# Introduction:

Shopping online is one of the demanding processes in this era, especially in this COVID-19 situation. People love to purchase things online. It saves time and money.   
Software or website is one of the important things of an online shopping system. Today, I am going to introduce an online shopping management software. By using this software, customers can see the products, purchase a product. And also admin can create, update, delete and include the new product in the list.

# Problem:

1. The physically shopping system takes more physical action. Sometimes, aged people face difficulties in the physical shopping system.
2. The physical shopping system needs more time. Sometimes it could take an entire day.
3. Bargaining with the buyer is one of the main problems of this shopping system.

# Solution:

1. This project was made for an online electric gadget shop. So, the electric gadget will be available in this project. But this could use another selling store.
2. Easy to use for everyone.
3. No need to bargaining.
4. Shop owners do not need any physical store.

# Facing problems during making this project:

1. The information about any user or new user or any kind of product information or order information was temporary. We could not store the information permanently.
2. Validating user information, who logged in in the system.
3. Distinguishing the user.
4. After closing the program, the order information is deleted. Because the data was not stored permanently.
5. Showing item table or order information was difficult. Aligning the information is almost impossible. Because each information has a different size.
6. Giving manually unique IDs for the product is difficult.
7. Giving choices accordingly to the user is difficult. For example, giving the delete product option to the admin instead of the customer is difficult.
8. The updateItems() or deleteItems() or writeItems() can use for one product at the same time.

# Solution of facing problem during making this project:

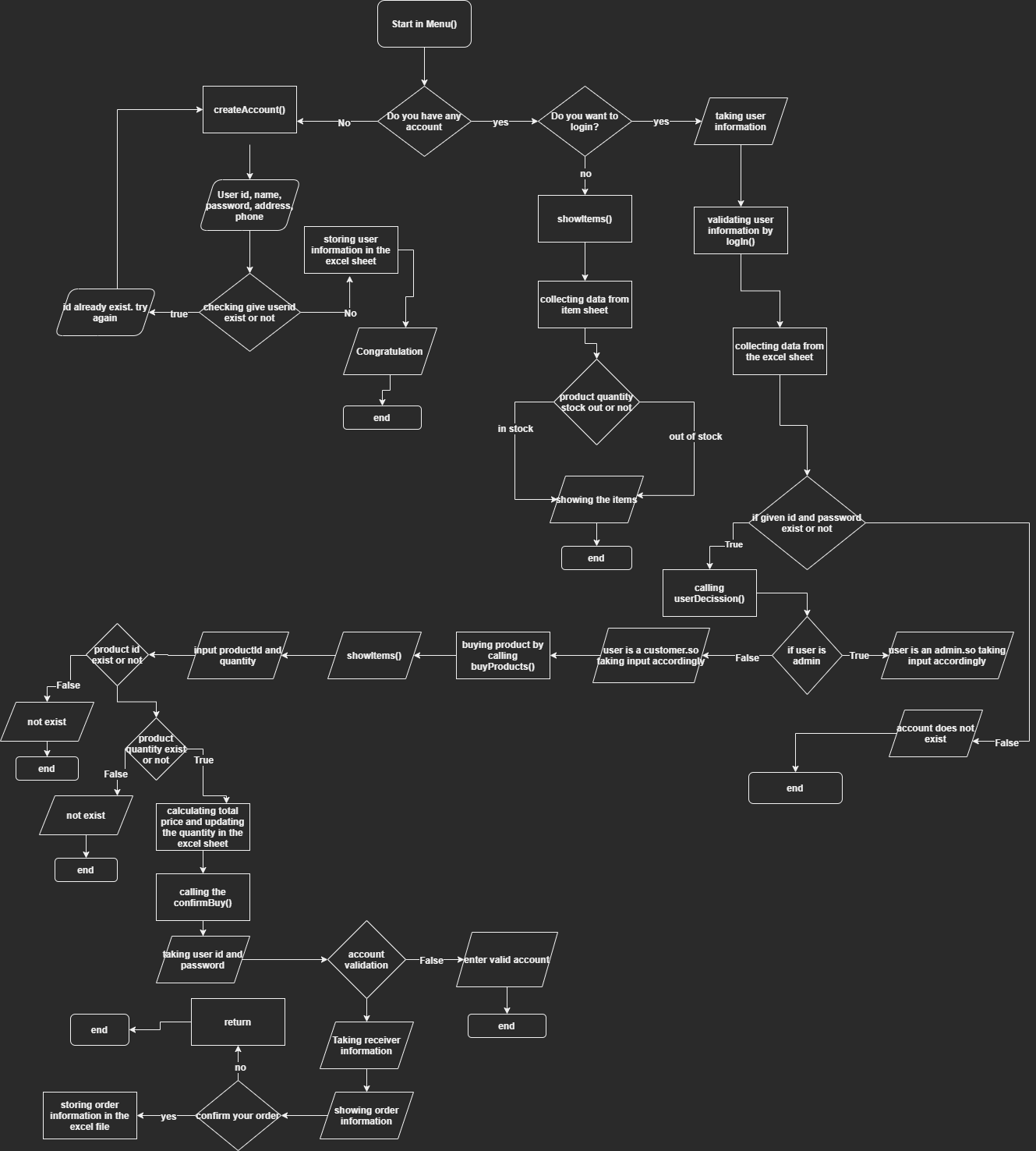
1. Storing all the information permanently, I used the file system. Actually, I used the openpyxl module to store user or product info. Updating, deleting, or adding a new product in the item table permanently, I used this openpyxl module. And by using this module, I can read and write Excel(.xlsx) file.
2. Whenever anyone trying to use this system for buying or other things, first s/he has to log in to the system. If s/he does not have any account, s/he must create an account before using this system. And new user’s information will be stored in the excel file.
3. So, the system can identify the user who is trying to login into the system by retrieving users' information from the “userAccount” sheet from the excel file. Then validate the user, and distinguish between admin and customer for giving choice accordingly.
4. After buying anything from this system, the order information will be stored in the “soldProduct” sheet.
5. Admin can update, delete or add new products in the item table. Then “items” sheet will be updated accordingly.
6. Showing item table or order information in a table, I used the tabulate module. By this module, I generate the table.
7. Giving manually unique product IDs is difficult. So that I create a giveId() to generate a unique product id by retrieving and adding 1 with the last product id of the item sheet.
8. I created a Menu() method for distinguishing the user and giving the choices accordingly. This method also checks the user validity by using the logIn() method.
9. The updateItems() or deleteItems() or writeItems() can use for one product at the same time. I used those functions under condition recursively to do this again and again.

