МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего профессионального образования

**"Южно-Уральский государственный университет"**

**(национальный исследовательский университет)**

**Факультет Вычислительной математики и информатики**

**Кафедра системного программирования**

**Разработка веб-приложения для мониторинга позиций сайтов по запросам в поисковой системе Яндекс**

КУРСОВАЯ РАБОТА

по дисциплине «Программная инженерия»

ЮУрГУ – 010300.62.2014.11-027-1909.КР

|  |  |
| --- | --- |
| Нормоконтролер,кандидат физико-математических наук, доцент каф. СП  \_\_\_\_\_\_\_\_\_\_\_\_\_\_Г.И. Радченко  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2014 г. | Научный руководитель:  кандидат физико-математических наук, доцент каф. СП  \_\_\_\_\_\_\_\_\_\_Г.И. Радченко  Автор работы:  студент группы ВМИ-311  \_\_\_\_\_\_\_\_\_\_Е.А. Неповинных  Работа защищена  с оценкой: \_\_\_\_\_\_\_\_\_\_\_  “\_\_\_”\_\_\_\_\_\_\_\_\_\_\_ 2014 г. |

Челябинск 2014

Оглавление

[Оглавление 2](#_Toc389594059)

[Словарь 3](#_Toc389594060)

[Введение 4](#_Toc389594061)

[Структура и объем работы 5](#_Toc389594062)

[1. Обзор литературы 6](#_Toc389594063)

[2. Требования к системе мониторинга позиций сайтов 9](#_Toc389594064)

*[2.1.](#_Toc389594065)**[Система мониторинга позиций сайтов](#_Toc389594065)* [9](#_Toc389594065)

*[2.2.](#_Toc389594066)**[Варианты использования системы мониторинга позиций сайтов](#_Toc389594066)* [9](#_Toc389594066)

*[2.3.](#_Toc389594067)**[Нефункциональные требования к системе мониторинга позиций сайтов](#_Toc389594067)* [12](#_Toc389594067)

*[2.4.](#_Toc389594068)**[Вывод](#_Toc389594068)* [13](#_Toc389594068)

[3. Архитектура системы мониторинга позиций сайтов 14](#_Toc389594069)

*[3.1.](#_Toc389594070)**[Архитектура MVC](#_Toc389594070)* [14](#_Toc389594070)

*[3.2.](#_Toc389594071)**[Интерфейс веб-приложения](#_Toc389594071)* [18](#_Toc389594071)

*[3.3.](#_Toc389594072)**[Вывод](#_Toc389594072)* [19](#_Toc389594072)

[4. Реализация системы мониторинга позиций сайтов 21](#_Toc389594073)

*[4.1.](#_Toc389594074)**[Структура базы данных мониторинга](#_Toc389594074)* [21](#_Toc389594074)

*[4.2.](#_Toc389594075)**[Страница аккаунта](#_Toc389594075)* [22](#_Toc389594075)

*[4.3.](#_Toc389594076)**[Отображение позиций](#_Toc389594076)* [23](#_Toc389594076)

*[4.4.](#_Toc389594077)**[Страница редактирования ресурса](#_Toc389594077)* [24](#_Toc389594077)

*[4.5.](#_Toc389594078)**[Организация безопасности приложения](#_Toc389594078)* [26](#_Toc389594078)

*[4.6.](#_Toc389594079)**[Вывод](#_Toc389594079)* [29](#_Toc389594079)

[5. Тестирование системы мониторинга позиций сайтов в поисковой системе Яндекс 30](#_Toc389594080)

[6. Заключение 33](#_Toc389594081)

[Приложение 1. Реализация прецедентов 35](#_Toc389594082)

# Словарь

*Пользователь* – конкретное лицо или представитель организации, использующий систему для того, чтобы выполнять конкретные задачи или функции.

*Организация* — группа людей, деятельность которых сознательно координируется для достижения общих целей.

*Ресурс* — веб-сайт, для которого осуществляется мониторинг позиций по заданному набору поисковых запросов.

*Аккаунт* — учётная запись, содержащая сведения об организации в системе.

*Мониторинг* —анализ позиции ресурса в результатах поиска.

*Позиция ресурса* в результатах поиска (далее: позиция) — порядковый номер ссылки на страницу определенного ресурса в результатах поиска по определенному поисковому запросу.

*Поисковый запрос (далее: запрос)* – строка для осуществления поиска с помощью поисковой системы.

*Регион мониторинга* – это параметр поискового запроса, осуществляющий территориальную специализацию результатов поиска.

*Результаты поиска* — упорядоченный список ссылок на различные ресурсы, которые пользователь видит в результате запроса к поисковой системе Яндекс.

# Введение

Продвижение интернет-сайта в поисковых системах позволяет значительно увеличить количество просмотров и качество активных пользователей сайта. На сегодняшний день продвижение сайтов в русско-язычном сегменте сети Интернет осуществляется в 5-ти наиболее популярных поисковых системах [4]: Яндекс, Google, поиск Mail.ru, Rambler, Bing. Существует целый ряд различных способов продвижения интернет-сайта в поисковых системах: генерация и размещение на сайте уникального и релевантного контента, продвижение сайта в каталогах и социальных сетях, использование определенных ключевых слов, покупка внешних ссылок на сайт и др. Для скорейшего достижения высокой позиции сайта и привлечения большего числа клиентов, необходимо постоянно отслеживать положение продвигаемого веб-ресурса в поисковых системах. Для решения этой задачи существуют системы мониторинга [5]. Система мониторинга отслеживает и позволяет сохранять для дальнейшего анализа позиции веб-сайта в поисковых системах за определённый период времени. В настоящий момент наиболее развиты системы мониторинга в виде веб-приложений. Их популярность объясняется, главным образом, тем, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами.

Актуальность данной темы обусловлена необходимостью постоянного анализа позиций коммерческих сайтов в поисковой выдаче для увеличения их посещаемости в процессе поисковой оптимизации (Search Engine Optimization, SEO) сайта. Оперативное определение неэффективных запросов, по которым позиции сайта долгое время остаются низкими, позволяет скорректировать стратегию и оптимизировать расходы продвижения сайта.

*Целью данной работы* является разработка веб-приложения для мониторинга позиций сайтов по запросам в поисковой системе Яндекс.

Для достижения указанной цели необходимо решить следующие задачи:

1. изучить особенности существующих систем мониторинга позиций сайтов в поисковых системах;
2. изучить особенности работы с платформами Django и Twitter Bootstrap;
3. изучить существующие подходы разработки систем мониторинга позиций сайтов в поисковых системах;
4. спроектировать архитектуру системы мониторинга позиций сайтов в поисковых системах;
5. реализовать и протестировать веб-приложение для мониторинга позиций сайтов по запросам в поисковой системе Яндекс.

## Структура и объем работы

Работа состоит из введения, пяти разделов, заключения, библиографии и приложения. Объем работы составляет 37 страниц, объем библиографии – 9 источников.

В первой главе дается обзор технологий работы поисковых систем. В этом разделе будут рассмотрены алгоритмы ранжирования наиболее популярных поисковых систем и выявлены наиболее перспективные сервисы для мониторинга позиций сайтов.

Вторая глава содержит описание и анализ требований к системе мониторинга позиций сайтов в поисковой системе Яндекс.

В третьей главе представлена архитектура системы мониторинга позиций сайтов в поисковой системе Яндекс. В этом разделе подробно рассмотрена общая архитектура системы, интерфейсы компонентов и детали их реализации.

Четвертая глава посвящена реализации системы мониторинга позиций сайтов в поисковой системе Яндекс с использованием технологий django, AJAX и MVC.

В пятой главе приводятся результаты тестирования системы мониторинга позиций сайтов в поисковой системе Яндекс.

В заключении сделаны выводы о проделанной работе.

Приложение содержит детальную реализацию основных прецедентов.

# Обзор литературы

Для успешного продвижения и мониторинга позиций сайта необходимо изучить и понять алгоритмы работы поисковых систем, в которых планируется продвигать сайт. Поисковая система получает и обрабатывает запрос, сортирует все имеющиеся в своей базе данные о сайтах, к которым может иметь отношение данный запрос, то есть производит ранжирование, и выдает результат. На странице результатов поиска в определенном порядке располагаются [ссылки](http://www.webeffector.ru/wiki/Ссылка) на веб-ресурсы, которые, по мнению поисковой системы, наиболее соответствуют указанному запросу [1, 6].

