



3. Semester Projekt

Gruppe 4

3A

Adam Shumiye Ziebe

Dina Johanne Sode Watson

Oliver Boots

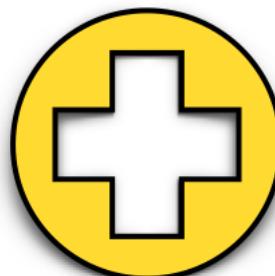
Signe Sjørslev Hansen

Organisation:

Københavns Kommunes Tandpleje

21. December 2023

Københavns Erhvervsakademi



Vejledere:

Per Fogt

Dan Madsen

Chalie Demasi

kea
COPENHAGEN SCHOOL OF DESIGN
AND TECHNOLOGY

Anslag
Med mellemrum:

70.089

1. Resumé

I begyndelsen af projektet præsenteredes en projektbeskrivelse for de undervisere, der har det overordnede ansvar for accept af projektet. Efter accept foretog projektgruppen den nødvendige research for at blive klogere på området. Derefter afholdtes et interview med en fagperson, som gav et bedre indblik i de behov og ønsker, målgruppen for løsningen har.

Projektgruppen begyndte at udarbejde det, der til slut blev den færdige prototype af løsningen. Undervejs blev der gjort erfaringer i forhold til at skifte en sensor ud, som oprindeligt skulle have været med, til fordel for en anden sensor. Til slut endte den samlede løsning med at virke: Der kunne ske kommunikation mellem device og webserver over et dedikeret netværk, som kunne vise data på et website. Det var målsætningen for projektet, og denne betragtes som værende opfyldt.

2. Indholdsfortegnelse

| | |
|---|----|
| 1. Resumé..... | 3 |
| 2. Indholdsfortegnelse..... | 4 |
| 3. Indledning | 6 |
| 4. Projektbeskrivelse | 7 |
| 4.1 Afgrænsning af scope..... | 7 |
| 5. Indledende undersøgelse..... | 8 |
| 5.1 Ideudvikling..... | 8 |
| 5.2 Research | 8 |
| 5.3 Brugerundersøgelse | 9 |
| 5.3.1 Databehandling af interview | 9 |
| 5.4 Persona og User stories | 10 |
| 5.4.1 Userstories - Device..... | 10 |
| 5.4.1 Userstory - Web-baseret del af løsningen | 12 |
| 6. Kravspecifikation og accepttest metode | 13 |
| 6.1 Kravspecifikationer - Device..... | 13 |
| 6.2 Kravspecifikationer - Netværk | 17 |
| 6.3 Rationale for prioritering af krav..... | 18 |
| 7. Analyse | 19 |
| 7.1 Arkitektur af pathfinding..... | 19 |
| 7.1.1 Blokdiagram..... | 19 |
| 7.1.2 Diagram for IR-temp-sensor | 20 |
| 7.1.3 Flowchart for programmering..... | 20 |
| 7.1.4 Første version af netværksopsætning..... | 21 |
| 7.1.5 Endelige version af netværksopsætningen | 22 |
| 7.2 Den endelige arkitektur | 22 |
| 7.2.1 Protokoller..... | 22 |
| 7.2.1.1 HTTP | 22 |
| 7.2.1.2 SMTP..... | 22 |
| 7.2.1.3 I2C | 22 |
| 7.2.1.4 OSPF | 23 |
| 7.2.2 BOM-liste | 23 |
| 7.2.3 Udvalgte programmeringsblokke..... | 25 |
| 8. Løsningsdesign | 26 |
| 8.1 Router/Switch opsætning og Interface forbindelser | 26 |
| 8.1.1 Konfiguration af Cisco-router 1941-series | 26 |
| 8.1.2 Konfiguration af switch | 27 |
| 8.2 Kodebeskrivelse | 29 |
| 8.2.1: Logging..... | 29 |
| 8.2.2: Cookies | 32 |

| | |
|---|----|
| 8.2.3: SMTP | 33 |
| 8.2.4 Hashing | 34 |
| 8.2.5: Encryption / Decryption..... | 35 |
| 8.2.6 Main.py på esp32 | 37 |
| 8.2.7: Ldr_reading.py på esp32..... | 41 |
| 8.2.8: Ir_test_works.py på esp32 | 42 |
| 8.2.9: Network_test_1 på esp32..... | 43 |
| 8.3 Konfiguration af valgte servere | 46 |
| 8.3.1 Hardeninglist af server..... | 47 |
| 8.4 Dokumentation af sikkerhedstiltag..... | 48 |
| 8.4.1 Kryptering af databasen | 48 |
| 8.4.2 Regex/SQL-injection-sikring ved login | 49 |
| 8.4.3 Multi-factor Authentication ved login | 49 |
| 9. Test af løsning | 50 |
| 9.1 Pathfinding testresultater | 50 |
| 9.2 Dokumentation af DUT | 50 |
| 9.2.1 Billeder og video af samlet løsning | 50 |
| 9.3 Udførel af test | 51 |
| 9.4 Udførel af Accepttest på DUT | 52 |
| 9.4.1 Første testcyklus..... | 52 |
| 9.4.2 Netværk..... | 59 |
| 9.4.3 Anden testcyklus:..... | 61 |
| 10. Implementering af løsning i drift | 68 |
| 11. Praktisk projektplanlægning og ledelse | 69 |
| 11.1 WBS | 69 |
| 11.2 Kanban..... | 69 |
| 11.3 Virksomhedsanalyse..... | 69 |
| 12. Konklusion | 71 |
| 13. Projektforløbet | 72 |
| 14. Perspektivering..... | 73 |
| 15. Litteraturliste | 74 |
| 15.1 Kilder fundet på internettet..... | 74 |
| 15.2 Personrelaterede kilder | 75 |
| 15.3 Kilder relateret til BOM-liste | 75 |
| 16. Bilag | 77 |
| 16.1 Bilags-oversigt..... | 77 |
| 16.1.1 Transkription af interview med Cecilie | 78 |
| 16.1.2 WBS..... | 83 |
| 16.1.3 Kanban | 84 |

3. Indledning

Nutidens befolkning bliver i højere og højere grad utsat for IoT-udstyr i dagligdagen. Overalt findes smarte enheder, som har forskellige indvirkning i aspekter af livet. Dette gør sig også gældende i sundhedssektoren, hvor IoT-udstyr for alvor er ved at vinde indpas i behandlingen af patienter. [8]

I forbindelse med behandling og pleje af ældre borgere, er der et ønske om, at de forbliver i eget hjem så længe som muligt. Det efterlader en del af borgerne i en gråzone, hvor modtagelse af hjemmepleje ikke er den komplette løsning, men samtidig hvor en plejehjemsplads heller ikke en realitet. [10]

Fokusgruppen for projektet har været mennesker med demens. Forløbet mennesker med demens gennemlever er meget unikt, og det påvirker dem i forskellig grad og på forskellig vis. Fælles for dem er, at personlig pleje ubevidst nedprioriteres i takt med, at sygdomsforløbet skrider fremad. Ting der engang har været en selvfølge bliver pludselig en pligt, som glider ud af bevidstheden. Dette gør sig også gældende for tandpleje, hvilket kan have alvorlige konsekvenser for borgernes livskvalitet. [13]

På baggrund af samtaler med fagpersoner valgte projektgruppen at udarbejde en IoHT-løsning, som skal hjælpe borgere med demens med at huske tandplejen. Forhåbningen er at forbedre livskvaliteten hos de borgere, der befinner sig i gråzonan mellem plejehjem og hjemmehjælp, ved hjælp af bedre tandhygiejne og dermed færre helbredskomplikationer.

4. Projektbeskrivelse

Som et ældre menneske med demens kan det være problematisk at opretholde det samme niveau af sanitet, som man har haft tidligere i livet. At huske de forskellige ting, der skal gøres, kan være en udfordring, og det giver anledning til problematikker i forbindelse med tandpleje. For at gøre tandplejen lettere, vil vi udvikle en smart tandbørsteholder, der gennem nudging med påmindelser og andre hjælpeteknikker gør det lettere for borgeren at huske tandbørstning. Tandbørsteholderen er kompatibel med de mest almindelige tandbørster.

Vi ønsker at lave en tandbørsteholder, der kan indikere og dokumentere, om tandbørstningen er foretaget, samt sender dataen til en server. Denne data bliver vist på et website, som kan tilgås af plejepersonalet.

Til projektet vil vi bruge følgende:

Aktuator:

- LED/LED-ring: Indikering af tid som der skal børstes, samt om der er blevet børstet i dag (de to gange der skal børstes)
- Buzzer: Påmindelse om at børste, samt fjernstyring fra den lokale server.

Sensor:

- Lyssensor: Bruges til at detektere, når tandbørsten er væk fra holderen.
- Varmesensor: Bruges til at bestemme, om tandbørsten reelt har været holdt i hånden, eller om den har været lagt ned.

Interview med fagperson: Tandlæge

Kategori: Telemedicin

Tidsestimat: 6 uger

4.1 Afgrænsning af scope

På baggrund af den satte tidsramme, har projektgruppen været nødsaget til at begrænse omfanget af projektet. Såfremt der havde været mere tid, havde det været muligt at forbedre og tilføje features til projektet, som der grundet tidsrammen ikke har været mulighed for. Der har desuden været begrænsning på midler, projektgruppen har haft til rådighed under forløbet, samt ændringer i kravene vi arbejder under. Med flere ressourcer økonomisk havde det været muligt at købe ind bedre udstyr, som overordnet havde givet løsningen et pænere finish og mere avancerede features.

På den baggrund opfordres til, at der tages højde for den begrænsning på midler og tid, som projektet er udarbejdet under.

5. Indledende undersøgelse

5.1 Ideudvikling

Idéudviklingsprocessen foregik således, at arbejdsgruppen i projektforløbets begyndelse afholdte en brainstorm, som skulle give anledning til gode projektforslag. Efter brainstormen afholdtes en afstemning, hvor gruppemedlemmerne gav forskellige forslag en score, som derefter blev lagt sammen. Idéen med flest point vandt afstemningen, og det blev til det IoT-enhed, som projektet omhandler. Idéen udsprang oprindeligt af, at en person fra et af gruppemedlemmernes omgangskreds havde givet udtryk for, at der er et behov for en smart enhed i forbindelse med tandpleje hos mennesker med demens. På baggrund af det behov og de krav, der var sat for projektet i forhold til sensorer og aktuatorer, blev de ønskede funktionaliteter for enheden udvalgt. Der blev også taget højde for den begrænsede tidsperiode og dedikeret arbejdstid til projektet, og det skar ned i scope størrelse.

5.2 Research

Efter beslutningen var taget om, hvilken retning projektet skulle føres i, startede den fase, hvor der blev foretaget research inden for emnet. En smart-tandbørste i forbindelse med IoT-healthcare er noget, der allerede er forsøgt implementeret flere steder. Det ses blandt andet i et projekt udført af Anna Flagg, Jennifer Boger og Alex Mihailidis, som havde fokus på at skabe en løsning, der vejlede mennesker med demens igennem processen med at børste tænder. [1] Ligeledes blev der foretaget et pilot-study og forsøgt med implementering af en smart-tandbørste til personer med demens af Mohammad Jannati, Sarah O’Byrne og Zahra Moussavi. Formålet var at skabe et device, som kan forlænge tiden, ældre mennesker kan være hjemmeboende inden, at de skal på plejehjem på baggrund af, at de ikke længere kan tage vare på sig selv. [3]

Der var på forhånd stillet krav til, at løsningen skulle have to aktuatorer og to sensorer. Det dannede grundlag for research af komponenter. Samtidig var arbejdsgruppen også bevidst om den begrænsede tidshorisont, der var sat for projektet. Det gjorde, at scopet for arbejdsbyrden også skulle være passende.

I research-stadiet foretog arbejdsgruppen et interview med en fagperson, som havde input i forhold til, hvilke funktioner løsningen kunne have. Det gav anledning til en kursændring i hvilke sensorer der oprindeligt var valgt til løsningen. Fagpersonen udleverede også et studie foretaget af vedkommende selv til arbejdsgruppen, der kunne bruges som inspiration. [13]

5.3 Brugerundersøgelse

I forbindelse med projektet foretog projektgruppen et interview med en fagperson for at skabe bedre forståelse for den efterspørgsel, der er i plejesektoren. Fagpersonen der blev interviewet er en tandlæge, som har praksis i Københavns Kommunes tandpleje. Dette gør hende og hendes klinik til potentielle brugere af løsningen. Vedkommende var relevant for projektet, fordi hun har indblik i hvilke problematikker, der opleves i praksis. En mere dybdegående analyse af interviewet foretages i afsnittet nedenfor.

5.3.1 Databehandling af interview

Der blev i forbindelse med research omkring emnet afholdt et interview med Cecilie Dahl Eriksen, som er tandlæge hos ‘Voksentandplejen Københavns Kommune’. Den fulde transskription af interviewet kan ses i bilag 1.

Cecilie fortæller, at hos Voksentandplejen er der ansat mellem 60 og 70 personer. De tilser tilsammen over 300 patienter månedligt. Patienterne er mennesker, som ikke er i stand til at gå hos en almindelig tandlæge. Det kan være af en lang række årsager, blandt andet psykisk sygdom, udviklingshæmmedes eller demens.

I projektet er fokusgruppen mennesker med demens.

Cecilie fortæller, at manglende fokus på tandplejen hos ældre mennesker er et problem, de ofte oplever. Hun fortæller, at cirka 70% af de ældre har behov for hjælp med tandpleje, men at kun 5-10% reelt modtager det. På nuværende tidspunkt er der ikke noget decideret udstyr til at afhjælpe problemet på trods af, at der har været forskellige bud på dette. Cecilie beskriver, at det ville hjælpe med en påmindelse - også for plejepersonalet, der skal være opmærksomme på at huske at tjekke, om tandbørstningen er blevet foretaget. Et gruppemedlem præsenterede herefter projektet for hende for at få feedback i forhold til, om løsningen kunne være brugbar i praksis. Hun svarede, at det ville være godt, hvis det var muligt at måle, hvorvidt tandbørsten reelt har været i brug, eller bare har været lagt uden for holderen. Da interviewet blev afholdt, var idéen at dokumentere forløbet med billeder. Efter feedback valgte projektgruppen at omlægge brugen af kamera til en infrarød sensor, der skal måle, om tandbørsten har været holdt i hånden, eller om den har været lagt ned. Et kamera ville have skabt problematikker omkring GDPR-loven og ud fra det fravalgte vi denne tilgang.

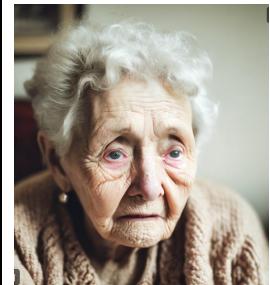
Ud fra interviewet kan man gøre sig en forsiktig udledelse der lyder på, at løsningen appellerer til et område af HealthCare, hvor der reelt er behov for den. De problematikker Cecilie oplever som fagperson, kan løsningen potentielt være med til at afhjælpe. Det var nyttigt at snakke med hende og modtage feedback på vores idéer, og det gav anledning til en kursændring i forhold til de sensorer, der bliver brugt i projektet.

5.4 Persona og User stories

5.4.1 Userstories - Device

Userstory 1.1 - Bente

Som ældre kan det være svært at overkomme den samme slags opgaver, man kunne før i tiden. Når man stadig er hjemmeboende og ikke længere er på plejehjem, skal man stadig i stor grad holde hus og hjem på samme måde som før, men med mindre kræfter at tage af. Efter jeg har lavet mad til mig selv og holdt huset, har jeg ikke altid overskud til personlig hygiejne. Hjemmehjælpen hjælper til med at komme i bad, men ofte svipser tandbørstningen, fordi den går glemt i processen. Hjemmeplejen har ikke tid til at komme og hjælpe med tandplejen to gange om dagen, så derfor ville det være smart med en tandbørste, der hjælper mig med at huske på tandbørstningen.

Story nr. 1.1**Prioritet: 4****Effort: 4**

Userstory 2.1 - Lars Ole

Jeg fik konstateret begyndende demens for et halvt år tilbage. Jeg er endnu ikke dybt plaget af det i en grad, hvor jeg skal på plejehjem, men jeg kan godt mærke, at min hukommelse ikke er, hvad den har været. Min kone døde for nogle år tilbage, så det er bare mig i vores hus, og jeg skal selv huske på de ting, der skal gøres. Jeg har hjemmepleje på en gang om dagen, så de kan hjælpe med medicin og støttestrømper. De har bare ikke alverdens tid, da de også har andre patienter at se til. Derfor er det ofte kun en kort overflyvning, når de er her. Det ville være praktisk at have ting i hjemmet, som kan huske en på forskellige opgaver, der ellers ville ryge i svinget. Jeg har ofte tandkødsbetændelse, og det gør ondt. Det sker, fordi jeg glemmer at børste tænder. Det kunne man undgå ved at huske det, men det er svært. Jeg tror, at jeg ville have god gavn af en tandbørste, der kunne hjælpe mig med at huske det.

