Banking System with Data Tracking

- A Project by : Aditya Ved (16010121208)
  Abhishek Gupta (16010121226)
  Rugved Palodkar (16010121211)

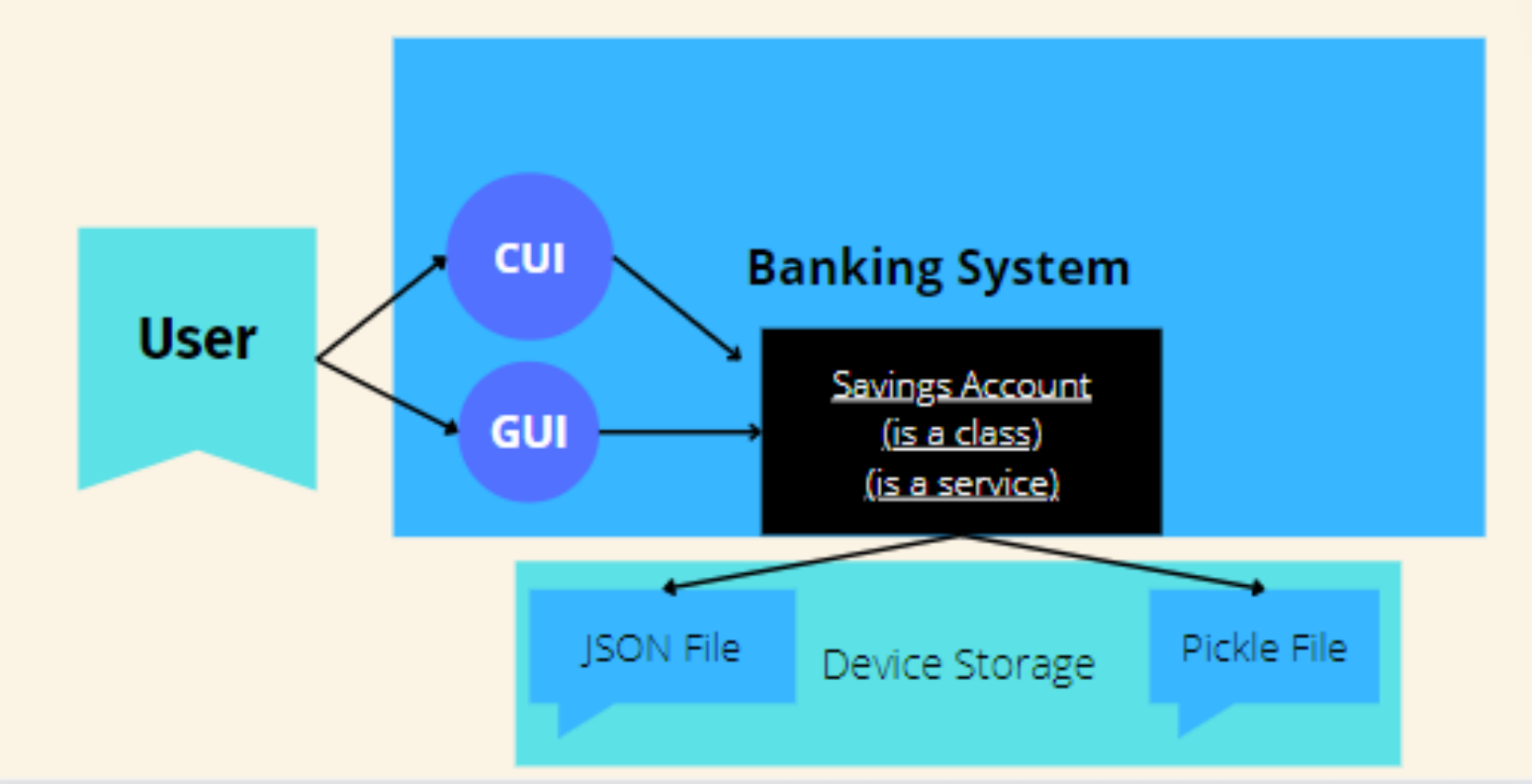
SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST

Problem Statement

To create a software system that provides access to the user to create an account, debit or credit money into another account and also keeping track of every transaction done by the user.

System Architecture



In this project we have created a banking system which provides a service called the savings account.

This service can be called by the user using 2 ways :

- CUI
- GUI

This service provides savings account functionality and stores the account information on the disk for permanent storage. The following python modules are used: Pickle JSON Tkinter

Design Principles : We have made a component and service based architecture so that the team can work independently.

