



Høgskulen
på Vestlandet

BACHELOROPPGAVE

Temperaturmålere i elektriske tavler

Temperature measurement devices in
electrical switchboards

Mirnes Sabaredzovic - 181345

Anders Torgilstveit - 938409

Dataingeniør

Informatikk

Fakultet for ingeniør- og naturvitenskap

Institutt for datateknologi, elektroteknologi og realfag

Veileder Remy Andre Monsen

Innleveringsdato 08.06.2021

Jeg bekrefter at arbeidet er selvstendig utarbeidet, og at referanser/kildehenvisninger til alle kilder som er brukt i arbeidet er oppgitt, *jf. Forskrift om studium og eksamen ved Høgskulen på Vestlandet, § 10.*

<i>Rapportens tittel:</i> <i>Temperaturmålere i elektriske tavler</i>	<i>Dato:</i> 04.06.2021
<i>Forfatter(e):</i> Mirnes Sabaredzovic Anders Torgilstveit	Antall sider u/vedlegg: 38 Antall sider vedlegg: 18
<i>Studieretning:</i> Dataingeniør og Informatikk	<i>Antall disketter/CD-er:</i> 0
<i>Kontaktperson ved studieretning:</i> Remy Andre Monsen	<i>Gradering:</i> ingen
<i>Merknader:</i>	

<i>Oppdragsgiver:</i> Leirvik Elektro	<i>Oppdragsgivers referanse:</i> Jan Ove Matre
<i>Oppdragsgivers kontaktperson:</i> Jan Ove Matre	<i>Telefon:</i> 900 89 149

<p><i>Sammendrag:</i></p> <p>Prosjekteier driver ett elektrofirma og trenger redskap til å overvåke temperaturer i elektriske tavler.</p> <p>Prosjektet lager en applikasjon. Det er en temperaturmåler som plasseres inne i en elektrisk tavle. Dersom temperaturen stiger over gitte grenser, som for eksempel 40 grader, så skal prosjekteier varsles. Prosjekteier kan også besøke en webside for å se oversikt over temperaturene i sikringsskapene. Prosjektet vil i denne oppgaven vise hvordan applikasjonen er konstruert.</p> <p><i>Summary</i></p> <p>The Project owner owns an electrician company and need a tool to monitor temperatures in electrical switchboards. The Project makes an application. It is a temperature measuring device which is placed inside an electrical switchboard. If the temperature rises above a limit, for example 40 degrees, then the project owner will be alarmed. The project owner can also visit a website to monitor the temperatures in the electrical switchboard. The project will in this rapport, show how the application is constructed</p>

Stikkord:

Temperaturmålinger	Internet of things	Microprosessor
--------------------	--------------------	----------------

Høgskulen på Vestlandet, Fakultet for ingeniør- og naturvitskap

Postadresse: Postboks 7030, 5020 BERGEN

Besøksadresse: Inndalsveien 28, Bergen

Tlf. 55 58 75 00

Fax 55 58 77 90

E-post: post@hvl.no

Hjemmeside: <http://www.hvl.no>

FORORD

Å arbeide med "tingenes internett" har vært en utfordring. Derfor er det kjekt å sitte igjen med en følelse av å ha lykket. Dette hadde prosjektet ikke klart uten god hjelp. Derfor vil vi takke Remy Andre Monsen for god veiledning. Dine tilbakemeldinger har vært ett svært viktig bidrag i prosjektet. Vi vil også takke Jan Ove Matre og Leirvik Elektro for godt samarbeid. Det har blitt mange gode konstruktive samtaler.

Innhold

1	INNLEDNING.....	1
1.1	KONTEKST	1
1.2	MOTIVASJON OG MÅL.....	1
1.3	AVGRENSNINGER.....	2
1.4	RESSURSER.....	2
1.5	OPPBYGGING AV RAPPORTEN.....	2
2	PROSJEKTBEKRIVELSE.....	3
2.1	PRAKTISK BAKGRUNN	3
2.1.1	<i>Prosjekteier</i>	3
2.1.2	<i>Tidligere arbeid</i>	3
2.1.3	<i>Initielle krav</i>	4
2.1.4	<i>Initiell løsnings-idé</i>	5
2.2	LITTERATUR OM PROBLEMSTILLINGEN.....	5
3	DESIGN AV PROSJEKTET.....	6
3.1	FORSLAG TIL LØSNING	6
3.1.1	<i>Første løsning</i>	6
3.1.2	<i>Andre løsning</i>	7
3.1.3	<i>Tredje løsning</i>	8
3.1.4	<i>Diskusjon av alternativene</i>	8
3.2	VALGT LØSNING.....	10
3.3	VALG AV VERKTØY.....	10
3.4	PROSJEKTMETODIKK.....	13
3.4.1	<i>Utviklingsmetodikk</i>	13
3.4.2	<i>Prosjektplan</i>	13
3.4.3	<i>Risikovurdering</i>	15
3.5	EVALUERINGSPLAN.....	16
4	DETALJERT DESIGN.....	17
4.1	OVERSIKT OVER WEBAPPLIKASJONEN.....	17
4.2	ARDUINO TEMPERATURMÅLERE.....	18
4.2.1	<i>Arduinokoden</i>	18
4.2.2	<i>ArduinoServlet og temperaturobjekt</i>	19
4.3	VARSLING	19
4.3.1	<i>Tidsinnstillinger</i>	19
4.3.2	<i>BackgroundJobManager</i>	20
4.3.3	<i>Temperaturalarm algoritme</i>	20

4.3.4	Måleralarm algoritme	21
4.3.5	Reaktivermåler algoritme	22
4.3.6	Brannalarm	22
4.4	STATISTIKK	23
4.5	LOGG.....	23
4.6	DATALAGRING	24
4.6.1	Databasetabeller.....	24
4.6.2	JPA	24
4.7	LOGIKK	25
4.8	ANNET	25
4.8.1	Design.....	25
4.8.2	Sikkerhet.....	25
5	EVALUERING	26
5.1	EVALUERINGSMETODER OG RESULTATER.....	26
5.1.1	Prosjektet sin vurdering av applikasjonen - brukertesting	26
5.1.2	Prosjekteier sin vurdering av applikasjonen - brukertesting	26
5.1.3	Prosjekteier sin sluttevaluering av løsningen	26
5.1.4	Evaluering av varslingssystemet - modultesting.....	27
5.1.5	Evaluering av ulike nettlesere - modultesting	27
5.1.6	Sikkerhetstesting - modultesting.....	27
5.2	DRØFTING AV RESULTATER	27
6	RESULTATER	28
6.1	TEST AV ARDUINO MKR WIFI 1010.....	28
6.2	TEST AV TEMPERATURMÅLINGENE	28
6.3	KONKLUSJON	28
7	DISKUSJON	29
7.1	VALG AV PROGRAMVARE	29
7.2	VALG AV TEMPERATURMÅLERE	29
7.3	RISIKOHÅNDTERING, KOMMUNIKASJON OG ORGANISASJON	29
7.4	VARSLING LOGG OG STATISTIKK	30
7.5	HVORDAN KUNNE RESULTATET BLITT BEDRE?	31
8	PROSJEKTMÅLENE	32
8.1	ER PROSJEKTMÅLENE NÅDD?.....	32
8.2	PROSJEKTET SIN OVERFØRINGSVERDI	32
8.3	FORSLAG TIL VIDERE ARBEID	32
9	REFERANSER	33

9.1	BØKER.....	33
9.2	NETTSIDER.....	33
10	APPENDIX.....	39
10.1	RISIKOLISTE.....	40
10.2	GANTT-SKJEMA.....	41
10.3	PROGRAMMERING OG MONTERING AV TEMPERATURMÅLER.....	42
10.3.1	<i>Programmering av temperaturmåler.....</i>	<i>42</i>
10.3.2	<i>Montering av temperaturmåler i sikringsskap.....</i>	<i>44</i>
10.4	TESTING OG RESULTATER.....	46
10.4.1	<i>Arduino og stabilitet.....</i>	<i>46</i>
10.4.2	<i>Test om temperaturmålinger er nøyaktige.....</i>	<i>47</i>
10.4.3	<i>Sikkerhetskopiering.....</i>	<i>50</i>
10.4.4	<i>Testing av webapplikasjonen.....</i>	<i>51</i>
10.4.5	<i>Testing av varsling.....</i>	<i>52</i>
10.5	LEIRVIK ELEKTRO ARDUINOKODE.....	53
10.6	ARBEIDSTEGNINGER.....	55
10.6.1	<i>Brukstilfelle.....</i>	<i>55</i>
10.6.2	<i>Sekvensdiagram og jsp oversikt.....</i>	<i>55</i>
10.6.3	<i>Servletoversikt.....</i>	<i>56</i>

1 INNLEDNING

Prosjektet “Temperaturmåler i elektriske tavler”, bygger på en ide av Jan Ove Matre.

1.1 Kontekst

Leirvik Elektro¹(Leirvik Elektro, u.å) ønsker en temperaturmåler i elektriske tavler. Grunnen til dette er at feil i elektriske anlegg er en kilde til brann. Elsikkerhetsportalen ²(Elsikkerhet Norge AS, u.å) varsler om at hver fjerde brann i norske hjem skyldes feil i det elektriske anlegget.

I dagens koronasituasjon er det viktig å begrense menneskelig kontakt. Ett eksempel på uønsket kontakt er elektrikere som utfører arbeid i sykehus. Elektrikerne kan ta med seg smitte inn på sykehuset, og de kan ta med seg smitte ut fra sykehuset. I slike sammenhenger er prosjektet “Temperaturmålere i elektriske tavler”, et nyttig redskap. Appen kan redusere antall besøk i f.eks. ett sykehus.

1.2 Motivasjon og mål

Prosjektet sitt mål er å lage en applikasjon for Leirvik Elektro som vil være nyttig for dem. Elektrikerne ønsker en temperaturmåler som måler temperaturene i elektriske tavler. Resultatet av målingene skal kunne leses på en webside. På websiden skal det finnes statistikk over temperaturene. Applikasjonen skal også varsle dersom temperaturene stiger over en gitt grense, for eksempel 40 grader celsius.

Formålet med produktet er å overvåke belastningen av elektriske komponenter og varsle dersom det trengs. Høy temperatur indikerer feil i det elektriske anlegget. Ved å ha temperaturmåler i elektriske anlegg kan Leirvik Elektro få opplysninger om hvordan anleggene takler høy belastning, f.eks. ved vinteren når strømforbruket er høyt. Elektriske komponenter liker ikke varme og ved å overvåke temperaturen kan det tidlig gjøres forandringer dersom det trengs. I følge Leirvik elektro så finnes det ikke et enkelt og rimelig system til salgs på markedet og det er derfor ønskelig at det blir laget.

¹ Leirvik Elektro <https://leirvikelektro.no/>

² Elsikkerhetsportalen.no
<https://www.elsikkerhetsportalen.no/presserom/brannfellene/#:~:text=Hver%20fjerde%20brann%20i%20norske,belasting%20på%20sikringskursen%20øker%20brannfaren>

1.3 Avgrensninger

Prosjekteiger har snakket lite om avgrensninger. Noen få punkt har likevel blitt nevnt. Det skal være en applikasjon som er intuitiv og enkel i bruk.

Det er mulig å lage løsning med GSM teknologi slik at den vil virke i områder uten tilgang til Wifi. Prosjekteier ønsker ikke denne løsningen. Derfor har prosjektet valgt å utelukke dette, og kun fokusere på løsninger der temperaturmålerene skal sende via wifi.

1.4 Ressurser

Prosjektet er ikke avhengig av mange ressurser. Alle redskap, som utviklingsverktøy, arduinoer, servere og internett adresser, er noe som prosjektet skaffer selv.

Prosjekteier sine kontinuerlige vurderinger, er en viktig ressurs. Derfor er prosjekteier sin tid, en viktig ressurs.

1.5 Oppbygging av rapporten

Kapittel 1 er innledning til oppgaven. Det presenterer prosjektets eier og konteksten til prosjektet. Det forteller også om mål, motivasjon og begrensninger for prosjektet.

Kapittel 2 prosjektbeskrivelse forteller mer om hvorfor prosjektet er viktig for Leirvik Elektro. Det vil også fortelle om andre lignende prosjekter som har blitt gjennomført. Deretter beskrives prosjekteier sitt opprinnelige forslag til løsning. Til slutt fortelles det om teorier og kunnskap som blir benyttet i dette prosjektet.

Kapittel 3 design av prosjektet med forslager til løsninger. Denne delen forteller om hvilke mulige løsninger som ble diskutert, og hvilken som ble valgt. Deretter vil det bli en redgjørelse for hvordan prosjektet er planlagt.

Kapittel 4 detaljert design handler om hvilke komponenter som er satt sammen for å nå målene.

Kapittel 5 evaluerer prosjektet. Det forteller om hvordan prosjektet ble testet og resultatet av testene.

Kapittel 6 forteller om viktige tester som har bidratt til prosjektet sitt resultat.

Kapittel 7 diskusjon om hvordan prosjektet sine valg har ført til resultat. Kunne andre valg ført til bedre resultat.

Kapittel 8 er en konklusjon om prosjektet har nådd målene side.

Kapittel 9 er referanselister.

Kapittel 10 er tilleggsdokumenter.

2 PROSJEKTBEKRIVELSE

2.1 Praktisk bakgrunn

2.1.1 Prosjekteier

Jan Ove Matre har arbeidet som montør i mer enn tretti år. Han har i tillegg utvidet kompetansen til installatør. I 2016 startet han firmaet Leirvik Elektro. Der har han oppgavene daglig leder, montør og faglig ansvarlig. Firmaet klarer seg godt. I dag er det fire ansatte montører. Til høsten vil det bli utvidet med ny lærling. Leirvik Elektro er lokalisert på Heiane i Stord kommune. Firmaet er en totalleverandør av små og mellomstore elektroinstallasjoner til boliger, fritidsboliger og næringsbygg.

2.1.2 Tidligere arbeid

Smartbjella

Produktet smartbjella ³(Smartbells, u.å) har flere likhetstrekk med prosjektet. Smartbjella har måling av temperaturer som blir sendt med GSM teknologi til servere og resultater som blir vist for bruker. Det er også en varslingstjeneste om en sau blir syk eller dør. Prosjektet har blitt inspirert av smartbjella sin varslingstjeneste og ønsker å lage en varsling på temperaturmålere som ikke virker.

Arduino

I opplæringen av arduino har det blitt brukt en webside som heter Tinkercad. Websiden sier dette om seg selv:

Tinkercad is a free online collection of software tools that help people all over the world think, create and make. We're the ideal introduction to Autodesk the leader in 3D design, engineering and entertainment software. ⁴(Autodesk, 2021).

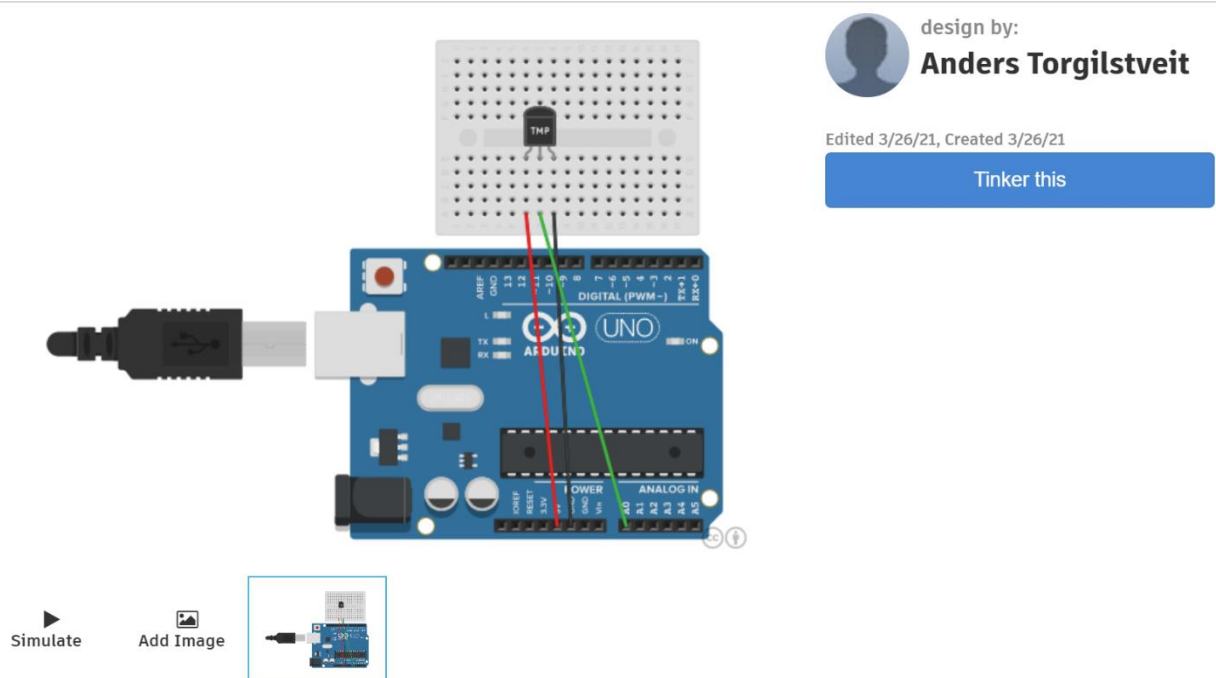
Tinkercad gjør det mulig å sette sammen arduino brett og komponenter. Så må du skrive koden. Deretter kan prosjektet startes i en simulator. Dette hjelpemiddelet har blitt brukt tidlig i prosjektet. Ved hjelp av Tinkercad er det mulig å gjøre feil, uten at elektriske komponenter blir ødelagt.

