

Portfolio Submission Zero

Abdul Mobarak

Executive Officer (E0)

DWP DevOps Engineer Apprenticeship

Department for Work & Pensions

# Contents

[Contents 2](#_Toc116893086)

[1 Introduction 4](#_Toc116893087)

[1.1 About Me 4](#_Toc116893088)

[1.2 About My Team 4](#_Toc116893089)

[1.3 Normal Working Day 5](#_Toc116893090)

[1.4 Responsibilities and Obligations 5](#_Toc116893091)

[1.5 Working Environment 5](#_Toc116893092)

[1.6 Training Undertaken 5](#_Toc116893093)

[2 Evidence List – Knowledge Skills and Behaviours (KSB’s) 6](#_Toc116893094)

[2.1 Knowledge 6](#_Toc116893095)

[2.2 Skills 7](#_Toc116893096)

[2.3 Behaviours 8](#_Toc116893097)

[3 Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts 9](#_Toc116893098)

[3.1 Situation 9](#_Toc116893099)

[3.2 Task 9](#_Toc116893100)

[3.2.1 Child Issues of Epic 10](#_Toc116893101)

[3.3 [Jira Issue – BCM-678] Action – Create Python Code to Identify Redundant Users - Minimum Viable Product (MVP) 11](#_Toc116893102)

[3.3.1 [Jira Issue - BCM-679] Write Code to Read User List File 11](#_Toc116893103)

[3.3.2 [Jira Issue - BCM-680] Write Code to Identify Redundant Users 14](#_Toc116893104)

[3.3.3 [Jira Issue - BCM-681] Write Code to Read Single Project User Request (PUR) Spreadsheet 15](#_Toc116893105)

[3.3.4 [Jira Issue - BCM-682] Write Code to Read Set of PUR Spreadsheets 16](#_Toc116893106)

[3.3.5 [Jira Issue - BCM-683] Write Code to Create User Suspension Script for StarTeam Command Line 17](#_Toc116893107)

[3.4 Additional Functionality (Addition to MVP) 19](#_Toc116893108)

[3.4.1 Config File 19](#_Toc116893109)

[3.4.2 Makefile 20](#_Toc116893110)

[3.4.3 HTML Report 21](#_Toc116893111)

[3.4.4 Improvements and Fixes 22](#_Toc116893112)

[3.5 Results 23](#_Toc116893113)

# Introduction

## About Me

My name is Abdul Mobarak, and I am very fortunate to be part of DWP’s DevOps Engineer Apprenticeship scheme, which is delivered by Makers. I chose to apply and become a DevOps Engineer because I enjoy problem solving, creative solutions and engineering. I studied Graphic Design at University and enjoy gaming and DIY. I believe that becoming a DevOps Engineer will allow me to bring out the best of my skills and help me grow as an individual, as well as build an enjoyable career which I hope will help me to provide value to DWP.

## About My Team

My team’s name is the Software Release Team. My team is part of DWP’s Digital Group and sits under Hybrid Cloud Services which is part of Technology Services as per the below Organisation Chart.

Diagram, table

Description automatically generated

Figure - Organisation Chart

My teams purpose is to provide DWP with efficient, secure and repeatable tools and/or assistance for transferring software and packages from repositories and vendors into the operation estate. This is achieved by providing both process and tooling support. Some of my team’s other activities include:

* Taking formal receipt of third-party software deliverables from suppliers
* Tool administration e.g., Micro Focus StarTeam and some file transfer services
* Supporting Labour Market System (doing software builds etc.) and Personal Independence Payments (branching and merging etc.)

## Normal Working Day

A normal working day in my role consists of working on tickets, taking up learning, and daily stand-up meetings with my team. My team stand-up meetings are usually 11am every day which is perfect for discussing work and issues. On Monday to Thursday, I work on tickets and complete outstanding tasks including learning courses, consulting with my manager when necessary as he kindly provides me great support. I also have weekly 1-2-1 meetings with my manager. On Fridays I have my dedicated time to complete apprenticeship activities.

