Labrapport Heislab

av Einar Johannes Sommerfeldt og Jacob Dahl

19. mars 2021

Labgruppe 69

TTK4235 Tilpassede datasystemer
Norges Teknisk-Naturvitenskapelige Universitet

Innholdsfortegnelse

1 Arkitektur

Arkitekturen til styringssystemet vårt består av fire moduler: Elevator, Queue, Fsm og Timer.

Elevator er hovedmodulen som eier de andre modulene og implementerer logikken koblet til alt det praktiske: Oppdatering av variable, bruker-input, endring av lys og bevegelse av heisen.

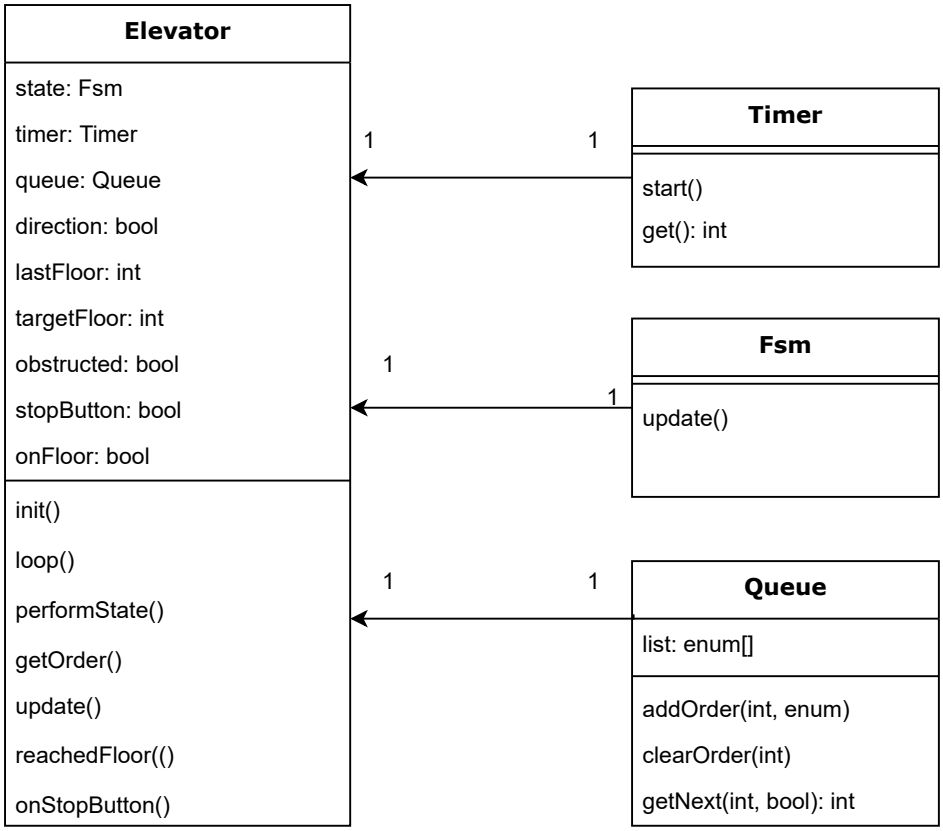
Queue-modulen implementerer heisens kø-logikk, den tar inn nye bestillinger fra Elevator og gir tilbake hvilke bestilling som skal betjenes.

Fsm-modulen oppdateter Elevators state basert på tilstandsvariablene i Elevator. Det er denne som bestemmer hvordan heisen skal oppføre seg og Elevator som utfører dette.

Timer-modulen er en enkel timer som gjør det mulig å holde døren åpen i 3 sekunder. Denne lagrer tiden i Elevator, men kan i utgangspunktet kalles i hvilken som helst modul så lenge modulen selv har tiden timeren ble startet.

1.1 Klasse-diagram

Figur 1.1 viser en oversikt over modulene systemet består av. Fsm står for finite state machine.



