

به نام خدا

سناریو دیاگرام (مدرسه)

درس آزمایشگاه مهندسی نرم افزار

استاد ساويز نائینی

دانشجویان

محمد غمگسار

محمد جواد علمدارپور

دانشکده شمسی پور

پاییز ۱۴۰۲

• شرح مسئله

UML به معنای زبان مدل سازی یکنواخت است که در حقیقت یک زبان مدل سازی گرافیکی استاندارد برای مستندسازی و تجسم سیستم ها است.

تعریف UML

Unified Modeling Language که معمولاً به اختصار UML نامیده می شود - یک زبان مدل سازی گرافیکی برای مستند سازی و تجسم سیستم های نرم افزاری است که انواع نمودارهای مختلف را برای تجزیه و تحلیل سازه ها و طراحی رفتار سیستم ها ارائه می دهد. در واقع می توان گفت که UML روشی برای به تصویر کشیدن یک برنامه ی نرم افزاری با استفاده از مجموعه ای از نمودارهاست.

ایده UML به Grady Booch ، James Rumbaugh و Ivar Jacobson برمی گردد که روش های شی گرا مختلفی را در زمینه های کاربردی تعریف شده در یک رویکرد تلفیقی توسعه دادند. رویکرد three amigos که از طرف شرکت راشنل رز معرفی شد، به سرعت تبدیل به یک شبه استاندارد شد (Object Management Group (OMG. این رویکرد را در پیش گرفت و نسخه ۱٫۱ UML را در سال ۱۹۹۷ منتشر کرد. نسخه فعلی ۲٫۵٫۱ UML است که توسط OMG در دسامبر ۲۰۱۷ منتشر شده است. همچنین نسخه ۲٫۴٫۱ توسط ISO (ISO/IEC ۱۹۵۰۵ استاندارد شده است.

انواع نمودارها در UML

UML در مجموع شامل ۱۴ نمودار است که به سه گروه یا نوع نمودار تقسیم می شوند:

- نمودارهای ساختاری یا structure diagrams
- نمودارهای رفتاری یا behavioural diagrams
- نمودارهای تعاملی یا interaction diagrams

نمودارهای ساختاری نمودارهایی هستند که یک جزء ثابت از یک سیستم را مدل می کنند، که در آن داده ها تغییر می کنند اما ساختار عناصر و روابط آنها با یکدیگر تغییر نمی کند. یک نمودار رفتاری جنبه های فردی یک سیستم، توالی فرآیندها و تغییرات آنها را در زمان اجرا به تصویر می کشد و نمودارهای تعاملی رفتار تعاملی یک سیستم و در نتیجه جریان اطلاعات در آن را تجسم می کنند.

تنوع نمودارها امکان استفاده عملی را در تمام مراحل توسعه فراهم می کند. نمودارهای UML را می توان به عنوان مثال برای مستند سازی نیازمندی ها، طراحی نرم افزار، مستندات فنی و پیاده سازی سیستم ها استفاده کرد.

انواع نمودارهای ساختاری:

- نمودار کلاس،
- نمودار شیء،
- نمودار ترکیب ساختار،
- نمودار اجزا،
- نمودار توزیع و استقرار
- نمودار پکیج

در نمودارهای کلاس، کلاس ها و روابط بین کلاس ها به صورت ارتباط، تجمع، ترکیب یا تعمیم مدل سازی می شوند. نمودار شی یک نمونه مشخص از نمودار کلاس را در یک نقطه زمانی تعریف شده توصیف می کند. نمودار ترکیب ساختار، ساختار داخلی یک کلاس را نشان می دهد و پیکربندی عناصری را که رفتار کلاس را تعیین می کنند، به تصویر می کشد. نمودار جزء ساختار و رابطه بین اجزای مختلف یک سیستم را نشان می دهد. نمودار توزیع، منابع فیزیکی یک سیستم را توصیف می کند، مثل نودها، کامپوننت ها) و اتصالات و نمودار پکیج برای نشان دادن گروهی از عناصر مدل - بسته ها - و وابستگی های بین پکیج های یک مدل استفاده می شود.

انواع نمودارهای رفتاری:

- نمودار مورد استفاده،
- نمودار فعالیت و
- نمودار حالت

نمودار مورد استفاده رفتار یک سیستم را از دیدگاه کاربر به تصویر می کشد و موارد استفاده را با روابط آنها با سایر موارد استفاده و سیستم های دیگر نشان می دهد. نمودارهای فعالیت دنباله ای از فعالیت ها، فرآیندها در سیستم ها یا فرآیندهای تجاری را نشان می دهند، و نمودارهای حالت - که گاهی اوقات به عنوان نمودار وضعیت هم شناخته می شوند - دنباله ای از حالت های مجاز را به تصویر می کشد که یک شی می تواند در چرخه زندگی خود فرض کند.

انواع نمودارهای تعاملی:

- نمودار توالی،
- نمودار ارتباطی،
- نمودار زمان بندی و
- نمودار نمای کلی تعاملات

نمودار توالی چگونگی تبادل اطلاعات توسط اشیا و نمونه های آنها را با ترتیب خاصی توصیف می کند. نمودار ارتباطی – که در نسخه های قبلی UML نمودار همکاری نامیده می شود – پیام های انتخابی را که برای توالی ارتباط بین اشیاء استفاده می شود، هایلایت می کند. نمودار زمان بندی اشیایی را نشان می دهد که در یک دوره زمانی مشخص با یکدیگر تعامل دارند و نمودار نمای کلی تعاملات، این نمودار، ترکیبی از نمودارهای فعالیت و توالی است. این نمودار یک توالی از عملیات ها را مدل می کند و می تواند تعاملات پیچیده تر را به صورت رخدادهایی قابل کنترل دریاورد. علاوه بر این، Unified Modeling Language یک نمودار فرامدلینگ هم دارد:

- نمودار پروفایل

این نمودار برای تجسم کلیشه های تعریف شده توسط کاربر، مقادیر دارایی و محدودیت ها در سطح متامدل استفاده می شود.

مزایای Unified Modeling Language

استفاده از Unified Modeling Language مزایایی دارد، اما ممکن است در عمل دارای معایبی نیز باشد. مزایای آن عبارتند از:

- چند منظوره و انعطاف پذیر است. برای مثال، برای تجسم یا مستندسازی سیستم ها یا نرم افزارها مناسب است و می توان آن را مستقل از بخش ها یا صنایع و در تمام مراحل توسعه مورد استفاده قرار داد.
- هر نمودار عناصر مناسبی را ارائه می دهد که با آن می توان اطلاعات و زمینه های خاصی را ارائه کرد.
- استفاده از Unified Modeling Language در حالت ایده آل منجر به درک بهتر سیستم مورد توسعه یا مستند سازی می شود و می تواند به کاهش هزینه های احتمالی، به عنوان مثال در مرحله اجرا یا نگهداری کمک کند.
- همچنین در حالت ایده آل، همکاری همه کاربران مانند معماران نرم افزار و توسعه دهندگان نرم افزار، تحلیلگران کسب و کار، مدیران پروژه یا مدیران محصول را نیز ارتقا می دهد.
- Unified Modeling Language علاوه بر اینکه یک نمای گرافیکی از یک سیستم را ارائه می دهد، شامل قالبی برای تبادل مدل ها یا نمودارها بین ابزارهای مختلف نیز هست و این انعطاف پذیری در برنامه را افزایش می دهد.
- و در آخر، ابزارهای مختلف از تبدیل مدل های UML دامنه گرا به مدل های فنی با تولید کد بعدی پشتیبانی می کنند.

معایب Unified Modeling Language

- مستندات ۲,۵ UML با ۷۹۴ صفحه بسیار گسترده است. اگرچه نسخه های قبلی گسترده تر بودند، این نسخه صفحات زیاد با اطلاعات زیادی را شامل می شود. نظرات در مورد سهولت خواندن و درک مستندات متفاوت هستند.
- هیچ فرمولی برای تعیین تعداد معقولی از نمودارها برای توصیف و تجسم روابط متقابل یک سیستم یا توسعه نرم افزار وجود ندارد. همچنین تعداد ایده آلی از عناصر نمودار در هر نمودار وجود ندارد. با افزایش اندازه نمودار، وضوح و خوانایی اغلب

تنزل پیدا می کند. درک روابط متقابل برای شخصی که نمودار را ایجاد می کند، احتمالاً آسان است. اما اگر برای اولین بار یک نمودار گسترده را ببینید، ممکن است نتوانید تمام اطلاعات آن را درک کنید.

- اگرچه درک روابط متقابل را می توان در زمینه های متعددی استفاده کرد، اما عمدتاً برای مستندسازی و به ندرت برای توسعه نرم افزار یا سیستم استفاده می شود.

تعداد زیادی ارائه دهنده نرم افزار UML وجود دارد و این می تواند یک مزیت یا یک نقطه ضعف باشد. از یک طرف، این تنوع انتخاب نرم افزار متناسب با شرایط و شرکت را برای شما دشوار می کند. از سوی دیگر استفاده از نرم افزار را برای انواع کاربران کم تجربه، حرفه ای یا سطح متوسط تسهیل می کند.

در آخر باید گفت این واقعیت که Unified Modelling Language شامل راهنمای توسعه سیستم ها یا نرم افزار نمی شود و یا حتی یک زبان برنامه نویسی است، نه مزیت است و نه نقص.

این اینفوگرافیک را ذخیره داشته باشید



ابزارهای UML

طیف و سببی از ابزارهای UML وجود دارد که به طور قابل توجهی با یکدیگر تفاوت دارند. برنامه های رایگانی برای رسم نمودارها و راه حل های حرفه ای وجود دارند که بر اساس نمودارها کد تولید می کنند. برخی از ابزارها از انواع نمودارهای انتخاب شده پشتیبانی می کنند، در حالیکه برخی دیگر از همه نمودارها پشتیبانی کامل به عمل می آورند.

برخی هم از چند نماد از مجموعه نمودارهای فردی استفاده می کنند و بعضی دیگر قادر به استفاده مجدد از عناصر تعریف شده در نمودارهای مختلف هستند. برخی از مهندسی معکوس یا رفت و برگشت پشتیبانی می کنند، برخی دیگر از XMI پشتیبانی می کنند. برخی از ابزارها برنامه های کاربردی سرویس گیرنده-سرور هستند، برخی دیگر به صورت آنلاین در مرورگر اجرا می شوند و داده ها را در یک ابر ذخیره می کنند. برخی از تبادل اطلاعات بین ابزارهای UML پشتیبانی می کنند و بعضی هم گزارش های قابل تنظیم ارائه می دهند. بنابراین اگر می خواهید از ابزار UML استفاده کنید، باید بر حسب نیاز خود تصمیم گیری کنید.

