ATM application

The purpose of ATM application

Nowadays, there are many ways to provide convenience to customers when doing bank transactions, such as Internet banking, phone banking, and credit card etc. Among them, ATM (Automatic Teller Machine) is one of the oldest ways for bank transactions.

This computer, which allows customers to conveniently manage their funds, provides balance checking, withdrawal and deposit, and remittance functions. Some countries offer services such as cash advance services and can even buy stamps.

When I started this project, this ATM device came to mind when I thought of terminals while doing various brainstorming. Assuming that this brand new ATM machine with a new transfer feature is installed at a virtual bank called Cooper Bank. ATM features of withdrawal, deposit, transfer, and balance check are implemented at this application.

I hope this application will be a good opportunity for young children and the elderly who are not familiar with ATM operation to learn how to operate it.

The link of Github

ATM application

Presentation Clip on YouTube

https://youtu.be/NhjopFiIVjA

Style Guide

The style guide I adhered to when creating this application is python's pep8 style. Source layouts such as indentation and line length refer to the coding convention presented in pep8. Also, the naming of variables, functions, and classes uses PascalCase, snake_case, and camelCase, which are widely used in the Python community.

Also, in case I missed this coding convention when coding, I received the guide of pylint, one of the vscode extensions, and the help of autopep8 in the Python package. This tool is very useful because it automatically enforce the coding conversion of pep8 and prevents my common mistakes in advance. By doing this, I can avoid inconsistency that mess my source code and increase your coding productivity by increasing readability.

Reference: https://peps.python.org/pep-0008/

Features

The ATM application has a total of 6 features that is features of real ATM such as verifying Identification, withdraw, deposit, transfer, balance check, and cancel transactions. Also, I obtained educator's approval.

https://discord.com/channels/738633600701825117/1019579052765491210

Verifying Identification

All transactions will begin with this transaction. If the client has his/her own card and right PIN this application will be shown main menu screen. the client has three chances to enter your PIN number correctly.

Withdrawal

This feature is literally able to withdraw money from your account if you have enough money as much as you can take. To do this, This feature will check the balance first. If a client has enough money you will be given money. Also, you can get a receipt when you complete the transaction. The client makes sure the balance is deducted as much as withdraws.

Deposit

A client can deposit the cash into his/her account. When this transaction is completed, print the balance on the receipt and also check it on the screen. Likewise, The client can get receipt and confirm the balance on the screen. The client can check the added amount of the account.

Transfer

A client can send friends and family your money immediately. Firstly, you enter bank details of a person who you want to transfer money. Before sending your money, the ATM have to check your balance if you have enough money to transfer. If you have, the trasfer transaction will be started soon. The client can get a receipt written your sending amount and also he/she is able to see your balance deducted amount of transfer on the screen.

Check balance

The client can check the balance by choosing this transaction. The client can see how much money is left in his/her account.

