

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования

**«Южно-Уральский государственный университет»
(национальный исследовательский университет)**

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент

генеральный директор

ООО «Вортекскод»

_____ П.А. Михайлов

“ ____ ” _____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой,

д.ф.-м.н., профессор

_____ Л.Б. Соколинский

“ ____ ” _____ 2017 г.

**РАЗРАБОТКА ВЕБ- ПРИЛОЖЕНИЯ
ДЛЯ АДМИНИСТРИРОВАНИЯ СЕРВИСА
"SUBSCRIPTION SERVICE"**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 02.03.02.2017.308-013-2046.ВКР

Научный руководитель

Старший преподаватель кафедры СП

_____ К.Ю. Никольская

Автор работы,

студент группы КЭ-402

_____ Тавфеек С.С.

Ученый секретарь

(нормоконтролер)

_____ О.Н. Иванова

“ ____ ” _____ 2017 г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	7
2. ТРЕБОВАНИЯ К СИСТЕМЕ	9
2.1. Функциональные требования	9
2.2. Нефункциональные требования	10
2.3. Варианты использования системы	10
3. ПРОЕКТИРОВАНИЕ	13
3.1. Постановка задачи	13
3.2. Макет главной страницы веб-сайта	14
3.3. Web-дизайн страниц	16
4. РЕАЛИЗАЦИЯ	17
4.1. Выбор средств реализации	17
4.2. Интерфейс главного страницы	17
4.3. Интерфейс авторизации	19
5. ТЕСТИРОВАНИЕ	21
ЗАКЛЮЧЕНИЕ	22
ЛИТЕРАТУРА	22
ПРИЛОЖЕНИЕ	25
Приложение 1	25
Приложение 2	29
Приложение 3	30
Приложение 4	30

ВВЕДЕНИЕ

Актуальность темы

На сегодняшний день в сети Интернет имеется большое количество ресурсов с постоянно обновляющимся контентом. Пользователи приходят на ресурс из поисковых систем, социальных сетей и сторонних ресурсов, в результате чего возникает необходимость установления связи со своей аудиторией, привлечения новых посетителей, а самое главное — поддержания их интересов. На данный момент возникает проблема, связанная с автоматическим уведомлением посетителей или постоянных читателей о выходе новой статьи, товара или версии приложения.

Сервис подписок — это возможность оперативно рассылать самый интересный и актуальный контент, начиная от новостей и прогноза погоды заканчивая появления нового товара в каталоге, используя SMS, MMS или email. Сервис подписок и рассылок предназначен для создания и отправки регулярных рассылок. С помощью функционала сервиса можно организовывать рассылку уведомлений для пользователей о различных событиях, произошедших на сайте: обновлениях разделов, добавленных комментариях.

В связи с этим, возникает потребность обрабатывать большие объемы информации для создания заявки на подписку, отслеживания ее статуса и актуальности, собирать информацию о пользователях заинтересованных в рассылке и автоматически отправлять им уведомления. Ежедневно в сети публикуется огромное количество предложений, что делает «ручной» сбор информации и ее редактирование неприменимым. По этой причине было решено создать сервис «Subscription Service» для работы с подписками.

Таким образом, разработка сервиса платных подписок «Subscription Service» является актуальной задачей на сегодняшний день. Система будет полезна как посетителям ресурса, помогая отслеживать и редактировать

всю информацию о своих подписках, так и владельцам ресурсов осуществляя оперативный сбор информации от пользователей и проводить автоматизированное уведомление.

Цель и задачи работы

Целью данной работы является разработка дизайна веб-приложения для администрирования сервиса платных подписок «Subscription Service».

Для достижения поставленной цели необходимо решить следующие задачи:

- провести анализ предметной области;
- проанализировать требования к веб-приложению;
- разработать структуру и дизайн веб-приложения;
- выполнить реализацию разработанных макетов;
- провести тестирование пользовательского интерфейса.

Структура и объем работы

Работа состоит из введения, пяти глав, заключения, библиографии и приложения. Объем работы составляет 30 страницы, объем библиографии – 16 источников.

Содержание работы

Первая глава, «Анализ предметной области», описывает предметную область, в рамках которой выполняется данная работа.

Вторая глава, «Требования к системе», описывает функциональные и нефункциональные требования к системе.

Третья глава, «Проектирование», описывает этап проектирования дизайна и пользовательского интерфейса веб-приложения.

Четвертая глава, «Реализация», описывает реализацию веб-приложения с технической точки зрения.

Пятая глава, «Тестирование», описывает тестирование веб-приложения.

В заключении описываются основные результаты, полученные при выполнении дипломной работы.

В приложении 1 «Скриншоты веб-страниц» представлены скриншоты веб-страниц разработанного веб-приложения.

В приложении 2 «Отчет о юзабилити-тестировании» представлен отчет об юзабилити-тестировании веб-приложения.

В приложении 3 представлены результаты функционального тестирования веб-приложения.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

1.1. Аналитический обзор существующих систем

Для обзора аналогичных систем был проведен анализ некоторых существующих систем. Далее приводится их описание.

FeedBurner – веб-сервис, пропускающий через себя RSS-потoki, исправляет в них мелкие ошибки и добавляющий потокам дополнительную функциональность, например, кнопку Play для подкастов. Кроме того, предоставляет статистику подписки на RSS-поток [1].

Основные достоинства FeedBurner:

- 1) высокая репутация. FeedBurner реализован так, что RSS попадает только то, что публикуется на страницах блога. Подписка FeedBurner имеет два уровня защиты от спама;
- 2) настраиваемая рассылка. Как и в любой другой рассылке в FeedBurner можно создавать и редактировать шаблоны писем.