Story nr. 2.1**Prioritet: 4****Effort: 3**

Userstory 3.1 - Birger

Jeg er en ældre mand, som bor sammen med min kone Johanne, der har fået konstateret demens for et stykke tid tilbage. Det er fremskredent, og hun kan ikke huske nær så godt, som hun engang har kunne. Jeg har meget at se til, da jeg stadig arbejder, og jeg skal også varetage alle de daglige ting som f.eks. Madlavning, rengøring og nu også pasning af hende. Hun kan godt selv finde ud af at gøre ting, men jeg skal huske hende på det. Det ville være rart med en smart tandbørste, som kunne huske hende på at gøre det, når det smutter for mig at huske hende på det. Det ville tage en byrde fra mine skuldre, hvis man kunne gøre det på en smartere måde, hvor jeg ikke længere skal huske på at huske hende på det.

Story nr. 3.1**Prioritet: 5****Effort: 4**

Userstory 4.1 - Johanne

Jeg er Birgers kone, og har fremskreden demens. Jeg husker ikke længere at gøre de samme ting som før, og han skal i høj grad hjælpe mig med at huske at gøre ting, som før var en selvfølge. Jeg kan mærke, at det er hårdt for ham at skulle bære vægten af at huske mig på ting, som aldrig før har været tilfældet, og det bliver kun tiltagende herfra. Jeg ville ønske, at det ikke skulle være så svært for ham - Jeg kan bare ikke gøre for, at jeg er i den tilstand. Jeg har altid gået meget op i at se præsentabel ud, men bad og tandbørstning kommer ikke så naturligt til mig som før i tiden, og det plager mig. Jeg kan sagtens udføre begge dele, jeg glemmer bare at gøre det. Jeg ville ønske, at man kunne få noget, der kunne minde en om at gøre det, så Birger ikke længere skal tænke på det, og så jeg kan forbedre min hygiejne.

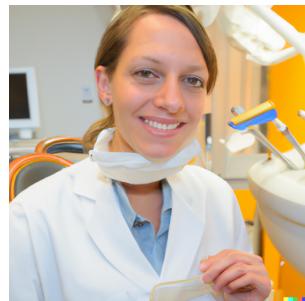
Story nr. 4.1**Prioritet: 4****Effort: 2**

Billeder: <https://openart.ai/create>

5.4.1 Userstory - Web-baseret del af løsningen

Userstory 1.2 - Anette

Jeg er tandlæge i sundhedssektoren. Det er altid frustrerende at se patienter med problemer, som ikke har overskuddet til at tage sig af dem for at komme i bedring. Jeg ville ønske, at vi i sundhedssektoren kunne gøre mere for den enkelte borgeres behov, men der er ganske enkelt ikke tid eller hænder nok til at løfte opgaven. IoHT-enheder er mere og mere på vej ind i sundhedssektoren, og jeg glæder mig til den dag, vi også kan få det implementeret hjemme hos borgerne som et plaster på såret for den tid, vi ikke har overskud til at give dem hjælp. Jeg tror, at de borgere jeg tilser ville have meget god gavn af en smart tandbørsteholder, som kan minde dem om at tage sig af den personlige hygiejne.

Story nr. 1.2**Prioritet: 4****Effort: 4**

Billede: <https://labs.openai.com/e/F5HFQDUIZbi5pna6uNeTIMSp/WOHbCkN81ju04vRXgFQgcJ9h>

6. Kravspecifikation og accepttest metode

6.1 Kravspecifikationer - Device

| | | |
|-----------------------------|---|----------------------|
| ID:1.1 | Krav: Løsningen skal kunne detektere, om tandbørsten har været løftet fra holderen. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan detektere, om tandbørsten har været løftet ud af holderen, betragtes kravet som passed. | Passed/failed/tested |

| | | |
|-----------------------------------|---|----------------------|
| ID:2.1 | Krav: Løsningen skal kunne afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidsrum. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen kan afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidspunkt, anses kravet for at være passed. | Passed/failed/tested |

| | | |
|-------------------------------------|--|----------------------|
| ID:3.1 | Krav: Løsningen er kompatibel med de mest almindelige forekommende tandbørster, elektriske som ikke elektriske. | Prioritet: 2 |
| Kategori: Brugervenlighed | Accepttest: Kan krav 1 opfyldes med forskelligt udformede tandbørster, anses kravet værende opfyldt. | Passed/failed/tested |

| | | |
|-----------------------------|---|----------------------|
| ID:4.1 | Krav: Løsningen skal have en indikator, der viser om tandbørstningen har været i gang i de anbefalede 2 min, samt en indikation for hvor tæt på de 2 min man er. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Kan løsningen vise når der er gået 2 min, og indikere hvor tæt på 2 min man er kommet, anses kravet for værende bestået. | Passed/failed/tested |

| | | |
|-----------------------------|---|----------------------|
| ID:5.1 | Krav: Tidtagning for 2 min anbefalet børstning, starter når tandbørsten løftes fra løsningen. | Prioritet: 2 |
| Kategori: Sensing | Accepttest: Når tandbørsten løftes fra løsningen og tidsendikationen starter, anses kravet for værende bestået. | Passed/failed/tested |

| | | |
|-----------------------------|---|----------------------|
| ID:6.1 | Krav: Løsningen skal kunne måle temperaturen for at bestemme, om tandbørsten har været holdt i hånden. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan måle temperatur for at bestemme, om tandbørsten har været holdt i hånden, anses kravet for værende bestået. | Passed/failed/tested |

| | | |
|-----------------------------------|--|----------------------|
| ID:7.1 | Krav: Kunne sende informationer om brug til serveren ved børstning. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Når serveren modtager informationerne fra løsningen omkring tandbørstning, anses kravet som værende bestået. | Passed/failed/tested |

| | | |
|-----------------------------------|---|----------------------|
| ID:8.1 | Krav: Hvis løsningen ikke kan nå serveren ved hjælp af det ønskede netværk, skal den kunne overgå til et alternativt netværk. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen ikke taber data ved at miste forbindelse til wifi, men i stedet overgår til et alternativt netværk anses kravet for at være opfyldt | Passed/failed/tested |

6.2 Kravspecifikationer - Netværk

| | | |
|-------------------------------|---|----------------------|
| ID:1.2 | Krav: Der skal gøres tre sikkerhedstiltag for netværk/enhed | Prioritet: 1 |
| Kategori: Hardening | Accepttest: Såfremt der er gjort tre tiltag for at sikre netværk/kommunikation mellem netværk og enheden, anses kravet for at være opfyldt. | Passed/failed/tested |

| | | |
|------------------------|--|----------------------|
| ID:2.2 | Krav: Klienten/klinikken skal kunne tilgå og se data fra databasen ved hjælp af et website. | Prioritet: 1 |
| Kategori: UI | Accepttest: Såfremt klienten/klinikken kan tilgå et website, der fremviser data fra databasen, anses kravet for at være bestået. | Passed/failed/tested |

| | | |
|-----------------------------------|--|----------------------|
| ID:3.2 | Krav: Den samlede løsning skal have to netværk, og der skal kunne kommunikeres via Wifi og ethernet mellem disse netværk. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt der skabes to netværk, som kan kommunikere via Wifi og ethernet, anses kravet som værende bestået. | Passed/failed/tested |

| | | |
|-----------------------------------|---|----------------------|
| ID:4.2 | Krav: Det skal være muligt at starte en aktuator fra server til enheden. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt der kan startes en aktuator fra server til enheden, anses kravet for at være opfyldt. | Passed/failed/tested |

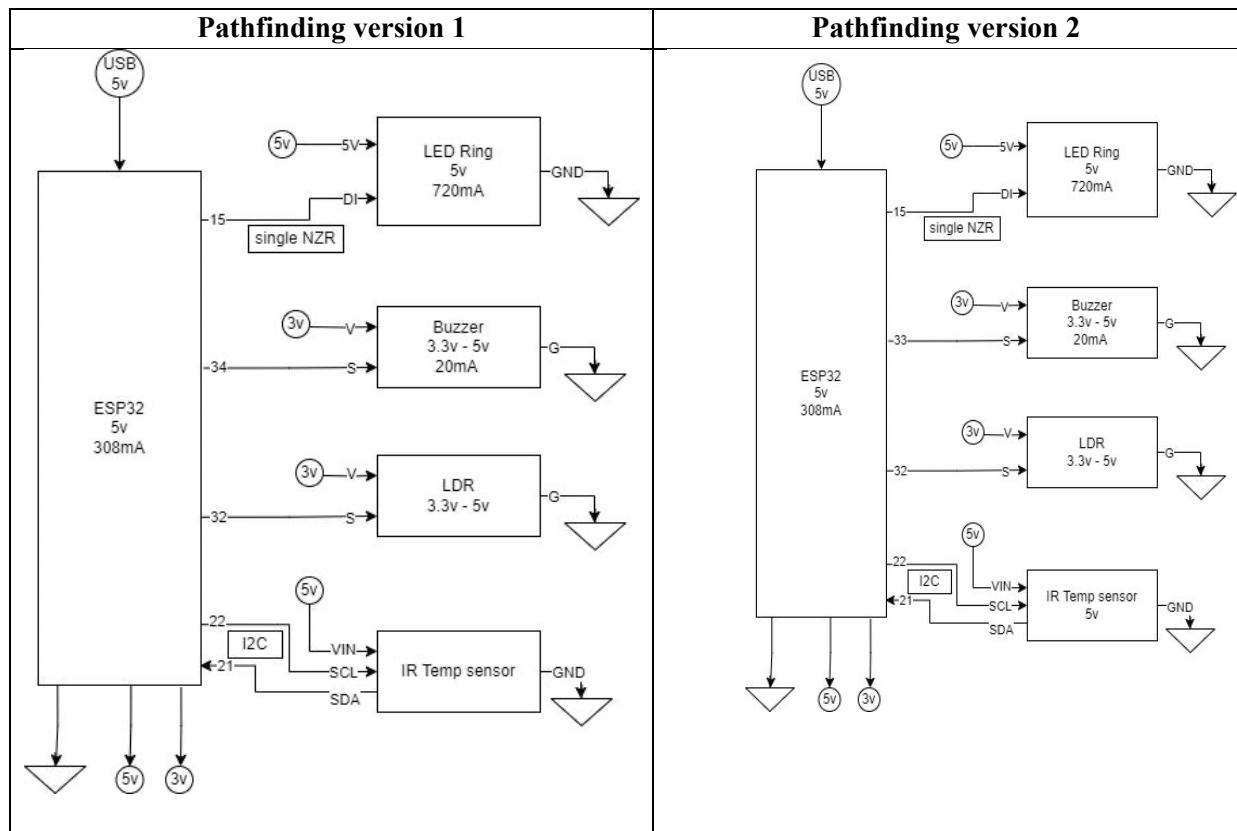
6.3 Rationale for prioritering af krav

De ovenstående krav er udarbejdet på baggrund af til dels de opstillede krav fra underviserne, men også ud fra hvad der er realistisk at kunne gennemføre inden for tidsrammen. Projektet er omfattende, og der er mange elementer, der skal falde på plads inden for en kort tidshorisont. Kravene er prioriteret efter, hvad løsningen essentielt skal kunne for at være komplet, samt hvad kravene fra undervisernes side var til den funktionelle løsning og netværk. Såfremt projektet havde løbet over en længere periode, ville det være muligt at skabe flere funktioner eller forbedre de funktioner, løsningen har.

7. Analyse

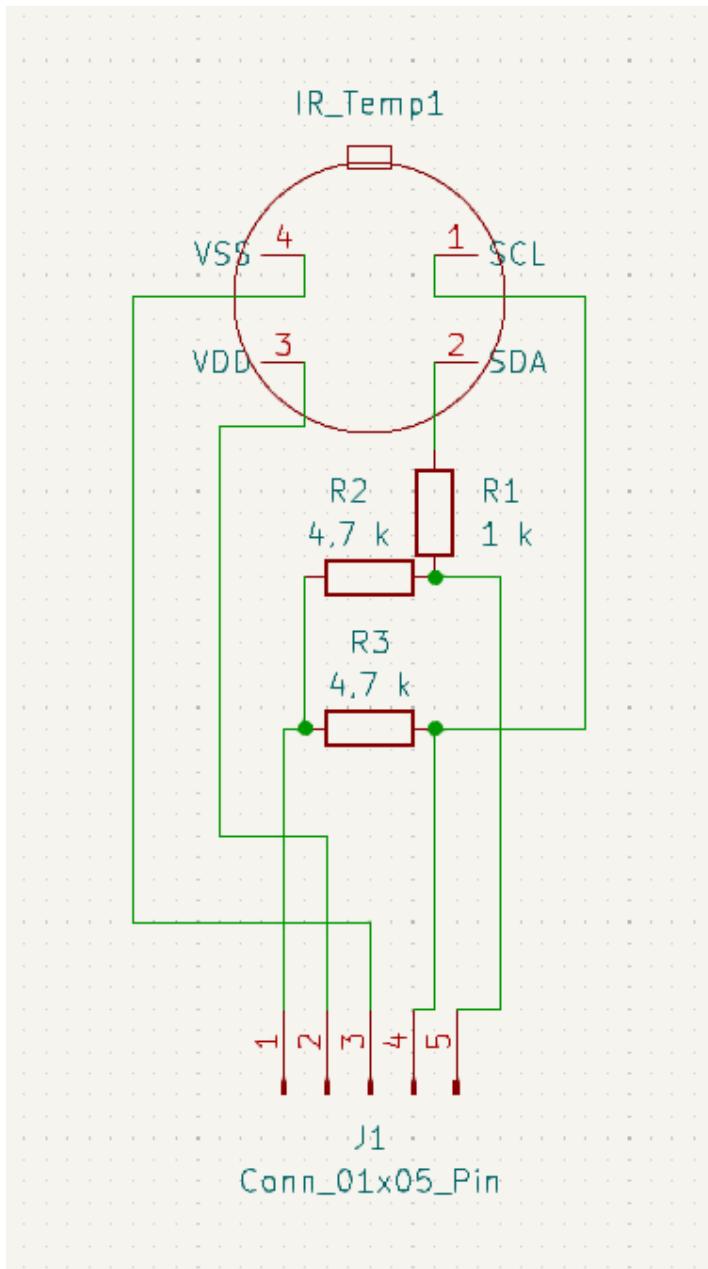
7.1 Arkitektur af pathfinding

7.1.1 Blokdiagram



Figur 7.1, første udkast af blokdiagram

7.1.2 Diagram for IR-temp-sensor

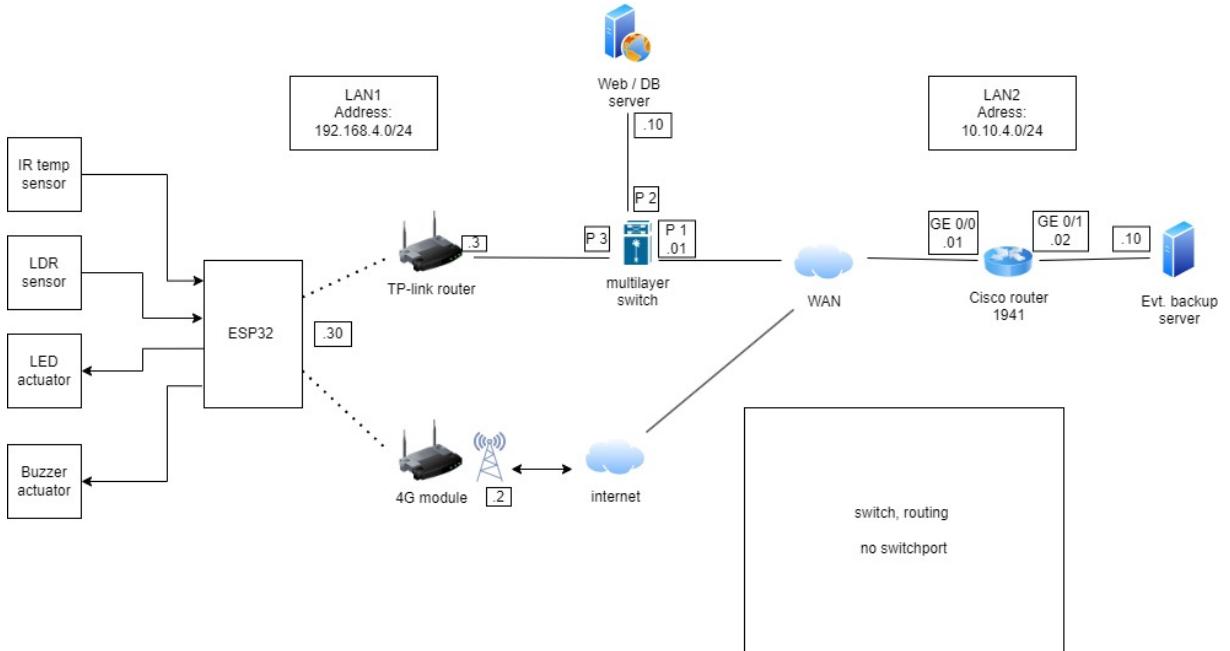


Figur 7.2, diagram for IR-temp-sensor

7.1.3 Flowchart for programmering

Da flowchartet for programmet er meget komplet, har det ikke været muligt at indsætte det i rapporten. Det kan derfor i stedet findes på [/Projekt Karibak_Afleverings_mappe/Dokumentation_og_Bilag/Media](#)

