Федеральное государственное автономное образовательное
учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт космических и информационных технологий
Кафедра «Информационные системы»

УТВЕРЖДАЮ
Зав. кафедрой ИС
_____ С.А. Виденин
«___» _____ 2016 г.

БАКАЛАВРСКАЯ РАБОТА

09.03.01 Информатика и вычислительная техника
«Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям»

Руководитель _____ С.А. Виденин
подпись, дата инициалы, фамилия

Выпускник _____ Д.А. Корнилов
подпись, дата инициалы, фамилия

Нормоконтролер _____ Л.С. Троценко
подпись, дата инициалы, фамилия

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме «Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям» содержит 49 страниц текстового документа, 2 приложения, 7 использованных источников, 18 листов графического материала.

АВТОМАТИЗИРОВАННОЕ РАБОЧЕЕ МЕСТО, PHP, CSS, HTML ЦДО, ПРОГРАММЫЙ ПРОДУКТ, ПО, РАСПИСАНИЕ, АВТОМАТИЗАЦИЯ, БАЗА ДАННЫХ, СУБД, MYSQL

Объектом исследования является автоматизированная система расписания для центра дополнительного образования.

Предметом исследования является применение современных информационных технологий и средств веб–разработки для создания автоматизированной информационной системы.

Целью данной ВКР, является – повысить эффективность составление расписания диспетчером, путем автоматизации процесса.

Основные задачи:

а) подробно описать актуальность проблему и предметную область, выявить основные проблемы;

б) провести анализ и подобрать удобную для студентов и преподавателей автоматизированную систему;

в) аргументирование экономической эффективности путем внедрение автоматизированной системы.

Основные результаты:

а) разработана автоматизированная система «Расписание для ЦДО»;

б) повышена эффективность составления расписания, путём автоматизации процесса.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ 4
1 Анализ предметной области 7
1.1 Анализ работы центра дополнительного образования 7
1.2 Обзор существующих решений проблемы составления расписания 8
1.3 Формирование требований к автоматизированной системе 15
1.4 Вывод к первому разделу 22
2 Реализация автоматизированной системы. 24
2.1 Выбор инструментов для разработки автоматизированной системы 24
2.2 Особенности программной реализации автоматизированной системы 35
2.3 Тестирование автоматизированной системы 40
2.4 Вывод ко второму разделу 46
ЗАКЛЮЧЕНИЕ 48
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 49
ПРИЛОЖЕНИЕ А 50
ПРИЛОЖЕНИЕ Б 59

ВВЕДЕНИЕ

На дворе уже XXI век, век, когда человек не мыслит себя, своё существование вне информационного потока. То, что ещё двадцать лет назад казалось результатом вымысла писателей – фантастов, сегодня ежедневная обыденность. Люди так быстро привыкли к новым видам связи: сотовой и спутниковой, интернету, гаджетам и компьютерам, что кажется, словно так было всегда. Развитие информационных технологий продвигается вперёд семимильными шагами, информационная техника беспрестанно совершенствуется. Компьютеры теперь необходимы абсолютно во всех сферах экономики: они контролируют производственные процессы на фабриках, заводах, фермах, содержат базы данных финансовых, социальных, коммерческих учреждений. Информационные технологии внедряются в образовательных организациях, организациях здравоохранения, культуры и науки.

А ведь исследования в сфере информатизации различных отраслей экономики начались ещё в пятидесятые годы прошлого столетия. Одной из задач таких исследований было составление расписаний. На первый взгляд, такая задача не кажется актуальной и значимой. Но это лишь на первый взгляд, ведь правильно составленное расписание способно оптимизировать работу транспортных систем, работу в промышленных цехах, на предприятиях массового обслуживания, в медицинских учреждениях и других организациях.

Современная реальность ставит перед специалистами немало задач, которые нельзя решить методом полного перебора, а для большинства моделей теории расписания составление наиболее оптимального расписания – задача почти неразрешимая, особенно если учитывать различного рода ограничения производственного, организационного характера и человеческий фактор.

ВУЗ тоже является предприятием, учреждением образования, и он тоже проходит через процесс автоматизации. И, хотя понятие образовательной деятельности является единым для всех образовательных учреждений

государства, каждый ВУЗ по–своему проходит процесс составления расписания. Огромное влияние на процессы автоматизации высшего учебного заведения оказывает состояние бюджета и возможность использования программных продуктов, которые предлагаются на рынке.

Одной из самых важных проблем правильной организации учебного процесса в ВУЗе – это задача составления автоматизированного учебного расписания, которая является главной для диспетчерской службы высшего учебного заведения. Такое расписание должно учитывать большое количество различного вида ограничений, чтобы оптимизировать учебную деятельность студентов, правильно распределить их физическую и психологическую нагрузку, обеспечить равномерную занятость студенческих групп и преподавателей.

Многие люди считают, что опытные диспетчеры могут вручную составить расписание, оптимальное для учебного процесса и общественной жизни ВУЗа, но, учитывая введение модульно–рейтинговой системы образования, увеличение количества студентов, укрупнение ВУЗов, необходимость использования и учёта большого количества данных, такое мнение, безусловно, можно считать ошибочным.

Конкретно задачу «Расписание» решают многие программные продукты, и хорошо с этим справляются. Но, задачей моей дипломной работы является сформировать оптимальное расписание для студентов центра дополнительного образования (в дальнейшем ЦДО), что является более узкой задачей. Главной проблемой составление данного расписания заключается в том, что в основном все студенты обучаются в разных ВУЗах, группах и т.д. Соответственно, свободное время для занятий дополнительного образования у них не совпадает. Было принято решение разработать автоматизированную систему, в которой студенты будут голосовать за удобные и не удобные для них дни для проведения занятий, в дальнейшем система, исходя из результатов голосования студентов, составит, наиболее, подходящее расписание.

Исходя из выше сказанного, была сформулирована цель данной выпускной квалификационной работы:

Объект исследования – автоматизированная система расписания для центра дополнительного образования.

Предмет исследования – применение современных информационных технологий и средств веб–разработки для создания автоматизированной информационной системы.

Целью данной выпускной квалификационной работы, является – повысить эффективность составление расписания диспетчером, путем автоматизации процесса.

Задачами квалификационной работы, в соответствие с целью, является:

а) подробно описать актуальность проблемы и предметную область и выявить основные проблемы;

б) провести анализ и подобрать удобную для студентов и преподавателей автоматизированную систему;

в) аргументирование экономической эффективности путем внедрение автоматизированной системы.

1 Анализ предметной области

1.1 Анализ работы центра дополнительного образования

Одним из первых этапов дипломного проекта, является выявление предметной области. Она играет роль в анализе, структурировании данных и позволяет классифицировать и формализовать информацию обо всех процессах данного исследования

Предметная область – совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей, для конкретного исследования это часть реального мира, представляющая интерес. Цель данной работы повысить эффективность составление расписания диспетчером, путем автоматизации процесса. Функция составления данного расписания всегда относилась непосредственно к преподавателю так как, центр дополнительного образования не относится к учебному отделу ВУЗа.

Преподаватель, который ведёт предметы дополнительного образования, должен выделить время в рабочей недели для проведения занятий, при этом это не должно препятствовать его основной работе. Таким образом, он сам обязан составить расписание, которое будет удовлетворять студентов и его самого. Для этого преподаватель просит от студентов информацию о том, когда у них есть свободное время для посещения занятий. После того когда каждый студент подаст информацию о своем свободном времени, преподавателю в ручном режиме приходится расформировывать студентов по группам исходя из их пожеланий.

Описанный процесс занимает ни один день, зачастую время на составление расписания растягивается до одной недели, что в современном мире считается нерациональным. В этом и заключается основная проблема составления расписания дополнительного образования.

Исходя из выше сказанного, мы видим активную взаимосвязь преподавателя и студентов, образующую потоки информации, т.е. перемещение информации от одного субъекта, т.е. преподавателя, к другому точнее к студентам. Представим потоки в виде модели изображенной на рисунке 1.

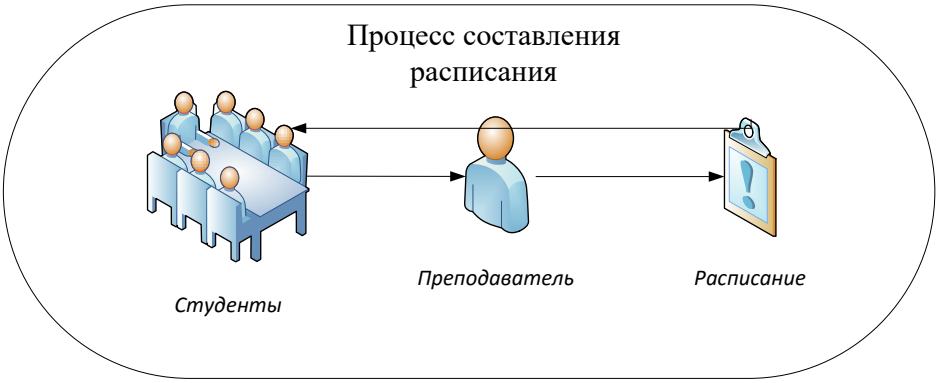

Рисунок 1 – Модель информационных потоков.

Проанализировав работу центра дополнительного образования, было решено автоматизировать процесс составления расписания, путем внедрения автоматизированной системы.

1.2 Обзор существующих решений проблемы составления расписания

В настоящий момент есть несколько программ, реализующих технологию составления расписания Представляем некоторые из них, наиболее востребованные.

Конфигурация «1С:ХроноГраф Расписание» – общая разработка фирм «1С» и «Хронобус» – разработана для автоматизации учебного процесса и построение расписания в первую очередь для всевозможных учреждений, дополнительного образования, повышения квалификации и переподготовки специалистов, а также данная система подходит для повышения эффективности

работы учебного отдела ВУЗа. На рисунке 2 изображен логотип данного программного обеспечения.

Рисунок 2 – Логотиа «1С: ХроноГраф Расписание»

«1С: ХроноГраф Расписание» предлагает:

а) Предоставить отчеты о периоде учёбы с учетом особенности устройства учебной функции определенного образовательного заведения на основе:

1) построения учебного курса;

2) деление учебного года на части учебного плана;

3) время звонков и перемен;

4) заполнение списка каникул и выходных;

5) построение отпусков и праздников.

Пример работы программы изображен на рисунке 3.

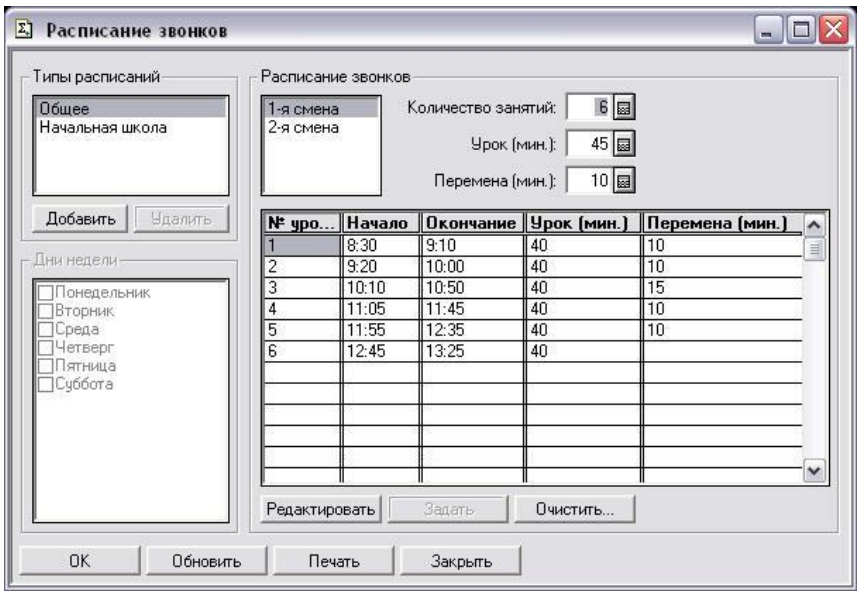

Рисунок 3 – Расписание звонков

б) Распределять учебную функцию всего ВУЗА или отдельных кафедр, применяя:

1) создание программы учебных курсов с установкой времени их подготовки;

2) формирование табеля обучающихся групп с учетом количества студентов;

3) построение подгрупп студентов в пределах одной группы по всем дисциплинам;

4) объединять потоки на общие лекции;

5) расчет нагрузки студентам на весь курс;

6) расчет финансовой нагрузки преподавателям изображено на рисунке 4;

7) разделение времени обучения в рамках одной кафедры между преподавателями на разные отрезки обучения в рамках одного учебного года.

