Software Requirements Specification

Author: Lodewijk Klassen & Job Nieling

Version: 1 (20-03-2024)

[1 Introduction 2](#_Toc162447833)

[1.1 Algemene Beschrijving 2](#_Toc162447834)

[1.2 Gebruikersklassen en Kenmerken 2](#_Toc162447835)

[1.3 Operationele Omgeving 2](#_Toc162447836)

[1.4 Ontwerp- en Implementatiebeperkingen 3](#_Toc162447837)

[1.5 Product Functies 3](#_Toc162447838)

[*2* *Domain Model* 4](#_Toc162447839)

[*3* Use-case Descriptions 6](#_Toc162447840)

[3.1 Registreren 6](#_Toc162447841)

[3.1.1 Fully-dressed use case description 6](#_Toc162447842)

[3.1.2 System Sequence Diagram 6](#_Toc162447843)

[3.2 Inloggen 7](#_Toc162447844)

[3.2.1 Fully-dressed use case description 7](#_Toc162447845)

[3.2.2 System Sequence Diagram 7](#_Toc162447846)

[3.3 Spelen Quiz 8](#_Toc162447847)

[3.3.1 Fully-dressed use case description 8](#_Toc162447848)

[3.3.2 System Sequence Diagram 9](#_Toc162447849)

[3.3.3 Operation Contracts voor "Spelen Quiz" 10](#_Toc162447850)

[3.4 Bijkopen credits 11](#_Toc162447851)

[3.4.1 Fully-dressed use case description 11](#_Toc162447852)

[3.4.2 System Sequence Diagram 11](#_Toc162447853)

[3.4.3 Operation Contracts voor "Bijkopen Credits" 12](#_Toc162447854)

[4 Other functional requirements 12](#_Toc162447855)

[5 Non-functional Requirements 13](#_Toc162447856)

[5.1 Functionaliteit 13](#_Toc162447857)

[5.2 Betrouwbaarheid 13](#_Toc162447858)

[5.3 Gebruiksvriendelijkheid 13](#_Toc162447859)

[5.4 Prestatie 13](#_Toc162447860)

[5.5 Ondersteunbaarheid 13](#_Toc162447861)

# Introduction

## Algemene Beschrijving

Parola is een quiz-applicatie ontwikkeld door MEC Games, bedoeld om beschikbaar te zijn op verschillende gangbare apparaten. Het doel van de software is om gebruikers een leuke en educatieve quizervaring te bieden. De applicatie streeft ernaar om quizzen van verschillende categorieën aan te bieden, waarbij spelers punten kunnen verdienen door vragen correct te beantwoorden en woorden te vormen.

## Gebruikersklassen en Kenmerken

**Gebruiker**:  
Spelers hebben de mogelijkheid om quizzen te kunnen spelen, ze kunnen credits kopen. Bij elke quiz dient de spelen aan het einde een woord te maken met de gekregen letters.

**Bezoeker:**Bezoekers kunnen zich aanmelden of inloggen, als de gebruiker zich inlogt of registreert wordt deze een Gebruiker.

**Medewerker:**

De medewerker kan de quizzen en vragen aanmaken, quizzen bewerken en vragen op non-actief zetten.

## Operationele Omgeving

Parola wordt gespeeld op een computer met een command prompt interface. Het spel draait op het Java-platform. Je kan zit spel spelen op elke computer met een werkende IDE. Wij hebben ervoor gekozen op dit project te maken met IntelliJ en dit zal dus ondersteund worden.

## Ontwerp- en Implementatiebeperkingen

**Taal en Technologie:**

Parola zal worden geïmplementeerd met Java voor de logica, en zal gebruik maken van UML voor objectgeoriënteerde analyse en ontwerp.

**Communicatie:**

Voor dit project zullen we de code schrijven in het Engels. De vragen, gebruikersinterface en documentatie zullen wel in het Nederlands worden opgeleverd.

**Hardcoded:**De data (spelers en quizzen) zit hard coded in de opgeleverde applicatie. We maken geen gebruik van een Database.

**Externe library:**

Voor de woorden maken wordt er gebruik gemaakt van een externe library die wij niet zullen opnemen in onze code.

**Tijd:**

We hebben een beperkte tijd om de applicatie te ontwerpen en te implementeren, dus efficiëntie is cruciaal.

## Product Functies

***De belangrijkste functies van Parola zijn:***

**Registreren en inloggen:**

Bezoekers moeten zich kunnen registreren en inloggen om toegang te krijgen tot de quizzen.

