Testing Report

UMBC Bazaar

**Client**

Ericka Marable

**Team 3: UMBC Obsidian**

Gene Burchette

Jerson Guansing

Karl Justice

Darren Stevens

Victor Wu

***UMBC Bazaar***

**Testing Report**

#### Table of Contents

Page

## Introduction 3

1.1 Purpose of This Document 3

1.2 References 3

1. Testing Process 3

2.1 Description 3

2.2 Testing Sessions 3-5

2.3 Impressions of the Process 5

1. Test Results 5

3.1 Testing Suite 6-10

3.2 Test Results 10-34

1. Appendix A – Team Review Sign-off 35
2. Appendix B – Document Contributions 36

### **1. Introduction**

Testing is a vital requirement for delivering a consistent and accurate product.

**1.1 Purpose of This Document**

This document is intended for system administrators, quality assurance teams, and other users interested in reviewing the test procedure used for the development of the first iteration of this product. This report will define and analyze the testing process and describe the results.

**1.2 References**

UMBC Bazaar System Requirements Specification

UMBC Bazaar Code Inspection Report

**2. Testing Process**

**2.1 Description**

The use cases described in the SRS documentation influenced each spiral’s testing procedure. Team members were tasked with satisfying specific test conditions derived from the original use cases. Driving project development through satisfying these test conditions was a major goal accomplished during each spiral.

Further testing not described in this document, beyond the planned test conditions, also took place. This included basic unit testing and the execution of the project through multiple usage scenarios. Once the use cases were tested, a desired iteration of the final product for the second spiral was verified.

**2.2 Testing Sessions**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Date | Location | Time Started | Time Ended | Performed By | Use Cases |
| 10/29/2016 | Home | 12:00AM | 7:00AM | Jerson | Register Account |
| 10/29/2016 | Home | 12:00AM | 7:00AM | Jerson | Login to an account |
| 10/30/2016 | Home | 7:00PM | 11:00PM | Jerson | Login to an account |
| 10/30/2016 | Home | 7:00PM | 11:00PM | Jerson | Logout an account |
| 11/05/2016 | Home | 3:00PM | 11:00PM | Jerson | Updating user profile |
| 11/06/2016 | Home | 2:00PM | 11:59PM | Jerson | Search for a product or service |
| 11/17/2016 | Home | 8:00PM | 10:00PM | Jerson | Search for a product or service |
| 11/05/2016 | Home | 3:00PM | 11:00PM | Jerson | Adding a product / service to sell |
| 11/06/2016 | Home | 2:00PM | 11:59PM | Jerson | View product or service detail |
| 11/06/2016 | Home | 2:00PM | 11:59PM | Jerson | Adding a product / service to sell |
| 11/25/2016 | Home | 12:00PM | 7:00PM | Jerson | Checking out with payment options |
| 11/25/2016 | Home | 12:00PM | 7:00PM | Jerson | Checking out with trade options |
| 11/26/2016 | Home | 3:00PM | 8:00PM | Jerson | Checking out with payment options |
| 11/26/2016 | Home | 3:00PM | 8:00PM | Jerson | Checking out with trade options |
| 11/07/2016 | Home | 8:30PM | 9:00PM | Gene | View product or service detail |
| 11/07/2016 | Home | 8:30PM | 9:00PM | Gene | Search for a product or service |
| 11/07/2016 | Home | 8:00PM | 8:30PM | Gene | Adding a product / service to sell |
| 11/07/2016 | Home | 7:00PM | 8:00PM | Gene | Register Account |
| 11/07/2016 | Home | 7:00PM | 8:00PM | Gene | Login to an account |
| 11/07/2016 | Home | 7:00PM | 8:00PM | Gene | Logout an account |

**2.3 Impressions of the Process**

A testing procedure focused on checking whether the use cases were satisfied worked well with verifying the base functionality of the web application. Through the use of test conditions derived from the use cases, necessary requirements were tested accurately. These use cases created a guideline to follow throughout project development and helped separate workload.

The strongest components of the modular units of the application are the user registration and login components. The two modules accesses several systems such as the MySQL database and UMBC Directory services as well as session variables which are all vital for testing the registration, security and login conditions.

See the Code Inspection report for details of areas of code unlikely and likely to contain flaws.

**3. Test Results**

Testing was broken down into equivalence partitions where boundary condition

segments and common segments were tested. Each use case was tested using boundary conditions. The test results shown are all from the server-side validation results. There is no way to screenshot most of the client-side validations that tests user input through active listeners in the javascript.

**3.1 Testing Suite**

The description of how testing was partitioned and implemented is outlined below.

|  |  |
| --- | --- |
| **Use Case** | **Register an account** |
| Valid Situation | User clicks the “Create a New Account” button and is prompted to enter their UMBC ID, username, password and other information. Then, hit the register button. |
| Invalid Situation | The user clicks the register button and the user credentials entered are invalid or information submitted is formatted incorrectly. |
| Purpose | To test the ability to register for an account. |
| Expected Results for Valid Situation | A message displayed instructing the user to confirm the registration with their UMBC email. |
| Expected Results for Invalid Situation | An error message is displayed prompting the user to retry and enter their information. |
| Boundary Conditions | 1. Invalid UMBC ID based on UMBC Directory Services. 2. The format of submitted information is invalid. |

|  |  |
| --- | --- |
| **Use Case** | **Login to an account** |
| Valid Situation | User clicks the login button and is prompted to enter their username and password. If the user is already logged in, the user is redirected. |
| Invalid Situation | The user clicks the login button and submitted user credentials are invalid. Invalid login attempt exceeded. |
| Purpose | To test the ability to login into the web application. |
| Expected Results for Valid Situation | The user is redirected to the main page that list the products and services posted. |
| Expected Results for Invalid Situation | An error message is displayed prompting the user to retry and enter their credentials. |
| Boundary Conditions | 1. Invalid login credentials result in error. 2. Invalid login attempt exceeded. |

|  |  |
| --- | --- |
| **Use Case** | **Logout an account** |
| Valid Situation | User clicks the logout button and the user is logged in. |
| Invalid Situation | The user manually enters the logout and is not logged in. |
| Purpose | To test the ability to logout an account. |
| Expected Results for Valid Situation | The session is invalidated and the user is redirected to the login page. |
| Expected Results for Invalid Situation | The session is invalidated and the user is redirected to the login page. |
| Boundary Conditions | Invalid situation has no effect and the outcome is the same. |

|  |  |
| --- | --- |
| **Use Case** | **Updating user profile** |
| Valid Situation | The user clicks the “Profile” button, and is prompted to enter their information (name, phone, new password). The information submitted are valid. The user clicks the “Update Profile” button. |
| Invalid Situation | The user click the “Profile” button and enters their information. The information is invalid. |
| Purpose | To test the ability to update user profile. |
| Expected Results for Valid Situation | The user submitted changes are updated on the database. A success message is displayed and the changes are reflected on the textboxes displayed. |
| Expected Results for Invalid Situation | An error message is displayed prompting the user to retry and enter validly formatted information. |
| Boundary Conditions | 1. User must be logged in. 2. Current password entered is invalid. 3. User information format is invalid (name, phone, new password, etc.). |

|  |  |
| --- | --- |
| **Use Case** | **Searching for a product or service** |
| Valid Situation | User searches for a product or service |
| Invalid Situation | Product or service not found or nothing is entered |
| Purpose | To search for a product or service |
| Expected Results for Valid Situation | The desired product or service appears after the search is performed |
| Expected Results for Invalid Situation | An error message saying that no results have been found or that the search string is empty is displayed |
| Boundary Conditions | 1. User must be logged in. 2. Nothing entered to search. |

|  |  |
| --- | --- |
| **Use Case** | **View product or service detail** |
| Valid Situation | The user clicks on the desired product or service |
| Invalid Situation | The user does not click on the product or service they wish to view. Invalid product. |
| Purpose | To provide greater detail for the selected product or service |
| Expected Results for Valid Situation | The user is directed to a page with more detail for a product or service |
| Expected Results for Invalid Situation | The user is either directed to the page of what they click or nothing happens |
| Boundary Conditions | 1. User must be logged in to view a product or service 2. Product ID submitted is not in the database |

|  |  |
| --- | --- |
| **Use Case** | **Adding a product / service to sell** |
| Valid Situation | Click “Add a new Product or Service” and enter the product/service information. |
| Invalid Situation | The product information entered is invalid. The user fields are required. |
| Purpose | To add new products or services to the list of available products and services |
| Expected Results for Valid Situation | The product or service is added to the system correctly and is able to be purchased |
| Expected Results for Invalid Situation | Displays an error message that fields are required except for image, must meet a pattern or file size is too big. |
| Boundary Conditions | 1. User must be logged in to add a product or service 2. Must have at least a dollar amount or trade selected. 3. Image size must be <= 500kb. 4. Price must be numeric. |

|  |  |
| --- | --- |
| **Use Case** | **Checkout with payment methods** |
| Valid Situation | The user purchases a product using payment methods |
| Invalid Situation | No Invalid situation. Cash on delivery only. Checkout will be disabled if cash payment is not accepted. |
| Purpose | Paying to buy products or services |
| Expected Results for Valid Situation | Email confirmation is sent to the buyer and seller |
| Expected Results for Invalid Situation | Will default back to the product listing page.  If the description box is left blank then an error message will be displayed |
| Boundary Conditions | 1. User must be logged in to checkout with payment options 2. Payment by cash on delivery must be enabled for the product listing. |

|  |  |
| --- | --- |
| **Use Case** | **Checkout with trade options** |
| Valid Situation | The user checks out with trade options |
| Invalid Situation | The user attempts to trade but has not products or services to trade with or the description box is empty |
| Purpose | Using trade to purchase a product or service |
| Expected Results for Valid Situation | The user selects what they are offering to trade. An email is then sent to confirm if the seller wishes to trade their product or service for the buyer's proposed trade. |
| Expected Results for Invalid Situation | The user will not be able to trade if they have nothing to trade with. If the description box is left blank then an error message will be displayed |
| Boundary Conditions | 1. User must be logged in to checkout with trade options 2. 2. Trade option must be enabled for the product listing. |

**3.2 Test Results**

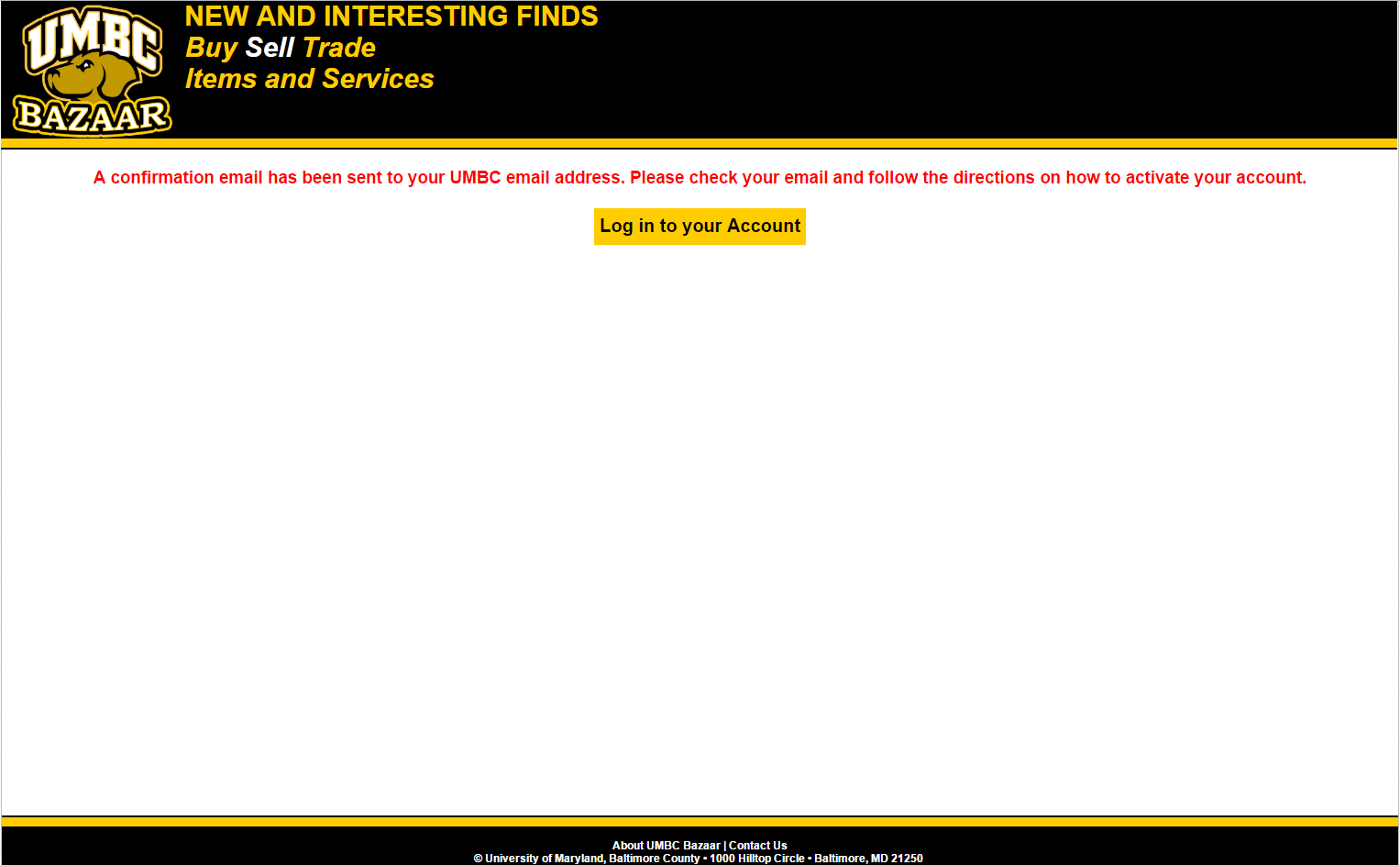
The results from testing are outlined in the tables below.