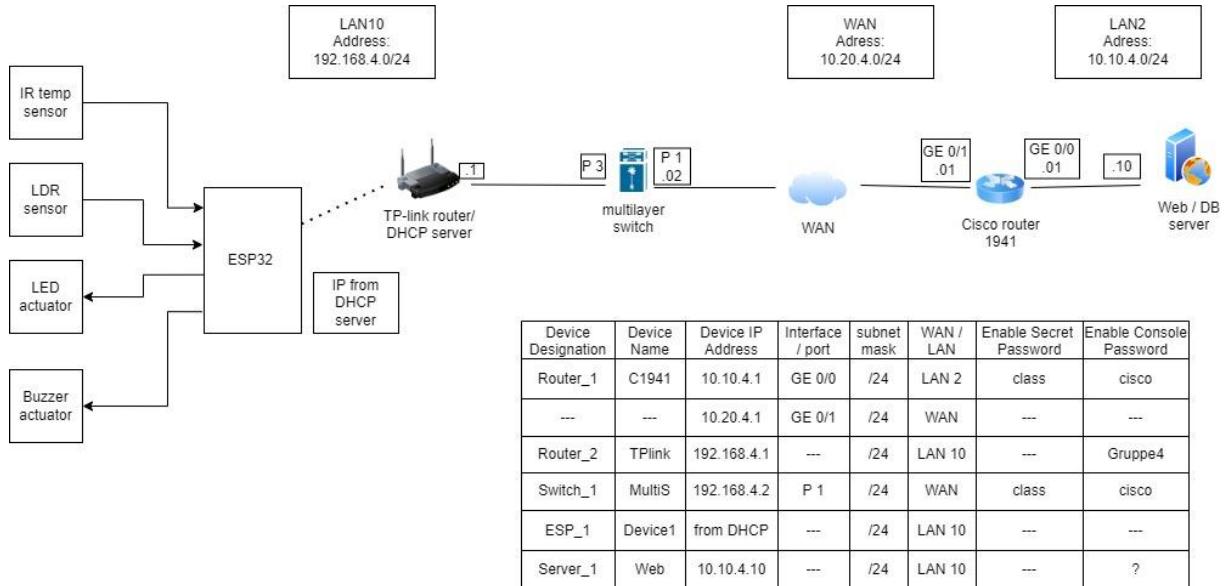
7.1.4 Første version af netværksopsætning



| Device Designation | Device Name | Device IP Address | Interface /port | Subnet mask | LAN | Enable Secret Password | Enable Console Password |
|--------------------|-------------|-------------------|-----------------|-------------|-------|------------------------|-------------------------|
| Router_1 | C1941 | 10.10.4.1 | GE 0/0 | /24 | LAN 2 | class | cisco |
| --- | --- | 10.10.4.2 | GE 0/1 | /24 | --- | --- | --- |
| Router_2 | 4G | 192.168.4.2 | --- | /24 | LAN 1 | N/A | |
| Router_3 | TPlink | 192.168.4.3 | --- | /24 | LAN 1 | N/A | |
| Switch_1 | MultiS | 192.168.4.1 | P 1 | /24 | LAN 1 | | |
| ESP_1 | Device1 | 192.168.4.30 | --- | /24 | LAN 1 | N/A | N/A |
| Server_1 | WEB | 192.168.4.10 | --- | /24 | LAN 1 | N/A | |
| Server_2 | Backup | 10.10.4.10 | --- | /24 | LAN 2 | N/A | |

Figur 7.3 og 7.4, Opsætning af netværk

7.1.5 Endelige version af netværksopsætningen



Figur 7.5, endelig version af netværksopsætning

7.2 Den endelige arkitektur

7.2.1 Protokoller

7.2.1.1 HTTP

Hypertext Transfer Protocol er en protokol på applikationslaget, som gør det muligt at websites og servere kan dele information mellem hinanden. Dette foregår via HTTP request og response messages.

7.2.1.2 SMTP

Simple Mail Transfer Protocol er en protokol på applikationslaget. Protokollen er en måde at muliggøre sending af email mellem client og server. SMTP-serveren lytter altid. Når en client vil sende en mail, bliver der skabt en tcp-forbindelse til serveren, og mailen kan sendes.

7.2.1.3 I2C

I2C protokollen er en master-slave seriel kommunikationsprotokol. Med I2C kan man enten have én master, der styrer mange slaver, eller mange masters der styrer én slave. I2C har en seriel clock linje og en seriel data linje. Seriel data står for overførslen af data mellem master og slave eller slave og master. Serial clock linjen står for at styre synkronisering.

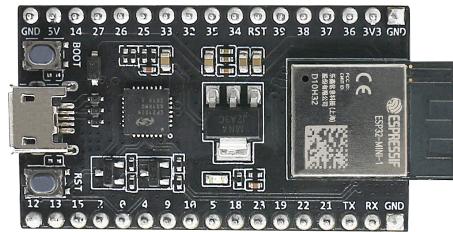
7.2.1.4 OSPF

OSPF er en routingprotokol. Det fulde navn for protokollen er Open Shortest Path First, som refererer til måden, routingen opereres på. Den benytter sig af link state routing, og opererer under et single autonomt system. OSPF samler link-state-information, som efterfølgende bruges til at mappe netværk. Herefter routes packages over netværket via den korteste vej til destination-IP-adressen. Såfremt et link lukker ned findes den korteste, alternativ rute til destinations-IP-adressen.

7.2.2 BOM-liste

ESP32

En microcontroller. Robust og kompakt, og er på grund af sit lave strømforbrug god til små systemer. Har mulighed for wifi og bluetooth. [1.1] Er i dette projekt brugt som central for softwaren i selve IoHT-systemet, som tandbørsteholderen er baseret på. Den har desuden til opgave at kommunikere ud til og gennem netværk.



Passiv buzzer

En buzzer der kan udsende forskellige lyde og toner, som sættes med software. Har en operativ spænding på 3,3-5V, og er derfor velegnet til små systemer. [1.3]

Er i dette projekt brugt som indikator for, hvis lyssensoren ikke har detekteret en ændring inden for en given tid.

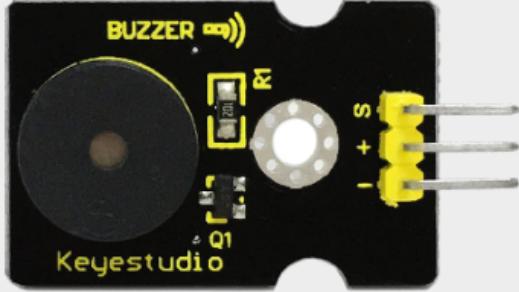


Foto: <https://mit.kea.dk/labs/alle-maskiner/passiv-buzzer>

Digital infrarød sensor

En infrarød sensor, der kan måle temperatur mellem -40 til 80 grader lokalt og -70 til 380 grader eksternt (celsius). Har udgangstyperne PWM og SMBus. Har en forsyningsspænding på 4,5-5V. [1.4] I dette projekt er sensoren brugt til at måle, hvorvidt tandbørsten har været holdt/brugt, eller om den bare har været lagt uden for holderen. Det skal indikere, om brugeren reelt har brugt tandbørsten, som løsningen er ment til at hjælpe med.

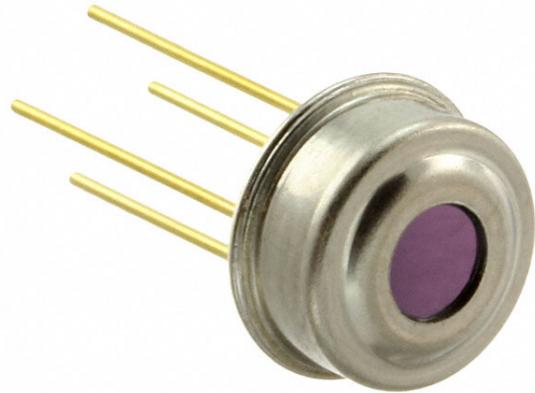


Foto:
<https://www.digikey.dk/dk/da/products/detail/melexis-technologies-nv/MLX90614ESF-AAA-000-TU/1647940>

Lysniveau-sensor:

En sensor der kan måle lysniveau med sin light-dependent resistor. Den har en operativ spænding på 3,3-5V, hvilket gør den velegnet i små systemer. [1.5]

I dette projekt bruges den til at indikere, om tandbørsten har været løftet ud af holderen.



Foto: <https://mit.kea.dk/labs/alle-maskiner/lysniveau-sensor>

7.2.3 Udvalgte programmeringsblokke

Få en læsning fra ldr sensor med funktionen `ldr_reading()`

Tag en aflæsning af ldr-sensoren. For at minimere udsving i målinger tages målingen 50 gange via et for-loop. Værdierne bliver lagt i en liste kaldet `readings`. Derefter tages gennemsnittet af de 50 læsninger. Det resultat bliver til sidst afrundet til et helt tal, og tallet returneres.

Dette pakkes ind i en funktion kaldet “`ldr_reading()`”. For at dette kan lade sig gøre, skal Pin og ADC importeres fra ‘machine’-modulet.

Gem data i en tekstfil med funktionen `write_data_to_file(data)`

Gem indholdet af argumentet i en tekstfil.

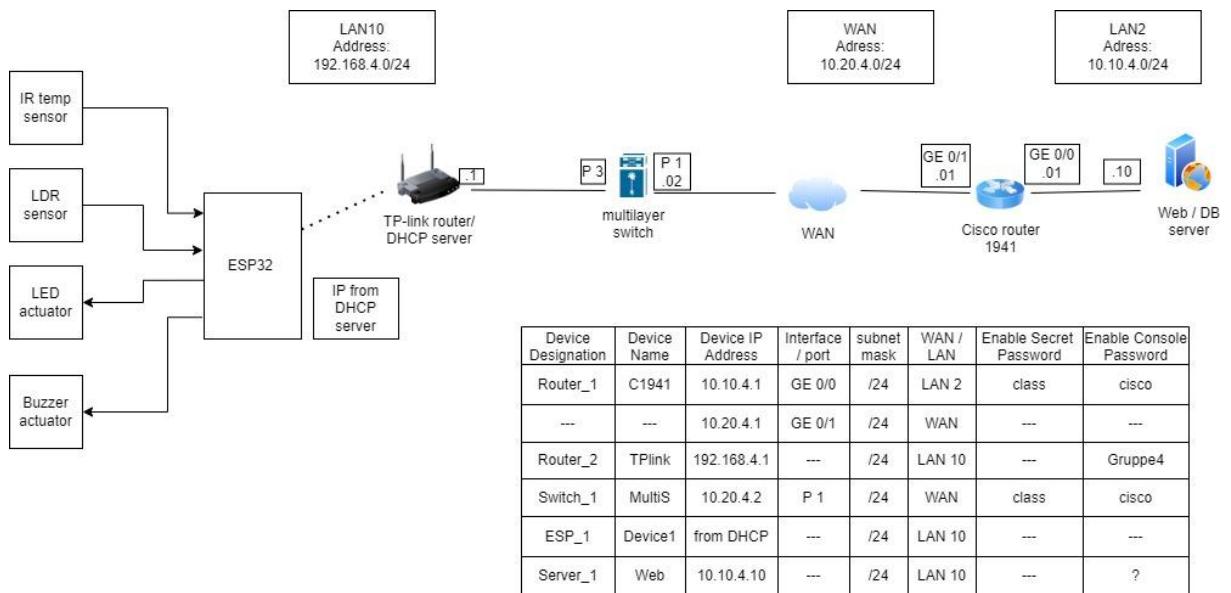
En variabel kaldet “`filename`” oprettes med indholdet “`log_of_data_not_sent.text`” som en string.

Via en context manager bliver filen åbnet, og dataen samt et timestamp bliver skrevet til en tekstfil.

8. Løsningsdesign

8.1 Router/Switch opsætning og Interface forbindelser

Her ses et skema med oversigt over opsætning af netværket.



Figur 8.1, opsætning af netværk

Netværket består af tre forskellige netværk, for at simulere at data bliver sendt over internettet. LAN 2 og 10 er tænkt som typiske perifere netværk med stjerne-topologi, hvis der havde været flere endpoints. Som man kan se i figur 8.1 har vi, for at gøre opsætning af løsningen nemmere hos klienten, valgt at vores løsning selv skal hente sin IP adresse hos en DHCP-server. I netværket er det TP-link routeren, som også har rollen som DHCP-server.

8.1.1 Konfiguration af Cisco-router 1941-series

Herunder vises de kommandoer, der er brugt til at konfigurere Cisco-routeren.

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname C1941
```

Figur 8.2, skift af hostname

```
C1941(config)#interface gigabitethernet0/0
C1941(config-if)#ip address 10.10.4.1 255.255.255.0
C1941(config-if)#no shutdown
```

Figur 8.3, konfigurerering af IP-adresse på interface0/0

```
C1941(config)#interface gigabitethernet0/1
C1941(config-if)#ip address 10.20.4.1 255.255.255.0
C1941(config-if)#no shutdown
```

Figur 8.4, konfigurering af IP-adresse på interface0/1

```
C1941(config)#router ospf 1
C1941(config-router)#network 10.10.4.0 0.0.0.255 area 1
C1941(config-router)#network 10.20.4.0 0.0.0.255 area 1
C1941(config-router)#network 192.168.4.0 0.0.0.255 area 1
C1941(config-router)#end
```

Figur 8.5, opsætning af OSPF-routing

```
C1941#wr
Building configuration...
[OK]
```

Figur 8.6, konfigurationsfilen gemmes

8.1.2 Konfiguration af switch

Herunder ses de kommandoer, der er blevet brugt til at konfigurere multilayer switchen.

```
Switch>enable
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname MultIS
```

Figur 8.7, Konfiguration af navnet på switchen.

```
MultIS#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
MultIS(config)#interface gigabitethernet0/1
MultIS(config-if)#no switchport
```

Figur 8.8, Konfiguration af interface 0/1, til at være routing port.

```
MultIS(config-if)#ip add 10.20.4.1 255.255.255.0
MultIS(config-if)#no shutdown
```

Figur 8.9, Tildeling af IP adresse til interface 0/1, routing porten.

```

MultiS#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
MultiS(config)#ip routing
MultiS(config)#router ospf 1

network 10.10.4.0 0.0.0.255 area 1
network 10.20.4.0 0.0.0.255 area 1
network 192.168.4.0 0.0.0.255 area 1

```

Figur 8.10, Konfiguration af OSPF som routing protokol

```

MultiS(config)#vlan 10
MultiS(config-vlan)#name VLAN10
MultiS(config-vlan)#end
MultiS(config)#int g0/2
MultiS(config-if)#switchport mode access
MultiS(config-if)#switchport access vlan 10
MultiS(config)#int g0/3
MultiS(config-if)#switchport mode access
MultiS(config-if)#switchport access vlan 10
MultiS(config)#int g0/4
MultiS(config-if)#switchport mode access
^
% Invalid input detected at '^' marker.

MultiS(config-if)#switchport mode access
MultiS(config-if)#

```

Figur 8.11, Opsætning af VLAN og tildeling af interfaces.

```

MultiS(config)#int vlan 10
MultiS(config-if)#ip add 192.168.4.1 255.255.255.0
MultiS(config-if)#end

```

Figur 8.12, Tildeling af IP adresse til VLAN

```

MultiS#wr
Building configuration...
[OK]

```

Figur 8.13, Gemme konfigurationen til startup-config filen

8.2 Kodebeskrivelse

Beskrivelserne nedenfor er dele af koden, vi følte var vigtige at påpege. Grunden kan være vigtighed, funktionalitet eller særlige områder, vi havde inddraget.

8.2.1: Logging

Når man opererer en server på den størrelse, vi potentielt ønsker, er det en rar funktion at kunne gå tilbage og gennemse, hvad præcist serveren har arbejdet med. I mindre programmer, som aktivt bliver observeret dagligt, er “**print()**”-funktionen nok. Men da vi skal tage højde for, at der kan potentielt kan gå dagevis uden, at der bliver kigget på serverens konsol, og med det potentielle scenarie at serveren var blevet lukket og startet op igen i det tidsrum, har man brug for at kunne kigge på informationen på en nem og overskuelig måde, der ikke slettes.

Vores løsning blev at bruge logging-modulet, der er indbygget i Python. Med dette modul har vi muligheden for at udpege den information vi vil have, og kategorisere det i filer, hvor vi selv bestemmer hvordan det bliver vist.

Visionen er at have specifik information omkring næsten alle funktioner i koden, når de køres, give dem et tidsstempel, og lægge det i filer. Vi har valgt at have fire kategorier: DEBUG, INFO, ERROR og WARNING.

Det meste af informationen placeres under INFO, mens ting der kan påvirke systemets drift bliver pålagt ERROR eller WARNING, afhængig af problematikken. DEBUG får både INFO-, ERROR- og WARNING-beskeder sammenlagt i et dokument.

Herunder vises funktionen, der opsætter logging.

```

def logging_init():
    #Below are the 4 log types we use. DEBUG, INFO, ERROR and WARNING
    logger = logging.getLogger()
    logger.setLevel(logging.DEBUG)

    debug_handler = logging.FileHandler("logs/debug_log.log")
    debug_handler.setLevel(logging.DEBUG)

    info_handler=logging.FileHandler("logs/info_log.log")
    info_handler.setLevel(logging.INFO)

    error_handler=logging.FileHandler("logs/error_log.log")
    error_handler.setLevel(logging.ERROR)

    warning_handler=logging.FileHandler("logs/warning_log.log")
    warning_handler.setLevel(logging.WARNING)

    formatter = logging.Formatter("%(asctime)s - %(name)s - %(levelname)s - %(message)s")
    debug_handler.setFormatter(formatter)
    info_handler.setFormatter(formatter)
    error_handler.setFormatter(formatter)
    warning_handler.setFormatter(formatter)

    logger.addHandler(debug_handler)
    logger.addHandler(info_handler)
    logger.addHandler(error_handler)
    logger.addHandler(warning_handler)
  
```

Figur 8.14, funktionen ‘logging_init()’

Dernæst laves en funktion, som refereres til i resten af koden til at nedskrive hvad der sker.

Funktionen “***log_func(e, msg, type)***” har tre argumenter. “***e***” er en string der tilhører ‘exception’ teksten, hvis der er det. Hvis ikke, er feltet tomt. “***msg***” er beskeden der manuelt bliver indtastet som en indikator for hvilken del af koden, der logges fra, og hvad der overfladisk sker. Til sidst er “***type***”, som ganske enkelt en string, der fortæller hvilken fil logging-informationen skal skrives i.

