PV Pleines

Digitalisatie postduivenvereniging Pleines.



Hogeschool Novi

Oefenen met documentatie

Inhoudsopgave

[Inleiding 3](#_Toc53041436)

[Op te leveren 3](#_Toc53041437)

[Casusbeschrijving 4](#_Toc53041438)

[Introductie 4](#_Toc53041439)

[De duif 4](#_Toc53041440)

[Wedstrijd: Inkorven 5](#_Toc53041441)

[Wedstriijd: Losplaatsen 5](#_Toc53041442)

[Wedstrijd: De lossing 5](#_Toc53041443)

[Wedstrijd: Thuiskomen 5](#_Toc53041444)

[Wedstrijd: Uitslag 5](#_Toc53041445)

[Administratie 6](#_Toc53041446)

[Samenvatting 6](#_Toc53041447)

[En dan nog dit 6](#_Toc53041448)

[Succes 7](#_Toc53041449)

# Inleiding

Deze week gaan we oefenen met het opstellen van documentatie voor een applicatie. We gebruiken hiervoor de casus ‘PV Pleines’. Het is de bedoeling dat je zowel de functionele als niet-functionele eisen opstelt en het klassendiagram maakt.

De casus staat uitgeschreven in het volgende hoofdstuk. Deze casus kan ingeleverd worden op Teams en zal dan ook nagekeken worden door de docent. Voordat je aan deze casus begint, is het de bedoeling dat je de cursus *Backend documentatie* op Edhub in zijn geheel hebt doorgewerkt.

Voor deze casus hebben we expres een onderwerp gekozen waar niet iedereen per se veel vanaf weet. Het iets afweten van een bepaald kennisgebied heet ook wel domeinkennis. Je ziet in het beroepsveld ook vaak dat als mensen eenmaal ontwikkelaar zijn in de Banking-sector dat ze daarna ook bijna niet meer switchen.

Daarom is het oefenen van het eigen maken van de domeinkennis onderdeel van deze opdracht. In deze opdracht zal dan misschien ook niet alles duidelijk voor je zijn. We willen dan ook dat je naast de requirements en het klassendiagram een lijst met vragen opstelt over onduidelijkheden.

## Op te leveren

Uiteindelijk dien je het volgende (als pdf of word-document & **niet** gezipt) in te leveren in Teams:

Een document met:

* Functional requirements
* Non functional requirements
* Klassendiagram
* Vragenlijst met onduidelijkheden die je aan de opdrachtgever wilt stellen.
* (Optioneel): Lijst met rollen in de applicatie
* (Optioneel): Use cases (Gegeven bij front-end)

### Extra oefenen met Java

* Vertaal je UML-diagram naar Java.
  + Modelleer eerst volgens je UML-diagram
  + Bekijk of de code werkt
  + Pas eventueel code en diagram aan.

# Casusbeschrijving

Deze casus beschrijving is onderverdeeld in 4 onderwerpen (introductie, duif, wedstrijd en administratie) waar de product-owner vertelt over het huidige niet digitale bedrijfsproces.

## Introductie

Beste programmeur,

Ik ben Kees Merelvanger en voorzitter van het bestuur van postduivenvereniging Pleines (P.V. Pleines). P.V. Pleines is een vereniging waar duivenmelkers wedstrijden vliegen met hun duiven. Deze duiven korven we in en brengen we vervolgens naar locatie in Nederland, België, Frankrijk of onze meest verre vlucht: Barcelona.

De duiven worden gelost op de losplaats en de duif die het snelste terug is, dat wil zeggen met de hoogste gemiddelde snelheid, wint vlucht. Op sommige vluchten, zoals Bergerac, wint de liefhebber met de snelste duif een wisseltrofee, die hij dan een jaar lang in bezit mag houden.

We hebben vijf soorten vluchten. Vitesse-fond, midfond, dagfond en overnachting en jonge duiven.

* *Vitesse* zijn de kortste vluchten. Vaak tot 300 kilometer.
* Midfond zijn korte vluchten, vaak gelost in Nederland of noord-België.
* Eendaagse fond zijn lange vluchten die duiven in 1 dag kunnen afleggen, meestal rond de 600 kilometer.
* Marathonvluchten zijn vluchten waar duiven onderweg overnachten, voordat ze thuis op het hok landen. Vanaf 700 kilometer
* Jonge duiven-vluchten zijn vluchten waarin duiven alleen mogen meedoen die in dat jaar geboren zijn.