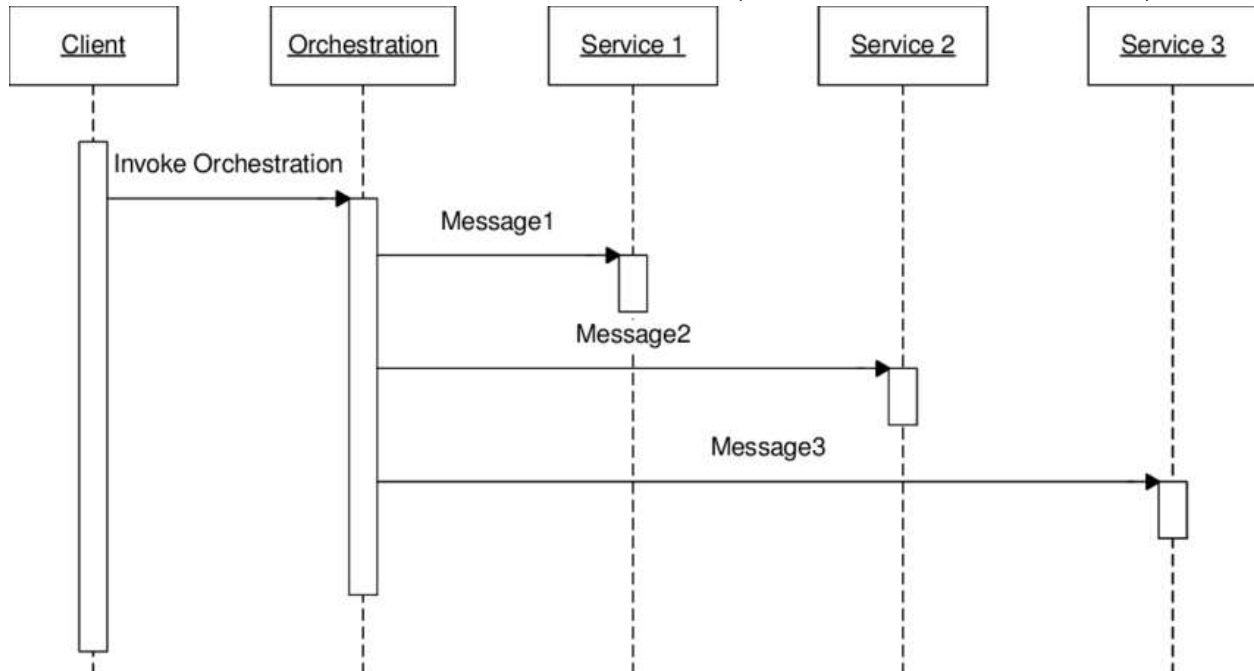
در ادامه می توانید لیستی از ابزارهای UML را مشاهده کنید:

- [Astah](#) •
- [BOUML](#) •
- [Cacoo](#) •
- [CaseComplete](#) •
- [Chart Mage](#) •
- [Concept Draw](#) •
- [creately](#) •
- [DotUML](#) •
- [Diagramo](#) •
- [Eclipse Papyrus](#) •
- [EdrawMax](#) •
- [Enterprise Architect](#) •
- [GenMyModel](#) •
- [GitMind](#) •
- [gliffy](#) •
- [IBM Engineering Systems Design Rhapsody](#) •
- [IBM Rational Software Architect Designer](#) •
- [JetUML](#) •
- [Lucidchart](#) •
- [MagicDraw](#) •
- [modelio](#) •
- [MS Visio](#) •
- [Open ModelSphere](#) •
- [PlantUML](#) •
- [Pópulo](#) •
- [QM](#) •
- [SinelaboreRT](#) •
- [Software Ideas Modeler](#) •

- smartdraw
- StarUML
- System Architect
- TextUML
- Together
- txtUML
- Umbrello
- UML Designer
- UML Graph
- UMLet
- UModel
- Umple
- USE
- Violet UML Editor
- Visual Paradigm
- WhiteStarUML
- xtUML
- yEd Graph Editor
- yuml
- ZenUML

دیاگرام ها

۱. UML Diagram (Sequence Diagram)



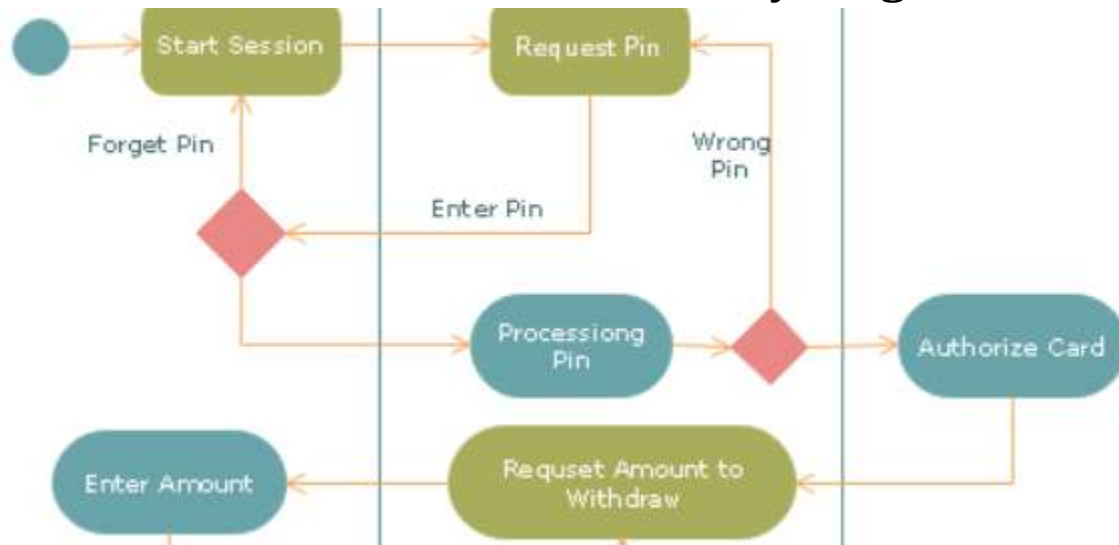
نمودار توالی یا Sequence Diagram و میخوام اول به نقل از ویکی پدیا شرح بدم:

نمودار توالی (به انگلیسی: Sequence Diagram) یکی از نمودارهای زبان مدل سازی یکپارچه است که روندی در یک پروژه را مرحله به مرحله نشان می دهد. این نمودار زمانی مفید است که بخواهید روند منطقی یک سناریو را به نمایش بگذارید.

حالا روند یک برنامه چیه؟

از زمانی که برنامه شروع میشه تا پايانش روند یک برنامه اس حالا ما میخوام روند یک فیچرمون رو تو سکونس دیاگرام بیاریم اونم میشه همه ی روند ها رو میتونیم بیاریم ببینید چقدر خوبه!

۲. Activity Diagram UML



اکتیویتی دیاگرام Activity Diagram چیست؟

همانطور که در توضیح یوزکیس دیاگرام Usecase Diagram هست ، یوزکیس ها به شما میگویند که نرم افزار قرار است چه کاری قرار انجام دهد ولی اکتیویتی دیاگرام Activity Diagram به شما خواهند گفت که این کارهای باید چگونه انجام شوند تا نیازمندی های نرم افزار برآورده شود. در نمودار اکتیویتی قرار است بصورت سطح بالا مشخص شود که چه فعالیتهایی باید انجام شود تا هر کدام از این کارها و usecase ها انجام شوند. در این دیاگرام مراحل و step ها و زنجیره انجام کارهای مشخص خواهند شد. میتوانیم اکتیویتی دیاگرام را به عنوان مدل سازی بیزینس و لاجیک Business Logic Modeling دانست. بیزینس مدل زنجیره ای فعالیت ها و Task ها است که با هم منجر به یک کاری خواهند شد. بعضی از نرم افزارهای مدل سازی پراسس های تجاری

Business Process Management Tools یا به اختصار BPM ها ابزاری را برای شما فراهم میسازند که فرآیند تجاری هر بخشی از کسب و کار خود را مدلسازی و ترسیم کنید.

اکتیویتی دیاگرام تنها نموداری است که در معماری نرم افزار در بخش Process Architecture ترسیم خواهد شد. قواعد و Syntax هایی که در اکتیویتی دیاگرام استفاده می شوند بسیار ساده و مانند فلوچارت هستند و notation های خیلی قابل فهمی برای عمومی دارند و به همین دلیل این نمودار بهترین دیاگرام برای ارائه به ذینفعان سطح بالای نرم افزار مانند هست مدیره و Member ها هستند. در اکتیویتی دیاگرام شما قرار است مراحل که برای انجام هر یوزکیس لازم است را مدلسازی کنید.

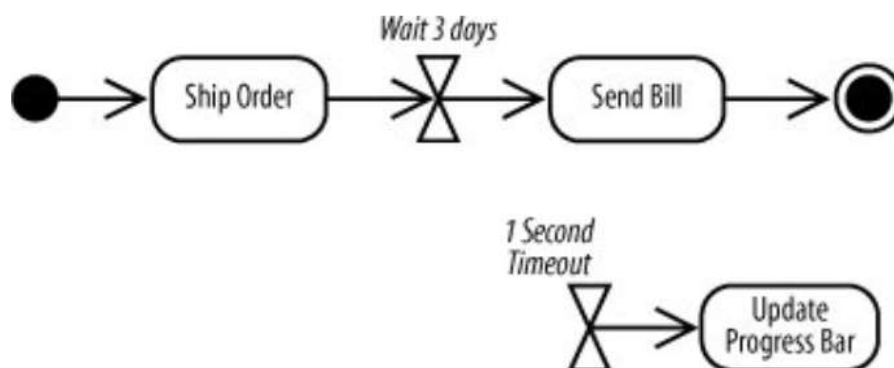
شروع هر نمودار اکتیویتی یک دایره توپر هست که بهش Initial Node گفته میشه و سپس با فلش هایی روبه پایین حرکت میکنیم که این فلش ها رو Edge یا Path میگویم و به سری باکس های مستطیلی که بهشون Action گفته میشه هم برای نشون دادن کاری هست که در هر قدم باید انجام بشه. علامت لوزی یعنی تصمیم گیری و Desecion که نشون میده که این فلو بعد از این نقطه دو حالت براش ممکنه پیش بیاد. مثلاً یا کاربر لاگین هست و یا کاربر لاگین نیست و به لوزی دیگه هم داریم که فلش ها بهش وارد میشن، یعنی چند تا حالت مختلف به یه لوزی وارد میشن و این یعنی از این مرحله به بعد فلوهاشون یکی میشه ولی در اولی فلش ها از لوزی خارج میشدند. به حالت دومی Merge میگویم و در نهایت Final Node رو داریم که یه دایره توپر هست که دورش یه دایره گرد داره.

