Additional Specs for Phases 2

In phase 2 you will have to create models for `Employee`, `Store` and `Assignment` and write unit tests for these models. To make sure you are able to build these models and tests, here are some specs for each of the three models:

Stores must:

1. have all proper relationships specified
2. have a name, street, and city
3. values which are the proper data type and within proper ranges
4. only states which are currently allowed are PA, OH, WV
5. have phone values that are saved in the system as a string of digits (no other characters allowed in db, but user may input values with dashes)
6. have store names which are unique in the system (case insensitive)
7. have the following scopes:
   a) 'active' -- returns only active stores
   b) 'inactive' -- returns all inactive stores
   c) 'alphabetical' -- orders results alphabetically

Employees must:

1. have all proper relationships specified
2. have a first name, last name, role, and a social security number
3. values which are the proper data type and within proper ranges
4. have ssn and phone values that are saved in the system as a string of digits (no other characters allowed in the database, but user may input values with dashes or spaces)
5. have the following scopes:
   a) 'younger_than_18' -- returns all employees under 18 years old
   b) 'is_18_or_older' -- returns all employees 18 years old or older
   c) 'active' -- returns all active employees
   d) 'inactive' -- returns all inactive employees
   e) 'regulars' -- returns all employees who have the role 'employee'
   f) 'managers' -- returns all employees who have the role 'manager'
   g) 'admins' -- returns all employees who have the role 'admin'
   h) 'alphabetical' -- orders results alphabetically
6. have the following methods:
   a) 'name' -- which returns the employee name as a string "last_name, first_name" in that order

b) 'proper_name' -- which returns the employee name as a string "first_name last_name" in that order

c) 'current_assignment' -- which returns the employee's current assignment or nil if the employee does not have a current assignment.

d) 'over_18?' -- which returns a boolean indicating whether this employee is over 18 or not

e) 'age' -- which returns the employee's current age

Assignments must:

1. have all proper relationships specified
2. have a store id, employee id, and a start date
3. values which are the proper data type and within proper ranges
4. have no start dates in the future or end dates that precede start dates
5. have employees and stores that are active and in the system (*applies to new assignments only*)
6. have the following scopes:

   a) 'current' -- which returns all the assignments that are considered current

   b) 'past' -- which returns all the assignments that have terminated

   c) 'for_store' -- which returns all assignments that are associated with a given store (parameter: store object)

   d) 'for_employee' -- which returns all assignments that are associated with a given employee (parameter: employee object)

   e) 'for_role' -- which returns all assignments that are associated with employees of a given role (parameter: role)

   f) 'by_store' -- which orders assignments by store

   g) 'by_employee' -- which orders assignments by employee name (last, first)

   h) 'chronological' -- which orders assignments chronologically with the most recent assignments listed first

6. have the following methods:

   a) 'end_previous_assignment' -- this method will be used as a part of a callback when creating a new assignment. (callbacks will be explained later in class). Essentially this method and callback will update any previously open assignment (if applicable) and terminate it by automatically by setting the end date of the old assignment to the start date of the new assignment.