Оглавление

Введение....2
1. Характеристика предприятия....4
1.1 О поликлинике....4
1.2 Характеристика информационных потоков....11
1.3 Текущее состояние автоматизации....15
1.3 Описание объекта (или процессов) автоматизации....17
2. Требования к функциональности системы....20
2.1 Функциональность системы....20
2.1.1 Требования к обеспечению надёжного функционирования системы....21
2.1.2 Удобство использования....22
2.2 Анализ существующих систем управления контентом сайта....23
2.2.1. Обзор существующих CMS. «Joomla»....24
2.2.2. Обзор существующих CMS. «Drupal»....26
2.2.3. Обзор существующих CMS. «Cotonti»....28
2.2.4. Обзор существующих CMS. «WordPress»....31
2.3.3. Обзор технических средств разработки....33
2.3.1 Локальный сервер....33
2.3.2. Веб-сервер....34
2.3.3 HTML....34
2.3.4 JavaScript....40
2.3.5 MySQL....41
2.3.6 Средства визуализации сайта....41
3. Практическая часть....43
3.1 Выбор доменного имени....43
3.2 Выбор хостинга....44
3.3 Разработка проекта на сервере....45
3.4 Установка CMS....45
3.5 Установка необходимых модулей и расширений....48
3.6 Базы данных....49
3.7 Разработка функционального элемента «Запись к врачу»....51
3.8 Разработка и структура таблиц баз данных....54
3.9 Интерфейс разработки....63
Заключение....68

Список используемой литературы.....69
Приложения.....71

Введение

Удаленная запись к врачам – это сервис направлен на повышение доступности медицинской помощи и улучшение качества обслуживания горожан.

Запись к врачу по Интернету является одним из самых простых из способов бронирования талонов на прием: Вам не приходится выходить из дома, набирать номер телефона и ждать на линии пока в регистратуре смогут ответить на ваш звонок.

Нет необходимости стоять в очереди в регистратуру поликлиники с самого раннего утра. Запись к врачу через Интернет разгружает сотрудников поликлиник и делает процесс бронирования талона на прием более легким и удобным. Заходя на сайте в раздел онлайн записи к врачу, у Вас перед глазами появляется список всех доступных на ближайшее время врачей для записи на прием.

Тема дипломной работы посвящена автоматизации работы с информацией для оптимизации процесса координации расписания персонала и посещения пациентов.

Перспективным способом решения данной проблемы является создание компьютерных баз данных, позволяющих хранить, систематизировать и обрабатывать данные.

Актуальность исследования определяется тем, что в настоящее время, люди активно пользуются интернетом, поэтому, неудивительно, что сейчас почти все можно сделать через интернет.

Поскольку больница — это такая организация, которая работает с большим количеством людей и информации, то целесообразно создать для нее базу данных с онлайн-записью, для удобства как пациентов, так и персонала.

При создании информационной системы онлайн-записи к врачу будет использована одна из наиболее распространенных СУБД MySQL,

инструмент для визуального проектирования баз данных MySQL – MySQLWorkbench.

1.**Характеристика предприятия**

1.1 О поликлинике

Настоящая работа выполнена на базе МБУЗ "Детская городская клиническая поликлиника *№*9с 28 мая 2018 года по 10 июня 2018.

О Поликлинике

1 октября 1971 года было открыто самостоятельное детское учреждение города – поликлиника № 13.

В 1976 году поликлиника № 13 была переименована в ДГБ № 4.

В 1978 году было открыто новое здание поликлиники по адресу: ул. Красного Урала, 1.

С 1985 года ДГБ № 4 стала базой кафедры педиатрии ЧГМИ, возглавляемой профессором Узуновой А.Н.

В 1994 году ДГБ № 4 была впервые лицензирована на II категорию.

23.03.2005 года МУЗ ДГБ №4 переименована в «Муниципальное учреждение здравоохранения Детская городская поликлиника № 9».

С 20 октября 2015 года Детской городской поликлинике № 9 присвоен статус клинической поликлиники в связи с размещением на ее базе кафедры «Пропедевтики детских болезней и педиатрии» ГБОУ ВПО «УЮГМУ».

С 2013 года – по настоящее время, возглавляемая главный врач педиатр высшей квалификационной категории Евчая Н.И.

Согласно Устава Учреждение является юридическим лицом, имеет в оперативном управлении обособленное имущество, учитываемое на его самостоятельном балансе, может от своего имени приобретать и осуществлять имущественные и личные неимущественные права, нести обязанности, быть истцом и ответчиком в суде. Организация имеет расчетный и иные счета в банках, печать и штамп со своим наименованием и указанием места нахождения, бланки установленного образца.

Организация относится к муниципальной системе здравоохранения. Учреждение создано с целью оказания не территории муниципального образования – города Челябинск первичной медико-социальной помощи

прикрепленному населению.

Органами управления учреждения являются:

- учредитель;

- единоличный исполнительный орган – главный врач;

Учредителем организации является администрация города Челябинска в лице комитета по управлению муниципальной собственностью города Челябинска, комитета по делам здравоохранения администрации города Челябинска.

Руководство текущей деятельностью организации осуществляет единоличный исполнительный орган учреждения – главный врач. Свою деятельность учреждение вправе осуществлять после получения соответствующей лицензии.

Главный врач является единоличным исполнительным органом. За различные сферы деятельности учреждения отвечают заместители главного врача:

- Начальник статистики;

- Заместитель главного врача по медицинской части;

- Заместитель главного врача по хозяйственным вопросам;

Главная медицинская сестра непосредственно подчиняется заместителю главного врача по медицинской части, она руководи старшими медицинскими сестрами отделений.

Старшие медицинские сестры контролируют медицинских сестер, младших медицинских сестер и санитарок отделений.

- Приказы, постановления, распоряжения вышестоящих организаций (комитет по делам здравоохранения города Челябинска; администрация района, города);

- Устав МУЗ ДГБ №9.;

- Коллективный договор МУЗ ДГБ №9.;

- Штатное расписание МУЗ ДГБ №9.

Персонифицированный учет в МУЗ ДГБ №9 осуществляется

работниками отдела кадров.

Руководитель отдела – начальник, непосредственно подчиняется главному врачу.

В соответствии с распоряжением Управления Здравоохранения Челябинска и Уставом больницы регулярно проводятся районные «Дни здоровья» и дни здоровья во дворах района в рамках городской программы по пропаганде здорового образа жизни и санитарно–просветительной деятельности среди населения.

В целях повышение доступности и улучшения качества медицинской помощи населению за счет оптимизации процессов и устранения потерь, были внедрены пилотные проекты, чтобы повысить уровень удовлетворенности пациентов работой поликлиники.

1. «Повышение эффективности работы участкового врача-педиатра на амбулаторном приеме МБУЗ ДГКП №9»

Задачи:

1. Создание алгоритма работы участкового педиатра для эффективности работы.
2. Создание алгоритма работы участковой медсестры для эффективности работы.
3. На рабочем месте врача-педиатра для правильной, безопасной и эффективности организации рабочего места.

2. «Организация работы кабинета медико – социальной помощи детям и их семьям МБУЗ ДГКП №9»

Задачи:

1. Оказание медико-социальной помощи детям и их семьям МБУЗ ДГКП №9
2. Определить точный алгоритм работы врачей педиатров и участковых медицинских сестер
3. Определить алгоритм работы для социального педагога.
4. Система внутреннего взаимодействия МБУЗ ДГКП № 9

Система внешнего взаимодействия МБУЗ ДГКП №9

3. «Оптимизация работы лаборатории в МБУЗ ДГКП №9»

Задачи:

1. Создание электронной очереди
2. Устранение многоэтапнности передачи результатов анализов.
3. Ликвидировать невостребованность анализов.

Поликлиника прошла участие в пилотном проекте и добилась более качественного, доступного, комфортного обслуживание пациентов. Выполнение поставленных задач сделает организацию лечебно-диагностических процессов прозрачной и понятной как пациентам, так и медицинским работникам и руководителям медучреждений.

В 2018 году, с целью оптимизации работы регистратуры, повышения качества труда медицинских работников, проведен ремонт помещений регистратуры (позволивший увеличить ее площадь) поликлиники, приобретена специальная техника, что позволило улучшить условия приема пациентов и в первую очередь пациентов, имеющих льготы по Федеральному закону от 22.08.2004 № 122-ФЗ «О внесении изменении в законодательные акты Российской Федерации и принятии утратившими силу некоторых законодательных актов Российской Федерации в связи с принятием федеральных законов «О внесении изменений и дополнений в Федеральный закон «Об общих принципах организации законодательных (представительных) и исполнительных органов государственной власти субъектов Российской Федерации и «Об общих принципах организации местного самоуправления в Российской Федерации».

Постоянно совершенствуется организация медицинской помощи, позволяющая избежать очередей и сократить время ожидания консультации или диагностического обследования до минимума (1-2 дня):

Предварительная самозапись к специалистам очно, по телефону, с 01.05.2011г. – через Интернет.

Осмотр диспансерной группы «узкими» специалистами в организованных коллективах.

Наличие дежурного администратора, регулирующего потоки пациентов, отслеживающего изменения в расписании, имеющего возможность видеонаблюдения и предотвращения скопления пациентов у кабинетов врачей.

Подводя итоги, можно сказать, что в МУЗ ГКБ №9 много подразделений областного и городского уровня, количество посещений к врачам всех специальностей растет, остается эффективной и доступной медицинской помощи, оказываемая жителям города.

Информационно-аналитический отдел.

В 2016 году, в связи с активным развитием информатизации в МБУЗ "Детская городская клиническая поликлиника №9 был организован информационно-аналитический отдел.

Основной задачей информационно-аналитического отделения является организационно-методическое обеспечение и планирование деятельности учреждения, а так же сбор, учет и обработка медико–статистической информации.

Не менее важной задачей сотрудников отделения является обеспечение доступности медицинской помощи детям. В регистратуре Вам предоставят полную информацию о расписании работы специалистов и лечебно-диагностических кабинетов. Медицинские регистраторы обеспечивают запись на прием к врачу, прием вызовов врача на дом, учет и заполнение историй развития ребенка с использованием медицинских информационных систем.

В состав отделения входит: IT-служба, call-центр, регистратура, централизованный кабинет выдачи больничных листов, кабинет медицинской статистики, картохранилище, регистратура подразделения поликлиники (Героя России Молодова 3/1), регистратура КДЦ (Куйбышева, 11А), регистратура поликлиники (Красного Урала, 1).

Общая численность отдела: 24 человека.

Цели отдела:

1. Использование возможностей современных информационных технологий для повышения качества медицинской помощи.

2. Обеспечение руководителей и специалистов всех уровней информационно-технической и аппаратной поддержкой для принятия эффективных управленческих решений и определения тактики лечебно-диагностического процесса.

3. Обеспечение руководителей и специалистов всех уровней программным и аппаратным инструментарием для автоматизированной обработки информации.

4. Формирование концепции единой медицинской информационной системы, сопряженной с лечебно-диагностической техникой и сегментом финансово-хозяйственной деятельности больницы

5. Введение статистики и сдача отчетности в государственные органы. Сбор и анализ информации.

6. Участие в организации закупок, ввод в эксплуатацию, настройка, поддержание в рабочем состоянии, развитие и оптимизация парка аппаратного и программного обеспечения автоматизированных рабочих мест и серверных станций, периферийной техники, компьютерных и телефонных сетей;

Задачи отдела:

1. Внедрение и обеспечение функционирования автоматизированных систем нормативно-справочной информации, классификации и кодирования данных

2. Внедрение и обеспечение функционирования информационных систем лечебно-диагностического процесса и финансово-хозяйственной деятельности больницы

3. Внедрение и обеспечение функционирования информационных систем медицинских аппаратно-технологических комплексов

4. Внедрение и обеспечение функционирования информационных систем мониторинга и эксплуатации аппаратных и программных средств автоматизации

5. Внедрение и обеспечение функционирования информационных систем информационной безопасности, защиты электронных средств хранения, обработки и передачи данных

Нормативные документы:

1. Об охране здоровья граждан № 323 ФЗ
2. Закон о персональных данных № 152 ФЗ
3. Об обязательном медицинском страховании № 326 ФЗ
4. Приказ Минздравсоцразвития России от 26.04.2012 № 406н «Об утверждении Порядка выбора гражданином медицинской организации при оказании ему медицинской помощи в рамках программы государственных гарантий бесплатного оказания гражданам медицинской помощи
5. Конституция РФ;
6. Гражданский кодекс РФ, Бюджетный кодекс РФ, Трудовой кодекс РФ и т.д.;
7. Основы законодательства РФ «Об охране здоровья граждан»;

1.2 Характеристика информационных потоков

В настоящее время в МУЗ ДГБ №9 существует множество информационных потоков.

Система информационных потоков в МУЗ ДГБ №9 представлена на рисунке 1.

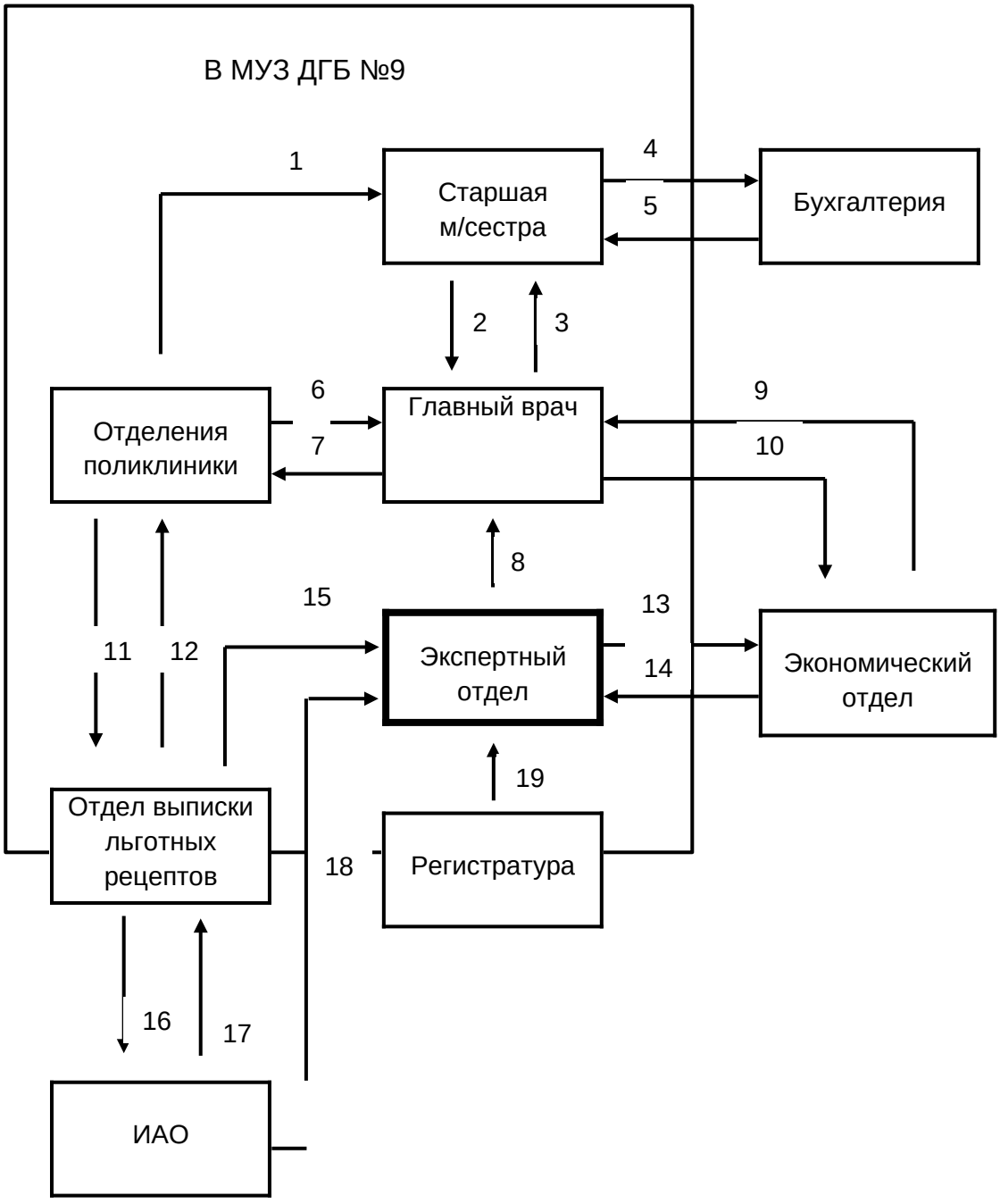

Рисунок 1.1 – Схема информационных потоков в МУЗ ДГБ №9

Условные обозначения потоков:

1, 3, 4 – требования на лекарственные средства, препараты, медикаменты и медицинские и расходные материалы, канцтовары. Накладные, заявки на приобретение и ремонт медицинского оборудования;

2 – отчетная информация о состоянии материально-технической базы, об удовлетворении требований лечебных и профилактических отделений;

5 – уведомления о поступлении средств и материалов, заработной платы, отчетные формы;

6 – отчетная информация о состоянии материально-технической базы, о выполнении приказов и распоряжений, о проведении лечебных и профилактических мероприятий;

7 – постановления, приказы, распоряжения;

8 – отчеты с выделением уровня качества лечения, экспертные заключения о качестве оказанной больному медицинской помощи, заключения ЛКК;

9 – приказы, постановления, плановая информация о сроках проведения лечебно-профилактических мероприятий, сроках сдачи отчетной информации о качестве оказываемой медицинской помощи;

10 – отчетная информация о качестве оказываемой медицинской помощи, состоянии материально-технической базы, проведении плановых мероприятий, требования, заявления;

11, 12 – требования, рецепты;

13 – экспертные заключения о качестве оказанной больному медицинской помощи, заключения ЛКК, отчеты с выделением уровня качества лечения;

14 – приказы, постановления;

15, 16 – отчетная и статистическая информация (о количестве выписанных рецептов на получение медикаментов льготным категориям граждан федеральной, региональной и муниципальной ответственности);

17 – обновления списков льготных категорий граждан, обновления программ;

18 – информация, о количестве обращений граждан лечение и выписку рецептов, сведенья о статталонах;

19 – заполненные статталоны.

Далее построим графики функционирования системы «Детская поликлиника».

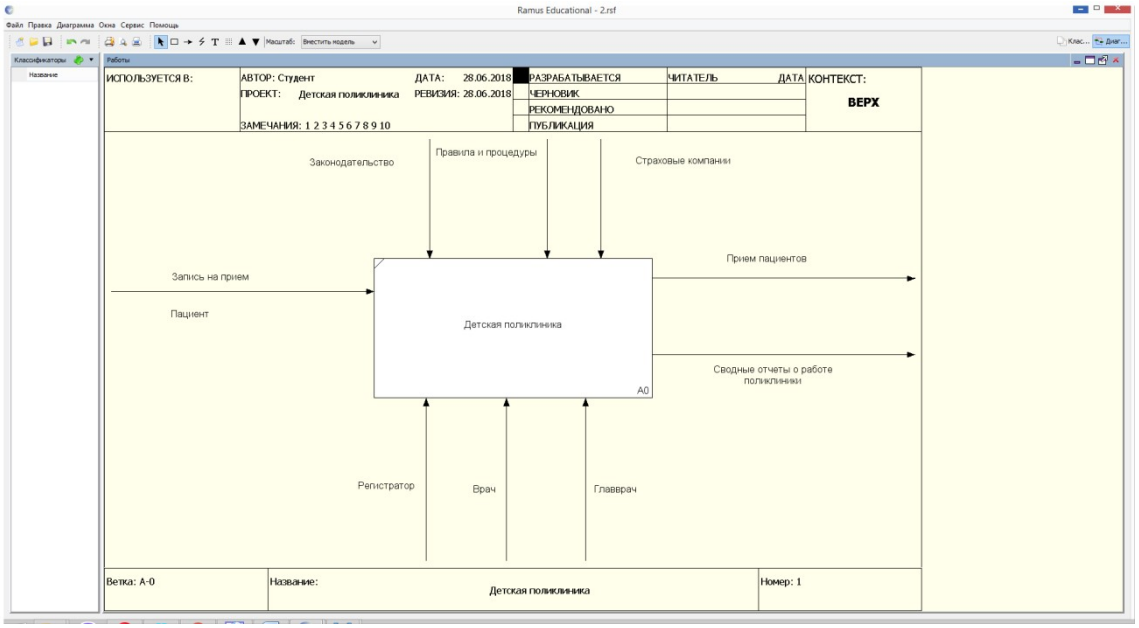

Рисунок 1.2 - IDEF0 диаграмма «Детская поликлиника»

Из всех лечебных и профилактических отделений поликлиники старшей медицинской сестре поступают требования на необходимые медицинские и расходные материалы, канцтовары, лекарственные препараты и медикаменты, заявки на приобретение и ремонт медицинского оборудования. Старшая медицинская сестра отчитывается перед заведующим поликлиникой и экономический отдел. Экономический отдел направляет уведомления о поступлении средств и материалов, заработной платы, отчетные формы, уведомления об удовлетворении требований. Все требования поступают по мере необходимости, количество экземпляров варьируется.

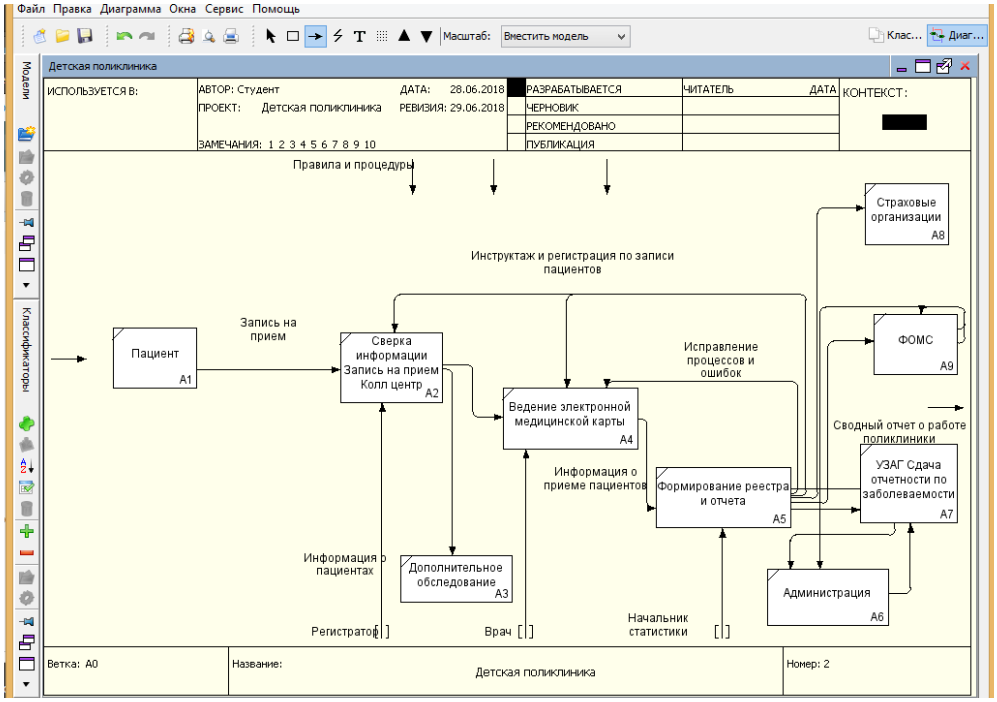

Рисунок 1. 3– Декомпозиция 1 уровня IDEF0 диаграммы «Детская поликлиника»

Главный врач контролирует работу отделений поликлиники посредством распоряжений, а также доводит до сведенья заведующих лечебными и профилактическими отделениями постановления Министерства здравоохранения Челябинской области, к главному врачу, куда они ежедневно поступают по электронной почте и факсу. В среднем в день поступает около 10 распоряжений и приказов.

Из регистратуры в экспертный отдел поступают заполненные статталоны для выставления экспертных оценок. Экспертный отдел направляет заведующему поликлиникой и в экономический отдел отчеты с выделением уровня качества лечения, экспертные заключения о качестве оказанной больному медицинской помощи, заключения ЛКК.

Количество экземпляров зависит от количества больных, оценку уровня лечения которых необходимо провести. Отчет по формам приложения №1 и №2 к приказу №653 МЗРФ от 06.07.2006 г. направляется на магнитном носителе (для отправки по электронной почте) и в бумажном виде (два экземпляра, один остается в экспертном отделе, второй

направляется в организационно-методический кабинет. Отчет по форме приложения №1 формируется еженедельно (каждую пятницу), отчет по форме приложения №2 формируется ежемесячно в последнюю пятницу месяца.

Отдел выписки льготных рецептов удовлетворяет требования отделений поликлиники о необходимости лекарственных препаратов, собирает для экспертного отдела и отдела ИАО статистическую информацию о количестве выписанных рецептов на получение медикаментов льготным категориям граждан федеральной, региональной и муниципальной ответственности и обменивается с ними другой технической информацией (обновления списков льготных категорий граждан, обновления программ).

1.3 Текущее состояние автоматизации

Информация о количестве рецептов доставляется еженедельно (за неделю) и ежемесячно (за месяц) на бумажном носителе. Обновление списков льготных категорий граждан и обновления программ происходит по мере выхода обновленных версий списков и новых версий программ в Министерстве здравоохранения Челябинской области. В свою очередь, из отдела ИАО в экспертный отдел поступает информация о количестве обращений граждан на лечение и выписку рецептов, а также сведения о статталонах. Информация о количестве обращений граждан поступает еженедельно (за неделю) и ежемесячно (за месяц) на бумажных носителях, сведения о статталонах поступают ежедневно на бумажных носителях, количество экземпляров варьируется.

В настоящее время признается безусловная необходимость использования компьютеров в медицинских учреждениях. Возможности современной компьютерной техники настолько велики, что позволяют не только решать традиционные задачи анализа, но и давать достаточно компетентные рекомендации для решения управленческих задач. В

МУЗ ДГБ №9 на сегодняшний день автоматизирована деятельность далеко не всех специалистов, выполняющих сложную работу с большим объемом информации.

В настоящее время полностью автоматизирована деятельность врачей-экспертов, основной задачей которых является выставление экспертных оценок качества медицинской помощи больным, вынесение этих сведений на ЛКК и формирование отчетов на основе оценок качества лечения.

Экспертная оценка качества оказанной больному помощи и лечения ставится на основании данных статталонов и амбулаторных карт пациента.

На основании данных о количестве льготных категорий граждан, обратившихся в, предоставляемых отделом ИАО, и данных о количестве отпущенных им медикаментов, предоставляемых отделом выписки льготных рецептов, врачом-экспертом формируются отчеты в Министерство здравоохранения Челябинской области.

Работа происходит с большим объемом ежедневной информации, при этом производятся математические расчеты. В связи с этим возможны погрешности в расчетах, ошибки, а также возможна потеря части информации вследствие ручной обработки всех документов.

Вышеперечисленные проблемы высокой загруженности информационных потоков, приводят к необходимости автоматизации ряда направлений деятельности больницы.

Одним из направлений является разделение внутренней АИС и создание сайта больницы, что позволит вынести управление записи к врачу, информационные материалы, очередность и расписание приема больных на внешний сайт. Согласно требованиям Здравоохранения, ФОМС, и страховым организациям.

Таким образом, в нашей работе мы рассмотрим создание сайта детской поликлиники.

1.3 Описание объекта (или процессов) автоматизации

Контекстная IDEF0 диаграмма декомпозируется на процессы, которые изображены на рисунке 2. Основные бизнес процессы декомпозируются на:

Первая модель отражает верхний уровень и является общим уровнем моделирования бизнес-процессов ИС «Сайт детской поликлиники», которая включает в себя контекстную диаграмму. Модель включает в себя один блок, характеризующий всю совокупность моделирующих процессов без подробностей.

На данной модели бизнес-процессов входной информацией и объектами, анализируемыми и изменяемыми в ходе выполнения работы, являются:

Регистрация больных на «Сайт детской поликлиники»;

Сопроводительная документация от специалистов.

Механизмами (ресурсами) являются:

Технологическое обеспечение ИС;

Административный контроль.

Управлением являются:

Раздел платные услуги;

Законодательство РФ;

Выходом являются:

Параметры оказываемой услуги;

Отчетность по совершенным приемам врачей и оказанным услугам;

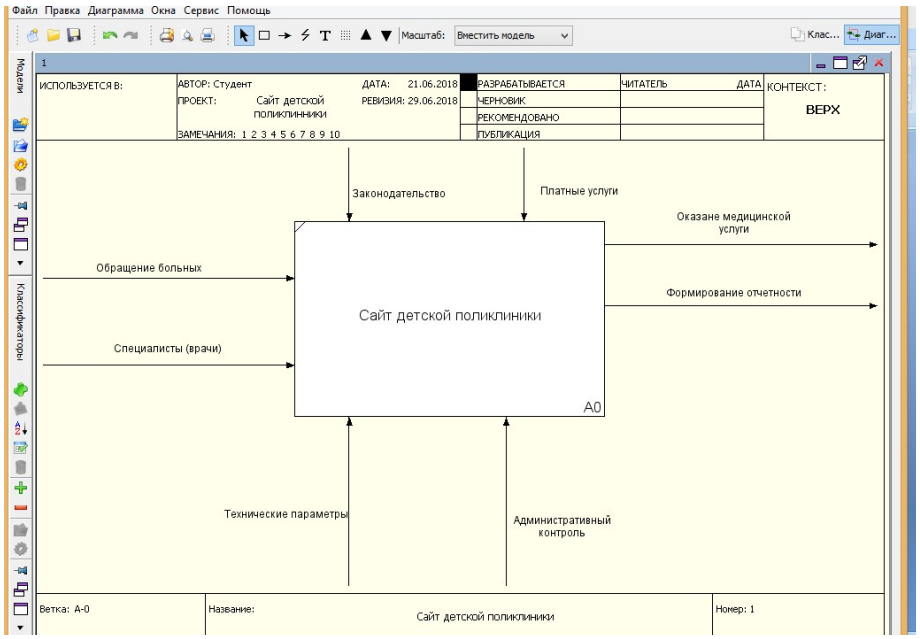

Рисунок 4 - Функциональная модель ИС «Сайт детской поликлиники»

Основными подпроцессами деятельности ИС «Сайт детской поликлиники» являются: техническое обеспечение автоматизированной площадки для формирования заявок на прием к врачу, обеспечение информацией населения о деятельности поликлиники, отчетности по выполненным приемам, представление расписания приемов врачей.

Указанные процессы состоят из совокупности операций, представленных на рис. 3.

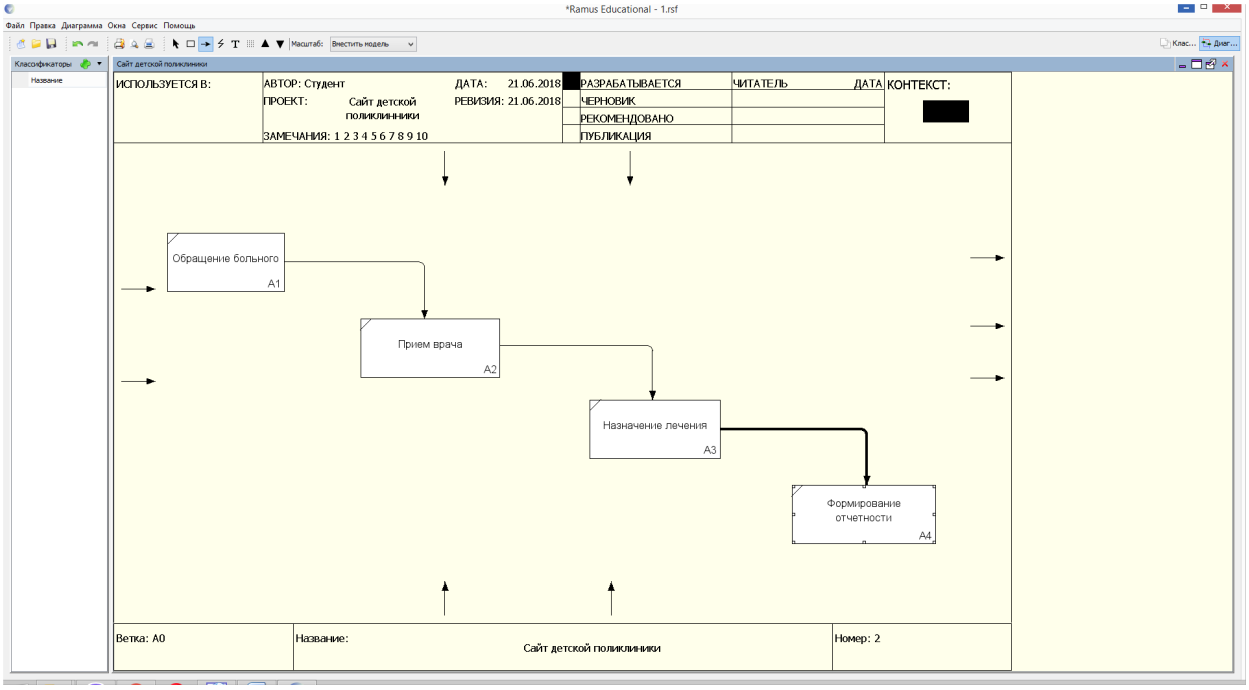

Рисунок 5 - Диаграмма процессов деятельности ИС «Сайт детской

Далее рассмотрим процесс заключения сделки.

Он состоит из следующих подпроцессов:

1. Размещение заявки на прием к врачу.

2. Прием врача.

3. Назначение лечения.

4.Формирование отчетности.

Преимуществами автоматизированной системы «Сайт детской поликлиники» перед традиционным являются:

1. Снижение себестоимости за счет уменьшения расходов на обслуживание документооборота;

2. Сокращение штата сотрудников и уменьшение расходов на заработную плату;

3. Ускорение процессов обработки информации;

3. Сокращение времени на оформление заявок;

4. Расширенные возможности по получению статистических и аналитических данных.

2. Требования к функциональности системы

2.1 Функциональность системы

Разрабатываемая информационная система «Сайт Детской поликлиники» должна обеспечивать следующие функции:

Авторизация пользователей.

Система предлагает всем пользователям системы авторизацию с указанием ролей (больной, специалист).

Аватары пользователей

Каждый зарегистрированный пользователь может настраивать свой профиль, в том числе загружать аватары.

Управление ролями

Этот модуль дает возможность создавать различные роли для пользователей и сотрудников детской поликлиники, а также модерировать права доступа и редактировать контент в кабинете пользователя и панели управления сайтом. Так, например, возможно распределить между различными исполнителями такие задачи, как управление, модерация пользователей, подготовка рассылки, проведение рекламной компании и т.д., предоставив им специализированные права.

Управление пользователями

Администратор располагает контактными данными всех зарегистрированных пользователей, может отслеживать их обращения и полностью контролировать действия, связанные с их работой на ИС «Сайт детской поликлиники».

Верификация через электронную почту

Подтверждение регистрации осуществляется посредством отправки на email пользователя уникального кода. В связи с этим, пользователю необходимо указывать реальные данные. В дальнейшем владелец сайта может проводить email рассылки по важным медицинским тематикам и отправлять пользователям различные уведомления. Верификация также важна для учета принятых специалистами поликлиники новых больных.

Разрабатываемая информационная система имеет несколько уровней доступа. Каждый уровень имеет свои функциональные роли.

Администратор – полный доступ к функционалу информационной системы. Возможность удаления пользователей, назначение ограничений на права пользователей. Удаление товаров, ограничение ставок.

Так же администратор имеет полные права по управлению финансовыми потоками системы.

Модератор – имеет доступ по управлению заказами. Может выносить предупреждения пользователям.

Специалист – имеет права на управление расписанием приема больных, может создавать публичные и личные сообщения. Управление аккаунтом.

Больной – имеет право размещать заявки на прием к врачу.

Незарегистрированный пользователь – имеет возможность просмотра расписания приема врачей, чтения информационных сообщений..

Забаненный пользователь – ограничен доступ к сайту. Выводится предупреждающая запись о сроке окончания ограничений.

2.1.1 Требования к обеспечению надёжного функционирования системы

Надежное (устойчивое) функционирование программы должно быть обеспечено выполнением администратором совокупности организационно-технических мероприятий, перечень которых приведен ниже:

а) организацией бесперебойного питания технических средств;

б) использованием лицензионного программного обеспечения;

в) регулярным выполнением рекомендаций Министерства труда и социального развития РФ, изложенных в Постановлении от 23 июля 1998 г.

Об утверждении межотраслевых типовых норм времени на работы по сервисному обслуживанию ПЭВМ и оргтехники и сопровождению программных средств»;

г) регулярным выполнением требований ГОСТ 51188-98. Защита информации. Испытания программных средств на наличие компьютерных вирусов

2.1.2 Удобство использования

Удобство использования пользовательского интерфейса (usability) - показатель его качества, который определяет количество усилий, необходимых для изучения принципов работы с программной системой при помощи данного интерфейса, ее использования, подготовки входных данных и интерпретации выходных. Иначе говоря, удобство использования определяет степень простоты доступа пользователя к функциям системы, предоставляемым через человеко-машинный (пользовательский) интерфейс.

