

Fall 2008 (*Assigned 11/24/2008*)  
COSC 355: Software Engineering I  
Professor M. Brian Blake  
Midterm Take-home Exam  
**Due 12/8/2008 12:00 am**

In completing this exam, you will exercise your full-range of knowledge with respect to managing the entire software engineering lifecycle. Below is a problem statement for a new software application for which you will need to provide analysis and design details. During this course, you have learned how to take a general description of a problem and show potential high-level functionality of the solution application using *scenarios* and *use cases*. Subsequently, as a software engineering student, you have developed a structural picture of the target software application using a *class diagram*. Also, using *sequence diagrams* and *state diagrams*, you have shown how the system works behaviorally. Finally, in managing the development project, you have to set up a *configuration management repository structure* that can be used for the software development environment. So for this exam, you must use the provided problem statement to create the following deliverables:

1. Create **3 scenarios** (numbered lists) that show how the major methods in which the system will be used (**10 points**)
2. Create a **use case diagram** that details the high-level functionality of the system (**20 points**).
3. Create a **class diagram** based on your problem statement and indirectly your use case diagram (**30 points**).
  - a. (Hint: Make sure to have *surrogate classes* for reviews and database tables. Also you should have *interface classes* for the database connection to your relational database and to the external sophisticated review extraction software, both detailed in the problem statement)
  - b. Use stereotypes to delineate the type of classes
4. Create **3 sequence diagrams** that show how your scenarios in (1) are implemented in your software (**20 points**).
5. Choose the controller class (should be obvious based on your sequence diagrams) in your class diagram and create a **state diagram** based on this class (10 points)
6. Show the **directory structure** of the configuration management environment that includes all types of artifacts that you envision that you will need during development. (10 points)

Grades will be based on how completely you address the problem statement, how integrated your models are, correctness in creating your models, and your adherence to scalability, reuse, and modularity. Please embed all of these diagrams into one document, if you don't combine them all, it will cost you points. Feel free to handwrite if you choose.

***THIS IS AN INDEPENDENT TEST. NO COLLABORATION IS ALLOWED!!***

## **Problem Statement: *Reviews-R-Us***

It is a common occurrence that on-line shoppers look for reviews on products before they make their purchase. Depending on the quality of their query on various search engines such as Google and Yahoo, they get any range of reviews. Reviews can be old or out-dated, since reviews tend to change over-time. Users do not always formulate their search appropriately such that they also get the competing products. As such, the Reviews-R-Us (RRU) system alleviates those concerns. This system will maintain a repository of reviews based on a list of products that are commonly researched over the Internet. RRU will contain an internal database that stores and manages reviews of many products. Users will be able to connect to the RRU web portal and submit queries based on a specified product name or be able to browse through categories of products provided by RRU. The RRU system is split into 3 major components, a *web user interface module*, a *review discovery module*, a *data management module*.

The web user interface module will allow a user to log into the system. This interface will show the user a list of his/her past queries. The user should either be able to enter product name into the system or have the ability to browse through a list of categories and pick a specific product. The web user interface would present information about products from the data management module.

The data management module is an interface to the RRU's internal relational database. The internal database should have objects that represent individual users, products, and reviews. The user objects should be connected to the reviews that he/she visited in the past. The data management module should allow the review discovery module or a human administrator to add, delete, or modify reviews. Reviews should be detailed by the date that the review was created by the reviewer, the date that the system captured the review, the text of the review, and overall rating of the review (i.e. Excellent, Very Good, Average, Bad, and Very Bad). The products should be connected to their reviews.

The review discovery module is a process within the RRU that runs on the server for which the RRU is present. This process uses continual queries via Google's search engine to capture reviews. The review discovery will be seeded by user requests for products. In addition, the system will use reviews that reside on the same page as the discovered reviews to search for reviews on other products. The system will use an external sophisticated mechanism for extracting review text and the date of the review from an HTML page.

The RRU will respond to a users request within 15 seconds of pressing the submit button. In addition, the review discovery system must be able to remove reviews after a certain time period as prescribed by the administrator.

