Interview Assessment

Testing Overview

Overall iOS:	14 passes, 3 failures
Overall Android:	12 passes, 3 failures

Login with Valid User Credentials

ID: TC01 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Given a valid username (email address) & password that matches an existing account, the platform should allow the user to perform a login successfully.

Preconditions

- An account already registered with the platform.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

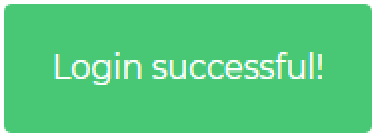
- Valid Email Address: `hello@ciedigital.com`
- Valid Password: `Test 1234`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the valid email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the matching password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was successful. Application should move forward to the **Home Screen**.



Login successful!

Login with No Provided Email Address

ID: TC02 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should refuse entry to the platform if a login request is submitted where the user credentials are incomplete (—such as when no **Email Address** is provided.)

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

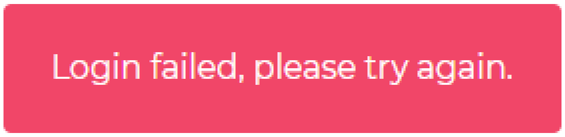
- Sample Password: `test123`

Test Case Steps

1. Do not enter any value into the **Email Address** field.
2. Tap into the **Password** field.
3. Enter the sample password as defined in the **Test Data**.
4. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

Login with No Provided Password

ID: TC03 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should refuse entry to the platform if a login request is submitted where the user credentials are incomplete (—such as when no **Password** is provided.)

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

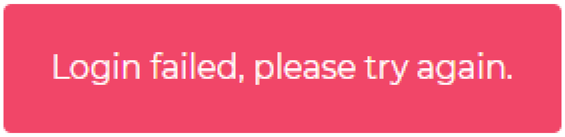
- Sample Email Address: hello@ciedigital.com

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the sample email address as defined in the **Test Data**.
3. Do not enter any value into the **Password** field.
4. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

Login with Valid Email Address, Invalid Password

ID: TC04 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Given a valid username (email address) that matches an existing account, the platform should reject a login request when the password does not match the specified user.

Preconditions

- An account already registered with the platform.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

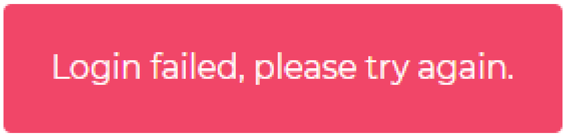
- Valid Email Address: `hello@ciedigital.com`
- Invalid Password: `InvalidPassword123`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the valid email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the invalid password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

Login with Incomplete Email Address

ID: TC05 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should refuse entry to the platform if a login request is submitted where the user credentials are incomplete (—such as when an incomplete **Email Address** is provided.)

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

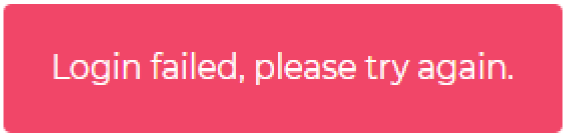
- Incomplete Email Address: `hello@cie`
- Sample Password: `test123`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the incomplete email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the sample password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

Login with Unregistered Email Address

ID: TC06 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should refuse entry to the platform if a login request is submitted where the user credentials are invalid (—such as when an unregistered user attempts to access the platform.)

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

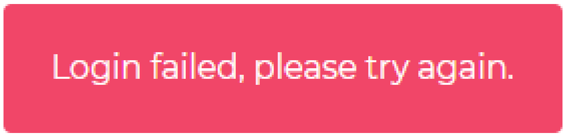
Unregistered Email Address:	`raftacon@gmail.com`
Sample Password:	`test123`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the unregistered email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the sample password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

Email Address with Too Many Characters

ID: TC07 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The login form should restrict or otherwise alert the user when they attempt to provide an **Email Address** exceeding the maximum character limit for the field.

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.
- **Email Address** field has a maximum character limit of 20.

Test Data

Unregistered Email Address:	`verylongemailaddress@gmail.com`
Sample Password:	`test123`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the unregistered email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the sample password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

When the user input exceeds the maximum character limit for the field, any remaining characters entered into the field should be ignored & discarded.

Email Address with Illegal Characters

ID: TC08 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The login form should restrict or otherwise alert the user when they attempt to provide any illegal characters in the **Email Address** field.

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.
- **Email Address** field has restrictions on the following characters: **.,;[\]**

Test Data

- Invalid Email Address: `ema[il\]@gmail.com`
- Sample Password: `test123`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the invalid email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the sample password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

As the user is entering their username in the **Email Address** field, any illegal characters should actively be ignored & discarded as they are encountered.

