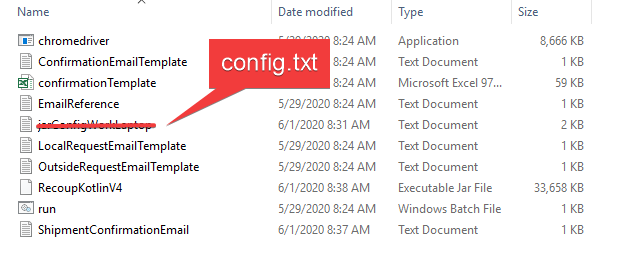
Recoup Automation Tool

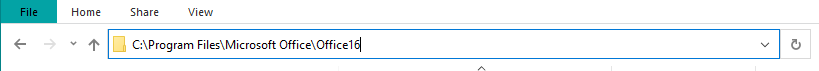
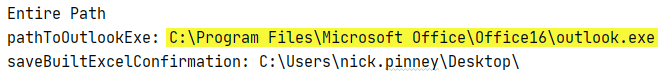
By Nicholas Pinney

**Setup**

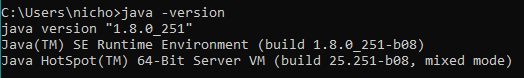
File Structure

* Create a folder anywhere on your PC and copy everything from the drive into that folder.

Config.txt

* In the file name “config.txt” you will need to double check the path to outlook.exe to launch emails. Copy the file path up to the file name from config.txt into the file explorer to see if outlook.exe is in the Office16 folder. See images
* Will also need to specify the path where you want to save the confirmation files to.

Java

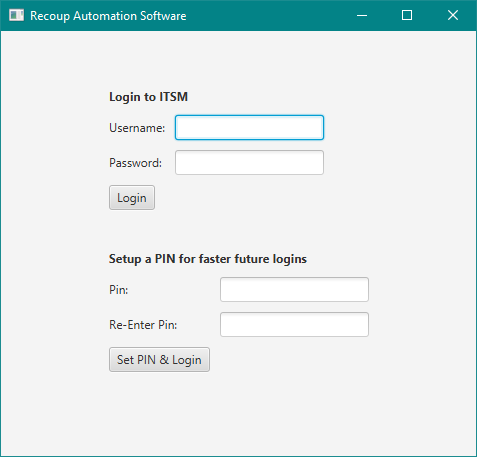
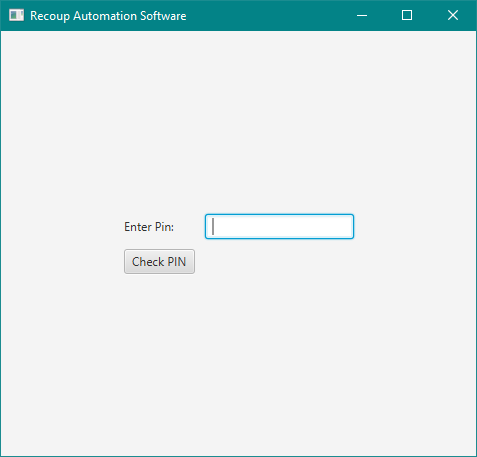
* Make sure java is installed on your PC by opening cmd and running **java -version.**

Run

* Double click on run.bat and program will launch.

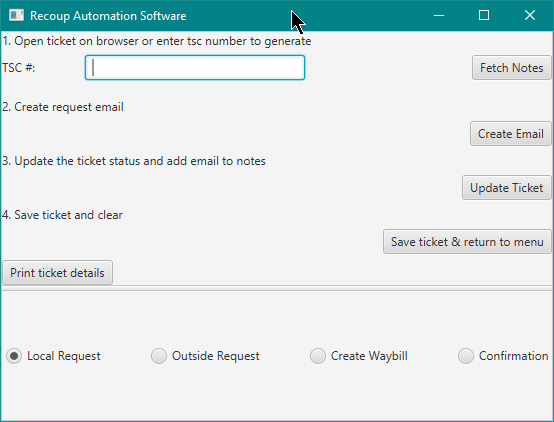
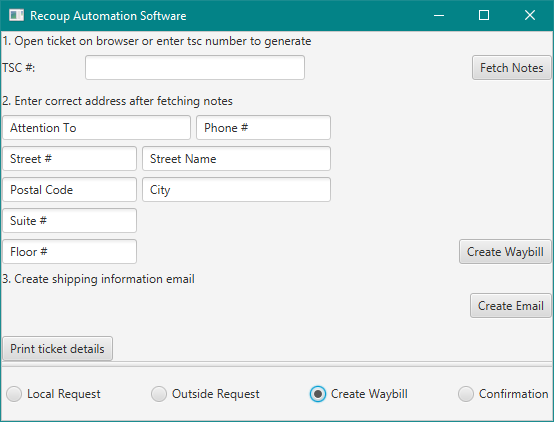
**Recoup Automation Tool**

Login to ITSM using application

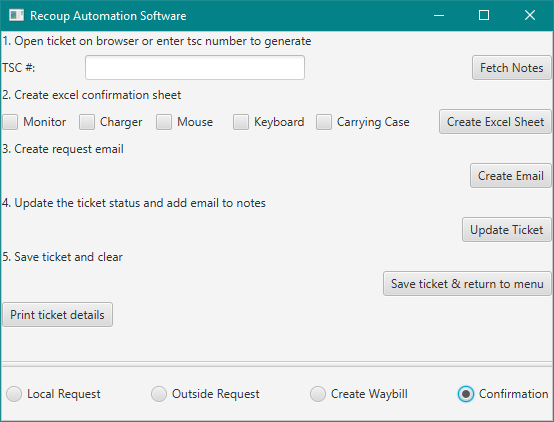
* At first launch after ITSM has loaded, the full login (left) will appear to create pin, store encrypted credentials into Cred.txt and log you in. After wards once Cred.txt is created, pin login screen will appear (right).
* If you’ve set a pin but entered the wrong username/password or forgot your pin, you must only delete the Cred.txt file found in the same folder as the program (I plan on adding another button for this functionality).

