

سند معماری نرم افزار سامانه

مدیریت کارت

SAD of Card Management System

نسخه: ۱.۱

فهرست

۳	مقدمه.....	۱
۳	حدوده.....	۲
۳	تعریف، اختصارات و واژه نامه.....	۳
۴	اهداف معماری.....	۴
۴	حدودیت.....	۵
۵	استاندارد ها.....	۶
۵	مراجع.....	۷
۶	نمای موارد کاربرد.....	۸
۷	ویژگی های کیفی.....	۹
۸	نمای منطقی.....	۱۰
۹	نمای استقرار.....	۱۱
۹	جدول استقرار.....	۱۱/۱
۹	زیرساخت.....	۱۱/۲
۹	فرایند نصب.....	۱۱/۳
۱۰	نمای پردازه.....	۱۲
۱۰	نمای پیاده سازی.....	۱۳
۱۲	نمای تست.....	۱۴
۱۲	نمای لاک.....	۱۵
۱۲	نمای پایش.....	۱۶
۱۲	نمای داده.....	۱۷
۱۳	الگوها و تکنیک ها	۱۸
۱۴	ابزارها و فناوری ها.....	۱۹
۱۴	تصمیمات مهم معماري.....	۲۰
۱۵	ریسک ها و بدهی های فنی	۲۱

۱ مقدمه

با توجه به اهمیت نقش مدیریت مالی در تبادلات اقتصادی نیاز است سهولت در نقل و انتقالات مالی برای مشتریان فراهم شود. کارت های بانکی برای شهروندان امکانات مناسبی را جهت امور مالی که بخش مهمی از آن پرداخت های کارتی است فراهم کرده است. جهت مدیریت

انتقالات مالی به صورتی سیستمی، سامانه مدیریت کارت^۱ طراحی و پیاده سازی شده است. این سامانه با خدماتی که به مشتریان می دهد امکان ثبت مشتری، ثبت کارت و بازیابی کارت های مشتری را فراهم نموده است.

سند پیش رو که به عنوان سند معماری نرم افزار شناخته میشود با همکاری معمار و مدیرفنی سامانه مدیریت کارت تهییه شده است و تلاش شده است تا ابتدا با تعریف مساله، محدوده و محدودیت های آن، معماری در نظرگرفته شده برای این سامانه به شکلی واضح و کامل تشریح شده و قابل استفاده برای تمامی بخش های تاثیرپذیر یا تاثیرگذار از این سامانه باشد. باید درنظر داشت که این سند در مدت زمان توسعه این سامانه به روز میشود تا همواره بتواند به عنوان یک مرجع مطمئن برای مراجعات تمامی افراد ذیفع باشد.

۲ محدوده^۲

این سامانه نیازهای مربوط به ارائه گزارش های کارت های مشتری را در بخش انتقال پول بانک را پوشش میدهد.

۳ تعاریف، اختصارات و واژه نامه

کاربر: استفاده کننده نهایی سامانه که می تواند کاربری از سامانه انتقال پول باشد. همچنین مشتری نیز می تواند به طور مستقیم یا غیرمستقیم از سرویس های این سامانه استفاده کند.

CMS : نام مخفف سامانه مدیریت کارت

۴ اهداف معماری

در انتخاب و طراحی معماری مناسب برای این سامانه با ارزیابی نیازمندی ها، پیچیدگی فرایندها و محدودیت ها، معماران هدف های زیر را مدنظر داشته اند:

^۱ Card Management System

^۲ scope

--	--

- انعطاف پذیری^۳ در برابر تغییرات نیازمندی، ابزار و تکنولوژی
- ارائه سرویس به صورت چندنخی^۴ و موازی^۵
- سادگی و پرهیز از پیچیدگی
- مقیاس پذیری^۶
- نگهداری^۷ بلندمدت با کمترین هزینه و خطر^۸

۵ محدودیت^۹

همانطور که بیشتر گفته شد سامانه مدیریت کارت متاثر از بخش های فنی وغیر فنی دیگر است و خود نیز تاثیراتی را در این بخش ها دارد. همین مساله محدودیت هایی را برای طراحی معماری ایجاد کرده است که منجر به تغییراتی در ابزارها ، تکنولوژی های استفاده شده و ارتباطات با بخش ها و سامانه های دیگر شده است. محدودیت هایی که معماران با آن مواجه بودند در ادامه ذکر شده اند که مatasفانه برخی از بخش های معماری را از حالت بهینه خود خارج کرده است. امید است در آینده بهبودهایی در این تاثیرات انجام شود.

