

УДК 004.8:004.9:004.75

Шифр специальности ВАК: 2.3.4

Н.А. Мелехов, В.С. Якименко, П.В. Сенченко

Платформа персонализированных рекомендаций медиаконтента на основе коллаборативной фильтрации

Актуальность. Рост объема цифрового медиаконтента усиливает проблему информационной перегрузки и усложняет выбор релевантных объектов для пользователя. **Цель исследования.** Спроектировать платформу персонализированных рекомендаций медиаконтента, ориентированную на фильмы, сериалы, книги, аниме и видеоигры. **Методы.** Используются коллаборативная фильтрация на основе косинусного сходства пользователей, анализ категориальных предпочтений, а также методы проектирования клиент-серверной архитектуры, сбора и нормализации медиаданных. **Научная новизна.** Предложена единая рекомендательная среда для разных типов медиаконтента, в которой рекомендации формируются с учетом как сходства пользовательских оценок, так и явных жанровых и категориальных предпочтений при дефиците данных. **Результаты.** Сформированы архитектурная схема платформы, механизм формирования рекомендаций, контур подготовки медиаданных и схема вычислительного эксперимента для оценки качества рекомендаций и характеристик серверной части. **Практическая значимость.** Предлагаемое решение может использоваться как основа для кроссплатформенного сервиса персонализированного подбора медиаконтента.

Ключевые слова: рекомендательная система, медиаконтент, персонализация, коллаборативная фильтрация, пользовательские предпочтения, мобильное приложение, Flutter, Serverpod, клиент-серверная архитектура.

DOI: 10.21293/1818-0442-2026-29-1-95-101

Введение

В условиях стремительного роста объема цифрового медиаконтента пользователи сталкиваются с проблемой информационной перегрузки. Даже при наличии развитых каталогов и систем поиска выбор релевантного фильма, сериала, книги, аниме или игры требует значительных временных затрат. В этой связи особую значимость приобретают рекомендательные системы, позволяющие автоматизировать подбор контента и учитывать индивидуальные предпочтения пользователя.

Для платформ, работающих с медиаконтентом, особенно важны алгоритмы, способные выявлять устойчивые закономерности в пользовательском поведении. Одним из наиболее распространенных подходов является коллаборативная фильтрация, основанная на анализе сходства интересов пользователей по их оценкам и истории взаимодействия с объектами. Выбор данного подхода в настоящей работе обусловлен тем, что он позволяет формировать рекомендации на основе коллективного пользовательского опыта без необходимости полного семантического анализа каждого объекта контента.

Вопросы построения рекомендательных систем рассматриваются как в обзорных, так и в прикладных работах. Так, в статье А.А. Таромова и Н.Г. Шиловой [1] представлены обзор, анализ и систематизация исследований в области рекомендуемых систем, выполнена классификация публикаций и выделены перспективные направления развития таких решений. В работе М.Ю. Катаева и др. [2] рассматривается структура информационной рекомендательной системы и подчеркивается значимость использования цифровых данных как основы для формирования рекомендаций. В обзорных зарубежных работах показано [3, 4], что коллаборативная фильтрация относится к числу базовых и наиболее распространенных подхо-

дов в рекомендательных системах, а ее применение оправдано при наличии пользовательских оценок и истории взаимодействий. Это обосновывает выбор коллаборативной фильтрации в качестве основы рекомендательной подсистемы в настоящем исследовании.

Вместе с тем применение коллаборативной фильтрации связано с рядом типичных ограничений, прежде всего с недостаточностью данных на начальном этапе работы системы и разреженностью пользовательских оценок. Поэтому при проектировании платформы необходимо учитывать не только архитектурные вопросы, но и особенности организации внутреннего наполнения, а также способы повышения устойчивости рекомендаций при неполноте входной информации.

Целью работы является разработка платформы персонализированных рекомендаций медиаконтента, в которой архитектурные решения подчинены задаче формирования релевантных рекомендаций на основе пользовательских оценок и предпочтений.

Постановка задачи

Рост объемов доступного медиаконтента и сопутствующая ему проблема информационной перегрузки пользователей обуславливают актуальность создания интеллектуальной рекомендательной системы. Проект нацелен на разработку кроссплатформенной архитектуры, которая объединит клиентские приложения и серверные компоненты в комплексную систему персональных рекомендаций для медиа.

Общая схема процесса сбора данных и поддержки медиатеки представлена на рис. 1.

Для реализации проекта необходимо решить следующие задачи:

1. Разработать модель представления пользовательских предпочтений на основе оценок, истории взаимодействий и предпочтительных категорий медиаконтента.