Cancel Transaction

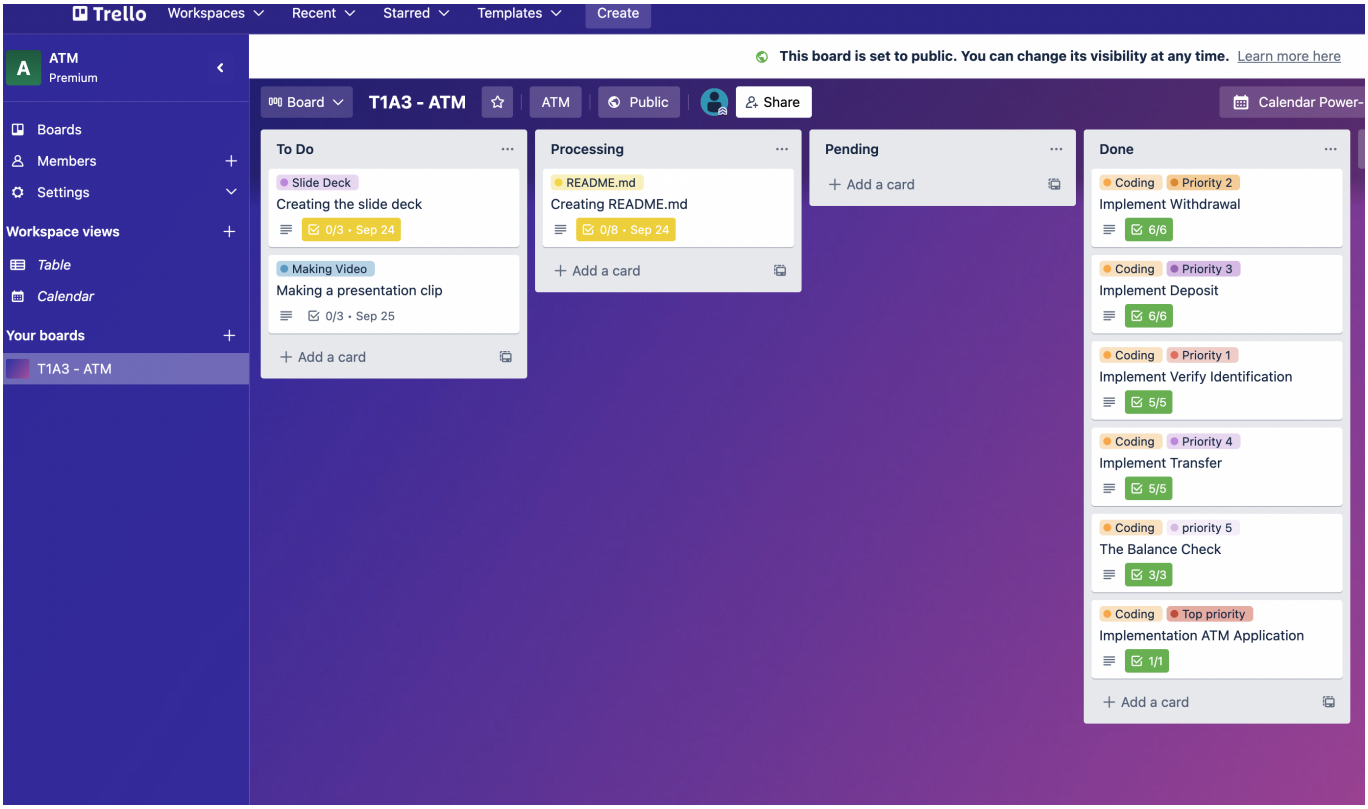
The transaction will be terminated when choosing this transaction.

Implementation Plan

The plan of project is an integral part of proceeding project. Efficient project management can be achieved by keeping the deadlines in the plan and proceeding with the project according to the processing priorities. I am using a project management tool called Trello for efficient project management, and I am dividing the project into **To Do**, **Processing**, **Pending**, and **Done** based on project processing and working on it according to work priority and deadline. You can track my ATM project below link.

