СОДЕРЖАНИЕ

СОДЕРЖАНИЕ..... 1

ВВЕДЕНИЕ..... 2

1 ПОСТАНОВОЧНАЯ ЧАСТЬ..... 5

1.1 Формулировка задачи..... 5

1.2 Описание входной и выходной информации..... 5

1.3 Требования к интерфейсу приложения..... 6

2 ПРОЕКТНАЯ ЧАСТЬ..... 9

2.1 Структура базы данных..... 9

2.2 Организация интерфейса пользователя..... 25

2.3 Разработка форм приложения, меню, отчетов..... 30

3 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ..... 33

3.1 Краткая характеристика веб-сервера Apache..... 33

3.2 Используемые языки программирования..... 35

3.3 Выбор базы данных..... 37

4 ЭКСПЛУАТАЦИЯ..... 39

4.1 Требования к аппаратному обеспечению..... 39

4.2 Инструкция пользователя..... 41

5 ЭКОНОМИЧЕСКАЯ ЧАСТЬ..... 43

6 МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ОКРУЖАЮЩЕЙ СРЕДЫ..... 52

6.1 Общие принципы и законодательство по охране труда..... 52

6.2 Пожарная безопасность при работе на персональном компьютере..... 53

6.3 Мероприятия по охране окружающей среды..... 55

ЗАКЛЮЧЕНИЕ..... 57

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.....60

Приложение А. Листинг страницы index.php.....61

Приложение Б. Листинг страницы login.php.....64

Приложение В. Листинг файла menu.php.....68

ВВЕДЕНИЕ

В настоящее время все большее и большее количество информации окружает нас – все это буквально опутывает человека паутиной информации. Растущие потоки, объемы и скорость поступления, генерирования, хранения и обработки информации обязывают использовать электронные средства – персональные компьютеры.

Связанные между собой компьютеры в Американской локальной сети Arpanet родили международную сеть под названием Internet. Международная сеть сейчас объединяет практически все маломальски «умные» smart-вещи: смартфоны, часы, телевизоры, холодильники и даже лампочки.

Интернет внедрился в нашу жизнь и стал полноправным ее участником. В процессе серфинга по страничкам в сети, мы посещаем разные веб-сайты. Это и торговые площадки, это и социальные сети, это и ресурсы с новостями. Есть и личные странички людей и компаний. Таковы реалии современной жизни – чтобы стать успешным, нужно в первую очередь заявить о себе в международном сообществе интернета. Получается, персональная страничка должна присутствовать у каждой организации. Однако простой сайт-визитка не всегда приносит максимум пользы. Высшие учебные заведения для

По всему миру постоянно функционируют миллионы учебных заведений, начиная дошкольными учреждениями образования и заканчивая высшими учебными заведениями. Ежедневно сотни миллионов людей обучаются либо в другом качестве участвуют в образовательном процессе – все они нуждаются в оперативной информации. Обеспечение свежей актуальной информацией – одна из потребностей современных технологий обучения. Самый простой способ – создание и использование специализированных информационных систем, что и реализовано на данный момент во многих организациях. Однако имеются весомые проблемы: как правило, пусть даже это электронные документы, но это оффлайн-версии, не

имеющие возможностей одновременного многопользовательского доступа, а также не являются адаптированными к различным операционным средам. Совершенно другое дело – информационная система в виде веб-портала, доступ к которому возможен из-под любой операционной системы, оснащенной браузером, а также к веб-порталу может подключаться одновременно несколько человек и выполнять свои функции параллельно, не мешая работе друг друга. Причем доступ можно получить к порталу учебного заведения, находясь дома, в дороге, в командировке – везде, где есть доступ к сети Интернет.

Эта проблема не является новой и поэтому на рынке программного обеспечения уже существует ряд продуктов, так или иначе пытающихся устранить коммуникационный провал. За решение этой задачи взялись как небольшие частные фирмы, так и организации типа банков и университетов.

В данный момент на рынке существует большое количество компаний, предоставляющих сервис электронных дневников. Каждая из них обладает своими достоинствами и недостатками.

Так, например, на рынке имеются масштабные решения, такие как dnevnik.ru, своим функционалом повторяющие полноценные социальные сети. Пользователям помимо основной функциональности электронного дневника предлагается также сервис хранения файлов, фотографий, видеозаписей, возможность объединения людей в сообщества, создание различных событий и прочие, полезные на первый взгляд, возможности. Но родителям трудно освоить весь предлагаемый им функционал, что приводит к путанице и полному непониманию программного продукта. Преподавателям же разобраться в системе еще сложнее. Поэтому одной из задач при реализации электронного дневника было обеспечение максимальной простоты для работы в системе.

Большинство проектов имеют бизнес-модель, предполагающую значительные траты со стороны образовательного учреждения. Например, проект 1dnevnik.ru берет за одно лишь подключение к системе 50000 руб.

Некоторые проекты готовы предоставлять свои услуги с помесячной оплатой и бесплатным подключением.

Ведущими электронными дневниками на сегодняшний день являются 1dnevnik.ru [2], dnevnik.ru [3], ballov.net [4]. Кроме приведенных выше примеров существует множество других, менее известных и распространённых систем.

В основном, эти ресурсы предназначены для школ, а для высших учебных заведений, где несколько иная специфика, каждому ВУЗу приходится действовать самостоятельно и разрабатывать свою информационную систему.

Объект нашего исследования – информационная система для ВУЗа.

Результатом работы представляется обеспечить возможность пользователю в любой точке мира доступ к информации об успеваемости, посещаемости и расписании - для студента, о нагрузке и расписанию - для преподавателей.

Целью работы является проектирование информационная система для ВУЗа.

Таким образом, указаны актуальность данной дипломной работы, а также предмет и объект исследования, сформулирована основная цель дипломного проекта.

1 ПОСТАНОВОЧНАЯ ЧАСТЬ

1.1 Формулировка задачи

Целью данной дипломной работы является реализация веб-сервиса «Электронный дневник» с изучением сопутствующих его реализации обеспечению бесперебойного высоконагруженного функционирования технологий.

Таким образом, необходимо:

- изучить необходимый минимум для реализации

Качественного веб-сервиса:

- стандарты веб-разработки;
- шаблоны проектирования;

- изучить способы ускорения высоконагруженных веб-приложений;
- выбрать конечную концепцию электронного дневника;
- реализовать веб-сервис «Электронный Дневник».

1.2 Описание входной и выходной информации

Входная информация:

1) списки кафедр,

2) списки преподавателей,

3) списки учебных групп,

4) списки студентов каждой учебной группы,

5) список типов занятий (лекции, лабораторные занятия, семинары, экзамены, зачеты, практические занятия и прочее),

6) учебное расписание каждой учебной группы,

7) список типов отметок (обычная, зачет, семинар, курсовой проект, экзамен),

8) список возможных отметок (пропуск, отметки от 1 до 10, «зачет-незачет», неявка на экзамен/зачет),

9) список времени начала и окончания учебных пар,

10) списки родителей студентов,

11) списки старост, кураторов групп,

12) отметки студентов по предметам.

Выходная информация:

1) учебное расписание в разрезе учебных недель,

2) нагрузка преподавателей,

3) нагрузка студентов,

4) отчет по успеваемости студентов,

5) отчет по посещаемости студентов.

1.3 Требования к интерфейсу приложения

Основные требования к интерфейсу перечислены в таблице 1.

Таблица 1 – Требования к интерфейсу

Описание	Приоритет	Сценарии юзабилити-тестирования
Интерфейс должен быть спроектирован с учетом экранных разрешений режимов отображения планшета: ни для одного сценария работы с приложением кнопки доступа к основным функциям не должны оказаться вне видимой области окна; для размещенных вне видимой области элементов управления должна быть предусмотрена возможность прокрутки; для всех допускающих прокрутку областей сам факт возможности прокрутки и метод прокрутки должен быть интуитивно понятен не менее чем 95% пользователей	Высокий	Сценарии юзабилити-тестирования
Кнопки доступа к основным функциям приложения должны быть спроектированы с учетом антропометрической совместимости: пространственная компоновка форм и размерные характеристики кнопок должны обеспечивать быстрый доступ к функциям приложения большими пальцами рук без стилуса и клавиатуры	Высокий	Сценарии юзабилити-тестирования
Должна быть предусмотрена возможность подключения и смены тем оформления интерфейса пользователя, включающих цветовую гамму, значки	Низкий	Сценарии настройки интерфейса
Должна быть предусмотрена возможность индивидуальной настройки функционала кнопок (назначения кнопкам функций из допустимого набора)	Низкий	Сценарии настройки интерфейса

Главное требование – наличие самой информации на сайте и насколько удобно, оперативно, доступно и современно она публикуется на сайте. Это также во многом зависит от системы управления сайтами и её свойств, которые это обеспечивают:

- Невысокие требования к квалификации пользователей

- Визуальное управление шаблонами (макетами)
- Неограниченный объем публикуемой информации
- Версионность хранения всех данных
- Богатство и удобство функционала для управления данными
- Неделимость функционала на модули, принцип "все включено"

2 ПРОЕКТНАЯ ЧАСТЬ

2.1 Структура базы данных

Схема базы данных разработана в программе Mysql Workbench в виде EER-диаграммы (рис.1).

У каждой таблицы задан первичный ключ (как правило, id), благодаря которому можно однозначно идентифицировать любую запись таблицы.

