Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Факультет информационных технологий

Кафедра прикладной математики

|  |  |
| --- | --- |
| Отчет защищен с оценкой |  |
|  | А. В. Астахова |
| «7» июня 2020 г. | |

**ОТЧЕТ**

О преддипломной практике

По теме «Разработка web-интерфейса для анализа учебных материалов с учетом их внутренней и внешней совместимости»

в Алтайском государственном техническом университете

им. И.И. Ползунова

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр. ПИ-61 |  | | В.Р. Оверченко |
|  | Подпись | | И.О. Ф |
| Руководитель от профильной организации (руководитель ВКР) | Доцент, к.э.н. |  | А.Ю. Андреева |
|  | Должность | Подпись | И.О. Ф |
| Руководитель от университета | доцент, к.э.н. | | А.В.Астахова |

Должность, ученая степень

2020

Министерство науки и высшего образования Российской Федерации  
федеральное государственное бюджетное образовательное учреждение высшего образования «Алтайский государственный технический университет им. И.И. Ползунова»

Кафедра прикладной математики

**Индивидуальное задание**

на производственную (преддипломную) практику

студенту 4 курса группы ПИ-61 Оверченко В.Р.

**Профильная организация**: «Алтайский государственный технический университет им. И.И. Ползунова»

**Сроки практики:**26.05.2020 г. – 08.06.2020 г.

**Тема**: Разработка web-интерфейса для анализа учебных материалов с учетом их внутренней и внешней совместимости.

**Рабочий график (план) проведения практики**

|  |  |  |  |
| --- | --- | --- | --- |
| **№ п/п** | **Содержание раздела (этапа) практики** | **Сроки выполне-ния** | **Планируемые результаты практики** |
| 1 | Анализ предметной области | 26-28.05.2020 | 1. **Формирование компетенций**:   Способность к самоорганизации и самообразованию (**ОК-7**)  Владение основными концепциями, принципами, теориями и фактами, связанными с информатикой (**ОПК-1**)  Готовность применять основные методы и инструменты разработки программного обеспечения (**ПК-1**)  Владение навыками использования операционных систем, сетевых технологий, средств разработки программного интерфейса, применения языков и методов формальных спецификаций, систем управления базами данных (**ПК-2**)  Владение навыками использования различных технологий разработки программного обеспечения (**ПК-3**)  Способность готовить презентации, оформлять научно-технические отчеты по результатам выполненной работы, публиковать результаты исследований в виде статей и докладов на научно-технических конференциях (**ПК-15**)  Владением навыками чтения, понимания и выделения главной идеи прочитанного исходного кода, документации (**ПК-21**)   1. **Разработка слайдов для предзащиты** 2. **Разработка доклада для предзащиты** |
| 2 | Проектирование состава и структуры программного проекта | 29-31.05.2020 |
| 3 | Реализация программного обеспечения | 01-06.06.2020 |
| 4 | Разработка слайдов для предзащиты | 07.06.2020 |
| 5 | Разработка доклада для предзащиты | 08.06.2020 |

|  |  |  |  |
| --- | --- | --- | --- |
| Руководитель практики от университета |  | 22.05.2020 | Астахова А.В., доцент |
| Руководитель практики от профильной организации (Руководитель ВКР) |  | 21.05.2020 | Андреева А.Ю., доцент |
| Задание принял к исполнению |  | 23.05.2020 | Оверченко В.Р. |
|  | Подпись | Дата |  |
|  |  |  |  |

**Инструктаж по ОТ, ТБ, ПБ, ПВТР**

Инструктаж обучающегося по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка проведен « 26 » мая 2020 г.

Руководитель практики от университета, являющегося профильной организацией

|  |  |  |  |
| --- | --- | --- | --- |
| Астахова А.В., доцент |  | 23.05.2020 |  |

# 

# **Введение**

Многие преподаватели и учителя часто задаются вопросом, какими знаниями должен обладать студент или школьник на данный момент времени по результатам изучения смежных дисциплин и предыдущих курсов. Знакомы обучающемуся те или иные понятия, теоремы, формулы, явления, процессы. Не найдется, пожалуй, ни одной учебной дисциплины, которая не имеет связи с другой. Так, например, неразрывно связаны между собой экономика Алтайского края и география. Для полного понимания особенностей экономики тяжелой промышленности следует знать о месторождениях полезных ископаемых. Таких примером можно приводить очень много. И это касается не только разных курсов. Даже в рамках одного предмета очень важна логически построенная структура информации.

В связи с этим для работников образовательных учреждений возникает соответствующая задача по анализу учебных материалов на совместимость. Ручной анализ может занять очень много времени и не всегда быть полным и достоверным. Именно поэтому появляется потребность в автоматизации данного процесса с предоставлением удобного инструмента для анализа преподавателям, учителям и другим сотрудникам образовательной и научной сферы. Так как необходимость в подобном анализе возникает практически у каждого работника, то целесообразно предоставить соответствующий инструмент на web-платформе.

Целью данной работы является разработка удобного и функционального web-интерфейса для сокращения трудовых затрат работников образования, связанных с анализом учебных материалов на совместимость.

Для исследования были поставлены следующие задачи:

1. Спроектировать сценарий анализа учебных материалов на внутреннюю и внешнюю совместимость;
2. Разработать архитектуру программы;
3. Реализовать программу;
4. Изучить возможность использования готовых модулей для обработки учебных материалов;

# **1 Описание предметной области**

## **1.1 Основные определения**

Система учебных текстов – конечное множество символов, знакообъединений, словоформ, словосочетаний, предложений и их групп, объединенных динамическими и статическими отношениями, которые с необходимостью и достаточностью обусловливают наличие целенаправленного свойства – текстовой совместимости, позволяющей диалектически решать многомерное дихотомическое противоречие между связностью и обособленностью письменной речи на выбранном комплекте вузовских учебников.

Совместимость учебно-научных текстов – междисциплинарная категория, отражающая диалектический тип отношений между связностью и обособленностью, а также и всеми выделенными дополнительными противоречиями, проявляющаяся при максимальной дихотомической мерности сложности, обусловленная задачами высшего образования. []

Другими словами, рассмотрение категории “совместимости вообще” должно осуществляться не как изолированной категории, а в рамках дихотомии *связное - обособленное* (Рисунок 1)*.*

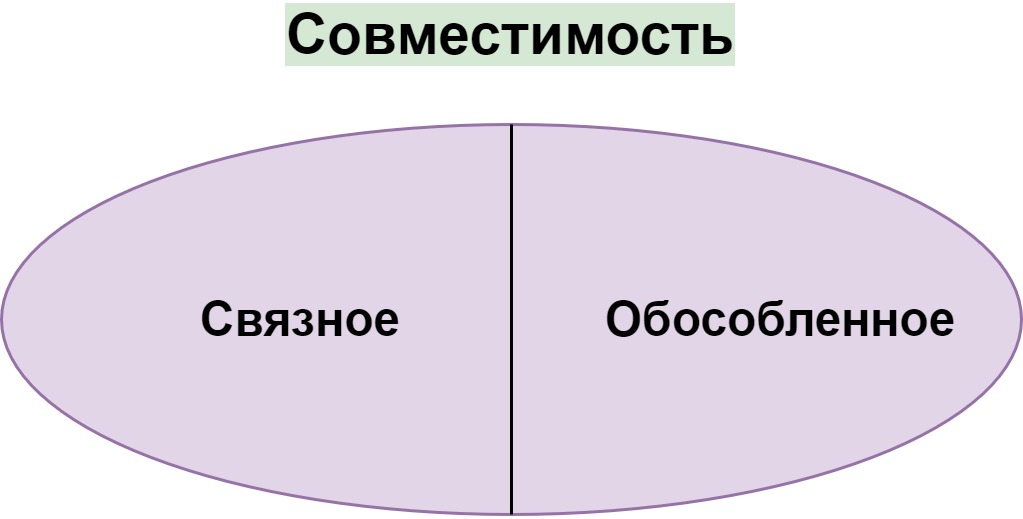


Рисунок 1 - Дихотомия совместимости

Можно выделить совместимость учебных материалов двух типов (Рисунок 2).

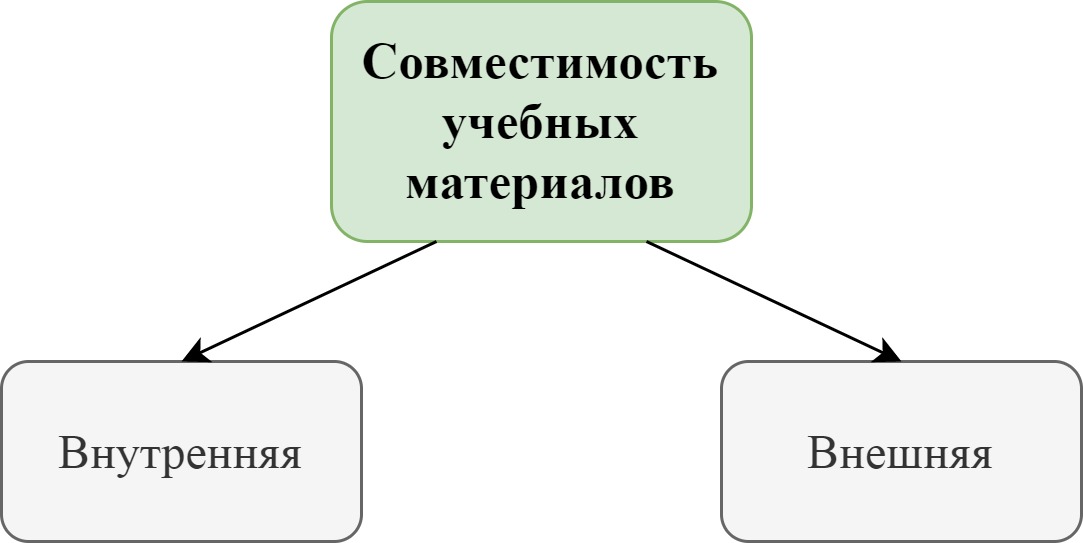


Рисунок 2 - Типы совместимости учебных материалов

Внутренняя совместимость отражает совместимость в пределах одного учебного текста, например, между главами или разделами. Внешняя совместимость - между материалами различных дисциплин.

Мы хотим представить совместимость учебных материалов как некоторую количественную величину, измеряемую в процентах.

Считается, что для внутренней совместимости прирост предметного тезауруса должен составлять не более 20%. Оптимальное значение совместимости учебных материалов смежных курсов должно быть в пределах 75-80%.

Для полного понимания исследуемой области стоит также выделить понятие тезауруса.

Тезаурус — словарь, собрание сведений, корпус или [свод](https://dic.academic.ru/dic.nsf/ruwiki/680867), полномерно охватывающие понятия, определения и [термины](https://dic.academic.ru/dic.nsf/ruwiki/1254724) [специальной](https://dic.academic.ru/dic.nsf/ruwiki/684921) области знаний или сферы деятельности, что должно способствовать правильной [лексической](https://dic.academic.ru/dic.nsf/ruwiki/29259), корпоративной [коммуникации](https://dic.academic.ru/dic.nsf/ruwiki/1526461).

## **1.2 Современные подходы к анализу**

Компьютерное моделирование предоставляет большие возможности для исследований не только в точных науках, но и в некоторых гуманитарных исследованиях, где параметры исследуемых объектов могут быть трудно формализованы. Методы моделирования становятся мощным инструментом в различных областях знаний, в том числе в педагогике, дидактике[[1]](#footnote-1) и смежных областях.

Мы исследовали такой значимый элемент образовательного процесса, как учебная информация и способы ее передачи, отражающие бессознательный спектр ее восприятия и усвоения. Таким образом показатели образовательной информации выступают объектом моделирования. Анализ научных публикаций показал противоречивость подходов, используемых для выбора аналитических показателей и методов моделирования. Это можно объяснить спецификой и многомерной сложностью формализованного объекта.

Для создания модели анализа системы учебных текстов на предмет их совместимости друг с другом необходимо обеспечить согласованность последовательных уровней сложности учебного материала для его последующего преобразования в знания. В работах Рыбаковой Г.Р.[[2]](#footnote-2) и Клочкова В.П.[[3]](#footnote-3) предлагается достичь этой цели путем последовательного обогащения тезауруса. В качестве базового понятия их исследования является фоносемантическое поле, которое понимается как система со связями между ее взаимодействующими элементами—знаковыми структурами и семантическим содержанием. Эти связи трудно формализовать, поскольку они затрагивают семантический аспект передаваемого знания.

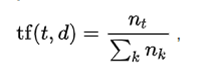
При анализе семантической совместимости учебной информации и ее влияния на бессознательные процессы усвоения и запоминания учебного материала следует учитывать его полевую природу. Фоносемантическое пространство представляет собой единую структуру семантических и символических единиц, сгруппированных вокруг семантического ядра (Рисунок 3). Сила связи между ними оценивается путем сравнения частотности ключевых слов (семантического ядра) и других слов, используемых с ядром и сгруппированных вокруг него. Сила внутренних связей уменьшается от семантического центра к периферии поля, но появляются новые связи, внешние для данного поля: это связи с другими семантическими полями и их пересечениями. Таким образом, в рамках системного подхода фоносемантическое поле можно рассматривать как совокупность элементов со связями и отношениями между ними.



Рисунок 3 - Фоносемантическое поле

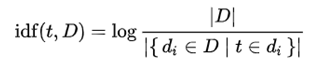
Для определения силы связи между семантическими и символическими единицами можно воспользоваться статической мерой TF-IDF (от англ. TF — term frequency, IDF — inverse document frequency). Данная мера используется для определения важности слова в контексте документа. TF-IDF - произведение TF-меры (отношения числа вхождений слова к общему числу слов документа) и инверсии частоты встречаемости слова в документах корпуса (IDF).

TF-мера определяет важность слова t в пределах отдельного документа d и определяется как

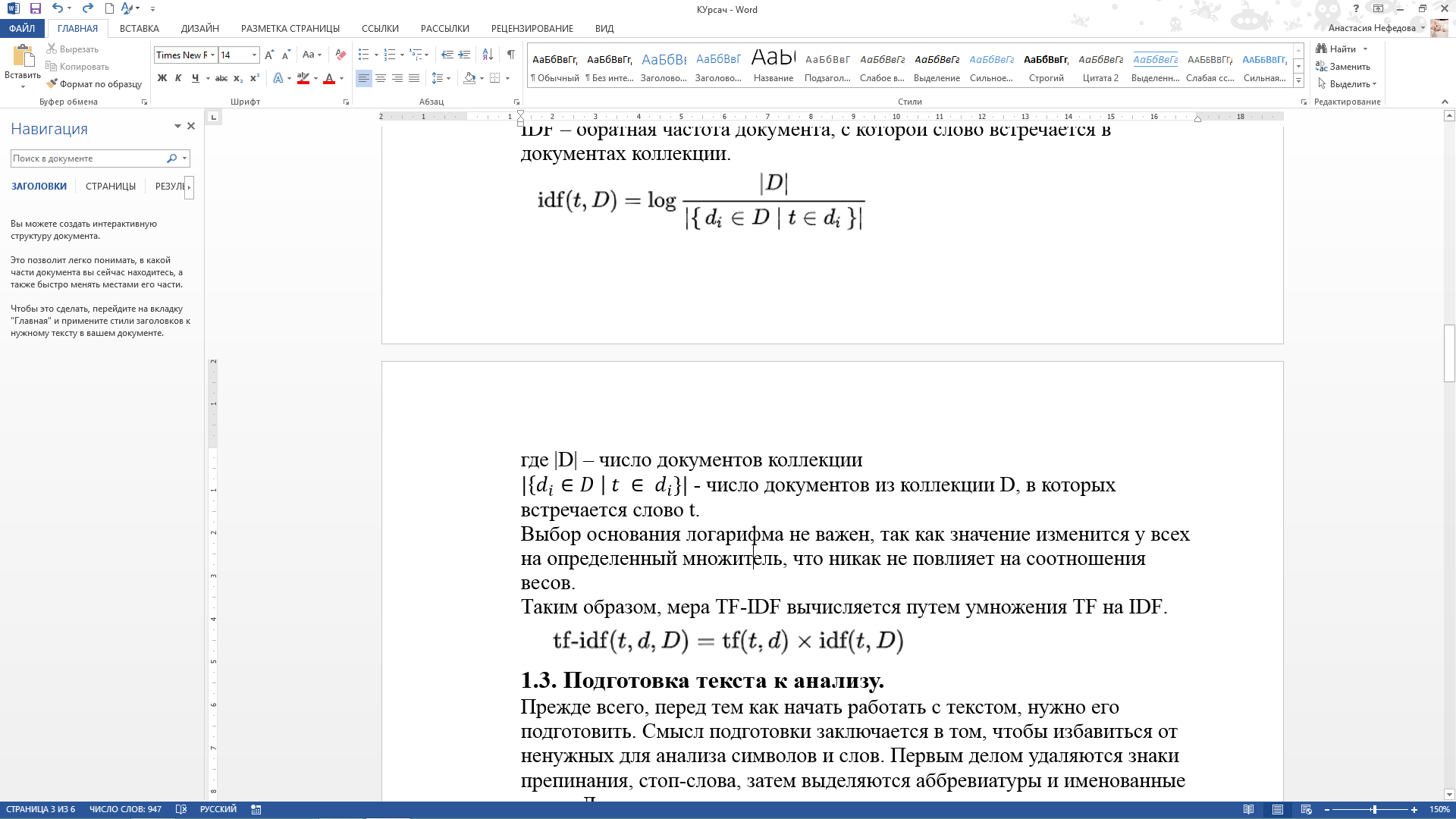


где nt  - это число вхождений слова в документе, а в знаменателе сумма вхождений всех слов в документе.

