

Projektbeskrivning för DA147A distans, VT15

Genomförande

Projektet ska utföras **i grupp** eller **som enskilt arbete**.

Om gruppen får problem så finns två alternativ:

1. Är frågan/problemet av allmän art så publicera på forumet.
2. Är frågan speciell för gruppen så skicka **Meddelande** via It's learning till Rolf Axelsson eller Kristofer Leifland.

Redovisning

Senast den 21/5 ska allt material levereras till kurssidan under **Inlämninguppgifter**.

Projektgrupper lämnar in under en av gruppdeltagarna, enskilda lämnar in som vid programmeringsuppgifter. Leveransen ska ske i form av en zip-fil, **projekt.zip**. Filen ska innehålla följande:

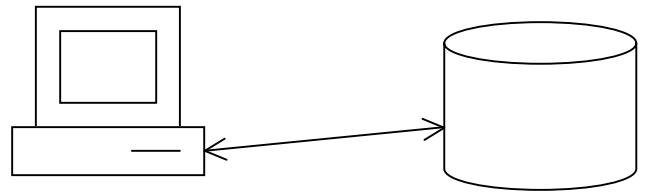
- Alla källkodsfiler som ingår i projektet. De ska vara i katalogen src. Källkodsfilerna ska vara javadoc-kommenterade. Tillsammans med källkodsfilerna ska det vara en instruktion så att jag kan kompilera och köra applikationen.
- Ansvar.txt vilket innehåller en lista över projektets deltagare och en beskrivning över vad varje deltagare bidragit med i lösningen (ej vid enskilt arbete).
- Skriptet **sql.txt**.
- **Systemmanual**. Kalla filen systemmanual.doc. Se avsnittet "Dokumentation" för mer information.

Viktiga datum

- **14 april**
Projektuppgiften presenteras för studenterna
- **20 april**
Studenterna presenterar färdiga projektgrupper för mig via **Meddelande** på It's learning. Alternativet till projektgrupp är att man uppger att man gör projektet på egen hand (en lite mindre uppgift).
- **27 april**
Lämna in dokumentation som innehåller följande:
 1. Beskrivning av vad programmet ska göra.
 2. Skisser på hur användargränssnittet i programmet ska se ut, dvs hur fönstret/fönstren som visar applikationen ska se ut. Fundera även över funktionen i gränssnittet, dvs vad användaren ska kunna göra.
 3. En uppräkningslista av de klasser som ska användas.
 4. En uppräkningslista av de tabeller som ska finnas i databasen.Lämna in via **Inlämninguppgifter – Projekt 27 april** på kurssidan.
- **22 maj**
Redovisning via **Inlämninguppgifter – Projekt 22 maj** på kurssidan.

Uppgift

Att designa och implementera en applikation som visar information som lagras i en databas. Applikationen ska även kunna lägga till / ta bort / modifiera informationen i databasen. Informationen kan t.ex. utgöras av varor i en nätbaserad affär, roliga historier,



I specifikationen nedan benämns informationen som varor.

Applikationen ska kunna hantera tre kategorier av användare:

1. Ej inloggad användare
2. Inloggad användare (kund)
3. Personal

De som arbetar **enskilt** ska hantera **kategori 1** och **kategori 2** ovan.

De som arbetar i **projektgrupp** ska hantera **samtliga kategorier** av användare.

Specifikation av applikationen

- Applikationen kopplar upp sig mot databasen vid start (egentligen en säkerhetsrisk).

