

تحلیل و طراحی سیستم‌ها در مهندسی نرم‌افزار ۱:

هدف ارائه فرآیند و مراحل مشخص جهت تولید نرم‌افزار است در این راستا متدولوژی‌های طراحی و تحلیل سیستم‌ها مطرح هستند.

۱- متدولوژی بیانگر روشی استاندارد برای تولید نرم‌افزار است در این راستا متدولوژی تحت عنوان Rational unified process (Rup) مطرح می‌شود. قبل از این به ذکر مقدمات تحلیل و طراحی سیستم‌ها می‌پردازیم.

۱- سؤال اول: یک سیستم چیست؟

۲- سؤال دوم: چگونه می‌توان سیستم‌ها را شناسایی نمود؟

تعریف سیستم، در ص ۷ و ص ۶۳ کتاب مفهوم سیستم و نگرش سیستمی مطرح شده است بر طبق تعریف سیستم به مجموعه‌ای از عناصر اطلاق می‌شود که در تعامل با یکدیگر ورودیهای خاصی را دریافت بر روی آنها پردازش و خروجیهای معین را تولید کند هر سیستم دارای هدفی مشخص است یک سیستم با هدف خاصی که در آن تقسیم کار ارتباطات و کنترل وجود داشته باشد را سازمان می‌گویند.

برای شناخت سیستم‌ها می‌بایست عناصر آنها را شناخت و روابط بین عناصر را تعیین کرد. اصولاً رابطه بین دو عنصر به صورت علت و معلول یا زمانی و یا بازخوردی می‌باشد. اصولاً بین عناصر سیستم می‌بایست تناسبی وجود داشته باشد کل سیستم چیزی برای حاصل جمع عناصر آن است. یک سیستم در حالت کلی دارای قابلیت‌هایی است که در مجموع اجزای آن ندارد درجه پیچیدگی یک سیستم وابسته به تعداد عناصر در روابط بین عناصر آن سیستم است. سیستمها از لحاظ

ساختاری دارای عناصر سرحدی یا مرزی، عناصر داخلی، شبکه ارتباطی و بالاخره مجموعه‌ای از انباره‌ها جهت حفظ اطلاعات و محصولات و امکانات خود هستند اجزاء یک سیستم معمولاً شامل تعدادی عنصر که هر کدام می‌توانند شامل یک سیستم مجزا باشند رابطه بین عناصر و بالاخره کنترل عناصر می‌باشند.

۱- تعریف سیستم ۲- اجزا سیستم ۳- هدف سیستم ۴- روش شناخت سیستم

در تجزیه و تحلیل سیستمها بر اساس شناخت اجزاء مبادرت به شناخت سیستمها می‌شود. اصولاً اجزاء زیر در هر سیستمی وجود دارد.

۱- هدف: هر سیستمی دارای هدفی است و اگر بی‌هدف باشد فلسفه وجودیش بی‌معناست برای

مثال سیستم حقوق و دستمزد با سیستم پرسنل هر کدام هدف خاص خود را دارند لذا به عنوان سیستم مطرح هستند.

۲- ورودیها: هر سیستمی برای ادامه حیات خود نیاز به منابع تغذیه کننده و داده‌های ورودی

دارد برای نمونه یک اتومبیل نیاز به بنزین دارد. یک سیستم حقوق و دستمزد نیاز به اطلاعات پرسنل و ساعات کارکرد آنها دارد.

۳- خروجیها: اگر ورودیها در سیستم موجود باشند و سیستم دارای هدفی باشد خروجیها ظاهر می‌گردد.

۴- توالی: برای اجرای عملیات معمولاً در سیستمها یک توالی می‌بایست در نظر گرفته شود در

طی توالی این خروجیها و اهداف سیستمها مشخص می‌گردد در واقع توالی بخش کنترلی عملیات سیستم است.

۵- محدوده و محیط: هر سیستمی تحت شرایط و قوانینی و محدوده‌های خاص و اولویت‌های مشخصی که محیط سیستم نامیده می‌شود کار می‌کند برای تشخیص سیستمها محدوده و محیط عملیاتی آنها نیز مشخص شود.

۶- ابزار: معمولاً هر سیستمی جهت حصول به اهداف خود ابزاری را به کار می‌گیرد برای مثال سیستمهای مکانیزه ابزار کامپیوتر را به کار می‌گیرند.

۷- عوامل انسانی: جهت اجرا شدن سیستمها معمولاً عواملی انسانی نقش مؤثری دارند.

متدلوژیها بر مبنایی به اجزای سیستمها مبادرت به شناسایی سیستمها می‌نماید. در طی مطالب این درس به بررسی متدلوژی شناخته شده‌ای تحت عنوان Rup یا Rutiond unfied process خواهیم پرداخت. همانطور که در بالا گفته شد برای هر متدلوژی مراحل خاصی جهت تولید نرم‌افزار وجود دارد به این مراحل در اصطلاح فرآیند تولید نرم‌افزار یا soft ware process یا چرخه حیات یا life cycle گفته می‌شود.

فرآیند Rup: متدلوژی Rup در حالت کلی شامل چهار مرحله اصلی بنام مرحله شناخت اولیه یا

۱- Inception یا تأکید (شناخت)

۲- Elaboration یا مرحله تشریح

۳- Construction یا ایجاد

۴- Transition یا انتقال است. (تحول)

در مرحله اول (شناخت اولیه) تأکید بیشتری بر روی تعیین نیازهای کاربر یا مشتری است.

مرحله تشریح یا Elaboration بیشتر تأکید به طراحی سیستمها دارد.

مرحله Construction تأکید بیشتری بر پیاده‌سازی دارد.

و بالاخره مرحله انتقال تأکید بر نصب و پشتیبانی نرم‌افزار دارد در هر مرحله از شرط حیات تعدادی Iteration (تکرار) وجود دارد. اما چرا به اینها می‌گویند. تکرار. تکرار چه چیزی هستند؟ هر Iteration یا تکرار یک مبنی mini Project است یا یک پروژه‌ای کوچک است با شناخت آغاز می‌شود و با تولید Code (کد) خاتمه می‌یابد.

اگر به صفحه ۱۱۵ کتاب مراجعه کنیم دقیقاً مراحل چرخه حیات را متوجه خواهیم شد.

هر مرحله از تعدادی تکرار تشکیل می‌شود اما چه چیزی است که در این تکرارها، تکرار می‌شود. در واقع هر تکرار یا Iteration تکراری از مراحل چرخه حیات آبخاری است در چرخه حیات آبخاری فرآیند نرم‌افزار با تعیین نیازها آغاز می‌شود و نهایتاً با نصب و پشتیبانی نرم‌افزار خاتمه می‌یابد. بنابراین هر تکراری در واقع یک تکرار از چرخه حیات آبخاری است اما چرخه حیات آبخاری به چه صورت می‌باشد چرخه حیات برای هر تکرار با Business Hodeling یا مدلسازی سیستم کاری یا سیستم جاری آغاز می‌شود و در این مرحله عملیات سیستم جاری تجزیه می‌گردد و مورد تحلیل قرار می‌گیرد تا نیازها بر مبنای آن مشخص گردد.

در طی این مرحله مشخص می‌کنیم چه کارهایی انجام می‌دهد بنابراین تعیین می‌کنیم که سیستم کامپیوتری کدام یک از این کارها را باید انجام دهد. چگونگی مهم نیست چه مهم است.

پس از اینکه مشخص کردیم سیستم کاری چه کارهایی انجام می‌دهد در مرحله Requirement تعیین می‌کنیم نیازها چیست؟

معمولاً ۸۰٪ نیازها در عملکرد سیستم جاری مستقر است و ۲۰٪ کارهای اضافی است که کاربر تقاضای آنها را می‌کند.

بعد از تعیین نیازها مرحله تحلیل و طراحی آغاز می‌گردد. در مرحله Analysis and design نیازها و شناختی که روی سیستم ایجاد شده مورد بررسی قرار می‌گیرد و سیستم مکانیزه یا کامپیوتری طراحی می‌گردد.

مرحله بعدی مرحله پیاده‌سازی است که در واقع در این مرحله کد برنامه‌ها ایجاد می‌گردد. مرحله آزمون برنامه‌ها را مورد تست قرار می‌دهد و در Deployment برنامه‌ها نصب می‌گردد. هر یک از چهار مرحله اصلی این مراحل فرعی را در بر دارد هر مرحله اصلی از چند تکرار تشکیل می‌گردد و هر تکرار کلیه مراحل آبخاری را در بر می‌گیرد. بنابراین در تکرارهای اولیه تأکید بر شناخت است و پیاده‌سازی در حد Prototype بدون طرح بانک در مرحله constraction تکرارها بیشتر تأکید بر پیاده‌سازی دارد در ادامه برای یک Iteration مراحل را توضیح می‌دهیم.

Business Modeling یا مدلسازی سیستم کاری:

هدف از مدلسازی سیستم کاری

۱- شناخت سیستم موجود به منظور تأمین نیازها

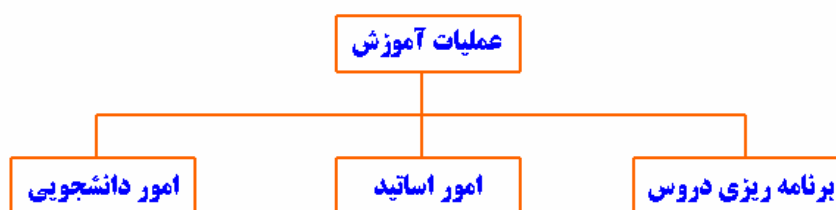
۲- گرفتن تأیید از کاربر در صحت شناخت

مرحله Business Modeling با تعیین موردهای استفاده از سیستم کاری یا سیستم جاری آغاز می‌گردد.

تعیین موردهای استفاده :

مورد استفاده یا use case در واقع سرویسی است که سیستم در اختیار مشتریان خود قرار می‌دهد. مشتریهای سیستم را Actor یا بازیگرهای سیستم کاری می‌نامند. هر Actor در واقع به فرد یا زیر سیستم دیگری اطلاق می‌شود خارج از سیستم قرار گرفته یا از سرویسهای سیستم استفاده می‌کند و یا اینکه کمک می‌کند تا سرویسهای سیستم ارائه شود. برای مثال در ابتدا ارتباط سیستم آموزش دانشکده، دانشجوی، آموزش کل، مدیریت دانشگاه اداره حسابداری، همگی عناصری هستند که در خارج از سیستم آموزش هستند منتها با آن در ارتباطند یا از آموزش دانشکده سرویس می‌گیرند و یا اینکه کمک می‌کنند تا آموزش دانشکده سرویس ارائه دهد لذا اینها Actor ها برای زیر سیستم آموزش دانشکده هستند سؤال اینجاست که چگونه می‌توان use case ها را تعیین کرد.

برای این منظور از تکنیک function Decomposition استفاده می‌شود. در این روش سیستم به عنوان یک function در نظر گرفته می‌شود بر اساس عملیات Decompose می‌شود هر واحد عملیاتی یک زیر سیستم یا یک دسته از عملیات را تشکیل می‌دهد برای نمونه.



برای مثال در چارت سازمانی واحد آموزشی دانشکده عملیات معاون آموزشی، کارشناس آموزشی، کارمند آموزش و تالیست قرا رمی گیرد. اما در چارت عملیاتی عملیات تقسیم بندی می شود.

سؤال اینجاست که چگونه می توانیم عملیات یک سیستم را تقسیم بندی نماییم.

مدل سازی سیستم جاری

اولین مرحله در چرخه حیات Rup مدل سازی سیستم جاری است به این ترتیب که سیستم موجود دستی و یا کامپیوتری مورد شناخت قرار می گیرد با این هدف که با شناخت سیستم موجود و عملکرد آن بتوانیم نیازها را تا ۸۰٪ مشخص کنیم سیستم موجود را مدل هم می کنیم علت این است که با مدل سازی سیستم موجود می توانیم حاصل شناخت خود را به کاربر نشان دهیم تا اگر اشتباه باشد کاربر یا مشتری برای ما کار را تصحیح کند. برای شناخت سیستم موجود و مدل سازی آن به صورت زیر عمل می شود.

(۱) تهیه چارت سازمانی

(۲) تعیین شرح وظایف برای پروسه سپتهای سازمانی

(۳) تعیین چارت عملیاتی

بدین وسیله عملیات سیستم تجزیه می شود.

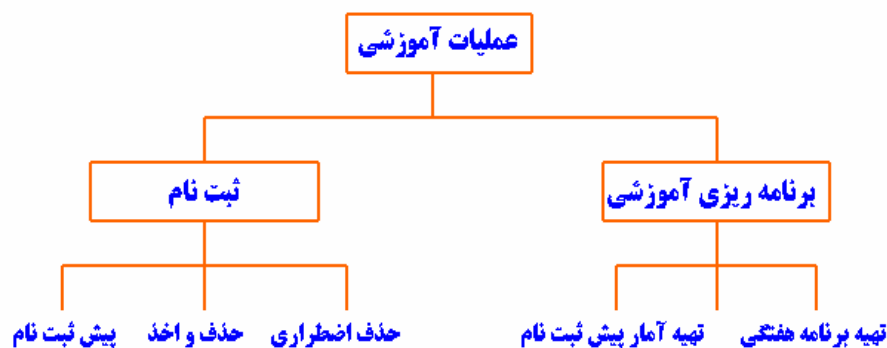
برای مثال شرح وظایف چارت عملیاتی استخراج می شود برای مثال در چارت عملیاتی واحد آموزش

حالا بر اساس چارت عملیاتی سیستم را تجزیه می کنیم تا بتوانیم بعداً مورد تحلیل قرار دهیم اما

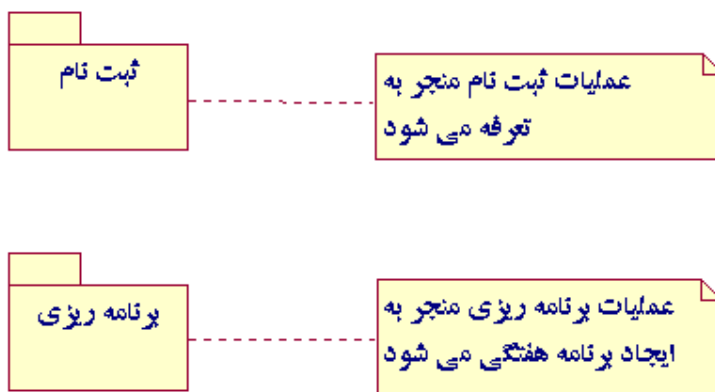
روش تجزیه به چه صورت است؟

بدین صورت با تجزیه سیستم بر اساس چارت عملیاتی می‌توانیم به شناسایی و شناخت ابزار پرداخته و به این صورت کل سیستم را مورد شناخت قرار دهیم.

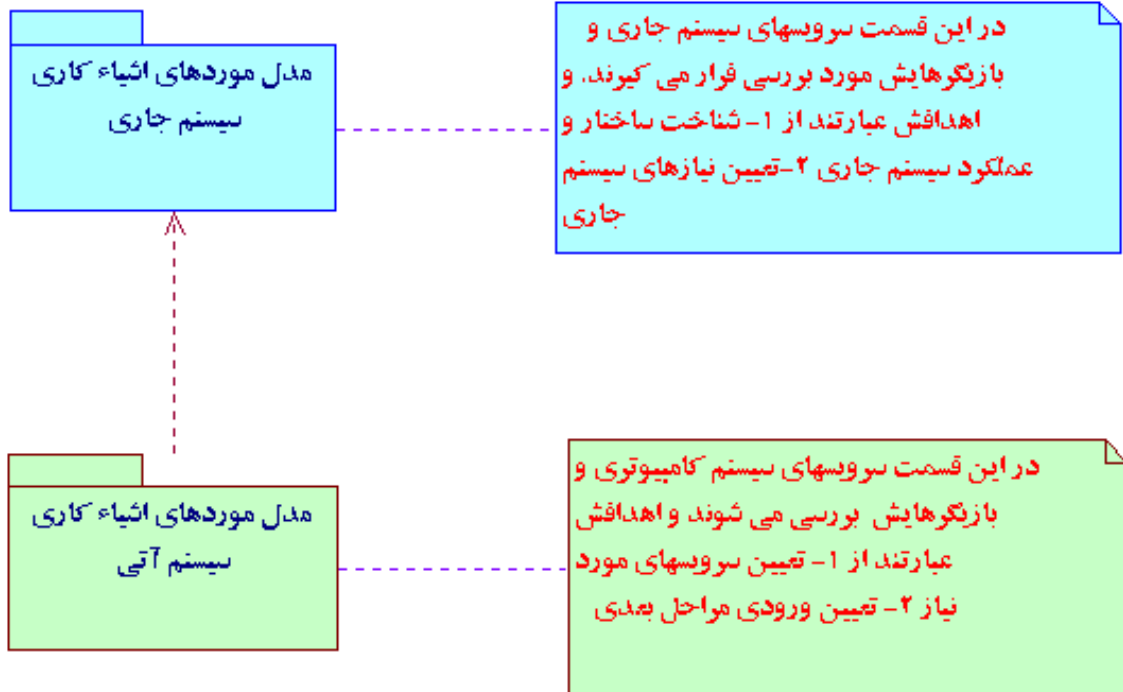
هر واحد عملیاتی در محیط Rutioned Rose در قالب یک Package یا بسته مشخص می‌شود.



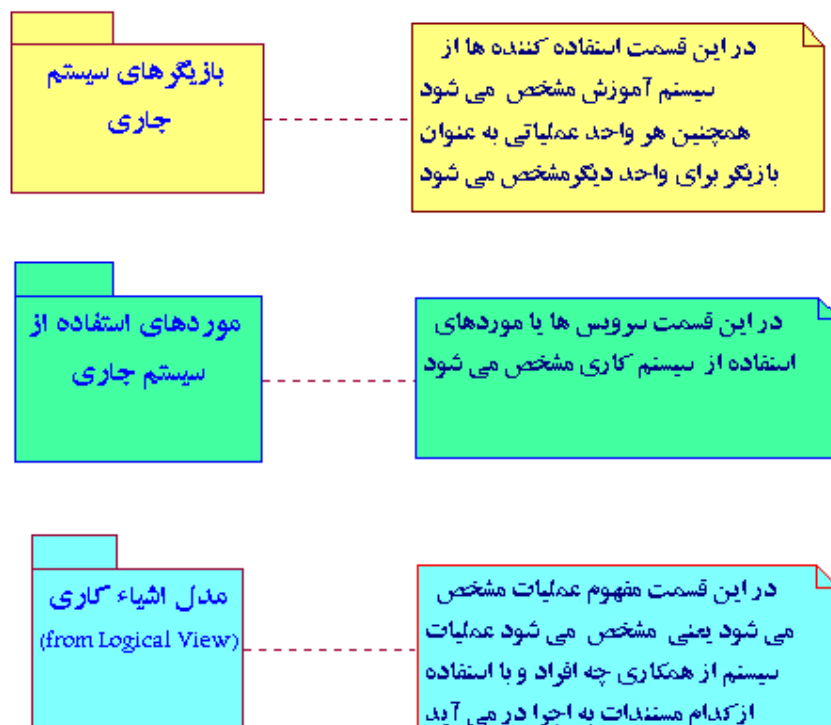
عملیات آموزش دانشکده



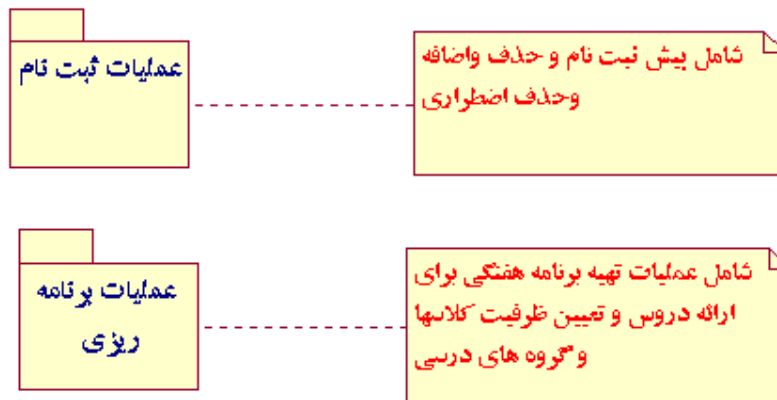
سیستم آموزش دانشکده



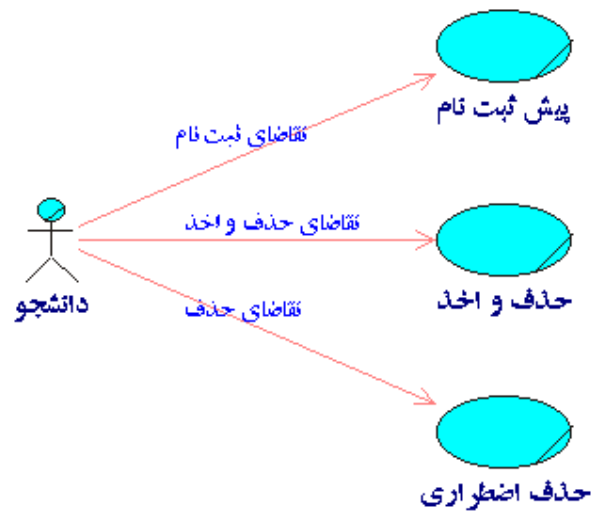
عملیات آموزش دانشکده



سرویسهای سیستم ثبت نام



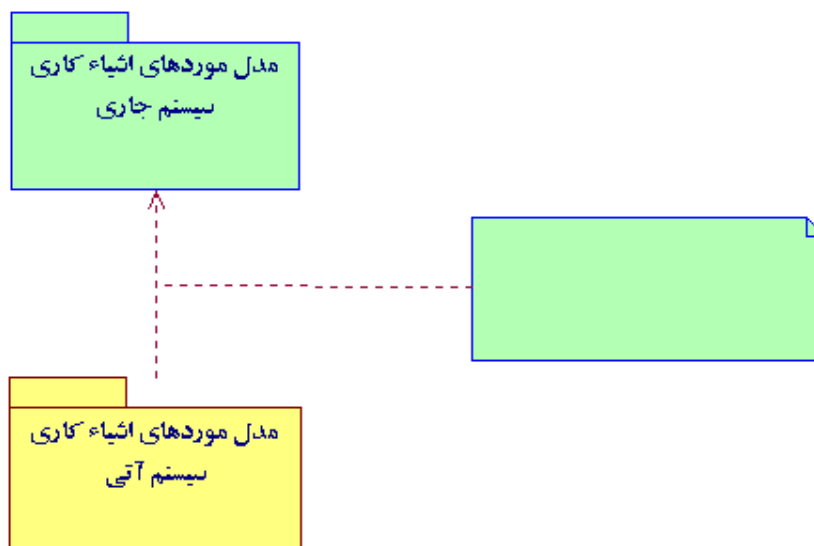
سرویسهای سیستم ثبت نام



هر package یک مجموعه‌ای از عملیات است. هر use case یک سرویس است که به کاربر از سیستم داده می‌شود، این سرویسها را use case یا مورد استفاده می‌گویند. برای سیستم کاری یا سیستم جاری بدانها Business use case گفته می‌شود. سرویس گیرنده‌ها را Actor یا بازیگر می‌نامند. بازیگرها یا External NTT اند یعنی در خارج از سیستم قرار دارند. مثل دانشجو که در خارج از سیستم آموزشی است و از آن سرویس می‌گیرد و یا اینکه ممکن است یک واحد عملیاتی باشد که از عملیات واحد دیگر استفاده می‌کند. برای مثال در اینجا برنامه‌ریزی آموزشی یک واحد عملیاتی است که به عنوان Actor می‌تواند از عملیات ثبت نام استفاده کند و بالعکس ثبت نام به عنوان یک واحد می‌توان یک Actor برای عملیات برنامه‌ریزی برای هر use case باید توضیحات آن use case را مشخص نمود یعنی مشخص کرد که عملیات آن به چه صورتی به اجرا در می‌آید. شرح use case را در اصطلاح سناریو یا فیلمنامه می‌گویند. بعد از اینکه use case model را مشخص کردیم می‌باید برای هر use case بر اساس ساختاری که در ادامه ارائه می‌شود شرح ایجاد کنیم.

در اینجا برای توصیف use case Business همانگونه که توضیح داده شد می‌توان از فیلمنامه یا سناریو استفاده کرد توصیف use case یا موردهای استفاده سیستم کاری معمولاً برای اینکه بتوان با مشتری به توافق رسید و به وی نشان داد که عملیات را درک کرده مورد استفاده است همچنین برای طرح معماری سیستم تشخیص این کار مهم است بر مبنای توصیف اینها در اقع مشخص می‌شود که عملکرد سیستم جاری چیست و سیستم در واقع می‌بایست این عملیات را به انجام رساند.

برای اینکه بتوانیم توصیف مورد‌های استفاده را ایجاد کنیم بر روی هر left use case کلیک کرده و گزینه files انتخاب و سپس در داخل محوطه خارجی Right کلیک و گزینه insert file را انتخاب می‌کنیم. حالا نام فایل مورد نظر را به عنوان یک فایل word مشخص می‌کنیم. یک مورد استفاده در واقع شاخص بخشی از عملیات سیستم است. سؤال اینجاست که چگونه مورد‌های استفاده از سیستم را مشخص می‌کنند؟ برای این منظور می‌بایست مشخص کنیم که ابتدا Actor ها چه کسانی هستند. Actor ها افرادی هستند و یا واحدهایی هستند که در خارج از یک واحد عملیاتی قرار دارند و از حاصل کار آن واحد عملیاتی بهره می‌برند. برای نمونه عملیات ثبت نام را به عنوان یک واحد عملیاتی در سیستم دانشگاه در نظر بگیرید. واحد مالی به عنوان یک Actor در اینجا در نظر گرفته می‌شود زیرا در واقع کمک می‌کند تا عملیات ثبت نام به انجام برسد.



مدلهای مورد استفاده سیستم جاری



Package specification for در این قسمت عملیات سیستم مدل سازی می شود. (use case view) با توجه به شکلهای جلسه قبل برای هر یک از use case ها با نوشتن سناریو عملکرد را مشخص می کنیم. برای این منظور دگمه mouse را use case می زنیم.

و همانگونه که در جلسه قبل توضیح داده شده یک فایل word را مشخص می کنیم که در داخل آن شرح هر یک از use case را قرار دهد.

در use case کلیک تا باز شود بعد files را انتخاب بعد Right کلیک بعد Insert file را انتخاب می کنیم (یک فایل word ایجاد می کنیم) برای نوشتن سناریو:

* گروه اگر مستقیماً با عملیات امور مالی در ارتباط بود Actor بود حالا نیست.

شرح موردهای استفاده از زیر سیستم ثبت نام

نام مورد استفاده: پیش ثبت نام

بازیگرها (اکتورها): دانشجو، حسابداری، مرکز آمار و اطلاعات

نوع: اصلی

شرح: هدف از این مورد استفاده تهیه آمار ثبت نام جهت برنامه‌ریزی دروس است.

نیازهای مربوطه:

سناریو:

سیستم	بازیگر
۲- مسئول ثبت نام پیش نیاز دروس را کنترل می کند.	۱- دانشجو فرم ثبت نام را به مسئول ثبت نام ارائه می دهد
۳- مسئول پیش ثبت نام جزئیات فرم پیش ثبت نام را وارد کامپیوتر می کند.	
۴- مسئول ثبت نام فرم پیش ثبت نام را تأیید نموده همراه با چک لیست به دانشجو تحویل می دهد	۵- دانشجو پس از پرداخت وجه پیش ثبت نام به اداره حسابداری رسید را تحویل می دهد.
۷- سیستم کامپیوتری در پرونده پیش ثبت نام دانشجو رسید وجه را ثبت مینماید.	۶- اداره حسابداری رسید وجه را وارد کامپیوتر میکند
	۸- دانشجو تقاضای برگه حذف و اضافه را می نماید
۱۰- برنامه کامپیوتری پیش ثبت نام پس از کنترل رسید وجه پیش ثبت نام برگه حذف و اضافه صادر می نماید	۹- مسئول مرکز اطلاعات شماره دانشجویی را دریافت به برنامه کامپیوتری پیش ثبت نام می دهد
	۱۱- مسئول مرکز اطلاعات فرم حذف و اخذ را به دانشجو تحویل می دهد.

موارد دیگر:

۲- در صورتیکه دانشجو پیش نیاز دروس را نگذرانده باشد مسئول ثبت نام آن درس را حذف و فرم را به دانشجو عودت می دهد.

۶- در صورتیکه رسید کامل نباشد ثبت نام حذف می شود.

۱۰- در صورتیکه رسید وجه تا سه روز پس از مهلت ثبت نام در کامپیوتر ثبت نشده باشد سیستم کامپیوتری ثبت نام دانشجو را حذف می نماید.

شرح موردهای استفاده از زیر سیستم حذف و اخذ نام مورد استفاده: حذف و اخذ

بازیگرها: دانشجو، حسابداری

نوع: اصلی

شرح: هدف از این مورد استفاده ثبت نام دانشجویان برای ترم تحصیلی است.

نیازهای مربوطه:

سناریو:...

سیستم	بازیگر
۲- مسئول ثبت نام پیش نیازها را کنترل می کند.	۱- دانشجو فرم حذف و اخذ اصلاح شده را به مسئول ثبت نام می دهد.
۳- مسئول ثبت نام شماره دانشجو را به سیستم کامپیوتری ثبت نام وارد می کند.	
۴- سیستم کامپیوتری جزئیات پیش ثبت نام را نمایش می دهد.	
۵- مسئول ثبت نام در صورت وجود اصلاحات را براساس فرم حذف و اخذ وارد کامپیوتر می کند.	
۶- کامپیوتر یک چک لیست جدید بر اساس اطلاعات حذف و اخذ صادر می نماید.	
	۷- دانشجو از مسئول مرکز اطلاع رسانی چک لیست را می گیرد.
	۸- دانشجو رسید وجه را به اداره حسابداری تحویل می دهد.
۱۰- سیستم کامپیوتری رسید را در پرونده دانشجو ثبت می نماید.	۹- اداره حسابداری از سیستم ثبت نام تقاضای ثبت رسید را می نماید.
۱۱- مسئول ثبت نام تأییدهای ثبت نام را از مسئول مرکز اطلاعات دریافت می کند.	
۱۳- مسئول ثبت نام تأییدیه را در دو برگ به دانشجو تحویل می دهد.	۱۲- دانشجو مدتی بعد تقاضای تأییدیه از مسئول ثبت نام می نماید.
۱۶- مسئول ثبت نام برگ تأییدیه امضا شده را در پرونده دانشجو قرار می دهد.	۱۴- دانشجو دو برگ تأییدیه را امضا و یک برگ را به مسئول ثبت نام می دهد.

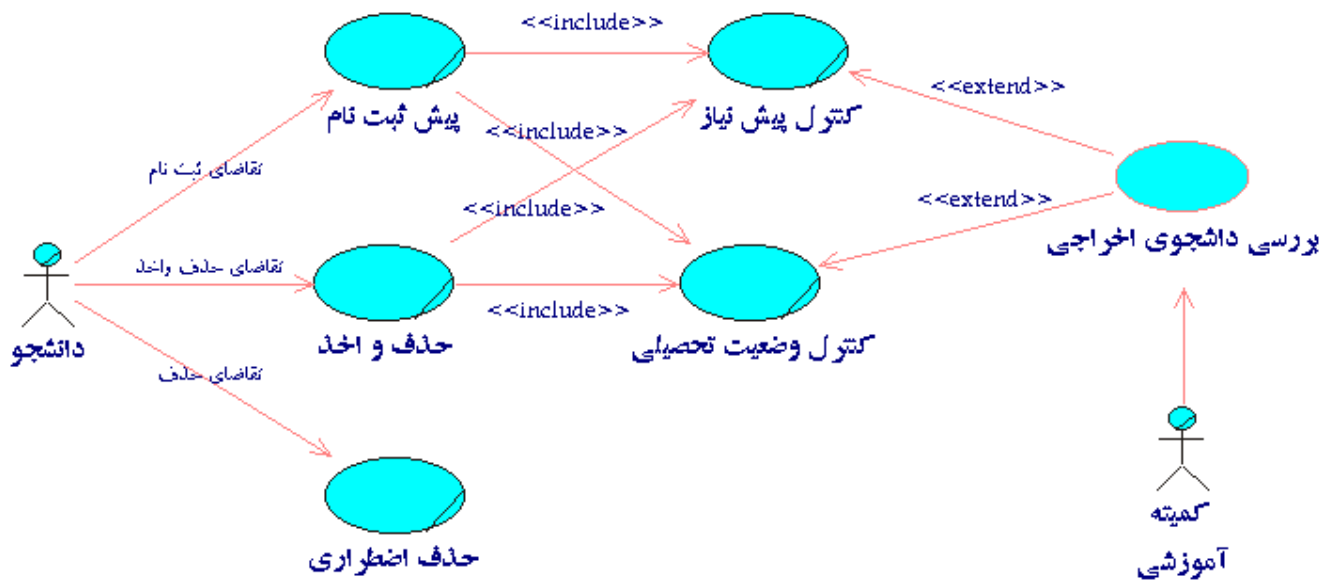
تمرین: چارت سازمانی - چارت عملیاتی - در داخل Rational Rose بر اساس چارت عملیاتی و با

استفاده از گزینه Package, Rup و نهایتاً مدل موردهای استفاده و شرح موردهای استفاده را قرار

می دهید.

Use case ها با یکدیگر ارتباط مستقیم ندارند ارتباط بین use case ها معمولاً از طریق بانکهای اطلاعاتی رخ می دهد بعد از اینکه use case ها را مشخص کردیم حالا به بررسی آنها می پردازیم و حالات استثنایی و عملیات مشترک را برای آنها استخراج می کنیم.

