Содержание

[1 Исследовательский раздел 7](#_Toc73913750)

[1.1 Общая характеристика системы 7](#_Toc73913751)

[1.2 Обзор алгоритмов, используемых для создания рекомендательных систем 10](#_Toc73913757)

[1.3 Постановка задачи на разработку ИС 14](#_Toc73913763)

[1.4 Выводы по исследовательскому разделу 17](#_Toc73913769)

[2 Аналитический раздел 19](#_Toc73913770)

[2.1 Разработка архитектуры ИС 19](#_Toc73913771)

[2.2 Разработка структуры базы данных 27](#_Toc73913775)

[2.3 Математическое обеспечение ИС 31](#_Toc73913782)

[2.4 Техническое обеспечение ИС 35](#_Toc73913786)

[2.5 Выводы по исследовательскому разделу 35](#_Toc73913787)

[3 Экономический раздел 36](#_Toc73913788)

[3.1 Организация и планирование работ по теме 36](#_Toc73913789)

[3.2 Выводы по экономическому разделу 43](#_Toc73913793)

[4 Технологический раздел 44](#_Toc73913794)

[4.1 Обоснование выбора средств разработки 44](#_Toc73913795)

[4.2 Описание разработанного программного продукта 45](#_Toc73913799)

[4.3 Демонстрация работы программного продукта 49](#_Toc73913803)

[4.4 Выводы по технологическому разделу 53](#_Toc73913804)

[Список источников 55](#_Toc73913805)

ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

|  |  |  |
| --- | --- | --- |
| ИС | - | Информационная система |
| БД | - | База данных |
| СУБД | - | Система управления БД |
| IDEF0 | - | Методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов |
| ID | - | Идентификатор, уникальный номер объекта в системе |

Введение

На сегодняшний день человек живет в постиндустриальном обществе. Одним из признаков этого общества является развитие сферы услуг, в которой занято больше половины трудоспособного населения России. Это важная часть экономики России и других развитых стран.

Каждый человек в своей жизни хоть раз пользовался услугами. Так как люди имеют разные предпочтения, сфера услуг предлагает большое количество вариантов по предоставлению услуг. Из-за большой вариативности, человеку не всегда удается подобрать то, что он действительно хотел бы. В таких ситуациях на помощь приходят рекомендательные системы, которые служат фильтром среди большого количества информации.

Рекомендательная система – это совокупность алгоритмов и программ, цель которой предсказать объект, который будет интересен пользователю. Рекомендательные системы делятся по предмету рекомендации, цели рекомендации, контексту рекомендации, источнику рекомендации, степени персонализации, формату рекомендации и прозрачности рекомендации. Основной составляющей таких систем является матрица предпочтений, в которой одна из осей содержит пользователей, а вторая – объект рекомендации. Пересечение пользователя и объекта заполняется значением, которое отвечает за заинтересованность пользователя в объекте. Пользователь оценивает небольшую часть объектов, а далее рекомендательная система должна предсказать отношение пользователя к объекту, которому пользователь не дал оценки.

Наличие хорошо проработанной рекомендательной системы может являться ключевым фактором для выбора сервиса по предоставлению услуг. Одним из наглядных примеров использования рекомендательных систем являются сервисы для прослушивания музыки. В России летом 2020 года официально запустился сервис «Spotify». Этот сервис стал успешным благодаря наличию рекомендательной системы, которая с большой точностью предсказывает музыку, которая понравится пользователю.

Люди незаметно для себя ежедневно сталкиваются с рекомендательными системами. Например, рекламные баннеры на сайтах. Они предлагают товары или услуги, которые с большой вероятностью могут заинтересовать пользователя. Такая рекомендательная подборка создается на основе данных, которые пользователь оставляет, когда производит любое действие на своем вычислительном устройстве.

Рекомендательные системы имеют множество способов применения, однако после изучения сфер применений, не было обнаружено использование рекомендательных систем для поиска мероприятий в Москве. В рамках данной работы будут рассмотрены методы создания рекомендательных систем и их сравнительный анализ. Также в работе будет произведена разработка компонентов информационной системы для поиска мероприятий.

Объект и предмет исследования

Объектом исследования является информационно-аналитическая система поиска мероприятий.

Предметом исследования является создание рекомендательной системы и разработка структуры базы данных информационно-аналитической системы.

Цели и задачи

Целью выпускной квалификационной работы является автоматизация выбора места проведения досуга.

Задачи выпускной квалификационной работы являются изучение и анализ логик и методов работы рекомендательных систем, систематизация и адаптация алгоритмов для переноса их в программную среду.

Гипотеза выпускной квалификационной работы

Создание работающего прототипа программных модулей позволит облегчить выбор места проведения досуга и повысит количество участников мероприятий.

Методы, используемые при написании работы

В данной работе будет использоваться инкрементный метод разработки программного продукта. Этот метод позволит выпустить рабочий прототип на ранних этапах и контролировать состояние продукта в реальном времени.

Научная новизна и практическая значимость исследуемой проблемы

Количество мероприятий растет с каждым днем и посетить все интересующие человека мероприятия физически невозможно. Предлагаемое решение избавит пользователя от траты большого количества времени на поиск интересующего мероприятия и выделит только те мероприятия, которые с наибольшей вероятностью понравятся пользователю.

1. Исследовательский раздел

Для достижения цели данной работы – автоматизации выбора места проведения досуга, необходимо решить следующие задачи:

* дать общую характеристику системы и выявить требования к ней;
* дать общий обзор распространенным алгоритмам рекомендательных систем и определить оптимальное решение для разрабатываемого прототипа;
* разработать архитектуру и структуру БД разрабатываемого продукта;
* определить математическое и техническое обеспечение ИС;
* произвести оценку экономических параметров разработки ИС;
* разработать функционирующий прототип ИС поиска мероприятий.

В результате решения данных задач будет получен работающий прототип ИС поиска мероприятий.

В данном разделе будет описана общая характеристика системы, обзор и анализ распространенных алгоритмов работы рекомендательных систем и описание требований к информационной системе.

* 1. Общая характеристика системы
     1. Полное название системы

Полное наименование системы: Информационная система поиска мероприятий.

* + 1. Плановые сроки начала и окончания работы по созданию системы

Начало – 01 февраля 2021 года.

Окончание – 31 мая 2021 года.

* + 1. Назначение системы

Информационная система поиска мероприятий предназначена для обеспечения учета рейтинга мероприятий и мест, в которых они проводятся, а также для создания уникальной подборки мероприятий на основе предпочтений пользователя.

* + 1. Цели создания системы

Главной целью создания ИС является оптимизация процесса поиска мероприятий для пользователя на основе прошлых оценок, которые пользователь поставил посещенным мероприятиям. Также одной из целей будет являться просмотр статистических данных о мероприятии, месте, где оно проводится, и других пользователях системы. Для достижения поставленных целей система должна иметь такой функционал как:

* возможность регистрации пользователя на мероприятия с целью дальнейшего посещения;
* процесс подписок пользователей друг на друга;
* анализ данных о мероприятиях и местах;
* подбор мероприятий для пользователя, на основе его предпочтений.
  + 1. Анализ бизнес-процессов

В ходе анализа бизнес-процессов было проведено моделирование этих процессов в области «Выбор места проведения досуга» и были составлены диаграммы в BPMN-нотации IDEF0. Диаграмма верхнего уровня представлена на рисунке 1.

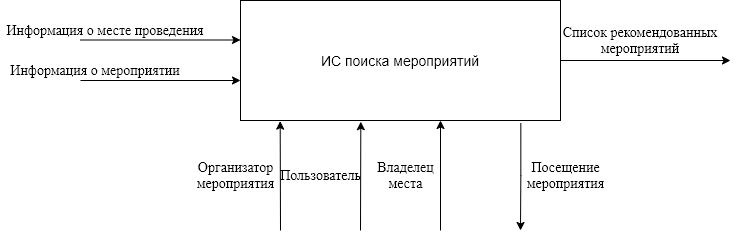


Рисунок 1 - Диаграмма IDEF0 верхнего уровня

На диаграмме в качестве входных данных можно увидеть: информацию о месте проведения и информацию о мероприятии. Ресурсами будут являться: пользователь, организатор мероприятия, владелец места. Посещение мероприятия выполняется за пределами рассматриваемого блока. Результатом работы этого процесса будет являться список мероприятий, которые с наибольшей вероятностью понравятся пользователю.

Для более подробного описания процесса была произведена декомпозиция диаграммы верхнего уровня, которая изображена на рисунке 2.

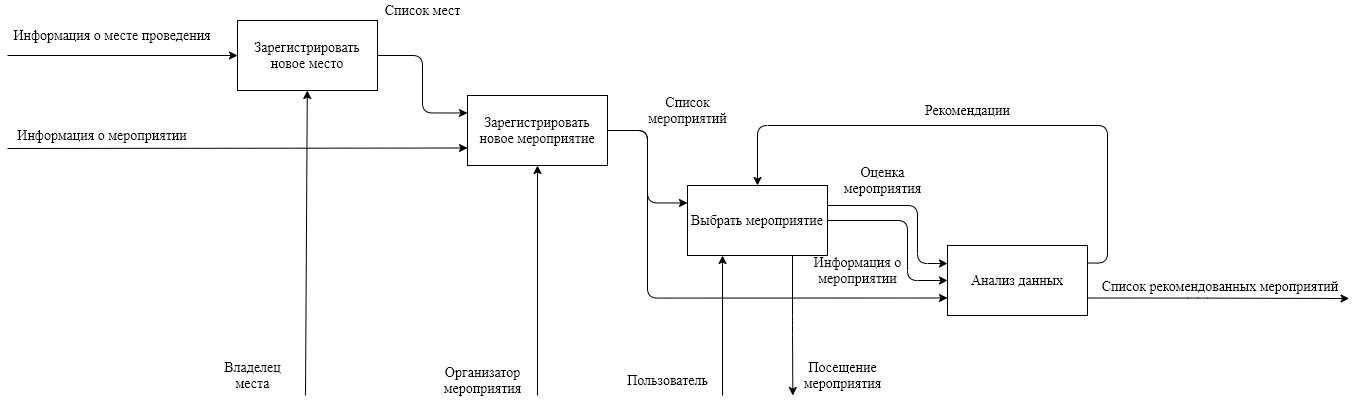


Рисунок 2 - Диаграмма декомпозиции

На данной диаграмме можно рассмотреть процесс выбора мероприятий более детально. Информация о месте проведения подается в блок регистрации нового места, где владелец места регистрирует свое пространство в системе. Информация о мероприятии попадает в блок регистрации нового мероприятия. Также, как и с новым местом, организатор мероприятия регистрирует свое событие. Одновременно с информацией о мероприятии в этот блок попадает список мест, из которого организатор должен выбрать одно для проведения мероприятия. После этого все зарегистрированные мероприятия списком отправляются в блок выбора мероприятия, где пользователь выбирает событие из списка, посещает его и оценивает. Данный блок регламентируется рекомендациями, составленными в блоке анализа данных. В блок анализа данных приходят информация о выбранном мероприятии и оценка мероприятия от пользователя из блока выбора мероприятия и список мероприятий из блока регистрации мероприятия. В блоке анализа данных происходит расчет коэффициента всех событий относительно одного пользователя на основе данных из блока выбора мероприятий. Все данные в обработанном виде отправляются в базу данных, из которой можно будет получить список рекомендованных мероприятий.

* 1. Обзор алгоритмов, используемых для создания рекомендательных систем
     1. Контент-ориентированный анализ

При использовании контент-ориентированного анализа учитываются данные о вкусах пользователей, которые они указали при регистрации в системе и информация об объекте рекомендации. Данный подход используется в основном для решения проблемы невозможности порекомендовать что-нибудь новому пользователю, так как в системе еще нет данных о его предпочтениях.

