# Irodalom

UML Distilled by Martin Fowler

Applying UML and Patterns by Craig Larman

# Típusok

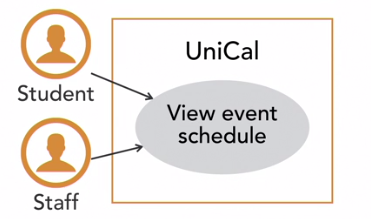
* Struktúra modellezés
  + Class diagram
  + Component diagram
  + Object diagram
  + Composite structure diagram
  + Package diagram
  + Deployment diagram
* Viselkedés
  + Use case diagram
  + Activity diagram
  + State machine diagram
  + Interaction
    - Sequence diagram
    - Communication diagram
    - Timing diagram
    - Interaction overview diagram

Nem kell mindent is megrajzolni.

# Use case diagram

Négy fő részből áll:

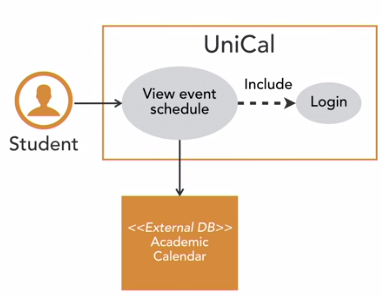
* szerep. Egy adott felhasználónak több szerepe is lehet
* rendszer
* use case, ami a rendszerben található
* kapcsolatok



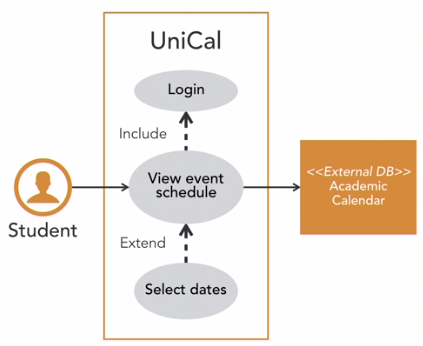
A use case-t az elsődleges felhasználó indítja, de legalábbis az ő érdekében fut le. A másodlagos felhasználó (akit jobb oldalra rajzolunk) csak résztvevő, közreműködő.

Egy kapcsolat definiálható felhasználó és use case, de két use case között is.

Nincs kapcsolat a felhasználó és a bejelentkezés között. Azért van így, mert a use case értéket képvisel a felhasználó számára. A bejelentkezés inkább egy szükséges rossz.



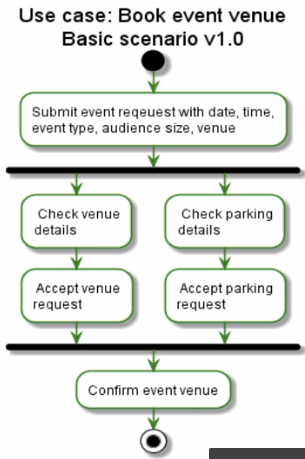
A dátumválasztás önmagában megint csak nem érték. Adott esetben ki sem feltétlenül kell írni, ha csak nem szeretnénk hangsúlyozni, vagy esetleg akkor rajzolható még fel, ha egy későbbi frissítéssel került be a rendszerbe. Ellentétben a bejelentkezéssel, az események megtekintése kvázi független a dátumtól.



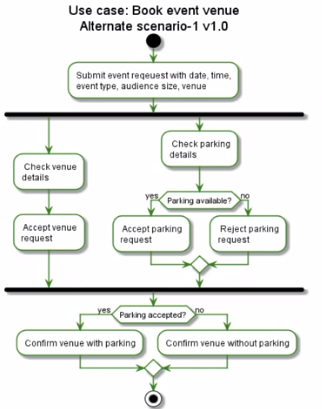
# Activity diagram

Ez arra lesz jó, hogy néhány folyamatot leírjak. Hogyan jön létre egy felhasználói fiók? Mi egy dokumentum életútja?

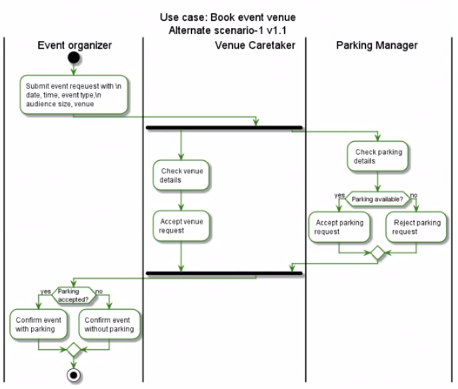
Egy use case kibontására lesz ez jó. Pl. a felhasználó megkezdi az esemény megszervezését. Utána párhuzamosan ellenőrizhető a szállás és a parkolás is. Az én esetemben valószínűleg nem lesz sok párhuzamosítás, de ki tudja.



