# О РАЗРАБОТКЕ ВЕБ-СЕРВИСА ДЛЯ СЕМАНТИЧЕСКОГО ПОИСКА ФИЛЬМОВ

Заблоцкий В. В., Кита М. А., Рудикова Л. В.
Кафедра современных технологий программирования, Гродненский государственный университет имени Янки Купалы
Гродно, Республика Беларусь
E-mail: viktorzablotsky@gmail.com, kitaetoya@gmail.com, rudikowa@gmail.com

*В работе излагаются общие подходы к проектированию и реализации веб-сервиса, предназначенного для поиска информации о фильмах на основе семантики их названия и описания.*

## Введение

В современном мире существует огромный спрос на различную продукцию киноиндустрии, в частности фильмы, сериалы, мюзиклы и пр. В этой сфере задействованы не только огромные людские ресурсы, но и бюджеты, сопоставимые с ВВП не самых бедных стран. Так, согласно англоязычной Википедии, годовой оборот киноиндустрии в 2018 году оценивается от 41 до 136 миллиардов долларов. В связи со значительно возросшим уровнем информационных технологий, большую популярность приобретают Интернет-ресурсы, предназначенные для сохранения информации о фильмах и удобного поиска. Такие ресурсы позволяют найти всю необходимую информацию, связанную с каким-либо фильмом, а также изучить оценки и отзывы критиков и зрителей, что позволяет принять взвешенное решение о просмотре фильма. Среди таких ресурсов можно выделить Интернет-базу данных фильмов IMDB и российский аналог – Кинопоиск. Эти ресурсы содержат информацию о миллионах фильмов, которые были созданы с самого зарождения кинематографа и создаются по сей день.

Разрабатываемое приложение преследует те же цели, что и перечисленные выше поисковые базы, однако также позволяет вести поиск не только по прямому совпадению слов в названиях фильмов, но и по семантическому значению поискового запроса. Информация о фильмах, использующаяся в приложении, получена из общедоступных датасетов сайта IMDB.

## I. Этапы разработки веб-сервиса

Работу над реализацией приложения можно разделить на несколько этапов.

Первый этап – анализ предметной области и определение функциональных требований к сервису. На данном этапе были выделены несколько источников данных, пригодных для реализации доменной области, проанализирована структура данных и на ее основе выделены основные направления функциональности, которые должен предоставлять сервис.

Второй этап – проектирование базы данных. Здесь были проанализированы и выбраны оптимальные средства работы с базами данных, а также разработана физическая модель данных приложения.

Третий этап – проектирование архитектуры приложения. В результате были выбраны технологии и средства реализации приложения, а также в качестве рабочей архитектуры была выбрана многоуровневая архитектура клиент-сервер.

Заключительный этап – реализация серверной и клиентской частей приложения, а также их тестирование и развертка в облачной инфраструктуре.

## II. Общие подходы к реализации веб-сервиса

Для реализации веб-приложения используется клиент-серверная архитектура с применением принципов REST. Серверная часть реализована с помощью языка программирования Kotlin с применением фреймворков Spring Boot и Hibernate. В качестве СУБД используется MySQL и хостинг Amazon RDS. Клиентская часть выполнена средствами библиотеки React с применением паттерна Flux (Redux) для управления состоянием приложения. Также была разработана собственная библиотека компонентов с использованием препроцессора стилей SASS. В качестве языка программирования клиентской части выбран Typescript. Веб-приложение разворачивается в изолированной среде с помощью технологии контейнеризации Docker и в среде Amazon Web Services. Сервис состоит из нескольких слабо связанных микросервисов, среди которых можно выделить следующие: сервис авторизации и регистрации, сервис поиска, сервис выработки рекомендаций и сервис обновления данных. Данные сервисы практически не взаимодействуют друг с другом, что позволяет добиться большей автономности в их работе и реализации, однако они используют общую базу данных.

В разрабатываемой поисковой системе можно выделить ключевые архитектурные компоненты и уровни: клиентский уровень (состоит из множества компонентов отображения, компонентов управления состоянием и сервисов, объединенных в модули), прикладной уровень, уровень

бизнес-логики, уровень доступа к данным, уровень хранения данных.

