МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего профессионального образования

**«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

**(ФГБОУ ВПО «КубГУ»)**

**Кафедра вычислительных технологий**

**КУРСОВАЯ РАБОТА**

**РАЗРАБОТКА АДАПТИВНОГО ВЕБ-ЧАТА**

Работу выполнил \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ К. А. Степаненко (подпись, дата)

Факультет компьютерных технологий и прикладной математики, курс 3

Направление 02.03.02 – «Фундаментальная информатика и информационные технологии»

Научный руководитель к.т.н.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Т. А. Приходько. (подпись, дата)

Нормоконтролер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Ю. С. Фисун

(подпись, дата)

Краснодар 2016

СОДЕРЖАНИЕ

ВВЕДЕНИЕ………………………………………………………………………..3

1. Отличие адаптивной версии от мобильного приложения и основные типы адаптивных макетов….………………………………………..………………….4

2. CSS-фреймворки и их сравнение……………………...………………………7

3. Безопасность веб-приложений…………………………..…………………...10

4. Большие данные. Визуализация данных…………………………………….15

5. Разработка чата………………………………………………………………..21

ЗАКЛЮЧЕНИЕ………………………………………………………………….30

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ……………………………………………...31

ВВЕДЕНИЕ

В настоящее время существует множество устройств, которыми люди пользуются, в том числе, и для того, чтобы выходить в Интернет. Все эти устройства различаются размером экрана, разрешением и, соответственно, тем, как может отображаться на них веб-страница. Поэтому важно, чтобы веб-приложение или веб-сайт хорошо смотрелись и правильно отображались у любого из пользователей, независимо от того, какое устройство он использует.

С ростом популярности мобильных устройств количества пользователей, которые заходят с них на сайты, заметно увеличилось, поэтому просто игнорировать их уже нельзя – это не один-два человека в полгода, это значительная часть аудитории, и им должно быть удобно пользоваться веб-сайтом или веб-приложением.

Данная проблема решается с помощью создания адаптивного веб-дизайна. Это достаточно новое направление в дизайне веб-ресурсов, но уже сейчас оно является одним из главных показателей качества веб-приложения или сайта.

1. Отличие адаптивной версии от мобильного приложения и основные типы адаптивных макетов

Мобильные приложения, специально разработанные для различных мобильных устройств, также решают проблему с удобством просмотра сайта, но имеют некоторые недостатки:

а) Под каждый тип операционной системы нужно своё приложение.

б) Необходимость загрузки.

 Для того, чтобы пользоваться приложением, пользователям необходимо его загрузить. Это требует каких-то дополнительных усилий от пользователей, и многие не будут этого делать, если точно не уверены, что приложение им очень нужно и они планируют регулярно его использовать.

Проектирование было начато с адаптивной версии веб-чата для мобильных устройств. На этом этапе нужно было правильно передать смысл и основные идеи с использованием небольшого экрана и всего одной колонки.

Были рассмотрены основные типы адаптивных макетов.

* 1. Резиновый

Данный макет простой в реализации и очевидный для пользователя тип представления веб-страницы. Основные блоки сжимаются до ширины экрана мобильного устройства, где такое невозможно — перестраиваются в одну длинную ленту.

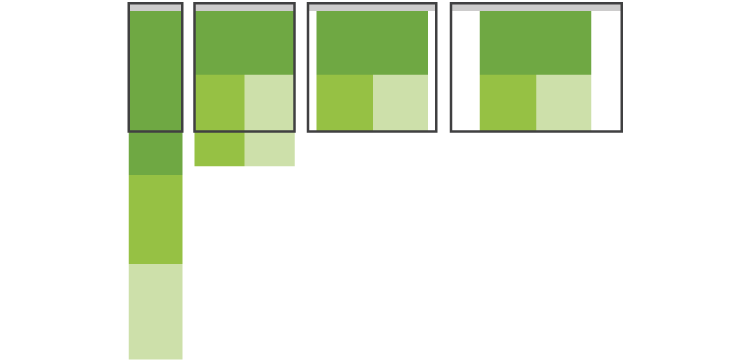


Рисунок 1. Резиновый адаптивный макет.

* 1. Перенос блоков

Этот тип подходит для многоколоночной веб-страницы: при уменьшении ширины экрана дополнительные блоки переносятся в нижнюю часть макета.

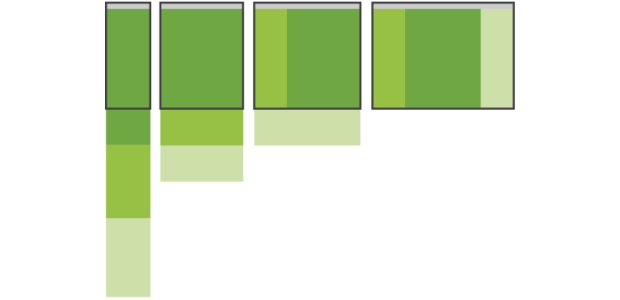


Рисунок 2. Перенос блоков.

* 1. Переключение макетов

Этот способ наиболее удобен при чтении сайта с различных устройств: под каждое разрешение экрана разрабатывается отдельный макет. Способ трудоемок, поэтому менее популярен, чем предыдущие два.

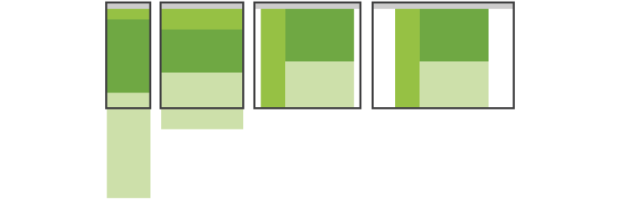


Рисунок 3. Переключение макетов

1.4 Адаптивность масштабированием

Очень простой способ, который подходит для несложных веб-страниц. Достигается элементарным масштабированием изображений. Не очень популярный способ, так как не обладает гибкостью.

