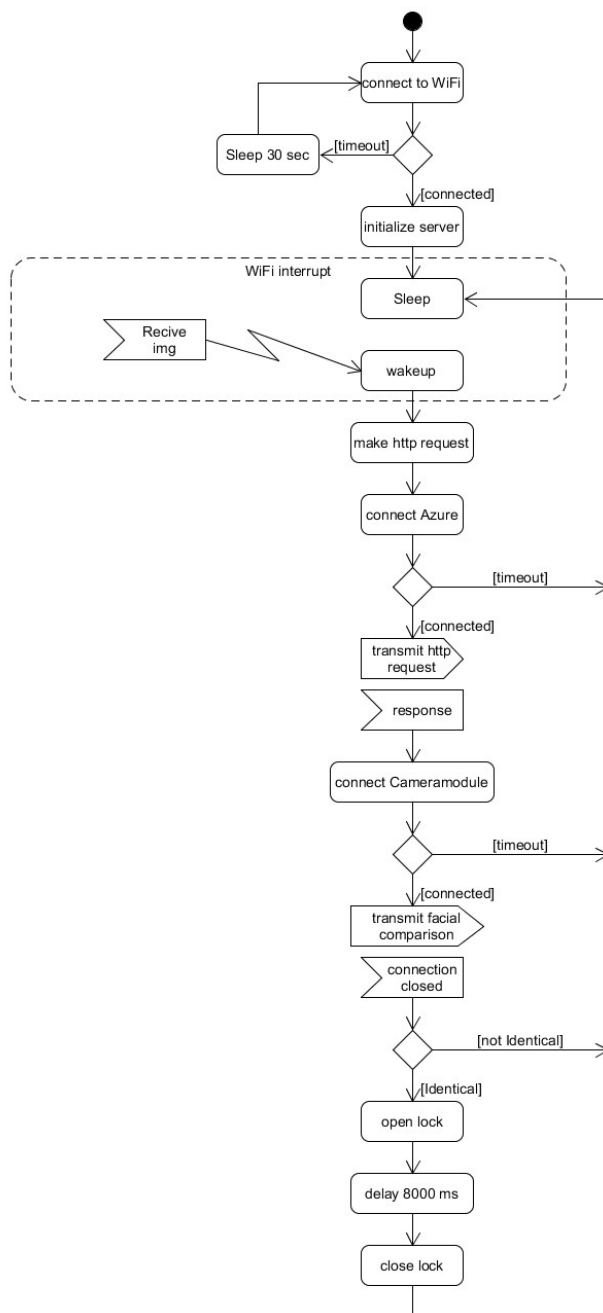


Hovedmodul

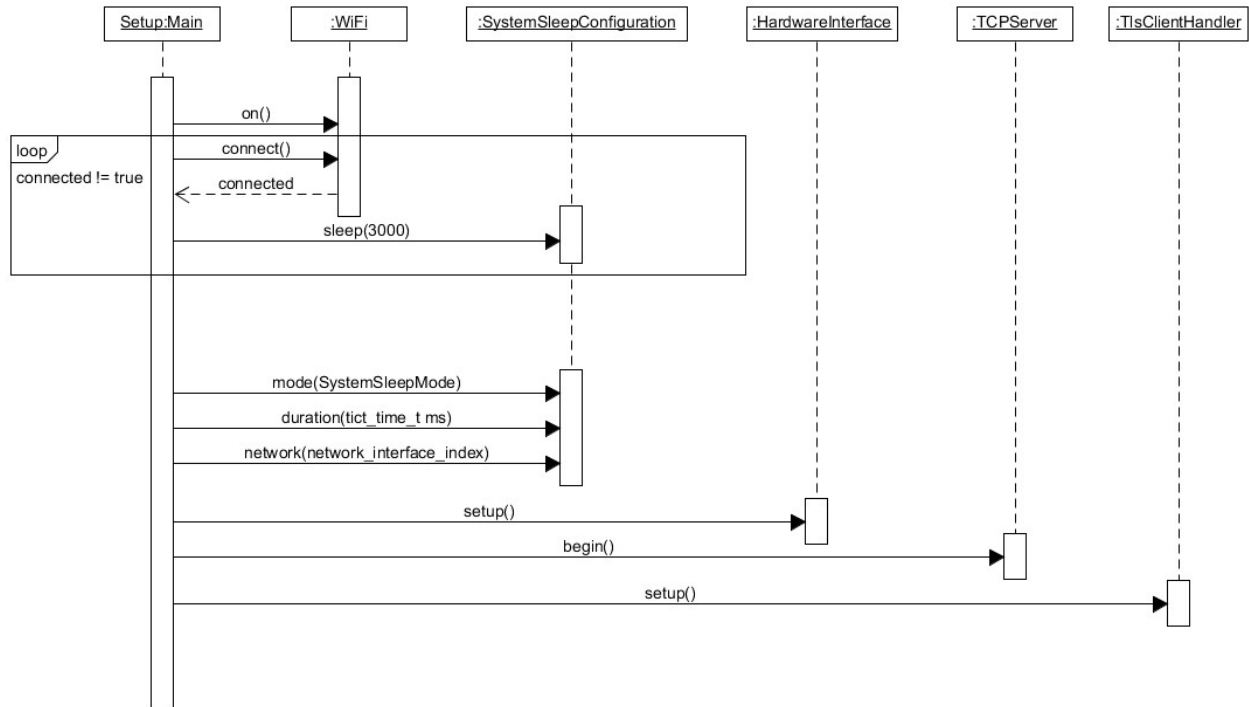
Her findes en mere dybdegående analyse af design af hovedmodelet

Hovedmodulets primære opgaver er at låse op for brugeren, hvis brugeren er i systemet. Derudover fungerer hovedmodulet som gateway imellem kameramodulet og Azure facial recognition service. På Figur 1 kan flowet af systemet ses.



