**TEST PLAN**

|  |  |  |  |
| --- | --- | --- | --- |
| **What am I testing?** | **Expected Outcome** | **Actual Outcome** | **Changes Made** |
| Login – user\_id | Accepting the input. Allowing the text file to be opened and proceed through to PIN entry.  Should an incorrect user\_name be entered, the user will be prompted to re-enter correct ID | The same | None  **Notes:** if an incorrect value or data type is added, the code will restart the login process. |
| Login - pin | Accepting pin based on the value stored in the text file.  If PIN is incorrect, the user should get three tries. Failing for a third time will resort in restarting the login process. | The same | None  **Notes:**   * I’d like to make the input appear as “\*” rather than as the integers. * If an incorrect value is added for a 3rd time the code will restart. |
| Menu Function | Upon pressing either of the 5 options; [D], [W], [B], [T] or [E] the program should successfully go through to the selected options screen.  Selecting [E] would successfully exit the program and produce an exit message “Goodbye” | The same.  If an option, character/integer is entered, the code resorts back to the main menu stating that a correct option from the menu should be picked. | None  **Notes:**   * I’d use Tkinter for any sort of advanced project. As this was a demonstration of our understanding so far I thought it fitting to build without a user interface. |
| Deposit Function [D] selection | Upon entering the deposit amount, the function checks which user is logged in by using the global variable user\_id and then proceeds to update balances within the users text file.  If a character is entered the Value Error corrects this by asking for an Integer and brings the user back to the main menu.  The deposit amount is entered into a list along with a time stamp. | The same | None  **Notes:**   * If I did it over, I’d make it ask for a deposit amount again rather than revert back to the main menu. |
| Withdraw Function [W] | Upon entering the withdrawal amount, the function checks which user is logged in by using the global variable user\_id and then proceeds to update balances within the user’s text file. If the user tries to withdraw more than they have in their overdraft. The message “Insufficient Funds” will appear. Then take the back to the main menu.  If a character is entered the Value Error corrects this by asking for an Integer and brings the user back to the main menu.  The deposit amount is entered into a list along with a time stamp. | The same | None  **Notes:**   * If I did it over, I’d make it ask for a deposit amount again rather than revert back to the main menu. * I could not add the functionality to clear the screen when withdrawing like the deposit function. I would add this next time after investigating how to do so. |
| Balance [B] | If the user selects this option. The users file is opened using the global user\_id and shows the balance along with the time stamp. | The same | None  **Notes:**   * I initially had this within a function but thought this was not needed. |
| Transactions Function [T] | Upon selecting this function, the user is greeted with a further selection of seeing Withdrawals [W] or Deposits [D].  Selecting either brings up the D list or W list which contain a list of previous transactions used within that programs run. | The same | Changed the two lists into one called “T” to mimic a normal banking system.  **Notes:**   * I would have liked the ability to write this to the file. Would have been a simple fix but only thought about this now and I think is associated with Level 3. * Merging the lists would also have been easy as I already have the “-“ and “+” signs. * On reflection, this wouldn’t have needed it’s own function either. Same problem with os.system(‘cls’) |
| Exit [E] | Raise system exit with a “Goodbye”message | The same | None |
| Bank system function | Loop works well and only ends when exit is selected | The same | None |