1. **Inputs given when the user registers into the system**

|  |  |  |
| --- | --- | --- |
| **Input** | **How the input correlates with the ontology** | **Comments/Examples** |
| Name | As a value of the data type property hasName associated to the PersonalData concept  Prerequisites: the property hasName must be added: domain=PersonalData, range=string | Add the value “John Brown” for the data type property hasName associated to the instance PersonalDataForJohnBrown of the PersonalData concept (this also implies adding the instance PersonalDataForJohnBrown to the PersonalData concept) |
| Address | Prerequisites: the property hasGeographicalArea must be added: domain=PersonalData, range= GeographicalArea  Similar to the name input |  |
| Gender | As a value of the data type property hasGender associated to the PersonalData concept | Add the “male” value for the data type property hasGender associated to the individual PersonalDataForJohnBrown |
| Age | As a value of the data type property hasAge associated to the PersonalData concept  Prerequisites: the property hasAge must be added: domain=PersonalData, range=integer |  |
| Weight | As a value of the data type property hasWeight associated to the AnthropometricMeasurements concept Prerequisites: the property hasAnthropometricMeasurements must be added: domain= PersonalData, range= AnthropometricMeasurements | Add a numerical value for the data type property hasWeight associated to the individual AnthropometricMeasurementsForJohnBrown (this implies adding an instance to the AnthropometricMeasurements concept named AnthropometricMeasurementsForJohnBrown) |
| Height | As a value of the data type property hasHeight associated to the AnthropometricMeasurements concept | Add a numerical value for the data type property hasHeight associated to the individual AnthropometricMeasurementsForJohnBrown |
| BMI – this is automatically computed based on the given weight and height | As a value of the data type property hasBMI associated to the AnthropometricMeasurements concept | Add the computed numerical value for the data type property hasBMI associated to the individual AnthropometricMeasurementsForJohnBrown |
| Physical activity | Prerequisites: the property hasPhysicalActivity must be added: domain= PersonalData, range= PhysicalActivity |  |
| PAF | As a value of the data type property hasPAF associated to the PhysicalActivity concept | PhysicalActivityForJohnBrown hasPAF pafvalue  \*possibilities for pafvalue presented in Personal-Health-Profile.doc |
| Allergies | Prerequisites: The object property hasAllergies must be added: domain=PersonData, range=Allergies  The object property hasAllergyToBasicFood must be added: domain= Allergies, range=BasicFood  As a value representing a basic food of the object type property hasAllergyToBasicFood | PersonalDataForJohnBrown hasAllergies AllergiesForJohnBrown (instance of the Allergies concept) hasAllergiesToBasicFood Nuts |
| Diseases | Prerequisites: The object property hasDiseases must be added: domain=PersonData, range=Diseases | PersonalDataForJohnBrown hasDiseases KneesArthritis |
| Food preferences | Prerequisites: add the object type property hasFoodPreferences with the domain=PersonData and range=BasicFood U CombinedFood |  |
| Price preferences | Prerequisites: add the data type property hasPricePreferences with the domain=PersonData and range=integer |  |

