A második beadandó feladatban az első beadandóban beadott dokumentáció készítését kell kibővítenünk egy Tervezés fejezettel.

4.) **Tervezés**

1. **A rendszer architectúrája**  
   Első sorban nagy rendszerek esetén érdemes dokumentálni, hogy milyen alrendszerekből épül fel a szoftverarchitektúránk.  
   Segítségünkre lehetnek az átlátható tervezéshez az implementációs szempont szerinti UML diagramok:
   * **Komponensdiagram és alrendszerdiagram**: [**Megtekintés**](http://www.agilemodeling.com/artifacts/componentDiagram.htm)

Ehhez a fejezethez tartoznak továbbá a moduláris felbontással kapcsolatos tervek.

1. **Adatbázis terv**
   * **Adatbázis sémák leírása**  
     Táblázat jellegűen [Oracle specifikus minta]: [**Megtekintés**](http://valdar.web.elte.hu/progtech2/data/dok2/adatbazis_terv.pdf)
   * **Egyed-kapcsolat diagram / Kapcsolati tábla**:  
     Entity-relationship model: [**Megtekintés**](https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model)  
     Gyakran táblázatosan ábrázoljak (Példa): [**Megtekintés**](http://valdar.web.elte.hu/progtech2/data/dok2/er_table.png)  
     Hasznos eszközök a tervezéshez (Enterprise Architect mellé):
     + Gliffy: [**Megtekintés**](https://www.gliffy.com/products/online/)
     + yEd Graph Editor: [**Megtekintés**](https://www.yworks.com/downloads#yEd)

Megjegyzés: Adatbázisok 1-ből tanuljátok részletesen.

1. **Statikus (Szerkezeti, Struktúrális) szempont szerinti nézetrendszer alapján**  
   Megmondja, hogy a rendszer milyen egységekből épül fel, mi ezeknek az egységeknek a feladata, milyen kapcsolatban vannak egymással a megoldás elérésének az érdekében. Jellemzően két féle diagramot szoktak használni a rendszer ezen szerkezetének leírásához:
   * **Osztálydiagram**: a rendszer objektum elvű szerkezetének leírására  
     Ismétléshez (Progtech 1): [**C-1**](http://valdar.web.elte.hu/progtech2/data/dok2/progtech2_uml_1.pdf) [**C-2**](http://valdar.web.elte.hu/progtech2/data/dok2/progtech2_uml_2.pdf) [**C-3**](http://valdar.web.elte.hu/progtech2/data/dok2/progtech2_uml_3.pdf)  
     További minta feladatok: http://valdar.web.elte.hu/progtech/
   * **Objektumdiagram**: az osztálydiagram egy példányát mutatja be  
     Lásd: Programozási technológia 1. diasor és [**itt**](http://agilemodeling.com/artifacts/objectDiagram.htm).

Megjegyzés: Az ábrákat jól láthatóan kell elkészíteni, több részre bontva. Minden diagram alatt érdemes egy általánosabb leírást adni, hogy mit fog megvalósítani az adott osztály.

1. **Dinamikus szempont szerinti nézetrendszer alapján**  
   Megmondja, hogy a rendszer egyes részegységei hogyan viselkednek a probléma megoldása során. Az egységek milyen állapotokat vesznek fel, milyen események hatására változik az állapotuk, milyen a köztük lévő együttműködés mechanizmusa, időben hogyan zajlanak közöttük az üzenetcserék.
   * **Állapotdiagram**: azt mutatja meg, hogy a rendszer milyen állapotokon keresztül, milyen átmenetekkel oldja meg a feladatot.  
     Minta: [**Megtekintés**](http://www.agilemodeling.com/artifacts/stateMachineDiagram.htm)
   * **Szekvenciadiagram**: az objektumok közötti üzenetváltások időbeli menetét szemlélteti.  
     Minta: [**Megtekintés**](http://agilemodeling.com/artifacts/sequenceDiagram.htm)
   * **Együttműködési diagram**: az objektumoknak a probléma megoldásában való együttműködését mutatja be.  
     Minta: [**Megtekintés**](http://agilemodeling.com/artifacts/communicationDiagram.htm)
   * **Aktivációs diagram**: a tevékenységek és az objektumok egymásra gyakorolt hatását fejezi ki (vezérlések, rendszerfunkciók).  
     Minta: [**Megtekintés**](http://agilemodeling.com/artifacts/activityDiagram.htm)
2. **Környezeti szempont szerinti nézetrendszer alapján**
   * **Konfigurációs diagram**: a szoftverrendszer környezetének, a hardver-szoftver konfigurációnak a szemléltetésére szolgál.

Általánosságban is elmondható, hogy nem csak diagramokat kell készíteni, hanem leírás is szükséges róluk.

Enterprise környezetben több eszközzel is bővül a fentebbi leírás. Például: Business Process Management -> (BPMN ábrák)

A **beadandó feladat** szempontjából kizárólag az Adatbázis terv és a Statikus (Szerkezeti, Struktúrális) szempont szerinti nézetrendszer alapján részekkel kell kiegészíteni az eddig elkészített dokumentációt.  
Objektumdiagram-ot nem szükséges készíteni.

**Fontos, hogy Enterprise Architect-tel kell a diagramokat elkészíteni!**

Az órai minta feladatok az osztálydiagram-os részhez megtalálhatóak a http://valdar.web.elte.hu/progtech/ (3. gyakorlat alatt [ArgoUML-es megoldások])

**Kiegészítő hivatkozások**

UML összefoglaló a nézetrendszer alapján: [**Megtekintés**](http://valdar.web.elte.hu/progtech2/data/dok2/progtech2_uml_0.pdf)

Derby Data types (Adatbázis tervezéshez): [**Megtekintés**](https://db.apache.org/derby/docs/10.7/ref/crefsqlj31068.html)

IEEE 1016 szabvány (SDD): [**Megtekintés**](https://en.wikipedia.org/wiki/Software_design_description)

SDD (Software Design Description) minta: [**Megtekintés**](http://www.san.uri.br/~pbetencourt/engsoftII/IEEE-P1016-d50.PDF)

Tervezési minták: [**Megtekintés**](http://java-design-patterns.com/)

Tervezési minták megvalósítása (GIT): [**Megtekintés**](https://github.com/iluwatar/java-design-patterns)

**További irodalmak**

Jeffrey D. Ullman, Jennifer Widom - Adatbázisrendszerek - Alapvetés - Második, átdolgozott kiadás

Sike Sándor, Varga László - Szoftvertechnológia és UML - Eötvös kiadó

Ian Sommerville - Szoftverrendszerek fejlesztése - Tervezés fejezet (243. oldaltól)