**Quizzen spelen:**

Gebruikers moeten in staat zijn om quizzen te selecteren en te spelen, waarbij ze punten en letters verdienen door vragen correct te beantwoorden. Daarnaast wordt er gevraagd van de letters een woord te maken. Na dat de quiz is gespeeld en het woord is gemaakt moet de applicatie punten toekennen aan spelers op basis van het aantal correct beantwoorde vragen, de lengte van gevormde woorden en de speeltijd.

**Beheer van quizzen en vragen**:

Beheerders moeten in staat zijn om quizzen en vragen te beheren, waaronder het toevoegen, bewerken, verwijderen, en vervangen ervan.

**Credits bijkopen:**

Spelers moeten de mogelijkheid hebben om credits bij te kopen voor het spelen van quizzen.

# *Domain Model*

A diagram of a computer

Description automatically generated

1. **Parola**:
   * Parola heeft een verzameling van Quizzen en Spelers. Parola kent de Spelers en de Quizzen maar andersom kennen ze elkaar niet.
2. **Speler**:
   * De klasse Speler staat voor een speler van de quizapplicatie. Het bevat de volgende informatie van de spelers, hun gebruikersnaam, wachtwoord en het aantal credits.
3. **QuizUitvoering**:
   * QuizUitvoering is een actieve uitvoering van een quiz door een speler. Het houdt informatie van de score en de duur van de quiz.
4. **Quiz**:
   * Quiz heeft meerdere quizzen, een quiz bestaat uit een de naam en vragen. Elke quiz kan gekoppelt worden aan meerdere quizuitvoeringen.
5. **Vraag**:
   * Vraag is een abstracte klasse die dient als de basis voor verschillende soorten vragen in de quizapplicatie. Het is uitgebreid met de specifieke vraagtypen Meerkeuzevraag en KortAntwoordVraag.
6. **GegevenAntwoord**:
   * GegevenAntwoord vertegenwoordigt een antwoord dat is gegeven door een speler tijdens het spelen van een quiz. Het bevat informatie over het gegeven antwoord, de tekst van het antwoord en of het antwoord correct is.
7. **Meerkeuzevraag**:
   * Meerkeuzevraag is een specifiek type vraag waarbij de speler uit een 4 mogelijke antwoorden moet kiezen. Het bevat eigenschappen van de categorie van de vraag, de letters die worden gegeven bij een goed antwoord en of de vraag op actief staat.
8. **KortAntwoordVraag**:
   * KortAntwoordVraag is een vraag waarbij de speler een kort antwoord moet geven in plaats van te kiezen uit een set mogelijke antwoorden. Het bevat eigenschappen voor de mogelijke antwoorden.
9. **Antwoord**:
   * Antwoord is een antwoord op een vraag. Het antwoord is gekoppeld aan een specifieke vraag en bevat de antwoord tekst.

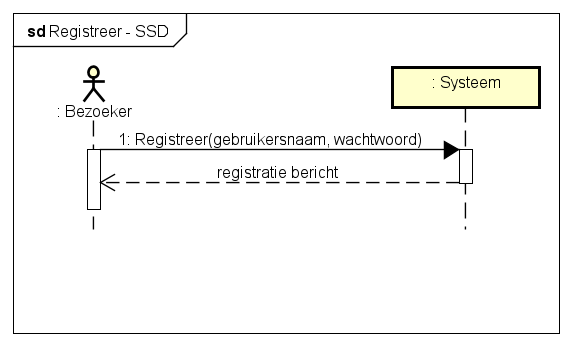
# Use-case Descriptions

## Registreren

### Fully-dressed use case description

|  |  |
| --- | --- |
| **Primary actor:** Bezoeker | |
| **Stakeholders and Interests:** Parola | |
| **Brief description:**  De bezoeker registreert zich met een gebruikersnaam en wachtwoord; het systeem maakt een account aan met deze gegevens. De bezoeker wordt door het systeem direct ingelogd door de use case Inloggen, maar zonder de gebruikersnaam en wachtwoord een tweede keer in te hoeven voeren. De bezoeker kan nu als gebruiker verder in de app. | |
| **Preconditions:** | |
| **Postconditions (Success Guarantee):**  Er is een account geregistreerd en heeft een startsaldo van 1000 credits ontvangen. De gebruiker is ingelogd met het nieuwe account. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De bezoeker navigeert naar de registratie pagina.  3. De bezoeker geeft een gebruikersnaam en wachtwoord op. | 2. Het systeem vraagt de registratie gegevens, gebruikers naam en wachtwoord.  4. Het systeem valideert en registreert een nieuwe account met een startsaldo van 1000 credits.  5. De bezoeker wordt automatisch ingelogd. |
| **Extensions (Alternative Flow):** | |
| 1b. De bezoeker geeft een invalide gebruikersnaam op. | 2b. Het systeem geeft aan dat de gebruikersnaam ongeldig is en de gebruiker mag het opnieuw proberen. |

