PROJECT PROPOSAL

Audience

Movie Classification and Rating Engine

Name: J. P. T. Jagodage

Introduction

Movie Review should give a brief synopsis of the film, mention the names of the director and actors, and the reviewer's opinion of the directing, the acting, and the film itself. The overall purpose of the review should be a recommendation on whether it is worth spending time and money to see the film.

Audience is a Movie classification and reviewing application which is help movie lovers to select a movie for the watch as well as write their opinion about that movie. This uses microservices architecture for the implementation.

Problem Statement

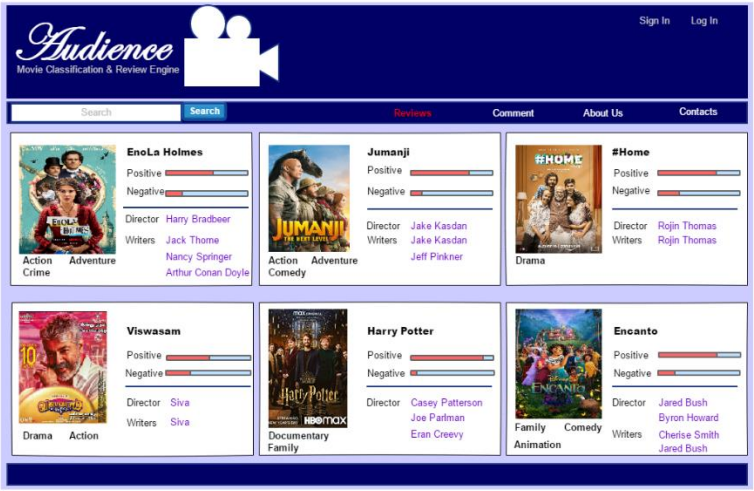
As movie lovers before buying a movie ticket or before downloading the movie, we like to know about that movie. Otherwise, we have to lose our money/data as well as time. The main purposes of watching a movie are a satisfy ourselves and well spend our leisure time. So, if we watched a movie that is not matching for our choice it is wasted. And also, after watching that we like to post our experiences and ideas about that particular movie. To accomplish the above-mentioned scenarios, we need a user-friendly web application that gives reviews about films using our comments.

Motivation

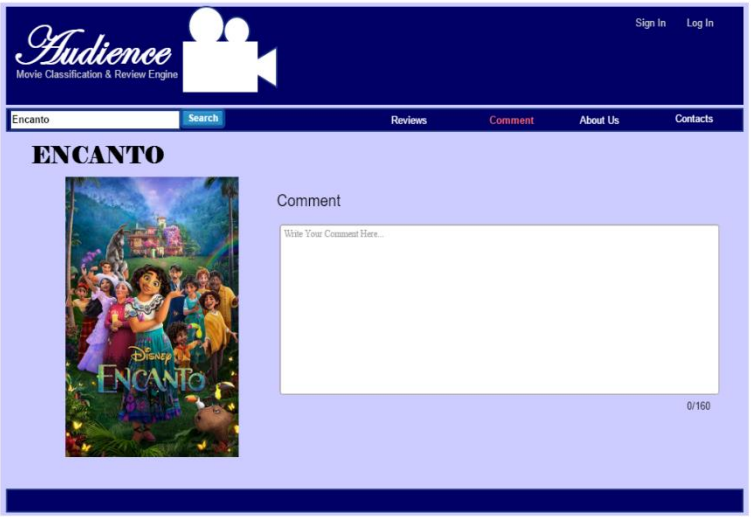
There are lots of movie reviewing applications. But most of them asked for a direct review from the user. It is very valuable if we have an application that can classify our thoughts which we provide in the English language as a comment and analyze the review from that comment. And also, most movie reviewing applications are used monolithic architecture. Microservice can give so many advantages to the application. Those are the motivation that helps to design Movie Classification and Review Engine using microservices.

Solution

To address above mention problems, here we introduce the **Audience** (Movie Classification and Review Engine). It is a microservice application that provides review management and Review Processes as services. Using this You can simply find a movie which is the best match for you to watch. And also, you can write your ideas about that movie using simple English. No need to worry about giving a rating. You can simply type your thoughts and post. The application analyzes your comment and gives the rate and it extracts the details from your comment for the classification.

