

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

SKOMPLEITZOVANÝ NÁVRH
E-aukcie

Obsah

ÚVOD	3
Predmet dokumentu	3
DIAGRAMY	3
ENTITNO RELAČNÝ DIAGRAM	3
USE-CASE DIAGRAM	4
STAVOVÝ DIAGRAM	5
SEKVENČNÝ DIAGRAM	6
POUŽÍVAĽSKÉ ROZHRANIE	7
ANALÝZA TECHNOLOGIÍ	7
DÁTOVÝ MODEL	8
DEKOMPOZÍCIA	9
DIAGRAM TRIED	9

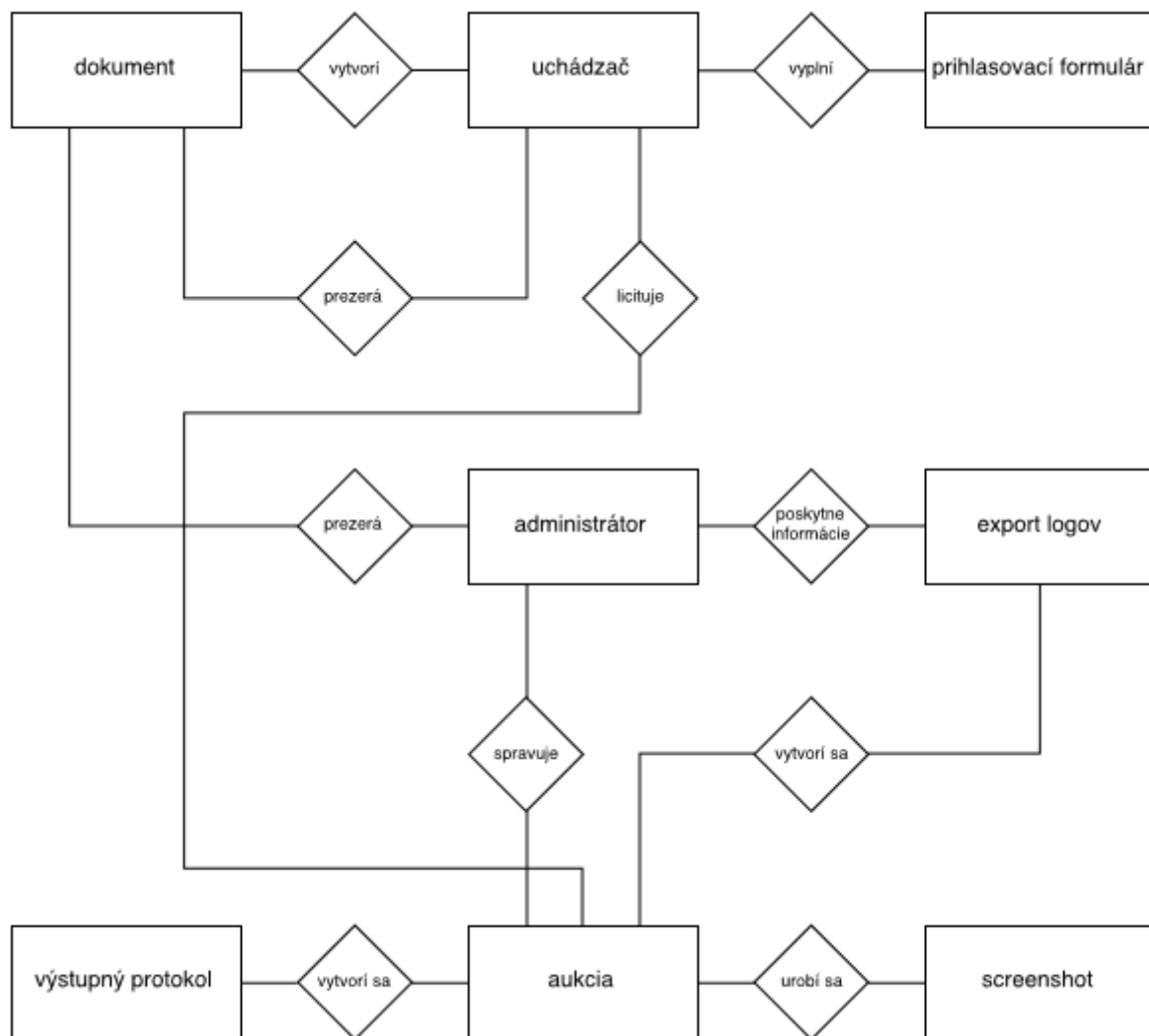
ÚVOD

Predmet dokumentu

Tento dokument slúži na kompletný popis štruktúry projektu E-aukcie. Implikuje úplné diagramy. V dokumente sa bude jednať o diagramy entitno relačný diagram, use-case diagram, stavový diagram a sekvenčné diagramy. Ďalej implikuje zadávateľom odsúhlasené používateľské rozhranie, analýza technológií, dátový model, dekompozícia, triedny diagram.

