

Tantárgyi tematika és ütemterv  
az **Informatikai Rendszerek Építése** c. tárgyhoz  
Mérnök Informatikus BSc szakos hallgatók számára

*A tárgy előadója / gyakorlatvezetője:*

Dr. Nehéz Károly / Szabó Martin

*Tankör:*

Mérnök Informatikus BSC, III. évf.

*Az előadások helye és ideje:*

órarend szerint

**Tantárgyi követelmények:**

Az aláírás megszerzésének feltétele a féléves feladat elégséges szintű teljesítése. A vizsgaanyag, a tematika azon része, mely az előadásokon elhangzott. A tárgyat szóbeli és írásbeli vizsga zárja.

**Féléves ütemterv**

	<b>Előadás</b>	<b>Gyakorlat</b>
<b>1</b>	Informatikai rendszerek alapfogalmai, osztályozása. Nyílt forráskódú szoftver rendszerek és alkalmazásai	Eclipse J2EE platform bemutatása
<b>2</b>	Kliens-szerver modell és elosztott rendszerek tervezési kérdései.	JBoss alkalmazáskiszolgáló telepítése és konfigurációja
<b>3</b>	A webszolgáltatás fogalma XML-RPC, SOAP, JSON, UDDI, WSDL, E-business lehetőségek, B2C, B2B	szervlet mintapélda
<b>4</b>	OLAP rendszerek alapjai – Adatkocka modell	session bean mintapélda
<b>5</b>	Adatbányászat alapjai, legalapvetőbb technikák	üzenetsor mintapélda
<b>6</b>	Informatikai rendszerek alapmodelljei – a 12 faktor modell	webszolgáltatás mintapélda
<b>7</b>	Modern vékonykliens alkalmazások	JMX, J2EE szolgáltatások mintapélda
<b>8</b>	Vállalti szintű rendszerintegrációs módszerek 1: EAI (Enterprise Application Integration)	Entity bean-ek
<b>9</b>	Vállalti szintű rendszerintegrációs módszerek 2: Üzenetküldő rendszerek (messaging systems)	JSF gyakorlat 1.
<b>10</b>	Alkalmazás kiszolgálók 1. - alapvető funkciók, alapfogalmak, szervlet, enterprise bean-ek.	JSF gyakorlat 2.
<b>11</b>	Alkalmazás kiszolgálók 2. - további alkalmazási lehetőségek, üzenetvezérelt bean-ek	Összetett mintapélda JEE-JPA-JSF alapokon
<b>12</b>	Alkalmazás kiszolgálók 3. - alkalmazási lehetőségek, esettanulmányok	Feladatkiadás, konzultáció
<b>13</b>	Angular esettanulmány bemutatása – ORM backend CRUD lehetőségek	Feladat bemutatás
<b>14</b>	Elővizsga	Pótlás

**Kötelező irodalom:**

1. Raffai Mária: *Információrendszerek fejlesztése és menedzselése*, Novadat kiadó. Budapest, 2003.
2. Imre Gábor: *Szoftverfejlesztés Java EE platformon*, Szak Kiadó, Budapest 2007.

**Ajánlott Irodalom:**

1. C. Edward, J. Ward, Andy Bytheway: *Az Információs rendszerek alapjai*. Panem. 1999.
2. Ross A. Malaga: *Information Systems Technology*, Prentice-Hall, Pearson 2005.
3. M. Lisa Miller: *MIS Cases: Decision Making With Application Software*, Prentice-Hall, Pearson 2005.
4. Robert C. Nickerson: *Business and Information Systems*, Prentice-Hall, Pearson 2001.
5. Eric S. Raymond: *A katedrális és a bazár*, Kiskapu, Budapest 2004.