### System Sequence Diagram

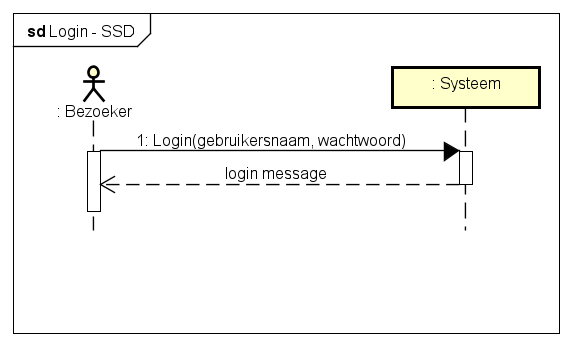


## Inloggen

### Fully-dressed use case description

|  |  |
| --- | --- |
| **Primary actor:** bezoeker | |
| **Stakeholders and Interests:** | |
| **Brief description:**  De bezoeker logt in met een gebruikersnaam en wachtwoord. Het systeem controleert deze gegevens. De bezoeker kan nu als gebruiker of als medewerker verder in de app. | |
| **Preconditions:** | |
| **Postconditions (Success Guarantee):**  De gebruiker is ingelogd en kan een quiz spelen. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De gebruiker gaat naar de login pagina.  3. De bezoeker vult een gebruikersnaam en wachtwoord in. | 2. Het systeem vraagt de Gebruikersnaam en wachtwoord van de bezoeker.  4. Het systeem controleert de gegevens en bevind dat ze goed zijn.  5. Het systeem geeft de gebruiker toegang tot het Parola-systeem. |
| **Extensions (Alternative Flow):** | |
|  | 2a. Het systeem controleert de gegevens en bevind dat deze niet goed zijn.  3a. Het systeem geeft een error melding voor verkeerde gebruikersnaam of wachtwoord. De gebruiker mag nogmaals proberen in te loggen. |
|  | |