Ik hoop u zo een korte introductie hebben kunnen geven over wat de duivensport in het kort inhoudt.

## De duif

De meeste liefhebbers hebben meerdere duiven. Een mannetjesduif heet een *Doffer* en een vrouwtjesduif een *duivin*. Elke duif mag meedoen aan alle wedstrijden, behalve de jonge duiven vluchten. Zou het jaar 2021 zijn, dan mogen alleen duiven geboren in 2021 meedoen aan de jonge duiven-vluchten.

Een duif heeft twee ringen. De eerste ring is de eigendomsring. De duivenmelker heeft een eigendomsbewijs van deze ring om aan te tonen dat de duif van hem is. Deze ring is niet afneembaar en hoort altijd bij 1 duif.

De tweede ring is de *chip-ring*. De chip-ring wordt gebruikt om de duif *in te korven* voor de vlucht. Zo wordt geregistreerd dat de duif meedoet aan de wedstrijd. Deze ring is afneembaar en deze ring hoort dus niet altijd bij dezelfde duif. Deze ring heeft ook een nummer.

De eigendomsring bevat een nummer die er altijd als volgt uit ziet: LANDAFKORTING – JAARTAL – NUMMER. Dus zou het gaan om een Nederlandse duif, geboren in 2021, dan is het nummer: NL 21 – 1234567. Er worden in Nederland jaarlijks meer dan een miljoen duiven ingeschreven. Deze moeten dus allemaal een uniek nummer kunnen krijgen.

## Wedstrijd: Inkorven

Om mee te kunnen doen aan een vlucht moet een duif ingekorfd worden. Tijdens dit proces wordt op het wedstrijdformulier het nummer van de eigendomsring en de nummer van de chip-ring opgeschreven. De duif wordt vervolgens in de mand gedaan en wordt pas gelost wanneer deze op de plaats van bestemming is.

## Wedstrijd: Losplaatsen

We hebben verschillende losplaatsen voor duiven. Bergerac en Barcelona zijn al genoemd, maar de duiven gaan ook naar minder exotische oorden, zoals Duffel (België). Van deze losplaatsen moeten wij de plaatsnaam en de lengte en de breedtegraad kunnen vastleggen. Op deze manier kunnen we precies uitrekenen hoe ver de duif moet vliegen.

## Wedstrijd: De lossing

Op het moment dat de vrachtwagen met alle duiven is aangekomen bij de losplaats, worden de duiven op een bepaald moment gelost. Dit gebeurt alleen als het weer niet te zwaar is voor de duiven. De chaffeur van de vrachtwachten krijgt een seintje wanneer de duiven gelost worden. Dit tijdstip is belangrijk voor ons, want zo weten we hoelang de duif over de vlucht heeft gedaan.

## Wedstrijd: Thuiskomen

Nadat de duiven gelost zijn, vliegen de duiven naar huis. Duiven doen dat op basis van een kompas in hun neus en visuele kenmerken die zij zien en herkennen. Wanneer de duif op het hok landt, wordt het tijdstip tot de miliseconde nauwkeurig en het serienummer van de ring geregistreerd. Dit wordt allemaal opgeslagen in een computertje dat op de vereniging uitgelezen kan worden.

De liefhebber dient zijn eerste duif binnen 1 uur telefonisch gemeld te hebben bij de vereniging. Is de liefhebber te laat, dan wordt de duif gediskwalificeerd.

## Wedstrijd: Uitslag

Een aantal uur nadat de eerste duif geland is (dit is verschillend per vlucht en wordt op de dag zelf bepaald), wordt er afgeslagen op de vereniging. De duivenmelkers die meededen aan de wedstrijd nemen hun computers mee met tijdregistratie en deze worden uitgelezen op de vereniging.

Tijdens dit uitlezen wordt de duif met behulp van zijn chipring-nummer gekoppeld aan het eigendomsnummer. Het eigendoms-nummer hoort altijd bij één liefhebber. Deze liefhebber heeft een verblijfplaats.

Op basis van de coördinaten van de liefhebber, de losplaats, de lostijd en de aankomsttijd van de duif kunnen we de gemiddelde snelheid van de duif per seconde uitrekenen. De duif met de hoogste gemiddelde snelheid wint.

Alle teruggekomen duiven worden op het wedstrijdformulier gezet op volgorde van snelheid. Dit wordt gedeeld met de liefhebbers. De liefhebber van de duif krijgt het volgende aantal punten:  
Aantal-kilometers + Aantal-Duiven-die-meededen + 1 – positie.

