

- Initialize all the needed files if they don't exist()
- Choose the type of the user (purchase or admin)
- If the user wants to make a purchase:
 - Print available drinks to the screen
 - Choose drink
 - Choose quantity
 - Enter cash()
 - Get the drinks()
 - Get the change
 - Write the transaction in a file()
- If the user is an admin:
 - Enter username
 - Enter password
 - Check credentials correctness
 - Get the desired information from a file or write a modification to a file:

Get the drinks():

- Get the desired entity in the 2d array of structs based on the user's input
- Check if the desired quantity is less than the qty available (if not equate them)
- Check if the cash is enough for the transaction (and calculate change)

Write the transaction to the file(file name, line to remove, line to add):

- Open the file for read
- Read line by line and store them in a string
- If the line to remove is found:
 - Don't store it in the string and store the line to add in its place
- If there was no line to remove, append the new line at the end
- Remove the old file and write the string in a new file with the same name

Enter cash():

- Enter the amount in RM
- If the amount inserted isn't a multiple of 5,10 or 20:
 - Call the function again

Initialize files():

- Open the file for read
- If it doesn't open:
 - Create it
- Fill in the file with its default quantities (eg. prices.txt file: price name)

Files used:

- Prices.txt file:
 - Stores the prices of the drinks and their names in the format:
 - Price name
 - (e.g. 2 Cola)
- Quantities.txt file:
 - Stores the quantities of the drinks and their names in the format:
 - Quantity name
 - (e.g. 20 Cola)
- purchases.txt file:
 - Stores the transaction information in the format:
 - soldQty name price totalPrice leftQty
 - (e.g. 1 Coke 15 15 24)