Under opplæring på arduino, ble alle arduinoprojekt først satt sammen i Tinkercad. Når prosjekten fungerte på Tinkercad, ble de overført til ett arduino brett. Ett eksempel på

³ Smartbjella <https://smartbjella.no/produkt/>

⁴ Autodesk – tinkercad - <https://www.tinkercad.com/>

dette er vist på bilde 2.1.2.1. Det viser et prosjekt på Tinkercad med temperatursensor tmp36.



Bilde 2.1.2.1 Tinkercad som viser Arduinobrett med tmp36 temperatursensor

Etter å ha lært grunnleggende bruk av Arduino, så har prosjektet brukt «Plant Communicator with MKR WiFi 1010»⁵ (Project hub, 2019) som utgangspunkt og inspirasjon til å lage temperaturmåleren. «Plant Communicator» beskriver fremgangsmåter for å sette opp en temperaturmåler på Arduino wifi og hvordan sende en forespørsel til server.

2.1.3 Initielle krav

Det innledende kravet til løsning er en temperatursensor som er plassert i ett sikringsskap. Den skal måle temperaturen i sikringsskapet. Det skal være mulig å se en oversikt over målingene. Oversikten skal inneholde nåverende temperatur, maks og minimums temperaturer, siste dag, uke, mnd og år. Dersom temperaturene stiger over en gitt grense så skal prosjekteier varsles.

⁵ Plant Communicator MKR WiFi 1010 https://create.arduino.cc/projecthub/Arduino_Genuino/plant-communicator-with-mkr-wifi-1010-efc920

2.1.4 Initiell løsnings-idé

I utgangspunktet var prosjektet sterkt inspirert av Smartbjella. Løsningen skulle ha temperaturmålere med batteridrift. Kommunikasjon skulle foregå via gsm nettet. Temperaturmålerne skulle plasseres inne i sikringskapet. Opprinnelig tenktes det at en mobil løsning var det som Leirvik Elektro ønsket. Det skulle også være en webside til brukerne.

2.2 Litteratur om problemstillingen

Boken "Databasesystemer" av Bjørn Kristoffersen har blitt brukt som grunnlag for SQL og databaser.

"HTML & CSS" en bok av Jon Duckett brukes som grunnlag for CSS og HTML.

"Core Servlets and JavaServer Pages" er brukt som utgangspunkt for å lage en webapplikasjon.

Når det gjelder opplæring av Arduino, så har det blitt brukt et hefte som fulgte med i Arduino startpakke. Det heter ⁶«The Arduino Projects Book» Dette heftet selges kun sammen med Arduino startpakken, og kan ikke kjøpes separat.

⁶ The Arduino Projects Book <https://store.arduino.cc/genuino-starter-kit>

3 DESIGN AV PROSJEKTET

3.1 Forslag til løsning

I samarbeid med Leirvik Elektro ble det diskutert mulige løsninger for prosjektet. En avklaring som kom tidlig var at GSM teknologi ⁷(Wikipedia, 2020) ikke var nødvendig. Den ble erstattet med Wifi ⁸(Wikipedia, 2021).

Leirvik Elektro var ikke opptatt av mobilitet eller ekstra sikkerhet med batteridrevne strømmålere. De ønsket stasjonære målinger over lang tid, som krevde minimalt med tilsyn.

3.1.1 Første løsning

Grunnleggende komponenter

Kjernen i systemet er en tjener som kjører Ubuntu ⁹(Canonical) med en Apache TomEE ¹⁰(The Apache Software Foundation 2021) webtjener og en PostgreSQL database ¹¹(The PostgreSQL Global Development Group 2021). Tjenermaskinen er satt opp som en virtuell maskin ved hjelp av Oracle Virtualbox ¹²(Oracle, u.å).

Grunnen til at det er valgt å bruke virtuell maskin, er sikkerhet. Det er lett å ta sikkerhetskopi av virtuelle maskiner. Dersom det skjer feil, er virtuelle maskiner lett å gjenopprette. Den kan gjenopprettes på samme vertsmaskin eller den kan kopieres til en til en annen vertsmaskin.

På TomEE webserveren er det en webapplikasjon. Den har tre hovedfunksjoner. Det ene er å ta imot temperaturmålinger fra Arduino temperaturmålerne. Det andre er å styre alarmer. Det tredje er å gi info til bruker, som for eksempel statistikk.

Temperaturmålingene skal sendes med jevne intervall. De skal sendes via Wifi til webapplikasjonen og lagres i databasen. En alarm skal bygges inn i Webapplikasjonen. Den kontrollerer temperaturene som blir lagret i databasen, og varsler til bruker dersom nødvendig. Alarmer deles i temperaturalarm og måleralarm. Temperaturalarm varsler

⁷ Gsm teknologi <https://no.wikipedia.org/wiki/GSM>

⁸ Wifi https://no.wikipedia.org/wiki/Tr%C3%A5dl%C3%B8st_lokalt_datanett

⁹ Ubuntu <http://ubuntu.com/download/desktop>

¹⁰ TomEE <https://tomee.apache.org/download.html>

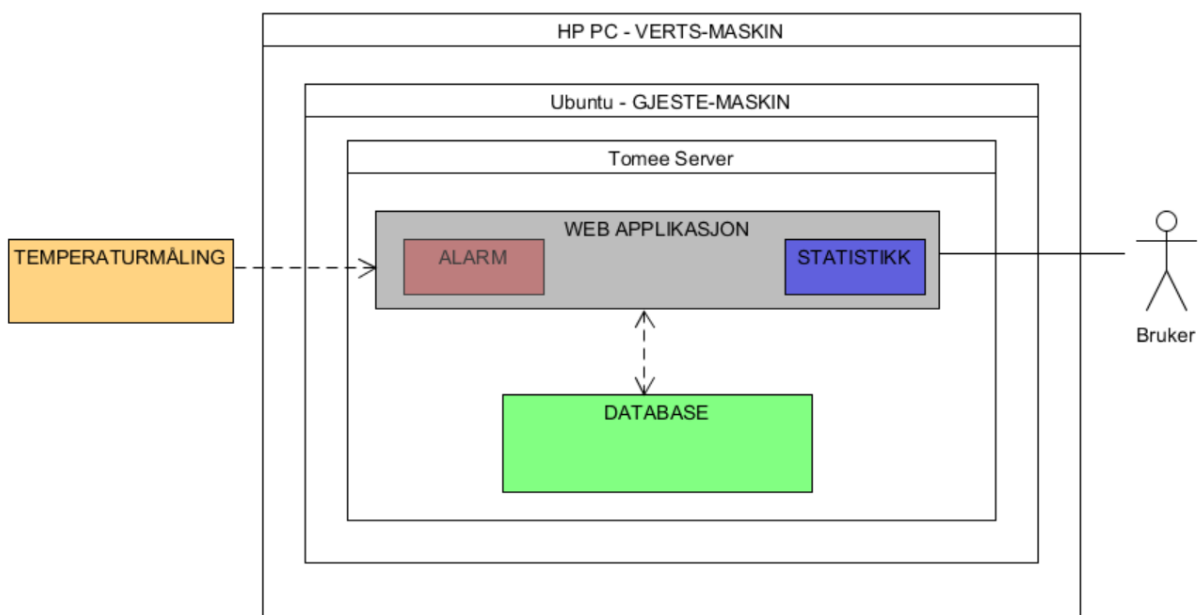
¹¹ PostgreSQL <http://www.postgresql.org/download/>

¹² Virtualbox <https://www.virtualbox.org/wiki/downloads>

når temperaturen blir for høy. Måleralarm varsler dersom det ikke er kontakt mellom server og temperaturmålerer.

I webapplikasjonen kan bruker stille inn varslingen. Det er mulig å sett grenser for når varslingene skal starte. F.eks kan alarmen utløses når temperaturene stiger over 40 grader på en temperaturmåler og en annen temperaturmåler kan stilles til for eksempel 30 grader. Varslingene kan også slås av dersom prosjekteier ønsker dette.

Ett supplement til webapplikasjonen hadde vært å lage en mobilapp. Dette har prosjektet ikke ressurser til å fullføre innen tidsfristen. Det er også vanskelig for prosjektet å oppdatere en mobilapp, dersom dette trengs. Derfor er mobilapp forkastet. En webapplikasjon dekker prosjekteier sine behov og det gir prosjektet god kontroll til å gjøre alle nødvendige oppdateringer.



Figur 3.1.1.1 Første løsning

3.1.2 Andre løsning

Utgangspunktet er løsningen fra figur 3.1.1.1. I tillegg blir det lagt til flere komponenter.

I denne løsningen blir det lagt maksimal vekt på sikkerhet. Det blir to servere (verts-maskiner), identiske med pcene i løsning 1. Begge serverne blir plassert på to ulike geografiske plasseringer, og begge har ulike ip-adresser. Temperaturmålerne sender til begge maskinene. Dersom en av serverne krasjer eller blir ødelagt vil systemet fortsette uten tap av verken data eller sikkerhet.

Det vil også tilbys Arduino temperaturmålere med gsm teknologi og strømforsyning fra batteri, som ett alternativ til temperaturmålere med wifi teknologi.

3.1.3 Tredje løsning

Denne er identisk med andre løsning. Det som er nytt, er at lokale servere blir byttet ut med servere i skyen.

3.1.4 Diskusjon av alternativene

Dette er momenter som må vurderes når det skal velges løsning:

- Leirvik Elektro ønsker en applikasjon som er rimelig og lett å vedlikeholde.
- Leirvik Elektro har lite datakompetanse. De har ingen motivasjon eller forutsetninger for å vedlikeholde temperaturmålere og servere.
- Datatap kan skje. Arduino senderne er koblet til wifi og det er regelen heller enn unntaket at i perioder vil internettet være nede. Så i perioder vil internett-tap blokkere sendinger av temperaturer. Hvor mye datatap kan tåles uten at det ødelegger appen sin verdi?
- Applikasjonen skal være ferdig innen en tidsfrist.
- Applikasjonen er ikke en brannvarsler. Applikasjonen er designet med tanke på å oppdage feil, lenge før det blir en eventuell brann.

Løsning 1 dekker kravene som Leirvik Elektro har bedt om. I tillegg er det den rimeligste løsningen. Den sparer penger på vedlikehold. I tillegg så trengs det kun innkjøp av en stk. server. Dersom denne løsningen skal forkastes, så gjøres det på grunnlag av at den ikke er sikker nok. Derfor vil dette bli diskutert grundig.

Datatap kan forekomme. Hvilke konsekvenser kan dette få? Dersom temperaturmålingene er stabile mellom 20 og 30 grader, så betyr det null, om det mistes målinger som bekrefter den stabile temperaturen. Tap av data har verdi, dersom det mistes temperaturmålinger, som ville ha startet alarmen.

Hvor ofte oppstår en situasjon der det bør startes en alarm, men på grunn av datatap, så blir ikke alarmen startet.? Svaret er "svært sjelden". Dette er ett scenario som skjer så sjeldent at det ikke blir tatt med i vurderingen, for valgt løsning.

Et scenario er brann og naturkatastrofer som ødelegger server. Dette har liten effekt på applikasjonen. Server kan fort gjenopprettes fra sikkerhetskopi. At det tapes noen temperaturmålinger betyr lite.

Hva skjer ved en "værste tenkelige tilfelle" der temperaturmålerne blir rammet av brann og strømbrudd samtidig. Internettet og temperaturmålerne vil være ubrukelig. En Arduino med GSM teknologi og batteri strømforsyning, kunne ha varslet i en slik situasjon. Leirvik Elektro ønsker ikke en brannvarsler til kundene sine. Applikasjonen skal kun ha en overvåkingsfunksjon. Gsm teknologi er også en dyrere løsning.

Løsning 2 er en dyr løsningen. Det blir ekstra investeringer i server og internettlinje. I tillegg så må det betales for vedlikehold av to servere. Arduino gsm teknologi er også dyrere enn Wifi teknologi. Gsm teknologi krever et mobil abonnement som vil utgjøre en fast kostnad. I tillegg er gsm teknologi nesten tre ganger så dyrt i innkjøp. En Arduino wifi ¹³(Kjell & Company, u.å) koster ca 500 kr mens en Arduino gsm ¹⁴(Kjell & Company. u,å) koster ca 1300 kroner.

Prosjekteier vil ha en applikasjon som varsler i god tid før brann. Produkteiger vil ikke varsles når det brenner i sikringsskapet. Han vil varsles når temperaturen stiger til rundt 40 grader. Da kan det gjøres tidlige endringer, slik at det aldri begynner å brenne.

Temperaturmålere med batteridrift er ikke ett alternativ. Batterier som må skiftes, er en ustabil løsning, ifølge Leirvik Elektro.

Løsning 3 er også ett alternativ. Dette er nok den dyreste løsningen. Den gir også høye månedlige utgifter, noe som prosjekteier ønsker å unngå. Jeg har hentet pris på Azure ¹⁵(Microsoft Azure, 2021). Det er vanskelig å gi en link til prisen, fordi den er hentet fra en betalingskurv, laget med en privat microsoft konto(Se bilde 3.1.4.1). Den rimeligste månedsprisen som Azure tilbyr er. 37,10 dollar. Da er kravene at det betales for en periode på 3 år. Dette gir en totalpris på kr 1335,6 dollar. Omgjort til norske kroner med dagens kurs, blir dette 11 111,48 kroner.

Ditt overslag

1 D2 v3 (2 vCPU-er, 8 GB RAM); Linux - U... Forskudd: USD 0,00 Månedlig: USD 37,10

Virtuelle maskiner

OMRÅDE: West US OPERATIVSYSTEM: Linux TYPE: Ubuntu NIVÅ: Standard

Bilde 3.1.4.1 Azure handlekurv

Prosjektet mangler kompetanse innen sky servere. Dersom prosjektet skal fortsette å ta sikre valg, så bør det ikke satses på denne løsningen.

¹³ Wifi kort Kjell & Company <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/utviklingskort/arduino-mkr-wifi-1010-utviklingskort-p87194>

¹⁴ Gsm kort Kjell & Company <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/utviklingskort/arduino-mkr-gsm-1400-utviklingskort-p87191>

¹⁵ Microsoft Azure - <https://azure.microsoft.com/nb-no/>

Om serveren sin sikkerhet skal vurderes, så er dette en sikker løsning. En skyserver er godt beskyttet mot strømstans, brann, naturkatastrofer og andre ting som kan stoppe en lokal server. Som for eksempel ødelagte komponenter på serverene.

Uansett valgt løsning vil kostnadene for Leirvik Elektro vil være innkjøp av temperaturmålere og vedlikehold av applikasjonen. En temperaturmåler koster ca kr 1000. Da betales det for Arduino ca kr 500, temperatursensor ca 150 og lader, kabel, deksel pluss diverse tilbehør ca kr 350. I tillegg må Leirvik Elektro forhandle frem en avtale med datakyndige for vedlikehold av server og applikasjon. Prosjektet er i dialog med Leirvik Elektro om å ta på seg ansvar, men ingen ting vil bli avklart før prosjektet er ferdig.

Konklusjonen er at løsning 1, er en god løsningen. Unntaket er dersom det skjer "verst tenkelige situasjoner", da er løsning 2 og 3, de beste valgene. Løsning 2 er en rimeligere løsning enn løsning 3.

3.2 Valgt løsning

Det er løsning 1 som blir valgt. Den har minst kompleksitet og oppfyller kravene som Leirvik Elektro har til løsning. Det er også det sikreste valget når det gjelder å bli ferdig innen tidsfristen.

Løsning 2 og 3, kunne vært bedre i et større og mer avansert system. Men kompleksiteten og prisen gjør det mindre egnet for dette prosjektet. Om nødvendig, kan utvidelsen i løsning 2 og 3, legges til seinere.

