

قالب سند معماری نرم افزار

→ بازگشت به صفحه اصلی

قالب مستند معماری نرم افزار

نسخه ۲.۲

تاریخ آخرین بازبینی: ۰۲/۰۹/۹۹

سند زیر شامل بخشی از قالب پیشنهادی سند معماری نرم افزار است

پیش‌گفتار: در مستند معماری نرم افزار، معماری یک سامانه نرم افزاری توصیف می‌شود. معماری نرم افزار، شامل تصمیمات مهم و کلان در طراحی نرم افزار است. هر آن چه از دید کلان و از دید معماری مهم است، در این مستند گنجانده می‌شود و جزئیات و فرعیات (مانند جزئیات پیاده‌سازی) در این مستند گنجانده نمی‌شود. قالب مطالب و سرفصل‌های لازم برای مستند معماری در ادامه آمده است. مستند معماری باید این سرفصل‌ها را به شکلی شفاف و مناسب، در کوتاه‌ترین و موجزترین حجم ممکن پوشش دهد. همچنین تغییرات مهم در نرم افزار که از نظر معماری مهم هستند، باید در سند معماری منعکس شوند. به عبارت دیگر، مستند معماری باید در طول فرایند توسعه نرم افزار، نگهداری و به‌روز شود.

مجدداً تاکید می‌شود که مستند معماری، باید در موجزترین و کوتاه‌ترین شکل ممکن، موارد زیر را پوشش دهد. هدف از تولید این مستند، مستندسازی حجیم و با جزئیات فراوان نیست، بلکه نگهداری یک دید سطح بالا و گویا از معماری کلان سیستم نرم افزاری است. تکرار کردن اطلاعات در این مستند و یا کپی کردن از مستندات موجود به منظور پر کردن یا حجیم کردن مستند معماری، آفتی است که معمولاً مستند معماری را طولانی و بی‌فایده می‌کند.

سرفصل‌های مستند معماری در ادامه آمده است.

عنوان سامانه: ???

سند معماری (SAD)

نسخه سند: ???

تاریخ آخرین بازبینی: ???

- [قالب مستند معماری نرم افزار](#)
- [نسخه ۲.۲](#)
- [عنوان سامانه: ???](#)
- [۱. مقدمه](#)
- [۲. نمای موارد کاربرد](#)
- [۳. ویژگی های کیفی](#)
- [۴. نمای منطقی](#)
- [۵. نمای استقرار \(Deployment\) یا \(Physical\)](#)
- [۶. نمای پردازش \(Process\)](#)
- [۷. نمای توسعه و پیاده سازی \(Development\) یا \(Implementation\)](#)
- [۸. نماهای متقاطع \(اختیاری\)](#)
- [۹. نحوه تحقق موارد کاربرد کلیدی](#)
- [۱۰. نمای تست](#)
- [۱۱. مدیریت لاگ](#)
- [۱۲. پایش \(Monitoring\)](#)
- [۱۳. نمای داده](#)
- [۱۴. نمای گزارش گیری](#)
- [۱۵. الگوها و تکنیک های مورد استفاده](#)
- [۱۶. ابزارها و فناوری ها](#)
- [۱۷. مهم ترین تصمیمات معماری](#)
- [۱۸. ریسک ها و بدهی های فنی](#)

- [۱۹. بهبودهای پیشنهادی در آینده](#)
- [۲۰. سایر نماهای مفید/لازم](#)
- [ترتیب پیشنهادی برای تکمیل سند معماری](#)
- [مراجع](#)
- [چک لیست](#)
- [اشتباه‌های رایج در نگارش سند معماری](#)
- [پیوست](#)

۱. مقدمه

محدوده

توصیفی کوتاه از محدوده (scope) کاربرد سامانه موردنظر

فهرست تعاریف، اختصارات و واژه‌نامه

واژه‌ها و عبارت‌هایی که در این مستند به آن‌ها اشاره خواهد شد و معنی آن‌ها بدیهی نیست و یا به شکل ساده و با جستجو در وب قابل حصول نیست، در این بخش ذکر می‌شوند. به‌ویژه، ذکر فهرست عبارت‌ها، اختصارات و واژه‌های تخصصی که در حیطه همین سامانه نرم‌افزاری تولید یا تبیین شده‌اند، الزامی است.

