

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ  
ФЕДЕРАЦИИ Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет»  
(национальный исследовательский университет)  
Высшая школа экономики и управления  
Кафедра «Информационные технологии в экономике»

РАБОТА ПРОВЕРЕНА

Рецензент, \_\_\_\_\_ / \_\_\_\_\_ /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Зав. кафедрой, д.т.н., с.н.с.  
\_\_\_\_\_ / Б.М. Суховилов /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Разработка сайта по поиску помощника в различных областях

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА  
ЮУрГУ – 09.03.03.2017.085.ВКР

Консультант, ассистент

\_\_\_\_\_ / С.Ю. Нестеренко /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Руководитель, к.т.н., доцент

\_\_\_\_\_ / А.Г. Палей /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Автор

студент группы ВШЭУ-436  
\_\_\_\_\_ / Ю.А. Понасенкова /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Нормоконтролер,  
старший преподаватель

\_\_\_\_\_ / Е.А. Конова /  
« \_\_\_\_ » \_\_\_\_\_ 2017 г.

Челябинск 2017

## АННОТАЦИЯ

Понасенкова Ю.А. Разработка сайта по поиску помощника в различных областях. Челябинск: ЮУрГУ, ВШЭУ-436, 57 с., 29 ил., 1 табл., библиографический список – 6 наим., 5 прил..

Выпускная квалификационная работа выполнена с целью разработки веб-приложения для поиска исполнителей в различных сферах деятельности, позволяющего создавать задачи различного характера для исполнителей, публиковать статьи, задавать вопросы и давать ответы на вопросы других пользователей.

Обоснована актуальность темы, сформулирована цель и вытекающие из нее задачи. Осуществлен анализ существующей информационной системы и проведен сравнительный анализ аналогов на рынке.

Описана структура веб-приложения и его работа.

В экономической части работы рассчитаны затраты на реализацию и внедрение программного продукта.

# ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ .....	5
1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ .....	6
1.1 Описание предметной области.....	6
1.2 Обзор аналогов.....	6
Вывод по первому разделу .....	7
2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА .....	8
2.1 Выбор программных средств .....	8
2.2 Шаблон проектирования MVC .....	9
2.3 Проектирование диаграммы прецедентов.....	10
2.4 Проектирование базы данных .....	12
2.5 Структура и описание работы приложения .....	13
Вывод по второму разделу .....	21
3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА .....	22
3.1 Базовая структура проекта .....	22
3.2 Migrations .....	22
3.2 Seeders .....	23
3.3 Controllers .....	23
3.4 Посредники .....	25
3.5 Маршрутизация (routes) .....	25
3.6 Представления (views) .....	25
3.7 Провайдеры(providers) .....	26
3.8 Фасады .....	26
3.9 Классы.....	26
Вывод по третьему разделу .....	26
4 РАСЧЕТ ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ.....	27
4.1 Внедрение .....	27
4.2 Обслуживание .....	27
Вывод по четвертому разделу.....	27

ЗАКЛЮЧЕНИЕ .....	28
БИБЛИОГРАФИЧЕСКИЙ СПИСОК .....	29

## **ВВЕДЕНИЕ**

Объектом исследования дипломного проекта является рынок фрилансеров в России.

Целью дипломного проекта является разработка веб-приложения для поиска исполнителей, позволяющая пользователям создавать задачи различного характера, выполнять задания других пользователей, а также помогать пользователям, отвечая на заданные вопросы и публиковать собственные статьи.

Из поставленной цели вытекают следующие задачи.

1. Сравнение аналогов на рынке и выявление их недостатков.
2. Разработка удобного, интуитивно-понятного интерфейса.
3. Предоставление возможности создания задач, вопросов и статей.
4. Расчет экономических показателей.

# **1 АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ**

## **1.1 Описание предметной области**

Фрилансер – это свободный работник, который самостоятельно занимается поиском заказов и клиентов, не зачисляясь в штат сотрудников компании.

В связи с масштабным сокращением работников, количество фрилансеров в России возросло в 3 раза с 2014 г. по 2015 г. Это также связано с тем, что компании стали прибегать к найму внештатных работников для сокращения расходов на аренду помещений, страховку и зарплаты, а также для сокращения бюджета и сроков проектов.

## **1.2 Обзор аналогов**

Существующие популярные веб-сервисы для поиска исполнителей такие «Freelance.ru», «Freelancejob.ru» ориентированы на компании и лиц, предоставляющих частные профессиональные услуги (такие как написание сайта под ключ, логотипов и фирменных стилей, создание ландшафтного дизайна).

«Freelance.ru» взимает комиссию за публикацию/выполнение заданий (8% для исполнителей) и имеет достаточно сложную систему рейтинга деловой активности, включающую в себя полноту наполненности профиля (10 баллов за каждый заполненный пункт), наличие работ в профиле (150 баллов за 1 работу) и количестве положительных и отрицательных отзывов (5 баллов и -5 баллов соответственно).

«Freelancejob.ru» имеет схожую систему рейтингов с вышеописанным сервисом. Не взимает плат за публикацию/выполнение заданий с пользователей, но ориентирован исключительно на профессиональных фрилансеров, предоставляющий частные услуги.

Основную конкуренцию составляет веб-сервис YouDo, основанный в 2015 г., специализирующийся на бытовых и бизнес-услугах.

В таблице 1 приведен сравнительный анализ имеющихся решений и собственного продукта.

Таблица 1 – Сравнение аналогов

Критерий Веб-сервис	Сложная система рейтингов	Ориентация на компания и част- ные лица с проф. услугами	Комиссия за пуб- лика- цию/выполнение задания
«Freelance.ru»	+	+	+
« Freelancejob.ru »	–	+	+
«YouDo»	+	–	+
Мое решение	+	–	–

### Вывод по первому разделу

На данный момент существующие веб-сервисы ориентированы на профессиональных фриласеров, которые являются компаниями и лицами, предоставляющие частные профессиональные услуги, но очень мало веб-сервисов, которые предоставляют возможность создания быстрой и одноразовой задачи не требующие профессиональной подготовки и опыта работы в данной сфере. Одним из таких является сервис «YouDo», который взимает процент за публикацию/выполнение задач.

Следовательно, необходимо разработать сайт, включающий в себя положительные качества конкурентов и содержащий новое интересное решение, которое будет отличать его от остальных.

## 2 ПРОЕКТИРОВАНИЕ ПРОГРАММНОГО ПРОДУКТА

### 2.1 Выбор программных средств

Для разработки данного дипломного проекта необходимы различные инструментальные средства разработки, которые позволят ускорить процесс создания веб-приложения.

Перечень необходимых инструментов и программных средств.

1. Atom [1].
2. phpMyAdmin.
3. Laravel 5.4 [2].
4. Интерпретатор PHP [3].
5. MySQL.
6. Bootstrap [4].
7. Apache 2 [5].
8. Composer [6].

Для установки инструментов разработки использован **OpenServer** – портативная серверная платформа и программная среда, включающий в себя все необходимые для работы дистрибутивы: систему управления базами данных MySQL, веб-сервер Apache, интерпретатор PHP, phpMyAdmin для администрирования СУБД MySQL и множество расширений.

**MySQL** – это система управления базами данных, которая поддерживает SQL и является подходящим продуктом для реализации базы данных сайта.

**phpMyAdmin** – свободное программное обеспечение, работу с MySQL посредством браузера. Оно поддерживает множество функций MySQL, осуществляет импорт и экспорт данных в форматы CSV, SQL, XML, PDF, Word и другие.

**Apache** – веб-сервер, реализующий взаимодействие по протоколу HTTP. Apache является кроссплатформенным программным обеспечением.



**Atom** – бесплатный текстовый редактор с открытым исходным кодом для macOS, Linux, Windows с поддержкой плагинов, написанных на Node.js, и встраиваемых под управлением Git Control.

**Laravel** – бесплатный веб-фреймворк с открытым кодом, предназначенный для разработки с использованием архитектурной модели MVC (Model View Controller — модель-представление-контроллер).

**Bootstrap** – свободный набор инструментов для создания сайтов и веб-приложений. Включает в себя HTML- и CSS-шаблоны оформления для типографики, веб-форм, кнопок, меток, блоков навигации и прочих компонентов веб-интерфейса, включая JavaScript-расширения.

## 2.2 Шаблон проектирования MVC

Фреймворк Laravel использует архитектурную модель MVC (модель-представление-контроллер), которая представляет собой схему разделения 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 (Model) предоставляет данные и реагирует на команды контроллера, изменяя свое состояние.

Представление (View) отвечает за отображение данных модели пользователю, реагируя на изменения модели.

Контроллер (Controller) интерпретирует действия пользователя, оповещая модель о необходимости изменений.

На рисунке 1 изображена схема архитектуры шаблона проектирования MVC.

