Analyse document



Iteratienummer: 2

Student: Zein Karzon

Versie: 1

Inhoud

[Inleiding/context 3](#_Toc95134861)

[Requirements deze iteratie 3](#_Toc95134863)

[Use Cases deze iteratie 4](#_Toc95134864)

[Schetsen deze iteratie 4](#_Toc95134865)

[Test Cases deze iteratie 5](#_Toc95134866)

[EER alle iteraties 6](#_Toc95134868)

# Inleiding/context

Voor mijn individuele project had ik gekozen om een taakverdeling app te maken waarin gebruikers, groepen kunnen maken, en dan taken tussen elkaar kunnen verdelen. In dit document lees je mijn eerste analyse van dit project.

# Requirements deze iteratie

Functionele requirements zijn eisen die aangeven welke functies een systeem moet kunnen bieden aan de gebruiker. In de functionele requirements staat de functionaliteit van het systeem beschreven, dus functies die de gebruiker van de applicatie later kan uitvoeren.

**FR-04 [M]** Gebruiker kan een groep maken.

**K-04.01** Als de gebruiker een groep maakt, dan is hij de leider van dit groep.

**K-04.02** De gebruiker kan gegevens van de groep toevoegen en wijzigen.

**B-04.01**

**FR-05 [M]** Gebruiker kan andere gebruikers toevoegen aan een groep.

**K-05.01** De groepsleider kan een andere gebruiker vinden en toevoegen door een ID.

**K-05.02** De groepsleider kan andere groepsleden verwijderen.

**B-05.01** Alleen de groepsleider kan andere gebruikers toevoegen of verwijderen.

**FR-06 [M]** Groepsleider kan groepstaken toevoegen en verwijderen

**B-06.01** Alleen de groepsleider kan dat doen.

**K-06.01** De gebruiker kan zelf kiezen wat hij/zij wil invoeren.

**FR-07 [M]** Gebruikers kan groepsgevoelens en taken in een dashboard zien.

**K-07.01** Alle gebruiker in de groep kunnen dit zien

**B-07.01** Alleen groepsleider kan iets in dashboard toevoegen, wijzigen of verwijderen.

Use Cases deze iteratie

Een Use case is een beschrijving van de acties die tussen de gebruiker en systeem gebeurt in de applicatie.

|  |  |
| --- | --- |
| Naam | UC03: Groep maken. (FR-04) |
| Samenvatting | Na het invoeren van gegevens wordt een groep gemaakt. |
| Actors | Gebruikers |
| Scenario | 1. De actor geeft aan dat hij/zij een groep wil maken 2. Het systeem toont een pagina waarop gegevens ingevoeld kunnen worden. 3. De actor voert de gevraagde gegevens in. 4. Het systeem controleert de ingevoerde gegevens en maakt een groep aan. |
| Uitzondering | Niet alle verplichte gegevens zijn ingevoeld. Toon een melding en ga terug  Naar stap 3 |
| Resultaat | Er is een nieuwe groep gemaakt en toegevoegd aan de database. |

|  |  |
| --- | --- |
| Naam | UC04: Gebruikers toevoegen aan een groep. (FR-05) |
| Samenvatting | Na het toevoegen van een gebruiker aan een groep, hij kan het dashboard van dit groep zien. |
| Actors | Groepsleider |
| Scenario | 1. De actor geeft aan dat hij/zij gebruiker wil toevoegen 2. Het systeem toont een pagina waarop de ID van een gebruiker wordt gevraagd. 3. De actor voert de gevraagde ID in. 4. Het systeem controleert de ingevoerde ID en voegt de gebruiker toe aan de groep. |
| Uitzondering | Gebruiker ID bestaat niet. Toon een melding en ga terug  Naar stap 3 |
| Resultaat | Er is een nieuwe gebruiker aan de groep toe gevoegd. |

|  |  |
| --- | --- |
| Naam | UC05: Groepstaak toevoegen. (FR-06) |
| Samenvatting | Na het invoeren van een groepstaak komt die in het groepsdashboard bij de taken te staan. |
| Actors | Groepsleider |
| Scenario | 1. De actor geeft aan dat hij/zij een groepstaak wil toevoegen 2. Het systeem toont een pagina waarop gegevens ingevoeld kunnen worden. 3. De actor voert de gevraagde gegevens in. 4. Het systeem controleert de ingevoerde gegevens en voegt de groepstaak toe. |
| Uitzondering | Niet alle verplichte gegevens zijn ingevoeld. Toon een melding en ga terug  naar stap 3 |
| Resultaat | Er is een nieuwe groepstaak toegevoegd aan het groepsdashboard |