3.3 Valg av verktøy

I valg av verktøy er det prioritert å velge programmer som alle i prosjektet mestrer. Prosjektet ønsker ikke å ta unødvendig risiko, derfor velges kjente verktøy der dette er mulig. Det er også viktig at programmene kan kjøres både på PC og Mac. Prosjektet får ingen datavareressurser fra Leirvik Elektro. Derfor brukes private Arduinoer, datamaskiner, servere og internett.

Temperaturmålere

Valg av temperaturmålere var et stort usikkerhetsmoment. Prosjektet hadde ingen oversikt over markedet. Det ble en stund vurder å ta kontakt med produktet Smarbjella, for å få kjøpe temperaturmålere og sendere fra dem. Om prosjektet hadde fått kjøpt komponenter av Smarbjella, så ville det vært stor usikkert om hvordan komponentene skulle programmeres.

Det andre alternativet var Arduino. Det er mikroprosessorer laget av det italienske selskapet Arduino, Brettene er forholdsvis store komponenter sammenlignet med Smarbjella. Fordelen med Arduinoen er at de er enkle å programmere, enkle å bruke og

enkle å få tak i. For å redusere risiko, arbeidsmengde og mest sannsynlig kostnad, ble derfor Arduino foretrukket.

Arduinoen som er valgt er av merket Arduino MKR WiFi 1010 ¹⁶(Arduino Store, u.å). Dette brettet velges fordi det kan kommunisere med trådløst internett.

Temperatursensor

Leirvik Elektro har informert om at det ikke er lov til å monter en Arduino inne i sikringskapet. Derfor må Arduinoen monteres på utsiden og temperatummåler på innsiden av sikringskapet. Temperatummåleren som er valgt er PROBE DS18B20 ¹⁷(Kjell & Company, u.å). Den velges fordi den har en 90 cm lang kabel festet til temperatursensoren.



Bilde 3.3.1 Arduino mkr 1010 og temperatursensor DS18B20 testes i ett sikringskap

Java

Java er valgt som programmeringsspråk. Dette valget er tatt på grunnlag av at prosjektmedlemmene har mye erfaring fra dette språket. Utviklingsverktøyet som er valgt er Eclipse og Java 8. Java kan kjøres på både pc, mac og ubuntu. Eclipse er prosjektmedlemmene sitt foretrukne verktøy når det gjelder skriving av kildekode. Java 8

¹⁶ Arduino MKR WiFi <https://store.arduino.cc/arduino-mkr-wifi-1010?queryID=undefined>

¹⁷ Probe DS18B20 <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/arduino-tilbehor/temperatursensor-med-kabel-for-arduino-p87081>

er en eldre java versjon. Versjon 16, er den nyeste versjonen, når dette skrives. Når det skal begrunnes hvorfor versjon 8 blir valgt, så henvises det til forelesinger i faget Dat108.¹⁸(HVL, u.å) Dette faget underviser i webprogrammering og Servletsteknologi¹⁹(Wikipedia, 2021). Foreleserne valgte servlets som grunnlag for å lære webapplikasjoner, høsten 2019. Foreleserne sa at servlets er langt på vei erstattet med bedre redskaper. Grunnen til at det likevel brukes i undervisningen ved HVL er at servlets gir det beste grunnlaget for å skjønne ny teknologi. Prosjektet har ikke tid til å lære ny webteknologi. Det ønskes derfor å bruke servlets grunnet kompetanse. Prosjektet har forsøkt å erstatte java 8, med nyere versjoner, som java 11, men dette fungerte ikke sammen med servlets.

Server

Prosjektet sin serverløsning bygger på en pc med Windows 10 pro som operativsystem. På denne maskinen kjøres virtuelle maskiner ved programvaren Virtualbox²⁰(Oracle, u.å). Operativsystemet til de virtuelle maskinene er Ubuntu²¹(Canonical). Grunnen til at Virtualbox og Ubuntu blir valgt er at dette er kjent kunnskap og derfor et sikkert valg. Virtuelle maskiner velges av pålitelighetsgrunner. Det blir tatt regelmessige kopier av den virtuelle maskinen. Ved eventuelle problemer så kan en virtuell maskin hurtig erstattes av en kopi.

De resterende programmene som blir valg, er også kjent for prosjektmedlemmene, derfor regnes dette som sikre valg. Det er Apache TomEE WebProfil²²(The Apache Software Foundation, 2021), PostgreSQL database²³(The PostgreSQL Global Development Group, 2021), Umllet kartleggingsverktøy²⁴(Open Source, u.å), Github synkroniseringsverktøy²⁵(GitHub, Inc, 2021), Trello digital tavle²⁶(Atlassian, 2021) og Microsoft Office²⁷(Microsoft Office, 2021).

Jarfiler

¹⁸ Hvl – dat108 <https://www.hvl.no/studier/studieprogram/emne/dat108>

¹⁹ Wikipedia servlets [Jakarta Servlet - Wikipedia](#)

²⁰ Virtualbox <https://www.virtualbox.org/wiki/downloads>

²¹ Ubuntu <http://ubuntu.com/download/desktop>

²² TomEE <https://tomee.apache.org/download.html>

²³ PostgreSQL <https://www.postgresql.org/download/>

²⁴ UMLet <https://www.umlet.com>

²⁵ GitHub <https://github.com>

²⁶ Trello <https://trello.com/home>

²⁷ Microsoft Office <https://www.office.com>

For at TomEE web 7.1.4, skal fungere sammen med Eclipse, må det legges til jar filer. Filene skal legges i TomEE sin lib mappe. Javax.mail.jar - er nødvendig for at prosjektet skal kunne sende email. For at Eclipse skal kommunisere med databasen må også eclipselink 2.7.1 og postgresql-42.2.5 legges til. Om det er mulig å bruke nyere versjoner av jarfilene har prosjektet ikke testet. Link til jarfilene finnes i appendix.

3.4 Prosjektmetodikk

3.4.1 Utviklingsmetodikk

Prosjektet vil bli gjennomført etter Scrum ²⁸(Visma) som er bygget på Agile ²⁹(Flexify, 2019) prinsipper. Scrum bruker korte sprint faser. I hver fase, kan det være flere iterasjoner. På grunn av tid, er det planlagt tre faser og fire iterasjoner.

Selve prosjektet gjennomføres på en inkrementell måte. Man starter med den viktigste og mest grunnleggende funksjonaliteten (altså den funksjonaliteten produkteier prioriterer høyest), og bygger deretter på mer og mer funksjonalitet. På denne måten maksimerer man hele tiden produktets verdi, og man får korte “feedback loops” – man får løpende tilbakemelding på hva som fungerer og ikke fungerer, og kan justere underveis ³⁰(Nes, 2019).

All programmering vil bygge på Grasp ³¹(Wikipedia, 2021). Dette er en forkorting for General Responsibility assignment Software Patterns/Principles. Det er verktøy av ti prinsipper som veileder til å skrive god kildekode.

3.4.2 Prosjektplan

Prosjektet sin samarbeidsavtale er å snakkes daglig i hverdager. Alle arbeidsoppgavene blir fordelt på Trello, og prosjektmedlemmene velger oppgaver selv. Samarbeidsavtalen med Leirvik Elektro er ett møte etter hver iterasjon. Prosjektet kan også ta kontakt med Prosjekteier utenom dette.

²⁸ Visma <https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>

²⁹ Agile - <https://www.flexify.no/blogg/agile-metoder>

³⁰ Scrum, av Solveig Maria Nes <https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>

³¹ Grasp https://en.wikipedia.org/wiki/GRASP_%28object-oriented_design%29#:~:text=General%20Responsibility%20Assignment%20Software%20Patterns%20%28or%20Principles%29%2C%20abbreviated,in%20his%201997%20book%20Applying%20UML%20and%20Patterns.

Ganttskjemaet er prosjektet sitt planleggingsverktøy (appendix 2). Der er det en oversikt over fire viktige deler av prosjektet. Det er apputvikling, diverse, oppgave og obliger. Under de fire neste overskriftene vil de fire delene forklares bedre.

3.4.2.1 Apputvikling

Prosjektet har planlagt tre programmeringsfaser: Oppstart, bygging og overlevering. Dette blir igjen delt opp i iterasjoner som varer fra en til tre uker. Det er satt av ni uker til programmering og det blir fordelt på fire iterasjoner.

Fase oppstart - Første iterasjon

Tid: uke 11, 12, 14. Det er satt av tre uker fordi det er mye arbeid i denne iterasjonen.

Programmering: Prosjektet har vurdert den mest grunnleggende funksjonalitet til å være følgende: En temperaturmåler som sender jevnlig temperaturmålinger til en webserver. På webserveren blir temperaturene lagret i en database. Resultatet blir vist på en webside.

Testing: Test at temperaturer blir sendt fra temperaturmåler og blir lagret i database.

Fase bygging - Andre iterasjon

Tid: uke 16 og 17.

Programmering: Etter første iterasjon sin tilbakemelding fra Leirvik Elektro vil prosjektet gjøre forandringer på websiden. Det blir mer informasjon samlet på startsiden.

Meldingstjeneste skal også implementeres i denne iterasjonen. Den skal varsle prosjekteier med e-post dersom temperaturen når kritisk nivå, 40 grader.

Testing: Test at e-postvarsel blir sendt når temperaturer når kritiske nivå. Prosjekteier vil få tilsendt en temperaturmåler som han skal bruke til å evaluere resultatet sammen med prosjektet. Det vil også kjøres tester på temperaturmåleren. Den skal stress-testes for å kontrollere at den er robust. Den vil bli testet ved strømbrudd, manglende wifi- og serverkontakt. De skal også testes om ytre påvirkninger som varme og kulde påvirker temperaturmålinger.

Fase bygging - Tredje iterasjon

Tid: uke 18, 19, 20. I denne iterasjonen er det planlagt mye arbeid med oppgaveskriving. Det blir da mindre tid til programmering. Derfor blir denne iterasjonen satt til tre uker.

Programmering: Prosjekteier ønsker å ha mulighet til å sette kritisk temperatur selv. Derfor vil applikasjonen bli utvidet med dette. Da må også sikkerhet forsterkes, og deler av webområdet bli sikret med brukernavn og passord. Det vil også jobbes mer med statistikk i denne fasen. Prosjekteier ønsker en historisk oversikt over temperaturene. Det

skal også implementeres varsling til prosjekteier dersom en temperaturmåler slutter å virke.

Testing: Det skal testes at det sendes varsel dersom temperaturmålerne slutter å virke. Det skal testes at statistikken stemmer. Det skal også testes at appen fungerer fra ulike nettleisere på pc, nettbrett og mobil.

Fase overlevering - Fjerde iterasjon

Tid: uke 21. I denne iterasjon skal applikasjonen være ferdig programmert. Det er ikke planlagt programmeringsarbeid nå, kun testing. Det er derfor bare satt av en uke. Hensikten er å få løst eventuelle startproblemer.

3.4.2.2 Diverse

Planlegging, der resultatet har blitt til pre-report.

Litteratursøk: Det skal leites etter litteratur som omhandler prosjekter om temperaturmåling. Litteraturen skal sammenlignes med prosjektet.

Eksperimentelt arbeid: I denne fasen vil det arbeides med forskningsspørsmål.

-Vil temperatur i sikringsskapet øke og minke dersom det brukes mer eller mindre strøm?

-Er Arduino nøyaktige temperaturmålere? Vil de bli påvirket av faktorer som kulde og varme

3.4.2.3 Oppgave

Dette er en oversikt over hvordan prosjektet har valgt å skrive bacheloroppgaven. Kapitlene er satt opp med milepæler for når de skal være ferdig skrevet.

3.4.2.4 Obliger

Obligatoriske innleveringer trenger ingen grundigere forklaring.

3.4.3 Risikovurdering

Prosjektet har laget en liste over risikoer og forslag til håndtering av dem. Listen finnes i kapitlet appendix. Håndteringen og prioriteringen av risikoer er avhengig av hvor alvorlig konsekvensen av risikoen er og sannsynligheten for at hendelsen oppstår. Det settes en tallverdi på både sannsynligheten for at risikoen oppstår og alvorligheten av konsekvensene. Tallverdiene går fra 1-5, der 1 er lav og 5 er høy. For å finne risikofaktor multipliseres disse to tallene. Risikofaktor går fra 5 til 25, hvor høyere betyr større fare og derfor større prioritet.

3.5 Evalueringsplan

Planen er å få en fungerende applikasjon så fort som mulig. Da kan testing av produktet starte tidlig. Det vil bli sendt temperaturmålere til Leirvik Elektro tidlig i prosjektet. Dermed blir det mulig å teste produktet i samarbeid med Leirvik Elektro. Det arbeides i iterasjoner og Leirvik Elektro får mulighet til å kommentere prosjektet etter hver iterasjon.

Prosjektet vil bli vurdert etter:

- Har elektrikerne nytte av appen.
- Er temperaturmålingene nøyaktige. Blir temperaturmålingene påvirket av faktorer som varme og kulde.
- Er temperaturmålerene robuste. Tåler de strømbrudd, kulde og varme. Takler de å være uten kontakt med wifi og server, over en lengre periode.
- Virker varslingsystemet. Blir prosjekteier varslet dersom det er unormale temperaturer eller temperaturmålerne slutter å virke?
- Fungerer webapplikasjonen fra ulike nettlesere?

4 DETALJERT DESIGN

I dette kapitlet vil prosjektet vise hvordan målene ble nådd gjennom å lage webapplikasjonen. Målene for prosjektet blir repetert først.

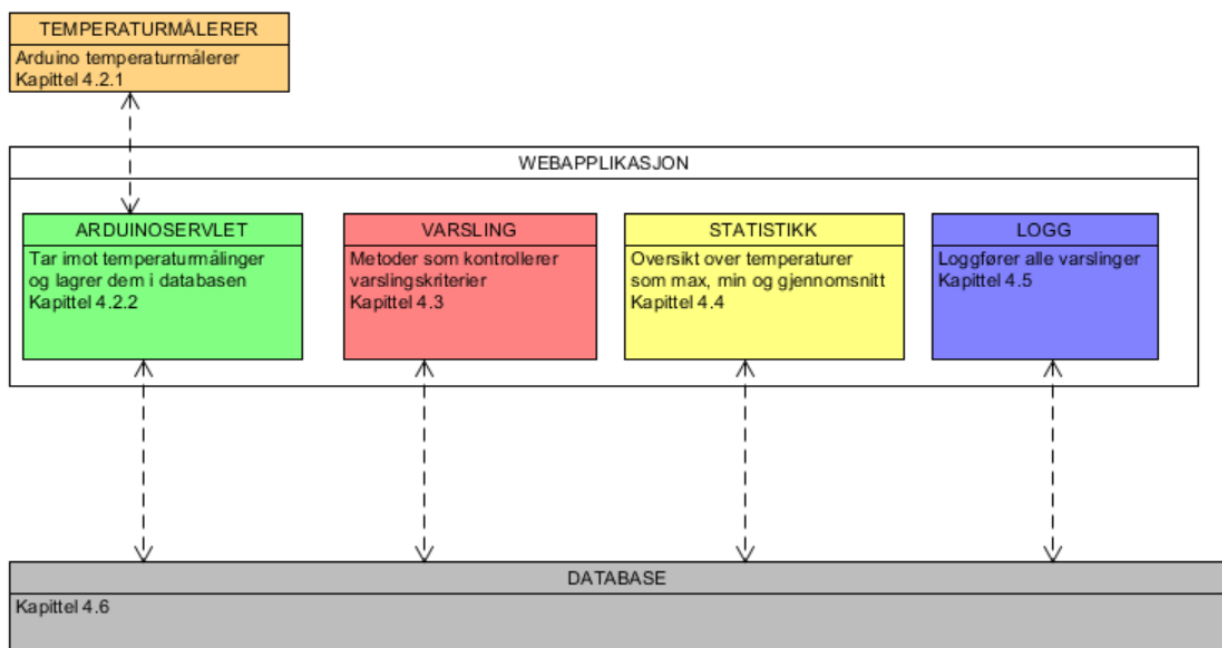
Elektrikerne ønsker termometerer som måler temperaturene i elektrisk tavler. Resultatet av målingene skal kunne leses på en webside. På websiden skal det finnes statistikk over temperaturene. Applikasjonen skal også varsle dersom temperaturene stiger over en gitt grense, for eksempel 40 grader celsius.

Kapittel 3.1.1, viste hvordan databasen og webapplikasjonen er plassert på en virtuell datamaskin. Nå skal komponentene i webapplikasjonen forklares, som er vist i figur 4.1.1. Deretter skal logikken mellom komponentene, databasen og termometerne utdypes.

