



دانشگاه شهید بهشتی

((مستند معماری پیشنهادی سامانه فیلمو))

تمرین درس معماری نرم افزار

محمد ربّانی بیدگلی

زمستان ۱۴۰۰

فهرست مطالب

۱	مقدمه	۱
۱.۱	هدف	۱
۱.۲	محدوده‌ی فعالیت	۱
۱.۳	لغت نامه	۱
۱.۴	نمای کاربرد	۲
۱.۵	ویژگی‌های کیفی	۴
۱.۶	مراجع	۵
۲	نمای منطقی	۵
۳	نمای استقرار	۷
۶	نمای پردازش	۸
۷	نمای پیاده‌سازی	۹
۸	نمای تست‌پذیری	۹
۹	نمای لاگ	۱۰
۱۰	نمای پایش	۱۱
۱۱	نمای تکنیک	۱۱
۱۲	ابزارها و فناوری‌ها	۱۳
۱۳	تصمیمات مهم معماری	۱۴

۱ مقدمه

۱.۱ هدف

این مستند برای شناخت معماری و زیرساخت سامانه‌ی فیلمو آماده شده که در آن شناخت نیازهای نرم‌افزاری و سخت‌افزاری نیز مشخص شده است. فیلمو یک سامانه‌ی پخش ویدئو است که در آن مردم می‌توانند انواع فیلم‌ها مانند سینما، سریال، مستند، کارتون و پویانمایی را مشاهده کنند، نظر دهند، پسند یا ناپسند کنند، دانلود کنند یا ذخیره نمایند. در این مستند ابتدا فضای فعالیت این سامانه، نمای موارد کاربرد و محدودیت‌ها جهت آشنایی اولیه توضیح داده می‌شود. سپس وارد فضای راه‌حل و توضیح معماری و نیازمندی‌های هر بخش خواهد شد.

۱.۲ محدوده‌ی فعالیت

با توجه به افزایش سرعت زندگی و عدم زمان کافی، بسیاری از مردم جهان، از جمله ایرانیان، امکان تطبیق زمان خود با زمان فیلم‌هایی که توسط صدا و سیما کشور پخش می‌شود را ندارد و رفتن به سینما هم محدودیت‌ها خود را دارد. این مشکل را با تلویزیون‌های اینترنتی می‌توان حل کرد و افراد در هر زمان و هر مکانی که می‌خواهند، فیلم مورد علاقه‌ی خود را مشاهده کنند. فیلمو نیز در همین راستا فعالیت دارد و هدفش آن است که سامانه‌ای را توسعه دهد تا در سراسر کشور مردم بتوانند بدون محدودیت به محصولات تلویزیونی مانند فیلم‌های با کیفیت بالا بروی هر دستگاه دیجیتال مانند لپ‌تاپ، موبایل یا تلویزیون خانگی دسترسی داشته باشند.

۱.۳ لغت نامه

پیش از آنکه وارد نمای‌های مختلف معماری سامانه فیلمو را شروع کنیم لازم است تا با برخی از کلمات و مفاهیم رایجی که در این سامانه و به طور کلی در کسب و کار مربوط به این سامانه وجود دارد آشنا شویم که در جدول ۱ به مهم‌ترین این واژگان اشاره شده است.

جدول ۱ - لغت نامه

لغت	تعریف
کاربر	به شخصی گفته می‌شود که از سامانه استفاده می‌کند تا بتواند فیلم‌های مورد نظر خود را ببیند

