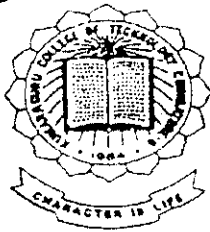


# "INVENTORY CONTROL SYSTEM" PORTAL FOR A MANUFACTURING COMPANY

## PROJECT REPORT

Submitted in partial fulfillment of the requirements for  
award of Degree M.Sc ( Applied Science)  
Software Engineering



Submitted by

**Mr. M. Raj Kumar**  
(Reg. No: 0037S0097)

## UNDER THE GUIDANCE OF

External Guide

**Ms. Meenakshi Ramesh**  
Global Software Ltd.,

Internal Guide

**Mr. A. Muthukumar**  
**M.C.A., M.Phil.,**  
Asst. Professor  
CSE Department

# CERTIFICATE

**Department of Computer Science and Engineering**

**Kumaraguru College of Technology**

**Coimbatore – 641 006**



This is to certify that the project work entitled

**“INVENTORY CONTROL SYSTEM”**

**PORTAL FOR A MANUFACTURING COMPANY**

Has been submitted by

**Mr. M. Raj Kumar**

**(Reg.No: 0037S0097)**

In partial fulfillment of the award of the degree of

**Master of Science in Applied Science – Software Engineering of**

**Bharathiar University, Coimbatore**

**During the academic year 2003-2004**

**Guide**

**Head of the Department**

Certified that we examined the candidate in the Project Work Viva Voce Examination held on 29.9.2003


**Internal Examiner**

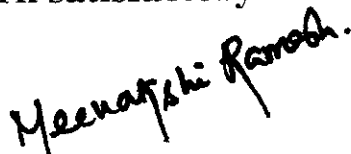
**External Examiner**

16/09/03

**CERTIFICATE**

This is to certify that Mr.Rajkumar .M. of IV th year Msc  
(S/w Engineering ) student of Kumaraguru College of  
Technology , Coimbatore , has done his project on "Inventory  
Control System" under my guidance during June 2003 to  
September 2003 . We find his project work satisfactory.

  
**J. Selvamuthukumar**  
(Deputy - Manager Training )

  
**Meenakshi Ramesh**  
( Project Guide )

# **DECLARATION**


## **DECLARATION**

I hereby declare that the project entitled “ **INVENTORY CONTROL SYSTEM** “ submitted to **GLOBAL SOFTWARE LIMITED, Chennai** in partial fulfillment of the requirements for the award of the degree of Master of Science (Applied Science) Software Engineering, is a record of original work done by me, under the supervision and guidance of **Mrs. Meenachi Ramesh, GLOBAL SOFTWARE LIMITED, Chennai.**

Place: Coimbatore.

Signature

Date: 25-09-2003



**Raj Kumar M**

**(Reg.No: 0037S0097)**

# **ACKNOWLEDGEMENT**

## **ACKNOWLEDGEMENT**

I deem it a great pleasure to place my deep sense of gratitude and indebtedness to, **Dr. K. K. Padmanaban, B.Sc. (Egg.), M.Tech. Ph.D., Principal**, Kumaraguru College of Technology for giving me the opportunity to undertake the project work.

I am grateful to, **Dr. S. Thangasamy, Ph.D., Professor and Head of the Department**, Kumaraguru College of Technology, for giving me this golden opportunity to carry out my project work successfully.

My sincere thanks are offered to **Mr. A.Muthukumar M.c.a., M.phil.**, for the encouragement and support bestowed on me as my Project Guide. I am very much indebted to him for the suggestions and guidance extended in successfully completing the project.

I thank all my faculties whose diligent efforts have led me to complete the project successfully.

I owe my deepest gratitude to **Mr. J.Selva Muthu Kumar GLOBAL SOFTWARE LIMITED**, for rendering me permission to carry out my project work in the esteemed concern.

# **SYNOPSIS**



My sincere thanks to **Mrs. Meenachi Ramesh**, GLOBAL SOFTWARE LIMITED, my project guide for her valuable guidance, timely suggestions and constant assistance in time of need.

I wish to express my sincere thanks to the people who have contributed a lot towards the successful completion of this project work.

## **SYNOPSIS**

The project entitled “ **INVENTORY CONTROL SYSTEM** ” is developed for a **MANUFACTURING** company using **COBOL** as Programming Language and **VSAM** as Back End under Mainframe platform.

“Inventory Control System” is developed for manufacturing company. The company manufacture electronics equipment has about 12000 items in there store consisting of mechanical hardware items, electronic components, subassemblies, electrical items, spares, packing materials etc.

The company purchase goods for manufacturing this item. The manufacturing has details for every item to be manufactured. Every item is purchased depending on the sales. The details of the material needed for to manufacturer a particular item is also include. The items manufactured is sold and also used for developing project by the company.

The application package has purchase file, manufacturing file, inventory master file, transaction history ledger file and transaction files. All this transaction is manipulated using transaction files. Every month the transaction files containing details are created afresh and posted in to history ledger, updating the corresponding master file.

The purchase file contains details of purchased items and their details. The manufacture file contains the details about an item to be manufactured.

After manufacturing a particular item the details are updated in the inventory master file. The history ledger contains every monthly transaction to the inventory master file. Using history ledger every monthly transactions can be retrieved.

# **CONTENTS**

# CONTENTS

	Page No.
1. Introduction. ....	01
1.1. Company Profile. ....	01
1.2. Existing System. ....	03
1.3. Proposed System. ....	05
1.4. Relevance and Importance. ....	06
2. Software Requirements. ....	07
2.1. Software Specification. ...	07
2.2. Hardware Specification. ...	08
2.3. Specific Requirements. ...	09
2.3.1. Functional Requirements. ....	09
2.3.2. Performance Requirements. ....	11
3. System Design And Development. ....	12
3.1. Module Description. ....	13
3.2. Integration of Modules. ....	15
3.3. Context Analysis Diagram. ....	16
3.4. Data Flow Diagram. ....	17
3.5. File Structure. ....	20
4. Implementation Details. ....	24
4.1. Implementing Planning. ....	24
4.2. User Training. ....	24
4.3. Data Transmission. ....	25

<b>5. Testing.</b>	<b>26</b>
<b>5.1. Functional Testing.</b>	<b>26</b>
<b>5.2. Stress Testing.</b>	<b>26</b>
<b>5.3. Performance Testing.</b>	<b>26</b>
<b>5.4. Structural Testing.</b>	<b>27</b>
<b>5.5. White Box Testing.</b>	<b>27</b>
<b>5.5.1. Control Structure testing</b>	<b>28</b>
<b>5.6. Data Flow Testing.</b>	<b>28</b>
<b>5.7. Loop Testing.</b>	<b>28</b>
<b>5.8. Black Box Testing.</b>	<b>28</b>
<b>6. Conclusion And Future Outlook.</b>	<b>30</b>
<b>7. References.</b>	<b>31</b>
<b>8. Appendices.</b>	<b>32</b>

# **INTRODUCTION**

## 1. INTRODUCTION

### 1.1 COMPANY PROFILE



Global software limited (gsl) is promoted by an industrial group with a proven track record in the IT arena. It is a technology focused multinational company that focuses on providing contemporary ESM solutions anchored by quality.

With a world class research and development center for Operating systems, Databases, Networks and Enterprise Security, GSL provides ESM Solutions from min-size to large organizations, portals and ISPs.

The company has offices in US, UK, SINGAPORE, MAURITIUS and INDIA with 150 experienced ESM professionals. It is backed by a core technology team with 100+ man-years of experience in systems software and networking technologies.

Global Software Ltd., India is a backend system integration company, focusing on Enterprise Systems Management (ESM).

The Company has excellent resources to offer the entire range of Backend Systems Integration Services in IT with specialization in the following areas.

- Managed Service
- Operating Systems Services
- Networking Services
- ESM Solution Consulting



## INVENTORY CONTROL SYSTEM

Our company is unique with 100% certified, thoroughly experience, highly qualified professionals offering tangible, scalable ESM solutions to achieve increase in service deliverables, sound knowledge and vast experience to handle heterogeneous complexity of multiple operating systems.

With its unique Competency Center (the only one of its kind in Asia pacific), Global's Competency center- Hardware and Software

GLOBAL Continually updates the competency center in line with market changes. The ESM Competency center in Chennai is a true world-class infrastructure with stare-of the –art equipments.

### Hardware

The Following are the wide range of Hardware available at GLOBAL at India.

- IBM S/390 Enterprise Server
- IBM RS/600 SP@Enterprise server with SAN
- Sun Enterprise Server – 3500 Series
- IBM Netfinity 5500 servers
- CISCO Recruiters and Switches

# INVENTORY CONTROL SYSTEM

## 1.2. EXISTING SYSTEM

In the existing system, all the processing are done manually, which asks for a very high amount of manual labor. It also tends to indulge errors due to the human factors. The time consumed for calculations and production of reports is wasted much.

