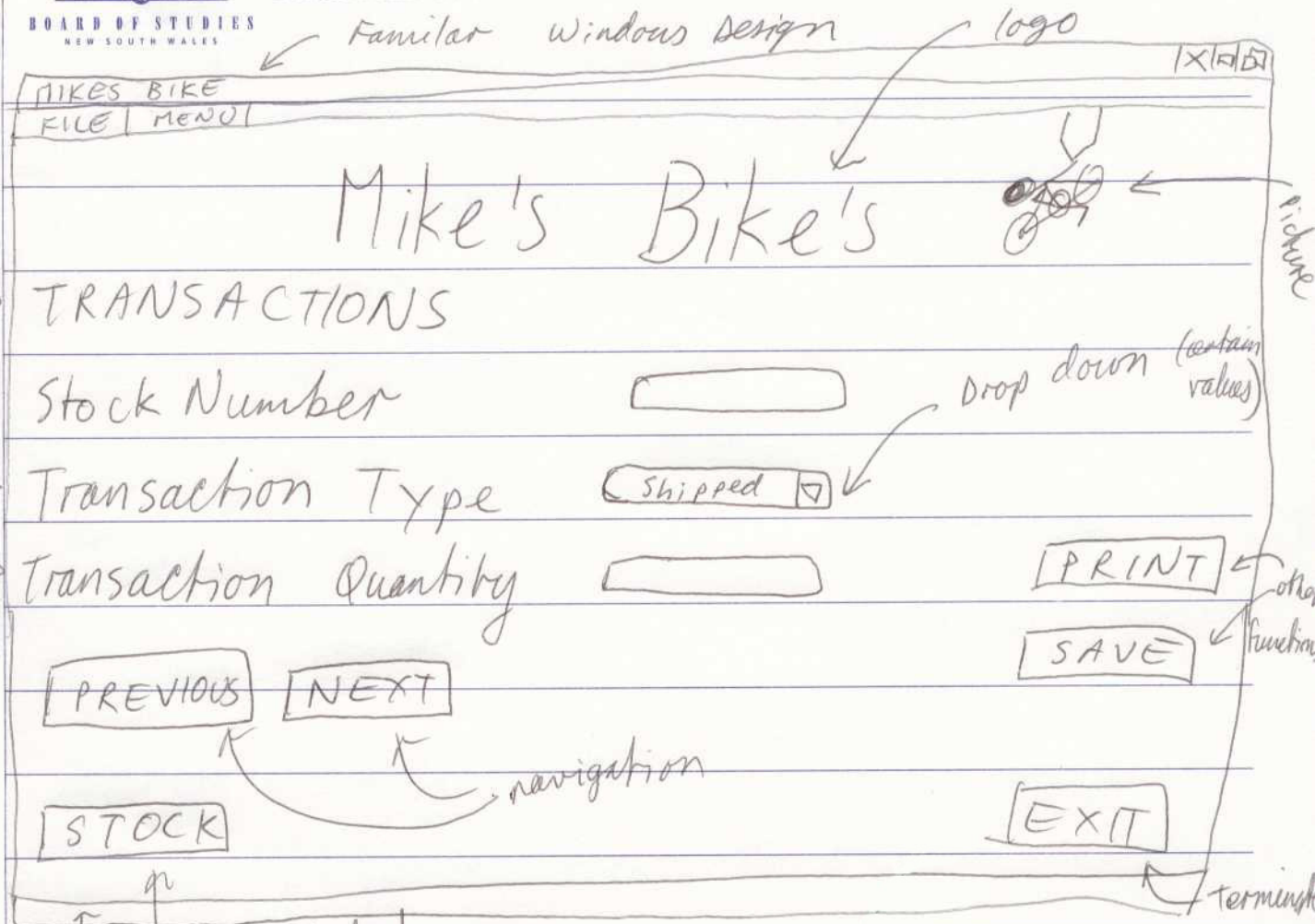
22.



ii)

INPUT	PROCESS	OUTPUT
Get Stock From File	Determine Change	Save change to file
Get Quantity From File		
Get ID		

Question 22

BOARD OF STUDIES
NEW SOUTH WALES

Centre Number: ..... Student Number: .....

ii)

INPUT	PROCESS	OUTPUT
request Stock Number	Goto Transaction File	Return StockNumber
request Quantity	Goto Transaction File	Return Quantity
Stock Number	Calculate stock change	Update Inventory File ~~Trade~~
Quantity		

iii) Get info from Transaction, read into an Array

Get info from Inventor, read into an Array

Check Transaction file

Assumming Inventory in order so Array index, see how long array is.

Menage for all Transaction numbers over array index

Read rest of Transaction into Inventory

Print out if Quantity < 3.