Основные отличия моделей ранжирования поисковых систем Яндекс и Google заключаются в алгоритмах расчета характеристик веб-ресурса, определяющих рейтинг сайта на странице результатов поиска. Такие алгоритмы являются запатентованными разработками и держатся в строгом секрете. Анализ поисковой выдачи позволяет сделать некоторые предположения о том, на основе каких признаков производится ранжирование результатов поисковой выдачи. Благодаря независимости признаков ранжирования от смыслового наполнения сайта, в рамках одной поисковой системы можно использовать одну и ту же модель ранжирования для запросов из совсем разных областей знаний [2, 3].

Результаты исследований показывают, что более 70% пользователей сети Интернет находят интересующие их ресурсы, используя поисковые системы. SEO оптимизация сайта направлена на повышение ранга сайта в поисковой выдаче, что позволяет увеличить посещаемость сайта. Мероприятия по продвижению сайта нужно проводить параллельно с мониторингом показателей – это дает возможность вовремя принимать важные решения и корректировать [стратегию продвижения](http://searchenginez.ru/strategii-prodvizheniya-sajtov/)[6].

Мониторинг позиций сайта дает возможность осуществлять контроль за рейтингом сайтов в поисковых системах по целому или частичному запросу и решает следующие задачи:

* контроль действий конкурентов;
* своевременная корректировка позиции сайта в соответствии с целевыми запросами;
* изменение текстового блока, находящегося на страницах сайта для улучшения позиции в поисковых машинах.

В настоящее время существует множество различных сервисов, которые позволяют проводить мониторинг позиций сайта в поисковых системах. Наиболее простым в использовании является бесплатный сервис «MegaIndex» ([http://www.megaindex.ru](http://www.megaindex.ru/)). Система «MegaIndex» позволяет создать проекты под каждый из своих Интернет-ресурсов и добавить поисковые запросы, по которым будет производиться мониторинг. На странице запросов можно увидеть, на каких позициях в поисковых системах России, Украины и Белоруссии находится продвигаемый ресурс, количество запросов в месяц. Недостатком данного сервиса является то, что его основная функция – платное размещение ссылок на сторонних ресурсах в целях увеличения показателей продвигаемых ресурсов либо повышения их посещаемости, поэтому отслеживать позиции сайтов там неудобно.

Множество функциональных возможностей предоставляет сервис «AllPositions» ([http://allpositions.ru](http://allpositions.ru/)), включая:

* контроль позиций сайтов в выдаче поисковых систем;
* анализ информации о текущих позициях конкурирующих ресурсов;
* получение отчетов о результатах продвижения в удобном формате, которые максимально подробно отобразят все запрашиваемые данные.

Однако данный сервис платный, и при первом использовании довольно трудно разобраться в настройках [7].

После изучения существующих решений было принято решение реализовать систему мониторинга позиций сайтов в поисковой системе Яндекс в виде отдельного django-приложения с использованием средств Twitter Bootstrap. Django [8] — это высокоуровневая веб-платформа для языка программирования Python, поощряющий быструю разработку и чистый, прагматичный дизайн. Twitter Bootstrap – это HTML, CSS и Javascript платформа для Web-разработки [9].

# Требования к системе мониторинга позиций сайтов

## *Система мониторинга позиций сайтов*

Система мониторинга позиций сайтов – это система, позволяющая отслеживать рейтинг исследуемых сайтов в поисковых системах по выбранным поисковым запросам. Под рейтингом понимается позиция ресурса на странице результатов поиска.

Разрабатываемая система мониторинга позиций сайтов не использует специальных программных интерфейсов для доступа к результатам поиска. Она имитирует действия посетителей поисковых ресурсов и работает только с той информацией, которая представлена на html-страницах.

Приложение работает с наиболее популярной в русскоязычном сегменте сети Интернет поисковой системой Яндекс.

## *Варианты использования системы мониторинга позиций сайтов*

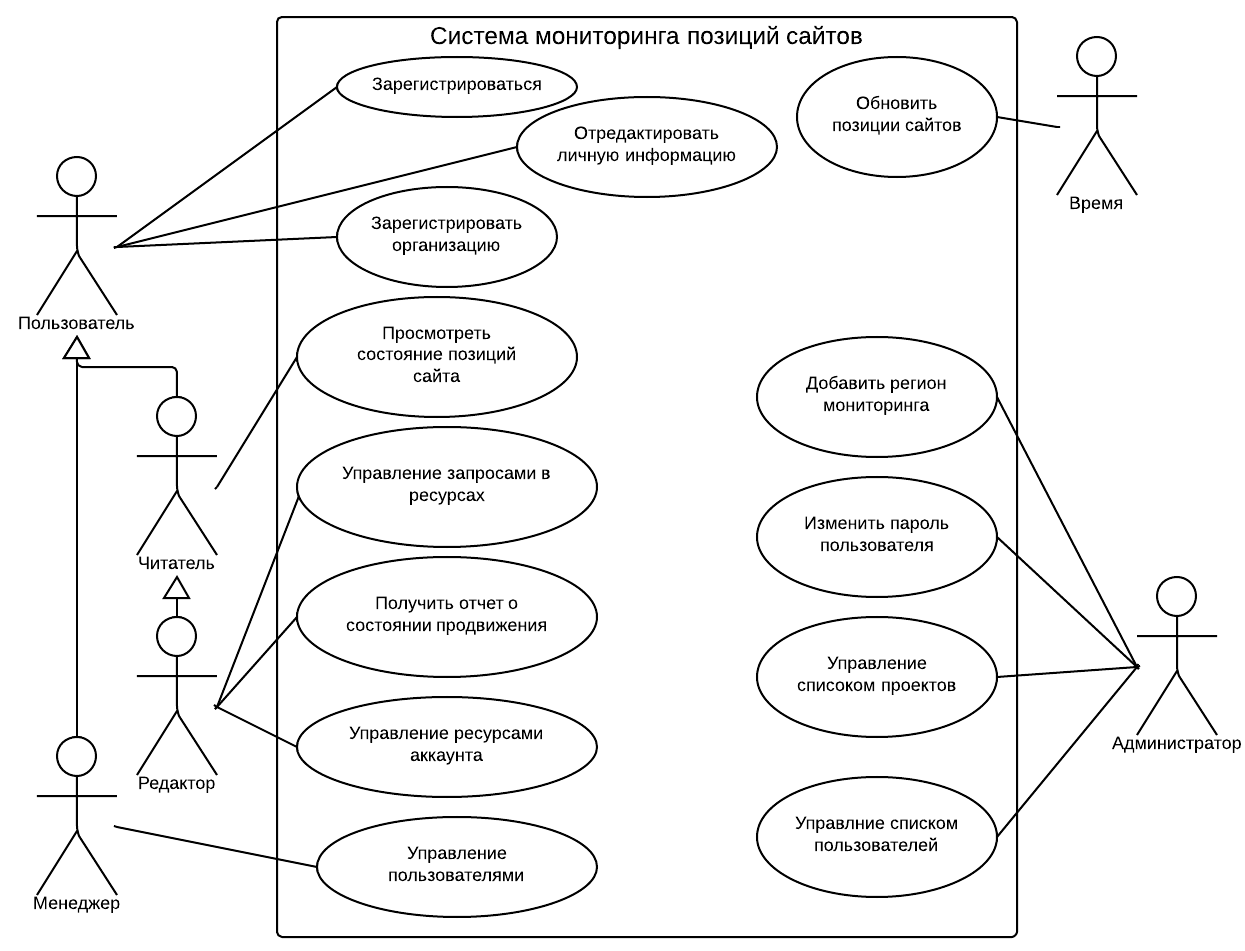
Можно выделить следующих актеров, взаимодействующих с системой мониторинга позиций сайтов (см. рис. 1):

* *администратор* – актер, отвечающий за поддержку работоспособности системы;
* *пользователь* – актер, использующий систему;
* *читатель* – пользователь, просматривающий результаты мониторинга;
* *редактор* – читатель, управляющий параметрами мониторинга;
* *менеджер* – пользователь, управляющий доступом других пользователей к ресурсам;
* *время* – актер, обновляющий состояние системы.