Основные недостатки FeedBurner:

- 1) двойная система подтверждения электронной почты. При подписка на FeedBurner читателям приходится проходить двойную авторизацию: сначала через форму на сайте, в дальнейшем в собственной почте. Такая система очень часто приводит к тому, что письмо с подтверждением теряются в большом количестве сообщений, а авторы редко разбирают то, что уже было просмотрено;
- 2) сложная настройка. Неправильные настройки FeedBurner могут привести к формированию страниц дубликатов на вашем сайте.

Subscribe2 – плагин, позволяющий организовать подписку по e-mail на WordPress блоге [2]. Основные достоинства плагина в том, что он предоставляет следующие возможности:

- 1) редактируемые шаблоны писем;
- 2) автоматическая рассылка уведомлений о новых постах в блоге на электронный ящик;
- 3) рассылки вручную писем из административной части;

4) возможность подписываться и отписываться от уведомлений как зарегистрированным пользователям, так и незарегистрированным;

5) зарегистрированные пользователи могут самостоятельно управлять своей подпиской: подписываться и отписываться от определенных рубрик;

6) для подписки требуется активация по e-mail;

7) рассылка писем из админ-части: всем или определенной группе пользователей.

Модуль «Подписка и рассылка» NetCat – модуль, предназначенный для создания и отправки регулярных рассылок [3]. С помощью функционала модуля можно организовывать рассылку уведомлений для пользователей о различных событиях, произошедших на сайте, таких как обновления разделов, добавленных комментариев. Кроме этого, с помощью модуля можно организовать классическую рассылку, при которой пользователи оставляют свои контактные данные в специальной форме, а вы осуществляете отправку писем вручную либо автоматически. Основным недостатком всех существующих модулей в том, что они работают поверх CMS систем и не предоставляют функционала для использования их в качестве внешнего сервиса.

Вывод

Проведенный обзор существующих систем показал, что разработка веб-приложения для администрирования сервиса платных подписок «Subscription Service» является актуальной задачей, так как он может быть использован в качестве стороннего сервиса на любом сайте. После проведения анализа данных систем, был сделан вывод, что ни одна из них не предоставляет функционала работы с системами, используя их в качестве внешнего сервиса и большинство сервисов поддерживают уведомления только через электронную почту, что не всегда удобно.

2. ТРЕБОВАНИЯ К СИСТЕМЕ

Веб-приложение для администрирования сервиса платных подписок «Subscription Service», реализует функционал, обеспечивающий связь ресурсов со своей целевой аудиторией, предоставляя возможность подписываться на рассылку новой информации о товарах и услугах веб-ресурсов.

2.1. Функциональные требования

Требование можно определить как подробное описание того, что должно быть реализовано.

Существует два основных типа требований: – функциональные требования – какое поведение должна предлагать система; – нефункциональные требования – особое свойство или ограничение, накладываемое на систему [5].

веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно удовлетворять следующим функциональным требованиям:

1) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность добавления информации о новой подписки;

2) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность удаления существующих подписок;

3) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность редактирования существующих подписок;

4) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность добавления и удаления категорий подписок;

5) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность

рассылки уведомлений подписанным на нее пользователям;

б) веб-приложение для администрирования сервиса платных подписок «Subscription Service» должно предоставлять возможность просмотра списка подписок и пользователей, подписанных на нее.

2.2. Требования к удобству и комфортности пользовательского интерфейса

Для удобства и комфортности пользовательского интерфейса важны следующие факторы:

- прозрачная для пользователя навигация и целевая ориентация в программе;
- ясность и четкость понимания пользователем текстов, значения иконок, кнопок;
- быстрота обучения при работе с программой, для чего необходимо использовать преимущественно стандартные элементы взаимодействия, их традиционное или общепринятое их расположение.

2.3. Варианты использования системы

В ходе выполнения выпускной квалификационной работы мною была разработана диаграмма вариантов использования, которая изображена на рис. 1. Диаграмма включает в себя двух основных актеров, взаимодействующих с веб-приложением для администрирования сервиса платных подписок «Subscription Service», актера «Администратор» и актера «Пользователь».



Рис. 1. Диаграмма вариантов использования

Можно определить следующие основные варианты использования сервиса платных подписок.

Администратор:

- «авторизация»: администратору доступна возможность авторизации в системе, позволяющая идентифицировать его;
- «добавление новой подписки»: администратору доступна возможность добавления новой подписки в систему;
- «удаление подписки»: администратору доступна возможность удаления подписки из системы;
- «редактирование подписки»: администратору доступна возможность редактирования подписки в системе;
- «добавление категории»: администратору доступна возможность добавления новой категории подписок в систему;
- «удаление категории»: администратору доступна возможность удаления существующей категории подписок из системы;
- «просмотреть информацию о подписках»: администратору доступна возможность просмотреть список подписок и подписанных на них пользователей.

Пользователь:

- «авторизоваться»: пользователю доступна возможность авторизации, позволяющая идентифицировать пользователя;
- «найти доступные подписки»: пользователю доступна возможность найти доступные подписки;
- «подписаться»: пользователю доступна возможность создать заявку на подписку;
- «изменить подписку»: пользователю доступна возможность редактировать свою подписку;
- «удалить подписку»: пользователю доступна возможность удалить свою заявку на подписку;

– «настроить уведомления»: пользователю доступна возможность настроить дату и способ уведомления в системе подписок, связанных с его подписками.

Вывод

В процессе анализа требований были сформулированы основные функциональные и нефункциональные требования, предъявляемые к разрабатываемому веб-приложению, определены пользователи веб-приложения и варианты использования веб-приложения.

3. ПРОЕКТИРОВАНИЕ

3.1. Постановка задачи

Постановка задачи – это первый этап разработки веб-дизайна. Постановка задачи может быть оформлена в виде брифа. Бриф – краткая описательная форма согласительного порядка между заказчиком проекта и его исполнителем [8]. Бриф на разработку веб-дизайна для администрирования сервиса платных подписок «Subscription Service» в табл. 1.