BOARD OF STUDIES
NEW SOUTH WALES

Question 22

Centre Number: ..... Student Number: .....

```
BEGIN MainProgram BikeTracker
    Read Transaction into Array
    Read Inventory into Array
    Transaction to Inventory Errors
    Read Transaction into Inventory
    Inventory deficiencies
END MAin program BikeTracker

BEGIN SUB ReadTransaction into Array
Declare Counter as Local integer
TYPE TranRec
    StockNo As integer * 3
    Quantity As integer * 4
END TYPE
Declare TranArray of TranRec from 1 to 1000 As Array of Records.
counter = 0
WHILE not EOF or StockNo <> 999
    increment counter
    read StockNo (counter) in file
    TranArray [counter]. StockNo = StockNo
    read Quantity (counter) in file
    TranArray [counter]. Quantity = Quantity
ENDWHILE
END SUB ReadTransaction into Array
```

Question 22

BOARD OF STUDIES
NEW SOUTH WALES

Centre Number: ..... Student Number: .....

```
BEGIN SUB  Read Inventory into Array
TYPE InvRec
   StockNo As integer *3
   Quantity As integer * 4
   Item As string *15
   UnitPrice As Currency * 5      (999.99)   3 dollars, 2 decimals
ENDTYPE
Declare InvArray of InvRec from 1 to 100 As Array of Records
Declare count as Local integer
count = 0
WHILE not EOF OR StockNo <> 999
   increment count
   Read StockNo(count) from file
   InvArray [count].StockNo = StockNo
   Read Quantity(count) from file
      InvArray [count].Quantity = Quantity
   Read Item(count) from file
      InvArray [count].Item = Item
   Read UnitPrice(count) from file
      InvArray [count].UnitPrice = UnitPrice
ENDWHILE
ENDSUB Read Inventory into Array.
```

Question 22

BOARD OF STUDIES
NEW SOUTH WALES

Centre Number: ..... Student Number: .....

```
BEGIN SUB Transaction into Inventory Errors

Dim Count1 as Local integer
count1 = 0
DIM Count2 as Local integer
comt 2 = 0
DIM StockMax as local integer
stock Max = 0
WHILE NOT EOF for InvArray
  count 1 = count 1 + 1
ENDWHILE
WHILE NOT EOF for Tran Array
  comt 2 = count 2 + 1
  IF Tran Array [count 2]. stockNo > StockMax THEN
    Stock Max = Tran Array [count 2]. Stock No
  ENDIF
ENDWHILE
IF StockMax > comt 1 THEN
  WHILE stockMax > count 1
    Display StockMax " is an invalid stock"
    count decrement Stock Max
  ENDWHILE
ENDIF
END SUB Transaction into Inventory Errors.
```

BOARD OF STUDIES
NEW SOUTH WALES

Question 22

Centre Number: ..... Student Number: .....

```
Begin SUB Read Transaction into Inventory
    DIM ~~Stock~~ QuanChange AS Integer
    DIM StockChange AS Integer
    DIM count as Local integer
    count = 1, stockchang = 0, Quanchange = 0
    WHILE TranArray not EOF or TranArray[count].StockNo <> 999
        ~~TranArray~~ StockChange = TranArray[count].StockNo
        ~~Stock~~ QuanChang = TranArray[count].Quantity
        InvArray[StockChange].Quantity = InvArray[StockChange].Quantity
                                         + QuanChange
        increment count by 1
    ENDWHILE
ENDSUB Read Transaction into Inventory

BEGINSUB Inventory Deficiencies
    DIM count as Local integer
    count = 1
    WHILE InvArray not EOF OR ~~Stock~~ InvArray[count].StockNo <> 999
        IF InvArray[count].Quantity < 3 THEN
            Display "less than 3 of this item:",
            Display InvArray[count].Item
        ENDIF
    ENDWHILE
ENDSUB Inventory Deficiencies.
```

Question 22

BOARD OF STUDIES
NEW SOUTH WALES

Centre Number: ..... Student Number: .....

b) User Documentation

- Made in 3rd Stage : Implemation
- Includes : User Manual, Installation Guide, Copyright, Liscence, Quick Reference Guide, Troubleshooting Guide
- The documentation for the user must be completed Here as the project has been implemented + only needs to be tested. Changes will not occur in the design of the program, only algorithms + code, Hence User can be given guides now.

Process Diaries

Made here as well but not part of project

- Made From Stage 1 to 4: All except Maintenance
- Allows any maintenance engineer to see problems encountered + how they were fixed. Allows him/her to determine whether these fixes can be used to current situation, or whether solution was only temporary
- The Documentation occurs in every stage because changes and planning + solving problems must be done in every stage.