Modules and Libraries used:

1)json: JSON is a syntax for storing and exchanging data. JSON helped us in solving the problem of: how to store a list (a python data structure) in a file and then also retrieve from it and make sense out of it which is not possible with the normal string read and string write operations supported in python. The main purpose of storing the list in a file is to confirm the existence of the username at multiple points in the program and for the program to remember the username signed in even when the program is run once again.

2)pickle: Python pickle module is used for serializing and de-serializing python object structures. The process to converts any kind of python objects (list, dict, etc.) into byte streams (0s and 1s)The pickle module in python allows us to store an entire object in a file in a binary form so that even if someone opens the file, he or she cannot make sense out of the data. Thus, the attributes of the Account object: like password, name, balance etc. are stored in the binary format which is a more secure way. Purpose of storing an object in a file: the program will remember the sign in that were made.

3)datetime: Python has a module named datetime to work with dates and times. We have used it on our program to record the date and time of the every transaction taking place

4)tkinter: Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications.

Features

1)**Pure OOP design**: Pure OOP design has helped us to give a structured approach to our program and the modules that are built around for OOP implementation (like pickle module) has helped us to keep a track of the data of the account object. We understand the power of OOP and how we can consider each account of the bank as an object. OOP has helped us to solve the problem using a real-world model which connects with us more intuitively.

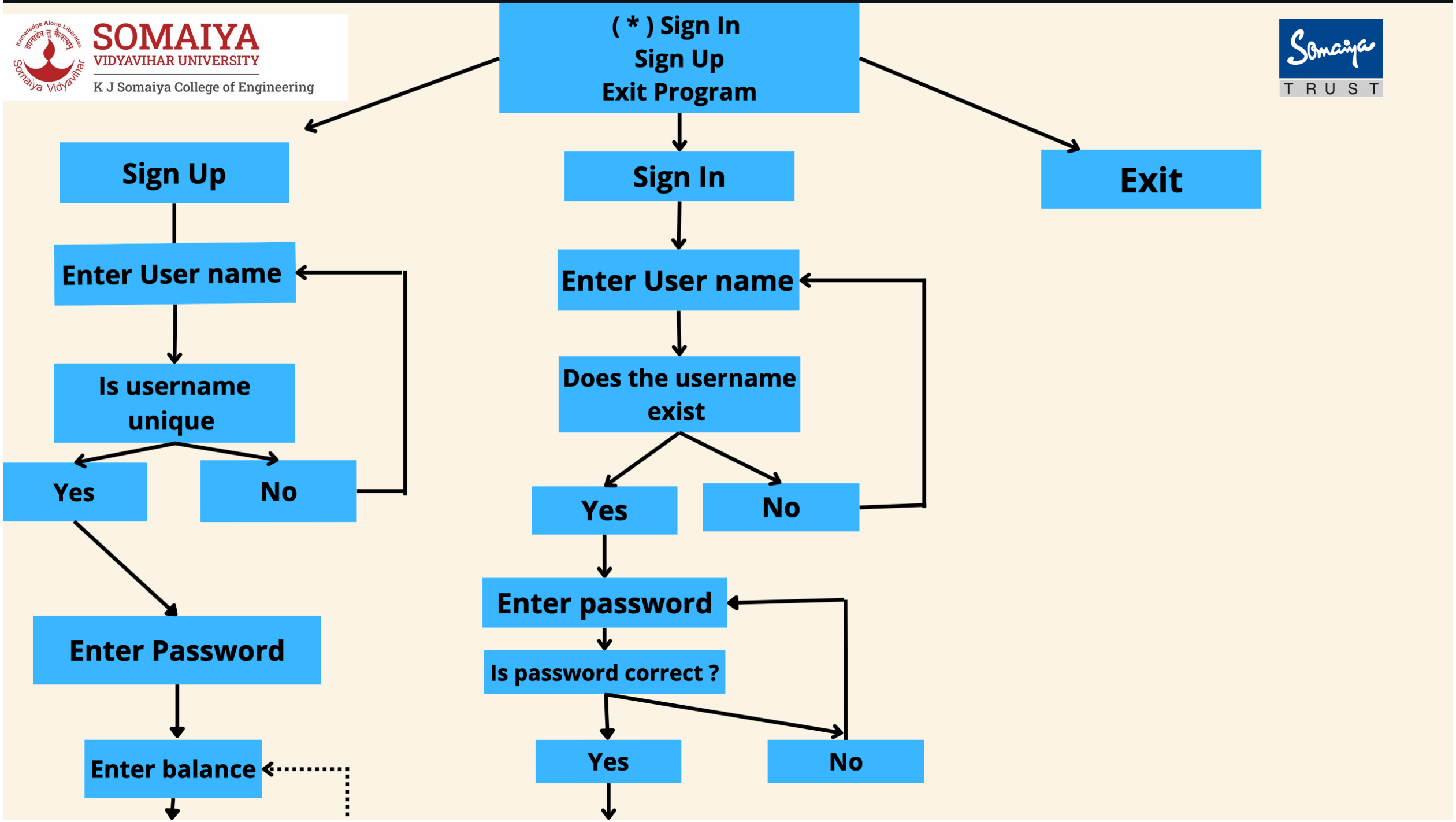
2)**File Handling** (both Binary and Text files): File Handling is the storing of data in a file using a program. We are using file handling in python to store data in the hard drive and make the project more real world.

