



دانشکده مهندسی  
کامپیوتر و فناوری اطلاعات

1399/8/18



## تمرین سوم



### مهندسی نرم افزار 2

#### گروه {۶}

##### اعضاء گروه:

- 1) علی خرمی پور
- 2) امیررضا شیرمست
- 3) علیرضا صدیقی مقدم



معماری نرم افزار پروژه گروهتان را معرفی کرده و توضیح دهید.

• ساختار: مولفه های اصلی، چیدمان و رابطه میان آنها

زیرسیستم های مورد بحث:

- A. ثبت سفارش
- B. انبارداری
- C. مدیریت منابع
- D. پرداخت
- E. مدیریت اطلاعات

A. ثبت سفارش:

این قسمت به صورت یک برنامه تعاملی است که کاربران با آن ارتباط برقرار می کنند (سفارش مورد نظر خود را انتخاب می کنند). به نوعی می توان این قسمت از سیستم را به سه بخش تقسیم کرد. بخش اول وظیفه دریافت اطلاعات از بخش مدیریت اطلاعات را دارد. بخش دوم وظیفه نشان دادن منو بر اساس اطلاعات دریافتی را بر عهده دارد. بخش سوم قابلیت انتخاب از منو را به کاربر می دهد. بر اساس این مسائل الگوی معماری MVC (Model-View-Controller) مناسب است اما برای پردازش سفارشیهای چند کاربر به صورت همزمان لازم است که از مدل PAC (Presentation-Abstraction-Control) استفاده کنیم.

B. انبارداری:

این قسمت از سیستم مسئول کنترل موجودی کالاها، تاریخهای انقضا، درخواست خرید به مدیریت منابع و غیره است. به دلیل سرویس های مختلفی که این سیستم ارائه می دهد، الگوی معماری Microservice مناسب است.

C. مدیریت منابع:

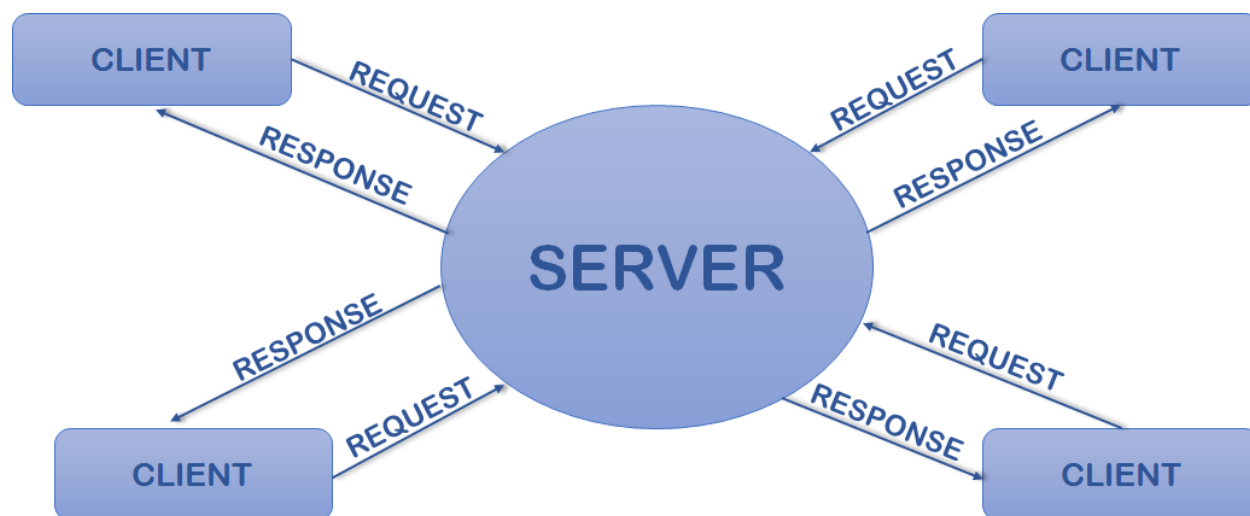


در این قسمت از سیستم باید درخواست خرید کالاهای مورد نیاز از سمت سیستم به سمت شرکت‌های تامین‌کننده فرستاده شود و تاییدیه از سمت آن‌ها دریافت شود. پس معماری مناسب این قسمت، client-server است.