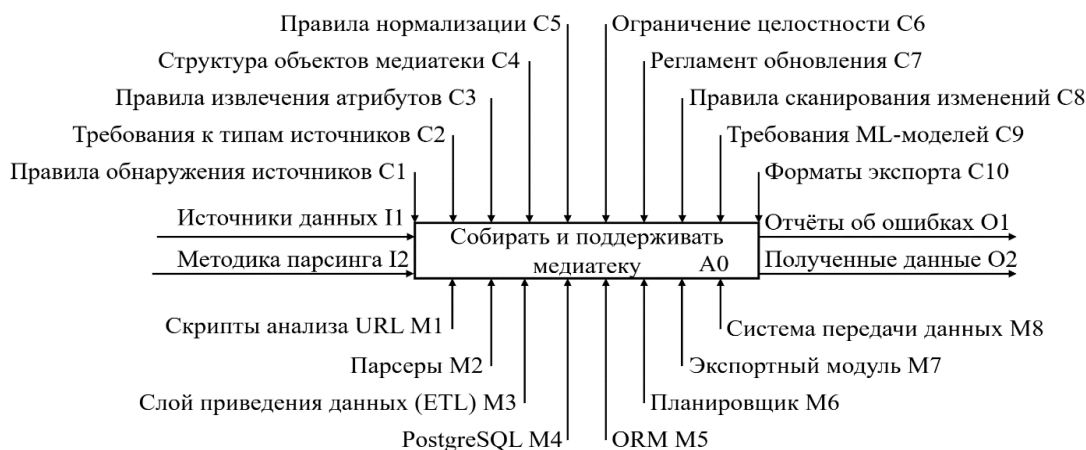


Рис. 1. Сбор и поддержка медиатеки

2. Реализовать рекомендательную подсистему на основе коллаборативной фильтрации для поиска пользователей со сходными интересами и формирования персонализированных рекомендаций.

3. Определить способы снижения влияния проблемы холодного старта и разреженности пользовательских данных.

4. Организовать сбор, очистку и нормализацию информации о медиаконтенте, используемой в рекомендательной подсистеме.

5. Разработать клиент-серверную платформу, обеспечивающую хранение данных, обработку пользовательских действий и выдачу персонализированных рекомендаций.

6. Провести сравнение разрабатываемой платформы с существующими решениями по критериям, значимым для рекомендательных систем.

В основе работы платформы лежит анализ пользовательских оценок, истории взаимодействий с медиаконтентом и предпочтений по категориям. Ключевым элементом системы является алгоритм коллаборативной фильтрации, который позволяет выявлять пользователей со сходными интересами и использовать их оценки для формирования персонализированных рекомендаций. Такой подход особенно целесообразен в медиасреде, где пользовательские реакции на контент выступают основным источником информации о предпочтениях.

Одним из важнейших компонентов инфраструктуры является административная панель, которая обеспечивает целостность данных и устойчивость системы к некорректным действиям. С её помощью выполняется наполнение медиакаталога, редактирование метаданных, управление пользовательскими правами и контроль контента.

Центром взаимодействия с системой для конечного пользователя является персональный профиль. Это динамически формируемое пространство отражает историю просмотров, рейтинги, созданные подборки и предпочтения. Возможность кастомизации, ведения личных коллекций и рефлексии над собственной активностью не только увеличивает вовлеченность, но и формирует осознанное взаимодей-

ствие с платформой, превращая её из инструмента в персонализированную медиасреду.

Для поддержания базы медиа в актуальном состоянии в системе используется автоматизированный контур сбора и обработки данных. Его основу составляют Python-скрипты [7], которые парсят авторитетные открытые источники – специализированные базы по кино, компьютерным играм, литературе и аниме. В основе сценариев парсинга лежит сочетание правил извлечения и текстовой обработки данных. Правила извлечения используются для выделения из веб-страниц структурированных полей, таких как название, жанр, рейтинг, дата выпуска и описание. Элементы обработки естественного языка применяются на этапе нормализации текстовых данных: удаления служебных символов, унификации жанровых обозначений, приведения названий к единому виду и устранения дублирующихся записей. Такой подход позволяет преобразовать неструктурированные данные в формат, пригодный для загрузки в основную базу и последующего использования в рекомендательной подсистеме. Подобная организация процесса устраняет необходимость ручного ввода, обеспечивает своевременное обновление базы и значительно повышает эксплуатационную надёжность системы.

