Land Use Permits I

The City of Seattle's Department of Planning and Development makes their database of Land Use Permit Applications publicly available at https://data.seattle.gov/Permitting/Land-Use-Permits/uyyd-8gak. Your task is to build a Core Data application to operate on the information from their database. This week you'll build the structure of the application, and populate it with dummy data. Next week you'll populate the application with real data, downloaded live.

Functionality and architecture

A document-based Core Data application (leave ARC enabled).

Three Core Data entities: Application, Property, Applicant.

Applicant: needs only one attribute, "name", a String.

Property: needs three attributes, "address" (String), and "latitude" and "longitude" (both Doubles).

Application: can use as many of the attributes in the web-based database viewer as you like. At a minimum, you must have: "applicationDate" (Date) and String attributes for status URL, permit number, and the category, types, and description on the permit.

Set up relationships between Applicant and Application, and between Property and Application. Each Application has exactly one Property and exactly one Applicant. A Property could be the subject of many Applications though, and an Applicant could have submitted many Applications (on the same or on multiple properties).

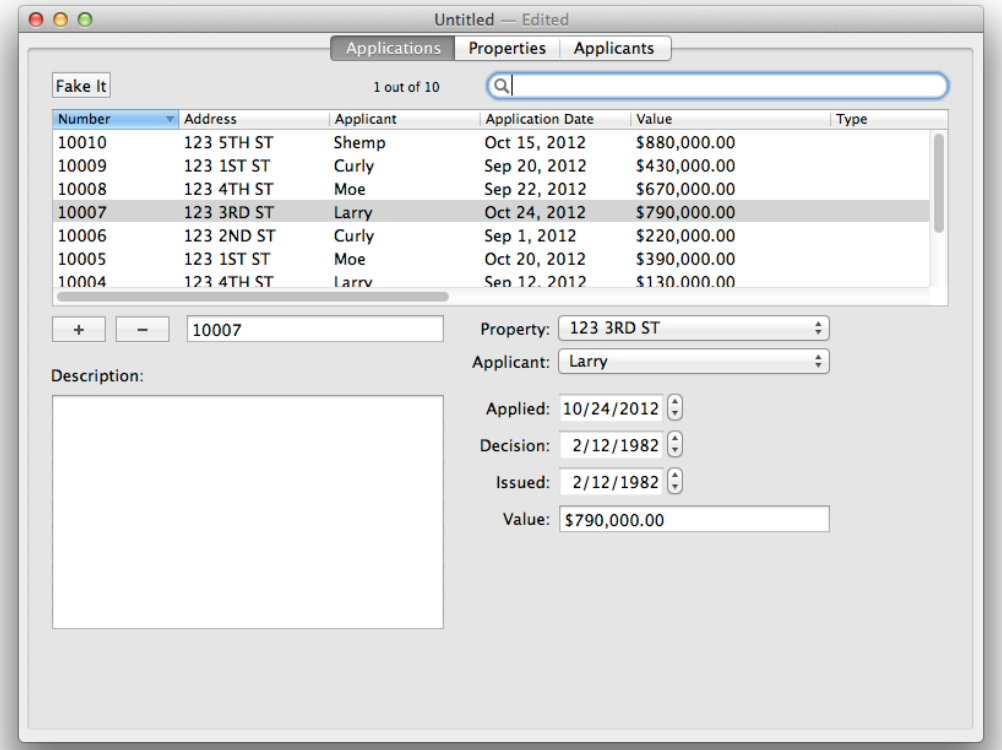
Create NSManagedObject subclasses for all three of these entities.