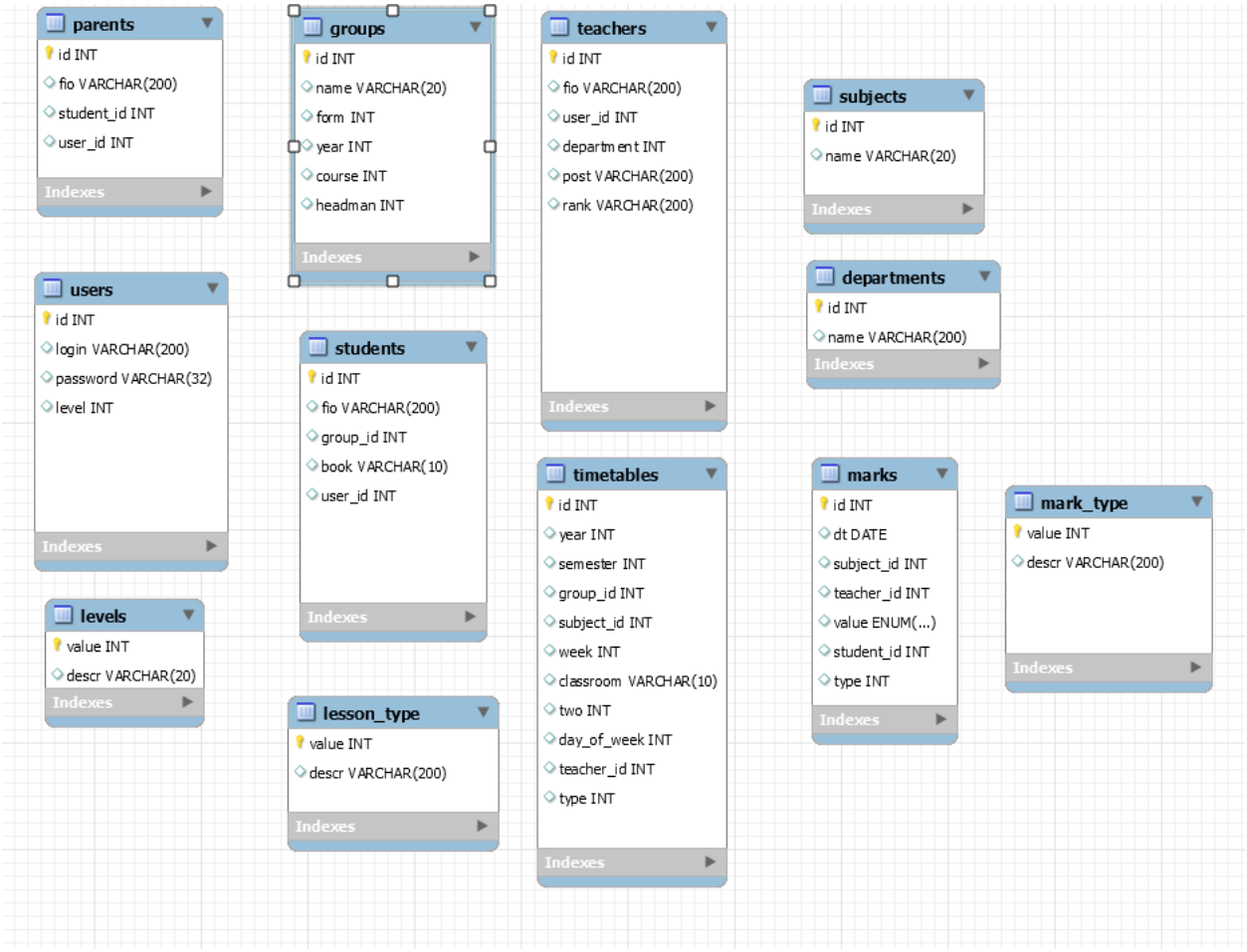

Рис. 1 – Схема базы данных

На рис.2 приведена структура таблицы departments (кафедры). Назначение и смысл полей таблицы понятен из столбца «комментарии».

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	*Нет*	код кафедры	AUTO_INCREMENT
name	varchar(200)	utf8_general_ci		Да	*NULL*	шифр группы	
abbr	varchar(20)	utf8_general_ci		Нет		сокращенное название	

Рис. 2 – Структура таблицы Departments

На рис. 3 приведен пример содержимого таблицы departments. Всего в таблице будет достаточно много строк.

id код кафедры	name шифр группы	abbr сокращенное название
1	Радиоэлектронной техники	РЭТ
2	Белорусского и русского языков	БиРЯ
3	Интеллектуальных информационных технологий	ИИТ
4	Вычислительных методов и программирования	ВМиП
5	Теоретических основ электротехники	ТОЭ
6	Систем управления	СУ
7	Гуманитарных дисциплин	ГД
8	Информационных технологий автоматизированных систе...	ИТАС
9	Экономики	Экономики

Рис. 3 – Содержание таблицы departments

На рис.4 приведена структура таблицы form_type (формы обучения). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии
1	value	int(11)			Нет	*Нет*	значение
2	descr	varchar(200)	utf8_general_ci		Да	*NULL*	описание

Рис. 4 – Структура таблицы form_type

На рис. 5 приведен пример содержимого таблицы form_type.

value значение	descr описание
1	дневная
2	заочная
3	дистанционная
4	вечерняя

Рис. 5 – Содержание таблицы form_type

На рис.6 приведена структура таблицы groups (группы). Назначение и смысл полей таблицы понятен из столбца «комментарии».

Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
id	int(11)			Нет	*Нет*	код группы	AUTO_INCREMENT
name	varchar(20)	utf8_general_ci		Да	*NULL*	шифр группы	
form	int(11)			Да	*NULL*		
course	int(11)			Да	*NULL*		
headman	int(11)			Да	*NULL*		

Рис. 6 – Структура таблицы groups

Headman – староста группы. Он выбирается и назначается в программе позже. Course – номер курса группы. Form – id формы обучения из таблицы form_type.

На рис. 7 приведен пример содержимого таблицы groups.

id код группы	name шифр группы	form	course	headman
1	810101	1	2	0
2	780961	1	3	0
3	810102	1	2	0
4	858641	1	2	0
5	858642	1	2	0
6	858401	1	2	0
7	962991	1	1	0
8	846541	1	2	0

Рис. 7 – Содержание таблицы groups

На рис.8 приведена структура таблицы lesson_time (время занятий). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Д
1	value	int(11)			Нет	*Нет*	значение	
2	start_time	varchar(10)	utf8_general_ci		Да	*NULL*	начало пары	
3	end_time	varchar(20)	utf8_general_ci		Нет	*Нет*	конец пары	
4	all_time	varchar(20)	utf8_general_ci		Нет	*Нет*	время пары	

Рис. 8 – Структура таблицы lesson_time

Headman – староста группы. Он выбирается и назначается в программе позже. Course – номер курса группы. Form – id формы обучения из таблицы form_type.

На рис. 9 приведено содержимое таблицы lesson_time.

value значение	start_time начало пары	end_time конец пары	all_time время пары
1	8:00	9:35	8:00-9:35
2	9:45	11:20	9:45-11:20
3	11:40	13:15	11:40-13:15
4	13:25	15:00	13:25-15:00
5	15:20	16:55	15:20-16:55
6	17:05	18:40	17:05-18:40
7	18:45	20:20	18:45-20:20

Рис. 9 – Содержание таблицы lesson_time

На рис.10 приведена структура таблицы lesson_type (типы занятий). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии
1	value	int(11)			Нет	*Нет*	значение
2	descr	varchar(200)	utf8_general_ci		Да	*NULL*	описание
3	abbr	varchar(20)	utf8_general_ci		Нет	*Нет*	сокращенное название

Рис. 10 – Структура таблицы lesson_ type

Поле abbr будет удобно использовать при отображении расписания.

На рис. 11 приведено содержимое таблицы lesson_ type.

value значение	descr описание	abbr сокращенное название
1	лекция	ЛК
2	лабораторные занятия	ЛР
3	семинар	сем
4	экзамен	экз
5	зачет	зач
6	практические занятия	ПЗ

Рис. 11 – Содержание таблицы lesson_ type

На рис.12 приведена структура таблицы levels (уровни доступа). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии
1	value	int(11)			Нет	*Нет*	значение
2	descr	varchar(20)	utf8_general_ci		Да	*NULL*	10-админ, 6-препод, 5-куратор, 4-староста, 3-студент, 2-родитель, 1-гость

Рис. 12 – Структура таблицы levels

Данная таблица содержит описания различных видов пользователей. В зависимости от принадлежности к определенному виду пользователь будет иметь свой функционал в программе.

На рис. 13 приведено содержимое таблицы levels .

value значение	descr 10-админ, 6-препод, 5-куратор, 4-староста, 3-студе...
1	гость
2	родитель
3	студент
6	преподаватель
10	админ

Рис. 13 – Содержание таблицы levels

На рис.14 приведена структура таблицы log (лог). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	login	varchar(200)	utf8_general_ci		Нет	*Нет*	логин юзера	
3	oper	varchar(20)	utf8_general_ci		Нет	*Нет*	операция	
4	ip	varchar(15)	utf8_general_ci		Нет	*Нет*	ip	
5	dt	timestamp		on update CURRENT_TIMESTAMP	Нет	CURRENT_TIMESTAMP	время операции	ON UPDATE CURRENT_TIMESTAMP

Рис. 14 – Структура таблицы log

Данная таблица может использоваться для сохранения информации о действиях пользователей (например, вход-выход из программы, изменение информации).

На рис. 15 приведено содержимое таблицы log .

id	login логин юзера	oper операция	ip ip	dt время операции
1	admin	login	::1	2019-11-09 10:42:21
2	admin	login	::1	2019-11-09 10:59:54
3	admin	login	::1	2019-11-09 11:00:59
4	student	login	::1	2019-11-09 11:01:34
5	student	login	::1	2019-11-09 11:09:03
6	admin	login	::1	2019-11-09 11:09:24
7	admin	login	::1	2019-11-09 11:10:30
8	admin	login	::1	2019-11-09 11:11:10
9	admin	login	::1	2019-11-09 11:24:26

Рис. 15 – Содержание таблицы log

На рис.16 приведена структура таблицы marks (отметки). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	dt	date			Да	*NULL*	дата отметки	
3	subject_id	int(11)			Да	*NULL*	предмет	
4	teacher_id	int(11)			Да	*NULL*	кто поставил (препод)	
5	value	enum('n', '1', '2', '3', '4', '5', '6', '7', '8', ...	utf8_general_ci		Да	*NULL*	n-пропуск, 1-10 - отметки, z-зачтено,nz-не зачтено	
6	student_id	int(11)			Да	*NULL*	кому эта оценка	
7	type	int(11)			Да	*NULL*	id типа занятия (экзамен, зачет, обычная оценка и т.д.)	

Рис. 16 – Структура таблицы marks

На рис. 17 приведено содержимое таблицы marks .

id	dt дата отметки	subject_id предмет	teacher_id кто поставил (препод)	value n-пропуск, 1-10 - отметки, z-зачтено,nz-не зачтено	student_id кому эта оценка	type id типа занятия (экзамен, зачет, обычная оценка и ...
7	0000-00-00	33	4	1	2	1
8	0000-00-00	33	4	z	2	2
9	2019-11-08	33	4	7	2	1
10	2019-11-13	33	1	4	2	1
11	2019-11-03	33	1	n	2	5
12	2019-11-05	33	1	n	2	3

Рис. 17 – Содержание таблицы marks

На рис.18 приведена структура таблицы mark_type (типы отметок). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Д
1	value	int(11)			Нет	*Нет*	значение	
2	descr	varchar(200)	utf8_general_ci		Да	*NULL*	описание	

Рис. 18 – Структура таблицы mark_type

На рис. 19 приведено содержимое таблицы mark_type .

value значение	descr описание
1	обычная
2	зачет
3	семинар
4	курсовой проект
5	экзамен

Рис. 19 – Содержание таблицы mark_type

На рис.20 приведена структура таблицы parents (родители). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	**id** 🔑	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	**fio**	varchar(200)	utf8_general_ci		Да	*NULL*	ФИО родителя	
3	**student_id**	int(11)			Да	*NULL*	чей родитель	
4	**user_id**	int(11)			Да	*NULL*		

Рис. 20 – Структура таблицы parents

User_id – id пользователя из таблицы users.

На рис. 21 приведено содержимое таблицы parents .

id	fio ФИО родителя	student_id чей родитель	user_id
1	Фролова Мария Олеговна	1	3

Рис. 21 – Содержание таблицы parents

На рис.22 приведена структура таблицы students (студенты). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	**id**	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	**fio**	varchar(200)	utf8_general_ci		Да	*NULL*	ФИО родителя	
3	**student_id**	int(11)			Да	*NULL*	чей родитель	
4	**user_id**	int(11)			Да	*NULL*		

Рис. 22 – Структура таблицы students

User_id – id пользователя из таблицы users.

