

Technisch Ontwerp | Communication Connoisseur

Wester Frenkel
V1C
1660019

Document Historie

Versie	Datum	Auteur(s)	Toelichting
1.0	16 juli 2017	W. Frenkel	Compleet technisch ontwerp gemaakt

Inhoudsopgave

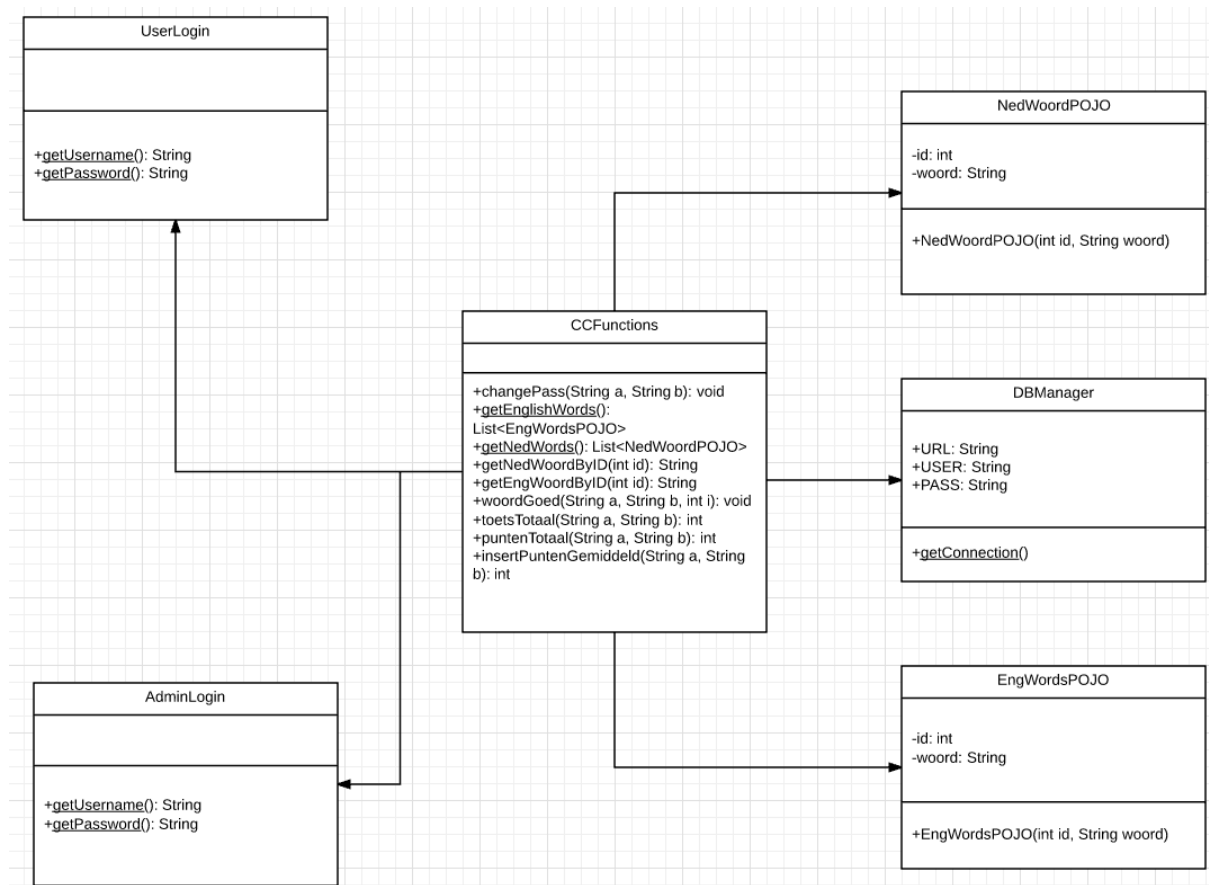
Document Historie.....	0
Inhoudsopgave.....	1
Technisch ontwerp.....	2
2.1 Hoofdstuk 1: Inleiding.....	2
2.2 Hoofdstuk 2: Klassendiagram.....	3
2.3 Hoofdstuk 3: Relatieeel model.....	4
2.4 Hoofdstuk 4: Discussie.....	5

Inleiding

In dit document wordt er een kleine uitleg gegeven over de gebruikte classes/methods in dit project. Naast de uitleg zijn er twee modellen beschikbaar voor een visuele weergave van het systeem. Het gaat hier over een **Klassendiagram**, waar alle belangrijke classes/methods worden weergegeven. Daarnaast is er ook een **Relationeel model** in dit document aanwezig. In het relationeel model staan de tables met hun attributen uit de database die gebruikt zijn in dit project.

