

TEK301 – Eksamen Dokumentasjon

Jan-Philippe Emmanuel Rasay Guerrero

Contents

Idè.....	1
Løsning.....	2
Oppsett.....	3
Particle photon.....	3
Prototype.....	4
Webhooks.....	4
Biblioteker	6
Backend/Firebase	6
Frontend	7
Sikkerhet.....	8
Videreutvikling	8
Kilder.....	9
Datasheets.....	9
Eksterne biblioteker	9
Vedlegg/Link.....	9

Vi har fått oppgave i å lage en IoT prototype innen sikkerhetsbransjen. Det er åpen oppgave, og vi får velge hva vi skal lage men må være innen sikkerhetsbransjen. Og inneholde disse 3 delene:

- DEVICE: Et embedded system med sensor/aktuator eller lignende
- BACKEND: Du må bruke en løsning som tar vare på dataene fra DEVICE
- FRONTEND: En klient (web/mobil/m.m) som viser fram dataene på en nyttig måte

Idè

Per dags dato er det slik at boligeiere eller hytteiere som leier ut boligene sine til andre må møte leietakerne for å levere nøkkene til boligen de leier ut. Det finnes en løsning som heter ShareBox hvor utleier kan putte nøklene i en box, og leietakeren får passord til den

boksen og få tilgang til nøklene. Disse boksene er som oftest plassert i butikker som er åpne 24/7, som for eksempel 7 eleven eller bensinstasjoner. Etter at leietakeren er ferdig med leieperioden, så leverer personen tilbake nøkkelen i samme boks. Utleier og leietaker trenger da aldri å møtes ansikti til ansikt for å utveksle nøkler. Problemet med dette er at disse boksene ikke finnes overalt, og det kan hende at det er langt fra shareboxen til boligen som leies ut. Dette er ugunstig for både utleiere og leietakere. Et annet problem er at utleierne kan risikere å miste nøklene ved at leietakerne enten mister nøklene eller velger å ikke levere de tilbake.

ShareBoxen fungerer da som et mellomledd for utleiere og leietakere. Men både utleier og leietakere må innom shareboxen for å hente og levere nøklene. Dette kan være et problem for hytteiere som leier ut hyttene sine. For det kan hende at de ikke er installert sharebox i nærheten av hyttene som skal leies ut. Som ender opp med at begge parter må møtes, og om utleier og leietaker bor på hver sin side av landet, og boligen/hytten som skal leies ut ligger i midten, da er det stress for begge parter.

Et annet problem er om man har en bolig i utlandet som du bruker som AirBnb, da må du igjen ha et mellomledd som gir utleieren nøkkelen. Siden sharebox ikke er installert over hele verden.

Min ide går ut på å ta bort det mellomleddet, og retter seg mot bolig og/eller hytteiere som leier ut i korte perioder. Ideen er å lage en smartlås som skal kunne styres via nett. Hvor man kan bytte kode på låsen hvor som helst i verden så lenge man er koblet til nett. Smartlåsen skal også registrere hver gang noen bruker låsen.

Løsning

Til prototypen min har jeg brukt en particle photon som mikrokontroller, dette er fordi den har innebygd wifi chip. Noe som jeg er avhengig av for at prototypen skal fungere som den skal. Jeg har koblet til en LCD skjerm og en keypad. Keypaden for å ta imot user input (Lås kode), og LCD for å vise litt informasjon. For å etterligne en lås som låses opp når man taster inn riktig pin, så har jeg brukt en servo.

Jeg har brukt Firebase til å lagre dataene mine I firebase databasen lagres både passord og nåværende utleier. Det blir også sendt data til firebasen fra photonen. Dataene som blir lagret er navnet på utleieren, koden som blir brukt til å låse opp og timestamp på når låsen blir låst opp.

Particle photonen henter passordet og utleieren som er lagret i databasen. Og det er dette passordet som blir brukt for å åpne låsen. For å hente og sende data fra photonen til firebase har jeg brukt webhooks

For å vise dataene som blir lagret i firebase, har jeg lagd en enkel nettside ved hjelp av html/css og javascript. Via nettsiden kan man også forandre på navnet på leietakeren og passordet.