### System Sequence Diagram

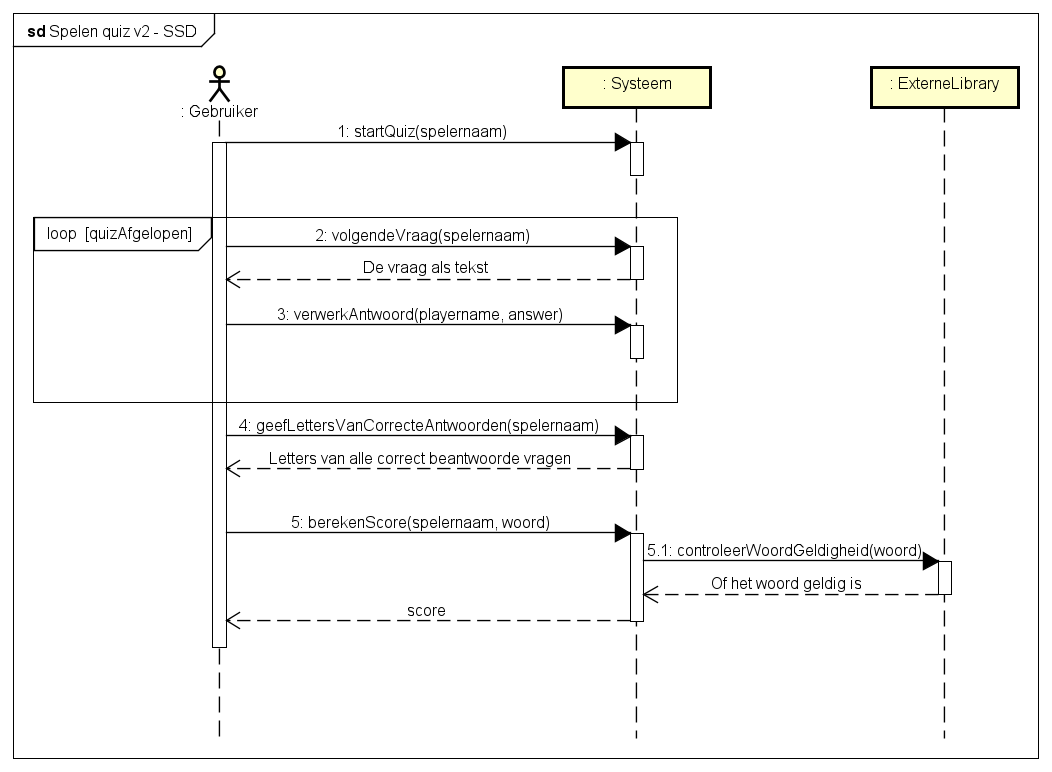


## Spelen Quiz

### Fully-dressed use case description

|  |  |
| --- | --- |
| **Primary actor:** Gebruiker | |
| **Stakeholders and Interests:** Parola | |
| **Brief description:**  De gebruiker begint een te quiz spelen door 40 credits te betalen. De gebruiker beantwoord de vragen 1 voor 1. Als alle vragen beantwoord zijn krijgt de gebruiker een letter voor elke goed beantwoorde vraag, waarmee de gebruiker een zo lang mogelijk woord vormt. Wanneer de gebruiker de quiz heeft afgerond wordt de score berekend op basis van het aantal goed beantwoorde vragen, de lengte van het gevormde woord, en de gebruikte tijd. | |
| **Preconditions:** | |
| **Postconditions (Success Guarantee):**  De gebruiker heeft een score behaald en het saldo is bijgewerkt. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De gebruiker kiest ervoor om een quiz te spelen.      LOOP:  4. De gebruiker beantwoord een vraag.  EINDE LOOP      7. De gebruiker maakt een woord met de gegeven letters. | 2. Het systeem schrijft het aantal credits af bij de gebruiker.  3. Het systeem registreert de huidige tijd als de starttijd van de quiz uitvoering.  LOOP:  5. Het systeem verwerkt het antwoord.  EINDE LOOP  6. Het systeem geeft de gebruiker voor elke goede vraag een letter.  8. Het systeem controleert het woord en ondervindt dat het woord geldig is.  9. Het systeem registreert de huidige tijd als de eindtijd van de quiz uitvoering.  10. Het systeem berekend de score op basis van de gespeelde tijd, woordlengte en goede aantal vragen. |
| **Extensions (Alternative Flow):** | |
|  | 2a. Het systeem ondervindt dat de gebruiker niet genoeg credits heeft.  EINDE USE CASE |
|  | 8b. Het externe systeem ondervindt dat het woord ongeldig is.  Vervolg use case by stap 7 |
|  | |

### System Sequence Diagram



### Operation Contracts voor "Spelen Quiz"

**Operatie: StartQuiz**

Input: Gebruikersnaam

Precondities: De gebruiker heeft voldoende credits betaald.

Postcondities: De quiz-uitvoering is gestart, de starttijd van de quiz is geregistreerd, en Credits zijn afgeschreven bij de speler.

**Operatie: VolgendeVraag**

Input: Gebruikersnaam

Preconditie: De gebruiker heeft een quiz gestart.  
Postcondities: -

**Operatie: VerwerkAntwoord**

Input: Gebruikersantwoord

Precondities: De quiz-uitvoering is gestart.

Postcondities: Er is een nieuwe instantie aangemaakt van gegevenAntwoord in de huidige quizuitvoering, met een link naar de huidige vraag. Het antwoord is door het systeem gevalideerd.

**Operatie: GeefLettersVoorCorrecteAntwoorden**

Input: Gebruikersnaam

Precondities: Alle vragen van de huidige quizuitvoering zijn beantwoord.

Postcondities: -

**Operatie: BerekenScore**

Input: Tijd gespeeld, lengte van het gevormde woord, aantal goed beantwoorde vragen

Precondities: De quiz-uitvoering is voltooid en woord is gevormd.

Postcondities: De score is berekend op basis van de gespeelde tijd, de lengte van het gevormde woord en het aantal correct beantwoorde vragen. De score is toegevoegd aan de huidige quizuitvoering.

**Operatie: ControleerWoordGeldigheid**

Input: Gebruikerswoord

Precondities: De gebruiker heeft een woord gevormd met de gegeven letters.

Postcondities: -

### Activity Diagram

We hebben een activity diagram gemaakt voor de usecase spelen quiz.