* + 1. Коллаборативная фильтрация

Основной идеей коллаборативной фильтрации является прогноз на основе предпочтений других пользователей. То есть анализируются данные об оценке продукта самим пользователем и другими. Система кластеризирует пользователей с учетом схожих оценок по одним и тем же товарам. После разделения система рекомендует конкретному пользователю тот контент, который он еще не оценил, но этот контент уже понравился другим пользователям из этого сегмента. Этот вид фильтрации называется «user-based», то есть основанный на соседстве. Такой подход имеет ряд проблем:

* для новых пользователей, которые делали мало оценок не найдется сегмент с похожими на него пользователями;
* если в сегменте никто не оценил объект, то система не сможет предсказать, понравится ли он этому кластеру;
* новые объекты не будут никому рекомендоваться, так как их еще никто не оценил.

Вторым типом коллаборативной фильтрации является «item-based», то есть основанный на модели. С помощью интеллектуального анализа данных находятся закономерности среди оценок и объектов. Данный подход выдает более точные прогнозы за счет выявления скрытых факторов, объясняющих оценки пользователей. В этом типе также наблюдается проблема холодного старта, то есть система не знает, кому рекомендовать новый объект.

Коллаборативная фильтрация также имеет гибридный подход, который объединяет в себе подходы основанный на соседстве и основанный на модели. Данный подход помогает преодолеть ограничения «user-based» метода и улучшает точность рекомендаций[[1]](#footnote-1). Гибридный подход дорог и сложен в реализации.

* + 1. Фильтрация, основанная на знаниях

Гомзин и Коршунов[[2]](#footnote-2) характеризуют принцип работы методов, основанных на знаниях, так: пользователи указывают свои требования к товару, а система пытается найти нужный товар. Такие методы, обычно делятся на две группы:

* выбор близких объектов;
* использование жестких ограничений.

При выборе близких объектов ищутся объекты, которые обладают характеристиками близкими к требованиям пользователя. В другом случае в качестве рекомендованных объектов выступают те товары, которые полностью соответствуют всем требованиям пользователя.

При использовании выбора близких объектов учитываются как свойства товара, так и пользовательские свойства. Методы этой группы рекомендуют объекты, которые наиболее близки к требованиям пользователя. То есть объект рекомендации могут не полностью удовлетворять требованиям пользователя. База знаний в такой системе, кроме знаний о самих товарах, может иметь ограничения на пользовательские свойства.

Идея использования жестких ограничений заключается в поиске по базе данных товаров, удовлетворяющих запросам пользователей. Основной проблемой будет являться получение критериев от пользователя. Таким образом, главная задача жестких ограничений сводится к получению системы, удобной для пользователя.

* + 1. Гибридные рекомендательные системы

В гибридных рекомендательных системах используются сочетания методов, описанных ранее. Комбинирование методов ограничивается фантазиями и возможностями разработчиков. Выделяют несколько типов организации объединений методов рекомендационных систем: монолитная, параллельная и конвейерная организации.

* + - 1. Монолитная организация

При монолитной организации рекомендательная система представляет собой один модуль, который включает в себя разные методы обработки данных. Схема организации такой системы представлена на рисунке 3.

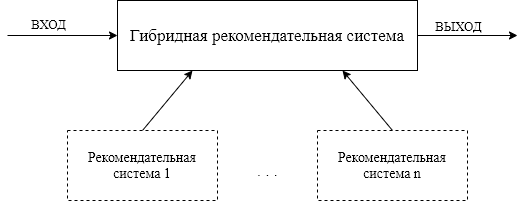


Рисунок 3 – Монолитная гибридная рекомендательная система

Большинство систем, использующих монолитную организацию, реализуют коллаборативную фильтрацию с фильтрацией, основанной на знаниях.

* + - 1. Параллельная организация

При параллельной организации данные подаются на каждый модуль системы. После того, как все модули выдадут свой ответ, начинает работу агрегирующий модуль. Схема организации такой системы представлена на рисунке 4.

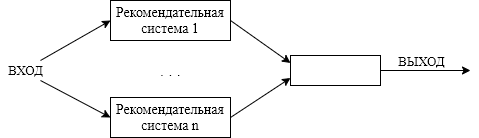


Рисунок 4 – Параллельная гибридная рекомендательная система

Агрегирующий модуль занимается выделением наиболее релевантных рекомендаций. Подходы работы такого модуля:

* смешивание всех рекомендаций и выдача их пользователю;
* анализ рейтинга для каждого объекта от всех модулей и выдача объектов с наибольшим рейтингом;
* выбор результатов одного модуля в зависимости от ситуаций и выдача результатов работы этого модуля.
  + - 1. Конвейерная организация

При работе гибридной рекомендательной системы с конвейерной организацией задача рекомендации делится на этапы, а результат работы предыдущего этапа передается в следующий. Схема организации такой системы представлена на рисунке 4.



Рисунок 5 – Конвейерная гибридная рекомендательная система

* + 1. Выводы

Среди представленных подходов к созданию рекомендательных систем для ИС поиска мероприятий наибольшую эффективность будет иметь контент-ориентированный анализ в связке с коллаборативной фильтрацией, так как адаптация «content-based» метода позволит минимизировать проблему холодного старта для новых пользователей и мероприятий, а коллаборативная фильтрация позволит отобрать среди результатов работы контент-ориентированного анализа наиболее релевантные объекты. Таким образом целесообразно выбрать гибридную рекомендательную систему с параллельной организацией.

* 1. Постановка задачи на разработку ИС
     1. Назначение ИС

ИС обеспечивает подбор мероприятий, которые с наибольшей вероятностью понравятся пользователю.

* + 1. Задачи

ИС должна решать следующие задачи:

* хранение и учет данных о мероприятиях;
* хранение и учет данных о местах проведения мероприятий;
* хранение и учет данных о пользователях системы;
* анализ данных о посещенных пользователями мероприятиях;
* разграничение прав доступа пользователей.
  + 1. Функциональные требования

ИС должна обеспечивать в части хранения и учета данных о мероприятиях следующие требования:

* регистрацию организатора мероприятий в системе и присвоение соответствующей роли;
* регистрацию нового мероприятия;
* архивирование неактуальных мероприятий;
* редактирование существующих мероприятий, принадлежащих определенному организатору.

ИС должна обеспечивать в части хранения и учета данных о местах проведения мероприятий следующие требования:

* регистрацию владельца места в системе и присвоение соответствующей роли;
* регистрацию нового места;
* удаление места;
* редактирование информации о существующих местах, принадлежащих определенному владельцу.

ИС должна обеспечивать в части хранения и учета данных о пользователях системы:

* регистрацию пользователя в системе и присвоение соответствующей роли;
* редактирование профиля пользователя;
* подписку на других пользователей системы;
* регистрацию на мероприятия;
* оценивание посещенных мероприятий;
* просмотр информации о других пользователях.

ИС должна обеспечивать в части анализа данных о посещенных пользователями мероприятиях:

* анализ данных мероприятий;
* анализ оценок мероприятий пользователями;
* предоставление результатов анализа в виде списка рекомендованных мероприятий.

ИС должна обеспечивать в части разграничения прав доступа пользователей:

* предоставление различных прав доступа пользователям, имеющим соответствующие роли.
* изменение данных о мероприятии только организаторами этих мероприятий;
* изменение данных о местах проведения мероприятий только владельцами этих мест;
* редактирование информации о пользователе только самими пользователями;
* изменение прав доступа пользователей, владельцев мест и организаторов мероприятий только администраторами информационной системы.
  + 1. Требование к ролям пользователей

Для пользователей с ролью «Администратор информационной системы» предоставляется доступ к изменению ролей пользователей, разархивации данных об архивированных мероприятиях и изменению логина пользователей.

Пользователи с ролью «Организатор мероприятий» должен иметь возможность зарегистрировать новое мероприятие, изменить данные о зарегистрированном мероприятии и архивировать мероприятие.

Пользователи с ролью «Владелец места» имеют доступ к регистрации места, изменению информации о зарегистрированном месте и удалению зарегистрированного места.

Пользователям с ролью «Пользователь» и остальными ролями предоставляется доступ к редактированию данных о своем профиле, подписке на других пользователей, регистрации на мероприятие и оценке посещенных мероприятий.

* + 1. Требования к эргономике и технической эстетике

Система должна обеспечивать удобный пользовательский интерфейс для конечного пользователя, который соответствует следующим требованиям:

* наличие локализированного (русскоязычного) интерфейса пользователя;
* экранные формы должны располагаться в видимой площади экрана монитора с диагональю 14 дюймов при разрешении экрана 1920 на 1080 пикселей и выше;
* отображение на экране возможностей, специфических для роли пользователя.
  1. Выводы по исследовательскому разделу

В данной главе были описаны цели создания и назначения ИС поиска мероприятий. Был проведен анализ алгоритмов работы рекомендательных систем и выбран тип рекомендательной системы с планом организации. Также в данном разделе были выявлены требования к ИС на основе анализа бизнес-процесса и выбранной стратегии работы рекомендательной системы.

1. Аналитический раздел

В данном разделе будет разработана архитектура и структура базы данных для ИС поиска мероприятий, а также определено математическое и техническое обеспечение ИС.

* 1. Разработка архитектуры ИС
     1. Контекстная диаграмма

При проектировании архитектуры системы необходимо в первую очередь создать смоделировать контекстную диаграмму. Эта диаграмма позволит предотвратить ошибки при дальнейшем проектировании. Контекстная диаграмма определяет окружение, контекст, в котором будет функционировать система. Для создания диаграммы необходимо определить пользователей и внешние системы. Контекстная диаграмма изображена на рисунке 6.

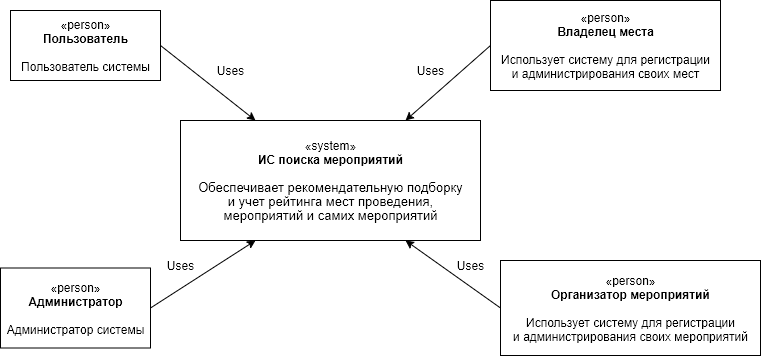


Рисунок 6 – Контекстная диаграмма

На данной диаграмме изображены объекты взаимодействия с системой. Рассмотрим их подробнее:

* ИС поиска мероприятий. Система, которая обеспечивает рекомендательную подборку и учет рейтинга мест проведения мероприятий и самих мероприятий.
* Администратор. Пользователь в системе, имеющий роль «Администратор», который управляет ролями других пользователей и имеет право на редактирование логинов пользователей, а также разархивировать информацию архивированных мероприятий.
* Владелец места. Пользователь в системе с ролью «Владелец места», который использует систему для регистрации и администрирования своих мест.
* Организатор мероприятий. Пользователь в системе с ролью «Организатор мероприятий», который использует систему для регистрации и администрирования своих мероприятий.
* Пользователь. Пользователь в системе с ролью «Пользователь», который использует систему для получения рекомендательной подборки мероприятий, а также информации о зарегистрированных мероприятиях в системе.
  + 1. Варианты использования

Для анализа вариантов использования была смоделирована диаграмма вариантов использования. Она демонстрирует взаимодействие пользователя с системой на высоком уровне. Данная диаграмма не описывает внутреннее устройство системы, однако помогает упростить взаимодействие с пользователями этой системы. Диаграмма основных вариантов использования представлена на рисунке 7.

