



SOFTWARE REQUIREMENT & DESIGN DOCUMENT

Home Automation System

Nickolas Helgeland, Jon Erik Nordskog og Kristian Sande Sjølyst

Innholdsfortegnelse

1.	Introduksjon.....	2
2.	Overordnet systemskisse.....	3
2.1.	SensorGetApp	3
2.2.	Database	3
2.3.	WebApp	3
3.	Eksempel på plantegning.....	4
4.	Tekniske Krav	5
4.1.	Funksjons krav.....	5
4.2.	Ikke-funksjonelle krav	5
5.	GUI-skisse.....	6
5.1.	Hovedsiden	6
5.2.	Historiesiden	7
5.3.	Innstillingssiden.....	7
6.	Liste over hardware	9
7.	Programvare	10
8.	Databasediagram.....	11
9.	Sikkerhet og GDPR	12
9.1.	Sikkerhet	12
9.2.	GDPR	12
10.	UML	13
10.1.	Use Case-diagram	13
10.2.	Sequencediagram	13
10.3.	Klassediagram	14
11.	System arkitektur	15
12.	IO-enhet	16

1. Introduksjon

Den teknologiske utviklingen går fort. De fleste har en rekke smarte enheter allerede. Bilen er et eksempel på noe har fått utvidet funksjonalitet de siste årene. Når du setter deg inn, kan bilen koble seg til telefonen din og starte spillelisten der du stoppet sist. Når du kommer veistrekningen der du hevet kjørehøyden sist vil den gjøre det når du kommer dit neste gang. Når du stiller enn en ønsket temperatur til kupeen på vinteren i bilen vil den oppnå temperaturen ved å regulere varmpumpe og varmeelement. Luftstrømmen reguleres og styres basert på utetemperatur, kupetemperatur og luftfuktighet. Alt for å øke komfort, sikkerhet og energieffektivitet.

Hvorfor skal bilen ha så mange muligheter, men ikke hjemmet ditt? Det er tross alt her vi tilbringer mest tid og også her det kan spares mest energi.

Home Automation System vil gi kunden mer kontroll over hjemmet sitt. Systemet øker sikkerheten og komfort i hverdagen. Hendelser skal kunne programmeres slik at f.eks. alle lys i huset skrues på når du kommer hjem fra jobb. Systemet skal også hjelpe kunden med energisparing ved å sørge for at huset er varmt når det er i bruk og senke temperaturen på natten og når ingen er hjemme. Alt loggføres i en database som kun kunden har tilgang til. Og alt skal kunne styres og kontrolleres fra hvilken som helst nettleser.

2. Overordnet systemskisse

Systemet er delt opp i tre deler. SensorGetApp er programmet som kjører hjemme hos brukeren. Her leses verdier fra sensorene og sender dette til databasen. Databasen og web serveren ligger i skyen. Fordelen ved å ha en skyløsning er at web serveren og databasen er lettere å gjøre tilgjengelig over internett og databasen. Oppdateringer vil også kunne gjøres enklere og være umiddelbart tilgjengelig for alle brukere. Energibruken vil også kunne holdes på et minimum ved skyløsningen.



Figur 2.1 Viser systemets oppdeling i dets 3 deler.

2.1. SensorGetApp

Dette er programvaren, kalt **SensorGetApp**, som er installert hos kunden. Denne leser det elektriske signalet fra transdusere og kalkulerer denne om til en verdi som f.eks. celsius og sender denne dataen til databasen.