The items to be purchased for manufacturing are not identified at the right time. The purchase reorder level is not analyzed. The maximum needed item for the manufacturer is not analyzed. The non-moving manufacturer items and the maximum moving manufacturer items to be analyzed are difficult by manually. During manual entry transaction error occurs. Project wise material consumed and other details are not analyzed.

## DRAWBACKS OF THE EXISTING SYSTEM

The existing system suffers from many drawbacks, which is quite natural to any manual system. Some of the demerits are:

- ❑ Time Consumption
- ❑ Error Factor
- ❑ Repetitiveness
- ❑ Bulk volume of data cannot be handled
- ❑ Storage of data

## 1.3. PROPOSED SYSTEM

In the proposed system, the whole thing is computerized using mainframe system which handles the transaction entry, master updates, history ledger entry, analyzing non-moving item and maximum moving item, indication of a particular item to be manufacture

The proposed system is designed keeping in mind the inevitable necessity of upgrading to the future technologies.

### **Advantages of the proposed system**

The proposed system has many advantages, some of which are mentioned below

- High response time
- Cost effective solution
- Ease of deployment
- High data security
- Huge data storage
- Scalability
- Possibility of upgrading
- Wider reach ability.

## INVENTORY CONTROL SYSTEM

- ❑ **Difficulty in accessing Data**
- ❑ **Data isolation**
- ❑ **Integrity problems**
- ❑ **Security issues**

## **1.4. RELEVANCE AND IMPORTANCE**

The Inventory control system is very big process. Here security is very important one. In the proposed system only the allowed user can enter the transaction entry.

Here database plays major role, which should be given high security. TO keep the details of 12000 items are easy and accessing the information is also easy. Database that handles large volume of data has to be used.

Since it is factory at the maximum only hundred users will be there, so accessing speed is not a matter, since it is a manufacturing company there may be more branches and many users who will access the database from anywhere in the globe. The software should be developed such that it should provide a very high access speed for the users.

## **2. SOFTWARE REQUIREMENTS**

The software required for this project is an IBM product. IBM technology supports Mainframe much more than other technologies

### **2.1 SOFTWARE SPECIFICATION**

- OPERATING SYSTEM : OS/390
- VERSION : 2.8
- LANGUAGE : JCL, VS COBOL II
- BACKEND : VSAM

# **SOFTWARE REQUIREMENTS**

## INVENTORY CONTROL SYSTEM

### **2.3 SPECIFIC REQUIREMENT:**

#### **2.3.1 FUNCTIONAL REQUIREMENTS:**

The Inventory control system involves calculating monthly Transaction processing, updating of master file, project wise material consumed, validation of the transaction process are processed.

##### List Of Inputs:

- Raw materials.
- Manufacture details.
- Transaction list.
- Project Details.

#### **RAW MATERIALS:**

The purchase is made only if it reaches the reorder specified in the master table. Each time when a purchase is performed it is updated in the master file using PS (physical sequential) file as input file.

Purchase of the item made and their details, the quantity available in the stock and the description about that item, the rate of that item, unit of measurement for that item are entered into to master purchase table.



## 2.3.2 PERFORMANCE REQUIREMENTS

### **Security**

This system is highly secured and any unauthorized person cannot access the system. Since we are using Mainframe the database is highly secured. Database is efficiently maintained. Only the authorized users can access the database. Whatever the virus is and how powerful it is the Mainframe will ditch off.

### **Capacity**

Mainframe stands for its access speed. More than 200 users can access the database at the same time. No other technology can provide such access. 90% of transaction will be processed within 2 seconds.

### **Availability**

Any authorized user can access the database using a common PC with a mainframe connection.

## INVENTORY CONTROL SYSTEM

### **MANUFACTURE DETIALS:**

This contains manufacturing details of a particular item. In this the input is from sales module. If the item in sales reaches the reorder specified in the Inventory master file then that particular item should be manufactured.

### **TRANSACTION LIST:**

Every transaction file containing transaction details are posted in to the History ledger and updating the corresponding master file. This is the input PS file for the Inventory master file.

In this the details of the item for sales and their quantity in stock, their description and other details about that item are included. In this the transaction type can be of four types. They are receipt, issue, transfer-in, transfer-out and write-off.

### **PROJECT DETIALS:**

In this the item consumed for a particular project are analyzed from the History ledger module. The material consumed from other project and their details are optioned.

# **SYSTEM DESIGN AND DEVELOPMENT**

## **3. SYSTEM DESIGN AND DEVELOPMENT**

Design is essentially creative activity does not mean that it consists simply have a series of bright ideas. Design requires a full understanding of the problem. There is need for analysis of the requirements and resources. The acceptable design is likely to compromise between the number of factors: particularly costs, reliability, accuracy, security, control, integration, expandability, availability and acceptability.

### **DATA BASE DESIGN**

The database has been carefully designed based on the needs and requirements of various designations in "Inventory control system". The details of the applicant given in the application form are also maintained for the workflow implementation.

The time factor involved during the transaction from one designation to higher designation is also noted in the database so higher official can identify that pending transaction and actions could be taken. The applicant will in the future use this for status tracking of any particular transaction.

# INVENTORY CONTROL SYSTEM

## 3.1. MODULE DESCRIPTION

The “INVENTORY CONTROL SYSTEM” consists the following modules:

- Purchase module.
- Manufacture module.
- Sales module.
- History ledger module.

### **PURCHASE MODULE:**

In this module the purchase of raw material used for manufacture is included. This module includes all the details such as purchase item description, rate, unit of measurement for that item, reorder level for that item and the available quantity in the stock.

The purchase is made only in it reaches the reorder level of the purchase file. The item is consumed for the manufacturing. The quantity is updated if that particular is consumed for manufacture.

Every time the purchase made is entered in to master file using a transaction file (Ps file). Every time the transaction is validated using the unit of measurement field. The master file is a KSDS (key sequential data set) and the transaction file is Ps (physical sequential) file.

### **MANUFACTURE MODULE:**

This module contains details of a particular item. To manufacture a particular item their needed items, the ratio and other details are stored in manufacture master file. The items consumed to manufacture a particular item are updated in the purchase master file.

## **INVENTORY CONTROL SYSTEM**

After a particular item is manufactured it is updated in the inventory master file. The manufactured of item is performed depending on that item moving in the sales. A particular item is manufactured only if the quantity in stock of inventory file reaches the reorder level.

### **SALES MODULE:**

In this it contains details of transaction for a particular items. It contains the details like item description, unit of measurement, quantity in stock, rate, last issue date, year open balance, item open date, yearly receipt quantity, yearly consume quantity and other details.

The transaction file is validated every time using unit of measure field. In this there are five Transaction type. They are the following:

#### **RECEIPT:**

The transaction type is receipt only if the item sold is for cash. The transaction value is entered only in the case of receipt.

#### **ISSUE:**

In this the transaction of particular item made with credit. This issue can be for project development for the same company or for others also.

#### **TRANSFER-IN:**

A particular item is consumed for a project from another project. For e.g. project1 consume a material from project2. Then that items transaction type is transfer-in to project1. It affects both the projects.

## INVENTORY CONTROL SYSTEM

### TRANSFER-OUT:

A particular item is transferred to a project from one project. For e.g. project1 transfer an item to project2. Then that item transaction type is transfer-out to project1. It affects both the projects. Then the type of transaction for project2 is Transfer-in.

### WRITE-OFF:

This means a particular item is decomposed.

In these non-moving items, maximum-moving items, items at reorder level and other are performed in this module.