Dr. Nehéz Károly sk.  
egyetemi docens

Féléves gyakorlati feladatok:

## Általános követelmények - Szoftver Architektúra

A feladatok tipikusan olyan alkalmazások példái, ahol egy cég, vagy szervezet ügyfeleinek és eszközeinek közös nyilvántartását kell megoldani. Az elkészült szoftver JBoss EAP vagy JBoss Wildfly alkalmazás legyen JSF (Primefaces keretrendszer felhasználásával):

- Adatbázis: JDBC alapú relációs adatbázis, lehet MySQL, de lehet a JBoss beépített Hypersonic adatbázis kezelője
- Az alkalmazottnak nem szükséges azonosítani magát az adatbázis elérésénél, és minden alkalmazott ugyan olyan felhasználói jogokkal rendelkezik.

## Eszközök

A fejlesztés során a tantárgy keretei között megismert eszközöket lehet felhasználni:

- programozási nyelv: Java (JDK - <http://java.sun.com/javase/downloads/index.jsp>)
- fejlesztői eszköz: JBoss Developer Studio javasolt
- Primefaces JSF keretrendszer
- A keretrendszer tutorial-szerű használata ebben a könyvben található: <https://www.packtpub.com/web-development/primefaces-cookbook> a könyv "más forrásból" is megszerezhető.

## Feladat leadása

A feladatot a félév végén kell leadni, személyesen bemutatva. Lehet saját laptopon is vagy a labor gépein. Email-ben elküldött megoldásokat nem fogadunk el.

### Határidők

- Az utolsó tanítási héttel bezárólag minden gyakorlaton.
- Levelező oktatás esetén a konzultációkon.

## Feladatok

### 1. Videó kölcsönző nyilvántartó rendszer

#### Általános leírás

Egy elképzelt videó kölcsönző nyilvántartását végző szoftver előállítás a cél. A tékában videó kazettákat és DVD-ket lehet kölcsönözni regisztrált felhasználóknak. Az alkalmazást a téka vezetője használja, hogy regisztrálja, és ellenőrizze az ügyfelek kölcsönzéseit. Az alkalmazásnak nem célja a kölcsönzési és késedelmi díjak kimutatása.

#### Felhasználási esetek

##### Ügyfelek nyilvántartása

A rendszer nyilvántartja a téka kölcsönzőit (ügyfelek). Új ügyfelek felvételénél a rendszer bekéri a nevet, telefonszámot, személyigazolvány számot, lakcímet. A regisztrálás után a rendszer egyedi azonosítóval látja el az ügyfeleket. A felvett ügyfelek listájában lehet keresni név, személyigazolvány szám, azonosító alapján. Az ügyfelek adatait lehet módosítani, vagy az ügyfelet törölni. A törlés csak státuszváltással jár, nem fizikai törléssel.

##### DVD-k, kazetták nyilvántartása

A rendszer nyilvántartja a kazetták és DVD- adatait: cím, beszerzés dátuma, sorszám, státusz (szabad, kikölcsönzött, selejtezett).

##### Kölcsönzés

Amikor egy ügyfél kölcsönözni szeretne szükség van az azonosító adataira, illetve egy vagy több kazetta/DVD azonosítójára (feltételezve, hogy a polcon megtalálta azokat). A rendszerben először a felhasználót kell kikeresni. Az adatlapon meg lehet nézni, hogy van-e már kikölcsönözve neki valami. Utána egyenként lehet hozzárendelni a kikeresett kazettát/DVD-t.

##### DVD visszahozatala

Amikor valaki visszahoz kazettát/DVD-t, akkor azt azonosító alapján lehet felszabadítani. Az ügyintéző a sorszám alapján rákeres a kazettára, és az adatlapján visszaállíthatja szabad státuszúra, így megszűnik a kölcsönzés. Az adatlapnak tartalmaznia kell, hogy mikor kölcsönözték ki a kazettát, és ki volt az, amennyiben kikölcsönzött állapotban van.

#### **Késések lekérdezése**

A rendszer lehetőséget ad egy bizonyos idő után (pl 1 nap – konfiguráció) még mindig kikölcsönzött kazetták/DVD-k lekérdezésére. A listában szerepelnie kell a kazetta és a kölcsönző adatainak is, illetve a kikölcsönzés dátumának, és a számított késésnek is.

