

Documentatie Lingo Partner App.

Student: Jan van Hest.

Studentnummer: 542565.

Inhoudsopgave

Inleiding.....	3
Showcase.	4
Functioneel Ontwerp.	5
<i>Noun - verb analysis.</i>	<i>5</i>
Nouns analysis:	5
Verb analysis:	7
<i>Conceptueel model.</i>	<i>9</i>
Entiteiten (Zelfstandige Naamwoorden):.....	9
Relaties (Werkwoorden):	9
Conceptueel model (Visueel).	11
<i>Use case diagram.</i>	<i>12</i>
Actoren:	12
Usecases:.....	12
UseCase Diagram.	13
<i>User stories.....</i>	<i>14</i>
Actor: Leerling	14
User Story 1: Voortgang bijhouden	14
User Story 2: Learning Streak.....	14
User Story 3: Interactieve spelletjes	14
User Story 4: Dashboard scores vergelijken	14
Actor: Leerkracht	15
User Story 5: Leermodules aanmaken	15
User Story 6: Leeractiviteiten aanmaken	15
User Story 7: Voortgang leerling bekijken	15
Actor: Medeleerling.....	16
User Story 8: Dashboard bekijken medeleerlingen	16
User Story 9: Medeleerlingen uitnodigen	16
Actor: Admin.....	16
User Story 10: Gebruikersaccounts beheren	16
User Story 11: Profiel aanpassen	17
User Story 12: Reeds gestarte collectie hervatten	17
<i>Acceptatie Test Plan.</i>	<i>18</i>
Toelichting:	18
Doel	18
Scope.....	18
Test Omgeving	18
Test Strategie	18
Use Case Descriptions	19
Use Case 1: Gebruiker Aanmaken	19
Use Case 2: Profiel Aanpassen	19
Use Case 3: Leermodule Aanmaken	20
Use Case 4: Leeractiviteit Aanmaken	20
Use Case 5: Voortgang Bekijken.....	21
Use Case 6: Leeractiviteit Uitvoeren.....	22
Use Case 7: Reeds Gestarte Collectie Hervatten	23
Use Case 8: Voortgang Delen met Medeleerlingen	23
Testcases	25
Use Case 1: Gebruiker Aanmaken.....	25
Use Case 2: Profiel Aanpassen	25
Use Case 3: Leermodule Aanmaken	26

Use Case 4: Leeractiviteit Aanmaken	27
Use Case 5: Voortgang Bekijken	28
Use Case 6: Leeractiviteit Uitvoeren	29
Use Case 7: Reeds Gestarte Collectie Hervatten	30
Use Case 8: Voortgang Delen met Medeleerlingen	31
Technisch Ontwerp	32
<i>Inleiding tot het technisch ontwerp</i>	<i>32</i>
<i>Domain Layer UML Diagram</i>	<i>33</i>
<i>Databaseschema diagram</i>	<i>35</i>
<i>Pakketdiagram</i>	<i>36</i>
LingoPartnerConsole	36
LingoPartnerDomain	36
LingoPartnerInfrastructuur	37
LingoPartnerShared	37
LingoPartnerTests	38
Toelichting op PackageDiagram schema	38
<i>Algoritme</i>	<i>39</i>
Het “LearningStreak” Algoritme.	39
LearningStreakService	39
LearningStreak	41
LearningStreak Activity Diagram	41
AddActivityDate Method Activity Diagram	42
UpdateDateRange Method Activity Diagram	42
MeetCriteria Method Activity Diagram	43
Strategy patterns	44
LearningStrategieën	44
Scorestrategieën	47
Toelichting op algoritme.	48
Aannames en Overwegingen	50
Evolutie van het Project	50
SOLID-principes en Dependency Injection	50
Toepassing van DIP en Dependency Injection	50
Herstructurering van de Administration Klasse (SRP, OCP)	51
Gebruik van ServiceCollection in C# (DIP, ISP)	51
BIJLAGE	52
<i>Lijst van Afbeeldingen.</i>	<i>53</i>
<i>Plantumlcode: conceptueel model</i>	<i>54</i>
<i>Plantuml: usecase diagram</i>	<i>55</i>

Inleiding.

Het leren van een nieuwe taal is een belangrijke stap voor sociale integratie en persoonlijke ontwikkeling. Mijn partner is momenteel bezig de Nederlandse taal te leren, wat me inspireerde om de "LingoPartner" app te ontwikkelen. Deze app is ontworpen om mensen te helpen die zich voorbereiden op het Nederlandse inburgeringsexamen op niveau A1 of A2. De app biedt een verscheidenheid aan interactieve en educatieve activiteiten, zoals spelletjes, invuloefeningen en multiple-choice vragen, die het leren leuk en effectief maken.

Dit verslag beschrijft het functioneel ontwerp van de "LingoPartner" app. Het doel van dit document is om een gedetailleerd overzicht te geven van de functies, gebruikersrollen en technische vereisten van het systeem. Door gebruik te maken van methodieken zoals requirementsanalyse, use-casediagrammen en user stories, wordt een helder beeld geschetst van de verwachtingen en mogelijkheden van de app.

In de volgende secties worden de verschillende onderdelen van het functioneel ontwerp besproken, waaronder de use cases, conceptuele modellen en technische specificaties. Deze elementen vormen samen een uitgebreide blauwdruk die dient als leidraad voor de ontwikkeling en implementatie van de "LingoPartner" app. Het uiteindelijke doel is om een gebruiksvriendelijke en effectieve leerervaring te bieden voor alle gebruikers.

Showcase.

LingoPartner" is een app voor het leren van talen, ideaal voor degenen die zich voorbereiden op het Nederlandse examen voor sociale integratie (Inburgering), gericht op niveau A1 of A2. Het is ontworpen om het leren van Nederlands aantrekkelijk te maken, niet alleen voor gezinnen maar ook voor vrienden. De app bevat activiteiten zoals interactieve spelletjes, invuloefeningen, multiple choice vragen, woorden vertalen, tekstbegrip, woordgebruik, toetsen en andere activiteiten die helpen bij het leren van nieuwe woorden en zinnen, op maat gemaakt voor zowel familie- als vriendengroepen. De activiteiten zijn gericht op woordbegrip, grammatica en spelling.

Een functie is de mogelijkheid voor gebruikers om vrienden uit te nodigen om deel te nemen aan het leertraject, waardoor een ondersteunende en leuke omgeving ontstaat. Gebruikers kunnen scores en voortgang vergelijken, waardoor een vriendelijk concurrentievoordeel aan de leerervaring wordt toegevoegd. Ook het behalen van een badge of diploma kan leuk zijn voor een competitieve leerervaring.

De inhoud kan door een docent worden aangeleverd. De inhoud is gestructureerd in collecties. Collecties bestaan uit hoofdstukken. Een hoofdstuk bestaat uit activiteiten binnen de app. Elke activiteit kan een score opleveren in de vorm van een cijfer. Wanneer de gebruiker een hoofdstuk voldoende afrond, kan de gebruiker een badge verdienen. Het afronden van een collectie van hoofdstukken op voldoende niveau kan een diploma opleveren.

Van de badges en diploma's kan de gebruiker een overzicht zien. Het overzicht kan men vergelijken met vrienden. Vrienden kunnen uitgenodigd worden om deel te nemen aan de activiteiten.

Per diploma kan men een voortgangs indicator zien.

De app past zich aan het individuele leertempo aan via de functie voor voortgangsregistratie. Integratie in het dagelijkse leven moedigt gebruikers aan om Nederlands te gebruiken bij routinematige activiteiten, waardoor de praktische taalvaardigheden worden vergroot. Binnen de activiteiten word een aantal getracked op a1, b1 niveau, onderwerp.

De taaluitwisselingsmodule verbindt leerlingen met moedertaalsprekers en biedt de mogelijkheid om contact te leggen met anderen. Zo kan men een afspraak maken plannen met een taalmaatje, of een videocall inplannen.

Samenvattend is "LingoPartner" meer dan een leermiddel; het is een platform om contact te maken met anderen, of het nu familie of vrienden zijn, met als gemeenschappelijk doel taalvaardigheid, waardoor de reis naar het beheersen van het Nederlands interactief en plezierig wordt.

Functioneel Ontwerp.

Noun - verb analysis.

Deze zelfstandige naamwoorden en werkwoorden een compleet beeld van een veelzijdige, interactieve taalleerapplicatie, ontworpen om gebruikers op een effectieve en aangename manier Nederlands te leren, met nadruk op zowel theoretische kennis als praktische toepassing.

Nouns analysis:

De zelfstandige naamwoorden omschrijven de sleutelcomponenten en gebruikers van de app. Ze omvatten niet alleen de basis elementen zoals 'App', 'Applicatie', en 'Talen', maar ook specifieke aspecten zoals 'Activiteiten', 'Spelletjes', en 'Invuloefeningen', die wijzen op interactieve en educatieve onderdelen binnen de app.

Belangrijke gebruikersgroepen zoals 'Gebruikers', 'Vrienden', en 'Families' worden benoemd, wat aangeeft dat de app zich richt op individueel én groepsleren.

'Scores', 'Voortgang', en 'Badges' duiden op een systeem van voortgangsbewaking en motivatie.

Meer geavanceerde functies zoals de 'Taaluitwisselingsmodule' en 'Videocall' benadrukken interactie en praktische toepassing van de taal.

- **App (Applicatie):** Een softwaretoepassing voor het leren van talen.
- **Families:** Groepen familieleden die gezamenlijk Nederlands leren met behulp van de app.
- **Talen:** De talen die geleerd kunnen worden, in dit geval Nederlands.
- **Examen:** Een specifieke toets of beoordeling, zoals het Nederlandse examen voor sociale integratie.
- **Niveau:** Vaardigheidsniveaus in een taal, hier A1 of A2.
- **Activiteiten:** Verschillende interactieve oefeningen binnen de app om taal te leren.
- **Spelling:** De manier waarop Nederlandse woorden correct worden geschreven.
- **Spelletjes:** Educatieve games in de app om taalvaardigheden op een leuke manier te oefenen.
- **Invuloefeningen:** Oefeningen waarbij gebruikers woorden of zinnen moeten aanvullen.
- **Multiple Choice Vragen:** Vragen met meerdere antwoordkeuzes ter beoordeling van taalkennis.
- **Woorden:** Individuele vocabulaire-items die geleerd worden.
- **Tekstbegrip:** Het vermogen om geschreven taal te begrijpen en te interpreteren.
- **Woordgebruik:** Het correct en gepast gebruiken van woorden in zinnen.
- **Toetsen:** Testen of beoordelingen om de voortgang in taalvaardigheid te meten.
- **Gebruikers:** Personen die de app gebruiken om talen te leren.
- **Vrienden:** Andere gebruikers van de app met wie men kan samenwerken of concurreren.
- **Scores:** Punten of beoordelingen verkregen voor prestaties binnen de app.
- **Badge:** Een digitale erkenning of beloning voor het bereiken van bepaalde mijlpalen.

- **Diploma:** Certificaat of bewijs van het succesvol afronden van een cursus of niveau.
- **Inhoud:** Educatief materiaal of lessen aangeboden in de app.
- **Docent:** De persoon die educatieve inhoud voor de app levert of creëert.
- **Collecties:** Verzamelingen van gerelateerde educatieve content binnen de app.
- **Hoofdstukken:** Onderdelen of secties binnen een collectie in de app.
- **Voortgang:** De vooruitgang van een gebruiker in het leren van de Nederlandse taal binnen de app.
- **Voortgangsregistratie:** Het systeem binnen de app dat bijhoudt hoe ver een gebruiker gevorderd is in het leerproces.
- **Voortgangsindicator:** Tool om de vooruitgang van de gebruiker in de app te volgen.
- **Leertempo:** De snelheid waarmee een gebruiker nieuwe informatie leert en verwerkt.
- **Taalvaardigheid:** De mate van beheersing en competentie in een taal, in dit geval Nederlands.
- **Taalvaardigheden:** De bekwaamheid in het spreken, schrijven, luisteren, en lezen in een taal.
- **Moedertaalsprekers:** Mensen voor wie Nederlands de moedertaal is en die als taalpartners kunnen dienen in de app.
- **Taaluitwisselingsmodule:** Een onderdeel van de app dat gebruikers verbindt met moedertaalsprekers voor praktijkgerichte taaluitwisseling.
- **Taalmaatje:** Een partner om mee te oefenen in de taaluitwisselingsmodule.
- **Videocall:** Video-oproepfunctie voor live gesprekken binnen de app.
- **Platform:** De digitale omgeving of infrastructuur van de app.

Verb analysis:

De werkwoorden beschrijven de acties en doelstellingen van de app. ‘Voorbereiden’, ‘Leren’, en ‘Helpen’ geven de primaire functies aan: educatie en ondersteuning bij het leren van Nederlands. ‘Maken’ en ‘Bevatten’ verwijzen naar de opbouw van de app met diverse leermaterialen en oefeningen. ‘Uitnodigen’, ‘Deelnemen’, en ‘Vergelijken’ suggereren sociale interactie en competitieve elementen. ‘Behalen’ en ‘Volgen’ duiden op het bereiken van mijlpalen en het persoonlijk volgen van een leertraject. ‘Aanpassen’ en ‘Integreren/Encouragen’ wijzen op de flexibiliteit van de app en de nadruk op praktische toepassing van de taal in het dagelijks leven. ‘Verbinden’ en ‘Plannen/Inplannen’ belichten de mogelijkheid om te interageren met moedertaalsprekers en persoonlijke leerervaringen te organiseren.