### HISTORY LEDGER MODULE:

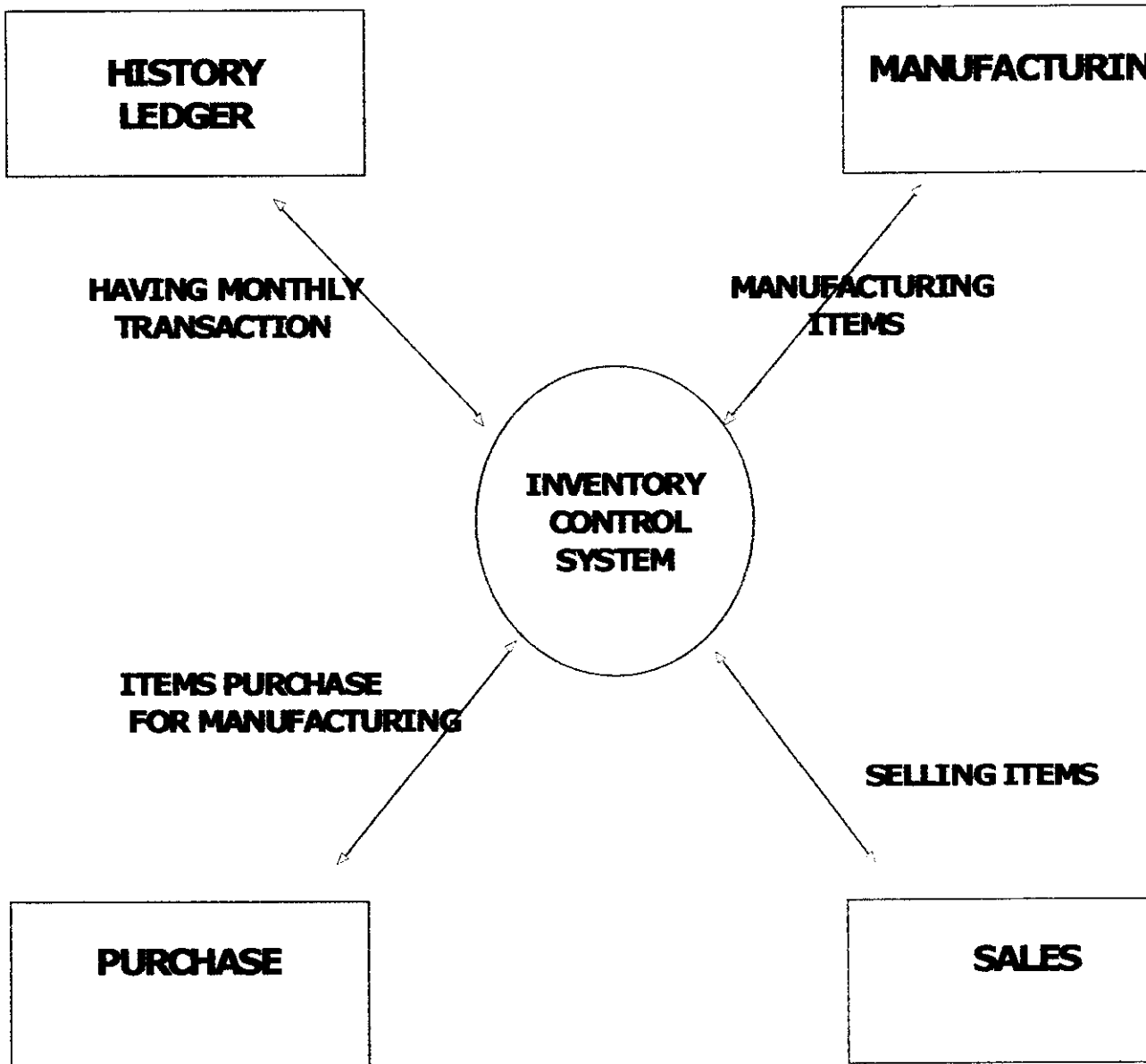
The monthly transaction of inventory master file is stored in the ledger file. A particular month transaction can be obtained in this module. The project wise material consumed can be viewed using this.

## 3.2. INTEGRATION OF MODULES

All modules are integrated with one another as well as with Inventory master file. The purchase module is integrated with the manufacture module. The manufacture module is linked with both purchase and Inventory maser file. The Inventory master file is integrated with history ledger file.