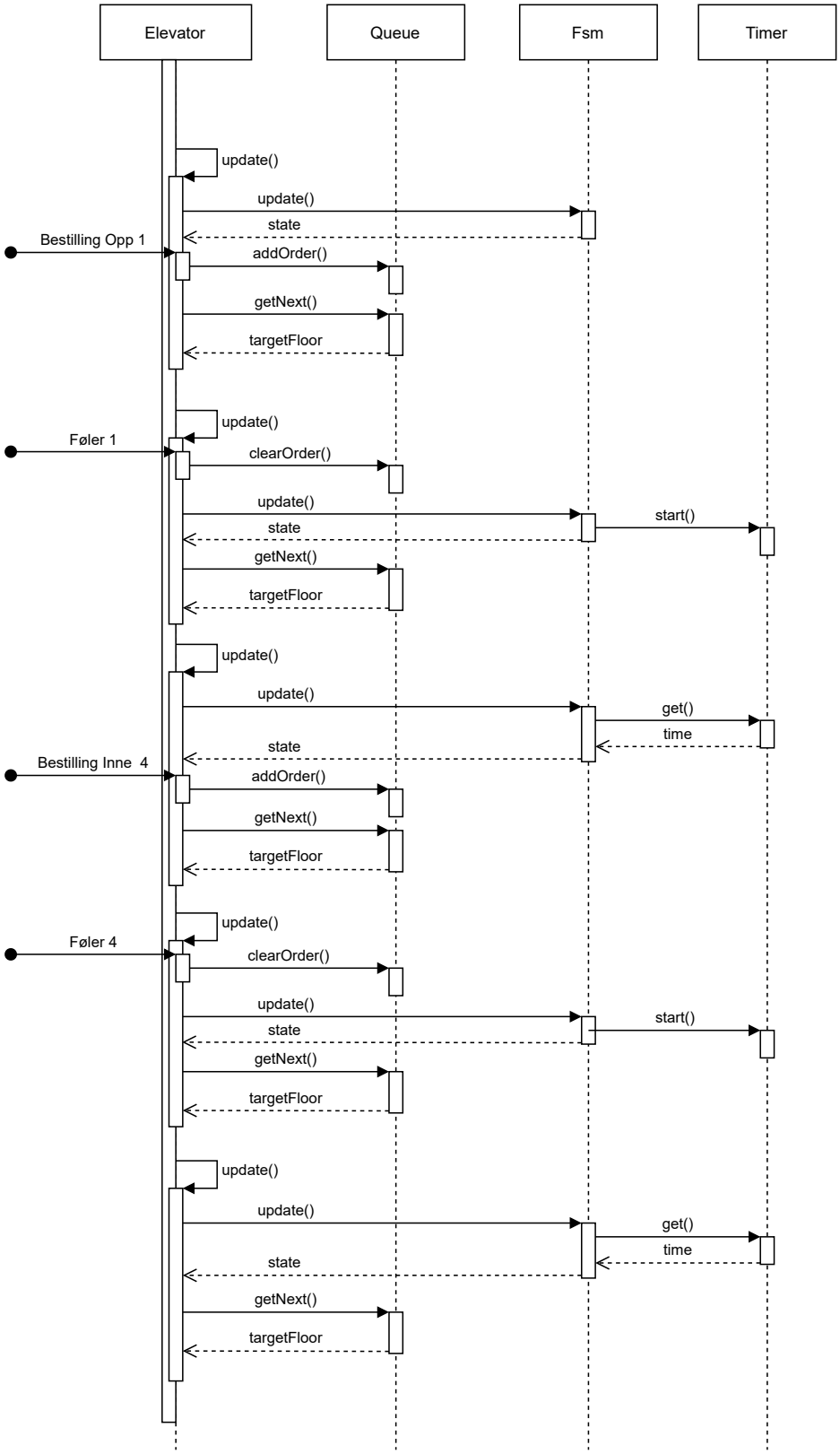
Figur 1.1: Klasse-diagram

1.2 Sekvensdiagram

Figur 1.2 viser et sekvensdiagram som beskriver sekvensen:

1. Heisen står stille i 2. etasje med døren lukket.
2. En person bestiller heisen fra 1. etasje.
3. Når heisen ankommer går personen inn i heisen og bestiller 4. etasje.
4. Heisen ankommer 4. etasje, og personen går av.
5. Etter 3 sekunder lukker dørene til heisen seg.

