**MBO Rijnland**

Database MBO

Volgens de ontwikkel concepten UML zijn de hierna volgende producten

uitgewerkt:

* Functioneel ontwerp
* Technisch ontwerp
* Bouwontwerp

MBO Rijnland Gouda

Versie : 1.0

Datum : 13 oktober 2022

**Inhoud**

[1 Opdrachtomschrijving](#_o9r8me3utlha) **3**

[1.1 Inleiding](#_evadfjpjiukj) 3

[2 Probleemstelling](#_us76879ysyov) **3**

[2.1 Probleem](#_uroykkryppvn) 3

[3 Planning & begroting](#_wjyq7tizli6q) **3**

[4 Functioneel Ontwerp](#_gjdgxs) **3**

[4.1 Inleiding](#_30j0zll) 3

[4.2 Detailstructuur toekomstige organisatie](#_1fob9te) 3

[4.3 Detail Functiestructuur](#_3znysh7) 4

[4.3.1 Voorwaarde gebruik van MBO database](#_2et92p0) 4

[4.3.2 Architectuur Client/Server](#_tyjcwt) 4

[4.3.3 Domein analyse (Context diagram)](#_3dy6vkm) 5

[4.3.4 Use Case diagram (Context diagram)](#_1t3h5sf) 6

[4.3.5 Sequence-Diagram](#_2s8eyo1) 6

[4.3.6 Toestandsdiagram](#_17dp8vu) 8

[4.4 Implementatie Klassendiagram](#_3rdcrjn) 9

[4.4.1 Gegevensstructuur](#_26in1rg) 10

[4.4.2 Relationeel Representatie model](#_lnxbz9) 11

# **Opdrachtomschrijving**

## **Inleiding**

De opdracht die ons is gegeven is om een nieuw bestandssysteem te ontwikkelen waarbij de klant alle gegevens kan opslaan in een database in plaats van een excel bestand bij te houden dit kost natuurlijk veel tijd en is af en toe best onoverzichtelijk.

# **Probleemstelling**

## **Probleem**

Het probleem is dat autobedrijf MBO nog geen bestandssysteem heeft waar alles wordt opgeslagen, alles wordt nu opgeslagen in een Excel bestand en dat is niet handig. Het bijwerken is Excel is erg tijdrovend aangezien het maar één keer per maand wordt gedaan en het beter zou zijn als dat 1 keer in de week simpel in een database wordt gedaan.

* 1. **Oplossing**

Als ICT’er werd ik gevraagd of het mogelijk is om voor alle activiteiten die nu in ms-excel worden bijgehouden, een bestandssysteem te ontwikkelen. Hierna is het de bedoeling om op basis van dit bestand-systeem dusdanige programmatuur te ontwikkelen, zodat alle activiteiten op 1 plek kan worden bijgehouden. En ook dat eens per maand automatisch deze gegevens kunnen worden geïmporteerd in het administratiesysteem Ook was de wens om de bestanden van dit bestandssysteem te tonen in een scherm programma.

# Planning & begroting

De planning is om het project binnen 2 maanden af te hebben en het ongeveer €2500,- gaat kosten. Dit bedrag bedraagt alle functionaliteiten die nodig zijn voor een goed werkend systeem. Ik ga in de 2 maanden alle tijd benutten om een mooi en simpel systeem neer te zetten zodat de klant tevreden is een gebruik gaat maken van dit bestandssysteem.

# **Functioneel Ontwerp**

## **Inleiding**

Het systeem moet simpel worden zodat de medewerkers makkelijk en snel dingen kunnen toevoegen zodat het tijd scheelt.

## **Detailstructuur toekomstige organisatie**

Door het invoeren van dit systeem ten opzichte van het huidig systeem moet met de hierna volgende punten rekening worden gehouden:

* Een gebruiksvriendelijke interface
* Een overzichtelijke database
* Een systeem dat snel werkt
* Bestandssysteem tonen in een schermpagina

## **Detail Functiestructuur**

In het basis ontwerp zijn de wensen beschreven en dit hoofdstuk zal beschrijven hoe en op welke voorwaarde de functies van het systeem naar de gebruikers toe zullen worden vervuld.