- تاکید بر همزمان بودن تمامی فرایندهایی که در آن کاربر مشارکت دارد
- تمامی api های سامانه باید از نوع Rest WebService باشند (به علت عدم سازگاری سامانه های دیگر با روش های ارتباطی دیگر)
- عدم پایداری سرویس های بیرونی. این سرویس های به علت بار ترافیکی زیاد در زمانهای زیادی قادر به پاسخگویی نیستند
- خروجی تمامی سرویس ها باید فرمت JSON را داشته باشند

^۳ Resilience

^۴ Multi-threading

^۵ Parallel

^۶ Scalability

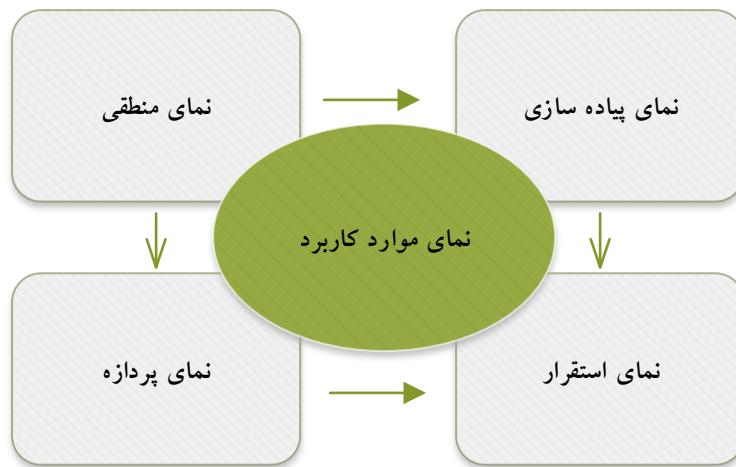
^۷ Maintenance

^۸ Risk

^۹ Constraint

۶ استاندارد ها

در تهییه این سند برای توصیف و ارائه معماری نرم افزار، از مدل $1+4$ استفاده شده است. که توضیح درباره این مدل خارج از محدوده این سند است. اما به طور خلاصه به شرح این مدل در این سند پرداخته میشود. این مدل شامل 5 نمای^{۱۰} است که 4 نمای پیاده سازی، منطقی، پردازه و استقرار در خدمت نمای موارد کاربرد هستند. با توجه به بزرگی سامانه مدیریت کارت، از نمای دیگری مانند نمای تست، نمای لاغ و نمای داده برای توصیف معماری نرم افزار این سامانه کمک گرفته شده است.



۷ مراجع

۸ نمای موارد کاربرد^{۱۱}

شناسه ^{۱۲}	کنشگر ^{۱۳}	موردنکاربرد ^{۱۴}	شرح
---------------------	---------------------	---------------------------	-----

^{۱۰} View

این نما به نام کارکردهای اصلی برنامه نیز شناخته می شود^{۱۱}

^{۱۲} Actor

^{۱۳} Usecase

کاربر سامانه	ثبت مشتری	ثبت مشتری	۱-۸
کاربر سامانه	ثبت کارت جدید برای مشتری	ثبت کارت جدید برای مشتری	۲-۸
کاربر سامانه یا مشتری	مشاهده کارت های یک مشتری	مشاهده کارت های یک مشتری	۳-۸
سیستم	چاپ وضعیت حافظه	چاپ وضعیت حافظه	۴-۸
سیستم	ثبت رخدادها	ثبت رخدادها	۵-۸

