## Observer pattern:

Observer pattern også kalt publisher subscriber pattern, er et design mønster som overordnet går ud på at få viderebragt information om at der er sket ændringer. Måden det foregår på er ved er at man har en observer, som kan observerer på noget (subject), og derved få besked om ændringer. Ret praktisk sker dette i 3 faser:

1. Observeren tilmelder sig som hos subject.
2. Subject meddeler observeren om tilstandsændringer hver gang det sker.
3. Observeren framelder som som observer på subject.

I observer pattern finde der to forskellige teknikker til at oplyse om tilstandsændringer, push og pull. Vi har valgt at bruge push, som går ud på at subjectet både sender en reference til som parameter, og en oplysning om hvad der er sket. Hvor pull kun fortæller hvem, men ikke hvad.

Vi har i vores projekt anvendt ideen fra observer pattern om at kunne få besked om når der er sket ændringer, og vi har lavet vores egen FFSObserver som er et interface, med 1 enkelt metode som hedder update. Update tager 2 parametre, den første som er et Object det andet som er en string. Object bruger vi til at kunne finde ud af hvilken af vores konkrete observere der har givet besked. String bruger vi til at finde ud af hvilken metode der er blevet kaldt altså push.

For at kunne få det til at fungerer, har vi så lavet 2 metoder i hver af vores controllere som som hedder tilmeldObserver med en observer som parameter og en notifyObserver med en string som parameter. TilmeldObserver tilføjer observeren til en liste hvis den ikke allerede findes i den. NotifyObserver kører så listen igennem og kalder update på observerne, med den string som kommer fra notifyObservers.

For så endelig at kunne holde styr på hvad der skal opdateres, har hver metode med tilstandsændringer en notifyObserver motode med en string med metodens navn, som vi så i update metoden kan bruge til vælge hvad der skal ske når en metode er blevet kaldt.