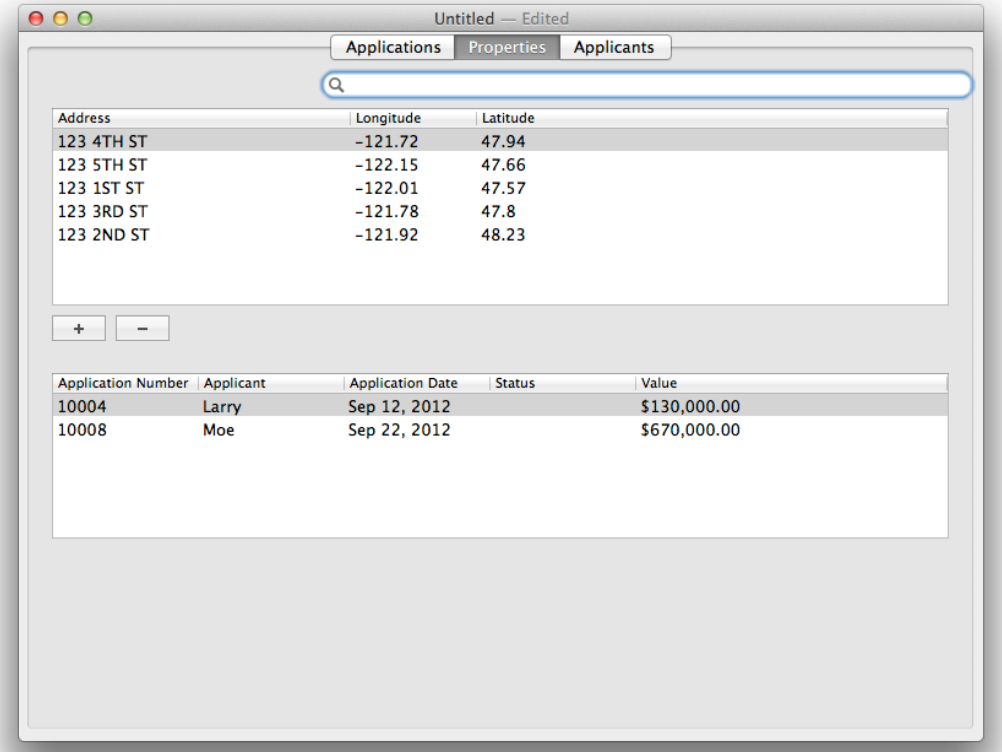
Implement find-or-create methods for Property, Applicant, and Application. Assume that your data will have unique values for Property address, Applicant name, and Application permit number (that is, treat every Applicant with the same name as the same Appliant). These should be class methods on the respective NSManagedObject subclasses. You'll probably want to create named Fetch Requests to do these searches.

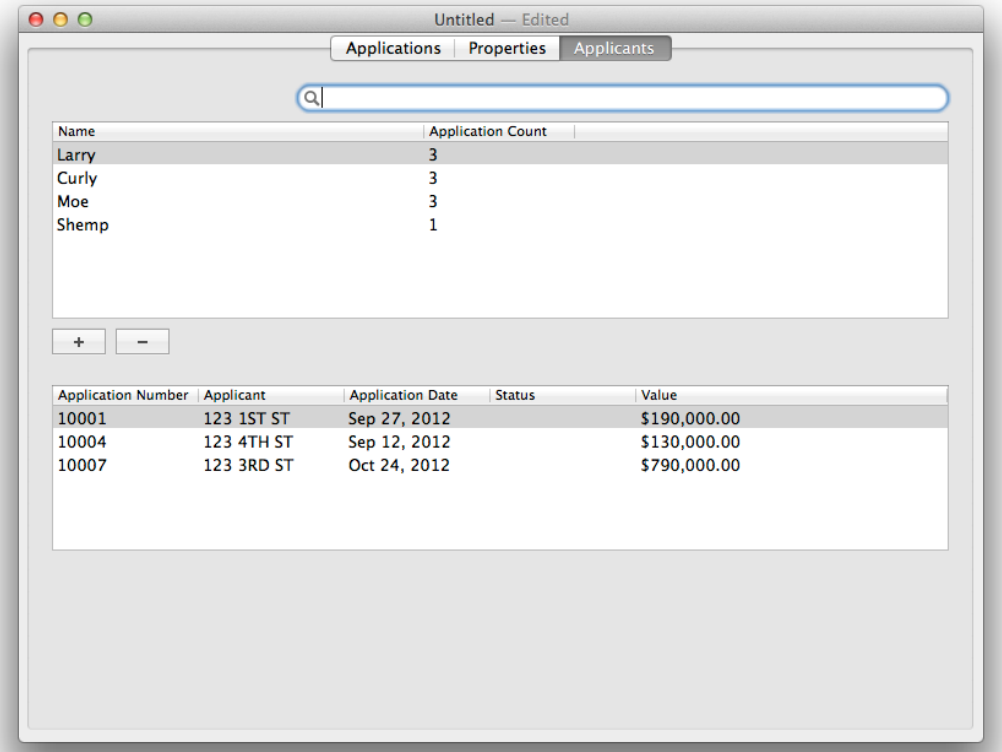
Provide a method to create a few sample entries for each entity, and to link them together appropriately. You may use mine as a starting point, or simply take it as is.

