

Tantárgyi követelmények

Egy viszonylag nagyobb méretű szoftverfejlesztési feladatot kell megoldani. A projekt 4 részfeladatból áll: elemzés, tervezés, implementáció, tesztelés. A szoftvert modellezni kell. Az elemzés, tervezés során UML diagramokat, képernyőterveket és szöveges leírásokat kell készíteni, CASE eszközzel. Az implementáció Java nyelven, integrált fejlesztőeszközzel történik. A program teszteléséről dokumentációt kell készíteni.

1. részfeladat - Követelményelemzés

Készítendő termékek: használati-eset diagram, a használati-esetek kötetlen formájú szöveges leírása, a használati-esetek forgatókönyveinek strukturált szöveges leírása; képernyőtervek (minden használati-esethez legalább egy képernyőtervet kell készíteni)

Bemutató: Egy *.EAP (Enterprise Architect Project) fájl, mely tartalmazza a használati-eset diagramot, a szöveges leírásokat és a képernyőterveket.

2. részfeladat - Tervezés

Készítendő termékek: egyed-kapcsolat diagram (adatbázis terv); osztálydiagram: A diagramon meg kell adni az osztályok sztereotípiáját (egyed <<entity>>, határ <<boundary>>, konténer <<container>>, vezérlő <<controller>>, egyéb), az öröklési és társítási kapcsolatokat, a társítási kapcsolatok fajtáját (asszociáció, kompozíció, aggregáció) és multiplicitását (1, *), a legfontosabb adatmezőket és metódusokat, ezek láthatóságát (+, -, #). Az osztályok felelősségéről szöveges osztályleírást kell készíteni.

Bemutató: A követelménykezelésnél létrehozott *.EAP (Enterprise Architect Project) fájl kiegészített változata, mely tartalmazza az egyed-kapcsolat diagramot, az osztálydiagramot és a hozzá tartozó szöveges osztályleírásokat.

3. részfeladat - Implementáció

Készítendő termék: Java nyelven megírt forráskód

Bemutató: A NetBeans projekt (Java forrásfájlok). A Javadoc megjegyzésekből generált HTML formátumú fejlesztői dokumentáció (osztályleírások).

4. részfeladat - Tesztelés

Feketedoboz tesztelés. A tesztelés során végre kell hajtani a használati-esetek forgatókönyveit.

Dokumentálni kell, hogy kezdetben milyen adatok vannak az adatbázis-táblákban. Az adatbázis táblákba a kezdeti adatokat SQL scripttel kell felvinni (input adatok). Táblánként minimum 10 rekordot kell rögzíteni.

A lekérdezés, szűrés használati-eseteknél képernyőkép másolatokkal (Alt+Print Screen) dokumentálni kell a megjelenő eredménytáblákat (output adatok).

A felvitel, módosítás, törlés használati-eseteknél dokumentálni kell az adattáblák megváltozott állapotát, ezt a NetBeans adatbázis-adminisztrátor eszközével kell megnézni és ellenőrizni. A NetBeans adatbázis-adminisztrátor képernyőjén megjelenő táblákat kell a dokumentációba másolni (Alt+Print Screen).

A felvitel használati-esetnél tesztelni kell, hogy helyesen ellenőrzi-e a program a felviteli úrlapra beírt adatok formátumára és értéktartományára vonatkozó megszorításokat. Dokumentálni kell a beviteli mezőkbe beírt teszt adatokat (input), a hibás adatbevitel esetén megjelenített hibaüzenetek képernyőkép másolatait (output).

Bemutató: A kezdeti adatokat rögzítő SQL script (*.SQL fájl). A tesztelési dokumentáció PDF formátumban.

Tantárgyi jegy

A 4 részfeladatra 4 részosztályzatot adunk. A 4 részosztályzat átlaga a tantárgyi jegy. Nincs zárthelyi dolgozat.

Beadási határidők

5-6. óra: 1. részfeladat (az 5. órára a névsor első 10 hallgatója jön, a többiek a 6. órára jönnek)

7-8. óra: 2. részfeladat (a 7. órára a névsor első 10 hallgatója jön, a többiek a 8. órára jönnek)

12-13. óra: 3-4. részfeladat (a 12. órára a névsor első 10 hallgatója jön, a többiek a 13. órára jönnek)

A 3-4. részfeladatot egyidőben kell bemutatni, de külön-külön osztályozzuk.

Feladatok

A szoftver nem-funkcionális követelményei

Model-View architektúrát kell alkalmazni.

