|  |
| --- |
| **Parola**  I-OSSE OOAD 2023/2024 S1  SRS document |
| Jochem  Sn: 1656142  Osama Halabi  Sn: 1628160  Docent: Michel Koolwaaij  15-10-2023  v. 3.09 |

Inhoudsopgave

[1 Introduction 3](#_Toc149204048)

[1.1 Overall Description 3](#_Toc149204049)

[1.2. Definitions, acronyms, and abbreviations 3](#_Toc149204050)

[1.2 User Classes and Characteristics 4](#_Toc149204051)

[1.3 Operating Environment 4](#_Toc149204052)

[1.3.1 Hardware 4](#_Toc149204053)

[1.3.2 Operating system 4](#_Toc149204054)

[1.3.3 Software components/applications 4](#_Toc149204055)

[1.4 Design and Implementation Constraints 5](#_Toc149204056)

[1.5 Product Functions 6](#_Toc149204057)

[1.5.1 Use case diagram 6](#_Toc149204058)

[1.5.2 Brief Use case beschrijving 7](#_Toc149204059)

[2 Domain Model 9](#_Toc149204060)

[3 Use-case Descriptions 11](#_Toc149204061)

[3.1 Use-case model 11](#_Toc149204062)

[3.2 Use case beschrijvingen 11](#_Toc149204063)

[3.2.1 UC1Speler Registreren 11](#_Toc149204064)

[3.2.2 UC2 Quiz starten 13](#_Toc149204065)

[3.2.3 Vraag beantwoorden 15](#_Toc149204066)

[3.2.4 Woord controleren 17](#_Toc149204067)

[3.2.5 Credits kopen 18](#_Toc149204068)

[3.3 Inloggen (Voor Spelers) 19](#_Toc149204069)

[3.4 Inloggen (MEC Medewerker) 20](#_Toc149204070)

[4 Other functional requirements 21](#_Toc149204071)

[5 Non-functional Requirements 22](#_Toc149204072)

# Introduction

## Overall Description

De opdracht **Parola** is een quiz-spel programmeren voor MEC Games. Met het oog op de huidige populariteit van pubquizzes en de behoefte aan een robuuste, flexibele en multi-device ondersteunde applicatie, heeft MEC Games besloten om Parola te ontwikkelen. Deze applicatie wordt ontworpen om beschikbaar te zijn op de meest gangbare devices, waardoor gebruikers op elk gewenst moment en vanaf elke locatie kunnen deelnemen aan deze quizzes.

Met Parola beoogt MEC Games de volgende voordelen en doelstellingen te bereiken:

* Breed Toegankelijk:
  + Door de applicatie beschikbaar te maken op de meest gangbare devices, kunnen meer mensen deelnemen en genieten van de quizzen die door de applicatie worden aangeboden.
* Flexibiliteit:
  + Door de applicatie te ontwerpen volgens de richtlijnen van Larman en met een nadruk op objectgeoriënteerde analyse, zal Parola flexibel en aanpasbaar zijn, wat toekomstige uitbreidingen en aanpassingen vergemakkelijkt.

Het doel van dit Software Requirement Specification (SRS) document is om de eindgebruikers en het domein van de Parola-applicatie in kaart te brengen en de functionaliteiten van de eindapplicatie zo vroeg mogelijk vast te leggen. Dit zal gebeuren via een usecase-diagram, gevolgd door uitgebreide beschrijvingen van de usecases. Elk van deze usecases zal worden verduidelijkt met een system sequence diagram om de flow van de operaties te illustreren. Het overkoepelende doel van dit SRS-document is om een helder en volledig beeld te schetsen van de functionaliteiten die Parola zal bevatten en hoe gebruikers met de applicatie zullen interacteren. Bestemt voor ontwikkelaars, testers, projectmanagers en andere belanghebbenden, dit SRS-document biedt inzichten en views om alle stakeholders een grondig begrip van de onderliggende architectuur en functionaliteiten van Parola te geven.