Password Masking

ID: TC09 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Passwords should not be displayed in plain-text on the page during the login process.

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

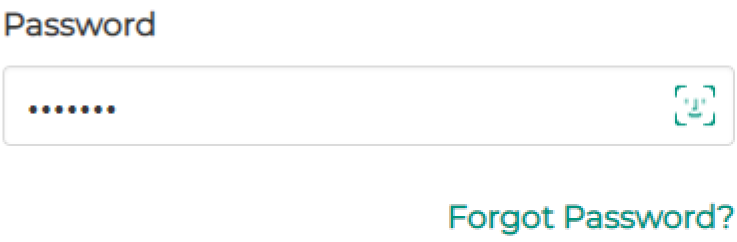
- Sample Password: `test123`

Test Case Steps

1. Tap into the **Password** field.
2. Enter the sample password as defined in the **Test Data**.

Expected Result

The password entered by the user into the **Password** field should display as masked during entry & remain masked post-entry.



Login Autofill Using Face ID

ID: TC10 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Application should support username (email address) & password autofill if **Face ID** / **Iris Scanner** icon is tapped and user is identified successfully.

Preconditions

- An account already registered with the platform.
- User's **Face ID** profile already configured on test device.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

- None.

Test Case Steps

1. Tap on the **Face ID** icon (—pulled to the right within the **Password** field.)

Expected Result

Once the **Face ID** authentication is completed successfully, both the **Email Address** & **Password** fields should autofill with the user credentials stored in the iCloud Keychain.

Autofill Rejection Using Face ID

ID: TC11 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Application should gracefully handle the scenario where an attempt is made to recognize the user using **Face ID** but ultimately fails to authenticate.

Preconditions

- An account already registered with the platform.
- User's **Face ID** profile already configured on test device.
- Ability to cause a rejection (i.e. using different individual than configured profile, etc.)

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

- None.

Test Case Steps

1. Tap on the **Face ID** icon (—pulled to the right within the **Password** field.)

Expected Result

Once the **Face ID** authentication fails, both the **Email Address** & **Password** fields should remain blank and **not** be auto-populated with user credentials pulled from the iCloud Keychain.

Login Autofill Using Touch ID / Fingerprint Scanner

ID: TC12 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

If pre-configured on the device, the application should prompt user for **Touch ID** / configured biometric data as soon as application is launched. On a successful authentication attempt, the username (email address) & password should autofill using credentials tied to biometric profile.

Preconditions

- An account already registered with the platform.
- User's **Touch ID** / biometric profile already configured on test device.

Assumptions

- Application not already launched.

Test Data

- None.

Test Case Steps

1. Launch the application from the device's home screen.

Expected Result

Once the **Touch ID** / biometric authentication is completed successfully, both the **Email Address** & **Password** fields should autofill with the stored user credentials.

Autofill Rejection Using Touch ID / Fingerprint Scanner

ID: TC13 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

If pre-configured on the device, the application should prompt user for **Touch ID** / configured biometric data as soon as application is launched. On an authentication failure, the username (email address) & password should autofill using credentials tied to biometric profile.

Preconditions

- An account already registered with the platform.
- User's **Touch ID** / biometric profile already configured on test device.
- Ability to cause a rejection (i.e. using different individual than configured profile, etc.)

Assumptions

- Application not already launched.

Test Data

- None.

Test Case Steps

1. Launch the application from the device's home screen.

Expected Result

Once the **Touch ID** / biometric authentication fails, both the **Email Address** & **Password** fields should remain blank and **not** be auto-populated with the stored user credentials.

Password Recovery Option

ID: TC14 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should provide a way for the user to confirm their identity & reset their old password.

Preconditions

- An account already registered with the platform.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

- None.

Test Case Steps

1. Tap on the **Forgot Password?** link located beneath the **Password** field.

Expected Result

The user should be presented with a **Security Question** dialog. If the **Security Question** is answered successfully, the user should then be allowed to enter & confirm a new password to use when accessing the platform in the future.

New User Registration

ID: TC15 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should provide a way for the user to register for a new account directly from the login screen.

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

- None.

Test Case Steps

1. Tap on the **Sign up Today?** link located beneath the **Login** button.

Expected Result

The user should be successfully transitioned to the **Registration** screen.

Timeout During Login Attempt

ID: TC16 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

The application should gracefully handle a scenario where a timeout occurs when attempting to validate the user credentials entered.

Preconditions

- An account already registered with the platform.
- Prepare a traffic analyzation utility (—such as Fiddler—) to generate a timeout that will effectively cause the authentication request from the login page to fail.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

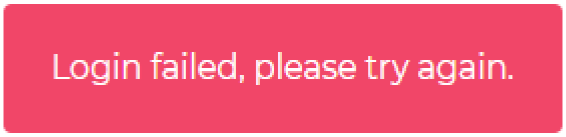
- Valid Email Address: `hello@ciedigital.com`
- Valid Password: `Test 1234`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the valid email address as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the matching password as defined in the **Test Data**.
5. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**.