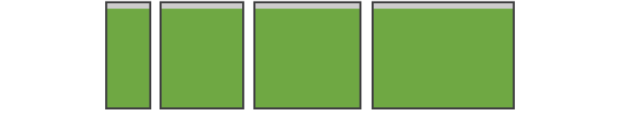


Рисунок 4. Адаптивность масштабированием

При проектировании веб-дизайна для чата был выбран способ переключения макетов. Благодаря данному способу имеется возможность сохранить доступ ко всем функциям чата, независимо от того с какого устройства был выполнен вход.

1. CSS-фреймворки и их сравнение

На сегодняшний день существует очень много фреймворков. Это касается веб-программирования серверной части, JavaScript фреймворков клиентской части, а также во многих других сферах. В общем случае, под фреймворком понимается некоторая кодовая база, решающая множество стандартных задач, и задающая архитектуру и стиль создания приложения.

 Сейчас существует множество фреймворков, которые обладают различным набором элементов. Все фреймворки можно разделить на некоторые подтипы:

- простые CSS-фреймворки

- веб-компоненты

- комплексные CSS-фреймворки

Был проведен анализ данных подтипов, с помощью которого удалось выявить что же включает в себя каждый подтип фреймворков и какими свойствами он обладает.

2.1 Простые CSS-фреймворки

Главное отличие фреймворков данного типа в том, что в них не используется JavaScript, а все модули пишутся только с помощью HTML и CSS. Данные фреймворки просты в изучении и научиться работать с ним можно довольно быстро. Главный минус простых CSS-фреймворков – недостаточный набор функциональных возможностей из-за отсутствия JavaScript-компонентов. Такие фреймворки могут включать в себя следующие модули: сетки, кнопки, таблицы, элементы форм, примитивы, меню и некоторые вспомогательные элементы. К простым CSS-фреймворкам относятся: Skeleton, Kube, Pure CSS и другие.

2.2 Веб-компоненты

Веб-компоненты — это комплекс стандартов, который добавляет в браузер технологию для удобной реализации UI-решений. Каждый такой элемент будет работать в своем личном DOM-дереве без перекрытия стилями основной страницы. Это одно из главных преимуществ веб-компонент, так как позволяет поддерживать ему свойство инкапсуляции.

2.3 Комплексные CSS-фреймворки

Это самые масштабные фреймворки. Они обладают как модулями на JavaScript, так и модулями, реализованными с помощью CSS и HTML. Данные фреймворки способны решить большинство задач по созданию веб-интерфейса. Обычно, в таких фреймворках присутствует следующий набор модулей: сетки, диалоговые окна, примитивы, пагинации, панели, карточки, шрифты с векторными иконками, списки данных, кнопки, таблицы, меню, элементы форм и многое другое. К комплексным CSS-фреймворкам относятся: Bootstrap3, Foundation, Semantic-UI и другие. Хотелось бы отметить, что фреймворк, который использовался для создания веб-интерфейса для чата относится к данному подтипу.

Для того, чтобы определиться какой фреймворк лучше использовать, было проведено сравнение трёх наиболее популярных фреймворков: Bootstrap3, Foundation и Skeleton. Результаты сравнения приведены на следующем скриншоте:



Рисунок 5. Сравнение популярных CSS-фреймворков [3].

Так же при сравнении была учтена поддержка CSS-фреймворка различными браузерами.



Рисунок 6. Поддержка браузерами фреймворков Bootstrap3, Foundation и Skeleton [3].

1. Безопасность веб-приложений

При разработке веб-чата было уделено большое внимание безопасности веб-приложения. Был рассмотрен список из самых опасных векторов атак на веб-приложения, созданный OWASP (Open Web Application Security Project). Далее будут описаны те уязвимости, которые были обнаружены в процессе разработки веб-чата и его тестировании.

* 1. Небезопасные прямые ссылки на объекты

Данный вид уязвимости является следствием недостаточной проверки пользовательских данных или неправильной реализации методов работы с информационными объектами и может привести к тому, что авторизованный пользователь веб-приложения может получить неавторизованный доступ к привилегированным функциям и данным. Суть уязвимости заключается в том, что при выводе каких-либо конфиденциальных данных, например, личных сообщений, для доступа к объекту используется идентификатор, который передается в открытом виде в адресной строке браузера, и не реализована проверка прав доступа к объектам. Например, есть страница, которая отображает личное сообщение и она имеет адрес вида:  
example.com/message.php?mid=1234, перебирая число после "mid=" можно будет читать чужие личные сообщения. Эксплуатация данной уязвимости очень проста и не требует никаких специальных навыков – достаточно лишь перебирать число в адресной строке браузера.

При разработке чата были устранены подобные уязвимости, например, доступ к нескольким комнатам одновременно.

* 1. Небезопасная конфигурация

Безопасность веб-приложения требует наличия безопасной конфигурации всех компонентов инфраструктуры: компонентов приложения (например, фреймворки), веб-сервера, сервера баз данных и самой платформы. Настройки компонентов сервера по умолчанию зачастую небезопасны и открывают возможности к атакам. Например, кража сессионной cookie через JavaScript при XSS-атаке становится возможна благодаря выключенной по умолчанию настройке cookie\_httponly. При правильной настройке сервера и включенной опции cookie\_httponly, получить сессионную cookie через JavaScript невозможно.

Кроме того, программное обеспечение должно быть в актуальном состоянии: уязвимости находят каждый день в самых различных программных компонентах – операционной системе, веб-серверах, серверах баз данных, почтовых серверах и т.д. И даже если приложение правильно написано и тщательно проверяет все входящие данные, и вообще, хорошо защищено, это не означает что в один момент не найдется уязвимость в ОС или веб-сервере. Примером данной уязвимости является вывод сообщений об ошибках, таким образом злоумышленник может лучше понять структуру базы данных и с помощью ошибок понять, в каком направлении лучшее действовать, какие команды и сценарии лучше пытаться внедрять. Например, сообщение об ошибке может информировать злоумышленника о том, что в результате выполнения вредоносного SQL-запроса было получено верное имя базы данных, но неверное количество полей.