# Schetsen deze iteratie

Dit is een simpele schets van de applicatie design om het duidelijker te maken wat velden en attributies ik later nodig heb.

Afbeelding met tekst, whiteboard

Automatisch gegenereerde beschrijving

# Test Cases deze iteratie

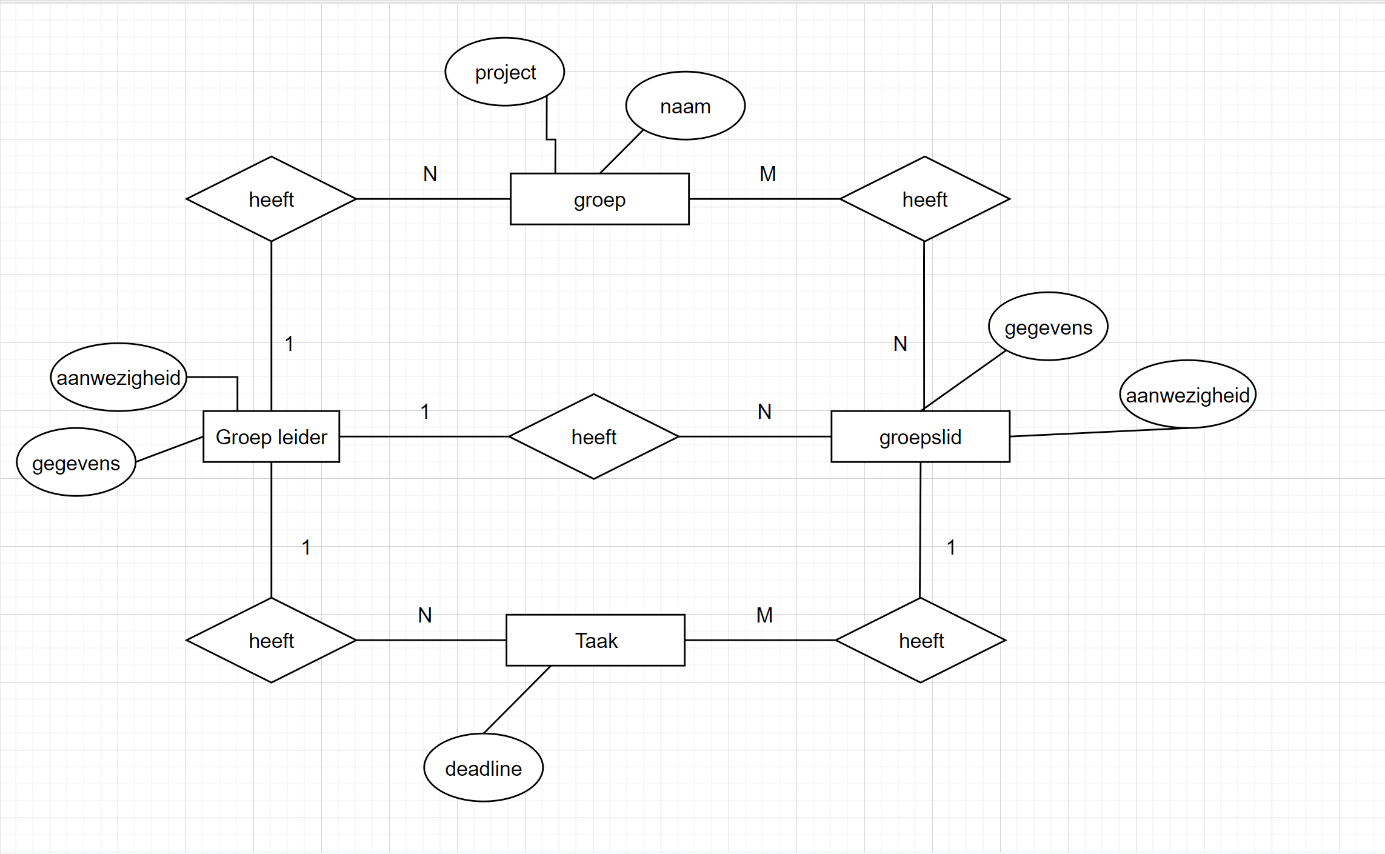
Test case is een beschrijving van stappen en verwachte resultaten om de applicatie te testen. Het wordt gemaakt doormiddel van de Use cases. Goed geschreven testcases kan de testcyclus van de applicatie soepel en efficiënt maken.