A programhoz létre kell hozni egy adatbázist, melyet fel kell tölteni tesztadatokkal, táblánként min. 10 rekordot kell felvinni. A feltöltéshez SQL scriptet kell írni.

Felhasználóbarát kezelőfelületet kell kialakítani.

A felvitel során ellenőrizni kell, hogy az adatok megfelelnek-e az adatformátumra és az értéktartományra vonatkozó megszorításoknak. Ahol megoldható, ott biztosítsunk kiválasztási lehetőséget (lista, naptár).

A Java kódot Javadoc megjegyzésekkel kell dokumentálni, és ebből HTML formátumú fejlesztői dokumentációt kell generálni.

A projekthez a következő eszközöket használjuk:

Programozási nyelv: Java Standard Edition JDK 8

IDE: NetBeans 8.2

CASE eszköz: Sparx Systems Enterprise Architect 13.5

<http://www.sparxsystems.com>

(A letölthető próbaverzióban 30 napos időkorlát van. A teljes verzió telepítve van a Lovardában és az összes laborban.)

Relációs adatbázis-kezelő rendszer: Java DB (Apache Derby) (A Glassfish alkalmazás-szerverbe integrált változatot használjuk.)

Adatbázis-kezelő Java API: JDBC

Felhasználói-felület: Swing

Az adatbázisban tárolt adatok

A feladatok szövege leírja, hogy az adatbázisban milyen adatokat kell tárolni. Ez alapján meg kell tervezni az adatbázis logikai szerkezetét, és meg kell tervezni a táblákat. A szöveg nem hangsúlyozza ki a megoldáshoz szükséges összes táblát (nem adja meg a kapcsolótáblákat és nem hangsúlyozza ki az összes törzstáblát sem) és nem adja meg teljes részletességgel a táblák mező szerkezetét.

1. Filmek nyilvántartása (Készítette: Bodó Alex)

Otthoni DVD, VHS filmjeinket szeretnénk nyilvántartani. A filmeket kölcsönadhatjuk a barátainknak, és a kölcsönzéseket is szeretnénk nyilvántartani.

Az adatbázisban tárolt adatok

film: cím, rendező, főszereplő (lehet több főszereplő is, egy főszereplő több filmben is szerepelhet), gyártási év, hossz (percben megadva), adathordozó típusa (DVD/VHS), a DVD/VHS borítóképe, az adathordozó eredeti vagy kalóz másolat, hányszor adtuk kölcsön, most kölcsön van-e adva

kölcsönzés: a kölcsönző személy neve (egy személy több filmet is kölcsönözhet, egy filmet többször is kölcsönadhatunk), a kölcsönadás dátuma, a visszaadás dátuma

A program funkciói

Új film felvitele

A program nem engedi a felvitelt, ha a megadott cím és rendező párossal már rögzítettünk egy filmet.

Filmek listázása

A filmek összes tulajdonsága megjelenik egy táblázatban.

A táblázatban kijelölhető egy film. A képernyőn látható a kijelölt film borítóképe.

Keresés (szűrés) a következő adatok alapján: cím, rendező, főszereplő, gyártási év, hossz, adathordozó típusa (DVD/VHS), az adathordozó eredeti vagy kalóz másolat, kölcsön van-e adva

Film adatainak módosítása

Film törlése

Egy filmet csak akkor lehet törölni, ha nincs kölcsönadva.

Film kölcsönadása

Kölcsönadott film visszavétele

Kölcsönzések listázása

A táblázatban látható a film címe, rendezője, a kölcsönző személy neve, a kölcsönadás dátuma, a visszaadás dátuma.

Beállítható, hogy a lista csak az aktuálisan kölcsönadott filmekre vonatkozzon, vagy a régi kölcsönzések (már visszahozott filmek) adatait is mutassa.

Kölcsönzések keresése (szűrés): a kölcsönző személy neve alapján

A programban legyen egy **Pánik** gomb. Ha jön a szoftverrendőrség, a Pánik gombbal azonnal törölhető az összes kalózmásolattal kapcsolatos információ.

2. Multiplex mozi (Készítette: Bodó Alex)

Az adatbázisban tárolt adatok

film: cím, gyártó ország, szinkronizált-e, rendező, a tartalom leírása, hossz, maximum hányszor lehet lejátszani, korhatár besorolás (1., 2., 3.), a film plakátjának a képe

terem: név, sorok és oszlopok száma

előadás: film azonosító, kezdő időpont, terem azonosító

hely: előadás azonosító, terem azonosító, sor, oszlop, státusz (foglalt, kiadott)

A program funkciói

Filmek listázása

A filmek összes tulajdonsága megjelenik egy táblázatban.