Табл. 1. Бриф

1.	Руководитель:	Никольская Ксения Юрьевна
2.	Название сайта:	Subscription Service
3.	Задачи разработчика:	Разработка веб-дизайна главной страницы и второстепенных страниц
4.	Цель создания сайта:	Предоставление широкой аудитории возможности подписываться на уведомления и услуги веб-ресурсов
5.	Целевая аудитория:	От 16 лет
6.	На какие географические регионы должен быть ориентирован сайт:	Россия
7.	Языковые версии сайта:	Русский
8.	Сайты конкурентов:	Анализ сайтов конкурентов приведен в главе 1 «Обзор существующих решений».
9.	Структура сайта:	1. Главная 2. Страница авторизации 3. Страница регистрации 3. Страница с подписками 4. Страница с настройками подписок пользователя 9 Страница настройки категорий

10.	Компоненты сайта (новости, анкетирование, интернет магазин, баннерная реклама, счетчики посещений, др.):	Поисковая строка, фильтр, каталог подписок
11.	Цветовая гамма сайта:	На усмотрение дизайнера
12.	Браузеры, в которых будет просматриваться сайт, и их минимальная версия:	Google Chrome 35, Mozilla Firefox 28, Internet Explorer 11, Opera 20
13.	График работ:	Лейаут страниц: 25.03.2017 Верстка страниц: 25.04.2017 Реализация: 15.05.2017 Передача готового проекта: 23.05.2017
14.	Передача разработки:	HTML-файлы, CSS-файлы, PHP-файлы

3.2. Макет главной страницы веб-сайта

В рамках проектирования дизайна сервиса платных подписок «Subscription Service» мной были разработаны модульные сетка основных страниц веб-приложения (см. приложение), пример панели администратора представлен на рис. 2.

Модульная сетка – это разделение макета вертикальными и/или горизонтальными направляющими включая поля, пространство и некоторое количество столбцов для того, чтобы заложить основу для организации контента [8].

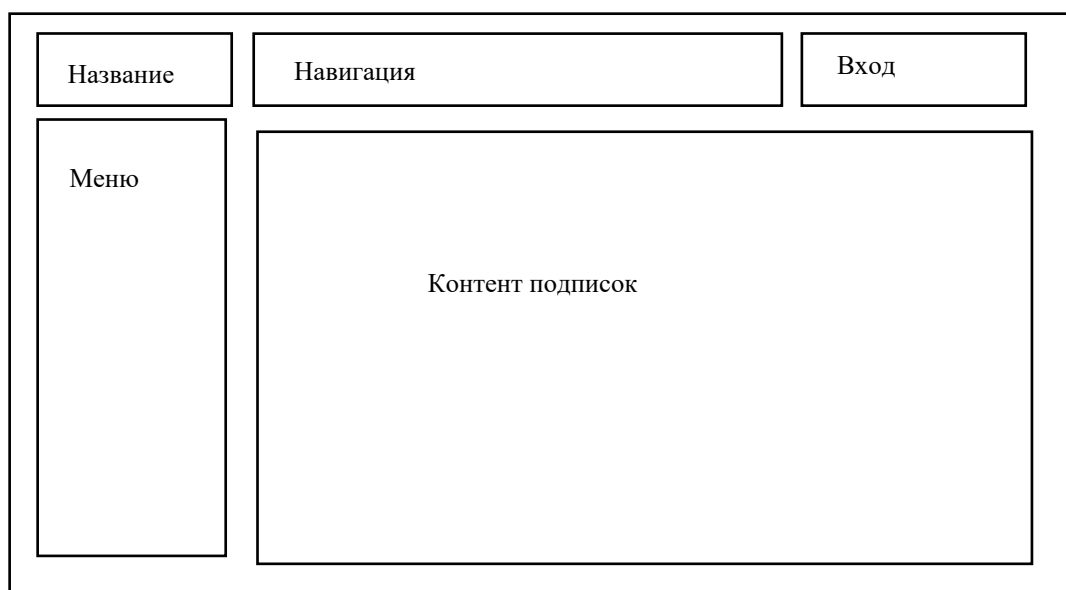


Рис. 2. Макет панели администратора

Модульная сетка – это разделение макета вертикальными и/или горизонтальными направляющими включая поля, пространство и некоторое количество столбцов для того, чтобы заложить основу для организации контента [8]. Сайт состоит из трех основных блоков: «Шапка», «Информационный блок», «Подвал».

«Шапка» (header) – верхняя, титульная часть дизайн макета веб-страницы. Она содержит название сервиса, кнопку входа и навигацию по сайту, с разделами «Главная», «Мы в Контакте».

В левой части макета находится основное меню сайта, которое будет содержать ссылка для перехода на страницы с информацией о пользователях, приложениях и категориях.

Блок фильтрации содержит основные фильтры по контенту страницы.

3.3. Web-дизайн страниц

Web-дизайн – разновидность дизайна, в задачи которой входит проектирование пользовательских веб-интерфейсов для сайтов или веб-приложений, призванный обеспечить им высокие потребительские свойства и эстетические качества [14]. Свойства web-дизайна [15]:

- грамотная структуру сайта;
- удобная навигация;
- красивый и хорошо подсознательно-воспринимаемый дизайн;
- правильно поданный и удобно расположенный текст.

Опираясь на вышеперечисленные свойства, мной было решено использовать следующий дизайн.

Цветовая гамма страниц состоит из четырех цветов:

- Серый, белый, красный, зеленый, серый.

Использовались 2 шрифта:

- Verdana и Arial.

4. РЕАЛИЗАЦИЯ

На данном этапе были рассмотрены средства реализации, обоснование их выбора и этапы реализации веб-приложения.

4.1. Выбор средств реализации

Для визуализации работы приложения использовались шаблоны, написанные на HTML5, применялись таблицы стилей CSS. Рассмотрим каждый из элементов более подробно.

