JARTE
Journal of Applied Research in Technology & Engineering

4(1): 13-23, 2023
https://doi.org/10.4995/jarte.2023.18776

Received: November 14, 2022; Accepted: January 08, 2023; Published: January 31, 2023

AUTOMATION OF MATERIAL COST COMPARATIVE ANALYSIS REPORT USING VBA EXCEL: A CASE OF FOOTWEAR COMPANY OF LAHORE

Muhammad Ahmed Kalwar [a,*], Asif Nawaz Wassan [b1], Zuhaib Phul [c], Muzamil Hussain Wadho [d], Tanveer Sarwar Malik [e], Muhammad Ali Khan [b2]

[a]Production, Shafi Private Limited, Lahore, Punjab, Pakistan.

[b]Industrial Engineering & Management, Faculty of Engineering, Mehran UET, Jamshoro, Sindh, Pakistan.

[c]Computer Science, Sindh Madressa-tul-Islam University, City Campus, Karachi, Sindh, Pakistan.

[d]Electrical Engineering, Faculty of Engineering, BBSUTSD, Khairpur Mirs, Sindh, Pakistan.

[e]Starlet Innovations Limited, Mandiali, Sheikhupura, Punjab, Pakistan.

[a] ahmedkalwar22@gmail.com, [b1] asif.wassan123@gmail.com, [c] zuhaibphul53@gmail.com, [d] muzzamilhussian@bbsutsd.edu.pk, [b2] shakeel.shaikh@faculty.muet.edu.pk, [e] malik.Jt786@gmail.com, [b] muhammad.nagar@faculty.muet.edu.pk

Abstract:

Small and medium-sized firms in developing countries perform the majority of all reporting tasks in Microsoft excel. The material cost comparative analysis (MCCA) report took the user 58.51 minutes to complete and there was also a higher likelihood of calculation error. The report was created in Microsoft excel by the Lahore-based ABC footwear business. This used to take a lot of time to produce and there was a higher chance that the report would contain errors. This is because Microsoft Excel visual basic for applications was used to automate the report. This report's manual preparation method was compiled by the company's planning and costing division. Each activity was listed, followed by the completion of a time study. VBA macros were built in Excel to automate all manual report-related tasks and reduce the possibility of human error. Conditional statements were applied to help automated decision-making. To enable command buttons to run macros, one user form was created. There were various worksheets included in the MCA template. The automated template was found to consume only 16.26 minutes and it was quicker than the current technique.

Keywords: material cost; footwear; shoe costing; cost comparison.

Cite as: Kalwar, M.A., Wassan, A.N., Phul, Z, Wadho, M.H., Malik, T.S., Khan, M.A. (2023). Automation of material cost comparative analysis report using VBA excel: a case of footwear company of Lahore. *J Appl Res Eng Technol & Engineering, 4*(1), 13-23. https://doi.org/10.4995/jarte.2023.18776

1. Introduction

Information technology is crucial for the calculation, processing, and retrieval of data. The primary tool used by corporations to process data and make decisions is a management information system (Karim, 2011). The goal of data management and organization is to create useful information from organized sets of data. In small and medium-sized businesses, spreadsheets are widely used for departmental reporting, covering, costing and planning, etc. Data entry, analysis, graphing, and storing are just a few of rem operations that may be done with spreadsheets. The use of various spreadsheet applications is required for each of the aforementioned tasks (Broman et al., 2017). It is challenging to create a trustworthy spreadsheet as the practitioner's experiences demonstrated (Dunn, 2009). The most popular programming language in use today is the spreadsheet (Fisher et al., 2002). Both consumers and businesses utilize spreadsheets for several functions, including doing quick computations (Abraham et al., 2008). To carry out these tasks, large companies require a complicated information system, but if the data is little and not variable, a similar system can be built using VBA and MS Excel. The visual basic for applications (VBA) Feature of Excel allows users to save formulas and operations as "macros", or lines of computer code (Perry, 2012). Systems created in Excel using VB are either free or inexpensive and they offer excellent performance for carrying out intricate studies. In addition, analysis performed by those with less experience can yield results more quickly (Blayney & Sun, 2019). Excel macros can be utilized after they have been programmed by simply pressing the button. Users can automate all spreadsheet operations and create user-defined routines by using VBA (Abraham et al., 2008). Excel formulas are significantly different from VBA, and the programming environment offered (visual basic editor) is also different (Abraham et al., 2008). Intelligent massive data analytics has been improved by Blayney and Sun (Blayney & Sun, 2019). Balson offered instruction in his study (Balson, 2012) on how to use Microsoft Excel to produce a condensed form of user-defined spreadsheet functions (USDFs). All pertinent USDFs should be programmed in a module with a dynamic input range for a regular user to later use the functions in the spreadsheet. Alexei Botchkarev concluded that excel was a useful tool for MC simulation after evaluating the applicability of Monte Carlo (MC) simulation in the Excel VBA (Botchkarev, 2015). According to research by Ajinkya et al., (Ajinkya et al., 2017), Microsoft Excel may be used to estimate

*Corresponding author: M.A. Kalwar, ahmedkalwar22@gmail.com the scope and expense of construction work including PPC, excavation, plastering, RCC and brickwork. To identify suspicious measuring points and add missing points, Raza and Gulwani employed VBA in Excel (Raza & Gulwani, 2017).

Very few research articles on spreadsheet automation connected to industrial planning reporting, such as budgeting, pricing, scheduling, etc. have been published except for fundamental materials. This study provides an answer for organizations by showing how conditional decision-making may be done automatically with VBA. It also demonstrated how flexible VBA can complete even the most straightforward tasks. Industries require the principles of visual designs for applications, particularly when reporting efficiency and accuracy are at issue. Because of the importance of this idea, every operation was automated to give users accurate reports as soon as possible. This study describes how the planning and costing departments of the ABC footwear company in Lahore, Pakistan, prepare the material cost comparison analysis report.

2. Literature Review

The mutual operation in the report was avoided while maintaining the costing calculations under the costing department's criteria for the ABC corporation by employing visual basics for applications. This context included a study on automation and optimization using VBA. According to Zainal Abidin et al. (2015), you may use VBA in Microsoft Excel to calculate the water quickly index (WQI) and air pollution index. Direct indices determination was built into the program. From the necessary formulas, the code was generated. Due to the way the descriptions were written the measured index clauses that were provided may automatically be displayed next to each index (Abidin et al., 2015). Ahmadi et al., (2018) employed VBA in Excel to perform research for the application of the CTR dairy model. The dynamic simulation model for grazing lactating dairy cows, or CTR dairy, is used to project milk output and profits based on several variables, such as ruminal digestion and nutrient absorption under intermittent feeding schedules. The CTR dairy mode was translated into excel VBA to convert the input into output because smart software failed and there weren't enough clients, this study provided access to a variety of dairy producers, dairy nutrition specialists, academics, and extension advisers (Ahmadi et al., 2018). Using a new application that Belchior Junior et al., (2011) created using excel VBA, the data from the reactor excursion and leak analysis program 5 (RELAP5) is a highly helpful tool for accelerating output data analysis (Belchior Junior et al., 2011). VBA excel programming was utilized by Rushit Hila (2009) to create the application for automatically arranging data and locating outliers in the data. Before importing data into the database, which was MS Access, several program tasks were automated (Hila, 2009). A new technique for the automatic creation of personal planning reports using VBA excel was developed by Cirujani and Zhu in 2013. The new procedure was used to construct the task and schedules of the engineers engaged in different projects. A compilation, review, and organization of the data gathering took place. This could allow for the creation of plans for the engineers and the collection of information regarding their involvement in various projects. At a costing organization with more than 100 employees, the technique was tested. The generation of personal planning reports automatically has the potential to save a lot of time and money (Cirujano & Zhu, 2013). Users can click a web icon to transfer image data from a data set to a worksheet by using an application created by Sato and Yokoyama (2001) that employs VBA in Excel (Sato & Yokoyama, 2001). With the use of visual basic for applications in excel. Lessa et al., (2016) automated a useful mathematical model for the logic program and the calculation of packing.