Verder is er aan het einde van dit document een kleine discussie over de gebruikte methods/classes en database tables. Dit onderdeel van het document zal gebruikt worden om punten te discussiëren waar het eventueel anders en/of beter kon qua aanpak.

Klassendiagram



CCFunctions

Dit is de grootste class van het systeem. In deze class worden alle functies aangemaakt die nodig zijn om de SQL-statements uit te voeren. Dit is class waar 90% van het project op rust.

User/Admin Login

Deze twee classes bevatten beide twee SELECT-statements om de gebruikersnaam en wachtwoord uit de Oracle Database te fetchen. De fetch-functies worden voornamelijk gebruikt in andere functies, om te kijken wie de handeling uitvoert.

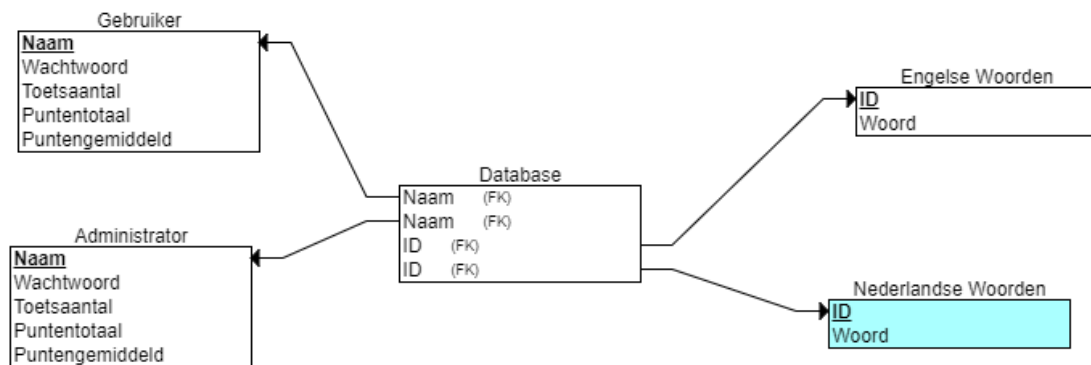
POJO's

De twee POJO's in het project zijn gebruikt om de woordenlijst aan te maken in de CCFunctions class. Door de POJO's kan je het ID en het Woord wat je van de SELECT * statement in een lijst op slaan.

DBManager

De DBManager is gebruikt om de **getConnection()** functie te maken. Deze functie kan op ieder moment in een functie worden aangeroepen om een connectie met de database te leggen.

Relationeel model



Administrator(Naam: varchar, wachtwoord: varchar, toetsaantal: integer, puntentotaal: integer, puntengemiddeld: integer)

Gebruiker (Naam: varchar, wachtwoord: varchar, toetsaantal: integer, puntentotaal: integer, puntengemiddeld: integer)

Engelse Woorden(ID: integer, woord: varchar)

Nederlandse Woorden(ID: integer, woord: varchar)

Discussie

Tijdens de ontwikkeling van Communication Connoisseur zijn er weinig besluiten genomen die niet op de planning stonden. Vrijwel alles verliep volgens het schema, totdat de functionaliteit van Java + Database in elkaar gezet moest worden met het HTTP-protocol en de Tomcat server. Hier ontstond er een **NullPointerException** error waarvan de oplossing nergens te vinden was.

Na deze catastrophale error was er een besluit genomen om de applicatie GUI te bouwen in javafx om zo toch nog een werkende applicatie te krijgen.

Het idee was om, net als de wireframes, meerdere scenes te gebruiken in javafx om zo de "tabs" op de website na te bootsen. Na een nacht doorgehaald te hebben en weer veel errors meegemaakt te hebben is het niet op tijd voor de deadline gelukt om alle functionaliteiten te implementeren.

Uiteindelijk zijn er in de javafx applicatie (4/6) functies terecht gekomen

+ = gelukt | - = niet gelukt

+ Log in

+ Log out

+ Wachtwoord wijzigen

+ Resultaten opvragen (Totaal aantal toetsen gemaakt, Totaal aantal punten behaald, Gemiddeld aantal punten behaald)

- Woordenlijst bekijken

- Overhoren

De javafx class is voor de overzichtelijkheid niet verwerkt in het klassendiagram, aangezien de class nog niet af was.