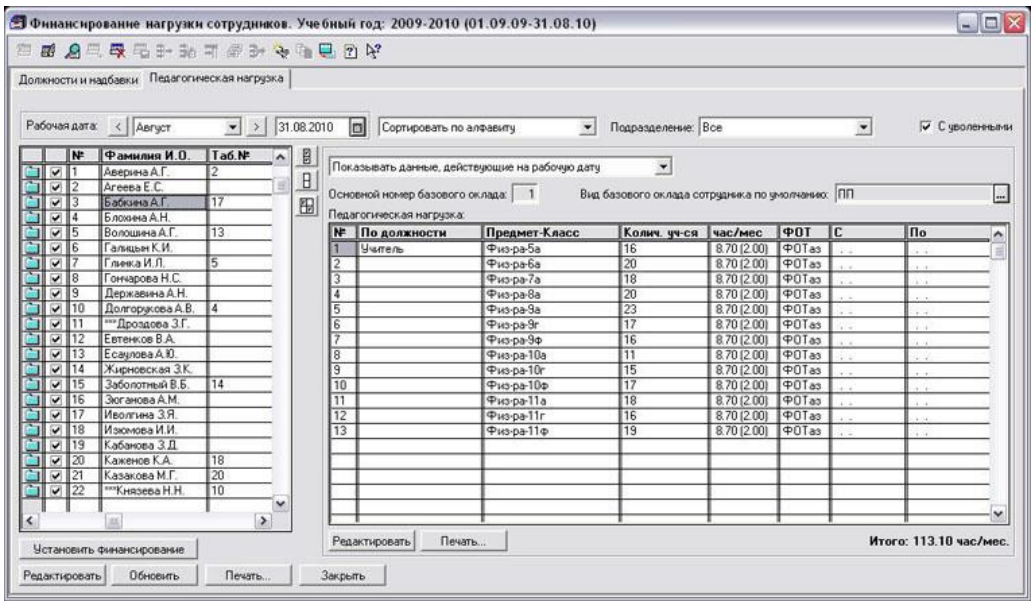

Рисунок 4 – Расчет финансовой нагрузки сотрудников.

в) Строить расписания преподавания на основе:

1) указания определенных выходных для преподавателей, и нерабочего времени кабинетов;

2) указание минимально возможного количества учебных дней преподавателя для всех недель обучения.

г) Сформировать оптимальное расписание, основываясь на:

1) назначения занятий преподавателям в режиме выбранной группы на текущую неделю;

2) использование эффективного алгоритма «первичного расчета» при составлении «черновых» вариантов расписания;

3) применения интерактивного «автоматического расчета» для составления «окончательного варианта» расписания на текущую неделю для любого количества учебных групп, с возможностями дальнейшего ручного редактирования;

4) назначения кабинетов (аудиторий) для проведения занятий с учетом их расположения, вместимости и численности учебных групп.

Выявив все достоинства данной разработки, можно сказать, что она передаёт целостность такого вопроса, как составление расписания. Но основным её недостатком, является то что, «1С: ХроноГраф Расписание» является дорогостоящим программным продуктом и судя по проведенным опросам она сложна в обращении. Также в рассматриваемой автоматизированной системе слишком много «лишних» функций для реализации основой цели данной работы, приобретение данного программного продукта будет считаться нецелесообразной.

Программа «Avtor» (АВТОРасписание)

Программа «АВТОРасписание» разработана для удобного и качественного формирования расписаний занятий и поддержания их в течение всего учебного года. Логотип данного программного продукта изображен на рисунке 5.

На данный момент разработано восемь основных модификаций программы для всевозможных образовательных учреждений, для средних общеобразовательных школ, лицеев и гимназий, колледжей, техникумов и профессиональных училищ, автошкол,музыкальных и художественных школ, и конечно для высших учебных заведений.

AVTOR способствует максимально упростить и автоматизировать процесс составления расписания диспетчером. Система имеет возможность легко строить, редактировать и распечатывать в виде удобных и наглядных документов:

– расписания занятий учебных групп;

– расписания преподавателей;

– расписание занятости аудиторий;

– учебные нагрузки.

Рисунок 5 – Система «АВТОРасписание»

AVTOR позволяет:

– формировать расписание без больших перерывов у студентов;

– строить расписание преподавателя без «окон»;

– учитывать требуемый диапазон дней/часов для учебных групп, для преподавателей и для аудиторий;

– ориентироваться на пожелания преподавателей;

– оптимально размещать занятия по кабинетам, обращая внимание, на специфику проводимого занятия. Пример работы программы на рисунке 6.

– строить расписание перемен;

– учитывать время перехода студентов между корпусами высшего учебного заведения;

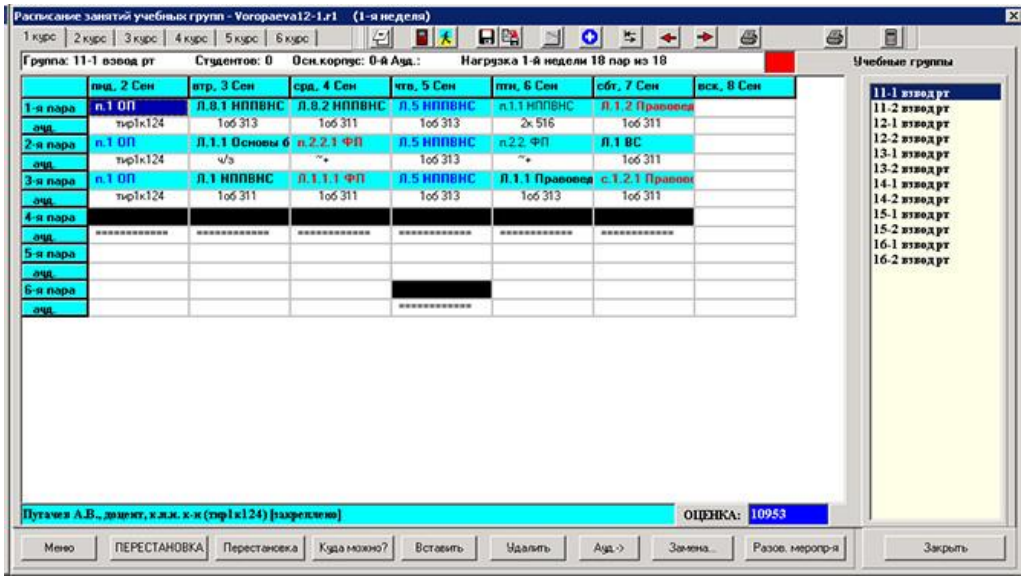

Рисунок 6 – Пример составленного расписания без окон

– легко объединять учебные группы в потоки при посещение общих лекций;

– строить расписание для факультативов и сдачи «долгов»;

– оптимизировать нагрузку студентов;

– вносить поправки в расписание;

– строить «запасные» варианты расписаний;

– самостоятельно перестраивать расписания при внесение новых записей в БД;

– вносить старые расписания в архив, копировать и отсылать по почте готовые варианты расписаний;

– выбрать замену при отсутствие преподавателя;

– самостоятельно анализировать расписание, исключая не состыковки;

– показывать расписания в виде стандартных форматов: txt, docx, xlsx;

– выгружать сформированные расписания во всемирную сеть.

Программа предназначена для автоматизации работы учебного отдела. Кроме расписания очного обучения программа позволяет формировать расписание для студентов, обучающихся на заочной форме, колледжа и других учебных структур вуза, обучающихся по другим графикам учебного процесса.

Работоспособность данного программного продукта оставляет двоякое впечатление, разработчики отнеслись к этой программе с полной отвественностью, рассмотрели много проблем и нюансов при составлении оптимального расписания. Очевидно, данная автоматизированная система окупит себя в течение недолгого времени , в виде трудозатрат сотрудников, но, тем не менее, она остается очень дорогим программным продуктом, не каждое образовательное учреждение может позволить такие затраты.

После анализа автоматизированных систем составления расписаний, стало ясно, что такие программы очень дорогостоящие, сделаны для крупных предприятий или сложны в обращении. Также для ведения таких программ требуется постоянная поддержка, которая требует приличных денежных затрат,

в частности придется иметь штат сотрудников для работоспособности системы. Но главный недостаток перечисленных программ, они совершенно не ориентированны на составление расписаний, путем рассмотрения свободного времени студентов, и преподователей.

После анализа вышеупомянутых программных продуктов, и выявления их основных преимуществ и недостатков, было принято решение о создание собственной автоматизированной системы, с простым и интуитивно понятным веб интерфейсом.

1.3 Формирование требований к автоматизированной системе

Выявив основные проблемы составления расписания, было решено создать сайт, на котором будут зарегистрированы преподаватели и студенты. У преподавателей будет возможность создавать голосование для студентов для составления расписания, исходя из своего свободного времени, и свободного времени студентов.

Полное наименование системы – «Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям»

Дав краткое название системы «Расписания для ЦДО» заказчиком были выдвинуты обязательные требования к системе.

Классы пользователей

В данной системе должно быть три класса пользователей такие как: преподаватели, студенты, администратор сервиса. Преподаватели должны иметь возможность создать голосование, затем исходя из голосов студентов сгенерировать расписание, также должна присутствовать возможность создание динамического PDF–файла с расписаниями для последующей печати. Студенты должны будут иметь возможность участвовать в голосование, которое создал преподавателей. После генерации расписания преподавателем, они должны иметь возможность просмотреть его, и также распечатать. Администратор это

пользователь, авторизованный на хостинге на котором будет находиться система, он будет иметь доступ непосредственно к базе данных, и полный доступ ко всем функциональным возможностям системы.

Авторизация

Авторизация должна проходить на первой странице сайта, также изначально должен быть выбор между входом в систему между студентом и преподавателем. В окне авторизации должны присутствовать такие поля как «Логин», «Пароль», и кнопки «Регистрация», «Сменить пароль» и отправка формы.

Создание голосование

Зайдя в систему, преподаватель должен иметь возможность создать голосование по какому–либо предмету для конкретной группы, с возможностью выбора количества занятий в неделю, и выбором дней недели в которые он сам может проводить занятие. Затем он отправляет форму и в базе данных должна создаться таблица, в которую будут записываться данные о голосование студентов.

Голосование

После того как преподаватель создал голосование, студент зайдя в систему, должен иметь возможность участвовать в этом голосование. В данном голосовании он видит список дней, которые включил в голосование преподаватель, и под каждым из дней должно быть поле с раскрывающимся списком, в котором есть три позиции: «удобно», «средне», «неудобно», под каждым из дней студент должен выставить одно из значений, и отправить форму.

Более подробно требования заказчика описаны в приложение А (техническое задание), при техническом проектировании были использованы следующие нормативно–технические документы:

а) Техническое задание по ГОСТ 34.602–89;

б) Информационная технология. Комплекс стандартов для автоматизированных систем. Автоматизированные системы. Стадии создания по ГОСТ 34.601–90;

в) Автоматизированные системы. Требования к содержанию документов по РД 50–34.698–90.

Очередность создания системы

Очередность создания системы описана в приложение А раздел 8 «Состав и содержание работ по созданию системы» технического задания.

После утверждения основных моментов в автоматизированной системе, был осуществлен переход к проектированию данного проекта.

SADT модель процесса

Метод SADT (IDEF0) (Structured Analysis and Design Technique) считается классическим методом процессного подхода к управлению. Основной принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес–процессами, а не организационно–штатной структурой. Функциональна модель IDEF0 – методология и графическая нотация, предназначенная для формализации и описания бизнес–процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматривается логические отношения между работами, а не их временная последовательность. Результатом применения метода SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы – главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация приходит в блок сверху, в этот момент входная информация, которая обрабатывается, показана с левой стороны блока, а результаты показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет

операцию, представляется дугой, входящей в блок снизу. Модель процесса «составления расписания» изображена на рисунках 7–8.