# Flow chart:

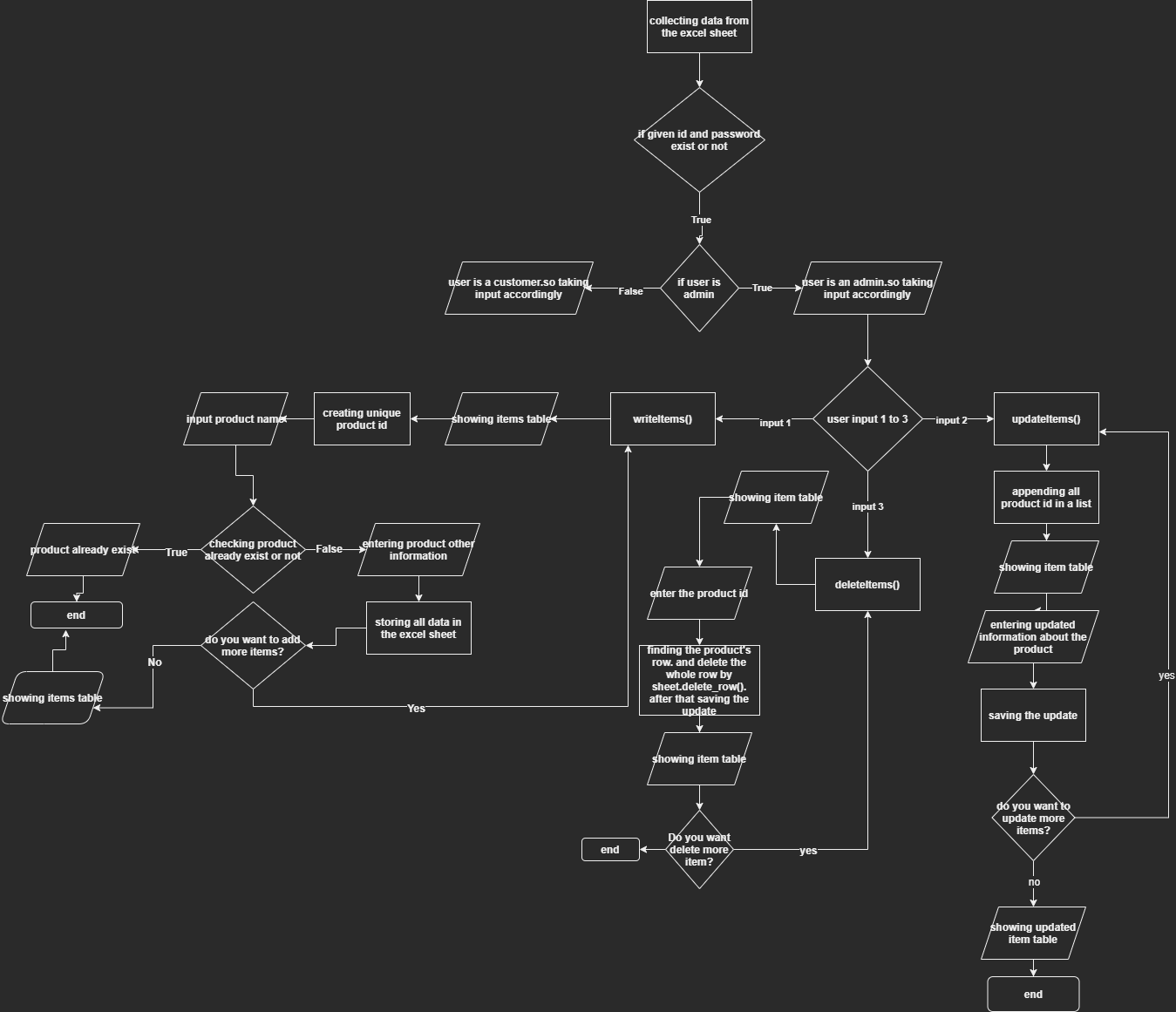
I have designed this program for two types of users.

1. Admin.
2. Customers.

That’s why I draw the flow chart on a different page for each user.



This is the flow chart of the customer and main menu.



This is the flow chart for the Admin

# Methods:

1. giveID()
2. createAccount()
3. logIn()
4. buyProducts()
5. confirmBuy()
6. writeItems()
7. checkExistingProduct()
8. printItems()
9. showItems()
10. updateItems()
11. deleteItems()
12. userDecission()
13. Menu()

# Code: