Functioneel Ontwerp

TypoTrainer v0.1.0

Project: TypoTrainer – Applicatie om snel en foutloos te leren typen

Sprint: 0 – Initiële analyse

Niveau: 2  
Versie: 0.1.0  
Datum: 14-11-2025

**Scrum Team**: Tim de Boer (1218918), Edward Popova (1208441), Kai Bloemers (1217499), Nicky Tse (117484)

2025

Versiebeheer

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Wijzigingen |
| 0.1.0 | 10-11-2025 | Initiële versie |
| 1.1.0 |  |  |
| 1.2.0 |  |  |
| 1.3.0 |  |  |
| 1.4.0 |  |  |
| 1.5.0 |  |  |

Distributie

|  |  |  |
| --- | --- | --- |
| Versie | Datum | Ontvangers |
| 1.0.0 | 10-11-2025 | Tim de Boer, Kai Bloemers, Nicky Tse |
| 1.1.0 |  |  |
| 1.2.0 |  |  |
| 1.3.0 |  |  |
| 1.4.0. |  |  |
| 1.5.0 |  |  |

Inhoud

[TypoTrainer v0.1.0 1](#_Toc213978041)

[1 Inleiding 4](#_Toc213978042)

[2 Domeinanalyse 4](#_Toc213978043)

[2.1 Gewenste situatie 4](#_Toc213978044)

[2.2 Domeinmodel 4](#_Toc213978045)

[2.3 Use case diagram 9](#_Toc213978046)

[5 Functionele en non-functionele requirements 13](#_Toc213978047)

[6 Definition of Done 13](#_Toc213978048)

[7 Use cases (UC) 14](#_Toc213978049)

[7.1 UC1 Inloggen 14](#_Toc213978050)

[7.2 UC2 Account aanmaken 15](#_Toc213978051)

[7.3 UC3 Typo-oefening starten 16](#_Toc213978052)

[7.4 UC4 Resultaten tonen 17](#_Toc213978053)

[7.5 UC5 Adaptieve oefening genereren 18](#_Toc213978054)

[7.6 UC6 Gebruikersvoortgang bijhouden 19](#_Toc213978055)

# Inleiding

Dit document beschrijft het Functioneel Ontwerp (FO) voor TypoTrainer, een leerapplicatie die gebruikers helpt om snel en foutloos te leren typen. Het Functioneel Ontwerp is gebaseerd op de gevalideerde requirements en beschrijft de functionele werking, de gebruikersscenario’s, het domeinmodel en de schermontwerpen. Het vormt de basis voor de implementatie in Sprint 1.

# Domeinanalyse

In dit hoofdstuk staat een beschrijving van het domein dat is voortgekomen uit een gesprek met de opdrachtgever. Het domein valt samen met de opgestelde requirements. Om het domein inzichtelijk te maken is gebruik gemaakt van een domeinmodel en een use case diagram.  
Het hoofdstuk begint met de huidige situatie, gevolgd door een beschrijving van de gewenste situatie. Daarna volgen de diagrammen met beschrijvingen van het domein.

## Gewenste situatie

De gebruiker logt in op het systeem, kiest een oefening en typt de getoonde tekst. De app meet snelheid en nauwkeurigheid, markeert fouten en toont resultaten. Het systeem past zich automatisch aan op basis van de prestaties van de gebruiker (adaptief leren). Docenten kunnen voortgang van studenten bekijken.

## Afbeelding met tekst, diagram, ontvangst, Lettertype Door AI gegenereerde inhoud is mogelijk onjuist.Domeinmodel

Afbeelding met tekst, diagram, software, Perceel

Door AI gegenereerde inhoud is mogelijk onjuist.

**Toelichting op het Domeinmodel TypoTrainer (v1.0)**

Het domeinmodel beschrijft de belangrijkste gegevensstructuren (entiteiten) van het systeem TypoTrainer en hoe deze met elkaar samenhangen.

Het model vormt de logische basis voor de database- en softwarearchitectuur.

**1.** **Gebruiker (leerling / student / docent)**

Hierin worden alle algemene gegevens over een gebruiker beheerd.

**Attributen:**

* GebruikerID (PK) – Unieke primaire sleutel (Primary Key) voor elke gebruiker.
* Naam – De volledige naam van de gebruiker.
* Email – Gebruikersadres voor login en communicatie.
* WachtwoordHash – Versleutelde versie van het wachtwoord (voor beveiliging).
* Rol – Geeft aan of iemand Student, Docent of Beheerder is.
* Aanmaakdatum – Datum waarop het account is aangemaakt.