|  |  |  |  |
| --- | --- | --- | --- |
| Testcase | Use case | Invoer | Verwachte uitvoer |
| TC01 | UC03 | Gebruiker navigeert naar de “Groep maken” pagina | Systeem geeft de “Groep maken” pagina weer met de velden: Groepsnaam en project |
| TC02 | UC03 | Gebruiker voert niet alle gevraagde gegevens in | Systeem toont een foutmelding |
| TC03 | UC03 | Gebruiker voert alle gevraagde gegevens in | Systeem slaat de gegevens op en groep komt in de database te staan |

|  |  |  |  |
| --- | --- | --- | --- |
| Testcase | Use case | Invoer | Verwachte uitvoer |
| TC01 | UC04 | Groepsleider navigeert naar de “Gebruiker toevoegen” pagina | Systeem geeft de “Gebruiker toevoegen” pagina weer met het veld: Gebruiker ID |
| TC02 | UC04 | Groepsleider voert een niet-bestaande ID in. | Systeem toont een melding “Gebruiker bestaat niet” |
| TC03 | UC04 | Groepsleider voert een bestaande ID in. | Systeem voegt gebruiker met de opgegeven ID toe aan de groep. |
| Testcase | Use case | Invoer | Verwachte uitvoer |
| TC01 | UC05 | Gebruiker navigeert naar de “Groepstaak toevoegen” pagina | Systeem geeft de “Groepstaak toevoegen” pagina weer met de velden: Naam taak, beschrijving en deadline |
| TC02 | UC05 | Gebruiker voert niet alle gevraagde gegevens in | Systeem toont een foutmelding |
| TC03 | UC05 | Gebruiker voert alle gevraagde gegevens in | Systeem slaat de gegevens op en die komen op de Groep overzichtspagina te staan. |

