Projekt daglig rapport

# Dag 1

Benjamin og jeg har siddet sammen og prøvet at få lagt en plan, over hvilken rækkefølge, samt hvilke måder vi laver tingene. Vi har siddet med Arduino og bread board for at få lys til at blinke og [termometer måler](https://create.arduino.cc/projecthub/pibots555/how-to-connect-dht11-sensor-with-arduino-uno-f4d239) til at (næsten) smelte. Benjamin og jeg har begge fint styr på API og har derfor valgt at fokusere på arduinoen til at starte med. Vi sørger for at vi begge forstår, hvordan vi får forbundet den til en API. Vi løb lidt ind i problemer med at finde et wifi modul, og da vi så fandt et, var det tid til at finde ud af at få den på wifi.

# Dag 2

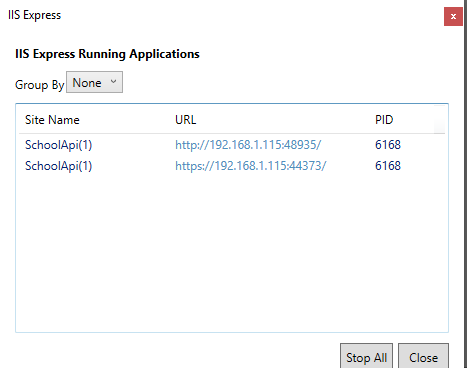
Dag 2 starter hårdt ud med at, Benjamin og jeg, blev enige om at vi skulle have en rigtig router. Grundet det ikke ville fungere, hvis vores ip adresse skiftede sig. Det var dog ret bøvlet med hvordan det skulle gøres, i forhold til efifs netværk. Benjamin og Mikkel sad og prøvede at få sat NAT op, mellem efif, og vores lokale netværk. Imens sad jeg og fik udarbejdet et rigt billede, samt kiggede med, over deres skuldre. Efter 1½-2 timers fejlfinding og en genstart senere, så kom vi begge på nettet uden problemer.

Benjamin var i stand til at forbinde til min api over wifi uden problemer, men arduinoen ville ikke.

Efter at have prøvet flere forskellige løsninger, er vi kommet frem til at vi vil køre det over ethernet i stedet. Vi har prøvet at fejlfinde, også på flere forskellige libraries. Vi har også prøvet at udskifte vores arduino, hvilket hjalp på det daværende problem, men ikke kunne løse det næste i rækken. Vi vil prøve at sende get/post requests over ethernet i stedet for nu.

## Dag 3

Da vi mødte ind I morges, opstod der en ”**unable to connect to web server iis express**” fejl. Denne her tog lidt tid at fikse, men vi gik ind i IIS Express menuen nede i hjørnet. Her kunne vi finde alle tidligere og nuværende porte samt ip adresser vi brugte. Vi kom igennem til api’en med arduinoen, da vi slog https redirects fra, inde i vores startup.cs fil. Vi har også valgt hvilke data vi vil måle på; luft fugtighed, bevægelse, lys og temperatur. Vi brugte det sidste af dagen på, at teste hvordan de forskellige komponenter udstillede dataen.



# Dag 4

I dag har vi begge siddet hjemme og lavet hvert vores research. Vi havde den forhenværende dag, siddet og aftalt hvilke emner, vi skulle undersøge. Jeg har siddet med videoer omkring entity framework, og Benjamin med angular/image mapping.

# Dag 5

Benjamin og jeg har siddet med fordelt arbejde i dag. Jeg har siddet med at lave et klasse diagram, mens Benjamin har lavet et mockup af hjemmesiden. Benjamin har også kigget på, og fået image mapping til at virke. Benjamin har også fået det til at være skalerbart.

Jeg har fortsat prøvet at få entity framework til at lave en normaliseret database. Jeg har derfor også lavet et E/R diagram, for at have et klart mål, at gå efter.

# Dag 6

Vi har nu mulighed for at poste data til vores api, fra arduinoen. Vi har også valgt at fjerne motion detektoren igen. Dette er grundet at den ville skulle have sin egen tråd, hvis den skulle gøre optimalt. Hvis vi delayer med 2 minutter, men den opfanger bevægelse inden i den sovende tråd, hvordan får vi det så ud? Dette ville blive en større logik som skulle udføres på arduionen, som vi har taget beslutningen sammen om at droppe. Vi har fået alle 3 komponenter på, og kan sende det over et post request, med querystring.

# Dag 7

Vi har i dag arbejdet hjemme, så vi har lavet tingene hver for sig. Benjamin har kigget på Angular, hvordan sidens layout skulle se ud, samt rode rundt og finde ud af angulars struktur.

Jeg har videre prøvet at fejlfinde på mit entity framework rod. Jeg har søgt omkring emner som, shadow properties / entities, relationships in entity framework og linking foreign key to non-primary key.

# Dag 8

I dag har vi Benjamin prøvet at hjælpe mig med at få opsat entity framework. Vi ændrede i vores DataEntry model, så den ikke længere bruger lister, samt har ID’erne på de relaterede klasser. For at få entity framework til at fungere, måtte vi blive nødt til, at gøre sådan. I mellemtiden, var jeg begyndt at sætte databasen op selv. Dette skrottede jeg hurtigt igen, da Benjamin og jeg besluttede, at bruge entity framework, da det nu virkede. Jeg har derefter oprettet Save og Get funktioner, som virker op i mod, den rigtige database. Jeg har rettet lidt i klassediagrammet.

# Dag 9

Vi har finaliseret elementer. Arduionen poster med delay dens data. Der er blevet oprettet en trigger til at slette gammel data fra databasen. Vi kan hente ting fra api’en til front-end i angular og dokumentation er blevet videreudbygget.

# Links

## Shadow properties

<https://www.entityframeworktutorial.net/efcore/shadow-property.aspx>

<https://stackoverflow.com/questions/47648487/ef-core-how-to-add-the-relationship-to-shadow-property>

<https://docs.microsoft.com/en-us/ef/core/modeling/shadow-properties>

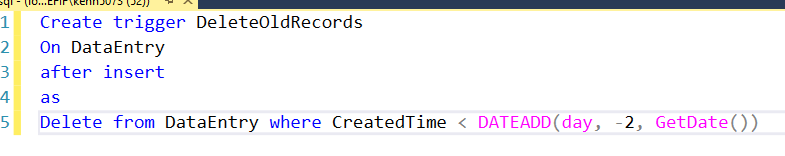
<https://stackoverflow.com/questions/41987847/entity-type-type-is-in-shadow-state-a-valid-model-requires-all-entity-types-t>

## Entity framework relations

<https://docs.microsoft.com/en-us/ef/core/modeling/relationships?tabs=data-annotations%2Cfluent-api-simple-key%2Csimple-key>

<https://gavilan.blog/2019/04/14/entity-framework-core-foreign-key-linked-with-a-non-primary-key/>

## Database Trigger



## Router configuration

Linksys router setup

SSID: WonderfulHippos

Pass: WonderfulHippos

DHCP: Off

## Data vi kan hive ud

**PIR (Motion detector) (not used)**

Motion detected = int (not used)

Motion ended = int (not used)

Temperatur & humidifier = float

Photoresistor = int