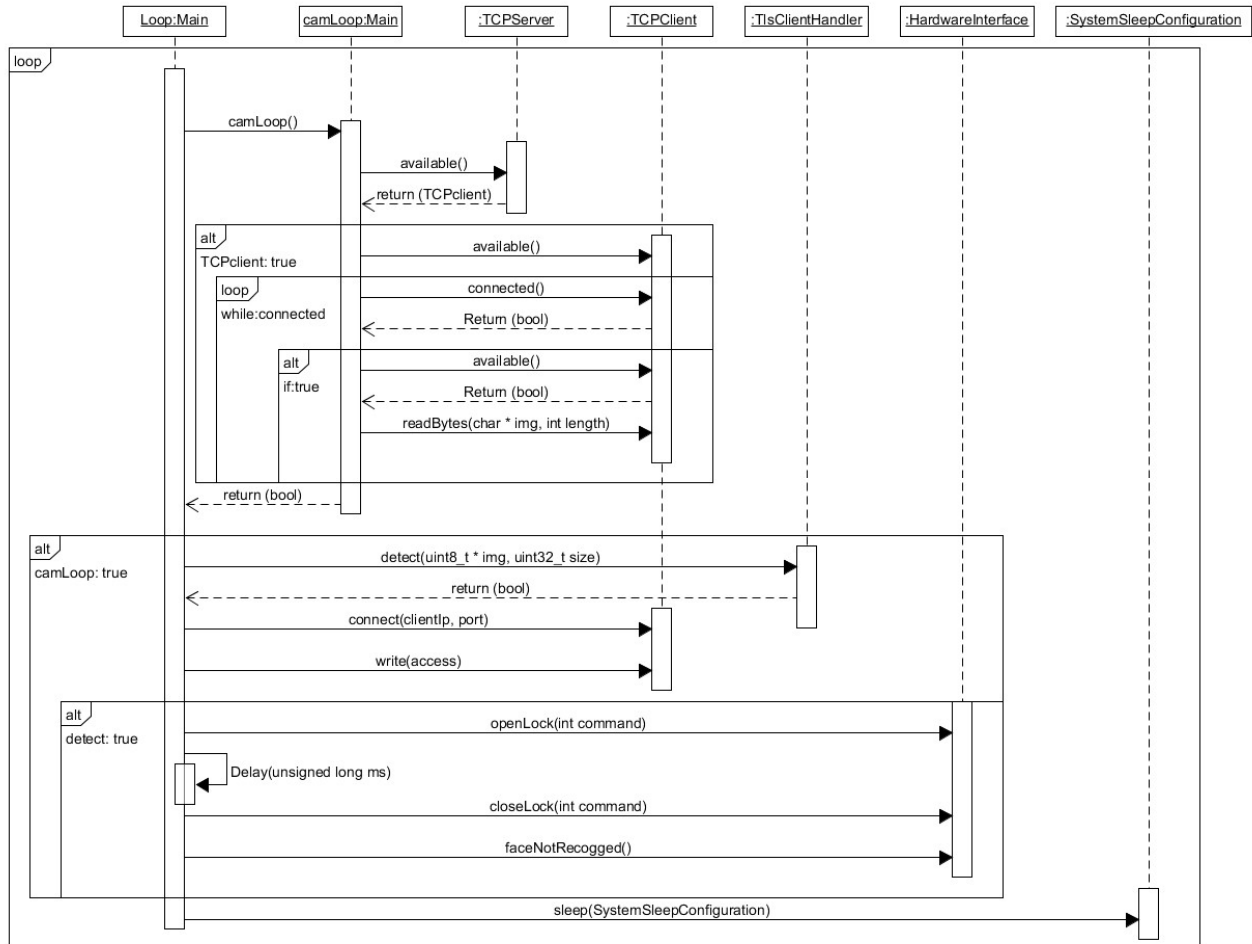
Figur 1 - aktivitetsdiagram over Hovedmodul

Figur 2 viser sekvensen over setup funktionen i arduino sketch filen. Denne funktion kører 1 gang som indledende setup, hvorefter funktionen loop sekventielt eksekveres i et loop. Sekvensdiagrammet for loop funktionen kan ses på Figur 3½