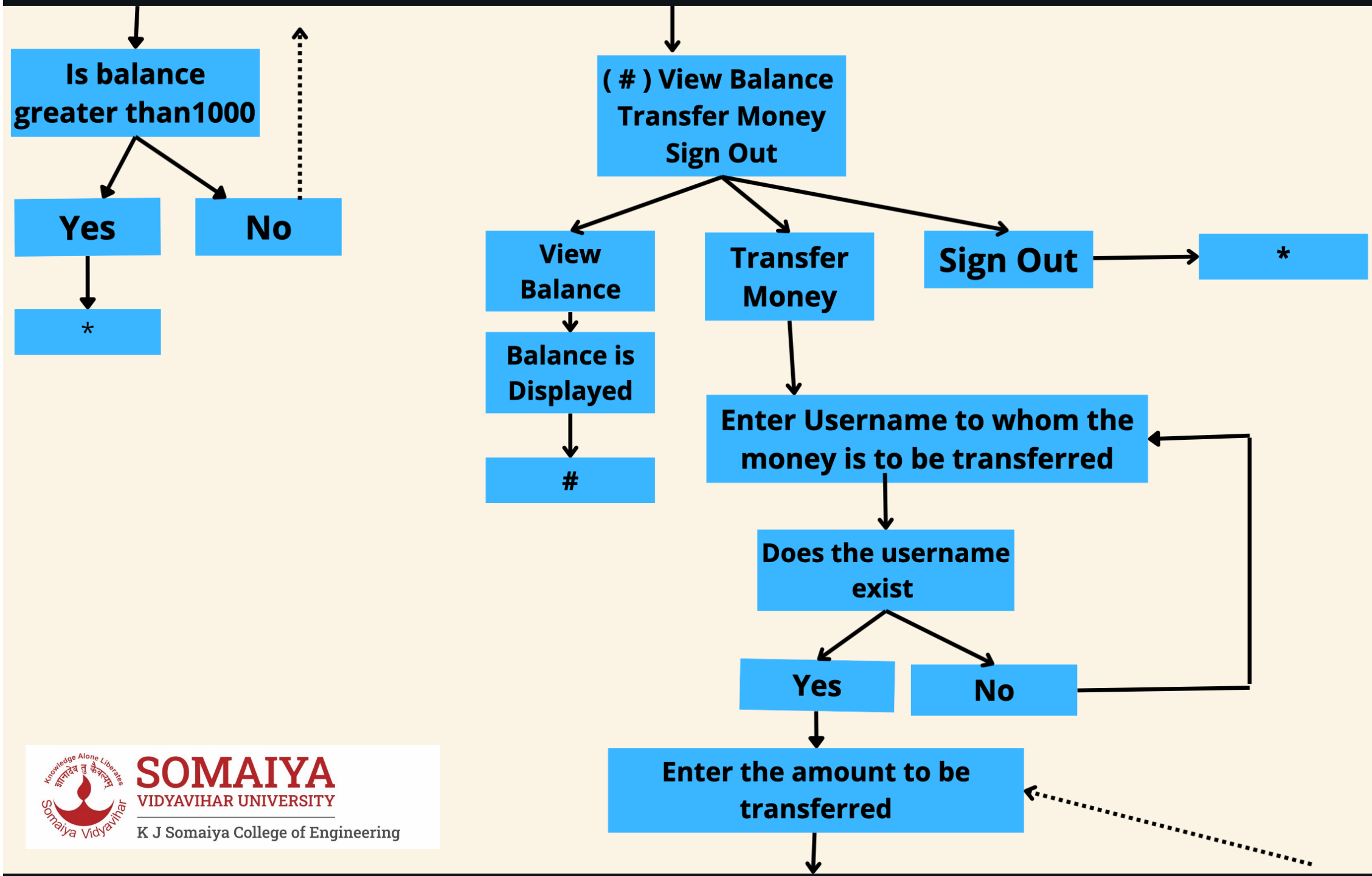
3)**GUI :** Everyone loves a user friendly and precise GUI. So, we also tried to make a simple GUI for the project and make it more real life and interesting.