اهداف معماری

اهداف کلان معماری به شکل کوتاه (معمولاً در حد یک پاراگراف) ذکر شود.

محدودیت‌ها و استانداردها

(Constraints, Standards, Background)

محدودیت‌ها، استانداردها و بایدهایی که در طراحی معماری به آن‌ها توجه شده است، ذکر شوند. همچنین پیش‌زمینه‌ای (Background) که در کسب‌وکار (Business)، تیم توسعه (مثلاً تجربه‌های مشابه یا تخصص‌های موجود) و یا تصمیمات معماری از قبل وجود داشته است، ذکر شود.

مراجع

اگر ارجاع به مستند یا مرجع دیگری لازم است، در این بخش ذکر شود.

۲. نمای موارد کاربرد

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۳. ویژگی‌های کیفی

در این بخش، ویژگی‌های غیرکارکردی (Nonfunctional) یا به عبارت دیگر ویژگی‌های کیفی (Quality Attributes) مورد نیاز در سامانه نرم‌افزاری موردنظر ذکر شود. به عنوان مثال امنیت، کارایی، مقیاس‌پذیری و ...

یکی از اهداف این فصل آن است که کارفرما یا بهره‌بردار مطمئن شود به همه نیازمندی‌های کیفی (غیرکارکردی) موردنظر وی در معماری سامانه نرم‌افزاری توجه شده است و راهکارهای لازم برای رسیدن به این نیازمندی‌ها در معماری نرم‌افزار پیش‌بینی و تبیین شده است.

این بخش به شکل یک جدول، مشابه جدول زیر بیان شود. مقادیر ستون «اندازه/محدوده مطلوب» به عنوان نمونه ذکر شده است. سطرهای لازم مطابق با نیازمندی‌های پروژه به این جدول اضافه شود. همان‌طور که در مثالهای موجود در این جدول مشهود است، برای هر ویژگی کیفی (مثلاً کارایی) معمولاً چند معیار ارزیابی ذکر می‌شود (مثلاً TPS و سرعت) و برای هر معیار ارزیابی اندازه یا بازه مطلوب نیز ذکر می‌شود.

توجه: در این بخش به ویژگی‌های پایه مانند کارایی، دسترس‌پذیری، مقیاس‌پذیری، آزمون‌پذیری، تغییرپذیری (قابلیت نگهداری و گسترش) و تعامل‌پذیری توجه ویژه شود و حتماً این موارد در جدول مربوطه ذکر شود.

ویژگی کیفی	معیار ارزیابی	اندازه/محدوده مطلوب
کارایی (Performance)	TPS	بیش از ۱۰۰
کارایی	تعداد کاربران هم‌زمان	بیش از ۱۰۰۰

کارایی	سرعت پاسخ‌گویی	میانگین تأخیر کمتر از ۰.۵ ثانیه در پاسخ به درخواست کاربر
دسترس‌پذیری (Availability)	Up-Time	۹۹.۹٪
دسترس‌پذیری	تعداد دفعات down-time	میانگین کمتر از یک بار در هفته
مقیاس‌پذیری (Scalability)	قابلیت scale-up	به شکل کاملاً خودکار پشتیبانی می‌شود
مقیاس‌پذیری	قابلیت Scale-out	در همه لایه‌های نرم‌افزاری
مقیاس‌پذیری	Database Scalability	امکان Scale-up وجود دارد ولی امکان Scale-out وجود ندارد
امنیت (Security)	...	...
آزمون‌پذیری (Testability)	زمان اجرای تست‌های خودکار	کمتر از یک ساعت
آزمون‌پذیری	پوشش تست	بیش از ۶۰ درصد
آزمون‌پذیری	پوشش نیازمندی‌ها در تست خودکار	بیش از ۴۰ درصد
قابلیت تغییر و نگهداری (Maintainability/Modifiability)	...	
...		