Определим основные варианты использования данными актерами нашей системы.

Администратор инициирует исполнение следующих прецедентов:

1. *управление списком пользователей*: зайдя на страницу редактирования пользователей, администратор может просмотреть список пользователей, добавить нового пользователя или удалить существующего;
2. *управление списком ресурсов*: зайдя на страницу редактирования ресурсов, администратор может просмотреть список ресурсов, добавить новый ресурс или удалить существующий;
3. *изменение пароля пользователя*: зайдя на страницу редактирования пользователя, администратор может задать новый пароль пользователя;
4. *добавление региона мониторинга*: зайдя на страницу редактирования регионов, администратор может добавить новый регион мониторинга.



**Рис. 1.** Диаграмма вариантов использования системы мониторинга позиций сайтов

Пользователь инициирует исполнение следующих прецедентов:

1. *отредактировать личную информацию*: пользователь заходит на форму редактирования личной информации, где он может указать следующую информацию о себе:
   * Ф.И.О.
   * E-mail;
2. *зарегистрироваться:* пользователь выбирает соответствующий пункт при входе на сайт, после чего заполняет форму регистрации для физического лица;
3. *зарегистрировать организацию:* пользователь выбирает соответствующий пункт при входе на сайт, после чего заполняет форму регистрации для организации. После регистрации для организации создается *аккаунт* организации в системе мониторинга.

Менеджер инициирует исполнение следующих прецедентов:

1. *управление пользователями организации*: менеджер может предоставить пользователю следующие права для всех или определенного ресурса организации:
   * просмотреть состояние позиций ресурса;
   * редактирование ресурса;
   * управление правами пользователей на доступ к ресурсу.

Читатель инициирует исполнение следующих прецедентов:

1. *просмотреть состояние позиций ресурса*: пользователь заходит на страницу ресурса и просматривает таблицу позиций сайта по заданным запросам в выбранном регионе и диапазоне дат.

Редактор инициирует исполнение следующих прецедентов:

1. *управление запросами в ресурсах*: нажав на кнопку «Редактировать» на странице ресурса, пользователь может добавить, удалить или изменить один или несколько запросов;
2. *получение отчета о состоянии продвижения*: нажав на кнопку «Экспорт» на странице ресурса, пользователь получает себе на компьютер отчет, в котором содержится таблица позиций сайта по выбранным запросам за заданный период времени в заданном регионе;
3. *управление ресурсами аккаунта:* на странице аккаунта организации пользователь может просмотреть список ресурсов и добавить новый ресурс.

Время инициирует исполнение следующих прецедентов:

1. *обновить позиции сайтов:* каждый день в заданное время запускается сборщик позиций сайтов; собранные позиции записываются в базу данных.

## *Нефункциональные требования к системе мониторинга позиций сайтов*

В результате интервью с заказчиком были сформулированы следующие нефункциональные требования.

Поскольку приложение ориентировано на клиентов, которые могут находится в географически удалённых точках , интерфейс системы мониторинга позиций сайтов в поисковой системе Яндекс должен быть доступен из сети Интернет. При этом все страницы веб-интерфейса должны корректно отображаться в следующих браузерах: Google Chrome 33 и выше, включая мобильную версию; Mozilla Firefox 29.0 и выше.

Для повышения удобства использования был выполнен анализ сценариев работы пользователей. Необходимо учесть результаты анализа для максимально удобного расположения основных элементов интерфейса.

Веб-интерфейс к системе мониторинга позиций сайтов в поисковой системе Яндекс должен быть выполнен с использованием платформы Django, поскольку с ее использованием реализовано большинство проектов заказчика.

## *Вывод*

Таким образом, система мониторинга позиций сайтов в поисковой системе Яндекс должна быть ориентирована на удобство использования клиентами и легко доступна через сеть интернет на различных устройствах. Также система должна интегрироваться в имеющуюся инфраструктуру заказчика.

# Архитектура системы мониторинга позиций сайтов

## *Архитектура MVC*

Веб-приложение — представляет собой вариацию модели MVC (Model-View-Controller). Шаблон проектирования MVC предполагает разделение данных приложения, пользовательского интерфейса и управляющей логики на три отдельных компонента: Модель, Представление и Контроллер – таким образом, что модификация каждого компонента может осуществляться независимо.

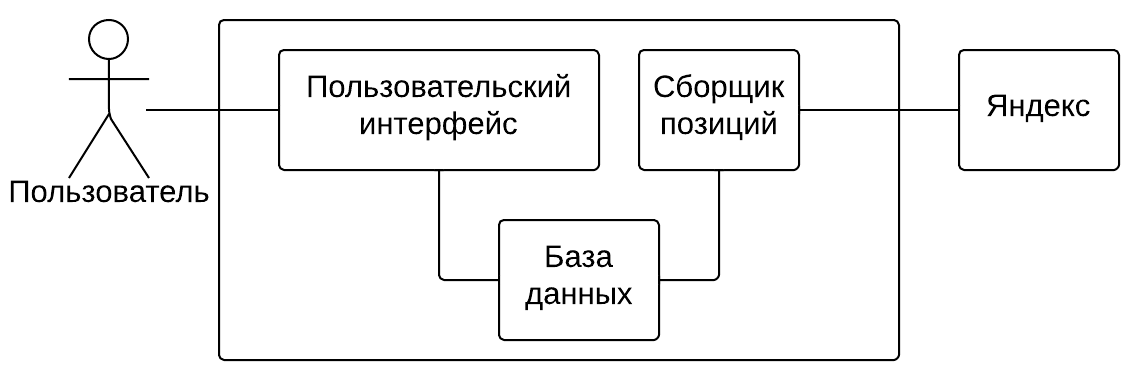
* Модель. Модель предоставляет знания: данные и методы работы с этими данными, реагирует на запросы, изменяя своё состояние. Не содержит информации, как эти знания можно визуализировать.
* Представление, вид. Отвечает за отображение информации (визуализацию). Часто в качестве представления выступает форма с графическими элементами.
* Контроллер. Обеспечивает связь между пользователем и системой: контролирует ввод данных пользователем и использует модель и представление для реализации необходимой реакции.

Представление и контроллер зависят от модели. Но модель не зависит ни от представления, ни от контроллера. Таким образом достигается назначение такого разделения: оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

В Django модели представляют собой классы, отвечающие за ORM и взаимодействие с базой данных, что представляет собой бизнесс-логику системы.

Представления в django —это функции, которые принимают объект класса Request, представляющего собой упрощенное представление http-запроса, и возвращающие объект класса Response, представляющего собой возвращаемый клиенту ответ сервера. Представления могут использовать шаблоны при получении ответа.

Контроллером является веб-платформа django и её механизм распределения путей по представлениям (urls).



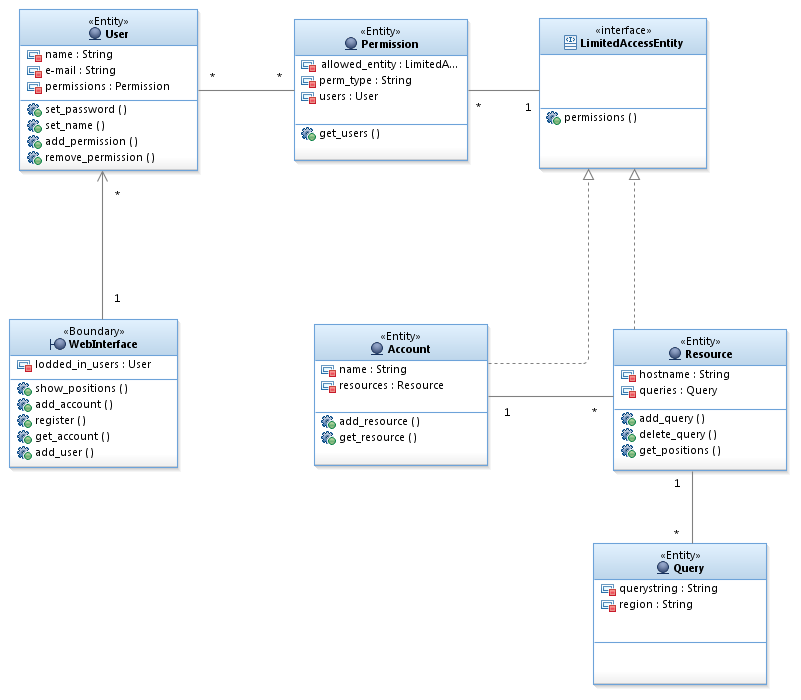
**Рис. 2.** Диаграмма архитектуры системы мониторинга позиций сайтов в поисковой системе Яндекс.

