



Werkplaats 2-2022: Het Vragen-hospitaal

Author	Tim Splinter (T&L Company), Mark Otting (Rotterdam Academy)
Version	1.0
Date	10-11-2022

Inleiding

De nieuwe opdracht krijgen we van [de Teach and Learn company](#), een bedrijf uit Rotterdam. Het belangrijkste product uit hun keuken is een digitaal toetsingsplatform, “test-correct”. Studenten maken hun toetsen online en docenten kunnen voortgang en groei analyseren middels verschillende rapportages. Daarnaast heeft het platform de unieke eigenschap om gezamenlijk, geanonimiseerd elkaars antwoorden te kunnen beoordelen.

Inmiddels heeft het platform een enorme databank van honderdduizenden vragen opgebouwd. In de strategie van het bedrijf gaat deze database een grote rol spelen en is het van belang dat de vragen van goede kwaliteit zijn. Om de redacteurs te helpen met verbeteren van de kwaliteit van de inhoud wil men een tool maken. Dit tool, een webapplicatie, zou aan de hand van bekende patronen vraag items moeten tonen waar zaken aan verbeterd moeten worden. Een soort “vragen hospitaal”.

Requirements

De gebruikers van dit platform zullen de redacteurs van de databank worden. We willen hen een platform bieden dat op basis van een aantal bekende patronen vraag items toont met een probleem en de redacteur de mogelijkheid geeft deze te verbeteren of te exporteren.

De applicatie zou normaliter direct op de database in prikken, maar die is heel groot. Test-correct heeft voor ons een demo dataset met vragen en een beperkt aantal kolommen voorbereid (zonder antwoorden en metadata) die de nu bekende problemen illustreren.

Vraag-item verbetering

Er zijn twee patronen in de vragen geïdentificeerd:

- **Er zijn vragen die geen of een ongeldig leerdoel hebben.** Redacteurs willen hier na het lezen van de vraag uit een opzoeklijst het juiste item kunnen kiezen en de wijziging kunnen opslaan.
- **Er zijn vragen waarin HTML of systeemcodes in staan.** Soms kopiëren docenten of redacteurs vragen van een andere bron en nemen zij per ongeluk HTML codes mee. De meest voorkomende zijn bijvoorbeeld “ ”: en “
”. Liefst zouden we ook andere HTML “escape characters” willen zien. Redacteurs willen hier de vraag met de hand wijzigen, of wellicht kunnen we iets maken om een verbetervoorstel te tonen.

We verwachten van de tool dat we per patroon een lijst met alle niet gecorrigeerde vragen te zien krijgen. We willen die vraag op zichzelf kunnen bekijken op en per vraag een verbetering kunnen invoeren. Op het detailscherm verwachten we een aantal zaken te zien:

- De gehele vraag moet zichtbaar zijn.
- Het unieke item-ID moet getoond worden met een naar de vraag in het test-correct platform zelf.
- De redacteur moet direct de vraag kunnen verbeteren.

Wat mooi zijn zijn is als er een optie is waarmee de redacteur een vraag of kan markeren als “uitzondering”. Uitzonderingen hoeven niet langer getoond te worden in de overzichtslijst.

Data kwaliteit

Daarnaast zijn er problemen in de database zelf, veroorzaakt door wijzigingen in de datastructuur, of zijn er problemen gecreëerd tijdens importeren van andere datasets:

- **Sommige kolommen hebben lege waardes waar dat niet mag - of omgekeerd, er staan waardes in een kolom die leeg zou moeten zijn.** Leeg wordt hier gezien als een “Null” waarde, een bekend database begrip. Als uitbreiding zou een redacteur hier een waarde willen invullen en alle items zien die deze waarde wel (of niet!) hebben in de gekozen kolom.
- **Sommige kolommen moeten converteerbaar zijn naar een datatype, maar hebben waardes waarbij dat niet kan.** Vaak is de kolom zelf een tekst-variant en staat deze alles toe en moet deze eigenlijk naar een “strenger” datatype toe. Bijvoorbeeld, een kolom waarvan we verwachten dat er alleen maar hele getallen (integer) waardes in staan, maar waar ook een “1.0” of “één” tussen te vinden is.
- **Redacteurs willen alle rijen zien waarvan een kolom tussen bepaalde waardes valt.** Soms willen redacteurs een set vragen verder onderzoeken in bijvoorbeeld Excel. Daarvoor willen ze eerst een ruwe selectie maken om te voorkomen dat men met miljoenen rijen moet werken. Dus bijvoorbeeld “alles uit de tabel met vragen waarin de kolom *datum_invoer* tussen tijdstip A en tijdstip B valt”. Of “alles uit de tabel met auteurs met een geboortedatum van 1990 tot nu”.