На первом этапе система определяет и сортирует источники информации, с которыми предстоит работать. Для минимизации простоев задействуются механизмы обработки сбоев: если возникают проблемы с сетью или источник временно недоступен, система автоматически повторяет попытки соединения.

После того как сырые данные успешно загружены, наступает их обработка. Информация проходит несколько стадий: очищается от лишнего, приводится к единому стандарту и проверяется на достоверность. Данный процесс необходим для того, чтобы данные соответствовали требованиям системы и были готовы к помещению в основную базу. Как именно выстроена эта последовательность действий, наглядно демонстрирует рис. 2.

Далее приведено описание основных функциональных блоков, входящих в контекст модулей системы.

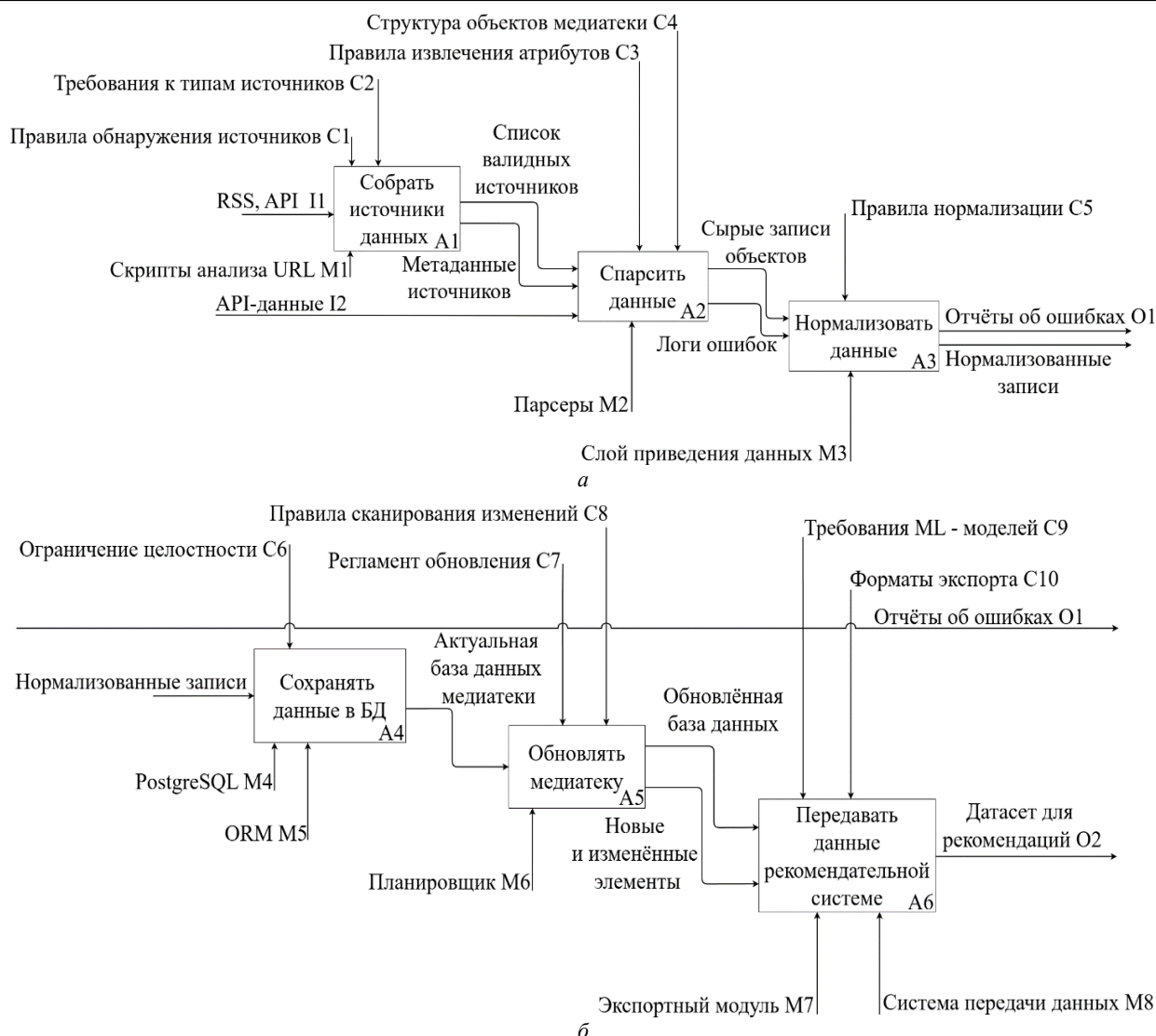


Рис. 2. Детализированная схема работы модуля парсинга (а – начало, б – конец)

1. Функциональный блок «A1. Собрать источники данных». Этот блок отвечает за получение актуальных ссылок и метаданных из открытых онлайн-баз фильмов, игр, книг и аниме. Система формирует список URL-адресов, с которых в дальнейшем извлекаются сведения о медиаконтенте.

