# Projectoefening Shopr

## Model

* Artikel
  + Id
  + Titel (lengte 100, verplicht)
  + Prijs (0,00, verplicht)
  + Id van de leverancier (string, lengte 100, verplicht)
  + Kan niet op zichzelf bestaan (is dus abstracte klasse, MappedSuperClass)
* Boek (is een artikel, via JOINED inheritance)
  + Auteur (lengte 100)
  + Isbn (bv 978-90-1234-567-8, valideer formaat, verplicht, uniek )
  + Aantal pagina’s
  + is abstracte klasse
* Fictie (is een boek, via SINGLE\_TABLE inheritance, er is dus slechts 1 tabel voor alle boeken)
  + Genre: thriller, fantasy, detective, romance, sci-fi (enumeratie)
  + Korte inhoud (lengte 255)
* Non-fictie (is een boek, via SINGLE\_TABLE inheritance, er is dus slechts 1 tabel voor alle boeken)
  + Onderwerp: geschiedenis, kookboek, wetenschap, sport (enumeratie)
* LP (is een artikel, via JOINED inheritance)
  + Uitvoerder (lengte 100, uniek tesamen met titel, dus combinatie uitvoerder-titel mag slechts 1x voorkomen)
  + Genre: klassiek, pop, rock, dance, r&b, hiphop (enumeratie)
* Game (is een artikel, via JOINED inheritance)
  + Uitgever (lengte 100, uniek tesamen met titel)
  + Minimum leeftijd
  + Genre: MMORPG, RPG, FPS, RTS, RACE (enumeratie)
* Bestelling
  + Id
  + Datum van bestelling (Timestamp)
  + Artikels (1 bestelling heeft meerdere artikels, één artikel kan tot meerdere bestellingen behoren)
  + Per artikel moet ook de huidige prijs en het aantal stuks bijgehouden worden
  + Gebruiker (FK)
* Beoordeling
  + Id
  + Score (0-10)
  + Omschrijving
  + Gebruiker (FK)
  + Artikel (FK)

## Functionaliteit

* Wanneer je de website bezoekt, kies je eerst een gebruiker via een dropdown (we bootsen hiermee login-functionaliteit na)
* We beginnen met het tonen van de gegevens. Zet dus eerst data klaar via een sql-script (bewaar dit script ook in het project)
  + Lijst met alle artikels, gesorteerd op de type, titel, prijs
  + Lijst per type, gesorteerd op de titel, auteur/uitvoerder/uitgever, prijs
  + Detailscherm met alle gegevens voor zowel Fictie, Non-Fictie, LP als Game
    - Met lijst van beoordelingen
* Aanmaken van de artikels
  + Met zowel front-end als back-end validatie
* Aanpassen van de artikels (probeer functionaliteit van aanmaken te herbruiken!)
  + Met zowel front-end als back-end validatie
* Verwijderen van de artikels
  + Knop moet op detailpagina staan
* Zoekpagina:
  + Er kan gezocht worden op id, type, titel EN prijs (max en minimum)
  + Er moet op minstens 2 criteria tesamen gezocht worden, zo niet geef je een foutmelding
  + Je kan ook sorteren op 1 van deze 4 criteria (te kiezen met dropdown, dus sortering via database, geen front-end sortering)
  + Je gaat dus je query dynamisch moeten opbouwen naargelang de zoekparameters
* Winkelmandje
  + Op de detailpagina kan je het artikel toevoegen aan uw winkelmandje. 1 artikel kan ook meerdere keren besteld worden, voorzie dus ook een veld om het aantal artikels in te geven (by default wordt daar 1 ingevuld)
  + Je kan ook bv eerst 2 bepaalde artikels bestellen en later nog 5 toevoegen aan je bestelling
  + Je hebt een overzichtspagina met
    - Aantal en prijs per artikel
      * kolom met type
      * kolom met titel
      * kolom met aantal
      * kolom met eenheidsprijs
      * kolom met totaalprijs
    - Aantal en prijs per type (fictie, non-fictie, lp, game)
      * kolom met type
      * kolom met aantal
      * kolom met totaalprijs
    - Totaal aantal en prijs
    - Knop om te bestellen
      * Plaats bestelling in de database
      * Als succes wordt winkelmandje leeggemaakt
    - Knop om te annuleren
      * Maakt winkelmandje leeg
* SupplierService
  + Bij het plaatsen van een bestelling moet elk item met zijn aantal doorgestuurd worden naar de SupplierService
  + Als een exceptie optreedt moet de hele bestelling geannuleerd worden en een foutboodschap op het scherm verschijnen
* Een gebruiker kan ook zijn favoriete artikels selecteren. Een gebruiker kan meerdere favorieten hebben, een artikel kan het favoriete artikel van meerdere gebruikers zijn (many-to-many). Voorzie op de detailpagina functionaliteit waarmee je kan aangeven dat dit je favoriete artikel is. ( je kan dus ‘togglen’ )
  + Let op: voor deze tabel, willen we geen klasse zien in het Java model. Een gebruiker moet een lijst hebben van favoriete artikels. Een artikel heeft zelf geen weet over welke gebruiker het artikel als favoriet heeft.

## Uitbreidingen

* Oops, die titel moest 255 characters lang zijn (en neen, we gaan niet heel het databaseschema droppen… Gebruik een alter table)
* Voeg een zoektitel toe waarin alle speciale karakters verwijderd worden
  + Bij het zoekscherm moet op deze waarde gezocht worden ( de zoekterm ondergaat dus zelfde bewerking)
  + Wordt automatisch toegevoegd/aangepast bij het bewaren van het artikel (prepersist, preupdate)
* Voeg aan alle entities (beoordeling, artikel, bestelling) auditgegevens toe (zoek op hoe je audit kan toevoegen)
  + Audit (is een embeddable)
    - Tijdstip creatie (timestamp)
    - Tijdstip laatste wijziging (timestamp)
    - Worden automatisch ingevuld bij opslaan en wijzigen
* Voeg Versioning toe (@Version), bestudeer ook waarom dit gebruikt wordt.
* Het winkelmandje moet ook gevuld kunnen worden via de detailschermen (meerdere selecteren)