Cairo University **Faculty of Computers and Artificial Intelligence**

Computer Science Department – Software Engineering Undergrad Program

CS 355: Software Design and Architecture, MIDTERM EXAMINATION

Instructor: Dr. Soha Makady

Fall 2020 – CLOSED Book Exam –Total marks: 20 – Duration: 60 mins

This exam comes in four pages

STUDENT NAME ______________________________________

STUDENT ID # _____________________

NOTE: This sheets functions as a **key to the answers not an exact model answer**.

Question 1 [6 marks]

a) Discuss a typical design tradeoff between two non-functional requirements. **[2]**

Functionality vs usability. Is a system with 100 functions usable? Adding more functionalities could lead to reduced usability as the system will have too many functionalities.

b) Provide an example for one decision that would be classified as an architectural design decision, and another that would be classified as a non-architectural design decision **[2].**

The selection of a data structure, and the algorithms to manage and access that data structure is a non-architectural design decision.

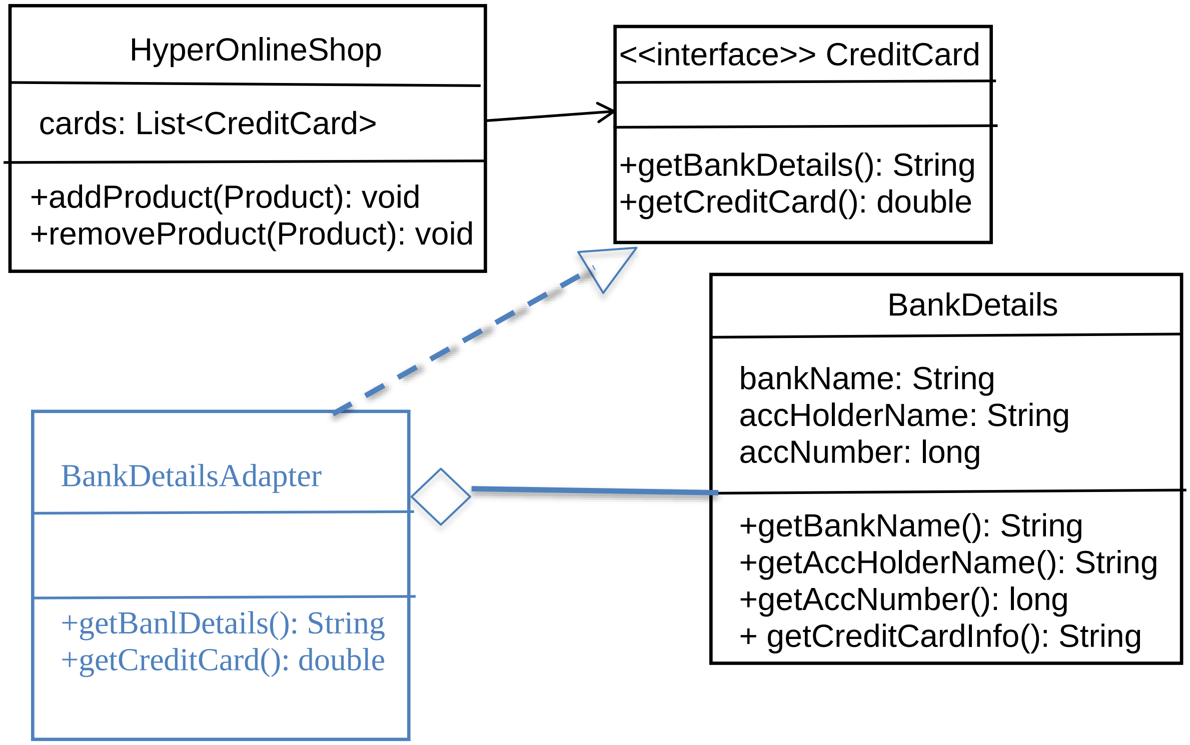
Reasoning about the components to include is an example of an architectural design decision (e.g., to support availability, you could include multiple copies of one component).

c) Consider a news update service that is provided by Al-Shorouk online newspaper website. Readers can subscribe to that service to receive news updates as soon as those news get published on the website. Give an example to illustrate the difference between the push and pull modes of the observer design pattern, if the observer pattern is applied within such website. *You don't need to draw a class diagram of the example if you can explain the solution in writing.* Make any additional needed assumptions (if any). **[2]**

The push mode implies enforcing a specific selective set of information to be given to the observer as updates, whereas the pull mode implies sending the observed item itself as an update. Hence, for the newspaper, the push mode implies sending the sports news updates only for instance. The pull mode would be sending a news object that includes the updates in all news categories.