DIAGRAMY

ENTITNO RELAČNÝ DIAGRAM



Entitno-relačný diagram nepatrí do jazyka UML, lebo ten sa zameriava predovšetkým na objektový návrh, zatiaľ čo ERD je všeobecnejší. Napriek tomu je ERD veľmi bežný a často využívaný. Znázorňuje vzťahy (relácie) medzi entitami. Entita je niečo, čo sa dá nazvať podstatným menom. Diagram sa typicky používa na modelovanie priestoru domény pre ktorú sa informačný systém vyvíja (napr. mliekárenský podnik, ak vyvíjame informačný systém pre výrobu mliečnych výrobkov). Entity sa zakresľujú do obdĺžnikov. Vzťahy medzi

entitami (relácie) sú v kosoštvorcoch a sú prepojené so všetkými entitami, ktoré do daného vzťahu vstupujú. Entity aj relácie môžu mať svoje atribúty, ktoré sa do diagramu môžu zakresliť ako ovály spojené so svojou entitou/reláciou úsečkou. Medzi entitami môže byť vzťah generalizácie/špecializácie vyjadrený trojuholníkom. Vzťah môže byť pomenovaný. Vzťahy medzi entitami môžu mať vyjadrenú násobnosť (koľkokrát sa daná entita daného vzťahu zúčastňuje).