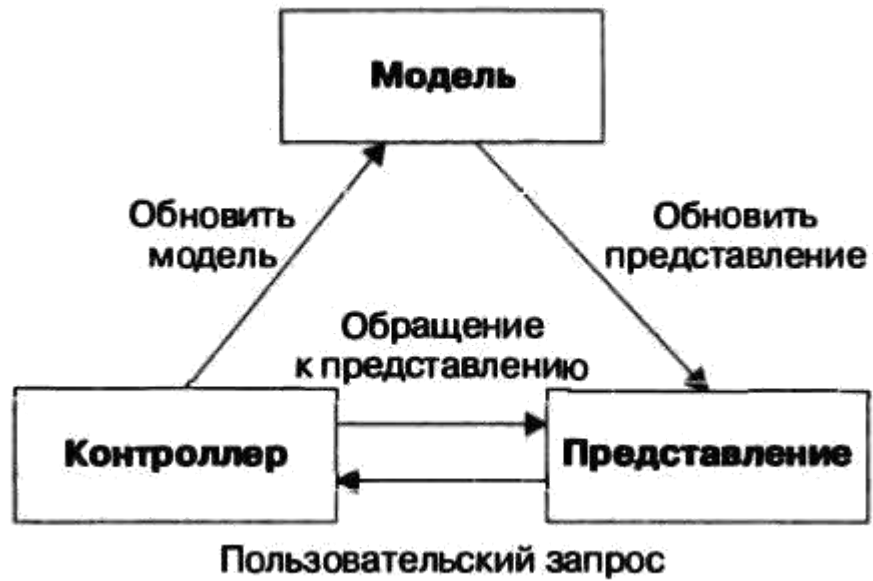


Рисунок 1 – Архитектура шаблона MVC

### 2.3 Проектирование диаграммы прецедентов

Диаграмма прецедентов удобна для проработки функционала программы и подготовки к дальнейшей реализации программного продукта.

На рисунке 2 изображена диаграмма прецедентов, которая отражает отношения между актерами и прецедентами.

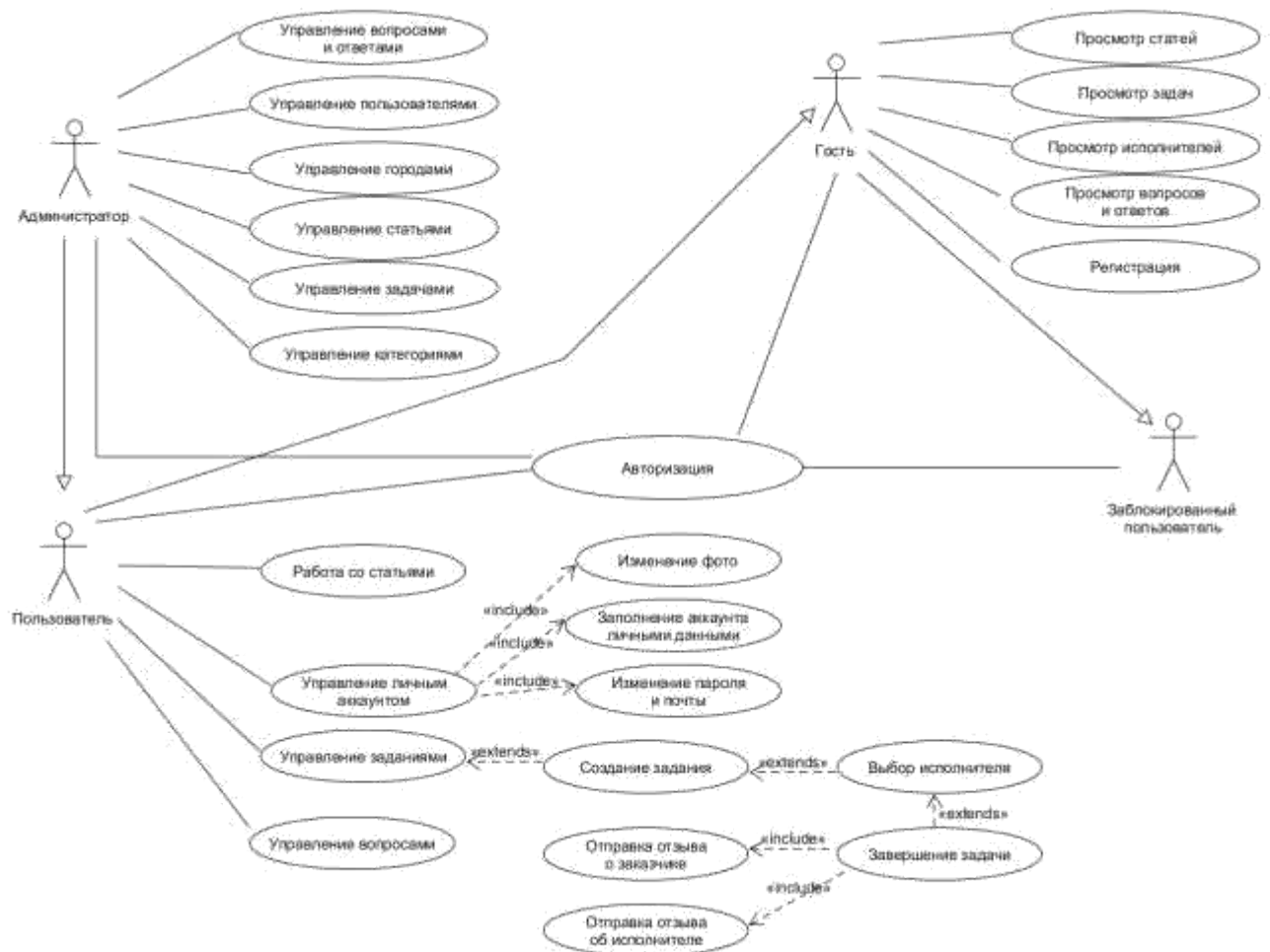


Рисунок 2 – Диаграмма прецедентов

Диаграмма прецедентов состоит из блоков (актеров): Пользователь, Администратор, Гость и Заблокированный пользователь и прецедентов (действий, которые могут совершать актеры)

Заблокированный пользователь может только авторизоваться на сайте.

Гость имеет возможность просматривать задачи, вопросы и ответы пользователей, статьи, исполнителей, а также может зарегистрироваться и авторизоваться на сайте.

Пользователь обладает такими же возможностями, как и гость, а также имеет возможность управления задачами (создавать задачу, устанавливать исполнителя для задачи, завершать задачу и отправлять отзыв о заказчике/исполнителе)

Администратор имеет возможности пользователя, а также может управлять аккаунтами других пользователей, и полный доступ к базе данных (добавление, редактирование, удаление данных).

## 2.4 Проектирование базы данных

На рисунке 3 приведена схема базы данных дипломного проекта, хранящаяся в СУБД MySQL. На ней изображено множество таблиц, которые взаимосвязаны друг с другом. Каждая из них отвечает за хранение определенной информации, необходимой для корректного отображения данных.