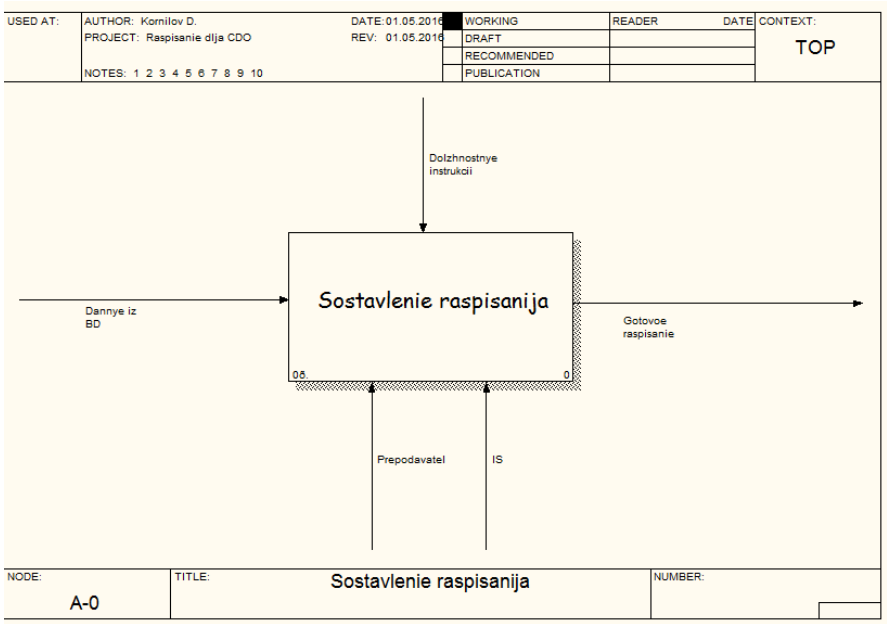

Рисунок 7 – Контекстная диаграмма процесса

Начальная контекстная диаграмма потоков данных содержит нулевой процесс с именем, отражающим деятельность организации, внешние сущности, соединенные с нулевым процессом посредством потоков данных.

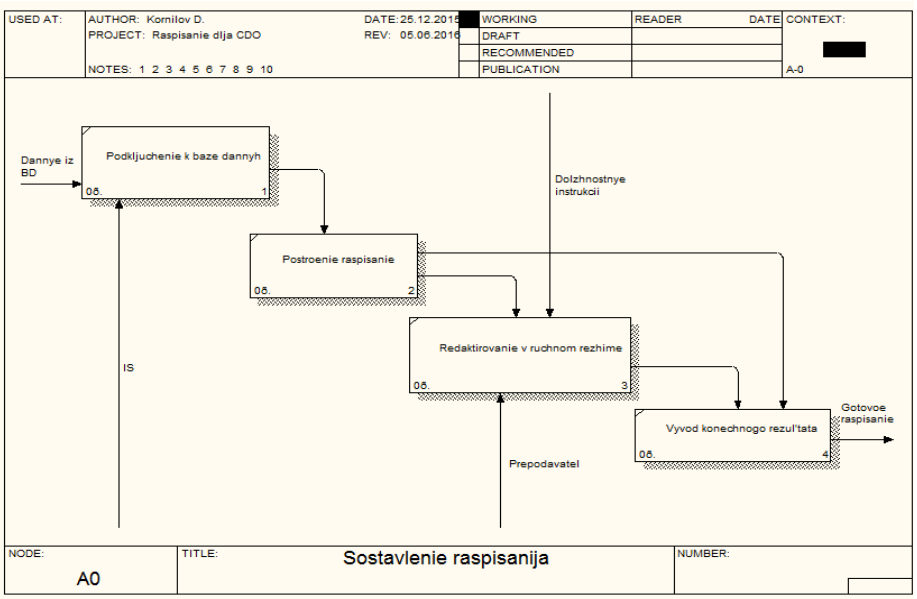

Рисунок 8 – Декомпозиция контекстной диаграммы

Представление системы с помощью Use Case–диаграммы

Основная задача каждой автоматизированной системы – создание программного продукта, помогающего упростить пользователю различного рода операции. В процессе создания, во–первых, необходимо определить все параметры, которым система должна соответствовать, какие функции она должна выполнять. Сами пользователи зачастую не могут конкретизировать свои требования, имея лишь чисто субъективное представление о своих желаниях. Поэтому эффективнее использовать для описания всех функций системы варианты использования: Use Case либо прецеденты.

Что же такое варианты использования? Это чёткое описания последовательных действий, осуществляемых системой в ответ на внешнее воздействие либо пользователей, либо иных программных систем. Варианты использования показывают, насколько функциональна система, с точки зрения получения итогового результата, такого, чтобы пользователь остался совершенно удовлетворён предоставленным программным продуктом.

Именно варианты использования помогают распределять функции по степени значимости готового результата. Диаграмма вариантов использования состоит из актеров, для которых система производит действие и собственно действия Use Case, которое описывает то, что актер хочет получить от системы. Актер обозначается значком человечка, а Use Case – овалом. Дополнительно в диаграммы могут быть добавлены комментарии.

При проектировании системы было выявлено три актера, такие как: Администратор, Преподаватель и Студент. На рисунке 9 изображена функциональная диаграмма автоматизированной системы «Расписание для ЦДО»

Первый актер системы это администратор, авторизованный на хостинге на котором находится система. Он имеет полный доступ к базе данных, может добавлять, редактировать, удалять таблицы и записи непосредственно из БД. Соответственно, он может провести голосования, и генерировать расписание.

Второй актер системы это преподаватель. Преподаватель это авторизованный пользователь, работник ЦДО, которому необходимо составить расписание для студентов. Заходя в систему под своим логином и паролем преподавателю доступно такое меню, как провести голосование для конкретной учебной группы. После того как студенты проголосуют, он может закрыть голосование, и произвести генерацию расписания, исходя из голосов студентов. Также ему доступно меню динамического составление PDF–файла с расписанием для последующей печати расписания.

Третьим актером системы является студент. Студент это авторизованный пользователь в системе, которой обучается по программе ЦДО. Используя свой логин и пароль, он имеет возможность зайти на сайт, где ему необходимо проголосовать за удобные и неудобные дни для проведения занятия. В дальнейшем он может просмотреть расписание и распечатать его.

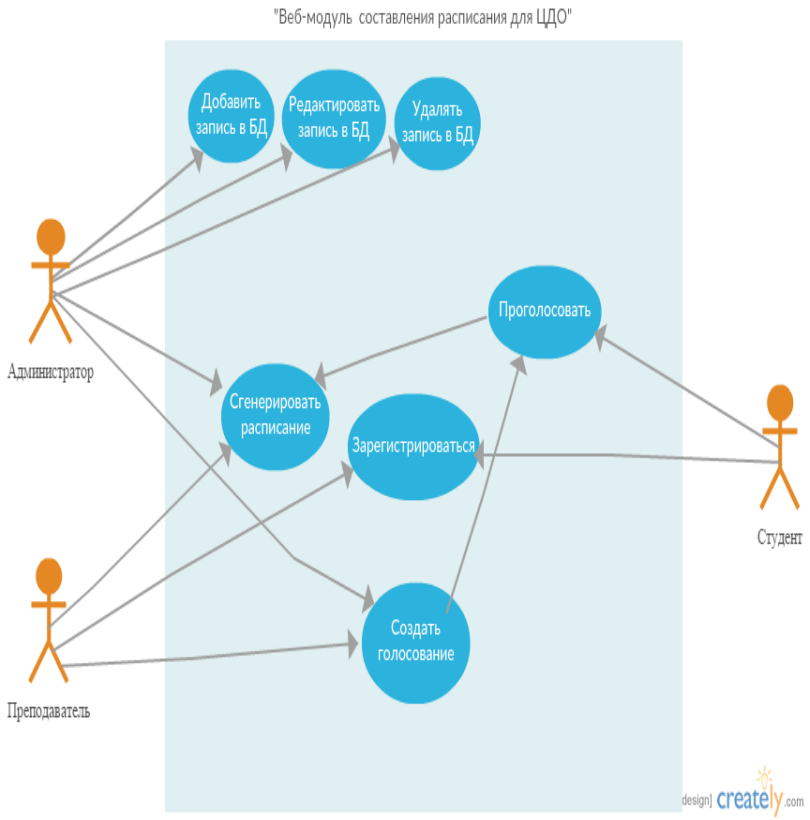

Рисунок 9 – Функциональная диаграмма автоматизированной системы «Расписание для ЦДО»

Прецедент К1: формирование расписания

Краткое описание:

Информационная система формирует расписания исходя из голосов студентов. Действующие лица этого прецедента – преподаватель и студент, изображены на рисунке 10.

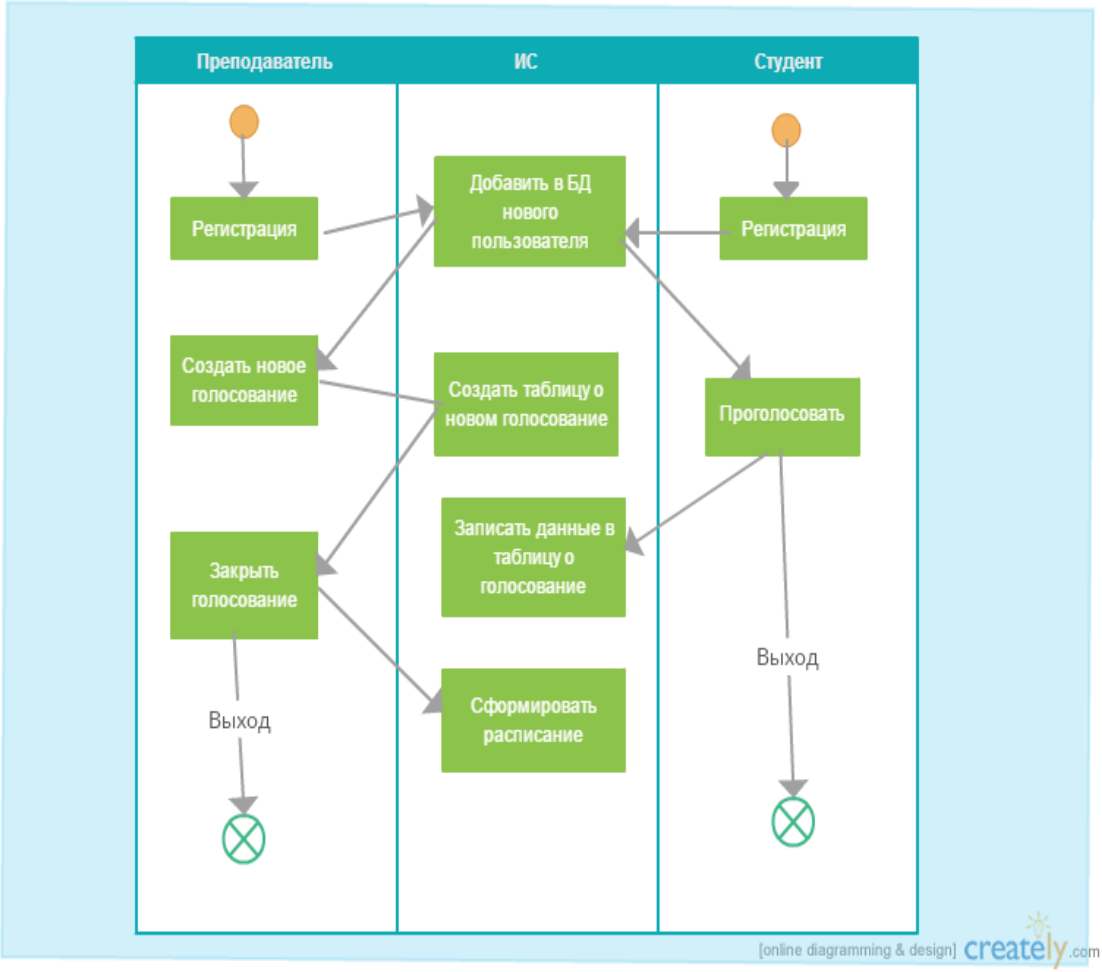

Рисунок 10 – Прецедент формирования расписания

Поток событий

Прецедент начинается с того, что преподаватель заходит в систему, ему нужно сформировать расписание:

а) преподаватель проходит процедуру регистрации;

б) система делает запись в базу данных о новом пользователе;

в) преподаватель создает голосование;

г) система создает в базе данных новую таблицу, в которую будут записываться голоса студентов;

д) студент заходит в систему и проходит процедуру регистрации;

е) студент участвует в голосование;

ж) система записывает данные о голосование студента в созданную таблицу;

з) преподаватель закрывает голосование;

к) система, исходя из голосов студентов, формирует расписание.