HTML (Hyper Text Markup Language) [11] – это язык для описания веб-страниц (дословно расшифровывается как «язык гипертекстовой разметки»). Под языком разметки понимается набор тегов, каждый из которых описывает содержимое документа. Язык HTML интерпретируется браузерами и полученный в результате интерпретации форматированный текст отображается в понятном для человека виде.

CSS (от англ. Cascading Style Sheets – каскадные таблицы стилей) – это набор параметров форматирования, который применяется к элементам веб-страницы для управления их видом и положением. Таблицы стилей позволяют изменить внешний вид и расположение всех страниц веб-сайта путем редактирования всего лишь одного файла [16].

4.2. Интерфейс главной страницы

Страницы сервиса платных подписок «Subscription Service» имеют некоторое фиксированное содержание, такое как меню, заголовок (шапка), поля для отображения информации, для этого было решено создать основной шаблон страницы (см. рис 3), скриншот страницы представлен в приложении 3.

Основной шаблон (layout) – это неизменяемая часть страницы, расположение элементов и блоков относительно друг друга в рамках страницы или интерфейса, включающая в себя основные статические части сайта: шапку, меню, подвал.

Структура содержимого задается с помощью HTML, для оформления внешнего вида контента используется CSS.


```

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">    <meta name="viewport"
        content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-scale=1.0,
minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>kurdIT>
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css" integri-
ty="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u" cros-
sorigin="anonymous">
    <link href="{{ URL::to('css/charisma-app.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/jquery.noty.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/noty_theme_default.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/elfinder.min.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/elfinder.theme.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/jquery.iphone.toggle.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/uploadify.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/animate.min.css') }}" rel="stylesheet">
    <link href="{{ URL::to('css/jquery.dataTables.min.css') }}" rel="stylesheet">
</head>
<body>
@include('account.includes.header')
<div class="ch-container">
    <div class="row">
        @include('account.includes.left-menu')
        @if ( \Illuminate\Support\Facades\Session::has('flash_message'))
            <div class="col-xs-4 alert alert-success">                {{
\Illuminate\Support\Facades\Session::get('flash_message') }}
            </div>
        @endif @yield('content')
    </div>
</div>
<script src="https://code.jquery.com/jquery-3.1.1.min.js"
    integrity="sha256-hVVnYaiADRTO2PzUGmuLJr8BLUSjGIZsDYGmIJLv2b8="
    crossorigin="anonymous">
</script>
<script src="https://cdn.datatables.net/1.10.12/js/jquery.dataTables.min.js"
type="text/javascript"></script>
</script>
<script>
    $(function() {
        console.log('work');
        $('table').DataTable({
            mark: true
        });
    });
</script>
@yield('scripts')
</body>
</html>

```

Рис. 3. Основной шаблон страницы

```

<style>
    div.search span {
        display: block;
    }

    div.panel {
        margin-bottom: 15px;
    }

    div.panel .panel-body p:last-child {
        margin-bottom: 0;
    }

    /**
     * Styles for datatables.mark.js
     */
    mark {
        padding: 0;
        background: #f1c40f;
    }
</style>

```

Рис. 4. CSS для основной страницы

4.3. Интерфейс авторизации

Аутентификация – это процесс определения личности пользователя [5]. Этот процесс, включает в себя сравнения введённого пользователем имя и пароля с паролем, сохранённым в базе данных пользователей;

В процессе создания элементов страницы, использовались полезные возможности HTML5. Полю, куда пользователь будет вводить свою почту, был присвоен тип email, это позволяет браузеру проверить, ввел ли пользователь правильный формат адрес электронной почты. Так же использовался атрибут required. Элементы, которые содержат атрибут required являются обязательными для заполнения, браузеры, которые поддерживают этот атрибут, не позволят пользователю отправить форму пока это поле не будет заполнено, без использования JavaScript.

В соответствии со спроектированным шаблоном был реализован макет авторизации представленный на рис.5, скриншот страницы представлен в приложении 4.

```
@extends('layouts.app')
@section('content')
<div class="container">
  <div class="row">
    <div class="col-md-8 col-md-offset-2">
      <div class="panel panel-default">
        <div class="panel-heading">Авторизация</div>
        <div class="panel-body">
          <form class="form-horizontal" role="form" method="POST" action="{{
url('/login') }}">
            {{ csrf_field() }}

            <div class="form-group{{ $errors->has('email') ? ' has-error' : ''
}}">
              <label for="email" class="col-md-4 control-label">E-
Mail</label>

              <div class="col-md-6">
                <input id="email" type="email" class="form-control"
name="email" value="{{ old('email') }}" required autofocus>

                @if ($errors->has('email'))
                  <span class="help-block">
                    <strong>{{ $errors->first('email') }}</strong>
                  </span>
                @endif
              </div>
            </div>

            <div class="form-group{{ $errors->has('password') ? ' has-error' :
'' }}">
              <label for="password" class="col-md-4 control-
label">Пароль</label>

              <div class="col-md-6">
                <input id="password" type="password" class="form-control"
name="password" required>

                @if ($errors->has('password'))
                  <span class="help-block">
                    <strong>{{ $errors->first('password') }}</strong>
                  </span>
                @endif
              </div>
            </div>

            <div class="form-group">
              <div class="col-md-6 col-md-offset-4">
                <div class="checkbox">
                  <input type="checkbox" name="remember"> Запомнить
меня

                </div>
              </div>
            </div>

            <div class="form-group">
              <div class="col-md-8 col-md-offset-4">
                <button type="submit" class="btn btn-primary">
Вход
                </button>
                <a class="btn btn-link" href="{{ url('/password/reset')
}} ">
                  Забыли пароль?
                </a>
              </div>
            </div>
          </form>
        </div>
      </div>
    </div>
  </div>
</div>
@endsection
```

Рис. 5. Страница авторизации в веб-приложении

5. ТЕСТИРОВАНИЕ

Для проверки эргономичности веб-приложения для администрирования сервиса платных подписок «Subscription Service» было проведено юзабилити-тестирование. Юзабилити-тестирование позволяет найти проблемы взаимодействия пользователя и веб-приложения, узнать, что пользователи думают и чувствуют, находясь на сайте, понять ожидания и потребности посетителей сайта [7]. Для проведения юзабилити-тестирования необходимо сформировать цели и задачи тестирования, подобрать респондентов, организовать инфраструктуру для тестирования. После проведения тестирования следует проанализировать ответы и комментарии респондентов, оформить результат анализа в виде отчета, указав выявленные ошибки и недочеты [6].

