# Kravspecifikation **for IoT-baseret** **solcelleoptimeringssystem**

## Indledning:

Formålet med dette projekt er at udvikle og implementere et IoT-baseret system, der kan overvåge vejrforhold og automatisk justere solcellepanelernes orientering i realtid for at maksimere energiproduktionen. Ved at bruge en agil udviklingsmetode, vil systemet blive udviklet iterativt, hvilket betyder, at vi kontinuerligt evaluerer og justerer funktionaliteten gennem sprints, baseret på brugerfeedback og skiftende krav.

## Funktionskrav (Use Cases):

**Use case 1 - Bevægelse**

| **Navn:** | **Bevægelse** |
| --- | --- |
| **Mål** | At justere solpanelernes orientering dynamisk for at optimere eksponeringen mod solen og maksimere energioptagelsen. |
| **Initering** | Brugeren starter systemet og processen modtager data fra lyssensorerne eller efter en fastlagt tidsperiode (f.eks. hvert 10. sekund). |
| **Aktør** | 1. Bruger (Primær) |
| **Antal samtidige forekomst** | 1 |
| **Prækondition** | Systemet er aktiv og alle komponenter virker. . |
| **Postkondition** | Solpanelerne er justeret til den mest optimale vinkel mod solen, baseret på lysintensiteter. Og er i standby. |
| **Hovedscenarie** | 1. Lyssensorer registrerer lysintensitet i fire retninger: Venstre, Højre, Op, Ned. 2. ESP32 videresender disse data via UART til Raspberry Pi. 3. Systemet identificerer retningen med den højeste lysintensitet. 4. Raspberry Pi læser data fra den serielle port 5. Raspberry Pi bruger PID-controlleren til at beregne, hvor meget motorerne skal justere:  * Stepmotoren justeres horisontalt (Venstre eller Højre). * Servomotoren justeres vertikalt (Op eller Ned).  1. Raspberry Pi sender kontrolsignaler til motorerne via GPIO.   EXT1:   1. Stepmotoren drejer det beregnede antal trin mod venstre eller højre. 2. Servomotoren bevæger sig til den beregnede vinkel op eller ned. 3. Raspberry Pi overvåger motorernes position og sikrer, at solpanelerne når den ønskede vinkel. 4. Hvis positionen er korrekt justeret, logger systemet resultatet. 5. Systemet går i standby og venter på næste dataindsamling (hver 10 sekunder eller en tilsvarende periode). |
| **Udvidelser(undtagelser)** | EXT 1   1. Motoren er allerede indenfor en acceptabel fejlmargin i forhold til den beregnede optimale vinkel 2. Systemet springer motoraktiveringen over og går direkte i standby. |

**Use case 2 - Dataindsamling**

| **Navn:** | **Dataindsamling** |
| --- | --- |
| **Mål** | At måle temperatur, luftfugtighed og lysintensitet, behandle data, sende data til webklienten og præsentere dem lokalt. |
| **Initiering** | Systemet starter processen ved at indsamle data fra HTU21-sensoren og lyssensoren. |
| **Aktør** | 1. HTTP Server (Sekundær) 2. Webklient (Sekundær) |
| **Antal samtidige forkomster** | Ingen |
| **Prækondition** | * Systemet er aktivt og fungerer korrekt. * HTU21-sensoren og lyssensoren virker. |
| **Postkondition** | Data er sendt til webklienten, og dataene er vist lokalt. |
| **Hovedscenarie** | 1. ESP anmoder om data fra HTU-sensorer. 2. Lyssensorernes data aflæses via analoge input (ADC). 3. ESP kontrollerer/validerer at data er inden for gyldige intervaller.   EXT 1   1. Rådata konverteres til det rigtige format. 2. Behandlede data præsenteres lokalt via TFT-skærmen. 3. Behandlede data sendes til serveren 4. HTTP serveren gemmer midlertidigt dataen 5. Serveren sender data til webklienten via HTTP POST-anmodning. 6. Systemet opdaterer data kontinuerligt (fx hvert minut eller efter brugerens præference). |
| **Udvidelser(undtagelser)** | EXT 1   1. Hvis ingen ændringer i data opdages 2. Spring ESP opdatering af displayet og HTTP-overførsel over. |

**Use case 3 - Real-Time Monitoring**

| **Navn:** | **Real-Time Monitoring Dashboard** |
| --- | --- |
| **Mål** | At give brugerne mulighed for at overvåge solpanelets præstation og miljødata i realtid via en webbaseret webclient |
| **Initering** | Brugeren åbner webclient via en webbrowser (ved hjælp af ESP's IP-adresse) eller aktiverer det lokale display. |
| **Aktør** | 1. Bruger (Primær) 2. Webklient (Sekundær) 3. HTTP-server (Sekundær) |
| **Antal samtidige forkomster** | 1 |
| **Prækondition** | * Systemet er tændt, og ESP er korrekt forbundet til sensorerne * Webclient er konfigureret og tilgængeligt via browseren. |
| **Postkondition** | Brugeren kan se modtaget opdateret information om temperatur, luftfugtighed, og solpanelernes energiproduktion. |
| **Hovedscenarie** | 1. Brugeren tilgår webklienten via en browser. 2. Webclient sender en HTTP-forespørgsel til serveren for at hente seneste data. 3. Serveren svarer med sensorernes data i JSON-format. 4. Webclient validerer og indlæser data. 5. Webclient genererer grafer ud fra dataene ved hjælp af grafbiblioteker som Chart.js eller D3.js. 6. Webclient viser graferne og dataen i et tabelformat med kolonner for:  * Temperatur * Luftfugtighed * Lysintensitet * Strøm  1. Brugeren har nu mulighed for at interagere med graferne. 2. Webclient opdaterer automatisk dataene med faste intervaller (fx hvert minut). |
| **Udvidelser(undtagelser)** |  |

**Use case 4 - Strømgenering**

| **Navn:** | **Strømgenering** |
| --- | --- |
| **Mål** | At generere strøm ved hjælp af solpaneler, og indikerer at strøm kører |
| **Initering** | Strømgenereringen starter automatisk, når sollyset registreres af sensorer, eller systemet aktiveres manuelt af brugeren. |
| **Aktør** | 1. Bruger (Primær) |
| **Antal samtide forkomster** | 1 |
| **Prækondition** | Alle vigtige komponenter er forbundet og virker. |
| **Postkondition** | Systemet kan producere strøm, og data overføres og vises på ESP32-displayet. |
| **Hovedscenarie** | 1. Solpanelerne begynder at generere elektricitet baseret på solens energi. 2. Elektricitet føres igennem strømsensoren via solcelle, 3. Strømsensoren konverterer målinger til et signal. 4. Strømsensoren sender signalet til ESP 5. ESP32 behandler data til de formaterede data. 6. ESP32 modtager strømdata og viser dem på et display som indikatorer (f.eks. energiproduktion og systemstatus). 7. Den genererede elektricitet overføres til en direkte tilsluttede apparater. 8. Systemet opdaterer statuslog og fortsætter overvågning og justering for kontinuerlig energiproduktion. |
| **Udvidelser(undtagelser)** |  |