Login failed, please try again.

SQL Injection Attempt During Login

ID: TC17 | Component: Sign In Screen | Last Modified: 4/7/2021

Objective

Check for potential SQL injection vulnerabilities where a malicious actor could mine for user data or otherwise compromise the platform due to a lack of using prepared statements to retrieve information from the database.

Preconditions

- None.

Assumptions

- Test case begins execution already on the **Sign In Screen**.

Test Data

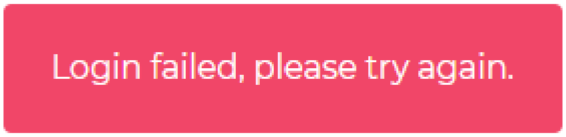
- Sample SQL Injection Phrase: `“ OR 1=1 --`

Test Case Steps

1. Tap into the **Email Address** field.
2. Enter the sample SQL injection phrase as defined in the **Test Data**.
3. Tap into the **Password** field.
4. Enter the sample SQL injection phrase as defined in the **Test Data**.
6. Tap the **Login** button.

Expected Result

A toast notification should alert the user that their login attempt was unsuccessful. Application should remain on the **Sign In Screen**. When inspecting any direct authentication responses from the platform to the application, validate that no extraneous user information is returned.



Login failed, please try again.

iOS Summary

#	Test Case Name	Result	Comments
TC01	Login with Valid User Credentials	**Pass**	
TC02	Login with Empty Email Address	**Pass**	
TC03	Login with Empty Password	**Pass**	
TC04	Login with Valid Email Address, Invalid Password	**Pass**	
TC05	Login with Incomplete Email Address	**Pass**	
TC06	Login with Unregistered Email Address	**Pass**	
TC07	Email Address with Too Many Characters	**Fail**	*(General failure.)* Maximum character limit not detected.
TC08	Email Address with Illegal Characters	**Fail**	*(General failure.)* Illegal characters not detected.
TC09	Password Masking	**Pass**	
TC10	Login Autofill Using Face ID*	**Fail**	*(Specific to iOS 12.0 on iPhone X.)* Face ID icon does not react to user input.
TC11	Autofill Rejection Using Face ID*	**Pass**	
TC12	Login Autofill Using Touch ID / Fingerprint Scanner	**Pass**	
TC13	Autofill Rejection Using Touch ID / Fingerprint Scanner	**Pass**	
TC14	Password Recovery Option	**Pass**	
TC15	New User Registration	**Pass w/ Exception**	Should *"Sign up Today?"* be *"Sign Up Today?"*
TC16	Timeout During Login Attempt	**Pass**	
TC17	SQL Injection Attempt During Login	**Pass**	

Operating Systems Tested

- 12.1, 14.0

Devices Tested

- iPad Air 2*, iPhone 7 Plus*, iPhone X
    - **No Face ID on this device, **TC10** & **TC11** not applicable*

Android Summary

#	Test Case Name	Result	Comments
TC01	Login with Valid User Credentials	**Pass**	
TC02	Login with Empty Email Address	**Pass**	
TC03	Login with Empty Password	**Pass**	
TC04	Login with Valid Email Address, Invalid Password	**Pass**	
TC05	Login with Incomplete Email Address	**Pass**	
TC06	Login with Unregistered Email Address	**Pass**	
TC07	Email Address with Too Many Characters	**Fail**	*(General failure.)* Maximum character limit not detected.
TC08	Email Address with Illegal Characters	**Fail**	*(General failure.)* Illegal characters not detected.
TC09	Password Masking	**Pass**	
TC10	Login Autofill Using Face ID	**N/A**	
TC11	Autofill Rejection Using Face ID	**N/A**	
TC12	Login Autofill Using Touch ID / Fingerprint Scanner	**Pass**	
TC13	Autofill Rejection Using Touch ID / Fingerprint Scanner	**Pass**	
TC14	Password Recovery Option	**Fail**	*(General failure.)* User not redirected to recovery page.
TC15	New User Registration	**Pass w/ Exception**	Should *"Sign up Today?"* be *"Sign Up Today?"*
TC16	Timeout During Login Attempt	**Pass**	
TC17	SQL Injection Attempt During Login	**Pass**	

Operating Systems Tested

- OS 9 (Pie), OS 10, OS 11

Devices Tested

- Google Pixel 3, Samsung Galaxy A40, Samsung Galaxy Tab S6 (WiFi)