

Feb 26, 2024

Assignment – 2
[200 Points]

CSC-413-02
Spring 2024

San Francisco State University
Computer Science Department

Assignment Goal

Using the SRS (System Requirements Specification) document and start identifying and creating business objects. This assignment initiates the process steps of application software development, will continue on through the course of the semester, which will mandate modeling, implementing and testing additional components

To keep in line with the real world, some requirements as stated here may/will be changed/updated during the course of application's Software Development Life Cycle(SDLC) process through the course of this class in the semester

1. Creating the core business objects, based on the discussion in class around requirements modeling and the SRS document provided (20 pts)
2. Identify and create the core attributes for each of the core business objects identified in the previous steps (20 pts)
3. Identify 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:
 - (a) Constructors (10 pts)
 - (b) Getters and setters (10 pts)
4. 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 (40 pts)
 - (a) Date of object creation
 - (b) The amount of balance in the account. The higher the balance, the higher the priority

Note:

- (i) The data structure should be implemented as a LinkedList
 - (ii) You will therefore have to create all class(es) in support of the data structure
 - (iii) Feel free to reuse your Queue data structure code from your Data Structure class assignments
 - (iv) Sample PriorityQueue and QueueInterface will be provided in Github as the starter code that you can use
5. Other considerations to keep in mind:
 - (a) 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 for any c

- (b) In having to use the PriorityQueue, your queue class will need to implement the Comparable interface, which will ensure that all objects that you retrieve from data store or create anew, are added to the queue in the order of their priority, as spoken off in 4 above (10 pts)
6. Based on the requirements outlined in the SRS document, you will need to identify the relationship between all business object entities. (20 pts)
- 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
7. Write Junit Test cases for each of the business object(s) and test for all functionality points defined (20 pts)
8. You will need to define the method stubs for each business object to fetch its data, as well as data for all its associations (20 pts)

Note:

- (i) For now, in this exercise, we're creating just the base objects and their cardinality associations
 - (ii) 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
9. Simple JSON Exercise (20 points)
- (a) Create test business objects using JSON, as discussed in class. You may use the sample code for JSON as the starter code base
 - (b) 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);

```

```

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

```

Note:

- (i) 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`
- (ii) Alternatively, you should be able to set this up directly from your IDE, by adding the Jackson library jars(annotation, databind, core) from the URL:
<https://mvnrepository.com/artifact/com.fasterxml.jackson.core>

10. Creating DTO(Data Transfer Objects) DAO(Data Access Objects) for your business objects in the Banking Application. Please refer to the in-class discussion on DTO/DAO Objects. You may use the sample code for DTO/DAO as the starter code base (30 pts)

- (a) Create a DTO for each of your business objects
- (b) Create DAO Objects for each of your business objects
- (c) Define constructor and the access methods for each
- (d) Create a method stub for the `get(id)` method for now. Additional methods will be added later