

Министерство образования и науки Российской Федерации  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ (НИ ТГУ)

Факультет информатики  
Кафедра теоретических основ информатики (ТОИ)

ДОПУСТИТЬ К ЗАЩИТЕ В ГЭК

Руководитель ООП

\_\_\_\_\_ А.Л. Фукс

« \_\_\_\_ » \_\_\_\_\_ 2016 г.

**ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА БАКАЛАВРА**

**РАЗРАБОТКА ВЕБ-СЕРВИСА ДЛЯ ИНТЕРНЕТ-МАРКЕТИНГА ЧАСТНЫХ  
РАБОТНИКОВ МЕДИЦИНЫ**

по основной образовательной программе подготовки бакалавров  
направление подготовки

02.03.02 Фундаментальная информатика и информационные технологии

Сарычев Максим Олегович

Руководитель ВКР

ген. директор ООО «Инвент»

\_\_\_\_\_ А.С. Передерей

«подпись»

« \_\_\_\_ » \_\_\_\_\_ 2016 г.

Автор работы студент

группы № 1421

\_\_\_\_\_ М.О. Сарычев

«подпись»

## РЕФЕРАТ

Выпускная квалификационная работа бакалавра 34 с., 22 рис., 1 таблица, 7 источников.

ВЕБ-СЕРВИС, МЕДИЦИНСКАЯ СФЕРА, ИНТЕРНЕТ-МАРКЕТИНГ, PHP, ФРЕЙМВОРК, LARAVEL, MVC

Цель работы: разработка веб-сервиса для интернет-маркетинга частных работников медицины.

Методология: унифицированный процесс разработки программного обеспечения.

Результат работы: реализованный веб-сервис внедрен в работу и размещен в сети интернет

## Содержание

Введение .....	4
1. Анализ .....	6
1.1 Анализ существующих решений .....	6
1.2 Требования к разрабатываемой системе .....	10
2. Проектирование .....	11
2.1. Выбор платформы разработки .....	11
2.2 Особенности архитектуры Laravel .....	12
2.2.1 Общий обзор .....	12
2.2.2 Ключевые особенности Laravel .....	13
2.3 Проектирование базы данных .....	15
2.4 Проектирование архитектуры приложения .....	18
3. Реализация .....	22
Заключение .....	29
Список использованной литературы .....	30
Приложение 1. Руководство пользователя .....	31

## Введение

**Интернет-маркетинг** – это комплекс мероприятий, цель которых – продвижение сайта, товара или услуги в интернете. В качестве продукта продвижения также может выступать конкретная личность.

Роль эффективного маркетинга и продвижения коммерческого продукта в Интернете в современном мире трудно переоценить. И это касается не только тех товаров/услуг, которые мы привыкли заказывать онлайн, и той информации, которую мы ищем на тематических сайтах. Стремительно растет роль интернет-маркетинга для организаций, работающих в сфере медицины, а также частных врачей. Это связано с несколькими основными тенденциями в медицинском интернет-маркетинге:

- Все больше пациентов ищут врачей, лечебные учреждения, санатории и поставщиков товаров медицинского назначения через Интернет.
- Перед тем, как воспользоваться медицинскими услугами, пациенты знакомятся с отзывами на сайтах и информацией, размещенной в Интернете.
- Все больше медицинских услуг предоставляется дистанционно. Так, например, стремительно развивается рынок онлайн-консультаций врачей. Достичь успеха в этом направлении позволяют современные инструменты продвижения и коммуникации.
- Новейшие технологии дают возможность консолидировать информацию о пациенте в электронных базах, что позволяет эффективно предоставлять медицинские услуги, организовывать медицинское обеспечение пациента и решать целый ряд других задач.

Под интернет-маркетингом можно понимать комплексную работу по взаимодействию с целевой аудиторией. Когда мы получаем обратную связь по своим услугам и товарам, оказываем дополнительный сервис (полезная тематическая рассылка, дистанционная помощь), информируем об услугах и

продуктах, доносим их ценность и уникальность. Проще говоря, интернет-маркетинг – это повышение, сохранение и продвижение своей репутации, как квалификационного специалиста в своей области медицины, в сети Интернет.

Подводя итог вышесказанному, можно сказать, что интернет-маркетинг одного частного врача включает в себя самые различные инструменты: сайт (обычно блог), группы(профиль) в социальных сетях, контекстная реклама, контент-маркетинг, видео реклама и обучающие видеоролики, обработка обратной связи с пациентами, тематические рассылки для пациентов, PR-кампании. Ежедневная обработка всех пунктов этого списка – весьма трудоемкий процесс, занимающий много времени. Из этого следует необходимость создания специального сервиса, упрощающего управление своим интернет-маркетингом.

Целью данной работы является разработка веб-сервиса, который предоставит частному врачу интерфейс для управления состоянием своего интернет-маркетинга из одного места, в удобном виде. Задача будет решаться для реалий Американского рынка медицинских услуг, так как в России и странах СНГ такая практика только начинает зарождаться, и, пока зарождается она только для частных клиник. Задача же предполагает, что основным ее пользователем будет частный американский врач.

## **1.Анализ**

Для того чтобы разрабатываемая система была эффективно реализована и имела успех, необходимо с самого начала понимать, что необходимо получить в результате. Поэтому сначала проанализируем существующие аналогичные решения и сформулируем основные требования к разрабатываемой системе.

### **1.1 Анализ существующих решений**

Прежде чем приступить к моделированию и проектированию проекта, необходимо изучить программные средства, которые уже функционируют и решают аналогичные задачи. Как уже упоминалось выше, практика использования интернет-маркетинга для частных врачей популярна только на Американском рынке. На Российском же рынке, интернет-маркетинг используется, в основном, для клиник, что для нас не очень информативно.

#### **1.1.1 Ratemds.com**

Сервис позволяет зарегистрироваться как частный врач и получать отзывы о себе. Пациенты могут зайти на сайт и читать\оставлять отзывы о конкретном враче. Сервис с большой посещаемостью и большой базой докторов. Пользование сайтом бесплатно. Данный сервис подходит только для одного инструмента интернет-маркетинга: получение и обработка обратной связи от пациентов, но только с этого сайта, что очень ограничивает возможную потенциальную аудиторию. На рисунке 1 изображен профиль одного врача с отзывами и оценками от пациентов.

The screenshot shows the RateMDs.com website interface. At the top, there is a search bar for finding a doctor by name and a navigation menu with options like 'Find A Doctor', 'Find A Facility', 'Health Library', 'Health Blog', and 'Health Forum'. The main content area features a profile for Dr. Max Gouverne, a Cosmetic Surgeon / Plastic Surgeon. The profile includes a photo, a 'Follow' button, and social media links for Twitter and Facebook. A map shows the location of Dr. Max Gouverne, MD Cosmetic Surgery. Below the profile, there is a 'You may also like' section featuring Dr. Theodore A. Saieh. The 'Rate Dr. Max Gouverne' section displays star ratings for Staff (5 stars), Punctuality (5 stars), Helpfulness (5 stars), and Knowledge (5 stars). A detailed review is provided below the ratings, dated February 26, 2015.

Рисунок 1 – Профиль мед. работника на сайте Ratemds.com

### 1.1.2 Doctorbase.com

Данный веб-сервис позволяет создавать электронное расписание приема пациентов, отправлять пациенту запрос об обратной связи после приема, из которой в дальнейшем формируется отзыв, отправлять пациентам напоминания о записи не только по электронной почте, но также и на мобильный телефон, предоставляет сервис вопросов-ответов, благодаря которому можно продемонстрировать свою квалификацию потенциальным клиентам. Сервис платный, минимальный план стоит \$150 в месяц. На рисунке 2 можно увидеть список возможностей для оплативших подписку врачей.

Сервис с хорошими возможностями, но использует мало инструментов для интернет-маркетинга. Больше подходит для упрощения работы врачей с большой базой клиентов, нежели для комплексного управления интернет-маркетингом.