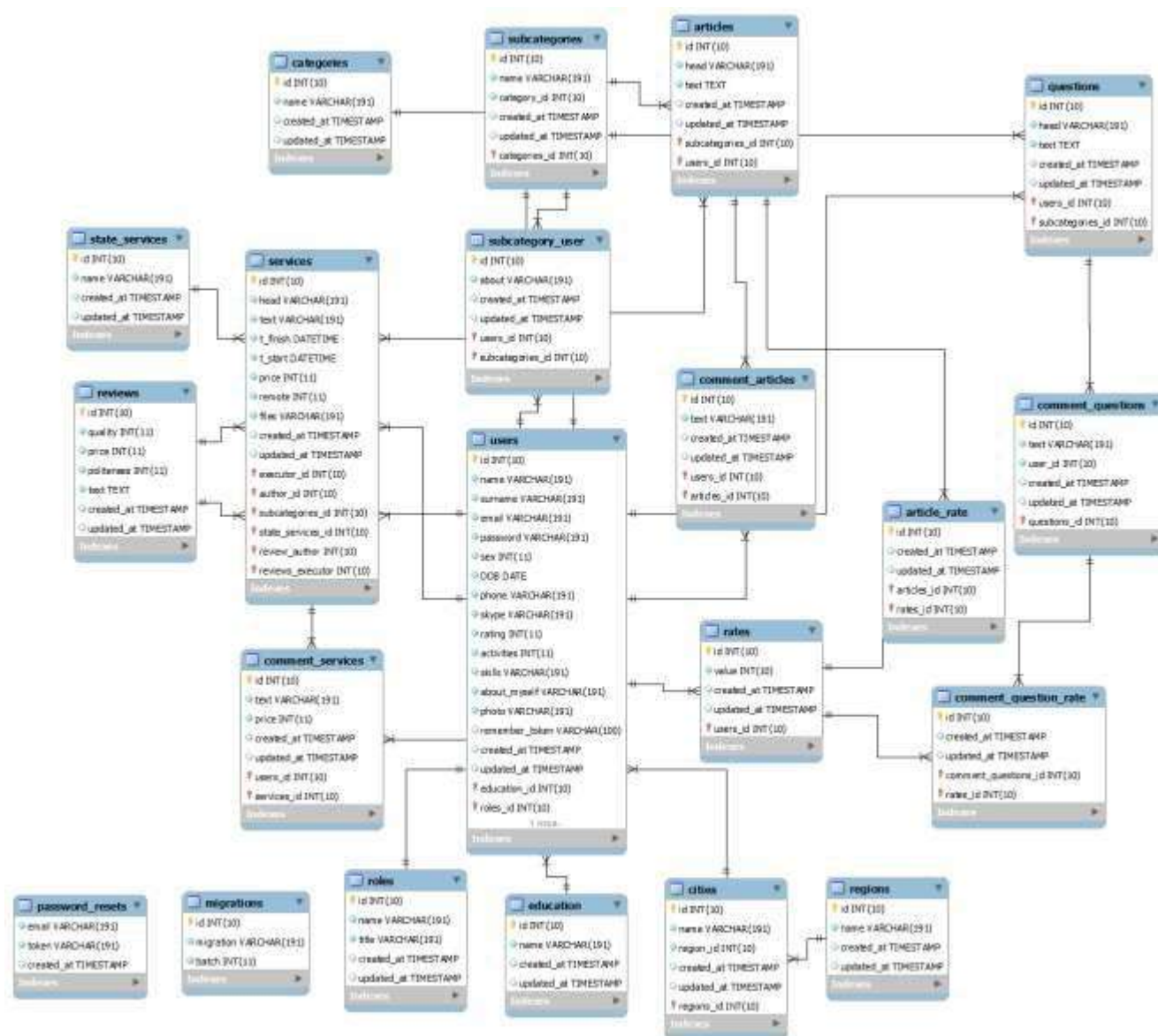


Рисунок 3 – Схема данных базы данных

## 2.5 Структура и описание работы приложения

Стартовая главная страница приложения содержит панель для входа на сайт и регистрации, а также меню, обеспечивающее переход ко всем основным разделам сайта. На главной странице представлены примеры категорий, исполнителей и статей. Главная страница представлена на рисунке 4.

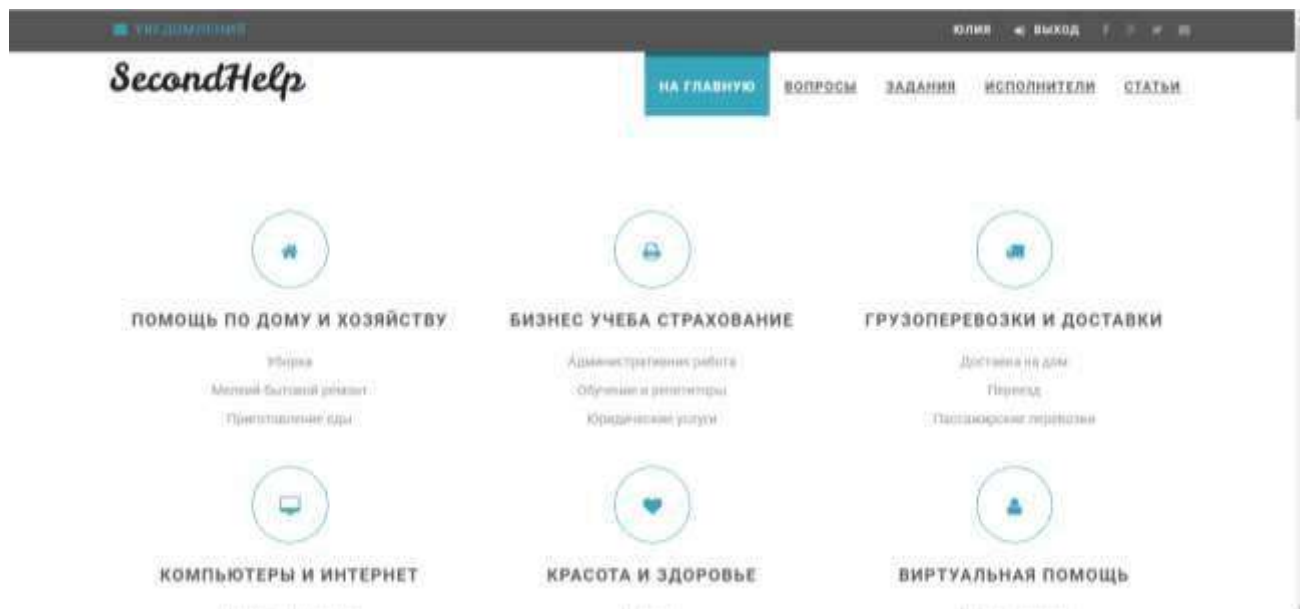


Рисунок 4 – Главная страница приложения

Пункт меню «На главную» осуществляет переход на главную страницу. Также переход на данную страницу осуществляется нажатием на логотип сайта.

Пункт меню «Вопросы» переносит на страницу со списком вопросов, заданных пользователями, отсортированным по убыванию даты. В правой части страницы появляется перечень категорий. Выбор конкретного элемента из списка категорий позволяет просматривать вопросы только выбранной категории. Имеется возможность создания вопроса при нажатии на кнопку «Задать вопрос». На рисунке 5 приведена страница «Вопросы».

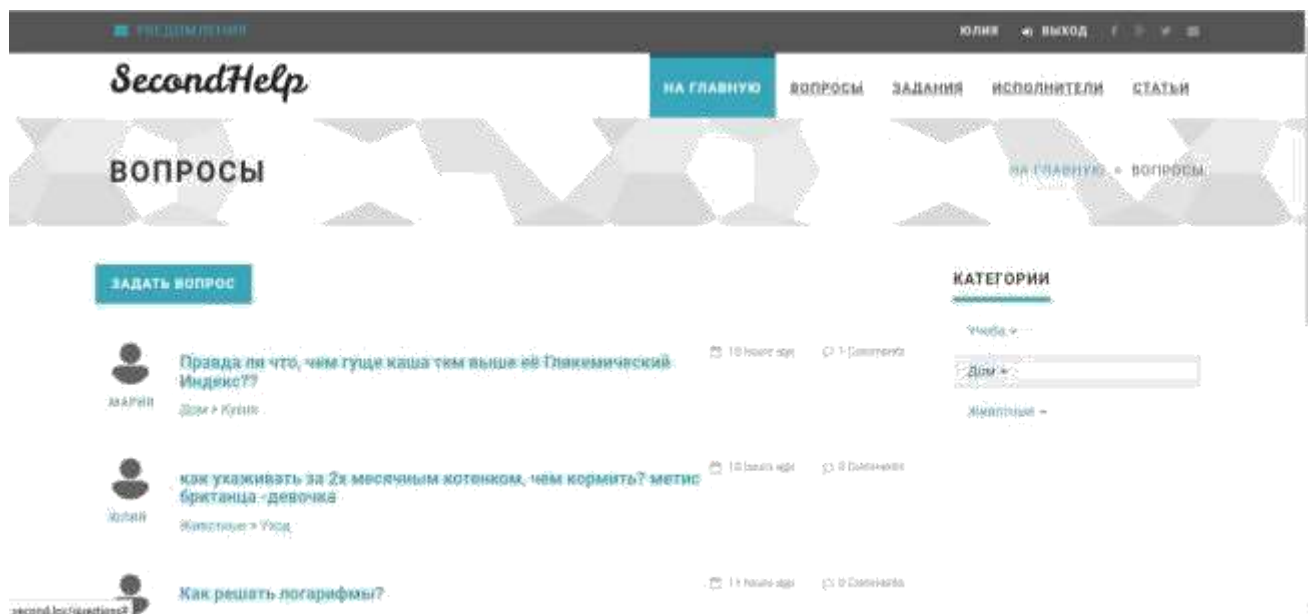


Рисунок 5 – Страница «Вопросы»

При выборе из списка конкретного вопроса осуществляется переход на страницу ответов на данный вопрос. Страница ответов на вопрос представлена на рисунке 6. Каждый ответ может быть оценен другим пользователем путем нажатия на кнопку с большим пальцем вверх. При необходимости оценку можно убрать, нажав на кнопку повторно (большой палец вниз). Данная оценка повлияет на рейтинг исполнителя пользователя.