اشتراک	برنامه و گزینه‌هایی که سامانه در اختیار کاربر می‌گذارد تا با خرید آن‌ها بتواند از امکاناتی که سامانه متناسب با اشتراک خریداری شده در اختیار کاربر می‌گذارد استفاده کند. در سامانه سه نوع اشتراک ۱ ماهه، سه ماهه و شش ماهه وجود دارد
اشتراک چند کاربره	برخی از اشتراک‌ها هستند که چند کاربر با یک حساب کاربری همزمان می‌توانند وارد سیستم شوند و برنامه‌های مختلف را مشاهده کنند. برای تعداد کاربران می‌توان محدودیت گذاشت
مدیر سامانه	به شخصی گفته می‌شود که عضو سازمان است و به پنل مخصوص مدیران که برای آپلود فیلم‌ها، مدیریت کاربران و اشتراکات ایجاد شده است دسترسی دارد
فیلم	هر نوع برنامه‌ی تلویزیونی که بتوان آن را بر بستر اینترنت منتشر(یا به اصطلاح استریم) کرد را فیلم می‌گویند که انواع مختلفی دارد: سریال، مستند، سینمایی، پویانمایی
استریم	به معنای همان پخش فیلم بر بستر شبکه و به صورت آنلاین است
دانلود درون برنامه‌ای	در این سامانه کاربر می‌تواند برنامه را دانلود کند، اما برنامه‌ی دانلود شده نباید بر روی خود دستگاه کاربر ذخیره شود، به گونه‌ای که به فیلم دانلود شده دسترسی داشته باشد و بتواند آن را کپی و انتشار دهد. فیلم دانلود شده باید تنها در خود سامانه قابل نمایش باشد اما به صورت آفلاین نیز کاربر بتواند فیلم را ببیند

۱.۴ نمای کاربرد

در نمای کاربرد به استفاده‌هایی که یک کاربر و مدیر سامانه می‌تواند از سامانه‌ی فیلمو انجام دهد آمده است. در جدول ۲ چند مورد کاربرد اصلی آمده است.

شناسه	کنشگر	مورد کاربرد	توضیح
U1	کاربر	مشاهده‌ی فیلم	سامانه این امکان را باید به کاربران بدهد تا بتوانند فیلم‌های مورد نظر خود را در هر زمانی که می‌خواهند ببینند
U2	کاربر	خرید اشتراک	برای مشاهده‌ی بیشتر فیلم‌ها کاربران باید اشتراک خریداری کرده باشند. این اشتراک‌ها به صورت ماهیانه تمدید می‌گردند
U3	کابر	دانلود فیلم	کاربرانی که اشتراک می‌خرند، سامانه باید برای آن‌ها امکان دانلود فیلم را به صورت درون برنامه‌ای فراهم کند. یعنی کاربران بدون اتصال به اینترنت باید بتواند از طریق برنامه فیلم دانلود شده را ببیند
U4	مدیر سیستم	مدیریت کاربران	سامانه باید این اجازه را به مدیران بدهد تا بتوانند اطلاعات تمام کاربرانی که از سامانه استفاده می‌کنند را ببینند. این اطلاعات می‌تواند شامل نام و نام‌خانوادگی، شماره همراه، ایمیل، اشتراکات خریداری شده، مدت اعتبار اشتراک، تاریخ عضویت باشد
U5	مدیر سیستم	بارگزاری فیلم	سامانه باید امکان آپلود فیلم، جهت قرار گرفتن بروی سامانه‌ی جهت دسترسی کاربران را فراهم کند
U6	کاربر	ذخیره‌ی فیلم	سامانه باید این امکان را به کاربر بدهد تا فیلم مورد نظر خود را ذخیره نماید تا در هر زمانی که می‌خواهد به فیلم مورد نظر خود دسترسی داشته باشد. اما برای مشاهده‌ی فیلم یا باید آن را دانلود کند یا می‌تواند به صورت برخط فیلم را ببیند
U7	کاربر	ثبت نظر	سامانه باید به کاربر این اجازه را بدهد تا نظر خود را در مورد فیلم مشاهده شده ثبت نماید
U8	کاربر	پسندیدن یا نپسندیدن	کاربر می‌تواند بدون نوشتن نظر نیز با پسندیدن یا نپسندیدن فیلم نظر خود را بگوید

۱.۵ ویژگی‌های کیفی

در این قسمت به توصیف برخی از ویژگی‌های کیفی پرداخته می‌شود که رعایت آن‌ها برای سیستم بسیار مهم است تا سیستم بتواند نیازهای کاربران و مدیران را به خوبی تامین کند. در جدول ۳ به توصیف این ویژگی‌های مهم پرداخته شده است.