На рис. 23 приведен пример содержимого таблицы students .

id	fio ФИО студента	group_id группа	book номер зачетной книжки	user_id
1	Фролов Фрол Геннадьевич	413	ав232144	2
2	Гурина Ольга Олеговна	413	ав3424324	2
3	Сомов Олег Романович	413	34234234	5
4	Ракова Ирина Ириновна	2	3536548	6

Рис. 23 – Содержание таблицы students

На рис.24 приведена структура таблицы subjects (учебные предметы). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	*Нет*	код предмета	AUTO_INCREMENT
2	name	varchar(20)	utf8_general_ci		Да	*NULL*	наименование предмета	

Рис. 24 – Структура таблицы subjects

На рис. 25 приведен пример содержимого таблицы subjects.

id код предмета	name наименование предмета
1	Эк
2	КС
3	ФизК
4	Матем
5	ЭП
6	КЧ
7	ООП
8	АЭСБ
9	БЖЧ
10	МСиСвИнфТ

Рис. 25 – Содержание таблицы subjects

На рис.26 приведена структура таблицы teachers (преподаватели). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	fio	varchar(200)	utf8_general_ci		Да	*NULL*	ФИО препода	
3	user_id	int(11)			Да	*NULL*	id пользователя	
4	department	int(11)			Да	*NULL*	кафедра	
5	post	varchar(200)	utf8_general_ci		Да	*NULL*	должность	
6	rank	varchar(200)	utf8_general_ci		Да	*NULL*		

Рис. 26 – Структура таблицы teachers

Rank – ученое звание.

User_id – id пользователя из таблицы users.

На рис. 27 приведен пример содержимого таблицы teachers.

id	fio ФИО препода	user_id id пользователя	department кафедра	post должность
1	Абрамов Игорь Иванович	4	27	Професс
2	Абрамович Михаил Семенович	*NULL*	20	Доцент
3	Аваков Сергей Мирзоевич	*NULL*	14	Професс
4	Авецюк Александр Игоревич	*NULL*	25	
5	Аврамец Дмитрий Владимирович	*NULL*	20	
6	Азаров Илья Сергеевич	*NULL*	18	Доцент
7	Аксенов Валерий Васильевич	*NULL*	22	Доцент
8	Аксенов Вячеслав Анатольевич	*NULL*	31	
9	Аксенчик Анатолий Владимирович	*NULL*	4	Професс
10	Албут Анастасия Анатольевна	*NULL*	37	
11	Алексеев Виктор Федорович	*NULL*	38	Доцент
12	Алексеев Игорь Геннадиевич	*NULL*	20	
13	Алексеев Юрий Игоревич	*NULL*	25	
14	Алефиренко Виктор Михайлович	*NULL*	38	Доцент

Рис. 27 – Содержание таблицы teachers

На рис.28 приведена структура таблицы timetables (расписание). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	id	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	year	int(11)			Да	*NULL*	учебный год	
3	semester	int(11)			Да	*NULL*	семестр (1, 2, 3)	
4	group_id	int(11)			Да	*NULL*	группа	
5	subject_id	int(11)			Да	*NULL*	предмет	
6	week	int(11)			Да	*NULL*	номер недели (1,2,3,4)	
7	classroom	varchar(10)	utf8_general_ci		Да	*NULL*	аудитория	
8	two	int(11)			Да	*NULL*	номер пары (для времени)	
9	day_of_week	int(11)			Да	*NULL*	день недели (1-7)	
10	teacher_id	int(11)			Да	*NULL*	препод	
11	type	int(11)			Да	*NULL*	код типа занятий (лк, лр и т.п.)	

Рис. 28 – Структура таблицы timetables

На рис. 29 приведен пример содержимого таблицы timetables.

id	year учебный год	semester семестр (1, 2, 3)	group_id группа	subject_id предмет	week номер недели (1,2,3,4)	classroom аудитория	two номер пары (для времени)	day_of_week день недели (1-7)	teacher_id препод	type код типа занятий (лк, лр и т.п.)
1	2019	1	1	1	3	607-2	0	0	119	6
2	2019	1	1	2	0	114-3	0	0	94	1
3	2019	1	1	2	1	114-3	0	0	94	1
4	2019	1	1	2	2	114-3	0	0	94	1
5	2019	1	1	2	3	114-3	0	0	94	1
6	2019	1	1	2	4	114-3	0	0	94	1
7	2019	1	1	3	0		3	0	0	6
8	2019	1	1	3	1		3	0	0	6
9	2019	1	1	3	2		3	0	0	6
10	2019	1	1	3	3		3	0	0	6
11	2019	1	1	3	4		3	0	0	6
12	2019	1	1	4	0	112-3	4	0	745	1
13	2019	1	1	4	1	112-3	4	0	745	1

Рис. 29 – Содержание таблицы timetables

На рис.30 приведена структура таблицы users (пользователи). Назначение и смысл полей таблицы понятен из столбца «комментарии».

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Комментарии	Дополнительно
1	**id**	int(11)			Нет	*Нет*		AUTO_INCREMENT
2	**login**	varchar(200)	utf8_general_ci		Да	*NULL*	логин	
3	**password**	varchar(32)	utf8_general_ci		Да	*NULL*	md5-хэш пароля	
4	**level**	int(11)			Да	2	уровень доступа =2	

Рис. 30 – Структура таблицы users

На рис. 31 приведен пример содержимого таблицы users.

id	login логин	password md5-хэш пароля	level уровень доступа =2
1	admin	pass10	10
2	student	pass3	3
3	parent	pass2	2
4	teacher	teacher6	6
5	student2	pass3	3
6	student3	pass3	3

Рис. 31 – Содержание таблицы users

2.2 Организация интерфейса пользователя

При входе пользователь увидит окно авторизации с запросом логина и пароля. Логин и пароль выдается администратором.

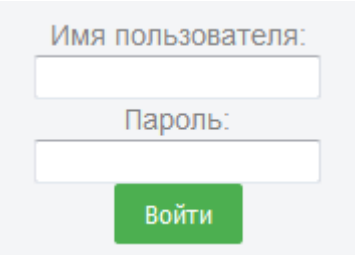

Рис. 32 – Окно авторизации

В тестовом режиме на данный момент (на момент разработки программы) окно авторизации позволяет легко войти под правами администратора (верный логин и пароль уже введены) (рис. 33). Пароль администратора тестовый: pass1.

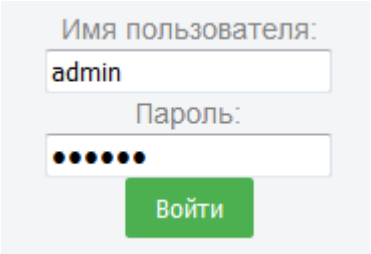

Рис. 33 – Окно авторизации в тестовом режиме

После авторизации с правами администратора доступно максимальное количество функций (рис.34).

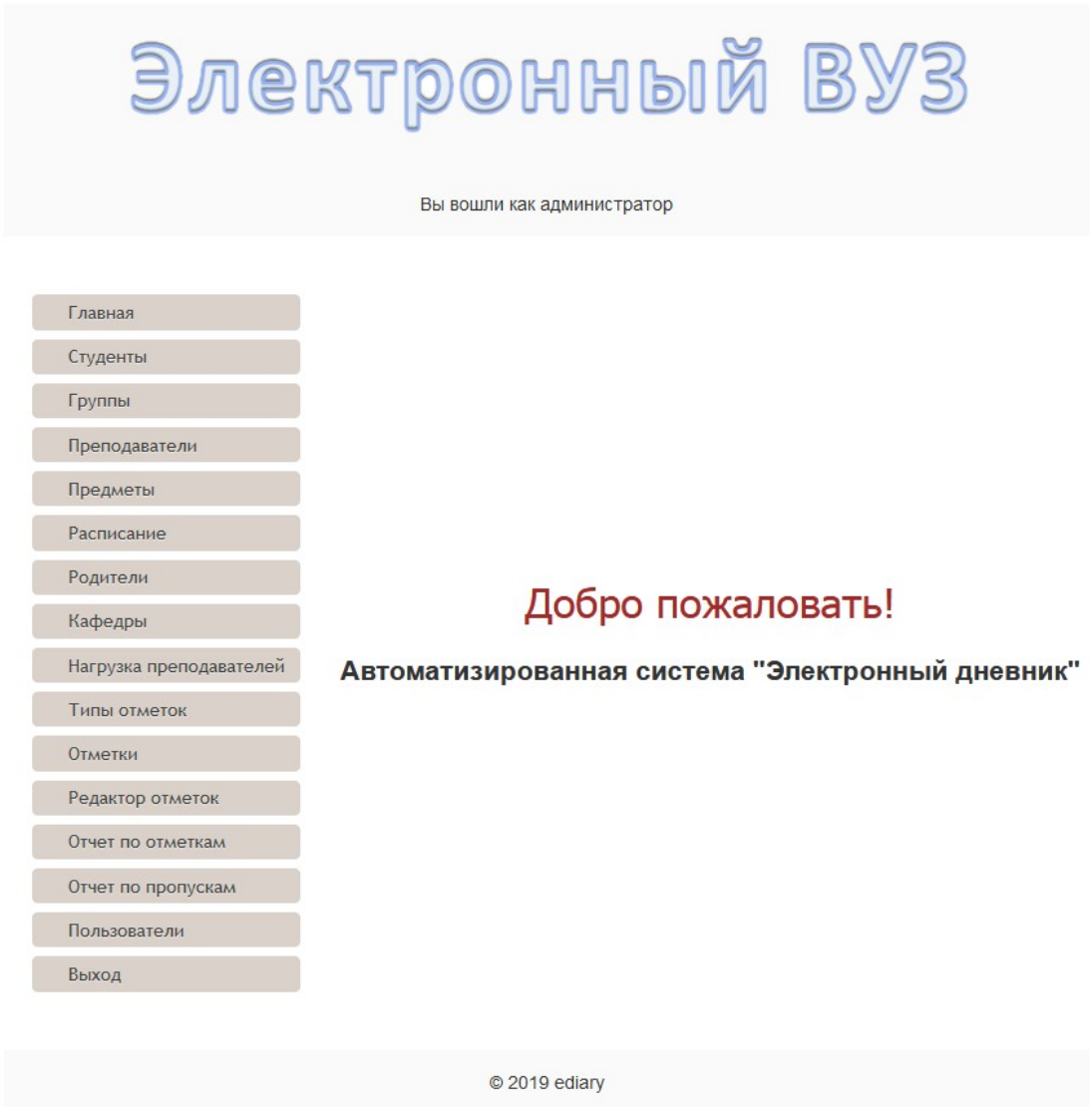

Рис. 34 – Главное окно с правами администратора

После авторизации с правами студента/родителя доступно ограниченное количество функций (рис.35).

Тестовый вход под студентом: логин – student, пароль – pass3.

