**Team 5 Test Cases**

Student list:

**Assignment Details**

As discussed in our Testing Module, I would like you all to work on your next phase of your project by creating test plan and test cases.

Here is what I am expecting you to do. This is a collective assignment, but you need to work individually and submit one test case document per team.

1. Each individual should pink up 2 use cases and write test cases for those use cases.

2. Each test case should clearly prove what you are testing, what is expected, what are the inputs.

3. Each test case should have a testing procedure as steps to facilitate the user what to test.

eg. If you expect me to test adding or 2 numbers you need to mention

      a) Input any number in text box 1

      b) Input any number in text box 2

      c) click on the calculation symbol (+,-,x,%)

     d) click answer button to see the result

4. Record the "observed" result

5. Compare the "Expected" vs "Observed" Result.

6. Determine the test passed or failed.

Nga Vu (1,2)

Bao Le (3,4)

Aleeza Iftikhar(5,6)

Jisoo Park (7,8)

Guillermo Clara (9,10)

1. **Use case 1 - Nga Vu = PASS**

| Use case name | Making Appointment |
| --- | --- |
| Participating actor | User, User database |
| Entry condition | Click “make appointment” in the salon website. |
| Flow of events | 1. User select the salon on the main homepage, 2. User view services and “make an appointment” button 3. User search for available appointment 4. User fill in their information and click “make appointment” |
| Exit condition | Users submit and see the appointment confirmation page. |

| Test-case identifier | Making Appointment |
| --- | --- |
| Test location | /appointment |
| Feature to be tested | To see if application is able to save the selected appointment |
| Feature Pass/Fail Criteria | Pass: Database able to save user’s selected available time  Fail: Database failed to save user’s input |
| Means of control | “Make appointment” calling /api/make\_appointment” api route |
| Data | A POST message with the client’s and selected appointment information |
| Test Procedure | 1. User select the salon on the main homepage 2. User click “book now” button and click “make an appointment” button. 3. User pick through services, stylist, address, and date then click search 4. User select an available time slot 5. User fill in their information 6. User click “Make Appointment” |
| Special requirements | Any user |

1. **Use case 2 - Nga Vu = PASS**

| Use case name | View Appointment |
| --- | --- |
| Participating actor | Business Owner |
| Entry condition | Business owner login with the Owner nav link |
| Flow of events | 1. User click “View Appointments” button 2. The page display the calendar and filter |
| Exit condition | Users go back to the previous page or click on the ”home” button. |

| Test-case identifier | View Appointment |
| --- | --- |
| Test location | /schedule |
| Feature to be tested | Database is able to show the list of available appointments on calendar |
| Feature Pass/Fail Criteria | Pass: able to see scheduled appointments  Fail: unable to see scheduled appointments |
| Means of control | callApi()called after schedule page is loaded |
| Data | Data from /api/schedule controller containing appointment detail data |
| Test Procedure | 1. User with business owner access click “owner login” and insert their email and password on the login box. Then click “sign in”. 2. User click “view appointments” on the business dashboard 3. The business owner will see the weekly calendar that shows the scheduled appointments. 4. Business owner can select any week on the calendar picker. 5. Business owner can select any scheduled appointment to see the detail. |
| Special requirements | The user have the business owner role |

1. **Use case 3 = PASS**

| Use case name | Cancel appointment |
| --- | --- |
| Participating actor | User, User database |
| Entry condition | Click “view appointment” in the user account center. |
| Flow of events | 1. User successfully login his/her account. 2. Users go to the account menu and click “view appointment”. |
| Exit condition | Users go back to the previous page/ click on the home page icon. |

| Test-case identifier | Cancel appointment |
| --- | --- |
| Test location | User\_account/history page |
| Feature to be tested | User databases are able to show his/her reservation appointment to users. |
| Feature Pass/Fail Criteria | The test passes if users can see their recent appointment and click to cancel the appointment. |
| Means of control | The function view appointment() is called when users click on the “cancle” button. |
| Data | 1. Users should be able to view his/her cancle appointment when he/she has at least one appointment at least one time before. |
| Test Procedure | 1. Login 2. Go to account page 3. Then “cancle appointment” 4. And check whether any data shows up. |
| Special requirements | The users are logged in. |