На удобство использования пользовательского интерфейса влияют следующие факторы:

- легкость обучения - быстро ли человек учится использовать систему;
- эффективность обучения - быстро ли человек работает после обучения;
- запоминаемость обучения - легко ли запоминается все, чему человек научился;
- ошибки - часто ли человек допускает ошибки в работе;
- общая удовлетворенность - является ли общее впечатление от работы с системой положительным.

Разрабатываемая информационная система должна иметь доступный и простой в использовании интерфейс. Расположение основных информационных элементов должно быть логичным и доступным.

Пользовательская информация и способы ее корректировки должны быть человекопонятны.

2.2 Анализ существующих систем управления контентом сайта

Web сайт- такая форма рекламы продукции, при которой просмотр и изучение товаров осуществляется через компьютерные сети. При этом в качестве потенциальных покупателей товаров (или услуг) могут выступать как частные лица, так и организации.

Глобальная сеть Internet сделала информационную систему доступной для фирм любого масштаба. Если раньше организация электронного обмена данными требовала заметных вложений в коммуникационную инфраструктуру и была доступна лишь крупным компаниям, то использование интернета позволяет сегодня вступить в ряды "электронных торговцев" и небольшим фирмам. Информационная система дает любой компании возможность привлекать клиентов со всего мира.

Под определение Информационная система подпадают системы, ориентированные на Интернет - «информационные сайты». В то же время процедуры реклам, инициированных информацией из сети интернет, но использующих для обмена данными факс, телефон и пр., могут быть лишь частично отнесены к классу Информационных систем.

Отмечу также, что, несмотря на то, что Интернет является технологической базой Информационной системы, в ряде систем используются и другие коммуникационные возможности. Так, запросы к продавцу для уточнения параметров товара или для отзывов о нем могут быть посланы и через электронную почту.

Осуществив анализ существующих информационных системы для поликлиники, на сегодняшний день было выявлено множество программных продуктов.Далее рассмотрим подробнее представленные на рынке системы.

2.2.1. Обзор существующих CMS. «Joomla»

- Полностью основанный на БД движок с использованием PHP/MySQL.
- Секции новостей, продукции или услуг легко редактируемы и управляемы.
- Разделы тем могут быть добавлены авторами.
- Полностью настраиваемые схемы расположения элементов, включая левый, правый и центральный блоки меню.
- Форум/Опросы/Голосования для эффективной обратной связи.
- Возможность создавать неограниченное количество страниц
- Для каждой динамической страницы можно создать свое описание и ключевые слова в целях повышения рейтинга в поисковых системах
- Возможность ограничить доступ к определенным разделам сайта только для зарегистрированных пользователей
- Секции новостей, продукции или услуг легко редактируемы и управляемы
- Закачка изображений при помощи браузера в вашу собственную библиотеку - для последующего использования с любого места сайта
- Различные модули - такие как последние новости, счетчик посещений, подробная статистика посещений, гостевая книга, форум и т.д.
- Причем Вы сами выбираете, будут ли показаны эти модули и на каких страницах
- Иерархия объектов - количество секций, разделов, подразделов и страниц, зависит от вашего желания.
- Библиотека изображений позволит Вам хранить все Ваши GIF- и JPEG-файлы под рукой для лёгкого доступа.
- Менеджер архива. Помещайте Ваши старые статьи в архив, чем просто удалять их.

- Выбор из 3 визуальных редакторов, что упрощает редактирование материалов до уровня редактирования текста в программе Word.
- Возможность легкой смены дизайна.
- Экономное использование места на сервере за счет использование базы данных MYSQL.

Таким образом, установив эту систему на свой веб-сайт, мы получаем полный контроль над обновлением информации и развитием нашего веб-сайта. И для этого совсем не нужно иметь никаких специальных знаний, ну и, конечно, эта система бесплатна.

В системе управления содержимым могут быть определены самые различные данные: документы, фильмы, фотографии, номера телефонов, научные данные и так далее. Такая система часто используется для хранения, управления, пересмотра и публикации документации. Контроль версий является одним из основных её преимуществ, когда содержимое изменяется группой лиц.

Какие CMS мы выберем в качестве анализируемых? Обратимся к открытым рейтингам российского интернета:

#	CMS	Проектов	Балл	Тренд
1	Joomla!	2 632	37.05	–
2	Drupal	922	24.02	–
3	MODx CMS / CMF	1 212	17.92	↗
4	WordPress	745	14.69	↘
5	TYPO3	165	4.71	–
6	ImageCMS	50	1.13	new
7	CMS Made Simple	37	0.49	↘

Рисунок 2.1 – Рейтинги систем управления контентом[1]

[1]ratingruneta.ru

Также мы представляем первый в истории рынка подробный сравнительный анализ качественных характеристик CMS.

	UMI.CMS	1С-Битрикс	NetCat	WordPress	Joomla!	MODx	Drupal	HostCMS
Удобство для контент-менеджера	9,0	7,5	8,3	8,3	7,0	7,8	7,6	7,6
Удобство для разработчика	8,6	7,4	7,8	7,6	7,8	8,3	7,4	7,9
Функциональность	8,5	8,6	7,7	7,3	8,2	7,7	8,6	8,2
Итоговая цена владения сайтом	8,9	7,8	7,9	8,1	8,1	8,4	8,2	8,5
Надежность	8,9	8,5	8,0	7,2	7,1	7,7	7,2	8,5
Скорость работы	8,5	6,5	8,5	8,1	6,4	7,9	6,9	8,9
Качество поддержки	8,4	7,6	7,6	5,9	6,4	5,3	4,4	8,6
Безопасность	9,0	8,7	7,9	6,3	6,4	7,1	7,1	8,4
Качество работы производителя CMS с партнерами	9,1	8,4	8,4	—	—	—	—	8,6
Ассортимент шаблонов дизайна	7,4	7,1	4,7	8,9	9,0	5,8	7,6	5,8
Возможности интеграции	8,8	8,3	7,9	7,4	8,0	7,8	6,3	8,0
Совместимость с хостингами	8,3	6,7	8,6	8,5	9,1	8,2	9,1	7,6
Общая оценка этой CMS-системы	9,0	8,3	8,1	8,3	7,9	8,7	8,0	8,1

* В таблице указан средний балл, который каждая CMS получила по тому или иному параметру по итогам опроса респондентов.

Рисунок 2.2 - 2011.tagline.ru

Судя по этим рейтингам в лидерах среди CMS с открытым кодом находятся Joomla, Wordpress, ModX и Drupal. Причем лидирует по суммарному баллу по одной версии Joomla, а по другой Wordpress.

2.2.2. Обзор существующих CMS. «Drupal»

Архитектура Drupal позволяет применять его для построения различных типов сайтов — от блогов и форумов до информационных архивов или сайтов новостей. Имеющийся по умолчанию функционал можно увеличивать подключением дополнительных расширений — «модулей» в терминологии Drupal. Наиболее важные функции, предоставляемые Drupal «из коробки»:

- единая категоризация всех видов содержимого (таксономия) — от форумных сообщений до блогов и новостных статей
- широкий набор свойств при построении рубрикаторов: плоские списки, иерархии, иерархии с общими предками, синонимы, родственные категории
- вложенность категорий любой глубины

- поиск по содержимому сайта, в том числе поиск по таксономии и пользователям
- разграничение доступа пользователей к документам (ролевая модель)
- динамическое построение меню
- поддержка XML-форматов:
- вывод документов в RDF/RSS
- агрегация материалов с других сайтов
- BlogAPI для публикации материалов с помощью внешних приложений
- авторизация через OpenID
- символьные осмысленные URL (иначе «человеко-понятные» — ЧПУ)
- переводы интерфейса сайта на разные языки, а также поддержка ведения разноязычного контента
- возможность создания сайтов с пересекающимся содержимым (например общей базой пользователей или общими настройками)
- раздельные конфигурации сайта для различных виртуальных хостов (в том числе собственные наборы модулей и тем оформления для каждого подсайта)
- механизм для ограничения нагрузки на сайт (автоматическое отключение при высокой посещаемости части информационных блоков и модулей)
- уведомления о выходящих обновлениях модуля.

2.2.3. Обзор существующих CMS. «Cotonti»

Cotonti сочетает гибкость веб-фреймворка с быстрым развертыванием и компонентами системы управления содержимым. Несмотря на то, что в нем «из коробки» есть учётные записи, создание контента, управление файлами и

инструментарий для сообществ, он может быть легко расширен с помощью модулей и плагинов. У Cotonti есть свой шаблонизатор, быстрый и легкий в освоении.

Главные особенности

Открытый код, развивается сообществом

Безопасный и надежный

Быстрый и легкий

Модульный и расширяемый

Вид отделен от логики

Быстрый и удобный шаблонизатор

Интернациональный.

Модель-представление-контроллер (Model-View-Controller или MVC)

Cotonti следует шаблону проектирования MVC, который свойственен для большинства интернет приложений. Суть данной концепции заключается в делении всей инфраструктуры сайта на 3 части:

Данные. Вся информация хранится в БД, вывод данных осуществляется посредством специального API. В настоящее время Cotonti поддерживает только СУБД MySQL. Поддержки других баз не планируется, так как MySQL является самой популярной и наиболее оптимальной для веб-приложений. Хотя, при необходимости возможно написание специального адаптера для других СУБД.

Представление (вид). Это непосредственно сам дизайн сайта, то, как посетитель вашего сайта его увидит. В Cotonti дизайн отделен от кода посредством шаблонов. Любой веб-дизайнер может быстро и без дополнительных усилий освоить работу с шаблонами. В качестве шаблонизатора (системы, которая обеспечивает поддержку шаблонов) мы использовали XTemplate, который является весьма гибким и очень простым в использовании.

Контроллер. Это сам код движка, который связует все звенья воедино и обеспечивает работу сайта.

Модульная архитектура

«Сердцем» данной системы является набор библиотек, которые необходимы для работы всего сайта. Этот набор библиотек называют ядром или CMF (ContentManagementFramework). В целом, это набор файлов, в которых находятся функции и классы, обеспечивающие работу модулей и плагинов.

Основные части сайта представлены модулями, т.е. скриптами, которые отвечают за работу основных областей сайта: форума, страниц, личных сообщений, опросов и т.д.

Гибкость модулей обеспечивается плагинами, небольшими скриптами, которые внедряются в модули, тем самым расширяя функциональность последних. Это внедрение позволяет получить необходимый результат, не прибегая к изменению ядра. Плагины могут быть полностью автономными, в виде всплывающих окон или ajax частей.

Макеты всех модулей и плагинов находятся в шаблонах. Из шаблонов формируется тема (скин) сайта. Каждый скин может иметь несколько вариантов оформления.

Интернационализация

Интернационализация (на англоязычных сайтах обычно используется сокращение i18n) является одной из обязательных особенностей CMS, которая обеспечивает поддержку различных языков. Это необходимый шаг вперед при формировании мультиязычного контента. В современном мире на сайтах все чаще и чаще встречается информация на различных языках. По этой причине все больше и больше людей выбирают в качестве кодировки символов Unicode.

В Cotonti изначально включена поддержка Unicode и по умолчанию используется UTF-8.

Интерфейс системы поддерживает интернационализацию посредством языковых файлов. Все языковые файлы делятся на файлы: ядра, модулей, плагинов, тем оформления. Если по какой-то причине нет необходимого вам

перевода системы, Вы можете сделать его самостоятельно или при помощи переводчика. Наше сообщество предоставит Вам необходимую помощь.

Стабильность и безопасность.

Seditio считался одной из самых стабильных систем управления сайтом в мире (если верить багтракам), а стабильность при написании системы стояла в качестве первостепенной задачи. При создании Cotonti мы придерживались данной традиции. Кодерам известно о SQL-инъекциях, межсайтовом скриптинге (XSS), подделке HTTP-запросов (CSRF), ошибках PHP, поэтому при создании Cotonti наши разработчики уделяют особое внимание защите, как пассивной (внимательное написание кода, исправление существующих ошибок в системе защиты), так и активной (создание специальных алгоритмов защиты).

Существует также четкое разделение прав на сайте посредством списков контроля доступа ACL (AccessControlLists) и режимов доступа. Они представляют собой матрицу групп пользователей и элементов сайта. Поля данной матрицы содержат права группы пользователей.

В Cotonti включена Защита от повторных соединений (Anti-Hammering), а так же защита против спама (anti–spam).

Одним из определяющих стабильность системы является отсутствие в коде ошибок. Достигается это двумя способами: тщательное аккуратное кодирование и сообщение пользователями о найденных ошибках тех. поддержке (так называемое Обеспечение качества или QA) .

Производительность и расширяемость

Когда речь идет о производительности, мы говорим в первую очередь о количестве обрабатываемых запросов в минуту (RPM, RequestsPerMinute) или запросов в секунду (RPS).

SEO (Поисковая оптимизация)

В стандартный комплект Cotonti входит мощная утилита по формированию произвольных URL, а так же настраиваемые заголовки, мета тэги, множество необходимых для этого настроек

AJAX

В Cotonti встроена популярная, хорошо докуметированная JavaScript библиотека jQuery, которая обеспечивает полнофункциональную поддержку AJAX. Вы можете легко и быстро написать плагины, которые поддерживают AJAX технологию.

Подписка (Синдицирование)

Cotonti поставляется со специальным модулем стандарта RSS, который позволит посетителям подписаться на различные каналы.

2.2.4. Обзор существующих CMS. «WordPress»

WordPress(Вордпресс) — это CMS (CMS — система управления содержимым) для создания блогов, новостных порталов, и даже для создания интернет-магазинов.

Характеристики WordPress.

- Благодаря CMS и, в данной случае WordPress, сильно упрощается процесс создания блога, сайта. Создавать сайты и блоги «с нуля» блокнотом довольно долгое занятие, требующее немало знаний в языках программирования (HTML, PHP, CSS и др.). А с помощью Вордпресса блог можно создать буквально за 1 день!
- Возможность написания статей прямо с рабочего стола в оффлайн режиме, с последующей публикацией оттуда.
- Простота установки (знаменитая 5-ти минутная установка WP) и легкость управления блогом с «админки».
- Легко подключить плагины, из-за которых блог можно поменять до неузнаваемости. Те же самые плагины позволяют намного расширить возможности WordPress.
- Поддержка «тем», которые меняют полностью внешний вид. А из-за того, что темы сделаны на PHP языке, велики их графическая и функциональные возможности.
- Очень большой список тем и плагинов для работы.

- Существование русских переводов.
- Легкость в продвижении поисковых роботов.
- WordPress — абсолютно БЕСПЛАТЕН.

При изучении некоторых из вышенаписанных CMS, я остановился на Wordpress, так как он прост и удобен в использовании.Функциональность настоящей CMS достаточная для решения поставленных в настоящей работе целей.

2.3.3. Обзор технических средств разработки

2.3.1 Локальный сервер

Для того чтобы сайт работал нам нужен сервер. Поэтому мне нужно будет использовать в своей работе локальный сервер.

XAMPP —кроссплатформенная сборка веб-сервера, содержащая Apache, MySQL, интерпретатор скриптов PHP, язык программирования Perl и большое количество дополнительных библиотек, позволяющих запустить полноценный веб-сервер.

Функциональность программы XAMPP:

• Apache HTTPD 2.2.6

• MySQL 5.0.45

• PHP 5.2.4 + 4.4.7 + PEAR + Switch

• MiniPerl 5.8.7

• Openssl 0.9.8e

• PHPMyAdmin 2.11.1

• XAMPP Control Panel 2.5

• Webalizer 2.01-10

• Mercury Mail Transport System v4.01a

• FileZilla FTP Server 0.9.23

• SQLite 2.8.15

• ADODB 4.94

• Zend Optimizer 3.3.0

• XAMPP Security

• Ming. For Windows 98, 2000, XP

AppServ — сборкавеб-сервера, содержащаяApache, MySQL, интерпретаторскриптовPHP, phpMyAdminидругиедополнения, позволяющаязапуститьполноценныйвеб-сервер.

Функциональность программы

- MySQL 5.0.24
- Apache 2.2.3
- PHP 5.1.6
- phpMyAdmin 2.9.0.2

2.3.2. Веб-сервер

Се́рвер (англ. server от англ. toserve — служить, мн. ч. се́рверы) — специализированный компьютер или специализированное оборудование для выполнения на нём сервисного программного обеспечения (в том числе серверов тех или иных задач).

Веб-сервер — сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными.

Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер (см.: Сервер (аппаратное обеспечение)), на котором это программное обеспечение работает.

Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы — это HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

2.3.3 HTML

HTML в данной работе используется для отображения на странице текстовой информации, фотографий, аудио и видео файлов.

HTML в данной работе используется для отображения на странице текстовой информации, фотографий, аудио и видео файлов.

Сеть постоянно развивается. Новые и инновационные сайты создаются каждый день, расширяя границы HTML в каждом направлении. HTML4 был вокруг нас в течение почти десяти лет, и издатели ищут новые методы, чтобы обеспечить расширенную функциональность, которая сдерживается ограничениями языка и браузеров.

Чтобы дать авторам больше гибкости и совместимости, сделать их сайты и приложения более интерактивными и захватывающими, HTML5 вводит и расширяет диапазон возможностей, включающий элементы форм, API, мультимедиа, структуры и семантики.

Работа над HTML5 началась в 2004 году и в настоящее время осуществляется в рамках совместных усилий между W3C HTML WG и WHATWG. Многие ключевые игроки участвуют в работе W3C, в том числе представители четырех основных поставщиков браузеров: Apple, Mozilla, Opera и Microsoft, а также ряд других организаций и частных лиц со своими интересами и опытом.

Структура

HTML5 вводит целый ряд новых элементов, которые упрощают структуру страниц. Большинство страниц на HTML4 содержат типовые элементы, такие как «шапка», «подвал» и колонки. Ныне, как правило, в коде документа они обозначаются с помощью элементов , описывая каждый атрибутом id или class.