- **Integreren:** De app integreert taalgebruik in het dagelijks leven en moedigt praktische toepassing aan.
- **Leren:** Het hoofddoel van de app, waarbij gebruikers Nederlands leren.
- **Bevatten:** De app omvat verschillende soorten leermateriaal en oefeningen.
- **Aanleveren/Structureren:** Docenten leveren en structureren de inhoud van de app.
- **Volgen:** Gebruikers kunnen hun eigen leerpad volgen door hoofdstukken.
- **Voorbereiden:** Het proces van studeren of trainen voor het Nederlandse examen.
- **Ontworpen:** Het creëren of structureren van de app met een bepaald doel.
- **Maken:** In deze context, het aantrekkelijk maken van het leerproces.
- **Bevat:** Het hebben of omvatten van bepaalde elementen of functies in de app.
- **Helpen:** Het bieden van ondersteuning of vergemakkelijking in het leerproces.
- **Gericht zijn op:** De focus of doelstelling van de app richten op specifieke vaardigheidsniveaus.
- **Uitnodigen:** De actie van gebruikers om anderen aan te moedigen deel te nemen.
- **Deelnemen:** Actief betrokken zijn bij het leerproces of activiteiten.
- **Ontstaan:** Het tot stand komen of vormen van een omgeving of situatie.
- **Vergelijken:** Het beoordelen of evalueren van prestaties ten opzichte van anderen.
- **Toevoegen:** Het introduceren van nieuwe elementen of aspecten in de ervaring.
- **Behalen:** Het succesvol voltooien of verkrijgen van een doel, zoals een badge of diploma.
- **Aangeleverd worden:** Het verstrekken of bezorgen van educatieve inhoud door een docent.
- **Bestaan uit:** Samengesteld zijn of gevormd worden door bepaalde elementen.
- **Opleveren:** Resulteren in of het verkrijgen van iets, zoals een score.
- **Afronden:** Het voltooien of afmaken van een hoofdstuk of collectie.
- **Verdiennen:** Het ontvangen van een beloning of erkenning voor prestaties.
- **Zien:** Het visueel waarnemen of bekijken van informatie, zoals scores of voortgang.
- **Uitgenodigd worden:** Het ontvangen van een uitnodiging om deel te nemen.

- **Aanpassen:** Het wijzigen of personaliseren van de app naar het leertempo van de gebruiker.
- **Moedigen aan:** Het stimuleren of bevorderen van bepaald gedrag, zoals het gebruik van Nederlands.
- **Worden vergroot:** Het toenemen of verbeteren van vaardigheden.
- **Getracked worden:** Het volgen of monitoren van activiteiten of prestaties.
- **Verbinden:** Het tot stand brengen van een relatie of connectie tussen gebruikers.
- **Leggen:** Het maken van een connectie of het initiëren van communicatie.
- **Maken:** In deze context, het plannen of organiseren van afspraken of videogesprekken.
- **Inplannen:** Het vaststellen of organiseren van een bepaalde activiteit op een specifiek moment.
- **Samenvatten:** Het beknopt overzien of beschrijven van de essentie van de app.

Conceptueel model.

Ik heb een kleine analyse gemaakt van zelfstandige naamwoorden en werkwoorden die in de casus voorkomen. Deze zal ik hieronder benoemen en later zal ik de belangrijkste onderdelen van de “LingoPartner” app en hoe ze samenwerken in een conceptueel model weergeven.

Entiteiten (Zelfstandige Naamwoorden):

- **Gebruikers:** Centraal in het model, omvatten individuen, families, vrienden en taalmaatjes.
- **Leeractiviteiten:** Hieronder vallen interactieve spelletjes, invuloefeningen, multiple choice vragen, Taaluitwisselingsmodule, etc., gericht op verschillende taalaspecten zoals woordbegrip, grammatica en spelling.
- **Educatieve Inhoud:** Bestaande uit collecties en hoofdstukken, aangeleverd en gestructureerd door docenten. Deze inhoud vormt de basis voor de leeractiviteiten.
- **Voortgang:** De voortgang van gebruikers in het leerproces, inclusief scores, badges, diploma's en voortgangsindicatoren.
- **Beloningen:** Badges en diploma's. Dit systeem motiveert gebruikers door het zichtbaar maken van hun voortgang.
- **Sociale Interactie:** Mogelijkheden voor gebruikers om vrienden uit te nodigen en deel te nemen aan leeractiviteiten, evenals het vergelijken van scores en voortgang.
- **Taaluitwisselingsmodule:** Verbindt gebruikers met moedertaalsprekers voor praktische taal oefeningen, waaronder het plannen van afspraken of videocalls.
- **Voortgangsregistratie:** Houdt de voortgang van gebruikers bij, waardoor het leerproces op individuele basis gevolgd kan worden.
- **Platform:** De digitale omgeving waarbinnen al deze componenten functioneren.

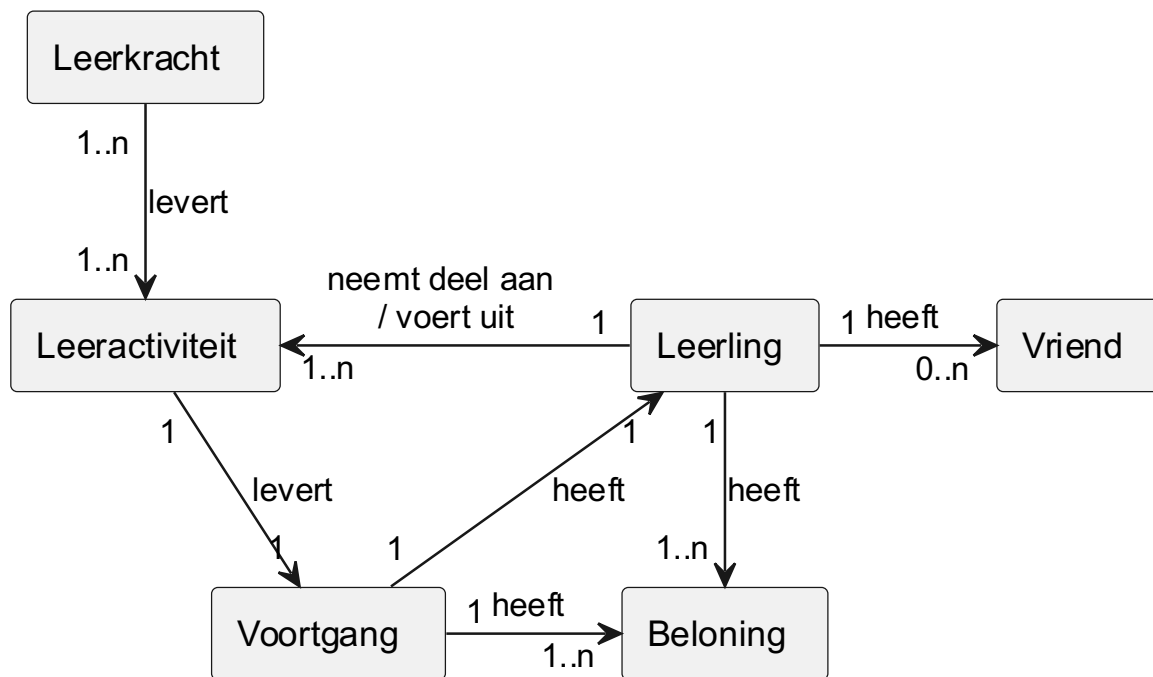
Relaties (Werkwoorden):

- **Integreren:** De app integreert Nederlands in het dagelijks leven van gebruikers, wat bijdraagt aan de ontwikkeling van praktische taalvaardigheden.
- **Leren:** Gebruikers leren Nederlands door interactieve activiteiten en oefeningen die door de app worden aangeboden.
- **Bevatten en Aanleveren/Structureren:** De app bevat een breed scala aan educatief materiaal, gestructureerd en aangeleverd door docenten, waardoor een solide leerbasis wordt geboden.
- **Volgen:** Gebruikers kunnen hun eigen leerp pad volgen door middel van gestructureerde hoofdstukken en collecties, wat leidt tot een gepersonaliseerde leerervaring.
- **Voorbereiden:** De app bereidt gebruikers voor op het Nederlandse inburgeringsexamen door gerichte inhoud en oefeningen aan te bieden.
- **Maken, Helpen, Gericht Zijn Op:** De app is ontworpen om het leerproces aantrekkelijk te maken en helpt gebruikers door zich te richten op specifieke vaardigheidsniveaus (A1 of A2).

- **Uitnodigen en Deelnemen:** Gebruikers kunnen vrienden uitnodigen om deel te nemen aan de leerervaring, wat zorgt voor interactie en samenwerking.
- **Vergelijken, Toevoegen, Behalen, Verdienen:** Gebruikers kunnen scores en voortgang vergelijken met vrienden, wat een element van competitie toevoegt. Het behalen van badges en diploma's wordt een motiverende factor.
- **Zien en Worden Uitgenodigd:** Gebruikers kunnen hun voortgang en behaalde badges/diploma's zien. Ze kunnen ook worden uitgenodigd door anderen om deel te nemen aan leeractiviteiten.
- **Aanpassen:** De app past zich aan het individuele leertempo van elke gebruiker aan, wat bijdraagt aan een effectievere en persoonlijke leerervaring.
- **Aanmoedigen en Worden Vergroot:** De app moedigt gebruikers aan om Nederlands te gebruiken in hun dagelijks leven, waardoor hun taalvaardigheden worden vergroot.
- **Getracked Worden:** De voortgang van de gebruiker op verschillende niveaus en onderwerpen wordt gevolgd, wat helpt bij het identificeren van leergebieden en vooruitgang.
- **Verbinden, Leggen, Maken, Inplannen:** De taaluitwisselingsmodule stelt gebruikers in staat om verbindingen te leggen met moedertaalsprekers, wat leidt tot praktische taal oefening. Gebruikers kunnen afspraken en videocalls plannen voor verdere taalpraktijk.
- **Samenvatten:** Ten slotte, "LingoPartner" wordt samengevat als een platform dat niet alleen dient als een leermiddel maar ook als een manier om contact te maken met anderen, wat de taalleerervaring verrijkt en verbreedt.

Conceptueel model (Visueel).

Het conceptuele model hieronder laat de belangrijkste onderdelen van de "LingoPartner" app en hoe ze samenwerken zien op een visuele manier. Dit omvat gebruikers, soorten leeractiviteiten, lesmateriaal, hoe ver iemand is met leren, beloningen, manieren om met anderen te communiceren, een speciaal deel voor taaluitwisseling, en hoe de app de voortgang van een gebruiker volgt. Het laat zien hoe al deze delen samen een app vormen die op een leuke en nuttige manier helpt bij het leren van Nederlands, met aandacht voor zowel theorie als praktijk.



Figuur 1 Conceptueel model

Use case diagram.

