

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ
Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математической теории упругости и биомеханики

**Проектирование и частичная реализация веб-приложения «Спортивная
жизнь университета»**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 442 группы

направления 09.03.03 «Прикладная информатика (прикладной бакалавриат)»

механико-математического факультета

Юналиева Шамиля Хайдаровича

Научный руководитель
доцент, к.ф.-м.н.

подпись, дата

Л.В. Бессонов

Зав. кафедрой
д.ф.-м.н., профессор

подпись, дата

Л.Ю. Коссович

Введение

Бакалаврская работа посвящена исследованиям в области разработки веб-приложений.

В современном мире интернет выступает в роли мощного инструмента по поиску и предоставлению информации. По статистике, более половины жителей планеты имеют доступ к сети Интернет. А значит, способы электронного улучшения жизни должны развиваться и совершенствоваться.

В каждом университете есть свои спортивные команды. Между этими командами проводятся турниры и различного вида соревнования. В целях удобства получения информации разрабатываются спортивные веб-приложения. Приложение предоставляет пользователям информацию о времени и месте проведения турниров, справочную информацию об игроках различных команд. Так же приложение реализует функционал поиска для ускорения процесса обработки информации. Обычно пользователями таких приложений являются сами игроки, студенты и преподаватели университета. За поддержание информации в актуальном состоянии должны отвечать администраторы.

Целью бакалаврской работы является проектирование и частичная реализация веб-приложения «Спортивная жизнь университета». Задачами выполняемой работы являются:

- 1) Анализ предметной области;
- 2) Изучение теоретических аспектов связанных с разработкой веб-приложений;
- 3) Анализ существующего программного обеспечения для разработки веб-приложений;
- 4) Разработка структуры и пользовательского интерфейса веб-приложения;
- 5) Частичная реализация веб-приложения с помощью выбранных средств.

Структура и объем работы. Бакалаврская часть состоит из введения, 3 глав, заключения и списка используемых источников, включающего 25 наименований. Работа изложена на 46 листах машинописного текста без приложений, содержит 18 рисунков.

Основное содержание работы. В введении описывается актуальность поставленной задачи, объект и предмет исследования, а также сформулирована цель работы и задачи.

Первая глава состоит из двух разделов. В ней рассмотрен обзор структуры стандартного веб-приложения, а также рассмотрены основные понятия и возможности CMS Drupal.

Веб-приложение — клиент-серверное приложение, в котором клиент взаимодействует с сервером при помощи браузера, а за сервер отвечает веб-сервер. Логика веб-приложения распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются кроссплатформенными сервисами [1].

Любое веб-приложение работает на клиент-серверной архитектуре. Клиентская часть работает на компьютере непосредственно пользователя и откликается на все пользовательские действия, серверная часть ожидает от клиента запросов, по поступлению которых начинает формировать нужный ответ и отсылать его обратно клиенту. Обычно клиент нужен для обеспечения красивого интерфейса, который был бы удобен для конечного пользователя, а сервер для проведения основных вычислений и хранения информации. Чаще всего на множество клиентских приложений требуется всего несколько серверов (рисунок 1.1), что существенно облегчает обслуживание, так как большая часть ошибок будет, скорее всего, лишь на нескольких машинах, напрямую принадлежащих производителю [2].

У описанной архитектуры есть как преимущества, так и недостатки, которые описаны ниже.

Преимущества:

- 1) Отсутствие дублирования кода программы-сервера программами-клиентами;
- 2) Сделать приложения кроссплатформенным при такой архитектуре гораздо легче, ведь нужно оптимизировать лишь клиентский код;
- 3) Так как все вычисления выполняются на сервере, то требования к компьютерам, на которых установлен клиент, снижаются;
- 4) Все данные хранятся на сервере, который, как правило, защищен гораздо лучше большинства клиентов. На сервере проще обеспечить контроль полномочий, чтобы разрешать доступ к данным только клиентам с соответствующими правами доступа [3].