ستون‌های اصلی این جدول عبارتند از:

- ویژگی کیفی موردنظر
- معیارهای ارزیابی
 - تعیین معیارهای مشخص (measurements یا metrics) برای ارزیابی وضعیت «ویژگی‌های کیفی» موردنظر. در این بخش، روش‌هایی برای محاسبه عددی مقدار ویژگی‌های کیفی مشخص می‌شود. به عبارت دیگر، ویژگی‌های کیفی، کمی (quantify) می‌شوند. به عنوان

مثال، برای ویژگی کیفی «کارایی» معیارهایی مثل TPS یا میانگین سرعت پاسخ به کاربر یا ... بیان می‌شود.

- تعیین مقدار یا محدوده مطلوب معیارهای ارزیابی
 - در این بخش، بازه‌ی مطلوب برای مقدار معیارهای مختلف ارزیابی بیان می‌شود. مثلاً: برای مقدار TPS، حداقل به ۱۰۰ تراکنش در ثانیه برسیم. یا حداکثر زمان پاسخ به هر کاربر ۵ ثانیه باشد. یا حداقل به ۱۰۰ کاربر هم‌زمان سرویس داده شود.

توجه: تا بدین جای مستند، تمرکز مستند معماری بر بیان «مسئله» (Problem) بوده است. به عبارت دیگر نیازمندی‌های کارکردی و غیرکارکردی را مرور کردیم و دغدغه‌هایی که باید در معماری به آن‌ها توجه شود بیان شده است. از این زیرفصل به بعد، بر «راه حل» (Solution) تمرکز می‌شود. به عبارت دیگر، کلیات تصمیمات معماری و راه‌حل ارائه‌شده برای حل مساله موردنظر (نرم‌افزار موردنظر) توصیف می‌شود.

۴. نمای منطقی

در این بخش، طراحی منطقی معماری توصیف می‌شود.

اجزای اصلی معماری

اجزای اصلی معماری شامل مولفه‌ها، زیرسیستم‌ها و سرویس‌ها توصیف می‌شوند. مهمترین کارکرد این بخش، توصیف نحوه تقسیم (شکست) این نرم‌افزار به اجزای اصلی آن است. وجود یک نمودار کلی شامل یک نمای مفهومی سطح بالا (conceptual view) در این بخش حیاتی است. بهتر است این بخش، متن کمتری داشته باشد و شامل یک یا چند نمودار گویا (حداقل یک نمودار و حداکثر پنج نمودار) باشد. به خصوص یک نمودار سطح بالا که اجزای اصلی (مؤلفه‌های کلان) معماری را نشان می‌دهد و حتی می‌تواند در قالب نمودارهای غیررسمی بیان شود (مثلاً برای هر جزء اصلی معماری یک مستطیل بکشید و ارتباطات بین اجزا را مشخص کنید).

اگر در معماری، به رویکرد Domain Driven Design توجه داشته‌اید، Bounded Context ها را در این بخش توصیف کنید.

در صورت نیاز فرایند و توالی تعاملات بین مؤلفه‌های مختلف هم در قالب نمودارهایی مثل sequence diagram ذکر شود که در آن‌ها نحوه محقق شدن موارد کاربرد توسط زیرسیستم‌های اصلی نمایش داده

می‌شود. اما همان‌طور که قبلاً ذکر شد، قبل از این نمودارهای توالی، ذکر یک نمودار ساده سطح بالا از مؤلفه‌های کلان سیستم ضروری است. به عبارت دیگر، حضور یک high-level conceptual view کمک می‌کند تقسیم‌بندی «ساختار» معماری نشان داده شود. همچنین حضور نمودارهای توالی (sequence-diagrams) برای برخی از موارد کاربرد اصلی سیستم، می‌تواند رفتار و دینامیک ارتباطات اجزای معماری را نشان دهد.

برخی از سوالاتی که در این بخش پاسخ داده می‌شوند:

- معماری کلان این سامانه، چه اجزا و مؤلفه‌هایی (Element/Component) دارد؟
- ارتباطات و تعاملات بین مؤلفه‌های مختلف چگونه است؟
- برای برخی موارد کاربرد اصلی، گامها و فرایند محقق شدن مورد کاربرد توصیف شود و تأثیر و نقش هر گام در هر مؤلفه اصلی معماری، ذکر شود.