2.2. Database

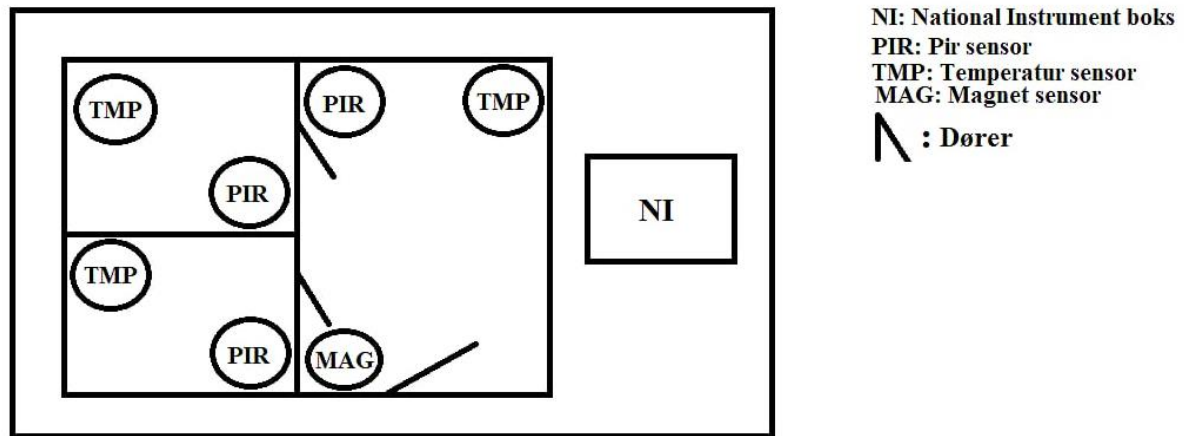
SQL databasen, kalt **databasen**. Her lagres alle data fra sensorene samt innstillinger og informasjon om brukere og dets eiendommer. Her er last- og stresstest sentralt. Tåler databasen utvidelser og høy belastning med mange samtidige brukere og mange sensorer?

2.3. WebApp

Programvare som leser fra databasen, kalt **WebApp**, viser data på websiden til brukerne. Fra websiden skal det også kunne gjøres endringer i innstillinger.

3. Eksempel på plantegning

Figur 3.1 viser et eksempel på plantegning til Home Automation System installert i en leilighet. Her er det satt opp temperatur-, fukt- og bevegelsessensorer i hvert rom. I tillegg er det en magnetsensor på ytterdøren.



Figur 3.1, Skisse av Home Automation System

Home Automation System lager systemet slik at det skal kunne brukes av større bedriftskunden som har flere eiendommer med flere bygg og et større antall rom med til sammen flere hundre sensorer.

4. Tekniske Krav

Alle kravene er etterfulgt av et tall (1), (2) eller (3). De betyr følgende:

- (1) dette er et krav til RTM
- (2) dette kan og bør være et krav til RTM dersom vi har tid.
- (3) dette er funksjonalitet som vi ser for oss i versjon 2.

4.1. Funksjons krav

Systemet skal:

- Hente inn informasjon fra sensorene.
- Lagre data fra sensorene i databasen.
- Viste oppdatert data fra sensorene.
- Kunne vise data fra
 - a. Temperatur (1)
 - b. Magnet sensorer (2)
 - c. Temperatur historikk grafisk og i tabell (2)
 - d. PiR, bevegelsessensor (3)

4.2. Ikke-funksjonelle krav

Bakgrunnen av systemet:

- Kobling mellom sensorer og databaser.
- Kobling mellom database og GUI.
- Bruke ekstern Azure server.

5. GUI-skisse

Programvaren skal kjøres på en datamaskin med Windows. Sensorene er koblet til en DAQ fra National Instruments med USB. GUI skal vises i en nettleser. I versjon 2.0 skal programvaren og databasen ligge på en sentral server. Data fra sensorene hos kunden skal kommunisere til serveren og databasen over internett.

Home Automation System skal ved hjelp av temperatur-, magnet- og bevegelsessensorer kunne:

- Vise temperaturen på forskjellige rom og steder som stue og kjøleskap. Aktuell temperatur skal vises på hovedsiden og historikk skal vises i grafer på en annen side.
- På hovedsiden skal også alarmen kunne aktiveres og deaktiveres. På en annen side skal historikken fra alarmen kunne vises i et skjema. Historikk inkluderer hendelse som aktivering, deaktivering og utløstalarm og tidspunktet for disse.
- En side skal ha innstillinger som å legge til nye sensorer og rom.