IDF – обратная частота документа, с которой слово встречается в документах коллекции.



где |D| – число документов коллекции,

** – число документов из коллекции D, в которых встречается слово t.

Выбор основания логарифма не важен, так как значение изменится у всех на определенный множитель, что никак не повлияет на соотношения весов.

Таким образом, мера TF-IDF вычисляется путем умножения TF на IDF.



Стоит отметить, чем больше частота использования слова в текстах, тем ниже будет значение меры IDF (стремится к нулю). Это относится не только к общей лексике (глаголы-связки, служебные части речи), но и к терминам, преобладающим в документе. Одновременно, общая лексика, обозначающая одни и те же ситуации либо действия с разных точек зрения и задающие так называемые конверсивные замены («приводить ⇔ являться следствием»), будут иметь более высокие значения меры IDF. Можно предположить, что наиболее уникальные слова в документе (с наибольшими значениями произведения мер TF и IDF) будут относиться к терминам предметной области документа. Если слово имеет синонимы, упомянутые в этом же документе, значение меры TF для этого слова будет ниже. Как и в случае конверсивных замен, имеем меньшую встречаемость в документах корпуса каждого слова из синонимического ряда и, следовательно, более высокие значения меры IDF по сравнению со случаем отсутствия синонимов у слова.

Следует отметить, что применимость такого подхода к анализу ключевых слов определяется спецификой учебной информации. Эта информация написана в научном стиле, и ее смысл существенно отличается от других видов текстовой информации. Терминологическая плотность - это условие однозначной интерпретации содержания. Кроме того, использование специальной терминологии приводит к высокому сгущению смысла в словах и словосочетаниях, а также значительно увеличивает частоту их употребления

Для того чтобы образовательная информация была адекватно декодирована, необходимо выполнить определенные требования: логика изложенного материала должна соответствовать логике мышления, стоящего за ним, а также логике науки, к которой относится передаваемая информация. Между различными частями учебного текста должны быть логические связи. На уровне учебника такие связи определяются содержанием текста, а на уровне курса или университетской специальности-учебным планом.

Компьютерный анализ текстов основан на том же логическом принципе. Условием формализации является логическое композиционное структурирование информационного массива, а результатом-возможность применения математических операций обработки данных к семантическому содержанию текстов.

Логика сознательного восприятия может быть проанализирована путем построения графа логических связей, вершинами которого будут являться смысловые центры (ключевые слова), с соответствующей матрицей логических связей. Использование аналогичной методики, апробированной И. Б. Моргуновым (2007) для изучения последовательности учебных дисциплин в учебном плане, позволит структурировать учебную информацию путем определения потенциального числа семантических ядер и оценки функции общего числа разрывов логических связей для конкретных текстовых блоков в рамках дисциплин каждого цикла. Это помогает определить оптимальную последовательность изучения материала при минимальном количестве нарушенных логических связей.

Таким образом, приходим к следующему умозаключению - для расчёта совместимости учебных материалов можно опираться на такой алгоритм:

1. Выделение тезауруса ключевых слов для каждого анализируемого учебного материала;
2. Применение к тезаурусам статической меры TF-IDF;
3. Вычисление процентной совместимости материалов на основе полученных данных по формуле:

Совместимость = Количество совпадающих слов предметного тезауруса в первом и втором текстах / Количество слов предметного тезауруса во втором тексте \* 100%.

## **1.3 Обзор существующих решений**

### **1.3.1 Программный продукт “Анализ учебной письменной речи”**

Группой авторов А.Д.Никина, Н.К.Криони и А.В.Филипповой. на основе методов математического программирования была разработана информационная система «анализа учебной письменной речи». Они автоматизировали процесс анализа учебного материала на сложность восприятия с дальнейшей адаптацией обучающимися.

В своей работе они придерживались следующих идей:

* Проблема трудности восприятия учебного текста может рассматриваться в двух аспектах: в аспекте генерации текста и в аспекте восприятия текста;
* Имея объективную оценку сложности учебного текста, преподаватель может оценить время, необходимое студенту на изучение текста, а студент, используя указания на элементы текста, определяющие его сложность, сможет при его изучении обратить особое внимание на данные фрагменты текста;
* Перечень ключевых слов, будучи предъявлен читателю перед чтением текста, способствует лучшему усвоению текста.

В связи с этим созданный программный продукт отвечал следующим функциям:

* автоматическое выделение ключевых слов;
* автоматический поиск параметров сложности текстов и их подсчет;
* диалог с пользователем с целью проверки правильности выделенных параметров;
* сохранение результатов анализа сложности текста и поиска в нем ключевых слов в отдельных файлах;
* возможность работы с документами в форматах doc, txt, rtf.

Обобщенная структура программы представлена на рисунке 4.

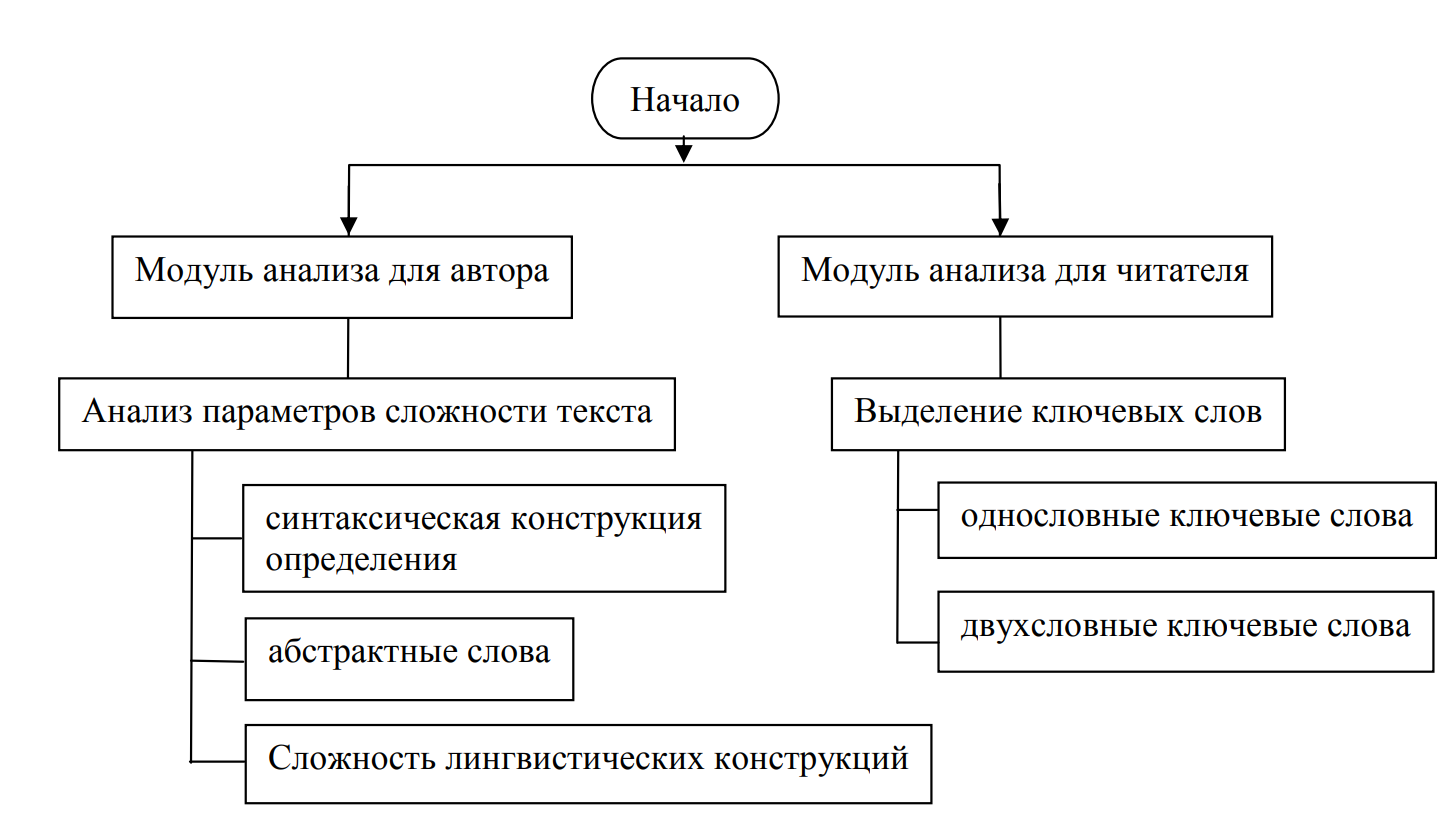


Рисунок 4 - Обобщённая структура программы “Анализ сложности учебных текстов”

В качестве параметров сложности текста выступали:

* синтаксическая конструкция определения;
* абстрактные слова;
* сложность лингвистических конструкций.

Алгоритм работы автора текста представлен на рисунке 5.

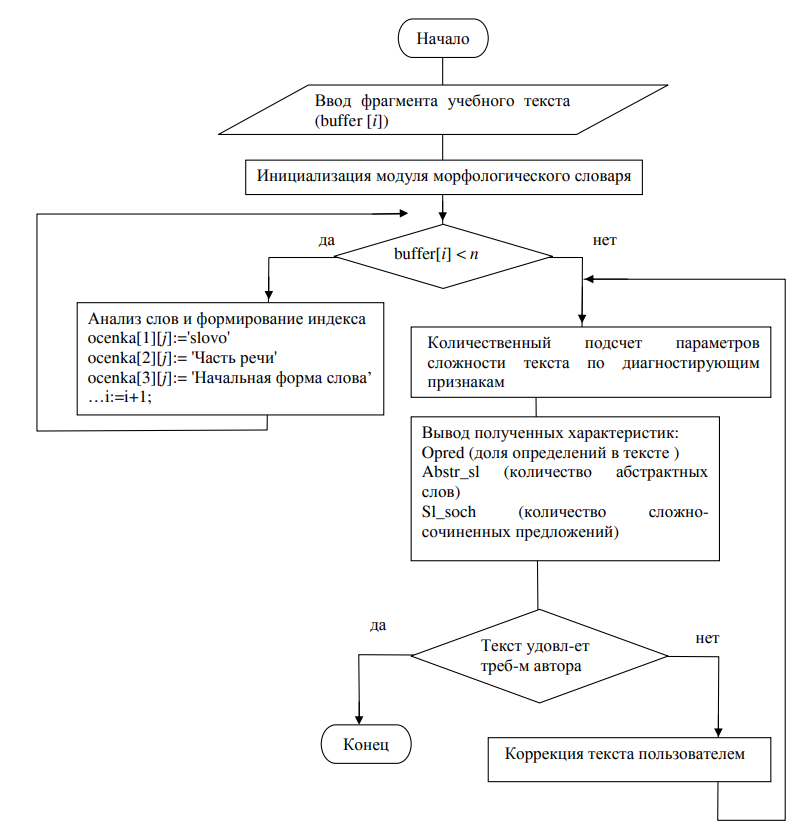


Рисунок 5 - Структура модуля анализа для автора

В качестве инструментального средства проектирования (пакета прикладных программ) был выбран продукт Borland Delphi 6.0. Для работы с естественным языком были использован морфологический словарь, содержащий все словоформы русского языка. В программе анализ наличия параметров сложности в тексте и их поиск производился на основе данных о словоформе каждого отдельного слова из текста. В качестве метода представления и сбора данных о слове был выбран метод индекса, когда для каждого термина записывается набор страниц, где может этот термин находиться. Аналогичный принцип использовался для работы программы, был создан многомерный массив, хранящий набор информации о слове: исходную форму слова, морфологическую часть речи, словоформу, количество гласных, номер нахождения в тексте. Анализ многомерного массива позволяет выявить наличие параметров сложности в тексте, выполнить их подсчет и вывести данные на экран. По результатам первичного анализа выводится таблица количественных характеристик параметров сложности текста, далее автору предлагается отдельно поработать с каждым параметром текста, а именно — просмотреть каждую группу найденных параметров сложности текста. Автор текста может либо оставить в тексте все как есть, либо упростить конструкцию предложения, убрать абстрактное слово, изменить определение и провести повторный анализ.

Интерфейс программы для автора представлен на рисунке 6.

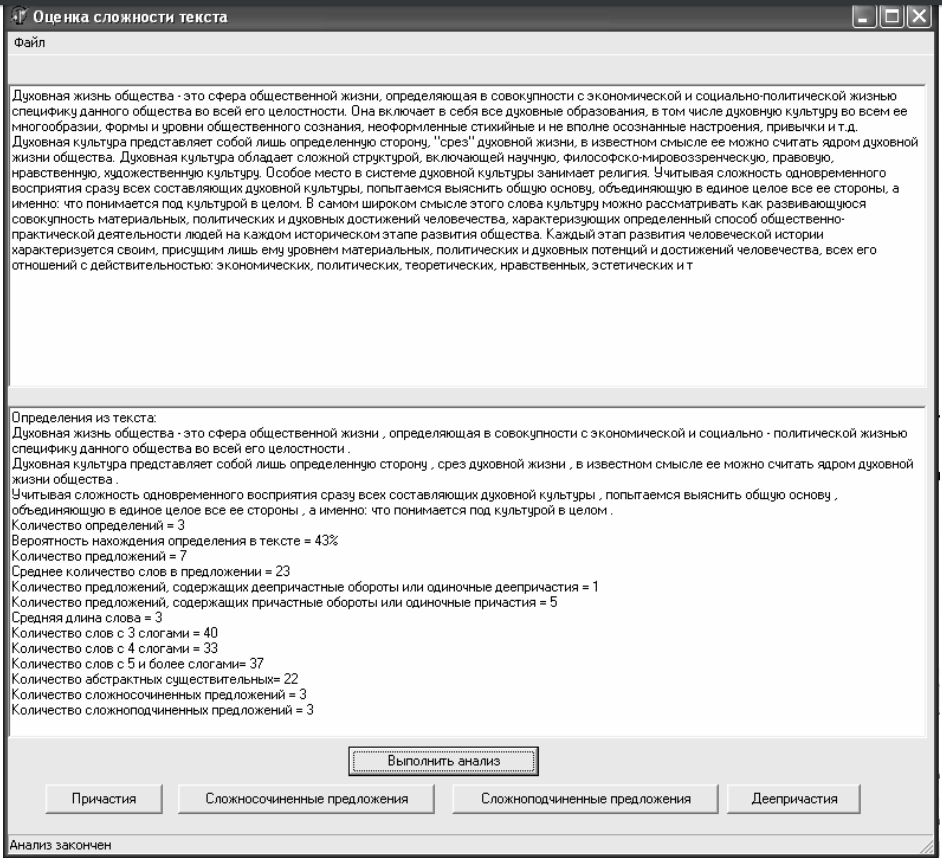


Рисунок 6 - Окно модуля анализа для автора

### **1.3.2 Программный продукт “LightReader”**

Схожая по функционалу программа LightReader представлена в работе И.В.Оборневой “Автоматизированная оценка сложности учебных текстов на основе статических параметров”. Автором с использованием методов математического программирования была разработана методика автоматизированной оценки сложности учебных текстов. При этом использовались только статистические показатели, такие как среднее число слов в предложении, среднее число слогов в слове, число многосложных слов и другие.

