

نمودار use-case

## ۲,۲. دیدگاه منطقی

مخاطب: طراح سیستم

محدوده: نیازمندی های عملیاتی، توضیح design's object model و توضیح use-case های

مهم.

محصول مرتبط: design model

## ۲,۳. دیدگاه داده

مخاطب: متخصص داده، سرپرست پایگاه داده

محدوده: توضیح نمای کلی داده

محصول مرتبط: data model

## ۳. مدل های معماری

### ۳,۱. مدل معماری پیشنهادی ۱: "مخزن داده"

زیر سیستم های سازنده ی سیستم باید اطلاعات را طوری مبادله نمایند که به طور موثر باهم کار کنند. در سیستم مدیریت داخلی رستوران رضایی باید داده های مشترک در یک بانک اطلاعاتی مشترک نگه داری شوند که تمام زیر سیستم ها بتوانند به آن دسترسی داشته باشند. مدل سیستمی که این گونه مبتنی بر داده های مشترک است مدل مخزن داده است.

مزایا:

- ۱- راه کارامدی برای به اشتراک گذاشتن حجم بالای داده است.
- ۲- زیر سیستم ها نیازی به نگرانی راجع به چگونگی تولید داده ندارند.
- ۳- در این روش در گرفتن پشتیبان از محتوا، امنیت و ... مدیریت متمرکز وجود دارد.

معایب:

- ۱- زیر سیستم ها باید بر روی مدل مخزنی توافق داشته باشند یا ناگزیر به توافق برسند که کارایی ممکن است در اثر این توافق تحت تاثیر قرار گیرد.
- ۲- تکامل داده دشوار و هزینه بر است.
- ۳- حوزه ای برای سیاست های خاص مدیریتی وجود ندارد.
- ۴- توزیع کارآمد و موثر در آن دشوار می باشد.

## ۳,۲. مدل معماری پیشنهادی ۲: "لایه ای"

این دیدگاه جهت تامین امنیت سیستم، مقیاس پذیری بالا و تامین قابلیت توسعه‌ی تدریجی می‌تواند انتخاب شود زیرا این مدل سیستم به صورت مجموعه‌ای از لایه‌ها سازماندهی می‌شود که هر لایه مجموعه‌ای از خدمات را ارائه می‌دهد.

این لایه‌ها در سیستم مدیریت داخلی رستوران رضایی عبارتند از:

- ۱- لایه واسط کاربر که کنترل ورود کاربران به سیستم را بر عهده دارد و کنترل کلیه عملیات‌های مربوط به رستوران مانند به روزرسانی، جستجو، بررسی گزارشات مالی و... در این لایه انجام می‌شود.
- ۲- در لایه عملیاتی به درخواست‌های کاربران از لایه واسط کاربر رسیدگی می‌شود و نتیجه به کاربر نمایش داده می‌شود.
- ۳- لایه‌ی سوم مربوط به بخش پایگاه داده‌ی سیستم است که اطلاعات کاربران و اقلام رستوران و غذاها در آن ذخیره می‌شود. وجود این بخش در لایه‌ی سوم باعث افزایش امنیت پایگاه داده می‌شود.

مزایا:

- ۱- معماری قابل حمل است.
- ۲- امنیت بالایی دارد.
- ۳- معماری قابلیت توسعه تدریجی دارد.

معایب:

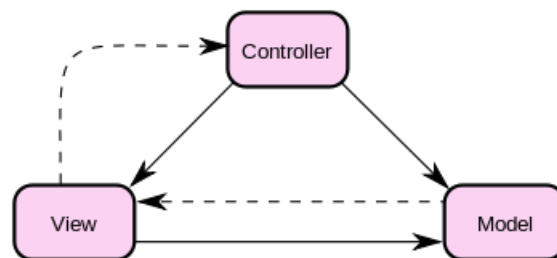
- ۱- در صورت زیاد بودن لایه‌ها مدیریت آنها باعث ایجاد سربار می‌شود.
- ۲- تعیین ساختار سیستم به راحتی امکان پذیر نیست.

## ۳,۳. مدل معماری پیشنهادی ۳: "MVC"

مدل MVC بر روی معماری‌های چند لایه‌ای جهت جداسازی قسمت‌های مختلف برنامه استفاده می‌شود. در واقع MVC یک معماری سه لایه شامل model, view, controller می‌باشد. Model: این بخش می‌تواند با داده‌ها در ارتباط باشد. به عبارتی مدل به تعریف نحوه مقدار دهی و دریافت داده‌ها می‌پردازد که این شامل توابع ضروری به جهت تضمین امنیت، تغییر یا تایید داده‌ها نیز می‌باشد.

View: این لایه وظیفه برقراری ارتباط با کاربر نهایی و گرفتن داده از کاربر و نمایش داده‌های آماده به کاربر از طریق برقراری ارتباط با دو بخش دیگر یعنی model و controller است. در واقع این بخش با داده‌های خام کار می‌کند.

**Controller** : یک بخش کنترل کننده می باشد و واسطی بین دو بخش قبلی است . در معماری MVC روند کلی برنامه به این شکل است که کاربر تقاضای خود را از طریق واسط های برنامه نویسی از برنامه در خواست می کند (از بخش view). بخش view در خواست ها را به controller فرستاده و این بخش با برقراری ارتباط با مدل درخواست های کاربر را پردازش کرده و سپس controller بخش view را آگاه می سازد تا خود را بر اساس تغییرات جدید به روز سازد.



از معماری MVC بیشتر برای برنامه های مبتنی بر وب استفاده می شود.

### ۳,۴. جمع بندی

در این سیستم مدیریت رستوران قسمت زیادی از پروژه مبتنی بر ذخیره داده هاست و قابلیت ذخیره و بازیابی اطلاعات مهم ترین وظیفه سیستم است در نتیجه استفاده از مدل مخزن داده ای کار ذخیره و استفاده از داده ها را آسانتر خواهد کرد و این مدل برای معماری پروژه پیشنهاد می شود.