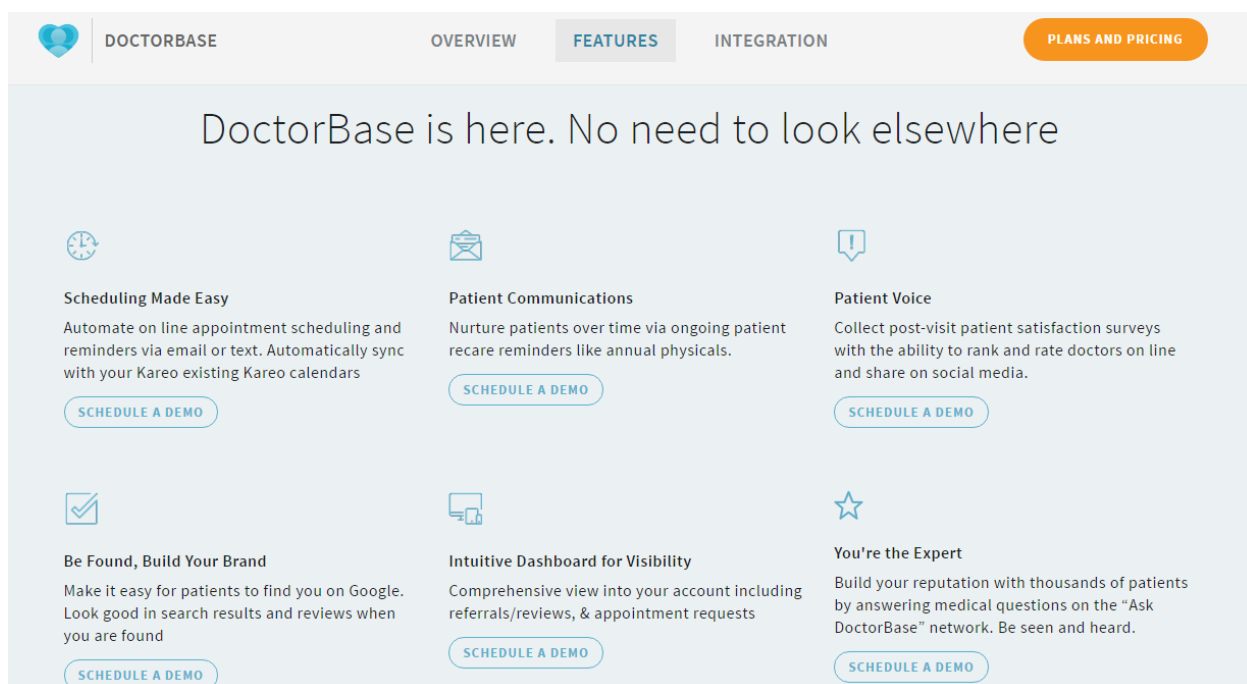
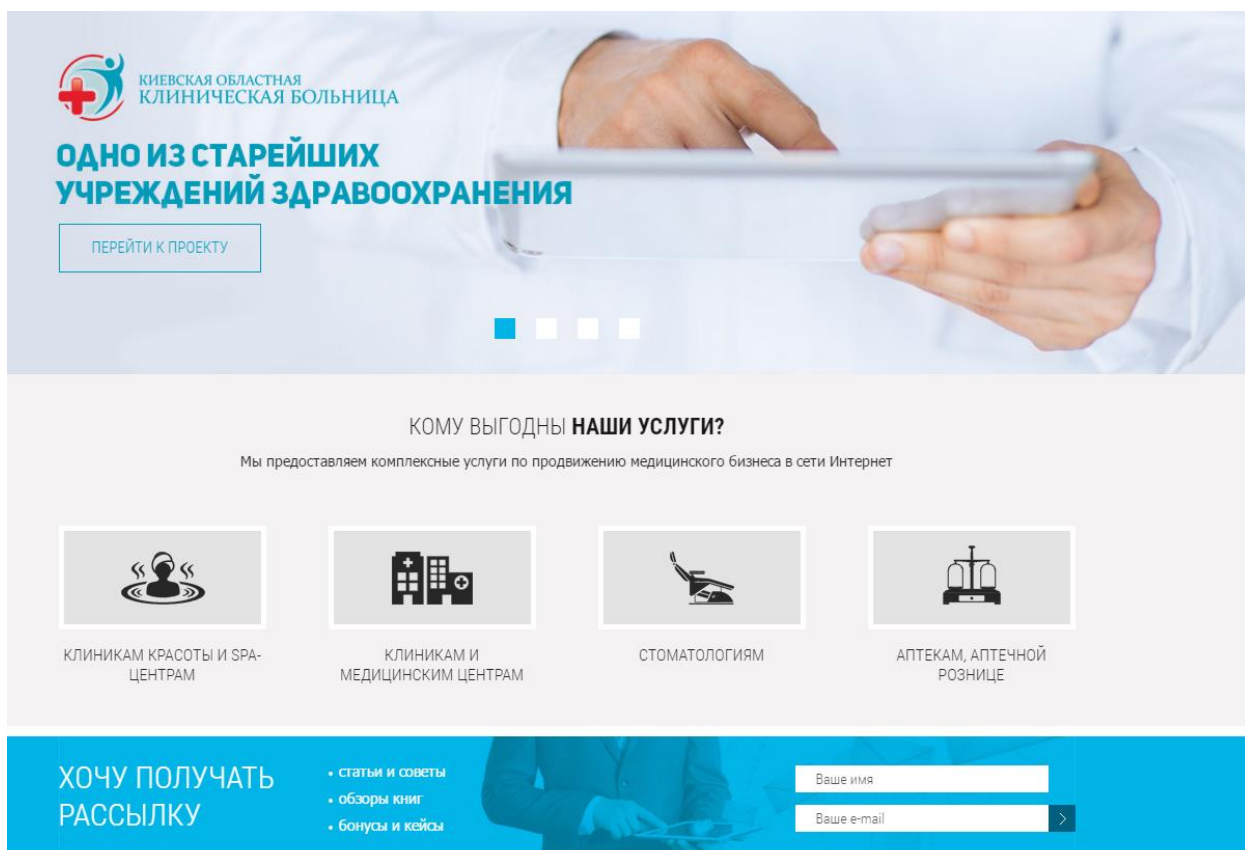


Рисунок 2 – Фрагмент страницы веб-сервиса «Doctorbase.com».

### 1.1.3 Medmarketing.ua

Рассмотрим сайт организации, работающей на территории СНГ. Фрагмент страницы представлен на рисунке 3. Сайт предлагает услугу продвижения в сети интернет сайтов медицинских клиник, стоматологий, медицинских центров. Анализ сайтов с подобной тематикой позволяет сделать вывод, что для Российского рынка понятие интернет-маркетинга распространяется исключительно на целые клиники и прочие организации медицинской сферы.





*Рисунок 3 – Фрагмент страницы сайта Medmarketing.ua*

В данной главе были рассмотрены веб-сервисы с функционалом, наиболее похожем на тот, что мы собираемся реализовать в рамках данной работы. Были проанализированы и другие сервисы, но все они так или иначе повторяют функционал сервисов из данного обзора.

Проанализировав существующие системы, можно сделать вывод, что в интернете существуют веб-сервисы с тематикой нашей задачи, но они, как правило, не предоставляют всех необходимых инструментов для интернет-маркетинга. Было решено разработать веб-сервис, который бы позволял использовать большинство возможных инструментов для интернет-маркетинга и предоставлял бы удобный пользовательский интерфейс для управления ими.

## 1.2 Требования к разрабатываемой системе

Разрабатываемый веб-сервис должен:

- содержать административную часть
- содержать пользовательскую часть
- иметь возможность бесплатного использования

Система должна предоставлять возможность использования медицинским работником следующих инструментов интернет-маркетинга:

- *Наполнение контентом личного блога* – интеграция блога с возможностью полного удаленного управления блогом. Также администраторами будет предоставляться готовый контент, которым смогут воспользоваться пользователи.
- *Размещение материалов\статей в социальных сетях* – интеграция с профилями клиента в социальных сетях с возможностью размещения контента от имени пользователя, в том числе, с функцией размещения контента по расписанию.
- *Отправка почтовых писем пациентам* – предоставление пользователю возможности управлять списками электронных почтовых адресов пациентов, проводить тематические рассылки.
- *Сервисы отзывов* - интеграция с аккаунтами на специализированных сервисах отзывов для обработки обратной связи с пациентами и сбор оценок\отзывов в одно место для последующего анализа и вывода в удобном виде.

*Административная часть* – часть веб-сайта, доступная только пользователям с административным доступом. Предоставляет интерфейс для взаимодействия с контентом сайта и выполнения настроек пользовательской части.

*Пользовательская часть* – общедоступная часть веб-сайта. Предоставляет пользователям интерфейс для настройки и управления инструментами для интернет-маркетинга, представленными на сайте.

## 2. Проектирование

### 2.1. Выбор платформы разработки

При проектировании любой системы, одним из первых вопросов, которые предстоит решить разработчику – платформа для программной реализации системы. Для случая разработки веб-сайтов, выбор делится на два варианта:

*1. Разрабатывать сайт «с нуля».*

Данный вариант позволяет иметь полный контроль над разработкой и ее результатами. Разработчик имеет полную свободу выбора в принятии решений. В то же время, скорее всего, придется реализовывать функционал, который в других случаях может уже быть реализован и готовым к использованию.

*2. Разрабатывать сайт, используя CMS в качестве основы.*

*CMS (Content Management System)* – система управления контентом.

Это информационная система или компьютерная программа, используемая для обеспечения и организации совместного процесса создания, редактирования и управления контентом (то есть содержимым). Данное решение очень распространено среди сайтов, функционал которых не сильно уходит за рамки «стандартного», либо в случае сильных временных ограничений для разработчика. С другой стороны, в качестве недостатков мы имеем необходимость использовать те стандарты, структуру и API, которые предоставляет данная система.

Так как никаких временных ограничений на проект не накладывалось и в разрабатываемой системе предусматривался не совсем «стандартный» функционал, было принято решение отказаться от варианта с CMS в пользу разработки сайта «с нуля».

Идеальным вариантом для разработки нашей системы был бы фреймворк, который бы поставлялся с частью уже готового «стандартного» функционала, чтобы свести минусы выбранного способа разработки к минимуму.

Принимая во внимание вышесказанное, было принято решение использовать PHP-фреймворк Laravel[2].

## 2.2 Особенности архитектуры Laravel

### 2.2.1 Общий обзор

Laravel — бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC.

Архитектурная модель(каркас) MVC (*Model-View-Controller, Модель-Представление-Контроллер*) – схема совместного использование нескольких шаблонов проектирования, позволяющая разделить данные, представление и обработку действий пользователя на три отдельных компонента:

- 1) *Модель* - предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
- 2) *Представление* – отвечает за визуализацию информации.
- 3) *Контроллер* - обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Основная цель применения этой схемы состоит в отделении бизнес-логики (модели) от её визуализации (представления, вида). За счет такого разделения повышается возможность повторного использования.

Диаграмма использования MVC в Laravel хорошо представлена на *рисунке 4*.

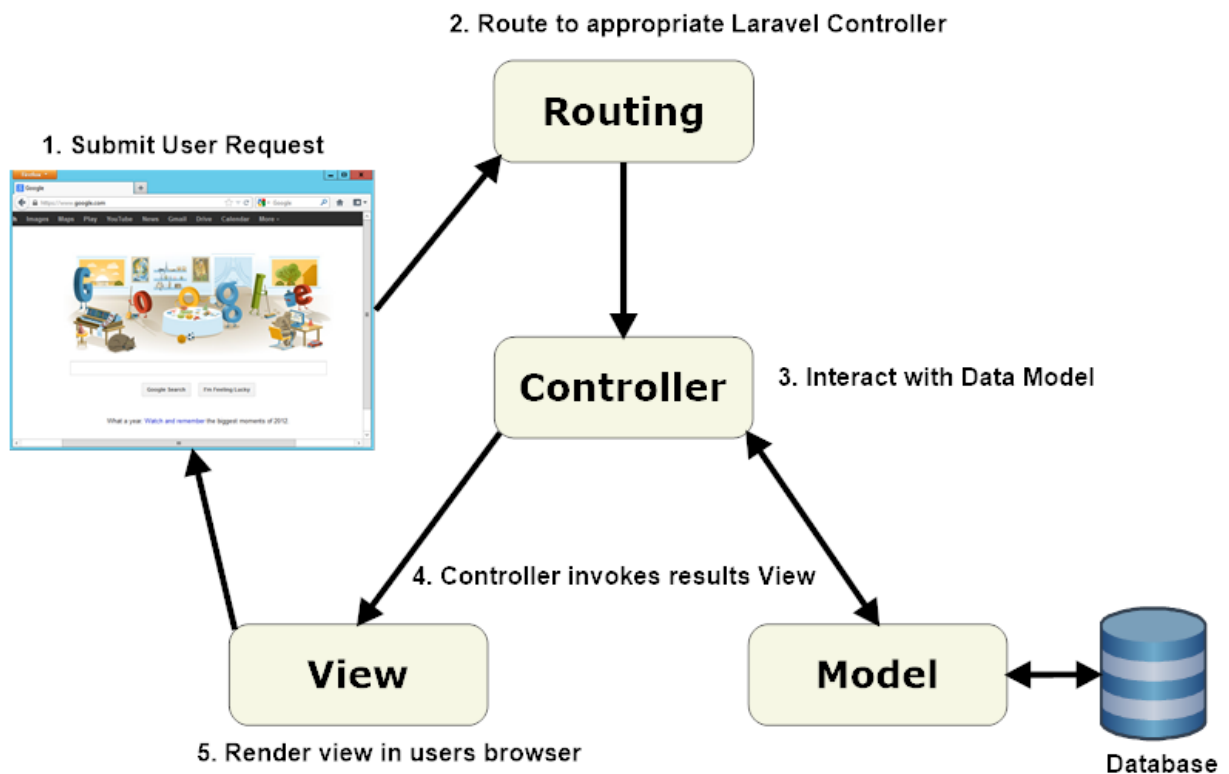


Рисунок 4 – MVC в Laravel

## 2.2.2 Ключевые особенности Laravel

В основе архитектуры Laravel, лежат следующие возможности:

- 1) *Пакеты* - позволяют создавать и подключать модули в формате *Composer*[2] к приложению на Laravel. Многие дополнительные возможности уже доступны в виде таких модулей.
- 2) *Eloquent ORM* — реализация шаблона проектирования *ActiveRecord* на PHP. Позволяет строго определить отношения между объектами базы данных.
- 3) *Логика приложения* — часть разрабатываемого приложения, объявленная либо при помощи контроллеров, либо маршрутов (функций-замыканий).
- 4) *Обратная маршрутизация* связывает между собой генерируемые приложением ссылки и маршруты, позволяя изменять последние с автоматическим обновлением связанных ссылок. При создании ссылок с помощью именованных маршрутов Laravel автоматически генерирует конечные URL.

- 5) *REST-контроллеры* — дополнительный слой для разделения логики обработки GET- и POST-запросов HTTP.
- 6) *Автозагрузка классов* — механизм автоматической загрузки классов PHP без необходимости подключать файлы их определений в include. Загрузка по требованию предотвращает загрузку ненужных компонентов; загружаются только те из них, которые действительно используются.
- 7) *Составители представлений* (англ. view composers) — блоки кода, которые выполняются при генерации представления (шаблона).
- 8) *Инверсия управления* (англ. Inversion of Control) — позволяет получать экземпляры объектов по принципу обратного управления. Также может использоваться для создания и получения объектов-одиночек (англ. singleton).
- 9) *Миграции* — система управления версиями для баз данных. Позволяет связывать изменения в коде приложения с изменениями, которые требуется внести в структуру БД, что упрощает развёртывание и обновление приложения.
- 10) *Модульное тестирование (юнит-тесты)* — играет очень большую роль в Laravel, который сам по себе содержит большое число тестов для предотвращения регрессий (ошибок вследствие обновления кода или исправления других ошибок).
- 11) *Страничный вывод* (англ. pagination) — упрощает генерацию страниц, заменяя различные способы решения этой задачи единым механизмом, встроенным в Laravel.

Также в стандартный установщик Laravel встроен модуль регистрации и аутентификации плюс уже сформированные миграции для таблиц пользователей с ролями, что сводит минусы выбора данного фреймворка вместо CMS, практически к нулю.

## 2.3 Проектирование базы данных

База данных является важнейшим элементом любой информационной системы. Laravel на данный момент поддерживает работу со следующими СУБД:

- MySQL
- Postgres
- SQLite
- SQL Server

Для реализации всех необходимых возможностей нашего веб-сервиса вполне будет достаточно свободной реляционной СУБД MySQL.

В ходе изучения предметной области были выделены следующие основные сущности:

- Пользователь
- Отзыв
- Список контактов
- Контакт
- Категория
- Статья
- Контент для соц сетей

При проектировании базы данных предметной области, была получена ER-модель, изображенная на рисунке 5

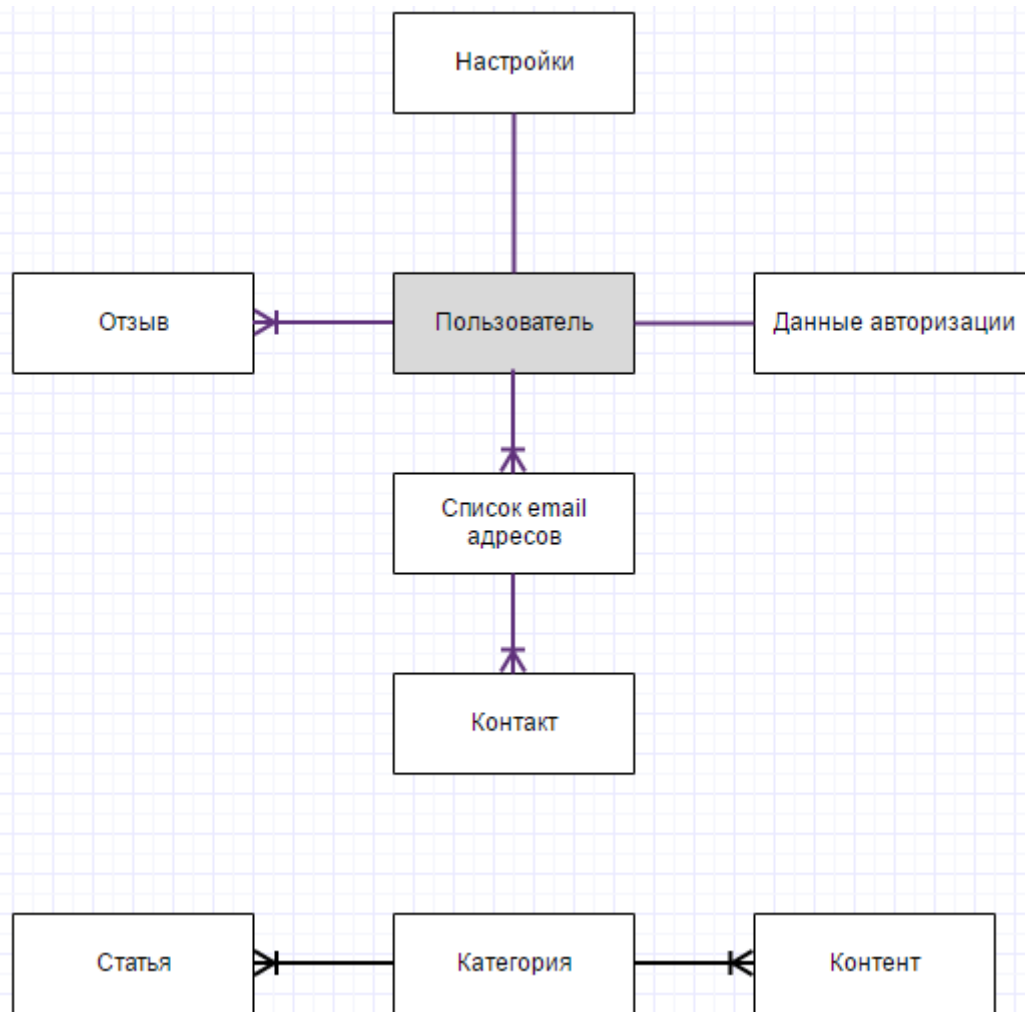


Рисунок 5 – ER-модель базы данных предметной области в нотации Баркера

При проектировании выяснилось, что в таблице Пользователь необходимо будет хранить много информации, будь то основные характеристики врача – имя, место проведения практики(адрес работы), так и дополнительные, такие как токены аутентификации в социальных сетях, данные доступа к личному блогу, информация о аккаунтах на специализированных сервисах отзывов и т.д. Было принято решение выделить две дополнительные сущности для уменьшения количества атрибутов в основной таблице. Были выделены две дополнительные сущности:

- *Данные авторизации*, где будет храниться информация об авторизации в соц. сетях. Пример атрибутов: *facebook\_token*, *fb\_account\_id*, *fb\_profile\_image*.



- *Настройки*, где будет храниться информация о настройках для авторизации на личном блоге, аккаунтах на специализированных сервисах отзывов. Пример атрибутов: *yelp\_id*, *zocdoc\_id*, *google\_place\_id*.

Краткая характеристика остальных сущностей:

- *Пользователь* – содержит основную информацию о пользователе веб-сервиса. Пользователем выступает частный мед. работник или администратор сайта. Пример атрибутов: *role*, *first\_name*, *email*.
- *Отзыв* – информация об отзыве о конкретном враче(пользователе). Пример атрибутов: *title*, *text*, *rating*.
- *Контакт* – информация о пациенте, представленная в виде его email-адреса, имени, фамилии.
- *Список email-адресов* – predetermined списки с контактами пациентов. Используются для массовой рассылки полезного контента или важной информации клиентам.
- *Статья* – администрацией веб-сервиса подразумевается написание\поиск статей на различную тематику. Такие статьи врач сможет использовать в своих целях, чтобы поделиться или со своими пациентами или как-то прокомментировать их, используя блог.
- *Контент* – имеется в виду контент для соц. сетей. Суть использования походит на *Статью*, но это более универсальная сущность. То есть, например, можно и просто картинкой поделиться с медицинской тематикой. Или просто цитатой известного доктора. Контент также предоставляется администрацией сервиса.
- *Категория* – Используется для разделения статей, генерируемых администрацией сервиса, на категории. Здесь под категорией имеется в виду сфера деятельности докторов. К примеру, статьи для стоматологов совершенно не подходят для терапевтов.

## 2.4 Проектирование архитектуры приложения

Чтобы понять, какие классы нам нужны и их взаимодействие между собой, необходимо специфицировать варианты использования. На рисунке 6 и рисунке 7 изображена диаграмма вариантов использования для пользователя и администратора соответственно. Диаграммы представлены сразу на английском языке, т.к. выглядит более семантически-значимо.

