## Criterion B - Design

User Interface/Switchboard Design

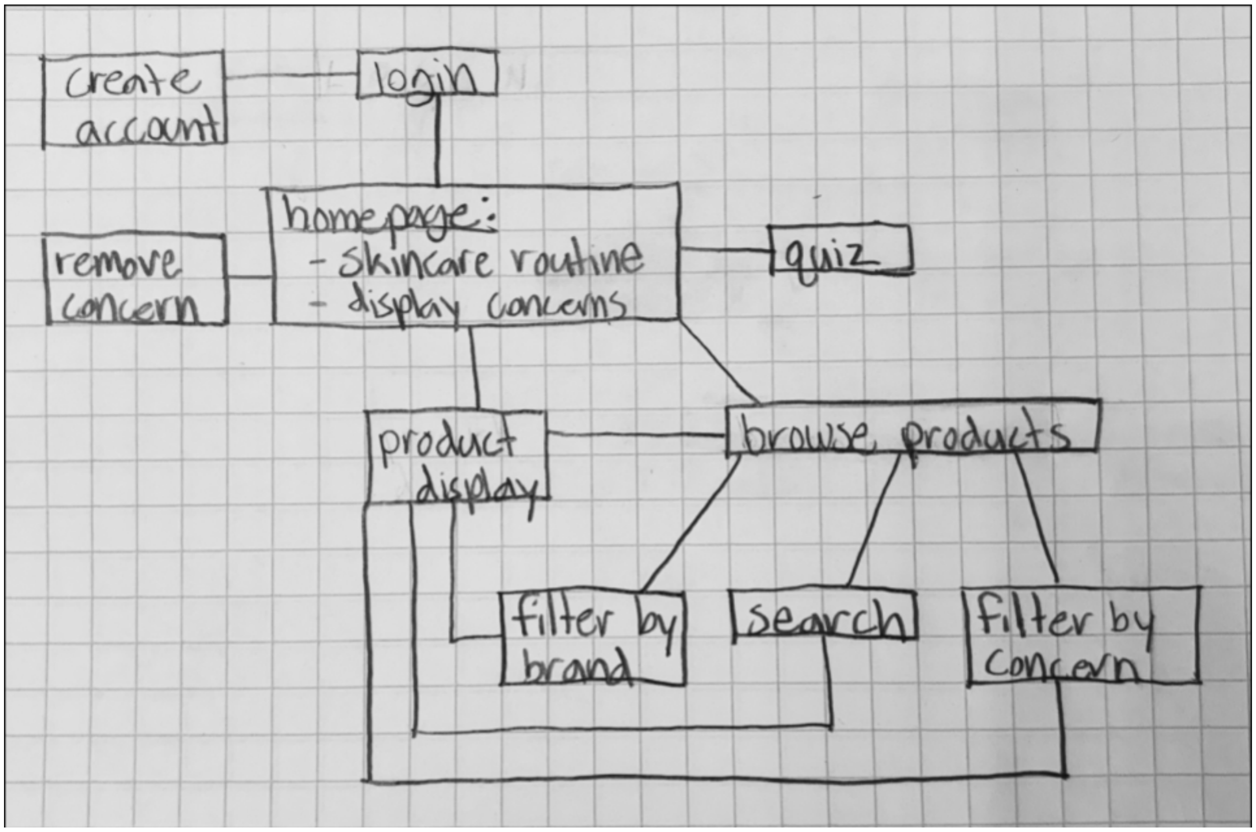


**Fig. 1: User Interface/Switchboard Design**

Hierarchy Diagram

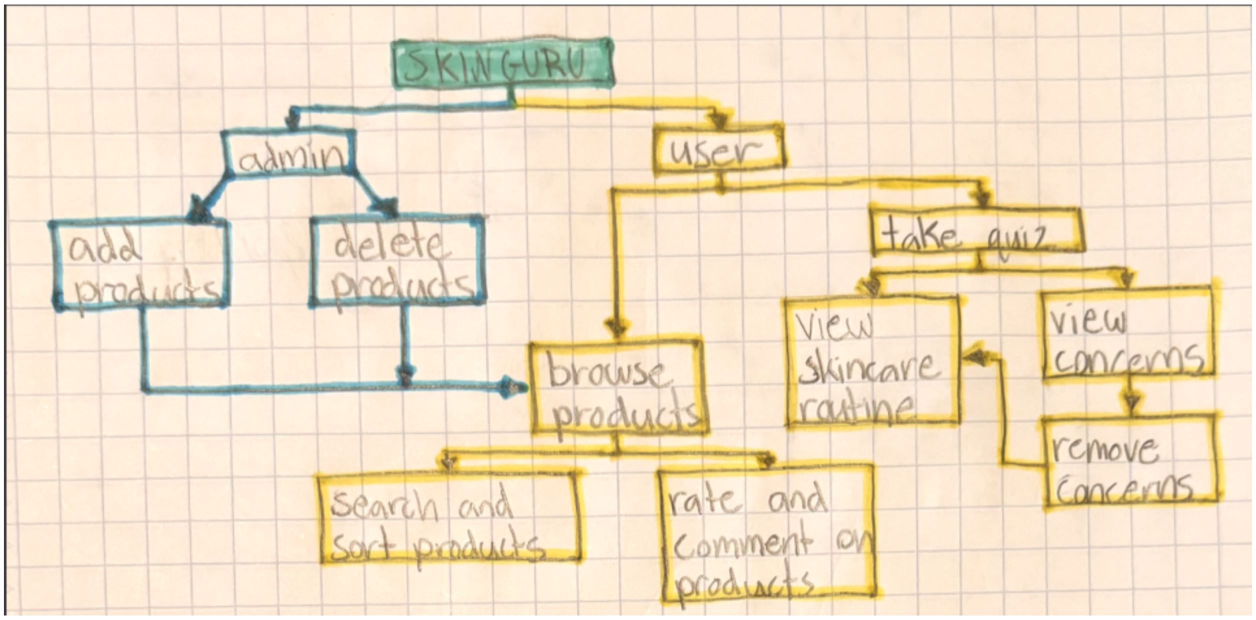


**Fig. 2: Hierarchy Diagram of the SKINGURU website**

## Brand Design

- Color Scheme: I chose this color scheme for the website because it appeals to users looking for a trustworthy place to find skincare products. My client specifically asked for a neat and clean-looking website, so I used a cool-toned, simple color palette and soft borders to give the user a sense of cleanliness as they browse the website.

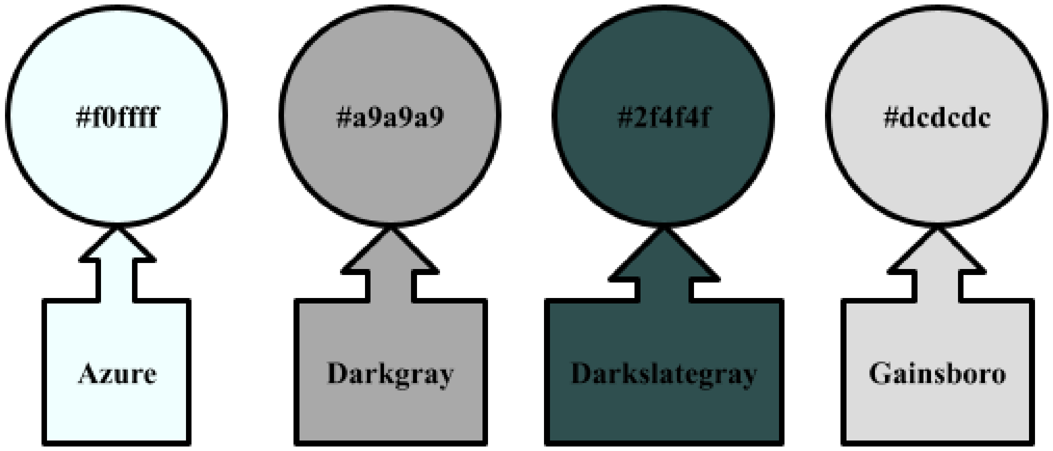


**Fig. 3: Color palate of the SKINGURU website**

- Font: I used Franklin Gothic Medium throughout the entire website for continuity and simplicity.
- Labels and Messaging: Most labels, buttons, and messages to the user are in capital letters because they are easy to read and emphasize important text. I use welcome messages and helpful information as the user logs in to ensure the website is both ‘friendly’ and user-friendly.

