

TECHNISCH ONTWERP

TEAM TCZ, PROJECT RECYCLE-INZAMEL APP

RIJKSWEG ZUID 1

6165 BS GELEEN

Inhoudsopgave

Technische specificaties	3
Algemeen	3
Relationeel datamodel	4
Technisch datamodel	4
DataDictionary	6
Technische toepassingen	7
Software	8

TECHNISCHE SPECIFICATIES

Algemeen

Deze Recycle-Inzamel applicatie word uitgewerkt in de vorm van een webapplicatie, deze moet te begrijpen zijn voor kinderen vanaf 6 jaar.

Het kleurschema moet duidelijk maken wat het doel is van deze applicatie, verduurzamen en recyclen voor het welzijn van de Aarde.

Deze applicatie moet als basis gestart worden met een database waarin 50 producten staan, deze producten moeten ieder zijn eigen barcode hebben en een afbeelding. Daarnaast moet het mogelijk zijn om als gebruiker producten toe te voegen, deze producten moeten dan door een admin/beheerder in behandeling genomen worden om zo de producten te filteren op foto's en kwaliteit van informatie.

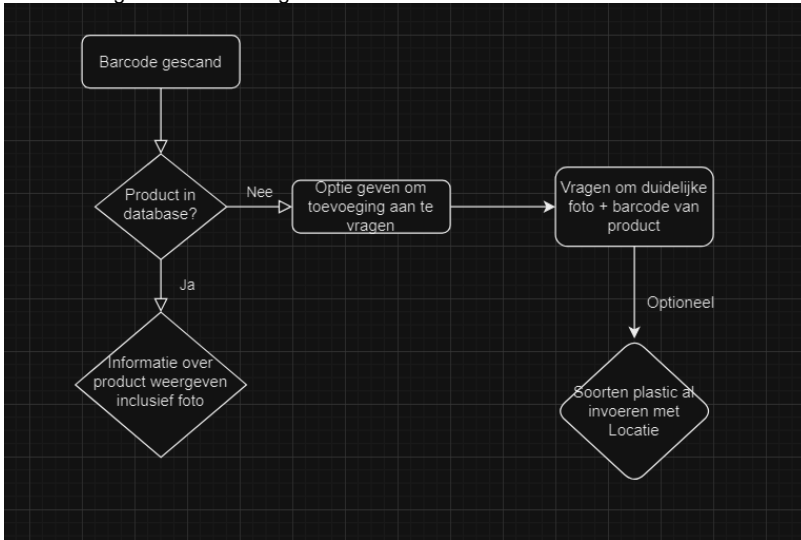
Er is dus ook functionaliteit nodig om als administrator de producten die ingediend zijn te kunnen bekijken via een externe login. Hiervoor zijn dus logingegevens nodig die toegang zullen geven tot een portaal waarin deze producten behandeld kunnen worden. Dit vereist een externe pagina waar alleen de bevoegde personen toegang tot mogen hebben.

Omdat deze app beschikbaar moet zijn voor kinderen vanaf 6 jaar die op de basisschool voorlichting krijgen over deze verduurzaming is het belangrijk dat er met zo veel mogelijk plaatjes en beeld word gewerkt in plaats van tekst.

