Task 1:

Test cases:

Sign in with known details -> successful log in.

Sign in with username and password -> successful log in.

Sign in with email address and password -> successful log in.

Register with new details -> successful log in.

Sign in with incorrect details -> Error message, no log in.

Sign in with incorrect username and correct password -> error message, no log in.

Sign in with incorrect email address and correct password -> error message, no log in.

Sign in with correct username, incorrect password -> error message, no log in.

Sign in with correct email address, incorrect password -> error message, no log in.

Set up a test for successful log in for each of the available languages, asserting text is correct.

Set up a test for unsuccessful log in for each of the available languages, asserting the error messages are correct.

Set up a test for registration for each of the available languages.

Register for a new account, using existing username, new email/password -> error message, no log in.

Register for a new account, using existing email address, new username/password -> error message, no log in.

Register for a new account, using new username/email and existing password -> successful registration and logged in.

Attempt to sign in using no username, but valid email & password -> error message, no log in.

Attempt to sign in using no email, but valid username & password -> error message, no log in.

Attempt to sign in using no password, but valid username & email-> error message, no log in.

Attempt to sign in using no username, email or password -> error message, no log in.

Reset password and login using username and new password -> successful log in.

Reset password and login using email address and new password -> successful log in.

Reset password and login using username and old password -> error message, no log in.

Reset password and login using email and old password -> error message, no log in.

All the above should be run cross browser to ensure that there’s no issues using different browsers - ideally MS edge, IE, chrome, Firefox, safari and opera.

Task 2:

Question 1:

1. Firstly I would ensure that we have full documentation available from the social sign up sites to ensure that we can draft a set of requirements from them. This then allows us to be able to draw up a reasonable set of test cases to ensure sufficient coverage. It would also enable us to ensure differences between the different sites’ implementations are covered. Each individual site would be a subset of the above test cases.
2. Time taken would depend on how many different sites were going to be used and how similar the implementations were. I would expect to test and automate a single site’s implementation in 1-2 days.
3. Risks include potential security risks from the other site’s being hacked/breached. It would also rely on connectivity being reliable between the external site and OneFlow’s systems.

Question 2:

Drag and drop can be troublesome when attempting to automate, but there are testing frameworks that can support it. Selenium can be leveraged to automate dragging and dropping, for example, and HP’s QTP used to be able to, although that product has now changed significantly. So some investigation would be required to see what various tools are capable of.

To make testing easier, each element on the page should have some sort of unique ID (which it seems is already a requirement for OneFlow’s coding as everything on the test webpage seemed to have unique references).

In such cases it’s unrealistic to assume that every combination can be tested as they would run into the thousands if not more. In which case it’s probably a prime candidate for a risk based approach; breaking down the most likely/risky scenarios and automating those.

In reality, achieving 100% coverage for this sort of system may not be possible, but high percentages of coverage would certainly be achievable.