

CHALLENGE #1 - Test Design Challenge

This challenge is supposed to help you to showcase your aptitude in understanding real-life challenges in software development, and application of test design techniques that would find a fine balance between risk coverage and invested effort.

Please read thoroughly through the problem statement below, keeping in mind that this description is intentionally incomplete and vague at points. After understanding the challenge, spend no more than 30 minutes to draft up a proposed design for implementing tests for validating the correctness of the system under test.

The challenge does not intend to restrict you in any way in choosing your approach, tools or even documentation format of your liking. Please keep in mind though that the output of your work shall be easy to interpret, help others reading follow your train of thought. The design doesn't need to be complete, perfect, shiny, but should call out any obvious omissions, uncovered areas, in order to help assess the remaining risk.

Problem statement

We have a software system managing the following entities:

- User accounts
- Brands (having many Product Lists)
- Product Lists (belonging to one Brand, having many Product Products)

Each of these entities have various attributes (for example: ID, Name).

User accounts

User accounts come in 3 flavors:

- Admin
- Brand Admin (connected to one Brand)
- User (not connected to any Brand)

Let's assume the Admin account already exists. Other accounts are created via registration, and it's Admin assigning the specific roles to newly created accounts.

Accesses

- Admin has access to everything, including:
 - Can create, and manage Brands, Product Lists
 - Can create, and manage (grant permissions) to User Accounts
- Brand admin has admin privileges in the context of their own Brand:
 - Can create, and manage Product Lists
 - Can manage the Brand
- User has read-only permissions on all of the entities

Your task is to design a set of tests that'd minimize unauthorized access to the entities in the software system.

CHALLENGE #2 - Test Automation Challenge

Using your preferred toolset to implement an automated test that:

- Navigates to <https://onskeskyen.dk/>
- Opens Inspiration > Brands
- Selects the **Børn & Baby** category
- Clicks **Plysdyr.dk**
- Opens the **De største bamser** trending list
- Asserts on the number of trending list followers

Please consider the following aspects:

- Reliability
- Readability
- Maintainability of your code,
- Usability of your automated test solution.

Please don't spend more than 30-60 minutes on this task.