Недостатки:

- 1) Неработоспособность сервера может сделать неработоспособной всю вычислительную сеть;
- 2) Поддержка работы данной системы часто требует отдельного специалиста — системного администратора;
- 3) Высокая стоимость оборудования [4].

Чаще всего веб-приложения состоят как минимум из трёх основных компонентов:

1) Клиентская часть веб приложения — это графический интерфейс. Это то, что вы видите на странице. Графический интерфейс отображается в браузере. Пользователь взаимодействует с веб-приложением именно через браузер, кликая по ссылкам и кнопкам.

2) Серверная часть веб-приложения — это программа или скрипт на сервере, обрабатывающая запросы пользователя (точнее, запросы браузера). Чаще всего серверная часть веб-приложения программируется на PHP. При каждом переходе пользователя по ссылке браузер отправляет запрос к серверу. Сервер обрабатывает этот запрос, инициируя некоторый PHP-скрипт, который

формирует веб-страничку, описанную языком HTML, и отправляет клиенту по сети. Браузер тут же отображает полученный результат в виде очередной веб-страницы [5].

3) База данных (БД, или система управления базами данных, СУБД) — программное обеспечение на сервере, занимающееся хранением данных и их выдачей в нужный момент. В случае форума или блога, хранимые в БД данные - это посты, комментарии, новости, и так далее. База данных располагается на сервере. Серверная часть веб-приложения (то есть, PHP скрипт) обращается к базе данных, извлекая данные, которые необходимы для формирования страницы, запрошенной пользователем. Браузер через Интернет отправляет HTTP-запросы веб-серверу. Веб-сервер вызывает PHP-скрипт, написанный разработчиком веб-приложения. PHP-скрипт обращается к базе данных, если это нужно. В результате PHP-скрипт возвращает клиенту веб-страницу, которую и отображает браузер [6].

Drupal — это популярная система управления контентом с открытым исходным кодом. Из-за ее мощного функционала разрабатывать сложные сайты на Drupal значительно проще, чем писать их с нуля. Неудивительно, что благодаря многочисленному сообществу пользователей и огромному количеству модулей, мы слышим о Drupal все чаще и чаще [7].

Достоинства:

1) Открытый код. Drupal распространяется с открытым исходным кодом, а это значит — вам доступны все плюсы открытого ПО;

2) Гибкость конфигурирования. Главное достоинство Drupal кроется в его гибкой архитектуре. Вы можете использовать его для построения сайтов любых типов: от социальных медиа-сайтов, позволяющих пользователям размещать собственное содержимое и голосовать за него, форумов и сайтов с вакансиями до галерей или публикаций портфолио дизайнеров;

3) Многочисленное и грамотное сообщество разработчиков. Вокруг Drupal сложилось многочисленное, яркое и открытое сообщество пользователей и разработчиков. Это значит, вам доступны отличные модули,

своевременное исправление багов и обновление ядра, а также неиссякаемый поток документации и руководств в сети;

4) Модули. Для Drupal разработано огромное число модулей, расширяющих его исходный функционал (о том, что такое модуль, мы поговорим немного позже);

5) Легкость для разработчиков. Drupal изначально был ориентирован на потребности разработчиков. Основным недостатком некоторых популярных CMS является излишнее выделение пользовательского интерфейса, что часто влечет недостаток внимания к нуждам тех, кто, собственно, будет разрабатывать и продвигать систему в дальнейшем;

6) Встроенная система кеширования. Drupal обладает встроенной системой кеширования, способной снизить нагрузку на сервер и сократить время генерации страницы. Кеширование позволяет исключить сложных запросов к базе данных, что повышает производительность сервера;

7) Встроенная поисковая система. В отличие от других CMS, в Drupal есть система поиска, реализованная на уровне ядра. Конечно, она не может конкурировать с такими поисковыми сервисами как Google Search, или Yahoo! Search BOSS, но тем не менее вполне пригодна для работы [8].

Недостатки.