Программный продукт LightReader являлся макросом для программного продукта Microsoft Office Word. Программа состоит из двух модулей: модуль для расчёта статических показателей текста и оценки его сложности, и модуль анализа текста в зависимости с заданными условиями относительно возраста учащихся.

Программа может обрабатывать текст не только на русском, но и на английском языке. В результате обработке получаем таблицу значений статических показателей и оценку сложности текста, рассчитанную по формуле Флеша:

K = 206,836 - 60,1 W - 1,3 - S, где

К - оценка трудности текста от 0 до 100,

W - среднее число слогов в слове,

S - среднее количество слов в предложении.

Дизайн продукта соответствует стандартному стилю Windows-приложений. Для демонстрации работоспособности программы использована последняя версия Microsoft Office Word (рисунок 7, рисунок 8).

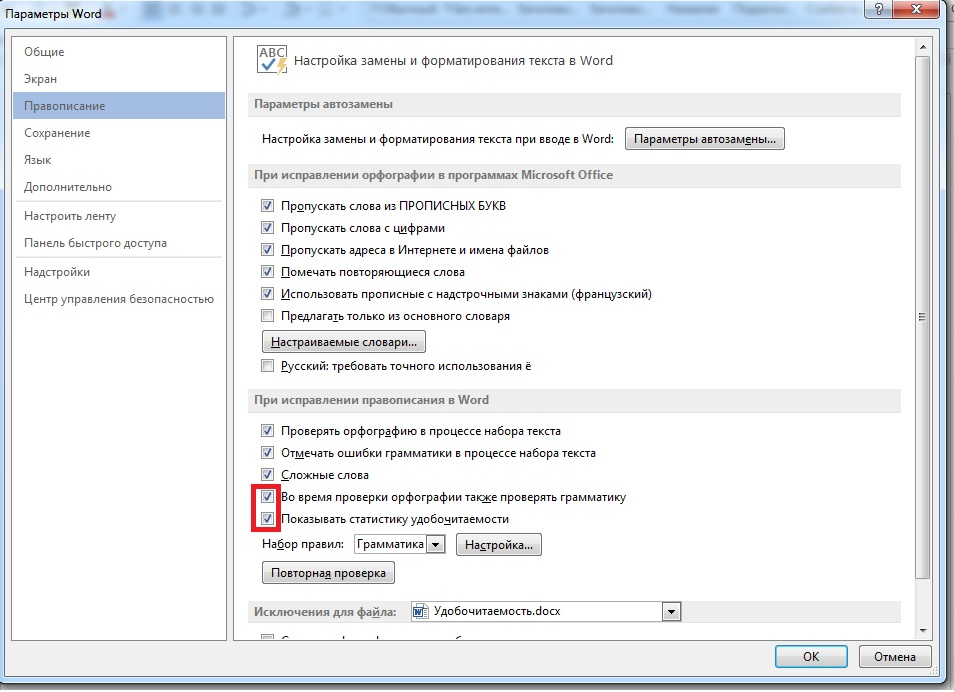


Рисунок 7 - Подключение статистики удобочитаемости

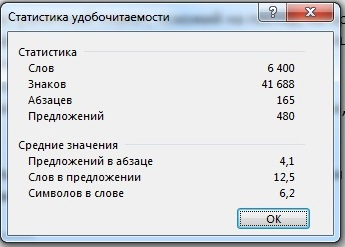


Рисунок 8 - Результат анализа текста на удобочитаемость

### **1.3.3 Программный продукт “Панда-копирайтинг”**

Панда-копирайтинг представляет собой веб-приложение для работы с текстом. Обладает широким функционалом, начиная от генератора заголовков для лендингов 4U до поиска тавтологий “Свежий взгляд”. особый интерес для нашей работы представляет раздел “Сравнение 2 текстов на схожесть” - онлайн сервис для поиска отличий.

Работа сервиса имеет несколько особенностей:

* для анализа можно добавить тексты вручную либо загрузить файлами;
* после загрузки тексты хранятся в буфере обмена 15 минут, а затем буфер очищается;
* тексты можно сравнивать как в общем массиве (изменения выделяются цветом), так и раздельно в колонках.

Интерфейс страницы для работы с текстом представлен на рисунке 9. Алгоритм работы с сервисом следующий:

1. Загрузка файла / набор текста вручную для анализа в левой колонке. Здесь предполагается загрузка оригинального текста (до изменений);
2. Загрузка файла / набор текста вручную для анализа в правой колонке. Здесь должен быть текст с поправками или изменениями;
3. Запуск процесса для сравнения текстов по нажатию на соответствующую кнопку.

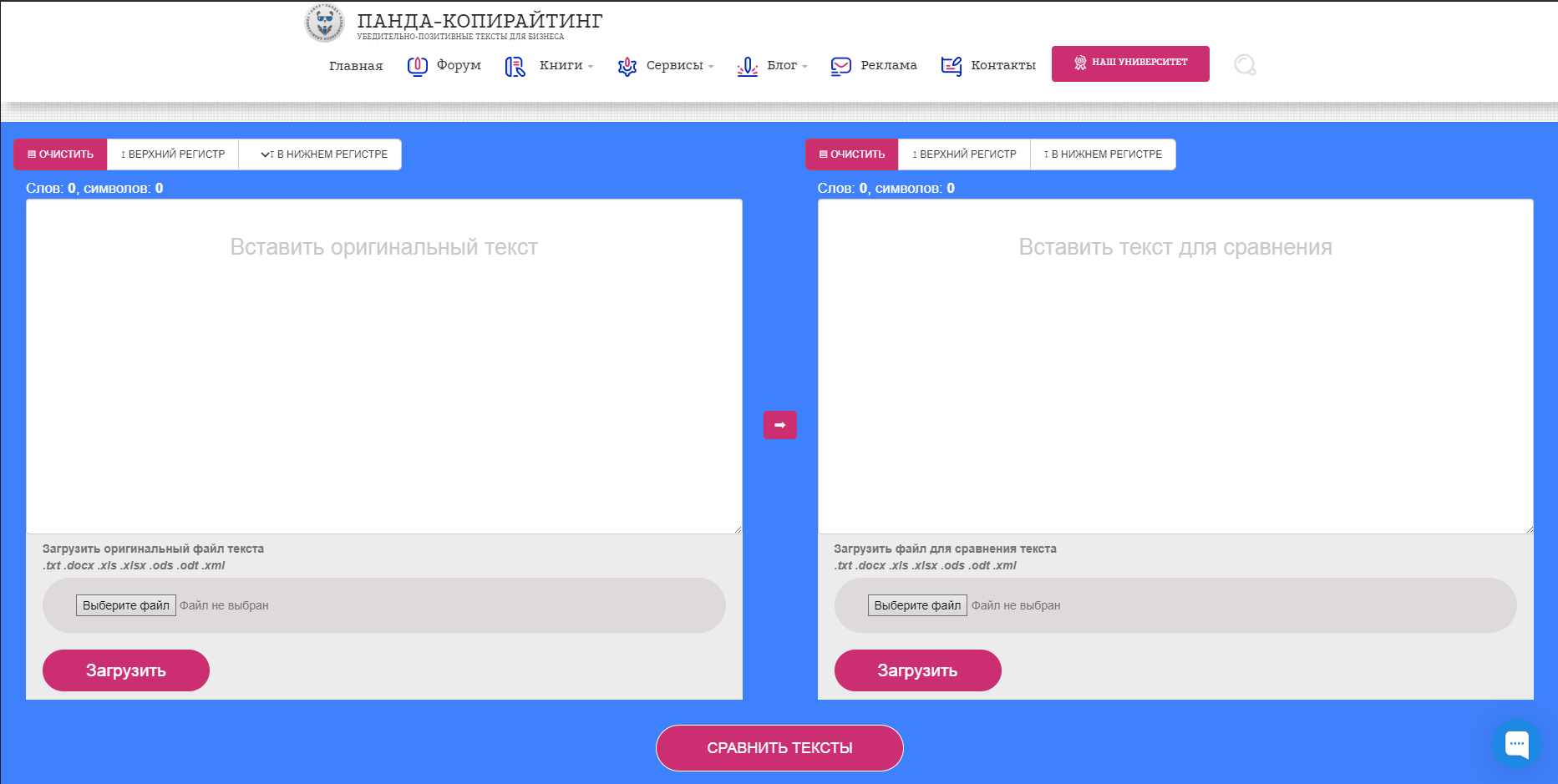


Рисунок 9 - Интерфейс страницы “Сравнение 2 текстов на схожесть”

Для тестирования сервиса взяли два различных текста. Первый текст описывал рельеф Алтайского края, второй - полезные ископаемые в Алтайском крае (Рисунок 10).

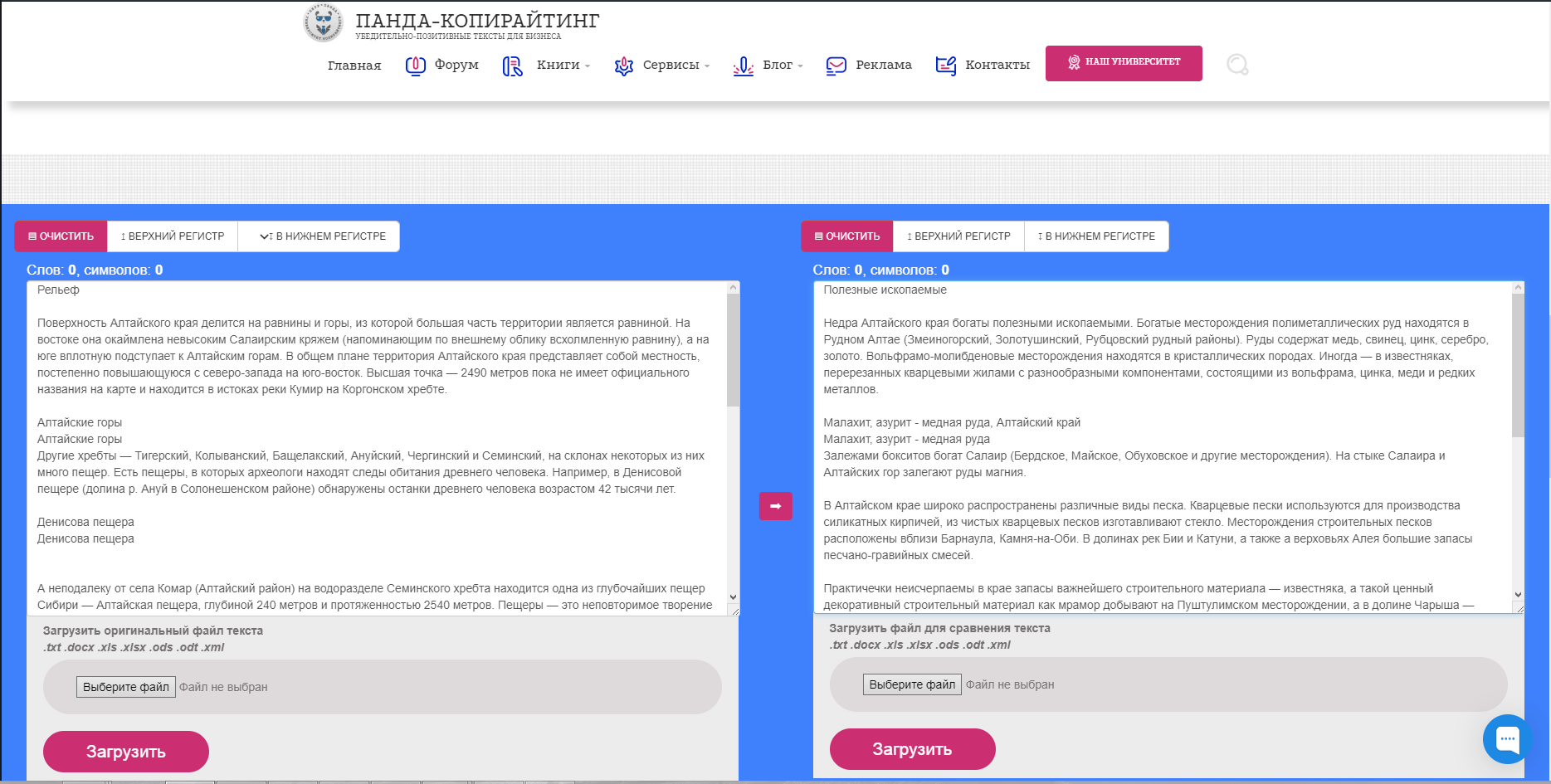


Рисунок 10 - Тестирование сервиса “Сравнение 2 текстов на схожесть”

На основе полученных результатов (Рисунок 11), можно сделать вывод, что тексты сравниваются посимвольно и никакого семантического анализа не производится. Т.е. если не вчитываться в текст, то невозможно понять, что и в одном, и в другом тексте речь идёт об Алтайском крае. Стоит отметить, что смысловой анализ не входит в задачи данного сервиса. Авторы работы предлагают лишь сравнение текстов.

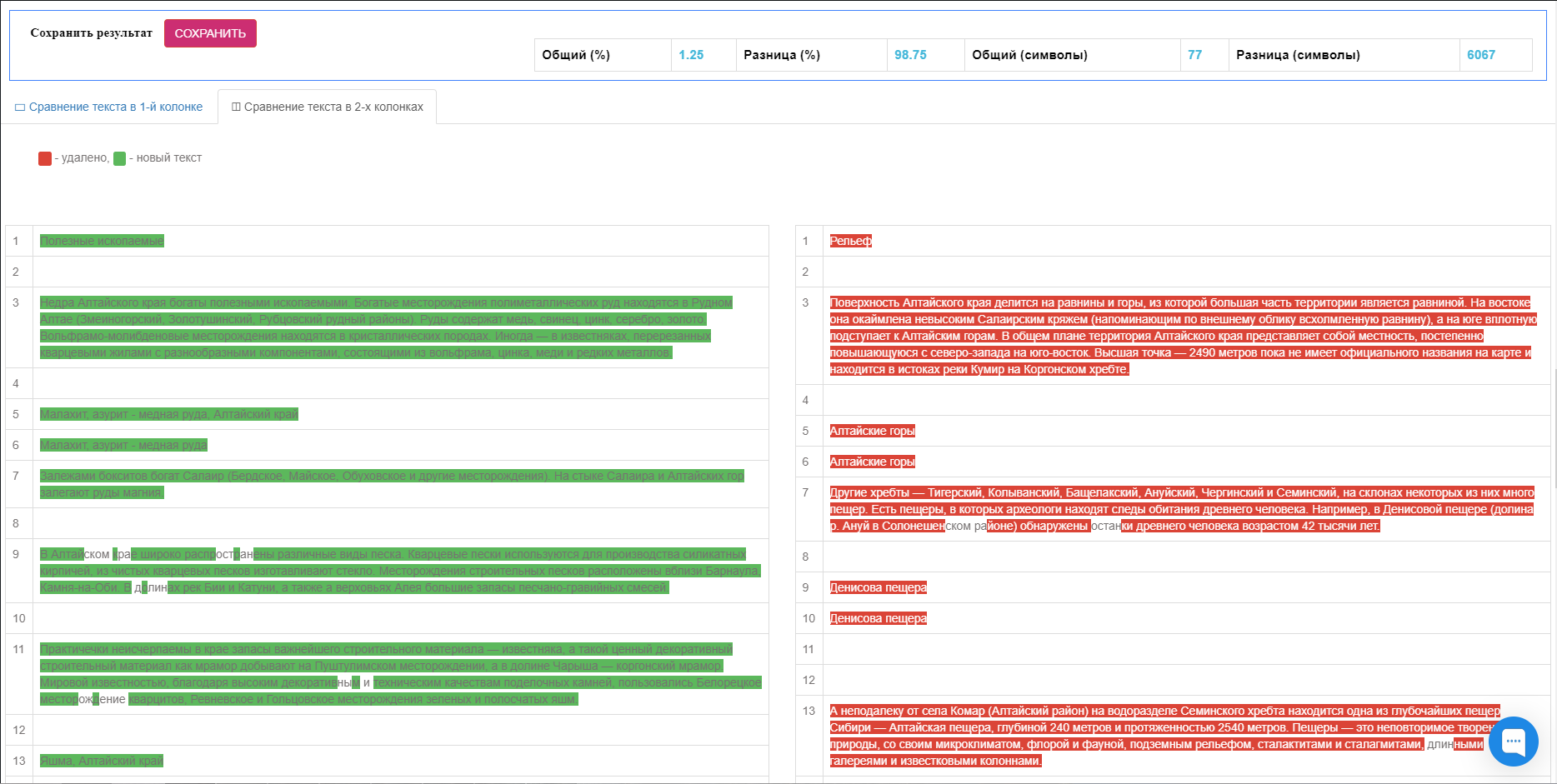


Рисунок 11 - Результат работы сервиса “Сравнение 2 текстов на схожесть”

### **1.3.4 Программный продукт “Miratext - биржа копирайтеров”**

Ещё одним инструментом для работы с текстом является web-приложение “Miratext - биржа копирайтеров”. Функционал сайта представлен также большим набором средств для анализа текста. Но этот сервис предоставляет возможность работы одновременно только с одним материалом.

