Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное

учреждение высшего образования

«Белгородский государственный технологический университет   
им. В.Г. Шухова»

Институт энергетики, информационных технологий и управляющих систем

Кафедра информационных технологий

**КУРСОВОЙ ПРОЕКТ**

**на тему: «Проектирование модуля управления персоналом»**

Выполнил:

студент группы ИТ-42

Курбатова Софья Андреевна

Принял:

ст.преподаватель

Веретенников О.В.

Белгород, 2022

**Содержание**

[Словарь используемых сокращений 3](#_Toc99295219)

[Введение 4](#_Toc99295220)

[1. Предпроектное обследование управления персоналом 5](#_Toc99295221)

[1.1. Постановка задачи проектирования 5](#_Toc99295222)

[1.2. Описание предметной области 5](#_Toc99295223)

[1.3. Цели и задачи программного средства 12](#_Toc99295224)

[1.3.1. Цель программного средства 12](#_Toc99295225)

[1.3.2. Задачи программного средства 13](#_Toc99295226)

[1.4. Анализ существующих аналогов 13](#_Toc99295227)

[1.4.1. Модуль SAP ERP Human Capital Management (SAP ERP HCM) 13](#_Toc99295228)

[1.4.2. Модуль Кадры в Турбо9 14](#_Toc99295229)

[1.4.3. Модуль HR в 1С:ERP 15](#_Toc99295230)

[1.4.4. Модуль Персонал в TURBO ERP: TURBO X 17](#_Toc99295231)

[1.5. Актуальность разработки 18](#_Toc99295232)

[Вывод к разделу 1 19](#_Toc99295233)

[2. Проектирование модуля управления персоналом 20](#_Toc99295234)

[2.1. Проектирование моделей предметной области 20](#_Toc99295235)

[2.2. Проектирование модели данных 27](#_Toc99295236)

[2.3. Выбор средства реализации 29](#_Toc99295237)

[2.3.1. Язык ТУРБО Скрипт (ТБ.Скрипт) 29](#_Toc99295238)

[2.3.1. 1. Язык описания моделей данных - MTL 29](#_Toc99295239)

[2.3.1. 2. Rec-классы 30](#_Toc99295240)

[2.3.1. 3. Frm-классы 30](#_Toc99295241)

[2.3.1. 4. Tab-классы 30](#_Toc99295242)

[2.3.2. Состав программного комплекса 30](#_Toc99295243)

[2.3.3. Серверы и службы серверов 31](#_Toc99295244)

[Вывод к разделу 2 32](#_Toc99295245)

[Библиографический список 33](#_Toc99295246)

[ПРИЛОЖЕНИЯ 34](#_Toc99295247)

# 

# Введение

В настоящее время информация играет значительную роль в жизни каждого человека. Количество информации производимой в мире в единицу времени неуклонно увеличивается, соответственно, искать нужную становится все сложнее. Для этой цели создано множество вспомогательные систем, например, поисковые системы типа Google, которые значительно упрощают этот процесс. Тем не менее, они в большей степени заточены на поиск конкретных фраз и понятий, а значит в большей степени информации в различных небольших статьях, чем каких-либо иных источников информации.

В то же время, значительную роль в жизни людей играют книги(, хоть и не у всех). Процесс увеличения количества информации не обошел и сферу художественных и иных литературных произведений, для которых поиск по фразам в ряде случаев не подходит. Например, если хочется найти книгу похожую на какую-либо из ранее прочитанных. Для этого случая тоже есть ряд систем и сервисов, большинство из которых опирается на мнения других пользователей. Любая информация, получаемая от пользователей, как правило, требует проверки, иначе может быть введена совершенная бессмыслица. Использовать при расчетах непроверенную информацию довольно рискованно, информацию же о том, кому какие книги нравятся проверить невозможно, тут остается только довериться пользователю. Вторым совершенно непредсказуемым фактором являются разные вкусовые предпочтения пользователей. Таким образом, такого рода системы имеют большую погрешность. Другими критериями сравнения книг могут быть авторы, жанры и другие метаданные книги. Однако они характеризуют ее лишь опосредованно, хотя в среднем достаточно точно. Жанр книги всегда можно оценить неточно (ведь это делает человек), особенно когда ей присваивается несколько жанров и нельзя заранее сказать какой из указанных преобладает. Стиль автора может меняться, равно как и тема его книг, что не дает возможности однозначно сказать, что все его книги похожи друг на друга. Вполне возможно такое, что понравится лишь одна книга, в то время как остальные разочаруют.

