Enterprise-Scale Software Architecture (COMP5348)

Semester 1, 2013

Practical Task A – A Recommendation Page for Video Store

VideoStore wants to provide their customers with a Recommendation Page that can inform customers of media items that they might be interested in purchasing. A sample for what this recommendation page might look like, and the information it should contain, is shown in Figure 1.

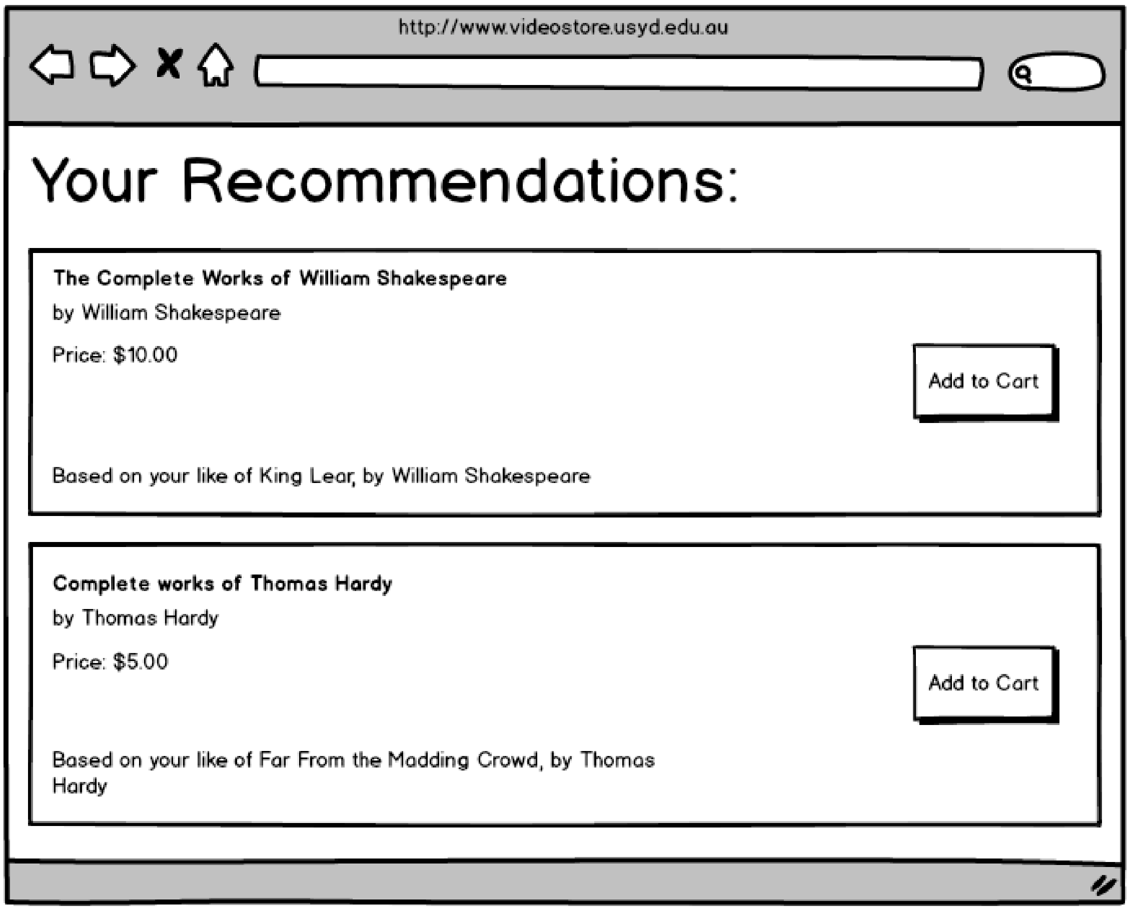
http://www.videostore.usyd.edu.au
Your Recommendations:
The Complete Works of William Shakespeare
by William Shakespeare
Price: $10.00
Add to Cart
Based on your like of King Lear, by William Shakespeare
Complete works of Thomas Hardy
by Thomas Hardy
Price: $5.00
Add to Cart
Based on your like of Far From the Madding Crowd, by Thomas Hardy

Figure 1: VideoStore Recommendation Page

As shown in the sample screen, the page should include a list of products, that are recommended based on the user's "*likes*". Each recommended item should include details about the recommended media item's title, author, price, and also why, (based on which *"liked"* product), this item was recommended. Users should also be able to quickly add recommended items to the cart.

