**Evidence matching task checklist:**

**The tasks consists of:**

SAP side: *(Lily said she get someone to make a script to automate below)*

1. Start SAP
2. Chose a section to make change log
3. Automate SAP logon (Seems to have done)
4. Chose the start date, end date and enter list of names in IDMS
5. Check the first and second selection tabs at the bottom of the page to get all changes
6. Generate the change log as an excel file
7. Save the file in the following format “<start to end date> (<section>)”
8. Repeat steps 2 to 7 for all sections
9. Compile all generated excel files into one

Checking: *(What we are trying to automate)*

1. Access the changelog file
2. Preferably sort the file by changer so that do not need to keep opening diff folders (might not be necessary for automation)
3. According to data retrieved from the row, do 4 to 7
4. Open the required folder
5. Open the file for the change
6. Look inside the file and see if the change matches up (Also check for approval etc)
7. If the check is good and clear (put an “ok” [might want to have something diff for automated checks] in the 1st col of the row that has been checked)
8. Save the file

**The following documentation will only refer to the checking portion of the task.**

What we have done: *(Python, openpyxl)*

1. Access the changelog file
   1. A pop-up window will appear prompting the user to select a file to use
   2. The program will stop if the pop-up is closed or cancel is clicked
   3. The file will then be opened in openpyxl
2. Preferably sort the file by changer so that do not need to keep opening diff folders (might not be necessary for automation)
   1. Not done, not needed, the computer can access these directories directly, and with no delay
3. According to data retrieved from the row, do 4 to 7
   1. The column headers (1st row) are scanned to get the letter of the column for the data needed (user, date, change by, action).
   2. After getting the column letters, Skip scanning the 1st 2 rows (second row is empty)
   3. Scan the rest of the rows and retrieve the data according to the column letters gotten initially
   4. Should the (user and change by) be the same as the previous row, skip the row
   5. Special cases done:
      1. If change by is “SAPJSF\_EIP” or “SAPJSF\_SXP” skip row
      2. If change by and user is DDIC and action is “Password changed” put “reset” in the cell of 1st column of the row. If the action is something else, put “ACTION NEEDED” into the cell of 1st column of the row while making the cell have a black background and red text. Continue with the next row after doing so.
      3. If change by and user is SAP\* and action is “Password changed” put “reset” in the cell of 1st column of the row. If the action is something else, put “ACTION NEEDED” into the cell of 1st column of the row while making the cell have a black background and red text. Continue with the next row after doing so.
      4. If change by and user is SUPER and action is “Password changed” put “reset” in the cell of 1st column of the row. If the action is something else, put “ACTION NEEDED” into the cell of 1st column of the row while making the cell have a black background and red text. Continue with the next row after doing so.
      5. If change by and user is the same, put a “N/A” into the cell of 1st column of the row while making the cell background orange. Continue with the next row after doing so.
4. Open the required folder
   1. The program will find the folder based on the date given as well as the changer
5. Open the file for the change
   1. Not done, have not tried to extract the data, the path to the file is gotten instead
   2. The file is found via searching for the user gotten in change log
6. Look inside the file and see if the change matches up (Also check for approval etc)
   1. Not done at all, instead if file can be found, a link is made in the excel file
7. If the check is good and clear (put an “ok” [might want to have something diff for automated checks] in the 1st col of the row that has been checked)
   1. The links are formatted as below:
      1. If a single file matches, “file” will be put in the 1st cell of the row and the cell will have a blue fill
      2. If multiple files match, “multi file” will be put in 1st cell of the row and the cell will have a green fill
      3. If no file matches, “no file” will be put in the 1st cell of the row and the cell will have a yellow fill
8. Save the file
   1. The file will be saved in the same directory as it is gotten from and will be saved as the file name with “-MECHchecked” added behind it

Limitations:

1. The file open will not be visible, the program only works with .xlsx files and not .xls
2. NIL (program might be faster if all the docs with the same date are together)
3. The scanning of the data of the cell is case sensitive, so care needs to be taken that the string the program searches for and what is in the cell matches. Should the 2nd row not be empty, the program won’t know to scan it, also, if there is an empty row between data-filled rows, the program will crash
4. Nothing can be seen. The algorithm (is it called that?) for finding the file will always assume that the year is the 1st folder (i.e. username/2017)[Choo Kiat’s folder], meaning if the folder is named like “Year 2017” it will fail. Also, when finding the folder for the month, should the month be written using 1 digit, like “2017 1”, instead of 2 digits, “2017 01”, the program will fail. In case of failure, the path will be as far as the program manages to find, so if it can’t find the month for example, the link generated at the end will bring the user inside the 2017 folder only. If there are folders with numbers in the name that match the date, it might select that folder instead
5. If the file does not contain the user data gotten from the excel sheet, the program will fail to find the file
6. The program is unable to check the contents of the file
7. NIL?
8. There is a chance that the file saved might end up corrupted

For improvement, future development/ RL consistent usage:

1. Have a pop-up to show the file selected, possibly ask for confirmation
2. Sort thee rows of data according to change by and date, then make the program not have to generate a new path if the change by and date is the same as the previous checked row
3. The scanning for headers can be made non-case sensitive as the headers needed are all unique. Also, enable the program to know how to skip empty rows automatically or prompt the user for action instead of crashing
4. The pathfinder can be improved by making it not assume that the year alone is the 1st folder name as it is not the case for Choo Kiat’s folder. Try to make it be able to differentiate between folders labelled by date and other folders
5. Make the program gather more data from the excel sheet so that it can match the results better with the name
6. See if it is possible to check the data inside the file so that checks for the simpler requests need not be done
7. Check if there is a clearer way to show the change?
8. NIL?