به فلش های ورودی به یک نود Incoming Edge و به خروجی Outcoming Edge گفته میشه. Action ها قدم های فعال در یک Process هستند و هر اکشنی میتونه یه چیز محاسباتی یا منطقی باشه، مثلاً محاسبه مالیات یا بررسی لاگین بودن و ... ضمناً میشه دور هر نمودار Activity یک باکس هم کشید که بهش میگوین Activity Frame و اختیاری هست و معمولاً نمیکشند و اگر در یک صفحه قرار شد بیش از یک نمودار اکتیویتی باشه اینکار میکنیم و داخل این اکتیویتی فریم هم اسم اکتیویتی رو Activity Name بصورت بولد مینویسیم.

در Outgoing Edge های هر Decssion روی edge ها باید داخل براکت شرطی که اتفاق افتاده رو بنویسیم و به اینها Guard Condition میگویم و البته این خروجی ها بصورت true-false هستند و فقط یکی از دو حالت ممکن را دارند. دقت کنید اگر بیش از دو تا guard condition داشته باشیم باید خیلی دقت کنیم که فقط یکی از همه اونها درست باشه

زمان در Activity Diagram

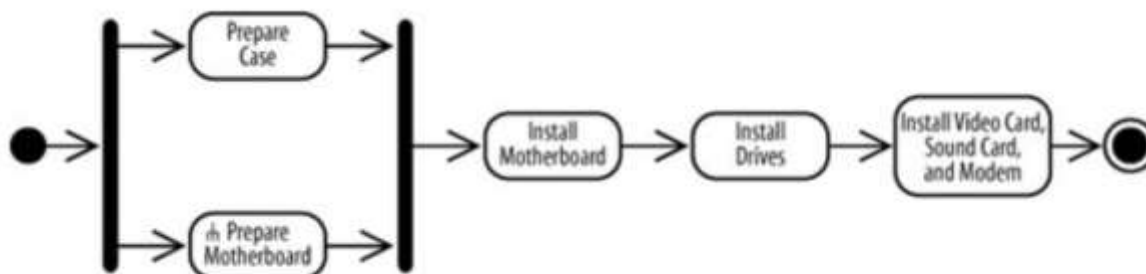
در نمودار اکتیویته بعضی مواقع زمان یک عامل در مدل سازی شما محسوب میشود. مثلاً نیاز است بعضی جاها یک وقفه زمانی ایجاد شود. به عنوان مثال ۳ روز بعد از اجرای فعالیت خاصی باید بک ار سال محصول اتفاق بیفتد یا اینکه مثلاً هر ۳ ماه یکبار باید بک آپ گیری اتفاق میفتد. علامت زمان در نمودار اکتیویته بصورت ساعت شنی است که روش میزان زمان را مینویسیم مثلاً `wait 3 weeks` نوشته می شود. اگر برای `time` ورودی خاصی وجود نداشته باشد این یعنی اکشن بعد آن بصورت `recurring` انجام و تکرار خواهد شد. مثلاً هر ۱ ثانیه یکبار.



استفاده از `time` بدون ورودی یکی از روش های شروع یک اکتیویته دیاگرام است.

فراخوانی سایر اکتیویته ها

وقتی ما جزئیات بیشتری به `activity diagram` اضافه میکنیم کم نمودار ما بسیار طولانی و بزرگ میشود و یا اینکه یک سری از کارها در اکتیویته در جاهای دیگر نیز ممکن است اتفاق بیفتند. در چنین شرایطی برای افزایش `readability` میتوانیم این بخش جزئیات یا تکراری را در یک دیاگرام دیگر ایجاد کنیم و در این دیاگرام اصلی فراخوانی کنیم و کمک کنیم دیاگرام اصلی ما مرتب تر باشد.

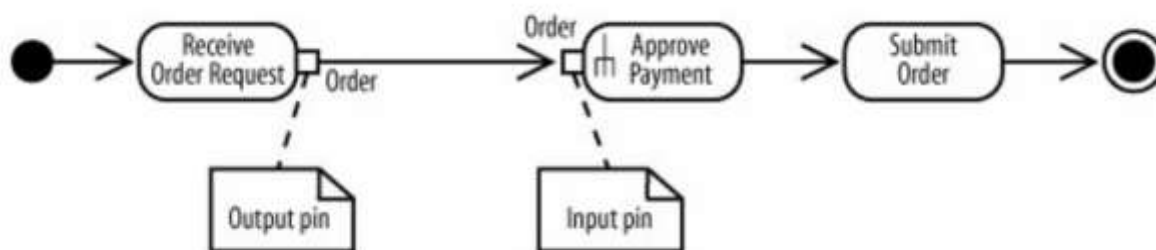


همانطور که در شکل بالا میبینید اکشن prepare Motherboard با علامت چنگک نشون میده که این خودش یک اکتیویتی دیاگرام جدا هست که اینجا include شده. خود اون اکتیویتی مجزا دقیقاً نقطه شروع و پایان خواهد داشت

آبجکت ها Objects در نمودار اکتیویتی دیاگرام

بسیاری جاها در نمودار اکتیویتی ما نیاز داریم که برای انجام یک کاری دسترسی به یه اطلاعاتی داشته باشیم. مثلاً برای بررسی سوابق مشتری نیاز داریم اطلاعات مشتری را داشته باشیم. این حالت را دیتا آبجکت میگوییم. این آبجکت ها قرار نیست فقط دسترسی به دیتا یا یک چیز نرم افزاری باشند، بعضی وقتها آبجکت ها میتوانند حتی یک کار فیزیکی باشند که باید با دست انجام شود. آبجکت یعنی در اون node این موضوع فراهم است. این دیتا فراهم است یا اینکار فیزیکی فراهم است و ...

آبجکت ها را با مستطیل با گوشه های ۹۰ درجه نشون میدیم. یک روش بهتر برای نشان دادن آبجکت ها استفاده از پین است. یعنی یک مربع کوچک که به ابتدا یا انتهای یک Action ای چسبیده است.



۳. UML Usecase Diagram

نمودار use case در ساده ترین حالت خود نمایشی از تعامل کاربر با سیستم است که رابطه بین کاربر و use case های متفاوتی که کاربر در آن درگیر است را نشان می دهد. این نمودار UML شکل اولیه الزامات سیستم برای یک برنامه نرم افزاری جدید در دست توسعه است. use case ها رفتار مورد انتظار (چه چیزی) را مشخص می کند و نه روش دقیق ایجاد آن (چگونه). مفهوم کلیدی مدل سازی use case این است که به ما کمک می کند یک سیستم را از دیدگاه کاربر نهایی طراحی کنیم. بنابراین، یک نمودار use case، نقشه های اولیه سیستم شما است، که نمایش ساده و گرافیکی کاری را که سیستم واقعاً باید انجام دهد را ارائه می کند.

نمودارهای use case، عملکردهای سطح بالا و دامنه یک سیستم را توصیف می کنند. این نمودارها همچنین تعاملات بین سیستم و بازیگران آن را مشخص می کند. در واقع توصیف می کنند که سیستم چه کاری انجام می دهد و بازیگران چگونه از آن استفاده می کنند، اما با نحوه عملکرد داخلی سیستم ارتباطی ندارد.

این طراحی سطح بالا بارها و بارها اصلاح می شود تا تصویری کامل و کاربردی از سیستم به دست آید.

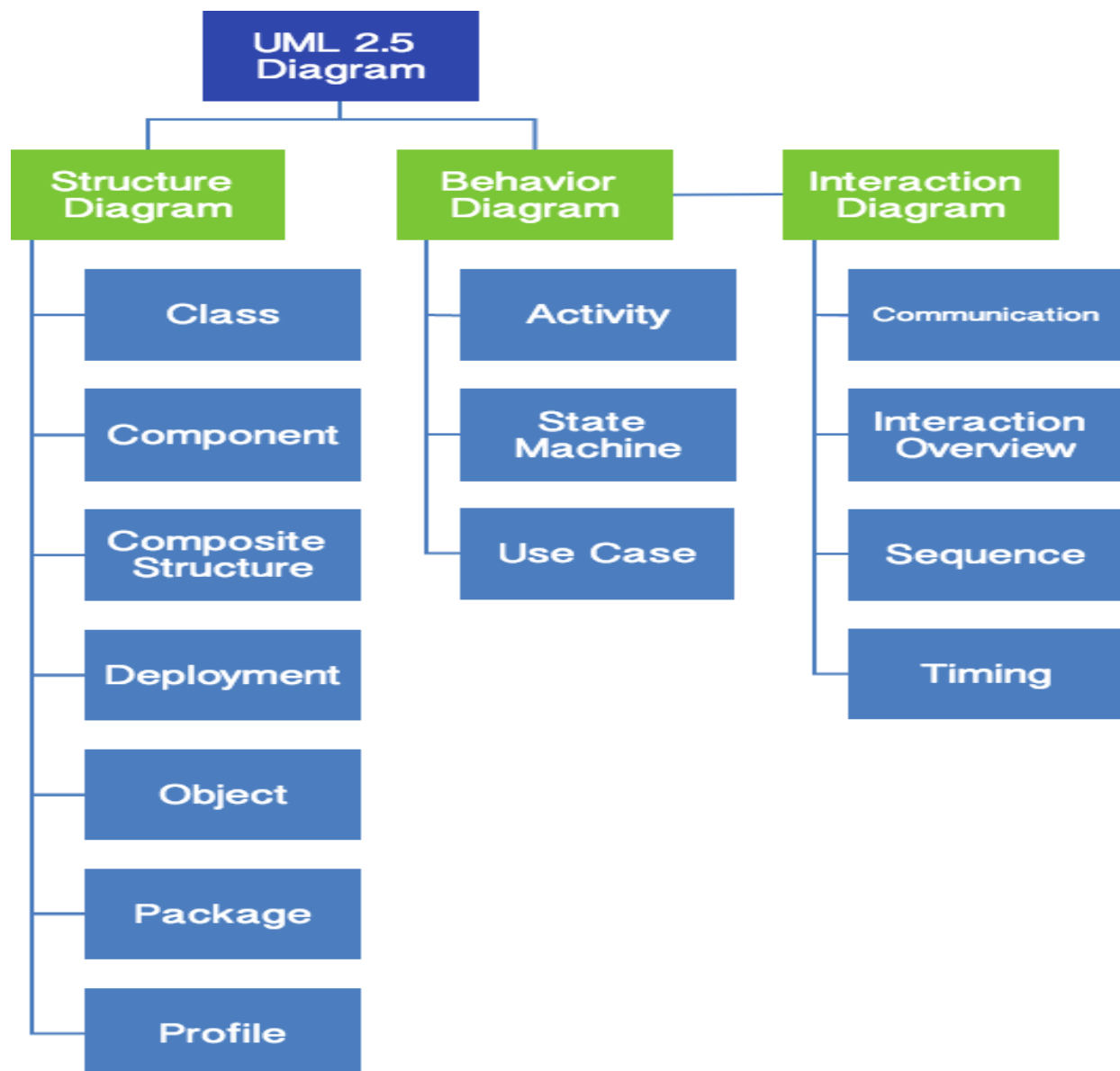
هر use case باید نتایج قابل مشاهده و ارزشمندی را به بازیگران سیستم ارائه دهد.