جدول ۳ - ویژگی‌های کیفی

امنیت	<ul style="list-style-type: none"> - تعداد کاربرانی که از یک اشتراک می‌توانند همزمان استفاده نمایند محدود است و نباید در یک زمان بیشتر از تعداد تعیین شده در حال فیلم دیدن باشند - جلوگیری از امکان ذخیره‌ی فیلم دانلود شده بر روی دستگاه کاربر جهت کپی - کنترل تعداد درخواست‌های بیش از اندازه توسط سیستم‌ها برای مشاهده‌ی فیلم‌ها در یک زمان (نوعی DDOS)
تست‌پذیری	<p>تست برنامه باید سریع و آسان باشد زیرا کمک می‌کند تا میزان صحت عملکرد برنامه و همچنین دسترس‌پذیری نیز افزایش یابد.</p> <p>میزان تست‌پذیری را می‌توان با تعداد موردهای تست نوشته شده، مدت زمان لازم برای اجرای تست و میزان درصد تست‌هایی که خودکار یا دستی هستند سنجید.</p> <p>میزان تست‌پذیری را می‌توان با توسعه‌ی ابزار تست خودکار و تهیه‌ی موارد تست‌های موثر افزایش داد</p>
مقیاس‌پذیری	<p>زیرساخت باید تحمل این را داشته باشد تا تعداد کاربرانی که در هر لحظه برای مشاهده‌ی یک فیلم مشترک وارد سیستم می‌شوند را کنترل کند و خاموش نشود یا دسترسی یک کاربر را نبندد.</p>
عملکرد	<p>منابعی مانند حافظه، CPU، پهنای باند شبکه باید به درستی مدیریت شوند و در ساعاتی که بار شبکه و درخواست‌ها زیاد می‌شود، بتوانند خوب کار کنند و وظایف به خوبی میان سیستم‌های موجود تقسیم شود</p> <p>زمان انجام کارها بسیار مهم است و زمان پاسخ به درخواست‌ها و استریم کردن فیلم باید متناسب با اینترنت پ پهنای باند شبکه‌ی کاربر تنظیم شود</p>

دسترس پذیری	<p>مقدار زمانی که فیلم‌ها برای تماشا در دسترس هستند که با رابطه‌ی زیر می‌توان میزان دسترسی را سنجید:</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p style="text-align: center;">میانگین زمان شکست</p> <hr style="width: 50%; margin: 5px auto;"/> <p style="text-align: center;">میانگین زمان تعمیر + میانگین زمان</p> </div> <p>با توجه به رابطه‌ی بالا باید با کاهش میانگین زمان تعمیر و افزایش میانگین زمان شکست، دسترسی را بهبود داد</p>
-------------	---

۱.۶ مراجع

- <https://www.filimo.com>
- <https://medium.com/@narengowda/netflix-system-design-dbec30fede8d>
- <http://highscalability.com/blog/2017/12/11/netflix-what-happens-when-you-press-play.html>
- <https://anushadatta.medium.com/netflix-system-architecture-aff09de0fc29>
- <https://www.aparat.com/v/6Rvzo>