# INVENTORY CONTROL SYSTEM

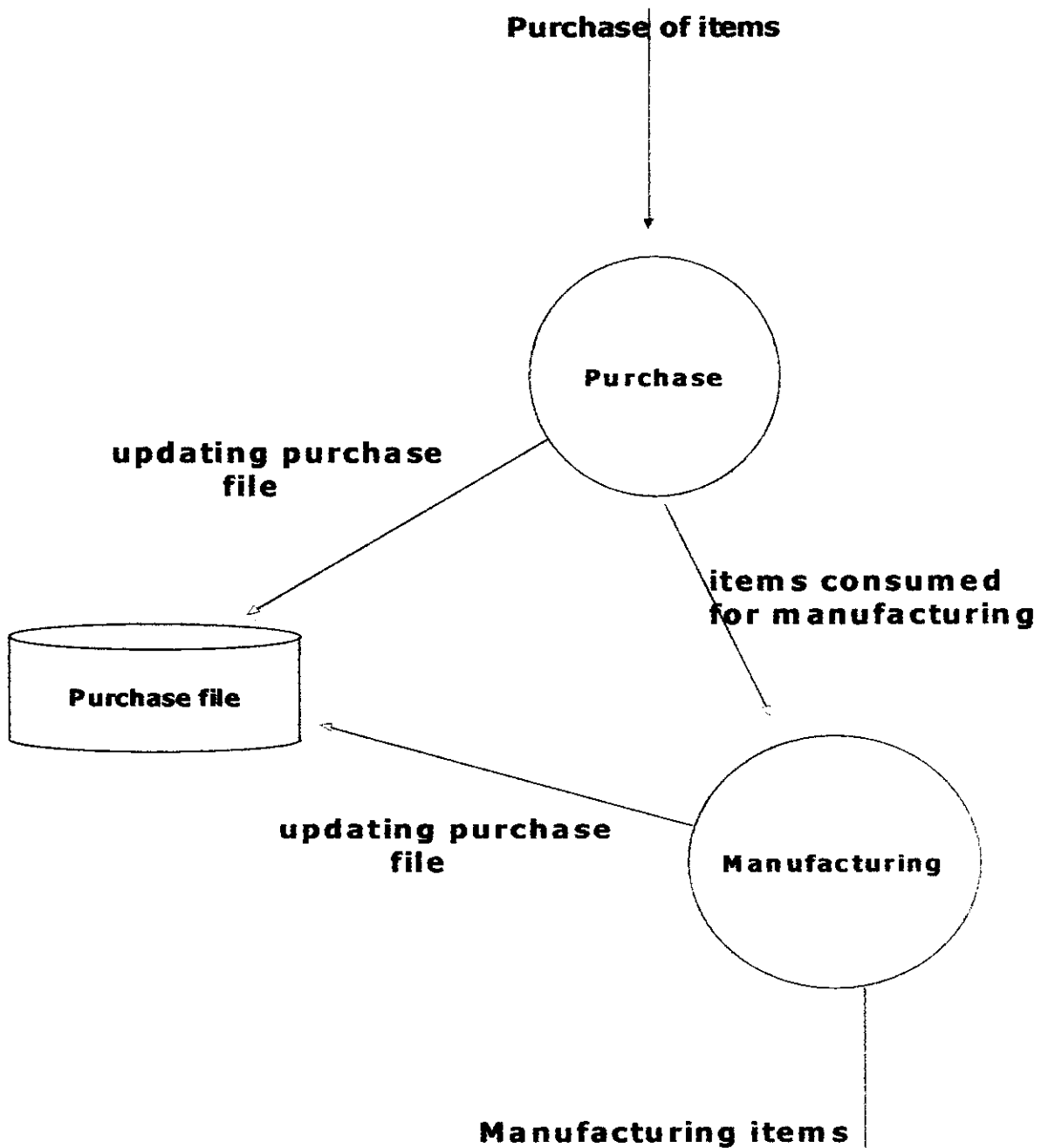
## 3.3. CONTEXT ANALYSIS DIAGRAM





3.4 DATA FLOW DIAGRAM:

**PURCHASE PROCESS:**



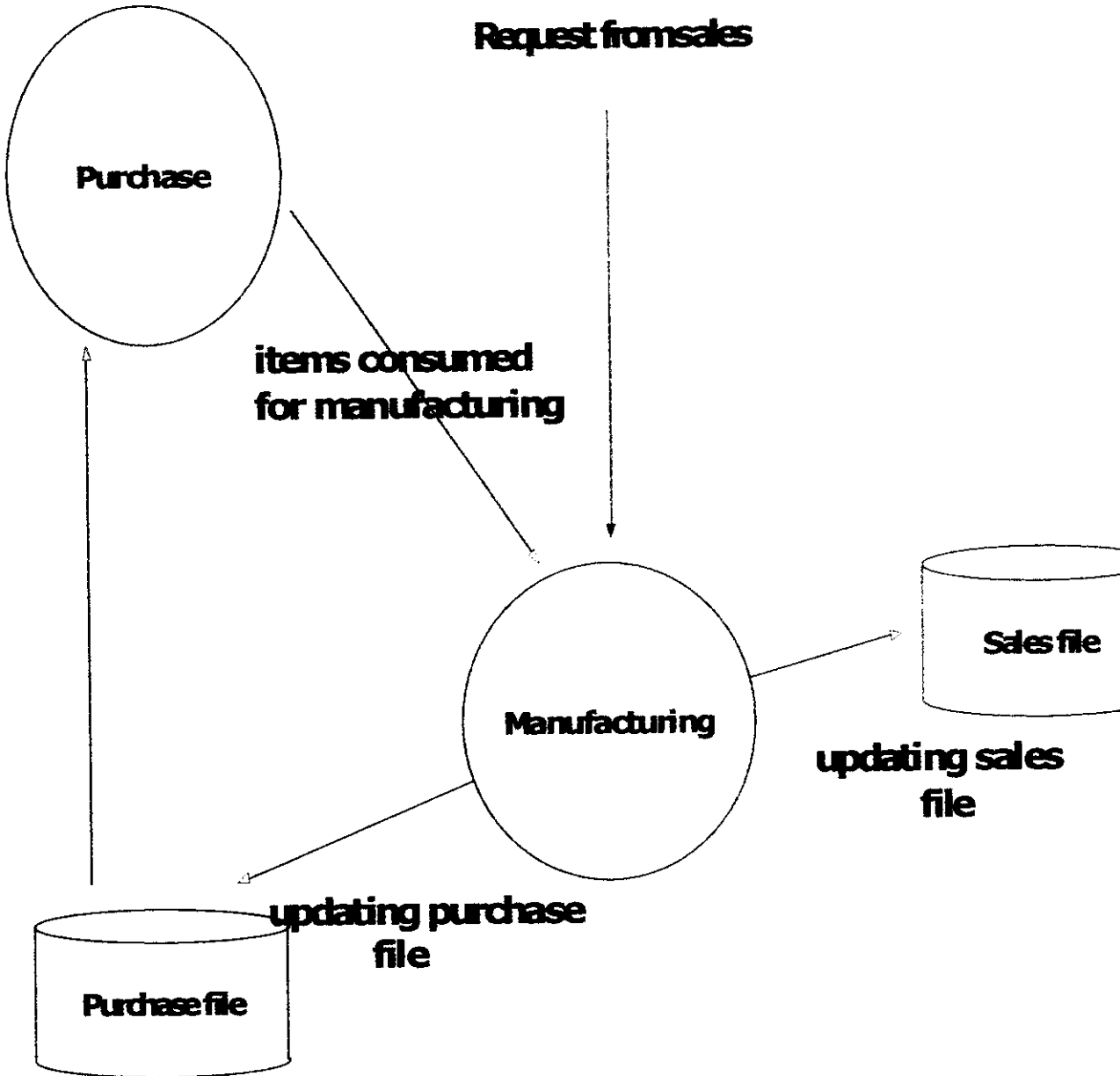
# INVENTORY CONTROL SYSTEM

## 2.2 HARDWARE SPECIFICATION

The hardware specification listed is on minimum basis for optimum performance.

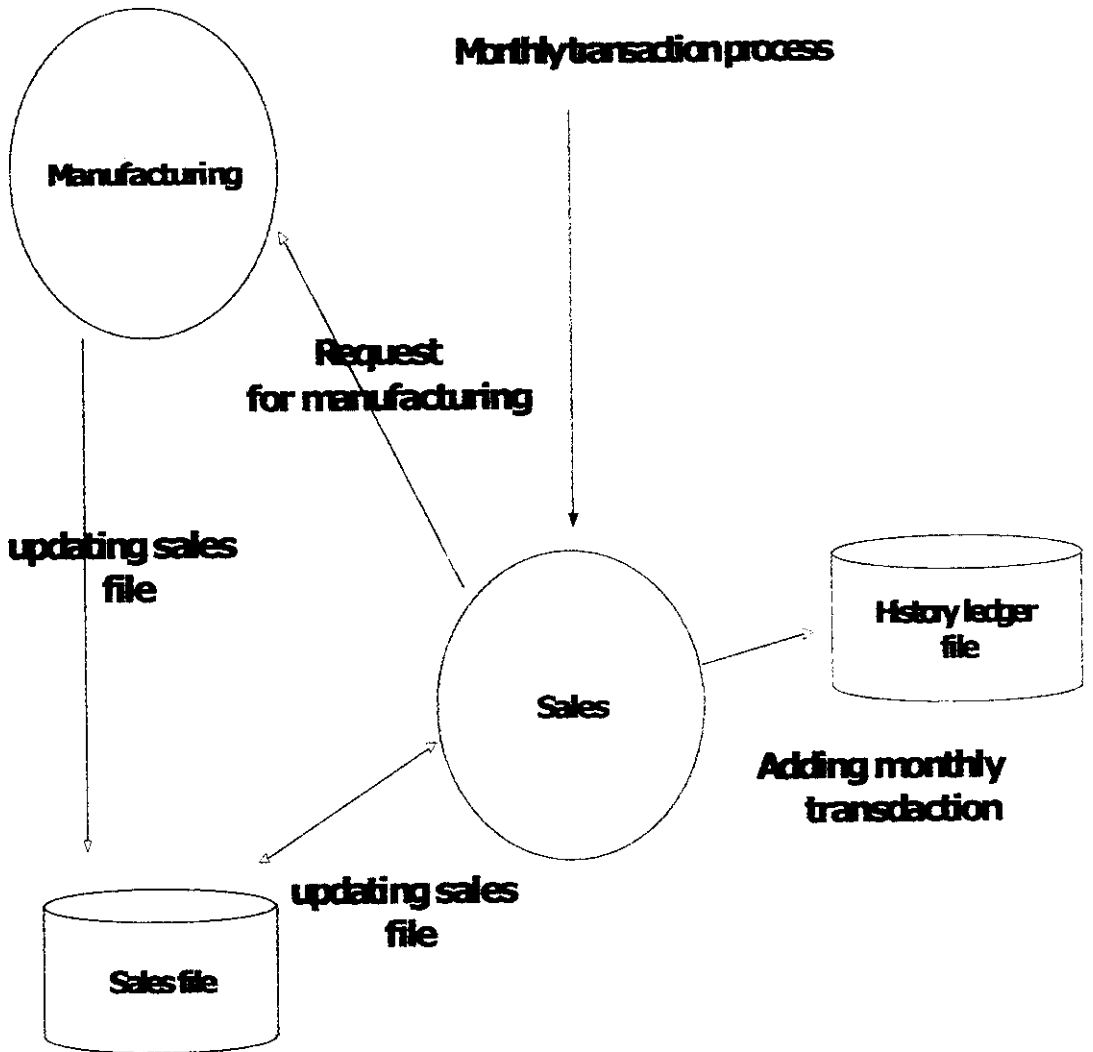
- PROCESSOR : IBM IS 3006
- MODEL : B01
- HARD DISK : 78 GB
- MONITOR : IBM SDT 3270
- MIPS : 100  
(Millions Instructions  
Per Second)

**MANUFACTURE PROCESS:**



# INVENTORY CONTROL SYSTEM

## SALES PROCESS:



### 3.5. FILE STRUCTURE:

#### MASTER TABLE

Purchase master file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
→ pu-item	Varchar	5	✓	Purchase item- code	Pk-1 of 1
Pu-desc	VarChar	25	×	Description of item name	
puom	Char	2	×	Unit of measurement	
pqis	Numeric	5	×	Quantity in stock	
prol	Numeric	5	×	Purchase reorder level	
prate	Numeric	7	×	Purchase rate	

Manufacture master file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
→ manu-code	Varchar	5	✓	Manufacture item code	pk-5 of 5
Manu-desc	VarChar	50	✓	Description of manufacturing itemUser name	
Manu-rate	Numeric	7	✓	Manufacture rate	
Manu-qty	Numeric	5	✓	Manufacture Quantity	
Manu-uom	Varchar	2	✓	Unit of measurement	

Inventory master file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
------------	------	-------------	----------	-------------	---------------------

# INVENTORY CONTROL SYSTEM

## TRANSACTION TABLE:

Transaction purchase file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
⑧ item-code	Varchar	5	✓	Purchase item code	Pk-1of 1
Uom	VarChar	2	×	Purchase unit of measurement	
Pu-qty	Numeric	5	×	Purchase quantity	

Transaction manufacture file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
⑧ manu-code	Varchar	5	✓	Manufacture code	Pk11 of 1
Manu-qty	Numeric	5	×	Manufacture quantity	
Manu-uom	VarChar	2	×	Unit of measurement	

Transaction Inventory master file and history ledger file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
⑧ item-code	Varchar	5	✓	Item code	Pk11 of 1
Trans-type	Numeric	1	×	Transaction type	
Trans-no	Numeric	4	×	Transaction number	
Trans-date	Numeric	8	×	Transaction date	
uom	VarChar	2	×	Unit of measurement	

# INVENTORY CONTROL SYSTEM

⚡ item-code	Varchar	5	✓	Item code	Pk-1 of 1
Item-desc	VarChar	25	x	Item description	
uom	VarChar	2	x	Unit of measurement	
Qis	Numeric	5	x	Quantity in stock	
Rate	Numeric	7	x	Cost of item	
Lid	Numeric	8	x	Last issue date	
Yob	Numeric	5	x	Year open balance	
Iod	Numeric	8	x	Item open date	
Yrq	Numeric	5	x	yearly receipt quantity	
Ycq	Numeric	5	x	Yearly consume quantity	
Lc	Varchar	1	x	Location code	
Rol	Numeric	5	x	Sales reorder level	

## History ledger file:

Field name	Type	Used Length	Not Null	Description	Primary Key Details
⚡ trans-no	Numeric	4	✓	Transaction number Item code	Pk- 1of 2
Item-code	Varchar	5	x	Item code	
Item-desc	VarChar	25	x	Item description	
⚡ project-id	Varchar	5	✓	Project-identification key	Pk-2 of 2
Trans-type	Numeric	1	x	Transaction type	
Trans-qty	Numeric	5	x	Transaction quantity	
Trans-date	Numeric	8	x	Transaction date	
Trans-value	Numeric	7	x	Transaction value	

# INVENTORY CONTROL SYSTEM

Trans-qty	Numeric	5	x	Transaction quantity
Trans-value	Numeric	7	x	Transaction value only in case of receipt
Project-id	Varchar	5	x	Project-identification key



# **IMPLEMENTATION DETAILS**

## **4. IMPLEMENTATION DETAILS**

Implementation is the stage of the project when the theoretical design is turned in to a system. This is the crucial phase in the system life cycle. Implementation means putting a new system design into operation. Following steps are considered in the implementation stage:

- Implementation Planning
- User Training
- Data Transmission

### **4.1. Implementation Planning:**

This planning is a logical starting point to manage different activities that must be covered. A pre-implementation meeting with the personals from all departments is arranged.