Все поставленные перед респондентами задачи были полностью выполнены, выявленные недочеты устранены. Результаты юзабилити-тестирования веб-приложения для администрирования сервиса платных подписок «Subscription Service» занесены в отчет (приложение 1). Для проверки функциональности веб-приложения было проведено функциональное тестирование. Суть функционального тестирования заключается в проверке работы всего описанного функционала: всех функций сайта, поисковой строки, гиперссылок, поиск нерабочих ссылок, проверки подгрузки файлов на сервер, просмотр на соответствие содержимого страниц сайта исходному контенту. Результаты функционального тестирования совпали с ожидаемыми (приложение 2).

ЗАКЛЮЧЕНИЕ

мной были решены следующие

- Проанализированы требования к веб-приложению;
- Разработана структура и дизайн веб-приложения;
- Реализованы макеты веб-приложения;
- Проведено тестирование пользовательского интерфейса.

ЛИТЕРАТУРА

1. Арлоу Д., Нейштадт А. UML 2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. – М.: Символ-Плюс, 2007. – 624 с.
2. Беллиньясо М. Разработка Web-приложений : Задача-проект-решение. М.: Изд. дом «Вильямс», 2007. – 640 с.
3. Бородаев Д. В. Веб-сайт как объект графического дизайна. Монография. – "Септима ЛТД", 2006. – С. 288 с.
4. Веб-сервис FeedBurner . [Электронный ресурс] URL: <http://feedburner.com> (дата обращения: 17.04.2017).
5. Гончаров А.Ю. Web-дизайн: HTML, JavaScript и CSS. Карманный справочник. – "КУДИЦ-ПРЕСС", 2007. – 319 с.
6. Канер С., Фолк Дж., Кек Нгуен Е. Тестирование программного обеспечения. – Киев: ДиаСофт, 2001. – 538 с.
7. Кузнецов А. Дизайн и юзабилити образовательных Интернет-ресурсов. // E LearningWorld, 2004. – № 3. – С. 30-37.
8. Кузнецов А.М. Мартынов В.В. Требования к графическому дизайну и юзабилити образовательных порталов. – М.: Просвещение, 2003. – С. 365-420.
9. Лабберс П., Олберс Б., Салим Ф. HTML5 для профессионалов: мощные инструменты для разработки современных веб - приложений.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2011. – 272 с.
10. Модуль «Подписка и рассылка» NetCat. [Электронный ресурс] URL: <https://netcat.ru/developers/docs/modules/module-subscriptions-and-newsletters/> (дата обращения: 17.04.2017).
11. Нильсен. Я. Веб-дизайн. – СПб.: Символ-Плюс, 2003. – 512 с.
12. Плагин Subscribe2. [Электронный ресурс] URL: <http://wordpress.org/plugins/subscribe2/> (дата обращения: 17.04.2017).
13. Справочник по HTML. [Электронный ресурс] URL: <http://htmlbook.ru> (дата обращения: 24.03.2014).

14. Фаулер М. UML. Основы. Краткое руководство по стандартному языку объектного моделирования. – СПб.: Символ-Плюс, 2005. – 195 с.

15. Шмитт К. CSS. Рецепты программирования. 3-е изд.: Пер. с англ. – М.: Издательство «Русская редакция»; СПб. «БХВ-Петербург», 2011. – 672 с.

ПРИЛОЖЕНИЕ

Приложение 1

Отчет о юзабилити-тестировании

1. Объект исследования

Объектом исследования является веб-приложения для администрирования сервиса платных подписок «Subscription Service» – это приложения, целью которого является демонстрация широкой аудитории подписок на контент и услуги веб-ресурсов.

2. Метод исследования

В качестве метода исследования был использован классический метод юзабилити-тестирования «Мысли вслух».

3. План проведения тестирования

1. Планирование

- а. Разработка заданий
- б. Набор участников тестирования

2. Проведение тестирования

3. Анализ полученных данных

4. Методика проведения тестирования

Для проведения тестирования использовалось следующее оборудование и ПО:

- 1. ноутбук (15")
- 2. микрофон (встроен в ноутбук)
- 3. браузер Mozilla Firefox 3.6.

Модератор выдает сразу весь список заданий участнику. Участник выполняет их последовательно самостоятельно, проговаривать свои мысли, чувства и мнения, возникающие в процессе взаимодействия с сайтом. Комментарии записываются через микрофон и затем представляются модератором в виде протокола тестирования.

5. Протокол заданий

Информационные задачи

1. Отправная точка: любая страница сайта.

Задание: найти способ авторизоваться на сайте.

2. Отправная точка: любая страница сайта.

Задание: найти информацию о какой-либо подписке.

3. Отправная точка: любая страница сайта.

Задание: найти информацию о категориях подписок.

Функциональные задачи

1. Отправная точка: главная страница сайта.

Задание: пройти процедуру авторизации.

2. Отправная точка: страница каталога подписок.

Задание: с помощью фильтра получить все подписки по категории «бизнес».

3. Отправная точка: страница каталога подписок.

Задание: создать подписку.

4. Отправная точка: страница каталога подписок.

Задание: настроить уведомления по подпискам

5. Респонденты

1. Саид Пешкар Мужда, студент очного отделения.