The graphic layouts were developed to match how the packets are filled (Lessa et al., 2016). The first version of the instrument's communication was made by H. Evensen (2014) in excel using VBA 2014 (Evensen, 2014). An automated report was created by Donald E. Blattner and Valrico Florida in 2007 utilizing VBA in an MS project. Using the help dialogue box that appeared on the screen, the user can choose to filter, format, and sort the report using the newly developed system (Patent No. US 2007/0055688 A1, 2007). To build mapping rules, Wettlaufer (2010) employed Excel VBA macros. A macro was assigned to each report. The second spreadsheet, often known as the expected values spreadsheet, included the expected values that were entered by the macros. Reports are produced and packaged as Winrar files after processing the patient follow-up on the mainline net server (Wettlaufer, 2010). VBA code was written by Norton and Tiwari (2013) to help engineering students comprehend the study of cutting-edge freezing technology (Norton & Tiwari, 2013). In their innovative method for investigating manufacturing processing, Bartoszewics and Wdowics (2019) paired the production planning module of SAP enterprise resource planning (ERP) with an excel spreadsheet and VBA for visualization and automation. The creation of a difficult analytical report was accelerated by redesigning and putting into practice a better, quicker, and more adaptable strategy for data transportation and analysis (operation times were reduced from 2 hours to 5 minutes) (Bartoszewicz & Wdowicz, 2019). Harahap and Azmi (2017) researched to create of a VBA excel application that could construct a small-scale rainwater conveyance system utilizing the reasoning offered in MSMA2 (Harahap & Azmi, 2017). Yan and Wan (2017) developed a program in Excel VBA to automatically calculate and produce the bill of material (BOM) for a transmission line. When making the whole steel BOM, the design and use of the template considerably increase efficiency and accuracy while lowering errors (Yan & Wan, 2017). For the footwear sector, Kalwar and Khan (2020) automated the procurement and purchase order report; before automation, the procurement report took 2076.751 seconds to complete. Furthermore, the automation cuts the processing time for a purchase order from 15-20 minutes to 2-3 seconds (Kalwar & Khan, 2020).

3. Research Methodology

The concerned employee at the ABC company's costing and planning division in Lahore, Pakistan gave details on the process and methods used to produce the material cost comparative analysis report. Each activity completed during the creation of the report was timed using a stopwatch. MS Excel was used to enter all the observations and calculate the average time for each activity. The average time for every activity was simultaneously shown in Microsoft Excel. The manual operations were automated using visual basic for applications in Microsoft Excel. Every operation, including building pivot tables, adding and deleting rows and columns, moving data between sheets, counting the number of rows in the worksheet, combining values from several columns that aren't empty, copying and pasting data, etc., was programmed using macros. All of the coded macros may be executed by clicking command buttons thanks to the creation of an alluring user interface in the shape of a user from the source code (compromising numerous command buttons). Pressing "Ctrl+q" causes the user form to appear on the screen.

4. Results

4.1. Existing Method

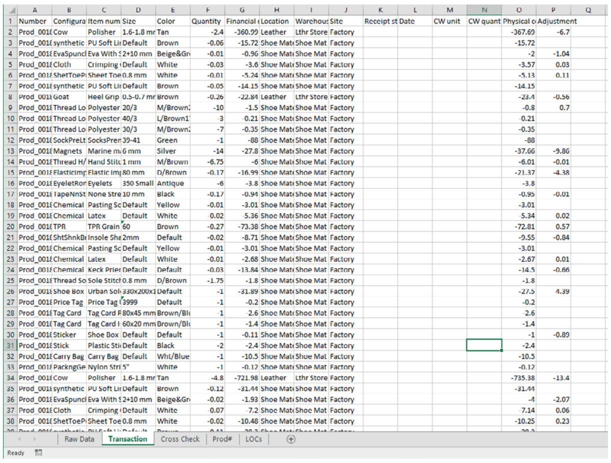
Various activities were supposed to be performed manually to make the material cost comparative analysis report. The description of activities was collected from the costing executive at the planning and costing department of the case company. A stopwatch was used to record the employee`s time that was consumed on each activity during the report preparation. Ten observations of time for each activity were collected and then the average time for each activity was computed in Microsoft Excel as given in Table 1. A look at Table 1 indicates that the manual operation of the material cost comparative analysis report used to take 58.51 minutes to be completed.

4.2. Suggested Method

The suggested method was based on the template that was programmed using visual basic for applications in Microsoft Excel.

4.2.1. Worksheets in the Automated Template

The 'Transaction' worksheet as given in Figure 1 was used to be pasted in the downloaded data from ERP and the data was processed by execution of the macros.