سرویسهای سیستم ثبت نام



همانگونه که در شکل فوق مشاهده می کنیم عملیات مشترک بین پیش ثبت نام و حذف و اخذ در قالب دو مورد کنترل پیش نیاز و کنترل وضعیت تحصیلی مشخص شد چون این عملیات همیشه بخشی از پیش ثبت نام و حذف و اخذ هستند رابطه بین آنها با این دو Use case با stereo type include مشخص شده است. رابطه Include مشخص می کند که برای نمونه کنترل پیش نیاز همیشه بخشی از عملیات پیش ثبت نام است و کنترل وضعیت تحصیلی هم به همین صورت می باشد، رابطه

extend شرط استثنائی را مشخص می‌کند و جهت فلش آن بر عکس include است در واقع در اینجا در صورتیکه در هنگام کنترل پیش نیاز و یا کنترل وضعیت تحصیلی متوجه شدیم که دانشجوی سنوات ندارد برای مورد استفاده بررسی دانشجوی اخراجی ارجاع می‌کنیم، توجه کنیم که این مورد استفاده همواره بخشی از عملیات کنترل پیش نیاز و کنترل وضعیت تحصیلی نیست بلکه تحت شرایط عملیات آنها را توسعه می‌دهد و برای همین خاطر است که جهت فلش آن بر عکس است.

نحوه مدل‌سازی برای یک سازمان بزرگ:

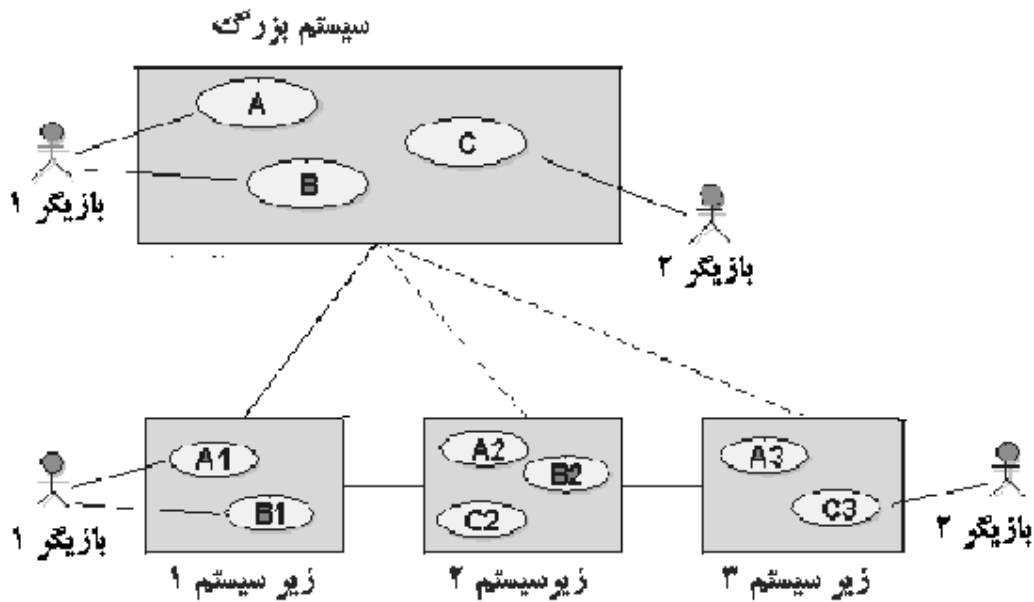
مسأله اینجاست که چگونه برای یک سازمان بزرگ با واحدهای بسیار می‌توان مدل سیستم جاری را ایجاد نمود. معمولاً سازمانها بر اساس پیچیدگی مقصودشان عملیاتشان پیچیده و جهت تغییر، پیچیدگی عملیات در بین واحدها توزیع می‌شود بنابراین تعداد موردهای استفاده کاری (business weave) ها بدین ترتیب افزایش می‌یابد. شما یک سازمان را در نظر بگیرید این سازمان اهدافی دارد و بر اساس این اهداف سرویسهایی خارج از خود می‌دهد.

معمولاً مدیران اجرایی و مدیران پروژه‌ها دو دیدگاه مختلف و یا دو تصویر مختلف از یک کمپانی یا سازمان دارند. مدیریت علاقمند به اهداف استراتژیک است. کلیات را می‌خواهد ببیند در صورتیکه مدیریت عملیاتی بیشتر به جزئیات می‌پردازد و می‌بایست business use case ها را برای هر دو این حالت ایجاد کرد یعنی business use case در سطوح بالاتر برای مدیریت اجرایی و در سطوح پایین‌تر با جزئیات بیشتر برای مدیریت عملیاتی می‌بایست ایجاد شود به این ترتیب برای هر use case سطح بالا یک use case model ایجاد می‌شود تا اینکه مدیریت عملیاتی جزئیات را ببیند.

معمولاً با business Actor اصلی کار را شروع می‌کنیم و سرویس‌ها یا موردهای استفاده از این

سیستم را مشخص می‌کنیم و این نقطه شروع کار است و برای هر business use case در سطح بالا

یک use case جداگانه ایجاد می‌کنیم برای نمونه به شکل زیر توجه نمایید.



در شکل فوق همانگونه که مشاهده می‌کنید سیستم بزرگ سه سرویس A, B, C را ارائه می‌کند که این سرویسها هر بخش از عملیاتشان توسط یک زیر سیستم ارائه می‌گردد یعنی سرویس A توسط سه زیر سیستم یک تا ۳ و تحت عناوین A1 تا A3 انجام می‌گیرد منظور اینجاست که اگر یک سازمان سرویسی را ارائه می‌دهد، واحدهای آن سازمان هر کدام بخشی از آن سرویس را به عهده دارند. سرویس ما در اینجا A نام دارد و این سرویس A توسط سه واحد عملیاتی به انجام می‌رسد در شکل زیر تولید محصول Produce product یک مورد استفاده کاری است که مشتری و Betavser و

Potential customer از آن استفاده میکنند برای ارائه محصول یا تولید محصول سه مورد استفاده

مشخص شده است که عبارتند از:

۱- Produce processor یا فرایند تولید

۲- Produce tool ابزار تولید

۳- بالاخره ایجاد بخشهای تکمیلی مذکور. اینها در مجموع سه use case هستند که توصیف

Produce Product use case را برعهده دارند.

برای اینکه بتوانیم مدل جاری یا business model را ایجاد کنیم آن را می توانیم لایه بندی بکنیم. به

این ترتیب که مشخص کنیم عملیات هسته ای (core business use case) کدام دسته می باشند و

برای اینکه این سرویسها ارائه شوند به سرویسهای دیگر نیاز است. به این ترتیب سیستم لایه بندی

می شود. در واقع با لایه بندی کارها در یک سازمان تقسیم بندی می شود. لایه های بالاتر از سرویسهای

لایه های زیرین خود استفاده می کنند. برای مثال در سیستم ثبت نام یک لایه، یک لایه مرکزی ثبت نام

اولیه دانشجوی است که لایه بالای یعنی ثبت نام ترم از آن استفاده می کند پس یک روش برای کنترل

پیچیدگی عملیات سازمانها تشخیص لایه های عملیاتی در واقع لایه بندی سیستم است همیشه

لایه های بالایی از لایه های زیرین خود استفاده می کند و لایه های بالایی در واقع از لایه های زیرین

سرویس می گیرد.

بنابراین برای لایه های بزرگ ما می بایست اکتورهای کل سیستم و سرویسهای آن را مشخص کنیم و

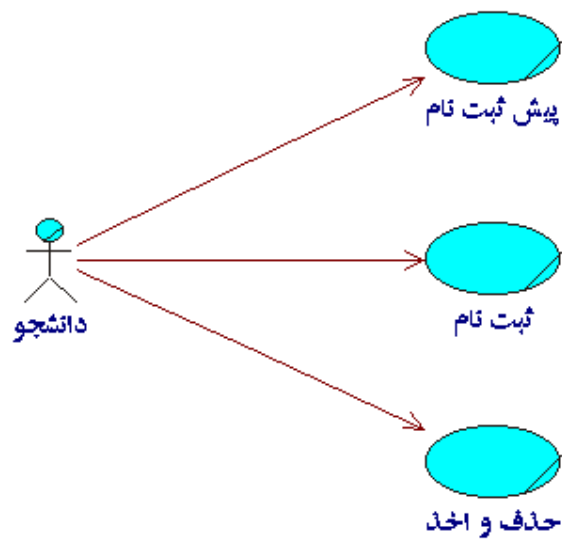
آنگاه است که برای مدیریت اجرایی عملکرد کلی سازمان تعیین می گردد بر مبنای آن سرویسهایی که

واحدها می دهند مشخص می شوند برای نمونه ببینید که عملیات ثبت نام به چه صورتی انجام می شود؟

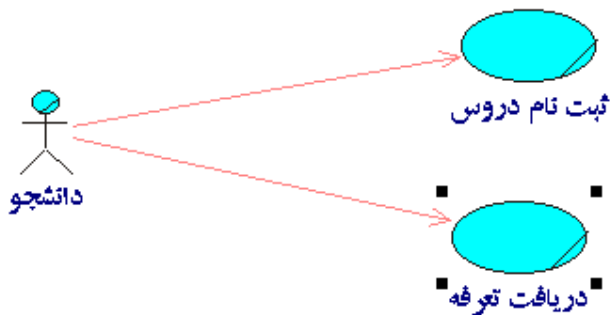


روی موردهای استفاده ثبت نام کلیک می کنیم.

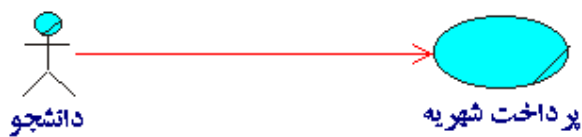
موردهای استفاده از سیستم جاری سیستم ثبت نام



روی ثبت نام کلیک می کنیم.

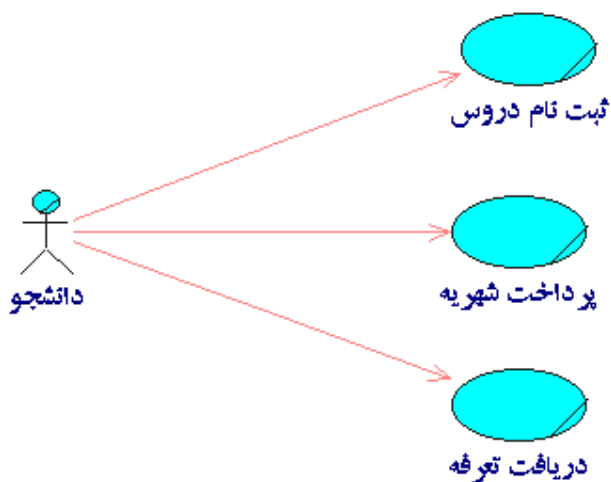


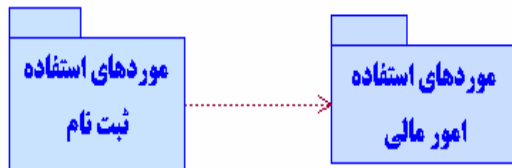
روی مورد های استفاده امور مالی کلیک می کنیم:



حال می خواهیم پرداخت شهریه مورد های استفاده امور مالی را به * اضافه کنیم در Browser کلیک

کرده و بعد در * کلیک می کنیم تا به آنجا منتقل شود.





(ولی اکتورش امور مالی می شود. چون امور مالی در زیر سیستم ما است پس آن را از اینجا حذف می کنیم.



روی موردهای استفاده مراکز آمار و اطلاعات کلیک می کنیم.

دریافت تعرفه حذف و ارائه تاییده ثبت نام توسط Browser به آن اضافه می شود.

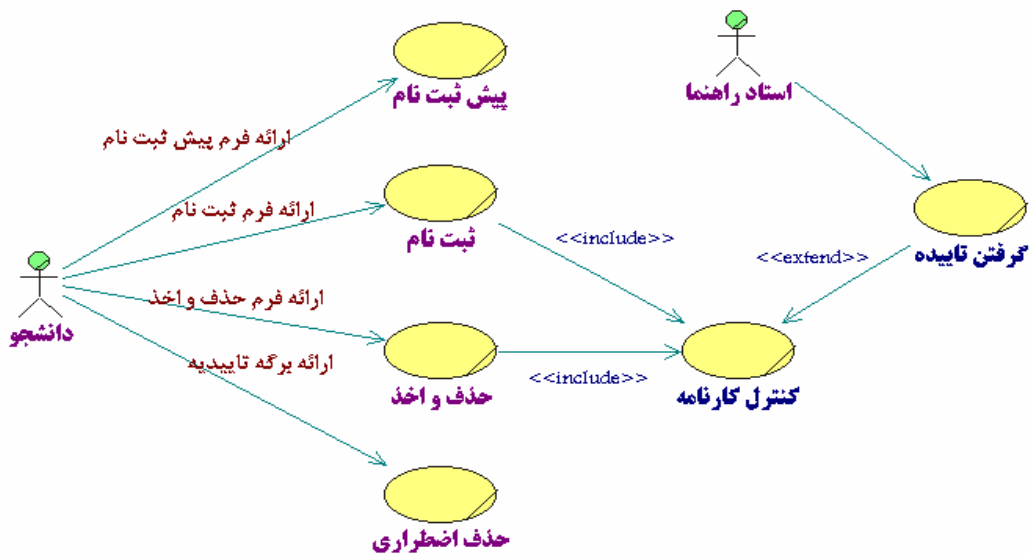
ارتباط بین موردهای استفاده:

موردهای استفاده مستقیماً در ارتباط با یکدیگراند مگر موارد استثنائی که در زیر توضیح داده خواهد شد. فرض بر این است که ارتباط بین use case ها از طریق یک بانک اطلاعاتی انجام می گردد. بانک اطلاعاتی که از دیدگاه پنهان است.

عملیات مشترک را می بایست مشخص نمود و در قالب use case های مشترک تعیین کرد برای نمونه به سیستم ثبت نام توجه نمایید در اینجا هم برای عملیات حذف و اخذ، هم برای عملیات ثبت

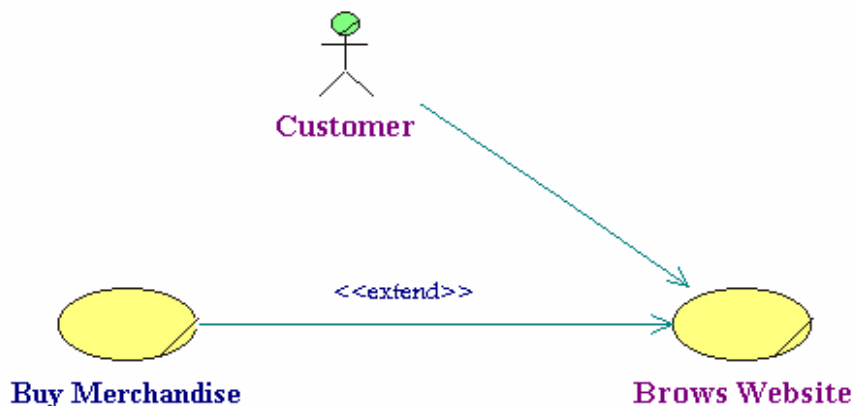
نام لازم است که پیش نیاز درس کنترل شود و اطمینان حاصل گردد که دانشجو پیش نیاز را گذرانده است این عمل را در قالب use case مشترک مشخص می‌کنیم. در عملیات فروش یا use case فروش یا در use case خرید و use case ثبت اسناد در حسابداری هر سه عمل ثبت سند وجود دارد لذا ثبت سند را به عنوان یک use case مشترک میتوان مشخص نمود چنانچه عملیات ثبت نام و حذف و اخذ را در نظر بگیرید در هر دو کنترل معدل یا کنترل پیش نیاز انجام می‌گردد لذا این دو عمل را به عنوان use case مشترک مشخص می‌نماییم.

موردهای استفاده سیستم ثبت نام



در مثال فوق همانگونه که مشاهده می‌کنید عملیات مشترک در قالب یک use case با رابطه include مشخص شد اما هنگام شرط معدل در صورتیکه معدل دانشجو کمتر از حدی باشد دانشجو برای گرفتن دروس نیاز به تأییدیه استاد راهنما را خواهد داشت در این صورت چون این حالت شرطی است و همواره جزئی از use case نیست آن را به صورت یک use case دیگر مشخص می‌کنند.

در مثال فوق در رابطه با include, extend مشخص شد رابطه include نشان می‌دهد که یک use case دیگر است برای این مثال در بالا همواره جزئی از عملیات ثبت نام کنترل کارنامه می‌باشد اما گرفتن تأییدیه حالت استثنایی اتفاق می‌افتد گرفتن تأییدیه در صورتی عمل کنترل کارنامه را توسعه می‌دهد که معدل دانشجو کمتر از ۱۲ و تعداد دروس مورد ثبت نام بیش از ۱۴ واحد می‌باشد. رابطه extend نشان می‌دهد که یک use case تحت شرایط استثنایی use case دیگر را extend یا توسعه می‌دهد رابطه include نشان می‌دهد که یک use case از عملیات use case دیگر استفاده می‌کنند.

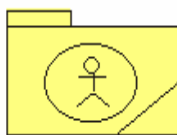


در مثال زیر مشتری مورد استفاده **Browes web** را فعال می‌کند اما در صورتیکه علاقمند باشد عمل خرید را انجام می‌دهد مفهوم از **use case Realizations** نشان دادن این واقعیت است که در دنیای واقعی که چگونه یک **use case** به انجام می‌رسد. در دنیای واقعی برای اینکه یک سرویس ارائه شود افراد یا کارمندا با یکدیگر همکاری می‌کنند تا آن سرویس ارائه شود افراد را اصطلاحاً **Business workers** یا کارکن‌ها می‌نامند از آنجایی که ما می‌خواهیم سیستم ارتباطی ایجاد کنیم و از آنجا که داده و اطلاعات برای ما مهم است باید مشخص کنیم که این کارکن‌ها چه اطلاعاتی را می‌گیرند و چه اطلاعاتی را فراهم می‌کنند.

این اطلاعات در قالب **Business Entity**ها مشخص می‌شود. حالا برای ایجاد سیستم کامپیوتری یک مرحله جلو رفته‌ایم. با نوشتن سناریو برای **use case** ما نشان دادیم که چگونه عملیات در سیستم جاری به انجام می‌رسد و با ترسیم **Business use case** در واقع مشخص می‌کنیم که چه اطلاعاتی مورد دسترسی قرار می‌گیرند.

این اطلاعات در قالب **entity class**ها در سیستم کامپیوتری درج می‌شود **Business workers** در قالب کلاسهای کنترلی در صورت امکان **Actor**ها برای سیستم کامپیوتری مشخص می‌شوند اما مسأله این است که چگونه این عناصر را پیدا کنیم و چگونه نشان دهیم که این عناصر عملیات یک **use case** را تحقق می‌دهند. برای این منظور در **Business object Moded** تحت **logical view** می‌بایست ابتدا سیستم را بر اساس واحدهای سازمانی تقسیم‌بندی کنیم. برای مثال در سیستم دانشکده واحد سازمانی به نام آموزش، واحد سازمانی دیگری بنام گروه و واحد سازمانی دیگری بنام مدیریت دانشکده وجود دارد.

واحدهای سازمانی را **organization** می‌گویند. یک واحد سازمانی شامل **Business worker** ها ، **Business Entity**ها و واحدهای سازمانی دیگر است. در **logical view** بر اساس چارت عملیاتی، واحدهای سازمانی را مشخص می‌کنیم یعنی بر اساس **package** اگر در واقعیت یک واحد سازمانی وجود دارد.



Organization Unit

از نوع خاصی از **package** به شکل زیر استفاده می‌کنیم.

در **use case** مشخص می‌شود که چه کارهایی باید انجام شود و در **logic, logical view** یا منطق کارها مشخص می‌شود.

عملیات سازمان بر اساس چارت عملیاتی درست همانطوری که به صورت فیزیکی وجود دارد در قالب واحدهای سازمانی به جای **package**ها تقسیم‌بندی می‌شود در یک واحد سازمانی ممکن است چند واحد سازمانی دیگر وجود داشته باشد و عملیات یک واحد سازمانی را می‌توانیم با استفاده از **Package**ها دسته بندی کنیم.

در صورتیکه عملیات ثبت نام به عنوان یک واحد سازمانی جدا شناخته شده باشد می‌بایست آن را به صورت یک **organization unit** یا به صورت یک واحد سازمانی مشخص کرده یا در غیر اینصورت به صورت یک **Package** مشخص شود بعد از اینکه واحدهای سازمانی را مشخص کردیم می‌بایست **Business workers** را مشخص کنیم در واقع **Business workers** شاخص نقش افراد هستند و نمایانگر افراد نمی‌باشند یک فرد ممکن است چند عضو سازمانی مختلف داشته باشد و یک **Business worker** در واقع یک پست سازمانی را مشخص می‌کند.



Business Worker

Business worker ها به صورت زیر مشخص می شوند.

در واقع Business worker یک چکیده سازی از انسانی است که در درون یک سیستم فعالیت می کند Business worker با یکدیگر همکاری می کند بر روی Business Entity ها عمل می کند.

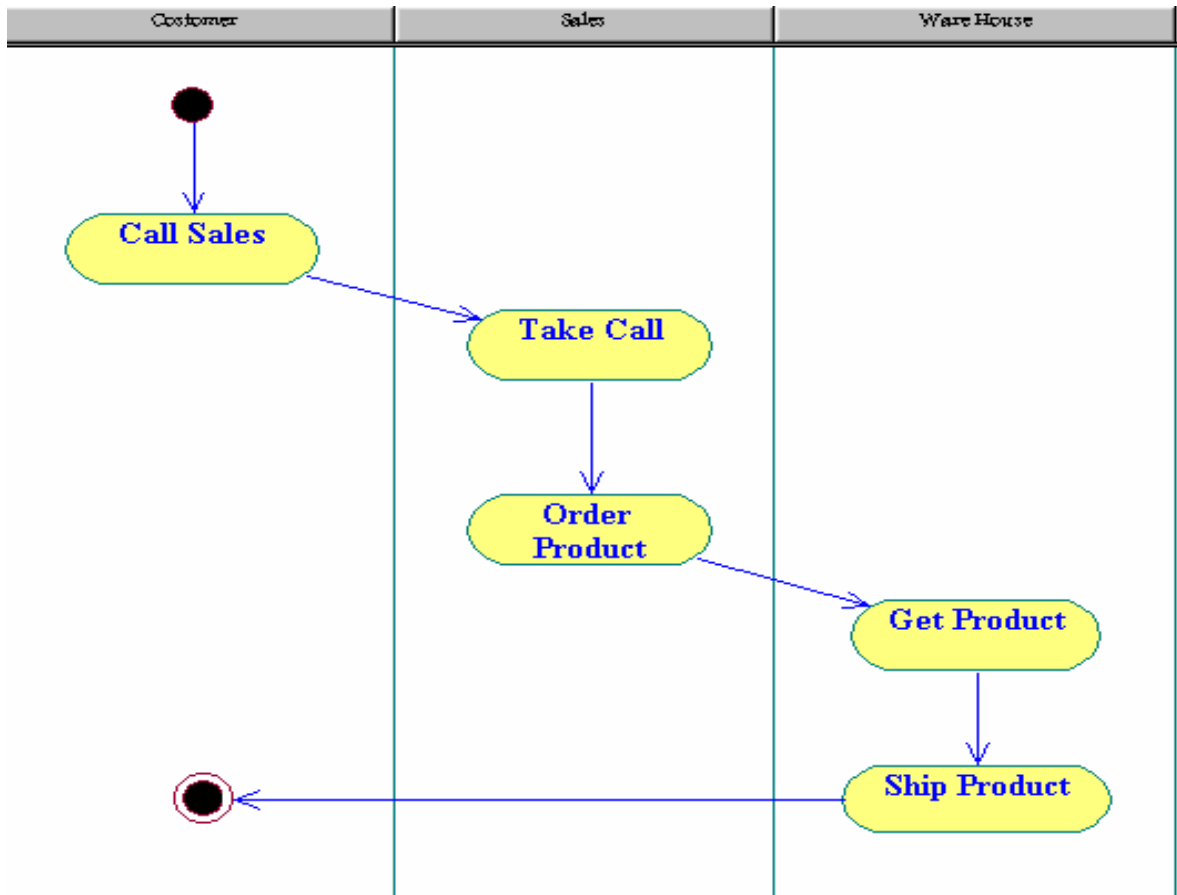
دیاگرام فعالیت: در حالت کلی ابزارهای پیشرفته از فلوچارت است جهت توصیف مفاهیم به کار می رود. بنابراین می توان از آن برای توصیف موردهای استفاده، استفاده کرده دیاگرامهای فعالیت در حالت کلی شامل عناصر زیر هستند.

۱- Activity

هر Activity یا فعالیت در واقع نمایانگر بخش مشخصی از عملیات یک use case یا هر نوع عملیاتی است و با علامت زیر مشخص می شود:

۲- Swimlane (باند شنا)

هر باند شنا نمایانگر یک واحد عملیاتی یا یک object می باشد که در مواقع مشخص می کند کدام واحد عملیاتی، عملی را انجام می دهد. در مدلسازی سیستم جاری هر swimlane شاخص یک واحد سازمانی و یا نقش در داخل یک سازمان است برای نمونه به مثال زیر توجه کنید.



دیاگرامهای فعالیت در 462 کتاب نیز هستند.

در مثال فوق دایره توپر نمایانگر نقطه شروع عملیات است در واقع مشتری یا customer واحد فروش را فراخوانی می کند از طریق تلفن و call sale یک فعالیت است به توسط مشتری به اجرا در می آید و Activity در اشکال oval مشخص می شوند take call درخواست را دریافت می کند و سپس فعالیت order Product سفارش محصول را انجام می دهد و این سفارش انبار را دریافت می کند.

انبار یا warehouse با انجام فعالیت Get Product محصول را آماده می کند و با انجام فعالیت ship product محصول را برای مشتری حمل می کند و دو دایره تو در تو شاخص خاتمه عملیات است.

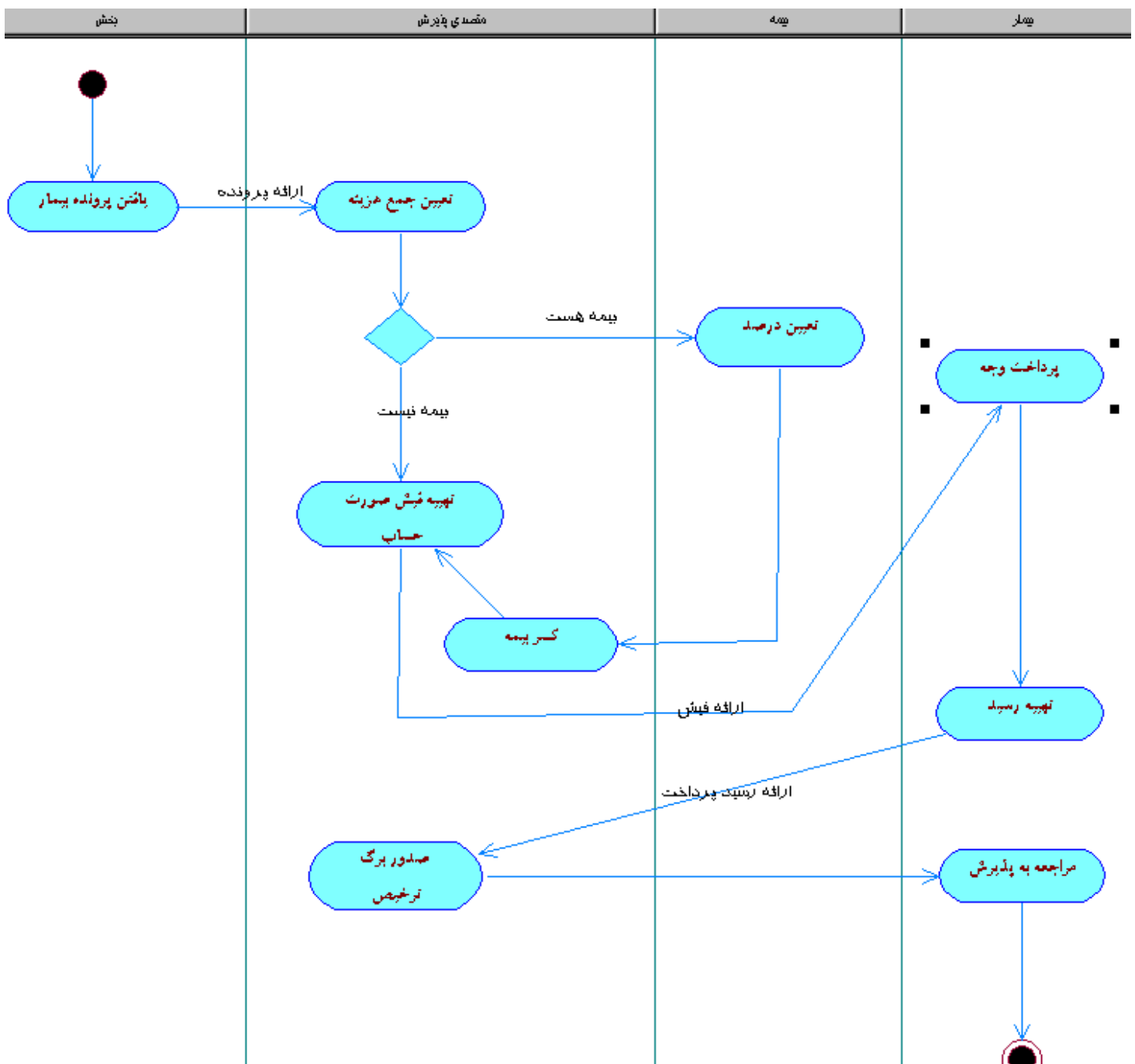
عنصری دیگر که در یاگرام فعالیت مورد استفاده قرار می گیرد state نام دارد فرق state با Activity این است که state یک حالت است برای ورود به حالت جدید نیاز به رویداد است اما فعالیت فربخشی از عملیات است.

در state chart فقط state وجود دارد و از Activity استفاده نمی شود اما state chart دقیقاً مثل Activity diagram ساده شده است یعنی می توان به جای state chart از Activity diagram استفاده کرد. State chart مدل رفتاری است. اما رفتار چیست؟ رفتار دنباله ای است از حالات.

State Diagram نوعی خاص از Activity Diagram است و مشخص می‌کند که رفتار

چيست؟ رفتار ممکن است متعلق به یک فرد یا یک واحد یا یک object باشد.

با استفاده از event می‌توان به object ها شعور بخشید.



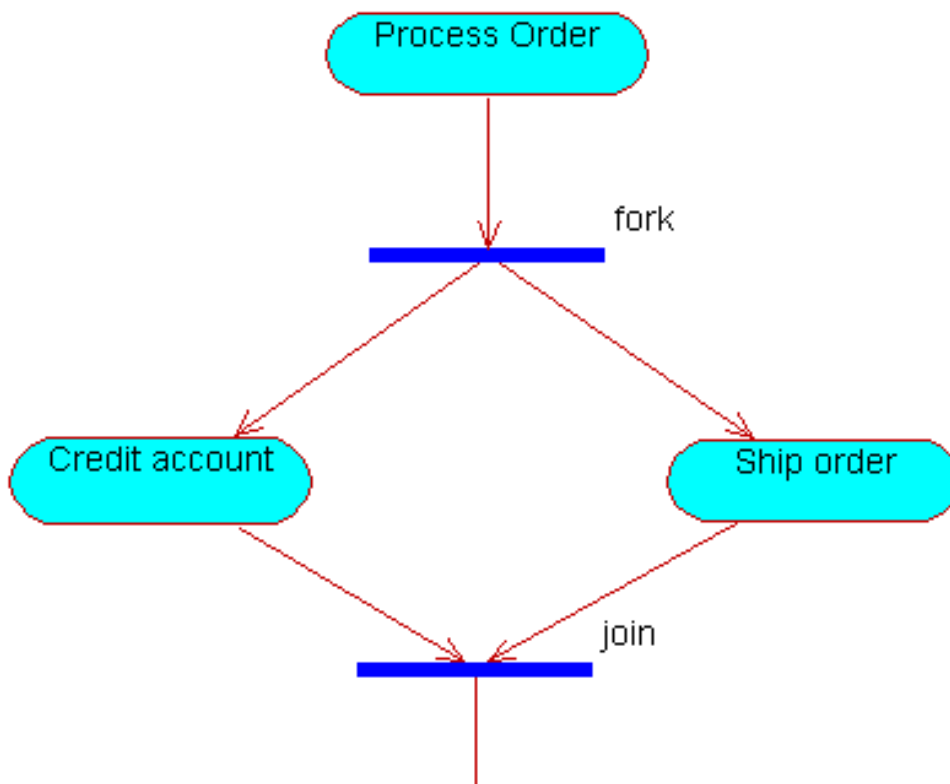
می توان با استفاده از علائمی (علائم هماهنگ کننده) تحت عنوان

Horizontal sychornization, vertical sychornization توازی در عملیات دیاگرام

فعالیت را مشخص کرد در واقع جائی که عملیات بصورت همزمان آغاز می گردد را به صورت چگال

(Fark) مشخص می کند و جائی که دوباره عملیات باید از حالت موازی به حالت خطی برگشت

کند را با علامت join مشخص می کنند برای نمونه به مثال زیر توجه کنید:



مفهوم عملیات را با استفاده از دیاگرام فعالیت انجام می‌گیرند.

دیاگرامهای فعالیت حالت پیشرفته فلوچارت هستند دیاگرامهای فعالیت قابلیت توازی عملیات را دارند که برای این منظور از `join`, `fork` استفاده می‌شود. در سیستم کارت ساعت در شیء گرا برای هر چیزی یک `object` داریم. (کارت را بخوان وضعیت را بخوان دو کار بطور همزمان را با `fork` نمایش می‌دهند).

`Join` زمانی است که لازم نیست سری ادامه پیدا کند.

تمرین: به عنوان تمرین برای `use case` های خود دیاگرام فعالیت بکشید سعی کنید همه شرایط را در نظر بگیرید.

بر اساس جریان گردش عملیات که برای مدل‌سازی سیستم جاری مشخص شده عملیات به چند روش انجام می‌شود در ابتدا وضعیت کاری ارزیابی می‌شود براساس وضعیت تصمیم‌گیری می‌شود که عملیات چگونه ادامه یابد توجه کنید که بهترین روش مدل‌سازی برای سیستم کاری چیست در مستند

به اسم چشم انداز پروژه یا **Business vision** اهداف سیستم مدل سازی برای سیستم جاری را تعیین کنید.