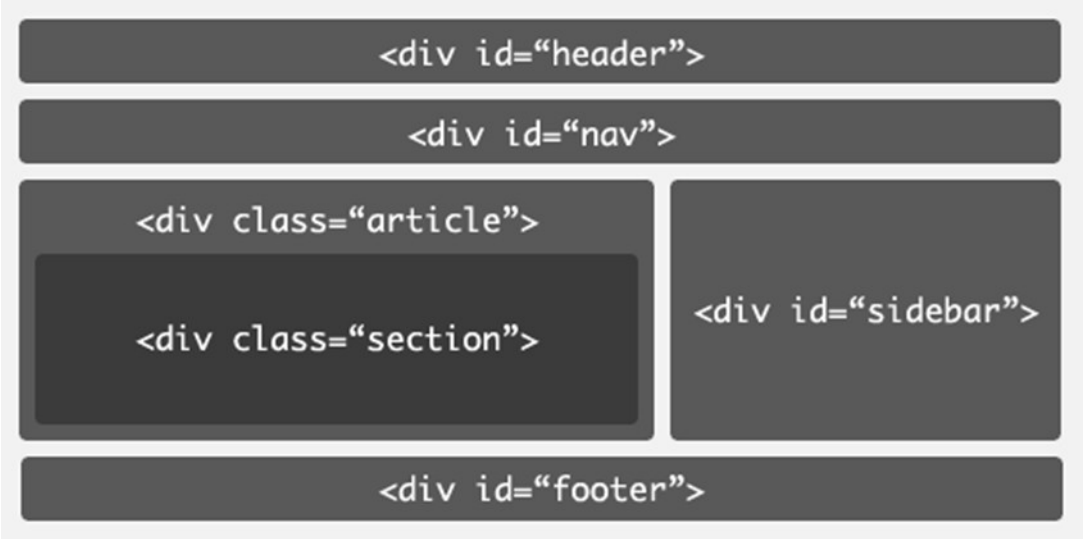

Рисунок 2.5 - Расположение колонок

На рисунке 1 показано типичное расположение двух колонок сверстанных с использованием с атрибутами id и class. Макет содержит шапку, подвал и горизонтальную навигацию ниже заголовка. Основное содержание включает статью и боковую панель справа (сайдбар).

Использовать элементы правильно, потому что в HTML4 не хватает нужной семантики для описания этих частей более конкретно. HTML5 решает эту проблему путем внедрения новых элементов для представления каждого из этих разных разделов.

Элементы могут быть заменены новыми элементами <header>, <nav>, <section>, <article> и <footer>

Видео и аудио

В последние годы видео и аудио в Интернете становится все более жизнеспособным и сайты вроде YouTube, Viddler, Revver, MySpace и десятки других облегчают жизнь тем, кто публикует видео и аудио. Поскольку в настоящее время в HTML не хватает необходимых средств для успешного внедрения и управления мультимедиа, многие сайты полагаются на Flash, чтобы обеспечить эту функциональность. Хотя можно вставлять мультимедиа с помощью различных плагинов (таких как QuickTime, WindowsMedia и др.), Flash сейчас единственный широко используемый плагин, обеспечивающий кроссбраузерную совместимость с требуемыми для разработчиков API-интерфейсами.

Как свидетельствуют различные медиаплееры на основе Flash, их авторы заинтересованы в поддержке собственных пользовательских интерфейсов, которые обычно позволяет пользователям воспроизводить, делать паузу, останавливать, искать и регулировать громкость. Планируется обеспечить эту функциональность в браузерах, добавив встроенную поддержку для видео и аудио и предоставив API скриптам для управления воспроизведением.

Новые элементы <lot1> и <lot> делают это очень простым. Большинство API одинаково для этих двух элементов, разница лишь связана с присущими различиями между визуальным и невизуальным медиа.

Opera и WebKit выпустили версии с частичной поддержкой элемента <lot1>. Вы можете скачать эти браузеры, чтобы опробовать примеры. Opera включает в себя поддержку OggTheora, WebKit поддерживает все форматы, которые поддерживает QuickTime, включая дополнительные кодеки.

Самый простой способ вставить видео это использовать элемент <lot1> и позволить браузеру обеспечить интерфейс по умолчанию. Атрибут controls

указывает, желает автор использовать пользовательский интерфейс или выключить его по умолчанию.

<lot1 src="lot1.ogv" controls poster="poster.jpg"

width="320" height="240">

Скачать ролик

</lot1>

Необязательный атрибут poster может быть использован для указания изображения, которое будет отображаться вместо видео до начала воспроизведения. Хотя некоторые видеоформаты вроде MPEG-4 поддерживают собственную функцию кадра, альтернативное решение может работать независимо от формата видео.

Так же просто вставить аудио на страницу с помощью элемента <lot>. Большинство атрибутов являются общими между элементами <lot> и <lot1>, хотя по понятным причинам у <lot> нет атрибутов width, height и poster.

<lot src="music.oga" controls>

Скачатьпесню

</lot>

HTML5 предоставляет элемент <source> для указания альтернативных аудио и видеофайлов, которые браузер может выбрать на основе типа носителя или кодека. Атрибут media может быть использован для указания типа носителя, чтоб отобрать определенные устройства, атрибут type — для указания типа файла и кодека. Обратите внимание, что при использовании элементов <source>, атрибут src у родительских <lot1> и <lot> или альтернативные<source> игнорируются.

<lot1 poster="poster.jpg">

<source src="lot1.3gp" type="lot1/3gpp" media="handheld">

<source src="lot1.ogv" type="lot1/ogg; codecs=theora, vorbis">

<source src="lot1.mp4" type="lot1/mp4">

</lot1>

<lot>

<source src="music.oga" type="lot/ogg">

<source src="music.mp3" type="lot/mpeg">

</lot>

Для авторов, которые хотят больше контроля над пользовательским интерфейсом, чтобы вписать его в дизайн веб-страницы, обширный API предоставляет несколько методов и событий, позволяющим скриптам управлять воспроизведением. Простейшие методы используют play(), pause() и currentTime для перемотки в начало. Следующий пример иллюстрирует их использование.

<lot1 src="lot1.ogg" id="lot1"></lot1>

<script>

var lot1 = document.getElementById("lot1");

</script>

<p><button type="button" onclick="lot1.play();">Воспроизвести</button>

<button type="button" onclick="lot1.pause();">Пауза</button>

<button type="button" onclick="lot1.currentTime = 0;"><<Перемотать</button>

Для видео и аудио есть много других атрибутов и API, которые здесь не обсуждались. Дополнительную информацию вы можете получить, ознакомившись с данным проектом спецификации.

Представление документа

В отличие от предыдущих версий HTML и XHTML, которые определены с позиции их синтаксиса, HTML5 в настоящее время определяется в терминах объектной модели документа (DOM) —дерево представления, которое используют браузеры для отображения документа. К примеру, рассмотрим очень простой документ, состоящий из названия, заголовка и абзаца. Дерево DOM может выглядеть примерно так.

Дерево DOM включает элемент <title> внутри <head>, а также <h1> и <p> в <body>

Преимуществом определения HTML5 в терминах DOM является то, что язык сам по себе может быть определен независимо от синтаксиса. Существует два синтаксиса, которые могут быть использованы для представления HTML-документов: публикация на HTML (известная как HTML5) и публикация на XML (известная как XHTML5).

Синтаксис HTML основан на SGML, ранней версии HTML, но определен как более совместимый с браузерами на практике.

```
<!DOCTYPE html>
<html>
<head>
<title>HTML-документ</title>
</head>
<body>
<h1>Пример</h1>
<p>Это пример HTML-документа
</body>
</html>
```

Как и в предыдущих версиях HTML, некоторые теги являются необязательными и подразумеваются автоматически.

Преимущества использования HTML

Обратная совместимость с существующими браузерами.

Авторы уже знакомы с синтаксисом.

Снисходительный синтаксис не приведет к появлению «Желтого экрана смерти», при возникновении случайной ошибки.

Удобный синтаксис сокращений, к примеру, авторы могут опускать некоторые теги и значения атрибутов.

Преимущества использования XHTML

Строгий синтаксис XML поощряет авторов писать правильный код, для некоторых авторов он проще в обслуживании.

Напрямую интегрируется с другими XML-технологиями вроде SVG и MathML.

Позволяет использовать обработчик XML, который некоторые авторы используют в своей деятельности.

2.3.4 JavaScript

JavaScript— мультипарадигменный язык программирования.Поддерживает объектно-ориентированный, императивный и функциональный стили. Является реализацией языка ECMAScript (стандарт ECMA-262).

JavaScript обычно используется как встраиваемый язык для программного доступа к объектам приложений. Наиболее широкое применение находит в браузерах как язык сценариев для придания интерактивности веб-страницам[9].

jQuery— библиотека JavaScript, фокусирующаяся на взаимодействии JavaScript и HTML. Библиотека jQuery помогает легко получать доступ к любому элементу DOM, обращаться к атрибутам и содержимому элементов DOM, манипулировать ими.

2.3.5 MySQL

На MySQL составлена сама база данных, используемая сайтом.

2.3.6 Средства визуализации сайта

CSS - используется для создании списка стилей.

LESS (язык стилей) - это динамический язык стилей, который разработал AlexisSellier. Он создан под влиянием языка стилей Sass, и, в свою очередь, оказал влияние на его новый синтаксис «SCSS», в котором также использован синтаксис, являющийся расширением CSS[1].

LESS — это продукт с открытым исходным кодом. Его первая версия была написана на Ruby, однако в последующих версиях было решено отказаться от использования этого языка программирования в пользу JavaScript. Less — это вложенный метаязык: валидный CSS будет валидной less-программой с аналогичной семантикой.

LESS обеспечивает следующие расширения CSS: переменные, вложенные блоки, миксины, операторы и функции[2].

Bootstrap (фреймворк) - свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

Основные инструменты Bootstrap:

Сетки — заранее заданные размеры колонок, которые можно сразу же использовать, например ширина колонки 140 px относится к классу .span2 (.col-md-2 в третьей версии фреймворка), который можно использовать в CSS-описании документа.

Шаблоны — фиксированный или резиновый шаблон документа.

Типографика — описания шрифтов, определение некоторых классов для шрифтов, таких как код, цитаты и т. п.

Медиа — представляет некоторое управление изображениями и видео.

Таблицы — средства оформления таблиц, вплоть до добавления функциональности сортировки.

Формы — классы для оформления форм и некоторых событий, происходящих с ними.

Навигация — классы оформления для табов, вкладок, страничности, меню и панели инструментов.

Алерты — оформление диалоговых окон, подсказок и всплывающих окон.

Таким образом, подводя итог анализа существующих систем разработки мы выбираем следующие технические средства:

Веб-сервер;

CMSWordpress;

В качестве основного средства визуализации выбираем фреймворкBootstrap, поскольку он в себе содержит практически все современные средства для проектирования шаблона сайта, его функциональности и дизайна.

3. Практическая часть

3.1 Выбор доменного имени

Перед тем, как начать работу с сайтом, мы должны выбрать ему домен или доменное имя.

Доменное имя - это определенный набор английских символов, цифр и дефиса идущих до точки. Исключением является регистрация доменного имени в кириллической зоне .РФ, в этом случае используются русские символы.

Для регистрации сайта для России коммерческой тематики предпочтителен выбор доменной зоны .ru. Само имя необходимо выбрать

такое, которое будет описывать ваш сайт и его содержимое. У потенциального посетителя при прочтении нашего доменного имени должно возникать понимание того, что он может получить, зайдя на ваш сайт. Это имя или словосочетание должно ассоциироваться с нашим бизнесом или с той нишей, которой посвящен сайт.

Для выбора хорошего доменного имени прежде всего нужно определиться с тематикой сайта, и в соответствии с ней подобрать несколько десятков английских слов ассоциирующихся с данной тематикой. Затем самое сложное: нужно найти свободное доменное имя для вашего сайта на основе этих слов.

В нашем случае это будет домен второго уровня http://mbuz-dgp9.ru. Во-первых имя домена конкретизирует сферу деятельности, во-вторых обозначает принадлежность к региону.

3.2 Выбор хостинга

Технология виртуального хостинга предусматривает размещение на одном физическом сервере множество клиентов. Вам гарантируется дисковое пространство, согласно вашему тарифному плану. А вот с процессором и оперативной памятью дело обстоит сложнее. Каждый сайт забирает вычислительных ресурсов столько, сколько ему требуется. На сервере может находиться 200-300-500 сайтов.

В качестве наиболее приемлемого хостинга выбираем хость группы компаний Ru-Center.

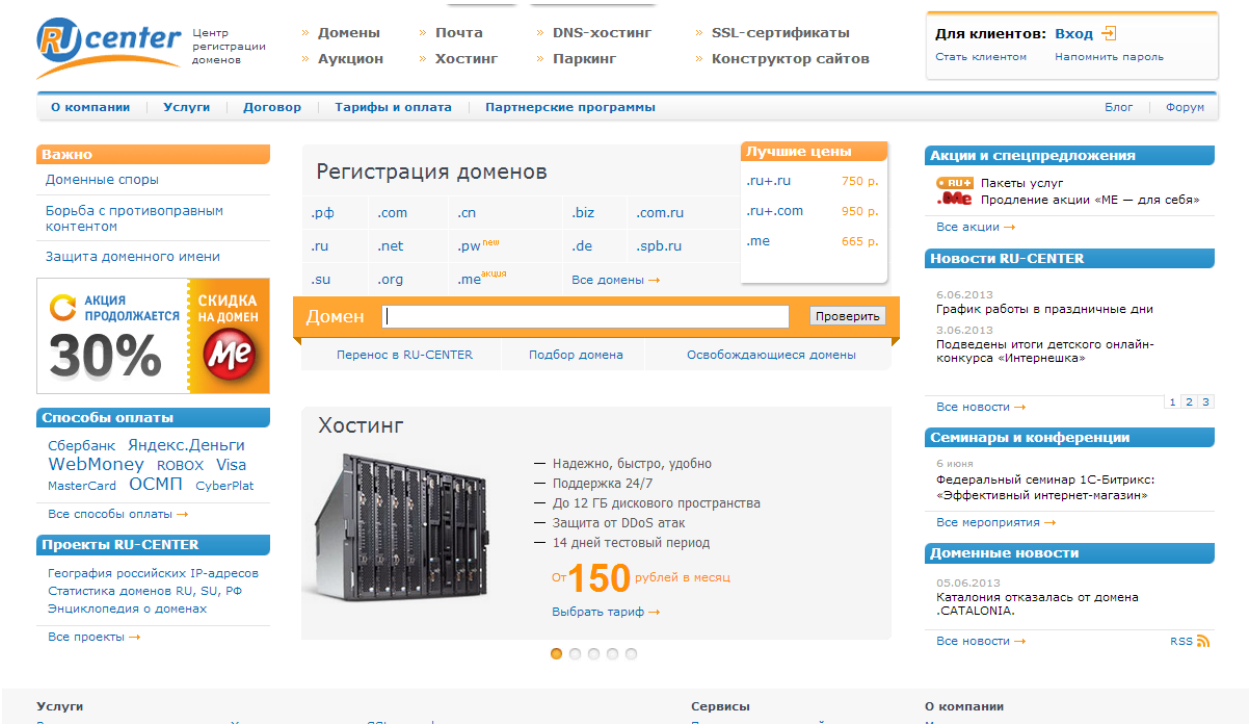

Рисунок 3.1 - Интерфейс хостинга Ru-Center

3.3 Разработка проекта на сервере

В дальнейшем переходим в пользовательский кабинет нашего хостинга и в разделе «сайты» создаем наш сайт: http://mbuz-dgp9.ru.

Предварительно прописываем у регистратора доменных имен DNS адреса.

Таким образом, получаем следующее (рисунок 3.2):

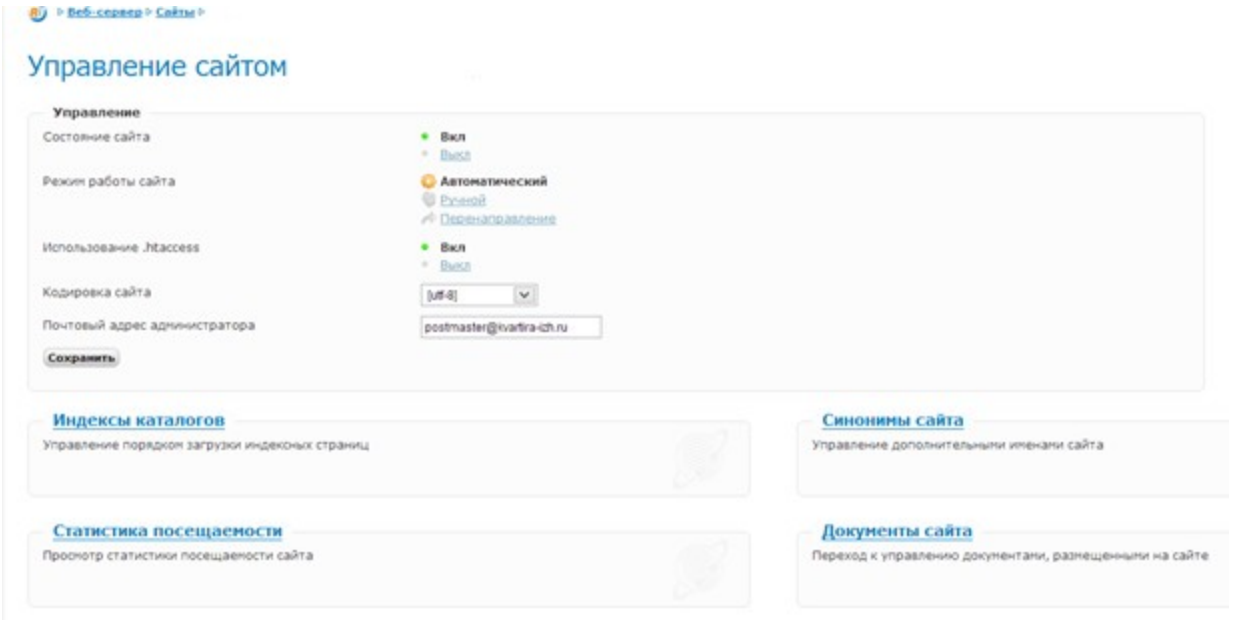

Рисунок 3.2 - Административная панель создаваемого сайта

На следующем этапе производим установку CMS.

3.4 Установка CMS

Как уже говорилось выше, в качестве проектного решения мы выбираем CMSWordPress.

Перед тем как закачивать дистрибутив WordPress на хостинг, необходимо создать базу данных для сайта. Для этого переходим в раздел СУБД MySQL хостинга (Рисунок3.3.).

