|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатика, искусственный интеллект и системы управления

КАФЕДРА Системы обработки информации и управления

**РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

***К ВЫПУСКНОЙ КВАЛИФИКАЦИОННОЙ РАБОТЕ***

***НА ТЕМУ:***

**Разработка портала и рекомендательной системы по туризму**

Студент \_\_ИУ5-84Б\_\_ **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_П.К.Шимолина\_**

(Группа) (Подпись, дата) (И.О.Фамилия)

Руководитель ВКР **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Нардид\_\_\_**

(Подпись, дата) (И.О.Фамилия)

Нормоконтролер **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_Ю.Н.Кротов \_\_**

(Подпись, дата) (И.О.Фамилия)

*2023 г.*

РЕФЕРАТ

Расчётно-пояснительная записка квалификационной работы бакалавра содержит 119 страниц с учетом всех приложений. В работе есть ссылки на 14 источников

Квалификационная работа выполнена на тему «Сегментация объектов на спутниковых снимках» и посвящена созданию интернет-портала. Данная система предоставляет пользователю возможность получать актуальную информацию по разным туристическим направлениям и подбирать туры, похожие на конкретное место отдыха, читать новости туризма и общаться с другими пользователями на форуме.

Цель работы заключается в облегчении процесса поиска подходящего места отдыха, уменьшении временных затрат путем создания многофункционального портала, предоставляющего пользователю точные рекомендации, актуальные новости туризма и возможность общаться с другими путешественниками.

В процессе написания квалификационной работы бакалавра была изучена предметная область, проведён анализ методов построения рекомендательных систем, обработки текста, реализована система рекомендаций content-based в веб-приложении.

СОДЕРЖАНИЕ

[РЕФЕРАТ 2](#_Toc136355430)

[СОДЕРЖАНИЕ 3](#_Toc136355431)

[СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ 5](#_Toc136355432)

[ВВЕДЕНИЕ 6](#_Toc136355433)

[1 ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ 8](#_Toc136355434)

[1.1. Описание предметной области 8](#_Toc136355435)

[1.2. Функциональные задачи разрабатываемого веб-приложения 10](#_Toc136355436)

[1.3. Выбор критериев качества для интернет-портала 11](#_Toc136355437)

[1.4. Анализ аналогов 12](#_Toc136355438)

[2 ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ 16](#_Toc136355439)

[2.1 Рекомендательные системы и принципы работы 16](#_Toc136355440)

[2.2 Content-based 16](#_Toc136355441)

[2.3 Сравнение типов рекомендательных систем 17](#_Toc136355442)

[2.4 Подготовка данных 19](#_Toc136355443)

[2.4.1 Удаление знаков препинания, приведение к нижнему регистр 21](#_Toc136355444)

[2.4.2 Удаление стоп-слов 22](#_Toc136355445)

[2.4.3 Лемматизация 23](#_Toc136355446)

[2.4.4 Стемминг 24](#_Toc136355447)

[2.5 Выбор библиотеки для лемматизации и стемминга данных для обработки данных на русском языке 26](#_Toc136355448)

[2.6 Векторизация текста на основе модели "мешка слов" 28](#_Toc136355449)

[3 КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ 31](#_Toc136355450)

[3.1 Постановка задачи 31](#_Toc136355451)

[3.2 Инфологическая и даталогическая модели 32](#_Toc136355452)

[3.3 Разработка архитектуры системы 35](#_Toc136355454)

[3.4 Используемые при разработке технологии 36](#_Toc136355455)

[3.5 Технологическая разработка 36](#_Toc136355468)

[3.6 Разработка диаграмм в нотации IDEF0 40](#_Toc136355469)

[3.7 Разработка интерфейса взаимодействия с пользователем 43](#_Toc136355471)

[3.9 Граф диалога 50](#_Toc136355472)

[ЗАКЛЮЧЕНИЕ 51](#_Toc136355473)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 52](#_Toc136355474)

[ПРИЛОЖЕНИЕ A ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 53](#_Toc136355475)

[ПРИЛОЖЕНИЕ B ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ 63](#_Toc136355476)

СПИСОК ОБОЗНАЧЕНИЙ И СОКРАЩЕНИЙ

1. NLP – Natural Language Processing, подраздел искусственного интеллекта и математической лингвистики, посвящённый обработке текстов на естественном языке.
2. Архитектура системы – фундаментальная организация системы, реализованная в ее компонентах, их взаимосвязях друг с другом и с окружающей средой, и руководящие правила проектирования и развития системы.
3. Django – фреймворк для веб-приложения на Python.
4. MySQL – свободная реляционная система управления базами данных. Разработку и поддержку MySQL осуществляет корпорация Oracle,

ВВЕДЕНИЕ

Туризм – это вид активного отдыха, позволяющий человеку знакомиться с культурными особенностями других стран и регионов, расширять свой кругозор и проводить время с пользой тела и души. Туризм оздоравливает и расслабляет человека, но выбор места отдыха и сервиса для покупки тура может стать настоящей проблемой.

С развитием интернет-технологий наблюдается тенденция повышения интереса к этому виду активности, но проблема в том, что большинство веб-сервисов либо перегружено большим количеством ненужной пользователю информации, среди которой сложно найти нужное, либо наоборот: функциональности недостаточно и приходится обращаться к сторонним ресурсам.

Также, существует проблема выбора конкретного места, среди огромного многообразия. Каждый регион, а уж тем более страна характеризуются своим колоритом, неповторимой культурой и национальными особенностями, уникальными для каждого места. Таким образом, некоторым пользователям, необходимо не просто предложить богатый выбор мест для отдыха, но и помочь с выбором, разработав качественную рекомендательную систему. Стоит также отметить, что большое число людей выбирая себе место для отдыха, интересуются отзывами уже «побывавших»туристов. Волнует не просто оценка того или иного тура или места для отдыха, а конкретные советы в посещении достопримечательностей, наилучшего времени для поездки и тонкостей в общении с местными жителями и знакомства с их традициями. Таким образом возникает необходимость создания общего форума, где каждый сможет получить ответ на интересующий его вопрос или поделиться свои опытом.

Кроме ответов на конкретные вопросы часть пользователей хочет услышать полную историю чьего-то отдыха, возможно лично пообщавшись с определенным человеком. То есть необходимо добавить в функциональность возможность публиковать посты о совершенных путешествиях, такая функция привлечет еще больше пользователей, предоставив путешественникам возможность обраться и делиться своими эмоциями.

Мы живем в быстроменяющемся и динамичном мире, каждая страна периодически обновляет информацию для желающих посетить ее туристов, но отследить новости в большом потоке информации бывает тяжело. Приходится идти на сторонние сайты, открывая множество вкладок в браузере… Именно поэтому я считаю, что сервисам по туризму не хватает новостной ленты, держащей пользователей в курсе последних новостей тризма.

Таким образом необходимо создать интернет-портал, на котором каждый сможет найти место для отдыха, получить совет от других пользователей, узнать последние новости сферы туризма, ознакомиться с подробной информацией по каждому туристическому направлению, получить рекомендации и поделиться своим опытом и эмоциями о совершенных поездках.

1. ПОСТАНОВКА ЗАДАЧ РАЗРАБОТКИ

Необходимо проанализировать основные типы рекомендательных систем, выбрав тот, который показывает лучшие результаты.

Для взаимодействия системы с пользователем необходимо разработать UI. Кроме того, интернет-портал должен предоставлять актуальную информацию по туристическим направлениям, осуществлять поиск места по фильтрам и по названию места, предоставлять возможность задавать вопросы на форуме и писать ответы, давать рекомендации по месту отдыха, основанные на предпочтениях и интересах человека. Помимо этого, пользователи должны иметь возможность посмотреть информацию по конкретному направлению, список всех мест, новости сферы туризма.

На основании вышеизложенного, можно заключить, что задача проектирования может быть сформулирована следующим образом:

* Исследование типов рекомендательных систем;
* Анализ предметной области;
* Анализ библиотек для обработки данных и обработка данных;
* Создание рекомендательной системы;
* Разработка интернет-портала для взаимодействия пользователя с системой.

1. **Описание предметной области**

Отдых – неотъемлемая часть жизни любого человека. В нашем динамичном, суетном мире это очень важная потребность, наравне с питанием и сном. При этом, правильно спланированный отдых, удачно выбранное туристическое направление могут оставить в душе неизгладимый след из приятных, наполненных счастливыми моментами воспоминаний. Существует огромное количество мест, способных поразить вас своей красотой и неповторимостью. Большие города, например, поражают высотой небоскребов, уникальным архитектурным стилем и быстром ритмом жизни. Тихие и самобытные горные поселения захватывают дух любителей высот и крутых скал. Теплое же и ласковое море не оставит равнодушным никого, так куда же поехать, когда в мире так много прекрасных мест?!

Сфера туризма развивалась постепенно и в сейчас пришла к тому, что может удовлетворить желания самых избалованных туристов, предложить множество самых разных направлений на любой вкус.

Побывав в конкретном месте, вы наверняка захотите увидеть что-то подобное еще раз, только еще больше и великолепнее.

Так как же удовлетворить потребности каждого человека, ведь вкусы у всех разные и представления об идеальном отдыхе тоже.

Проблема выбора туристического направления сейчас, когда людям доступны неограниченные источники данных, очень актуальная в современном мире. Есть несколько способов найти подходящее место отдыха.

Можно посоветоваться с друзьями и близкими. Однако советы людей всегда субъективны и могут не подойти конкретному человеку. Кроме того, не у всех близкие люди любят путешествовать.

Еще одним вариантом является поиск мест отдыха в туристических книгах или на тематических сайтах. В таком случае появляется проблема больших временных затрат. Даже поиск нужного туристического направления на сайте с реализованной поисковой системой может занять большое количество времени вследствие большого выбора. Проблема туристических книг также заключается в ограниченности разнообразия.

Также, постоянно меняются условия въезда, получения визы, таможенного контроля. Получается кроме поиска места отдыха нужно будет отдельно искать актуальную информацию по данной теме, а это дополнительные неудобства и временные затраты.

Решением может стать веб-портал с рекомендациями, составленными под запросы конкретного пользователя. Такая система позволит пользователям изучить туристические направления в соответствии с их собственными предпочтениями и выбрать только те, которые им нравятся больше всего. Более того, сфера туризма является быстро развивающейся и претерпевает изменения почти каждый день. По этой причине необходимо, чтобы на веб-портале пользователь мог ознакомиться с актуальными новостями. Ну и наконец, часть пользователей, делая выбор хотят получить совет и отзыв уже побывавших и знающих туристов. Поэтому форум путешественников станет отличным решением этой проблемы, позволив пользователям делиться своим опытом и получать новые знания и ответы на вопросы. Графическая модель предметной области представлена в приложении А.

1. **Функциональные задачи разрабатываемого веб-приложения**

Разрабатываемая система должна обладать следующим функционалом:

* Регистрация пользователя
* Авторизация пользователя
* Просмотр информации по всем туристическим направлениям
* Поиск места отдыха по фильтрам
* Поиск места отдыха по названию
* Поиск мест отдыха доступных российским туристам
* Получение подробной информацией по каждому месту отдыха
* Ознакомление с информацией о разработчике
* Ознакомление с лентой новостей
* Ознакомление с подробной информацией о новости
* Добавление новости (для администратора)
* Удаление новости (для администратора)
* Редактирование новости (для администратора)
* Ознакомление с информацией о системе
* Ознакомление с лентой форума
* Поиск вопроса по его формулировке
* Поиск вопроса по тегу

Просмотр конкретного вопроса и ответов на него

* Создание нового вопроса на форуме
* Написание ответа на существующий вопрос форума
* Удаление вопроса на странице форума (для администратора)
* Удаление ответа на странице форума (для администратора)
* Получение рекомендации по конкретному туристическому направлению.