SKINGURU

**Fig. 4: Website Logo**

- Brand Logo and Name: I named the website ‘SKINGURU’ since it is a simple way to represent the purpose of the website, a skincare product recommendation platform. The logo is simplistic and uses the same font as the rest of the website to ensure neatness and consistency.

EAR Diagram

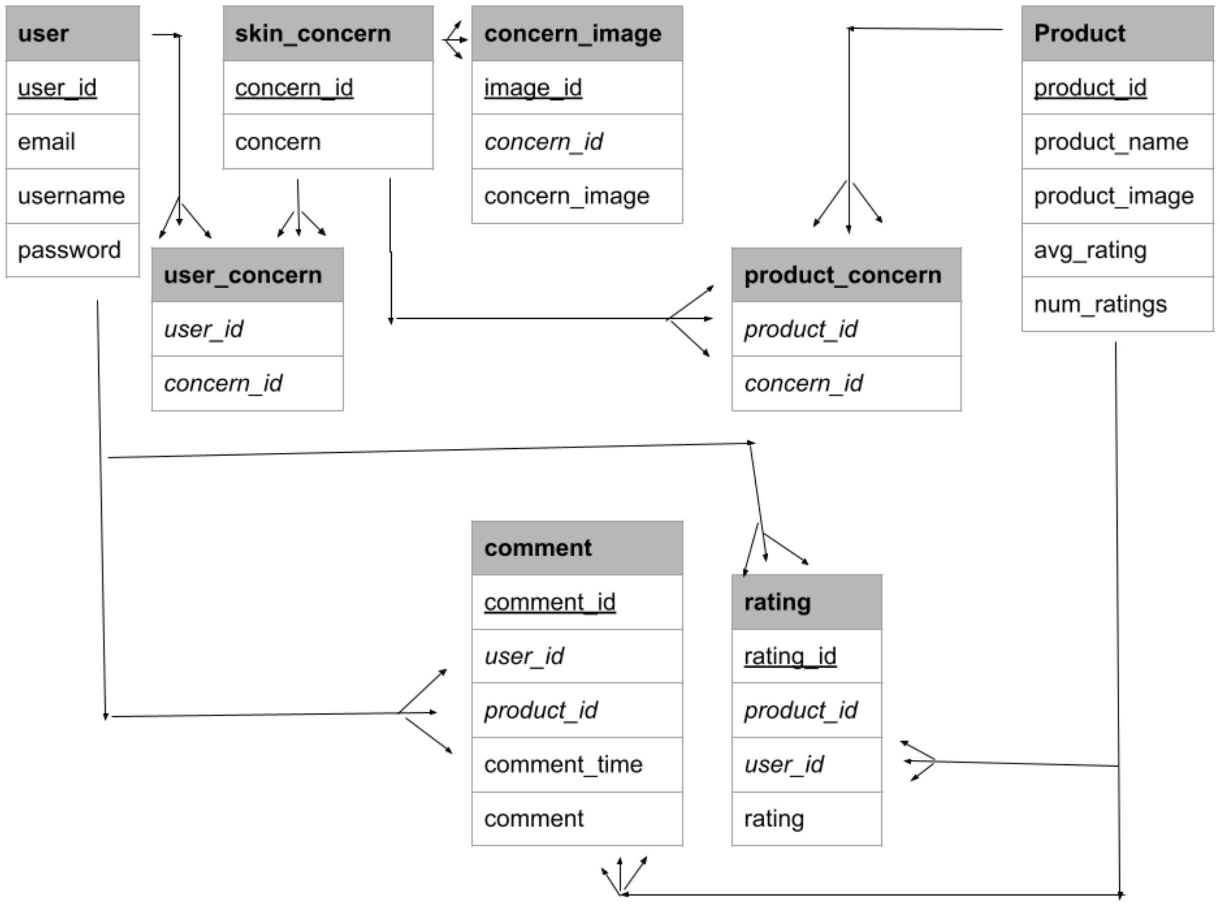
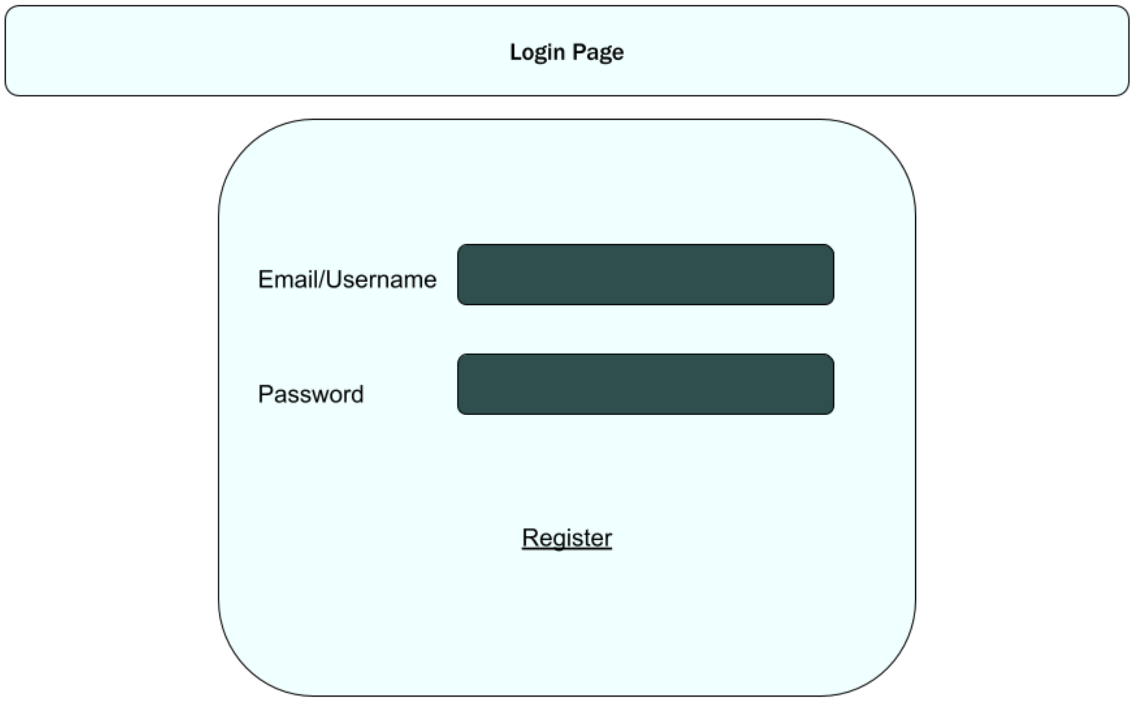


**Fig. 5: EAR Diagram**

## Prototype Screen Designs



The login page is essential for user authentication as well as making it possible for each user to have a custom experience with the website. Since each person has different skin concerns, users should have their own accounts so that they can take the quiz and see their personalized skincare routines.