2. А.И. Рекачинский, студент очного отделения.

3. яна Алексеевна Щеголева, студент заочного отделения.

6. Результаты тестирования

Табл. 2.

Респонденты	Доля выполнен ных задач	Доля правильно выполненных задач	Время выполнения	Кол-во ошибок
Саид Пешкар Мужда	100%	100%	7	2
А.И. Рекачинск ий	100%	100%	7	2
яна Алексеевна Щеголева	100%	100%	3	1

Выявленные недостатки

В ходе исследования были выявлены следующие недочеты. У респондентов вызвало дискомфорт чтение разделов меню из-за недостаточно большого размера шрифта. Данный недочет был исправлен.

Приложение 2

Табл. 3. Результаты функционального тестирования

№	Название теста	Шаги	Ожидаемый результат	Тест пройден?
1	Авторизация	1. Пользователь заходит в систему. 2. Заполняет предоставленные ему формы. 3. Нажимает на кнопку «Войти».	Авторизуется и автоматически переходит на страницу с рабочей панелью	да
2	Поиск подписок по категории «бизнес» с помощью фильтра	1. Используя фильтр, пользователь выбирает категорию «бизнес» 2. Нажимает кнопку «Применить»	Вывод всех подписок из категории «бизнес»	Да
3	Поиск подписок по категории «программирования» с помощью фильтра	1. Используя фильтр, пользователь выбирает категорию «программирования» 2. Нажимает кнопку «Применить»	Вывод всех подписок из категории «программирования»	Да
4	Создание подписки	1. Пользователь на рабочей панели нажимает на кнопку «Подписаться». 2. Заполняет предоставленную ему форму. 3. Нажимает на кнопку «Сохранить»	Подписка сохраняется в базе данных.	Да