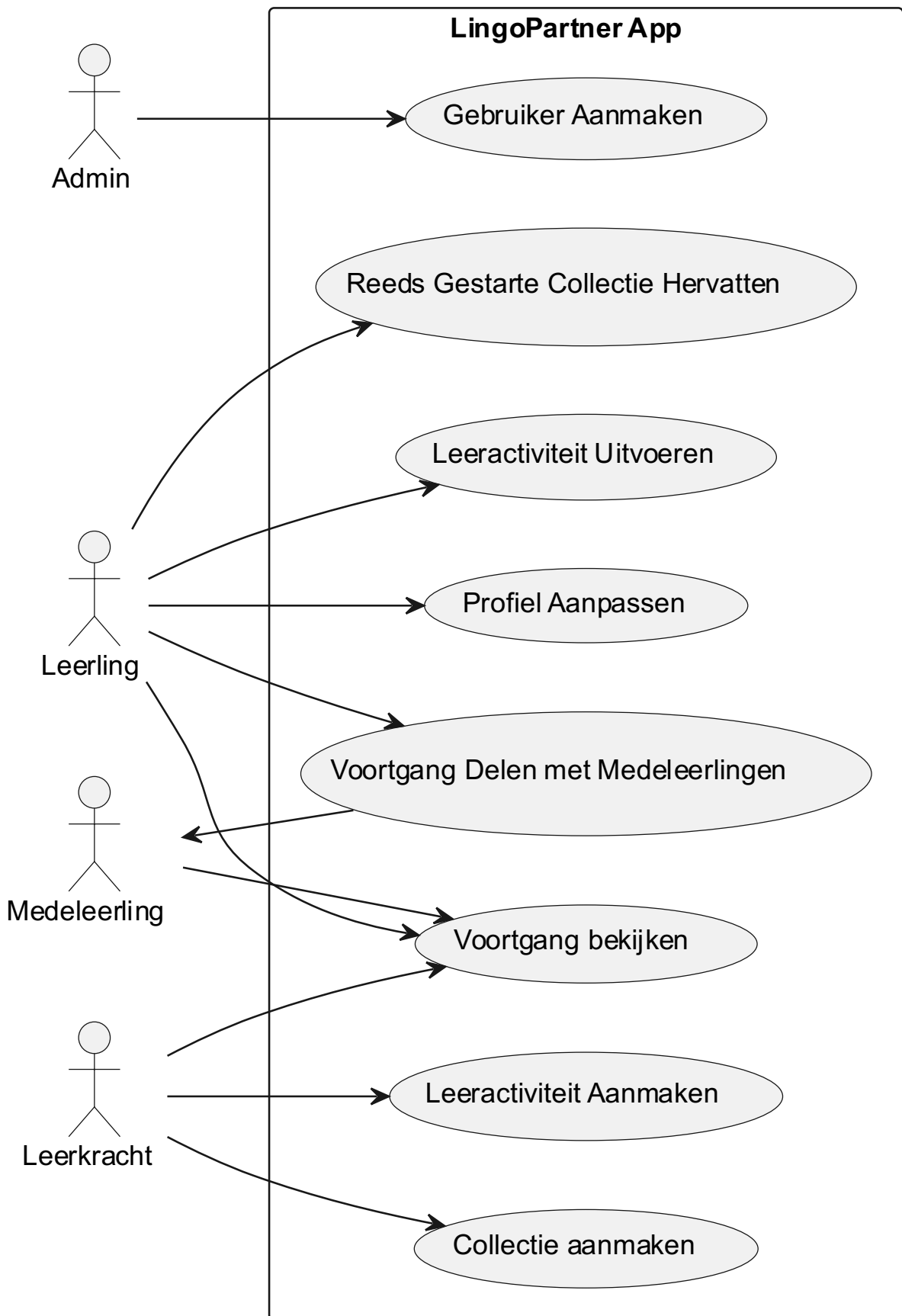
Actoren:

- **Leerling:** *De gebruiker van de app, die deelneemt aan leeractiviteiten, voortgang maakt, beloningen verdient, en communiceert met vrienden en taalmaatjes.*
- **Leerkracht:** *De leverancier van educatieve inhoud, die collecties en hoofdstukken aanmaakt en aanlevert.*
- **Vriend:** *Een andere gebruiker van de app, met wie de leerling scores en voortgang kan vergelijken, en met wie de leerling kan communiceren.*
- **Admin:** *Een beheerder van de app, die gebruikersaccounts kan aanmaken en beheren.*

Usecases:

- **Gebruiker Aanmaken:** *Een nieuwe gebruiker maakt een account aan om de app te gebruiken.*
- **Profiel Aanpassen:** *Een gebruiker past zijn of haar profiel aan, inclusief persoonlijke informatie en voorkeuren.*
- **Collectie aanmaken:** *Een leerkracht maakt een nieuwe collectie aan, bestaande uit verschillende hoofdstukken.*
- **Leeractiviteit Aanmaken:** *Een leerkracht maakt een nieuwe leeractiviteit aan, die deel uitmaakt van een hoofdstuk.*
- **Voortgang bekijken:** *Een leerling, leerkracht of vriend bekijkt zijn of haar voortgang, inclusief scores, badges, diploma's en voortgangsindicatoren.*
- **Leeractiviteit Uitvoeren:** *Een leerling voert een leeractiviteit uit, zoals een interactief spel, invuloefening of multiple choice vraag.*
- **Woorden Leren:** *Een leerling leert nieuwe woorden en zinnen door middel van educatieve activiteiten.*
- **Invuloefening en Meerkeuzevraag Uitvoeren:** *Een leerling voert een invuloefening of meerkeuzevraag uit als onderdeel van een leeractiviteit.*
- **Reeds Gestarte Collectie Hervatten:** *Een leerling hervat een collectie die eerder is gestart, om de voortgang voort te zetten.*
- **Voortgang Delen met Vrienden:** *Een leerling deelt zijn of haar voortgang met vrienden, om scores en badges te vergelijken.*

UseCase Diagram.



User stories.

NOOT: De userstories komen niet 1 op 1 overeen met de usecases. Voor het gemak heb ik bij elke userstory vermeld bij welke usecase het hoort.

Zo kun je als leerling een dashboard score vergelijken. Wil je iets kunnen vergelijken zal je daardoor ook een andere student moeten hebben uitgenodigd. User Story 8 “Dashboard bekijken medeleerlingen” heeft daardoor een relatie met Use Case 5: “Voortgang Bekijken” en Use Case 8: “Voortgang Delen met Medeleerlingen”

Actor: Leerling

User Story 1: Voortgang bijhouden

- Als leerling
- Wil ik mijn voortgang kunnen bijhouden
- Zodat ik kan zien hoeveel ik al geleerd heb.

Acceptatiecriteria:

- De voortgang moet visueel weergegeven worden in een voortgangsbalk op het dashboard.
- Er moet een overzicht zijn van behaalde badges en diploma's.

Relevant Use Cases:

- ✎ Use Case 5: Voortgang Bekijken

User Story 2: Learning Streak

- Als leerling
- Wil ik gemotiveerd worden door subtiele hints in het programma, zoals een “learning streak” op het dashboard
- Zodat ik dagelijks blijf leren.

Acceptatiecriteria:

- De learning streak wordt weergegeven op het dashboard.
- De duur van de learning streak wordt weergegeven met behulp van sterren.

Relevant Use Cases:

- ✎ Use Case 5: Voortgang Bekijken

User Story 3: Interactieve spelletjes

- Als leerling
- Wil ik interactieve spelletjes kunnen spelen
- Zodat ik op een leuke manier Nederlands kan leren.

Acceptatiecriteria:

- De app moet minimaal drie verschillende soorten interactieve spelletjes bieden.
- Spelletjes moeten punten of scores toekennen voor feedback.

Relevant Use Cases:

- ✎ Use Case 6: Leeractiviteit Uitvoeren

User Story 4: Dashboard scores vergelijken

- Als leerling

- Wil ik mijn scores kunnen vergelijken met die van mijn vrienden
- Zodat ik mijn voortgang kan meten ten opzichte van anderen.

Acceptatiecriteria:

- Er moet een scorebord weergegeven worden op het dashboard.
- Leerlingen die elkaar als “vriend” hebben aangemerkt, kunnen elkaars dashboard bekijken.

Relevant Use Cases:

- ✎ Use Case 5: Voortgang Bekijken
- ✎ Use Case 8: Voortgang Delen met Medeleerlingen

Actor: Leerkracht

User Story 5: Leermodule aanmaken

- Als leerkracht
- Wil ik leermodule kunnen aanmaken
- Zodat ik gestructureerde leerinhoud kan aanbieden.

Acceptatiecriteria:

- ✓ Leerkrachten moeten de mogelijkheid hebben om modules aan te maken, te bewerken en te verwijderen.
- ✓ Er moet een eenvoudige interface zijn voor het beheren van leerinhoud.

Relevant Use Cases:

- ✎ Use Case 3: Leermodule Aanmaken

User Story 6: Leeractiviteiten aanmaken

- Als leerkracht
- Wil ik leeractiviteiten kunnen toewijzen aan specifieke hoofdstukken
- Zodat ik de leerinhoud kan organiseren.

Acceptatiecriteria:

- ✓ Leerkrachten moeten leeractiviteiten kunnen koppelen aan specifieke hoofdstukken.

Relevant Use Cases:

- ✎ Use Case 4: Leeractiviteit Aanmaken

User Story 7: Voortgang leerling bekijken

- Als leerkracht
- Wil ik de voortgang van mijn leerlingen kunnen volgen
- Zodat ik hun leerproces kan ondersteunen.

Acceptatiecriteria:

- ✓ Er moet een dashboard zijn waar leerkrachten de voortgang van individuele leerlingen kunnen zien.
- ✓ Leerkrachten moeten meldingen krijgen bij belangrijke mijlpalen of problemen in de voortgang.

Relevant Use Cases:

- ✎ Use Case 5: Voortgang Bekijken

Actor: Medeleerling

User Story 8: Dashboard bekijken medeleerlingen

- Als medeleerling
- Wil ik mijn voortgang/dashboard kunnen vergelijken met die van mijn medeleerlingen
- Zodat ik gemotiveerd blijf om te leren.

Acceptatiecriteria:

- ✓ Er moet een functie zijn om medeleerlingen toe te voegen.
- ✓ Er moet een functie zijn om de voortgang van je medeleerlingen te zien

Relevant Use Cases:

- ✎ Use Case 5: Voortgang Bekijken
- ✎ Use Case 8: Voortgang Delen met Medeleerlingen

User Story 9: Medeleerlingen uitnodigen

- Als medeleerling
- Wil ik mijn medeleerlingen kunnen uitnodigen op de app
- Zodat we samen kunnen leren.

Acceptatiecriteria:

- ✓ Er moet een uitnodigingsfunctie zijn waarmee gebruikers medeleerlingen via gebruikersnaam kunnen uitnodigen.
- ✓ De app moet meldingen sturen wanneer een uitnodiging geaccepteerd is.
- ✓ De app toont een melding wanneer men een uitnodiging heeft gekregen.

Relevant Use Cases:

- ✎ Use Case 8: Voortgang Delen met Medeleerlingen

Actor: Admin

User Story 10: Gebruikersaccounts beheren

- Als admin
- Wil ik gebruikersaccounts kunnen beheren
- Zodat ik de toegang tot de app kan controleren.

Acceptatiecriteria:

- ✓ Er moet een beheerdersdashboard zijn waar admins accounts kunnen aanmaken, bewerken en verwijderen.
- ✓ Admins moeten wachtwoorden kunnen resetten en gebruikersrollen kunnen toewijzen.

Relevant Use Cases:

- ✎ Use Case 1: Gebruiker Aanmaken

User Story 11: Profiel aanpassen

- **Als** leerling
- **Wil ik** mijn profiel kunnen aanpassen
- **Zodat** ik mijn persoonlijke informatie en voorkeuren up-to-date kan houden.

Acceptatiecriteria:

- ✓ De gebruiker moet persoonlijke informatie en voorkeuren kunnen wijzigen.
- ✓ Wijzigingen moeten worden opgeslagen en bevestigd door het systeem.

Relevant Use Cases:

- ✎ Use Case 2: Profiel Aanpassen

User Story 12: Reeds gestarte collectie hervatten

- **Als** leerling
- **Wil ik** een reeds gestarte collectie kunnen hervatten
- **Zodat** ik mijn leeractiviteiten kan voortzetten vanaf waar ik was gebleven.

Acceptatiecriteria:

- ✓ De leerling moet de mogelijkheid hebben om eerder gestarte collecties te hervatten.
- ✓ De voortgang in de collectie moet worden hersteld en correct weergegeven.

Relevant Use Cases:

- ✎ Use Case 7: Reeds Gestarte Collectie Hervatten

Acceptatie Test Plan.

Toelichting:

Doel

Het doel van dit acceptatie test plan is om te verifiëren dat de “LingoPartner” console applicatie voldoet aan de gespecificeerde user stories en use cases, en dat het systeem correct functioneert volgens de opgegeven acceptatiecriteria.

Scope

Dit plan omvat de acceptatietesten voor de volgende actoren en hun respectieve user stories:

- Leerling
- Leerkracht
- Vriend
- Admin

Test Omgeving

Platform: Console applicatie

Test Data: Vooraf gedefinieerde gebruikers en leeractiviteiten

Test Tools: Handmatige tests uitgevoerd in de console omgeving

Test Strategie

Elke user story en use case wordt getest door het uitvoeren van specifieke testcases die zijn ontworpen om te verifiëren dat de acceptatiecriteria worden nageleefd. De testcases omvatten zowel de hoofd stroom als alternatieve stromen.

Use Case Descriptions

Use Case 1: Gebruiker Aanmaken

Doel: Een nieuwe gebruiker aanmaken zodat deze toegang heeft tot de app.

Actoren: Admin, Leerkracht

User Story: 10 Gebruikersaccounts beheren

Precondities: Admin of Leerkracht is ingelogd.

Postcondities: De nieuwe gebruiker is aangemaakt en kan inloggen in de app.

Hoofd Stroom:

1. Admin of Leerkracht logt in op de console applicatie.
2. Admin of Leerkracht kiest de optie “Gebruiker Aanmaken”.
3. Admin of Leerkracht voert de vereiste gegevens in (gebruikersnaam, wachtwoord, rol, etc.).
4. Het systeem slaat de gebruikersgegevens op.
5. Het systeem bevestigt dat de nieuwe gebruiker succesvol is aangemaakt.

Alternatieve Stroom:

- ✎ 3a. Als de ingevoerde gebruikersnaam al bestaat, geeft het systeem een foutmelding en vraagt om een andere gebruikersnaam. De admin of leerkracht voert een andere gebruikersnaam in en gaat verder met stap 3.

Use Case 2: Profiel Aanpassen

Doel: Een gebruiker past zijn of haar profiel aan, inclusief persoonlijke informatie en voorkeuren.

Actoren: Leerling, Leerkracht, Medeleerling, Admin.

Userstory: geen

Precondities: De gebruiker is ingelogd.