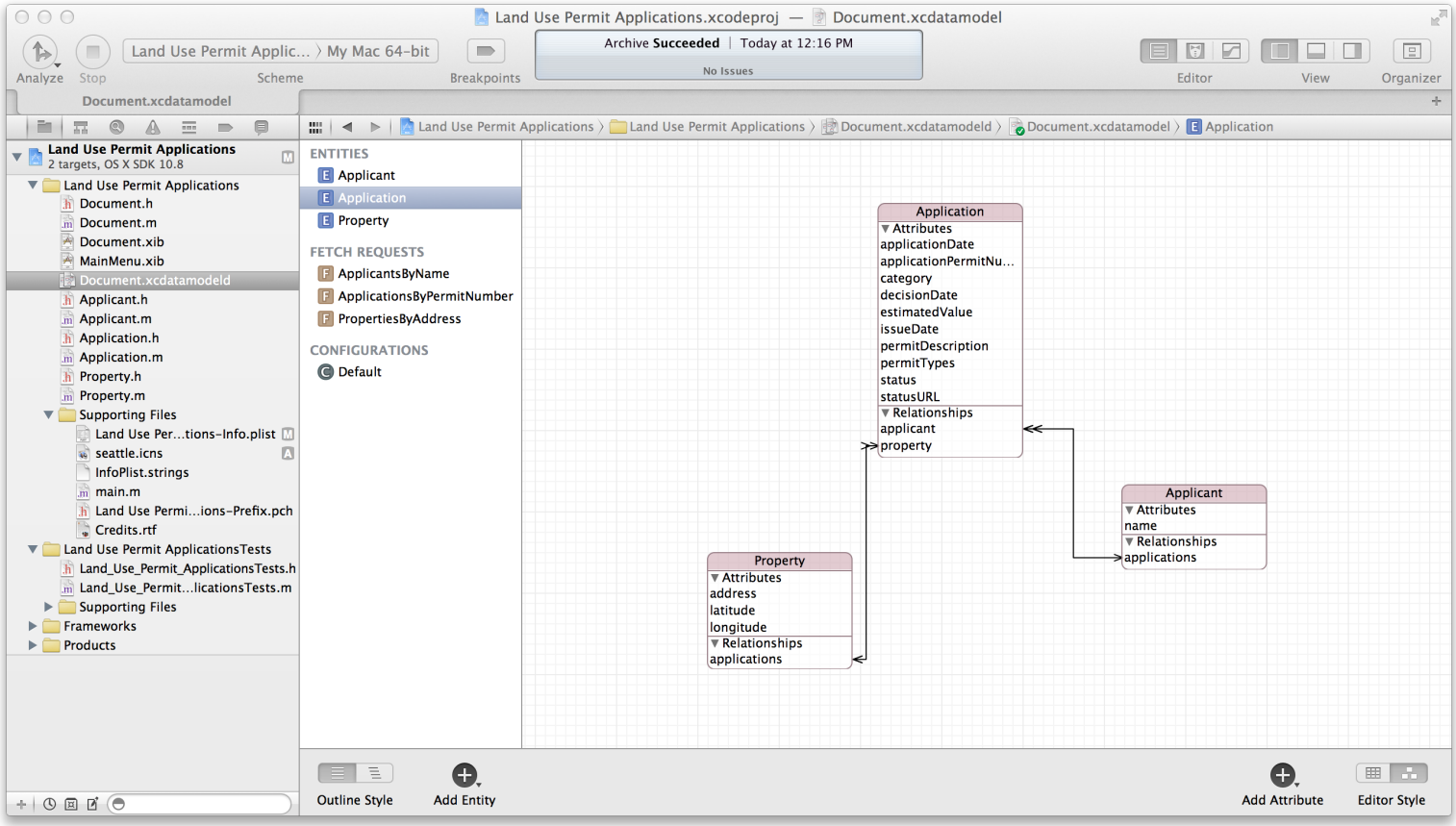
Use a tabview based UI. At a minimum, the Properties tab shows all Properties and the Applicants tab shows all Applicants, with insert/remove buttons. The extra tables, the ones at the bottom that show the Applications linked to the selected Properties or Applicants, are optional. Applications tab shows a master-detail view, with insert/remove buttons, popups for choosing Property and Applicant, and some UI of your choosing for editing the other fields (you don't have to use the date steppers).