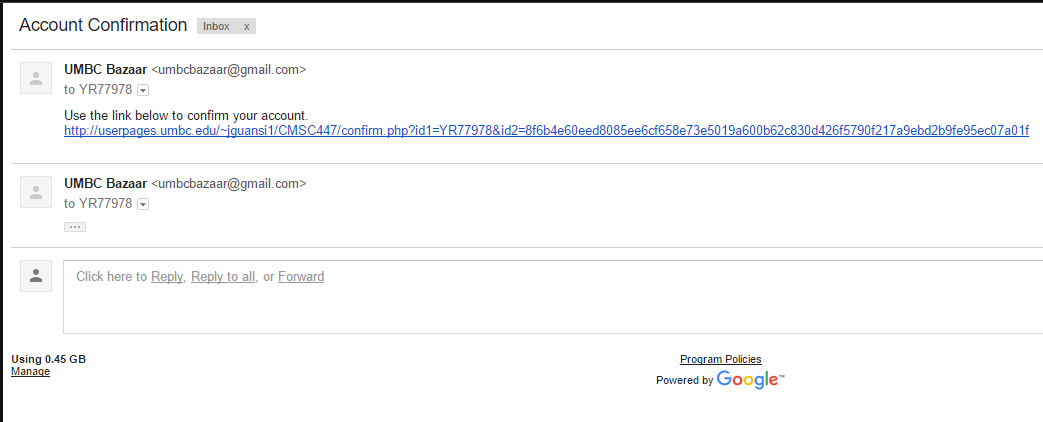
**3.2.1 Use Case: Register an account**

Conducted by: Jerson Guansing

**Valid Situation:**

Message is displayed requiring the user to confirm the registration.

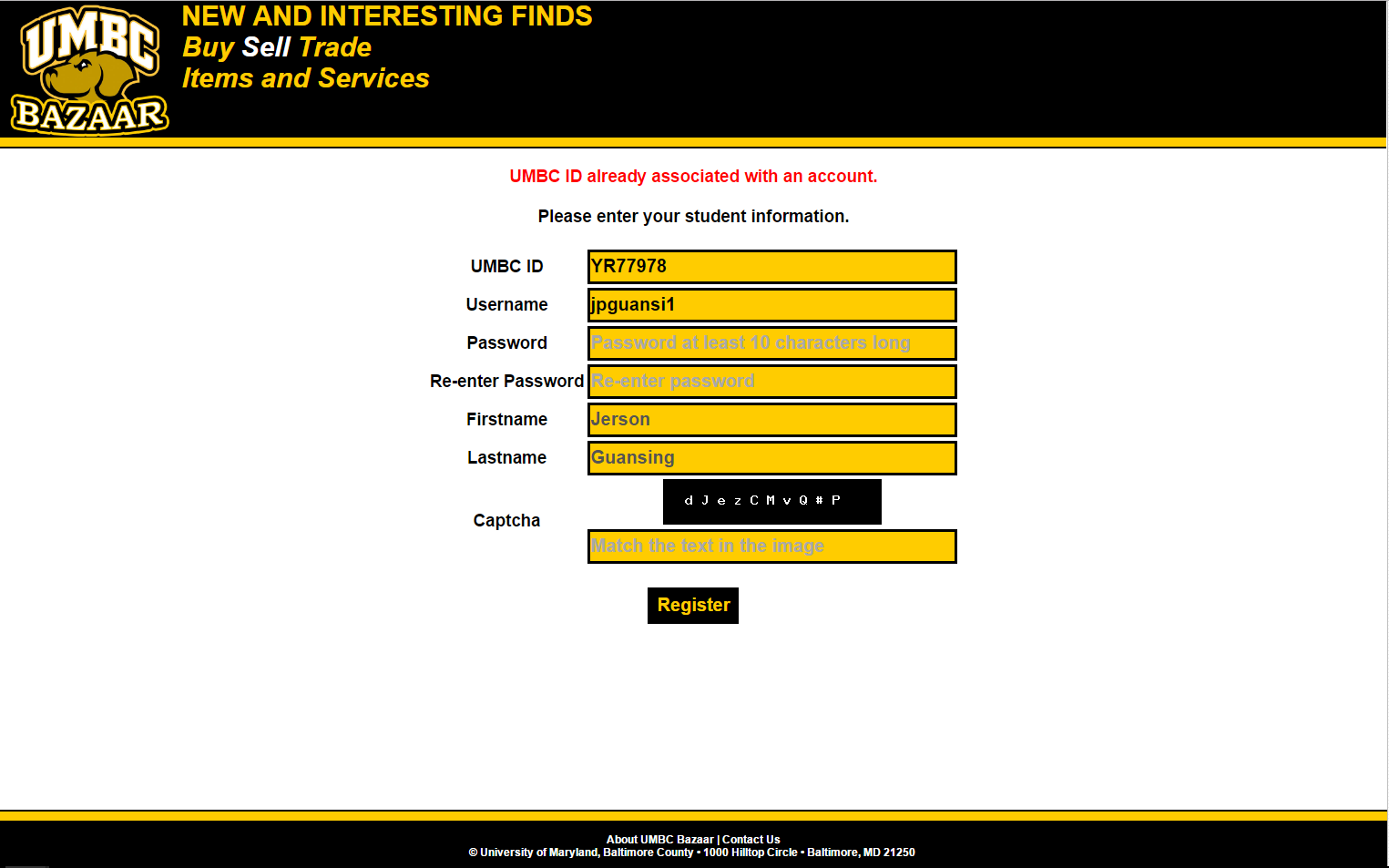




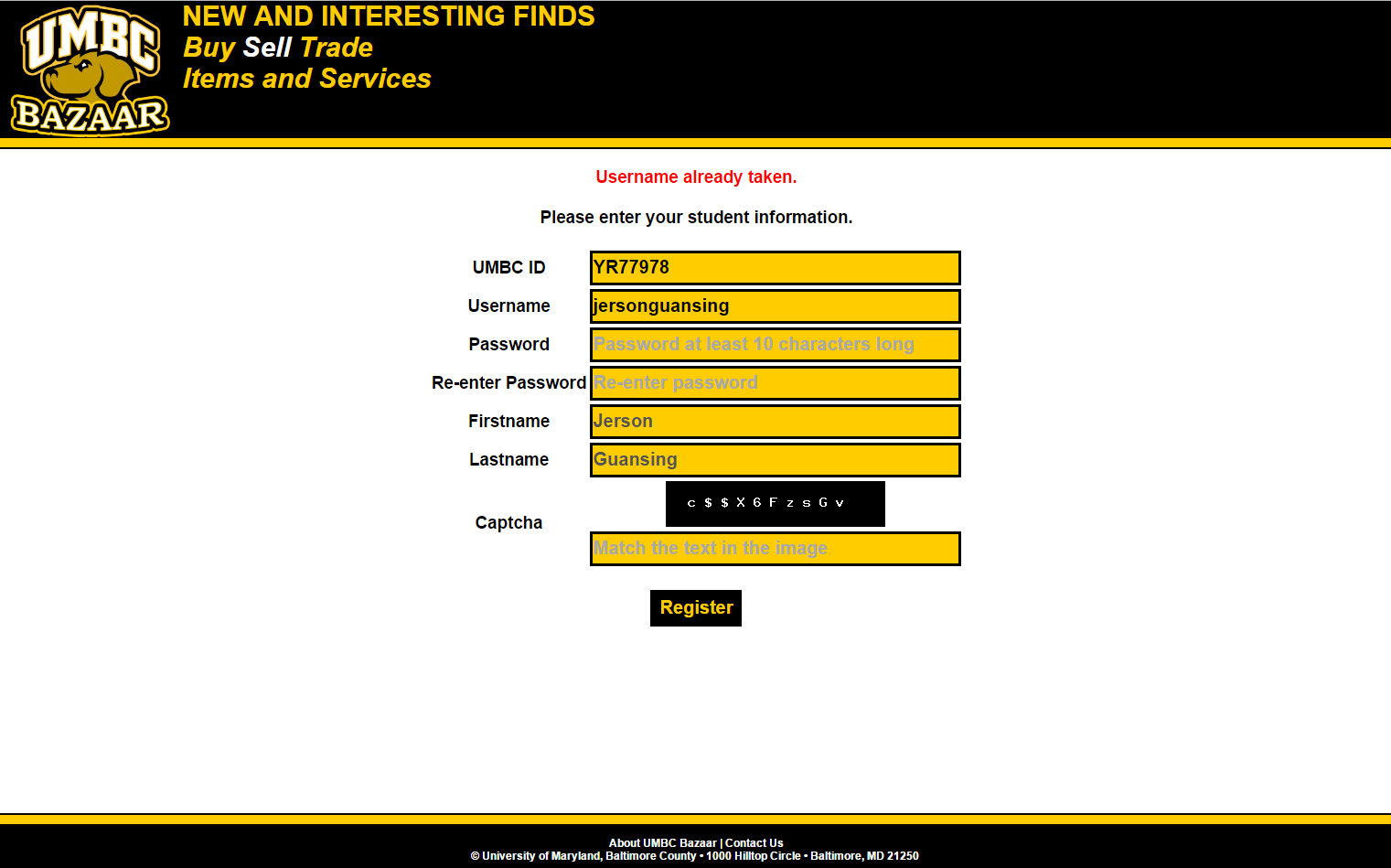
**Invalid Situation:**

Input listeners applied to each textbox to ignore invalid characters.

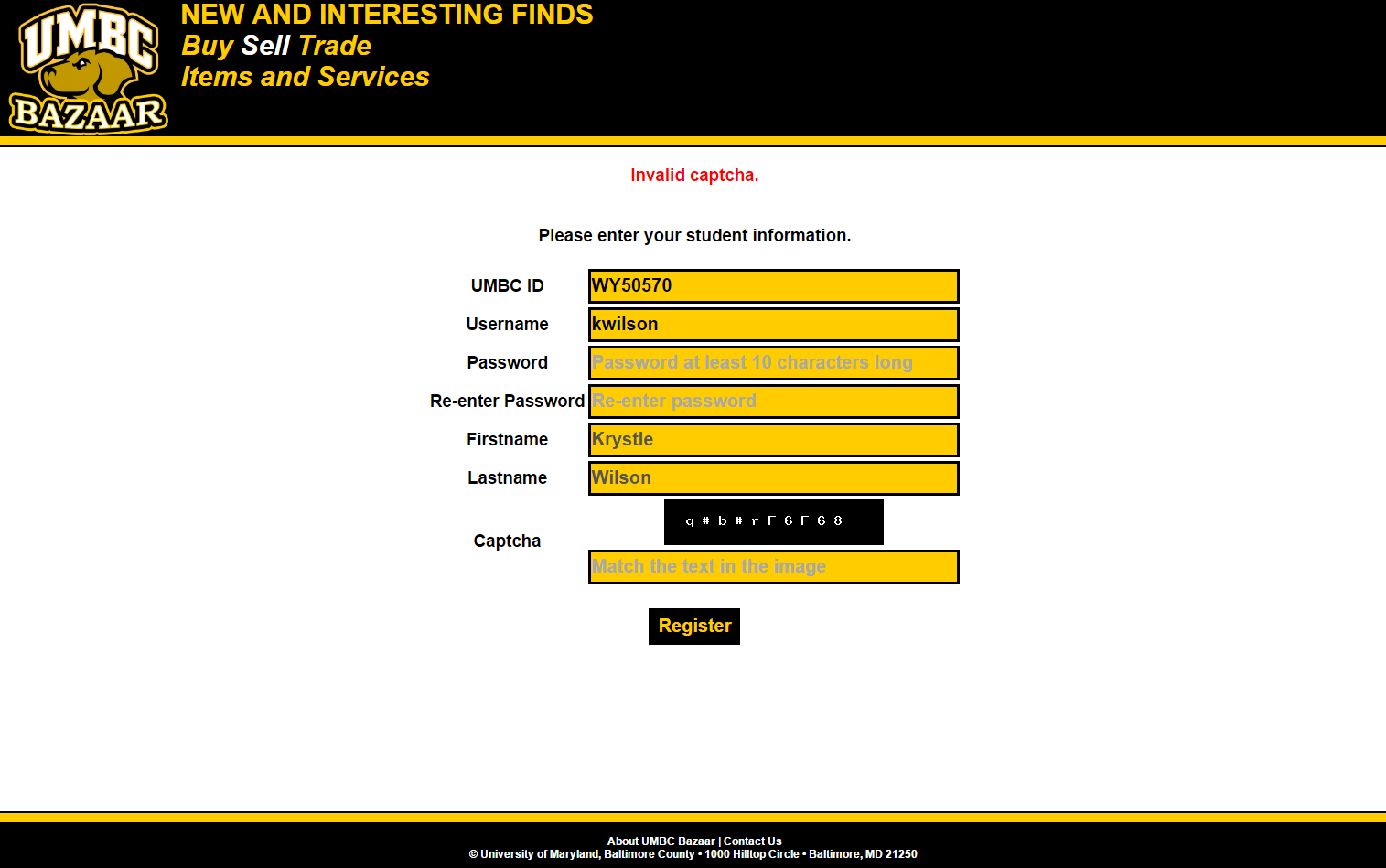
UMBC ID is already associated with an account.