**Fig. 6: Login page design**

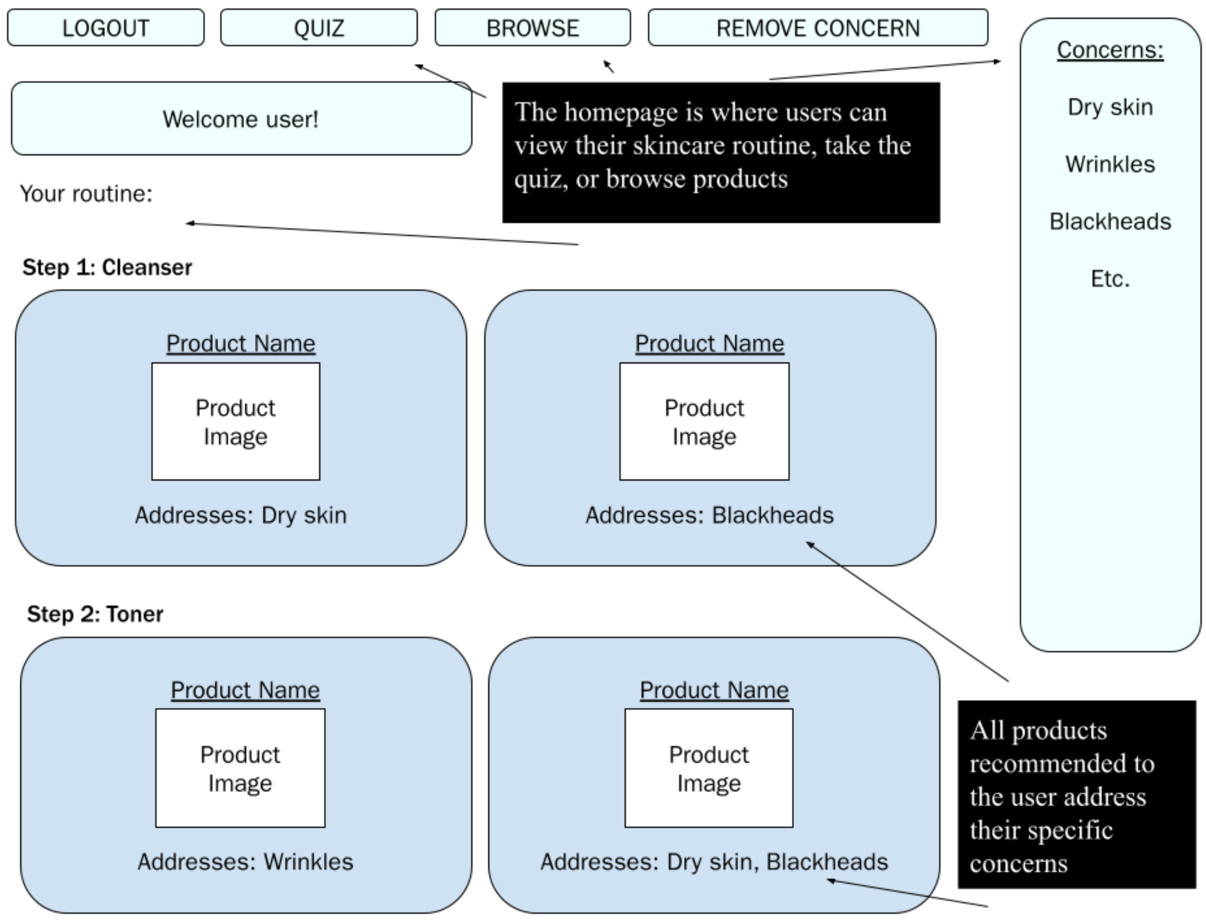


**Fig. 7: Home page design**

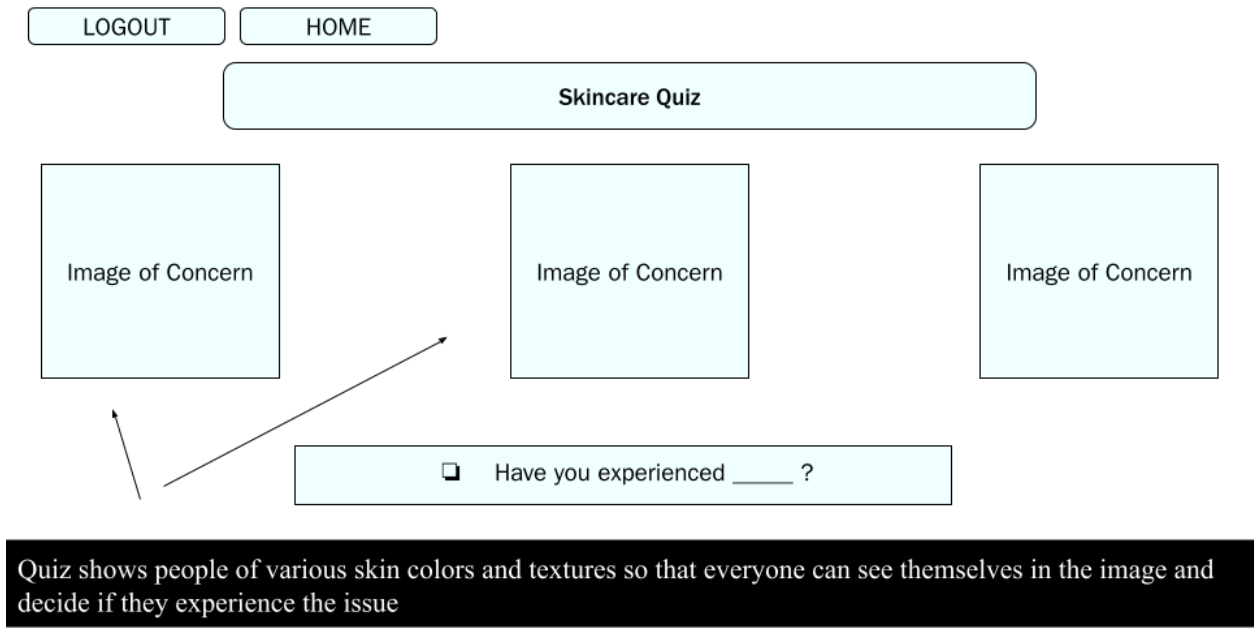

LOGOUT
HOME
Skincare Quiz
Image of Concern
Image of Concern
Image of Concern
❏ Have you experienced _____ ?
Quiz shows people of various skin colors and textures so that everyone can see themselves in the image and decide if they experience the issue