В результате анализа требований к системе было решено выделить следующие классы-сущности (рис. 2):

* User – класс, соответствующий сущности пользователя;
* Account – класс, соответствующий сущности организации;
* Resource – класс, соответствующий сущности ресурса;
* Permission – класс объектов, которые представляют из себя право пользователя на совершение определенных действий с другой конкретной сущностью.

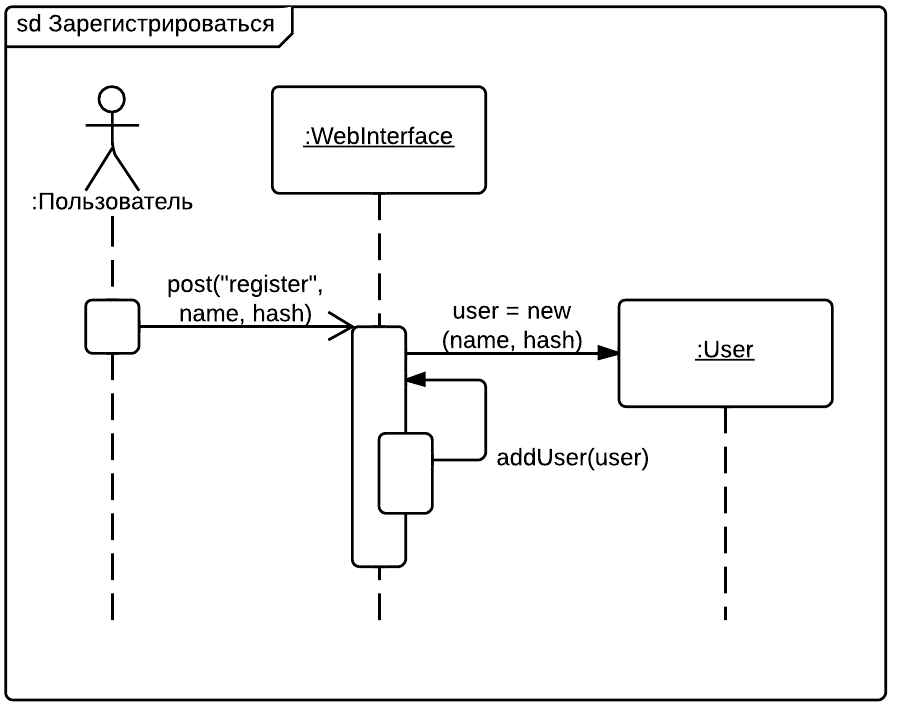
Для обеспечения взаимодействия системы с внешней средой были выделены следующие классы-интерфейсы:

* LimitedAccessEntity – общий интерфейс, представляющий собой сущность, права доступа к которой можно ограничивать с помощью Permission;
* WebInterface – класс, обеспечивающий взаимодействие системы с пользователем.

****

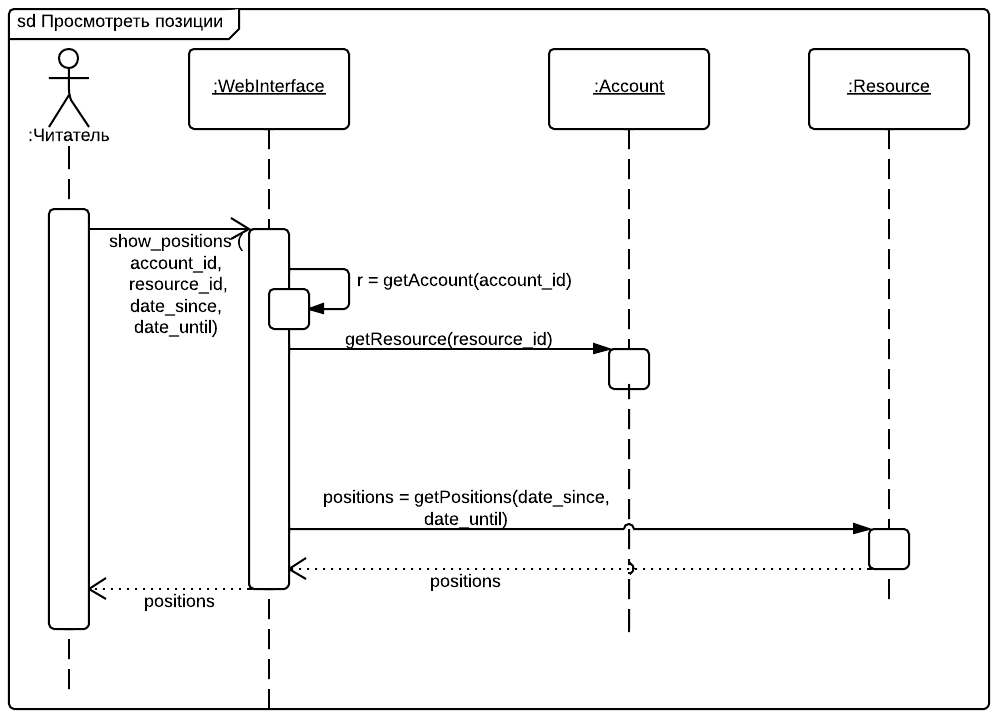
**Рис. 3.** Диаграмма классов анализа системы мониторинга позиций сайтов в поисковой системе Яндекс

На диаграмме последовательностей (рис. 4) изображены экземпляры объектов и сообщения, которыми они обмениваются в рамках прецедента «Зарегистрироваться». Пользователь отправляет сообщение о регистрации в веб-интерфейс, после чего создаётся новый объект «пользователь» и добавляется в хранилище, становясь таким образом доступным через веб-интерфейс.

****

**Рис.4.** Регистрация пользователя в системе мониторинга позиций сайтов

На рис. 5 изображена диаграмма последовательности прецедента «Просмотреть позиции». Пользователь, имеющий право на чтение ресурса, делает запрос на просмотр позиций этого ресурса в поисковой системе Яндекс с помощью веб-интерфейса, указав интересующий промежуток времени. После этого система возвращает требуемый объект типа «Ресурс», и таблицу позиций за указанный промежуток времени. Веб-интерфейс отображает пользователю желаемые данные.



