

# Dilusha Alponso Senior QA Engineer hiring assignment

## Test Application

For the real application testing, I am using a website that I created as a part of the Software engineering advanced diploma program conducted by Brainstation. The web application is called **Childcare Connect**. It's built on **Node.js** and **React** and is currently deployed in deployed on **Netlify** (for the front) and **Heroku** (for the backend). The application is designed to connect parents who are looking for quality childcare for their child

This is my app URL - <https://endearing-salamander-3ce910.netlify.app/>

## Test Strategy for Child Care App

A Test Strategy outlines the objectives, scope, overall approach, and testing methodologies to ensure that the functions of the Child Care App work as expected

### Objectives

The main focus of the child care app is to help parents find quality child care for their children in their surrounding area. The app allows users to view all nearby daycare centers, check reviews, and choose the best option. Additionally, parents can directly communicate with daycare providers through the app

### Scope

Features to be tested

1. **As a user**, I should be able to navigate between the **Daycare** and **Profile** options.
2. **As a user**, I should be able to search for a specific childcare provider.
3. **As a user**, I should be able to land on the childcare details page upon clicking a childcare provider's name on the home page.
4. **As a user**, I should be able to sort childcare providers based on ratings.
5. **As a user**, I should be able to view the **Contact Us** page when clicking the email link for a childcare provider on the home page.
6. **As a user**, I should be able to submit the **Contact Us** form.
7. **As a childcare provider**, I should be able to receive email notifications once a user submits the **Contact Us** form.
8. **As a user**, upon clicking the **ChildcareConnect** logo, I should be redirected to the home page.
9. **As a childcare provider**, I should be able to sign in.
10. **As a childcare provider**, I should be able to log in

11. **As a user**, I should be able to add my childcare details upon submitting the form.
12. **As a user**, I should be able to see my childcare location on **Google Maps** after entering the childcare details.
13. **As a childcare provider**, I should be able to see my childcare listing once I submit the form.
14. **As a parent user**, I should be able to submit a review.
15. **As a user**, I should be able to see reviews displayed at the bottom of the page.

## Test Approach

There are two new core features released as part of this release and currently, the app doesn't have any significant issues or bugs in the core or business critical functionalities. As a test approach, I will be using a combination of Automated and manual testing methodologies.

The **signup process** is a prerequisite for other features, making it a critical area of focus. Additionally, I have recently finalized and released this feature. Since it is widely used, I will test it using automation to ensure the expected results.

Furthermore, the **review-adding feature** was also recently released. As this is a core functionality of the application that helps identify quality daycare providers, I have included it in automation testing to ensure the functionality of the app

Typically, I prefer **manual testing** for UI/UX testing to evaluate the user-friendliness of the application and ensure a seamless user experience

## Test Environment

- Browser - Chrome
- Device - Windows
- Staging URL - <https://endearing-salamander-3ce910.netlify.app/>

## Risk Analysis & Mitigation

I will be prioritizing the impact in the following order

**1. New Features and bug fixes** - Stakeholders who are expecting the new release would immediately use/test these features hence they should be considered as high-risk features that had to be tested. New features/Bug fixes that don't have any automation coverage will be given priority.

If the critical new features/bug fixes are not working then there is no point in proceeding with the release hence it has to be tested first.

**2. Business Impact -Features**, which are related to core business functionality in which a failure can cause a downtime or high failure cost have to be tested next.

**3. Historical and defect data** - Based on the past release cycles features/modules that had high bug rates will be tested

**4. Frequency of use** - Focused on the functionality commonly used by the user

Risk	Impact	Testcase priority	Mitigation plan
Signup functionality failure	High priority	High priority	New Feature, The signup process is crucial as it is the first interaction the user has with the application. Any issues with this functionality can prevent users from accessing the app entirely. To ensure its stability, automated tests should trigger every time there is a build since this would be a prerequisite for other test cases.
Adding child care	High priority	High priority	Adding a childcare facility is a critical feature that depends on the successful completion of the signup process. If there are any bugs or issues fixed in the system, automation should verify this feature to ensure that users can continue adding childcare options without disruptions
Adding review	High priority	High priority	New Feature, This is a core feature that enables users to evaluate childcare facilities, making it essential for user satisfaction and app engagement. It's important to check that users can submit reviews successfully every time changes are made to the build, as this affects the overall usability of the app and the business
Adding contact form	High priority	High priority	The contact form allows users to communicate with childcare providers, which is vital for exchanging needs and expectations. Testing should ensure that the childcare provider can get the email notification