Oppsett

Particle photon

For å sette opp photonen har jeg brukt particle applikasjonen.

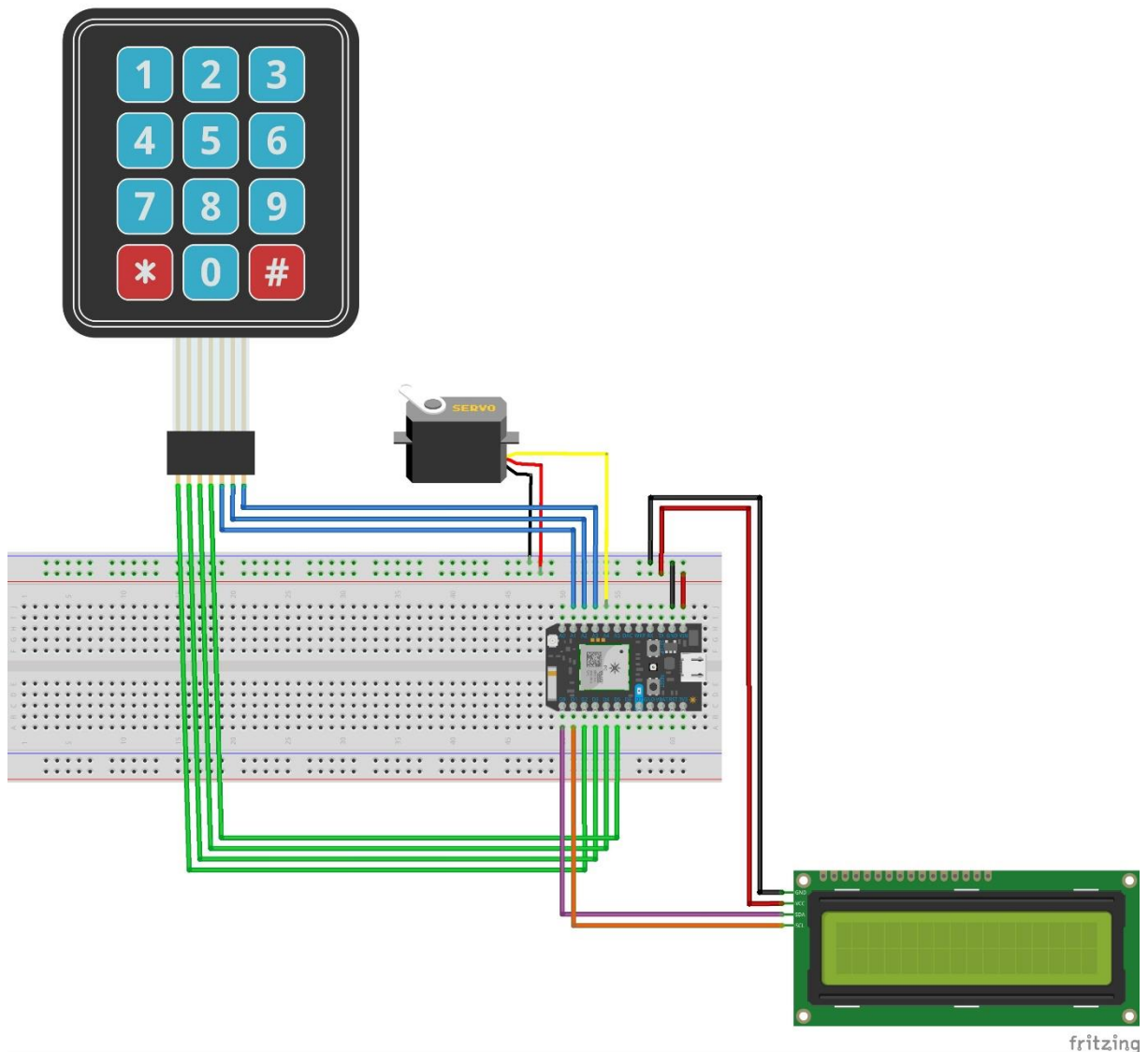
1. Lag en bruker på <https://www.particle.io/>
2. Last ned particle applikasjonen på telefonen
3. Hold inne setup knappen i 3 sek på photonen, til photonen begynner å blinke blått
4. Følg instruksjonene på applikasjonen