På hver side skal det være en tabulator med tre faner for å vise de forskjellige sidene:

- Hovedside
- Historieside
- Innstillingsside

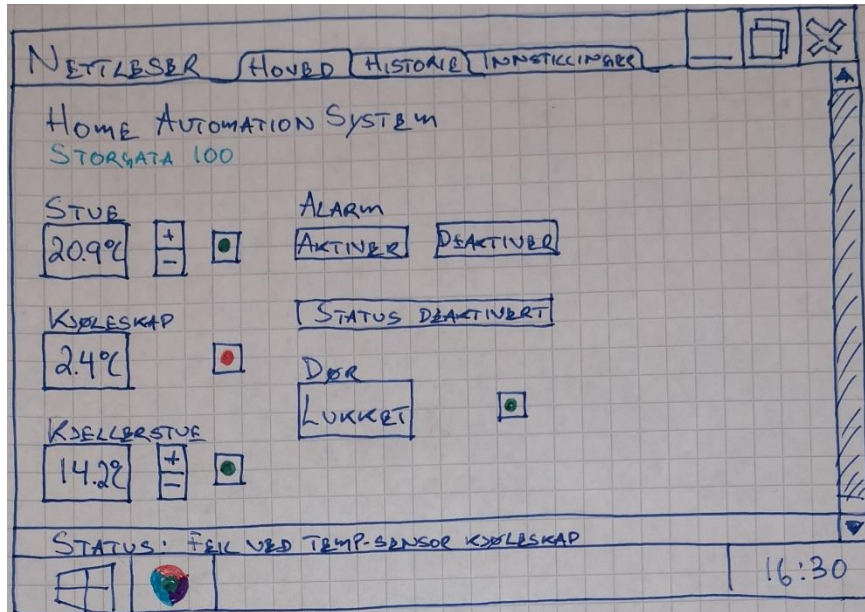
5.1.Hovedsiden

Hovedsiden viser aktuelle temperaturer og statuslys på hver sensor som indikerer om sensoren er tilkoblet og ok. I versjon 2.0 skal det være mulig å stille inn ønsket temperatur i rommet som kommuniseres til aktuell termostat eller regulator.

Alarmen skal kunne aktiveres og deaktiveres og vise aktiveringsstatus. Sensorene skal vise om dører er lukket eller åpne og om det registreres bevegelse på bevegelsensorene. Hver sensor har hvert sitt statuslys som viser om sensoren er ok

Nederst finner du statusbaren som viser overordnede beskjeder fra systemet.

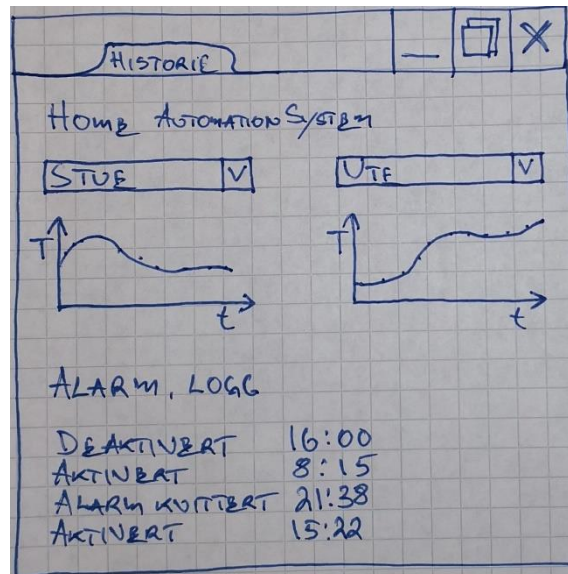
Versjon 2.0 skal ha mulighet til å administrere flere eiendommer.



Figur 5.1 Hovedside

5.2. Historiesiden

Historiesiden skal vise historikken fra sensorene i grafer. Logg fra alarmdelen skal vises i en liste.



Figur 5.2 Historieside

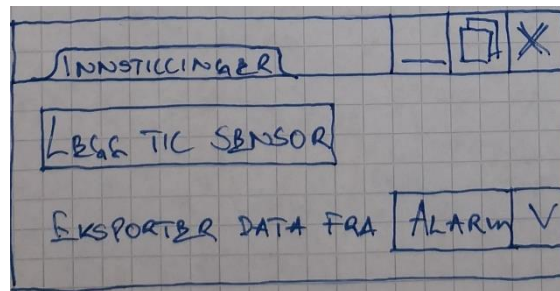
5.3. Innstillingssiden

Innstillingssiden skal ha mulighet til å eksportere en logg fra valgt sensor. En knapp skal kunne nullstille databasen.