## Responsibilities and Obligations

In my role I am required to abide by rules and regulations set by DWP and the wider government. These include my obligations to Health and Safety, General Data Protection Regulation (GDPR), etc. To ensure I am aware of my responsibilities I am required to complete mandatory training every year or two, to ensure my understanding is kept up to date. As my role may involve building capabilities and components that affect services in DWP internally and externally, it is crucial that I abide by and follow guidelines, therefore this training is invaluable to the work I do now and in the future. My role also requires for me to practice DWP values which are the following: We care, we deliver, we adapt, we work together, and we value everybody. In short, ensuring I am inclusive of all people, and ideas, ensuring that I deliver my expectations and help others is crucial.

## Working Environment

My role consists of hybrid working, from home and the office. I am based in Manchester and my office is just a few miles away from home. However, my team is based in Newcastle, this means that a digital communications approach is vital to keep in touch and up to date. We use Slack and Microsoft Teams mainly for communication.

## Training Undertaken

During my apprenticeship at DWP it is necessary for me to upskill and carry out training for different reasons. Some training is required simply due to the training being mandatory for all staff in DWP, for example, Health and Safety. Additionally with guidance and help from my manager I will be undertaking training relevant to my role and apprenticeship to allow me to improve my skills and knowledge to help deliver objectives within my team and the wider department.

|  |  |  |  |
| --- | --- | --- | --- |
| Training Course | Relevance | Time Spent | Completion Date |
| Software Development Bootcamp | Initial bootcamp training provided by Makers. This bootcamp provided me with basic Software Development knowledge to supplement my DevOps apprenticeship | 10 weeks | 9th March 2022 |
| DevOps Engineering Bootcamp | Further bootcamp training provided by Makers to give me a basic overview and idea of DevOps before starting my role in DWP as an apprentice. | 8 weeks | 6th May 2022 |
| Personal Health and Safety | Mandatory online training required for all DWP staff. |  | 23rd May 2022 |
| Display Screen Equipment | Mandatory online training required for all DWP staff. |  | 18th May 2022 |
| Think Secure: Security & Data Protection at DWP 2022 | Mandatory online training required for all DWP staff. |  | 7th June 2022 |
| Fraud Error and Debt Awareness in DWP 2020-2021 | Mandatory online training required for all DWP staff. |  | 7th June 2022 |
| Public Sector Equality Duty | Mandatory online training required for all DWP staff. |  | 24th May 2022 |
| Introduction to Python Scripting | Python Scripting Introduction training on A Cloud Guru to support my first *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts.* | 6 hours | 01/07/2022 |
| Python Object-Oriented Programming Basics | Python Object-Oriented Programming Introduction training on A Cloud Guru to support my first *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts.* | 2 hours | 11/07/2022 |

Table – Table of Training Undertaken

# Evidence List – Knowledge Skills and Behaviours (KSB’s)

This section is to demonstrate how I have met the KSB’s within different tickets.

## Knowledge

Table – Table of Knowledge Criteria

|  |  |  |
| --- | --- | --- |
| Item | Description | Evidence |
| K3 | How to use data ethically and the implications for wider society, with respect to the use of data, automation, and artificial intelligence within the context of relevant data protection policy and legislation. |  |
| K6 | A range of problem-solving techniques appropriate to the task at hand, such as affinity mapping, impact maps, plan-do-check-act/Deming. |  |
| K18 | Roles within a multidisciplinary team and the interfaces with other areas of an organisation. | *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts* |
| K19 | Different methods of communication and choosing the appropriate one - e.g. face-to-face (synchronous, high bandwidth), instant messaging, email (asynchronous, low bandwidth), visualisations vs. words. |  |
| K20 | Pair/mob programming techniques and when to use each technique. | *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts.* This ticket demonstrated pair programming. The intention is to demonstrate mob programming on a future ticket. |
| K22 | How their occupation fits into the wider digital landscape and any current or future regulatory requirements. |  |
| K23 | The importance of continual improvement within a blameless culture. |  |
| K24 | The difference between Software-as-a-Service (SaaS) v bespoke v enterprise tooling and how to make an informed choice that suits each use case. |  |
| K25 | Maintain an awareness of cloud certification requirements. |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Skills

Table - Table of Skills Criteria

|  |  |  |
| --- | --- | --- |
| Item | Description | Evidence |
| S1 | Communicate credibly with technical and non-technical people at all levels, using a range of methods; e.g. ‘Show and Tell’ and ‘Demonstrations’. |  |
| S2 | Work within different organisational cultures with both internal and external parties |  |
| S4 | Initiate and facilitate knowledge sharing and technical collaboration |  |
| S8 | Work in agile, multi-disciplinary delivery teams, taking a flexible, collaborative and pragmatic approach to delivering tasks. | *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts* |
| S13 | Engage in productive pair/mob programming. |  |
| S16 | Invest in continuous learning, both your own development and others, ensuring learning activities dovetail with changing job requirements. Keep up with cutting edge. |  |
| S21 | Application of lightweight modelling techniques, such as whiteboarding, in order to gain consensus as a team on evolving architecture. |  |

## Behaviours

Table - Table of Behaviours Criteria

|  |  |  |
| --- | --- | --- |
| Item | Description | Evidence |
| B1 | Exhibits enthusiasm, openness and an aptitude for working as part of a collaborative community; e.g. sharing best practice, pairing with team members, learning from others and engaging in peer review practices. | *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts* |
| B2 | Invests time and effort in their own development, recognising that technology evolves at a rapid rate. |  |
| B4 | Is inclusive, professional and maintains a blameless culture. | *Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts* |

# Ticket 1 [Jira Epic BCM-656] – Develop tooling for auditing Micro Focus StarTeam user accounts

## Situation

The purpose of this project is to develop tools to review StarTeam users to effectively manage all users, removing redundant users and updating existing users with correct details. This is crucial to maintain secure access control and ensure that future migration possibilities can be considered and implemented with ease. Other benefits include developing and improving my Python and Unix skills, team work and collaboration, and a better understanding of how DWP Digital teams operate.

## Task

Diagram

Description automatically generated with medium confidenceOur team oversees access and control of the StarTeam server application. The Data Platforms Team support the database. The Windows Server Team support the virtual servers that the application and database run on. A range of other teams support the underlying infrastructure e.g. networks.My task for this ticket is outlined in section 3.2.1 Child Issues of Epic. I will be pair-programming with my manager throughout the entire project, learning and building my skills with my manager’s support. I will be reviewing documentation and upskilling during the project to ensure my understanding is clear for each aspect and component. For the most part, my manager will be operating as the navigator and myself as the driver which is an aspect I practiced in the Bootcamp with Makers. My manager will point me in the right direction and advise me on best practices. My manager has outlined the project on Jira as a ticket and created a diagram to clarify the requirements and monitor progress, ensuring that we can meet the objectives required. Throughout the project, dummy data will be used that matches the format of the real data, this is to ensure GDPR obligations are met and that experiments can be made until otherwise necessary.

Figure – StarTeam User Audit Process Diagram

Text

Description automatically generated

Figure - StarTeam User Audit Ticket Overview

### Child Issues of Epic

Below is the current table of tickets for Project 1 and the status of each ticket. Each ticket is broken down into actions required to fulfil each ticket.

Table - Child Issues of StarTeam User Audit Epic

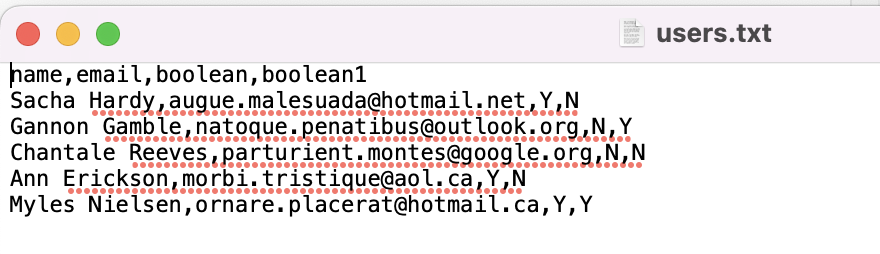
|  |  |  |
| --- | --- | --- |
| Ticket Name | Description | Status |
| BCM-679 | Write Code to Read User List File | **Completed** |
| BCM-680 | Write Code to Identify Redundant Users |
| BCM-681 | Write Code to Read Single Project User Request (PUR) Spreadsheet |
| BCM-682 | Write Code to Read Set of PUR Spreadsheets |
| BCM-683 | Write Code to Create User Suspension Script for StarTeam Command Line |

## [Jira Issue – BCM-678] Action – Create Python Code to Identify Redundant Users - *Minimum Viable Product (MVP)*

### [Jira Issue - BCM-679] Write Code to Read User List File

#### Write standalone Python code to read in text file and print lines to console

To start the project, I used a [free online dummy data generator](https://generatedata.com/generator) to come up with a fake set of users to practice with the  [Python CSV Module](https://docs.python.org/3/library/csv). I stored the users inside a text file. I imported the CSV module to read in users. Using the CSV module, I printed out each line of data to the console. This step would help me further when I come to reading the users data as a CSV file. I reviewed the code with my manager who suggested it would be better encapsulated in a class. We then began pair-programming using the driver-navigator approach to tackle the ticket.

Text

Description automatically generated

Figure - Users Sample Text File

Figure - Read in Text File

#### Create skeleton class to hold information about StarTeam user

**Text

Description automatically generated**I created a skeleton class named StUser to hold the data for each individual user. The values were set to false on initialisation. This skeleton class represents the data that we will be reviewing from StarTeam.

Figure - StUser Skeleton Class

#### Add unit test code to skeleton class

*Text

Description automatically generated*After creating the skeleton class, I created tests using the [Python Unit Test module](https://docs.python.org/3/library/unittest.html). I followed the Python documentations method of creating Set Up and Tear Down helper methods that run before and after each test. This allowed me to re-create objects to ensure the tests run independently without conflicts from each other. I checked that the skeleton class was holding the user’s data as expected.

Figure - StUser Unit Test

#### Create code for parsing user list file into collection of user objects + run unit tests

I created a new class named StUserCollection which would parse each user with the help of the StUser class into the user\_dict object in StUserCollection. The StUserCollection class contains a module named populate\_from\_csv which I designed to open and review the CSV file ensuring the format is as expected, if not errors would be thrown. Using Pythons custom exception handling I was able to print out messages to the console to make it easy for debugging and future users.

I carried out unit tests to ensure that the populate CSV function was working as expected, returning the correct user counts as well as adding in test users to ensure the StUserCollection class was working alongside the StUser class.

Text

Description automatically generatedText

Description automatically generated

Figure - StUserCollection CSV Unit Tests

Figure - StUserCollection populate\_from\_csv

### [Jira Issue - BCM-680] Write Code to Identify Redundant Users

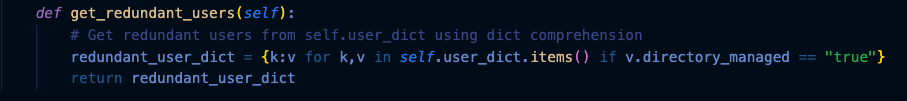
After ensuring my script can read the CSV file I moved on to reviewing users whom would be considered as redundant. I was informed by my manager that any user that were flagged as directory managed in the CSV file would be classed as redundant. This led me to create a simple method get\_redundant\_users which would add all redundant users to a dictionary named redundant\_user\_dict. Whilst coding this method, my manager taught me the basics of [dictionary comprehensions.](https://www.freecodecamp.org/news/dictionary-comprehension-in-python-explained-with-examples/)

Figure - Redundant Users(directory managed)

I then completed the module with a simple test to confirm that the users flagged as directory managed would be collected in redundant\_user\_dict.