1. **Personal Health Profile Generation From Input Data**

\*also see Personal-Health-Profile.doc

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Value/Concept** | **How the value/concept correlates with the ontology** | | | **Examples/Comments** |
| Weight Diagnostic | Prerequisites: The object property hasWeightDiagnostic must be added: domain=PersonData, range=WeightDiagnostic | | | PersonalDataForJohnBrown hasWeightDiagnostic one of:   * Underweight * NormalWeight * OverWeight * ObesityClass1 * ObesityClass2 * MorbidObesity |
| PersonalData(?p), hasAnthropometricMeasurements(?p, ?a), hasBMI(?d, ?bmi), lessThan(?bmi, "18.5"^^double) -> hasWeightDiagnostic(?p, Underweight)  PersonalData(?p), hasAnthropometricMeasurements(?p, ?a), hasBMI(?a, ?bmi), greaterThanOrEqual(?bmi, "18.5"^^double), lessThan(?bmi, "25.0"^^double) -> hasWeightDiagnostic(?p, NormalWeight)  PersonalData(?p), hasAnthropometricMeasurements(?p, ?a), hasBMI(?a, ?bmi), greaterThanOrEqual(?bmi, "25.0"^^double), lessThan(?bmi, "30.0"^^double) -> hasWeightDiagnostic(?p, Overweight)  PersonalData(?p), hasAnthropometricMeasurements(?p, ?a), hasBMI(?a, ?bmi), greaterThanOrEqual(?bmi, "30.0"^^double), lessThan(?bmi, "35.0"^^double) -> hasWeightDiagnostic(?p, ObesityClass1)  PersonalData(?p), hasAnthropometricMeasurements(?p, ?a), hasBMI(?a, ?bmi), greaterThanOrEqual(?bmi, "35.0"^^double), lessThan(?bmi, "40.0"^^double) -> hasWeightDiagnostic(?p, ObesityClass2) | | | | |
| Daily intake values | Prerequisites: The object property hasDailyIntakeValues must be added: domain=PersonData, range=DailyIntakeValues | | | PersonalDataForJohnBrown hasDailyIntakeValues DailyIntakeValuesForJohnBrown |
| BMR (kcal) | As a value of the data type property divHasBMR associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasBMR value  \*value computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasAnthropometricMeasurements(?p, ?am), hasDailyIntakeValues(?p, ?d), hasGender(?p, "male"^^string), hasHeight(?am, ?h), hasWeight(?am, ?w), hasAge(?p, ?a), add(?r1, ?m1, 66), add(?r2, ?r1, ?m2), multiply(?m1, ?w, "13.75"^^double), multiply(?m2, ?h, 5), multiply(?m3, ?a, "6.76"^^double), subtract(?rez, ?r2, ?m3) -> divHasBMR(?d, ?rez)  Female  PersonalData(?p), hasAnthropometricMeasurements(?p, ?am), hasDailyIntakeValues(?p, ?d), hasGender(?p, "female"^^string), hasHeight(?am, ?h), hasWeight(?am, ?w), hasAge(?p, ?a), add(?r1, ?m1, 655), add(?r2, ?r1, ?m2), multiply(?m1, ?w, "9.56"^^double), multiply(?m2, ?h, "1.85"^^double), multiply(?m3, ?a, "4.68"^^double), subtract(?rez, ?r2, ?m3) -> divHasBMR(?d, ?rez) | | | | |
| Basic energy intake (kcal) | As a value of the data type property divHasBasicEnergyNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasBasicEnergyNeed value  \*value computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasPhysicalActivity(?p, ?pa), hasPAF(?pa, ?paf), divHasBMR(?d, ?bmr), multiply(?rez, ?bmr, ?paf) -> divHasBasicEnergyNeed(?d, ?rez) | | | | |
| Final Energy Need (kcal) | As a value of the data type property divHasFinalEnergyNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasFinalEnergyNeed value  \*value computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, Underweight), hasGender(?p, "female"^^string), add(?rez, ?be, 125) -> divHasFinalEnergyNeed(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, Underweight), hasGender(?p, "male"^^string), add(?rez, ?be, 250) -> divHasFinalEnergyNeed(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, NormalWeight) -> divHasFinalEnergyNeed(?d, ?be)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, Overweight), multiply(?m, ?be, 0.15), subtract(?rez, ?be, ?m) -> divHasFinalEnergyNeed(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, ObesityClass1), multiply(?m, ?be, 0.17), subtract(?rez, ?be, ?m) -> divHasFinalEnergyNeed(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, ObesityClass2), multiply(?m, ?be, 0.2), subtract(?rez, ?be, ?m) -> divHasFinalEnergyNeed(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasBasicEnergyNeed(?d, ?be), hasWeightDiagnostic(?p, MorbidObesity), multiply(?m, ?be, 0.25), subtract(?rez, ?be, ?m) -> divHasFinalEnergyNeed(?d, ?rez) | | | | |
| Carbohydrates  (grams) | As a value of the data type properties divHasLowerCarbsLimit and divHasUpperCarbsLimit associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasLowerCarbsLimit value1 and divHasUpperCarbsLimit value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.45), divide(?rez, ?m, 4) -> divHasLowerCarbsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.65), divide(?rez, ?m, 4) -> divHasUpperCarbsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasWeightDiagnostic(?p, Overweight), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.5), divide(?rez, ?m, 4) -> divHasUpperCarbsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasWeightDiagnostic(?p, ObesityClass1), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.5), divide(?rez, ?m, 4) -> divHasUpperCarbsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasWeightDiagnostic(?p, ObesityClass2), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.5), divide(?rez, ?m, 4) -> divHasUpperCarbsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasWeightDiagnostic(?p, MorbidObesity), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.5), divide(?rez, ?m, 4) -> divHasUpperCarbsLimit(?d, ?rez) | | | | |
| Proteins  (grams) | As a value of the data type properties divHasLowerProteinLimit, divHasUpperProteinLimit and divHasAvgProteinNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasLowerProteinLimit value1, divHasUpperProteinLimit value2 and divHasAvgProteinNeed value3  \*value1, value2, value3 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.1), divide(?rez, ?m, 4) -> divHasLowerProteinLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.35), divide(?rez, ?m, 4) -> divHasUpperProteinLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAnthropometricMeasurements(?p, ?a), hasWeight(?a, ?w), multiply(?rez, ?w, 0.75) -> divHasAvgProteinNeed(?d, ?rez) | | | | |