2. Функциональный блок «A2. Спарсить данные». На этом этапе выполняется автоматический сбор информации с веб-страниц. Python-скрипты анализируют HTML-разметку, извлекая ключевые атрибуты – название, жанр, рейтинг, дату выпуска и описание.

3. Функциональный блок «A3. Нормализовать данные». После парсинга полученные данные проходят обработку: удаляются дубликаты, устраняются ошибки форматирования, приводятся к единой структуре и кодировке. Это обеспечивает целостность информации в базе.

4. Функциональный блок «A4. Сохранять данные в БД». Структурированные записи сохраняются в базе данных PostgreSQL [8]. Для удобства используется ORM-модель, что упрощает взаимодействие

и поддерживает согласованность данных при обновлениях.

5. Функциональный блок «A5. Обновлять медиатеку». Этот блок отвечает за периодическую проверку источников и автоматическое добавление новых элементов. Таким образом, медиатека всегда содержит актуальные данные без необходимости ручного редактирования.

6. Функциональный блок «A6. Передавать данные рекомендательной системе». На последнем этапе данные передаются в рекомендательную подсистему. Информация о жанрах, рейтингах и пользовательских отзывах используется для построения персональных рекомендаций и дальнейшего анализа интересов пользователей.

Механизм формирования рекомендаций

Можно выделить следующие шаги алгоритма:

После авторизации в приложении пользователь получает доступ к коллекции фильмов, сериалов, книг, аниме и видеоигр. Для сокращения времени выбора система формирует персонализированные подборки. Основу рекомендательной подсистемы со-

ставляет коллаборативная фильтрация (Collaborative Filtering), базирующаяся на принципе сходства пользовательских предпочтений [3–6].

На начальном этапе, когда у системы ещё недостаточно данных о предпочтениях нового пользователя, рекомендации формируются на основе наиболее популярного и высоко оценённого контента на платформе.

1. На первом этапе система анализирует выставленные пользователем оценки с целью формирования детального профиля пользовательских интересов.

2. На втором этапе система осуществляет поиск пользователей, имеющих сходные профили, вычисляя степень их соответствия на основании вычисления специальной математической меры – косинусного сходства (1).

$$\text{similarity}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^n A_i \cdot B_i}{\sqrt{\sum_{i=1}^n A_i^2} \times \sqrt{\sum_{i=1}^n B_i^2}}, \quad (1)$$

где $\text{similarity}(\mathbf{A}, \mathbf{B})$ – мера косинусного сходства между пользователями A и B , значение в диапазоне от -1 до 1 , где 1 означает полное совпадение предпочтений; $\mathbf{A}$ – вектор оценок текущего пользователя; $\mathbf{B}$ – вектор оценок другого пользователя из системы; i – конкретный объект контента (фильм, книга и т.д.); n – общее количество объектов, которые оценили оба пользователя.

Система анализирует найденных пользователей со схожими предпочтениями и отбирает контент, который они высоко оценили, но который ещё не знаком нашему пользователю.

Параллельно приложение учитывает и конкретные интересы пользователя, анализируя предпочитаемые им жанры и категории. Этот подход, основанный на содержании контента, создает гибридную систему: она не просто опирается на мнение похожих людей, но и тонко настраивается под уникальные предпочтения каждого.

Такой подход позволяет частично снизить влияние проблемы холодного старта и недостатка пересечений по оценкам, поскольку при нехватке пользовательских данных рекомендации дополняются информацией о предпочтительных жанрах и категориях контента.

Контуры алгоритмического обеспечения

Алгоритмическое обеспечение платформы включает два взаимосвязанных контура. Первый контур отвечает за подготовку данных: сбор сведений из открытых источников, очистку, нормализацию, устранение дубликатов и актуализацию медиатеки. Его результатом является согласованный каталог объектов, пригодный для последующей аналитической обработки.