۹ ویژگی های کیفی^{۱۵}

ویژگی کیفی	معیار ارزیابی	اندازه / محدوده مطلوب
امنیت	جلوگیری از افشای اطلاعات مهم و حیاتی	%۱۰۰
تست پذیری ^{۱۶}	تست های unit test در کدها سامانه	%۵۰
تست پذیری	اجرای تست های در هر بار نسخه گذاری سامانه	۱۰۰%

^{۱۴} log

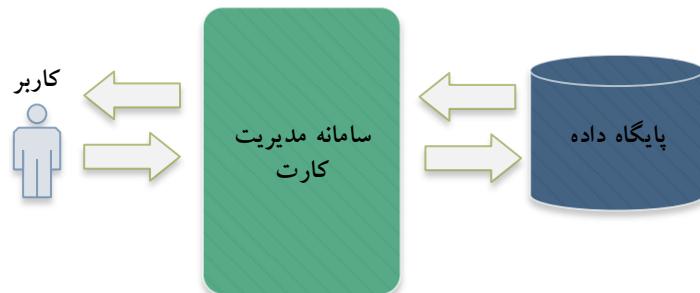
^{۱۵} Quality Attributes

^{۱۶} Testability

مقیاس پذیری ^{۱۷}	تست بار	نامشخص
قابل حمل بودن ^{۱۸}	نصب سامانه در سرور ویندوز و لینوکس	% ۱۰۰
بهره وری ^{۱۹}	تعداد تراکنش های انجام شده در ثانیه ^{۲۰}	نامشخص
بهره وری	تعداد کاربرانی که می توانند همزمان از سیستم استفاده کنند	نامشخص
سازگاری ^{۲۱}	تغییر نسخه ابزارهای استفاده شده	۱۰۰٪
قابلیت اطمینان ^{۲۲}	زمان پاسخگویی سرویس ها	نامشخص
دسترس پذیری ^{۲۳}	زمان در دسترس بودن سامانه بعد از نصب نسخه	حداکثر ۳ ثانیه
دسترسی پذیری	زمان در دسترس بودن سامانه برای کاربر	حداکثر ۳ ثانیه
دسترسی پذیری	زمان در دسترس بودن سامانه	۱۰۰٪

۱۰ نمای منطقی

شکل زیر سامانه مدیریت کارت را به صورت کلی نمایش میدهد:



^{۱۷} Scalability

^{۱۸} Portability

^{۱۹} Efficiency

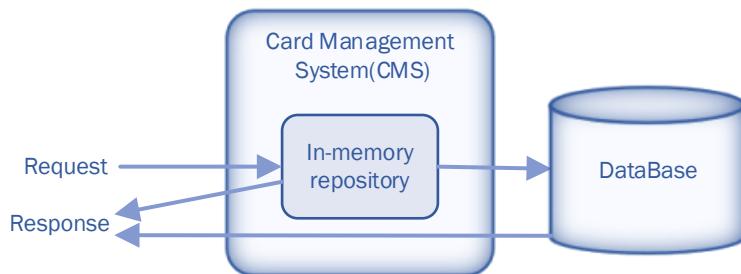
^{۲۰} tps

^{۲۱} Compatibility

^{۲۲} Reliability

^{۲۳} Availability

سامانه مدیریت کارت شامل مولفه هایی است که این مولفه ها در شکل زیر مشخص شده اند. هر یک از این مولفه ها به اختصار در ادامه معرفی می شوند.



In-Memory Repository: حافظه ایی که مشخصات مشتری را به همراه مشخصات کارت های فعال او را نگهداری میکند. این فضای ذخیره سازی موافق بوده و بخشی از حافظه سیستم را برای خود نگه داشته است.

Database: پایگاه داده که تمامی اطلاعات مربوط به کارت و مشتری را ذخیره میکند.

۱۱ نمای استقرار^{۲۴}

معماری در نظر گرفته شده برای سامانه مدیریت کارت، محیط نصب و اسقرار سامانه را تحت تاثیر قرار میدهد. از این رو در ادامه با مشخصات محیط سخت افزاری نصب و استقرار این سامانه آشنا میشویم.

۱۱/۱ جدول استقرار

شناسه ماشین	توضیح	نوع ماشین	تعداد	تعداد و نوع	حافظه	حافظه	حافظه
	دیتابیس	Virtual Machine	3	پردازنده	ماشین	اصلی	جانبی
VM-LNDBB				Corei7*2		64GB	1 TB
VM-LNDGW	CMS	Virtual Machine	3	Corei7*2	ماشین	64GB	64GB

در جدول زیر ارتباط بین ماشین ها با یکدیگر مشخص شده است:

ماشین مبداء	ماشین مقصد	پروتکل	توضیح	میانگین تعداد فرآخوانی	نحوه:
انجام درج و واکشی اطلاعات کارت و مشتری	۵۰ درخواست هر ثانیه	Sql		VM-LNDB	VM-LNDGW

۱۱/۲ زیرساخت

<توسط تیم عملیات تکمیل شود>

۱۱/۳ فرایند نصب

<توسط تیم عملیات تکمیل شود>

۱۲ نمای پردازه^{۲۵}

برنامه هایی که در زمان اجرای سامانه مدیریت کارت در حال اجرا هستند در جدول زیر مشخص شده اند:

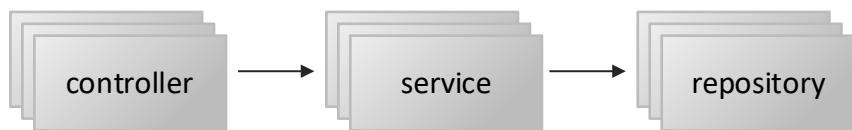
پردازه	Server/vm/container	توضیح
دیتابیس	VM-LNDB	برنامه پایگاه داده
CMS	VM-LNDGW	سامانه مدیریت کارت

۱۳ نمای پیاده سازی

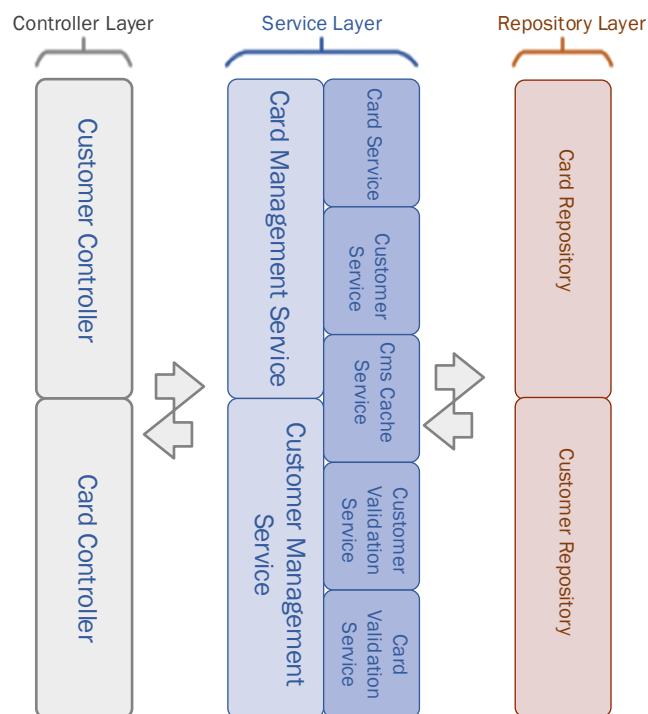
همانطور که در بخش نمای منطقی گفته شد، سرویس های سامانه مدیریت کارت در مولفه CMS قرار دارد. در این بخش به لایه های پیاده سازی این مولفه به همراه فریمورک ها و کتابخانه های مهم استفاده شده در آنها می پردازیم.

