Python Mini Project

Problem Statement:

As in modern times it’s important to compile all the data of a company's production on a computer system rather than maintaining books for the same. There are several advantages of using an application for inventory management than book keeping because it makes less human calculation work and hence decreasing human errors. It’s easy to store and find a single data from a thousand’s on data when used as an application rather than finding in the 100 of pages of the books maintained. So our aim is to develop a desktop application that manages all the records of entries of raw material, production and sales of products in thread dyeing factory

Scope of the project:

The system consists of the interface for operations like add, modify, delete and view for all the types of raw material used in the company, for maintaining or creating the recipes of the end products.

The application consists of an inventory management system that performs different operations like addition, modification and deletion of raw material, shade number and also allows the user to view stock for particular raw material or shade number using several parameters like view of any transactions by transaction Id or view by today’s date or view by custom dates.

The system also includes adding the details of sales i.e. the transaction of quantity of end product sold. All the above stages have operations like add records, modify the records, delete present records or view the records using several parameters like view of any transactions by transaction Id or view by today’s date or view by custom dates..

The additional operations are to view the closing stock of all the raw materials and the end products and the transactions for a given raw material or product in a date range.

The system provides a facility of lot number which means distributing the complete stock in small divisions and maintaining small division stock along with the total stock combined in several lots. This helps in an easy and very effective maintaining method for inventory management.

The application also provides an interface for the user to add, modify, delete and view the opening stock for all the raw material and end products or processed products.

This system manages all the transactions that occur in a thread dyeing factory from the records of buying the raw materials to the sales of the end products produced in the factory.

The system does not provide the printing operations for any transactions or any details at present.

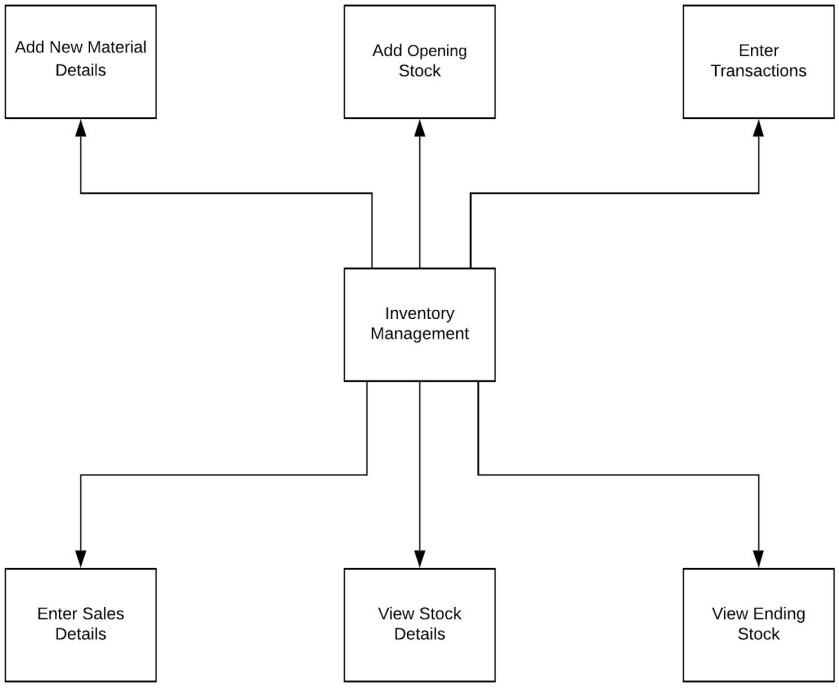
Team Members:

1. Nishit Shah(1811114)
2. Parth Shah (1811115)
3. Nilay Sheth(1811121)

Proposed Model:

Based on Scope of our project our application comprised of six main parts/components:

1. **Add New Material Details**
2. **Add Opening Stock**
3. **Enter Transactions**
4. **Enter Sales Details**
5. **View Stock Details**
6. **View Ending Stock**



Now we move to each of the functions performed by the above components.

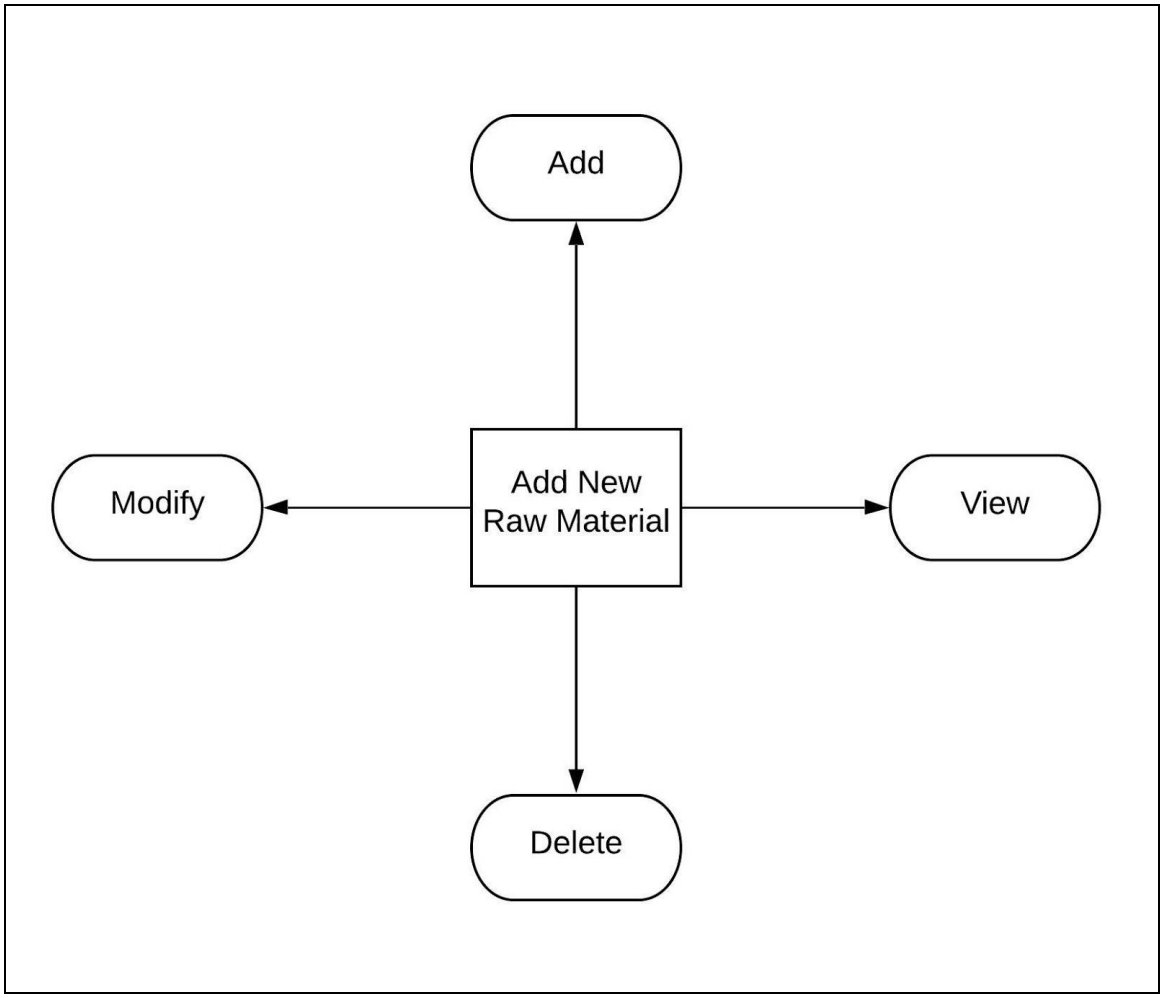
1. **Add new Material Details**

   It consists of two subtasks which are as follows:

   1. **Add new Raw Material**
   2. **Add new Shade number**

   Each of the subtask consists of four main operations which can be performed :

   1. Add
   2. Modify
   3. Delete
   4. View



Sr No.	Operations	Function
1	**Add (For Raw Material)**	In Add option,user can add the raw material detail into the database. It takes input as product code , product name , product price and product type and outputs if the entry was added successfully or not.
	Add (For Shade Number)	It takes input as shade number and product code and product percentage by which that shade number has been made and outputs the entry was added successfully or not.
2	**Modify (For Raw Material)**	In the modify option user can modify the details of the raw material added into the database. It takes input product code and output all the details of the raw material and the user can modify the same.
	Modify (For Shade Number)	It takes input shade number which has been added into the database and output its product code and product percentage. Once modified these details it outputs whether modification is successful or not.
3	**Delete**	User can delete the raw material detail which have been added into the database.It takes input as (product code)/(shade number) and output if the product has been deleted successfully or not.
4	**View**	User can view all the raw material details which have been added into the database.

2. Add Opening Stock

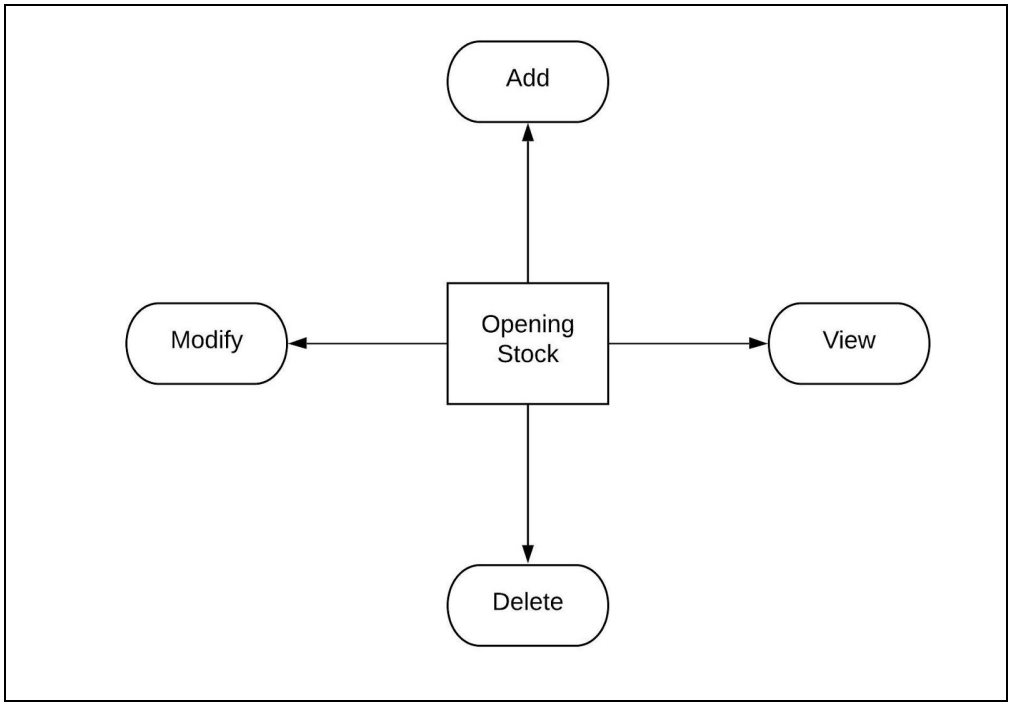
It consists of two subtasks which are as follows:

1. **Opening Stock of Raw Material**
2. **Opening Stock of Shade number**

User can manage the opening stock of each of the raw material added into the database.

Operations which can be performed on the above sub-tasks are as follows:

1. Add
2. Modify
3. Delete
4. View



Sr No.	Operations	Function
1	**Add (For Raw Material)**	In add operation user can add opening stock of raw material. It takes input as product code, lot number and opening stock and outputs if opening stock was added successfully or not.
	Add (For Shade Number)	It takes input as shade number , product code, lot number and opening stock and outputs if opening stock was added successfully or not.
2	**Modify (For Raw Material)**	User can modify opening stock of raw material. It takes input as product code and outputs product name , lot number and opening stock and upon modifying these changes it outputs whether modification was successful or not.

	Modify (For Shade Number)	It takes input as shade number and displays product code , product name , lot no. , opening stock upon modification it outputs whether modification is successful or not.
3	**Delete**	User can delete the opening stock of raw material detail which have been added into the database.It takes input as product code / shade number and output if the product has been deleted successfully or not.
4	**View**	User can view the opening stock raw material details which have been added into the database. It takes input product code / shade number and outputs the respective details.

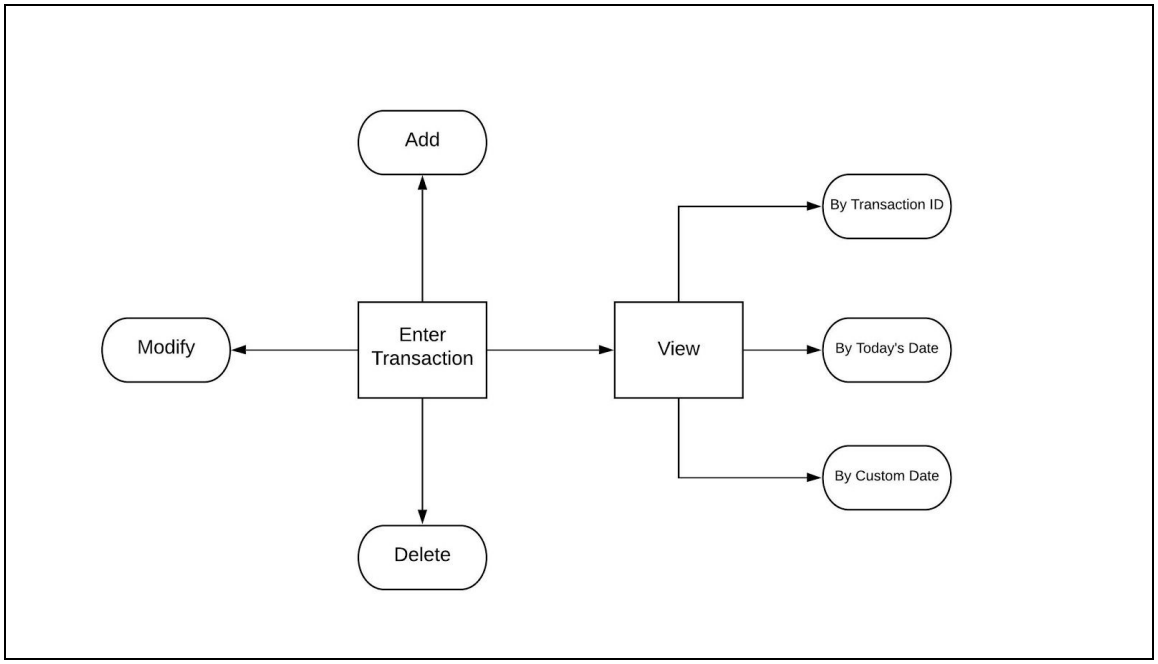
3. Enter Transactions :

It consist of two sub tasks which are as follows :

1. **Transaction of Raw Material**
2. **Transaction of Shade Number**

Operations which can be performed on the above sub-tasks are as follows:

1. Add
2. Modify
3. Delete
4. View
    a. By Transaction Id
    b. By Today's Date
    c. By Custom Date



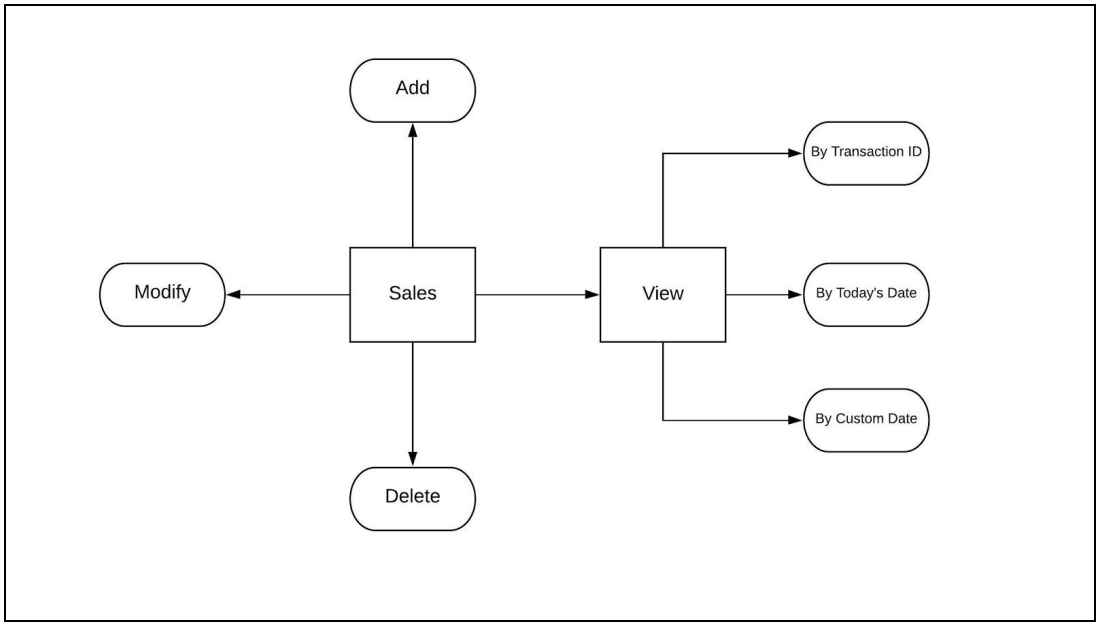
Sr No.	Operations	Function
1	**Add Transaction (For Raw Material)**	User can add transactions of the raw material. Transaction Id and Date is auto-generated from the database. User inputs customer name ,remark and list of product code , quantity and lot no. of raw material which is added into the database and outputs whether the transaction was successfully added or not.
	Add Transaction (For Shade Number)	Transaction ID and Date is auto-generated from the database. User inputs customer name , remark and shade number which is already added into the database and list of product code , quantity and lot no of the raw material and total quantity of the raw material is used to calculate the quantity , price of every colour in the shade number based on their respective percentage and outputs whether the transaction was successfully added or not.

2	**Modify Transaction (For Raw Material)**	User can modify the transaction of raw material. User inputs the transaction id and it displays all the details of that transaction and user can modify the same but User cannot modify the transaction id and date. After modification it outputs whether modification was successful or not.
	Modify (For Shade Number)	It inputs the transaction id of the shade transaction and it displays the details of that transaction. Upon modification of the details it outputs whether the modification was successful or not.
3	**Delete**	User can delete the transactions. It inputs transaction id and outputs whether deletion is successful or not.
4	**View**	User can view there transaction using three methods i.e. By Transaction ID , By Today's Date and By Custom Date . User inputs transaction id in by transaction id and application displays the details of that transaction id. By Today's Date displays all the transaction generated today . And user can input custom date for viewing all the transaction between that dates.

4. Sales Details

Sales details manages the sales of the final product produced. It comprises of four main Operations :

1. Add
2. Modify
3. Delete
4. View



Sr No.	Operations	Function
1	**Add Sales**	User can add sales details. It inputs customer name , remark , shade number , product code , quantity and lot no. Transaction id and date is auto generated from the database. Once user inputs the data and confirms it the application outputs the closing stock of the product for that shade number and also whether sales was added into database or not.
2	**Modify Sales**	User can modify the sales transactions . It inputs the transaction id of the sale user want to modify and system displays all the details of that transaction id and after modification it displays the closing stock and whether sales details was modified or not.
3	**Delete**	Users can delete the sales transaction by giving the transaction id which they wish to delete and then the application displays all the details of that transaction before deletion . Once the user confirms deletion the sales transaction is deleted from the database .

4	**View**	User can view it's sales transaction using three methods i.e. By Transaction ID , By Today's Date and By Custom Date . User inputs sales transaction id transaction id and application displays the details of that transaction id. By Today's Date displays all the sales transactions generated today . And user can input custom dates for viewing all the transactions between those dates.

5. View Stock details:

User can view stock details of the following products :

1. Raw Material
2. Shade Number

Sr No.	Operations	Function
1	**View Stock Details (For Raw Material)**	Users can view the stock details of product code in particular lot no. or in all the lot no. User inputs the product code , lot no. for particular lot no. or else "all" for all of the lot and custom dates and application outputs the transaction id , date and transaction type i.e IN or OUT , lot no., Opening Stock , and also Closing Stock.
2	**View Stock Details (For Shade Number)**	User inputs shade number , product code and lot no. for particular lot no. or else "all" for all of the lot no. and custom date and application outputs the transaction id ,date , transaction type , opening stock and closing stock .

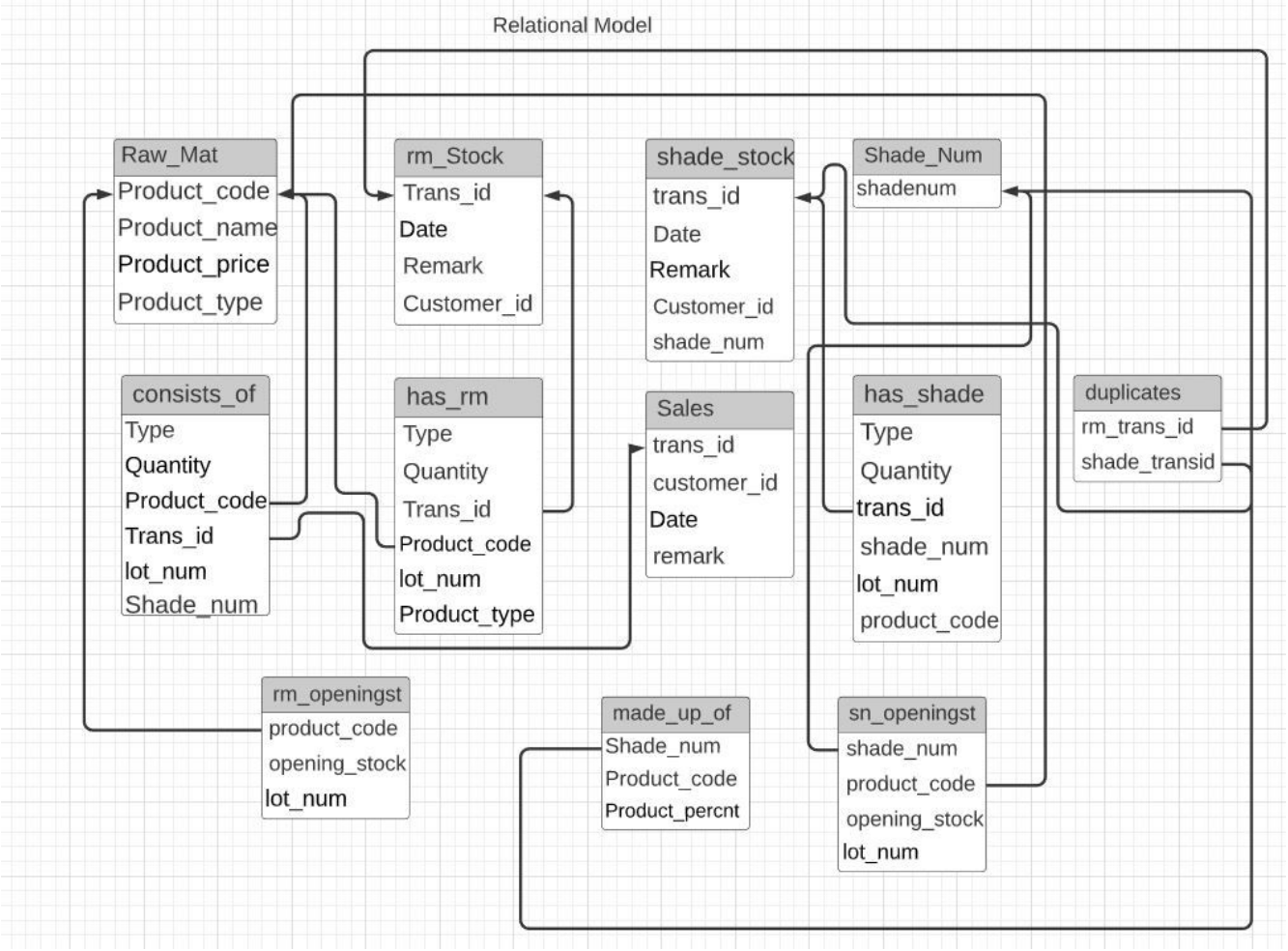
6. View Ending Stock :

User can view the ending stock of following products:

1. Raw Material
2. Colour
3. Shade Number

The closing stock for raw material and colour shows the closing stock for all the available raw material and colours data till date.
For shade number the user has to enter the combination of shade number and product code to find the closing stock for it till date. Users can enter several input data together.
The Negative Closing stock is displayed in red and positive closing stock in green .

This is the **relational model** of our Database :

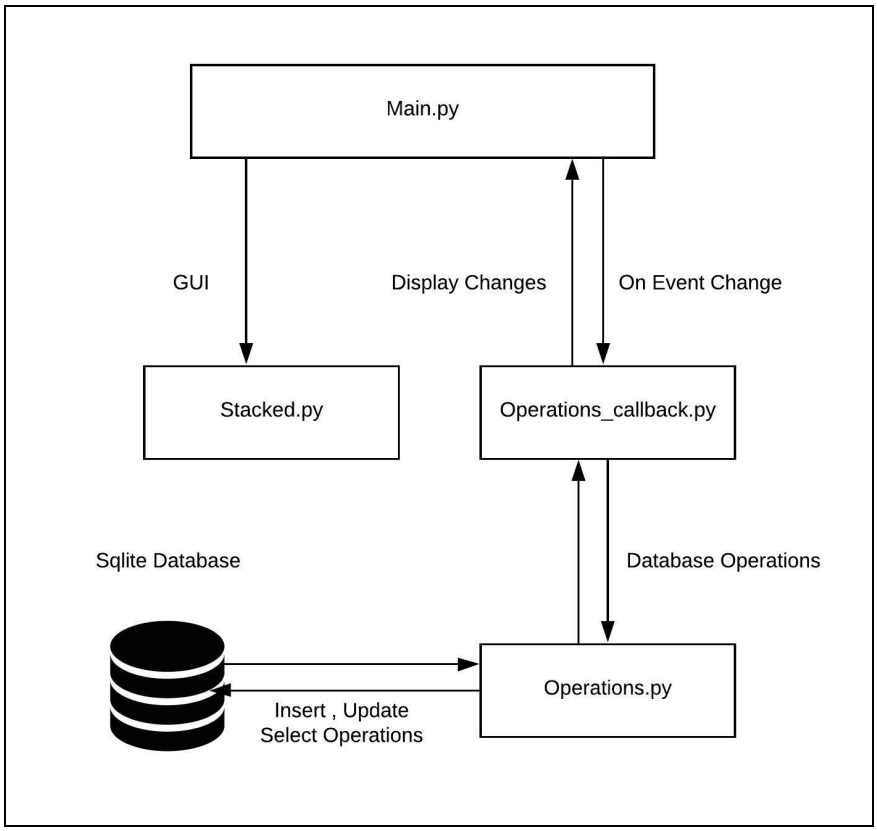


Implementation details:

Github Link: https://github.com/nilay1221/Inventory_Management

The implementation is done with the help of pyqt module in frontend, python 3 in backend and sqlite for database. The basic structure of gui is made with the help of Qt designer. The application is structured in such a way that even in future if any changes are to be made in the ui the current functions are unaffected and similarly if new functions are added or database is changed the ui is unaffected.

The flow diagram for our system is given below:



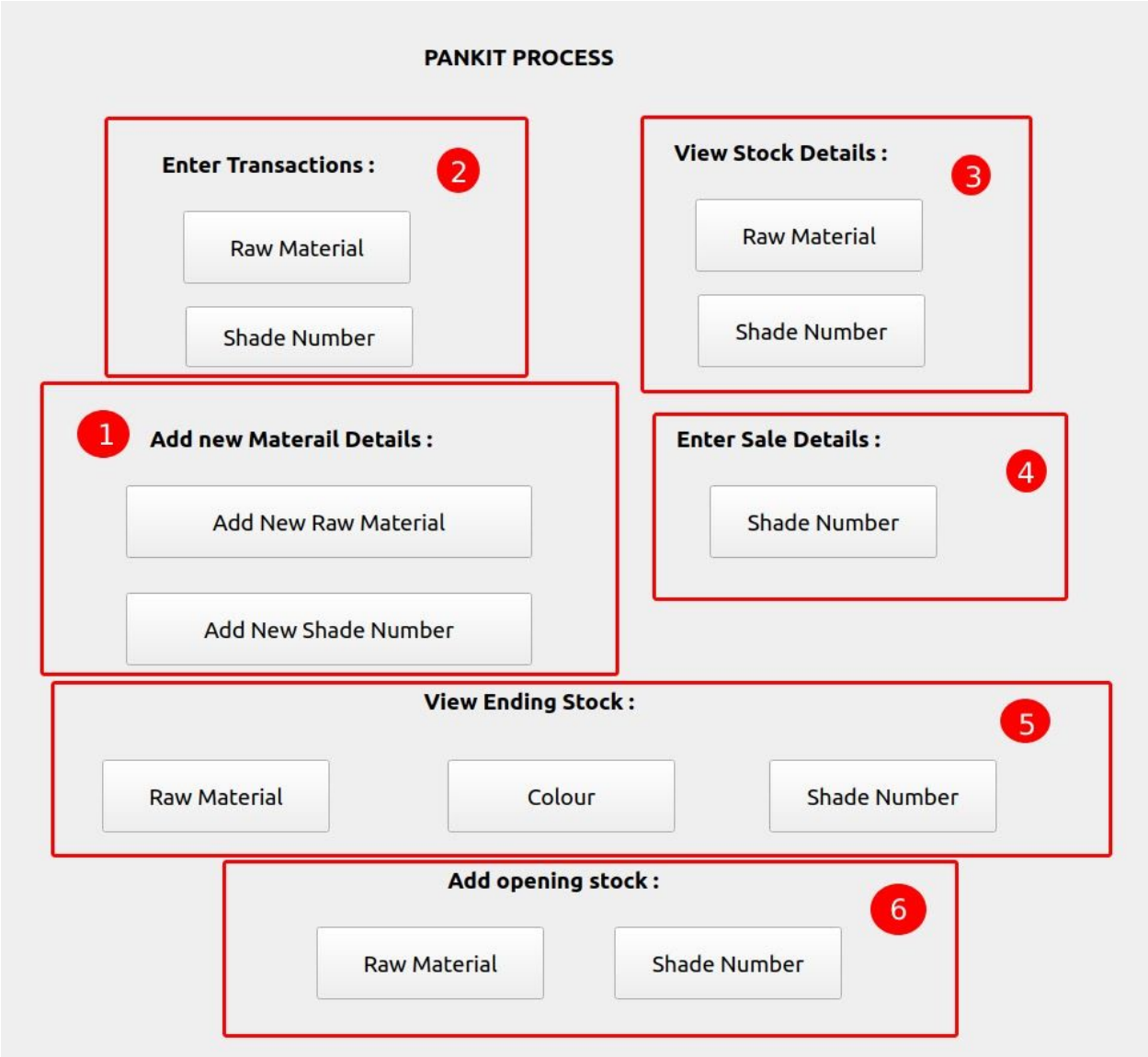
1. **Main.py :** This file is the integration of the GUI with its callback and indirectly with the database operations . It consists of pages_dict which has all the defined pages of Stack View defined in Stacked.py. Main Window class is the core part of it which links the stacked.py and its functions in it. It has a method startUIWindow in which all the pages are linked with one another. Maintain_operations method has been used to link the callback functions of the GUI interface which are defined in operations_callback.py . Operations_callback is linked to all of the database operations / functions defined in Operations.py . For clearing the elements of the GUI clear.py is linked which has

functions defined for the same. Segmentation of each of the main components gives flexibility for writing code as well as maintaining the integrity of the code i.e Any error caused in one does not affect the whole of the Application .

2. **Stacked.py :**Stacked.py file is the file that is used for designing gui. This file uses a stack widget so that all the gui files are bundled in single widget so that even in future, if one needs to add new gui then this other files will remain unaffected and independent of other files.This file contains other widgets like Qtable widget for displaying table, QLineEdit for displaying text boxes, Qlabel widget for Labels, QButton for displaying buttons. Every object in this file has a unique name so whenever new ui will be added the current operations and ui is not affected.This file has the complete ui.
3. **Operations_callback.py :** Operation_callback is the file which declares all the functions which are to be called on the buttons clicked that are the add,view,modify,delete,back functions in different modules of the system. This file consists of the functions that are the mediator between gui and database.According to the module operations the functions are designed like for add operations the function takes the input data from the current ui page and stores into the database or for functions like view, the function collects the data from the database and displays onto the current ui. Operations like modify first collect the data from the database prints on the ui and then the changes in the data in the ui page are committed back to the database.This file handles all the errors that the user makes while inputting a given field like entering a non existent shade number or incomplete data entered,etc. There are appropriate warning messages or info messages to display the errors or whether the transaction for successful or unsuccessful.
4. **Operations.py :** It consists of functions which link our sqlite database used for the Inventory management. Functions are used to insert , update , delete , query from the database .This file is only linked to the database and operation_callbacks file. It takes the input parameters from the operations_callback file and execute sql queries accordingly and collects the result from the database and return the result to the operations_callback file.For example it return success, failure or errors for operations like add, modify,delete and returns a collection of data for view operations, This file handles all the errors related to the database like constraint errors and simultaneously returns the message to the operation_callback file which displays the message to the user.
5. **NewDatabse.db :** This is the local database file for the system. It contains all tables and structures of the data. All the data is saved and used from this file for all the operations in the application.
6. **Clear.py :** This file consists of functions of clear button present. This file describes the operation to be performed on clicking the clear button that is when a clear button for a particular interface is clicked then the parameters to be cleared are described in clear.py file.

Screenshots of the application :

This is the home page. The details of the red marked modules are given below.

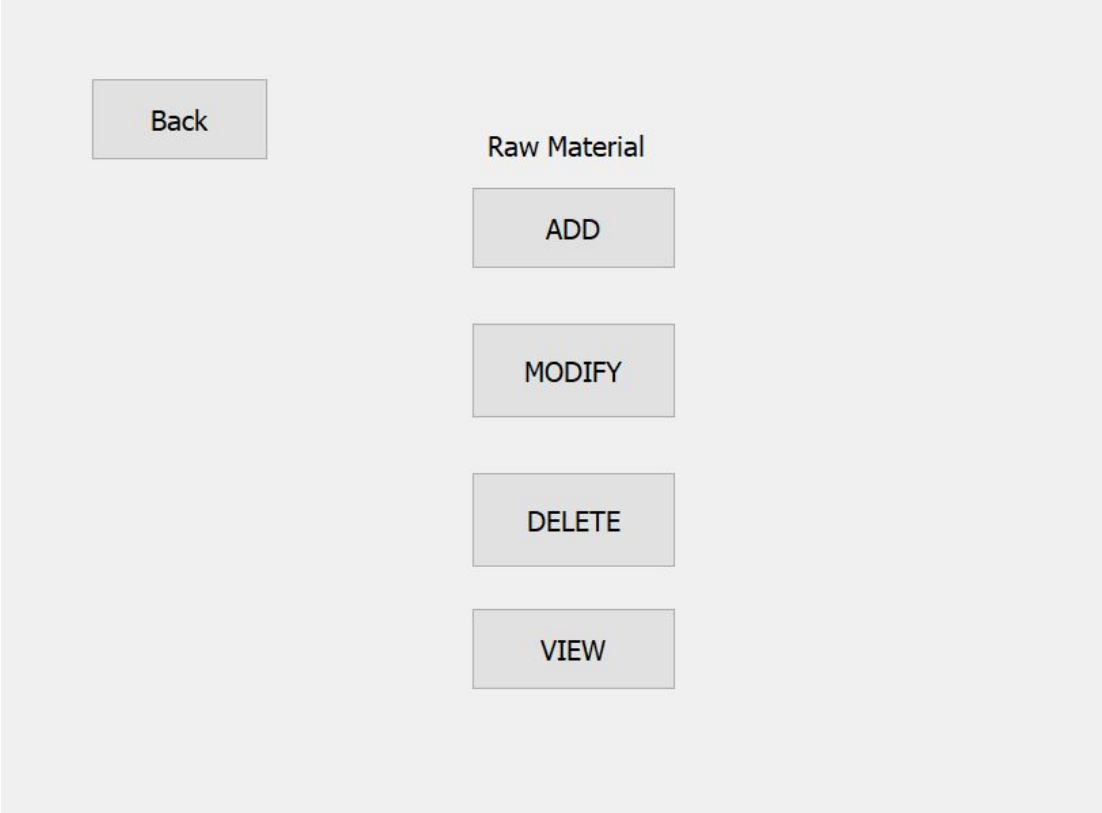


The modules in the above image are:

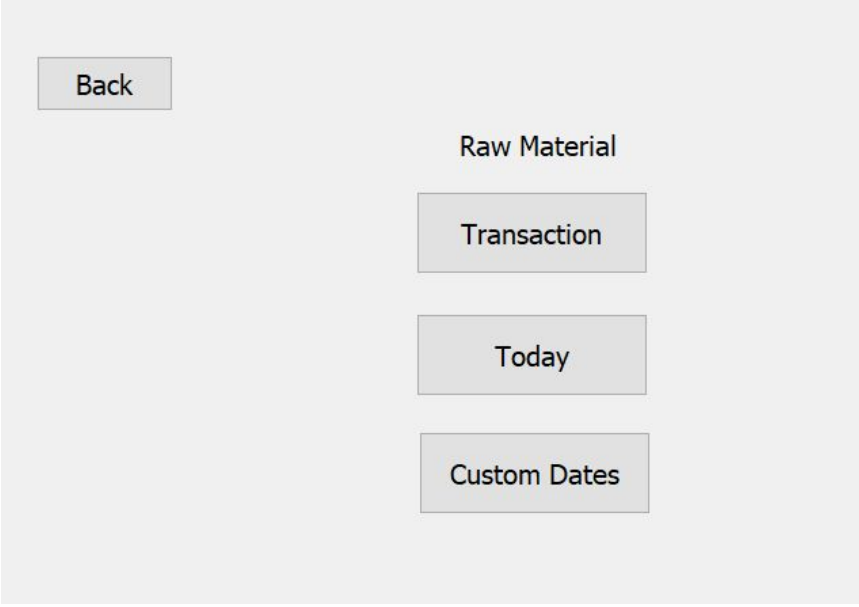
1. **Add New Material Details**
2. **Add Opening Stock**
3. **Enter Transactions**
4. **Enter Sales Details**
5. **View Stock Details**
6. **View Ending Stock**

Some screenshot for each of the modules are as followed:

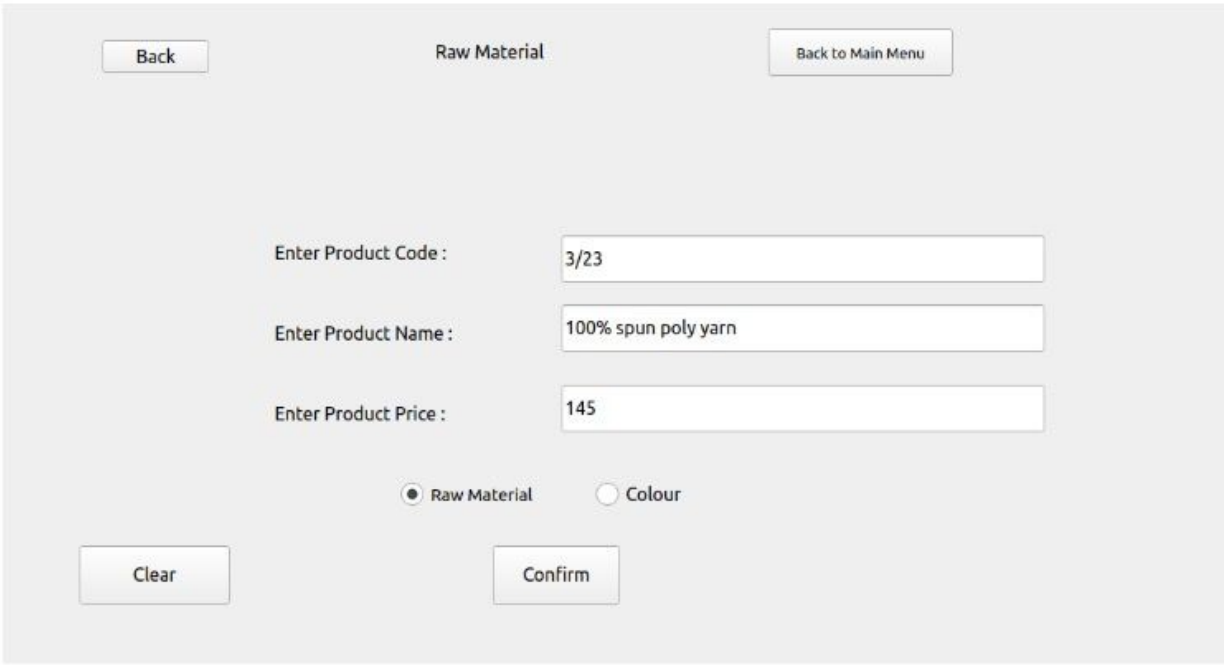
Most of the modules like add new material details, add transactions, add sales details, add opening stock have 4 operations each which are add,view modify,delete and view. Their gui looks like:

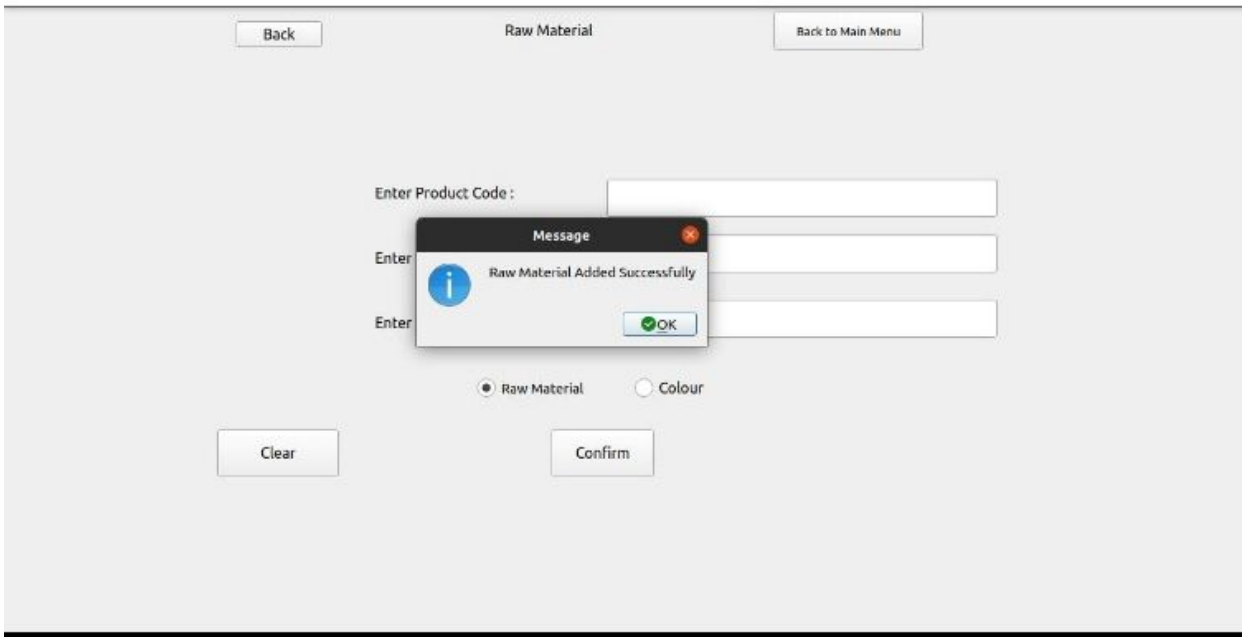


Add transaction and sales modules have further three options for view operations they are:

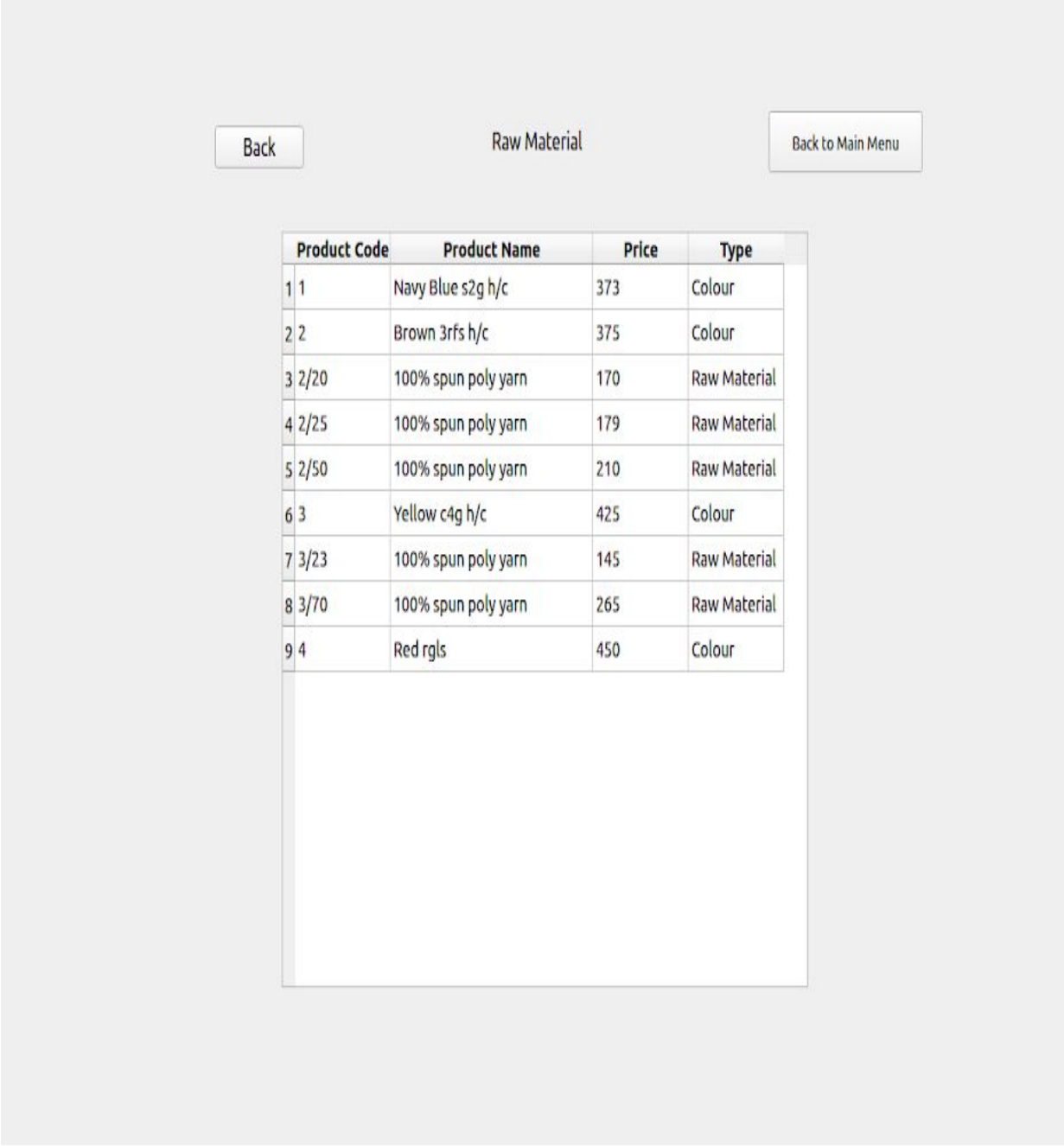


Add new Raw Material:



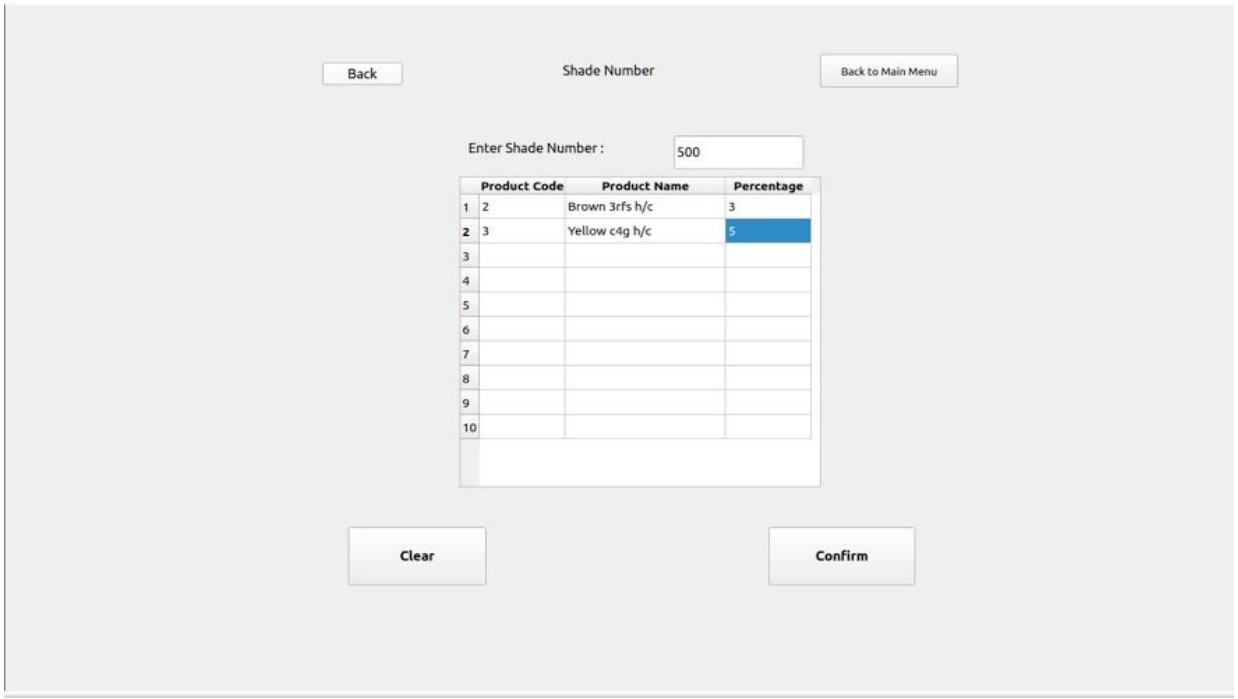


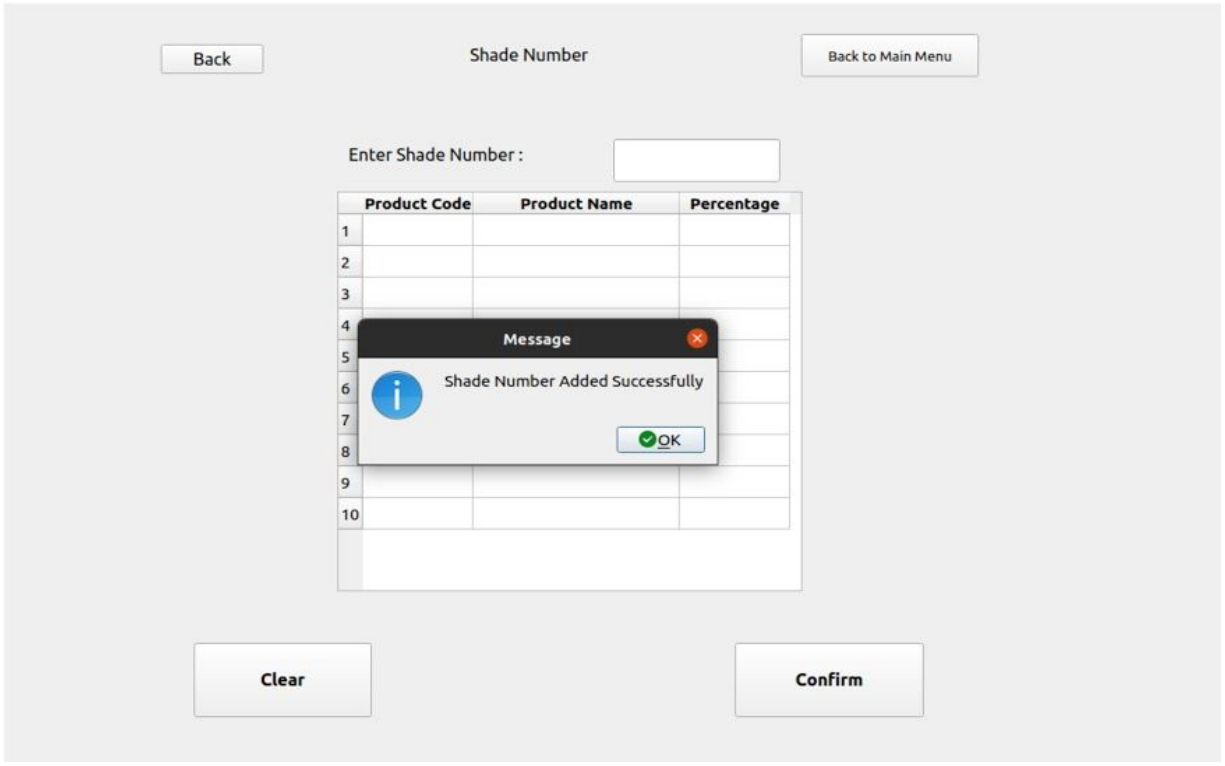
View New Raw Material :



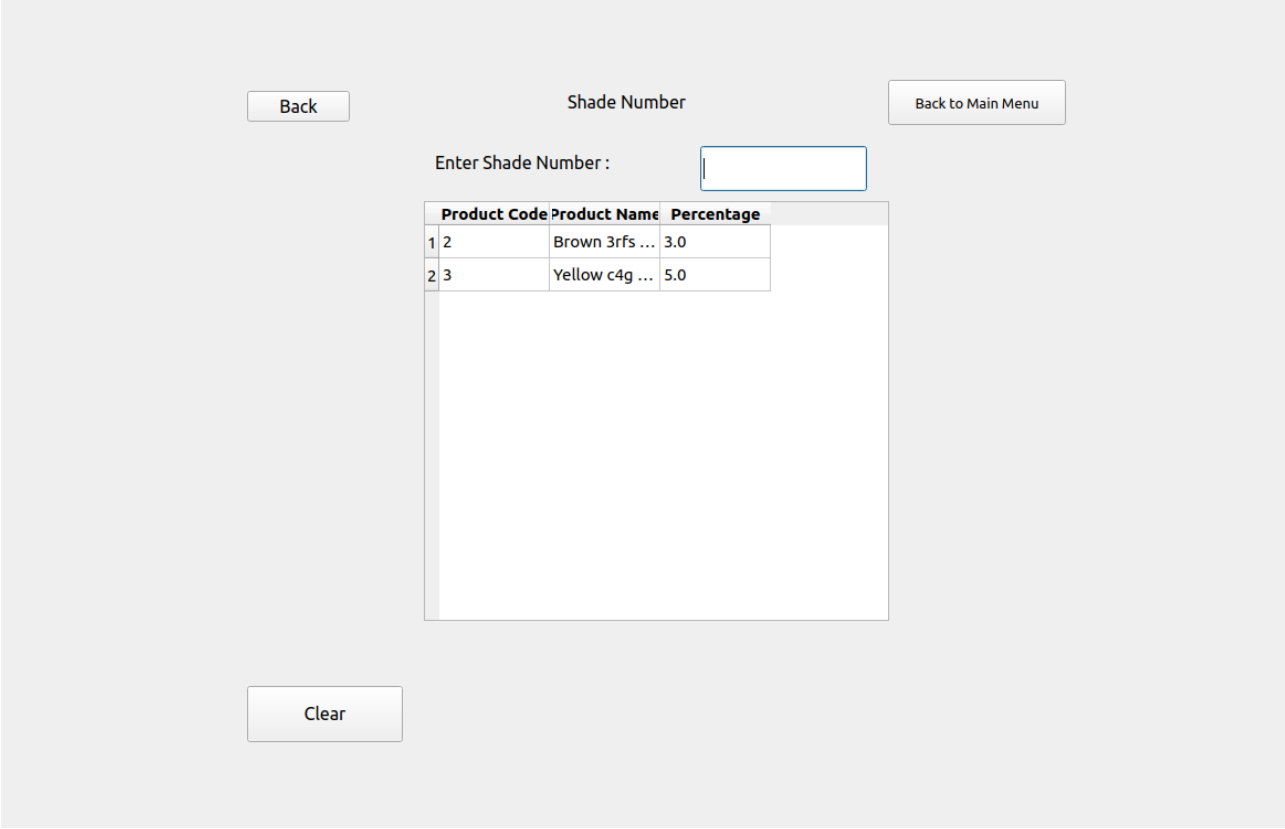
	Product Code	Product Name	Price	Type
1	1	Navy Blue s2g h/c	373	Colour
2	2	Brown 3rfs h/c	375	Colour
3	2/20	100% spun poly yarn	170	Raw Material
4	2/25	100% spun poly yarn	179	Raw Material
5	2/50	100% spun poly yarn	210	Raw Material
6	3	Yellow c4g h/c	425	Colour
7	3/23	100% spun poly yarn	145	Raw Material
8	3/70	100% spun poly yarn	265	Raw Material
9	4	Red rgls	450	Colour

Add new Shade number :

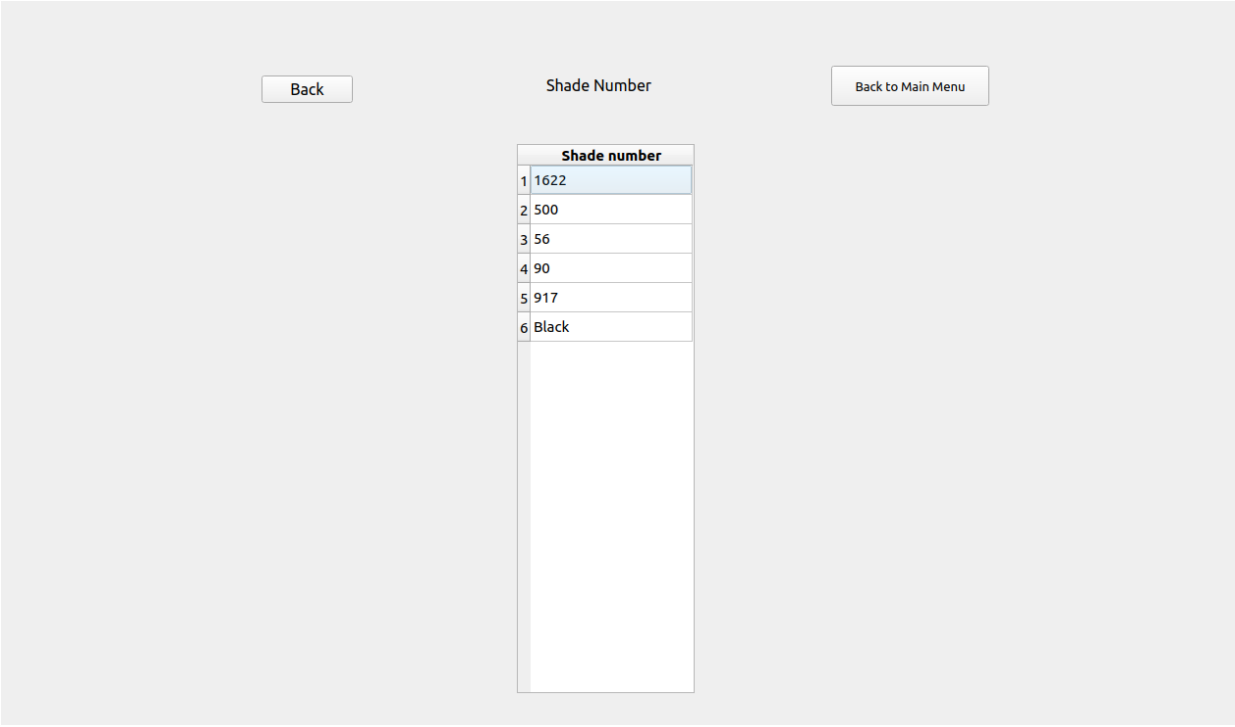




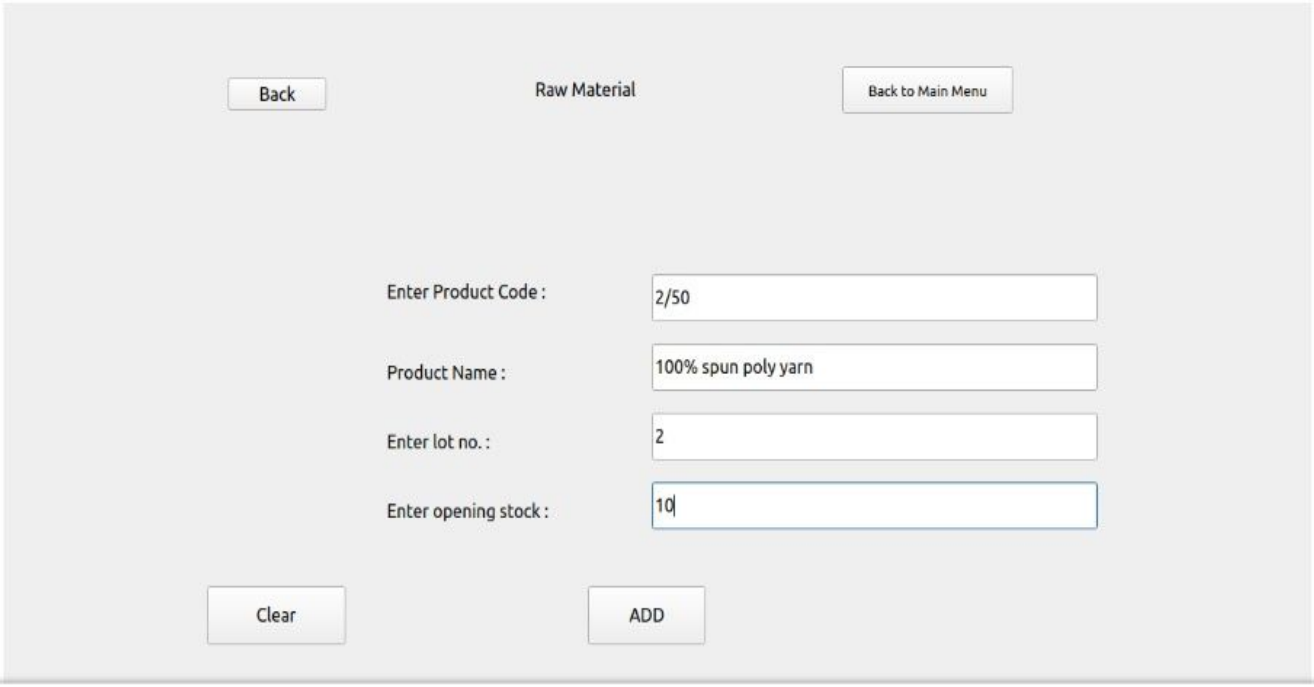
View new Shade number:

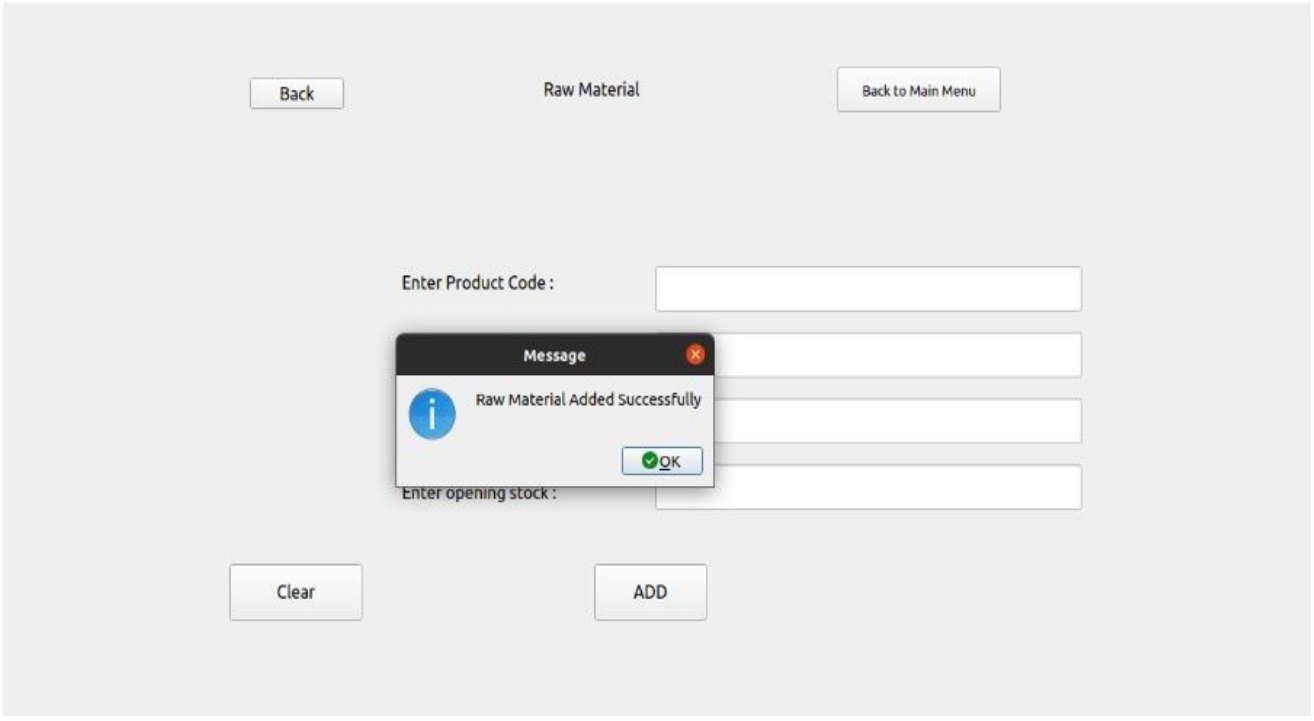


View all shade numbers :