Постусловия:

При успешном окончании прецедента преподаватель получает готовое расписание. При неуспешном окончание – преподаватель получает отчет о нехватке данных для составления расписания.

1.4 Вывод к первому разделу

В ходе структурного анализа системы, были выявлены основные функциональные требования:

– регистрация пользователей;

– возможность создать голосование;

– возможность проголосовать;

– обработка голосов для генерации расписания.

Обзор аналогов показал, что существующие программы не способны выполнить все функции, которые требуются для центра дополнительного образования. Также они являются дорогостоящими программами, и сложны в управление, также для содержания и поддержки описанных программных продуктов требуется штат работников, все это будет считаться экономически невыгодно для центра дополнительного образования.

Для реализации всех требуемых функций, описанных в техническом задании (приложение А), необходимо спроектировать и реализовать

авторизированную систему формирования расписания для центра дополнительного образования.

2 Реализация автоматизированной системы.

2.1 Выбор инструментов для разработки автоматизированной системы

Хостинг

Первый вопрос, который возник при разработке системы, каким образом студенты и преподаватели смогут попасть на данный портал. Изначально, на этапе разработки было решено вести работу с помощью Веб–сервера Denwer 3, Denwer это программное обеспечение, которое содержит набор дистрибутивов Apache, PHP, MySQL, Perl и т.д. По мнению специалистов, данное программное обеспечение считается лучшим среди аналогов, и идеально подходит для разработки на «домашней машине», которая даже может быть не подключено к мировой паутине, на рисунке 11 изображен запуск данного программного продукта.

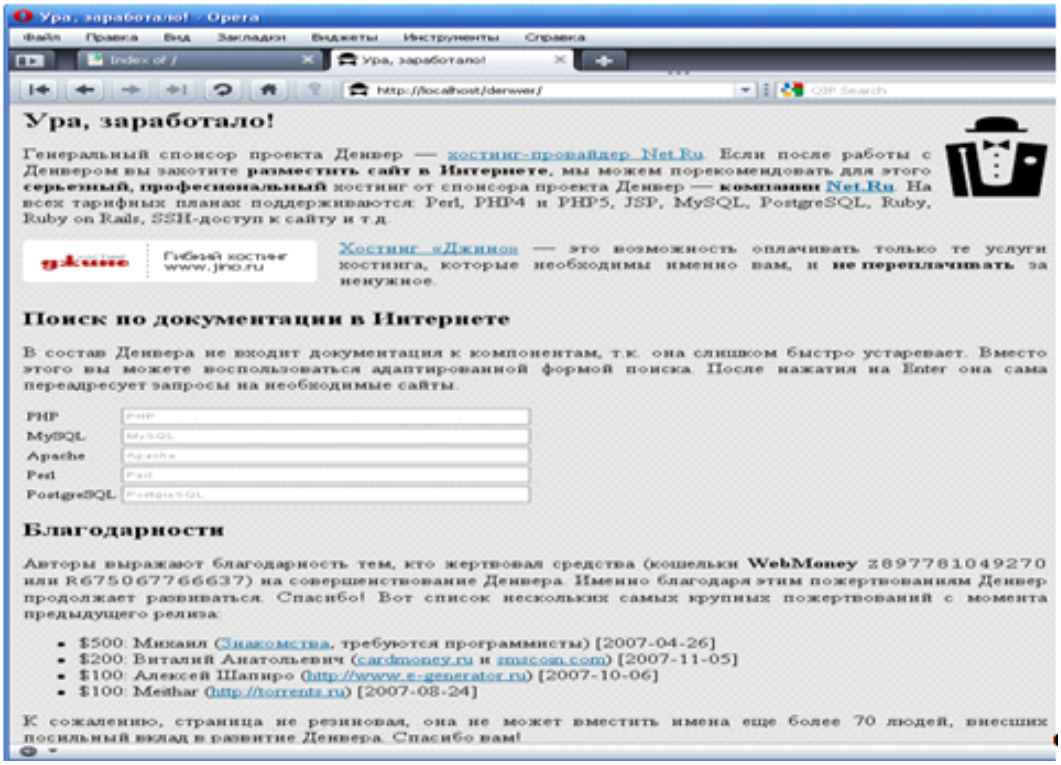

Рисунок 11 – Запуск программы Denwer

Но в дальнейшем на этапе тестирование было принято решение о переносе ИС на реальный хостинг. После недолгих поисков бесплатного хостинга был выбран сайт https://beget.ru/. Компания «Beget» является одним из крупнейших хостинг–провайдеров России. Доверие клиентов обусловлено проверенной временем технической поддержкой, надежностью и развитием собственных программных продуктов. Данная компания занимается предоставлением бесплатного хостинга с теми же возможностями, что и предоставляет Denwer. Пройдя простейшую процедуру регистрации на сайте, нам стал доступен удобный функционал, который изображен на рисунке 12 . Общая информация о предоставленном домене на рисунке 13.

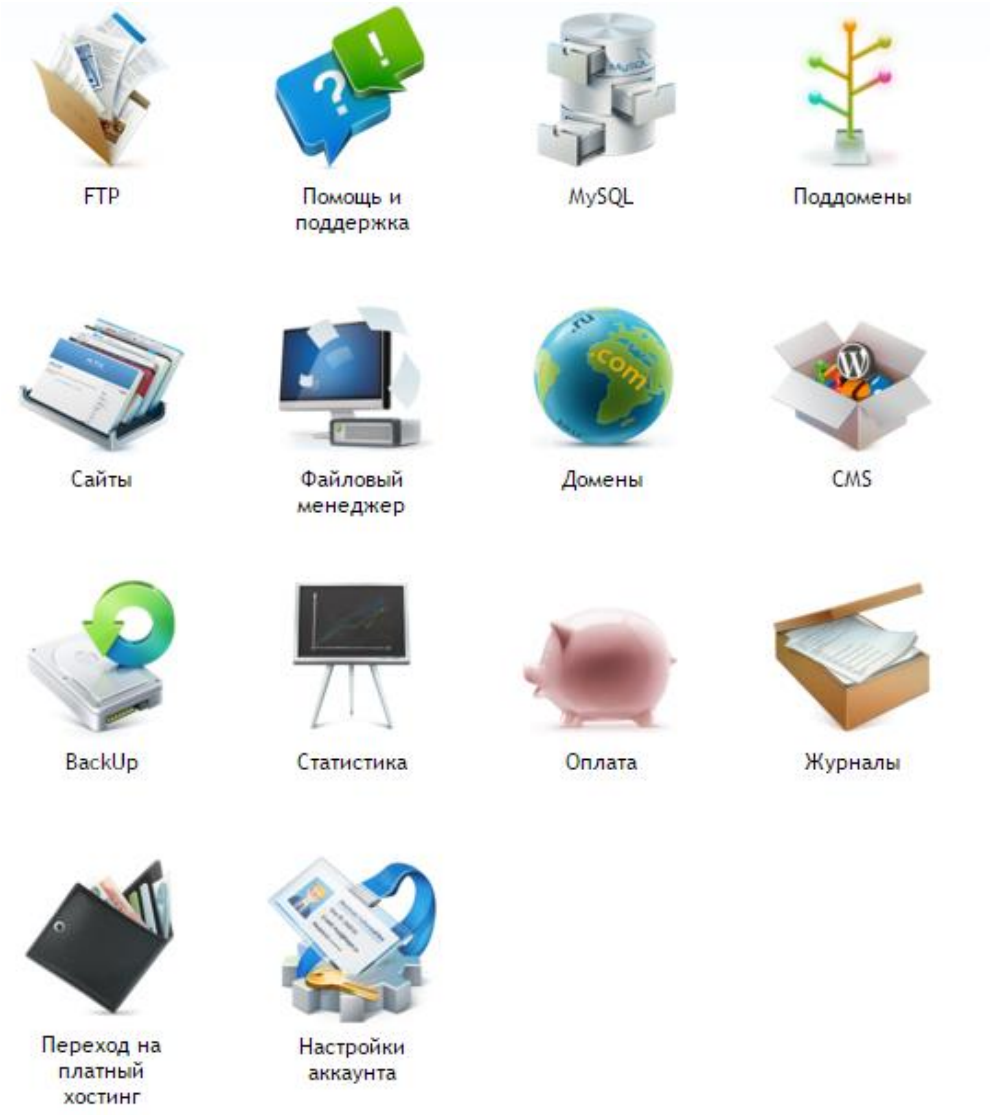

Рисунок 12 – Функционал хостинга «Beget»

Общая информация:

Имя пользователя, ID	d91304zf (339793)
Тарифный план	FreeHosting
Разрешенная нагрузка	10 CP Посмотреть
Разрешенная нагрузка MySQL	500 CP
Сервер	free2.beget.com
IP адрес	87.236.19.14
Сайты	1 / 1
Домены	1 / неогр.
Базы данных MySQL	1 / 1
Размер баз данных	1 Мб.
Дисковое пространство	1 Мб / 1000 Мб (1%)
Лимит файлов	69 / 25000
FTP-аккаунты	0 / неогр.

Рисунок 13 – Общая информация о предоставленном домене

С помощью внутреннего файлового менеджера мы смогли разместить файлы интернет–страниц нашей ИС на веб–сервисе «Beget», таким образом, наша ИС получила выход в сеть Internet, и стала доступна по адресу http://d91304zf.bget.ru/.

Выбор среды программирования

Для реализации поставленной задачи нужно было определиться со средой программирования. Первым рассматриваемым языком являлся PHP, логотип изображён на рисунке 14. Главным фактором выбора языка PHP является практичность. Практический характер PHP обусловлен пятью важными характеристиками: традиционностью; простотой; эффективностью; безопасностью; гибкостью. Также стоит сказать еще одна «характеристику», которая делает PHP очень привлекательным: он распространяется абсолютно бесплатно.

Рисунок 14 – Изображение логотипа языка «PHP»

Код PHP очень похож на тот, который встречается в обычных программах на C или Pascal. Это очень сильно снижает издержки при изучении PHP. PHP – язык, сочетающий лучшие стороны Perl и Си и специально ориентированный на работу в веб–пространстве, язык с универсальным и понятным синтаксисом. PHP – язык, который легко встраивается в html –код страниц, которые, в свою очередь будут корректно обрабатываться PHP – интерпретатором. Мы можем привлекать PHP для формирования HTML–документов, избавившись от множества вызовов внешних сценариев. Большое разнообразие функций PHP избавят вас от написания многострочных пользовательских функций на C или Pascal . PHP имеет развитую поддержку баз данных. «Движок» PHP является транслирующим интерпретатором. Такое устройство «движка» PHP позволяет обрабатывать сценарии с очень высокой скоростью. По общему мнению, большинство PHP–сценариев обрабатываются быстрее аналогичных им программ, написанных на Perl. Однако чтобы не делали разработчики PHP, откомпилированные исполняемые файлы будут исполняться значительно быстрее – в десятки, а иногда и в сотни раз. Но производительность PHP вполне достаточна для создания вполне серьезных веб–приложений. PHP предоставляет в распоряжение разработчиков и администраторов гибкие и эффективные средства безопасности, которые условно делятся на две категории: средства системного уровня и средства уровня приложения. PHP в процессе работы может оперировать достаточно большим количеством типов данных. Типы данных, используемые в PHP, изображены на рисунке 15.