Второй контур связан с формированием рекомендаций. На его вход поступают пользовательские оценки, история взаимодействий и категориальные предпочтения, а на выходе формируется ранжирован-

ный список объектов. При этом вычисление сходства пользователей по формуле (1), описанное в предыдущем разделе, выступает центральным шагом рекомендательного контура, а учёт жанров и типов контента используется как механизм повышения устойчивости рекомендаций при разреженности данных.

Тем самым алгоритмическое обеспечение охватывает полный цикл обработки данных – от подготовки медиатеки до выдачи рекомендаций, а рекомендательный механизм из предыдущего раздела является его ключевым, но не единственным компонентом.

Сравнение с существующими решениями

Сравнение разрабатываемой платформы целесообразно проводить как с косвенными, так и с частично прямыми аналогами. На основе изучения сервисов была составлена сравнительная таблица. В нее включены «КиноПоиск», «IMDb», «Amazon Prime Video» и TasteDive. Первые три сервиса отражают различные модели медиасервисов с рекомендациями, тогда как TasteDive включён в сравнение как тематически близкий сервис, позволяющий подбирать фильмы, сериалы, книги, музыку и игры в рамках единого интерфейса.

Критерии оценивания:

«+» – 1 балл, система полностью удовлетворяет данному требованию;

«+/-» – 0,5 балла, система частично удовлетворяет данному требованию;

«-» – 0 баллов, система не удовлетворяет данному требованию.

Результаты сравнения систем

Критерий	КиноПоиск	IMDb	Amazon Prime Video	TasteDive
Географическая доступность	+	–	–	+/-
Русский язык интерфейса	+	–	+/-	–
Наличие отдельного раздела рекомендаций	+/-	+	–	+/-
Автоматическая персонализированная рекомендация	+/-	+	+/-	–
Возможность ручной астройки предпочтений	+	+	+	+/-
Навигация (интерфейс)	+	+	+/-	+/-
Логическая структура меню	+	+/-	+	+/-
Отсутствие рекламы в бесплатной версии	–	+/-	–	+/-
Наличие бесплатного доступа	+	+	–	+
Гибкость подписки	+	–	+/-	–
Результат	8,0	6,0	4,0	4,0

Состав критериев сравнения сформирован с учётом целей разрабатываемой системы. Включены общие пользовательские характеристики и признаки, непосредственно связанные с рекомендательной подсистемой: географическая доступность; русскоязычная локализация интерфейса; отдельный раздел рекомендаций; автоматическая персонализированная рекомендация и возможность ручной настройки предпочтений.

Такой подход позволяет сопоставить аналоги не только как медиасервисы в целом, но и как решения, поддерживающие персонализацию пользовательского опыта.

Результаты сравнения показывают, что сильные стороны частичных аналогов связаны либо с развитой локализацией и адаптацией под российскую аудиторию («КиноПоиск»), либо с масштабом базы медиаданных и механизмами пользовательских оценок («IMDb»), либо с интеграцией рекомендаций в потоковый сервис («Amazon Prime Video»). Включение в сравнение TasteDive позволяет дополнительно сопоставить предлагаемое решение с сервисом, поддерживающим подбор сходного контента разных типов в единой среде.

Вместе с тем TasteDive не следует рассматривать как полный прямой аналог предлагаемой платформы. Его рекомендации в большей степени строятся как подбор похожих объектов по выбранной позиции, а не как развитая персонализированная система, опирающаяся на целостный пользовательский профиль, историю взаимодействий и совокупность оценок. Поэтому TasteDive корректно трактовать как тематически близкий сервис, демонстрирующий востребованность единой рекомендательной среды для разных типов медиаконтента, но не реализующий в полной мере ту степень персонализации, которая закладывается в проектируемую платформу.

Принцип работы платформы

Архитектура платформы построена по клиент-серверной схеме и служит основой для функционирования рекомендательной подсистемы. Клиентская часть обеспечивает взаимодействие пользователя с медиатекой, отображение рекомендаций и фиксацию действий пользователя. Серверная часть отвечает за хранение данных, обработку пользовательских запросов и запуск алгоритма формирования рекомендаций.

Такая организация позволяет объединить сбор пользовательских сигналов, актуализацию медиакаталога и персонализированную выдачу контента в рамках единой системы. Следовательно, архитектурные решения в работе рассматриваются как средство обеспечения рекомендательного механизма, а не как самостоятельная цель разработки. Схема архитектуры системы представлена на рис. 3.

С технологической точки зрения клиентская часть платформы будет реализована на Flutter [9, 12–15], что обеспечит кроссплатформенность и адаптацию интерфейса под мобильные устройства. Серверная часть проектируется на основе Serverpod [10] – backend-фреймворка для экосистемы Flutter/Dart,

позволяющего реализовать API, бизнес-логику и взаимодействие с базой данных в едином технологическом стеке.