1. Выбор критериев качества для интернет-портала

Для того, чтобы оценить качество разрабатываемой системы рассмотрим следующие критерии, представленные в таблице 1:

Таблица 1 – Критерии качества

| **Код** | **Название критерия** |
| --- | --- |
| К1 | Объективность рекомендаций |
| К2 | Многообразие фильтров для поиска |
| К3 | Возможность общения с другими пользователями |
| К4 | Удобство пользования |
| К5 | Возможность получить актуальные новости туризма |

Также необходимо оценить коэффициенты важности критериев. Для этого будем использовать метод бальной оценки [1] (таблица 2).

Таблица 2 – Метод балльной оценки

| **Код** | **Название критерия** | **Балл** | **Коэффициент важности локального критерия (** |
| --- | --- | --- | --- |
| К1 | Объективность рекомендаций | 80 | 0,2 |
| К2 | Многообразие фильтров для поиска | 60 | 0,15 |
| К3 | Возможность общения с другими пользователями | 90 | 0,225 |
| К4 | Удобство пользования | 100 | 0,25 |
| K5 | Возможность получить актуальные новости туризма | 70 | 0,175 |

Коэффициент важности локального критерия по методу балльной оценки ( высчитываются по следующей формуле :

где – балл критерия,

– количество критериев.

1. Анализ аналогов

В качестве аналогов будем рассматривать следующие системы:

1. Travelata.ru. Онлайн-агрегатор туров. Сравнивает предложения по всем туроператорам и показывает выгодные путевки на заданные числа. Рекомендации строятся на основе местоположения просматриваемого тура.
2. LevelTravel. Сервис покупки и бронирования туров онлайн. Собирает предложения от разных туроператоров и анализирует их. Рекомендации формируются исходя из страны путешествия и цены тура.
3. Onlineturs. Интернет магазин путешествий. Рекомендованные туры подбираются по ближайшим отелям.
4. Youtravel.me. Сервис по поиску и покупкке туров по России. Рекомендации строятся на основе типа тура (автотур, фототур и тд.).

Аналоги рассмотрены в таблице 3.

Таблица 3 – Аналоги

| **Код** | **Название аналога** |
| --- | --- |
| В1 | Travelata.ru |
| В2 | LevelTravel |
| В3 | Onlineturs |
| В4 | Youtravel.me |

Далее будем проводить сравнение аналогов нашей системой (В5). Для этого будем использовать метод взвешенной суммы [1] (таблица 4).

Таблица 4 – Расчет весовых коэффициентов

| **Код критерия** | **Критерий** | **Весовой коэффициент** | **В1** | **В2** | **В3** | **В4** | **В5** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| К1 | Объективность рекомендаций | 0,2 | хорошо | удовлетворительно | хорошо | отлично | отлично |
| К2 | Многообразие фильтров для поиска | 0,15 | отлично | отлично | очень хорошо | удовлетворительно | хорошо |
| К3 | Возможность общения с другими пользователями | 0,225 | отсутствует | отсутствует | отсутствует | отсутствует | есть |
| К4 | Удобство пользования | 0,25 | очень хорошо | хорошо | хорошо | хорошо | отлично |
| K5 | Возможность получить актуальные новости туризма | 0,175 | есть | отсутствует | отсутствует | отсутствует | есть |

Оценим исходные варианты на Парето-оптимальность (таблица 5). По итогу все варианты являются Парето-оптимальными, поэтому можем продолжить их сравнение [5] (таблица 5).

Таблица 5 – Сравнение вариантов на Парето-оптимальность

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **В1** | **В2** | **В3** | **В4** | **В5** |
| **В1** | 0 | 0 | 0 | 0 | 0 |
| **В2** | 0 | 0 | 0 | 0 | 0 |
|  | **В1** | **В2** | **В3** | **В4** | **В5** |
| **В3** | 0 | 0 | 0 | 0 | 0 |
| **В4** | 0 | 0 | 0 | 0 | 0 |
| **Результат сравнения** | 0 | 0 | 0 | 0 | 0 |
| **Парето-оптимальность варианта** | да | да | да | да | да |

В таблице 6 представлена вербально-числовая шкала для части критериев.

Таблица 6 – Вербально-числовая шкала для К1, K2 и К4

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Отлично** | **Очень хорошо** | **Хорошо** | **Удовлетворительно** | **Плохо** |
| 1 | 0,85 | 0,75 | 0,6 | 0,5 |

Таблица 7 поясняет значение критериев K3 и K5.

Таблица 7 – Вербально-числовая шкала для К3 и К5

| **Отсутствует** | **Есть** |
| --- | --- |
| 0 | 1 |

В таблице 8 можно ознакомиться с нормированными значениями.

Таблица 8 – Нормированные значения сравниваемых вариантов

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Код фактора** | **Весовой коэффициент** | **Значения показателей** | | | | |
| **В1** | **В2** | **В3** | **В4** | **В5** |
| К1 | 0,2 | 0,75 | 0,6 | 0,75 | 1 | 1 |
| К2 | 0,15 | 1 | 1 | 0,85 | 0,6 | 0,75 |
| К3 | 0,225 | 0 | 0 | 0 | 0 | 1 |
| К4 | 0,25 | 0,85 | 0,75 | 0,75 | 0,75 | 1 |
| К5 | 0,175 | 1 | 0 | 0 | 0 | 1 |
|  | | 0,6875 | 0,4575 | 0,465 | 0,4775 | 0,9625 |

Ранжирование вариантов по предпочтительности:

В5 > В1 > В4 > В3 > В2

Применив метод взвешенной суммы нормированных показателей сравнения мы получили, что разрабатываемая система является один из самых оптимальных вариантов среди рассмотренных аналогов.

1. **ИССЛЕДОВАТЕЛЬСКАЯ ЧАСТЬ**
   1. Рекомендательные системы и принципы работы

Рекомендательные системы — это алгоритмы, подбирающие релевантные товары и услуги на основе данных о пользователе. На сегодня используются два основных подхода в рекомендательных системах: коллаборативная фильтрация и модель, основанная на контенте. Есть еще два вида рекомендательных систем – это гибридная рекомендательная система, совмещающая в себе функционал упомянутых выше методов, а также существует рекомендательные системы, основанные на знаниях, нов рамках этой работы мы их рассматривать не будем.

Цель коллаборативной фильтрации — найти пользователя, оценившего конкретный объект, и рассчитать коэффициент корреляции векторов его оценок всех объектов в базе данных. Для этого часто используют метод [k-ближайших соседей](https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D1%82%D0%BE%D0%B4_k-%D0%B1%D0%BB%D0%B8%D0%B6%D0%B0%D0%B9%D1%88%D0%B8%D1%85_%D1%81%D0%BE%D1%81%D0%B5%D0%B4%D0%B5%D0%B9). В центре же модели, **основанной на контенте**, находится объект. Для работы алгоритма оценки пользователя не нужны. Модели важно знать все свойства, относящиеся к объекту: сезон, виза, и т. д.

2.2 Content-based

Подробнее рассмотри данный вид рекомендательной системы.

Этот метод менее универсальный, чем метод коллаборативной фильтрации, так как он зависит от данных предметной области. Достоинством метода является то, что маленькое количество оценок никак не повлияет на качество рекомендаций. То есть можно внедрять рекомендательную систему данного типа только на старте работы [11].

На рисунке 1 продемонстрирована разница между content-based, collaborative filtering и гибридным методом.

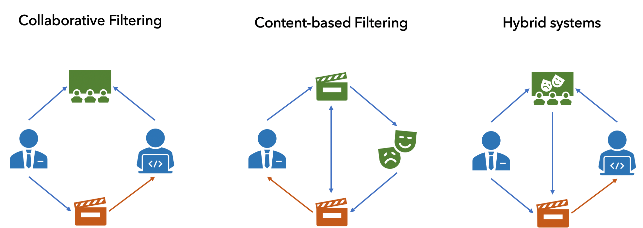


Рисунок 1 – Принципы рекомендаций

Для content-based системы не нужны какие-то дополнительные данные о действиях других пользователей. Как было сказано ранее для collaborative filtering систем необходимо хранить большие матрицы N\*N, где N - количество пользователей, и матрицу размером M\*M, где M - количество объектов рекомендаций.

2.3 Сравнение типов рекомендательных систем

Необходимо выбрать тип рекомендательной системы, дающий наиболее четкие рекомендации.

Рассмотрим ранее описанные виды рекомендательных систем. В таблице 9 представлены типы систем, участвующие в сравнении.

Таблица 9 – Сравниваемые архитектуры

| **В** | **Вариант системы** |
| --- | --- |
| В1 | Content-based |
| В2 | Collaborative filtering |
| В3 | Hybrid methods |

Будем сравнивать эти типы по набору критериев, которые представлены в таблице 10.

Таблица 10 – Критерии сравнения типов систем

| **Код** | **Название критерия** | **Описание** |
| --- | --- | --- |
| К1 | «Холодный старт» | Оценка возможности внедрения системы при отсутствии данных о предпочтениях пользователей; оценка быстрого старта. |
| К2 | Количество собираемых данных | Оценка собираемого объёма данных. |
| К3 | Индивидуальность рекомендация | Оценка индивидуальности рекомендаций для конкретного пользователя. |
| К4 | Адаптивность | Оценка изменения рекомендаций в случае, например, смены интересов пользователя. |

При помощи метода балльной оценки [1] оценим коэффициенты важности критериев (таблица 11).

Таблица 11 – Метод балльной оценки

| **Код критерия** | **Название критерия** | **Балл** | **Коэффициент важности локального критерия по методу бальной оценки (** |
| --- | --- | --- | --- |
| К1 | «Холодный старт» | 100 | 0,333 |
| К3 | Количество собираемых данных | 50 | 0,167 |
| К4 | Индивидуальность рекомендация | 90 | 0,3 |
| К5 | Адаптивность | 60 | 0,2 |

По методу бальной оценки коэффициент важности локального критерия ( высчитываются по формуле (1).

В качестве интегрального критерия выбора наилучшего варианта используем колонки взвешенную сумму, рассчитанная в таблице 12. Будем считать, что чем ближе число к 1, тем больше удовлетворяет система по данному критерию, чем в других вариантах.

Таблица 12– Нормированные значения сравниваемых вариантов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Код критерия** | **Весовой коэффициент** | **Значения показателей** | | |
| **В1** | **В2** | **В3** |
| К1 | 0,333 | 0,7 | 0 | 0,3 |
| К2 | 0,167 | 0,6 | 0,4 | 0 |
| К3 | 0,3 | 0,65 | 0,1 | 0,25 |
| К4 | 0,2 | 0,8 | 0 | 0,2 |
|  | | **Значения показателей** | | |
| **В1** | **В2** | **В3** |
|  | | 0,64255 | 0,1123 | 0,24515 |

Ранжирование вариантов по предпочтительности:

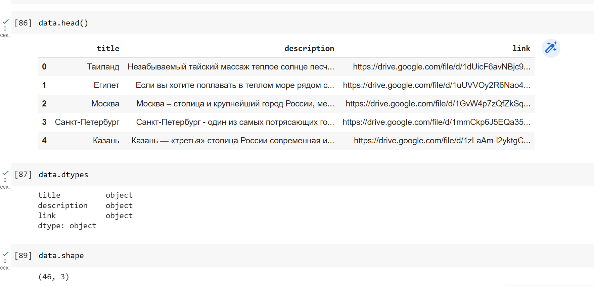
В1 > В3 > В2

Применив метод взвешенной суммы мы увидели, что, лучшим видом системы рекомендаций является Content-based. Следовательно, этот метод будет применяться в разрабатываемой рекомендательной системе.

2.4 Подготовка данных

Так как подходящего набора данных найти не удалось, датасет формировался мной самостоятельно. Для рекомендательной системы в набор данных были добавлены описания мест на русском языке и ссылки на фотографии мест в облачном хранилище. Описания будем использовать для content-based рекомендаций.

На фрагменте кода 1 представлен датасет, с которым ведётся работа, и его размер – 46 туристических мест.