1. **Use case 4** **= PASS**

| Use case name | View confirmation |
| --- | --- |
| Participating actor | User - customer |
| Entry condition | Click “view confirmation” in the user account center. |
| Flow of events | 1. User successfully login his/her account. 2. Users go to the account menu and click “view confirmation”. |
| Exit condition | Users go back to the previous page/ click on the home page icon. |

| Test-case identifier | View confirmation |
| --- | --- |
| Test location | User\_account/history page or email link |
| Feature to be tested | User databases are able to show his/her confirmation to users. |
| Feature Pass/Fail Criteria | The test passes if users can see their recent confirmation in their email |
| Means of control | The function view confirmation is called when users click on the “confirmation appointment” button. |
| Data | 1. Users should be able to view his/her confirmation when he/she has at least one appointment at least one time before. |
| Test Procedure | 1. Login 2. Go to account page 3. Then “view confirmation appointment” 4. And check whether any data shows up. |
| Special requirements | The users are logged in. |

1. **Use case 5 - Aleeza Iftikhar = PASS**

| Use case name | **List your Business** |
| --- | --- |
| Participating actor | Salon Owner, Salon Manager |
| Entry condition | Click “Business Portal” in petit homepage. |
| Flow of events | 1. Enter account information and business information (username, email address, phone number, password, and description) in respective text fields. 2. Click Sign Up button. |
| Exit condition | Users go to login page. |

| Test-case identifier | **List your Business** |
| --- | --- |
| Test location | http://localhost:3000/sign-up |
| Feature to be tested | Salon owners can list their business on petit website. |
| Feature Pass/Fail Criteria | The test passes if users can successfully login. |
| Means of control | handleSubmit() function is called which has POST method inside it, it updates database by adding new business object with unique id. |
| Data | Once users sign up, they will receive an email verifying their successful account creation. |
| Test Procedure | 1. Enter account information and business information (username, email address, phone number, password, and description) in respective text fields. 2. Click Sign Up button. |
| Special requirements | Valid email address is required to sign up. |

1. **Use case 6 - Aleeza Iftikhar = PASS**

| Use case name | **Edit Employees** |
| --- | --- |
| Participating actor | Salon Owner, Salon Manager |
| Entry condition | First login to business dashboard and then click “Employees” in top navigation bar. |
| Flow of events | 1. On petit homepage, click Business Portal. 2. On sign up page, click login. 3. On login page, enter email address and password, and click login button. 4. On Business dashboard, click employees. 5. To add an employee, enter employee’s name, phone number, email address. Then click add new employee button and click save changes button. 6. To remove an employee, click the remove button and then click save changes. |
| Exit condition | List of employees is updated. |

| Test-case identifier | **Edit Employees** |
| --- | --- |
| Test location | http://localhost:3000/employees |
| Feature to be tested | Salon managers can update list of employees. |
| Feature Pass/Fail Criteria | The test passes if users can see updated list of employees. |
| Means of control | The function handlesubmit() is called when users click on the “save changes” button, it has PUT function inside it which updates list of employees in database. |
| Data | Database will now have updated list of employees. |
| Test Procedure | 1. On petit homepage, click Business Portal. 2. On sign up page, click login. 3. On login page, enter email address and password, and click login button. 4. On Business dashboard, click employees. 5. To add an employee, enter employee’s name, phone number, email address. Then click add new employee button and click save changes button. 6. To remove an employee, click the remove button and then click save changes. |
| Special requirements | Users should be logged in the business dashboard. |

1. **Use case 7 - Jisoo Park = PASS**

| Use case name | Manage Business Information (Business Description and User info) |
| --- | --- |
| Participating actor | Business Owner, Business Database |
| Entry condition | When the owner enter the owner dashboard |
| Flow of events | 1. Log in to the owner account  2. On the owner dashboard, the owner will see “Settings”  3. Click Settings  4. The owner will see the text box where they are able to change the Business Description  5. Click the save changes button (This will update the database) |
| Exit condition | When the save button is pressed |

| Test-case identifier | Business Description Modification |
| --- | --- |
| Test location | Owner Account / Owner Dashboard |
| Feature to be tested | Only after an owner logs in. Through the owner dashboard, the owner account can make changes in the database. |
| Feature Pass/Fail Criteria | Pass when the Business User information has been successfully changed in the database through the Business dashboard. Fail when Business dashboard is accessible without login and database does not update when change is made. |
| Means of control | A “POST” request is sent when clicking on the save button in the settings page. |
| Data | Users should be able to view his/her changed data after clicking the save button in the dashboard. |
| Test Procedure | 1. Login and stay in  2. Go to settings page  3. Change anything in the text page  4. Click on save  5. Check if displayed data changed |
| Special requirements | The users are logged in. |