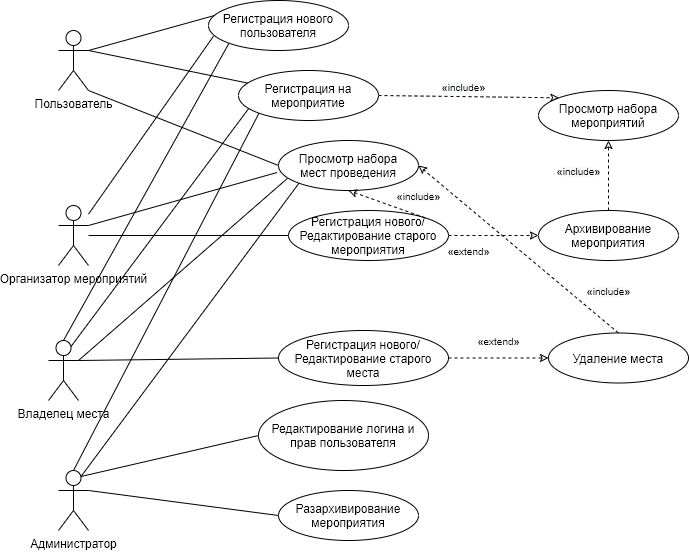


Рисунок 7 - Диаграмма основных вариантов использования

В данной диаграмме можно увидеть обязательные и необязательные отношения среди прецедентов, а также основные прецеденты, которые могут выполнять пользователи с соответствующими ролями.

Рассмотрим основные сценарии вариантов использования ИС поиска мероприятий, представленных в таблицах 1 – 4.

Таблица 1 – Сценарий «Регистрация нового пользователя»

|  |  |
| --- | --- |
| **Название прецедента** | Регистрация нового пользователя |
| **Действующие лица** | Пользователь |
| **Цель** | Зарегистрировать нового пользователя |
| **Предусловия** | Нет |
| **Основной сценарий:**   1. Пользователь открывает форму регистрации; 2. Пользователь вводит данные в форму; 3. Пользователь нажимает на кнопку «Зарегистрироваться»; 4. Система записывает информацию о новом пользователе в БД; 5. Система высылает код активацию на почту пользователя; 6. Пользователь переходит по ссылке и активирует свой аккаунт.   **Альтернативный сценарий:**   1. Пользователь открывает форму регистрации; 2. Пользователь вводит данные в форму; 3. Пользователь нажимает на кнопку «Зарегистрироваться»; 4. Пользователю выводится сообщение, что пользователь с желаемым логином уже зарегистрирован. | |
| **Результат основного сценария** | Пользователю предоставлен доступ к системе |

Таблица 2 – Сценарий «Регистрация на мероприятие»

|  |  |
| --- | --- |
| **Название прецедента** | Регистрация на мероприятие |
| **Действующие лица** | Пользователь |
| **Цель** | Зарегистрироваться на мероприятие |
| **Предусловия** | Пользователь авторизирован |
| **Основной сценарий:**   1. Пользователь нажимает кнопку «Подробнее» на интересующем мероприятии; 2. Пользователю выводится информация о мероприятии; 3. Пользователь нажимает на кнопку «Подписаться»; 4. Система регистрирует пользователя на мероприятие. | |
| **Результат основного сценария** | Пользователь в списках участников мероприятия |

Таблица 3 – Сценарий «Регистрация нового мероприятия»

|  |  |
| --- | --- |
| **Название прецедента** | Регистрация нового мероприятия |
| **Действующие лица** | Организатор мероприятий |
| **Цель** | Зарегистрировать в системе новое мероприятие |
| **Предусловия** | Пользователь авторизирован, пользователь имеет роль организатора мероприятий |
| **Основной сценарий:**   1. Пользователь открывает форму регистрации нового мероприятия; 2. Пользователь вносит данные о мероприятии; 3. Система записывает информацию о новом мероприятии в базу данных; 4. Пользователь видит новое мероприятие среди остальных. | |
| **Результат основного сценария** | Новое мероприятие в списке мероприятий |

Таблица 4 – Сценарий «Регистрация нового места»

|  |  |
| --- | --- |
| **Название прецедента** | Регистрация нового места |
| **Действующие лица** | Владелец места |
| **Цель** | Зарегистрировать в системе новое место |
| **Предусловия** | Пользователь авторизирован, пользователь имеет роль владельца мест |
| **Основной сценарий:**   1. Пользователь открывает форму регистрации нового места; 2. Пользователь вносит данные о месте; 3. Система записывает информацию о новом места в базу данных; 4. Пользователь видит новое место среди остальных. | |
| **Результат основного сценария** | Новое место в списке мест |

* + 1. Описание архитектуры ИС

Для удовлетворения выявленных требований для ИС поиска мероприятий из исследовательского раздела было принято решение использовать многоуровневую клиент-серверную архитектуру с тремя уровнями:

1. Клиентская часть;
2. Сервер приложений;
3. Сервер базы данных.

Схематическое представление трехуровневой архитектуры представлено на рисунке 8.

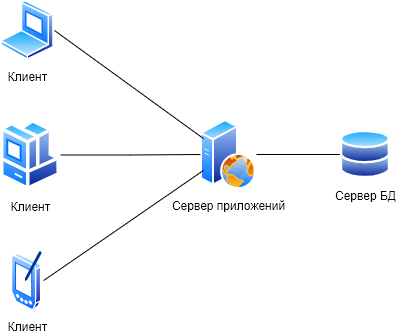


Рисунок 8 - Трехуровневая клиент-серверная архитектура

Клиентская часть представляет собой приложение для конечного пользователя, которое отвечает за отображение информации.

Сервер приложений включает в себя все основные расчеты системы, работу с базой данных, а также отправку клиентской части на устройство конечного пользователя.

Сервер базы данных отвечает только за хранение данных. К этому серверу имеет доступ только сервер приложений. Физически сервера базы данных и приложений будут развернуты на одном физическом сервере.

Описанная выше архитектура позволит использовать систему на клиентских терминалах без высоких требований к производительности и техническим характеристикам устройства пользователей. Также, из достоинств данной архитектуры можно выделить:

* повышенную защищенность от различных угроз;
* разграничение уровней доступа к уровням ИС;
* кроссплатформенность, то есть независимость от операционной системы на клиентском устройстве.

Для рассмотрения физического уровня представления разрабатываемой системы была создана диаграмма компонентов, которая представлена на рисунке 9.

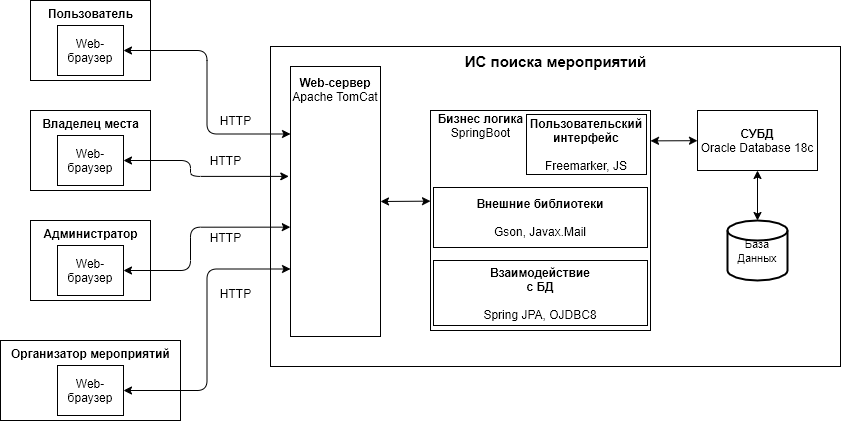


Рисунок 9 – Диаграмма компонентов

В данной диаграмме представлены основные компоненты, которые взаимодействуют между собой в системе: пользователи и информационная система.

К группе компонентов пользователей относятся:

* пользователь;
* владелец места;
* организатор мероприятий;
* администратор.

Использование Web-браузера на стороне клиента позволит добиться от информационной системы высокой доступности, так как в современных условиях доступ в интернет пространство имеет практически каждое устройство. Также, при использовании браузера, устройство конечного пользователя не будет нуждаться в специфическом программном обеспечении для доступа к серверу.

Блок информационной системы делится на внутренние компоненты: Web-сервер, модуль бизнес логики, пользовательский интерфейс, внешние библиотеки, модуль взаимодействия с базой данных и система управления базой данных.

Web-сервер принимает обращения от пользовательских клиентов, передает их в модуль бизнес логики. После обработки запроса логическим модулем, ответ передается обратно на Web-сервер, который передает полученные данные на устройство пользователя.

Модуль бизнес логики занимается обработкой данных в соответствии с запросом пользователя. Во время обработки запросов, логический модуль может обращаться к системе управления базой данных для получения необходимой информации для обработки этих запросов.

Модуль пользовательского интерфейса отвечает за представление данных пользователю. В данном модуле использовался язык JavaScript для динамического отображения контента на web-странице, загружаемой пользователем. Вместе с этим была использована Java-библиотека Freemarker для отображения данных.

Среди внешних библиотек использовались Gson и Javax.Mail.

Gson был необходим для сериализации данных в модуле бизнес логики, чтобы эти данные можно было в дальнейшем использовать в языке программирования JavaScript.

Библиотека Javax.Mail использовалась для отправки электронных писем зарегистрировавшимся пользователям с ссылкой на подтверждение зарегистрированного аккаунта.

Для взаимодействия с базой данных и представления объектов базы в системе использовались библиотеки OJDBC8 и Spring Data JPA, часть обширного семейства библиотек Spring Data.

OJDBC8 используется для подключения к системе управления базой данных Oracle Database 18c.

Библиотека Spring Data JPA необходима для маппинга объекта класса системы в таблицу базы данных и наоборот, представлять значения из таблицы в виде объекта классы системы. Также данная библиотека решает большой спектр задач, связанных с объектно-реляционным отображением.

Компонент базы данных отвечает за хранение информации в базе.

Для взаимодействия непосредственно с базой данных используется система управления базой данных Oracle Database 18c. СУБД занимается созданием и управлением базами данных.

* 1. Разработка структуры базы данных

Для хранения информации в ИС поиска мероприятий будет использоваться система управления базой данных Oracle Database 18c. Для описания данных будут использоваться 9 таблиц, которые представлены на рисунке 10.

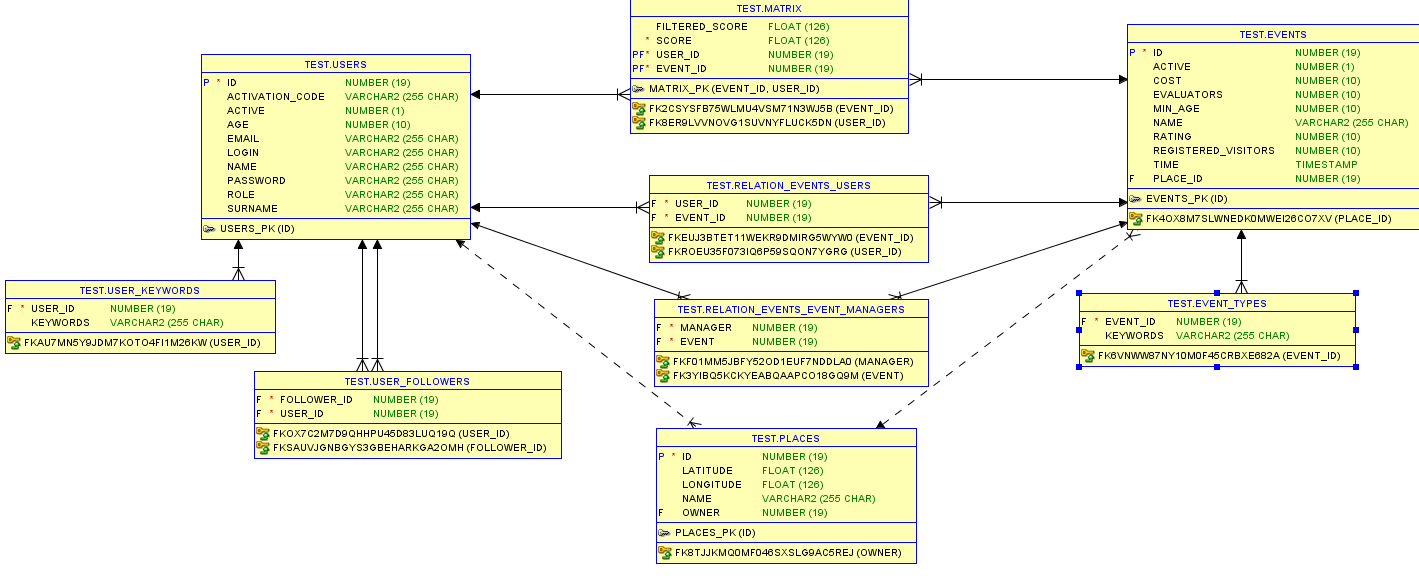


Рисунок 10 – ER-модель базы данных

ER-модель предназначена для описания связей между таблицами в базе данных.