Do not use mogenerator.


Untitled — Edited
Applications
Properties
Applicants
Fake It
1 out of 10
Number
Address
Applicant
Application Date
Value
Type
10010 123 5TH ST Shemp Oct 15, 2012 $880,000.00
10009 123 1ST ST Curly Sep 20, 2012 $430,000.00
10008 123 4TH ST Moe Sep 22, 2012 $670,000.00
10007 123 3RD ST Larry Oct 24, 2012 $790,000.00
10006 123 2ND ST Curly Sep 1, 2012 $220,000.00
10005 123 1ST ST Moe Oct 20, 2012 $390,000.00
10004 123 4TH ST Larry Sep 12, 2012 $130,000.00
+
−
10007
Property: 123 3RD ST
Applicant: Larry
Description:
Applied: 10/24/2012
Decision: 2/12/1982
Issued: 2/12/1982
Value: $790,000.00


Untitled — Edited
Applications
Properties
Applicants
Address
Longitude
Latitude
123 4TH ST -121.72 47.94
123 5TH ST -122.15 47.66
123 1ST ST -122.01 47.57
123 3RD ST -121.78 47.8
123 2ND ST -121.92 48.23
+
−
Application Number
Applicant
Application Date
Status
Value
10004 Larry Sep 12, 2012 $130,000.00
10008 Moe Sep 22, 2012 $670,000.00


Untitled — Edited
Applications
Properties
Applicants
Name
Application Count
Larry 3
Curly 3
Moe 3
Shemp 1
+
−
Application Number
Applicant
Application Date
Status
Value
10001 123 1ST ST Sep 27, 2012 $190,000.00
10004 123 4TH ST Sep 12, 2012 $130,000.00
10007 123 3RD ST Oct 24, 2012 $790,000.00


Land Use Permit Applications.xcodeproj — Document.xcdatamodel
Archive Succeeded | Today at 12:16 PM
No Issues
Analyze
Stop
Scheme
Breakpoints
Editor
View
Organizer
Land Use Permit Applications
2 targets, OS X SDK 10.8
Document.h
Document.m
Document.xib
MainMenu.xib
Document.xcdatamodeld
Applicant.h
Applicant.m
Application.h
Application.m
Property.h
Property.m
Supporting Files
Land Use Permit ApplicationsTests
Frameworks
Products
ENTITIES
Applicant
Application
Property
FETCH REQUESTS
ApplicantsByName
ApplicationsByPermitNumber
PropertiesByAddress
CONFIGURATIONS
Default
Application
Attributes
applicationDate
applicationPermitNu…
category
decisionDate
estimatedValue
issueDate
permitDescription
permitTypes
status
statusURL
Relationships
applicant
property
Applicant
Attributes
name
Relationships
applications
Property
Attributes
address
latitude
longitude
Relationships
applications
Outline Style
Add Entity
Add Attribute
Editor Style

Optional Challenges

Implement -awakeFromInsert to populate the latitude, longitude, value, application date, and other attributes.

Provide search boxes.

Implement the per-applicant/per-property subtables.

Note: Using Xcode's automatic generation of .h/.m files for your NSManagedObject subclasses works great the first time you do it. But as you modify the model, you'll need to regenerate those files, which will overwrite any additional code you've already written. Consider using an Objective-C Category to separate your own code from the stuff Xcode generates. I'll show you a better approach than that next week.

Grading Focus

Core Data model implemented, with correct entities, attributes, and relationships.

NSManagedObject subclasses are provided.

UI works without crashing, and provides ability to edit each piece of information.

A method is provided and used to populate the datastore with sample data.