| Fats  (grams) | As a value of the data type properties divHasLowerFatsLimit and divHasUpperFatsLimit associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasLowerFatsLimit value1 and divHasUpperFatsLimit value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.2), divide(?rez, ?m, 4) -> divHasLowerFatsLimit(?d, ?rez)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), divHasFinalEnergyNeed(?d, ?fe), multiply(?m, ?fe, 0.35), divide(?rez, ?m, 4) -> divHasUpperFatsLimit(?d, ?rez) | | | | |
| Vitamin A (milligrams) | As a value of the data type properties divHasVitaminANeed and divHasMaxVitaminANeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasVitaminANeed value1 and divHasMaxVitaminANeed value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d) , hasGender(?p, "male"^^string) -> divHasVitaminANeed(?d, 0.9)  PersonalData(?p), hasDailyIntakeValues(?p, ?d) , hasGender(?p, "female"^^string) -> divHasVitaminANeed(?d, 0.7)  PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasMaxVitaminANeed(?d, 3.0) | | | | |
| Vitamin B (milligrams) | As a value of the data type properties divHasVitaminBNeed and divHasMaxVitaminBNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasVitaminBNeed value1 and divHasMaxVitaminBNeed value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasVitaminBNeed(?d, 1.3)  PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasMaxVitaminBNeed(?d, 100.0) | | | | |
| Vitamin C (milligrams) | As a value of the data type properties divHasVitaminCNeed and divHasMaxVitaminCNeed associated to the DailyIntakeValues concept | | | As a value of the data type properties divHasVitaminCNeed and divHasMaxVitaminCNeed associated to the DailyIntakeValues concept |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasGender(?p, "male"^^string) -> divHasVitaminCNeed(?d, 90.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasGender(?p, "female"^^string) -> divHasVitaminCNeed(?d, 75.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasMaxVitaminCNeed(?d, 2000.0) | | | | |
| Vitamin D (milligrams) | | As a value of the data type properties divHasVitaminDNeed and divHasMaxVitaminDNeed associated to the DailyIntakeValues concept | As a value of the data type properties divHasVitaminDNeed and divHasMaxVitaminDNeed associated to the DailyIntakeValues concept | |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 1) -> divHasVitaminDNeed(?d, 0.01)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 70), greaterThanOrEqual(?a, 1) -> divHasVitaminDNeed(?d, 0.015)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 70) -> divHasVitaminDNeed(?d, 0.02)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 1) -> divHasMaxVitaminDNeed(?d, 0.038)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 3), greaterThanOrEqual(?a, 1) -> divHasMaxVitaminDNeed(?d, 0.063)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 9), greaterThanOrEqual(?a, 3) -> divHasMaxVitaminDNeed(?d, 0.075)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 9) -> divHasMaxVitaminDNeed(?d, 0.1) | | | | |
| Calcium (milligrams) | As a value of the data type properties divHasCalciumNeed and divHasMaxCalciumNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasCalciumNeed value1 and divHasMaxCalciumNeed value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 50) -> divHasCalciumNeed(?d, 1000.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 50) -> divHasCalciumNeed(?d, 1200.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasMaxCalciumNeed(?d, 2000.0) | | | | |
| Iron (milligrams) | As a value of the data type properties divHasIronNeed and divHasMaxIronNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasIronNeed value1 and divHasMaxIronNeed value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 1) -> divHasIronNeed(?d, 11.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 3), greaterThanOrEqual(?a, 1) -> divHasIronNeed(?d, 7.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 8), greaterThanOrEqual(?a, 3) -> divHasIronNeed(?d, 10.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 13), greaterThanOrEqual(?a, 8) -> divHasIronNeed(?d, 8.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 18), greaterThanOrEqual(?a, 14), hasGender(?p, "male"^^string) -> divHasIronNeed(?d, 11.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 18), greaterThanOrEqual(?a, 14), hasGender(?p, "female"^^string) -> divHasIronNeed(?d, 15.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 50), greaterThanOrEqual(?a, 18), hasGender(?p, "male"^^string) -> divHasIronNeed(?d, 8.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 50), greaterThanOrEqual(?a, 18), hasGender(?p, "female"^^string) -> divHasIronNeed(?d, 18.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 50) -> divHasIronNeed(?d, 8.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 19) -> divHasMaxIronNeed(?d, 40.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 19) -> divHasMaxIronNeed(?d, 45.0) | | | | |
| Sodium (milligrams) | As a value of the data type properties divHasSodiumNeed and divHasMaxSodiumNeed associated to the DailyIntakeValues concept | | | DailyIntakeValuesForJohnBrown divHasSodiumNeed value1 and divHasMaxSodiumNeed value2  \*value1, value2 computed using formulas from Personal-Health-Profile.doc translated into SWRL rules |
| PersonalData(?p), hasDailyIntakeValues(?p, ?d) -> divHasSodiumNeed(?d, 1500.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), lessThan(?a, 51) -> divHasMaxSodiumNeed(?d, 2300.0)  PersonalData(?p), hasDailyIntakeValues(?p, ?d), hasAge(?p, ?a), greaterThanOrEqual(?a, 51) -> divHasMaxSodiumNeed(?d, 1500.0) | | | | |