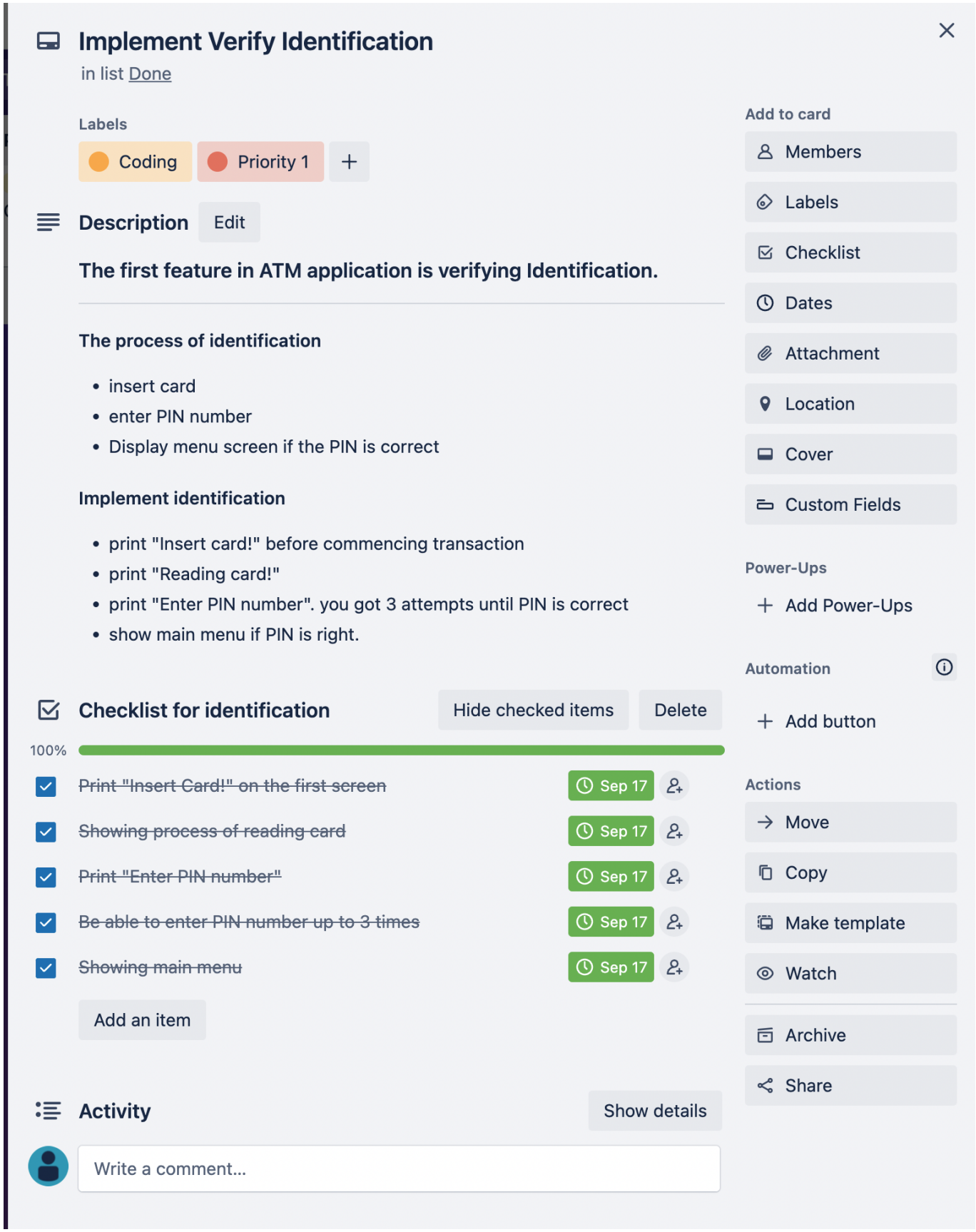
Trello

Implementation plan in Trello



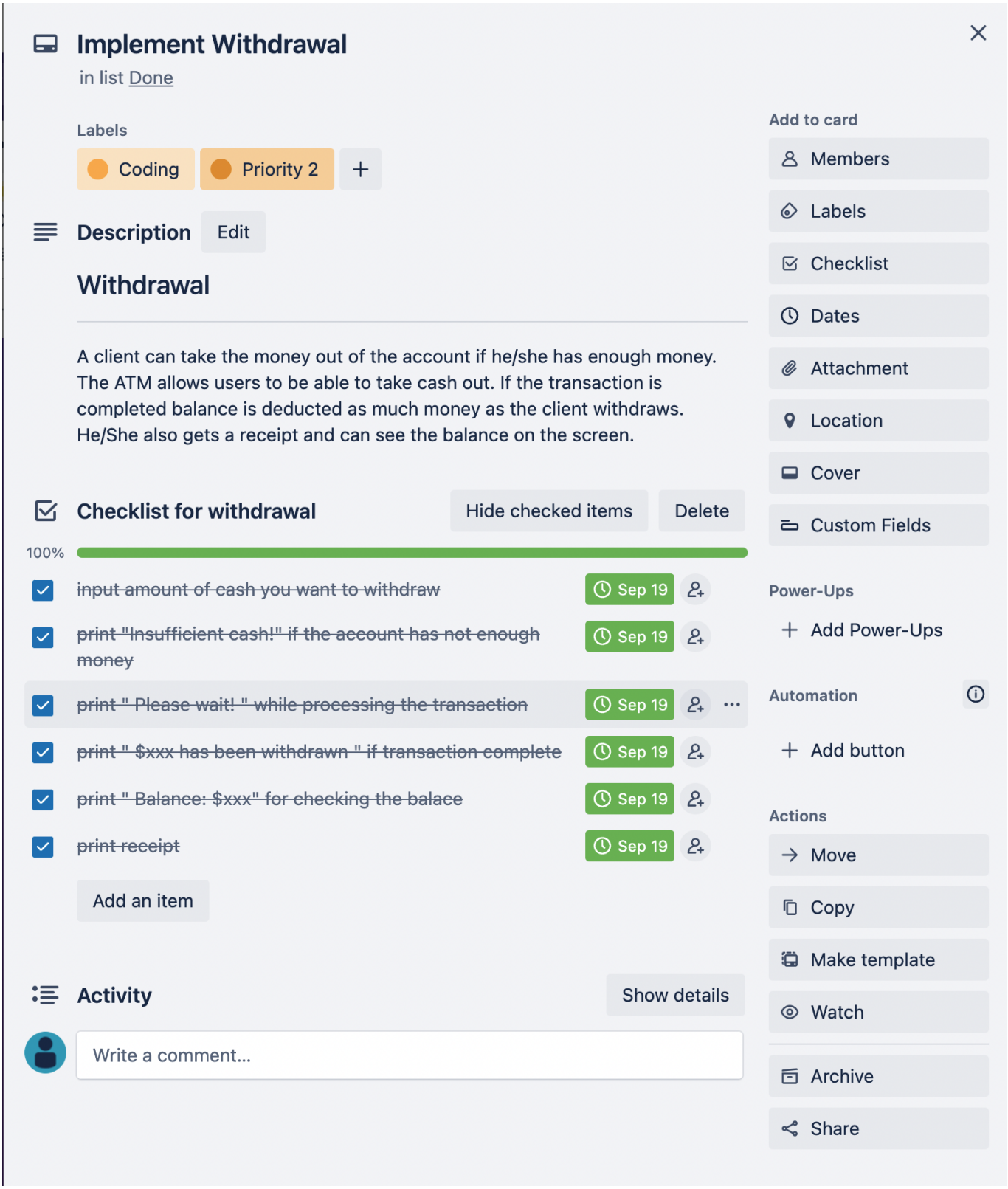
As mentioned above, I sperate the plan into `To Do`, `Processing`, `Pending`, and `Done`. This capture is made 24th September 2022.