### **4.2. User Training:**

Hands-on training to user is essential to make them comfortable with the system. Accordingly, two or three day demonstration and practical training with the past data are to be given to the users of the system.

**TESTING**

## **5. TESTING**

Testing is a vital process to the success of any system. At first, the system is tested to see whether it produces correct outputs. Then, the system is tested for volume of transactions, stress and recovery from failure and usability.

### **5.1. Functional testing**

Functional testing is performed to specify the operating conditions, input values and expected results. All the functions in the system are tested with required parameters.

### **5.2. Stress testing**

The purpose of stress testing is to determine the limitation of the system. In this system, stress testing is performed to identify whether the package is able to handle the entire abnormal situation.

### **5.3. Performance testing**

Performance testing is done with this system to verify the response time, execution time, throughput, and primary and secondary memory utilization and traffic rate on data channel and communication links.

## **5.4. Structural testing**

Structural testing is performed to examine the internal processing logic of the system in each and every phase.

## **5.5. White Box testing:**

White box testing is done with the system, which derives test cases that do the following:

- Guarantee that all the independent paths within a module have been exercised at least once in the package.
- Exercise all logical decisions on their true and false sides.
- Execute all loops at their boundaries and within their operation bounds.
- Exercise internal data structures to ensure their validity.

One of the test case tools of white box testing is control structure testing.

### **5.5.1. Control Structure testing:**

These test case design exercises the logical conditions contained in a program module of the package.

### **5.6. Data flow testing:**

The data flow testing method selects test paths of program according to the locations of definitions and uses of variables in the package.

### **5.7. Loop testing**

Loops are cornerstones for the vast majority of all algorithms implemented in software. Loop testing is done with the system that focuses exclusively on the validity of loop constructions.

### **5.8. Black box testing:**

Black box testing method focuses on the functional requirements of the software. Using the black box testing method, the following errors are identified and rectified in the package.

- Incorrect or missing functions
- Interface errors
- Errors in data structures or external databases access
- Performance errors
- Initialization and termination errors.

## **INVENTORY CONTROL SYSTEM**

**Using the above testing procedures, the “Inventory control system” has been validated and the outcome of the test was in accordance with the requirements of the Management.**

# **CONCLUSIONS AND FUTURE OUTLOOK**



## INVENTORY CONTROL SYSTEM

### **4.3. Data Transmission:**

The initially data is entered into PS (physical sequential) file. And then data are populated from PS file to the master files using COBOL coding or Reprokes by JCL coding and then manipulation of the master file is performed.

## **6. CONCLUSIONS AND FUTURE OUTLOOK**

The project is implemented keeping in mind the possible future enhancements and the modules are designed in such a way that enhancements are possible without any change in the basic structure of the system.

The database design has provisions for enhancements. The relationships between data are well-defined and key columns identified so that addition of new tables to the system in future can be done with ease.

Provisions for including session and transaction files are given to handle transaction processing in future. The file designs as well as the program logic are done bearing possible future changes and enhancements in mind.

The system is developed in a self-documentary way, which would help any programmer to analyze it and incorporate enhancements to it.

## **REFERENCE**

## **7. REFERENCE**

### **BOOKS:**

- Zamer Ranade, **MVS JCL Primer - VI Edition**, Tata Mc Graw Hill Company Ltd, year 2000.
- Nancy Stern & Robert .A. Stern, **Structured COBOL Programming – VII Edition**, JhonWilley Publication, Year 2001.
- Roy & Dastidae, **COBOL Programming – II Edition**, Mc Graw Hill Company Ltd, Year 1989.

### **WEBSITES:**

[www.mvshelp.com](http://www.mvshelp.com)

[www.mainframeforum.com](http://www.mainframeforum.com)

# **APPENDIX**

## 8. APPENDIX

### SAMPLE CODE:

### JCL CODE TO RUN A COBAL PROGRAM:

```
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>      your edit profile using the command RECOVERY ON.
000100 //TRG079A JOB ,NOTIFY=&SYSUID,PRTY=15
000200 // JCLLIB ORDER=(TRG079.COB.PGM)
000300 //STEP1 EXEC PROC=COBCLG11,MEM=PRG456
000400 //SYSPRINT DD SYSOUT=*
000500 //COB.SYSIN DD
DSN=TRG079.PROJECT.COB(&MEM),DISP=SHR
000600 //GO.INFILE DD DSN=TRG079.PROJECT.COOB,DISP=SHR
000700 //GO.OUTFILE DD DSN=TRG079.KSDS.INMF,DISP=SHR
000800 //GO.OFILE DD DSN=TRG079.KSDS.HISTORY,DISP=SHR
000900 //GO.OFILE1 DD DSN=TRG079.KSDS.PATH,DISP=SHR
001000 //GO.SYSIN DD DUMMY
001100 /*
001200 //
***** Bottom of Data *****
```

# INVENTORY CONTROL SYSTEM

## JCL CODE TO CREATE A KSDS FILE:

```
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you change
==MSG>      your edit profile using the command RECOVERY ON.
000100 //TRG079A JOB ,,NOTIFY=TRG079,MSGCLASS=X
000200 //DEFKSDS EXEC PGM=IDCAMS
000300 //SYSPRINT DD SYSOUT=*
000400 //SYSIN DD *
000500 DEFINE CLUSTER(NAME(TRG079.KSDS.HISTORY)      -
000600 VOLUMES(USR003)                                -
000700 TRACKS(5,5)                                    -
000800 CONTROLINTERVALSIZE(4096)                    -
000900 RECORDSIZE(80,80)                            -
001000 KEYS(4,0)                                      -
001100 FREESPACE(10,20)                              -
001200 INDEXED)                                     -
001300 DATA(NAME(TRG079.KSDS.HISTORY.DATA))      -
001400 INDEX(NAME(TRG079.KSDS.HISTORY.INDEX))
001500 /*
001600 //
*** ***** Bottom of Data *****
```

## INVENTORY CONTROL SYSTEM

### JCL CODE TO CREATE ALTERNATIVE INDEX:

\*\*\*\*\* Top of Data \*\*\*\*\*

==MSG> -Warning- The UNDO command is not available until you change

==MSG> your edit profile using the command RECOVERY ON.

000100 //TRG079A JOB ,,NOTIFY=TRG079,MSGCLASS=X,PRTY=15

000200 //DEFESDS EXEC PGM=IDCAMS

000300 //SYSPRINT DD SYSOUT=\*

000400 //SYSIN DD \*

000500 DEFINE ALTERNATEINDEX (NAME(TRG079.KSDS.ALT) -

000510 RELATE(TRG079.KSDS.HISTORY) -

000600 VOLUMES(USR001) -

000700 TRACKS(3,1) -

000800 CONTROLINTERVALSIZE(4096) -

000900 RECORDSIZE(80,80) -

001000 KEYS(5,9) -

001100 FREESPACE(10,20) -

001200 NONUNIQUEKEY -

001210 UPGRADE) -

001300 DATA(NAME(TRG079.KSDS.ALT.DATA)) -

001400 INDEX(NAME(TRG079.KSDS.ALT.INDEX))

001500 /\*

001600 //

\*\*\*\*\* Bottom of Data \*\*\*\*\*

### JCL CODE TO CREATE A BUILD INDEX:

\*\*\*\*\* Top of Data \*\*\*\*\*

==MSG> -Warning- The UNDO command is not available until you change

==MSG> your edit profile using the command RECOVERY ON.