**Fig. 8: Skincare quiz design**

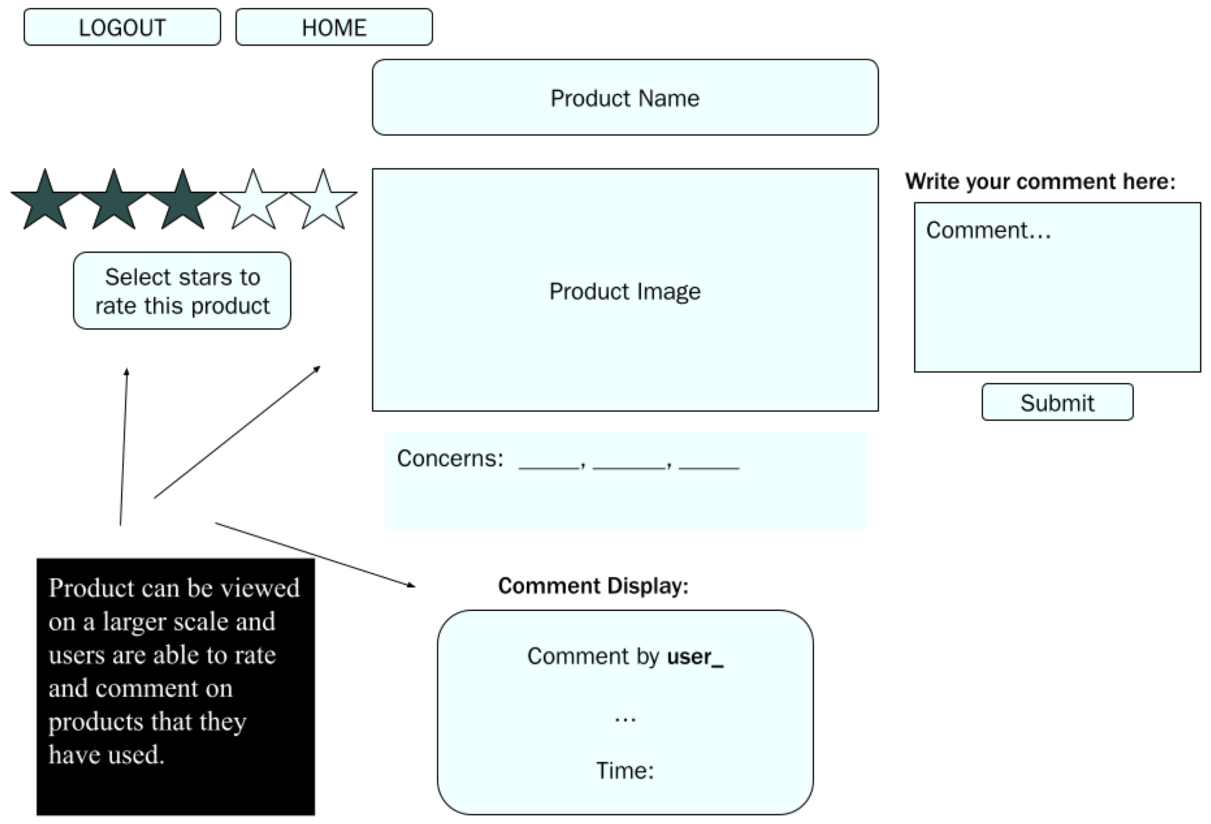

LOGOUT
HOME
Product Name
Select stars to rate this product
Product Image
Write your comment here:
Comment…
Submit
Concerns: _____, ______, _____
Product can be viewed on a larger scale and users are able to rate and comment on products that they have used.
Comment Display:
Comment by user_
…
Time:

**Fig. 9: Product info page design**

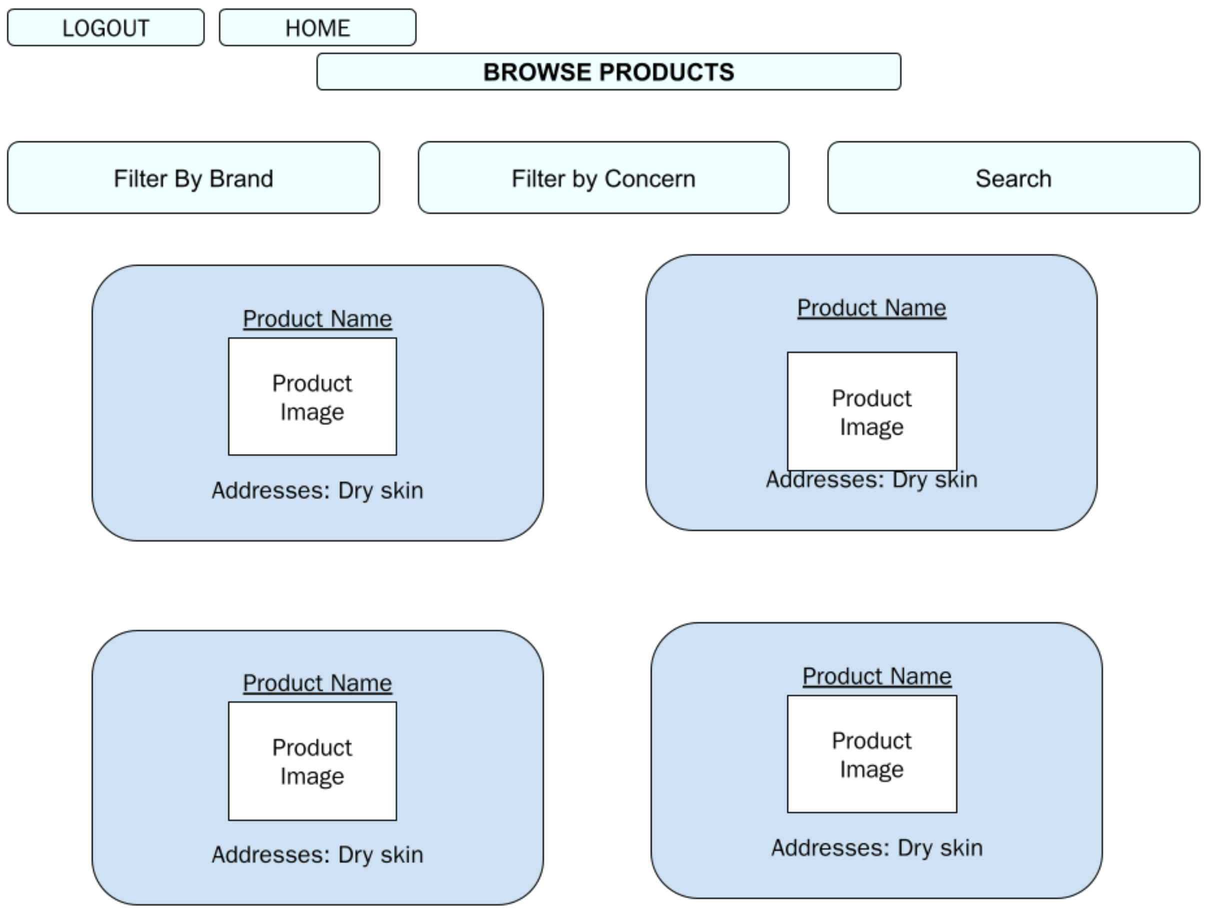


**Fig. 10: Browse products page design**

Products will also be ordered by highest rated.

## Algorithmic Design

❖ **Login Algorithm:**

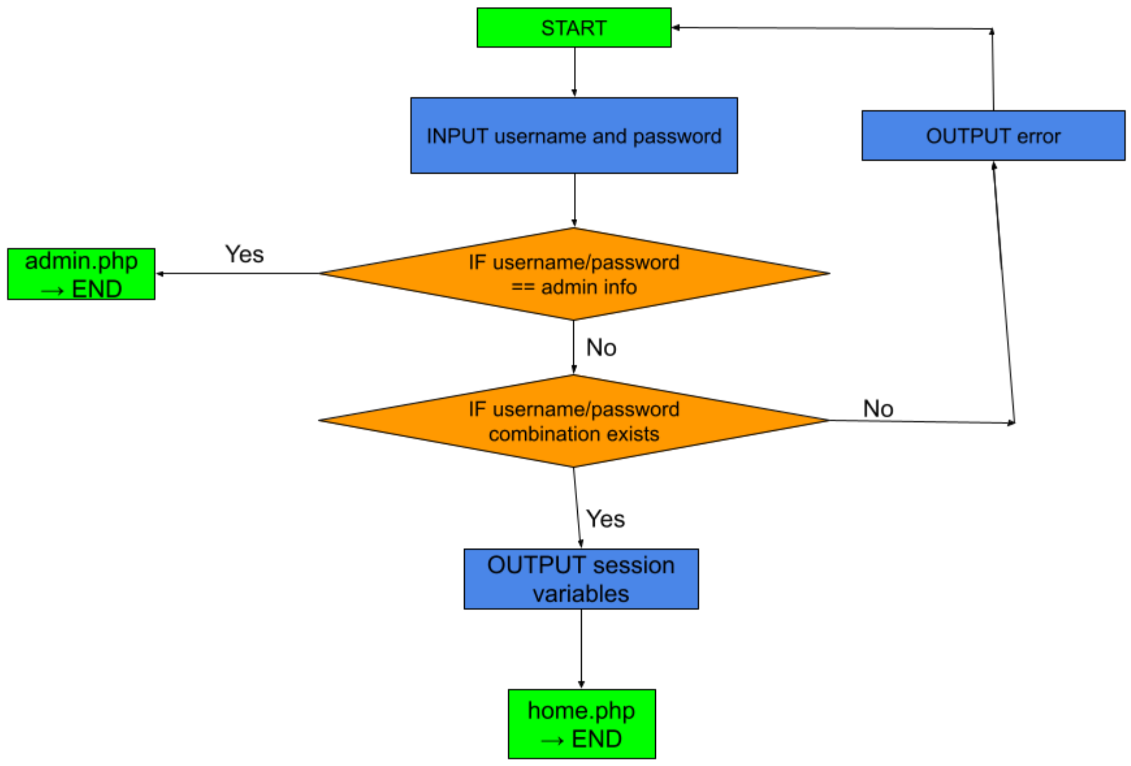

START
INPUT username and password
OUTPUT error
Yes
admin.php → END
IF username/password == admin info
No
IF username/password combination exists
No
Yes
OUTPUT session variables
home.php → END