نمودار use case معمولاً ساده است. use case جزئیات را نشان نمی دهد.

نمودار use case باید ساده باشد و فقط شامل چند شکل باشد. اگر سیستم شما بیش از مثلاً ۲۰، use case دارد، احتمالاً در مسیر اشتباهی از مدلسازی سیستم خود در استفاده از نمودار usecase هستید.

نمودارهای use case فقط الزامات عملکردی یک سیستم را نشان می دهد. سایر الزامات مانند قوانین کسب و کار، الزامات کیفیت خدمات و محدودیت های پیاده سازی باید به طور جداگانه و دوباره با سایر نمودارهای UML نشان داده شوند.

شکل زیر سلسله مراتب نمودار UML و موقعیت UML Use Case Diagram را نشان می دهد. همانطور که مشاهده می کنید، نمودارهای use case از خانواده نمودارهای رفتاری هستند.

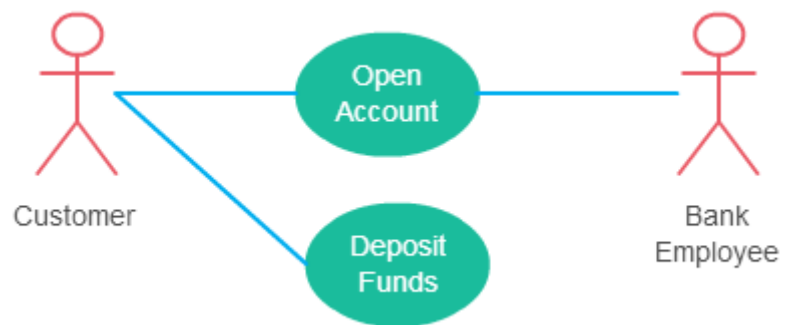


بازیگر (Actor)

- کسی که با use case (عملکرد سیستم) تعامل دارد.
- با اسم مشخص (مثلا مشتری/متصدی باجه و ...) نامگذاری می شود.
- بازیگر نقشی را در business (کسب و کار مورد نظر/محیط عملیاتی مورد نظر ما) ایفا می کند.
- مشابه مفهوم کاربر است، اما یک کاربر می تواند نقش های متفاوتی ایفا کند
مثلا:
یک پروفیسور می تواند مربی و محقق نیز باشد
۲ نقش با دو سیستم بازی می کند
- بازیگر بر روی use case ها اثر می گذارد.
- بازیگر نسبت به سیستم (ورودی ها) مسئولیت دارد و Actor انتظاراتی از سیستم (خروجی ها) دارد.

مورد کاربرد (use case)

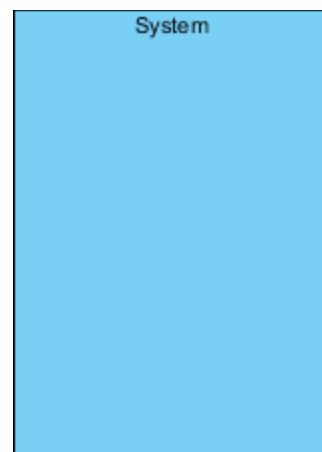
- عملکرد سیستم (System function) فرآیند بصورت خودکار یا دستی است.
- با فعل + اسم (verb + Noun) یا عبارت اسمی نامگذاری می شود.
- یعنی یه کاری بکن (Do something)
- هر بازیگر باید به یک use case مرتبط باشد، در حالی که برخی use case ها ممکن است به بازیگری مرتبط نباشند.



لینک ارتباط (Communication Link)

ارتباط یک Actor با یک use case توسط یک پیوند (خط) نشان داده می شود.

بازیگران ممکن است توسط association ها به use case ها متصل شوند، که نشان می دهد بازیگر و use case با استفاده از پیام ها با یکدیگر ارتباط برقرار می کنند.



Boundary of system

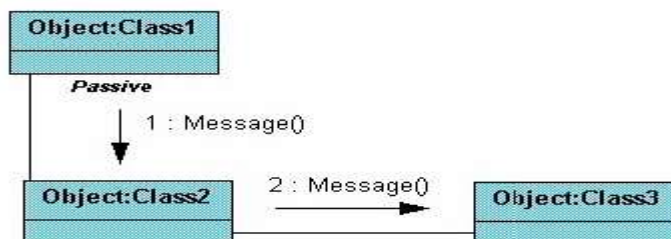
مرز سیستم (Boundary of system)

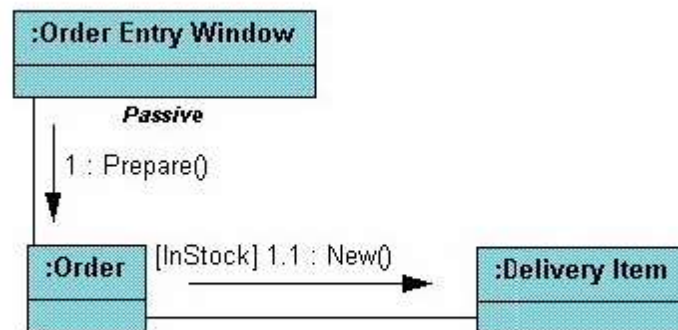
- مرز سیستم به طور بالقوه کل سیستم است که در سند الزامات تعریف شده است.
- برای سیستم های بزرگ و پیچیده، هر ماژول ممکن است مرز سیستم باشد. به عنوان مثال، برای یک سیستم ERP برای یک سازمان، هر یک از ماژول ها مانند پرسنل، حقوق و دستمزد، حسابداری و غیره.

۴. Collaboration Diagram UML

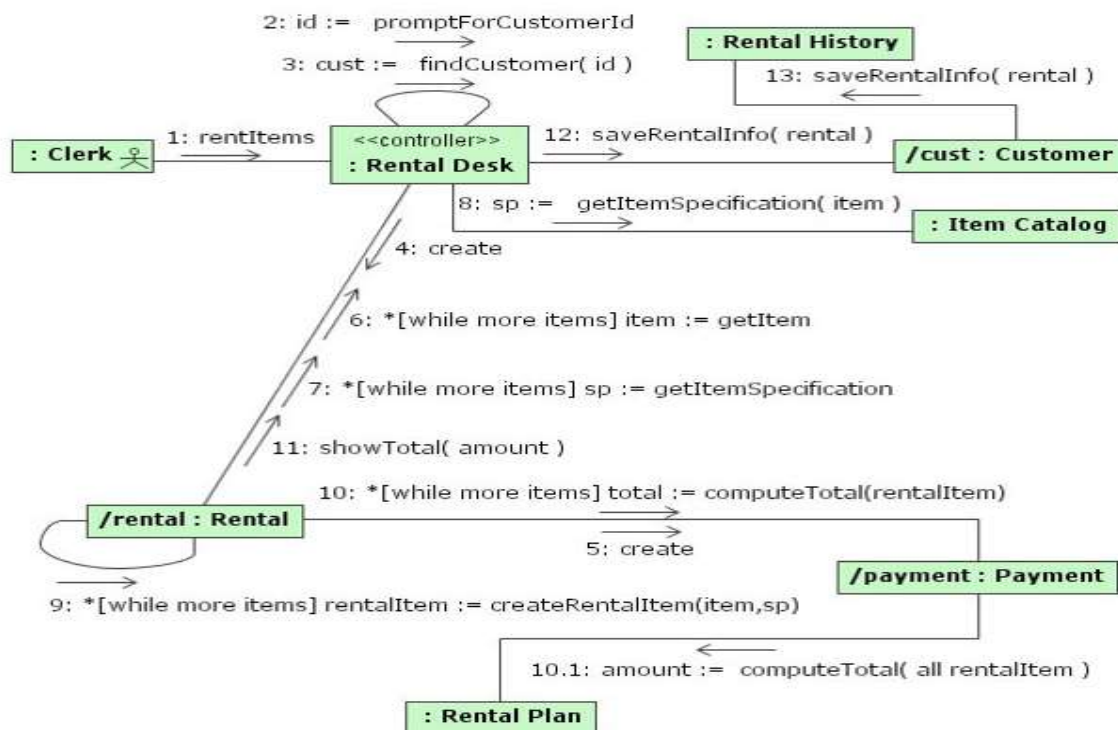
نمودار همکاری (collaboration diagram)، نموداری تعاملی است که بر رابطه ی اشیاء ارسال کننده و دریافت کننده ی پیام تاکید دارد، در واقع همان اطلاعات نمودار را در قالبی دیگر نمایش می دهد:

مثال هایی از نمودار همکاری (collaboration diagram)، در زیر بررسی میکنیم:





Collaboration Diagram (for rentItem)