Question 2 [4 marks]

Assume that you have a HyperOnlineShop which maintains the information of its registered customers, who use their online shop, as a list of CreditCard. Later on, HyperOnlineShop acquired another store (Tasawwaq). However, Tasawwaq maintains the list of its registered customers as BankDetails. We now want to add those Tasawwaq customers to our HyperOnlineShop.



a) **[2 marks]** **Apply** the adapter pattern to allow adding those Tasawwaq customers' details. You need to **sketch** a class diagram and to **explain** it.

The above updated diagram should be sketched along with a written explanation justifying the need for such updates.

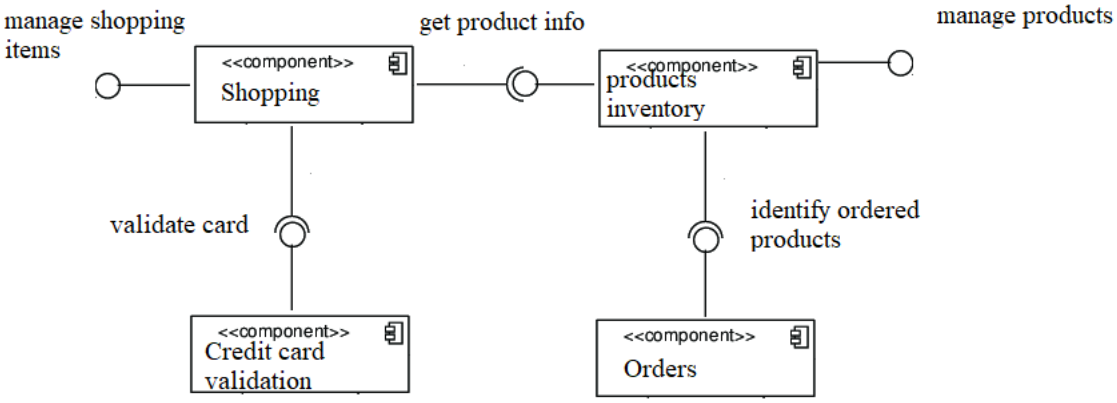
b) **[2 marks]** We discussed two possible classifications for design patterns. **Classify** the applied adapter pattern **two times**: one time according to each classification. For each classification, **explain** why you made that specific classification for the applied pattern.

The adapted pattern applied above is an object-level pattern as it uses composition to satisfy the needed interface (CreditCard) within the target system.
The adapter pattern is classified as a structural pattern as it deals with the composition of classes and how the current structure of the BankDetails class does not match the needs of the target application.

Question 3 [4 marks]

Souq is an online shopping website. Souq allows **shoppers** to search for products or browse existing products. However, for the searching feature to work properly, the searching needs to retrieve products information from an inventory that holds all the information of the different products. Shoppers are also allowed to add items to their shopping cart. For the shopping process to succeed, the shopping process demands validating the shoppers' credit cards through a third-party (ready-made component). Souq **employees** should be able to add/remove products to the inventory, to list the available pieces from each product, to update product information, and to identify which products are part of existing customers' orders. To identify the products that are parts of existing orders, the identification needs to retrieve orders information from an orders repository that holds all the current shoppers orders.

a) **[2 marks]** Create a component diagram showing the main components of the above system. Your diagram should show the components, and the component interfaces as well using the UML notations.



b) **[2 marks]** For **one component** from part (a), you need to **mention** the services that it provides and **provide** their corresponding APIs.

