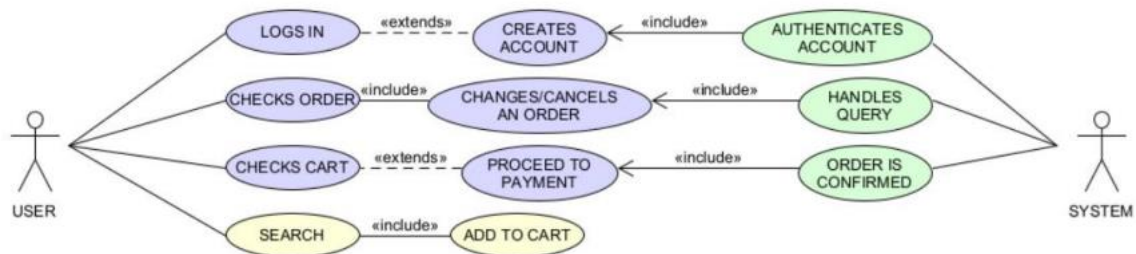# Requirements Testing

Use Case Diagram:

1: Login System:

The user is able to access the website and will be redirected to a login page. If the user has an account, they can enter the details on this page which will give them access to the site.

If the user does not currently have an account, they can click the register button to be taken to a registration page, where they can enter their details to be used as their login information.

This information is then stored in the database, which is what allows the user to immediately use the same details to login. Because of this, the user cannot register an account with a username that already exists in the database. This is to maintain security amongst accounts and to make authentication easier. Email addresses and phone numbers are naturally original. The user must follow certain requirements when creating a password (more than 5 characters, must include a special character) for security purposes.

Password must contain numbers

Beans and Machines

The system then authenticates if the user exists or not, by matching all of the data that the user entered against entries in the database. If the credentials exist, the user will be logged in, but if not, the user will get the opportunity to try again.

```php
public function getUserFromDatabase($username, $password) {
    try {
        // requiring the database connection file
        require_once '../src/DBconnect.php';

        $connection = DBconnect();

        // sql query for getting data from the database
        $query = 'SELECT * FROM users WHERE username = :username AND password = :password';

        // preparing and executing the query
        $stmt = $connection->prepare($query);
        $stmt->bindParam(":username", $username, PDO::PARAM_STR);
        $stmt->bindParam(":password", $password, PDO::PARAM_STR);
        $stmt->execute();

        // fetching
        $result = $stmt->fetch(PDO::FETCH_ASSOC);

        // assigning the database variables to php variables
        $this->username = $result["username"];
        $this->password = $result["password"];
        $this->email = $result["email"];
        $this->phoneNum = $result["phoneNum"];
        $this->isAdmin = $result["isAdmin"];

        // return true if username & password found, else return false
        if ($result) {
            return true;
        }
        else {
            return false;
        }
    }
    catch (PDOException $err) {
        echo $query . "<br>" . $err->getMessage();
    }
}
```

2: Cart System:

The user has a cart that is linked to their login session. Adding products to this cart will save those products in the cart until the user manually removes them, or until they can complete their order. The user can access the cart at any time on any page through a button placed in the top right of the screen. The cart will also display on the payments page, so that the user can view their contents and make any last-minute changes to the products they're purchasing.

Your Cart

Product Name	Product Price	Product Quantity	Total
Arabica Dark Roast	12.99	5	64.95
		Total	64.95

Payment Information

Card Number [　　　　　　　]
Expirary Date [dd / mm / yyyy 🗓]
CVV Number [　　　　　　　]
MasterCard ○ Visa ○
[Purchase]

Adding content to the cart can be done right by the products list for ease of use. Clicking "add to cart" on a product will add one of whatever product the user clicks this option on.

Products

Product Name		Product Description		Product Age		Product Vendor		Product Price		Product Stock	
Arabica Dark Roast	Caramel and walnut notes.	10	3fe	12.99	25	Add to Cart

When the user enters their payment information and clicks purchase, they get given a confirmation of their order with a summary of products. This also communicates to the system to remove the quantity of each product from the stock level of the same products using a function.

Your order has been placed for the following products:

Product Name	Product Price	Product Quantity	Total
Arabica Dark Roast	12.99	5	64.95
		Total	64.95

Thank you for your order!

[Return to Home]

```
public function updateStock($productID, $stock) {
    try {
        require_once "../src/DBconnect.php";
        require_once "../src/validation.php";

        $connection = DBconnect();

        $query = "UPDATE products SET productStock = :productStock WHERE id = :id";
        $statement = $connection->prepare($query);
        $statement->bindValue("productStock", $stock, PDO::PARAM_INT);
        $statement->bindValue("id", $productID, PDO::PARAM_STR);
        return $statement->execute();
    }
    catch (PDOException $err) {
        echo $query . "<br>" . $err->getMessage();
    }
}
```

Note: Although stated in the requirements, a delivery system is currently not being implemented in relation to payment.

3: Administrator Access:

A default administrator account is hard coded into the database as a way for admin users to be able to add, remove and edit products straight from the site. This makes it so that admin users do not need to have direct access to the database to be able to edit stock or product details.

Current Products

Product Name	Product Description	Product Age	Product Vendor	Product Price	Product Stock	Edit Product	Delete Product
Arabica Dark Roast	Caramel and walnut notes.	10	3fe	12.99	25	Edit Product	Delete Product
Decaf Blonde Roast	Floral flavour and texture.	12	Bad Habits	9.99	20	Edit Product	Delete Product

Add Product

Product Name

Product Description

Product Age

Product Vendor

Product Price

Product Stock

Add Product

The admin page can only be accessed if a user is logged in as an admin. The admin page cannot be forcefully directed to even if the user is logged in as a regular user it still directs them to the log in page.

Admins can also access the user database through the administrator page. More admin accounts can be created through this page, as well as editing or removing currently existing users.

Current Users

User Name	User Password	User Email	User Phone	Is User Admin	Edit User	Delete User
admin	admin	N/A	N/A	True	Edit User	Delete User
jake	Pass1!	jake@email.com	1234567890	False	Edit User	Delete User

Add User

User Name

User Password

User Email

User Phone

Admin?
☐
Add User

Finally, the admin can also access previous orders made by customers and then edit and remove these orders.

Order ID	User Name	Product Name	Quantity	Edit Order	Delete Order
3	jake	Arabica Dark Roast	5	Edit Order	Delete Order
4	jake	Decaf Blonde Roast	3	Edit Order	Delete Order

4: Products:

The user can access a products page once they are logged in. This will list all of the available products that the user can purchase. These products are being pulled directly from the database, in the case where that item is in stock. Additionally, the user can add a product their cart straight from this page, which will accurately deplete the products stock.

Products

ct Age	Product Vendor	Product Price	Product Stock	In Cart
3fe		12.99	29	1	Add to Cart

Products will stop displaying on the products page if their stock depletes to zero, to maintain the clean aesthetic of the site and to avoid confusion from customers. However, these products will still be visible regardless of stock count on the admin page, allowing administrators to update the stock when needed without needing to directly interact with the database.