**Phone-Dikhao Pseudo code**

**Admin Application**

**main.c:**

main(){

initDisplay()

login()

{

Ask Admin to enter Username and Password

validCredentials()

}

validCredentials()

{

if username and password are **not** “admin”

then

login()

}

displayTopsellers(){

Open file mobileData

Sort by count

Display top 5 or first 5 phones

}

displayMainMenu(){

print

1. Add Mobile
2. Delete Mobile
3. Edit Mobile
4. Search Mobile
5. View TopSellers
6. Exit

}

}

addMobile.c:

addMobile()

{

askDetails(){

take name, brand, price, discount, display\_flag, quantity, count, configurations(Ram,storage,chipset,camera)

Final price will be calculated from price and discount and will be stored in variable,

count should be initialized by 0, and when user buy we will increment it, hence we will not ask count as an input but will store it in variable

For display\_flag we will show menu to select which will be enum

Quantity will be taken as input but will be decremented when user will buy that phone

There will be different structure for configurations in which there will be ram,storage,cipset,camera as variables

}

confirm(){

Print “confirm”

Read

If (n/N)

displayMainMenu()

}

}

searchMobile.c:

displaySearchMenu()

searchByMobileName()

searchByBrancdName()

mobileFound()

Enter escape character to return to main menu

editMobile.c:

editMobile(){

displaySearchMenu()

isIdValid()

print “Enter id”

read editId

fopen

idFound = 0

do:

mobile = readMobile(file)

if mobile != NULL and mobile.id == idToDelete:

idFound = 1 // Set the flag if the ID is found

break // Exit the loop early since the ID was found

while mobile is not NULL

if !idFound:

print “Invalid ID.”

displaymainMenu()

displayEditMenu(){

switch

1. Price
2. Discount
3. display\_flag
4. Quantity

}

}

file = openFile("mobileData.bin", "rb")

if file is NULL:

error(\_\_func\_\_, \_\_LINE\_\_, "Failed to open mobileData.bin file")

print "Error: Unable to open mobileData.bin file."

return

mobile = readMobile(file)

printMobileDetails(mobile)

closeFile(file)

function openFile(filename, mode):

// Open a file with the specified mode

file = fopen(filename, mode)

return file

FILE \*openFile(const char \*filename, const char \*mode)

{

FILE \*file = fopen(filename, mode);

return file;

}

function closeFile(file):

// Close the file

fclose(file)

function readNextMobile(file):

// Read details of the next mobile phone from the binary file

// Return NULL if end of file is reached

mobile = createEmptyMobile()

readResult = fread(mobile, sizeof(MobileData), 1, file)

if readResult == 1:

return mobile // Successful read

else if feof(file):

return NULL // End of file

else:

error("readNextMobile", "Error reading mobile from file")

print "Error reading mobile from file."

return NULL

function confirm():

print "Confirm? (y/n): "

read confirmation

if (confirmation is 'n' or 'N'):

displayMainMenu()

else if (confirmation is 'y' or 'Y'):

return 1

else:

print "Invalid Input!"

displayMainMenu()

function isIdValid():

print "Enter the ID of the mobile to delete"

read idToDelete

file = openFile("mobileData.bin", "rb")

if file is NULL:

print "Error: Unable to open mobileData.bin file."

return 0

idFound = 0 // Initialize a variable to track whether the ID was found

do:

mobile = readMobile(file)

if mobile != NULL and mobile.id == idToDelete:

idFound = 1 // Set the flag if the ID is found

break // Exit the loop early since the ID was found

while mobile is not NULL

closeFile(file)

if !idFound:

print “Invalid ID.”

displaymainMenu()

return idFound

For display\_flag we will show menu to select which will be enum

**deleteMobile.c:**

displayOutdatedMobiles(){

outdatedCount=0

do:

mobile = readMobile(file)

if mobile != NULL and mobile.displayFlag == Outdated:

printMobileDetails(mobile)

outdatedCount = outdatedCount + 1

while mobile is not NULL and outdatedCount < 5

}

deleteMobile(){

Take id as input

confirm()?

If yes

Open file and delete structure

If no display main menu

}

filterByPrice(){

print “Enter min. Price”

read minPrice

print ”Enter max. price”

read maxPrice

file = openFile("mobileData.bin", "rb")

if file is NULL:

print "Error: Unable to open mobileData.bin file."

return

tempFile = openFile("tempMobileData.bin", "wb")

if tempFile is NULL:

print "Error: Unable to open tempMobileData.bin file."

return

foundMobiles = 0

do:

mobile = readMobile(file)

if mobile != NULL and mobile.finalPrice < maxPrice and mobile.finalPrice > minPrice:

printMobiledetails(mobile)

fwrite(&mobile, sizeof(struct MobileData), 1, tempFile);

foundMobiles = foundMobiles + 1

while mobile is not NULL

closeFile(file)

closeFile(tempFile)

mobileFound(foundMobiles)

buyMobile()

}

function mobileFound(foundMobiles):

if foundMobiles == 0:

print "No matching mobiles found."

displayMainMenu()

else:

print "-- Mobiles Found."

Function buyMobile:

Print “Want to buy?”