В веб-чате была учтена данная уязвимость и проведена настройка соответствующих параметров веб-сервера.

* 1. Отсутствие функций контроля доступа

Суть уязвимости заключается в отсутствии проверки наличия надлежащего доступа к запрашиваемому объекту. Большинство веб-приложений проверяют права доступа, прежде чем отобразить данные в пользовательском интерфейсе. Тем не менее, приложения должны выполнять те же проверки контроля доступа на сервере при запросе любой функции. Ведь есть еще множество вспомогательных служебных запросов, которые, зачастую отправляются в фоновом режиме асинхронно, при помощи технологии AJAX. Если параметры запроса недостаточно тщательно проверяются, злоумышленники смогут подделать запрос для доступа к данным без надлежащего разрешения.

В начальной версии чата было обнаружено множество уязвимостей данного типа, все они были устранены путем добавления функции проверки прав доступа, которая выполняется при вызове любых других функций чата.

* 1. Использование компонентов с известными уязвимостями

Любой сторонний компонент веб-приложения – будь то двоичный или исходный код, коммерческое или приложение с открытым кодом – должны проверяться на отсутствие уязвимостей. Зачастую веб-приложения написаны с использованием специальных библиотек или «фреймворков», которые поставляются сторонними компаниями. В большинстве случаев эти компоненты имеют открытый исходный код, а это означает, что они есть у миллионов людей во всем мире, которые изучают исходный код, в том числе, и на предмет уязвимостей. Также уязвимости ищут (и находят) в более низкоуровневых компонентах системы, таких как сервер базы данных, веб-сервер, и наконец, компоненты операционной системы вплоть до ее ядра.  
Очень важно использовать последние версии компонентов и следить за появляющимися известными уязвимостями на специально предназначенных для этого сайтах. Например, в 2014 году в нескольких версиях библиотеки OpenSSL (криптографический пакет с открытым исходным кодом для работы с SSL/TLS) была обнаружена критическая уязвимость Heartbleed, которая позволяла несанкционированно читать память на сервере или на клиенте, в том числе для извлечения закрытого ключа сервера.

* 1. Непроверенные переадресации и пересылки

Веб-приложения зачастую переадресуют пользователя с одной страницы на другую. В процессе могут использоваться ненадлежащим образом проверяемые параметры с указанием страницы конечного назначения переадресации. Без соответствующих проверок, атакующий может использовать такие страницы для переадресации жертвы на вредоносную веб-страницу, которая, к примеру, может иметь очень схожий или неотличимый интерфейс, но украдет ваши данные.  
Данный вид уязвимостей схож с тем, который описан в пункте №3, соответственно и устранить данную уязвимость можно таким же способом.

* 1. Инъекции

Все данные, как правило, хранятся в специальных базах, обращения к которым строятся в виде запросов, чаще всего написанных на специальном языке запросов SQL (Structured Query Language – структурированный язык запросов). Приложения используют SQL-запросы для того, чтобы получать, добавлять, изменять или удалять данные, например, при редактировании пользователем своих личных данных или заполнении анкеты на сайте. При недостаточной проверке данных от пользователя, злоумышленник может внедрить в форму веб-интерфейса приложения специальный код, содержащий SQL-запрос. Такой вид атаки называется инъекцией, в данном случае самый распространённый — SQL-инъекция. Это опаснейшая уязвимость, позволяющая злоумышленнику получить доступ к базе данных и возможность читать/изменять/удалять информацию, которая для него не предназначена.   
Эта уязвимость является следствием недостаточной проверки данных, поступающих от пользователя. В целом эта разновидность атак имеет общее название «Ошибки валидации», к ней относятся далеко не только SQL-инъекции.

Подробно SQL-инъекции были рассмотрены в предыдущей курсовой работе, а также была проведена работа по устранению данной уязвимости.

Одна-единственная уязвимость в веб-приложении может привести к глобальной утечке данных, которая может разрушить до основания даже самую большую компанию, вызвав негативные отзывы в средствах массовой информации по всему миру, серьезные финансовые санкции и потерю общественного доверия. Для защиты веб-приложений сегодня используются общепринятые стандарты и методы, такие как OWASP. Также популярны и инструменты поиска уязвимостей, такие как IBM Rational AppScan.

1. Большие данные. Визуализация данных

4.1 Большие данные

Перед анализом данных чата было рассмотрено такое понятие как “Большие Данные” (Big Data).

Большие Данные – новое направление для нашей страны, но в западных странах оно получило широкое распространение. Связано это с тем, что в эпоху информационных технологий, особенно после роста популярности социальных сетей, по каждому пользователю интернета стало накапливаться значительное количество информации, что в конечном счете дало развитие направлению Big Data. К данной сфере относится обработка большого объема информации.

Сфера использования технологий Больших Данных обширна. Так, с помощью Больших Данных можно узнать о предпочтениях клиентов, об эффективности маркетинговых кампаний или провести анализ рисков. В веб-чате реализовано несколько модулей для анализа данных. Каждый пользователь может просмотреть информацию о своей анкете или общую статистику чата. Вся информация предоставляется в удобном для пользователя формате – графическом.

4.2 Визуализация данных

Визуализация данных сегодня занимает одно из важных мест в работе любой организации, так как с помощью графиков и диаграмм можно намного быстрее донести информацию до пользователя, а также натолкнуть его на нужные выводы, которые они можно сделать самостоятельно. С помощью JavaScript-библиотек можно визуализировать данные веб-приложений.

Перед тем, как выбрать подходящую библиотеку для визуализации данных чата, были рассмотрены и проанализированы наиболее популярные.

