Exercise #2

Group GOLF:

Anish Girija Sivarajan, Md. Shohel Ahamad,

Slobodan Kocevski, Veselin Stefanov

06.12.2016

Exercise 1: HTTP

a. Is a HTTP-based communication between a client and a server synchronous or asynchronous? Explain (1 point)

b. Is the HypertextTransferProtocol stateless or stateful? Please explain. (1 point)

Answer:

a.

HTTP is synchronous by nature because a client sends a request to a server and does nothing until he gets his corresppponding response.

An example for asynchronous communication is AJAX (Asynchronous Javascript And XML). It is used by browsers to send a HTTP request, using JavaScript, to load data without leaving/reloading the current page. When the corresponding response arrives a handler function is called to deal with it.

b.

HTTP is a stateless protocol. Byte streams are used to deliver data. Stateless means that every request-response pair is independent from any other request-response pair. HTTP does not require a server to store information about the HTTP request. Every request carries all the necessary information for the server to create a response.

Exercise 2: JavaScript

a) How can external JavaScript files be included into HTML? (1 point)

b) Where in a HTML page can JavaScript code be embedded? (1 point)

Answer:

a.

External JavaScript files can be included in head or body sections of a html document by using the **script** tag and its **src** attribute: <script type =" text / javascript " src =" jsfile .js" ></ script > In this case the **jsfile.js** file is included.

b.

JavaScript can be embedded either in **head** or **body** sections of an html page. It is recommended to be consistent with the place where we put our javascript – either only in **head** element or only in **body** element.

The script itself is embedded using the **script** element:

<script type ="text / javascript">
... //here is the JavaScript
</ script >

Exercise 3: Ajax (2 points)
The following figure presents the classic web application model and the Ajax application model.

1. Considering this figure, please explain how Ajax differentiates from classic web application models from:
    a. The technical point of view.
    b. The user point of view.

2. Please provide an example of a web application that uses Ajax.

Answer:

1a.

In the classical web application model the browser (the client) interprests the HTML and CSS data which is being sent by a web server. Using AJAX the client can do some of the data processing that is usually done by the server. With AJAX the server sends XML data. XML is more flexible since it can transport more information from the server in a specific way. On the other hand HTML and CSS only contain visible content. XML data can not be directly

interpreted by the browser. AJAX engine is needed to process this data. The output of this AJAX engine is generated HTML and CSS.

1b.

