

Feb 19, 2024

Assignment – 2
[100 Points]

CSC-413-02
Spring 2024

San Francisco State University
Computer Science Department

Assignment Goal

In this assignment, you will be using the SRS (System Requirements Specification) document and do the following:

- A. Creating the core business objects, based on the discussion in class around requirements modeling
- B. Identifying and creating the core attributes for each of the core business objects identified in the previous steps
- C. Identifying the base functionality points(methods) stubs for each of the business objects identified. Consider the following for each object of the application as determined from the SRS document:
 - (i) Constructors
 - (ii) Getters and setters
- D. The objects should be stored in the PriorityQueue data structure. The priority logic of the queue should be determined by the following, in that order:
 - (i) Date of object creation
 - (ii) The amount of balance in the account. The higher the balance, the higher the priority

Note:

 - (a) The data structure should be implemented as a LinkedList
 - (b) You will therefore have to create all class(es) in support of the data structure
 - (c) Feel free to reuse your Queue data structure code from your Data Structure class assignments
- E. Based on the requirements outlined in the SRS document, you will need to identify the relationship between all business object entities.

For example, we know that a customer object may have 1 or more accounts. Therefore, you will need to make sure the account attribute for a customer object is a collection, that will allow you to store all account objects associated with that customer

Conversely, since an account cannot exist without an associated customer, the creation process of an account must be provided with the customer object to whom it is related

- F. Write Junit Test cases for each of the business object(s) and test for all functionality points defined
- G. You will need to define the method stubs for each business object to fetch its data, as well as

data for all its associations

Note:

- (a) For now, in this exercise, we're creating just the base objects and their cardinality associations
- (b) In subsequent exercises we will be wrapping the base objects around a known pattern, as we begin to formalize the application architectural components, in an iterative manner

H. Other considerations to keep in mind:

- (i) Since the business object, with their requisite data will be stored in PriorityQueue data structure, you will need to ensure the queue implements all methods as defined in QueueInterface. Please refer to topics covered in your Data Structures class
- (ii) In having to use the PriorityQueue, your class will need to implement the Comparable interface, which will provide means to keep all objects that you retrieve from data store or create anew, are in the order of their priority

I. Simple JSON Exercise

- (i) Create test business objects
- (ii) Use the ObjectMapper to translate the object to JSON string and vice versa, similar to the following format.

```
ObjectMapper om = new ObjectMapper();
String jsonString = "{\"name\":\"John Doe\",\"email\":\"jdoe@sfsu.edu\"}";
YOUR_OBJECT obj = om.readValue(jsonString, YOUR_OBJECT.class);
```

Your Output should look like follows when you print your object
 //User object with name 'John Doe' and email jdoe@sfsu.edu '

Note:

- (a) You will need to add Jackson Library to your Java test project. One way to do that, if you are using Maven, would be to setup dependency as follows in your pom.xml
- (b) Alternatively, you should be able to set this up directly from your IDE, by adding the Jackson library jars(annotation, databind, annotations) from the URL:
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core>