D. پرداخت:

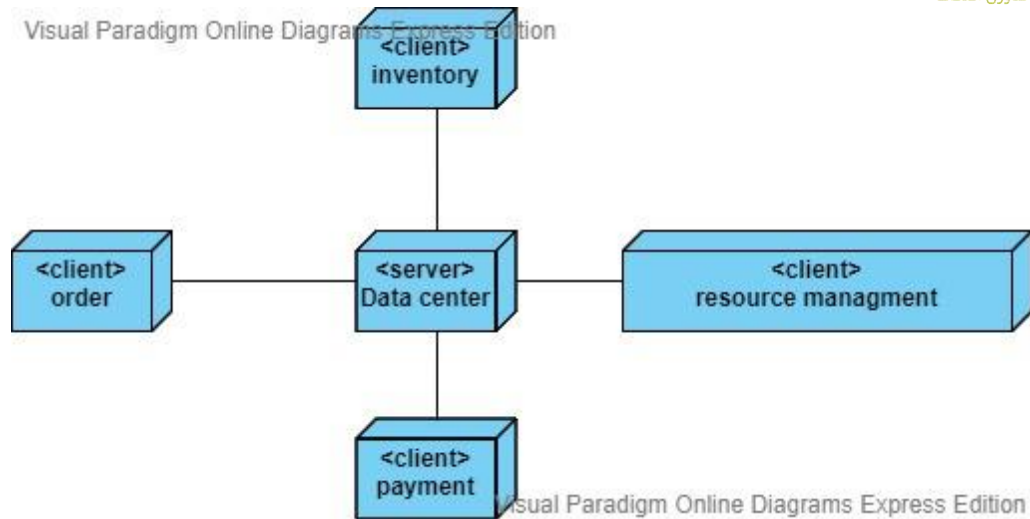
در این قسمت از سیستم باید درخواست پرداخت از سمت کاربر به سمت ارائه دهنده سرویس فرستاده شود تا عملیات پرداخت از سمت سرور انجام بگیرد. پس معماری مناسب این قسمت، client-server است.

Client-Server: در این مدل از یک سرویس دهنده و یک سرویس گیرنده در شبکه استفاده می‌گردد و منطق برنامه بین دو محل فیزیکی توزیع می‌گردد.



E. مدیریت اطلاعات:

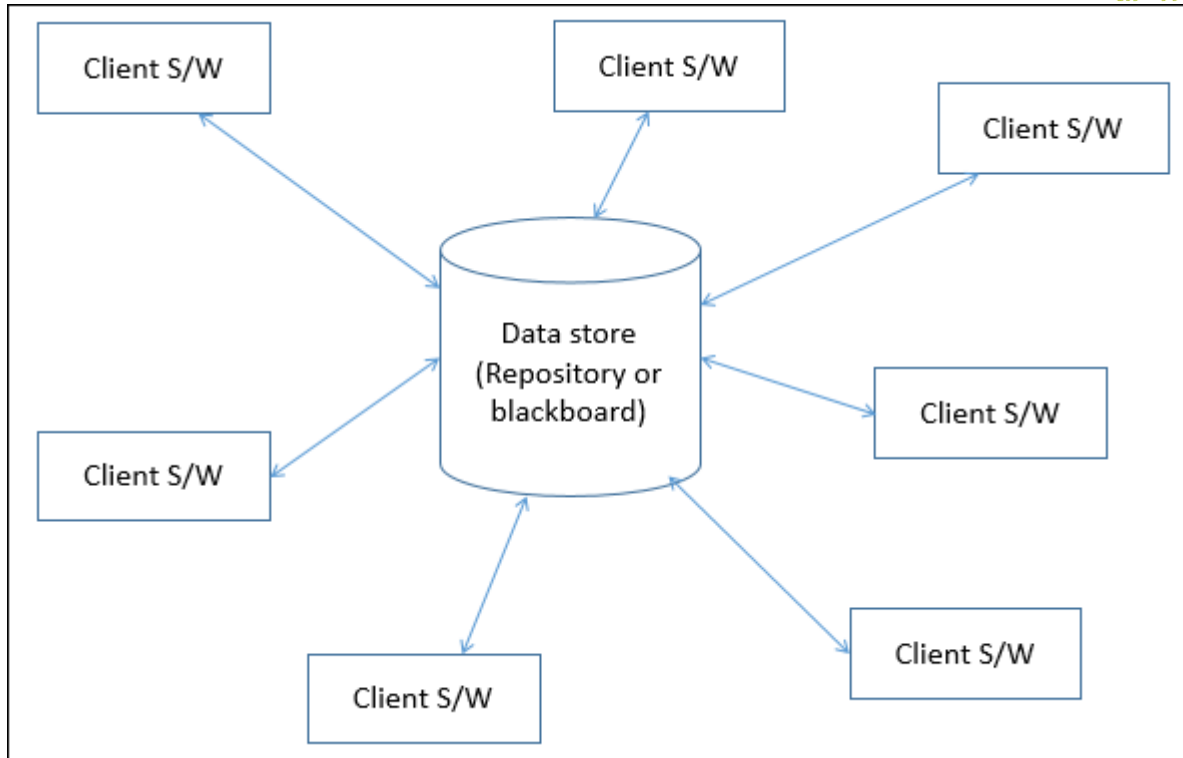
در این قسمت از سیستم باید اطلاعات بر اساس نوع درخواست کننده ارائه شود. که بر این اساس مدل Broker مناسب این قسمت است.