USE-CASE DIAGRAM

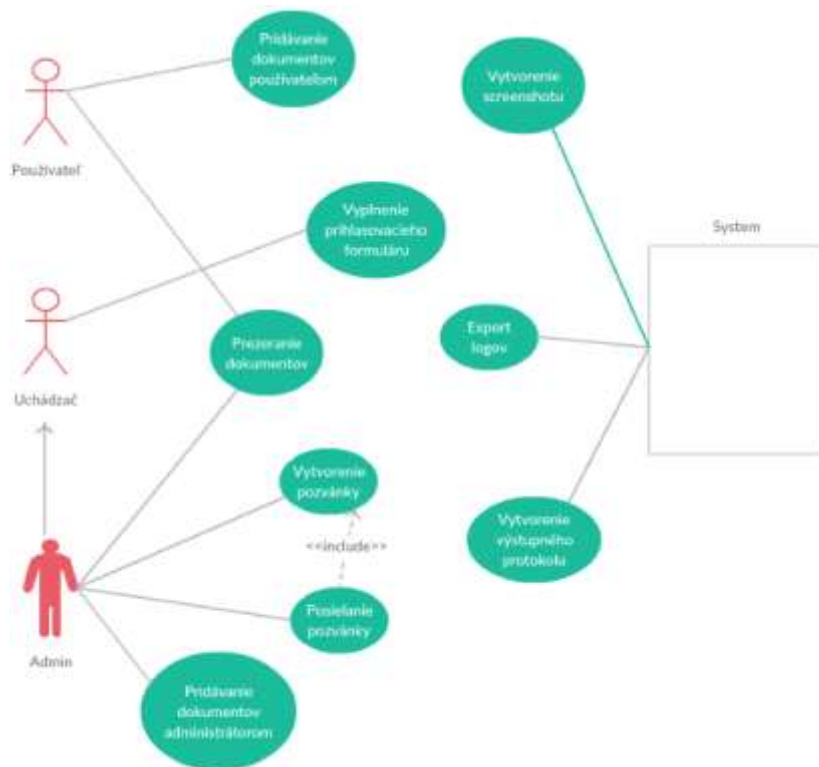


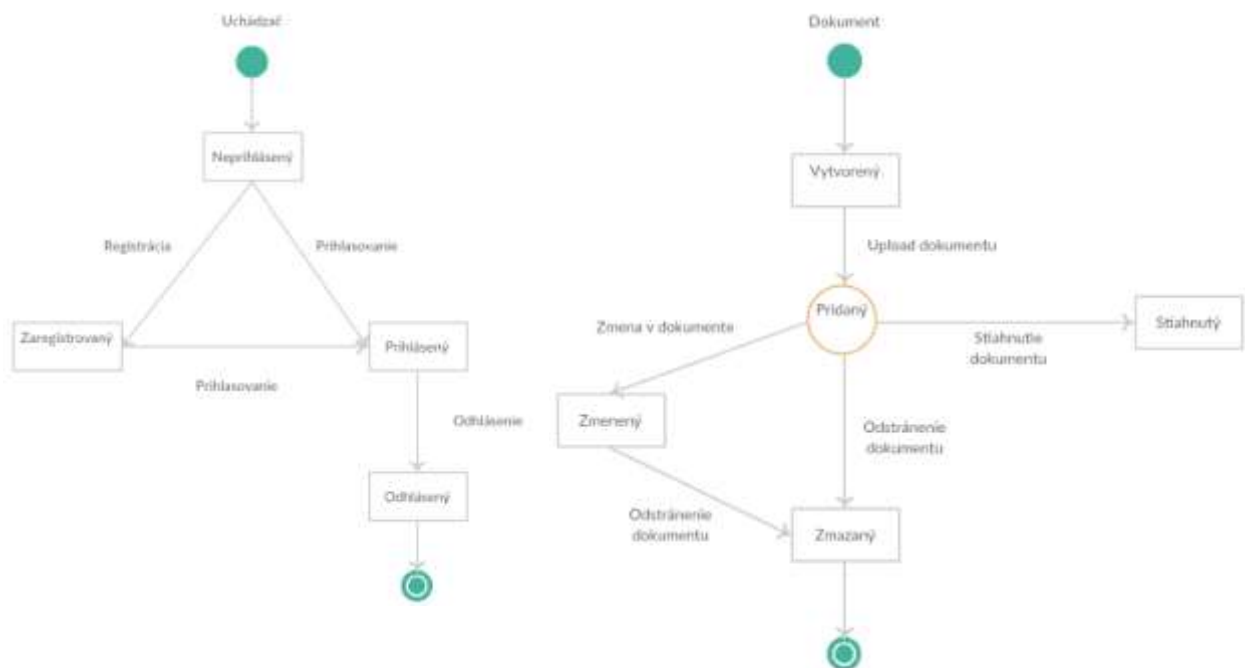
Diagram modeluje činnosti, ktoré informačný systém pri interakcii s používateľmi poskytuje.

Diagram sa používa najmä v skorších fázach vývoja, pri špecifikácii a analýze. Slúži na pomenovanie základných hrubých používateľských scenárov a rozličných rolí, v ktorých používatelia vystupujú pri interakcii so systémom. Definuje ktoré roly sa zúčastňujú ktorých scenárov.

Jednotlivé činnosti (scenáre, prípady použitia) sú zakreslené ako ovály, sú vyjadrené slovne v nedokonavom tvare. Nemali by to byť jednorázové akcie, ale nejaké postupy, ktoré sa skladajú z viacerých krokov. Používateľské roly sa nazývajú aktori a sú zakreslené ako schématické postavičky. Jedna fyzická osoba môže vystupovať aj v rozličných roliach. Aktorom nemusí byť živá bytosť, môže to byť aj fyzická entita, ktorá v systéme hrá nejakú aktívnu rolu. Od aktorov vedú úsečky k oválom, ktoré reprezentujú scenáre, ktorých sa daný aktor zúčastňuje. V prípade potreby je možné scenáre rozkresliť podrobnejšie: pomocou

relácie extends alebo pomocou relácie uses (niekedy označovanej includes). Extends znamená, že príslušný nadscenár môže zahŕňať príslušnú špecializáciu, pričom spravidla sa prejaví len jedna z možných špecializácií. Relácia uses/includes znamená, že príslušný nadscenár vždy zahŕňa aj všetky zobrazené prepojené podscenáre. V prípade aktorov môžeme využiť reláciu generalizácie (dedičnosť) na vyjadrenie vzťahu medzi dvomi aktormi - všeobecnejším a jeho konkrétnejšou špecializáciou. Znamená to, že konkrétnejší aktor sa môže zúčastniť všetkých scenárov ako aktor všeobecnejší, ale zúčastňuje sa nejakých špecifických scenárov navyše.

