О.С. Самохвалова, К.Д. Уляшев, Д.Е.Попова

Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ПОСТРОЕНИЯ РЕКОМЕНДАЦИЙ СООБЩЕСТВ СОЦИАЛЬНОЙ СЕТИ "ВКОНТАКТЕ"

Целью работы являлась разработка рекомендательной системы для социальной сети ВКонтакте. Предметом рекомендаций были выбраны сообщества в социальной сети ВКонтакте. В качестве основной идеи алгоритма рекомендаций был выбран принцип коллаборативной фильтрации, но в более простом виде: рекомендации составляются на основе знаний о сообществах, в которых состоят друзья пользователя. В результате выполнения работы было разработано приложение с интерфейсом пользователя, позволяющее пользователю получить список рекомендованных сообществ социальной сети ВКонтакте.

The main goal of the paper was to develop a recommendation system for the social network VK. The Communities in the VK were chosen as the subject of recommendations. As the main ideas of the algorithm of recommendations, the principle of collaborative filtering was chosen, but in a simpler form: recommendations are compiled on the basis of knowledge about the communities in which the friends of the user are composed. As a result, we developed an application with a user interface which allows the user to get a list of recommended communities of the social network VK.

Социальные сети, рекомендации, построение рекомендаций

Введение

В настоящее время в связи с ростом количества информации по любым темам встает вопрос о том, как найти интересную или необходимую пользователю информацию в сети интернет, в библиотечных системах или хранилищах данных. В связи с этим активно стали развиваться алгоритмы контекстного поиска, которые позволяют предложить пользователю то, что его скорее всего интересует. Одной из разновидностей контекстного поиска являются рекомендательные системы.

Рекомендательные системы – инструменты автоматической генерации предложений товаров и услуг на основе знаний о потребностях клиентов, стали появляться еще в 2000-х [1], однако настоящий подъем в этой области случился примерно 5-10 лет назад.

На данный момент существует немало успешных самостоятельных рекомендательных сервисов, таких как Last.fm [2], Pandora[3] и IMDb[4]. Помимо этого, все крупные социальные сети используют подобные алгоритмы для рекомендаций контента своим пользователям, например, Instagram[5] предлагает потенциально интересные пользователю профили, а ВКонтакте[6] составляет плейлисты с теми музыкальными композициями, которые, по мнению рекомендательной системы, должны понравиться пользователю.

В социальной сети "ВКонтакте" созданы тысячи сообществ по интересам, популярность которых стремительно набирает обороты, но инструменты сети не предоставляют пользователям возможность получить рекомендации в этой области. Также, на просторах интернета сложно найти удобное и доступное решение этой проблемы. Разработанное нашей командой приложение, реализующее построение рекомендаций на основе интересов друзей, решает эту проблему.

Для решения данной задачи было решено использовать принцип коллаборативной фильтрации[7]. Коллаборативная фильтрация предполагает расчет рекомендаций на основе оценок других пользователей. В данном случае мы учитываем информацию о тех сообществах, в которых состоят друзья пользователя, составляем граф интересов и делаем рекомендацию.

В итоге, было разработано веб-приложение с простым и удобным интерфейсом пользователя, с помощью которого пользователь может получить 2 версии корректного списка рекомендованных

сообществ: на основе информации о подписках его друзей с учетом и без учета количества общих интересов с пользователем.

Задачи данной работы:

- Исследование существующих решений по рекомендациям сообществ пользователям ВКонтакте;
- Реализация приложения, позволяющего пользователям получать список рекомендуемых к просмотру сообществ ВКонтакте.

Обзор предметной области

Перед созданием приложения был произведен поиск аналогичных инструментов для построения рекомендаций сообществ популярных социальных сетей, но все найденные решения рекомендовали всё что угодно, кроме сообществ. В рамках исследования методов решения был проведен анализ приложений, которые так же исследовали рекомендательные алгоритмы для социальных сетей.

Сравнение аналогов

РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ» [8]

В статье рассматривается проблема автоматизации процесса веб-сёрфинга и фильтрации контента, а также, описано проектирование и реализация мультиагентной рекомендательной системы «EZSurf», обеспечивающей анализ интересов и предоставление рекомендаций пользователям социальной сети «ВКонтакте» на основе данных из профиля конкретного пользователя. Спроектированная авторами статьи рекомендательная система, реализующая метод фильтрации содержимого (content-based), использующая профиль пользователя социальной сети «ВКонтакте».

РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ [9]

В этой работе исследована применимость алгоритма Random Walk with Restarts в крупных социальных сетях для рекомендации музыкальных треков. Предложены модификации оригинального алгоритма, позволяющие улучшить качество рекомендаций. Предложен способ составления рекомендации в режиме реального времени по запросу пользователя. Так как оригинальный алгоритм при работе использует структуры данных, которые приобретают колоссальные размеры в реальных социальных сетях, автором был представлен «облегченный» вариант алгоритма, описаны качество и время его работы.

АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM [10]

Статья посвящена разработке рекомендательной системы для пользователей интернет-сервиса Steam, специализирующемся на продаже компьютерных игр. Разрабатываемая система использует гибридную модель фильтраций, состоящую из «collaboration filtration» (коллаборативная фильтрация), а также «content-based» (фильтрация, основанная на контенте). Такие системы направлены на то, чтобы компенсировать недостатки одного подхода внедрением другого.

Критерии сравнения аналогов

Основной идеей работы было создание простого рабочего приложения, чтобы любой пользователь мог воспользоваться данным инструментом. Также, предполагается, что для пользователя имеет значение время, требуемое для решения поставленной задачи. Это говорит о том, что одной из главных целей должна быть быстрота работы алгоритмов рекомендаций. Отсюда вытекает третий критерий: простота алгоритма. Под простотой подразумевается низкая сложность вычислений, производимых данным алгоритмом.

Было проведено сравнение рассмотренных выше решений и алгоритма, полученного в результате выполнения данной работы (Таблица 1).

Таблица 1. Сравнение по критериям

Название статьи	Доступность приложениям	Скорость построения рекомендаций	Простота алгоритма
Построение рекомендаций сообществ социальной сети «Вконтакте»	+	Алгоритм срабатывает за несколько секунды, но загрузка данных занимает несколько минут.	Примитивный алгоритм. Основная сложность заключается в запросах к базе Neo4j.
РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ»	-	-	Сложный мультиагентный механизм
РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ	-	Требует значительных временных ресурсов	Алгоритм несет в себе нетривиальные расчеты
АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM	-	-	Алгоритм несет в себе нетривиальные расчеты

Выводы по итогам сравнения

Рассмотренные рекомендательные системы используют более сложные алгоритмы, из чего можно предположить, что время работы этих алгоритмов значительно выше значений, полученных при тестировании нашего приложения (в одной из статей автор указывает на то, что алгоритм не справляется за приемлемое для пользователя время). Надо отметить, что найденные аналоги не связанны с построением рекомендаций именно сообществ сети ВКонтакте. Это говорит о том, что созданное нами решение потенциально найдет отклик среди пользователей социальной сети "ВКонтакте".

Обзор подходов и выбор метода решения

Существует множество подходов для поиска рекомендаций, самыми популярными из которых являются коллаборативная фильтрация, content - based и гибридный подход.

- Коллаборативная фильтрация - самый популярный вид рекомендательных систем на данный день. Системы коллаборативной фильтрации – это модели, которые пытаются предсказать, насколько вам понравится тот или иной продукт, получая на вход данные о том, как вы и другие пользователи оценивали этот и другие продукты в прошлом. Хотя такие системы достаточно эффективны, их точность сильно зависит от того, на сколько пересекаются оцениваемые объекты между отдельными пользователями. Чем больше и чаще эти пересечения, тем точней будет система. Этот факт ограничивает сферу применения данной модели; если каждый пользователь будет свою уникальную информацию, которую никто другой не видел, коллективные рекомендательные системы будут бессильны. Также, для одним из основных недостатков коллаборативной фильтрации является проблема холодного старта: такая система не может построить рекомендации для нового пользователя, так как еще ничего про него не знает, аналогично и с новыми продуктами - они никому не будут рекомендоваться, пока не накопится информация о них.