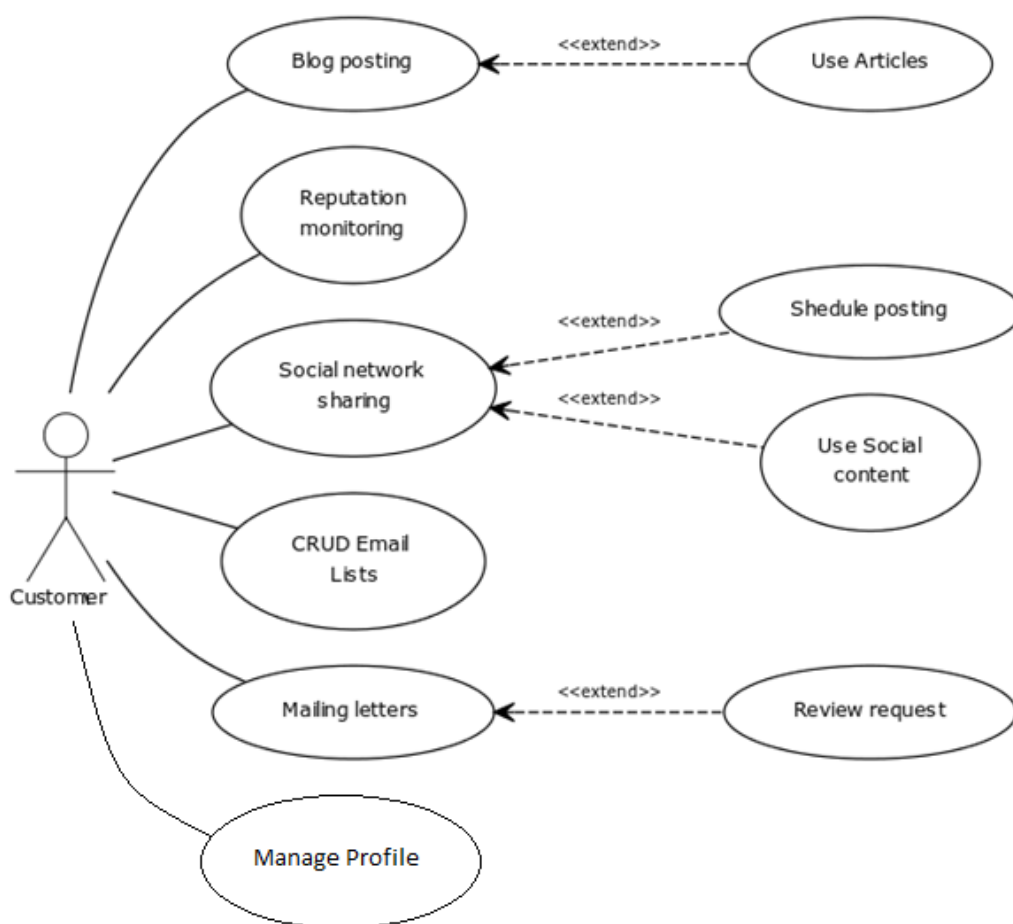
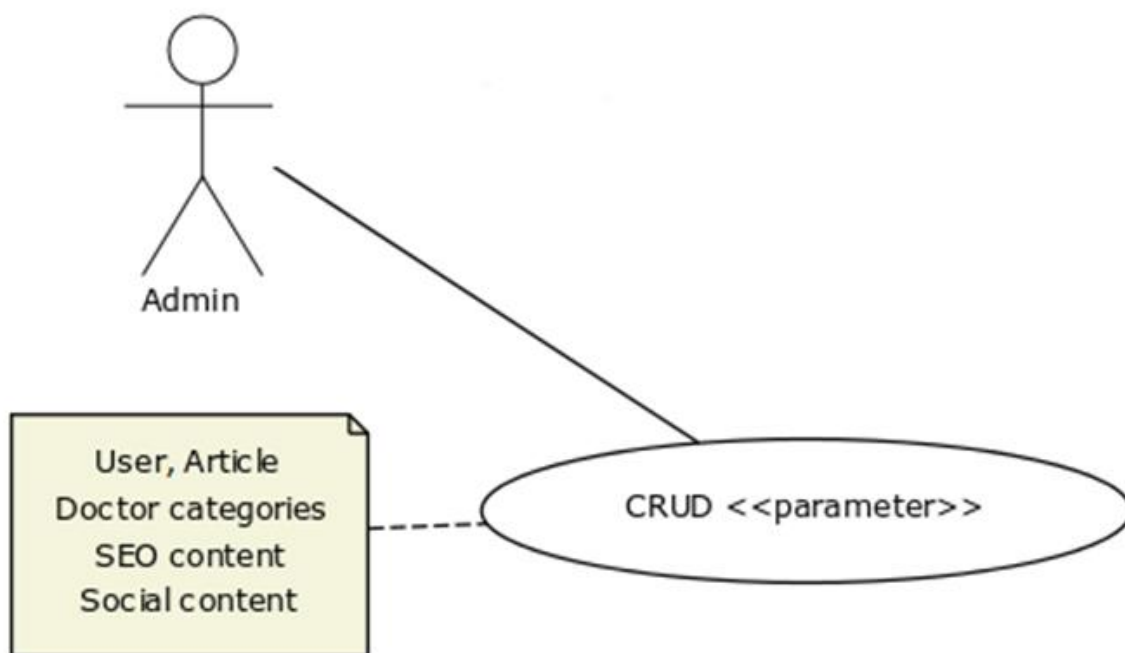


Рисунок 6 – Диаграмма вариантов использования для роли “Пользователь”



*Рисунок 7 – Диаграмма вариантов использования для роли «Администратор»*

Некоторые пояснения:

- Customer – User - пользователь, аутентифицированный участник системы, имеет следующие возможности:
  - Размещать статьи\контент в интегрированный блог, при этом, может использовать статьи, написанные или найденные заранее администраторами веб-сервиса.
  - Следить за своей репутацией у пациентов, просматривая и анализируя отзывы, собранные в специальной вкладке.
  - Размещать контент в социальные сети, при этом, также, может использовать контент, найденный или сгенерированный администраторами сервиса, в том числе, с возможностью установления расписания или отложенного постинга.
  - Создавать, просматривать, изменять или удалять предустановленные списки email-адресов(контактов)

- Устраивать массовые email-рассылки. Опционально использовать встроенный шаблон для предложения пациентам оставить отзыв о данном враче после приема.
- Admin – Пользователь с ролью администратора сервера. Может управлять всем контентом сервиса.

На рисунке 8 отображена полученная диаграмма классов.

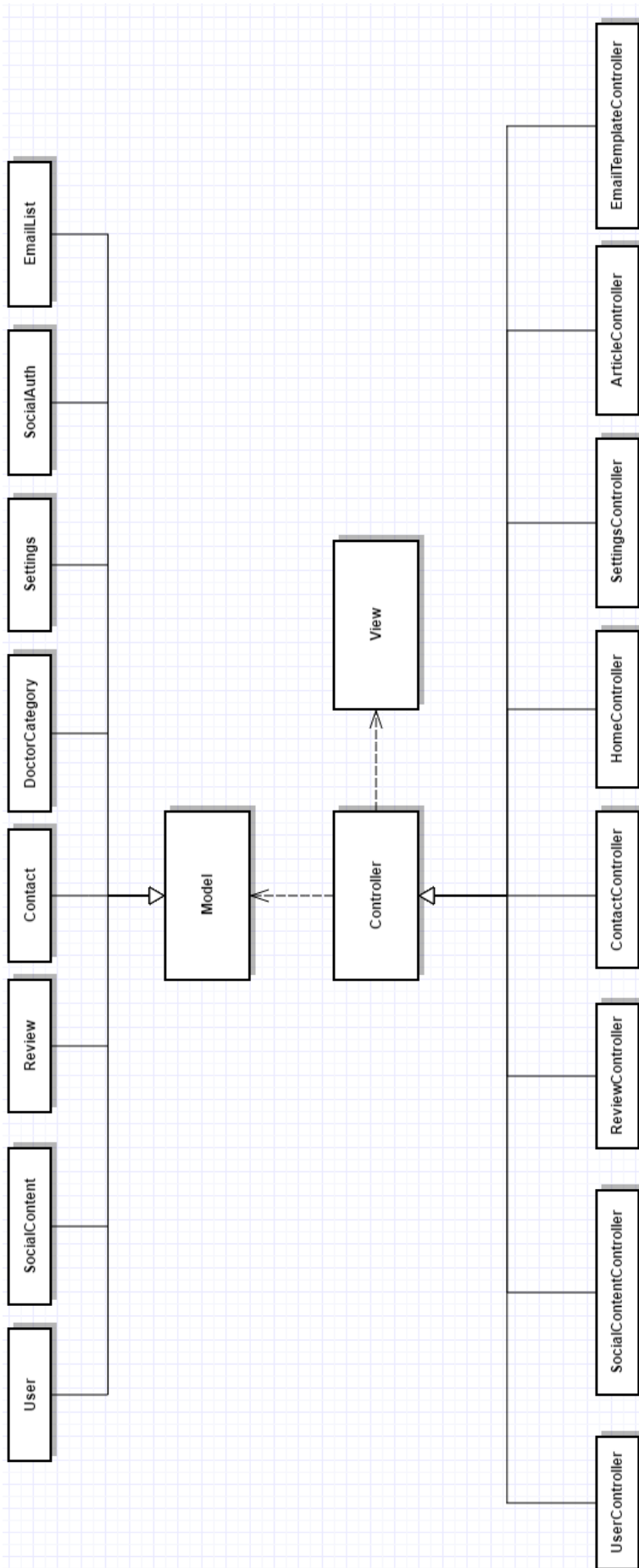


Рисунок 8 – общая диаграмма классов.

### 3. Реализация

При обработке запросов, Laravel опирается на систему маршрутизации, которая сопоставляет все входящие запросы с определенными в системе маршрутами, которые указывают какой контроллер и метод должен обработать данный запрос.