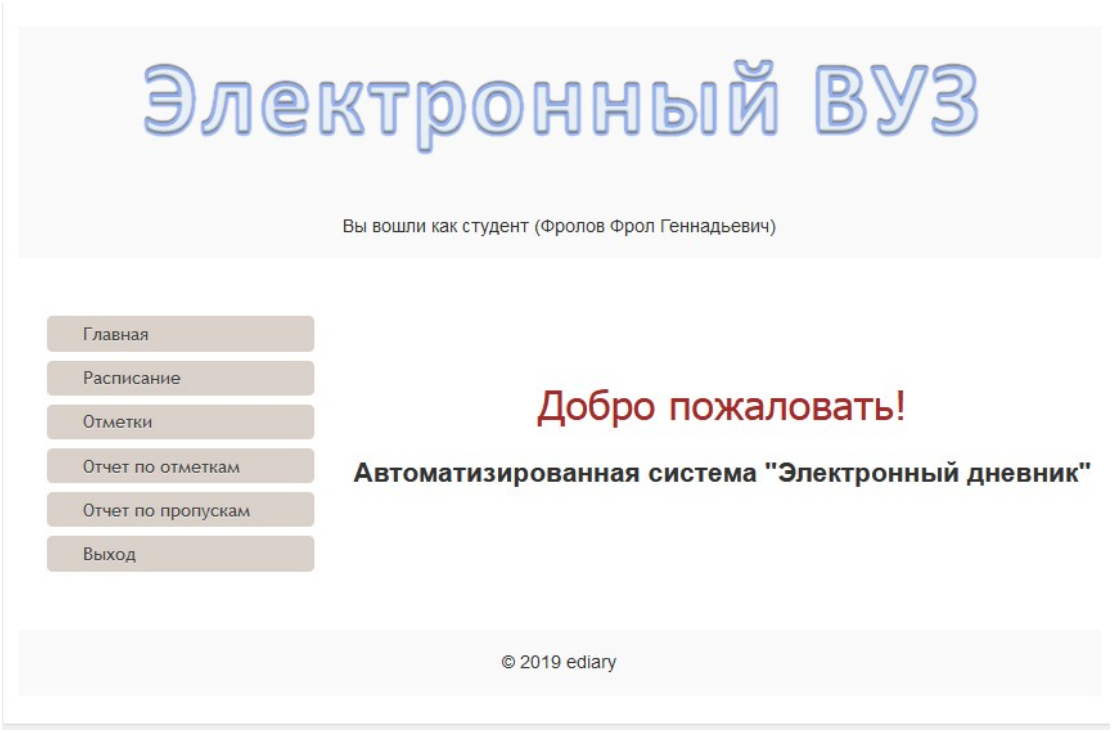

Рис. 35 – Главное окно с правами студента/родителя

После авторизации с правами преподавателя доступно ограниченное количество функций (рис.36).

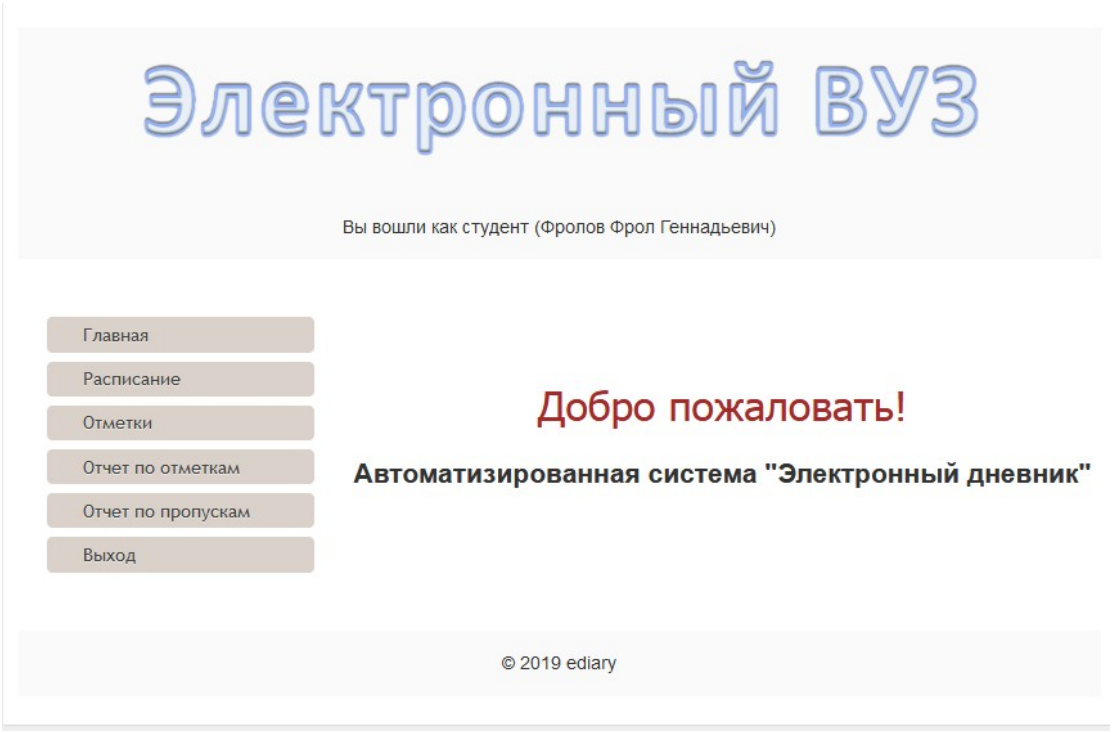

Рис. 36 – Главное окно с правами преподавателя

После авторизации с правами преподавателя доступно ограниченное количество функций (рис.37).

Тестовый вход под студентом: логин – teacher, пароль – pass6.

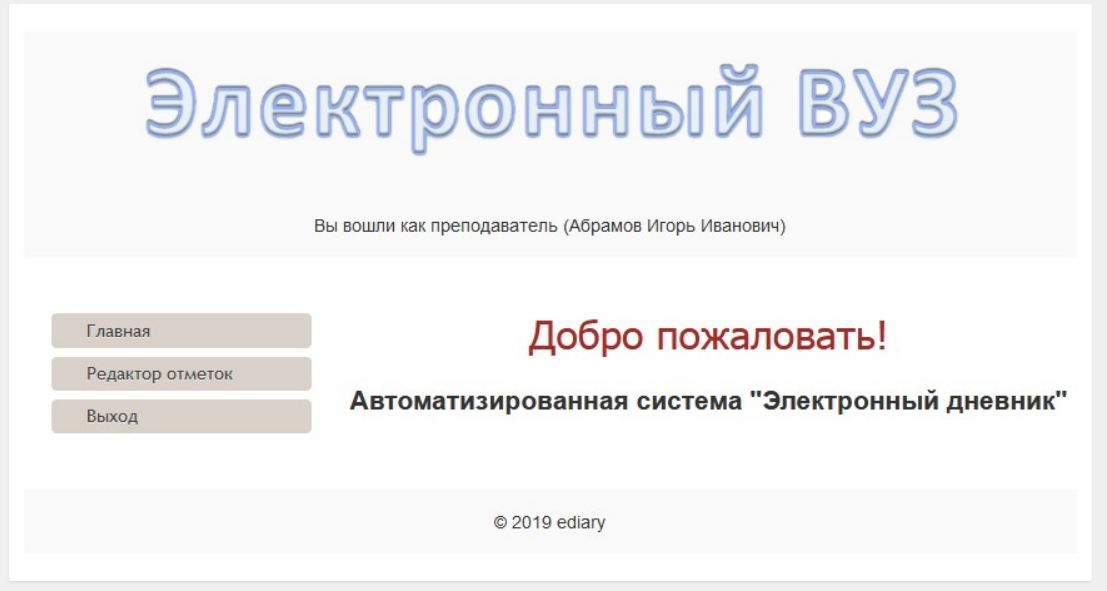

Рис. 37 – Главное окно с правами преподавателя

2.3 Разработка форм приложения, меню, отчетов

Преподаватель может выставлять оценки в разделе «Редактор отметок» (рис. 38).

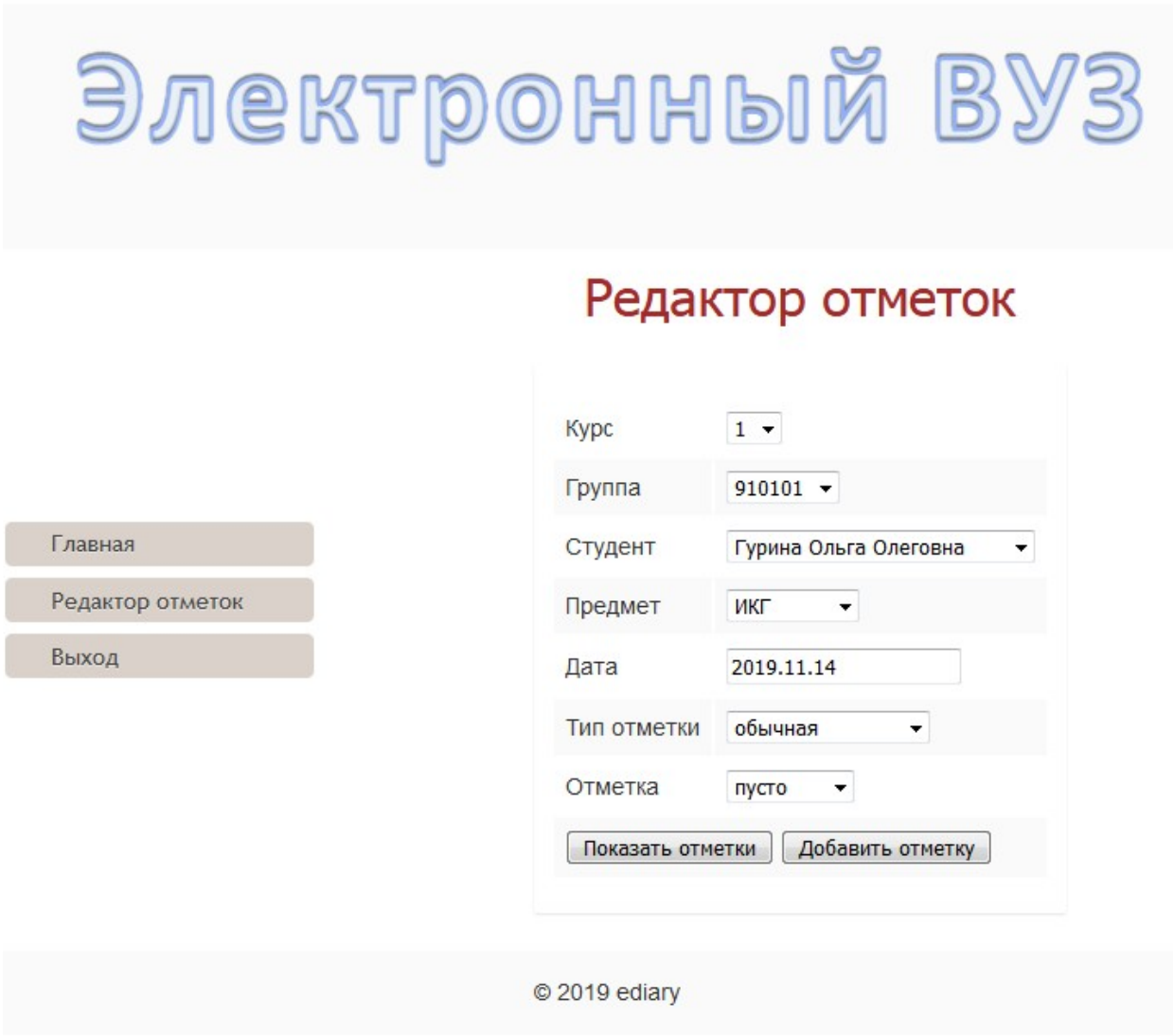

Рис. 38 – Редактор отметок

Чтобы выставить отметку необходимо выбрать курс. При этом автоматически будет подгружен список групп, принадлежащих выбранному курсу. При выборе нужной группы автоматически будет подгружен список студентов данной группы и список предметов данной группы. Т.е. исключены некорректные выборы.