* + 1. Chartist.js

Chartist.js является простой библиотекой графиков на основе векторного формата SVG, которая предназначена для легко масштабируемых сайтов. Графики всегда будут в хорошем качестве независимо от того, просматривает ли пользователь их на компьютере или мобильном устройстве.

* + 1. Vis.js

Vis.js — динамическая библиотека визуализации в браузере. Библиотека предназначена для простой обработки больших объемов динамических данных и позволяют ими управлять.

* + 1. Epoch

Epoch позволяет создавать графики с хорошей визуализацией в реальном времени. Библиотека нацелена на создание исторических отчетов, а также на представление часто обновляемых данных в виде графиков.

* + 1. Ember Charts

Ember Charts — это библиотека диаграмм, созданная на основе Ember.js и d3.js, которая включает в себя разного вида графики, которые с легкостью можно модифицировать. Также библиотека станет хорошим помощником в подготовке презентаций и создании интерактивных графиков.

* + 1. CanvasJS

CanvasJS является простой в использовании библиотекой для построения графиков, которые будут правильно отображаться на всех устройствах, включая iPhone, iPad, Android, Windows Phone, настольные компьютеры и т.д. Это позволяет создавать многофункциональные информационные панели, которые работают на всех устройствах без ущерба для функциональности веб-приложения.

* + 1. D3.js

D3.js (или просто D3 – Data Driven Documents) – это библиотека JavaScript, использующая цифровые данные для создания и контроля динамических и интерактивных графических элементов, которые могут отображаться в веб-браузере. Кроме всего прочего, D3 как инструмент визуализации данных поддерживает технологии, предусмотренные стандартом W3C – SVG (Scalable Vector Graphics – масштабируемая векторная графика), JavaScript, HTML5 и CSS3.

D3 – не первый подход к веб-визуализации данных. Предшественники D3 не стали популярны лишь потому, что используемые классы были скрыты от программиста. В этом смысле D3 является инновационным подходом, поскольку в библиотеке используется стандартная DOM-модель (Document Object Model, объектная модель документа). Любой элемент (например, HTML-документ или веб-приложение) известной структуры, таким образом, может быть представлен в виде дерева узлов, каждый из которых представляет собой образец данных – текстовый, графический или любой другой объект. Узлы связаны между собой отношением «родитель-потомок», что позволяет программам и скриптам получать доступ к веб-документам, изменять их структуру и содержимое, а также оформление.

Для чата была выбрана библиотека D3, так как она предоставляет широкие возможности для визуализации данных (доступно более 1000 различных способов графического представления данных), открытый исходный код и полную документацию[9], на нескольких языках, в том числе и русском.

Каждый пользователь может просматривать данные своей страницы, а именно: топ-5 пользователей, которые чаще всего посещают анкету, возраст и пол гостей. Все эти данные визуализированы с помощью ресурсов библиотеки D3. Код, с помощью которого происходит отображение круговой диаграммы:

<script>

**var** height = 500,

width = 500,

margin=30,

data=[

{s\_pol: "Мужчины - ", age: 25},

{s\_pol: "Женщины - ", age: 64},

]

// данная функция устанавливает цвета, используемые в диаграмме

**var** color = d3.scale.category10();

// задается радиус круговой диаграммы

**var** radius = Math.**min**(width - 2\*margin, height- 2\*margin) / 2.5;

**var** arc = d3.svg.arc()

.outerRadius(radius)

.innerRadius(0);

**var** pie = d3.layout.pie()

.**sort**(null)

.value(**function**(d) { **return** d.age; });

**var** svg = d3.select("div.pie\_diagram").append("svg")

.attr("class", "axis")

.attr("width", width)

.attr("height", height)

.append("g")

.attr("transform",

"translate(" +(width / 2 )+ "," + (height / 2 ) + ")");

**var** g = svg.selectAll(".arc")

.data(pie(data))

.enter().append("g")

.attr("class", "arc");

g.append("path")

.attr("d", arc)

.style("fill", **function**(d) { **return** color(d.data.s\_pol); });

g.append("text")

.attr("transform", **function**(d) {

**return** "translate(" + arc.centroid(d) + ")"; })

.style("text-anchor", "middle")

.text(**function**(d) { **return** d.data.age + "ч."; });

**var** legendTable = d3.select("svg").append("g")

.attr("transform", "translate(0, "+margin+")")

.attr("class", "legendTable");

**var** legend = legendTable.selectAll(".legend")

.data(pie(data))

.enter().append("g")

.attr("class", "legend")

.attr("transform", **function**(d, i) {

**return** "translate(0, " + i \* 20 + ")";

});

legend.append("rect")

.attr("x", width - 10)

.attr("y", 4)

.attr("width", 10)

.attr("height", 10)

.style("fill", **function**(d) { **return** color(d.data.s\_pol); });

legend.append("text")

.attr("x", width - 14)

.attr("y", 9)

.attr("dy", ".35em")

.style("text-anchor", "end")

.text(**function**(d) { **return** d.data.s\_pol; });

</script>

Полученная диаграмма, показывающая количество женщин и мужчин, посетивших анкету, будет выглядеть следующим образом:

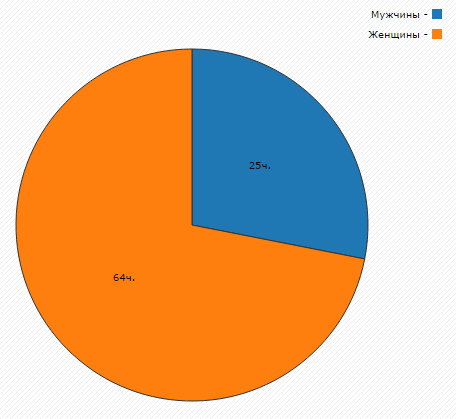


Рисунок 7. Пример круговой диаграммы нарисованной с помощью библиотеки D3.

