

Rapport Lab 5 Version 3 – Örjan Andersson (alfoande100)

OBS!

Denna rapport innehåller bara nya saker från rapport lab 4. Båda rapporterna måste läsas för att få hela bilden.

1. Antaganden

- Lab 5 bygger på kod från lab 4.
- Det är centrallagret som alltid bestämmer vad som är gällande på lagerstatus och pris
- När en enskild produkt uppdateras på lager eller köps i kundkorg uppdateras den direkt i centrallager via API
- Har gjort det enkel att alla produkter med ID under eller lika med 15 uppdateras från/till centrallager. Andra produkter ignoreras.
- När applikationen startar antas det är ok att alla produkter hämtas direkt från centrallager med API och uppdateras lokalt med pris och lagerstatus. Det sker då före automatisk synkronisering som sker varje minut för att hämta lagerstatus och pris från centrallager. Samma sak sker när man trycker på "synkronisera" knappen.
- Det är ok att använd en tredjepartsprodukt för att visa grafer
- Information om statistik på historisk lagerstatus och priser lagras enbart i minnet. Det lagras statistik så fort man tar kontakt med API (automatiskt eller manuellt med knapp) för att hämta information om lagerstatus/pris. Det lagras också när en person som använder lager lokalt i applikationen för att uppdatera en produkt manuellt. Om en produkt blir såld så loggas också detta.
- Det är ok att enbart visa de 15 sista historiska lagerstatus per produkt. All data finns men visuellt visas enbart de 15 sista med hjälp av ett filter.
- Vid en uppdatering från centrallager av pris och lagerstatus så uppdateras produkter i lager, produkter i butik och kundkorg.
- data som lagras lokalt har nu fått en v2 efter sig. Detta gäller nu order_data_v2.csv och products_data_v2.csv. Det är för att tidigare filer inte skall störa om man testar lab5 på samma maskin som man kört lab4. Se lab4 för mer information om dessa filer.

2. Översikt

Under tabben "Lager" finns det 2 nya knappar.

Knapp 1 är "Uppdatera från centrallager" laddar ned data från centrallager och uppdaterar produkter i lokala lager. Knappen visas i Figur 1.

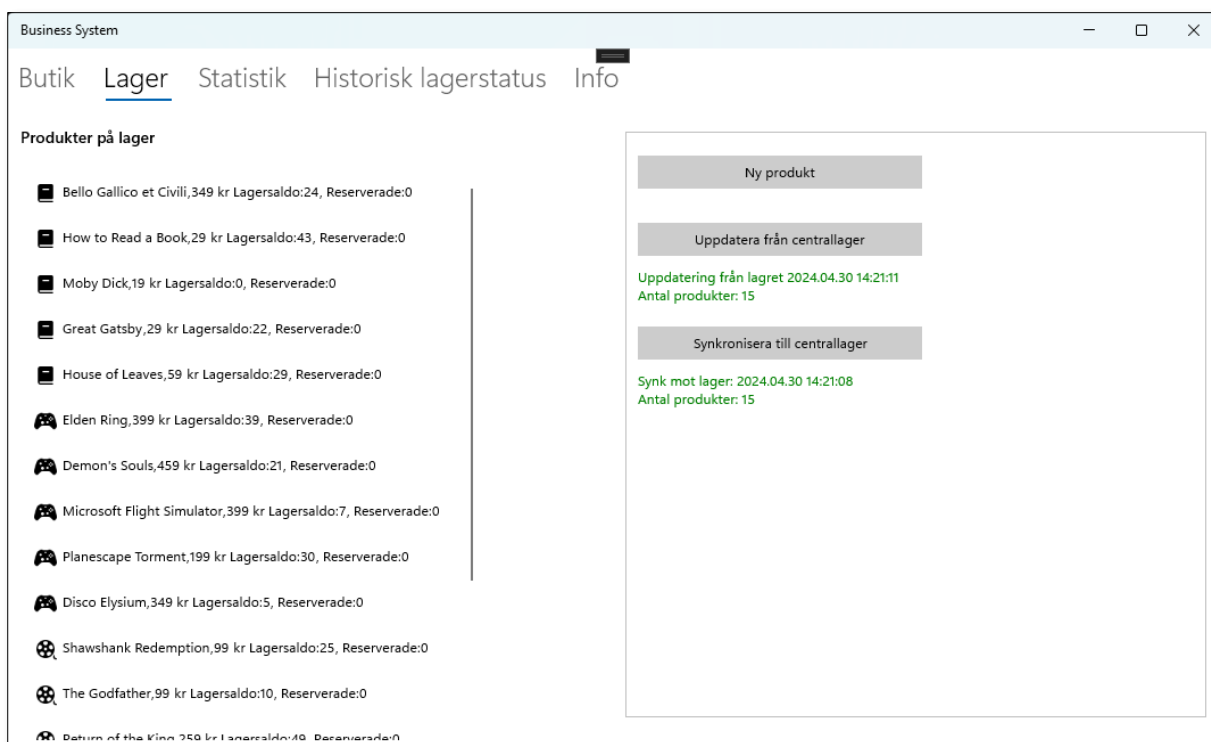
Tidpunkt när detta sker blir uppdaterar rätt under knappen samt hur många produkter som är uppdaterade. Texten för denna uppdatering sker vid följande scenarion:

- Uppstart av applikation. API anropas för att hämta senaste data från centrallager
- När en lagerarbetare trycker på knappen "Uppdatera från centrallager" (grundversion)
- Vis automatiskt synkronisering som sker varje minut (version 3 – uppgift 1).

Knapp 2 är "Synkronisera till centrallager" uppdaterar lagerstatus till centrallager. Knappen visas i Figur 1.

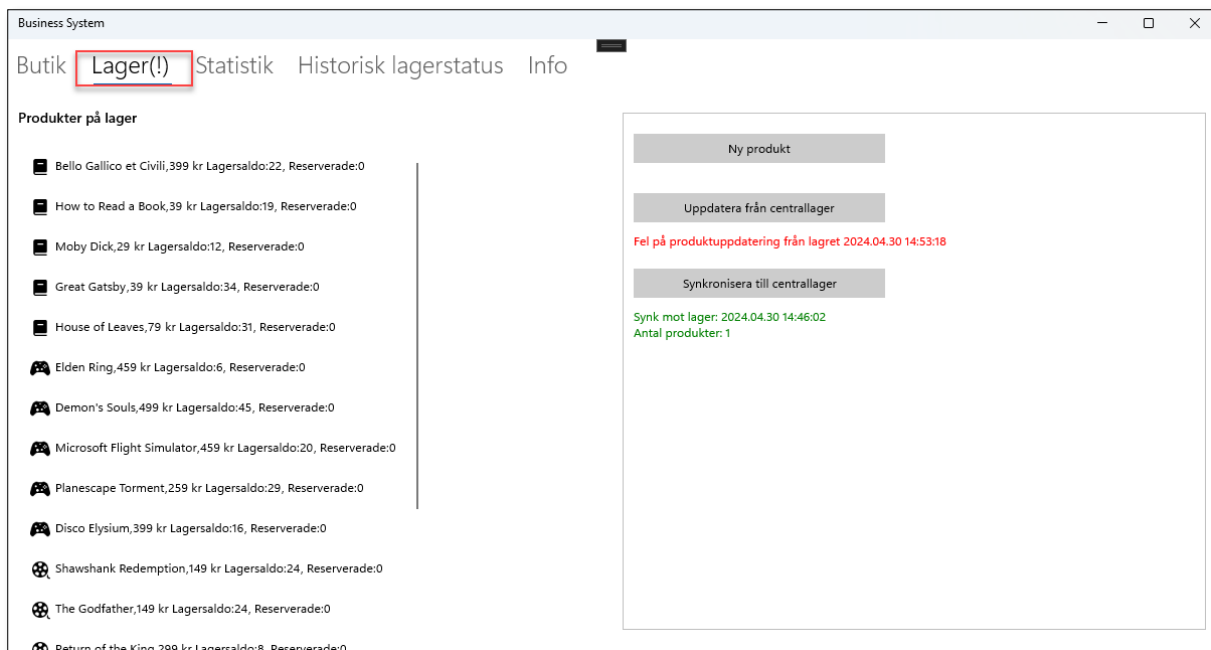
Tidpunkt när detta sker blir uppdaterar rätt under knappen samt hur många produkter som blir synkroniserade. Texten för denna synkronisering sker vid följande scenarion:

- När en lagerarbetare trycker på knappen "Synkronisera till centrallager" (version 2 – uppgift 1)
- När en produkt köps. Här är det valt att enbart synka de produkter som faktiskt säljs för att göra en så skyndsam synkronisering som möjligt. Det ligger kod inne som kan inkluderas för att göra en full synkronisering som nämnt i uppgift. Anser att skyndsam synkronisering prioriteras före i denna uppgift så därför är koden bortkommenderad. (version 2 – uppgift 2)

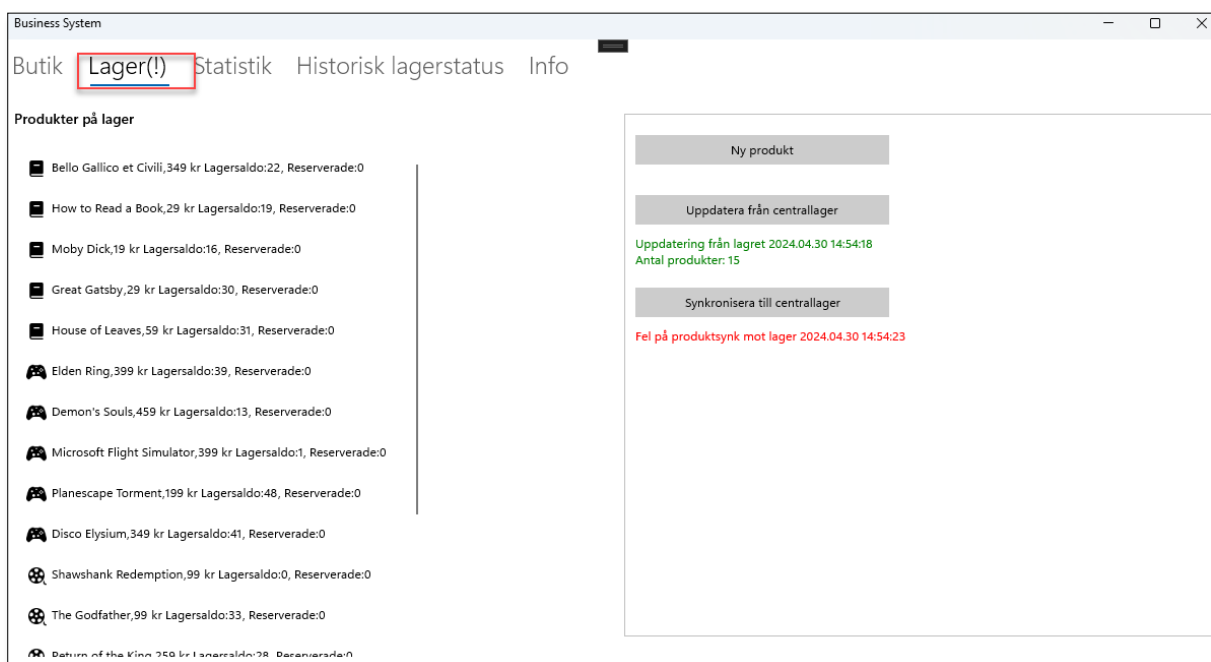


Figur 1 – Knappar för att kommunicera med centrallager

Vid ett eventuellt fel vid uppdatering eller synkronisering så meddelas kassa eller lagerarbetare med ett (!) tecken efter Lager i meny. Det uppdateras också med en röd text under knapparna "Uppdatera från centrallager" och "Synkronisera till centrallager". Hur detta ser ut illustreras i Figur 2 och Figur 3.