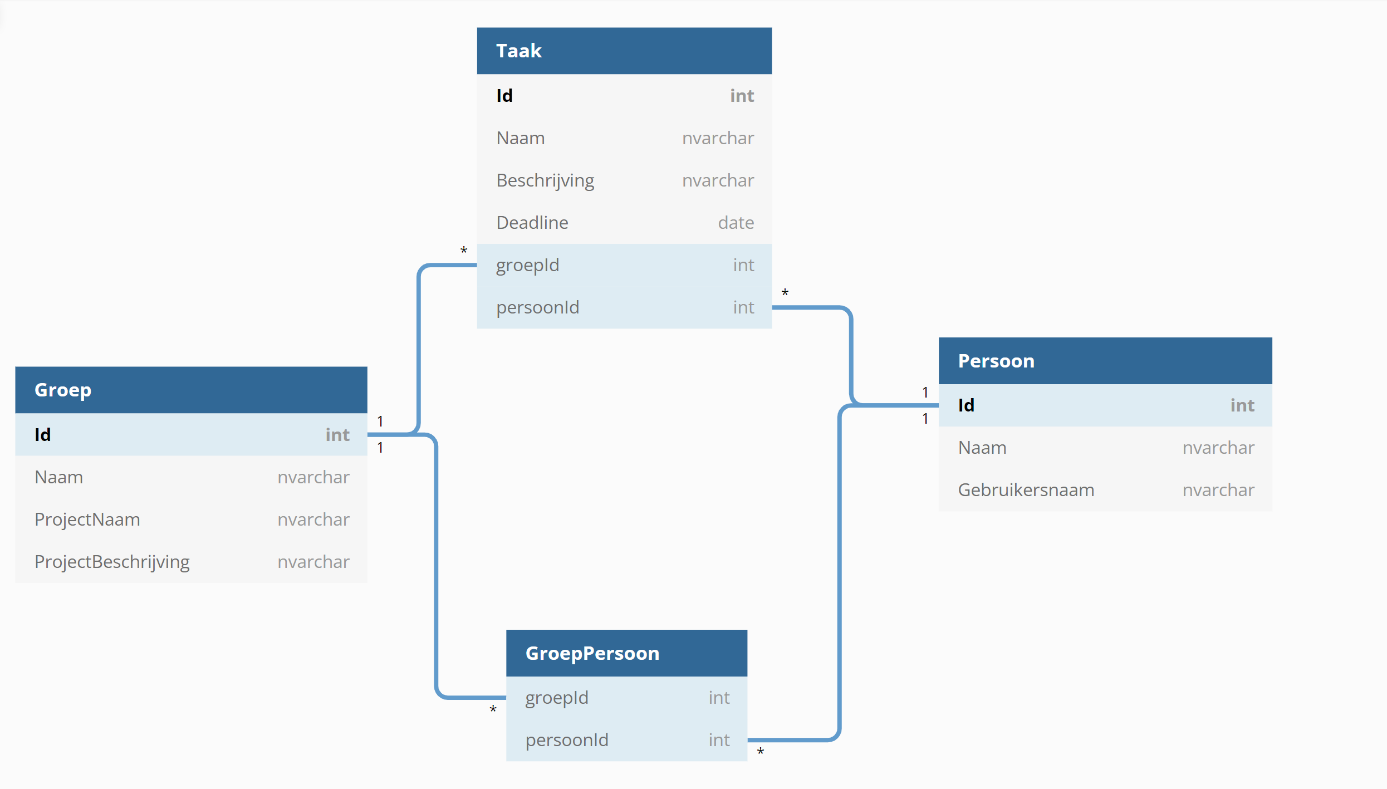
# EER alle iteraties

Het entity-relationship diagram (ERD) is een diagram voor het inzichtelijk te maken van een conceptueel datamodel Het is een visuele weergave van de entiteiten, relaties en attributies om een logisch ontwerp te maken.



# Database tot diagram Deze iteraties

Databasediagram toont grafisch de structuur van de database en relaties tussen databaseobjecten.



# Reflectie

**Wat ging er goed dit sprint?**

Volgens mij was dit sprint een succes want ik heb veel nieuwe dingen geleerd, een van de belangrijkste is dat ik nu op een basisniveau database kan maken en gebruiken in mijn applicatie. Ik was deze sprint voor het meest bezig met de database, maar ook had ik mijn applicatie uitgebreid om het klaar te maken voor het asp.net volgende sprint. Ik had dit sprint geprobeerd om meer feedback te vragen van Rob en Bjorn om mijn code te blijven verbeteren.

**Wat ging er minder goed?**

Ik had dit sprint geen echt problemen gekregen, alles liep wel goed in het individuele project.

**Wat kan beter?**

Ik was altijd nieuwe dingen aan het zoeken en proberen daarom was ik deze sprint meer met de code bezig terwijl ik zoveel mogelijk in de analyse document moest doen dus Iets dat ik later kan verbeteren is dat de diagrammen (zoals UML en DBO) moeten in het begin gemaakt worden.

# Leeruitkomsten

LU1: feedback van **stakeholders: ik had deze sprint geprobeerd om twee keer per week feedback te vragen bij de docenten om verbetering en suggesties te krijgen**

LU2: Samenwerken en communiceren: ik heb met Amier de samen gewerkt om een database te maken voor de proftaak project

LU6: Een databasesysteem maken en integreren met een applicatie: Want ik heb een database gemaakt en gebruikt in mijn applicatie, Ook heb ik een andere gemaakt voor de proftaak project

LU7: Je **verbetert** en **toont** de kwaliteit van je software **continu** **aan: ik heb dit sprint nieuwe technieken gebruikte zoals class library , gebruik van git , gebruik van summaries om mijn code duidelijker te maken.**