در طراحی مولفه CMS از معماری سه لایه^{۲۶} استفاده شده است. تشریح این مدل خارج از محدوده این مستند است و تنها به نمایش یک شکل

از این مدل اکتفا میکنیم.



در شکل زیر لایه بندی سامانه با جزئیات بیشتری تشریح شده است:



^{۲۶} 3-layer

در جدول زیر تکنولوژی های استفاده شده در مولفه ها معماری معرفی شده است:

عنوان مولفه	فناوری های استفاده شده	شرح مولفه
CMS	<ul style="list-style-type: none"> • Java 17 • Spring boot • Spring project(data-jpa,web,test) • springdoc-openapi • h2database 	<p>ارائه سرویس های مربوط به ثبت مشتری، ثبت کارت و بازیابی لیست کارت های مشتری</p>

۱۴ نمای تست^{۴۷}

به دلیل اهمیت تست در سامانه مدیریت کارت، زیرساخت پیاده سازی تست در معماری آن دیده شده است. در جدول زیر همه انواع تست هایی که برای سامانه در نظر گرفته شده است به همراه ابزارها و فریمورک های مورد استفاده ذکر شده است:

عنوان تست	فناوری مورد استفاده
Unit Test	<ul style="list-style-type: none"> • Junit • Mocito

۱۵ نمای لاغ^{۲۸}

ثبت تمامی وقایق سامانه از جهت تحت نظر داشتن وضعیت سامانه و رفع خطا در سریعترین زمان ممکن بسیار حائز اهمیت است. بنابراین

در سامانه مدیریت کارت، تمامی رخدادها در فایل های لاغ ذخیره میشوند. سطوح این لاغ ها در سه سطح **Error** ، **Debug** و **Info** هستند.

۱۶ نمای پایش^{۲۹}

«این بخش توسط تیم عملیات تکمیل شود».

۱۷ نمای داده^{۳۰}

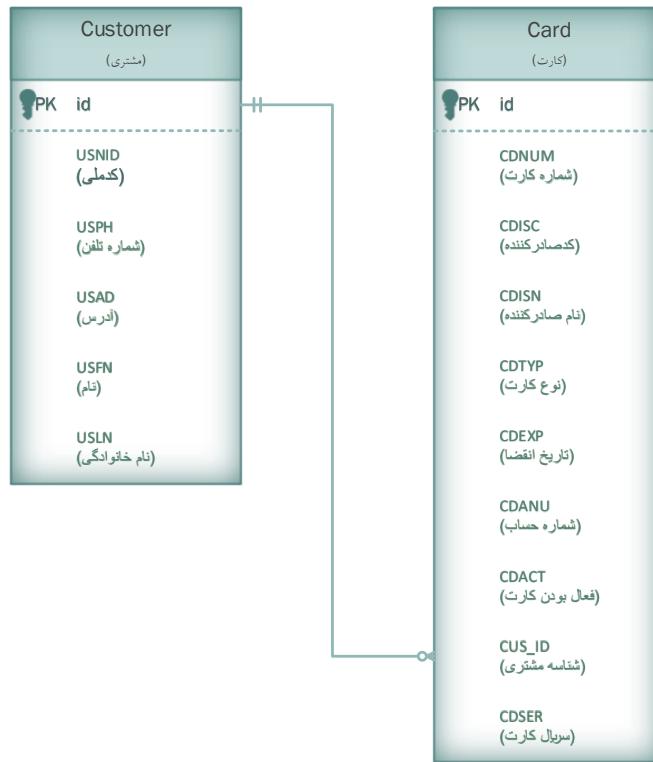
در سامانه مدیریت کارت از پایگاه داده **h2** استفاده شده است. جهت جلوگیری از بار زیاد برروی این پایگاه داده از حافظه **In-memory** در

داخل مولفه **CMS** استفاده شده است. یک بخش کوچک از ارتباط جداول این سامانه در شکل زیر نمایش داده شده است:

^{۲۸} Log view

^{۲۹} Monitoring view

^{۳۰} Data view



۱۸ الگوهای و تکنیک ها

شرح	تکنیک/ابزار	ویژگی کیفی
مراجعةه به نمای منطقی و پیاده سازی	عدم لایک گزاری داده های حساس	امنیت
مراجعةه به نمای پیاده سازی	تست unit test	تست پذیری
مراجعةه به نمای منطقی	اجرای هر باره تست ها در زمان نسخه گذاری	تست پذیری
مراجعةه به نمای پیاده سازی	توسعه سامانه با حداقل تاثیرات با استفاده از معماری سه لایه	مقیاس پذیری

۱۹ ابزارها و فناوری ها

عنوان	نسخه	توضیح
Java	۱۷	توسعه مولفه ها
Spring boot	۳.۵.۹	فریمورک مولفه ها
hibernate	۶.۶.۳۹.Final	توسعه مولفه ها
springdoc	۲۸.۵	مستندسازی برای سرویس های ارائه شده توسط مولفه ها

۲۰ تصمیمات مهم معماری

تصمیم معماری ^{۳۱}	گزینه های دیگر ^{۳۲}	دلیل این تصمیم ^{۳۳}	تاریخچه این تصمیم ^{۳۴}
استفاده از حافظه داخلی	redis	استفاده از کارتهای فعال کاربر با سرعت بالاتری انجام میشود	از ابتدای توسعه سامانه
تمامی سرویس های ارائه شده به صورت rest هستند	gRpc	به دلیل آنکه سامانه های دیگر که این سامانه به آنها سرویس میدهد هنوز زیرساخت لازم برای استفاده از gRpc را ندارند	از ابتدای توسعه سامانه
استفاده از پایگاه داده h2	دیتابیس اوراکل	رایگان بودن و سادگی	از ابتدای توسعه سامانه
استفاده از معماری سه لایه	DDD	به دلیل عدم دانش و تجربه از DDD	از ابتدای توسعه سامانه

^{۳۱} Architectural Decision

^{۳۲} Alternatives

^{۳۳} Reason

^{۳۴} History

از ابتدای توسعه سامانه	به دلیل داشتن کنترل بیشتر برروی مدیریت سرویس ها	استفاده از لایه hibernate	استفاده لایه های مختلف در لایه سرویس جهت مدیریت همزمانی درخواست ها
------------------------	---	---------------------------	--

۲۱ ریسک ها و بدھی های فنی ^{۳۵}

ریسک/بدھی فنی	توضیح	اولویت	احتمال رخداد	تأثیر رخداد
یکی بودن پایگاه داده های واکشی و درج داده ها	در حال حاضر پایگاه داده های واکشی و درج داده ها	بالا	بعد از رسیدن تعداد کارت ها به ۵۰۰ هزار، کندی در واکشی داده ها مشاهده خواهد شد.	باعث مشاهده کندی در زمان واکشی و همینطور عدم سریع این جدا سازی زمان تغییرات در سمت پیاده سازی را بیشتر میکند
پیاده سازی زیرساخت i18n	در حال حاضر عبارت هایی به زبان فارسی در پروژه وجود دارد که باید همه آنها از فایل هایی که بر اساس زبان مورد نظر جدا شده اند خوانده شود	بالا	عبارت های فارسی در کدها باعث ایجاد مشکل در سیستم های مختلف میشود	اشتباه نمایش داده می شود

سامانه هایی که مجاز به استفاده از این سرویس ها نیستند می توانند این سرویس ها را فراخوانی کنند	فراخوانی سرویس ها برای همه آزاد است	بالا	در حال حاضر تمامی سرویس ها حتی با درسترسی های غیر مجاز قابل فراخوانی هستند	استفاده از توکن jwt برای امن سازی سرویس ها
ممکن است که مقایبرت در داده ها پیش آید	ریسک همزمان شدن درخواست ها بالاست	بالا	بررسی کامل اینکه برنامه tread safe است نیاز به تست های بیشتری دارد که انجام نشده است	انجام تست load جهت بررسی همزمانی درخواست ها و thread safe بودن برنامه