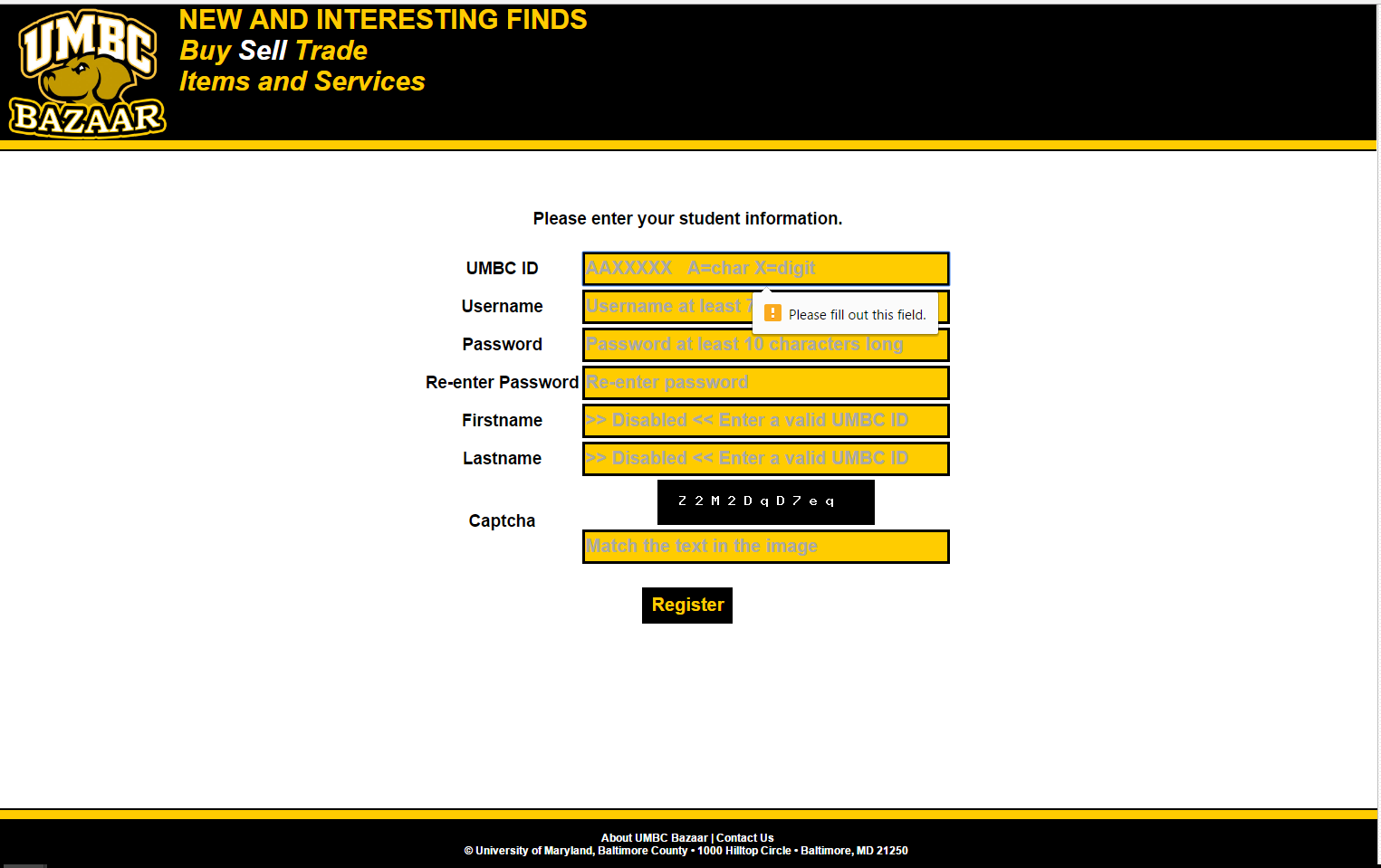
Username is already taken.



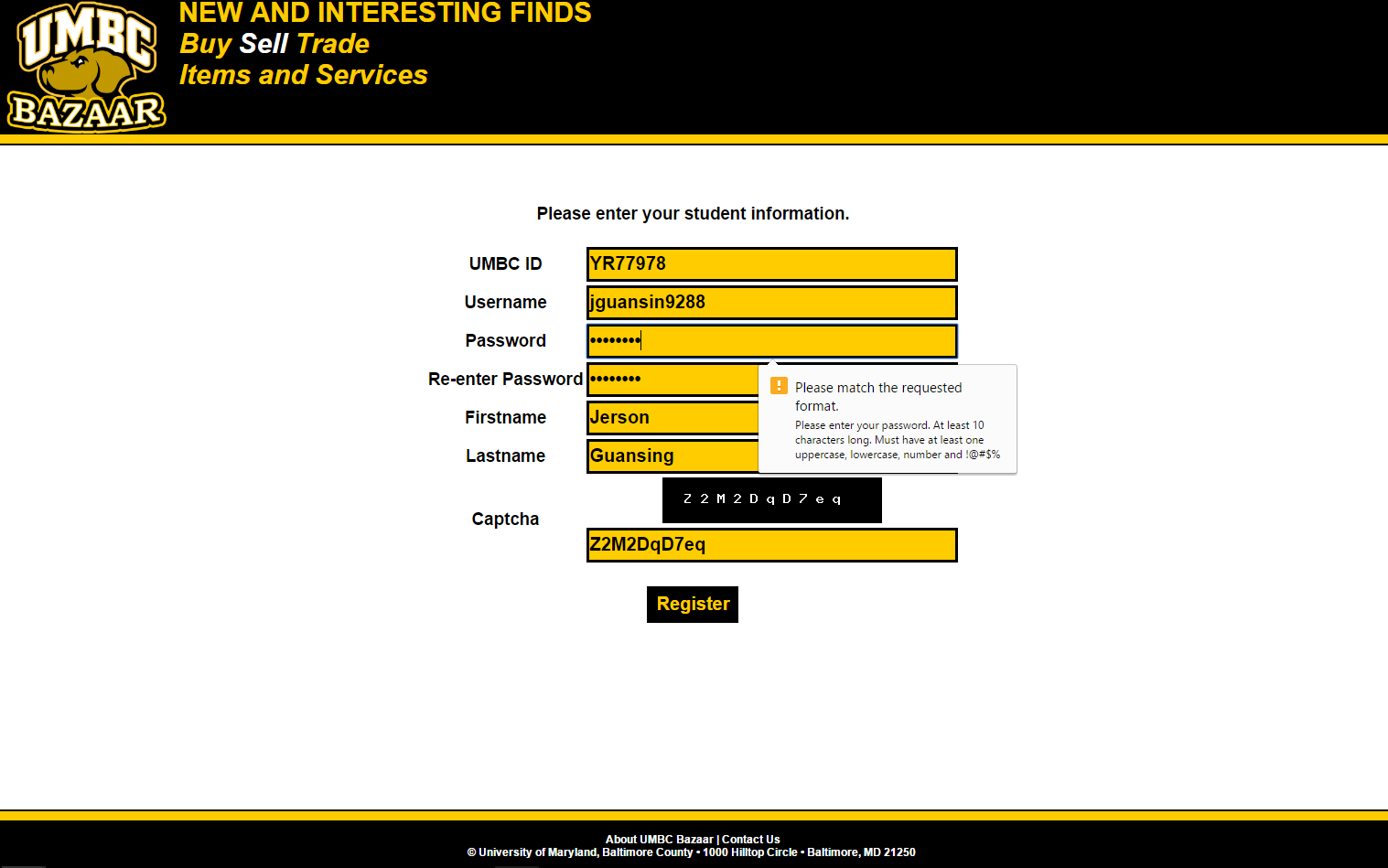
Invalid captcha



All the textboxes are required fields.



The user input must meet the required regular expression pattern.