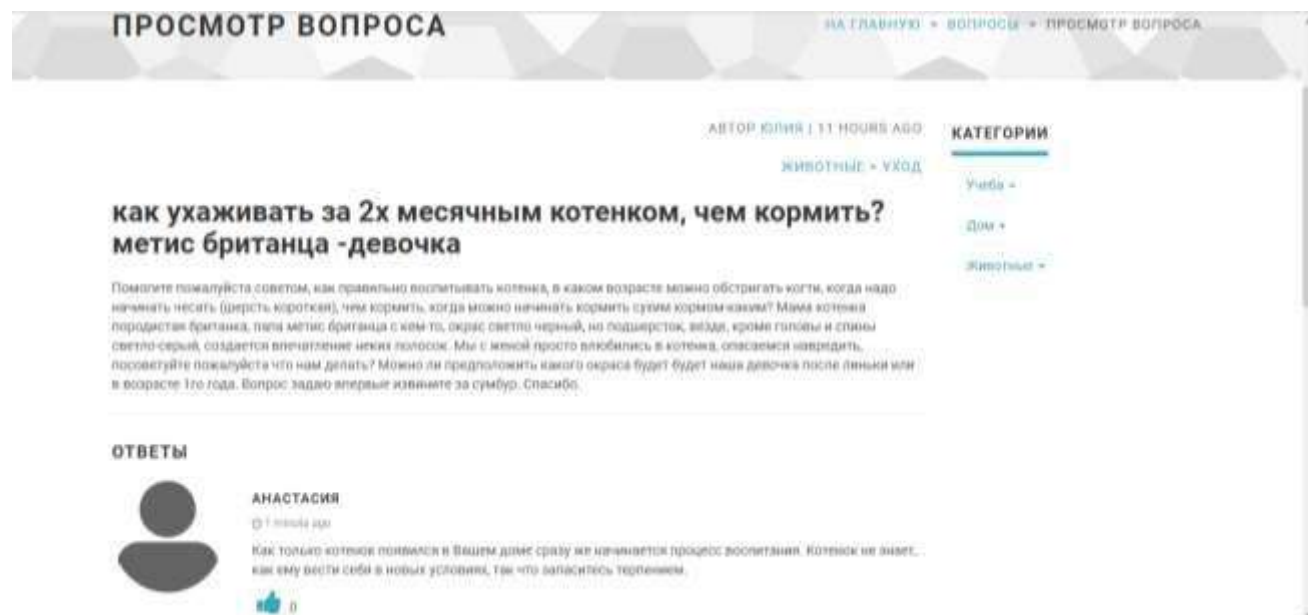


Рисунок 6 – Страница «Ответ на вопрос»

Пункт меню «Задания» осуществляет переход к странице с задачами, которые опубликовали пользователи. Выбор конкретного элемента из списка категорий в правой части окна позволяет просматривать задачи только выбранной категории. Имеется возможность создания задачи при нажатии на кнопку «Создать задание». Страница заданий представлена на рисунке 7.

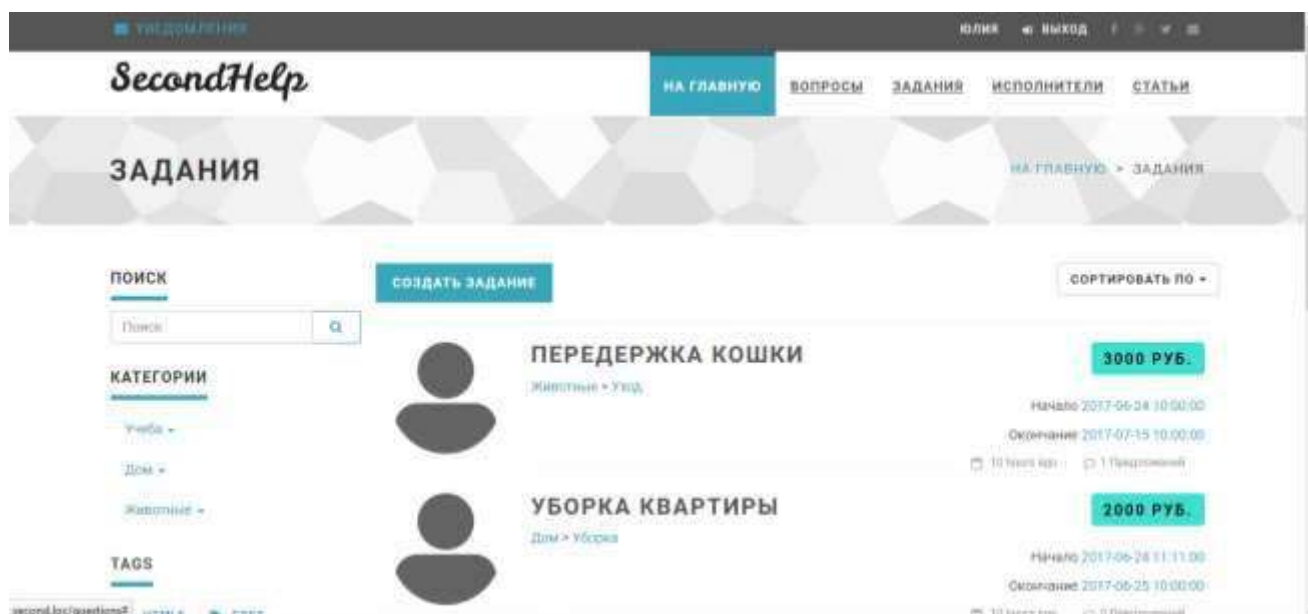


Рисунок 7 – Страница «Задания»

После нажатия на кнопку «Создать задание» переносит на страницу создания задания. На данной странице можно описать задачу, выбрать время начала и окончания задачи и указать желаемую цену.

Страница создания задания представлена на рисунке 8.

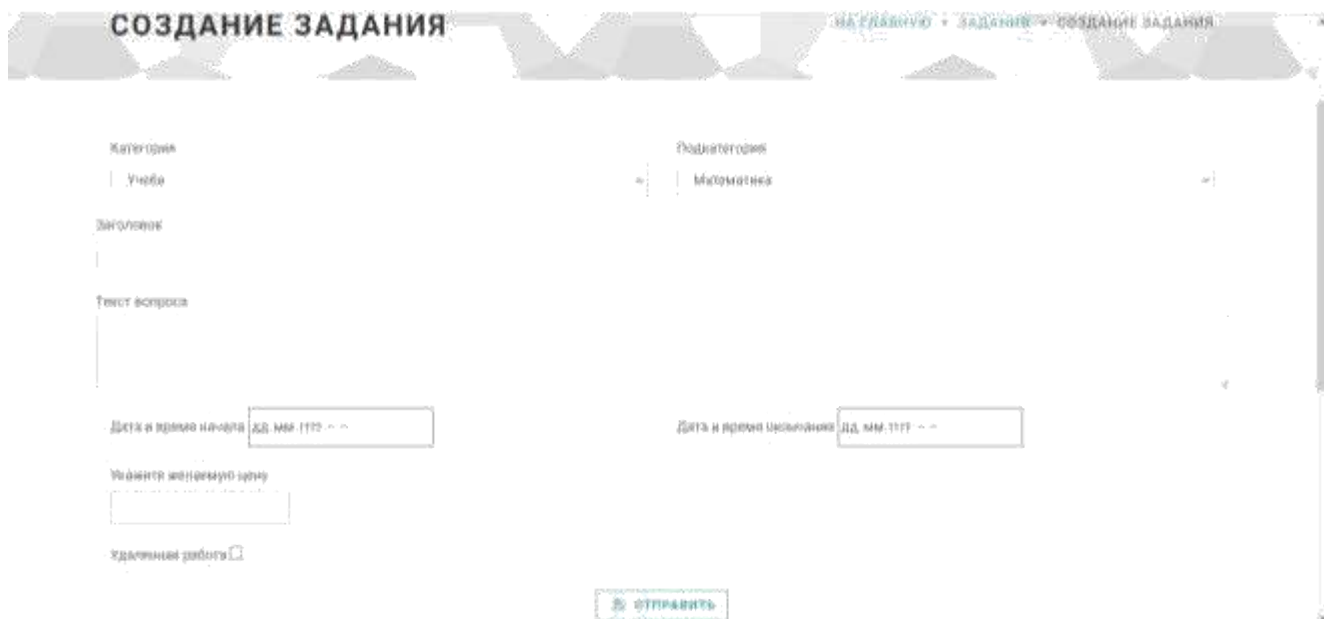


Рисунок 8 – Страница «Создание задания»

При нажатии на заголовок статьи переносит на страницу просмотра задания. При просмотре задачи предложения исполнителей может видеть только заказчик. На рисунке 9 представлена страница, отображающая просмотр задания другого пользователей. На данной странице исполнитель может откликнуться на задание описав условия и желаемую цену за работу.