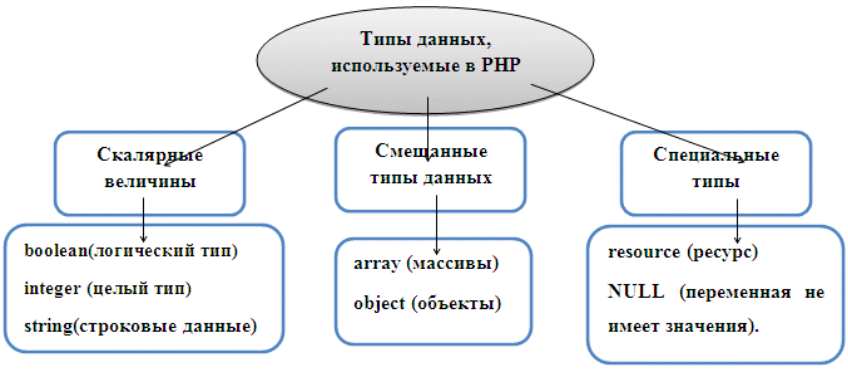

Рисунок 15 – Типы данных, используемые в PHP

Но, не смотря на все преимущества, PHP имеет ряд недостатков. На PHP нельзя осуществить написание десктопных приложений или системных компонентов. Еще одним немаловажным недостатком является то, что передача объектов происходит не по ссылке, а по значению, также в приложениях написанных существуют проблемы с безопасностью.

Вторым рассматриваемым языком был Ruby, логотип изображён на рисунке 16. Ruby является динамическим и императивным, объектно–ориентированным языком программирования. Он имеет абсолютно те же преимущества, что и PHP, и в некоторых моментах он даже превосходит его. Например, возможности языка можно расширить путем подключения библиотек написанных на C, имеются дополнительные возможности по обеспечению информации, также в Ruby есть встроенный отладчик. Но и не обошлось без недостатков, для изучения языка недостаточно начального уровня, а информационных ресурсов по данному языку очень мало. Но главным его недостатком является то, что он менее производителен по сравнению с другими языками, которые применяются в веб–программирование.

После анализа двух языков были выявлены их преимущества и недостатки. Оба языка полностью подходят для реализации данной работы, но для разработки был выбран язык PHP, т.к. он является более простым в обучение, и данный язык знаком нам из учебного курса «Интернет технологии»

Рисунок 16 – Изображение логотипа языка «Ruby»

Выбор СУБД

База данных представляет собой совокупность данных. Эти данные могут быть абсолютно разные – от простого списка продуктов, до перечня экспонатов в музее. Для обработки, записи и выборки структурированных данных требуется система управления базой данных (СУБД). Примером такой программы является MySQL, логотип изображён на рисунке 17.

MySQL – это одна из самых популярных и самых распространенных систем управления базами данных в Интернете. Она не предназначена для работы с большими объемами информации, но ее применение идеально для интернет сайтов, как небольших, так и достаточно крупных.

MySQL отличатся хорошей скоростью работы, надежностью, гибкостью. Работа с ней, как правило, не вызывает больших трудностей. Поддержка сервера MySQL автоматически включается в поставку PHP. Также MySQL поддерживает большое количество типов данных. MySQL является легко масштабируемой СУБД.

Рисунок 17 – Изображение логотипа «MySQL»

Одним из главным определяющим фактором является ее бесплатное распространение. MySQL распространяется на условиях общей лицензии GNU (GPL, GNU Public License).

MySQL является системой клиент–сервер, которая содержит многопоточный SQL–сервер, обеспечивающий поддержку различных вычислительных машин баз данных, а также несколько различных клиентских программ и библиотек, средства администрирования и широкий спектр программных интерфейсов (API).

Структура MySQL трехуровневая: базы данных – таблицы – записи. Базы данных и таблицы MySQL физически представляются файлами с расширениями frm, MYD, MYI. Логически – таблица представляет собой совокупность записей. А записи – это совокупность полей разного типа. Имя базы данных MySQL уникально в пределах системы, а таблицы – в пределах базы данных, поля – в пределах таблицы. Один сервер MySQL может поддерживать сразу несколько баз данных, доступ к которым может разграничиваться логином и паролем. Зная эти логин и пароль, можно работать с конкретной базой данных. Например, можно создать или удалить в ней таблицу, добавить записи и т. д. Обычно имя–идентификатор и пароль назначаются хостинг провайдерами, которые и обеспечивают поддержку MySQL для своих пользователей.

Также примером СУБД является программ PostgreSQL, логотип

изображён на рисунке 18. PostgreSQL является профессиональной программой, и максимально соответствует стандартам SQL. Она также как и MySQL является открытым ПО. PostgreSQL поддерживает на порядок больше типов данных, чем MySQL. Но главным ее преимуществом является поддержка объектности, соответственно она не только реляционная, а еще и объектно–ориентированной, и поддерживает такие вещи как наследование, полиморфизм, инкапсуляция и т.д. Но, не смотря на очевидные преимущества PostgreSQL, выбор был в сторону MySQL. Т.к. простейшие операции в PostgreSQL выполняются намного дольше, и существуют проблемы с поиском хостинга, который поддерживает PostgreSQL. Также проблемой является непопулярность данной СУБД.

Рисунок 18 – Изображение логотипа СУБД «PostgreSQL»

Администрирование СУБД

После того когда было определено, что СУБД будет MySQL, нужно выбрать систему для администрирования. Одной из таких систем является phpMyAdmin, изображение логотипа на рисунке 19. phpMyAdmin дает администратору через браузер осуществлять управлением сервером MySQL, изменять содержимое таблиц и баз данных. Приложение написано на языке PHP, и является открытым бесплатным программным обеспечением.

Рисунок 19 – Изображение логотипа «phpMyadmin»

Главными преимуществами phpMyAdmin является то, что приложение в большинстве случаях не требует знания языка SQL, вследствие этого работа с БД не составит труда даже пользователю, который поверхностно знаком с SQL. Простой и понятный интерфейс принес phpMyAdmin большую популярность среди разработчиков, а также большинство российских провайдеров используют эту систему в качестве панели управления базами данных. Именно это приложение и предоставила нам вышеупомянутая компания «Beget», поэтому альтернатива в выборе администрирования СУДБ отсутствует.

Язык гипертекстовой разметки документов HTML

Создание веб – страниц осуществляется с помощью языка разметки гипертекста (Hyper Text Markup Language – HTML), изображение логотипа на рисунке 20.

Язык HTML описывает отображение форматирования шрифтов и абзацев, табличное представление, обработку форм, гипертекстовые ссылки на другие документы, поведение веб–страницы в программе–обозревателе при перемещении над ней указателя мыши.

Рисунок 20 – Изображение логотипа языка «HTML»

Самыми важными тегами языка HTML являются теги формирования и выполнения гипертекстовых ссылок. С любым фрагментом текста (обычно выделенным), значком, картинкой с помощью такого тега можно связать гиперссылку – в общем случае другой объект (документ, рисунок, мультимедийный компонент). Щелчком левой клавиши мыши на выделенном фрагменте инициируется запрос на доставку этого другого объекта, и, если объект на данном компьютере имеется, то он воспроизводится или отображается. Открывшийся на экране объект может быть в свою очередь гипертекстовым и по его гиперссылкам можно получить доступ к связанным объектам.

Каскадная таблица стилей CSS

Основным понятием CSS является стиль – т. е. набор правил оформления и форматирования, который может быть применен к различным элементам страницы, логотип изображение на рисунке 21. В стандартном HTML для присвоения какому–либо элементу определенных свойств (таких, как цвет, размер, и т. п.) приходилось каждый раз описывать эти свойства, даже если на одной страничке должны располагаться 10 или 100 таких элементов, ничуть не отличающихся один от другого. Мы должны были десять или сто раз вставить один и тот же кусок HTML–кода в страницу, увеличивая размер файла и время загрузки на компьютер просматривающего ее пользователя.

Рисунок 21 – Изображение логотипа CSS

CSS действует более удобным и экономичным способом. Для присвоения какому–либо элементу определенных характеристик мы должны один раз описать этот элемент и определить это описание как стиль, а в дальнейшем просто указывать, что элемент, который мы хотим оформить соответствующим образом, должен принять свойства стиля, описанного вами.

Более того, мы можем сохранить описание стиля не в тексте вашей странички, а в отдельном файле – это позволит использовать описание стиля на любом количестве страниц, а также изменить оформление любого количества страниц, исправив лишь описание стиля в одном (отдельном) файле.

Кроме того, CSS позволяет работать со шрифтовым оформлением страниц на гораздо более высоком уровне, чем стандартный HTML, избегая излишнего утяжеления страниц графикой.

И так подытожим:

– средой разработки является язык программирование PHP версии 5.2;

– система управления базами данных является MySQL версии 5.0;

– для администрирования системой управления базой данных; используется приложение phpMyAdmin версии 3.0;

– для верстки страниц используется язык разметки HTML версии 5.0;

– для определения стиля страниц используется таблица стилей CSS версии 3.0.

Утвердив выбор инструментов для разработки, можно переходить к реализации данного проекта.

2.2 Особенности программной реализации автоматизированной системы

Создание таблиц в БД

В первую очередь, при создании авторизированной системы требуется создание информационной базы, с которой в дальнейшем будет работать эта система. Преимуществами средств PHP+MySQL перед другими аналогичными программами являются надежность, интуитивно понятный интерфейс, хорошо организованная структура, и безопасность.

Для создания базы данных разрабатываемой автоматизированной системы ранее была выбрана система управления базами данных MySQL. С помощью конструктора в MySQL создаются необходимые для реализации данного проекта таблицы.

Для хранения данных в базе используются следующие таблицы, изображённые на рисунке 22.

Рисунок 22 – Список таблиц, используемый в системе

В таблице «users» хранится информация о зарегистрированных в системе преподавателях и студентах. Структура таблицы «users» представлена на рисунке 23.

Рисунок 23 – Структура таблицы «users»

Запись в этой таблице появляется после регистрации нового пользователя в системе. Более подробно структура изображена на рисунке 24.

id	name	fam	ot	group	email	login	pass	reg_date
14	Иванов	Иван	Иванович	КИ12-116	ivanov@mail.ru	ivanov	3f74ed1b90de7d06a51891228750fcb1	2016-06-11 13:06:19
13	Анжелика	Шалаева	Сергеевна	КИ12-106	lera.gg@yandex.ru	LIKA	2467d3744600858cc9026d5ac6005305	2016-05-19 13:49:28
11	Корнилов	Дмитрий	Александрович	pers	krova_777@mail.ru	krova777	970ef371747b0239549b20bc3f658d0e	2016-05-19 13:23:18
6	Сергей	Василовский	Сергеевич	КИ12-116	gabriel666_23@mail.ru	Sergey	e10adc3949ba59abbe56e057f20f883e	2016-05-17 13:35:49
5	Корнилов	Дмитрий	Александрович	КИ12-116	gabriel666-23@yandex.ru	kornilov	2467d3744600858cc9026d5ac6005305	2016-05-17 13:32:08
10	Виденин	Сергей	Александровича	pers	videnin3@yandex.ru	Sergey9	e10adc3949ba59abbe56e057f20f883e	2016-05-17 13:40:50
15	Петр	Петров	Петрович	КИ12-116	PETROV@MAIL.RU	PETROV	9ea5e6f10d48803ae38499c0d5e6d93f	2016-06-11 13:08:29

Рисунок 24 – Зарегистрированные пользователи в системе

В таблице «groups» хранится список учебных групп центра дополнительного образования. Структура таблицы «groups» представлена на рисунке 25.

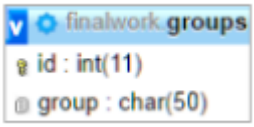

Рисунок 25 – Структура таблицы «groups»

Управление этой таблицей осуществляется непосредственно из базы

данных, соответственно добавлять учебные группы имеет возможность только администратор. Пример записанных групп изображен на рисунке 26.

id	group
1	КИ12-116
2	КИ12-106
3	КИ12-096
4	КИ12-086

Рисунок 26 – Учебные группы центра дополнительного образования

В таблицу «cons» записывается информация о создание нового голосования. Структура таблицы «cons» представлена на рисунке 27.