Для тестирования воспользуемся только двумя функциями сервиса - полный семантический анализ текста и контроль ключевых слов.

Алгоритм работы для полного семантического анализа текста весьма примитивен, достаточно вставить текст вручную или добавить ссылку на текст и воспользоваться кнопкой анализировать.

Воспользуемся вторым вариантом - добавим ссылку на источник (Рисунок 12).

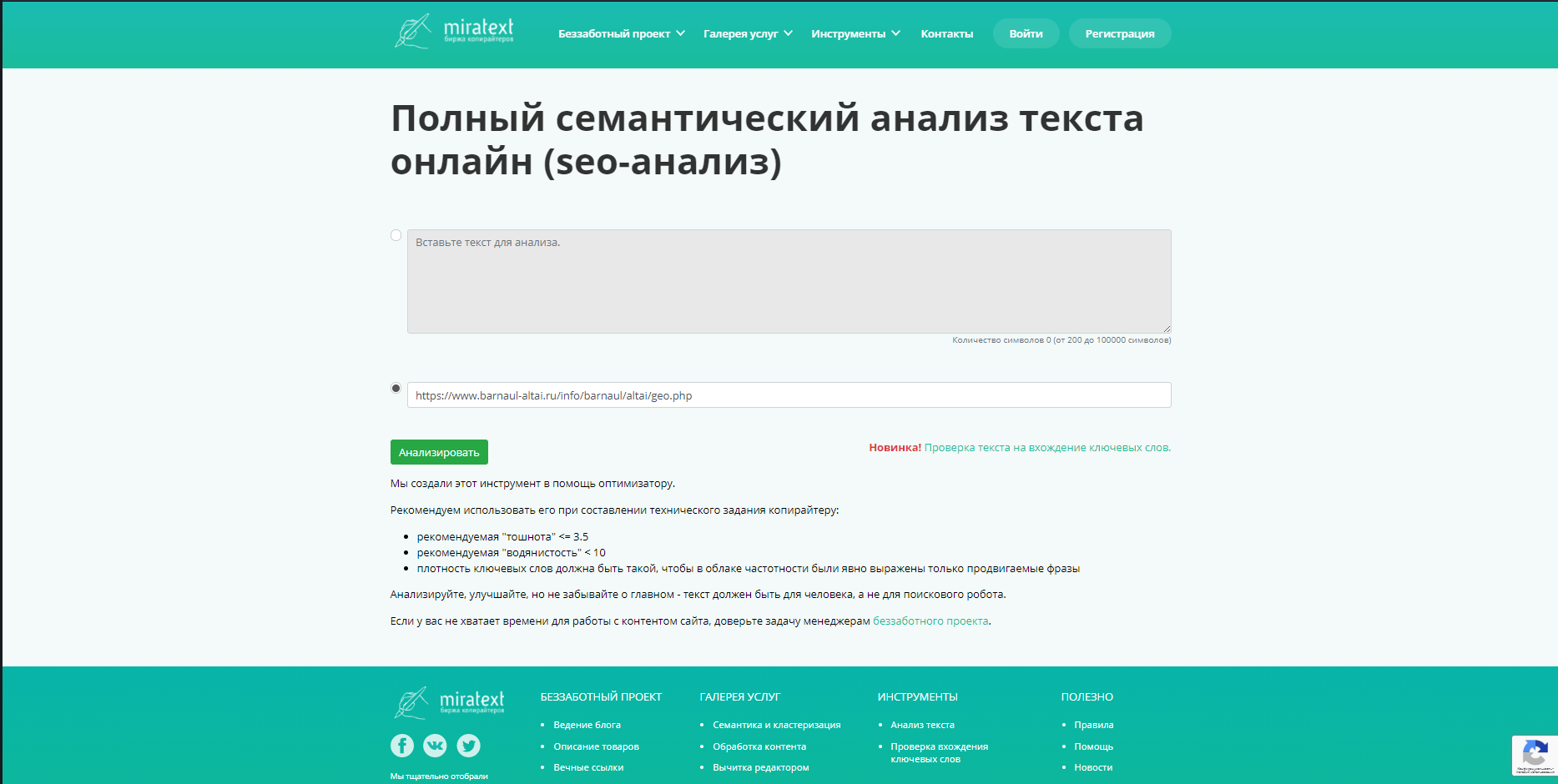


Рисунок 12 - Тестирование сервиса “Полный семантический анализ текста онлайн”

По результатам анализа (рисунок 13) пользователю предоставляется различные статические данные, такие как количество повторений (одно слово), количество повторений (словосочетание из двух слов), тошнота и др.

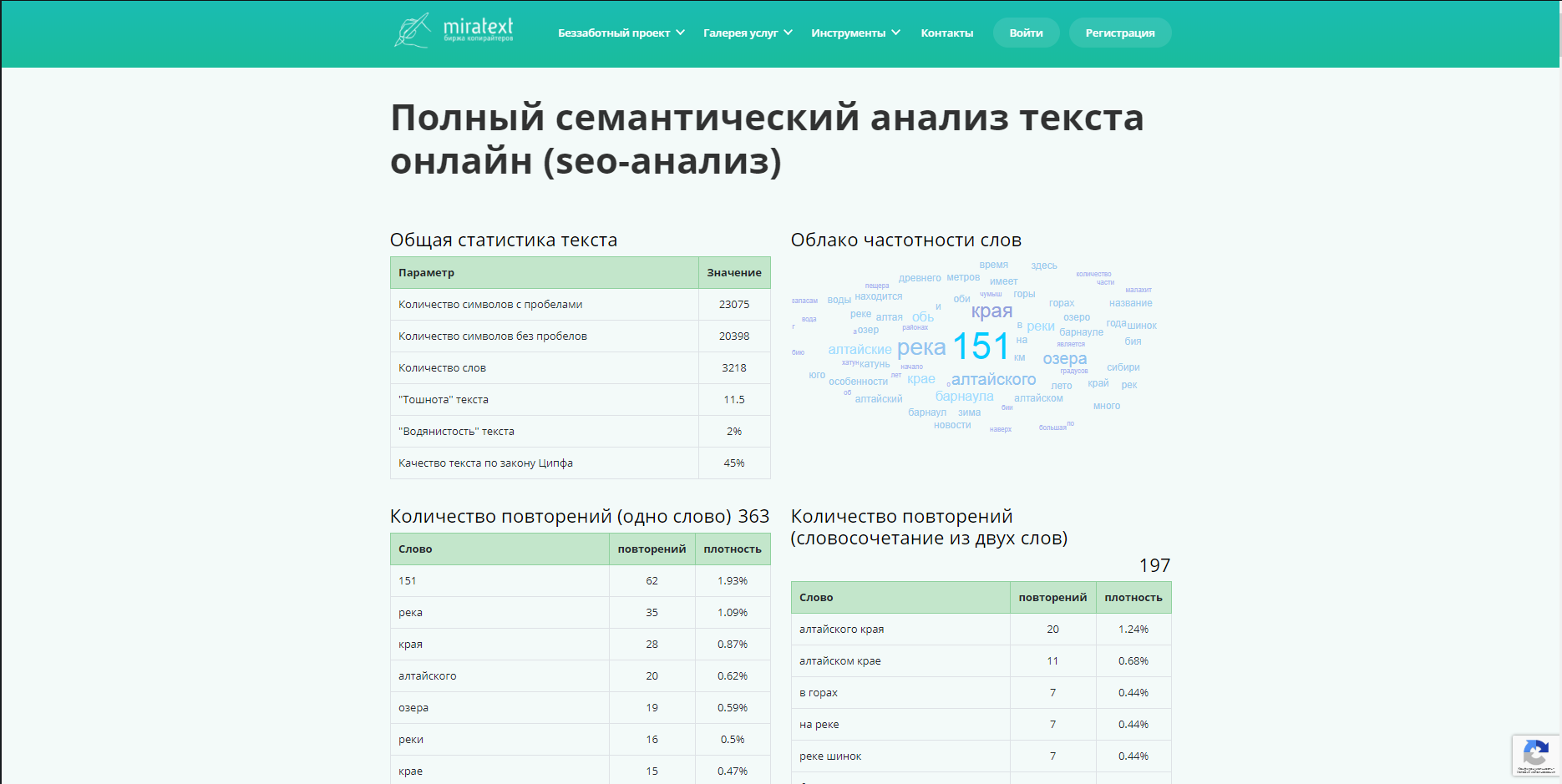


Рисунок 13 - Результат работы сервиса “Полный семантический анализ текста онлайн”

Кроме такого анализа можно выполнить проверку текста на вхождение ключевых слов. Для этого необходимо в текстовое поле добавить текст и далее ввести слова, которые нужно найти (совпадение может быть точным, разбавленным или типа LSI). Процесс поиска запускается с помощью соответствующей кнопки (Рисунок 14).

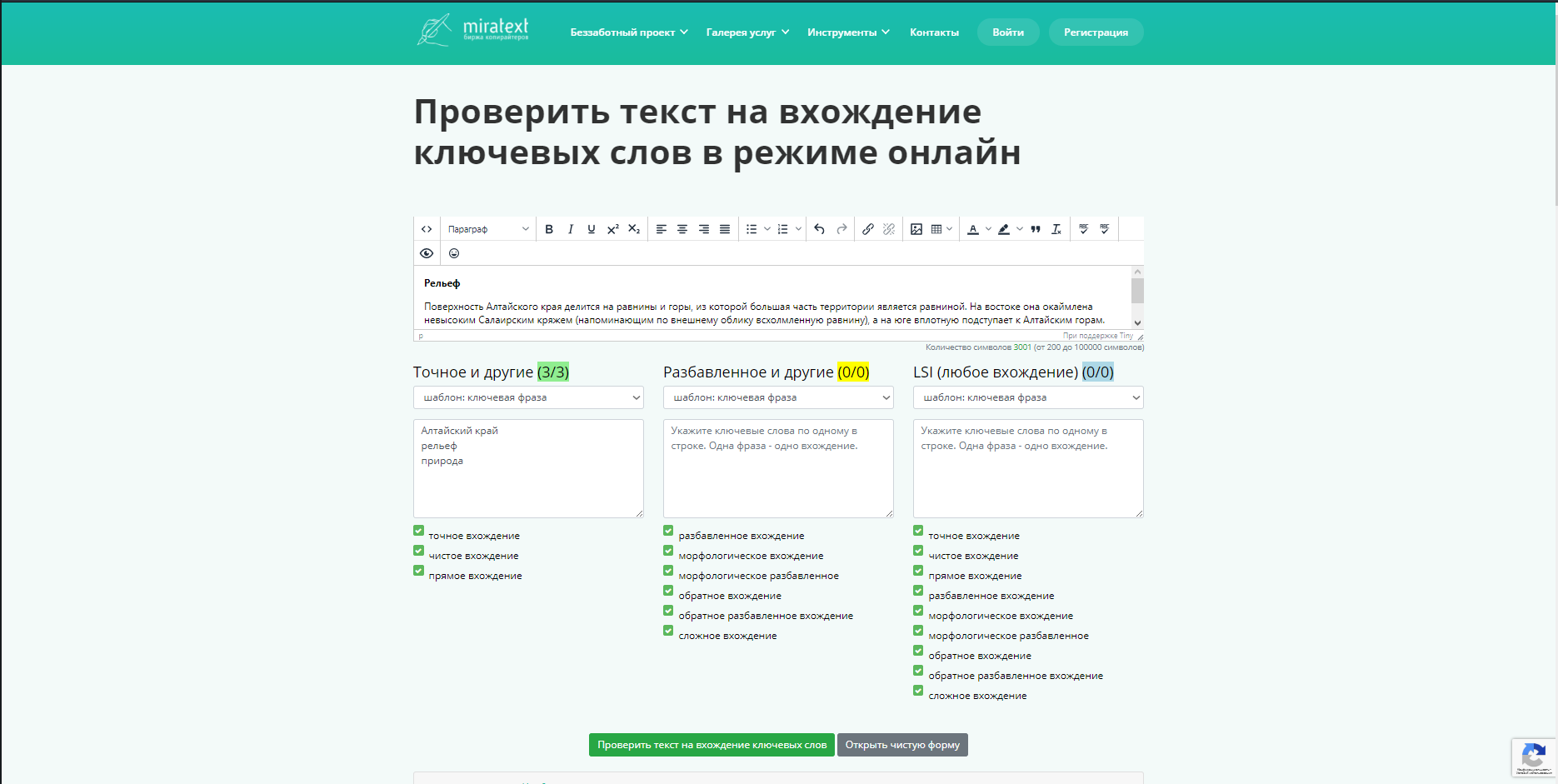


Рисунок 14 - Тестирование сервиса “Контроль ключевых слов”

В итоге получаем количественную оценку вхождений ключевых слов в текст (Рисунок 15).

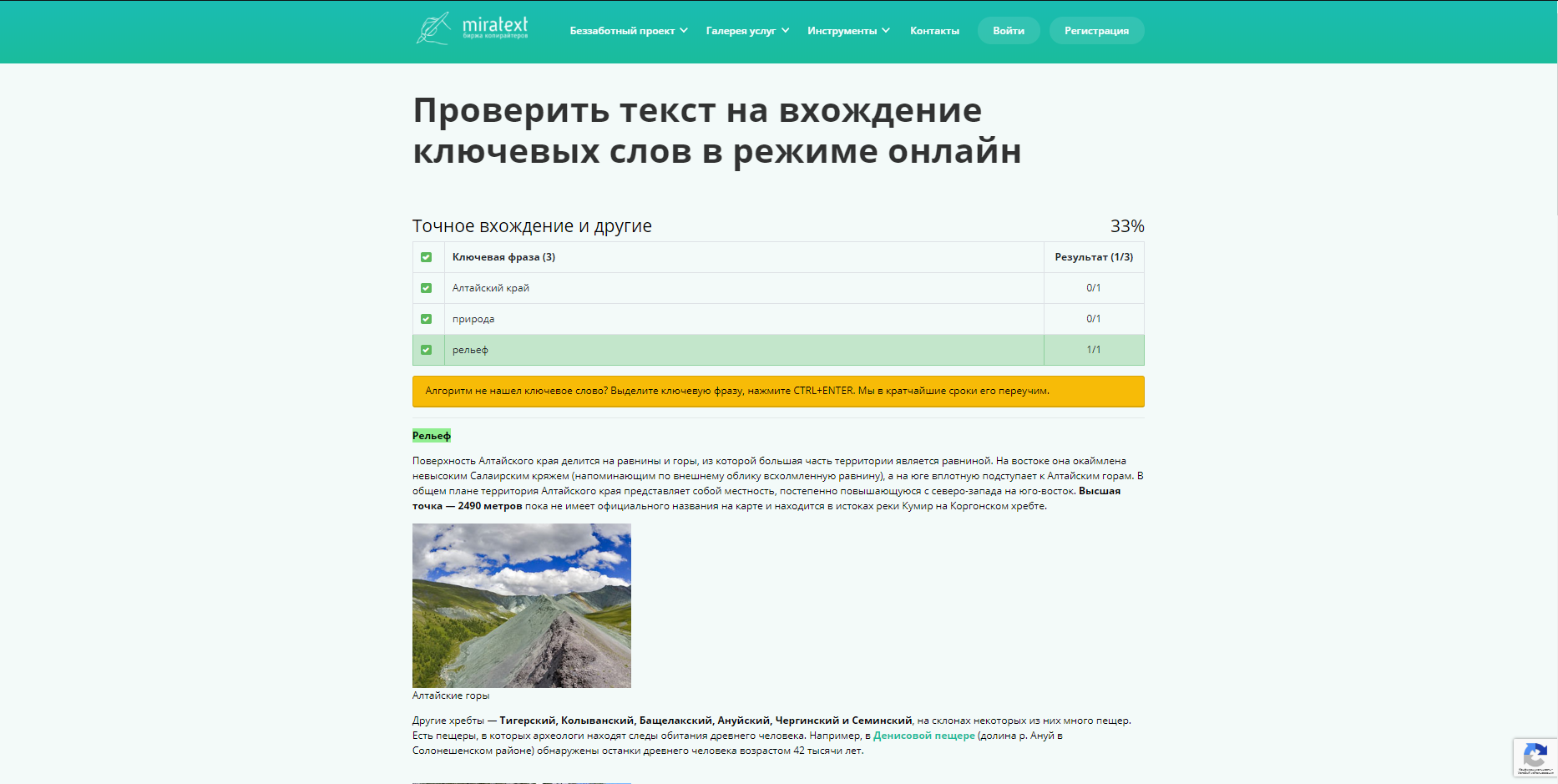


Рисунок 15 - Результат тестирования сервиса “Контроль ключевых слов”

Исходя из анализа различных ресурсов для работы с текстом, можем сделать вывод, что на данный момент в сети Интернет нет ПО, полностью отвечающего критериям поставленной задачи.