Verify Identification feature



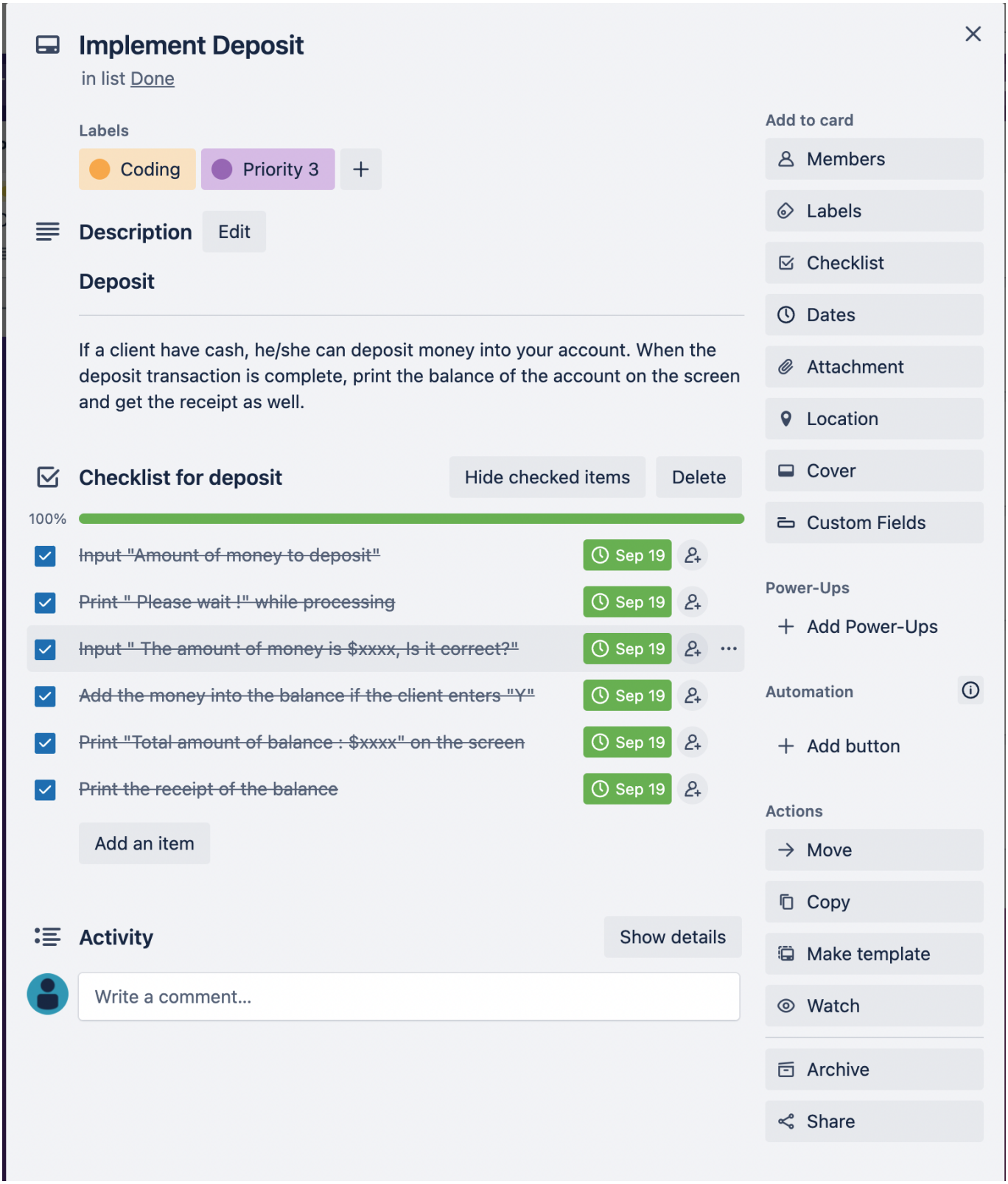
You can checklists, duedate outline of this feature. I gave this `priority 1` in order to make it first.