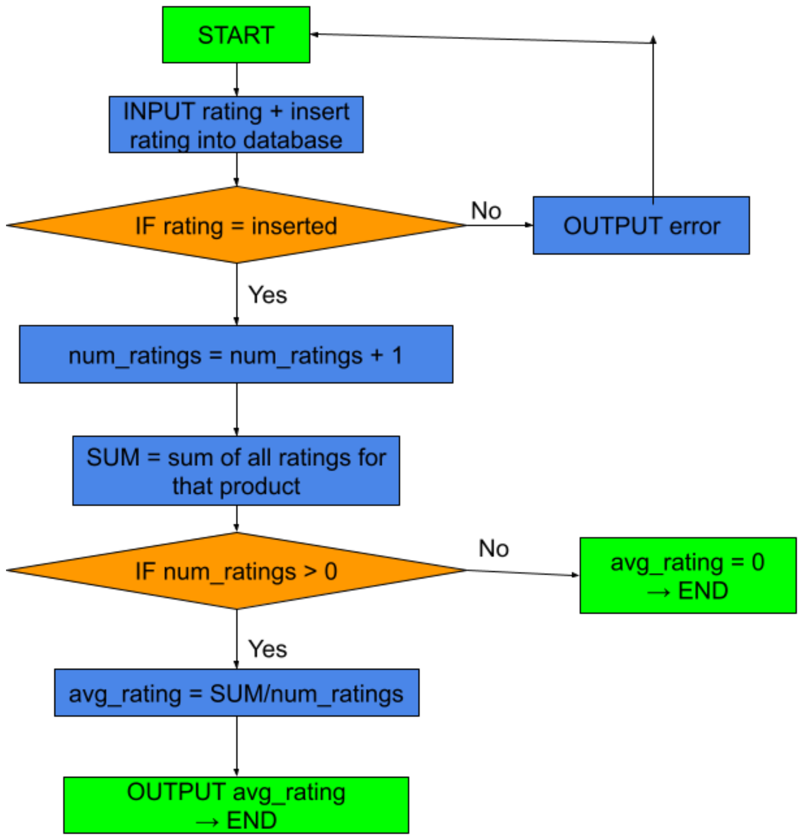
**Fig. 11: Login algorithm flowchart**

```
Start session

EMAIL = GET['email'];
PASS = GET['password'];

SQL = "SELECT * FROM user WHERE email = 'EMAIL AND password = 'PASS";
RESULT= result of SQL

IF RESULT > 0 then

        Login session = True

        $_SESSION variables = user_id, email, username

        IF $_SESSION['EMAIL'] == 'admin@admin.com' then
                Redirect user to 'admin.php'
        ELSE
```

```
                Redirect user to 'home.php'
        END IF

ELSE
        OUTPUT "Email or password is incorrect!"
        Redirect user to 'login.php'
END IF
```

The algorithm above authenticates the user, allowing them to login. It works by first accepting input from the user in the email/password input fields. It checks if the login information matches the admin login. If not, the algorithm checks if the username and password match with an existing user. Once the user is authenticated, the session variables are set, allowing the user to log in and be transferred to the homepage.

- **Rate Algorithm:**



START
INPUT rating + insert rating into database
IF rating = inserted
No
OUTPUT error
Yes
num_ratings = num_ratings + 1
SUM = sum of all ratings for that product
IF num_ratings > 0
No
avg_rating = 0 → END
Yes
avg_rating = SUM/num_ratings
OUTPUT avg_rating → END

**Fig. 12: Rating algorithm flowchart**

```
IF submit_rating = True then
        CHECKEDCOUNT = number of stars checked
        OUTPUT "You gave this product: " + CHECKEDCOUNT
```

```
PRODUCT_ID = fetched product id
USER_ID = $_SESSION['USER_ID']

RATING_SQL = "INSERT INTO rating (product_id, user_id, rating)
                 VALUES ('PRODUCT_ID, 'USER_ID', 'CHECKEDCOUNT')";
RATING_RESULT = result of RATING_SQL;

IF RATING_RESULT NOT found then
        OUTPUT "Rating insertion failed: "
END IF


TOTAL_RATINGS_SQL = "UPDATE product SET num_ratings = num_ratings + 1
                                 WHERE product_id = 'PRODUCT_ID";
 TOTAL_RATINGS_RESULT = result of TOTAL_RATINGS_SQL;

 IF TOTAL_RATINGS_RESULT NOT found then
          OUTPUT "Total ratings update failed: "
 END IF

  SUM_SQL = "SELECT SUM(rating) AS total_sum FROM rating
                 WHERE product_id = 'PRODUCT'";
  SUM_RESULT = result of SUM_SQL

  NUM_RATINGS_SQL = "SELECT num_ratings FROM product
                           WHERE product_id = 'PRODUCT_ID";
  NUM_RATINGS_RESULT = result of NUM_RATINGS_SQL

  IF NUM_RATINGS_RESULT > 0 then
        AVG_RATING = SUM_RESULT['total_sum'] /
                             NUM_RATINGS_RESULT['num_ratings'];
        UPDATE_RATING_SQL= "UPDATE product SET avg_rating ='AVG_RATING'
                                 WHERE product_id = 'PRODUCT_ID";
        UPDATE_RATING_RESULT = result of UPDATE_RATING_SQL

        IF UPDATE_RATING_RESULT = False then
                OUTPUT "Updating average rating failed: "
        END IF
        ELSE
                AVG_RATING = 0
        END IF
ELSE
        OUTPUT "Please select a rating.";
END IF
```

The algorithm above allows users to submit a rating into the database and then calculate the average rating for that product. It takes user input, where they can rate a product out of five stars, and then inserts information into the database. The total number of ratings is increased by one so that it can be used for the sum of all ratings and, subsequently, the calculation for the average rating. If the product has no ratings, the average rating is set to zero until more than one rating is inputted to avoid dividing by zero and causing an error.