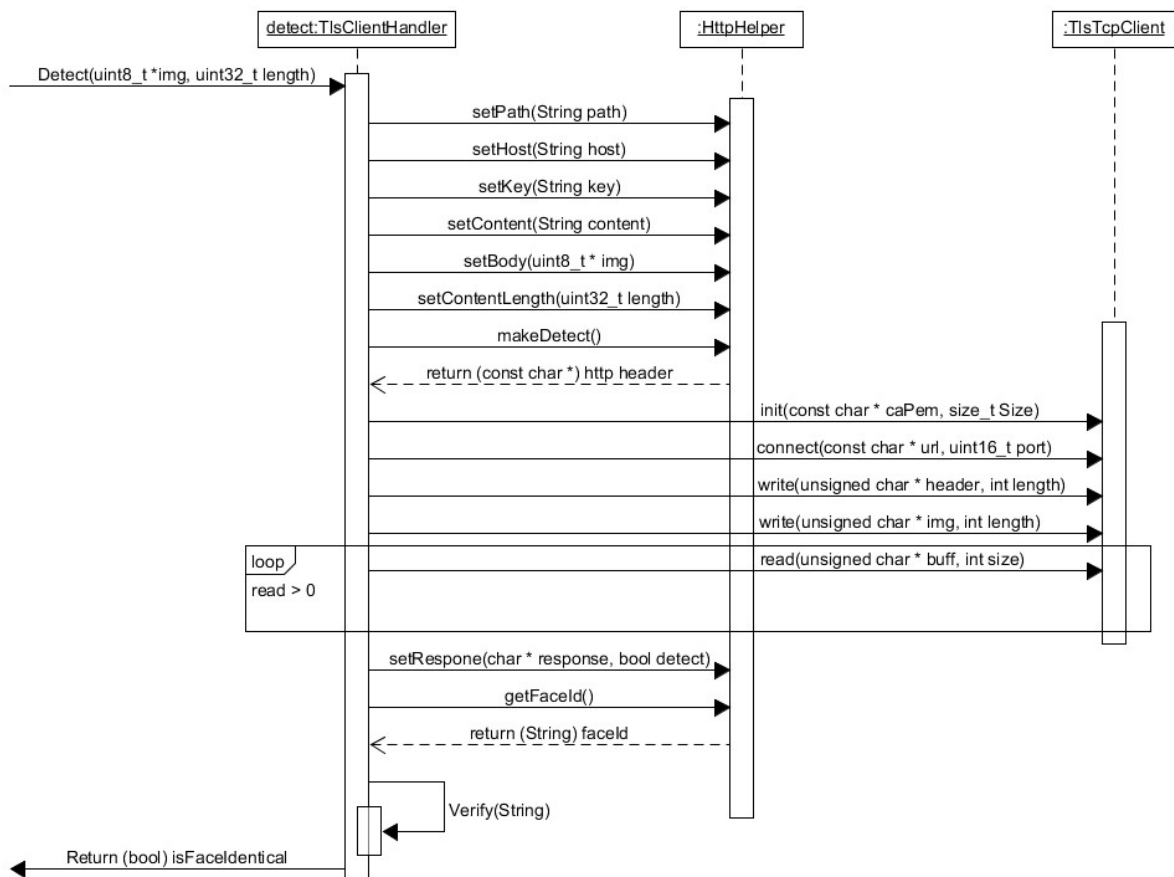
Figur 2 - sekvens diagram over setup i arduino sketch

Loop funktionen står for at styre mainflowet af programmet. Her ventes der på forbindelse TCP serveren der blev opretter i setup loopet. Når kameramodul opretter forbindelse, modtages et billede der sendes over til TlsClientHandler klassen. Afhænging af retur værdien fra kaldet, sendes et svar tilbage til kameramodul, indeholdende om personen blev godkendt eller ej. Blev Personen godkendt vil Servomotoren åbne i 8 sekunder, hvorefter den lukkes og system går i sleep tilstand