## **1.4 Функциональные возможности разрабатываемого приложения**

На основе анализа предметной области, а также изучения различных инструментов для работы с текстами, в том числе учебными материалами, определим необходимую функциональность будущего веб-приложения.

1. Загрузка учебных материалов в формате docx, doc, txt или путём ручного набора текста:
   1. Ввод названия учебного материала (для авторизованного пользователя);
   2. Ввод текста учебного материала в textarea;
2. Просмотр и удаление ключевых слов;
3. Просмотр результатов анализа учебных материалов на совместимость;
4. Добавление одного документа (текста) и выделение для него ключевых слов с возможностью удаления;
5. Регистрация пользователей:
   1. Ввод логина;
   2. Ввод пароля;
   3. Ввод пароля повторно;
   4. Ввод почтового адреса;
6. Авторизация пользователей:
   1. Ввод логина;
   2. Ввод пароля;
7. Восстановление пароля пользователя:
   1. Ввод почтового адреса;
   2. Ввод нового пароля;
   3. Повторный ввод нового пароля;
8. Просмотр загруженных файлов для зарегистрированного пользователя:
   1. Удаление файла;
   2. Удаление ключевых слов в просматриваемом документе;
   3. Скачивание файла;
9. Просмотр истории анализов для зарегистрированного пользователя;
10. Выход из аккаунта;
11. Изменение личных данных (для зарегистрированного пользователя):
    1. Изменение пароля.
12. Администрирование сайта.

# **2 Проектирование web-приложения**

## **2.1 Структура приложения**

Исходя из описанной выше функциональности определим структуру будущего сайта, которая обеспечит удобную и интуитивно понятную работу с сервисом.

Сайт должен содержать следующие страницы:

1. Главная страница;
2. Страница регистрации;
3. Страница авторизации;
4. Страница восстановления пароля;
5. Личный кабинет;
6. Страница добавления для анализа одного материала;
7. Страница просмотра/удаления ключевых слов одного материала и результата анализа;
8. Страница добавления для анализа двух материалов;
9. Страница просмотра/удаления ключевых слов для двух материалов и результата анализа;
10. Страница просмотра/удаления загруженных материалов;
11. Страница детального просмотра одного загруженного материала;
12. Страница просмотра истории проведенных анализов;
13. Страница администратора сайта.

Общая структура сайта представлена на рисунке 16.

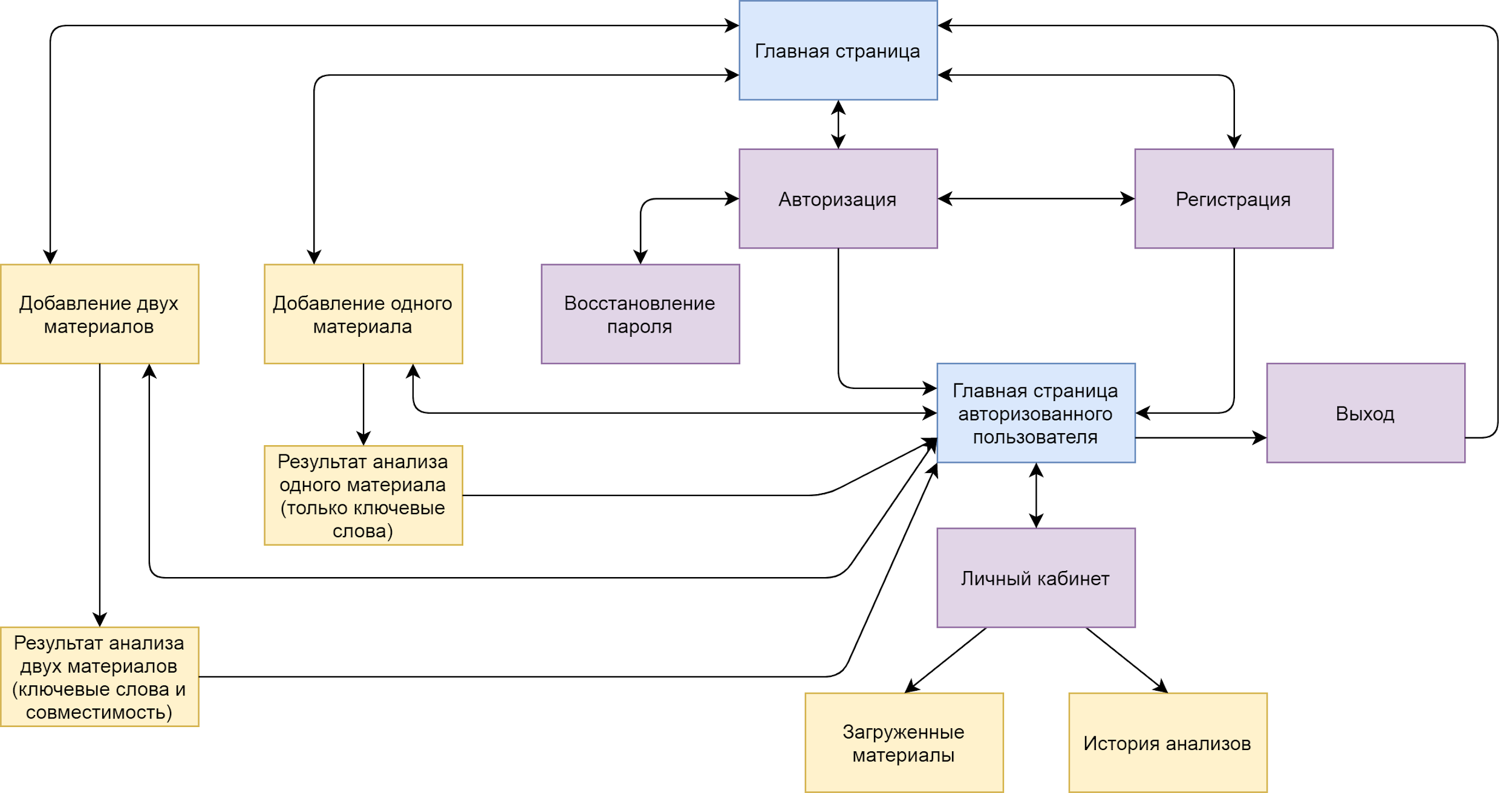


Рисунок 16 - Структура приложения

## **2.2 Описание процессов взаимодействия пользователей и системы**

### **2.2.1 Описание групп пользователей**

Основными пользователями системы будут люди, относящиеся к области образования, а именно педагоги ВУЗов, колледжей, учителя, преподаватели различных онлайн-школ и курсов. Также к списку пользователей системы могут примкнуть учёные, журналисты, писатели, копирайтеры, студенты, школьники и прочие категории людей, которым необходим комплексный подход в изучении, обучении или анализе материалов какой-либо области знаний.

Каждый пользователь может зарегистрироваться на сервисе. История авторизированного пользователя сохраняется, т.е. этому пользователю будут доступны все материалы, которые он когда-либо загрузил, а также результаты всех произведенных анализов.

Для незарегистрированного пользователя возможен анализ загруженных материалов только во время текущей сессии. При использовании сервиса повторно, придётся вновь загружать материалы.

Таким образом система должно иметь возможность работать с двумя типами пользователями: зарегистрированными и незарегистрированными.

Для незарегистрированного пользователя доступны такие функции как:

* добавление одного/двух материала(-ов) для анализа;
* просмотр/ удаление слов для одного/двух материалов;
* просмотр результатов анализа на совместимость.

Для зарегистрированного пользователя доступны возможности незарегистрированного пользователя, а также просмотр/удаление загруженным материалов и просмотр истории проведенных анализов.

### **2.2.2 Описание процесса регистрации и авторизации**

Для использования всех возможностей сайта пользователь должен пройти процесс регистрации. Наш сервис не предполагает хранение персональной информации , а также проведение коммерческих операций. Поэтому для регистрации пользователя было решено использовать минимум параметров, а именно логин, адрес электронной почты и пароль. Адрес электронной почты необходим для восстановления пароля в случае его утери пользователем. После регистрации пользователь попадает на главную страницу зарегистрированного пользователя.

Для входа в систему пользователь, прошедший регистрацию не в рамках текущей сессии, должен пройти процесс авторизации. Для этого необходимо ввести логин и пароль учетной записи.

### **2.2.3 Описание процесса работы с одним учебным материалом**

У пользователей системы может возникнуть потребность работы только с одним учебным материалом. Причем у зарегистрированного и незарегистрированного пользователи эти потребности могут различаться.

Неавторизованный пользователь может загрузить через соответствующую форму документ или ввести его вручную в поле textarea и получить список ключевых слов для данного документа.

Такая операция для зарегистрированного пользователя носит расширенный характер. Документ и выделенные ключевые слова сохраняются в библиотеке пользователя, а также пользователь может удалять определенные ключевые слова, если таковыми их не считает. Это же ему доступно и в случае просмотра документа из библиотеки. Но пользователю нужно будет заполнить обязательное поле “Наименование материала” для дальнейшего корректного отображения в библиотеке.

### **2.2.4 Описание процесса работы с двумя учебными материалами**

Процесс работы пользователей с двумя материалами аналогичен работе с одним материалом, но для двух документов можно рассчитать внутреннюю и внешнюю совместимость текста. Для расчёта внешней совместимости предполагается загрузка (ручной набор) материалов различных дисциплин. Для расчёта внутренней совместимости - загрузка (ручной набор) отдельных тем одного ресурса.

Так как для разных видов анализа должна быть получена соответствующая им оценка, пользователю необходимо установить значение чекбокса в зависимости от проводимого анализа. Например, при расчете внутренней совместимости вместе с результатом анализа пользователь получит сообщение о том, какой диапазон значений считается оптимальным для внутренней совместимости текстов и попадает ли полученный результат анализа в этот диапазон. Аналогично и для внешней совместимости.

### **2.2.4 Описание процесса работы с личным кабинетом**

В личном кабинете предполагается отображение библиотеки пользователя - комплекс загруженных пользователем материалов и истории проведенных анализов.   
 Работа с библиотекой перетекает в работу с отдельными файлами. Имеется возможность удаления файла. Работа с историей - просмотр списка, включающего как анализ одного документа на ключевые слова, так и анализ двух документов на совместимость с указанием типа совместимости.

Кроме этого с помощью личного кабинета пользователь может изменить данные учетной записи, например, поменять пароль.

# **3 Выбор инструментов разработки**

## **3.1 Клиентская часть**

Для разработки пользовательского интерфейса использовались следующие средства.

Bootstrap (также известен как Twitter 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.

## **3.2 Серверная часть**

Для разработки серверной части веб-приложения использовались такие инструменты.

Python – высокоуровневый язык программирования общего назначения. Синтаксис языка минималистичен, легко читаем и легок в изучении. Он является наиболее распространенным инструментом для работы с текстами. Также он имеет множество полезных библиотек.

Django — фреймворк для перфекционистов с дедлайнами. Свободный фреймворк для веб-приложений на языке Python, использующий шаблон проектирования MVC. Проект поддерживается организацией Django Software Foundation.

Jinja — это шаблонизатор для языка программирования Python. Он подобен шаблонизатору Django, но предоставляет Python-подобные выражения, обеспечивая исполнение шаблонов в песочнице. Это текстовый шаблонизатор, поэтому он может быть использован для создания любого вида разметки, а также исходного кода. Лицензирован под BSD лицензией. Шаблонизатор Jinja позволяет настраивать теги, фильтры, тесты и глобальные переменные. Также, в отличие от шаблонизатора Django, Jinja позволяет конструктору шаблонов вызывать функции с аргументами на объектах.

AJAX, Ajax (от англ. Asynchronous Javascript and XML — «асинхронный JavaScript и XML») — подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся быстрее и удобнее.

MySQL — [свободная](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%9F%D0%9E) [реляционная система управления базами данных](https://ru.wikipedia.org/wiki/%D0%A0%D0%B5%D0%BB%D1%8F%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%A1%D0%A3%D0%91%D0%94). Разработку и поддержку MySQL осуществляет корпорация [Oracle](https://ru.wikipedia.org/wiki/Oracle). Продукт распространяется как под [GNU General Public License](https://ru.wikipedia.org/wiki/GNU_General_Public_License), так и под собственной коммерческой лицензией.

## **3.3 Дополнительные инструменты**

Кроме средств для разработки клиентской и серверной части, для более удобной работы с проектом использовались следующие инструменты.

GitHub — крупнейший веб-сервис для хостинга IT-проектов и их совместной разработки. Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL).

Atom — бесплатный [текстовый редактор](https://ru.wikipedia.org/wiki/%D0%A2%D0%B5%D0%BA%D1%81%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%80%D0%B5%D0%B4%D0%B0%D0%BA%D1%82%D0%BE%D1%80) с [открытым исходным кодом](https://ru.wikipedia.org/wiki/%D0%9E%D1%82%D0%BA%D1%80%D1%8B%D1%82%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) для Linux, macOS, Windows с поддержкой [плагинов](https://ru.wikipedia.org/wiki/%D0%9F%D0%BB%D0%B0%D0%B3%D0%B8%D0%BD), написанных на [Node.js](https://ru.wikipedia.org/wiki/Node.js), и встраиваемых под управлением [Git](https://ru.wikipedia.org/wiki/Git). Большинство плагинов имеют статус свободного программного обеспечения, разрабатываются и поддерживаются сообществом.

Apache HTTP-сервер — [свободный](https://ru.wikipedia.org/wiki/%D0%A1%D0%B2%D0%BE%D0%B1%D0%BE%D0%B4%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5) [веб-сервер](https://ru.wikipedia.org/wiki/%D0%92%D0%B5%D0%B1-%D1%81%D0%B5%D1%80%D0%B2%D0%B5%D1%80). Apache является [кроссплатформенным ПО](https://ru.wikipedia.org/wiki/%D0%9A%D1%80%D0%BE%D1%81%D1%81%D0%BF%D0%BB%D0%B0%D1%82%D1%84%D0%BE%D1%80%D0%BC%D0%B5%D0%BD%D0%BD%D0%BE%D0%B5_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%BD%D0%BE%D0%B5_%D0%BE%D0%B1%D0%B5%D1%81%D0%BF%D0%B5%D1%87%D0%B5%D0%BD%D0%B8%D0%B5), поддерживает операционные системы [Linux](https://ru.wikipedia.org/wiki/Linux), [BSD](https://ru.wikipedia.org/wiki/BSD), [Mac OS](https://ru.wikipedia.org/wiki/Mac_OS), [Microsoft Windows](https://ru.wikipedia.org/wiki/Microsoft_Windows), [Novell NetWare](https://ru.wikipedia.org/wiki/Novell_NetWare), [BeOS](https://ru.wikipedia.org/wiki/BeOS).

Основными достоинствами Apache считаются надёжность и гибкость конфигурации. Он позволяет подключать внешние модули для предоставления данных, использовать [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) для [аутентификации](https://ru.wikipedia.org/wiki/%D0%90%D1%83%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%86%D0%B8%D1%8F) пользователей, модифицировать сообщения об ошибках и т. д. Поддерживает [IPv6](https://ru.wikipedia.org/wiki/IPv6).

MySQL Workbench - унифицированный визуальный инструмент для архитекторов, разработчиков и администраторов баз данных. MySQL Workbench предоставляет моделирование данных, разработку SQL и комплексные инструменты администрирования для настройки сервера, администрирования пользователей, резервного копирования и многого другого. MySQL Workbench доступен в Windows, Linux и Mac OS X.

