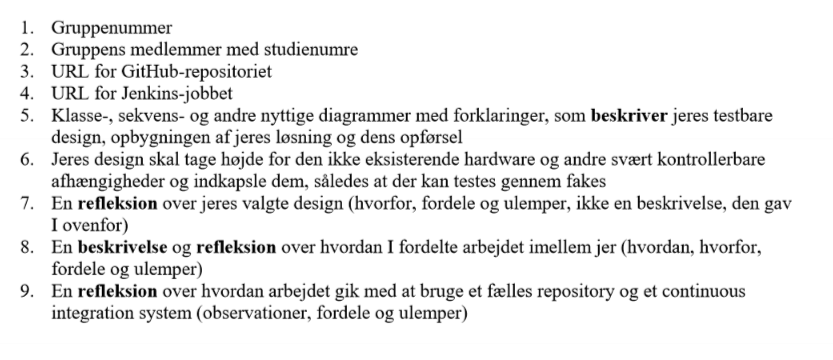
# Handin 2 – Gruppe 50.

|  |  |
| --- | --- |
| **Navn** | **Studienummer** |
| Peter Gehlert Theilgaard | 201907648 |
| Frederik Damgaard Laugesen | 201906694 |
| Jacob Bak Rosenkjær | 201904882 |
| Asbjørn Schöneberg Krogh | 201908477 |

# Handin 2

Indhold

[Handin 2 1](#_Toc67225425)



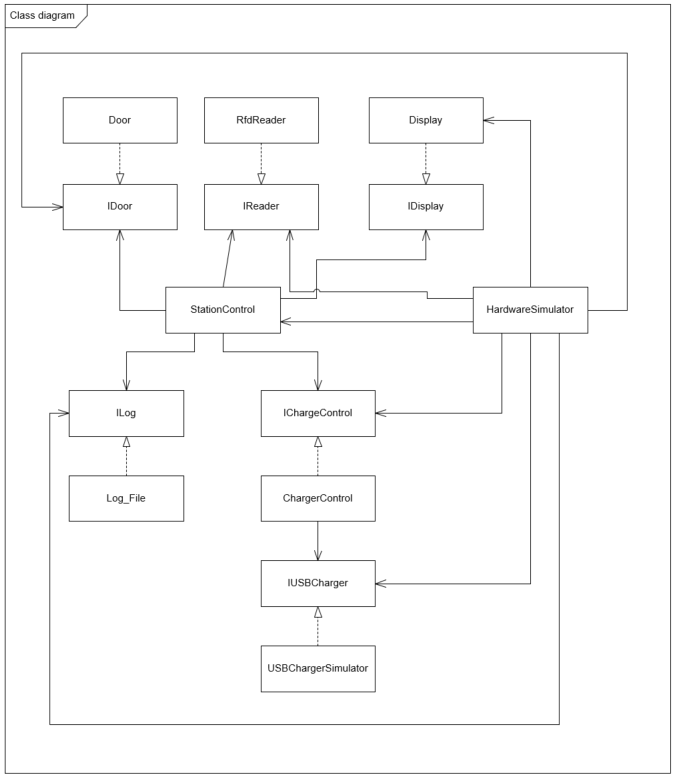
# 3. URL for GitHub-repositoriet

https://github.com/pete7513/Grp50\_HandInTwo

# 4. URL for Jenkins jobbet.

http://ci3.ase.au.dk:8080/job/Grp50\_HandInTwo/

# 5. Klassediagram, Sekvensdiagrammer (Beskrivelse)



Figur 1 Overordnet klasse diagram for systemet

For at gøre systemet mere testbart har vi benyttet os af interfaces alle de steder det har været muligt for at sænke koblingen, og derved gøre systemet mere testbart.

# 6. Design ift. ikke eksisterende Hardware og andre svært kontrollerbare afhængigheder.

Vores program klasse tage højde for det ikke eksisterende hardware i form af døren, RFID readeren og lade stikket. Vi har taget udgangspunkt i det udleverede kode og tilskrevet det ift. de funktioner som vi har ønsket at lade skabet skal kunne.

Altså:

* Åben døren
* Luk døren
* Tilslut lade kabel
* Frakoble lade kabel.
* Læs RFID

Derudover skal systemet kunne simulere en opladning af telefonen, denne kode er dog opgivet på forhånd.

# 7. Refleksion over valgt design.

Under valg af design har vi haft fokus på enkelte dele af SOLID.

Vi har især haft fokus på S – Single Responsibility og O – Open Close Principle.

Single Responsibility har vi benyttet ved at dele ansvaret for de enkelte opgaver ud i mindre klasser, fx har vi oprettet en display klasse som kun har til opgave at udskrive ting på vores ”display”.

Dette er godt i forhold til at gøre systemet mere testbart, da det øger samhørigheden.

Open Close Principle har vi benyttet ved at oprette interfacet til de enkelte klasser så vi får sænket koblingen i vores system og testbarheden derved burde være bedre.

# 8.1 Beskrivelse arbejdets fordeling.

## Asbjørn

Har programmeret dele af door, ChargerControl, Stationcontrol, Program og USBCharger.

Har lavet test på door klassen.

## Frederik

Har programmet dele af Logfile, RFID reader, Display, og stationcontrol.

## Jacob

Har programmet dele af Logfile, RFID reader, Display, og stationcontrol.

## Peter

Har programmeret dele af door, ChargerControl, Stationcontrol, Program og USBCharger.

# 8.2 Refleksion over arbejdsfordeling.

Vi startede med at gennemlæse hele opgaven som gruppe for at danne os et overblik over opgavens omfang, størrelse og evt. udfordringer som kunne opstå.

Vi lavede derefter en grovskitse af arkitekturen, som vi gerne ville have den skulle opbygges ud fra figur 2 i Specifikation af Handin 2 dokumentet. Har tog vi også udgangspunkt i figur 3, sekvensdiagrammet.

Med Grovskitsen på plads og ”Godkendt” af en hjælpelærer, påbegyndte vi programmeringsprocessen.

Peter og Asbjørn kiggede på door, ChargerControl, Stationcontrol, display, Program og USBCharger, mens Frederik og Jacob tog sig af Logfile, RFID reader, Display, og stationcontrol.

Opdelingen af programmet gav god mening for teamet, da ikke alle fire kan side omkring 1 arbejdsopgave. Det er bedre at dele det ud 2 og 2, så en programmerer og den anden sider og følger med og evt. kommer med gode råd og sparring.

Med programmet i mål, kunne test processen begynde. Det kunne have været en fordel at lave test undervejs, men pga. Events var nyt, ville gruppen gerne skrive nogle få test samlet først og dernæst skrive test selv, for at kunne skrive så mange test som muligt hurtigst muligt efterfølgende.

# 9 Refleksion over hvordan arbejdet gik med fælles repository og CIS

|  |  |  |
| --- | --- | --- |
| Observation | Fordele ved observationen | Ulemper ved observationen. |
| Der blev konstant generet en ny SUO fil. |  | Kunne hvergang tilføjes gitignore |
| Hvis StationControl ikke blev gemt ordenligt, blev ændringer fra et andet commit slettet. |  | Det andet commit’s gode skulle så genskrives i stationControl. |
| Vi har ikke benyttet branches ordentligt i det fælles repository. | Næste gang vi opretter et repository kan vi bedre benytte os af forskellige branches. |  |
| Vi har benyttet CI til at overvåge vores dækning af koden. | Vi ved nu hvordan vi kan se hvordan hvor meget af vores kode der bliver berørt i test. |  |