Dus de liefhebber van de duif die vijfde van de 100 is geworden in een vlucht van 800km krijgt: 800 + 100 + 1 – 5 = 896 punten. Een liefhebber kan met meerdere duiven spelen en zo dus meerdere punten halen.

Per seizoen (een seizoen loopt van januari tot en met december) zijn er twee algemene klassementen: Liefhebberprijs en duifprijs. Deze is er ook voor elke soort vlucht. Dus zowel de duif als de liefhebber krijgt punten.

De liefhebber en/of duif met het hoogste aantal punten over alle vluchten wint de all-round beker. De liefhebber en/of duif met het hoogste aantal punten per vluchtsoort wint de beker van de desbetreffende vluchtsoort.

## Administratie

Ik hoop dat alles rondom de vluchten en het kampioenschap nu duidelijk is. Naast al deze wedstrijdorganisatie dingen die we in een web-applicatie willen hebben die zowel van de laptop als tablet toegankelijk is, willen we ook ledenadministratie bijhouden.

Van elke liefhebber willen we weten waar hij woont, welke duiven hij heeft, zijn lengte en breedtegraden en of de contributie van het jaar al betaald is. Daarnaast is een e-mailadres en een telefoonnummer ook een vereiste.

Liefhebbers kunnen verschillende rollen binnen onze club hebben en soms ook meerdere taken. Zo hebben we voorzitter, duivenmelker, stil lid, inkorver, bar-medewerker en wedstrijd-berekenaar.

## Samenvatting

Omdat het een hoop tekst is, willen we hier nog een verduidelijken en samenvatten wat we willen. Het tot nu beschreven proces wordt allemaal met de hand gedaan. Dat betekent dus ledenadministratie, maar ook het bereken van de uitslag van de vluchten. Dit willen we zoveel mogelijk automatisch gaan doen.

Dit bevat, maar limiteert zich niet tot: Het beheren van losplaatsen, het aanmaken van seizoenen, toevoegen van wedstrijden aan seizoenen. Duiven toevoegen aan wedstrijden. Uitslagen verwerken voor wedstrijden. Uitslagen tonen van wedstrijden, deelkampioenschappen en kampioenschappen kunnen publiceren. Het bijhouden van de ledenadministratie. Het verplaatsen van duiven tussen leden en het toevoegen en verwijderen van duiven bij leden.

(Voorlopige) puntentelling van de verschillende kampioenschappen kunnen tonen en puntentelling van vorige kampioenschappen kunnen zien.

Ook willen we het adres van de vereniging, het emaildres, het telefoonnummer kunnen bijhouden in het elektronisch systeem.

## En dan nog dit

Nadat jij het systeem gebouwd hebt, gaat dit onderhouden worden door het neefje van de voorzitter. Pim kan alleen programmeren in Java, html, css en React. Het systeem moet ook in deze taal geschreven worden. Van onze overkoepelend orgaan (NPO), hebben wij een database tot onze beschikking gekregen. Dit is een postgresql database. Die willen wij ook gaan gebruiken voor de applicatie.

We hebben veel oude leden, dus de applicatie moet goed leesbaar zijn, makkelijk te vergroten zijn en rekening houden met kleurenblindheid.

# Succes

Maar, nog wat tips:

* Bedenk welke data je nodig hebt om dingen (automatisch) te verwerken. EdHub (Backend Documentatie: H2.3) heeft hier wel goede tips voor.
* Dingen die niet digitaal gebeuren, hoef je niet in een klassendiagram te beschrijven
* Begin klein. Misschien is Lid, Losplaats of Duif wel een mooie & makkelijke klasse om mee te beginnen.
* Je hoeft geen getter en setter methodes in je klassendiagram toe te voegen. Andere methodes mogen wel
* Constructors hoeven niet per se aan het klassendiagram toegevoegd te worden.
* Gebruik Appendix A.1: Klassendiagram uit de cursus.
* Vertaal je klassendiagram naar code en kijk of je code werkt. Pas eventueel code en diagrammen aan.
  + Doe dit in kleine stapjes. Maak een paar klassen op basis van een use-case of relaties in UML en programmeer deze vervolgens. Test of de werking gewenst is.
* Schrijf wat de applicatie moet kunnen in eigen woorden. In het echte leven zou je na deze stap contact opnemen met de product-owner om te kijken of je het bij het goede eind hebt.
* Je mag extra informatie op het Internet zoeken. De informatie in dit document is echter leidend. Het is ook beter om te vragen dan aannames te maken!