## **3.4 Библиотека для расчёта совместимости учебных материалов**

Основной задачей работы является расчёт внутренней и внешней совместимости учебных материалов. Для реализации основной функциональности приложения было решено использовать модуль для работы с естественными языками, разработанный Нефёдовой Анастасией в рамках выпускной квалификационной работы.

Модуль имеет следующие функциональные возможности:

* Выделение слов из текста;
* Вычисление TF-IDF для каждого слова;
* Составление тезауруса, используя закон Ципфа и закон Хикса;
* Проведение анализа совместимости учебных материалов.

Данное программное обеспечение позволяет в полной мере реализовать требования к разрабатываемой нами системе. Также стоит отметить, что модуль написан на языке программирования Python, что позволяет легко интегрировать его в нашу задачу.

# **4 Архитектура приложения**

## **4.1 База данных**

В ходе анализа предметной области и требований к функционалу были выделены следующие сущности базы данных:

* пользователь;
* документ;
* тема;
* результат анализа;
* ключевое слово.

Разработана реляционная база данных. Концептуальная модель базы представлена на рисунке 17.

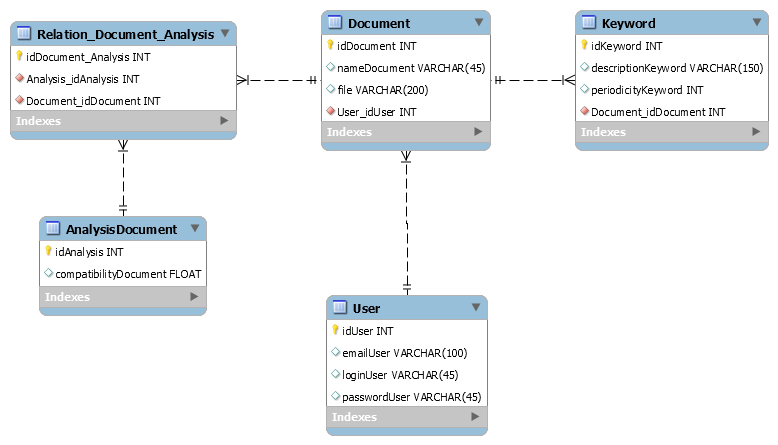


Рисунок 17 - Концептуальная модель базы данных

Ниже представлено подробное описание сущностей.

Сущность User (пользователь) содержит информацию о зарегистрированном пользователе системы.

Таблица 1 - Описание сущности User (пользователь)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Назначение |
| idUser | int | первичный ключ |
| emailUser | varchar(100) | адрес электронной почты |
| loginUser | varchar(45) | логин |
| passwordUser | varchar(45) | пароль |

Сущность Document (документ) содержит информацию о загруженном материале для анализа на совместимость.

Таблица 2 - Описание сущности Document (документ)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Назначение |
| idDocument | int | первичный ключ |
| nameDocument | varchar(45) | наименование |
| file | varchar(200) | путь к файлу на сервере |
| User\_idUser | int | внешний ключ таблицы “User” |

Сущность Keyword (ключевое слово) содержит информацию о ключевых словах материала.

Таблица 3 - Описание сущности Keyword (ключевое слово)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Назначение |
| idKeyword | int | первичный ключ |
| descriptionKeyword | varchar(150) | описание ключевого слова |
| periodicityKeyword | int | частота слова в материале |
| Document\_idDocument | int | внешний ключ таблицы “Document” |

Сущность AnalysisDocument (анализ материалов) содержит результат анализа двух материалов.

Таблица 4 - Описание сущности AnalysisDocument (анализ материалов)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Назначение |
| idAnalysisDocument | int | первичный ключ |
| compatibilityDocument | float | совместимость в процентах |

Сущность Relation\_Document\_Analysis (отношение документ - анализ) - расшивочная таблица для отношения документ-анализ.

Таблица 5 - Описание сущности Relation\_Document\_Analysis (отношение документ - анализ)

|  |  |  |
| --- | --- | --- |
| Поле | Тип данных | Назначение |
| idDocument\_Analysis | int | первичный ключ |
| Document\_idDocument | int | внешний ключ таблицы “Document” |
| Analysis\_idAnalysis | int | внешний ключ таблицы “AnalysisDocument ” |

## **4.2 Описание архитектуры веб-приложения**

Приложение имеет трехуровневую архитектуру (рисунок 18). В функцию клиентской части входит только интерактивное взаимодействие с пользователем, бизнес-логика вынесена на сервер приложения, который, собственно, и обеспечивает формирование запросов к базе данных, передаваемых на выполнение серверу базы данных.

К достоинствам такой архитектуры стоит отнести: масштабируемость, высокую безопасность и надежность. К недостаткам - сложность администрирования и обслуживания, высокие требования к производительности серверов приложений и серверов базы данных, высокие требования к скорости канала между сервером базы данных и сервером приложений.

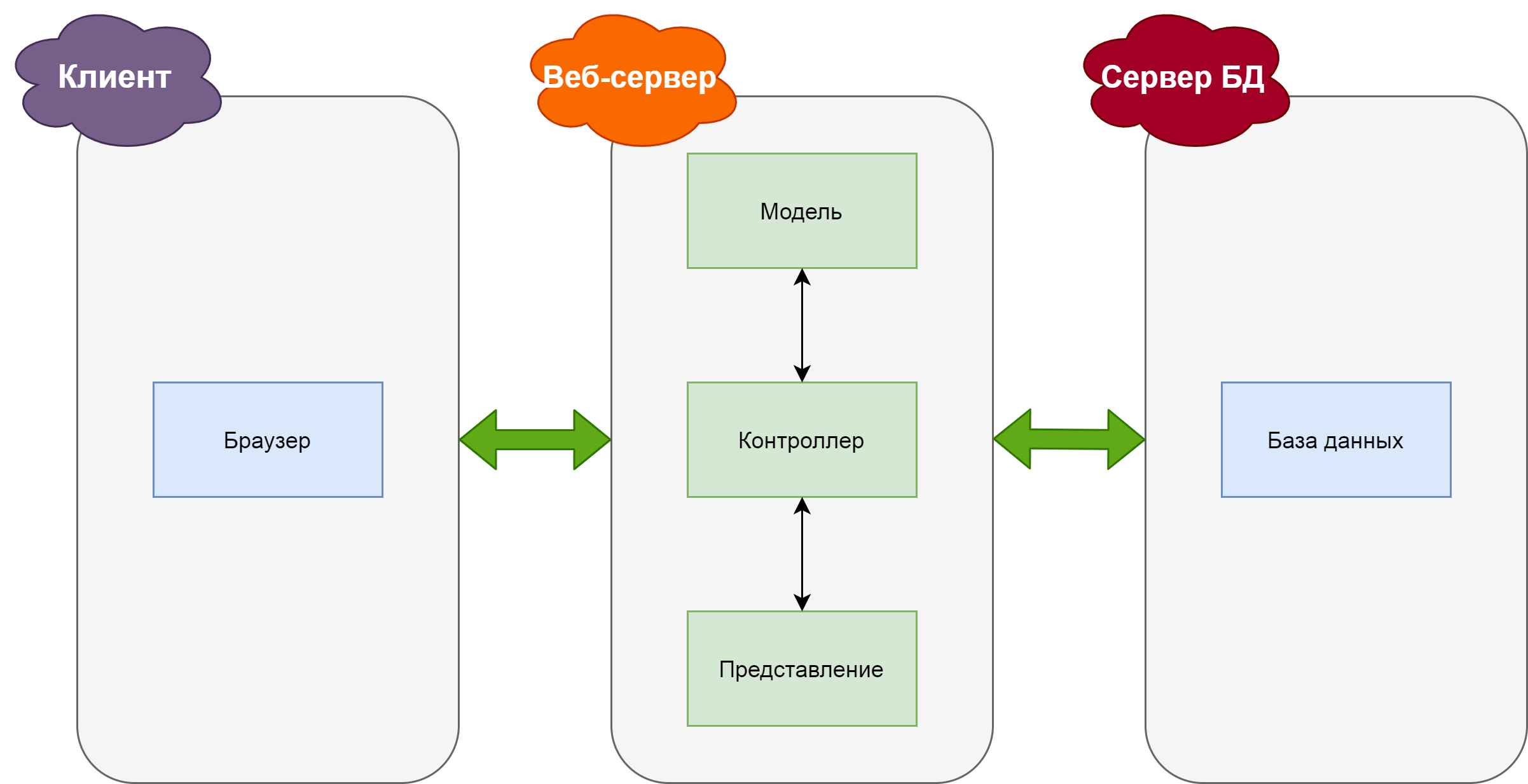


Рисунок 18 - Архитектура приложения

Уровень приложения реализующий бизнес-логику системы спроектирован на основе паттерна проектирования MVC (Model-View-Controller) (рисунок 19). Model-View-Controller («Модель-Представление-Контроллер») — схема 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 своё состояние.

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

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

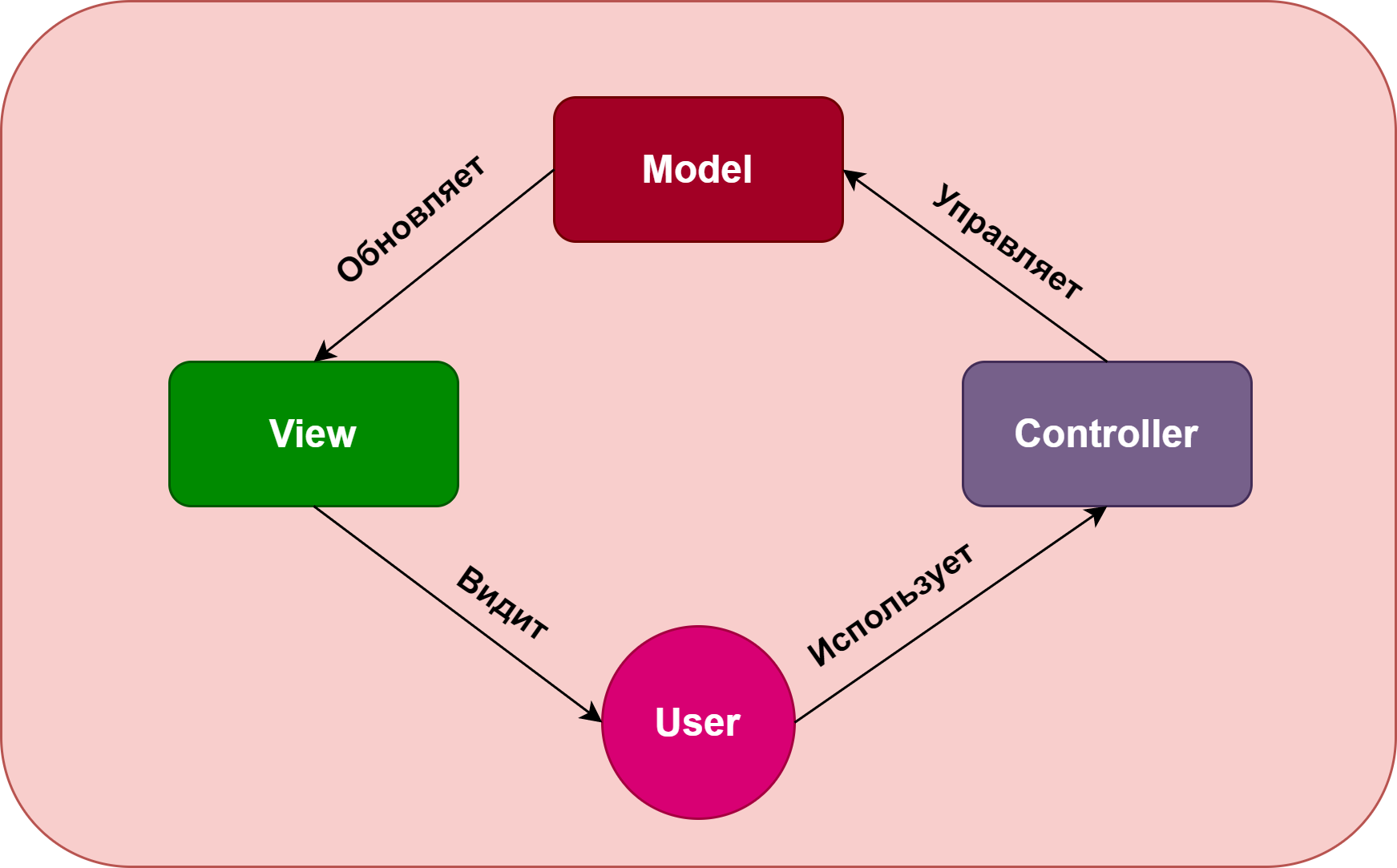


Рисунок 19 - Шаблон проектирования MVC

# **5 Проектирование справочной системы**

## **5.1 Руководство пользователя**

Пользователь открывает приложение, появляется главная страница.

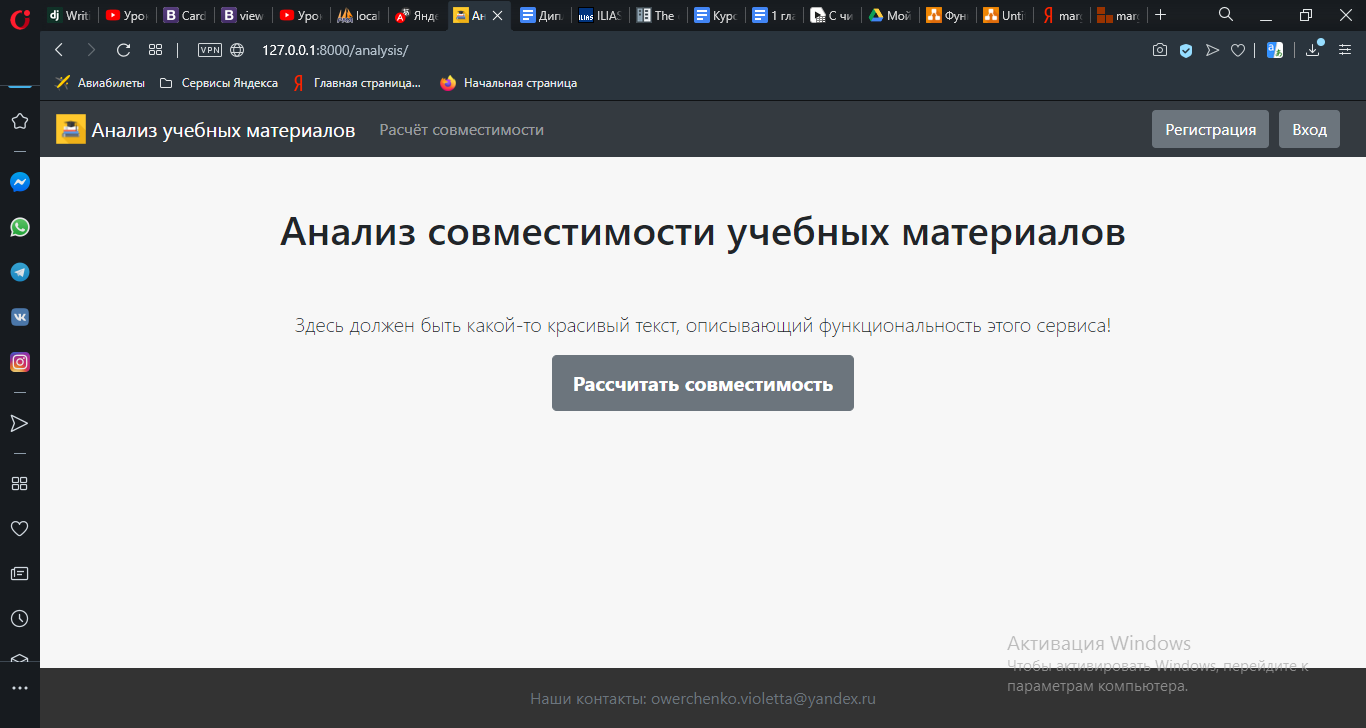


Рисунок 20 - Главная страница

По нажатию на “Расчёт совместимости”.