Postcondities: De profielinformatie van de gebruiker is bijgewerkt.

Hoofd Stroom:

1. De gebruiker logt in op de console applicatie.
2. De gebruiker kiest de optie “Profiel Aanpassen”.
3. De gebruiker voert de wijzigingen in de profielinformatie in.
4. Het systeem slaat de nieuwe profielinformatie op.
5. Het systeem bevestigt dat het profiel succesvol is bijgewerkt.

Alternatieve Stroom:

- ✎ 3a. Als de ingevoerde informatie ongeldig is, geeft het systeem een foutmelding en vraagt om correcte informatie. De gebruiker voert correcte informatie in en gaat verder met stap 3.

Use Case 3: Leermodule Aanmaken

Doel: Een leerkracht maakt een nieuwe leermodule aan, bestaande uit verschillende hoofdstukken.

Actoren: Leerkracht

User story: Leermodule aanmaken.

Precondities: Leerkracht is ingelogd.

Postcondities: De nieuwe collectie is aangemaakt en klaar voor gebruik.

Hoofd Stroom:

1. De leerkracht logt in op de console applicatie.
2. De leerkracht kiest de optie “Leermodule Aanmaken”.
3. De leerkracht voert de naam en beschrijving van de collectie in.
4. Het systeem slaat de collectiegegevens op.
5. Het systeem bevestigt dat de collectie succesvol is aangemaakt.

Alternatieve Stroom:

- ✓ 3a. Als de naam van de collectie al bestaat, geeft het systeem een foutmelding en vraagt om een andere naam. De leerkracht voert een andere naam in en gaat verder met stap 3.

Use Case 4: Leeractiviteit Aanmaken

Doel: Een leerkracht maakt een nieuwe leeractiviteit aan, die deel uitmaakt van een hoofdstuk.

Actoren: Leerkracht

User Story: 6 Leeractiviteiten aanmaken

Precondities: Leerkracht is ingelogd.

Postcondities: De nieuwe leeractiviteit is aangemaakt en toegewezen aan een hoofdstuk.

Hoofd Stroom:

1. De leerkracht logt in op de console applicatie.
2. De leerkracht kiest de optie “Leeractiviteit Aanmaken”.
3. De leerkracht selecteert de collectie en het hoofdstuk waaraan de leeractiviteit wordt toegevoegd.
4. De leerkracht voert de details van de leeractiviteit in (naam, beschrijving, type activiteit, etc.).

5. Het systeem slaat de leeractiviteit op en koppelt deze aan het gekozen hoofdstuk.
6. Het systeem bevestigt dat de leeractiviteit succesvol is aangemaakt.

Alternatieve Stroom:

- ✓ 3a. Als de naam van de leeractiviteit al bestaat, geeft het systeem een foutmelding en vraagt om een andere naam. De leerkracht voert een andere naam in en gaat verder met stap 3.

Use Case 5: Voortgang Bekijken

Doel: Een leerling, leerkracht of medeleerling bekijkt het dashboard van de leerling om de voortgang, inclusief scores, badges, diploma's, voortgangsindicatoren, huidige en afgeronde modules, en de learning streak te zien.

Actoren: Leerling, Leerkracht, Medeleerling

User Story:

- ✗ User Story 1: Voortgang bijhouden
- ✗ User Story 2: Learning Streak
- ✗ User Story 4: Dashboard scores vergelijken
- ✗ User Story 7: Voortgang leerling bekijken
- ✗ User Story 8: Dashboard bekijken medeleerlingen

Precondities:

- ✓ De gebruiker is ingelogd.
- ✓ Er zijn activiteiten uitgevoerd door de leerling.
- ✓ Er is ten minste één module gestart door de leerling.
- ✓ Er is ten minste één module afgerond door de leerling.

Postcondities:

- ✓ De voortgang van de leerling wordt weergegeven op het dashboard.
- ✓ De gebruiker heeft een overzicht van de voortgangsbalken voor de huidige en afgeronde modules.
- ✓ De gebruiker kan de behaalde badges en diploma's van de leerling zien.
- ✓ De learning streak van de leerling wordt weergegeven, inclusief de activiteitscore.
- ✓ De gebruiker kan de scores van de leerling vergelijken met die van vrienden op het scorebord.
- ✓ Meldingen en belangrijke mijlpalen in de voortgang van de leerling worden weergegeven.
- ✓ De voortgang van medeleerlingen kan worden bekeken indien gewenst.

Hoofd Stroom:

- 1 De gebruiker logt in op de console applicatie.
- 2 De gebruiker kiest de optie "Dashboard Bekijken".

- 3 Het systeem toont de voortgang van de leerling in een voortgangsbalk (relevant voor User Story 1).
- 4 Het systeem toont een overzicht van behaalde badges en diploma's (relevant voor User Story 1).
- 5 Het systeem toont de learning streak van de leerling met behulp van een score die de activiteit weergeeft (relevant voor User Story 2).
- 6 Het systeem toont een scorebord waarop de leerling zijn of haar scores kan vergelijken met die van vrienden (relevant voor User Story 4).
- 7 Het systeem toont meldingen en belangrijke mijlpalen in de voortgang van de leerling (relevant voor User Story 7).
- 8 Het systeem geeft de mogelijkheid om de voortgang van medeleerlingen te bekijken (relevant voor User Story 8).
- 9 Het systeem toont de modules waarmee de leerling momenteel bezig is.
- 10 Het systeem toont de voortgang in deze modules door middel van een voortgangsbalk.
- 11 Het systeem toont de modules die de leerling heeft afgerond.

Alternatieve Stroom:

- ✎ 3a. Als er geen voortgangsgegevens beschikbaar zijn, toont het systeem een bericht dat er nog geen data beschikbaar is.

Use Case 6: Leeractiviteit Uitvoeren

Doel: Een leerling voert een leeractiviteit uit, zoals een interactief spel, invuloefening of multiple choice vraag.

Actoren: Leerling

User story: 3 Leeractiviteiten.

Precondities: De leerling is ingelogd en heeft toegang tot de leeractiviteit.

Postcondities: De leeractiviteit is voltooid en de resultaten zijn opgeslagen.

Hoofd Stroom:

1. De leerling logt in op de console applicatie.
2. De leerling kiest de optie "Leeractiviteit Uitvoeren".
3. De leerling selecteert een beschikbare leeractiviteit.
4. Het systeem start de leeractiviteit.
5. De leerling voltooit de leeractiviteit.
6. Het systeem slaat de resultaten van de leeractiviteit op.
7. Het systeem bevestigt dat de leeractiviteit succesvol is voltooid.

Alternatieve Stroom:

- ✎ 4a. Als de leeractiviteit niet geladen kan worden, geeft het systeem een foutmelding en vraagt om opnieuw te proberen.

Use Case 7: Reeds Gestarte Collectie Hervatten

Doel: Een leerling hervat een collectie die eerder is gestart, om de voortgang voort te zetten.

Actoren: Leerling

User story: Geen specifieke user story genoemd, maar relevant voor leerlingen die hun voortgang willen hervatten.

Precondities: De leerling is ingelogd en heeft eerder een collectie gestart.

Postcondities: De leerling hervat de collectie en kan verdergaan waar deze gebleven is.

Hoofd Stroom:

- 1 De leerling logt in op de console applicatie.
- 2 De leerling kiest de optie “Reeds Gestarte Collectie Hervatten”.
- 3 Het systeem toont de eerder gestarte collecties.
- 4 De leerling selecteert een collectie om te hervatten.
- 5 Het systeem herstelt de voortgang van de leerling in de geselecteerde collectie.
- 6 De leerling gaat verder met de leeractiviteiten binnen de collectie.
- 7 Het systeem slaat de voortgang van de leerling op.

Alternatieve Stroom:

- ✗ 3a. Als er geen eerder gestarte collecties zijn, toont het systeem een bericht dat er geen collecties beschikbaar zijn om te hervatten.

Use Case 8: Voortgang Delen met Medeleerlingen

Doel: Een leerling kan zijn of haar voortgang delen met medeleerlingen om toegang te krijgen tot elkaars dashboard en voortgang met vrienden, scores en badges te vergelijken.

Actoren: Leerling, Medeleerling

User story:

- ✗ User Story 4: Dashboard scores vergelijken
- ✗ User Story 8: Dashboard bekijken medeleerlingen
- ✗ User Story 9: Medeleerlingen uitnodigen

Precondities: De leerling is ingelogd.

Postcondities: De voortgang van de leerling is gedeeld met de geselecteerde medeleerlingen.

Hoofd Stroom:

- 1 De leerling logt in op de console applicatie.
- 2 De leerling kiest de optie “Voortgang Delen met Medeleerlingen”.
- 3 Het systeem toont de lijst van medeleerlingen.
- 4 De leerling selecteert de medeleerlingen met wie de voortgang gedeeld moet worden.

- 5 Het systeem deelt de voortgangsinformatie met de geselecteerde medeleerlingen.
- 6 Het systeem bevestigt dat de voortgang succesvol is gedeeld.

Alternatieve Stroom:

- ✎ 3a. Als de leerling geen medeleerlingen heeft toegevoegd, toont het systeem een bericht dat er geen medeleerlingen beschikbaar zijn om voortgang mee te delen.

Testcases

Use Case 1: Gebruiker Aanmaken

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. Admin of Leerkracht logt in op de console applicatie.	Inlogschermbord verdwijnt, dashboard verschijnt.		
2. Admin of Leerkracht kiest de optie "Gebruiker Aanmaken".	Formulier voor nieuwe gebruiker verschijnt.		
3. Admin of Leerkracht voert de vereiste gegevens in.	Gegevens worden ingevoerd in de juiste velden.		
4. Het systeem controleert of de gebruikersnaam al bestaat.	Gebruikersnaam wordt gecontroleerd.		
5. Als de gebruikersnaam uniek is, slaat het systeem de gebruikersgegevens op.	Gebruiker wordt succesvol aangemaakt.		
6. Het systeem bevestigt dat de nieuwe gebruiker succesvol is aangemaakt.	Bevestigingsbericht verschijnt.		
3a. Als de ingevoerde gebruikersnaam al bestaat, geeft het systeem een foutmelding en vraagt om een andere gebruikersnaam.	Foutmelding verschijnt.		

Use Case 2: Profiel Aanpassen

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De gebruiker logt in op de console applicatie.	Inlogschermbord verdwijnt, dashboard verschijnt.		
2. De gebruiker kiest de optie "Profiel Aanpassen".	Formulier voor profielaanpassing verschijnt.		
3. De gebruiker voert de wijzigingen in de profielinformatie in.	Gegevens worden ingevoerd in de juiste velden.		
4. Het systeem controleert de ingevoerde informatie.	Informatie wordt gecontroleerd.		
5. Het systeem slaat de nieuwe profielinformatie op.	Profielinformatie wordt bijgewerkt.		
6. Het systeem bevestigt dat het profiel succesvol is bijgewerkt.	Bevestigingsbericht verschijnt.		
3a. Als de ingevoerde informatie ongeldig is, geeft het systeem een foutmelding en vraagt om correcte informatie.	Foutmelding verschijnt.		

Use Case 3: Leermodule Aanmaken

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De leerkracht logt in op de console applicatie.	Inlogschermbewordwijnt, dashboard verschijnt.		
2. De leerkracht kiest de optie "Leermodule Aanmaken".	Formulier voor leermodule verschijnt.		
3. De leerkracht voert de naam en beschrijving van de collectie in.	Gegevens worden ingevoerd in de juiste velden.		
4. Het systeem controleert of de naam van de collectie al bestaat.	Naam wordt gecontroleerd.		
5. Als de naam uniek is, slaat het systeem de collectiegegevens op.	Leermodule wordt succesvol aangemaakt.		
6. Het systeem bevestigt dat de collectie succesvol is aangemaakt.	Bevestigingsbericht verschijnt.		
3a. Als de naam van de collectie al bestaat, geeft het systeem een foutmelding en vraagt om een andere naam.	Foutmelding verschijnt.		

Use Case 4: Leeractiviteit Aanmaken

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De leerkracht logt in op de console applicatie.	Inlogschermbord verdwijnt, dashboard verschijnt.		
2. De leerkracht kiest de optie "Leeractiviteit Aanmaken".	Formulier voor leeractiviteit verschijnt.		
3. De leerkracht selecteert de collectie en het hoofdstuk waaraan de leeractiviteit wordt toegevoegd.	Selectieschermbord verschijnt.		
4. De leerkracht voert de details van de leeractiviteit in.	Gegevens worden ingevoerd in de juiste velden.		
5. Het systeem controleert of de naam van de leeractiviteit al bestaat.	Naam wordt gecontroleerd.		
6. Als de naam uniek is, slaat het systeem de leeractiviteit op en koppelt deze aan het gekozen hoofdstuk.	Leeractiviteit wordt succesvol aangemaakt.		
7. Het systeem bevestigt dat de leeractiviteit succesvol is aangemaakt.	Bevestigingsbericht verschijnt.		
3a. Als de naam van de leeractiviteit al bestaat, geeft het systeem een foutmelding en vraagt om een andere naam.	Foutmelding verschijnt.		