Versjon 2.0 skal i tillegg ha følgende funksjonalitet:

- Mulighet til å legge til nye sensorer. Når brukeren trykker på «Legg til sensorer» skal det starte en veiledning for å legge til en ny sensor til systemet. 2

- Status på betalt lisens. Vi ser for oss at systemet er gratis å bruke med inntil 5 sensorer og at kunden må betale for en månedlig, årlig eller evig lisens om det skal brukes med flere enn 5 sensorer.



Figur 5.3 Innstillingsside

6. Liste over hardware

- Alle sensorene er koblet til en DAQ til USB-enhet fra National Instruments, Modell: 6008.
 - Sensorer
 - Temperatursensor.
 - Magnetbryter som brukes for å registrere om dører og vinduer er åpne eller lukket.
 - Bevegelsessensor, PIR-sensor
 - Fuktsensor.
- Bærbar PC.
 - USB for tilkobling til DAQ.
 - Wi-Fi for tilkobling til internett

Hvilke spesifikke leverandører og komponenter velges senere og blir dokumentert i systemdokumentasjonen.

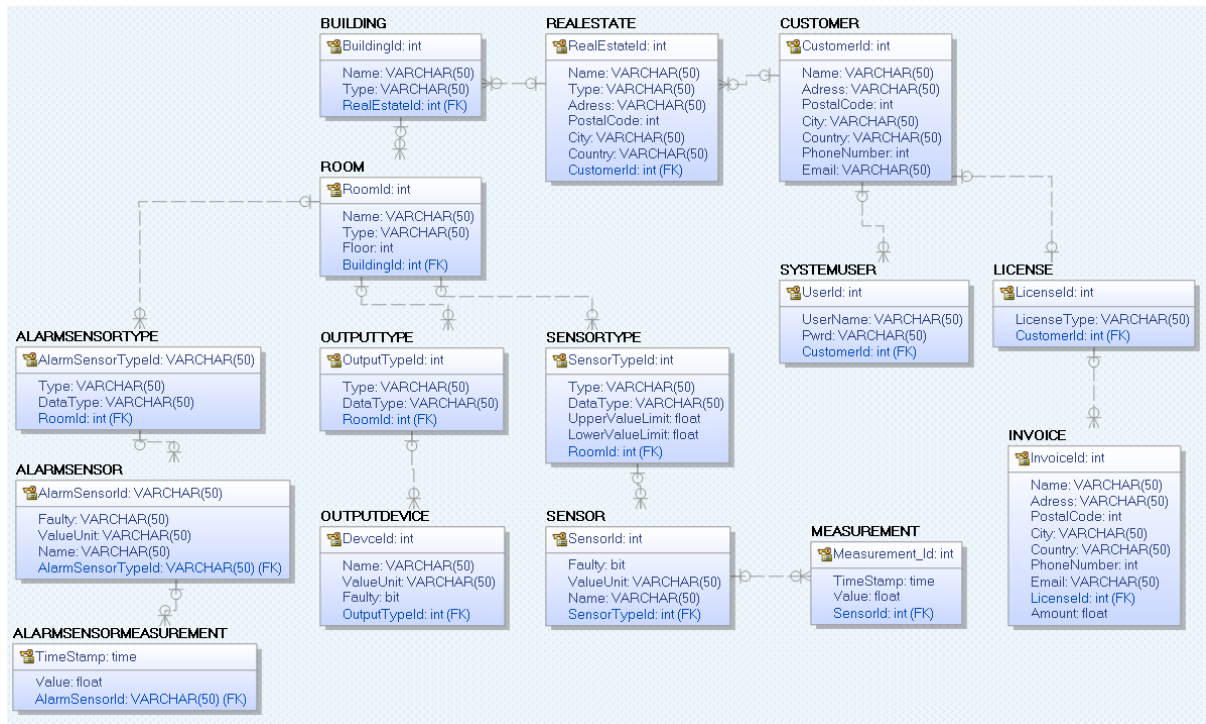
7. Programvare

Følgende programvare ser vi for oss er nødvendig når sluttproduktet skal brukes.