Figur 3 - sekvens diagram over loop i arduino sketch

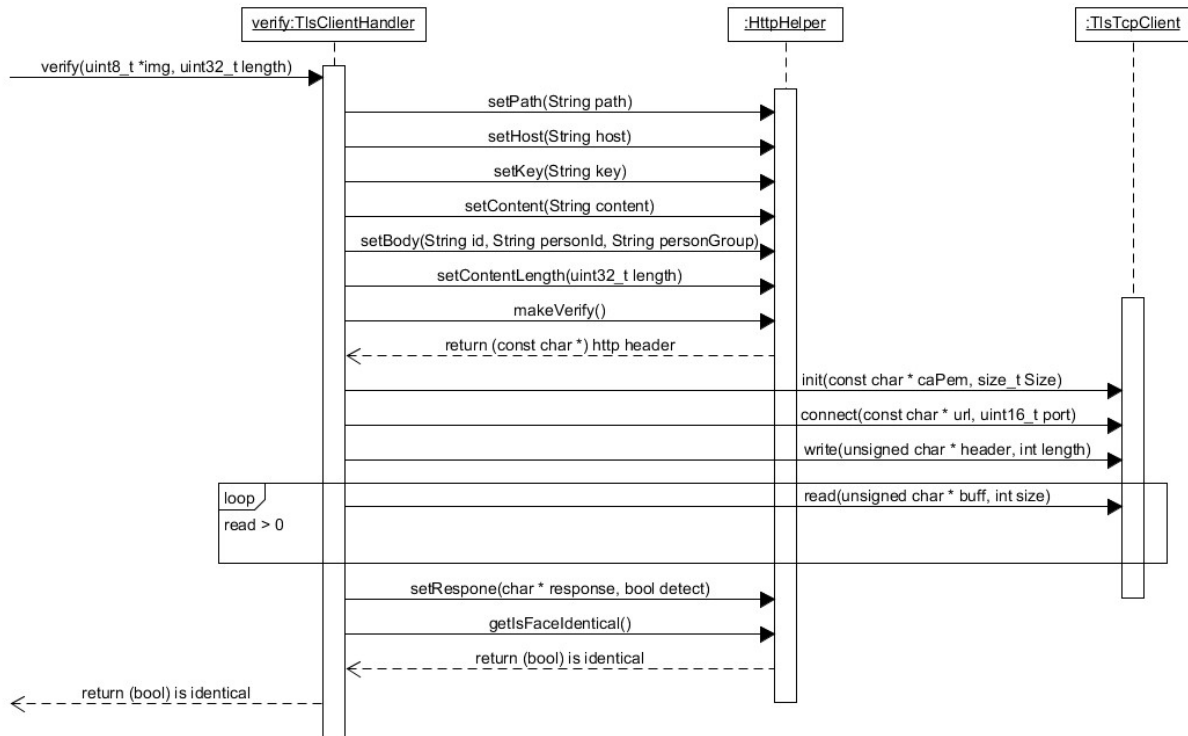
Når detect bliver kaldt skal der forberedes til og sendes data til Azure, hvor der skal laves et http kald. Det er muligt og lave http kald vha. Particles cloud platform, men den understøtter ikke binary octet streams. Derfor er der designet en HttpHelper klasse, der står for at parse http kald og svar. Forbindelsen til Azure oprettes via det eksterne bibliotek TlsTcpClient der er bygget op omkring mbedTLS OG Particles TCPClient. For at samle tls klienten med klassen HttpHelper er der designet en TlsClientHandler, der HttpHelper og kommunikationen imellem Azure.



Figur 4 - sekvens diagram over detect

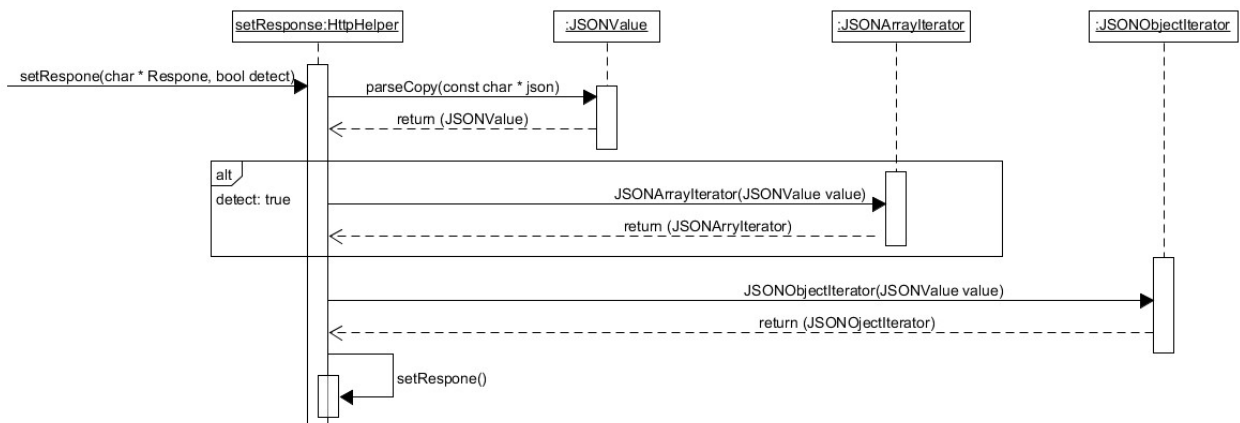
På Figur 4 kan det ses hvordan TlsClientHandler interagerer med de 2 klasser. For at sammenligne et ansigt skal Azure først detektere et ansigt, hvorefter der returneres et id. Dette id kan efterfølgende sendes til sammenligning med en person, allerede gemt i Azures database. Dette gøres med kaldet Verify, som kan ses på Figur 5.