Еще один пример графического представления данных – гистограмма, которая показывает возраст посетивших страницу пользователей:

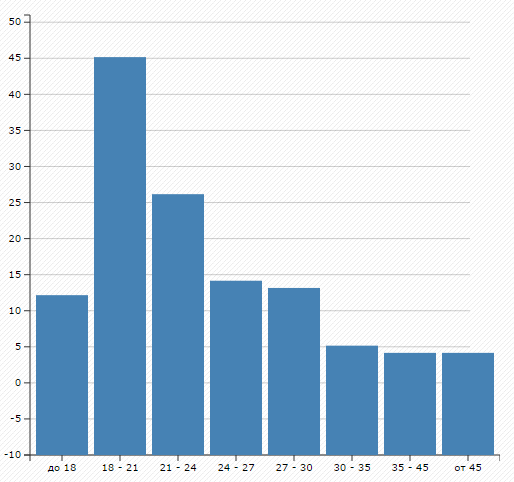


Рисунок 8. Пример гистограммы нарисованной с помощью библиотеки D3.

Общая статистика чата включает в себя следующие данные: самые популярные темы, количество регистраций каждый день. Регистрация доступна в трех вариантах: через интернет-портал Яндекс, социальную сеть Вконтакте и через форму регистрации на сайте. Таким образом, можно увидеть сколько человек предпочитают регистрацию через сторонние сервисы и какие именно.

Сбор статистики вынесен в отдельный модуль. Таким образом, можно легко и быстро добавлять новые графики и анализировать новые данные.

1. Разработка чата

5.1 База данных

Вся информация чата хранится в базе данных MySQL. Общая структура базы данных, для хранения информации о пользователях и общих данных о чате, выглядит следующим образом:

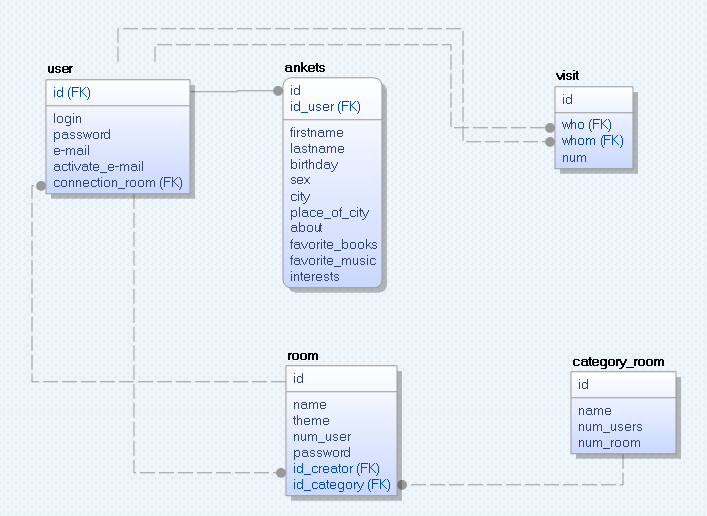


Рисунок 10. Общая структура базы данных для хранения информации о пользователях и общей информации чата

5.2 Анкета пользователя

**Анкета пользователя** – это страница с его личной информацией. В анкете отражаются данные пользователя - информация о себе, которую он указывает самостоятельно - фото, рассказ о себе и прочее. Анкета в чате выглядит следующим образом:

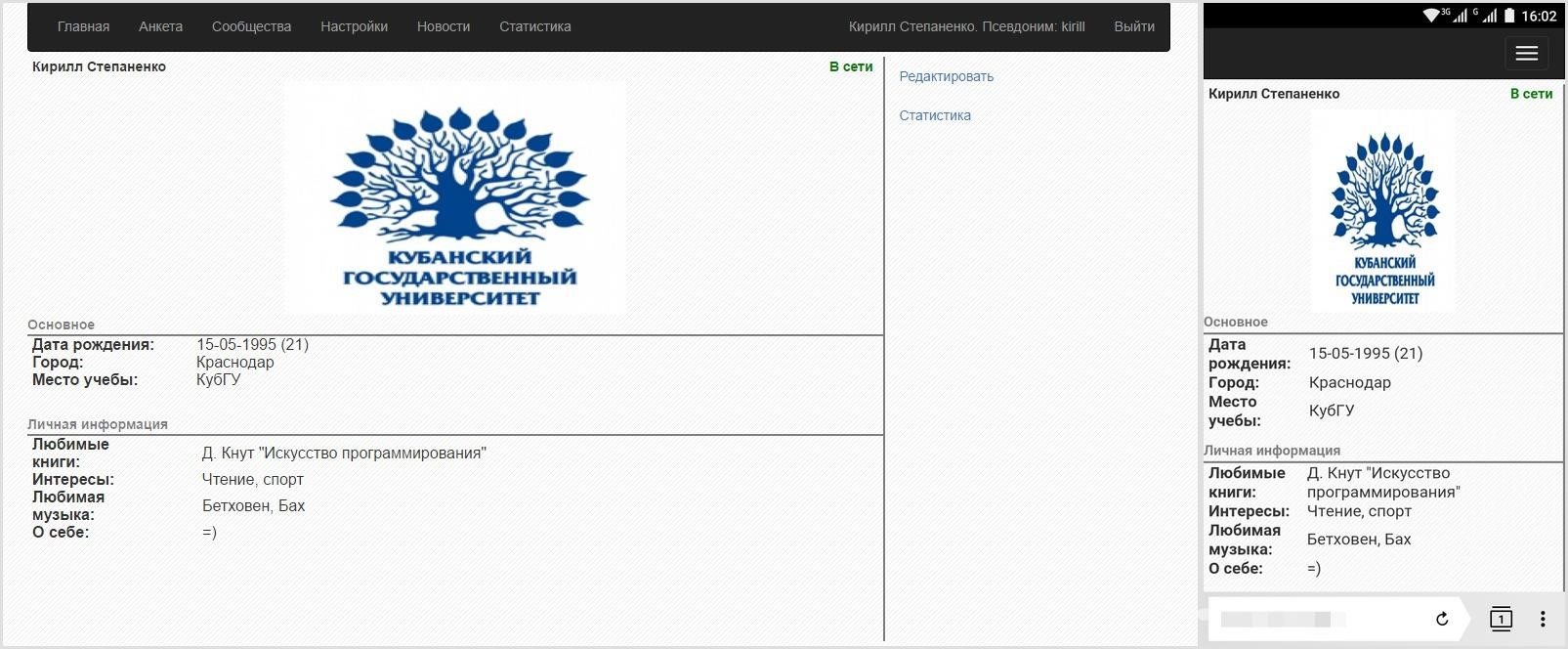


Рисунок 11. Вид анкеты пользователя на компьютере и мобильном устройстве

Данные анкеты хранятся в отдельной таблице базы данных. Для того, чтобы знать какому именно пользователю принадлежит анкета, в базе данных применена идентифицирующая связь – это такая связь между таблицами, когда строка (или строки) дочерней таблицы строго определяются родительскими и не могут существовать без них. Т.е. если не будет пользователя, например, с id = 7 то и анкеты с id = 7 тоже не будет. Для организации такой связи в одну таблицу (в данном случае это таблица ankets) помещается поле, содержащее адрес записи другой таблицы (user). Получается следующая схема:

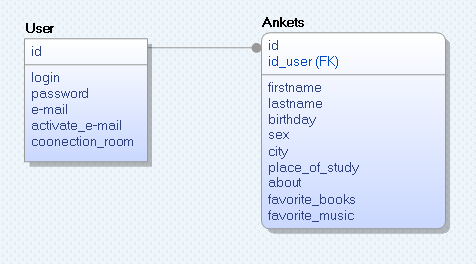


Рисунок 12. Структура таблиц базы данных для анкеты пользователя

5.3 Настройки пользователя

Каждому пользователю доступна страница с настройками личной информации, а также некоторыми настройками чата. Настройки каждого пользователя хранятся в отдельной таблице (ankets) базы данных. Схема базы данных показана на предыдущем рисунке.

5.4 Главная страница

Главная страница представляет из себя два блока для экранов с большим разрешением и один блок для экранов с маленьким разрешением.

В главном блоке выводится информация для пользователя: приветствие, новости и прочее. Именно этот блок остается доступным по умолчанию в версии для экранов с маленьким разрешением.

Блок, в котором отображаются категории тем, каждый день имеет разный вид. Связано это с тем, что каждый в день в 23:59 на сервере в автоматическом режиме происходит анализ данных, на основе которого строится список тем, которые были популярны за прошедшие сутки. Таким образом каждые сутки строится список, который ранжирует категории тем по популярности в данное время.

Автоматический анализ данных удалось организовать с помощью команды cron, доступной на сервере. Cron — классический демон-планировщик задач в UNIX-подобных операционных системах, использующийся для периодического выполнения заданий в определённое время. Регулярные действия описываются инструкциями, помещенными в файлы crontab и в специальные директории. **Cron** ежеминутно просматривает свои таблицы, в которых содержатся информация о периодичности запуска команд и запускает команды, когда значения полей “минута”, “час”, “месяц” и, хотя бы одно из полей “число” и “день недели”, совпадают с текущим временем. Cron играет незаменимую роль в автоматизации нужных процессов.

5.5 Окно чата

Окно чата так же доступно в двух вариантах: для экранов с большим разрешением и для экранов с маленьким разрешением.