För användare i kategori 1 (ej inloggad användare) gäller:

- En användare ska kunna se samtliga varor som erbjuds. Användaren ska också kunna välja att endast se en viss kategori av varor, t.ex. dvd-spelare. Valet av vad som ska visas kan göras med hjälp av knappar, combobox, list-komponent el.dyl.
För varje vara ska kundernas betyg av varan presenteras (som ett medelbetyg)
- En användare ska kunna registrera sig som kund. Vid registreringen ska uppgifter som namn, personnummer, adress, telefonnummer, någon form av användarnamn och lösenord anges.
- En användare ska kunna logga in som kund. Vid inloggning ska någon form av användarnamn och lösenord användas.
- En användare ska kunna logga in som personal. Vid inloggning ska någon form av användarnamn och lösenord användas. Denna punkt gäller endast projektgrupper.

För användare i kategori 2 (inloggad användare – kund) gäller:

- En kund ska kunna se samtliga varor som erbjuds. Användaren ska också kunna välja att endast se en viss kategori av varor, t.ex. dvd-spelare. Valet av vad som ska visas kan göras med hjälp av knappar, combobox, lista el.dyl.
För varje vara ska kundernas betyg av varan presenteras (t.ex. som ett medelbetyg)
- En kund ska kunna sätta betyg på de olika varorna i affären
- En kund ska kunna logga ut och bli kategori 1-användare

För användare i kategori 3 (personal) gäller:

- En inloggad personal ska kunna:
 - * lägga in en ny personal i systemet.
 - * lägga in en ny vara i sortimentet.
 - * ta bort en vara ur sortimentet.
 - * ändra i en vara, t.ex. antalet varor i lager.
- En personal ska kunna logga ut och bli kategori 1-användare

Databas

Använd någon gruppmedlems databaskonto här på MAH. Eftersom inloggningsuppgifterna är samma som till It's Learning mm så ska lösenordet till databaskontot ändras innan databaskontot börjar användas i projektet (se <https://ddwap.mah.se/dotnet/index.aspx>).

En databas ska lagra nödvändig information i systemet. Tabeller i databasen kan t.ex. vara:

- vara(varuID, namn, *kategoriID*, antal, pris)
- kategori(kategoriID, namn)
- person(personnummer, namn, gata, postnr, postadr)
- kund(personnummer, användarnamn, lösenord, ...)
- personal(personnummer, användarnamn, lösenord, ...).
- telefon(personnummer, telefonnummer) – medger flera telefonnummer till en kund/personal.
- betyg(varuID, personnummer, betyg)

Primärnyckeln är understruken och referenser till andra tabeller är kursiva.

Vid enskilt arbete kan tabellerna person och kund slås ihop till en tabell. Samtidigt utgår tabellen personal.

Gruppen ska skriva ett **skript**, **sql.txt**, vilket förser en tom databas med tabeller och testdata. Detta för att det ska gå att återställa förlorad information om något händer databas-servern. Det blir dessutom möjligt att återställa en databas till dess ursprungliga form genom att tömma den och sedan köra skriptet.

Dokumentation

Följande dokument ska levereras den 22/5:

Systemmanual	Systemmanualen beskriver applikationen och databasen. Här begränsar vi oss till följande: Applikation: Beskrivning av ingående klasser med klassdiagram. Databas: Beskrivning av ingående tabeller. Dessutom en uppräknig av de SQL-frågor som ställs till databasen, inklusive förklaring till vad resultatet av frågorna är.
--------------	--

Exempelfiler

I filen **DA147AProjekt.zip** hittar du några exempel och en bild-komponent:

- Att visa olika paneler i samma fönster med hjälp av *JTabbedPane* (flikar). Filen heter **ExampleJTabbedPane.java**.
- Att visa olika paneler i samma fönster med hjälp av layoutmanager *CardLayout*. Filen heter **ExampleCardLayout.java**.
- Att använda färdiga paneler i dialoger-fönster. Detta för att erhålla information från användaren. Det är två exempel
 - 1 **ExampleJOptionPane.java + JColor.java**
 - 2 **Example2JOptionPane.java + ComponentPanel.java**
- Att visa en bild med **ScaleIcon** eller **JImage**-komponenten. Komponenterna demonstreras med **TestScaleIcon.java / TestJImage.java**.