توجه شود که این بخش در سطح معماری بیان شود و در سطح پیاده‌سازی توصیف نشود (جزئیات کلاس‌ها و فرم‌ها و ... بیان نشود بلکه کلیات طرح کلان معماری بیان شود).

۵. نمای استقرار (Physical یا Deployment)

در این بخش اطلاعات و توضیحات مربوط به نمای استقرار سامانه داده می‌شود. علاوه بر اطلاعات خواسته‌شده در بخش‌های زیر، نمایش یک Deployment Diagram نیز در این بخش مفید است، ولی کافی نیست (جدول زیر و توضیحات بعدی لازم است).

جدول استقرار

از دید سخت‌افزاری، فیزیکی و محیط بهره‌برداری (مثلاً دیتاسنتر)، فهرست سرورها و محیط‌های اجرایی (مانند ماشین‌های مجازی و فایروال‌ها و ...) مشخص شوند.

نکته: جزئیاتی مثل IP سرورها در این مستند جای نگیرند.

اسم (شناسه) ماشین	محل استقرار (دیتاسنتر)	توضیح کارکرد این ماشین	نوع ماشین	تعداد ماشین	تعداد و نوع پردازنده	حافظه اصلی	حافظه جانبی
-------------------	------------------------	------------------------	-----------	-------------	----------------------	------------	-------------

TB ۱	GB ۳۲	* ۲ vCorei	۳	Virtual Machine	دیتابیس اوراکل	دیتاسنتر توحید	VM-DB
TB ۲۰	۳۲۰ GB	CPUs ۲۵	۱۲۰	جمع کل			

ستون‌های جدول نمای استقرار:

- اسم ماشین: شناسه ماشین
- محل استقرار ماشین: نام دیتاسنتر
- توضیح کارکرد: شرح کوتاه مسئولیت این سرور در سیستم موردنظر. نوع سرور از قبیل Active/Passive یا Hot/Cold Spare بیان شود.
- نوع ماشین: ماشین مجازی یا سرور فیزیکی
- تعداد ماشین: تعداد ماشین‌های مورد استفاده از این جنس در سامانه موردنظر
- منابع مورد استفاده/مورد نیاز ماشین: پردازنده (CPU)، حافظه اصلی (RAM) و حافظه جانبی به ازای هر ماشین

توجه: مجموع کل منابع مورد استفاده (تعداد سرورها و مجموع حافظه و پردازنده‌ها) هم حتماً در آخرین سطر این جدول ذکر شود.

زیرساخت و مجازی سازی

در این بخش، به این موضوعات اشاره شود:

- در صورت استفاده از مجازی سازی، زیرساخت مجازی سازی (مثل ESXi و ...) معرفی شود.
- در صورت استفاده از Containerها، زیرساخت مدیریت containerها (مثل Docker و Kubernetes) توصیف شود.
- در صورت استفاده از Container Orchestration، روش کار کلاستر توصیف شود، مثلاً ساختار ماشین‌های Master و Worker در Kubernetes توصیف شوند.

فرایند کلان نصب

- «فرایند نصب» به صورت کلی توصیف شود: چگونه و با چه روال و ابزارهایی، نرم افزار در محیط عملیاتی نصب و بروزرسانی می شود؟

۶. نمای پردازش (Process)

این بخش شامل جدول پردازش ها و همچنین جدول ارتباطات بین پردازش ها است. هر پردازش، یک واحد اجرایی (Executable) است، مثلاً یک برنامه یا نخ (thread) اجرایی.

- نگاشت پردازش ها روی سرورها هم مشخص می شود: هر پردازش روی چه سروری اجرا می شود (نگاشت پردازش ها روی نمای استقرار در همان جدول اول نمایش داده می شود)
- ویژگی های ارتباطات بین پردازش ها ذکر شود: مثل پروتکل ارتباطی و ...

منظور از پردازش، برنامه های اجرایی است که در نهایت در سرورها یا محیط بهره برداری، به اجرا درخواهند آمد. مثال: برنامه tomcat، برنامه دیتابیس اوراکل، برنامه مانیتورینگ abc، عامل (agent) نصب و بروزرسانی xyz. ذکر همه پردازش های در حال اجرا روی سرورها مهم است.