000010 //TRG079A JOB ,,NOTIFY=TRG079,MSGCLASS=X



# INVENTORY CONTROL SYSTEM

```
000020 //STEP1 EXEC PGM=IDCAMS
000030 //SYSPRINT DD SYSOUT=*
000040 //SOURCE DD DSN=TRG079.KSDS.HISTORY,DISP=SHR
000050 //TARGET DD DSN=TRG079.KSDS.ALT,DISP=SHR
000060 //SYSIN DD *
000070   BLDINDEX -
000080   INFILE(SOURCE) -
000090   OUTFILE(TARGET)
000100 /*
000200 //
***** Bottom of Data *****
```

## JCL CODE TO SET THE PATH FOR ALTERNATIVE INDEX:

```
***** Top of Data *****
==MSG> -Warning- The UNDO command is not available until you
change
==MSG>   your edit profile using the command RECOVERY
ON.
000010 //TRG079A JOB
,.,NOTIFY=TRG079,MSGCLASS=X,PRTY=15
000020 //STEP1 EXEC PGM=IDCAMS
000030 //SYSPRINT DD SYSOUT=*
000060 //SYSIN DD *
000070   DEFINE PATH -
000080   (NAME(TRG079.KSDS.PATH)-
000090   PATHENTRY(TRG079.KSDS.ALT) -
000091   UPDATE)
000092 /*
000100 //
***** Bottom of Data *****
```

# INVENTORY CONTROL SYSTEM

```
003000      02 ROL      PIC 9(5).
003100      02 F        PIC X(19).
003200      FD OUT-KSDS-FILE.
003300      01 OUT-KSDS-REC.
003400      02 ITEM-CODE  PIC X(5).
003500      02 ITEM-DESC  PIC X(25).
003600      02 UOM        PIC XX.
003700      02 QIS        PIC 9(5).
003800      02 RATE      PIC 9(5)V9(2).
003900      02 LID       PIC 9(8).
004000      02 YOB       PIC 9(5).
004100      02 IOD       PIC 9(8).
004200      02 CYRQ     PIC 9(5).
004300      02 CYCQ     PIC 9(5).
004400      02 LC        PIC X.
004500      02 ROL      PIC 9(5).
004600      02 F        PIC X(19).
004700      WORKING-STORAGE SECTION.
004800      01 EOF      PIC 9 VALUE 0.
004900      77 PS-STATUS PIC X(2).
005000      77 KSDS-STATUS PIC X(2).
005100      PROCEDURE DIVISION.
005200          PERFORM OPEN-PARA
005300          PERFORM READ-PARA
005400          PERFORM PROCESS-PARA UNTIL EOF = 1
005500          PERFORM CLOSE-PARA.
005600      OPEN-PARA.
005700          OPEN INPUT IN-PS-FILE
005800          IF PS-STATUS NOT = '00'
005900              DISPLAY 'PS FILE OPENING ERROR ' PS-STATUS
006000              PERFORM CLOSE-PARA
006100          ELSE
006200              DISPLAY 'PS FILE OPENING SUCCESSFULL'
006300          END-IF
006400          OPEN OUTPUT OUT-KSDS-FILE
006500          IF KSDS-STATUS NOT = '00'
006600              DISPLAY 'KSDS FILE OPENING ERROR ' KSDS-STATUS
006700              PERFORM CLOSE-PARA
006800          ELSE
```

# INVENTORY CONTROL SYSTEM

## COBOL PROGRAM TO UPDATE THE MASTER FILE:

\*\*\*\*\* Top of Data \*\*\*\*\*

```
000100    *THIS IS A PROGRAM TO ACCESS KSDS FILE RANDOMLY
AND DO ALL KIND
000200    *OF MANIPLUATIONS
000300    IDENTIFICATION DIVISION.
000400    PROGRAM-ID. RANDOMAC.
000500    ENVIRONMENT DIVISION.
000600    INPUT-OUTPUT SECTION.
000700    FILE-CONTROL.
000800        SELECT IN-PS-FILE ASSIGN TO INFILE
000900        FILE STATUS PS-STATUS.
001000        SELECT OUT-KSDS-FILE ASSIGN TO OUTFILE
001100        ORGANISATION INDEXED
001200        ACCESS RANDOM
001300        RECORD KEY ITEM-CODEK
001400        FILE STATUS KSDS-STATUS.
001500    DATA DIVISION.
001600    FILE SECTION.
001700    FD IN-PS-FILE.
001800    01 IN-PS-REC.
001900        02 CHANGECODE PIC X.
002000        88 CHANGECHK VALUE 'R' 'W' 'U' 'D'.
002100        02 ITEM-CODE  PIC X(5).
002200        02 ITEM-DESC  PIC X(25).
002300        02 UOM        PIC XX.
002400        02 QIS        PIC 9(5).
002500        02 RATE       PIC 9(5)V9(2).
002600        02 LID        PIC 9(8).
002700        02 YOBS       PIC 9(5).
002800        02 IOD        PIC 9(8).
002900        02 CYRQ       PIC 9(5).
```

# INVENTORY CONTROL SYSTEM

003000        02 CYCQ        PIC 9(5).  
003100        02 LC            PIC X.  
003200        02 ROL            PIC 9(5).  
003300        02 F              PIC X(18).  
003400        FD OUT-KSDS-FILE.  
003500        01 OUT-KSDS-REC.  
003600        02 ITEM-CODEK    PIC X(5).  
003700        02 ITEM-DESC    PIC X(25).  
003800        02 UOM            PIC XX.  
003900        02 QIS            PIC 9(5).  
004000        02 RATE           PIC 9(5)V9(2).  
004100        02 LID            PIC 9(8).  
004200        02 YOB            PIC 9(5).  
004300        02 IOD            PIC 9(8).  
004400        02 CYRQ           PIC 9(5).  
004500        02 CYCQ           PIC 9(5).  
004600        02 LC            PIC X.  
004700        02 ROL            PIC 9(5).  
004800        02 F              PIC X(19).  
004900        WORKING-STORAGE SECTION.  
005000        77 EOF            PIC 9 VALUE 0.  
005100        77 PS-STATUS    PIC X(2).  
005200        77 KSDS-STATUS PIC X(2).  
005300        PROCEDURE DIVISION.  
005400            PERFORM OPEN-PARA  
005500            PERFORM READ-PARA  
005600            PERFORM PROCESS-PARA UNTIL EOF = 1  
005700            PERFORM CLOSE-PARA.  
005800        OPEN-PARA.  
005900            OPEN INPUT IN-PS-FILE  
006000            IF PS-STATUS NOT = '00'  
006100                DISPLAY 'PS FILE OPENING UNSUCCESSFULL' PS-  
STATUS  
006200                PERFORM CLOSE-PARA  
006300            ELSE  
006400                DISPLAY 'PS FILE OPENING SUCCESSFULL'  
006500            END-IF.  
006600            OPEN I-O OUT-KSDS-FILE  
006700            IF KSDS-STATUS NOT = '00'

INVENTORY CONTROL SYSTEM

```
006800      DISPLAY 'KSDS FILE OPENING UNSUCCESSFULL'
KSDS-STATUS
006900      PERFORM CLOSE-PARA
007000      ELSE
007100      DISPLAY 'KSDS FILE OPENING SUCCESSFULL'
007200      END-IF.
007300      READ-PARA.
007400      READ IN-PS-FILE AT END MOVE 1 TO EOF PERFORM
CLOSE-PARA.
007500      IF PS-STATUS NOT = '00'
007600      DISPLAY 'PS FILE READING UNSUCCESSFULL' PS-
STATUS
007700      PERFORM READ-PARA
007800      ELSE
007900      DISPLAY 'PS FILE READING SUCCESSFULL'
008000      END-IF.
008100      PROCESS-PARA.
008200      IF NOT CHANGECHK
008300      DISPLAY 'INVALID CHANGE CODE!! CURRENT INPUT
RECORD WAS
008400      - 'DISCARDED'
008500      PERFORM READ-PARA
008600      ELSE
008700      DISPLAY 'PS FILE PROCESSING SUCCESSFULL'
008800      PERFORM PROCESS-KSDS-PARA
008900      END-IF.
009000      PROCESS-KSDS-PARA.
009100      IF CHANGECODE = 'R'
009200      PERFORM READ-KSDS-PARA
009300      ELSE
009400      IF CHANGECODE = 'W'
009500      PERFORM WRITE-KSDS-PARA
009600      ELSE
009700      IF CHANGECODE = 'U'
009800      PERFORM UPDATE-KSDS-PARA
009900      ELSE
010000      PERFORM DELETE-KSDS-PARA
010100      END-IF
010200      END-IF
```

# INVENTORY CONTROL SYSTEM

```
010300     END-IF.
010400     READ-KSDS-PARA.
010500     MOVE ITEM-CODE TO ITEM-CODEK
010600     READ OUT-KSDS-FILE
010700     IF KSDS-STATUS NOT = '00'
010800         DISPLAY 'KSDS FILE READING UNSUCCESSFULL'
KSDS-STATUS
010900     DISPLAY 'THE REQUESTED RECORD WAS NOT FOUND'
011000     ELSE
011100         DISPLAY 'THE REQUESTED RECORD IS '
011200         DISPLAY OUT-KSDS-REC
011300     END-IF
011400     PERFORM READ-PARA.
011500     WRITE-KSDS-PARA.
011600     MOVE ITEM-CODE TO ITEM-CODEK
011700     MOVE CORRESPONDING IN-PS-REC TO OUT-KSDS-REC
011800     WRITE OUT-KSDS-REC
011900     IF KSDS-STATUS NOT = '00'
012000         DISPLAY 'KSDS FILE WRITING UNSUCCESSFULL' KSDS-
STATUS
012100         DISPLAY 'RECORD ALREADY EXISTS!! CANNOT
DUPLICATE'
012200     ELSE
012300         DISPLAY 'THE WRITTEN RECORD IS '
012400         DISPLAY OUT-KSDS-REC
012500         MOVE SPACES TO OUT-KSDS-REC
012600     END-IF
012700     PERFORM READ-PARA.
012800     UPDATE-KSDS-PARA.
012900     MOVE ITEM-CODE TO ITEM-CODEK
013000     READ OUT-KSDS-FILE
013100     IF KSDS-STATUS NOT = '00'
013200         DISPLAY 'KSDS FILE READING UNSUCCESSFULL'
KSDS-STATUS
013300     DISPLAY 'THE REQUESTED RECORD NOT FOUND'
013400     ELSE
013500         IF ITEM-DESC OF IN-PS-REC NOT = SPACES
013600             MOVE ITEM-DESC OF IN-PS-REC TO ITEM-DESC OF
OUT-KSDS-REC
```

