**Use Case:** Registreren

|  |  |
| --- | --- |
| **Primary actor:** Docent | |
| **Stakeholders and Interests:** Odinido, docent | |
| **Brief description:**  Een docent dient zich voor het aanmaken van een kennistoets te registreren. Bij de registratie moet een docent in een formulier zijn e-mailadres (die als gebruikersnaam wordt gebruikt), voornaam en achternaam, wachtwoord en de school/organisatie waar hij werkzaam is opgeven. Als hij deze gegevens heeft ingevoerd wordt een account aangemaakt. | |
| **Preconditions:**  Docent heeft nog geen account | |
| **Postconditions (Success Guarantee):**  Docent heeft een account aangemaakt | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Docent geeft aan te willen registreren | 2. Systeem geeft registratie formulier weer |
| 3. Docent vult email, voornaam, achternaam, wachtwoord en organisatie in | 4. Systeem geeft formulier voor keuze type weer |
| 5. Docent kiest type Basis | 6. Systeem maakt account aan |
| **Extensions (Alternative Flow):** | |
| 3A. Docent vult niks in  3C. Docent vult email in die al gebruikt is  5A. Docent kiest type Premium | 3B. Systeem geeft foutmelding weer en toont formulier opnieuw  3D. Systeem geeft foutmelding weer en toont formulier opnieuw  5B. Systeem geeft betaalopties weer  5C. Betaling wordt afgehandeld door online betaalservice / creditcard validator |
|  | |

**Use Case:** Aanmaken kennistoets

|  |  |
| --- | --- |
| **Primary actor:** Docent | |
| **Stakeholders and Interests:** Odinido, docent | |
| **Brief description:**  Na registratie kan een docent een kennistoets aanmaken. Een docent kan bij het aanmaken van een toets meerkeuzevragen, juist/onjuist-vragen en kort-antwoord-vragen toevoegen. Bij een kort-antwoord-vraag kan een docent vastleggen welke verschillende antwoorden correct zijn. | |
| **Preconditions:**  Docent is ingelogd | |
| **Postconditions (Success Guarantee):**  Docent heeft kennistoets aangemaakt | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Docent geeft aan kennistoets aan te willen maken | 1. Systeem vraagt om toetsnaam |
| 1. Docent vult toetsnaam in | 1. Systeem geeft mogelijkheid om vraag toe te voegen |
| 1. Docent geeft aan vraag toe te willen voegen | 1. Systeem vraagt om type vraag |
| 1. Docent kiest type van vraag | 1. Systeem vraagt om vraag |
| 1. Docent vult vraag in | 1. Systeem vraagt om antwoorden |
| 1. Docent vult mogelijke antwoorden in | 1. Systeem voegt vraag toe aan toets |
| Zolang een docent vragen wil toevoegen, worden stappen 5-12 herhaald.  13. Docent geeft aan klaar te zijn met aanmaken | 14. Kennistoets wordt aangemaakt |
| **Extensions (Alternative Flow):** | |
| 3A. Docent vult niks in  8A. Docent vult niks in  9A. Docent vult niks in  10A. Docent vult niks in | 3B. Systeem geeft foutmelding  8B. Systeem geeft foutmelding  9B. Systeem geeft foutmelding  10B. Systeem geeft foutmelding |
|  | |

**Use Case:** teamgevecht uitvoeren

|  |  |
| --- | --- |
| **Primary actor:** Docent | |
| **Stakeholders and Interests:** Odinido, student | |
| **Brief description:**  Na registratie van een Premium account kan een docent een teamgevecht starten. Een teamgevecht is een bestaande kennistoets die door teams gespeeld wordt. Een docent kiest een kennistoets en kiest vervolgens een lokaal om deze toets te starten en stelt een tijd in die gebruikt kan worden voor het doorlopen van de toets. | |
| **Preconditions:**  Docent is ingelogd  Docent heeft premium account | |
| **Postconditions (Success Guarantee):**  Docent heeft teamgevecht uitgevoerd | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Docent geeft aan teamgevecht uit te voeren. | 1. Systeem geeft kennistoetsen weer. |
| 1. Docent kiest kennistoets. | 1. Systeem vraag om tijd die student krijgt om toets te maken |
| 1. Docent vult tijd in | 1. Systeem geeft lokalen weer. |
| 1. Docent kiest lokaal. | 1. Systeem start de kennistoets. |
| **Extensions (Alternative Flow):** | |
|  | 2.A. Er zijn geen kennistoetsen van de docent.  4.A. Er is geen lokaal gereserveerd voor de docent. |
|  | |

**Use Case:** teamgevecht aanmelden

|  |  |
| --- | --- |
| **Primary actor:** Student/aanvoerder | |
| **Stakeholders and Interests:** Odinido, student | |
| **Brief description:**  Per team meldt één persoon, de aanvoerder, zich aan bij het desbetreffende lokaal. Hij dient daarvoor een studentnaam in te voeren. Vervolgens krijgt de aanvoerder automatisch een teamnummer toegewezen. | |
| **Preconditions:**  Er moet een teamgevecht gestart zijn | |
| **Postconditions (Success Guarantee):**  Team is aangemeld bij teamgevecht | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
| 1. Aanvoerder geeft aan dat hij wil aanmelden voor een teamgevecht | 1. Systeem vraagt lokaalnummer |
| 1. Student vult lokaalnummer in | 1. Systeem vraagt studentnaam |
| 1. Student vult zijn/haar naam in | 1. Systeem meldt student aan |
|  | 1. Systeem wijst teamnummer toe aan student |
|  | 1. Systeem start toets |
| **Extensions (Alternative Flow):** | |
| 3A. Student vult verkeerde lokaalnummer in  5A. Student vult niks in | 3B. Systeem toont foutmelding  5B. Systeem toont foutmelding |
|  | |

**Use Case:** teamgevecht spelen

|  |  |
| --- | --- |
| **Primary actor:** Studenten | |
| **Stakeholders and Interests:** Odinido, studenten | |
| **Brief description:**  Als een team zich heeft aangemeld bij een teamgevecht kan hij/zij beginnen met het beantwoorden van de vragen. Het team beantwoord de vragen een voor een. Een team kan eventueel terug navigeren naar een eerder bekeken vraag om alsnog een antwoord te geven of om het eerder gegeven antwoord aan te passen. Als een team alle vragen heeft beantwoord, kan het aangeven dat ze klaar zijn. | |
| **Preconditions:**  Er moet een teamgevecht gestart zijn  Team moet aangemeld zijn | |
| **Postconditions (Success Guarantee):**  Team heeft teamgevecht gespeeld | |
| **Main Success Scenario (Basic Flow):** | |
| **Actor Action** | **System Responsibility** |
|  | 1. Systeem toont vraag |
| 1. Team beantwoord de vraag | 1. Systeem gaat verder naar volgende vraag 2. Systeem toont voortgang op scorebord |
| Zolang er vragen zijn, worden stappen 1-3 herhaald  5. Team geeft aan klaar te zijn |  |
|  | 6. Systeem registreert het aantal goed beantwoorde vragen en de tijd |
| **Extensions (Alternative Flow):** | |
| 2A. Team navigeert naar volgende of vorige vraag.  5.A Student geeft aan terug te willen gaan naar een vraag. | 2.B. Systeem toont volgende of vorige vraag.  5.B. Systeem toont de vraag. |
|  | |