

Week 1 – opdracht OOP Python – Receptenboek

Een anonieme opdrachtgever zegt een *million dollar idea* te hebben en heeft jou geselecteerd als de uitverkorene! Het idee is om een receptenboek te maken, waarbij mogelijk is om recepten toe te voegen, ingrediënten eraan te koppelen en stappen te noteren om het te bereiden.

Goed plan: het droomidee van de opdrachtgever wordt werkelijkheid, jij scherpt je OOP-vaardigheden aan en wellicht doe je gelijk al wat inspiratie op voor jezelf (qua recepten!). Win-win. Noot: het wordt een console applicatie en niet een GUI applicatie... het is immers een *million dollar* en niet een *billion dollar idea*.

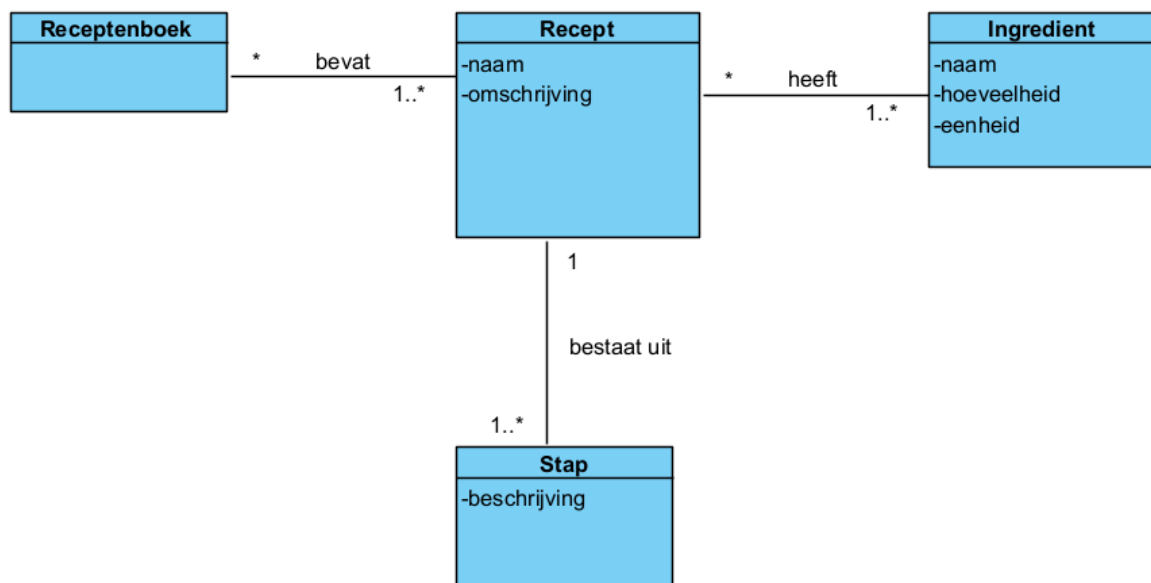
Er zijn een aantal must-have functionele requirements opgesteld, te weten:

- Het systeem moet een overzicht kunnen laten zien van recepten met ingrediënten en bereidingsstappen [FR-1].

Met de volgende acceptatiecriteria (= duidelijke, meetbare voorwaarden waaraan een functionaliteit moet voldoen om “voltooid” en geaccepteerd te worden door de opdrachtgever):

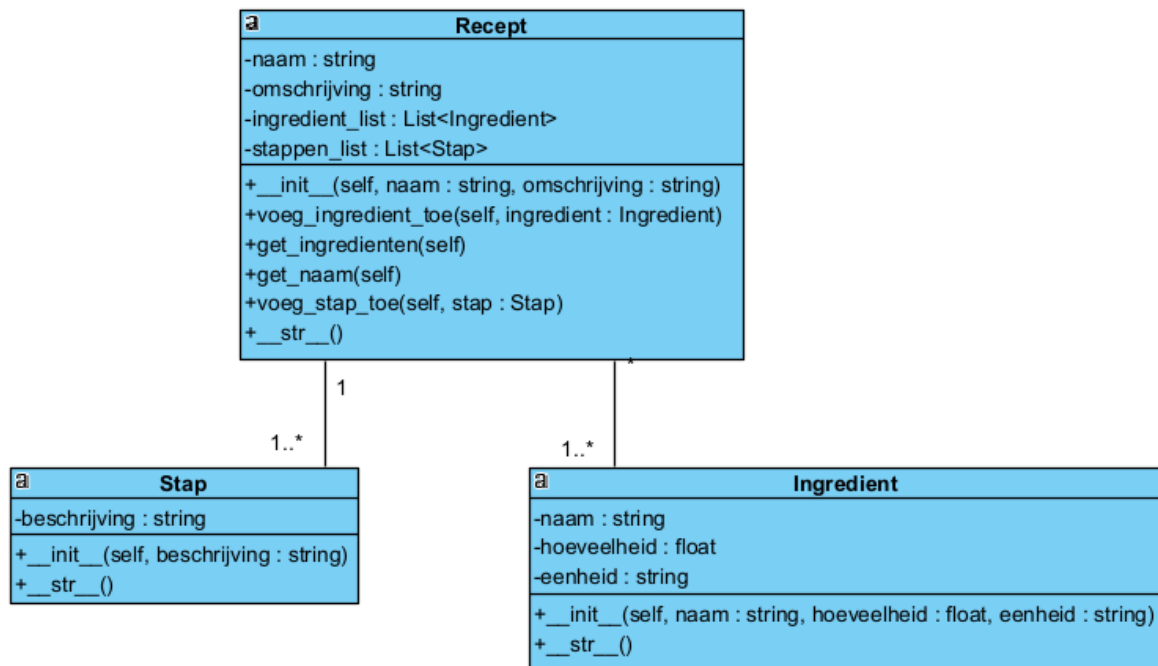
- De gebruiker krijgt, bij opstart van de applicatie, een genummerde lijst te zien met alleen de naam van recepten;
- Wanneer de gebruiker een recept selecteert wordt de naam, omschrijving, ingrediënten en bereidingsstappen weergegeven, in deze volgorde;
- Ingrediënten moeten als een lijst met bullets worden weergegeven;
- Stappen moeten als een genummerde lijst worden weergegeven;
- Alle ingrediënten (hoeveelheden) zijn op basis van 1 persoon.

Om het probleemdomein/de context inzichtelijk te maken is er een domeinmodel opgesteld.



Zo zien we dat een receptenboek één of meerdere recepten heeft en dat een recept één of meerdere ingrediënten bevat. Daarnaast bestaat een recept ook uit één of meerdere bereidingsstappen.

Voor de (technische) implementatie is er een klassendiagram opgesteld. Want een domeinmodel is een conceptueel model, dat weergeeft **wat** er in het domein speelt en het is dus geen technisch model dat weergeeft **hoe** de software geïmplementeerd moet worden.



Wat extra toelichting: Recept, Stap en Ingredient zijn allemaal klassen. Wat hier niet op staat en ook geen klasse is, is de bestand `main.py`, wat bestaat uit een `main()`-functie waarin de applicatie start (conform de Python-conventie). In deze `main()`-functie maak je een lijst met recepten aan en wordt het overzicht geprint.

Er is ook voor gekozen om iedere klas in een eigen bestand te zetten.

Week 1 – opdracht

Er is een analyse gedaan (requirementsanalyse) en er zijn acceptatiecriteria opgesteld om het te meten dan wel te testen. Het domein is inzichtelijk gemaakt door het ontworpen domeinmodel. De structuur van de code is ook ontworpen (klassendiagram).

Jouw opdracht is om de applicatie te realiseren aan de hand van deze analyse en ontwerpen. Kijk goed naar wat er verwacht wordt op functioneel en technisch gebied. Houd rekening met de volgende aanvullende eisen:

- Zorg voor 3 **volledig uitgewerkte** recepten die jij wel eens kookt of interessant lijkt. Dat wil zeggen: echte data en geen dummy/placeholder data!
- Iedere klasse in een eigen bestand.
- Een main-bestand waarin de `main()`-functie wordt aangeroepen en waarbij dus alle code wordt geïnitieerd.