۵. Class Diagram UML

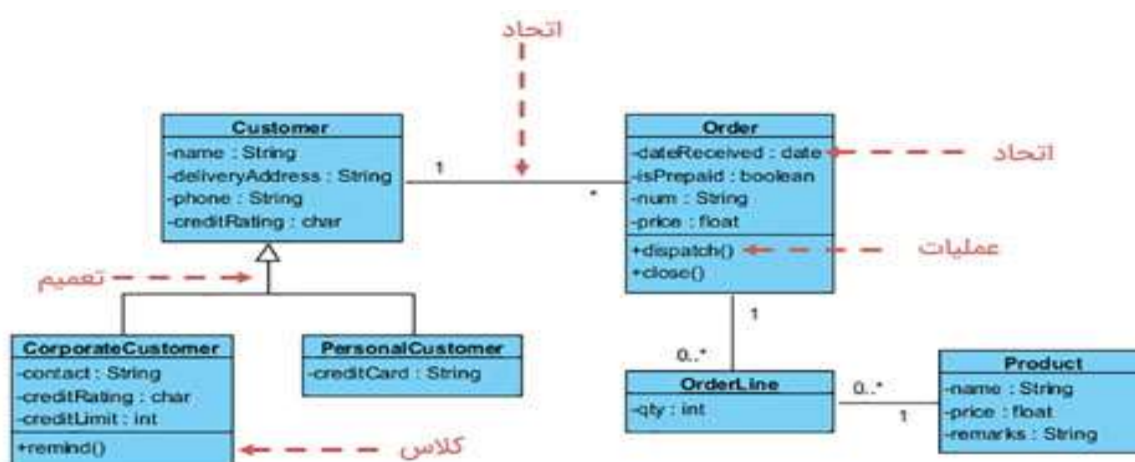
نمودار کلاس (Class Diagram) در UML یک نمودار ایستاست و در واقع این نمودار دید ایستایی یک برنامه را نشان میدهد. Class Diagram برای تجسم، توصیف و یا مستند سازی و در نهایت ساخت کد اجرایی یک سیستم و یا نرم افزار کاربرد دارد. همچنین Class Diagram محدودیت های تحمیل شده بر یک سیستم را شرح میدهد. نمودارهای کلاس به طور گسترده ای در مدل سازی سیستم های شی گرا مورد استفاده قرار می گیرند زیرا آن ها تنها نمودارهای UML هستند که می توانند مستقیماً با زبان

های شی گرا نگاشت شوند. نمودار کلاس (Class Diagram) در UML مجموعه ای از کلاس ها ، رابط ها ، انجمن ها ، همکاری ها و محدودیت ها را نشان می دهد و به طور کلی یک نمودار ساختاری است.

رابطه

سه نوع رابطه اساسی مهم وجود دارند:

- **اتحاد: (Association)** نماینده روابطی بین وهله های انواع است (یک شخص که برای یک شرکت کار می کند، یک شرکت چند اداره دارد).
- **وراثت: (Inheritance)** بدیهی ترین افزودنی به نمودارهای ER برای استفاده در شیء گرایی است. تناظر پی واسطه ای با وراثت در طراحی شیء گرایی دارد.
- **تجمع: (Aggregation)** شکلی از ترکیب بندی شیء در طراحی شیء گرا محسوب می شود.



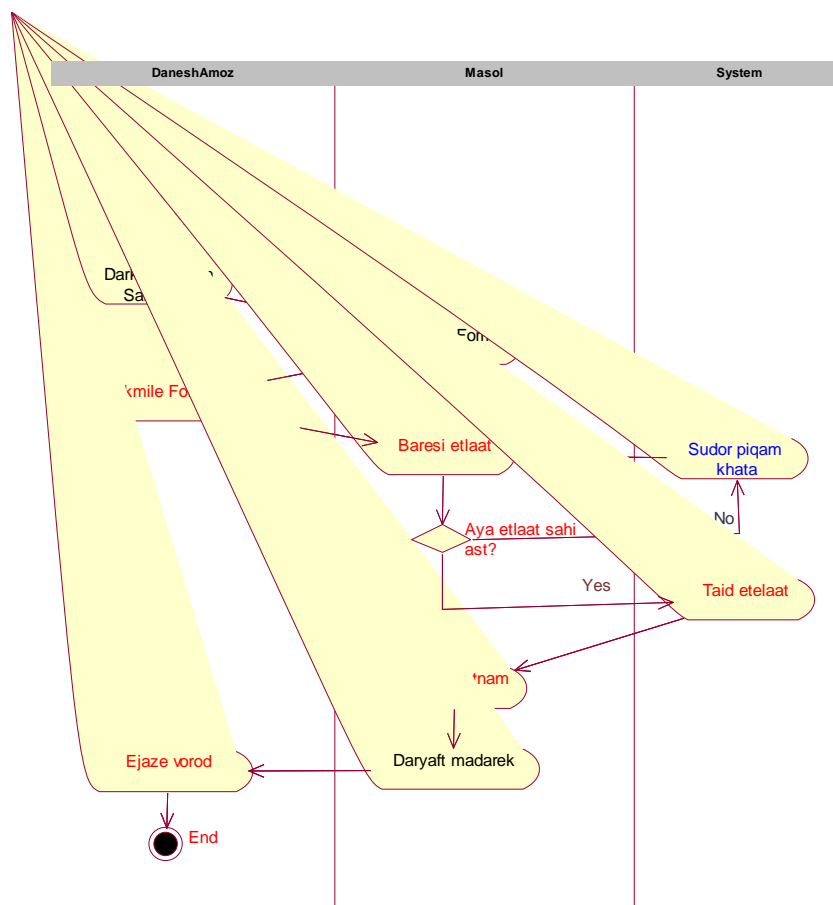
کاربرد Class Diagram

- نمودار کلاس برای مدل سازی نمای استاتیک (static view) یک سیستم استفاده می شود.
- نمای ایستا واژگان سیستم را توصیف می کند.
- نمودار کلاس به عنوان پایه و اساس نمودارهای اجزا (component) و استقرار (deployment) در نظر گرفته می شود.
- نمودار کلاس به وضوح نگاشت با زبان های شی گرا مانند ++Java ، C و ... را نشان می دهد، نمودار کلاس به طور کلی برای ساخت برنامه های کاربردی استفاده می شود.

حال با توجه به توضیحات داده شده در خصوص UML و Diagram ها ، نمودارهای مربوط به مسئله ما با نرم افزار **IBM Rational Rose Enterprise** یکی از ابزارهای مدل سازی نرم افزار از طریق عمومی ترین زبان مدل سازی (UML) می باشد، رسم شده و در زیر قرار داده شده است :

۱- Activity Diagram

در شکل زیر نمودار فعالیت ثبت نام در مدرسه غیر انتفاعی نشان داده شده است.

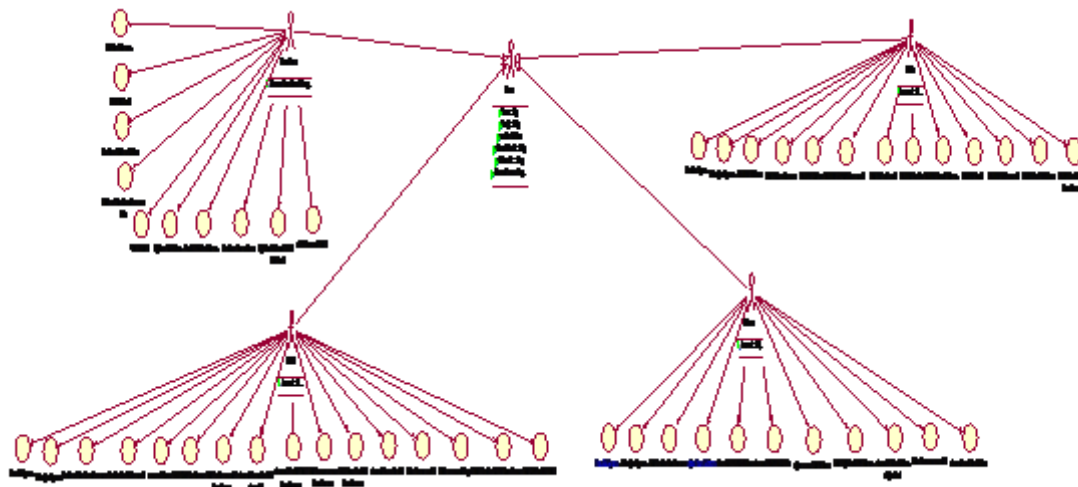


نمودار فعالیت ثبت نام در مدرسه غیر انتفاعی

همانطور که از شکل بالا مشاهده می گردد، فرآیند انجام نمودار فعالیت برای سیستم ذکر شده به شرح ذیل می باشد.

ابتدا دانش آموز وارد سیستم می شود و از مسئول ثبت نام درخواست فرم مربوطه را کرده و فرم مورد نظر را دریافت و تکمیل میکند. مسئول اطلاعات دانش آموز را بررسی و ثبت در سیستم میکند که ممکن است سیستم خطا صادر کرده ،(قبلا ثبت نام کرده و یا مشخصات اشتباه است) یا اطلاعات ثبت شود. بعد از ثبت نهایی مسئول مدارک مربوطه را دریافت و اجازه ورود دانش آموز به مدرسه داده میشود.

۲- Usecase Diagram



شکل یوزکیس دیاگرام سیستم مدرسه غیرانتفاعی

سیستم مدرسه غیرانتفاعی

با توجه به شکل بالا ، یک دیاگرام مدرسه غیرانتفاعی ، که شامل چهار اکتور با یوزکیس های مختلف می باشد را مشاهده می کنید. این دیاگرام فرآیند ثبت نام و امور مربوط به آموزش دانش آموزان را در مدرسه نمایش می دهد که در ادامه به تجزیه و تحلیل هر کدام از اکتورها و یوزکیس ها بطور جداگانه پرداخته می شود. .

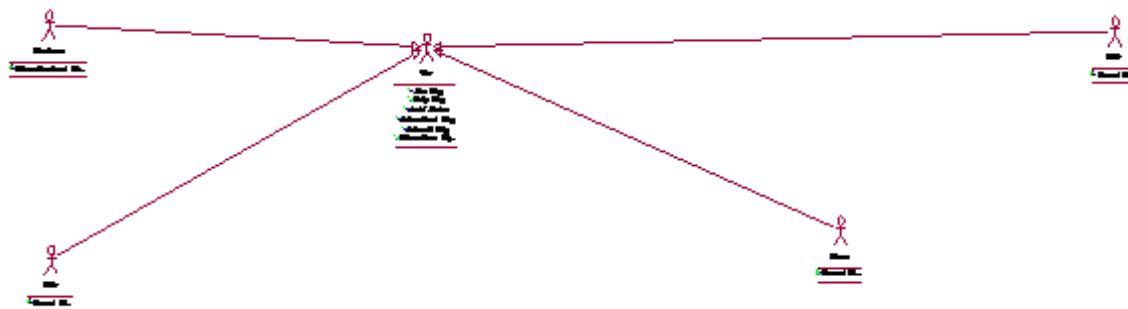
• اکتورهای سیستم مدرسه غیر انتفاعی

۱- دانش آموز

۲- دبیران

۳- معاون

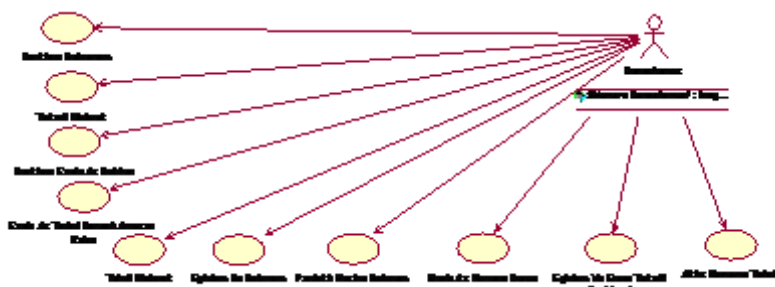
۴- مدیر



شکل اکتورهای سیستم مدرسه غیر انتفاعی

۱_ دانش آموز

این اکتور تمامی فرآیندهای مرتبط با ثبت نام و آموزش را طی ده یوزکیس انجام می دهد.



