Velkommen til 3. gruppetime

...

Gruppe 2
Andrea og Fredrik

Agenda

- Teamarbeid
- Arkitektur
- Lab

Teamarbeid

- Smidig: scrum, kanban, scrumban
- Vær inkluderende og positiv til innspill
- Teamkontrakt

Teamarbeid diskusjon

- Hva mener dere er god profesjonell holdning i et team?
- Hvordan vil ‘tilbakemeldinger’ påvirke individer?
- Hvordan bygge tillit i et team, evt teknikker for å bygge/ forbedre tillit?

Arkitektur

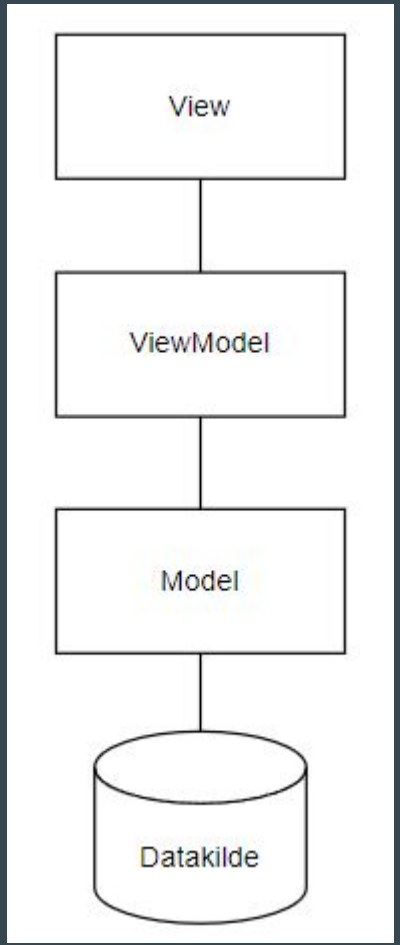
- Hvorfor burde arkitektur være (ca.) på plass før man starter å programmere?

- I **startfasen**, så hjelper det oss å **visualisere** og tenke over **hvordan appen skal fungere.**
  - Hva skal appen gjøre?
  - Hvilke type data har vi/trenger vi?
  - Hvordan skal vi representere dataen?
  - Må vi gjøre noe med dataen? e.g behandle den / beregninger?

- Under **utviklingsprosessen:**
  - Lettere å vedlikeholde, skalere og teste appen.
  - Endringer i arkitektur i store prosjekter er tidkrevende og dyrt, teknisk gjeld

MVVM -> Model, View, ViewModel

- Standard arkitektur i Android
- Poenget?
  - Klart skille mellom User Interface (UI) og appens data. SOC prinsippet
- UI laget:
  - **View** ansvar:
    - Presentere state til bruker (ved å observere state i ViewModel)
    - muliggjøre brukerinteraksjon
  - **ViewModel** ansvar:
    - presentere state til View
    - oppdatere state (kan innebære å starte henting av data, og å reagere på brukerinteraksjon)
- Data laget:
  - **Model** ansvar:
    - hente data(faktisk gjøre kallet) og behandle dataen.
    - presentere ferdig behandlet data til ViewModel
  - Ofte så vil man skille enda litt mer her
    - feks: En klasse for å hente data, en for å presentere ferdig data...



Quick tips for MVVM

Noen fine “enkle” retningslinjer ifht. MVVM:

Ui laget:

1. UI laget: Hver View skal ha hver sin ViewModel

Data laget:

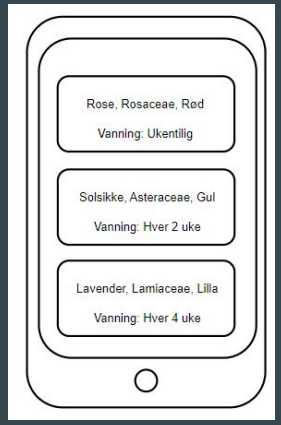
2. Hver egne type data burde behandles i sin egen-allokerte klasse.
3. Om man ønsker å kombinere data fra ulike datakilder, burde dette gjøres i en repository. (repository pattern)
4. Obs: behandling av hentet data, kun i datalaget.

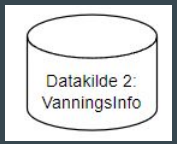
Vi ser på et eksempel!

Eksempel: Plante info app

- Hva skal appen gjøre?
  - Appen skal presentere informasjon om planter og hvor ofte de skal vannes.
- Hvilke type data har vi/trenger vi?
  - Datakilde 1: Informasjon om plantenavn, plantefamilie, farge, tilhørende planteId
  - Datakilde 2: Informasjon om vannings hyppighet, tilhørende planteId
- Hvordan skal vi representere dataen? (Hvordan vil vi presentere dataen til UI-en)
  - Dataklasse: PlanteInfo(planteId, planteNavn, planteFamilie, farge, vanningsHyppighet)
- Må vi gjøre noe med dataen? e.g behandle den / beregninger?
  - Vi må sette sammen dataen fra datakilde 1 og 2
  - Dette kan vi gjøre på planteId.

Vi tegner opp, og følger reglene quick tipsene fra forrige slide

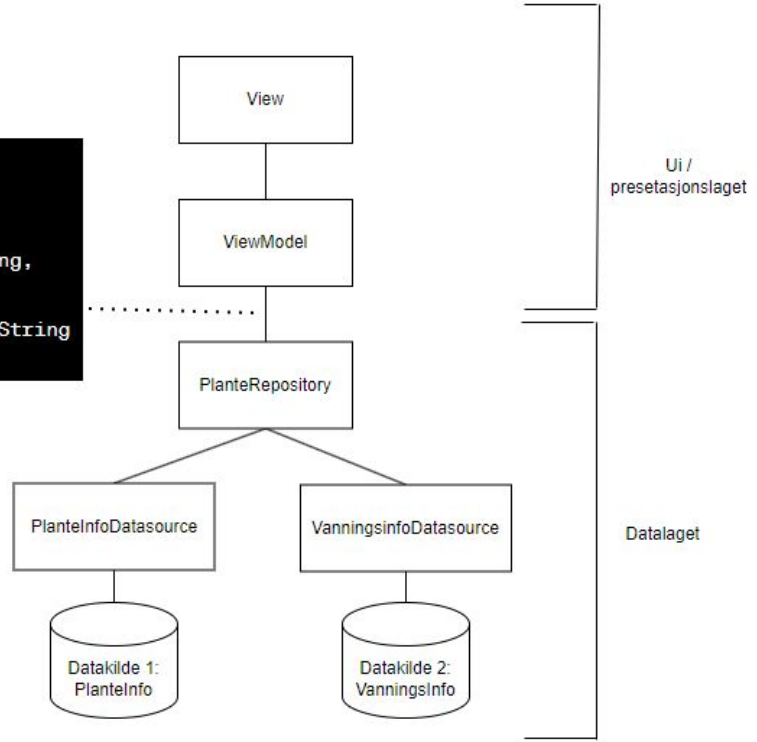





```
data class PlanteInfo(
    val planteId: Int,
    val planteNavn: String,
    val planteFamilie: String,
    val farge: String,
    val vanningsHyppighet: String
)
```

Resultat

```
data class PlanteInfo(
    val planteId: Int,
    val planteNavn: String,
    val planteFamilie: String,
    val farge: String,
    val vanningsHyppighet: String
)
```



Lab