- **Skincare Routine Maker Algorithm:**

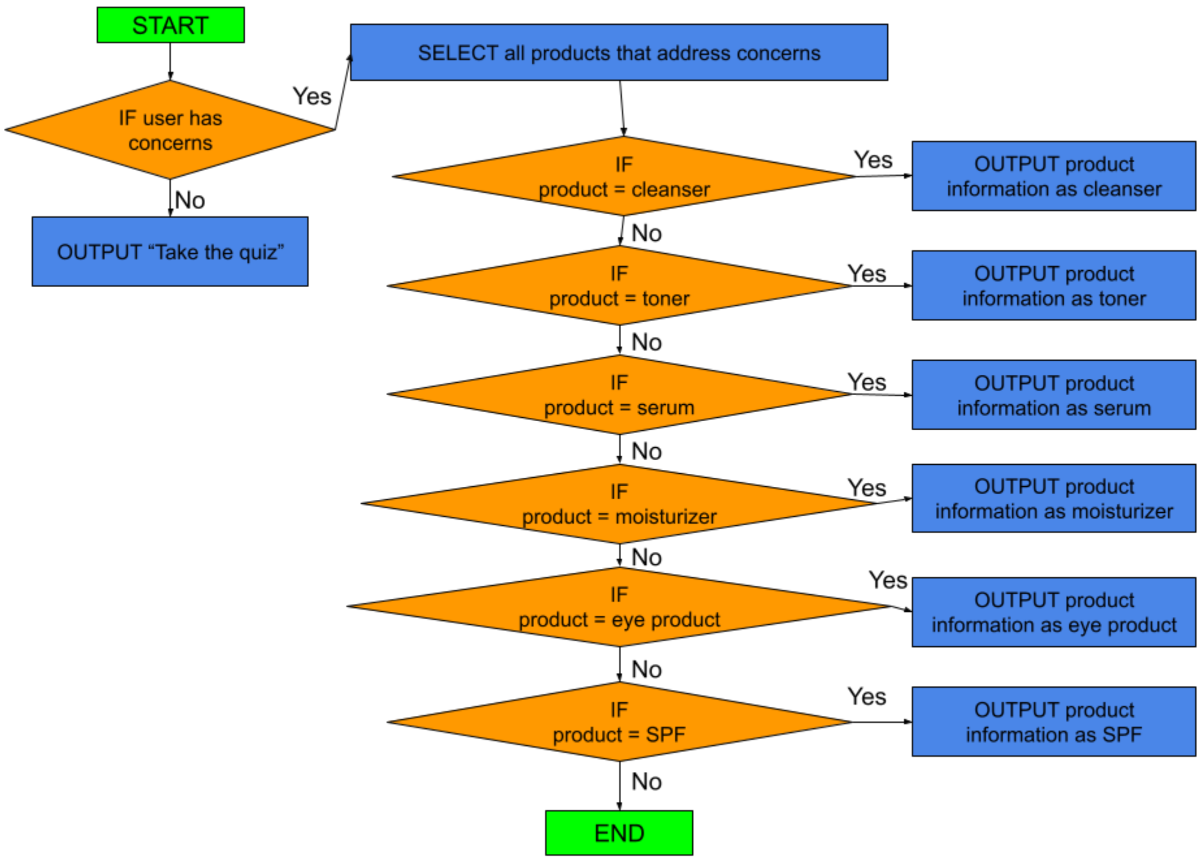

START
IF user has concerns
Yes
No
OUTPUT "Take the quiz"
SELECT all products that address concerns
IF product = cleanser
OUTPUT product information as cleanser
IF product = toner
OUTPUT product information as toner
IF product = serum
OUTPUT product information as serum
IF product = moisturizer
OUTPUT product information as moisturizer
IF product = eye product
OUTPUT product information as eye product
IF product = SPF
OUTPUT product information as SPF
END

**Fig. 13: Skincare routine maker algorithm flowchart**

<u>Example of 'cleanser' since the code is repeated for each step:</u>

CLEANSER_SQL = "SELECT product.product_name, product.product_image,

```
                              GROUP_CONCAT(DISTINCT user_concern.concern_id)
                              AS concern_ids,
                              GROUP_CONCAT(DISTINCT skin_concern.concern)
                              AS concerns, product_concern.product_id
                              FROM product_concern
                              INNER JOIN user_concern
                              ON product_concern.concern_id = user_concern.concern_id
                              INNER JOIN product
                              ON product_concern.product_id = product.product_id
                              INNER JOIN skin_concern
                              ON product_concern.concern_id = skin_concern.concern_id
                              WHERE product.product_name LIKE '%cleanser%'
                              AND user_concern.user_id = '$user_id'
                              GROUP BY product.product_name, product.product_image,
                              product_concern.product_id
                              ORDER BY avg_rating DESC";

CLEANSER_RESULT = result of CLEANSER_SQL

LOOP WHILE CLEANSER_ROW = CLEANSER_RESULT
      OUTPUT "---product display---";
END LOOP
```

The algorithm above sorts products by usage, rating, and concerns they address. The product type is deduced by using a 'LIKE' statement to find product names similar to the product usage. Products are ordered by rating using an 'ORDER BY' statement to display the best-reviewed products first.

# Data Dictionary

**User**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| User_ID | Int | Auto Incriment | 11 | Primary key to identify the user | 3 |
| Email | Varchar | N/A | 100 | Email for the user to enter during login | user@gmail.com |
| Username | Varchar | N/A | 100 | Username used to welcome the user on the homepage | user_name |
| Password | Varchar | N/A | 100 | Gives privacy to user so that other people can't login to their account | password123 |

**Skin Concern**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| Concern_ID | Int | Auto Incriment | 2 | Primary key to identify the concern | 6 |
| Concern | 12 | N/A | 100 | The name of the concern | Dry skin |

**User Concern**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| *User_ID* | Int | N/A | 11 | Forgein key to connect the user with a concern | 9 |
| *Concern_ID* | Varchar | N/A | 11 | Forgein key to identify the concern connected with | 3 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | the user | |

**Product**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| Product_ID | Int | Auto Incriment | 10 | Primary key to identify the product | 3 |
| Product Name | Varchar | N/A | 100 | The name of the product | Cera Ve Serum |
| Product Image | Varchar | N/A | 1000 | Contains the URL to the product image so that it can be called to display the product | https://m.media-amazon.com/images/I/31j+VyDf+zL._SY300_SX300_.jpg |
| Avg_Rating | Int | Default is 0 | 10 | Contains the average rating for a product | 4.3 |
| Num_Ratings | Int | Default is 0 | 10 | Contains the total number of ratings for a product so that the average can be calculated | 9 |

**Product Concerm**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| *Product_ID* | Int | N/A | 11 | Forgein key to identify the product | 3 |
| *Concern_ID* | Int | N/A | 11 | Forgein key to identify the concern | 6 |

**Concern Image**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| Image_ID | Int | NULL | 2 | Primary key to identify the image | 3 |
| *Concern_ID* | Int | NULL | 2 | Forgein key to connect the image with the concern | 3 |
| Concern_Image | Varchar | NULL | 193 | Contains the URL to the concern image so that it can be called to display the concern | https://i.redd.it/83lzbng4o6h01.jpg |

**Rating**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| Rating_ID | Int | Auto Incriment | 11 | Primary key to identify the rating | 3 |
| *Product_ID* | Int | N/A | 11 | Forgein key to connect the rating with the product | 6 |
| *User_ID* | Int | N/A | 11 | Forgein key to connect the user with their rating | 9 |
| Rating | Int | N/A | 11 | The rating from the user's input | 3 |

**Comment**

| **Field Name** | **Data Type** | **Data Format** | **Field Size** | **Description** | **Example** |
|---|---|---|---|---|---|
| Comment_ID | Int | Auto Incriment | 11 | Primary key to identify the comment | 3 |
| *User_ID* | Int | N/A | 11 | Forgein key to | 6 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | connect the user with the comment | |
| *Product_ID* | Int | N/A | 11 | Forgein key to connect the comment with their product | 9 |
| Comment_Time | Timestamp | Current Timestamp | N/A | The time that the comment was submitted | January 24, 2024, 3:00 PM |
| Comment | Varchar | N/A | 500 | The text field containing the comment | I love this product! |

## SQL Query Design

**Skincare routine step query:**

*This query filters skincare products for the user's routine, and all selected products are ordered by average rating.*

```
SELECT product.product_name, product.product_image,
GROUP_CONCAT(DISTINCT user_concern.concern_id)
        AS concern_ids,
GROUP_CONCAT(DISTINCT skin_concern.concern)
        AS concerns, product_concern.product_id
FROM product_concern
INNER JOIN user_concern
        ON product_concern.concern_id = user_concern.concern_id
INNER JOIN product
        ON product_concern.product_id = product.product_id
INNER JOIN skin_concern
        ON product_concern.concern_id = skin_concern.concern_id
WHERE product.product_name LIKE '%sun%'
AND user_concern.user_id = '$user_id'
GROUP BY product.product_name, product.product_image, product_concern.product_id
ORDER BY avg_rating DESC;
```