Use Case 5: Voortgang Bekijken

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De gebruiker logt in op de console applicatie.	Inlogschermbewerking verdwijnt, dashboard verschijnt.		
2. De gebruiker kiest de optie "Dashboard Bekijken".	Dashboard verschijnt.		
3. Het systeem toont de voortgang van de leerling in een voortgangsbalk.	Voortgangsbalk verschijnt.		
4. Het systeem toont een overzicht van behaalde badges en diploma's.	Badges en diploma's verschijnen.		
5. Het systeem toont de learning streak van de leerling.	Learning streak verschijnt.		
6. Het systeem toont een scorebord waarop de leerling zijn of haar scores kan vergelijken met die van vrienden.	Scorebord verschijnt.		
7. Het systeem toont meldingen en belangrijke mijlpalen in de voortgang van de leerling.	Meldingen verschijnen.		
8. Het systeem geeft de mogelijkheid om de voortgang van medeleerlingen te bekijken.	Optie om voortgang van medeleerlingen te bekijken verschijnt.		
9. Het systeem toont de modules waarmee de leerling momenteel bezig is.	Lijst met huidige modules verschijnt.		
10. Het systeem toont de voortgang in deze modules door middel van een voortgangsbalk.	Voortgangsbalk voor huidige modules verschijnt.		
11. Het systeem toont de modules die de leerling heeft afgerond.	Lijst met afgeronde modules verschijnt.		
3a. Als er geen voortgangsgegevens beschikbaar zijn, toont het systeem een bericht dat er nog geen data beschikbaar is.	Bericht geen data beschikbaar verschijnt.		

Use Case 6: Leeractiviteit Uitvoeren

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De leerling logt in op de console applicatie.	Inlogschermbewerking verdwijnt, dashboard verschijnt.		
2. De leerling kiest de optie "Leeractiviteit Uitvoeren".	Lijst met leeractiviteiten verschijnt.		
3. De leerling selecteert een beschikbare leeractiviteit.	Geselecteerde leeractiviteit start.		
4. Het systeem start de leeractiviteit.	Leeractiviteit wordt geladen.		
5. De leerling voltooit de leeractiviteit.	Leeractiviteit wordt voltooid.		
6. Het systeem slaat de resultaten van de leeractiviteit op.	Resultaten worden opgeslagen.		
7. Het systeem bevestigt dat de leeractiviteit succesvol is voltooid.	Bevestigingsbericht verschijnt.		
4a. Als de leeractiviteit niet geladen kan worden, geeft het systeem een foutmelding en vraagt om opnieuw te proberen.	Foutmelding verschijnt.		

Use Case 7: Reeds Gestarte Collectie Hervatten

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De leerling logt in op de console applicatie.	Inlogschermbewijnt, dashboard verschijnt.		
2. De leerling kiest de optie "Reeds Gestarte Collectie Hervatten".	Lijst met eerder gestarte collecties verschijnt.		
3. Het systeem toont de eerder gestarte collecties.	Collecties worden weergegeven.		
4. De leerling selecteert een collectie om te hervatten.	Geselecteerde collectie wordt hervat.		
5. Het systeem herstelt de voortgang van de leerling in de geselecteerde collectie.	Voortgang wordt hersteld.		
6. De leerling gaat verder met de leeractiviteiten binnen de collectie.	Leeractiviteiten worden hervat.		
7. Het systeem slaat de voortgang van de leerling op.	Voortgang wordt opgeslagen.		
3a. Als er geen eerder gestarte collecties zijn, toont het systeem een bericht dat er geen collecties beschikbaar zijn om te hervatten.	Bericht geen collecties beschikbaar verschijnt.		

Use Case 8: Voortgang Delen met Medeleerlingen

Instructie	Verwachte Resultaat	Resultaat	Geslaagd
1. De leerling logt in op de console applicatie.	Inlogschermbewerking verdwijnt, dashboard verschijnt.		
2. De leerling kiest de optie "Voortgang Delen met Medeleerlingen".	Lijst met medeleerlingen verschijnt.		
3. Het systeem toont de lijst van medeleerlingen.	Lijst medeleerlingen verschijnt.		
4. De leerling selecteert de medeleerlingen met wie de voortgang gedeeld moet worden.	Geselecteerde medeleerlingen worden gemarkeerd.		
5. Het systeem deelt de voortgangsinformatie met de geselecteerde medeleerlingen.	Voortgang wordt gedeeld.		
6. Het systeem bevestigt dat de voortgang succesvol is gedeeld.	Bevestigingsbericht verschijnt.		
3a. Als de leerling geen medeleerlingen heeft toegevoegd, toont het systeem een bericht dat er geen medeleerlingen beschikbaar zijn om voortgang mee te delen.	Bericht geen medeleerlingen beschikbaar verschijnt.		

Technisch Ontwerp.

Inleiding tot het technisch ontwerp

Dit document schetst het technische ontwerp voor de LingoPartner-applicatie, die in meerdere lagen is gestructureerd om modulariteit, onderhoudbaarheid en schaalbaarheid te garanderen. Het ontwerp omvat domeinmodellen, databaseschema's en verschillende algoritmen om gebruikersinteracties en leervoorstapen af te handelen.

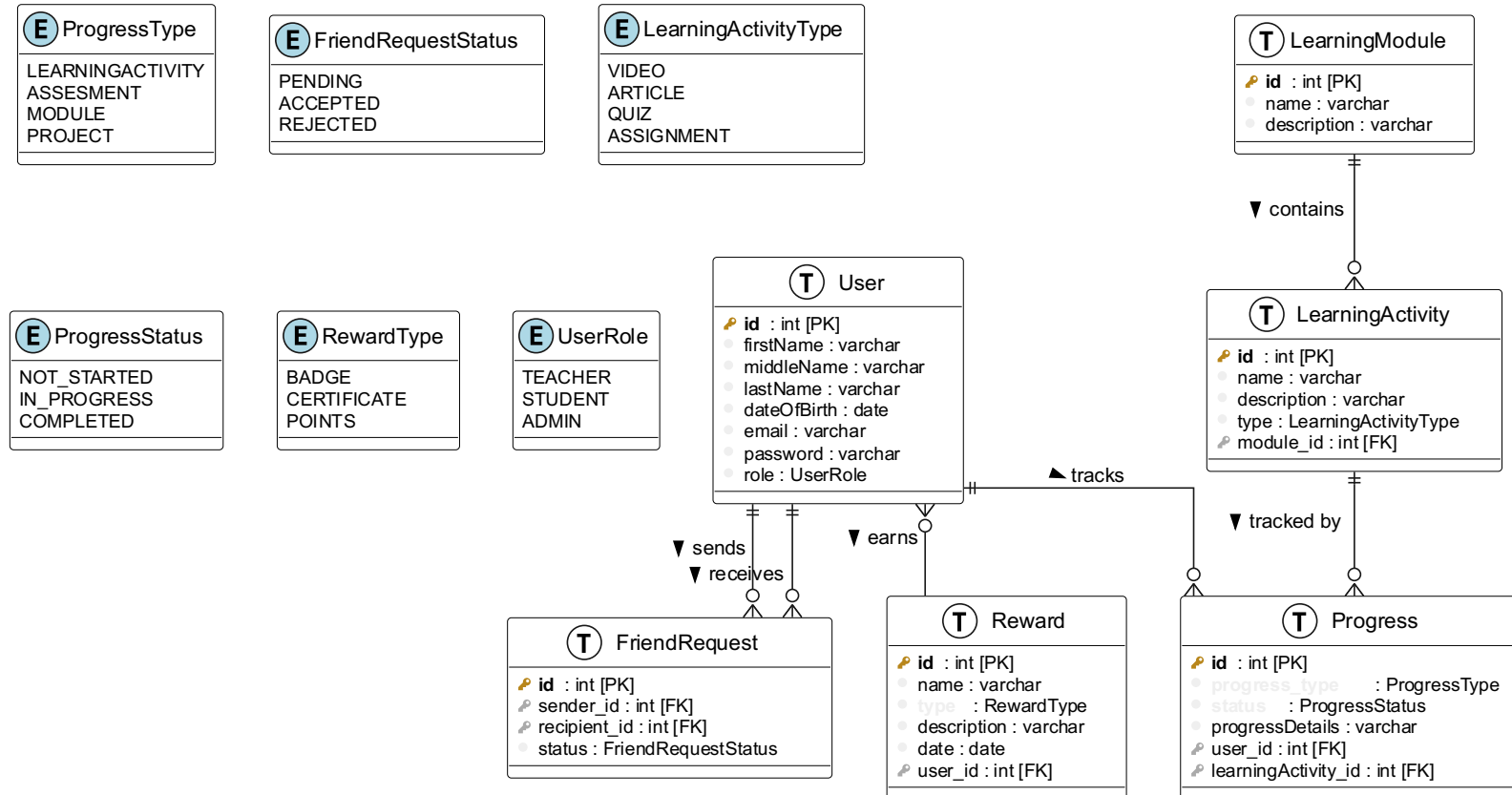
Domain Layer UML Diagram

Het Domain Layer UML-diagram toont de kernentiteiten en hun relaties binnen de LingoPartner-applicatie. De belangrijkste klassen zijn **User**, **LearningModule**, **LearningActivity**, **Progress**, en **Reward**. Enumeraties definiëren vaste sets constanten zoals gebruikersrollen, leeractiviteittypes, en voortgangsstatussen. Deze laag bevat de bedrijfslogica en regels.

Het model heeft verschillende revisies ondergaan, alsmede het pakketdiagram. In het onderdeel aannames en overwegingen zal ik daar dieper op in gaan.

Databaseschema diagram

Het databaseschemadiagram biedt een gedetailleerd overzicht van de databasestructuur, inclusief tabellen, kolommen en relaties. Dit schema waarborgt gegevensintegriteit en ondersteunt efficiënt gegevensbeheer. Ik heb het als geheugen steun gebruikt om de database op te kunnen zetten.



Pakketdiagram

Het pakketdiagram toont de organisatie van de codebase in verschillende pakketten, elk verantwoordelijk voor specifieke functionaliteit:

LingoPartnerConsole

Doel:

Bevat het toegangspunt van de applicatie en verwerkt de consolegebaseerde gebruikersinterface.

Bevat o.a.:

- Hoofdingang: Programma
- Weergaven: ConsoleDashboardView, UserList, UserAdd, LearningActivityList, LearningActivityAdd, LearningModuleList, LearningModuleAdd en andere welke de usecases ondersteunen.
- Menu: Menu, MenuHelper
- Helpers: ConsoleHelper, AuthenticatieHelper, LoggingHelper

Afhankelijk van:

- LingoPartnerDomain
- LingoPartnerInfrastructure
- LingoPartnerShared.

Toelichting:

In het LingoPartnerConsole-project wordt de consolegebaseerde gebruikersinterface geïmplementeerd. Het project bevat verschillende weergaven die de gebruiker in staat stellen om de applicatie te gebruiken. De weergaven zijn verantwoordelijk voor het weergeven van informatie en het verwerken van invoer van de gebruiker. Het project bevat ook een menuklasse die de navigatie tussen de verschillende weergaven regelt. De menuklasse maakt gebruik van een menuhelperklasse om de navigatie te vereenvoudigen. Het project bevat ook verschillende hulpprogramma's die worden gebruikt om de gebruikersinterface te ondersteunen, zoals een hulpprogramma voor authenticatie en een hulpprogramma voor logging.

LingoPartnerDomain

Doel:

Bevat kernbedrijfslogica, domeinmodellen en interfaces.

Bevat o.a.:

- Domeinmodellen: FriendRequest, LearningModule, LearningActivity, User, Progress, Reward, LearningStreak.
- Interfaces: ILearningModuleRepository, ILearningActivityRepository, IUserRepository, IProgressRepository.
- Diensten: LearningStreakService, LearningModuleService, AuthenticatieService. (hebben tevens hun eigen interfaces)
- Strategieën: SimpleScoringStrategie, BonusScoringStrategie, Opeenvolgende DagenStrategie, WeekendSkipStrategie.
- Enums: UserRole, RewardType, ProgressType, ProgressStatus, LearningActivityType, FriendRequestStatus.

Afhankelijk van:

- LingoPartnerShared.

Toelichting:

In het LingoPartnerDomain-project wordt de kernbedrijfslogica van de applicatie geïmplementeerd. Het project bevat domeinmodellen die de entiteiten van de applicatie vertegenwoordigen, zoals gebruikers, leermodules, leeractiviteiten en voortgang. Het project bevat ook interfaces die worden gebruikt om de toegang tot gegevens en services te definiëren. Daarnaast bevat het project diensten die de bedrijfslogica implementeren en strategieën die worden gebruikt om de scoring en streaks te berekenen. Het project bevat ook enumeraties die worden gebruikt om verschillende statussen en typen te definiëren.

LingoPartnerInfrastructuur

Doel:

Verzorgt de toegang tot en opslag van gegevens en implementeert opslagplaatsen en services.

Bevat:

- Opslagplaatsen: ProgressRepository, LearningActivityRepository, UserRepository, LearningModuleRepository.
- Helpers: InfraStructureHelper.

Afhankelijk van:

- LingoPartnerDomain
- LingoPartnerShared.