<https://docs.particle.io/guide/getting-started/start/photon/#step-1-power-on-your-device>

Etter man har satt opp particle photonen, så kan man flashe koden til prototypen min til photonen

https://go.particle.io/shared_apps/5a24291d3f643a7349000d7f (Kildekode)

Prototype



Webhooks

Jeg har brukt tre webhooks for å hente og sende data fra firebase

1. Gå inn på <https://console.particle.io/devices>
2. Trykk på integrations
3. Trykk på «New integration»
4. Trykk webhooks
5. Trykk på custom template
6. Lim inn:

For å hente password

```
{  
  "event": "passNow",
```

```
"url": "https://tek301-c67c0.firebaseio.com/Password/PassNow.json",
"requestType": "GET",
"noDefaults": true,
"rejectUnauthorized": true,
"responseTemplate": "",
"query": {
  "auth": "SHaAN0WxC8k0uAuo4LpCxxahigiwV1kcLH6aSJji"
}
}
```

Lag en ny webhook for å hente customer/utleier og lim inn

```
{
  "event": "customer",
  "url": "https://tek301-c67c0.firebaseio.com/Password/Customer.json",
  "requestType": "GET",
  "noDefaults": true,
  "rejectUnauthorized": true,
  "query": {
    "auth": "SHaAN0WxC8k0uAuo4LpCxxahigiwV1kcLH6aSJji"
  }
}
```

Lag en ny webhook for å sende data og lim inn:

```
{
  "event": "sendData",
  "url": "https://tek301-c67c0.firebaseio.com/Data.json/",
  "requestType": "POST",
  "noDefaults": true,
  "rejectUnauthorized": false,
  "json": {
    "customer": "{{customer}}",
  }
}
```

```
"pin": "{{pin}}",  
"ts": "{{PARTICLE_PUBLISHED_AT}}"  
},  
"query": {  
  "auth": "SHaAN0WxC8k0uAuo4LpCxxahigiwV1kcLH6aSjji"  
}  
}
```

<https://docs.particle.io/guide/tools-and-features/webhooks/>

For å koble photonen til firebase har jeg fulgt denne guiden:

https://github.com/rickkas7/firebase_tutorial

Biblioteker

Jeg har brukt to ekstra biblioteker for prototypen min. «Keypad_Particle» og «LiquidCrystal_I2C_Spark». For å legge disse til koden, kan man gjøre ved å gå inn på «libraries» på particle ide'en og søke på disse to bibliotekene og deretter legge disse til i prosjektet.

Backend/Firebase

Jeg har valgt å bruke firebase på grunn av firebase har en realtime database. Noe som gjør at det blir synkronisert med alle clientene med engang. Så om noe blir forandret i databasen, så blir den synkronisert med photonen med engang. Det er gunstig siden jeg lagrer passord i databasen, og photonen trenger den dataen med engang. Og passordet kommer til å bli forandret ganske ofte, så da er det greit at den synkroniseres med photonen.

For å sette opp firebase, så trenger man en google konto. Når man har lagd seg en bruker og logget inn, så kan man følge instruksene på firebase docs for å sette opp en database og koble den til.

For å se på firebase databasen min så kan man logge inn med denne brukeren:

Brukernavn: tek301.eksamen@gmail.com

Passord: viewertek301

Etter at man har logget inn, så trykker du på «GO TO CONSOLE» øverst i høyre hjørre. Dertetter velger du prosjektet «TEK301». Så kan du trykke på «Database» på sidemenyen for å se databasen og dataene som er lagret.