- OS til server: Microsoft Windows Server.
- OS til bruk av SensorGetApp hos kunden: Microsoft Windows 10.
- Microsoft SQL Express Server.
- Drivere til DAQ fra National Instruments.
- Microsoft Internet Information Services (IIS).
- Home Automation System sine applikasjoner.

8. Databasediagram

Vi har brukt Erwin Data Modeler til å lage databasemodellen, Figur 8.1. Her har gått ut fra at vi har en kunde som har en eller flere eiendommer. Denne har en eller flere bygninger og igjen har et eller flere rom. I rommene har vi en eller flere sensortyper som igjen har en eller flere sensorer av samme type. Hver sensor har hver sin tabell som lagrer målingen og tiden målingen ble utført. På outputsiden har vi tabellene for ting som ønsker å styres som lys og varme.



Figur 8.1 Erwin databasemodell

9. Sikkerhet og GDPR

9.1.Sikkerhet

Denne versjonen har en del kjente sikkerhetsutfordringer. Verken databasen eller noe av kommunikasjonen er kryptert som betyr at data, brukernavn og passord lett kan avsløres og komme på avveie. Home Automation System må derfor anses som åpent og at all informasjon tilgjengelig.

9.2.GDPR

Fordi Home Automation System er basert omfattet av GDPR av følgende årsaker:

- HMS er lokalisert innen EØS.
- Data samles inn fra bygninger lokalisert innenfor EØS.
- Programmet er utviklet med hensikt å generere en inntekt fra borgere innen EØS.
- Data som samles ses på som overvåkning.

GDPR krever¹ bl.a. at data som samles inn anses som privat og ikke åpent for andre som standard. Data må krypteres for å hindre i størst mulig grad for andre enn den som eier dataene å aksessere dem. Informasjon om brukere skal holdes til et minimum. Kunden og brukere må informeres og godta at programmet hentes inn og samler data før det kan tas i bruk. Og sist, men ikke minst så må det dokumenteres at GDPR er tilfredsstilt.

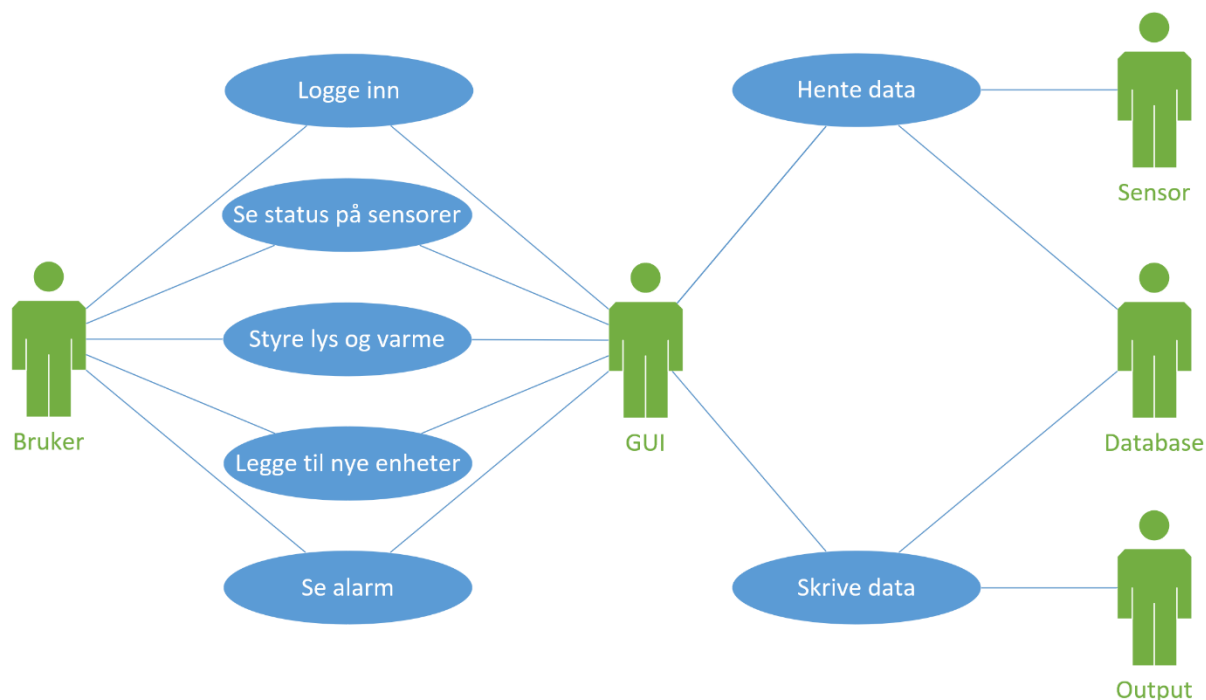
¹ <https://indatalabs.com/blog/data-strategy/guide-gdpr-compliant-software-development>