Фрагмент кода 1. Датасет и структура

Естественный язык своей первоначальном варианте из-за обилия регистров, знаков препинания, падежей плохо поддается обработке. Одно и то же слово, например, в разных падежах или регистрах может восприниматься по-разному. Чтобы от этого избавиться необходимо произвести предварительную обработку текста. Так как под разные задачи требуется разная обработка одного и того же текста, нужно выбрать наилучший вариант, подходящий именно нам.

Далее рассмотри разные способы обработки текста.

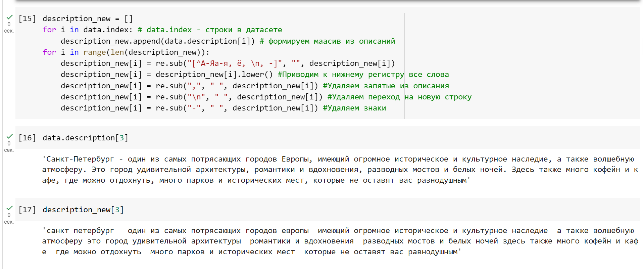
* + 1. Удаление знаков препинания, приведение к нижнему регистр

Один из этапов обработки – приведение всех слов исходного текста к одному регистру. Это сделает модель более точной и позволит избежать ситуаций, при которых одно и то же слово в разных регистрах имеет разный вес для нашей модели.

По той же причине необходимо удалить пунктуацию. В нашей задаче нас интересует только суть текста, а не его эмоциональная окраска. Знаки препинания могут помешать работе модели.

Пропустим текст «Санкт-Петербург - один из самых потрясающих городов Европы, имеющий огромное историческое и культурное наследие, а также волшебную атмосферу. Это город удивительной архитектуры, романтики и вдохновения, разводных мостов и белых ночей», и получим следующий результат:

«санкт петербург один из самых потрясающих городов европы имеющий огромное историческое и культурное наследие а также волшебную атмосферу это город удивительной архитектуры романтики и вдохновения разводных мостов и белых ночей» (фрагмент кода 2).



Фрагмент кода 2. Описание после удаления пунктуации и приведения к одному регистру

Мы привели текст описания к нижнему регистру и удалили знаки препинания, которые могли помешать работе нашей модели. Теперь мы можем перейти к следующему этапу обработки.

* + 1. Удаление стоп-слов

Стоп-слова – это слова, не несущие особого смысла, поэтому от будет логично от них избавиться. Нагромождение таких слов может уменьшить точность рекомендаций конечной модели.

Сделать это можно с помощью библиотеки NLTK [12].

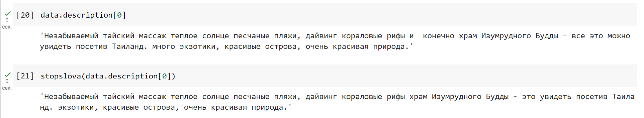
Код с помощью которого будут удалены стоп-слова из описаний приведен ниже, на фрагменте кода 3:



Фрагмент кода 3. Метод удаления стоп-слов

Возьмем следующее описание «Незабываемый тайский массаж теплое солнце песчаные пляжи, дайвинг крокодиловая ферма и конечно храм Изумрудного Будды - все это можно увидеть посетив Таиланд». После удаление стоп-слов получим следующий результат:

«Незабываемый тайский массаж теплое солнце песчаные пляжи, дайвинг крокодиловая ферма храм Изумрудного Будды - это увидеть посетив Таиланд» (фрагмент кода 4).



Фрагмент кода 4. Описание после удаления стоп-слов

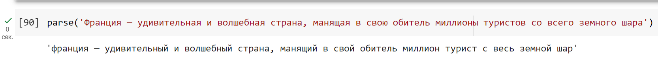
* + 1. Лемматизация

Кроме приведения слов к одному регистру, удаления пунктуации и стоп слов необходимо также привести все слова к начальной форме. Лемматиза́ция — процесс приведения словоформы к лемме — её нормальной форме.

Подобную предобработку могут проводить разные библиотеки.

Одной из самых популярных является PyMorphy2[3] — она встречается почти в каждом решении, которое можно найти в сети.

Дадим на вход данной функции следующее описание «Франция — удивительная и волшебная страна, манящая в свою обитель миллионы туристов со всего земного шара» и получим на выходе следующее – «франция — удивительный и волшебный страна, манящий в свой обитель миллион турист с весь земной шара» (фрагмент кода 5).



Фрагмент кода 5. Результат обработки текста в PyMorphy2

Кроме анализатора PyMorphy2 есть библиотека SpaCy [4]. Для выбора наилучшего алгоритма считаю необходимым сравнить обработку через PyMorphy2 и через SpaCy.

Если в функцию обработки SpaCy подать тот же текст «Франция — удивительная и волшебная страна, манящая в свою обитель миллионы туристов со всего земного шара», то на выходе получим «франция — удивительный и волшебный страна , манить в свой обитель миллион турист со весь земной шар» (фрагмент кода 6).



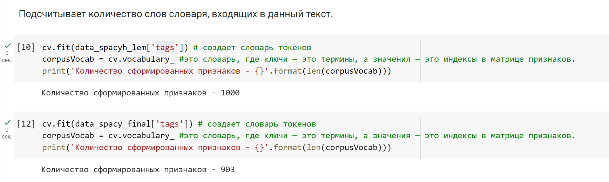
Фрагмент кода. 6. Результат обработки данных в SpaCy

Из данного примера видно, что SpaCy хуже справился с задачей. Причастия и предлоги обрабатываются хуже по сравнению с PyMorphy2.

* + 1. Стемминг

Последний этап обработки текста, позволяющий получить основу слова, называется стеммингом. Это грубый способ обработки, приводящий к потере части словаря, за счет отсечения словообразовательных суффиксов. Я считаю необходимым сравнить работу рекомендательной системы на двух датасетах – с применением стемминга и без него (только с ламматизацией).

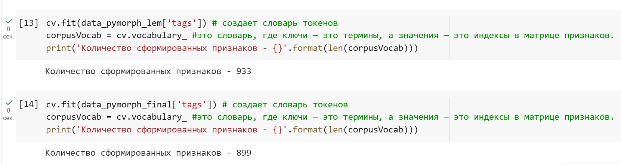
На фрагменте кода 7 показано численное изменение словаря после стемминга на данных, в которых использовалась библиотека SpaCy при лемматизации.



Фрагмент кода 7. Сформированные признаки до и после стемминга для данных.

Видно, что результат словаря уменьшился, в дальнейшем посмотрим, как это повлияет на рекомендации.

На фрагменте кода 8 показано численное изменение словаря после стемминга на данных, в которых использовалась библиотека PyMorphy2 при лемматизации.



Фрагмент кода 8. Сформированные признаки до и после стемминга для данных.

Следующим шагом считаю необходимым выбрать библиотеку для стемминга и лемматизации.

* 1. Выбор библиотеки для лемматизации и стемминга данных для обработки данных на русском языке

Рассмотри библиотеки PyMorphy2 и SpaCy. Они и будут сравниваемыми вариантами (таблица 13).

Таблица 13 – Сравниваемые библиотеки

| **В** | **Вариант системы** |
| --- | --- |
| В1 | PyMorphy2 |
| В2 | SpaCy |

Для сравнения этих библиотек будем использовать критерии, приведенные в таблице 14.

Таблица 14 – Критерии сравнения типов систем

| **Код** | **Название критерия** | **Описание** |
| --- | --- | --- |
| К1 | Качество лемматизации | Оценка качества лемматизации по количеству сформированных признаков. |
| К2 | Время работы | Оценка времени обработки данных. |
| К3 | Разница в количестве сформированных признаков | Оценка количества сформированных признаков словаря без применения стемминга и с применением. |

При помощи методы балльной оценки [1] оценим коэффициенты важности критериев (таблица 15).

Таблица 15 – Метод балльной оценки

| **Код критерия** | **Название критерия** | **Балл** | **Коэффициент важности локального критерия по методу бальной оценки (** |
| --- | --- | --- | --- |
| К1 | Качество лемматизации | 90 | 0,47 |
| К2 | Время работы | 30 | 0,16 |
| К3 | Разница в количестве сформированных признаков | 70 | 0,37 |

Далее рассчитаем весовые коэффициенты (таблица16).

Таблица 16 – Расчёт весовых коэффициентов

| **Код критерия** | **Весовой коэффициент** | **Значения показателей** | |
| --- | --- | --- | --- |
| **В1** | **В2** |
| К1 | 0,47 | удовлетворительно | отлично |
| К2 | 0,16 | хорошо | оч хорошо |
| К3 | 0,37 | 34 | 97 |

Вербально числовая шкала представлена в таблице 17.

Таблица 17 – Вербально-числовая шкала для К1-К2

| **Отлично** | **Очень хорошо** | **Хорошо** | **Удовлетворительно** | **Плохо** |
| --- | --- | --- | --- | --- |
| 1 | 0,85 | 0,75 | 0,6 | 0,5 |

По методу бальной оценки коэффициент важности локального критерия ( высчитываются по формуле (1).

В качестве интегрального критерия выбора наилучшего варианта используем колонки взвешенную сумму, рассчитанная в таблице 18.

Таблица 18 – Нормированные значения сравниваемых вариантов

| **Код критерия** | **Весовой коэффициент** | **Значения показателей** | |
| --- | --- | --- | --- |
| **В1** | **В2** |
| К1 | 0,47 | 0,6 | 1 |
| К2 | 0,16 | 0,75 | 0,85 |
| К4 | 0,2 | 0,39 | 1 |
|  | | 0,48 | 0,806 |

Ранжирование вариантов по предпочтительности:

В2 > В1

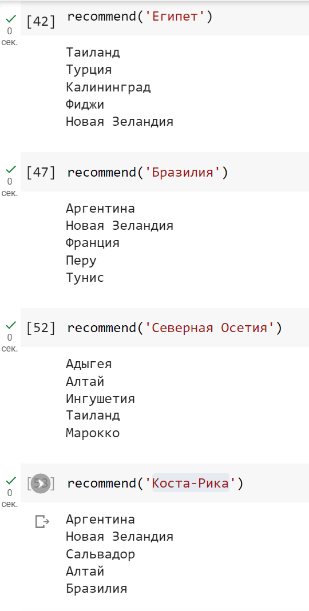
Применив метод взвешенной суммы мы убедились, что лучшей библиотекой для лемматизации и стемминга является SpaCy. Следовательно, эту билиотеку и будем применять в разрабатываемой рекомендательной системе.

* 1. Векторизация текста на основе модели "мешка слов"

Теперь проведем векторизацию двух датасетов с описаниями. Над одним из них была выполнена процедура леммантизации и стемминга, над другим – только лемантизация. Важно узнать, в каком случае рекомендательная система будет давать более четкие рекомендации.

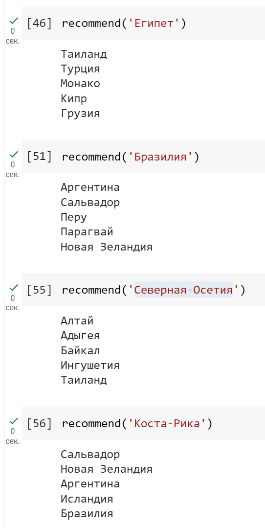
Процесс векторизации будет подробно рассмотрен в 3 главе. Здесь нас интересует сравнение рекомендательных систем, построенных на датасете с применением стемминга и без него.

Дадим на вход несколько названий мест и посмотрим на результат (фрагмент кода 9).



Фрагмент кода 9. Рекомендации полученные на основе датасета без использования стемминга.

Теперь посмотрим рекомендательную систему на основе датасета, к описанию которого применялась процедура стемминга. Так как процесс создания рекомендательной системы аналогичен приведенному выше, предлагаю сразу перейти к сравнению результатов (фрагмент кода 10).



Фрагмент кода 10. Рекомендации полученные на основе датасета с использованием стемминга.

Проанализируем полученные результаты. Сравнительный анализ работы рекомендательных систем находится в приложении А (таблица 19).

