# Stel een use case model op dat bestaat uit een use case diagram en de use case beschrijvingen in brief format.

A diagram of a diagram

Description automatically generated

|  |  |  |
| --- | --- | --- |
| Use case | Actor | Brief description |
| Registreren | Bezoeker | De bezoeker registreert zich met een gebruikersnaam en wachtwoord. Het systeem controleert of er nog geen account bestaat met de opgegeven gebruikersnaam, en maakt een account aan aan de hand van de opgegeven gegevens. De bezoeker wordt ook door het systeem direct ingelogd, zonder de gebruikersnaam en wachtwoord een tweede keer in te hoeven voeren. De bezoeker kan nu als gebruiker verder in de app. |
| Inloggen | Bezoeker | 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. |
| Spelen quiz | Gebruiker | 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. |
| Bijkopen credits | Gebruiker | 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. |

# Stel de overige functionele eisen (d.w.z. aanvullend op de use cases) en de niet-functionele eisen op, genummerd en ingedeeld volgens FURPS+. In de tekst staan een paar hints voor niet-functionele eisen. Deze hints moet je nog wel wat SMARTer maken.

* 1. Functionele eisen
     1. Bij registratie ontvangt een gebruiker een startsaldo van 1000 credits.
     2. Een vraag kan een meerkeuzevraag of een kort-antwoord-vraag.
        1. Een meerkeuzevraag heeft altijd 4 keuzes en 1 goed antwoord.
        2. Een kort-antwoord-vraag heeft altijd een of meer goede antwoorden.
     3. Elke vraag behoort tot een categorie.
     4. De totale tijd die een gebruiker nodig heeft om een quiz te spelen moet (per quiz) worden bijgehouden.
     5. De totale tijd die een gebruiker nodig heeft om een quiz te spelen moet (per quiz) moet optioneel zijn voor de score-berekening.
     6. De wachtwoorden zijn vanaf de registratie beschermd opgeslagen.
  2. Niet-functionele eisen
     1. Gebruikers mogen bij het spelen van de quiz geen hinderlijke vertragingen in de applicatie ondervinden.
     2. Alle quizzen kost 40 credits.
     3. Een gebruikers score van een quiz wordt opgeslagen.
     4. Alle tekst moet in het Nederlands zijn.
     5. Het moet mogelijk zijn om in de toekomst meerdere talen te ondersteunen.
     6. Aanname (geverifieerd): gebruiker kan niet minder dan 0 credits hebben.

# Maak een domeinmodel dat alles bevat dat relevant is voor het spelen van de quiz. Ook de antwoorden die een speler geeft op de quizvragen die hij krijgt voorgelegd, dienen in het domeinmodel te worden opgenomen.

A diagram of a computer

Description automatically generated

# Beschrijf de use cases in fully-dressed format. De CRUD use cases mag je daarbij buiten beschouwing laten. Gebruik een activity diagram om de flow van de meest complexe use cases weer te geven.

**Use Case: Registreren**

|  |  |
| --- | --- |
| **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:**  Geen bestaande registratie met dezelfde gebruikersnaam. | |
| **Postconditions (Success Guarantee):**  Er is een account geregistreerd en heeft een startsaldo. De gebruiker is ingelogd met het nieuwe account. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De bezoeker geeft een gebruikersnaam en wachtwoord op. | 2. Het systeem valideert en registreert een nieuwe account met een startsaldo van 1000 credits.  3. De bezoeker wordt automatisch ingelogd. |
| **Extensions (Alternative Flow):** | |
| 1. De bezoeker geeft een invalide gebruikersnaam op. | 2. Het systeem geeft aan dat de gebruikersnaam ongeldig is. |

**Use Case: Inloggen**

|  |  |
| --- | --- |
| **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:**  De gebruiker heeft een account. | |
| **Postconditions (Success Guarantee):**  De gebruiker is ingelogd en kan een quiz spelen. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De bezoeker vult een gebruikersnaam en wachtwoord in. | 2. Het systeem controleert de gegevens en bevind dat ze goed zijn.  3. 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. |
|  | |

**Use Case: Spelen quiz**

|  |  |
| --- | --- |
| **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 controleert 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 |
|  | |

**A diagram of a workflow

Description automatically generated**

**Use Case: Bijkopen credits**

|  |  |
| --- | --- |
| **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 ingelogt. | |
| **Postconditions (Success Guarantee):**  Het saldo van de gebruiker is bijgewerkt. | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. De gebruiker selecteert het gewenste aantal credits.  3. De gebruiker voltooit de betaling via de betalingsgateway. | 2. Het systeem stuurt de gebruiker door naar de betalingsgateway.  4. Systeem verandert het saldo van de gebruiker met het gekochte aantal credits. |
| **Extensions (Alternative Flow):** | |
| 3a. gebruiker annuleert de betaling |  |
|  | |

# Werk de belangrijkste systeemoperaties van de use case(s), die je bij stap 4 hebt uitgeschreven in fully dressed format, uit met behulp van sequence diagrams via system sequence diagrams.

# Stel een design class diagram op.

# Pas het Strategy Pattern toe voor het flexibiliseren van de puntentelling. Laat in zowel het design class diagram als ook in het betreffende sequence diagram zien hoe het Strategy Pattern wordt toegepast. Mogelijk is er nog een ander GoF design pattern dat voor de hand ligt om te gebruiken. Denk daar eens goed over na en werk het, indien van toepassing, uit in het SDD.

# Beschrijf aan de hand van een aantal goed gekozen voorbeelden, die zijn te zien in het design class diagrams en/of sequence diagrams, hoe je OO ontwerpprincipes (denk met name aan SOLID, GRASP, GoF design patterns) hebt toegepast.

# Realiseer een console applicatie in Java waarmee de quiz kan worden gespeeld (zie de beschrijving hierboven onder het kopje Spelverloop quiz). Opmerking 1: De main class van deze applicatie zal via OnderwijsOnline beschikbaar worden gesteld. Je dient gebruik te maken van deze main class! Opmerking 2: De data voor deze applicatie mag gemockt worden. Je kunt hiervoor zelf een class maken die een methode bevat om een quiz op te vragen.