Toelichting:

In het LingoPartnerInfrastructure-project wordt de toegang tot en opslag van gegevens geïmplementeerd. Het project bevat opslagplaatsen die worden gebruikt om gegevens op te halen en op te slaan. Het project bevat ook hulpprogramma's die worden gebruikt om de toegang tot gegevens te vergemakkelijken.

LingoPartnerShared

Doel:

Omvat gedeelde hulpprogramma's, helpers en gemeenschappelijke klassen die in projecten worden gebruikt.

Bevat:

- algemene utility classes, uitbreidingsmethoden, gedeelde bronnen.

Afhankelijk van:

Geen.

Toelichting:

In het LingoPartnerShared-project worden hulpprogramma's, helpers en gemeenschappelijke klassen geïmplementeerd die in verschillende projecten worden gebruikt. Het project bevat algemene hulpprogramma's die worden gebruikt om de ontwikkeling van de applicatie te vergemakkelijken. Vooralsnog heeft dit project alleen een LoggingHelper en een Resultclass. De resultclass is niet gebruikt in de huidige implementatie, maar kan in de toekomst worden gebruikt om resultaten van bewerkingen terug te geven. De logginghelper wordt gebruikt om logboeken te schrijven naar de console en naar een logbestand.

LingoPartnerTests

Doel:

Gebruikt voor unit-tests, integratietests en andere geautomatiseerde tests.

Bevat:

- Testklassen zoals AdministrationTests.

Afhankelijk van:

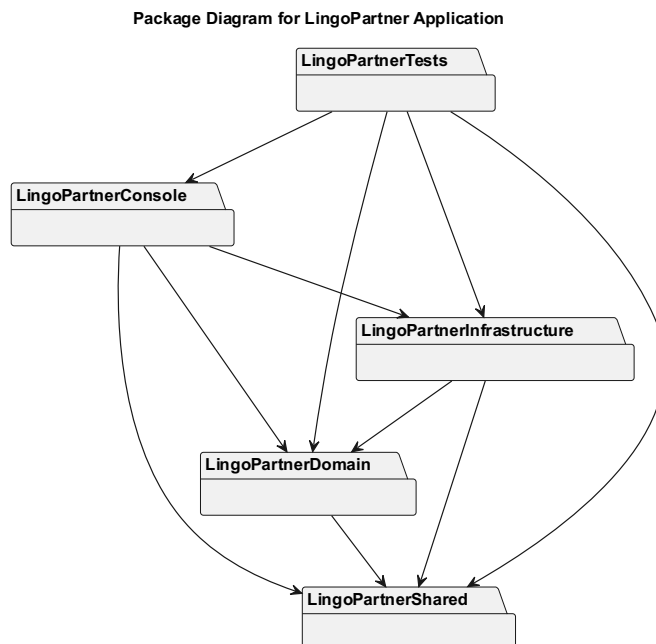
- Alle andere projecten voor testdoeleinden.

Toelichting:

In het LingoPartnerTests-project worden verschillende soorten tests geïmplementeerd, zoals unit-tests en integratietests. De testklassen worden gebruikt om de functionaliteit van de applicatie te testen en ervoor te zorgen dat de applicatie correct werkt. De testklassen zijn afhankelijk van de andere projecten in de oplossing en maken gebruik van de interfaces en klassen die in die projecten zijn geïmplementeerd.

Toelichting op PackageDiagram schema.

Hetgeen hierboven beschreven is, wordt weergegeven in het onderstaande packagediagram. Het diagram toont de structuur van de codebase en de relaties tussen de verschillende projecten. De pijlen geven de afhankelijkheden tussen de pakketten aan. Je ziet dat het Console project afhankelijk is van het Domain, Infrastructure en Shared project. Het Domain project is afhankelijk van het Shared project en het Infrastructure project. Het Infrastructure project is afhankelijk van het Domain en Shared project. Het Shared project is niet afhankelijk van andere projecten. Het Tests project is afhankelijk van alle andere projecten voor testdoeleinden.



Figuur 3 Package Diagram

Algoritme

Het “LearningStreak” Algoritme.

In de volgende paragraaf wordt de interne werking van het algoritme uitgelegd. Ik heb een specifieke service voor dit algoritme gemaakt, genaamd ‘LearningStreakService’. Ik dacht dat het een goed idee zou zijn om een specifieke klasse te hebben die verantwoordelijk is voor het LearningStreak-algoritme.

Het algoritme heeft een paar ingrediënten nodig om goed te kunnen werken: het moet de gebruiker kennen, het moet weten wanneer (niet noodzakelijkerwijs precies wat) een gebruiker interactie had of activiteiten deed met de app. Wanneer een gebruiker iets doet, wordt dit in de database bijgehouden in de “progress” tabel. Om het simpel te houden, zijn we alleen geïnteresseerd in de datum van de interactie en maken we indien nodig een lijst met unieke data.

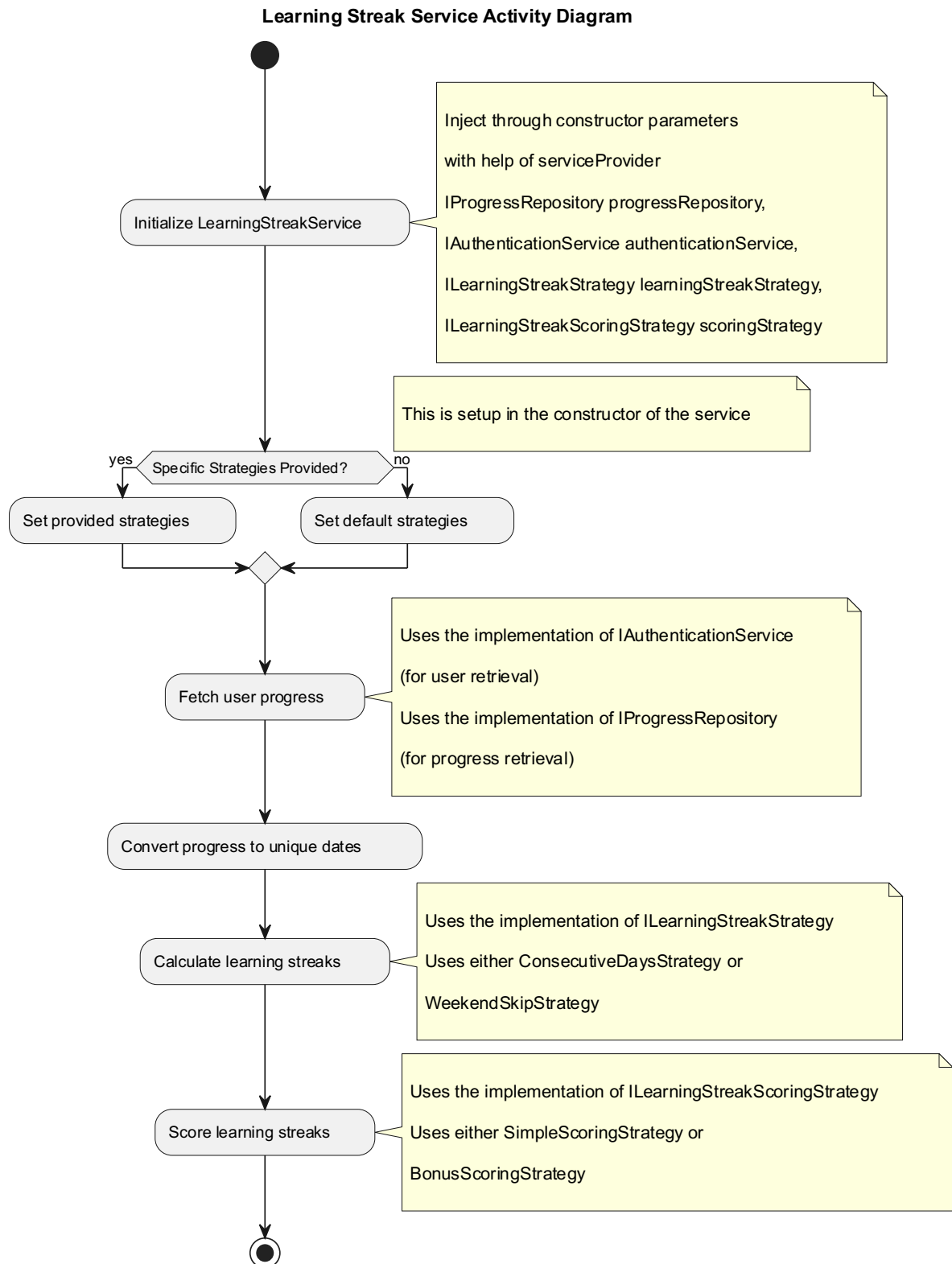
Terwijl we gebruik maken van de serviceprovider, kunnen we de ProgressRepository en de UserAuthenticationService injecteren om de benodigde gegevens te verkrijgen. Ik vind dit soort abstracties wel leuk. Voor het gebruiksgemak heb ik dit als toelichting in het diagram vermeld.

LearningStreakService

Ik wil graag één plek hebben die verantwoordelijk is voor “leerresultaten”. Dit is de reden waarom ik de LearningStreakService heb gemaakt. Deze dienst is verantwoordelijk voor het berekenen van de “LearningStreaks” en het scoren ervan.

We kunnen de “progressRepository” en “userAuthenticationService” in de constructor van de “LearningStreakService” injecteren en we hoeven ons geen zorgen te maken over hun exacte implementatiedetails; de interne werking wordt beschreven in interfaces.

Ik heb alles samengebracht in een activiteitendiagram. Dit diagram toont de stappen die de “LearningStreakService” neemt om de leerreeksen te berekenen en te scoren. Het is een leuke manier om het algoritme te visualiseren en te zien hoe de verschillende onderdelen samenwerken.



Figuur 4 LearningStreakService activiteiten diagram

LearningStreak

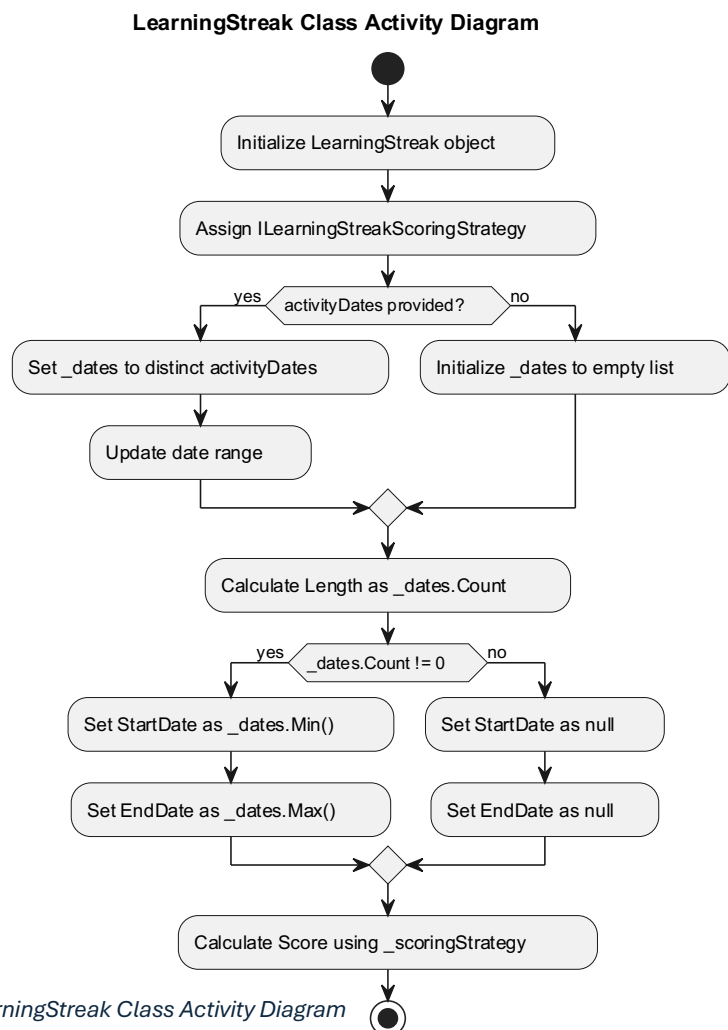
Als we het hebben over een “LearningStreakService” die meerdere leerreeksen afhandelt, hebben we een klasse nodig die verantwoordelijk is voor één enkele leerreeks. Dit is de reden waarom ik de “LearningStreak” klasse heb gemaakt.

Deze klasse is verantwoordelijk voor het bijhouden van de leerreeks van een gebruiker. Het heeft een paar eigenschappen, zoals de gebruiker, de startdatum en de einddatum van de streak. Het heeft ook een score-eigenschap, die wordt berekend door de scorestrategie. Het is niet zo spannend, maar het is een leuke manier om alle informatie over een leerreeks op één plek te bewaren. Het is ook een leuke manier om het algoritme schoon en gemakkelijk te begrijpen te houden. Op deze manier is er een klasse die verantwoordelijk is voor één enkele leerreeks, en die complexiteit wordt niet vermengd met de leerreeksservice.

