

# Reflektioner Workshop 2 - GaldanCRUD

## Implementation/Kodning:

- Vi har inte implementerat möjligheten att avbryta ett användningsfall.
- Vi känner en stor osäkerhet, trots ett aktivt sökande av svar, kring innebörden av dolda beroenden. Vi är därför osäkra på om vår lösning uppfyller kravet kring detta.
- Vi har eftersträvat att kapsla in våra klassmedlemmar som långt som möjligt. Dock finns en risk för "privacy leak" på våra listor då dessa ej är "read-only".

## Klassdiagram:

- Vi är lite osäkra på klassdiagrammet, det vi inte vet är vad skillnaden är på dependency och association. Det fanns också en osäkerhet när composition skulle användas istället för dependency.

## Sekvensdiagram:

- Vi har fått en bättre uppfattning om sekvensdiagram. Dock så känner vi att det hade behövts en mer introducerande uppgift till sekvensdiagram innan denna uppgift.

## Patterns:

### - High Cohesion

Med tanke på storleken på projektet så fungerar mängden kod skriven i kontrollern bra, men vi hade möjligtvis kunna bryta ut vissa delar från kontrollern till exempelvis en userController och en boatController.

### - Low Coupling

Vi tycker att vår lösning uppfyller low coupling då våra klasser är väldigt självständiga och det inte finns så många beroenden.

### - Information expert

Vi upplever att vi uppfyller detta krav.

**Om uppgiften:**

- Det känns som att uppgiften har gett mycket repetition av C#. Det kändes inte riktigt som om vi fick ut så mycket UML-kunskaper av uppgiften. Det hade kanske varit bra med mer "tvingande" krav, men innan det skulle vi behövt mer handhållning i uppgiftsform.
- Vi har fortfarande svårt att se skillnad på associationer, dependencies (hidden dependencies).
- Vi har insett värdet av att göra sekvensdiagram före/ i samband med början av projektet. Att göra diagrammet efter visade tydligt att vi fått duplicerat kod. Man ser även tydligt kopplingarna mellan klasserna i sekvensdiagrammen.