

۱. تعریف زبان مدل سازی یکپارچه (UML) یک زبان مدل سازی بصری استاندارد است که برای تجسم، ساخت، مستند سازی و مشخص کردن محصولات یک سیستم نرم افزاری استفاده می شود.

۲. فهرست نمودارها: یک نمودار UML یک نمایش بصری از یک یا چند جنبه از یک سیستم نرم افزاری است. هر نمودار یک منظر (view) خاصی از سیستم را نشان دهد و شامل عناصری مانند کلاس ها، اشیاء، رابط ها، فعالیت ها و روابط بین آن ها می شود. نمودارهای UML به دو دسته

اصلی تقسیم می شوند: ۱- نمودارهای ساختاری (Structure Diagrams): این نمودارها ساختار ایستا (static) سیستم را نشان می دهند. آنها اجزای سیستم و روابط بین آنها را به تصویر می کشند.

۲- نمودارهای رفتاری (Behavior Diagrams): این نمودارها رفتار پویا (dynamic) سیستم را نشان می دهند. آنها نحوه تعامل اجزا در طول زمان و نحوه تغییر حالت سیستم را به تصویر می کشند. ۳- کاربردها: از کاربردهای UML می توان به تحلیل و طراحی سیستم، ارتباط تیم، مدل سازی رفتار، مستند سازی و تعیین رفتار سیستم اشاره کرد.

۴- مزایا: استاندارد بودن، شفافیت، انعطاف پذیری، پشتیبانی از ابزارها مختلف از جمله مزایای UML هستند. همچنین می تواند باعث ایجاد زبان مشترک بین توسعه دهندگان، تحلیل گران و دانی نظام شود.

۵- معایب: از معایب UML می توان به پیچیدگی آن، هزینه زمانی و مالی و نیاز به تخصص اشاره کرد. همچنین برای پروژه های کوچک ممکن است بیش از حد سبک باشد.

سیرتین مستطوری

۶- جنبه های ایستا و پویا : جنبه ایستا (Static Aspect) : به ساختار سیستم در یک زمان خاص اشاره دارد. تمرکز آن بر روی عناصر ثابت یک سیستم است مانند کلاس ها، اشیاء و روابط بین آن جنبه پویا (Dynamic Aspect) : به رفتار سیستم در طول زمان اشاره دارد. تمرکز آن بر تعاملات، تغییر وضعیت و ترتیب رویدادها است.

۱- تعریف نمودارهای زبان UML (Use case Diagrams): تعریف این نمودار مورد استفاده نمودار تعاملات بین کاربران (بازیگران) و سیستم را نمایش می دهد و به شناسایی نیازمندی های سیستم کمک می کند. مثالی در یک سیستم فروشگاه آنلاین، بازیگران می توانند «پرداخت» و «بازگشت» داشته باشند. نمودار استفاده در مثال «پرداخت» و «بازگشت» می تواند به سبب خرید «پرداخت» داشته باشند.

(۲) - نمودار فعالیت (Activity Diagram): تعریف نموداری برای نمایش جریان کارها و فرایندها در سیستم، مشابه با نمودار جریان کار (DFD).
مثال: در فرآیند خرید آنلاین فعالیت‌ها شامل «انتخاب محصول»، «اضافه کردن به سبد» و «تایید سفارش» هستند.

نمودار حالت (State Diagram) : تعریف : نموداری برای نمایش وضعیت های مختلف یک شیء و انتقال بین آن ها در طول زمان. مثال : در یک سیستم چاپگر، وضعیت های می تواند «در آماده» ، «در حال چاپ» ، «در انتظار» و «در حال پرینت» باشند.

نمودار ارتباط (Communication Diagram) : تعریف : نمودارک
برای نمایش تعاملات بین اشیاء « مشتری » ، « سیستم رازور » و
« پردازشگر پرداخت » با یکدیگر تعامل دارند

⑤ نمودار توالی (Sequence Diagram) تعریف نموداری برای نمایش ترتیب زمانی تعاملات بین اشیاء در یک سناریو خاص مثال: در فرایند ورود به سیستم، اشیاء (کاربر) «صفحه ورود» و «سیستم احراز هویت» به ترتیب تعامل دارند

⑥ نمودار زمان بندی (Timing Diagram) تعریف: نموداری برای نمایش تغییرات وضعیت اشیاء در طول زمان به ویژه برای سیستم های بلادرنگ مثال: در یک سیستم کنترل دما، وضعیت «دگرمایش» و «دسرمایش» و «خاموش» در طول زمان نمایش داده می شود

⑦ نمودار نمایشی تعامل (Interaction Overview Diagram) تعریف: نموداری که ترکیبی از نمودارهای فعالیت و نمودارهای تعامل را برای نمایش جریان های کنترل پیچیده ارائه می دهد مثال: در یک سیستم پردازش سفارش، مراحل مختلف مانند «در بررسی موجودی»، «در پرداخت» و «در ارسال» به صورت نمودارهای تعامل جداگانه نمایش داده می شوند

⑧ نمودار پروفایل (Profile Diagram) تعریف: نموداری برای تعریف پروفایل های سفارشی که برای گسترش UML با ویژگی های خاص استفاده می شود مثال: در توسعه سیستم های موبایل، پروفایل هایی برای تعریف ویژگی های خاص دستگاه های موبایل استفاده می شود