Для простоты разработки функция удаления оценки за нужную дату по нужному студенту и нужному предмету реализована выбором отметки «пусто» и нажатие на кнопку «Добавить отметку».

При нажатии на кнопку «Показать отметки» показывается ведомость отметок по данному студенту (рис. 39).

Отметки по предмету ИКГ студента группы 910101 Гурина Ольга Олеговна

Дата	Преподаватель	Отметка	Тип отметки
0000.00.00	Авецюк Александр Игоревич	1	обычная
0000.00.00	Авецюк Александр Игоревич	z	зачет
2019.11.03	Абрамов Игорь Иванович	n	экзамен
2019.11.05	Абрамов Игорь Иванович	n	семинар
2019.11.08	Авецюк Александр Игоревич	7	обычная
2019.11.13	Абрамов Игорь Иванович	4	обычная

Рис. 39 – Ведомость отметок

Студенты и их родители могут просмотреть свое расписание (рис.40). При этом нет необходимости выбирать группу. Т.к. группа уже известна (по логину). Необходимо лишь выбрать номер учебной недели (если расписание меняется и зависит от недели).

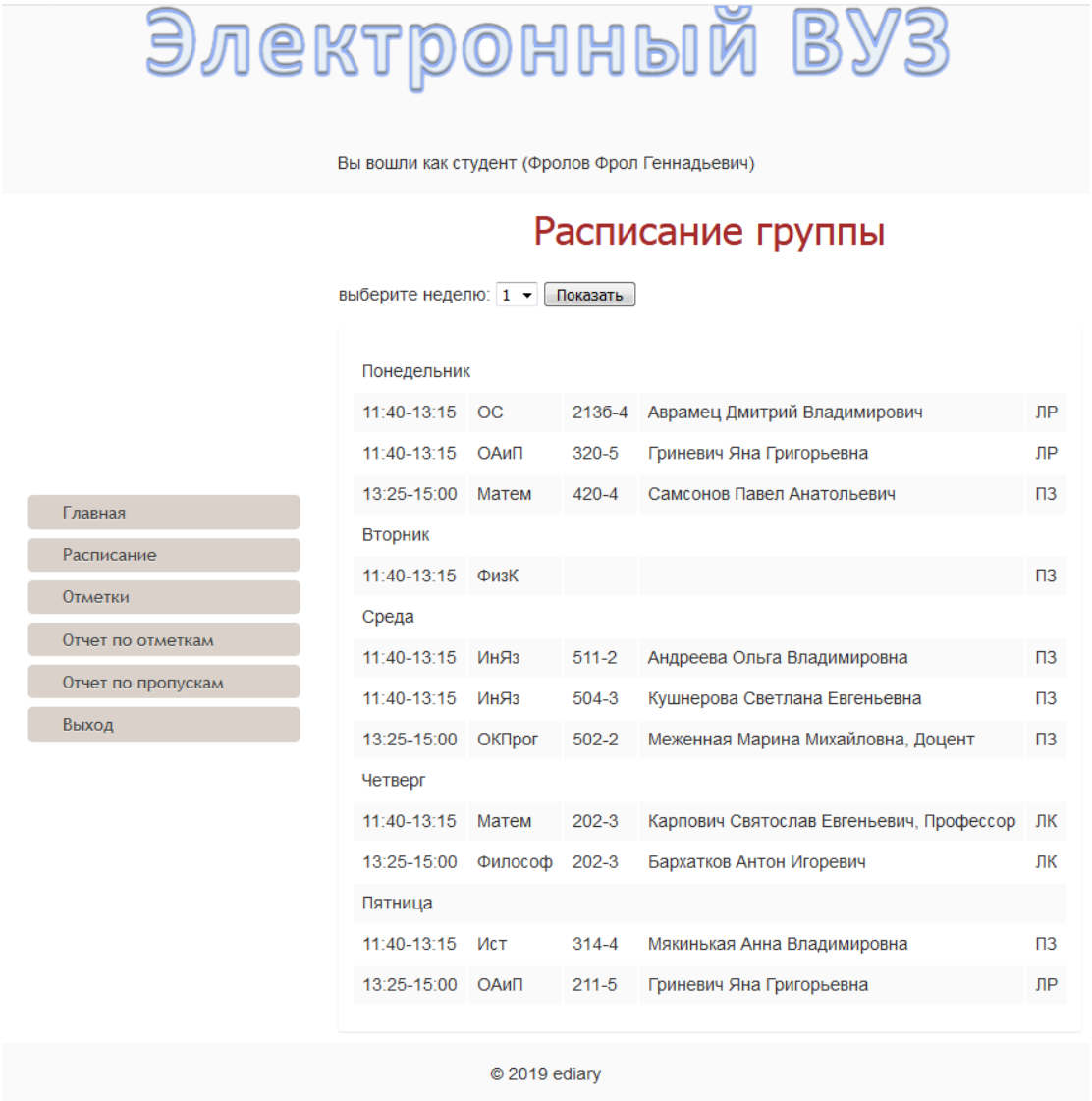

Рис. 40 – Расписание студента

3 ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА РАЗРАБОТКИ

3.1 Краткая характеристика веб-сервера Apache

Чтобы опубликовать веб-сайт, необходим либо статический, либо динамический веб-сервер.

Статический веб-сервер, состоит из компьютера ("железо") с сервером HTTP (ПО). Мы называем это «статикой», потому что сервер посылает размещенные файлы в браузер «как есть».

Динамический веб-сервер состоит из статического веб-сервера и дополнительного программного обеспечения, чаще всего сервера приложения и базы данных. Мы называем его «динамическим», потому что сервер приложений изменяет исходные файлы перед отправкой в ваш браузер по HTTP.

Веб-сервер сложно настраивать для обычного пользователя. Его настройками занимаются специально обученные люди – системные администраторы. Администрирование веб-сервера – очень сложный процесс, требующий высокой квалификации. Это обусловлено в первую очередь тем, что необходимо максимум внимания оказать безопасности. Ведь сервер, т.е. компьютер, должен быть подключен к сети интернет, и любой может подключиться к нему и произвести несанкционированные действия в случае обнаружения дыр в безопасности. Кроме того, веб-сервер должен работать (быть включенным) и быть подключен к сети Интернет круглосуточно. Что не всегда возможно для обычных компьютеров. В таких случаях на помощь приходят специализированные организации – веб-хостеры. Они предоставляют услуги аренды своих мощностей – позволяют арендовать веб-серверы, либо виртуальные вычислительные мощности, на которых можно установить веб-серверы. Эти услуги носят название веб-хостинг.

На данное время самым известным и распространенным является Apache. Он распространяется бесплатно, т.е. является свободно-

распространяемым. Используется как часть UNIX – подобной операционной системы, однако программисты нашли способы откомпилировать его и под другие операционные системы, в том числе, под windows, например.

Также имеются другие варианты веб-серверов:

IIS от компании Microsoft, распространяемый с ОС семейства Windows

nginx — свободный веб-сервер, разрабатываемый Игорем Сысоевым с 2002 года и пользующийся большой популярностью на крупных сайтах[2],[3]

lighttpd — свободный веб-сервер.

Google Web Server — веб-сервер, основанный на Apache и доработанный компанией Google.

Resin — свободный веб-сервер приложений.

Cherokee — свободный веб-сервер, управляемый только через web-интерфейс.

Rootage — веб-сервер, написанный на java.

THTTPD — простой, маленький, быстрый и безопасный веб-сервер.

Open Server — бесплатная программа с графическим интерфейсом использует множество исключительно свободного программного комплекса.

H2O — свободный быстрый веб-сервер, написанный на C.

Однако использовать в проекте будем именно Apache.

Данный веб-сервер имеет следующие преимущества:

- Бесплатность использования, даже при использовании с коммерческими целями.
- Надежность, стабильность. Т.к. он сделан на основе системы модуля unix, известной своей надежностью, простотой и стабильностью.
- Безопасный благодаря регулярным обновлениям.
- Гибкость. Apache гибок, учитывая наличие модульной структуры. Можно, например, подключить/отключить один или несколько любых модулей. Для этого необходимо лишь подредактировать ini-файл и перезапустить веб-сервер.

- Простота в использовании. Как правило, он работает сразу после установки. Во многих сборках unix-linux-подобных ОС Apache уже установлен и работает «из коробки». В других сборках его установка достаточно проста. Разве что есть вопросы по конфигурированию для работы с базами данных.
- Кроссплатформенность. После успеха на Unix-системах, была разработана версия и для windows.
- Большое сообщество. Например, на форуме https://www.linux.org.ru/tag/apache?section=2 находятся много советов и статей по настройкам.

Недостатки:

- Производительность. На высоконагруженных сайтах бывают проблемы производительности, особенно, при работе с большим трафиком.
- Сложность в конфигурации безопасности. Настройкой Apache необходимо заниматься профессиональным системным администраторам, иначе могут быть проблемы с безопасностью.

3.2 Используемые языки программирования

Для программирования сайтов используются разные языки программирования. Охарактеризуем самые часто используемые.

HTML (Hyper Text Mark Up Language) – не совсем язык программирования. Скорее, это язык текстовой разметки. От состоит из управляющих слов – тэгов. Эти тэги обрабатываются в браузере и с помощью их браузер создает и отображает веб-страницы.

CSS (cascad style sheet) – каскадные таблицы стилей. Также не совсем язык программирования. Однако он идет рука в руку с HTML и является неотъемлемой частью HTML, т.к. любое оформление веб-страницы возможно лишь с помощью CSS. Современные стандарты быстро изменяются и сейчас доступны CSS2 с множеством новых эффектов и возможностей. Например, на замену долго главенствовавшему Flash для отображения видео пришел

специальный тэг в HTML5 для аналогичных целей без применения flash-плагинов.

Всегда рядом с HTML и CSS идет javascript. По сути, эта троица языков является минимально необходимой для разработки любого статического сайта. Javascript выполняется на стороне клиента, в браузере. Обычно с его помощью настраивается поведение и реакции сайта на действия посетителя – обработка кликов по кнопкам, спискам, иным органам управления страницей.