- Content – based [7] - это система, которая ищет объекты,похожие на те, что пользователь уже положительно оценил. Описываемый подход чаще всего используется для текстов: документов, сайтов, блогов; или объектов, которые можно описать ключевыми словами. В рекомендательных системах контентного типа полезность рекомендуемых объектов находится из оценок пользователя, данных схожим объектам. Например, для того чтобы посоветовать человеку документ,

контентная система пытается найти сходство между различными объектами, которые ранее получили у пользователя высокую оценку (сходство может заключаться в тематике, авторе и даже в похожих названиях глав в документе).

- Демографический подход [7] - сравнивает характеристики объекта с характеристиками пользователя. Рассматриваться могут простые дискретные свойства: возраст, пол, место проживания, национальность так и более сложные, например, интерес пользователя к определенному объекту. Некоторые из них система может легко определить самостоятельно: страну и даже город, в котором проживает пользователь, можно определить по IP, предпочитаемый язык можно узнать из свойств браузера использую JavaScript [7]. Другие данные можно попросить пользователя указать явно. Несмотря на то, что объем полученной информации о пользователе будет достаточно ограничен, она может помочь принять важные решения, например, кому рекомендовать куклу, а кому видео для взрослых.

- Статистический подход [7] - это системы, которые основываются на статистических данных, собранных с пользователей. Проще говоря, это привычные нам рейтинги самых-самых: самые скачиваемые, самые читаемые, самые популярные и т.п.. Данный подход решает проблему холодного старта.

- Ассоциативный подход [7] - строит рекомендации на основе данных о том, какие объекты используются вместе. Яркий пример применения таких систем — это анализ покупок пользователей. Например, кто-то купил себе новый телефон и, вполне вероятно, ему понадобятся к нему наушники, чехол, зарядка, дополнительная карта памяти и прочие аксессуары. Прочие рекомендательные системы здесь могут быть бессильны, ведь у этих объектов нет никаких общих параметров и сравнить их не представляется возможным, их объединяет лишь то, что они используются вместе.

- Гибридный подход [7] объединяет в себе основные принципы сразу нескольких подходов. Концепция данного подхода заключается в том, что разные системы могут эффективно взаимодействовать, тем самым компенсируя недостатки друг друга.

Многие из алгоритмов, используемых рекомендательными системами, пришли из области машинного обучения, которая занимается алгоритмами для обучения, прогнозирования и принятия решений. Эти алгоритмы реализуют нетривиальные математические вычисления, так как в их задачу входит не только построение списка рекомендаций, но и масштабируемость, анализ поведения пользователей и предсказание их дальнейших действий. В рамках выполнения данной работы решение вышеперечисленных задач избыточно, поэтому использование данных алгоритмов нецелесообразно.

Полученная рекомендательная система основана на использовании нереляционной базы данных Neo4j [11]. Вся логика построения рекомендаций реализована через запросы к базе. Такой подход позволяет максимально облегчить алгоритм и ускорить его. По принципу построения рекомендаций данный способ имеет общие черты с коллаборативной фильтрацией: в качестве основы рекомендаций лежит информация о сообществах друзей пользователя. Говоря другими словами, используется информация о том, как оценили то или иной сообщество другие пользователи социальной сети.

Исследование модели данных

Было произведено исследование аналогов выбранной модели данных для SQL СУБД, в ходе которого было выяснено, что для решения поставленной задачи больше подходит NoSQL модель.

Данный вывод был сделан по следующим причинам:

- NoSQL модель занимает меньше ресурсов памяти по сравнению с SQL моделью. Neo4j и SQL модель не подразумевают коллекций, однако если говорить о коллекции как о наборе однородных данных, то стоить отметить, что

в нашей NoSQL модели различается 2 набора: множество вершин графа и множество ребер, в то время как SQL модель содержит 4 набора: таблицы Person, Friend, Group, Take_Part_In (рис.3).

- SQL-запросы были бы более громоздкими и сложными, чем NoSQL-запросы.

Таблица «Person» содержит информацию о пользователе и его друзьях:

- personId (int) – id пользователя в базе данных;
- name (String) – имя пользователя в социальной сети;
- vkId (int) – id пользователя в социальной сети;
- avatar (String) – ссылка на фотографию пользователя в профиле.