Read buyChoice

if (buyChoice is ‘n’ or ‘N’):

displayMainMenu()

if (buyChoice is ‘y’ or ‘Y’):

print “enter mobile name: ”

read mobName

tempFile = openFile("tempFile.bin", "rb")

if file is NULL:

print "Error: Unable to open tempFile.bin file."

return

foundMobiles = 0

do:

mobile = readMobile(tempFile)

if mobile != NULL and strcasecmp ( mobile.name, mobName)==0:

foundMobiles = 1

buyFile = openFile(“buyFile.txt”,”a”)

print “Enter your name”

read buyerName

print “Enter Quantity to buy”

read buyQuantity

mobile.count=mobile.count + buyQuantity

mobile.quantity=mobile.quantity-buyQuantity

fprintf(buyFile,”%s,%s,%d”,mobile.name,buyerName,buyQuantity)

fseek(tempfile, -sizeof(struct MobileData), SEEK\_CUR)

writeMobile(tempFile, mobile)

closeFile(tempFile)

closeFile(buyFile)

while mobile is not NULL

if !foundMobiles

print "No matching mobiles found."

displayMainMenu()

closeFile(file)

################################################################

function filterByPrice():

print "Enter minimum price: "

read minPrice

print "Enter maximum price: "

read maxPrice

file = openFile("mobileData.bin", "rb")

if file is NULL:

print "Error: Unable to open mobileData.bin file."

return

tempFile = openFile("tempMobileData.bin", "wb")

if tempFile is NULL:

print "Error: Unable to open tempMobileData.bin file."

closeFile(file)

return

foundMobiles = 0

do:

mobile = readMobile(file)

if mobile != NULL and mobile.finalPrice >= minPrice and mobile.finalPrice <= maxPrice and mobile.display\_flag!=Outofstock:

printMobileDetails(mobile)

fwrite(&mobile, sizeof(struct MobileData), 1, tempFile)

foundMobiles = foundMobiles + 1

while mobile is not NULL

closeFile(file)

closeFile(tempFile)

mobileFound(foundMobiles)

buyMobile()

function mobileFound(foundMobiles):

if foundMobiles == 0:

print "No matching mobiles found."

displayMainMenu()

else:

print "-- Mobiles Found."

function buyMobile():

print "Want to buy? (y/n): "

read buyChoice

if (buyChoice is 'n' or 'N'):

displayMainMenu()

else if (buyChoice is 'y' or 'Y'):

print "Enter mobile name: "

read mobName

tempFile = openFile("tempMobileData.bin", "rb")

if tempFile is NULL:

print "Error: Unable to open tempMobileData.bin file."

return

foundMobiles = 0

do:

mobile = readMobile(tempFile)

if mobile != NULL and strcasecmp(mobile.name, mobName) == 0:

foundMobiles = 1

// Ask for buyer information

print "Enter your name: "

read buyerName

print "Enter Quantity to buy: "

read buyQuantity

if buyQuantity <= mobile.quantity:

mobile.count = mobile.count + buyQuantity

mobile.quantity = mobile.quantity - buyQuantity

if mobile.quantity = 0:

mobile.display\_flag = Outofstock

buyFile = openFile("buyFile.txt", "a")

// Write the purchase information

fprintf(buyFile, "%s,%s,%d\n", mobile.name, buyerName, buyQuantity)

// Update the tempFile

fseek(tempFile, -sizeof(struct MobileData), SEEK\_CUR)

writeMobile(tempFile, mobile)

// Close the files

closeFile(tempFile)

closeFile(buyFile)

else:

print "Not Enough Quantity"

print "Operation Cancelled"

print "Max quantity mobile.quantity "

while mobile is not NULL

if !foundMobiles:

print "No matching mobiles found."

displayMainMenu()

#############################################################

function filterByPrice():

print "Enter minimum price: "

read minPrice

print "Enter maximum price: "

read maxPrice

file = openFile("mobileData.bin", "rb")

if file is NULL:

print "Error: Unable to open mobileData.bin file."

return

tempFile = openFile("tempMobileData.bin", "wb")

if tempFile is NULL:

print "Error: Unable to open tempMobileData.bin file."

closeFile(file)

return

foundMobiles = 0

do:

mobile = readMobile(file)

if mobile != NULL and mobile.finalPrice >= minPrice and mobile.finalPrice <= maxPrice and mobile.display\_flag != Outofstock:

printMobileDetails(mobile)

fwrite(&mobile, sizeof(struct MobileData), 1, tempFile)

foundMobiles = foundMobiles + 1

while mobile is not NULL

closeFile(file)

closeFile(tempFile)

mobileFound(foundMobiles)

buyMobile()

function mobileFound(foundMobiles):

if foundMobiles == 0:

print "No matching mobiles found."

displayMainMenu()

else:

print "-- Mobiles Found."

function buyMobile():

print "Want to buy? (y/n): "

read buyChoice

if (buyChoice is 'n' or 'N'):

displayMainMenu()

else if (buyChoice is 'y' or 'Y'):

print "Enter mobile name: "

read mobName

tempFile = openFile("tempMobileData.bin", "rb")

if tempFile is NULL:

print "Error: Unable to open tempMobileData.bin file."

return

foundMobiles = 0

do:

mobile = readMobile(tempFile)

if mobile != NULL and strcasecmp(mobile.name, mobName) == 0:

foundMobiles = 1

// Ask for buyer information

print "Enter your name: "

read buyerName

print "Enter Quantity to buy: "

read buyQuantity

if buyQuantity <= mobile.quantity:

mobile.count = mobile.count + buyQuantity

mobile.quantity = mobile.quantity - buyQuantity

if mobile.quantity == 0:

mobile.display\_flag = Outofstock

buyFile = openFile("buyFile.txt", "a")

// Write the purchase information

fprintf(buyFile, "%s,%s,%d\n", mobile.name, buyerName, buyQuantity)

// Update the tempFile

fseek(tempFile, -sizeof(struct MobileData), SEEK\_CUR)

writeMobile(tempFile, mobile)

// Close the files

closeFile(tempFile)

closeFile(buyFile)

else:

print "Not Enough Quantity"

print "Operation Cancelled"

print "Max quantity mobile.quantity "

while mobile is not NULL

if !foundMobiles:

print "No matching mobiles found."

displayMainMenu()