Приложение 3

Основные макеты веб-страниц

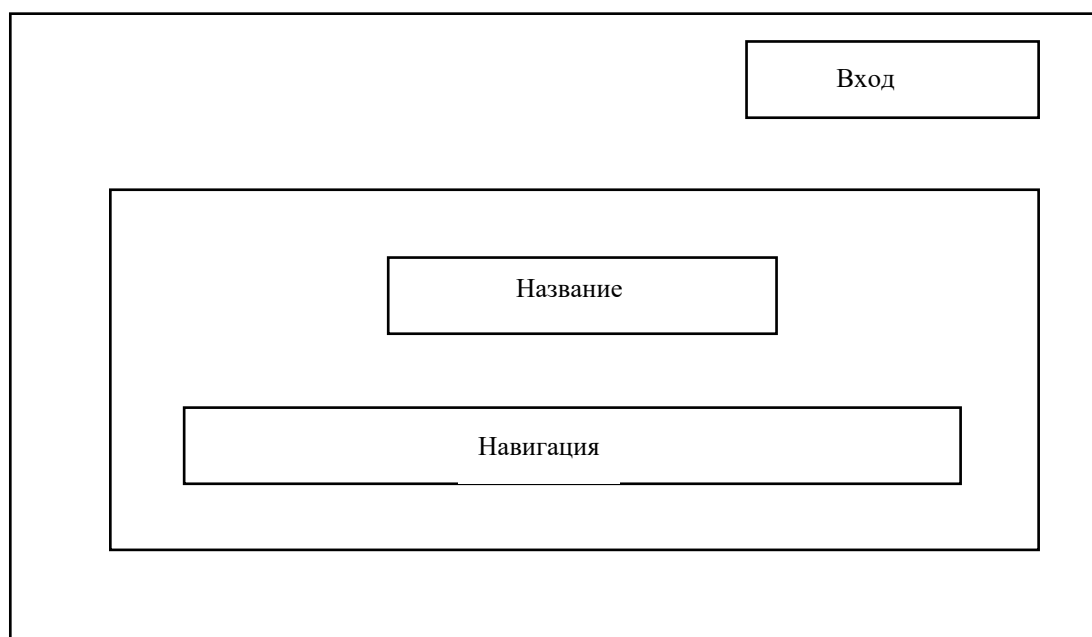


Рис. 6. Макет главной страницы

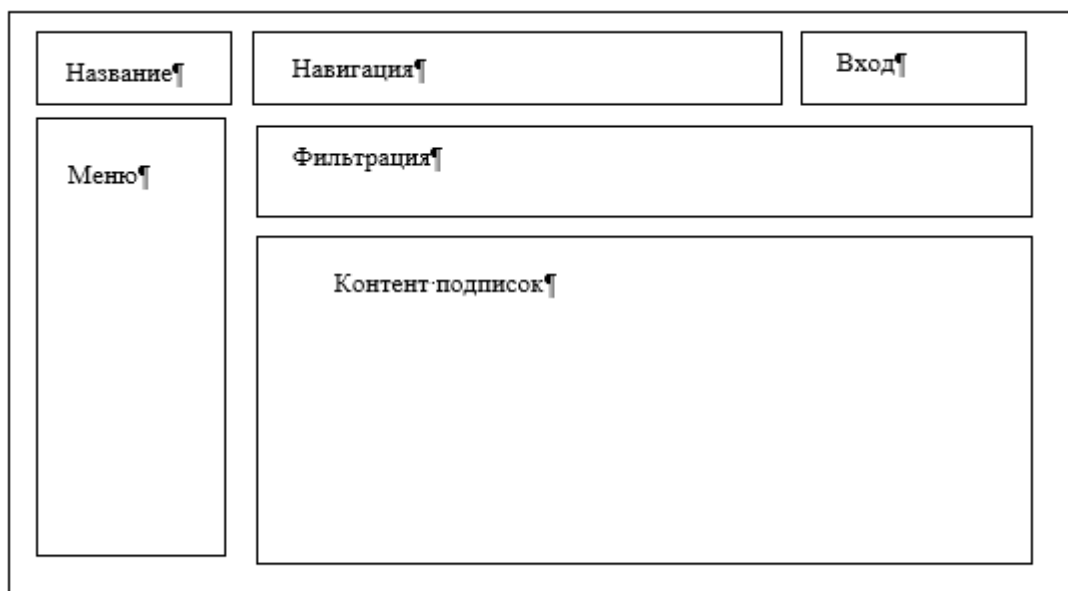


Рис. 7. Макет панели администратора

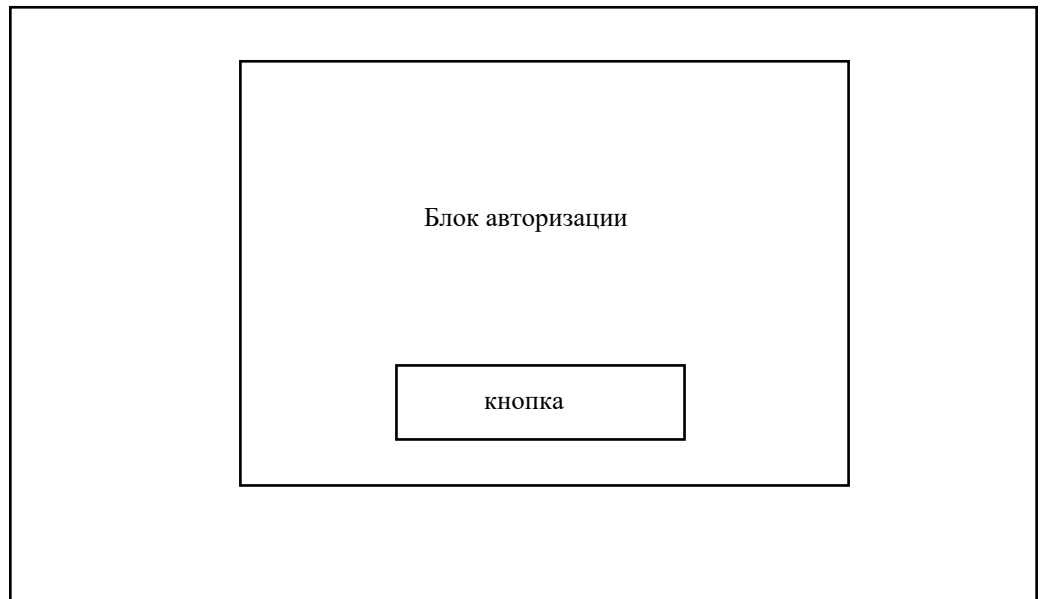


Рис. 8. Макет авторизации

Приложение 4

Скриншоты веб-страниц



Рис. 9. Внешний вид главной страницы