## **2. Könyvtári nyilvántartás**

---

### **Általános leírás**

A cél, egy képzeletbeli könyvtári kölcsönző-nyilvántartó szoftver tervezése és implementálása. A könyvtárban könyveket és multimédia anyagokat (cd, kazetta) lehet kölcsönözni, beiratkozott tagoknak. Az alkalmazást a könyvtár vezetője és dolgozói használják, hogy regisztrálják a ki és beiratkozásokat és ellenőrizzék a tagok kölcsönzéseit, listázzák azokat a tagokat, akik nem vitték vissza a kölcsönzött anyagokat időben.

### **Felhasználási esetek**

#### **Tagnyilvántartás**

A rendszer nyilvántartja a könyvtár kölcsönzőit. Új tagok felvételénél a rendszer bekéri a nevet, telefonszámot, személyigazolvány számot, lakcímet. A regisztrálás után, a rendszer egyedi azonosítóval látja el a tagokat. A felvettek listájában lehet keresni név, személyigazolvány szám, azonosító alapján. A tagok adatait lehet módosítani vagy egy létező tagot törölni. A törlés csak státuszváltással jár, nem fizikai törléssel.

#### **Készletnyilvántartás**

A rendszer nyilvántartja a könyvek és multimédia anyagok adatait: pl. típus (cd, könyv, kotta), szerző, cím, beszerzés dátuma, sorszám, státusz (szabad, kikölcsönzött, selejtezett). Itt lehet az új könyveket nyilvántartásba venni. A könyvtár készletét lehet listázni, keresni cím, szerző szerint.

#### **Kölcsönzés**

Amikor egy tag kölcsönözni szeretne, szükség van az azonosító adataira, illetve egy kölcsönzendő könyv, cd azonosítójára. A rendszerben először a felhasználót kell kikeresni. Az adatlapon meg lehet nézni, hogy van-e már kölcsönözve neki valami. A kikölcsönzött tételek száma maximálva van. (pl 6)

#### **Kölcsönzés befejezése**

Amikor valaki visszahoz a könyvtári egységét, akkor azt azonosítója alapján lehet felszabadítani. Az ügyintéző a sorszám alapján rákeres az adott könyvre, cd-re és az adatlapján visszaállíthatja szabad státuszúra, így megszűnik a kölcsönzés. Az adatlapnak tartalmaznia kell, hogy mikor kölcsönözték ki, és ki volt az, amennyiben kikölcsönzött állapotban van.

#### **Késések lekérdezése**

A rendszer lehetőséget ad egy bizonyos idő után (pl 30 nap – konfigurálható) még mindig kikölcsönzött könyvek lekérdezésére. A listában szerepelnie kell a tételeknek és a kölcsönző adatainak is, illetve a kikölcsönzés dátumának, és a számított késésnek is.

## **3. Banki számlavezetés**

---

### **Általános leírás**

A feladat célja, egy leegyszerűsített banki számlavezető rendszer elkészítése. A program felhasználói alapvetően a banki ügyintézők, akik az ügyfelek számláit megnyitják, lezárják, azokon tranzakciókat végeznek. Különböző pénznemeket nem kell kezelnie a rendszernek. Feltételezzük, hogy csak forint számlákat vezet a bank.

### **Felhasználási esetek**

#### **Ügyfelek nyilvántartása**

Az ügyintézőnek lehetősége van új ügyfeleket létrehozni, meglévő ügyfeleit módosítani, esetleg törölni. A létrehozásnál a következő adatokat kell bekérni: név, lakcím, telefonszám, személyigazolvány szám. A rendszer automatikusan generál egy 6 jegyű ügyfélszámot a létrehozott ügyfélhez. Módosítás esetén az ügyintézőnek lehetősége van több szempont alapján keresni az ügyfélre (ügyfélszám, szig. szám, név). Amennyiben több ügyfelet talál a rendszer listából választhat az ügyintéző melyikkel akar dolgozni. Az ügyfél szám kivételével minden

adat módosítható. A törlés nem fizikai törléssel jár, hanem csak státusz módosítással. A töröl státuszú ügyfeleket le lehet kérdezni, de módosítani már a továbbiakban nem lehet.