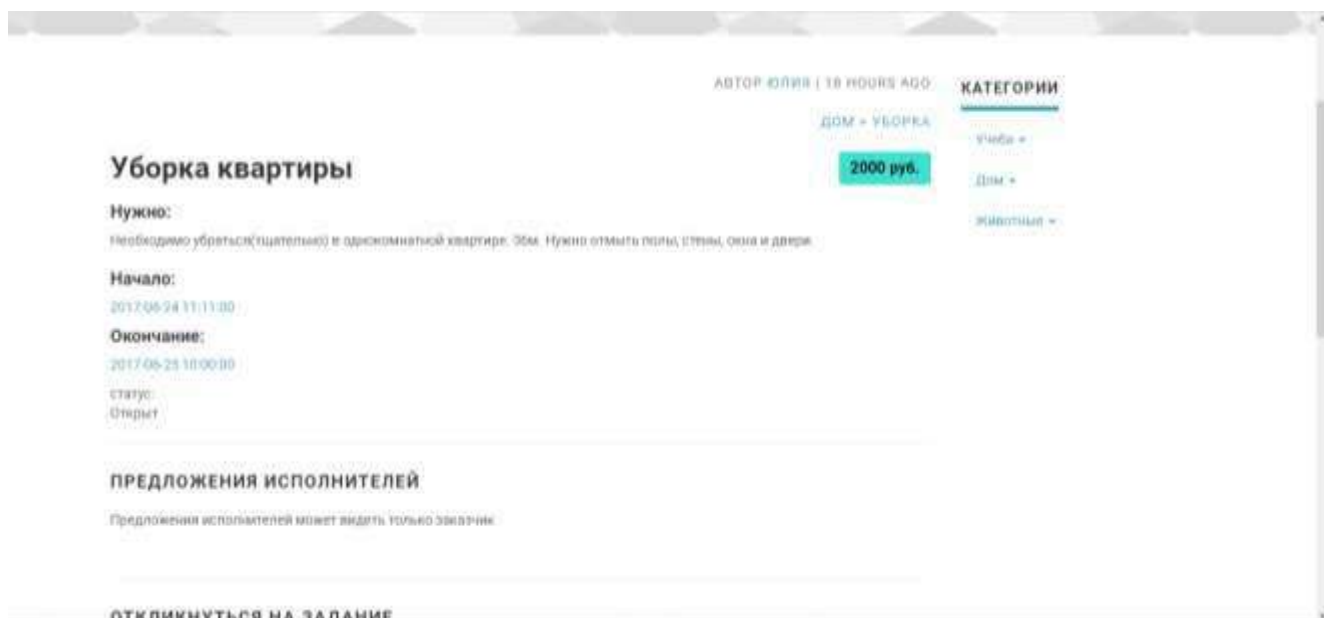


Рисунок 9 – Страница «Просмотр задания»



На рисунке 10 представлена страница просмотра задания заказчиком. На данной странице заказчик может выбрать исполнителя задания из предложений пользователей или закрыть задание.

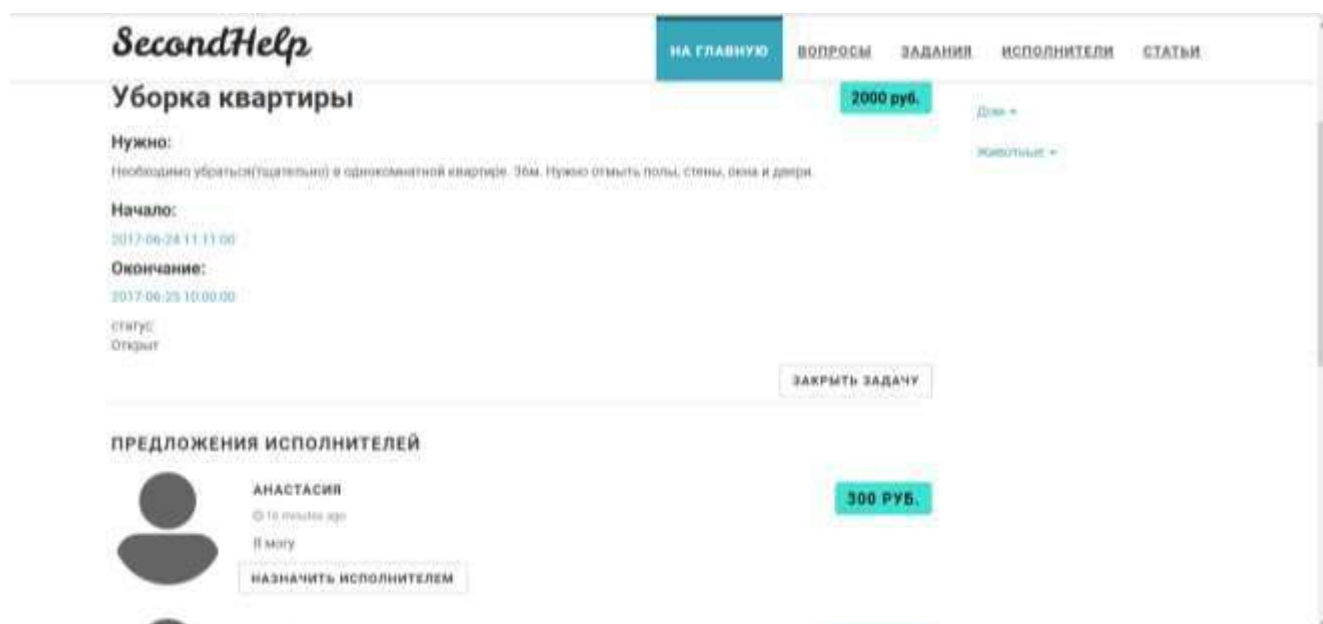


Рисунок 10 – Страница «Просмотр задания»

Пункт меню «Исполнители» переводит на страницу, содержащую перечень всех исполнителей на сайте. В левой части страницы имеется перечень категорий. Изначально страница загружается с исполнителями по всем категориям.

Страница «Исполнители» представлена на рисунке 11.

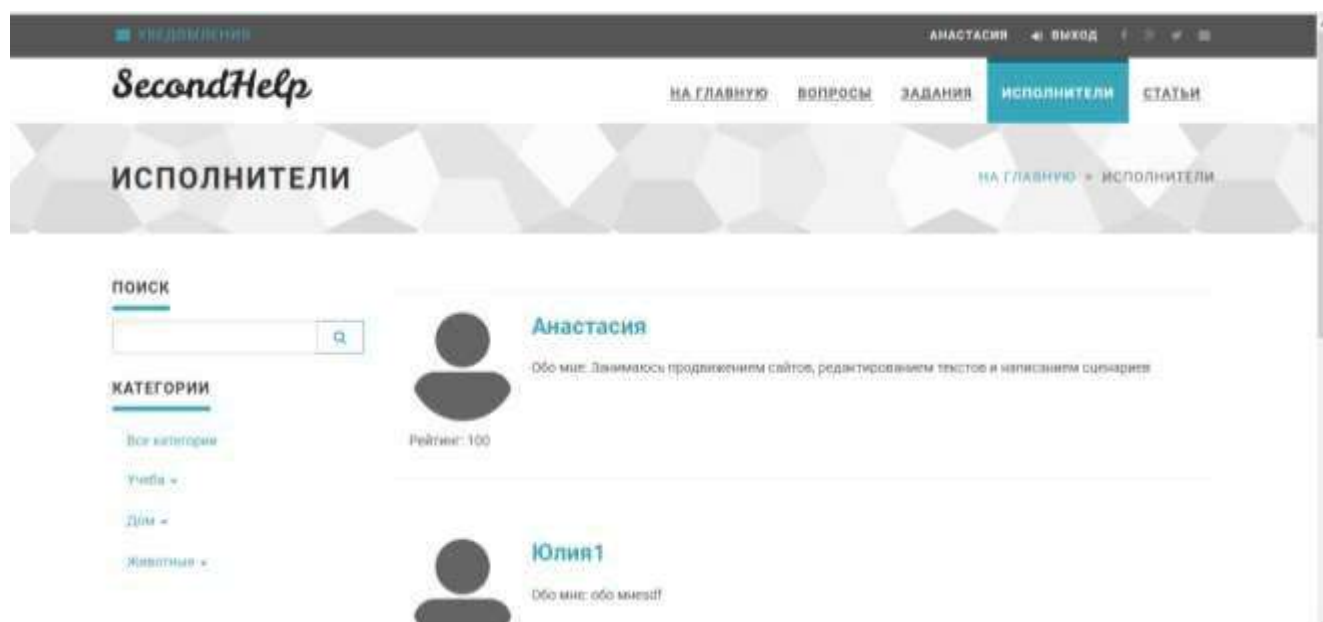


Рисунок 11 – Страница «Исполнители»

Пункт меню «Статьи» переводит на страницу, содержащую список всех статей, опубликованных пользователями. Страница «Статьи» представлена на рисунке 12.