**Рис. 5.** Просмотр позиций ресурса по заданным запросам в поисковой системе Яндекс

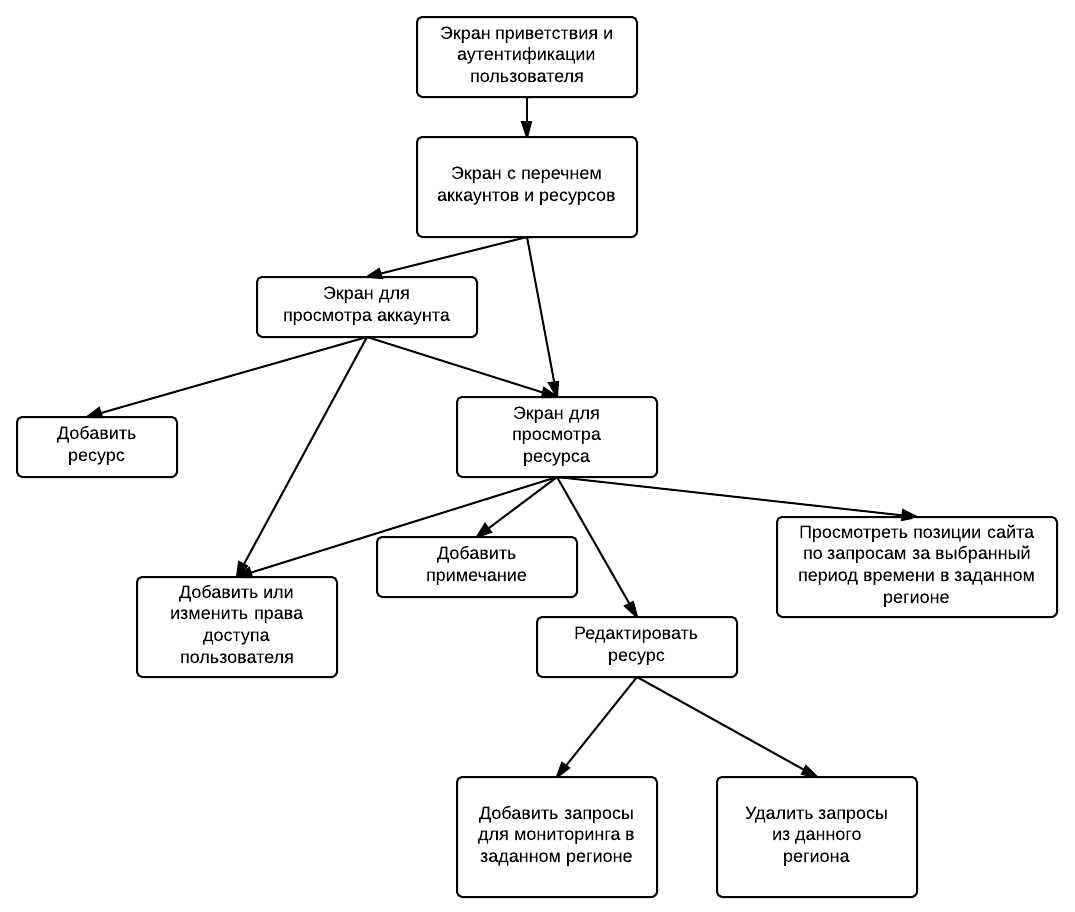
## *Интерфейс веб-приложения*

Интерфейс веб-приложения предоставляет отдельные методы работы с системой в зависимости от того, какими правами обладает пользователь, вошедший в систему:

* интерфейс пользователя;
* интерфейс читателя;
* интерфейс редактора;
* интерфейс менеджера;
* интерфейс администратора.

Один пользователь может получить доступ сразу к нескольким интерфейсам, если соответствующие роли прописаны в его пользовательском профиле.

Диаграмма доступных пользовательских интерфейсов представлена на Рис. 6.



**Рис. 6.** Диаграмма доступных пользовательских интерфейсов системы мониторинга позиций сайтов

## *Вывод*

В соответствии с системными требованиями была спроектирована архитектура веб-приложения для отображения данных пользователю системы мониторинга позиций сайтов в поисковой системе Яндекс. Веб-приложение было решено реализовать с использованием шаблона проектирования MVC, разделяющего данные приложения на три отдельных компонента. Кроме того, в соответствии с функциональными требованиями были выделены основные классы-сущности и классы-интерфейсы. В соответствии с нефункциональными требованиями была подробно разработана реализация наиболее важнейших прецедентов. В соответствии с анализом сценариев работы пользователя были разработаны доступные пользовательские интерфейсы системы мониторинга позиций сайтов в поисковой системе Яндекс.

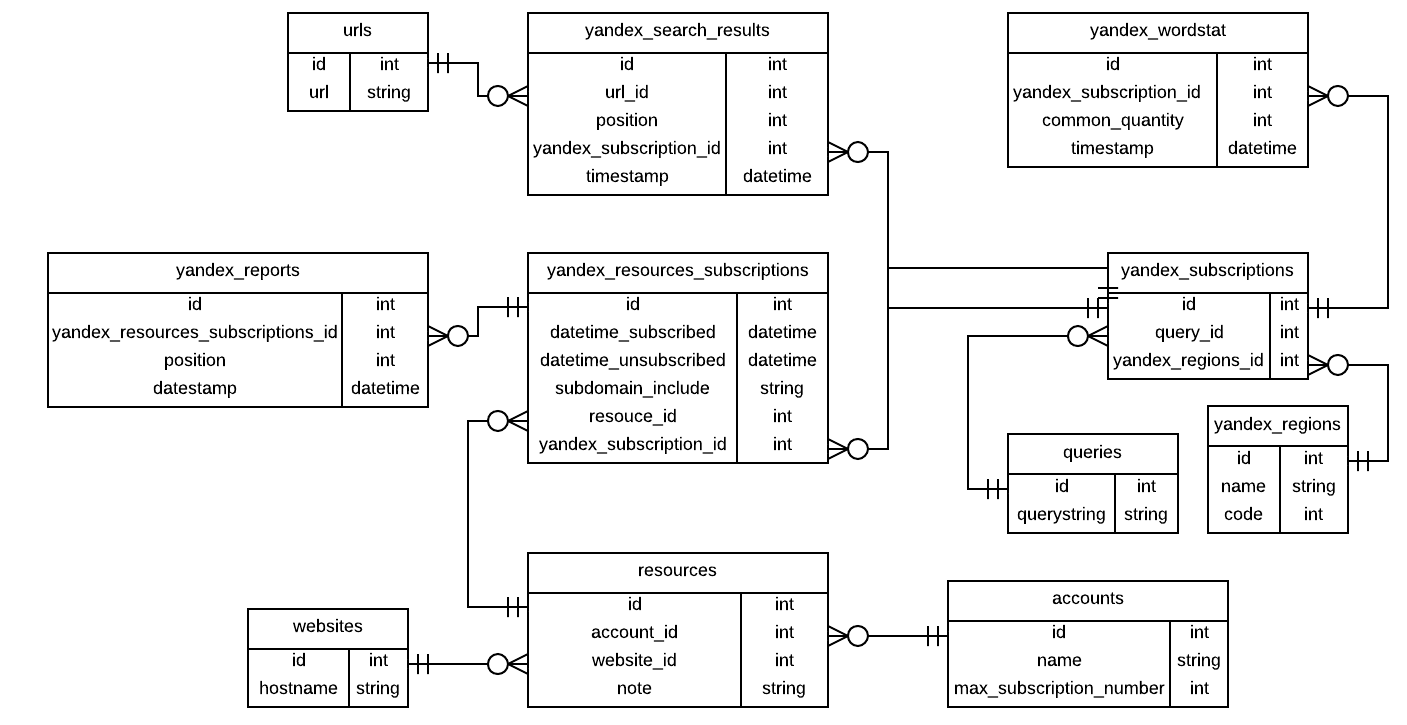
# Реализация системы мониторинга позиций сайтов

## *Структура базы данных мониторинга*

Для реализации веб-приложения для работы с системой мониторинга позиций сайтов была использована база данных, состоящая из следующих таблиц:

* таблица аккаунтов, состоящая из данных обо всех организациях, использующих систему;
* таблица ресурсов, содержащая информацию обо всех ресурсах, используемых системой;
* таблица для связи информации о запросе с конкретным регионом;
* таблица, содержащая данные для формирования отчета о состоянии продвижения;
* таблица, содержащая имена хостов веб-сайтов, которые использует система для мониторинга позиций;
* таблица, содержащая строки запросов, по которым производится мониторинг;
* таблица, содержащая коды регионов, по которым производится мониторинг;
* таблица, связывающая поисковую строку и код региона, по которым производится мониторинг;
* таблица, в которую записывается информации о частотности запросов по регионам;
* таблица, содержащая информацию о позиции запроса для каждого исследуемого региона в поисковой системе Яндекс;
* таблица, содержащая пути доступа к информации о результатах мониторинга.

Схема используемой базы данных изображена на рис. 7.

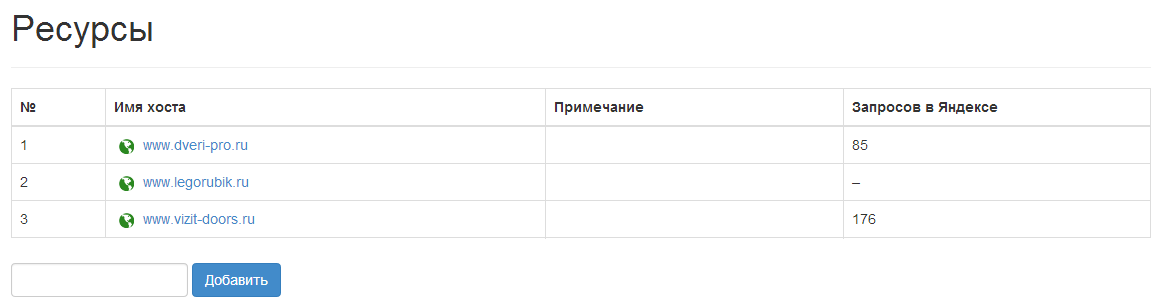


**Рис.7.** Схема базы данных системы мониторинга позиций сайтов

## *Страница аккаунта*

Страница аккаунта представляет собой перечень ресурсов, позиции по которым доступны к просмотру пользователям данного аккаунта. Ключевым элементом страницы аккаунта является таблица, в которой перечислены все доступные ресурсы, количество запросов, по которым проводится мониторинг этих ресурсов и примечание. Для пользователей, имеющих право редактирования данного аккаунта, доступна возможность добавлять новые ресурсы, по которым будет производиться мониторинг. Для этого пользователю необходимо ввести доменное имя интересующего ресурса в предназначенное для этого текстовое поле и нажать кнопку «Добавить». Реализация данной функциональности представлена в листинге 1 приложения 1.