Система, которая могла бы подбирать литературу без помощи данных предоставляемых другими пользователями, а также без учета метаданных, работая непосредственно с содержанием книги, была бы значительно более точной и не требовала бы периода сбора статистической информации от других пользователей. Кроме того, это вполне естественно оценивать книгу по ее содержанию.

Такого рода система была полезна в многих случаях, например, для книжного магазина. Во-первых, есть возможность дать клиентам инструмент для выбора книг себе, что должно увеличить продажи. Во-вторых, этот же инструмент должен привлечь больше пользователей. В-третьих, менеджеры магазина смогут с большей точностью и качеством подбирать и рекомендовать книги. Например, в случае акции на какие-то из книг, отправить уведомления об этом тем пользователям, которым это может быть интересно.

# Предпроектное исследование информационной системы для подбора литературы

## Постановка задачи проектирования

Целью данной работы является создание системы, позволяющей рекомендовать книги пользователям на основании прочитанных ими ранее других книг, которая может быть использована в книжном магазине, работающем в частности с электронными копиями книг.

Должен быть спроектирован и реализован алгоритм подбора литературы на основании литературы, указанной пользователем как прочитанная, с использованием указанной им оценки. Алгоритм должен использовать данные самой книги, в первую очередь ее контент, для подбора. Алгоритм не должен использовать оценки других пользователей для генерации нового набора рекомендаций.

Должен быть обеспечен интерфейс для указания прочитанных книг их оценки. Оценка книги должна использоваться при подборе рекомендаций.

Интерфейс должен быть предоставлен как клиентам магазина (в том числе и потенциальным), так и сотрудникам для формирования ими информационной рассылки.

Интерфейс должен быть реализован в виде мобильного приложения для операционной системы Android.

## Описание предметной области информационной системы

В данной работе в качестве сферы использования системы рассматривается книжный магазин. Особенностью этой сферы являются, конечно же, книги. Книги пишут люди, авторы этих книг, авторы бывают талантливы и не очень, и, как правило, пишут книги в одном и том же жанре или в смежных жанрах, что позволяет заранее судить о книге по автору, пусть и не всегда верно. Книги классифицируются по жанрам. Жанров огромное множество, кроме того, очень многие книги относятся сразу к нескольким жанрам, по которым и можно понять общую тему произведения.

Чтобы лучше понять заранее, о чем книга и стоит ли ее читать, к ним часто идут аннотации, отзывы других пользователей.

Предполагается, что пользователи будут регистрироваться в системе, чтобы получить доступ к ее функциям и будут в первую очередь работать с электронными копиями книг, поэтому их бумажные аналоги необязательны. Поэтому такая система подойдет как электронным книжным магазинам, так и обычным, которые могут работать с электронными копиями книг. Важно, чтобы эти копии были, так как они необходимы для обработки и извлечения данных, которые в дальнейшем будут использоваться для подбора рекомендаций.

Пользователи могут составлять свои виртуальные библиотеки, на основании которых и будут подбираться рекомендации. Также вполне разумно позволить создавать несколько «сборок», независимых друг от друга, чтобы можно было получить рекомендации по каждой из них отдельно, для получения какого-то контроля над этим процессом.