۲ نمای منطقی

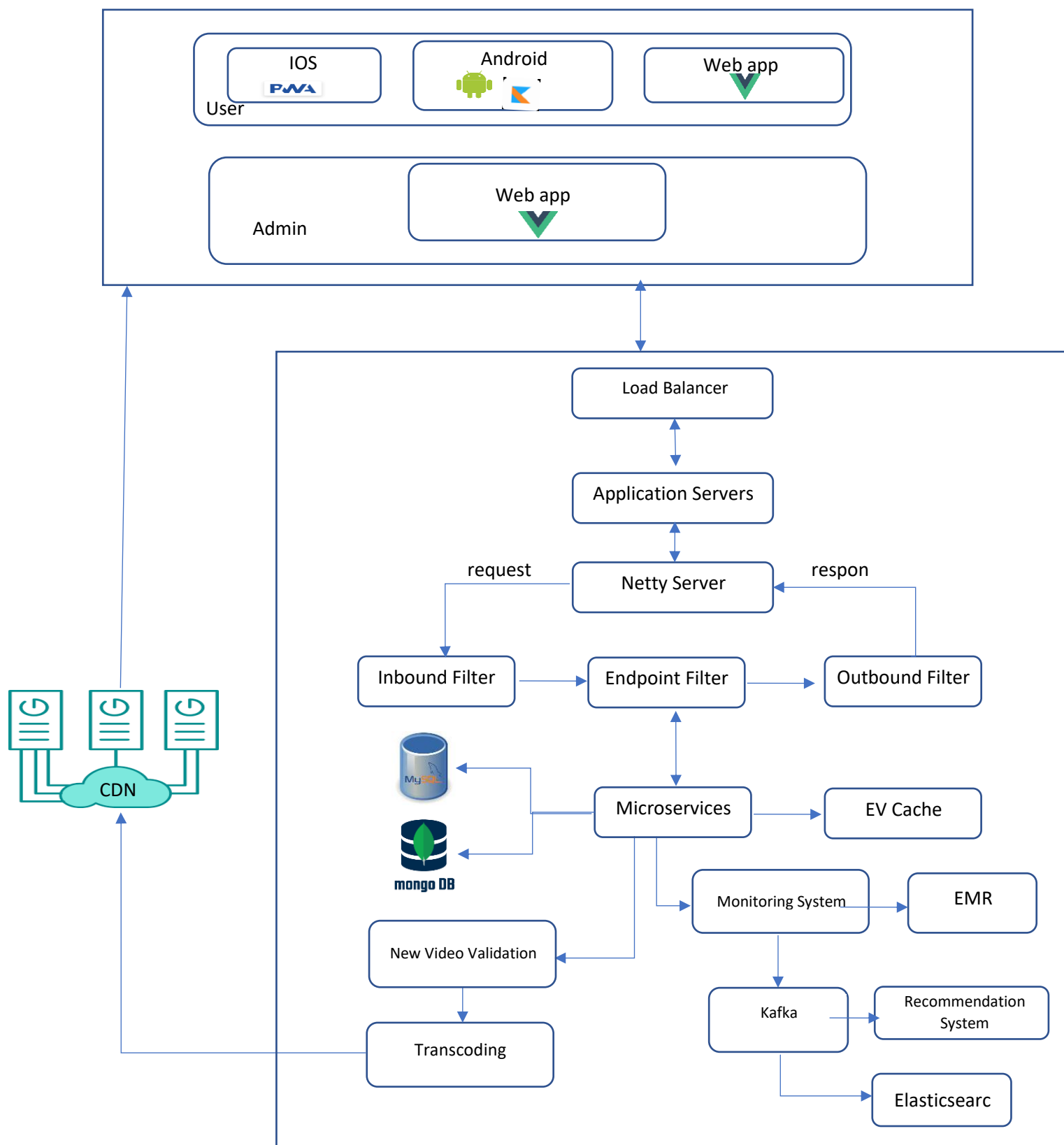
در نمای منطقی شکل کلی معماری اتخاذ شده در تصویر ۱ نمایش داده شده است. این معماری از سه قسمت تقسیم می‌شود: Client، Back-End و CDN.

Client قسمتی است که کاربر و مدیران سیستم با آن کار می‌کنند و از امکاناتی که سامانه در اختیار آن‌ها قرار می‌دهد استفاده می‌کنند.

Back-End نیز قسمتی است که در آن تمام زیرساخت‌های سامانه مانند پایگاه داده، میکروسرویس‌ها، سرورهای Cache و غیره قرار دارند.

CDN نیز برای قرار گرفتن فیلم‌ها بر روی آن و دسترسی سریع‌تر به آن‌ها است. زیرا بر روی CDN میتوان دارایی‌های ثابت برنامه را قرار داد و به اینصورتی بار ترافیکی را تا حد زیادی از روی سرورهای اصلی کاهش داد. بنابراین پس از آنکه فیلم‌ها توسط مدیران سامانه آپلود می‌شوند و تبدیل فرمت فیلم‌ها به فرمتی که بر روی دستگاه‌ها بهتر اجرا شوند توسط فرایند transcoding، این فیلم بر روی سرورهای CDN قرار می‌گیرند

تصویر ۱- نمای کلی معماری سامانه



۳ نمای استقرار

در این نما مشخصات زیرساخت‌های مورد نیاز برای راه‌اندازی سامانه توضیح داده شده. در جدول ۴ توضیح مشخصات هر کدام از سرورهای مورد نیاز نمایش داده شده است.