### **Voorwaarde gebruik van MBO database**

Het gebruik van de database is aan gebruiksvoorwaarden gebonden:

1. Een gebruiker moet via ip-adres een applicatie kunnen aanroepen (object -> gebruiker & website)
2. De applicatie moet voorzien zijn van een header, body en footer (object -> header & body & footer)
3. Bij de applicatie moet een optie voor het inloggen aanwezig zijn (object -> login)
4. De klant moet kunnen inloggen om gegevens te kunnen bekijken/muteren/aanmaken (object -> login & gebruiker)
5. Indien geen gegevens aanwezig, moet de gebruiker deze gegevens kunnen aanmaken (object -> gebruiker)
6. Er moet een melding kunnen worden opgemaakt, indien gegevens zijn aangemaakt/gewijzigd (object -> melding)

### **Architectuur Client/Server**

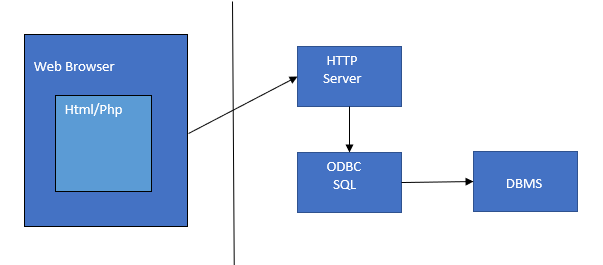
Middels een situatieschets zal de client/server architectuur nader worden toegelicht. Een applicatie presenteert de user-interface aan de gebruiker met daarin de functionaliteiten, dit gebeurt meestal op de homepagina waar de gebruiker belandt als deze de app opent:

Aanroepen webpagina

Login

Vastleggen-, aanvragen en muteren van gegevens

App-Architectuur:

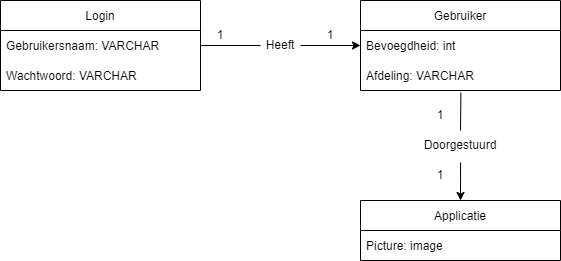


**Legenda:**

| **Nr** | **Naam** | **Omschrijving** |
| --- | --- | --- |
| 1 | Web Browser | Een web browser is computerprogramma om webpagina’s te bekijken. Voorbeelden van web browsers zijn: Microsoft Edge en Google Chrome |
| 2 | HTTP | HTTP staat voor Hypertext Transfer Protocol. HTTP is het protocol voor de communicatie tussen een webclient en een webserver |
| 3 | ODBC | ODBC staat voor Open DataBase Connectivity. Het is gemaakt om elk [programma](https://nl.wikipedia.org/wiki/Computerprogramma) te kunnen laten communiceren met een database en dit onafhankelijk van het type database. |
| 4 | DBMS | DBMS staat voor database management system. Het is een systeem dat als [database](https://nl.wikipedia.org/wiki/Database) opgeslagen gegevens ontsluit, bewaakt en beheert. |

### **Domein analyse (Context diagram)**

Een Domein analyse is een contextueel klasse diagram als uitgangspunt voor de verder ontwerpen.



Legenda domein diagram:

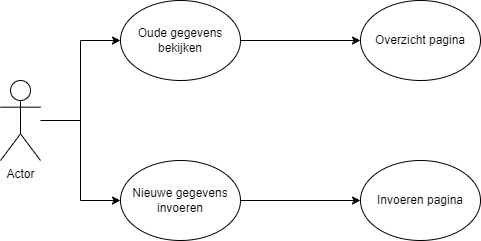
| ***Nr*** | ***Klasse*** | ***Omschrijving*** |
| --- | --- | --- |
| 1 | Login | Inloggen is de term die gebruikt wordt voor het proces van aanmelding van een individu bij een informatiesysteem of computer. |
| 2 | Gebruiker | De gebruiker is de persoon die de applicatie gebruikt. |
| 3 | Applicatie | Met de applicatie wordt de gehele app bedoeld waarmee de gebruiker interactief bezig is. |

### **Use Case diagram (Context diagram)**

Een Use Case diagram is een diagram met als doel een overzicht te geven van het te ontwikkelen systeem en hoe deze zal/kan worden gebruikt.

Dit is de basis voor het beschrijven van de functionaliteiten van het systeem in relatie tot het te ontwerpen systeem.

