Technische Documentatie

# Website

## Authenticatie

Authenticatie bij ons project werkt aan de hand van Firebase.

Als een gebruiker een nieuw account aanmaakt, zullen de email en het wachtwoord dat meegegeven worden gebruikt worden om een account aan te maken in de database. Het aanmaken van een account zal enkel gebeuren als de gebruiker zowel een correcte email (die nog niet opgeslagen is in de database) en een wachtwoord ingeeft. Er zal bij het aanmaken van het account ook extra informatie nodig zijn die moet ingegeven worden. Zonder de extra info zal het formulier niet verzenden. De info zal niet op dezelfde plaats worden opgeslagen als de email en het wachtwoord. Authenticatie heeft een database die los staat van de normale en geen extra info kan opslaan. Om opslag van aanvullende gegevens mogelijk te maken, moet er een collectie gemaakt worden in de normale database waar documenten in opgeslagen worden die als referentie het UID van de gebruiker hebben. Als een gebruiker succesvol een account aanmaakt zal er ook automatisch ingelogd worden en komt de gebruiker terug op de homepagina van de website uit.

Het aanmaken van een account werkt met onderstaande code. Eerst worden alle error variabelen overschreven met een lege string om te voorkomen dat er errors van vorige pogingen getoond worden. Daarna wordt er gekeken of alle data meegegeven is, als dit zo is wordt er met *createUserWithEmailAndPassword* een gebruiker aangemaakt. Zoals de naam van de functie aanduidt, moet er een email en wachtwoord worden meegegeven. Na het aanmaken van het account zelf zal het document voor de gebruiker worden gemaakt met de informatie die ingegeven is. Als alles succesvol voltooid is, zal er worden geredirect naar de homepagina. Als er ergens iets niet klopt zal er een error worden opgeslagen in de bijhorende variabele en op de website getoond worden.  
A picture of the code that is ran when creating an account.

Description automatically generated

Om in te loggen als er al een account bestaat, volstaat het om enkel de email en het daarbij horende wachtwoord in te geven. Als de gegevens correct zijn en bij een bestaand account horen, zal de inlogpoging slagen en de gebruiker ingelogd zijn. Als dat niet het geval is, zullen er error berichten worden getoond die zullen beschrijven wat er misging. Indien succesvol ingelogd, zal de gebruiker naar de homepagina van de website worden gestuurd.

Het inloggen van een gebruiker werkt met onderstaande code. Met *sigInWithEmailAndPassword* zal er een poging gedaan worden om in te loggen. Deze methode heeft net zoals die voor het aanmaken van een account een email een paswoord nodig. Als het inloggen succesvol is, zal de pagina naar home redirecten.  
Log in function.

Description automatically generated

Gebruikers kunnen uitloggen via de accountpagina. Indien de website gesloten is zonder uit te loggen, zal de gebruiker automatisch ingelogd worden als ze de website terug betreden.

Het uitloggen van een gebruiker werkt met onderstaande code. De in de functie *logOut* zal de functie *signOut* opgeroepen worden en het huidige auth object meegegeven worden. Eenmaal de gebruiker succesvol uitgelogd is, zal de pagina naar home redirecten.  
Log out function.

Description automatically generated

## Routering

Voor navigatie tussen pagina’s wordt er gebruik gemaakt van react-router-dom. Met die library is het mogelijk om aan de hand van URL-routes pagina’s te tonen. Ook de navigatiebalk bovenaan de pagina is gemaakt met behulp van de library.

Routering werkt met onderstaande code. Eerst wordt er een navigatie element gemaakt dat af zal hangen van of de gebruiker ingelogd is of niet. In dit element staan er link elementen om naar andere pagina’s te kunnen gaan. Onder het navigatie element worden er ook routes gedefinieerd die een URL-pad mee krijgen waar een bestand aan verbonden wordt. Dit bestand wordt dan opgeroepen als er naar een URL gesurft wordt.  
Navigation system.

Description automatically generated with medium confidence

## Producten kopen en verkopen

Ingelogde gebruikers kunnen producten kopen en verkopen. Om een product te verkopen moet er een formulier worden ingevuld en een foto worden meegegeven. Als één of meerdere van de velden leeg gelaten zijn of er geen foto wordt toegevoegd, zal er geen nieuw product worden toegevoegd aan de database. Gebruikers krijgen errors te zien die meer uitleg geven over wat er nog anders moet om een product te kunnen toevoegen. Als het formulier wel volledig is, zal er een nieuw product aangemaakt worden in de productencollectie van de database. Na het toevoegen zal de gebruiker een prompt krijgen om terug te keren naar de homepagina.