شکل یوزکیس های دانش آموز

• یوزکیس های دانش آموز :

_درخواست ثبت نام

در این مرحله دانش آموز با ورود به مدرسه درخواست ثبت نام می دهد.

_تکمیل مدارک

دانش آموز مدارک مورد درخواست و نیاز جهت ثبت نام را تکمیل می کند.

_درخواست اطلاع از دبیران

در این مرحله دانش آموز خواستار اطلاعات مربوط به نام و سابقه آموزشی دبیران مربوطه می شود.

_اطلاع از تعداد دانش آموزان کلاس

دانش آموز تعداد دانش آموزان کلاس را بررسی و اطلاع کسب می کند.

_تحويل مدارک

دانش آموز مدارک تکمیل شده را جهت انجام امور ثبت نام ارائه می دهد.

_اقدام به ثبت نام

در این مرحله دانش آموز بعد از بررسی های مربوطه و تکمیل مدارک اقدام به ثبت نام می کند.

پرداخت هزینه ثبت نام

دانش آموز بعد از نام نویسی هزینه ثبت نام را پرداخت می کند.

اطلاع از نمرات دروس

دانش آموز در این مرحله از نمرات کسب شده مربوط به هر دروس اطلاع کسب می کند.

اقدام و ارائه تکالیف درخواستی

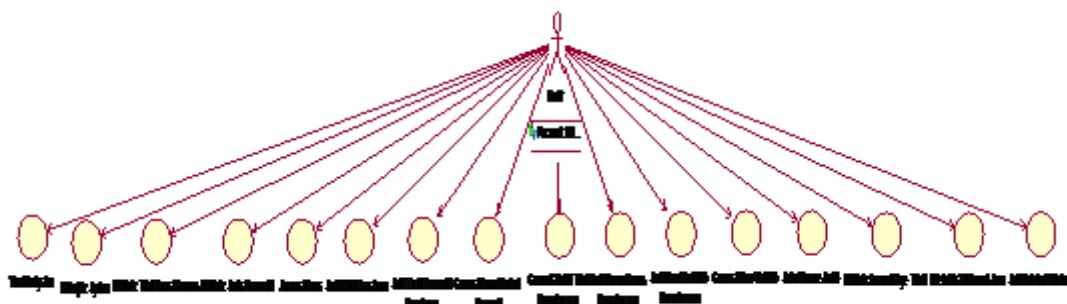
دانش آموز به انجام تکالیف درخواستی دبیران پرداخته و آنها را جهت ارائه آماده می نماید.

اخذ کارنامه تحصیلی

دانش آموز در این مرحله کارنامه تحصیلی مربوط به هر ترم را دریافت می نماید.

۲_ دبیر

فعالیت های مربوط به آموزش و پرورش دانش آموزان را طی شانزده یوزکیس توسط این اکتور انجام می شود.



شکل یوزکیس های دبیر

• یوزکیس های دبیر:

ورود به سیستم

دبیر در این مرحله با وارد نمودن نام مدیری و کلمه عبور اختصاصی خود وارد سیستم شده و فعالیت خود را آغاز می نماید.

اطلاع از تعداد دانش آموزان

در این مرحله دبیر از تعداد دانش آموزان کلاس اطلاع کسب می کند.

اطلاع از سرفصل آموزشی

دبیر از سرفصل های آموزشی اطلاع کسب می نماید.

آموزش دروس

در این مرحله دبیر به آموزش دروس به دانش آموزان می پردازد.

ثبت اطلاع دانش آموزان

دبیر اطلاعات دانش آموزان را در سیستم وارد و ثبت می نماید.

ثبت وضعیت آموزشی دانش آموزان

در این مرحله دبیر روند وضعیت آموزشی دانش آموزان را ثبت می کند.

گزارش روند سرفصل آموزشی

دبیر چگونگی و روند سرفصل آموزشی را گزارش می دهد.

گزارش وضعیت تحصیلی دانش آموزان

در این مرحله دبیر گزارش وضعیت تحصیلی مربوط به هر دانش آموز را به مدیر آموزشگاه و اولیا ارائه می دهد.

ثبت نمرات آزمون دانش آموزان

دبیر نمرات کسب شده مربوط به آزمون دروس توسط دانش آموزان را در سیستم ثبت می نماید.

ثبت حضور و غیاب دانش آموزان

در این مرحله حضور و غیاب دانش آموزان روزانه توسط دبیر در سیستم ثبت می شود.

گزارش حضور و غیاب

دبیر حضور و غیاب دانش آموزان را در پایان ساعت آموزشی به معاون و مدیر آموزشگاه گزارش می دهد.

صدور کارنامه اولیه

دبیر در این مرحله کارنامه اولیه برای هر دانش آموز را صادر می کند.

اطلاع از آموزش پایه تحصیلی

دبیر از امور آموزش مربوط به پایه تحصیلی اطلاع کسب می نماید.

ارتباط با ولی دانش آموزان

دبیر جهت گزارش و اطلاع رسانی وضعیت آموزشی و نمرات دانش آموزان به برقراری ارتباط با اولیا می پردازد.

ثبت درخواست مرخصی

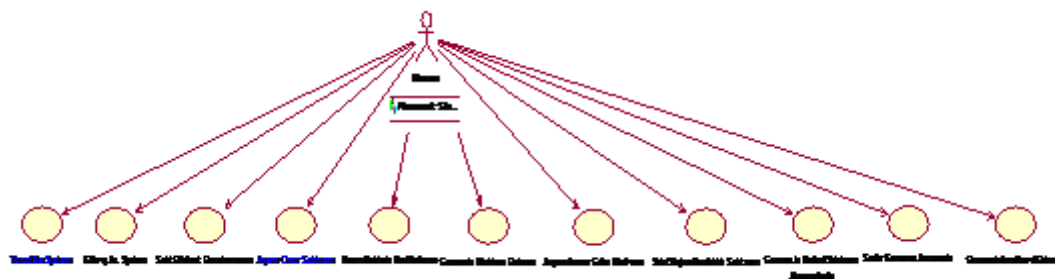
دبیر درخواست مرخصی را در این مرحله ثبت می نماید.

خروج از سیستم

دبیر بعد از اتمام فعالیت و ثبت های مربوطه از سیستم خارج می شود.

۳_معاون

تمامی فعالیت های نظارت بر فعالیت دبیران و دانش آموزان توسط این اکتور و طی یازده یوزکیس انجام می شود.



شکل یوزکیس های معاون

• یوزکیس های معاون :

_ورود به سیستم

معاون با وارد نمودن نام مدیری و کلمه عبور به سیستم دسترسی نموده و شروع به فعالیت می کند.

-ثبت اطلاعات دانش آموزان

معاون اطلاعات مربوط به هر دانش آموز را در سیستم ثبت می کند.

_انجام امور ثبت نام

در این مرحله معاون امور مربوط به ثبت نام هر دانش آموز را انجام می دهد .

_نظم بخشی به مدرسه

معاون نظم در مدرسه را مدیریت می کند.

_گزارش مرخصی دبیران

معاون تعداد روز و نوع مرخصی دبیران را بررسی و جهت تنظیم گزارش آن اقدام می نماید.

_انجام امور اداری

معاون امور اداری را در این مرحله انجام می دهد.

_ثبت وجه پرداختی ثبت نام

معاون هزینه های پرداختی ثبت نام توسط دانش آموزان را در سیستم ثبت می کند.

_گزارش تعداد ثبت نام انجام شده :

_ در این مرحله معاون تعداد دانش آموزان ثبت نامی را گزارش می دهد.

_صدور کارنامه آموزشی

معاون کارنامه تحصیلی جهت ارائه به دانش آموزان را صادر می کند.

_گزارش عملکرد دبیران

در این مرحله معاون عملکرد دبیران در زمینه نحوه و چگونگی آموزش و همچنین برخورد با دانش آموزان را بررسی و

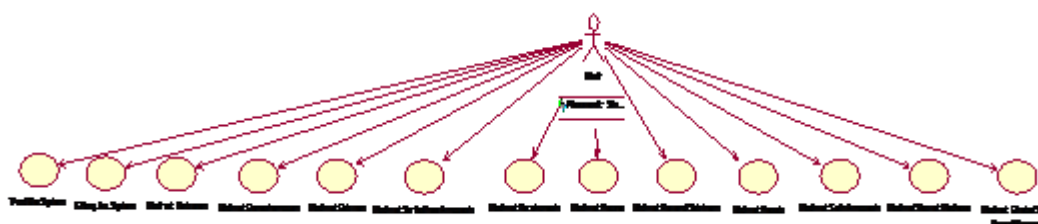
گزارش جهت ارائه به مدیر را تنظیم می کند.

_خروج از سیستم

معاون بعد از اتمام فعالیت و ثبت های مربوطه از سیستم خارج می شود.

۴_مدیر

این اکتور تمامی فعالیت های مرتبط با مدرسه را تحت سیزده یوزکیس مدیریت می کند.



شکل یوزکیس های مدیر

• یوزکیس های مدیر:

_ورود به سیستم

مدیر با وارد نمودن نام مدیری و کلمه عبور به سیستم دسترسی نموده و شروع به فعالیت می کند.

_مدیریت مدیران

مدیر در این مرحله می تواند به فعالیت های مدیران دسترسی داشته باشد و مدیریت کند.

_مدیریت دانش آموزان

در این مرحله مدیر درخواست ها و وضعیت تحصیلی دانش آموزان را بررسی و مدیریت می کند.

_مدیریت دبیران

مدیر فعالیت ها و عملکرد دبیران را پیگیری و مدیریت می کند.

_مدیریت سرفصل های آموزشی

مدیر چگونگی ارائه سرفصلهای آموزشی مربوط به هر پایه تحصیلی را مدیریت می کند.

_مدیریت تراکنش

کنترل و بررسی پرداخت های انجام شده ، اهم از پرداخت های انجام شده در این مرحله توسط مدیر ، مدیریت می شود.