Verify kaldet minder meget om Detect kalder og vil derfor ikke gennemgås.



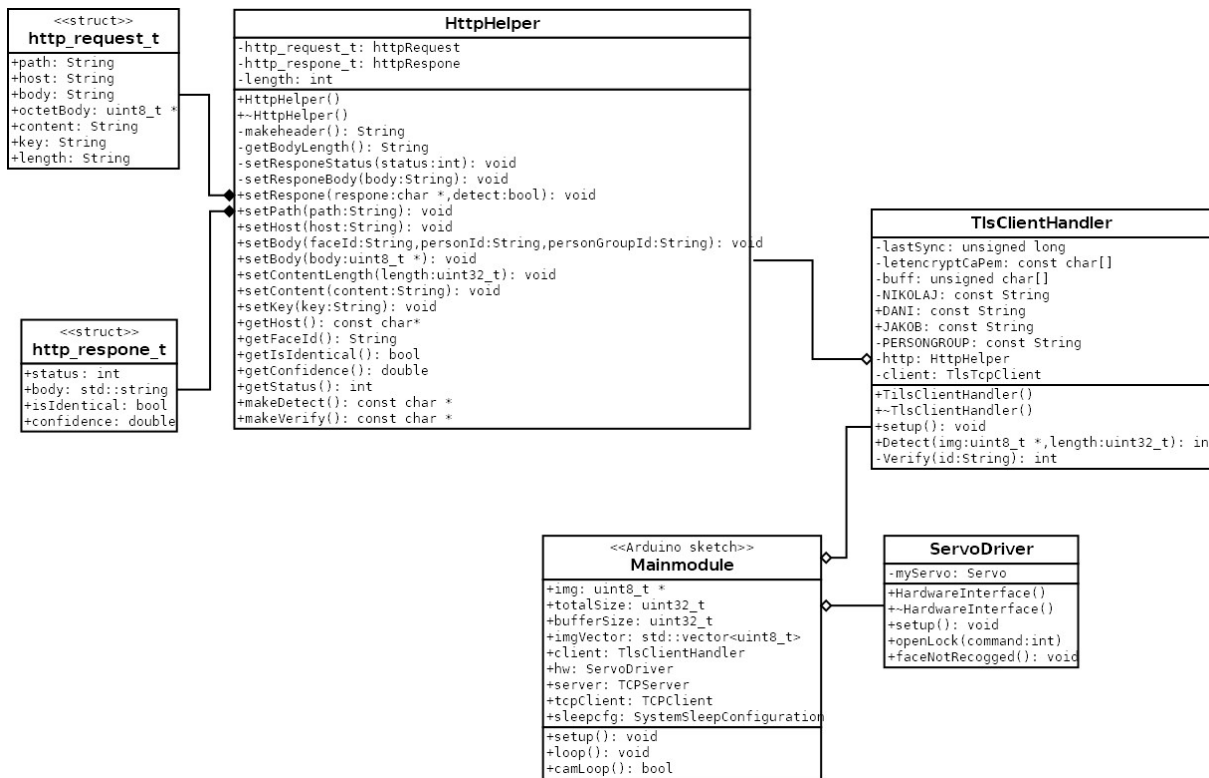
Figur 5 sekvens diagram over verify

Når systemet har modtaget et svar fra Azure er det formateret som et http respons. Dette skal pakkes ud så data fra responset kan bruges. Her bruges setResponse, der parser svaret vha. de Particles JSON klasser.



Figur 6 sekvens diagram over setResposns

Dette giver nedenstående klassesdiagram.



Figur 7 - klassesdiagram