# Begrundelse af valget af 4+1 View Model

I de tidligere semesterprojekter har der altid indgået noget hardware, som der også skulle beskrives i systemarkitekturen for projektet. Til dette semesterprojekt er der derimod kun software, så der er en masse, der førhen er blevet lavet i systemarkitekturen, som ikke skulle laves for dette projekt. Der findes derimod andre modeller, der er gode til at beskrive arkitekturen af meget softwareprægede systemer. Fra skolens side var der lagt op til, at der kunne benyttes modellen, der hedder ”4+1 View Model”.

Modellen består af fire views, der bruges til at beskrive systemet fra forskellige synspunkter. De fire views er logical view, development view, process view og deplyment view. Derudover består modellen også af et sidste view, som omhandler scenarierne i systemet. Disse scenarier beskriver de interaktioner, der finder sted i systemet, og er beskrevet i form af use cases eller user stories.

Det logiske view viser de funktionaliteter, som systemet besidder. Der er her valgt at udarbejde klassediagrammer, da de giver det gode overblik over systemet, samt sekvensdiagrammer, for at vise hvordan kommunikationen foregår i systemet for de forskellige funktionaliteter.

I development view bliver systemet beskrevet ved at blive opdelt i mindre dele, som subsystemer og lag. Her er der valgt at lave package diagrammer, som deler systemet op i mindre pakker, og herved danner overblik over de forskellige dele, der indgår i systemet. Der er også valgt at benytte package diagrammerne til at opdele klassediagrammerne, så afhængigheder imellem klasser i forskellige pakker, og de klasser som hver pakke indeholder, kan vises på en overskuelig måde. Et diagram, der også typisk kan benyttes i dette view, er et component diagram. Det bliver typisk brugt i systemer, hvor komponenter, som andre tidligere har lavet, genbruges og sættes sammen. Dette diagram er fravalgt i dette projekt, da gruppen ikke følte det tilførte nogen værdi.

Process view fokuserer på dynamikken i systemet, og viser de forskellige processer, og hvordan de kommunikerer. Her vil der typisk blive benyttet flow charts til at illustrere dette. Gruppen var dog enige om, at flow charts ikke viste noget, som sekvensdiagrammerne i det logiske view ikke allerede viste. Flow charts er derfor valgt ikke at udarbejde, da sekvensdiagrammerne lettere kan relateres til koden. Selve process view er derfor heller ikke implementeret i arkitekturen, da der ikke var noget, gruppen følte var relevant for projektet at vise her.

Deployment view viser, hvordan softwarekomponenter er distribueret på moduler i det fysiske plan, samt hvordan der kommunikeres mellem disse komponenter. Diagrammet, der benyttes til at illustrere dette, er et deployment diagram. Det er valgt at benytte dette diagram, da det viser, hvordan der kommunikeres mellem applikationerne, der kører på en computer og serveren.