اهداف می توان بهینه سازی سیستم جاری باشد که کار سیستم دستی را بهینه کنند ممکن است اهداف تعیین نیازها و یا صرفاً شناخت سیستم باشد.

Business vision در اختیار مدیریت قرار می گیرد و بر اساس آن تصمیم گیری پروژه ادامه می یابد. برای اینکه ما بتوانیم مدل سازی سیستم جاری را شروع کنیم ابتدا باید چارت سازمانی را تعیین کنیم. به این ترتیب مشخص می کنیم که چه نقشهایی (چه کسانی) در داخل سیستم وجود دارند. سپس مرحله **Domain- Modeling** آغاز می شود. هدف از **Domain Modeling** در واقع مدلی است از سیستم جاری، مدلی از اطلاعات مورد استفاده در سیستم جاری که در واقع یک **object Model** است که در این جلسه درباره آن صحبت می کنیم. در واقع با این روش مشخص می کنیم که چگونه **use case** ها به واقعیت می پیوندند.

تاکنون مشخص کردیم که سیستم جاری چه سرویسها یا **use case** ها را ارائه می کند. حالا می خواهیم مشخص کنیم که این سرویسها چگونه ارائه شوند. همانگونه که می دانیم این سرویسها را افراد ارائه می کنند. افراد با هم همکاری می کنیم تا سرویسها ارائه شوند همکاری افراد در قالب دیگرامهایی بنام **collaboration** مشخص می شوند.

بنابراین ابتدا باید **Business worker** را تعیین کرد هر **Business worker** در واقع یک نقش است. **Business worker** در واقع بر روی **Business Entity** می گردد تا با کمک یکدیگر و با

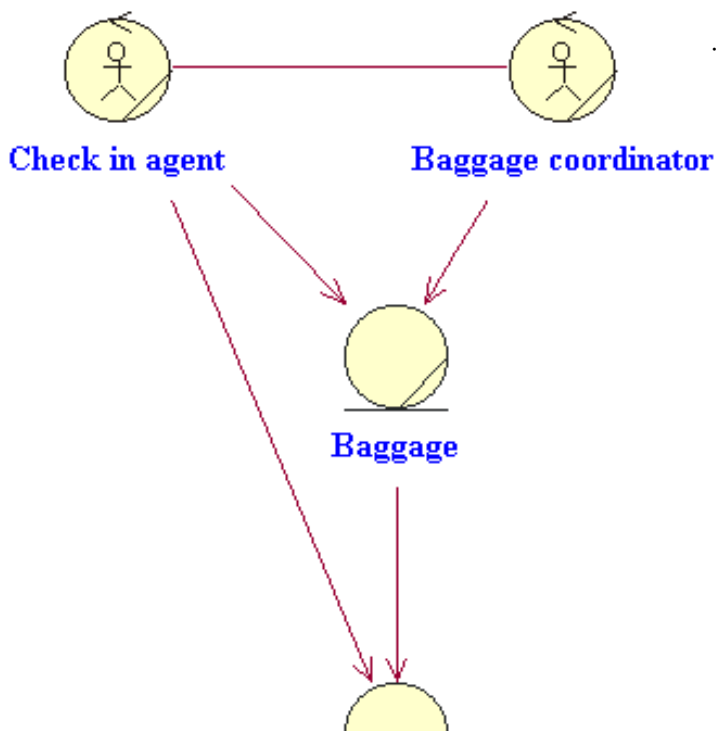
استفاده از **Business entity** بتواند سرویسهای هر **use case** را ارائه دهد برای نمونه به شکل زیر توجه کنید. شکل زیر یک مدل ارتباطی اشیاء کاری است.

بر مبنای سناریو **use case** ایجاد شده است در این مثال **check in Agent** مسئول کارت پرواز است این فرد بلیط مسافری را گرفته و پس از تعیین صندلی کارت پرواز به آنها می دهد با این فرض، فرد دیگری بنام متصدی بار همکاری می کند. متصدی بار از شکل زیر **Baggage coordinator** نامیده می شود. **Baggage** در اینجا همان اطلاعات بار مسافر است برای مثال مشخص می کند که این بار چقدر وزن دارد و متعلق به کدام بلیط (کارت پرواز) است.

این **Business Entity** حاوی جزئیات بار مسافر است.

در شکل زیر **Baggage** بصورت یک **Business Entity** مشخص شد این **Business Entity** حاوی جزئیات بار مسافر است **Baggage tag** نیز یک **Business Entity** است که حاوی اطلاعات برجسب بار است به این ترتیب در شکل فوق مشخص شده که چگونه **Business worker** ها با یکدیگر همکاری کنند و به چه مستنداتی ارجاع می کنند تا اینکه نهایتاً عمل

checking یا کنترل مسافر انجام گیرد.



مثال فوق نمایانگر Business object Model است که مشخص می‌کند چگونه Business worker در ارتباط با Business Entity ها کار check in را یا در واقع عملیات ورود مسافر به هواپیما را انجام می‌دهد. دیاگرام فوق ساختار را نشان می‌دهد همانطور که در بالا نیز توضیح داده شده می‌بایست مشخص کرد که چگونه افراد با یکدیگر همکاری می‌کنند تا سرویسها ارائه شود افراد را Business worker می‌گویند یک Business worker در واقع یک نقش است. یک نقش سازمانی است که در واقعیت وجود دارد Business workerها معمولاً برای انجام عملیات خود شرح وظایف دارند و یا اینکه یک check list دارند که در آن مشخص می‌کنند چه کارهایی تا به حال انجام داده‌اند بنابراین اینها بر روی Business worker ها عمل می‌کنند تا اینکه سرویس مورد نظر (Business use case) انجام گیرد.

معمولاً در سازمانها برای افراد شرح وظایف وجود دارد شما بهتر است بعد از اینکه چارت سازمانی را بدست آورد به شرح وظایف پردازید و مشخص نمایید که هر پست سازمانی چه وظایفی بر عهده دارد تا بر مبنای آن چارت عملیاتی را مشخص نمایید.

Business Entity بصورت فیزیکی وجود دارند. باید برای **Business worker** مشخص کرد که با چه **Business Entity** هائی در ارتباط است در واقع اینها دفاتر و پرونده‌هایی هستند که بر روی آنها کار می‌شود. باید مشخص کرد که با چه **Business worker** هایی در ارتباط هستند. باید حداقل هر **Business worker** در **Business use case** دخیل باشد.

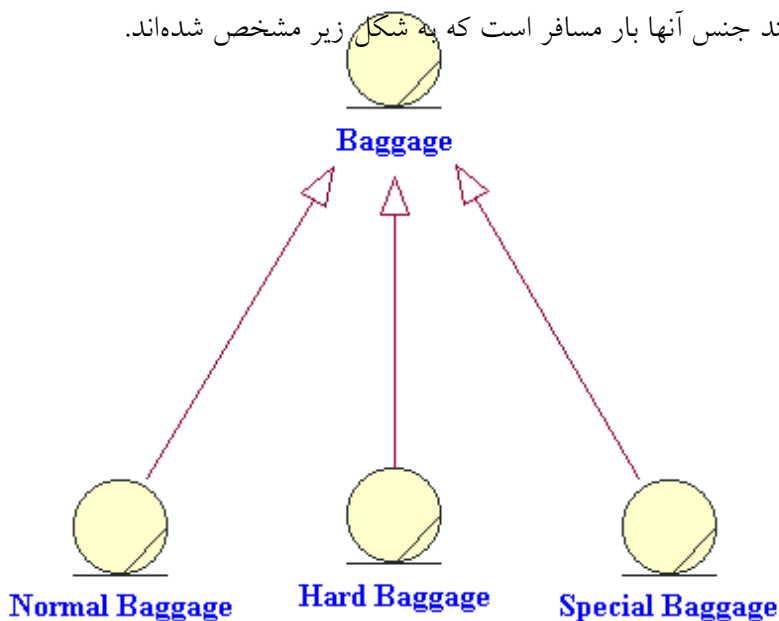
Business Entity در واقع در نهایت تبدیل به **Business class** ها و بانک اطلاعاتی سیستم می‌شود لذا از اهمیت ویژه‌ای برخوردارند. باید مشخص کرد که چگونه **Business Entity** ها در ارتباط با یکدیگر قرار می‌گیرند توجه کند یک **Business Entity** بنام مشخصات دانشجو در ارتباط با کارنامه دانشجو قرار دارد باید بتوانیم این ارتباطات را مشخص کنیم. برای نمونه به شکل زیر توجه کنید.

در شکل زیر هیأت مدیره یک کمپانی شامل رئیس، مدیر اجرایی و تعداد یک یا بیشتر صاحبان سهام در آن کمپانی است مدیر هیأت مدیره یک **Business worker** است و مدیر اجرایی **chairman** (CEO) یا مدیر هیأت مدیره و صاحب سهم **shreholder** هر کدام یک **Business worker** هستند که در مجموع **Business worker** دیگری بنام هیأت مدیره (**board**) را تشکیل می‌دهند.

رابطه بین **Business Entity**, **Business worker** در اصطلاح (**Agregation**) یا اجتماع به آنها می‌نامند. می‌بایست این اجتماع را مشخص کرد در مثال فوق رابطه **Agregation** یا تجمع

مشخص شود (بصورت یک لوزی توخالی) و مشخص کرد که هیأت مدیره کمپانی شامل مدیر اجرایی، رئیس هیئت مدیره و یک یا بیشتر سهامدار است رابطه بین دوشی ممکن است یک به یک، یک به چند، چند به چند باشد. و در شکل فوق همانگونه که مشخص است رابطه هیأت مدیره با سهامدار ۱ به چند است یعنی هیأت مدیره شامل چند سهامدار است.

نوع دیگر ارتباط، ارتباط وراثتی یا ارتباط جزئی یا کلی و یا به عبارت دیگر ارتباط **Generalization** است در واقع اگر حسابدار و صاحب حساب را بعنوان **Business worker** در نظر بگیریم هر دوی اینها از جنس کارمند حسابداری هستند و کارمند حسابداری جنس این دو را مشخص می‌کند در شکل زیر بار مسافر ۳ نوع است یکی کیف دستی دیگر **special Baggage** یا بار خاص (مثل شکستی)، **Normal Baggage** یا با عادی اینها هر یک ، یک **Business Entity** هستند جنس آنها بار مسافر است که به شکل زیر مشخص شده‌اند.

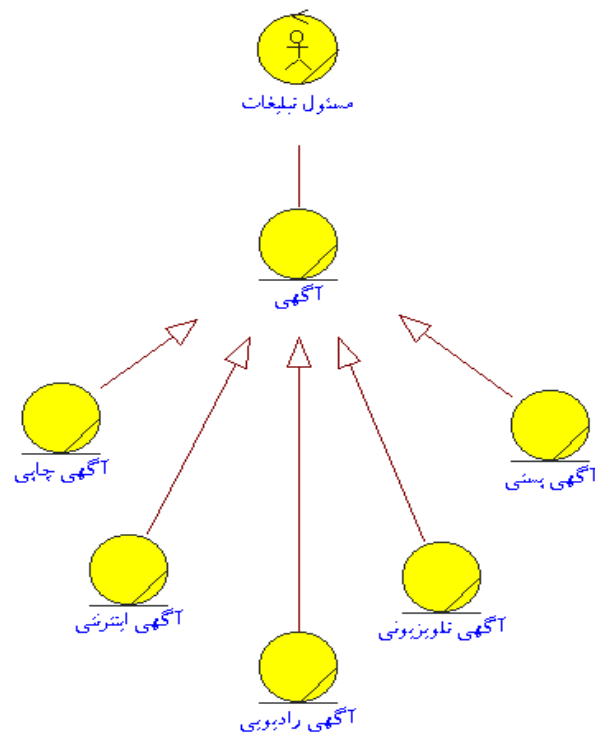


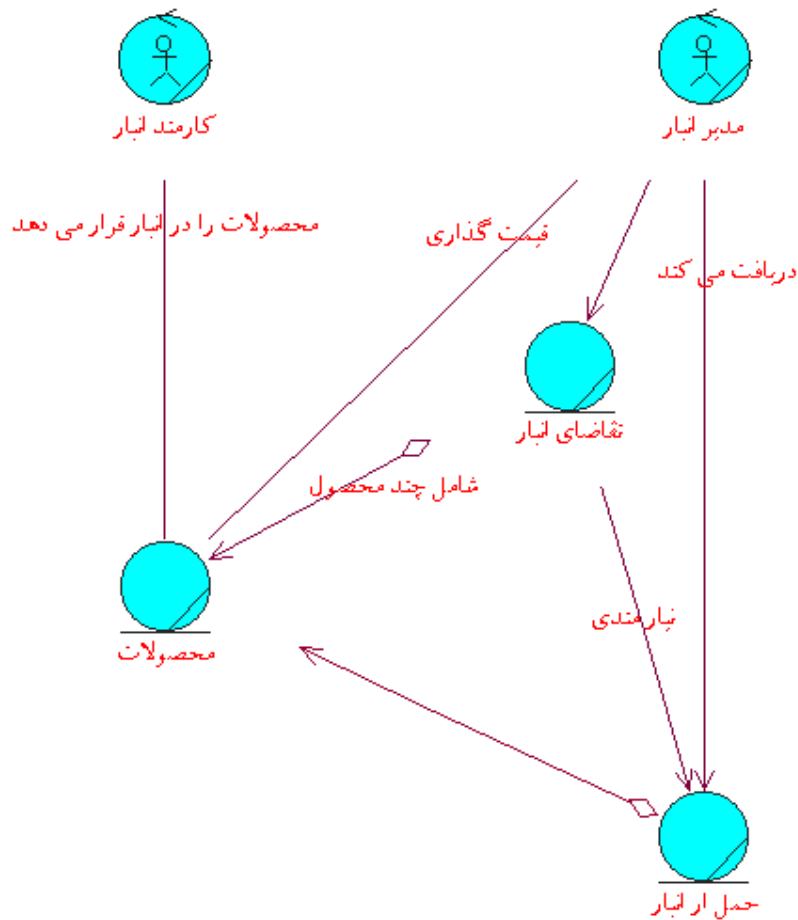
چرا رابطهٔ وراثت را مشخص می‌کنیم؟ چگونه می‌خواهیم مدل ما گویا باشد و این گویای با تعیین جنس objectها امکان‌پذیر می‌گردد.

معمولاً Generalization را بعد از اینکه مدل ارتباطی کلاسها کامل شد انجام می‌دهند تا اینکه مدل گویا شود.

در واقع مدل ارتباطی بین object یک نوع class diagram است class diagram ها در واقع بیانگر چگونگی Business use caseها هستند.

برای اینکه مشخص کنیم use case ها چگونه به اجرا در می‌آیند از دیاگرامهای بنام collaboration Diagram استفاده می‌شود در واقع class Diagram مشخص می‌کند که چگونه objectها با یکدیگر همکاری می‌کنند تا اهداف یک use case برآورده شود. برای use case تبلیغات در زیر کشیده شده.





دیاگرام فوق شاخص ساختار عملیاتی واحد انبار خرده فروشی است.

معمولاً ابتدا **logical view** بر اساس خود را عملیاتی برای سازمان و واحدهای عملیاتی

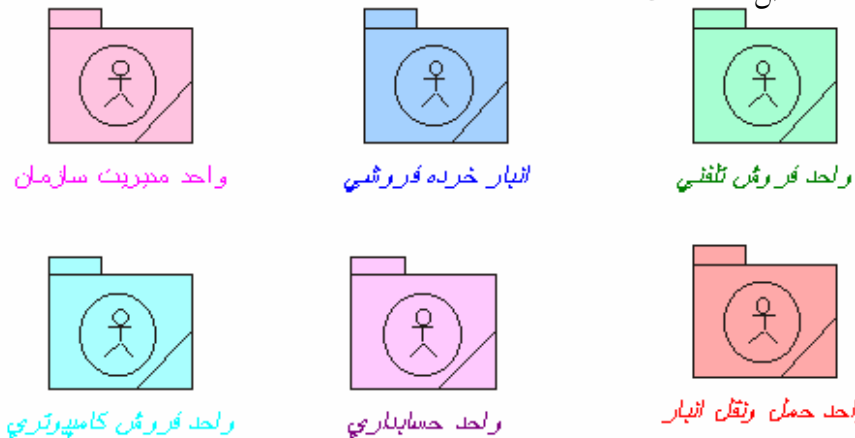
Business object model را به تعدادی واحد تقسیم می کند.

Logical view

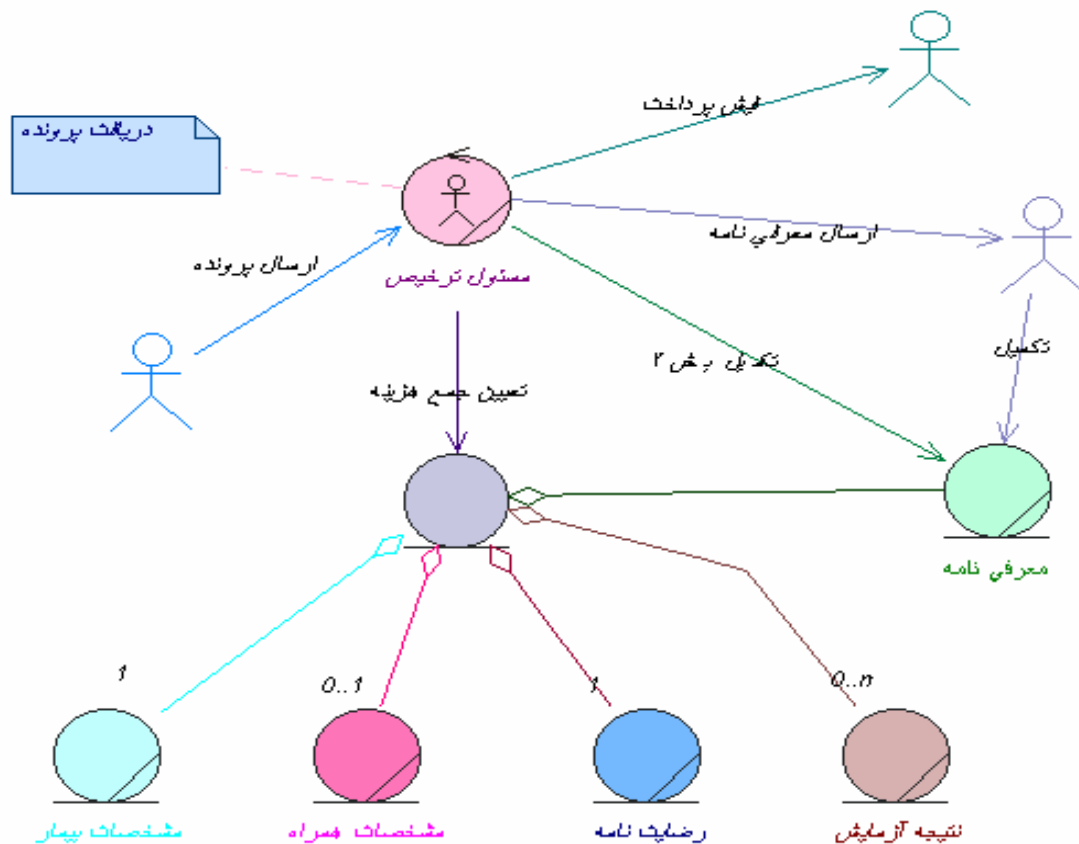
- * analysis Model
- * Business object Model
- * Design Model
- * Reverse Engineered

برای نمونه به شکل زیر توجه کنید در این شکل مدل سیستم کاری برای واحد انبار فروش

محصولات مشخص شده است.



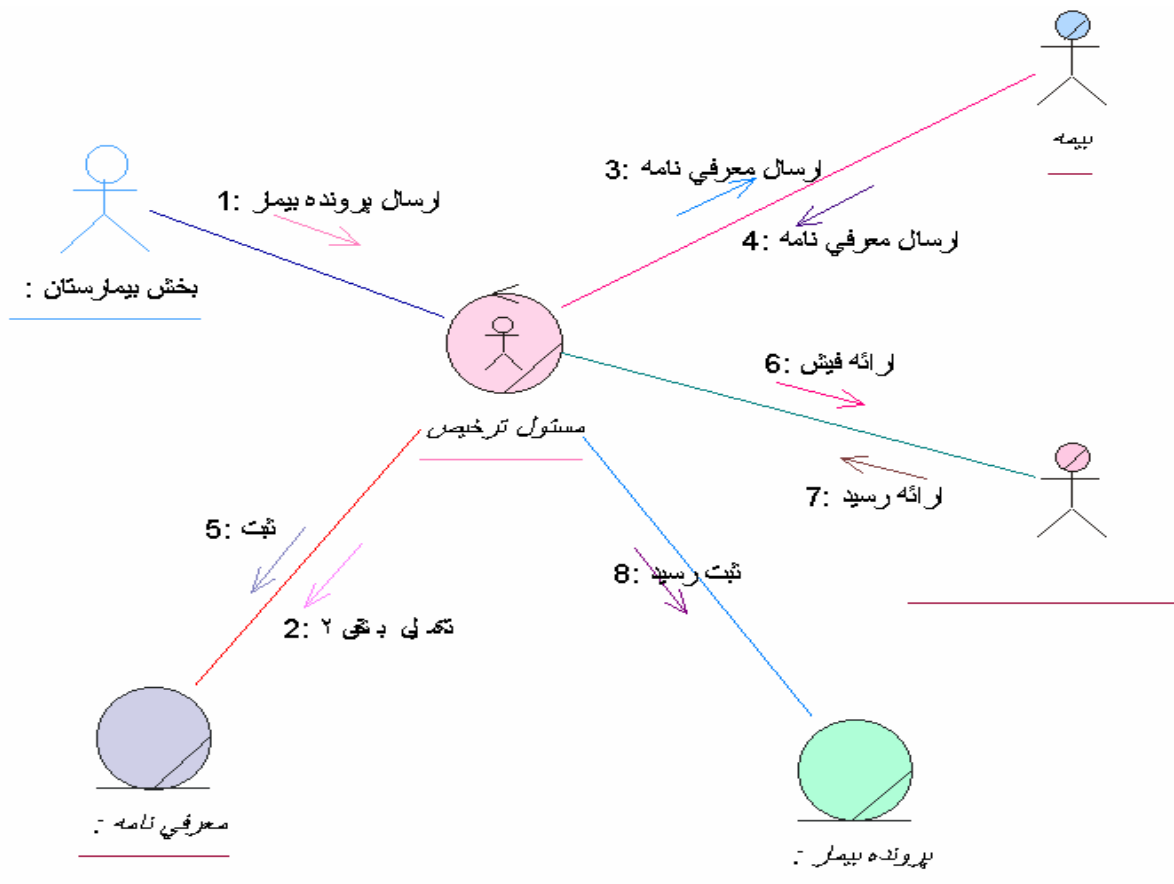
درون این Business object Model یک شکل داریم که درون این شکل ها اشکال زیر را داریم.



واحد حسابداری چون خارج است پس Business Actor است.

یک unit برای unit دیگر Business Actor است. دیاگرام کلاس برای مورد ترخیص بیمار در شکل فوق ارائه شد در اینجا ساختار نمایش داده شد. چگونگی انجام عملیات تعیین نشد برای این منظور از دیاگرامی بنام دیاگرام همکاری استفاده می‌شود.

Business object Model را از قسمت logical view برداشته و درون use case view مدل اشیاء کاری را حذف کرده Business object درون use case ترخیص insert collaboration دیاگرام می‌کنیم. درون این Collaboration Diagram از قسمت logical view تمام محتویات صفحه قبل را می‌آوریم.



در شکل فوق نشان دادیم که چگونه عمل ترخیص از همکاری بین Business worker ها و عملکرد آنها بر روی Business Entity ها انجام می‌گردد و یک روش کار این بود که logical use case Relazation به view use case ترخیص را دوباره ترسیم می‌کردیم به ازاء آن یک use case Relazation قرار می‌دادیم و به صورت زیر (صفحه قبل) در اینجا بهتر بود که برای عملیات ترخیص بیمار به صورت زیر داشته باشیم و Callabration Diagram , Class Diagram را درون ترخیص قرار می‌دهیم.

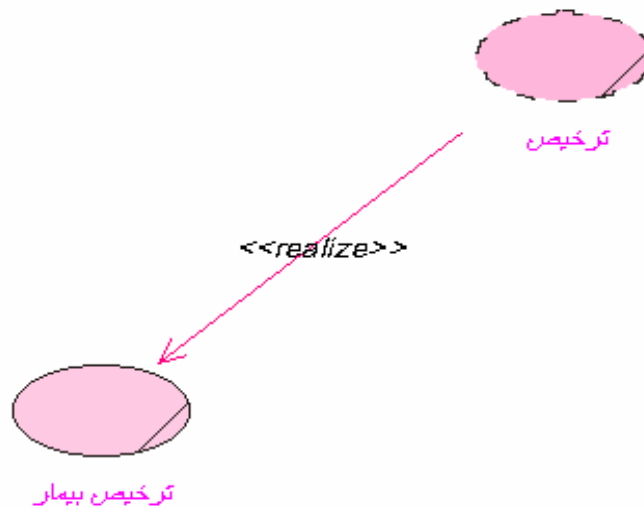
زمانی که وارد رشنال رز می‌شوید یک صفحه در مقابل شما ظاهر می‌شود البته با انتخاب گزینه New منوی فایل که از شما می‌خواهد مشخص کنید بر طبق چه استاندارد می‌خواهیم مدل ایجاد شود استاندارد Rup را انتخاب می‌کنید با انتخاب Rupها وارد Business Object Model می‌شوید. 3 عدد package در اینجا می‌گذارید.

Business use case -۲

Business Actors -۱

و سومی را خالی می‌گذاریم.

بعد از اینکه use case را تعیین کردید Activity دیاگرام و object Model را تعیین می‌کنید و وارد logical view می‌شوید در logical view بر اساس چارت عملیاتی و چارت سازمانی واحدهای سازمانی را مشخص می‌کنید در داخل واحدها رابطه use case با Realization آن را مشخص می‌کنید و بر روی use case realization که به صورت بیضی خط چین است و در بالا برای ترخیص ایجاد شد کلیک می‌کنید و در قسمت collaboration Diagrams و هم class Diagram را ترسیم می‌کنید.



Actor به سیستم سرویس می دهد یا می گیرد.

Business worker را پستهای سازمانی می گیریم.

۱- Business worker ها در قالب پستهای سازمانی مطابق با واقعیت مشخص گردند.

۲- Business Entity ها به عنوان مستندات مورد استفاده در متن سناریو مشخص و در

صورت وجود قالب بندی و به فرم ضمیمه گردند و بدانها ارجاع شوند.

۳- Actor ها یا افرادی در خارج سیستم هستند و یا زیر سیستم هایی هستند که از زیر سیستم

دیگر سرویس می گیرند و یا کمک می کنند که سرویسی ارائه شود حالا بعد از اینکه سناریو را

دقیقاً مشخص کردیم با کشیدن دیاگرام فعالیت گردش کار و فعالیتها را مشخص می کنیم. در

مرحله بعدی وارد Business object Model می شویم دو قسمت logical view

می شویم دو قسمت logical view و با استفاده از روند زیر نشان می دهیم که چگونه افراد

به واقعیت پیوند و یا در اصطلاح realite شود یا به عبارت دیگر یک سرویس ارائه گردد این

سرویس (مورد) را اصطلاحاً collaboration diagram گویند.

چگونگی → logical view

و

چرا → use case

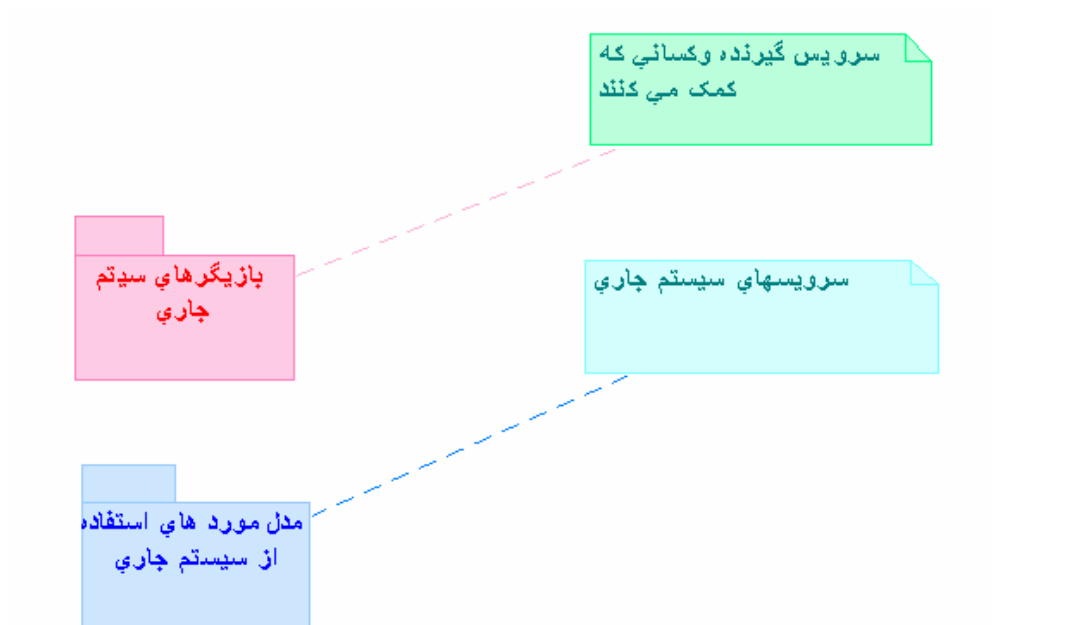
را مشخص می کنند.

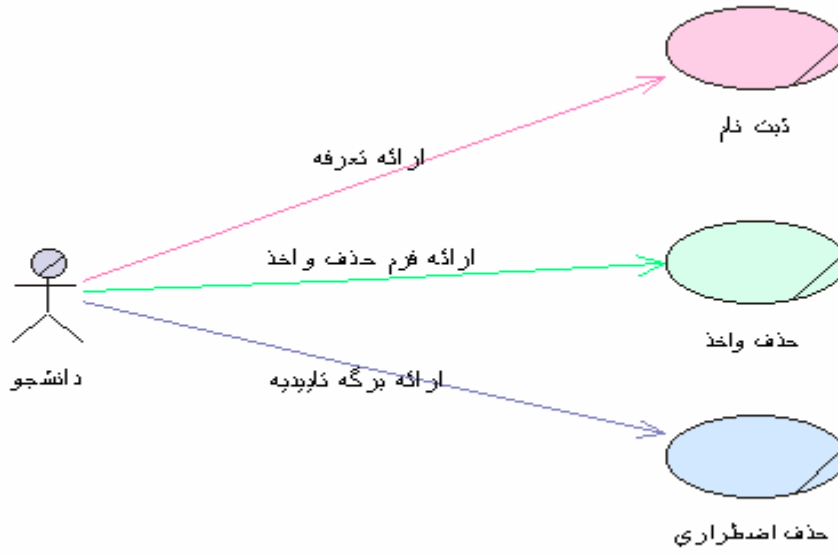
Class Diagram فقط نشان می دهد که افراد چگونه با هم ارتباط دارند.

Collaboration Diagram نشان می دهد (در واقع روح دارد) که افراد چگونه با هم همکاری

می کنند.

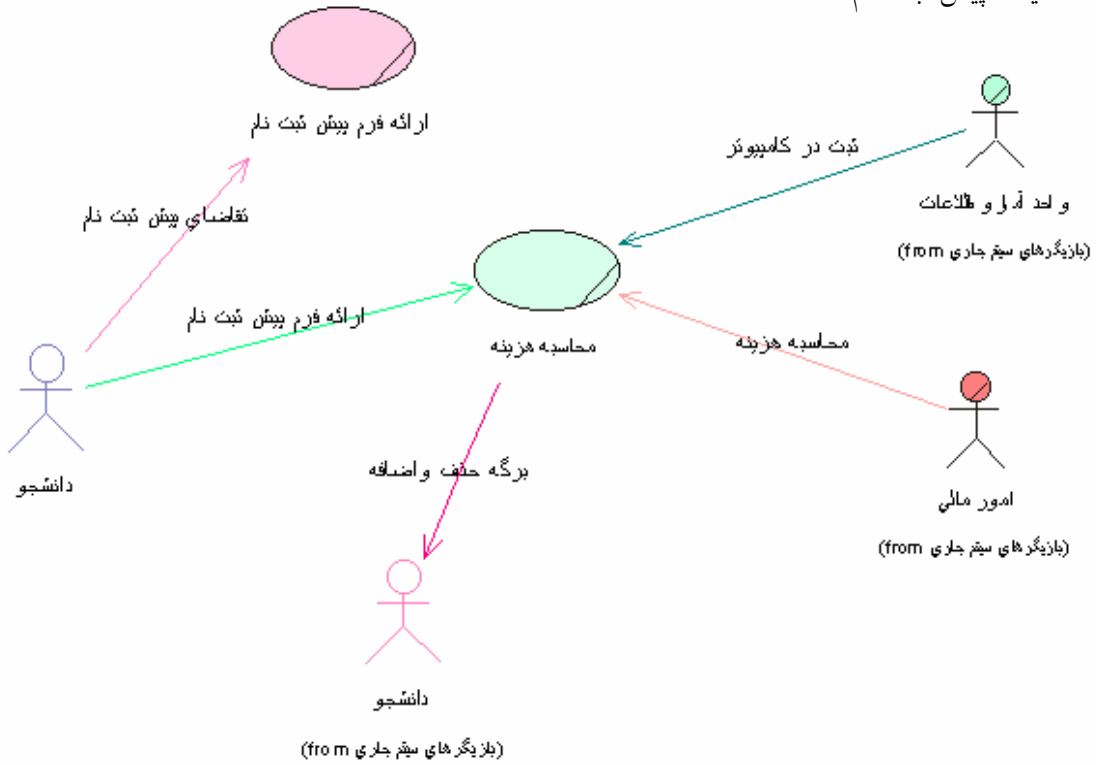
سرویسهای سیستم ثبت نام





insert use case diagram → پیش ثبت نام

عملیات پیش ثبت نام



اون use case هایی که include , extend نیستند اصلی می باشند.