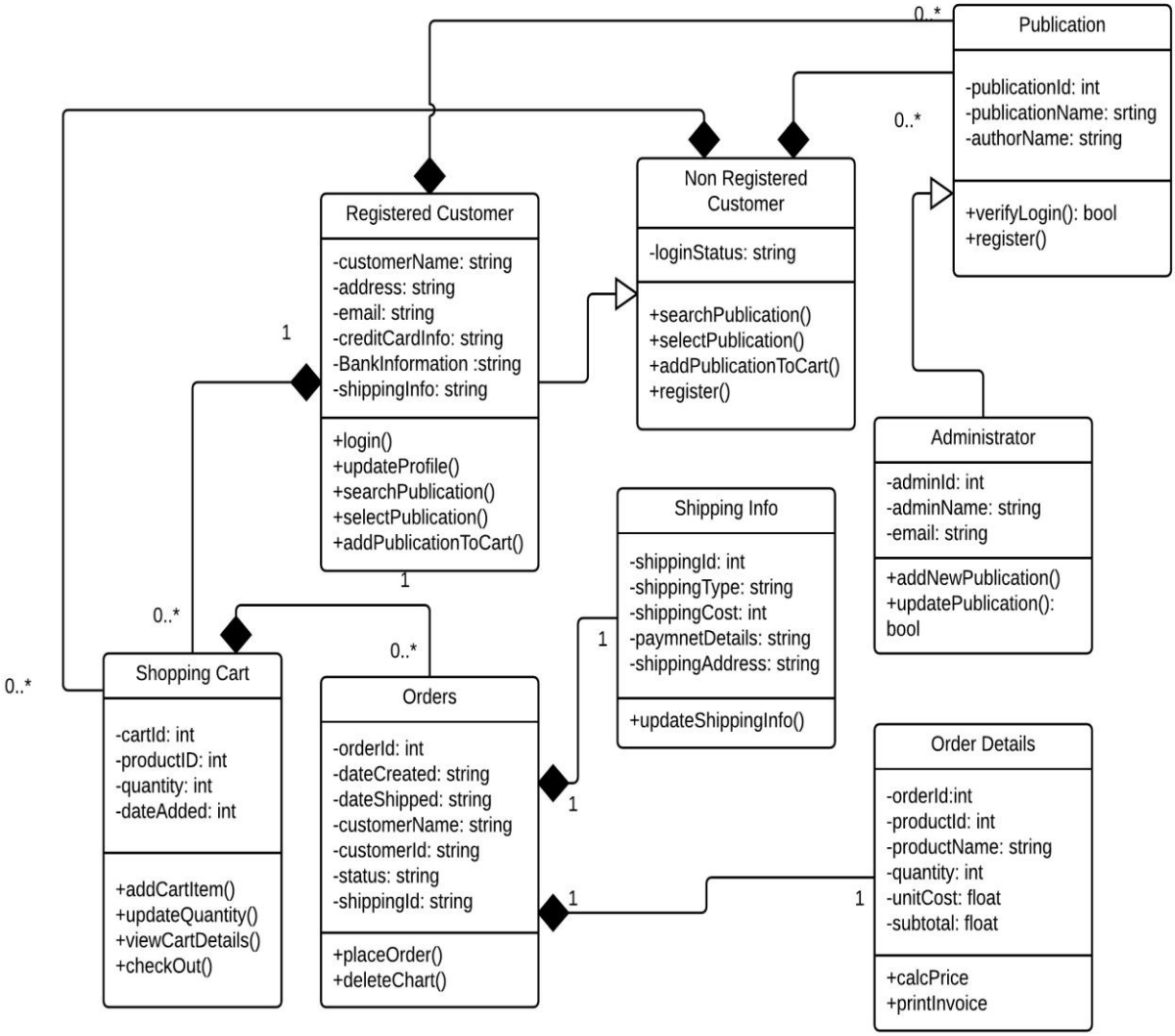
In order the user to be able to use AJAX he or she needs to enable JavaScript on their browser. AJAX is different because users can get new chunks of content without reloading the page. The user doesn't need to wait for a response from the server.

2.

Google Maps is an example for a web application that uses AJAX. Users can view new territories' data without having to reload the page.

Exercise 4: UML Modelling

Answer:


Publication
-publicationId: int
-publicationName: srting
-authorName: string
+verifyLogin(): bool
+register()
0..*
0..*
Registered Customer
-customerName: string
-address: string
-email: string
-creditCardInfo: string
-BankInformation :string
-shippingInfo: string
+login()
+updateProfile()
+searchPublication()
+selectPublication()
+addPublicationToCart()
1
1
Non Registered Customer
-loginStatus: string
+searchPublication()
+selectPublication()
+addPublicationToCart()
+register()
Administrator
-adminId: int
-adminName: string
-email: string
+addNewPublication()
+updatePublication(): bool
Shipping Info
-shippingId: int
-shippingType: string
-shippingCost: int
-paymnetDetails: string
-shippingAddress: string
+updateShippingInfo()
1
0..*
0..*
Shopping Cart
-cartId: int
-productID: int
-quantity: int
-dateAdded: int
+addCartItem()
+updateQuantity()
+viewCartDetails()
+checkOut()
0..*
Orders
-orderId: int
-dateCreated: string
-dateShipped: string
-customerName: string
-customerId: string
-status: string
-shippingId: string
+placeOrder()
+deleteChart()
1
1
1
Order Details
-orderId:int
-productId: int
-productName: string
-quantity: int
-unitCost: float
-subtotal: float
+calcPrice
+printInvoice