پروژه در سه سرویس جداگانه مدیریت کارت ، عملیات پرداخت و ارسال پیام پیاده سازی شده است

تمام تنطیمات پروژه به صورت جداگانه در پوشه config وجود دارد و قابل تغییر است

برای اجرا در محیط عملاتی لازم است پوشه config در کنار پروژه وجود داشته باشد ، همچنین پوشه log میبایست توسط سرویس پرداخت و ارسال پیام قابل رویت باشد

با توجه به حجم بالای دیتا در جدول تراکنش بخش هایی از پردازش داده ها برای آماده سازی خروجی مناسب در برنامه و به وسیله streaming انجام شده است

با توجه به Tolerant Fault بودن عملیات ارسال پیام و ارسال حتمی پیام ، چند راهکار از جمله استفاده از معماری Aspect Oriented ، استفاده از Database Trigger ، استفاده از فیلد وضعیت تراکنش بررسی شد ولی با توجه به مفروضات پروژه این عملیات از طریق پردازش بر روی لاگ تراکنش انجام میشود.

نحوه کار به این صورت است که بعد از هر تراکنش اطلاعات آن در فایل log ذخیره میشود و همزمان با استفاده از logstash اطلاعات خوانده شده و به rabbitmq ارسال میشود ، سرویس ارسال پیام نیز اطلاعات را از روی rabbitmq برداشته و در صورتی که موفق به ارسال شود آن را از صف خارج میکند.

در بررسی های انجام شده بهترین روش بررسی لاگ های تولید شده از commit های دیتابیس بود که با توجه به زمان و عدم شناخت از ابزارهای مناسب این روش پیاده سازی نشد

همچنین در محیط توسعه در صورتی که پروژه با profile=LocalApi ( موجود در فایل های config) اجرا شود ، سرور های WireMockh اجرا شده که شبیه ساز api های بانک و پیامک هستند و در غیر این صورت api ها به سرور های اصلی متصل میشود

در پیاده سازی این پروژه مباحث authentication و authorization و encryption در نظر گرفته نشده است

اطلاعات مربوط به user در این پروژه وجود ندارد و صرفا از userId به عنوان شناسه user استفاده شده است