## Definitions, acronyms, and abbreviations

Het doel van dit hoofdstuk is om een ​​duidelijke en consistente terminologie aan te geven binnen de casus, wat verwarring en misverstanden voorkomt.

| **Term** | **Definition** |
| --- | --- |
| UC | Use Case |
| NFR | Non-functional Requirements |
| FR | Functional Requirements |

Tabel 1: Definitions, acronyms, and abbreviations

## User Classes and Characteristics

| **Actor** | **Omschrijving** |
| --- | --- |
| Speler | Een individuele gebruiker die deelneemt aan de quiz. |
| Medewerker van MEC Games | Een medewerker (of administrator) die verantwoordelijk is voor het beheren van de vragenlijst binnen de Parola-applicatie |
| Externe Software Library | Een library die door de Parola-applicatie wordt gebruikt om het eind ingevoerde woord te controleren. |
| Betaalsysteem | Een extern systeem waarmee gebruikers credits kunnen kopen voor gebruik binnen de Parola applicatie |

Tabel 2: User Classes and Characteristics

## Operating Environment

### Hardware

Doordat het systeem op veel verschillende systemen moet kunnen werken, is het niet mogelijk om de benodigde hardware vast te stellen.

### Operating system

Parola werkt op Windows, MacOS en Linux

### Software components/applications

De enige vereiste is dat Java, met name versie 17 of nieuwer.

## Design and Implementation Constraints

|  |  |  |
| --- | --- | --- |
| **Limitatie** | **Beschrijving** | **Bron** |
| Parola Data | De data voor deze applicatie mag gemockt worden. | Case Study Parola OOSE-OOAD 2023-2024 P1 |
| Betaling systeem | Om credits bij te kopen moet de speler een vaste prijs betalen voor het aantal credits dat hij wil kopen, hiervoor wordt ook gebruik maken een extern betalingssysteem. Het externe betalingssysteem hoeven we niet te implementeren | Gesprek met de opdrachtgegever |
| Controle woord | Om te controleren of het door de speler gevormde woord ook een bestaand woord is zal gebruik worden gemaakt van een externe software library. | Case Study Parola OOSE-OOAD 2023-2024 P1 |

