Bangabandhu Sheikh Mujibur Rahman Digital University

COURSE TITLE & CODE: OBJECT ORIENTED PROGRAMMING (ICT 4251)

Project report on Bank Reconciliation System using Python

Submitted To:

Farzana Akter

Lecturer, Internet of Things (IoT) Program,

Department of Information & Communication Technology (ICT)

Faculty of Engineering,

Submitted By:

Group No - 14

Anima Zaman Proma
(ID:1901017)

Tasnima Hamid
(ID: 1901043)

Session: 2019-20

Second Year First Semester,

Internet of Things (IoT)Program,

Department of Information & Communication Technology (ICT)

Faculty of Engineering

Submission Date: 19th March 2022.

Acknowledgements:

We would like to express our sincere gratitude to several individuals for supporting us throughout the project. First, we wish to express our sincere gratitude to our teacher, Farzana Akter, for her guidance, insightful comments, helpful information, practical advice and unceasing ideas that have always helped us tremendously in our project. We also wish to express our sincere thanks to our honourable teachers, our families and our friends who were with us all the time.

Abstract:

The goal of a bank reconciliation is to reconcile the difference between the cheque issued balance and the bank statement balance. These should be equal for accurate accounting records and prevent potential theft or misuse. This procedure is maintained by humans which is time consuming and erroneous. This can also be manipulated by dishonest people and hard to find because of huge number of accounts. To eradicate these problems, we automated the whole process. Using Python along with several modules namely, Pandas, NumPy, Tkinter and Tabula-py, the program compares the two amounts and categorizes them. It also saves a csv file with the final remarks and presents the results to the user using graphical user interface.

Table of Contents

TABLE OF FIGURES

TABLE OF TABLES

Introduction

Bank Reconciliation Statement (BRS) is a very convenient way to maintain accounts. Reconciliation is the process of matching internal records of transactions against internal external sources. Reconciliation makes sure that accounting records are accurate, by detecting bookkeeping errors and fraudulent transactions. The difference may sometimes be acceptable due to the timing of payments and deposits, but any unexplained differences may point to potential theft or misuse of funds. This project will help to reconcile accounts without mistakes and we will be able to cheque vast number of records with it.

Literature Review

The term "reconciliation" refers to the process of bringing two sets of amounts correspond with each other (i.e., making them equal) by explaining why they differ. The bank balance in the passbook and the bank account maintained by the company differs. Many factors contribute to these differences. The purpose of a bank reconciliation statement is to try to figure out what's causing these differences and, if required, take corrective action. Generally, this is done by humans, which takes a lot of time and can lead to erroneous calculations or assumptions. We tried to automate the whole process using python. This checks the cheque issued amount and bank paid amount against the cheque and writes the result in a very short time than the usual process. Thus, it saves time and gives proper calculation. It represents the data using Tkinter which is easy to use and the user doesn’t need to check the programming terminal output which can be a little difficult for people. Our project also stores the data in a csv file for future/further calculations.

Problem definition and requirement analysis

Problem Statement

Bank reconciliation statements can be used to explain why there are inconsistencies in transactions made by the bank or the organization and to find errors and omissions in both papers so that they can be corrected as soon as possible. These happens because there may be unpresented cheques, dishonoured cheques and payment made due to uncredited items can lead to a mismatch. Bank reconciliation statement is necessary because of these. Bank Reconciliation is now done by humans. A person needs to compare all the records manually then write the remarks and calculate the whole thing. It can lead to error. It can also be huge time consuming.

Requirements

This system requires modules

Category	Name	Version	Purpose
Modules	Tkinter	8.6	For user interface
	Numpy	1.19.5	For computing array
	Panda	1.4.1	For using data frame and working with data
	Tabula	2.3.0	For converting pdf into csv
Language	Python	3.9	For writing the script
Source-Code Editor	Visual Studio Code/PyCharm		
Operating System	Windows	10	