Ik heb de “scoringStrategy” geïmplementeerd in de “LearningStreak-klasse”. Hoewel ik het niet actief gebruik, is het er voor toekomstig gebruik. Op dit moment gebeurt de scoring in de “LearningStreakService”, dus alles staat op één plek.

LearningStreak Activity Diagram

Het activiteitendiagram van de “LearningStreak” klasse toont de stappen die de “LearningStreak” klasse neemt om te initialiseren, de lengte te berekenen, de begin- en einddatum in te stellen en, indien nodig, de score te berekenen.

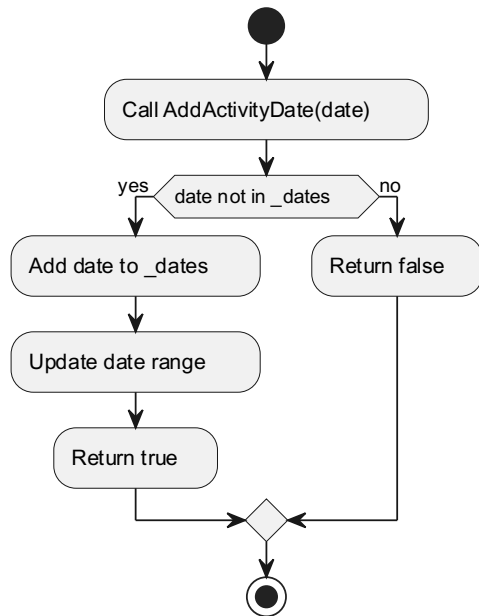


Figuur 5 LearningStreak Class Activity Diagram

AddActivityDate Method Activity Diagram

Just needed to add a date to the list, makes sure it is unique, and updates the date range.

AddActivityDate Method Activity Diagram

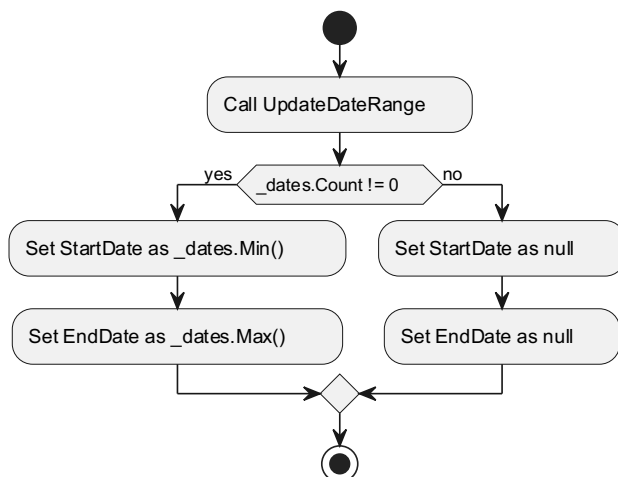


Figuur 6 AddActivityDate Method ActivityDiagram

UpdateDateRange Method Activity Diagram

Sets the start and end date of the streak based on the activity dates.

UpdateDateRange Method Activity Diagram

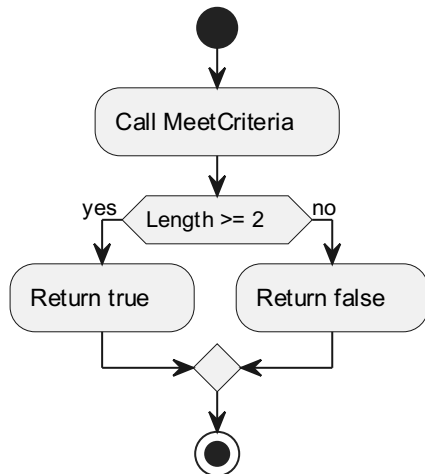


Figuur 7 UpdateDateRange Method ActivityDiagram

MeetCriteria Method Activity Diagram

Een methode die specifiek controleert of een “learningStreak” aan specifieke voorwaarden voldoet. In dit geval moet de learningstreak uit ten minste 2 dagen bestaan

MeetCriteria Method Activity Diagram



Figuur 8 MeetCriteria Method Activity Diagram

hoe zat het ook alweer met de strategieën? Ik zal ze in de volgende paragraaf uitleggen.

Strategy patterns.

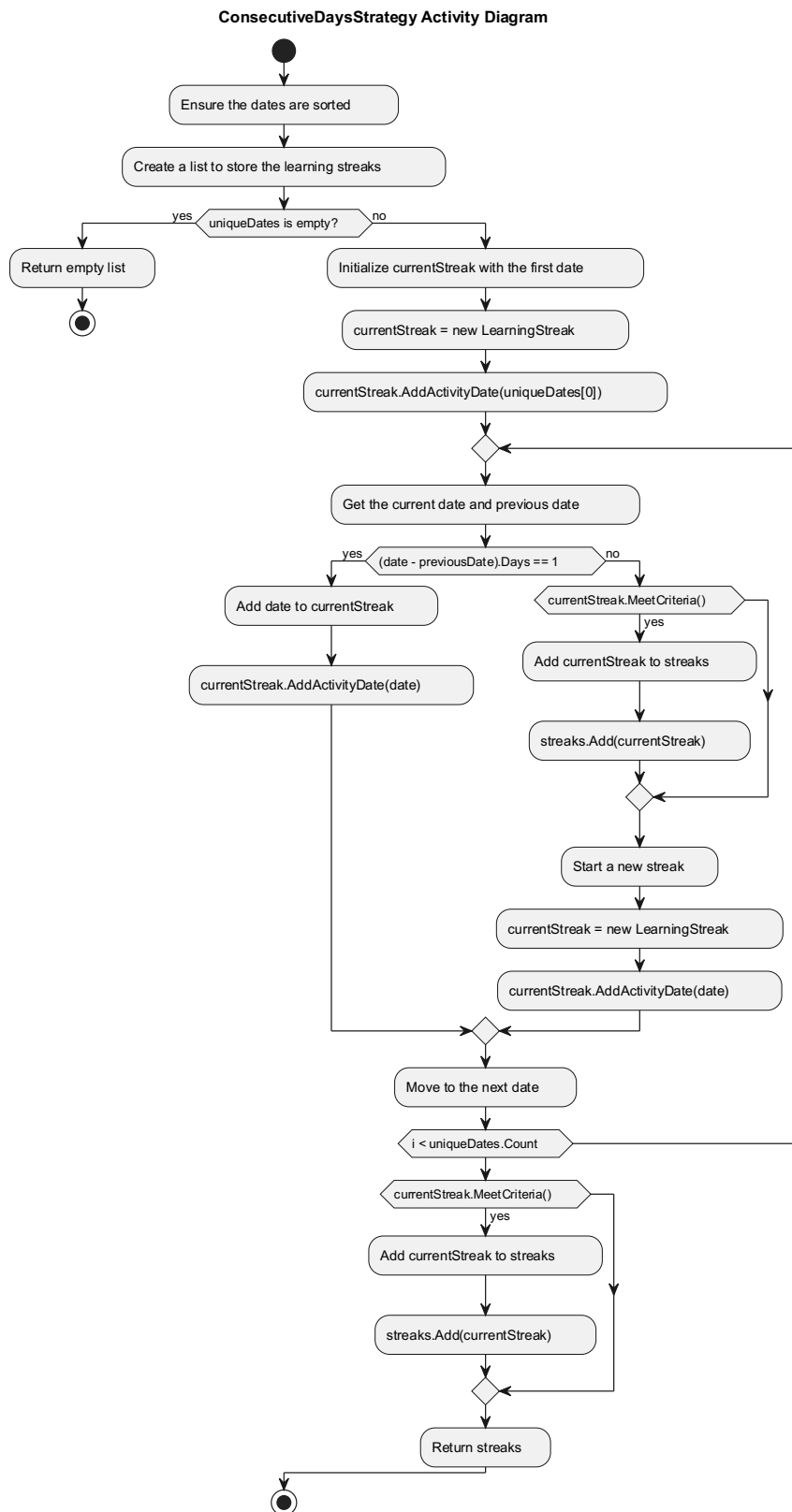
Het learning streak-algoritme is gebaseerd op een aantal strategieën. Deze strategieën zijn verantwoordelijk voor het verzamelen van “LearningStreaks” en er zijn die verantwoordelijk zijn voor het scoren van de “LearningStreaks”.

LearningStrategieën

Ik heb twee strategieën ontwikkeld, de ConsecutiveDaysStrategy en de WeekendSkipStrategy. De ConsecutiveDaysStrategy is verantwoordelijk voor het berekenen van de leerreeksen op basis van opeenvolgende dagen. De WeekendSkipStrategy is verantwoordelijk voor het berekenen van de leerreeksen op basis van het overslaan van de weekenden.

ConsecutiveDaysStrategy

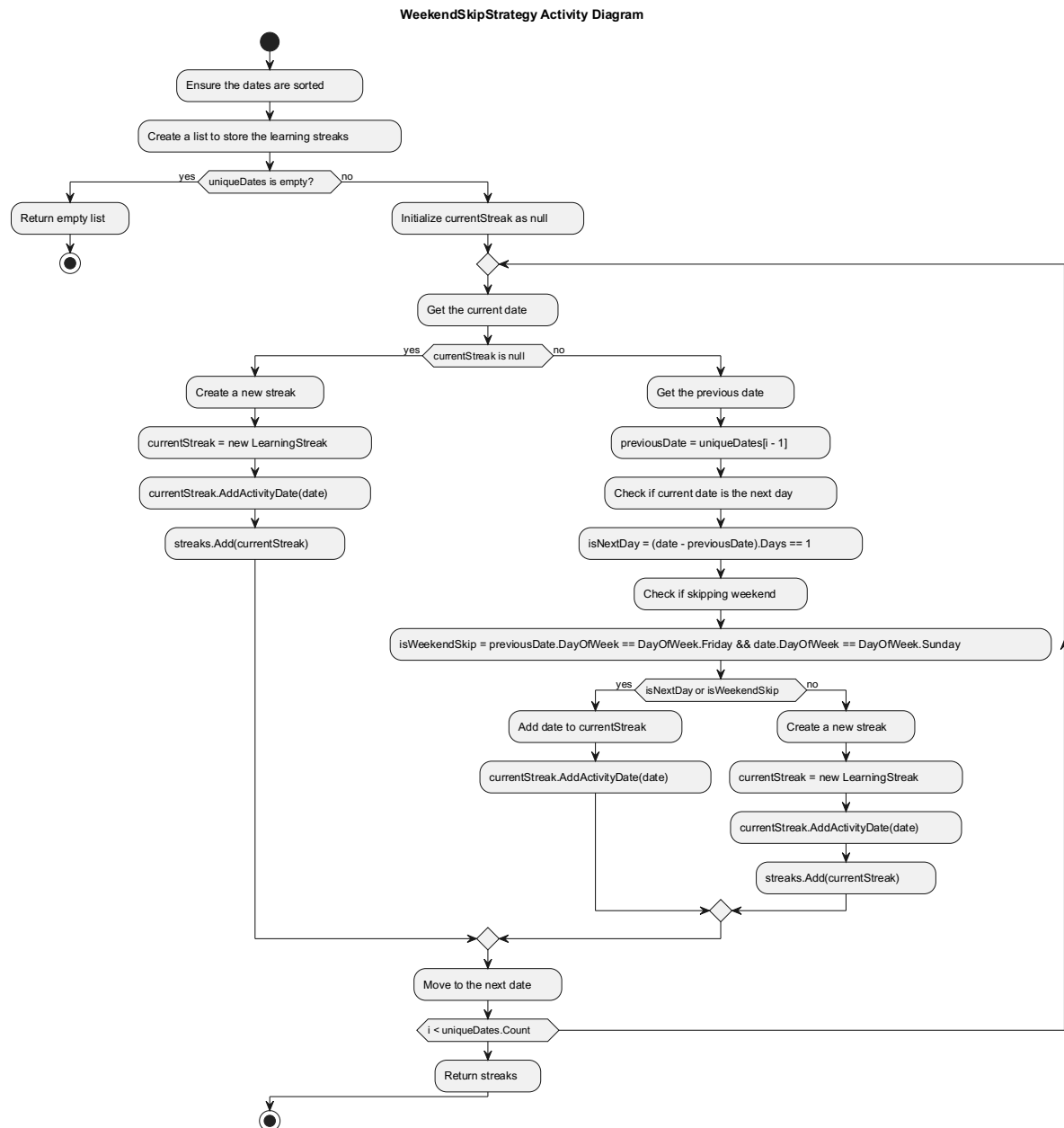
Berekent leerreeksen op basis van opeenvolgende dagen van activiteit.



Figuur 9 Calculate Learning Streaks with WeekendSkipStrategy

WeekendSkipStrategy

Maakt het mogelijk om weekenden over te slaan zonder de leerreeks te onderbreken.