В Laravel мощная и гибкая система маршрутизации. Все маршруты определяются в файле `routes.php`. Например, сгенерированные маршруты для контроллера `UserController` можно увидеть на таблице 1:

Verb	Path	Action	Route Name
GET	<code>/user</code>	index	user.index
GET	<code>/user/create</code>	create	user.create
POST	<code>/user</code>	store	user.store
GET	<code>/user/{id}</code>	show	user.show
GET	<code>/user/{id}/edit</code>	edit	user.edit
PUT/PATCH	<code>/user/{id}</code>	update	user.update
DELETE	<code>/user/{id}</code>	destroy	user.destroy

Таблица 1 – Маршруты, создаваемые Laravel для определенного контроллера

Покажем обработку запроса на примере варианта использования “Manage profile”. Когда пользователь хочет просмотреть свой профиль – он обращается через ссылку на адрес `/user/id`. По заданному маршруту, запрос перенаправляется на метод `show` контроллера `UserController`, который, в свою очередь, делает запрос к классу модели `User`, получает экземпляр класса `Eloquent[2]`, и вызывает метод отрисовки представления (`view`), которое и увидит пользователь.

Для наглядности, покажем диаграмму последовательностей на рисунке

9.

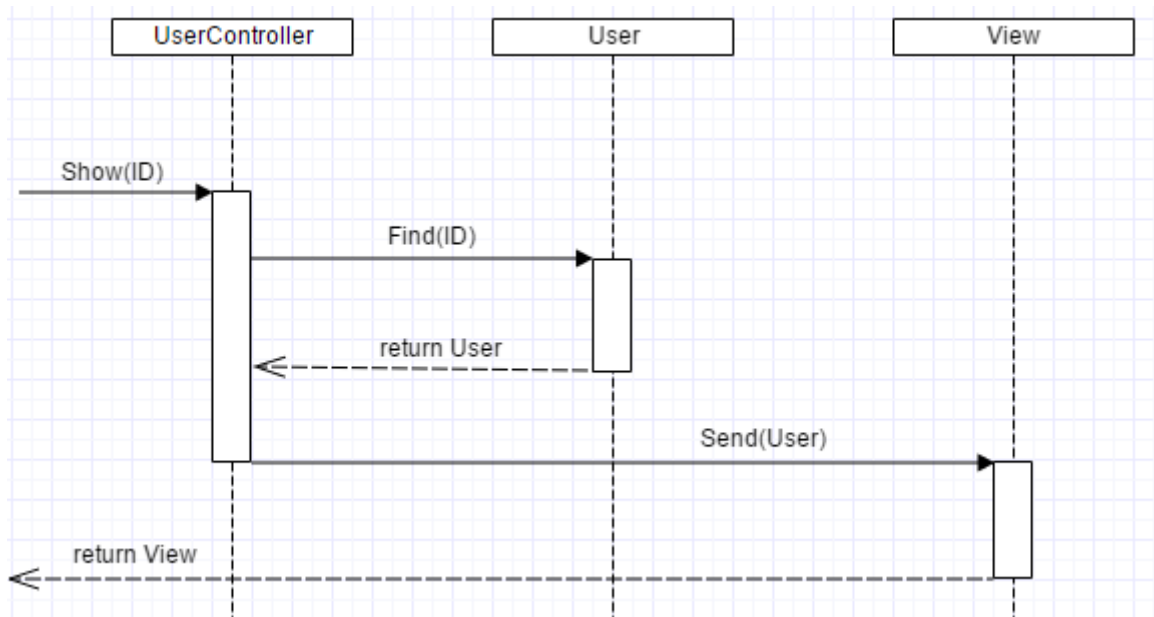


Рисунок 9 – Диаграмма последовательностей просмотра профиля

Подобным образом реализуются все маршруты в веб-сервисе. Ниже, на рисунке 10 представлены классы с некоторыми полями и методами, используемые для отображения домашней страницы сервиса с данными о количестве текущих подписчиков в различных социальных сетях, а также количестве различных типов отзывов со специализированных сервисов отзывов. На рисунке 11 представлен внешний вид этой страницы.

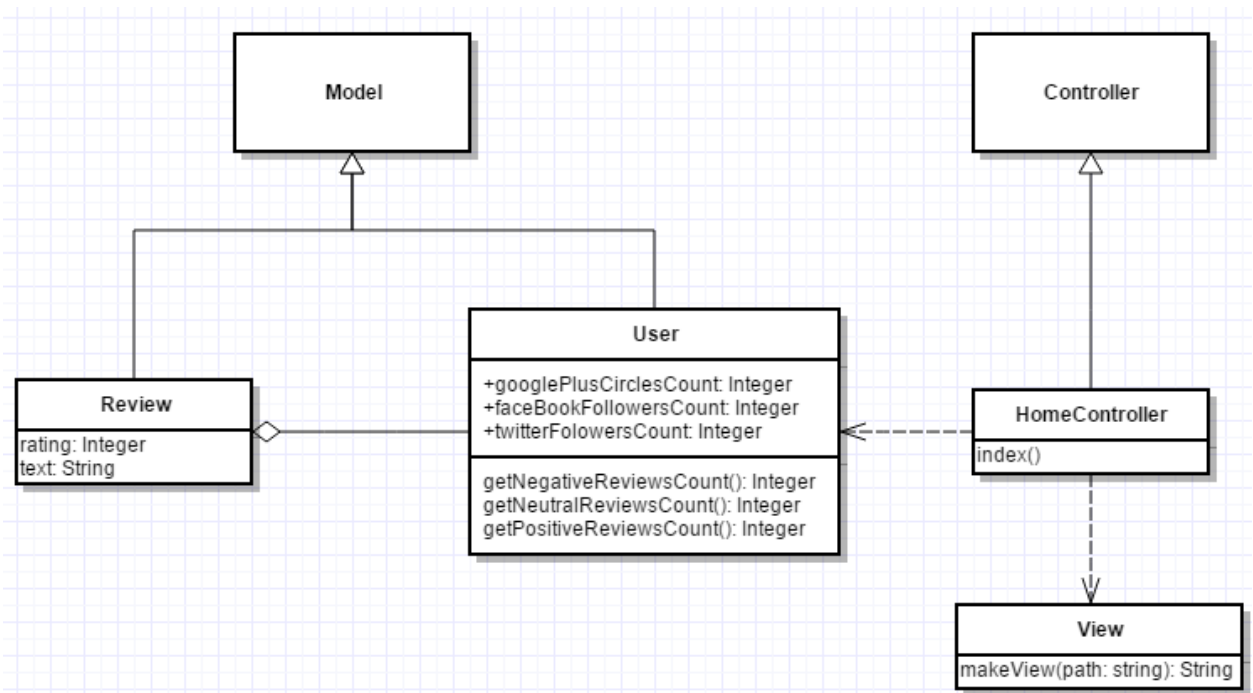


Рисунок 10 – Диаграмма классов отображения домашней страницы

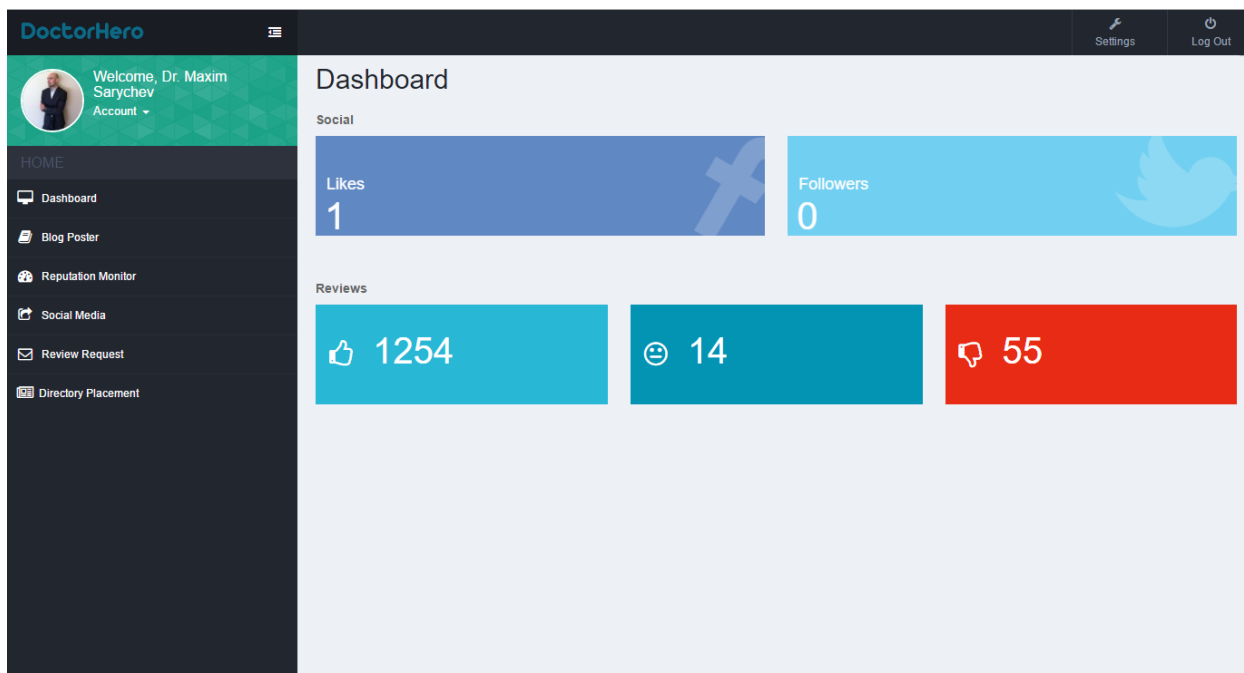


Рисунок 11 – Внешний вид домашней страницы

Продemonстрируем реализацию части необходимого функционала. Для примера будет взят модуль размещения контента в социальных сетях. Пользователь может вставить в специальное текстовое поле ссылку на статью/новость. В частности, будет рассмотрен метод получения мета-данных статьи/новости по ссылке.

За обработку логики модуля отвечает контроллер `SocialController`. Данные получаются через Open Graph тэги[7] на странице. Используется класс `ContentExtractor` и внешняя библиотека `OpenGraph`, где реализован удобный парсинг мета-тэгов.

Протокол `Open Graph` позволяет любой веб-странице стать полноценным объектом в социальных сетях. Например, протокол используется на Facebook, чтобы разрешить любой веб-странице иметь такую же функциональность, как и любой другой объект на Facebook. Хотя многие различные технологии и схемы существуют и могут быть объединены вместе, нет ни одной технологии, которая предоставляет достаточно информации, чтобы красочно представлять какие-либо веб-страницы внутри социальных сетей. Протокол `Open Graph` строится на этих существующих технологиях и



предоставляет разработчикам возможность реализовать все это в одном протоколе. Простота разработки является ключевой задачей протокола Open Graph, сообщили многие, на основе технических проектных решений.

