Verantwoording eindopdracht React Joost Lauwen

Starten van de app:

Navigeer via de terminal naar de map “sweet-coffee”. Run in de map het command: npm install. Vervolgens run het command: npm start om de server te starten.

Verantwoording use cases:

Ik ben begonnen met de pagina op te splitsen in componenten, om zo een overzicht te krijgen met de te maken componenten en zo ben ik dan ook aan de componenten gekomen die er nu zijn. Dit maakte het makkelijk om de interface al meteen in te richten zoals het er nu uitziet. Hierna ben ik begonnen met de states en de functies te schrijven voor het gedrag van de interface.

Begonnen om de knoppen te disablen. Zodra ik de disable functie had kon ik deze toepassen op de sliders. Heel simpel via een onclick event te maken die de buttons disabled. De disable functie kon ik ook toepassen op de sliders. Later kon ik de disable functie uitbreiden met requirements zodat het hoofdscenario voltooid was. De status gebruikt de drinkName van de gekozen knop en update zo de state van het component status. Door de random ready callback te gebruiken vanuit de SweetCoffeMock.js kon ik de state weer liften na een x aantal seconden.

Zoals ik al zei kon ik via de disable functie met extra requirements redelijk simpel de alternatieve scenario’s implementeren. Hiervoor heb ik wel een functie moeten maken die bijhoudt hoeveel melk en suiker er is en vervolgens dit van de total suiker af halen (of melk). Dit gedrag kan ik weer gebruiken om buttons te disablen en later de error messages te throwen. Dus zodra de koffie en suiker op zijn, disablen sommige elementen op basis van of ze suiker of melk nodig hebben, of chocolade.

De error messages van scenario 2 zijn ook via de errorcallback method uit de SweetCoffeeMock.js gehaald. Op een random nummer wordt een tekst in de div bepaald. Alles disabled en na 15 seconden gaat de error weg en is de state weer bij af, klaar om een nieuw drankje te maken.