Forwarding IDMS requests:

The task: *(UPDATE 1- in italics)*

1. *Pop-up asking the user to choose who to delegate to*
2. Open the IDMS page in Internet Explorer
3. Log in to the IDMS system
4. Go to the Bulk assign group tasks page
5. Check tasks of a certain “Application Name” (Windows (ITD Only))
6. Assign the tasks to self and save
7. Go to My delegations -> Create delegation
8. Chose person in charge to delegate to (the person changes weekly, info gotten via *~~email from wintel team~~ popup box at start*)
9. Type “FYA, please” into comments
10. Check the tasks to delegate (same things to check as step 4)
11. Click save to delegate tasks

What has been done: *(Python, Selenium)*

1. *Pop-up asking the user to choose who to delegate to* ○○
   1. A pop-up box will appear, with a drop down menu for selecting what tasks to delegate (default is “Windows (ITD Only)”, option might not be needed) and a series of radio buttons from which to select person to delegate to
   2. Clicking cancel will quit the entire program
2. Open the IDMS page in Internet Explorer ○○
   1. The current url is [http://qid](http://qidmwb01.singaporepower.local) (is this the non-QA one?)
3. Log in to the IDMS system ○
   1. Program automatically logs into the QA account, this can be easily removed which would result in the program waiting for user to log in manually. (Automating the log in for actual accounts is not done as the password would be stored in plain text)
4. Go to the Bulk assign group tasks page ○○
   1. This is done though going to the url directly
5. Check tasks of a certain “Application Name” (Windows (ITD Only)) ○○
   1. The program will find all checkbox in rows that have “Windows (ITD Only)” as Application Name
   2. It will then click on all checkboxes found
6. Assign the tasks to self and save ○○
   1. Filling in of name is now done via typing in the name, wait for 1.5 seconds for the autocomplete menu to appear before stimulating an enter key press.
   2. The person to assign to (self) is now gotten from the top right, where the logout button is placed
7. Go to My delegations -> Create delegation ○○
   1. This is done though going to the url directly
8. Chose person in charge to delegate to (the person changes weekly, info gotten via *~~email from wintel team~~* popup box at start) ○○
   1. Person to delegate to is selected in at the start of the code
   2. The auto complete is also done in the same as 6.i
9. Type “FYA, please” into comments ○○
10. Check the tasks to delegate (same things to check as step 4) ○○
    1. Done the same way as 5..
11. Click save to delegate tasks ○○

Limitations/disadvantages of the current code:

1. Pop-up box will have default options selected which could result in the user accidentally running the program with the wrong set of data as there is no confirmation dialog, also, the data that is gotten is not displayed
2. The IDMS page will always open in a new browser
3. The user must log in manually to stay secure
4. If the url changes the code needs to be updated
5. The code will only find exact matches, so if the “Application Name” choices for the pop-up box is poorly done (typos, wrong capital letters), the program will fail to be able to find the task
6. The way the code is done will make it so that the 1st name of the autocomplete is selected. Thus, to ensure that the correct person is selected, the full name in IDMS should be typed out.
7. Same as 4.
8. Every time the code is run, the person to delegate to must be selected. The Autocomplete also has the same issue as 6.
9. Nil
10. Same as 5.
11. Although the browser will remain open at the end of the code, there is currently no indication to the user that the code has finished running.

Overall: Xpath is currently used to locate majority elements (the things we interact with) thus should the HTML be changed, there would be a need to update these. The automation may require the Internet Explorer browser to be focused on. While testing, it has been seen that the program can still run while it is not focused provided the window is not minimised. Also, should the program run into an error, there will be no indication to the user.

For improvement, future development/ RL consistent usage:

1. The UI of the pop-up interface can be improved upon. Also, might want to increase the size of it to make it more obvious when it appears.
2. IDMS open in a new browser window, this is unavoidable until selenium comes up with support for it. (cannot be found as of 13/09/2017)
3. It seems like there is something that allows for automatic IDMS login (IE autofill), see if it can be used with the python, selenium code. Otherwise, it should be possible to store the data in an encrypted way, thereby allowing the password to be stored safely, without compromising on security too much.
4. It is possible to make it navigate to the page via the clicking of buttons like a user would do, however, this method is significantly slower
5. It should be noted that clicking the next page button will clear any selection that has been made if the save button is not pressed. If the task to be selected only appears on the second page, the program will not be able to find it. The program could be made to check all pages for the task that is requested for it to find. Otherwise, it might be good for the program to filter the results before checking, even if it slows the program slightly due to the page having to reload.
6. If the autocomplete cannot find a name, the autocomplete field will be left blank, thereby causing the code to fail even though it will continue running, something should be done to check the value in the autocomplete box
7. Same as 4.
8. Same as 5.
9. Nil
10. Same as 5.
11. Add a pop-up message or alert to let the user know that program has finished running

Overall: It would be good if the IDMS could be made such that certain fields such as the Assign to box and the like could have better unique IDs or names in HTML so that the code could be written in a neater way. To allow use of computer while code is running, headless browsers can be used, however although headless browsers are faster and do not require focus, there will not be anything that can be seen to see what is going on in the “browser”. Make pop-up for errors so that the user would know when an error has occurred, also maybe a log file so that what caused the error can be better identified and sloved