```
def log_func(e, msg, type):
    logger = logging.getLogger()
    print(msg)
    try:
        if type == "debug":
            logger.debug(msg)
        elif type == "info":
            logger.info(msg)
        elif type == "error":
            logger.warning(msg)
        elif type == "warning":
            logger.error(msg)
        if e == "":
            return
        else:
            print("printing error")
            logger.error(e)
            print(e)
    except Exception as e:
        logger.error("Logging failed")
        print(e)
```

Figur 8.15, funktionen 'log_func()'

Når noget ønskes logget, referere man til funktionen i scriptet, og udfylder argumenterne som vist nedenfor:

```
except Exception as e:
    print(log_manager.log_func(e,f"Could not hash password","error"))
```

Figur 8.16, udfyldning af arguments

Vi tilføjer også et print-statement, for også at få det med i konsollen. Dette sker for både at have det i filer, men også for at nemmere kunne se hvad der sker direkte.

På figur 4 ses en andel af DEBUG.log filen, og hvad den har indsamlet. Med omhyggelig logging, kan man nemmere få et overblik over potentielle fejl, og hvordan det påvirker serveren.

```

2906 2023-12-14 11:39:21,666 - root - INFO - DB initialized
2907 2023-12-14 11:39:21,681 - root - INFO - Deleted all OTPs as a startup procedure
2908 2023-12-14 11:39:21,685 - root - INFO - New table created in /Users/oliverboots/Documents/GitHub/project_tbd_gruppe4/DEV/karibak_0.1/karibak_web/Test.db
2909 2023-12-14 11:39:36,079 - root - INFO - Regex accepted: jjefjslkfjsldkjf
2910 2023-12-14 11:39:36,080 - root - INFO - Regex accepted: kpheotnpnrwemf
2911 2023-12-14 11:39:36,080 - root - WARNING - Could not check for admin
2912 2023-12-14 11:39:36,082 - root - ERROR - 'NoneType' object is not subscriptable
2913 2023-12-14 11:39:45,660 - root - INFO - Regex accepted: hashed_oliver
2914 2023-12-14 11:39:45,661 - root - INFO - Regex accepted: 999
2915 2023-12-14 11:39:46,004 - root - INFO - Admin user hashed_oliver log in requested
2916 2023-12-14 11:39:46,005 - root - INFO - Checking key in database...
2917 2023-12-14 11:39:46,005 - root - INFO - Could not check otp
2918 2023-12-14 11:39:48,449 - root - INFO - Sending mail to user
2919 2023-12-14 11:39:48,454 - root - INFO - New key created in /Users/oliverboots/Documents/GitHub/project_tbd_gruppe4/DEV/karibak_0.1/karibak_web/Test.db
2920 2023-12-14 11:40:00,279 - root - INFO - Regex accepted: 677
2921 2023-12-14 11:40:00,280 - root - INFO - Checking key in database...
2922 2023-12-14 11:40:00,280 - root - INFO - Checking key with username...
2923 2023-12-14 11:40:00,280 - root - INFO - Checking key in database...
2924 2023-12-14 11:40:00,282 - root - INFO - Deleted the key 677
2925 2023-12-14 11:40:00,283 - root - INFO - Correct key. User logged in
2926 2023-12-14 11:40:30,547 - root - INFO - updated name of id: c1ac11a9-dc44-46c9-a990-ee2ac711e9c6
2927 2023-12-14 11:40:30,549 - root - INFO - changed user info for c1ac11a9-dc44-46c9-a990-ee2ac711e9c6
2928 2023-12-14 11:40:42,842 - root - INFO - Values sent:[True]

```

Figur 8.17, logs

8.2.2: Cookies

Da serveren har adgang til sensitiv information og mulighed for at påvirke dagligdagen for dens klienter, skal der være implementeret protokoller, der forhindre uautoriserede i at få adgang. Alt hvad vi bruger er beskrevet i vores sikkerhedstiltag, men for at undgå frustrationen fra klientens side at skulle logge ind med en lang procedure hver gang der går ind på siden, eller ved en fejl skifter til en anden hjemmeside, gemmer vi deres bekræftelse på login i en cookie i deres web-browser.

Dette gøres kun EFTER de er logget ind, og bliver slettet når web-browseren lukkes helt ned. Der er også mulighed for at sætte en tid på denne cookie, hvis der er efterspørgsel for det.

```
50 | session_cookie = request.get_cookie('karibak_login')
```

Figur 8.18, session-cookie

Når man kommer ind på hvilken som helst side på serveren, bortset fra “**/login**”, bliver man tjekket for denne cookie og hvis det stemmer, kommer man ind, mens hvis “bounceren” ser et forkert ID, smider de dem helt tilbage til login-skærmen. Der gøres ingen undtagelser for dette.

```

71 @route('/verify/<username>', method=['POST', "GET"])
72 def verify(username):
73     if request.method=="POST":
74         number=request.forms["number"]
75         #print(f'This is the username otp thingy... {username}')
76         if login_manager.check_for_regex(number) == True:
77             if db_manager.otp_check(number, username)==True:
78                 db_manager.delete_otp(number)
79                 response.set_cookie('karibak_login', 'karibak_id', path='/')

```

Figur 8.19, verificering af bruger

```

91 def home():
92     session=request.get_cookie('karibak_login')
93     print(f"Session:{session}")
94
95     if session=='karibak_id':
96         holders=db_manager.init_holders()
97         print(holders[0][5])
98         if request.method=="POST":

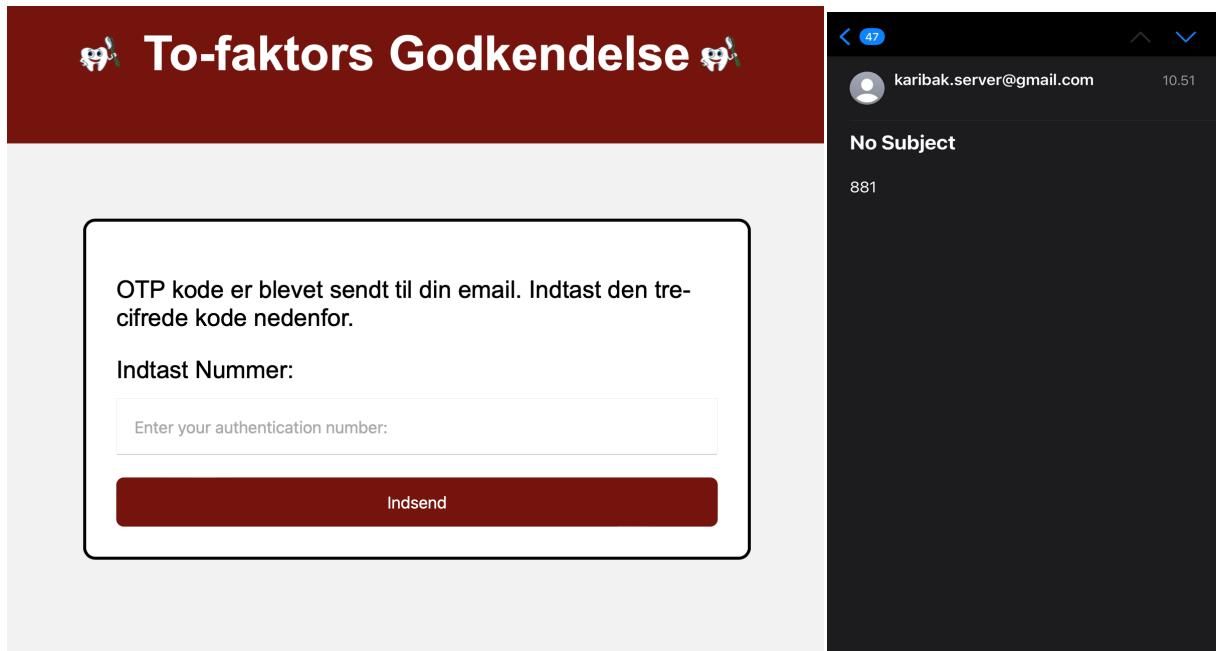
```

Figur 8.20, funktionen home()

8.2.3: SMTP

Det er ikke kun et login med brugernavn og adgangskode man skal igennem for at komme ind på siden. Hver bruger i databasen har også en email, den pågældende bruger er associeret med. Når brugeren logger ind med deres login-oplysninger, bliver de sendt en trecifret kode til deres mail, som de derefter skal indtaste inden for et vist tidsrum, for at de får deres login-cookie og bliver sendt til dashboardet. Dette kaldes for Multi-factor-authentication eller MFA.

Nedenfor ses siden med godkendelsen og emailen sendt til kontoen.



Figur 8.21 og 8.22, to-faktor-godkendelse

For at få dette til at virke, har vi importeret modulet “**smtplib**”, der indeholder det meste af det vigtige funktioner til at få dette op og køre. Med modulet kan vi indstille hvilken email-server, vi ønsker at forbinde til. Dette er en besværlig, men ikke lang proces. Det indebærer at give andre programmer tilladelse til at kontrollere emails til og fra adressen. Vi brugte Gmail til projektet, da det var det umiddelbart nemmest at opsætte.

I koden skulle vi indstille forbindelsen med disse variabler:

“**SMTP_SERVER**” er selve platformens adresse og “**SMTP_PORT**” er porten vi forbinder til. “**SMTP_USERNAME**” er email addressen beskeden bliver sendt fra og “**SMTP_PASSWORD**” er den specifikke nøgle vi er blevet tildelt af platformen til dette formål. Dette er ikke adgangskoden til emailen selv.

```
SMTP_SERVER = 'smtp.gmail.com'
SMTP_PORT=587
SMTP_USERNAME="karibak.server@gmail.com"
SMTP_PASSWORD="uzjx ilgq jpbb bpse"
```

Figur 8.23, SMPT

Efter dette køres en funktion, der samler beskeden og sender den til personens email.

Beskeden kunne være næsten alt, men vi sender en enkelt besked der består af det autogenereret nummer fra “*rand_otp()*” funktionen.

```
def send_mail(recipient, username):
    try:
        msg=str(rand_otp(username))
        #print(f"The message:{msg}")
        with smtplib.SMTP(SMTP_SERVER,SMTP_PORT) as server:
            server.starttls()

            server.login(SMTP_USERNAME, SMTP_PASSWORD)

            server.sendmail(SMTP_USERNAME,recipient,msg)

            print(log_manager.log_func("","Sending mail to user","info"))
            db_manager.save_otp(msg, username)
    except Exception as e:
        print(log_manager.log_func(e,"Could not send email","error"))
```

Figur 8.24, funktionen *send_mail()*

Koden der bliver sendt kan kun bruges til den bruger, og bliver ubrugelig efter 100 sekunder.

8.2.4 Hashing

For at sikre serverens integritet, har det været nødvendigt at kigge på en måde at skjule informationer, der ikke må ses af uautoriserede. Der er mange løsninger til dette problem. Vi har valgt at bruge modulet “*bcrypt*” til at lave funktioner, der sløre og gemmer dataen i vores database, så kun gennem programmet selv, kan man se de vigtige informationer.

Vi ville forklare omkring to forskellige brug af disse. Den ene er hashing og den anden er byte-kryptering.

Nedenunder ses de funktioner, vi har brugt til at sløre adgangskoden som administrationen bruger til at logge ind på dashboardet med.

```
#This function hashes a password. For use in database login.
def hash_password(password):
    try:
        print(log_manager.log_func("",f"{password} hashed...","info"))
        return bcrypt.generate_password_hash(password).decode('utf-8')

    except Exception as e:
        print(log_manager.log_func(e,f"Could not hash password","error"))

#This function decrypts a hashed password with the input password from the website.
def decrypt_password(hashed_password, password):
    try:
        return bcrypt.check_password_hash(hashed_password, password)
    except Exception as e:
        print(log_manager.log_func(e,f"Could not decrypt password","error"))
```

Figur 8.25, sløring af adgangskode

“**Bcrypt**” håndterer alle salt- og hashing-algoritmer der er nødvendige, mens vi giver dem den tekst der skal hashes eller un-hashes.

“**Bcrypt**”-funktionen giver os tilbage en lang string, som så sættes ind i databasen. Når en bruger forsøger at logge ind med adgangskoden, sammenligner “**decrypt_password**”-funktionen det indtastede adgangskode med det der ligger i databasen, un hasher og tjekker om de er det samme.

Så en string med få symboler kan blive hashet til:

\$2b\$12\$toQZAgncgEevtxyn56t3gue668zCYoq0YxMGQz6Xd5.4JYbcBb572

Den anden form for kryptering laves om på selve variablen, og det forklares i sektion 8.2.5.

8.2.5: Encryption / Decryption

Databasen indeholder vigtige informationer omkring alt, der har noget at gøre med klientens brugere. Nogle af disse informationer er sensitive, såsom navn, adresse, IP-adresse og andet. Det ville være en katastrofe hvis disse informationer var frit tilgængelig i plain tekst i databasen. Derfor må vi gemme denne tekst, så de kun kan ses i deres oprindelige form, hvis det bliver kaldt efter.

Vi ville kun have teksten til at blive vist på dashboardet. Alle andre tidspunkter skulle det gerne være umuligt at tilgå.

Her bruger vi et andet modul: Fernet.

Fernet fungerer ved at ændre på filtypen af et variable frem og tilbage hvis man har den rigtige krypteringsnøgle.

Nøglen har vi genereret på forhånd og har lagt det i plain tekst i mappen. For at teste med, var denne tilgang passende, men hvis programmet skulle bruges i et rigtigt scenarie, er det vigtigt at gemme denne nøgle et sikkert sted. Filen kunne ligge hvor som helst, og være krypteret selv, som vi vidste da vi lavede funktionen, men vi vidste ikke hvor lang denne krypterings-kæde kunne blive. Så vi gjorde det simpelt for dette projekt, men dette skal tænkes godt over, hvis dette skal bruges i et rigtigt scenarie.

Som det ses nedenfor, læser funktionen nøglen fra filen, tager teksten og laver det om til et byte format i en UTF-8 encodning. ***“Decrypt_text()”*** gør det modsatte med samme nøgle.

```
#Encrypts a string with Fernet. Used to protect important database data.
def encrypt_text(text):
    try:
        with open ('db_key.key', 'rb') as file:
            key=file.read()
            encrypt_key=Fernet(key)
            b = bytes(text, 'utf-8')
        return encrypt_key.encrypt(b)
    except Exception as e:
        print(log_manager.log_func(e,"Could not encrypt text","error"))

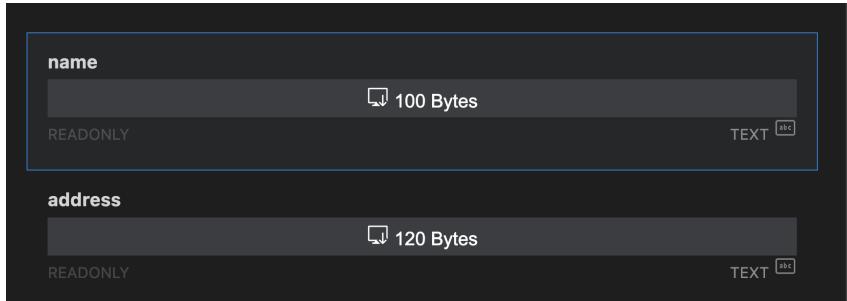
#Used to decrypt a variable in the database.
def decrypt_text(text):
    try:
        with open ('db_key.key', 'rb') as file:
            key=file.read()
            encrypt_key=Fernet(key)
            decrypted_string= encrypt_key.decrypt(text).decode('utf-8')
        return str(decrypted_string)
    except Exception as e:
        print(log_manager.log_func(e,"Could not decrypt text","error"))
```

Figur 8.26, encryption og decryption af tekst

Her ses et eksempel på hvordan det ser ud i databasen, efter variabel er lavet om.

Bytet for navnet er således:

gAAAAABleu555UZU7ViflHbD8zfwjWpIKmCCvJ3NrsI14SnXcyShYxAAZsGSfHXX_EaCVNTbVVi4EOlrBmlXn4bg7kjiGrreCA==



Figur 8.27, slørret data

8.2.6 Main.py på esp32

Her beskrives indholdet af main.py filen som ligger på ESP'en.

For at undgå at bruge sleep-funktionen for meget i koden, bruges funktionen “*time()*” fra time modulet en del.