Recommendations should be ordered in descending order of their recommendation score. An outline for how the recommendation score should be calculated, and how recommendations should be generated is described on the next page.

Creating Recommendations and Calculating the Recommendation Score:

For each media item X that the logged in user has *liked*:

- Create a recommendation for the most frequently *liked* item, (which we will call item Y), by other users who have also *liked* item X. (No recommendation is generated if no one else has liked item X).

- Record the number of users who have liked item Y, that have also liked item X. This will be known as the recommendation score for item Y.

In order to make use of *likes* to generate recommendations, you will also need to allow users to *like* items that are visible from the existing ListMedia page (accessible when you click on "Browse Media"). To do this, an additional "Like" button should be created for each item on the ListMedia page, and "*Liked"* should be shown for previously liked items, as highlighted below in Figure 2. Note that there is no need to implement *Unlike* functionality.

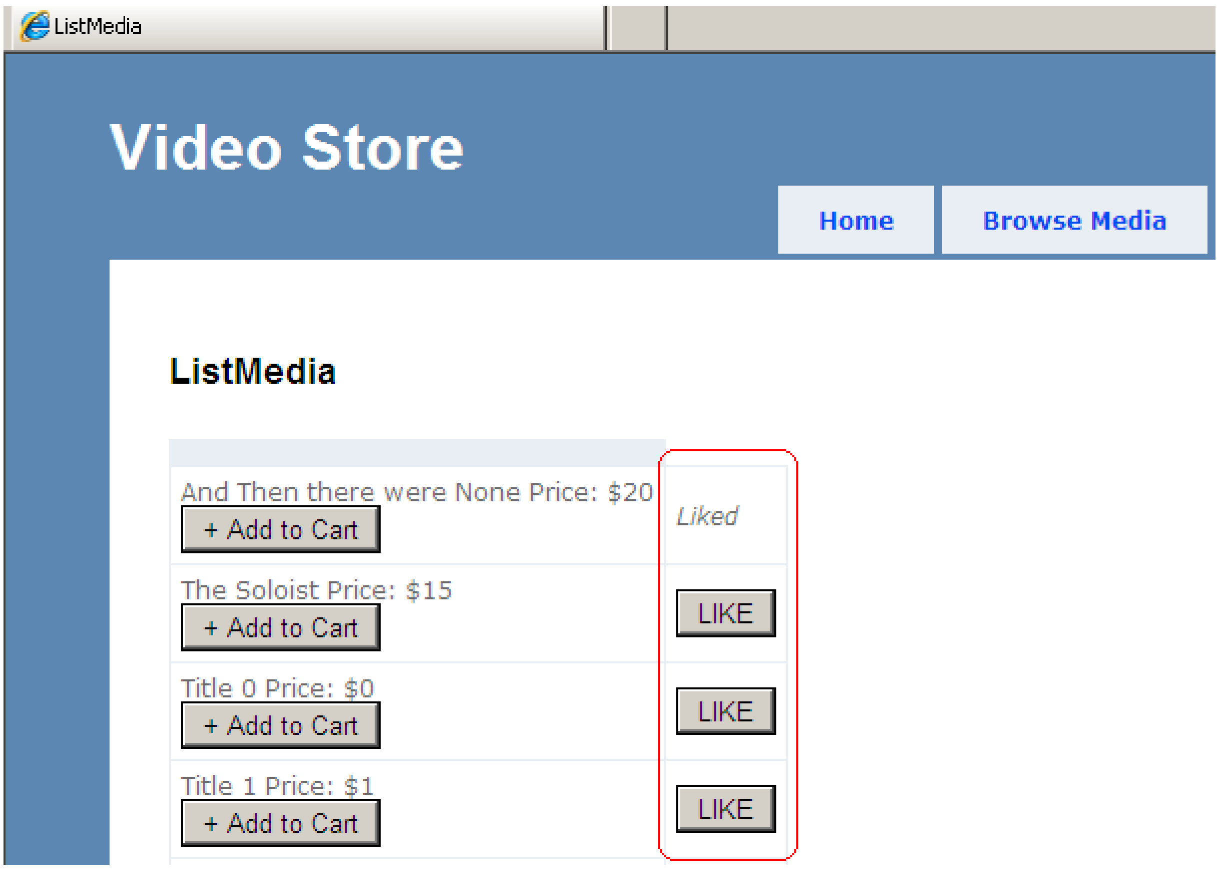

Figure 2: Addition of a like button for each media item on the ListMedia page

In this assignment, you must:

- Implement the recommendation feature, and supporting like feature, as described above, *by extending the existing VideoStore solution posted on eLearning.*

- Among other things, consider how your entity model needs to be modified, what services and business components are needed, and what MVC artefacts are needed in order to support the requirement for this assignment.

Extension (Note: Only attempt this extension AFTER your solution fulfils the requirements described above.)

For full marks, you will need to perform some performance tests on VideoStore, to answer questions 1 and 2 below, and to draw the relationship from 3:

1. What is the maximum number of concurrent requests for the browse media page before response time exceeds 1 second?

2. What is the maximum throughput for browse media requests before the system becomes saturated?

3. What does the relationship between concurrent browse media request count and browse media request throughput look like? Your answer must include a graph showing the throughput leading up to saturation.

All tests should be based on a VideoStore database that has the same state as when VideoStore is first started and the default media items are created in VideoStore.Process.Program. Be sure to describe in detail the configuration on which you ran the experiments (what sort of CPU, what speed, how much memory and cache etc, what OS version was installed, etc).

