1. **Verzin een onderwerp voor je applicatie**

In mijn vrije tijd doe ik graag fitnessen, daarom wil ik hier een web app voor gaan maken.

1. **Wat is een probleem dat met software kan worden opgelost?**

Mijn idee is een soort calorie tracker (macro’s). Aangezien je bij het sporten ook goed op je eten moet letten denk bijv. aan kcal, eiwitten etc. wil ik een webapplicatie gaan maken waarin ik dit per dag kan vastleggen en terugzien.

Wat lost dit nu op?

Veel van de al bestaande calorie trackers hebben veel producten maar niet precies de producten en merken die ik eet, daarom wil ik een applicatie maken waaraan ik zelf producten kan toevoegen met een admin account. Wanneer eten wordt toegevoegd vanuit het admin account zal dit worden teruggerekend naar 1 gram en opgeslagen in de database. Dit voor wanneer ik maar bijv. een halve pot kwark eet i.p.v. een hele pot.

Ook zijn vaak de al bestaande calorie trackers betaald of er zitten niet alle functionaliteiten bij die ik graag zou willen. Voor deze reden zou ik er zelf graag een willen maken, zo kan ik ook met mijn vrienden overleggen of zij nog behoeftes binnen deze web app hebben.

1. **Hoe complex moet dat zijn?**

Voor dit project wil ik minimaal het volgende kunnen: (MVP)

* Inloggen
* Registreren
* Een of meerdere admin account(s) wat producten met macro’s kan toevoegen
* Producten invullen wat je gegeten hebt per gebruikers account
* Hoeveelheid gegeten invullen per product
* Terugzien wat je vorige dagen gegeten hebt voor je eigen account
* Voedings doelen instellen

Optioneel:

* Training schema
* Gewicht per training in kg bijhouden in schema
* One rep max bijhouden in schema
* Toekomst doelen instellen voor gewicht per oefening
* Creatine check-mark

1. **Hoe pak ik dat aan?**
   1. **Implementatie: ASP.NET MVC core**

Aangezien bij onze leeruitkomsten staat dat we minimaal een web-based applicatie moeten maken ga ik ASP.Net met het MVC-patroon gebruiken. MVC is een veel gebruikt patroon. Dit patroon zorgt voor een leesbare en gesorteerde code. Ik heb voorheen wel eens met Laravel een website gemaakt. En nu ben ik vooral benieuwd hoe anders deze twee zijn.

* 1. **Database: SQL-server**

Voor database ga ik een MSSQL-server gebruiken. Deze database wordt vanuit school gratis aangeboden en ASP.Net en SQL is beide van Microsoft waardoor de kans kleiner wordt dat iets niet goed met elkaar werkt.

* 1. **Version control: Git (Gitlab)**

Git is de meest gebruikte versie beheer, dit zorg ervoor dat mijn projecten makkelijk te delen zijn met bijvoorbeeld medestudenten wanneer nodig. Voor vorige projecten heb ik nog altijd github gebruikt aangezien ik hier nog een studenten account van heb door mijn vorige school. Ik was dus ook aan de UI van github.com gewend. Aangezien school aanbiedt om gitlab te gebruiken en dus ook alle docenten hier een account op hebben ga ik komende projecten op gitlab zetten.

* 1. **Ontwerp: UML-klassendiagram**

Aangezien ik OOP ga programmeren in het project heb ik classes nodig. Hiervoor ga ik de UML-klassendiagram toepassen.

* 1. **Analyse: Requirements en use-cases**

Voor mijn project duidelijk voor mezelf te hebben wil ik requirements opstellen. Ik heb dan voor mezelf een duidelijke prioriteiten lijst opgesteld die ik kan aanhouden.

Use-cases helpen bij het verduidelijken van bijvoorbeeld een admin of gebruikers account, zo kan ik makkelijk aangeven wie wat kan.

* 1. **Tests: Unit tests**

Voor het testen van de applicatie ga ik unit tests gebruiken. Unit tests zijn tests per functie of stuk code (unit), deze test zorgen ervoor dat je snel en makkelijk een bug kan vinden. Het risico dat een stuk code niet meer werkt door het aan te passen wordt hierdoor ook een stuk kleiner en hoef je niet zolang te debuggen.