Afbeelding met tekst, schermopname, diagram, lijn

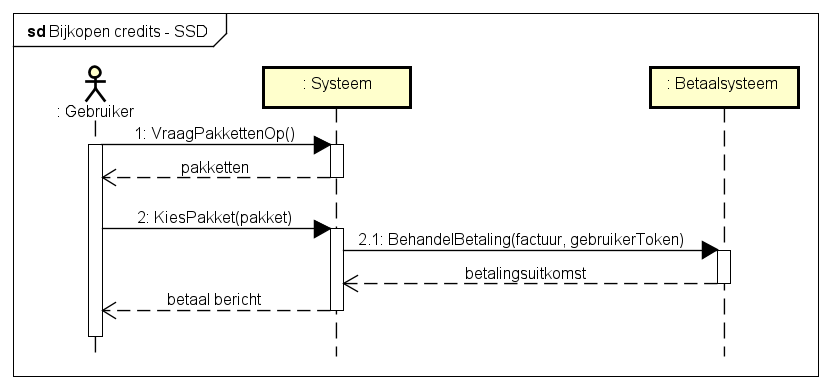
Automatisch gegenereerde beschrijving

## Bijkopen credits

### Fully-dressed use case description

|  |  |
| --- | --- |
| **Primary actor:** Speler | |
| **Stakeholders and Interests:** Betalingsgateway | |
| **Brief description:**  De gebruiker krijgt vanuit het systeem de opties om nieuwe credits te kopen en laat de verschillende pakketten zien. De gebruiker kiest het pakket wat die wil hebben. De gebruiker wordt door het systeem doorverwezen naar een extern betaalsysteem, waar de gebruiker betaalt. Het systeem keert de gekochte credits uit aan de gebruiker. | |
| **Preconditions:**  Betalingsgateway is beschikbaar, gebruiker is ingelogd. | |
| **Postconditions (Success Guarantee):**  Het saldo van de gebruiker is bijgewerkt. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De gebruiker gaat naar de webshop  3. De gebruiker selecteert het gewenste pakket.  5. De gebruiker voltooit de betaling via de betalingsgateway. | 2. Het systeem haalt de pakketten op en laat deze aan de gebruiker zien.  4. Het systeem stuurt de gebruiker door naar de betalingsgateway.  6. Het systeem krijgt een melding van het betaalsysteem of de betaling met het factuur nummer is voldaan.  7. Het systeem verandert de saldo van de gebruiker met het gekochte aantal credits en stuurt een betaal bericht terug naar de gebruiker. |
| **Extensions (Alternative Flow):** | |
| 5a. gebruiker annuleert de betaling | 6a. Het systeem krijgt de melding van het betaalsysteem dat de betaling met het factuur nummer niet is voldaan.  7a. De gebruiker krijgt een melding dat de betaling niet is geslaagd en krijgt de kans de betaling alsnog te verrichten. |
|  | |

### System Sequence Diagram



### Operation Contracts voor "Bijkopen Credits"

**Operatie: VraagPakkettenOp**

Input: -

Precondities: Gebruiker is ingelogd.

Postcondities: -

**Operatie: SelecteerCreditPakket**

Input: Gebruikersnaam, geselecteerd creditpakket

Precondities: Gebruiker is ingelogd.

Postcondities: De creditsaldo van de gebruiker is aangepast.

**Operatie: BehandelBetaling**

Input: factuur, gebruikerToken

Precondities: Gebruiker heeft een pakket gekozen

Postcondities: -

# Other functional requirements

*Quizfunctionaliteit*

|  |  |
| --- | --- |
| Code | Beschrijving |
| FR1 | De totale tijd die een gebruiker nodig heeft om een quiz te spelen moet (per quiz) worden bijgehouden. |
| FR2 | De totale tijd die een gebruiker nodig heeft om een quiz te spelen moet (per quiz) optioneel zijn voor de score-berekening. |
| FR3 | De score van de gespeelde quiz wordt opgeslagen in de quizuitvoering. |
| FR4 | Een gebruiker moet in staat zijn om een woord te vormen met de ontvangen letters na het spelen van een quiz. |

# Non-functional Requirements

## Functionaliteit

|  |  |
| --- | --- |
| Code | Beschrijving |
| NFR2 | Alle quizzen kosten 40 credits. |
| NFR4 | Het moet mogelijk zijn om in de toekomst meerdere talen te ondersteunen. |

## Betrouwbaarheid

|  |  |
| --- | --- |
| Code | Beschrijving |
| NFR5 | De wachtwoorden zijn vanaf de registratie beschermd opgeslagen. |

## Gebruiksvriendelijkheid

|  |  |
| --- | --- |
| Code | Beschrijving |
| NFR6 | Alle tekst moet in het Nederlands zijn. |

## Prestatie

|  |  |
| --- | --- |
| Code | Beschrijving |
| NFR7 | Gebruikers mogen bij het spelen van de quiz geen hinderlijke vertragingen in de applicatie ondervinden. |

## Ondersteunbaarheid

|  |  |
| --- | --- |
| Code | Beschrijving |
| NFR8 | een gebruiker kan niet minder dan 0 credits hebben. |