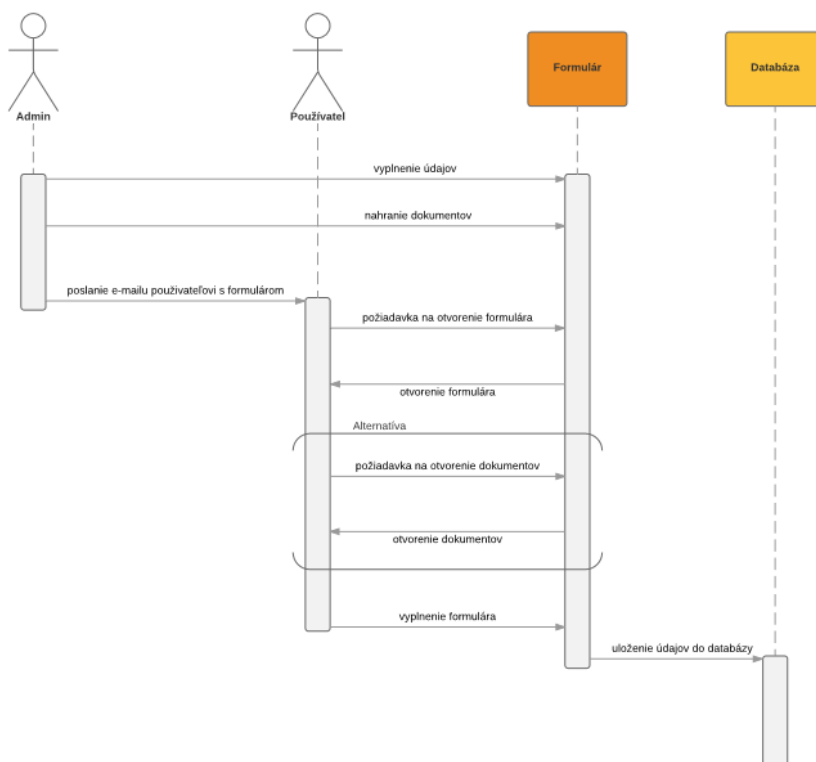
STAVOVÝ DIAGRAM



Stavový diagram je veľmi špecifický, ale dôležitý nástroj modelovania správania sa nejakej entity. Každý stavový diagram musí mať jasne určenú entitu, ktorej stavy zobrazuje. Stav entity je pasívny, nie je to akcia. Stav trvá nejakú dobu, entita v ňom zotrúva po určitú dobu, kým nenastane nejaká udalosť, alebo sa nevykoná nejaká akcia. Následkom udalosti/akcie sa entita dostáva do ďalšieho stavu. Niektorý zo stavov je počiatočný, niektoré stavy môžu byť označené ako koncové. Stavy sa zakresľujú ako ovály a orientované šípky znázorňujú dvojice stavov, medzi ktorými existuje prechod. Stavové prechody sú popísané udalosťami/akciami, ktoré vedú ku stavovému prechodu. Nasledovaním stavových prechodov môže dochádzať k cyklu. Ľubovoľný stav diagramu môže byť prepojený prechodom s ľubovoľným iným. Treba si uvedomiť rozdiel medzi stavovým diagramom a diagramom činností. Zatiaľ čo stavový diagram zobrazuje pasívne stavy, diagram činností (activity diagram, niekedy nazývaný aj flow-chart, po slovensky vývojový diagram) zobrazuje postupnosť akcií, ktoré spolu tvoria nejaký postup/algoritmus. Zameriavajú sa teda na celkom odlišný uhol pohľadu na určitý proces. Niektoré stavy môžu mať vnútornú štruktúru rozobratú na podstavy: takýto stav má vnútri svoj iniciálny stav a stavové prechody

vychádzajúce von automaticky opúšťajú príslušný nadstav. Stavové prechody nadstavu sa môžu realizovať z ľubovoľného podstavu.

SEKVENČNÝ DIAGRAM



Sekvenčný diagram sa spravidla týka jedného konkrétneho scenára. Môže byť využitý pri podrobnom analyzovaní používateľského scenára zachyteného v use-case diagrame na hrubo-zrnnej úrovni, alebo na vyšpecifikovanie následnosti komunikácie objektov (čomu väčšinou zodpovedá volanie metód príslušných objektov) na podrobnej a nízkej úrovni. Sekvenčný diagram sa zvlášť hodí na zakreslenie nejakého komunikačného protokolu.