### **Számlakezelés**

Az aktív ügyfelekhez lehet számla számokat nyitni. A számlaszám nyitáskor a rendszer legenerálja a számlaszámot, ami formátuma: <ügyfélkód>+<4 jegyű sequence>. A 4 jegyű sequencenek egyedinek kell lennie egy ügyfélre nézve. Nyitáskor meg lehet adni a számla kezdő egyenlegét, ami kezdeti befizetesként jelenik meg. A számlaszámoknak egyik jellemzője az egyenlegük, hogy éppen mekkora összeg van rajta. Másik, hogy minden az egyenlegben történt módosításról egy adatbázis bejegyzés születik, amin végig lehet követni a számlát érintő tranzakciókat. Egy tranzakcióról tudnia kell a rendszernek, hogy: melyik számlán történt, mekkora összegben (előjellel), egy tetszőleges leírás, dátum, illetve a másik számla, ahonnan a pénzt, amennyiben volt ilyen. A rendszernek lehetőséget kell adnia, hogy egy ügyfélnek a számlát kilistázza a hozzátartozó egyenlegekkel együtt. A számlákat le lehet zárni (státusz váltás). Ebben az esetben az egyenleg pénztári ki fizetéssel egybekötve lenullázódik. A számlák egyenlege nem mehet negatívba. (Folyószámlahitel nincs.)

### **Átutalások**

Ezen a funkción keresztül az ügyintéző kijelölhet két számlaszámot, amit a banknál vezetnek, és nincs lezárva (forrás illetve cél számlát), és meghatározhatja az átutalás összegét, és egy megjegyzést fűzhet az átutaláshoz. A rendszer az adott összeget azonnal átvezeti, amennyiben a megfelelő egyenleg rendelkezésre áll a célszámlán.

### **Pénztri befizetés**

Ezen a funkción keresztül lehet növelni a számla egyenlegét tetszőleges összeggel.

### **Pénztári kifizetés**

Itt a számlán rendelkezésre álló pénzt lehet kivenni, ezzel csökkentve a rendelkezésre álló egyenlegét. Több pénzt nem lehet kivenni, mint amennyi a számlán van.

### **Jelentések**

Itt lehetőség van számlán történt tranzakciók lekérdezésére. Az ügyintéző szűrhet számlaszámra, összegre, illetve dátum tartományra is.

## **4. Autókölcsonzó nyilvántartás**

### **Általános leírás**

A cél, egy képzeletbeli autókölcsonzók munkáját segítő szoftver tervezése és implementálása. A kölcsönzőben 4 kerekű és vízi járműveket lehet kölcsönözni. Az alkalmazást a kölcsönző vezetője és dolgozói használják a kölcsönzések menedzselésére, a kölcsönzési díj kiszámítására, az esetleges sérülések regisztrálására.

### **Felhasználási esetek**

#### **Ügyfelek nyilvántartása**

Csak az kölcsönözhet, akit a rendszerben nyilvántartásba vettek. Név, cím, igazolványszám, telefon, stb megadása kötelező.

#### **Készletnyilvántartás**

A rendszer nyilvántartja a különböző járművek adatait: pl. típus, gyártó, rendszám, alvázszám, beszerzés dátuma, sorszám, kölcsönzési díj, futott km díj, státusz (szabad, kikölcsönzött, selejtezett). Itt lehet az új járműveket nyilvántartásba venni. A kölcsönző járműveit lehet listázni, keresni típus, rendszám szerint. Meg lehet tekinteni melyik járművet kölcsönözték ki éppen és melyik szabad.