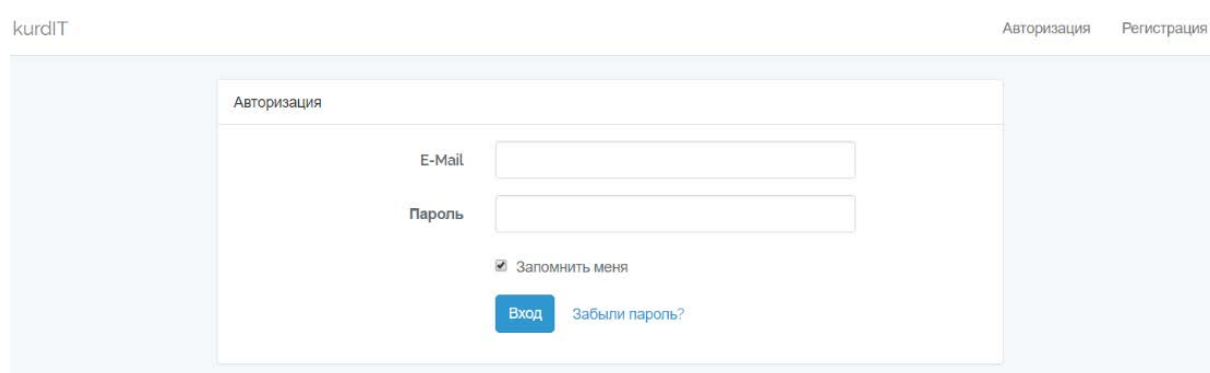


Рис. 10. Скриншот страницы авторизации

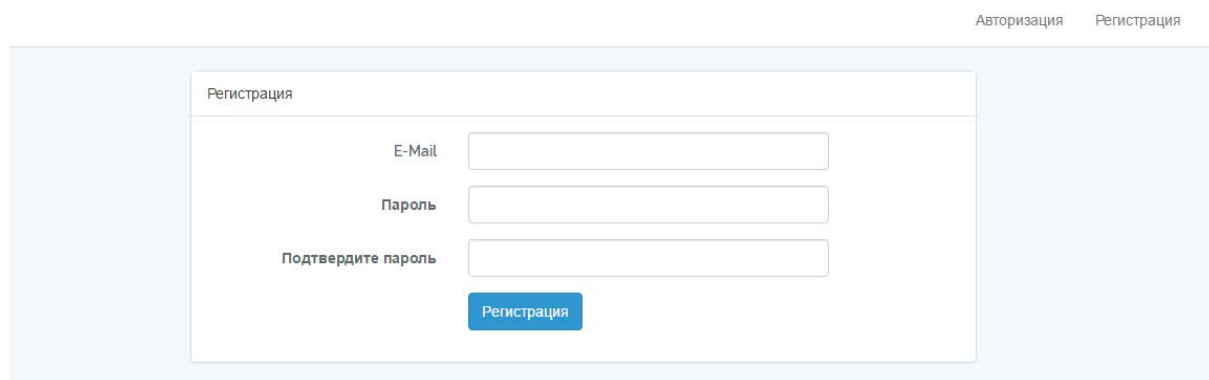


Рис. 11. Скриншот страницы регистрации

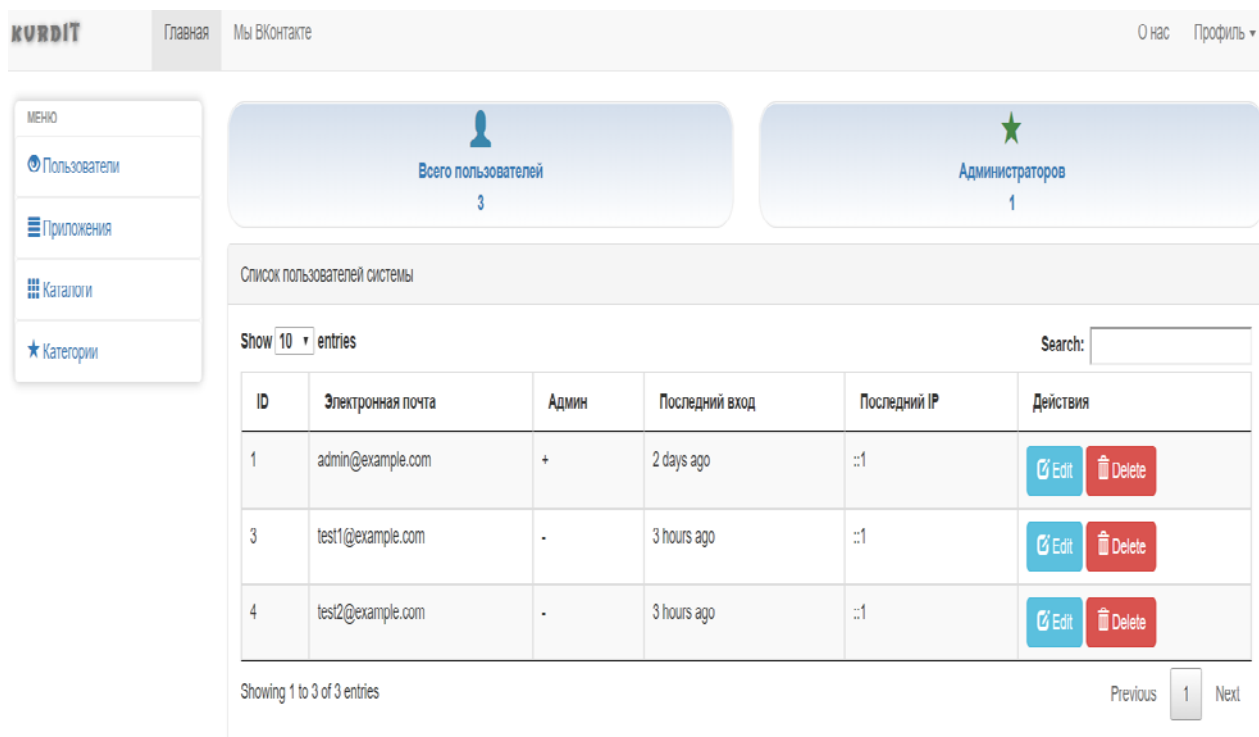


Рис. 12. Скриншот панели администратора

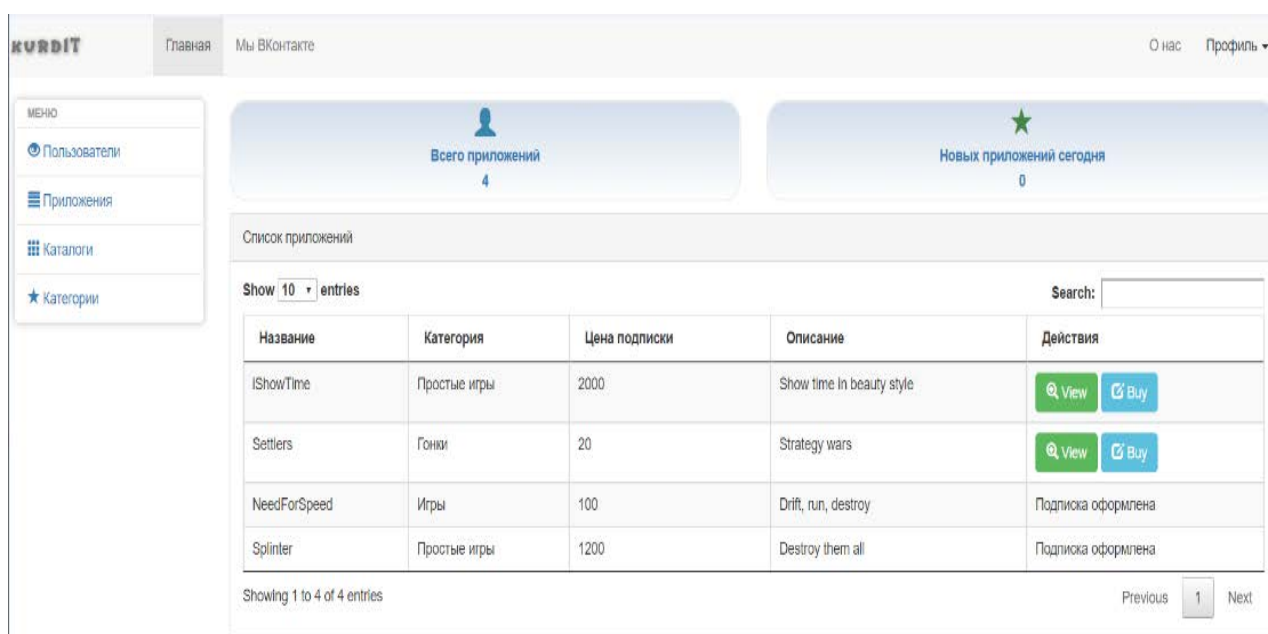


Рис. 13. Скриншот панели пользователя