Diagram sa zakresľuje ako skupina zvislých prerušovaných čiar, pričom každá z nich zodpovedá jednej entite a v hornej časti diagramu je popísaná. Zvislé čiary zodpovedajú plynutiu času - ten plynie v celom diagrame synchronizovane a rovnako, zhora smerom nadol. Scenár začína tak, že jedna entita vyšle do druhej entity správu, znázornenú

vodorovnou šípkou. To spôsobí vytvorenie kontextu spracovania správy na danej cieľovej entite - je zobrazený úzkym zvislým obdĺžnikom a trvá dovtedy, kým požiadavka nie je celá spracovaná. Entita typicky oslovuje ostatné entity v diagrame zaslaním ďalších správ, ktoré nasledujú neskôr v čase (čiže nižšie). Tie opäť vedú k vytvoreniu kontextu na spracovanie danej správy, znovu oslovujú ďalšie entity atď. Správy sú vždy konkretizované textom umiestneným nad šípkou. Po spracovaní požiadavky entita odpovedá návratovou správou (bodkovaná čiara) a presne v tom okamihu jej kontext zaniká. Na jednej entite môže vzniknúť aj nový kontext, pokiaľ predchádzajúci kontext trvá. Jazyk UML dovoľuje v sekvenčnom diagrame špecifikovať časti, ktoré sa nevykonajú vždy (optional, znázornené rámečkom s označením opt v rohu), alebo viaceré alternatívne časti, ktoré sa vykonajú podmienene, v závislosti od splnenia stanoveného predikátu (alt). Podobne je možné zakresliť cykly (loop), prípadne súčasne vykonávané aktivity (par).

POUŽÍVAĽSKÉ ROZHRAŇIE

The screenshot shows a web application interface for an e-auction system. It features a light blue header with navigation icons and a main content area with a light green border. The form is organized into two columns of input fields. The left column includes fields for 'Názov e-aukcie', 'Vyhlasovateľ', 'Prevádzkovateľ e-aukcií', 'Predmet obstarávania', 'Meno uchádzača', and 'Kontaktná osoba'. The right column includes fields for 'Termín doručenia podkladov', 'Termín odovzdania cenovej ponuky', 'Termín e-aukcie', 'Požadované podklady', and 'Kritériá, priebeh aukcie'. Below these fields are three buttons: 'nahrať dokumenty' and 'Stiahnuť inf. dokumenty' in blue, and 'Odoslať' in a larger blue button at the bottom left.

ANALÝZA TECHNOLOGIÍ

Java Itext:

Itext je opensource knižnica na vytváranie a manipulovanie s PDF súbormi v Jave.

Pomocou tejto knižnice budeme vytvárať výstupné PDF dokumenty.

HTML:

Je značkový jazyk určený na vytváranie webových stránok a iných informácií zobraziteľných vo webovom prehliadači.

HTML využijeme na základnú štruktúru formulára.

CSS:

Je jednoduchý mechanizmus na vizuálne formátovanie internetových dokumentov.

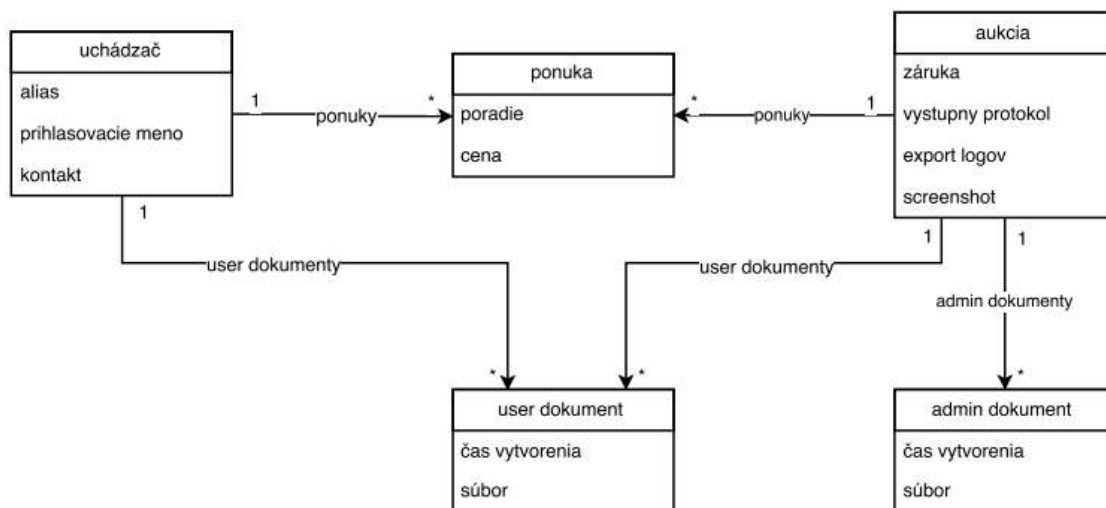
Využijeme na dizajn formulára.