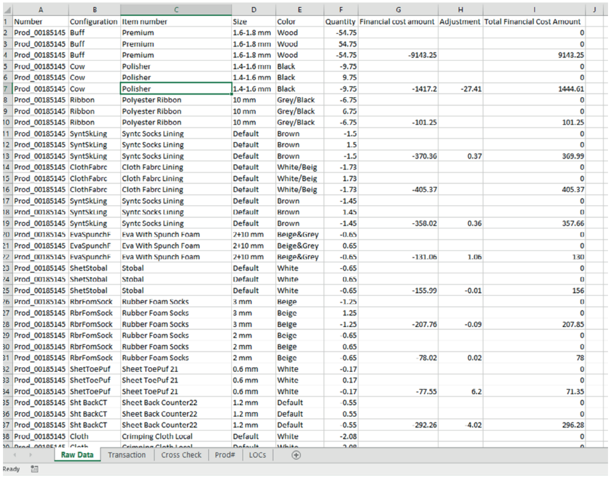
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Number	Configura	Item num	Size	Color	Quantity	Financial (	Location	Warehous	Site	Receipt st	Date	CW unit	CW quant	Physical c	Adjustment	
2	Prod_0018	Cow	Polisher	1.6-1.8 mr	Tan	-2.4	-360.99	Leather	Lthr Store	Factory					-367.69	-6.7	
3	Prod_0018	synthetic	PU Soft Lir	Default	Brown	-0.06	-15.72	Shoe Mate	Shoe Mat	Factory					-15.72		
4	Prod_0018	EvaSpuncl	Eva With S	2+10 mm	Beige&Gr	-0.01	-0.96	Shoe Mate	Shoe Mat	Factory					-2	-1.04	
5	Prod_0018	Cloth	Crimping (	Default	White	-0.03	-3.6	Shoe Mate	Shoe Mat	Factory					-3.57	0.03	
6	Prod_0018	ShetToePı	Sheet Toe	0.8 mm	White	-0.01	-5.24	Shoe Mate	Shoe Mat	Factory					-5.13	0.11	
7	Prod_0018	synthetic	PU Soft Lir	Default	Brown	-0.05	-14.15	Shoe Mate	Shoe Mat	Factory					-14.15		
8	Prod_0018	Goat	Heel Grip	0.5-0.7 mr	Brown	-0.26	-22.84	Leather	Lthr Store	Factory					-23.4	-0.56	
9	Prod_0018	Thread Lo	Polyester	20/3	M/Brown:	-10	-1.5	Shoe Mate	Shoe Mat	Factory					-0.8	0.7	
10	Prod_0018	Thread Lo	Polyester	40/3	L/Brown1	-3	-0.21	Shoe Mate	Shoe Mat	Factory					-0.21		
11	Prod_0018	Thread Lo	Polyester	30/3	M/Brown:	-7	-0.35	Shoe Mate	Shoe Mat	Factory					-0.35		
12	Prod_0018	SockPreLt	SocksPren	39-41	Green	-1	-88	Shoe Mate	Shoe Mat	Factory					-88		
13	Prod_0018	Magnets	Marine m.	6 mm	Silver	-14	-27.8	Shoe Mate	Shoe Mat	Factory					-37.66	-9.86	
14	Prod_0018	Thread H/	Hand Stitc	1 mm	M/Brown	-6.75	-6	Shoe Mate	Shoe Mat	Factory					-6.01	-0.01	
15	Prod_0018	ElasticImp	Elastic Imp	80 mm	D/Brown	-0.17	-16.99	Shoe Mate	Shoe Mat	Factory					-21.37	-4.38	
16	Prod_0018	EyeletRon	Eyelets	350 Small	Antique	-6	-3.8	Shoe Mate	Shoe Mat	Factory					-3.8		
17	Prod_0018	TapeNnSt	None Stre	10 mm	Black	-0.17	-0.94	Shoe Mate	Shoe Mat	Factory					-0.95	-0.01	
18	Prod_0018	Chemical	Pasting Sc	Default	Yellow	-0.01	-3.01	Shoe Mate	Shoe Mat	Factory					-3.01		
19	Prod_0018	Chemical	Latex	Default	White	-0.02	-5.36	Shoe Mate	Shoe Mat	Factory					-5.34	0.02	
20	Prod_0018	TPR	TPR Grain	60	Brown	-0.27	-73.38	Shoe Mate	Shoe Mat	Factory					-72.81	0.57	
21	Prod_0018	ShtShnkBr	Insole Sha	2mm	Default	-0.02	-8.71	Shoe Mate	Shoe Mat	Factory					-9.55	-0.84	
22	Prod_0018	Chemical	Pasting Sc	Default	Yellow	-0.01	-3.01	Shoe Mate	Shoe Mat	Factory					-3.01		
23	Prod_0018	Chemical	Latex	Default	White	-0.01	-2.68	Shoe Mate	Shoe Mat	Factory					-2.67	0.01	
24	Prod_0018	Chemical	Keck Prier	Default	Default	-0.03	-13.84	Shoe Mate	Shoe Mat	Factory					-14.5	-0.66	
25	Prod_0018	Thread So	Sole Stitcl	0.8 mm	D/Brown	-1.75	-1.8	Shoe Mate	Shoe Mat	Factory					-1.8		
26	Prod_0018	Shoe Box	Urban Sol	330x200x1	Default	-1	-31.89	Shoe Mate	Shoe Mat	Factory					-27.5	4.39	
27	Prod_0018	Price Tag	Price Tag (	3999	Default	-1	-0.2	Shoe Mate	Shoe Mat	Factory					-0.2		
28	Prod_0018	Tag Card	Tag Card F	80x45 mm	Brown/Blu	-1	-2.6	Shoe Mate	Shoe Mat	Factory					-2.6		
29	Prod_0018	Tag Card	Tag Card I	60x20 mm	Brown/Blu	-1	-1.4	Shoe Mate	Shoe Mat	Factory					-1.4		
30	Prod_0018	Sticker	Shoe Box	Default	Default	-1	-0.11	Shoe Mate	Shoe Mat	Factory					-1	-0.89	
31	Prod_0018	Stick	Plastic Stid	Default	Black	-2	-2.4	Shoe Mate	Shoe Mat	Factory					-2.4		
32	Prod_0018	Carry Bag	Carry Bag	Default	Wht/Blue	-1	-10.5	Shoe Mate	Shoe Mat	Factory					-10.5		
33	Prod_0018	PackngGe	Nylon Stri	5"	White	-1	-0.12	Shoe Mate	Shoe Mat	Factory					-0.12		
34	Prod_0018	Cow	Polisher	1.6-1.8 mr	Tan	-4.8	-721.98	Leather	Lthr Store	Factory					-735.38	-13.4	
35	Prod_0018	synthetic	PU Soft Lir	Default	Brown	-0.12	-31.44	Shoe Mate	Shoe Mat	Factory					-31.44		
36	Prod_0018	EvaSpuncl	Eva With S	2+10 mm	Beige&Gr	-0.02	-1.93	Shoe Mate	Shoe Mat	Factory					-4	-2.07	
37	Prod_0018	Cloth	Crimping (	Default	White	-0.07	-7.2	Shoe Mate	Shoe Mat	Factory					-7.14	0.06	
38	Prod_0018	ShetToePı	Sheet Toe	0.8 mm	White	-0.02	-10.48	Shoe Mate	Shoe Mat	Factory					-10.25	0.23	

Raw Data | Transaction | Cross Check | Prod# | LOCs

Ready

Figure 1: Transaction worksheet.

The 'Raw Data' worksheet as given in Figure 2 used to process the data at the last stage and the final summary of the report is pasted in this worksheet by the automated template.