Text

Description automatically generated

Figure - Redundant Users Unit Testing

### [Jira Issue - BCM-681] Write Code to Read Single Project User Request (PUR) Spreadsheet

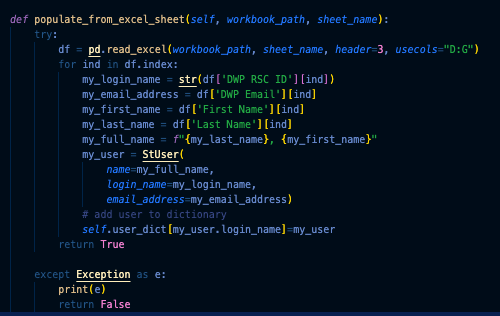
After successfully reading and filtering the StarTeam User data(CSV) I turned my attention to the PUR Spreadsheets which are used to manage access requests of users to compare them against the StarTeam User list. I used the [Pandas Python package](https://pandas.pydata.org/docs/getting_started/index.html) which is very handy for reading in Excel files with minimal code. Using the StUser populate\_from\_excel\_sheet method I was able to read in each user contained in the spreadsheet and store them into the user\_dict object.

Figure - StUser populate\_from\_excel\_sheet method

Text

Description automatically generatedI then completed the method with some unit tests. During testing of the method, I found that the tests were failing. After using a handy trick to print out the expectation vs results using the Assertion Error catch, I found that Python was storing some text as a data type of number because the text contained numerical values for staff ID’s. This was a simple fix as seen above by wrapping the ID’s with the str method, which converts the numbers to a string format.

Figure - StUser populate\_from\_excel\_sheet unit tests

### [Jira Issue - BCM-682] Write Code to Read Set of PUR Spreadsheets

Text

Description automatically generatedNow that my script can successfully read a single excel sheet, the next step was for me to program reading multiple excel sheets in separate files. I created a for loop to cycle through each workbook using the workbook\_path\_list argument in the populate\_from\_multiple\_excel\_sheets method. Within this method I created an if statement to check if the user already exists on the spreadsheets, if so, check for any differences and store them inside a dictionary, so that we can view users whose data has conflicts within each set of spreadsheets.

Figure - Populate from Multiple Excel Sheets & User Note Methods

Text

Description automatically generatedFinally, I added unit tests to ensure the reliability of the code. During the debugging of the tests, it was hard to establish which tests were failing due to creating many tests during the project, therefore I wanted to improve visibility of my unit tests by outputting more information on the console. I added a print statement which outputs the test method name if the test failed. The code for this handy trick was obtained from the following [resource](https://www.oreilly.com/library/view/python-cookbook/0596001673/ch14s08.html).

Figure - Multiple Excel Sheets & User Note Unit Tests

### [Jira Issue - BCM-683] Write Code to Create User Suspension Script for StarTeam Command Line

#### User Suspension Script

A screenshot of a computer

Description automatically generated with medium confidenceThe final ticket for the MVP for the StarTeam User Audit Tool project was to create a user suspension script that can be ran for the StarTeam Command Line. This script would remove the redundant users identified in earlier steps. I created the method below which would be called when running the audit tool. Some of the properties of the command are hard coded to connect to the StarTeam interface. I used a for loop to loop over the redundant users returned when running the tool to compile a list and print out the suspension script lines.

Figure - User Suspension Script Method

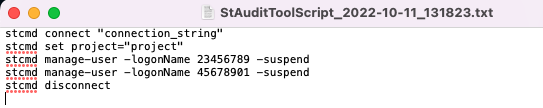


Figure - User Suspension Script

#### Audit Tool Run

Finally, I created the run method which is the method that pieces together all the classes and methods to output the results of this project. I began by creating a variable named desktop to dynamically store the user’s Desktop direct path. This would be useful to output and store the files which we will create when the tool is ran. I then created dynamic file names for the deliverables which would include the current time and date. Once the file names were created, I checked the desktop to see if an existing output folder exists, if not, one will be created. I then ran the StUserCollection class and identified redundant users. In this directory two files are outputted when the tool is called, the User Suspension Script and an additional file that will output a HTML Report that will detail all the relevant information gathered.