_مدیریت هزینه

در این مرحله مدیر هزینه های وارده بر مدرسه را مدیریت می کند .

_مدیریت روند ثبت نام

مدیر امور مربوط به ثبت نام و چگونگی آن را مدیریت می کند.

_مدیریت حساب

مدیر امور مالی مدرسه را کنترل و مدیریت می کند.

_مدیریت سطح آموزش

مدیر سطح آموزشی و علمی هر پایه را مدیریت می کند.

_مدیریت وضعیت مدرسه

مدیر وضعیت و امور مربوط به ساختمان مدرسه را بررسی و مدیریت می کند.

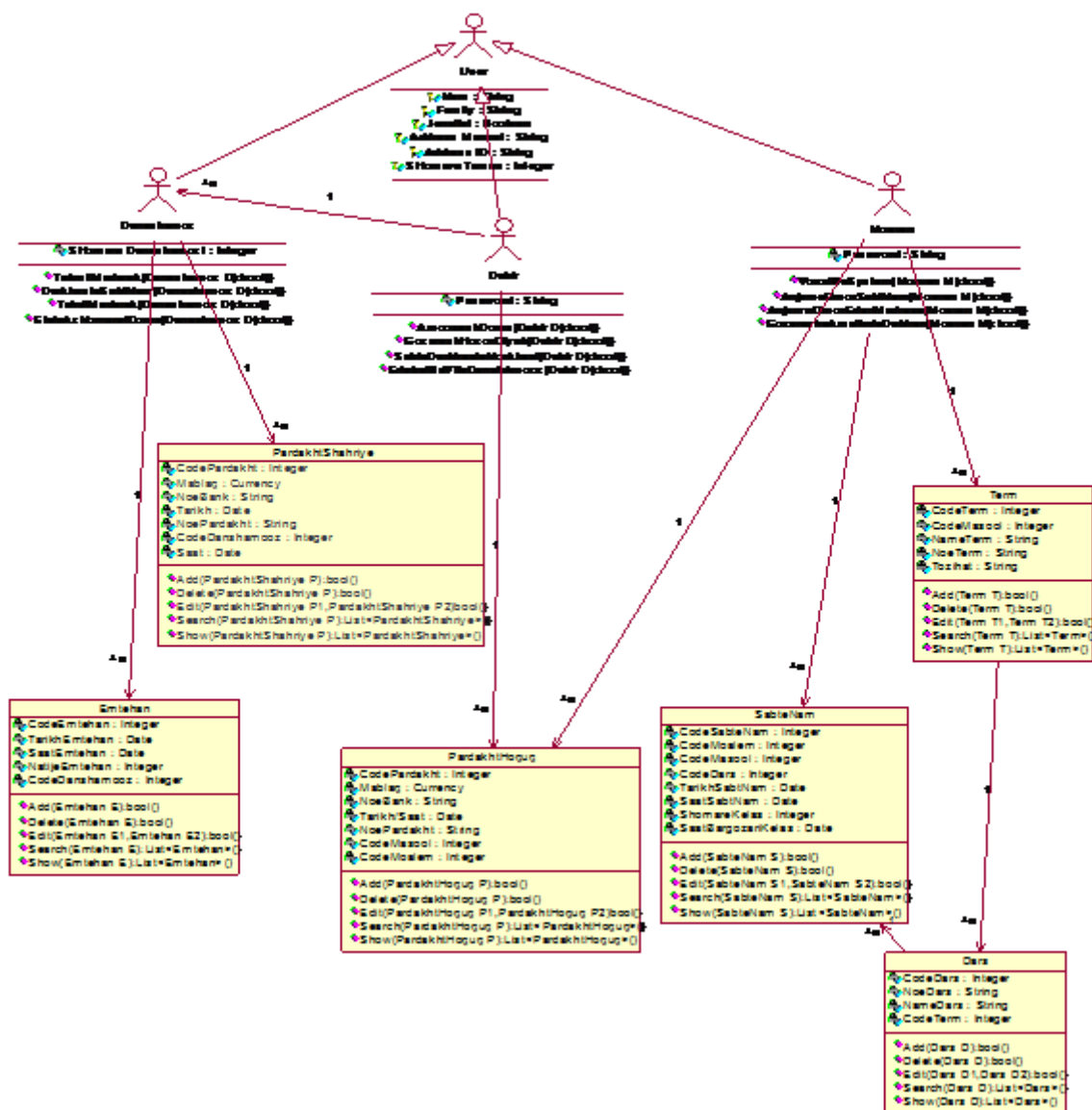
_مدیریت ارتباط ولی دانش آموزان

مدیر ارتباط ولی دانش آموزان با کادر آموزشی را مدیریت می کند.

_خروج از سیستم

مدیر بعد از اتمام فعالیت و ثبت های مربوطه از سیستم خارج می شود.

در شکل زیر نمودار کلاس دیاگرام سیستم مدارس غیر انتفاعی نشان داده شده است.



نمودار کلاس دیاگرام سیستم مدارس غیر انتفاعی

بیان مسئلہ تحقیق:

در این تحقیق به ارائه نمودار کلاس دیاگرام و تجزیه و تحلیل سیستم مدارس غیر انتفاعی با استفاده از نرم افزار رشنال رز پرداخته می شود. در این سیستم، مجموعه ای از موجودیت ها و اشیاء، جهت رسیدن به یک هدف خاص در تعامل هستند. موجودیت های سیستم مدارس غیر انتفاعی عبارتند از:

- پرداخت شهریه
- معلم
- دانش آموز
- پرداخت حقوق
- ثبت نام
- امتحان
- مسئول ثبت
- ترم
- درس

هر کدام از این موجودیت ها که بیانگر یک جدول در بانک اطلاعاتی است، متشکل از چندین ویژگی می باشد که در نمودار ذیل به وضوح مشاهده می گردد.

همان طور که از نمودار کلاس دیاگرام بالا که با ابزار رشنال رز تهیه شده است، مشاهده می گردد هر دانش آموز پس از اینکه به سیستم مدارس غیر انتفاعی وارد می شود از مسئول ثبت درخواست خدمات می کند. هر دانش آموز در این سیستم چندین پرداخت شهریه را انجام میدهد بنابراین ارتباط بین دانش آموز و پرداخت شهریه یک ارتباط یک به چند می باشد.

هر دانش آموز چندین بار فرآیند امتحان را میتواند داشته باشد در این حالت ارتباط بین دانش آموز و امتحان یک ارتباط یک به چند می باشد.

هر معلم چندین دانش آموز را شامل میشود در این حالت ارتباط بین معلم و دانش آموز یک ارتباط یک به چند می باشد.

هر معلم چندین پرداخت حقوق را دریافت میکند در این حالت ارتباط بین معلم و پرداخت حقوق یک ارتباط یک به چند می باشد.

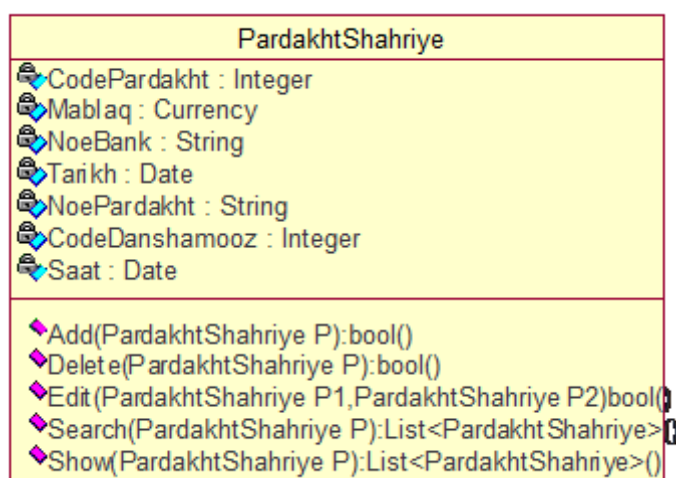
هر مسئول ثبت نام چندین ثبت نام را انجام میدهد در این حالت ارتباط بین مسئول ثبت نام و ثبت نام یک ارتباط یک به چند می باشد.

هر مسئول ثبت نام چندین ترم را انجام میدهد در این حالت ارتباط بین مسئول ثبت نام و ترم یک ارتباط یک به چند می باشد.

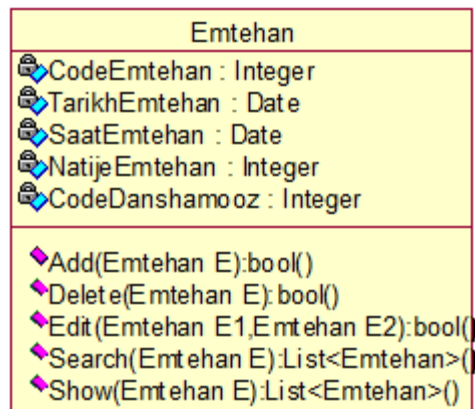
هر ترم چندین درس را شامل میشود در این حالت ارتباط بین ترم و درس یک ارتباط یک به چند می باشد.
هر درس چندین ثبت نام را شامل میشود در این حالت ارتباط بین درس و ثبت نام ارتباط یک به چند می باشد.
اسامی مهمترین کلاس های این سیستم عبارتند از:

کلاس سیستم مدارس غیر انتفاعی

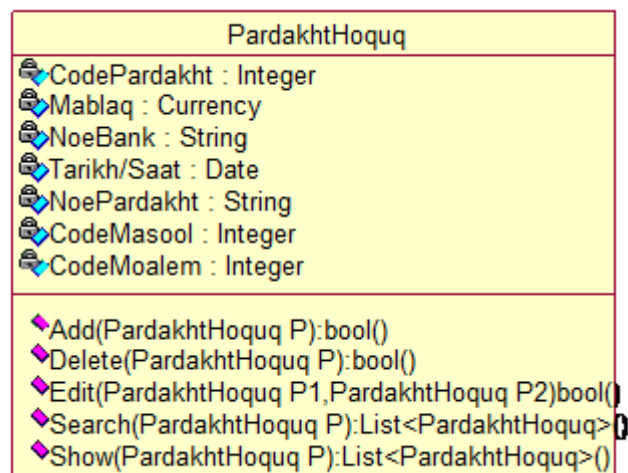
در نمودار زیر کلاس مربوط به سیستم مدارس غیر انتفاعی نشان داده شده است.