For the component “products inventory”, its services are: add product, remove product, and update product info

The API for such services would be:
Boolean addProduct(String productName, int productID, double productPrice, Producer productProducer)
Boolean removeProduct(int productID)
Boolean updateProductInfo(int productID, Object updatedInfo)

Question 4 [6 marks]

Consider a graphics library that allows creating images from those types: GIF, JPEG, and BMP. The library allows its user to traverse the created images’ pixels in two different traversal mechanisms: horizontally, and vertically. The traversal needs to allow the user to get the current pixel, get the next pixel, and inform the user when there are no more pixels left. Once an image is created, the library allows users to display the created images in three layouts: stretched layout, squeezed layout, and image-with-filters layout. **Sketch one** class diagram to model that library after applying the needed design pattern(s). The diagram should clearly show the attributes/operations/relationships as needed. **Justify** why you selected each pattern(s).

The solution will involve three patterns including one consolidated class diagram showing all the patterns.

- Iterator pattern to allow iterating using different traversal mechanisms (horizontal traversal iterator, and vertical traversal iterator). Each iterator would have the methods getCurrentPixel(), getNextPixel() and boolean hasMorePixels()).

- Factory method pattern to allow creating images from the three mentioned image types

- Strategy patterns to allow different algorithms for displaying the images (i.e., stretched, squeezed, image-with-filters).

- A partial class diagram is shown for applying the strategy pattern.

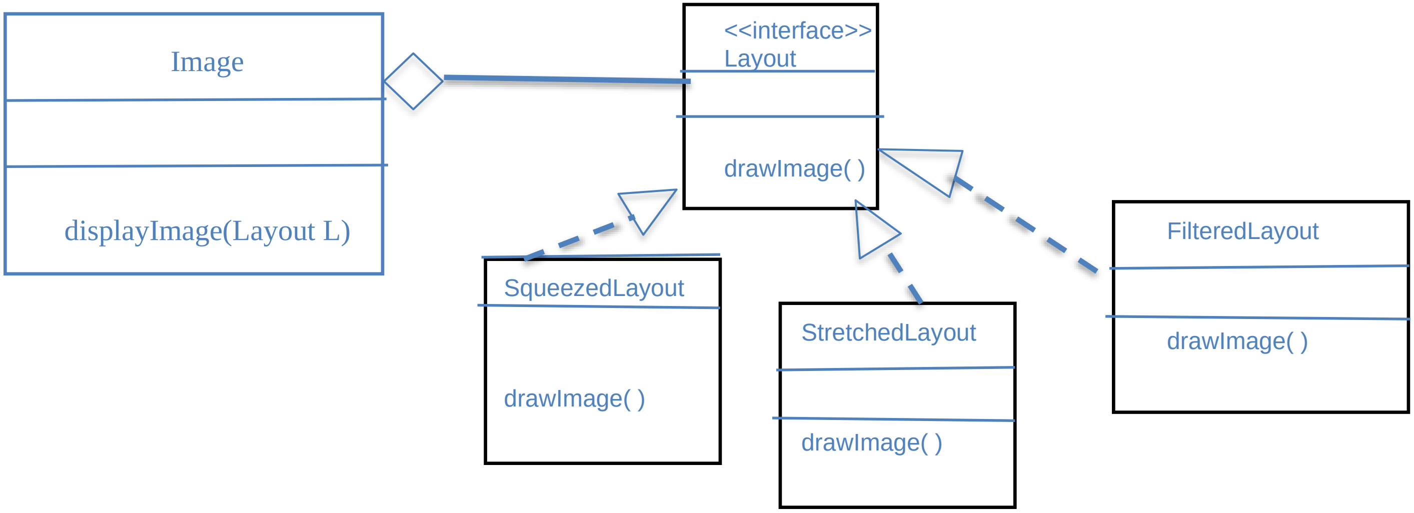


Image
displayImage(Layout L)
<<interface>>
Layout
drawImage( )
SqueezedLayout
drawImage( )
StretchedLayout
drawImage( )
FilteredLayout
drawImage( )