#### **Kölcsönzés**

A kölcsönzéshez szükséges egy ügyfél létrehozása. A kölcsönzéskor egy adott ügyfélhez rendeljük a kölcsönzött jármű azonosítóját. Egy ügyfél több járművet is kölcsönözhet egyszerre. (pl. ha az ügyfél egy cég)

#### **Kölcsönzés befejezése**

A kölcsönvett járművet visszahozzák. A kölcsönzési díj kiszámítása ilyenkor történik. A díj, napi alapon kerül meghatározásra, ehhez hozzáadódik a futott kilométerdíj. Ha az autó sérülten érkezik vissza, akkor a kölcsönzési díj egy előre megadott összeggel nő. (mivel a többi kárt a biztosító fizeti)

## **5. Lakóközösségi könyvelés**

## Általános leírás

A program célja, hogy megkönnyítse a ház befizetéseit nyilvántartó könyvelő munkáját, illetve hogy személyre szabott jelentéseket tudjon produkálni a lakók számára esetleges számla reklamációk esetén.

### Felhasználási esetek

#### Lakások nyilvántartása

A szoftvernek képesnek kell nyilvántartani több lakás adatait: lakás száma (emelet, ajtó), alapterület (négyzetméterben), légtér (légméterben). Alapfeltételezés legyen, hogy egy négyemeletes bérház nyilvántartását könyvelik a programmal.

#### Lakók nyilvántartása

A rendszer a lakók számára számlát vezet, ahol a befizetéseket és költségeket nyilvántartja. A számlát akkor hozza létre, amikor a lakó beköltözik a lakásba. Lakót törölni a nyilvántartásból nem lehet, még ha el is költözik, hisz tartozásait azután is nyilván kell tartani. Lakó létrehozása a „költözés” funkción keresztül lehet. Itt kiválasztva a lakást, megadva az új lakó nevét, és induló egyenlegét (ami átvállalt tartozásból származhat), a lakó bekerül a nyilvántartásba.

#### Egyéni befizetések

A rendszerben lehetőség van egyéni befizetések nyilvántartására. Bármely lakó fizethet be tetszőleges összeget. A program nyilvántartja a befizetések idejét, és összegét, illetve a lakó számláján az egyenlegét (aktuális tartozások a ház felé).

#### Költségek elosztása

A szoftvernek alapvetően két lehetőséget kell biztosítani a költségek nyilvántartására: négyzetméter árak felszorozása: pl közös költség kiszámítása, ahol egy négyzetméterre vonatkoztatott díjat kell a lakók számlájára betehelni. összköltség leosztása: egy nagy összegű díj (pl lépcsőház felújítás) leosztása arányosan az alapterületek szerint. Költséget csak aktív lakóra lehet leosztani. Kiköltözött lakókra csak befizetéseket lehet felvinni. A rendszer nyilvántartja a költség keletkezésének idejét, illetve egy leírást, ami a költség okát jelöli. Ez utóbbit a felhasználó írja be a költség definiálásakor. A rendszernek a költségek véglegesítése előtt mutatnia kell egy listát, hogy felhasználó átnézhesse a kiszámított díjakat, illetve az összes elszámolt díjat abban a tranzakcióban.

#### Jelentések

A rendszer képesnek kell lennie készíteni kimutatást egy adott lakóra, és időszakra vonatkozólag, aminek tartalmaznia kell:

- időszak elején fenn álló tartozások
- időszak alatt befutó tranzakciók leírása, összege, és dátuma
- időszak végén fenn álló tartozás.

Ezenkívül egy másik kimutatás keretében képesnek kell lennie mutatni egy összesítő listát minden lakóra nézve, de csak egy adott időszakon belül (pl összes lakó az elmúlt egy évben):

- lakónként: nyitó egyenleg, befizetések összege, elszámolt költségek összege, záró egyenleg
- illetve egy összesítő sor formájában: a ház nyitó egyenlege az időszak elején, összes befizetés, összes költség, és záró egyenleg.

## 6. Pizza rendelések

### Általános leírás

A cél, egy telefonos pizzarendelő szolgáltatás informatikai hátterének tervezése és megvalósítása. A rendelések telefonon érkeznek. Az 5000 ft-nál (konfigurálható) többet rendelő ügyfelek automatikusan kedvezményt kapnak.