Het verkopen van een product werkt met onderstaande code. Eerst worden de error variabelen overschreven met een lege string om te voorkomen dat er errors zouden getoond worden die nog van een vorige poging zijn. Daarna wordt er gekeken of alle data meegegeven is. Als dat in orde is, zal er een referentie worden aangemaakt en daarna aan de hand daarvan de foto worden geüpload. Meteen na het uploaden van de foto wordt de download URL ervan opgevraagd, die wordt dan samen met alle andere data opgeslagen in het document van het product. Na het opslagen zal er een modal openen die zegt dat het gelukt is en een knop heeft om terug naar home te redirecten.  
Sell item function.

Description automatically generated  
Graphical user interface, text, application, email

Description automatically generated

Als een gebruiker wil zien welke producten hij of zij momenteel te koop aanbiedt, kunnen ze dit raadplegen op de ‘your items for sale’.

Dit werkt via onderstaande code. Er wordt een query gemaakt waar alle producten worden opgevraagd waarbij het UID van de eigenaar gelijk is aan het UID van de huidig ingelogde gebruiker. Die query wordt uitgevoerd en dan zal de teruggegeven data met behulp van een foreach in een array worden opgeslagen. Die array zal daarna worden opgeslagen in een useState variabele zodat de data kan worden uitgelezen in de html.  
Function to see the products a user is currently selling.

Description automatically generated

Om een product te kopen kan een gebruiker het product aanduiden op de homepagina. Dat zal hun naar de detailpagina van het product brengen. Eenmaal op de detailpagina is er extra info zichtbaar en kan er op kopen gedrukt worden. Als er op de knop wordt gedrukt, moet er eerst nog eens confirmatie worden gegeven dat de gebruiker effectief het product wil kopen. Als de confirmatie gekregen is, wordt het product verkocht en krijgt de gebruiker een code om de locker te openen waar het product in zal zitten. De code zal verstuurd worden naar de locker die het dan zal opslaan voor gebruik.

Bij het kopen van een product wordt onderstaande code gebruikt. Als er op kopen wordt gedrukt zal er eerst worden gekeken of de gebruiker is ingelogd, indien dit niet zo is, zal er naar de login pagina worden geredirect. Als de gebruiker wel ingelogd is, zal er een modal worden geopend om confirmatie te vragen over het wel of niet kopen. Bij het confirmeren zal er dan een code worden gegenereerd en doorgestuurd naar de locker. Na het doorsturen zal het document worden verwijderd. Een goed alternatief voor verwijderen zou werken met een actief/ non-actief systeem zijn waarbij verkochte producten op non-actief worden gezet. Op die manier kunnen zelfs verkochte producten nog nagekeken worden of in statistieken opgenomen worden. Na het kopen zal er een modal worden geopend met informatie en de code om de locker te openen. Functions for buying a product.