10. UML

10.1. Use Case-diagram

Use case-diagrammet beskriver typiske oppgaver en bruker ønsker å gjøre med Home Automation System.

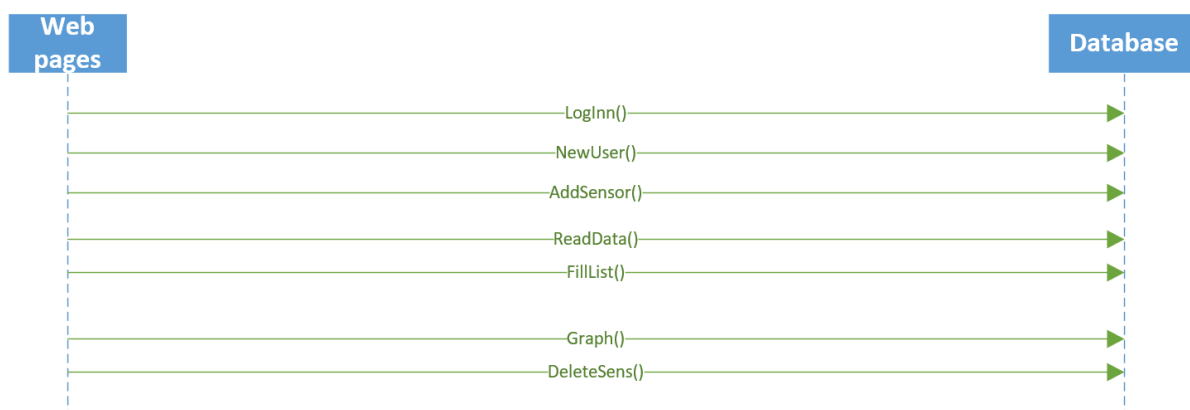
Et eksempel er en bruker som ønsker å slå på et lys. Brukeren bruker GUI som skrur på lyset og skriver hendelsen i databasen.



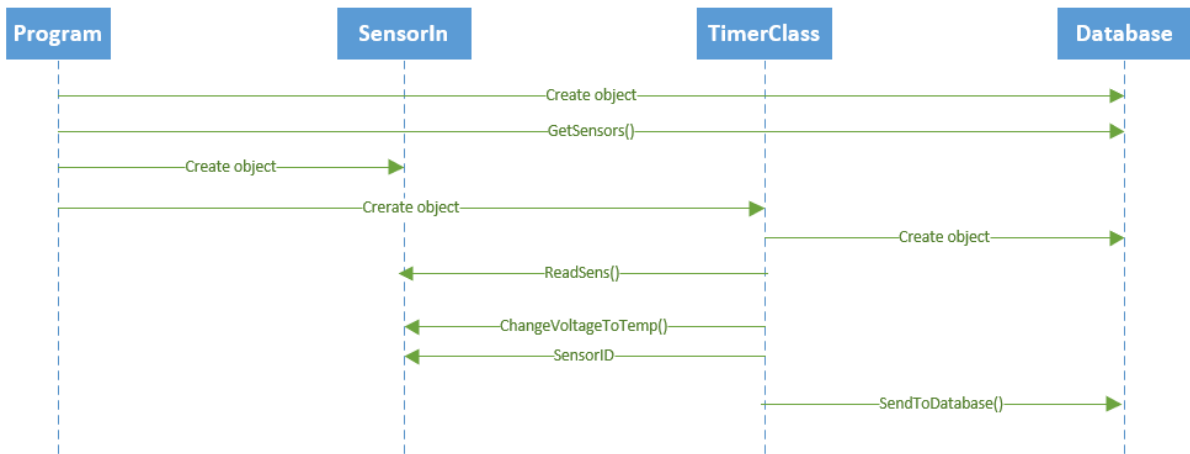
Figur 10.1 Use case-diagram

10.2. Sequencediagram

Figur 10.2 Sequencediagram viser dataflyten mellom objekter og deres metoder i WeApp.