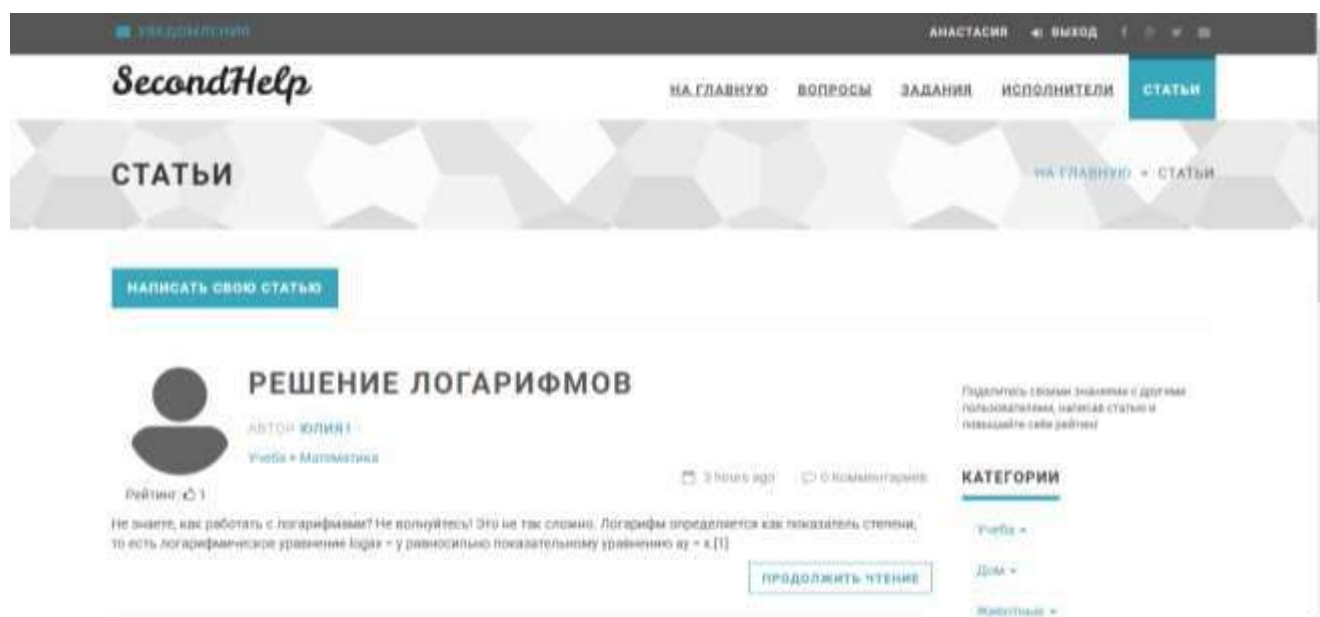


Рисунок 12 – Страница «Статьи»

При нажатии на заголовок статьи или кнопку «Читать далее» переносит на страницу просмотра статьи. Пример статьи представлен на рисунке 13

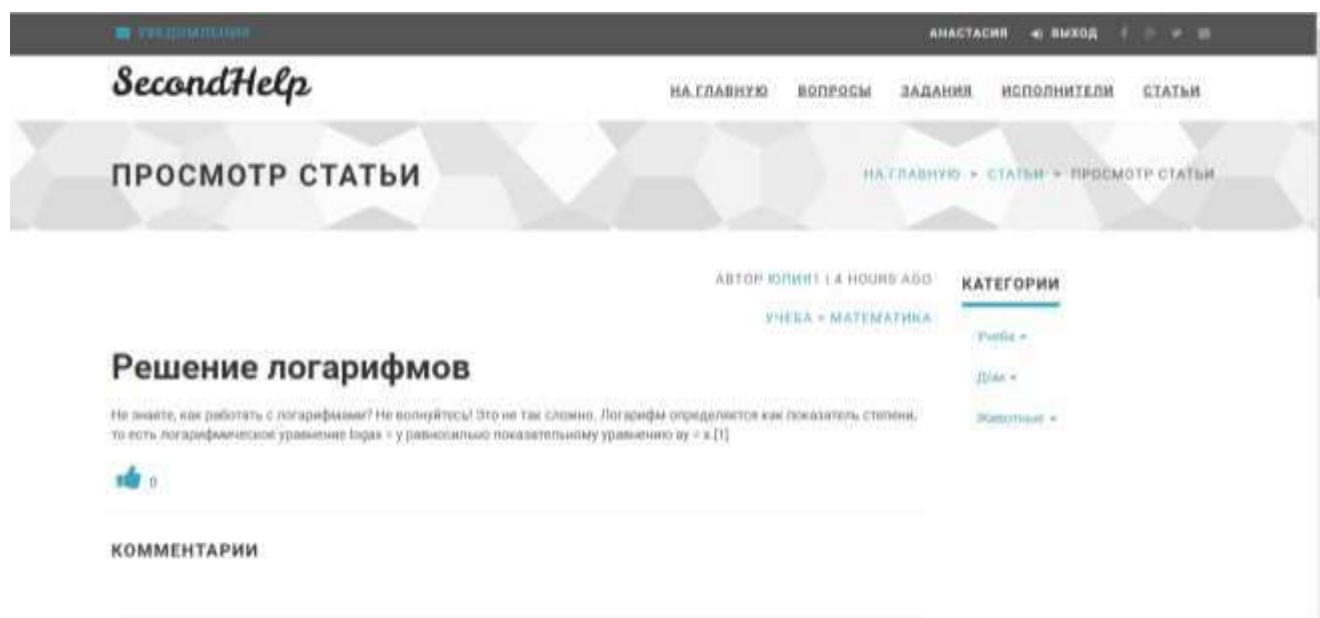


Рисунок 13 – Страница «Просмотр статьи»

На данной странице пользователи могут оценить статью, нажав на кнопку с большим пальцем вверх. Оценка статьи влияет на рейтинг исполнителя автора 18

данной статьи. Также пользователи имеют возможность прокомментировать просматриваемую статью.

На сайте имеется регистрация и вход на сайт через соответствующие страницы. Пример страницы регистрации представлен на рисунке 14. Для регистрации необходимо ввести свое имя, e-mail и пароль.

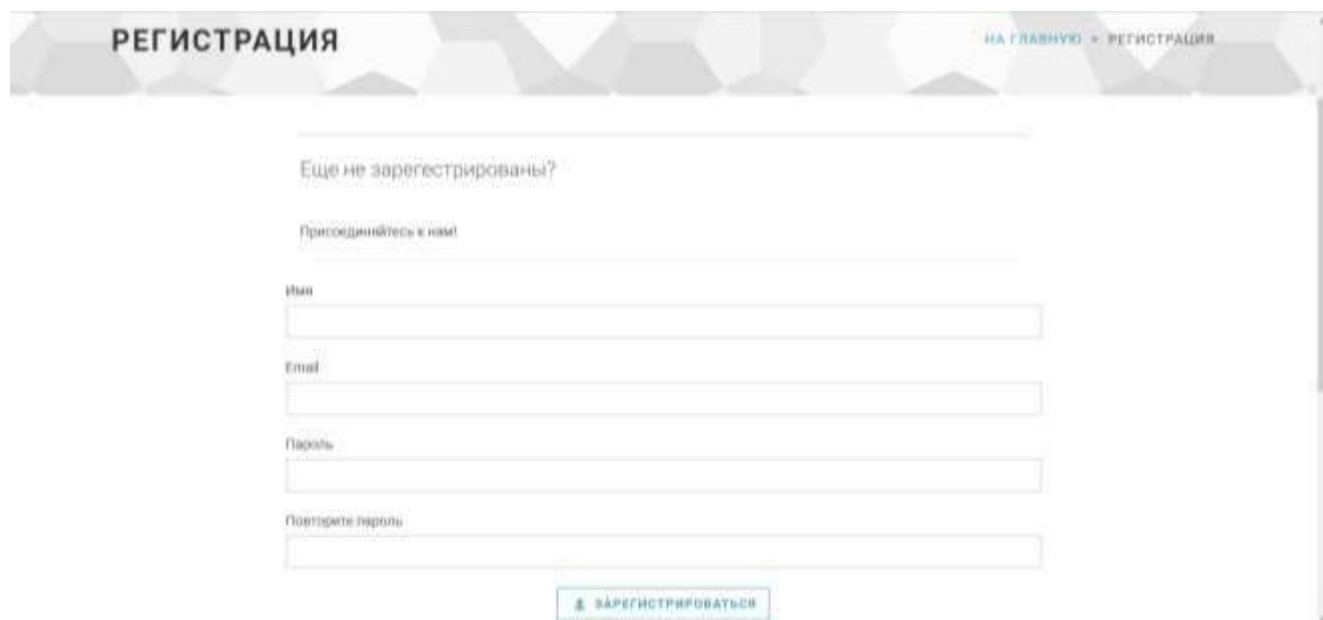


Рисунок 14 – Страница «Регистрация»

Страница авторизации представлена на рисунке 15.