A táblázatban látható az egyes filmekre eladott jegyek száma.

A táblázatban kijelölhető egy film. A képernyőn megjelenik a kijelölt film plakátjának a képe.

Új előadás meghirdetése

Meg kell adni a filmet, a termet, az időpontot.

A programnak figyelnie kell a következő korlátozásokra: Egy film nem vetíthető a megengedettnél több alkalommal. Egy film maximum 3 teremben vetíthető párhuzamosan. Egy terembe egyszerre csak egy előadás osztható be (a filmvetítés után fél óra takarítási idő van). Az 1. korhatár besorolású filmek bármely időpontban vetíthetők, a 2. csak 17 óra után, a 3. csak 21 óra után.

Előadás törlése

Csak akkor törölhető egy előadás, ha senki sem foglalt rá helyet.

Előadások listázása

Látható, hogy az egyes előadásokra hány szabad hely van.

Előadások szűrése

Film vagy terem alapján.

Helyfoglalás

A meghirdetett előadásokra lehet helyet foglalni. A program grafikusan megjeleníti a képernyőn a terem széksorait. A szabad és a lefoglalt székeket más-más szín jelöli. A helyfoglalásnál ellenőrizni kell, hogy az adott szék szabad-e.

Foglalás visszavonása

3. Könyvtár (Készítette: Bodó Alex)

Az adatbázisban tárolt adatok

tag: név, cím, könyvtárjegy szám

könyv: szerző, cím, ISBN szám, kiadás éve, a könyvtári könyvpéldányok száma, a könyv borítójának a fényképe

könyvpéldány: könyv azonosító, raktári szám (egy könyvből több könyvpéldány van)

kölcsönzés: tag azonosító, könyvpéldány azonosító, kölcsönzés dátuma, visszahozás dátuma

A program funkciói

Tagok listázása

A tagok összes tulajdonsága megjelenik egy táblázatban.

Könyvek listázása

A könyvek összes tulajdonsága megjelenik egy táblázatban. A táblázat mutatja, hogy egy könyvből hány szabad könyvpéldány van bent a könyvtárban. A táblázatban kijelölhetünk egy

könyvet. A képernyőn megjelenik a kijelölt könyv borítójának a fényképe, és egy listában megjelennek a könyvhöz tartozó könyvpéldányok raktári számai.

Könyv zárolása

A könyvek zárolhatók. Ilyenkor a könyv példányaira új kölcsönzés nem vehető fel. (Nem az egyes könyvpéldányokat zároljuk, hanem magát a könyvet. A zárolás a könyv összes könyvpéldányára vonatkozik.)

Könyv zárolásának feloldása

Új tag felvétele

Nevet és címet kell megadni. A könyvtárjegy száma automatikusan generálódik.

Tag adatainak módosítása

A könyvtárjegy száma nem módosítható.

Tag törlése

Egy tag csak akkor törölhető, ha nincs nála könyvpéldány.

Új kölcsönzés felvitele

Egy tagot és egy könyvpéldányt kell kijelölni. Csak akkor lehet a könyvpéldányt kikölcsönözni, ha szabad, és a könyv nem zárolt. Egy tag maximum 6 kikölcsönzött könyvpéldányt tarthat magánál. A kölcsönzés dátumát tároljuk.

Kölcsönzések listázása

Ha egy tagot kiválasztunk, egy táblázatban megjelennek a kölcsönzései.

Kölcsönzött könyvpéldány visszavétele

Kijelölünk egy kölcsönzést. A visszavétel dátumát tároljuk.

4. Internet kávézó (Készítette: Bodó Alex)

Az adatbázisban tárolt adatok

ügyfél: név, cím, személyi igazolvány szám, felhasználónév, jelszó, az ügyfél fényképe

számítógép: azonosító, hardver leírás, operációs rendszer, működik-e a gép

használat: ügyfél azonosító, számítógép azonosító, bejelentkezés időpontja, kijelentkezés időpontja

számla: ügyfél azonosító, összeg, kijelentkezés időpontja

A program funkciói

Számítógépek listázása

A számítógépek összes jellemzője megjelenik egy táblázatban.

Új számítógép adatainak rögzítése

Egy új számítógépet állítunk rendszerbe.