	A	B	C	D	E	F	G	H	I	J
1	Number	Configuration	Item number	Size	Color	Quantity	Financial cost amount	Adjustment	Total Financial Cost Amount	
2	Prod_00185145	Buff	Premium	1.6-1.8 mm	Wood	-54.75			0	
3	Prod_00185145	Buff	Premium	1.6-1.8 mm	Wood	54.75			0	
4	Prod_00185145	Buff	Premium	1.6-1.8 mm	Wood	-54.75	-9143.25		9143.25	
5	Prod_00185145	Cow	Polisher	1.4-1.6 mm	Black	-9.75			0	
6	Prod_00185145	Cow	Polisher	1.4-1.6 mm	Black	9.75			0	
7	Prod_00185145	Cow	Polisher	1.4-1.6 mm	Black	-9.75	-1417.2	-27.41	1444.61	
8	Prod_00185145	Ribbon	Polyester Ribbon	10 mm	Grey/Black	-6.75			0	
9	Prod_00185145	Ribbon	Polyester Ribbon	10 mm	Grey/Black	6.75			0	
10	Prod_00185145	Ribbon	Polyester Ribbon	10 mm	Grey/Black	-6.75	-101.25		101.25	
11	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	-1.5			0	
12	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	1.5			0	
13	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	-1.5	-370.36	0.37	369.99	
14	Prod_00185145	ClothFabrc	Cloth Fabrc Lining	Default	White/Beig	-1.73			0	
15	Prod_00185145	ClothFabrc	Cloth Fabrc Lining	Default	White/Beig	1.73			0	
16	Prod_00185145	ClothFabrc	Cloth Fabrc Lining	Default	White/Beig	-1.73	-405.37		405.37	
17	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	-1.45			0	
18	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	1.45			0	
19	Prod_00185145	SyntSkLing	Syntc Socks Lining	Default	Brown	-1.45	-358.02	0.36	357.66	
20	Prod_00185145	EvaSpunchF	Eva With Spunch Foam	2+10 mm	Beige&Grey	-0.65			0	
21	Prod_00185145	EvaSpunchF	Eva With Spunch Foam	2+10 mm	Beige&Grey	0.65			0	
22	Prod_00185145	EvaSpunchF	Eva With Spunch Foam	2+10 mm	Beige&Grey	-0.65	-131.06	1.06	130	
23	Prod_00185145	ShetStobal	Stobal	Default	White	-0.65			0	
24	Prod_00185145	ShetStobal	Stobal	Default	White	0.65			0	
25	Prod_00185145	ShetStobal	Stobal	Default	White	-0.65	-155.99	-0.01	156	
26	Prod_00185145	RbrFomSock	Rubber Foam Socks	3 mm	Beige	-1.25			0	
27	Prod_00185145	RbrFomSock	Rubber Foam Socks	3 mm	Beige	1.25			0	
28	Prod_00185145	RbrFomSock	Rubber Foam Socks	3 mm	Beige	-1.25	-207.76	-0.09	207.85	
29	Prod_00185145	RbrFomSock	Rubber Foam Socks	2 mm	Beige	-0.65			0	
30	Prod_00185145	RbrFomSock	Rubber Foam Socks	2 mm	Beige	0.65			0	
31	Prod_00185145	RbrFomSock	Rubber Foam Socks	2 mm	Beige	-0.65	-78.02	0.02	78	
32	Prod_00185145	ShetToePuf	Sheet ToePuf 21	0.6 mm	White	-0.17			0	
33	Prod_00185145	ShetToePuf	Sheet ToePuf 21	0.6 mm	White	0.17			0	
34	Prod_00185145	ShetToePuf	Sheet ToePuf 21	0.6 mm	White	-0.17	-77.55	6.2	71.35	
35	Prod_00185145	Sht BackCT	Sheet Back Counter22	1.2 mm	Default	-0.55			0	
36	Prod_00185145	Sht BackCT	Sheet Back Counter22	1.2 mm	Default	0.55			0	
37	Prod_00185145	Sht BackCT	Sheet Back Counter22	1.2 mm	Default	-0.55	-292.26	-4.02	296.28	
38	Prod_00185145	Cloth	Crimping Cloth Local	Default	White	-2.08			0	

Raw Data | Transaction | Cross Check | Prod# | LOCs

Ready

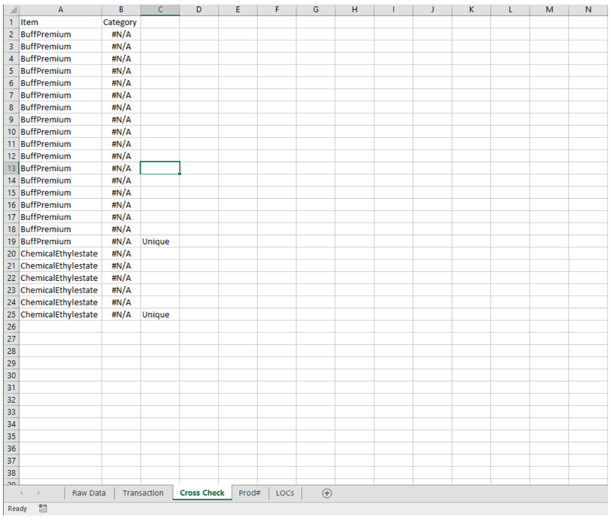
Figure 2: Raw Data worksheet.

The 'cross check' worksheet as given in Figure 3 is kept to cross-check if any item in the report is not classified.

Table 1: Set of observations recorded for each activity to complete the report.

Activity	Obs. 1	Obs. 2	Obs. 3	Obs. 4	Obs. 5	Obs. 6	Obs. 7	Obs. 8	Obs. 9	Obs. 10	Average (min)
a1	12.16	7.72	9.83	9.89	10.07	8.48	6.01	8.24	9.74	10.12	10.20
a2	3.96	5.68	3.71	5.96	7.97	4.82	5.42	5.21	4.53	5.46	5.73
a3	7.71	6.57	6.44	7.40	7.30	7.66	5.92	7.47	6.23	7.23	7.62
a4	6.83	5.96	6.94	6.83	6.56	7.52	7.08	7.19	6.81	7.80	7.63
a5	21.45	19.34	18.24	19.53	20.11	24.22	19.26	23.63	19.78	15.50	22.08
a6	5.54	4.69	4.79	5.62	4.95	3.96	4.65	4.72	4.60	4.43	5.26
					Total						**58.51**

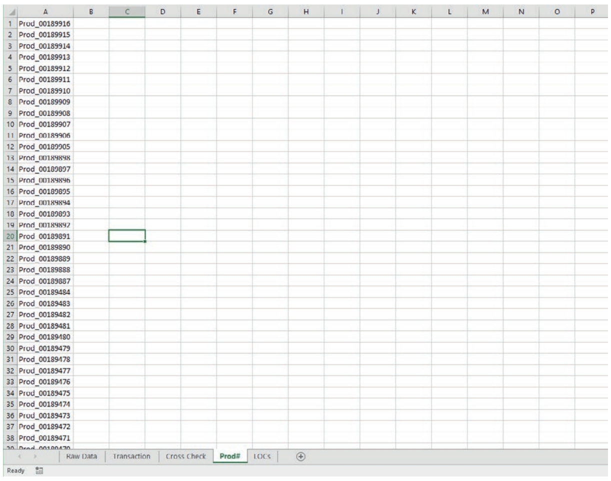
a1 = Write down the production numbers that are used in the specific month.
a2 = Call transaction of collected production numbers and download the data into Microsoft Excel
a3 = Filter the data according to the obtained production numbers
a4 = Make preliminary calculations on the downloaded data
a5 = Add categories for each item
a6 = Calculate a summary of the quantity and cost as per the material classification added in the report



	A	B	C
1	Item	Category	
2	BuffPremium	#N/A	
3	BuffPremium	#N/A	
4	BuffPremium	#N/A	
5	BuffPremium	#N/A	
6	BuffPremium	#N/A	
7	BuffPremium	#N/A	
8	BuffPremium	#N/A	
9	BuffPremium	#N/A	
10	BuffPremium	#N/A	
11	BuffPremium	#N/A	
12	BuffPremium	#N/A	
13	BuffPremium	#N/A	
14	BuffPremium	#N/A	
15	BuffPremium	#N/A	
16	BuffPremium	#N/A	
17	BuffPremium	#N/A	
18	BuffPremium	#N/A	
19	BuffPremium	#N/A	Unique
20	ChemicalEthylestate	#N/A	
21	ChemicalEthylestate	#N/A	
22	ChemicalEthylestate	#N/A	
23	ChemicalEthylestate	#N/A	
24	ChemicalEthylestate	#N/A	
25	ChemicalEthylestate	#N/A	Unique

Figure 3: Cross check worksheet.

The ‘Prod#’ worksheet as given in Figure 4 was used to keep the production numbers in the first column, based on which the report was designed to start.



	A
1	Prod_00189916
2	Prod_00189915
3	Prod_00189914
4	Prod_00189913
5	Prod_00189912
6	Prod_00189911
7	Prod_00189910
8	Prod_00189909
9	Prod_00189908
10	Prod_00189907
11	Prod_00189906
12	Prod_00189905
13	Prod_00189898
14	Prod_00189897
15	Prod_00189896
16	Prod_00189895
17	Prod_00189894
18	Prod_00189893
19	Prod_00189892
20	Prod_00189891
21	Prod_00189890
22	Prod_00189889
23	Prod_00189888
24	Prod_00189887
25	Prod_00189484
26	Prod_00189483
27	Prod_00189482
28	Prod_00189481
29	Prod_00189480
30	Prod_00189479
31	Prod_00189478
32	Prod_00189477
33	Prod_00189476
34	Prod_00189475
35	Prod_00189474
36	Prod_00189473
37	Prod_00189472
38	Prod_00189471

Figure 4: Worksheet for production numbers (Prd#).

The ‘List of categories’ (LOCs) worksheet was used to classify the items into various categories so that a summary of items used could be extracted easily.

4.2.2. Link Between Worksheets

An automated MCCA template was designed and programmed in a way that five worksheets are needed in the template to function perfectly. At the very first, production numbers of the current month are supposed to be listed in the ‘Pord#’ worksheet. Then the data transaction is supposed to be called from ERP against the obtained production numbers and then the downloaded data is supposed to be put into the ‘Transaction’ worksheet.

The template then filters the values of production numbers in the “Transaction” worksheet. The transaction is further processed by the template after the execution of macros and then the final summary of the report is pasted into the “Raw Data” worksheet by the template as given in Figure 5.

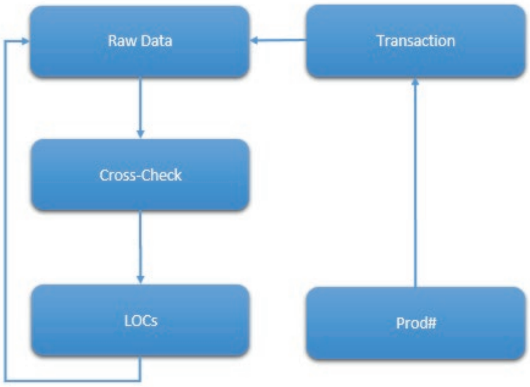


Figure 5: Data Fetching among various worksheets in the automated template.

4.2.3. The function of Automated Template

The userform given in Figure 6 was designed with command buttons so that the various programmed macros could be called when the button is clicked. VBA code behind each command button is given in separate appendices at the end of the present research.

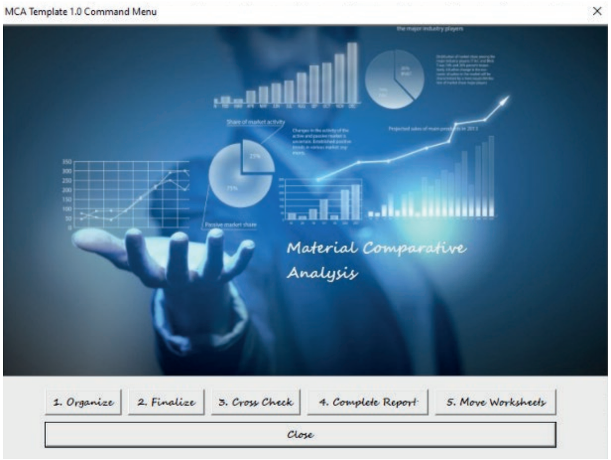


Figure 6: Userform designed for the execution of macros through command buttons.

Figure 7 is the representation of a sequence of clicking command buttons so that the report can be completed properly. The number of macros behind each command button is mentioned in Figure 7.

4.2.3.1. Organize

There were only two macros in the ‘Organize’ command button. The VBA code of both macros is given in Appendix 1. The output obtained after clicking the ‘Organize’ command button.

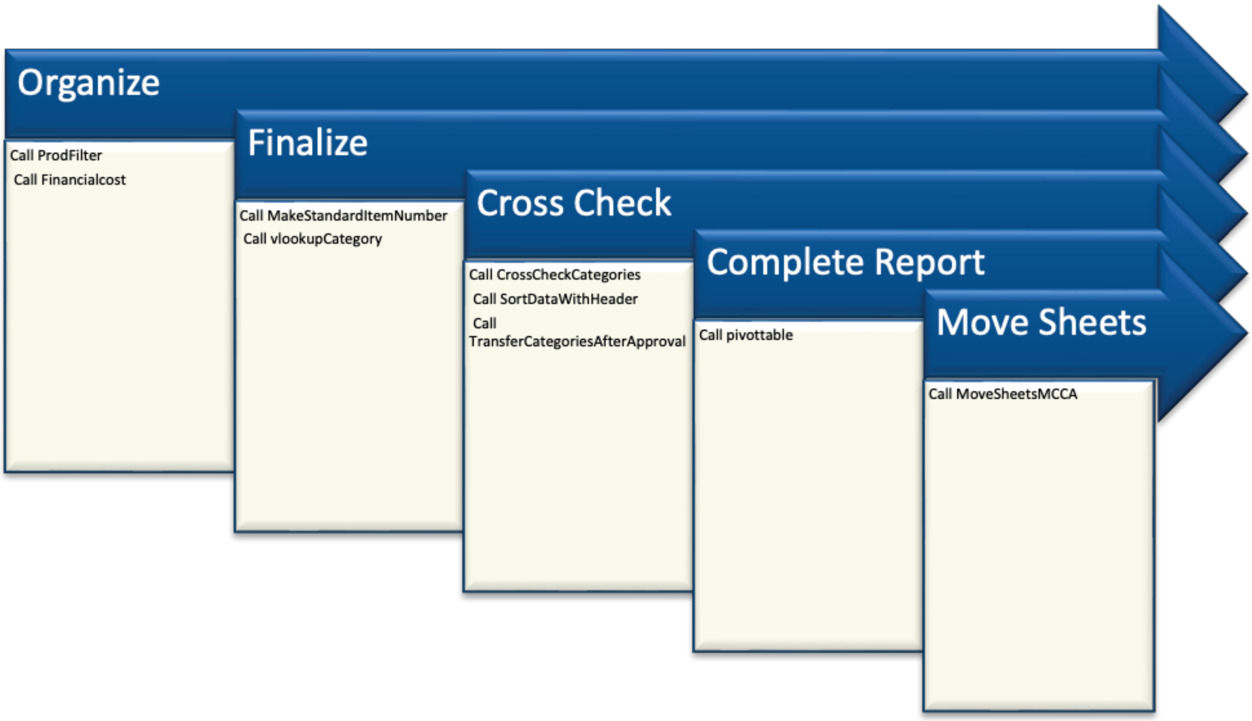


Figure 7: Used Command buttons and macros at their back in automated material cost comparative analysis.

4.2.3.2. Finalize

There were only two macros in the 'Finalize' command button. The VBA code of both macros is given in Appendix 2. The output obtained after clicking the 'Finalize' command button.

4.2.3.3. Cross Check

There were three macros in the 'Cross Check' command button. The VBA code of both macros is given in Appendix 3. This command button was included in the userform to check if any item is left to be classified in the LOCs worksheet. The msgbox is supposed to pop up to take the confirmation of the user to add the missing items in the LOCs; so that the final summary can adequately be calculated. The output obtained after clicking the 'Cross Check' command button.

4.2.3.4. Complete Report

There was only one macro at the back of the 'Complete Report' command button. The VBA code of the macro is given in Appendix 4. The output obtained after clicking the 'Complete Report' command button.