```

1 import time
2 import ir_test_works
3 import network_test_1
4 #import buzzer
5 import _thread
6 import ldr_reading
7 import led_ring
8 import socket_buzz
9
10 # socket thread
11 _thread.start_new_thread(socket_buzz.init_socket, ())
12
13 # network thread
14 _thread.start_new_thread(network_test_1.net_work_test, ())
15
16 # IR sensor
17 MLX90614_TOBJ1 = 0x07 # Object temperature address
18 MLX90614_TA = 0x06 # Ambient temperature address
19
20 # count
21 count=0
22
23 # taking a base light sample
24 base_light = ldr_reading.ldr_reading()
25 print(f"base light : {base_light}")
26 base_light_timer_start = time.time()
27
28 # timer start
29 start_time = time.time()
30
31 # variable for toothbrushing start
32 brush_time_start = 0
33
34 # completed brushes
35 brushes_completed = 0
36 brushes_completed_reset_timer = time.time()
37
38 # list of data to be sent
39 list_of_data_to_send = []
40
41 while True:
42     # get sensor reading
43     light_value = ldr_reading.ldr_reading()
44     #print('light value: ', light_value)
45     time.sleep(0.1)
46
47     # base light sample reset
48     base_light_timer_reset = time.time()
49     if base_light_timer_reset - base_light_timer_start > 600:
50         base_light = ldr_reading.ldr_reading()
51         print(f"new base light: {base_light}")
52         base_light_timer_start = time.time()
53

```

Figur 8.28, main.py-filen

I toppen af filen bliver der importeret forskellige moduler.

På linje 11 og 14 startes to threads, socket threaden på linje 11 som sørger for esp'en lytter på en port og kan buzzze hvis den får en besked fra serven.

På linje 13 startes funktionen “***net_work_test()***” som en tread, denne funktion sørger for esp'en forbinder til et wifi, og hvis wifi forbindelsen bliver afbrudt, forsøger den at forbinde igen. På linje 17 og 18 bliver der oprettet to variabler, der indeholder I2C adresser til den infrarøde temperatursensor.

Der tages en lysmåling på linje 24, denne bruges som en baseline for lysændringer.

På linje 41 starter et while-True loop som kører konstant, i toppen af det loop bliver der oprettet en variabel der hedder “***light_value***” som tager en lysmåling med et sleep på 0.1 sekunder.

Fra linje 49 til 52 er en condition som opdaterer baseline lysværdien hver gang der er gået 10 minutter.

```

54     # variable to reset LED ring color from green to red after some time
55     brush_timer = time.time()
56     if brush_timer - brushes_completed_reset_timer > 43200: # 12 hour reset timer
57         brushes_completed = 0
58         brushes_completed_reset_timer = time.time()
59         print('ring turns red - brush_completed reset to 0')
60         # led ring turns red
61         led_ring.led_red()
62
63     # if toothbrush is removed from stand
64     #print('light_value', light_value)
65     if light_value>base_light*1.2: # ændret fra 1.3
66
67         # tooth brush timer
68         current_time = time.time()
69         # check if time brushed timer has started or not
70         if brush_time_start == 0:
71             brush_time_start = time.time()
72             print('brushing started..')
73             # turn led ring off
74             led_ring.led_off()
75
76
77         #print('light value after lift: ',light_value,' ',count)
78         if current_time - start_time > 10 and count < 12:
79             start_time = time.time()
80             count=count+1
81             print(f'count: {count}')
82
83         # led ring lights up as time passes
84         led_ring.led_brush(count)
85
86         # if the 12 seconds have passed blink blue
87         if count == 12:
88             led_ring.led_blink()
89
90
91         # when toothbrush is placed back
92         if light_value < base_light*1.2 and brush_time_start > 0 or light_value == 0 and brush_time_start > 0:
93
94             brush_time_finish = time.time()
95             total_time_brushed = (brush_time_finish - brush_time_start)
96             #print('light value after placed back: ',light_value, ' base_light after placed back: ', base_light)
97             #print('toothbrush placed back, brush time start: ', brush_time_start,' brush time finish:',brush_time_finish,
98             print('brushing finished..')
99             print(f'total time brushed: {total_time_brushed} seconds')
100
101
102         # take new base light reading
103         time.sleep(0.5)
104         base_light = ldr_reading.ldr_reading()
105         print(f'new base light : {base_light}')
106

```

Figur 8.29, en del af main.py filen

Fra linje 56 til 61 ses en condition, der sørger for at led-ringens skifter farve til rød hver gang der er gået 12 timer.

På linje 65 er en condition som tjekker om den konstant målte “*light_value*” er højere end den målte baselight værdi. Hvis “*light_value*” er højere er tandbørsten blevet løftet fra glasset.

Tiden gemmes i en variabel kaldet “*current_time*” på linje 68. Denne variabel opdateres konstant, så længe tandbørsten er væk fra glasset.

Fra linje 70 til linje 74 er en condition der tjekker om tandbørstningen allerede er igang eller den skal til at begynde. Hvis “*brush_time_start*” ikke er nul, er børstningen allerede i gang. Hvis den er nul skal tandbørstningen til at begynde, og led-ringens slukkes for at være klar til optælling.

På linje 78 er der en condition, der tæller en variabel ved navn “*count*” en op, hver gang 10 sekunder er passeret, så længe at *count* er mindre end 12. Funktionen “*led_brush(count)*” på

linje 84 sørger for at en ekstra led i led-ringen bliver tændt hver gang “count” tæller en op, og der altså er gået 10 sekunder.

Når “**count**”-variablen når 12, er 120 sekunder passeret, og led-ringen blinker blåt. Dette ses på linje 87 og 88.

For at vide hvornår tandbørsten bliver placeret i glasset igen, er der lavet en condition som ses på linje 92. Denne condition tjekker om “**light_value**” er mindre end “**base_light**” + 20% og om “**brush_time_start**” er mere end nul, eller om “**light_value**” er nul og “**brush_time_start**” er mere end nul. Hvis dette er sandt er tandbørsten højest sandsynligt placeret tilbage i glasset, og tandbørstningen er afsluttet.

På linje 95 beregnet tiden, at tandbørsten har været væk fra glasset. Der taget en ny “**base_light**” måling efter tandbørsten er tilbage i glasset, dette ses på linje 104.

```

107     # turn off led ring again
108     led_ring.led_off()
109
110
111     # check if total time brushed was long enough. if yes, turn green, if no turn red
112     if total_time_brushed > 110 or brushes_completed == 1:
113         brushes_completed = 1
114         led_ring.led_green()
115
116     if total_time_brushed > 110:
117
118         # temp check
119         object_temp = ir_test_works.read_temperature(MLX90614_TOBJ1)
120         ambient_temp = ir_test_works.read_temperature(MLX90614_TA)
121         print('object temp:',object_temp)
122         print('ambient temp:',ambient_temp)
123         #tooth_brush_temp = object_temp-ambient_temp
124         if object_temp > ambient_temp:
125             tooth_brush_temp = object_temp-ambient_temp
126             print(f'toothbrush has been touched, its {tooth_brush_temp} *C hotter')
127
128         # try to send data to server
129         list_of_data_to_send.append(total_time_brushed)
130         list_of_data_to_send.append(f'tandbørste berørt og {tooth_brush_temp}*C varmere')
131
132         #print('list of data to send',list_of_data_to_send)
133         if total_time_brushed > 110:
134             print(f'simuleret send: {list_of_data_to_send[-1]}, {list_of_data_to_send[-2]}')
135             network_test_1.init_data(list_of_data_to_send[-2],list_of_data_to_send[-1])
136         else:
137             if total_time_brushed > 110:
138                 print('toothbrush was maybe not held in hand')
139                 network_test_1.init_data(total_time_brushed,'tandbørste var muligvis ikke holdt i hånd')
140             # if brush time is too short and one fully completed brush haven't been done since reset, turn led ring red
141             if total_time_brushed < 110 and brushes_completed == 0:
142                 led_ring.led_red()
143                 print("didn't brush long enough")
144
145             # reset brush timer, count and start time
146             brush_time_start = 0
147             count = 0
148             start_time = time.time()
149

```

Figur 8.30, en del af main.py-filen

På linje 112 tjekkes om der er blevet børstet mindst 110 sekunder, eller om der allerede har været foretaget en fuldendt børstning tidligere, som var godkendt. Hvis ja lyser led-ringens op grønt. Der bliver på linje 118 og 119 taget to temperaturmålinger med den infrarøde temperatursensor, en “**object_temp**” som målet temperaturen på tandbørsten og en “**ambient_temp**” som mäter temperaturen på omgivelserne. Hvis “**object_temp**” er højere end “**ambient_temp**” kan man gå ud fra, at tandbørsten har været holdt i en hånd, og ikke bare været taget op og lagt på et bord. Hvis tandbørsten er varmere end lufttemperaturen, sendes dataen på linje 135 via funktionen “**init_data()**”. Hvis tandbørsten ikke er varmere end lufttemperaturen sendes det på linje 139, også med “**init_data()**” funktionen.

På linje 141 er der en condition der får led ringen til at lyse rødt hvis tiden tandbørsten har været væk fra glasset er under 110 sekunder, og der ikke har været foretaget en godtaget tandbørstning endnu siden sidste reset.

8.2.7: Ldr_reading.py på esp32

Denne funktion her bruges til at tage lysmålinger med en lys-niveau-sensor også kaldet en light-depending resistor.

```

1 from machine import Pin,ADC
2 from time import sleep
3
4 sensor = ADC(Pin(32))
5 sensor.width(ADC.WIDTH_10BIT)
6 sensor.atten(ADC.ATTN_11DB)
7
8 number_of_readings = 50
9
10 def ldr_reading():
11     reading = []
12     for i in range(number_of_readings):
13         reading.append(sensor.read())
14     average_of_readings = sum(reading)/number_of_readings
15     average_of_readings = round(average_of_readings)
16     #print(average_of_readings)
17     return average_of_readings
18

```

Figur 8.31, ldr_reading.py-filen

Pin, ADC importeres i toppen. Der oprettes et ADC objekt kaldet “**sensor**” forbundet til pin 32, dette ses på linje 4.

På linje 5 og 6 bliver “**width**” og “**attenuation**” sat, til henholdsvis 10 bit og 11db.

En variabel kaldet “**number_of_readings**” bliver sat til 50 på linje 8. Denne variabel bestemmer, hvor mange læsninger der ønskes at tages.

Fra linje 10 oprettes funktionen “**ldr_reading()**”, på linje 10 laves en tom liste kaldet “**readings**”, som skal indeholde de målte værdier.

På linje 12 starter et for-loop, der løber igennem 50 gange, og for hver gennemgang bliver der taget et måling fra LDR-sensoren med funktionen “**read()**”. Målingen kommer i listen readings med funktionen “**append()**”. Dette ses på linje 13.

Gennemsnittet af målingerne bliver udregnet på linje 14, og kommer i en variabel kaldet “**average_of_readings**”. På linje 15 bliver tallet afrundet til tal uden decimaler. Til sidst på linje 17 bliver resultatet så returneret, så det kan bruges når funktionen bliver kaldt.

8.2.8: Ir_test_works.py på esp32

Denne fil har koden til den infrarøde temperatursensor kaldet MLX90614.

Der bliver importeret Pin og SoftI2C i toppen af filen.

På linje 5 bliver en variabel oprettet, der indeholder sensorens i2c-adresse.

```

1 from machine import Pin, SoftI2C
2 import time
3
4 # MLX90614 I2C address
5 MLX90614_I2C_ADDR = 0x5A
6
7 # MLX90614 register addresses
8 MLX90614_TA = 0x06 # Ambient temperature
9 MLX90614_TOBJ1 = 0x07 # Object temperature
10
11 # Define I2C pins
12 i2c = SoftI2C(scl=Pin(22), sda=Pin(21), freq=100000)
13
14 def read_temperature(reg):
15     # Read 2 bytes from the specified register
16     data = i2c.readfrom_mem(MLX90614_I2C_ADDR, reg, 2)
17     # Convert the bytes to temperature value (Celsius)
18     temperature = (data[1] << 8 | data[0]) * 0.02 - 273.15
19     return temperature
20

```

Figur 8.32. ir_test_works.py-filen

Der oprettes to variabler: en til ambient temperatur og en til object. Temperatur adresser på linje 8 og 9. På linje 12 laver vi vores IC2-object, og her specificeres pin til serial clock og serial data. Fra linje 14 oprettes funktionen “**read_temperature()**”.

På linje 16 oprettes variablen “**data**” som indeholder en liste af to bytes, læst fra sensoren på den specificerede register adresse, enten ambient temperature eller object temperature.

På linje 18 oprettes variablen “**temperature**”, som indeholder et samlet 16-bit tal, som bliver rykket rundt korrekt via bit shifting, da sensoren sender den vigtigste byte først. Resultatet skal ganges med en faktor 0.02. Derefter har man temperaturen i kelvin, derfor skal der til

sidst trækkes 273.15 fra , så resultatet ender som celcius.

Det returneres til sidst på linje 19, så det kan bruges når funktionen kaldes.

8.2.9: Network_test_1 på esp32

```

1 import ubinascii
2 import network
3 import urequests
4 import time
5 import socket
6 import neopixel
7 from machine import Pin
8 import esp
9 esp.osdebug(None)
10 import gc
11 gc.collect()
12 import led_ring
13 import log_function
14 import _thread
15
16 # send check
17 sending_failure = False
18
19 # Network credentials
20 ssid_list = ['Oliver Bootss iPhone 11 Pro Max','TP-Link_99E2','LTE-1931','Watson'] #'Watson'
21 password_list = ['RuneLite','15853172','12345678','vuvcth8txxmeisx'] #'vuvcth8txxmeisx'
22
23 #Passcode for website access:
24 web_passcode="a4a6d723-8ece-4c24-b662-0285cf9f1e50"
25 #user_id="c1ac11a9-dc44-46c9-a990-ee2ac711e9c6" # user 1
26 user_id="4a8e5bf7-b83d-4d90-9f0c-c3fab3c2ce6b" # user 2
27 # send timer
28 send_timer = time.time()
```

Figur 8.33, network_test_1.py-filen

Diverse imports ses i toppen af filen. “**sending_failure**” er en boolean, der er sat til False på linje 17. Hvis en sending af data fejler, bliver den sat til True, og en funktion forsøger at gensende dataen så længe “**sending_failure**” er False. Når sendingen lykkedes, bliver “**sending_failure**” sat til false igen.

På linje 20 og 21 ses to lister af ssid og korresponderende kodeord. Disse bruges senere i funktionen “**net_connect()**” til at forbinde til et wi-fi.

Variablen “**web_passcode**” bruges til at få adgang til serveren, og variablen “**user_id**” fortæller hvem enheden er.

```

29
30 def connect_to_wifi(ssid, password):
31     #print('inside connect_to_wifi function')
32     try:
33         # For ESP to connect
34         station = network.WLAN(network.STA_IF)
35         station.active(True)
36         station.connect(ssid, password)
37         #print('ssid: ', ssid, 'pass: ', password)
38         station.mode(network.WLAN.N)
39     except Exception as e:
40         #print(f"Failed to connect to {ssid}")
41         #print(f"error: {e}")
42         #print('free memory: ',gc.mem_free())
43         print('...')

44
45
46 def net_connect():
47     for i in range(len(ssid_list)):
48         #print('len of ssid_list:', len(ssid_list))
49         #print('i: ', i)
50         print(f'trying to connect to {ssid_list[i]}')
51         connect_to_wifi(ssid_list[i], password_list[i])
52         #print(f'network: {ssid_list[i]} index: {i} - password: {password_list[i]} index: {i}')
53         time.sleep(10)
54         if network.WLAN(network.STA_IF).isconnected():
55             print(f'Connected to {ssid_list[i]}')
56             # led ring blinks blue when connected
57             led_ring.led_connect()
58             print('ip address: ',get_ip_address2()[0])
59             # exit function if connected
60             return

```

Figur 8.34, network_test_1.py-filen

Funktionen “**connect_to_wifi()**” ses fra linje 30 til 43. Den forsøger at forbinde til et wifi med et givent ssid og password. Hvis den ikke kan forbinde printer funktionen tre prikker.

“**connect_to_wifi**” funktionen bruges inde i den næste funktion, der hedder “**net_connect()**”. Den ses fra linje 46 til linje 60. Denne funktion har et for-loop, der kører for hvert element der er i “**ssid_list**”.

På linje 51 forsøges der så at forbinde til et wi-fi via den tidligere funktion “**connect_to_wifi**”. Funktionen prøver først at forbinde med “**ssid_list[0]**” og “**password_list[0]**”. Hvis den ikke forbinder, prøver den med index 1, så 2 osv.

Hvis den forbinder indikeres det på led-ringens via funktionen “**led_connect()**” som ses på linje 57.

```

115 def net_work_test():
116     # disconnect timer
117     disconnect_timer = time.time()
118     while True:
119         current_time = time.time()
120         # try to connect to available networks
121         if not network.WLAN(network.STA_IF).isconnected():
122             net_connect()
123             print('-----')
124             # disconnect from network every hour to get back on wifi in case it was down
125             if current_time - disconnect_timer > 3600: #3600 is 1 hour
126                 net_disconnect()
127                 disconnect_timer = time.time()
128

```

Figur 8.35, network_test_1.py-filen

“**net_work_test()**” funktionen starter på linje 115. Denne funktion bliver kørt som en thread inde i main-filen, og sørger for at så længe enheden ikke er forbundet til et wi-fi, forsøger den at forbinde til et wifi indtil det lykkedes.