Для понимания, какие данные будут храниться в базе необходимо подробнее рассмотреть таблицы, представленные на диаграмме.

* + 1. Таблица USERS

В данной таблице хранятся данные об учетных записях пользователей. Рассмотрим поля этой таблицы:

* ID. ID пользователя в системе Служит первичным ключом таблицы. Тип данных – девятнадцатизначное целое число.
* ACTIVATION\_CODE. Код активации, которая задает система при регистрации пользователя. Тип данных – строка, длиной до 255 символов.
* ACTIVE. Статус пользователя, который говорит об активности профиля. Тип данных – однозначная цифра, где 1 обозначает, что профиль активен.
* AGE. Возраст пользователя. Тип данных – целое число.
* EMAIL. Электронная почта пользователя. Тип данных – строка, длиной до 255 символов.
* LOGIN. Логин пользователя в системе. Тип данных – строка, длиной до 255 символов.
* NAME. Имя, указанное пользователем при регистрации. Тип данных – строка, длиной до 255 символов.
* SURNAME. Фамилия, указанная пользователем при регистрации. Тип данных – строка, длиной до 255 символов.
* PASSWORD. Пароль для входа в учетную запись пользователя. Тип данных – строка, длиной до 255 символов.
* ROLE. Роль пользователя в системе. Тип данных – строка, длиной до 255 символов.
  + 1. Таблица EVENTS

Данная таблица хранит данные о событиях, которые регистрируют организаторы мероприятий.

Поля таблицы:

* ID. Id мероприятия в системе. Служит первичным ключом таблицы. Тип данных – девятнадцатизначное целое число.
* ACTIVE. Статус мероприятия, который говорит об его активности. Тип данных – однозначная цифра, где 1 обозначает, что мероприятие активно.
* COST. Цена за вход на мероприятие. Тип данных – десятизначное целое число.
* MIN\_AGE. Минимальный возраст для посещения мероприятий. Тип данных – десятизначное целое число.
* NAME. Название мероприятия. Тип данных – строка, длиной до 255 символов.
* RATING. Рейтинг мероприятия, которые выставляют пользователи. Тип данных – десятизначное целое число.
* REGISTERED\_VISITORS. Количество пользователей, которые зарегистрировались на мероприятие. Тип данных – десятизначное целое число.
* TIME. Время проведения мероприятия. Тип данных – дата.
* PLACE\_ID. ID места проведения мероприятия. Это поле связано внешним ключом с таблицей PLACES. Тип данных – девятнадцатизначное число.
* EVALUATORS. Количество пользователей, оценивших мероприятие. Тип данных – десятизначное число.
  + 1. Таблица PLACES

Таблица PLACES предназначена для хранения данных о местах проведения мероприятий. Эта таблица заполняется пользователями с правами «Владелец места».

Поля таблицы:

* ID. ID места в системе. Служит первичным ключом таблицы. Тип данных – девятнадцатизначное целое число.
* LATITUDE. Координата широты геопозиции места. Тип данных – число с плавающей точкой.
* LONGITUDE. Координата долготы геопозиции места. Тип данных – число с плавающей точкой.
* NAME. Название места. Тип данных – строка, длиной до 255 символов.
* OWNER. Владелец места. Поле связано внешним ключом с таблицей USERS. Тип данных – девятнадцатизначное число.
  + 1. Таблица MATRIX

Данная таблица хранит в себе матрицу предпочтений, которая необходима для реализации метода коллаборативной фильтрации. В ней содержатся данные об оценках мероприятий от пользователей. Также здесь будет хранится информация об уже рассчитанных рекомендациях.

Поля таблицы:

* USER\_ID. ID пользователя в системе. Является частью составного первичного ключа таблицы. Связан с помощью внешнего ключа с таблицей USERS. Тип данных – девятнадцатизначное число.
* EVENT\_ID. ID мероприятия в системе. Также часть составного первичного ключа таблицы. Связан с помощью внешнего ключа с таблицей EVENTS. Тип данных – девятнадцатизначное число.
* SCORE. Оценка, который дал пользователь мероприятию. Тип данных – число с плавающей точкой.
* FILTERED\_SCORE. Предполагаемая оценка, которую рассчитала рекомендательная система. Тип данных – число с плавающей точкой.
  + 1. Таблицы USER\_KEYWORDS и EVENT\_TYPES

Данные таблицы хранят в себе ключевые слова, отображающие интересы пользователей и характеристики мероприятий.

Поля таблицы USER\_KEYWORDS:

* USER\_ID. ID пользователя в системе. Связано с помощью внешнего ключа с таблицей USERS. Тип данных – девятнадцатизначное число.
* KEYWORDS. Ключевое слово, отображающее предпочтение пользователя. Тип данных – строка, длиной до 255 символов.

Поля таблицы EVENT\_TYPES:

* EVENT\_ID. ID мероприятия в системе. Связано с помощью внешнего ключа с таблицей EVENTS. Тип данных – девятнадцатизначное число.
* KEYWORDS. Ключевое слово, отображающее характеристику мероприятия. Тип данных – строка, длиной до 255 символов.
  + 1. Остальные таблицы

Таблицы RELATION\_EVENTS\_USERS и RELATION\_EVENTS\_EVENT\_MANAGERS отображают связи между таблицами EVENTS и USERS. Каждая из таблиц имеет два поля, которые содержат ID пользователя и ID мероприятия в системе. Каждое из полей связано с соответствующей таблицей с помощью внешнего ключа. Таблица RELATION\_EVENTS\_USERS хранит данные о пользователях, которые зарегистрировались на мероприятие. Таблица RELATION\_EVENTS\_EVENT\_MANAGERS содержит информацию об организаторах мероприятий и их мероприятиях.

Таблица USER\_FOLLOWERS предназначена для отображения отношений между пользователями. Таблица содержит информацию о том, кто на кого подписан. В таблице используются два поля, оба из которых связаны с помощью внешних ключей с таблицей USERS. Сразу на два поля определено ограничение уникальности, то есть в таблице не может быть две одинаковые записи.

* 1. Математическое обеспечение ИС

Для более лучшего понимания работы рекомендательной системы в составе ИС поиска мероприятий в данном раздели будут описаны математические методы и алгоритмы, используемые при создании рекомендательной системы.

* + 1. Коллаборативная фильтрация, основанная на соседстве товаров

В основе рекомендательной системы, реализованной методом коллаборативной фильтрации, лежит матрица предпочтений, которая представляет собой множество пользователей по вертикали и множество объектов рекомендации по горизонтали. Такая матрица заполняется оценками, которые ставит пользователь товару. Задача рекомендательной системы – предсказать оценку пользователя для товара, которому пользователь еще не дал своей оценки. Демонстрация матрицы предпочтений представлена на рисунке 11.

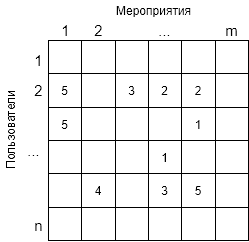


Рисунок 11 – Матрица предпочтений

Для реализации метода коллаборативной фильтрации, основанной на соседстве товаров, необходимо для каждого мероприятия *m* рассчитать схожесть с мероприятием *a*. Для этой оценки в работе будет использована мера близости, которая будет обозначена, как *sim(a, m)* – коэффициент корреляции Пирсона (формула 1).

(1)

В этой формуле *U –* множество пользователей, которые оценили оба мероприятия: *a* и *m.* и являются средними оценками мероприятий, то есть сумма всех оценок пользователей, деленая на количество оценивавших пользователей. является оценкой, данной пользователем *x* мероприятию *y.* В числителе высчитывается произведение отклонений оценок мероприятий от средних значений для одного мероприятия. Знаменатель регулирует расположение коэффициента на отрезке от -1 до 1. Чем коэффициент ближе к 1, тем выше схожесть мероприятий.

После определения схожести мероприятий необходимо отобрать такое множество *M*, которое будет включать наиболее похожие на *m* мероприятий. Для этого установим количественный порог для множества *M* равный 5. Таким образом, с помощью сортировки по коэффициенту корреляции Пирсона, во множество *M* будут входить пять мероприятий, которые наиболее похожи на мероприятие *m.*

После определения множества *M* задача сводится к расчёту взвешенной средней оценки похожих объектов, полученной пользователем *x* (формула 2).

(2)

* + 1. Контент ориентированный анализ

Контент ориентированный анализ будет использоваться для решения проблем холодного старта. Данная проблема возникает, когда пользователь только зарегистрировался в системе и в ней еще нет данных об оценках мероприятий пользователем. Также «content-based» анализ будет использоваться для устранения других проблем, когда коллаборативная фильтрация не будет давать результата, например:

* при слишком большой разреженности матрицы предпочтения;
* при регистрации новых мероприятий, которые будут рекомендоваться реже проверенных старых, периодически повторяющихся, популярных мероприятий;
* при учете тех оценок пользователей, которые не совпадают с большинством других пользователей.

Для такого вида рекомендаций будут использоваться знания о предпочтениях пользователя, которые он указал при регистрации в системе, и анализ мероприятий по ключевым словам. Таким образом, сравнение двух мероприятий будет сводится к задаче вычисления коэффициента Дайса, продемонстрированном в формуле 3.

(3)

В формуле Х является множеством ключевых слов мероприятия , которые организатор занес при регистрации мероприятия. Данная формула считает количество элементов, которые существуют сразу в двух множествах и делит их на сумму количества элементов множеств. Коэффициент Дайса лежит в пределах отрезка от 0 до 1 и показывает, насколько близки сравниваемые множества.

Адаптация данного алгоритма будет заключаться в использовании для сравнительного анализа только тех мероприятий, которым пользователь поставил оценку не ниже определенного в системе коэффициента.

Для пользователей, которые еще не оценили ни одного мероприятия будут рекомендованы те события, которые максимально похожи на предпочтения, указанные пользователем при регистрации.

* + 1. Агрегатор рекомендательных систем

Суть данной модели заключается в определении наиболее релевантных мероприятий для пользователя. С помощью списков, отсортированных по предпочтениям пользователя от наиболее подходящего к менее, выбираются те элементы, которые пересекаются среди двух множеств. Дополнением будет результат работы коллаборативной фильтрации с высокой оценкой, который не пересекается с результатом «content-based» анализа.

Для пользователя, который оценил количество мероприятий, больше установленного в системе, множество рекомендуемых мероприятий будет рассчитано, как показано на формуле 4.

(4)

В формуле 4 *A –* множество рекомендованных мероприятий, полученных с помощью коллаборативной фильтрации, *B* ***–*** множество мероприятий, полученных в результате работы контент ориентированного анализа. Условием Q(m) является принадлежность элемента m множеству *A* и высокая предсказанная оценка элемента *m.*

Для пользователя, который только зарегистрировался в системе множество рекомендованных мероприятий будет определено, как показано на формуле 5.

(5)

Из этой формулы видно, что для нового пользователя множество рекомендованных мероприятий будет составлять только из результата работы контент ориентированного анализа.

* 1. Техническое обеспечение ИС

Техническое обеспечение ИС включает в себя комплекс технических средств, необходимых для работы этой ИС.

Техническое обеспечение информационной системы поиска мероприятий должно быть организованно централизованным способом, то есть будет использоваться база данных в единственном экземпляре, а взаимодействие с ней будет обеспечиваться сервером с помощью удаленных запросов. Благодаря такой схеме у пользователя не будет доступа к исходным данным, а информация будет иметь незначительный размер для передачи на клиентское устройство пользователя.