4.2.3.5. Transfer Worksheets

When the report is completely made the three worksheets (i.e. Raw Data, Transaction, and Prod#) are transferred to the new workbook; the template file becomes havier because of a lot of data pasted into it; thus the important worksheets were transferred to the new workbook so that the file could be mailed conveniently. The code behind the 'Transfer Worksheets' command button is given in Appendix 5.

4.2.3.6. Time Required to Prepare the Report by Suggested Method

The time required to make the report through the automated MCCA template is given in Table 2. Each command button's execution time is mentioned in the table. Calculating the total duration of all the activities, it comes up as 16.26 minutes.

Table 2: Time required for the report to be made using the suggested method.

Activity	Average Time (Sec)
Write down the production numbers that are used in the specific month	612
Call transaction of collected production numbers and download the data into Microsoft Excel	343.8
Time taken by 'Organize' button to execute its macros	4
Time taken by 'Finalize' button to execute its macros	3
Time taken by 'Cross Check' button to execute its macros	2
Time taken by the 'Complete Report' button to execute its macros	5
Time taken by Move the' Worksheets' button to execute its macros	6
Total	975.8
Total Time in Minutes	16.26

1. Comparison of Both Methods

The manual operation of the report used to take 58.51 minutes from the employee's time. Whereas, the MCCA template took only 26.26 minutes to make the report. The comparison of time consumption of methods indicates that the MCCA template took 72.2% less time as compared to the manual preparation of the report.

5. Discussion

To create daily, monthly, quarterly and annual excel reports every small and medium-sized business has at least one employee. Excel tasks that must be carried out manually take too much time for staff to complete, and manual work has a higher risk of errors. With relation to office suites like Word, Excel, PowerPoint, and MS Projects, etc. the software manufacturer has maintained proficiency. Microsoft created numerous technologies, including VBA, ActiveX, VSTO, and many others, in response to user demands (Porter & Stretcher, 2012; Ding et al., 2017).

Many different window-based software programs are compatible with VBA technology (Kuka & Karamani, 2011; Norton & Tiwari, 2013; Harahap & Azmi, 2017). Companies today assist their staff members in enhancing their expertise in Excel and VBA (Chatvichiencha, 2015). The integrated development environment (IDE), which is used to create and modify VBA technology, is used by office programs. As a result, difficult and repetitive tasks can be automated and simplified respectively (Minto, 2009; Kuka & Karamani, 2011; Evensen, 2014; Ding et al., 2017; Harahap & Azmi, 2017). Automated procedures are used by modern office productivity software to finish mundane tasks (Chatvichiencha, 2015; Ding et al., 2017). All manual processes, including the creation of the report, were to be entirely automated as part of this study. VBA uses forms to gather user input (Evensen, 2014). To get user feedback on this article, the author added a combo box to the page (the report he wants o work on). To complete the duties automatically, the user form could also contain a code and actions (Evensen, 2014) (Harahap & Azmi, 2017; Kalwar & Khan, 2020). Similar to the order costing report, one-user forms are built with command buttons that have macros to swiftly carry out the preprogrammed operations on their backs (one macro for each task). The entire process of the challenging report was completed utilizing the faster, more flexible data migration and analysis technique developed and implemented by Bartoszewis and Wdowics (2019) using VBA (operation time decreased from 2 hours to 5 minutes) (Bartoszewicz & Wdowicz, 2019). The human power research planning report was automated by Cirjuano and Zhu (2013) using excel VBA. An inexperienced reporter used to need 30 working hours to complete the manual report; now, with automation, it just takes 10 minutes (99.4% less time) (Cirujano & Zhu, 2013). By Kalwar and Khan (2020), staff members were able to save 75% of the time they would have needed to manually produce the purchase report. In a similar line, automating the paper report for the production pan takes 83.18% less time than doing it manually. Yan and Wan (2017) utilized Excel VBA to design an application for the automatic computation and development of the bill of materials or a transmission line (BOM), designing and using a template dramatically enhances efficiency and accuracy while lowering errors while generating the whole steel BOM (Yan & Wan, 2017). VBA was used by Abidin et al.,(2013) To generate automatic WQI and API estimates. A convenient method for calculating WQI and API was supplied by the application and the automation decreased computation time and error (Abidin et al., 2015). Similarly, the material cost comparative analysis report There were various worksheets included in the MCA template. The automated template was found to consume only 16.26 minutes and it was far quicker than the current technique. The suggested method took 72.2% less time as compare to the current method of preparing the material cost cost comparative analysis report. Moreover, the the report was programmed, troubleshooted and its output verified in 15 hours. It was quite a less time to invest, when it was about saving the time of several employees every month.

6. Conclusion

When machines are used for repetitive tasks to be performed, the expected accuracy is supposed to be 100% after reasonable troubleshooting of the system. At the same time, the investment in machine or the incurred time to automate the manual operation, this investment of time or money brings work simplification, accuracy and increased productivity with itself. In the case of the present template i.e. automated MCCA template, the accuracy was 100% supplemented with a 72.20%-time reduction in the making of the report. Moreover, 15 hours were incurred to program and verify the template`s output. When a greater amount of time of an employee is saved they get free to relax and focus on other innovative ideas that can bring improvement in daily operations at the company. After this report automation, creative thinking, and ideas were noticed in the employee whose time was saved by the automated MCCA template. Therefore, it is beneficial to facilitate employees rather than stressing them and especially when there is the question of productivity enhancement.

7. Future Implications

The person who utilized the template lacked the expertise required to fix any coding mistakes. The researcher volunteered to teach VBA, but nobody could enroll because of his busy schedule. At the same time, nobody in the entire business has the required VBA knowledge. Person utilized although the template is still devoid of errors, there is a chance (even if it is quite less).

8. Limitations

Since the data that is supposed to be processed and summarized can be a million rows. The suggested method can take a little more time (around 2 -3 minutes). This results from processing an enormous amount of data across a variety of rows. MS Excel will undoubtedly be slow when the data is large and the process is drawn out because it is a little application compared to a database like SQL Server or Oracle. The automated MCCA template's main drawback.

9. Acknowledgement

The co-author's encouragement and support are appreciated, and I would like to thank the employee of the company for their devoted cooperation throughout this project.

10. Conflict of Interest

The current study's authors have no conflicting interests.

References

Abidin, I.Z., Juahir, H., Azid, A., Mustafa, A.D., & Azaman, F. (2015). Application of Excel-VBA for computation of water quality index and air pollutant index. *Malaysian Journal of Analytical Sciences*, *19*(5), 1056–1064.

Abraham, R., Burnett, M., & Erwig, M. (2008). Spreadsheet Programming. In *Wiley Encyclopedia of Computer Science and Engineering*. https://doi.org/10.1002/9780470050118.ecse415

Ahmadi, A., Robinson, P.H., Elizondo, F., & Chilibroste, P. (2018). Implementation of CTR dairy model using the visual basic for application language of Microsoft excel. *International Journal of Agricultural and Environmental Information Systems*, *9*(3), 74–86. https://doi.org/10.4018/IJAEIS.2018070105

Ajinkya, W.S., Sachin, R.S., Manoj, S.D., Rajesh, R.M., & Mangesh, J. B. (2017). Preparing excel sheet for estimation and costing. *International Journal of Engineering Sciences & Management*, *7*(1), 310–317.

Balson, D. (2012). User Defined Spreadsheet Functions in Excel. *Proceedings of EuSpRIG 2012 Conference "The Science of Spreadsheet Risk Management*.