Пример таблицы ресурсов на странице редактирования аккаунта представлен на рис.8. Прочерк в столбце «Запросов в Яндексе» означает, что мониторинг по данному ресурсу временно не ведется.



**Рис. 8.** Таблица ресурсов на странице аккаунта

## *Отображение позиций*

Отображение позиций запрашиваемого ресурса по заданным поисковым запросам в поисковой системе Яндекс пользователю реализовано с помощью таблицы, столбцы которой представляют собой даты из выбранного диапазона, а строки – текст поискового запроса. В каждой ячейке таблицы содержится натуральное число, обозначающее позицию ресурса в данную дату по конкретному запросу.

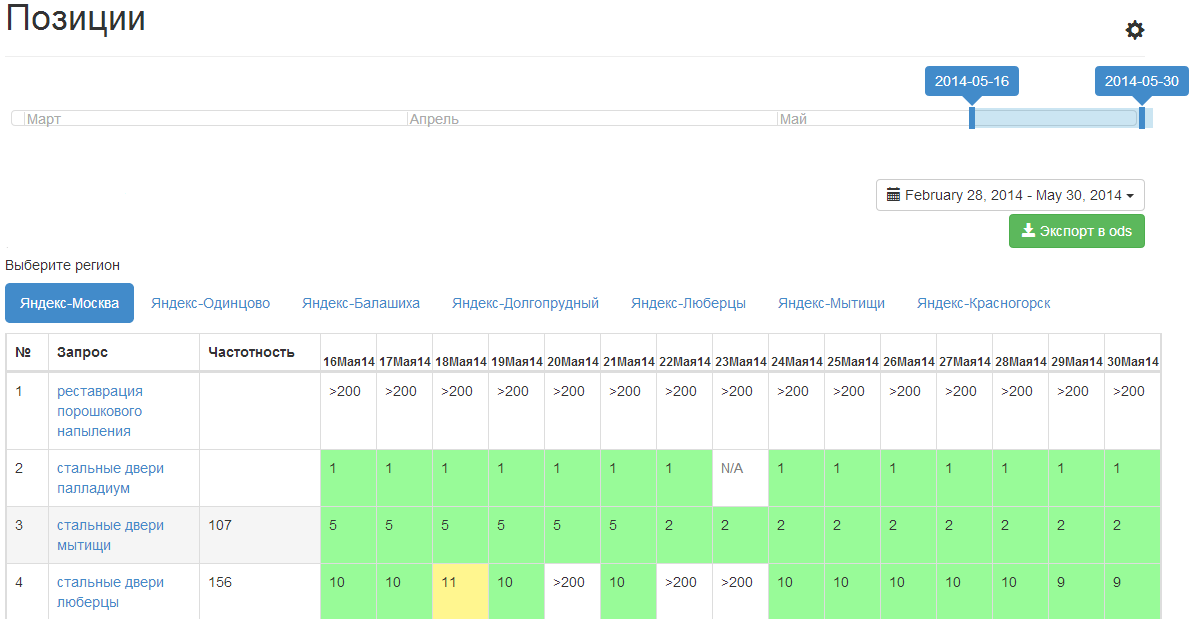
Выбор интересующего региона реализован с помощью таблеток, находящихся над таблицей, каждая из которых отвечает за конкретный регион, в котором производится мониторинг конкретного запроса. При клике на таблетку происходит загрузка таблицы из базы данных, в которой отображаются только те запросы, по которым производится мониторинг данного ресурса в выбранном регионе.

Выбор диапазона дат реализован в виде слайдера, с помощью которого можно выделить интересующий временной промежуток с точностью до дня. По умолчанию загружаются данные за последние две недели. Пользователь может менять длины выбранного диапазона от одной до трех недель, а также выбрать любой промежуток времени мониторинга. Временной интервал диапазона, из которого можно выбирать также регулируется пользователем с помощью выпадающего списка. По умолчанию длина данного диапазона составляет три последних месяца.

Функция, используемая при отображении позиций для связи с базой данных, представлена в листинге 2 приложения 1.

На рис. 9. Представлен результат мониторинга позиций интернет ресурса mirdverey.com в регионе Яндекс-Москва с 16 по 30 мая 2014 года по следующим поисковым запросам:

* реставрация порошкового напыления;
* стальные двери палладиум;
* стальные двери мытищи;
* стальные двери люберцы.



**Рис.9.** Таблица мониторинга позиций сайта mirdverey.com

## *Страница редактирования ресурса*

Переход на страницу редактирования ресурса осуществляется со страницы просмотра позиций и доступен только пользователям, имеющим соответствующее право. На странице редактирования ресурса пользователь может изменить примечание, добавить или удалить запросы для мониторинга, добавить или удалить регион мониторинга для одного или нескольких запросов.

Ключевым элементом страницы редактирования ресурса является таблица запросов, заголовками столбцов которой являются названия регионов, для которых производится мониторинг данного ресурса. Если для конкретного запроса осуществляется мониторинг в конкретном регионе, то на их пересечении в ячейке таблицы содержатся глубина поиска и поддомены – основные параметры для мониторинга позиций ресурса в поисковой системе Яндекс. По умолчанию глубина поиска равна 200, а мониторинг осуществляется по доменам и поддоменам. Пользователь может изменить эти параметры, выбрав другое значение из выпадающего списка. Если мониторинг не осуществляется, ячейка на пересечении запроса и региона остается пустой. Для того, чтобы мониторинг данного запроса по выбранному региону начался, достаточно кликнуть левой кнопкой мыши на пустую ячейку, после чего информация об этом будет записана в базу данных. Реализация данной функциональности представлена в листинге 3 приложения 1.

Для того, чтобы мониторинг данного запроса по выбранному региону остановился, достаточно кликнуть левой кнопкой мыши на крестик, находящийся в соответствующей ячейке, после чего информация об этом будет записана в базу данных. В случае, если по региону в данном ресурсе не производится мониторинг ни по одному запросу или запрос не мониторится ни по одному региону, он удаляется из таблицы автоматически.

Для добавления региона, мониторинг по которому для ресурса ранее не осуществлялся, над таблицей запросов находится кнопка «Добавить регион», где пользователь может выбрать необходимое из выпадающего списка. При клике на регион в данном списке, в таблицу запросов добавляется новый столбец, заголовок которого соответствует названию выбранного региона.

Для добавления нового запроса пользователь может воспользоваться формой, расположенной под таблицей, куда необходимо ввести имя нового запроса и отметить регионы, по которым будет производиться мониторинг.

При нажатии левой кнопкой мыши на заголовок строки или столбца, выделяются вся строка или столбец и становится доступна кнопка «Действие», расположенная в верхней части страницы. При нажатии на нее появляется возможность изменить глубину поиска, поддомены или удалить все выделенное.