4.1 Oversikt over webapplikasjonen

Figur 4.1.1 viser de viktige delene i webapplikasjonen. Den er bygget opp av arduinoservlet, statistikk, varsling og logg. Termometerne og databasen samarbeider med webapplikasjonen og det er derfor naturlig å fortelle om dem samtidig som webapplikasjonen blir presentert.

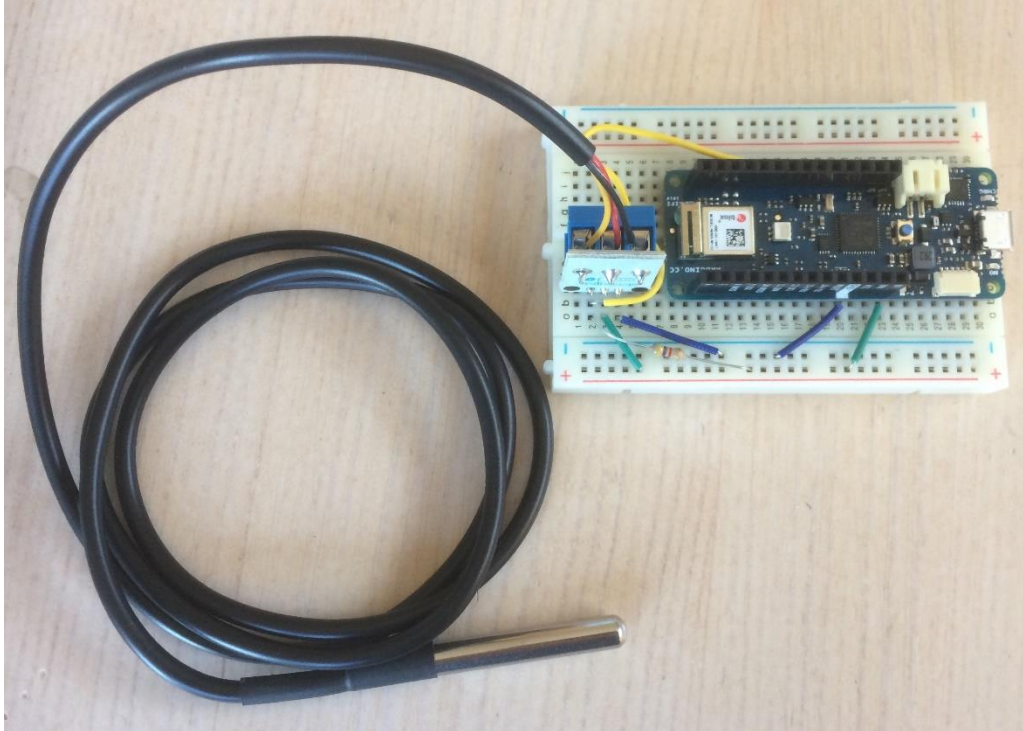
Logg er lagt til på grunn av tilbakemeldinger fra Leirvik Elektro, da de hadde begynt å bruke applikasjonen i tredje iterasjon. Derfor er loggen ikke nevnt i kapittel 3.1 og opprinnelig løsning.



Figur 4.1.1 Oversikt over applikasjonen sine viktigste komponenter

4.2 Arduino temperaturmålere

Arduino Mkr Wifi 1010 (Bilde 4.2.1) er valgt som temperaturmålere. De kommuniserer både via blåtann og Wifi. Arduinokoden og forespørselen til webapplikasjonen er også viktige komponenter som prosjektet vil fortelle om under denne overskriften.



Bilde 4.2.1 Arduino Mkr Wifi 1010

4.2.1 Arduinokoden

Kildekoden kan leses i helhet i appendix «Leirvik Elektro arduinokode». Her følger en enkel oversikt:

```
setup()
loop(){
    readTemp()
    post()
    Wifitest
    pause
}
```

Koden starter med metoden setup, som kobler arduinoen til wifi nettverket. Deretter blir tråden sendt videre til loop() metoden, som kjører til strømmen blir brutt. I loop metoden blir tråden først sendt til metoden readTemp(), der temperaturen blir lest i fra temperatursensor. Deretter blir tråden sendt videre til post() metoden som sender en request til webapplikasjonen. I requesten, blir det sendt to parameter. Det ene er navnet

på måleren, og det andre er siste leste temperatur. I Wifitest blir det kontrollert om måleren fremdeles er koblet til internettet. Dersom ikke, blir det forsøkt å koble til. Så er det 60 sekunder pause, før loopen blir kjørt på nytt.

Det er valgt å sende en forespørsel pr minutt. Grunnen til dette kan du lese om i Varsling, kapittel 4.3.6.

4.2.2 ArduinoServlet og temperaturobjekt

I webapplikasjonen er det laget en Servlet som kalles «ArduinoServlet» (Se figur 4.1.1). Den har til oppgave å ta imot forespørsler i fra temperaturmålerne. Forespørslene sender to variabler. Det er målernavnet og målt temperatur. Variablene blir overført til ett nytt temperaturobjekt. I tillegg blir det gitt temperaturobjektet en unik id, og dato, tid og tidspunkt. Deretter blir temperaturobjektet lagret i databasen.

4.3 Varsling

Varslingen er en viktig del av applikasjonen. (Se figur 4.1.1) Varsel skal sendes til prosjekteier dersom temperaturen i sikringsskapet går forbi en gitt varselgrense. Prosjekteier har satt 40 grader som en standard varselgrense. Når temperaturgrensen krysser 40 grader, vil det bli sendt ett varsel med e-post til prosjekteier. E-posten forteller at temperaturen i sikringsskapet er over 40 grader. Denne varslingen kalles «Temperaturalarm».

Det er annen varsling som kalles «Måleralarm». Denne varslingen har til hensikt å fortelle produkteier at en måler har sluttet å sende temperaturmålinger. Også her vil varselet være en e-post til produkteier.

4.3.1 Tidsinnstillinger

Hvor ofte skal det varsles til produkteier? Dette temaet er det brukt mye tid på. Det er mulig å varsle for mye, og det er også mulig å varsle for lite. Prosjekteier ønsker ikke å bli varslet for mye. Derfor kom det et krav fra prosjekteier om mulighet til å kunne skru av varsling. Han har dårlige erfaringer fra andre apper, som sender ekstremt mange unødvendige varsler. Prosjektet lagde da «innstillinger.jsp». Der kan prosjekteier, for hver enkelt måler, sette ønsket varslingsgrense, eller skru av varsling.

Måleralarm varsler ved inaktivitet, etter en time. Grensen er satt av prosjekteier. En time gir mulighet for korte strøm og Wifi brudd, uten at det blir sendt varsel.

Temperaturalarm varsler når varselgrensen krysses, for eksempel ved 40 grader. Dette gav en problemstilling. Om temperaturen i en periode svinger mellom 39 og 41 grader, vil det bli mange varslinger til prosjekteier. Løsningen på dette problemet var å lage en automatisk justering. Det blir varslet ved 40 grader, og deretter økes varselsgrensen, med +1 grad. Neste varsling blir da når temperaturen krysser 41 grader. Neste varsling

igjen blir å varsle ved 42 grader, neste deretter blir 43 osv. Prosjekteier likte denne løsningen.

Når prosjekteier har fått varsel, kan det blir gjort tiltak for å redusere temperaturen. Derette kan varselgrensen manuelt settes til 40 grader igjen. Dersom dette blir glemt, blir varselgrensen automatisk nedjustert til 40 grader, første dag i uken.

Auto-justeringen har hensikten å redusere varslinger når temperaturen svinger, og øke varslingene dersom det er en kritisk situasjon (Kapittel 4.3.6).

4.3.2 BackgroundJobManager

Dette er tidsstyrte oppgaver, som starter med jevne mellomrom. Denne klassen starter metodene `temperaturalarm`, `resetTemperaturAlarm`, `måleralarm` og `reaktivermåler`.

En alternativ løsning til tidsstyrte oppgaver, er å starte temperaturalarm hver gang server mottar en forespørsel. Prosjektet prøvde også denne løsningen, men gikk vekk fra den. Grunnen var at prosjektet mente at kildekoden ble uoversiktlig. Prosjektet følger GRASP ³²(Wikipedia, 2021), som vil at en klasse, skal ha ett ansvar. Derfor fikk BackgroundJob Manager, ett ansvar, og det var alarmene og `ArduinoServlet` fikk ett ansvar, som var å ta imot forespørselene.

4.3.3 Temperaturalarm algoritme

Denne alarmen varsler ved en gitt varselgrense. Den varsler ved å sende e-post. Dersom en temperaturmåling er høyere enn varselgrensen, vil varselgrensen automatisk justere seg med +1 grad. Dette for å unngå for mange varslinger, om temperaturen svinger rundt varslingsgrensen.

Temperaturalarm kjøres av BackgroundJobManager, med samme intervall som forespørselene til server, en gang hvert minutt.

³² Grasp https://en.wikipedia.org/wiki/GRASP_%28object-oriented_design%29#:~:text=General%20Responsibility%20Assignment%20Software%20Patterns%20%28or%20Principles%29%2C%20abbreviated,in%20his%201997%20book%20Applying%20UML%20and%20Patterns.

Innstillinger

Måler: SVEIARHAUGEN ▾ Alarm: Aktiv ▾ Temperaturgrense: 20 ▾

Adresse	Varsel	Grense
SVEIARHAUGEN	Inaktiv	40
BANDADALEN	Inaktiv	40
STORASLETTO	Aktiv	61

Bilde 4.4.3.1 Innstillinger for varsling

Bilde 4.4.3.1 viser hvordan prosjekteier kan stille varslingsystemet. Storasletto er aktiv, og har automatisk justert varselgrensen til 61 grader celsius. Det kan velges hvilken temperaturgrense der varslingen skal starte. Det er nytting å kunne sette lave temperaturer, når temperaturalarmen skal testes.

Temperaturalarm samarbeider med metoden `resettTemperaturAlarm` som kjøres første dag i uken. Alle varselgrenser som har autojustert seg til over 40 grader, blir da resatt til 40 grader.

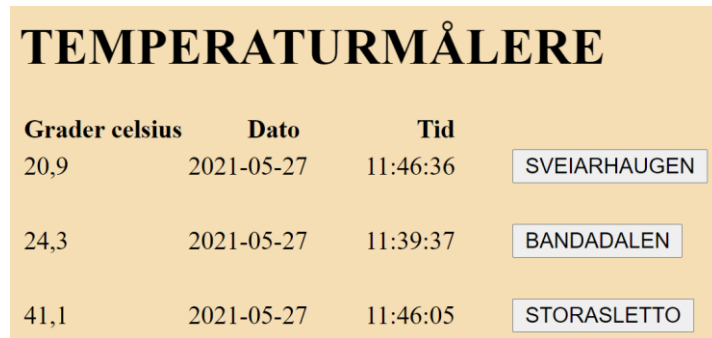
4.3.4 Måleralarm algoritme

Måleralarm har til hensikt å varsle dersom en temperaturmåler slutter å sende målinger. For hver enkelt aktiv måler, vil måleralarm hente den siste lagrede temperaturmålingen og kontrollere tiden. Dersom det er mer enn en time siden målingen ble lagret i databasen, så regnes måleren som defekt. Det blir sendt e-post varsel om dette til produkteier. Måleren blir da registrert som inaktiv. Så lenge som måleren er inaktiv, blir det ikke sendt flere varsler.

Måleralarm bør ikke kjøres for ofte. Dersom serverne er nede i lenger tid, så kan du få følgende situasjon: Serveren starter opp, etter f.eks. en og en halv time nedetid. Deretter kjøres måleralarm før serveren får inn nye temperaturmålinger. Serveren registrerer at det er mer enn en time siden den mottok varsler fra målerne, og sender derfor ut e-post varsel på alle målerne. For å unngå dette scenarioet, er måleralarm satt til å kjøre hvert femte minutt. Dersom server starter opp etter lengre fravær, vil det gå fem minutter før måleralarm blir kjørt første gang. Dette er rikelig tid til å få inn nye målinger fra alle aktive målere.

4.3.5 Reaktivermåler algoritme

Dette er en hjelpemetode til måleralarm. Reaktivermåler brukes til å kontrollere om en inaktiv måler har gjenopptatt sendingene. Dersom den oppdager at dette er tilfelle, blir den inaktive måleren satt til aktiv igjen. Prosjekteier blir ikke varslet om dette, fordi det kan kontrolleres når siste måling ble mottatt, ved å lese på applikasjonen sine websider. Denne metoden kjøres hvert femte minutt, og styres av BackgroundJobManager.



Grader celsius	Dato	Tid	
20,9	2021-05-27	11:46:36	SVEIARHAUGEN
24,3	2021-05-27	11:39:37	BANDADALEN
41,1	2021-05-27	11:46:05	STORASLETTO







Bilde 4.4.5 Siste mottatt måling vises ved under «Tid»

4.3.6 Brannalarm

Applikasjonen har ingen brannalarm. Men den kan gi en indikasjon på et branntilløp. Som eksempel brukes et scenario, der varslingsgrensen er satt til 40 grader, og temperaturen stiger fra 20 grader til 70 grader.

Temperaturalarm begynne å varsle når temperaturen krysser 40 grader. Deretter vil den fortsette å varsle etter som temperaturen stiger. Produkteier vil da motta mange e-poster, der hver eneste e-post varsler om at temperaturen stiger. Dette kan gi produkteier tid til å reagere. For å registrere hurtige temperaturøkninger er derfor Arduinoene instillt til å sende en måling pr. minutt.

Produkteier ønsker ikke å bli varslet for mye. Dette gjelder ikke i situasjoner som kan indikere ett branntilfelle. Da er det bedre med for mye varsling. Bilde 4.4.1 viser innboksen ved en hurtig temperaturøkning.

	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 56.19 grader. Den har oversteget kri	12.27
	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 54.56 grader. Den har oversteget kri	12.26
	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 53.44 grader. Den har oversteget kri	12.25
	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 52.12 grader. Den har oversteget kri	12.24
	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 50.69 grader. Den har oversteget kri	12.23
	anderstorgilstveit@gmail.com Alarm temperaturmaaler Måler STORASLETTO er nå 47.69 grader. Den har oversteget kri	12.22

Bilde 4.4.1 E-post varsling ved hurtig temperaturøkning.

4.4 Statistikk

Dette er også en viktig del av applikasjonen (Se figur 4.1.1). Websiden viser en statistisk oversikt over temperaturene til de enkelte temperaturmålerne. Statistikken er laget ved å lage spørringer til databasen ved «Typed Query». Her kan produkteier finne informasjon om høyeste-, laveste- og gjennomsnittstemperatur i ulike tidsrom.

4.5 Logg

(Se også figur 4.1.1). Logg viser oversikt over alle måleralarmer og temperaturalarmer som er sendt til prosjekteier. Grunnen til loggen er at det kan være vanskelig å ha god oversikt over e-post i innboks, spesielt for produkteier, som ofte har en hektisk hverdag med lite kontortid. Når produkteier har fått nødvendig oversikt, kan loggen slettes.

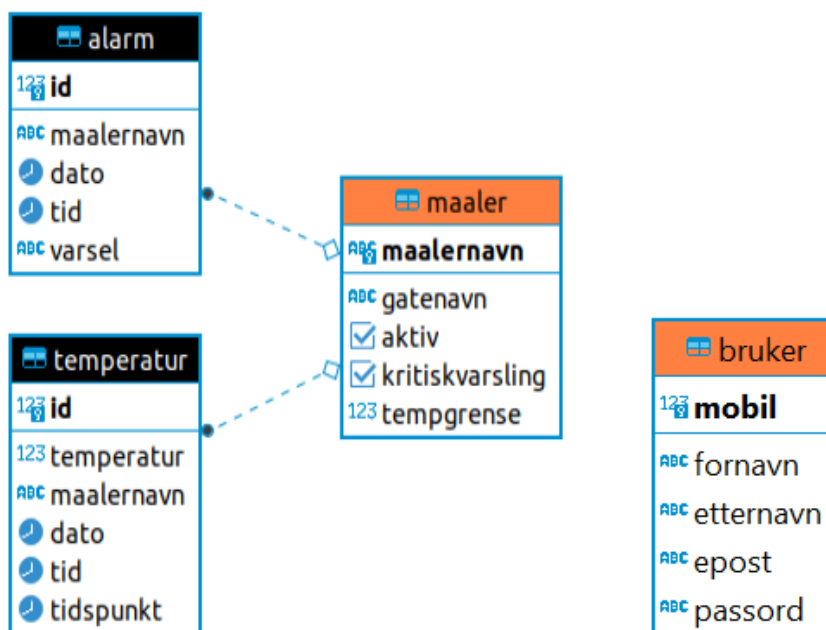
4.6 Datalagring

4.6.1 Databasetabeller

Databasen består av fire tabeller. Maaler-tabellene registrerer Arduino temperaturmålerne. Her har du maalernavn som primærnøkkel. Hver måler skal ha ett unikt maalernavn som aldri skal forandres. Gatenavn kan forandres.

