

Projectwerk

13. Logboek scrum/agile

Wagemakers Jan

17 februari 2018

29/10/2017

We gaan gewoon starten met het schrijven van korte omschrijving designdocument om zo al een duidelijker beeld te krijgen van wat de software juist moet doen.

Documenten moeten nog niet 100% in orde zijn. Belangrijker is dat er een begin is en er over het probleem goed wordt nagedacht.

04/11/2017

Korte omschrijving + definitie-studie : 1ste draft is gemaakt, later nog terug bekijken en verder uitwerken.

Besloten om als volgende stap de opbouw van de database te bekijken. Dit hoort bij analyse/datawoordenboek. Reden om hier mee te starten is dat de database in dit project een zeer belangrijke rol speelt.

Jan gaat database structuur bekijken en eventueel al eens uittesten in bv. MySQL workbench.

11/11/2017

Op scrumbord een extra veld **DRAFT's ter nazicht** aangemaakt om een onderscheid te maken tussen zaken waar als iets voor gedaan is.

Database : zekeringkasten, verdeelborden, tranfso's worden gezien als **Aansluitpunten**, van waaruit **Aansluitingen** naar andere **Aansluitpunten** of **Machines** of andere eindverbruikers vertrekken. Zie **datawoordenboek** voor meer info.

Tegen volgende week moet er een eerste aanzet gemaakt zijn voor de feature list.

18/11/2017

Feature list is aangemaakt.

Use case diagram + visueel ontwerp gelijktijdig uitwerken omdat er een groot verband is tussen de **gebruikersschermen** en de **use cases**.

26/11/2017

Feedback doorgenomen.

Aan het datawoordenboek zijn nog een aantal zaken die verbeterd kunnen worden. Is het bv. een goed idee om met waarden NULL te werken als een aansluiting niet naar een ander aansluitpunt gaat. Zie ook naar de **checklist** die bij datawoordenboek toegevoegd is aan het scrum-board.

Een ander aandachtspunt in verband met het datawoordenboek is de vraag of we de aanduiding van een aansluitpunt als primary key nemen, of toch beter met een extra ID werken. Feedback aan Joris gevraagd die meer ervaring in deze materie heeft.

Momenteel wordt er dus aan het **datawoordenboek**, **gebruikersschermen (visueel ontwerp)** en **use cases** gewerkt.

30/12/2017

Door de drukke agenda van het ontwikkelteam zijn de scrum-meetings uitgesteld geweest.

Het datawoordenboek is aangepast om de benamingen consistent'er te maken. Een SQL-script is aan het datawoordenboek toegevoegd zo dat hiermee de nodige testen gedaan kunnen worden bij het ontwikkelen. Waarden NULL in datawoordenboek zijn behouden. Een aantal testen hebben uitgewezen dat dit geen probleem is om op te vangen en het geeft de mogelijkheid om CONSTRAINT's (zie mySQL script) aan de database toe te voegen.

Aan de feature-list zijn nog een aantal zaken toegevoegd.

Use case diagram is aangemaakt.

Visueel ontwerp is aangemaakt.

Volgende stap is om de **klassendiagrammen** te maken.

28/01/2018

Van de **klassendiagrammen** zijn de verschillende klassen opgeschreven. De klassen voor communicatie met de database (Database) en het tonen van het hoofdscherm (Hoofdscherm) zijn op papier (klad) redelijk uitgewerkt. De andere klassen moeten nog uitgewerkt worden. Er is besloten om met de verdere uitwerking van het klassendiagram te wachten en te starten met het programmeren van de database en hoofdscherm klassen aan de hand van de documentatie die nu is uitgewerkt. Dit om te testen of de theorie die er is uitgewerkt ook praktisch haalbaar is, of er toch iets bij gestuurd moet worden.

Van het visueel ontwerp moeten er een aantal zaken herbekeken worden:

- Er zijn knoppen voorzien om naar beneden te bladeren. Bv. Van VB810 naar K810a. Er is echter geen knop voorzien om terug naar boven te gaan. Het veld voeding op het hoofdscherm kan gebruikt worden als terug-knop. Aangeduid op de afdruk van het visueel ontwerp.
- Het overzicht van de transfo's is een uitzondering ten op zichte van het tonen van de gegevens van een aansluitpunt. Dit komt in het visueel ontwerp niet naar voren. Zaken als de +-A knoppen te bekijken.
- Als er nog zaken naar boven komen, worden deze aangeduid op de afdruk van het visueel ontwerp.

Vraag use case diagram is nog te bekijken (zie smartschool).

Bedoeling is om tegen volgende week voldoende code geschreven te hebben die laat zien dat de gegevens uit de database gelezen kan worden en er tussen de verschillende aansluitpunten gebladerd kan worden. Deze broncode uploaden naar smartschool.

04/02/2018

Tijdens het programmeren van de **Database** en **Hoofdscherm** klassen is gebleken dat louter de grafische uitwerking van het klassendiagram met methodes niet handig is om mee te werken. Daarom zijn de me-

thodes in een klassendiagram-werkdocument duidelijker omschreven door het **doel** van de te programmeren methodes te omschrijven.

De **Database** en **Hoofdscherm** klassen zijn geschreven in die mate dat er gebladerd kan worden tussen de verschillende aansluitpunten. Er kan ook een lijn gewist worden. Bewaren en zoeken werkt ook. Deze code is echter nog in een beginstadium en met heel wat onvolkomenheden (bv. in de Database klasse geen rekening houden met de return-value van open() en close()). Dit moet nog beter afgewerkt worden. Toch laat de huidige code de mogelijkheden reeds zien.

Over zo'n twee weken zouden de attributen en methodes voor de klasse **AansluitingAanpassen** uitgewerkt moeten zijn en al een gedeelte geprogrammeerd moeten zijn. Door deze klasse uit te werken kan dan in de klasse **Hoofdscherm** de + en A knop (in de methode **dgvLaagspanningsnet_CellContentClick** geprogrammeerd worden.

17/02/2018

Attributen en methodes voor de klasse **AansluitingAanpassen** zijn uitgewerkt en ook geprogrammeerd. In het **Hoofdscherm** werken de + en A knoppen. Er zijn tijdens het programmeren wel zaken aangepast aan het klassendiagram. Zo was er in eerste instantie enkel voorzien om een **DataRow** door te geven naar **AansluitingAanpassen**, maar om de uniekheid van een aansluiting te testen was het nodig om de volledige **DataTable** door te geven.

De **Database** klasse is opgekuisd en er wordt nu wel rekening gehouden met de return-value van Open() en Close(). Er is echter nog werk aan, denk bv. aan **SQL-injection**.

Verder is de methode **GetTransfos()** in de **Database** klasse zo aangepast dat deze de lijst van de transfos teruggeeft als een DataSet van aansluitingen van het aansluitpunt **Hoogspanning**. Op deze manier kon de code van de klasse **Hoofdscherm** sterk vereenvoudigd worden.

Opgepast, in tegenstelling tot JAVA beginnen bij C# de methodes met een hoofdletter. Aangepast.

Volgende stap is om het **Klassendiagram** uit te werken voor de klassen Machines/Aansluitpunten - toevoegen/verwijderen/aanpassen. Dit zou ergens begin maart uitgewerkt moeten zijn. Aan alles wat al geprogrammeerd is, is ook nog veel opkuiswerk.