## Product Functions

### Use case diagram

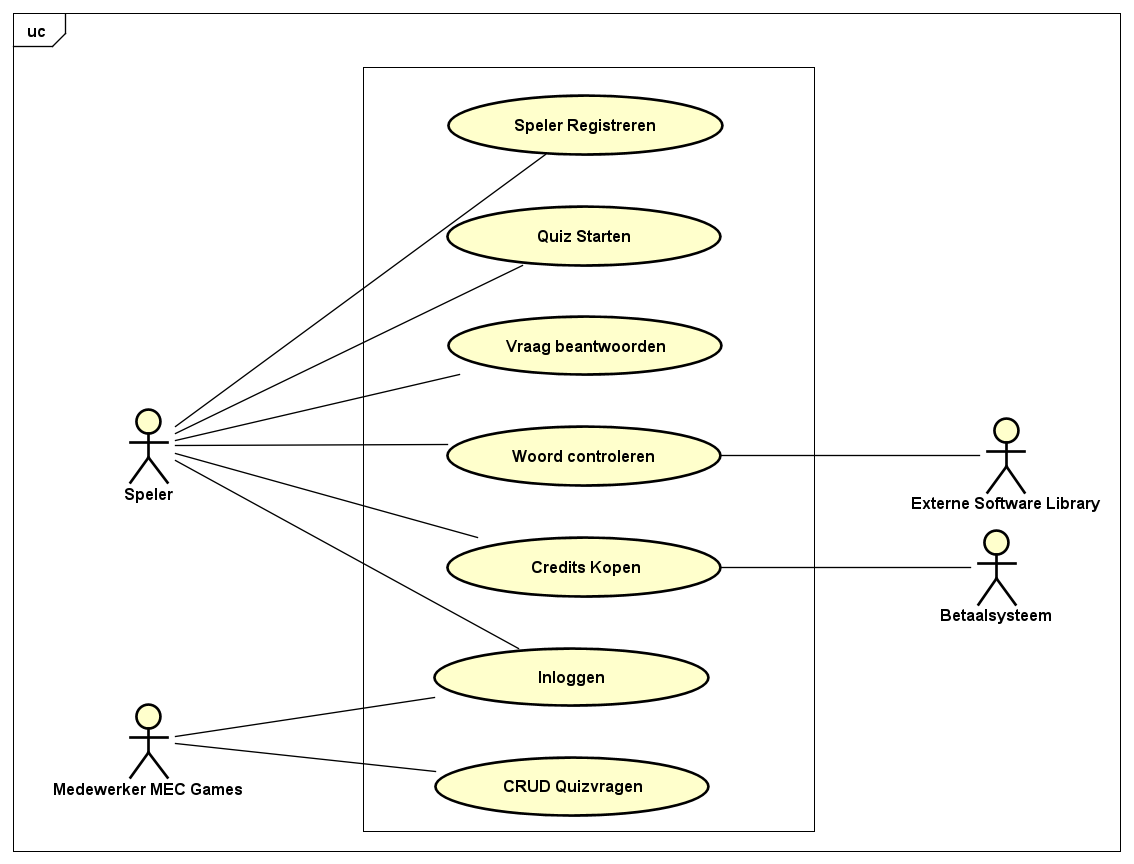


Figure 1 Use Case Diagram Parola

### Brief Use case beschrijving

Dit hoofdstuk bevat korte beschrijvingen van alle Use Cases die in figuur 1 worden weergegeven.

|  |  |
| --- | --- |
| **UC** | Speler Registreren |
| **Actor** | Speler |
| **Beschrijving** | Een nieuwe speler maakt een account aan door zijn gegevens in te voeren. Het systeem slaat deze gegevens op. |

|  |  |
| --- | --- |
| **UC** | Quiz starten |
| **Actor** | Speler |
| **Beschrijving** | De speler selecteert een quiz om te spelen. Het systeem maakt een quiz aan en toont de eerste vraag voor de speler. |

|  |  |
| --- | --- |
| **UC** | Vraag beantwoorden |
| **Actor** | Speler |
| **Beschrijving** | Nadat de speler een vraag heeft ontvangen, voert de speler een antwoord in. Het systeem valideert het antwoord en geeft de speler een letter van het magic woord als het antwoord correct is. |

|  |  |
| --- | --- |
| **UC** | Woord controleren |
| **Actor** | Speler |
| **Beschrijving** | Na het voltooien van de quiz, probeert de speler een woord te vormen van alle letters die hij ontvangen heeft voor elke goed antwoord. Het systeem valideert het woord door gebruik te maken van een externe library. |

|  |  |
| --- | --- |
| **UC** | Credits kopen |
| **Actor** | Speler |
| **Beschrijving** | De speler kan credits kopen volgens de gespecificeerde tarieven. |

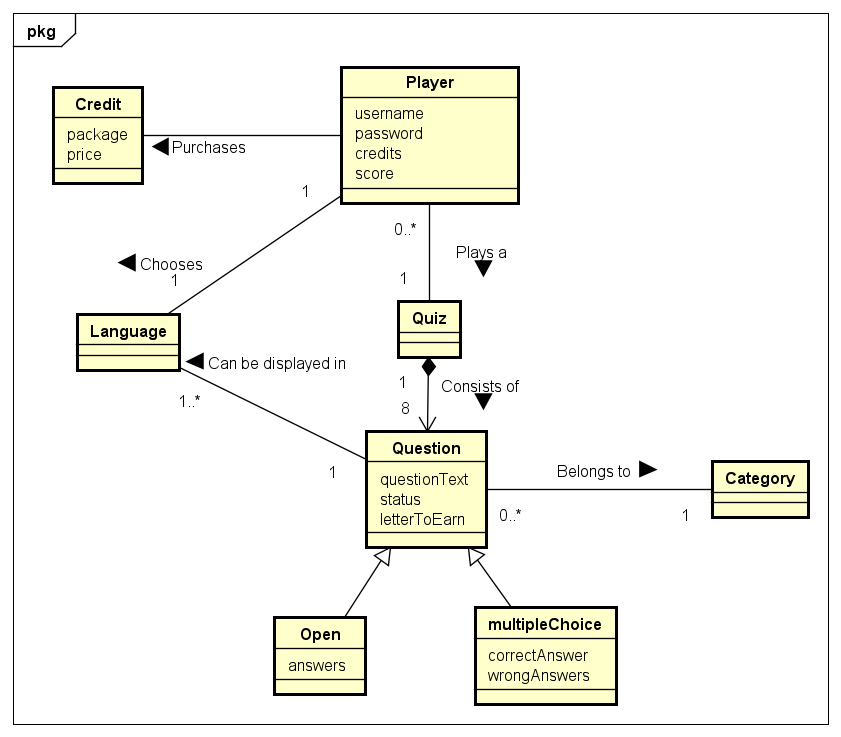
|  |  |
| --- | --- |
| **UC** | Inloggen |
| **Actor** | Speler |
| **Beschrijving** | De speler voert zijn/haar inloggegevens in (zoals gebruikersnaam en wachtwoord) om toegang te krijgen tot zijn/haar account. |

|  |  |
| --- | --- |
| **UC** | Inloggen |
| **Actor** | MEC Medewerkers |
| **Beschrijving** | De medewerker voert specifieke admin-inloggegevens in om toegang te krijgen tot het beheerdersgedeelte van de applicatie. |

|  |  |
| --- | --- |
| **UC** | CRUD Quizvragen |
| **Actor** | MEC Medewerkers |
| **Beschrijving** | De medewerker kan de vragenlijst beheren, dit bestaat uit het toevoegen, wijzigen, of verwijderen van vragen. |