На основе приведенного анализа можно сказать о том, что рекомендательная система показывает более точные результаты, если при обработке текста был использован стемминг. Таким образом рекомендательная система content-based будет разработана с использованием библиотеки SpaCy с применением лемантизации и стемминга.

1. КОНСТРУКТОРСКО-ТЕХНОЛОГИЧЕСКАЯ ЧАСТЬ

**3.1 Постановка задачи**

В процессе исследований типов рекомендательных систем для нашего набора данных было выявлено, что лучше рекомендации сделать на методе content-based. Для того, чтобы пользователь мог максимально быстро найти наилучшее для себя место отдыха в системе должны быть предусмотрены следующие функции.

* Регистрация пользователя
* Авторизация пользователя
* Просмотр информации по всем туристическим направлениям
* Поиск места отдыха по фильтрам
* Поиск места отдыха по названию
* Поиск мест отдыха доступных российским туристам
* Получение подробной информацией по каждому месту отдыха
* Ознакомление с информацией о разработчике
* Ознакомление с лентой новостей
* Ознакомление с подробной информацией о новости
* Добавление новости (для администратора)
* Удаление новости (для администратора)
* Редактирование новости (для администратора)
* Ознакомление с информацией о системе
* Ознакомление с лентой форума
* Поиск вопроса по его формулировке
* Поиск вопроса по тэгу
* Просмотр конкретного вопроса и ответов на него
* Создание нового вопроса на форуме
* Написание ответа на существующий вопрос форума
* Удаление вопроса на странице форума (для администратора)
* Удаление ответа на странице форума (для администратора)
* Получение рекомендации по конкретному туристическому направлению

3.2 Инфологическая и даталогическая модели

Для того, чтобы подбирать рекомендации с помощью content-based метода, необходимо хранить информацию о туристических направлениях. Каждое туристическое направление связано с определенным типом визы, частью света, сезоном для посещения. Новости будут хранится в отдельной таблице. Для форума необходимы таблицы вопросов, ответов, тегов, профиля, связанных с системной таблицей пользователей. Таким образом, можно выделить следующие сущности:

* Пользователи (системная таблица Django)
* Профили
* Теги
* Вопросы
* Ответы
* Виза
* Места для путешествий
* Сезон
* Часть света
* Новости

ER-диаграмма позволяет моделировать объекты предметной, отображая отношения между ними. Инфологическая модель изображена на рисунке 2.

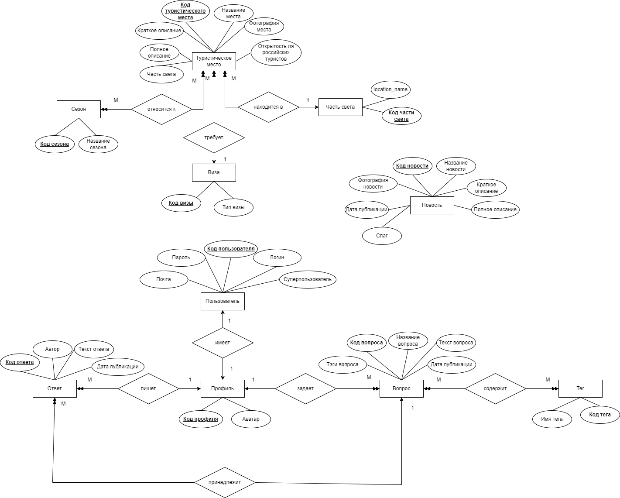


Рисунок 2 – Инфологическая модель

Сущность – это объект в базе данных, который можно выделить, исходя из сути предметной области. У каждой сущности есть набор атрибутов. Все сущности и их атрибуты, предназначающиеся для описания сущностей, описаны в таблице 20 приложения А. Каждая сущность должна иметь свой уникальный идентификатор – первичных ключ.

В разработанной БД есть все 3 типа связей между таблицами. Первый тип 1-М (один ко многим). Такой тип связи говорит о том, что одной записи первой таблицы может соответствовать несколько записей из второй таблице, в то время как одной записи второй таблице может соответствовать только одна запись первой. Например, каждому туристическому направлению соответствует только один тип визы, но с одним и тем же типом визы связано несколько мест. Связь М-М (многие ко многим). При таком типе связи одной записи каждой таблицы соответствует несколько записей другой таблицы. Например, у вопроса может быть несколько тэгов, при этом один и тот же тэг может соответствовать нескольким вопросам. Связь 1-1 (один ко одному). То есть каждому элементу одной таблицы соответствует только один элемент другой. Для связей БД была создана отдельная таблица. Датологическая модель изображена на рисунке 3.

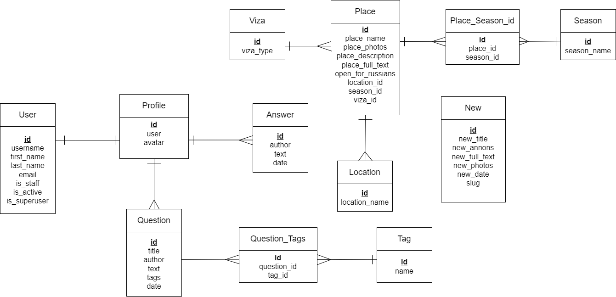


Рисунок 3 – Даталогическая модель

Как можно заметить, в даталогическую модель были введены две дополнительные таблицы для реализации связи многие-ко-многим. Это Question\_Tags и Place\_Season\_Id. Структура этих таблиц состоит из трех полей. Уникальный идентификатор таблицы и два поля, относящиеся к связываемым таблицам.

**3.3 Разработка архитектуры системы**

Рассмотрим архитектуру рекомендательной системы, представленную на рисунке 4.



Рисунок 4 – Архитектура рекомендательной системы

Работа данного метода зависит от выбранных данных, на которых будут строиться рекомендации [2]. После обработки данных идет процесс векторизации. В систему поступает название места отдыха, к которому пользователь желает получить рекомендацию, и происходит ранжирование, то есть запускается процесс подбора похожих мест на основе вектора. В итоге, выводятся рекомендуемые места отдыха (рисунок 5).



Рисунок 5 – Архитектура блока «Предобработка»

* 1. **Используемые при разработке технологии**

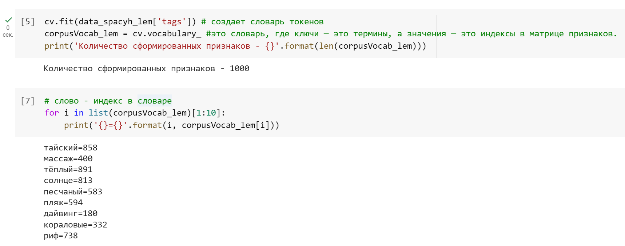
При разработке системы использовались следующие технологии:

1. Python 3.11 [литература] — это высокоуровневый язык программирования, который отличается простотой и понятностью синтаксиса. Он широко используется для разработки веб-приложений, анализа данных и многих других областей. В данной работе используется для разработки бэкэнда и обучения модели сегментации.
2. JavaScript 16.14.2 [литература] — это высокоуровневый язык программирования, который применяется для создания интерактивных веб-страниц и веб-приложений. Используется для разработки фронтэнда.
3. Google Colab (Colaboratory) [литература] — это сервис от Google, который предоставляет среду для разработки и выполнения кода на языке Python в облачной среде. Используется для обучения модели.
4. Visual Studio Code (VS Code) [литература] — это современный редактор кода, разработанный для кроссплатформенной разработки веб-приложений. Он отличается расширяемостью и поддержкой множества языков программирования и технологий. Использовался для разработки веб-сервиса.
5. Django [литература] — это фреймворк для веб-разработки на языке Python. Он предоставляет инструменты и шаблоны для создания веб-приложений, обладает мощными функциями, такими как ORM [сокращение] для работы с БД [сокращение], встроенная административная панель, система аутентификации и авторизации, управление URL. Используется для создания бэкенд-части.
6. Django REST framework [литература] — это фреймворк для создания веб API на основе Django. Он предоставляет удобные инструменты и функции для быстрой разработки RESTful API [сокращение]. Используется для разработки API [сокращение].
7. React [литература] — это библиотека JavaScript, которая используется для создания пользовательских интерфейсов веб-приложений. Она отличается своей декларативной и компонентной структурой. React позволяет создавать переиспользуемые компоненты, которые отвечают за отображение данных и реагирование на изменения состояния приложения. Используется для разработки компонентов веб-сервиса.
8. React Router DOM [литература] — это набор библиотеки для маршрутизации веб-приложений на основе React. Используется для создания динамических маршрутов, которые позволяют пользователям навигироваться по различным страницам приложения.
9. Bootstrap [литература] — это фреймворк, который предоставляет готовые стили для создания адаптивных и красивых веб-интерфейсов. Bootstrap включает в себя множество компонентов, сеток, стилей и JavaScript плагинов. Используется для разработки интерфейса сервиса.
10. NumPy — Библиотека для работы с многомерными массивами и матрицами, предоставляющая множество функций для математических операций. Используется для преобразования изображения в формате матрицы.
11. Pandas — Библиотека для обработки и анализа данных, предоставляющая структуры данных и инструменты для работы с таблицами и временными рядами. Используется для перевода изображения в датафрейм.
12. Matplotlib — Библиотека для визуализации данных, позволяющая создавать различные типы графиков и диаграмм. Используется для визуализации результатов обучения модели.
13. Scikit-Learn — Библиотека, предоставляющая инструменты для машинного обучения. Используется для разделения данных на обучающие и тестовые наборы.
14. PyTorch — Библиотека машинного обучения с открытым исходным кодом, предоставляющая гибкие инструменты для создания нейронных сетей и обучения моделей.
15. Torchvision — Подмодуль PyTorch для работы с компьютерным зрением, содержащий инструменты для загрузки и обработки изображений.
16. Pillow — Библиотека для работы с изображениями, предоставляющая функционал для открытия, обработки и сохранения изображений.
17. OpenCV — Библиотека для компьютерного зрения и обработки изображений, предоставляющая широкий спектр алгоритмов и функций.
18. Albumentations — Библиотека для аугментации изображений, предоставляющая различные трансформации для улучшения качества данных в обучающем наборе.
19. Time — Встроенная библиотека Python для работы со временем и временными задержками.
20. OS — Встроенная библиотека Python для работы с операционной системой, предоставляющая функции для взаимодействия с файловой системой и окружением.
21. TQDM — Библиотека для отображения прогресса выполнения итераций в цикле.
22. Torchsummary — Библиотека для отображения сводки модели нейронной сети, включая информацию о количестве параметров и размере выхода слоев.
23. Segmentation Models PyTorch — Библиотека для сегментации изображений с использованием PyTorch, предоставляющая реализации различных моделей для задач сегментации.

**3.5 Технологическая разработка**

Далее мы должны сделать векторизацию с помощью класса CountVectorizer.

CountVectorizer – это метод преобразования текста в числовые данные; подсчитывает слова.

Посчитаем, сколько раз встречается каждое слово в описании туристических направлений, создадим массив N\*N, где N – количество сформированных признаков словаря (фрагмент кода 11).

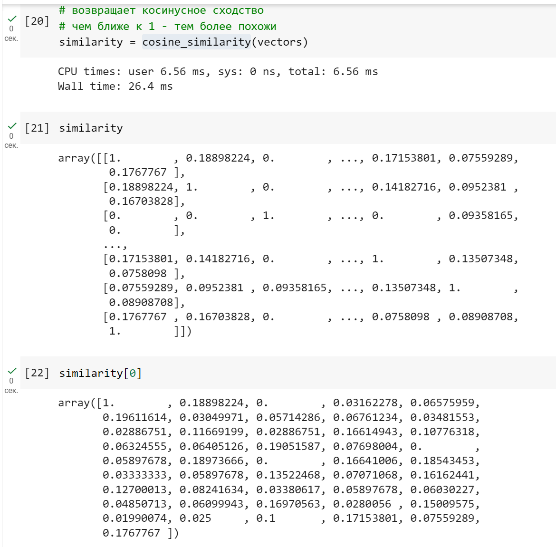
Фрагмент кода 11. Формирование признаков словаря

Далее найдём косинусное расстояние. Косинусное расстояние (сходство) - используется для измерения косинуса угла между ними. Формула выглядит следующим образом:

где: – векторы,

– угол между векторами.