```

86 def send_data(data):
87     s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
88     s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
89     s.connect(("10.10.4.10", 3000))
90
91     print('data: ', data)
92
93     s.sendall(str.encode("\n".join([str(data[0]), str(data[1]), str(data[2]), str(data[3]), str(data[4]), str(data[5]), str(data[6])])))
94
95     msg = s.recv(1024).decode("utf-8")
96     print('msg: ', msg)
97     print("values sent")
98
99
100 def init_data(length,activity):
101     global sending_failure # indsats global variabel, slet hvis alt fucker!
102     try:
103         new_data=[web_passcode, user_id, get_ip_address2()[0], mac_address(), time.time(), length, activity]
104         send_data(new_data)
105     except Exception as e:
106         print(f'Exception: {e}')
107         sending_failure = True # indsats bool, slet hvis alt fucker!
108         print(f'sending failure fra init_data = {sending_failure}')
109         log_function.write_data_to_file(new_data)
110         #resend_data()
111         # resend thread test
112         _thread.start_new_thread(resend_data,())
113
114

```

Figur 23, network_test_1.py-filen

Funktionen “**send_data()**” tager et argument kaldet data, og starter på linket 86.

Et socket objekt bliver oprettet på linje 87.

Den opretter en socket forbindelse til vores server på IP adressen og porten der ses på linje 89.

På linje 93 samles dataen i en string, og laves om til bytes og sendes.

På linje 95 bliver et svar decoded til utf-8 og printet på 96.

Denne funktion “**send_data()**” bruges inde i den næste funktion kaldet “**init_data()**”.

“**init_data**” tager to argumenter. Disse argumenter bliver indsatt i listen “**new_data**”, som ses på linje 103.

På linje 104 bliver funktionen “***send_data(new_data)***” såkaldt, nu med listen “***new_data***” som argument. Her bliver dataen forsøgt sendt.

Hvis sendingen fejler, bliver variablen “***sending_failure***” sat til værdien True. Derefter bliver den forsøgte sendte data gemt i et tekstdokument via funktionen “***write_data_to_file()***”, der ses på linje 109.

På linje 112 startes en tread, der forsøger at gensende dataen via funktionen “***resend_data()***”.

8.3 Konfiguration af valgte servere

I projektet har der været benyttet en server med MacOS-styresystem. Den har været hostet lokalt på et af gruppemedlemmernes computer, og der har i konfigurationsdelen derfor været taget udgangspunkt i dette. Dog er det muligt at omlægge konfigurationen til andre styresystemer, såfremt der er behov for det. Det virtuelle miljø kan tilpasses til andre styresystemer, eksempelvis Windows eller Linux. Hvis man ønsker at gøre det, kan man opsætte et nyt virtual environment i “app”-mappen, og skrive følgende kommando:

“pip install -r requirements.txt”

Ved at gøre det importeres de følgende moduler, og miljøet tilpasses styresystemet:
async-timeout==4.0.3

bcrypt==4.0.1

blinker==1.7.0

bottle==0.12.25

bottle-session==1.0

cffi==1.16.0

click==8.1.7

cryptography==41.0.7

Flask==3.0.0

Flask-Bcrypt==1.0.1

importlib-metadata==6.8.0

itsdangerous==2.1.2

Jinja2==3.1.2

MarkupSafe==2.1.3

Paste==3.7.1

pycparser==2.21

pyopenssl==23.3.0

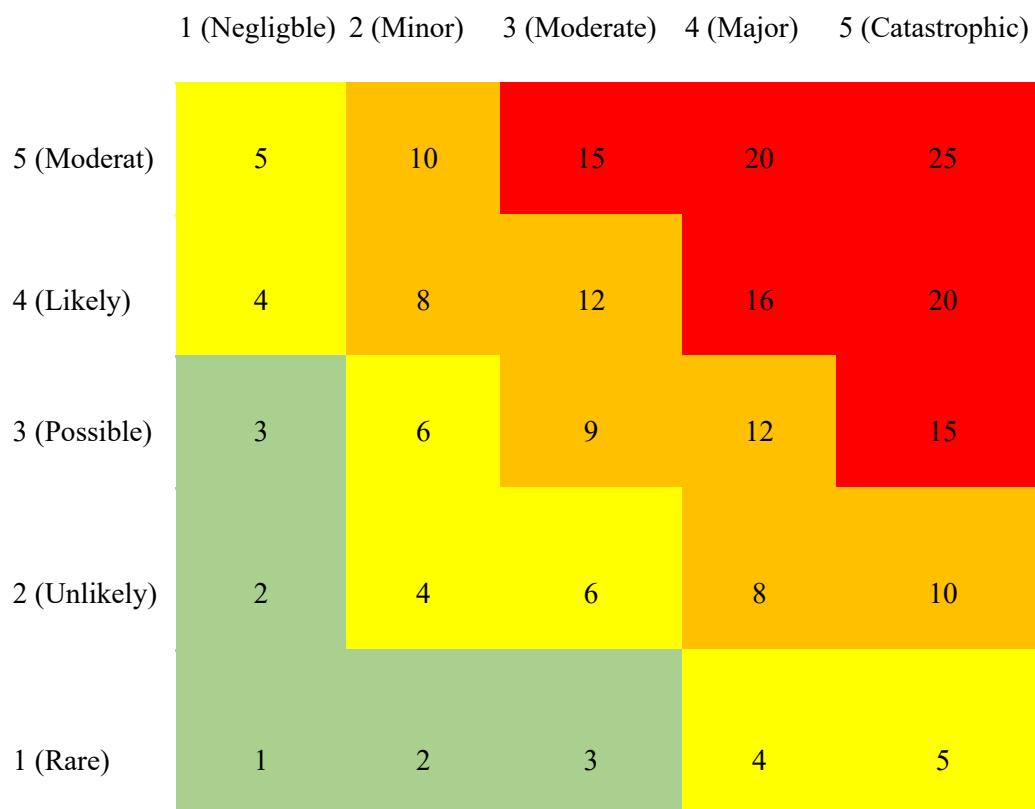
redis==5.0.1

six==1.16.0

Werkzeug==3.0.1

zipp==3.17.0

8.3.1 Hardeninglist af server



Figur 8.36, Risk table

Det er vigtigt at gøre sikkerhedsmæssige tiltag, når man har en server. I dette afsnit er der taget udgangspunkt i et scenarie, hvor der er benyttet en ubuntu-server. Det har ikke været tilfældet under projektet, men i et produktionsscenario ville man formentlig vælge en server med linux.

Herunder ses en liste af tiltag, man kan gøre sig for at hardne sin server:

- Kryptering af data

Det er vigtigt at kryptere den data, som ens server opbevarer. Såfremt dataen ligger i plaintext, er det lettere for angriberen at udnytte dataen, hvis vedkommende får uautoriseret adgang til serveren. Vurderingen på risiko-skalaen ville i dette tilfælde være 4(possible)/4(major), hvilket giver risikoen en samlet score på 12.

- Begrænsning af adgang til filer

Ved at begrænse adgangen til specifikke filer for de forskellige brugere af serveren begrænses angrebsfladen, da det dermed bliver sværere for en angriber at bevæge sig uautoriseret rundt i systemet. Det gør det også sværere at trække potentielt følsomme data ud af systemet. Vurderingen på risikoskalaen ville i dette tilfælde være 3(Possible)/3(Moderate), hvilket giver risikoen en samlet score på 9.

- Opsætning af firewall/begrænsning på traffik

Ved at opsætte en firewall eller sætte begrænsninger på trafikken til og fra serveren gøres det mere usandsynligt, at der er traffik til og fra uhensigtsmæssige steder. At opsætte begrænsning på trafikken er et relativt simpelt tiltag, som serveren sikkerhedsmæssigt kan have stor gavn af. Eksempelvis kunne der opsættes regler for hvilke IP-adresser serveren må kommunikere til og modtage trafik fra.

Dette er mere en anbefaling end en risikovurdering, og der tages derfor ikke direkte stilling til en score for dette hardeningtiltag.

- Backup

Det er vigtigt jævnligt at tage backup af den data, der ligger på ens server. På den måde er der større sikring for, at man ikke har databasen komplet mister data under eksempelvis et ransomware-angreb. Backups bør opbevares et remote sted fra serveren, så der ikke er nogen direkte kontakt mellem serveren og lokale backups opbevares. Vurderingen på risikoskalaen ville i dette tilfælde være 3(Possible)/5(Moderate). Det giver risikoen en samlet score på 15, da det ville have meget store konsekvenser at miste data på baggrund af ransomware.

8.4 Dokumentation af sikkerhedstiltag

Et krav for projektet var, at der skulle være minimum tre sikkerhedstiltag. Herunder beskrives de sikkerhedstiltag, der er gjort i projektet.

8.4.1 Kryptering af databasen

Som et sikkerhedstiltag er data, som opbevares i databasen, krypterede. Årsagen til at man vælger at gøre det er, at det giver privacy og security af user data, samt at gøre det sværere at udnytte dataen, hvis databasen komprimeres. [6] At opbevare passwords og andet personligt data i plaintext giver stor risiko for udnyttelse. Det er derfor vigtigt at hashe/kryptere data

med en stærk hashing algoritme. '**bkrypt**' er den hashing algoritme, der er benyttet i dette projekt. Dens generelle funktionalitet er at tage et user-input og filtrere det igennem en ukendt algoritme med en nøgle som kun serveren har.

8.4.2 Regex/SQL-injection-sikring ved login

SQL-injection er en type angreb, der udnytter sårbarheder i databasekald og brugerinput. Sårbarheden forekommer, hvis ikke inputtet fra brugeren er saniteret for specielle tegn og sætninger ved login forsøg. Hvis ikke dette sikkerhedstiltag er foretaget, har angriberen mulighed for at sende uhensigtsmæssige databasekald, som kan give uautoriseret adgang til sider af websitet, vedkommende ikke bør have adgang til. [12]

Man kan som udvikler sikre sit website mod SQL-injections ved at implementere Regex, som sætter krav og begrænsninger til måden, der kan sendes databasekald på. Brugeren skal holde sig inden for nogle satte parametre for, hvordan f.eks. brugernavn og password kan være opbygget. Der er implementeret Regex i websitet for dette projekt som et sikkerhedstiltag.

8.4.3 Multi-factor Authentication ved login

Når der logges ind på websitet tjekkes der i databasen for, om brugeren findes. Såfremt brugeren gør det, modtager vedkommende en mail med en tre-cifret kode, som skal bruges til at gennemføre det sidste stadie af login. Ved at implementere dette ekstra sikkerhedslag gives der ekstra validering for, at det er den rigtige person, som får adgang til websitets data. Som beskrevet i navnet gives der "**multiple factors of validation**" for, at vedkommende er den, de giver sig ud for at være. [2]

9. Test af løsning

9.1 Pathfinding testresultater

Der blev foretaget tests af løsningen over to dage. Disse tests kalder vi testcyklus 1 og testcyklus 2. Testcyklus 1 blev foretaget på første testdag, og der testes enhed og website, men uden opsætning af netværket. I testcyklus 2 var netværket implementeret og indgik i testen. Når der herfra henvises til testcyklus 1 og testcyklus 2 henvises der til dette.

Testcyklus 1 forløb relativt uproblematisk. Enheden og hjemmesiden opførte sig forholdsvis hensigtsmæssigt, og i det store hele endte testen med at være vellykket. Der var mindre fejl undervejs, blandt andet u-pålidelige målinger fra enheden og at databasen ikke krypterede altting korrekt. Det blev løst, og derefter fungerede testen som håbet.

Testcyklus 2 forløb en smule mere problematisk, da det var nødvendigt at foretage forskellige ændringer for at få netværket til at fungere. Dette inkluderede blandt andet justering af portnummer, justering af IP-adresser så de var tilsvarende det interne netværk og upålidelige målinger fra enheden. Til slut endte den samlede løsning med at fungere over netværket, og der sendtes resultater fra enheden til server.

En dybdegående beskrivelse af begge test-cyklusser kan findes nedenfor.

9.2 Dokumentation af DUT

9.2.1 Billeder og video af samlet løsning

Gennem følgende links findes videodokumentation af begge test cyklusser.

Testcyklus 1: <https://www.youtube.com/watch?v=Eq8trwjDqTg>

Testcyklus 2: <https://www.youtube.com/watch?v=qkrNCXcN0ug>

9.3 Udførelsel af test

| Dato | Testbeskrivelse og resultater - Testcyklus 1 |
|--|---|
| 15-12-23 | <p>Første testcyklus som bestod i at teste enheden i samspil med website. Løsningen var på dette tidspunkt ikke sat til det dedikerede netværk, men kommunikerede over internettet.</p> <p>Der var små fejl i målinger fra temperatursensoren, som senere blev justeret. Lyssensoren opførte sig først ikke hensigtsmæssigt, men dette blev rettet til efterfølgende. Websitet opførte sig som forventet. Dog krypterede databasen i første omgang ikke altid korrekt, men fejlen blev rettet efter den blev opdaget.</p> |
| Samlet test tid under cyklus: 75 minutter | |

| Dato | Testbeskrivelse og resultater - Testcyklus 2 |
|---|---|
| 18-12-23 | <p>Anden testcyklus bestod i at teste kommunikation mellem enheden og hjemmesiden over det dedikerede netværk.</p> <p>Der opstod komplikationer med først ukorrekt portnummer, og derefter u-korrekte IP-adresse på serveren. Det gav nogle fejl, som projektgruppen brugte et stykke tid på at finde frem til.</p> <p>Da dette blev løst, og derefter var det muligt for enheden og serveren at kommunikere over netværket.</p> <p>Der var problematikker i forhold til, at det ikke var muligt at generere nok varme på ‘tandbørsten’ hver gang til kontinuerligt at give en korrekt melding i forhold til, om tandbørsten havde været holdt i hånden eller ej. Der kom derfor nogle ustabile meldinger/målinger fra enheden til serveren, som gjorde, at løsningen ikke var komplet pålidelig i sine svar. Efter flere justeringer blev resultaterne gjort mere præcise, men enheden er i skrivende stund endnu ikke komplet pålidelig i det aspekt af resultaterne.</p> |
| Samlet test tid under cyklus: 120 minutter | |

9.4 Udførsel af Accepttest på DUT

9.4.1 Første testcyklus

| | | |
|-----------------------------|---|---------------------|
| ID:1.1 | Krav: Løsningen skal kunne detektere, om tandbørsten har været løftet fra holderen. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan detektere, om tandbørsten har været løftet ud af holderen, betragtes kravet som passed. | Passed |

Resultat: Med vores test-objekt, som lå i holderen da testen begyndte, løftede vi den fra holderen og LED-ringene gik i gang med at lyse op som programmeret. Da de krævede to minutter var gået, lagde vi objektet tilbage i holderen, og i det sende dataen til serveren.

| | | |
|-----------------------------------|---|---------------------|
| ID:2.1 | Krav: Løsningen skal kunne afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidsrum. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen kan afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidspunkt, anses kravet for at være passed. | Passed |

Resultat: Da serveren havde fået IP'en af holderen, trykkede vi på "Påmindelse" knappen på dashboardet. Holderen gav lyd fra sig, som fortæller os at forbindelsen lykkedes og dataen blev sendt til holderen.

| | | |
|-------------------------------------|--|---------------------|
| ID:3.1 | Krav: Løsningen er kompatibel med de mest almindelige forekommende tandbørster, elektriske som ikke elektriske. | Prioritet: 2 |
| Kategori: Brugervenlighed | Accepttest: Kan krav 1 kan opfyldes med forskelligt udformede tandbørster, anses kravet værende opfyldt. | tested |

Resultat: Vi testede i denne cyklus med én tusch, der har ca. samme omrids som en almen ikke-elektrisk tandbørste. Vi testede ikke med rigtige tandbørster her, så vi kan ikke konkludere at testen har haft direkte succes. Dog kan vi heller ikke sige, at det har fejlet, da omridset passer til de fleste af ikke-elektriske tandbørster.

| | | |
|-----------------------------|---|---------------------|
| ID:4.1 | Krav: Løsningen skal have en indikator, der viser om tandbørstningen har været i gang i de anbefalede 2 min, samt en indikation for hvor tæt på de 2 min man er. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Kan løsningen vise når der er gået 2 min, og indikere hvor tæt på 2 min man er kommet, anses kravet for værende bestået. | Passed |

Resultat: Da børsten blev løftet fra holderen, startede protokollen, der tæller tiden mellem løftet og nedsætningen af børsten. Tælleren går op til to minutter, hvorpå LED-ringens gradvist lyser op, som tiden går. Når de to minutter er gået, blinker den blåt, som fortæller brugeren at børstningen af færdig. Når brugeren så lægger børsten ned, starter den forsendelses-protokollen. Hvis brugeren lægger børsten ned før disse to minutter er gået, stopper den protokollen og går tilbage til det tidligere stadie. Derfor kan vi med sikkerhed sige at dette krav er vellykket.

| | | |
|-----------------------------|---|---------------------|
| ID:5.1 | Krav: Tidtagning for 2 min anbefalet børstning, starter når tandbørsten løftes fra løsningen. | Prioritet: 2 |
| Kategori: Sensing | Accepttest: Når tandbørsten løftes fra løsningen og tidsendikationen starter, anses kravet for værende bestået. | Passed |

Resultat: Med kontekst fra resultatet af 4.1, kan vi også sige at denne test også er vellykket, da lige så snart vores objekt blev løftet ud af holderen, gik tælleren i gang med at tælle op og LED-ringens viste denne fremgang.

| | | |
|-----------------------------|---|---------------------|
| ID:6.1 | Krav: Løsningen skal kunne måle temperaturen for at bestemme, om tandbørsten har været holdt i hånden. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan måle temperatur for at bestemme, om tandbørsten har været holdt i hånden, anses kravet for værende bestået. | Passed |

Resultat: Efter at de to krævede minutter er gået, og objektet bliver lagt tilbage i holderen, mäter den interne temperatur mäter hvor varm "skaftet" er på børsten. Vi kan se at den opsamler denne information og sender den til serveren med en før-skrevet string, der beskriver om børsten har været holdt ordentligt eller slet ikke været brugt. Hvilken besked der bliver sendt, afhænger af børstens temperatur.