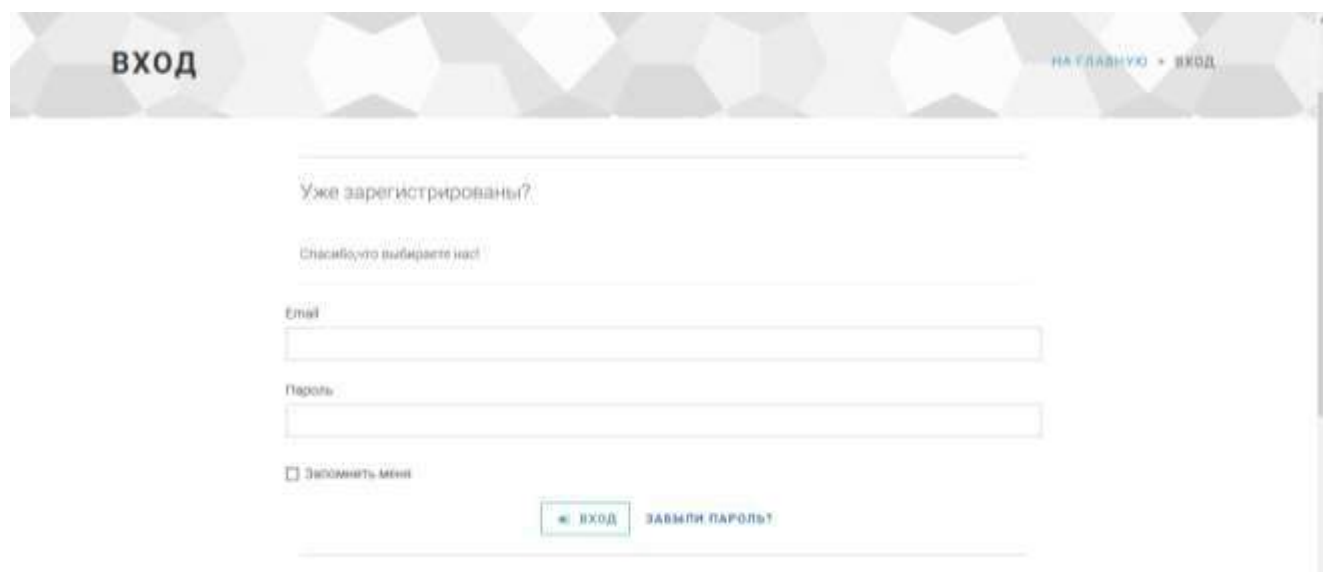


Рисунок 15 – Страница «Авторизация»

Авторизованный пользователь может просматривать и редактировать свой аккаунт на странице «Мой аккаунт», которая представлена на рисунке 16.

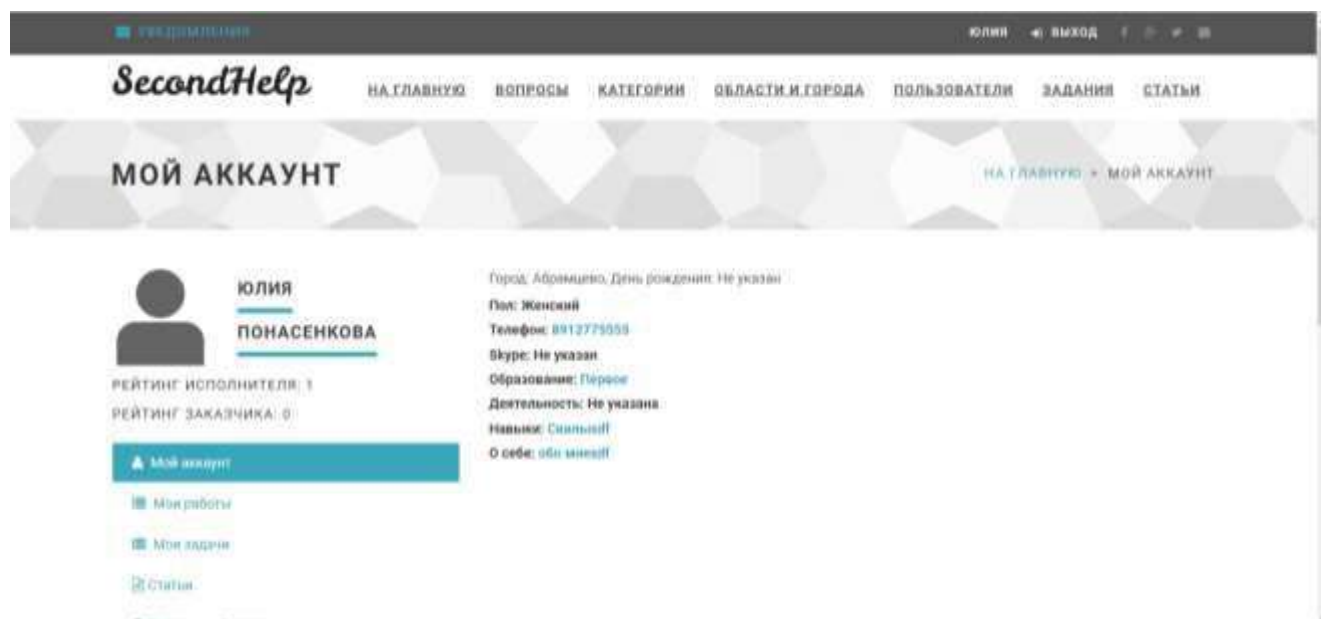


Рисунок 16 – Страница «Мой аккаунт»

Слева находится фотография, имя и фамилия, информация о рейтинге пользователя, а также меню. При нажатии на пункт меню «Мои работы» переносит на страницу с перечнем задач, в которых пользователь выступает в роли исполнителя. Пункт меню «Мои задачи» переносит на страницу со списком всех задач, которые создал пользователь. Пункт меню «Статьи» переносит на страницу со списком всех статей, которые опубликовал пользователь. Пункт «Вопросы и ответы» отвечает за вопросы, которые задал пользователь и ответы, на вопросы других пользователей.

Пункт меню «Отзывы» переносит на страницу со списком отзывов заказчиков и исполнителей о пользователе. Отзывы заказчиков влияют на рейтинг исполнителя, а отзывы исполнителей влияют на рейтинг заказчика пользователя соответственно.

Пункт меню «Настройки» переносит на страницу, содержащую поля для заполнения профиля пользователя и смены пароля. Страница настройки аккаунта представлена на рисунке 17.

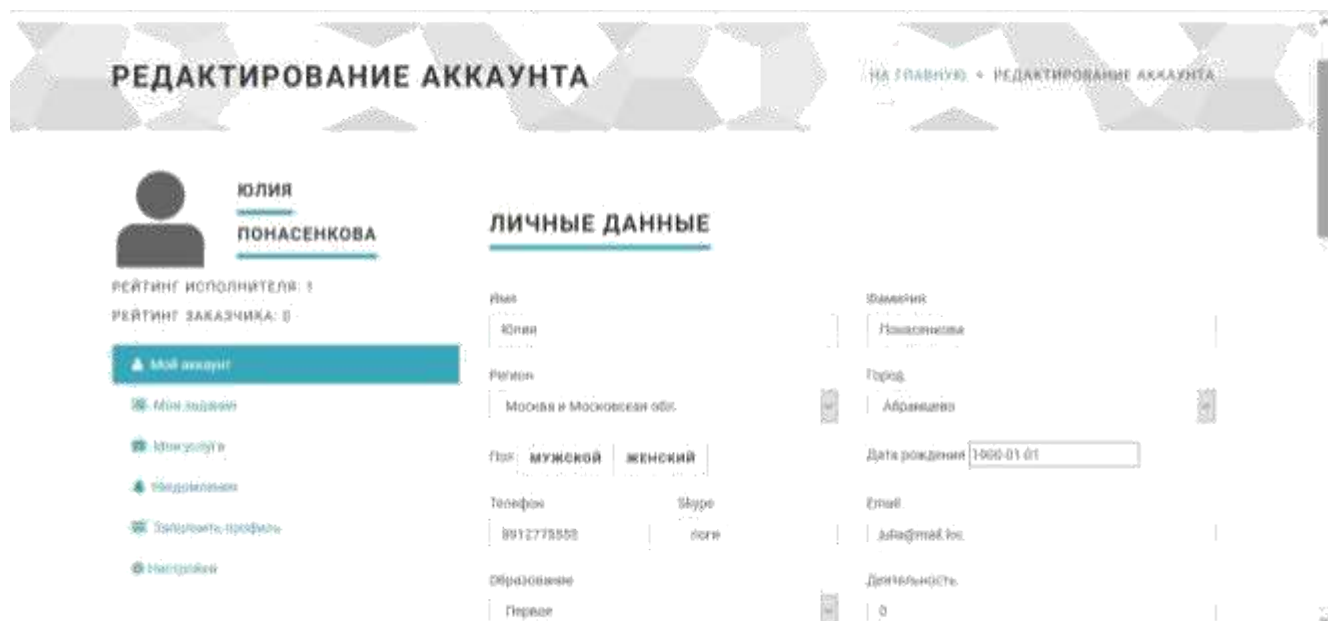


Рисунок 17 – Страница «Настройки»

### Вывод по второму разделу

Таким образом, спроектировано веб-приложение, обозначен его основной функционал и пользователи, которые могут задействовать его в работе. Определены программные инструменты, с помощью которых реализован дипломный проект, обоснована их актуальность и необходимость.

## 3 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

### 3.1 Базовая структура проекта

Проект состоит из основных директорий.