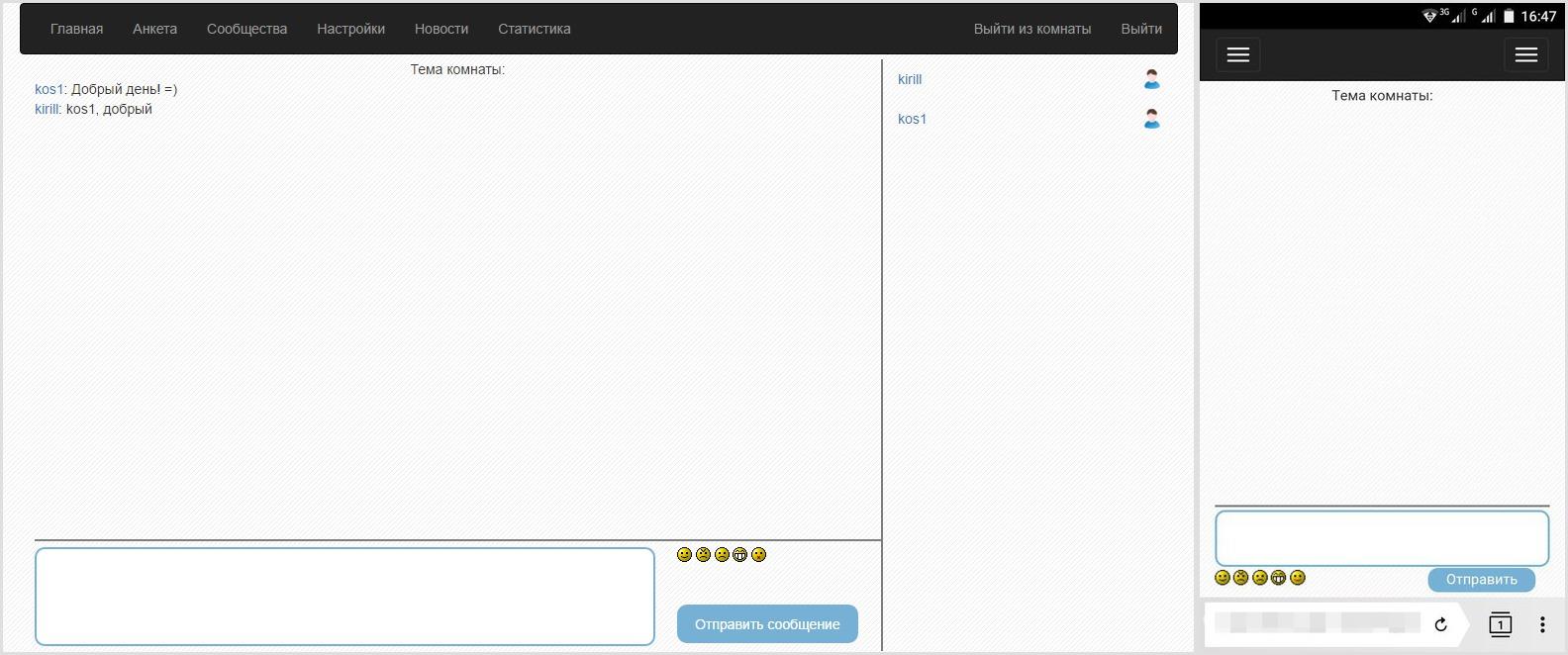


Рисунок 13. Вид окна чата на компьютере и мобильном устройстве

Все сообщения чата хранятся в отельной базе данных, а для каждой комнаты используется своя таблица. Данный способ хранения не единственный. Можно было использовать два подхода для хранения сообщений: тот, что описан выше и хранить сообщения в одной таблице для каждой темы, но с указанием id комнаты для каждого сообщения. Если выбрать второй способ, то время ответа сервера значительно увеличится при большом количестве сообщений, так как их нужно не просто получить списком (как это делается в первом способе), а еще и отсортировать, т.е. выбрать именно те сообщения, id которых соответствует id нужной комнаты.

Таблица для хранения сообщений одной комнаты выглядит следующим образом:



Рисунок 14. Структура таблицы для хранения сообщений одной комнаты.

5.6 Адаптивный дизайн

Для разработки адаптивного дизайна был выбран CSS-фреймворк Bootstrap3. В фреймворк интегрирована поддержка мобильных устройств с шириной окна браузера менее 992 пикселей. Это позволяет шаблону сайта адаптироваться под размер экрана разных устройств, в том числе планшетов компьютеров и мобильных телефонов. Для того чтобы сделать сайт адаптивным применяется адаптивная сетка - коллекция стилей, основанных на классах, которые позволяют разработчику контролировать макет страницы, используя систему строк и колонок. В ходе разработки шаблона, были проанализированы преимущества адаптивной сетки:

1. Блоки с содержимым, которые размещаются в одной строке, при уменьшении размера экрана браузера переходят друг под друга или складываются друг на друга.

2. Ширина блоков изменяется пропорционально изменению ширины окна браузера, за счет чего сайт будет хорошо отображаться на мобильных устройствах и планшетных компьютерах.

Например, на следующих двух скриншотах показано как будет выглядеть шаблон сайта на устройстве с шириной экрана от 768 до 992 пикселей и на устройстве с шириной экрана менее 768 пикселей.

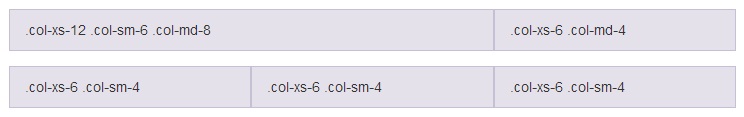


Рисунок 15. Вид шаблона сайта на устройстве с шириной экрана 768-992 пикселей

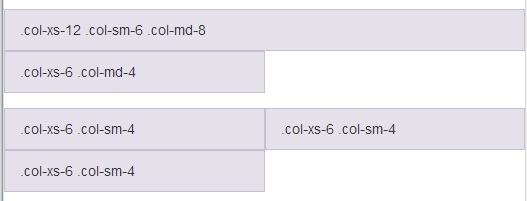


Рисунок 16. Вид шаблона сайта на устройстве с шириной экрана менее 768 пикселей

В пункте документации по Bootstrap3[5] определены основные принципы использования сетки для верстки шаблона сайта:

1. Весь контент сайта помещается в блоки <div></div>. Для блоков задается определенная ширина и к нему применяется определенный класс.

2. Все блоки сайта помещаются в строки, которые являются их родителями.

3. Строки должны помещаться в общий контейнер с фиксированной или плавающей шириной для правильного размещения контента относительно друг друга:

<div class="container-fluid">

<div class="row">

...

</div>

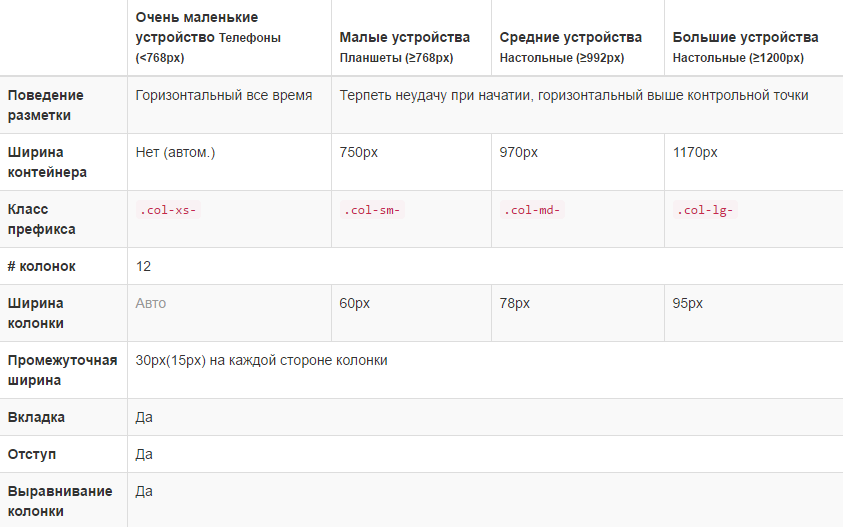
</div>

4. Любой блок можно рассматривать как контейнер и создавать в нем вложенные строки с блоками.