Withdrawal feature



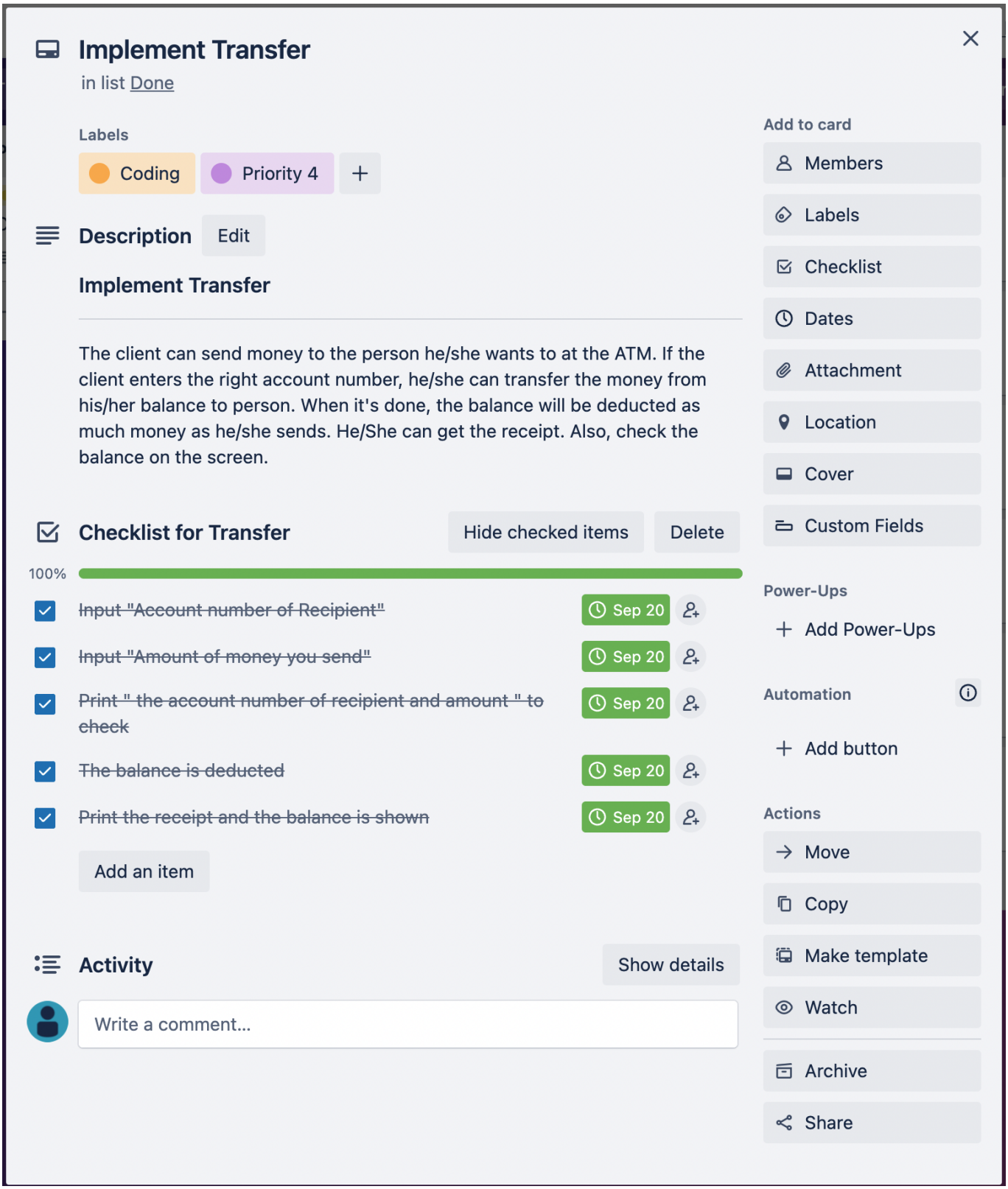
As above, I created checklists, duedate to make it easy manage the Implementation plan. This is `priority2`.

