

Verantwoording

Jurjen Vorhauer, 800009793

2024-02-23

- Technische Keuzes
- Limitaties
- Persoonlijke Opmerkingen

Technische Keuzes

Meeste technische keuzes zijn gemaakt door Novi. Alleen JavaScript zou ik zelf niet meer kiezen.

De backend is een voortgaande ontwikkeling op basis van de eindopdracht voor de Backend (Java met Spring Boot) module van Novi. Deze host ik zelf, heb ik zelf ontwikkeld en beheer ik ook zelf. Maximale DevOps met minimaal team...

JavaScript

Door Novi opgelegd. Zou ik zelf niet meer voor kiezen: TypeScript of een andere hoger niveau taal en omgeving. Blijft een vreemde programmeertaal ten opzichte van de bij mij meer bekende talen als Java, Kotlin, Python en Scala. Een programmeertaal zonder een type-system dat me helpt om tijdens het bouwen en uitvoeren van een applicatie veelgemaakte fouten te voorkomen is echt wennen en zorgt voor veel langere ontwikkel- en testtijd.

React

Is best een fijn en veelzijdig framework. Vlot nieuwe applicaties bouwen en bestaande applicaties uitbreiden zijn na enig oefenen goed te doen.

Axios

Maakt communiceren met mijn eigen en andere backends/APIs overzichtelijk en beheersbaar.

html

Niet aan te ontkomen en dat zou ik ook niet willen. Ik vind, na 25 jaar regelmatig ermee werken, dat html niet voor niets nergens door vervangen is. Soms wat veel

tekst nodig om iets te definiëren, met name tabellen, maar over het algemeen begrijpelijk en structureel goed doordacht.

css

Voor layout, kleuren en typografie is css onmisbaar. Ondanks veel bezwaren vanuit ontwikkelaar-vriendelijkheid is css onlosmakelijk verbonden met het ontwikkelen van mooie en aansprekende websites en webapplicaties.

Limitaties

Algemeen

Werkt alleen met Internet verbinding, aangezien backend daar ergens draait; geen offline mogelijkheid. Maar dat geldt ook voor het clonen van het project vanaf GitHub en voor het downloaden van een release van NodeJS + npm; en ook voor het downloaden van bijvoorbeeld het React framework..

Functioneel

1. Een gebruiker heeft maar erg weinig mogelijkheden om zijn of haar informatie te bewaren: naam, email adres en password. Een profiel met (veel) meer data zou aantrekkelijk kunnen zijn;
2. Een taak is opzettelijk inhoudelijk beperkt opgezet. Dit is in lijn met het BaseCamp gedachtengoed: zorg dat de tool geen doel op zich wordt, waarmee ook gezorgd wordt dat de functioneliteit niet ondergesneud wordt door vele velden;
3. Een notitie is om dezelfde reden bescheiden gehouden. Het risico dat veel verschillende invoer/data om steeds meer functionaliteit vraagt is te groot (zie Jira);
4. Gebruikers leven nu in het backend systeem. Gegeven de grote hoeveelheden Google, GitHub, Microsoft, etc. bestaande accounts zou OAuth als authenticatie mechanisme wellicht makkelijker en zelfs drempelverlagend kunnen zijn.
5. Op dit moment hebben taken wel een deadline, maar daar wordt verder niets actief mee gedaan. Het zou bijvoorbeeld handig zijn als er dan een email verstuurd wordt.

Mogelijke toekomstige verbeteringen

- relatie tussen taken en notities
- omzetten van een taak naar een notitie en vice versa
- tags om taken en notities te kunnen groeperen of in ieder geval visueel beter te kunnen identificeren
- sociale functies: zoeken naar andere gebruikers en volgen van die gebruikers, waardoor publieke notities te zien zijn
- taak kunnen voorzien van prioriteit

- notitie kunnen voorzien van een vlag om aan te geven dat die notitie publiek (zie ook “sociale functies” hierboven) of privé is.
- deadline-gerelateerde berichten versturen (zie punt 5. van de functionele beperkingen hierboven)

Technisch

1. De enige formattering in de tekst-velden van notitie en taak zijn regelovergangen (linebreaks), maar om die goed weer te geven moet je in React nogal wat zoeken, maar er blijkt een nette css oplossing voor te zijn;
2. Heel soms werkt het automatisch verversen van de inhoud van een pagina niet de eerste keer: als bijvoorbeeld een notitie gewijzigd werd, dan wordt bijna altijd de inhoud van die notitie direct correct weergegeven. Maar heel soms niet.
3. Graag had ik `useForm` gebruikt, maar dat wilde niet werken met de door mij gekozen opzet met een pagina waarop een herbruikbare `dialog` het invul-formulier toont: alle velden werden dan read-only en dat is wel veilig, maar niet gewenst.

enige persoonlijke opmerkingen achteraf

Het deel wireframes-op-papier en Figma-prototype heb ik expres overgeslagen: kost veel tijd en levert 5% van het eind-cijfer op. Ik ben geen grafische vormgever of designer. Dat is een vak apart. Daar komt bij dat het Agile werken voor backend services (mijn dagelijks werkend bestaan) een stricte ontwerp naar ontwikkel methodiek erg onaantrekkelijk maakt. Zo nu en dan mag ik een frontend maken, meestal van interne, administratieve aard, en dan ga ik regelmatig naar de gebruikers met de werkende frontend en laat die zien. Daardoor krijg ik feedback over het daadwerkelijke product en niet op een niet- of nauwelijks werkend geheel. Vaak gaan de vragen voor dat soort interne tools ook helemaal niet over de layout maar veel meer over de werking van de app. Functie over vorm, zeg maar.

Ik heb grote moeite gehad om de 50 gevraagde functionele en niet-functionele eisen vol te krijgen. Wellicht heb ik dat deel niet correct geïnterpreteerd?

En, last-but-not-least, het is onzettend onhandig om je enige gebruiker én de developer van de applicatie te zijn... feedback is daardoor heel eenzijdig en erg subjectief.