# Uitbreiding van de IT-vitae chat app

Anki kaarten – Helpend bij het beoefenen van leerstof.

## Wat moet er allemaal gebeuren?

Client-side Robert (en Maurice als server-side eerder klaar is)

* **Pagina’s (Views) die moeten worden aangemaakt.**
  + Alle decks pagina
  + Decks inhoud pagina
  + Kaart pagina
* **Alle decks pagina (AllDecksPage.xaml)**
  + Geeft alle (lokale) decks weer.
  + Decks kunnen toevoegen, aanpassen, deleten, delen en downloaden.
* **Decks inhoud pagina (DeckContentPage.xaml)**
  + Wordt weergegeven als de gebruiker klikt op een deck
  + Geeft een overzicht weer van alle anki kaarten uit het betreffende deck.
  + Kaarten kunnen toevoegen, aanpassen en deleten.
  + Mogelijkheid om subdecks toe te voegen.
  + Mogelijkheid om een titel aan het deck toe te voegen.
  + Rechts onderin staat het versienummer. Increment with 0,1.
  + Als er feedback is gegeven wordt er een knop met ‘Upload feedback’ zichtbaar.
* **Kaart pagina (CardPage.xaml)**
  + Geeft de vraag weer, alle antwoorden, welke antwoord juist is, welke niet en waarom.
  + Mogelijkheid om velden aan te passen, deleten en toe te voegen.
  + Mogelijkheid om feedback te geven. Dit houd in dat er een rating, opmerkingen en fouten kunnen worden toegevoegd aan andermans kaarten. (Alleen mogelijk bij gedownloade kaarten).
  + Zodra er wat voor feedback dan ook is gegeven moet er een ‘save feedback’ knop zichtbaar worden. Dit stuurt:
    - De gemaakt feedback. Check of er daadwerkelijk feedback is gegeven. Programma’s zoals Postman bestaan en daar moeten we rekening mee houden.
    - Kaart id(’s)
    - Het deck id.
* **Feedback luister achtergrond service**
  + Er moet een achtergrond service worden aangemaakt die luistert naar feedback pingen/meldingen van de server.
  + Nodig om de gebruiker op de hoogte te stellen dat er feedback op zijn of haar gemaakte deck of kaarten is gegeven.

Server-side Maurice

* **Welke API controllers moeten er worden aangemaakt?**
  + Deck delen.
  + Feedback
* **Deck delen**
  + Mogelijkheid om decks te delen. (uploaden en downloaden).
  + Elk opgeslagen deck heeft alle hier bovenstaande facetten en het unieke ‘author id’. (IT-vitae account)
  + Het moet mogelijk zijn een simpel overzicht te downloaden in plaats van dat alle decks ooit gelijk worden gedownload. De gebruiker kan dus kiezen welke decks hij daarna wil downloaden. Het simpele overzicht moet minimaal bestaan uit:
    - Vraag
    - Datum?
    - Author?
* **Er moet database worden aangemaakt waar decks en kaarten kunnen worden opgeslagen.**
  + Entity Framework opzetten.
  + Models aanmaken.
  + Tabellen genereren met ‘model first’ methode.
* **TODO’s van file upload api oplossen.**
* **Mocht de upload author id en titel overeenkomen van een deck, dan moet die kaart worden aangepast. Anderzijds moet er een nieuwe toevoeging worden gemaakt.**
* **Feedback API** 
  + Zie hier bovenstaande bij client-side.
  + Bij ontvangst van feedback 🡪 stuur een ping/melding naar de betreffende client.
* **Verificatie verplaatsen naar serverside.** 
  + Tokens daadwerkelijk gebruiken.
  + Azure portal opgaan kijken wat voor soort tokens er terug gegeven kunnen worden.