Understanding Repository - 04

Eurico Costa

CIT-360 W16

# UML

This week I’ve been working on creating the UML diagrams for my personal app, as such, based on the research I did last week, I have created the following diagrams, all based on the Missing Person Finder Application:

**Use Case Diagram**: <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2004/Use%20Case%20Model%20-%20Missing%20Person%20Finder.png>

**Use Case Document**: <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2004/Use%20Case%20Document%20-%20Missing%20Person%20Finder.docx>

**Class Diagram**: <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2004/Class%20Model%20-%20Missing%20Person%20Finder.png>

**State Diagram**: <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2004/State%20Diagram%20-%20Missing%20Persons%20Finder%20Lifecycle.png>

# Hibernate: One-to-Many Example

The link to the code below expands on the previous hibernate example to use a one-to-many relationship map. The code allows the user to select a record to display from the list, and uses the toString() method to display the details of the missing person (Person class), and all the notes (Notes class) associated to the person.

Hibernate mapping file with one-to-many relationship (lines 56-59): <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2002/Hibernate/mpfinder/resources/person.hbm.xml>

Person POJO, with the Notes attribute that will contain the associated notes, as well as getter/setters for this Set collection: <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2002/Hibernate/mpfinder/src/com/cit360/mpfinder/model/Person.java>

Display of the associated Notes, resulting from the one-to-many relationship (lines 76-90): <https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2002/Hibernate/mpfinder/src/com/cit360/mpfinder/MissingPersonFinder.java>

# Collections: TreeMap, Map, Set

Enhanced the HTTPUrlConnection example with an example of how to use a TreeMap, which is a sorted list. The two examples of collections in this code shows getting header key pairs from an http request using Map and Set (lines 26 to 41), and displaying the unsorted result, and then, using a TreeMap (lines 38-57), a sorted collection, showing that the output comes out sorted, without additional code. It shows the advantages of a sorted collection, when the result needs to be sorted:

<https://github.com/Costa-Eurico/CIT-360-W16-Understanding-Portfolio/blob/master/Understanding%20Portfolio%20Submission%2002/HttpURLConnection/src/HttpUrlConnectionDemo.java>

The example also uses JSON. It creates a JSON object out of the response from the http request, and accesses the JSON object to manipulate and display the results from the call to the weather underground api. I believe this example was missed in my last submission, and not accounted for (lines 68-81).