

# CodeCadette Requirements

## Functioneel

- Als gebruiker wil ik de mogelijkheid hebben om meerdere users aan te maken en hier tussen te switchen
- Als gebruiker wil ik de mogelijkheid hebben om mijn voortgang te resetten
- Als gebruiker wil ik een hoofdmenu waar ik kan kiezen tussen de verschillende leerdoelen
- Als gebruiker wil ik een uitleg scherm waarin wordt uitgelegd wat de bedoeling van het leerdoel is
- Als gebruiker wil ik na het invullen van mijn antwoorden een scherm waarin wordt geverifieerd of mijn antwoorden goed zijn
- Als gebruiker wil ik een melding krijgen zodra ik een leerdoel af heb gerond
- Als gebruiker wil ik beperkt worden in mijn keuze van antwoorden zodat ik langzaam de juiste kant op wordt begeleid

## Technisch

- Het systeem slaat data op die tussen sessies moet blijven bestaan in een SQLite database
  - Iedere tabel in de database heeft een eigen model klasse
  - De database wordt als App asset gekopieerd de eerste keer dat de applicatie wordt opgestart
  - Queries op de database vinden plaats via het SQLite package
  - Het DatabaseModel is een static klasse, bij het openen van de applicatie wordt de database ge'set', hierna kunnen alle klassen die het DatabaseModel importeren hier gebruik van maken.
  - SQL injectie wordt voorkomen door de standaard SQLite parameter (WhereArgs)
- De applicatie is opgebouwd uit op zichzelf staande Widgets, data mag alleen maar tussen Widgets worden gedeeld die elkaar niet opvolgen in de tree door middel van provider/consumer principe.
- De naamgeving van Widgets, Component Widgets, model klassen en Change Notifier klassen moet duidelijk genoeg zijn om verwarring te voorkomen
- Het ContentControlScreen is een model die aan de hand van de huidige State een scherm (Scaffold Widget) teruggeeft, ook wordt hier de methode gettriggerd op de database waarmee de vragen, en antwoordenlijsten worden opgehaald
  - Als data nog niet is ingeladen een laadscherm
  - Zodra de data is ingeladen het speelscherm
  - Zodra een antwoord is gegeven het verificatiescherm
    - Het verificatiescherm stuurt een gebruiker bij een fout antwoord terug naar het speelscherm met dezelfde vraag

- Het verificatiescherm stuurt een gebruiker bij een goed antwoord terug naar het speelscherm met dezelfde vraag
- Dit sturen gebeurt door middel van een SetState Functie die alleen bestaat in het ContentControlScreen, maar wordt gelinkt aan de knop in het verificatiescherm
- De AnswerBox is een widget waarin dynamisch de vraagstelling wordt opgebouwd aan de hand van de meegegeven AnswerList.
  - als een Antwoord.filledIn == true is, wordt er een Text Widget getoond met de waarde van het antwoord
  - als een Antwoord.filledIn == false is, wordt er een lege DynamicTextField widget getoond, hier moet de gebruiker zelf een antwoord invullen
  - de Text Controller die hangt aan het toetsenbord wordt dynamisch aangepast door middel van het Provider/Consumer principe zodra de gebruiker op een DynamicTextField klikt
  - het AntwoordModel wordt door middel van Provider/Consumer doorgegeven aan het ChoiceBoxKeyboard, waar op deze manier de verificatie kan plaatsvinden zodra men op de enter knop drukt
- ChoiceBoxKeyboard is het custom toetsenbord van de applicatie, het toetsenbord verandert dynamisch op basis van het type vraag zoals gedefiniëerd in de Vraag tabel

## Database