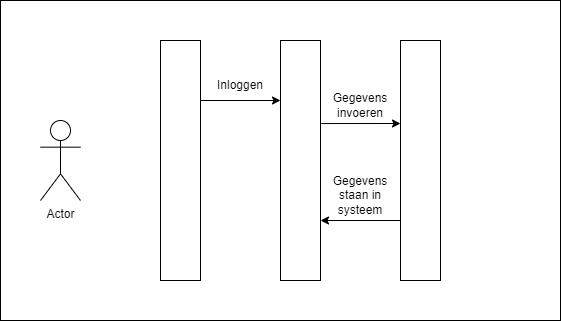
***Use Case Diagram***

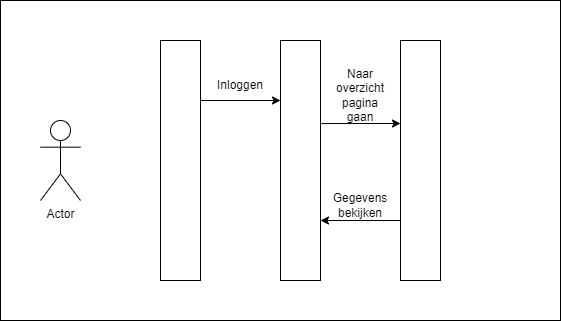


Data Dictionary Use Case:

| Actor | De actor is eigenlijk de klant |
| --- | --- |
| Oude gegevens bekijken | De actor wil eerder ingevoerde gegevens bekijken |
| Nieuwe gegevens invoeren | De actor wil nieuwe gegevens invoeren |
| Overzicht pagina | De actor komt op de overzicht pagina om gegevens te bekijken |
| Bericht versturen | De actor komt op de invoeren pagina waar hij gegevens kan invoeren |

### **Sequence-Diagram**

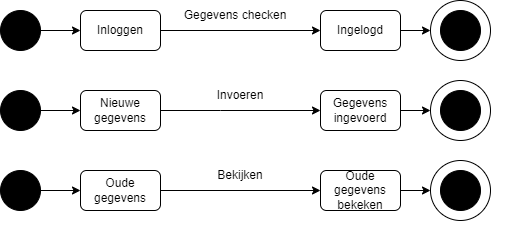


******

### **Toestandsdiagram**

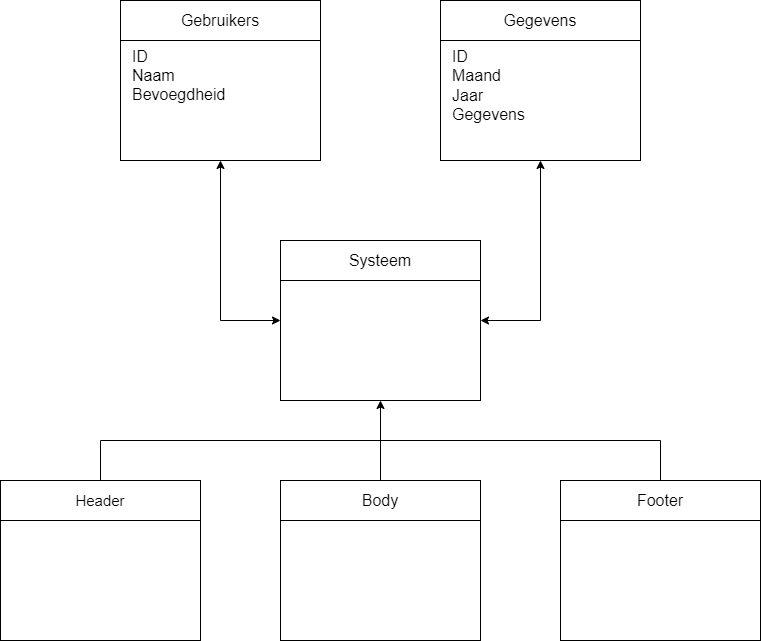
Toestandsdiagrammen zijn activiteiten diagrammen waarin alle toestanden zogenaamde activiteit toestanden zijn. In elke activiteit toestand voert het object een activiteit uit waarna het in een volgende toestand terecht komt. In feite is een toestand een stabiele staat waarin een object zich gedurende enige tijd bevindt.