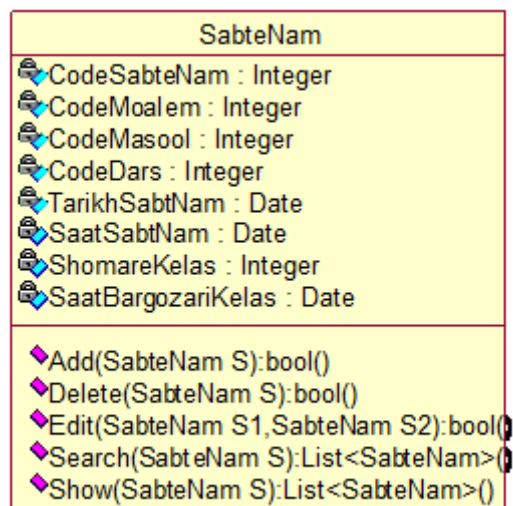
همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد پرداخت، مبلغ، تاریخ، نوع بانک، نوع پرداخت، کد دانش آموز و ساعت است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک پرداخت شهرییه می باشد.



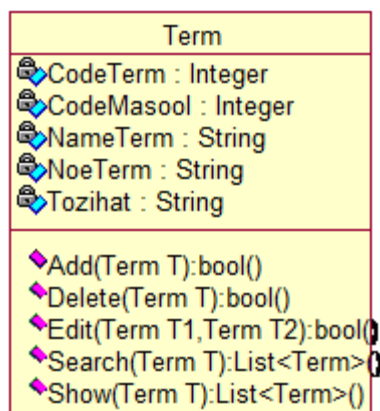
همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد امتحان، تاریخ امتحان، ساعت امتحان، نتیجه امتحان و کد دانش آموز است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک امتحان می باشد.



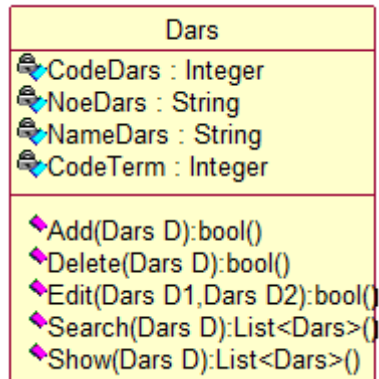
همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد پرداخت، مبلغ، نوع بانک، تاریخ و ساعت، نوع پرداخت، کد مسئول و کد معلم است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک پرداخت حقوق می باشد.



همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد ثبت نام، کد معلم، کد مسئول، کد درس، تاریخ ثبت نام، ساعت ثبت نام، شماره کلاس و ساعت برگزاری کلاس است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک ثبت نام می باشد.



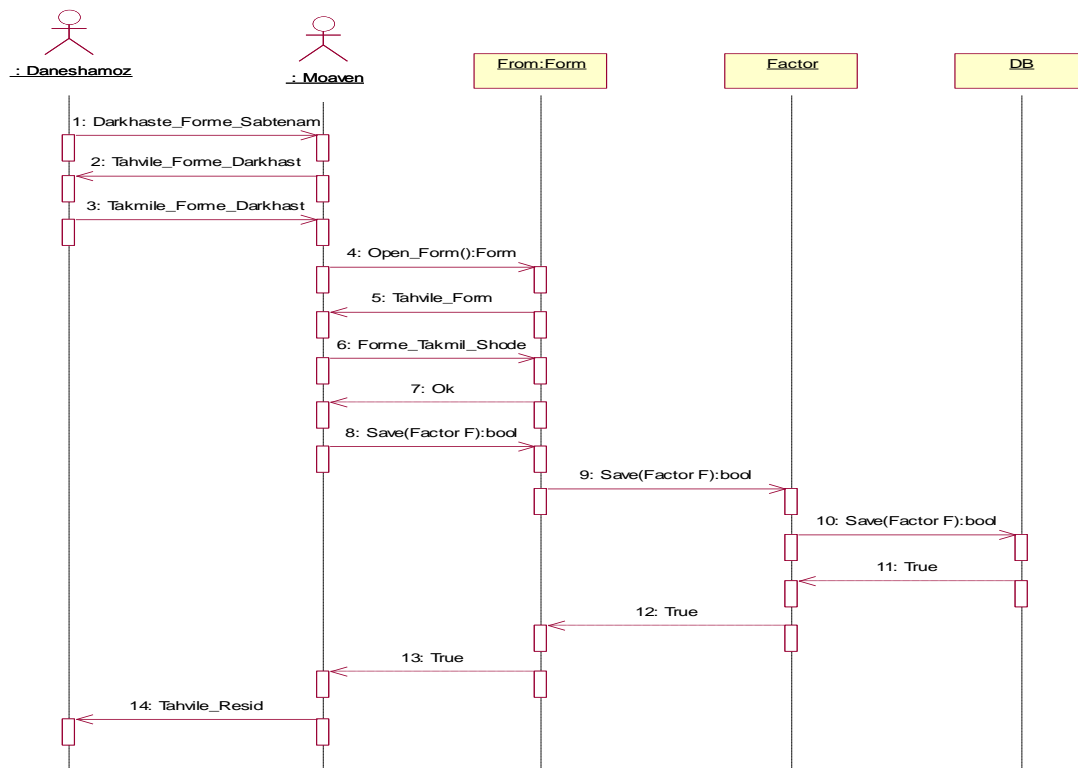
همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد ترم، نام ترم، نوع ترم و توضیحات است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک ترم می باشد.



همانطور که از شکل بالا مشاهده می گردد، این کلاس دارای ویژگی های کد درس، نوع درس، نام درس و کد ترم است. همچنین این کلاس دارای متدهای اضافه، حذف، ویرایش، جستجو و نمایش نمودن یک درس می باشد.

Sequence Diagram -۴

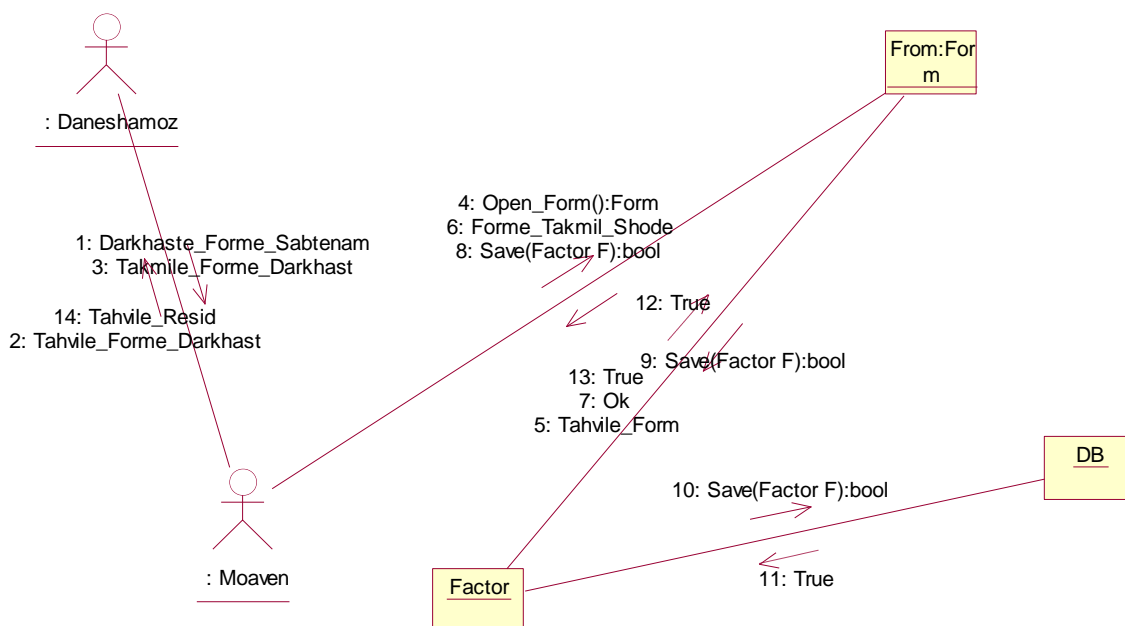
در شکل زیر نمودار ترتیبی درخواست ثبت نام در سیستم مدارس غیرانتفاعی نشان داده شده است.



- نمودار ترتیبی سیستم مدارس غیرانتفاعی
- روال کلی فرآیند نمودار ترتیبی درخواست ثبت نام در سیستم مدارس غیرانتفاعی به شرح ذیل می باشد:
- در مرحله اول، دانش آموز وارد سیستم می شود. سپس دانش آموز درخواست ثبت نام را برای معاون ارسال می کند. معاون، فرم درخواست را به دانش آموز تحویل می دهد. دانش آموز فرم درخواست را تکمیل نموده و به معاون تحویل می دهد. سپس معاون فرم درخواست را باز می کند. سپس معاون فرآیند تکمیل فرم دانش آموز را بر اساس اطلاعات دانش آموز تکمیل می کند. سیستم اطلاعات دانش آموز را بررسی کرده و تاییدیه را برای معاون ارسال می کند. معاون درخواست ذخیره اطلاعات و درخواست ثبت نام را به سیستم ارسال می کند. سیستم به جدول فاکتور اطلاعات درخواستی را ثبت نموده و در نهایت در دیتابیس نهایی ذخیره می شود. سیستم تاییدیه ثبت را به معاون ارسال نموده و در نهایت معاون به دانش آموز رسید نهایی درخواست را تحویل می دهد.

۵- Collaboration Diagram

در شکل زیر نمودار همکاری درخواست ثبت نام در سیستم مدارس غیرانتفاعی نشان داده شده است.



نمودار همکاری سیستم مدارس غیرانتفاعی

روال کلی فرآیند نمودار همکاری درخواست ثبت نام در سیستم مدارس غیرانتفاعی به شرح ذیل می باشد:

در مرحله اول، دانش آموز وارد سیستم می شود. سپس دانش آموز درخواست ثبت نام را برای معاون ارسال می کند. معاون، فرم درخواست را به دانش آموز تحویل می دهد. دانش آموز فرم درخواست را تکمیل نموده و به معاون تحویل می دهد. سپس معاون فرم درخواست را باز می کند. سپس معاون فرآیند تکمیل فرم دانش آموز را بر اساس اطلاعات دانش آموز تکمیل می کند. سیستم اطلاعات دانش آموز را بررسی کرده و تاییدیه را برای معاون ارسال می کند. معاون درخواست ذخیره اطلاعات و درخواست ثبت نام را به سیستم ارسال می کند. سیستم به جدول فاکتور اطلاعات درخواستی را ثبت نموده و در نهایت در دیتابیس نهایی ذخیره می شود. سیستم تاییدیه ثبت را به معاون ارسال نموده و در نهایت معاون به دانش آموز رسید نهایی درخواست را تحویل می دهد.