На рис.10. показан пример страницы редактирования ресурса mirdverey.com



**Рис. 10.** Страница редактирования ресурса mirdverey.com

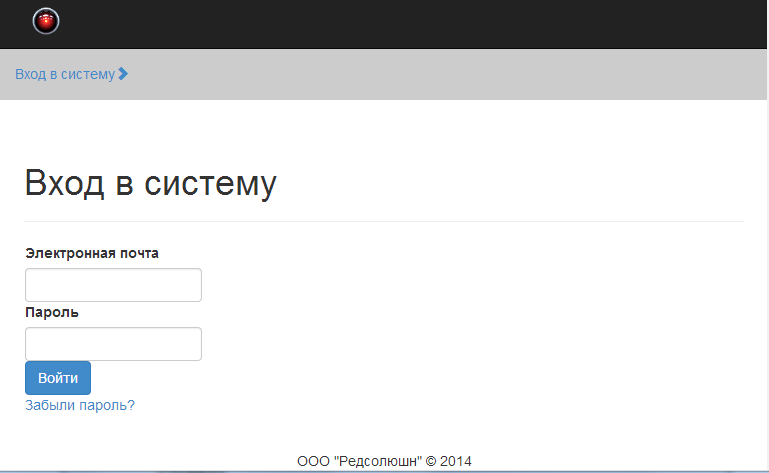
## *Организация безопасности приложения*

Безопасность приложения осуществляется на основе создания пользователей и ролей. Ролям соответствуют логические роли внутри приложения. Логические роли описывают правила, согласно которым разрешается доступ к различным частям приложения. Каждому пользователю назначается одна или несколько логических ролей.

Подключение к приложению происходит в два этапа:

1. Аутентификация. Система безопасности проверяет права доступа пользователя на основании его имени и пароля.

2. Авторизация. После прохождения этапа авторизации система безопасности передает приложению информацию о пользователе в виде списка предоставленных ему ролей. (рис.11.)



**Рис. 11.** Реализация входа в приложение

В зависимости от имени подключившегося пользователя и его прав можно скрыть или сделать недоступными определенные части приложения.

Для того, чтобы иметь доступ к системе, каждый пользователь должен быть зарегистрирован администратором с помощью интерфейса администрирования. После того, как созданы пользователи, нужно создать роли и распределить по ним пользователей (Рис. 12). В веб-приложении для организации работы системы мониторинга позиций сайтов создано 3 роли:

1. reader – роль читателя приложения;
2. editor – роль редактора приложения;
3. manager – роль управляющего доступом пользователей к приложению;

Для каждой сущности роли пользователям присваиваются отдельно. Так, если пользователю дано право читать аккаунт, то ему автоматически присваивается право чтения для всех ресурсов, принадлежащих аккаунту. Интерфейс управления доступом пользователей представлен на рис.12.

**Рис. 12.** Интерфейс управления доступом пользователей

Модель безопасности приложения разделяет все страницы на несколько подмножеств в зависимости от того, какие группы пользователей могут иметь к ним доступ.

1. *Не подключившиеся пользователи.* Этим пользователям доступна только начальная страница.
2. *Читатели.* Этим пользователям доступны начальная страница в состоянии после входа в систему и страницы просмотра ресурсов аккаунта в случае предоставления права всему аккаунту. Иначе – только список доступных ему ресурсов. На странице ресурса пользователи могут просматривать позиции в любом регионе за любой выбранный период времени, в котором производился мониторинг.
3. *Редакторы.* Этим пользователям доступен весь интерфейс, к которому имеют доступ читатели. При предоставлении права редактирования аккаунта, пользователю, обладающему ролью редактора, предоставляется интерфейс добавления ресурсов для мониторинга у этого аккаунта. При предоставлении права редактирования ресурса пользователю, обладающему ролью редактора, предоставляется интерфейс добавления, удаления запросов из данного ресурса.
4. *Менеджеры.* Этим пользователям доступны начальная страница в состоянии после входа в систему и страницы аккаунта и ресурса, на которых ему отображены только таблицы управления доступом пользователей к данному объекту (рис. 8).
5. *Администраторы.* Этим пользователям доступны все страницы приложения, включая интерфейс администрирования.

Задача ограничения доступа к страницам задача решается средствами авторизации, предоставляемыми платформой Django. Реализация функциональности добавления и удаления прав пользователей представлена в листинге 4 приложения 1.

## *Вывод*

На основе разработанной архитектуры было реализовано веб-приложение, представляющее собой пользовательский интерфейс системы мониторинга позиций сайтов. Информация для отображения берется из базы данных, созданной с помощью СУБД PostreSQL. С целью систематизации данных каждая страница содержит в себе таблицу, где содержится основная информация. Для удобства редактирования информациио ресурсе создана отдельная страница, на которой можно производить манипуляции с запросами. Безопасность приложения организована с помощью предоставления каждому пользователю различных видов прав на различные объекты системы.

# Тестирование системы мониторинга позиций сайтов в поисковой системе Яндекс

Для тестирования веб-приложения использовались следующие методы тестирования:

* функциональное тестирование;
* проверка эргономичности;
* тестирование ссылок;
* тестирование интерфейса пользователя.

Тестирование проводилось на реальных данных пользователей, которые являются сотрудниками организации.

**Функциональное тестирование** – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям. Функциональные требования определяют, что именно делает программное обеспечение, какие задачи оно решает. Набор тестов на функциональность представлен в таблице 1.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название теста** | **Шаги** | **Ожидаемый результат** | **Тест пройден?** |
| 1 | Регистрация | 1. На странице аутентификации выбрать кнопку «Зарегистрироваться». 2. Заполнить форму регистрации. 3. Нажать на кнопку «Готово». | В базе данных появляется информация о новом пользователе. | Да |
| 2 | Вход в систему | 1. На странице аутентификации ввести зарегистрированный адрес электронной почты и верный пароль. 2. Нажать на кнопку «Войти». | Пользователю становится доступна главная страница сайта. | Да |
| 3 | Забытый пароль | 1. На странице аутентификации нажать на кнопку «Забыли пароль». 2. Ввести адрес электронной почты. 3. Нажать на кнопку «ОК» | На указанный электронный адрес приходит сообщение с запросом на смену пароля. | Да |
| 4 | Добавление ресурсов | 1. На главной странице выбрать один из доступных пользователю аккаунтов. 2. В форму для добавления ресурса вписать корректное доменное имя ресурса. 3. Нажать на кнопку «Добавить». | В таблице ресурсов для аккаунта появляется новое поле. Пользователю выдается сообщение о добавлении ресурса | Да |
| 5 | Ошибка при добавлении ресурсов | 1. На главной странице выбрать один из доступных пользователю аккаунтов. 2. В форму для добавления ресурса вписать некорректное доменное имя ресурса. 3. Нажать на кнопку «Добавить». | Пользователю выдается сообщение об ошибке при добавлении ресурса | Да |
| 6 | Просмотр позиций | 1. На странице ресурса выбрать с помощью слайдера промежуток времени. 2. Выбрать регион с помощью таблеток. | Пользователю отображается таблица позиций для данного ресурса за выбранный промежуток времени в выбранном регионе. | Да |
| 7 | Ошибка при просмотре позиций | 1. На странице ресурса выбрать с помощью слайдера промежуток времени. 2. Выбрать регион с помощью таблеток. | Пользователю отображается сообщение «Нет позиций» | Да |
| 8 | Добавление запроса | 1. На странице редактирования ресурса вписать строку запроса в текстовое поле. 2. Нажать на кнопку «Добавить». 3. С помощью кликов левой кнопки мыши по пустым ячейкам добавить регионы мониторинга. | В таблице запросов появляется новое поле с только что добавленным запросом. Ему присвоены параметры по умолчанию. | Да |
| 9 | Редактирование параметров запроса | 1. На странице редактирования ресурса выделить одну или несколько ячеек. 2. Нажать на кнопку «Действие». 3. Выбрать новые параметры запросов из выпадающего списка. 4. Нажать на кнопку «Применить». | Изменены параметры в таблице запросов. | Да |
| 10 | Удаление запроса | 1. На странице редактирования ресурса нажать на крестик в непустой ячейке. 2. Подтвердить действие. | Выбранная ячейка оказывается пуста. | Да |
| 11 | Добавление прав | 1. На странице аккаунта или ресурса выбрать право для пользователя из выпадающего списка. 2. Подтвердить действие. | Пользователь получил добавленное право. | Да |
| 12 | Удаление прав | 1. На странице аккаунта или ресурса нажать на крестик, расположенный на таблетке права для обозначения права пользователя. 2. Подтвердить действие. | Пользователь лишился удаленного право. | Да |

**Проверка эргономичности** – метод оценки удобства продукта в использовании, основанный на привлечении пользователей в качестве тестировщиков, испытателей и суммировании полученных от них выводов.