Код построения массива косинусового расстояния показан на фрагменте кода 12.



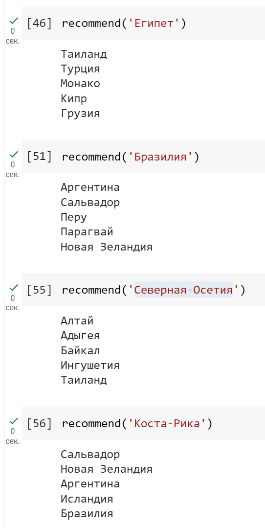
Фрагмент кода 12. Вычисление косинусового расстояния.

Напишем функцию, которая на вход будет принимать название места, а возвращать – 3 рекомендаций, по данному туристическому направлению (фрагмент кода 13).



Фрагмент кода 13. Функция получения рекомендаций.

Результат работы функции получения рекомендаций показан на фрагменте кода 14.



Фрагмент кода. 14. Результат работы функции рекомендаций.



**3.6 Разработка диаграмм в нотации IDEF0**

Для описания функционала разработанного интернет-портала были созданы диаграммы в нотации IDEF0. Работа отдельных блоков представлена ниже (рисунки 6-10). Декомпозиция всей системы имеется в приложении А.

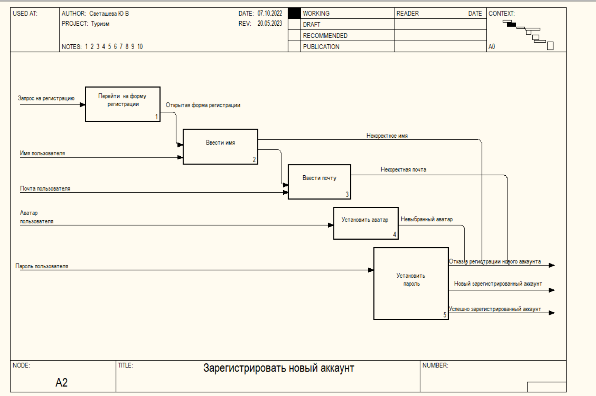


Рисунок 6 – Блок регистрации

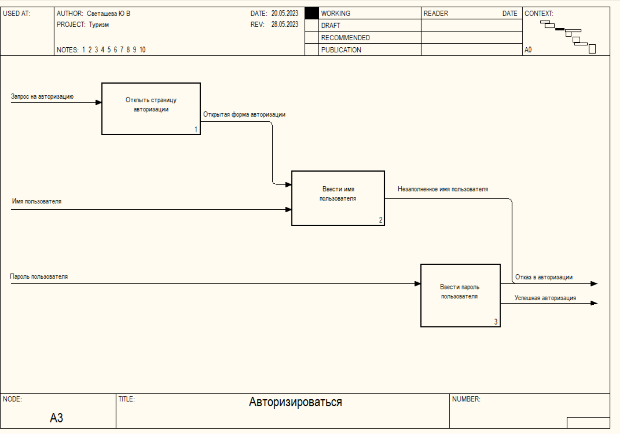


Рисунок 7 – Блок авторизации

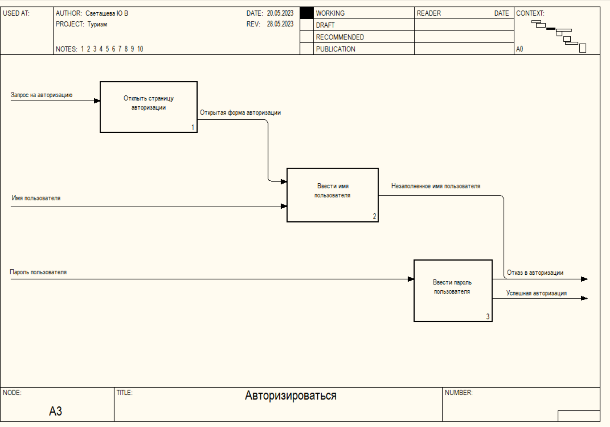


Рисунок 8 – Блок работы с главной страницей

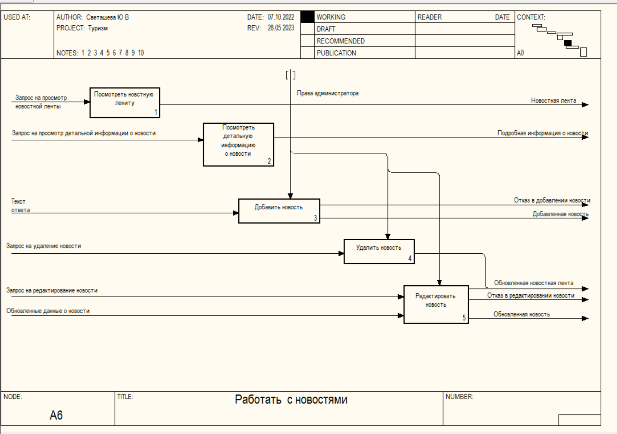


Рисунок 9 – Блок работы с новостями

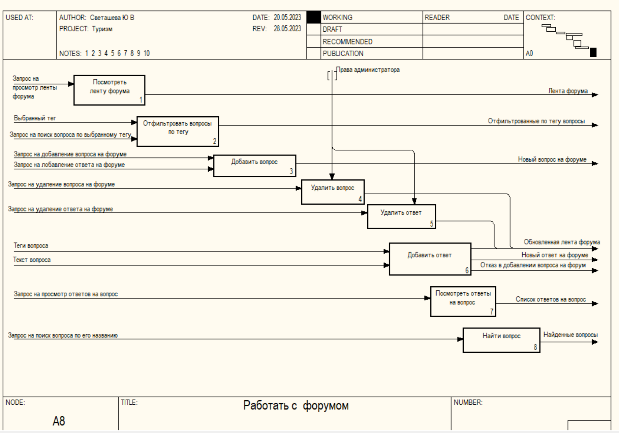


Рисунок 10 – Блок работы с форумом

**3.7 Разработка интерфейса взаимодействия с пользователем**

Теперь необходимо создать приятный и удобный интерфейс, чтобы логика взаимодействия с интернет-порталом была пользователю интуитивно понятна. Важно заметить, что для работы с форумом и получения рекомендаций пользователь должен быть авторизирован. Незарегистрированные пользователи имеют доступ только к поиску мест по фильтрам, просмотр новостей и информации о системе и разработчике. Это нужно, чтобы заинтересовать новых пользователей. Таким образом еще не авторизованный посетитель сможет ознакомиться с информацией о туристических направлениях, новостях туризма, но у него не будет доступа к форуму и персональным рекомендациям и ленте форума.

Сервисная часть приложения была разработана с помощью фреймворка Django. Пользовательская часть создана на HTML5, CSS и библиотеки Bootstrap.

Для более удобной работы с системой была добавлена навигационная панель. В ней есть ссылка на основные блоки приложения: главная страница, страница рекомендаций, новостная лента, лента форума, страницы с информацией о системе и разработчике.

Для неавторизованного пользователя самые важные разделы – работа с новостями, информация о системе и разработчике, главная страница, где пользователь может ознакомиться с туристическими направлениями. (рисунок 11).



Рисунок 11 – Навигационная модель неавторизованного пользователя

Авторизованному пользователю доступен более широкий функционал. Кроме всего вышеперечисленного он так же имеет доступ к форуму и получению рекомендаций (рисунок 12).



Рисунок 12 – Навигационная модель авторизованного пользователя

Для переключения страниц в некоторые разделы веб-портала была добавлена пагинация (рисунок 13). В пагинации отображается информации об общем количестве страниц и о номере текущей страницы.

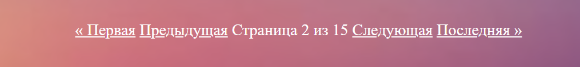


Рисунок 13 – Пагинация

Места передаются на страницу в виде списка, разбиваясь на страницы в соответствии с настройками пагинации. Каждая страница – это блок div с привязанными к нему стилями (рисунок 14).

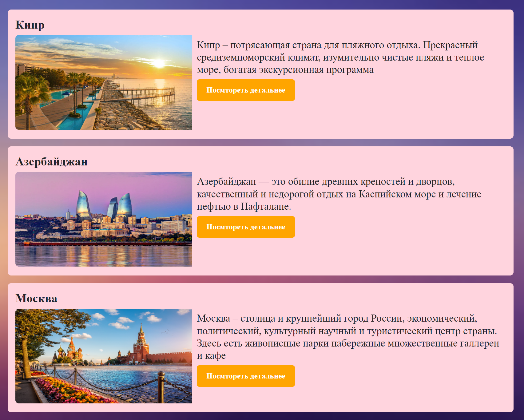


Рисунок 14 – Список мест на главной странице

На главной странице, а также на форуме и в разделе рекомендаций есть форма поиска. Она состоит из поля ввода (input) и кнопки поиска (рисунок 15).

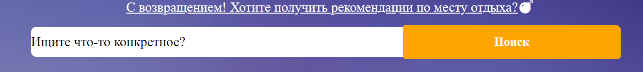


Рисунок 15 – Список мест на главной странице

Также на главной странице имеется боковая панель. Она представляет из собой форму, состоящую из checkbox-ов и span-блоков (рисунок 16).

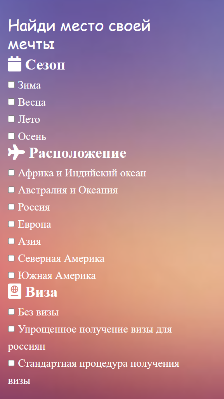


Рисунок 16 – Боковая панель фильтров

При нажатии на кнопку «Посмотреть детальнее» пользователь попадет на страницу с подробной информацией о месте. Здесь указаны правила въезда-выезда, информация о погоде, местных традициях, о которых стоит знать и многое другое (рисунок 17). Весь интерфес здесь выполнен с помочью стандартных компонентов HTML (div, img, a и др.)

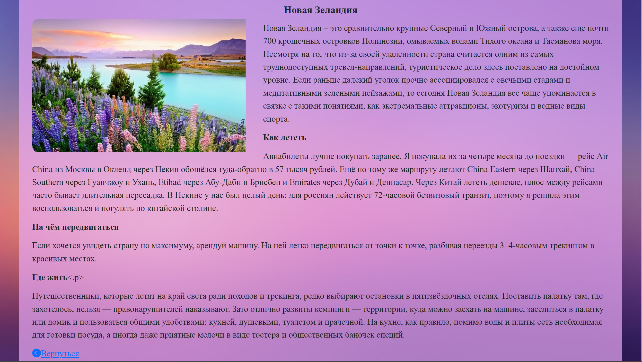


Рисунок 17 – Страница с подробной информацией о туристическом направлении

Таким образом любой пользователь сможет осуществить поиск места и подробно ознакомиться с его описанием на главной странице. На веб-портале в разделе новостей также имеются формы по добавлению/редактированию новостей. Формы выполнены с использованием элемента form. Здесь помимо ввода информации происходит проверка корректности введенных данных и, если данные введены неправильно или не заполнены вообще – появляется информация об ошибке (рисунок 18).

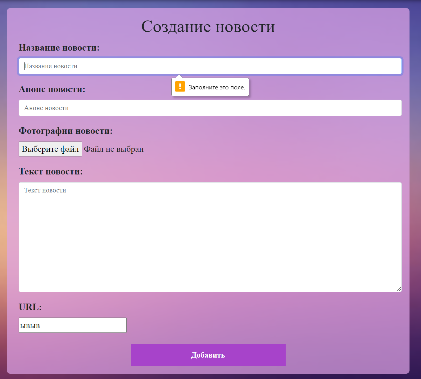


Рисунок 18 – Сообщение на форме создания новости

На странице форума имеется список вопросов, с формой поиска и пагинацией. Пользовательский интерфейс здесь был написан таким же образом, как и на главной странице. Страница форума показана на рисунке 19.