Figur 2 – Fel vid uppdatering mot centrallager



Figur 3 – Fel vid synkronisering till centrallager

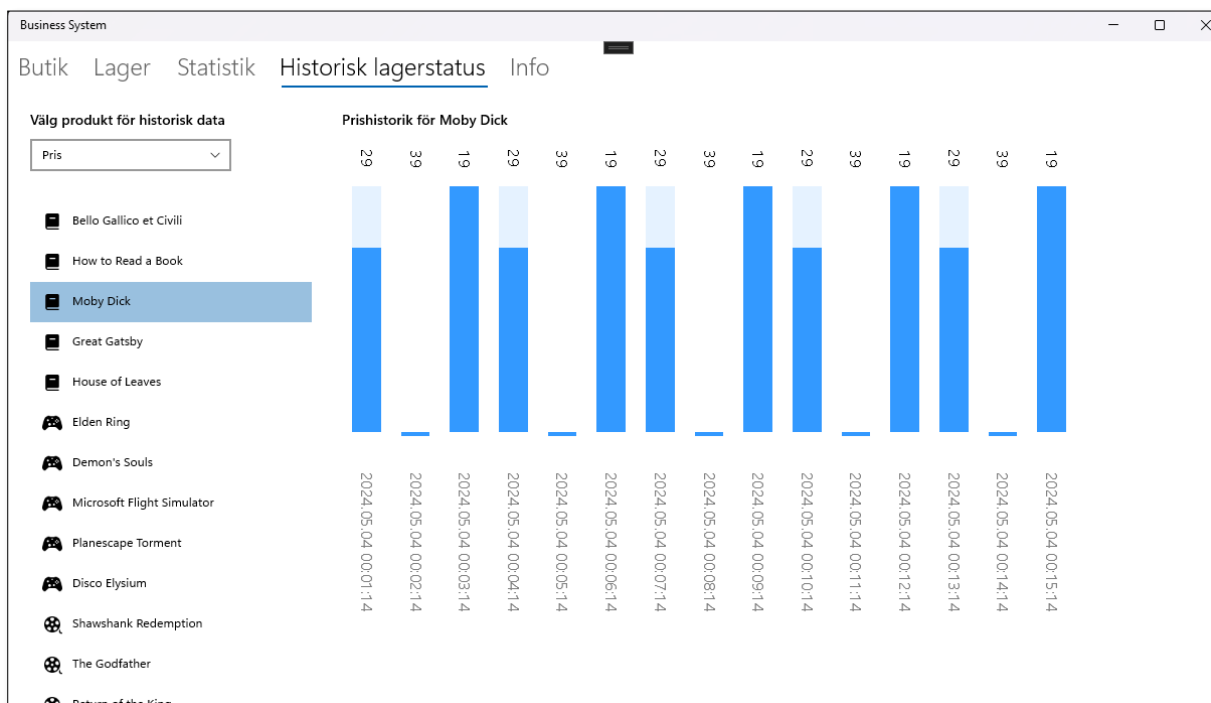
En ny flik "Historiskt lagerstatus" visar historiska lagerstatus och priser för varje produkt.

En kassa- eller lagerarbetare som önskar se på historik över pris (se figur 4) eller lagerstatus (se figur 5) måste aktivt välja det ena eller andra och sedan välja produkt.

Grafen uppdateras enbart vid klick på produkterna. Produkten som används för att visa en graf är Microcharts.

Det loggas historisk information om pris och lagerstatus vid följande tillfällen:

- Vid manuell uppdatering av en produkt i lager
- När produkter hämtas automatiskt per minut
- När en lagerarbetare trycker på knappen "Uppdatera från centrallager"