# INVENTORY CONTROL SYSTEM

## COBOL PROGRAM TO POPULATE TO A MASTER FILE:

\*\*\*\*\* Top of Data \*\*\*\*\*

000100 \*THIS IS A PROGRAM TO READ INPUT FROM A PS-  
FILE IT INTO

000200 \*A KSDS-FILE(INVENTORY MASTER FILE)

000300 IDENTIFICATION DIVISION.

000400 PROGRAM-ID. PSTOKSDS.

000500 ENVIRONMENT DIVISION.

000600 INPUT-OUTPUT SECTION.

000700 FILE-CONTROL.

000800 SELECT IN-PS-FILE ASSIGN TO INFILE

000900 FILE STATUS PS-STATUS.

001000 SELECT OUT-KSDS-FILE ASSIGN TO OUTFILE

001100 ORGANISATION INDEXED

001200 ACCESS SEQUENTIAL

001300 RECORD KEY ITEM-CODE OF OUT-KSDS-REC

001400 FILE STATUS KSDS-STATUS.

001500 DATA DIVISION.

001600 FILE SECTION.

001700 FD IN-PS-FILE.

001800 01 IN-PS-REC.

001900 02 ITEM-CODE PIC X(5).

002000 02 ITEM-DESC PIC X(25).

002100 02 UOM PIC XX.

002200 02 QIS PIC 9(5).

002300 02 RATE PIC 9(5)V9(2).

002400 02 LID PIC 9(8).

002500 02 YOB PIC 9(5).

002600 02 IOD PIC 9(8).

002700 02 CYRQ PIC 9(5).

002800 02 CYCQ PIC 9(5).

002900 02 LC PIC X.

INVENTORY CONTROL SYSTEM

```
006900      DISPLAY 'KSDS FILE OPENING SUCCESSFULL'
007000      END-IF.
007100      READ-PARA.
007200      READ IN-PS-FILE AT END MOVE 1 TO EOF PERFORM
CLOSE-PARA.
007300      IF PS-STATUS NOT = '00'
007400          DISPLAY 'PS FILE READING ERROR ' PS-STATUS
007500          PERFORM READ-PARA
007600      ELSE
007700          DISPLAY 'PS FILE READING SUCCESSFULL'
007800      END-IF.
007900      PROCESS-PARA.
008000      MOVE CORR IN-PS-REC TO OUT-KSDS-REC
008100      DISPLAY OUT-KSDS-REC
008200      WRITE OUT-KSDS-REC
008300      MOVE SPACES TO OUT-KSDS-REC
008400      IF KSDS-STATUS NOT = '00'
008500          DISPLAY 'KSDS FILE WRITING ERROR ' KSDS-STATUS
008600      ELSE
008700          DISPLAY 'KSDS FILE WRITING SUCCESSFULL'
008800      END-IF
008900      PERFORM READ-PARA.
009000      CLOSE-PARA.
009100      CLOSE IN-PS-FILE
009200      IF PS-STATUS NOT = '00'
009300          DISPLAY 'PS FILE CLOSING ERROR ' PS-STATUS
009400      ELSE
009500          DISPLAY 'PS FILE CLOSING SUCCESSFULL'
009600      END-IF
009700      CLOSE OUT-KSDS-FILE
009800      IF KSDS-STATUS NOT = '00'
009900          DISPLAY 'KSDS FILE CLOSING ERROR ' KSDS-STATUS
010000      ELSE
010100          DISPLAY 'KSDS FILE CLOSING SUCCESSFULL'
010200      END-IF.
010300      STOP RUN.
```

\*\*\*\*\* Bottom of Data \*\*\*\*\*



# INVENTORY CONTROL SYSTEM

```
013700      END-IF
013800      IF UOM OF IN-PS-REC NOT = SPACES
013900          MOVE UOM OF IN-PS-REC TO UOM OF OUT-KSDS-REC
014000      END-IF
014100      IF QIS OF IN-PS-REC NOT = SPACES
014200          MOVE QIS OF IN-PS-REC TO QIS OF OUT-KSDS-REC
014300      END-IF
014400      IF RATE OF IN-PS-REC NOT = ZERO
014500          MOVE RATE OF IN-PS-REC TO RATE OF OUT-KSDS-
REC
014600      END-IF
014700      IF LID OF IN-PS-REC NOT = SPACES
014800          MOVE LID OF IN-PS-REC TO LID OF OUT-KSDS-REC
014900      END-IF
015000      IF YOB OF IN-PS-REC NOT = SPACES
015100          MOVE YOB OF IN-PS-REC TO YOB OF OUT-KSDS-REC
015200      END-IF
015300      IF IOD OF IN-PS-REC NOT = SPACES
015400          MOVE IOD OF IN-PS-REC TO IOD OF OUT-KSDS-REC
015500      END-IF
015600      IF CYRQ OF IN-PS-REC NOT = SPACES
015700          MOVE CYRQ OF IN-PS-REC TO CYRQ OF OUT-KSDS-
REC
015800      END-IF
015900      IF CYCQ OF IN-PS-REC NOT = SPACES
016000          MOVE CYCQ OF IN-PS-REC TO CYCQ OF OUT-KSDS-
REC
016100      END-IF
016200      IF LC OF IN-PS-REC NOT = SPACES
016300          MOVE LC OF IN-PS-REC TO LC OF OUT-KSDS-REC
016400      END-IF
016500      IF ROL OF IN-PS-REC NOT = SPACES
016600          MOVE ROL OF IN-PS-REC TO ROL OF OUT-KSDS-REC
016700      END-IF
016800      REWRITE OUT-KSDS-REC
016900      IF KSDS-STATUS NOT = '00'
017000          DISPLAY 'KSDS FILE UPDATING UNSUCCESSFULL'
KSDS-STATUS
017100      ELSE
```

# INVENTORY CONTROL SYSTEM

```
017200      DISPLAY 'THE UPDATED RECORD IS '  
017300      DISPLAY OUT-KSDS-REC  
017400      MOVE SPACES TO OUT-KSDS-REC  
017500      END-IF  
017600      END-IF  
017700      PERFORM READ-PARA.  
017800      DELETE-KSDS-PARA.  
017900      MOVE ITEM-CODE TO ITEM-CODEK  
018000      READ OUT-KSDS-FILE  
018100      IF KSDS-STATUS NOT = '00'  
018200      DISPLAY 'KSDS FILE READING UNSUCCESSFULL'  
KSDS-STATUS  
018300      DISPLAY 'THE REQUESTED RECORD NOT FOUND'  
018400      ELSE  
018500      DELETE OUT-KSDS-FILE  
018600      IF KSDS-STATUS NOT = '00'  
018700      DISPLAY 'KSDS FILE DELETING UNSUCCESSFULL'  
KSDS-STATUS  
018800      ELSE  
018900      DISPLAY 'THE REQUESTED RECORD IS DELETED'  
019000      END-IF  
019100      END-IF  
019200      PERFORM READ-PARA.  
019300      CLOSE-PARA.  
019400      CLOSE IN-PS-FILE  
019500      IF PS-STATUS NOT = '00'  
019600      DISPLAY 'PS FILE CLOSING UNSUCCESSFULL' PS-  
STATUS  
  
019700      ELSE  
019800      DISPLAY 'PS FILE CLOSING SUCCESSFULL'  
019900      END-IF  
020000      CLOSE OUT-KSDS-FILE  
020100      IF KSDS-STATUS NOT = '00'  
020200      DISPLAY 'KSDS FILE CLOSING UNSUCCESSFULL' KSDS-  
STATUS  
020300      ELSE  
020400      DISPLAY 'KSDS FILE CLOSING SUCCESSFULL'
```

# INVENTORY CONTROL SYSTEM

020500 END-IF

020600 STOP RUN.

\*\*\*\*\* Bottom of Data \*\*\*\*\*

## COBOL PROGRAM FOR MANIPULATION:

\*\*\*\*\* Top of Data \*\*\*\*\*

000100 \*THIS IS A PROGRAM TO MOVE TRANSACTION FILE TO  
THE MASTER

000200 \*INVENTORY FILE AND IN THE HISTORY LEDGER FILE  
AND FIND ITEMS AT

000300 \*ROL, NON MOVING STOCK AND TRANSACTION  
VALIDATION

000400 IDENTIFICATION DIVISION.

000500 PROGRAM-ID. ALTERKEY.

000600 ENVIRONMENT DIVISION.

000700 INPUT-OUTPUT SECTION.

000800 FILE-CONTROL.

000900 SELECT IN-PS-FILE ASSIGN TO INFILE

001000 FILE STATUS PS-STATUS.

001100 SELECT OUT-KSDS-FILE ASSIGN TO OUTFILE

001200 ORGANISATION INDEXED

001300 ACCESS DYNAMIC

001400 RECORD KEY ITEM-CODE OF OUT-KSDS-FILE

001500 \* ALTERNATE KEY DEPT-CODE WITH DUPLICATES

001600 FILE STATUS KSDS-STATUS.

001700 SELECT OUT-KSDS1-FILE ASSIGN TO OFILE

001800 ORGANISATION INDEXED

001900 ACCESS DYNAMIC

002000 RECORD KEY TRANS-NO OF OUT-KSDS1-FILE

002100 ALTERNATE KEY PROJ-ID OF OUT-KSDS1-FILE WITH  
DUPLICATES

002200 FILE STATUS KSDS1-STATUS.

002300 DATA DIVISION.

002400 FILE SECTION.

002500 FD IN-PS-FILE.

002600 01 IN-PS-REC.

