The MSDTool application is intended to both be a finished useful application as well as a highly dynamic one where new tools can be added in the form of plug-ins (dlls). This is necessary because the application is intended to continually assist material researchers as they develop new techniques in which to process and analyze materials.

The MSDTool as a whole is split into two separate applications, a client, and a server. Acceptance test cases in this document will pertain to the two separately. The following use cases will direct the actions that the tools should be able to perform correctly for the application to be a success.

The MSDTool application has been developed alongside the Materials Engineering Department at Drexel University. Their faculty, research students, and colleagues at Brigham Young University will evaluate the application and apply the use cases described below to judge the acceptance of the application.

The MSDTool Server Application

The server application’s main purpose is to initiate, deploy, and maintain the material database. The material database will mainly contain user information such as login information and user privileges, sample material data, point set data, precomputed material closures, and general scientific information used by material engineers such as mathematical constants.

MSDTool Server Application Use Cases

1. Name: Server Install
   Pre: The user has acquired the MSDTool server installer package.
   Actions: The user runs the server installer.
   Post: The user completes the install, an initial database is properly created and populated with information contained in the install.

2. Name: User Addition
   Pre: The database server is running.
   Actions: A user addition request is sent to the server.
3. Name: User Deletion

Pre: The database server is running.

Actions: A user deletion request is sent to the server.

Post: A user account is removed if the request has the proper permissions.

4. Name: Data Addition

Pre: The database server is running.

Action: The database receives a request to add a sample, point set, or mathematical constant.

Post: A sample, point set, or mathematical constant is added to the database for later use.

5. Name: Technique Addition

Pre: The database server is running.

Action: The database receives a request to add a new technique.

Post: A technique is added as well as the point set it should run on to start calculating closures.

6. Name: Technique Request

Pre: The database server is running.

Action: The client requests techniques.

Post: Techniques are sent to update the user’s local technique list.

7. Name: Data Request

Pre: The database server is running.

Action: A request is sent to the server to send data.
Requested samples, closures, or mathematical constants are sent to the client.

As well as the above use cases the database server once started, is meant to continually run constantly acquiring data and calculating new closures according to client requests. The client application in general must support the ability to add at least two types of techniques, particularly elastic and plastic stiffness for cubic symmetry.

The MSDTool Client Application

The client application allows material designers to load in information from files and the server in which to visualize and apply new techniques too. The client enables designers to work with data collected from research tools, apply techniques to calculate closures, and visualize their data in 2D and 3D graphs. The client application in general must support the ability to add at least two types of techniques, particularly elastic and plastic stiffness for cubic symmetry.

1. Name: Login

   Pre: The client application is opened and running.
   Action: The user enters their user name, password, and chooses a database server to connect to.
   Post: The database validates the user and if the user has an account they are logged into the client.

2. Name: Logout

   Pre: The client application is running and a user is logged in.
   Action: The user chooses to log out of the application.
   Post: The user is logged out and can’t do anything else but log back in.

3. Name: Application Update

   Pre: The application has just been logged into.
   Action: N/A
   Post: The client syncs up with the server and downloads materials properties, techniques, and sample / point set names.

4. Name: Add Sample/Point Set to Client

   Pre: The client application is running and the a user is logged in.
Action: The user selects a sample from the known samples from the database server or loads in a sample from a file.

Post: A sample is added to the user space.

Name: Add Sample to Database

Pre: The client application is running.

Action: The user chooses to add a sample to the database by naming it, importing its data (Euler angles or Fourier coefficients), and giving it permissions.

Post: The sample is added to the database for global use if the permissions allow.

5. Name: Add Point Set to Database

Pre: The client application is running.

Action: The user chooses to add a point set to the database by naming it, importing its data (Euler angles or Fourier coefficients), giving it permissions, and assigning it properties (e.g. stiffness).

Post: The point set is added to the database for global use if the permissions allow.

6. Name: Edit Sample / Point Set

Pre: The application is running and there is a sample / point set either local or on the database.

Action: The user chooses a sample / point set so they can edit the labeling information, the data (Euler angles or Fourier coefficients), or change the permissions.

Post: A sample / point set is edited and its changes are stored locally or on the database.

7. Name: Add Sample to Graph

Pre: The client application has local samples for the user to view.

Action: The user selects a local sample to be viewed and the graph contains closures from sample points in which to graph the sample.
Post: Where the sample lies within the closure is displayed to the user.

8. Name: Add Point Set to Graph

Pre: The client application has local point sets for the user to view.

Action: The user selects to add a point set to the graph, the raw point set can be visualized or if the database finished calculating the closure that can also be viewed.

Post: The point set is displayed to the user.

9. Name: Add Property

Pre: The client application is running.

Action: The user chooses to add a property to the current property list.

Post: The user is now enable to select this property for use in the sample / point set graph.

10. Name: Change Property

Pre: The client application is running.

Action: The user selects a new property.

Post: The axis of the graph that the property applies too is changed, if a sample / point set is loaded into the graph it is updated.

11. Name: Contour Pole Figure

Pre: The client application is running and a sample is graphed.

Action: The user chooses to view a 3D pole figure of the sample data.

Post: The Pole Figure Viewer is shown to the user.

12. Name: Manipulate Pole Figure

Pre: The Pole Figure Viewer is open.

Action: The user manipulates controls in the Pole Figure Viewer.

Post: The pole figure is modified to match the user’s request such as camera angle changes and flat view of the data.
13. Name: Export

Pre: The application is running and there is data (samples, point sets, mathematical constants, or graphs) to export.

Action: The user chooses to export their personal material database or an image of a particular graph.

Post: A material database file is exported or an image of a selected graph.