Deposit feature



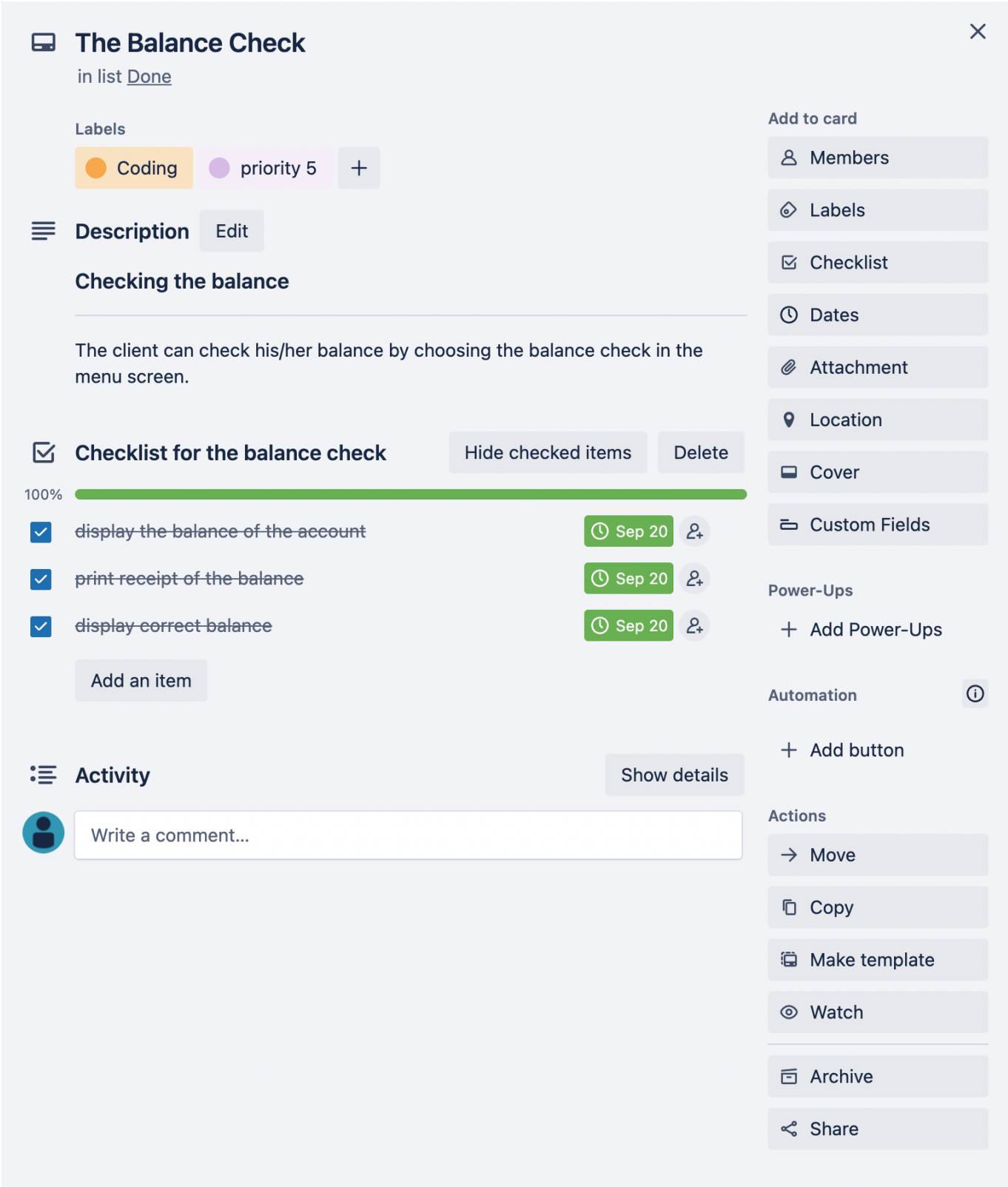
The priority is `priority3`. I already had it done.