نکته: در صورتی که معماری شما مبتنی بر Docker است، به جای پردازش ها بهتر است Docker ها را توصیف کنید. در این صورت برای معرفی هر Docker، توضیح کوتاهی درباره کارکرد هر Docker بدهید و مشخصاتی مثل Port و Volume و محدودیت پردازنده و حافظه و اطلاعات مهم موجود در Dockerfile ها را ذکر نمایید.

رفع سوء تفاهم: منظور از پردازش، فرایندها (business process) یا گردش کارهای سازمانی (workflow) نیست. بلکه درباره برنامه ها و حتی گاهی بندهای اجرایی است (processes and threads). یعنی آن چه بر روی سیستم عامل سرورها اجرا خواهد شد.

اطلاعات پردازش ها را در جدولی مشابه جدول زیر توصیف کنید.

توجه: در این بخش نیز مثل سایر بخش های این سند، بر اجزای داخلی سیستم تمرکز می شود. به عبارت دیگر سامانه های خارجی (سیستم های خارج از سیستم مورد بررسی) در فهرست پردازش ها ذکر نشود.

پردازش	ماشین/ماشین مجازی	توضیح کارکرد
--------	-------------------	--------------

پردازشی تامکت به عنوان application-server عمل می کند	ماشین مجازی ۱	Tomcat-server
دیتابیس اطلاعات تراکنشها	ماشین مجازی ۲	Oracle-DB
دیتابیس اطلاعات کاربران	ماشین مجازی ۲	MySQL-DB

همچنین ارتباطات بین پردازها، در جدولی مشابه جدول زیر و یا به صورت یک نمودار مناسب ارائه شود. سطرهای جدول زیر به عنوان نمونه و مثال ذکر شده اند. مثلاً اگر از رویکرد مایکروسرویس استفاده می کنید، می توانید ارتباط بین مایکروسرویس ها را در یک نمودار Microservices Diagram (و یا مشابه جدول زیر) توصیف کنید.

پردازش مبدأ	پردازه مقصد	پروتکل	فرکانس (تعداد) فراخوانی	توضیح
Tomcat-Server	Oracle-DB	SQL	۱۰ TPS	هدف ارتباط: دریافت اطلاعات تراکنشها
Tomcat-Server	ABC-Server	فراخوانی متد (RMI)	۱۰ request per day	هدف ارتباط: ارسال نوتیفیکیشن
ABC-Server	XYZ-Server	REST	۱۰ request per minute	هدف ارتباط: همگام سازی اطلاعات مربوط به حسابداری
...	...	SOAP	۱۰ per hour	

۷. نمای توسعه و پیاده سازی (Development یا Implementation)

این نما ساختار کدهای منبع را توصیف می کند. به عنوان مثال، بیان بسته ها و لایه های نرم افزاری که در پیاده سازی نرم افزار ایجاد شده اند.

این بخش فقط برای برنامه‌ها (کدها)یی که به صورت داخلی در تیم پروژه توسعه داده شدند ذکر شود. پروژه‌هایی مثل دیتابیس و وب‌سرور، که توسط تیم پروژه تولید نشده‌اند، در این بخش جایی ندارند. برای تکمیل این بخش، می‌توان با مراجعه به مخزن کد (مثلاً git) لیست پروژه‌ها یا ماژول‌ها را استخراج کرد.

این بخش حداقل شامل یک جدول خواهد بود که در هر سطر آن به یک پروژه (Source Project) یا یک بسته مهم و کلان (Package) یا یک ماژول (Module) اختصاص دارد. (در صورتی که ساختار کد بزرگ و پیچیده و مشتمل بر چندین مولفه است در جدول، بسته‌های آن را به تفکیک توضیح بدهید). در هر سطر جدول به این موارد اشاره می‌شود:

- عنوان پروژه یا بسته یا ماژول
- توضیح
- اهم فناوری‌ها و کتابخانه‌های مورد استفاده (با ذکر نسخه)
- وابستگی‌ها: سایر ماژول‌هایی که به آن‌ها وابسته است.