 I testen er dette vellykket, og derfor er set som Passed

| | | |
|-----------------------------------|--|---------------------|
| ID:7.1 | Krav: Kunne sende informationer om brug til serveren ved børstning. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Når serveren modtager informationerne fra løsningen omkring tandbørstning, anses kravet som værende bestået. | Passed |

Resultat: Som beskrevet i 6.1, sender den informationen til serveren via sockets, som derefter kan ses på hjemmesiden.

| | | |
|-----------------------------------|---|---------------------|
| ID:8.1 | Krav: Hvis løsningen ikke kan nå serveren ved hjælp af det ønskede netværk, skal den kunne overgå til et alternativt netværk. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen ikke taber data ved at miste forbindelse til wifi, men i stedet overgår til et alternativt netværk anses kravet for at være opfyldt | passed |

Resultat: Når holderen ikke kan forbinde til det lokale netværk, skifter den til andre netværk den kender. Dog på grund af vores opsætning i cyklus 1, virker forbindelsen ikke på nettet. Kun lokalt.

Hvis den prøver at sende oplysningerne, men kan ikke forbinde til serveren, gemmer den resultatet midlertidigt i en fil, og prøver at sende dem lidt senere.

9.4.2 Netværk

| | | |
|-------------------------------|---|---------------------|
| ID:1.2 | Krav: Der skal gøres tre sikkerhedstiltag for netværk/enheden | Prioritet: 1 |
| Kategori: Hardening | Accepttest: Såfremt der er gjort tre tiltag for at sikre netværk/kommunikation mellem netværk og enheden, anses kravet for at være opfyldt. | Passed |

Resultat: Serveren har flere sikkerhedstiltag, der forhindrer uautoriseret adgang til, og beskyttelse af, filer og data associeret med den. Disse inkluderer kryptering/hashing af password, borgerinformation og andet, samt MFA til login og mere. Derfor kan vi med sikkerhed godkende dette krav som opfyldt.

| | | |
|------------------------|--|---------------------|
| ID:2.2 | Krav: Klienten/klinikken skal kunne tilgå og se data fra databasen ved hjælp af et website. | Prioritet: 1 |
| Kategori: UI | Accepttest: Såfremt klienten/klinikken kan tilgå et website, der fremviser data fra databasen, anses kravet for at være bestået. | Passed |

Resultat: Klienten, som har autoriseret ansatte til at holde styr på siden, kan tilgå siden med deres unikke login. Derfra kan de via dashboardet se alle enheder i systemet og påvirke enheden, samt ændre på oplysningerne. Derfor er dette krav opfyldt.

| | | |
|-----------------------------------|--|---------------------|
| ID:3.2 | Krav: Den samlede løsning skal have to netværk, og der skal kunne kommunikeres via Wifi og WAN mellem disse netværk. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt der skabes to netværk, som kan kommunikere via Wifi og ethernet, anses kravet som værende bestået. | failed |

Resultat: I cyklus 1 var forbindelsen mellem serveren og holderen gennem et mobilt-hotspot. De to enheder kunne ikke forbinde til hinanden gennem andre netværk, så derfor må vi sige at dette krav ikke er opfyldt i cyklus 1.

| | | |
|-----------------------------------|---|---------------------|
| ID:4.2 | Krav: Det skal være muligt at starte en aktuator fra server til enheden. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt der kan startes en aktuator fra server til enheden, anses kravet for at være opfyldt. | Passed |

Resultat: Hjemmesiden har en knap der forbinder enheden som den specifikke borger er tildelt, og sender et signal som enheden genkender via sockets. Dette signal bærer en boolean, som enheden ud fra det, aktiverer en buzzer, der er programmeret til en alarm, der påmindrer borgeren. Derfor er dette krav opfyldt.

9.4.3 Anden testcyklus:

| | | |
|-----------------------------|---|---------------------|
| ID:1.1 | Krav: Løsningen skal kunne detektere, om tandbørsten har været løftet fra holderen. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan detektere, om tandbørsten har været løftet ud af holderen, betragtes kravet som passed. | Passed |

Resultat: vi tilsluttede vores tandbørsteholder og placerede en tusch i holderen, efter det havde forbundet til wi-fi løftede tuschen, enheden registrerede løftet og begyndte en optælling af tid. efter 120 sekunder var forløbet blinkede vores tandbørsteholder, og vi placerede tuschen tilbage i holderen.

| | | |
|-----------------------------------|---|---------------------|
| ID:2.1 | Krav: Løsningen skal kunne afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidsrum. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen kan afgive lyd ved manuelt input fra klinikken over nettet, hvis tandbørsten ikke har været løftet ud af holderen inden for et givent tidspunkt, anses kravet for at være passed. | Passed |

Resultat: Efter serveren fik Ip adressen fra holderen blev “påmindelse” knappen på dashboardet trykket
Tandbørsteholderen buzzede tre gange og havde altså modtaget påmindelsen fra serveren.

| | | |
|-------------------------------------|--|---------------------|
| ID:3.1 | Krav: Løsningen er kompatibel med de mest almindelige forekommende tandbørster, elektriske som ikke elektriske. | Prioritet: 2 |
| Kategori: Brugervenlighed | Accepttest: Kan krav opfyldes med forskelligt udformede tandbørster, anses kravet værende opfyldt. | tested |

Resultat: vi testede denne cyklus med to forskellige tuscher, en tynd og en tykkere, men da vi ikke har testet med rigtige tandbørster endnu, hverken “analog” eller elektriske kan vi ikke sige denne del er rigtigt opfyldt.

| | | |
|-----------------------------|---|---------------------|
| ID:4.1 | Krav: Løsningen skal have en indikator, der viser om tandbørstningen har været i gang i de anbefalede 2 min, samt en indikation for hvor tæt på de 2 min man er. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Kan løsningen vise når der er gået 2 min, og indikere hvor tæt på 2 min man er kommet, anses kravet for værende bestået. | Passed |

Resultat: løftes tandbørsten fra holderen går tælleren igang med at tælle op, når tælleren til 120 sekunder blinker led-ringen blåt for at indikere at børstningen kan afsluttes, bliver tandbørsten placeret i glasset efter ringen blinker blåt, skifter led-ringens farve til grøn og indikere nu en komplet børstning er fuldendt. Hvis tandbørsten kommer tilbage i glasset før de 120 sekunder er passeret, lyser led-ringens rødt.

| | | |
|-----------------------------|---|---------------------|
| ID:5.1 | Krav: Tidtagning for 2 min anbefalet børstning, starter når tandbørsten løftes fra løsningen. | Prioritet: 2 |
| Kategori: Sensing | Accepttest: Når tandbørsten løftes fra løsningen og tidsendikationen starter, anses kravet for værende bestået. | Passed |

Resultat: Med kontekst fra resultatet af 4.1, kan vi også sige at denne test også er vellykket, da lige så snart vores objekt blev løftet ud af holderen, gik tælleren i gang med at tælle op og LED-ringen viste denne fremgang.

| | | |
|-----------------------------|---|---------------------|
| ID:6.1 | Krav: Løsningen skal kunne måle temperaturen for at bestemme, om tandbørsten har været holdt i hånden. | Prioritet: 1 |
| Kategori: Sensing | Accepttest: Såfremt løsningen kan måle temperatur for at bestemme, om tandbørsten har været holdt i hånden, anses kravet for værende bestået. | Passed |

Resultat: Efter at de to krævede minutter er gået, og objektet bliver lagt tilbage i holderen, mäter den interne temperatur mäter hvor varm ”skaftet” er på børsten. Vi kan se at den opsamler denne information og sender den til serveren med en før-skrevet string, der beskriver om børsten har været holdt ordentligt eller slet ikke været brugt. Hvilken besked der bliver sendt, afhænger af børstens temperatur.

I testen er dette vellykket, og derfor er set som Passed

| | | |
|-----------------------------------|--|---------------------|
| ID: 7.1 | Krav: Kunne sende informationer om brug til serveren ved børstning. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Når serveren modtager informationerne fra løsningen omkring tandbørstning, anses kravet som værende bestået. | Passed |

Resultat: Som beskrevet i 6.1, sender den informationen til serveren via sockets, som derefter kan ses på hjemmesiden.

| | | |
|-----------------------------------|---|---------------------|
| ID:8.1 | Krav: Hvis løsningen ikke kan nå serveren ved hjælp af det ønskede netværk, skal den kunne overgå til et alternativt netværk. | Prioritet: 1 |
| Kategori: Kommunikation | Accepttest: Såfremt løsningen ikke taber data ved at miste forbindelse til wifi, men i stedet overgår til et alternativt netværk anses kravet for at være opfyldt | Passed |

Resultat: Kan tandbørsteholderen ikke forbinde til vores netværk, forsøger den at forbinde til en række andre netværk indtil den får forbindelse i et loop indtil den får forbindelse. Hvis tandbørsteholderen forsøger at sende data til serveren, men ikke kan nå den, gemmes den forsøgt sendte data i en tekstfil, og tandbørsteholderen forsøger så at sende dataen igen med et 10 sekunders interval indtil det lykkedes.

10. Implementering af løsning i drift

I og med at løsningen på nuværende tidspunkt blot er en prototype på et færdigt produkt, ville der være en del tilrettelser at gøre, hvis man skulle forsøge at bringe den ud på markedet. I flere aspekter er der ting på løsningen, som kunne forbedres eller forfines inden den bringes i produktion. Herunder er blandt andet investering i bedre eller anderledes sensorer, samt forfine design og funktionalitet.

Hvem løsningen skal appellere til er todelt: Klinikker og borgere. Det er vigtigt at imødekomme begge, da den ene ikke kan eksistere uden den anden. Derfor ville løsningen jævnligt evalueres for, om brugerne på begge sider ønsker noget anderledes.

Opbygningen for hver sektion af systemerne ville være, at der er en klinik, som er bruger af websitet med nogle tilknyttede klienter, som har hver deres enhed i hjemmet. Man kunne opsætte en dedikeret server til hver sektion, eller lave en samlet server i en større skala. Alle enheder skal kunne konfigureres til udelukkende at have kommunikation til serveren. Der ville derfor være sikkerhedsmæssige tiltag at skulle gøre i den forstand. Dette kunne opnås ved at opsætte regler for kommunikation mellem serveren og enheder.

Hjemmesiden i sig selv er en basisk prototype i både design og funktionalitet. Den kunne indrette sig efter de etablerede design normer af traditionelle programmer som sundhedssektoren bruger, for at gøre det lettere for brugeren at tilpasse sig det nye system. Klarheden i elementerne på siden kan forstærkes, og funktionalitet kan blive udvidet.

Løsningen har også potentielle til at nå ud til en klientel, som ikke udelukkende er individer med demens. Hjemmeboende ældre kan potentielt også have gavn af løsningen, da tandpleje også kan være en problematik i sig selv, selvom vedkommende helbred stadig er godt.

11. Praktisk projektplanlægning og ledelse

11.1 WBS

WBS er forkortelsen for det projektledelsesværktøj, der hedder Work Breakdown Structure. Det bruges til at nedbryde et projekt i mindre opgaver, hvilket har til formål at gøre arbejdsprocessen mere overskuelig. Strukturen er som vist i bilag 2. Fordelene ved brug af WBS i projektledelse er bedre overblik, bedre synlighed af fremskridt i processen, reduceret risiko for at lave fejl ved at overse opgaver og bedre planlægning. [7]

I dette projekt har WBS været brugt til at nedbryde en kompleks arbejdsproces i mindre opgaver. Det har givet arbejdsgruppens medlemmer bedre overblik over de opgaver, der skulle udføres for at komme i mål med projektet til tiden. WBS har også gjort det lettere at uddeleger opgaver til medlemmerne af arbejdsgruppen.

11.2 Kanban

Kanban er et projektstyringsværktøj, som bruges til at visualisere fremgang og proces i et agilt projekt. Grundpræmissen for Kanban er at lave et board, hvor på der sættes cards i forskellige kolonner. Det fortæller noget om hvilke opgaver, der er hvilke steder i arbejdsprocessen. [8]

I dette projekt er kanban brugt som projektledelsesværktøj i den forstand, at det er brugt til at holde styr på arbejdsprocessen internt i gruppen. Ved visuelt at vise, hvilke opgaver der er, og hvor opgaverne er i arbejdsprocessen, er det lettere at uddeleger og prioritere opgaver. Et udsnit af projektets kanban-board i de forskellige stadier i projektet kan ses i bilag 3.

11.3 Virksomhedsanalyse

I projektet beskæftiges der med København Kommunes tandpleje, som er en del af den offentlige sektor. Den offentlige sektor er i alle henseender tilbud for borgere, og udgifterne betales i langt de fleste tilfælde af staten. Organisationen får tildelt penge af staten, som kan bruges på forskellige typer behandling i en lang række tilbud til borgere [5]. I projektet er der besluttet at lave organisationsanalyse med baggrund i analysemetoden ‘Laloux farver’.

Den offentlige sektor er styret fra toppen og ned. Øverst tages beslutninger vedrørende den overordnede økonomi, og de underliggende afdelinger indordner sig efter dette. Der er hierarkisk inddeling, og hver underafdeling svarer til den ledelse, der ligger lige over. Organisationen følger fastsatte procedure, som kan være svære at ændre på. Det kan være vanskeligt fra en nedre afdeling at få kontakt til en afdeling længere oppe i hierarkiet, og det kan potentielt fraholde organisationen fra ændringer til det bedre for den enkelte afdeling. [9]. På den baggrund kan man kategorisere organisationen til at have et ravfarvet verdensbillede.

Dog kan man argumentere for, at organisationen har et ønske om at bevæge sig i en retning, som har værdier fra det orange og det grønne verdensbillede. Der udgives årligt en rapport med nationale mål for sundhedsvæsenet. Som i det orange verdensbillede forsøger sygehusvæsenet konstant at bevæge sig i en mere effektiv retning med udbytte af højere kvalitet. Det er også hensigten at skabe et godt miljø for de mennesker, der er ansat i sundhedsvæsenet, da det giver dem en større lyst til at blive i branchen. [4] At fokusere på medarbejdernes arbejdsglæde og motivation stammer fra det grønne verdensbillede. Det er ofte real-billedet, at organisationer indeholder værdier fra flere forskellige verdensbilleder, og den offentlige sektor er ingen undtagelse.

12. Konklusion

Ud fra ovenstående kan det konkluderes, at det var muligt at lave en simpel prototype, der matchede ønsket i projektbeskrivelsen. Forventningerne til projektet var en funktionelt prototype, der har basale features og kan vise data på et website. Dette lykkedes i løbet af projektet, og det skete inden for tidsrammen.

Det lykkedes at gennemføre et interview med en fagperson, som havde input til, hvordan man kan udarbejde en løsning, der efterkommer nogle af de behov, som målgruppen har. Dog var det ikke muligt at opfylde alle behov inden for tidsgrænsen. Derfor er nogle af de valgte features til prototypen baseret på, hvad der var muligt med den tid og ressourcer, der var givet.