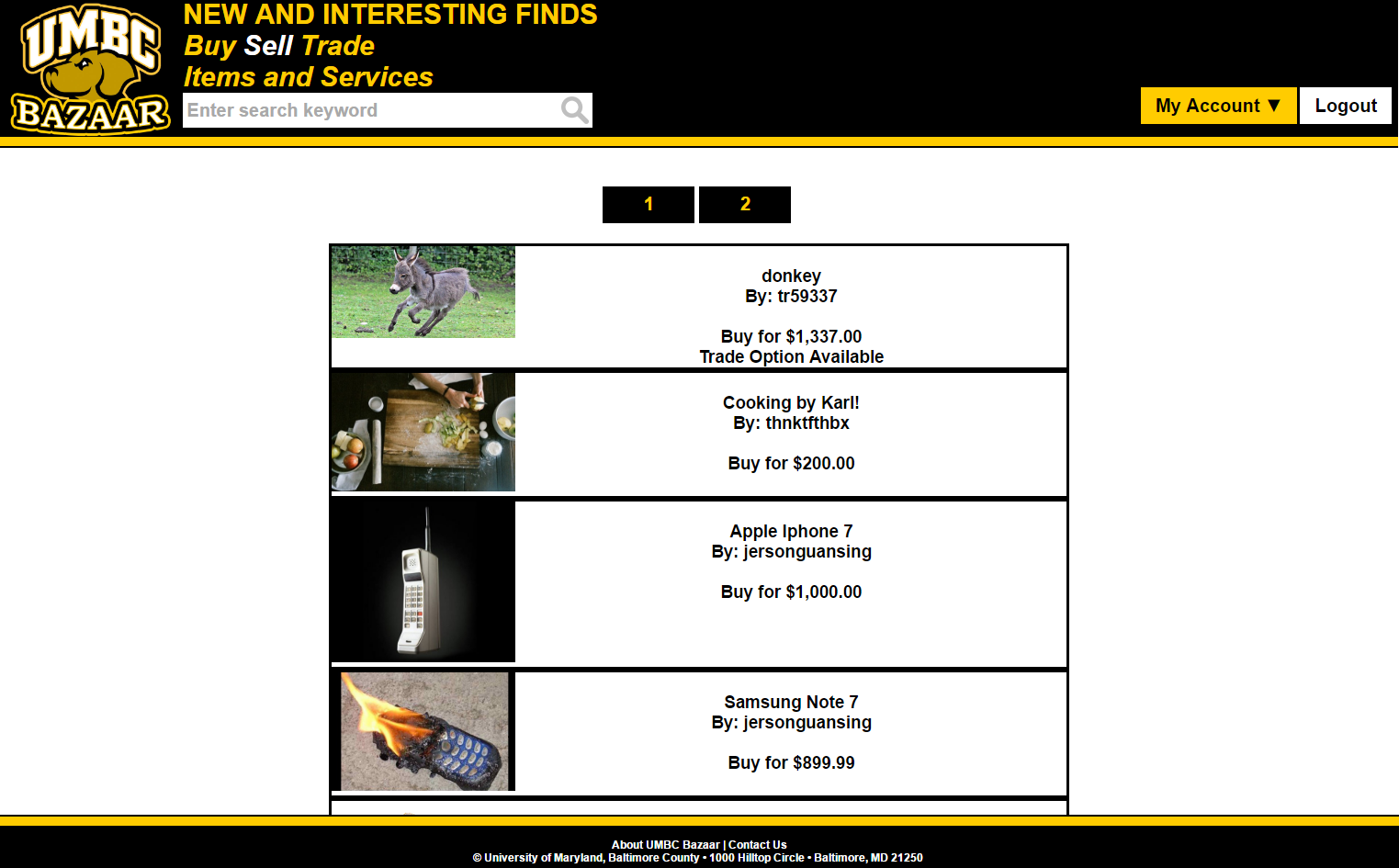


**3.2.2 Use Case: Login to an account**

Conducted by: Jerson Guansing

**Valid Situation:**

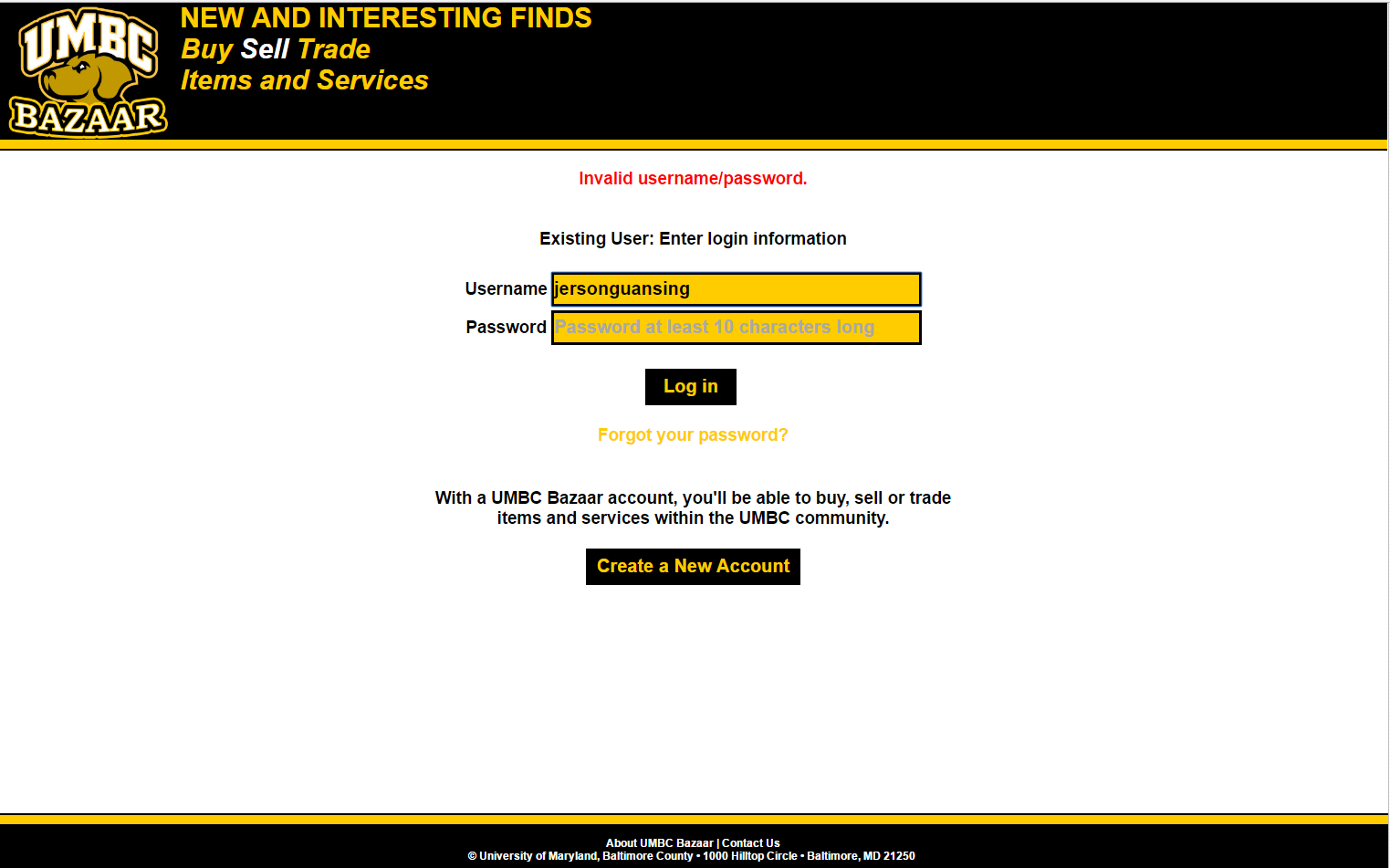
User redirected to the main page



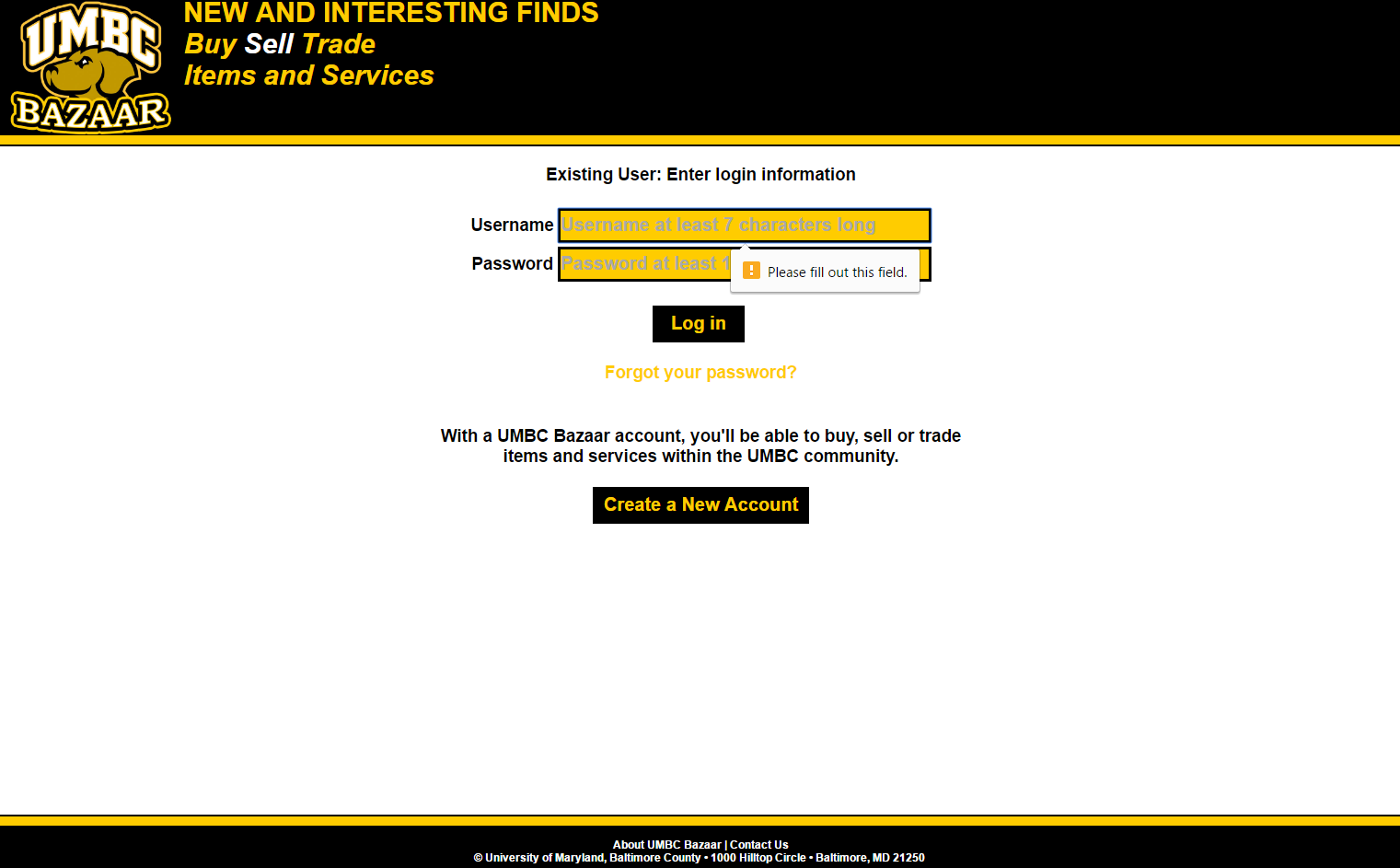
**Invalid Situation:**

Input listeners applied to each textbox to ignore invalid characters.

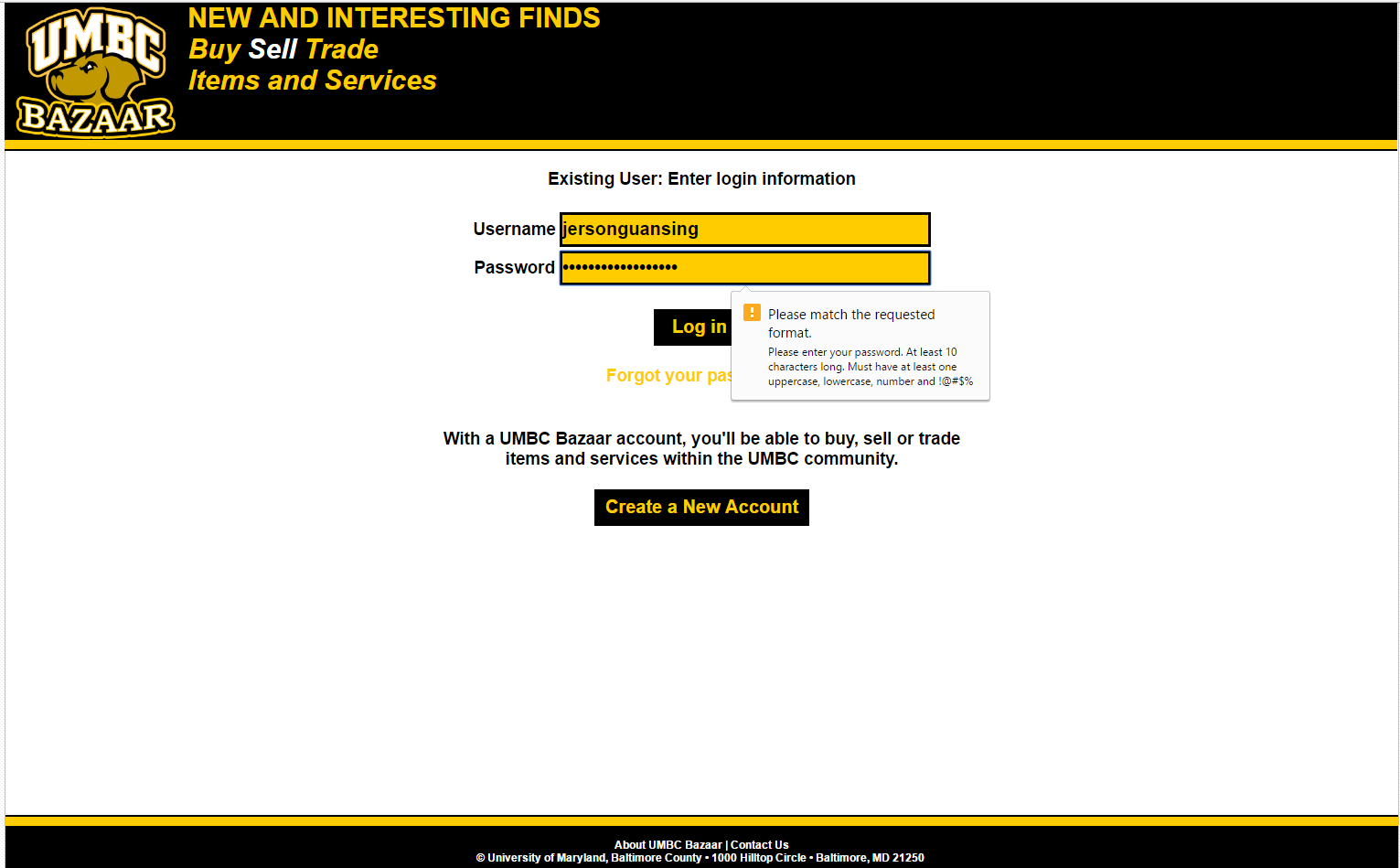
Invalid username and/or password



Username and password are required fields.



The user input must meet the required regular expression pattern.

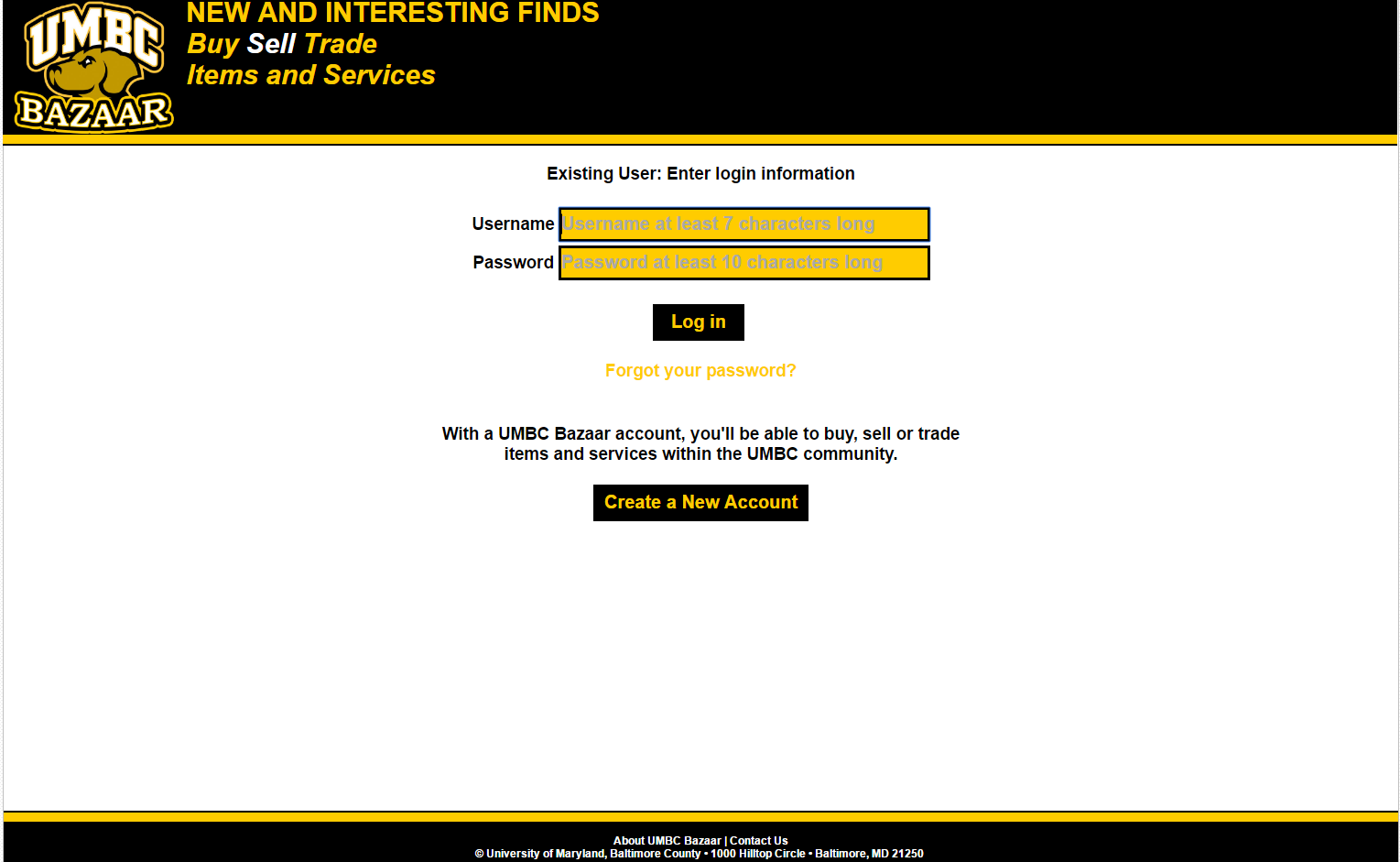


**3.2.3 Use Case: Logout an account**

Conducted by: Jerson Guansing

**Valid or Invalid Situation:**

Logout.php invalidates session and just redirects to the login page.

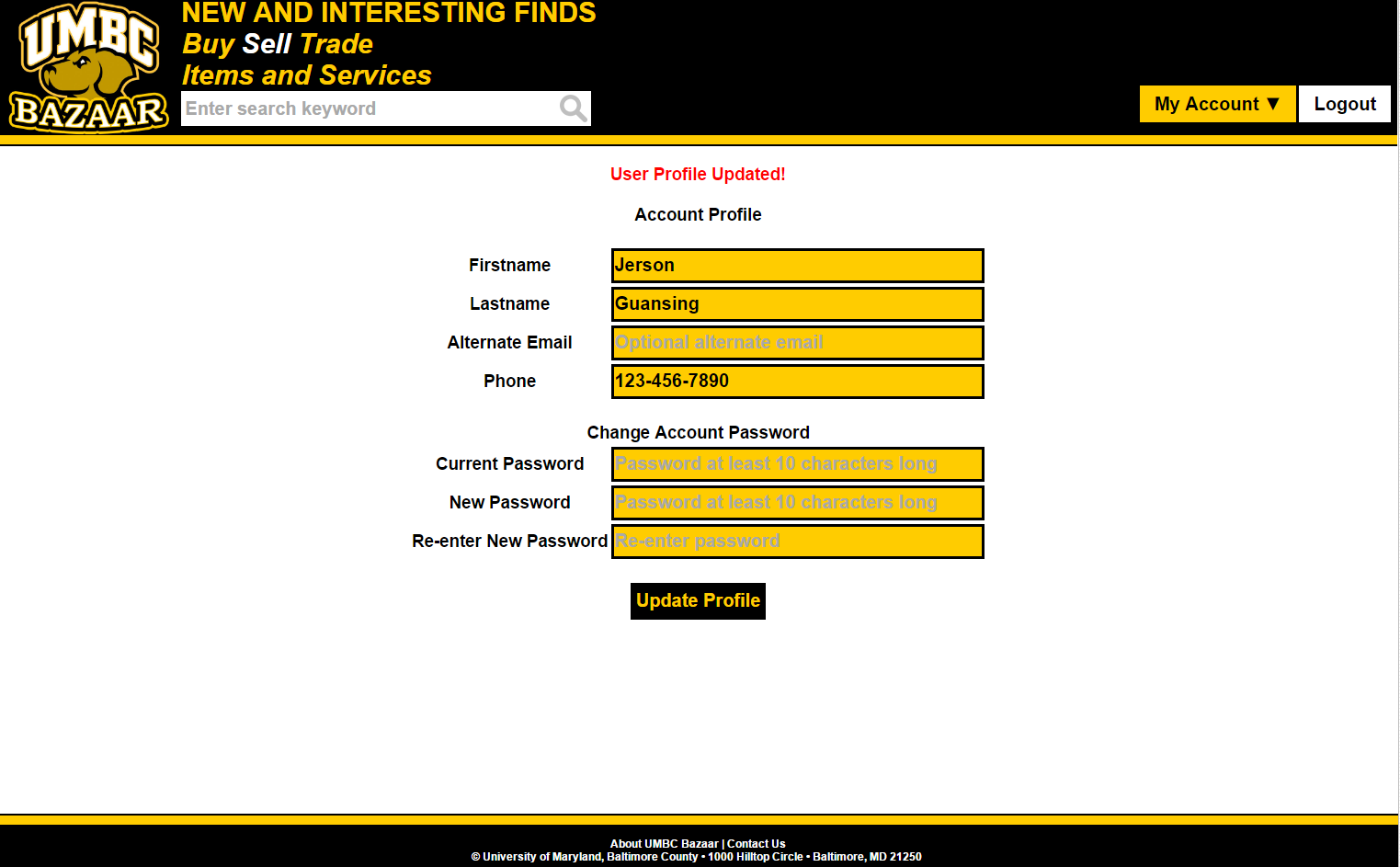


**3.2.4 Use Case: Updating user profile**

Conducted by: Jerson Guansing

**Valid Situation:**

Message is displayed to the user that their information has been updated.