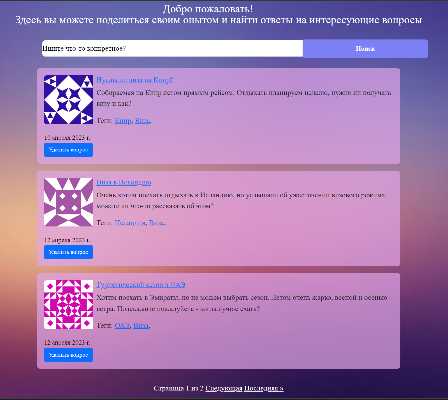


Рисунок 19 – Страница форума

На странице рекомендаций пользователь вводит название места, к которому желает подобрать рекомендации. Если рекомендации не найдены (такого места нет в датасете), то пользователь получит об этом сообщение (рисунок 20).

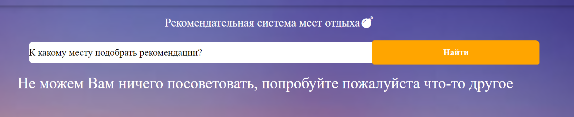


Рисунок 20 – Сообщение на странице рекомендаций

Если рекомендации все же были найдены, то пользователь увидит их списком. Здесь, как и на главной странице блок каждого рекомендованного места сделан с помощью таких элементов HTML как div, span, img. Button и др. По каждому месту пользователь получает 4 рекомендации (рисунок 21).

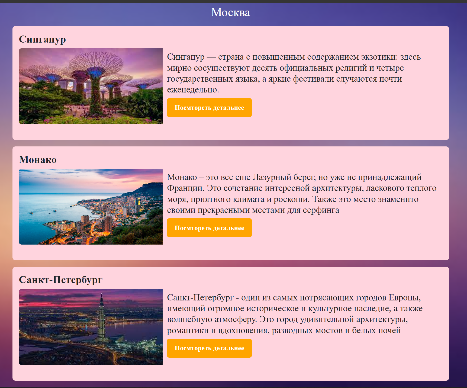


Рисунок 21 – Результат работы рекомендательной системы

3.9 Граф диалога

В процессе разработки системы был составлен граф диалога. Он описывает переходы на страницы при работе с интернет-порталом, упрощает понимание функционирования системы, помогает выявить тупиковые ситуации в работе портала, оптимизировать переход от одного состояния в другое. Граф диалога представлен в приложении А.

ЗАКЛЮЧЕНИЕ

В ходе выполнения Выпускной квалификационной работы бакалавра были достигнуты следующие результаты:

* Была изучена предметная область, связанная с рекомендательной системой и обработкой текста.
* Рассмотрены виды рекомендательных систем, подготовки данных к content-based рекомендательной системе, библиотеки для лемматизации и стемминга текста.
* Проведен анализ аналогов и разработанной системы, библиотек для лемматизации и стемминга PyMorphy и SpaCy.
* Проведен сбор и обработка данных для рекомендательной системы.
* Разработана рекомендательная система, основанная на контенте.

Разработанная система представляет собой многофункциональную систему, которую можно интегрировать в онлайн-магазин туров. Это позволит привлечь пользователей, предоставив на одной платформе всю необходимую информацию.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

* + - 1. Постников В. М. Основы эксплуатации автоматизированных систем обработки информации и управления. Краткий̆ курс: учеб. пособие. — 177 с.
      2. Kim Falk. Practical Recommender Systems / Kim Falk – Manning Publications, 2019. – 432 c.
      3. Документация библиотеки pymorphy2 [Электронный ресурс] // pymorphy2.readthedocs.io URL: <https://pymorphy2.readthedocs.io/> (дата обращения: 11.05.2023).
      4. Документация библиотеки SpaCy. [Электронный ресурс] // spacy.io URL: https://spacy.io/ (дата обращения: 12.05.2023).
      5. Решение NLP задач при помощи spacy [Электронный ресурс] // habr.com URL: https://habr.com/ru/post/531940/ (дата обращения: 12.05.2023).
      6. Документация Python [Электронный ресурс] // www.python.org URL: https://www.python.org/doc/ (дата обращения: 22.02.2023).
      7. Документация Jupyter [Электронный ресурс] // docs.jupyter.org URL: https://docs.jupyter.org/en/latest/ (дата обращения: 22.02.2023).
      8. PyCharm IDE [Электронный ресурс] // www.jetbrains.com URL: https://www.jetbrains.com/ru-ru/pycharm/ (дата обращения: 26.04.2023).
      9. Документация NumPy [Электронный ресурс] // numpy.org URL: https://numpy.org/ (дата обращения: 26.04.2023).
      10. Документация Pandas [Электронный ресурс] // pandas.pydata.org URL: https://pandas.pydata.org/docs/ (дата обращения: 27.04.2023).
      11. Matplotlib [Электронный ресурс] // matplotlib.org URL: https://matplotlib.org/ (дата обращения: 27.04.2023).
      12. Пакет библиотек NLTK [Электронный ресурс] // www.nltk.org URL: https://www.nltk.org/ (дата обращения: 01.05.2023).
      13. Библиотека Scikit-learn [Электронный ресурс] // scikit-learn.org URL: https://scikit-learn.org/stable/user\_guide.html (дата обращения; 08.05.2023).
      14. Документация модуля re [Электронный ресурс] // docs.python.org URL: https://docs.python.org/3/library/re.html (дата обращения: 25.02.2023).

ПРИЛОЖЕНИЕ A ГРАФИЧЕСКАЯ ЧАСТЬ

В графическую часть выпускной квалификационной работы входят:

А.1 Интернет-портал по туризму с рекомендательной системой в нотации IDEF0.

А.2 Подробное описание функциональности системы в нотации IDEF0.

А.3 Граф пользовательского диалога.

А.4 Таблица 19 – Сравнительный анализ работы рекомендательных систем.

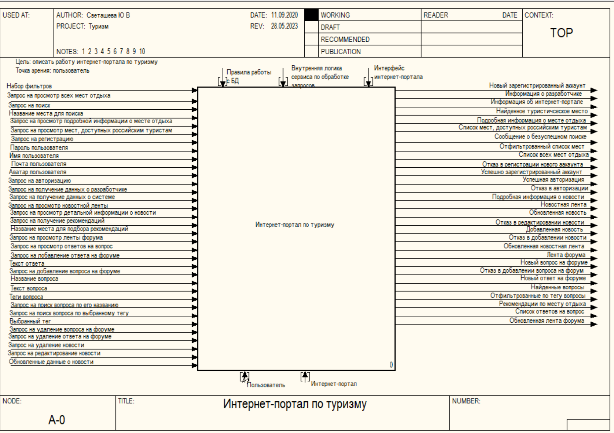
А.5 Таблица 20 – Описание атрибутов сущностей

А.6 Экранная форма «Раздел рекомендаций».

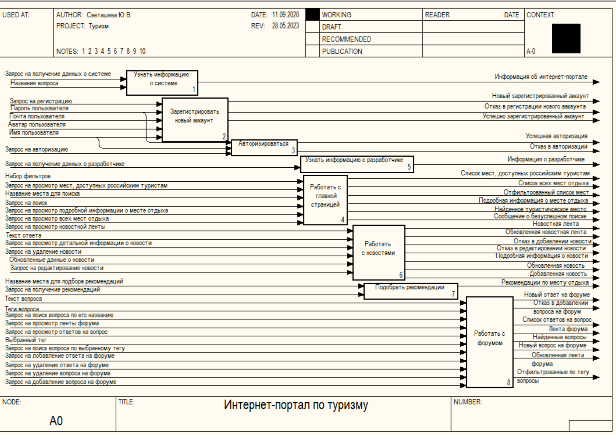
А.7 Графическая модель предметной области

А.8 Графическая модель функций системы

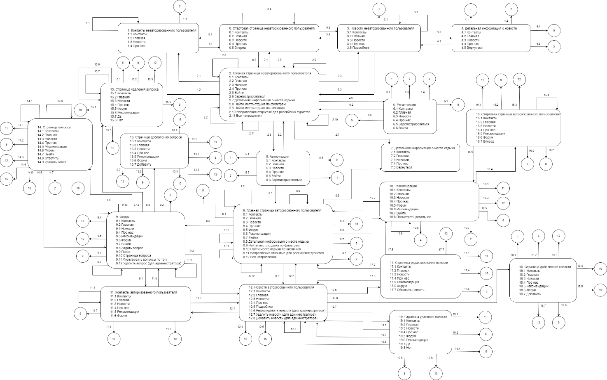
А.1 Интернет-портал по туризму с рекомендательной системой в нотации IDEF0



А.2 Подробное описание функциональности системы в нотации IDEF0



А.3 Граф пользовательского диалога



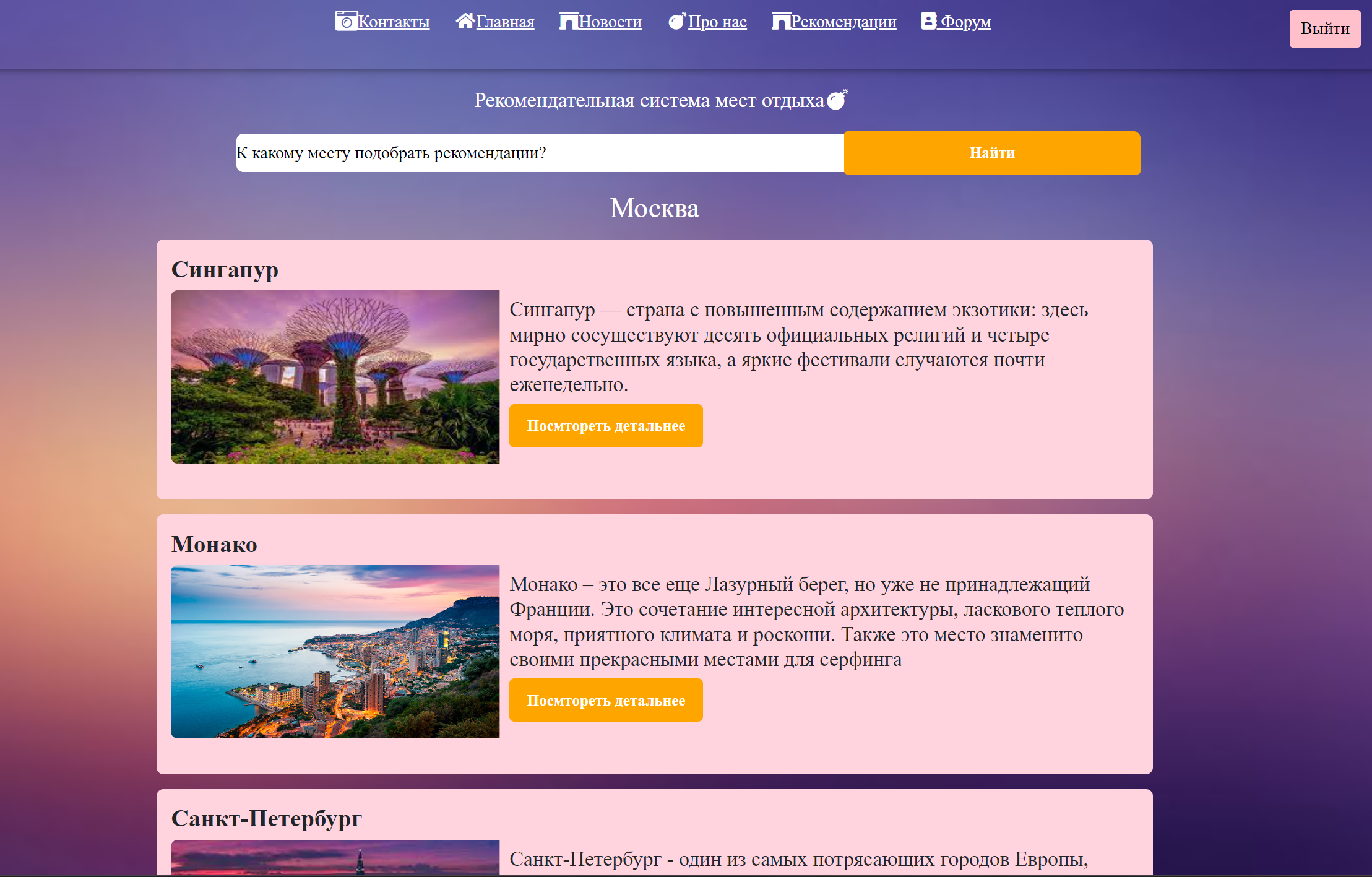
А.4 Таблица 19 – Сравнительный анализ работы рекомендательных систем