سناریوی حذف و اخذ

شرح مورد های استفاده

نام مورد استفاده: حذف و اخذ

نوع: اصلی

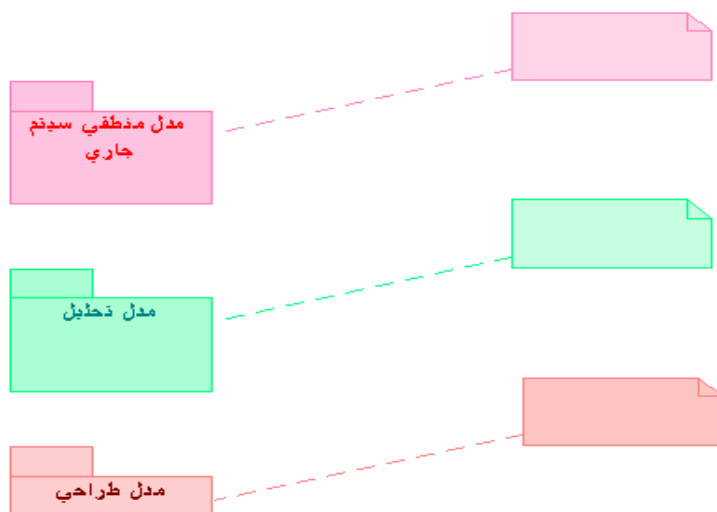
بازیگرها: دانشجو - امور مالی - واحد کار و اطلاعات

شرح: دانشجو با ارائه فرم حذف و اخذ تقاضای ثبت نام اصلی خود را می نماید. جزئیات فرم در

کامپیوتر ثبت می شود و عملیات مالی انجام می گیرد.

سیتم	بازیگر
۲- مسئول آموزش پیش نیازها را کنترل می کند	۱- دانشجو فرم حذف و اخذ را تکمیل و به مسئول آموزش می دهد.
۳- مسئول آموزش شماره دانشجویی را به سیستم کامپیوتری ثبت نام وارد می کند.	
۴- سیستم کامپیوتری جزئیات پیش ثبت نام را نمایش می دهد.	
۵- مسئول ثبت نام در صورت وجود اصلاحات را براساس فرم حذف و اخذ وارد کامپیوتر می کند.	
۶- کامپیوتر یک چک لیست جدید بر اساس اطلاعات حذف و اخذ صادر می نماید.	
	۷- دانشجو از مسئول ثبت نام مرکز ا

دیدگاه منطقی سیستم ثبت نام دیدگاه منطقی سیستم ثبت نام



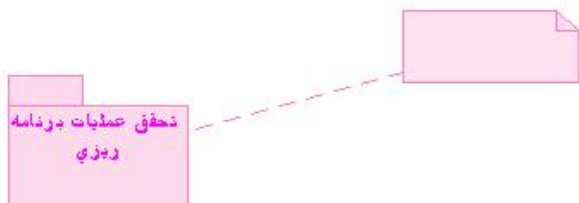
مدل منطقی سیستم جاری

تحقق موردهای
استفاده جاری

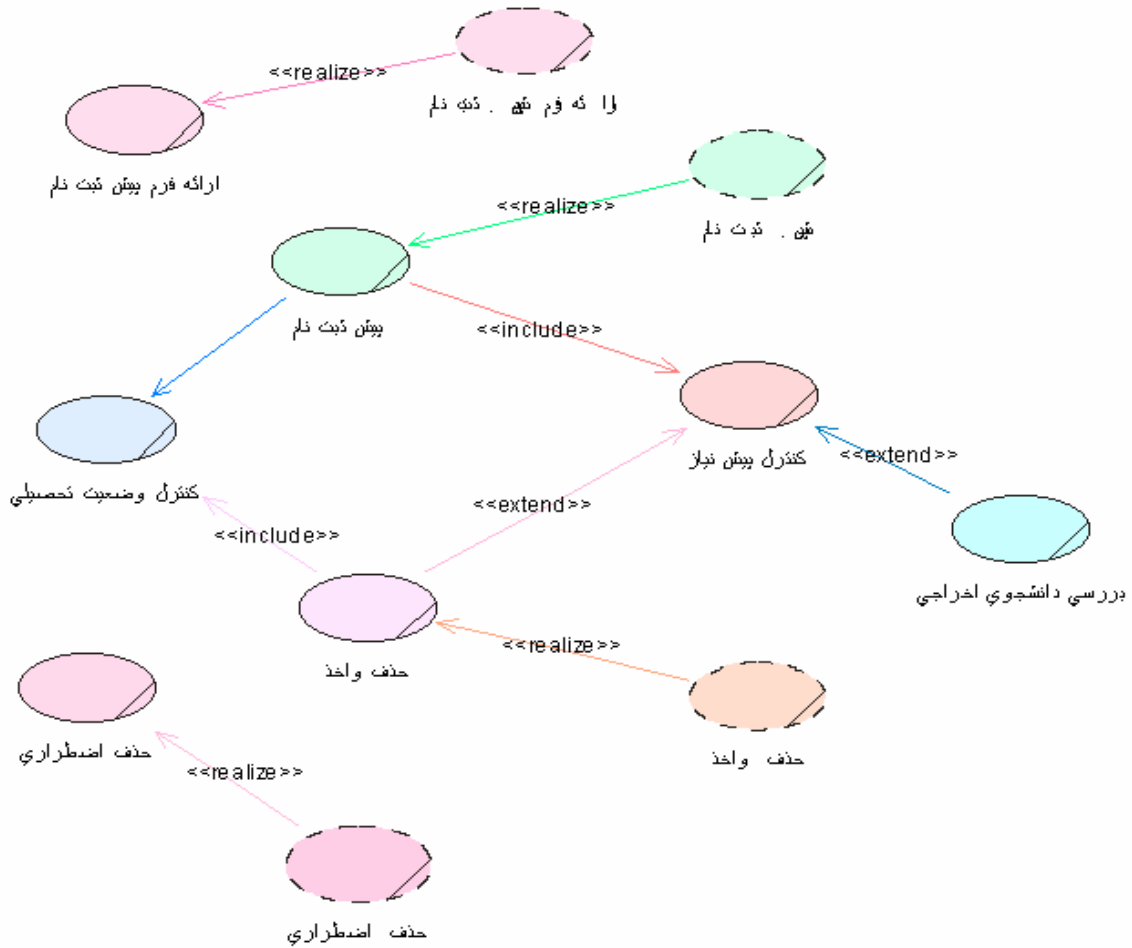
مدل منطقی سیستم جاری

مدل ارتباطی اندیاء
سیستم جاری

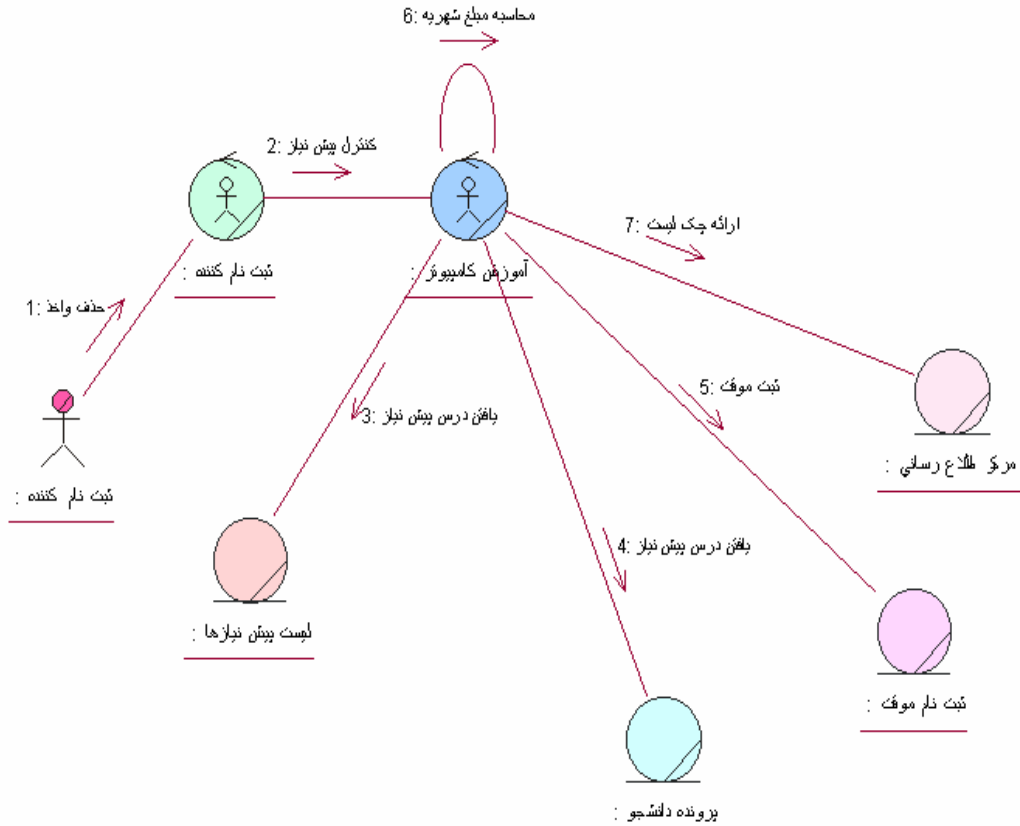
در تحقق موردهای استفاده کلیک



در تحقق عملیات ثبت نام کلیک

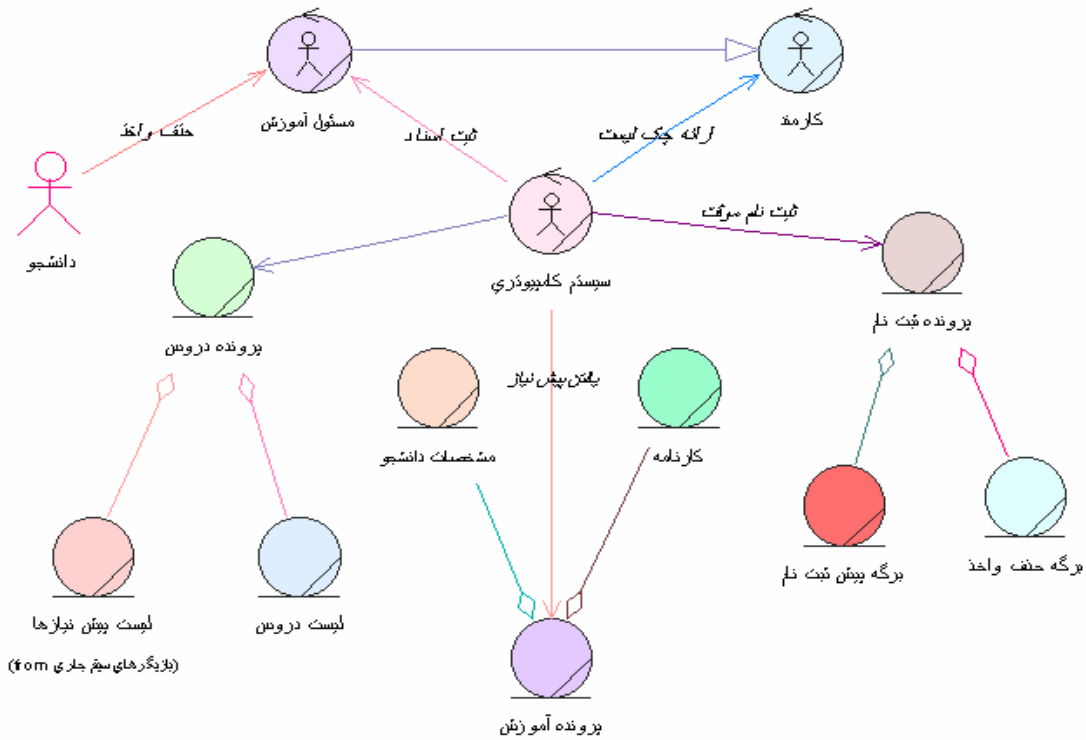


مدل همکاری برای عملیات حذف و اخذ



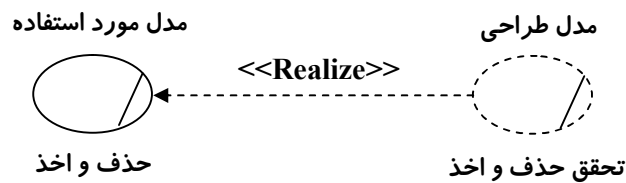
بقیه سناریو را در یک use case دیگر می کشیم چون فاصله زمانی اتفاق می افتد.

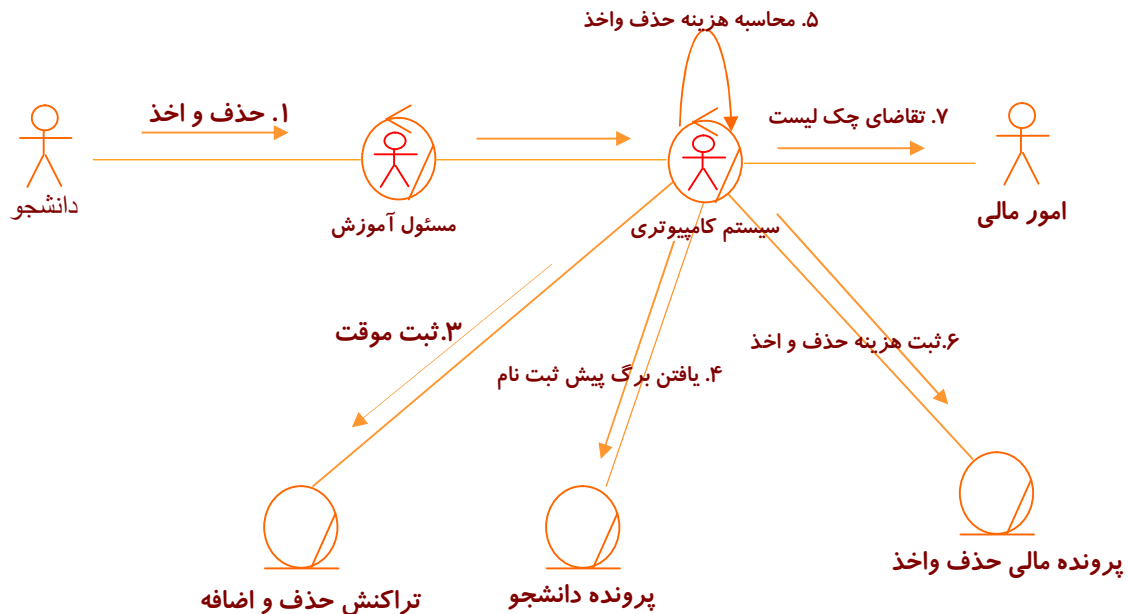
باز در realize حذف و اخذ کلیک می کنیم class alagram را رسم می کنیم.



فرق Collaboration Diagram با Business object Model؟

B.O.M خودش Package است یک Package شامل Business class Diagram و یکی دیگر شامل Business use case realization بر روی هر یک use case realization هنگامی کلیک نمودیم رفتار را در قالب Collaboration Diagram مشخص می‌کنیم یعنی نشان می‌دهیم چگونه business worker را فراخوانی می‌کنند فراخوانی‌ها را مشخص می‌کنیم تا اینکه نهایتاً سرویس use case ارائه شود در اینجا مستندات پرونده‌هایی را که مورد استفاده قرار می‌گیرد در قالب Business Entity مشخص می‌کنیم برای نمونه به مثال زیر توجه کنید.





تحلیل نیازها:

در جلسه قبل آموختیم که چگونه می توان با تعیین مدل برای سیستم جاری در واقع از عملکرد سیستم جاری شناخت پیدا کرده البته امروزه مدل سازی سیستم جاری به عنوان یک روش قدیمی و کلاسیک مطرح و با جزئیات مورد قبول نیست. یوردون در سال ۱۹۸۹ اعلام کرد که چرا سیستم جاری را مدل می کنیم و نیازی به شناخت سیستم جاری وجود ندارد. امروزه سعی می شود که در آغاز کار تأکید را بیشتر بر سیستم آتی قرار دهد و تأکیدی بر سیستم جاری و شناخت آن بدان صورت ندارد. البته باید بدانید که سیستم جاری را باید شناخت تا اینکه بدانیم بر مبنای شناخت خود نیازها را تعیین می کنیم.

نکته قابل بحث مدل سازی سیستم جاری است که امروزه بر آن تأکیدی نیست. اصولاً نیازها یا user centric (متمركز بر خواسته های کاربر) هستند و یا system centric چنانچه بخواهیم به صورت ترکیب متمركز بر سیستم عمل نماییم می بایست ابتدا شرح وظایف افراد را بخواهیم در سیستم هایی که

ارزش آنالیز کردن دارند شرح وظایف افراد موجود است و سپس بر اساس شرح وظایف و در ارتباط با هر وظیفه فرد به صورت user centric عمل نموده و نیازها " در قالب جملات سیستم باید عمل × را انجام دهد" مشخص می‌نماییم.

برای نمونه بر اساس تقسیم بندی عملیاتی در سیستم کارت ساعت واحدهای کارگزینی، واحد مدیریت، واحد نگهداری، تشخیص داده شد برای هر یک از این واحدها نیازهای عملیاتی و کیفی تعیین شده علاوه بر این نیازهای سخت افزاری و نرم‌افزاری نیز به صورت مستقل مشخص می‌شود. اینها نیازهای عملیاتی یا در مجموع functional می‌باشد. برای نمونه ارتباط با سیستم کارت ساعت کارگزینی نیازهای زیر را مشخص کرده است.

۲- کارگزینی.

نوع	نیاز عملیاتی	کد
E	سیستم باید قابلیت پذیرش مشخصات پرسنل را داشته باشد.	۲-۱
E	سیستم باید قابلیت پذیرش تغییرات برای مدت معینی داشته باشد.	۲-۲

بقیه در ص ۴۹۲ کتاب

در اینجا نوع نیاز Hidden می‌باشد که با کد H مشخص شده و یا Elvident می‌باشد که با E مشخص شد و یا optional می‌باشد. (اختیاری)

یعنی اینکه نیاز واضح و مشخص است و یا اینکه نیاز وجود دارد اما شاید علت برای شما مشخص و واضح نباشد و یا اینکه نیاز optional یا اختیاری است یعنی اینکه اگر پاسخگویی شود چه بهتر و

اگر نشد مشکلی وجود ندارد برای اینکه بتوانیم نیازها را ردیابی کنیم به هر واحد و نیازهای آن کدی تخصیص می‌دهیم تا بعداً مطمئن شویم همه نیازها از طرح سیستم در نظر گرفته شده‌اند.

پاسخ به هر نیاز در واقع یک قابلیت است یا یک (feature) قابلیت‌ها در قالب use case ها تعریف می‌شوند لذا برای هر use case باید مشخص کنیم چه نیازهایی را پوشش می‌دهد لذا در شرح Use case ها بخشی داریم بعنوان نیازهای مربوطه می‌بایست در این بخش مشخص کنیم use case ها چه نیازهایی را پوشش می‌دهند.

برای نمونه فرض کنید در ارتباط با سیستم ثبت نام تعیین شده که:

E	سیستم باید پیش نیاز دروس را در هنگام ثبت نام کنترل نماید	۳-۱
H	سیستم باید معدل دانشجوی را کنترل نماید	۳-۲
O	سیستم باید ظرفیت کلاسها را کنترل نماید	۳-۳

نام مورد استفاده: ثبت نام اولیه (موقت)

اکتورها: دانشجو

نوع: اصلی

شرح:

نیازهای مربوطه:

۳-۱

۳-۲

۳-۳

۴-۵

سناریو:

باید توجه نماییم که سرویسهای سیستم یعنی **use case**ها حتماً باید پوشش دهنده کلیه نیازها باشد یعنی در نیازهای مربوطه در مجموع همه نیازها درج شوند در غیر اینصورت سیستم ناقص است و نیازی را که درج شده و مشخص پوشش نداده است اینها نیازهای عملیاتی بودند دسته‌دیگر از نیازها، نیازهای کیفی می‌باشد.

نیازهای کیفی در واقع مبین کیفیت و چگونگی و یا در واقع شرایط انجام نیازهای عملیاتی هستند.

نیازهای کیفی معمولاً در ارتباط با یک **response lime**

۲- **Interface** (رابطها)

۱- **speed** (سرعت)

۳- **Environment** کیفی (محیط)

۳- **fault tolerance** (تحمل خطا)

نیازهای کیفی در ارتباط با **Interface** کیفیت **Interface** را مشخص می‌کند یعنی تعیین می‌کند **Interface** یا رابط کاربر چگونه می‌بایست طرح گردد اصولاً کاربرها یا حرفه‌ای هستند یا همبستگی یا بی‌تجربه و یا ناشی. کاربر حرفه‌ای بد نیست که با سیستم کامپیوتری آشنایی دارد لذا **Interface** قابلیت توسعه داشته باشد کاربر دائمی یک فردی است که دائماً سیستم را استفاده می‌کند لذا می‌بایست **Interface** برای وی امکان ورود سریع اطلاعات را داشته باشد. در صورتیکه کاربر دائمی

نباشد و بعضاً سیستم را استفاده می‌کند سیستم باید **userfreely** به سادگی قابل استفاده باشد چرا که کاربر همیشگی نیست ممکن است فراموش کند.

کاربر ممکن است ناشی باشد و برای مثال **key board** را نشناسد و جای کلیدها را بلد نباشد لذا رابط کاربر حداقل استفاده از صفحه کلید را در بر می‌گیرد.

در مورد **environment** منظور محیطی است که نرم‌افزار در آن فعال خواهد شد و بستر کامپیوتری، سیستم عامل، شبکه محیط اداری همه اینها شرایط را ممکن است داشته باشد که بر نرم‌افزار تأثیر بگذارد. مثلاً ممکن است امنیتی باشد محیط ممکن است سیستم عامل را مشروط نماید.

برای مثال سیستم عامل حتماً باید **unix** باشد در ارتباط با تحمل خطا مسأله بسیار قابل توجه است تشخیص و تحمل خطا یک مبحث نرم‌افزاری است در ارتباط با تشخیص و تحمل خطا می‌بایست باید پیش بینی کرد. که در هنگام اجرای نرم‌افزار چه رویدادهایی موجب می‌شود که نرم‌افزار با شکل مواجه شود برای افزایش میزان تحمل در مقابل خطا در این مورد پیش بینی می‌شود که چه باید کرد. برای نمونه در ارتباط با سیستم کارت ساعت این امکان وجود دارد که در هنگام بارندگی دستگاه کارت ساعتی که زیر باران است درست کار نکند. در این صورت می‌بایست دستگاه یدکی در داخل نگهداری و یا از طریق **key board** بتوان ورود افراد را مشخص کرد ممکن است خط شبکه قطع شود در اینصورت با استفاده از نرم‌افزاری بنام **M.S Messenger** می‌توان تمام اطلاعاتی را که برای شبکه ارسال می‌شود در یک صف نگهداری و در صورت اتصال خط به صورت اتوماتیک برای مقصد ارسال نمود. مسأله دیگر **response time** یا زمان پاسخگویی است.

برای نمونه ممکن است مطرح شود و اگر Query یا هر گونه پرس و جوی از سیستم شود و هرگونه شرایطی حداکثر در ظرف سه دقیقه پاسخ ایجاد شود.

در ارتباط با speed یا سرعت ممکن است که مطرح شود نمی‌خواهیم هنگامی که کارخانه تعطیل می‌شود و برای مثال 12000 کارگر به سمت درهای نگهداری هجوم می‌آورند بیش از ۵ دقیقه در صف انتظار برای ورود کارت خود قرار گیرند. پس بدین ترتیب سرعت نیز به عنوان یک نیاز کیفی مطرح است. این نیازهای کیفی تأثیری عمده بر طراحی نرم‌افزار دارد و باید حتماً در آغاز کار مشخص گردند. اصولاً تحلیل نیازها مرحله‌ای بعد از Business Modeling در فرآیند Rup است بر طبق Rup نیازها یا عملیاتی هستند که در اینصورت با عملکردها مطرح می‌شوند نیازها ممکن است در ارتباط با کارایی مطرح شوند که در اینصورت این گونه نیازها مانند نیازهای کیفی شرایطی هستند بر نیازهای عملیاتی اعمال می‌شوند و در ارتباط با سرعت بازدهی و قابلیت دسترسی مطرح هستند و ممکن است در ارتباط با قابلیت‌های پشتیبانی در ارتباط با قابلیت حمل، قابلیت ترمیم و توسعه، قابلیت تطابق با نرم‌افزارهای موجود، قابلیت آزمون سریع و پشتیبانی مطرح شوند که اینها همگی نیازهای کیفی است که مشخص می‌کند تحت چه شرایطی سیستم برای کاربر قابل استفاده خواهد بود.

حاصل تحلیل نیازها سه دسته مستند است که عبارتند از:

(۱) مستند چشم‌انداز پروژه vision

(۲) مستند درج تکمیلی supplementary specification

(۳) Use case Model یا مدل موردهای استفاده برای سیستم آتی

در بحث تحلیل نیازها سه مستند هستند به عنوان خروجی:

۱- مستند چشم‌انداز پروژه

۲- شرح تکمیلی

۳- Use cae Model

چشم‌انداز پروژه:

دیدگاهی است از قابلیت‌های پروژه که جهت بستن قرارداد و تکمیل و به عنوان یک پیشنهاد در اختیار مشتری قرارداد می‌شود.

چشم‌انداز پروژه از ابتدای پروژه تهیه می‌شود و در ابتدا به عنوان گزارش امکان‌سنجی در اختیار کاربر قرار داده می‌شود و نهایتاً در خاتمه مرحله تحلیل نیازها به عنوان چشم‌اندازی از عملکرد پروژه در اختیار کاربر قرار داده می‌شود. در چشم‌انداز پروژه انتظارات مشتری از سیستم کامپیوتری و قابلیت‌های سیستم کامپیوتری مشخص می‌گردد چشم‌انداز بر اساس نیازهای عملیاتی تعیین شده برای سیستم مکانیزه ایجاد می‌گردد برای ایجاد چشم‌انداز باید مسأله مطرح و تحلیل شود مراحل تحلیل عبارتند از:

۱- توافق در مورد مسأله‌ای که باید حل شود. ۲- تعیین کاربرها

۳- تعیین سرردهای سیستم (محدوده) ۴- تعیین محدودیت‌های اعمال شونده بر سیستم

بنابراین باید نه تنها نه پاسخ به نیازها ابتدا مسأله مشخص شود و در واقع مسأله این است که چگونه پاسخ‌گوی نیازهای باشیم بلکه می‌بایست راه‌حل‌ها در قالب قابلیت‌های سیستم مکانیزه مشخص شود معمولاً راه‌حل‌ها در قالب فرم منو مشخص می‌شود و همچنین مشخص شود که چه کسانی از این

قابلیت استفاده می‌کنند و چه محدودیتهایی سیستم بر آن اعمال می‌کند به طور خلاصه چشم‌انداز پروژه موارد زیر را پوشش می‌دهد.

۱- چرا؟ برای تعیین علت یا علت انجام پروژه‌ها موقعیتها باید مشخص شود. برای مثال موقعیت کاری نشان داده شود که باید کارگیری مسئول حاصل از پروژه چه موقعیت کاری برای متقاضی به وجود می‌آید.

باید مشکلات مشخص شود و نشان داده شود که سیستم کامپیوتری چگونه بر آنها قائل می‌آید. باید وضعیت بازار و موقعیت محصول در بازار مشخص شود به این ترتیب علت انجام پروژه مشخص می‌شود.

۲- چه کسانی؟

برای تعیین اینکه چه کسانی در ارتباط با پروژه و محصول نرم‌افزاری و محصول نرم‌افزاری حال از آن قرار می‌گیرند. افراد مرتبط با پروژه بازار محصول نرم‌افزاری شامل آمار تعداد متقاضی‌های احتمالی به کاربرها و نیازهای کلیدی آنها و پاسخ سیستم به نیازهای آنها مشخص میشود. در این ارتباط در بخش بعد، محصول و ویژگیهای آن در پاسخ به نیازها مشخص می‌شود

برای این منظور محصول را نیز توصیف کنیم در این راستا محیط عملیاتی نرم‌افزار طراحی قابلیتها و هزینه‌های لازم برای دسترسی به قابلیتها مشخص می‌شود.

۳- چه محصولی؟

مسلماً پس از ذکر مشکلات و نیازها باید محصول نرم‌افزاری و قابلیتهای آن مشخص گردد برای این منظور محیط عملیاتی خلاصه‌ای از قابلیتها و مزایای آنها برای مشتری.

۴- چگونه؟

در این راستا محدودیتهای سیستم، چگونگی محصول، کیفیت محصول، چگونگی ارزیابی آن مشخص می‌شود.

چشم‌انداز پروژه:

۱- مقدمه:

۱-۱- هدف: هدف از این پروژه چیست؟

تأمین نیازها و قابلیت‌ها

۱-۲- دامنه: مشخص می‌شود که سیستم در ارتباط با چه کسانی و یا سازمانها و یا سیستم‌های کامپیوتری دیگر قرار می‌گیرند.

۱-۳- تعاریف: در این بخش ارجاع به Dictionary لغات داده می‌شود.

۱-۴- مراجع:

۱-۵- خلاصه: در این بخش از مقدمه، خلاصه‌ای از سایر قسمتهای گزارش چشم‌انداز ارائه می‌شود.

۲- موقعیت

۲-۱- موقعیت کاری: نشان داده می‌شود که از لحاظ کاری چه موقعیت جدیدی ایجاد می‌شود.

۲-۲- مشکل: در این بخش مشکلات و مسائلی که سیستم جاری با آن مواجه است مشخص

می‌شود. برای هر مشکلی باید در ابتدا تعریف ارائه شود سپس باید مشخص شود که این

مشکل بر چه مواردی تأثیر می‌گذارد. و تأثیر آن چیست؟ نهایتاً باید مشخص شود که سیستم

مکانیزه چه مزایایی در جهت رفع مشکلات ذکر شده خواهد داشت.

برای نمونه مشکل روش قدیمی و دستی برای ثبت نام در کالج است که دانشجویان، استاد و

کارمندان آموزش را تحت تأثیر قرار داده است. کندی مراحل ثبت نام موجب عدم رضایت این

افراد شده است. وجود سیستم مکانیزه تصویر بهتری از کالج فراهم نموده و موجب رضایت کارمندان،

استاد، دانشجویان و بالنتیجه موجب جذب تعداد دانشجویان بیشتری برای کالج خواهد شد.

۲-۳ موقعیت محصول: در این قسمت مشخص می‌کنیم که چه ویژگی‌های خاصی از محصول

نرم‌افزاری موجب موفقیت آن در بازار خواهد شد.

Point of sete نقطه‌ی بسیار حائز اهمیتی برای مطرح کردن محصول در بازار می‌باشد اینکه

محصول شما در بازار نسبت به سایر محصولات موجود چه برتری دارد و نقطه‌ای است بسیار حائز

اهمیت.

۳- توصیف افراد و کاربرها:

در این بخش باید افراد که در ارتباط با پروژه هستند مشخص شوند تا تعیین شود که هر دسته از

افراد چگونه در ارتباط با سیستم قرار می‌گیرند و چه نیازهایی دارند و سیستم چگونه پاسخگو خواهد

بود. در واقع مسأله user cert بودن، را بر لذا چارت سازمانی را باید به دست آوریم البته آمار بازار

نیز در اینجا مهم است اینها افرادی هستند که مشتری سیستم ما خواهند بود اینجاست نیازهایی دارند

که در پاسخ به آنها قابلیت‌های سیستم مشخص می‌شود بنابراین هدف تعیین نیاز است.

در این راستا چارت سازمانی و بازار فروش متقاضی‌ها را مشخص می‌کند و در پاسخ به متقاضی‌ها قابلیت‌های سیستم مشخص می‌شود سیستم نیازها به چارت سازمانی و چارت عملیاتی و بازار فروش برای محصول مطرح است.

۳-۱ آمار بازار : آمار بازار در واقع مشخص می‌کند که چه متقاضی‌یابی در بازار به چه مقدار برای محصول نرم‌افزاری وجود دارد و نرخ رشد آن چیست.

۳-۲ افراد مرتبط: در اصطلاح stake holder گویند.

که به نحوی در ارتباط پروژه قرار می‌گیرند. برای این منظور می‌بایست جدول از افراد و نقش آنها در ارتباط با پروژه مشخص شود برای نمونه در جدول زیر دانشجو به عنوان نماینده دانشجویی در نقش بیانگر خواسته‌های دانشجو برای سیستم ثبت نام مشخص شد.

استاد: به عنوان نماینده هیأت علمی مسئولیت ارائه خواسته‌های دانشکده را بر عهده دارد.

نام :	نماینده	نقش
دانشجوها	دانشجوها	تأیید خواسته‌های دانشجو
استاد	هیأت علمی	ارائه خواسته‌های دانشکده
مدیر فن‌آوری اطلاعات	واحد فن‌آوری کالج	مسئولیت تصویب پروژه - کنترل کننده پیشرفت پروژه
آموزش	واحد ثبت نام	حصول اطمینان از برآورده شدن خواسته‌های آموزشی که ثبت نام را انجام می‌دهند.
	کارمندا	
	وارد کننده‌های جزئیات	

۳-۳ خلاصه کاربر

در این قسمت کاربرها دسته‌بندی شوند و شرحی از هر کدام تهیه می‌گردد.

نام	شرح	افراد مرتبط
آموزش	کنترل و ثبت اطلاعات دانشجویان و اساتید و اعلان شروع و خاتمه ثبت نام	نماینده خود

۳-۳ خلاصه کاربر

در این قسمت در واقع شرح پستهای سازمانی داده می‌شود به این ترتیب که کاربرها مشخص می‌شوند و تعیین می‌شود که هر کاربری چه عملیاتی را با سیستم انجام می‌دهد و اینکه کاربر نماینده و شاخص کدام دسته از افراد است. برای مثال به جدول زیر توجه نمایید.