Számítógép adatainak módosítása

Az azonosító nem módosítható.

Ügyfelek listázása

Az ügyfelek összes adata megjelenik egy táblázatban. A táblázatban kiválaszthatunk egy ügyfelet. A kiválasztott ügyfél fényképe megjelenik a képernyőn.

Új ügyfél felvitele

Ügyfél adatainak módosítása

Ügyfél beléptetése egy számítógépre

Egy ügyfelet és egy számítógépet jelölünk ki. Csak szabad számítógépet lehet kijelölni, mely működik. Egy ügyfél egyszerre csak egy számítógépet használhat. A belépés időpontja automatikusan tárolódik.

Ügyfél kiléptetése egy számítógépről

Ha az ügyfél befejezi a számítógép használatot, a kezelő pulthoz jön, és szól, hogy léptessük ki. Kijelöljük az ügyfélre vonatkozó aktuális használati bejegyzést, és az ügyfelet kiléptetjük. A kiléptetés időpontja automatikusan tárolódik. A program kiszámolja a számla összegét. A számítógép használatnak van egy óradíja. Az óradíj a programban beállítható. A fizetendő összeget kedvezmény csökkentheti. Egy ügyfél 20 óra számítógép használat után 1% állandó kedvezményt kap, mely 20 óránként további 1%-kal nő, de legfeljebb 10% lehet. A számlát elmentjük az adatbázisba.

Ügyfél számláinak listázása

5. Autó kölcsönző (Készítette: Bodó Alex)

Az adatbázisban tárolt adatok

ügyfél: azonosító szám, név, cím, telefonszám

autó: rendszám, márka, típus, évjárat, napi bérleti díj, az utolsó szervizelés dátuma, az autó most szervizben van-e, az autó fényképe

kölcsönzés: autó azonosító, ügyfél azonosító szám, a kölcsönzés kezdő dátuma, a kölcsönzés végének tervezett dátuma, a visszahozatal dátuma

A program funkciói

Autók listázása

Az autók összes adata megjelenik egy táblázatban. A táblázat minden sorában megjelenik az autó fényképe is.

Új autó felvitele

Autó adatainak módosítása

A módosítható adatok: napi bérleti díj, az utolsó szervizelés dátuma, az autó most szervizben van-e, az autó fényképe

Új ügyfél felvitele

Kölcsönzések listázása

A táblázatban megjelenik az ügyfél azonosító száma, neve; az autó rendszáma; a kölcsönzés kezdő dátuma, a kölcsönzés végének tervezett dátuma, a visszahozatal dátuma. A már visszahozott autóknál a fizetett összeg is látható. Beállítható, hogy csak a folyamatban lévő kölcsönzések legyenek láthatók vagy a már befejezett kölcsönzések is megjelenjenek.

Kölcsönzések szűrése

Rendszám szerint vagy ügyfélnév szerint.

Új kölcsönzés indítása

Egy ügyfelet és egy autót jelölünk ki. A kölcsönzés az aktuális nappal indul, és meg kell adni a kölcsönzés végének a tervezett dátumát (ez egy jövőbeli időpont). Egy kikölcsönzött autó nem kölcsönözhető. Egy szervizben lévő autó nem kölcsönözhető. Egy ügyfél egyszerre csak egy autót kölcsönözhet.

Az 5 évnél fiatalabb autókat évente, az 5-10 év közöttieket fél évente szervizelni kell. Az autókat csak a következő szerviz idejét megelőző lejáratú dátummal lehet kölcsönözni. A 10 évnél idősebb autók nem kölcsönözhetők.

Kölcsönzés lezárása

Az ügyfél visszahozza az autót. Kijelölünk egy kölcsönzést és lezárjuk. Ez lehet előbb, illetve később is, mint a kölcsönzés végének a tervezett dátuma. Ha az ügyfél előbb hozza vissza az autót, akkor a fennmaradó napokra a napi bérleti díj felét kell fizetni. Ha később hozza vissza az autót, akkor a kölcsönzés végének a tervezett dátuma utáni napokra a napi bérleti díj dupláját kell fizetni. A program írja ki, hogy mennyit kell fizetni az ügyfélnek.

6. Trónok Harca (Készítette: Nagy Krisztián)

Lord Varys (a pók), egész Westeros-ra és azon túlra is kiterjedő kémhálózatot üzemeltet. Mivel az elmúlt években nem sikerült elérnie a célját, így újabb követet küldött hozzád, hogy készíts neki egy olyan adatbázis alkalmazást, melyben eltárolhatja a kémhálózatán keresztül összegyűjtött információkat.