**OLD – ruby code details (less efficient) [only for reference]**

What has been done: *(Ruby, Capybara used) -OLD, Outdated, refer below for more efficient Python code details*

1. Open the IDMS page in Internet Explorer ○○
   1. The current url is (is this the non-QA one?)
2. Log in to the IDMS system ◌◌
   1. Program has been made to wait till user logs in (Automating the log in has not been done as the password would be stored in plain text)
3. Go to the Bulk assign group tasks page ○○
   1. This is done though going to the url directly
4. Check tasks of a certain “Application Name” (Wintel) ○○
   1. The tasks are filtered to contain xxx application name (currently is set to filter “Non Employee”) similar to below:
   2. Checks all the tasks that are now filtered
   3. A for loop repeats the checking for each row, when the end of table is reached, an error is raised which is then caught and alerts the program that it has reached the end of the table and has nothing else to check
5. Assign the tasks to self and save ○◌
   1. The autocomplete of the name is done via clicking on another part of the screen
6. Go to My delegations -> Create delegation ○○
   1. This is done though going to the url directly
7. Chose person in charge to delegate to ○◌
   1. As of this point, the person delegated to is not taken from elsewhere but typed in (in the code)
   2. The auto complete is also done in the same as 5.i
8. Type “FYA, please” into comments ○○
9. Check the tasks to delegate (same things to check as step 4) ○○
   1. As the tasks filter does not work here, this is done via scanning each row one by one and checking the box if the application name is correct
10. Click save to delegate tasks ○○

Limitations/disadvantages of the current code: *(OLD)*

1. The IDMS page will always open in a new browser, also the browser will always close immediately when the code meets an error or is finished
2. The user must log in manually to stay secure
3. If the url changes the code needs to be updated
4. Should there be application names with the same name/ part of name (ie. “Wintel(IDMS)” and “Wintel(IDMS)xx”) though unlikely, the code will tick both should the filter be for “Wintel(IDMS)”. Also, this is not much faster than a person clicking normally as the check boxes are found one by one, scanning each row individually
5. The autocomplete will sometimes fall to get the name fast enough, thus causing the field to be left blank if the click is too fast. Also, it should be noted that typing in a new line (to represent pressing the ENTER key) does not seem to trigger the autocomplete to take the topmost name as it would have when typing normally
6. Same as 3.
7. Each week someone must update the code to change the person to delegate to. The Autocomplete also has the same issue as 5.
8. Nil
9. As the filter does not work, it will check each row individually, before ticking, this causes the code to be significantly slower as it will check every single row even if it has finished checking all of a certain application name. This is slow as each time it finds a new row, it does a search completely anew.
10. Note that the browser will close 5 second after this is done. (The timing can be adjusted, however, the code will not stop until the end of the time is reached and thus I(MJ) cannot recommend this)

Overall: Xpath is currently used to locate majority elements (the things we interact with) thus should the HTML be changed, there would be a need to update these. Troubleshooting errors might be hard as the browser will close immediately upon having an error encountered in the code. Also, the automation requires the Internet Explorer browser to be focused on, meaning that the usage of the computer for other task while the automation is running is not possible.

For improvement, future development/ RL consistent usage: (OLD)

1. Instead of using Capybara directly, use selenium to start up the browser instance before connecting it to Capybara. (Unsure if this would solve the issue of the browser automatically closing) As for having IDMS open in a new browser window, this is unavoidable until selenium comes up with support for it. (cannot be found as of 13/09/2017)
2. It seems like there is something that allows for automatic IDMS login, see if it can be used with the ruby, capybara code. Otherwise, it should be possible to store the data in an encrypted way, thereby allowing the password to be stored safely, without compromising on security too much.
3. It is possible to make it navigate to the page via the clicking of buttons like a user would do, however, this method is significantly slower
4. In actual use, should the number of tasks filtered out be more than 10, the IDMS table might have a next page as it shows 10 tasks per page. It would be good to check if it is the case and display 50 tasks per page. It should be noted that clicking the next page button will clear any selection that has been made if the save button is not pressed. The checking of checkboxes might be faster if the .all function of capybara is used.
5. This part here can be made more reliable should there be a way to get the autocomplete select the top most option without clicking elsewhere. The best method would be to make it so that the enter key pressed is stimulated. Otherwise, another more reliable solution would be to click on the suggestion given, however, this is hard to do as the element for the autocomplete suggestions cannot be found. (we failed to find it) A suggestion would be to not force the autocomplete for the
6. Same as 3.
7. Same as 5.
8. Nil
9. It would be best if the IDMS filter could work as intended so that the program need not go through every single task on the list. Otherwise, it might be good to make it so that it will stop searching the rest of the tasks once it has selected all the tasks of a certain application name (provided the tasks are sort such that all tasks with the same application name are together). The checking of rows might be faster if the .all function of capybara is used.
10. Browsers in selenium code do not close automatically unlike Capybara

Overall: It would be good if the IDMS could be made such that certain fields such as the Assign to box and the like could have better unique IDs or names in HTML so that the code could be written in a neater way. To allow use of computer while code is running, headless browsers can be used, however although headless browsers are faster and do not require focus, there will not be anything that can be seen to see what is going on in the “browser”