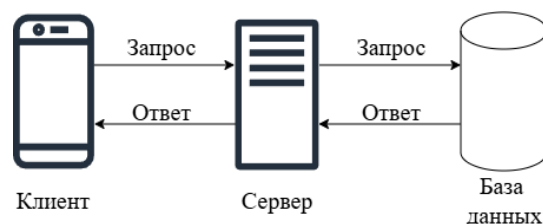


Рис. 3. Архитектура системы

Интерфейс приложения разработан в соответствии с принципами Material Design, обеспечивая визуальную консистентность, удобную навигацию и адаптивность под различные размеры экранов. Комбинация Flutter и Serverpod позволяет реализовать устойчивую клиент-серверную платформу, в которой рекомендательная подсистема интегрирована в общий цикл работы с медиаконтентом.

Схема вычислительного эксперимента

Для последующей верификации предложенного решения целесообразно провести вычислительный эксперимент в двух взаимосвязанных направлениях. Во-первых, должна быть проверена собственно рекомендательная подсистема на исторических пользовательских данных. Для этого массив оценок и взаимодействий разделяется на обучающую и тестовую выборки, после чего качество выдачи сопоставляется с базовыми подходами: рекомендацией по популярности и вариантом без учёта категориальных предпочтений. Основными метриками предлагается использовать Precision@K , Recall@K , NDCG@K и покрытие рекомендаций; для сценариев холодного старта дополнительно анализируется качество рекомендаций для пользователей с малым числом оценок [5].

Во-вторых, должен быть оценён архитектурный контур платформы: среднее время формирования рекомендаций, время ответа API, устойчивость работы при росте числа запросов и полнота обновления медиатеки после очередного цикла парсинга. Целевым состоянием эксперимента является подтверждение того, что гибридизация коллаборативной фильтрации с учётом категориальных предпочтений повышает качество топ-N-рекомендаций по сравнению с базовыми схемами, а архитектура платформы обеспечивает приемлемое время отклика и воспроизводимое обновление данных.

Заключение

В результате работы спроектированы архитектурная схема платформы персонализированных рекомендаций медиаконтента и механизм формирования рекомендаций на основе коллаборативной фильтрации. Предложенное решение ориентировано на использование пользовательских оценок, истории взаимодействий и явных предпочтений для формирования персонализированных подборок.

В ходе исследования показано, что применение коллаборативной фильтрации является обоснованным для рассматриваемой предметной области,

поскольку позволяет использовать коллективный опыт пользователей при формировании рекомендаций. Дополнительный учёт категориальных предпочтений пользователя повышает устойчивость работы системы в условиях недостатка данных на начальном этапе.

На текущем этапе работа носит проектно-алгоритмический характер: сформулированы структура платформы, логика подготовки данных и схема рекомендательной подсистемы. Для количественного подтверждения эффективности предложенного решения в дальнейшем должен быть проведён вычислительный эксперимент по схеме, изложенной в предыдущем разделе, с оценкой как качества рекомендаций, так и эксплуатационных характеристик архитектуры.

Перспективы дальнейшего развития связаны с расширением набора пользовательских сигналов, совершенствованием алгоритма ранжирования, реализацией полнофункционального программного прототипа и экспериментальной проверкой качества рекомендаций на реальных сценариях использования.

Работа выполнена в рамках государственного задания Минобрнауки России; проект FEWM-2026-0011.