**Afkortingen:**

* PK = Primary Key
* FK = Foreign Key

**2. Student (subklasse van Gebruiker)**

Bevat specifieke gegevens over een student die oefeningen uitvoert.

**Attributen:**

* Leerdoelen – Persoonlijke doelen (zoals typesnelheid verbeteren).
* Typeniveau – Huidige vaardigheidsniveau van de student.
* TotaalOefeningen – Aantal uitgevoerde oefeningen.

**Relaties:**

* Wordt begeleid door één Docent (N:1).
* Voert meerdere Oefeningen uit via Resultaat (N:M).
* Heeft één Statistiek (1:1).
* Kan gekoppeld zijn aan één AdaptieveOefeningGenerator.

**3. Docent (subklasse van Gebruiker)**

Beheert studenten en monitort hun voortgang.

**Attributen:**

* School – De onderwijsinstelling waaraan de docent verbonden is.
* Vakgebied – Onderwijsvak waarin de docent actief is.

**Relaties:**

Kan meerdere Studenten begeleiden (1:N).

**4. Oefening**

Vertegenwoordigt een oefening die de student uitvoert.

**Attributen:**

* OefeningID (PK) – Uniek ID van de oefening.
* Titel – Naam van de oefening.
* Type – Aard van de oefening (bijv. tekst, spraak of beginner).
* Moeilijkheidsgraad – Niveau van de oefening (bijv. makkelijk, gemiddeld, moeilijk).
* Duur – Gemiddelde tijdsduur van de oefening.
* DatumToegevoegd – Wanneer de oefening aan de database is toegevoegd.
* Actief – Boolean (waar/niet waar) die aangeeft of de oefening beschikbaar is.

**Relaties:**

* Eén Oefening gebruikt één Oefentekst (1:N).
* Eén Student kan meerdere Oefeningen uitvoeren (via Resultaat).

**5. Oefentekst**

De tekst of spraakbron die tijdens de oefening wordt gebruikt.

**Attributen:**

* TekstID (PK) – Uniek ID van de oefentekst.
* Inhoud – De eigenlijke tekst of verwijzing naar een audiobestand.
* Categorie – Soort oefenmateriaal (letters, woorden, zinnen of thema’s).
* Taal – De taal van de oefentekst.
* Lengte – Lengte van de tekst (in woorden of karakters).

**Relatie:**

Eén Oefentekst kan worden gebruikt in meerdere Oefeningen.

**6. Resultaat**

Koppelt een Student aan een uitgevoerde Oefening en slaat de prestaties op.

**Attributen:**

* ResultaatID (PK) – Uniek ID van het resultaat.
* GebruikerID (FK) – Verwijzing naar de Student.
* OefeningID (FK) – Verwijzing naar de Oefening.
* Snelheid – Typ­snelheid in woorden per minuut (WPM).
* AantalFouten – Aantal gemaakte fouten.
* Nauwkeurigheid (%) – Percentage correcte toetsaanslagen.
* Datum – Wanneer de oefening is uitgevoerd.
* Opmerkingen – Extra observaties of notities.

**Relatie:**

* Elk Resultaat hoort bij één Student en één Oefening.
* Een Resultaat kan één Foutanalyse hebben (1:1).

**7. Foutanalyse**

Bevat de foutgegevens en analyse van een specifieke oefening.

**Attributen:**

* AnalyseID (PK) – Uniek ID voor de foutanalyse.
* ResultaatID (FK) – Verwijzing naar het bijbehorende resultaat.
* FouteLetters – Lijst met fout getypte letters.
* FouteWoorden – Lijst met fout getypte woorden.
* Correctievoorstel – Suggestie voor verbetering.
* GemiddeldeReactietijd – Tijd tussen toetsaanslagen.

**Relaties:**

* 1:1 met Resultaat.
* Wordt gebruikt door AdaptieveOefeningGenerator om nieuwe oefeningen te genereren.

**8. AdaptieveOefeningGenerator**

Automatisch systeem dat nieuwe oefeningen aanpast op basis van fouten en prestaties.

