**Shot prep Tool UI**

1. File Menu:
   1. New file
      1. Ensure that it will create a new h5 file with the correct file name
      2. Ensure that the GUI will open a warning dialog box if the user tries to create a file with the same filename as a file that already exists
      3. Ensure that the GUI will open a warning dialog if the user tries to create a new file while there are unsaved changes in the current file
      4. Verify that the GUI will raise a warning if there is a file lock on the h5 file
      5. Test that cancelling the operation works as expected
   2. Open file
      1. Ensure that the prep tool will open an h5 file.
      2. Verify that a warning dialog will pop up if there are any unsaved changes.
      3. Verify that the GUI will raise a warning if the user attempts to load an invalid (corrupt or misnamed) h5 file. Verify that both options (Discard and Cancel) work as expected for the warning box.
      4. Verify that the GUI will raise a warning if there is a file lock on the h5 file
      5. Test that cancelling the operation works as expected
   3. Save file
      1. Ensure that saving the file will actually save the changes made into the file.
   4. Save as
      1. Ensure that the save as function will save the changes to the new file specified.
      2. Verify that the original file will not be changed.
      3. Ensure that save as will open a warning dialog box if the prep tool attempts to overwrite a file that already exists. Verify that both options (Discard and Cancel) work as expected for the warning box.
      4. Test that cancelling the saving operation works as expected
   5. Close
      1. Ensure that the file will be closed.
      2. Verify that a warning dialog will pop up if there are any unsaved changes
   6. Exit
      1. Verify that a warning dialog will pop up if there are any unsaved changes. Verify that both options (Discard and Cancel) work as expected for the warning box.
      2. Verify that the UI will exit.
      3. Verify for both the ‘X’ in the top right corner and the exit button in the file menu.
2. Device menu
   1. Verify that devices can be added to the h5 file
      1. Verify that the UI will show a meaningful warning if the user can’t add that device. For instance if the user attempts to add a device with the same name as one that already exists.
   2. Verify that devices can be removed from the h5 file
3. Row menu
   1. Verify that values can be added to the current device
      1. Verify that the UI will show a meaningful warning if the user can’t add that value. For instance if the user attempts to add a value with the same name as one that already exists.
   2. Verify that attributes can be removed from the current device
4. File name test cases – For New File and Save As functions
   1. Test the prep tool with filenames with no explicit extension
   2. Test the prep tool with the wrong file extension.
5. Warning and dialog boxes.
   1. Ensure that the warning messages make sense.
   2. Verify that both the option buttons work as expected
6. Multiple prep tools open at the same time
   1. Ensure that the tool raises a warning dialog if the user tries to open an h5 file that’s already in use.
7. Test the buttons in various configurations and orders
8. Make sure that the Prep tool will no longer create any temporary files. The tool should now do all work in memory.

**Shot Runner Tool UI**

1. File Menu:
   1. New file
      1. Ensure that it will create a new json file with the correct file name
      2. Ensure that the GUI will open a warning dialog box if the user tries to create a file with the same filename as a file that already exists
      3. Ensure that the GUI will open a warning dialog if the user tries to create a new file while there are unsaved changes in the current file
      4. Verify that the GUI will raise a warning if there is a file lock on the json file
      5. Test that cancelling the operation works as expected
   2. Open file
      1. Ensure that the runner tool will open a json file.
      2. Verify that a warning dialog will pop up if there are any unsaved changes.
      3. Verify that the GUI will raise a warning if the user attempts to load an invalid (corrupt or misnamed) json file. Verify that both options (Discard and Cancel) work as expected for the warning box.
      4. Verify that the GUI will raise a warning if there is a file lock on the json file
      5. Test that cancelling the operation works as expected
   3. Save file
      1. Ensure that the save function will actually save changes made to the file.
   4. Save as
      1. Ensure that the save as function will save the changes to the new file specified.
      2. Verify that the original file will not be changed.
      3. Ensure that save as will open a warning dialog box if the runner tool attempts to overwrite a file that already exists. Verify that both options (Discard and Cancel) work as expected for the warning box.
      4. Test that cancelling the saving operation works as expected
   5. Close
      1. Ensure that the file will be closed.
      2. Verify that a warning dialog will pop up if there are any unsaved changes
   6. Exit
      1. Verify that a warning dialog will pop up if there are any unsaved changes. Verify that both options (Discard and Cancel) work as expected for the warning box.
      2. Verify that the UI will exit.
      3. Verify for both the ‘X’ in the top right corner and the exit button in the file menu.
2. Run Button:
   1. Ensure that the run button will run the scripts in the correct order with the correct settings files
3. File name test cases – For New File and Save As functions
   1. Test the runner tool with filenames with no explicit extension
   2. Test the runner tool with the wrong file extension.
4. Warning and dialog boxes.
   1. Ensure that the warning messages make sense.
   2. Verify that both the option buttons work as expected
5. Multiple runner tools open at the same time
   1. Ensure that the tool raises a warning dialog if the user tries to open an h5 file that’s already in use.
6. Test the buttons in various configurations and orders

**Integration Tests**

1. Verify that scripts ran with the runner tool will output the same results as scripts ran on the old framework.
2. Collect the survey on the usability of our GUI.

**PrismaPlus Controller Driver**