* 1. Выводы по исследовательскому разделу

В данном разделе была спроектирована архитектура ИС поиска мероприятия и разработана структура базы данных, в соответствии с требованиями, выявленными в аналитическом разделе. Также было определено и описано математическое и техническое обеспечение ИС, рассмотрены некоторые варианты использования системы.

1. Экономический раздел

В данном разделе будет произведена организация и планирование работ по разработке ИС поиска мероприятий, оценка затрат на разработку и расчет договорной цены разработки ИС.

* 1. Организация и планирование работ по теме

В составе работы задействовано 3 человека:

1. руководитель (научный руководитель выпускной квалификационной работы, доцент базовой кафедры №250) – отвечает за грамотную постановку задачи, контролирует этапы работы, вносит необходимые коррективы и оценивает выполненную работу в целом;
2. консультант (консультант по экономической части выпускной квалификационной работы) – отвечает за консультирование экономической части выпускной квалификационной работы;
3. разработчик (студент 4 курса Института информационных технологий) – реализация всех поставленных задач, в том числе проведение тестирования готового продукта и подготовка проектной документации.

Состав задействованных в работе участников представлен на рисунке 12.

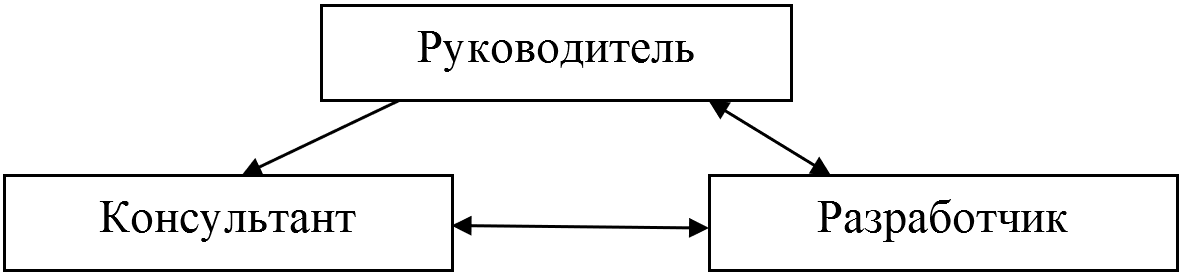


Рисунок 12 – Схема задействованных участников в работе

* + 1. Организация работ

Для создания рабочего прототипа ИС поиска мероприятий необходимо разбить работу на несколько этапов. На разработку отводится 82 рабочих дня. Этапы разработки представлены в таблице 5.

Таблица 5 - Этапы разработки

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Название этапа** | **Исполнитель** | **Трудоемкость,**  **чел/дни** | **Продолжительность работ, дни** |
| 1 | Разработка и утверждение технического задания | Руководитель | 5 | 5 |
| Разработчик | 5 |
| 2 | Технические предложения | Руководитель | 6 | 6 |
| Разработчик | 6 |
| 3 | **Эскизный проект** | | | 18 |
| 3.1 | Анализ исходных данных и требований | Руководитель | 5 |
| Разработчик | 7 |
| 3.2 | Постановка задачи | Руководитель | 2 |
| Разработчик | 2 |
| 3.3 | Разработка общего описания алгоритма функционирования | Руководитель | 5 |
| Разработчик | 10 |
| 4 | **Технический проект** | | | 15 |
| 4.1 | Определение формы представления входных и выходных данных | Руководитель | 3 |
| Разработчик | 5 |
| 4.2 | Разработка структуры программы и логической структуры | Разработчик | 10 |
| 5 | **Рабочий проект:** | | | 38 |
| 5.1 | Программирование и отладка программы | Разработчик | 25 |
| 5.2 | Испытание программы | Разработчик | 3 |

Продолжение таблицы 5

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5.3 | Корректировка программы по результатам испытаний | Разработчик | 5 |  |
| 5.4 | Подготовка технической документации на программный продукт | Руководитель | 5 |
| Консультант | 5 |
| Разработчик | 5 |
| 5.5 | Сдача готового продукта и внедрение | Руководитель | 1 |
| Консультант | 1 |
| Разработчик | 2 |
| **Итого** | | | **118** | **82** |

* + 1. График проведения работ

Календарный график исполнения работы представлен на рисунке 13. Из графика так же видно, что общий срок разработки составит 82 дня.

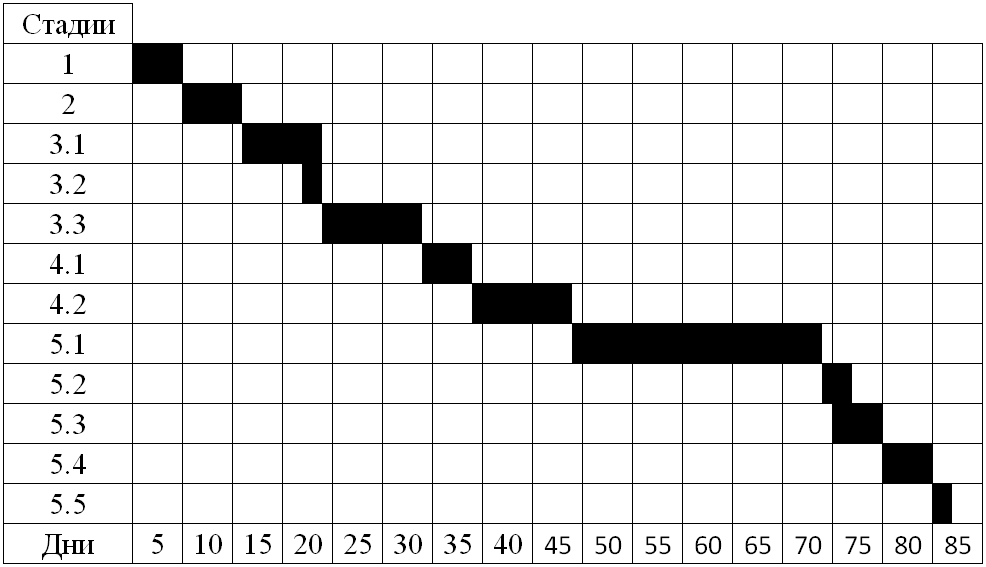


Рисунок 13 – Календарный график работы

* + 1. Расчет стоимости проведения работ по теме
       1. Общие сведения

1 статья «Материалы, покупные изделия и полуфабрикаты + ТЗР (15%) от ∑ итого по материалам

себестоимость

2 статья «Специальное оборудование» - как правило, затрат нет

3 статья «Основная заработная плата»

4 статья «Дополнительная заработная плата» 20-30% от основной заработной платы

5 статья «Страховые отчисления» - 30% от ФОТ

6 статья «Командировочные расходы» - как правило, затрат нет

7 статья «Контрагентские услуги» - как правило, затрат нет

8 статья «Накладные расходы» - 250% от основной заработной платы

9 статья «Прочие расходы» - затрат нет

В выпускной квалификационной работе объем затрат на НИР и ОКР был проведен методом калькулирования.

* + - 1. Статья №1 «Материалы, покупные изделия и полуфабрикаты»

Для вычисления затрат для данной статьи необходимо определить состав всех затрачиваемых материалов и покупных изделий: бумага для печати, запоминающие устройства, пишущие принадлежности и другие канцелярские товары. Затраты по этой статье представлены в таблице 6.

Таблица 6 – Затраты по статье «Материалы, покупные изделия и полуфабрикаты»

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№ пп** | **Наименование материалов** | **Единицы измерения** | **Кол-во** | **Цена за единицу (руб)** | **Стоимость (руб)** |
| 1 | USB накопитель 8Гб | шт | 1 | 490 | 490 |
| 2 | Бумага А 4 | пачка | 1 | 350 | 350 |
| 3 | Картридж для принтера | шт | 1 | 1089 | 1089 |
| 4 | Ручка | шт | 2 | 42 | 84 |

Продолжение таблицы 6

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | Папка | шт | 1 | 215 | 215 |
| 6 | Карандаш | шт | 1 | 49 | 49 |
| 7 | Брошюровка | До 150 листов | 1 | 450 | 450 |
| **Итого материалов** | | | | | **2727** |
| **Транспортно-заготовительные расходы** | | | | | **410** |
| **Итого** | | | | | **3 137** |

* + - 1. Статья №2 «Специальное оборудование»

Расходы по данной статье отсутствуют.

* + - 1. Статья №3 «Основная заработная плата»

Данная статья содержит расходы на плату работникам за отработанное время и выполненную работу. Для расчета основной заработной платы учитываются: величина месячного должностного оклада исполнителя и трудоёмкость каждого из этапов работы. Данные о расходах по этой статье представлены в таблице 7.

Таблица 7 - Затраты на основную заработную плату

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **№ пп** | **Наименование этапа** | **Исполнитель (должность)** | **Мес. оклад (руб)** | **Трудоемкость (чел/дни)** | **Оплата за день (руб)** | **Оплата за этап (руб)** |
| 1 | ТЗ | Руководитель | 80 000 | 5 | 3636 | 18180 |
| Разработчик | 40 000 | 5 | 1818 | 9090 |
| 2 | ТП | Руководитель | 80 000 | 6 | 3636 | 21816 |
| Разработчик | 40 000 | 6 | 1818 | 10908 |
| 3 | Эскизный проект | Руководитель | 80 000 | 12 | 3636 | 43632 |
| Разработчик | 40 000 | 19 | 1818 | 34542 |

Продолжение таблицы 7

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 4 | Технический проект | Руководитель | 80 000 | 3 | 3636 | 10908 |
| Разработчик | 40 000 | 15 | 1818 | 27270 |
| 5 | Рабочий проект | Руководитель | 80 000 | 6 | 3636 | 21816 |
| Консультант | 60 000 | 6 | 2272 | 13632 |
| Разработчик | 40 000 | 40 | 1818 | 72720 |
| **Итого** | | | | | | **284 514** |

* + - 1. Статья №4 «Дополнительная заработная плата»

Эта статья содержит расходы, которые предусмотрены законодательством, на оплату за неотработанное по уважительным причинам время.

Дополнительная заработная плата, обычно, является 20% от суммы основной заработной платы.

* + - 1. Статья №5 «Страховые отчисления»

Страховочные отчисления составляют 30% от фонда оплаты труда. Фонд оплаты труда состоит из суммы основной и дополнительной заработных плат.

После вычисления фонда оплаты труда можно рассчитать страховые выплаты.

* + - 1. Статья №6 «Командировочные расходы»

Расходы по данной статье отсутствуют.

* + - 1. Статья №7 «Контрагентские услуги»

Так как в разработке данного проекта сторонние организации не привлекались, расходы по данной статье отсутствуют.

* + - 1. Статья №8 «Накладные расходы»

Накладные расходы представляют собой траты, которые не связаны напрямую с основным производством. Расходы по данной статье составляют 200% от основной заработной платы.

* + - 1. Статья №9 «Прочие расходы»

Расходы по данной статье отсутствуют.

* + - 1. Полная себестоимость проекта

Информация о полной себестоимости проекта представлена в таблице 8.

Таблица 8 – Себестоимость продукта

|  |  |  |
| --- | --- | --- |
| **№ пп** | **Номенклатура статей расходов** | **Затраты (руб)** |
| 1 | Материалы, покупные изделия и полуфабрикаты (за вычетом отходов) | 3 137 |
| 2 | Специальное оборудование для научных (экспериментальных) работ | ⎯ |
| 3 | Основная заработная плата научного и производственного персонала | 284 514 |
| 4 | Дополнительная заработная плата научного и производственного персонала | 56 902 |
| 5 | Страховые взносы в социальные фонды | 102 454 |
| 6 | Расходы на научные и производственные командировки | ⎯ |

Продолжение таблицы 8

|  |  |  |
| --- | --- | --- |
| 7 | Оплата работ, выполненных сторонними организациями и предприятиями | ⎯ |
| 8 | Накладные расходы | 569 228 |
| 9 | Прочие прямые расходы | ⎯ |
| **Итого** | | 1 016 235 |

* + - 1. Договорная цена программного продукта

Договорная цена представляет собой сумму себестоимости, прибыли и налога на добавленную стоимость.