نام	شرح	افراد مرتبط
آموزش	کنترل و ثبت اطلاعات دانشجویان و اساتید، اعلان شروع و خاتمه ثبت نام	نماینده خود
داشجو	ثبت نام دروس، درخواست نمره و جزئیات درس	نماینده اتحادیه دانشجویی
استاد	انتخاب دروس برای تدریس، وارد کردن نمرات دانشجو	نماینده هیأت علمی

۳-۴ محیط کاربر:

در این قسمت محیط کاری، کاربر مشخص می‌شود ممکن است کاربر قبلاً خود از سیستم کاربری استفاده کند در اینصورت باید مشخص نمود که چگونه سیستم جدید با سیستم موجود ارتباط برقرار کند و یا اینکه در صورت جایگزینی بانک اطلاعاتی موجود تبدیل به بانک اطلاعاتی قابل استفاده در سیستم جدید گردد. ارتباط سیستم جدید کامپیوتری با برنامه‌های موجود چگونه خواهد بود باید مشخص کرد که چه سیستم عاملی مد نظر کاربر است ممکن است محدودیتی وجود داشته باشد. محدودیتهای امنیتی یا محدودیتهای دسترسی، همگی اینها می‌بایست در این قسمت مشخص گردد. البته در اینجا مشخص شده است که در ارتباط با سیستم ثبت نام استفاده کننده‌ها جامعه بزرگی را

تشکیل می‌دهند، محدودیت زمان ثبت نام و سرعت پاسخگویی سیستم به تعداد زیاد ثبت نام کننده است که ممکن است به طور همزمان بخواهند از سیستم استفاده کنند. برنامه می‌بایست طوری تهیه شود که قابل نصب در دانشگاه‌های مختلف باشد.

۳-۵ وضعیت افراد مختلف:

در این قسمت برای هر کدام از کاربرها پس از توصیف نوع کاربر و اینکه شاخص چه مجموعه‌ای از کاربرها هستند مسئولیتها و معیارهای وی در ارتباط با موفقیت پروژه مشخص می‌شود همچنین نوع ارتباط هرکاربر با پروژه تعیین می‌شود برای این منظور جداول به شکل زیر تهیه می‌شوند.

نام کاربر	
نماینده	کاربر نماینده کدام دسته از افراد است
شرح	شرح خلاصه‌ای از این دسته از کاربرها در این قسمت ارائه شود
نوع	تجربه فرد، سابقه تکنیکی، نوع کاربر، ناشی مجرب، دائمی و یا گهگاه
مسئولیتها	مسئولیت‌های کلیدی کاربر در ارتباط با سیستم مکانیزه جدید
معیار موفقیت	از دیدگاه کاربر در چه صورتی نرم‌افزار موفق خواهد شد
نوع ارتباط	کاربر به چه صورتی در پروژه موثر خواهد بود برای نمونه ممکن است در تهیه لیست نیازمندیها بررسی مستندات و غیره از وی استفاده شود
محصول	چه محصولی کاربر برای چه واحدی تولید می‌کند
نکات	مشکلاتی که مانع موفقیت پروژه می‌شوند. این نکته در ارتباط با نرم‌افزارهایی است که وجود آنها کار کاربر را سخت نموده است.

برای نمونه در سیستم ثبت نام دانشگاه می‌توان سابقه افراد مرتبط با پروژه را در قالب مدیر فن‌آوری، نماینده واحد ثبت نام، استاد و دانشجو مشخص نمود. در واقع با تعیین انتظارات این افراد و اینکه معیار موفقیت پروژه از دیدگاه آنها چیست مقوله مهمی بنام *use ability* یا قابلیت استفاده در تهیه نرم‌افزار مطرح می‌گردد. برای مثال:

نام کاربر: دانشجو

نماینده دانشجویان: حسن علی

شرح: استفاده کننده

نوع: نماینده شورای صنفی دانشجویی

مسئولیتها: اطمینان حاصل می‌کند که سیستم برای دانشجویان قابل استفاده است به خصوص معیارهای سهولت و استفاده، قابلیت اطمینان و کارایی برای دانشجویان بسیار مهم است. حدوداً ۱۵٪ از دانشجویان برای اولین بار از سیستم استفاده می‌کنند و باید نماینده دانشجویان اطمینان حاصل کند که این ۱۵٪ که با کامپیوتر آشنایی ندارند بتوانند به راحتی سیستم را مورد استفاده قرار دهند در مجموع ۲۰۰۰ دانشجو از سیستم استفاده خواهد کرد که نماینده دانشجویان مسئولیت دارد که اطمینان حاصل کند سیستم برای این دانشجویان به صورت همزمان با کارایی باید قابل دسترسی است.

معیار موفقیت:

موفقیت از سهولت استفاده از سیستم و سرعت آن برای نماینده دانشجویی است

نوع ارتباط: نماینده دانشجویی، امور سیستم ثبت نام، که در ارتباط با دانشجویان هست را بررسی می‌کند.

نکات: هیچ

۳-۶ وضعیت کاربر:

در این قسمت با استفاده از جدول مشابه جدول استفاده شده برای توصیف افراد مرتبط انواع کاربرها مشخص می‌شود.

۳-۷- نیازهای کلیدی کاربر:

این قسمت بسیار مهم است. چراکه در این قسمت نیاز، مشکلات و راه‌حلهای پیشنهادی ارائه می‌شود نیازها می‌بایست دسته بندی شود برای نمونه نیازها عملیاتی هستند یا نیازها ممکن است

کیفی باشد یعنی که چگونگی نیازهای عملیاتی را مشخص کنند برای نمونه در سیستم ثبت نام، نیازهای نماینده دانشجویها و نماینده ثبت نام و نماینده هیأت علمی به صورت زیر مشخص شده است:

نیاز	اولویت	دلیل	راه حل کنونی	راه حل پیشنهادی
ثبت نام	بالا	کند بودن ثبت نام	تکلیف فرم -انظار برای حداقل دو هفته	دانشجویها می‌خواهند به صورت برخط از
دروس	نام	تغییر برنامه در این مدت نیاز به تکرار مراحل	بواسطه تغییرات - عدم موفقیت دانشجو در ثبت نام	وضعیت دروس اطلاع پیدا کرده در هر درس
نام باری کلیه دروس مورد نظر وی				

دسترسی سریع	متوسط	تأخیر دور	استاد درس را به واحد	راه حل پیشنهادی
به نمره‌ها	نمره گرفتن	ثبت نام اعلان می‌کند و دانشجو ملزم به ارجاع به دسترسی بر خط به نمرات	واحد آموزشی برای مشاهده نمرات خود است.	از طریق سیستم کامپیوتری

تقلیل پرداخت حقوق	متوسط	پرداختهای نسبتاً بالا	دو تا سه کارمند آموزش و دو تا سه نفر افراد کمکی برای مدت ۴۰ الی ۵۰۰ ساعت برای عملیات ثبت نام و همچنین از بین بردن همزمانی در ارائه دروس با متقاضی زیاد	راه حل پیشنهادی
				این کار با وجود سیستم کامپیوتری به صفر تقلیل می یابد.

۳-۸ راه حل‌های مختلف

در این قسمت لیستی از نرم‌افزارهای موجود که اهداف پروژه را برآورده می‌کنند مشخص می‌شود و تحلیل می‌شود که چرا این نرم‌افزارها مشکل دارند و چرا مقرون به صرفه است که نرم‌افزار جدید تولید گردد.

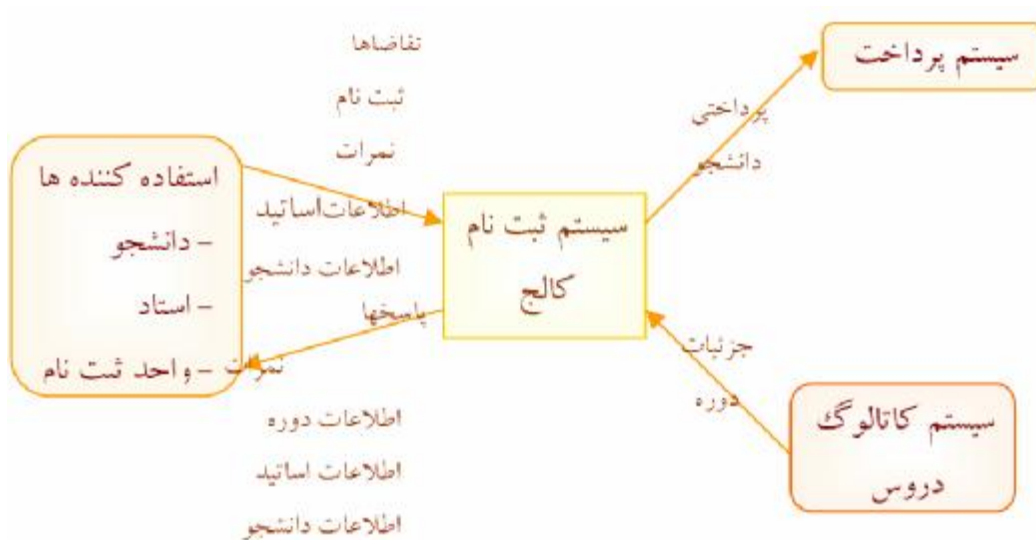
۴- شرح کلی محصول:

در این بخش قابلیت‌های محصول نرم‌افزاری مشخص می‌شود و در واقع مشخص می‌شود که ابعاد محصول چیست عملکرد محصول چیست پیش فرض‌ها در مورد محصول چیست و محصول چه وابستگی‌هایی به سایرین دارد.

۴-۱- ابعاد محصول:

در این قسمت مشخص می‌شود که آیا محصول نرم‌افزاری به صورت مستقل عمل خواهد کرد و یا اینکه بخشی از نرم‌افزار موجود خواهد بود مشخص خواهد شد که ارتباط بین نرم‌افزار شما با

نرم افزار موجود به چه صورت است این ارتباطات به صورت بلوک دیاگرام مشخص خواهد شد.
 برای مثال در مورد سیستم ثبت نام کالج ارتباط آن با سایر سیستم به صورت زیر مشخص شده است.



در واقع دیاگرام فوق مشخص می کند که چه واحدهایی در خارج از سیستم یا چه کسانی در خارج از سیستم قرار گرفته و چه نوع ارتباطاتی با سیستم دارند سیستم ثبت نام کالج در ارتباط با استفاده

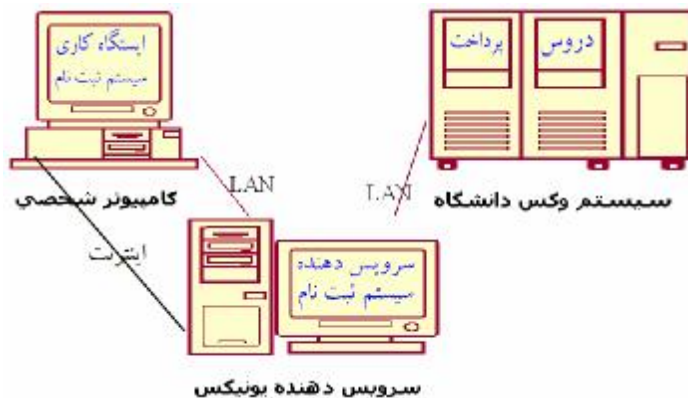
کننده‌ها اطلاعات اساتید با دانشجویان را دریافت می‌کند. اطلاعات دانشجویان، اساتید دوره، نمره و پاسخهای متفرقه را در اختیار آنها قرار می‌دهد.

سیستم کاتالوگ دروس در اینجا یک سیستم نرم‌افزاری موجود است که حاوی دروس و توضیحاتی در ارتباط با دروس است. سیستم پرداخت نیز یک سیستم موجود مالی است که سیستم ثبت نام می‌بایست جزئیات ثبت نام را جهت هرگونه پرداخت مالی در اختیار آن قرار دهد.

با تعیین نبودن ارتباطات خارجی در واقع حوزه یا عملکرد سیستم مشخص می‌شود.

نرم‌افزار سیستم ثبت نام از دو بخش تشکیل شده است:

یک بخش بر روی سرور نصب می‌شود و بخش دیگر نرم‌افزار بر روی سرویس گیرنده‌ها یا clientها نصب می‌شود. این سرویس دهنده در ارتباط با بانک اطلاعاتی سیستم‌های پرداخت و کاتالوگ دروس در کامپیوتر بزرگ و دانشگاه است.



۴-۲- خلاصه قابلیت‌های محصول:

در این بخش خلاصه ای از قابلیت‌های محصول تهیه می‌شود در واقع قابلیت‌ها عملیاتی است که سیستم انجام می‌دهد و نتایج آن را در اختیار مشتری می‌گذارد در ادامه جدولی ارائه شده که شاخص مزایای عمده سیستم ثبت نام است در واقع این جدول قابلیت‌ها و مزایای کلی سیستم ثبت نام را مشخص می‌کند قابلیت‌ها به صورت دقیق‌تر در بخش هفت مشخص شده‌اند.

مزایا برای مشتری	قابلیتها
اطلاعات به روز درس	سیستم به بانک اطلاعاتی دروس دسترسی دارد تا آخرین تغییرات دروس را در اختیار قرار دهد برای هر درس استاد و دانشجو پیش نیازها، مباحث درس، محل برگزاری و نام استاد را می‌توانند مشاهده کنند.
اطلاعات به روز ثبت نام	ثبت نامها برای هر درس در بانک اطلاعاتی ذخیره می‌شود. تابثوان در مورد برگزاری یا حذف درس تصمیم‌گیری نمود.
دسترسی سهل و سریع به نمره	اساتید نمره دانشجویها راحتی در خانه در سیستم ثبت نام وارد می‌کند. دانشجویها می‌توانند در خانه و یا در دانشگاه وارد سیستم ثبت نام شده نمره خود را مشاهده کنند.

همانگونه که در جدول فوق مشاهده می‌شود مزایای محصول باری مشتریها به تفکیک مشخص شده و در واقع در چشم‌انداز نشان داده شده که محصول چه مزایایی خواهد داشت.

۳-۴ پیش فرضها و وابستگیها:

تمامی عوامل موثر بر کارایی و قابلیت‌های محصول نرم‌افزاری را مشخص می‌کند پیش فرض سخت‌افزار و نرم‌افزار در واقع محدودیتی است که ممکن است مشتری اعمال نماید به این ترتیب قابلیت‌ها تحت تأثیر این محدودیتها قرار خواهد گرفت و برای نمونه در ارتباط با سیستم ثبت نام پیش فرضها و محدودیتها به صورت زیر هستند.

۱- فرض بر این است که سیستم کنونی کاتالوگ دروس و پرداختهای مالی که بر روی کامپیوتر

وکس قرارداد پشتیبانی شود و حداقل تا سال ۲۰۰۵ این ارتباط برقرار باشد.

۲- می‌بایست بانک اطلاعاتی موجود برای کاتالوگ دروس و پرداختهای مالی به همان ترتیبی

است که باقی بماند و هیچ گونه تغییراتی در آن اعمال نشود.

۳- تحویل به موقع نرم‌افزار وابسته به پرداخت به موقع به مشتری است.

۴- ۴- هزینه‌ها و قیمت‌ها:

محدودیت‌های مالی مسلماً تأثیر مستقیم بر قابلیت‌ها و کارایی محصول دارد باید این محدودیت‌ها را

مشخص نمود و برای مثال محدودیت‌ها جهت بخش محصول به رابطه کمبود بودجه.

۴-۵ مجوز نصب:

مجوز نصب می‌تواند تأثیر مستقیم بر یک برنامه داشته باشد برای نمونه ایجاد کلمه عبور و یا

مجوزهای ارتباطی در شبکه بار برنامه‌نویسی را افزایش می‌دهد.

به طور خلاصه در ارتباط با امنیت کنترل دسترسی، مجوزها، در این قسمت بحث می‌شود چرا که اینها

همگی بار کاری ایجاد می‌کنند.

۵- قابلیت‌های محصول:

در این قسمت لیستی از قابلیت‌ها مشخص می‌شود باید توجه نمایید که منظور از قابلیت توانایی‌های

سطح بالای سیستم است که برای کاربر سودمند خواهد بود. سطح بالا بردن قابلیت‌ها به این مفهوم

است که وارد جزئیات نشوید مشخص کنید که فقط چه قابلیت‌هایی وجود دارد برای نمونه در زیر لیست

قابلیت‌های سیستم ثبت نام مشخص شده البته این بخشی از قابلیت‌ها است لیست کامل در مستندات

Rup موجود است.

۵-۱-۱ ورود:

سیستم قابلیت کلمه عبور نام کاربر را خواهد داشت افراد قادر به تعویض کلمه عبور خود باشند.

۲-۵ ثبت نام دروس:

سیستم باید امکان ثبت نام را به دانشجو بدهد. پیش نیاز ظرفیت کلاسها توسط سیستم چک می شود.

۳-۵ حذف دروس:

باید کارمندیهای آموزش امکان حذف دروس را داشته باشد درسی که به اندازه کافی در آن ثبت نشده باشد به طور اتوماتیک توسط سیستم حذف می شود.

صورت حساب دانشجوها مشاهده مشخصات اساتید، مشاهده کارنامه، انتخاب دروس برای تدریس، درج جزئیات دانشجوها با ثبت نمره دانشجوها، مشاهده لیست دروس، مشاهده برنامه هفتگی دروس در مجموع قابلیتهای سیستم را تشکیل می دهند که برای هر یک توضیحی در این قسمت داده می شود.

۶- محدودیتها:

هرگونه محدودیتی که بر سیستم وارد می گردد در این قسمت می بایست مطرح شود برای مثال محدودیتها در مورد سیستم ثبت نام این است که محدودیت سخت افزاری - دسترسی به بانک اطلاعاتی مجزا می باشد.

۷- دامنه های کیفیت:

در این قسمت موارد کیفی مثل قابلیت استفاده، تحمل خطا و یک پارچگی نرم افزار مطرح می گردد برای نمونه در سیستم ثبت نام کیفیتها به صورت زیر مشخص شده:

۱- دسترسی: سیستم باید در تمامی مدت شبانه روز قابل دسترسی باشد.

۲- بهره‌وری : سیستم برای افراد غیر کامپیوتری به سادگی قابل استفاده خواهد بود.

۳- قابلیت ترمیم سیستم طوری طراحی شود که به سادگی قابل ترمیم شود.

۸- اولویتها:

در این بخش برای هر یک از قابلیت‌های تعیین شده اولویت مشخص می‌شود. بر اساس اولویت‌ها به قابلیت‌ها به دست می‌آید و ارزیابی می‌شود.

۹- سایر انتظارات از محصول:

در این قسمت به صورت کلی استانداردهای مورد استفاده نیازهای سخت‌افزاری و ایستگاههای کاری نیازهای کارایی و نیازهای محیطی مشخص می‌شود.

۹-۱ استانداردها

در این قسمت کلیه استانداردهای مورد استفاده مشخص می‌شود. این استانداردها برای نمونه می‌توانند در ارتباط با پروتکل‌های شبکه، سیستم‌های عامل، ایمنی و کیفیت باشد. برای مثال در ارتباط با سیستم ثبت نام مشخص شد که در رابطه کاربر می‌بایست قابل نمایش بر روی windows 2000 باشد.

۹-۲ نیازهای سیستمی:

باید هر گونه امکانات سخت‌افزاری و نرم‌افزاری و شبکه مورد نیاز سیستم مشخص شود. برای نمونه در ارتباط با سیستم ثبت نام سیستم باید بانک اطلاعاتی موجود برای جزئیات دروس در ارتباط قرار گیرد. سیستم باید با سیستم حسابداری موجود ارتباط برقرار کند.

۹-۳ نیازهای کارایی:

در ارتباط با کارایی سرعت انتقال داده‌ها مطرح است. نیازهای کارایی در برگیرنده قابلیت اطمینان است. سیستم باید امکان دسترسی توسط ۲۰۰۰ را داشته باشد. سیستم باید حداقل در دو ثانیه به اطلاعات مورد نیاز دروس دسترسی پیدا کند. سیستم باید حداکثر در عرض ۲ دقیقه ۸۰٪ از اطلاعات مورد نیاز از بانک اطلاعاتی بخواند.

۹-۴ نیازهای محیطی:

در واقع معین تأثیر محیط بر محصول نرم‌افزاری است. چنانچه نرم‌افزار درون سخت‌افزار گنجانده شود تا عملیاتی را کنترل کند برای سایر نرم‌افزارها عواملی مثل درجه حرارت امواج الکترومغناطیسی، تشعشع و شوک اینها همگی می‌توانند بر عملکرد تأثیر بگذارند.

۱۰- مستندات

در این قسمت انواع مستنداتی که برای سیستم باید ایجاد شود مشخص می‌شود. مستندات شامل سؤال کاربر، راهنمای online راهنمای نصب فایل‌های و بر چسب گذاری و بسته‌بندی می‌باشد.

۱۰-۱ منوال کاربر

چگونگی استفاده از محصول نرم‌افزاری را برای کاربر مشخص می‌کند در هر مورد توضیح کامل فراهم کرده است. این منوال باید شامل جزئیات زیر باشد. معمولاً منوال کاربر بر طبق استانداردهای مشخص شده تعیین شود این استانداردها در مورد اندازه، مباحث، سرفصل‌های موجود در یک منوال

کاربر (user Guide)

۱۰-۲ راهنمای برخط online help:

حتی امروزه زبانهای برنامه‌سازی امکان افزودن **help** برای هر قطعه‌ای از برنامه را فراهم کرده‌اند باید با ایجاد راهنما که ضمیمه برنامه است در ضمن اجرای برنامه بتوان از امکانات **help** یا راهنما استفاده کرد.

۱۰-۳ راهنمای نصب:

در واقع راهنمای نصب مشخص می‌کند که سیستم به چه صورت **setup** یا **Install** شود امروزه زبانهای برنامه‌سازی شامل **VB** امکان ایجاد نسخه قابل نصب را فراهم کرده است برای مثال **Install sheet** ابزاری جهت تولید نسخه‌های قابل نصب است.

۱۰-۴ برجسب گذاری و بسته‌بندی:

در این بخش آرم شرکت کامپیوتری (آرم مشتری) انحصاری بودن محصول، امتیاز استفاده از آن مشخص می‌شود. برای مثال در سیستم ثبت نام آرم کالج باید درج شود.

۱۰-۵ بررسی چشم‌انداز پروژه:

در این قسمت به بررسی و ارزیابی مستند چشم‌انداز پروژه می‌پردازیم و مشکلات و معایب را مشخص می‌کنیم تا اینکه یک ساختار قابل قبول برای چشم‌انداز تهیه کنیم اما ببینیم که چه چیزهایی می‌بایست دقیقاً باید مشخص شود که در چشم‌انداز به اینصورت عمل نشده است.

۱- شرح نیازها:

در چشم‌انداز پروژه، همانگونه که در بالا مشاهده کردیم نیازها در بخشهای ۷-۳ در قالب نیازهای کلیدی کاربر به طور عمده مشخص شده جدا از این منظور جدولی به صورت زیر مشخص شد.

نیاز	اولویت	دلیل	راه حل کنونی	راه حل پیشنهادی
------	--------	------	--------------	-----------------

مشکل این است جدول نوع نیاز را مشخص نمی‌کند نیاز را به درستی توضیح نمی‌دهد بر طبق استاندارد Up می‌بایست نیازها در قالب جملات سیستم باید عمل X را انجام دهد مشخص گردد. برای این منظور جداولی به صورت زیر تهیه می‌شود ابتدا چارت عملیاتی را مشخص می‌کنیم و سپس برای هر واحد عملیاتی نیازها را خیلی خوب تعیین می‌کنیم.

نیازها باید از دید افراد مرتبط با سیستم همانگونه که در چشم‌انداز پروژه تعیین شده مشخص شود اما این کم است نیازها باید از دیدگاه واحدهای عملیاتی نیز مشخص شود و تأیید گردد برای نمونه در ارتباط با سیستم ثبت نام عملیات به سه دسته عملیات دانشجویی، عملیات مربوط به هیأت علمی و عملیات مربوط به امور آموزش تقسیم می‌شود که از دیدگاه هر کدام نیازها باید مشخص گردد.

چارت عملیاتی سیستم موجود نیز این کمک را به ما می‌کند برای مثال برای عملیات ثبت نام که آن چارت عملیاتی وجود دارد نیازها را از دیدگاههای کاربرهای مختلف یعنی دانشجو، استاد و واحد آموزشی باید بررسی کرد و در زیر عملیات ثبت نام در یک جدول تعیین نمود. برای هر دسته از عملیات می‌بایست یک کد مشخص گردد و برای هر نیاز تحت آن واحد عملیاتی زیر کد تعیین شود برای نمونه عملیات واحد کارگزینی دارای کد ۲ است و هر یک از نیازهای تحت این عملیات زیر کد خود را دارد.

۲- کارگزینی:

کد	کارگزینی	نوع
----	----------	-----

E	سیستم باید قابلیت پذیرش مشکلات پرسنل را داشته باشد	۲-۱
E	سیستم باید قابلیت پذیرش تغییرات را برای مدت معین داشته باشد	۳-۲

همانگونه که مشاهده شده نیازها یا functional هستند یا عملیاتی Non prnctial یا به عبارتی کیفی برای هر واحد عملیاتی نه تنها نیازهای عملیاتی باید مشخص شود بلکه نیازهای کیفی نیز باید مشخص گردد اما یک دسته‌بندی دیگر برای نیازها وجود دارد نیازها یا بر اساس عملکرد سیستم موجود مشخص می‌شوند که اینها نیاز به تایپ ندارد. نیازها اگر بر اساس خواسته‌های افراد و برای آنچه باشد که سیستم انجام می‌دهد این نیازها غیر معقول هستند (غیر معقول می‌توانند باشند) لذا لازم است که تأیید گردند در جدول ستونی جدید ایجاد می‌کنیم و نوع نیازها را جدید (ج) یا قدیم (ق) و رد شده (ر) مشخص می‌کنیم.

۲- کارگزینی

کد	نیاز عملیاتی	نوع	رده
۲-۱	سیستم بار قابلیت پذیرش مشخصات پرسنل را داشته باشد	E	ج

بنابراین به جای بند ۷-۳ بهتر است که مطالب کتاب نیازهای عملیاتی و کیفی با توسعه‌ای که در بالا داده شد مشخص می‌شود و قبل از آن چارت عملیاتی مشخص گردد. افراد مرتبط با پروژه رده‌بندی خوبی است که در چشم‌انداز مشخص شده و به نظر می‌رسد که این بخش ۵-۳ است بخش مناسبی می‌باشد و باید در مستندات جدید ما بگنجد اما بعد از اینکه نیامندیها را با کاربر مشخص نمودیم به تهیه کاتالوگ نیازمندیها می‌پردازیم این کاتالوگ بر مبنای نیازها ساخته می‌شود و به شرح زیر است.

فرم : مشکلات و نیامندیها	تاریخ ۱/۱/۱
پروژه: مطب پزشکی	تهیه کننده: سعید پارسا
تایید کننده: کارمند کوچولو	
شرح مشکل : نیاز اثرات سوئ: راه حل پیشنهادی مزایا طرح پیشنهادی به فرم ضمیمه ارجاع شود.	

طرح پیشنهادی در قالب فرم و منو مشخص می شود. برای نمونه برای مثال فوق در داخل VB یک منو برای فیش تهیه می کنیم یکی از گزینه های آن حذف پرونده های راکده است و در فرم ضمیمه مشخص می کنید که پرونده های راکد با چه جزئیاتی نمایش داده می شوند تا منشی آنها را به صورت کلی و یا تک به تک حذف نماید.

شماره پرونده	نام بیمار	شرح خلاصه	حذف
۱۱۱-۸۲	حسینی	بیماری یرقان	

باید توجه کنید که هر یک از این فرمها نمایانگر یک قابلیت و use case سیستم است که use case در بخش دیگری در داخل مدل Rutional Rose ایجاد می شوند قبل از اینکه به بیان

نیازمندیها بپردازیم در گزارش خود نام پروژه، نام مشتری و هدف از انجام پروژه را مشخص کنیم در هیچ مکانی از چشم‌انداز از هدف مشخص نمی‌شد در بخش مقدمه هدف از چشم‌انداز مشخص شد و نه هدف از انجام پروژه هدف چیست؟

هدف شاخص یک رابط منطقی بین آنچه هست و آنچه که در آتیه به دست خواهد آمد. برای بیان هدف مشخص کنید که مشکل چیست و سیستم نرم‌افزاری چگونه بر آن فائق می‌آید. برای نمونه هدف حال حاضر به واسطه دستی بودن عملیات ثبت نام هزینه و وقت زیادی مصروف عملیات ثبت نام می‌گردد با استفاده از یک سیستم کامپیوتری از این نوع هزینه‌ها و اتلاف وقت می‌توان جلوگیری به عمل آورد.

هدف این است که دانشجو بتواند مستقیماً عمل ثبت نام را انجام دهد به قسمی که سیستم بر اساس سال ورود دانشجو به دانشگاه تشخیص دهد که در چه ترمی قرار دارد و برنامه‌ریزی از قبل تعیین شده برای آن ترم را بر اساس کارنامه دانشجو برای وی نمایش دهد دانشجو آن را تصحیح کند. پس از بیان هدف، چارت سازمانی را باید مشخص کرد تا مشخص شود که چه کسانی در ارتباط با سیستم قرار می‌گیرند و سپس به شرح مسأله می‌پردازیم.

شرح مسأله بر اساس کتاب OMT می‌بایست نکته اصلی را بپوشاند.

۱- چه کسانی؟ در این قسمت از شرح مسأله مشخص می‌شود که حوزه عملکرد سیستم چیست؟ در واقع با کشیدن یک Contact دیاگرام مطابق آنچه که در چشم‌انداز پروژه برای سیستم ثبت نام دیدیم مشخص می‌کنیم که چه واحدها و سازمانها و افرادی در خارج از سیستم قرار می‌گیرند.

چرا هر کدام از اینها نیازهایی را قاعدتاً مطرح خواهند کرد برای مثال ما در ارتباط با سیستم ثبت نام کالج سیستم پرداخت، سیستم کاتالوگ دروس، استفاده کننده‌ها شامل دانشجو تا استاد و واحد ثبت نام شاخص حوزه عملکرد سیستم بوده‌اند.

در واقع نشان می‌دادند که چه کسانی یا چه واحدهایی در ارتباط با سیستم کامپیوتری ثبت نام قرار می‌گیرند و سپس برای هر کدام از اینها نوع اطلاعات رد و بدل شونده مشخص شد در شرح مسأله ابتدا دیاگرام متن قرار می‌گیرد و سپس برای هر موجودیت خارجی استفاده کننده (دانشجو، استاد، واحد آموزش، سیستم پرداخت و سیستم کاتالوگ دروس) نیازها به صورت کلی مشخص می‌گردد.

بنابراین تا اینجا مشخص کردیم چه کسانی چه نیازهایی دارند بند سه و شرح مسأله پاسخ سیستم به نیازهاست یعنی use case شامل قابلیت‌هایی هستند که به صورت کلی در شرح مسأله و در پاسخ به نیازها گنجانده می‌شوند و بالاخره قلم چهارم در شرح مسأله نکته فروش سیستم است یعنی باید مشخص کنیم چه سیستم چه قابلیت یکتایی دارد که آن را نسبت به سایر سیستم‌های موجود ارجحیت خواهد داد.

در بالا آموختیم که مستندات چشم‌انداز پروژه یکی از محصولات مرحله requirement یا تحلیل نیازهاست اما گفتیم که این مستندات به طور دقیق و یک جا نیازها و قابلیت‌ها را مشخص نمی‌کند روش RPM که در کتاب مشخص شده برای تعیین نیازهای هر واحد عملیاتی پیشنهاد شد پس از تعیین چارت عملیاتی باید برای هر واحد عملیاتی نیازهای عملیاتی و کیفی مشخص شود، اما در مورد قابلیت‌ها، قابلیت‌ها در مواقع پاسخ به نیازهاست در واقع همان سرویس‌ها یا Use case های

سیستم کامپیوتری می‌باشد که می‌بایست حتماً در برگیرنده نیازها باشد در این راستا ابتدا کاتالوگ نیازمندیها تهیه می‌شود در این کاتالوگ موارد زیر مشخص می‌شود.

۱- نام نیاز یا مشکل

۲- شرح نیاز یا مشکل

۳- اثرات سوء (ریالی)

۴- راه حل پیشنهادی

۵- مزایا (ریالی)

۶- طرح پیشنهادی (در قالب فرم و منو)

بدین ترتیب از مشکلات و نیازها به طرح سیستم کامپیوتری می‌رسیم. برای اینکه مطمئن شویم که مشکل را درست درک کرده‌ایم و راه حل ما کامل است از **Prototyping** یا نمونه‌برداری سریع استفاده می‌کنیم به این ترتیب که سریعاً منو و فرمهای پیشنهادی را پیاده‌سازی می‌کنیم البته بدون بانک اطلاعاتی و با ارائه **Interface** از کاربر می‌خواهیم که ببیند آیا کامل و قابل قبول است حالا می‌توانیم **use case** ها را مشخص کنیم در واقع یکی دیگر از محصولات مرحله تحلیل نیازها **use case view** می‌باشد یا **use case view** دیدگاه موردهای استفاده در اینجا موردهای استفاده از سیستم آتی یا در واقع همان سیستم کامپیوتری را مشخص می‌کند.

موردهای استفاده از سیستم کامپیوتری:

یکی دیگر از محصولات مرحله تحلیل نیازها **use case view** است **use case model** یا مدل موردهای استفاده بیانگر عملیات مورد نظر برای سیستم کامپیوتری است و در واقع به عنوان یک

قرارداد بین مشتری و تولید کننده استفاده می‌شود چرا که مشخص می‌کند سیستم کامپیوتری چه سرویسهایی در اختیار کاربرهای خود قرار خواهد داد.

مدل موردهای استفاده به عنوان ورودی اصلی برای مرحله تحلیل و طراحی مورد استفاده قرار می‌گیرد.

در **use case model** در واقع مشخص می‌کنیم که سیستم کامپیوتری چه سرویسهایی به کاربر می‌دهد لذا می‌بایست این سرویسها مورد تأیید کاربر قرار می‌گیرد در واقع مدل موردهای استفاده مشخص می‌کند که چه عملیاتی یا چه سرویسهایی سیستم فراهم می‌کند.