Table 1: Requirements

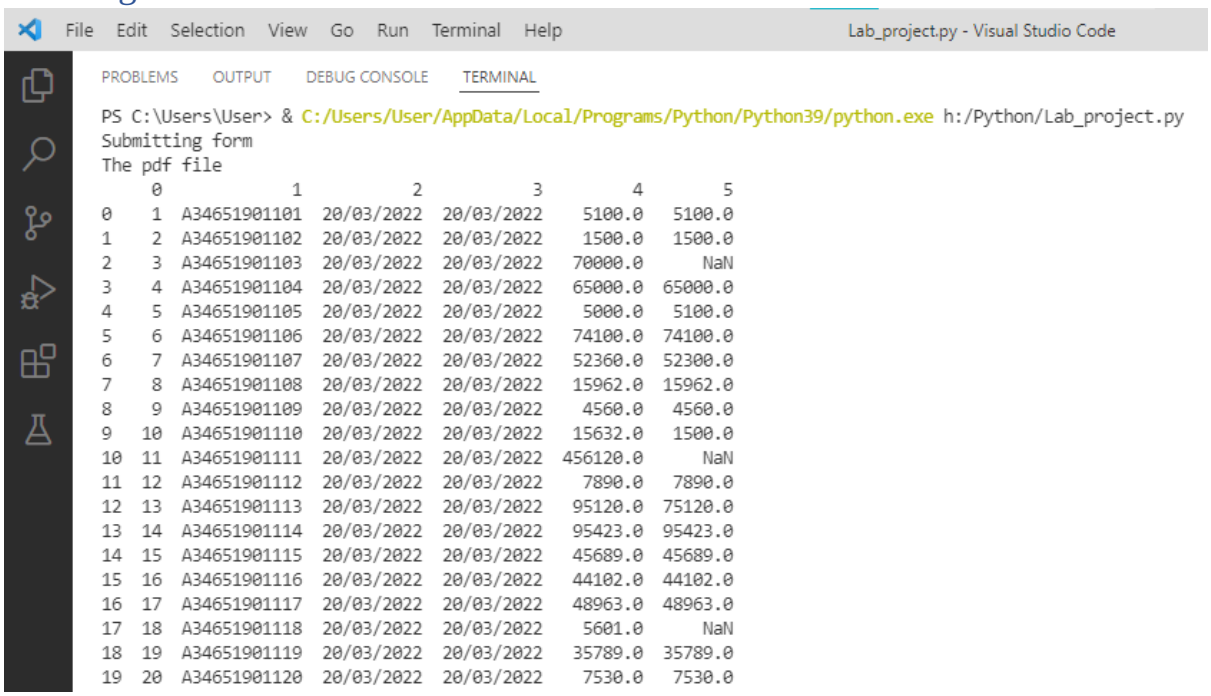
Design and Implementation

Algorithm:

The algorithm works as follows-

- The program takes the path of the pdf file from user.
- Then it converts the contents of the pdf file to csv file.
- Then it compares the issued amount and bank paid amount against each cheque.
- If the amounts are same, it writes Ok as remarks.
- Else it checks if the bank paid amount is lesser or greater than issued amount. If so, it writes 'To be reconciled' as remarks and stores/appends the cheque no in 'recon_acc' list.
- Otherwise, it writes 'No value' as remarks because of missing value or inappropriate value.
- Then it stores the remarks along with the data in a csv file named 'bank_copy'.
- It also calculates the total issued amount, total bank amount and total reconciled amount and shows the result in text widget.

Working Process:



	0	1	2	3	4	5
0	1	A34651901101	20/03/2022	20/03/2022	5100.0	5100.0
1	2	A34651901102	20/03/2022	20/03/2022	1500.0	1500.0
2	3	A34651901103	20/03/2022	20/03/2022	70000.0	NaN
3	4	A34651901104	20/03/2022	20/03/2022	65000.0	65000.0
4	5	A34651901105	20/03/2022	20/03/2022	5000.0	5100.0
5	6	A34651901106	20/03/2022	20/03/2022	74100.0	74100.0
6	7	A34651901107	20/03/2022	20/03/2022	52360.0	52300.0
7	8	A34651901108	20/03/2022	20/03/2022	15962.0	15962.0
8	9	A34651901109	20/03/2022	20/03/2022	4560.0	4560.0
9	10	A34651901110	20/03/2022	20/03/2022	15632.0	1500.0
10	11	A34651901111	20/03/2022	20/03/2022	456120.0	NaN
11	12	A34651901112	20/03/2022	20/03/2022	7890.0	7890.0
12	13	A34651901113	20/03/2022	20/03/2022	95120.0	75120.0
13	14	A34651901114	20/03/2022	20/03/2022	95423.0	95423.0
14	15	A34651901115	20/03/2022	20/03/2022	45689.0	45689.0
15	16	A34651901116	20/03/2022	20/03/2022	44102.0	44102.0
16	17	A34651901117	20/03/2022	20/03/2022	48963.0	48963.0
17	18	A34651901118	20/03/2022	20/03/2022	5601.0	NaN
18	19	A34651901119	20/03/2022	20/03/2022	35789.0	35789.0
19	20	A34651901120	20/03/2022	20/03/2022	7530.0	7530.0

Figure 1: Reading the csv file.

