

۱- از چه فرمتی برای مدیریت داده ها استفاده می کنید. دلیل خود را بیان و تحلیل کنید.
از Relational DBMS استفاده میکنیم زیرا ساختار دیتا بیس ما object oriented نیست و نوع داده ای پیچیده نیز نداریم. همچنین به دلیل ماهیت سیستم نیاز به ساپورت کردن transaction داریم و همچنین به دلیل اینکه بسیار زیاد توسط دیگران استفاده شده پشتیبانی کامل و امکانات خوبی دارد و تمام نیاز های ما را رفع میکند.

۲- بررسی کنید آیا Diagram Class ارائه شده توسط گروه شما در فاز Modelling Structural نیازمند بهینه سازی می باشد. برای پاسخ خود دلیل بیاورید.

Cohesion:

Method Cohesion:

در این مورد باید اشاره کنم که تمامی متد های کلاس ما تقریباً این ویژگی را رعایت میکنند و هر کدام از آنها یک کار واحد را انجام میدهد. یعنی توابعی که برای آنها تعریف شده به یک منظور تعریف شده و دارای یکپارچگی است. به طور مثال یکی از متد های کلاس User ، Create Account است که وظیفه ی این تابع فقط و فقط ساخت کاربر است. دیگر متد ها نیز تا حد ممکن از این قاعده پیروی کرده و یکپارچگی و اصل ماژولار بودن را رعایت میکنند. پس میتوان گفت این قسمت دارای نوع Functional است.

Class cohesion:

در این قسمت نیز باید اشاره کنم باید ما حداقل ارتباط بین کلاس ها را داشته باشیم و بدین منظور مجبور به استفاده از اتریبیوت هایی از کلاس های دیگر شدیم ولی تا حد ممکن به صورت یک کلید خارجی از آنها در بعضی از کلاس ها استفاده کردیم. به طور مثال برای تفکیک سفارشات و خرید ها از یکدیگر اتریبیوتی به نام User Id درون آن تعریف کردیم. در اصل این user id مربوط به کلاس user است ولی برای اینکه ارتباط بین کلاس ها را کاهش دهیم و به عبارتی coupling را رعایت کنیم مجبور به استفاده از آن شدیم. بنابراین میتوان گفت این قسمت دارای نوع Mixed-Role است یعنی بعضی از اتریبیوت های کلاس های دیگر در این کلاس وجود دارند.

Coupling:

در رابطه با این قسمت باید عرض کنم همانطور که در قسمت قبلی اشاره شد ما برای کاهش ارتباط بین کلاس ها از اتریبیوت های دیگر کلاس ها در کلاس های دیگر به عنوان کلید خارجی استفاده کردیم. ولی بعضی از کلاس ها نیاز به ارتباط با یکدیگر داشتند. طبق نمودار ترتیبی و نمودار ارتباطی میتوان مشاهده کرد که کمترین ارتباط بین کلاس ها در نظر گرفته شده و برای موارد مخالف سعی شده به طور مستقیم با کلاس مربوطه ارتباط برقرار کرد مگر اینکه دیگر مجبور به برقراری ارتباط با دیگر کلاس ها باشیم پس میتوان گفت که این قسمت دارای نوع Data Interaction Coupling است یعنی در ارتباط با دیگر کلاس ها فقط ولیو ها مبادله میشوند. و از دیگر نوع ها در ارتباط بین کلاس های ما خبری نیست.

با توجه به موارد اشاره شده در قسمت بالا میتوان گفت که تا حد زیادی کلاس های ما بهینه هستند .

۳- به منظور طراحی صفحات واسط کاربری نمودار Diagram Navigation Windows را رسم نمائید

۴- صفحات واسط کاربری مهم ترین سناریوی سیستم پروژه را طراحی کنید.

۵- به نظر شما از چه نوع معماری بایستی برای طراحی سیستم خود استفاده می کنید. دلیل خود را بیان و تحلیل کنید.
معماری 4 لایه:

یک لایه برای پایگاه داده که همان database server است مورد استفاده قرار میگیرد.

یک لایه برای application server که شامل لاجیک و منطق برنامه ی ماست مورد استفاده قرار میگیرد.

یک لایه برای web server که شامل web-related application logic است

و یک لایه هم شامل کلاینت که همان presentation logic است.

در واقع در نظر بگیرد که لایه ی سوم که لایه ی وب است برای هندل کردن درخواست های کاربران قرار داده شده است و لایه ی دوم که شامل درخواست های دیگر لایه ها است که منطق برنامه ی ما را در بر دارد.

دلیل اینکه برای پلتفرم وب یک لایه طراحی کردیم این است که می خواهیم اپلیکیشن وب که آماده شد را روی سروری دیگر قرار دهیم تا اگر سرور منطق برنامه با مشکلی روبرو شد سایت ما همچنان در دسترس باشد. به عبارتی با این کار می خواهیم بین سرور های مختلف تقسیم کار انجام دهیم تا اگر سروری دچار مشکل شد فرآیند جایگزینی آن آسان تر باشد.

البته میتوانیستم به دو لایه نیز این کار را انجام دهیم ولی به دلیل اینکه قصد داریم امنیت داده ها و همچنین نرم افزار ها را بالا ببریم و در صورت خرابی یکی از سرور ها داده ها و برنامه ی ما دچار خرابی نشود سرور پایگاه داده را از سرور منطق جدا میکنیم.