• سبک‌ها و الگوهای استفاده شده

### انتخاب و استدلال استایل‌های کاندید برای معماری سیستم

Data-centered: در این استایل ما به یک data store نیاز داریم، که component های مستقل سیستم با توجه به الگویمان با data store ارتباط برقرار می‌کند. از مزایای این استایل قابلیت scalability است زیرا client ها مستقل هستند و از معایب آن این است که client ها به data store وابسته هستند و بدون آن نمی‌توانند کاری انجام دهند.

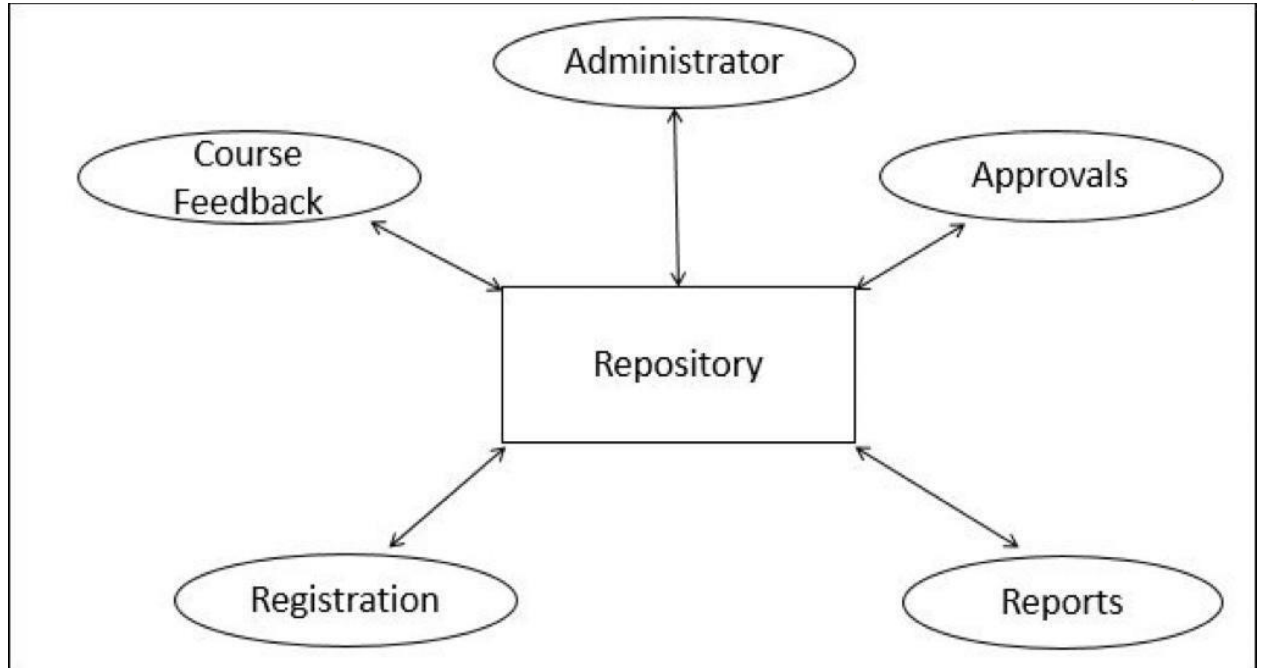


با توجه به نیازمندی‌های استخراج شده سیستم، وابستگی اجزا به database و استقلال client ها برداشت می‌شود و به همین دلیل data-centered یکی از کاندیدهای معماری سیستم ما است.