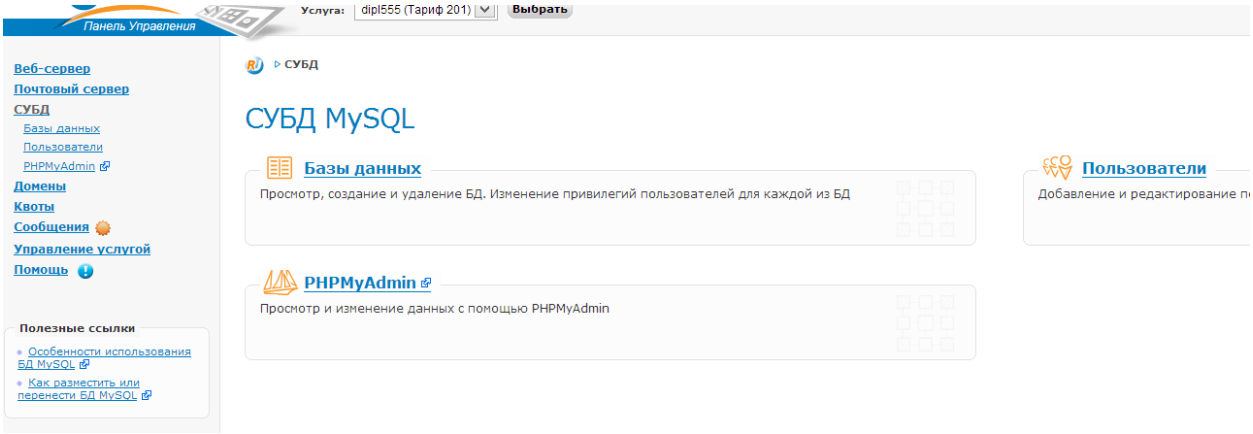

Рисунок3.3 - Панель управления базами данных

И создаем базу данных для нашего сайта.

Далее устанавливаем CMSWordPress. Для этого через FTP клиент закачиваем на хостинг дистрибутив системы управления контентом сайта.

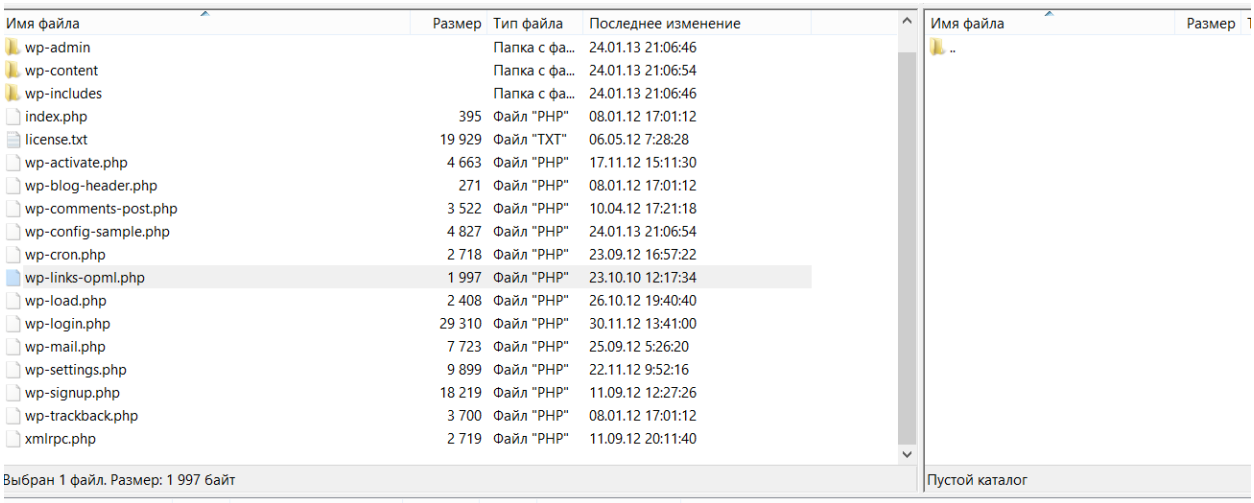

Рисунок 3.4 - Загрузка CMSWordPress

Процесс установки следующий:

1. Разархивируйте пакет в пустой каталог и загрузите на сервер.

2. Откройте wp-admin/install.php в вашем браузере. Этот скрипт поможет вам создать файл wp-config.php с параметрами доступа к вашей базе данных.

3. Если по какой-то причине автоматическая установка не удалась, не волнуйтесь. Она работает не на всех серверах. Откройте wp-config-sample.php в текстовом редакторе наподобие WordPad и заполните параметры доступа к вашей базе данных.

4. Сохраните файл под именем wp-config.php и загрузите на сервер.

5. Откройте wp-admin/install.php в вашем браузере.

6. Как только файл конфигурации будет создан, скрипт установки создаст таблицы, необходимые для вашего блога. Если произошла ошибка, перепроверьте ваш файл wp-config.php и попробуйте ещё раз. Если снова не удалось, обратитесь на форум поддержки с как можно большим количеством собранной информации.

7. Если вы не вводили пароль, запомните тот, который вам выдан. Если вы не вводили логин, им будет admin.

8. Скрипт установки перенаправит вас на страницу входа. Войдите с логином и паролем, выбранными во время установки. Если пароль был

создан автоматически, можно перейти на страницу «Профиль», чтобы сменить его.

В итоге получаем (рисунок 3.4):

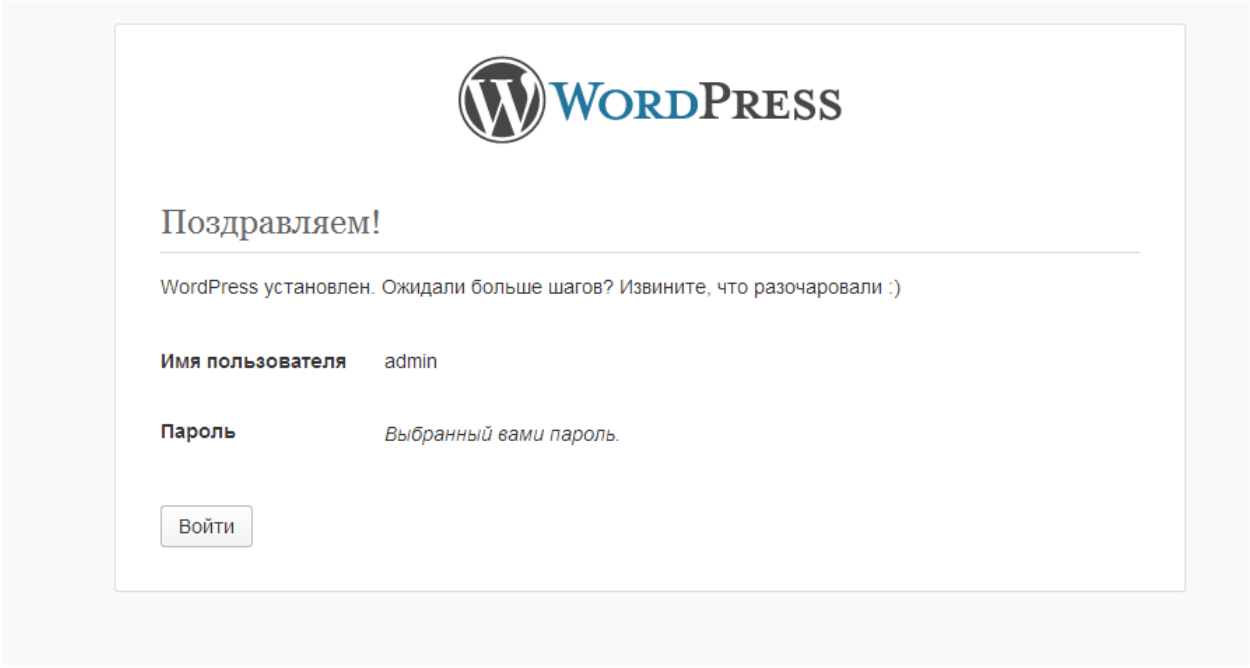

Рисунок 3.4- Завершение установки

3.5Установка необходимых модулей и расширений

Итак, мы установили сайт, теперь у нас есть абсолютно рабочий сайт оказываемых услуг и нам его необходимо заполнить содержанием и функционалом.

На счет наполнения содержанием поговорим в графической части проекта, а сейчас разработаем модульный функционал сайта.

Перейдем в административную часть нашего сайта.

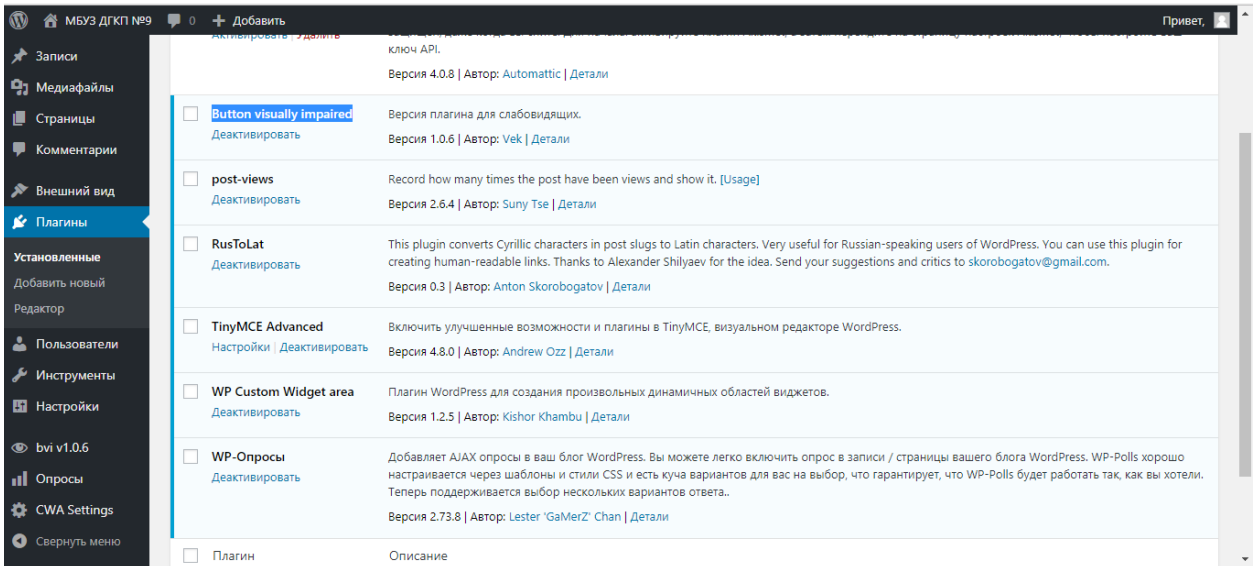

Рисунок3.5 - Административная часть сайта

В ходе анализа предложений в интернете, мы выбрали следующие плагины:

- Buttonvisuallyimpaired;
- WP-Опросы;
- Akismet:
- HelloDolly;
- TinyMCEAdvanced.

Выше указанные плагины позволят реализовать полноценный сайт с функционалом больницы необходимой информацией.

3.6 Базы данных

Как известно для сайта нужна база данных.

Создание базы.

Для создания базы данных запускаем наш сервер. Затем в строке браузера обращаемся к панели управления базами данных phpmyadmin на нашем сервере, которая находится по адресу https://server73.hosting.reg.ru/manager/ispmgr

В окошке "Создать новую БД" набираем имя нашей базы и нажимаем на кнопку "Создать".

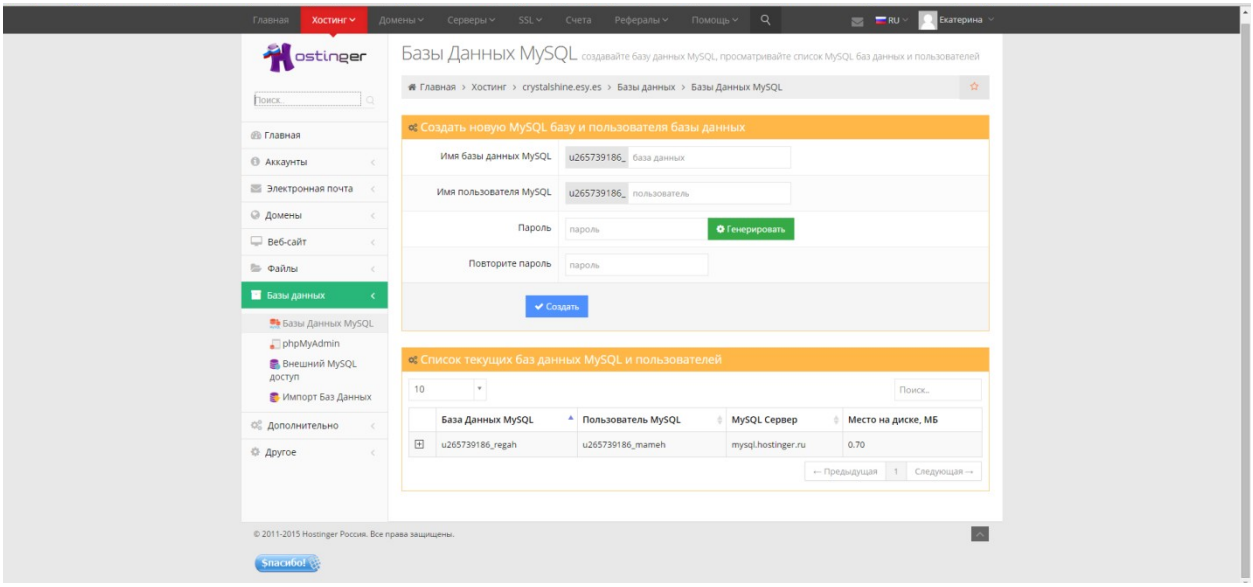

Рисунок 3.6 - Создание базы данных

Теперь надо создать пользователя для нашей базы данных, определить его привилегии и назначить пароль. Для этого выбираем пункт «добавить нового пользователя» В окне добавления нового пользователя набираем:

"Имя пользователя" – ваше имя

"Хост" - localhost,

"Пароль" - ваш пароль

"Подтверждение" - повторить пароль.

Так как сайт был разработан в Wordpress, который сам генерирует базу данных, то инфологическую и даталогическую модель базы нет возможности предоставить.

Физическая модель базы данных предоставлен на рисунках 3.6, 3.7, 3.8.

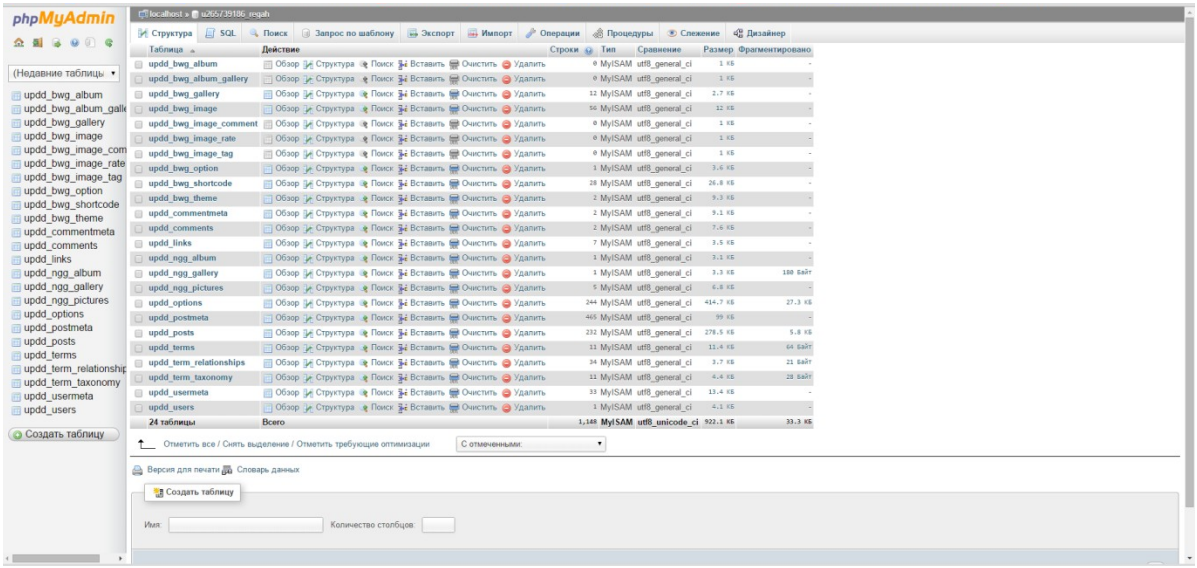

Рисунок 3.6 - Внешний вид базы данныхWp_commentmeta

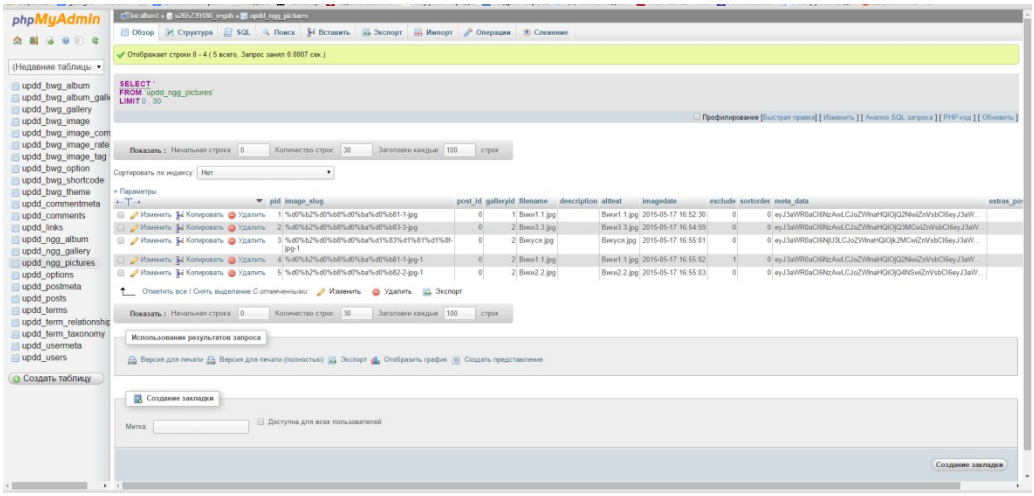

Рисунок 3.7 - updd_ngg_pictures`

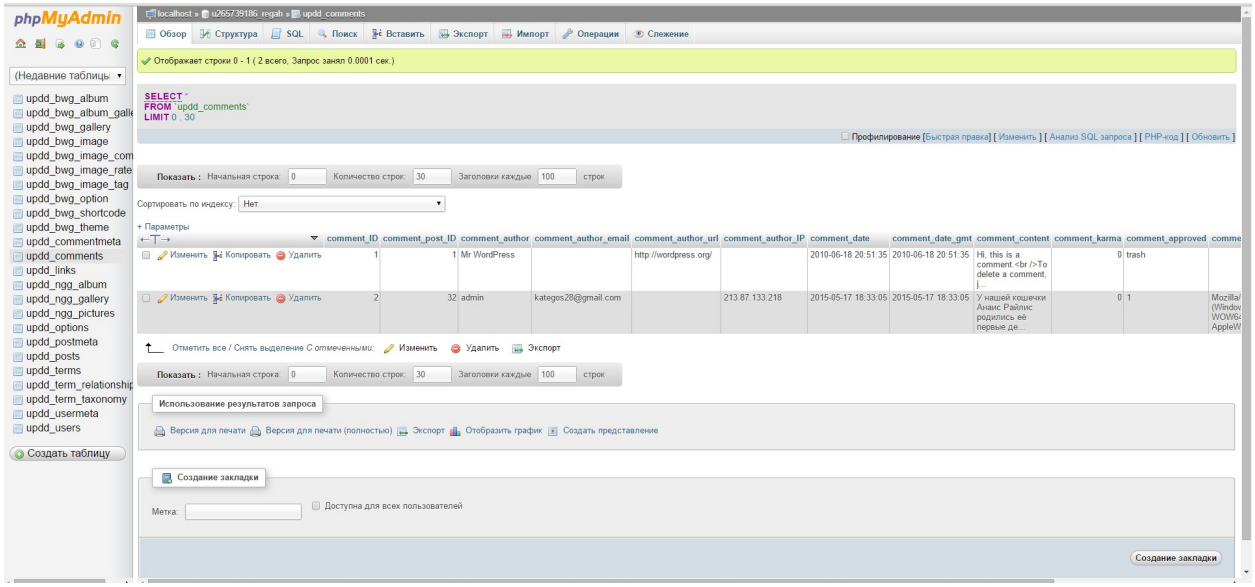

Рисунок 3.8 -Wp_comments

3.7 Разработка функционального элемента «Запись к врачу»

Разработка данного раздела сайта выполнена с использованием фреймворка Cotonti. Система Cotonti состоит из модулей и плагинов, представленных на рисунке 3.8.