5. Сумма чисел в классах, примененных к блокам одной строки, **не должна превышать 12.**

На следующем рисунке показа функционал разметки Bootsrap3:

Рисунок 17. Функционал разметки Bootstrap3



По таблице видно, что ширина блока зависит от размера окна браузера и составляет либо от 0 до 60 пикселей, либо 60 пикселей, либо 78 пикселей, либо 95 пикселей. Каждому значению ширины блока соответствует критический размер окна браузера, при уменьшении которого ширина блока пропорционально уменьшается.

В таблице приведены префиксы классов для каждого диапазона ширины блока. Классы можно применять к блокам одной строки. Для задания ширины блока указывается число от 1 до 12 в префиксе, что позволяет разместить один или несколько блоков в строке.

В следующем примере показана разметка главной страницы чата:

<div class="container">

<div class="row no-gutter">

<div class="col-md-9">

<!—Центральная область, в которой выводится информация -->

</div>

<div class="navmenu-brand visible-md visible-lg" id="responsive-menu">

<div class="col-md-3">

<!—Область, в которую выводятся категории комнат -->

</div>

</div>

</div>

</div>

Стоит отметить, что код разметки приведен в общем виде, так как оригинальный код достаточно большой. Все другие страницы чата имеют примерно такую же разметку.

5.7 Интеграция чата с ВКонтакте на базе Open API

Социальная сеть ВКонтакте предоставляет возможности для интеграции со сторонними сайтами. В основном эти возможности представлены уже готовыми виджетами. Однако, есть еще и Open API, которое позволяет не только авторизовать пользователя, но и выполнить любой метод. Работа с Open API проста и понятна, во многом это благодаря подробной документации[6], которая доступна на сайте ВКонтакте.

Для того, чтобы сайт или веб-приложение получило доступ к данным, достаточно зарегистрировать его в кабинете разработчика на сайте социальной сети ВКонтакте. При регистрации пользователю выдается специальный ID, с помощью которого можно инициализировать API (делается это с помощью метода VK.init):

VK.init({

apiId: 1234567

});

После инициализации API, можно переходить к авторизации, для этого предусмотрен метод VK.Auth.login:

VK.Auth.login(null, VK.access.FRIENDS);

В параметрах указываются те данные, к которым нужно получить доступ. В примере, указанном выше, доступ запрашивается к списку друзей.

Для вызова любого метода служит метод VK.Api.call, который принимает три параметра — название метода, объект, представляющий собой параметры, передаваемые в указанный метод, и callback-функцию, в которую будет передан результат вызова метода. Например, для получения списка друзей необходимо вызвать метод friends.get. Метод не имеет обязательных параметров и в случае вызова без параметров вернет список всех друзей. Допустим, нам нужны фотографии друзей пользователя, передадим в метод параметр fields со значением 'photo\_rec', что соответствует квадратной фотографии со стороной 50 пикселей:

VK.Api.call('friends.get', {

fields: 'photo\_rec'

}, function(data){

// обработка списка друзей

});

Благодаря Open API любой пользователь может пройти быструю регистрацию или авторизацию в чате, что делает чат более удобным для использования. В статистике чата можно посмотреть сколько человек зарегистрировалось с помощью ВКонтакте, а сколько через стандартную регистрацию чата. В дальнейшем, с помощью Open API, планируется реализовать автоматическое заполнение анкеты чата данными социальной сети ВКонтакте.

ЗАКЛЮЧЕНИЕ

В ходе продолжения разработки веб-чата были получены навыки создания адаптивного дизайна сайта или веб-приложения. Перспектива адаптивного дизайна может быть продемонстрирована статистикой за два года. В 2009 году использовали сеть Интернет с компьютера 1,4 млрд. человек, а в 2014-м – 1,6 млрд. В 2009 году использовали Интернет с мобильных устройств лишь 800 млн. человек, а в 2014 эта цифра приближается к 2 млрд. Также в ходе разработки веб-приложения были рассмотрены основные понятия Больших Данных и инструменты, которые применяются для их визуализации. Подробно была рассмотрена и применена на практике библиотека D3.js, которая является одной из лучших для интерактивной визуализации данных. Были получены знания по безопасности веб-приложений, некоторые из которых были применены при разработке чата. В ходе разработки были улучшены навыки владения языком PHP, JavaScript, а также администрирования баз данных с помощью различных СУБД.

Полученное веб-приложение в дальнейшем будет дорабатываться и улучшаться, будут добавляться новые возможности.

ЛИТЕРАТУРНЫЕ ИСТОЧНИКИ

1. Скембрей Д., Шема М. Секреты хакеров. Безопасность Web-приложений - готовые решения// Издательский дом "Вильямс", 2003 -356 с.

2. Маркотт И. Отзывчивый веб-дизайн// Издательство: Манн, Иванов и Фербер, 2012 – 277с.

3. Сравнение адаптивных CSS фреймворков. [Электронный ресурс] Статья, URL: https://habrahabr.ru/post/156425/ [Дата обращения: 20 апреля 2016].

4. Введение в D3. [Электронный ресурс] Статья, URL: https://habrahabr.ru/company/datalaboratory/blog/217905/ [Дата обращения: 18 апреля 2016].

5. Документация на русском языке. [Электронный ресурс] Документация, URL: http://bootstrap-3.ru/index.php/ [Дата обращения: 18 февраля 2016]

6. Документация VK API. [Электронный ресурс] Документация, URL: https://new.vk.com/dev/main/ [Дата обращения: 17 мая 2016]

7. Классификация угроз безопасности Web-приложений. [Электронный ресурс] Статья, URL: http://www.infosecurity.ru/iprotect/websec/classification/ [Дата обращения: 29 апреля 2016]

8. 9 основных принципов отзывчивого веб-дизайна. [Электронный ресурс] Статья, URL: https://habrahabr.ru/post/243247/ [Дата обращения: 25 февраля 2016]

9. Документация по D3.js. [Электронный ресурс] Документация, URL: https://github.com/d3/d3/wiki/ [Дата обращения: 18 апреля 2016]