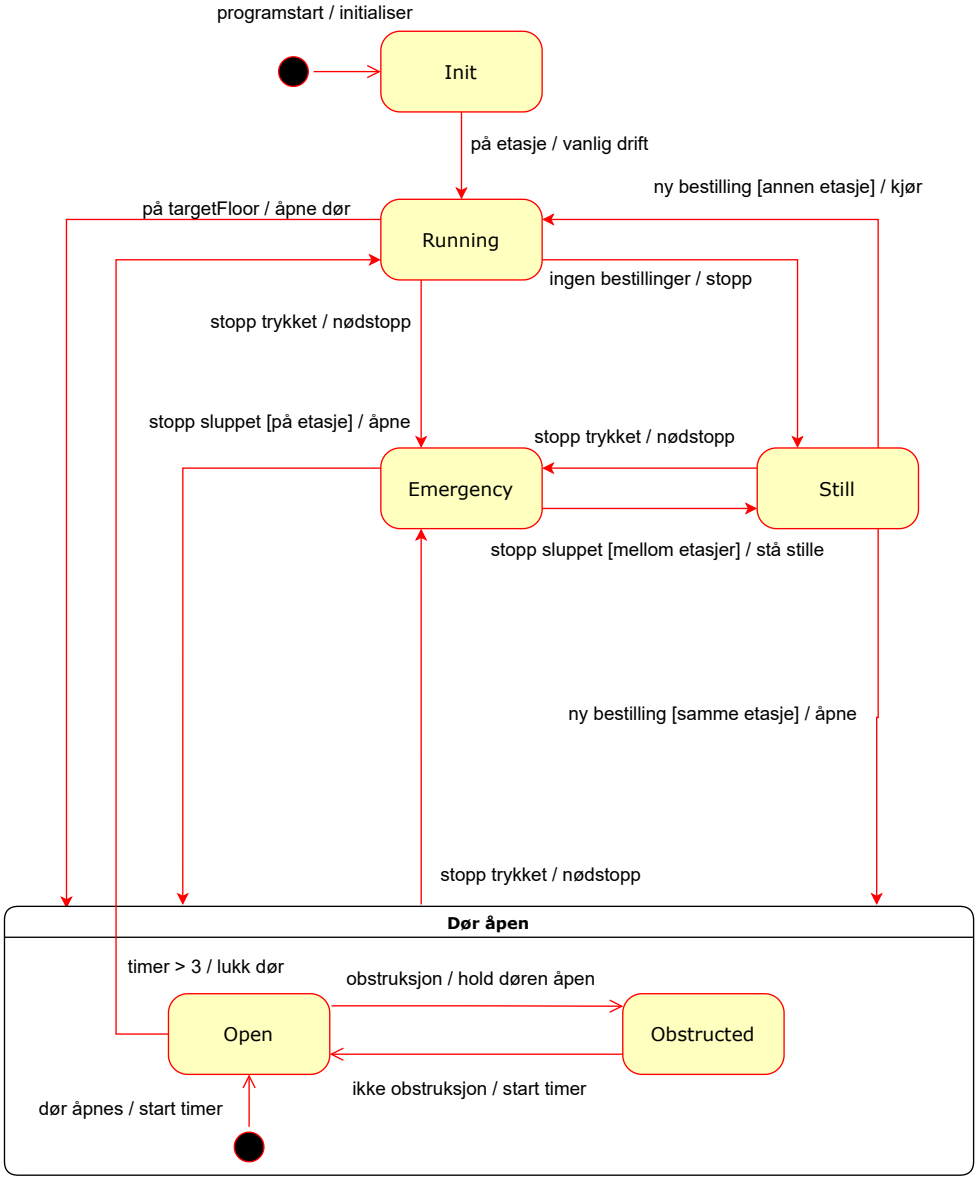
Siden Elevator tar seg av lys og bevegelse internt ble ikke dette tydelig i sekvensdiagramet. Det beskriver derimot fortsatt loopen for å oppdatere tilstandene til heisen godt. Kallene til Timer fra Fsm gjøres når heisen åpner døra. For enkelhets skyld kalles bare get() når døren lukkes i dette diagramet, men i koden vil denne metoden kalles mange ganger til timeren blir 3.



Figur 1.2: Sekvensdiagram

1.3 Tilstandsdiagram

Figur 1.3 viser tilstandsmaskinen fsm-modulen oppdaterer.



Figur 1.3: Tilstandsdiagram

1.4 Argumentasjon

Arkitekturen vår er laget med heisspesifikasjonene i fokus. For å oppfylle disse er det tydelig at man trenger et køsystem for bestillinger og en timer for å styre hvor lenge døra er åpen. En tilstandsmaskin er også hensiktsmessig for å forenkle både skriving av- og forståelse av koden. En sentral Elevator modul som kaller på de andre modulene og implementerer logikken for styringssystemet er praktisk og intuitivt.