# Domain Model

Het domeinmodel biedt een overzicht van de structurele associaties tussen de diverse concepten binnen het domein. Dit model wordt gevolgd door omschrijvingen van de gebruikte concepten en attributen in het model.



Figuur 2 Domain model

Glossary domeinmodel:

|  |  |
| --- | --- |
| **Domain** | **Description** |
| magicWord | Het woord waarvan de speler een letter voor elke goed antwoord krijgt. |
| Categorie | Elke vraag behoort tot een categorie |
| Status | De status van een vraag, deze is actief of niet-actief |
| Taal | In welke taal Parola gespeeld wordt. |
| Player | Een persoon die een quiz wil spelen. |
| Credit | Het saldo dat de speler heeft om te gebruiken in het spel. |
| Score | De score die de speler heeft verdiend. |
| Open | Een vraag met een kort antwoord. Elke open vraag heeft meerdere antwoorden die de speler kan invoeren, dit betreft meestal verschillende manieren om hetzelfde antwoord te geven (bijvoorbeeld: AUB of A.U.B). |
| multipleChoice | Een Meerkeuze vraag waar maar een antwoord juist is. |
| letterToEarn | De letter die de speler verdient als hij de vraag goed heeft beantwoord. |

# Use-case Descriptions

## Use-case model

Het use-case model is te vinden in het hoofdstuk [Product Functions](#_Use_case_diagram).

## Use case beschrijvingen

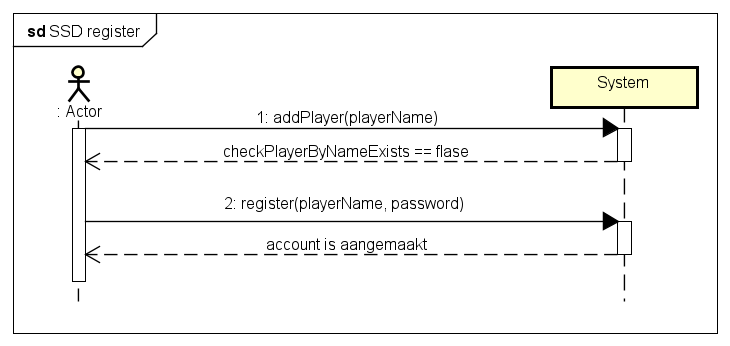
In dit hoofdstuk worden de use cases fully dressed beschreven. Hier worden verschillende happy en alternatives flows beschreven per use case. Daarnaast worden de use cases eventueel ondersteund met system sequence diagrammen.

### **UC1** Speler Registreren

| **Use case ID** | **UC1** |
| --- | --- |
| **Naam** | Speler Registreren |
| **Beschrijving** | Een nieuwe speler maakt een account aan door zijn gegevens in te voeren. Het systeem slaat deze gegevens op. |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler heeft nog geen account aangemaakt. |
| **Post conditie** | * De speler heeft een account aangemaakt en kan een quiz starten. |
| **Happy flow: De gebruikersnaam bestaat niet** | |
| **Actor** | **Systeem** |
| 1- De speler voert zijn\haar de gebruikersnaam in. | 2- Het systeem checkt of de speler bestaat gebruikersnaam. |
|  | 3- Het systeem vraagt de gebruiker om zijn\haar wachtwoord in te voeren |
| 1- De speler voert zijn\haar wachtwoord in. | 4- Het systeem geeft een keuzemenu terug. |
| **Alternative flow: validatiefout optreedt (bijv. de gebruikersnaam bestaat wel).** | |
|  | 3a- Het systeem met **UC6** inloggen |

