CSE 460

Software Analysis and Design

Programming Assignment

Spring 2019

Assigned Date: March 22, 2019
Due Date: April 17, 2019

Introduction

yaBe is a *fictional Customer-to-Customer* (C2C) online marketplace where people sell their merchandises directly to other people. Sellers may post new individual items for sale. Buyers can view the list of the posted items and make purchases. Since good deals come and go all the time, popular demand from yaBe buyers is the ability to subscribe to certain keywords and receive notifications after any listing containing the keywords that the buyer is interested in becomes available.

Project Description & Requirements

You are tasked to design and implement the above C2C software system using the publish/subscribe design pattern. The following descriptions cover a *minimal* set of characteristics and functionalities for the C2C software system. You may expand upon this set if you believe it would be beneficial to do so.

- An item listing (a.k.a. a post) contains a title, a price, an ID for the seller, and optionally a description for the item. The title is a set of concatenated keywords with whitespaces separating them. A keyword is defined as a string of at least 3 alphanumeric characters (`A-Z`, `a-z`, `0-9`), underscores (_), dashes (`-`), and no whitespaces. For example, "Banana", "One", "oNE", "2019" and "TRUTH" are legitimate keywords. To keep things simple, regular expressions and logical operators will not be supported nor tested.
- Seller ID must be unique. Price for every item is a finite positive~~,~~ ~~natural~~ real number. One item can be posted at a time.
- A seller may modify any of his posted items. Only the price and description can be changed. The modified post should be considered a new listing for the purpose of notification.
- If a seller lists an item with a title that contains one or more keywords that a buyer have subscribed to, then the buyer will receive a notification for that item. The seller who posts items will not be notified (i.e., a seller of an item cannot be the buyer of the item he is selling). The buyer will receive one notification for every posted item that has any of the keywords that match their subscription. Sellers do not know who the potential buyers of their items are.
- A buyer is able to subscribe to one or more keywords, however, only one keyword can be subscribed to at a time.
- The buyer can undo any of his subscriptions. After a buyer unsubscribes from keywords, he will no longer receive notifications for the posts that have the unsubscribed keywords (hint: also

consider the case where the buyer attempts to unsubscribe from the keywords he has *not* subscribed to).

Bonus:

Certain merchandises may not be listed using this system. Assume all such items are unique. The list of prohibited items (keywords) is known a-priori to the system (see separate Blackboard attachment `ProhibitedItems.txt`).This list is fixed. All posts that include any prohibited keywords will be silently ignored (i.e., the seller is not notified). If a buyer attempts to subscribe to any such keywords, his request will be ignored too (i.e., the buyer is not notified).

Input/output Format

Apart from your main software implementation, your submission should also be able to interpret **command line queries** and provide **console outputs** to these command line queries. See below for a minimum set of queries required. The order and the parameters of the queries will be random, and in no circumstance should your program crashes or throw exceptions. The software does not output debug messages to console.

The following four queries are required and will be tested:

```
publish listing,[keywords],[seller],[price],[description]
publish listing,[keywords],[seller],[new price],[new description]
subscribe,[buyer],[keyword]
unsubscribe,[buyer],[keyword]
```

The square brackets are not included in the input and should not be in the output; they are included to help with reading of the entries in input files. The entries are separated by a newline character (‘`\n`’), and there is no space before or after commas (‘`,`’). There will be no comma, newline, or semicolon (‘`;`’) in the parameters surrounded by the brackets.

Input parameters can contain any alphanumeric characters, underscores, dashes, and no white spaces. Your program should be able to handle uppercase and lowercase seller and buyer names, keywords, and listing titles and treat them equally. For example, a seller named “`totallyNotRobots`” should be treated as the same person as “`TOTALLYNOTROBOTS`”.

The only output that this C2C software system should produce are the notifications sent to buyers. The format for such notifications is

```
Sent to [buyer]: A new listing named [title] was just added to
yaBe by [seller]! The item matches your keyword(s) [keyword(s)].
It is for sale at $[price].
```

The output should be a verbatim copy of the above format with the placeholders substituted with data contained in subscribe and publish requests. Should a listing matches multiple keywords, they should be arranged in the order that buyer subscribed to the keywords, with a comma and a whitespace separating each. See samples.

Design, Coding, and Testing

To get started, it is recommended that you review Publisher-Subscriber pattern and related topics. Then, utilize Astah to come up with a few class diagrams and sequence diagrams. You need to use *forward engineering* to generate a skeleton code that reflects your design. Finally, add your code and logic *on top* of the stubs generated by Astah. Manually added code must be identified using **// Begin** and **// End** comments. The UML diagrams must completely match your code to the degree possible. Do not delete or change any of the Java code produced by Astah. If you need to change any line of code from Astah, such as for a return statement, leave a commented out copy of the original line in the program. ~~The UML diagrams should match your implementation as much as humanly possible. If you must alter any generated code, comment them out instead of deleting them.~~

Your program will be tested with text inputs. You can find few sample inputs/outputs in a separate zipped file in Blackboard (when they become available). The actual test cases can differ from the samples and can have different querying orders. We will use System Rules to simulate querying on and reading from console. The texts will be stripped (remove leading & trailing whitespaces) and converted to lower case before comparing, however missing a whitespace or having an extra newline character in the middle of the output will fail automated test cases. Therefore, it is for your best interest to follow the instructions carefully and match your output with the format shown above.

Submission & FYIs

- Your code should target **Java SE 11**. Cannot use Java EE.
- Your submission will be automatically compiled and tested on Windows 10. Make sure your program does not rely on system-specific features or environment.
- **Make sure that the driver class that starts the system is called `Main`.** Failure to do so will result in compilation failures.
- Make sure your submission compiles and runs from command line.
- Use of external APIs and libraries is not allowed. You must stick with strictly "vanilla" libraries and packages.
- You are not allowed to copy code written by others. You are not allowed to copy or reuse other students', current or previous, submission. You may, however, study existing publisher-subscriber implementations and come up with your own idea. Remember to give credit where it's due.
- Check out the samples provided in Blackboard to get a better understanding of the software system.

Your submission should be a single zip (`PostingID.zip`) file containing

- A report (`PostingID.docx` or `PostingID.pdf`). The report must be written using the template available in Blackboard.
- Java source files (`*.java`).
- Astah project file (`PostingID.asta`).

Turn in hard copy of the report to class one due date. Turn in the zip file to Blackboard.

Note: your report on paper and the Java files are the primary source of evaluation. Astah file will be used should disputes arise.

Rubric

Grading Category	Grading Item	Points
Design	Class diagrams	10
	Sequence diagrams	10
	Publisher-subscriber (design)	10
	Overall quality	15
Code	Publisher-subscriber (implementation)	10
	Correct output	20
	Forward engineering	5
	Code quality	5
Documentation	Overall effort in documentation (including comments in code and description in your report)	15
Extra Credit	Correct implementation of the Bonus question	10
Total		110