Alle programmeerwerk wordt gedaan in het mapje “week 1”.

Week 2 – opdracht OOP Python – Receptenboek

Ah, onze anonieme opdrachtgever komt met nieuwe ontwikkelingen! Schijnt dus dat hij een uitbreiding wil op deze fantastische receptenboek applicatie. Dat is prima, het is maar goed dat we lekker *agile* bezig zijn en we iteratief werken. Afijn, wat moet er uitgebreid worden? Nou het volgende: het zou fijn zijn als we voor ieder recept het aantal personen mee kunnen geven en dat, afhankelijk van het aantal, de hoeveelheden van ingrediënten ook worden aangepast. Alle ingrediënten zijn in de eerste instantie voor 1 persoon. Ook zou het mooi zijn om een plantaardig alternatief (indien toepasbaar) te koppelen aan een ingrediënt (bijvoorbeeld aquafaba voor slagroom). Iets van voedingswaarden (kcal) is ook gewenst en de opdrachtgever wil de mogelijkheid hebben om een tip te kunnen toevoegen aan een bereidingsstap.

Al met al zijn de volgende must-have requirements opgesteld:

- Het systeem moet aan de hand van de opgegeven aantal personen bij een recept de ingrediëntenlijst en de hoeveelheden bijwerken [**FR-2**];
- Het systeem moet plantaardige alternatieven aan een ingrediënt kunnen koppelen [**FR-3**];
- Een ingrediënt moet van kcal voorzien zijn [**FR-4**];
- Het systeem moet ondersteuning bieden om een tip aan een stap te koppelen [**FR-5**];

Met de volgende acceptatiecriteria:

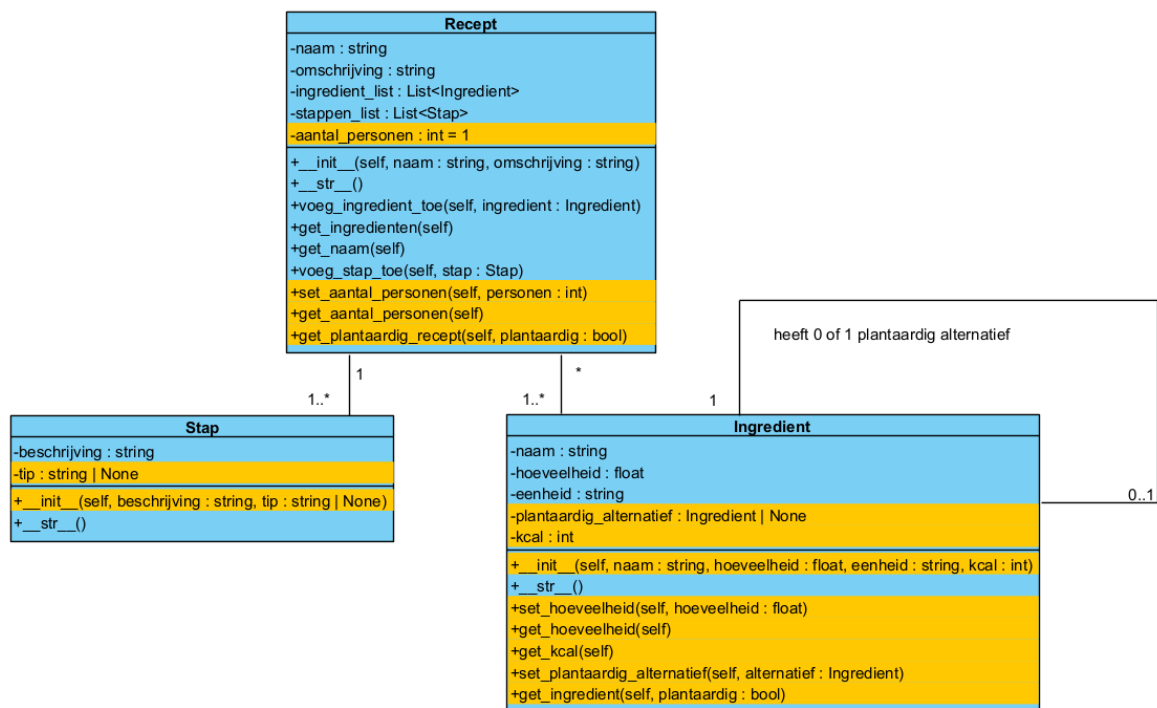
- De gebruiker kan aangegeven voor hoeveel personen het recept is;
- Vervolgens kan de gebruiker aangeven of de plantaardige versie gewenst is (indien mogelijk)
- Aan het eind wordt het totaal aantal kcal weergegeven.