Hver modul har tydelig adskilte oppgaver og særlig Timer, Queue og Elevator er svært robuste ovenfor endringer i de andre modulene. Fsm er avhengig av Elevator, men innmaten i Fsm kan endres uten at noen av de andre modulene må skrives om. Alt dette mener vi bidrar til at det ikke ville vært mye hodebry for en programmerer å sette seg inn i systemet og gjøre endringer. Selv om Elevator har et relativt stort ansvarsområde er modulen delt opp i funksjoner som gjør den oversiktlig.

2 Moduldesign

2.1 Elevator

Elevator-modulen er den største modulen i systemet. Denne tar seg av alt som har med selve heisen å gjøre. Den oppdaterer og lagrer alle variabler de andre modulene trenger. I tillegg tar den seg av setting av lys, bruker-input og bevegelse av heisen. Nye bestillinger gir den videre til Queue-modulen og den gjør kall til Queue for å få etasjen den skal bevege heisen til. Heisens bevegelse bestemmes av hvilken tilstand den befinner seg i; tilstanden til heisen bestemmes av Fsm-modulen som Elevator kaller på.

Vi valgte å bake inn bruker-input og lys-styring i denne modulen siden det allerede var funksjoner for dette i den utdelte koden og eventuelle lys- og io-moduler bare hadde blitt et skall oppå dette igjen som ikke ville tilført ny funksjonalitet eller mindre kode. Hadde det vært nødvendig med f. eks memory-mapping for å få til dette ville de vært egne moduler.

2.2 Queue

Queue-modulen er ansvarlig for kø-systemet. Dette er implementert som en liste med en enums, en enum per etasje. Disse beskriver hva slage bestilling etasjen har. Queue bruker retningen til heisen samt siste etasje heisen var på til å bestemme hvilken bestilling som burde betjenes. Implementerte køen som en enkel liste siden det førte til enkel intuitiv koding og passet heisspesifikasjonene.

Modulen har det mest minimale interfaceet med Elevator som vi mener er mulig for at den skal funke. Den er bare avhengig av retning og siste etajse. Man står dermed fritt til å endre Elevator så mye man vil uten å få problemer med Queue.

2.3 Fsm

Fsm-modulen er ansvarlig for tilstandsmaskinen for oppførselen til heisen gjengitt i figur 1.3. Modulen består bare av en funksjon, fsm_update(), som tar inn en peker til Elevator-modulen og oppdaterer tilstanden. Naturligvis er disse to mudulen tett koblet sammen, men vi så dette som den beste måten å implementere modulen på. Ettersom tilstandsmaskinen er avhengig av så mange forskjellige variabler i Elevator er dette nødvendig.

Dersom noen vil videreutvikle tilstandsmaskinen må de være forsiktige hvis man fjerner variabler fra Elevator, men ellers er de to modulene relativt robuste mot endringer i hverandre.

2.4 Timer

Timer-modulen består av en start og en get funksjon. Når vi desginet systemet var vi usikre på om vi trengte flere timere som kjørte samtidig så vi valgte å gjøre slik at Timer ikke lagrer sin egen tid. Tiden lagres i Elevator som kan ha så mange forskjellige tidsvariable den vil som kan settes uavhengig av hverandre.

Det ble til slutt bare nødvendig å lagre en tidstilstand, men vi mener dette fortsatt var et bra valg siden dette gjør det lettere å utvide timer-funksjonaliteten til heisen i fremtiden.

3 Testing

Testingen av systemet ble gjort ved å gå gjennom heisspesifikasjonene og sjekke at hver av disse ble oppfylt av systemet vårt. Vi har dermed bygget opp denne seksjonen slik at den speiler heisspesifikasjonene. Systemet fikk full pott på FAT-testen så dette bygger opp under vår egen testing.

3.1 Oppstart

O1: Ved oppstart skal heisen alltid komme til en definert tilstand. En definert tilstand betyr at styresystemet vet hvilken etasje heisen står i.
Test: Startet heisen mellom etasjer og så at den alltid gikk opp til nærmeste etasje.