Transfer feature



The priority is `priority4`. The duedate was 20/09/2022

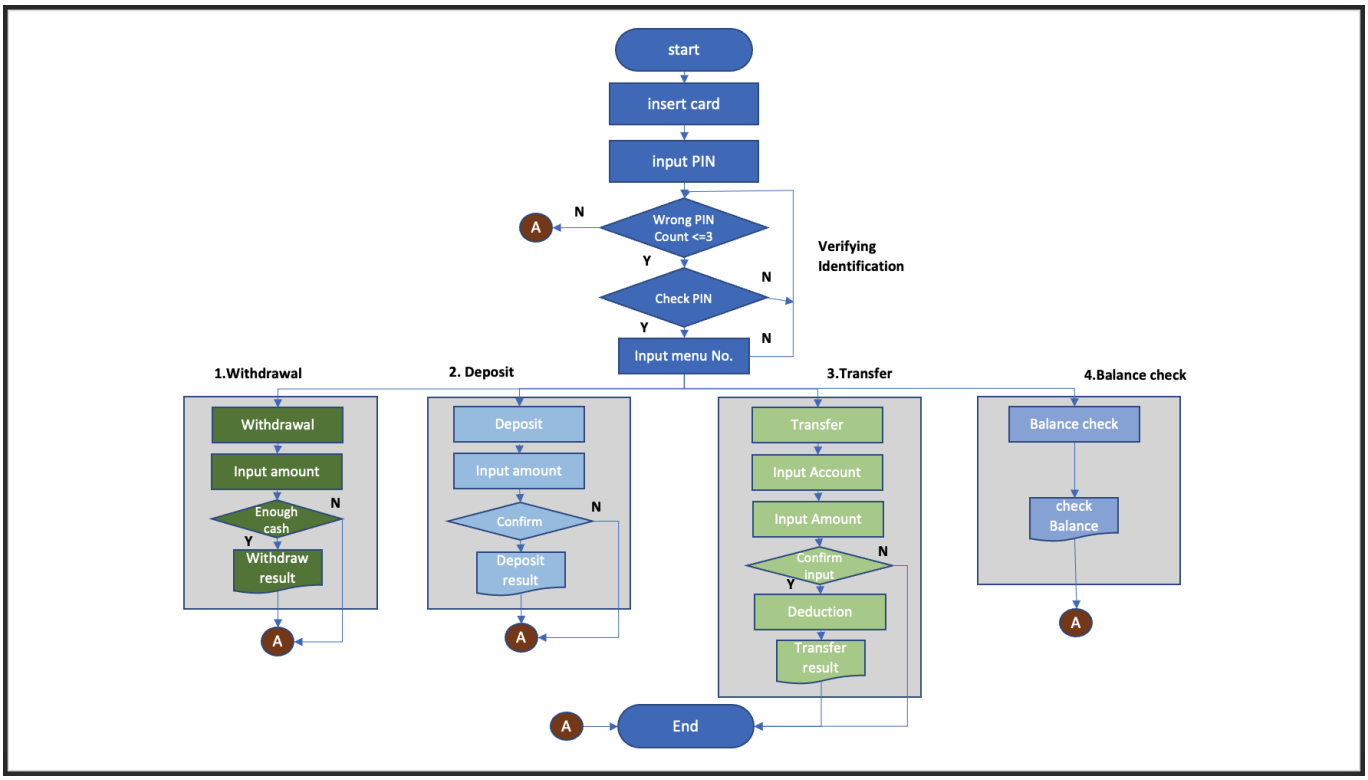
Balance Check feature



This feature is the method in Transaction class that used every feature. so the functionality of this transaction had already been verified by other features. So I put just 2 checklists instead 5.

Flow Chart

We often use the flow chart to understand the application structure at a glance. It helps to understand the overall flow of the application by identifying where to use conditional statements and loop statements in the application.



Testing

The ATM application I made has a total of six features. A unit test is conducted to verify that the function of each feature works accurately. So I made a total of six test suites by this feature.

outline of the testing procedure.

When I did the test, I conducted the test by mixing the **automatic tes**t and the **manual tests**. The automatic test targets are withdrawal, deposition, and transfer feature among features. For each feature, the test module is `test_transaction_withdraw`l, `test_transaction_deposit`, and `test_transaction_transfer`, which can be found in the code source. The test was conducted with pytest according to the test cases below, and the test results were also confirmed.

The target of the manual test is `verify identification`, `balance check`, and `cancel transaction`. It was tested whether the main function of each feature was properly implemented and whether the message was properly output when meet specific condition.

The test suits of each feature is as follows.