Temperatur-tabellen registrerer temperaturmålingene. Her har det blitt tre tidsvariabler, dato, tid og tidspunkt. Dette blir gjort på grunn av det skal være enklere å lage spørringer som sorterer ut riktige temperaturer.

Alarm-tabellen lagrer hver gang det blir sendt ett varsel. Bruker-tabellen lagrer informasjon om bruker.



Bilde 4.3.1 Databasen

4.6.2 JPA

Java prosjektet er ett JPA prosjekt (Java Persistence API)³³(Vogella, 2017). JPA knytter javaobjekter til databaser og omvendt. Alle tabeller i databasen har ett tilsvarende objekt i kildekoden. Tabellene og objektene knyttes sammen ved hjelp av en egen klasse i kildekoden som er merket EAO.

³³ Java API

<https://www.vogella.com/tutorials/JavaPersistenceAPI/article.html#:~:text=The%20Java%20Persistence%20API%20%28JPA%29%20is%20one%20possible,is%20a%20specification%20and%20several%20implementations%20are%20available.>

4.7 Logikk

Etter å ha sett på de ulike komponentene, kan logikken i webapplikasjonen nå oppsummeres.

- Den tar imot forespørsler i fra temperaturmålerne og lagrer dem i databasen.
- Den starter og styrer alle metoder som kontrollerer varsling.
- Den viser oversikt over temperaturmålerer og statistikk.
- Den fører logg over alle varslinger som er sendt.

4.8 Annet

4.8.1 Design

Brukergrensesnittet er enkelt og applikasjonen skal være enkel i bruk. Alle websider skal være selvforklarende.

4.8.2 Sikkerhet

Det er lagt til sikkerhet. Det er kun innloggede brukere som får tilgang til websidene. Sesjon er også tilført, slik at bruker blir automatisk logget ut ved inaktivitet. Passordene er kryptere med hash algoritmen sha-256.

Forespørslene til server, gjennom ArduinoServlet, er sikret. Det skal ikke være mulig for en hacker å utføre sql angrep mot databasen. Det som ikke er godt nok, er beskyttelse mot «Denial of service» angrep.

Dersom målernavn er kjent, kan en hacker sende falske temperaturmålinger til databasen. I verste tenkelige tilfelle kunne en hacker sendt inn en høy temperatur, som f.eks 10 000 grader. Dette ville ha startet temperaturalarm, og kjørt den 9600 ganger. På grunn av dette har prosjektet lagt inn en sperre, for å begrense skaden av ett angrep. Det kan kun lagres temperaturer i databasen opp til 100 grader. Dersom et hackerangrep, vil det kun være mulig å kjøre temperaturalarm 60 ganger pr. temperaturmåler. Grensen på 100 grader er satt, for å være sikker på at kvaliteten på applikasjonen ikke blir redusert.

5 EVALUERING

5.1 Evalueringsmetoder og resultater

5.1.1 Prosjektet sin vurdering av applikasjonen - brukertesting

Et tidlig mål var å produsere en applikasjon som var i kontinuerlig drift. Hensikten var å finne feil. Når applikasjonen kjøres daglig, over en lengre periode, så øker også sannsynligheten for å finne eventuelle feil. Strategien hjalp prosjektet med å finne flere feil og tabber.

5.1.2 Prosjekteier sin vurdering av applikasjonen - brukertesting

Prosjekteier har vært svært delaktig i vurderingen av prosjektet. Etter hver eneste iterasjon har det vært samtaler om resultatene. Prosjektet fulgte planen for apputvikling i 3.4.2.1 og sendte en temperaturmåler til prosjekteier, som han mottok på slutten av andre iterasjon. Etter dette har produkteier hatt applikasjonen i daglig bruk og med det grunnlaget vurdert resultatene sammen med prosjektet.

5.1.3 Prosjekteier sin sluttevaluering av løsningen

Etter å ha snakket med prosjekteier en uke før prosjektslutt, fikk prosjektet denne tilbakemeldingen.

Prosjekteier er imponert over helheten i prosjektet. Når han skal nevner detaljer så, er det positivt at det er små temperaturmålerer. Han synes også at varslingen fungerer svært godt.

Prosjekteier mener at applikasjonen er klar til videre testing. Som eksempel nevner han en av sine tidligere ansatte som nå er vaktmester og arbeider på en gammel skole. Der hadde prosjektet passet fint til å overvåke det elektriske anlegget.

Prosjekteier er også fornøyd med nettsiden som var enkel og brukvennelig.

Det er positivt at arduinoene har fått strøm fra usb lader, for da blir det færre lovverk. Produkteier informerer at hvis det skal settes inn elektriske komponenter uten ekstern lader, så ville man blitt regnet som en produsent av elektrisk utstyr. Dette kommer med mange regler og lovverk og er ikke gunstig for dette prosjektet.

Når det gjelder forslag til forbedringer, så hadde prosjekteier foretrukket varslings direkte på en egen app på mobiltelefon istedenfor e-post.

Prosjektet har ikke hatt mulighet til å lage en app. I stedet har det på websiden blitt laget en logg, der alle varslinger blir samlet. Dersom prosjekteier mister oversikt over meldinger i innboksen, vil det være mulighet å få oversikt i loggen.

5.1.4 Evaluering av varslingsystemet - modultesting

Varsling har blitt testet ved å lage situasjoner som skal utløse alarmen.

Temperaturalarm har blitt testet gjentatte ganger. På websiden kan varselgrensen settes ned til 20 grader. Dette har vært nyttig ved utviklingen av produktet.

Måleralarm testes ved at strømmen har blitt fjernet, og deretter kontrollert at det har blitt varslet etter en time. Det har også blitt kontrollert at om server er avslått i flere timer, så blir det ikke sendt varsel på målerene når server startes igjen.

5.1.5 Evaluering av ulike nettlesere - modultesting

Websiden har blitt testet fra ulike nettlesere for å kontrollere kompatibilitet. Det har blitt testet fra iPhone, iPad, Android, Mac, PC og Linux maskiner. I tillegg er det prøvd nettleserene Safari, Microsoft Edge, Chrome og Opera. Ingen feil er funnet. Men prosjektet har konstantert at ulike nettlesere kan påvirke presentasjonen av webapplikasjonen. Det som er erfart er at javascriptet på websiden «nybruker.jsp» fungerer perfekt på de fleste nettlesere. Men Google Chrome ville ikke godta hvit bakgrunn på ett «sprett opp» vindu.

5.1.6 Sikkerhetstesting - modultesting

Webapplikasjonen har helt fra andre iterasjon blitt kjørt på database tom for temperaturmålinger. Dette for å luke vekk nullpointer exceptions som krasjer appen. Prosjektet har også testet å gjenopprette både databaser og server fra sikkerhetskopier.

5.2 Drøfting av resultater

Prosjektet og prosjekteier er fornøyd med resultatene. Applikasjonen fungerer tilfredstillende. En viktig strategi har vært å ta liten risiko. Dette mener mener prosjektet har gitt en stabil applikasjon.

6 RESULTATER

Alle logger fra testene som er gjennomført finnes i appendix "Testing og resultater".

6.1 Test av arduino mkr wifi 1010

Det har vært brukt fire temperaturmålere i dette prosjektet av merket Arduino Mkr Wifi 1010. De har nå blitt testet i en periode på tre måneder. Det har ikke vært feil på noen av dem. Strømbrydd har ikke vært et problem. De takler også godt å miste wifi tilgang. Når strøm og wifi fungerer så fungerer også Arduinoene.

6.2 Test av temperaturmålingene

Ved å test Arduinoene sine temperaturmålinger fant prosjektet at de ikke måler nøyaktig i forhold til ett kvikksølvtermometer. Der har det blitt registrert ulikheter på opp mot to grader. Temperaturen er avgjørende. På temperaturer under 40 grader kan Arduinoene registrere høyere enn kvikksølvtermometer. På temperaturer over 40 grader er det motsatt.

Alt som ligger innenfor plus minus 2,5 grader vil være godt nok, har prosjekteier informert om.

6.3 Konklusjon

Konklusjonen av testene er at Arduino temperaturmålerne er robuste og stabile. De måler temperaturene stabilt i forhold til hverandre. I forhold til kvikksølvtermometer, så måler de litt ulikt. Dette er ikke et problem. Prosjektet mener at Arduinoene er nøyaktige nok til å gi ønskede resultat.

7 DISKUSJON

Her vil det diskuteres resultatene av ulike valg som prosjektet har tatt.

7.1 Valg av programvare

For å oppnå minst risiko er det blitt brukt verktøy som medlemmene har kunnskap til fra før, som Java 8 Servlets. Det ble vurdert at de kjente verktøyene ville få jobben gjort. Det antas at med nyere verktøy kunne løsningen blitt bedre. Samtidig ville nok det gått mer tid til å lære seg nye verktøy.

Konsekvensen av valgte verktøy er at alle programmer har fungert som forventet og prosjektet er fornøyd med valget av verktøy.

7.2 Valg av temperaturmålere

Det var stor usikkerhet rundt valget av temperaturmålerer. Prosjektet manglet kunnskap og erfaring angående mikroprosessorer. Det var viktig å velge noe sikkert for å i størst grad garantere at prosjektet kom i mål. Ut fra prosjektet sin kunnskap om mikroprosessorer var Arduino det sikreste valget. Arduino viste seg å være en flexibel løsning fordi det selges mye ekstra komponenter til denne mikroprosessoren. Som eksempel kan nevnes monteringen av Arduinoene. I starten tenkte prosjektet å montere dem på innsiden av sikringsskapet og planlagte temperatursensorer deretter. Da Leirvik Elektro informerte oss om at dette ikke var mulig, måtte prosjektet tenke nytt. Det var da kun nødvendig å finne en annen temperatursensor og så var problemet løst. Det anses derfor at valget av arduino har vært et godt valg.

Som alternativ kunne prosjektet prøvd å få tak i ferdige løsninger liknende som Smartbjella. Dette valget ville nok vært et bedre valg estetisk sett, for prosjektet kunne fått en trådløs sender på innsiden av sikringsskapet. I etterkant så viser det seg at dette hadde vært et dårlig valg. Ifølge Leirvik Elektro er det en del regelverk som blir brutt dersom det blir plassert en batteridrevet enhet i et sikringsskap.

Ved å velge Arduino ble prosjektet tvunget til å oppdatere seg på mikroprosessorer som måtte monteres og programmeres. Dette har gitt prosjektet verdifull kunnskap.

7.3 Risikohåndtering, kommunikasjon og organisasjon

Risiko ble minimalisert ved å angripe de vanskeligste problemene tidligst mulig. Dette ble vurdert til å være temperaturmålerne og kommunikasjonen med webapplikasjonen. Dersom prosjektet hadde tatt feil valg, ville det være lett å foreta justeringer i tidlig fase.

Prosjektet brukte også Scrum arbeidsmetodikk som delte prosjektet opp i iterasjoner. Denne metodikken førte til mange forandringer underveis i prosjektet.

Det ble gjort forandringer på alarmalgoritmene, lagt til instillinger og laget logg. Prosjektet hadde ukentlige samtaler med prosjekteier. Underveis ble viktige ting oppklart. Ett eksempel på dette er at det ble informert om at en ikke kan sette hva som helst inn i et sikringsskap. Dette førte til endring av temperatursensor. Uten denne endring ville muligens ikke løsningen vært innenfor lovens rammeverk. Gode tilbakemeldinger fra prosjekteier førte til en applikasjon som skal passe svært godt til Leirvik Elektro sitt bruk. Derfor har det lønnet seg å ha regelmessig kontakt med prosjekteier og bruke Scrum.

På grunn av iterasjonen ble det laget en enkel applikasjon i starten. Deretter ble applikasjon mer og mer avansert. Prosjektet erfarte at jo lenger ut i prosjektet, jo vanskeligere ble det å fikse på feil. Ett eksempel på dette er hacking og «Denial of service». Da dette ble diskutert i fjerde iterasjon var det umulig å løse dette problemet på en god måte. Det var for mye kode som måtte flyttes, justeres og testes på nytt.

Ganttskjemaet var godt planlagt og svært verdifullt. Det hjalp prosjektet til å bli ferdig på estimert tid.

7.4 Varsling logg og statistikk

Måleralarm har vært en suksess. Temperaturalarm med auto-justering fungerer godt. Det kan diskuteres om det hadde vært en bedre løsning å dele temperaturalarm i to metoder. En metode kunne våket over varselgrensen og sendt max ett temperaturvarsel pr. dag. Den andre metoden kunne sendt ett varsel dersom det kom hurtig temperaturøkning. Denne løsningen hadde blokkert problemet med «Denial of service» angrep. Prosjektet mener likevel at auto-justering er den best løsningen i branntilfeller. Hastigheten som temperaturen stiger, i forkant av en brann, er uforutsigbar. Auto-justering er den algoritmen som dekker uforutsigbarheten best. Derfor foretrekkes auto-justering.

Logg har blitt lagt til seint i prosjektet. Det er derfor ikke gjort mange erfaringer med den.

Statistikk har vært nyttig. Oversikten gir en god indikasjon på den elektriske tavlen sin tilstand.

7.5 Hvordan kunne resultatet blitt bedre?

I starten av prosjektet ble det brukt tid på Arduino og koblingsbrett ³⁴(Kjell&Company, u.å). Det burde også samtidig blitt brukt tid på valg av deksel. Å legge til deksel seinere viste seg å være en dårlig løsning, da det ikke var laget egne deksel for de koblingsbrettene som prosjektet hadde valgt.

Det burde vært tenkt mer på beskyttelse mot hacking alt fra første iterasjon. Denne delen ble fokusert på først i iterasjon fire. Da var det mange biter som skulle pusles sammen. Blant annet hadde Leirvik Elektro fått tilsendt termperaturmålere, og de kunne ikke omprogrammeres og testes på den tiden som var igjen av prosjektet. Resultatet ble at prosjektet har en svakhet for «Denial of service» angrep.

³⁴ Koblingsbrett <https://www.kjell.com/no/produkter/elektro-og-verktoy/elektronikk/elektroniklab/koblingsbrett-400-hull-p36285>

8 Prosjektmålene

Dette kapitlet vil starte med å repetere prosjektmålene, før de blir evaluert.

Prosjektet sitt mål er å lage en temperaturmåler som måler temperaturene i en elektrisk tavle. Brukeren skal kunne lese temperatur og statistikk over temperatur på en webside. Prosjekteier skal varsles når temperaturene når en gitt grense, for eksempel 40 grader celsius. Prosjekteier skal varsles dersom temperaturmålerne slutter å sende målinger til server.

8.1 Er prosjektmålene nådd?

Prosjektet mener at alle mål med applikasjonen ble nådd. Leirvik Elektro har fått en godt fungerende applikasjon. Prosjekteier forventer ikke at den ferdige applikasjonen skal være klar til å brukes av kundene. Leirvik Elektro vil nå leite etter flere aktører, som kan være med å teste produktet ytterligere. Når prosjekteier ønsker å fortsette testing, så tror prosjektet at Leirvik Elektro har fått det produktet som de ønsket.

Det er noen delmål i Ganttplanen som ikke ble møtt. Litteratursøk og eksperimentelt arbeid ble det ikke tid til. Ifølge planen skulle bacheloroppgaven være ferdig skrevet innen uke 20, og referanser i uke 19. Dette gikk ikke som planlagt. Grunnen er at fremgangen gikk saktere enn forventet, og at arbeidsomfanget måtte bli nedjustert.

8.2 Prosjektet sin overføringsverdi

Kan resultatene være nyttig for andre situasjoner? Utgangspunktet Arduino, temperaturmåler og server, kan brukes til lignende prosjekter. For eksempel kan temperaturmålerne byttes ut med andre sensorer, som bevegelsessensor, lyssensor, lydsensor osv. Da har du et godt utgangspunkt for å lage andre varslingssystemer. Det kan også lages varslingssystemer som kombinerer ulike sensorer.

8.3 Forslag til videre arbeid

Dersom arbeidet fortsatte så burde temperaturmålerne blitt redesignet. Det må brukes mer tid på det estetisk. Det må bli mindre deksel, dersom de skal plasseres på utsiden av sikringsskap. Det burde også vært mål å bygge målerne inn i vegger i lag med det elektriske systemet. Men da må prosjektet løse utfordringen med å kommunisere med temperaturmålerne fra utsiden.

Ved videre arbeid anbefales det å lage en mobilapp som kan gi varslinger, ettersom prosjekteier informerte at dette var savnet. Deretter kan man legge til funksjonaliteten fra nettsiden til appen. Det vil si muligheten til å se en oversikt over målerne og temperaturene, og mulighet til å endre alarmene til hver av målerne.