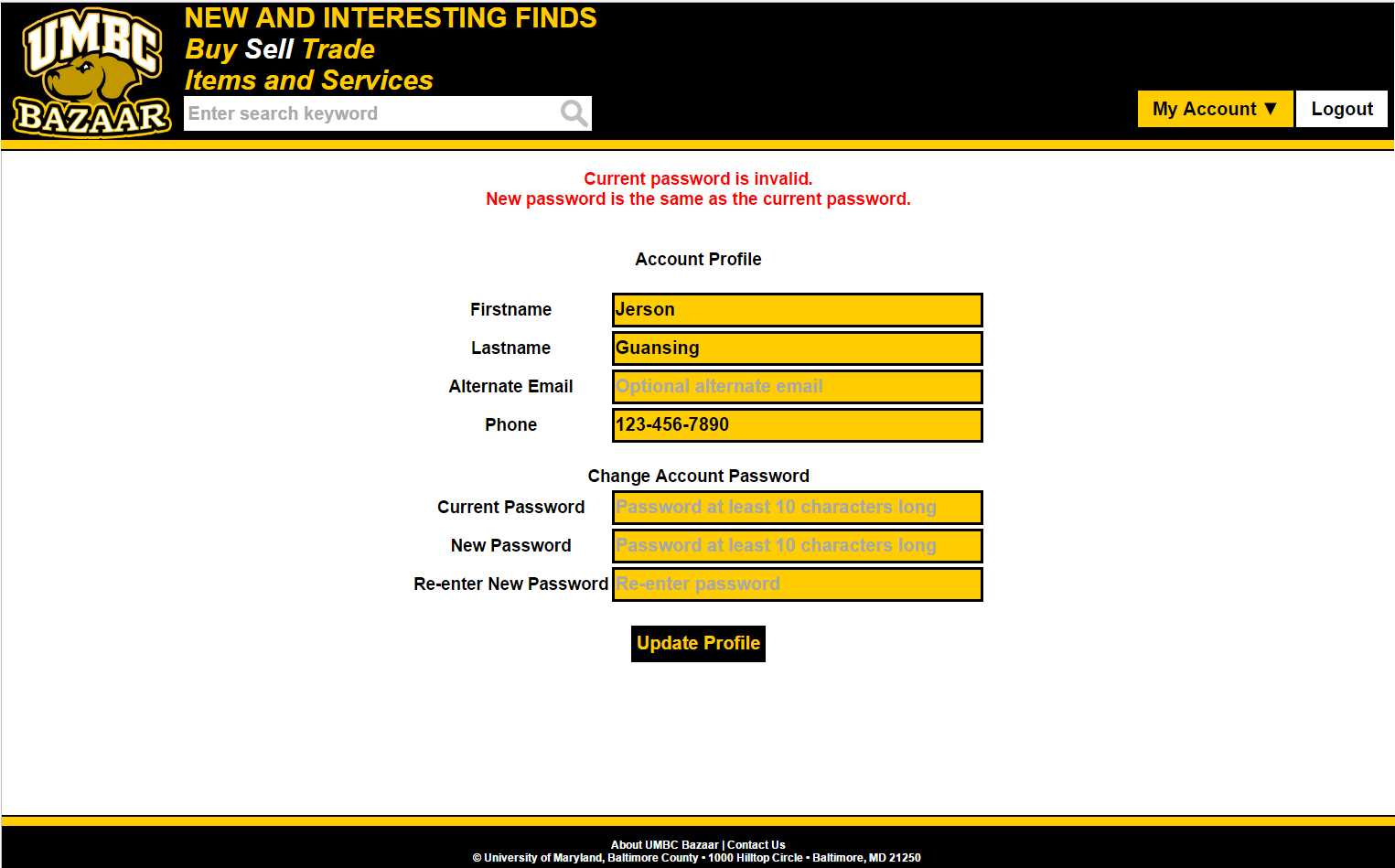


**Invalid Situation:**

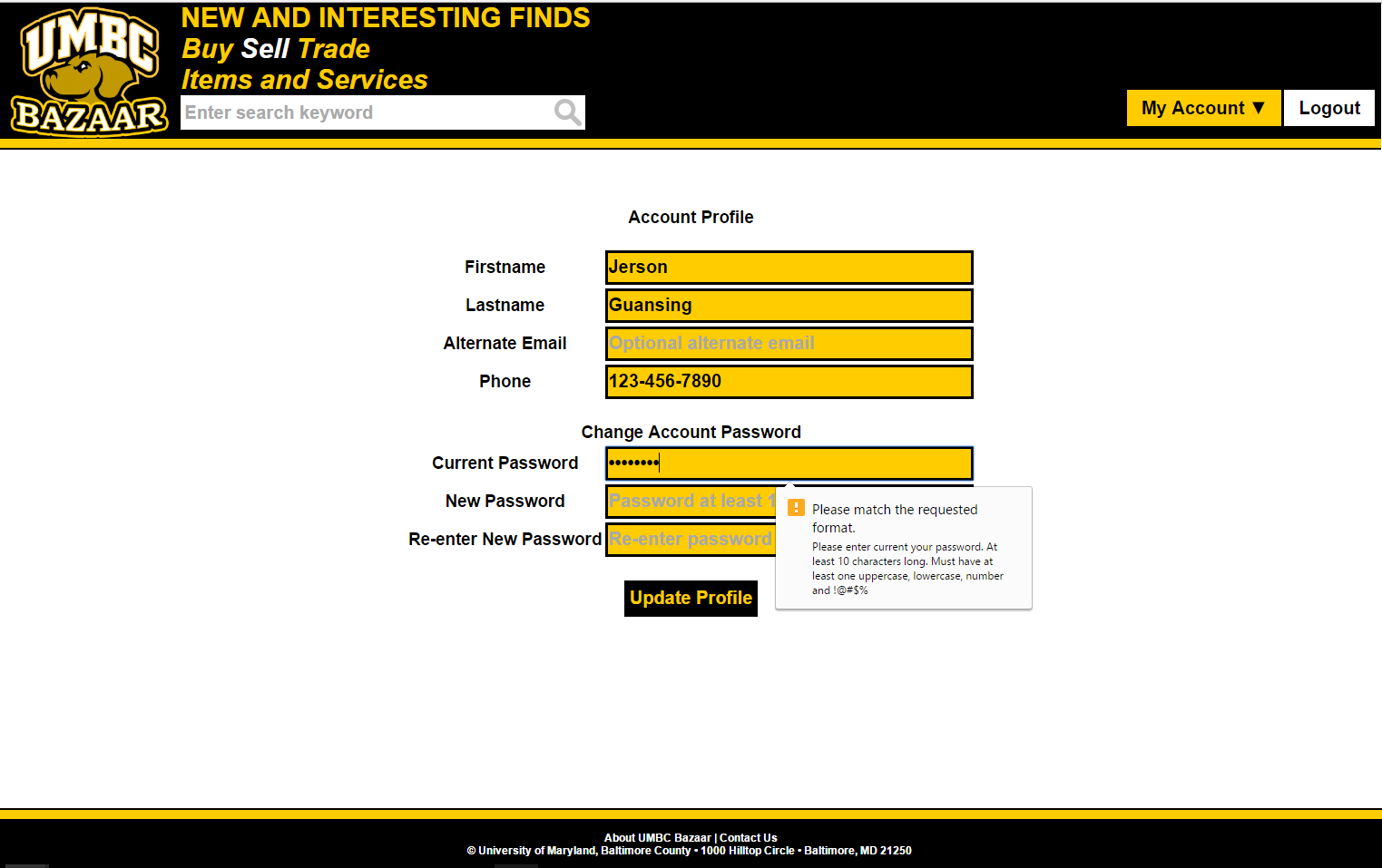
Input listeners applied to each textbox to ignore invalid characters.

The current password entered is invalid.

The new password is the same as the current password.



The user input must meet the required regular expression pattern.

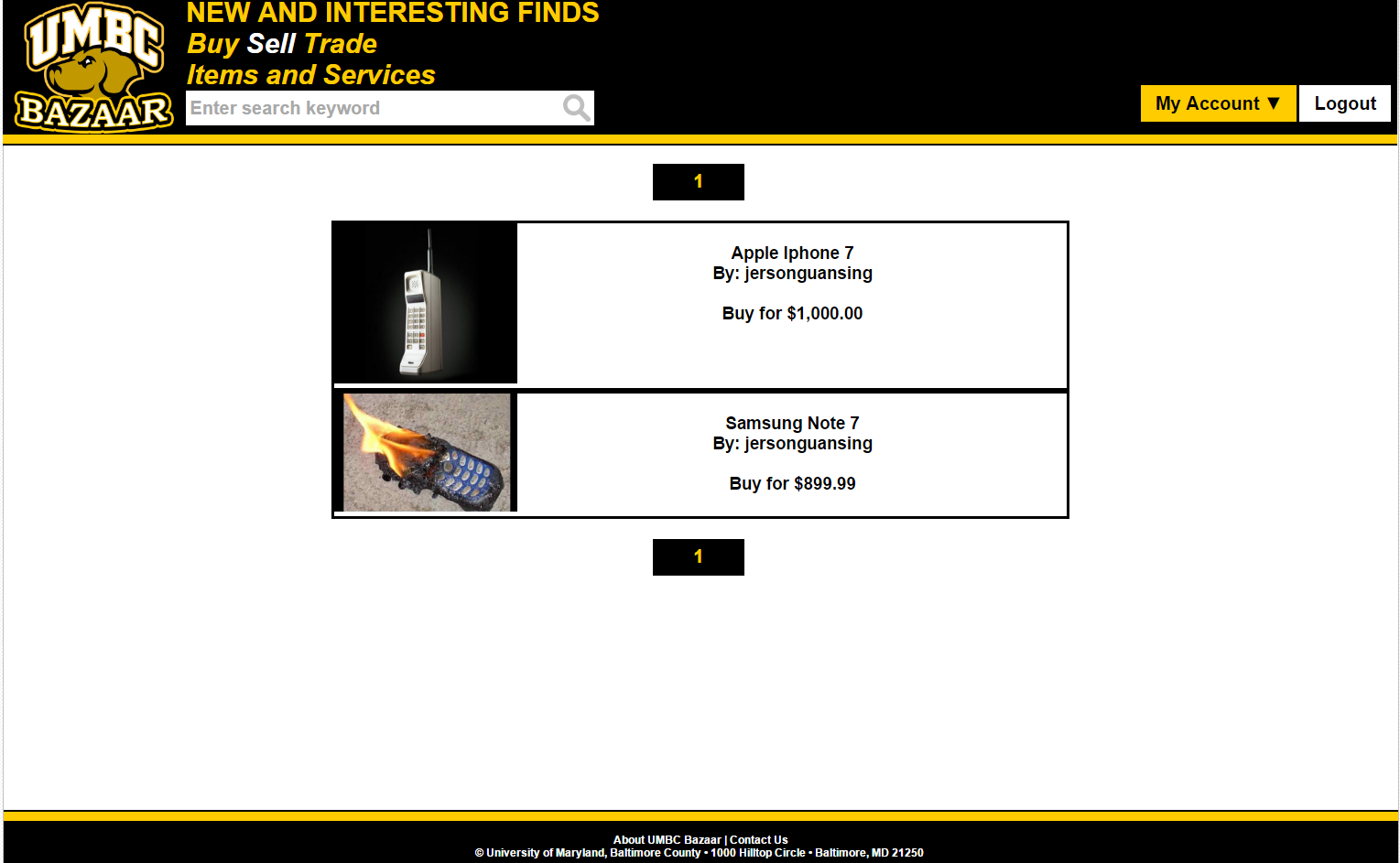


**3.2.5 Use Case: Searching for a product or service**

Conducted by: Jerson Guansing

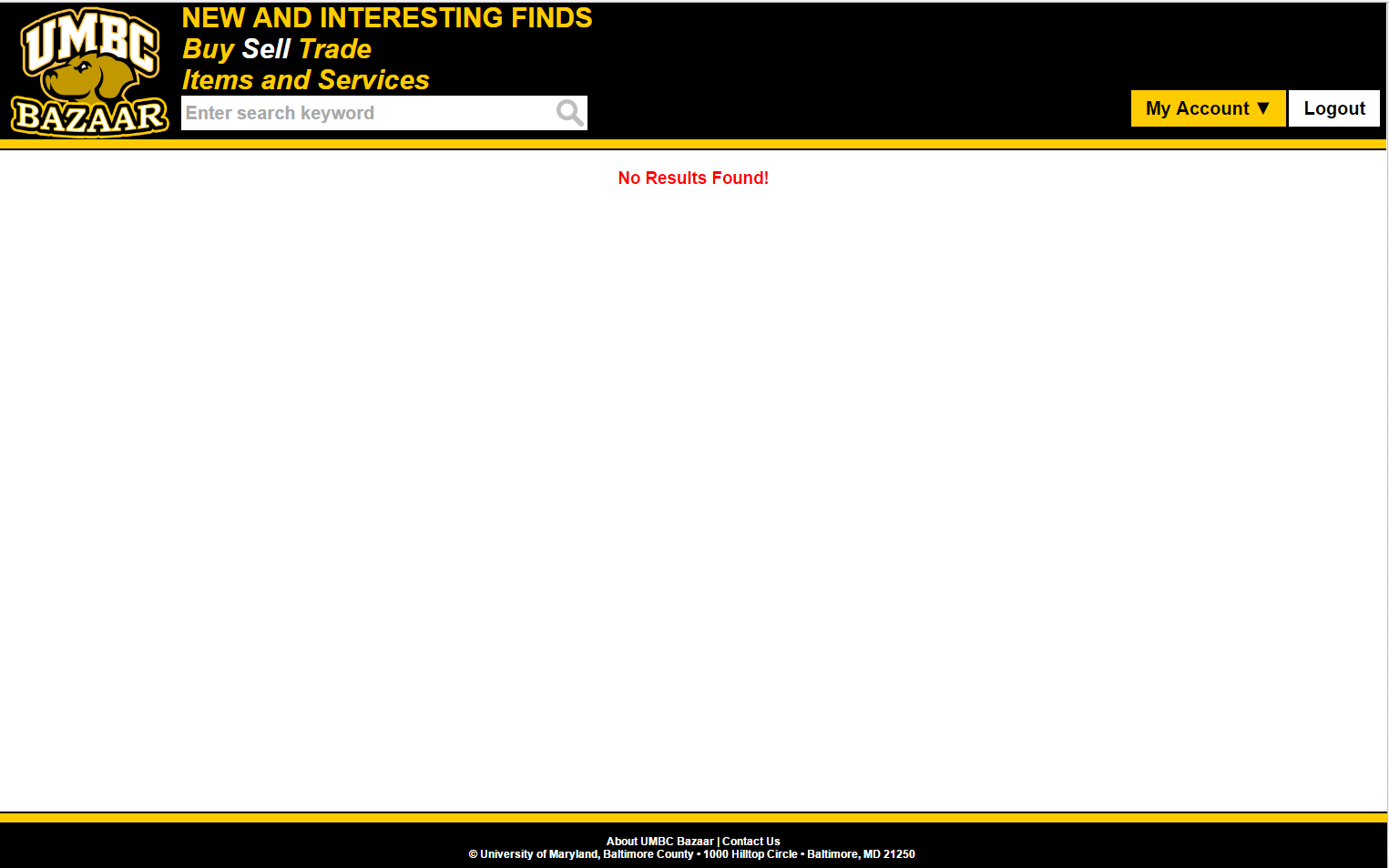
**Valid Situation:**

Product listing is displayed that matches the search keyword.