Text

Description automatically generated

Figure - StarTeam Audit Tool Run Method

## Additional Functionality (Addition to MVP)

During and after the project, I made improvements to certain aspects of the script(s) to improve the usability of the development. These improvements were not necessary however allowed me to push my skills further and ensure that I was satisfied with my deliverables.

### Config File

My manager taught me about configuration files which can help manage variables. I created a config file which could be used within my scripts. This allows for easy visibility and tweaking for future users if for example the path for a certain file was to change because the user chose to use a different directory.

Text

Description automatically generated

Figure - Config File

Text

Description automatically generatedI read the config file into my script using the configparser module. A try method was implemented to catch any errors when reading the config file. Variables were then created to store the variables of the config file into memory so that they can be used in other modules.

Figure - StarTeam Script Reading In Config File

### Makefile

Another handy utility my manager taught me about was the Makefile. My manager showed me how to implement the Makefile and access it within a virtual environment to run repeatable commands with ease. For example to run our tests I just have to enter the virtual environment via the command line and run “make test” which will run the commands under the test task. This is particularly useful when sharing or exporting the tool for others to use as they don’t need to remember the commands. Multiple commands can also be pieced together which also has a time saving aspect.

Text

Description automatically generated

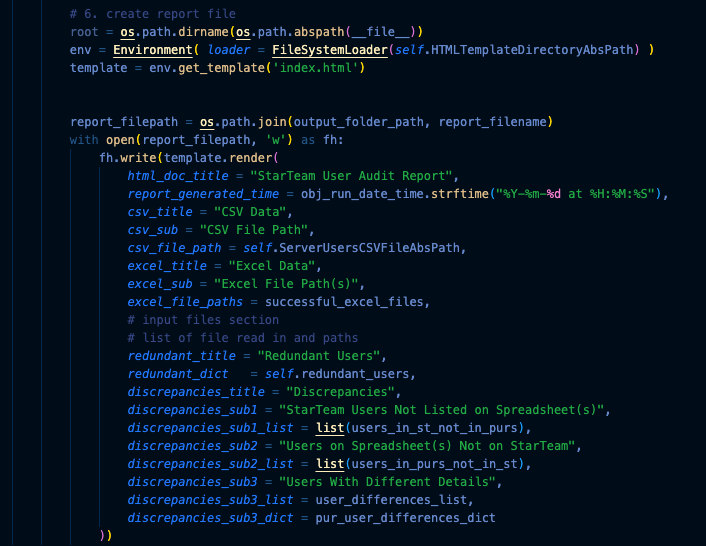
Figure - Makefile

### HTML Report

Text

Description automatically generatedAnother additional functionality I programmed was a HTML Report. I designed the HTML report to contain all the relevant steps the tool took in the run method. It lists the files I used in the tool, any redundant users, and discrepancies. The HTML formatting was managed with the help of [Bootstrap](https://getbootstrap.com/docs/5.1/getting-started/introduction/) for easy styling. In order to dynamically insert the data into the HTML report, [Jinja2](https://jinja.palletsprojects.com/en/3.1.x/) was used create a HTML template so that the Python code can interact with the document and insert the data. Although the HTML report was not listed in the MVP, it is extremely helpful and useful for the tool to give the user visibility easy to read results.