В самом книжном магазине, безусловно, есть огромное число бизнес процессов. Однако в данном контексте большинство из них не играют роли. В данной работе решается только вопрос с подбором рекомендаций, соответственно, имеет смысл остановиться лишь на этом.

## Перечень задач, подлежащих решению в процессе разработки

В процессе разработки должны быть решены следующие задачи:

1. Реализация алгоритма подбора литературы с учетом указанных требований к нему в виду отдельного модуля.
2. Проектирование и реализация базы данных. База должна содержать информацию о книгах, пользователях, рекомендациях для них и акциях магазина.
3. Реализация серверной части системы, которая будет предоставлять данные из базы клиенту, также выполнять операции, как аутентификация пользователей, изменение данных по запросам авторизованных пользователей.
4. Организация взаимодействия между серверной частью и модулем подбора литературы при помощи посредника в виде базы данных.
5. Реализация клиента, который соответствует указанным выше требованиям, а именно:
   1. предоставляет обычным пользователям системы (предположительно покупателям) интерфейса для поиска книг и получения рекомендаций;
   2. предоставляет пользователям с расширенными права (сотрудникам) интерфейс для подбора рекомендаций другим пользователям.

## Актуальность разработки информационной системы

Система, рекомендующая книги, полезна любому книжному магазину или библиотеке, или же другому агрегатору книг за счет того, что привлекает пользователей. Разумеется, если целью является привлечение пользователей. Чем лучше и качественней рекомендации, тем пользователям будет находить новые книги, тем чаще они будут обращаться к такому сервису и больше доверять ему в плане выбора. Отсутствие же системы рекомендаций ведет к тому, что пользователи должны ориентироваться при поиске на различные метаданные книги, которые, как правило, описывают ее лишь в самом общем смысле. Даже максимально уточненные жанры не дают достаточного представления. Самым точным в описании книги является, пожалуй, автор, но в случае, если несколько авторов пишут в одном ключе, сопоставить их пользователь сможет только при прочтении нескольких книг каждого из них.

Представленные выше сервисы используют различные системы подбора книг. Ряд из них основан на все тех же метаданных книг. Соответственно, они не могут учитывать индивидуальные особенности каждой из книг. К тому же необходимо проверять метаданные книги на соответствие их содержанию.