Om het te verduidelijken is er een use case beschrijving en activiteitendiagram gemaakt om het proces inzichtelijker te maken.



Naam	Selecteren recept
Actor(en)	Primair: (thuis)kok
Precondities	Het systeem moet voorzien zijn van recepten.
Postcondities	Het systeem toont het volledige recept, inclusief ingrediënten, bereidingsstappen en evt. tips en voedingswaarden (kcal).
Hoofdscenario	<ol style="list-style-type: none"> 1. De thuis kok opent de applicatie. 2. Het systeem toont een overzicht van alle recepten. 3. De thuis kok selecteert een recept. 4. Het systeem vraagt aan de thuis kok om een aantal. De thuis kok voorziet het van een aantal. 5. Het systeem vraagt aan de thuis kok of plantaardig gewenst is. De thuis kok reageert met 'ja'. 6. Het systeem toont het recept + voedingswaarde.
Uitzonderingen	<p>3. <foutief of niet bestaand recept geselecteerd> Het systeem toont de melding: "Recept niet gevonden." en voert vervolgen stap 2 uit.</p> <p>4 en 5. <foutieve invoer> Het systeem toont de melding: "Foutieve invoer." en herhaalt de stap.</p>

Het klassendiagram is aangevuld met de nieuwe/bijgewerkte requirements.



Alles wat oranje gemarkeerd is, is toegevoegd t.o.v. week 1. Een paar aandachtspunten:

- Attribuut `aantal_personen` in `Recept` staat standaard op 1 (ingrediënten zijn op basis van 1 persoon);
- Attribuut `plantaardig_alternatief` bevat één `Ingredient` object of niks (`None`);
- Wanneer de `set_aantal_personen()`-methode in `Recept` wordt aangeroepen wordt de hoeveelheid van de verschillende `Ingredient`en bijgewerkt (a.d.h.v. de `set_hoeveelheid()`-methode);

- Een bereidingsstap kan of een tip bevatten of niet;
- `get_ingredient()`-methode in klasse `Ingredient` zorg ervoor dat als er `True` wordt meegegeven, de plantaardige variant wordt opgehaald en bij `False` niet de variant.
- Als het nodig is mag je extra methodes aanmaken

Week 2 – opdracht

Jouw opdracht is om de applicatie te realiseren aan de hand van deze analyses en ontwerpen. Kijk goed naar wat er verwacht wordt op functioneel en technisch gebied.

Alle programmeerwerk wordt gedaan in het mapje “week 2”. Kopieer je bestanden uit “week 1”. Je gaat niet verder in “week 1”, dit is om het proces te laten zien (door de weken) voor de docent.

Week 3 – opdracht OOP Python – Receptenboek

Hoe is het mogelijk! Onze anonieme opdrachtgever wil nogmaals een uitbreiding op de applicatie. Zijn *million dollar idea* loopt al flink op in de kosten. Desalniettemin wil hij dat het keuzemenu (wat nu eigenlijk alleen maar bestaat uit het selecteren van een recept) wordt uitgebreid met functionaliteiten: toevoegen en verwijderen van recepten. Prima, zolang hij maar blijft betalen...