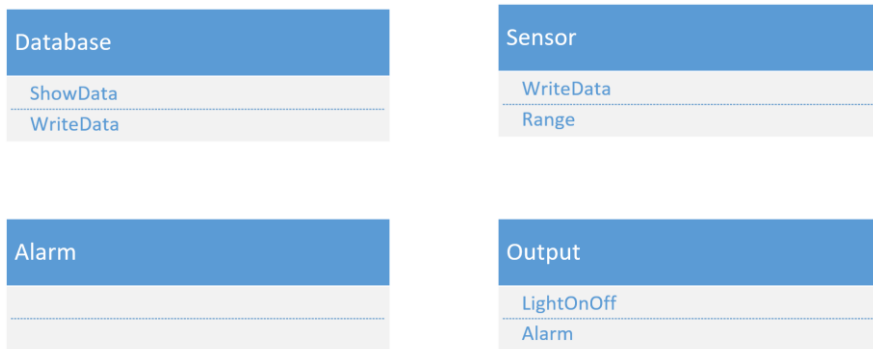
Figur 10.2 Sequencediagram til WebApp



Figur 10.3 Sequencediagram til SensorGetApp

10.3. Klassediagram

Figur 10.4 viser en oversikt over klassene ut fra sequence-diagramet.

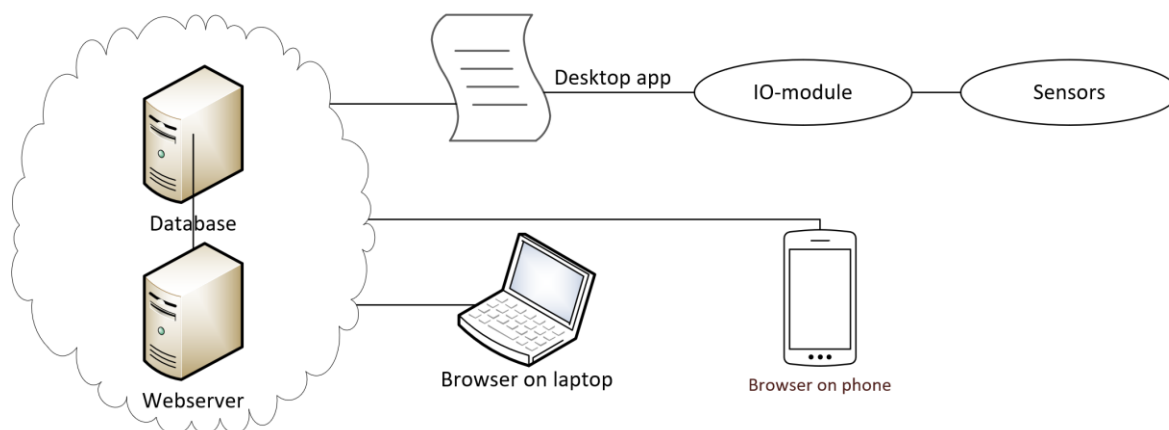


Figur 10.4: Klassediagram

11. System arkitektur

Systemet er basert på to programmer, desktop app og websiden. I tillegg brukes en database.

Ulike sensorer er tilkoblet en IO-modul som igjen er tilkoblet en bærbar PC². Programmet desktop app (SensorGetApp) kjører her og kommuniserer via Wi-Fi til databasen som er i sky. Når en klient logger seg inn på webserveren henter den all nødvendig data og historikk fra databasen og viser dette på en webside.