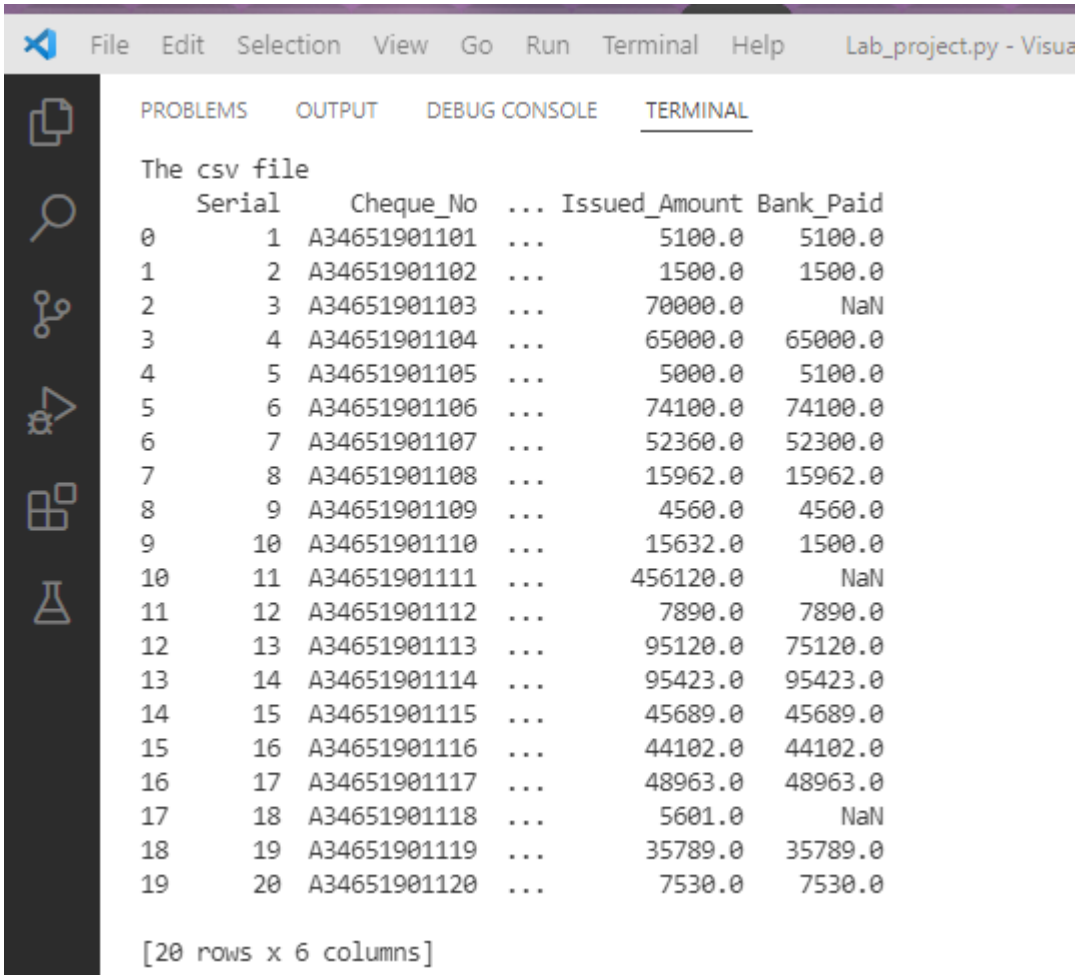


Figure 2: Converting the pdf file to csv file

File Edit Selection View Go Run Terminal Help Lab_project.py - Visual Stud

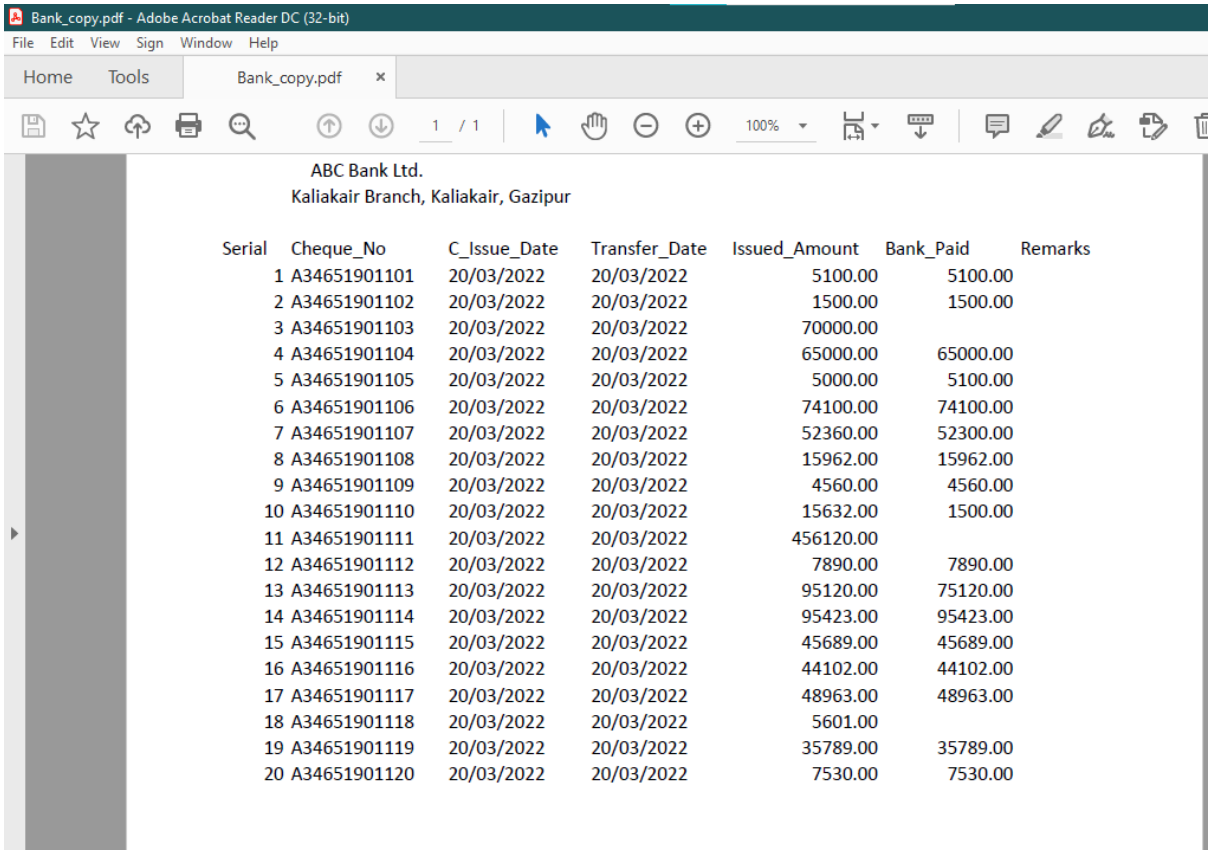
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
The csv file with remarks
    Serial     Cheque_No  ... Bank_Paid           Remarks
0        1  A34651901101  ...    5100.0                Ok
1        2  A34651901102  ...    1500.0                Ok
2        3  A34651901103  ...       NaN          No Value
3        4  A34651901104  ...   65000.0                Ok
4        5  A34651901105  ...    5100.0  To be reconciled
5        6  A34651901106  ...   74100.0                Ok
6        7  A34651901107  ...   52300.0  To be reconciled
7        8  A34651901108  ...   15962.0                Ok
8        9  A34651901109  ...    4560.0                Ok
9       10  A34651901110  ...    1500.0  To be reconciled
10      11  A34651901111  ...       NaN          No Value
11      12  A34651901112  ...    7890.0                Ok
12      13  A34651901113  ...   75120.0  To be reconciled
13      14  A34651901114  ...   95423.0                Ok
14      15  A34651901115  ...   45689.0                Ok
15      16  A34651901116  ...   44102.0                Ok
16      17  A34651901117  ...   48963.0                Ok
17      18  A34651901118  ...       NaN          No Value
18      19  A34651901119  ...   35789.0                Ok
19      20  A34651901120  ...    7530.0                Ok