جدول ۴ - مشخصات سرورها مورد نیاز

شناسه	توضیح کارکرد	نوع ماشین	تعداد ماشین	تعداد و نوع پردازنده	حافظه اصلی	حافظه جانبی
DEP1	DB server	مجازی	2	4x3.8GHz	32 GB	50 GB SSD
DEP2	Cache server	مجازی	1	4x2GHz	16 GB	100 GB SSD
DEP3	CDN	مجازی	1	Quad core	8 GB	1 TB
DEP4	Application servers	مجازی	2	8	32 GB	200 GB SSD
DEP5	Load balancer	مجازی	1	2	4 GB	32 GB
DEP6	Monitoring	مجازی	3	4	16 GB	2400 GB
DEP7	CI/CD	مجازی	1	4	16 GB	1 TB

۶ نمای پردازش

در بخش نمای پردازش به برنامه‌ها و پردازش‌هایی پرداخته شده است که بر روی سرورها در حال اجرا هستند. در جدول ۵ نام پردازشی در حال اجرا و آنکه بر روی کدام سرور در حال اجرا می‌باشد نشان داده است.

جدول ۵- پردازش‌های در حال اجرا

پردازش	سرور/ماشین مجازی/مکان	توضیح
Nginx	ماشین مجازی DEP4 و DEP5	به عنوان application server و load balancer
Mysql	ماشین مجازی DEP1	بر روی سرور DB برای ذخیره‌ی اطلاعات کاربران
mongoDB	ماشین مجازی DEP1	برای ذخیره اطلاعات فیلم‌ها
Elasticsearch	ماشین مجازی DEP6	بر روی Monitoring و برای یافتن مشکل ایجاد شده در سیستم
Kafka	ماشین مجازی DEP6	به عنوان سرویس مسیریابی برای انتقال اطلاعات به Elasticsearch
Redis	ماشین مجازی DEP5	به عنوان سرویس Cache
Jeinkins	ماشین مجازی DEP7	به عنوان سرویس CI/CD
PRTG	ماشین مجازی DEP6	به عنوان سرویس پایش سیستم
Docker	ماشین مجازی DEP4	به عنوان Container

۷ نمای پیاده‌سازی

در این نما به فناوری‌های برنامه‌نویسی که برای پیاده‌سازی برنامه استفاده شده است اشاره شده است. در جدول ۶ برخی از فناوری‌های نرم‌افزاری اصلی که برای پیاده‌سازی برنامه مورد نیاز است آمده است. فناوری‌های استفاده شده در سه سطح مشتری، سمت سرور و زیرساخت استفاده شده‌اند.

جدول ۶ - فناوری‌های برنامه‌نویسی

واستگی	فناوری	شرح	Module/package/sub-project
Vue 3, node	Vue Typescript Css Html yarn	پیاده‌سازی front وب	Client (مشتری)
	Kotlin	پیاده‌سازی فرانت Android	
Vue 3, Service worker	مانند پیاده‌سازی وب ولی با امکان PWA	پیاده‌سازی iOS	
Php 8	PHP Laravel-blade	پیاده‌سازی سرویس‌ها	سمت سرور
	Bash Script PHP	پیاده‌سازی زیرساختی	زیرساخت

۸ نمای تست‌پذیری

در این بخش ابزاری که برای تست برنامه در قسمت‌های مختلف مانند واسط کاربری و سمت سرور (back-end) استفاده شده است در جدول ۶ نشان داده شده است.

نوع	توضیح	کتابخانه	توضیح
UI testing	Component testing	Cypress	هر component که پیاده‌سازی می‌شود به صورت ایزوله عملکرد هم کدام تست می‌گردد
	Integration testing		هر صفحه‌ای که پیاده‌سازی می‌شود که مجموعه‌ای از componentها است، تست می‌شود و صحت ارتباط میان componentها و عملکرد کلی صفحه بررسی می‌گردد
Logic testing	Unit testing	Jest	هر تابعی که پیاده می‌شود به صورت جداگانه تست می‌شود

۹ نمای لاگ

تمام فعالیت‌های کاربر و سیستم، خطاها و هشدارهایی که در سیستم رخ می‌دهد در یک سرور ذخیره می‌شوند. این رخدادهایی که ذخیره می‌شوند عبارت‌اند از:

- زمان و تاریخ عضویت یک کاربر
- زمان و تاریخ ورود و خروج کاربر
- زمان و تاریخ خرید اشتراک و نوع اشتراک خریداری شده
- فیلم‌هایی که کاربران جهت مشاهده انتخاب می‌کنند
- زمان و تاریخ ثبت نظر، نام کاربری نویسنده و متن نوشته شده
- زمان و تاریخ ثبت پسندیدن یا نپسندیدن به همراه نام کاربری
- زمان و تاریخ خطاهایی که در سیستم رخ می‌دهد به همراه گزارش کننده خطا (کدام سرور است)، از سمت سرور یا از سمت کلاینت، صفحه و endpoint که خطا رخ داد
- زمان و تاریخ آپلود فیلم که توسط چه کسی انجام شده است و فیلم آپلودی دارای چه مشخصاتی است

همچنین لاگ‌ها در سیستم با یک ساختار مشخصی ذخیره می‌شوند که به صورت زیر است:

- تاریخ: زمان - نوع فعالیت - اکشن یا خطا - actor - توضیحات

برای مدیریت لاگ نیز از ابزار Fluent استفاده می‌شود که یک ابزار متن باز است و می‌توان با اتصال آن به Elasticsearch جستجو در لاگ‌ها را بهتر انجام داد. در این ابزار می‌توان لاگ‌ها را دسته‌بندی کرد و ساختار مشخصی نیز تعریف نمود. همچنین امکان گزارش گیری و تحلیل بر روی لاگ‌ها را نیز داراست.

۱۰ نمای پایش

برای پایش سیستم از یک ابزار معروفی به نام PRTG استفاده شده است که به کمک آن می‌توان بر موارد زیر نظارت داشت و کنترل و بررسی نمود:

- سلامت سرورها
- میزان درخواست‌ها
- مدت پاسخ به زمان درخواست‌ها
- فضای حافظه‌ی اشغال شده در هر سرور
- میزان تحمل بار هر سرور

همچنین این ابزار را به گونه‌ای می‌توان تنظیم کرد که در صورت بروز اختلال یا خطای یک پیامک یا ایمیل به مسئولین پشتیبانی و زیرساخت ارسال شود تا سریعاً متوجه مشکل شوند و در راستای برطرف نمودن مشکل اقدامات لازم را انجام دهند.

۱۱ نمای تکنیک

در این بخش به معرفی و توضیح تکنیک‌ها و روش‌هایی که برای رعایت و افزایش ویژگی‌های کیفی ذکر شده در فصل ۱۰.۳، پرداخته می‌شود. در جدول ۸ توضیح این موارد آمده است.

جدول ۸- تکنیک‌های مورد استفاده برای بهبود ویژگی‌های کیفی

ویژگی کیفی	تکنیک مورد استفاده	توضیح
امنیت	<ul style="list-style-type: none"> • استفاده از Object level Authorization • استفاده از JWT برای احراز هویت کاربران • استفاده از تکنیک Rate-limit 	مراجعه به فصل پیاده سازی
دسترسی	<ul style="list-style-type: none"> • استفاده از سرویس‌های Cache برای دسترسی و پردازش سریع • استفاده از سرورهای پشتیبانی در صورت از بروز مشکل • پایش تمام سرورها و برنامه‌های در حال اجرا به کمک ابزار PRTG 	مراجعه به فصل نمای منطقی و پایش
تست‌پذیری	<ul style="list-style-type: none"> • استفاده از خودکار سازی تست‌های مربوط به زمان توسعه برای منطق و ظاهر برنامه 	مراجعه به فصل تست‌پذیری
عملکرد	<ul style="list-style-type: none"> • استفاده از سرورهای Cache جهت افزایش سرعت پاسخگویی • استفاده از CDN برای کاهش بار بر روی سرورها پردازشی و سرعت بیشتر در استریم برنامه • استفاده از load balancer برای پخش کردن درخواست به چندین سرور • برای استقرار سریع برنامه از Jenkins به عنوان CI/CD استفاده شده است 	مراجعه به فصل استقرار و نمای منطقی
مقیاس‌پذیری	<ul style="list-style-type: none"> • استفاده از Docker container • استفاده از روش میکروسرویس 	مراجعه به فصل نمای منطقی

۱۲ ابزارها و فناوری‌ها

در این بخش ابزارها، فناوری‌ها و کتابخانه‌هایی که برای پیاده‌سازی سامانه استفاده شده است، در جدول ۹ به همراه شماره‌ی نسخه و جایگاه استفاده هر کدام نشان داده شده است.