Description automatically generated

# Elektronica

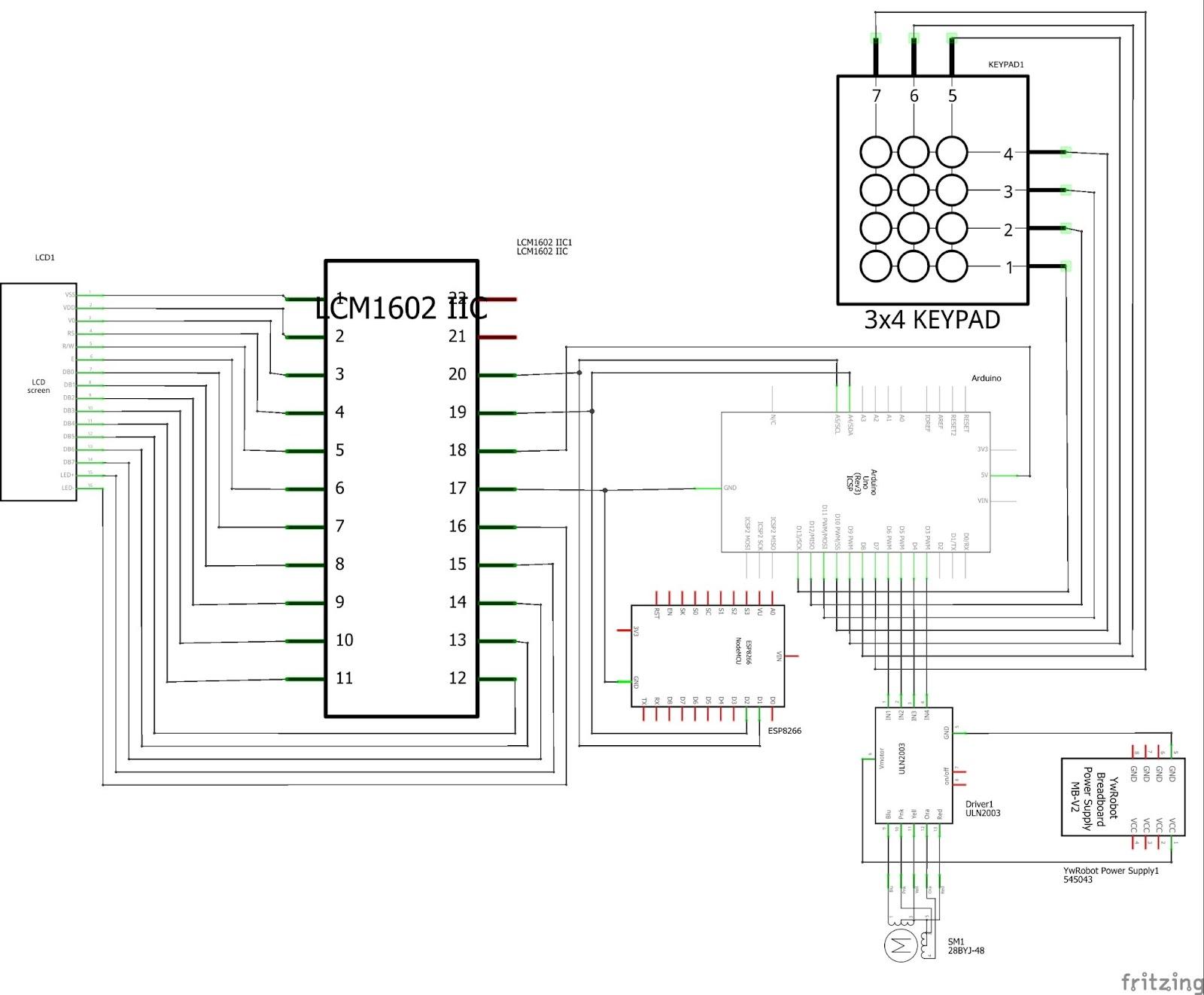
## Hardware

Het project maakt gebruik van een locker om producten op te bergen. Deze locker bevat elektronica om de locker te sluiten/openen en de code te tonen via de lcd. In totaal worden acht verschillende componenten gebruikt.

Voor het project word er gebruik gemaakt van:

* 2 microcontrollers en hun respectievelijke programma’s (Arduino en ESP8266)
* een keypad
* een LCD
* een I²C module (LCM1602)
* een stappenmotor (28BY-48)
* een motordriver (ULN2003)
* een externe voeding

In de onderstaande afbeelding is het circuit setup van het project zichtbaar.



### ESP8266

De locker moet de mogelijkheid hebben om te communiceren met de website. Deze communicatie wordt geregeld door de microcontroller ESP8266 omdat het beschikt over wifi. Hierdoor kan het dus pakketten ontvangen van de website en er naartoe sturen.   
Daarom wordt het masterprogramma op de ESP8266 uitgevoerd. De microcontroller is ook verbonden met de Arduino via I²C pinnen. Deze verbinding zorgt er voor dat deze twee kan communiceren onderling.

A picture containing text, electronics

Description automatically generated

### Arduino

Als microcontroller wordt Arduino gebruikt om al de rest van de componenten te besturen.  
Arduino is een open source elektronicaplatform gebaseerd op gebruiksvriendelijke hardware en software.