**Attributen:**

* GeneratorID (PK) – Uniek ID van de generator.
* GebruikerID (FK) – Verwijzing naar de Student.
* DatumLaatsteAnalyse – Laatste datum waarop de analyse is uitgevoerd.
* GebaseerdOpAnalyse (boolean) – True/False of de oefening gepersonaliseerd is.
* AanbevolenOefeningID (FK) – Oefening die als volgende wordt aanbevolen.

**Relaties:**

* Gebruikt Foutanalyse als input.
* Genereert nieuwe Oefening voor de Student.

**9. Statistiek**

Verzamelt overkoepelende prestatiegegevens van een Student.

**Attributen:**

* StatistiekID (PK) – Uniek ID van de statistiek.
* GebruikerID (FK) – Verwijzing naar de Student.
* GemiddeldeSnelheid – Gemiddelde woorden per minuut.
* GemiddeldeNauwkeurigheid – Gemiddelde juistheid in %.
* TotaalOefeningen – Totaal aantal uitgevoerde oefeningen.
* LaatsteUpdate – Datum van laatste berekening.

**Relatie:**

1:1 met Student.

**Samenvatting**

|  |  |  |
| --- | --- | --- |
| **Relatie** | **Type** | **Betekenis** |
| Docent – Student | 1:N | Eén docent begeleidt meerdere studenten |
| Student – Oefening | N:M (via Resultaat) | Studenten kunnen meerdere oefeningen uitvoeren |
| Oefening – Oefentekst | N:1 | Meerdere oefeningen kunnen dezelfde tekst gebruiken |
| Resultaat – Foutanalyse | 1:1 | Elk resultaat heeft maximaal één foutanalyse |
| Foutanalyse – AdaptieveOefeningGenerator | N:1 | Meerdere analyses kunnen leiden tot één nieuw oefenadvies |
| Student – Statistiek | 1:1 | Eén student heeft één overzicht met gemiddelde resultaten |

**Opmerking over notaties**

* PK (Primary Key) = unieke sleutel voor identificatie van een record.
* FK (Foreign Key) = verwijst naar de PK van een andere entiteit.
* 1:N / N:M / 1:1 = kardinaliteit (verhouding tussen objecten).
* Boolean = logische waarde (True/False).
* (abstract) = klasse die niet direct wordt geïnstantieerd (zoals “Gebruiker”).

**De opdrachtgever wil een systeem dat:**

* **Gebruiksvriendelijk is, met overzichtelijke schermen voor oefeningen, resultaten en voortgang;**
* **Data consistent opslaat in een beveiligde database, inclusief oefenresultaten en gebruikersprofielen;**
* **Persoonlijke leertrajecten ondersteunt, waarbij oefeningen zich automatisch aanpassen aan de fouten en snelheid van de gebruiker;**
* **Veilig omgaat met gebruikersgegevens, conform de AVG (**Algemene Verordening Gegevensbescherming - Europese wetgeving over de bescherming van persoonsgegevens)**, en betrouwbare sessies biedt bij in- en uitloggen;**
* **Duidelijke en motiverende feedback geeft tijdens en na het typen, zodat de gebruiker inzicht krijgt in snelheid en nauwkeurigheid:**
* **Voorbereid is op uitbreiding, zoals spraakgestuurde oefeningen, multiplayer-modi of integratie met educatieve platforms in toekomstige releases.**

## Afbeelding met tekst, schermopname, software, Computerpictogram Door AI gegenereerde inhoud is mogelijk onjuist.Use case diagram

student / leerling

*Figuur 2a Use Case Diagram (Student/Docent/Systeem)*

Het Use Case Diagram UI toont de interactie tussen de gebruiker en het systeem.

De belangrijkste use cases zijn:  
- Inloggen  
- Typo-oefening starten  
- Resultaten tonen  
- Adaptieve oefening genereren  
- Gebruikersvoortgang bekijken

Afbeelding met tekst, diagram, Lettertype, lijn

Door AI gegenereerde inhoud is mogelijk onjuist.