O2: Om heisen starter i en udefinert tilstand, skal heissystemet ignorere alle forsøk på å gjøre bestillinger, før systemet er kommet i en definert tilstand.
Test: Startet heisen mellom etasjer og prøvde å trykke bestillingsknappene, disse ble ignorert og ingen lys ble slått på.

O3: Heissystemet skal ikke ta i betraktning urealistiske start.betingelser, som at heisen er over fjerde etasje, eller under første etasje idet systemet skrus på.
Test: Ikke nødvendig å teste dette.

3.2 Håndtering av bestillinger

H1: Det skal ikke være mulig å komme i en situasjon hvor en bestilling ikke blir tatt. Alle bestillinger skal betjenes selv om nye bestillinger opprettes.
Test: Trykket alle mulige bestillinger når heisen var mellom etasjer og døren var åpen. Trykket flere knapper samtidig og alle bestillinger ble alltid betjent.

H2: Heisen skal ikke betjene bestillinger fra utenfor heisrommet om heisen er i bevegelse i motsatt retning av bestillingen.
Test: Bestilte opp i 2. etasje når heisen var på vei ned til 1. etasje fra 4. etajse og tilsvarende når heisen var på vei opp. Heisen ventet alltid med å betjene bestillingene til den var ferdig med bestillingene i retningen den allerede gikk.

H3: Når heisen først stopper i en etasje, skal det antas at alle som venter i etasjen går på, og at alle som skal av i etasjen går av. Dermed skal alle ordre i etasjen være regnet som ekspedert.
Test: Trykket flere bestillinger for samme etasje og så at alle lys ble slukket, brukte GDB og så at køen tømte alle bestillinger når heisen stoppet på en etasje.

H4: Heisen skal stå stille om den ikke har noen ubetjente bestillinger.
Test: Brukte stoppknappen til å fjerne alle bestillinger og så at heisen ble stående til den fikk nye bestillinger selv når stoppknappen var sluppet.

3.3 Bestillingslys- og etasjelys

L1: Når en bestilling gjøres, skal lyset i bestillingsknappen lyse helt til bestillingen er utført. Dette gjelder både bestillinger inne i heisen, og bestillinger utenfor.
Test: Gjorde diverse bestillinger og så at lysene var på til heisen stoppet på den tilhørende etasjen. Satte vi bestillingene manuelt i GDB uten knappetrykk ble lysene ikke skrudd på.

L2: Om en bestillingsknapp ikke har en tilhørende bestilling, skal lyset i knappen være slukket.
Test: Gjorde diverse bestillinger og så at det aldri ble igjen noen lys på etasjer uten bestilling.

L3: Når heisen er i en etasje skal korrekt etasjelys være tent.
Test: Observerte heisen under drift og så at dette alltid stemte.

L4: Når heisen er i bevegelse mellom to etasjer, skal etasjelyset til etasjen heisen sist var i være tent.
Test: Observerte heisen under drift og så at dette alltid stemte.

L5: Kun ett etasjelys skal være tent av gangen.
Test: Observerte heisen under drift og så at dette alltid stemte.

L6: Stoppknappen skal lyse så lenge denne er trykket inne. Den skal slukkes straks knappen slippes..
Test: Trykket og slapp stoppknappen og så at den bare lyste så lenge den var trykket inne.

3.4 Heis-dør

D1: Når heisen ankommer en etasje det er gjort bestilling til, skal døren åpnes i 3 sekunder, for deretter å lukkes.
Test: Gjorde bestilinger i diverse etasjer og så at heisen bare stoppet og skrudde på dørlyset i etasjene med bestillinger i. Tok tiden og så at dørlyset var på i ca. 3 sekunder.

D2: Heisen skal være lukket når den ikke har ubetjente bestillinger.
Test: Gjorde bestillinger og så at heisen lukket seg 3 sekunder etter den hadde betjent siste bestilling og forble lukket.

D3: Hvis stoppknappen trykkes mens heisen er i en etasje, skal døren åpne seg. Døren skal forholde seg åpen så lenge stoppknappen er aktivert, og ytterligere 3 sekunder etter at stoppknappen er sluppet. Deretter skal døren lukke seg.
Test: Trykket stoppknappet mens heisen var i en etajse og observerte at dørlyset var på hele tiden stoppknappen var trykket og 3 sekunder etter den ble sluppet.