Литература

1. Тарамов А.А. Рекомендующие системы для информационной поддержки водителя: анализ состояния исследований / А.А. Тарамов, Н.Г. Шилов // Доклады ТУСУР. – 2018. – Т. 21, № 2. – С. 68–74. DOI: 10.21293/1818-0442-2018-21-2-68-74.
2. Катаев М.Ю. Структура информационной рекомендательной системы поддержки принятия решений при оказании услуг государственным учреждением / М.Ю. Катаев, Н.В. Лосева, Л.А. Булышева // Доклады ТУСУР. – 2018. – Т. 21, № 2. – С. 83–87. DOI: 10.21293/1818-0442-2018-21-2-83-87.
3. Roy D. A systematic review and research perspective on recommender systems / D. Roy, M. Dutta // Journal of Big Data. – 2022. – Vol. 9. – Art. 59. DOI: 10.1186/s40537-022-00592-5.
4. Afoudi Y. Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network / Y. Afoudi, M. Lazaar, M. Al Achhab // Simulation Modelling Practice and Theory. – 2021. – Vol. 113. – Art. 102375. DOI: 10.1016/j.simpat.2021.102375.
5. Zangerle E. Evaluating recommender systems: Survey and framework / E. Zangerle, C. Bauer // ACM Computing Surveys. – 2022. – Vol. 55, No. 8. – Art. 170. – P. 1–38. DOI: 10.1145/3556536.
6. Panteli A. Addressing the cold-start problem in recommender systems based on frequent patterns / A. Panteli, B. Boutsinas // Algorithms. – 2023. – Vol. 16, No. 4. – Art. 182. DOI: 10.3390/a16040182.
7. Python documentation [Электронный ресурс]. – URL: <https://docs.python.org/> (дата обращения: 16.04.2026).
8. PostgreSQL: Documentation [Электронный ресурс]. – URL: <https://www.postgresql.org/docs/> (дата обращения: 16.04.2026).
9. Flutter documentation [Электронный ресурс]. – URL: <https://docs.flutter.dev/> (дата обращения: 16.04.2026).
10. Serverpod documentation [Электронный ресурс]. – URL: <https://docs.serverpod.dev/> (дата обращения: 6.04.2026).
11. Lercher A. Microservice API evolution in practice: A study on strategies and challenges / A. Lercher, J. Glock, C. Macho, M. Pinzger // Journal of Systems and Software. – 2024. – Vol. 215. – Art. 112110. DOI: 10.1016/j.jss.2024.112110.
12. Material Design 3 [Электронный ресурс]. – URL: <https://m3.material.io/> (дата обращения: 17.04.2026).
13. Чуриков Е.А. Сравнение современных средств разработки мобильных приложений / Е.А. Чуриков, Т.В. Зудилова, И.В. Ананченко, Н.А. Осипов, С.Е. Иванов, И.С. Осетрова // Современные наукоемкие технологии. – 2022. – № 12. – С. 82–87. DOI: 10.17513/snt.39441.
14. Souha A. Comparative analysis of mobile application frameworks: A developer's guide for choosing the right tool / A. Souha, L. Benaddi, C. Ouaddi, A. Jakimi // Procedia Computer Science. – 2024. – Vol. 236. – P. 597–604.
15. Oliveira W. Analyzing the resource usage overhead of mobile app development frameworks / W. Oliveira, B. Moraes, F. Castor, J.P. Fernandes // EASE '23: Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering. – University of Oulu, Finland: ACM, 2023. – P. 152–161. DOI: 10.1145/3593434.3593487.

Мелехов Никита Александрович

Магистрант каф. автоматизации обработки информации (АОИ) Томского государственного университета систем управления и радиоэлектроники (ТУСУР)
Вершинина ул., 74, г. Томск, Россия, 634045
Тел.: +7-999-440-08-30
Эл. почта: melekhov206@gmail.com

Якименко Вячеслав Сергеевич

Магистрант каф. АОИ ТУСУРа
Вершинина ул., 74, г. Томск, Россия, 634045
Тел.: +7-923-432-07-49
Эл. почта: slava.yakimenko.2015@gmail.com

Сенченко Павел Васильевич

Канд. техн. наук, доцент, декан фак-та систем управления ТУСУРа
Вершинина ул., 74, г. Томск, Россия, 634045
Тел.: +7 (382-2) 70-15-46
Эл. почта: pvs@tusur.ru

Поступила в редакцию: 26.11.2025.

Принята к публикации: 28.04.2026.

Melekhov N.A., Yakimenko V.S., Senchenko P.V.

Platform for Personalized Media Content Recommendations Based on Collaborative Filtering

Relevance. The growing volume of digital media content exacerbates the problem of information overload and complicates the selection of relevant items for the user. **The purpose of the study** is to design a personalized media content recommendation platform focused on movies, TV series, books, anime, and video games. **Methods.** The study employs collaborative filtering based on cosine similarity of users, categorical preference analysis, as well as methods for designing client-server architecture, collecting and normalizing media data. **Novelty.** A unified recommendation environment for different types of media content is proposed, where recommendations are generated considering both the similarity of user ratings and

explicit genre and categorical preferences in cases of data scarcity. **Results.** The platform architecture, recommendation generation mechanism, media data preparation framework, and computational experiment design for evaluating the quality of recommendations and server-side characteristics were developed. **Practical significance.** The proposed solution can be used as the basis for a cross-platform service for personalized media content selection.