В среднем прибыль составляет 20% от себестоимости продукта.

Так как разработка ведется для коммерческой организации, на разработку возлагается налог в размере 20%.

Таким образом, договорная цена будет составлять 1 463 378 рублей.

* 1. Выводы по экономическому разделу

Исходя из планирования работ и проведенных расчетов в данном разделе, разработка ИС поиска мероприятий должна занимать по времени 82 дня, а договорная цена системы будет составлять 1 463 378 рублей.

1. Технологический раздел

В технологическом разделе будет обоснован выбор средств для создания ИС поиска мероприятий, а также будет описан разработанный прототип и приведена его демонстрация.

* 1. Обоснование выбора средств разработки
     1. Язык программирования

Язык программирования Java, вместе с фреймворком Spring Boot является автономным и производительным инструментом, который содержит готовый набор настроенных конфигураций и позволит при разработке сосредоточится на бизнес-логике. Благодаря такой связке компонентов, разработчик имеет много возможностей для быстрого создания готового продукта и его дальнейшего масштабирования. Также стоит отметить, что из-за легкого в восприятии синтаксиса языка Java, поддержка системы не будет такой трудозатратой, как при использовании других языков программирования. Из остальных преимуществ языка Java стоит отметить большое количество материалов с ответами и вопросами по поводу программирования и автоматическое управление памятью.

* + 1. СУБД

Система управления базой данных Oracle Database имеет ряд, необходимых для работы с большим количеством данных, преимуществ и является одной из самых мощных реляционных СУБД. Среди преимуществ следует выделить технологию In-Memory Column Store, которая позволяет хранить в быстро доступной памяти данные из таблиц в колоночном формате, что ускоряет вставку строк.

* + 1. Пользовательский интерфейс

Для динамического отображения контента на web-странице был выбран язык JavaScript, так как он не имеет аналогов. Также была использована библиотека FreeMarker, так как она проста для освоения и обладает минимальным набором функций, необходимых для реализации работающего прототипа системы.

* 1. Описание разработанного программного продукта
     1. Разработанные классы и интерфейсы

В ходе разработки ИС поиска мероприятий было создано 25 классов и 5 интерфейсов.

* + - 1. Обработчики запросов

Обработчики запросов принимают http запросы и отправляют в ответ отображения данных. К таким обработчикам относятся 8 классов:

* AdminController – класс, отвечающий за обработку запросов от пользователей с ролью Администратор;
* EventController – класс, отвечающий за обработку запросов от пользователей, касающихся любых действий с мероприятиями, доступных обычному пользователю системы;
* EventManagerController – класс, отвечающий за обработку запросов от пользователей с ролью Организатор мероприятий;
* JSONController – класс, возвращающий в ответе на запросы пользователей информацию в формате JSON;
* MainController – класс, содержащий обработку запросов, связанных с главной страницей системы и активацией аккаунта;
* PlaceController – класс, обрабатывающий запросы, связанные с местами проведения мероприятий;
* PlaceManagerController – класс, который обрабатывает запросы пользователей с ролью Владелец места;
* UserController – обработчик запросов, связанных с данными о пользователях.
  + - 1. Классы-конфигурации

В классах конфигурациях содержатся настройки некоторых компонентов системы. К таким классам относятся:

* GsonConfig – класс с настройкой внешней библиотеки Gson сериализации и десериализации формата JSON;
* HiddenAnnotationExclusionStrategy – класс, который реализует функции интерфейса библиотеки Gson ExclusionStrategy и необходим для настройки библиотеки Gson;
* MailConfig – класс, содержащий конфигурацию для клиента отправки электронных писем пользователю;
* MvcConfig реализует интерфейс WebMvcConfigurer, используется для настройки библиотеки Spring Boot Security;
* WebSecurityConfig – класс, реализующий интерфейс WebSecurityConfigurerAdapter и используется для настройки библиотеки Spring Boot Security.
  + - 1. Сервисные классы

Сервисные классы содержат в себе вычисления и бизнес логику. К таким классам относятся:

* EventService – класс, содержащий логику операций с мероприятиями;
* MailService – класс, содержащий логику для отправки электронных сообщений;
* MatrixService – класс, содержащий логику операций с матрицей предпочтений;
* PlaceService – класс, содержащий логику операций с местами проведения мероприятий;
* UserService – класс, содержащий логику операций с данными пользователей;
* ContentBasedFilter – класс, содержащий логику контент-ориентированного анализа;
* ItemBasedFilter – класс, содержащий логику коллаборативной фильтрации, основанной на соседстве объектов рекомендаций;
* RecomendSystem – класс, который агрегирует результаты работы классов ContentBasedFilter и ItemBasedFilter.
  + - 1. Классы и интерфейсы для работы с БД

Для работы с базой данных было создано 5 классов, описывающих сущности таблиц и их связи:

* Event – описание таблицы EVENTS;
* Matrix – описание таблицы MATRIX;
* MatrixPK – класс для описания первичного ключа таблицы MATRIX;
* Place – описание таблцы PLACES;
* User – описание таблцы USERS.

Также, для работы с базой данных были определены интерфейсы-репозитории, которые наследуются от интерфейса механизма SpringData JpaRepository. Данные интерфейсы обеспечивают операции по поиску, удалению и сохранению информации в БД для соответствующих описанных сущностей.

* + - 1. Остальные классы и интерфейсы

К остальным классам и интерфейсам были отнесены:

* DateFormatter – класс для форматирования данных о дате;
* SpringappApplication – класс, запускающий работу системы;
* Hidden – интерфейс для настройки библиотеки Gson, скрывает поля сущности для сериализации.
  + 1. Сценарий диалога

Взаимодействие пользователя с системой может развиваться по сценарию диалога, который представлен на рисунке 14.

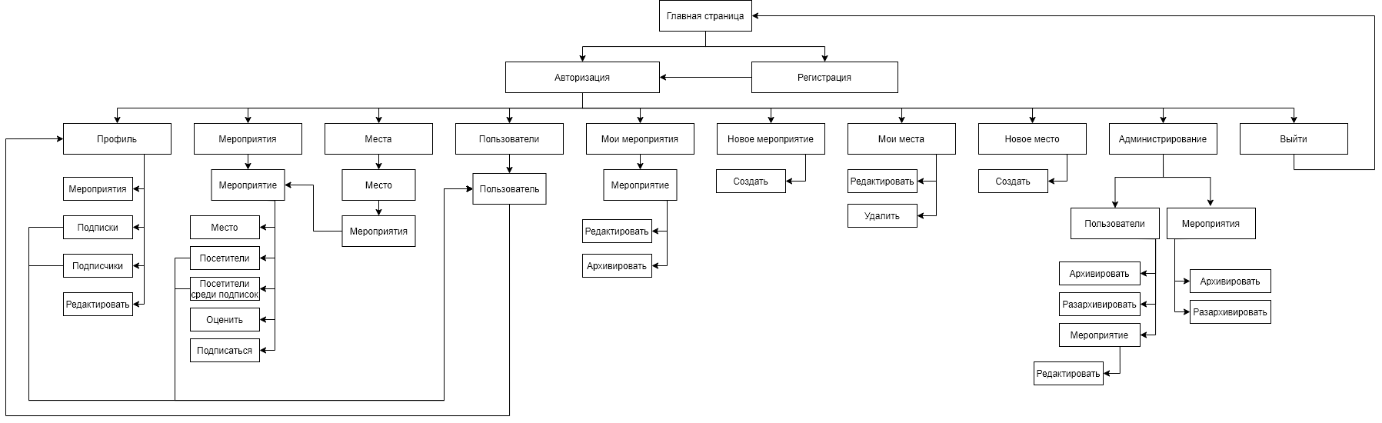


Рисунок 14 – Сценарий диалога

* + 1. Дерево функций

Дерево функций представляет собой диаграмму, на которой отображены основные функции ИС поиска мероприятий. Дерево функций представлено на рисунке 5.

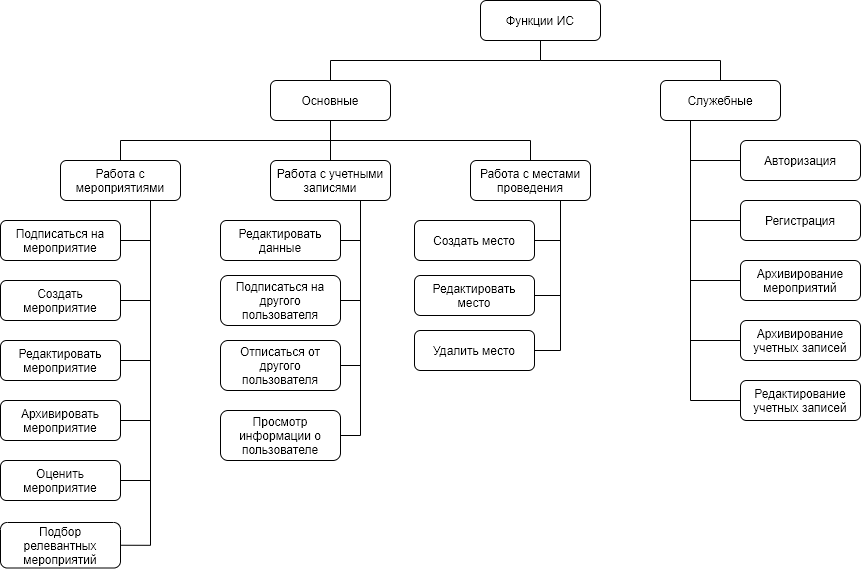


Рисунок 15 – Дерево функций ИС поиска мероприятий

* 1. Демонстрация работы программного продукта

В данном разделе будут представлены макеты экранных форм их описание.

При входе на сайт пользователю предложено зарегистрироваться или ввести свои данные и авторизоваться, что демонстрируется на рисунке 16.

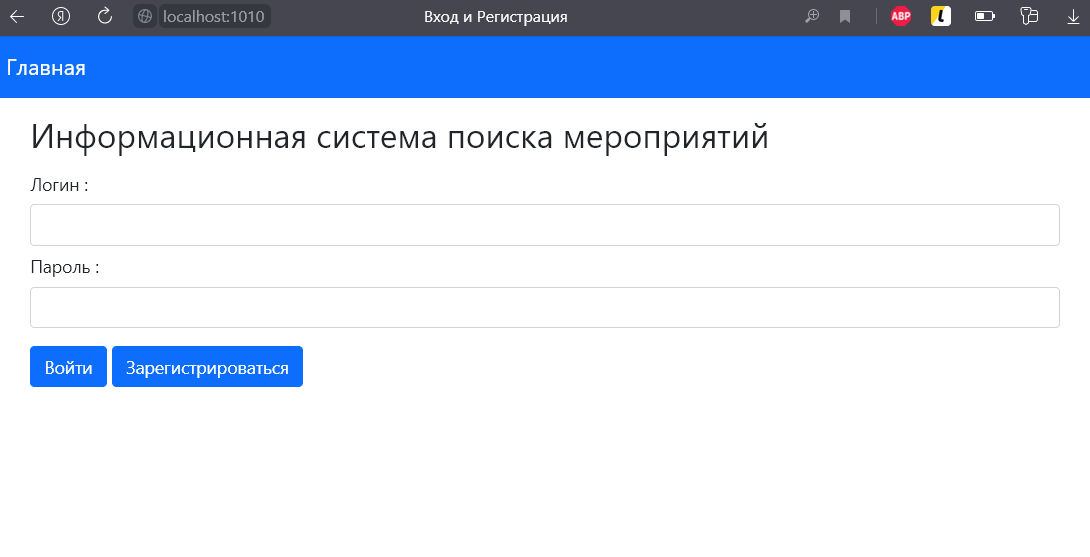


Рисунок 16 – Стартовая страница

Если пользователь нажал кнопу регистрации, перед ним откроется форма регистрации, продемонстрированная на рисунке 17.

