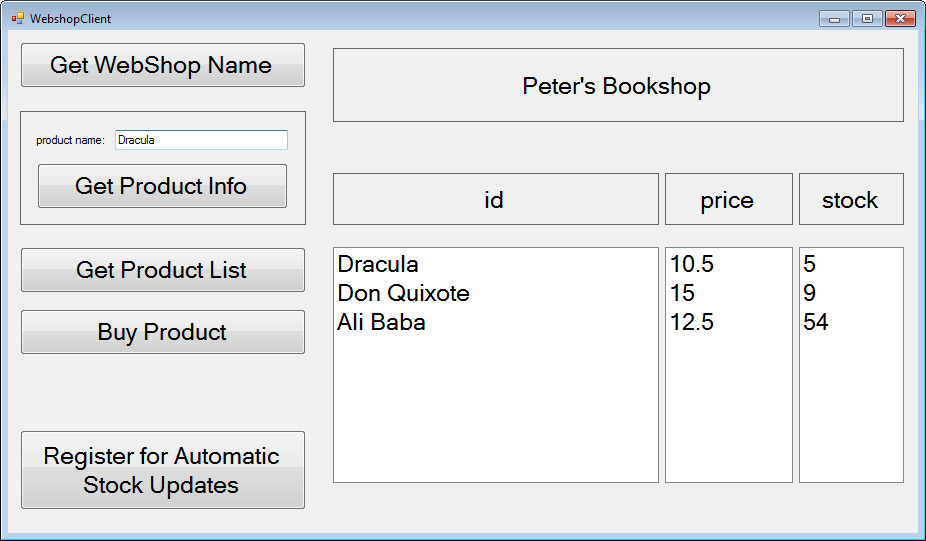
**CSA Mock Exam**

Introduction

In this exam you are going to implement a webshop client-service system. The core of the system is the WebShop service, which sells books. The webshop has a name (string, e.g., “Peter's Bookshop”) and it stores a list of products. Each product has:

1. a unique name (string, e.g., “Dracula”),
2. description (string, e.g., “This is a very scary book.”)
3. price (decimal, e.g., 10.5 euro),
4. profit margin (double, e.g., 12.6 %), and
5. stock (int, e.g., 5 pieces).



WebShop

Figure 1

Download and unzip the start Visual Studio solution. This solution already contains two different Client projects with each one a Form. You should at least add a new project for the service to this solution.

Exam points:

\* If submitted solution contains compiler errors, your end grade will be 1.

\* If submitted solution does not contain compiler errors:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Assignment | 1 | 2 | 3 | 4 | 5 | 6 | Bonus | Total |
| Points | 1 | 1 | 1.5 | 1.5 | 2 | 2 | 10 | 10 |

In this exam you are free do decide which type of endpoint configuration you will use (programmatic or administrational).

**Exercise 1 [1 points]**

By clicking on the “**Get WebShop Name**” button, the Client should retrieve the name of the WebShop and show the name in label lblNotifications.

**Exercise 2 [1 points]**

Add several ‘hard coded’ books to the list of products at the startup of the WebShop service.

By typing a product name in the tbProductName and clicking on the “**Get Product Info**” button, the Client should retrieve the description of the selected product from the WebShop. Show the retrieved description in label lblNotifications.

**Exercise 3 [1.5 points]**

By clicking on the “**Get Product List**” the Client should retrieve a list of all products from the WebShop.

Note that for each product only the name, price and stock information should be given to Clients. The profit margin and description is hidden from Clients. The client should show the product information as shown in Figure 1.

**Exercise 4 [1.5 points]**

By selecting one product name in the list and clicking on the “**Buy Product**” button, the Client should buy one piece of the selected product in the WebShop. Add a boolean method to the WebShop for buying a product for a given product name.

* The WebShop should always check if the product is on stock. If the product is in stock, the WebShop should decrease the stock of the product and return value true. If the product is not on stock, the WebShop should just return value false.
* The client should show whether buying was successful or not in label lblNotifications. In case buying was successful, the client should decrease the stock for this product shown on the form. For example, after successfully buying book “Dracula”, the Client should change the first item in the stock list to 4.

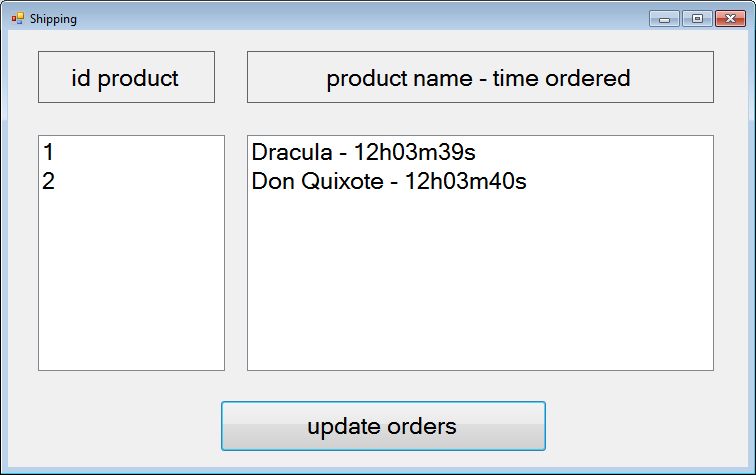
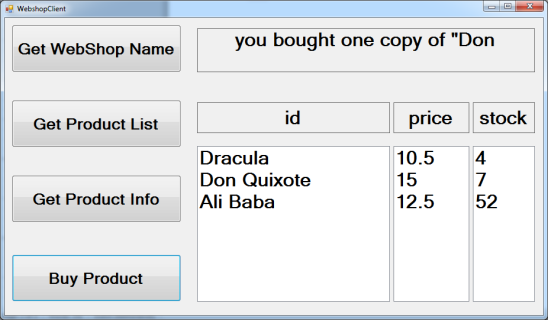
**Exercise 5 [2 points]**

The WebShop should work with multiple clients at the same time. If one client buys a product, the WebService should automatically notify other connected Clients about the new stock value for this product. Upon such a notification all notified Clients should adjust the stock value for that product’s stock (and show the new stock value on their Form). At the startup of the client application the client should automatically register for receiving automatic notifications of stock.

**Exercise 6 [2 points]**

Add a new business endpoint to the WebShop: the Backoffice endpoint will be used by the Backoffice Client to connect to the service in order to ship the orders. Extend the buy product method in the WebShop by creating a new Order whenever a product is bought. The order should have a unique id (number, e.g., 1), product name, and time when the order is placed.

By clicking on the “**update orders**” button, the Backend Client should retrieve all orders from the WebShop and show it on it Form as shown in Figure 2.



Client Backend client

WebShop

Figure 2