Figuur 2b Uitgebreid Use Case Diagram

**Toelichting op het Use Case Diagram – TypoTrainer**

**Subsystem**

* TypoTrainer

**Actoren**

* Student
* Docent
* Systeembeheerder
* <<Service>> Identity Provider
* <<Service>> Interne database

**Use Cases binnen het subsysteem “TypoTrainer”**

1. Inloggen

* Actor: Student, Docent
* Relatie: <<include>> Authenticatie bij Identity Provider

1. Account aanmaken

* Actor: Student
* Relatie: <<extend>> Validatie via personal code

1. Typo-oefening starten

* Actor: Student
* Relaties:
* <<include>> Oefentekst ophalen uit Interne database

1. Fouten analyseren

* Actor: Student
* Relatie: <<include>> Resultaten opslaan in Interne database

1. Adaptieve oefening genereren

* Actor: Systeem (intern proces)
* Relaties:
* <<include>> Analyse resultaten uit vorige oefeningen
* <<extend>> Genereer moeilijkere oefening

1. Resultaten bekijken

* Actor: Student, Docent
* Relatie: <<include>> Data ophalen uit Interne database

1. Voortgang monitoren

* Actor: Docent
* Relatie: <<include>> Aggregatie van studentenresultaten

1. Systeem beheren

* Actor: Systeembeheerder
* Relatie: <<include>> Gebruikers- en dataonderhoud

**Relaties overzicht**

|  |  |
| --- | --- |
| Actor | Interacties met Use Cases |
| Student | Inloggen, Account aanmaken, Typo-oefening starten, Fouten analyseren, Adaptieve oefening genereren, Resultaten bekijken |
| Docent | Inloggen, Resultaten bekijken, Voortgang monitoren |
| Systeembeheerder | Systeem beheren |
| Speech Recognition API (mogelijke uitbreiding) | Ondersteunt spraakgestuurde oefening |
| Identity Provider | Authenticeert gebruikers tijdens het inloggen |
| Interne database | Slaat oefenteksten en resultaten op en levert ze terug aan de applicatie |

**UML-definitie**

|  |  |  |
| --- | --- | --- |
| Element | Betekenis | Voorbeeld in jouw diagram |
| Subsystem | De afbakening van het systeem dat je modelleert | <<Subsystem>> TypoTrainer |
| Actor | Een externe gebruiker of systeem dat met TypoTrainer communiceert | Student, Docent, Systeembeheerder |
| Use Case | Een handeling of functie die het systeem uitvoert | Inloggen, Oefening starten, Fouten analyseren, Resultaten bekijken |
| Include-relatie | Een verplicht onderdeel dat altijd plaatsvindt binnen een andere use case | *Inloggen* → <<include>> Authenticatie bij Identity Provider |
| Extend-relatie | Een optionele of voorwaardelijke uitbreiding van een andere use case | *Account aanmaken* → <<extend>> Validatie via personal code |
| Service-actor | Externe systemen die ondersteuning bieden | Identity Provider, Interne database |

1. Gevalideerde schermontwerpen en mockups

**Doel:**  
Dit hoofdstuk beschrijft de eerste schermontwerpen (User Interface (UI) -mockups) en de bijbehorende gebruikersscenario’s. De ontwerpen dienen als basis voor validatie met de product owner en toekomstige sprintplanning.

**Validatiecriteria:**

* De schermen sluiten aan op de functionele en non-functionele requirements.
* De navigatie en interacties zijn duidelijk en intuïtief.
* De visuele elementen ondersteunen het leerdoel (typen, feedback, motivatie).
* Feedback van de product owner is verwerkt in de definitieve versie.

**Overzicht schermen (voorbeeld):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nr | Schermnaam | Doel | Belangrijkste elementen | Validatiestatus |
| S1 | Inlog/registratie scherm | Account aanmaken  Toegang tot persoonlijke omgeving | Gebruikersnaam, wachtwoordveld, login-knop | Goedgekeurd |
| S2 | Dashboard | Overzicht voortgang en oefeningen | Navigatiebalk, voortgangsbalk, oefening-knop | In review |
| S3 | Typo-oefening | Uitvoeren van typetest | Oefentekst, timer, foutmarkering, pauzeknop | Concept |
| S4 | Resultatenpagina | Weergave van prestaties | Woorden per minuut, foutenpercentage, herstartknop | Concept |

1. Scenario’s per scherm

Voor elk scherm wordt hieronder een gebruiksscenario beschreven volgens het “Given – When – Then”-formaat (zoals in Behavior-Driven Development testen).

**Scenario 1 – Inloggen, account aanmaken**

* Given: De gebruiker bevindt zich op het inlogscherm.
* When: De gebruiker voert een geldige gebruikersnaam en wachtwoord in en klikt op “Login”.
* Then: Het systeem controleert de gegevens, logt de gebruiker in en toont het dashboard.