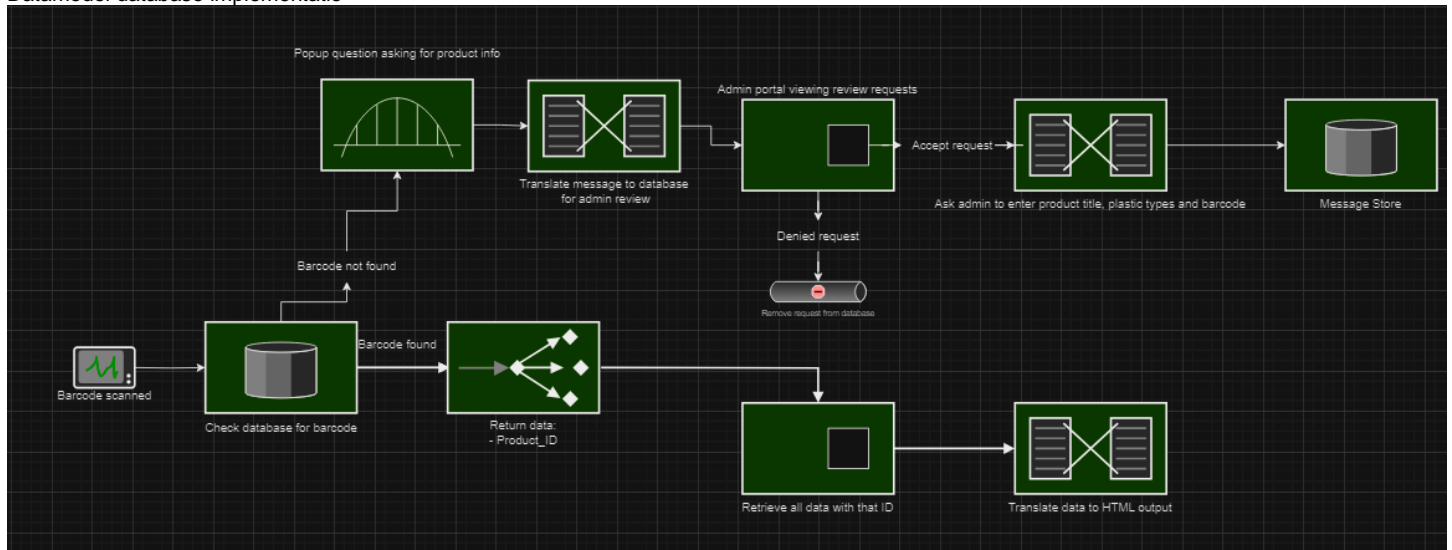
RELATIONEEL DATAMODEL

Technisch datamodel

Datamodel gebruikerservaring



In bovenstaand datamodel is te zien hoe de gebruiker deze applicatie hoort te ervaren op een zo simpel mogelijk uitgelegde manier. Zodra de gebruiker de applicatie opent krijgt hij/zij de mogelijkheid om een barcode scanner aan te klikken om een product te scannen. Dan controleert de applicatie of dit product te vinden is in onze database. Als dit product te vinden is in onze database word informatie over het product en een afbeelding van het product getoond. Als dit product niet te vinden is in onze database word er gevraagd of de gebruiker een product wilt toevoegen, zo ja krijgt de gebruiker de vraag om een foto van het product te maken en de barcode te scannen. En als optionele steun aan het project kan de gebruiker ervoor kiezen om ook direct de soorten plastic aan te kaarten.



In bovenstaand datamodel is te zien hoe de database en webapplicatie moeten reageren op bepaalde acties van gebruikers, zodra een barcode is gescand moet die op zoek gaan in de database of deze bekend is in ons systeem, zo niet moet de gebruiker een melding krijgen dat deze niet gevonden is en of hij/zij wilt helpen de app te verbeteren door deze toe te voegen. Dit moet dan zichtbaar zijn in de adminportaal zodat de administrator dit kan behandelen, als dit product word afgekeurd word deze verwijderd uit de database, als deze is goedgekeurd krijgt de administrator nog enkele velden om in te vullen voordat deze opgeslagen word in de producten database.

DataDictionary

Products database table

Naam	Type	Verplicht	Omschrijving
Product_ID	int(50)	ja	Ieder product word automatisch een ID gegeven door middel van de auto_increment functie, deze moet een primary key hebben.
Product_Title	SMALLTEXT	ja	Ieder product moet een title hebben die een korte duidelijke naam van het product bevat.
Product_Barcode	SMALLTEXT	ja	Ieder product moet een barcode hebben, deze barcodes bestaan uit letters en cijfers gecombineerd of slechts een van deze.
Product_Class t/m Product_Class7	TEXT	ja	Ieder product moet per plastic categorie een beschrijving krijgen welke onderdelen van het product hierin moeten staan.
Product_Description	TEXT	Nee	Ieder product kan de mogelijkheid krijgen tot een bijpassende uitleg, hierin kan bijvoorbeeld gezet worden hoe je dit product het beste kan scheiden, hoe schadelijk dit product kan zijn in de natuur, hoelang het duurt voordat dit product geheel is vergaan.

