Fase 3 - Logboek

# Sprintlog/ Ontwikkellogboek

## Week 11: Ontwikkeling van frontend prototype

### Wat is er gedaan:

Op basis van de opgemaakte wireframes zijn drie hoofdschermen uitgewerkt in React/TypeScript:  
• Het workflow canvas inclusief node-selectiebar  
• De workflows-pagina (voor gebruikers)  
• De admin-pagina (voor data-analisten)

### Waarom:

Deze onderdelen zijn essentieel binnen het MVP (FR-DA-07, FR-DA-06). Doel is om gebruikers self-service functionaliteit te bieden en analisten de mogelijkheid te geven workflows centraal te beheren.

### Hoe:

De frontend is opgezet met React en TypeScript. Voor het canvas is gewerkt met XYFlow als basis. State management gebeurt via React Context. De schermstructuur is opgezet op basis van de wireframes die eerder in samenwerking met stakeholders zijn opgesteld.

### Wat is het resultaat:

Een klikbaar prototype is beschikbaar, inclusief visueel canvas en workflow beheerfunctionaliteit. Er is nog geen backend-koppeling. Screenshots zijn toegevoegd aan bijlage X. Het prototype is gedeeld met de opdrachtgever.

### Vervolg:

In week 12 wordt het prototype besproken tijdens een review met de opdrachtgever. Feedback wordt verwerkt. Parallel wordt gestart met de project setup en configuratie voor de start van de ontwikkelfase (Sprint 1).

## Week 12/13: Project Setup en Configuratie (Sprint 1)

### Wat is er gedaan:

Door de product owner is een Epic aangemaakt voor het project. Alle rapporten zijn toegevoegd. Ook alle tekeningen en schema’s. Binnen dit Epic zijn stories aangemaakt en zullen verder worden uitgewerkt per sprint. In de komende 5 sprints (5x 2 weken) zal er een MVP worden ontwikkeld dat de basisfunctionaliteit bevat die de belangrijkste werklastvermindering zal bewerkstelligen. Het maandelijkse handmatige proces dat data-analisten nu uitvoeren voor het genereren van rapportages zal worden geautomatiseerd. Dit is gecommuniceerd met de opdrachtgevers. Alle requirements die hier geen bijdrage aan leveren zullen pas hierna verder worden opgepakt.

Er is een projectnaam bedacht: **Toskr** (/’Tɔs.kar), afgeleid van de Noorse mythologische eekhoorn Ratatoskr.

Er is een project aangevraagd voor een Kubernetes-namespace. Er zijn rechten aangevraagd voor gebruikers en admins. Er is een project aangemaakt in GitLab. Er is een database aangevraagd voor het opslaan van workflows, statussen en logging. Ook worden de benodigde schema’s aangemaakt voor task queuing via JobRunr.

Er is een werkend Java-project opgezet met Spring Boot als framework en Swagger API-documentatie. Daarnaast zijn de eerste API-endpoints opgezet.

Voor het testen is gekozen voor:

* **Backend**: Java tests (unit, …)
* **Frontend**: React TypeScript tests met Playwright (end-to-end tests)

### Waarom:

Om op een gestandaardiseerde en herhaalbare manier te kunnen ontwikkelen in teamverband, is het essentieel dat het project technisch goed is ingericht. De keuzes voor GitLab, Kubernetes, Swagger, Spring Boot en Playwright zijn gemaakt op basis van eerder afgestemde architectuurprincipes (zoals “Cloud First”, “Open Standards”, “Testbaarheid”) en maken CI/CD en schaalbare uitrol mogelijk.

### Hoe:

### Wat is het resultaat:

### Sprint review:

### Feedback verwerking:

### Vervolg:

## Template sectie

### Waarom dit logboek voldoet:

* **Volgt de ontwikkeling stap voor stap** (je toont continuïteit en voortgang).
* **Koppelt werkzaamheden aan requirements (FR/NFR)** – **superbelangrijk** voor traceerbaarheid.
* **Toont methodisch werken**: per stap beschrijf je *Wat/ Waarom/ Hoe/ Resultaat/ Vervolg*.
* **Bewijslast wordt opgebouwd**: je verwijst naar artefacten (code, wireframes, screenshots, bijlagen).
* **Maakt het reflecteerbaar en bespreekbaar** tijdens begeleidings- of beoordelingsmomenten.

### Bonus (en goed om mee te nemen):

* Voeg af en toe screenshots toe als bijlage.
* Koppel je commits (indien Git gebruikt) aan de logboekweken.
* Voeg codefragmenten toe als “bewijs” bij technische keuzes of complexe onderdelen.
* Beschrijf ook eventuele **problemen en hoe je die hebt opgelost** — dat toont probleemoplossend vermogen.
* Laat aan het einde van elke sprint even kort weten hoe je de feedback hebt verwerkt.

Tip: voeg na afloop van elke sprint een “Sprint Review” sectie toe waarin je reflecteert op het proces en het resultaat.