1. **Use case 8 - Jisoo Park = PASS**

| Use case name | Searching Salon |
| --- | --- |
| Participating actor | Customers, Business Database, Address Database |
| Entry condition | When the owner enters the dashboard |
| Flow of events | 1. On homepage locate the search bar  2. Type in the desired address(Zip, City) or name of the business,  3. Look for the desired businesses in the list of filtered businesses  4. Click on the profile of the business to see the list of services. |
| Exit condition | When the save button is pressed |

| Test-case identifier | Searching Salon |
| --- | --- |
| Test location | Petit Home Page (Search) |
| Feature to be tested | Website is able to show shops according to what the customer searches, Zip, City, and Name |
| Feature Pass/Fail Criteria | The test passes if users can see the correct shops according to their search. |
| Means of control | A “GET” request is sent when clicking on search, communicating with the Backend to retrieve and display data. |
| Data | Users should be able to view their search results on the search page. |
| Test Procedure | 1. Go to Petit Website (do not login)  2. Locate the search bar and decide what to search by (Zip, City, or Name)  3. Enter text search by the criteria  4. Click search  5. Check if displayed data changed |
| Special requirements | The users are NOT logged in |

1. **Use case 9 - Guillermo Clara = PASS**

| Use case name | Edit Services |
| --- | --- |
| Participating actor | Business Owner |
| Entry condition | When the owner enter the business dashboard. |
| Flow of events | 1. On homepage click the login button  2. Enter account credentials  3. Click login button  4. Select Manage Services in the business dashboard  5. Perform action (If delete, click trash icon next to service, If add, enter text in input box and click Add)  6. Click save changes button  7. Continue editing |
| Exit condition | The Salon Owner/Manager is satisfied with changes made to services list |

| Test-case identifier | Business Services Modification |
| --- | --- |
| Test location | /services |
| Feature to be tested | Backend is able to process changes done by the user and return the updated services data |
| Feature Pass/Fail Criteria | The test passes if the server returns the services data with all changes made (if any) |
| Means of control | A POST request with the changes data is sent to the backend, after processing changes, backend returns a response with an updated services list |
| Data | Array of services objects that represent changes |
| Test Procedure | 1. Log in the Business Dashboard 2. Click Services 3. Perform changes (delete or create) 4. Click Save button 5. Check if data was updated |
| Special requirements | The users are logged in. |

1. **Use case 10 - Guillermo Clara = PASS**

| Use case name | Account Recovery |
| --- | --- |
| Participating actor | Business owner, User Database, Recovery Database |
| Entry condition | User clicks on Forgot Password in Login page |
| Flow of events | 1. On homepage click Business Portal  2. Go to Forgot Password  3. Enter email of account  4. Click on request recover button  5. Enter code sent to email  6. Click submit code button  7. Enter new password  8. Save changes |
| Exit condition | Password change is applied to User database |

| Test-case identifier | Recover account |
| --- | --- |
| Test location | /recovery and /login |
| Feature to be tested | User is able to change their login credentials using their email and a recovery code |
| Feature Pass/Fail Criteria | The test passes if the user can get a recovery code in his/her email and when inputting it is allowed to modify his/her account password. It Fails if the code is not delivered or the changes to the password are not applied in the database |
| Means of control | When the submit button is clicked, POST request is sent to api/recovery/request where it generates a recovery code, stores it in the database and sends it to the provided email using SMTP. When user submits recovery code, make POST request to api/recovery/check. When user submits new passwords make POST request to api/recovery/change |
| Data | User input (email, recovery code, new passwords) sent thru POST request  Action outcome (valid or not valid) sent as a response for every POST request |
| Test Procedure | The test is started by entering and submitting an existing account email in the recovery page. Verify if code is sent to the provided email. Enter and submit code. Enter and submit new passwords. Go to Login page and check if new password works |
| Special requirements | The user must not be logged |