

2025-W44

Week 44 – Design

Doelen voor deze week

- Design (-)
- ER-diagram (x)
- Flowchart (-)
- State-diagram (x)
- database (x)
- Feedback linear leren template (x)

Focus

Design

Verwachte uitdagingen

Genoeg tijd vinden om alles te doen wat ik wil

Ondersteuning of afstemming

Feedback voor de template die ik heb gemaakt om bij te houden wat ik per week leer.

Verwachte volgende stap

Project te gaan maken

Wat heb ik gedaan

- Feedback verwerkt van Kees in Weekkaart Template
 - Emoties krijgen nu een grotere focus door bij het aanmaken van items verplicht Energie (1–5), Motivatie (1–5) en Emotie (keuzelijst) te registreren.
- ER-diagram gemaakt voor mijn project en feedback gevraagd aan Kees
 - Eerst geprobeerd met verschillende applicaties (oa. Gleek, Visual Paradigm), uiteindelijk besloten handmatig te tekenen met behulp van Draw.io.
 - Chen-notatie toegepast, Crow's Foot door elkaar gehaald, feedback verwerkt.
 - Nieuw inzicht: Chen-notatie is academisch, ik twijfel aan de geschiktheid voor grote ontwerpen. Voor het maken van Chen-notatie werkt Draw.io goed.
- C# Loops en Arrays geoefend

- Resultaten en leerpunten vastgelegd in [Workshop Programmeren extra - c_sharp oefenen](#).
- State-diagram voor software gemaakt
 - Origineel met Draw.io, later geprobeerd in Mermaid voor versiebeheer en overzicht, maar nog niet helemaal tevreden. [Powerlifting Arduino State Diagram](#)
 - Feedback gevraagd en verbeterpunten genoteerd.
- ER-diagram met Mermaid gemaakt [Powerlifting Database diagram](#)
 - Leerpunten vastgelegd in [Powerlifting Database diagram Learned](#).
- Flowchart voor PC-logica van mijn software gemaakt [Powerlifting PC Flow chart](#)
 - Geconstateerd dat het beter is eerst een state-diagram en/of klasse-diagram te maken om spaghetti-code te voorkomen.
- Klasse-diagram overwogen als volgende stap
 - Verwacht dat dit diagram helpt om state-diagram en flowchart beter te structureren.
- Original Design-bestand herzien
 - Tijdens het maken van diagrammen veel verkend en ontdekt dat de volgorde belangrijk is.
 - Nieuwe structuur voor het document voorbereid om te maken in LaTeX of Typst, eventueel vanuit Obsidian. Dit is nog niet af, volgende week mee verder gaan.

Wat heb ik geleerd

- ER-diagram maken is een puzzel: alles logisch met elkaar verbinden vraagt om overzicht en planning.
- Chen-notatie is academisch, praktische tools zoals Draw.io zijn effectiever voor het ontwerpen van conceptuele databases.
- Feedback vragen is cruciaal: helpt fouten zoals Chen vs. Crow's Foot notatie te corrigeren.
- Het oefenen van C# Loops en Arrays versterkt programmeervaardigheden en inzicht in logica voor flow- en state-diagrammen.
- Het maken van state-diagrammen en flowcharts is waarschijnlijk makkelijker als je eerst een goed klasse- of ER-diagram hebt.
- Documentstructuur en volgorde van diagrammen bepalen het overzicht en de effectiviteit van je designdocument.
- Het bijhouden van energie, motivatie en emotie helpt waarschijnlijk om reflectie concreter en inzichtelijker te maken. In elk geval om uiteindelijk te kunnen constateren wat mij aanzet.

Reflectie (einde van de week)

Wat werkte goed?

- Het vragen en verwerken van feedback helpt sneller fouten in mijn gedachtegang te ontdekken. (chen-notatie vs crows feet notatie)
- Hands-on experimenteren met verschillende tools hielp bij het vinden van de meest geschikte aanpak.
- Het bijhouden van een logboek ondersteunt het invullen van deze weekkaart.
- C# oefeningen hebben mijn basis in programmeren versterkt.

Wat kan beter?

- Mijn logboek volledig digitaal bijhouden in Obsidian in plaats van op papier. Digitaal kan je makkelijker knippen, plakken en samenvatten met tools zoals ChatGPT.
- Beter vooraf bepalen welke diagrammen in welke volgorde nodig zijn. Het is een soort flow, waar de ene eindigt is het de begin/basis voor de volgende diagram.
- Een efficiënte manier vinden om verschillende documenten samen te voegen tot één overzichtelijk document.

Wat was impactvol of opvallend?

- Het werken met grafische versus syntax-gestuurde methodes: syntax lijkt beter geschikt voor versiebeheer en grotere, complexere projecten. Het maakt aanpassingen achteraf eenvoudiger. Althans dat denk ik nu.
- Door simpele C# programmeeroefeningen te doen, wordt mijn basis steeds sterker. Vooral voor loops en arrays wil ik mijn begrip verder verbeteren.

Hoe voelde de week?

- Van gefrustreerd naar gemotiveerd: het begin van de week was frustrerend vanwege tools zoals Visual Paradigm, waar alles met de hand moest worden getekend. Het ontdekken van syntax-gestuurde diagrammen opende een nieuwe wereld en motiveerde mij verder.

Belangrijkste inzicht

- Diagrammen en opmaak kunnen effectief worden gemaakt op basis van syntax, wat voordelen biedt voor overzicht en versiebeheer.

Inzichten of ideeën

- Documenten vervangen door Markdown, Typst of LaTeX voor betere controle en overzicht.
 - Docenten vragen om advies: wat is het handigst en zien zij problemen die ik nog niet zie?
- Voor ER-diagrammen eerst scenario's of verhaaltjes maken voordat de diagrammen worden opgezet; navragen of deze aanpak helpt (use case studies).

- Klasse-diagram concreet uitwerken om flowchart en state-diagram overzichtelijker te maken.
- State- en flow-diagrammen beter baseren op een goed uitgewerkt klasse- of ER-diagram om overzicht te behouden.

Dataview overzicht

File (2)	week	thema	focus	doelen	volgende_stap
2025-W44	44	Design	Design	<ul style="list-style-type: none"> • Design • ER-diagram • Flowchart • State-diagram • database • Feedback linair leren template 	Project te gaan maken
2025-W43	43	Leerdoelen	Leerdoelen - het aantonen van linair leren	<ul style="list-style-type: none"> • Leerdoelen formuleren • Portfolio bijwerken • Workshops volgen 	Terug te gaan naar mijn project om verder te gaan met design

 [Bekijk Canvas-overzicht](#)

 [Weektekening](#)