### Felhasználási esetek

#### Rendelés

Csak az rendelhet pizzát, akit a rendszerben nyilvántartásba vettek. Információk: név, cím, telefon, stb megadása. A regisztrált felhasználók rendelhetnek pizzát és egyéb ételeket. A rendelés végén az diszpécser meadja, hogy mennyi lesz a fizetendő összeg és mennyi idő múlva tudják teljesíteni a rendelést. Ezt az alapján számolja ki, hogy minden pizza 20 percig (konfigurálható) sül, és a pizzéria rendelkezik 5 (konfigurálható számú) sütővel. Azaz 5 pizzát lehet egyszerre sütni. A kiszállítás átlagos ideje 20 perc. Így a minimum kiszállítási idő 40 perc.

Ha jelenleg 5 pizza sül és a legelső 5 perc múlva készül el, akkor  $5+20+20 = 45$  percet kell mondani a vevőnek. Ha a vevő sokallja ezt az időt a megrendelést lehet törölni.

### **Étlap menedzselés**

Itt lehet az egyes pizza és étel típusokat bevinni a rendszerbe. Lehetséges attributumok: ár, méret, leírás, sütési/elkészítési idő.

### **Sütőmenedzsment**

Ez a funkció felelős a sütő és a konyha foglaltsági állapotának vizsgálatára: meg lehet tekinteni az egyes sütők állapotát. (süt/nem süt) Egyszerűsítésképpen a konyhában egyszerre 3 étel készíthető. A szakácsokat nem adminisztráljuk.

## **7. Orvosi törzsadat kezelés**

---

### **Általános leírás**

A cél, egy olyan szoftver tervezése és megvalósítása, amely képes segítséget nyújtani egy háziorvosi praxis beteg adminisztrációjához: betegek nyilvántartása, vizitek adminisztrációja, szűrővizsgálatokra való kiértékelés adminisztrációja.

### **Felhasználási esetek**

#### **Betegek nyilvántartása**

A háziorvos beteget regisztrálni kell a programban. A tárolandó információk: név, szül. dátum, taj szám, kórtörténet: felírt gyógyszerek és kezelések listája.

#### **Vizit menedzselése**

A beteg megjelenik az orvosnál a TAJ kártyájával. A program listázza a kórtörténetet. Az orvosnak lehetősége van egy új mezőbe beírni a diagnózist és a felírt gyógyszerek és kezelések listáját. Az orvos, a beteg által hozott leleteket is mentheti a rendszerrel.

#### **Szűrővizsgálatok**

Megadott feltételek alapján a rendszer listát készít azokról a páciensekről, akiknek szűrővizsgálatokon kell résztvenniük. Pl. évente 1-szer tüdőszűrő vizsgálat a 18 évesnél idősebbeknek, nemtől függetlenül. 2 évente prosztata vizsgálat a 35 évesnél idősebb férfiaknak (nem röhög!). 3 évente mammográfiai vizsgálat a 45 évesnél idősebb nőknek. 5 évente általános vizsgálat mindenkinek. A lista alapján az asszisztensek levelet küldenek az érintetteknek (ezt a részt nem kell implementálni).

## **8. Oktatási nyilvántartó rendszer**

---

### **Általános leírás**

Minden hallgató szidta már a NEPTUN rendszer megalkotóit, ezért ebben a feladatban meg kell mutatni, hogy mennyivel jobb rendszer alapjainak megtervezésére és megvalósítására vagyunk képesek. A rendszer alapadatai: oktatók, hallgatók, tantárgyak és ezek kurzusai. A tervezendő rendszert csak adminisztrátorok használhatják az egyes hallgatók és oktatók nem. A hallgatókat és tárgyakat is az adminisztrátorok adják a rendszerbe, és a hallgató-tárgy összerendelést is ők végzik.