**Invalid Situation:**

No results found for the given keyword.

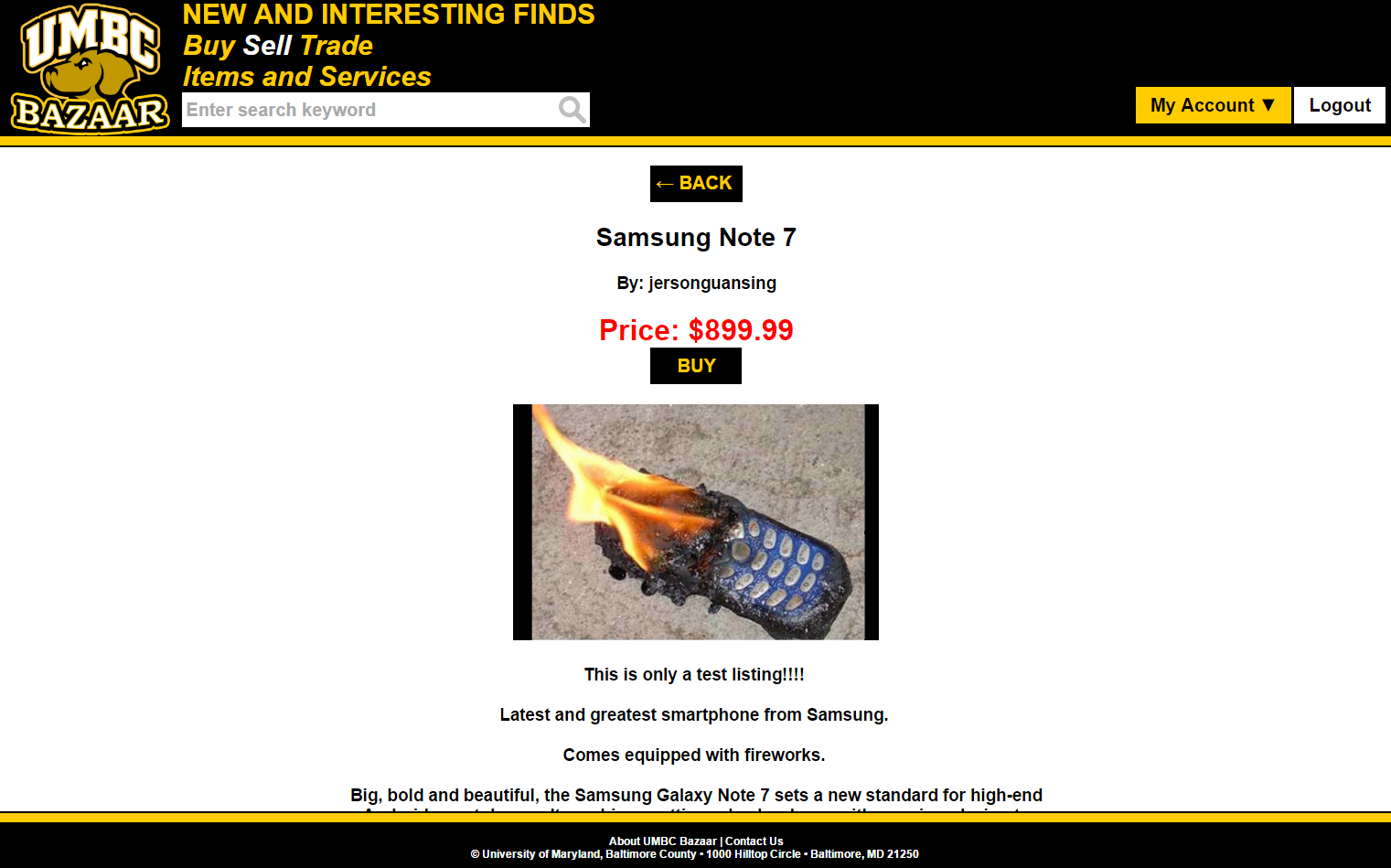


**3.2.6 Use Case: View a product / service to sell**

Conducted by: Jerson Guansing

**Valid Situation:**

Displays details about the specific product or service.



**Invalid Situation:**

Displays an error message that the product is not found.

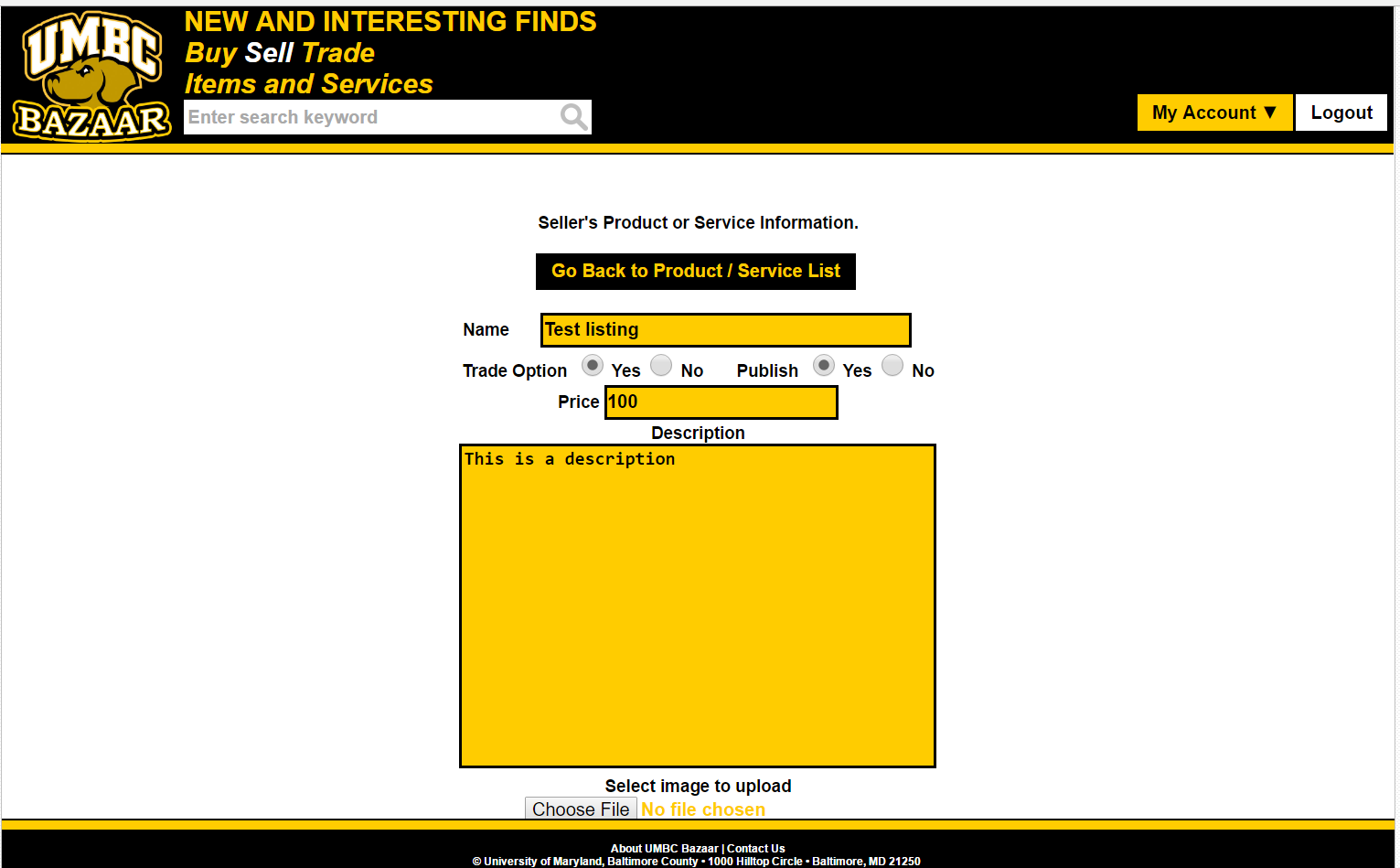


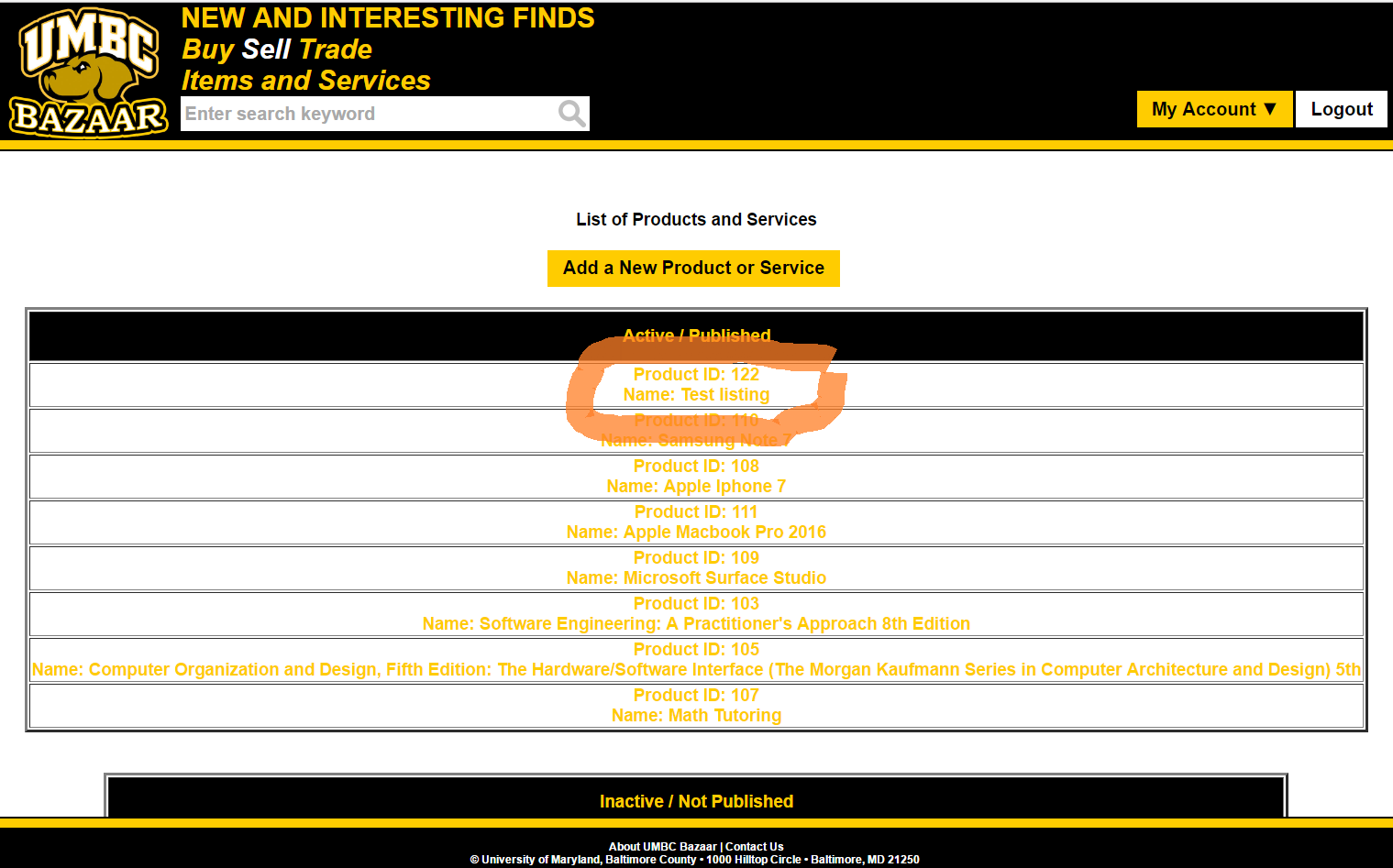
**3.2.7 Use Case: Adding a product / service to sell**

Conducted by: Jerson Guansing

**Valid Situation:**

The user is redirected back to “My Products” page showing the newly added product.