Bartoszewicz, G., & Wdowicz, M. (2019). Automation of the Process of Reporting the Compliance of the Production Plan with Its Execution Based on Integration of SAP ERP System In Connection With Excel Spreadsheet and VBA Application. In *Digitalization of Supply Chains*. https://doi.org/10.17270/b.m.978-83-66017-86-3

Belchior Junior, A., Bruel, R.N., Andrade, D.A., Sabundjian, G., Macedo, L.A., Angelo, G., … Conti, T.N. (2011). Development of a Vba Macro-Based Spreadsheet Application for Relap5 Data Post-Processing. *International Nuclear Atlantic Conference*, 978–985.

Blattner, D.E., & Valrico, F. (2007). *Patent No. US 2007/0055688 A1*. United States.

Blayney, P.J., & Sun, Z. (2019). Using Excel and Excel VBA for Preliminary Analysis in Big Data Research. In *Managerial Perspectives on Intelligent Big Data Analytics. IGI Global* (pp. 110–136). https://doi.org/10.4018/978-1-5225-7277-0.ch007

Botchkarev, A. (2015). *Assessing Excel VBA Suitability for Monte Carlo Simulation*. 1–30. Retrieved from http://arxiv.org/abs/1503.08376

Broman, K.W., Woo, K.H., & Woo, K.H. (2017). *Data organization in spreadsheets Data organization in spreadsheets and* (Vol. 1305). https://doi.org/10.1080/00031305.2017.1375989

Chatvichiencha, S. (2015). Enhancing Computational Thinking by Excel-VBA Based Problem Solving. *The 2nd International Conference on Innovation in Education*, 284–290.

Cirujano, J., & Zhu, Z. (2013). Automatic reporting for manpower resources. *Proceedings, Annual Conference - Canadian Society for Civil Engineering*, *1*(January), 710–719.

Ding, H.L., Qi, K.Y., Zhaq, X.L., & Xu, G.F. (2017). Tibetan Typographical Specifications and Technical Realization Based on Word VBA. *4th International Conference on Advanced Education and Management*, 407–412. https://doi.org/10.12783/dtssehs/icaem2017/19117

Dunn, A. (2009). Automated Spreadsheet Development. In *arXiv preprint arXiv* (pp. 908–928).

Evensen, H.T. (2014). A versatile platform for programming and data acquisition: Excel and Visual Basic for Applications. *ASEE Annual Conference and Exposition*. https://doi.org/10.18260/1-2--20017

Fisher, M., Cao, M., Rothermel, G., Cook, C.R., Burnett, M.M. (2002). Automated Test Case Generation for Spreadsheets. *24th International Conference on Software Engineering*, 141–153. https://doi.org/10.1145/581339.581359

Harahap, M.I.P., & Azmi, M.H. (2017). Development of Excel VBA Program for Small Drainage Network. *E-Academia Journal*, *6*(1), 216–227.

Hila, R. (2009). Water Quality Data Management Database.

Kalwar, M.A., & Khan, M.A. (2020). Optimization of Procurement & Purchase Order Process in Foot Wear Industry by Using VBA in Ms Excel. *International Journal of Business Education and Management Studies (IJBEMS)*, *5*(2), 80–100. Retrieved from https://www.ijbems.com/doc/IJBEMS-124.pdf

Karim, A.J. (2011). The Significance of Management Information Systems For Enhancing Strategic And Tactical Planning. *Journal of Information Systems and Technology Management*, *8*(2), 459–470. https://doi.org/10.4301/S1807-17752011000200011

Kuka, S., & Karamani, B. (2011). Using Excel and VBA for Excel to Learn Numerical Methods. *1st International Sympsosium on Computing in Informatics and Mathematics*, 365–376.

Lessa, J.D.R., Lessa, P.F.R., Almeida, P.A., Junior, M., & Guimarães, H. de V. (2016). Mathematical Model and Programming in VBA Excel for Package Calculation. *International Journal of Engineering Research and Applications*, *6*(5), 55–61.

Minto, C.F. (2009). *PKPD Tools for Excel*.

Norton, T., & Tiwari, B. (2013). Aiding the understanding of novel freezing technology through numerical modelling with visual basic for applications (VBA). *Computer Applications in Engineering Education*, *21*(3), 530–538. https://doi.org/10.1002/cae.20498

Perry, K.M. (2012). *The Call Center Scheduling Problem using Spreadsheet Optimization and VBA*. Virginia Commonwealth University.

Porter, D., & Stretcher, R. (2012). Automating Markowitz Optimizations Using VBA. *Journal of Instructional Techniques in Finance*, *4*(1), 9–16.

Raza, M., & Gulwani, S. (2017). Automated data extraction using predictive program synthesis. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 882–890. https://doi.org/10.1609/aaai.v31i1.10668

Sato, K., & Yokoyama, R. (2001). Teaching Aid for Remote Sensing and Map Imagery Analysis Using Excel Spreadsheet and VBA. *22nd Asian Conference on Remote Sensing*. Retrieved from http://www.crisp.nus.edu.sg/~acrs2001/pdf/015venka.pdf

Wettlaufer, G.J. (2010). *Merlin.Net Automation of External Reports Verification Process* (California Polytechnic State University). California Polytechnic State University. https://doi.org/10.15368/theses.2010.3

Yan, Q., & Wan, Y. (2017). Using the special font and VBA program to make bill of materials in the transmission line engineering. *Revista de la Facultad de Ingenieria*, *32*(2), 335–341.

Appendix 1

Macro 1

```
Sub ProdFilter()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
   Dim prd() As Variant
   Dim rowscount As Variant
   Dim arr As Variant
   Dim rnge As Range
   With Worksheets("Transaction")
      .Columns("H:H").EntireColumn.Delete
      .Columns("I:N").EntireColumn.Delete
   End With
   arr = Array("Lthr Store", "Shoe Mat")
   Worksheets("Prod#").Select
    rowscount = Worksheets("Prod#").Cells(Rows.Count, 1).End(xlUp).row
   ReDim Preserve prd(rowscount) As Variant
   For i = 1 To rowscount
      prd(i) = Worksheets("Prod#").Range("A" & i)
   Next i

     rowscount = Worksheets("Transaction").Cells(Rows.Count, 1).End(xlUp).row

   With Worksheets("Transaction")
      .Select
       .Range(Cells(1, 1), Cells(rowscount, 10)).AutoFilter 1, prd(), xlFilterValues
       .Range(Cells(1, 1), Cells(rowscount, 10)).AutoFilter Field:=8, Criteria1:=arr, Operator:=xlFilterValues
      .Range(Cells(1, 1), Cells(rowscount, 10)).Select
      Selection.Copy
   End With
   With Worksheets("Raw Data")
      .Select
      .Range("A1").Select
      .Paste
   End With
   Worksheets("Transaction").AutoFilterMode = False
                Worksheets("Raw Data").Columns("H:H").EntireColumn.Delete
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Macro 2

```
Sub Financialcost()
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
On Error Resume Next
Dim rowscount As Long
rowscount = Worksheets("Raw Data").Cells(Rows.Count, 1).End(xlUp).row
With Worksheets("Raw Data")
   .Range("I2").Formula = "=-(G2+H2)"
   .Range("I2").Select
      Selection.AutoFill Destination:=.Range(Cells(2, 9), Cells(rowscount, 9))
   .Range("I1").Value = "Total Financial Cost Amount"
End With
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Appendix 2