Ami már előfordulhat nálam is, az a feltételes elágazás. Pl. a felhasználó törli-e a dokumentumot a rendszerből?

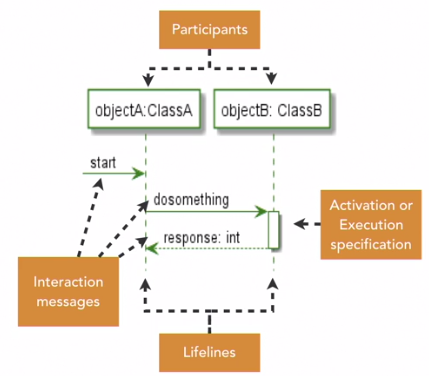


Ha pedig több felhasználó is részt vesz a folyamatban, akkor jöhetnek a medencesávok. Például a dokumentum életútja, hogy azt valaki létrehozza, a felhasználó meg majd elolvassa. (Ezt még át kell gondolni. Jó megoldás lehet egy dokumentum létrehozói szerepkör megalkotása.)



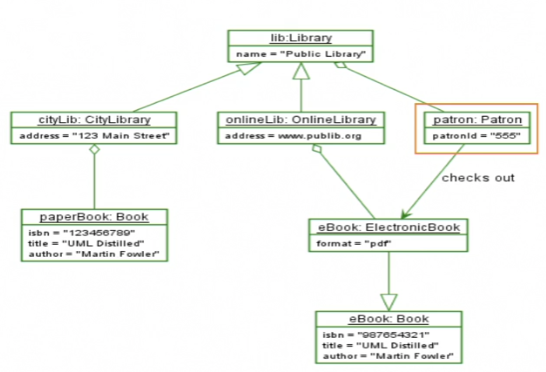
# Sequence diagram

A komponensek (pl. osztályok, de én nem szeretnék OO-t) közötti kommunikáció leírására alkalmas eszköz.



# Object diagram

Ugyanaz, mint az osztálydiagram, de példányokat hivatott bemutatni. Egy aktuális állapotot vázol, így tartalmazza a kapcsolatok aktuális állapotát is. (Pl. egy adott könyv éppen egy adott könyvtártagnál van).



# Webporta

A legelső lépés a felhasználók kategorizálása, az segít megérteni az összes többit.

* Felhasználó
  + Munkavállaló (Magánszemély? Ügyfél? Zery mindkettőt használta.)
    - Aktív munkavállaló
    - Inaktív munkavállaló
  + Adminisztrátor
    - Ügyféltevékenységet monitorozó adminisztrátor
    - Dokumentumok küldésére jogosult adminisztrátor
    - Személyes adatok kezelésével megbízott adminisztrátor

A hierarchián még el lehet gondolkodni, de ez a négy fő szerepkör adódik. Készítünk egy bevezető rövid szöveges leírást mindegyik szerepkörről. A szerepkörök modulárisak, egyetlen regisztráció mögött elérhető mindegyik.

Minden funkcionalitás regisztrációhoz kötött. Az első belépéshez tartozik a regisztráció véglegesítése, amihez rajzolunk egy use case diagramot. A többinél külön már nem rajzoljuk fel, hogy be kell jelentkezni, ez egy alapértelmezett követelmény. Megköveteljük a 90 naponkénti jelszócserét, a jelszó nem egyezhet meg a megelőző három jelszóval.

A munkavállaló:

* listázhatja a saját dokumentumait
* a listát szűrheti és rendezheti (név, dátum, csak el nem olvasott, etc.)
* kiválaszthat egy dokumentumot megtekintésre
* letöltheti a dokumentumhoz tartozó csatolmányt
* letöltés után törölheti a dokumentumot

Ügyféltevékenység monitorozása

* Keresés ügyfélre, dokumentum típusra, rendezési lehetőségek
* Ezeket a szűrőket el tudja menteni, a mentett szűréseket módosíthatja, törölheti
* Látja a dokumentum kézbesítésének időpontját

Személyes adatok kezelése

* Módosíthatja a felhasználó nevét, email címét, illetve szervezeti egységét
* Aktiválhatja az inaktív felhasználót
* Exportálhatja a felhasználók tárolt adatait és a felhasználó rendelkezésére bocsáthatja
* Ténylegesen törölheti a felhasználót

Na, ezekről kellene use case diagramokat rajzolni, és írni hozzájuk némi magyarázatot.