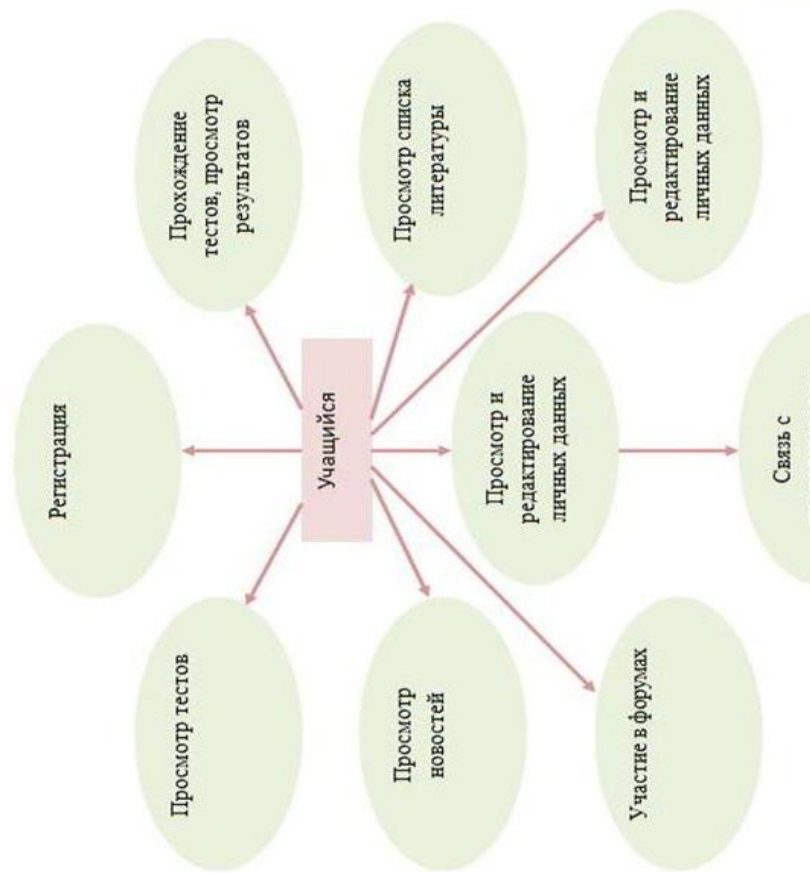


Диаграмма вариантов использования для преподавателя

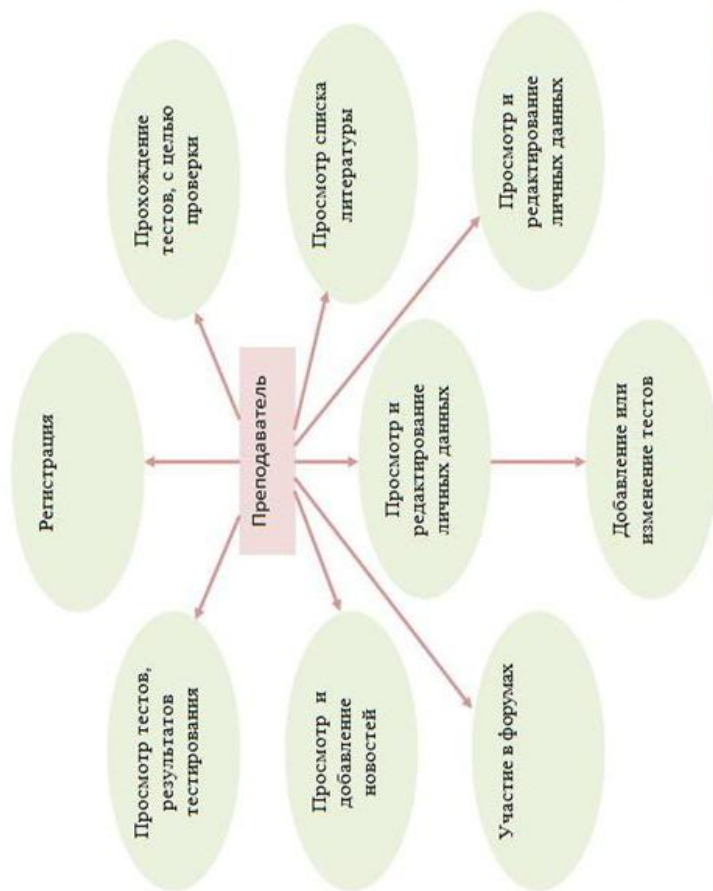
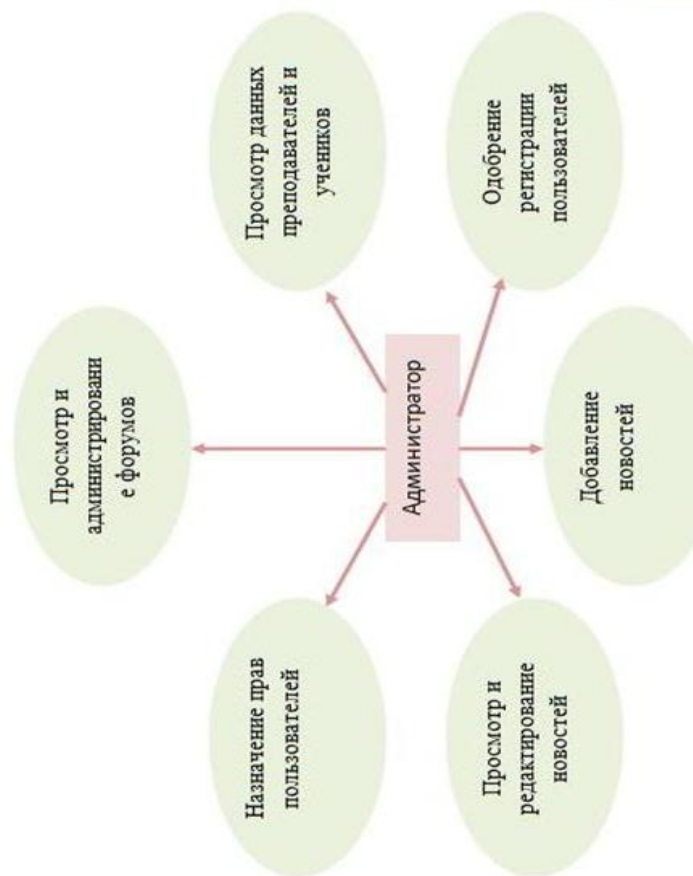


Диаграмма вариантов использования для администратора



- новое Web – приложение способно давать и проверять полученные ответы;
- можно усовершенствовать разработанное приложение, добавив:
 - удобный функционал управления администратором;
 - удобный личный кабинет;
 - ввести систему авторизации и регистрации;
 - ввести лекции, улучшить оформление сайта;

ПАСИБО ЗА ВНИМАНИЕ

					ПГУ1.090301.45.001 ПЗ	Лист
Изм.	Лист	№ докум.	Подпись	Дата		144