You may find the resources below useful for this extension:

- JMeter Testing Tool: http://jmeter.apache.org/

- An Introduction to Performance Counters:
  http://www.codeproject.com/Articles/8590/An-Introduction-To-Performance-Counters

- How to Create Custom Performance Monitor (PerMon) Counters:
  http://www.benday.com/2011/12/17/how-to-create-custom-performance-monitor-perfmon-counters/

- Performance By Design: Computer Capacity Planning by Example.
  Daniel A. Menascé et al

Marking Criteria

2/5 marks: The entity model is updated to include the entity types, associations, properties, and behaviour that are necessary to support the business cases.

3/5 marks: As above, and your solution is able to correctly recommend media items to users, based on their likes, as described in the business case above.

4/5 marks: As above, and the design:

- Factors components correctly within the data/business/service/presentation layers
- Uses the MVC pattern appropriately to construct a screen for recommending media items, as well as for updating the ListMedia page for *liking* media items.
- System components are built for change, where components are abstracted behind interfaces where appropriate.

5/5 marks: As above, and also answers the questions, and completes the task described for the assignment extension.

Submission Instructions

1. You should submit your solution and a readme.txt file together in a zip file on eLearning. The zip file should include your unikey in the file name. For example, if your unikey is abcd1234, you would label your zip file as abcd1234PracA.zip
2. The readme.txt file should provide an outline of:
    a. The components you modified or introduced into the VideoStore solution. Make sure you clearly identify where these modifications or additions can be found. Please include comments in the modified or added files to indicate your contribution.
    b. How to run your solution in one of the SIT labs 115, 116, or 117.
3. You must submit your solution by Friday 9 pm in Week 5. Late submissions will not be marked. A submission link will be provided on eLearning.
4. Submit a hard copy assignment cover sheet with your signature to your tutor in Week 5. You can download the cover sheet from: http://sydney.edu.au/engineering/it/current_students/undergrad/policies/assignment_sheet_individual.pdf
5. In Week 5 you'll be asked to run your application and show correctly produced

recommendations. If you’ve also implemented the extension for this assignment, you MUST advise your tutor and demonstrate this.