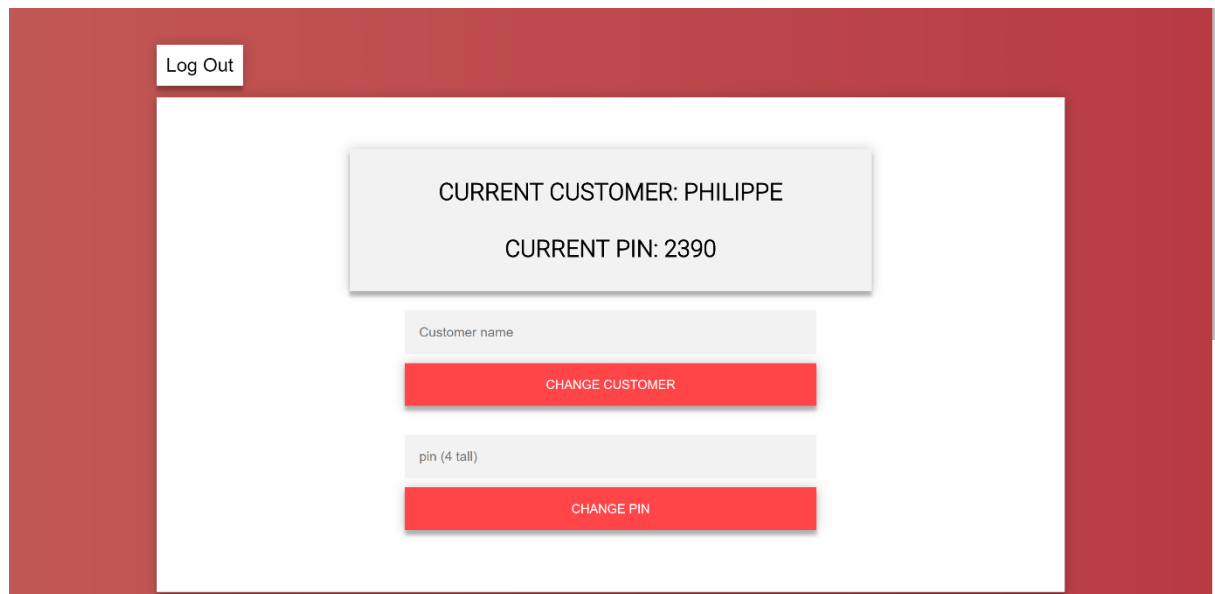
Frontend

Jeg har lagd en enkel nettside ved hjelp av html/css og javascript. Som henter og endrer data i firebase. Nettsiden viser i realtime hver gang noen låser opp låsen. . Hvem som har låst opp og klokkelsett når det ble låst opp. Her kan man også forandre på hvem som er leietakeren og pinkoden til låsen. For å komme seg inn på siden, så trenger man å logge inn med en bruker. Jeg har lagd en testbruker. Bruker firebase sin authentication for å logge inn.

Link: http://tek.westerdals.no/~guejan15/TEK301_Front/

Brukernavn: admin@admin.com

Passord: testadmin



The screenshot shows a web application interface with a dark red header and a white main content area. In the top left of the header, there is a 'Log Out' button. The main content area displays the current customer and pin information, and options to change them.