| **Место** | **Результаты без стемминга** | **Результаты со стемингом** | **Вывод** |
| --- | --- | --- | --- |
| Египет | Таиланд | Таиланд | Оба места подходят. |
| Турция | Турция | Оба места подходят. |
| Калининград | Монако | Со стеммингом рекомендации лучше, так как Монако теплая страна, имеющая выход к морю.  Калининград – северный город. |
| Фиджи | Кипр | Со стеммингом рекомендации лучше, так Кипр ближе по ценовой категории к Египту, чем Фиджи (в датасете есть эта информация). |
| Новая Зеландия | Грузия | Со стеммингом рекомендации лучше, так Грузия ближе по ценовой категории к Египту, чем Новая Зеландия (в датасете есть эта информация). |
| Бразилия | Аргентина | Аргентина | Оба места подходят. |
| Новая Зеландия | Сальвадор | Оба места подходят. |
| Франция | Перу | Со стеммингом рекомендации .лучше, так как Перу – соседняя для Бразилии страна. По культурным и природным особенностям она ближе к Бразилии. |
| Перу | Парагвай | Без стемминга рекомендации лучше. Перу – соседняя для Бразилии страна. По культурным и природным особенностям она ближе к Бразилии. |
| Тунис | Новая Зеландия | Оба места подходят |
| Северная Осетия | Адыгея | Алтай | Оба места подходят. |
| Алтай | Адыгея | Оба места подходят. |
| Ингушетия | Байкал | Оба места подходят. |
| Таиланд | Ингушетия | Со стеммингом рекомендации лучше. Так как Ингешетия, так же, как и Северная Осетия – регион Кавказа. По ценовой категории, культурным и национальным особенностям он подходит лучше, чем Таиланд. |
| Марокко | Таиланд | Оба подгодят, так как и в Марокко и в Таиланд есть горные массивы. |
| Коста-Рика | Аргентина | Сальвадор | Со стеммингом рекомендации лучше. Так как Сальвадор, так же, как и Коста-Рика – небольшое государство Панамского перешейка. По культурным и национальным особенностям он подходит больше, чем Аргентина. |
| Новая Зеландия | Новая Зеландия | Оба подходят |
| Сальвадор | Аргентина | Без стемминга рекомендации лучше. Так как Сальвадор, так же, как и Коста-Рика – небольшое государство Панамского перешейка. По культурным и национальным особенностям он подходит больше, чем Аргентина. |
| Алтай | Исландия | Со стеммингом рекомендации лучше. Так как Исландия по ценовой категории и местному колориту ближе к Коста-Рике, чем Алтай. |
| Бразилия | Бразилия | Оба подходят. |

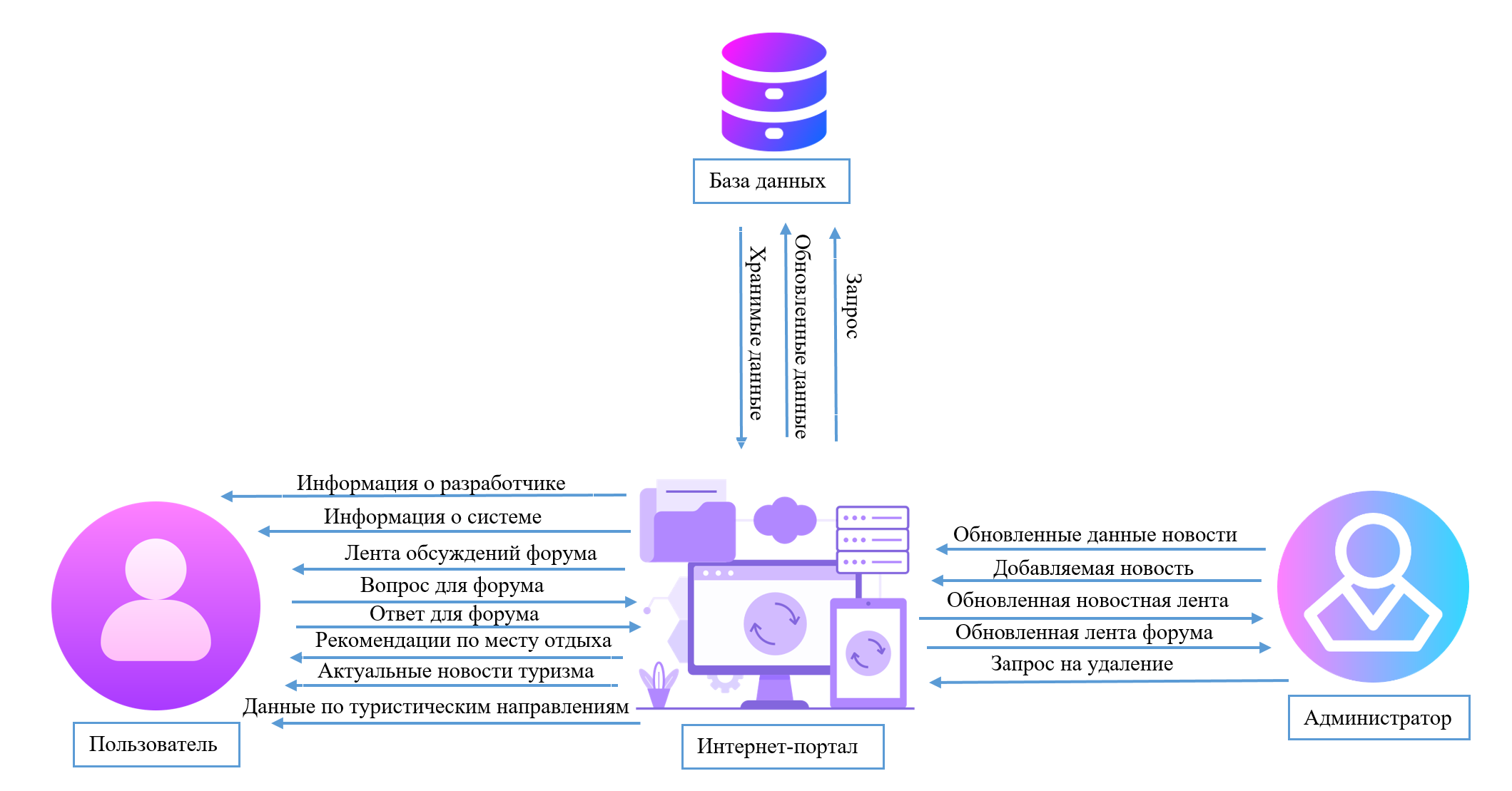
А.5 Таблица 20 – Описание атрибутов сущностей

| **Сущность** | **Атрибут** | **Тип данных** | **Краткое описание** |
| --- | --- | --- | --- |
| Сезон | Id | Int | ID сезона |
| season\_name | CharField(80) | Название сезона |
| Туристическое место | Id | Int | ID места |
| place\_name | CharField(80) | Название места |
| place\_photos | ImageField | Фотография места |
| place\_description | CharField(450) | Краткое описание места |
| place\_full\_text | CharField(4000) | Полный тест |
| location\_id | ForeignKey  ('Location') | Часть света |
| season\_id | ForeignKey('Season') | Сезон, в который лучше всего посетить локацию |
| viza\_id | ForeignKey('Viza') | Виза |
| open\_for\_russians | BooleanField | Поле, показывающее открыто ли место для российский туристов |
| Часть света | Id | Int | ID тега |
| location\_name | CharField(80) | Название тега |
| Виза | Id | Int | ID записи |
| viza\_type | CharField(80) | ID добавленного рецепта |
| Новость | Id | Int | ID записи |
| new\_title | CharField(180) | Название новости |
| new\_annons | CharField(250) | Кратное описание новости |
| new\_full\_text | CharField(1400) | Полный текст новости |
| new\_photos | ImageField | Картинка новости |
| new\_date | DateTimeField | Дата создания/обновления новости |
| slug | SlugField | Уникальный символьный идентификатор новости |
| Пользователь | id | Int | ID добавившего пользователя |
| username | CharField(150) | Ник пользователя |
| first\_name | CharField(150) | Имя пользователя |
| last\_name | CharField(150) | Фамилия пользователя |
| email | EmailField | E-mail пользователя |
| password | CharField | Пароль пользователя |
| is\_superuser | BooleanField | Определяет, обладает ли пользователь правами супер-пользователя |
| Профиль | Id | INT | ID оценки |
| user | OneToOneField(User) | Ссылка на пользователя |
| avatar | ImageField | Аватар пользователя |
| Вопрос | Id | Int | ID записи |
| title | CharField(150) | Название вопроса |
| author | ForeignKey(‘Profile’) | Автор вопроса |
| text | CharField(150) | Текст вопроса |
| tags | CharField(150) | Тэги вопроса |
| date | DateField | Дата публикации вопроса |
| tag\_id | ForeignKey(‘Tag’) | Таги, относящиеся к вопросу |
| Ответ | Id | INT | ID рекомендаций |
| text | CharField(2000) | Текст ответа |
| date | DateField | Дата ответа |
| author | ForeignKey(‘Profile’) | Автор ответа |
| question | ForeignKey(‘Question) | Вопрос, к которому относится ответ |
| Тэг | Id | INT | ID рекомендаций |
| name | CharField(50) | Имя тэга |

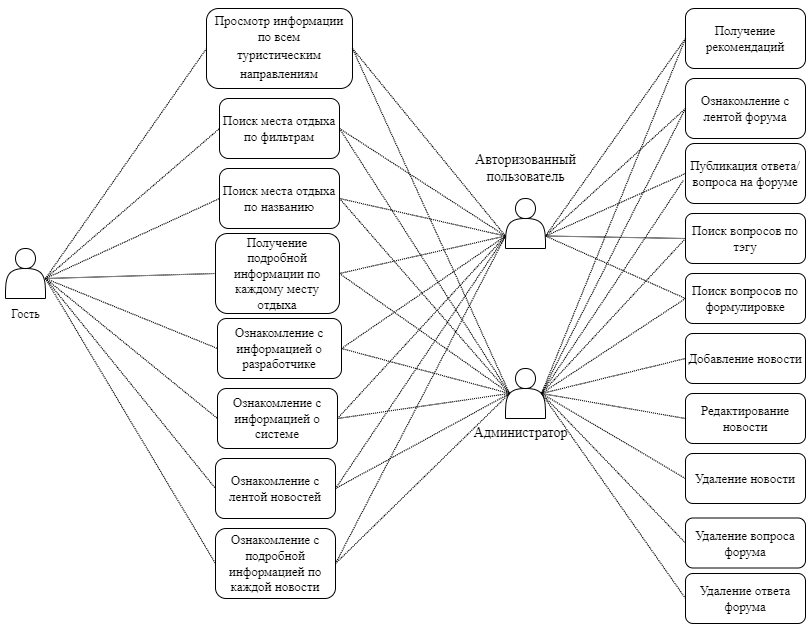
А.6 Экранная форма «Раздел рекомендаций»



А.7 Графическая модель предметной области.