⑨ نمودار ساختار ترکیبی (Composite Structure Diagram) تعریف: نموداری برای نمایش ساختار داخلی کلاس ها و نحوه تعامل اجزای آن در زمان اجرا مثال: در یک سیستم مدیریت پروژه، کلاس «دپویو» می تواند شامل اجزای «دو ضایف» و «دو متا» و «دو زمان بفری» می باشد

⑩ نمودار کلاس (Class Diagram) تعریف: نموداری برای نمایش ساختار ایستا سیستم، شامل کلاس ها و ویژگی ها و روابط بین کلاس ها مثال: در یک سیستم بانکی، کلاس های «حساب» و «مستر کلا» و «دو ترکن» با ویژگی ها و روابط مشخص نمایش داده می شوند

نمودار بسته (Package Diagram): تعریف نموداری برای نمایش گروه بندی کلاس ها و سایر عناصر مدل به بسته ها و روابط بین آنها. مثال: در یک سیستم مدیریت محتوای «گزارش گیری» و «مدیریت کاربران».

نمودار شیء (Object Diagram): تعریف نموداری برای نمایش نمونه های خاص از کلاس ها و روابط بین آنها در یک لحظه خاص. مثال: در یک سیستم رزرو هتل، نمونه هایی از کلاس «اتاق» یا ویژگی های خاص مانند «شماره اتاق» و «وضعیت» نمایش داده می شوند.

نمودار اجزاء (Component Diagram): تعریف نموداری برای نمایش اجزای نرم افزاری و نحوه اتصال آن ها به یکدیگر. مثال: در یک سیستم مدیریت کتابخانه، اجزایی مانند «دایالگ داده» یا «رابط کاربری» و «پردازشگر درخواست ها» وجود دارند.

نمودار استقرار (Deployment Diagram): تعریف نموداری برای نمایش توزیع فیزیکی اجزای نرم افزاری بر روی سخت افزار و ارتباطات آنها. مثال: در یک سیستم توزیع خبره، سرورهای مختلف و ارتباطات شبکه ای بین آن ها نمایش داده می شوند.

بخش دوم ASP.NET Core

ASP: فناوری قدیمی مایکروسافت برای ساخت صفحات وب یو.آر.آی که با استفاده از اسکریپت نویسی سمت سرور (مثل VBScript) توسعه داده می شد. پایه گذار فعلی بعدی مائیکروسافت ASP.NET بود.

ASP.NET: چارچوب توسعه نرم افزار مایکروسافت که امکان ساخت برنامه های مختلف (وب، اسکریپت و...) را فراهم می کند. شامل کتابخانه ها و محیط اجرایی (CLR) است.

ASP.NET Core: نسخه مدرن و متن باز ASP.NET است که بر پایه .NET Core ساخته شده و برای ساخت برنامه های وب، API و آیلیکیشن های کراس پلتفرم استفاده می شود.

Middleware: اجزای کوچکی از نرم افزار که در زنجیره پردازش درخواست قرار می گیرند. هر Middleware می تواند در خواستی را دریافت، پردازش و به Middleware بعدی ارسال کند. مثل اجزای هویت، لاگ گیری، کش و... (Web Routing)

Web Routing: مکانیزمی برای مسیریابی درخواست های HTTP به کنترلر یا Endpoint خاص در ASP.NET Core. با تعریف الگوهای URL درخواست ها به متد مناسب هدایت می شوند.

سید مبین مسعودی

⑥ Data Annotation: ویژگی‌های (Attributes) برای تزئین مدل‌ها که اطلاعاتی در مورد اعتبار سنجی، نمایش یا رفتار دارها به فریم‌ورک می‌دهند (مثلاً [Required]، [StringLength(100)] و [100])

⑦ DbContext: کلاس پایه در Entity Framework که نمایانگر ارتباط پایگاه داده است و شامل مجموعه‌ای از DbSet ها برای کار با داده‌ها می‌باشد. ① Connection String: رشته‌ی متنی حاوی اطلاعات

اتصال به پایگاه داده (مانند نام سرور، نام پایگاه داده، کاربر و رمز عبور) که در appSettings یا فایل پیکربندی ذخیره می‌شود.

⑨ Dependency Injection: الگویی برای مدیریت وابستگی در ASP.NET Core به صورت پیش فرض از DI پشتیبانی می‌کند و وابستگی

با از طریق سازنده (Constructor Injection) فراهم می‌کند. ⑩ Controller: کلاسی در معماری MVC یا API که وظیفه پردازش درخواست‌ها، اجرای منطق و برگرداندن پاسخ به کاربر را دارد. کنترلرها معمولاً شامل متدهایی به نام "Action" هستند.

ORM (Object-Relational Mapping): روشی برای کار با پایگاه داده از طریق (PPTing) اشیاء برنامه نویسی به جای SQL. مستقیم ORM یا خازنه می‌دهد داده‌ها به صورت شیء مدل‌سازی شوند و تعامل دیتابیس ساده شود.

⑪ Entity Framework: امکان ارتباط ساده با .NET، مایکروسافت برای ORMWORKER را فراهم Migration پایگاه داده، انجام کوئری و مدیریت می‌کند. EF core نسخه سبک‌تر و مدرن‌تر آن است. ⑫ Razer: زبان

قالب نویسی برای ساخت صفحات وب در ASP.NET. ترکیب HTML و کدهای # است که با @ شروع می‌شوند. ⑬ Validation: فرایند بررسی درستی داده‌های ورودی کاربر. در ASP.NET از طریق Fluent،

Data Annotation یا کدهای سفارشی انجام می‌شود. Validation