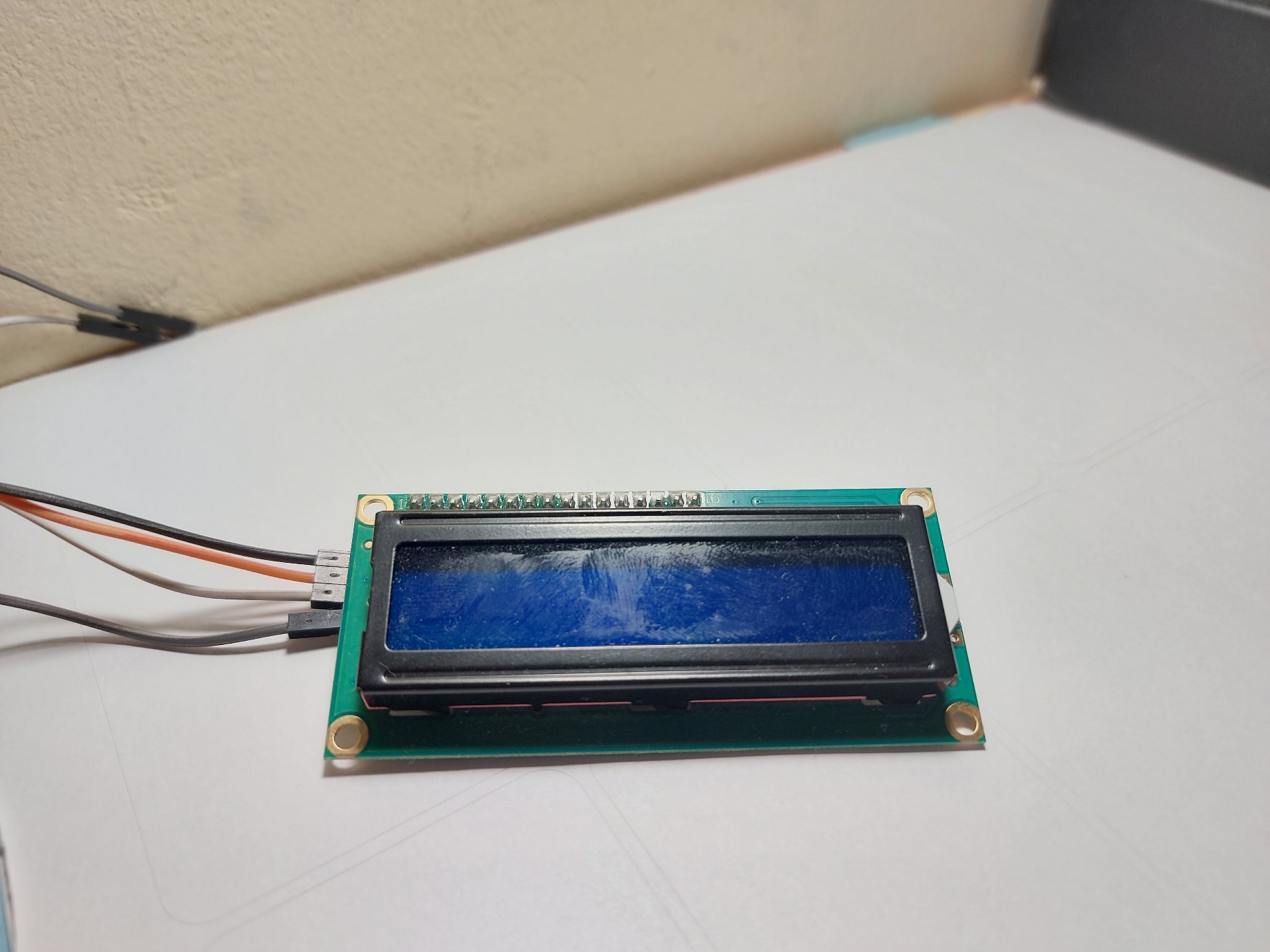
A close-up of a circuit board

Description automatically generated with low confidence

Aan deze microcontroller zijn de volgende componenten verbonden:

* lcd met LCM1602 I²C module
* drie-op-vierraster keypad
* 28BY-48 stappenmotor met ULN2003 motordriver

de lcd of liquid crystal display zorgt ervoor dat er tekst en getallen visueel weergegeven worden. Hierdoor kan de gebruiker zien wat hij/zij moet doen. De LCM1602 I²C module zorgt ervoor dat de lcd minder pinnen vraagt waardoor het gemakkelijker te gebruiken is.



Door de keypad kan de gebruiker wachtwoorden invoeren om zo de locker te openen. Deze zal dan natuurlijk alleen geopend worden als het wachtwoord correct is. Het keypad is gerangschikt in een   
drie-op-vierraster. Dit betekent dat gebruiker 12 verschillende karakters kan invoeren. Deze toetsten zijn #, \* en de cijfers van 0 tot 9.

Graphical user interface

Description automatically generated

Ten laatste is er ook de 28BY-48-stappenmotor en zijn ULN2003-motordriver. De stappenmotor is verantwoordelijk voor het openen/sluiten van de locker. Wanneer een correct wachtwoord is ingevoerd zal de stappenmotor 90 °-draaien. Vervolgens als de locker weer gesloten moet worden, zal het 90 °-graden draaien in de tegengestelde richting.  
De ULN2003-motordriver zorgt er juist voor dat het besturen van de motor gemakkelijker is.

A picture containing wall, indoor

Description automatically generatedA circuit board with wires

Description automatically generated with low confidence

## Software

De software voor de elektronica is gesplitst in twee delen, de master en de slave. Dankzij deze twee programma’s functioneert de locker als een automatische kluis. Het masterprogramma zorgt voor de communicatie met de website. Het slaveprogramma is dan op zijn beurt verantwoordelijk voor de besturing van de kluis zelf.

