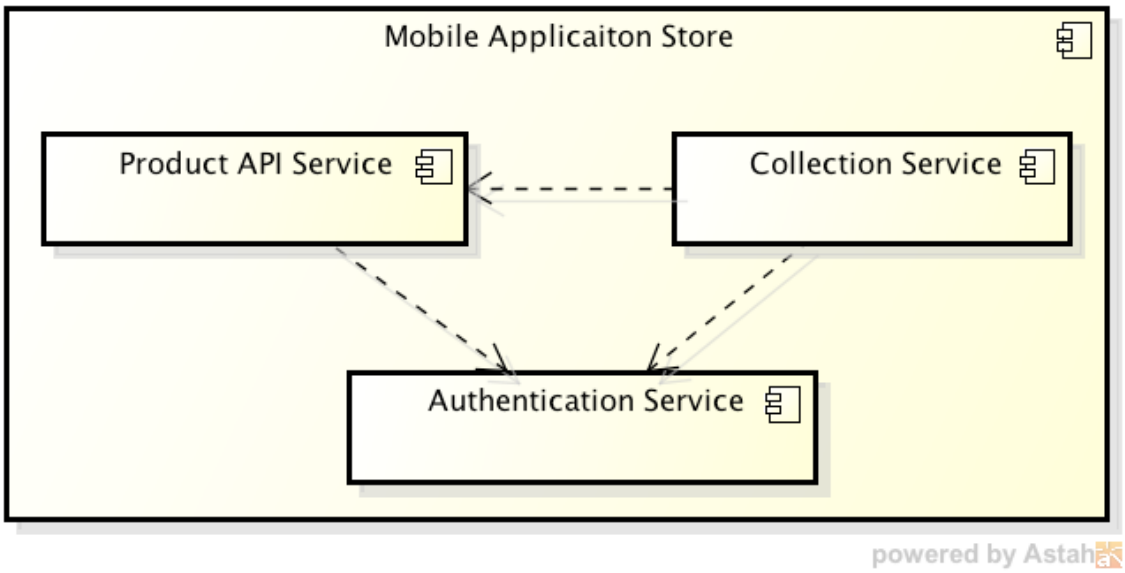
CSCI E-97
Assignment 3
Due: Wednesday, 10/30/2013

Introduction

In this assignment you will continue the development of the Mobile Application Store. You will design and implement a Collection Service in support of the Mobile Application Store.

Overview

The following diagram from the Mobile Application Store System Architecture document shows how the Collection Service fits into the overall structure.



The Collection Service supports the organization of products into Collections. The Collection Service supports two types of Collections: static and dynamic. Static collections specify the contained products as a list of products. Dynamic collections specify their contained products as Product API search criteria. As new products are added to the Product API, dynamic collections are automatically updated with any products that match their search criteria. Both static and Dynamic Collections can contain other collections. Collections and their contained products are an example of the Composite design pattern:

http://en.wikipedia.org/wiki/Composite_pattern

As part of your solution, you should include a collection Iterator. Your TestDriver class should obtain a Collection iterator from the the Collection Service and iterate through the collection

hierarchy and print out the contained collections and products. Your iterator should follow the Iterator Design Pattern discussed in lecture.

Also, apply the Factory pattern to create the static and dynamic collections.

See the Collection Service Requirements document for detailed functional requirements.

In the design portion of the assignment, you will create a software design document that satisfies the Collection Service requirements and also adheres to the System Architecture guidelines.

In the implementation portion of the assignment, implement your design and test your solution.

You will have 3 documents as input to your design:

- Collection Service Requirements Document describing the functional requirements.
- System Architecture document (from assignment 2) that provides a high level description of how the Product Catalog component will fit in the overall eCommerce system.
- Software design template (from assignment 2) as a base for your design.

Development Process

This will be the second of 3 sprints to implement the Mobile Application Store. We will continue to follow the development process outlined in assignment 2.

For this assignment, the peer design review is required. If you already have a review partner from assignment 2, you can continue to work together. Otherwise, please send a partner request (include your timezone) to ta.cscie97@gmail.com. Please request your partner no later than Monday October 14th, and then work with your partner to complete the design reviews by the following Monday, October 21st. This should provide enough time to incorporate the comments from the design review and complete the implementation before the due date, Wednesday, October 30th.

Design reviews can be conducted via:

- email
- screen sharing
- in person
- google docs
- other

Assignment Notes:

The goal of this assignment is to design and implement within the context of a collaborative agile

development environment.

Reuse the design template from assignment 2. Your design document should include the following:

- UML Use Case Diagram (with descriptions for each use case)
- UML Class Diagram
- Class Dictionary
- UML Sequence Diagram(s) (showing how the dynamic and static collection classes are integrated with the product API)

You should implement the Collection Service classes as defined by the class diagram and class dictionary specified in your design document. All collection classes should be defined within the package “cscie97.asn3.ecommerce.collection”.

Reuse your Product API classes from assignment 2. Move the Product API classes to the package: “cscie97.asn3.ecommerce.product”.

Reuse your TestDriver class from assignment 2 to load in the sample Country, Device and Product information. Modify the TestDriver to import the collection data. For each of the imported collections, use your Collection Iterator to iterate over the contents and print out the collection and product details. The new TestDriver should be placed in the package: “cscie97.asn3.test”.

When implementing your design, please document any variances from the design, provide justification for your changes and describe how your changes continue to support the requirements.

Remember to use Java doc to document all classes and methods. Add java comments inline where appropriate to explain code logic.

Sample Data

The following input files will provide data and queries for your TestDriver class to load and run.

countries.csv	country data
devices.csv	device data
products.csv	product data
queries.csv	query data
collections.csv	collection data

What To Turn In

You'll turn in a zip file containing

- Your source code (no .class files)
- Your data files
- Sample output
- Your design document (in pdf format)
- Include a document (in pdf format) describing your results:
    - Comments from peer design review and optionally the functional review
    - Any changes that you made to your design and how they continue to support the requirements
    - Did the design document make the implementation easier?
    - How could the design have been better, clearer, or made the implementation easier?
    - Did the design review help improve your design?

We should be able to unzip your file into a directory, then cd into that directory and compile your program with the command.

- javac cscie97/asn3/ecommerce/product/*.java cscie97/asn3/ecommerce/collection/*.java cscie97/asn3/test/*.java

We should be able to run your program with the command

- java -cp . cscie97.asn3.test.TestDriver countries.csv devices.csv products.csv collections.csv

where countries.csv contains the country info, devices.csv contains the device info, products.csv is a file containing a list of of products and collections.csv is a list of collection definitions.

Caution: When you believe you're done, try zipping your files, then unzipping them into a totally different directory and following the steps above. In other words, test your packaging before you submit your assignment.

Directions for submitting your solution and a grading sheet specifying the criteria for grading this assignment will be posted on the course website.