Det bør også lages og testes ut en versjon, der det blir brukt gsm arduino teknologi. Det er greit å ha den kompetansen i prosjektet.

9 REFERANSER

Harvard referansestil er brukt ³⁵(UiB, UiO og HVL, 2020).

9.1 BØKER

Brown, Larry, Marty, Hall (2004) *core SERVLETS and JAVA SERVER PAGES* 2nd edition

California: Santa Clara

Duckett, Jon (2014) *HTML & CSS design and build websites*

Indianapolis: John Wiley & Sons, Inc.

Fitzgerald, Scott, Shiloh, Michael (2015) *The ARDUINO PROJECTS BOOK*

(ingen informasjon om forlag)

This book can not be sold seperately from the Arduino Starter Kit ³⁶(Arduino Store, u.å)

Kristoffersen, Bjørn(2016) *Databasesystemer* 4. utg.

Oslo: Universitetsforlaget

9.2 NETTSIDER

Arduino Documentation (2021) *Installing Additional Arduino Libraries.*

Tilgjengelig fra: <https://www.arduino.cc/en/guide/libraries>

(Hentet 18.04.2021)

³⁵ Harvard referansestil: <https://sokogskriv.no/referansestiler/harvard.html>

³⁶ Arduino Store starter kit <https://store.arduino.cc/genuino-starter-kit>

Arduino Software (2021) Downloads

Tilgjengelig fra: <https://www.arduino.cc/en/software>

(Hentet 18.04.2021)

Arduino Store (u.å) Tittel: " ARDUINO STARTER KIT MULTI-LANGUAGE"

Tilgjengelig fra: <https://store.arduino.cc/genuino-starter-kit>

(Hentet 25.04.2021)

Arduino Store (u.å) ARDUINO MKR WIFI 1010

Tilgjengelig fra: <https://store.arduino.cc/arduino-mkr-wifi-1010?queryID=undefined>

(Hentet 16.04.2021)

Atlassian (2021) Trello

Tilgjengelig fra: <https://www.trello.com/home>

(Hentet 31.05.2021)

Autodesk (2021) From mind to design in minutes

Tilgjengelig fra: <https://www.tinkercad.com/>

(Hentet 25.05.2021)

Biltema(2020) Koblingsboks

Tilgjengelig fra: <https://www.biltema.no/bygg/elinstallasjoner/elbokser/koblingsboks-2000021071>

(Hentet 01.06.2021)

Canonical Ltd.Ubuntu (2021) Download Ubuntu Desktop

Tilgjengelig fra: <http://ubuntu.com/download/desktop>

(Hentet 27.05.2021)

Elsikkerhet Norge AS (u.å) Brannfellene

Tilgjengelig fra:

<https://www.elsikkerhetsportalen.no/presserom/brannfellene/#:~:text=Hver%20fjerde%20brann%20i%20norske,belastning%20på%20sikringskursen%20øker%20brannfaren>

(Hentet 19.05.2021)

Flexify(2019) En rask introduksjon til Agile Metoder – med eksempler

Tilgjengelig fra: <https://www.flexify.no/blogg/agile-metoder>

(Hentet 16.04.2021)

GitHub, inc (2021) Where the world builds software

Tilgjengelig fra: <https://github.com>

(Hentet 31.05.2021)

HVL (u.å) DAT108 Programmering og webapplikasjoner

Tilgjengelig fra: <https://www.hvl.no/studier/studieprogram/emne/dat108>

(Hentet: 21.05.2021)

Kjell&Company (u.å) Arduino MKR GSM 1400 Utviklingskort

Tilgjengelig fra: <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/utviklingskort/arduino-mkr-gsm-1400-utviklingskort-p87191>

(Hentet 25.05.2021)

Kjell&Company (u.å) Arduino MKR Wifi 1010 utviklingskort

Tilgjengelig fra: <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/utviklingskort/arduino-mkr-wifi-1010-utviklingskort-p87194>

(Hentet 16.04.2021)

Kjell&Company (u.å) Koblingsbrett 400 hull

Tilgjengelig fra: <https://www.kjell.com/no/produkter/elektro-og-verktoy/elektronikk/elektroniklab/koblingsbrett-400-hull-p36285>

(Hentet 02.06.2021)

Kjell&Company (u.å) Temperatursensor med kabel for Arduino

Tilgjengelig fra: <https://www.kjell.com/no/produkter/elektro-og-verktoy/arduino/arduino-tilbehor/temperatursensor-med-kabel-for-arduino-p87081>

(Hentet 16.04.2021)

Leirvik Elektro (u.å)

Tilgjengelig fra: <https://www.leirvikelektro.no>

(Hentet: 16.04.2021)

Microsoft Office(2021) Welcome to Office

Tilgjengelig fra: <https://www.office.com>

(Hentet 31.05.2021)

Microsoft Azure (2021) Vi leverer verktøyene. Du forandrer verden.

Tilgjengelig fra: <https://azure.microsoft.com/nb-no/>

(Hentet: 25.05.2021)

Nes, Solveig Maria (2019) En kort introduksjon til Scrum

Tilgjengelig fra: <https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>

(Hentet 17.04.2021)

Open Source (u.å) UMLet 14.3 Free UML Tools for fast UML diagrams

Tilgjengelig fra: <https://www.umlet.com>

(Hentet 31.05.2021)

Oracle(u.å) Virtualbox

Tilgjengelig fra: <https://www.virtualbox.org/wiki/downloads>

(Hentet 27.05.2021)

Project hub (2019) Plant Communicator with MKR WiFi 1010

Tilgjengelig fra: https://create.arduino.cc/projecthub/Arduino_Genuino/plant-communicator-with-mkr-wifi-1010-efc92

(Hentet: 16.04.2021)

Smartbells AS (u.å) Smartbjella

Tilgjengelig fra: <https://smartbjella.no/produkt/>

(Hentet 16.04.2021)

The Apache Software Foundation (2021) Downloads

Hentet fra <https://tomee.apache.org/download.html>

(Hentet 31.05.2021)

The PostgreSQL Global Development Group(2021) PostgreSQL: Downloads

Hentet fra: <http://www.postgresql.org/download/>

(Hentet 31.05.2021)

UiB, UiO og HVL (2020) Harvard.

Tilgjengelig fra: <https://sokogskriv.no/referansestiler/harvard.html>

Hentet(18.05.2021)

Visma (2019) En kort introduksjon til Scrum

Tilgjengelig fra: <https://www.visma.no/blogg/en-kort-introduksjon-til-scrum/>

(Hentet 02.06.2021)

Vogella(2017) Java persistence API - Tutorial

Tilgjengelig fra:

<https://www.vogella.com/tutorials/JavaPersistenceAPI/article.html#:~:text=The%20Java%20Persistence%20API%20%28JPA%29%20is%20one%20possible,is%20a%20specification%20and%20several%20implementations%20are%20available.>

Hentet(27.05.2021)

Wikipedia(2020) GSM

tilgjengelig fra: <https://no.wikipedia.org/wiki/GSM>

(Lest 16.04.2021)

Wikipedia(2021) GRASP (object-oriented design)

Tilgjengelig fra: [https://en.wikipedia.org/wiki/ GRASP_%28object-oriented_design%29#:~:text=General%20Responsibility%20Assignment%20Software%20Patterns%20%28or%20Principles%29%2C%20abbreviated,in%20his%201997%20book%20Applying%20UML%20and%20Patterns.](https://en.wikipedia.org/wiki/GRASP_%28object-oriented_design%29#:~:text=General%20Responsibility%20Assignment%20Software%20Patterns%20%28or%20Principles%29%2C%20abbreviated,in%20his%201997%20book%20Applying%20UML%20and%20Patterns.)

(Hentet 16.01.2021)

Wikipedia (2021) Jakarta Servlet

Tilgjengelig fra: https://en.wikipedia.org/wiki/Jakarta_Servlet

(Hentet 16.04.2021)

Wikipedia(2021) Trådløst lokalt datanett

Tilgjengelig fra: https://no.wikipedia.org/wiki/Tr%C3%A5dl%C3%B8st_lokalt_datanett

(Hentet 16.04.2021)

10 APPENDIX

Prosjektet kan besøkes på adressen: <http://91.149.56.109:8081/Temperatur/>

Mobil: 12345678 Passord: leirvik

NB! Prosjekteier har blitt informert om at prosjektet kan bli testet av hvl- og ekstern sensor i perioden 04.06 – 10.06. Alle varslinger vil derfor bli sendt til prosjektet i denne perioden. Så dere er velkommen til å besøke prosjektet og teste applikasjonen.

Produkteier tester fremdeles temperaturmålerne og på grunn av dette kan flere temperaturmålinger i applikasjonen være unormalt høye.

På grunn av flytting kan server være nede opp til 1,5 timer.

Dersom sensorer vil kjøre applikasjonen på egen maskin, er det følgende fremgangsmåte:

- Java 8, Eclipse 2021-03 og TomEE Webprofile 7.1.4 må være installert
- Jarfiler må legges til i TomEE sin lib mappe
- Kildekoden med navn «Temperatur» som er lagt ved i WISEflow importeres til Eclipse
- I Kildekoden mappe src/META-INF/config.properties må e-post info fylles ut.
- Kildekoden sin mappe sqlscript inneholder script til å opprette skjema og tabeller i postgresql database.
- Skriv inn brukernavn og passord til egen database i filen src/META-INF/resources.xml.
- Data til databasetabellene kan legges til. Det er lagt ved i ett eget vedlegg i WISEflow, med navn sql.
- For å logge inn er det samme brukernavn og passord som beskrevet øverst.

Linker til jarfiler;

Eclipse link 2.7.1 https://www.eclipse.org/eclipselink/downloads/previous_releases.php

Postgresql-42.2.20 <https://jdbc.postgresql.org/download.html>

Javax.mail.jar https://javaee.github.io/javamail/#Download_JavaMail_Release

10.1 Risikoliste

Risiko	S	K	RF	Håndtering(er)
Klarer ikke å finne temperaturmålere som er egnet til å løse oppgave	3	5	15	Lete, kjøpe inn og teste temperaturmålere så tidlig som mulig
Elektronikk kan føre til interferens og ødelegge wifisignalet til temperaturmålere	2	3	6	Starte tidlig. Teste tidlig. Eventuelt be om hjelp fra prosjekteier, som muligens har kompetansen som trengs.
Brannfare grunnet feil montering av temperaturmåler i sikringsskap	1	5	5	Ha flere møter med prosjekteier for å sikre at alt er forsvarlig montert.
Mangel på kunnskap innen Arduino, IoT, nettverkskommunikasjon	5	3	15	Starte tidlig. Fordele oppgaver for å spesialisere seg slik at man lærer raskere. Alternativt, så må vi hente inn konsulenthjelp.
Oppstå problem med servere	3	2	6	Servere blir satt opp på virtuell maskin for at det skal være lett å sette opp igjen hvis noe går galt.
Miste data av diverse grunner	2	4	8	Ta sikkerhetskopier av data.
Sykdom eller frafall fra prosjektoppgaven	3	4	12	Alle parter skal ha de ressursene som trengs for å kunne drive prosjektet videre alene. Eventuelt må prosjektet nedjusteres.
Misforstå hva prosjekteier egentlig ønsker	3	3	9	Flere møter med prosjekteier. Vise resultat av framgang for å sikre at alt er riktig.
Ulik arbeidsfordeling	3	3	9	Prøve å fordele oppgavene likt.
Samarbeidsproblemer	3	3	9	Gruppen snakker sammen og prøver å løse problemene. Dersom ingen løsning, blir veileder involvert.
For lite tid.	3	4	12	Planlegge. Nedjustere prosjektet.

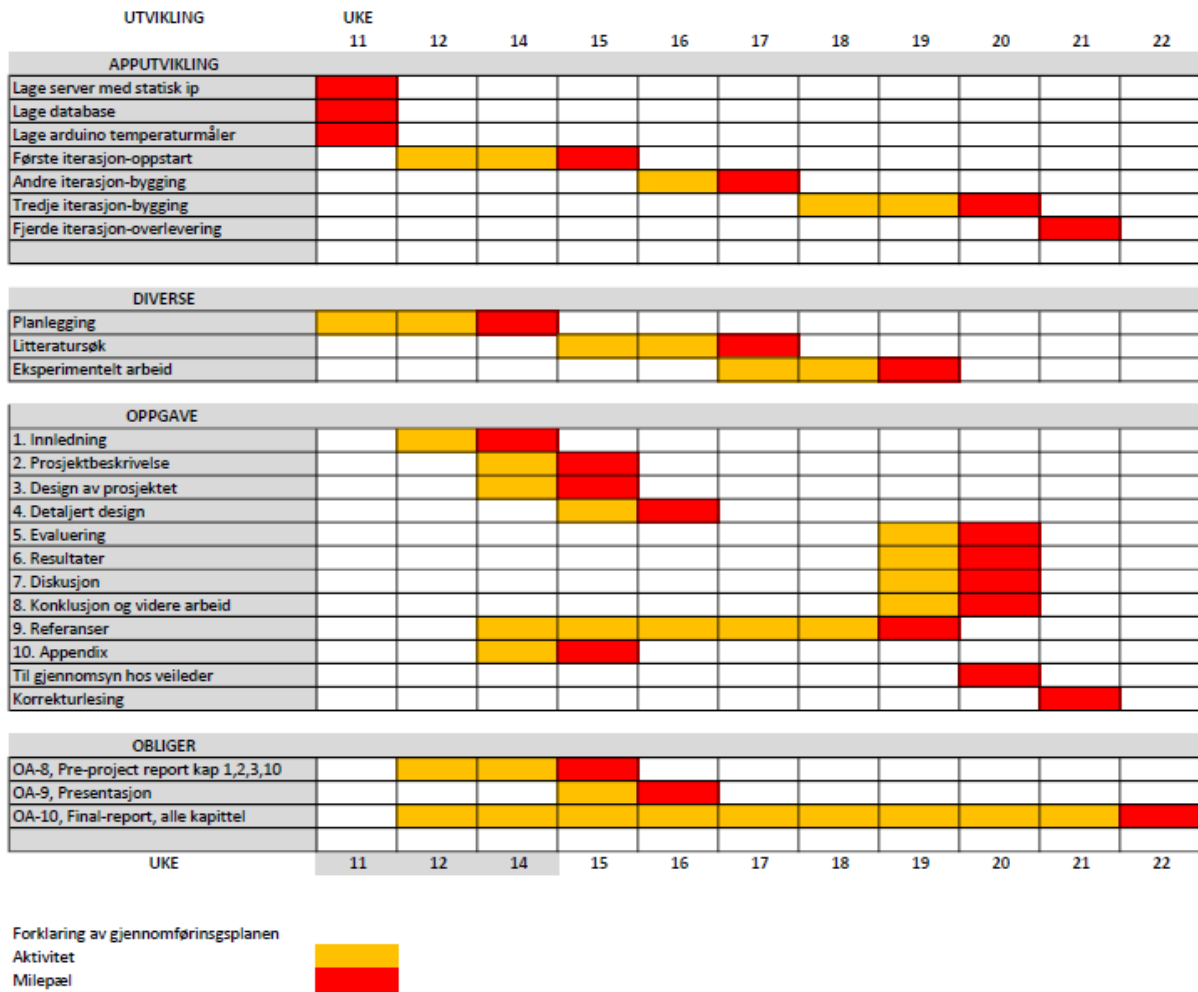
Tabell 10.1.1 Risikoliste

S = sannsynlighet for at risiko oppstår

K = alvorlighetsgrad på konsekvenser av risiko

RF = risk faktor (S multiplisert med K)