Figure - HTML Template



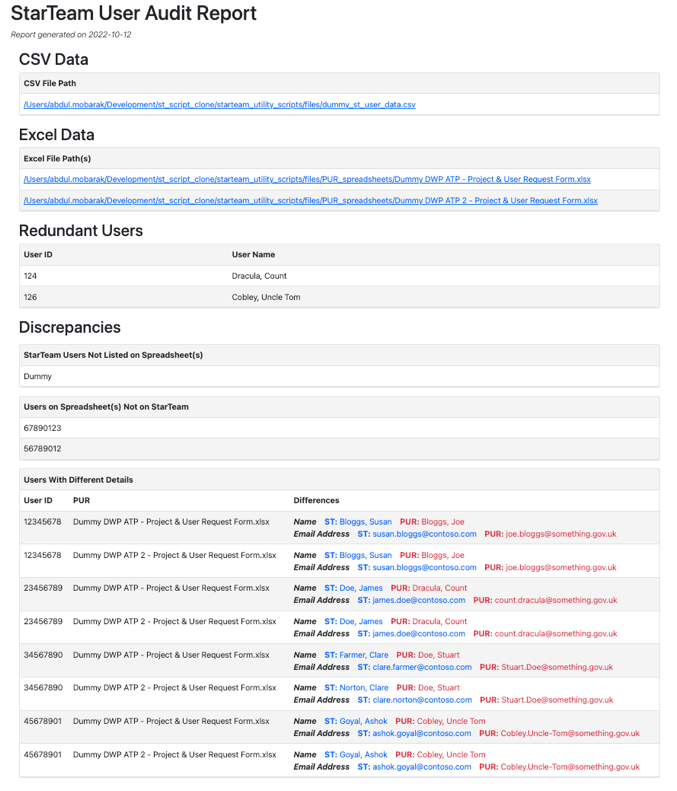


Figure - Jinja 2 Data Templating

Graphical user interface, text, application, email

Description automatically generated

Figure - Desktop Output Folder for Report and Script

Figure - HTML Report Output

### Improvements and Fixes

Throughout the ticket, I made improvements to the tool ensuring that various components were updated and improved, such as ensuring all unit tests were passing, 100% test coverage was implemented, linting and efficiency of the code. As the project lasted several weeks, it would not be possible to demonstrate all the improvements and fixes however some examples will be shown below.

#### Testing

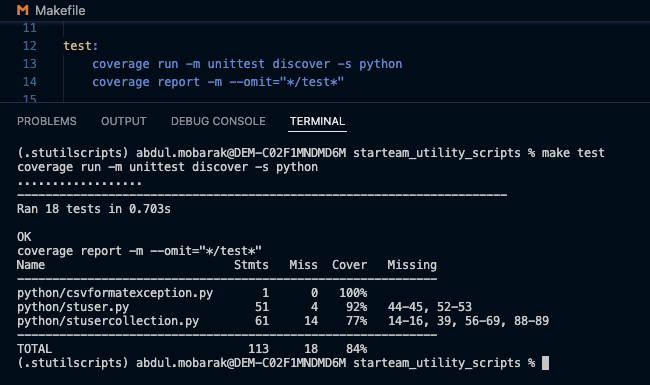
Using the Python [Coverage](https://coverage.readthedocs.io/en/6.5.0/) module, I was able to identify methods and code that was not targeted by my unit tests. The coverage command was ran via the Makefile for simplicity and repeatability. I spent some time towards the end of the project to improve the testing coverage, ultimately reaching 100% coverage.

Figure - Makefile and Test Coverage Low

#### 

Figure - Test Coverage 100%

Figure - Test Coverage Improved

#### Text Description automatically generatedGeneral Improvements

Figure - Commits

Throughout the project, my manager and I made many considerations for each aspect of the tool, from where we will store the user list files, to what methods to write, and what methods or libraries we will use within the project. I learnt a lot of techniques and applications alongst the way and really enjoyed completing this ticket. With the help of Git it’s easy to see the progress and improvement I made on the ticket with the helpful commit messages. This has been particularly useful when needing to recap and review code and changes we made.

## Results

As a result of several weeks of pair-programming with my manager, I am very pleased to confirm that the audit tool is complete, and the MVP requirements were met with extra functionality. I found the ticket challenging but enjoyable as my manager was hugely supportive and helped me take critical directions throughout the ticket. I have pushed my programming my skills further as well as other skills such as teamwork and research. More importantly the business benefits are advantageous as now a clearer picture can be made of the current up to date users on the StarTeam application, management of users going forward will be more maintainable, migration off StarTeam can be managed and security controls can be implemented.