SQL:

Je relačná databáza bez servera.

Využijeme ju na uloženie dát.

DÁTOVÝ MODEL



DEKOMPOZÍCIA

Screenshot modul nekomunikuje s žiadnym ďalším modulom.

Modul na výstupný protokol bude komunikovať s SQL databázou.

Modul na export log komunikuje s SQL databázou

Formulár komunikuje s SQL databázou.

DIAGRAM TRIED

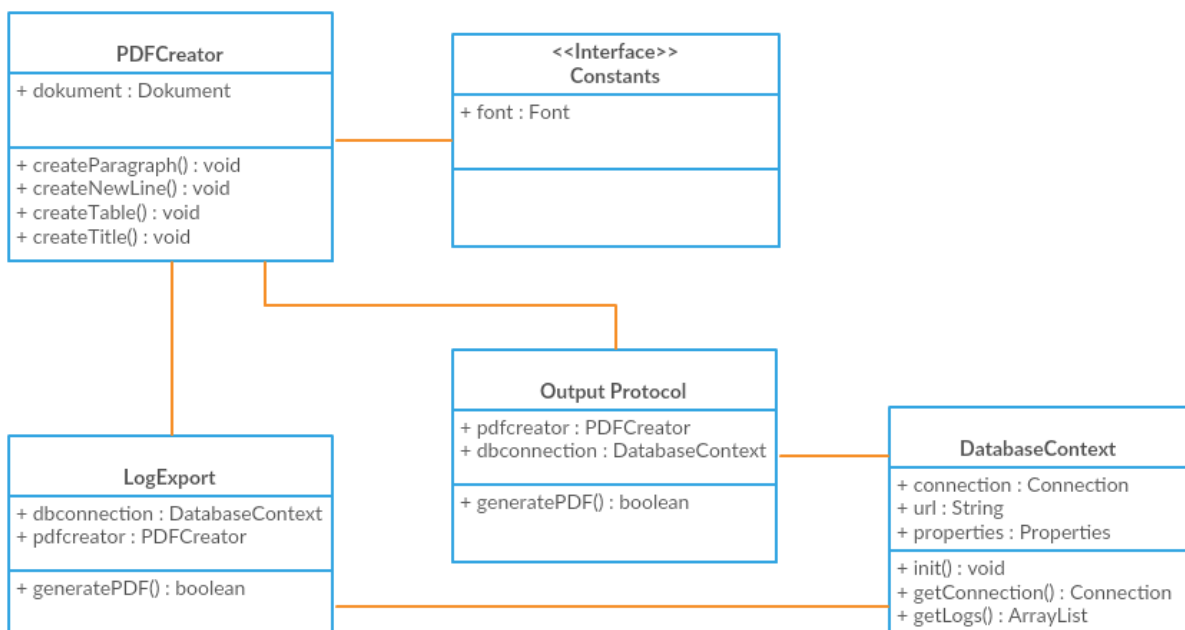


Diagram tried (Class diagram) je jeden z najpoužívanějších UML diagramov. Znázorňuje triedy použité v systéme, alebo jeho časti, prípadne ich metódy a polia a vzťahy medzi triedami. Medzi triedami môžu byť tri základné vzťahy: generalizácia, agregácia a asociácia. Triedy, medzi ktorými je vzťah sú prepojené úsečkou. Prvý z nich vyjadruje vzťah dedičnosti - jedna trieda je špecializáciou inej - dedí z nej všetky vlastnosti a pridáva nejaké svoje špecifické črty. Generalizácia sa znázorňuje trojuholníčkom na strane všeobecnejšej triedy. Agregácia označuje vzťah celok-časť. Na strane celku sa zakresľuje kosoštvorec. V prípade, že je prázdny, ide o bežnú agregáciu, ak je plný, ide o kompozíciu. Kompozícia je silnejšia agregácia, kde celok a časť nemôžu samostatne existovať a dávajú zmysel len ako celok. Asociácia je všeobecný vzťah medzi dvoma triedami, nevyjadruje sa žiadnym symbolom a stanovuje, že jedna trieda nejakým spôsobom využíva služby druhej. Špecifickým typom asociácie je dependencia, ktorá sa definuje takto: medzi triedou A a triedou B je dependencia, ak zmena funkcionality v triede B môže vynútiť úpravy v triede A.