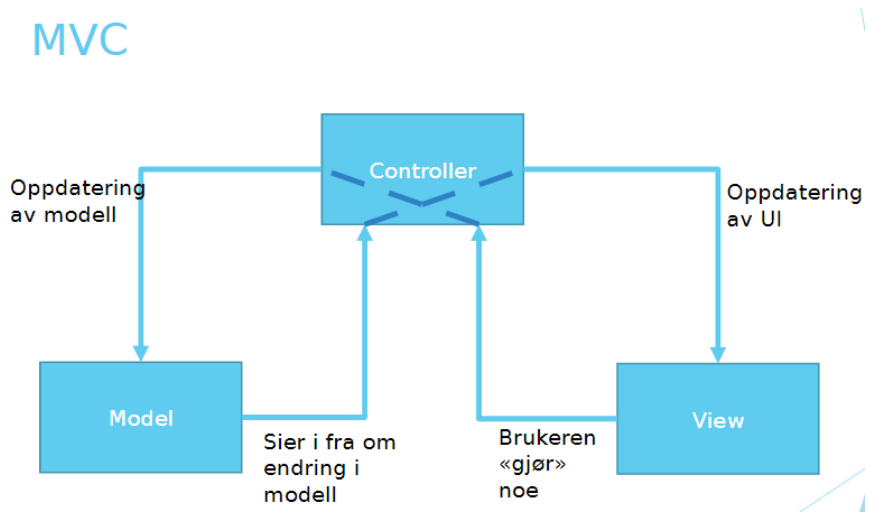
Janaaththan

Javalin: er hovedsakelig et enkelt rammeverk for Java. For å bruke dette må man ha en instans av Javalin-klassen. Etter å ha definert en instans kan man lage pather i applikasjonen til forskjellige sider og bruk. Javalin bruker builder-pattern så man kan unngå konstruktører, og man får muligheten til å kalle metoder etter hverandre.

Vue.js: er et JavaScript-rammeverk som brukes for enkel frontend, og kan enkelt kobles til Javalin. Vue er opensours, og er designet slik at det kan være trinnvis adaptivt til vilkårlige endringer. Hovedbiblioteket har fokus på fronteden, men har fortsatt muligheter for å integrere andre typer av biblioteker.

Anonym klasse: er en nestet klasse som blir dannet for å gjøre koden mer konsis. Hovedfunksjonen er at man kan deklarere og instansiere klassen samtidig. Dette blir for det meste tatt i bruk i forbindelse med å lage implementasjoner av Interface i klassen som tar disse i bruk.

MVC: står for model view controller, noe som er oppbygningen og sammenkoblingen av front og bakenden. Her vil de forskjellige funksjonalitetene bli oppdelt etter sin bruk, noe som lager god struktur og oppbygning.



- **Model:** representerer data som behandles i programmet. Klassene som blir dannet vil representere data som blir tatt i bruk. Inneholder hovedsakelig bare informasjon om modellene.
- **View:** grensesnittet, altså det brukeren ser. Presenterer dataen (GUI). Har muligheten til å forstå om noe blir utført på nettstedet, men kan ikke utføre noe uten hjelpen fra controlleren.
- **Controller:** kodelogikken i programmet. Kommuniserer mellom Model og View, dvs. at den henter ut data fra modellen og sender det til viewet som har bruk for det. Controlleren har et overblikk over alle klassene som er tilgjengelig i koden.

Sammenligning: André Finstad

De første oppgavene på forrige oblig er utført nokså likt. Siden Celestialbody være på toppen av «treet» er de andre klassen extendet til dette. Da har det blitt tatt i bruk super i konstruktørene for å sende med dataen opp. Hovedforskjellen er at jeg hadde Celestialbody som en abstrakt klasse. I de forskjellige metodene lager han nye variabler og bruker det inn i metoden, imens jeg bruker de globale variablene som er tilgjengelig og har noen ikke-dynamiske verdier. Dette burde jeg nok ha endret på slik at alt fungerer dynamisk. Ellers har oppgavene blitt gjort relativt likt grunnet samarbeid, mer har kun litt forskjell på toStringen.