### Master

Het masterprogramma is verantwoordelijk voor de communicatie tussen de lockers en de website. Dit programma wordt uitgevoerd door de ESP8266. Maar standaard kan de ESP8266 niks ontvangen of sturen naar de website. Daarom zorgt het programma ervoor dat de microcontroller kan functioneren als server. Op deze manier heeft de microcontroller een eigen ip en de mogelijkheid om pakketten te ontvangen.

De werking van het masterprogramma wordt geïllustreerd met de onderstaande flowchart.

Diagram

Description automatically generated

In het begin van het programma wordt er geprobeerd om een connectie te maken met het internet.  
Dit is een cruciale stap want zonder deze connectie kan er niet gecommuniceerd worden met de website. Daarom zal de microcontroller deze stappen blijven herhalen tot het lukt voordat het naar de volgende stap kan gaan.

Graphical user interface, text, application, email

Description automatically generated

Nadat er connectie is gemaakt verwacht de ESP8266 pakketten die de codes bevatten om de lockers te openen. Maar zolang er niks ontvangen wordt, zal het programma wachten tot het een pakket binnenkrijgt.

A picture containing logo

Description automatically generated

Op het moment dat de master een pakket ontvangt, zal het wachtwoord uit het pakket gehaald worden.   
Na het ontvangen van het wachtwoord zal de ESP8266 een conformatiepakket terug naar de website sturen. Op deze manier weet de website dat het initiële pakket goed is ontvangen.  
De microcontroller zal dan op zijn beurt het wachtwoord verder versturen naar de slave via de I²C pinnen.



Nadat de slave het wachtwoord heeft ontvangen zal de master opnieuw wachten voor een pakket van de website. Deze cyclus van ontvangen en versturen zal blijven herhaald worden.

### Slave

Naast de communicatie tussen de locker en de website is de besturing van de elektronica zelf ook belangrijk. Deze taak wordt volbracht door de slave die uitgevoerd wordt door de Arduino. Dit houdt in het openen/sluiten van de locker, het weergeven en invoeren van het wachtwoord.

Voor de werking van het slaveprogramma kan de onderstaande flowchart gevolgd worden.

Diagram

Description automatically generated

Het slaveprogromma maakt gebruik van een verzameling libraries om de verschillende componenten aan te sturen. Deze worden bovenaan het programma vermeld, als gewoonlijk. Hier worden ook cruciale variabelen benoemd en geïnitialiseerd. Dit moet gebeuren voordat het programma start, zodat de componenten correct gebruikt kunnen worden.

Text

Description automatically generated

De slave doet twee taken tegelijkertijd. Het besturen van de locker zelf, terwijl het ook constant luistert naar de master voor een wachtwoord. Als de slave een wachtwoord ontvangt wordt er eerst gecheckt als dit wachtwoord al gekend is door het systeem. In het geval dat het wachtwoord al gekend is, wordt het inkomend wachtwoord verwerpt. Maar in de andere gevallen, wordt het opgeslagen. Zodat het gebruikt kan worden door het systeem.

Text

Description automatically generated

Text

Description automatically generated

Buiten het luisteren naar de master en het checken van de inkomende wachtwoorden, heeft de slave een cyclus om de locker de besturen. In het begin wordt er via de lcd gevraagd om een wachtwoord in te geven. De gebruiker moet dan via het keypad het wachtwoord intikken om zo de locker te openen. Het ingegeven wachtwoord wordt dan vergeleken met de gekende wachtwoorden.   
Als het ingegeven wachtwoord correct is zal de stappenmotor draaien zodat de locker geopend wordt. Zo kan de gebruiker het product verwijderen van de locker. Vervolgens moet de gebruiker de locker zelf nog sluiten door de instructies te volgen die op de lcd getoond worden.

Text

Description automatically generated

Nadat de locker is gesloten, wordt ook het gebruikte wachtwoord verwijdert. Zo kan hetzelfde wachtwoord niet meerdere keren gebruikt worden.

Graphical user interface, text

Description automatically generated

In het geval dat het ingegeven wachtwoord incorrect is, zal de kluis niet geopend worden.  
Na dit zal de volledige cyclus herhaald worden. Zodat de locker hergebruikt kan worden voor een ander product.

Text

Description automatically generated

Om de stappenmotor aan te sturen, Wordt er ook gebruik gemaakt van een functie. Met behulp van deze functie worden graden omgezet naar stappen die stappenmotor begrijpt. Hierdoor wordt besturen van de motor gemakkelijker.

Text

Description automatically generated with medium confidence