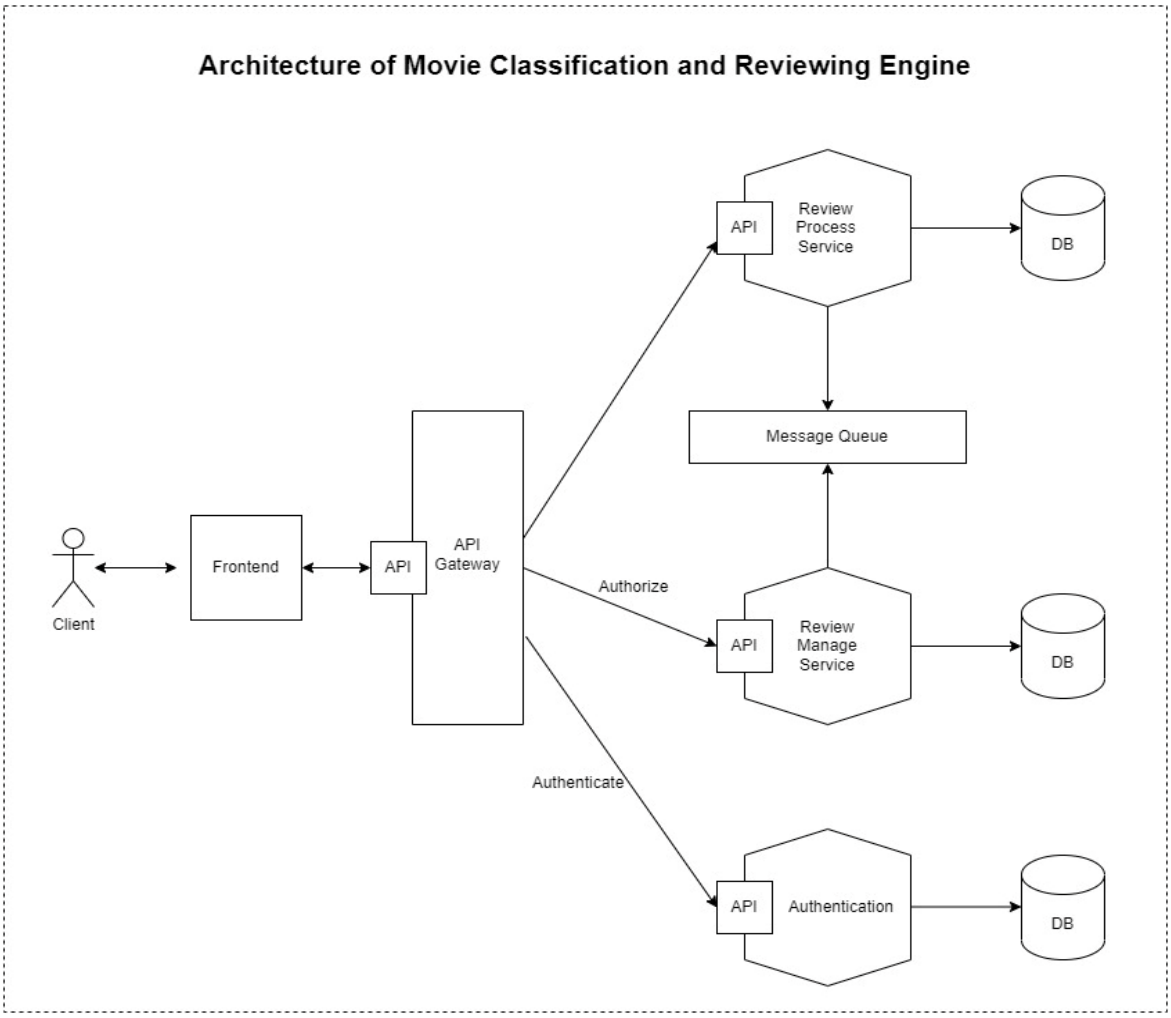


Frontend Review Page



Frontend Comment Page

Architecture



Here used Microservice architecture. Client/User can access the services using the frontend. To build the frontend this used React. API gateway is responsible for request routing, composition, and protocol translation. All request from the client first goes through the API gateway. It then routes requests to the appropriate microservice. Here used three microservices (Authentication, Review Manage Service, and Review Process Service). Those are using Spring Boot for implementation. First Users need to log in to the system then they can select the movie and put a comment. Review Manage Service has a Database it stores that comment and puts a message as "New Review" to the Message Queue. Review Process Services Collect that message. As a Message queue, this uses RabbitMQ. After collecting comments from the Review Manage Service, Review Process services classify that comment using NLP and extract the data from the comment. And stored that data on the database. Then it posts a review as negative, positive, or mixed. The architecture uses Rest APIs for their communication purpose. All databases are implemented using MySQL.

Gantt Chart

Activity	Timeline (2022.02.01-2022.5.31)															
	Feb				March				April				May			
	1st week	2nd week	3rd week	4th week	1st week	2nd week	3rd week	4th week	1st week	2nd week	3rd week	4th week	1st week	2nd week	3rd week	4th week
Title																
Proposal																
Background Study																
Planning																
Frontend Development																
Authentication																
Review Manage Service																
Review Process Service																
Testing																
Documentation																