10.2 Gantt-skjema



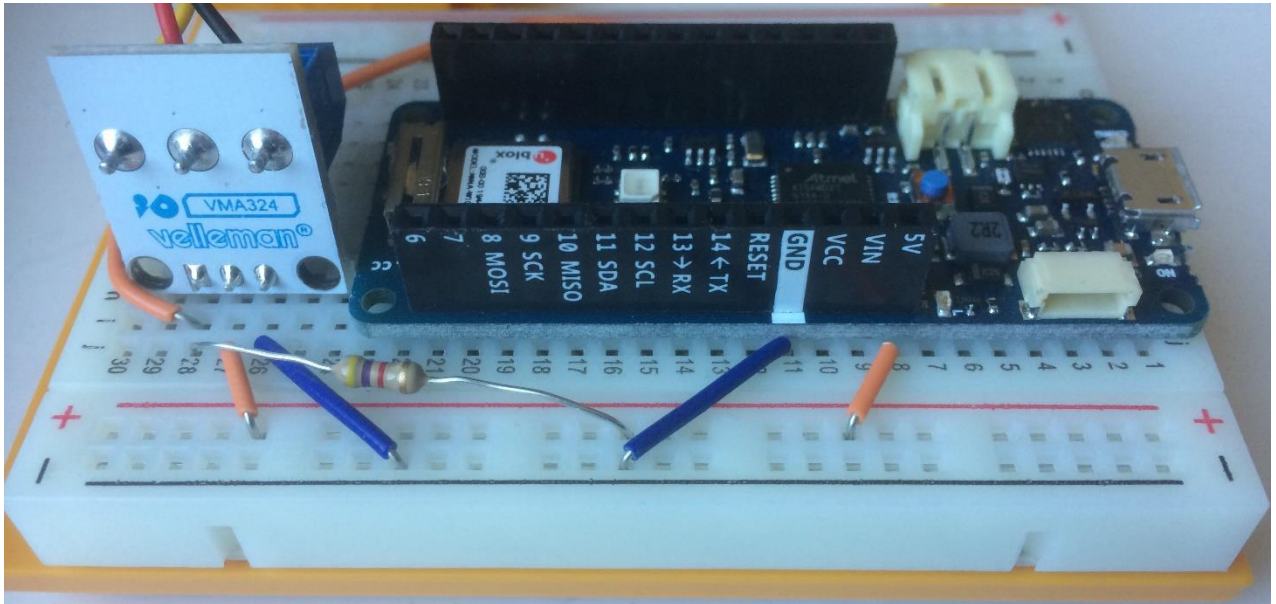
Tabell 10.2.1

10.3 Programmering og montering av temperaturmåler

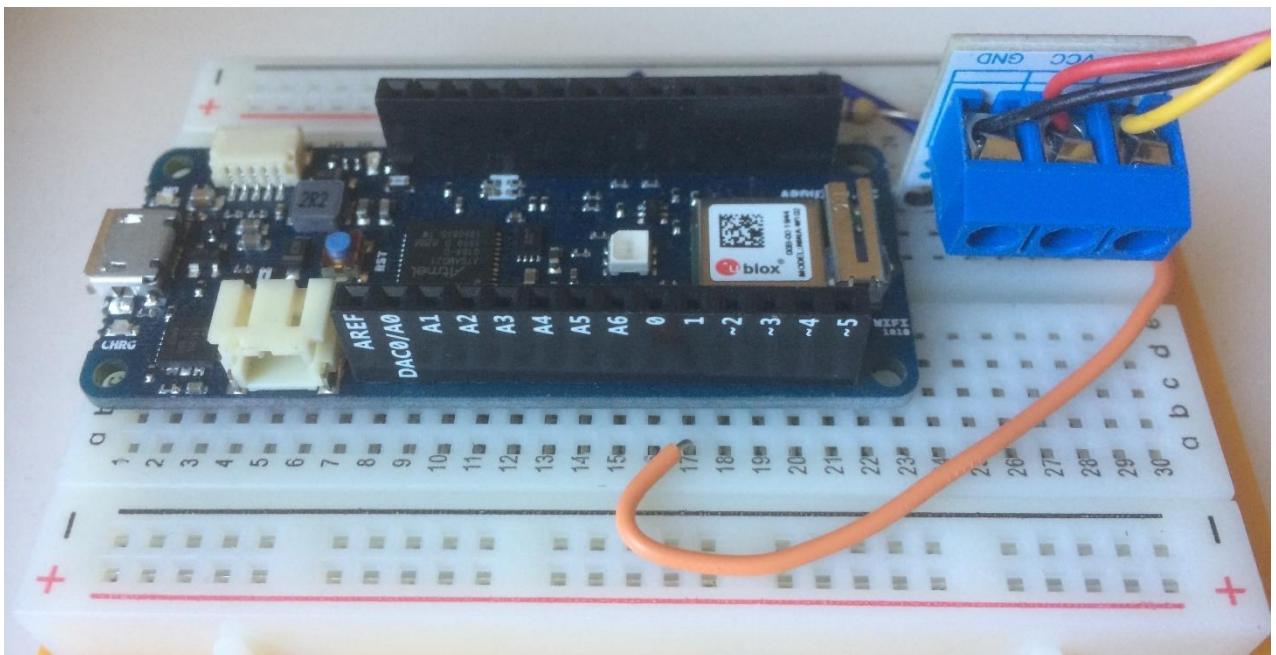
10.3.1 Programmering av temperaturmåler

Dette er fremgangsmåten for å legge til en Arduino temperaturmåler i prosjektet.

-Koble arduino kretser, som på bilde 10.3.1.1 og 10.3.1.2



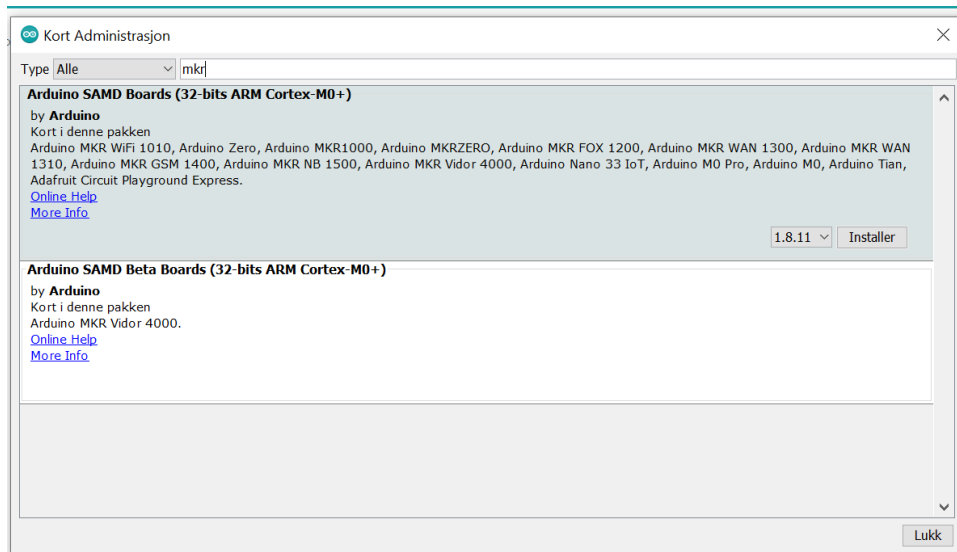
Bilde 10.3.1.1 Bildetekst Utgang 5v, Ground og koblinger mot temperatursensor



Bilde 10.3.1.2 Bildetekst Digital inngang 1

-Last ned Arduino IDE ³⁷(Arduino software, 2021).

-Installer bibliotek, som gjør at Arduino Mkr Wifi 1010 kan kommunisere med datamaskinen.



Bilde 10.3.1.3 Arduino Mkr wifi bibliotek 1010

-Legg til ytterligere biblioteket som skal brukes i kildekoden

WifiNina

WiFiSSLClient.

OneWire (Brukes av DS18B20)

DallasTemperaturer (Brukes av DS18B20)

-Rediger kildekoden som er utviklet. (Appendix Leirvik Elektro arduinokode)

Fyll inn kildekoden, med nettverksnavn, nettverkspassord, og et unikt navn slik at denne måleren kan gjenkjennes.

```
IPAddress server (91,149,56,109);
```

```
char ssid[] = "nettverksnavnet";
```

```
char password[] = "passordet";
```

```
String navn = "leia"; // De tre første bokstavene er kortnavn for firmaet, og fjerde bokstav er målernavnet.  
Så Leirvik elektro har lei +a,b,c,d osv.
```

-Overfør koden til arduino-brettet

-Koble til Arduino til mobillader 5v, og mikro usb.

³⁷ Arduino software <https://www.arduino.cc/en/software>

10.3.2 Montering av temperaturmåler i sikringsskap

Leirvik Elektro har informert om at det er kun autoriserte fagfolk som kan montere temperaturmåleren på innsiden av sikringsskapet. Prosjektet kan kun montere på utsiden og trekke kabel inn i sikringsskapet. Dette er enkelt, på de sikringsskap der inngangene er tilgjengelige. Dersom sikringsskapet er bygget inn i en vegg, så er tilgangen til inngangene sperret. Da må det finnes alternative løsninger, som å trekke kabelen inn gjennom døren. Leirvik Elektro anbefaler ikke at uautoriserte borrar hull i kabinettet.

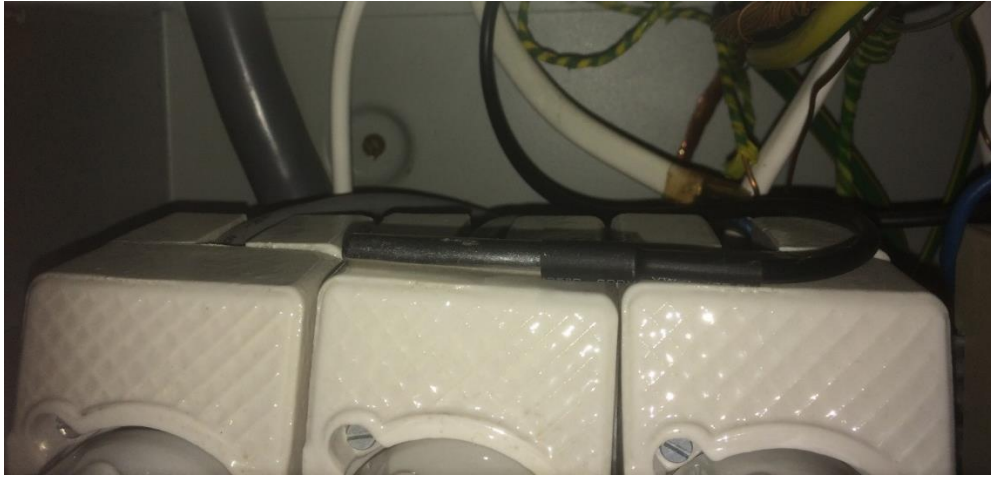
En annen utfordring med montering er det estetisk. Prosjektet har ingen god løsning på dette. En arduino med lader og kabel på utsiden av sikringsskapet er godt synlig. Prosjektet plasserte temperaturmålerne på toppen av sikringsskapet, slik som er gjort på bilde 9.3.2.1. Det antas at det er mulig å feste temperaturmålerene på siden, eller undersiden ved hjelp av dobbeltsidig tape.

Dekselet er en koblingsboks ³⁸(Biltema, 2020). På bunnen av koblingsboksen, er det limt et stykke med papp. Arduinoen sitt koblingsbrett er limt på pappen med dobbeltsidig teip. Se bilde 10.3.2.4



Bilde 10.3.2.1 Temperaturmåler i deksel på toppen av sikringsskapet

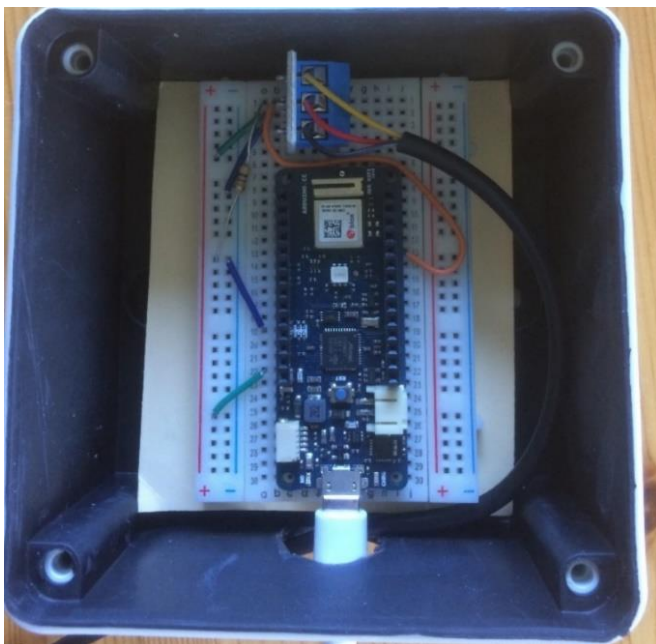
³⁸ Biltema koblingsboks <https://www.biltema.no/bygg/elinstallasjoner/elbokser/koblingsboks-2000021071>



Bilde 10.3.2.2 Nærbilde av temperatursensor



Bilde 10.3.2.3 Deksel utvendig



Bilde 10.3.2.4 Deksel innvendig

10.4 TESTING OG RESULTATER

10.4.1 Arduino og stabilitet

Hensikten med testene i dette kapitlet, er å forsikre seg om at Arduinoene er stabile og påitelige.

10.4.1.1 Testing av arduinomåler og strømbrudd

Hensikt: Vil test at Arduino er robust mot strømbrudd

Dato: første test 24.04.2021, siste test 25.05.2021

Antall ganger testen er utført: mer enn 10 ganger.

Test: Trekker ut strømadapteren fra stikk-kontakten, venter minst 30 sekunder, for å være sikker på at Arduinoen er slått av.

Resultat: Arduino fungerer som forventet. Når strømtilførselen blir gjenopprettet, fortsetter Arduinoen å sende temperaturmålinger.

Konklusjon: Arduinoene tåler strømbrudd.

10.4.1.2 Test av Arduino når den mangler kontakt med Wifi

Hensikt: Vil teste hvordan Arduino oppfører seg når den mangler kontakt med wifi.

Dato: første test 24.04.2021, siste test 25.05.2021

Antall ganger testen er utført: mer enn ti ganger.

Test: Arduinoene kjører og det sendes jevnlig temperaturer til databasen. Wifi blir slått av i fem minutter og så blir wifi slått på igjen.

Resultat: Arduino fungerer som forventet. Kort tid etter at wifi blir slått på igjen, sender temperaturmåleren til databasen som forventet

Konklusjon: Arduinoene er robust mot manglende wifi.

10.4.1.3 Testing av Arduino når den mangler kontakt med Server

Hensikt: Vil test at Arduino er robust mot manglende server

Dato: første test 24.04.2021, siste test 25.05.2021

Antall ganger testen er utført: mer enn ti ganger

Test: Appen kjører og det sendes jevnlig temperaturer til databasen. Server blir slått av i minst fem minutter og så blir server slått på igjen.

Resultat: Arduino fungerer som forventet. Arduino fortsetter å sende innen kort tid (Maximum ca. ett minutt) når server blir slått på igjen.

Konklusjon: Arduinoene tåler å være uten kontakt med server.

10.4.1.4 Konklusjon av arduino-testene

Konklusjon av testene i kapittel 1, er at Arduino er robust mot strømbrudd, manglende wifi og manglende kontakt med server.

Oppfølging, dato 25.05.2021. Arduinoene har nå vært i funksjon 24 timer i døgnet, den siste måneden, og det har ikke vært noen feil på arduinoene. De fungerer perfekt.

10.4.2 Test om temperaturmålinger er nøyaktige

10.4.2.1 Måsertest kvikksølvtermometer vs. Arduinomåler

Hensikt: Det skal finnes ut om temperaturmålingene fra arduinoene er stabile i forhold til ett kvikksølv termometer

Dato for første testkjøring 05.05.2021

Antall ganger testen er kjørt 2

Test: Det vil i 45 min bli utført jevnlig målinger fra arduino og sammenlignet med kvikksølvtermometer. Det blir utført ti målinger med fem minutters mellomrom. Prosjektet antar at dette er tilstrekkelig med målinger for å få gode resultat. Det blir i testene brukt ulike arduinomålere og ulik romtemperatur.

Tabell 10.4.2.1.1 viser alle resultater fra test nummer 1.

Tid	Arduino-måler	Kvikksølv-måler	Differanse
11.15	25,4	24	1,4
11.20	25,3	24	1,3
11.25	25,3	24	1,3
11.30	25,3	24	1,3
11.35	25,3	24	1,3
11.40	25,3	24	1,3
11.45	25,2	24	1,2
11.50	25,2	24	1,2
11.55	25,3	24	1,3
12.00	25,3	24	1,3

Tabell 10.4.2.1.1



Bilde 10.4.2.1 Testmiljø arduinomåler vs. kvikksølvtermometer

Når det gjelder den andre målertestene, så har prosjektet valgt å samle testresultatene i tabellen under. Det er ikke hensiktsmessig å ta med alle detaljer, slik som i test 1. Det er kun resultatet som er viktige.

Tabell 10.4.2.1.2 er en oversikt over alle testresultat.

Test nr.	Romtemperatur	Arduinomåler	Gjennomsnittsdifferanse	Største differanse
1	Ca. 24	leic	1,3	1,4
2	Ca. 15	leid	1,6	1,8

Tabell 10.4.2.1.2 Testresultat

Konklusjon

Arduinomåleren måler ca. 1,3-1,6 grader høyere enn kvikksølvtermometeret.

Arduinomåleren viser jevne målinger over en periode på 45 minutter.