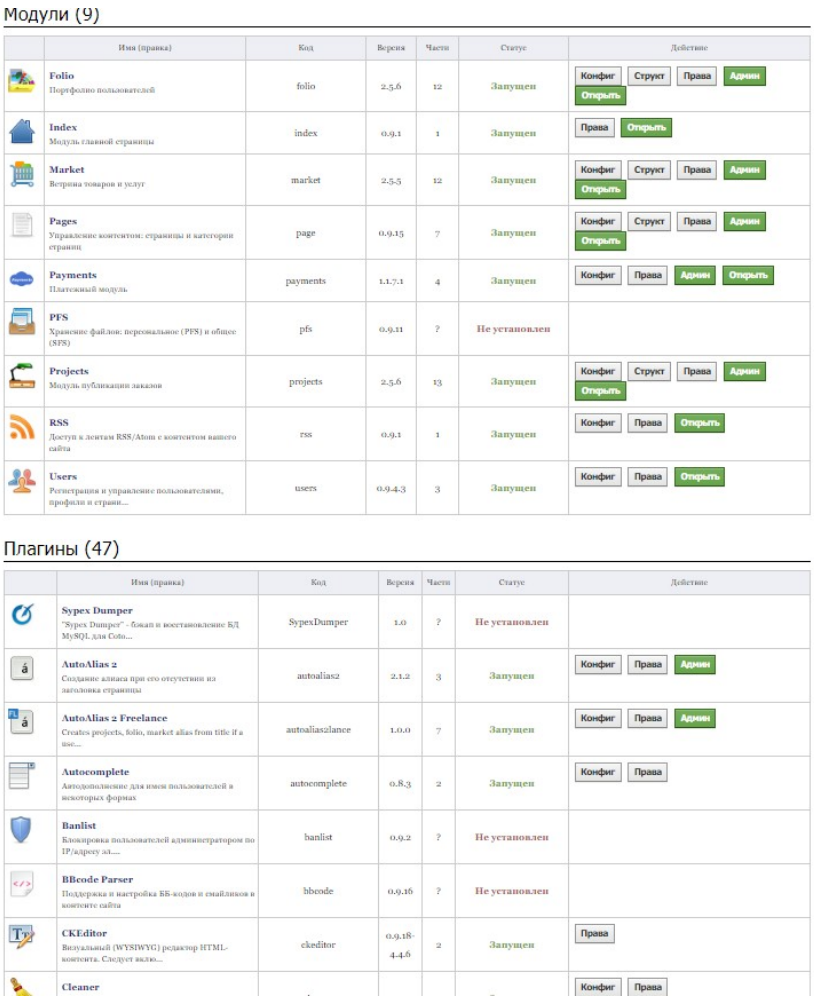

Рис. 3.8 Система модулей и плагинов

Рассмотрим ключевые:

Модули:

projectsmodule

```php
defined('COT_CODE') or die('Wrong URL');

require_once cot_incfile('projects', 'module');

if (!in_array($m, array('add', 'edit', 'preview', 'useroffers')))
{
    if (isset($_GET['id']) || isset($_GET['al']))
    {
        $m = 'main';
    }
    else
    {
        $m = 'list';
    }
}
require_once cot_incfile('projects', 'module', $m);

require_once $cfg['system_dir'].'/header.php';
echo $module_body;
require_once $cfg['system_dir'].'/footer.php';
```

Рис. 3.9 Модуль «Заявки к врачу»

Модуль page выводит страницы новостей

```php
defined('COT_CODE') or die('Wrong URL.');

// Environment setup
define('COT_PAGES', TRUE);
$env['location'] = 'pages';

// Additional API requirements
require_once cot_incfile('extrafields');

// Self requirements
require_once cot_incfile('page', 'module');

// Mode choice
if (!in_array($m, array('add', 'edit')))
{
    if (isset($_GET['id']) || isset($_GET['al']))
    {
        $m = 'main';
    }
    else
    {
        $m = 'list';
    }
}

require_once cot_incfile('page', 'module', $m);
```

Рис. 3.10 Модуль «Новости»

Модуль users выводит группы пользователей

```php
defined('COT_CODE') or die('Wrong URL.');

// Environment
define('COT_USERS', TRUE);
$env['location'] = 'users';

require_once cot_incfile('extrafields');
require_once cot_incfile('uploads');

require_once cot_incfile('users', 'module');

if (!in_array($m, array('details', 'edit', 'passrecover', 'profile', 'register')))
{
    $m = 'main';
}

include cot_incfile('users', 'module', $m);
```

Рис. 3.11 Модуль «Пользователи»

3.8 Разработка и структура таблиц баз данных

Перед тем как приступить к модулю, необходимо спроектировать базу данных (БД) с таблицами, так как практически каждая страница связывается с БД для добавления, удаления, обновления или поиска различных записей.

Для удобства создания и редактирования БД я использовал программу MySQLWorkbench.

После регистрации, вся информация о нем должны записываться в базу. Для этого я создал таблицу и дал название____. Таблица создается с помощью оператора CREATETABLE.

CREATE TABLE устанавливает новую таблицу, ее столбцы и ограничения целостности в существующей базе данных. Пользователь, который создает таблицу, становится владельцем таблицы и получает полные привилегии для этого, включая возможность предоставления (GRANT) привилегий другим пользователям, триггерам, и сохраненным процедурам.

CREATE TABLE поддерживает несколько опций для определения столбцов:

- Локальные столбцы определяющие имя и тип данных для данных введенных в столбец.

- Вычисляемые столбцы, базирующиеся на расширении. Значение столбца вычисляется каждый раз при доступе к таблице. Если тип данных не определен, *InterBase* вычисляет как соответствующий. Столбцы, к которым обращается выражение, должны существовать раньше, чем столбец может быть определен.

- Основанные на доменах столбцы, наследуемые все характеристики домена, но определение столбца может включать новое значение по умолчанию, атрибут NOT NULL, дополнительные ограничения CHECK или переопределять порядок сортировки, которые отменяют определение домена.

- Описание типа данных для столбца типа CHAR, VARCHAR или BLOB-тектможет включать предложение CHARACTER SET определяя специфическую кодировку для одиночного столбца. Иначе столбец использует определенную по умолчанию для базы данных кодировку. Если кодировка базы данных изменена, все столбцы впоследствии определенные имеют новую кодировку, но существующие столбцы не изменяются.

Предложение COLLATE позволяет указать специфический порядок сортировки для типов данных CHAR, VARCAHR и BLOB-текст. Выбор порядка сортировки ограничен теми которые поддерживаются для данной кодировки столбца, это или кодировка по умолчанию для базы данных в целом или другая установленная в предложении CHARACTER SET как часть определения типа данных. Смотри *LanguageReference,* чтобы получить полный список возможных сортировок.

Атрибут NOT NULL предотвращает ввод NULL или неизвестного значения в столбец. Атрибут NOT NULL проявляется во всех INSERT и UPDATE операциях над столбцом (листинг 3.1).

Листинг3.1

```
CREATE TABLE `pacient` (
```

```
  `id` int(6) NOT NULL AUTO_INCREMENT,
  `fio` varchar(100) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB AUTO_INCREMENT=184 DEFAULT CHARSET=cp1251 AUTO_INCREMENT=184;
```

В этой таблице, в общем, создал 25 полей, где хранится вся информация о пациенте, которая нужна для заполнения талона на прием к врачу. Структуру этой таблицы можно просмотреть на рисунке 3.11. С этой таблицей будут работать практически все страницы модуля.

Информация, в таблицу pacient, будет добавляться с помощью оператора INSERT. Синтаксис этого оператора представлен в листинге 2.2.

Листинг 3.2

INSERTINTO<название таблицы>([<Имя столбца>,...])**VALUES**(<Значение>,...)

Таким же образом с помощью запроса из модуля вставил значения в таблицу (листинг 3.3).

Листинг 3.3

INSERT INTO `pacient` VALUES (100, 'Биккулов Рушан Рафикович', '13.11.1988', 'г. Рузаевка, ул Ленина 45-55', '8909 234432, Отдел УФМС Республики Мордовия', '3214234234', '234432-2', 'РГС Медицина', 'Р-432234', '23-3212222', '12.12.2011', '12.12.2012', '12.12.2011', '12.12.2012', '54645645645', '1', 'Программист', 'НУЗ Узловая больница на станции Рузаевка', '321-4', 'работующий', '-', '-', 'rubik', '3f8da8d150df71f64db5f8e96438c567', 'ruzz_08@mail.ru');



Рисунок 3.11 - Структура таблицы pacient

Далее создал еще 6 таблиц:

- pacient – в этой таблицы храниться вся информация о пациенте после его регистрации;
- records – эта таблица используется для хранения записей пациентов;
- spec – тут хранятся имена всех специалистов;
- dolzhn – в этой таблице хранятся все должности врачей данной больницы;
- mail – сюда помещаются все онлайн консультации;
- time_rec – таблица для хранения общего времени работы специалиста.
- admins – в этой таблице хранятся логины и пароли для административного входа.

После создания всех таблиц приступил к созданиям связей между ними. Все связи между таблицами изображены на рисунке 3.12.

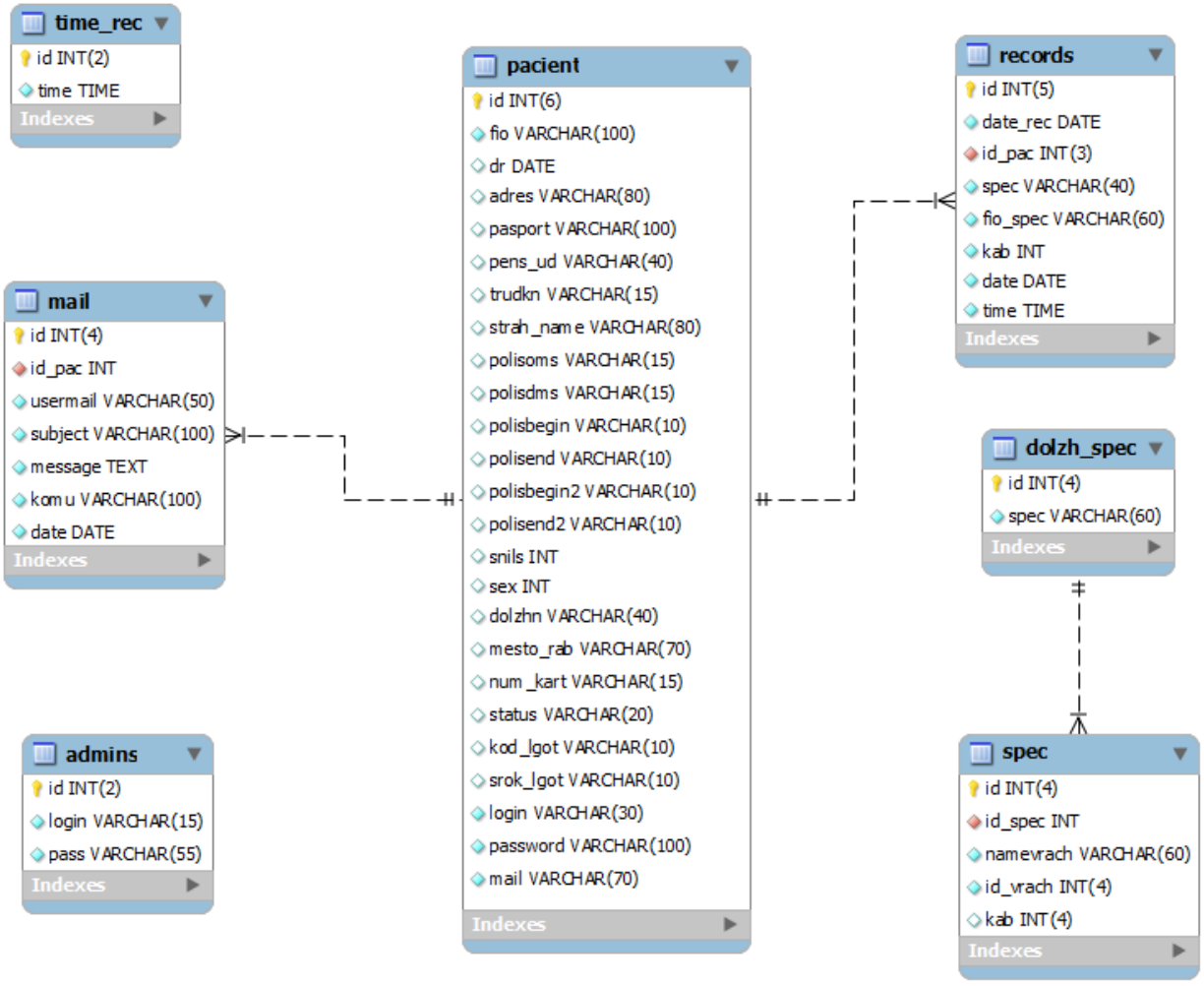

Рисунок 3.12 - ER-диаграмма базы данных online_record

Как видно из рисунка 3.12, все таблицы имеют поле id, которое является ключевым, то есть все значения в этом поле уникальные. Это необходимо для того, что бы осуществлять все основные операции, ориентируясь по полю id.

Чтобы таблица занимала меньше места, везде, где нужно было вставлять ФИО пациента, вставлял его id номер, который брал из таблицы pacient. Таким образом, связал таблицу pacient с mail и pacient с records отношением один ко многим. То есть в таблице mail и records создал поле id_pac и связал это поле с полем id в таблице pacient отношением один ко

многим, так как в таблице pacient поле id уникально, а в mail и records поле id_pac может повторяться.

Так же связал еще две таблицы отношением один ко многим – dolzh_spec и spec, где поле id из таблицы dolzhn ключевое, а значит уникальное, соединил с полем id_spec таблицы spec.

Оставшиеся две таблицы time_rec и admins остались не связанными и являются независящими.

Так же в этих таблицах идет обновление записей с помощью оператора UPDATE. В приложении А можно посмотреть всю структуру таблиц.

Синтаксис оператора можно просмотреть в листинге 3.4.

Листинг3.4

```
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name
   SET col_name1=expr1 [, col_name2=expr2 ...]
   [WHERE where_definition]
   [ORDER BY ...]
   [LIMIT rows]
или
UPDATE [LOW_PRIORITY] [IGNORE] tbl_name [, tbl_name ...]
   SET col_name1=expr1 [, col_name2=expr2 ...]
[WHERE where_definition]
```

Где в выражении SET указывается, какие именно столбцы следует модифицировать и какие величины должны быть в них установлены. В выражении WHERE, если оно присутствует, задается, какие строки подлежат обновлению. В остальных случаях обновляются все строки. Если задано выражение ORDER BY, то строки будут обновляться в указанном в нем порядке.

Если указывается ключевое слово LOW_PRIORITY, то выполнение данной команды UPDATE задерживается до тех пор, пока другие клиенты не завершат чтение этой таблицы.

Если указывается ключевое слово IGNORE, то команда обновления не будет прервана, даже если при обновлении возникнет ошибка дублирования ключей. Строки, из-за которых возникают конфликтные ситуации, обновлены не будут.

Если доступ к столбцу из указанного выражения осуществляется по аргументу tbl_name, то команда UPDATE использует для этого столбца его текущее значение.

Для удаления строк использовал оператор DELETE. Синтаксис этого оператора написан в листинге 3.5.

Листинг3.5

```
DELETE [LOW_PRIORITY] [QUICK] FROM table_name
    [WHERE where_definition]
    [ORDER BY ...]
[LIMITrows]
```

Если оператор DELETE запускается без определения WHERE, то удаляются все строки. При работе в режиме AUTOCOMMIT это будет аналогично использованию оператора TRUNCATE. «Синтаксис оператора TRUNCATE». В MySQL оператор DELETE без определения WHERE возвратит ноль как число удаленных записей.

Следует учитывать, что эта форма работает намного медленнее, чем DELETE FROM table_name без выражения WHERE, поскольку строки удаляются поочередно по одной.

Если указано ключевое слово LOW_PRIORITY, выполнение данной команды DELETE будет задержано до тех пор, пока другие клиенты не завершат чтение этой таблицы.

Если задан параметр QUICK, то обработчик таблицы при выполнении удаления не будет объединять индексы – в некоторых случаях это может ускорить данную операцию.

Выборка из таблиц осуществляется с помощью оператора SELECT. Я его применял, если необходимо что-либо найти в таблице или выбрать значения для дальнейшей с ней работой. Полный синтаксис оператора SELECT представлен в Листинге 3.6.