Log Out

CURRENT CUSTOMER: PHILIPPE

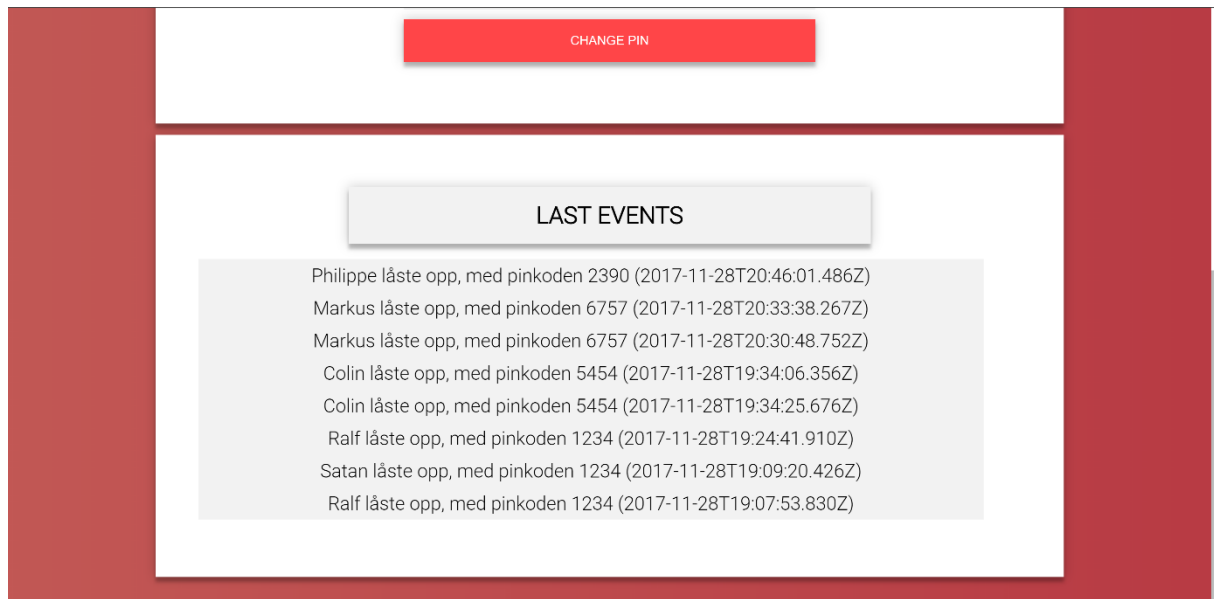
CURRENT PIN: 2390

Customer name

CHANGE CUSTOMER

pin (4 tall)

CHANGE PIN



Sikkerhet

Siden det kun er en prototype så den ikke så rustet mot sikkerhet enda. Men de tiltakene jeg har gjort er å sette passord på nettsiden hvor man kan forandre pinkoden til låsen. Noen av svakhetene er at photonen er avhengig av internett for å oppdatere pinkoden til låsen, det jeg har tenkt er at man om photonen ikke er koblet til. Så skal passordet til låsen være den sist lagrede passordet den hadde.

Pin koden på låsen er satt til 4 siffer, men kan forandres i koden.

Jeg har også satt opp prototypen slik at låsen låses automatisk etter noen sekunder den har blitt låst opp.

Man skal kunne låse opp låsen med vanlig nøkkel også, i tilfelle det blir strømbrudd. En annen sikkerhetstiltak som kunden kan gjøre er å sette opp flere nettverks SSID'er, så en som er spesielt for photonen. Slik at om man blir hacket, så har ikke hackeren tilgang til alle enhetene.

Det er selvfølgelig viktig at man har et godt passord på routeren. At man bruker WPA2 encryption istedenfor WEP.

Eventuelt liste alle MAC adresser som kobler seg til nettet og blokkerer alle ukjente fra å koble seg til nettet.

Videreutvikling

Til videreutvikling tenker jeg at nettsiden skal forbedres, evt lage en native applikasjon. Slik at kundene kan styre låsen gjennom en applikasjon.

Jeg har tenkt at det skal legges til flere funksjoner til låsen. Som at utleieren skal kunne låse opp låsen via nettsiden eller applikasjonen. En annen funksjon som skal legges til er at utleieren skal få en notification dersom det blir tastet feil kode x antall ganger.

Kilder

Slides fra forelesningene i TEK301

<https://docs.particle.io/guide/getting-started/intro/photon/>

<https://firebase.google.com/docs/>

https://github.com/rickkas7/firebase_tutorial (Koble photonen til firebase)

<https://codepen.io/colorlib/pen/rxddKy> (Login template)

<https://community.particle.io/t/simple-keypad-tutorial/8758>

<https://www.bensound.com/royalty-free-music/track/funky-element> (Bakgrunns musikk til video)

Datasheets

<https://www.parallax.com/sites/default/files/downloads/27899-4x4-Matrix-Membrane-Keypad-v1.2.pdf> (Keypad)

<https://opencircuit.nl/ProductInfo/1000061/I2C-LCD-> (LCD)

<http://akizukidenshi.com/download/ds/towerpro/SG90.pdf> (Servo)

Eksterne biblioteker

<LiquidCrystal_I2C_Spark.h>

<Keypad_Particle.h>

Vedlegg/Link

Vedlegg 1: Koblingsskjema

Vedlegg 2: Fritzing

Vedlegg 3: Kildekode for nettsiden

Link til video: <https://www.youtube.com/watch?v=CcTDaEwqpNU>

Link til kildekode: https://go.particle.io/shared_apps/5a2440b83f643ad7e2001276