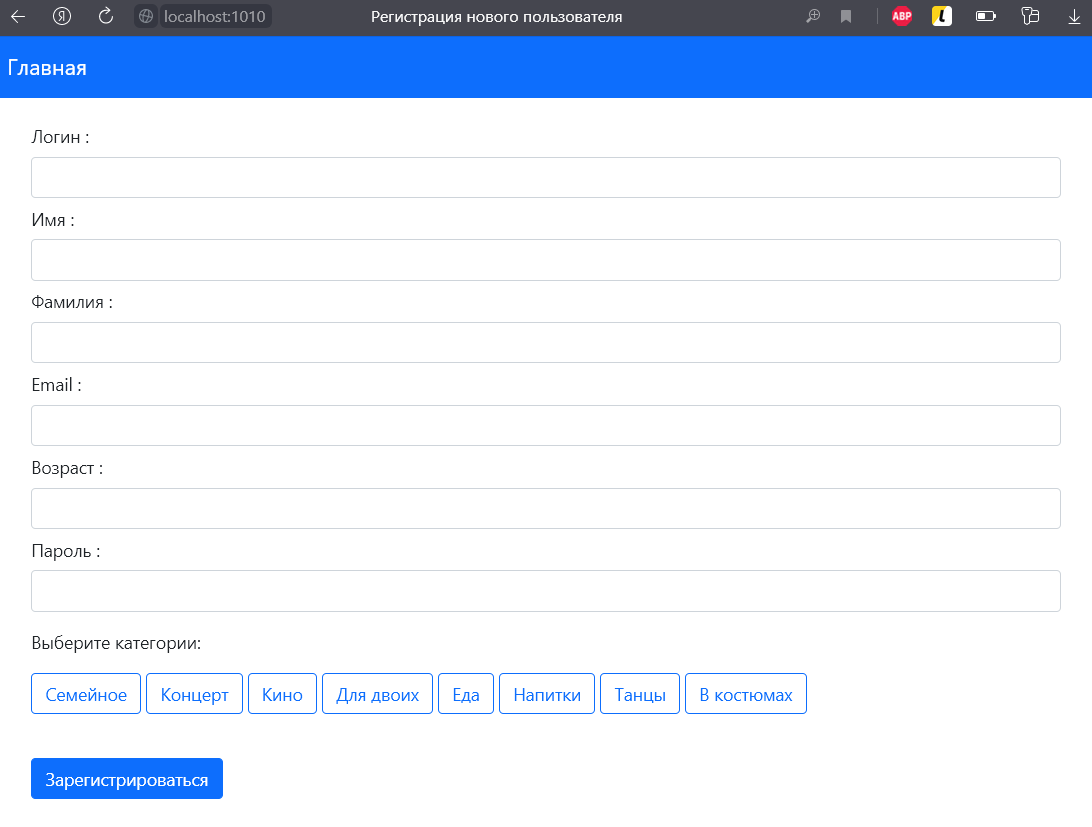


Рисунок 17 – Форма регистрации

После нажатия кнопки регистрации пользователю выводится сообщение о необходимости подтверждения почты, и он оказывается на странице авторизации.

После успешной авторизации пользователь оказывается на главной странице сайта, где ему доступен поиск по пользователям, мероприятиям, местам, а также панель навигации. Главная страница представлена на рисунке 18.

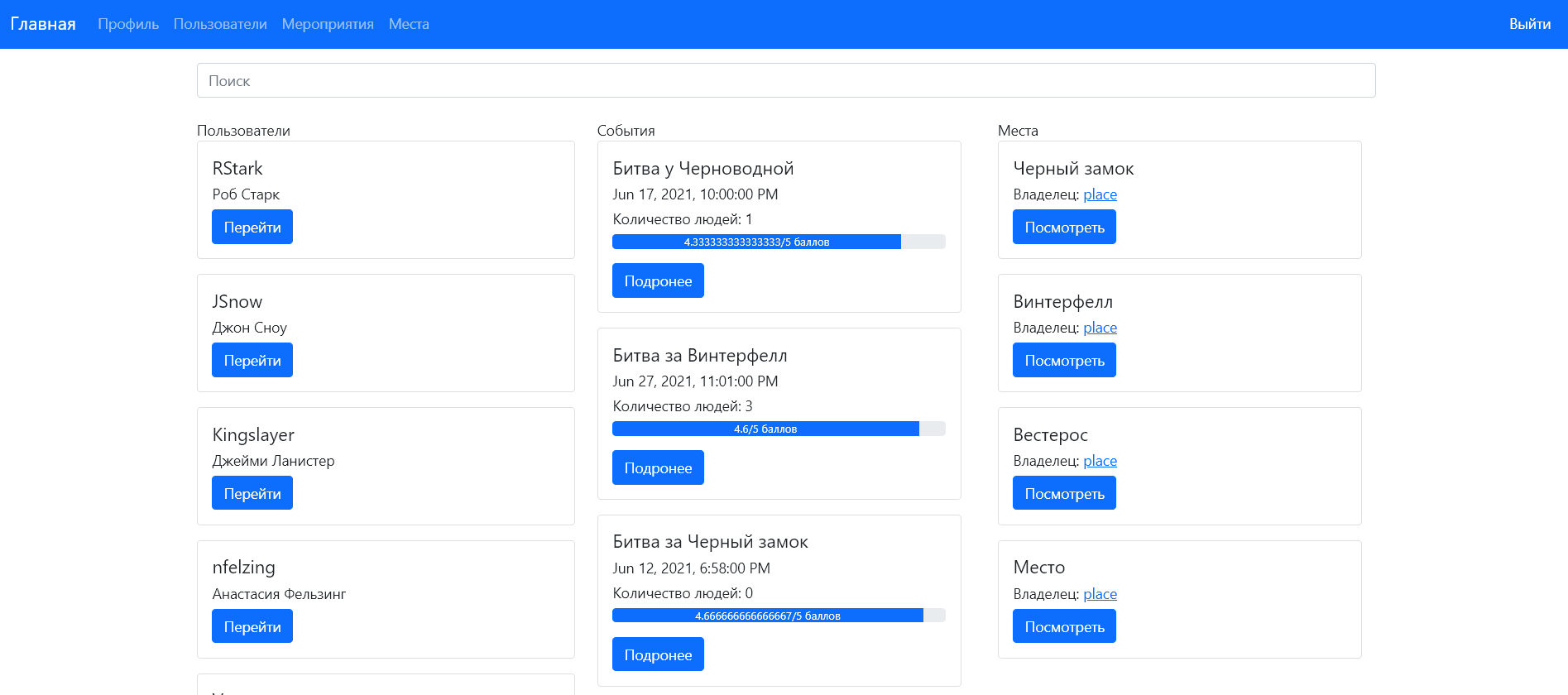


Рисунок 18 – Главная страница

Пользователь может выбрать любой из доступных ему компонентов навигационной панели, чтобы получить информацию о конкретных объектах: мероприятиях, пользователях и местах. При нажатии кнопки «Профиль» пользователь попадает на страницу своего профиля, где может отредактировать информацию о себе, а также посмотреть подписчиков, подписки и мероприятия, на которые он зарегистрировался. Страница профиля пользователя показана на рисунке 19.

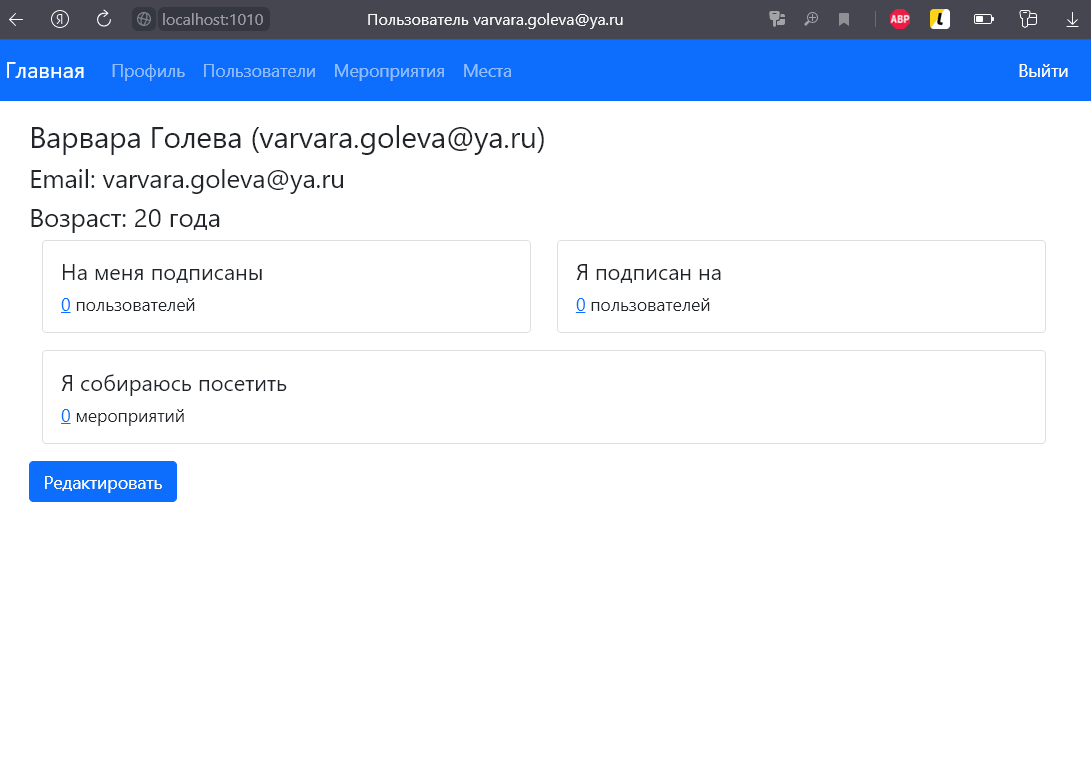


Рисунок 19 – Профиль пользователя

Вкладки «Места» и «Пользователи» содержат информацию о местах проведения мероприятий и пользователях системы соответственно. Вкладки отличаются между собой содержанием карточек с информацией, все типы которых представлены на рисунке 18. Интерфейс вкладки пользователей представлен на рисунке 20.

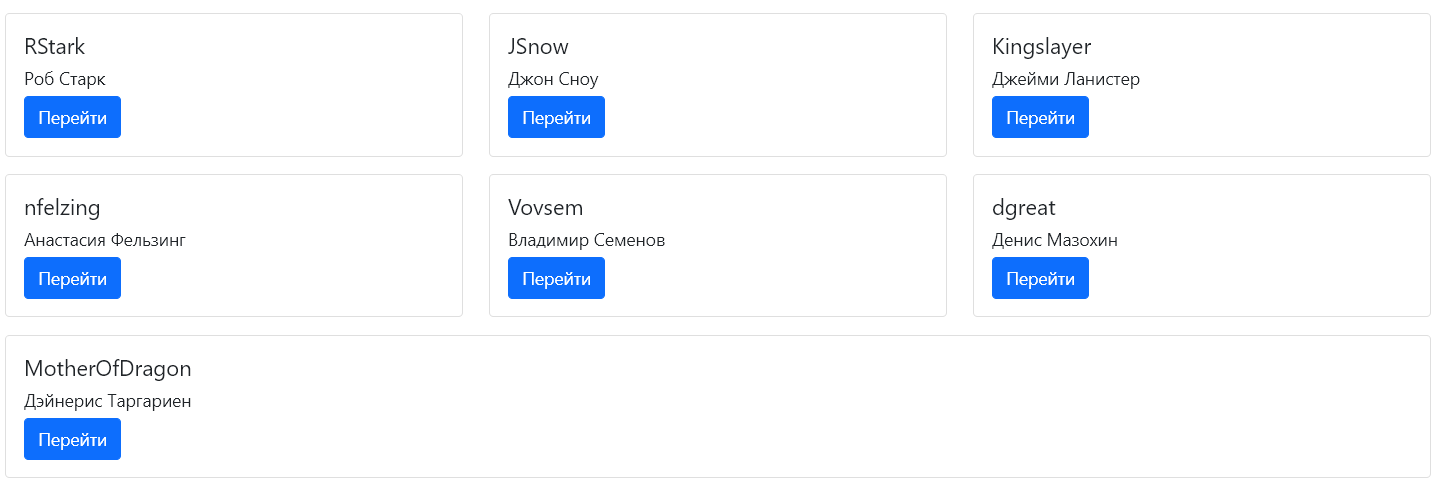


Рисунок 20 – Вкладка «Пользователи»

Для пользователей с ролями организатор мероприятий, владелец места и администратор навигационная панель имеет дополнительные компоненты в соответствии с ролью пользователя. Навигационная панель для организатора мероприятий представлена на рисунке 21.