Листинг 3.6

```
SELECT [STRAIGHT_JOIN]
[SQL_SMALL_RESULT] [SQL_BIG_RESULT] [SQL_BUFFER_RESULT]
      [SQL_CACHE | SQL_NO_CACHE] [SQL_CALC_FOUND_ROWS]
[HIGH_PRIORITY]
    [DISTINCT | DISTINCTROW | ALL]
expression,...
  [INTO {OUTFILE | DUMPFILE} 'file_name' export_options]
  [FROM table_references
    [WHERE where_definition]
    [GROUP BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
    [HAVING where_definition]
    [ORDER BY {unsigned_integer | col_name | formula} [ASC | DESC], ...]
    [LIMIT [offset,] rows]
    [PROCEDURE procedure_name]
    [FOR UPDATE | LOCK IN SHARE MODE]]
```

Все параметры, начинающиеся с SQL_, STRAIGHT_JOIN и HIGH_PRIORITY, представляют собой расширение MySQL для ANSI SQL. Опции DISTINCT, DISTINCTROW и ALL указывают, будут ли возвращаться дублирующиеся записи. По умолчанию установлен параметр (ALL), т.е. возвращаются все встречающиеся строки. DISTINCT и DISTINCTROW

являются синонимами и указывают, что дублирующиеся строки в результирующем наборе данных должны быть удалены.

Выражение expression задает столбцы, в которых необходимо проводить выборку. Команда INTO OUTFILE 'file_name' осуществляет запись выбранных строк в файл, указанный в file_name. Данный файл создается на сервере и до этого не должен существовать. Для использования этой формы команды SELECT необходимы привилегии FILE. Если вместо INTO OUTFILE использовать INTO DUMPFILE, то MySQL запишет в файл только одну строку без символов завершения столбцов или строк.

После формирования БД необходимо страницы модуля соединить с ней. Для этого создал файл *bd.php* и помести в папку *include*. В этом файле с помощью функции *mysql_connect()* выполнил соединение к MySQL, где необходимо указать имя хостинга, имя и пароль. Далее выбрал необходимую базу данных с помощью функции *mysql_select_db()* (листинг 3.7).

Листинг 3.7

```
$hostname = "localhost";
$username = "root";
$password = "root";
$dbName = "online_record";

$db = mysql_connect($hostname,$username,$password) OR DIE("Чтототынапортачил");  //соединениекбазе
mysql_select_db($dbName, $db) or die(mysql_error());
```

Если на странице есть какой либо запрос к БД, то я добавляю в самом начале кода файл *bd.php*: **include**'bd.php';.

3.9 Интерфейс разработки

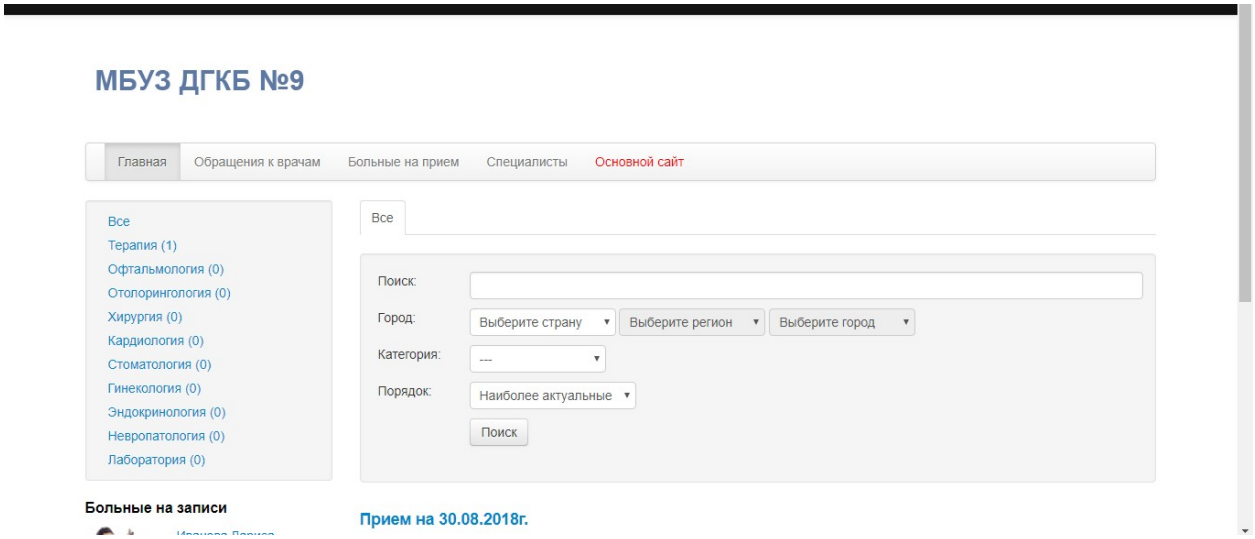

Рисунок 3.13 - Назначение кнопок в меню сайта

Меню «Главная» открывает страницу, где отображена вся информация о больнице, действующих врачах, текущий лист заявок к врачам.

Меню «Обращения к врачам» содержит информацию о текущих размещенных заявках о приеме врачей.

Меню «Новости» содержит информациюю о больнице, новостные материалы, полезную информацию по лечению и профилактике заболеваний.

Меню «Специалисты» отображает группы врачей в зависимости от профессиональной специализации.

Персональное меню отображает заявки выполненные больным на прием к тому или иному врачу,

Слева выведен список категорий специалистов. Отображаются текущие заявки на прием к врачам отдельных специальностей.

Меню «Администрирование» ведет в административную часть сайта.

Выходными данными из этой системы служит отчеты: Заявки к врачам, история 3.9.

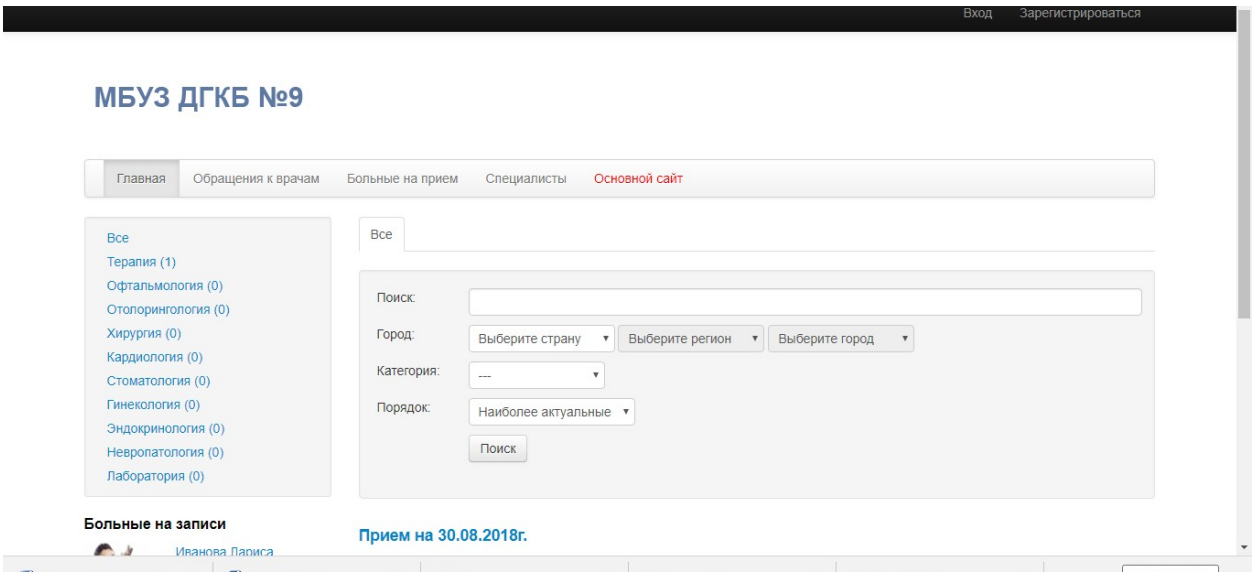

Рисунок 3.13 - Главная страница

Форма регистрации, авторизации пользователя представлена на рисунке 3.14.

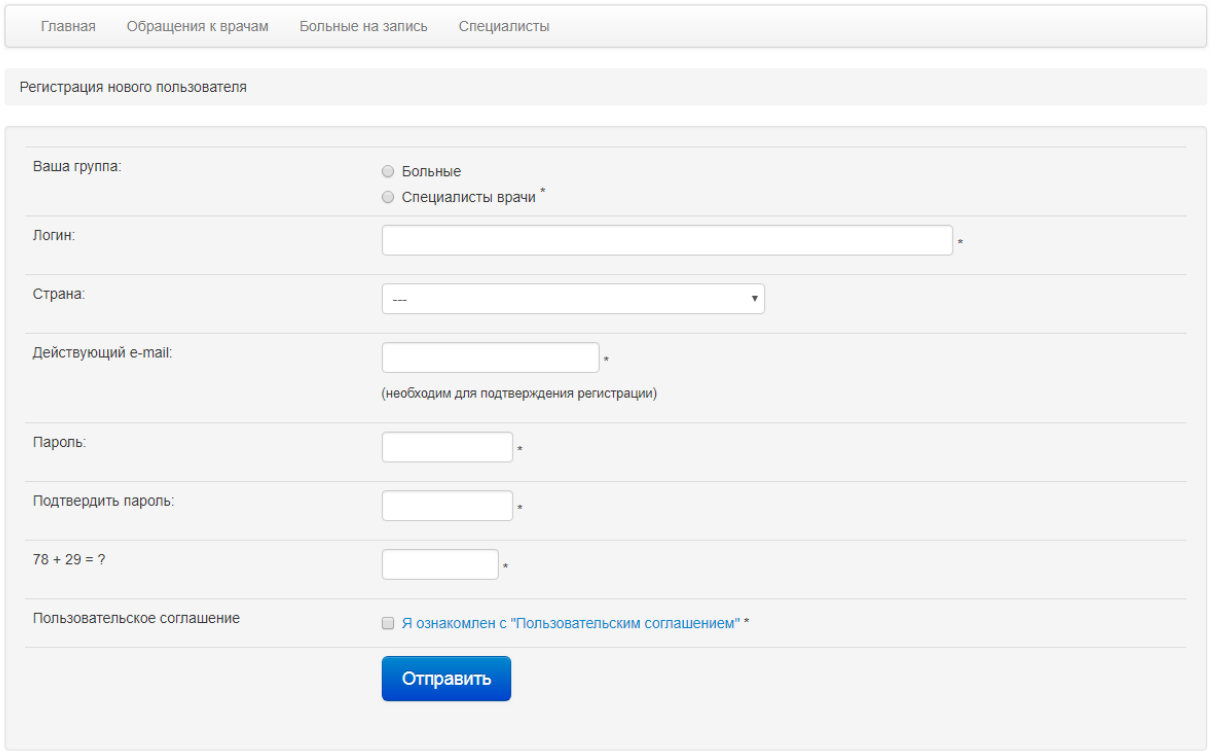

Рисунок 3.14 Форма регистрации

Форма настроек пользователя представлена на рисунке 3.15.

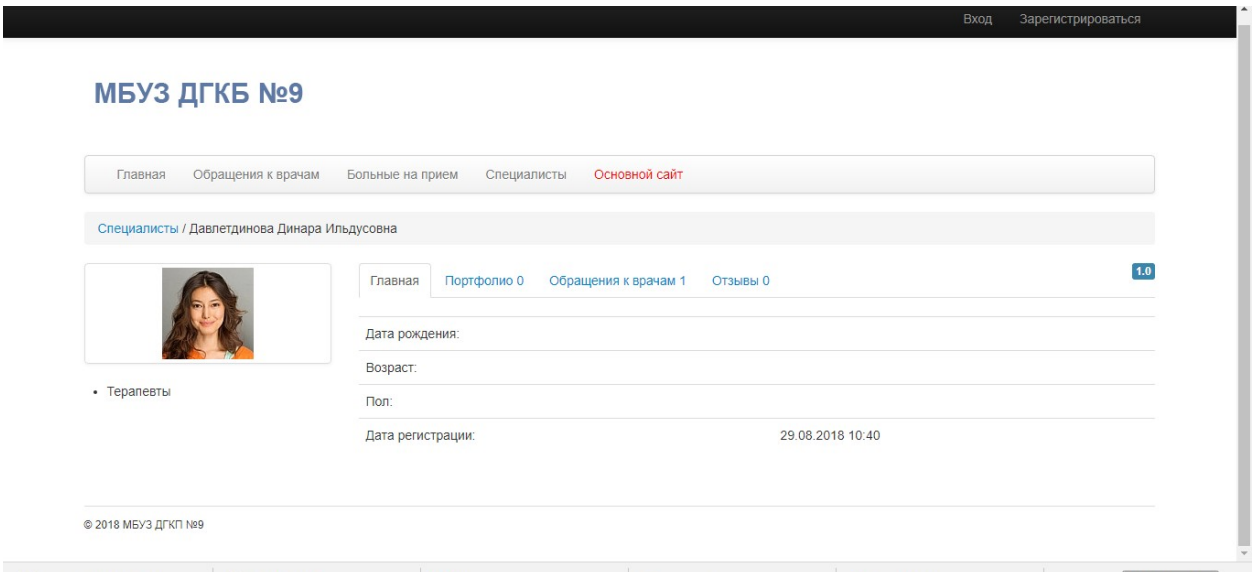

Рис. 3.15 Форма настроек пользователя

Форма размещения заявки на запись к врачу представлена на рисунке 3.16 .

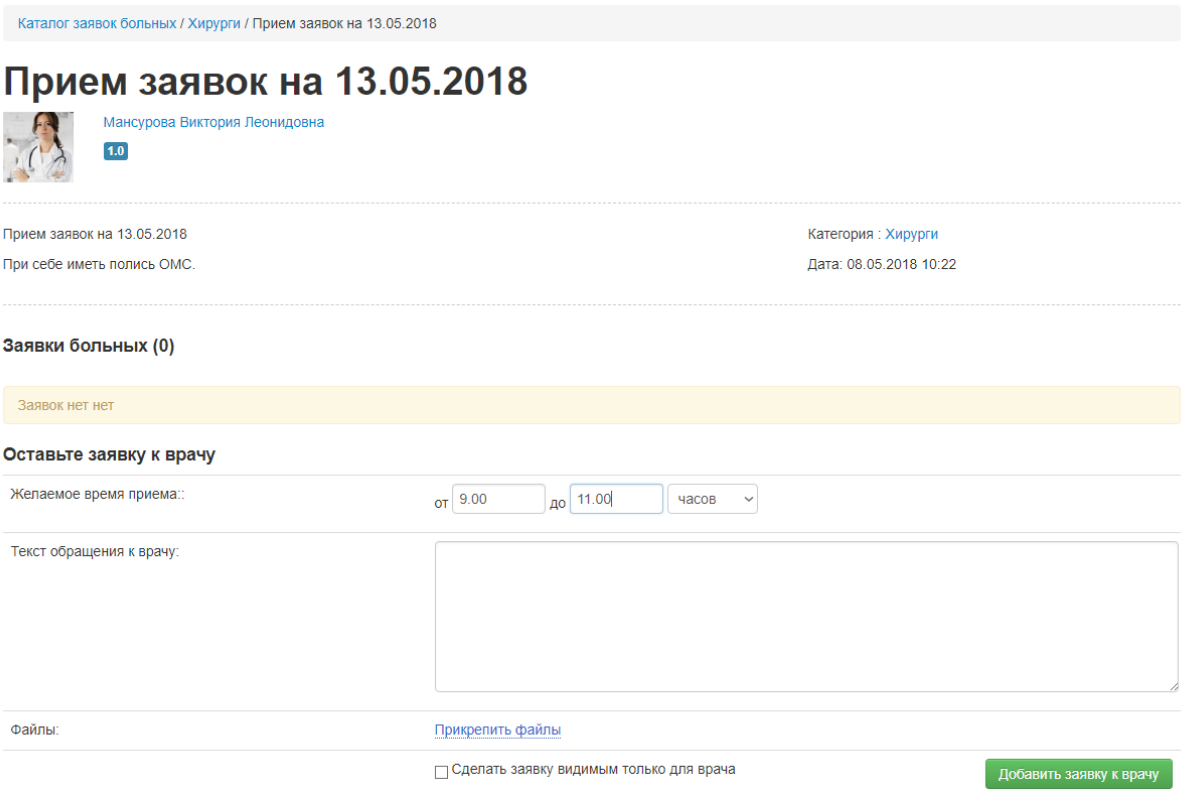

Рис. 3.16 Форма размещения заявки на прием к врачу

Врач размещает информацию о ведении приема на определенную дату. В форме имеется возможность указывать комментарии относительно формы

работы врача на определенный день, особые условия или определенное время приема врача.

Больные однажды зарегистрировавшись на сайте имеют возможность подписаться а отдельные группы специалистов.

К примеру больной Иванов Егор имеет проблемы со здоровьем зубов и ему необходимы услуги стоматологов. При регистрации больной выбирает подписку на определенную группу врачей.

При публикации врачами данных о приеме на определенный день, больному на почту приходит уведомление о том, что специалист «N» на такое то число ведет прием.

Отсюда пользователь может сразу перейти в форму приема заявок и оставить свою запись к врачу.

Далее приведем пример переписки по заявке на запись.

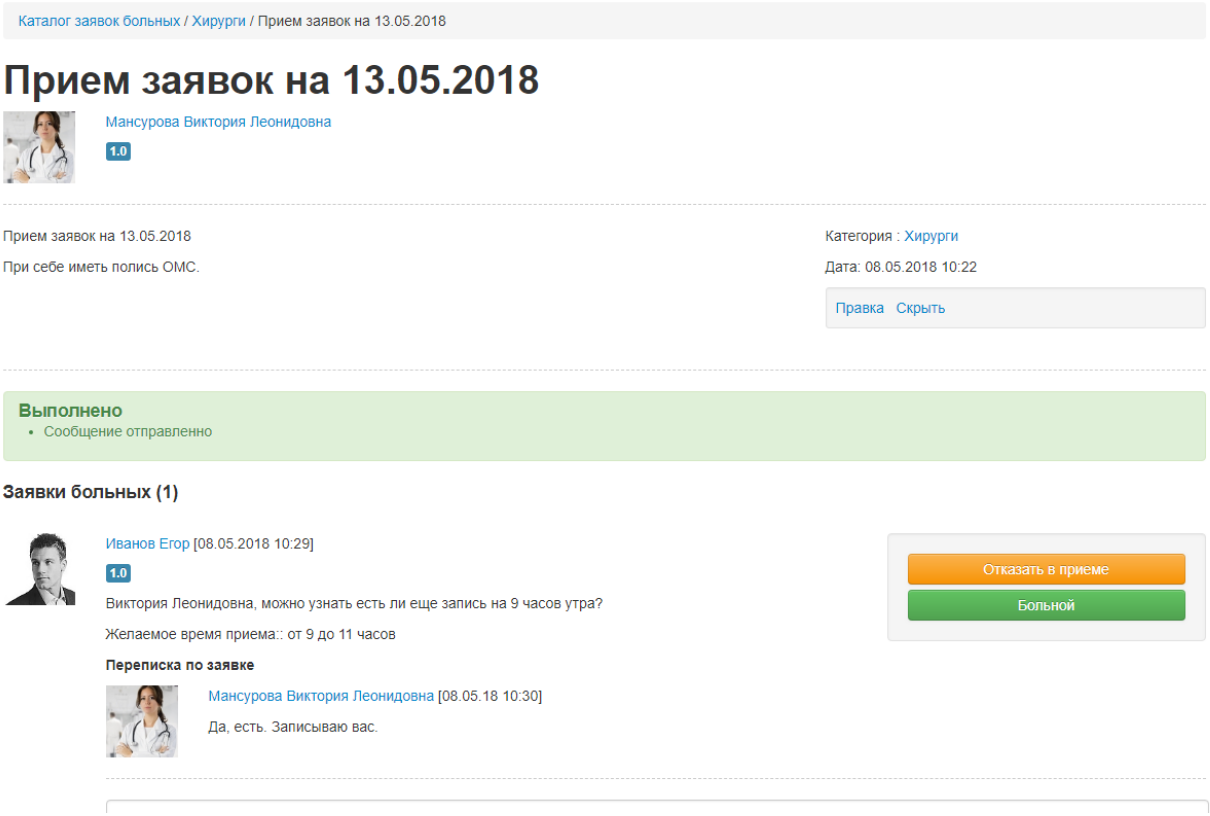

Рис. 3.17 Переписка по записи на прием

После публикации приема заявок на определенный день специалистом, на сайте открывается возможность записи на прием к этому специалисту.