А.8 Графическая модель функций системы



ПРИЛОЖЕНИЕ B ТЕХНИЧЕСКОЕ ЗАДАНИЕ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |  |  |
| --- | --- | --- | --- |
|  | Утверждаю  Заведующий кафедрой ИУ-5 |  | Согласовано  научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_В.И.Терехов  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |  | \_\_\_\_\_\_\_\_\_\_\_\_\_А.Н.Нардид  "\_\_"\_\_\_\_\_\_\_\_\_\_2024 г. |

**Сегментация объектов на спутниковых снимках**

Техническое задание

(вид документа)

писчая бумага

(вид носителя)

5

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Шимолина Полина Кирилловна |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2024 г. |  |

Москва - 2024

1. Введение

Сервис сегментации объектов на спутниковых снимках. Предназначен для автоматического выделения различных объектов на изображениях.

1. Основания для разработки

Основанием для разработки является задание на выпускную квалификационную работу, подписанное руководителем выпускной работы и утверждённое заведующим кафедрой ИУ5 МГТУ им. Н.Э. Баумана 15 декабря 2023 года.

1. Назначение разработки

Назначением работы является создание сервиса, позволяющего автоматизировать выделение различных объектов на изображениях с целью улучшения анализа и интерпретации данных, полученных с помощью спутниковых снимков.

1. Требования к программе или программному изделию
   1. Требования к функциональным характеристикам:

Программа должна выполнять следующие функции:

* + 1. Получать изображение от пользователя.
    2. Возвращать сегментированное изображение пользователю.
    3. Поддерживать различные типы объектов – программа должна быть способна выделять разнообразные объекты на изображениях.
  1. Требования к надёжности:

Приложение должно надёжно и устойчиво функционировать. При сбоях восстанавливаться после перезагрузки.

* 1. Требования к составу и параметрам технических средств:

Минимальные системные требования для работы веб-сервиса:

* + 1. Подключение к Wi-Fi
    2. Установленный веб-браузер
    3. Компьютерная мышь
  1. Требования к информационной и программной совместимости
     1. Требования к исходным кодам и языкам программирования
        1. Python 3.11 для написания модели
        2. Python Django для backend
        3. JavaScript для frontend
        4. SQLite для хранения данных
        5. Библиотеки для создания модели и обработки данных в Python

1. Требования к программной документации

Для представления заказчику разрабатываются следующие документы:

* 1. Техническое задание.
  2. Программа и методика испытаний.
  3. Руководство пользователя – c описанием всех действий, которые может произвести пользователь, и реакции системы на эти действия.
  4. Рабочий материал по выполняемому проекту.
  5. Графический материал по проекту в формате презентации.

1. Технико-экономические показатели

Требования к данному разделу не предъявляются.

1. Стадии и этапы разработки
   1. Этапы разработки

График выполнения отдельных этапов работ приведён в таблице 1 соответствии с приказом об организации учебного процесса в 2023/20234 учебном году.

Таблица 1 – Этапы разработки

| **№ п/п** | **Наименование этапа и содержание работ** | **Сроки исполнения** |
| --- | --- | --- |
| 1 | Постановка задач разработки, анализ инструментов и программного обеспечения для создания приложения | Ноябрь-декабрь 2023 г. |
| 2 | Формулирование проблемы, цели и задач работы | Январь 2024 г. |
| 3 | Разработка модели для сегментации и веб-сервиса | Февраль-март 2024 г. |
| 4 | Тестирование и отладка модели и сервиса | Март-апрель 2024 г. |
| 5 | Оформление документации | Апрель 2024 г. |
| 6 | Демонстрация научному руководителю итоговой работы | Май 2024 г. |

* 1. Перечень разрабатываемых программных документов.

График разработки отдельных программных документов приведён в таблице 2 соответствии с утвержденным календарным планом для выпускной квалификационной работы.

Таблица 2 – График разработки отдельных программных документов

| **№ п/п** | **Наименование этапов разработки отдельных программных документов** | **Сроки выполнения этапов** | **Принимающее лицо** | |
| --- | --- | --- | --- | --- |
| **Должность** | **подпись, ФИО** |
|  | Разработка технического задания | 10.03.2024 | Руководитель ВКР | А.Н.Нардид |
|  | Разработка программы и методики испытания | 01.04.2024 | Руководитель ВКР | А.Н.Нардид |
|  | Подготовка доклада и презентации | 25.05.2024 | Руководитель ВКР | А.Н.Нардид |
|  | Получение заключения научного руководителя | 28.05.2024 | Руководитель ВКР | А.Н.Нардид |

1. Порядок контроля и приёмки

Приём программного изделия в виде испытаний осуществляется в ходе «Защиты макетов программ – предварительной защиты ВКРБ» в период с 15 по 24 мая 2024 года в соответствие с разработанной программой и методикой испытаний.

ПРИЛОЖЕНИЕ C ПРОГРАММА И МЕТОДИКА ИСПЫТАНИЙ

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

им. Н.Э. Баумана

Кафедра «Системы обработки информации и управления»

|  |  |
| --- | --- |
|  | Утверждаю  Научный руководитель |
|  | \_\_\_\_\_\_\_\_\_\_\_ Михеев В.А.  "\_\_"\_\_\_\_\_\_\_\_\_\_2023 г. |

**Разработка портала и рекомендательной системы по туризму**

Программа и методика испытаний

(вид документа)

писчая бумага

(вид носителя)

8

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Светашева Юлия Васильевна |
| "\_\_"\_\_\_\_\_\_\_\_\_\_\_2023 г. |  |

Москва 2023 г.

# Объект испытаний

Объектом испытаний является веб-приложение с использованием рекомендательной системы по подбору мест отдыха.

# Цель испытаний

Испытания проводятся с целью проверки соответствия клиента требованиям функциональных характеристик, описанным в п. 5.2 Технического задания.

# Состав предъявляемой документации

* 1. Техническое задание
  2. Программа и методика испытаний

# Технические требования

# 4.1 Требования к составу аппаратного обеспечения

Минимальные требования для работы клиента:

1. Персональный компьютер с ОС Windows
2. Подключение к интернету

# 4.2 Требования к составу программного обеспечения

Для доступа к интернет-порталу необходимо наличие браузера.

# Порядок проведения испытаний

Испытания системы будут проводиться в следующем порядке:

1. Взаимодействие с интерфейсом веб-приложения
2. Получение рекомендаций по месту отдыха.

Приемочные испытания включают проверку:

- полноты реализации функций, указанных в Техническом задании

- полноты и достаточности действий, доступных пользователю, для нормального функционирования системы

- степени сложности взаимодействия системы с пользователем

# Методы испытаний

Шаги для ПМИ представлены в таблице 22

Таблица 22 - шаги ПМИ.

| **№** | **Наименование / № пункта в ТЗ** | **Выполняемые действия** | **Результат** |
| --- | --- | --- | --- |
| 1 | 5.2.1 Регистрация пользователя | Кликнуть на кнопку «Зарегистрироваться» навигационной панели. Заполнить форму регистрации. Нажать на кнопку «Зарегистрироваться». | Переход на главную страницу авторизованного пользователя. |
| 2 | 5.2.2 Авторизация пользователя | Кликнуть на кнопку «Войти» навигационной панели. Заполнить форму авторизации. Нажать на кнопку «Войти». | Переход на главную страницу авторизованного пользователя. |
| 3 | 5.2.3 Просмотр информации по всем туристическим направлениям | Нажать на кнопку «Главная» на навигационной панели. | Переход на главную страницу web-приложения, содержащую информацию о всех туристических направлениях. |
| 4 | 5.2.4 Поиск места отдыха по фильтрам | Выбрать необходимые фильтры на боковой панели и нажать на кнопку «Найти». | Вывод мест, соответствующих указанным фильтрам. |
| 5 | 5.2.5 Поиск места отдыха по названию. | Ввести название места в строку поиска и нажать на кнопку «Поиск». | Вывод места, соответствующего запросу пользователя. |
| 6 | 5.2.6 Поиск мест отдыха доступных российским туристам. | На главной странице нажать на кнопку «Направления, доступные для российских туристов». | Вывод мест, доступных для отдыха российских туристов. |
| 7 | 5.2.7 Получение подробной информации по каждому месту отдыха. | На главной странице нажать на кнопку «Посмотреть детальнее». | Переход на страницу подробной информации о туристическом направлении. |
| 8 | 5.2.8 Ознакомление с информацией о разработчике. | Нажать на кнопку «Контакты» на навигационной панели. | Переход на страницу с информацией о разработчике. |
| 9 | 5.2.9 Ознакомление с лентой новостей. | Нажать на кнопку «Новости» на навигационной панели. | Переход на страницу новостей. |
| 10 | 5.2.10 Ознакомление с подробной информацией о новости. | На странице новостей нажать на кнопку «Подробнее». | Переход на страницу с подробной информацией о новости. |
| 11 | 5.2.11 Добавление новости. | Авторизоваться как администратор. Нажать на кнопку «Новости» на навигационной панели. Нажать на кнопку «Добавить новость» на странице новостей. В форме создания новости заполнить все поля и нажать на кнопку «Добавить». | Переход на страницу новостей, где отобразится добавленная новость. |
| 12 | 5.2.12 Удаление новости. | Авторизоваться как администратор. Нажать на кнопку «Новости» на навигационной панели. Нажать на кнопку «Удалить новость» на странице новостей. Подтвердить удаление новости. | Переход на страницу новостей, где не будет удаленной новости. |
| 13 | 5.2.13 Редактирование новости. | Авторизоваться как администратор. Нажать на кнопку «Новости» на навигационной панели. Нажать на кнопку «Редактировать новость» на странице новостей. В форме редактирования новости изменить нужные поля и нажать на кнопку «Обновить новость». | Переход на страницу новостей, где будут отображены внесенные изменения. |
| 14 | 5.2.14 Ознакомление с информацией о системе. | Нажать на кнопку «Про нас» на навигационной панели. | Отображение информации о разработанной системе. |
| 15 | 5.2.15 Ознакомление с лентой форума. | Авторизоваться в системе. Нажать на кнопку «Форум» на навигационной панели. | Переход на страницу форума. |
| 16 | 5.2.16 Поиск вопроса по его формулировке. | Ввести формулировку вопроса в строку поиска и нажать на кнопку «Поиск». | Вывод найденного вопроса или информации об отсутствии результатов. |
| 17 | 5.2.17 Поиск вопросов по тегу. | На странице форума нажать на тег вопроса. | Переход на страницу с вопросами, содержащими выбранный тег. |
| 18 | 5.2.18 Просмотр конкретного вопроса и ответов на него. | На странице форума нажать на название вопроса. | Переход на страницу выбранного вопроса и ответов на него. |
| 19 | 5.2.19 Создание нового вопроса на форуме. | На странице форума нажать на кнопку «Задать вопрос», корректно заполнить поля формы «Задать новый вопрос» и нажать на кнопку «Спросить!». | Переход на страницу форма, где будет отображен добавленный вопрос. |
| 20 | 5.2.20 Написание ответа на существующий вопрос форума. | На странице форума нажать на название вопроса, на которые нужно написать ответ. На странице вопроса вписать свой ответ в поле ввода и нажать на кнопку «Ответить». | Отображение добавленного ответа на странице вопроса. |
| 21 | 5.2.21 Удаление вопроса на странице форума. | Войти на страницу форума с правами администратора. Нажать на кнопку «Удалить вопрос». Подтвердить удаление. | Переход на страницу форума, где не будет удаленного вопроса. |
| 22 | 5.2.22 Удаление ответа на странице форума. | Войти на страницу форума с правами администратора и нажать на название интересующего вопроса. Нажать на кнопку «Удалить ответ». | Переход на страницу вопроса, где не будет удаленного ответа. |
| 23 | 5.2.23 Получение рекомендации по конкретному туристическому направлению. | Нажать на кнопку «Рекомендации» на навигационной панели. В строке ввода на странице рекомендательной системы ввести название места, на которое хотим получить рекомендации. Нажать на кнопку «Найти». | Отображение мест, рекомендованных к данному. |