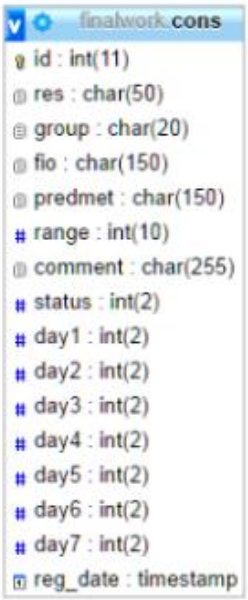

Рисунок 27 – Структура таблицы «cons»

Добавлять запись в эту таблицу может только преподаватель, так как студент не может создать новое голосование. Пример заполненной таблицы на рисунке 28.

id	res	group	fio	predmet	range	comment	status	day1	day2	day3	day4	day5	day6	day7	reg_date
1	res_cons_1	КИ12-11б	Виденин Сергей Александровича	Базы Данных	3		3	-5	-5	2	5	-5	0	0	2016-05-19 14:50:12
2	res_cons_2	КИ12-10б	Виденин Сергей Александровича	Информационные системы	2		3	0	0	-5	-5	5	3	0	2016-05-19 14:51:25
3		КИ12-11б	Виденин Сергей Александровича	Информационные системы	3		3	-5	-5	-5	-5	-5	-5	-5	2016-05-31 12:36:36
4	res_cons_4	КИ12-11б	Виденин Сергей Александровича	Информатика	2		3	-5	0	0	1	0	-5	0	2016-06-11 12:50:53

Рисунок 28 – Таблица голосований

Таблицы «res_cons_N» является временной, они создаются для записи голосов студентов на время голосование. Структура таблиц «res_cons_N» представлена на рисунке 29

Рисунок 29 – Структура таблицы «res_cons_N»

Данные таблицы создаются одновременно с записями в таблице «cons» о новом голосование. После того, когда преподаватель закрывает голосование и создает расписания, таблица удаляется. Пример с голосованием студентов на изображение 30.

id	fio	day1	day2	day3	day4	day5	day6	day7	count
1	Петров Петр Петрович	2	2	1	0	0	0	0	1
2	Иван Иванов Иванович	1	1	2	2	0	2	0	2
3	Дмитрий Корнилов Александрович	0	2	1	2	0	2	0	3

Рисунок 30 – Результаты голосования студентов

После того как все таблицы были созданы с помощью системы управления данных MySQL, а также приложения phpMyAdmin, можно переходить к визуальной части нашего проекта, а точнее создания веб–интерфейса.

Графическое представление системы

В настоящее время уже практически не осталось таких систем, которые работают с помощью рукописного ввода команд. Примером такой системы можно назвать «MS–DOS», действия в которой, осуществлялись при помощи определённых команд. Обычному пользователю без специальных навыков было невозможно пользоваться данной системой. Что же происходит сейчас? В настоящее время любой компьютер, телефон, планшет и т.д. поддерживает простой, и в тоже время функциональный графический интерфейс, который обеспечивает взаимодействия между пользователем и системой, используя при этом всевозможные диалоговые окна и элементы управления. Графический интерфейс пользователя автоматизированной системы «Расписание для ЦДО » представлен на рисунке 31.

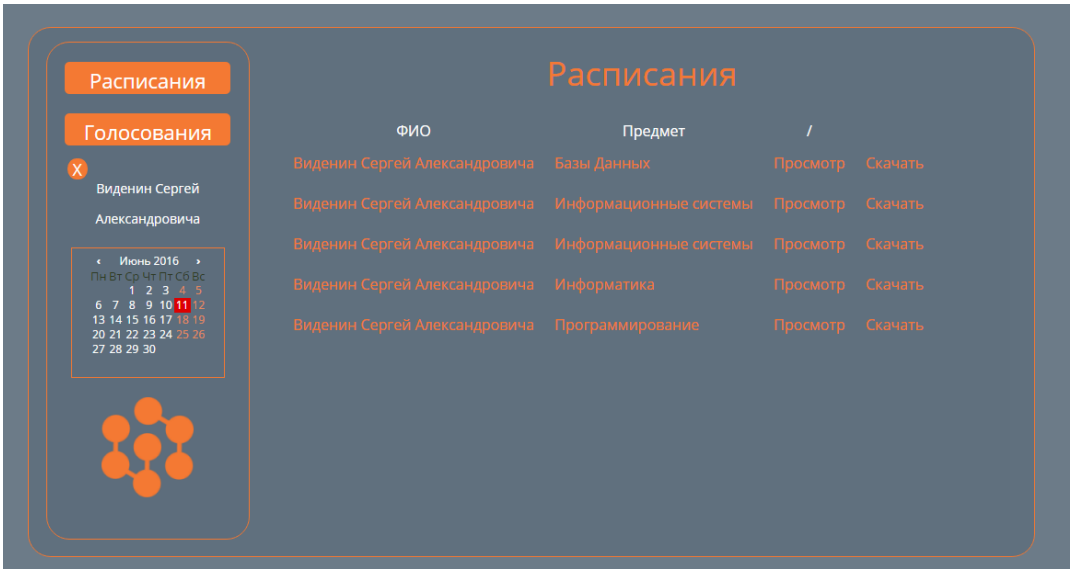

Рисунок 31 – Графический интерфейс авторизированной системы «Расписания для ЦДО»

При реализации данного графического веб интерфейса авторизированной системы, использовался язык разметки HTML, верстка осуществлялась при помощи «div'ов». Дизайн сайта разрабатывался при помощи таблицы стилей CSS.

Система разрабатывалась с простым и легким в понимание интерфейсом, чтобы ни у преподавателя, ни у студента не возникало проблем при работе с сайтом. Оформление разрабатывалось в консервативном ключе, поэтому в дизайне использовались преимущественно серые и оранжевые тона.

При разработке были учтены все пожелания клиента, в разработанной системе присутствуют все функции, которые были изложены в техническом задание. Теперь можно перейти непосредственно к тестированию реализованной системы.

2.3 Тестирование автоматизированной системы

Регистрация и авторизация

При входе в автоматизированную систему пользователю на первой странице нужно сделать выбор, нажав на кнопку «Преподаватель», либо на кнопку «Студент», рисунок 32.

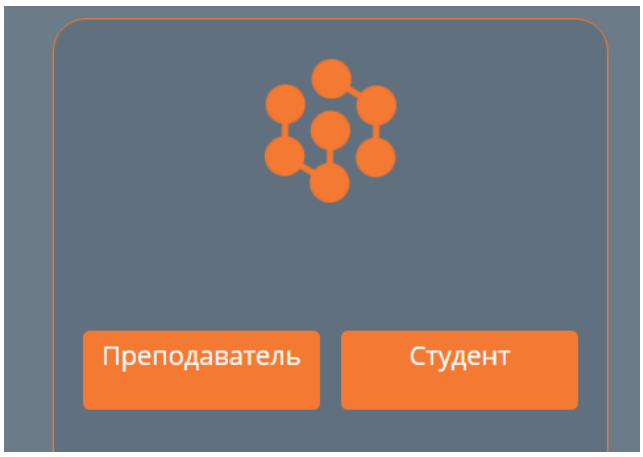

Рисунок 32 – Начальная страница

Далее, перед пользователем открывается окно авторизации, рисунок 33, которое содержит два поля «Имя пользователя» и «Пароль» для преподавателя и студента это два идентичных окна. Но в системе это два разных php файла, которые предоставляют разные права для доступа к системе. Соответственно, преподаватель не сможет авторизоваться на странице студента, также и студент не сможет пройти авторизацию на странице преподавателя. Для преподавателя файл авторизации это auth_pers.php, а для студента auth_stud.php

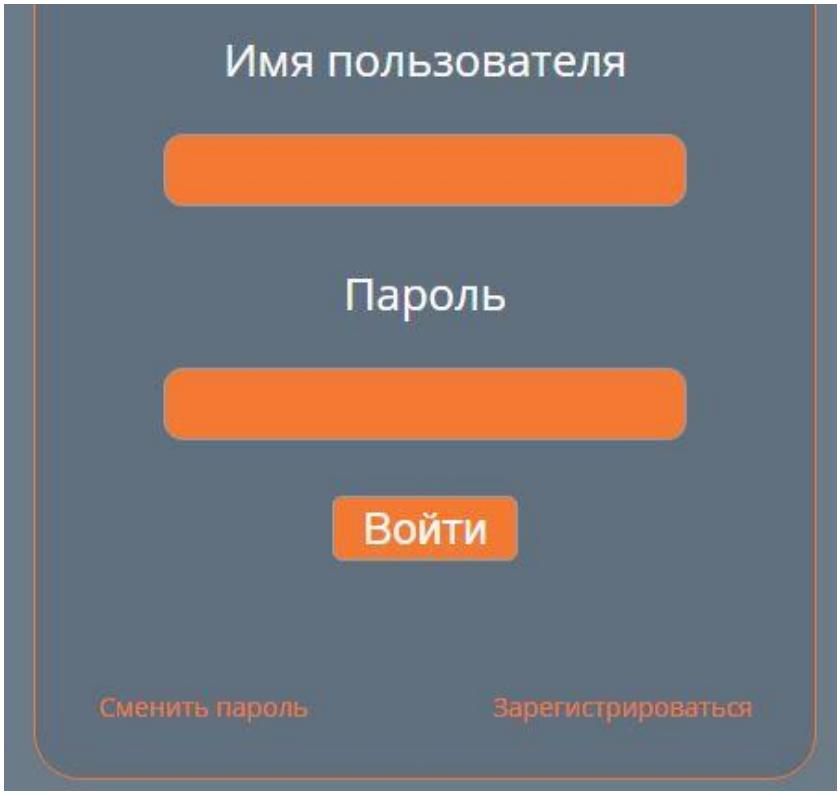

Рисунок 33 – Страница авторизации для студента и преподавателя

Также на странице авторизации присутствуют две кнопки, «Сменить пароль», и «Зарегистрироваться»

Нажав кнопку регистрации, перед пользователем открывается новое окно, рисунок 34. В этом окне ему предлагается заполнить поля «Имя», «Фамилия», «Отчество», учебную группу, «e–mail», «Логин», «Пароль». Все эти данные записываются в таблицу «users». Регистрация преподавателя отличается отсутствием выбора учебной группы. После отправки формы, страница

обновляется, и открывается окно авторизации. Далее пользователь вводит свой логин и пароль, и таким образом авторизуется в системе. Если пользователь ввёл неправильный логин и пароль, то система оповестит пользователя, о неправильно введённых данных.

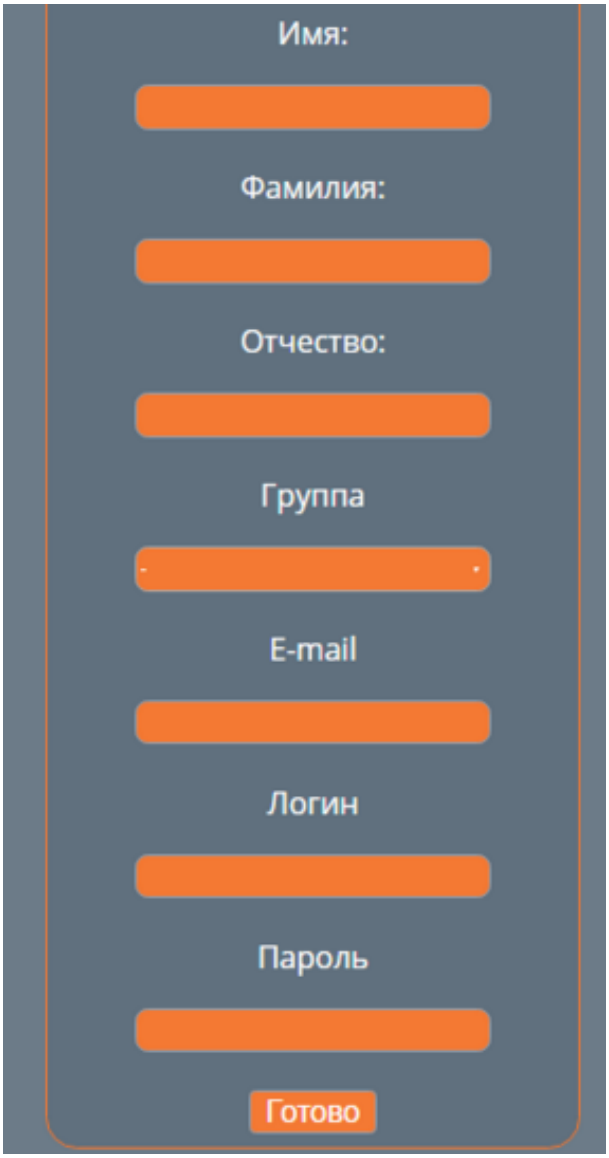

Рисунок 34 – Окно регистрации студента

Создание голосования преподавателем

После авторизации и студенту и преподавателю доступно меню сайта, рисунок 35, в котором доступно два пункта, «Расписание» и «Голосование», но когда преподаватель заходит в меню голосование, ему доступна функция

создания голосования, рисунок 36. В окне создания голосования имеются такие поля как имя преподавателя, название предмета, ползунок для выбора количества занятий в неделю, список дней недели из которых преподаватель выбирает дни удобные ему для проведения занятия, и выбор группы, для которой проводится голосование. Перечисленные поля являются обязательными к заполнению, также есть поле «Комментарий», но оно не является обязательным к заполнению.