چگونگی پیاده‌سازی در مراحل بعد مشخص می‌شود. این مدل برای تعیین معماری نرم‌افزار نیز موثر است. شما باید دقت کنید که در **use case Model** کلیه نیازهای سیستم کامپیوتری درج شود برای این منظور باید در توصیف **use case**ها مشخص کنیم که هر **use case** کدام نیازها را مشخص می‌کند برای مثال مورد استفاده ثبت نام را در نظر بگیرید. در هنگام مصاحبه با افراد مشخص شده که سیستم باید پیش‌نیازها را کنترل کند - سیستم باید سقف تعداد واحدها را رعایت نماید - حالا می‌بایست مورد استفاده ثبت نام در توصیفش مشخص شود که این نیازها را پوشش می‌دهد برای همین منظور است که به هر نیازی یک شماره داده می‌شود.

برای هر **use case** می‌بایست مشخص شود که چه نیازی را پوشش می‌دهد در غیر اینصورت آن **use case** قابل قبول نیست چرا که هیچ نیازی برای آن وجود نداشته.

نیازها باید کد عددی در توصیف use case مشخص می‌گردند. یک use case ممکن است چندین نیاز را پوشش دهد اگر برای use case نیازی مشخص نگردد آن use case لزومی برای وجودش نیست.

موردهای استفاده از سیستم کامپیوتری در واقع عملکرد سیستم کامپیوتری را مشخص می‌کند لذا می‌بایست اولاً برای هر فرم ورودی - خروجی که توسط کاربر گشوده می‌شود یا دسترسی می‌شود یک use case در نظر بگیریم و ثانیاً عملیات Administer و عملیات پشتیبانی که در سیستم کامپیوتری مورد نظر است به عنوان Use case ها یا موردهای استفاده از سیستم کامپیوتری در نظر گرفته شود. باید توجه نمایید که سناریو برای use case های کامپیوتری در واقع توصیفی است از عملکرد برنامه لذا باید برای برنامه‌نویسی و تحلیل‌گر قابل درک باشد. برای این منظور عملیات delete, update, Read - write می‌بایست دقیقاً در شرح use case مشخص گردد و همچنین اسامی table و اسامی object ها در داخل سناریو مشخص باشد. برای مثال در سیستم ثبت نام که در Rational Rose, Package مستندات آن ارائه شده دارای use case های زیر می‌باشد.

Maintain student Information	نگهداری اطلاعات دانشجو
submit Grades	درج نمرات توسط اساتید
select courses to teach	انتخاب درس برای تدریس
Register for courses	ثبت نام برای دوره
maintain Professor Information	نگهدار اطلاعات اساتید
login	لاگین

close Registration

بستن ثبت نام

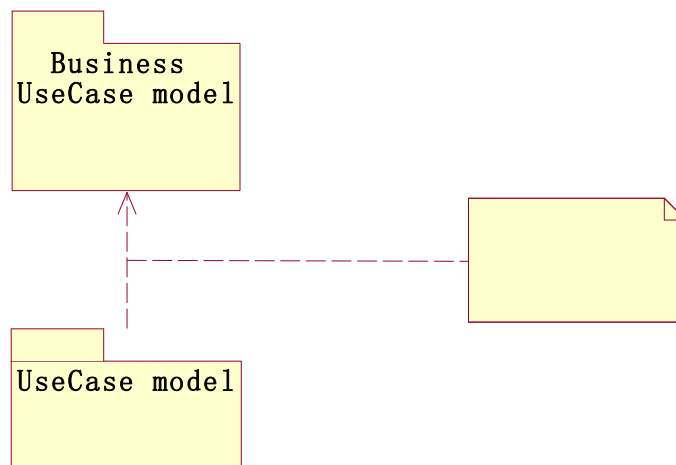
view Report card

مشاهده کارت گزارش

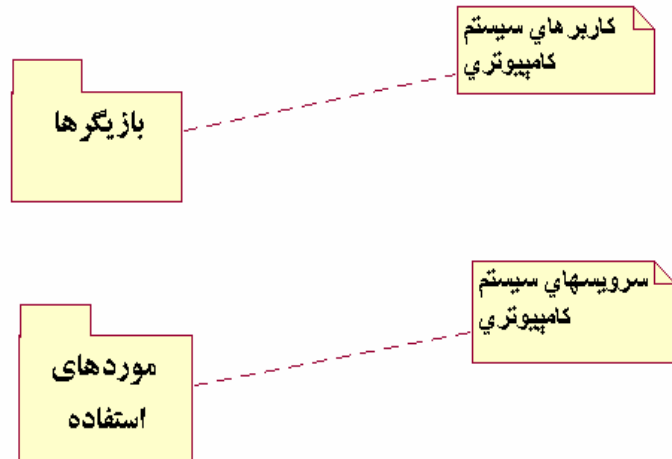
جهت ترسیم use case Diagram از مرورگر یا Browser گزینه use case view را انتخاب

و تحت آن use case Model به عنوان یک Package اضافه می شود این package در کنار

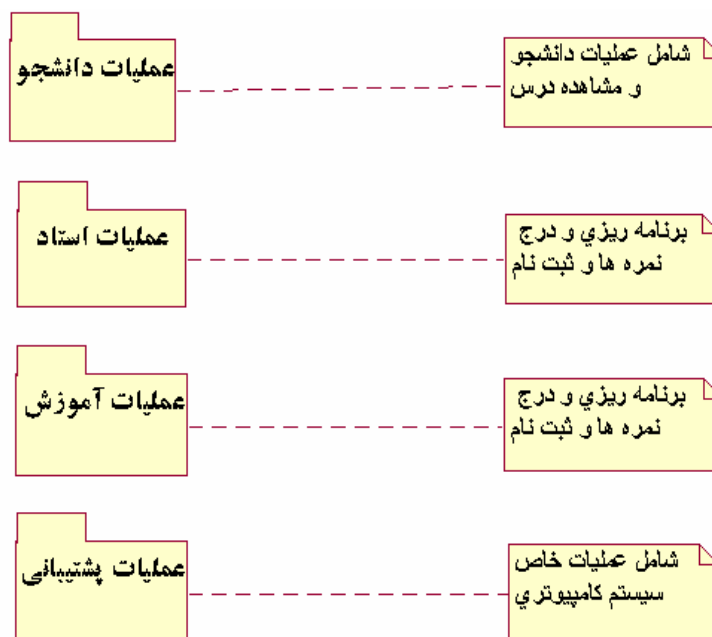
package سیستم کاری یا جاری یعنی Business use case Model قرار می گیرد.



با ورود به یک Package, use case Model های زیر ظاهر می شود



با فشردن روی موردهای استفاده عملیات به چها رسته به صورت زیر تقسیم می شود

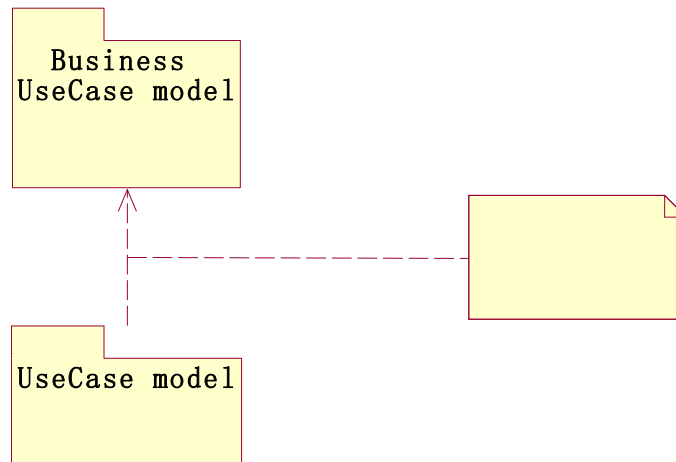


برخی از عملیات یک use case ممکن است در یک زیر سیستم دیگر انجام گردد برای نمونه ثبت نام خود در سیستم کامپیوتری دانشگاه را در نظر بگیرید. توسط یک use case دانشجو دروس مورد نظر خود را مشخص می کند توسط use case دیگر که در واحد آموزش تحت عملیات آموزش انجام می شود کارمند آموزش ثبت نام اولیه را تعیین می کند.

Use case ثبت نام و package ای تحت عنوان عملیات دانشجو مشخص شده و use case دومی تحت عنوان تأیید ثبت نام توسط کارمند آموزش به انجام می رسد اما در اینجا use case سومی هم وجود دارد که عملیات آن در واحد مالی انجام می گیرد و آن تولید فیش پرداختی برای ثبت نام است و use case چهارم در واقع تأیید می کند و use case پنجم تعرفه ثبت نام نهایی را در اختیار ثبت نام کننده قرار می دهد این عملیات در واحد فرایندی انجام می گردد بنابراین مشاهده می کنیم که برای انجام یک use case ثبت نام use case های متفاوت در واحدهای متفاوت فعال می شود. سؤال اینجاست که چگونه می توان نشان داد که این use case ها در جاهای مختلف انجام می شوند تا عمل ثبت نام انجام گردد.

در ادامه می خواهیم برای عملیات ثبت نام در سیستم کامپیوتری use case view را مشخص کنیم.

وارد use case می شویم با فشردن use case view شکل زیر ظاهر می شود.

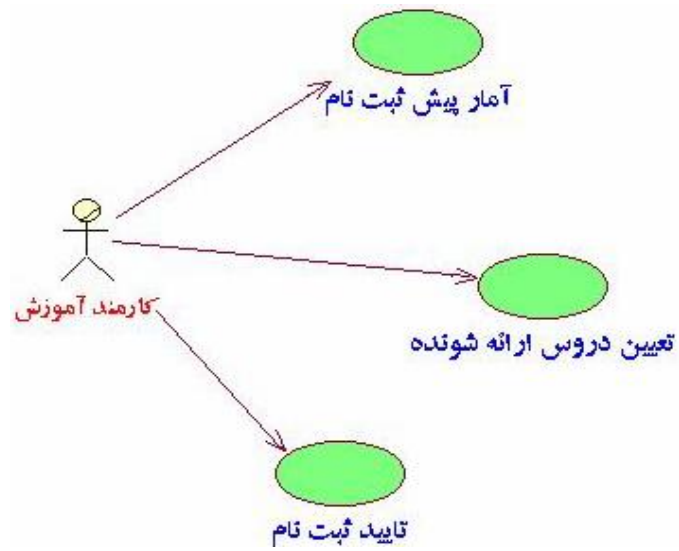


اکنون تحت use case Model باز هم در Package قرار می‌گیرد یکی نمایانگر کاربرها یا بازیگرهای سیستم کامپیوتری و دیگری شاخص مدل موردهای استفاده است.

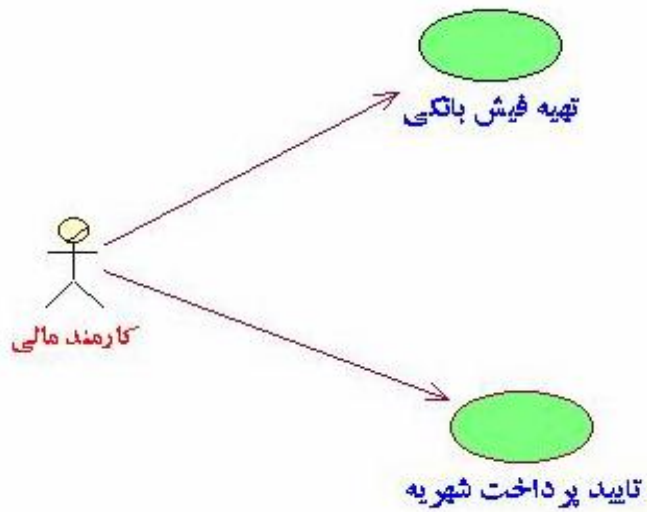


در داخل مدل مورد استفاده آتی اکنون عملیات سیستم ثبت نام را خود بر اساس مفهوم بودن و نزدیک بودن عملیات با یکدیگر مشخص می‌کنیم.

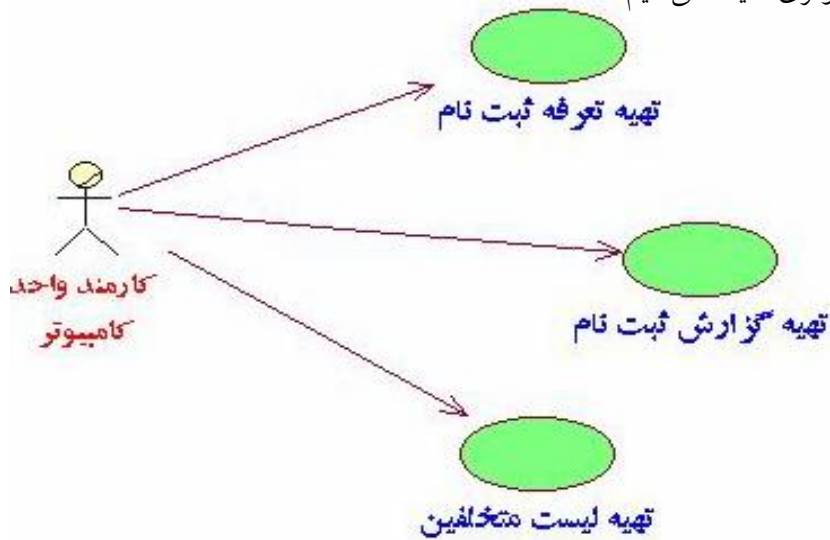
در عملیات آموزشی کلیک می کنیم.



در عملیات مالی کلیک می کنیم.

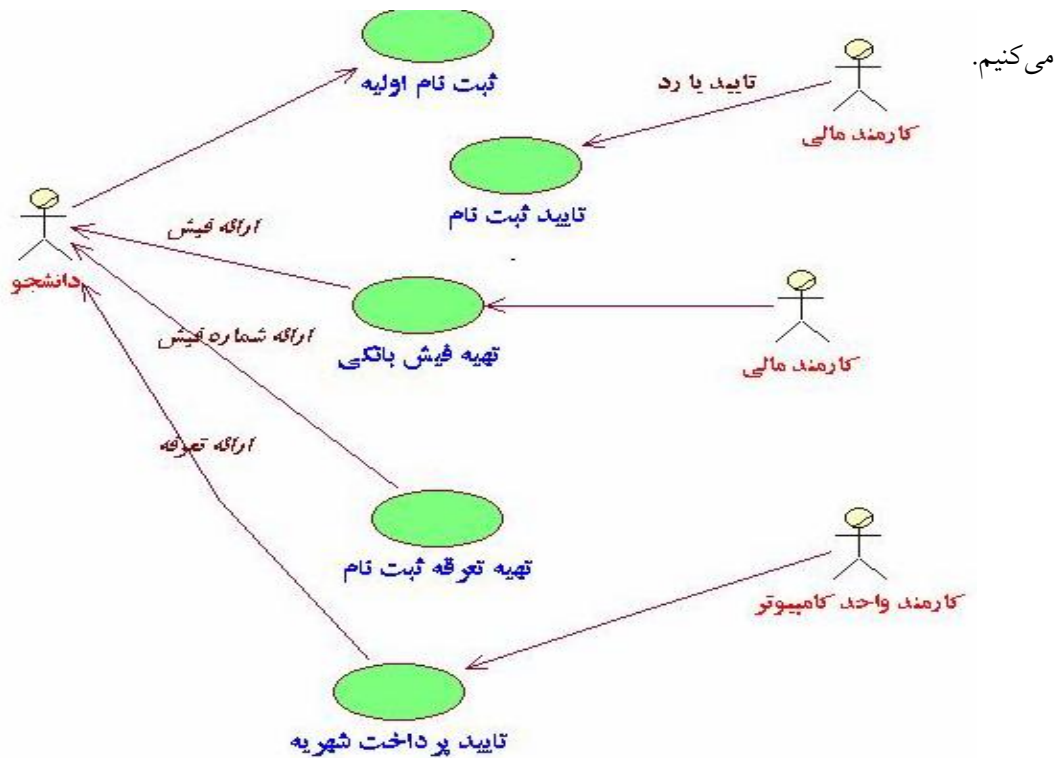


در عملیات واحد کامپیوتری کلیک می کنیم.



اکنون نکته قابل توجه ثبت نام دانشجو است که برای آن یک use case Model می کشیم تا مشخص می کنیم که برای ثبت نام عملیات گسترده ای انجام می گردد.

با کلیک روی use case ثبت نام یک فرم خالی باز می شود گزینه Diagram را انتخاب و سپس با Right click کردن در بخش خالی یک منو باز می شود گزینه use case Diagram را انتخاب



همانگونه که در شکل فوق مشاهده می‌کنید توانستیم یک عملیات گسسته را در قالب یک سرویس برای سیستم کامپیوتری ارائه می‌شود در قالب یک use case Model مشخص کنیم همانگونه که مشاهده می‌کنیم هر بخش عملیات در یک واحد عملیاتی جداگانه انجام می‌گردد تا اینکه نهایتاً عملیات ثبت نام به انجام رسد.

اکنون می‌بایست برای مورد های استفاده سناریو در صورت لزوم activity Diagram تهیه گردد. برای این منظور می‌بایست یا از قالب بندی ارائه شده توسط Rup استفاده کنیم و یا از قالب زیر روی ثبت نام اولیه کلیک بعد منوی files را انتخاب بعد راست کلیک بعد insert file در word سناریو می‌نویسیم.

شرح مورد های استفاده

نام مورد استفاده : ثبت نام اولیه

نوع: اصلی

اکتور: دانشجو

شرح: هدف از این مورد استفاده ثبت نام اولیه است.

نیاز های مربوطه: ۱-۴ و ۱-۷

سناریو :

بازیگر	سیستم
<p>۱- دانشجو گزینه ثبت نام اولیه را از منوی دانشجو انتخاب می کند</p>	<p>۲- سیستم تاریخ کنونی را با فاصله زمانی مجاز برای ثبت نام کنترل می کند.</p>
<p>۴- دانشجو شماره دانشجویی خود را وارد می کند</p>	<p>۳- سیستم تقاضای ورود شماره دانشجو را می کند.</p> <p>۵- سیستم از پرونده های دانشجو وضعیت تحصیلی وی را که یکی از موارد اخراج یا فارغ التحصیل، در حین تحصیل و اخراجی را مشخص می کند.</p> <p>۶- سیستم از فایل کارنامه دانشجو شماره ترم جاری را برای وی مشخص می کند.</p> <p>۷- سیستم از پرونده فایل برنامه ترم به ترم با در نظر گرفتن شماره ترم جاری برنامه ترم دانشجو را مشخص می کند.</p> <p>۸- سیستم برای هر درس در برنامه ترم پیش نیاز را از پرونده دروس مشخص می کند.</p> <p>۹- سیستم پیش نیازها را کارنامه دانشجو پیدا می کند.</p> <p>۱۰- سیستم برنامه ترم را برای دانشجو نمایش می دهد.</p>
<p>۱۱- دانشجو برنامه را تأیید می کند</p>	<p>۱۲- سیستم برنامه ترم دانشجو را در پرونده ثبت نام اولیه قرار می دهد.</p>

موارد دیگر:

مورد ۲: چنانچه درفاصله مجاز زمانی برای ثبت نام نباشیم سیستم پیام موعده ثبت نام نیست را مشخص می‌کند.

مورد ۵- چنانچه دانشجو در حین تحصیل نباشد پیام ثبت نام مجاز نیست ظاهر شود.

مورد ۹- چنانچه دانشجو پیش نیازهایی نگذارنده باشد سیستم در برنامه‌های ترم پیش‌نیاز را جایگزین درس می‌کند.

۱- هر use case معادل یک session است.

۲- معمولاً کنترل دسترسی یک use case است که توسط کلیه use case ها یا include

می‌شود و یا اینکه از طریق بانک اطلاعات مورد دسترسی قرار می‌گیرد.

کلیه عملیات مدیریتی system Adintunation ارتباطات شبکه‌ای در نظر گرفته شود.

توجه نمایید که ارتباط بین Use case ها معمولاً By Default از طریق بانک اطلاعات انجام

می‌شود. use case ها معمولاً ارتباط مستقیم ندارند مگر رابطه‌ای extend, include

۳- در شرح Use case های کامپیوتری اسامی tabel ها اسامی object به دستورالعمل

Replace, Delet, write, Read باید دقیقاً مشخص گردد تا برنامه‌نویس با ارجاع به

شرح use case بتواند به دسترسی کار خود را به انجام رساند.

اکنون در داخل محیط Rational Rose بر روی مورد ثبت نام اولیه کلیک می‌کنیم گزینه files را

انتخاب در محوطه خالی زیر آن Mouse را Rightclick نمود. Insert files را انتخاب و بدین

وسیله فایل word را که شرح use case در آن قرار گرفته اضافه می‌کنیم. برای هر use case یک

فرم ورودی و خروجی باید مشخص گردد که در زیر آن در داخل فایل word قرار داده می‌شود و هر use case حتماً باید پوشش دهنده‌های چند نیاز باشد. یعنی از نیازهای مربوط شماره نیازهایی را که use case پوشش می‌دهد می‌بایست مشخص گردد. همچنین هر use case به عنوان یک قابلیت می‌بایست در مستندات چشم‌انداز پروژه مشخص شده باشد.

Use case view کامل می‌شود و وارد مرحله تحلیل و طراحی می‌شویم. البته use case به عنوان بخشی از تحلیل یا آنالیزیم مشخص شده اند در گام بعدی می‌بایست در آنالیزیم مدل تحت logical view برای هر use case Realization use case مشخص کنیم و برای هر realization دیاگرام‌های مربوطه که در ادامه توضیح داده خواهد شد بکشیم.

تحلیل و طراحی

در مرحله تحلیل و طراحی ابتدا مشخص می‌شود که چه امکاناتی سیستم می‌بایست در اختیار کاربر قرار دهد این حاصل مرحله تحلیل است و در مرحله طراحی مشخص می‌گردد که این سرویس‌هایی که قرار است سیستم فراهم کند و امکانات آن چگونه می‌بایست ایجاد گردد بنابراین مرحله تحلیل مشخص می‌کند چه مرحله طراحی مشخص می‌کند چگونه.

در مرحله در واقع طراحی، معماری سیستم نیز مشخص می‌گردد که در مورد انواع معماری‌ها توضیح کافی داده خواهد شد. بنابراین به طور خلاصه می‌توان گفت که هدف از این مرحله

الف: تبدیل نیازها به طرح سیستم آتی

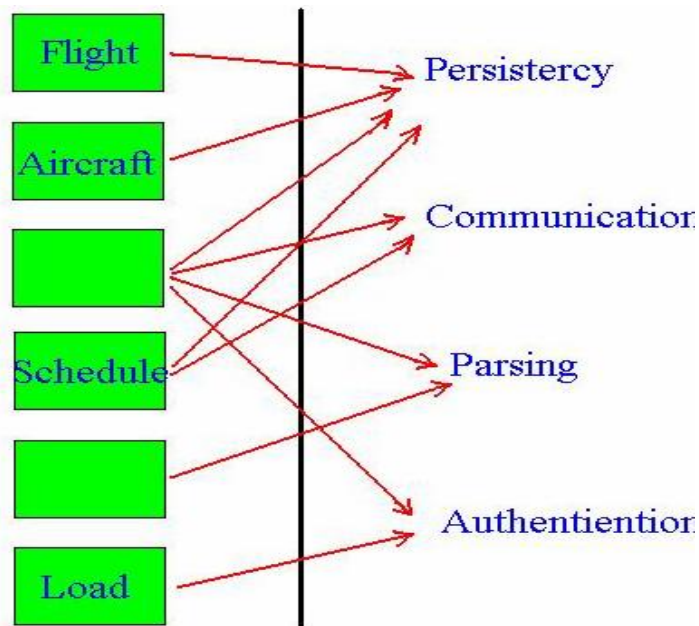
ب: توسعه یک معماری جامع برای سیستم.

ج: مطابقت دادن طراحی با محیط پیاده‌سازی به واسطه افزایش کارایی است.

مکانیزم تحلیل

در واقع بیانگر تکنیک‌هایی است که می‌بایست برای پیاده‌سازی نرم‌افزار مد نظر داشت. در واقع اینها مسائل کلی هستند که می‌بایست برای آنها راه حل کلی ارائه شود در واقع با تعیین این مکانیزم‌ها کار طراحی و پیاده‌سازی نرم‌افزار ساده می‌گردد. برای مثال : مکانیزم‌هایی برقراری ارتباط در بین ایستگاههای کاری، مکانیزم‌هایی را برای نگهداری اطلاعات در حافظه، اینها مکانیزم‌هایی عمومی هستند که به مسأله خاص مربوط نمی‌شوند. بلکه می‌بایست برای آنها یک راه حل عمومی در نظر گرفت به این ترتیب با تعیین این مکانیزم‌ها کار طراحی ساده‌تر می‌شود.

مکانیزم‌های تحلیل در مرحله طراحی به صورت دقیق‌تری بیان می‌گردد و در واقع اینها حاصل تجربیات و قابلیت‌های یک تیم نرم‌افزاری می‌باشد. از جمله این مکانیزم‌ها می‌توان مکانیزم‌های دادن مجوز برقراری ارتباط، نگهداری در انواع حافظه‌ها را نام برد باید مشخص کرد که هر کدام از کلاسها در ارتباط با کدام یک از مکانیزم‌های تعیین شده قرار می‌گیرد. برای نمونه در شکل زیر چهار مکانیزم تحت عنوانهای *persistency* (پایداری در حافظه) *cum unication* (ارتباطات) ، *Parsing* (تحلیل خوبی) ، *Authen fication* (تعیین مجوز) قرار گرفته، کلاسهای پرواز هواپیما، *Mission* و مأموریت) (زمان‌بندی) ، *Route* (مسیرهای پرواز) ، *load* (بازکردن اطلاعات) در ارتباط با مکانیزم‌های شناخته شده فوق قرار گرفته‌اند. در اینجا منظور *per sistently* این است که اطلاعات لازم است که در حافظه *RAM* دسترسی و در حین حال تغییرات آنها بر روی دیسک ضبط شود، نگهداری اطلاعات در داخل *RAM* سرعت دستیابی را افزایش می‌دهد.



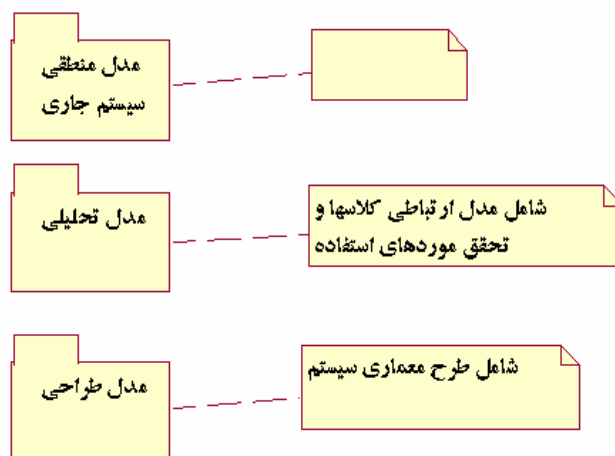
بدین وسیله باتعین مکانیسم‌ها در واقع ابزار لازم یا ابزار عمومی برای طرح و پیاده‌سازی سیستم‌ها مشخص می‌گردد. اگر به محیط Ralianal Roze وارد شوید و گزینه Rup را انتخاب کنید مشاهده می‌کنید که تحت logical view آنالیز مدل business object Model و Design Model قرار گرفته حالا در داخل package آنالیز مدل دو Package قرار می‌گیرد که در یک package، use case realization و داخل package دیگر class diagram قرار داده می‌شود در use case realization به ازای هر use case از use case view یک یا چند use case realization قرار می‌دهیم و در واقع هر use case realization شدن آن یا به واقعیت پیوستن آن ارتباط می‌دهیم ممکن است برای هر use case چند طرح مختلف داشته باشیم. به ازاء هر طرح یک realization قرار می‌دهیم حالا برای هر use case realization یک Diagram و callohran diagram و یک class Diagram می‌کشیم و به این ترتیب آنالیز نیز مدل بنا می‌شود. در ادامه با ذکر مثال این مطالب را تفهیم می‌نماییم.

مراحل تحلیلی و طراحی:

اجزاء معماری را در برنامه شیء گرا کلاسها تشکیل می دهند. ارتباط و همکاری پویا بین کلاسها توسط دیاگرامهای همکاری (Collaboration Diagram) و یا دیاگرامهای توالی مشخص می شود. با تعیین دیاگرامهای همکاری می توان کلاسها و مدل ارتباطی آنها را مشخص نمود. کلاسها از متن سناریوی use case ها مشخص می شود. توجه کنید که اگر برای یک سیستم طرحها و پیاده سازیهای مختلف در نظر گرفته شود مدل آنالیز ایجاد می گردد از مدل آنالیز تحت Package بنام آنالیز نیز مدل قرار می گیرد دو package قرار می گیرد یک Package بنام use case realization و دیگری بنام مدل ارتباطی کلاسها می باشد و برای اینکه use case ها را realize کنیم یعنی آنها را به واقعیت ببندیم آنها را با نمادهای بیضی شکل به صورت خط چین استفاده می کنیم در حالت کلی برای نمونه به شکل زیر توجه گردد.

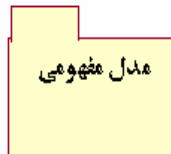
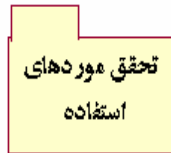


برای نمونه تحت logical view به صورت زیر سه Package قرار می گیرد.



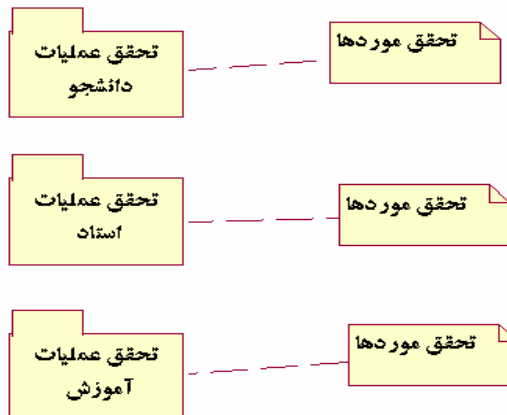
پس از کلیک کردن بر روی مدل تحلیل صفحه زیر ظاهر می شود.

مدل حاصل از تحلیل سیستم ثبت نام



در تحقیق موردهای استفاده کلیک می کنیم.

تحقیق عملیات ثبت نام

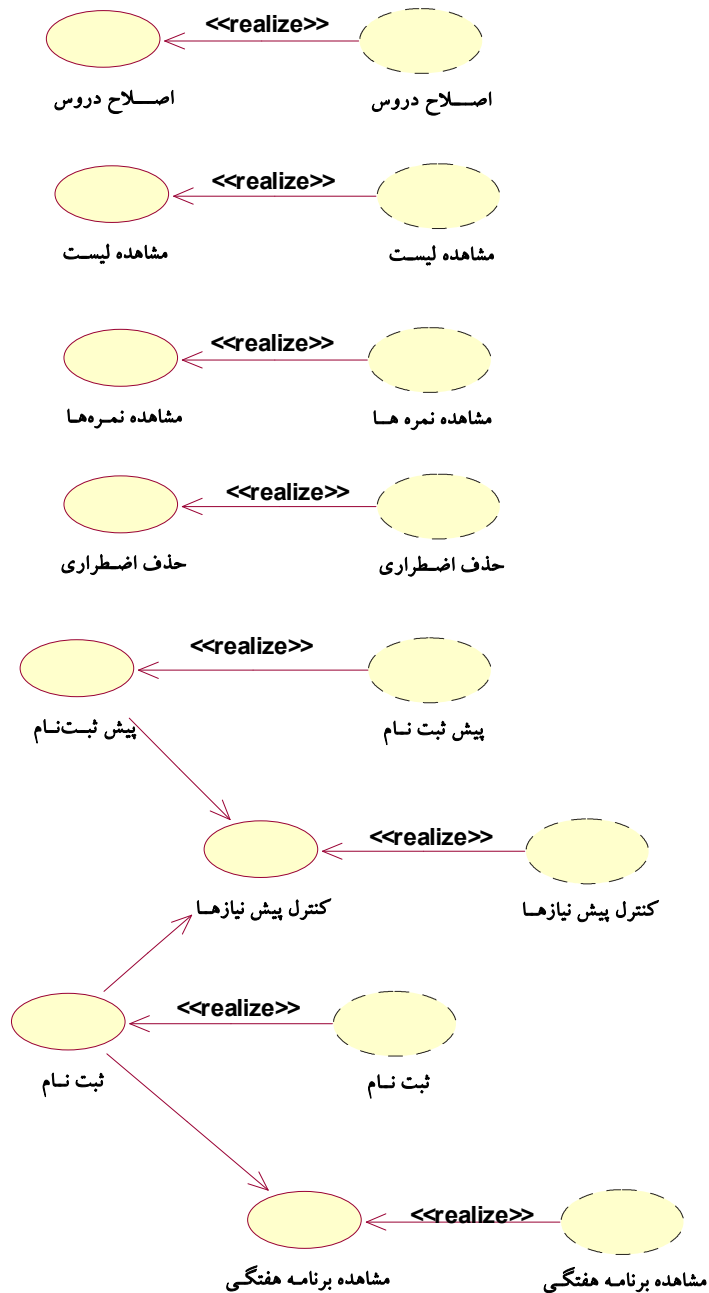


در تحقیق عملیات دانشجو کلیک می کنیم. اگر توجه نمائید مشاهده می کنید که بر اساس ساختار use

case view که شامل سه package بود سه Package برای تحقیق موردهای استفاده مشخص شد

حالا برای عملیات دانشجو use case Realization رامشخص می کنیم که در اینجا امری زائد است

چرا که به ازای هر use case فقط یک realize موجود است.