اگر از رویکرد میکروسرویس استفاده می‌کنید، هر یک از میکروسرویس‌ها را به عنوان یک ماژول توصیف کنید.

ماژول / پروژه / بسته	توضیح	فناوری‌ها	وابستگی‌ها
UI	پیاده‌سازی بخش front-end	HTML, JS, CSS, Angular	-
Business-Services	پیاده‌سازی سرویس‌ها	Java, Spring	Project-core
Project-core	مؤلفه‌های زیرساختی و Reusable	Java, Spring	-

۸. نماهای متقاطع (اختیاری)

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۹. نحوه تحقق موارد کاربرد کلیدی

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۰. نمای تست

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۱. مدیریت لاگ

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۲. پایش (Monitoring)

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۳. نمای داده

در این بخش، تصمیمات مهمی که در زمینه داده‌های سامانه گرفته شده ذکر می‌شود. به عنوان مثال، نحوه ذخیره داده‌ها، نوع پایگاه داده‌های مورد استفاده (./sql/nosql)، مدیریت داده‌ها و فراداده‌ها، الگوها و تکنیک‌های اصلی در لایه داده (مثل CQRS و محل استفاده از Cache)، فرایندهای پشتیبان‌گیری و بازیافت داده‌ها، امنیت داده‌ها و ساختار داده‌های کلیدی سیستم.

به طور کلی، آنچه از منظر معماری نرم‌افزار به یکپارچه‌سازی داده‌ها و مهاجرت داده‌ها مربوط است، در این بخش ذکر شود.

۱۴. نمای گزارش‌گیری

در این نما، تصمیمات اصلی فنی برای تسهیل و تسریع گزارش‌گیری (Reporting) در سامانه نرم‌افزاری ذکر شود. در سامانه‌های بزرگ نرم‌افزاری، امکاناتی برای گزارش‌گیری داده‌ها و اطلاعات، از جمله گزارش‌های

مدیریتی، تحلیلی و گزارش‌های کارشناسی قرار می‌گیرد. گاهی از ابزارها، الگوها و فناوری‌های مشخصی به این منظور استفاده می‌شود که در این بخش به آنها اشاره شود. برخی از مثال‌ها عبارتند از:

OLAP, Data-warehousing, Report Engines, Report Generators, Report Designers, Materialized View, Denormalization, Power BI, Jasper Reports, BIRT, ...

در این بخش، طراحی کلان امکاناتی که برای گزارش‌گیری تعبیه شده است، ذکر شود. با کمک محتوای این فصل، به سؤالات و دغدغه‌هایی مثل موارد زیر پاسخ داده می‌شود:

- در این سامانه نرم‌افزاری، چه نوع گزارش‌هایی قابل دریافت است؟ این گزارش‌ها توسط چه کاربران/نقش‌هایی قابل دریافت هستند؟
- چه ابزارها یا فناوری‌هایی برای تولید گزارش استفاده شده‌اند؟
- امکان طراحی گزارش (report designer) توسط کاربر یا مدیر وجود دارد؟
- گزارش‌ها در چه قالب‌ها و فرمت‌هایی قابل مشاهده است؟

۱۵. الگوها و تکنیک‌های مورد استفاده

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۶. ابزارها و فناوری‌ها

در این بخش، مجموعه ابزارها، استانداردها، الگوها و فناوری‌های مورد استفاده در توسعه محصول که از جنبه معماری اهمیت دارند، مرور می‌شود. برای هر ابزار یا استاندارد، در صورت امکان نسخه (version) و جایگاه استفاده هم ذکر شود. در واقع، جایگاه همه یا اکثر این موارد در فصل‌های قبلی تبیین شده است و در این فصل، فهرستی از این موارد به صورت تجمیعی و متمرکز ذکر می‌شود. این فصل فقط شامل یک جدول است شامل سه ستون است:

۱. نام یک ابزار/استاندارد/الگو/فناوری
۲. نسخه مورد استفاده
۳. جایگاه استفاده (معنی یا تعریف فناوری ذکر نشود، بلکه در حداکثر یک جمله و یا حداکثر یک خط، جایگاه آن ذکر شود)

یک فهرست فرضی به عنوان نمونه در ادامه آمده است:

ابزار/استاندارد/فناوری/الگو	نسخه	جایگاه	محل استفاده (پروژه/مؤلفه/بسته)
Java	۱۰	زبان برنامه‌نویسی سرورساید	
Spring	۵	IOC, AOP	
JPA	۲.۲	ORM	
Hibernate	۵.۴	ORM Implementation	
Angular	۷	پیاده‌سازی کلاینت‌ساید	
CQRS	-	جهت افزایش سرعت درج و بهبود مقیاس‌پذیری	
Redis	۵	در سرورهای شماره ۱ و ۴	
J4SLF	۱.۷	-	
Tomcat	۹	در سرورهای شما ۱ و ۲ و ۴	
ActiveMQ	۵.۱۵	برای مدیریت صف ارسال پیامک	
Docker	۱۸	...	
JUnit	۵	-	

۱۷. مهم‌ترین تصمیمات معماری

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۸. ریسک‌ها و بدهی‌های فنی

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۱۹. بهبودهای پیشنهادی در آینده

برای مشاهده محتوای این بخش به نسخه کامل قالب سند معماری مراجعه نمایید.

۲۰. سایر نماهای مفید/لازم

در این بخش، سایر اطلاعاتی که در سطح معماری از نظر معمار یا تیم معماری نرم‌افزار اهمیت دارند ولی در بخش‌های قبلی پوشش داده نشده‌اند، اضافه گردد. توجه شود که اگر این اطلاعات در قالب نماها و بخش‌های فوق قابل بیان است، حتماً باید در همان بخش‌ها ذکر شود. به عبارت دیگر، حضور این بخش، دلیلی برای ناقص شدن بخش‌های فوق نخواهد بود. بلکه فصل‌های قبلی باید به طور کامل و دقیق پوشش داده شوند و اگر چیزی در قالب سند گنجانده نشده است، در این بخش اضافه شود. اگر از نظر معمار نرم‌افزار سیستم موردنظر، این بخش لازم نیست، این بخش خالی گذاشته شود (تکمیل این بخش، اختیاری است)

ترتیب پیشنهادی برای تکمیل سند معماری

پیشنهاد می‌کنیم سند معماری با یک فرایند Iterative & incremental تکمیل شود. لزومی ندارد که فصل‌های این سند به ترتیب فوق تکمیل شوند. گام‌های پیشنهادی:

۱. چرخه اول: نگارش فضای مسأله.

در این چرخه، یک دور (چرخه) تلاش شود و نیمه اول سند معماری که شامل «فضای مسأله» است، مستند شود. فضای مسأله شامل فصل‌های «مقدمه»، «[نمای موارد کاربرد](#)» و «ویژگی‌های کیفی» است. پس از این چرخه، صورت مسأله و نیازمندی‌های کلان نرم‌افزار مشخص می‌شوند. ترتیب پیشنهادی:

○ مقدمه

○ نمای موارد کاربرد

○ ویژگی‌های کیفی

۲. چرخه دوم: توصیف طراحی کلان سامانه نرم‌افزاری، شامل فصلهای «نمای منطقی» و «ابزارها و فناوری‌ها». ترتیب پیشنهادی :

○ نمای منطقی

○ ابزارها و فناوری‌ها

○ بازنگری فصلهایی که قبلاً نگارش شده

۳. چرخه سوم: تکمیل سایر فصلهای اصلی جهت توصیف معماری کلان. ترتیب پیشنهادی :

○ نمای پردازش

○ نمای استقرار

○ نمای توسعه و پیاده‌سازی

○ نماهای متقاطع (در صورت نیاز)

○ بازنگری فصلهایی که قبلاً نگارش شده

۴. چرخه چهارم. ترتیب پیشنهادی :

○ فصل نحوه تحقق موارد کاربرد کلیدی

○ نمای تست

○ نمای داده

○ الگوها و تکنیک‌های مورد استفاده

○ ریسک‌ها و بدهی‌های فنی

○ بازنگری فصلهایی که قبلاً نگارش شده

۵. چرخه پنجم. ترتیب پیشنهادی :

○ سایر فصلهای باقی‌مانده

○ بازنگری فصلهایی که قبلاً نگارش شده