Рисунок 35 – Меню сайта

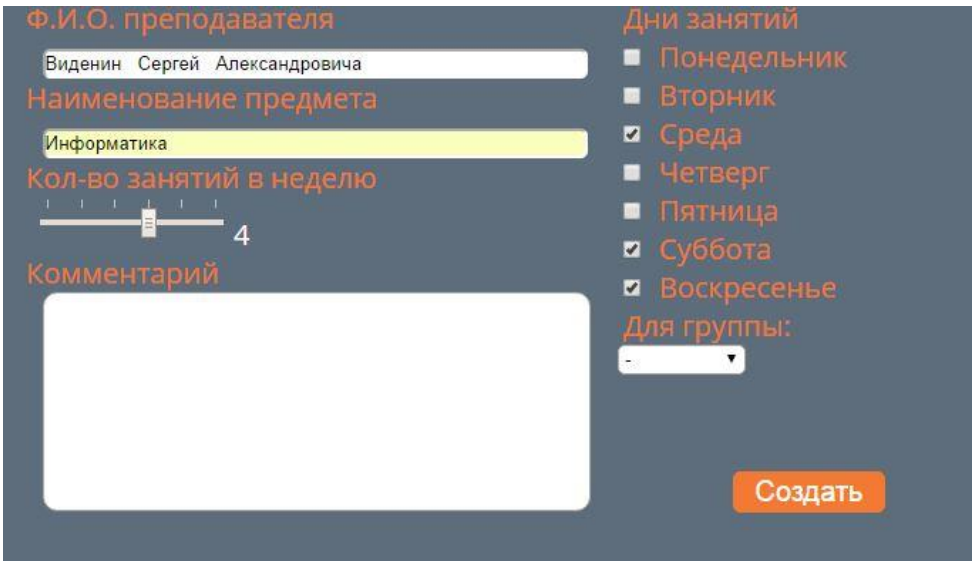

Рисунок 36 – Окно создания голосования

После отправки формы, в таблице «cons» производится запись о создание голосования, одновременно с этой записью создается таблица «res_cons_N», в которую будут записываться голоса студентов.

Голосование студентом

После того как преподаватель создал голосование, студент зайдя в систему, и выбрав пункт меню «Голосование», откроется окно со списком активных голосований, изображенный на рисунке 37.

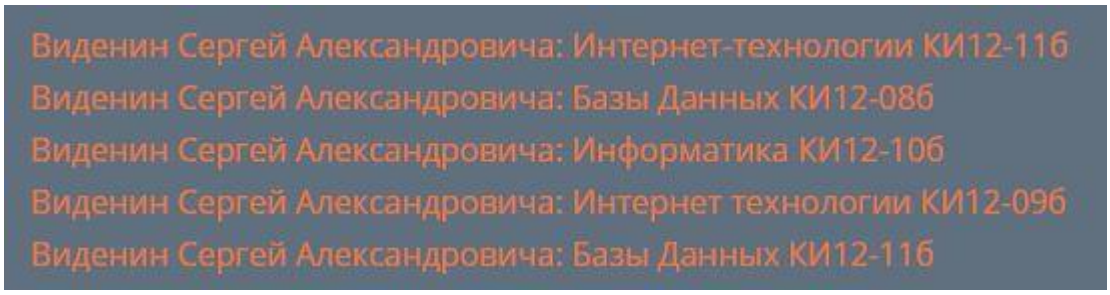

Рисунок 37 – Список голосований

Нажав в списке на голосование для своей учебной группы, студенту открывается окно с голосованием, изображенное на рисунке 38.

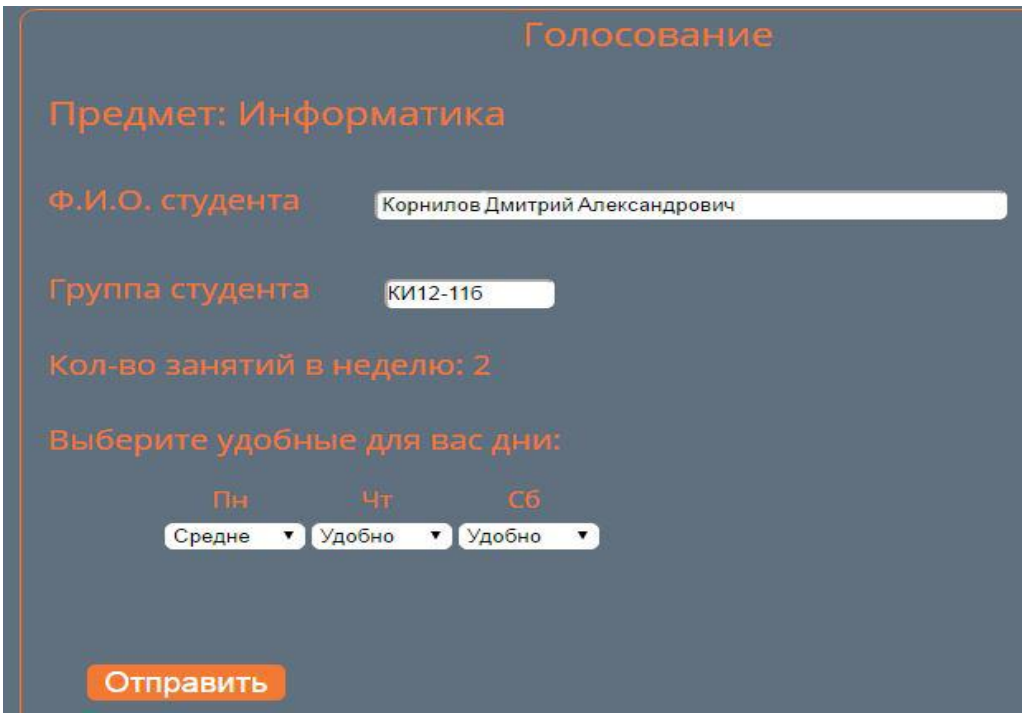

Рисунок 38 – Окно голосования для студента

В этом окне имеются поля, Ф.И.О. студента, который голосует, группа студента, далее система указывает количество занятий в неделю, под этой записью расположен список дней недели, которые указал преподаватель при создание голосования. Под каждым из этих дней имеется раскрывающийся список со значениями приоритета. Студенту каждому из дней нужно присвоить один из трех приоритетов, «удобно», «неудобно», «средне». После отправки им формы, в таблицу «res_cons_N» запишутся результаты его голосования.

Преподаватель в свою очередь может просматривать результаты голосования студентов. Они ему предоставляются в виде массива, рисунок 39, в котором значение «0» это означает, что либо день не участвует в голосование, либо студент указал, что этот день не неудобен для занятия, второе значение, это «1», что обозначает значение приоритета «средне», третье значение «2», обозначающее приоритет «удобно». Также система показывает преподавателю число проголосовавших студентов.

ФИО	Пн	Вт	Ср	Чт	Пт	Сб	Вс
Иван Иванов Иванович	0	0	2	2	0	1	0
Петров Петр Петрович	0	0	0	2	0	2	0
Дмитрий Корнилов Александрович	0	0	2	1	0	2	0

Количество голосов: 3

Рисунок 39 – Результаты голосования студентов

Генерация расписания

После того когда все студенты группы проголосовали преподаватель может закрыть голосование, и система исходя из голосов студентов составит, наиболее подходящее, расписание для учебной группы. Зайдя в пункт меню «Расписание», перед пользователем открывается список готовых расписаний, рисунок 40.

Расписания

ФИО	Предмет	/	
Виденин Сергей Александровича	Базы Данных	Просмотр	Скачать
Виденин Сергей Александровича	Информационные системы	Просмотр	Скачать
Виденин Сергей Александровича	Информационные системы	Просмотр	Скачать

Рисунок 40 – Список готовых расписаний.

Далее, выбрав просмотр, пользователь может просмотреть расписание, или скачать его себе на компьютер.

2.4 Вывод ко второму разделу

В данном разделе были подробно описаны инструменты для разработки, реализованы все функции, которые подразумевала в себя разработанная

автоматизированная система «Расписание для ЦДО». В ходе тестирования было выявлено, что все функции созданы правильно, и работают корректно.

Проанализировав работу системы, были сделаны выводы, что повысилась эффективность составление расписания. Данный процесс стал занимать намного меньше времени, в частности увеличилась скорость сбора данных о свободном времени студентов. Соответственно, можно сказать достигнута главная цель данной выпускной квалификационной работы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения задач выпускной квалификационной работы были выполнены пункты:

– был проведен анализ работы центра дополнительного образования, и выявлена основная проблема работы данного учреждения;

– был разработан проект автоматизированной системы, которая позволит, повысит эффективность составления расписания;

– в дальнейшем проект автоматизированной системы был реализован и протестирован.

Разработка автоматизированной системы для составления расписания является актуальной темой, о чем говорят множество реализованных проектов. На сегодняшний день существует достаточно количество подобных программ. Но анализ показал, что существовала потребность в разработке собственной системы.

Главным достоинством описанного в этой работе проекта является то, что разработанная система функционирует на различных устройствах под любой операционной системой, что делает ее очень удобной в использовании.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Алексеев, А. А. HTML5. Основы клиентской разработки/ А.А. Алексеев, А. О.Савельев. – М.: Национальный Открытый Университет «ИНТУИТ», 2016. – 272 с.

2. Гома, Х. Designing Concurrent, Distributed, and Real–Time Applications with UML. 2–е издание/ Х. Гома. – М.: Пресс, 2011. – 704 с.

3. Дронов, В.И. HTML, JavaScript, PHP и MySQL. Джентльменский набор Веб–мастера. 4–е издание/ В.И. Дронов, Н.В. Прохоренок. – СПБ.: БХВ, 2015. – 215 с.

4. Дунаев, В.А. HTML, скрипты и стили. 3–е издание / В.А. Дунаев. – СПБ.: БХВ, 2011. – 225 с.

5. Маклафлин, Б. PHP и MySQL. Исчерпывающее руководство/ Б. Маклафлин. – СПБ.: Питер, 2013. – 352 с.

6. Ульман, Л. PHP и MySQL. Создание интернет–магазинов/ Л. Ульман. – М.: Вильямс, 2015. – 180 с.

7. Уоррен, К. Книга веб–программиста. Секреты профессиональной разработки веб–сайтов/ К.Уоррен, Б. Хоган. – СПБ .: Питер, 2013. – 254 с.

ПРИЛОЖЕНИЕ А

Техническое задание

Полное наименование системы: «Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям».

Основание для разработки: приказы по практике о дипломном проектировании.

Заказчик:

Наименование: Центр дополнительно образования.

Адрес фактический: г. Красноярск, ул. Киренского, д. 26–б.

Краткое наименование системы: «Расписание для ЦДО».

А.1 Обоснование задач проекта

Целью данного проекта является повышение эффективности составления расписания диспетчером путем автоматизации процесса:

а) подробно описать актуальность проблемы и предметную область и выявить основные проблемы;

б) провести анализ и подобрать удобную для студентов и преподавателей автоматизированную систему;

в) обоснование экономической эффективности путем внедрение автоматизированной системы.

А.2 Требования к графическому дизайну автоматизированной системы

При разработке автоматизированной системы должны быть использованы преимущественно серые и оранжевые цветовые решения. Оформление должно быть разработано в достаточно консервативном ключе.

Главные разделы системы должны быть доступны после регистрации. На первой странице не должно быть большого объема текстовой информации.

В дизайне системы не должно присутствовать:

– рекламных баннеров;

– не читающегося текста;

– gif–анимации.