1) Длительный процесс обучения. Drupal поддается изучению несколько труднее, чем другие CMS. Уходит много времени, чтобы в полной мере понять Drupal и начать разрабатывать собственные проекты;

2) Сложность для новичков. Тем, кто не очень знаком с информационными технологиями, потребуется некоторое время, чтобы привыкнуть к Drupal. Это значит, что специалистам, разбирающимся в веб-разработке лишь поверхностно, будет сложно создавать (или даже администрировать) сайты на Drupal;

3) Сложный интерфейс. Интерфейс администрирования сайта в Drupal несколько сложен для понимания; в целом он не такой дружелюбный к пользователю, каким мог бы быть [9].

Drupal используется для создания веб-сайтов. Это модульная платформа с открытым исходным кодом, предназначенная для совместного управления содержимым (контентом). Она легко поддается расширению, соответствует большинству стандартов, ее код прост, понятен и невелик по объему. Система Drupal устанавливается в варианте, обеспечивающем выполнение лишь ее основных функций, а дополнительные возможности реализуются встроенными и сторонними модулями. Система Drupal допускает настройку, но только с помощью полной перезаписи ядра или добавления модулей, а не за счет изменения кода ядра. Кроме того, она эффективно отделяет управление контентом от его представления пользователям. С помощью Drupal можно создать Интернет-портал, персональный, ведомственный или корпоративный веб-сайт, сайт Интернет-магазина, каталог сетевых ресурсов, Интернет-газету, сайт социальной сети, галерею изображений, интрасеть и практически любую другую разновидность веб-сайта. Специальная группа, отвечающая за безопасность, стремится обеспечить защиту Drupal от новых угроз, выпуская обновления для системы безопасности. Некоммерческая организация под названием “Ассоциация Drupal” поддерживает Drupal, улучшая инфраструктуру веб-сайта drupal.org и организуя конференции и мероприятия, посвященные Drupal. А непрерывно пополняющееся сетевое сообщество пользователей, администраторов сайтов, проектировщиков и веб-разработчиков упорно трудится над улучшением программного обеспечения Drupal [10].

Вторая глава состоит из двух разделов в которых были спроектированы пользовательские сценарии и типы материалов.

Пользовательский сценарий детально описывает, что пользователь делает на сайте, и, особенно, почему он это делает. Пользовательский сценарий – это короткая история, рассказывающая о пользователе, который посещает ваш сайт с определенной мотивацией и целью. Хороший пользовательский сценарий включает в себя всю информацию о шагах, предпринимаемых пользователем для достижения своей цели.

Пользовательские сценарии – это очень удобный инструмент для разработки пользовательских интерфейсов. Ведь важно не только «познакомиться» с пользователем, но и понять его цели. Только тогда, когда мы поймем, кто и что делает на нашем сайте, а также то, как и зачем они это делают, мы сможем четко определиться с требованиями дизайна, которые нам предстоит удовлетворить. Так что нам нужно сжать контент веб-сайта до определенных целей, а затем представить всё это в виде сценариев.

определить не только набор функций, но и их приоритет для пользователя.

Сценарии позволяют перейти от выработки стратегии сначала к набору возможностей, а затем и к проектированию интерфейса. Они помогают лучше понять проблему, прежде чем непосредственно перейти к ее решению. Без этого для оценки качества решений придется руководствоваться интуицией и «вкусовщиной», а это, как правило, ничем хорошим не заканчивается.

Построим основные варианты использования системы, как со стороны непосредственного пользователя, так и администратора. Основным критерием качества таких сценариев выберем принцип «минимального количества кликов», подразумевающий минимизацию количества действий, требуемых для получения результата, а также принцип согласованности реакции системы с ожиданиями пользователя.

Один из самых принципиальных моментов проектирования системы – это проектирование структуры данных, позволяющей в нормализованном виде хранить информацию.

Для реализации этого процесса созданы специальные механизмы, разработаны типы материалов. Тип материала платформы CMS Drupal 7 является косвенным аналогом сущности. Таким образом, при реализации инфологической модели будем считать, что сущности – это конкретный тип материала.

Третья глава состоит из двух разделов. Она посвящена разработке интерфейсов веб-приложения и реализации типов представления.