جدول ۹ - ابزارها، فناوری و کتابخانه‌های استفاده شده در سامانه

ابزار/الگو/فناوری	نسخه	جایگاه
Vue	3.14.0	web app ,Front-End
Vuex	4.0.2	web app ,Front-End
Vue-router	4.0.0	web app ,Front-End
node	16.13.1	DEP4 ,web app ,Front-End
Yarn	3.1.1	DEP4 ,web app ,Front-End
TypeScript	4.1.0	web app ,Front-End
Sass	1.26.5	web app ,Front-End
Sass-loader	8.0.2	web app ,Front-End
Axios	0.21.1	web app ,Front-End
Gradle	7.3.0	Android ,Front-End
Kotlin	1.6.0	Android ,Front-End
Retrofit	2.9.0	Android ,Front-End
Kafka	3.0.0	DEP6 ,Back-End
Redis	6.2.6	DEP3 ,Back-End
Elasticsearch	7.16.2	DEP6 ,Back-End
MySQL	8.0	DEP1 ,Back-End
MongoDB	5.1	DEP1 ,Back-End
Docker	20.10	DEP7 ,Back-End
Php	8	Back-End
Laravel-blade	8.1	Back-End
Nginx		DEP4 ,Back-End
JWT		Back-End
WebStorm IDE	2021.3.1	Front-End
PHPStorm IDE	2021.3.1	Back-End
Android Studio IDE	2020.3.1	Front-End

۱۳ تصمیمات مهم معماری

در این قسمت توضیح تصمیمات گرفته شده و علت گرفتن آن تصمیم پرداخته می‌شود. همچنین تصمیم جایگزینی هم می‌شد گرفت نیز آمده است که در جدول ۹ قابل مشاهده استفاده است.

جدول ۱۰ - تصمیمات مهم گرفته شده در معماری سامانه فیلیمو

تصمیمات معماری	گزینه‌های دیگر	دلیل این تصمیم
استفاده از vue به عنوان فریم‌ورک سمت وب	React	سادگی و یادگیری سریع، سرعت بهتر و افزایش تعداد توسعه دهندگان این تکنولوژی
استفاده از TypeScript به عنوان زبانی پیاده‌سازی برنامه‌های وب	JavaScript	امکانات زیادی که TypeScript نسبت به JavaScript فراهم می‌کند. مانند تعریف نوع داده، قابلیت ارث بری انواع داده تعریف شده، پیدا کردن راحت‌تر خطاها و تمیز تر شدن کد
استفاده از Kotlin در توسعه‌ی برنامه‌ی اندروید	Java	به دلیل سادگی در برنامه‌نویسی نسبت به جاوا و سرعت بهتر
استفاده از PWA برای iOS	استفاده از Swift	وجود تحریم و عدم امکان ثبت برنامه در app store
استفاده از MongoDB	Cassandra	سرعت و کارایی بالاتر
استفاده از میکروسرویس	یکپارچه یا سرویس‌گرایی (سنتی)	سرعت در پیاده‌سازی عدم وابستگی اجزا تقسیم راحت‌تر بین تیم شکسته‌بندی ریزتر
استفاده از kafka	Redis/RabbitMQ	امکانات بیشتری که در اختیار توسعه‌دهندگان می‌گذارد تجربه‌ی پیاده‌سازی
استفاده از Redis به عنوان Cache	Memcache	امکانات بسیار زیادی که نسبت به memcache دارد

پیاده سازی ساده و بدون پیچیدگی		
تجربه‌ی پیاده‌سازی Docker و برآورده کردن نیازها	Kubernetes	استفاده از Docker
به دلیل وجود تحریم و همچنین هزینه‌ی زیاد استفاده از سیستم‌های خارجی	AWS، Azure	استفاده از CDN شرکت ابرآروان

پایان