Figur 4 - historisk pris för varje produkt



Figur 5 - historisk lagerstatus för varje produkt

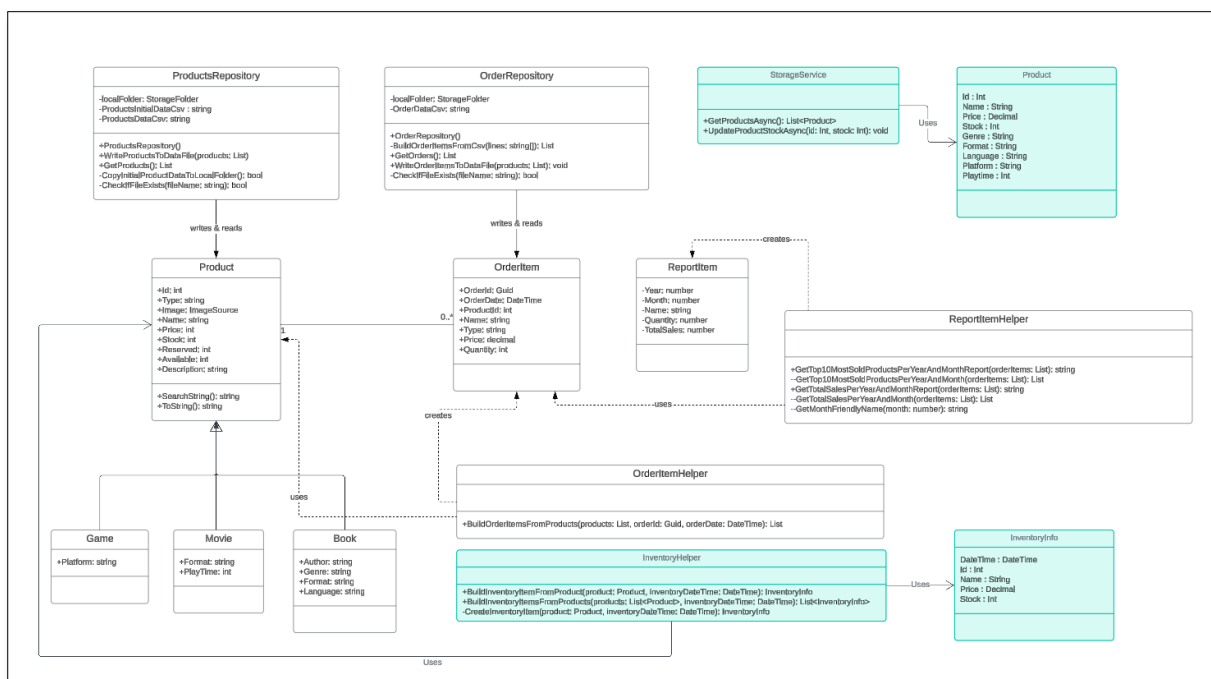
3. Beskrivning av applikation och klasser

En ny lista är upprättad för att hålla information om historisk data. Den heter InventoryList och håller information om ett nytt objekt som heter InventoryInfo. För att jobba med denna används en ny helper class som heter InventoryHelper

För att kommunicera med centrallager är det upprättad en ny service som heter StorageService. Den används för att hämta och skriva produktinformation mot centrallager. StorageService använder en egen version av Produkt class för att bättre matcha XML från API.

För att rita grafer så upprättas det objekt baserat på klasser som finns från Microcharts. Dessa byggs upp genom att använda sig av InventoryHelper

Nytt klassdiagram efter ändringar. Se lab 4 för övrig beskrivning.



Figur 6 – klassdiagram efter uppdatering lab 5.

4. Problem

Problem med att hitta ett fungerande bibliotek för visning av grafer. Microsoft var inte direkt mycket till hjälp med sin dokumentation. Fick prova flera olika bibliotek innan jag hittade en som fungerade.

Väldigt kort tid på uppdatering av data från API. Gör det svårt att testa när data uppdateras lite för fort eller av andra som kodade mot samma API.

Med fördel kan ett API implementeras som är personligt där det inkluderas id till student i url till API. Med fördel kan API valfritt stötta JSON. En API punkt för att kunna uppdatera mer än 1 produkt åt gången kan kanske också byggas för mer effektiv kommunikation.

5. Sammanfattning

Det som tog väldigt lång tid var att få ordning på visning av grafer på ett bra sätt. Här hade det med fördel funnits refererat till ett bibliotek som skulle användas.

Estimerad tidsåtgång:

Grundversion: 10 timmar

Version 2: 10-15 timmar

Version 3: 20-25 timmar

Dokumentation: 1 dag

6. Referenslista och bilagor

Referenser till källor finns dokumenterat i källkoden.

Bilagor Lab5_Version3.zip