10.4.2.2 Målertest arduinomåler vs. arduinomåler

Hensikt: Det skal finnes ut om temperaturmålingene fra to arduinoer er stabile

Dato for første testkjøring 25.05.2021

Antall ganger testen er kjørt 1

Test: Det vil bli utført jevnlig målinger fra arduinoene. Det blir utført målinger med minst en times mellomrom. Prosjektet antar at det er tilstrekkelig med åtte målinger for å få gode resultat.

Konklusjon: Temperaturmålingene viser jevne resultater når du sammenligner to arduinomålere.

Tid	Arduinomåler leic	Arduinomåler leid	ulikhet mellom arduinomålerene
18.00	23,9	23,9	0
19.00	23,9	23,9	0
20.00	23,9	24,0	0,05
21.00	24,1	24,1	0
22.00	24,1	24,1	0
08.00	24,19	24,25	0,06
09.00	24,3	24,3	0
10.00	24,12	24,25	0,13

Tabell 10.4.2.2.1 Testresultat

10.4.2.3 Varmetest

Siden tidligere sammenligning mellom arduino og kvikksølvtermometer er stabile, blir testene utført kun en gang. Det bør være tilstrekkelig for å gi en indikasjon på oppførsel ved høy varme. Dersom det indikeres en feil, blir det testet grundig.

10.4.2.4 2.3.1 Bilkupe

Hensikt: Sammenligne målinger på kvikksølvtermometer og arduinomåler ved høye temperaturer.

Dato: 31.05.2021

Antall ganger testen er kjørt: 1

Test: Plasserer termometrene i en bil, en solskinnsdag. Plasserer termometrene i skygge. Deretter følge med på målingene.

Resultat: Testen startet på rundt 30 grader. Da målte arduino ca 1,5 grad høyere enn kvikksølv. Etter som temperaturen steg mot 40 grader målte termometrene mer og mer likt.

Da temperaturen var 40 grader målte de likt. Når temperaturen stiger over 40, måler kvikksølvtermometeret høyere enn arduinoen. Når kvikksølv måler 45, er arduino 44

Konklusjon: Termometrene måler likt, når temperaturen er 40. Når temperaturen er over 45 grader, måler kvikksølv minst en grad høyere.

10.4.2.5 Konklusjon

Konklusjon av temperaturmålingene sin nøyaktighet er at arduinomålerene gir stabile og pålitelige målinger i forhold til hverandre. De gir også stabile og pålitelige målinger i forhold til ett kvikksølvtermometer. Arduinoene er ikke 100% nøyaktige i forhold til ett kvikksølvtermometer.

10.4.3 Sikkerhetskopiering

10.4.3.1 Sikkerhetskopiering av gjestemaskin

Sikkerhetskopidisk skal alltid inneholde: Programvaren Virtualbox, siste utgave, pluss extension pack og Windows 10 pro og lisensnøkkel

Hensikt: Vil teste å slette server/gjestemaskinen og gjenopprett gjestemaskinen i fra sikkerhetskopi.

Dato: Testen ble kjørt den 06.05.2021. Antall ganger testen er utført: 1

Test: Tok sikkerhetskopi med virtualbox. Hele gjestemaskinen ble kopiert til en ekstern disk. Deretter ble gjestemaskinen slettet fra vertsmaskinen. Gjenopprettet gjestemaskine ved å kopiere den fra eksterne disk og startet filen som ender med «.vbox». Serveren ble da automatisk lagt til i Virtualbox programmet og kunne startes på vanlig måte.

Resultat: Klarte å gjenopprette en gjestemaskin fra sikkerhetskopi.

Konklusjon: Prosjektet klarer å gjenopprette en gjestemaskin-server fra sikkerhetskopi.

10.4.3.2 Sikkerhetskopier og gjenopprett en database

Hensikt: Kontrollere at det er mulig å ta sikkerhetskopi av databasen, og gjenopprette en database fra sikkerhetskopi.

Dato: Testen ble kjørt den 25.05.2021

Antall ganger testen er utført: 1

Test: Alle kommandoene blir gitt i terminal. Start med å forandre innlogget bruker til en bruker som er registrert i postgres databasen. I dette tilfellet postgres sin superuser «postgres». Kommando: «sudo su – postgres»

gi kommandoen: «pg_dump postgres > postgres_db.bak», som sikkerhetskopierer databasen.

Deretter slettes alle tabellene i databasen. For å gjenopprette databasen, gis kommandoen:

«psql postgres < postgres_db.bak»

Resultat: Hele databasen blir sikkerhetskopiert, slettet og gjenopprettet

Konklusjon: Det er lett å gjenopprette en database fra sikkerhetskopi

10.4.4 Testing av webapplikasjonen

10.4.4.1 Test applikasjonen på tom database.

Hensikt: Vil leite etter «nullpointer exceptions» som krasjer applikasjonen.

Dato: Testen ble kjørt den 05.05.2021. Antall ganger testen ble utført: 1

Test: Alle temperaturmålingene ble fjernet, og applikasjonen ble startet. Deretter ble alle jsp og html sider åpnet.

Resultat: Ingen sider krasjet.

Resultat: Applikasjonen er robust mot tom database og «nullpointer exceptions».

10.4.4.2 Sende temperaturmålinger fra en måler som ikke finnes i databasen

Hensikt: Vil teste om det blir komplikasjoner dersom en temperaturmåler som ikke er kjent av server, starter å sende temperaturmålinger til server. Menneskelig svikt, kan resultere i at en temperaturmåler blir programmert med ett navn som ikke er kjent av server. En hacker kan også finne på å sende en modifisert request til server.

Dato: Testen ble kjørt den 25.05.2021

Antall ganger testen er utført: mer en en gang pr navn.

Test: En arduinomåler blir gitt ett navn som ikke er kjent av server. Så blir arduinoen koblet til server og starer å sende regelmessige målinger til server.

Dette er navnene som er testet: «abcd», «», «æøå», «1234», «@£\$+».

Resultat: Ukjente målernavn har ingen påvirkning på webapplikasjonen. Databasen oppfører seg også normalt.

Konklusjon: Server er robust, mot ukjente temperaturmålerer.

10.4.4.3 Hvordan takler applikasjonen sql angrep

Hensikt: Vil teste om en hacker kan sende sql kommandoer til databasen, via en forespørsel til ArduinoServlet.

Dato: Testen ble kjørt den 02.06.2021

Antall ganger testen er utført: 1

Test: Koden på en arduinomåler blir forandret. I stedet for å sende temperaturer, blir det sendt en string «hello».

Resultat: Server tar ikke imot en forespørsel som inneholder tekst. Kun tall blir godkjent.

Konklusjon: Server er robust mot sql angrep.

10.4.5 Testing av varsling

10.4.5.1 Test av Måleralarm

Hensikt: Kontrollere at det blir sendt varsling dersom en måler slutter å sende målinger.

Dato for første test 11.05.2021. Dato for siste test 25.05.2021

Antall ganger testen er utført: Mer enn 20 ganger.

Test: En måler blir slått av, ved at strømmen blir fjernet. Deretter ventes det i en time. Da skal det sendes en e-post om at målereren ikke lenger sender.

Resultat: Det ble varslet etter en time. Det blir kun sendt en mail. Når måleren kobles til strøm igjen, sender den som forventet, og webapplikasjonen har registrert at måleren er aktiv.

Konklusjon: Måleralarm og reaktivermåler virker.

10.4.5.2 Test av Temperaturalarm

Hensikt: Kontrollere at det blir sendt varsling dersom en målt temperatur stiger over varslingsgrensen, som ved testgjennomføringene er 40 grader.

Dato for første test 11.05.2021. Dato for siste test 25.05.2021

Antall ganger testen er utført: Mer enn 20 ganger

Test: En måler sender en temperaturmåling, høyere enn varselgrensen. Varsling er satt til aktiv. Webapplikasjonen kjører temperaturalarm

Resultat: Temperaturalarm registrerer at det skal varsles. Sender e-post med varsel, til produkteier, og øker varselgrensen med +1 grad.

Konklusjon: Temperaturalarm virker.

10.4.5.3 Konklusjon

Varsling fungerer som forventet.

10.5 Leirvik Elektro arduinokode

```
//Arduino mkr wifi 1010v og DS18B20 digital temperaturmåler
#include <WiFiNINA.h>
#include <WiFiSSLClient.h>
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 1

// Create a new instance of the oneWire class to communicate with any OneWire device:
OneWire oneWire(ONE_WIRE_BUS);

DallasTemperature sensors(&oneWire); // Pass the oneWire reference to
DallasTemperature library:
char ssid[] = "nettverksnavn"; // your network SSID (name)
char password[] = "passord"; // your network password
IPAddress server(91,149,56,109);
String navn = "leic"; // Id name of temperature device used for testing
int port = 8081;
int readingInterval = 60000; // How often arduino sends data to server.
float temperatureC;
int tempPin = 0;
int status = WL_IDLE_STATUS;
String postData = "empty";
String url = "/Temperatur/ArduinoServlet";
WiFiClient client;
void setup() {
  Serial.begin(9600);
  sensors.begin(); //Starter temperatursensoren
  Serial.println("STARTER");
  while (status != WL_CONNECTED) {
    status = WiFi.begin(ssid, password);
    delay(10000);
  }
}
} //setup
```

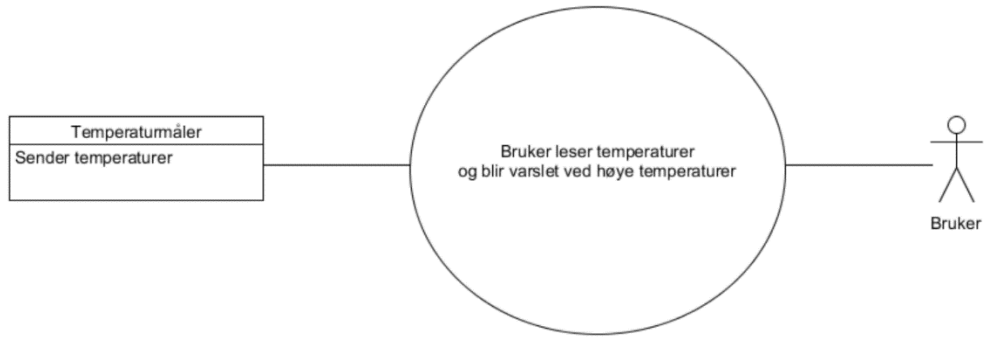
```

void readTemp()
{
  sensors.requestTemperatures(); // Fetch the temperature in degrees Celsius for device
index:
  temperatureC = sensors.getTempCByIndex(0); // the index 0 refers to the first device
  Serial.println(temperatureC);
}
void post()
{
  postData = "navn=" + navn + "&" + "temp=" + String(temperatureC);
  if (client.connect(server, port)) {
    client.println("POST " + url + " HTTP/1.1");
    client.println("Host: 91.149.56.109");
    client.println("User-Agent: Arduino/1.0");
    client.println("Connection: close");
    client.println("Content-Type: application/x-www-form-urlencoded;");
    client.print("Content-Length: ");
    client.println(postData.length());
    client.println();
    client.println(postData);
  } else {
  }
}
void loop() {
  readTemp(); // Reads temperature and updates temperatureC
  post();
  if (WiFi.begin(ssid, password) != WL_CONNECTED) {
    status = WiFi.begin(ssid, password);
  }
  delay(readingInterval);
}

```

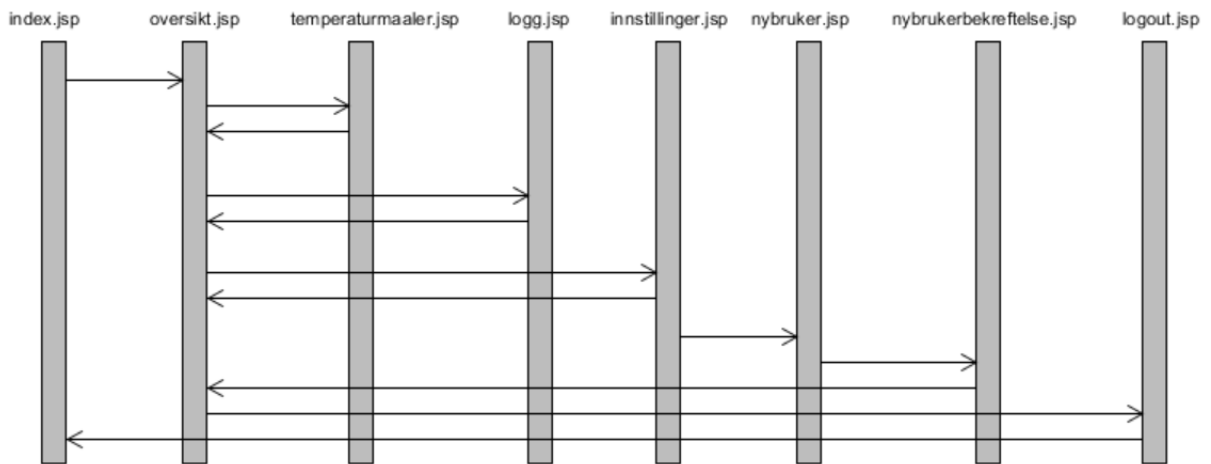
10.6 ARBEIDSTEGNINGER

10.6.1 Brukstilfelle



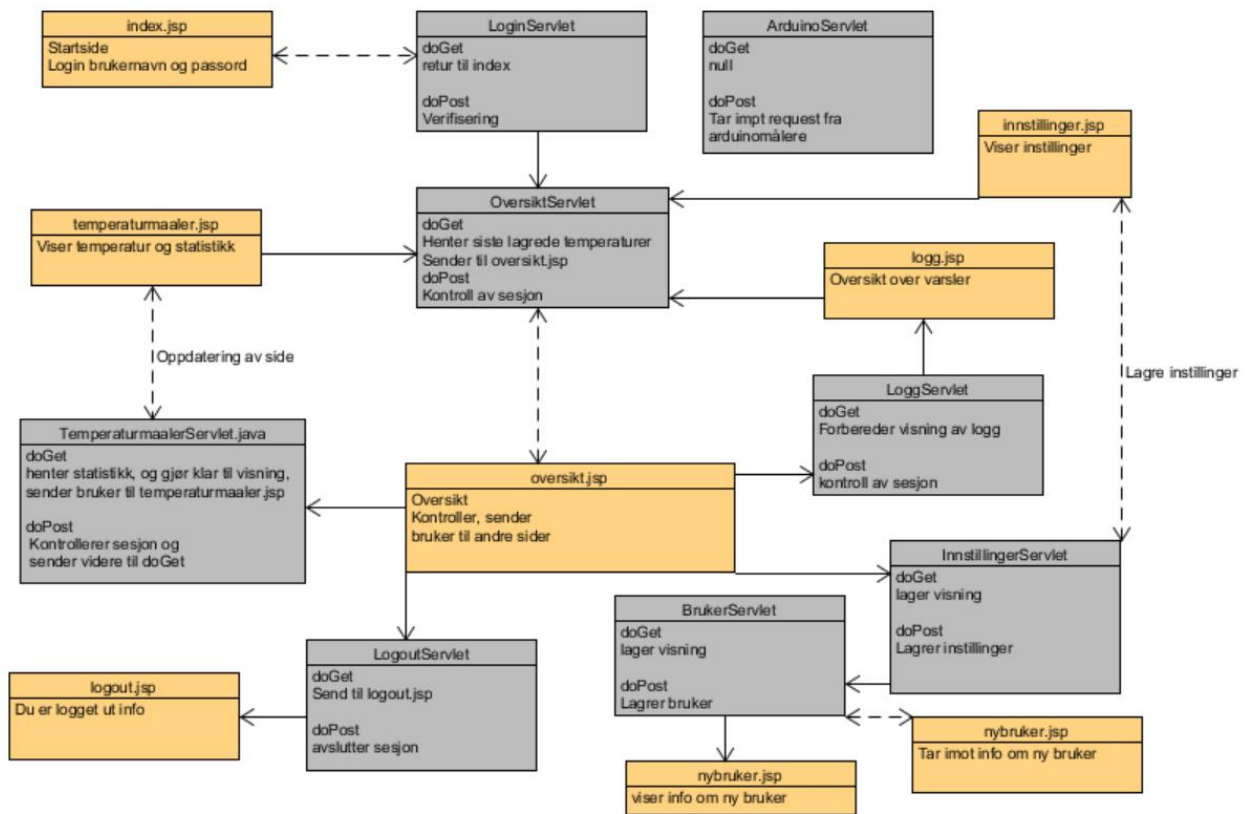
Figur 10.6.1.1 Brukstilfelle

10.6.2 Sekvensdiagram og jsp oversikt



Figur 10.6.1.2 Sekvensdiagram og jsp oversikt

10.6.3 Servletoversikt



Figur 10.6.3.1 Servletoversikt