Пользовательский интерфейс – это совокупность информационной модели проблемной области, средств и способов взаимодействия пользователя с информационной моделью, а также компонентов, обеспечивающих формирование информационной модели в процессе работы программной системы. Под информационной моделью понимается условное представление проблемной области, формируемое с помощью компьютерных (визуальных и звуковых) объектов, отражающих состав и взаимодействие реальных компонентов проблемной области.

Группа модулей Views – данная группа модулей предназначена для настройки вывода типа материала на страницы сайта, расположения полей, добавления полей из других типов материалов, добавления фильтров, связей, сортировок и прочего. С помощью этих модулей в веб-приложении были созданы представления в виде таблиц с фильтрами, которые использовались для вывода всех игроков и для вывода результатов матчей.

Основное достоинство хорошего интерфейса пользователя заключается в том, что пользователь всегда чувствует, что он управляет программным обеспечением, а не программное обеспечение управляет им.

Для реализации веб-приложения использовался стандартный набор модулей CMS Drupal, использовались модули уже написанные сообществом разработчиков CMS Drupal.

На базе стандартного модуля Views, было построено представление, которое выводит на страницу типы материалов, такие как «Игрок» и «Результат матча».

Заключение. Цель данной работы заключалась в проектировании и частичной реализации веб-приложения «Спортивная жизнь университета». Для выбранной цели был поставлен ряд задач.

В результате анализа предметной области были рассмотрены структуры стандартных веб-приложений, спроектированы сценарии использования сервиса как со стороны рядового пользователя, так и со стороны,

администратора системы, спроектированы типы материалов и инфологическая модель для хранения данных сервиса.

По артефактам проектирования была произведена частичная реализация, в ходе которой был создан образец веб-приложения с ограниченным функционалом. Таким образом, поставленные задачи решены в полном объеме, а цель выпускной квалификационной работы полностью достигнута.

Список использованных источников

- 1 Front end и back end [Электронный ресурс] : [сайт]. - URL: [https://ru.wikipedia.org/wiki/ Front_end_и_back_end](https://ru.wikipedia.org/wiki/Front_end_и_back_end) (дата обращения 25.04.2019). – Загл. с экрана. – Яз. рус.
- 2 Особенности тестирования веб-приложений [Электронный ресурс] : [сайт]. - URL: <http://quality-lab.ru/key-principles-of-web-testing/> (дата обращения 03.05.2019). – Загл. с экрана. – Яз. рус.
- 3 Структура веб-приложения [Электронный ресурс] : [сайт]. - URL: <http://labaka.ru/likbez/struktura-veb-prilozheniya> (дата обращения 03.05.2019). – Загл. с экрана. – Яз. рус.
- 4 Обзор средств создания интерфейсов веб-приложений на языке Java [Электронный ресурс] : [сайт]. - URL: <http://masters.donntu.org/2013/fknt/riabinin/library/article.htm> (дата обращения 05.05.2019). – Загл. с экрана. – Яз. рус.
- 5 Томсон, Л. Разработка Web-приложений на PHP и MySQL / Л. Томсон, Л. Веллинг. - СПб. : ДиаСофтЮП, 2012. - 672 с.
- 6 Грекул, В.И. Проектирование информационных систем / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. - М. : Интернет-университет информационных технологий, 2010. - 304 с.
- 7 Колисниченко, Д.Н. Drupal 7. Руководство пользователя / Д.Н. Колисниченко. - М. : Диалектика, 2011. - 256 с.
- 8 Колисниченко, Д.Н. Выбираем лучший бесплатный движок для сайта. CMS Joomla! и Drupal / Д.Н. Колисниченко. - М. : БХВ-Петербург, 2010. - 288 с.

- 9 Мерсер, Д. Drupal. Создание надежных и полнофункциональных веб-сайтов, блогов, форумов, порталов и сайтов-сообществ / Д. Мерсер. - М. : Высшая школа, 2013. - 272 с.
- 10 Томлинсон, Т. CMS Drupal 7: руководство по разработке системы управления веб-сайтом, 3-е издание. / Т. Томлинсон. - М. : «Вильямс», 2011. - 560 с.