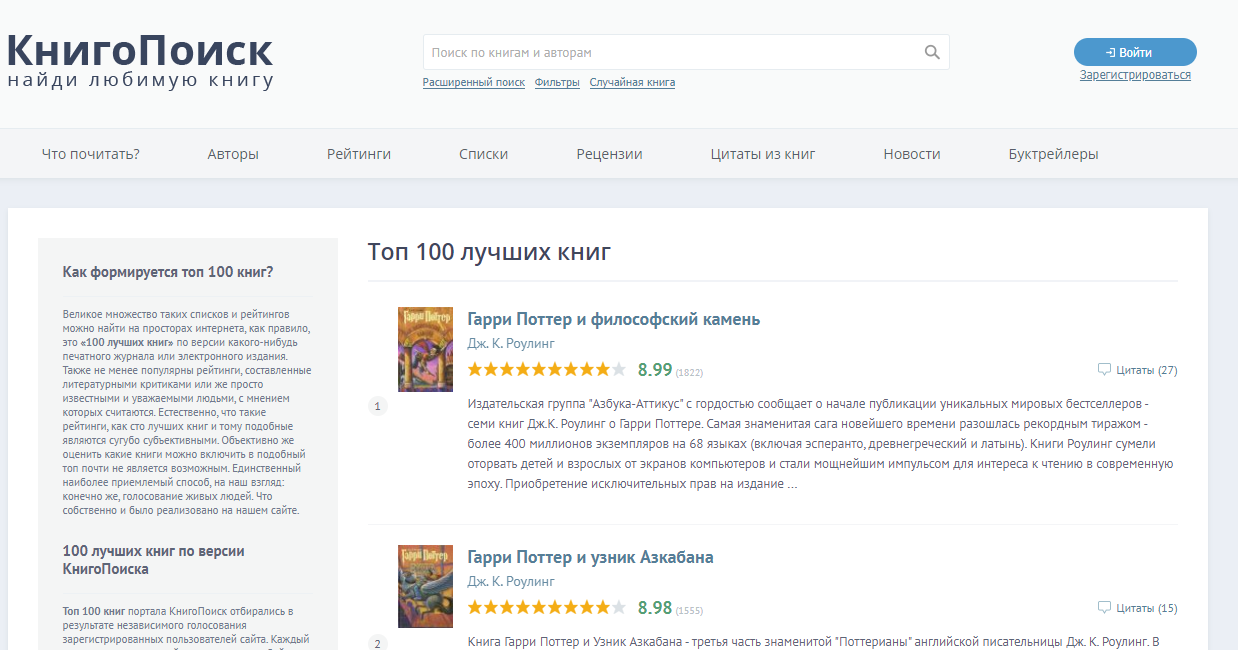
Другие алгоритмы подбора основаны на мнениях других пользователей. Самый простой вариант, это советовать пользователям те книги, которые понравились другим пользователям с теми же интересами. В этом случае, очевидна неточность, связанная с индивидуальностью каждого человека. К тому же возможна намеренная диверсия, путем указания заведомо абсурдных и противоречивых данных, что будет очень тяжело проверить, из-за общего большого объема данных (без которого такая система не будет работать) и все той же индивидуальности каждого человека. Ну и опять же, необходимо заранее иметь информацию о предпочтениях большого количества людей, что иногда сложно. В целом, сказанное выше можно отнести к любому алгоритму, который в качестве исходных данных для подбора использует данные полученные от других пользователей.

В тоже самое время, все необходимые данные уже есть в самом контенте книги, необходимо лишь правильно его проанализировать. Этот подход имеет множество плюсов, например, отсутствие недостатков перечисленных выше подходов, как необходимость модерации данных о предпочтениях пользователей и необходимость иметь набор таких данных для работы. Такая система требует для работы лишь, чтобы был доступен контент книг и модерацию сохраняемых на сервер книг, что и без того должно быть уже реализовано в любом агрегаторе книг, в частности в книжном магазине. Отсутствие зависимостей в виде пользовательских данных, делает систему легко встраиваемой в любую уже существующую.

## Анализ существующих аналогов систем подбора литературы.

### Сайты с популярными книгами

Такого рода рекомендующие системы просто предлагают пользователю набор самых популярных книг, иногда с различными фильтрами. Различаются интерфейсом и набором данных (последнее, впрочем, не всегда).



Сайт КнигоПоиск

Достоинства:

1. просто получить рекомендации;

Недостатки:

1. рекомендации не связаны никак напрямую с предпочтениями пользователя;
2. рекомендации могут включать в себя те книги, которые пользователь уже прочел.

# Выбор алгоритмов для системы подбора литературы

## Описание решаемой задачи

Исходная задача состоит в том, чтобы определить какие из книг будут близки по содержанию и насколько. Имеется в виду не только общий смысл, но и стилистика написания, так как она оказывает огромное влияние на восприятие книги. Близкие в указанном выше смысле книги должны нравиться пользователям примерно одинаково. Разумеется, вопрос дублирования этих книг, например, с незначительными изменениями остается открытым, это больше вопрос модерации добавления новых книг.

Кроме того, должна быть возможность определять близость отдельных книг к другой группе книг, так как рекомендации надо получать для группы прочитанных пользователем книг. Также необходимо учитывать оценку книг.

## Существующие алгоритмы для сравнения текстовых документов

На данный момент существует много различных методов и подходов для анализа текстовых документов на предмет их схожести. Представленные ниже алгоритмы можно разделить на 3 группы: простейшие алгоритмы, которые сравнивают строки (любой текстовый документ можно представить в виде обычной строки), алгоритмы, работающие уже с документами целиком, и алгоритмы, которые обрабатывают текстовые документы для достижения большей точности или скорости работы.