[20 rows x 7 columns]
Total Issued Amount by company(in Tk.):  1151441.0
Total Paid Amount by bank(in Tk.):  585628.0
Needs to be reconciled
A34651901103
A34651901105
A34651901107
A34651901110
A34651901111
A34651901113
A34651901118
Bank Reconciled Amount(in Tk):  451608.0
```

Figure 3: Calculating and deciding remarks

Testing and deployment



Bank_copy.pdf - Adobe Acrobat Reader DC (32-bit)

ABC Bank Ltd.
Kaliakair Branch, Kaliakair, Gazipur

Serial	Cheque_No	C_Issue_Date	Transfer_Date	Issued_Amount	Bank_Paid	Remarks
1	A34651901101	20/03/2022	20/03/2022	5100.00	5100.00	
2	A34651901102	20/03/2022	20/03/2022	1500.00	1500.00	
3	A34651901103	20/03/2022	20/03/2022	70000.00		
4	A34651901104	20/03/2022	20/03/2022	65000.00	65000.00	
5	A34651901105	20/03/2022	20/03/2022	5000.00	5100.00	
6	A34651901106	20/03/2022	20/03/2022	74100.00	74100.00	
7	A34651901107	20/03/2022	20/03/2022	52360.00	52300.00	
8	A34651901108	20/03/2022	20/03/2022	15962.00	15962.00	
9	A34651901109	20/03/2022	20/03/2022	4560.00	4560.00	
10	A34651901110	20/03/2022	20/03/2022	15632.00	1500.00	
11	A34651901111	20/03/2022	20/03/2022	456120.00		
12	A34651901112	20/03/2022	20/03/2022	7890.00	7890.00	
13	A34651901113	20/03/2022	20/03/2022	95120.00	75120.00	
14	A34651901114	20/03/2022	20/03/2022	95423.00	95423.00	
15	A34651901115	20/03/2022	20/03/2022	45689.00	45689.00	
16	A34651901116	20/03/2022	20/03/2022	44102.00	44102.00	
17	A34651901117	20/03/2022	20/03/2022	48963.00	48963.00	