## Testing Tool

Tool	Software	Reasons
Automation	Selenium with Java	Selenium is an open-source tool, making it cost-effective. It supports test automation across multiple browsers and platforms, which is ideal for functional test cases.
Testcase management	TestLink	TestLink allows for easy management and execution of test cases. Failed test cases can be linked directly to bugs for better tracking and resolution
Bug reporting	Jira	Jira provides efficient bug tracking and management. It offers detailed reports, enabling teams to manage and track bugs effectively throughout the development process.

## Test Automation framework

I have used the Selenium test automation framework with Java, following the Page Object Model (POM) design pattern. The driver initialization is set up in the BaseTest.java class, ensuring that the WebDriver instance is available for all tests. Since some elements lack unique identifiers, I have implemented a flexible strategy for handling such elements, utilizing relative XPaths and CSS selectors. To enhance test reliability, I make use of explicit waits, ensuring elements are interactable before performing actions. In case of test failures, I capture screenshots for better debugging and documentation. The framework also includes reusable methods for test setup and teardown, guaranteeing a clean execution environment across test cases. I use assertions to verify the expected behavior, while soft assertions allow for continuous test execution even if some checks fail. This helps in gathering all failure details for thorough analysis. The framework is managed using Maven, ensuring proper dependency management and project build automation. Additionally, I use TestNG for executing test cases, enabling parallel execution, detailed reporting, and the grouping of tests for efficient test suite management

For this assignment, I will be covering two test cases which are

1. User signup - A New Feature and to do other main tasks the signup has to be tested first hence automating this use-case is crucial. Also, this would be one of the first interactions by the end users hence it is business-critical.

2. Adding Childcare Provider Reviews - A New Feature and the idea of the site is to discover and read reviews of existing childcare facilities around a selected location therefore adding reviews is a critical feature that has to be tested.

## Test Case Selection

### Test Case 1:Signup:

#### Positive Testcase

- **Description:** Verify that a user can successfully sign up and access the childcare management page.
- **Steps:**
  1. Navigate to the URL
  2. Click the profile button
  3. Enter the name
  4. Enter email
  5. Enter Password
  6. Click the "Signup" button.

**Expected result** - Verify that the user should be able to navigate the childcare adding page after successful signup

#### Negative TestCases

1. Verify if the **Signup page** displays the right error messages for invalid or wrong inputs
2. Verify that if a user tries to register an existing username then an error message should be displayed
3. Verify validation on the email fields. Only valid email IDs should be allowed.
4. Verify that the mandatory field is empty, the user should be able to get an error message
5. Verify that the user can leave the optional fields blank and on clicking the submit button no validation error appears
6. Verify that the validation of numeric fields by entering alphabets and characters
7. Verify the maximum and minimum length accepted for each field
8. Verify that the registration page properly handles passwords with spaces

9. Verify that the password should be masked

10. Verify that the password field enforces the minimum and maximum character length requirements

### **Test Case 2: Add to review**

#### **Positive TestCases**

- **Description:** The user should be able to add the reviews
- **Steps:**
  1. Navigate to the URL
  2. Select any childcare
  3. Fill in the fields (review name, star, comments)
  4. Click submit button

**Expected** - The user should be able to add the review

2. Verify that the latest review should be displayed on the top

3. Verify that the current date is automatically picked and displayed correctly in the review.

4. Verify that the review details (e.g., title, rating, comments) are displayed correctly after submission.

#### **Negative TestCases**

1. Verify that the mandatory field is empty, the user should be able to get an error message

2. Verify The user should be able to add the review, without the optional field

3. Verify if the user can view their past ratings for an item

## **Conclusion**

I sincerely appreciate you taking the time to review my hiring assignment document. I hope you have also had the chance to go through the attached videos and Git repositories. Please do reach out if you have any questions or concerns about the Test strategy document, test cases, or Automation test code.

I truly hope to be selected for the next steps in the process, as I am eager to contribute and work hard for your team. If given the chance, I will bring my dedication, enthusiasm, and strong work ethic to help achieve great results.

Thank you once again for your time and consideration. I look forward to the possibility of working with you and your team

Your Sincerley

Dilusha Alponso