### **Felhasználási esetek**

#### **Oktatók/hallgatók menedzselése, listázása**

Oktatók és hallgatók hozzáadása a rendszerhez. Alapinformációk (ezek nem feltétlenül tábla oszlopok): Oktatók: Név, tanszék, oktatott tárgyak. Hallgatók: név, tankör, hallgatott tárgyak, érdemjegyek. A program listázza egy adott oktató: oktatott tárgyak listáját. Egy adott hallgató: felvett tantárgyak listája, érdemjegyekkel.

#### **Új kurzus felvétele**

Az oktatókhoz tantárgyak és a tantárgyakhoz kurzusok rendelhetőek. Természetesen törölhetőek is.

#### **Tárgyak hallgatók összerendelése**

Egy adott hallgató hozzárendelése egy kurzushoz. A rendszer figyelje, hogy nem vette e fel a hallgató az adott tárgyak korábban.

#### **Érdemjegy menedzselés**

Egy adott hallgató érdemjegyet kap egy adott tárgy, adott kurzusán.

#### **Statisztikák kezelése**

Egy adott tankör, tankörátlagának kiszámítása. Egy adott hallgató átlagának kiszámítása.

## 9. Raktárnyílvántartó rendszer

---

### Általános leírás

A cél egy képzeletbeli üzem raktározási problémáinak számítógépes segítése. A cég egy szerelő üzem, amely a raktáron lévő alkatrészekből késztermékeket szerel. A szoftvert a gyártást vezető mérnök kezeli.

#### Felhasználási esetek

##### Raktár nyílvántartás

Raktári egységek felvétele, törlése. Információk: raktári szám, bevételezés ideje, hány darab van a raktárban az adott alkatrészből. Az alkatrészek feltöltését a db-szám növelésével elegendő megoldani.

##### Gyártmány menedzselés

Gyártmányok létrehozása, törlése. Egy gyártmány adott számú raktári egységekből áll. Pl ha a gyártmány egy kisautó, akkor kell hozzá 1 db motor, 1 db test, 4 db kerék, 12 db M6 os csavar stb. Az alkatrészek neveit a rendszer a raktári készletek alapján ajánlja fel. Egy új gyártmány nemcsak alap raktári készletekből állhat hanem más gyártmányokból is.

##### Megrendelés menedzselés

Az üzem megrendelést kap egy adott termékre. A megrendelő alapadatait le kell tárolni. Ha pl 10 db kisautó a megrendelés akkor 10db motor, 40 db kerék és 120 db csavar szükséges a raktáron. Ha a részegységekből nincs elegendő a raktáron, a szoftver jelzi és nem engedi a megrendelés létrehozását. A megrendelése listázására is van lehetőség.

## 10. Üzemirányító rendszer

---

### Általános leírás

A kifejlesztendő alkalmazás alkalmas egy képzeletbeli üzem erőforrásainak menedzselésére. A rendszer alapvető objektumai: munkások, gépek, munkák, feladatok. Egy munka több feladatból áll (legalább 1). Egy feladatot csak 1 gépen lehet elvégezni. A gépek alaptípusai pl: CNC marógép, CNC eszterga, mérőgép. Pl. Az M1 munka három feladatot tartalmaz: marás (a marógépen végezhető feladat), esztergálás (az esztergán végezhető), mérés (a mérőgépen végezhető). A munkák a munkásokhoz rendelhetőek, feltéve ha szabadok éppen.

#### Felhasználási esetek

##### Munkások menedzselése

Munkások felvétele, listázása. Munkáshoz tartozó információ: név, szakképzettség, órabér, státusz (szabadságon, dolgozik, szabad). A szabad munkásokhoz lehet munkákat rendelni.

##### Gépek menedzselése

Új gépek megadása/törlése

##### Munkák menedzselése

Új munka létrehozása. A munkához feladatok rendelése/törlése. A feladatokhoz gépek rendelése/törlése.

##### Listázási funkciók

Szabad munkások, gépek megjelenítése. Egy adott gép foglaltságának %-os kiadása.

