

# Handledarmöte 5

- **Närvarande:** Edvin, Tina, Oscar, Norbert, Jacob, Elin
- **Datum:** 18/10-2019
- **Mötespunkter:**
  - **Från Edvin:**
    - MVC: Just nu hög coupling, inget gemensamt gränssnitt för controller och vy. Refaktorera!
    - Sista mötet med Edvin: Förbereda presentation för edvin så han kan ge feedback. (i mån av tid)
    - Viktigt för inlämningen: Allt ska vara konsekvent. Domänmodell, UML och kod ska stämma överens.
    - Skriv mer om vad en habit är, att det är vana osv. Gör horisontella linjer mellan US. Använd IEEE när det kommer till källor.
    - UML: svårt att få överblick. svårt att se våra designtankar, designmönster osv .Om flera går till samma grej, ha en pil som splittar upp sig.
    - Agg: tar jag bort objekt skulle det finnas kvar, composition då försvinner objektet när det tas bort.
    - Composition mellan factory o handler?
    - Composition mellan dag o veckovyn - Helge vet mer.
    - Aggregation kan vara att föredra, men måste utgå från vår egna situation.
    - Has-a är högre upp än comp o aggr. Om man inte vet mellan comp o aggr. Eller inte bryr sig kan man Sätta has a
    - SDD: I system design, skriv om designmönster.
- **Frågor:**
  - **Hur tester för vy och controller**  
Vy: Inte för vy. Vy är så dum att det inte finns beräkningar att räkna

Controller: Skapa mockobjekt för att göra tester, måste ha speciella bibliotek. Strunta i det säger Edvin, bättre att satsa på mer på modellen o få den så nära 100% som möjligt. (behöver ej testa getters o setters). Testa att skicka in felvärden.

Modell: Hellre 100% test coverage här

- **Ska man gå igenom rad (o sdd) i peer review? Står inget specifikt om det i inlämningen...**  
Ha med, det är bra! Ju mer desto bättre.
- **Presentation**  
Lämna in en lista med US och gå igenom dem. Ha en us om att lägga till todo, visa att det går att göra i vår demo.  
Teknisk genomgång: highlighta någon del av koden vi är extra nöjda med, hur vi tänkt. Vår process. Lyft designmönster. Kommer bara titta på innehåll ej själva presentationen.
- **När får man responsen från peer review? Är tanken att vi ska implementera enligt responsen vi fått?**  
Inte hinna implementera, ej förväntat. Kunna ge tips till typ dokumenten.
- **Peer review, står att det ska vara 1-2 sidor. Ingår rubrikerna i det eller är det bara vår respons som räknas?**  
Bra som det är.
- **Hur gör man FAKTISKT mvc pattern när man använder javaFX, view är ju faktiskt fxml-filen men man manipulerar den view med controllern som också tar emot input, hur ska man tänka?**  
Vi vill inte få det fel. Finns viss dispans med javafx pga av starka kopplingen mellan vy o c. Vi borde implementera en facade i modellen för att inte visa upp allt. Vill inte ha mer än en ingång egentligen. Kanske inte går att ha bara en eftersom vi de kanske inte går ihop konceptuellt. Kan behöva en annan typ av identifiering. Måste få mindre coupling  
Listeneer mellan vy o modell  
Facade mellan controller och modell.  
Måste lösa det med chain of responsibility eller state-pattern(controller) för att följa SRP.
- **Listeners och habits**  
Skapa två lyssnare, en i handler och en i habit. Behöver inte vara kopplade till varandra utan kan vara två helt enskilda. Måste koppla det till facade så att det inte blir beroende direkt till modellen.
- **Vi måste komma igång med databasen, har du tips o ideer/kan hjälpa oss att komma igång.**  
Alternativen: Text-file, xml eller Json

Textfil: Java standardbibliotek Har grejer för textfiler. Kommer få en stor stäng, som vi sen kommer behöva parca. Kan ha en för varje del, habit, todo...  
Stackoverflow

Json: lite smartare, kan spara i objekt. Finns mkt bibliotek för Json. ej en del av java standard-bibliotek. Kolla om det finns ändå, edvin ej helt säker. Tror dock inte det. Man ha sin json filer och sen nått som kan läsa dem filerna och tolka.

Börja med att installera ett json bibl. Skpa filerna och få den att läsa av filerna.

Fördel: vi behöver ej skriva egen parcerdet måste vi göra i te

Vad händer om vi ej lyckas: rent funktionsmässigt ät det viktigt,