18	A34651901118	20/03/2022	20/03/2022	5601.00		
19	A34651901119	20/03/2022	20/03/2022	35789.00	35789.00	
20	A34651901120	20/03/2022	20/03/2022	7530.00	7530.00	

Figure 4: The given pdf file

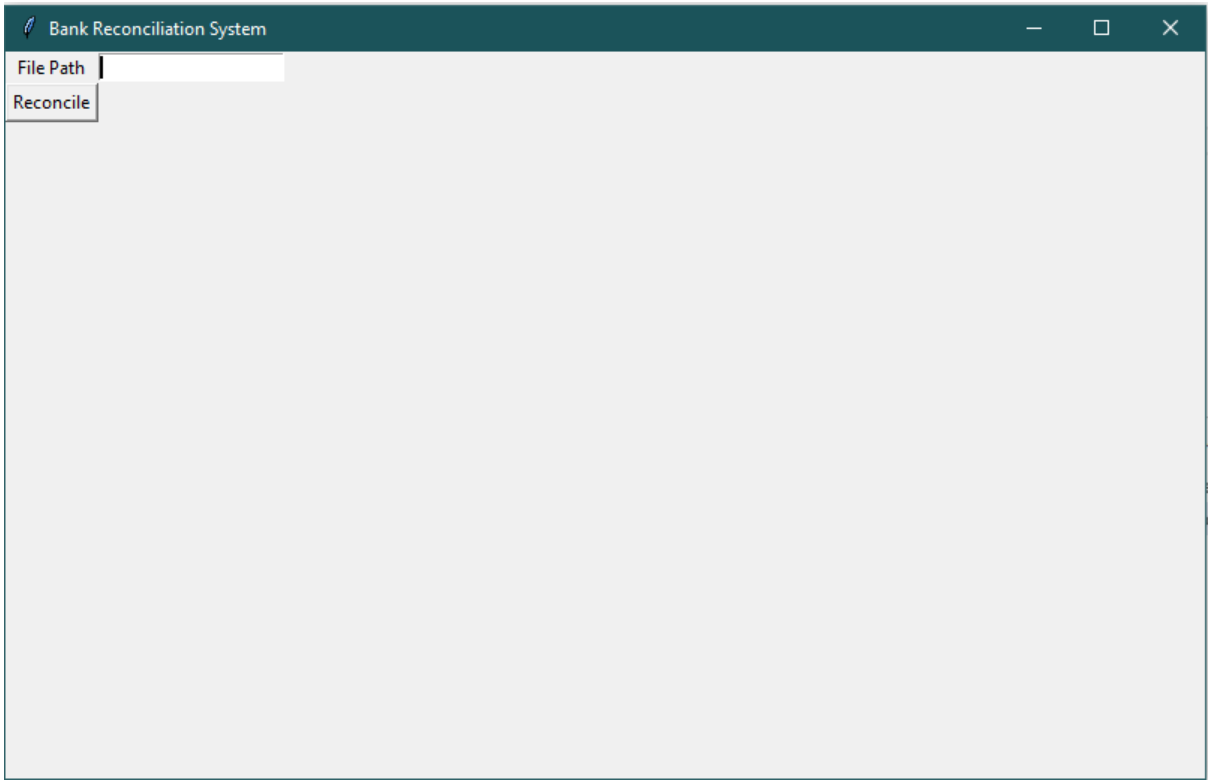


Figure 5: UI to take user input

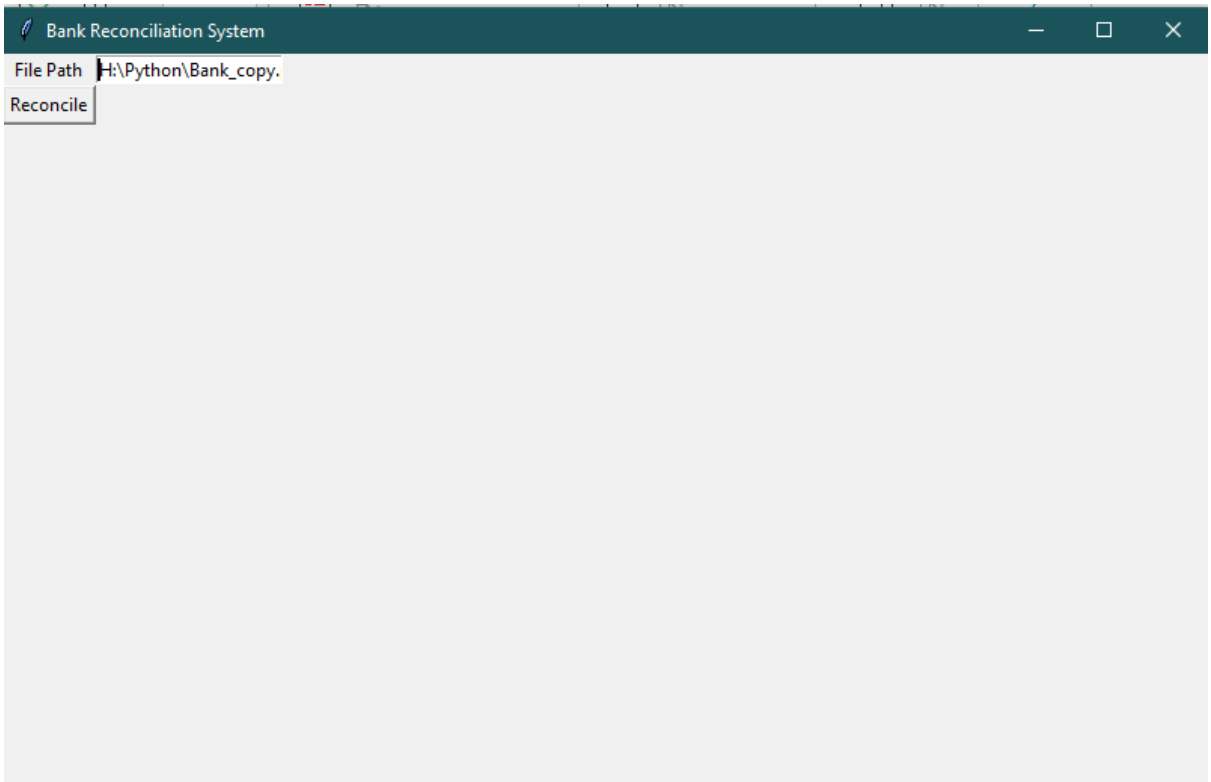


Figure 6: User entering the pdf file path

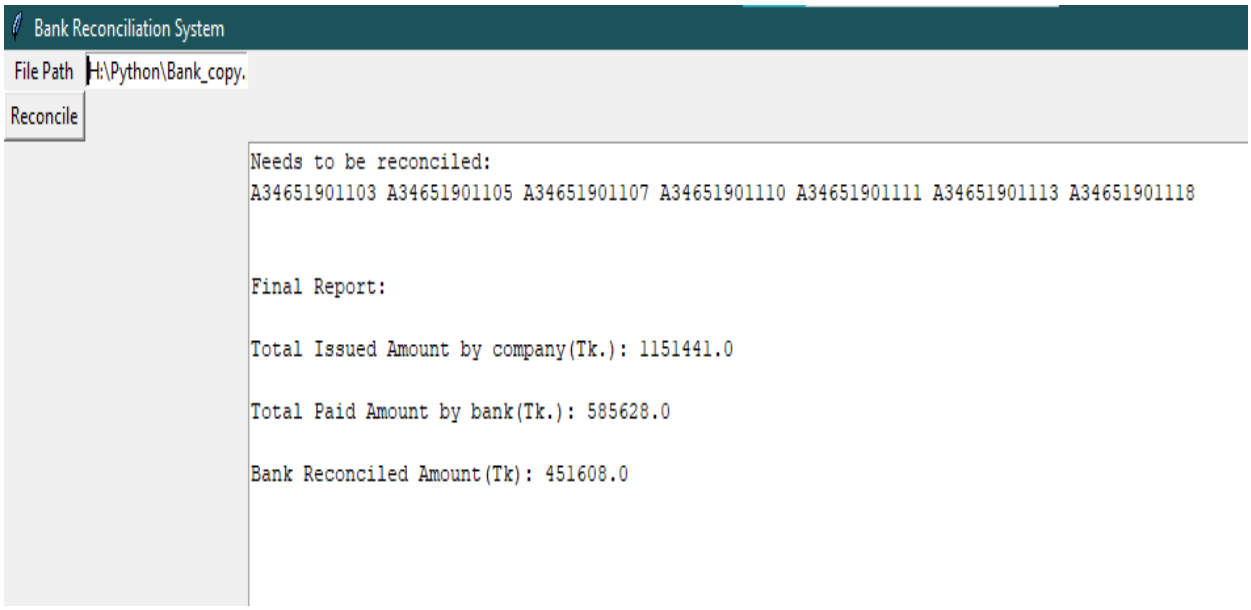
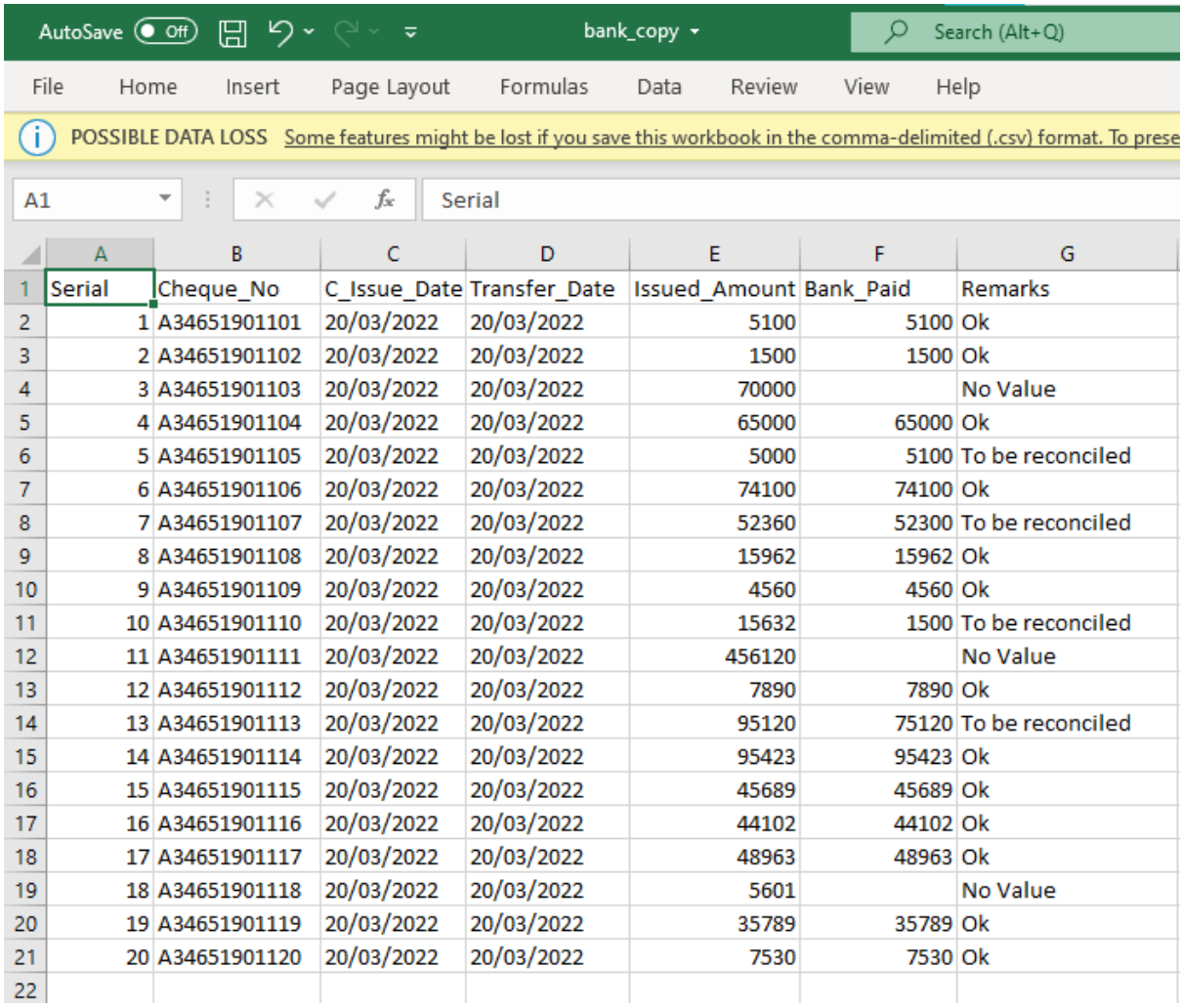


Figure 7: The output shown to the user using GUI



	A	B	C	D	E	F	G
1	Serial	Cheque_No	C_Issue_Date	Transfer_Date	Issued_Amount	Bank_Paid	Remarks
2	1	A34651901101	20/03/2022	20/03/2022	5100	5100	Ok
3	2	A34651901102	20/03/2022	20/03/2022	1500	1500	Ok
4	3	A34651901103	20/03/2022	20/03/2022	70000		No Value
5	4	A34651901104	20/03/2022	20/03/2022	65000	65000	Ok
6	5	A34651901105	20/03/2022	20/03/2022	5000	5100	To be reconciled
7	6	A34651901106	20/03/2022	20/03/2022	74100	74100	Ok
8	7	A34651901107	20/03/2022	20/03/2022	52360	52300	To be reconciled
9	8	A34651901108	20/03/2022	20/03/2022	15962	15962	Ok
10	9	A34651901109	20/03/2022	20/03/2022	4560	4560	Ok
11	10	A34651901110	20/03/2022	20/03/2022	15632	1500	To be reconciled
12	11	A34651901111	20/03/2022	20/03/2022	456120		No Value
13	12	A34651901112	20/03/2022	20/03/2022	7890	7890	Ok
14	13	A34651901113	20/03/2022	20/03/2022	95120	75120	To be reconciled
15	14	A34651901114	20/03/2022	20/03/2022	95423	95423	Ok
16	15	A34651901115	20/03/2022	20/03/2022	45689	45689	Ok
