# Demo

Laat bij elk onderdeel de happy path en de alternative path zien. Laat daarna bij elk onderdeel de bijbehorende user story/acceptatiecriteria zien(kan ook eerst criteria laten zien, dan testen). Dit demo wordt uitgevoerd vanuit het perspectief van de gebruiker.

Alles wat in rood staat weet ik niet zeker of dit van toepassing is in deze app of dat ik iets anders aan moet pakken

1. Homepagina
   1. Even de homepagina laten zien. Vertellen dat er nog geen content erin zit omdat prioriteit van functionaliteit hoger ligt.
   2. Navigeer naar registratie
2. Registratie
   1. Alternative path: Foute email, niet gelijke wachtwoorden
   2. Happy path: Alle geldige gegevens om te registreren
   3. Volgende pagina -> Login.
   4. Laat acceptatiecriteria zien (US 3)
3. Login
   1. Alternative path: Login met ongeldige gegevens
   2. Happy path: Login met zojuist gemaakte account
   3. Laat acceptatiecriteria zien (US 1)
4. Homepagina ingelogde gebruiker
   1. Laat menu zien, Pagina leeg door gebrek aan opgeslagen artikelen
   2. Navigeer naar opslaan van artikel
5. Artikel opslaan
   1. Alternative path
      1. Vul een ongeldige URL in.
      2. Check tags op ongeldige tekens (Als dit bestaat)
   2. Happy path
      1. Pak een willekeurige artikel op NOS of nu.nl.
      2. Pak de URL van dit artikel
      3. Voeg deze URL toe aan de input van de pagina
      4. Voeg een aantal tags toe.
      5. Sla artikel op
   3. Laat acceptatiecriteria zien (US 5, US 9)
6. Homepagina ingelogde gebruiker
   1. Laat nieuw opgeslagen artikel zien
   2. Klik op read article
   3. Laat acceptatiecriteria zien (US 6)
7. Pagina lezen van artikel
   1. Scroll langzaam door de artikel heen
   2. Laat tags zien op de pagina
   3. Laat originele pagina zien van de artikel
   4. Laat acceptatiecriteria zien (US 12)
8. Log uit gebruiker
   1. Klik op menu
   2. Klik op log uit knop
   3. Laat acceptatiecriteria zien (US 2)
9. Extensie.

# C4 Model

6. Back-end

Ophalen van alle artikelen bij openen van dashboard

* API doet fetch request naar back-end
* Request komt aan in app.js
* App.js stuurt request door naar user.js
* De request zoekt naar de correcte route binnen user js
* Vanuit user js gaat de request naar passport.js
* In passport js kijkt de file of er een cookie meegegeven is en of deze cookie niet expired is
* Als de cookie correct is, haalt passport.js de user\_id uit de cookie
* Vervolgens verbindt Passport.js met de user database en haalt de user op
* Passport.js koppelt de user aan de request
* Passport.js stuurt een request terug naar de route in user.js
* User.js verbindt met de database om de user op te halen
* Pakt alle gegevens van de user en gaat terug naar user.js
* User.js haalt alle article id’s, die in de user object zijn opgeslagen, op in de article database deze informatie gaat terug naar user.js.
* Alle informatie van de opgehaalde artikelen stuurt hij naar de laterlezer app.