Таблица Group содержит информацию о сообществах, в которых состоят пользователь и\или его друзья:

- groupId (int) – id сообщества в базе данных;
- name (String) – имя сообщества в социальной сети;
- vkId (int) – id сообщества в социальной сети;
- avatar (String) – ссылка на фотографию сообщества в социальной сети.

Таблица Friend описывает связи между пользователем и его друзьями:

- personId (int) - id пользователя в базе данных (друзья);
- userId (int) id пользователя в базе данных (пользователь приложения).

Таблица Takes_Part_In описывает связи между людьми и сообществами:

- personId (int) – id пользователя в базе данных;
- groupId (int) – id сообщества в базе данных.

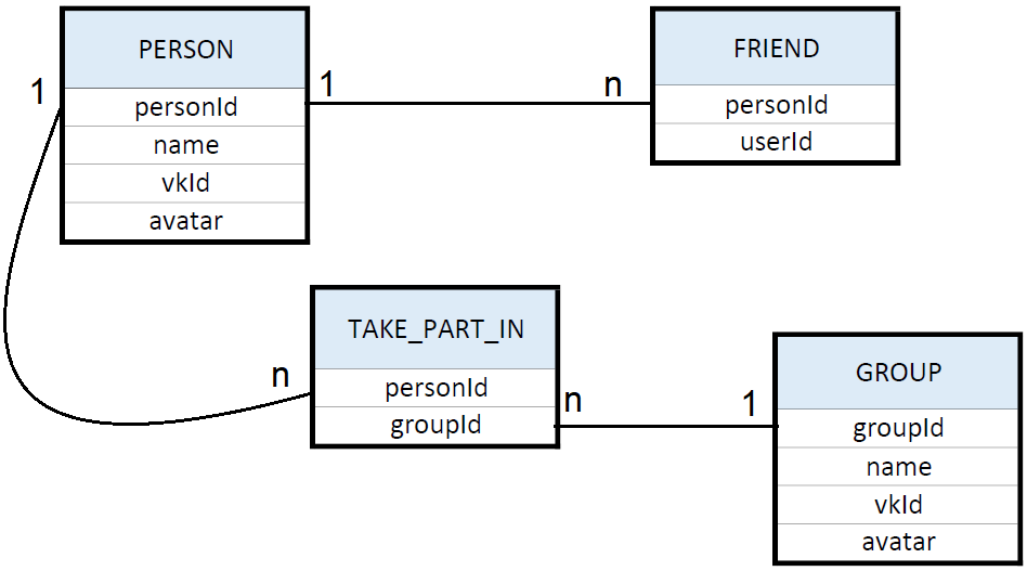

рис. 3 Графическое представление SQL модели данных

Описание решения и используемые технологии

Полученный инструмент представляет собой веб – приложение с интерфейсом пользователя. Разработка осуществлялась с использованием технологии Spring Boot [12]. Интерфейс пользователя реализован с помощью фреймворка Angular 5 [13].

Для работы приложения необходимо загрузить данные. Загрузка данных в программу происходит автоматически после авторизации пользователя с помощью профиля ВКонтакте. Авторизация реализована с помощью средств Spring Security [14] и протокола OAuth [15]. Используя VK API [16], приложение получает информацию об авторизованном пользователе и его друзьях, а также о сообществах, в которых они состоят. Стоит отметить, что также было разработано API для импорта данных вручную. После загрузки данные сохраняются в базу данных Neo4j (рис.1).