А.3 Функциональные требования

Классы пользователей

а) Студент – авторизованный пользователь, обладает правами:

1) просмотр расписания;

2) участвовать в голосование, которое создал преподаватель.

б) Преподаватель – авторизованный пользователь, обладает правами:

1) просмотр расписания;

2) проведения голосование для создания расписания;

3) закрыть голосование;

3) сгенерировать расписание, исходя из голосования студентов;

4) динамически создать PDF–файл с расписанием для последующей печати.

в) Администратор – пользователь, авторизованный в интерфейсе хостинга, на котором расположена система. Имеет полный доступ ко всем функциональным возможностям администрирования системы. Может добавлять, удалять, редактировать записи в БД.

А.4 Требования к представлению сайта

Начало работы

Первая страница системы должна содержать форму регистрацию в системе. Форма имеет поле входа на сайт с выбором между студентом и преподавателем, или регистрации нового пользователя.

После регистрации пользователю становится доступна главная страница системы, которая содержит такие пункты меню, как Расписание и Голосование, также должен присутствовать календарь.

Во время работы с системой при разрешении экрана не менее 1024x768 пикселей не должна появляться горизонтальная полоса прокрутки. При

увеличенном разрешении функциональные блоки расширяются в соответствующих пропорциях, заполняя всю рабочую область экрана браузера.

Авторизация

Пользователи могут авторизоваться после выбора на первой странице системы между Сотрудником и Студентом с помощью специальной формы авторизации. Форма содержит:

– текстовое поле для ввода логина и пароля пользователя;

– кнопку отправки формы.

Данные для доступа (авторизации):

– логин – имя пользователь выбирает сам при регистрации, состоящее из из A–z, 0–9;

– пароль – строка, содержащая от 8 символов, состоящая из A–z, 0–9.

Ниже формы располагаются ссылка:

– сменить пароль;

– регистрация нового пользователя.

Форма «Сменить пароль» содержит поля:

– e–mail адрес пользователя, указанный при регистрации;

– имя пользователя, указанное при регистрации;

– новый пароль.

При неудачной попытке авторизации – появляется сообщение для повторной попытки авторизации.

Создание голосования

После регистрации преподаватель имеет возможность создать голосование. Зайдя в меню Голосование, ему предлагается следующие поля:

– заполнить название предмета;

– выбрать кол–во занятий в неделю;

– выбрать дни недели, которые удобны для проведения занятие непосредственно ему, и соответственно, между этими днями студенты будут голосовать;

– указать учебную группу;

– кнопка отправки формы.

Голосование

После регистрации студент имеет возможность зайти в меню «Голосование», перед ним должен открыться список голосований, в данном списке отображается имя преподавателя создавшего голосование, название предмета, название учебной группы, для которой конкретное голосование.

Нажав на голосование для своей группы, ему открывается форма с полями:

– фамилия, имя, отчество студента;

– учебная группа;

– список дней недели, которые участвуют в голосование;

– под каждым из дней есть раскрывающийся список, в котором есть три позиции «удобно», «средне», «неудобно», студенту нужно под каждым из дней указать одно из значений;

– кнопка отправка формы.

А.5 Требования к информационному обеспечению

Требование к хранению данных

Все данные сайта должны храниться в структурированном виде под управлением реляционной системой управления базой данных MySQL 5.0

Требования к языкам программирования:

– для реализации статических страниц и шаблонов должны использоваться языки HTML 5.0 CSS 3.0;

– для реализации динамических страниц должен использоваться язык PHP 5.2.

Требования к организации гиперссылок

Все ссылки на сайте должны быть относительными.

Требования к объему одной страницы

Объем одной стандартной загружаемой страницы сайта в среднем не должен превышать 150 кб.

Требования к программному обеспечению

Серверная часть:

– PHP 5.2 и выше;

– СУБД MySQL 5.0 и выше;

– возможность доступа к localhost по FTP протоколу;

– 1 пользователь БД.

Клиентская часть:

Любой из перечисленный ниже браузеров (указана минимальная версия):

–Internet Explorer 7.0;

–Mozilla 3.0;

–Opera 9.0;

–Google Chrome 4.0.

Требования к техническому обеспечению

Рабочая станция должна удовлетворять минимальным требованиям:

Серверная часть:

– CPU: 2x2.5GHZ;

– RAM: 4 Gb;

– HDD: от 1 Gb.

Клиентская часть:

– CPU: 2x1.5GHZ;

– RAM: 2 Gb.

Требования к лингвистическому обеспечению

Сайт должен выполняться на русском языке.

Требования к технической эстетике

Сайт должен быть оптимизирован для просмотра при разрешении 1024*768, 1280*1024 без горизонтальной полосы прокрутки и без пустых полей для основных типов разрешения.

А.6 Требования к верстке страниц

html – документ должен соответствовать стандарту w3c в xHTML Strict, и быть сверстан с применением CSS.

html – документ сайта должен иметь блочную верстку (верстку div'ами), вложенные блоки следует отмечать отступами, для отступов использовать табуляцию.

html – код сайта должен быть удобен для понимания и структурирован, сложные и неоднозначные моменты прокомментированы.

Страница должна максимально идентично отображаться во всех современных браузерах: Internet Explorer 7.0 и выше, Mozila FireFox 3.0 и выше, Opera 9.0 Google Chrome 4.0 и выше, и при разрешениях монитора от 1024x768 до 1920x1080.

Все стили следует вынести в файл main.css, определение стилей непосредственно на странице допускается.

Результат требуется представить в следующей структуре файлов:

– /index.php – файл с вёрсткой страницы;

– /main.css – файл стилей сайта;

– /images/ – каталог с графическими файлами дизайна сайта.

Все названия стилей должны быть английскими (без русских слов на латинице).

У всех ссылок должен быть прописан параметр title="".

У всех картинок должен быть прописан параметр alt="".

А.7 Требования к персоналу

Для эксплуатации веб–интерфейса системы от преподавателя не должно требоваться специальных технических навыков, знания технологий или программных продуктов, за исключением общих навыков работы с персональным компьютером и стандартным веб–браузером.

Преподаватель: уверенный пользователь сети Интернет.

Студент: уверенный пользователь сети Интернет.

А.8 Состав и содержание работ по созданию системы

В соответствии с ГОСТ 34.601–90 в таблице А.1 приведены следующие стадии и сроки выполнения работ.

Таблица А.1 – Этапы работ

Стадии	**Этапы работ**	**Сроки**	**Итоги выполнения**
Формирование требований к ИС	1.1. Обследование объекта и обоснование необходимости создания ИС. 1.2. Формирование требований пользователя к ИС. 1.3. Оформление отчёта о выполненной работе и заявки на разработку ИС (тактико–технического задания)	Выполнено	Список требований, ТЭО, Заявка на разработку ИС, Отчет о проделанной работе
Разработка концепции ИС	2.1. Изучение объекта. 2.2. Проведение необходимых научно–исследовательских работ. 2.3. Разработка вариантов концепции ИС, удовлетворяющего требованиям пользователя. 2.4. Оформление отчёта о выполненной работе.	Выполнено	Отчет о проделанной работе
Техническое задание	Разработка и утверждение технического задания на создание ИС.	Выполнено	ТЗ
Технический проект	5.1. Разработка проектных решений по системе и её частям. 5.2. Разработка документации на ИС и её части. 5.3. Разработка и оформление документации технических требований на их разработку. 5.4. Разработка заданий на проектирование в смежных частях проекта объекта автоматизации.	Выполнено	ТП
Ввод в действие	6.1. Подготовка объекта автоматизации к вводу ИС в действие. 6.2. Проведение опытной эксплуатации.	Через 5 месяцев после начала выполнения работ	Акт о приёмке ИС в постоянную эксплуатацию

Перечень документов, предъявляемых по окончанию стадий работ по ГОСТ 34.201–89.

А.9 Порядок контроля и приёмки системы

Система подвергается испытаниям по ГОСТ 34.603–92 следующих видов:

а) предварительные испытания;

б) опытная эксплуатация;

в) приемочные испытания.

А.10 Дополнительные требования

Требования к безопасности:

Требуется защитить исходный код общей части сайта. Не должно быть возможности считать php–код скриптов. Требуется разграничение доступа. Пароли пользователей хранятся в зашифрованном виде. На уровне СУБД должно быть реализовано разграничение доступа к данным в БД.

А.11 Источники разработки

Настоящее Техническое Задание разработано на основе следующих документов и информационных материалов:

– федерального закона Российской Федерации от 27.07.2006 г. № 152–ФЗ «О персональных данных»;

– положения об обеспечении безопасности персональных данных при их обработке в информационных системах (утв. Постановлением правительства РФ от 17 ноября 2007 г. № 781);

– указа Президента Российской Федерации от 17 марта 2008 г. № 351 «О мерах по обеспечению информационной безопасности Российской Федерации при использовании информационно–телекоммуникационных сетей международного информационного обмена»;

– ГОСТ 21958–76 «Система "Человек–машина"»;

– ГОСТ 21552–84 «Средства вычислительной техники. Общие технические требования, приемка, методы испытаний, маркировка, упаковка, транспортирование и хранение»;

– ГОСТ 34.003–90 «Персонал автоматизированной системы»;

– ГОСТ 34.201–89 «Виды, комплектность, обозначения документов при создании АС»;

– ГОСТ 34.601–90 «Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания»;

– ГОСТ 34.602–89 «Техническое задание на создание автоматизированной системы»;

– ГОСТ 34.603–92 «Информационная технология. Виды испытаний автоматизированных систем»;

– ГОСТ Р 50739–95 «Средства вычислительной техники. Защита от несанкционированного доступа к информации. Общие технические требования»;

– ГОСТ Р 51583 «Защита информации. Порядок создания автоматизированных систем в защищенном исполнении»;

– РД 50–34.698–90 «Автоматизированные системы. Требования к содержанию документов».

ПРИЛОЖЕНИЕ Б

Календарный план ВКР

на тему: «Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям» выполнил студент: Корнилов Дмитрий Александрович.

Таблица Б.1 – Календарный план ВКР

№	Этапы	Срок выполнения
1	предварительная формулировка темы и содержания выпускной квалификационной работы	1.11.2015
2	утверждение темы выпускной квалификационной работы; начало работы над обзорной частью выпускной квалификационной работы	15.02.2016
3	сбор материала для 1 главы	21.03.2016
4	написание 1 главы, сбор материала для 2 главы	1.04.2016
5	написание 2 главы	18.04.2016
6	разработка решения выпускной квалификационной работы	15.05.2016
7	оформление выпускной квалификационной работы и электронной копии	02.06.2016
8	подготовка 10 минутного доклада и демонстрационных материалов для защиты работы презентационными средствами (в PowerPoint и др.)	03.06.2016
9	предзащита работы на кафедре с демонстрацией результатов, в том числе возможно показ программных модулей, автоматизирующих результаты	08.06.2016
10	сдача работы научному руководителю для написания отзыва и рецензии на работу	13.06.2016
11	защита работы перед государственной аттестационной комиссией (доклад в сопровождении презентации в течении 10 минут, чтение доклада не допускается)	21.06.2016

Студент _____

Научный руководитель _____

Федеральное государственное автономное образовательное учреждение высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Информационные системы

УТВЕРЖДАЮ
Заведующий кафедрой ИС
_____ С.А. Виденин
подпись
« 16 » 04 2016 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
в форме бакалаврской работы

Студенту: Корнилову Дмитрию Александровичу

Группа: КИ12-11Б Направление: 09.03.01 Информатика и вычислительная техника

Тема выпускной квалификационной работы: Программный модуль оптимизации расписаний занятий для ЦДО по информационным технологиям.

Утверждена приказом по университету № 5150/с от 14.04.2016 г.

Руководитель ВКР: С. А. Виденин, К.П.Н. заведующий кафедрой «Информационные системы» ИКИТ СФУ.

Исходные данные для ВКР: список требований к разрабатываемой системе, методические указания научного руководителя, учебные пособия.

Перечень разделов ВКР: анализ предметной области, реализация автоматизированной системы.

Перечень графического материала: презентация, выполненная в Microsoft Office PowerPoint 2010.

Руководитель ВКР _____ С.А. Виденин
(подпись)

Задание принял к исполнению _____ Д.А. Корнилов
(подпись)

« 16 » 04 2016 г.