Figuur 10 Calculate learning Streaks With WeekendSkipStrategy

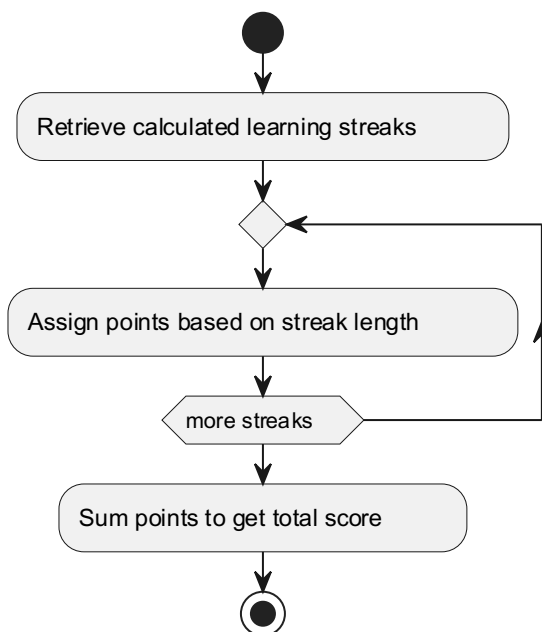
Scorestrategieën

Er zijn twee scorestrategieën, de **SimpleScoringStrategy** en de **BonusScoringStrategy**. De **SimpleScoringStrategy** is verantwoordelijk voor het scoren van de leerreeksen op basis van de lengte van de reeks. De **BonusScoringStrategy** is verantwoordelijk voor het scoren van de leerreeksen op basis van de lengte van de reeks en het aantal activiteiten.

SimpleScoringStrategie

Wijst een score toe op basis van de lengte van de leerreeks.

Score Learning Streaks with SimpleScoringStrategy

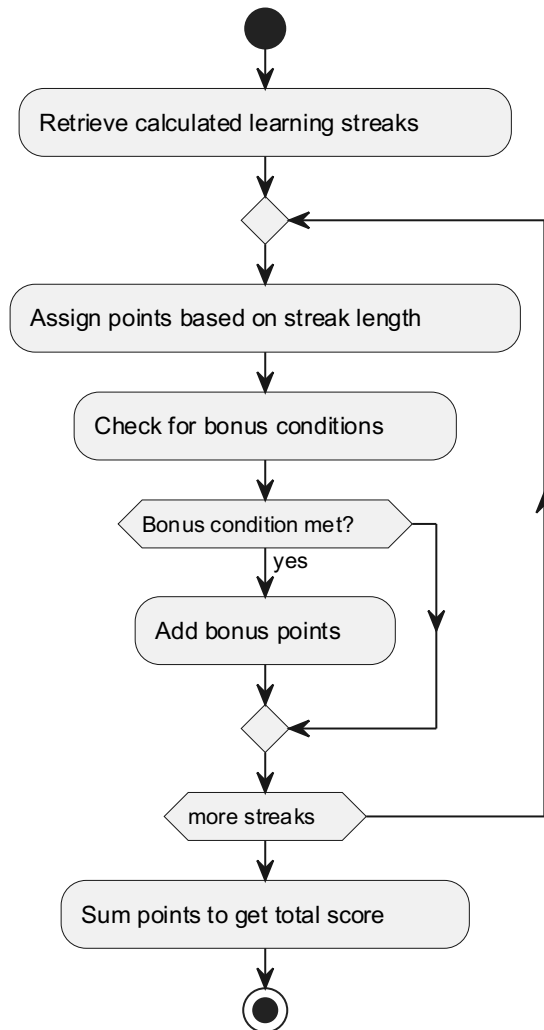


Figuur 11 Score Learning Streaks with SimpleScoringStrategy

BonusScoringStrategy

Wijst extra bonuspunten toe voor langere streaks of speciale omstandigheden.

Score Learning Streaks with BonusScoringStrategy



Figuur 12 Score Learning Streaks With BonusScoringStrategy

Toelichting op algoritme.

Waarom zouden we al dit gedoe doen? Welnu, we willen het algoritme flexibel en gemakkelijk te onderhouden houden. We kunnen verschillende strategieën gebruiken om de leerreeksen te berekenen en te scoren. We kunnen eenvoudig tussen strategieën schakelen door ze aan te bieden in de constructor van de LearningStreakService. Ook in de toekomst kunnen we eenvoudig nieuwe strategieën toevoegen zonder de bestaande code te wijzigen. Nu hebben we een mooie en schone service die verantwoordelijk is voor het leerreeksalgoritme. Wij kunnen deze dienst eenvoudig gebruiken in onze verwerkingsverantwoordelijken of andere diensten.

Bij het omgaan met meerdere learningStreaks is alle logica ingekapseld in de service. Als we te maken hebben met een enkele LearningStreak, kunnen we de LearningStreak-klasse gebruiken.

Deze abstractie zou het eenvoudig moeten maken om het algoritme te testen. We zouden de afhankelijkheden gemakkelijk moeten kunnen bespotten en het algoritme met verschillende scenario's kunnen testen.

Het algoritme is niet extreem complex, maar het is een mooi voorbeeld van hoe je abstracties en interfaces kunt gebruiken om je code flexibeler en onderhoudbaarder te maken.

Aannames en Overwegingen

Tijdens de ontwikkeling van de LingoPartner-applicatie zijn enkele belangrijke aannames en overwegingen gemaakt die de architectuur en het ontwerp hebben beïnvloed. Het model heeft verschillende revisies ondergaan, evenals het pakketdiagram. In dit onderdeel wil ik hier dieper op ingaan.

Evolutie van het Project

In het begin van het project had ik een eenvoudiger package diagram. Het bestond uit de domeinlaag, de infrastructuur en de console. Er was geen shared project en geen tests project. Strategieën en specifieke services voor streaks ontbraken eveneens. Alle data werd onbewust verzameld in de `Administration` klasse, waar de repository's direct werden geïnitieerd. Deze aanpak bleek onhandig aangezien ik nu een klasse had die voor bijna alles verantwoordelijk was. Testen was hierdoor eveneens onhandig. Later ben ik “Reverse Dependency Injection” gaan toepassen. Wat ik in een latere paragraaf toelicht.

SOLID-principes en Dependency Injection

Tijdens een les werd het principe van dependency inversion uitgelegd, wat een eye-opener was. Dit principe maakt deel uit van de SOLID-principes, die een leidraad vormen voor goed objectgeoriënteerd ontwerp. De SOLID-principes zijn:

- **Single Responsibility Principle (SRP):** Een klasse zou slechts één verantwoordelijkheid moeten hebben.
- **Open/Closed Principle (OCP):** Software-entiteiten moeten open zijn voor uitbreiding, maar gesloten voor modificatie.
- **Liskov Substitution Principle (LSP):** Subtypes moeten kunnen worden vervangen door hun base types zonder de correctheid van het programma aan te tasten.
- **Interface Segregation Principle (ISP):** Klassen zouden niet gedwongen moeten worden om interfaces te implementeren die ze niet gebruiken.
- **Dependency Inversion Principle (DIP):** Hoog-niveau modules moeten niet afhankelijk zijn van laag-niveau modules. Beide moeten afhankelijk zijn van abstracties.

Met reverse dependency injection kun je je codebase flexibeler en testbaarder maken. Door een interface te maken en deze te injecteren in de klasse die de interface nodig heeft, wordt testen eenvoudiger. Je kunt een mock maken van de interface en deze injecteren in de klasse die je wilt testen. Dit is een van de redenen waarom ik de LingoPartner applicatie heb aangepast.

Toepassing van DIP en Dependency Injection

Ik heb de afhankelijkheden van de console omgedraaid. In het oorspronkelijke package diagram stonden de pijlen van de view layer naar de domain layer en van de domainlayer naar de infrastructuurlaag. Dit heb ik veranderd. Nu staan de pijlen van de `LingoPartnerConsole` (view layer) en de `LingoPartnerInfrastructure` (infrastructuurlaag) naar de `LingoPartnerDomain` (domainlaag). De `LingoPartnerConsole` en de `LingoPartnerInfrastructure` zijn nu afhankelijk van de `LingoPartnerDomain`. Dit maakt de codebase flexibeler en testbaarder. Ik hoef ook geen data transfer objects te gebruiken wat voorheen wel nodig was.

Dependency injection heeft een aantal voordelen die nauw aansluiten bij de SOLID-principes:

- **Eenvoudiger onderhoud (SRP):** Wijzigingen hoeven maar op één plek doorgevoerd te worden.
- **Betere testbaarheid (ISP, DIP):** Makkelijk testen door onderdelen te vervangen met testversies.
- **Loose coupling (OCP):** Onderdelen zijn minder afhankelijk van elkaar, wat de code flexibeler maakt.
- **Herbruikbaarheid (LSP):** Centraal beheerde onderdelen kunnen makkelijk hergebruikt worden.

Herstructurering van de Administration Klasse (SRP, OCP)

In de beginfase was de `Administration` klasse verantwoordelijk voor veel functionaliteiten, wat tegen de "single responsibility" van SOLID inging. Ik heb de verantwoordelijkheden van de `Administration` klasse herverdeeld naar andere klassen, zoals een aparte service voor authenticatie en een aparte service voor streaks. Dit zorgt voor een flexibeler en testbaarder codebase. Wanneer ik een geauthenticeerde gebruiker nodig heb, kan ik de authenticatie service injecteren in de klasse die de geauthenticeerde gebruiker nodig heeft, wat testen eenvoudiger maakt.

Gebruik van ServiceCollection in C# (DIP, ISP)

In C# kun je een `ServiceCollection` gebruiken om afhankelijkheden te injecteren. Dit biedt voordelen zoals loose coupling, betere testbaarheid en onderhoudbaarheid. Het implementeren van deze principes heeft geleid tot een meer robuuste en onderhoudbare codebase voor de LingoPartner-applicatie. Door de verantwoordelijkheden te scheiden en afhankelijkheden om te keren, is de applicatie flexibeler en beter bestand tegen veranderingen geworden.

BIJLAGE.

Lijst van Afbeeldingen.

Figuur 1 Conceptueel model	11
Figuur 2 Uml klassendiagram	34
Figuur 3 Package Diagram	38
Figuur 4 LearningStreakService activiteiten diagram	40
Figuur 5 LearningStreak Class Activity Diagram	41
Figuur 6 AddActivityDate Method ActivityDiagram	42
Figuur 7 UpdateDateRange Method ActivityDiagram	42
Figuur 8 MeetCriteria Method Activity Diagram	43
Figuur 9 Calculate Learning Streaks with WeekendSkipStrategy	45
Figuur 10 Calculate learning Streaks With WeekendSkipStrategy	46
Figuur 11 Score Learning Streaks with SimpleScoringStrategy	47
Figuur 12 Score Learning Streaks With BonusScoringStrategy	48

Plantumlcode: conceptueel model.

```
@startuml
rectangle Leerling
rectangle Leerkracht
rectangle Vriend
' rectangle Educatievelnhoud as "Educatieve Inhoud"
' rectangle Collectie
' rectangle Hoofdstukken
rectangle Leeractiviteit
rectangle Voortgang
rectangle Beloning
' rectangle Diploma
' rectangle Badge
' rectangle SocialeInteractie as "Vriendenlijst"
' rectangle LeerlingDashboard as "Leerling Dashboard"
```

```
Leerkracht "1..n" -down-> "1..n" Leeractiviteit : levert
' Educatievelnhoud -down-> Collectie : bestaat uit
' Collectie -right-> Hoofdstukken : omvat
' Hoofdstukken -down-> Leeractiviteit : bevat
' Educatievelnhoud "1" -down-> "1..n" Leeractiviteit : bevat
```

```
Leerling "1" -left-> "1..n" Leeractiviteit : "neemt deel aan\n/ voert uit"
Leerling "1" -right-> "0..n" Vriend : heeft
' Leerling "1" -down-> "1" LeerlingDashboard : heeft
Leeractiviteit "1" -down-> "1" Voortgang : levert
Voortgang "1" -up-> "1" Leerling : heeft
Leerling "1" -down-> "1..n" Beloning : heeft
Voortgang "1" -right-> "1..n" Beloning : heeft
' Diploma "1" -up-> "n" Voortgang : heeft
' Badge "1" -up-> "1" Voortgang : heeft
' Beloning "1..n" -up-> "1" Diploma : is
' Beloning "1..n" -up-> "1" Badge : is
' Beloning "n" -right-> "1" LeerlingDashboard : toont
@enduml
```

Plantuml: usecase diagram.

@startuml

left to right direction

actor Admin

actor Leerling

actor Vriend

actor Leerkracht

rectangle "LingoPartner App" {

usecase "Gebruiker Aanmaken" as UC1

usecase "Profiel Aanpassen" as UC2

usecase "Collectie aanmaken" as UC3

usecase "Leeractiviteit Aanmaken" as UC4

usecase "Voortgang bekijken" as UC5

usecase "Leeractiviteit Uitvoeren" as UC6

usecase "Woorden Leren" as UC7

usecase "Invuloefening en Meerkeuzevraag Uitvoeren" as UC8

usecase "Reeds Gestarte Collectie Hervatten" as UC9

usecase "Voortgang Delen met Vrienden" as UC10

Admin --> UC1

Leerling --> UC2

Leerling --> UC5

Leerling --> UC6

Leerling --> UC7

Leerling --> UC8

Leerling --> UC9

Leerling --> UC10

Leerkracht --> UC1

Leerkracht --> UC2

Leerkracht --> UC3

Leerkracht --> UC4

Leerkracht --> UC5

Vriend --> UC5

Vriend <-- UC10

}

@enduml