17	16	A34651901116	20/03/2022	20/03/2022	44102	44102	Ok
18	17	A34651901117	20/03/2022	20/03/2022	48963	48963	Ok
19	18	A34651901118	20/03/2022	20/03/2022	5601		No Value
20	19	A34651901119	20/03/2022	20/03/2022	35789	35789	Ok
21	20	A34651901120	20/03/2022	20/03/2022	7530	7530	Ok
22							

Figure 8: The csv file with remarks

Future enhancements

Thus, creating and applying this project we will be able to fulfil the following goals-

- Accurate annual accounts must be maintained by all organizations.
- Avoid late payments and penalties from the bank.
- Maintain good relationships with suppliers.

As this project will be applied or used in organizational purpose and for individual purpose it will be convenient. It will take less time and thus it will be very helpful for reconciling a good number of accounts. It will take less time than average human being and it will be able to calculate more accurately than an average human

being. So, this project will be stable and heavily used. This will play a good part in automation if applied properly.

References:

[1] https://docs.python.org/3/library/csv.html
[2] https://bdnews24.com/business/2015/06/30/counterfeiting-micr-cheques-increasing
[3] https://www.researchgate.net/figure/General-format-of-Bank-checks-used-in-Bangladesh_fig15_271823793
[4] https://m.facebook.com/bankingcareerbd/photos/what-is-micr-cheques-features-specifications-and-advantagesmicr-cheque-the-short/1642642672524158/