Keywords: recommendation system, media content, personalization, collaborative filtering, user preferences, mobile application, Flutter, Serverpod, client-server architecture.

Funding. The work was carried out within the state assignment of the Ministry of Education and Science of Russia; project FEWM-2026-0011.

DOI: 10.21293/1818-0442-2026-29-1-95-101

References

1. Taramov A.A., Shilov N.G. [Recommender Systems for Driver Information Support: State-of-the-Art Review]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki*, 2018, vol. 21, no. 2, pp. 68–74. DOI: 10.21293/1818-0442-2018-21-2-68-74 (in Russ.).
2. Kataev M.Yu., Loseva N.V., Bulyševa L.A. [Structure of an Information Recommender System for Decision Support in the Provision of Services by a Public Institution]. *Doklady Tomskogo gosudarstvennogo universiteta sistem upravleniya i radioelektroniki*, 2018, vol. 21, no. 2, pp. 83–87. DOI: 10.21293/1818-0442-2018-21-2-83-87 (in Russ.).
3. Roy D., Dutta M. A Systematic Review and Research Perspective on Recommender Systems. *Journal of Big Data*, 2022, vol. 9, art. 59. DOI: 10.1186/s40537-022-00592-5.
4. Afoudi Y., Lazaar M., Al Achhab M. Hybrid Recommendation System Combined Content-Based Filtering and Collaborative Prediction Using Artificial Neural Network. *Simulation Modelling Practice and Theory*, 2021, vol. 113, art. 102375. DOI: 10.1016/j.simpat.2021.102375.
5. Zangerle E., Bauer C. Evaluating Recommender Systems: Survey and Framework. *ACM Computing Surveys*, 2022, vol. 55, no. 8, pp. 1–38. DOI: 10.1145/3556536.
6. Panteli A., Boutsinas B. Addressing the Cold-Start Problem in Recommender Systems Based on Frequent Patterns. *Algorithms*, 2023, vol. 16, no. 4, art. 182. DOI: 10.3390/a16040182.
7. Python documentation. Available at: <https://docs.python.org/> (accessed: 16 April 2026).
8. PostgreSQL: Documentation. Available at: <https://www.postgresql.org/docs/> (accessed: 16 April 2026).
9. Flutter documentation. Available at: <https://docs.flutter.dev/> (accessed: 16 April 2026).
10. Serverpod documentation. Available at: <https://docs.serverpod.dev/> (accessed: 16 April 2026).
11. Lercher A., Glock J., Macho C., Pinzger M. Micro-service API Evolution in Practice: A Study on Strategies and Challenges. *Journal of Systems and Software*, 2024, vol. 215, art. 112110. DOI: 10.1016/j.jss.2024.112110.
12. Material Design 3. Available at: <https://m3.material.io/> (accessed: 17 April 2026).
13. Churikov E.A., Zudilova T.V., Ananchenko I.V., Osipov N.A., Ivanov S.E., Osetrova I.S. [Comparison of Modern Mobile Application Development Tools]. *Modern High Technologies*, 2022, no. 12, pp. 82–87. DOI: 10.17513/snt.39441 (in Russ.).
14. Souha A., Benaddi L., Ouaddi C., Jakimi A. Comparative Analysis of Mobile Application Frameworks: A Developer's Guide for Choosing the Right Tool. *Procedia Computer Science*, 2024, vol. 236, pp. 597–604.
15. Oliveira W., Moraes B., Castor F., Fernandes J.P. Analyzing the Resource Usage Overhead of Mobile App Development Frameworks. EASE '23: Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering, University of Oulu, Finland, ACM, 2023, pp. 152–161. DOI: 10.1145/3593434.3593487.

Nikita A. Melekhov

Master's Student, Department of Automation of Information Processing (AOI)
Tomsk State University of Control Systems and Radioelectronics (TUSUR)
74, Vershinina St., Tomsk, Russia, 634045
Phone: +7-999-440-08-30
Email: melekhov206@gmail.com

Vyacheslav Sergeevich Yakimenko

Master's Student, Department of AOI TUSUR
74, Vershinina St., Tomsk, Russia, 634045
Phone: +7-923-432-07-49
Email: slava.yakimenko.2015@gmail.com

Pavel Vasilievich Senchenko

Candidate of Technical Sciences, Associate Professor,
Dean of the Faculty of Control Systems TUSUR
74, Vershinina St., Tomsk, Russia, 634045
Phone: +7 (382-2)-70-15-46
Email: pvs@tusur.ru

Received: 26.11.2025.

Accepted: 28.04.2026.