D4: Om obstruksjonsbryteren er aktivert mens døren først er åpen, skal den forbli åpen

så lenge bryteren er aktiv. Når obstruksjonssignalet går lavt, skal døren lukke seg etter 3 sekunder.
Test: Skrudde på obstruksjonbryteren når døra ikke var åpen og så at ingenting skjedde. Skrudde på obstruksjonbryteren når døra var åpen og så at den var åpen helt til 3 sekunder etter bryteren ble slått av.

3.5 Sikkerhet

S1: Heisen skal alltid stå stille når døren er åpen.
Test: Lagde nye bestillinger når døre sto åpen og heisen sto stille.

S2: Heisdøren skal aldri åpne seg utenfor en etasje.
Test: Heisen åpnet seg aldri utenfor en etasje i løpet av vanlig drift. Obstruksjonsbryteren påvirket ikke dette.

S3: Heisen skal aldri kjøre utenfor området definert av første til fjerde etasje.
Test: Trykket bestillingsbryteren for 1 og 4 når heisen sto i disse etajsene, dette fikk heisen til å kjøre utenfor området i tidligere versjoner av systemet, men aldri nå.

S4: Om stoppknappen trykkes, skal heisen stoppe momentant
Test: Trykket stoppknappen når heisen var i bevegelse og heisen stoppet alltid.

S5: Om stoppknappen trykkes, skal alle heisens ubetjente bestillinger slettes.
Test: Trykket stoppknappen og så i GDB at alle bestillinger ble fjernet.

S6: Så lenge stoppknappen holdes inne, skal heisen ignorere alle forsøk på å gjøre bestillinger.
Test: Holdt inne stoppknappen og trykekt alle bestilingsknappene, ingen nye bestillinger ble gjort.

S7: Etter at stoppknappen er blitt sluppet, skal heisen stå i ro til den får nye bestillinger.
Test: Stoppet heisen mellom etasjer og så at den ble stående stille til den fikk nye bestillinger selvom den betjente bestillinger før stopp ble trykket.

3.6 Robusthet

R1: Obstruksjonsbryteren skal ikke påvirke systemet når døren ikke er åpen.
Test: Aktiverte bryteren når døren ikke var åpen og heisen åpnet seg aldri før stoppet på en etasje for å betjene en bestilling.

R2: Det skal ikke være nødvendig å starte programmet på nytt som følge av eksempelvis udefinert oppførsel som for eksempel at programmet krasjer, eller minnelekkasje.
Test: Dette har aldri vært nødvendig

R3: Etter at heisen først er kommet i en definert tilstand ved oppstart, skal ikke heisen

trenge flere kalibreringsrunder for å vite hvor den er.
Test: Startet heisen mange ganger og så at den alltid gikk opp til nærmeste etajse og startet vanlig drift.

4 Diskusjon

Vi har kodet et styringssystem som følger spesifikasjonene oppgitt bra. Forbedringspotensialet ligger hovedsakelig i arkitekturen og moduldesignet. Elevator ble en veldig stor modul som kanskje gjorde mer enn nødvendig og gjorde kodingen mer komplisert. Et mer naturlig valg ville vært om tilstandsmaskinen lagret alt av systemets tilstander og at denne eide moduler som styrte lys, bevegelse og brukerinput. Dette ville gjort arkitektruen mer oversiktlig og kanskje gjort systemet lettere å sette seg inn i for en utenforstående. Som nevnt i tidligere gjorde vi valget siden lys, bevegelse og io-funksjonaliteten allerede var programmert inn slik at det føltes som unødvendig tidsbruk å kode nye moduler for det.

Timer-modulen er i klasse-diagramet, og koden, eid av Elevator, men i praksis blir den bare kalt av Fsm-modulen. Dette passer med at Elevator lagrer alle tilstandene til systemet, men det blir strukturelt litt rart. Dette er igjen et resultat av at Elevator har et for stort arbeidsområde.

Når det kommer til kodekvalitet mener vi at systemet er bra skrevet, vi har fulgt mange av retningslinjene gitt i kompendiet og alle funksjonsnavnene er skrevet slik at det er lett å skjønne hva forskjellige deler av koden gjør. Det ble litt få kommentarer ble det litt lite utover dokumentasjonen til doxygen, men det meste av koden er selvforklarende nok til at dette ikke er noe stort minus.