Основные Метаданные:

- og:title - Название объекта, как он должен отображаться в графе
- og:type - Типы объектов
- og:image - URL-адрес изображения, который должен определить объект в графе для изображения.
- og:url - Канонический URL-адрес объекта, который будет использоваться в качестве его постоянного ID.

На рисунке 12 изображена диаграмма классов, использующихся для реализации данного функционала. На рисунке 13 – диаграмма последовательностей. Рисунок 14 – итоговый пользовательский интерфейс для страницы размещения контента в социальные сети.

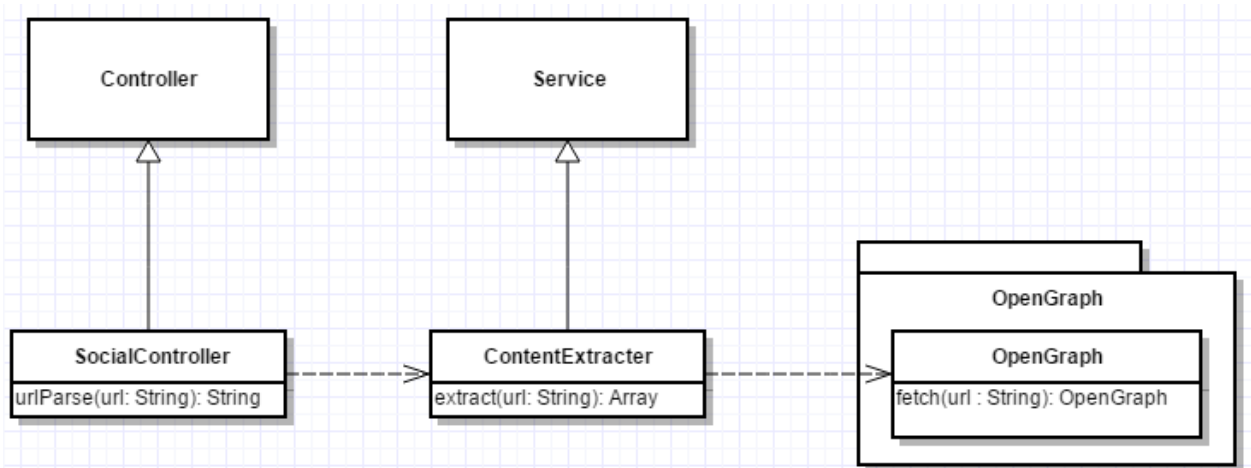


Рисунок 12 – Диаграмма классов получения мета-данных страницы

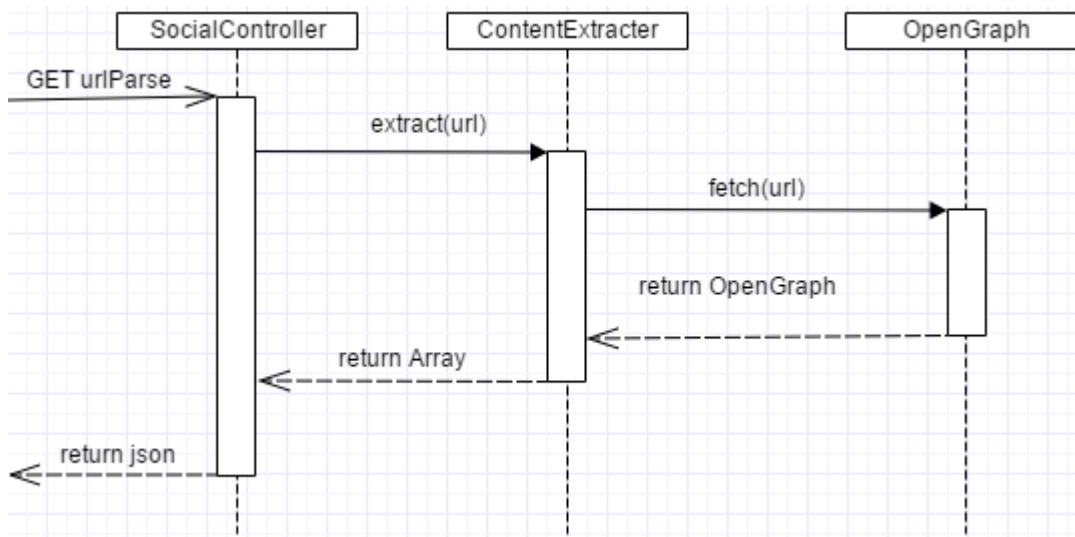


Рисунок 13 – Диаграмма последовательностей получения мета-данных страницы по ссылке

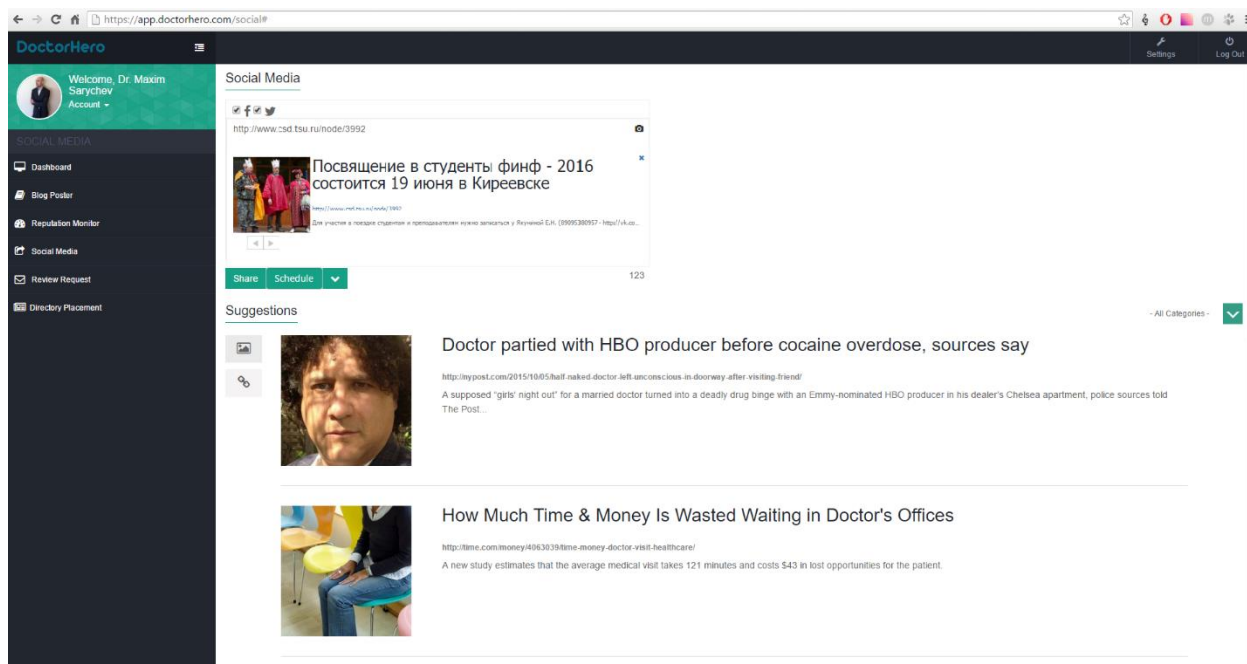


Рисунок 14 – Внешний вид страницы размещения контента в соц. сети

Также рассмотрим реализацию модуля размещения контента в удаленный блог.

WordPress имеет два механизма публикации постов. Первый – web интерфейс (в данном случае он нас не интересует). И второй – с помощью XML-RPC. RPC расшифровывается как Remote Procedure Call (удаленный вызов процедур/функций). В принципе эта технология очень похожа на обычные запросы. От клиента к серверу передается строка с данными. Но XML-RPC требует, чтобы эта строка имела специальный формат. Используя

XML-RPC можно опубликовать пост с помощью всего одного запроса, а специальные библиотеки позволяют не углубляться в тонкости стандарта. Для наших целей была найдена библиотека IXR - The Incutio XML-RPC Library.

Алгоритм удаленного размещения статьи в общем случае, примерно выглядит так:

1. Загружается библиотека XML-RPC и указывается адрес блога, для которого будет идти публикация (отмечу, что если необходимо опубликовать пост в несколько блогов, то метод вызывается для каждого URL-а отдельно);
2. Формируется XML-RPC запрос.
3. В завершении сформированный XML запрос отправляется на сервер, и в зависимости от ответа принимается решение о результатах публикации.

Диаграмма используемых классов изображена на рисунке 15. Диаграмма последовательностей – на рисунке 16.

