**FM PROJECT**

**Requirements**

Propose the requirements of a simple CRUD application (4-5 classes, at least 3 non-trivial constraints), within a problem domain of your choice (attached, you can find two samples, of similar complexity to what you are required to accomplish).  
Starting from the associated class diagram, and using the UML to B translation rules discussed in Lecture 6-7, specify the corresponding B system and verify its consistency using AtelierB (type- check, generate PO's and use the automatic prover to discharge them). Interactive proof is not required, but the PO's not handled by the automatic prover should be analysed for validity.

**Solution**

Food tracking system

* classes
  + User, Food, FoodRecord, Meal, FluidRecord, TrackingSystem
  + User:
    - id
    - email
    - password
    - enabled
    - [Obs.: email as id is not suitable because primary key is immutable]
  + Food
    - id
    - name
    - calories/100g
    - carbs/100g
    - fat/100g
    - protein/100g
  + Meal (enum.) = [breakfast, lunch, dinner, snack]
  + Record
    - user: User
    - food: Food
    - date
    - meal: Meal
    - quantity
  + FluidRecord
    - user: User
    - date
    - volume
* constraints
  + User key
  + User.email is unique
  + Food key
  + 0 <= Food.caloriesPer100g <= 100
  + 0 <= Food.carbsPer100g <= 100
  + 0 <= Food.fatPer100g <= 100
  + 0 <= Food.proteinPer100g <= 100
  + 0 <= Food.saltPer100g <= 100
  + FoodRecord key
  + FoodRecord.quantity > 0
  + No CRUD operations on FoodRecord when user is disabled
  + FluidRecord key
  + FluidRecord.volume > 0
  + No CRUD operations on FluidRecord when user is disabled