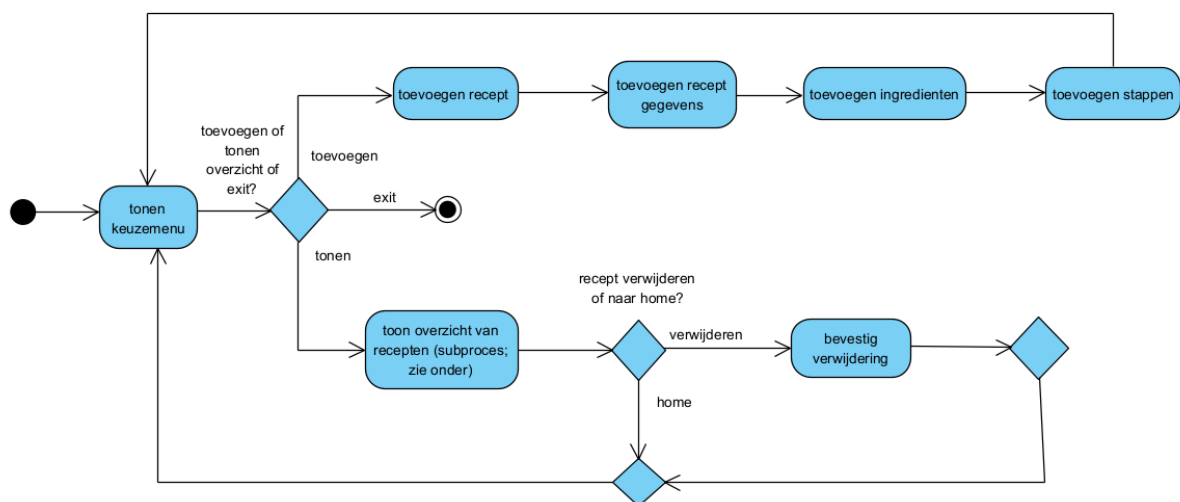
De must-have requirements zijn als volgt:

- Het systeem moet in staat zijn om, aan de hand van gebruikersinvoer, recepten te kunnen...
 - toevoegen **[FR-6]**;
 - verwijderen **[FR-7]**.
- Het keuzemenu biedt de mogelijkheid om de eerdergenoemde activiteiten (toevoegen en verwijderen) uit te voeren (door de gebruiker) **[FR-8]**.

En dit zijn de acceptatiecriteria:

- De gebruiker krijgt in de eerste instantie een keuzemenu gepresenteerd: tonen overzicht en toevoegen recept.
 - Bij het tonen van een recept krijgt de gebruiker naderhand de optie om het recept te verwijderen of weer terug te gaan naar het keuzemenu.
- Bij het toevoegen van een recept kunnen alle gegevens (naam, omschrijving, ingrediënten, hoeveelheden, tips, plantaardige alternatieven, etc. worden toegevoegd.
- Bij het verwijderen wordt het recept verwijderd uit het overzicht.

Hoog over wordt dit de flow:



Het “toon overzicht van recepten” activiteit/subproces verschilt niet van die van week 2. Voor de volledigheid is het wel opgenomen, zie onder.



Het toevoegen van een recept bestaat uit verschillende stappen. Om deze stappen te verduidelijken is een use case beschrijving gemaakt.