# INVENTORY CONTROL SYSTEM

002700 02 ITEM-CODET PIC X(5).  
002800 02 TRANS-TYPE PIC 9.  
002900 88 TRANS VALUE 1 2 3 4 5 .  
003000 02 TRANS-NO PIC 9(4).  
003100 02 TRANS-DATE PIC 9(8).  
003200 02 UOM PIC XX.  
003300 02 TRANS-QTY PIC 9(5).  
003400 02 TRANS-VALUE PIC 9(5).  
003500 02 PROJ-ID PIC X(5).  
003600 02 F PIC X(45).  
003700 FD OUT-KSDS-FILE.  
003800 01 OUT-KSDS-REC.  
003900 02 ITEM-CODE PIC X(5).  
004000 02 ITEM-DESC PIC X(25).  
004100 02 UOM PIC XX.  
004200 02 QIS PIC 9(5).  
004300 02 RATE PIC 9(5)V9(2).  
004400 02 LID PIC 9(8).  
004500 02 YOB PIC 9(5).  
004600 02 IOD PIC 9(8).  
004700 02 CYRQ PIC 9(5).  
004800 02 CYCQ PIC 9(5).  
004900 02 LC PIC X.  
005000 02 ROL PIC 9(5).  
005100 02 F PIC X(19).  
005200 FD OUT-KSDS1-FILE.  
005300 01 OUT-KSDS1-REC.  
005400 02 TRANS-NO PIC 9(4).  
005500 02 ITEM-CODE PIC X(5).  
005600 02 PROJ-ID PIC X(5).  
005700 02 TRANS-TYPE PIC 9.  
005800 02 TRANS-QTY PIC 9(5).  
005900 02 TRANS-DATE PIC 9(8).  
006000 02 TRANS-VALUE PIC 9(5).  
006100 02 F PIC X(47).  
006200 WORKING-STORAGE SECTION.  
006300 77 EOF PIC 9 VALUE 0.  
006400 77 EOF1 PIC 9 VALUE 0.  
006500 77 III PIC X(5) VALUE 'AA000'.

# INVENTORY CONTROL SYSTEM

```
010000     END-IF.
010100     READPS-PARA.
010200     READ IN-PS-FILE AT END MOVE 1 TO EOF CLOSE IN-PS-
FILE.
010300     IF PS-STATUS NOT = '00'
010400         DISPLAY 'PS READING UNSUCCESSFULL' PS-STATUS
010500     ELSE
010600         DISPLAY 'PS READING SUCCESSFULL' PS-STATUS
010700         PERFORM READ-PARA1
010800     END-IF.
010900     READ-PARA1.
011000     MOVE ITEM-CODET TO ITEM-CODE OF OUT-KSDS-REC
011100     READ OUT-KSDS-FILE
011200     IF KSDS-STATUS NOT = '00'
011300         DISPLAY 'KSDS FILE READING UNSUCCESSFULL'
KSDS-STATUS
011400         DISPLAY 'THE REQUESTED RECORD IS NOT
AVAILABLE'
011500         PERFORM READPS-PARA
011600     ELSE
011700         DISPLAY 'KSDS FILE READING SUCCESSFULL' KSDS-
STATUS
011800         PERFORM PROCESS-PARA1
011900     END-IF.
012000     PROCESS-PARA1.
012100     IF NOT TRANS
012200         DISPLAY 'THE TRANSACTION TYPE IS NOT VALID'
012300     ELSE
012400         IF ITEM-CODET = ITEM-CODE OF OUT-KSDS-REC
012500         IF UOM OF IN-PS-REC = UOM OF OUT-KSDS-REC
012600         DISPLAY ' THE RECORD IS VALID ONE AND FOUND '
012700         IF TRANS-TYPE OF IN-PS-REC = 1
012710             DISPLAY 'PERFORM PARA1'
012800             PERFORM PARA1
012900         ELSE
013000             IF TRANS-TYPE OF IN-PS-REC = 2
013010                 DISPLAY 'PERFORM PARA2'
013100                 PERFORM PARA2
013200             ELSE
```

INVENTORY CONTROL SYSTEM

```

013210          DISPLAY 'PERFORM PARA3'
013300          PERFORM PARA3
013400          END-IF
013500          END-IF
013600    *      PERFORM READ-PARA1
013700          ELSE
013800          DISPLAY ' THE RECORD IS NOT VALID ONE'
013900          END-IF
014000          ELSE
014100          DISPLAY 'THE PARTICULAR ITEM CODE: ' ITEM-
CODET ' IS NOT
014200    -      'FOUND'
014300          END-IF
014400          END-IF
014500          PERFORM READPS-PARA.
014600    PARA1.
014700          COMPUTE QIS = QIS - TRANS-QTY OF IN-PS-REC
014800          COMPUTE CYRQ = CYRQ + TRANS-QTY OF IN-PS-REC
014900          REWRITE OUT-KSDS-REC.
015000    PARA2.
015100          COMPUTE QIS = QIS - TRANS-QTY OF IN-PS-REC
015200          COMPUTE CYCQ = CYCQ - TRANS-QTY OF IN-PS-REC
015300          MOVE TRANS-DATE OF IN-PS-REC TO LID
015400          REWRITE OUT-KSDS-REC.
015500    PARA3.
015600          COMPUTE QIS = QIS - TRANS-QTY OF IN-PS-REC
015700          REWRITE OUT-KSDS-REC.
015800    ROL-PARA.
015900          MOVE III TO ITEM-CODE OF OUT-KSDS-REC
016000          START OUT-KSDS-FILE KEY >= ITEM-CODE OF OUT-
KSDS-REC
016100          IF KSDS-STATUS NOT = '00'
016200          DISPLAY 'KSDS FILE STARTING UNSUCCESSFULL'
KSDS-STATUS
016300          DISPLAY 'THE REQUESTED RANGE IS NOT AVAILABLE'
016400          PERFORM CLOSE-PARA1
016500          ELSE
016600          DISPLAY 'KSDS FILE STARTING SUCCESSFULL' KSDS-
STATUS

```

# INVENTORY CONTROL SYSTEM

```

016700     END-IF.
016800     ROL1-PARA.
016900     READ OUT-KSDS-FILE NEXT RECORD AT END MOVE 1
TO EOF1
017000             PERFORM CLOSE-PARA1.
017100     IF KSDS-STATUS NOT = '00'
017200     DISPLAY 'KSDS FILE READING UNSUCCESSFULL'
KSDS-STATUS
017300     DISPLAY 'THE REQUESTED RECORD IS NOT
AVAILABLE'
017400     ELSE
017500     DISPLAY 'KSDS FILE READING SUCCESSFULL' KSDS-
STATUS
017600     IF QIS = ROL
017700     DISPLAY 'THE ITEM CODE: ' ITEM-CODE OF OUT-KSDS-
REC ' IS
017800     -   'AT REORDER LEVEL'
017900     ELSE
018000     IF QIS < ROL
018100     DISPLAY 'THE ITEM CODE: ' ITEM-CODE OF OUT-
KSDS-REC ' I
018200     -   'S LESS THAN REORDER LEVEL'
018300     END-IF
018400     END-IF
018500     IF CYRQ = 0 AND CYCQ = 0
018600     DISPLAY 'THE ITEM CODE: ' ITEM-CODE OF OUT-
KSDS-REC ' IS
018700     -   'STILL NOW NON-MOVING ITEM'
018800     END-IF
018900     END-IF
019000     PERFORM ROL-PARA.
019100     CLOSE-PARA1.
019200     *   CLOSE IN-PS-FILE
019300     *   IF PS-STATUS NOT = '00'
019400     *   DISPLAY 'PS FILE CLOSING UNSUCCESSFULL' PS-
STATUS
019500     *   ELSE
019600     *   DISPLAY 'PS FILE CLOSING SUCCESSFULL'
019700     *   END-IF

```