Macro 1

```
Sub MakeStandardItemNumber()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
rowscount = Worksheets("Raw Data").Cells(Rows.Count, 1).End(xlUp).row
With Worksheets("Raw Data")
   .Columns("B:B").Insert
   .Cells(1, 2).Value = "Item"
   .Range("B2").Formula = "=concatenate(C2,D2)"
   .Range("B2").Select
      Selection.AutoFill Destination:=.Range(Cells(2, 2), Cells(rowscount, 2))
End With
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Macro 2

```
Sub vlookupCategory()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
Dim vlookuparray As Range
Dim vlookupvalue As Range
Dim row As Variant
row = 2
rowscount = Worksheets("Raw Data").Cells(Rows.Count, 1).End(xlUp).row
Worksheets("Raw Data").Cells(1, 11).Value = "Category"
With Worksheets("Raw Data")
                        .Range("K2").Formula = "=VLOOKUP(B2,LOCs!$B$2:$C$50000,2,FALSE)"
   .Range("K2").Select
      Selection.AutoFill Destination:=.Range(Cells(2, 11), Cells(rowscount, 11))
End With
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Appendix 3

Macro 1

```
Sub CrossCheckCategories()
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
On Error Resume Next
Dim rowscount As Variant
Dim arr As Variant
Dim rnge As Range
rowscount = Worksheets(“Raw Data”).Cells(Rows.Count,
1).End(xlUp).row
  rowscount = Worksheets(“Raw Data”).Cells(Rows.
Count, 1).End(xlUp).row
Set rnge = Worksheets(“Raw Data”).Range(Cells(1, 1),
Cells(rowscount, 11))
   With Worksheets(“Raw Data”)
            rnge.AutoFilter Field:=11, Criteria1:=”#N/A”,
Operator:=xlAnd
      rnge.Select
      Selection.Copy
   End With
   With Worksheets(“Cross Check”)
      .Select
      .Range(“A1”).Select
      .Paste
      .Columns(“A:A”).EntireColumn.Delete
      .Columns(“B:I”).EntireColumn.Delete
   End With
   Worksheets(“Raw Data”).ShowAllData
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Macro 2

```
Sub SortDataWithHeader()
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
   With Worksheets(“Cross Check”)
      rowscount = .Cells(Rows.Count, 1).End(xlUp).row
            Worksheets(“Cross Check”).Range(Cells(1,
1), Cells(rowscount, 3)).Sort Key1:=Range(“A1”),
Order1:=xlAscending, Header:=xlYes
      For i = 2 To rowscount
         If .Cells(i + 1, 1).Value = .Cells(i, 1).Value Then
            .Cells(i, 3).Value = “”
         Else
            .Cells(i, 3).Value = “Unique”
         End If
      Next
   End With
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Macro 3

```
Sub TransferCategoriesAfterApproval()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
rowscount = Application.WorksheetFunction.
CountIf(Worksheets(“Cross Check”).
Range(“C1:C1048576”), “Unique”)
   If rowscount > 0 Then
      result = MsgBox(“There are blank categories i.e. “ &
rowscount & vbNewLine & “Do you want to add categories
in LOCs Worksheet?”, vbYesNo + vbInformation)
         If result = vbYes Then
               rowscountC = Worksheets(“Cross Check”).
Cells(Rows.Count, 1).End(xlUp).row
                  rowscountLOC = Worksheets(“LOCs”).
Cells(Rows.Count, 2).End(xlUp).row + 1
            For i = 2 To rowscountC
             If Worksheets(“Cross Check”).Cells(i, 3).Value
= “Unique” Then
                With Worksheets(“LOCs”)
                        .Cells(rowscountLOC, 2).Value =
Worksheets(“Cross Check”).Cells(i, 1)
                End With
                rowscountLOC = rowscountLOC + 1
             End If
            Next
            Unload UserForm1
         Else
            Unload UserForm1
         End If
         With Worksheets(“LOCs”)
            .Select
        rowscountLOC = .Cells(Rows.Count, 2).End(xlUp).
row + 1
            .Cells(rowscountLOC - rowscount, 2).Select
         End With
   End If
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Appendix 4

```
Sub pivottable()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
Dim Cache As PivotCache
Dim Table As pivottable
Dim pRange As Range
Dim rowscount As Variant
rowscount = Worksheets(“Raw Data”).Cells(Rows.Count,
1).End(xlUp).row
With Worksheets(“Raw Data”)
      .Range(“G:G”).Replace What:=”-”, Replacement:=””,
LookAt:=xlPart, SearchOrder:=xlByRows, MatchCase:=-
False, SearchFormat:=False, ReplaceFormat:=False
```

```
End With
   Set pRange = Worksheets("Raw Data").Range(Cells(1,
1), Cells(rowscount, 11))
     Set Cache = ActiveWorkbook.PivotCaches.Create(-
SourceType:=xlDatabase,     SourceData:=pRange.Ad-
dress(False, False, xlA1, xlExternal))
      Set Table = Cache.CreatePivotTable(TableDesti-
nation:=Worksheets("Raw  Data").Cells(1,  12),  Table-
Name:="Transaction")
     With Worksheets("Raw Data").PivotTables("Transac-
tion").PivotFields("Category")
      .Orientation = xlRowField
      .Position = 1
   End With
     Worksheets("Raw Data").PivotTables("Transaction").
AddDataField     ActiveSheet.PivotTables("Transaction").
PivotFields("Quantity"), "Sum of Quantity", xlSum
     Worksheets("Raw Data").PivotTables("Transaction").
AddDataField     ActiveSheet.PivotTables("Transaction").
PivotFields("Total Financial Cost Amount"), "Amount", xl-
Sum
   With Worksheets("Raw Data")
      .Columns("A:K").EntireColumn.Delete
       rowscount = Worksheets("Raw Data").Cells(Rows.
Count, 1).End(xlUp).row
      .Range(Cells(2, 1), Cells(rowscount - 1, 3)).Select
      Selection.Copy
      .Cells(1, 4).Select
      .Paste
      .Columns("A:C").EntireColumn.Delete
      .Cells(1, 1).Value = "Category"
      .Cells(1, 3).Value = "Financial Cost Amount"
   End With
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```

Appendix 5

```
Sub MoveSheetsMCCA()
On Error Resume Next
Application.ScreenUpdating = False
Application.DisplayAlerts = False
Application.Calculation = xlCalculationManual
Dim RawData As Worksheet
Dim Prod As Worksheet
Dim Trans As Worksheet
   Set RawData = ThisWorkbook.Worksheets("Raw Data")
   Set Prod = ThisWorkbook.Worksheets("Prod#")
   Set Trans = ThisWorkbook.Worksheets("Transaction")
   Workbooks.Add.SaveAs
                  "MaterialCostComparativeAnalysis"
   Workbooks("MaterialCostComparativeAnalysis").
                 Activate
   Prod.Copy Before:=Workbooks
                 ("MaterialCostComparativeAnalysis").
                 Sheets(1)
   Trans.Copy Before:=Workbooks
                  ("MaterialCostComparativeAnalysis").
                  Sheets(1)
   RawData.Copy Before:=Workbooks
         ("MaterialCostComparativeAnalysis").Sheets(1)
   Sheets("Sheet1").Delete
Application.ScreenUpdating = True
Application.DisplayAlerts = True
Application.Calculation = xlCalculationAutomatic
End Sub
```