Naam	Toevoegen recept
Actor(en)	Primair: (thuis)kok
Precondities	-
Postcondities	Het systeem bevat het nieuwe recept
Hoofdscenario	<ol style="list-style-type: none"> 1. De thuiskok opent de applicatie. 2. Het systeem toont een keuzemenu voor het toevoegen of tonen van recepten. 3. De thuiskok kiest toevoegen. 4. Het systeem vraagt aan de thuiskok de naam en omschrijving van het recept. 5. De thuiskok voert de gevraagde gegevens in. 6. Het systeem vraagt aan de thuiskok om de gegevens van een ingrediënt. 7. De thuiskok voert de gevraagde gegevens in. 8. Het systeem vraagt aan de thuiskok of plantaardig gewenst is. 9. De thuiskok reageert met 'ja'. 10. De stappen 6 en 7 worden herhaald voor het plantaardige ingrediënt. 11. Het systeem vraagt aan de thuiskok of er nog een ingrediënt moet worden toegevoegd. 12. De thuiskok reageert met 'nee'. 13. Het systeem vraagt om een stap. 14. De thuiskok voert de stap in. 15. Het systeem vraagt of een tip moet worden toegevoegd. 16. De thuiskok reageert met 'ja'. 17. Het systeem vraagt om de tip. 18. De thuiskok voert de tip in. 19. Het systeem vraagt aan de thuiskok of er nog een stap moet worden toegevoegd. 20. De thuiskok reageert met 'nee'. 21. Het systeem toont het recept + voedingswaarde. 22. Het systeem gaat terug naar het keuzemenu.
Uitzonderingen	<ol style="list-style-type: none"> 3. <foutieve invoer> Het systeem toont de melding: "Foutieve invoer. Kies toevoegen, tonen of exit" en voert vervolgens stap 2 uit. 5. <foutieve invoer> Het systeem toont de melding: "Foutieve invoer." en herhaalt de stap. 9. <thuiskok antwoordt nee> Het systeem gaat verder bij stap 11. 12. <thuiskok antwoordt ja> De stappen 6 t/m 11 worden herhaald. 16 <thuiskok antwoordt nee> Het systeem gaat verder bij stap 19. 20 <thuiskok antwoordt ja> De stappen 13 t/m 19 worden herhaald.

De klassen zijn niet of nauwelijks gewijzigd ten opzichte van week 3, dus het klassendiagram is daarom niet opgenomen hier. De realisatie van deze requirements zit vooral in het *main* gedeelte van het programma. Mocht het nodig zijn dan is het toegestaan om in de klassen extra methodes aan te maken.

Week 3 – opdracht

Jouw opdracht is om de applicatie te realiseren aan de hand van deze analyses en ontwerpen. Kijk goed naar wat er verwacht wordt op functioneel en technisch gebied.

Alle programmeerwerk wordt gedaan in het mapje “week 3”. Kopieer je bestanden uit “week 2”. Je gaat niet verder in “week 1” en “week 2”, dit is om het proces te laten zien (door de weken) voor de docent.

Week 4 – opdracht OOP Python – Receptenboek

Deze week is een cruciale week, want het is de laatste iteratie waarbij we aan onze applicatie kunnen werken. We willen absoluut niet onze anonieme opdrachtgever teleurstellen, daarom is het van belang dat alle must-have requirements uit weken 1, 2 en 3 écht af zijn. Mocht dat niet het geval zijn dan is dat de opdracht van deze week: het afmaken van die requirements!

Is het echter wel gelukt met het realiseren van die requirements dan zijn er nog een aantal would-have requirements, te weten:

- Het systeem moet in staat zijn om een PDF-pagina te genereren van het recept **[FR-10]**;
- De gegevens moeten persistent zijn (a.d.h.v. een database) **[FR-11]**.

Voor beide requirements is geen ontwerpdocumentatie beschikbaar, maar om je op weg te helpen zijn hier een paar tips/bronnen.

Voor FR-10 kun je gebruik maken van de Python module **reportlab**. Met deze module ben je in staat om vrij snel een PDF te genereren. Zie bronnen:

- <https://www.geeksforgeeks.org/creating-pdf-documents-with-python/>
- <https://pythonassets.com/posts/create-pdf-documents-in-python-with-reportlab/>

En maak gebruik van je creativiteit om die Pdf'jes mooi vorm te geven!

Voor FR-11 maak je gebruik van wat je tijdens de colleges hebt geleerd (klassen, objecten en databases). Pas je opgedane kennis toe en zorg voor een relationele database.

Week 4 – opdracht

Alle programmeerwerk wordt gedaan in het mapje “week 4”. Kopieer je bestanden uit “week 3”. Je gaat niet verder in “week 1”, “week 2” en “week 3”, dit is om het proces te laten zien (door de weken) voor de docent.