توجه کنید که ساختار به کار برده شده برای use case realization مشابه ساختار use case realization, use case برای آن دیاگرام view است حالا در مرحله بعد با click بر روی هر realization, use case برای آن دیاگرام توالی، دیاگرام همکاری و مدل ارتباطی کلاسها را ترسیم می‌کنیم برای نمونه در شکل زیر پس از

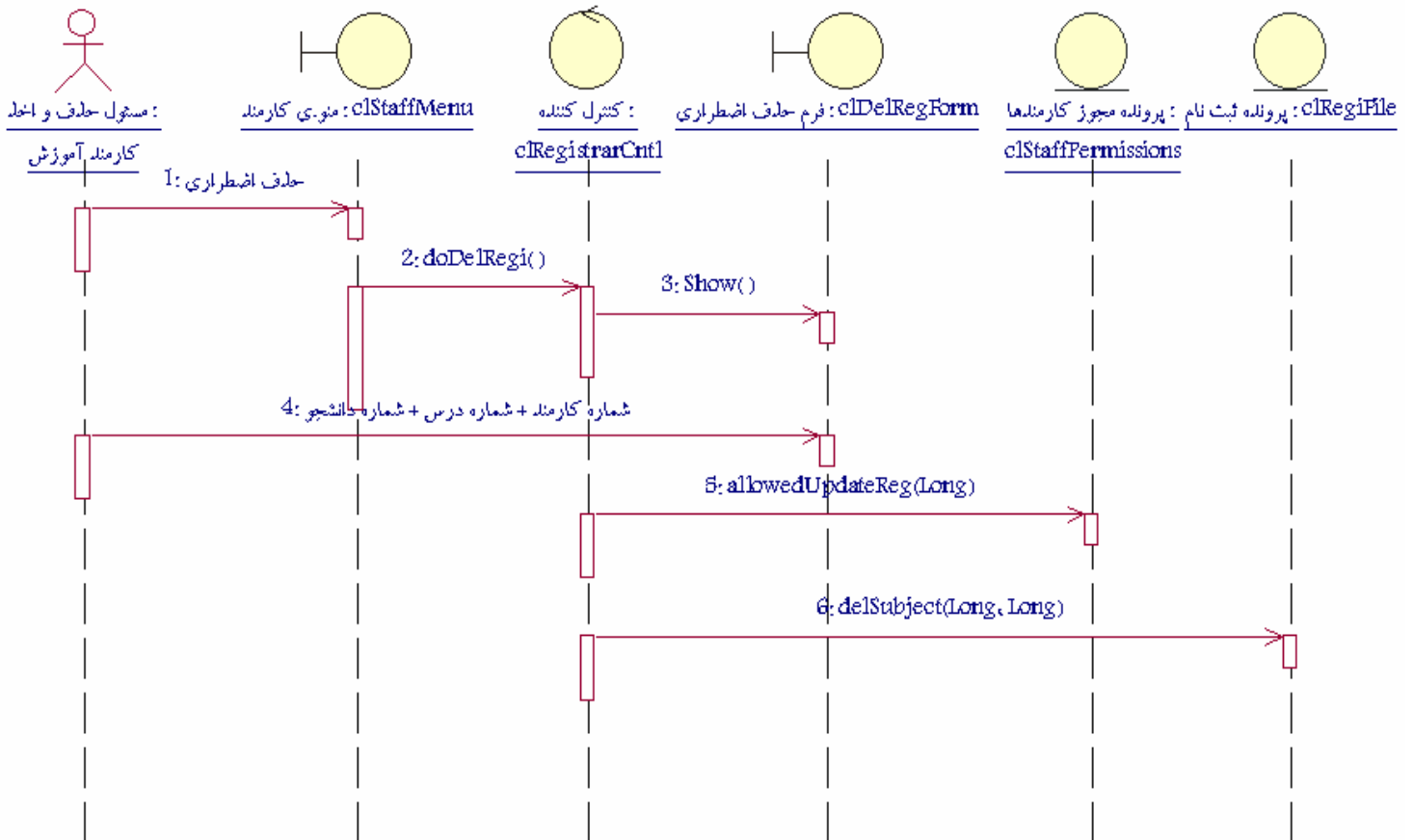
click بر بیضی خطچین که شاخص تحقق مورد پیش ثبت نام است دیاگرام توالی به صورت زیر ترسیم می شود.

دیاگرامهای توالی

دیاگرامهای توالی همانند دیاگرامهای همکاری (collaboration, diagram) بدین منظور استفاده می شود که نشان دهنده سناریوی use case یا عملیات آن در سیستم کامپیوتری به چه صورت به واقعیت می پیوندد برای این منظور می بایست اشیاء و کلاسهای مربوط به آن مشخص گردد در شکل زیر یک دیاگرام همکاری برای حذف اضطراری ارائه شده است. ابتدا بر روی مورد استفاده حذف اضطراری کلیک می شود در قسمت دیاگرام و داخل کادر سفید ماوس Right click می شود و گزینه insert sequence Diagram تولید می شود در sequence Diagram می بایست ابتدا Actor را باید مشخص کرد در اینجا یک object یا شی اضافه می شود بنام مسئول حذف و اخذ از منوی کارمند آموزش.

بعد از اینکه یک object از toolbar کنار صفحه انتخاب کردیم و بر روی صفحه قرار دادیم بر روی آن object کلیک می کنیم و نام آن را منوی دانشجو می گذاریم. و کلاس آن را یک کلاس جدید نام cl student menu انتخاب می کنیم و با انتخاب کردن این کلاس جدید یک صفحه باز می شود نام کلاس مشخص می گردد. بنام cl student Menu نوع کلاس boundary (سرحدی) انتخاب می شود اولین interface برای کاربر در قالب یک منو مشخص می شود معمولاً بر روی خطوط ارتباطی برای منوها، گزینه ای که از منو باید انتخاب شود تعیین می شود در اینجا گزینه حذف اضطراری است.

همانگونه که در بالا مشاهده می‌نماییم **object** مسئول حذف و اخذ از کلاس کارمند آموزش گزینه حذف اضطراری از شیء منوی دانشجو از کلاس **cl student menu** را انتخاب می‌نماید. با این انتخاب بلافاصله مدیریت حذف و اخذ که یک کلاس کنترلی است مورد فراخوانی قرار می‌گیرد.



اکنون برای کنترل عملیات حذف اضطراری یک شیء کنترل کننده بنام کنترل کننده از نوع کلاس **clregisral cntl** مشخص می‌کنیم کار کلاسهای کنترلی نظارت و کنترل عملیات است این کلاسها در واقع مثل یک مدیر نظارت‌گر بر عملیات‌اند در واقع این وابسته به طراح است که استفاده از کلاس

کنترلی را جایز بدانند یا خیر. بعدها هر گونه کنترل در عملیات **use case**ها از طریق این کلاس کنترلی اعمال می‌شود. در اینجا با استفاده از یک **object message** که یک فیش پر می‌باشد ما ارسال پیام از منوی کارمند به کنترل کننده را مشخص می‌کنیم اگر این پیام یک فراخوانی باشد فراخوانی متدی از یک کلاس کنترل کننده آنگاه می‌بایست بر روی **object message** یعنی فلش ارتباطی **Right click** کنیم و گزینه **new operation** را انتخاب می‌کنیم و یا اینکه در **operation**های موجود برای کلاس کنترلی انتخاب کنیم تا در واقع پیام ارسال گردد .

این توضیح را می‌نویسیم:

این متد از کلاس کنترل کننده عملیات واحد آموزش بر عملیات حذف و اضطراری نظارت می‌نماید. مقدار برگشتی از این متد شماره دانشجویی، دانشجو است.

برای اینکه **class dodelreg** را ببینیم روی کنترل کننده کلیک می‌کنیم بعد **Browse class** را انتخاب بعد **operation** را انتخاب می‌کنیم بعد متن را باز می‌کنیم.

شی کنترل کننده با فراخوانی متد **show** از **object** فرم حذف اضطراری موجب می‌شود که فرم خالی برای حذف اضطراری بر روی صفحه نمایش داده شود در این جا می‌بایست کد درس، کد دانشجو، کد کارمند وارد شود تا حذف اضطراری انجام گردد. چون این یک داده است این بار بر روی خط اتصالی بین مسئول حذف و اخذ و فرم حذف اضطراری **Right click** نمی‌کنیم و صرفاً **left click** می‌کنیم.

اکنون با در دست داشتن شماره کارمند از طریق کلاس کنترل کننده در پرونده مجوز کارمندا یک شی از جنس **staff permission** است می‌بایست دسترسی نمود و اطمینان حاصل کرد که کارمند

برای عملیات ثبت نام و حذف و اخذ مجاز می‌باشد. برای این منظور متد **new operation** در **Name: allowed up date Reg** فراخوانی می‌شود این تابع مقدار برگشتی‌اش از نوع **Boolean** است. با ارجاع به پرونده مجوزها اطمینان حاصل می‌کند که عملیات اصلاح جزئیات حذف و اخذ و ثبت نام برای کارمند مجاز است.

اما این تابع یا به عبارت دیگر این متد از کلاس مجوزهای کارمندا پارامتری نیز دارد. این پارامتر شماره کارمندی، کارمند است که به عنوان پارامتر برای این متد یا در اصطلاح تابع مشخص می‌شود. هر متدی ممکن چند پارامتر داشته باشد برای این منظور گزینه **Details** را انتخاب می‌کنیم و در قسمت سفید نام پارامترهای با **Right click** و انتخاب گزینه **insert** مشخص می‌کنیم و نام پارامتر را **staffnumber** می‌گذاریم و کلیک می‌کنیم.

Name: staffnumber

Type: long

در قسمت **Document** این توضیح را می‌نویسیم:

این پارامتر شماره کارمندی مسئول حذف و اخذ را مشخص می‌کند.

Name: del subject در پرونده ثبت نام :

Return type: Boolean

این متد درس مشخص شده برای دانشجوی مورد نظر را از پرونده ثبت نام دانشجو حذف می‌نماید دارای دو پارامتر ورودی است یک پارامتر شاخص که درس و پارامتر دیگر شاخص شماره دانشجو است.

بعد در Detail بعد Right click بعد subject No را می نویسیم.

Type: long

Document بعد در

شماره درسی را که باید به صورت اضطراری حذف شود.

بعد پارامتر بعدی

Name: student No

Type: long

شماره دانشجویی که می خواهد حذف اضطراری نماید.

اما همانگونه که در بالا مشاهده می کنید در واقع دیاگرام توالی یک نوع برنامه نویسی است برنامه ای

که از دیاگرام توالی فوق قابل استنتاج است بصورت زیر است.

در word به صورت زیر می نویسیم.

() DeDelReg کنترل کننده

: () show فرم حذف و اضطراری

ورود شماره کارمند و شماره دانشجویی و کلاس

(شماره کارمند) allowed updateReg پرونده مجوز کارمندها

(شماره دانشجو و شماره درس) delsubject پرونده ثبت نام

در ادامه دیاگرام توالی به شرح زیر ارائه می گردد.

در دیاگرامهای توالی مثل سناریو همه چیز مثبت است و حالت‌های منفی و مشتق‌های دوم از نظر گرفته نمی‌شود به طور خلاصه چهار نوع کلاس در بالا ظاهر شد که برای هر کدام شمایل (icon) متفاوت در نظر گرفته شده.

نمایانگر فرم‌های ورودی و خروجی یا

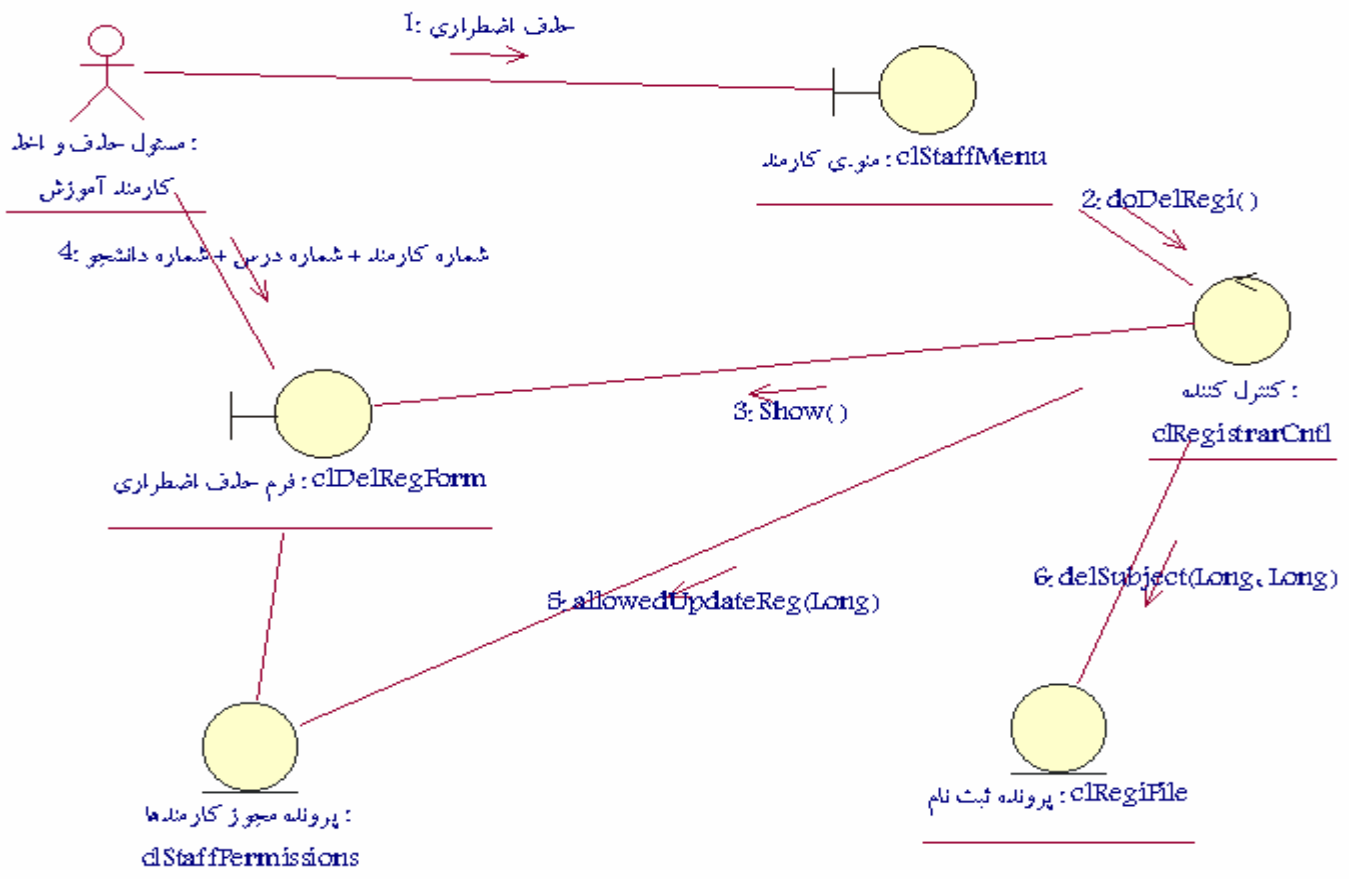
به طور کلی کلاسها از نوع boundary

نماینده کلاسهای کنترلی است مهم‌ترین نوع کلاسها، کلاسهای Entity هستند.

چرا که این نوع کلاسها بانک اطلاعاتی را کنترل می‌کند. کلاسهای Entity در واقع اطلاعاتشان در

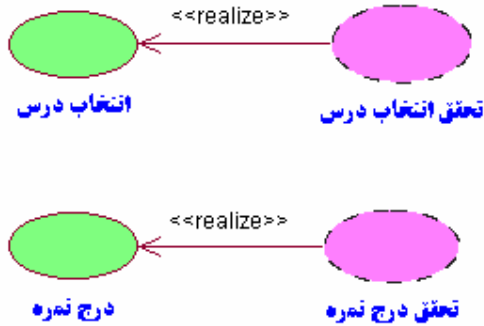
یک table و یا چند table ذخیره می‌شود اکنون با فشردن دکمه F5 دیاگرام توالی مورد استفاده

قرار گرفته و یک دیاگرام همکاری برای مورد استفاده مربوطه ایجاد می‌شود.



تحقق عملیات استاد

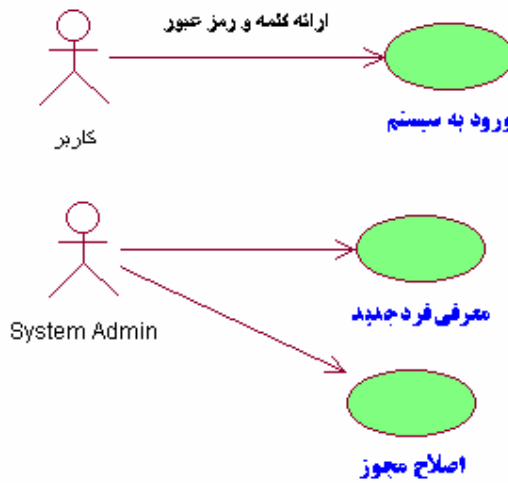
در تحقق عملیات استاد کلیک می‌کنیم.

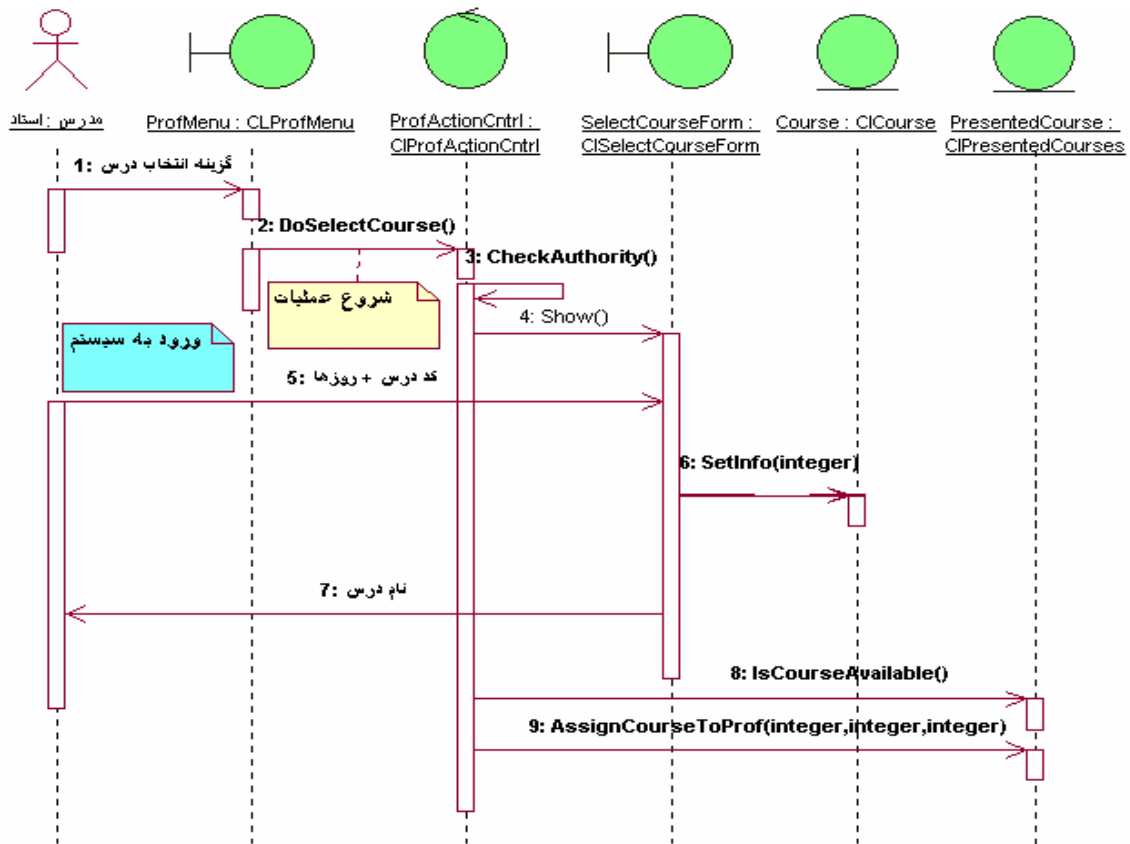


در تحقق انتخاب درس بعد Diagram را انتخاب بعد user sequence diagram را insert می‌کنیم.

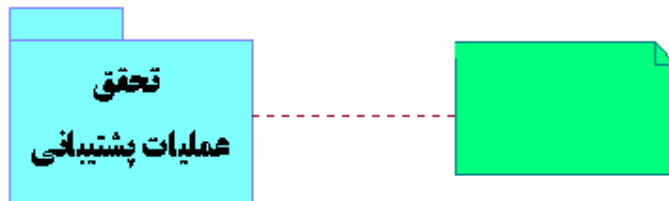
در عملیات پشتیبانی یک use case بنام ورود به سیستم و یک use case دیگر بنام معرفی فرد جدید قرار می‌دهیم.

عملیات پشتیبانی

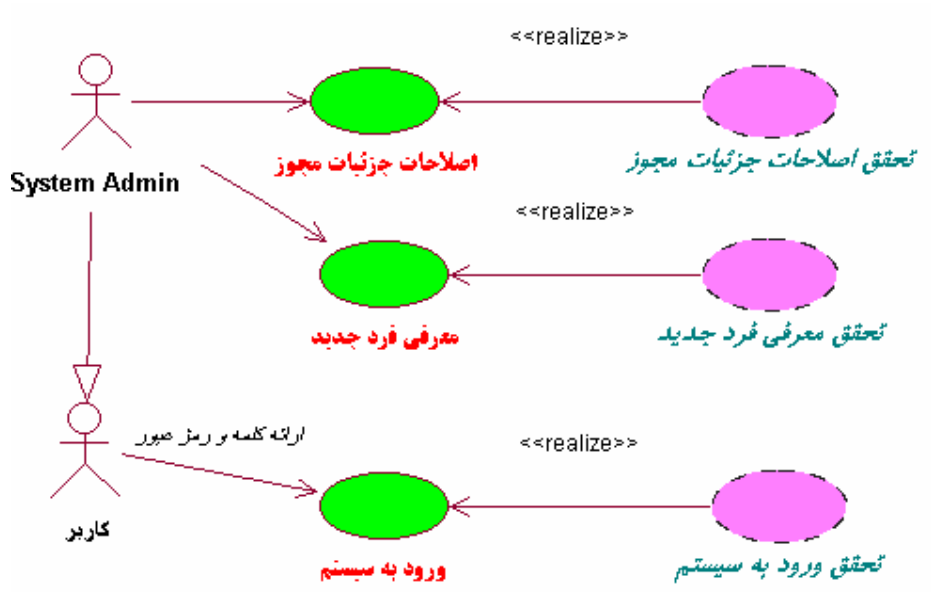




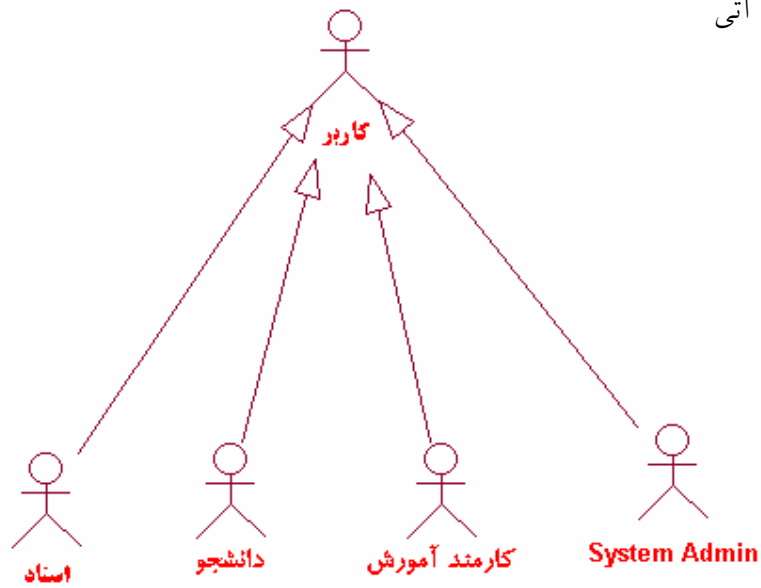
بعد در logical view به مدل آنالیز (تحلیل) می‌رویم بعد تحقق موردهای استفاده بعد تحقق عملیات ثبت نام بعد یک Package بنام تحقق عملیات پشتیبانی اضافه می‌کنیم.



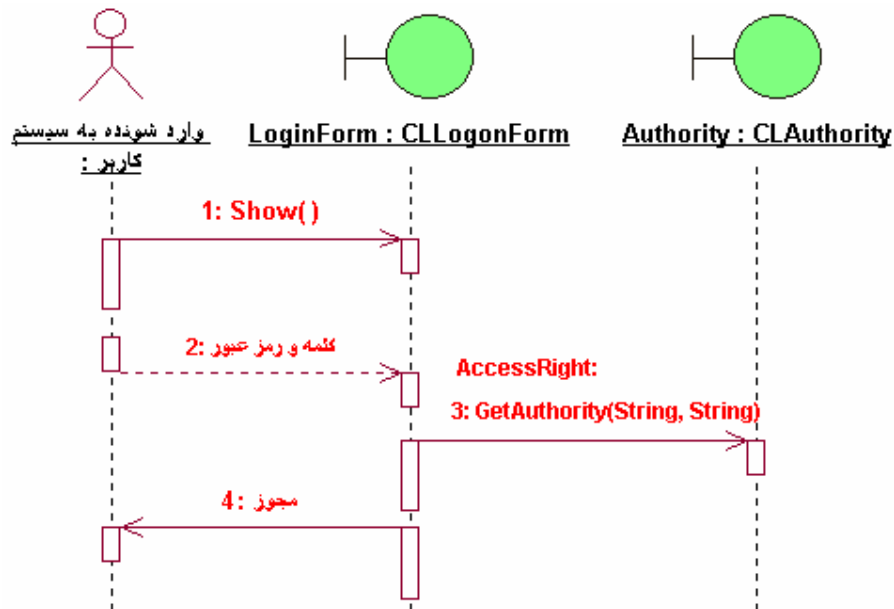
در تحقق عملیات پشتیبانی کلیک می‌کنیم.



بازیگرهای سیستم آتی



بر روی ورود به سیستم کلیک بعد sequence diagram می‌کنیم.



اکنون ببینیم این دیاگرام توالی که برای کنترل مجوزها و در واقع عمل login انجام می‌گردد چگونه در سایر قسمت‌ها استفاده می‌شود برای این منظور به صورت زیر عمل می‌شود.

اکنون با قراردادن یک mode به صورت زیر و درگ کردن sequence Diagram برای ورود به سیستم از داخل Browser به داخل Node این امکان را ایجاد می‌کنیم که بتوان ورود به مجوز را از داخل دیاگرام توالی برای use case انتخاب درس نیز به انجام رسانید.

همانگونه که در مثال فوق مشاهده نمودید عملیات پشتیبانی شامل دادن مجوزها است که در مثال فوق ورود به سیستم در قالب یک دیاگرام توالی به صورت مجزا مشخص گردید و از طریق قرار

دادن یک Node , Drag کردن دیاگرام توالی مربوط به عملیات منطقی یا دیاگرام توالی برای انتخاب درس مورد دسترس قرار گرفت.

اکنون دیاگرام توالی برای عملیات انتخاب درس را به صورت زیر تکمیل می‌نمایم.

ابتدا روی تحقق عملیات استاد کلیک کرده سپس روی تحقق انتخاب درس کلیک کرده.

همانگونه که در مثال فوق توجه می‌کنید می‌توان مقادیر بازگشتی از متدها را مشخص نمود که در

داخل کدام متغیرها نگهداری می‌شود. در مثال فوق برای نمونه `course available` یک مقدار

Boolean برمی‌گرداند که در متغیر `available` نگهداری می‌شود. به شرط اینکه `course` قابل

دسترسی یا `available` باشد آنگاه متد `assign course to prof` فراخوانی می‌شود. شرط را

توسط [] مشخص می‌کنند. در اینجا `available` در واقع مشخص می‌کند اگر `available` باشد

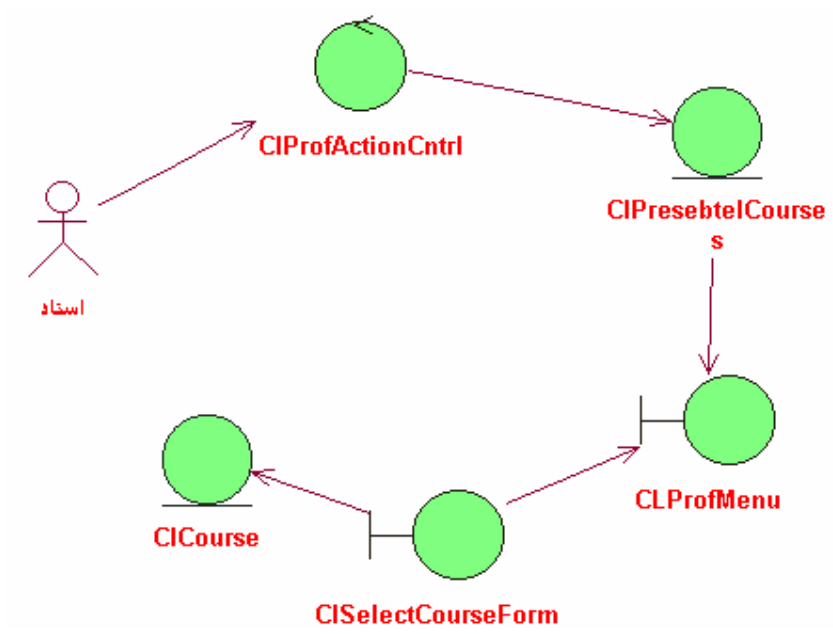
آنگاه `assign course` اجرا شود. حالا با فشردن دکمه F5 می‌توانیم برای دیاگرام توالی فوق

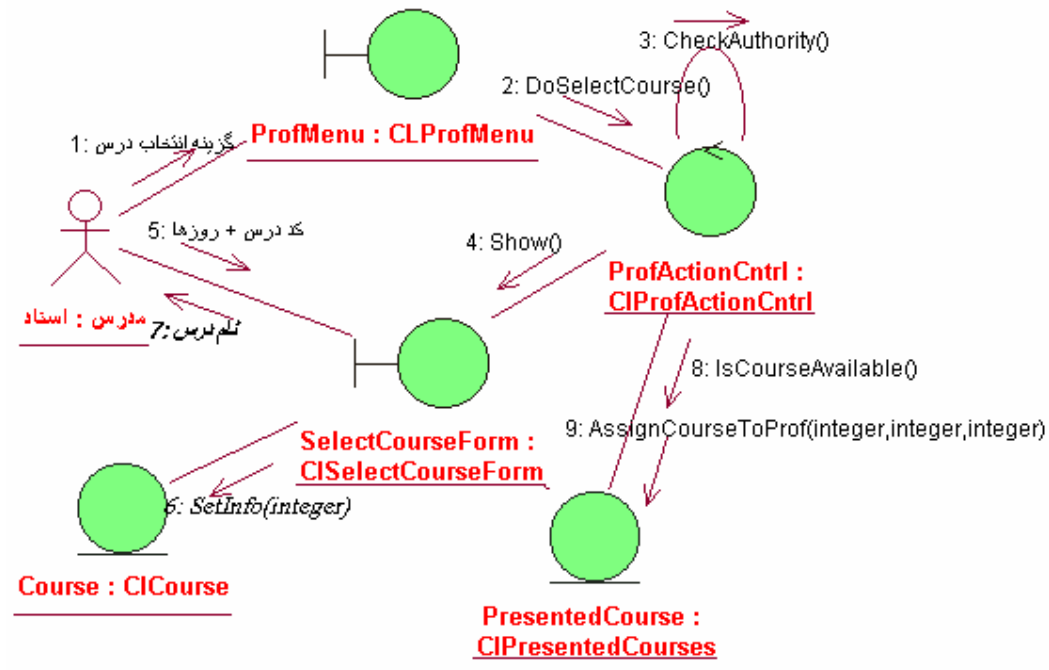
دیاگرام همکاری را ایجاد کرد. دیاگرام همکاری k شان می‌دهد که چه کلاس‌هایی با یکدیگر ارتباط

دارند. از وی دیاگرام همکاری می‌توانیم مدل ارتباطی کلاس‌های برای عملیات انتخاب درس

مشخص نماییم.

در انتخاب درس `class Diagram` را `insert` می‌کنیم.



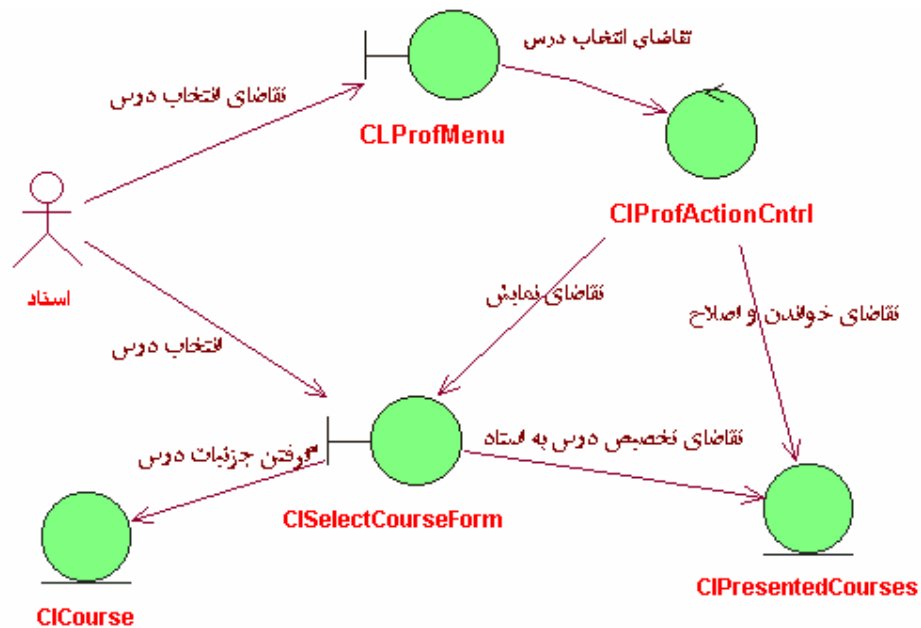


مدل ارتباطی کلاسها

همانگونه که قبلاً نیز توضیح داده شد رابطه بین دو کلاس را رابطه‌ی اجتماعی یا *asosiation* و رابطه‌ی بین دو *object* را *link* یا پیوند می‌نامند.

نوع خاصی از رابطه بین کلاسها را رابطه‌ی تجمع یا *Agregation* می‌نامند در واقع این رابطه بیانگر *is a part of* (هست بخشی از) می‌باشد. برای مثال ردیف درس اخذ شده بخشی از تعرفه‌ی دانشجو می‌باشد در اینجا ردیف درس، درس اخذ شده و تعرفه‌ی دانشجو دو کلاس می‌باشند و رابطه‌ی بین آنها رابطه‌ی تجمع است.