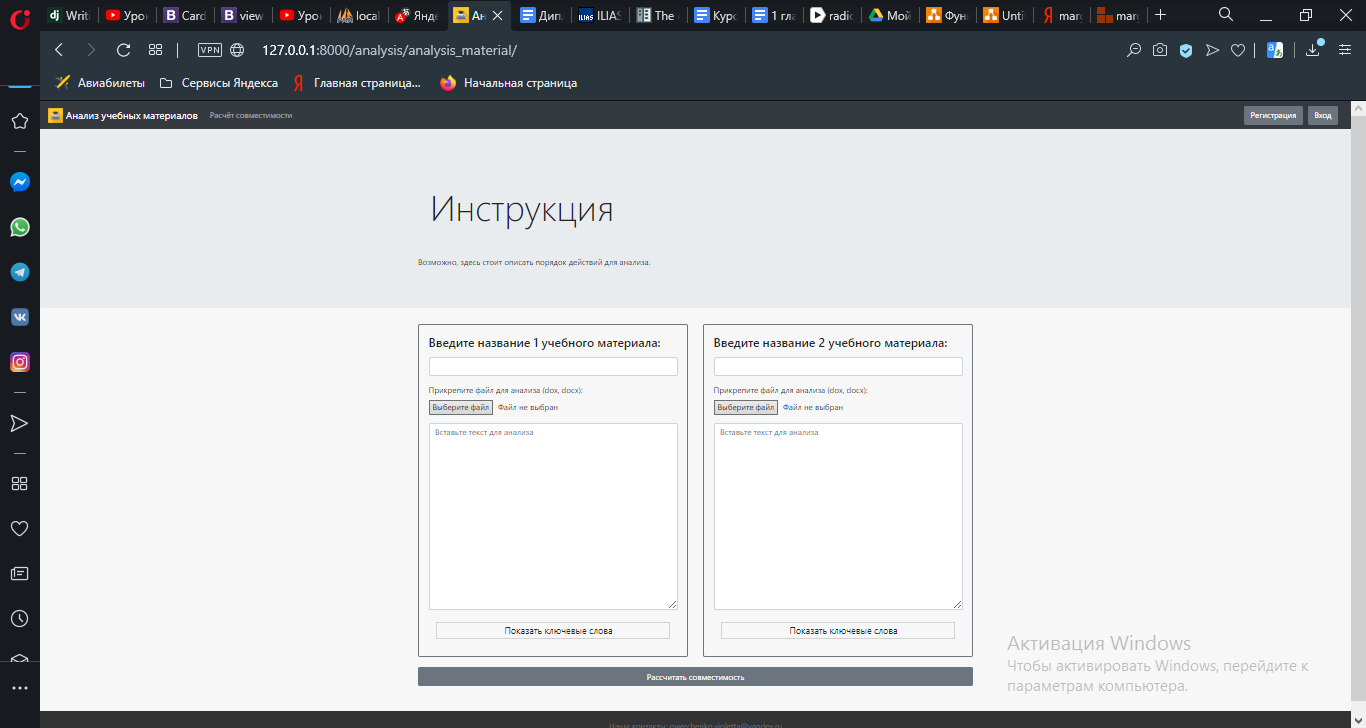


Рисунок 21 - Страница для расчёта совместимости

По нажатию на кнопку “Показать ключевые слова”.

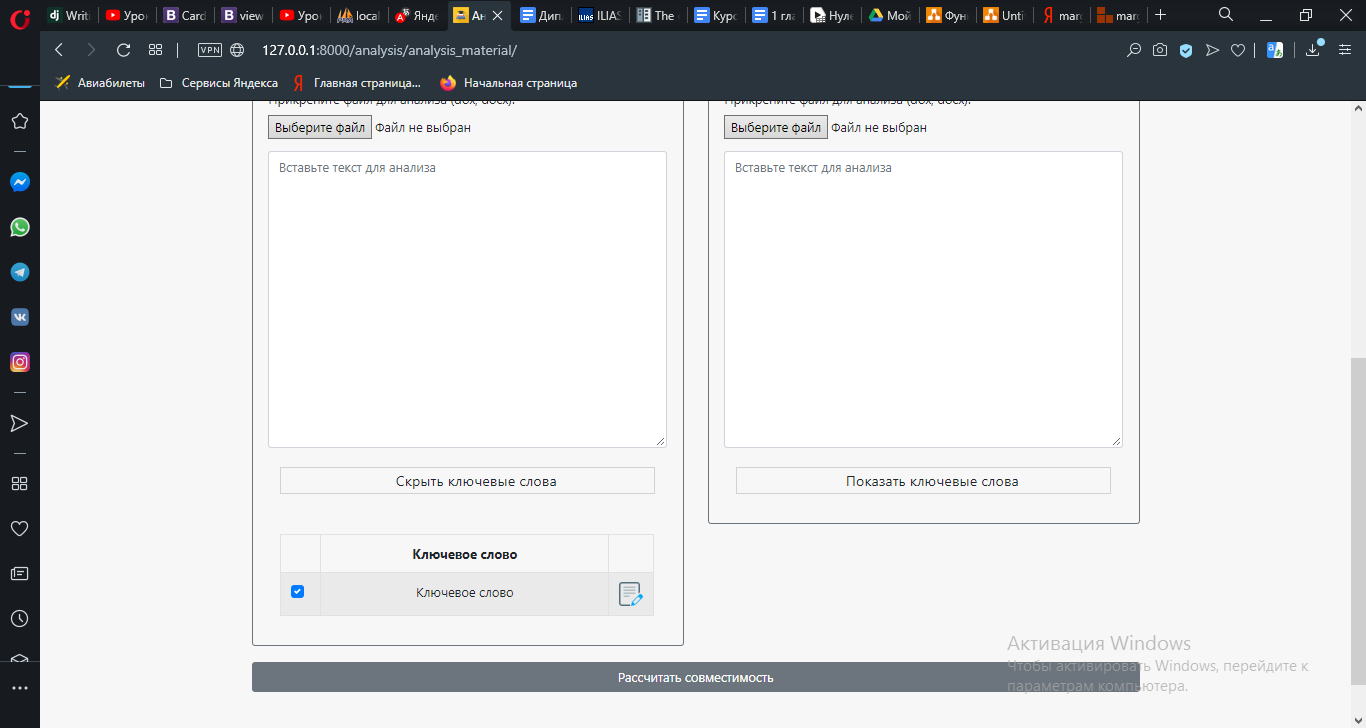


Рисунок 22 - Страница для расчёта совместимости. Просмотр ключевых слов

По нажатию на кнопку “Вход”.

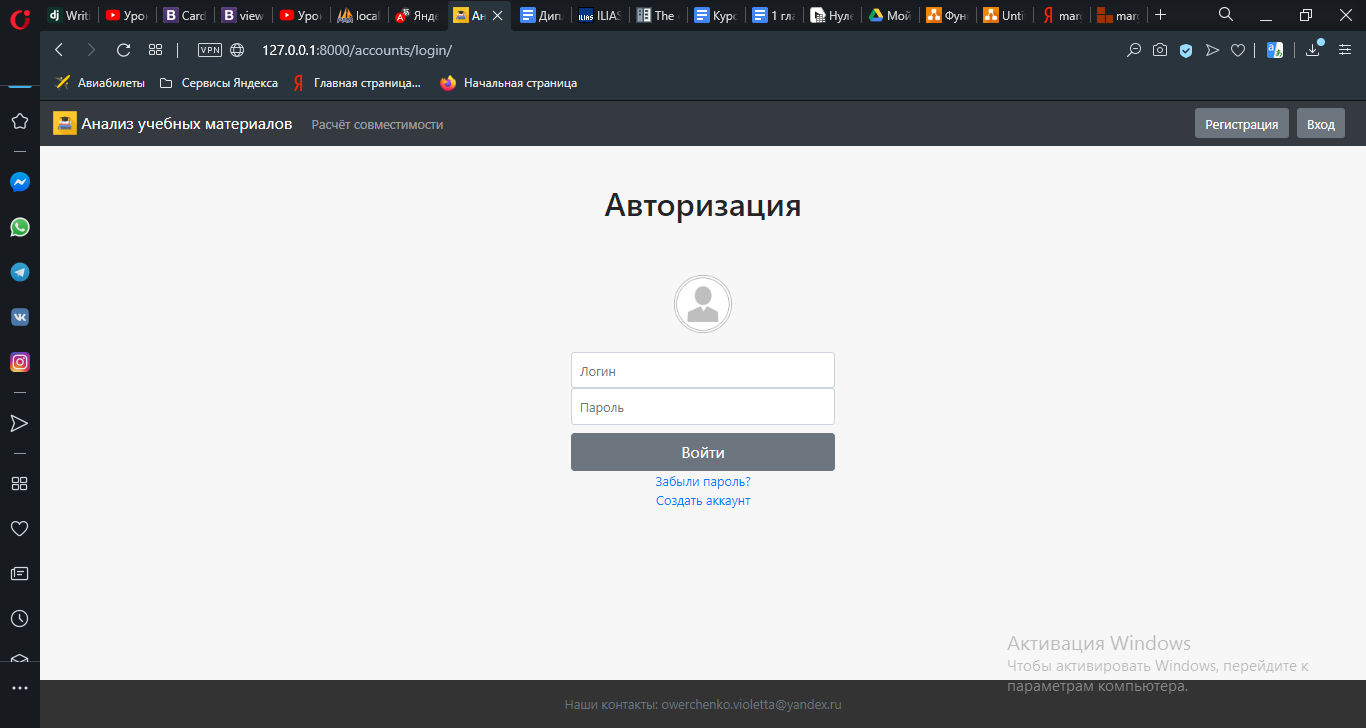


Рисунок 23 - Страница авторизации пользователя

По нажатию на кнопку “Забыли пароль”.

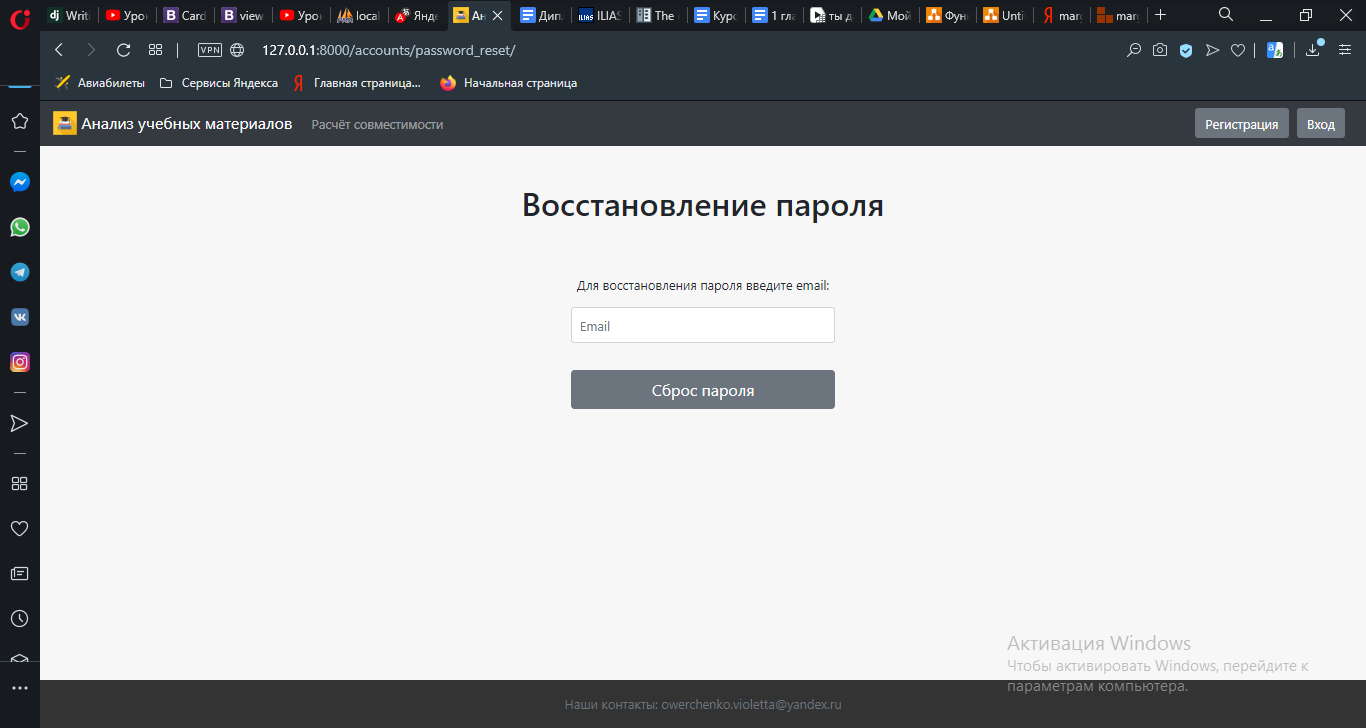


Рисунок 24 - Страница восстановления пароля

По нажатию на кнопку “Сброс пароля”.

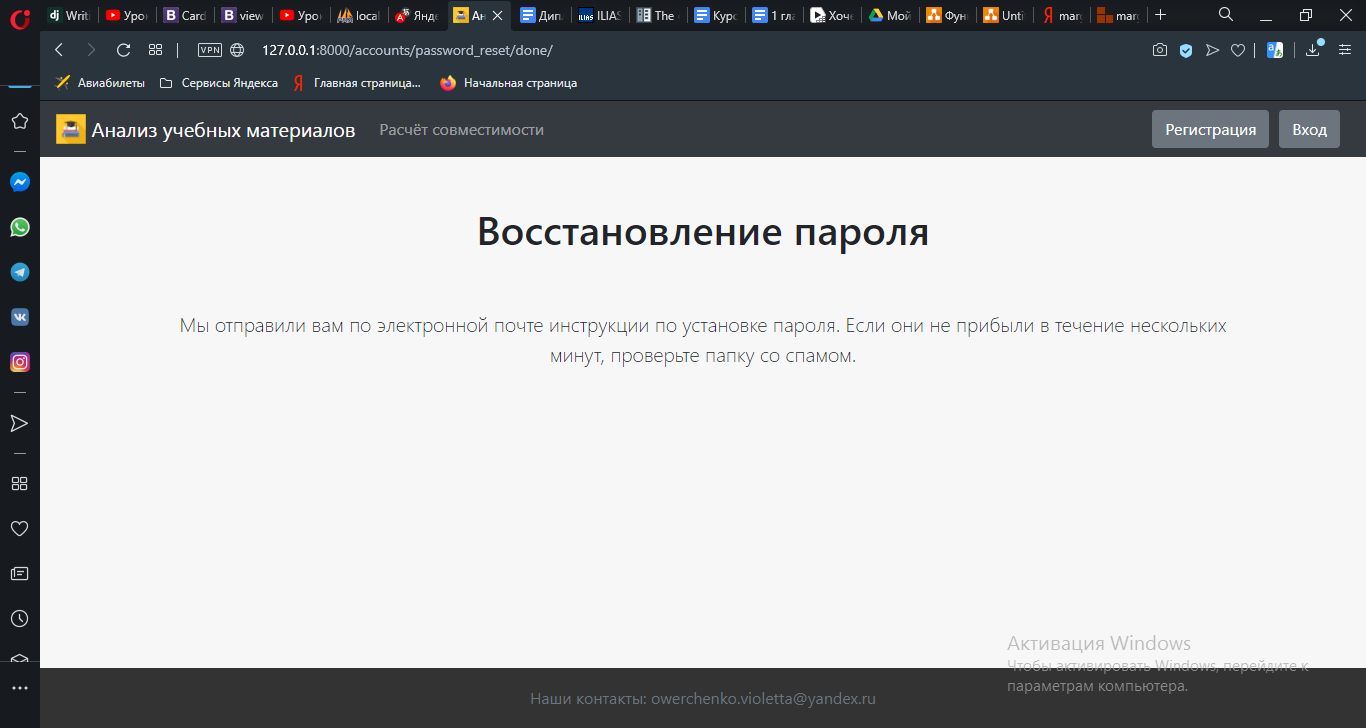


Рисунок 25 - Информационная страница для восстановления пароля

При нажатии на кнопку “Регистрация” или ссылку “Создать аккаунт”.