Admin database table

Naam	Type	Verplicht	Omschrijving
Admin_ID	int(50)	ja	Iedere admin user word automatisch een ID gegeven door middel van de auto_increment functie, deze moet een primary key hebben.
Admin_Username	SMALLTEXT	ja	Iedere admin moet een eigen username hebben (uniek)
Admin_Password	SMALLTEXT	ja	Iedere admin moet een eigen wachtwoord hebben, deze moet met encryptie beveiligd zijn (Bcrypt etc.)
Admin_Name	TEXT	ja	Iedere admin moet een eigen naam hebben, deze word weergegeven op het adminportaal

Products review requests table

Naam	Type	Verplicht	Omschrijving
------	------	-----------	--------------

Review_ID	int(50)	ja	Iedere aanvraag word automatisch een ID gegeven door middel van de auto_increment functie, deze moet een primary key hebben.
Review_Title	SMALLTEXT	ja	Iedere Review moet een title hebben die een korte duidelijke naam van het product bevat.
Review_Barcode	SMALLTEXT	ja	Ieder product moet een barcode hebben, deze barcodes bestaan uit letters en cijfers gecombineerd of slechts een van deze. De gebruiker scant deze barcode bij de aanvraag
Review_Classes	TEXT	nee	De Gebruiker krijgt de mogelijkheid om een bijdrage te leveren aan de innovatie door alvast aan te geven welke soorten plastic dit artikel bevat.
REVIEW_Description	TEXT	Nee	De Gebruiker kan hier eventuele toevoegingen aan het product geven, denk hierbij aan: Waar deze te koop zijn, hoe je deze kunt scheiden, hoeveel soorten plastic dit artikel bevat.

TECHNISCHE TOEPASSINGEN

Software

Voor het maken en implementeren van de data als vorm van html output voor de webapplicatie maken wij gebruik van een API. Deze keuze is door het team zorgvuldig gekozen op de volgende manier. Wij hebben een start gemaakt door te informeren op welke manieren wij het beste data kunnen implementeren in een webapplicatie.

De volgende opties hebben wij overwogen:

- PHP
- JS
- API maken

De volgende overwegingen hebben wij besproken, in Plain JavaScript is er een mogelijkheid voor slimme gebruikers om direct toegang tot onze database te krijgen. Dit is dus een groot beveiligingsrisico die wij in geen enkele situatie mogen lopen. JS viel dus af.

Daarnaast viel de keuze tussen het maken van een externe API of het gebruikmaken van PHP. PHP is een oudere taal die niet vaak meer gebruikt wordt i.v.m. geen updates meer. Een externe API kan je altijd zelf nog onderhouden en wordt door Microsoft zelf beheerd zodra jij dit in een .net manier maakt. Wij hebben dus gekozen voor een ASP.NET 8.0 web API.

Dit met de volgende pluspunten:

- Veiligste manier
- Makkelijkste data transfer
- Goed onderhouden

Daarnaast hebben wij gekeken naar de mogelijkheden om dit product te kunnen testen. Er zijn testservers beschikbaar in het gebouw van TCZ. Deze kunnen wij gebruiken voor enkele testomgevingen.

Implementatie

Omdat onze gebruikers niet verplicht zijn een account aan te maken is het belangrijk om ervoor te zorgen dat zij ook niet op een login pagina terecht kunnen komen, deze is enkel voor administrators beschikbaar. Zo houden wij het zo simpel mogelijk voor onze klant en gebruikers. De gebruikers zullen dus een directe homepage krijgen waarin zij kunnen scannen. Onze admin moet door middel van een login pagina inloggen op zijn/haar account.

Externe software

Voor de barcodescanner en de database is externe software nodig.

Voor de barcodescanner hebben wij research gedaan naar verschillende manieren om dit te implementeren. Diverse JavaScript libraries bieden dit aan, helaas zijn enkele van deze niet geheel werken voor onze doeleinden en blijven deze producten oneindig scannen. HTML5QRcode maakt gebruik van de HTML5 functionaliteiten om barcodes te scannen. Deze zullen wij dan ook gebruiken: <https://github.com/mebjas/html5-qrcode>

Daarnaast hebben wij voor de database nog externe software nodig, dit in combinatie met hardware om de database op te draaien. Hiervoor gaan wij MySQL gebruiken, dit is de meest geadviseerde manier voor de koppeling met PHP en sluit dus aan op onze eisen.