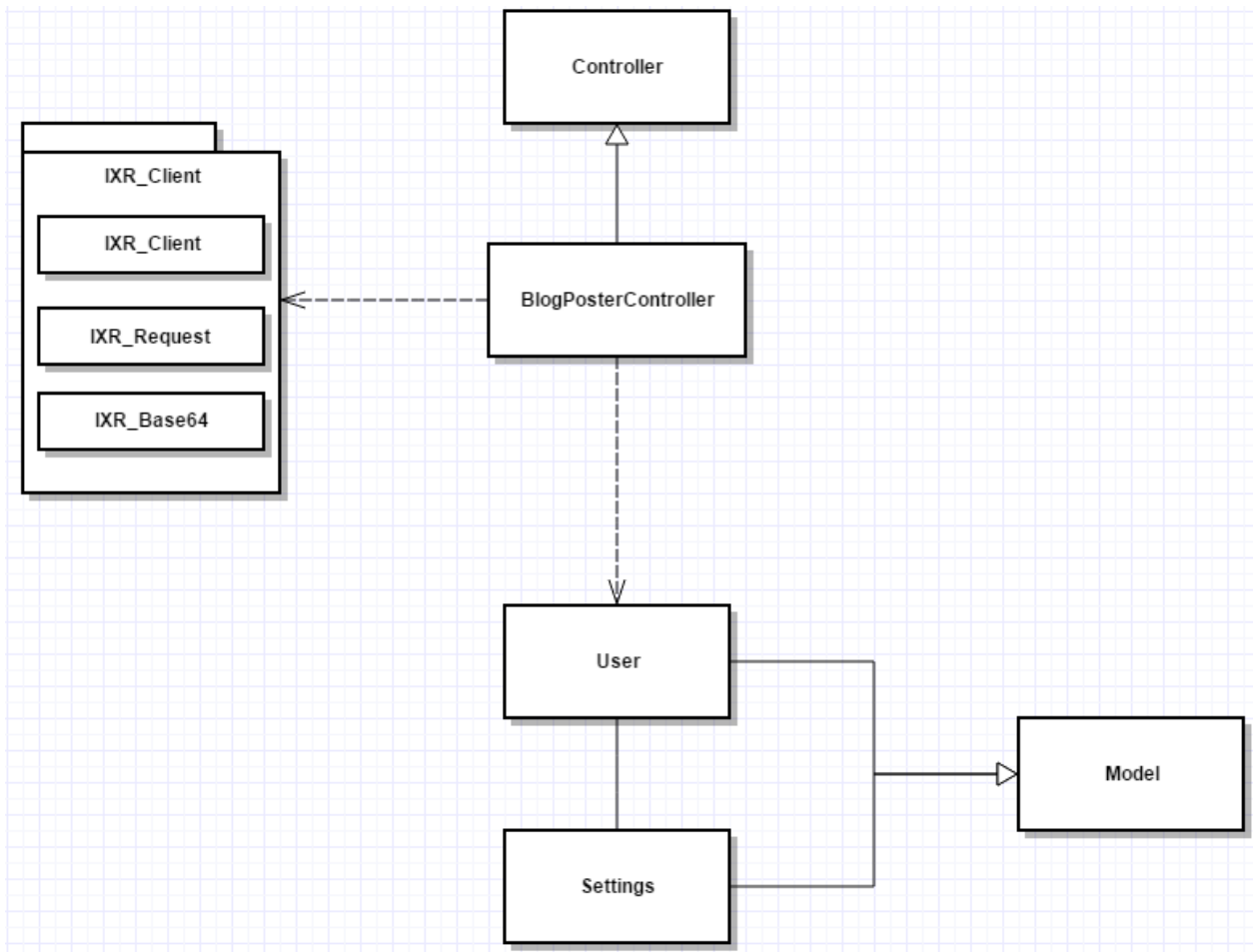


Рисунок 15 – Диаграмма классов модуля размещения контента в удаленный блог

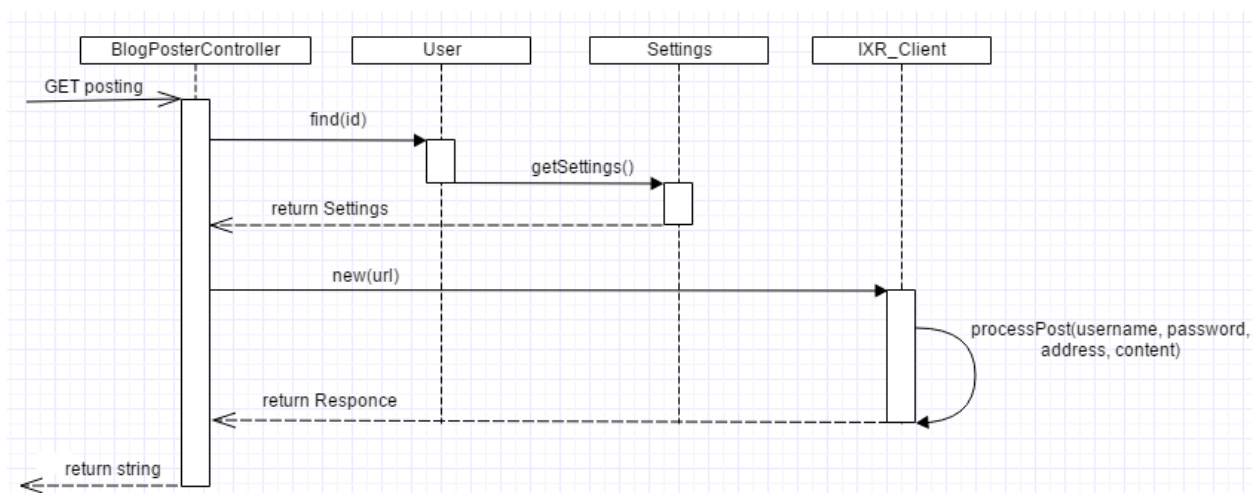


Рисунок 16 – Диаграмма последовательностей модуля размещения контента в удаленный блог

Результаты реализации остальных частей веб-сервиса, можно посмотреть в приложении 1.

## **Заключение**

Целью данной работы являлась разработка веб-сервиса, который бы предоставлял частным работникам медицинской сферы необходимый функционал для управления своими инструментами для интернет-маркетинга из одного места.

В ходе выполнения работы был выполнен анализ существующих систем, выявлены требования к разрабатываемому приложению, спроектирована архитектура и реализован веб-сервис для интернет-маркетинга частных работников медицины.

Таким образом, все поставленные задачи выполнены и цель работы достигнута.

Разработанный веб-сервис введен в эксплуатацию на поддомене сайта [www.doctorhero.com](http://www.doctorhero.com) по адресу [app.doctorhero.com](http://app.doctorhero.com).

## Список использованной литературы

1. Гамма Э. Приемы объектно-ориентированного проектирования. Паттерны проектирования / Э. Гамма [и др.]. – СПб.: Питер, 2001. - 368 с.
2. Документация по работе с php-фреймворком Laravel [Электронный ресурс] // URL: <https://laravel.com/docs/5.2> (дата обращения: 08.04.2016).
3. Клиентская библиотека работы с URL [Электронный ресурс] // URL: <http://php.net/manual/ru/book.curl.php> (дата обращения: 14.04.2016).
4. Справочник по CSS [Электронный ресурс] // URL: <http://htmlbook.ru/css> (дата обращения: 27.03.2016).
5. Справочник по PHP [Электронный ресурс] // URL: <http://www.php.su> (дата обращения: 20.04.2016).
6. Удаленная публикация в WordPress через XML-RPC [Электронный ресурс] // URL: <http://azzrael.ru/wordpress-xml-rpc-metaweblog-api> (дата обращения: 14.04.2016).
7. The Open Graph protocol [Электронный ресурс] // URL: <http://ogp.me/> (дата обращения: 14.02.2016).

## Приложение 1. Руководство пользователя

На рисунке 11 изображен главный экран веб-сервиса. Навигационное меню слева отображает ссылки на страницы с инструментами интернет-маркетинга. Над ним изображение профиля и ссылка на страницу профиля. В правом верхнем углу ссылка на панель настроек аккаунта и кнопка Выхода.

Продемонстрируем некоторые варианты использования от лица пользователя:

**Manage profile** – Рисунок 17 - секция настроек разбита на вкладки. В каждой вкладке настройки сгруппированы по типу.

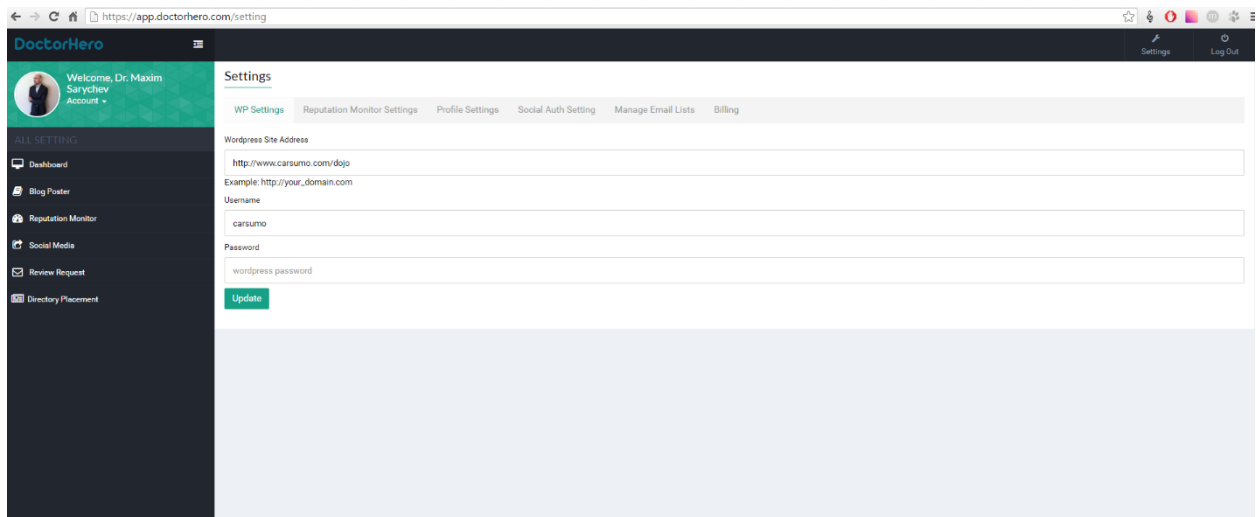


Рисунок 17 – Настройки профиля

Вкладка секции настроек - Авторизация в соц сетях. Рисунок 18

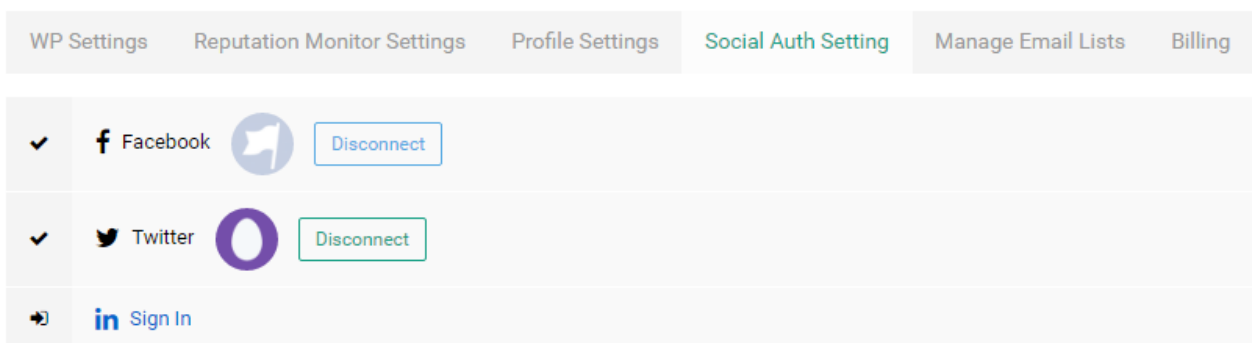
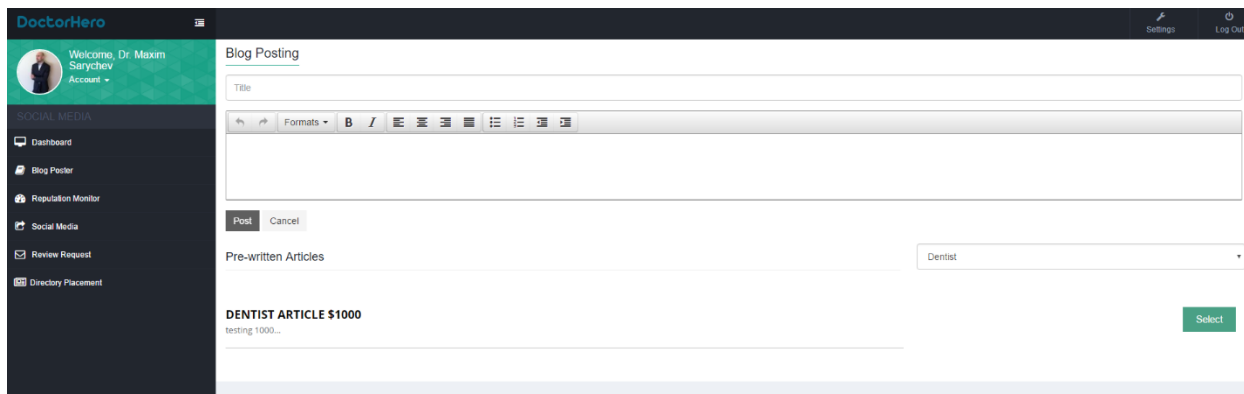


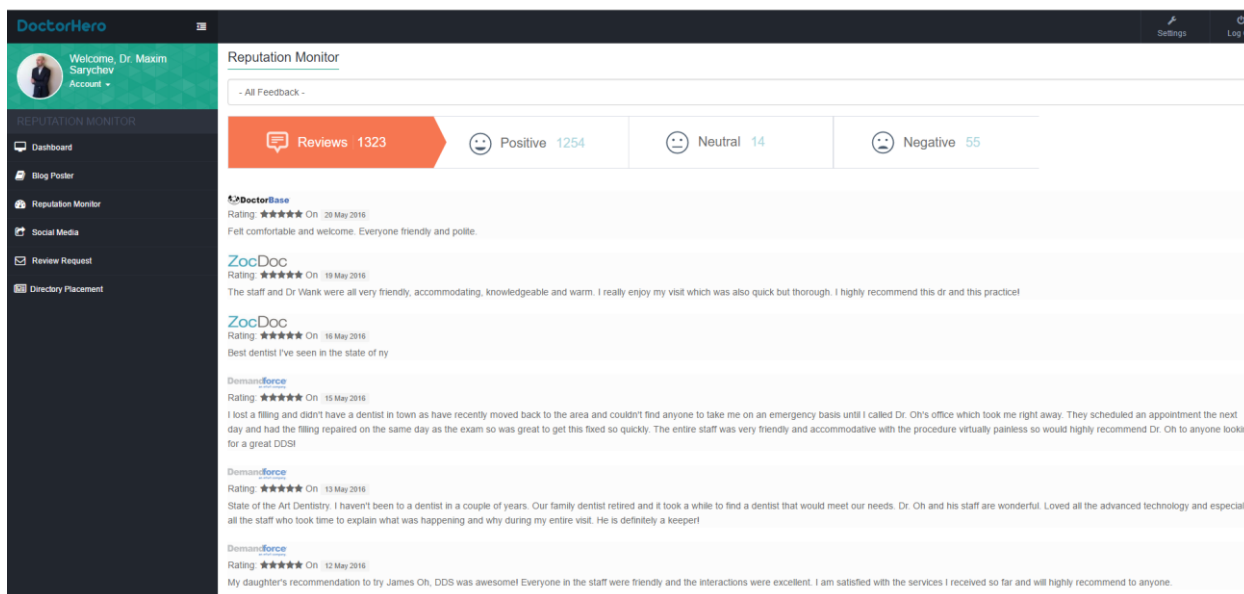
Рисунок 18 – Авторизация в социальных сетях

**Blogposting** – снизу секция готовых статей. На рисунке 19 - выпадающий список позволяет отфильтровать статьи по нужной категории. При нажатии на кнопку *Select* – статья вставится в поле для редактирования. При нажатии на кнопку *Post* – статья разместится на блоге пользователя.



*Рисунок 19 – Размещение контента в блог*

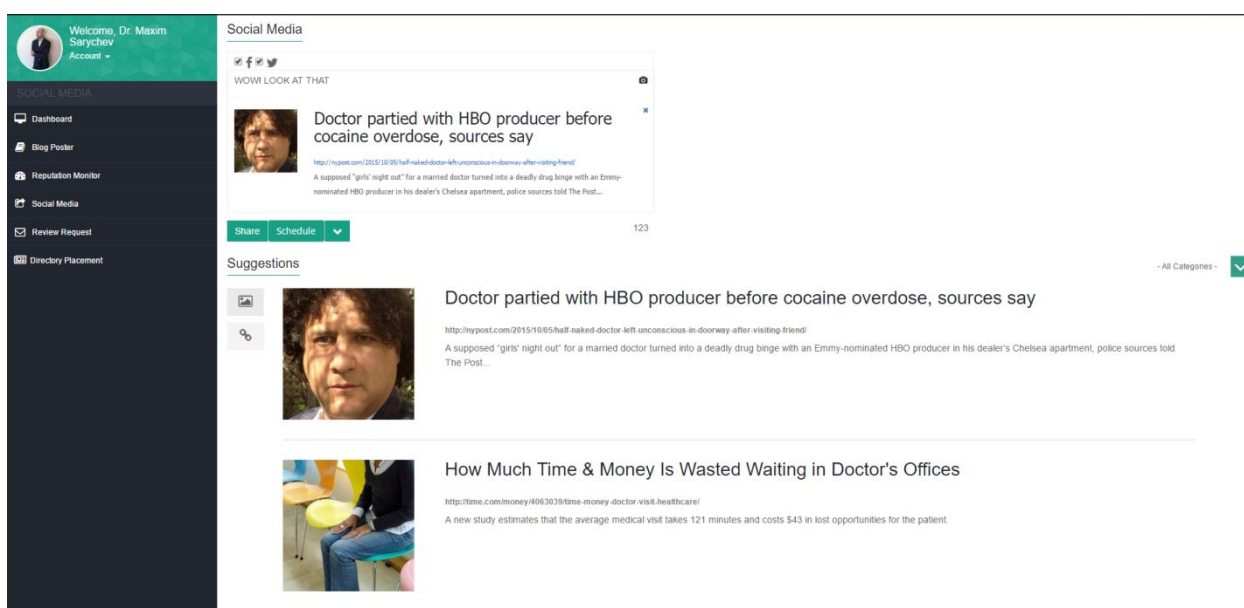
**Reputation monitor** – рисунок 20 – страница со всеми отзывами пользователя, собранными с разных сервисов отзывов. Можно увидеть подсчет количества типов отзывов. Выпадающий список сверху позволяет отфильтровать отзывы по конкретному сервису отзывов.



*Рисунок 20 – Просмотр отзывов*

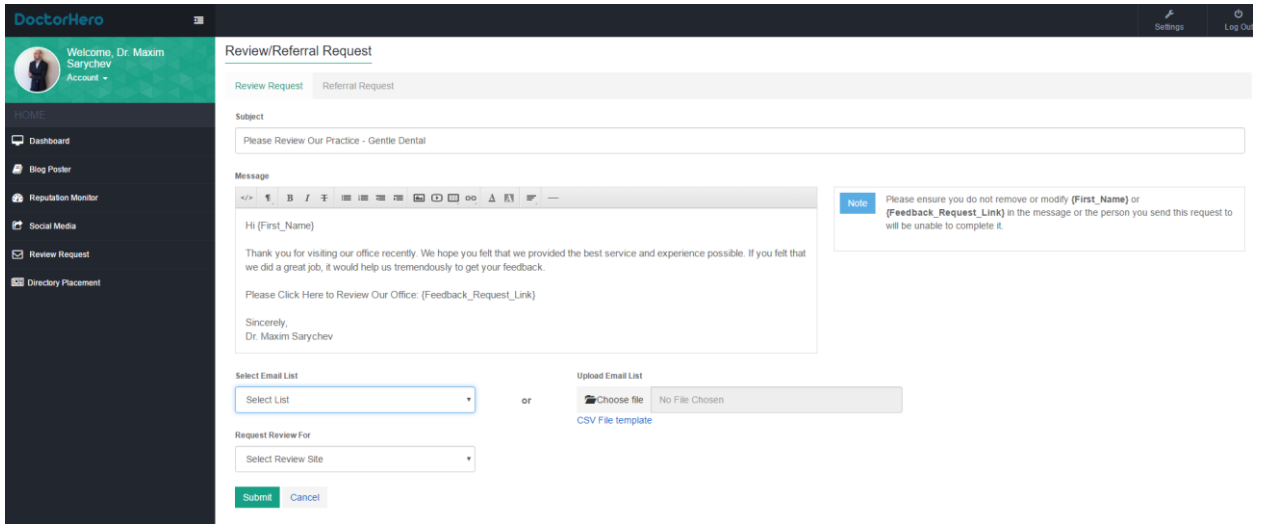


**Social media** – на рисунке 21 используется контент, предоставленный администраторами сервисами. Достаточно навести на изображение с нужным контентом. Появится кнопка, позволяющая разместить контент в поле для отправки в соц.сети. Галочки соответственно определяют, в какую именно соц. сеть необходимо выполнить размещение контента. Также можно вставить свою ссылку в поле ввода текста. Из страницы, расположенной по адресу ссылки будет выделена полезная информация и отобразится предпросмотр, как размещенная ссылка будет выглядеть в социальных сетях, куда пользователь собирается разместить контент.



*Рисунок 21 – Размещение контента в социальные сети*

**Mailing letters** – на рисунке 22 изображен процесс отправки писем. Она может осуществляться как одному пациенту, так и нескольким. Имеется возможность выбора из преднастроенных списков адресов. Также, список может загружаться из .csv файла.



*Рисунок 22 – Рассылка писем пациентам*