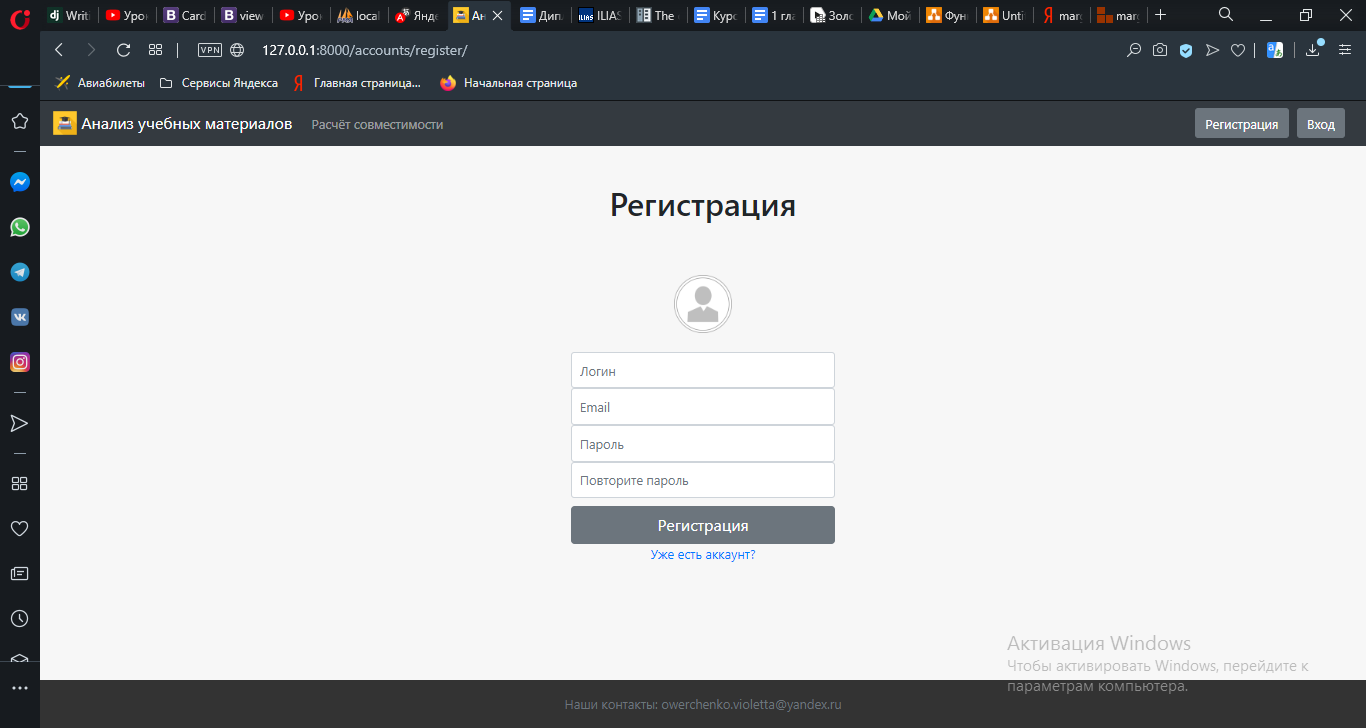


Рисунок 26 - Страница регистрации

При нажатии на ссылку “Уже есть аккаунт?” будет выполнен переход на страницу авторизации.

После авторизации пользователь попадает на главную страницу для зарегистрированного пользователя.

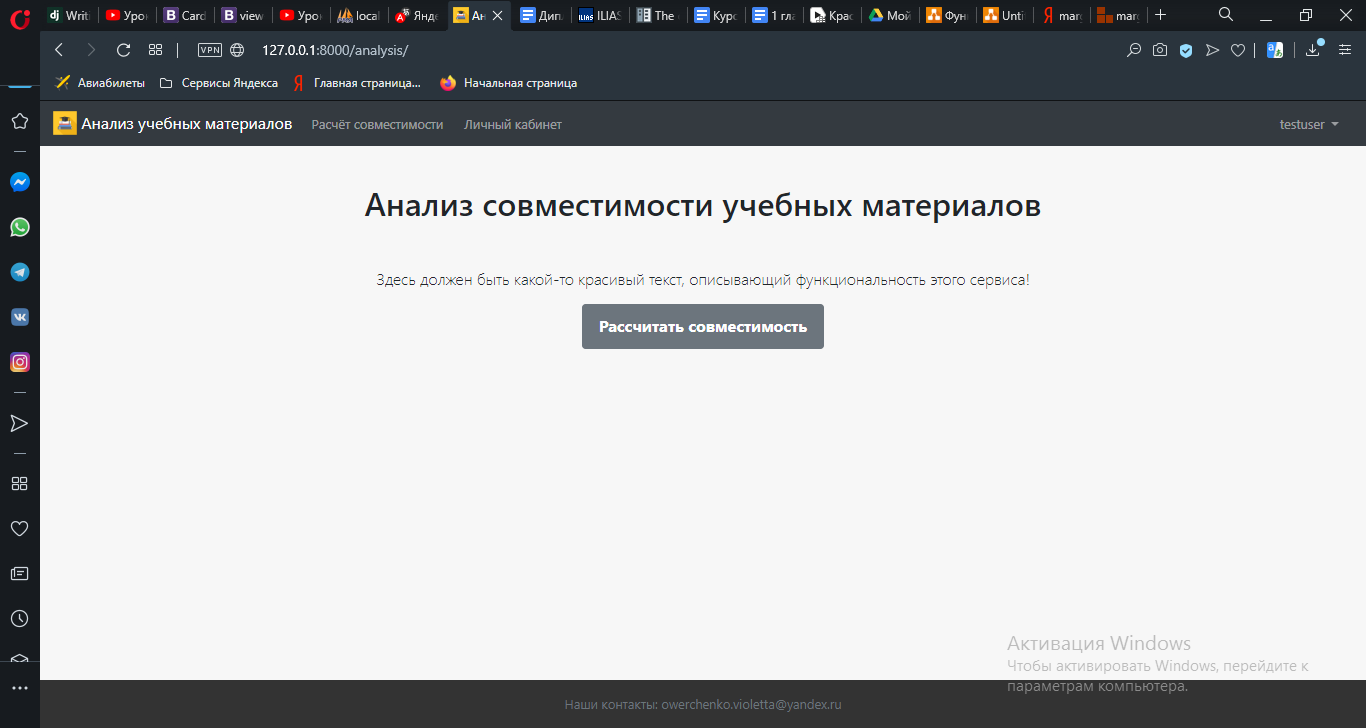


Рисунок 27 - Главная страница авторизованного пользователя

По нажатию на “Расчёт совместимости” пользователь попадает на страницу для расчёта совместимости (рисунок 5).

По нажатию на “Личный кабинет” пользователь переходит на страницу с его загруженными материалами и проведенными анализами.

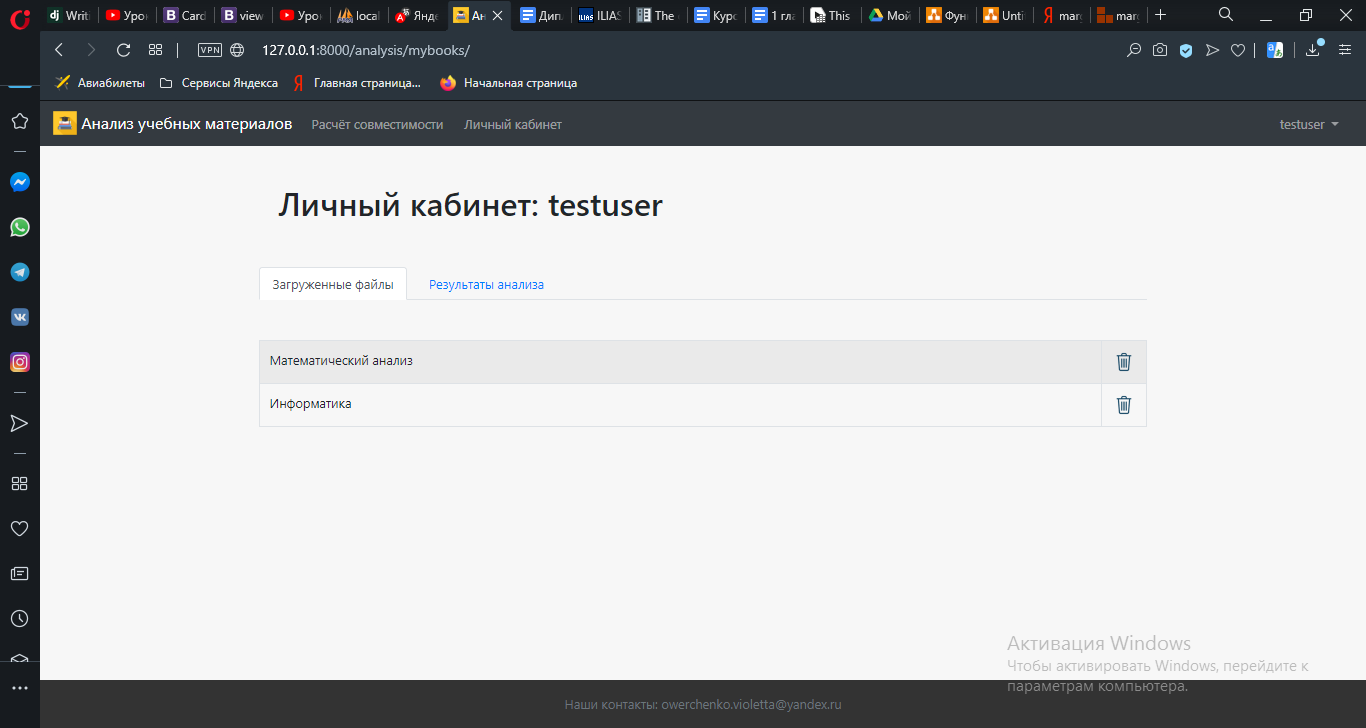


Рисунок 28 - Страница “Личный кабинет”. Загруженные материалы

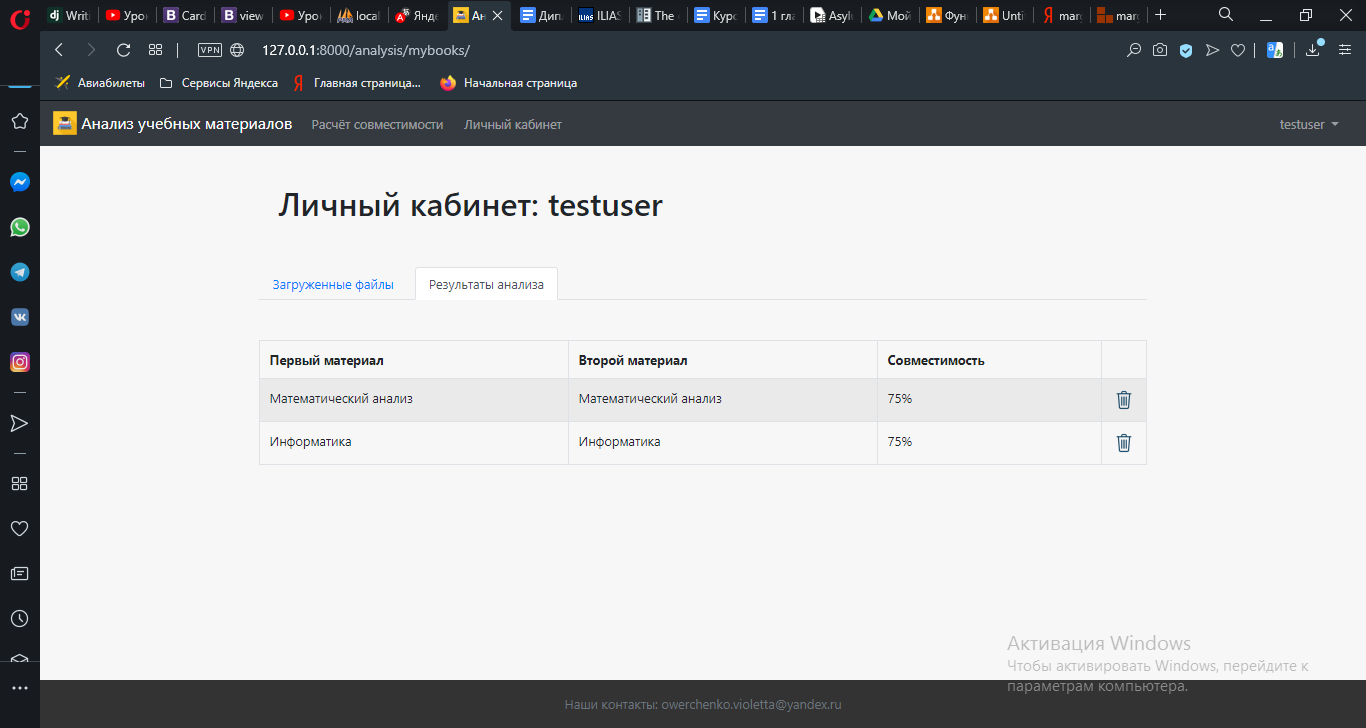


Рисунок 29 - Страница “Личный кабинет”. Результаты анализа

При нажатии на “логин” -> “выход” будет выполнен выход из аккаунта на главную страницу.

## **5.2 Требования к системному ПО и техническим средствам**

Для функционирования разработанного программного продукта необходим компьютер, удовлетворяющий системным требованиям:

* ОС не ниже Windows 7 или Ubuntu 18.04;
* Процессор: Intel Core i3 5005U и новее;
* RAM: 4 Gb DDR3;
* HDD: 50 Gb;
* Монитор, клавиатура, мышь или трекпад;
* Браузер: не ниже Mozilla Firefox 55 и новее, Google Chrome 75 и новее или Opera 60 и новее;
* Сервер с установленной базой данных MySQL;
* Сервер с установленным ПО (Django);
* Доступ к интернету (для корректной работы библиотеки Bootstrap).

# **Список использованных источников**

1. Влад Головач. Дизайн пользовательского интерфейса . [Электронный ресурс]. – Режим доступа: http://uibook2.usethics.ru/, свободный.
2. Документация по Django [Электронный ресурс] // Документация по Django [сайт]. Режим доступа: https://docs.djangoproject.com/en/2.2/.
3. Документация по Python [Электронный ресурс] // Руководство по Python [сайт]. Режим доступа: https://docs.python.org.
4. Документация по Bootstrap [Электронный ресурс] // Руководство по Bootstrap [сайт]. Режим доступа: <https://getbootstrap.com/docs/4.5/getting-started/introduction/>/
5. Документация JavaScript [Электронный ресурс] // Руководство по Bootstrap [сайт]. Режим доступа: https://www.javascript.com.
6. Рыбакова Г.Р., Кротова И.В., Камоза Т.Л., Моделирование совместимости учебной информации: методологические подходы [Текст] / Рыбакова Г.Р., Кротова И.В., Камоза Т.Л. , 2015. — 11 с.
7. Клочков В.П., Зайцев А.В., Научные предпосылки возникновения междисциплинарного дихотомического подхода [Текст] / Клочков В.П., Зайцев А.В. , 2013. — 4 с.
8. Шишкина Л.П,, Мощанская Т.В., Логико-семантическая структура темы лексикона-тезауруса как основа структурирования предметно-тематического содержания текстов [Текст] / Шишкина Л.П,, Мощанская Т.В., 2015. — 9 с.
9. Васильева Н.О., Современный системный подход и проблема совместимости текстов [Текст] / Васильева Н.О., 2012. — 7 с.
10. Клочков В.П., Васильева Н.О., Дихотомический анализ основной мерности сложности общенаучной категории “совместимость” [Текст] / Клочков В.П., Васильева Н.О., 2012. — 3 с.
11. Криони Н.К., Никин А.Д., Филиппова А.В. Автоматизированная система анализа сложности учебных текстов [Текст] / Криони Н.К., Никин А.Д., Филиппова А.В., 2008. — 7 с.
12. Оборнева И.В. Автоматизированная оценка сложности учебных текстов на основе статических параметров [Текст] / Оборнева И.В., 2006. — 20 с.
13. Ajax-запрос [Электронный ресурс]. – Режим доступа: http://jquery.page2page.ru/index.php5/Ajax-%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81. – Загл. с экрана.
14. GitHub [Электронный ресурс]. – Режим доступа: https://github.com/. – Загл. с экрана.

1. Раздел педагогики, излагающий общую теорию образования и обучения. [↑](#footnote-ref-1)
2. Рыбакова Галина Раисовна - кандидат биологических наук, доцент Сибирского федерального университета Торговоэкономического института (Красноярск). [↑](#footnote-ref-2)
3. Клочков Владимир Павлович - вице-президент Международной академии фундаментального образования, руководитель её Сибирского отделения; академик Нью-Йоркской академии наук; член.-корр. Международной академии психологических наук; член.-корр. Российской академии естествознания (РАЕ); заслуженный деятель науки и образования РАЕ; доктор педагогических наук, доктор философии (PhD), доктор психологии (Psy.D), доктор делового администрирования (DBA); профессор кафедры психологии и управления. [↑](#footnote-ref-3)