4)**Data Tracking:** All the Transaction History is recorded in text file for future reference

5) **Improved Code Performance**

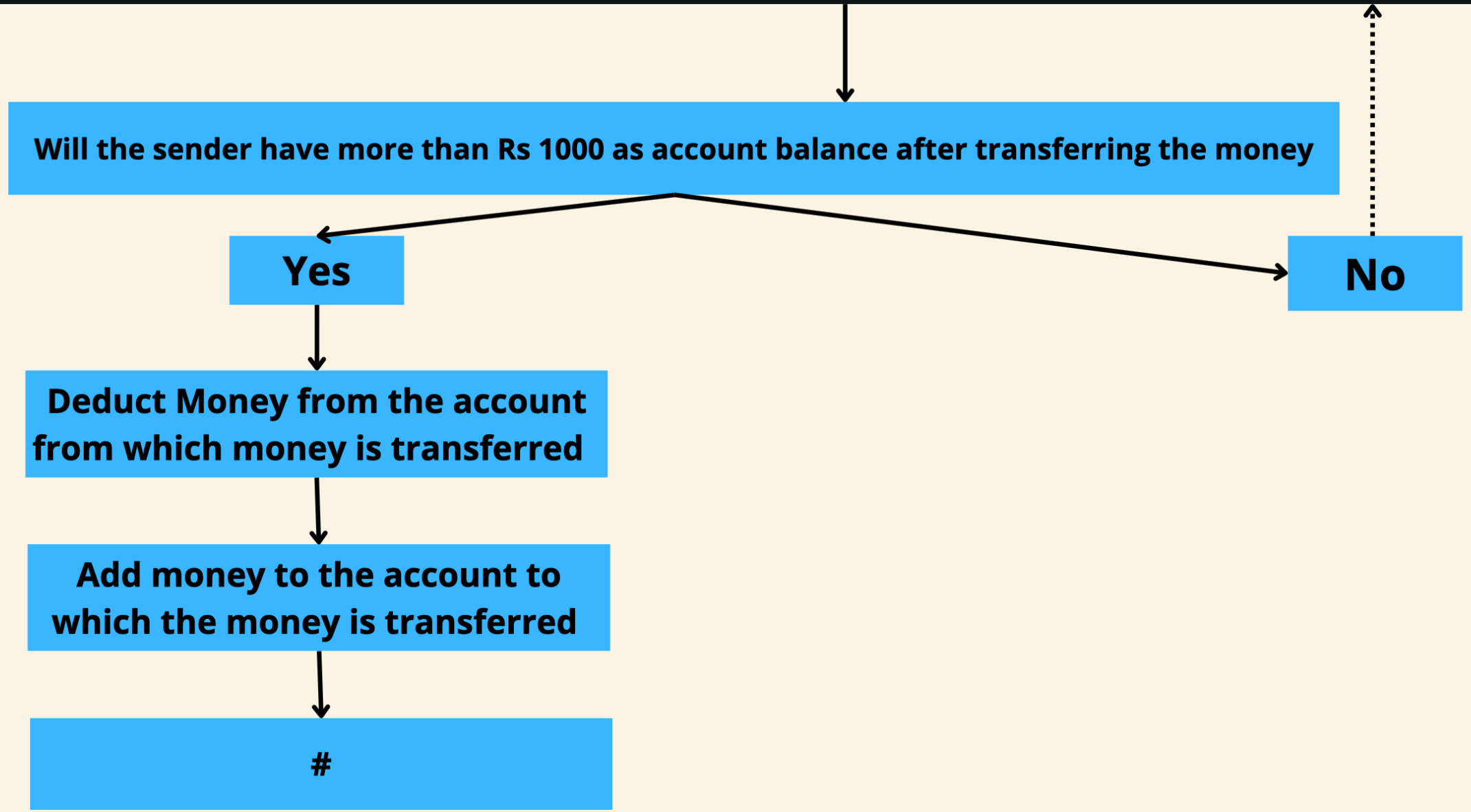
Result : Flowchart of User Story


(*) Sign In
Sign Up
Exit Program
Sign Up
Sign In
Exit
Enter User name
Is username unique
Yes
No
Enter Password
Enter balance
Enter User name
Does the username exist
Yes
No
Enter password
Is password correct ?
Yes
No



SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya College of Engineering

Somaiya TRUST


Will the sender have more than Rs 1000 as account balance after transferring the money
Yes
No
Deduct Money from the account from which money is transferred
Add money to the account to which the money is transferred
#

Conclusion

- The coding of the project was difficult with many errors arising. Many systems had to be written numerous ways before a final working solution was found.
- We made the use of GUI to make it more user friendly and real worldly while also making a separate CUI.
- All this process started from learning the modules like tkinter, json, pickle. We tried to take most of these modules and use them to make the code simpler and attractive.
- Using functions made it easier to determine where errors were occurring when debugging the code.
- It also kept the code more organized. After all these hardships we made a Banking system with a data tracking facility. Finally to conclude we would say that the project helped us enhance not only our coding skills but also our teamworking capabilities.

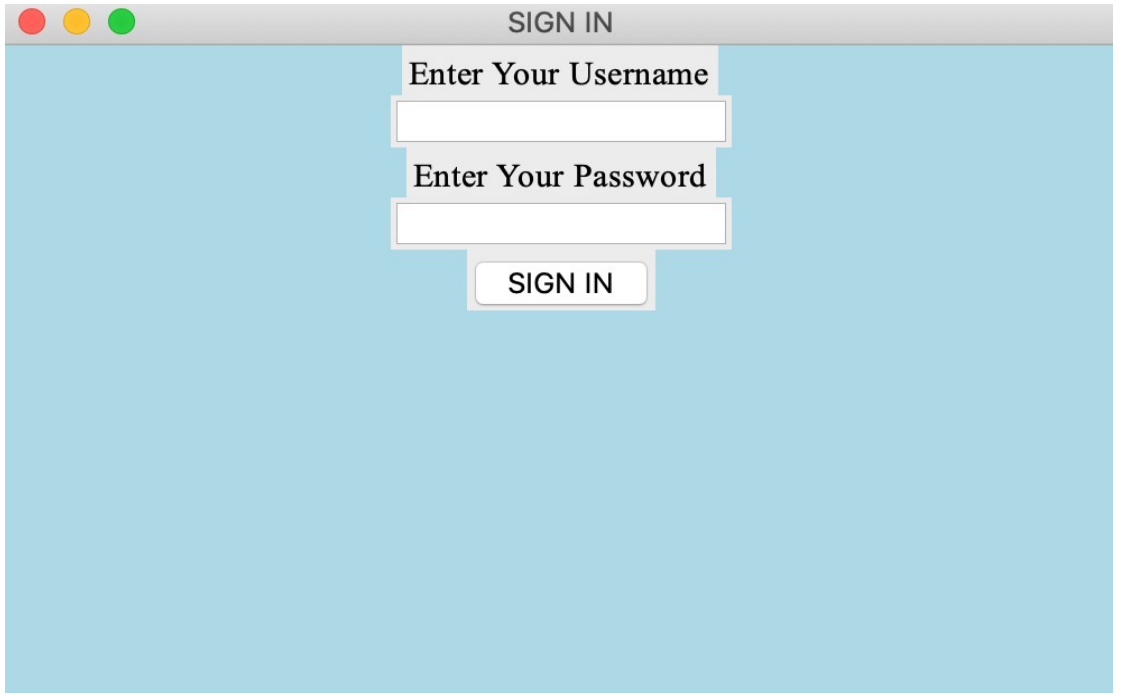
- Initially we created a cui design which worked great. We thought that we would then integrate our class with the gui design but in the process we found that it was not possible because the class itself used things like print statement which is irrelevant for gui .
- Thus we understood the flaw in the Account class, it could not act as a true service which is callable from both the gui and the cui.
- Thus another gui file was created to facilitate the entire class as well as the gui.

Learnings:

- We learnt how the needs of cui and gui are different and we also learnt how difficult it is to integrate the class as well the gui if we have used the features of cui already inside the class Thus we learnt that in complex projects : the gui file is different , the service file is different and the cui file is different.
- The component/module (here : class) which does the actual compute is like a service and both the gui and the cui can call the service on their own independently (still retaining the data) .
- For Eg a user can sign up using gui and then he can sign in using cui. This is how real world projects should look like and through this project we really got a feel of how to implement them using the correct model. It is not only the OOP model that matters but also how we de couple a true service from the mode using which it is called creates a big difference when it comes to integration of code made my multiple teams working on a single project

Output

These are some snippets from the output of the code



Thank you