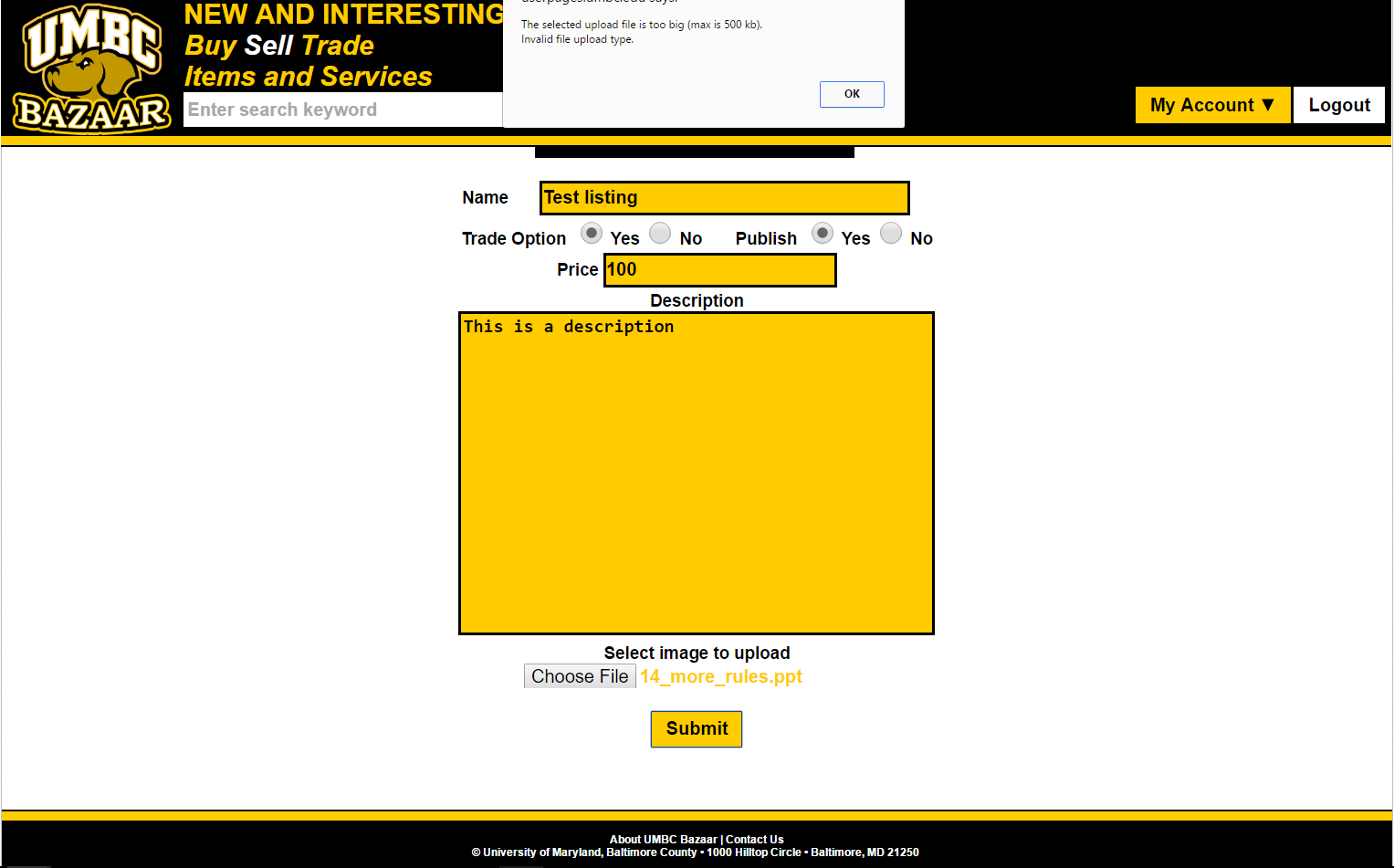




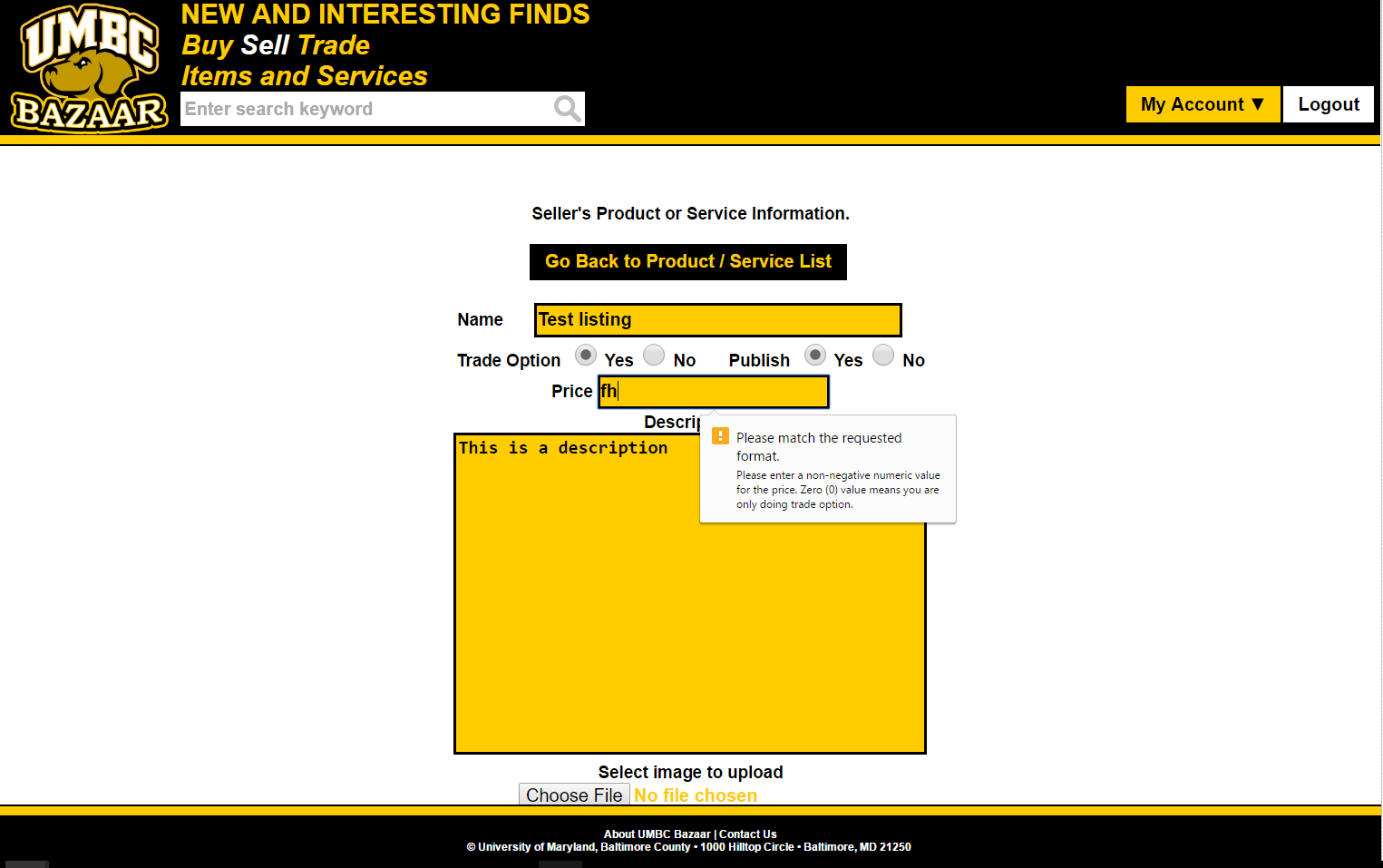
**Invalid Situation:**

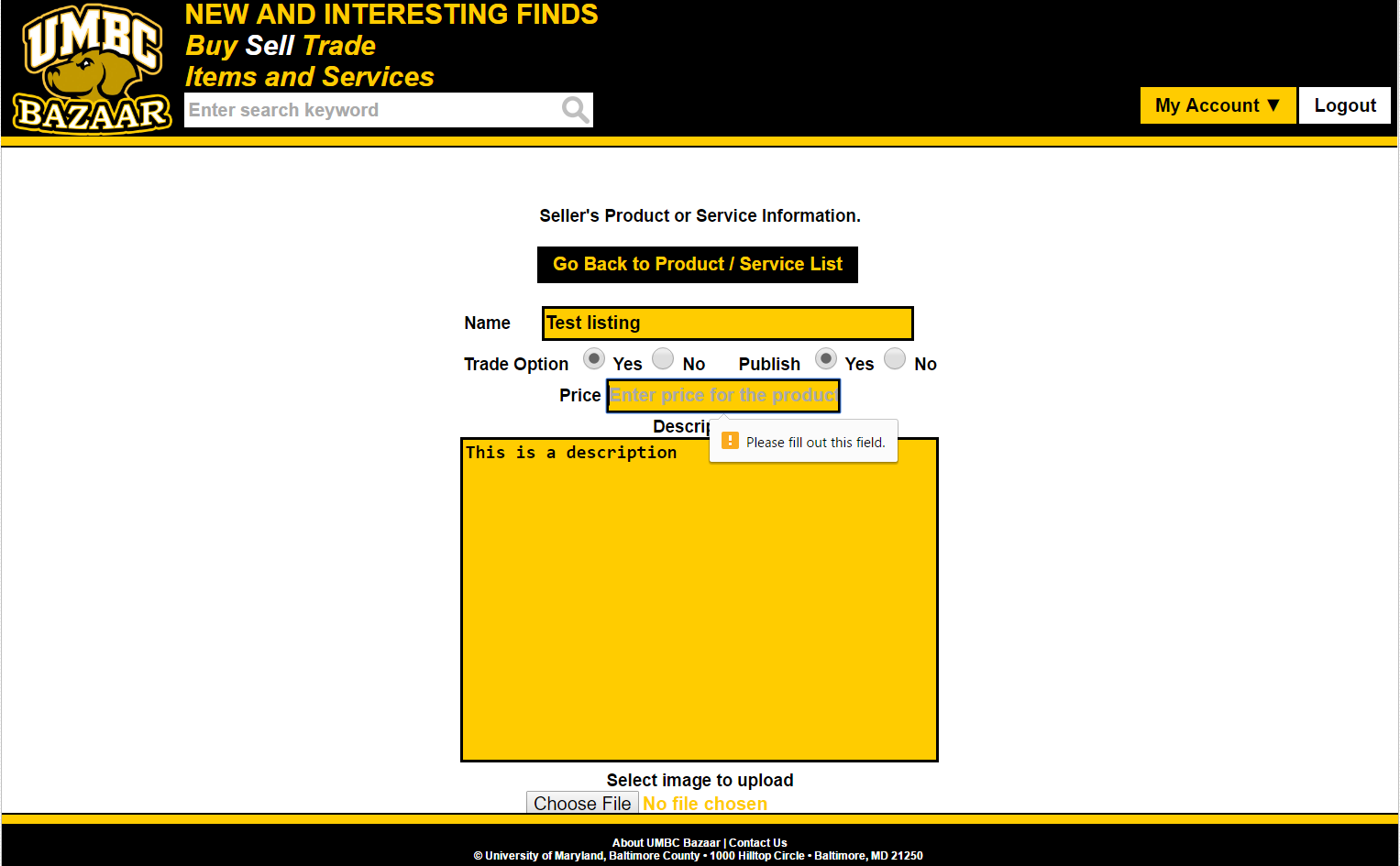
Input listeners applied to each textbox to ignore invalid characters.

A message is displayed prompting the user to reenter the information.



The user input must meet the required regular expression pattern.



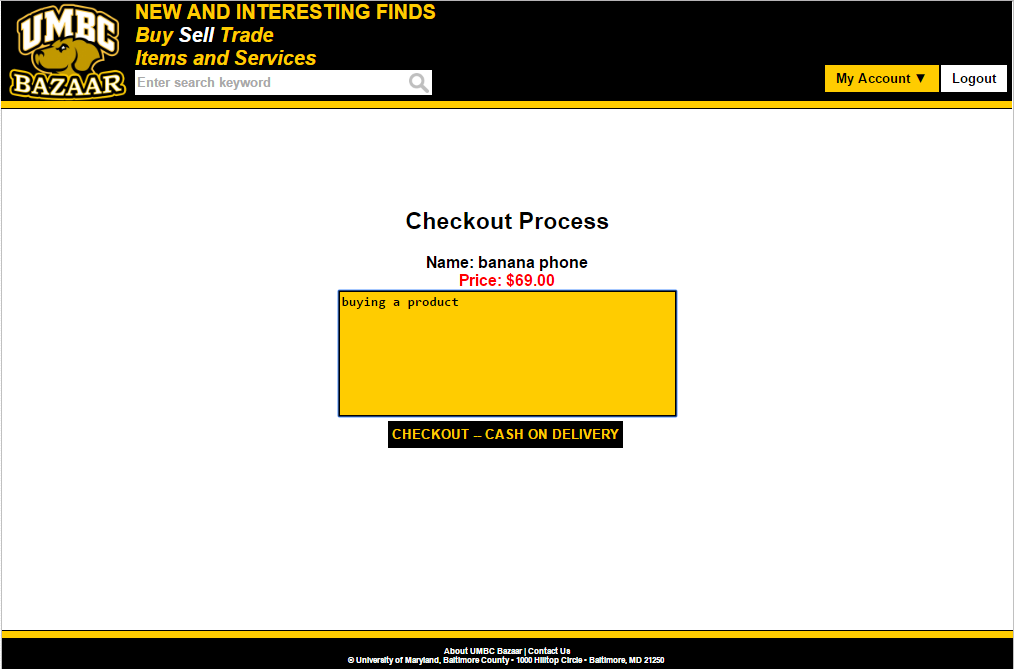
All fields are required except for image to upload.