Перечисленные три технологии могут обеспечить лишь статические страницы, т.е. без загрузки динамической информации. Эта проблема решается с помощью PHP (рекурсивное название PHP – Hypertex Preprocessor). Этот язык обрабатывает скрипты (обычные текстовые файлы, написанные с соблюдением синтаксиса PHP) и выдает динамические веб-страницы. PHP используется для backend – разработчиков.

Для создания динамических страниц почти всегда необходимо хранить большие объемы информации. Это реализуется с помощью баз данных и систем управления базами данных. Язык запросов SQL (structure query language) – используется для управления этими данными. В веб-серверах чаще всего используются MySQL или PostgreeSQL.

Python – также скриптовый язык, во многом схожий с PHP. Язык этот активно развивается, создан достаточно недавно, имеет высокую гибкость и мощность разработки.

Go – корпорация Google разработала свой язык и платит за создание веб-приложений за языке Go. Пока что мало распространен.

Ruby – быстрый язык веб-разработки, аналог PHP и Python. Близок к ним по синтаксису. Twitter использует этот язык в своих разработках.

3.3 Выбор базы данных

База данных – неотъемлемая часть практически любого веб-ресурса. Сравним две наиболее часто используемые сейчас базы данных: MySQL и PostgreSQL.

MySQL - это реляционная база данных. В ней работа с внутренними движками. На синтаксис запросов и их выполнение движок не влияет. Поддерживаются такие основные движки MyISAM, InnoDB, MEMORY, Berkeley DB. Они отличаются между собой способом записи данных на диск, а также методами считывания.

При выполнении запроса MySQL загружает весь ответ сервера в память клиента, при больших объемах данных это может быть не совсем удобно.

Postgresql представляет из себя объектно реляционную базу данных, которая работает только на одном движке - storage engine. Все таблицы представлены в виде объектов, они могут наследоваться, а все действия с таблицами выполняются с помощью объективно ориентированных функций. Как и в MySQL все данные хранятся на диске, в специально отсортированных файлах, но структура этих файлов и записей в них очень сильно отличается.

Postgresql - это проект с открытым исходным кодом, он разрабатывается командой энтузиастов, и разработчики пытаются максимально соответствовать стандарту SQL и реализуют все самые новые стандарты. Но все это приводит к ущербу простоты. Postgresql очень сложный и из-за этого он не настолько популярен как MySQL.

Postgresql для навигации по результатам запроса использует специальные указатели – курсоры. Можно получить ответ от запроса, получить маркер на данные, потом перемещаться по ним, например, построчно последовательно.

Mysql разрабатывалась из расчета получения наибольшей скорости работы. А Postgressql, как более современное программное обеспечение, основной упор сделала на стандарты.

В целом PostgreSQL работает быстрее, за исключениям использования первичных ключей.

В нашем проекте используется БД MySQL, т.к. она наиболее органично работает в связке с языком PHP и веб-сервером Apache.

Таким образом, для разработки будем использовать самый распространенный веб-сервер Apache, связку языков HTML+Javascript+CSS+PHP. В качестве базы данных – MySQL как самую массово используемую и предоставляемую на веб-хостингах в сети Интернет.

4 ЭКСПЛУАТАЦИЯ

4.1 Требования к аппаратному обеспечению

Общие требования к клиентским станциям

Обязательные требования:

- наличие сетевой карты (соединение с сетями передачи данных);
- веб-браузеры, рекомендуемые для работы в системе: Google Chrome, Mozilla Firefox;
- установленный пакет офисных приложений (для MS Office версия не ниже 2003).

Рабочие места ОС Windows

Минимальная комплектация:

- операционная система: Windows 7/8/10;
- тактовая частота процессора: не менее 2 Ghz;
- оперативная память (ОЗУ): не менее 2 Gb;
- жёсткий диск: не менее 10 Gb свободного дискового пространства;
- экран: не менее 17”, разрешение не менее 1024×768.

Рекомендуемая комплектация:

- операционная система: Windows 7/8/10;
- тактовая частота процессора: не менее 2 GHz. Dual Core;
- оперативная память (ОЗУ): не менее 4 Gb;
- жёсткий диск: не менее 15 Gb свободного дискового пространства;
- экран: не менее 17”, разрешение не менее 1280×1024.

Оптимальная комплектация (рабочие места делопроизводителей и ответственных руководителей):

- операционная система: Windows 7/8/10 64 bit;

- тактовая частота процессора: не менее 2 GHz. Dual Core;
- оперативная память (ОЗУ): не менее 8 Gb. (ОС 64 bit);
- жёсткий диск: не менее 25 Gb свободного дискового пространства;
- экран: не менее 17”, разрешение не менее 1920×1080.

Рабочие места Mac OS

Минимальные требования:

Для корректной работы программы пользовательский компьютер должен соответствовать следующим минимальным программно-аппаратным требованиям:

- операционная система: Mac OS X 10.5 (Leopard) и выше;
- тактовая частота процессора: не менее 1.0 ГГц;
- оперативная память (ОЗУ): не менее 4 Гб;
- жёсткий диск: не менее 5 Gb свободного дискового пространства;
- экран: разрешение не менее 1200×800.

Рабочие места ОС Linux

Минимальная комплектация:

жёсткий диск: не менее 10 Gb свободного дискового пространства;

экран: не менее 17”, разрешение не менее 1280×1024.

операционные системы: Ubuntu, Linux Mint, OpenSUSE, Fedora, CentOS;

тактовая частота процессора: не менее 1,5 Ghz;

оперативная память (ОЗУ): не менее 2 Gb.

Рекомендуемая комплектация:

операционные системы: Ubuntu, Linux Mint, OpenSUSE, Fedora, CentOS;

тактовая частота процессора: не менее 2 GHz. Dual Core;

оперативная память (ОЗУ): не менее 4 Gb.

Учитывая тот факт, что рассматривается веб-приложение, вообще к данной автоматизированной системе можно получить доступ через веб-

браузер, то тип и характеристики устройства влияют мало. Главное требование – наличие браузера на устройстве с доступом в интернет и дисплей, способный корректно отобразить информацию.

4.2 Инструкция пользователя

Пользователю необходимо:

1) Получить ссылку для входа на сайт.

2) Получить личный логин и пароль от администратора системы.

3) Авторизоваться, введя логин и пароль.

Интерфейс автоматизированной системы интуитивно понятен.

Авторизация.

При входе пользователь увидит окно авторизации с запросом логина (имени пользователя) и пароля (рис.41). Логин и пароль выдается администратором.

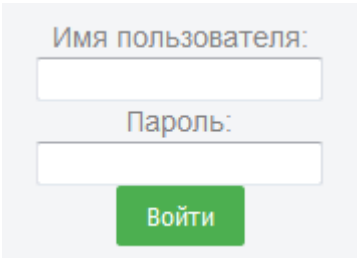

Рис. 41 – Окно авторизации

Администратор.

После авторизации с правами администратора доступно максимальное количество функций.

После авторизации с правами студента/родителя/преподавателя доступно ограниченное количество функций.

Преподаватель.

Преподаватель может выставлять оценки в разделе «Редактор отметок».

Чтобы выставить отметку необходимо выбрать курс. При этом автоматически будет подгружен список групп, принадлежащих выбранному курсу. При выборе нужной группы автоматически будет подгружен список студентов данной группы и список предметов данной группы. Т.е. исключены некорректные выборы.

Для простоты разработки функция удаления оценки за нужную дату по нужному студенту и нужному предмету реализована выбором отметки «пусто» и нажатие на кнопку «Добавить отметку».

При нажатии на кнопку «Показать отметки» показывается ведомость отметок по данному студенту.

Студенты и родители.

Студенты и их родители могут просмотреть расписание обучающегося. При этом нет необходимости выбирать группу. Т.к. группа уже известна (по логину). Необходимо лишь выбрать номер учебной недели (если расписание меняется и зависит от недели).

5 ЭКОНОМИЧЕСКАЯ ЧАСТЬ

В финансовой части курсового плана обязана быть определена: себестоимость - стоимость, финансовая эффективность плана.

Смета затрат включает следующие статьи:

– дополнительная заработная плата разработчиков web-сайта;

– отчисления на социальное страхование;

– создание фирменного логотипа компании:

– создание дизайна web-сайта:

– верстка основного шаблона сайта;

– заполнение сайта уникальным контентом.

Для подсчета себестоимости продукции и финансовой производительности предоставленного приложения, надо знать такие составляющие, как:

- расходов на техническое и ПО проекта;
- расплата расходов на вещественные ресурсы;
- расплата расходов на энергоресурсы;
- амортизационные отчисления;
- расплата фонда заработной платы;- прочие расходы.

Время на проект:

Трудоемкость ПО разработанного за время:

$$T=to+ta+tб+tn+tд+tот,$$

где

to- затраты на подготовку;

ta- затраты на сам алгоритм;

tб- затраты на составление блок-схемы алгоритма решение задачи;

tn- затраты на написание листинга по блок-схеме;

tд- затраты на разработку документации задачи;

tот- затраты на тестовые запуски и отладочные операции, поиск ошибок.

Если число операторов около Q=4800, рассчитаем затраты на обработку описания. Учитывая уровень знаний программиста, находим:

To = Q*B\ (75…85*K).

Где B – во сколько раз затраты будут осложнены недостаточным описанием и, соответственно, необходимостью доработки описания.

(B = от 1,2 до 5). B = 2,1

K- коэффициент квалификации разработка (K = от 0,8 до 1,5) K = 0,9

Тогда вычислим трудоемкость (табл.2).

Таблица 2.

№	Этапы проектирования	B	K	Формула расчета	Трудоемкость чел. час.
1	Подготовительный t_o	2,1	0,9	T_o=4800*2,1 10080 82*0,9 = 73,8 =	137
2	Алгоритм решения задачи t_a			t_a= 4800 =68*0,9	78
3	Блок-схема алгоритма $t_б$			$t_б$= 4800 =72*0,9	74
4	Составление программы по блок-схеме t_n			t_n= 4800 =75*0,9	71
5	Отладка программы на ЭВМ тот			$t_{от}$=1,5*$t_{от}^{A}$ = $t_{от}^{A}$= 4800 = 45*0,9	177 118
6	Подготовка документации t_g			$t_{др}$ = 4800 =165*0,9	32
				$T_{до}$= 0,75*32 =	24
	Итого:				$T_н$=522

Примечание:

$t_g = t_{gp} + t_{go}$,

где

t_{gp}- затраты труда на подготовку материалов;

t_{go}-затраты труда на полный цикл создания документации.

Примем, что над программой трудятся коллектив из двух профессиональных программистов.

Инженер-программист 13 разряда, с трудоемкостью работ $t_n = T_н * 0,6 =$ 313руб.

Расчет фонда заработной платы:

Работа программиста по условиям труда квалифицируется вредной. Согласно положению об оплате труда доплата за вредность составляет до 12% в зависимости от аттестации рабочего места.

Для расчета заработной платы, тарифной ставки учитывается поправочный коэффициент за условие труда 1,12. Тарифный коэффициент инженера-программиста 2,37 разряд 13.

Дополнительный фонд заработной платы – 10%

Отчисления на социальные нужды- 20% Определение фонда заработной платы инженера-программиста 13 разряда.