In alle drie de gevallen wil de redacteur een tabel kiezen, daarna een kolom en daarop alle probleemgevallen zien met tenminste het ID veld. Later zou men deze lijsten willen hebben als CSV (https://en.wikipedia.org/wiki/Comma-separated_values) export.

In principe wil men uitgaan van de kolommen in de “questions” tabel, maar indien mogelijk zou deze optie uitgebreid kunnen worden met een keuze uit alle tabellen in de database.

SQL

De specifieke SQL code om patronen te herkennen in de dataset en een eventuele vertering toe te voegen vind je terug in de appendix van dit document.

Login

Omdat dit platform direct toegang tot de vragenbank zal bieden mogen alleen geautoriseerde gebruikers de vragen kunnen inzien en verbeteringen kunnen maken. Er zal een login scherm moeten komen en de tool zal gebruikersnamen en wachtwoorden moeten opslaan.

Vanaf het login scherm verwachten we een overzicht of menu waar gebruikers een keuze voor één van de eerder beschreven problemen kunnen maken.

Styling

De interface moet qua uiterlijk aansluiten bij de test-correct huisstijl. Kleuren, lettertype en schermopbouw kun je hiervoor lenen van de [test-correct](#) website.

We noemen het nu, maar het is vanzelfsprekend dat de interface prettig moet werken. Leef je in in het gedrag van een redacteur die snel zou willen werken. Voorbeelden: bij een vraag zou je een “terug naar lijst” knop willen hebben. En na het bijwerken van een vraag zou je ook terug naar deze lijst willen springen.

Should-haves

Daarnaast zijn er een aantal wensen die het *MVP* zullen opvolgen:

- Er zijn CRUD schermen nodig om de gebruikers en hun wachtwoorden te kunnen instellen. Deze schermen zouden alleen beschikbaar moeten zijn voor gebruikers met “beheer” permissies.
- We verwachten dat er meer verbeteringen geïdentificeerd worden, waar we vergelijkbare schermen voor moeten opzetten. Het zou helpen als de huidige opzet hier al rekening mee houdt en gemakkelijk uitbreidbaar is in toekomstige sprints. Instructie voor toekomstige ontwikkelaars hiervoor zou in de README opgenomen moeten worden.

Database

Omdat de database van test-correct niet openbaar benaderbaar is werken we met een lokale (SQLite) database. Deze is gevuld met vragen door test-correct. Het staat je vrij deze uit te breiden met eigen tabellen om bijvoorbeeld de loginnamen en wachtwoorden van gebruikers op te slaan.

Appendix: SQL queries voor probleem patronen

Vragen die geen of een ongeldig leerdoel hebben:

SELECT * FROM vragen WHERE leerdoel NOT IN (SELECT id FROM leerdoelen);

Updaten van een leerdoel in een gegeven vraag:

UPDATE vragen SET leerdoel = ? WHERE id = ?;

Vragen waarin HTML of systeemcodes in staan:

SELECT * FROM vragen WHERE vraag LIKE "%
%" OR vraag LIKE "% %;";

Rijen waarin een bepaalde kolom leeg is

SELECT * FROM ? WHERE ? IS NULL;

Rijen waarin een bepaalde kolom juist niet leeg is

SELECT * FROM ? WHERE ? IS NOT NULL;

Selecteer ID en een andere specifieke kolom uit een tabel

SELECT id, leerdoel FROM vragen;

Selecteer alle rijen met een waarde tussen bepaalde extremen

SELECT * FROM ? WHERE ? BETWEEN ? and ?;