### Алгоритмы, сравнивающие строки

Алгоритмы этого типа рассматривают строку как набор символов.

#### Расстояние Левенштейна

Этот алгоритм вычисляет близость строк как минимальное количество операций вставки, замены или удаления символов. Чем больше таких операций, тем менее близки документы.

#### Алгоритм N-грамм

Этот алгоритм заключается в том, что каждый документ (или же строка) разбивается на n-граммы – последовательности символов длинной n. Затем ищется количество совпавших n-грамм:



S(A,n) это множество n-грамм в документе A. Аналогично с S(B,n).

Далее находится значение близости двух документов по мере включения (2.2) или же по мере Жаккара (2.3):





В отличие от алгоритма расстояния Левенштейна, здесь теряется информация о взаимном расположении символов, тем не менее, из-за фиксированности размера n-грамм (при условии, что n != 1), изменение расположения блоков текста может полностью изменить результат.

#### Поиск самой длинной общей подпоследовательности

Подпоследовательность это совокупность символов, которые появляются в тексте в определенном порядке. Так во фразе «мама мыла раму», «мамы» является подпоследовательностью, а «мыма» нет.



Обобщенная блок-схема получения рекомендаций

Таким образом, для работы системы, необходимо при добавлении книг предварительно сформировать вектора. При самом же запросе рекомендаций от пользователя используются уже готовые вектора.

# Проектирование системы подбора литературы на основе векторной модели

## Проектирование общей архитектуры системы

В целом, информационная система состоит из 3 модулей: модуль, реализующий непосредственно всю логику для подбора рекомендаций, а именно, составление векторов из книг и сам подбор рекомендаций; серверная часть системы, которая реализует API для предоставления доступа к данным базы клиентским приложениям; клиентское приложение в виде мобильного приложения под операционную систему Android. Модуль подбора рекомендаций отделен от серверного приложения для большей гибкости: таким образом его можно реализовать на другом языке, отличном от языка серверного приложения, а также при необходимости впоследствии гораздо легче будет перенести его на другую систему.

Напрямую между собой взаимодействуют, соответственно, клиентское и серверное приложения, а также серверное приложение и модуль подбора рекомендаций. Для взаимодействия клиентского и серверного приложения использует протокол HTTP. Для взаимодействия серверного приложения и модуля подбора рекомендаций было решено использовать базу данных и предоставляемый модулем подбора рекомендаций консольный интерфейс. При помощи базы данных модуль подбора рекомендаций будет возвращать результат работы.



Схема взаимодействия модулей системы

Стрелки обозначают направления обращения за данными модулей.

## Проектирование модуля подбора литературы

Реализовать этот модуль было решено в виде консольного приложения, так как более чем достаточно консольного интерфейса для взаимодействия с ним.

Так как он должен выполнять большое количество вычислительно сложных операций, его надо было реализовать на низкоуровневом языке для достижения наилучшей производительности. Выбор пал на язык Rust, который при тестировании показал наилучший результат [].

Очевидно, что подбор рекомендаций при большом количестве книг будет проходить долго. Серверное приложение не может ждать все это время, ответ пользователю надо отдать сразу. Поэтому результат должен быть сохранен где-нибудь, где серверное приложение впоследствии сможет его прочесть. По сути, есть лишь 2 оптимальных варианта – файл или база данных. Выбор пал на базу данных, потому что СУДБ предоставляет удобный интерфейс для доступа и записи данных. В качестве СУБД было решено использовать Postgresql. Минусом такого выбора является привязка к базе данных, но все имеет свою

# Реализация системы для подбора литературы на основе векторной модели

# Заключение

.

# Список литературы

# Приложение А. Техническое задание