Note: There are lots of fiddly little bits to get all the tables to preserve sorting and column ordering, and to permit/deny editing in certain places. Those bits are not the focus of these exercises. Although my sample does preserve that information, your solution does not need to do that.

Starting Code

Here's my -fakeIt: method, implemented in my NSPersistentDocument subclass:

```
- (IBAction)fakeIt:(id)sender
{
    Applicant *larry = [Applicant findOrCreateApplicantWithName:@"Larry"
                                                        context:self.managedObjectContext];
    Applicant *moe = [Applicant findOrCreateApplicantWithName:@"Moe"
                                                      context:self.managedObjectContext];
    Applicant *curly = [Applicant findOrCreateApplicantWithName:@"Curly"
                                                        context:self.managedObjectContext];

    Property *one = [Property findOrCreatePropertyWithAddress:@"123 1ST ST"
                                                      context:self.managedObjectContext];
    Property *two = [Property findOrCreatePropertyWithAddress:@"123 2ND ST"
                                                      context:self.managedObjectContext];
    Property *three = [Property findOrCreatePropertyWithAddress:@"123 3RD ST"
                                                        context:self.managedObjectContext];
    Property *four = [Property findOrCreatePropertyWithAddress:@"123 4TH ST"
                                                       context:self.managedObjectContext];

    Application *a = [Application findOrCreateApplicationWithPermitNumber:@"10001"
                                                                  context:self.managedObjectContext];
    a.applicant = larry;
    a.property = one;
```

```
    Application *b = [Application findOrCreateApplicationWithPermitNumber:@"10002"
                                                                  context:self.managedObjectContext];
    b.applicant = moe;
    b.property = two;

    Application *c = [Application findOrCreateApplicationWithPermitNumber:@"10003"
                                                                  context:self.managedObjectContext];
    c.applicant = curly;
    c.property = three;

    Application *d = [Application findOrCreateApplicationWithPermitNumber:@"10004"
                                                                  context:self.managedObjectContext];
    d.applicant = larry;
    d.property = four;

    Application *e = [Application findOrCreateApplicationWithPermitNumber:@"10005"
                                                                  context:self.managedObjectContext];
    e.applicant = moe;
    e.property = one;

    Application *f = [Application findOrCreateApplicationWithPermitNumber:@"10006"
                                                                  context:self.managedObjectContext];
    f.applicant = curly;
    f.property = two;

    Application *g = [Application findOrCreateApplicationWithPermitNumber:@"10007"
                                                                  context:self.managedObjectContext];
    g.applicant = larry;
    g.property = three;

    Application *h = [Application findOrCreateApplicationWithPermitNumber:@"10008"
                                                                  context:self.managedObjectContext];
    h.applicant = moe;
    h.property = four;

    Application *i = [Application findOrCreateApplicationWithPermitNumber:@"10009"
                                                                  context:self.managedObjectContext];
    i.applicant = curly;
    i.property = one;

    Application *j = [Application findOrCreateApplicationWithPermitNumber:@"10010"
                                                                  context:self.managedObjectContext];
    j.applicant = [Applicant findOrCreateApplicantWithName:@"Shemp" context:self.managedObjectContext];
    j.property = [Property findOrCreatePropertyWithAddress:@"123 5TH ST"
                                                   context:self.managedObjectContext];
}
```

Headers for my find-or-create methods:

Applicant:

```
+ (Applicant *)findOrCreateApplicantWithName:(NSString *)searchName
                                     context:(NSManagedObjectContext *)moc;
+ (NSArray *)applicantsWithName:(NSString *)searchName
                        context:(NSManagedObjectContext *)moc;
```

Property:

```
+ (Property *)findOrCreatePropertyWithAddress:(NSString *)searchName
                                      context:(NSManagedObjectContext *)moc;
+ (NSArray *)propertiesWithAddress:(NSString *)searchName
                           context:(NSManagedObjectContext *)moc;
```

Application:

```
+ (Application *)findOrCreateApplicationWithPermitNumber:(NSString *)searchName
                                                 context:(NSManagedObjectContext *)moc;
+ (NSArray *)applicationsWithPermitNumber:(NSString *)searchName
                                  context:(NSManagedObjectContext *)moc;
```