1. App/Database/Migrations — содержит файлы миграций баз данных.
2. App/Database/Seeds — хранит файлы с классами, реализующие заполнение таблицы базы данных тестовыми значениями.
3. App/Http/Controllers — содержит контроллеры проекта.
4. App/Http/Middleware — здесь располагаются посредники проекта.
5. App/routes/web.php — файл роутинга (маршрутов пользователя).
6. App/resources/views — хранит представления, содержащие HTML-код, передаваемый приложением.
7. App/Providers — данная директория хранит провайдеров проекта.
8. App/Classes — директория, которая хранит в себе классы (зачастую хелперы), которые в последствии подлежат регистрации в сервис-контейнере(необязательная).
9. App/Facades — директория содержит фасады проекта (необязательная).

### 3.2 Migrations

Это директория хранит миграции базы данных. Для каждой таблицы создается отдельный файл миграции, в котором описываются все поля для ввода информации.

Таблица «Пользователи»: Реализация миграции «CreateUsersTable» представлена в приложении А. Имеет поля для ввода имени, фамилии даты рождения, пола, телефона, скайпа, адреса электронной почты, пароля. А также поля для связи с моделями «Город», «Образование», «Роли», «Рода деятельности». Имеются поля для ввода информации о навыках, информации о себе и фотографии.

Таблица «Роли»: Реализация миграция «CreateRolesTable» представлена в приложении А. Имеет поля для ввода заголовка роли, названия, а также время создания и изменения.

Также созданы миграции для остальных таблиц («Города», «Регионы», «Задачи», «Вопросы», «Комментарии вопросов», «Задачи», «Комментарии задач», «Состояние задачи», «Статьи», «Комментарии статей», «Отзывы», «Рейтинг», «Рейтинги статей», «Рейтинги Комментариев», «Категории», «Подкатегории», «Образование») аналогично вышеописанным миграциям с необходимыми полями.

### **3.2 Seeders**

Директория, которая хранит файлы с классами, реализующие заполнение таблицы базы данных начальными данными.

Для реализации заполнения таблицы «Роли», реализован Класс «RolesTableSeeder», который представлен в приложении Б.

Также сидеры реализованы для некоторых таблиц: «Категории», «Подкатегории», «Регионы», «Города», «Состояние задачи».

### **3.3 Controllers**

В зависимости от того, по какому url - адресу перейдет пользователь, будет задействован тот или иной контроллер. Контроллеры могут группировать связанную с обработкой HTTP-запросов логику в отдельный класс. Контроллеры хранятся в папке app/Http/Controllers.

Для пользователей реализован контроллер UserController.php, Класс – «UserController» представлен в приложении В, в котором реализованы все стандартные методы:

1. index, возвращает представление users.index;
2. create, возвращает представление для добавление пользователя users.create, содержащее форму для заполнения данных;

3. store, отправляет данные о пользователе в базу данных и перенаправляет на представление users.index ;
4. show, возвращает представление для просмотра информации о пользователе users.show;
5. edit, возвращает представление users.edit, на котором отображается форма для редактирования данных и перенаправляет на представление users.show;
6. save, сохраняет отредактированные данные в базу данных;
7. destroy, удаляет данные о пользователе из базы данных.

Имеется функция проверки авторизации пользователя.

Для остальных элементов (категорий, подкатегорий, городов, статей, задач, вопросов, комментариев к статьям, вопросам и задачам) в созданных контроллерах используются аналогичные методы.

Статьи: контроллер ArticleController.php, Класс – «ArticleController» имеется функция проверки авторизации пользователя, добавление, удаление и редактирование доступно только авторизованному пользователю.

Категории: контроллер CategoryController.php, Класс – «CategoryController» имеется функция проверки авторизации пользователя, доступно только администратору.

Подкатегории: контроллер SubcategoryController.php, Класс – «SubcategoryController» имеется функция проверки авторизации пользователя, доступно только администратору.

Города: контроллер CityController.php, Класс – «CityController» имеется функция проверки авторизации пользователя, доступно только администратору.

Комментарии к статьям: контроллер CommentArticleController.php, Класс – «CommentArticleController» имеет только метод store, доступен только авторизованным пользователям.

Комментарии к вопросам: контроллер CommentQuestionController.php, Класс – «CommentQuestionController» имеет только метод store, доступен только авторизованным пользователям.



Вопросы: контроллер `QuestionController.php`, Класс – «`QuestionController`» имеется функция проверки авторизации пользователя, добавление, удаление и редактирование доступно только авторизованному пользователю.

### **3.4 Посредники**

HTTP-посредники предоставляют удобный механизм для фильтрации HTTP-запросов приложения. В приложении используется посредник «`admin`», который проверяет, является ли пользователь администратором. В зависимости от результата проверки открывается доступ к пунктам навигации «Категории», «Города» и другим элементам, которые недоступны обычному пользователю.

Если пользователь не аутентифицирован, посредник перенаправит его на экран входа в систему. Если же пользователь аутентифицирован, посредник позволит запросу пройти далее в приложение.

Реализация класса «`admin`» приведена в приложении Г.

### **3.5 Маршрутизация (routes)**

Все маршруты (routes) определены в файле `app/routes/web.php`, который автоматически загружается фреймворком. Фрагмент кода представлен в приложении Д. Маршруты сопоставляют url-адрес с методом определённого контроллера для каждого элемента.

### **3.6 Представления (views)**

Представления (views), они же макеты, содержат HTML-код, передаваемый приложением. Это удобный способ разделения бизнес-логики и логики отображения информации. Представления находятся в каталоге `resources/views` и отображают страницу пользователю.

Пример отображения статей представлением `articles/index.blade.php` представлен на рисунке 12.

**Директория Layouts** отвечает за хранение основного макета, который является каркасом для всех страниц и включает в себя основную структуру. Приложение имеет элементы, которые присутствуют на каждой странице сайта (навигация, подвал и так далее), Laravel упрощает использование этих общих функций на всех страницах, используя **макеты Blade**. Это позволяет сформировать грамотную и удобную для работы структуру, а также добиться гибкости в управлении. Blade предоставляет возможность импортировать другие файлы шаблонов предварительно проверяя их наличие. Если файл отсутствует, то процесс формирования страницы не будет прерван и не возникнет проблем отображения страницы.

### **3.7 Провайдеры(providers)**

В проекте создан провайдер `RoleServiceProvider.php`, Класс - «`RoleServiceProvider`», через который загружается класс «`Role`» в сервис-контейнер. Реализация класса «`RoleServiceProvider`» представлена в приложении Е.

### **3.8 Фасады**

Фасады предоставляют «статический» интерфейс к классам, доступным в сервис-контейнере.

В проекте реализуется единственный фасад `Role.php`. Реализация класса «`Role`» представлена в приложении Ж.

### **3.9 Классы**

В проекте реализуется единственный класс `Role.php`, в котором реализуется логика проверки пользователя на принадлежность к одной из ролей.

Реализация класса «`Role`» (Приложение Ж).

### **Вывод по третьему разделу**

Реализованы модули управления основными элементами веб-приложения (статьи, вопросы, задачи, пользователи, категории, подкатегории и т.д.), которые формируют основной функционал сайта.

## **4 РАСЧЕТ ЭКОНОМИЧЕСКИХ ПОКАЗАТЕЛЕЙ**

### **4.1 Внедрение**

Для внедрения сайта понадобится хостинг с полным доступом по ssh и зарегистрированное доменное имя «ru». Средняя цена на виртуальный сервер (VDS) со минимальными характеристиками составляет около 400 рублей в месяц. Доменное имя в «ru» стоит 100-150 рублей в год. Таким образом, средние затраты на внедрение в год составят порядка 500-550 рублей.

### **4.2 Обслуживание**

Для дальнейшего обслуживания сайта (исправление ошибок, расширение функционала, дальнейшее развитие сайта, техподдержка пользователей) необходимо задействовать персонал. Средняя цена работы программиста в час составляет 300 руб. в час, консультанта – 150 рублей в час. Следовательно, обслуживание и поддержка сайта составит около 15000 рублей в неделю (по 8 часов в день и 5 дней в неделю).

### **Вывод по четвертому разделу**

Таким образом, произведен расчет основных затрат, включающих в себя расходы на внедрение и обслуживание сайта.

## **ЗАКЛЮЧЕНИЕ**

В процессе выполнения выпускной квалификационной работы поставлены цели и задачи, обоснована актуальность и необходимость выбранной темы. Проведен анализ предметной области. Проведен сравнительный анализ аналогов на рынке.

Обоснован выбор инструментов для разработки проекта. Спроектирована и реализована база данных системы. Спроектирован функционал и реализован в модульной иерархии. Приложение протестировано на возможные ошибки в коде и интерфейсе.

Произведен расчет затрат на внедрение и дальнейшее обслуживание программного продукта.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

- 1 Документация по Atom – <https://atom.io/docs>.
- 2 Документация по Laravel 5.4 – <https://laravel.com/docs/5.4>
- 3 Документация по PHP – <http://php.net/docs.php>
- 4 Документация по Bootstrap – <http://getbootstrap.com/>
- 5 Документация по Apache2 – <http://httpd.apache.org/docs/>
- 6 Документация по Composer – <https://getcomposer.org/doc/>