**Scenario 2 – Typo-oefening uitvoeren**

* Given: De gebruiker bevindt zich op het dashboard.
* When: De gebruiker selecteert een oefening en begint te typen.
* Then: De timer start automatisch, fouten worden gemarkeerd en na afloop wordt een resultatenoverzicht getoond.

**Scenario 3 – Foutenanalyse**

* Given: De gebruiker heeft een oefening afgerond.
* When: De resultatenpagina wordt getoond.
* Then: De app markeert de gemaakte fouten in rood en toont statistieken over snelheid en accuraatheid.

**Scenario 4 – Adaptieve oefening starten**

* Given: De gebruiker heeft meerdere oefeningen voltooid.
* When: De app detecteert veelgemaakte fouten en genereert nieuw oefenmateriaal.
* Then: Het oefenmateriaal bevat meer woorden of letters die eerder verkeerd getypt werden.

# Functionele en non-functionele requirements

De volgende tabellen geven de gevalideerde functionele (FR) requirements en non-functionele (NFR) requirements weer, gebaseerd op de requirementsanalyse.

|  |  |  |
| --- | --- | --- |
| ID | Beschrijving | Prioriteit (MoSCoW) |
| FR1 | De gebruiker kan inloggen met gebruikersnaam en wachtwoord. | Must |
| FR2 | De gebruiker kan een oefening starten en fouten worden gemarkeerd. | Must |
| FR3 | Resultaten worden opgeslagen en weergegeven in grafiekvorm. | Should |
| NFR1 | Reactietijd van het systeem < 2 seconden bij laden oefeningen. | Should |
| NFR2 | Dataopslag is AVG-conform en veilig. (AVG - Algemene Verordening Gegevensbescherming) | Must |

# Definition of Done

- Functionaliteit is getest en goedgekeurd door minimaal één reviewer.

- Code bevat geen kritieke bugs en is gekoppeld aan een issue in GitHub/Azure.

- Documentatie is bijgewerkt en gevalideerd door de product owner.

- Alle requirements uit Sprint 0 zijn geïmplementeerd of gepland voor Sprint 1.

# Use cases (UC)

Dit hoofdstuk beschrijft de use cases vanuit functioneel perspectief.

## UC1 Inloggen

|  |  |
| --- | --- |
| Use case naam | Inloggen |
| Actoren | Gebruiker |
| Beschrijving | Toegang krijgen tot het persoonlijke dashboard en opgeslagen resultaten. |
| Precondities | De gebruiker heeft een geregistreerd account. |
| Hoofdscenario | 1. Gebruiker opent de app.  2. Gebruiker voert gebruikersnaam en wachtwoord in.  3. Systeem valideert de gegevens.  4. Dashboard met voortgang wordt getoond. |
| Alternatieve scenario’s | 1a. Onjuiste login → foutmelding “Ongeldige gegevens”.  1b. Wachtwoord vergeten → gebruiker kiest “Wachtwoord herstellen”. |
| Postcondities | De gebruiker is ingelogd en heeft toegang tot het persoonlijke profiel. |

Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.De grafische implementatie van het wireframe vrij is in te vullen door de developer.

Figuur 3 Wireframe inloggen TypoTrainer Figuur 4 Wireframe Dashboard TypoTrainer

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC2 Account aanmaken

|  |  |
| --- | --- |
| Use case naam | Account aanmaken |
| Actoren | Gebruiker (nieuwe student/docent) |
| Beschrijving | Een nieuwe gebruiker maakt een persoonlijk account aan om toegang te krijgen tot TypoTrainer. Na registratie ontvangt de gebruiker (optioneel) een bevestigingsmail voordat hij of zij kan inloggen. |
| Precondities | * De gebruiker heeft nog geen geregistreerd account. * De gebruiker bevindt zich op het inlog- of registratiescherm. |
| Hoofdscenario | 1. De gebruiker opent het inlog-/registratiescherm. 2. Het systeem toont het registratievelden: e-mail, wachtwoord. 3. De gebruiker voert geldige gegevens in. 4. De gebruiker kiest voor “Account aanmaken”. 5. Het systeem valideert de ingevoerde gegevens. 6. Het systeem maakt een nieuw account aan en slaat de gegevens beveiligd op. 7. Het systeem toont de melding: “Account succesvol aangemaakt.” 8. De gebruiker kan nu inloggen met de nieuwe gegevens. |
| Alternatieve scenario’s | 2a. De gebruiker heeft al een account  → Het systeem toont een melding: “Er bestaat al een account met dit e-mailadres.”  4a. Onvolledige of ongeldige invoer  → Het systeem toont een foutmelding en vraagt de gebruiker om de gegevens te controleren. |
| Postcondities | * Het nieuwe account is succesvol aangemaakt en opgeslagen in de database. * De gebruiker kan inloggen met de geregistreerde gegevens. |

De grafische implementatie van het wireframe vrij is in te vullen door de developer.

Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC3 Typo-oefening starten

|  |  |
| --- | --- |
| Use case naam | Typo-oefening starten |
| Actoren | Gebruiker |
| Beschrijving | Een oefening selecteren en uitvoeren. |
| Precondities | Gebruiker is ingelogd en er is oefenmateriaal beschikbaar. |
| Hoofdscenario | 1. Gebruiker kiest een oefening.  2. Systeem toont de voorbeeldtekst.  3. Timer start bij eerste toetsaanslag.  4. Fouten worden real-time gemarkeerd. |
| Alternatieve scenario’s | 2a. Oefening niet beschikbaar → foutmelding.  4a. Gebruiker pauzeert oefening → systeem slaat voortgang tijdelijk op. |
| Postcondities | Oefening is voltooid en resultaten worden opgeslagen. |

Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.De grafische implementatie van het wireframe vrij is in te vullen door de developer.

Figuur 5 Wireframe Typo Oefening starten

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC4 Resultaten tonen

|  |  |
| --- | --- |
| Use case naam | Resultaten tonen |
| Actoren | Gebruiker |
| Beschrijving | Inzicht krijgen in snelheid, fouten en verbeterpunten. |
| Precondities | Er is een oefening afgerond. |
| Hoofdscenario | 1. Gebruiker voltooit oefening.  2. Systeem berekent snelheid (woorden per minuut) en foutpercentage.  3. Fouten worden in rood gemarkeerd in de tekst.  4. Grafieken met voortgang worden getoond. |
| Alternatieve scenario’s | 3a. Geen resultaten beschikbaar → melding “Nog geen data”. |
| *Postcondities* | *Resultaten zijn opgeslagen en zichtbaar voor gebruiker of docent.* |

Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.

*Figuur 6 Wireframe Resultaten tonen*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## Afbeelding met tekst, schermopname, software, Computerpictogram Door AI gegenereerde inhoud is mogelijk onjuist.UC5 Adaptieve oefening genereren

|  |  |
| --- | --- |
| **Use case naam** | **Adaptieve oefening genereren** |
| Actoren | Systeem |
| Beschrijving | Gepersonaliseerde oefening aanbieden op basis van foutanalyse. |
| Precondities | Minimaal drie eerdere resultaten beschikbaar. |
| Hoofdscenario | 1. Systeem analyseert fouten en letters met hoge foutfrequentie.  2. Op basis van analyse wordt een nieuwe oefening samengesteld.  3. Gebruiker krijgt melding “Aanbevolen oefening klaar”. |
| Alternatieve scenario’s | 1a. Onvoldoende data → systeem biedt standaard oefening aan. |
| Postcondities | Nieuwe oefening is toegevoegd aan de beschikbare lijst. |

*Figuur 7 Wireframe Adaptieve oefening genereren*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.

## UC6 Voortgang monitoren

|  |  |
| --- | --- |
| **Use case naam** | **Voortgang monitoren** |
| Actoren | Gebruiker, Docent |
| Beschrijving | Overzicht krijgen van leerontwikkeling over tijd. |
| Precondities | Gebruiker heeft minstens één oefening afgerond. |
| Hoofdscenario | 1. Gebruiker opent “Voortgang”.  2. Systeem haalt resultaten op uit database.  3. Resultaten worden in grafiekvorm getoond (WPM – woorden per minuut, foutenpercentage). |
| Alternatieve scenario’s | 2a. Geen resultaten → melding “Nog geen oefenresultaten beschikbaar”. |
| Postcondities | Voortgangsgegevens worden weergegeven en kunnen geëxporteerd worden. |

Afbeelding met tekst, schermopname, software, Computerpictogram

Door AI gegenereerde inhoud is mogelijk onjuist.*Figuur 8 Wireframe Gebruikersvoortgang bijhouden*

### Evil user stories

Geen evil user stories kunnen bepalen. Geen nadere actie nodig.