13. Projektforløbet

Dette projektforløb markerede det fjerde større projekt, som gruppen har udarbejdet sammen. Dynamikken gruppemedlemmerne imellem var derfor på forhånd etableret, og der var ikke behov for at se hinanden an eller gå forsigtigt til værks. Det gjorde, at projektet fra start kunne planlægges og uddelegeres hensigtsmæssigt, hvilket gav anledning til mindre tidspres til slut i forløbet. Gruppen havde allerede tidligt i forløbet en klar forestilling om, hvor projektet skulle bære henad, og den tankegang holdt ved hele projektperioden ud. Planlægning og tidsstrukturering forløb optimalt. Alle gruppemedlemmer kendte til hinandens styrker og svagheder, og det gjorde, at alle på forhånd var indforståede med, hvor man hver især kunne gavne hinanden bedst.

I tidligere projektforløb har der undervejs været elementer, der har forsinket den samlede proces. Dette har haft konsekvenser for det tidspres, som gruppen har oplevet til sidst i forløbet. I dette forløb har der ikke været uhensigtsmæssige forsinkelser i tidsrammen for de forskellige elementer i projektet. Dette har resulteret i, at der har været mere tid til forfinelse af de enkelte elementer i løsningen. Dette indebærer blandt andet forbedring af hjemmesiden med ekstra layoutmæssige funktioner og bedre tid til print af låg til modellen.

14. Perspektivering

Dette projekt er blot et lille skrab i overfladen af det enorme felt, der udgører IoHT-udstyr i sundhedssektoren, og skal betragtes som et simpelt bud på et enhed, der kunne være fordelagtigt at implementere.

Såfremt der havde været adgang til flere økonomiske ressourcer og et længere tidsforløb, havde det været muligt at udvikle mere komplekse funktioner. I udvælgelsen af sensorer til løsningen var budget og tidsbegrænsning ligeledes en hæmmende faktor, som afgjorde valget. Der findes mere fintfølende sensorer til formålet, som ville have givet løsningen et nærmere finish.

Generelt er der mange facetter i projektet, som kunne have været forfinet eller forbedret. Der er forud for udarbejdelsen af vores projekt lavet forskellige prototyper og tiltag, som minder om denne løsning. Disse projekter har hver især haft forskellige innovative idéer som vi kunne have haft brug af.

Det ville også være fordelagtigt at foretage mere dybdegående research af målgruppen for løsningen. Dette gør sig både gældende for de borgere, der skal være brugere af enheden, og for de klinikker, der skal være brugere af websitet. At inddrage flere fagpersoner og flere borgere ville have givet bedre indblik i de behov, den respektive målgruppe har. Da der har været en begrænset tidsramme for projektet, skal de foretagne undersøgelser derfor blot ses som en stikprøve i en stor målgruppe.

15. Litteraturliste

15.1 Kilder fundet på internettet

- [1] Flagg, Anna, Boger, Jennifer, Mihailidis, Alex, "An intelligent toothbrush: Machines for Smart Brushing", https://www.resna.org/sites/default/files/legacy/conference/proceedings/2011/RESNA_ICTA/flagg-70178.pdf (5-12-23)
- [2] "Authentication (MFA) and How Does it Work?", <https://www.onelogin.com/learn/what-is-mfa> (10-12-23)
- [3] Jannati, Mohammad, O'Byrne, Sarah, Moussavi, Zahra, "Design, Implementation and Evaluation of a smart toothbrush for individuals with dementia", <https://asmadigitalcollection.asme.org/BIOMED/proceedings/DMD2023/86731/V001T09A011/1163830> (6-12-23)
- [4] Tjørnelund, Anne-Mette, Pedersen, Louise Møller, "Et godt arbejdsmiljø får sygeplejesker til at blive i faget", <https://dsr.dk/fag-og-udvikling/sygeplejersken/arkiv/fag-forskning-argang-2022-nr-3/et-godt-arbejdsmiljoe-faar-sygeplejersker-til-at-blive-i-faget/> (6-12-23)
- [5] Borum, Finn, "Forskning i sygehuse under forandring", https://research-api.cbs.dk/ws/portalfiles/portal/61314474/Forskning_i_Sygehuse_under_Forandring.pdf (4-12-23)
- [6] Heddings, Anthony, "How to Properly Store Passwords: Salting, Hashing, and PBKDF2", <https://www.hwtogeek.com/devops/how-to-properly-store-passwords-salting-hashing-and-pbkdf2/> (10-12-23)
- [7] Guldbrandsen, Mark, "Hvad er Kanban? Få en introduktion til Kanban!", <https://blivprojektleder.dk/kanban/> (29-11-23)
- [8] "IoT kan støtte fremtidens digitale sundhedssektor", <https://indblikplus.dk/fremtidens-sundhedssektor/iot-kan-stoette-fremtidens-digitale-sundhedssektor/> (4-12-23)

- [9] "Nationale mål for sundhedsvæsenet",
<https://sum.dk/Media/637697073524473744/Nationale%20M%C3%A5l%20for%20Sundhedsv%C3%A6senet%202021.pdf> (5-12-23)
- [10] "Plejebolig og plejehjem - Få hjælp til et godt liv på plejehjem",
<https://www.aeldresagen.dk/viden-og-raadgivning/hjaelp-og-stoette/plejebolig-og-plejehjem> (4-12-23)
- [11] Guldbrandsen, Mark, "Projektnedbrydning - Hvad er PBS/WBS?",
<https://blivprojektleder.dk/projektnedbrydning/> (29-11-23)
- [12] Kingthorin, "SQL Injection", https://owasp.org/www-community/attacks/SQL_Injection (10-12-23)
- [13] Wagner, Stefan, Eriksen, Cecilie, Hede, Børge, Christensen, Lisa, "Toothbrushing compliance tracking in a nursing home setting using telemonitoring-enabled powered toothbrushes" (6-12-23)

15.2 Personrelaterede kilder

- [14] Cecilie, Interview med tandlæge fra Københavns Kommunes tandpleje.

15.3 Kilder relateret til BOM-liste

- [1.1] Dokumentation for ESP32, <https://www.espressif.com/en/products/socs/esp32>
- [1.2] Dokumentation for Neopixel-LED-ring, <https://mit.kea.dk/labs/alle-maskiner/neopixel-led-ring>
- [1.3] Dokumentation for passiv buzzer, <https://mit.kea.dk/labs/alle-maskiner/passiv-buzzer>
- [1.4] Dokumentation for IR-temperatursensor,
<https://www.digikey.dk/da/products/detail/melexis-technologies-nv/MLX90614ESF-AAA-000-TU/1647940>

- [1.5] Dokumentation for lyssensor, <https://mit.kea.dk/labs/alle-maskiner/lys-niveau-sensor>

16. Bilag

16.1 Bilags-oversigt

- 16.1.1 Transkription af interview med Cecilie
- 16.1.2 WBS
- 16.1.3 Kanban

16.1.1 Transkription af interview med Cecilie

Adam:

Hvad er din proffesion?

Cecilie:

Jeg er tandlæge

Adam:

Hvor arbejder du henne?

Cecilie:

Det hedder voksentandplejen københavnns kommune

Adam:

Hvor mange ansatte er i ca i voksentandplejen?

Cecilie:

Vi er mellem 60 og 70 ansatte

Adam:

Har du en ide om hvor mange borgere i tilser om måneden ca?

Cecilie:

Om måneden.. ? nej det har jeg faktisk ikke - men jeg ved hvor mange vi har tilmeldt. Jeg kan godt finde ud af det hvis du godt vil vide det mere præcist, det er rimeligt mange. Vi har nu skal jeg regne.. Vi har 3 store klinikker med 7 stole.. Hvis du siger syv stole som kører hver dag og hvis du regner med der er i gennemsnit,, hvad kan vi nå på en dag.. Mellem 6 og 10 patienter, så lad os sige der er 8 patienter på en dag, gange med 7, gange 5, og så også det løse oven i .. så hva bliver det? Ihvertfald over 300.

Adam:

Er det udelukkende seniorer i tilser eller?

Cecilie:

Nej nej, vi har både omsorgstandplejen, som jo er ældre med funktionshæmning, psykisk eller fysisk, typisk er det plejehjemsbeboere som ikke kan komme til normal tandlæge.

Så har vi specialtandplejen som er det jeg primært er I nu, hvor vi har alle over 18, som ikke kan benytte normal tandplejetilbud på grund af deres fysiske og psykiske kunnen. Det vil typisk være folk med syndromer, folk der har udviklingshæmning, hjerneskader, erhvervede skader og så er der alle de fysiske skader, også erhvervede skader og så er der dem som også ligesom måske har nogle nerveskader, altså sklerose, ALS, sådan noget, parkinsons, og så er der en rimlig stor gruppe I københavn særligt psykisk syge, mange skitsofrene, PTSD, angst, OCD, borderline og så har vi en del autister også.

Adam:

Aha

Cecilie:

Og socialtandplejen som er for socialt udsatte som ikke kan gå ved en normal tandlæge da de har nogle sociale udfordringer, så det vil typisk være gadehjemløse, stofmisbrugere eller andre misbrugere eller skæve eksistenser. så vi dækker alle dem, engang var det va. 1% af befolkningen som ikke kan gå, det vil nok være flere idag, men dem som ikke kan være andre steder, dem har vi – og det er udelukkenden på grund af deres fysiske eller psykiske tilstande og ikke på grund af økonomi.

Adam:

Nej klart,

Cecile:

Den artikkel du har fået er på plejehjemskontorer

Adam:

Yes, og tak for den! den var meget spændende

Cecilie:

Velbekommen

Adam:

Pudsigt nok ligger det sig også ret meget op af det vi havde tænkt faktisk, men det kommer jeg ind på om lidt

Cecilie:

Spændende

Adam:

Det måke et lidt stort spørgsmål men hvilke udfordringer oplever du med ældre borgere I forbindelse med tandpleje

Cecile:

Klart personalet, altså på deres plejehjem. Hvis de bor på et plejehjem, så er der ikke fokus på tandplejen og der er der ikke fordi at personalet ikke er uddannet nok, og der er der ikke fordi at det er meget sådan noget med at man spørger de gamle om de har børstet tænder, så siger de gamle ja og så tager man det for gode vare.. Og altså 70% har faktisk behov for hjælp men det kun faktisk 5 eller 10% der faktisk får den hjælp de skal have – det har vi lavet nogle studier på tidligere. Så det er klart den største udfordring, der bliver ikke givet den fornødne hjælp.

Adam:

I forbinde med udfordringen der, har I på nuværende tidspunkt noget udstyr som kan afhjælpe det problem?

Cecilie(5min40sek):

Vi har ikke noget udstyr nej, det har vi ikke, kun det jeg har opfundet, som jo ikke er permanent, det blev taget ned, de har kun en tandbørste. Så vi har ikke noget der kan afhjælpe det, for I virkeligheden er det jo en påmindelse. Men det har vi så, altså vi har klinikassistenter der går rundt I et program der hedder “shared oral care” det kan du også google lidt om hvis det er – der er vores programleder johan der står for det, som går ud og screener og laver nogle undersøgelser, jeg kan ikke huske hvor tit de kommer, ca en gang om måneden på hvert plejehjem. De laver nogle registreringer som kommer ind I deres plejejournal, og så kan de ligeosm se at deres tandhygiejne ikke er god, men det batter ikke rigtigt nok, altså det rykker ikke vi ved ligesom at det status que ik. Men det vel det udsyr vi har .. klinikassistenter, hvis man må kalde dem for udstyr

Adam:

Hvis man skulle lave en ting eller et device som skulle kunne hjælpe på problemet, er der så nogle spicifikke features som du mener skal med?

Cecilie:

Øhm neej altså ikke ud over det beskevet I artiklen , og vi kunne også se at da vi prøvede det at der er hawthorne-effekt, altså det virker når vi er der, når vi så ikke er der så virker det ikke, så det ville jo være de ting man skulle have fokus på ik. Jeg tror at det ville være et godt sted at starte simpelthen. Måske kunne det være at komme direkte ind I deres journal, det har vi ikke kunne fordi københavn kommune har blokering på noget it-sikkerhed,, altså vi kan godt for ham der laver det kan godt gøre det men vi kan ikke få tilladelse til at gøre det,

Adam:

Nej selvfølgeligt,

Cecile:

Simpelthen så det kommer direkte ind I plejejournalen. De går ind hver morgen på noget der hedder fura (kunne ikke helt høre?) og så kigger de ligesom hvad er dagens opgaver, og der ville det være rigtigt fint hvis den også lå der som en aktiv ting at man skulle gå ind og tjekke den af.

Men altså generelt jeg underviser også tandplejen I innovation hvor de også skal tænke I alternative løsninger, og generelt ved vi bare at altså, de kan godt være at de så vinger det af men vi skal bruge tiden på borgere og ikke gå og kigge på en tablet, så derfor lavede vi det device med den lille lampe der, der kunne lyse, det er ikke beskrevet særligt meget I artiklen, men det brugte de rigtigt meget ovre I Billund. De havde også et projekt lignende derovre hvor der så var sådan en lille rød lampe der lyste hvis de ikke var gjort, og så en grøn lampe der lyste hvis tandbørstningen var gjort. - meget simpelt.

Men altså man bliver jo blind, den lyser rødt og ja hvad så, det er virkelig svært det her, der er at flytte bjerge.

Adam:

Nu kommer vi så til vores ide, og den ligger sig faktisk ret meget op af jeres ide, vi havde forestillet os at lave en holder til en elektrisk tandbørste, med en lyssensor I, som detektere om tandbørsten bliver fjernet eller ej, hvis tandbørsten er blevet fjernet er der sandsynligvis foretaget en tandbørstning og så også med en LED-lampe som lyser grønt hvis der er tandbørsten har været fjernet et specifikt stykke tid, og lyser rød hvis den ikke har været brugt, samt led lyset kan hjælpe til at vise hvor længe man skal børste tænderne med en nedtælling.

Så havde vi også tænkt os at sætte et webcam på for at dokumentere forløbet og tjekke om tandbørsten bare blev fjernet og lagt på bordet eller om den faktisk blev brugt

Cecilie:

Ja for det kan godt sku, særligt med de demente , vi havde nogle udfordringer med nogle der havde slukket dem og ja så troede de de bare skulle tænde dem uden at putte dem ind I munden og sådan noget ik. Altså den største udfordring med den gruppe er at de ikke er vant til elektriske tandbørster, jeg tror det bliver nemmer med de ældre om 10 år, da de nok er lidt mere vant til elektriske tandbørster

Adam:

Er der nogle af de ting vi havde tænkt os at prøve at arbejde med du synes er mærkelige eller dumme?

Cecilie:

Egentligt det med at den bliver fjernet og at de kun er det den registrere, det smarte ville være hvis den registrerede om den havde været utsat for fugt, eller tryk måske

Adam:

Aha det er smart, det må vi lige se på

Cecilie:

Det ved jeg jo at oral-b kan, men de vil jo ikke fortælle os hvordan.. så en af vores it-medarbejdere hackede det jo bare, kan man sige ik, og jeg er helt sikker på jeg godt kan sætte dig I forbindelse med ham hvis du gerne vil vide noget mere om it-delen, for det kan han helt sikkert fortælle en masse om

Adam:

Det ku muligvis godt være spændende, men det kan også være det bliver lidt for dybt for os

Cecilie:

Han er virkelig passioneret omkring det, men hans siger at de kan vi sagtens, de vi gjøre var sådan set bare at hacke den der bluetooth del og så I stedet for at den sendte dataen til et andet sted så sendte den dataen til sådan en lille boks. Der er også noget GDPR I forhold til hvis I vil sætte et webcam op ik

Adam:

Ja det kunne jeg godt forestille mig, det måske heller ikke alle der er lige begejstrede for at have et kamera forbundet til internettet på sit badeværelse men altså..

Cecilie:

Men så kunne man måske have en eller anden sensor, måske en bevægelses sensor, som registrerer om folk er dehydrerede eller registrer om folk er faldet - så behøver det måske ikke at være et kamera men måske en eller anden form for.. Man kunne måske have noget der lyttede om tandborsten var tændt på lyden måske

Adam:

Det var rigtigt godt, det var gode indputs, tusinde tak for at have tid til det her

Cecile:

Jammen det var da så lidt, og skriv endelig hvis der er noget , og ellers så er det min kollega , for han ved alt der er værd at vide om it. Du har min nummer og min email. Held og lykke med det,

Adam:

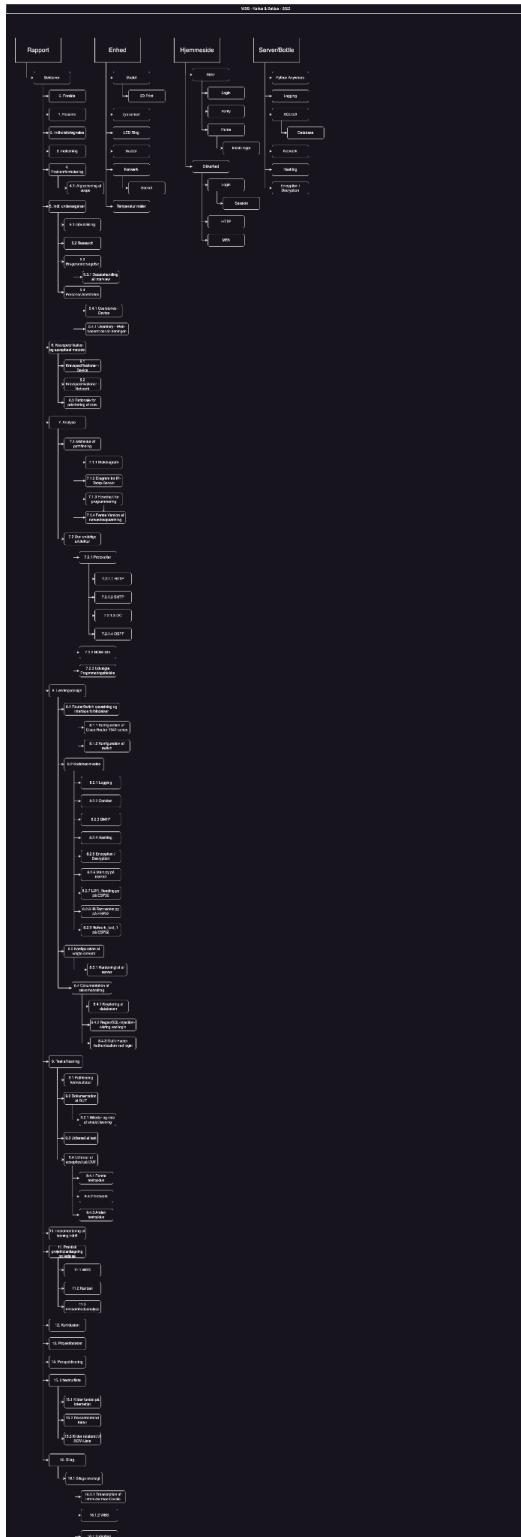
Tak og god weekend

Cecilie:

I lige måde, hej.

16.1.2 WBS

(Kig i projektmappen for høj-kvalitets billede og fil)



16.1.3 Kanban