**Search products query:**

*This query is used to search for specific products or concerns based on user input, the products are then ordered by highest rating.*

```
SELECT product.product_id, product.product_name, product.product_image,
product.avg_rating,
GROUP_CONCAT(DISTINCT skin_concern.concern)
        AS concerns
FROM product_concern
INNER JOIN product
        ON product_concern.product_id = product.product_id
INNER JOIN skin_concern
        ON product_concern.concern_id = skin_concern.concern_id
WHERE product.product_name LIKE '%$search%'
OR skin_concern.concern LIKE '%$search%'
GROUP BY product.product_id, product.product_name, product.product_image
ORDER BY product.avg_rating DESC;
```

**Admin function - Adding a product query:**

*This query is used by the administrator to add new products to the database. Data is inserted into the product table, allowing the product ID to be used in the product_concern table.*

```
INSERT INTO product (product_name, product_image)
VALUES ('$product_name', '$product_image')";

SELECT * FROM product WHERE product_name = '$product_name';

INSERT INTO product_concern (concern_id, product_id)
VALUES ('$concern_id', '$product_id');
```

# Data Flow Diagrams

**Login System:**

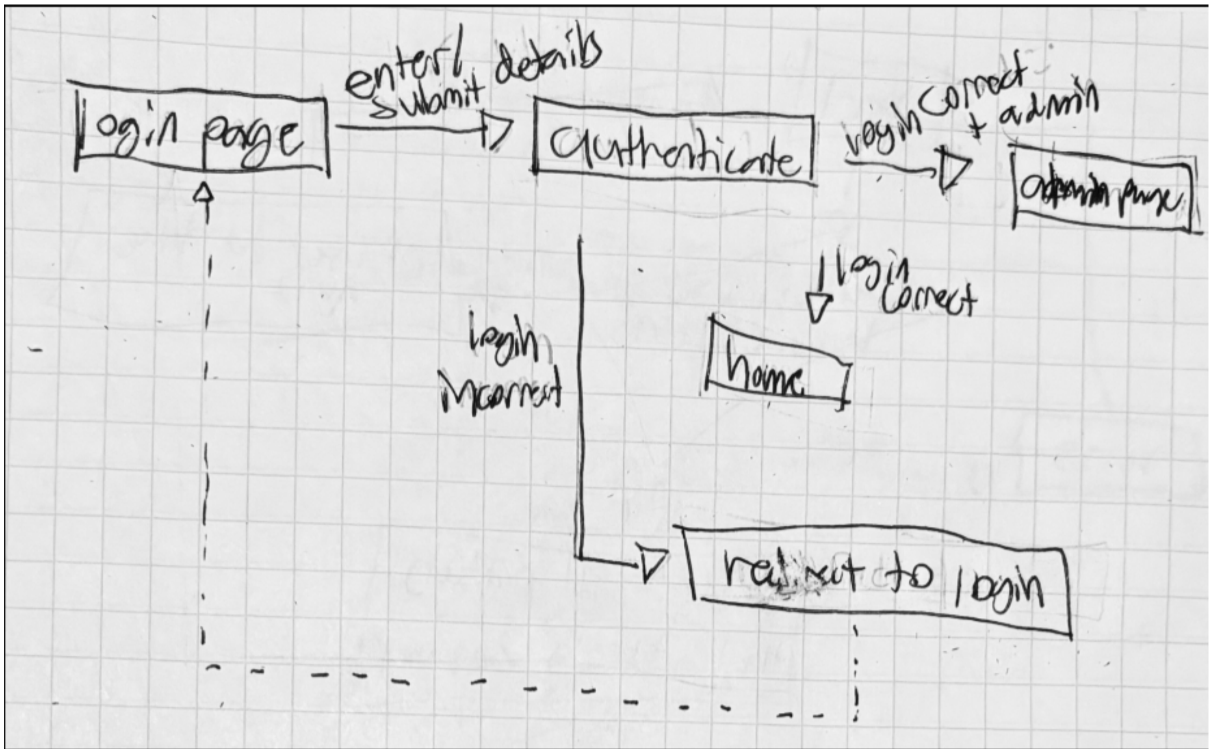


**Fig. 14: Login authentication system**

**Quiz Submission:**

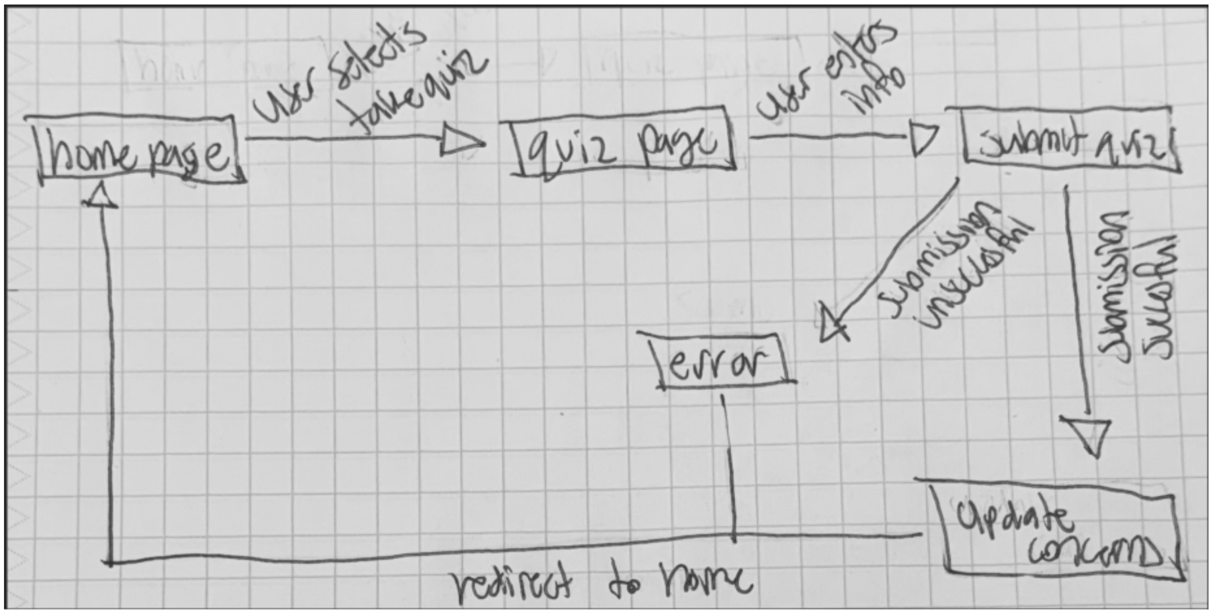


**Fig. 15: Quiz submission and entry system**

**Rate/Comment Submission:**

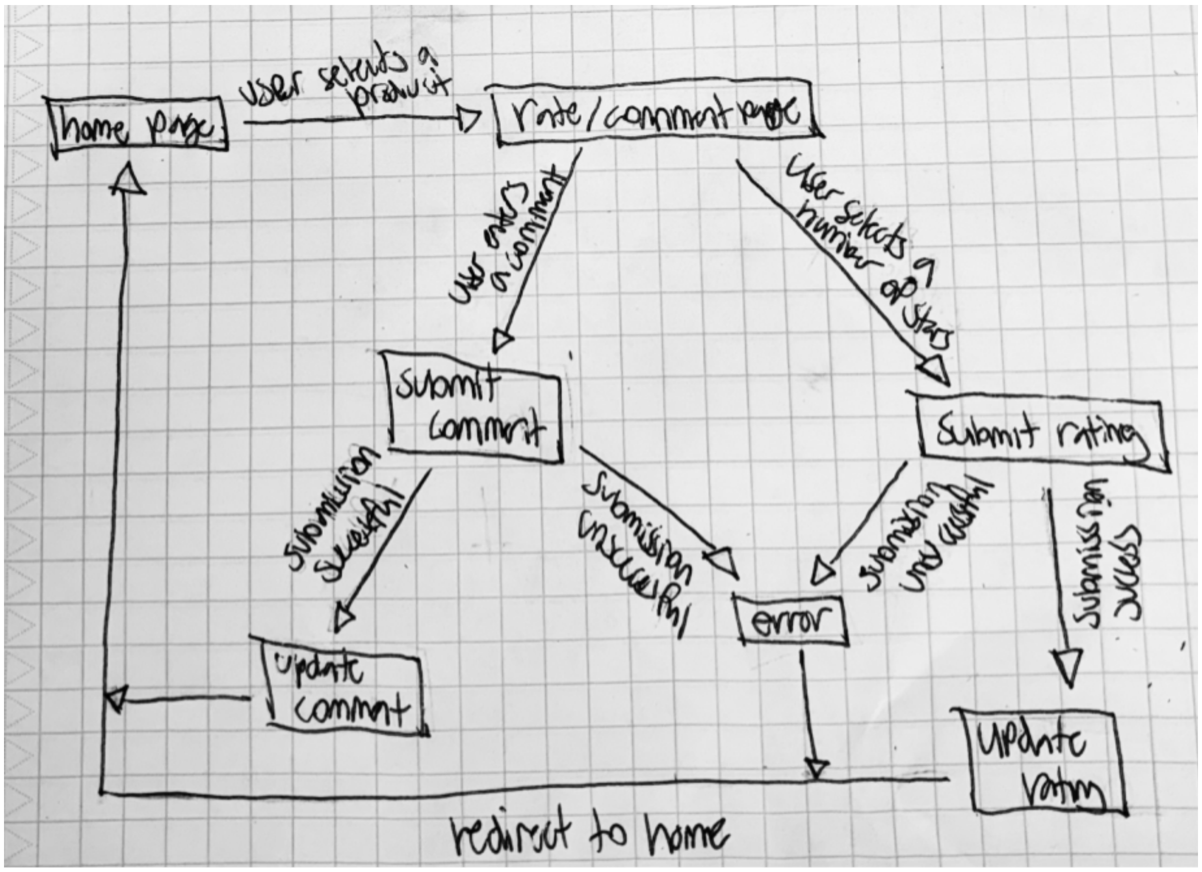


**Fig. 16: Rate/comment entry and submission system**

## Test Plan

| Test Number | Success Criteria | Description of Test | Expected Outcome |
|---|---|---|---|
| 1 | 1 | Use an existing account to log in to the website | The user will be logged in |