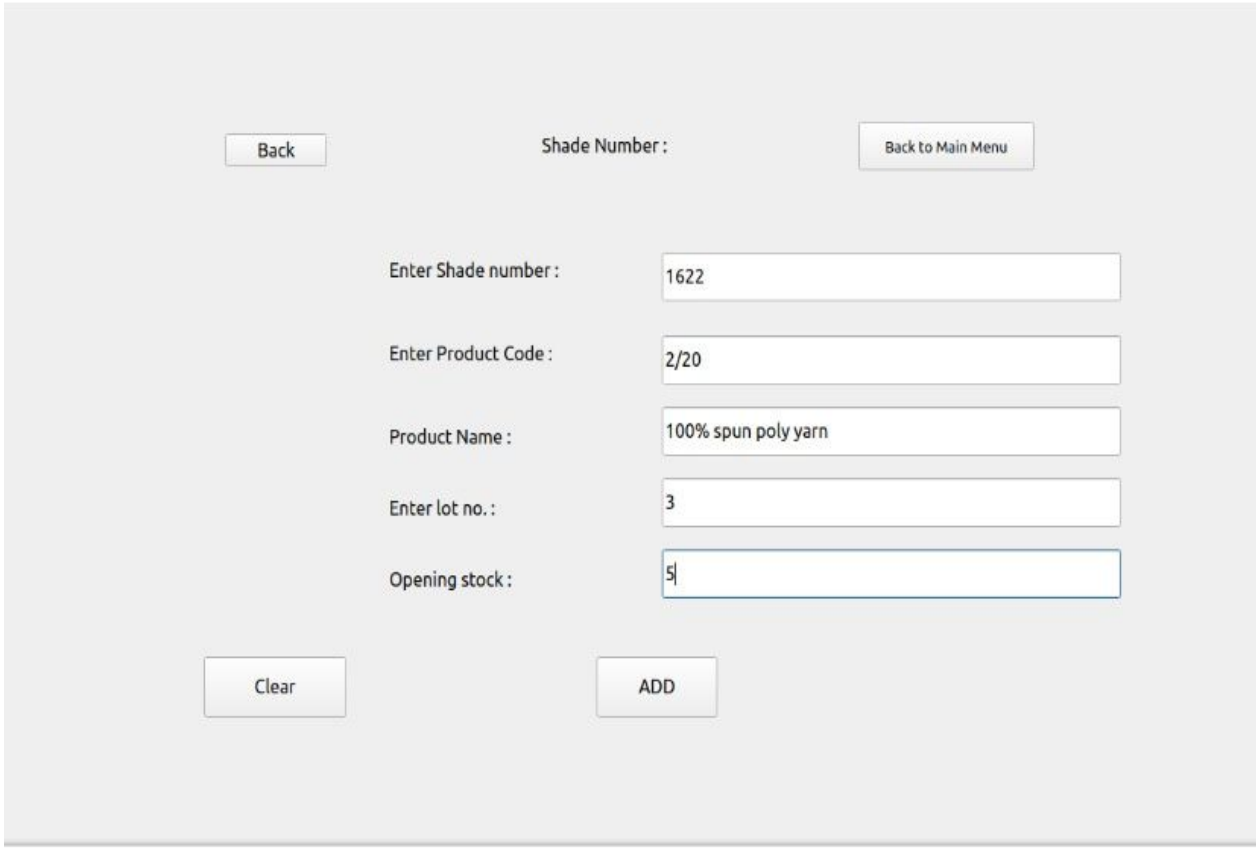


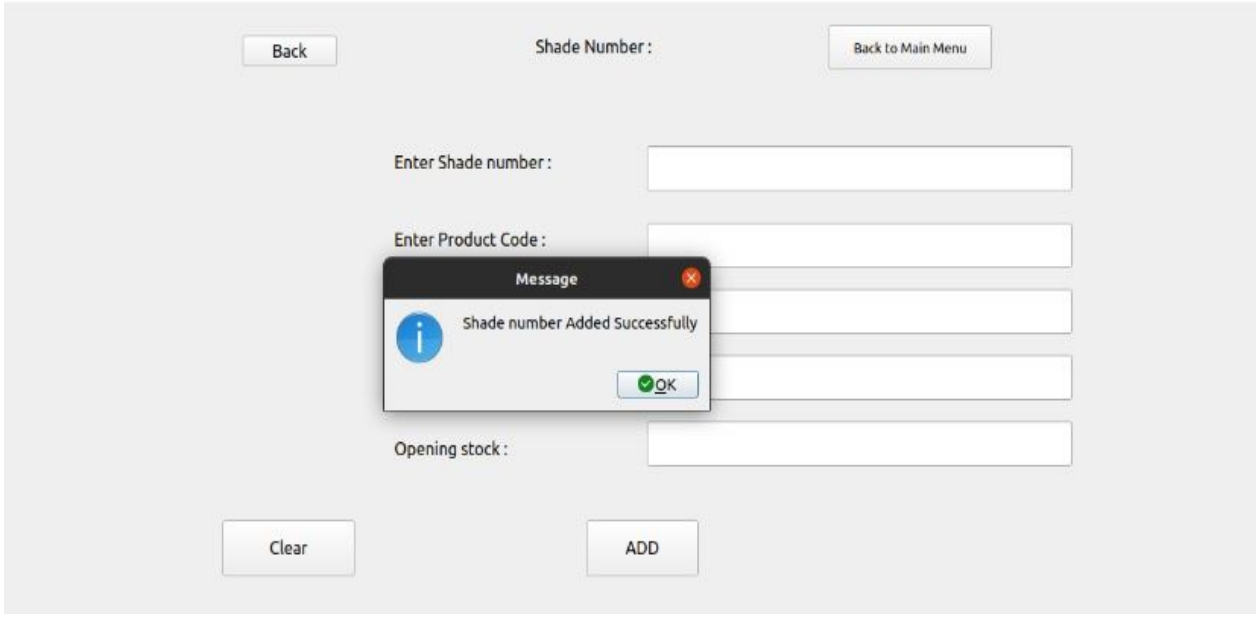
Add raw material opening stock:



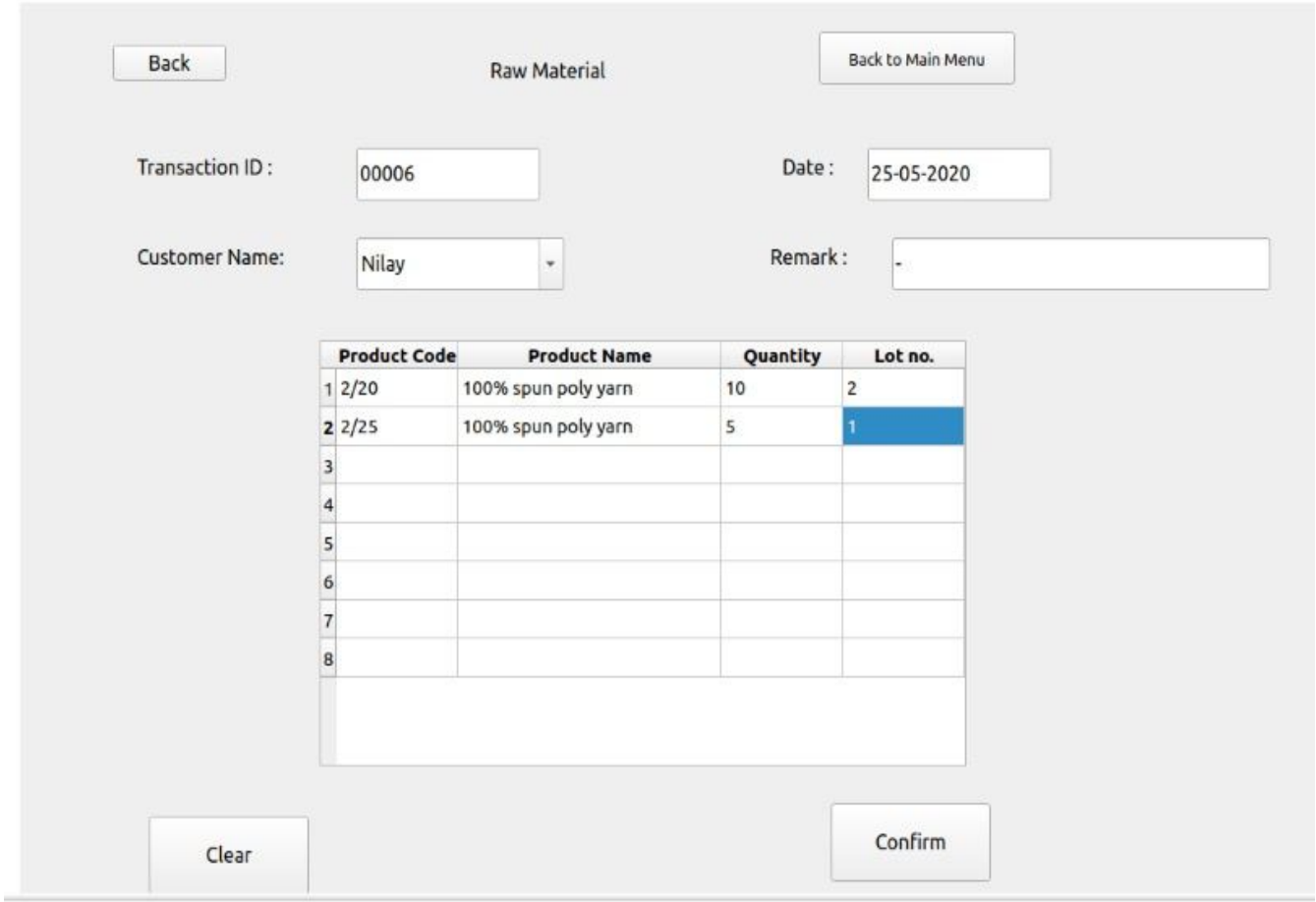


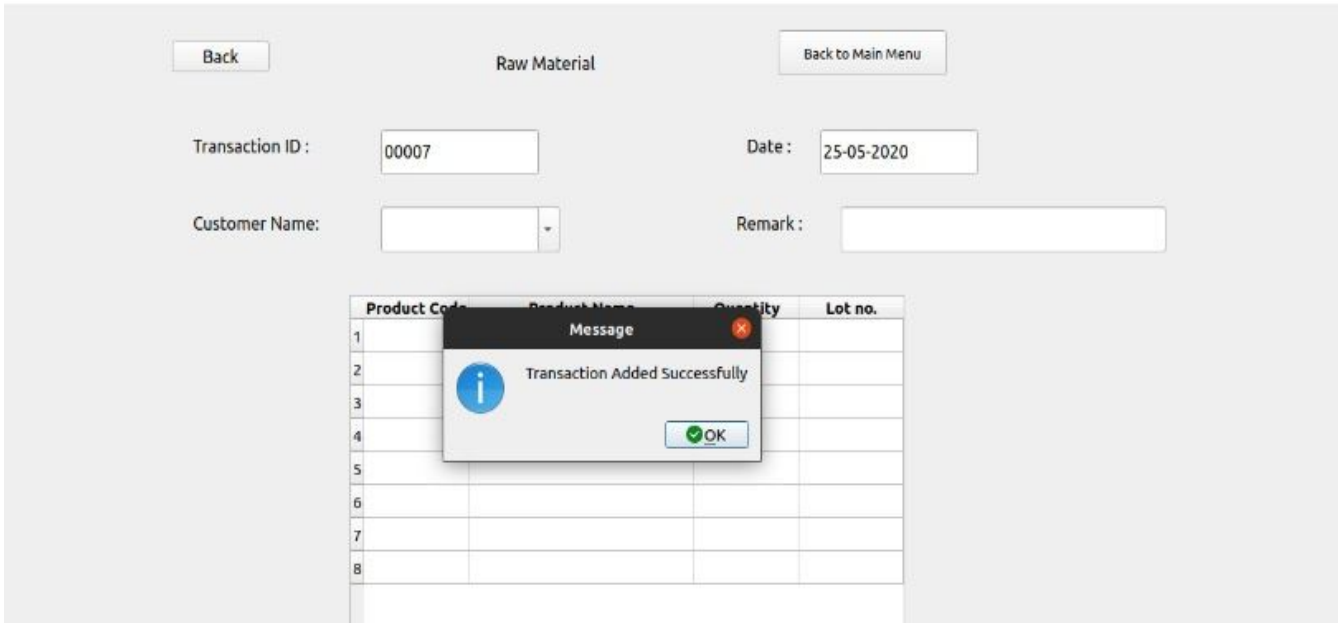
Add shade number opening stock:





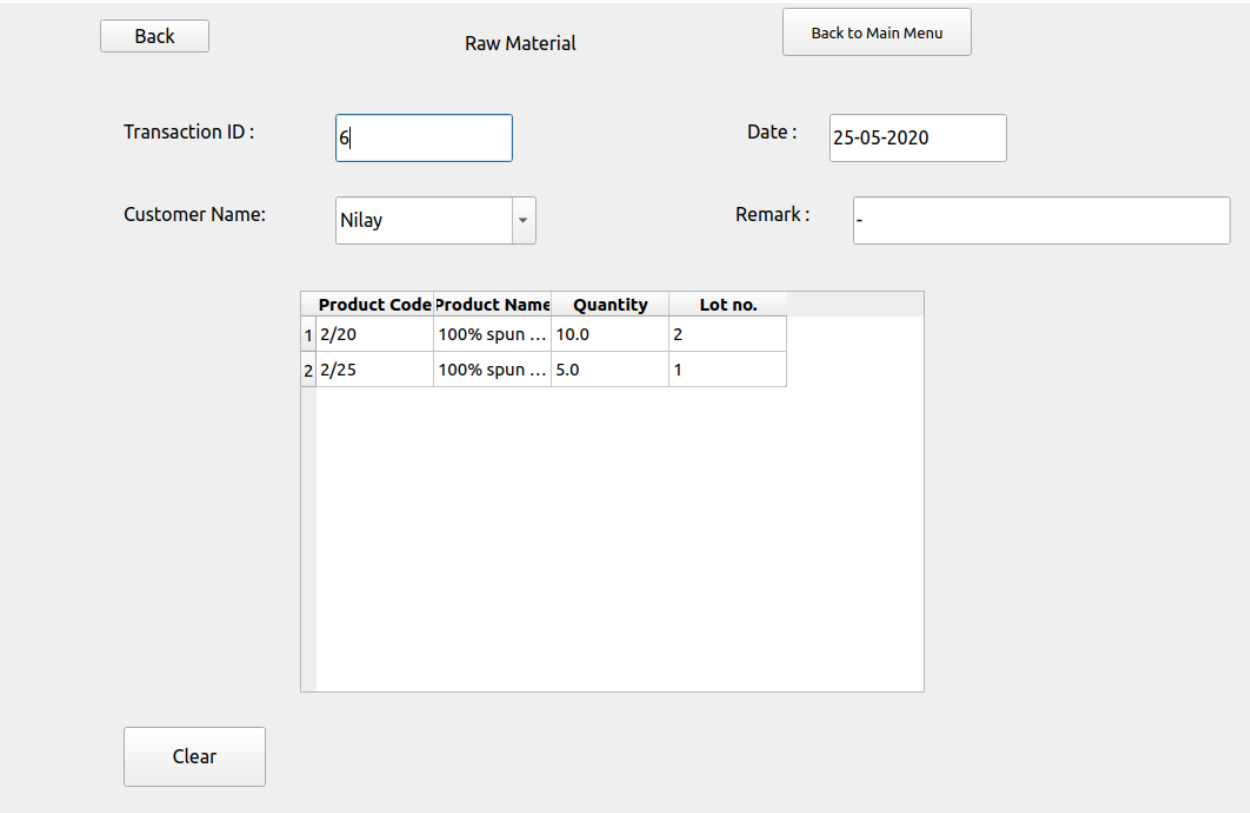
Add raw material transaction:



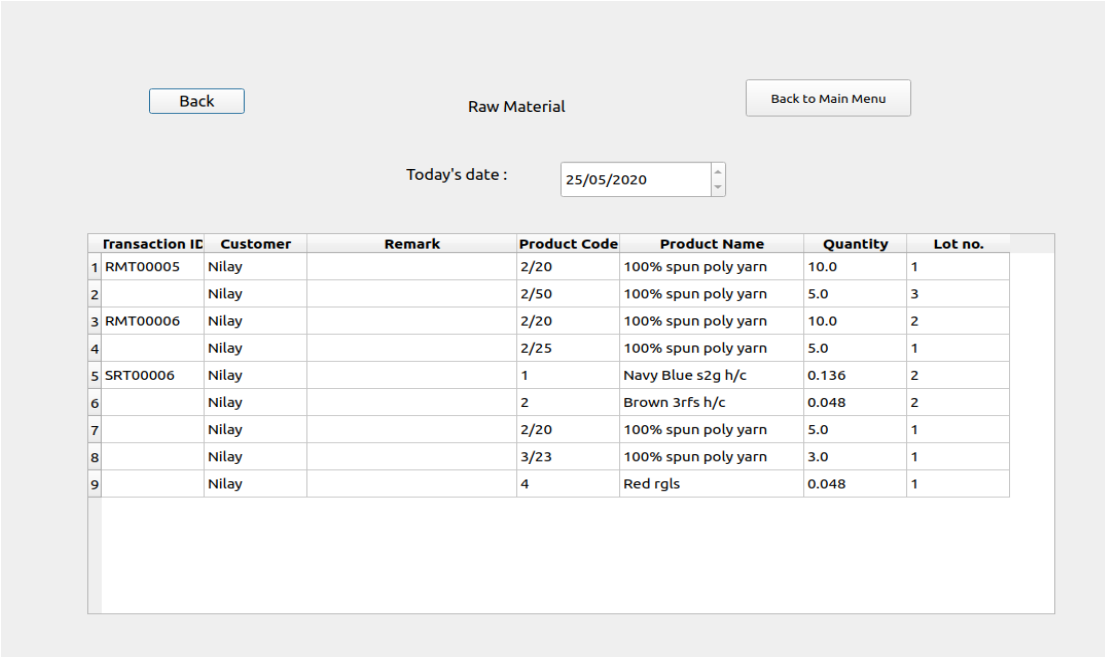


View raw material by:

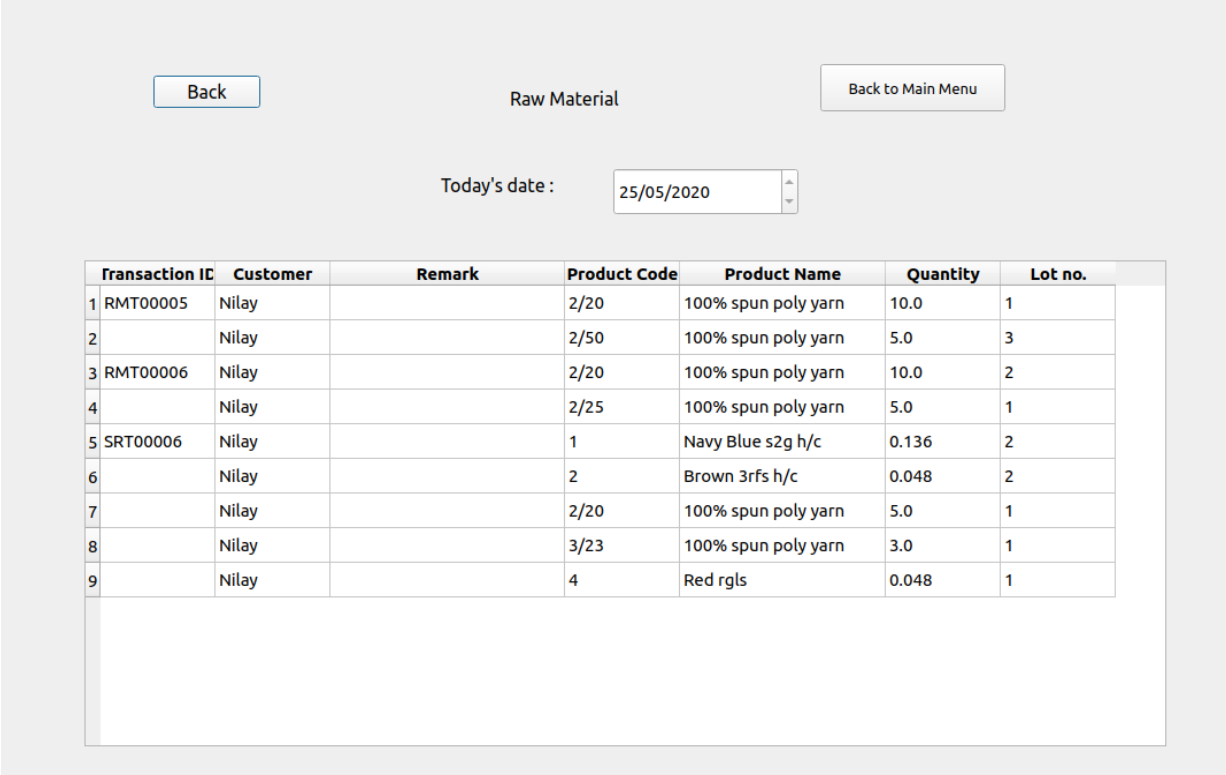
1. **Transaction_ID**



2. **Today**

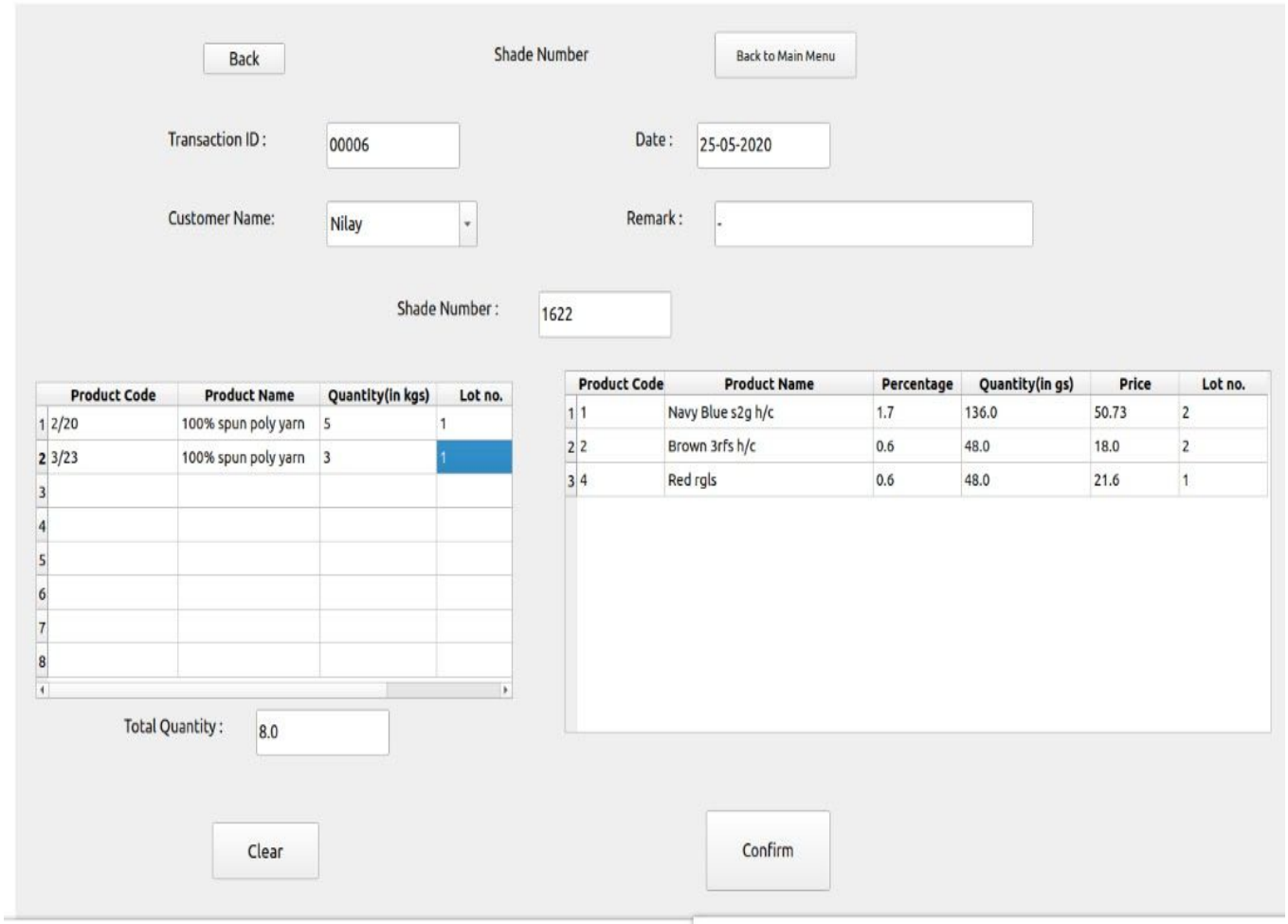


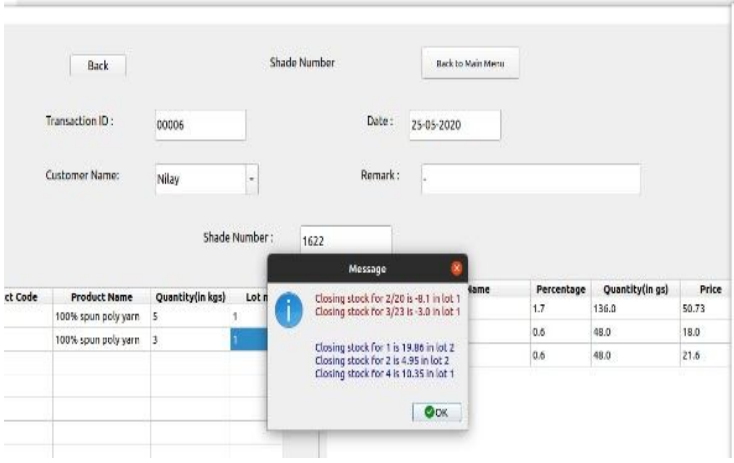
3. Custom Date

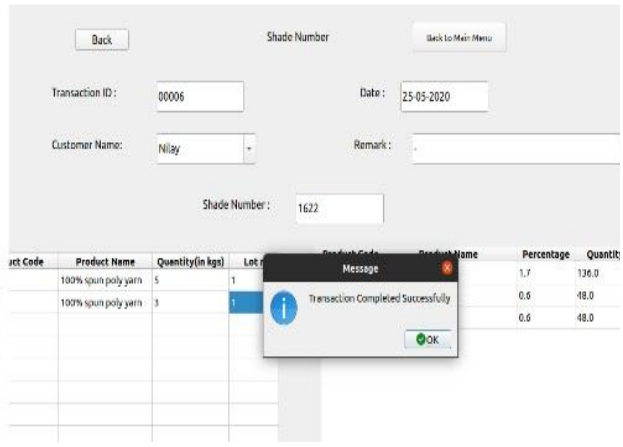


	Transaction ID	Customer	Remark	Product Code	Product Name	Quantity	Lot no.
1	RMT00005	Nilay		2/20	100% spun poly yarn	10.0	1
2		Nilay		2/50	100% spun poly yarn	5.0	3
3	RMT00006	Nilay		2/20	100% spun poly yarn	10.0	2
4		Nilay		2/25	100% spun poly yarn	5.0	1
5	SRT00006	Nilay		1	Navy Blue s2g h/c	0.136	2
6		Nilay		2	Brown 3rfs h/c	0.048	2
7		Nilay		2/20	100% spun poly yarn	5.0	1
8		Nilay		3/23	100% spun poly yarn	3.0	1
9		Nilay		4	Red rgls	0.048	1

Add shade number transaction:

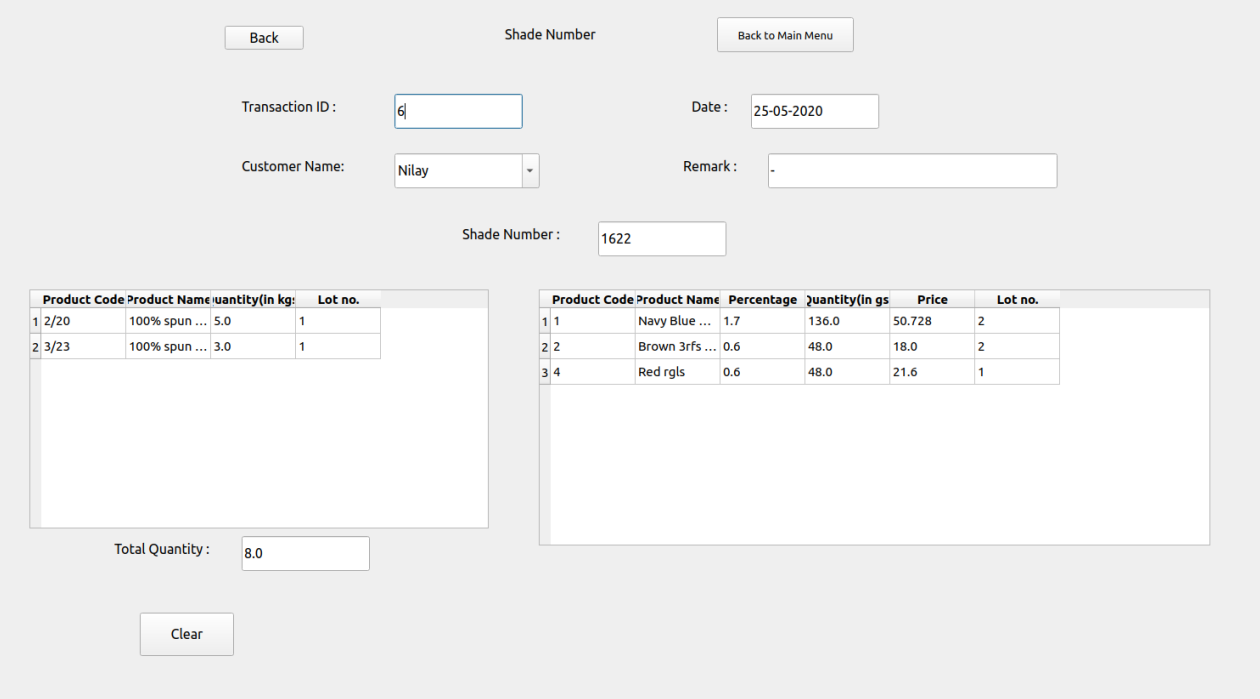




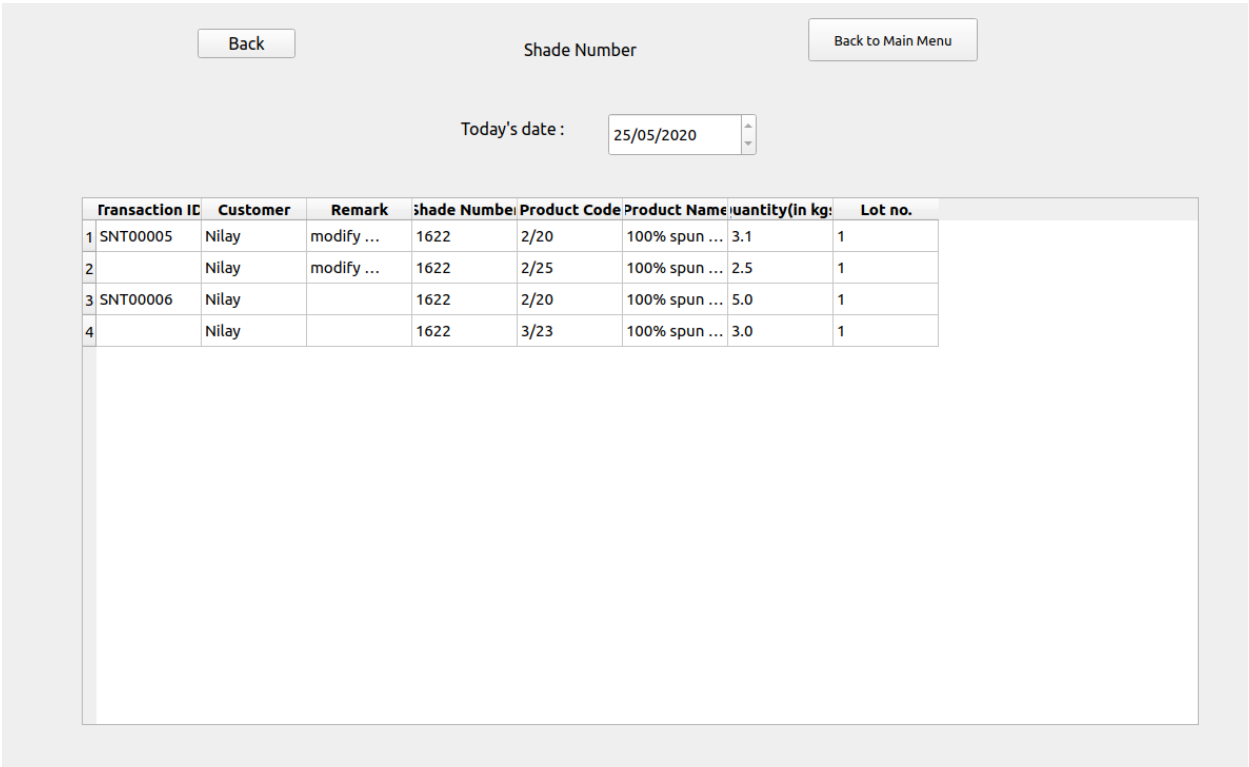


View Shade number transaction by:

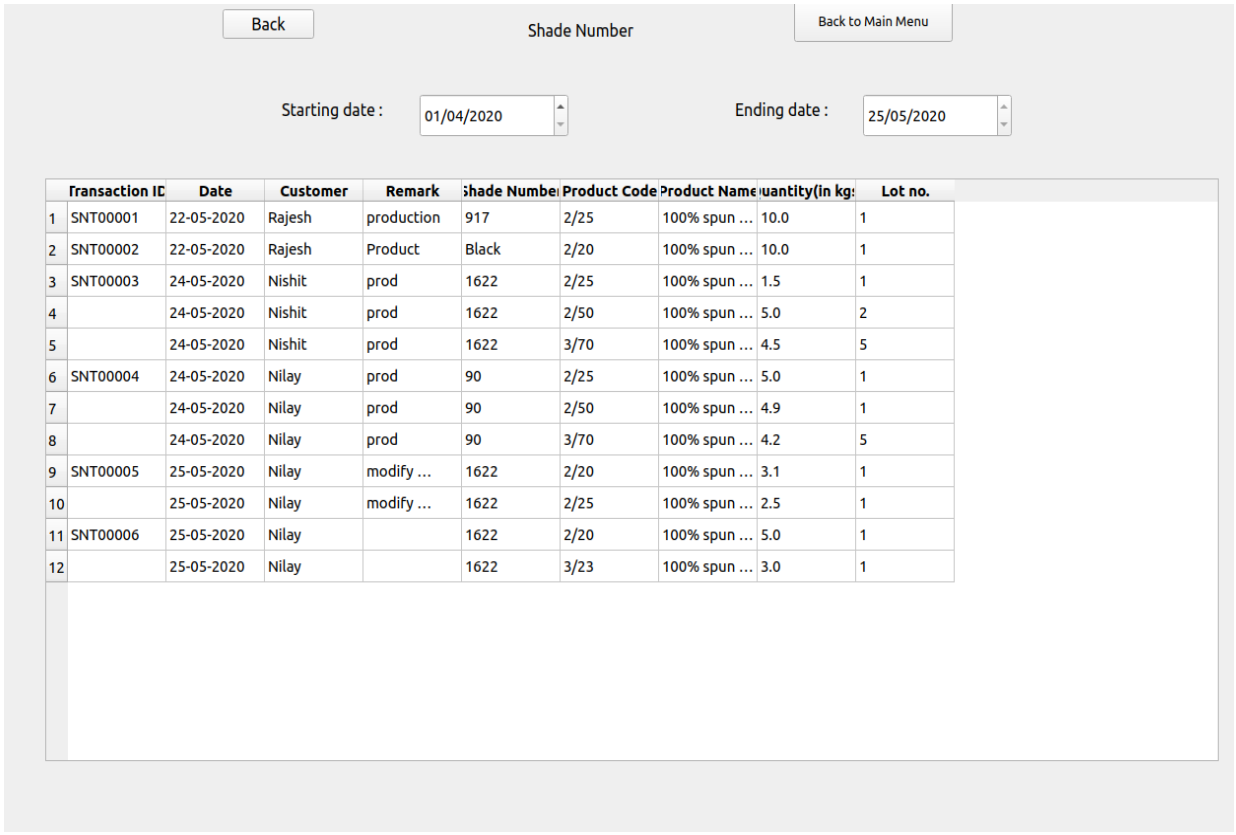
1. **Transaction_ID**



2. **Today**

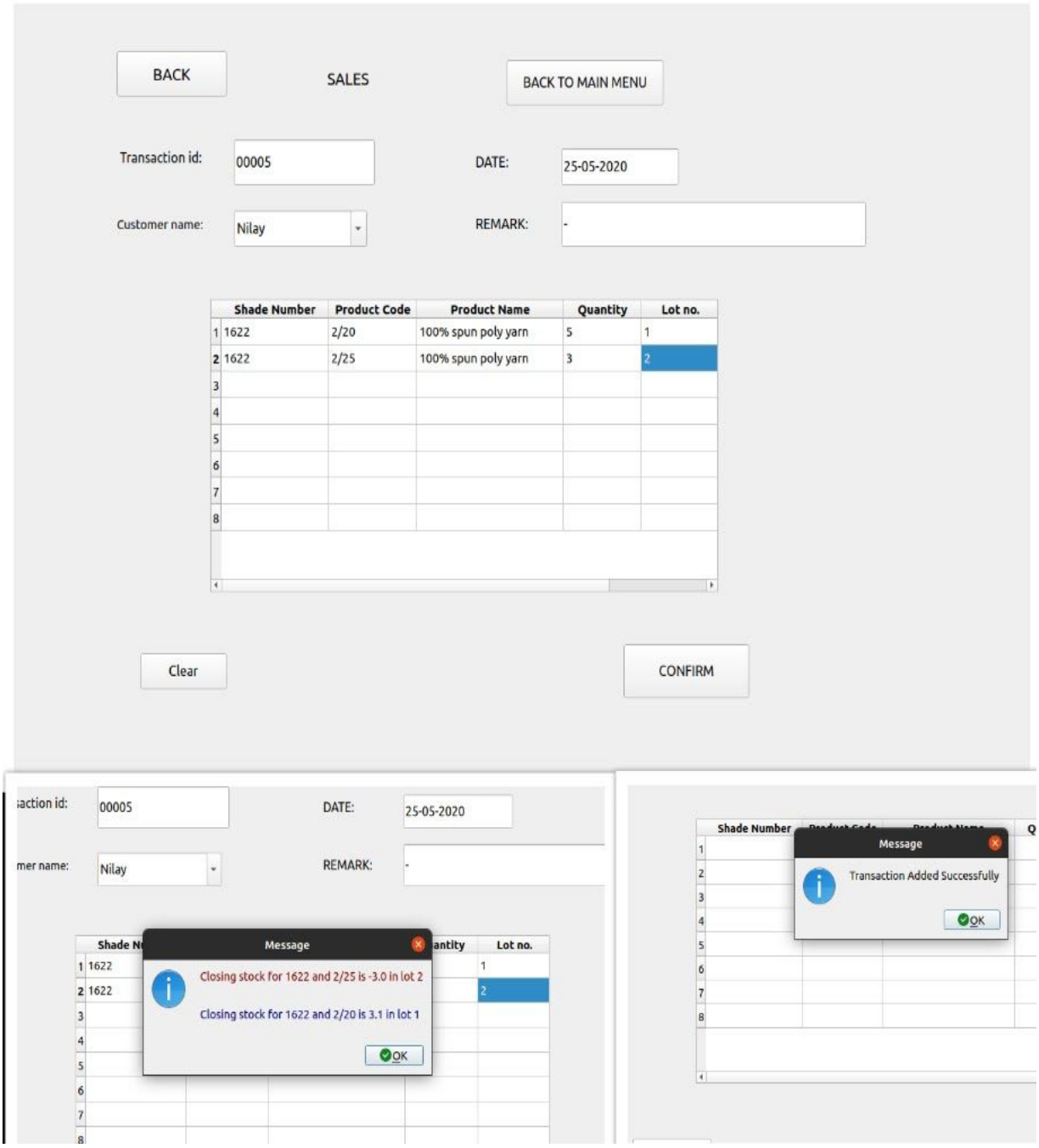


3. Custom Date



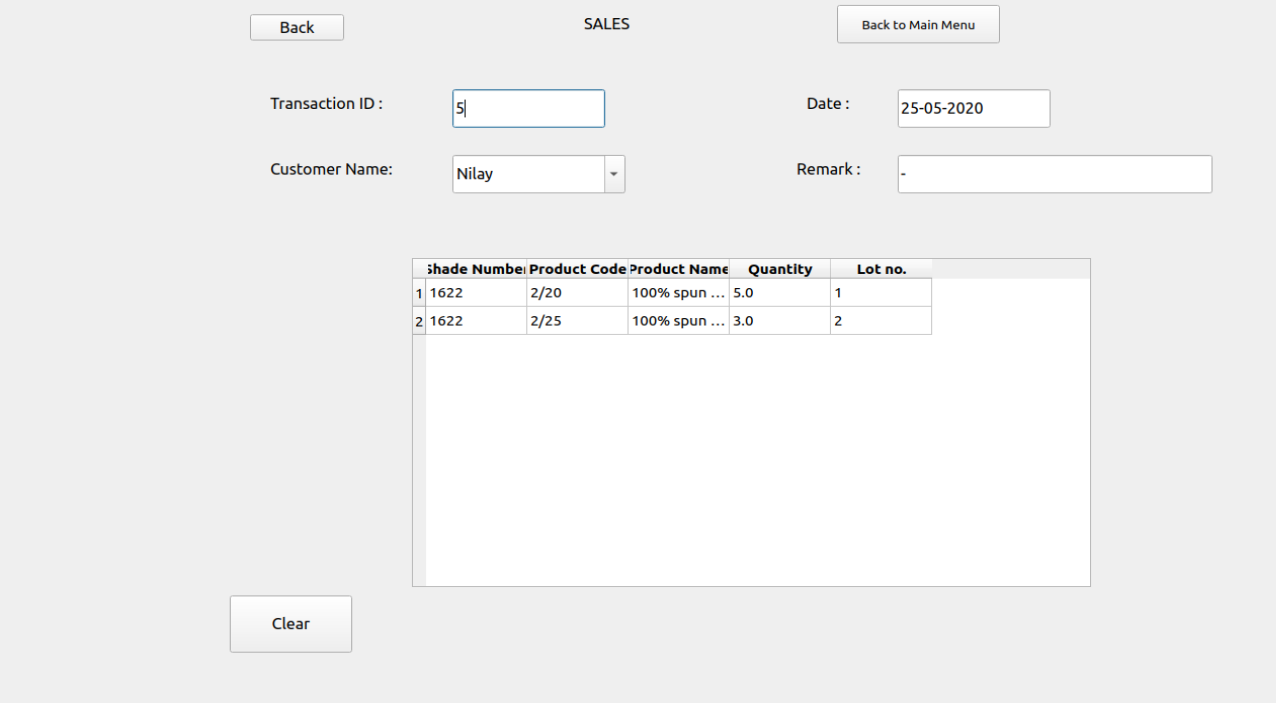
	Transaction ID	Date	Customer	Remark	Shade Number	Product Code	Product Name	Quantity(in kg	Lot no.
1	SNT00001	22-05-2020	Rajesh	production	917	2/25	100% spun ...	10.0	1
2	SNT00002	22-05-2020	Rajesh	Product	Black	2/20	100% spun ...	10.0	1
3	SNT00003	24-05-2020	Nishit	prod	1622	2/25	100% spun ...	1.5	1
4		24-05-2020	Nishit	prod	1622	2/50	100% spun ...	5.0	2
5		24-05-2020	Nishit	prod	1622	3/70	100% spun ...	4.5	5
6	SNT00004	24-05-2020	Nilay	prod	90	2/25	100% spun ...	5.0	1
7		24-05-2020	Nilay	prod	90	2/50	100% spun ...	4.9	1
8		24-05-2020	Nilay	prod	90	3/70	100% spun ...	4.2	5
9	SNT00005	25-05-2020	Nilay	modify ...	1622	2/20	100% spun ...	3.1	1
10		25-05-2020	Nilay	modify ...	1622	2/25	100% spun ...	2.5	1
11	SNT00006	25-05-2020	Nilay		1622	2/20	100% spun ...	5.0	1
12		25-05-2020	Nilay		1622	3/23	100% spun ...	3.0	1

Add sales details:

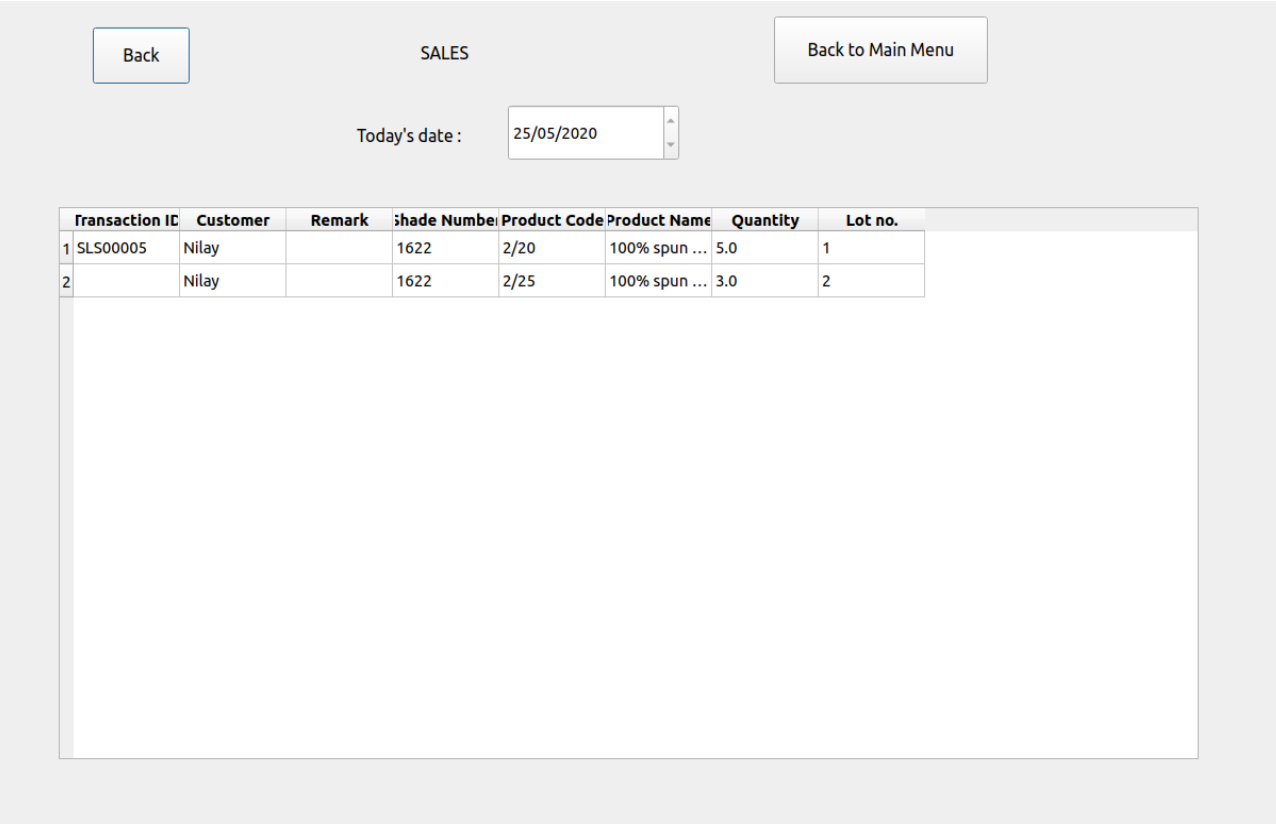


View sales details by:

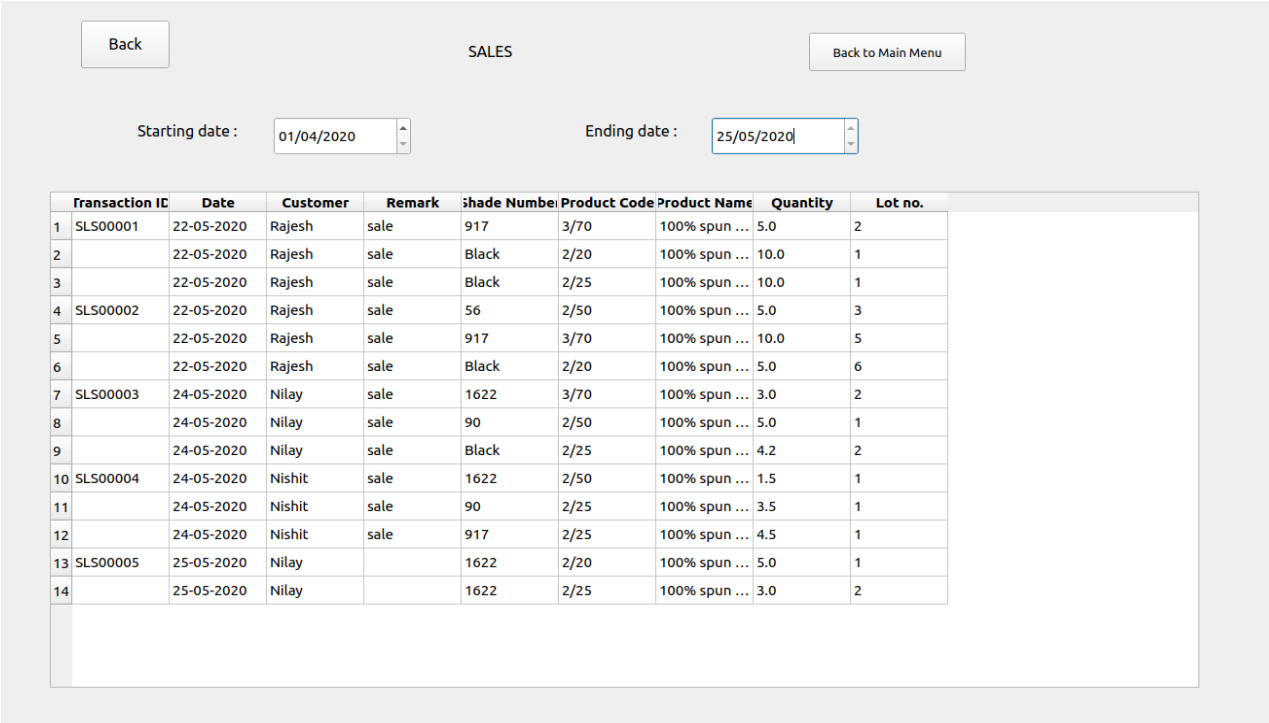
1. **Transaction ID**



2. **Today**

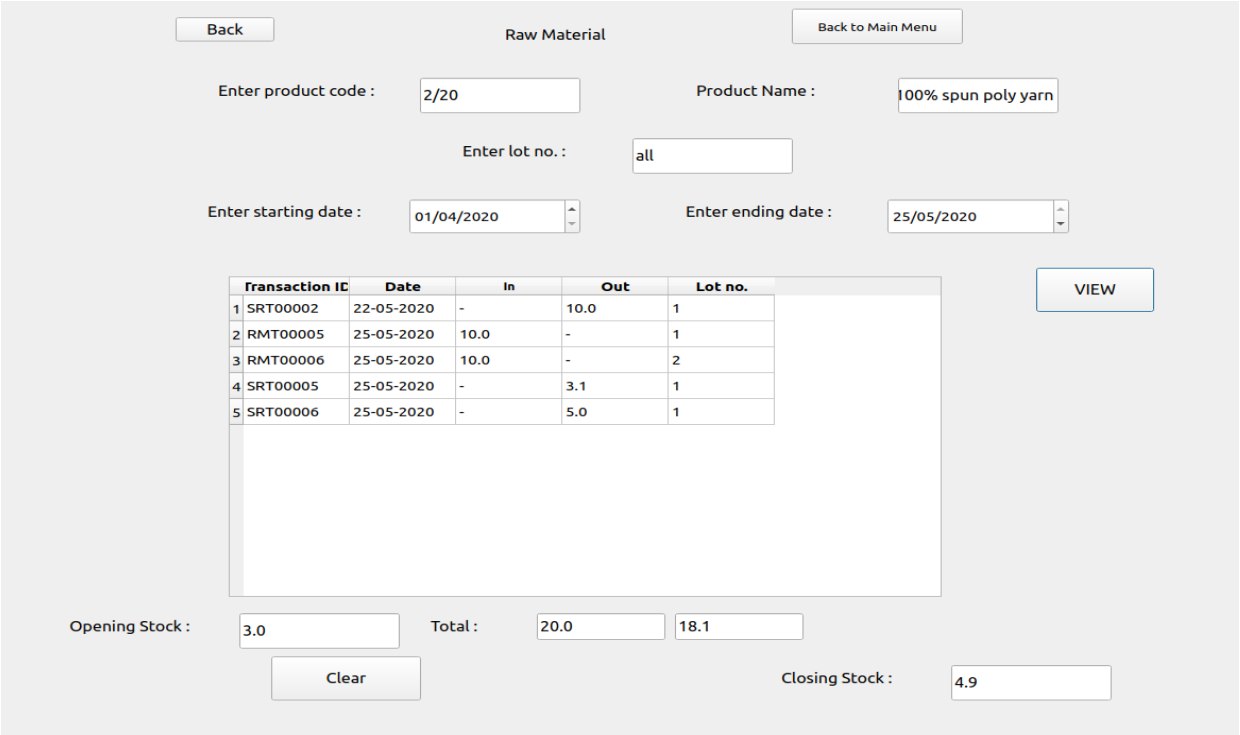


3. Custom dates



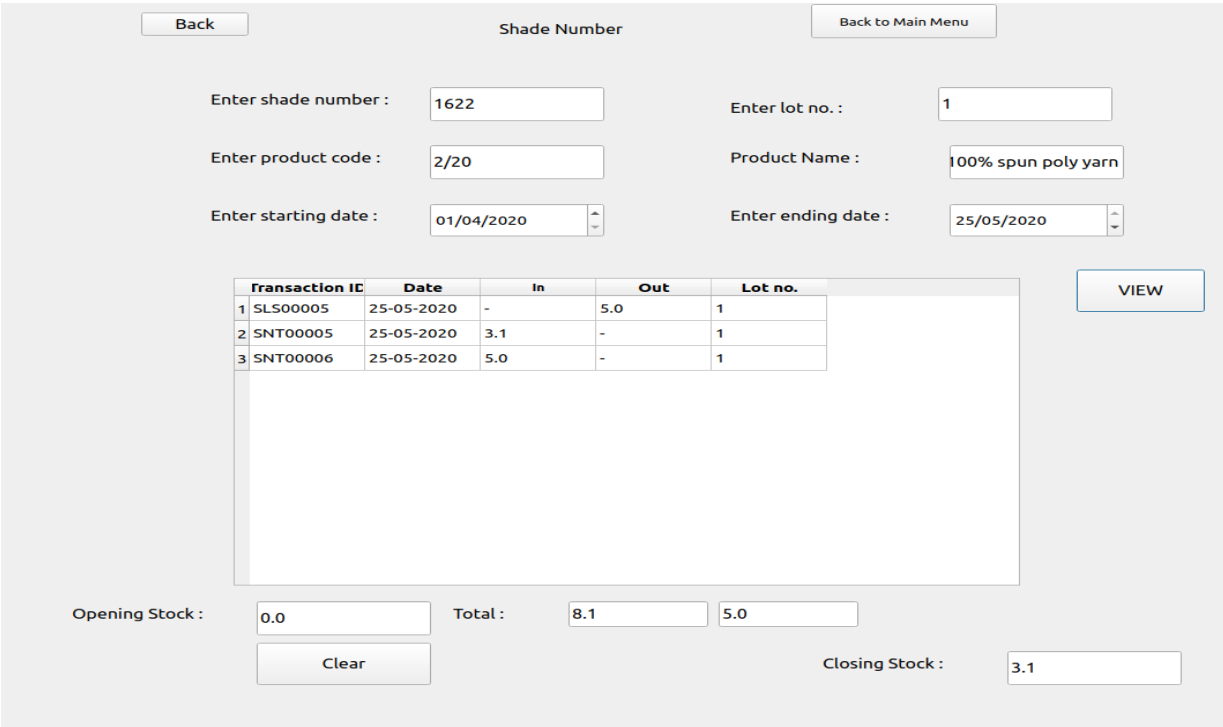
	Transaction ID	Date	Customer	Remark	Shade Numbe	Product Code	Product Name	Quantity	Lot no.
1	SLS00001	22-05-2020	Rajesh	sale	917	3/70	100% spun ...	5.0	2
2		22-05-2020	Rajesh	sale	Black	2/20	100% spun ...	10.0	1
3		22-05-2020	Rajesh	sale	Black	2/25	100% spun ...	10.0	1
4	SLS00002	22-05-2020	Rajesh	sale	56	2/50	100% spun ...	5.0	3
5		22-05-2020	Rajesh	sale	917	3/70	100% spun ...	10.0	5
6		22-05-2020	Rajesh	sale	Black	2/20	100% spun ...	5.0	6
7	SLS00003	24-05-2020	Nilay	sale	1622	3/70	100% spun ...	3.0	2
8		24-05-2020	Nilay	sale	90	2/50	100% spun ...	5.0	1
9		24-05-2020	Nilay	sale	Black	2/25	100% spun ...	4.2	2
10	SLS00004	24-05-2020	Nishit	sale	1622	2/50	100% spun ...	1.5	1
11		24-05-2020	Nishit	sale	90	2/25	100% spun ...	3.5	1
12		24-05-2020	Nishit	sale	917	2/25	100% spun ...	4.5	1
13	SLS00005	25-05-2020	Nilay		1622	2/20	100% spun ...	5.0	1
14		25-05-2020	Nilay		1622	2/25	100% spun ...	3.0	2

View raw material stock details:

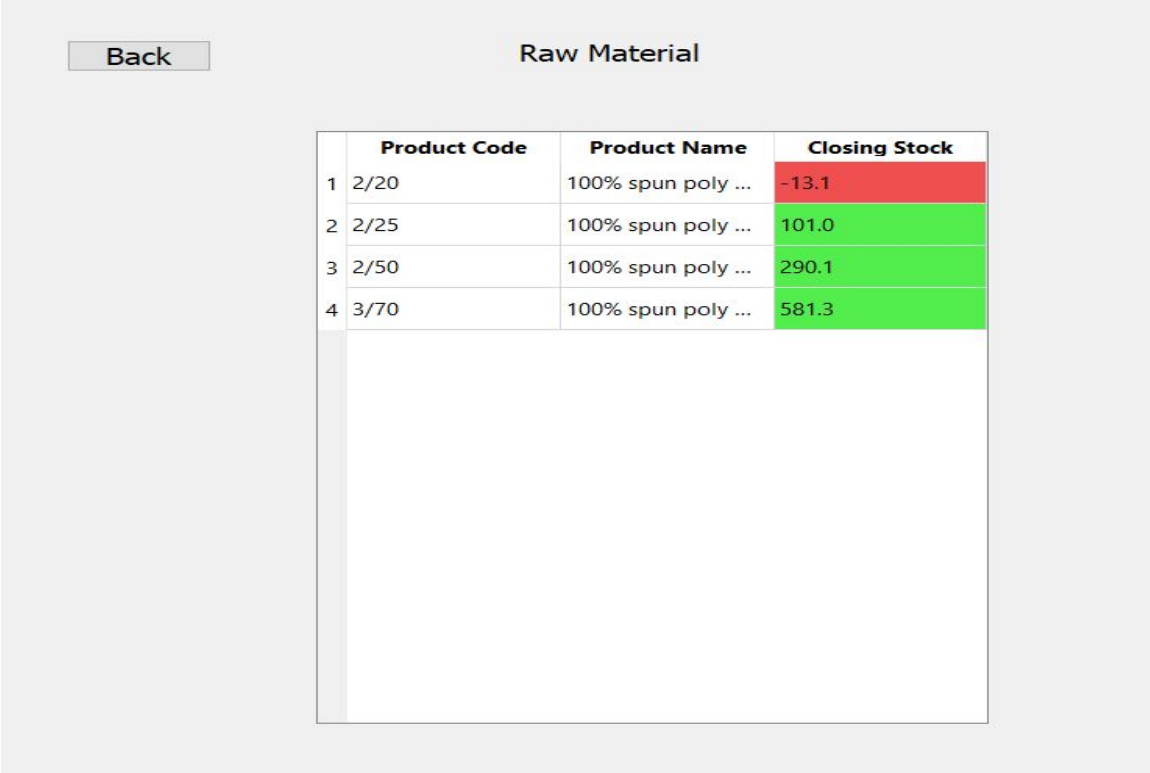


	Transaction ID	Date	In	Out	Lot no.
1	SRT00002	22-05-2020	-	10.0	1
2	RMT00005	25-05-2020	10.0	-	1
3	RMT00006	25-05-2020	10.0	-	2
4	SRT00005	25-05-2020	-	3.1	1
5	SRT00006	25-05-2020	-	5.0	1

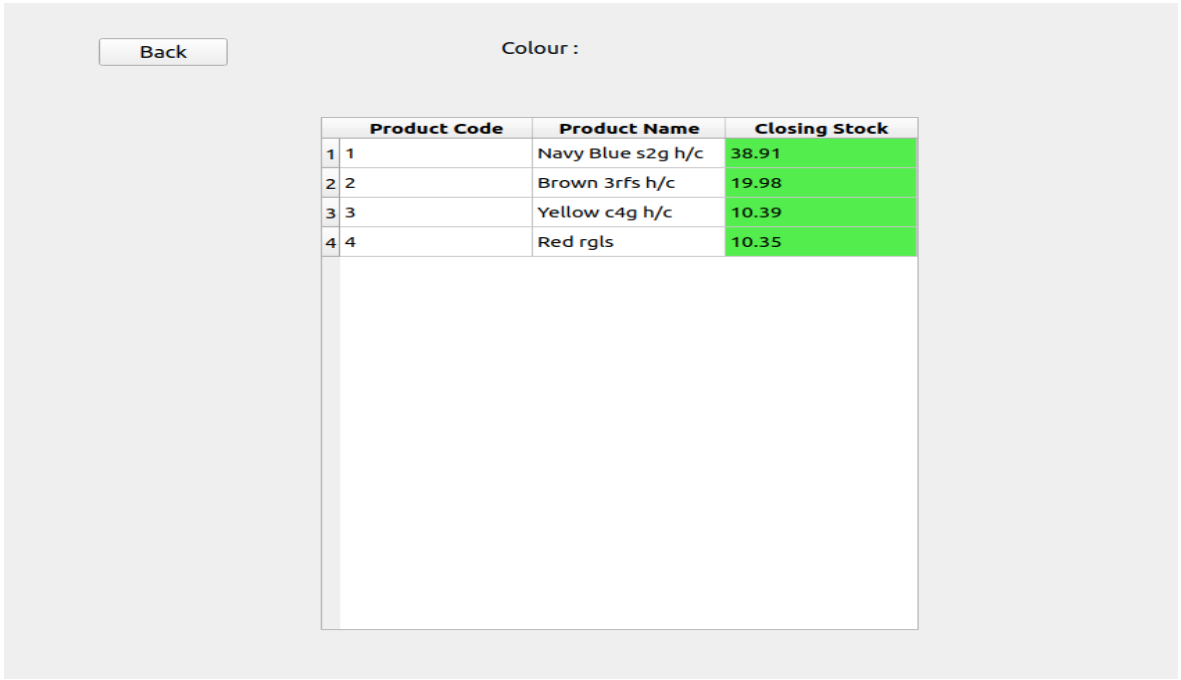
View shade number stock details:



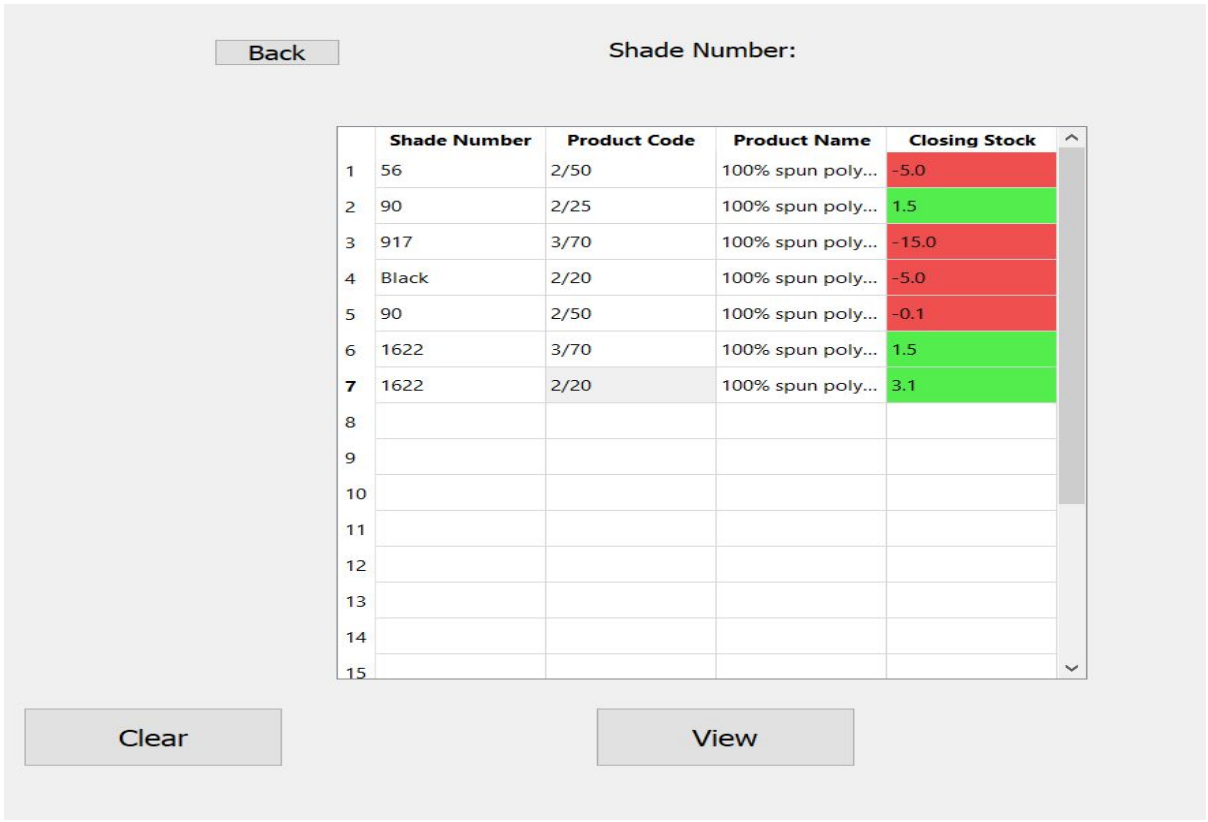
View raw material closing stock:



View colour closing stock:



View shade number closing stock:



Conclusion:

We have successfully designed and developed a desktop application for inventory management for a thread dyeing factory using Python.The GUI of the application is developed with the PyQt module of python and database was created with the help of sqlite module.

Future Scope:

The future work for inventory management will focus on:

1. Adding printing transactions facility along with billing facility.
2. Adding more operations for maintaining the customer and supplier base. Which means providing an interface to save the details of the various suppliers and consumers and so can also track the transactions according to given suppliers and consumers.