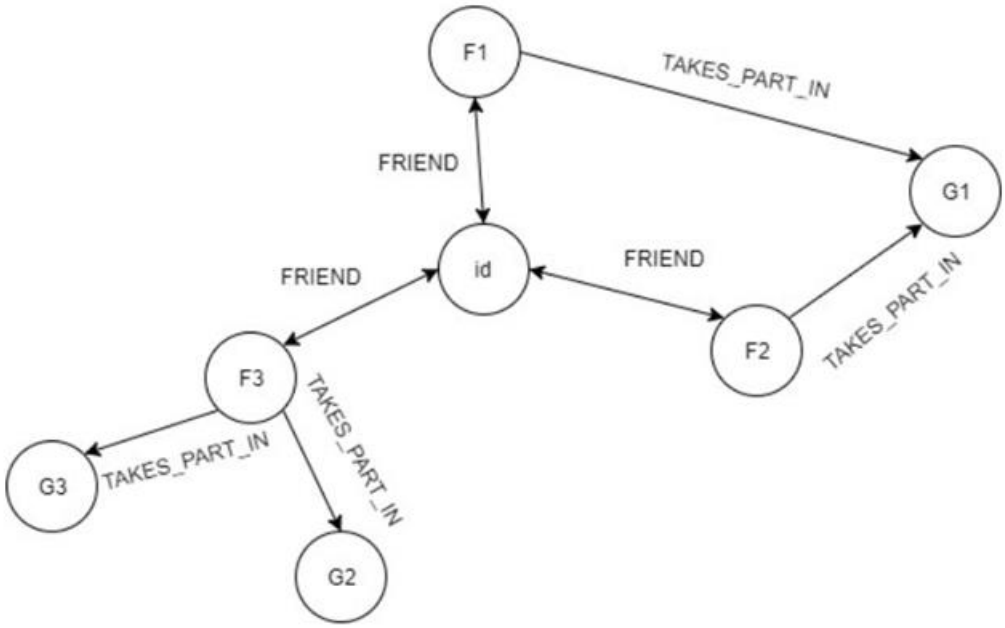

рис. 1 Графическое представление NoSQL модели данных

Вершины графа бывают двух типов: Person и Group. Корневая вершина имеет тип Person и хранит идентификатор пользователя, остальные вершины типа Person хранят идентификаторы его друзей, вершины – листья имеют тип Group и хранят идентификатор сообщества. Вершины имеют следующие свойства:

- id(int) – id, присваиваемый базой данных конкретной вершине;
- name(String) – имя пользователя в социальной сети ВКонтакте;
- vkId(int) – id пользователя в социальной сети ВКонтакте;
- avatar(String) – ссылка на фотографию пользователя из профиля ВКонтакте.

Ребра между корнем и Person – вершинами создают связи FRIEND (связь пользователя с другом). Ребра между Person – вершинами и Group – вершинами создают связи TAKES_PART_IN (связь человека с сообщества). Каждый режим построения алгоритмов реализуется посредством одного запроса к базе данных.

Запросы к базе данных:

- Рекомендации на основе количества друзей:

MATCH (g:Group), (g)-[r]-(p2:Person), (p:Person) WHERE NOT (g)-[:TAKES_PART_IN]-(p) AND p.vkId = {0} RETURN g as group, COLLECT(p2) as friends, COUNT(r) AS rating ORDER BY rating DESC LIMIT {1};

Рекомендации на основе общих интересов с друзьями:

MATCH (g:Group), (g)-[r]-(p3:Person), (p:Person) WHERE NOT (g)-[:TAKES_PART_IN]-(p) and p.vkId = {0} WITH g, p3 MATCH (p1:Person)-[:TAKES_PART_IN]-(g1:Group)-[:TAKES_PART_IN]-(p3) WHERE p1.vkId = {0} " + "WITH g, p3, COUNT(g1) as weight RETURN g as group, COLLECT(p3) as friends, SUM(weight) as rating ORDER BY rating DESC LIMIT {1};

Рекомендательная система работает в двух режимах:

Первый режим загружает из базы данных все Group – вершины, которые не имеют связей с корневой вершиной, и ранжирует их в порядке убывания количества связей типа TAKE_PART_IN. Иными словами, рассматриваются сообщества, в которых состоят друзья пользователя, но не состоит сам пользователь, составляется рейтинг по количеству вхождений в сообщество друзей.

Второй режим сначала анализирует назначает каждой некорневой Person - вершине вес: чем больше количество Group - вершин, с которыми и корневая вершина, и рассматриваемая Person - вершина имеют связи типа TAKE_PART_IN, тем больше вес. После назначения весов повторяются действия, описанные в пункте 1, но ранжирование происходит в порядке убывания суммы всех весов Person - вершин, которые имеют с рассматриваемой Group - вершиной связь типа TAKE_PART_IN. Проще говоря, рассматриваются сообщества, в которых состоят друзья пользователя, но не состоит сам пользователь, составляется рейтинг по количеству вхождений в сообщество друзей, но с учетом информации о том, сколько общих сообществ у пользователя с другом. Чем больше общих сообществ у пользователя и друга, тем больший вес в рейтинге имеют сообщества, в которых состоит этот друг.

После завершения поиска рекомендаций полученная информация выводится на интерфейс пользователя (рис.2) в удобном для восприятия виде: результаты работы алгоритмов визуально отделены друг от друга, карточки, отображающие необходимую информацию, выводятся в порядке релевантности.

рис. 2 Интерфейс пользователя

Главным минусом данного решения является время, требуемое на загрузку данных в базу: при больших объемах данных приложение не сможет отработать за приемлемое для пользователя время, несмотря на то, что сам алгоритм построения рекомендаций занимает всего несколько секунд. Для пользователя с небольшим количеством друзей время работы программы в целом занимает от полутора до трех минут. Под небольшим количеством подразумеваем до 200 человек.

Тестирование приложения проводилось на компьютере ACER Core i5 3210M 2500 Mhz, память 4 Гб DDR3, SSD 500 Гб, Windows 10.

Заключение

В рамках создания рекомендательной системы для социальной сети "ВКонтакте" были рассмотрены наиболее распространённые подходы для поиска рекомендаций. Было выяснено, что большинство рассмотренных алгоритмов используют сложные математические вычисления и решают ряд побочных задач, что говорит о нецелесообразности их использования для решения поставленных задач.

Разработанное приложение [17] успешно решает задачу построения рекомендаций сообществ для пользователей социальной сети "ВКонтакте". Используемый в рекомендательной системе алгоритм показал свою рентабельность, так как при своей простоте реализации он гарантирует минимальные временные затраты. В дальнейшем планируется оптимизировать загрузку данных из социальной сети "ВКонтакте", чтобы максимально сократить время работы программы.

СПИСОК ЛИТЕРАТУРЫ

1. ЛОМАШ ДМИТРИЙ АЛЕКСЕЕВИЧ, ХЛОПИН КОНСТАНТИН ВИКТОРОВИЧ (2013). Реализация рекомендательных систем на основе алгоритмов коллаборативной и контекстной фильтрации. Вестник РГУПС №1/2013 стр. 75-84
2. Last.fm // https://www.last.fm/ru/
3. PANDORA MUSIC // https://www.pandora.com/
4. IMdb// http://www.imdb.com/
5. Instagram // https://www.instagram.com
6. VK // https://vk.com
7. Хабрахабр // habrahabr URL: https://habrahabr.ru/post/58309/ (дата обращения 08.12.2017)
8. АВХАДЕЕВ БУЛАТ РИНАТОВИЧ, ВОРОНОВА ЛИЛИЯ ИВАНОВНА, ОХАПКИНА ЕЛЕНА ПАВЛОВНА (2014). РАЗРАБОТКА РЕКОМЕНДАТЕЛЬНОЙ СИСТЕМЫ НА ОСНОВЕ ДАННЫХ ИЗ ПРОФИЛЯ СОЦИАЛЬНОЙ СЕТИ «ВКОНТАКТЕ», Вестник Нижневартовского Государственного Университета №3 стр. 68-76
9. А.А. Дзюба. (2012)РЕКОМЕНДАЦИИ ТРЕКОВ В СОЦИАЛЬНЫХ СЕТЯХ. Магистерская работа. СПБГУ
10. ГАЖА КОНСТАНТИН ВЛАДИМИРОВИЧ, КАЛМЫКОВ ВЛАДИСЛАВ ВЯЧЕСЛАВОВИЧ, МИТЯНИНА АНАСТАСИЯ ВЛАДИМИРОВНА. (2017) АЛЬТЕРНАТИВНАЯ РЕКОМЕНДАТЕЛЬНАЯ СИСТЕМА ДЛЯ ПОЛЬЗОВАТЕЛЕЙ STEAM. Научная дискуссия №1/2017 стр. 78-93
11. Noe4j // https://neo4j.com/
12. Spring Boot // https://projects.spring.io/spring-boot/
13. Angular 5 // https://angular.io/
14. Spring Security // https://projects.spring.io/spring-security/
15. OAuth // https://oauth.net/
16. VK API // https://vk.com/dev/methods
17. https://github.com/moevm/nosql-2017-social_network_recomendations

O.S. Samokhvalova, K.D. Uliashev, D.E. Popova