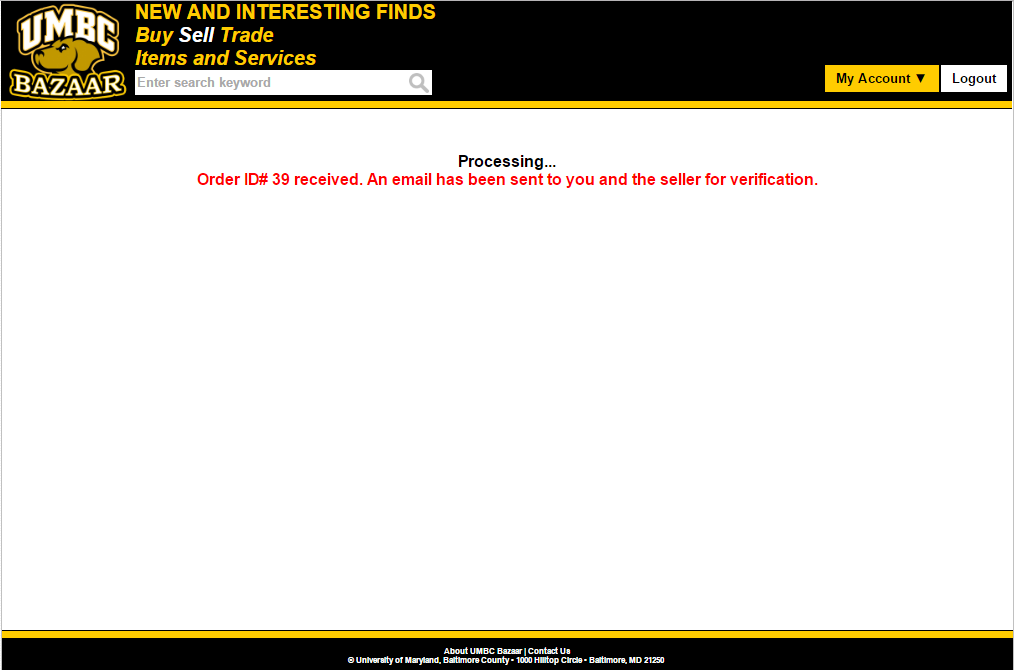
**3.2.8 Use Case: Checkout with payment methods**

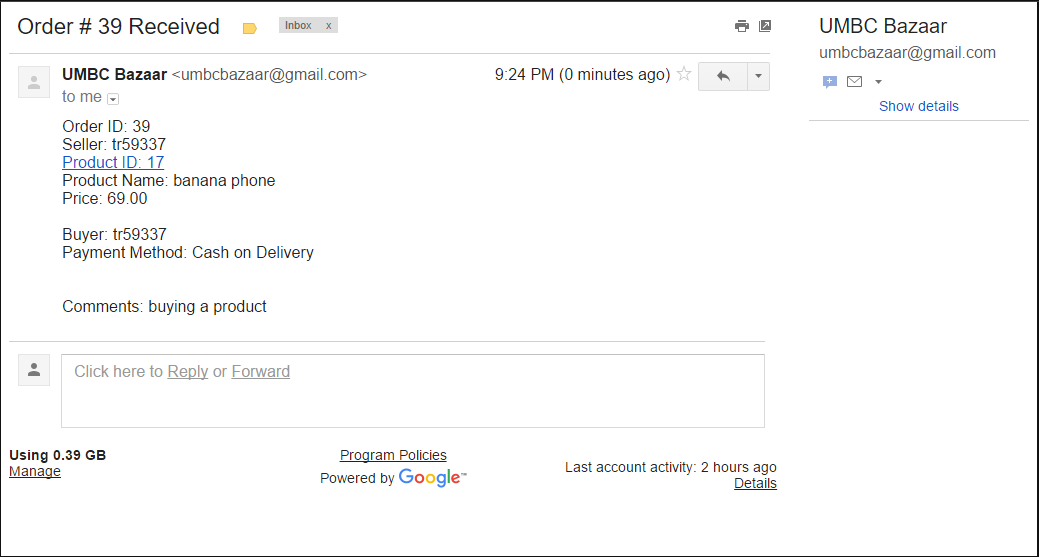
Conducted by: Jerson Guansing

**Valid Situation:**

The user successfully checks out using payment methods.







**Invalid Situation:**

Checking out an invalid product id.

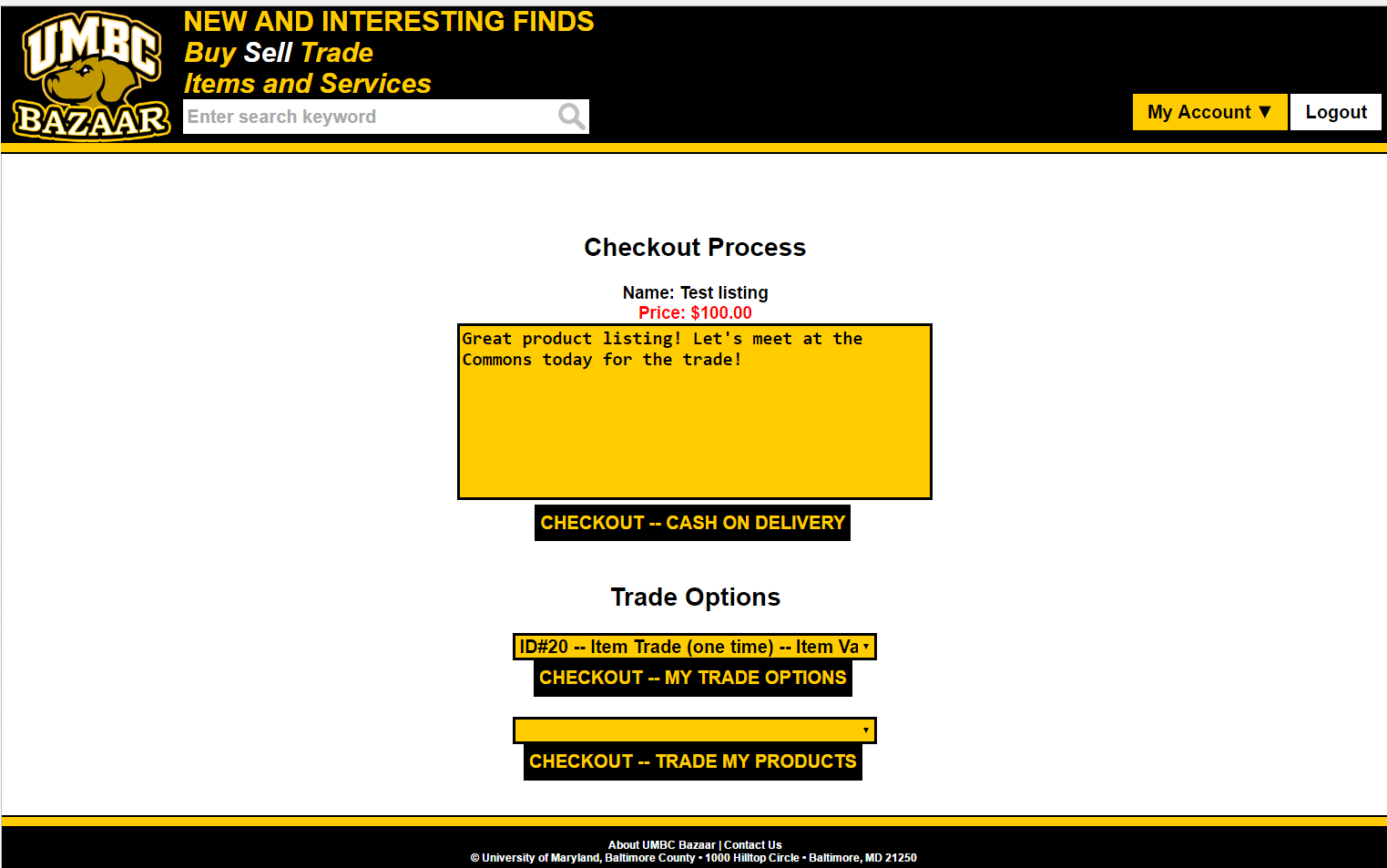


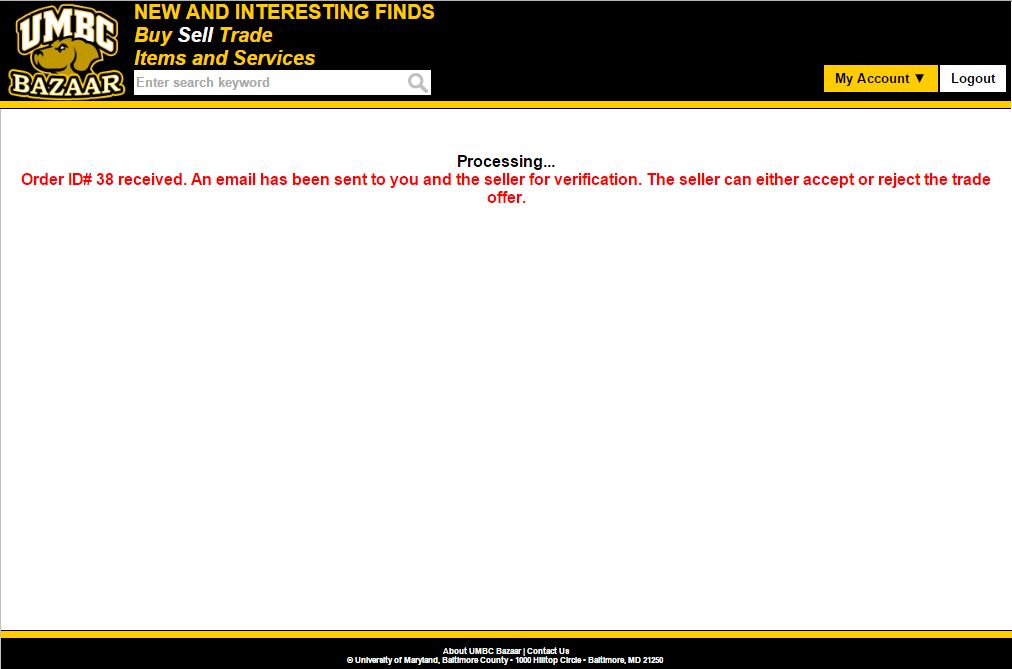
**3.2.9 Use Case: Checkout with trade options**

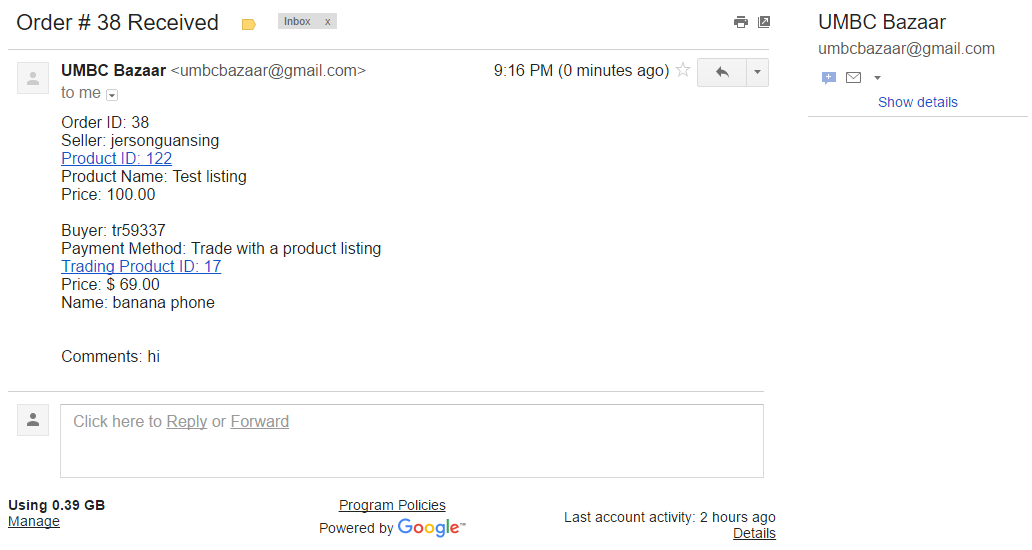
Conducted by: Jerson Guansing

**Valid Situation:**

The user has products to trade and checks out using trade successfully.

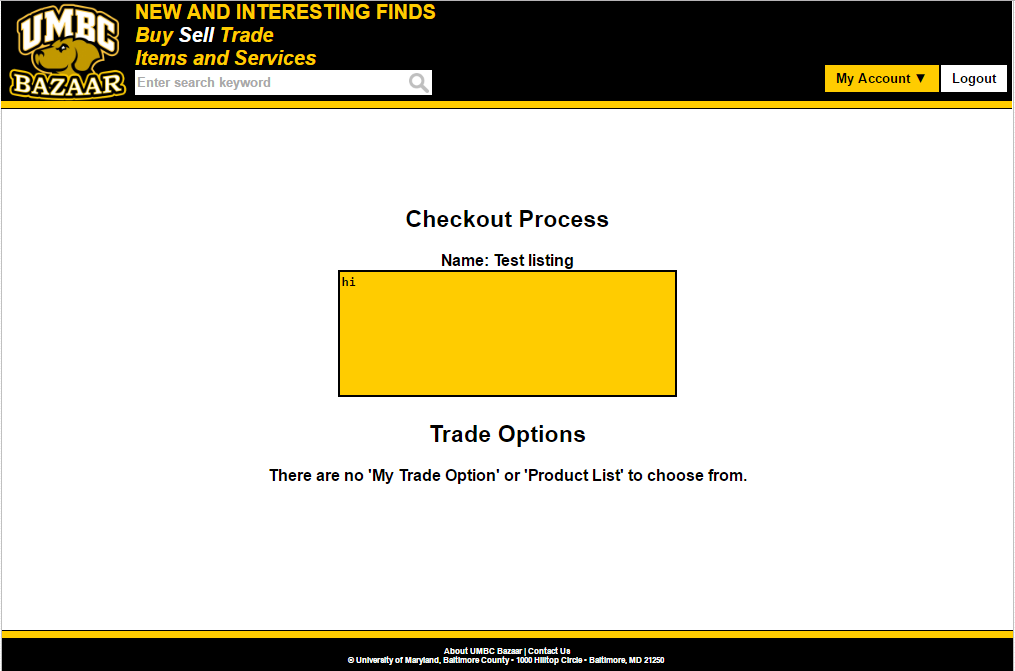






**Invalid Situation:**

The user has no items to trade with and thus will not be able to check out.



Checking out with invalid product id.



**Potential Improvements:**

* Searching by price

Sorting by price

**Appendix A – Team Review Sign-off**

This document has been collaboratively written by all members of the team. Additionally, all team members have reviewed this document and agree on both the content and the format. Any disagreement or concerns are addressed in team comments below.

Gene Burchette\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 12/01/2016

Gene Burchette

\_\_\_\_\_Jerson Guansing\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 12/01/2016

Jerson Guansing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date:

Karl Justice

\_\_Darren Stevens\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 12/01/2016

Darren Stevens II

\_Victor Wu\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: 12/01/2016

Victor Wu

**Appendix B – Document Contributions**

Gene - Added to Section 2.2 and Improvements

Jerson - Added to Sections 1, 2 and 3.

Victor - Added to Sections 1, 2 and 3.

Darren - Added to Section 3 and formatting