EXAMPLE:  
- Input: username: user1, name: John Roberts, age:62, gender: male, weight:79, height: 180, pricePref: 50

|  |  |  |
| --- | --- | --- |
| Individual | Attributes | Values |
| PersonalDataForJohnRoberts | hasUsername | "user1"^^xsd:string |
|  | hasName | "John Roberts"^^xsd:string |
|  | has Age | 62 |
|  | hasGender | "male"^^xsd:string |
|  | hasGeographicalArea | GeographicalArea40001 |
|  | hasAnthropometricMeasurements | AnthropometricMeasurementsForJohnRoberts |
|  | hasPricePreferences | 50 |
|  | has Allergies | AlergiesForJohnRoberts |
|  | hasDiseases | KneesArthritis |
|  | hasPreferences | - |
|  | hasDailyIntakeValues | DailyIntakeValuesForJohnRoberts |
| GeographicalArea40001 | geographicalAreaHasId | 40001 |
|  | geographicalAreaHasCountry | "Romania"^^xsd:string |
|  | geographicalAreaHasCity | "Cluj"^^xsd:string |
|  | geographicalAreaHasRegion | "Calea Motilor"^^xsd:string |
| AnthropometricMeasurementsForJohnRoberts | hasHeight | 180.0E0 |
|  | hasWeight | 79.0E0 |
|  | hasBMI | "24.382716049382715"^^xsd:double |
| DailyIntakeValuesForJohnRoberts | hasBMR | "1633.13"^^xsd:double |
|  | divHasBasicEnergyNeed | "1959.756"^^xsd:double |
|  | divHasFinalEnergyNeed | "1959.756"^^xsd:double |
|  | divHasLowerCarbsLimit | "220.47255"^^xsd:double |
|  | divHasUpperCarbsLimit | "318.46035"^^xsd:double |
|  | divHasLowerProteinLimit | "48.993900000000004"^^xsd:double |
|  | divHasUpperProteinLimit | "171.47865"^^xsd:double |
|  | divHasAvgProteinNeed | "59.25"^^xsd:double |
|  | divHasLowerFatsLimit | "97.98780000000001"^^xsd:double |
|  | divHasUpperFatsLimit | "171.47865"^^xsd:double |
|  | divHasVitaminANeed | "0.9"^^xsd:double |
|  | divHasMaxVitaminANeed | "3.0"^^xsd:double |
|  | divHasVitaminBNeed | "1.3"^^xsd:double |
|  | divHasMaxVitaminBNeed | "100.0"^^xsd:double |
|  | divHasVitaminCNeed | "90.0"^^xsd:double |
|  | divHasMaxVitaminCNeed | "2000.0"^^xsd:double |
|  | divHasVitaminDNeed | "0.015"^^xsd:double |
|  | divHasMaxVitaminDNeed | "0.1"^^xsd:double |
|  | divHasCalciumNeed | "1200.0"^^xsd:double |
|  | divHasMaxCalciumNeed | "2000.0"^^xsd:double |
|  | divHasIronNeed | "8.0"^^xsd:double |
|  | divHasMaxIronNeed | "45.0"^^xsd:double |
|  | divHasSodiumNeed | "1500.0"^^xsd:double |
|  | divHasMaxSodiumNeed | "1500.0"^^xsd:double |

1. **Ontology population**

* Add instances for the FoodServiceProvider concept
  + We consider that each provider instance provides only daily menus
  + Each instance has an id, a name, and geographical area
* Add instances for the DailyMenu concept – these are the actual packages
  + Each instance has an id, and a menu associated
* Add instances for the Menu concept
  + Each instance has an id and a meal variant associated
* Add instances for the MealVariant concept
  + Each instance has an id, a diet associated (halal, nutrition modified, texture modified, vegetarian), a meal type associated (breakfast, lunch, dinner, snack), and a dish (dessert, main course, starter)
* Add instances for the Dish concept
  + Each instance has an id, and a recipe associated
* Add instances for the Recipe concept
  + Each instance has an ID, a cooking method, and a set of basic food items associated
* Add instances for the BasicFood concept
  + Add nutritional facts to the instances of the BasicFood concept – through a relation with the NutritionIntake concept