نوع دیگر رابطه، رابطه وراثت یا inheritance است رابطه وراثت یکنوع رابطه اجتماع است که با گذاره (یا نوعی است از) مشخص می‌شود. برای مثال Bolbol is a kind of Parandeh در اینجا رابطه وراثت بین دو کلاس بلبل و پرنده مشخص شده است. برای توضیحات بیشتر به کتاب و بخشهای قبلی جزوه رجوع شود.

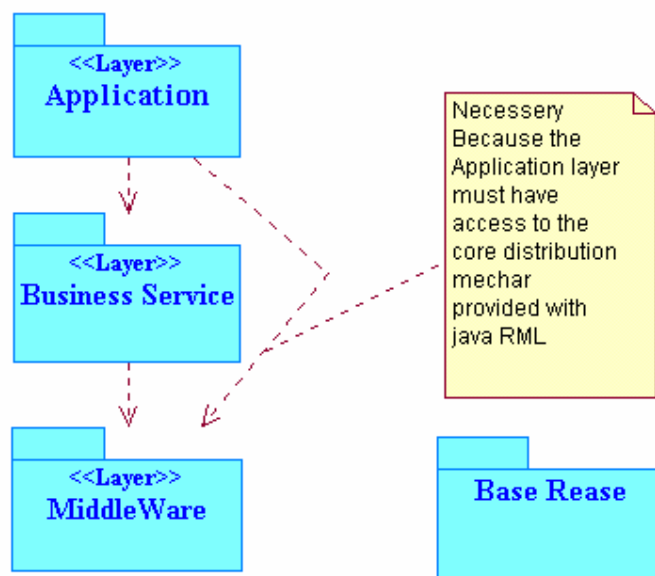


پس از تعیین مدل ارتباطی کلاسها معمولاً جهت گویا شدن مدل روابط وراثت را و همچنین روابط تجمع را مشخص می‌کنند از روی مدل ارتباطی کلاسها برای object ها(کلاسها) از روی Entity مدل بانک اطلاعاتی که در درس پایگاه داده خوانده‌اید ایجاد می‌شود البته در مثال از طریق امکاناتی بنام data Modeler می‌توان مدل بانک اطلاعاتی را از روی مدل ارتباطی کلاسها استخراج نمود. مدل ارتباطی کلاسها برای زیر سیستمها و یا کل سیستم تحت package مدل مفهومی قرار می‌گیرد.

برای نمونه در مثال فوق برای کل عملیات استاد یک مدل مفهومی با یک package بنام مدل مفهومی برای عملیات استاد و یک package بنام مدل مفهومی برای عملیات دانشجو و یک package برای عملیات واحد آموزش ایجاد می شود و داخل این package مدل ارتباطی کلاسها برای هر کدام از use case ها در کنار همدیگر قرار می گیرد بدین ترتیب دیگر کلاسهای تکراری نخواهیم داشت و کلاسهای مشترک مشخص می شود. حالا می توانیم به سراغ مرحله طراحی برویم.

مرحله طراحی:

در مرحله طراحی ابتدا معماری را مشخص می کنیم معمولاً معماری لایه ای ترجیح داده می شود. جهت تعیین مدل طراحی تحت logical view , package ای بنام Design model قرار می گیرد. البته package دیگری بنام Process در اینجا ایجاد می شود که این package در واقع برای مدل سازی فرآیندها یا Process حاوی و همروند مورد استفاده قرار می گیرد با click بر روی Design model بلافاصله چهار package به صورت زیر قرار می دهیم.



نکته اینجاست که چگونه برای یک stereo type, package مشخص می‌شود. برای این منظور می‌بایست بر روی Right click, package نمود. روی open specification می‌رویم. Stereo type آن را مشخص می‌کنیم. Stereo type می‌تواند به چهار صورت layer, Domain stereo type package, subsystem, organization unit stereo type global, package layer را انتخاب می‌کنیم. در قسمت Detail در صورتیکه کلاسهای داخل global, package باشند global.tag را علامت می‌زنیم. در همین قسمت می‌توانیم class Diagram برای packageها بکشیم. توجه کنید که ما class Diagram را به صورت جداگانه برای هر use case در Analyses mode تعیین کردیم حالا در اینجا پس از تعیین معماری نرم‌افزاری کلاسها را به هر Package تخصیص می‌دهیم. در اینجا معماری انتخابی معماری سه لایه است. در معماری لایه‌ای که ممکن است به هر تعدادی لایه داشته باشد لایه فوقانی بر مبنای ، لایه زیرین خود بنا می‌شود. لایه فوقانی را در طرح سه لایه، لایه کاربردی یا Application layer می‌نامند. در Application layer معمولاً کلاسها از نوع boundry قرار می‌گیرند. در واقع Application layer نمایانگر interface یا رابط سیستم با محیط جانبی آن است. تحت این package یعنی Application layer چندین Package قرار می‌گیرد که هر package در واقع یک دسته‌بندی از کلاسها را در این لایه مشخص کند.

در قالب Business layer با لایه کاری کلاسهای گنجانده می‌شوند که اطلاعات را از فرمهای ورودی می‌خوانند و حاصل را بر روی فرمهای خروجی قرار می‌دهند در واقع Business serveis لایه‌ای است که کلاسها (متدها) داخلی آن از طریق Application layer مورد

دسترسی قرار می‌گیرند. توجه کنید که کلاسهای کنترلر نیز در **Application** قرار می‌گیرند و در واقع اینها است که **object**ها در **Business. Services** را مورد دسترسی قرار می‌دهند.

در واقع شعور عملیات در قالب کلاسهای کنترلر در **Application layer** قرار می‌گیرد و اجرای عملیات در **Business layer** فرمان دهنده در اینجا همان کلاسهای کنترلر هستند که در **Application layer** قرار می‌گیرند.

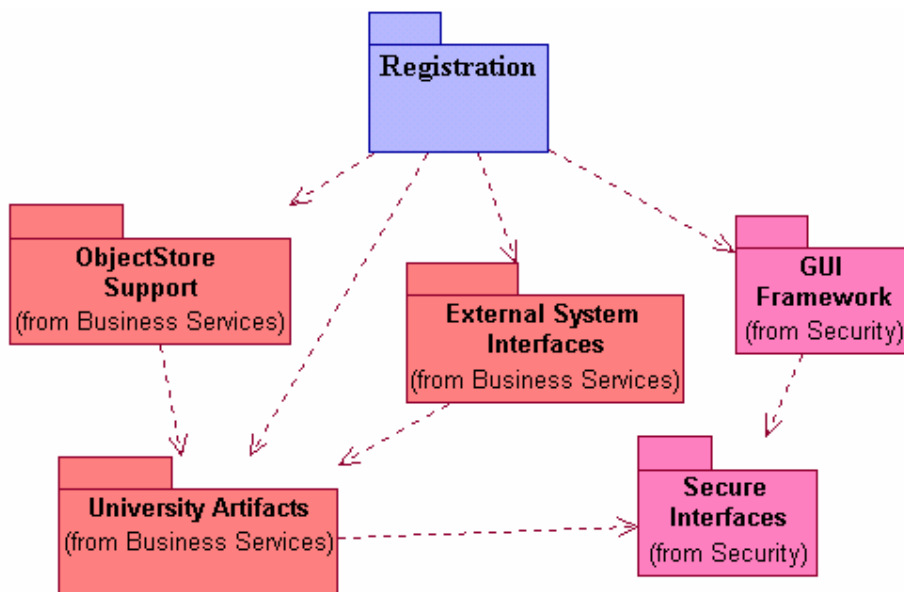
در لایه **Middel layer** مشخص می‌شود که چه نرم‌افزارهای آماده‌ای و چه بسته‌های نرم‌افزاری در قالب میان نرم‌افزاری مورد استفاده قرار گرفته‌اند.

برای نمونه فرض کنید می‌خواهیم کلاسهای مربوط به عملیات لیست حقوق کارمندان را لایه‌بندی کنیم یک کلاس از نوع **menu** وجود دارد که گزینه چاپ یا نمایش لیست حقوق کارمندان از طریق آن انتخاب می‌شود این کلاس در لایه **Application** قرار می‌گیرد.

یک کلاس از نوع کنترلر داریم که بلافاصله با انتخاب گزینه نمایش لیست حقوق کارمندان بلافاصله این کلاس متد تهیه گزارش لیستش فراخوانی می‌شود. این کلاس در لایه **Application** قرار می‌گیرد. یک کلاس از نوع **Entity** داریم که حاوی جزئیات عملکرد کارمندان است. یعنی مشخص می‌کند که هر شماره کارمندی چند ساعت کار، چند ساعت اضافه‌کاری، چند ساعت تاخیر یا غیبت داشته، این کلاس در لایه **Business** قرار می‌گیرد یک کلاس هم داریم که برای هر کارمند پایه حقوقی وی را مشخص می‌کنید. این کلاس شماره کارمندی است. البته کلاسها کلید ندارند. هویت **Identity** دارند که یک فیلد از یک کلاس را تشکیل می‌دهند. برای کلاسها از نوع **Entity** می‌بایست **table** ایجاد کرد تا اطلاعات مربوطه به رکوردها در آن **table** نگهداری شود. اما برای

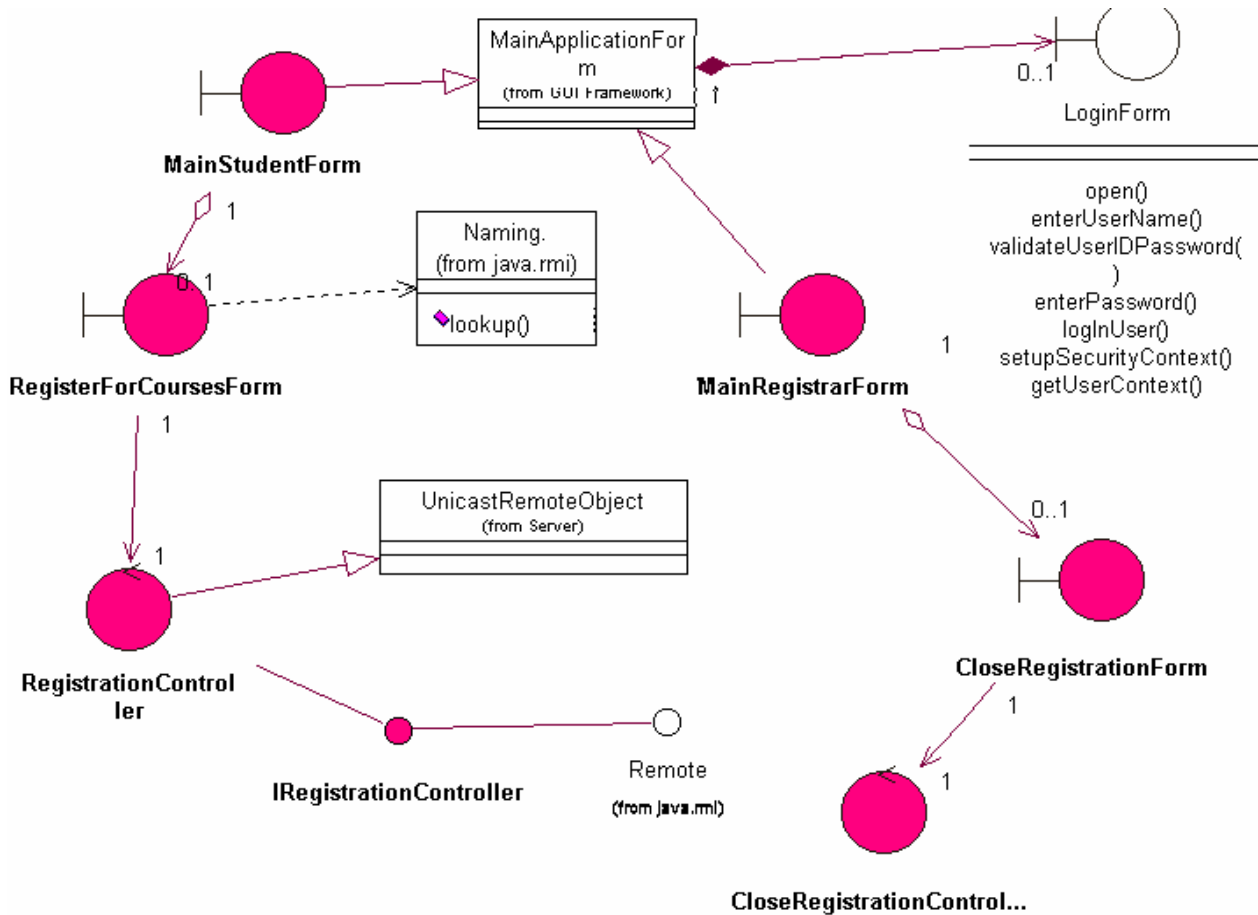
اینکه اطلاعات ذخیره و بازیابی شود ممکن است از **Sqlserver** یا اوراکل و یا هر نوع بسته بانک اطلاعاتی استفاده شود نوع بسته در قالب یک کلاس در لایه **Middle ware** مشخص می شود اگر فراخوانی ها از راه دور از طریق **corba** باشد یا **Java Rmi** یا بسته نرم افزاری **voijer** باشد می بایست همگی در لایه **Middle ware** مشخص شود.

اما در اینجا یک **package** بنام **global** می گذارند **package global** شامل کلیه کلاسهای است که بطور **global** از طریق سایر **package** مورد استفاده قرار می گیرد. اکنون با فشردن بر روی **Application layer** دیاگرام زیر ظاهر می شود.



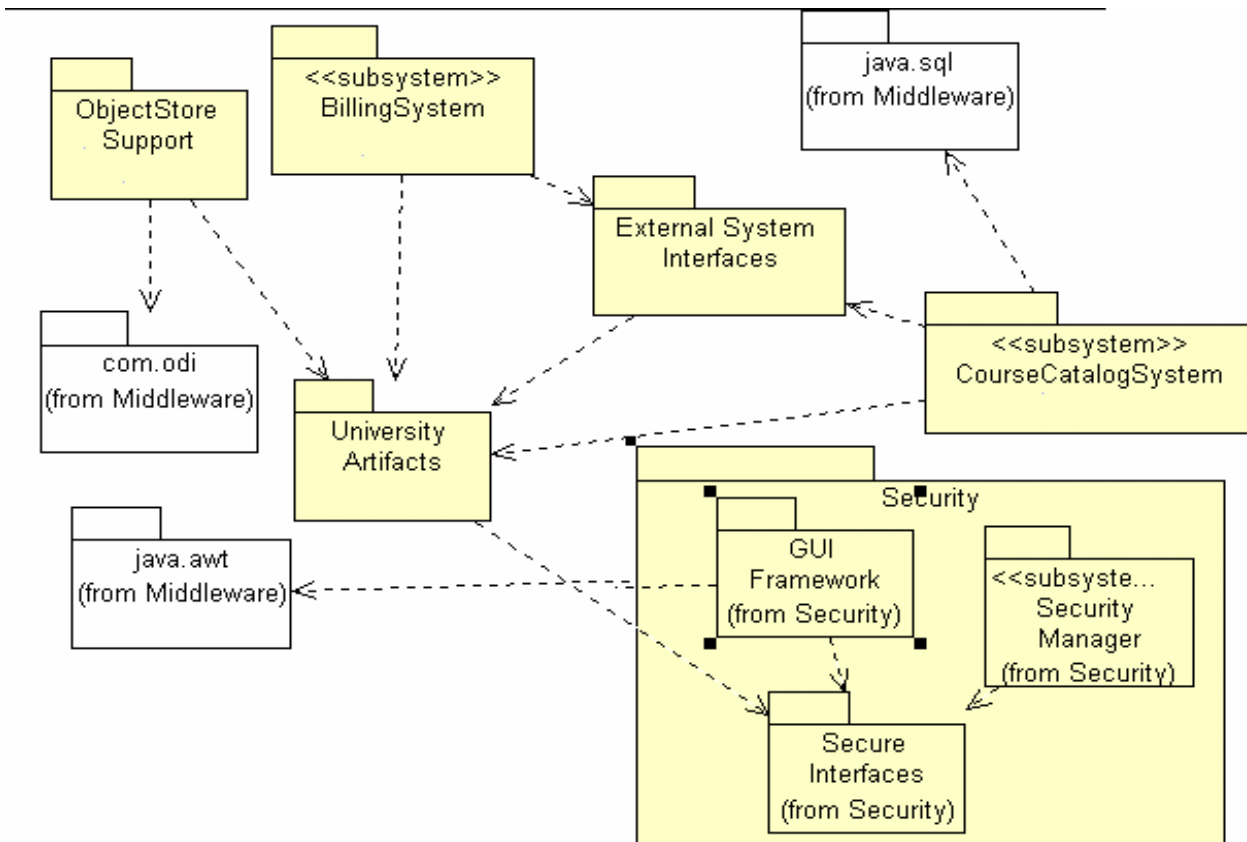
همانگونه که در مثال فوق مشاهده می‌کنید در واقع بسته Registration در لایه Application قرار دارد بسته‌های دیگر از لایه Business و از طریق Browser در اینجا درگ شده‌اند تا اینکه نشان داده شود لایه Application در ارتباط با Registration از کلاسهای کدام بسته‌ها در لایه Business استفاده می‌کند.

بنابراین به ترتیب مشخص شده که چگونه لایه Application بر روی لایه Business مستقر شده است. حالا در زیر لایه Registration مدل ارتباطی کلاسها قرار می‌گیرد.



اگر به مثال فوق توجه نمایید مشاهده می‌کنید که دو کلاس `Registration controller` و `Remote` (فراخوانی از راه دور) دو کلاس از نوع `interface` می‌باشند به عبارت دیگر `stereotype` برای این کلاسها از نوع `interface` تعیین شده است. کلاسهایی که به رنگ سفید مشخص شده‌اند از لایه `middle ware` برداشته شده‌اند. به عبارت دیگر وارد `Browser` شده‌ایم و در جایی که لازم بوده کلاسهای مورد استفاده شده را از لایه‌های `middle ware` به این جا `Drag` یا کشیده ایم تا نشان دهیم که کلاسها در لایه `Application` به کدام کلاسها در لایه `middle ware` دسترسی نمایند.

در واقع این کلاسها، کلاسهای جاوا هستند که برای برنامه‌نویسی مورد استفاده واقع خواهند شد. با فشردن بر روی لایه `Business services` شکل زیر بر روی صفحه که شاخص لایه `Business` است مشخص می‌گردد. در این شکل همچنین مشخص شده که در لایه `business` به کدام `package` ها از لایه تحتانی آن یعنی لایه `middle ware` دسترسی وجود دارد. به شکل زیر توجه نمایید.



در طراحی سه لایه کلاسها را دسته بندی می نماییم کلاسها از نوع controlly, boundry,

interface در لایه Application قرار می گیرند.

کلاسها از نوع Entity و غیره (غیر از boundry, interface در لایه Business servecy قرار

می گیرند.

کلاسها از نوع بسته های آماده نرم افزاری در لایه middle ware قرار می گیرند. برای نمونه در

ارتباط با سیستم آموزشی تحت مدل Design model و تحت Application layer سه

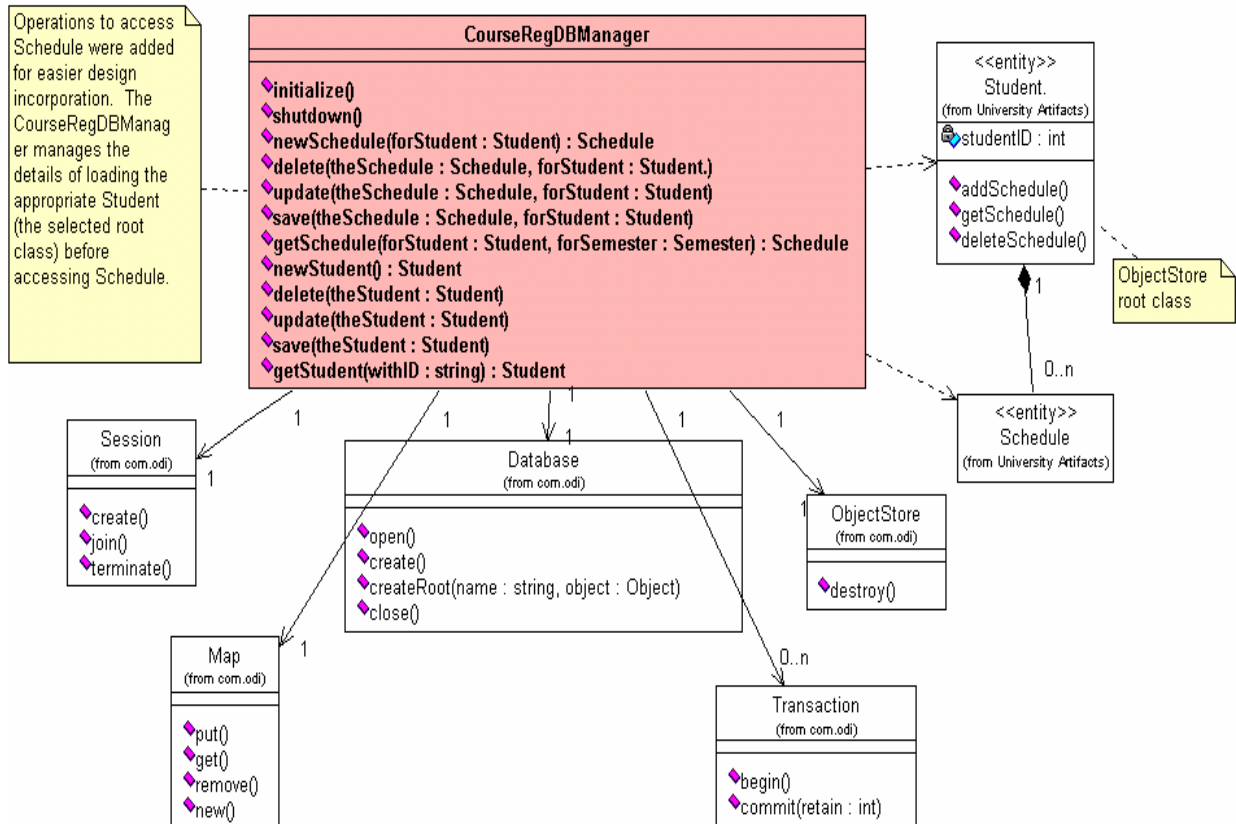
Package قرار می‌گیرد. یک Package شامل لایه کاربردی برای عملیات استاد package دیگر دارای نام لایه کاربردی برای عملیات دانشجو و بالاخره Package سوم دارای نام لایه کاربردی برای عملیات آموزشی است.

به همین ترتیب همین تقسیم‌بندی را در صورت نیاز در لایه business services ممکن است انجام دهیم یا ممکن است ندهیم. چرا که لایه business در واقع کلاس‌هایی است که به کل لایه Application سرویس می‌دهد و ممکن است این کلاسها برای نمونه بین عملیات استاد و دانشجو مشترک باشد لذا تقسیم‌بندی در این لابه ممکن است برای مثال : شامل کلاسهای عملیات مالی، کلاسهای عملیات برنامه‌ریزی و کلاسهای عملیات آموزشی باشد اما نکته دیگر نشان دادن چگونگی قرار گرفتن لایه‌ها بر روی یکدیگر است که در اینجا برای مثال در لایه Application می‌خواستیم نشان دهیم در عملیات Registration له کلاسهای object store support در لایه Business باید دسترسی شود. برای این منظور این package را (object store support) را از داخل Browser با کلیک کردن ماوس بر روی آن به داخل لایه Application، درگ نمودیم و سپس یک فلش ارتباط وابستگی که به صورت خط‌چین است بین این دو ترسیم کردیم تا نشان دهیم که برای انجام عملیات ثبت نام به کلاسهای این package در لایه business نیاز است.

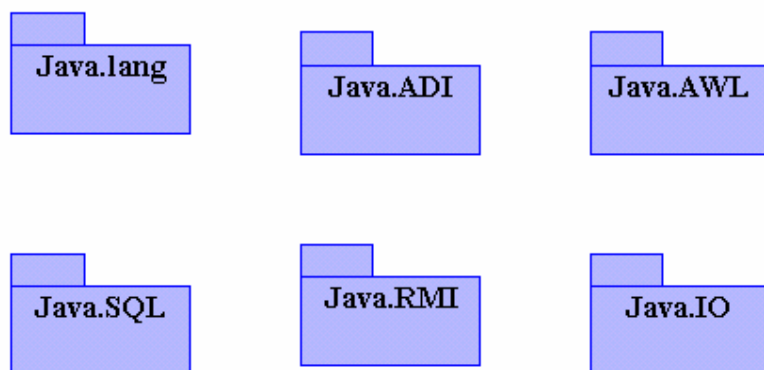
در لایه Business services معمولاً کلاسها از نوع Entity قرار می‌گیرند برای نمونه اگر به شکل زیر توجه کنید مشاهده می‌کنید که کلاس بنام course reg dB manager یا به عبارت دیگر کلاس مدیریت (Manager) بانک اطلاعاتی (DB) برای ثبت نام (Reg) دروس (Course) در

لایه Business تحت Package object store یا انباره object ها به صورت زیر قرار گرفته

است.



اگر به عملیات کلاس Course Reg DB Manager نگاه کنید مشاهده می‌کنید که برای عملیات با بانک اطلاعاتی و table از لایه middle ware که شامل Package ها و بسته‌های آماده نرم‌افزاری است استفاده می‌کند. برای مثال عمل initialize در واقع بانک اطلاعاتی را باز می‌کند و () shot down بانک اطلاعاتی را می‌بندد. برای انجام عملیات save, update, delete نیز به کلاسهای جاوا که در شکل مشخص شده ارجاع می‌شود. بنابراین در لایه business معمولاً کلاسها از جنس Entity قرار می‌گیرند و برای انجام عملیات خود به کلاسها در لایه زیرین یعنی لایه middle ware دسترسی پیدا می‌کنند. در لایه middle ware نرم‌افزارهای آماده و کلاسهای مربوط به آنها مشخص می‌شود اگر بر روی package لایه middle ware کلیک کنیم package های زیر ظاهر می‌شوند.



اینها در واقع **library** های جاوا هستند برای مثال **java.sql** خود دارای تعدادی کلاس جاوا است و شما اگر برای مثال زبان **VB** را استفاده کنید و بانک شما **oracle** است یک **package** بنام **oracle** و یک **package** بنام **library** های **vb** ایجاد می کنید.

همانگونه که در مثال فوق مشاهده می کنید در طرح سه لایه هر لایه فوقانی مبتنی بر لایه های زیرین بنامی شود برای مثال لایه **Application** مبتنی بر دو لایه **Business** و **middleware** است در حالی که لایه **Business** هیچ استفاده ای از لایه **Application** استفاده نمی کند و مستقل از آن ایجاد شده است بعد از اینکه مرحله طراحی خاتمه یافت وارد مرحله پیاده سازی می شویم در اینجا معماری برنامه های باید مشخص کنیم. معماری برنامه را **component** یا دیدگاه قطعات می نامند.

دیدگاه قطعات

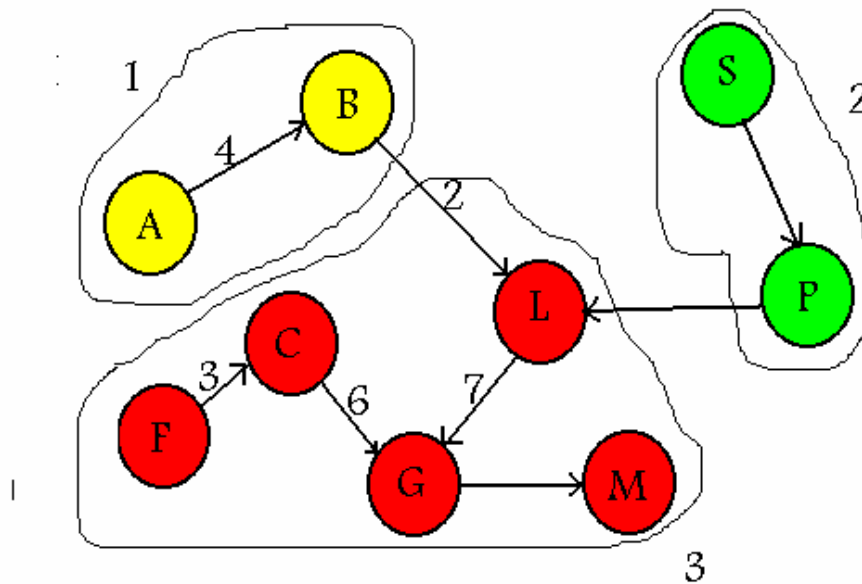
در برنامه سازی شیء گرا به جای لفظ ماجول از لفظ **component** استفاده می شود در واقع در طرح معماری برنامه ها اصول ماجولار مطرح است. در ارتباط با برنامه های شیء گرا **Class** **dependency Graph** در واقع بیانگر مدل ارتباطی کلاسهاست ارتباطات معمولاً یا بوسیله فراخوانی ها، ارسال **event** یا پیام، رابطه تجمع و بالاخره رابطه وراثت ایجاد می شود.

Class Dependency Graph گرافی است که گره های آن کلاسها، لبه های آن جهت دار و وزن دار می باشند وزن هر لبه میزان ارتباط یک کلاس با کلاس دیگر را مشخص می کند.

بعد از اینکه بر اساس میزان فراخوانی ها در ارتباطات مدل ارتباطی کلاسها تبدیل به یک مدل **Class** **dependency Graph** شد یعنی لبه های آن وزن دار شد کلاسها را بر اساس میزان ارتباطات دسته بندی می کنند و هر دسته شامل کلاسهایی است که ارتباطات آنها با یکدیگر بسیار زیاد است.

کلاسهایی که ارتباطاتشان با یکدیگر زیاد است در یک component قرار می‌گیرند یا در اصطلاح ساختار یافته در یک ماجول قرار می‌گیرند.

در اینجا فواصل coulision (چسبندگی) کلاسهای هر component یا ماجول یا decoupling یا عدم اتصال componentها با یکدیگر بسیار حائز اهمیت است.



حالا پس از مرحله طراحی کلیه کلاسها مشخص اند متدهای آنها و attribut دقیقاً مشخص شده بر اساس sequence Diagram ها برای متد کلاسها sodo code یا شبه دستورالعمل تعیین شده و حالا می‌دانید که هر کلاس تا چه حد با کلاسهای دیگر در ارتباط است وارد component view می‌شویم و تصمیم‌گیری می‌کنیم که بر اساس خوشه‌بندی کلاسها که در بالا نشان داده شد چند عدد خوشه یا چند عدد component لازم داریم در مثال فوق سه عدد component لازم بود مسأله

بعدی چگونگی تخصیص کلاسها به component ها است. در مثال فوق برای نمونه دو کلاس B,A می‌بایست به component شماره یک Assign یا تشخیص شوند.

بنابراین موضوع چگونگی تخصیص کلاسها به component یا قطعات است. برای نمونه به مثال سیستم کارت ساعت در کتاب در ص 524 ارجاع نمایید.

در اینجا برنامه سیستم کارت ساعت شامل سه برنامه مجزا است که از طریق خطوط شبکه، یکدیگر در ارتباطند. این سه برنامه یکی در واحد نگهداری، دیگری در واحد کارگزینی، آخری در واحدهای سازمانی نصب می‌شوند. اگر بر روی Gurd Module, package کلیک کنیم Component به صورت زیر برای برنامه واحد نگهداری مشخص شود، ص 525

در اینجا یک component ساده و دو component از نوع task یا وظیفه مشخص شده است. Taskها در واقع thread مشخص شده است. در واقع task ها شاخص قطعاتی هستند که به صورت همزمان به اجرا در می‌آیند. در اینجا دو task با اسامی Guard interface یا رابط نگهدار یا input card Data یا رابط دستگاه کارتخوان وجود دارد. این دو ماجول به صورت موازی و همزمان فعالند. برای تخصیص کلاسها بدانها ابتدا بر روی هر کدام click می‌شود یک صفحه باز می‌شود در گزینه General در این صفحه برای component زبان برنامه‌سازی مشخص می‌شود که در اینجا VB است. می‌تواند vc یا دلفی هم باشد.

حالا گزینه realizes را انتخاب دار این قسمت list کلیه کلاسها ظاهر می‌شود حالا ما کلاس مورد نظر خود را به assign, component می‌کنیم بر روی این منظور بر روی کلاس right click می‌کنیم و گزینه assign را انتخاب می‌کنیم کلاس قرمز رنگ می‌شود به علامت اینکه تخصیص داده

شده است. باید کلیه کلاسهای ما قرمز رنگ شوند برای مطالعه بیشتر به component به صفحه 471 کتاب مراجعه فرمائید.

اکنون بعد از اینکه component ها به کلاسها تخصیص شدند. می توان با استفاده از منوی Tools گزینه visual Basic بعد update code مبادرت به تولید کد vb برای قطعات نمود.

علاوه بر مدل قطعات یکی دیگر از دیدگاهها deployment view یا مدل نصب است. در واقع در این دیدگاه طرح شبکه ایستگاههای کاری، دستگاههای جانبی، دقیقاً مشخص می شود. برای نمونه در زیر سیستم نگهداری یا کارت ساعت Deployment view به شرح زیر است.

برای نمونه به مثال ارائه شده در صفحه 475 و همچنین در ص 527 کتاب ارجاع شود بنابراین به طور خلاصه مشاهده می کنیم که در دیدگاه logical view شامل آنالیز مدل Design model (در قالب طرح سه لایه) ، component view و بالاخره deployment view مطرح می گردد. معماری تنها لایه ای نیست نوع دیگر از معماری لوله و فیلتر است. (pipe and filter) در این نوع معماری سیستم از چند برنامه مجزا به نام فیلتر تشکیل شده که از طریق برنامه ها و توابع ارتباطی بنام pipe با یکدیگر ارتباط برقرار می کنند. معماری دیگر Black board است که در سیستمهای هوش مصنوعی و سیستمهای خبره مورد استفاده قرار می گیرد. معماری بر اساس قطعات امروز با یکی از رایجترین انواع معماری است اینکه بتوانیم سیستم را از قطعات اسمبل یا ایجاد نماییم نقطه ای است بسیار قابل توجه در طرح معماری، قطعات در اینجا معمولاً به صورت activim یا complan و قطعاتی مثل جاواییلز مطرح هستند.