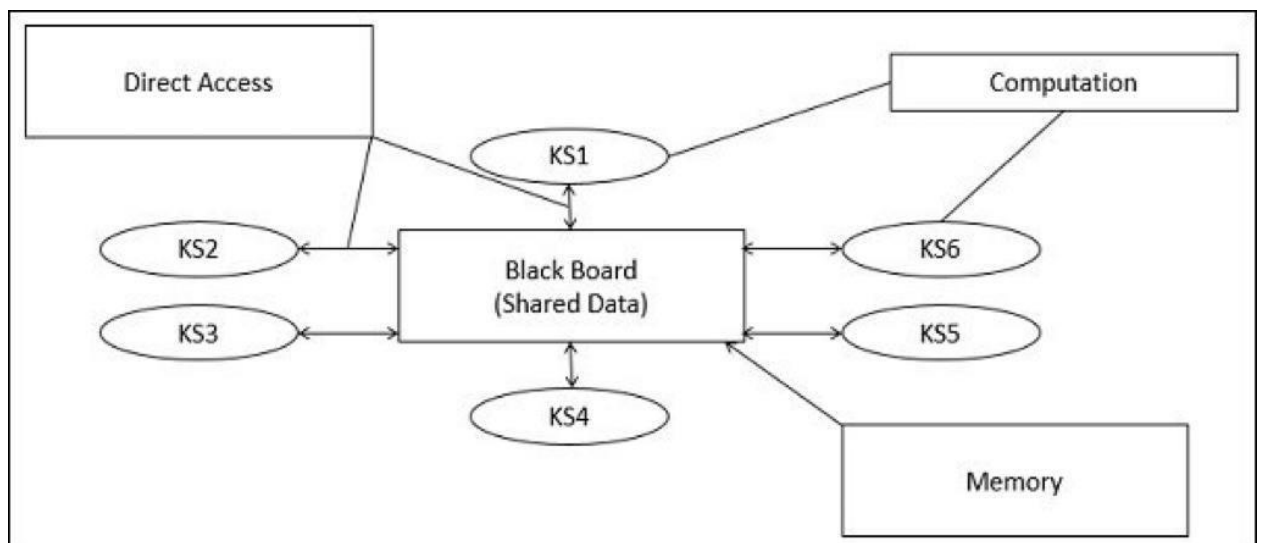
• تاکتیک های معماری استفاده شده برای دستیابی به ویژگی های کیفی

استایل معماری موردنظر ما data-centered:

Repository: در این pattern یک repository به صورت passive قرار دارد؛ یعنی به تنهایی کاری انجام نمی‌دهد و client های متصل به آن به صورت active درخواست می‌دهند. به عنوان مثال insert



Blackboard: در این pattern یک Blackboard به صورت active قرار دارد؛ یعنی اجزای متصل به آن بر اساس تغییرات blackboard فعالیت می کنند. مثال sound/image recognition





برای object-oriented الگوی انتخابی ما PAC و برای data-centered, blackboard است.

- پلتفرم، زبان (های) برنامه نویسی، چارچوب‌های مورد استفاده
- ما زبان جاوا را استفاده خواهیم کرد زیرا برای object-oriented جاوا مناسب است.
- موارد فنی دیگر!

سند معماری تحویل داده شده، می تواند شامل اطلاعات دیگری علاوه بر موارد بالا (به غیر از مورد آخر!)، مانند چیدمان فیزیکی نهایی سیستم، باشد. موارد بالا حداقل اطلاعات ضروری را نشان می دهد.

موعد تحویل: یک هفته پس از تحویل تمرین ۳

- پاسخ تمرین ها را به زبان فارسی و به صورت تایپ شده، در قالب یک فایل Pdf، در مدل بارگزاری کنید.
- سوالات خود را می توانید از طریق ایمیل از دستیاران تدریس بپرسید.
- فایل پاسخ تمرین را تنها با قالب **SE2-HW3-2-GroupX.pdf** در مدل بارگزاری کنید.
- بارگزاری تمرین توسط یکی از اعضاء گروه کافی است.
- برای پاسخ های هر قسمت منابع استفاده شده را درج نمایید.
- فایل زیپ ارسال نکنید.
- به ازای هر روز تاخیر در تحویل تمرین 20٪ از نمره تمرین کسر خواهد شد.
- حداقل برخورد به پاسخ های مشابه، تخصیص نمره کامل منفی به طرفین خواهد بود.