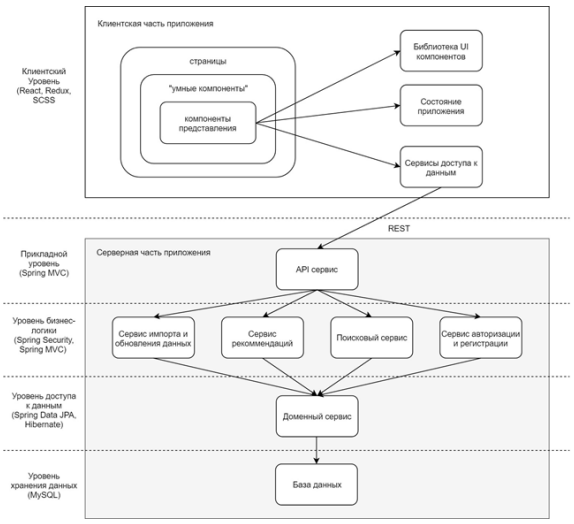


Рис. 1 – Архитектура веб-сервиса

Приложение реализовано по принципам многоуровневой архитектуры и разбито на несколько слоев. Каждый слой является абстракцией над более низкими слоями, предоставляя свой API верхним слоям. Благодаря такому разделению реализуется принцип ограниченной ответственности компонентов и упрощается их параллельная разработка и модификация.

На самом нижнем уровне находится база данных, в которой хранится вся импортированная из датасетов информация, а также пользовательские данные.

Доменный уровень представлен набором сущностей, моделей, репозиториев и сервисов, реализующих доступ к данным и их отображение в объектно-ориентированном стиле.

На уровне бизнес логики находятся сервисы импорта, обновления и поиска данных, а также сервис авторизации и регистрации пользователей.

Прикладной уровень содержит набор конечных точек (endpoints), через которые клиентские приложения могут осуществить доступ к данным, используя HTTP запросы. Таким образом, приложение придерживается RESTful архитектуры, предоставляя API клиентскому приложению.

Важной составляющей частью данного вебсервиса является семантический поиск. Семантический поиск – способ и технология поиска информации, основанная на использовании контекстного значения запрашиваемых фраз, вместо словарных значений отдельных слов или выражений при поисковом запросе. При семантическом поиске учитывается информационный контекст, местонахождение и цель поиска пользователя, словесные вариации, синонимы, обобщенные и специализированные запросы, язык запроса, а также другие особенности, позволяющие получить соответствующий результат. В данной работе, используется уже готовый инструмент word2vec.

Работа с технологией word2vec осуществляется следующим образом: word2vec принимает большой текстовый корпус в качестве входных данных и сопоставляет каждому слову вектор, выдавая координаты слов на выходе. Сначала он создает словарь, «обучаясь» на входных текстовых данных, а затем вычисляет векторное представление слов. Векторное представление основывается на контекстной близости: слова, встречающиеся в тексте рядом с одинаковыми словами (а, следовательно, имеющие схожий смысл), в векторном представлении будут иметь близкие координаты векторов-слов. Полученные векторы-слова могут быть использованы для обработки естественного языка и машинного обучения.

В word2vec существуют два основных алгоритма обучения: CBOW (Continuous Bag of Words) и Skip-gram. CBOW – «непрерывный мешок со словами» модельная архитектура, которая предсказывает текущее слово, исходя из окружающего его контекста. Архитектура типа Skip-gram действует иначе: она использует текущее слово, чтобы предугадывать окружающие его слова. Пользователь word2vec имеет возможность переключаться и выбирать между алгоритмами. Порядок слов контекста не оказывает влияния на результат ни в одном из этих алгоритмов. Получаемые на выходе координатные представления векторов-слов позволяют вычислять «семантическое расстояние» между словами. И, именно основываясь на контекстной близости этих слов, технология word2vec совершает свои предсказания. Так как инструмент word2vec основан на обучении нейронной сети, чтобы добиться его наиболее эффективной работы, необходимо использовать большие корпусы для его обучения. Это позволяет повысить качество предсказаний. В данной работе информация о фильмах кодируется в векторы, которые затем используются для предоставления пользователю возможностей семантического поиска фильмов по их сюжетам.

## III. Заключение

Разработанное приложение представляет собой пользовательский интерфейс для поиска контента, предоставленного датасетами сервиса IMDb. Приложение позволяет просматривать фильмы по категориям, искать и фильтровать по различным критериям фильмы, а также просматривать детализированную информацию о фильмах и личностях. В дальнейшем приложение можно развивать в сторону увеличения отображаемого контента, а также расширения поисковых возможностей и социальной составляющей.