Больные записываются к врачу посредством размещения заявки в этой форме. Функционально выведена возможность уточнять желательное время приема врача.

Врач имеет возможность определять принять больного или отказать в приеме.

После нажатия врачом кнопки «записать на прием», или «отказать в приме», больному в почтовый ящик приходит уведомление о решении врача.

Таким образом, разработанная нами информационная система «Онлайн запись к врачу» позволяет упростить и автоматизировать процедуру записи к нужному врачу.

Заключение

В рамках данного дипломного проекта была проделана следующая работа: проведен анализ современных требований к системам управления удаленных записей к врачам, спроектированы и разработаны основные концепции функционирования, добавлен весь дополнительный функционал, необходимый заказчику.

При разработке данного дипломного проекта постарался собрать весь имеющийся опыт подобных разработок, спроектировать и создать систему, которая могла бы максимально упростить и наладить процесс удаленной записи к специалистам через интернет.

При реализации были использованы современные технологии, такие как: AJAX –для создания наиболее дружественного и интерактивного интерфейса, язык MySQL без различных ORM – позволяющая снизить нагрузку и не предоставит проблем при доработке системы другими разработчиками. Аналогично был использован PHP, что также позволяет без проблем изменять программу другими разработчиками.

Для визуализации и внедрения функции адаптивности сайта был использован плагин Bootstrap 3.

Таким образом, разработанная нами информационная система «Онлайн запись к врачу» позволяет упростить и автоматизировать процедуру записи к нужному врачу.

Список используемой литературы

Учебная литература

1. Айзенмегер Р. HTML 3.2/4.0. Справочник. - СПб: Бином, 2008, 249с
2. Гагарина Л.Г., Кокорева Е.В., Виснадул Б.Д. Технология разработки программного обеспечения. М.: Форум, 2008. – 400 стр.
3. Гвоздева В. А. Основы построения автоматизированных информационных систем: учебник. – М.: ИНФРА-М, 2007. – 320 стр.
4. Горсиа-Молина Г., Ульман Д., Уилом Д. Системы баз данных. Полный курс. – М.: Вильямс ИД, 2014 г. – 1088 стр.
5. Грабер, М. Введение в SQL / М. Грабер. – М.: Изд. Лори, 2008. – 379 стр.
6. Инькова Е. А. Создание Web-сайтов». Тамбов, 2102 г.
7. Как создать свой блог http://wpnew.ru/
8. Кирсанов Д. «Веб-дизайн» –М.: Символ-Плюс, 2011.
9. Орлов Л.В. Как создать электронный магазин в Интернет – М.: Новый издательский Дом, 2014.
10. Панфилов, К. Создание веб-сайта от замысла до реализации / К. Панфилов. М.: ДМК Пресс, 2012. – 440 с
11. Полонская Е.Л. «Самоучитель по HTML 5.0» - 2016 г.
12. Пярин В.А., Кузьмин А.С, Смирнов С.И. Безопасность электронного бизнеса // Под.ред. проф. В.А. Манаева – М.: Гелиос АРВ, 2016.
13. Раздьяконов Н. и др. JumpIn: Учеб. пособие для слушателей курсов "Амадеус" – М.; СПб.: AmadeusRussia, 2016
14. Титов О. «Сделать сайт в оффлайне, от и до. Реально» Москва 2017 г.
15. Карпова Т.С. Базы данных: модели, разработка, реализация. Учебное пособие, СПБ: Питер – 2011 г. – 304с.
16. 14. Каймин В.А. "Интернет-Технологии". Учебное пособие. М.,

WDU. 2011, - 236с.

17. 15. Горев А., Макашарипов С., Ахаян Р. Эффективная работа с СУБД СПб, «Питер», 1997, 704с.

18. 16. Мартин Дж. Организация баз данных в вычислительных системах. – М.: Мир, 1980.

19. 17. Создание Intranet: Официальное руководство Microsoft / Под ред. В.Сергеева.пер. с англ.- Спб.: BHV – Санкт-Петербург, 1998, - 672с.

20. 18. Электронная встроенная гипертекстовая справочная система «HTML и CSS», 2011 г.

21. 19. Спецификация CSS. http://www.w3.org/TR/REC-CSS2

Интернет источники

22. http://www.e-commerce.ru/ — Интернет-ресурсы информационно-консалтингового центра по электронному бизнесу.

23. http://www.e-management.ru/ — консультационный центр развития электронного бизнеса.

24. http://www.marketingpower.com/welcome.php — сайт американской маркетинговой ассоциации (AMA).

25. http://www.ram.ru/ — сайт российской ассоциации маркетинга (PAM).

26. http://www.telenews.ru/ibiz — iBusiness.

27. http://www.emarketer.com/arts/eglobal — агентство e-marketer.

28. Хитрости WEB-дизайна, http://www.khv.ru/redirect.cfm?CFID=136786&CFTOKEN=78992983&CFApp=2&Message_ID=1245

29. Основы создания сайта, http://coolhitman.narod.ru/osnovsaitaftp.htm.

30. Статья о хостинге, http://www.webmasterpro.com.ua/articles/article29.html.

Приложения

Основные файлы шаблона модуля «Запис к врачу»

header.tpl

```
<!-- BEGIN: HEADER -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" >
<head>
<title>{HEADER_TITLE}</title>
<!-- IF {HEADER_META_DESCRIPTION} --><meta name="description"
content="{HEADER_META_DESCRIPTION}" /><!-- ENDIF -->
<!-- IF {HEADER_META_KEYWORDS} --><meta name="keywords"
content="{HEADER_META_KEYWORDS}" /><!-- ENDIF -->
<meta http-equiv="content-type" content="{HEADER_META_CONTENTTYPE};
charset=UTF-8" />
<meta name="generator" content="Cotonti http://www.cotonti.com" />
<link rel="canonical" href="{HEADER_CANONICAL_URL}" />
{HEADER_BASEHREF}
{HEADER_HEAD}
<link rel="shortcut icon" href="favicon.ico" />
<link rel="apple-touch-icon" href="apple-touch-icon.png" />
</head>

<body>

	<!-- IF {PHP.usr.id} == 0 -->
	<div id="AuthModal" class="modal hide fade">
		<div class="modal-header">
			<h3 id="myModalLabel">{PHP.L.Login}</h3>
			</div>
			<div class="modal-body">
				<form class="form-horizontal"
action="{HEADER_GUEST_SEND}" method="post">
					<div class="control-group">
						<label class="control-label"
for="inputEmail">{PHP.L.users_nameormail}</label>
						<div class="controls">
							<input type="text" name="rusername"
id="inputEmail" />
						</div>
					</div>
					<div class="control-group">
						<label class="control-label"
for="inputPassword">{PHP.L.Password}</label>
						<div class="controls">
							<input type="password"
name="rpassword" id="inputPassword" /><br/>
							<a rel="nofollow" class="link small"
href="{PHP|cot_url('users','m=passrecover')}">{PHP.L.users_lostpass}</a>
						</div>
					</div>
```

```
                                        <div class="control-group">
                                                <div class="controls">
                                                        <label class="checkbox">
                                                        {HEADER_GUEST_COOKIETTL}
{PHP.L.users_rememberme}
                                                        </label><br/>
                                                        <button type="submit" class="btnbtn-
primary btn-large">{PHP.L.Login}</button>
                                                </div>
                                        </div>
                                </form>
                        </div>
                        <div class="modal-footer">
                        <button class="btn" data-dismiss="modal" aria-
hidden="true">{PHP.L.Close}</button>
                </div>
        </div>
        <!-- ENDIF -->

        <div id="wrapper" class="container">
                <div class="navbarnavbar-inverse navbar-fixed-top">
                        <div class="navbar-inner">
                                <div class="container">
                                        <ul class="nav pull-right">
                                                <!-- BEGIN: GUEST -->
                                                <li><a href="{PHP|cot_url('login')}" data-
toggle="modal" onClick="$('#AuthModal').modal(); return false;">{PHP.L.Login}</a></li>
                                                <li><a href="{PHP|
cot_url('users','m=register')}">{PHP.L.Register}</a></li>
                                                <!-- END: GUEST -->

                                                <!-- BEGIN: USER -->
                                                <li><a href="{PHP.usr.name|
cot_url('users', 'm=details&u='$this)}">{PHP.usr.name}</a></li>
                                                <li><a href="{PHP|cot_url('users',
'm=profile')}">{PHP.L.Profile}</a></li>
                                                <!-- IF {PHP.cot_modules.payments}
AND {PHP.cfg.payments.balance_enabled} -->
                                                <li><a
href="{HEADER_USER_BALANCE_URL}">{PHP.L.payments_mybalance}:
{HEADER_USER_BALANCE|number_format($this, '2', '.', ' ')}
{PHP.cfg.payments.valuta}</a></li>
                                                <!-- ENDIF -->
                                                <!-- IF {PHP.cot_modules.projects} -->
                                                <li class="dropdown">
                                                        <a class="dropdown-toggle" data-
toggle="dropdown" href="#">{PHP.L.projects_projects}<b class="caret"></b></a>
                                                        <ul class="dropdown-menu">
                                                                <!-- IF {PHP|
cot_auth('projects', 'any', 'W')} -->
```

```
                                                    <li><a href="{PHP.usr.id|
cot_url('users', 'm=details&id='$this'&tab=projects')}">{PHP.L.projects_myprojects}</
a></li>
                                                    <!-- ENDIF -->
                                                    <!-- IF
{PHP.cot_plugins_active.sbr} -->
                                                    <li><a href="{PHP|
cot_url('sbr')}">{PHP.L.sbr_mydeals}</a></li>
                                                    <!-- ENDIF -->
                                                    <!-- IF {PHP|
cot_auth('projects', 'any', '1')} -->
                                                    <li><a href="{PHP|
cot_url('projects', 'm=useroffers')}">{PHP.L.offers_useroffers}</a></li>
                                                    <!-- ENDIF -->
                                                </ul>
                                            </li>
                                            <!-- ENDIF -->
                                            <!-- IF {PHP.cot_modules.market} -->
                                            <li class="dropdown">
                                                <a class="dropdown-toggle" data-
toggle="dropdown" href="#">{PHP.L.market}<b class="caret"></b></a>
                                                <ul class="dropdown-menu">
                                                    <li><a href="{PHP.usr.id|
cot_url('users', 'm=details&id='$this'&tab=market')}">{PHP.L.market_myproducts}</a></
li>
                                                    <!-- IF
{PHP.cot_plugins_active.marketorders} -->
                                                    <li><a href="{PHP|
cot_url('marketorders', 'm=sales')}">{PHP.L.marketorders_mysales}</a></li>
                                                    <li><a href="{PHP|
cot_url('marketorders', 'm=purchases')}">{PHP.L.marketorders_mypurchases}</a></li>
                                                    <!-- ENDIF -->
                                                </ul>
                                            </li>
                                            <!-- ENDIF -->
                                            <!-- IF {PHP.cot_plugins_active.paypro}
-->
                                            <li>
                                                <!-- IF
{HEADER_USER_PROEXPIRE} -->
                                                <a href="{PHP|cot_url('plug',
'e=paypro')}"
title="{PHP.L.paypro_header_extend}">{PHP.L.paypro_header_expire_short}
{HEADER_USER_PROEXPIRE|cot_date('d.m.Y', $this)}</a>
                                                <!-- ELSE -->
                                                <a href="{PHP|cot_url('plug',
'e=paypro')}" title="{PHP.L.paypro_header_buy}">{PHP.L.paypro_header_buy}</a>
                                                <!-- ENDIF -->
                                            </li>
                                            <!-- ENDIF -->
```

```
                                                <!-- IF
{HEADER_USER_PMREMINDER} --><li>{HEADER_USER_PMREMINDER}</li><!--
ENDIF -->
                                                <!-- IF {HEADER_NOTICES} -->
                                                <li class="dropdown">
                                                        <a class="dropdown-toggle" data-
toggle="dropdown" href="#">{PHP.L.header_notice}<b class="caret"></b></a>
                                                        <ul class="dropdown-menu">
                                                                {HEADER_NOTICES}
                                                        </ul>
                                                </li>
                                                <!-- ENDIF -->

        <li>{HEADER_USER_ADMINPANEL}</li>
                                                <li>{HEADER_USER_LOGINOUT}</li>
                                                <!-- END: USER -->
                                        </ul>
                                </div>
                        </div>
                </div>

                <div id="header" class="row">
                        <div class="span4">
                                <div class="logo"><a href="{PHP|cot_url('index')}"
title="{PHP.cfg.maintitle} {PHP.cfg.separator}
{PHP.cfg.subtitle}"><imgsrc="themes/{PHP.theme}/img/logo.png"/></a></div>
                        </div>
                        <div class="span5">

                        </div>
                        <div class="span3 textright paddingtop10">
                                <!-- BEGIN: I18N_LANG -->
                                        <!-- BEGIN: I18N_LANG_ROW -->
                                        <a href="{I18N_LANG_ROW_URL}"
class="{I18N_LANG_ROW_CLASS}"><imgsrc="images/flags/{I18N_LANG_ROW_FLA
G}.png"/></a> 
                                        <!-- END: I18N_LANG_ROW -->
                                <!-- END: I18N_LANG -->
                        </div>
                </div>

                <div class="navbar">
                        <div class="navbar-inner">
                                <ul class="nav">
                                        <li<!-- IF {PHP.env.ext} == 'index' -->
class="active"<!-- ENDIF --><a href="{PHP|cot_url('index')}">{PHP.L.Home}</a></li>
                                        <!-- IF {PHP.cot_modules.projects} -->
                                        <li<!-- IF {PHP.env.ext} == 'projects' -->
class="active"<!-- ENDIF --><a href="{PHP|
cot_url('projects')}">{PHP.L.projects_projects}</a></li>
                                        <!-- ENDIF -->
```

```
                                        <li<!-- IF {PHP.env.ext} == 'users' AND ({PHP.group}
== {PHP.cot_groups.4.alias} AND {PHP.m} == 'main' --> class="active"<!-- ENDIF --
>><a href="{PHP.cot_groups.4.alias|cot_url('users',
'group='$this)}">{PHP.cot_groups.4.name}</a></li>
                                        <li<!-- IF {PHP.env.ext} == 'users' AND ({PHP.group}
== {PHP.cot_groups.7.alias} AND {PHP.m} == 'main' --> class="active"<!-- ENDIF --
>><a href="{PHP.cot_groups.7.alias|cot_url('users',
'group='$this)}">{PHP.cot_groups.7.name}</a></li>
                                        <!-- IF {PHP.cot_modules.market} -->
                                        <li<!-- IF {PHP.env.ext} == 'market' AND !{PHP.type} --
> class="active"<!-- ENDIF -->><a
href="{PHP|cot_url('market')}">{PHP.L.market}</a></li>
                                        <!-- ENDIF -->
                                    </ul>
                                </div>
                            </div>

                            <div id="main" class="content">

<!-- END: HEADER -->
```

Index.tpl

```
<!-- BEGIN: MAIN -->
<div class="row">
	<div class="span12">
		<!-- IF {PHP.cot_plugins_active.paytop} -->
		{PHP|cot_get_paytop ('top')}
		<!-- ENDIF -->
	</div>
</div>
<div class="row">
	<div class="span3">
		<div class="well well-small">{PROJECTS_CATALOG}</div>

		<!-- IF {PHP.cot_plugins_active.userpoints} -->
		<h4 class="mboxHD">Больные на записи</h4>
		{PHP|cot_get_topusers (4, 5)}
		<br/>
		<br/>
		<h4 class="mboxHD">Специалисты на приеме</h4>
		{PHP|cot_get_topusers (7, 5)}
		<!-- ENDIF -->

		<!-- IF {INDEX_NEWS} -->
		<br/>
		<br/>
		<div class="block">
			{INDEX_NEWS}
		</div>
		<!-- ENDIF -->
	</div>
	<div class="span9">
		{PROJECTS_SEARCH}

		<!-- IF {PHP.cot_plugins_active.paypro} -->
			<!-- IF !{PHP|cot_getuserpro()} AND
{PHP.cfg.plugin.paypro.projectslimit} > 0 AND {PHP.cfg.plugin.paypro.projectslimit} <=
{PHP.usr.id|cot_getcountprjofuser($this)} -->
			<div class="alert alert-
warning">{PHP.L.paypro_warning_projectslimit_empty}</div>
			<!-- ENDIF -->
			<!-- IF !{PHP|cot_getuserpro()} AND
{PHP.cfg.plugin.paypro.offerslimit} > 0 AND {PHP.cfg.plugin.paypro.offerslimit} <=
{PHP.usr.id|cot_getcountoffersofuser($this)} -->
			<div class="alert alert-
warning">{PHP.L.paypro_warning_offerslimit_empty}</div>
			<!-- ENDIF -->
		<!-- ENDIF -->
				{PROJECTS}
	</div>
</div>
	<!-- END: MAIN -->
```

footer.tpl

```
<!-- BEGIN: FOOTER -->

        </div>
</div>
<div id="footer" class="container">
        &copy; {PHP.sys.now|cot_date('Y', $this)} {PHP.cfg.maintitle}
</div>

<script type="text/javascript"
src="themes/{PHP.theme}/bootstrap/js/bootstrap.min.js"></script>
{FOOTER_RC}
</body>
</html>
<!-- END: FOOTER -->
```