**Notulen Sprint 1 - 26-11-2021**

**Git Repository**

1. Maandag een Spring project aanmaken (dmv Spring Initializer), dat kun je lokaal in Git zetten.
2. Vervolgens kun je die lokale git aan deze Remote repository koppelen (=Vault), dit gebeurt door 1 persoon. (dmv Git Init)
3. Onderaan in onze Repository vind je “push existing repository” > Stap twee hoef je niet uit te voeren (dus de rename)
4. De rest van het team voert een pull uit
5. We laten security nog zitten totdat we weten wat we moeten doen na de workshop.

**Scrum in Gitlab**

* Milestones
  + Product (de grootste milestone)
  + Voor iedere sprint een aparte milestone

**Planning**

*In totaal zijn er 35 punten, 1 punt per dagdeel per persoon*

Er moet voor elke User Story een Use Case worden uitgewerkt. De use cases worden aangepast aan wat we deze sprint gaan doen, oftewel **alleen back-end.**

De user strories opdelen

* Wat is de user story
* Wat is de Definition of Done
* Wat zijn de acceptatiecriteria

**We doen 2 dingen**

(1) we maken een use case voor de hele epic met de Definition of Done

(2) een tweede “User Interface Document” met hoe de user-interface eruit gaat zien

> daarin staat uitgelegd hoe alle stappen worden uitgevoerd voor de gebruiker, bijvoorbeeld welke methodes moeten worden aangemaakt, wat voor API-calls er gemaakt moeten worden.

- Wat is de URL  
 - wat voor parameters moeten mee  
 - wat zit er in de body  
 - GET/PUT/POST

> Dit document wordt **dinsdag** ingeleverd

**User Interface document met alle klassen**

* Hierin wordt voor elke klasse bijgehouden
  + Wie hem gemaakt heeft
  + Wie hem ge-reviewed heeft
  + Zijn er unit tests aanwezig?
    - Hiervoor is een use case noodzakelijk
  + Voldoet dit aan de code conventions?
  + Remote: Draait het op een externe machine? (Dinsdag a.s. wordt uitgelegd hoe we dit doen: miw-team-2.nl, de PO zal dit via Postman testen).
  + Er moet voor elke klasse documentatie zijn in de vorm van Javadoc.
    - Alleen voor de *public klassen*
    - Wat doet deze klasse, wat zijn de parameters etc.

**User Story 1 – benodigdheden**

* Het systeem moet aangeven welke gegevens er nodig zijn, een request geeft aan wat er nodig is
* Het volgende is nodig:
  + Naam (zoals dat in je paspoort staat) 1 String
  + Geboortedatum
  + BSN
  + Adres
  + Username + Password
    - Twee belangrijke opties voor de username:
      * 1 genereer, 2 gebruiker bepaalt
      * Moet uniek zijn
    - Password:
      * Moet aan bepaalde criteria voldoen
      * Moet encrypted worden opgeslagen in de database, dus niet in plain tekst.
      * Key: zit opgeslagen in de machine, de salt en de pepper ook (hier krijgen we later uitleg over)
* Response moet teruggegeven worden in Json formaat, oftewel een omschrijving van de zojuist aangemaakte gebruiker
* Er moet een juiste HTTP-response worden teruggegeven met een overzicht

**Q-A PO**

Vraag: wil de product owner al een database hebben?

Antwoord: nee, maar wel DAO’s

Vraag: is de portefeuille een aparte entiteit? (Als in: los van de rekening)

Antwoord: dat is een ontwerpkeuze van ons team

Vraag: wat wil de PO als de gebruiker de inloggegevens kwijt is?

Antwoord: op dit moment nog niks

Vraag: hoe gebruiken we de IBAN’s, als je een rekening en portefeuille object hebt

Antwoord: gebruik losse primary keys