Часовая тарифная ставка 13 разряда, с учетом доплаты за вредность, составляет:

$C_ч = C_1 * K_{тар} * K_{ут} \backslash 169,6 = 9752 * 2,37 * 1,12$

169,6 = 152,63руб.

Тарифная зарплата за отработанное время:

$З_{тар} = C_ч^n * T^{из}$ общ = 152,63*313 = 47773руб.

Дополнительная зряплата-10% от $З_{тар}$

$З_{доп} = 0,1*З_{тар} = 16614*0,1 = 4777$руб.

Оплата программиста:

$З_{общ}^{n} = З_{тар}+З_{доп} = 47773+4777 = 52550$руб.

Фонд оплаты труда на разработку программы составит:

$Ф_{от} = З_{общ} = 52550$руб.

Отчисления на социальные нужды:

$Ф_{соц} = 0,20*Ф_{от} = 52550*0,20 = 10510$ руб.

Общий фонд заработной платы на разработку программного обеспечения составит:

$Ф = Ф_{от}+Ф_{общ} = 52550 + 10510 = 63060$ руб.

Среднемесячная заработная плата специалиста составит:

$З_{ср} = Ф\Т_{н}*169,6 = 52550/522*169,6 = 17073$ руб.

Расчет затрат на техническое обеспечения проекта.

Для разработки программного обеспечения необходимы технические средства:

- монитор - 15000 руб

- системник - 60000 руб

- клавиатура – 1000 руб

- манипулятор типа мышь - 1000 руб

- лазерный принтер - 12000 руб

Затраты на техническое и программное обеспечение $С_{оф}$ = 89000руб.

Основные фонды переносят свою стоимость на создаваемую продукцию в виде амортизационных отчислений.

Срок использования технических средств 3 года.

Амортизация- это денежное выражения износа основных фондов.

$А = С_{оф}*п\% / 100*262*t = 89000*33,3/100*262 * 53 = 2134$ **руб.**

Прочие затраты.

Затраты на оплату машинного времени при отладке программы определяются путем умножения фактического времени отладки программы на цену машино-часа арендного времени:

$З^{мв} = С_{час}*t_{эвм}$;

Где $С_{час}$- цена машино-часа арендного времени, руб/час;

$t_{эвм}$- фактическое время отладки программы ЭВМ;

Фактическое время отладки вычислим по формуле:

$t_{эвм} = t_{н}+t_{до}+t_{от}$ = 272 руб.

По усредненным расчетам фирм цена машино-часа арендуемого времени составляет 22,8руб.

Общие затраты на оплату машинного времени составляют:

$З_{мв} = 272*22,8 = 6202$руб.

Расчет затрат на текущий ремонт.

Затраты на текущий и профилактический ремонт принимаются равными 5% от стоимости ЭВМ

$З_{тр} = 0,05*С_{бал} = 89000*0,05 = 4450$руб.

Затраты на материалы, необходимые для обеспечения нормальной работы ПЭВМ составляют около 1% от стоимости ЭВМ:

$З_{вм} = 0,01*С_{бал} = 89000*0,01 = 890$руб.

Затраты на материальные ресурсы:

1. Вспомогательная литература 1000руб.

2. Канцтовары 1000руб.

Итого материальные ресурсы 2000руб.

Расчет затрат на энергоресурсы:

Для выполнения программного проекта используется электроэнергия на технологические цели:

$С_{эр} = N*n*ц*t = 0.5*8*4.84*424 = 8209$

N- мощность технических средств- 0,5квт

n- количество используемых технических средств.

ц- цена одного киловатта электроэнергии – 5.38 руб.

t- время работы в часах. 53*8 = 424

Расчет экономической эффективности проекта (табл. 3).

Таблица 3 - Калькуляция себестоимости и отпускной цены

№ п/п	Статьи и затраты	Сумма (руб)
1	Материалы	2830
2	Энергоресурсы	8209
3	Основная заработная плата	47773
4	Дополнительная заработная плата	4777
5	Отчисление на социальные нужды	10510
6	Амортизационные нужды	13485
7	Прочие затраты	18214
8	Расходы на подготовку и осваивание производства 0,5% от Σ 1¿7	529
9	Общехозяйственные расходы 3 %Σ от 1¿/7	3174
10	Производственная себестоимость 1/9	109501
11	Прибыль предприятия 20% от гр 10	21900
12	Цена продукции гр 10+11	131401
13	НДС- 14% от гр 12	18396
14	Отпускная цена гр 12+гр 13	149797

Эф = (Ц-С)/К

Где Эф- экономическая эффективность.

Готовая эффективность создание системы определяется из выражение:

$Э_г = \Delta З/(Ен+Кр) = 124437/0,44 = 282811$

Где $\Delta З$ - изменение размера затрат на создание и обслуживание информационной системы определяется из выражения:

$\Delta З = \Delta И\sum$ - К*(Ен+Кр) = 170988-105798*(0.15+0.29)=170988-46551=124437

где ΔИΣ - абсолютное изменение издержек работы с использованием программ.

$К_р$- коэффициент реновации, принимается как норма амортизации с учетом срока службы существующего оборудования (срок службы системы Т сл – 3 года) и определяется по формуле:

$К_р$ = Ен/((1+Ен) тел -1),

$К_р$ = 0,15/((1+0,15)3 – 1) = 0,29

К- единовременное капиталовложение

К = 105798 руб (программные и технические затраты, фонд заработной платы, прочие расходы, материальные затраты, затраты на энергоносители и прочие расходы)

Тогда изменение размера затрат составит:

ΔЗ = 124437 руб.

Точная эффективность планируется в размере

$Э_г$ = 282811 руб.

Теперь можно определить срок окупаемости ток создания новой системы из выражения:

$Т_{ок}$ = К/$Э_г$ = 0,37

Данный срок окупаемости учитывает только материальные преимущества создания новой системы.

В целом система считается рентабельной если ее срок окупаемости меньше года.

Технико-экономические показатели проекта приведены в табл.4.

Таблица 4 – Технико-экономические показатели

№ п/п	Показатели	Единица измерения	Количество
1	Общая трудоемкость работ в том числе	Чел/час	522
1.1	Техническое задание	Чел/час	137
1.2	Техно-рабочий проект	Чел/час	223
1.3	Внедрение	Чел/час	233
2	Срок разработки проекта по плану	Дни	53
3	Численность персонала	Чел.	1
4	Себестоимость продукции	руб.	109501
5	Прибыль предприятия	руб.	21900
6	Отпускная цена продукции	руб.	149797
7	Среднемесячная заработная плата	руб.	17073
8	Производительность труда	%	100
9	Срок окупаемости затрат	год	0,37

$ПТ = T_н/T_{пл}*100\% = 522/424*100\% = 100$

$T_н$- общая трудоемкость на программу

$T_ф$- фактическая трудоемкость

$T_ф = ч*т*8 = 1*53*8 = 424$

Где: Ч- численность программистов-1

t- плановый срок выполнения работ-53

8- продолжительность

6 МЕРОПРИЯТИЯ ПО ТЕХНИКЕ БЕЗОПАСНОСТИ И ОХРАНЕ ОКРУЖАЮЩЕЙ СРЕДЫ

6.1 Общие принципы и законодательство по охране труда

Вся работа по охране труда должна носить системный характер. В разделе X гл. 33 ст. 209 Трудового кодекса Российской Федерации дано следующее определение понятия "охрана труда".

Охрана труда - система сохранения жизни и здоровья работников в процессе трудовой деятельности, включающая в себя:

• правовые,

• социально-экономические,

• организационно-технические,

• санитарно-гигиенические,

• лечебно-профилактические,

• реабилитационные и иные мероприятия.

Задачу эффективного обеспечения охраны труда нельзя решить за счет проведения одних или группы перечисленных мероприятий.

Трудовой кодекс Российской Федерации определяет:

• основные направления государственной политики в области охраны труда;

• государственные нормативные требования охраны труда;

• обязанности работодателя по обеспечению безопасных условий и охраны труда;

• обязанности работника в области охраны труда;

• государственное управление охраной труда;

• мероприятия по обеспечению охраны труда;

• основные функции органов государственного надзора и контроля за соблюдением законодательства о труде и об охране труда;

• организации, которые должны осуществлять общественный контроль за охраной труда;

• принципы управления охраной труда непосредственно в организации;

• обеспечение прав работника на охрану труда;

• право и гарантии работника на труд в условиях, соответствующих требованиям охраны труда;

• финансирование мероприятий по улучшению условий и охраны труда.

6.2 Пожарная безопасность при работе на персональном компьютере

Пожарная безопасность обеспечивается следующими мерами:

1) системой предотвращения пожаров;

2) системой пожарной защиты;

3) организационными мероприятиями по пожарной безопасности.

Предотвращение пожара достигается следующими мерами:

1) предотвращение образования горючей среды;

2) предотвращение образования в горючей среде источников зажигания.

В систему пожарной защиты входят первичные средства пожаротушения, приведенные в таблице 5.

Таблица 5 – Перечень первичных средств пожаротушения

Площадь, кв.м.	Первичные средства пожаротушения (тип, наименование)	Количество, шт.	Огнегасящий эффект
24	углекислотные огнетушители и ручные ОУ-2, войлок, кошма, асбест	2	Разбавление воздуха углекислым газом и снижение в нем содержания кислорода до концентрации, при которой прекращается горение. Огнетушащий эффект указанным газом обуславливается потерями теплоты и снижением теплового эффекта реакции прекращение доступа кислорода к горящим элементам.

Категория пожарной и взрывопожарной опасности здания определяется как категория В (пожароопасное), так как здесь находятся твердые сгораемые вещества и материалы, такие как компьютеры, мебель, проводка, бумага и т.д. Способность конструкций сопротивляться воздействию пожара в течение определенного времени при сохранении его эксплуатационных функций называется огнестойкостью.

В соответствии со строительными нормами и правилами для здания, в котором проводилась данная работа, степень огнестойкости I - III, но, с учетом этажности здания (больше 5 этажей) данное железобетонное здание имеет II степень огнестойкости.

Организационными мероприятиями пожарной профилактики является обучение производственного персонала противопожарным правилам, издание необходимых инструкций и плакатов, средств наглядной агитации. Обязательным является наличие плана эвакуации, пожарного водопровода и пожарной сигнализации.

6.3 Мероприятия по охране окружающей среды

Экологическая служба предприятия должна решать следующие вопросы:

- поддерживать контроль за соблюдением в подразделениях предприятия экологического законодательства, инструкций, стандартов и нормативов по охране окружающей среды,
- разрабатывать проекты перспективных и текущих планов по охране окружающей среды,
- контролировать выполнение планов по охране окружающей среды,
- участвовать (в качестве представителя предприятия) в проведении экологической экспертизы технико-экономических обоснований, проектов расширения и реконструкции действующих производств, а также создаваемых новых технологий и оборудования, разработке мероприятий по внедрению новой техники,
- участвовать в проведении научно исследовательских и опытных работ по очистке промышленных сточных вод, предотвращению загрязнения окружающей среды выбросами вредных веществ в атмосферу, уменьшению или полной ликвидации технологических отходов, рациональному использованию земельных и водных ресурсов,
- контролировать соблюдение технологических режимов природоохранных объектов,
- контролировать состояние окружающей среды в районе расположения предприятия,
- составлять технологические регламенты, графики аналитического контроля, паспорта, инструкции и другую техническую документацию,

- проверять соответствие технического состояния оборудования требованиям охраны окружающей среды и рационального природопользования,
- вести в установленном порядке отчетность о выполнении мероприятий по охране окружающей среды.

