**Application Testing Template:**

**Authentication Testing**

There are two steps involved in testing authentication:

1. Set up the integration with UGA SSO
2. Login with one or more test accounts to determine if authentication is successful.

*Note: you can submit a request for test accounts if needed*

|  |  |  |
| --- | --- | --- |
| **Authentication Testing for:** [Application Name] | | |
| **Environment**: Development | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

|  |  |  |
| --- | --- | --- |
| **Authentication Testing for:** [Application Name] | | |
| **Environment**: Staging | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

**Authorization Testing**

There are additional steps if only specific users are authorized to use the application:

1. Set up test accounts with appropriate authorization criteria
   * i.e., the user must be a student
2. Set up the integration with UGA SSO
3. Login with specific test accounts to determine if specific authorization scenarios are successful.

|  |  |
| --- | --- |
| **Authorization Test Account Descriptions** | |
|  | |
| **Test Account Name:** | **Specifications:** |
| i.e., test123 | i.e., FS Code = ‘00’ |
| *Add rows as needed* |  |

|  |  |  |
| --- | --- | --- |
| **Authorization Testing for:** [Application Name] | | |
| **Environment**: Development | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

|  |  |  |
| --- | --- | --- |
| **Authorization Testing for:** [Application Name] | | |
| **Environment**: Staging | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

**Attribute Retrieval Testing**

There are additional steps if the application accesses additional attribute from the SSO ticket or session:

1. Set up test accounts with known values for the requested attributes
   1. i.e., the UGAID number, last name, department, etc, associated with the test account
2. Set up the integration with UGA SSO including access to requested attributes.
3. Login with specific test accounts to determine if the data from the requested fields is correctly retrieved by the application.

|  |  |
| --- | --- |
| **Attribute Descriptions** | |
|  | |
| **Field to be Retrieved:** | **Description:** |
| i.e., UGAID | UGA Identification Number |
| *Add rows as needed* |  |

|  |  |  |
| --- | --- | --- |
| **Retrieval Testing for:** [Application Name] | | |
| **Environment**: Development | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

|  |  |  |
| --- | --- | --- |
| **Retrieval Testing for:** [Application Name] | | |
| **Environment**: Staging | | |
|  | | |
| **Task** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] |  |  |
| *Add rows as needed* |  |  |

**Application Session Testing**

Your application may have special behavior with session management:

1. Set up test accounts with known values for the requested attributes
   1. i.e., the UGAID number, last name, department, etc, associated with the test account
2. Set up the integration with UGA SSO including access to requested attributes.
3. Login with specific test accounts to determine if the and review the application behavior on how it handle browser and sessions.

|  |  |  |
| --- | --- | --- |
| **Browser Testing** | |  |
|  | |  |
| **Browser:** | **Expected Behavior** | **Result (Pass/Fail)** |
| Safari | Log in successfully; visually looks appropriate; application handles changes in cookies and session configuration |  |
| *Edge* | Log in successfully; visually looks appropriate; application handles changes in cookies and session configuration |  |
| *Chrome* | Log in successfully; visually looks appropriate; application handles changes in cookies and session configuration |  |
| *IE* | Log in successfully; visually looks appropriate; application handles changes in cookies and session configuration |  |
| *Add Rows as needed* |  |  |

|  |  |  |
| --- | --- | --- |
| **Session Management Testing for:** [Application Name] | | |
| [**Session Managment Testing by OWASP**](https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/README.html) | | |
|  | | |
| **Checks** | **Expected Behavior** | **Result (Pass/Fail)** |
| Login with Test Account [Name] | Only authorized users can get in |  |
| Use HTTP and HTTPS | Only HTTPS is active |  |
| Session Timeout / Duration | Timeout is appropriate |  |
| User Logoff validations | When you log out it requires reauthentication to get back in even in a new tab |  |
| Web Browser Cross Tab Sessions | re-authenticate if a new web browser tab or window is opened against the same web application |  |
| Renewal timeout / Automatic Logout | When a new session would be required |  |
| Cookies | [Cookie Managment Testing by OWASP](https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html) |  |
| Single Sign On capabilities (if applicable) | Ensure that your application accepts previously established sessions |  |
| Browser session exits | Exit browser and re-establish session |  |
| Multi applications | Log in and out of different applications while testing your session duration |  |
| *Add rows as needed* |  |  |