Met behulp van toestandsdiagrammen zullen wij de processen nader uitwerken met de bijbehorende pre- en postcondities:



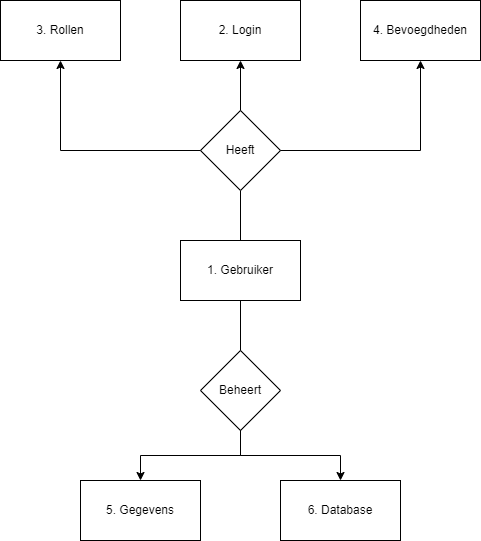
| ***Proces*** | ***Omschrijving*** |
| --- | --- |
| Inloggen  Precondities  Postcondities | Inloggen  Pre-> De applicatie is beschikbaar en er is een inlogsysteem  Post-> De applicatie heeft de inlog goedgekeurd en stuurt de gebruiker door |
| Nieuwe gegevens  Precondities  Postcondities | Nieuwe gegevens invoeren  Pre-> Er zijn nog geen gegevens van de laatste maand.  Post-> De gegevens van de laatste maand zijn ingevoerd in het systeem |
| Oude gegevens  Precondities  Postcondities | Oude gegevens bekijken  Pre-> De oude gegevens staan in het systeem  Post-> De oude gegevens zijn bekeken |

## **Implementatie Klassendiagram**



### **Gegevensstructuur**

Het doel van deze activiteit is om te komen tot een logisch ontwerp van de database (opslag van gegevens ) voor het ontwikkelen van WhatsApp. Hierbij kan worden vastgesteld welke gegevens noodzakelijk zijn om de functies van dit systeem te vervullen. De relaties tussen de gegevens zullen worden weergegeven middels een ERD diagram



***Datadictionary ERD***

| ***Nr*** | ***KLasse*** | ***Omschrijving*** |
| --- | --- | --- |
| 1 | Gebruiker | File Name: Klant-Data |
| 2 | Login | File Name: Login-Data |
| 3 | Rollen | File Name: Rollen-Data |
| 4 | Bevoegdheden | File Name: Bevoegdheden-Data |
| 5 | Gegevens | File Name: Gegevens-Data |
| 6 | Database | File Name: Database-Data |

### **Relationeel Representatie model**

De namen van de velden ten behoeve van de op te bouwen bestanden zullen in groepen worden benoemd. Zo nodig zullen ook de primary- en secondary –keys worden beschreven.

***Opslagstructuur database namen velden Klant-Data***

| Nbr | Gebruikers | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Gebruiker\_id |  |  | 6 | Int | # \* Secondary Key (Unique) |
| 2 | Naam |  |  | 50 | varchar | \* |
| 3 | Bevoegdheid |  |  | 10 | int | \* |

***Opslagstructuur database namen velden Login-Data***

| Nbr | Login | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Gebruiker\_id |  |  | 6 | Int | # \* Primary Key (Unique) |
| 2 | Maand |  |  | 2 | Int | \* |
| 3 | Jaar |  |  | 4 | Year | \* |
| 4 | Gegevens |  |  |  |  | \* |

***Opslagstructuur database namen velden Rollen-Data***

| Nbr | Rollen | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Gebruiker\_id |  |  | 6 | Int | # \* Secondary Key (Unique) |
| 2 | Rol |  |  | 30 | Varchar | \* |

***Opslagstructuur database namen velden Bevoegdheden-Data***

| Nbr | Bevoegdheden | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Admin\_id |  |  | 6 | Int | # \* Primary Key (Unique) |
| 2 | Naam |  |  | 50 | Varchar | \* |
| 3 | Rechten |  |  | 30 | Varchar | \* |

***Opslagstructuur database namen velden Gegevens-Data***

| Nbr | Gegevens | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Auto\_id |  |  |  | Int | # \* Primary Key (Unique) |
| 2 | Controle |  |  |  | boolean | \* |

***Opslagstructuur database namen velden Database-Data***

| Nbr | Database | From | To | Length | Type | Description |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | Auto\_id |  |  | 6 | Int | # \* Secondary Key (Unique) |
| 2 | Bouwjaar |  |  | 4 | Year | \* |
| 3 | Merk |  |  | 25 | Varchar | \* |
| 4 | Model |  |  | 20 | Varchar | \* |

# unique identifier

\*C

***Na ERD concept traject kunnen bestanden worden aangemaakt***