Minimum elvárásai a programmal kapcsolatban:

- Legyenek megtekinthetők az ismert házak, karakterek és a házak közötti szövetségek.
- Legyen lehetősége új házat felvenni a nevének, címerének és a mottójának megadásával. A címerek képét tárolni kell és meg kell jeleníteni a programban.
- Legyen lehetősége új karaktert felvenni a nevének, irányítása alatt álló seregek számának és sorozatbeli státuszának (élő/elhunyt) megadásával. Fontos, hogy egy karakter tartozhat valamelyik házhoz!

- Legyen lehetősége egy szövetség megadására. Egy szövetség két ház között jöhet létre és mindig van egy kezdődátum a megpecsételésekor és előfordulhat egy végdátum is, abban az esetben, ha megromlott a kapcsolat.
- A házakról tudjuk, hogy nem változtatnak nevet, címet és mottót. A karakterekhez tartozó irányított seregek száma és a sorozatbeli státusza változhat. Nyilvánvaló, hogy amennyiben elhunyt egy karakter, úgy nem irányíthat sereget és az Éjkirály mágiája se tudja újra életre kelteni. A szövetségek esetén pedig csak a végdátumot módosíthatjuk.
- Legyen lehetősége szűrni egy adott karakterre a neve alapján.

A felületen nem szükséges megjeleníteni az azonosítókat.

Az adatbázisban az alábbi adatokat tároljuk (ezek még nem feltétlenül a fizikai adattáblák):

- ház (név, címer, mottó, a címer képe)
- karakterek (név, sereg szám, státusz, ház)
- szövetség (ház1, ház2, kezdete, vége)

Az adatbázis tervezésénél ügyelj az adatbázis táblák normalizálására!

7. Harry Potter (Készítette: Nagy Krisztián)

Roxfortban a felvett varázsló ifjoncokat a Teszlek Süveg osztotta be az iskola 4 házának (Griffendél, Hugarbug, Hollóhát, Mardekár) egyikébe, azonban a hosszú évek során szegény süveg elkopott. Albus Dumbledore-nak új megoldást kellett találnia a diákok elhelyezésének és számontartásának a biztosítására, továbbá fel szeretné jegyezni a környéken élő ismert lényeket is. 2 nap tanakodás után felkérte Pomona Bimba professzort, hogy termesszen mágikus babot (NetBeans), melyet felhasználva létrejöhet a mágiaügyi nyilvántartó rendszer.

Dumbledore a történetek után felkért téged, hogy készítsd el az alkalmazást.

Minimum elvárásai a programmal kapcsolatban:

- Legyenek megtekinthetőek a házak, tanulók, a lények és a jelleme.
- Legyen lehetősége új ház felvitelére a ház nevének és címerének megadásával. (Az iskola bővíthet új házakkal.) A címerek képét tárolni kell és meg kell jeleníteni a programban.
- Legyen lehetősége új tanulók felvételére a tanuló nevének és életkorának megadásával. Fontos, hogy egy tanuló tartozhat valamelyik házhoz!
- Legyen lehetőségünk új lény felvitelére a lény nevének megadásával.
- Minden lényről és tanulóról számon tartjuk, hogy milyen a jelleme. A jellem lehet törvényes jó, semleges vagy kaotikus gonosz. Új lények és tanulók esetén tegyük fel, hogy mindenki semleges. A jellemet lehessen bővíteni.
- A tanulókról tudjuk, hogy a nevük nem változhat meg. A jelleme elnevezése sem változhat. A házak esetén csak a házhoz tartozó tanulók száma változhat, azonban ez a mező sem szerkeszthető felhasználói interakcióval. A lények esetében számon tartunk egy első találkozási dátumot is, melyet csak abban az esetben lehet szerkeszteni, ha még nem volt töltött.
- Legyen lehetőségünk szűrni tanulókra és lényekre a nevük alapján.

A felületen nem szükséges megjeleníteni az azonosítókat.

Az adatbázisban az alábbi adatokat tároljuk el (ezek még nem feltétlenül a fizikai adattáblák):

- ház (név, címer, tanulók száma, a címer képe)
- jellem (megnevezés)
- tanuló (név, életkor, jellem)
- lény (név, első találkozás dátuma, jelleme)

Az adatbázis tervezésénél ügyelj az adatbázis táblák normalizálására!