# Vizsgatételek

## Informatikai rendszerek építése

1. Mutassa be a tranzakció feldolgozó rendszerek (TPS) tulajdonságait.
2. Mutassa be röviden az informatikai stratégia fejlesztésének főbb lépéseit.
3. Fejtsd ki a „Nyílt forráskódú” szoftverrendszerek jellemzőit és alkalmazásuk előnyeit.
4. Mutassa be a folyamat irányító informatikai rendszerek tulajdonságait (PCS).
5. Milyen szoftverrendszereket jelöl a Groupware és Workflow System kifejezések?
6. Mutassa be részletesen a vezetői munkát támogató informatikai rendszerek tulajdonságait. (EIS, SIS, DSS, stb..)
7. Kliens szerver modellek, osztott rendszerek fontos jellemzői. Osztott rendszerek tervezési szempontjai. Háromrétegű modell. Jellemezze a Vékony kliens, vastag kliens, Java Applet modelleket. (Java webstart)
8. Hogyan mérhető az informatikai rendszerek teljesítménye? Little törvénye. Skálázhatóság definíciója
9. Mit jelöl általában a webszolgáltatások kifejezés? E-kereskedelem, B2B, B2C rendszerek lényege.
10. Mutassa be az XML-RPC lényegét, mutasson be egy kérés és egy válasz XML-t.
11. Mit jelent a fast cgi kifejezés? Milyen előnyei vannak a hagyományos CGI-vel szemben?
12. Jellemezze a forward és reverse proxy-k működési elvét.
13. Mit jelent a fast-cgi koncepció a webszerverek esetén?
14. Mutassa be az úgynevezett 12 faktor elv: kódbázis, függőségek, konfigurációval kapcsolatos elveit a mikro szolgáltatások tekintetében.
15. Mutassa be az úgynevezett 12 faktor elv: kódbázis, függőségek, konfigurációval kapcsolatos elveit a mikro szolgáltatások tekintetében.
16. Mutassa be az úgynevezett 12 faktor elv: szolgáltatások, build-release-run, futó folyamatokkal kapcsolatos elveit a mikro szolgáltatásokkal kapcsolatban.
17. Mutassa be az úgynevezett 12 faktor elv: port binding, konkurencia, eldobhatósággal kapcsolatos elveit a mikro szolgáltatások tekintetében.
18. Mutassa be az úgynevezett 12 faktor elv: dev/prod hasonlóság fenntartása, logok kezelése, admin folyamatokkal kapcsolatos elveit a mikro szolgáltatások tekintetében.
19. Mi az internetes cookie-k lényege? Hogyan működnek? Mutassa be néhány mondattal.
20. Hogyan működik az AJAX, mutassa be lépéseit egy egyszerű példa alapján.
21. Milyen feladatokat lát el egy terheléelosztó egy informatikai rendszerben.
22. Egy informatikai rendszer kialakításánál hogyan válasszunk megfelelő adatbáziskezelő alrendszert? Nyílt forrású – zárt forrású, in process – service, SQL89-SQL99, stb.
23. Mutassa be az OLAP rendszerek legfontosabb tulajdonságait. Mutassa be egy példán keresztül mi a különbség az adatkocka és relációs modell között.
24. Jellemezze a az alábbi integrációs lehetőségeket: fájl átvitel; osztott adatbázis; távoli eljárashívás; üzenetváltás;
25. Jellemezze a következő üzenetcsatorna típusokat: pont-pont csatorna; publikálás – feliratkozás csatorna; rögzített adattípusú csatorna;
26. Jellemezze a következő üzenetcsatorna típusokat: érvénytelen üzenet csatorna; halott levél csatorna; garantáltan kézbesítő csatorna;
27. Mutassa be az adatbányászati módszerek lényegét, alkalmazásainak fontosabb típusait.
28. Mutassa be a következő alapvető adatbányászati fogalmakat: klaszter analízis, döntési fa, regresszió, neurális hálók.

Dr. Nehéz Károly  
tárgyjegyző