###### Tabel 1: UC 1 Speler Registreren fully dressed format

#### System sequence diagram **UC1** Speler Registreren



### UC2 Quiz starten

| **Use case ID** | **UC2** |
| --- | --- |
| **Naam** | Quiz starten |
| **Beschrijving** | De speler selecteert een quiz om te spelen. Het systeem presenteert vervolgens een reeks vragen aan de speler. |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler is ingelogd en heeft een hoofdmenu voor zich. |
| **Post conditie** | * De quiz is gestart. * Het systeem toont de eerste vraag van de quiz aan. |
| **Happy flow: Speler heeft voldoende credits** | |
| **Actor** | **Systeem** |
| 1- De speler kiest de optie "Start Quiz". | 2- Het systeem controleert of de speler voldoende credits heeft om deel te nemen. |
|  | 3- Het systeem de quiz en toont de eerste vraag in de console. |
| **Alternative flow: De speler heeft onvoldoende credits.** | |
|  | 3a- Het systeem toont een melding over onvoldoende credits in de console. |
|  | 4a- Het systeem geeft de speler de mogelijkheid om credits bij te kopen. |

###### Tabel 2: UC 2 Quiz spelen fully dressed format

#### System sequence diagram **UC2** Quiz starten

A screenshot of a quiz

Description automatically generated

### Vraag beantwoorden

| **Use case ID** | **UC3** |
| --- | --- |
| **Naam** | Vraag beantwoorden |
| **Beschrijving** | Nadat de speler een vraag heeft ontvangen, geeft/kiest hij/zij een antwoord uit de gegeven opties. |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler heeft de quiz gestaart. |
| **Post conditie** | * De speler heeft de vraag beantwoord. |
| **Happy flow:** **1-open vraag**  **2- Correct antwoord** | |
| **Actor** | **Systeem** |
|  | 1-Het systeem toont een open vraag. |
| 2-De speler voert zijn/haar antwoord in. | 3-Het systeem checkt het ingevoerde antwoord. |
|  | 4- Het systeem toont de verdiende letter aan. |
|  | 5- Het systeem toont de volgende vraag |
| **Happy flow:** **1- multipleChoice vraag**  **2- Correct antwoord** | |
|  | 1a-Het systeem toont een meerkeuze vraag. |
| 2a-De speler kiest zijn/haar antwoord van het keuzemenu. | 3a-Het systeem checkt het gekozen antwoord. |
|  | 4a- Het systeem toont de verdiende letter aan. |
|  | 5a- Het systeem toont de volgende vraag |
| **Alternative flow: fout aantwoord** | |
|  | 4b-Het systeem toont de volgende vraag. |

###### Tabel 3: UC 3 Vraag beantwoorden fully dressed format

#### System sequence diagram **UC3** vraag beantwoorden.

A screenshot of a computer

Description automatically generated

### Woord controleren

| **Use case ID** | **UC4** |
| --- | --- |
| **Naam** | Woord controleren |
| **Beschrijving** | Na het voltooien van de quiz, probeert de speler een woord te vormen van alle letters die hij ontvangen heeft voor elke goed antwoord. Het systeem valideert het woord door gebruik te maken van een externe library. |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler de quiz beëindigt. * De speler ziet alle verdiende letters voor zich. |
| **Post conditie** | * De speler krijgt ziet zijn score voor zich. |
| **Happy flow:** | |
| **Actor** | **Systeem** |
| 1-De voert zijn/haar woord in. | 2-Het systeem checkt alle letters van het woord op basis van alle verdiende letters. |
|  | 3- Het systeem checkt of het ingevoerde woord bestaat. |
|  | 4- Het systeem toont de score voor de speler. |
| **Alternative flow: Het ingevoerde woord bestaat niet uit alle verdiende letters.** | |
|  | 4a- Het systeem geeft een foutmelding. |
| **Alternative flow: Het ingevoerde woord bestaat niet.** | |
|  | 4a- het systeem geeft aan dat het woord bestaat niet uit. |

###### Tabel 4: UC 4 Woord controleren fully dressed format

### Credits kopen

| **Use case ID** | **UC5** |
| --- | --- |
| **Naam** | Credits kopen |
| **Beschrijving** | De speler kan credits kopen volgens de gespecificeerde tarieven |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler heeft credits kopen uit het keuzemenu. |
| **Post conditie** | * Het Saldo van de speler is gelijk aan het oude saldo plus het aantal gekochte credits. |
| **Happy flow: betaling gelukt** | |
| **Actor** | **Systeem** |
| 1- De speler kiest de optie "Credits kopen". | 2- Het systeem toont verschillende pakketten met credits en de bijbehorende kosten. |
| 3- De speler kiest een pakket. | 4- Het systeem instrueert de speler over de betalingsprocedure. |
|  | 5- het systeem voegt de gekochte credits toe aan het account van de speler. |
| **Alternative flow: betaling mislukt** | |
|  | 5a- het systeem toont een melding dat de betaling is misgelukt |