Feature	Test Type	Test Case	Expected Outcome	Actual Outcome
Verify Identification	Manual	1. When enter correct PIN	1. Display Main menu	Confirm display main menu
		2. When wrong PIN	2. Display "Wrong PIN "	Confirm "Wrong PIN"
		3. When more than 3 PIN wrong	3. Transaction is terminated	Confirm Transaction is terminated
Withdrawal	Automatic	**test_withdrawal_amount**		
		1. When enter '1' in transaction id, check input mes	1.Display "Type your withdrawal amount: "	Confirm as expected
		2. When enter amount greater than 0	2. Pass to next step	Confirm as expected
		3. When enter amount 0 or less	3. Value Error	Confirm as expected
		4. When enter string in amount	4. Value Error	Confirm as expected
		5. Check message according to Exception	5. Display "Wrong number"	Confirm as expected
		6. When enter amount greater than balance	6. Display " Insufficient funds"	Confirm as expected
		7. Check print receipt	7. Print withdrawal amount	Confirm as expected
		8. Check balance amount when displaying	8. Display balance deducted withdrawn amount	Confirm as expected
Deposit	Automatic	**test_transaction_diposit**		
		1. When enter '1' in transaction id, check input mes	1. Display "Type your deposit amount "	Confirm as expected
		2. When enter amount greater than 0	2. Pass to next step	Confirm as expected
		3. When enter amount 0 or less	3. Value Error	Confirm as expected
		4. When enter string in amount	4. Value Error	Confirm as expected
		5. Check message according to Exception	5. Display "Wrong number"	Confirm as expected
		6. Check print Receipt	6. Print deposit amount	Confirm as expected
		7. Check balance amount when displaying	7. Display balance added deposit amount	Confirm as expected
Transfer	Automatic	**test_transaction_trasfer**		
		1. When enter bank details less than 3 digit	1. Value Error	Confirm as expected
		2. When enter bank details greater than 3 digit	2. Keep going to next step	Confirm as expected
		3. When enter amount of transfer less than 0	3. Value Erro	Confirm as expected
		4. When enter amount of transfer greater than 0	4. Keep going to next step	Confirm as expected
		5. When enter transfer amount great than balance	5. Display insufficient funds	Confirm as expected
		6. When enter transfer amount less than balance	6. Keep going to next step	Confirm as expected
	Manual	7. When N if correct amount	7. Transaction's terminated to check amount	Confirm as expected
	Manual	8. When Y if correct amount	8. Keep going to next step	Confirm as expected
		9. Check if correct receipt	9. Display transferred amount	Confirm as expected
		10. Check balance subtracted transferred amount	10. Display balance deducted transferred amount	Confirm as expected
Balance	Manual	1. Display balance	1. Display balance	
Cancel Transaction	Manual	1. Transaction's terminated	1. Terminated	

The part to be tested for each feature was made of test cases, and it was tested whether each function was properly branched according to the conditional statement, the while loop deviated according to the condition, and the declared variable worked well according to the scope.

Install ATM Application

To use this application, you must follow the steps below in order.

install Python

If you do not have Ruby installed on your computer, please go Pythone.org page and then choose your Operating system on your computer. Then download Top on Stable Release. Next, follow the installation instructions.

Create the virtual environment

Move to a directory you will install this application. Excute `python –m venv [name of the virtual environment]` (for example python -m venv .venv)

Activate the virtual environment

Excute `$ source (venv)/bin/activate`, here `venv` is the directory where the virtual environment is.

Install packages

`pip install -r requirements.txt` , You can install packages

Excute bash script

Run `-/(bashfile).sh -auto`. then packages I installed when I made this application will be installed. (You can create this file name - for example xxxx.sh)

Please enjoy this ATM application

In my bash script, I made this process to excute automatically so you just type `./t13_atm.sh` on terminal.

Reference: https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/

bash scripting: https://edstem.org/au/courses/9040/lessons/24167/slides/171038

https://www.taniarascia.com/how-to-create-and-use-bash-scripts/

Required Dependencies in ATM Application

colorama

Produce colored terminal text and cursor positioning on Unix, Windows, and Macs. I used this to make the transaction title highlighted.

Reference: https://pypi.org/project/colorama/

datetime

Provide a DateTime date Type. I used it when getting trasaction date and time on the receipt.

Reference: https://pypi.org/project/DateTime/

prettytable

On terminal environmentm, the result of transaction and the receipt provides well-organised table. Thus, it gives the view a feeling of trust and neatness about the results.

Reference: https://pypi.org/project/prettytable/

pytest

Pytest is one of the most popular packages in Python. It makes testing easier so it can contribute to increase productivity.

Reference: https://pypi.org/project/pytest

---

system/hardware requirements

In fact, I'm not sure what sys requirements are needed for runing this ATM application. Instead, I suggeste Phyton System Requirements.

- CPU : intel Pentium 4 2.00GHz or higher
- Memory: 2GB RAM
- Operating system: Linux-Ubuntu 16.04 to 17.10 or Windows 7 to 10