| 2 | 2 | Use register functionality to create a new account | A new account will be shown in the database upon submission |
| 3 | 3 | Login to the website with incorrect information | An error message is displayed when incorrect information is entered |
| 4 | 4 | Login to website with a new account | There should be no products displayed for the user's skincare routine and a message should tell the user to take the quiz |
| 5 | 5, 7, 8 | Submit the quiz for a new user account | Upon submission, the website should give the user a skincare routine and display the concerns they submitted on the home page |
| 6 | 6 | Remove concerns from an existing account with submitted concerns | The user should be able to select the concerns they want to remove and submit the form, causing their profile to be updated |
| 7 | 9, 10 | Check that products are in order of highest rated and that the product display includes all the necessary details | The products are sorted by rating and the display includes the product name, image, rating, and concerns that it addresses |
| 8 | 11, 12 | Comment on a product | Product comment is uploaded and displayed on product page |
| 9 | 13 | Rate a product | Rating is updated into the database and the average rating is updated |
| 10 | 14, 15, 16 | Use the search and filter functions | The search and filters select the correct products based on the conditions provided by the user |

| | | | |
|---|---|---|---|
| 11 | 17, 18 | Login to admin and add/remove a product from the database | The product is successfully added to the database, then the product is successfully removed from the database |

## Modified Screen Designs

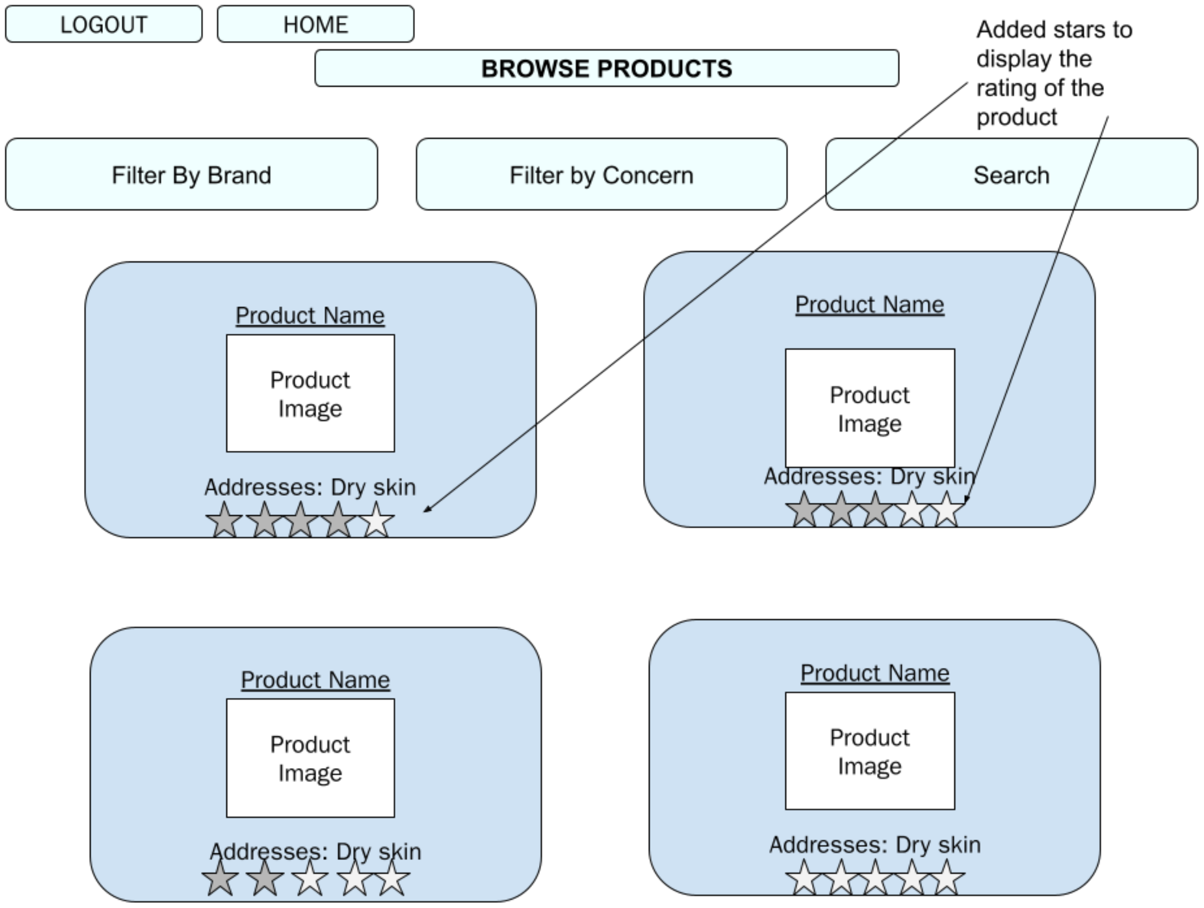


**Fig. 17: Modified Screen Design**

<u>Word count: 447</u>