###### Tabel 5: UC5 Credits kopen fully dressed format

## Inloggen (Voor Spelers)

| **Use case ID** | **UC6** |
| --- | --- |
| **Naam** | Inloggen |
| **Beschrijving** | De speler voert zijn/haar inloggegevens in (zoals gebruikersnaam en wachtwoord) om toegang te krijgen tot zijn/haar account. |
| **Actor(s)** | Speler |
| **Pre conditie** | * De speler heeft een geldig account. |
| **Post conditie** | * De speler is ingelogd. * Het systeem toont een keuzemenu aan. |
| **Happy flow: 1-De gebruikersnaam bestaat wel**  **2- het wachtwoord is correct** | |
| **Actor** | **Systeem** |
| 1- De speler voert zijn/haar gebruikersnaam. | 2- Het systeem checkt of de speler bestaat gebruikersnaam. |
|  | 3- Het systeem vraagt de gebruiker om zijn\haar wachtwoord in te voeren. |
| 4- De speler voert zijn/haar wachtwoord | 5- Het systeem controleert het wacntwoord |
|  | 6- Het systeem geeft een keuzemenu terug. |
| **Alternative flow:** **Als de gebruikersnaam bestaat niet.** | |
|  | 6a- Hier gaat het systeem verder met **UC1**. |
| **Alternative flow:** **Als het wachtwoord niet correct.** | |
|  | 6b- het systeem voert het deze use case opnieuw uit. |

###### Tabel 6: UC6 Inloggen fully dressed format

## Inloggen (MEC Medewerker)

| **Use case ID** | **UC6** |
| --- | --- |
| **Naam** | Inloggen |
| **Beschrijving** | De medewerker voert specifieke admin-inloggegevens in om toegang te krijgen tot het beheerdersgedeelte van de applicatie. |
| **Actor(s)** | MEC Medewerker |
| **Pre conditie** | * De medewerker heeft toestemming en de benodigde gegevens om in te loggen. |
| **Post conditie** | * De medewerker heeft toegang tot het beheerderspaneel. |
| **Happy flow: de gebruikersnaam bestaat wel** | |
| **Actor** | **Systeem** |
|  | 1- Het systeem vraagt de medewerker naar zijn/haar inloggegevens |
| 2- De medewerker voert zijn/haar gebruikersnaam en wachtwoord in. | 3- Het systeem valideert de inloggegevens. |
|  | 4- Bij correcte gegevens krijgt de medewerker toegang tot het beheerderspaneel van het spel. |
| **Alternative flow:** **Als de inloggegevens onjuist zijn.** | |
|  | 4a- Het systeem geeft een foutmelding aan in de console. |
| 5a- De medewerker voert opnieuw zijn/haar gebruikersnaam en wachtwoord in. |  |

###### Tabel 7: UC6 Inloggen fully dressed format

# Other functional requirements

De Other functional Requirements zijn de requirements waaraan het systeem moet voldoen die niet direct onder Use cases vallen of een specifieke actie nodig hebben om behaald te worden.

|  |  |
| --- | --- |
| Code | Description |
| FR1 | Het system moet gebruik maken van een externe software library om het ingevoerde woord te controleren. |

# Non-functional Requirements

De Non-functional Requirements zijn de requirements waaraan het systeem moet voldoen die niet direct onder Use cases vallen of een specifieke actie nodig hebben om behaald te worden. Deze requirements worden hieronder onderverdeeld volgens het principe van FURPS+.

|  |  |  |
| --- | --- | --- |
| Code | Description | Type |
| NFR1 | Tijdens een quiz, moet het systeem binnen 0.5 seconden de volgende vraag tonen vanaf het moment dat de speler de huidige vraag heeft beantwoord. | Performance |
| NFR2 | Het overstappen naar een alternatief – al bestaand – systeem voor puntentelling, moet niet langer duren dan 10 minuten. | Useability |
| NFR3 | Het systeem moet de mogelijkheid voor andere talen ondersteunen | Supportability |