Action

* I have developed 3 out of the 4 main actions/tasks related to recoups.
  1. Local requests – Creekbank clients returning their devices.
  2. Outside requests – Clients outside creekbank returning their devices via shipping.
  3. Create Waybill – Not yet developed. To be used once client confirms serial number of returning device and shipping address. Create Waybill button will navigate to Purolator to automate the creation of a waybill based on the information that will be provided from the text fields.
  4. Confirmation – Once client has returned device and both serial number from ticket and device match, buttons are used to create the excel confirmation sheet (based off the confirmationTemplate.xls file found in project directory), generate email and update ticket.
* I have created each action view with steps to make it easy for anyone to understand how to do it. It is also required to follow the steps to get the correct execution of the task... Without fetching any notes, important information cannot be added to an email.

Local Request and outside request are identical in view

Create waybill component only creates email template for now.

Confirmation component with checkboxes to manipulate returned items

**Important features and functionality**

Interaction with ITSM

* 2 functions in program that control ITSM:
  + Fetch Notes Button
    - 2 functionalities:
      1. TSC# typed into the text field before clicking “Fetch notes” will search for the ticket, select the first occurrence found and load the notes into memory. This feature is mostly used for confirmation when hardware with order confirmation is brought back.
      2. Blank TSC# field when clicking “Fetch notes” will assume that you have a ticket open. Will fetch the notes of the ticket in focus.
  + Update Ticket Button
    - Based on the task, the update button will update the ticket accordingly. For example: when on Local or Outside Request, update will put the ticket into “In Progress” whereas when on “Confirmation”, update ticket will put the ticket into “Pending”, “Local Site Action Required”, set the target date field to current date and time and add appropriate notes.
  + Save ticket & return to menu was a button I was planning to add functionality to but I never saw the need for it so it does nothing (plan on removing it).

TSC# Text Field

* After fetching notes, the order # of the return will be displayed in the TSC# text field. This is one way to check if the fetch was successful.
* Some functionality has been added to the text field:
  + When the text field is populated (after a note fetch), 1 click on the text field will turn the border green and another click will clear the text from the field.
  + Green border means that text inside the field has been copied to the system clipboard as if you did CTRL+C (try CTRL+V after it goes green).
  + The second click clears the field to make it easier and more convenient to fetch the next ticket (no need to highlight delete the TSC# or spam the backspace every fetch). **If you navigate to another ticket and forget to clear the text field before clicking “fetch notes”, it will bring you right back to the previous ticket.**

Status Checks

* During execution of program, the command prompt opened from the bat file can be monitored to see successful fetches, full ticket details, etc.
* Print ticket details is a convenient way to check which ticket is loaded. Error will show if no ticket has been fetched when button is pressed.
* If any errors are to occur, they will appear here.

Email Recipients

* The recipients (including choosing CCs) of the emails generated are chosen by the program from the clients mentioned in the notes of the tickets. I’ve formulated statements to properly choose who should be addressed for each action/task.
* John and Sayan are always CC recipients of confirmation emails.

**Customization Through Files**

Templates

* The email that will be generated from the “Create Email” button is based on which action is selected. These emails are derived from an email template stored as text files.
* On email creation, the program will go through the email template and replace <> with the tickets corresponding information.
* Look at OutsideRequestEmailTemplate.txt, the email template can be modified depending on taste or message you’re trying to convey. The first line is the subject field and then the body and signature are below the dashed lines. **The lines must stay and not be altered so the program can properly parse the templates.**
* EmailReference.txt can be viewed to see a collection of available tags for the email templates.
* confirmationTemplate.xls is the basic template used to record the hardware returned. On top of specifying which hardware was returned, the program also updates the date field and TSC of the template. TSC of the currently fetched ticket is used.

Config

* config.txt is used to keep track of where these crucial files reside in the file system.
* Easiest if everything is kept in the same folder.

**Restrictions/Known Bugs**

Fetching Notes

* When program is accessing the website, the program will not respond until task on website is complete. Be patient, it will finish.
* Entering the wrong pin (or no pin) when trying to pin login will throw an error.
* If outlook is not already open when creating an email, then program will be frozen for as long as the email created remains open.
* Will encounter an error if an email is created while a previous created email is still open.
* When logging in, if the credentials are incorrect the Cred.txt file will still be created – it will need to be deleted.
* Clicking print ticket details before fetching a ticket will cause error.
* Not being in a ticket or have a number in the field when trying to fetch notes will cause an error.
* Many others

Notes

A lot of these conditions I’ve known for a long time now and are easily avoidable and are still aspects of the application that I have not yet finished. Though you are seeing a close version of the finished product, I still have plans to add many more error checks, interface changes, visual updates throughout task executions and many more to further improve functionality.

I didn’t explain everything as I did this in a bit of a rush and late at night so please ask me questions if you have any and let me know what you think.

Thank you,

Nick