Рисунок 21 – Навигационная панель для организатора мероприятий

Для демонстрации работы была создана страница, которая отображает результаты работы компонентов рекомендательной системы. На рисунке 22 представлена демонстрация работы системы в условиях недостатка данных для работы коллаборативной фильтрации.

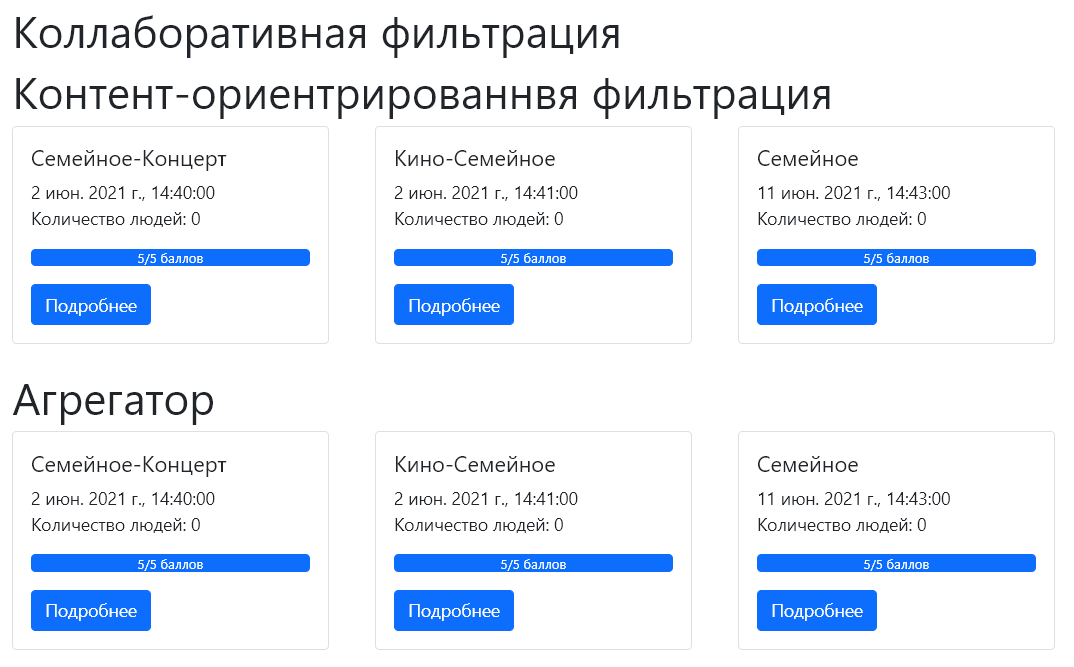


Рисунок 22 – Визуализация результатов работы методов рекомендательных систем

На рисунке видно, что коллаборативная фильтрация не дала никаких результатов из-за недостатка данных для обработки, а контент-ориентированный анализ нашел мероприятия, которые схожи с предпочтениями пользователя. Карточки с мероприятиями отсортированы слева направо от наиболее релевантного события к менее.

Для демонстрации коллаборативной фильтрации и работы агрегатора методов рекомендательной системы пришлось смоделировать ситуацию и внести данные в ИС. На рисунке 23 представлена работа всех методов и агрегатора рекомендательной системы.

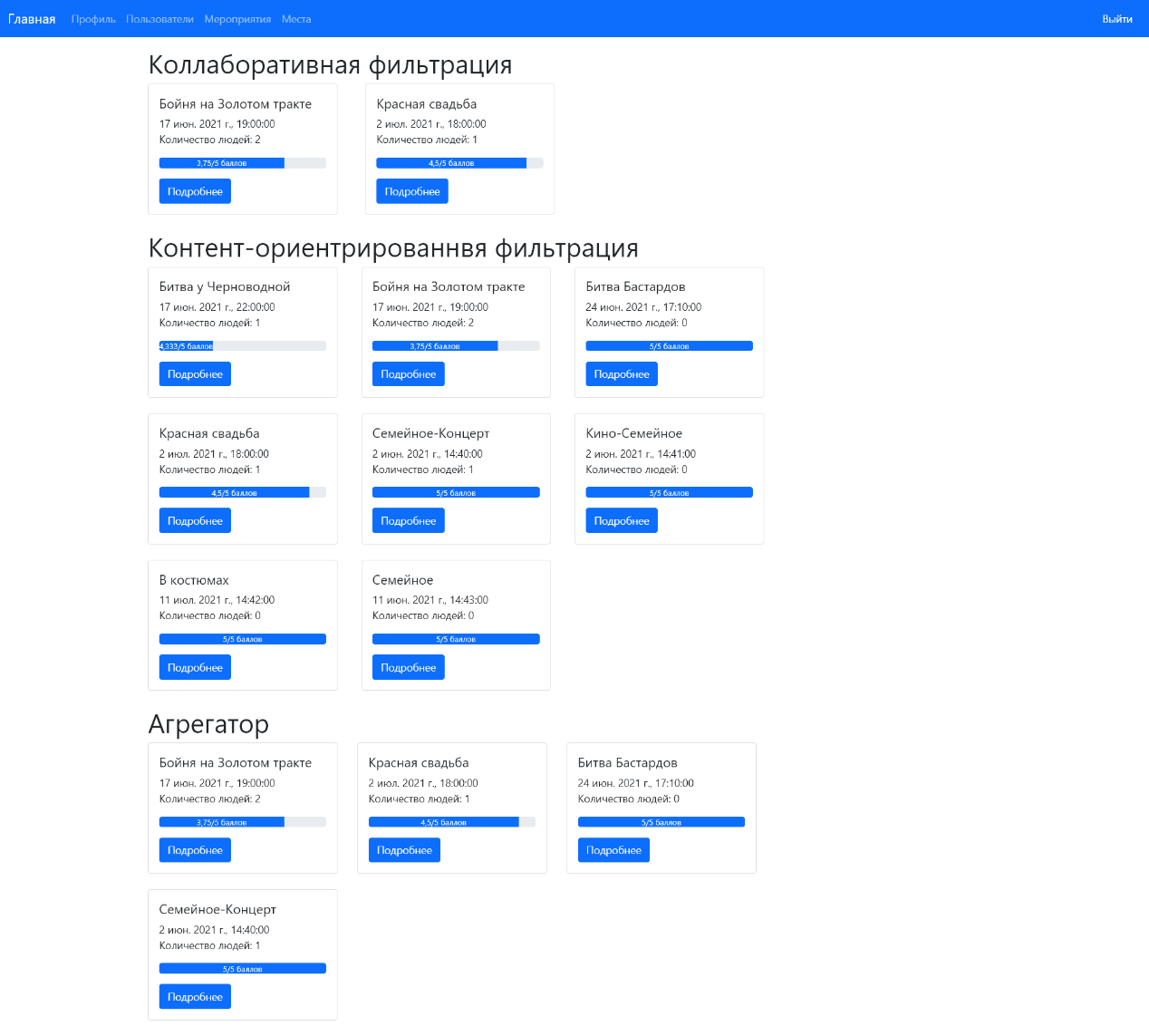


Рисунок 23 – Демонстрация работы компонентов рекомендательной системы

В смоделированной ситуации персонаж Джон Сноу оценил битвы, в которых он участвовал, и остальные герои сериала «Игра Престолов» дали оценки битвам. На рисунке 23 видно, что система рекомендует мероприятия, в которых персонаж Джон Сноу не участвовал, но, предположительно, хотел бы посетить.

* 1. Выводы по технологическому разделу

В данном разделе был обоснован выбор: языка программирования для реализации логики ИС поиска мероприятий, системы управления базой данных и средств для отображения информации пользователю. Также были описаны назначения компонентов разработанного прототипа и представлена демонстрация работы ИС поиска мероприятий.

Заключение

В процессе выполнения работы был создан рабочий прототип информационной системы поиска мероприятий.

Для создания рабочего прототипа ИС были определены требования к ИС, разработана архитектура и структура базы данных системы, был произведен обзор и анализ алгоритмов работы рекомендательных систем, адаптация и реализация этих алгоритмов в ИС поиска мероприятий.

Разработанный программный продукт позволит существенно уменьшить время, затрачиваемое на поиск места проведения досуга, а также позволит организаторам мероприятий повысить уровень осведомленности о предстоящем событии.

# Список источников

1. Рекомендательные системы // Викиконспекты URL: https://neerc.ifmo.ru/wiki/index.php?title=Рекомендательные\_системы (дата обращения: 16.03.2021).
2. Рекомендательные системы: как помочь пользователю найти то, что ему нужно? // vc.ru URL: https://vc.ru/marketing/152926-rekomendatelnye-sistemy-kak-pomoch-polzovatelyu-nayti-to-chto-emu-nuzhno (дата обращения: 16.03.2021).
3. Методические рекомендации по выполнению организационно-экономической части выпускных квалификационных работ [Электронный ресурс]: метод. указания / Т. Ю. Гавриленко, О. В. Григоренко, Е. К. Ткаченко. — М.: РТУ МИРЭА, 2019. — Электрон. опт. диск (ISO)
4. Григоренко О.В., Садовничая И.О., Мыльникова А. Экономика предприятия и управление организацией М.: РУСАЙНС, 2017-235с.
5. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов // Труды ИСП РАН. 2012. №. URL: https://cyberleninka.ru/article/n/sistemy-rekomendatsiy-obzor-sovremennyh-podhodov (дата обращения: 05.06.2021).
6. Many-To-Many Relationship in JPA | Baeldung // Baeldung URL: https://www.baeldung.com/jpa-many-to-many (дата обращения: 23.05.2021).
7. A Guide to CSRF Protection in Spring Security | Baeldung // Baeldung URL: https://www.baeldung.com/spring-security-csrf (дата обращения: 23.05.2021).
8. FreeMarker Common Operations | Baeldung // Baeldung URL: https://www.baeldung.com/freemarker-operations (дата обращения: 23.05.2021).
9. Introduction to Using FreeMarker in Spring MVC | Baeldung // Baeldung URL: https://www.baeldung.com/freemarker-in-spring-mvc-tutorial (дата обращения: 23.05.2021).
10. Spring Boot Reference Documentation // Spring URL: https://docs.spring.io/spring-boot/docs/2.4.5/reference/html/ (дата обращения: 23.05.2021).
11. Сьера К., Бейтс Б. Изучаем Java. 2 изд. O'REILLY, 2016.
12. google/gson: A Java serialization/deserialization library to convert Java Objects into JSON and back // Github URL: https://github.com/google/gson (дата обращения: 13.05.2021).

1. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов // Труды ИСП РАН. 2012. №. URL: https://cyberleninka.ru/article/n/sistemy-rekomendatsiy-obzor-sovremennyh-podhodov (дата обращения: 05.05.2021). [↑](#footnote-ref-1)
2. Гомзин А. Г., Коршунов А. В. Системы рекомендаций: обзор современных подходов // Труды ИСП РАН. 2012. №. URL: https://cyberleninka.ru/article/n/sistemy-rekomendatsiy-obzor-sovremennyh-podhodov (дата обращения: 05.05.2021). [↑](#footnote-ref-2)