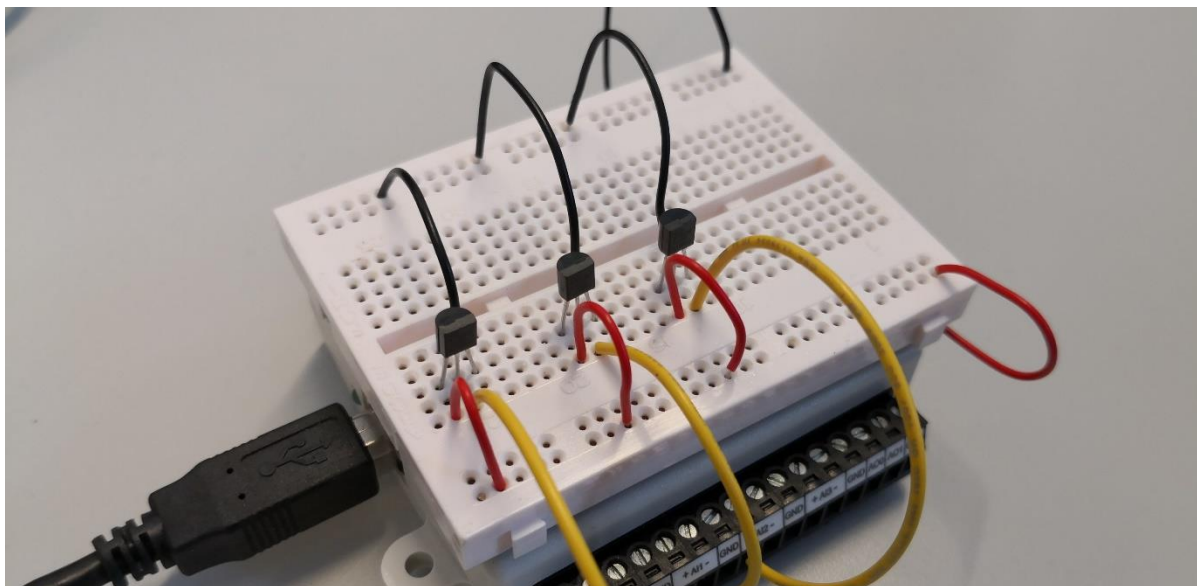


Figur 11.1: System arkitektur

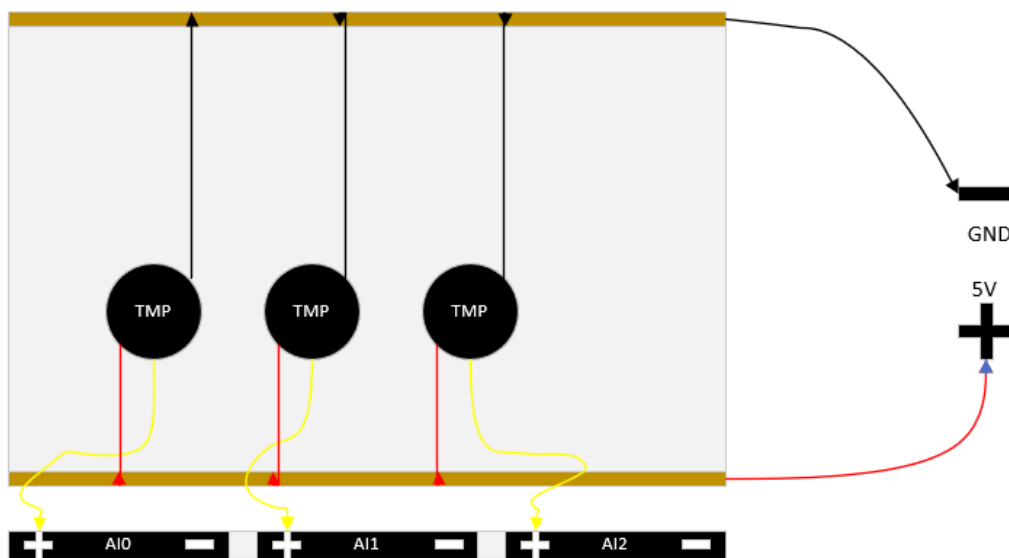
² Å bruke en bærbar PC brukes kun under utviklingsstadiet til versjon 1.0. I versjon 2.0 skal det utvikles en wifitilkoblet integrert enhet.

12. IO-enhet

Som IO-enhet har vi valgt en Daq fra National Instruments. Modellnummer på denne er oppført i strykkelisten i systemdokumentasjonen.



Bilde 12.1 NI-Daq tilkoblet tre temperatursensorer



Figur 2 Koblingskjema