[5] https://betterprogramming.pub/simple-hacks-to-automate-python-code-beautification-5ad934cf5a29
[6] https://www.w3schools.com/python/pandas/default.asp
[7] https://www.w3schools.com/python/numpy/default.asp
[8] https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.html
[9] https://www.tutorialspoint.com/how-to-create-a-simple-screen-using-tkinter
[10]https://realpython.com/python-gui-tkinter/
[11] https://realpython.com/python-gui-tkinter/#getting-user-input-with-entry-widgets
[12] https://www.tutorialsteacher.com/python/create-gui-using-tkinter-python
[13] https://www.tutorialspoint.com/update-tkinter-label-from-variable
[14] https://www.geeksforgeeks.org/python-setting-and-retrieving-values-of-tkinter-variable/
[15] https://www.geeksforgeeks.org/python-tkinter-text-widget/
[16] https://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.DataFrame.insert.html
[17] https://pandas.pydata.org/pandas-docs/stable/user_guide/style.html
[18] https://stackabuse.com/reading-and-writing-csv-files-in-python-with-pandas/
[19] https://www.geeksforgeeks.org/working-csv-files-python/
[20] https://pandas.pydata.org/docs/reference/api/pandas.read_csv.html
[21] https://www.analyticsvidhya.com/blog/2021/08/python-tutorial-working-with-csv-file-for-data-science/
[22] https://towardsdatascience.com/7-best-ui-graphics-tools-for-python-developers-with-starter-codes-2e46c248b47c

Appendix:

Source Code:

```
"""Python Script to automate bank reconciliation"""
# importing necessary modules
import tkinter as tk

import numpy as np
import pandas as pd
import tabula

# creating entry widget to receive info from user
window = tk.Tk()
window.geometry("800x800")
window.title("Bank Reconciliation System")


def getvals():
    """Checking the rows of the file and remarking them"""
    text_area = tk.Text(master=window, height=400, width=300)
    text_area.grid(row=4, column=2)
    print("Submitting form")
    table_file = e1.get()
    output_csv = r"H:\Python\bank_copy.csv"
    df = tabula.read_pdf(table_file, pages="all",
pandas_options={"header": None})[0]
    print("The pdf file")
    print(df)
    # converting PDF into CSV
    tabula.convert_into(table_file, output_csv, output_format="csv",
pages="all")
    DF = pd.read_csv(
        output_csv,
        names=[
            "Serial",
            "Cheque_No",
            "C_Issue_Date",
            "Transfer_Date",
            "Issued_Amount",
            "Bank_Paid",
        ],
    )
    print("The csv file")
    print(DF)
    # remarking the lines
    conditions = [
        DF["Issued_Amount"] == DF["Bank_Paid"],
        DF["Issued_Amount"] >= DF["Bank_Paid"],
        DF["Issued_Amount"] <= DF["Bank_Paid"],
```

```
    ]
    choices = ["Ok", "To be reconciled", "To be reconciled"]
    # DF['Remarks'] = np.select(conditions, choices, default = 'No
Value')
    DF.insert(6, "Remarks", value=np.select(conditions, choices,
default="No Value"))
    print("The csv file with remarks")
    print(DF)
    # calculating the totals and mismatched account
    total_issue = 0.0
    total_bank = 0.0
    reconcile_amount = []
    total_issue = DF["Issued_Amount"].sum()
    print("Total Issued Amount by company(in Tk.): ", total_issue)
    total_bank = DF["Bank_Paid"].sum()
    print("Total Paid Amount by bank(in Tk.): ", total_bank)
    print("Needs to be reconciled ")
    text_area.insert(tk.END, "Needs to be reconciled:\n")
    recon_acc = []
    for i, d in DF.iterrows():
        if pd.isna(d["Bank_Paid"]) or (d["Issued_Amount"] !=
d["Bank_Paid"]):
            recon_acc.append(d["Cheque_No"])
            print(d["Cheque_No"], "\t")
        if d["Remarks"] == "Ok":
            reconcile_amount.append(d["Bank_Paid"])
            ##DF.style.applymap('green', subset=['Remarks'])
        ##else: DF.style.applymap('red', subset=['Remarks'])
    text_area.insert(tk.END, recon_acc)
    print("Bank Reconciled Amount(in Tk): ", sum(reconcile_amount))
    # DF.to_csv(output_csv, index=None)
    e2 = "\n\n\nFinal Report:\n\nTotal Issued Amount by
company(Tk.): {issue}\n\nTotal Paid Amount by bank(Tk.):
{bank}\n\nBank Reconciled Amount(Tk): {recon}".format(
        issue=total_issue, bank=total_bank,
recon=sum(reconcile_amount)
    )
    text_area.insert(tk.END, e2)
    # converting the file
    DF.to_csv(output_csv, index=None)


# showing the output in text widget
e1 = tk.StringVar()
tk.Label(window, text="File Path").grid(row=0)
e1 = tk.Entry(window)
e1.grid(row=0, column=1)
tk.Button(text="Reconcile", command=getvals).grid(row=3)
window.mainloop()
```

--End of the Report--