# Design og implementering af software

I det følgende afsnit beskrives det detaljerede softwaredesign for projektet. Designprocessen, samt hvilke overvejelser og valg der er blevet truffet, vil ligeledes blive beskrevet.

For uddybende beskrivelser af design og implementering af software, henvises til dokumentationen (REFERENCE)

Identifikation af ønskede softwaredele:

I prototypen af vores produkt er der kun implementeret vejen fra hovedmenuen og ned til oprettelse af en ny sensorkonfiguration

**Grafisk brugeroverflade (GUI) – Rock**, bestående af grafisk platform med følgende afgrænset indhold:

* Hovedmenu: Generelt overblik over de mulige konfigurationer  
   Herunder: Sensorer, Presets og Lydpakker
* Sensorer: Menuen for mulige konfigurationer af sensorer   
   Herunder: Ny sensorkonfiguration, Rediger sensorkonfiguration og Slet sensorkonfiguration
* Ny sensorkonfiguration: Muligheden for at konfigurere indstillingerne for en ny sensorkonfiguration
* messageHandler: Denne klasse håndtere beskederne mellem GUI klasserne og Controller klassen

**Controller klassen – Rock,**

## GUI

### Hovedmenu

Hovedmenuen er implementeret som klassen mainWindow. Der oprettes en instance af denne klasse i vores main program. Inde i mainWindows konstruktor oprettes det grafiske UI som danner rammerne for at vi kan designe diverse elementer i UI’et, såsom knapper, dropdowns og lignende.

Brugeren bliver i hovedmenuen præsenteret for følgende tre valgmuligheder:

1. Sensorer Danner en instans af sensorWindow
2. Lydpakker Danner en instans af lydpakkeWindow
3. Presets Danner en instans af presetWindow

Alt efter brugerens valg dannes det grafiske ui for den tilhørende klasse.

For yderligere beskrivelse af Hovedmenuens funktionalitet, henvises der til dokumentationen (REFERENCE)

### Sensorer

Efter brugeren har valgt menu punktet Sensorer dannes der en instance af klassen sensorWindow, der har samme funktionalitet i konstruktoren som mainWindow.

Brugeren bliver i Sensorkonfigurationer(instancen af sensorWindow) præsenteret for følgende tre valgmuligheder:

1. Ny Sensorkonfiguration
2. Rediger Sensorkonfiguration
3. Slet Sensorkonfiguration

Alt efter brugerens valg dannes det grafiske ui for den tilhørende klasse, og der sendes besked til messegeHandler klassen efter den nødvendige info, da der tages udgang i prototypen er det Ny sensorkonfiguration der er implementeret.

### Ny sensorkonfiguration

Efter brugeren har valgt Ny Sensorkonfiguration, dannes der en instance af klassen nySensorkonf, der har samme funktionalitet i konstruktoren som de ovenstående klasser. Derudover kaldes en funktion der ligger i klassen messageHandler, der efterspørge den nødvendige information der skal bruges til at oprette en ny sensorkonfiguration. Denne information er følgende:

1. Sensor type
2. Mapping scheme
3. Akse
4. Lydpakke

Sensor type og akse er ”hardcoded” i prototypen, da der ikke er implementeret mere end en type sensor og der kun er tre mulige akser (x, y, z), de to andre informationer hentes gennem messegeHandleren’s funktion getSensorKonfInfo, der henter en struct med lister af strings med navne på de Mapping schemes og Lydpakker der ligger i Datastorage (Klassen der håndtere vores gemte data). Derved dannes der et ui med mulighed for at vælge den information vi skal bruge til at oprette en ny sensorkonfiguration.

BILEDE!!!!!!!!

Efter brugeren har valgt sin ønskede Sensorkonfigurationer har han muligheden for at gemme disse ved anvendelse af en gem knap. Når denne anvendes sendes de fire valgte informationer til messageHandleren der sender det videre til Controller klassen der gemmer informationen i Datastorage.

Den nyoprettede Sensorkonfiguration er nu klar til at blive benyttet i den yderligere del af systemet.

For yderligere information om GUI klasserne henvises til dokumentationen (REFERENCE)