ЗАКЛЮЧЕНИЕ

Электронный дневник – это интерактивная система, создающая единое информационное пространство, объединяющая ВУЗ, семьи учащихся и органы управления образованием. Данная система создает прозрачную, открытую среду информирования пользователей, обеспечивает полноту, доступность и достоверность информации, является современным и эффективным инструментом управления ВУЗом и значительно влияет на повышение качества образования.

При использовании системы студент имеет возможность видеть свои оценки, а так же получать сообщения и любую другую информацию, которую преподаватели или деканат хотели бы донести до своих студентов.

Может отслеживать возможные изменения в расписании занятий.

Благодаря электронному дневнику повышается успеваемость и посещаемость студентов. Кроме того, эта система позволяет родителям не оставаться в стороне от учебного процесса. Родители получают возможность своевременно видеть выставляемые отметки, контролировать посещаемость, общаться с преподавателями и представителями администрации ВУЗа в режиме online, получать сообщения о жизни ВУЗа и группы, своевременно могут принять меры для исправления ситуации с успеваемостью.

Использование в деятельности всех преподавателей электронного дневника завершает построение единого информационного пространства ВУЗа. С помощью этой системы преподаватели имеют возможность отслеживать успеваемость обучающегося, автоматически получать любую качественную и количественную информацию, использовать элементы дистанционного обучения и электронного документооборота. Система может помочь преподавателю и с составлением отчетной документации, т.к. можно предусмотреть возможность снабжать домашнее задание дополнительными электронными материалами (файлами).

Преподаватели могут получать информацию о своей нагрузке или об изменениях в расписании, тратить поменьше времени на выполнение арифметических или алгебраических операций для выведения итоговых отметок студентам, планировать свою недельную нагрузку, делиться опытом с коллегами.

Администрацию образовательного учреждения интересуют вопросы, связанные с реализацией государственного образовательного стандарта. Администрация контролирует заполняемость журнала и видит полную картину успеваемости в любых срезах: по группам, по предметам, индивидуально по преподавателю или студенту. С помощью электронного журнала деканат контролирует посещаемость занятий. Контролируется работа со слабоуспевающими студентами, прослеживается накопляемость оценок. Ведется контроль за преподаванием учебных предметов - выполнение учебной программы, выполнение практической части, своевременность выставления оценок за письменные работы (практические, семинары и т.д.). Введение электронного дневника - это новый стандарт информатизации ВУЗов в ближайшем будущем. Он позволяет ВУЗу встать на современную ступень информационно-коммуникативных технологий (ИКТ), соответствовать всем ее требованиям.

Система электронных дневников - это удобный, мощный, а главное совершенно бесплатный инструмент для создания единого информационно-образовательного пространства учебного заведения и взаимодействия образовательного учреждения с родителями студента.

Таким образом, ввод электронного журнала ориентирует на построение в учреждении образования открытого образовательного пространства в котором взаимодействуют все участники учебного процесса.

Результатом дипломной работы стало создание автоматизированной информационной системы. Она имеет интуитивно-понятный и дружелюбный интерфейс и функционал.

Во время работы над данным проектом были решены следующие задачи:

1. Была поставлена и изучена предметная область;
2. Спроектирована структура сайта;
3. Создан многостраничный веб-сайт;
4. Разработана база данных.

Возможные недостатки и доработки сайта на перспективу:

1) Облегчить импорт данных в справочники.

2) Улучшить дизайн страниц.

3) Использовать модель MVC.

4) Использовать фреймворк.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Образовательные и воспитательные стратегии в современном обществе: сборник научных трудов по материалам I Международной научно-практической конференции 31 мая 2016 г. Нижний Новгород: НОО «Профессиональная наука», 2016. 183 с.

2. Якушина Е.В., Буланов С.В., Гаврилин И.А. Электронный журнал и виртуальный дневник ученика. Журнал «Школьные технологии», 4-2010.

3. Электронный ресурс: Дневник.ру, школьная социальная сеть. Режим доступа: http://dnevnik.ru/ Дата доступа: 14.11.2019.

4. Электронный ресурс: Баллов.нет, система электронных школьных журналов, 2010. Режим доступа:http://ballov.net/. Дата доступа: 14.11.2019.

5. Электронный ресурс: MySQL Reference Manual: Режим доступа: http://dev.mvsql.eom/doc/refman/5.0/en/historv.html Дата доступа: 14.11.2019.

6. Кристиан Дари, Богдан Бринзаре, Филип Черчез-Тоза, Михай Бусика. AJAX и PHP. Разработка динамических веб¬приложений. - СПб.: Символ-Плюс, 2007. - 336 стр.

7. Стелзнер, М. Контент-маркетинг / Стелзнер М. – М.: Манн, Иванов и Фербер, 2013. – 288 с.

8. Уйэншенк, С. 100 новых главных принципов дизайна / С. Уйэншенк. – СПб.: Питер, 2015. – 289с.

Приложение А. Листинг страницы index.php

```
<?php
    /*
    Главная страница
    */
    header('Content-type: text/html; charset=utf-8');
    error_reporting(E_ALL);
    include('auth.php');
    include('func.php');
    $info=get_user_info();
?>
<html>
<head>
    <meta charset="utf-8">
    <title>Электронный дневник</title>
    <link rel="stylesheet" type="text/css" href="style.css">
</head>

<body>
<table border="0">
    <!-- баннер -->
    <tr>
    .....<td colspan=2 style="text-align:center">
    .....
<a href="index.php"><img src="images/logo.png"/></a><br>
    .....
<p style='text-align:center'><?php echo $info['text_enter'];?></p>
    .....</td>
    </tr>
```

```
        <tr>
        .....<!-- меню -->
        .....<td width="30%">
        .....
<?php
        .....
include('menu.php');
        .....  ?
>
        .....</td>

        .....<!-- контент -->
        .....<td width="70%">

        .....
<h1>Добро пожаловать!</h1>
        .....
<h2>Автоматизированная система "Электронный дневник"</h2>

        .....</td>
        </tr>

        <!-- подвал -->
        <tr>
        .....<td colspan=2>
        .....
<p style='text-align:center'>&copy; 2019 ediary</p>
        .....</td>
        </tr>
```

```
</table>

</body>
</html>
```

Приложение Б. Листинг страницы login.php

```
<?php
    /*
    Скрипт авторизации юзера
    */
    header('Content-type: text/html; charset=utf-8');
    if (session_id() == '') session_start();
    include "database.php";
    include "func.php";
    $con=connect();

    if(isset($_POST['submit'])){
    $login=htmlentities(trim($_POST['user']), ENT_QUOTES, 'UTF-8');
    ...$password=htmlentities(trim($_POST['pass']), ENT_QUOTES,
'UTF-8');
    .....// ищем такой логин и пароль
    .....$query="
    .....
SELECT id, login, level
    .....
FROM users
    .....
WHERE 1
    .....
AND (users.login='$login'.....OR users.id='$login')
    .....
AND (
    .....
    users.password=MD5('$password')
```

```
        .....
        OR users.password='$password'
        .....
    )
        .....
    LIMIT 1
        ..... ";
        …..$res=mysqli_query($con, $query) or die(mysqli_error($con));
        .....$row=mysqli_fetch_array($res, MYSQLI_ASSOC);
        .....if (
        .....
    (!empty($row['id']))
        ..... )        {
//успешный вход

        .....          /
/ запись в лог инфы о входе
        .....
    $login_str=mysqli_real_escape_string($con, $row['login']);
        .....
    $ip=$_SERVER['REMOTE_ADDR'];
        .....
    $oper='login';
        .....
    $query="
        .....
    INSERT INTO log
        .....
    SET ip='$ip', login='$login_str', oper='login'
```

```
        ..................................................................................................."
;
        ..................................................................................................
$res=mysqli_query($con, $query) or die(mysqli_error($con));

        ..................................................................................................
$_SESSION['login'] =  $row['login'];
        ..................................................................................................
$_SESSION['level'] =  $row['level'];
        ..................................................................................................
$_SESSION['id'] =  $row['id'];
        ..................................................................................................
header("Location: index.php");
        ..................................................................................................
exit;
        ..................................................................................................}
        ..................................................................................................else
        ..................................................................................................
echo '<p>Логин или пароль неверны!</p>';
        };
    ?>
    <head>
        <title>Авторизация</title>
        <meta charset="UTF-8">
        <link href="style.css" rel="stylesheet" />
    </head>

    <form id="loginForm" method="post">
        <div class="field">
        ..................................................................Имя пользователя:
```

```
            <div class="input"><input type="text" name="user" value="admin"
id="login" /></div>
            </div>

            <div class="field">
            .....Пароль:
            .....<div class="input"><input type="password" name="pass"
value="pass10" id="pass" /></div>
            </div>

            <div class="submit">
            .....<button type="submit" name="submit">Войти</button>
            </div>
        </form>
```

Приложение В. Листинг файла menu.php

```
<?php
//  header('Content-type: text/html; charset=utf-8');
    error_reporting(E_ALL);
//  include('auth.php');
    $menu= '
    <link rel="stylesheet" type="text/css" href="menu_style.css">
    <ol class="rounded">
    .............................<li> <a href="index.php">Главная</a> </li>';
    // меню по уровням доступа: 10 - админ, 6-препод, 3-студент, 2 - родитель, 1- гость
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="students.php">Студенты</a> </li>';
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="groups.php">Группы</a> </li>';
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="teachers.php">Преподаватели</a> </li>';
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="subjects.php">Предметы</a> </li>';
    if ( in_array($_SESSION['level'], array(10,3,2)) ) $menu.='<li> <a href="timetable.php">Расписание</a> </li>';
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="parents.php">Родители</a> </li>';

    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="departments.php">Кафедры</a> </li>';
    if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="hard.php">Нагрузка преподавателей</a> </li>';
```

```
        if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="mark_types.php">Типы отметок</a> </li>';

        if ( in_array($_SESSION['level'], array(10,3,2)) ) $menu.='<li> <a href="marks.php">Отметки</a> </li>';
        if ( in_array($_SESSION['level'], array(10,6)) ) $menu.='<li> <a href="marks_editor.php">Редактор отметок</a> </li>';
        if ( in_array($_SESSION['level'], array(10,3,2)) ) $menu.='<li> <a href="report_marks.php">Отчет по отметкам</a> </li>';
        if ( in_array($_SESSION['level'], array(10,3,2)) ) $menu.='<li> <a href="report_skipping.php">Отчет по пропускам</a> </li>';

        if ( in_array($_SESSION['level'], array(10)) ) $menu.='<li> <a href="users.php">Пользователи</a> </li>';
        $menu.='
        .....<li> <a href="?do=exit">Выход</a> </li>
        </ol>';
        echo $menu;
    ?>
```