При прохождении данного теста были привлечены сотрудники компании. Тестировщикам было предложено решить следующие задачи:

* зарегистрироваться;
* добавить ресурсы в аккаунт;
* добавить права доступа другим пользователям к данному аккаунту;
* добавить запросы в созданные ресурсы, добавить для них параметры и регионы мониторинга;
* просмотреть собранные позиции для добавленных ресурсов спустя несколько дней;
* удалить запросы из ресурса.

Все задачи были решены всеми тестировщиками без каких-либо затруднений. Тест пройден успешно.

Далее проводилось тестирование ссылок на работоспособность. Тестирование прошло успешно.

Тестирование интерфейса пользователя. Интерфейс был протестирован на браузерах Internet Explorer, Mozilla Firefox, Opera, Google Chrome. Не возникло никаких проблем с версткой. Интерфейс во всех браузерах был идентичным. Тест пройден успешно.

# Заключение

В ходе проделанной работы мной были изучены особенности существующих систем мониторинга позиций сайтов в поисковых системах, выделены их преимущества и недостатки. На основе полученных знаний была спроектирована архитектура системы мониторинга позиций сайтов в поисковых системах.

В процессе работы были изучены существующие подходы разработки систем мониторинга позиций сайтов в поисковых системах, выбран наиболее оптимальный из них, и на его основе реализовано веб-приложение для мониторинга позиций сайтов в поисковой системе Яндекс. Для этого были изучены возможности работы со средой программрования django и платформой Twitter Bootstrap.

Было произведено тестирование приложения на реальных пользовательских данных. Приложение успешно прошло предложенные тесты.

Разработанное веб-приложение внедрено в эксплуатацию, и сейчас его активно используют сотрудники и клиенты IT-компании.

**Литература**

1. Bar-Ilan J., Mat-Hassan M., Levene M. Methods for comparing rankings of search engine results // Comput. Networks. 2006. Vol. 50, № 10. P. 1448–1463.

2. В.В. Зосимов, В.С. Степашко. Построение и сравнительный анализ моделей ранжирования результатов работы поисковых систем google и яндекс // Індуктивне моделювання складних систем. № 3. 2011. P. 69–77.

3. И. Сегалович, М. Маслов Некоторые аспекты полнотекстового поиска и ранжирования в Яндекс. 2004. [Электронный ресурс] URL: <http://download.yandex.ru/company/experience/romip2004/romip2004_aspects.pdf> (дата обращения 01.06.2014)

4. Статистика поисковых систем . [Электронный ресурс] URL: http://www.liveinternet.ru/stat/ru/searches.html (28.05.2014).

5. Определение позиций сайта в поисковиках Яндекс и Гугл . [Электронный ресурс] URL: http://www.semonitor.ru/positioning.html (29.05.2014).

6. Как работают поисковые системы - SearchEngines.ru . [Электронный ресурс] URL: http://www.searchengines.ru/articles/004556.html (30.05.2014).

7. SEO анализ . [Электронный ресурс] URL: http://www.seobuilding.ru/ (01.06.2014).

8. Докуентация Django . [Электронный ресурс] URL: https://www.djangoproject.com/ (28.05.2014).

9. Документация Twitter Bootstrap . [Электронный ресурс] URL: http://getbootstrap.com/ (дата обращения 28.05.2014).

# Приложение 1. Реализация прецедентов

**Листинг 1.** Добавление ресурса

class AddResourceForAccount(permission\_on\_object\_required(Account, "edit", "pk"), base.View):

@staticmethod

def post(request, pk):

account = get\_object\_or\_404(Account, pk=pk)

form = WebsiteAddForm(request.POST)

if form.is\_valid():

hostname = form.cleaned\_data.get('hostname')

website = get\_object\_or\_none(Website, hostname=hostname)

if not website:

website = Website.objects.create(hostname=hostname)

resource = get\_object\_or\_none(Resource, account=account, website=website)

if not resource:

resource = Resource.objects.create(account=account, website=website, note=u"")

messages.add\_message(request, messages.SUCCESS, u"Ресурс %s успешно добавлен" % resource.website.hostname)

else:

messages.add\_message(request, messages.WARNING, u"Ресурс %s уже существует" % resource.website.hostname)

else:

messages.add\_message(request, messages.ERROR, u"Неверное имя хоста", extra\_tags='danger')

return HttpResponseRedirect(reverse("account", kwargs={'pk': pk}) + '#sites')

**Листинг 2.** Отображение позиций

class PositionView(permission\_on\_object\_required(Resource, 'read', 'resource\_id'), base.View):

@staticmethod

def get(request, resource\_id):

get\_object\_or\_404(Resource, pk=resource\_id)

if request.GET['region'] and request.GET['until'] and request.GET['since']:

region = get\_object\_or\_404(Region, pk=request.GET['region'])

date\_until = dateutil.parser.parse(request.GET['until']).date()

date\_since = dateutil.parser.parse(request.GET['since']).date()

checks = [date\_since + datetime.timedelta(days=x+1)

for x in range(0, (date\_until - date\_since).days + 1)]

positions = view.get\_positions(checks[0], checks[-1], region)

if positions:

fixed\_dates = map(lambda x: x.date, view.fixed\_dates.order\_by("date"))

checks = fixed\_dates + checks

return render\_to\_response('resources/positions.html',

RequestContext(request,{'checks':checks, 'fixed\_dates': fixed\_dates, 'positions': positions,

'region': region,

'current\_date': datetime.datetime.today(),

'object': view}))

return render\_to\_response('resources/empty\_positions.html')

**Листинг 3.** Изменение параметров запроса

class ChangeAll(permission\_on\_object\_required(Resource, 'edit', 'resource\_id'), base.View):

@staticmethod

def post(request, resource\_id):

resource = get\_object\_or\_404(Resource, pk=resource\_id)

number\_search = int(request.POST["number\_search"])

number\_subdomain = int(request.POST["number\_subdomain"])

action\_type = request.POST["type"]

search\_depths = [50, 100, 200, 350, 500]

subdomain\_includes = ['strict\_domain', 'only\_subdomains', 'domain\_with\_subdomains']

if(number\_search < 0 or number\_search > len(search\_depths)-1 or number\_subdomain < 0 or

number\_subdomain > len(subdomain\_includes)-1):

return HttpResponseBadRequest()

search\_depth = search\_depths[number\_search]

subdomain\_include = subdomain\_includes[number\_subdomain]

for q\_id, r\_id in literal\_eval(request.POST['subscriptions']):

query = get\_object\_or\_404(Query, pk=q\_id)

region = get\_object\_or\_404(Region, pk=r\_id)

if ResourceSubscription.objects.filter(subscription\_\_query=query, subscription\_\_region=region, resource=resource).exists():

rs = ResourceSubscription.objects.get(subscription\_\_query=query, subscription\_\_region=region, resource=resource)

if action\_type == 'change':

if rs.datetime\_unsubscribed:

rs.datetime\_unsubscribed = None

rs.datetime\_subscribed = datetime.datetime.now()

rs.search\_depth = search\_depth

rs.subdomain\_include = subdomain\_include

rs.save()

elif action\_type == 'delete':

rs.datetime\_unsubscribed = datetime.datetime.now()

rs.save()

elif action\_type == 'change':

subscription, created = Subscription.objects.get\_or\_create(query=query, region=region) ResourceSubscription.objects.create(subscription=subscription,

resource=resource, search\_depth=search\_depth,

datetime\_subscribed=datetime.datetime.now(), subdomain\_include=subdomain\_include)

return HttpResponseRedirect(reverse("edit\_resource", kwargs={"resource\_id": resource\_id}))

**Листинг 4.** Управление правами пользователей

class ManagePerm(base.View):

@staticmethod

def get(request, action, model, pk, username, perm\_type):

obj = get\_object\_or\_404(get\_model('resources', model), pk=pk)

if has\_perm('manage')(request.user, obj):

redirect = '/'

if 'redirect' in request.GET:

redirect = request.GET['redirect']

perm = obj.perms.get(perm\_type=perm\_type).permission

user = User.objects.get(username=username)

if action == 'remove':

user.user\_permissions.remove(perm)

return HttpResponseRedirect(redirect)

elif action == 'add':

user.user\_permissions.add(perm)

return HttpResponseRedirect(redirect)

else:

raise Http404