VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
INTERNATIONAL UNIVERSITY
DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

APPLYING MATERIAL REQUIREMENT PLANNING SYSTEM FOR SAIGON GARMEX COMPANY

Submitted in partial fulfillment of the requirements for the Degree of Bachelor of Engineering in Industrial and Systems Engineering

Student: HO LE THANH TUNG – IEIEIU10028
Thesis advisor: Dr. NGUYEN VAN CHUNG

Ho Chi Minh city, Vietnam
07/2015

VIETNAM NATIONAL UNIVERSITY – HOCHIMINH CITY
INTERNATIONAL UNIVERSITY
DEPARTMENT OF INDUSTRIAL & SYSTEMS ENGINEERING

APPLYING MATERIAL REQUIREMENT PLANNING SYSTEM FOR SAIGON GARMEX COMPANY

Submitted in partial fulfillment of the requirements for the Degree of Bachelor of Engineering in Industrial and Systems Engineering

Student: HO LE THANH TUNG – IEIEIU10028
Thesis advisor: Dr. NGUYEN VAN CHUNG

Ho Chi Minh city, Vietnam
07/2015

APPLYING MATERIAL REQUIREMENT PLANNING SYSTEM FOR SAIGON GARMEX COMPANY

By
Ho Le Thanh Tung

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN INDUSTRIAL AND SYSTEMS ENGINEERING
AT THE DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING, INTERNATIONAL UNIVERSITY, HO CHI MINH CITY

JULY 2015

Signature of Student: _ _____
Ho Le Thanh Tung

Certified by : _____
Dr. Nguyen Van Chung
Thesis Advisor

Approved by : _____
Assoc. Prof. Ho Thanh Phong
Head of ISE Department

APPLYING MATERIAL REQUIREMENT PLANNING SYSTEM FOR SAIGON GARMEX COMPANY

By
Ho Le Thanh Tung

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF ENGINEERING IN INDUSTRIAL AND SYSTEMS ENGINEERING
AT THE DEPARTMENT OF INDUSTRIAL AND SYSTEMS ENGINEERING, INTERNATIONAL UNIVERSITY, HO CHI MINH CITY

JULY 2015

ABSTRACT

Garmex Saigon Company has been suffered uncontrollable material requirement for years. Sixty percent of necessary material for production is imported from China, which lead to many problems for the company. The precise time and quantity to release the purchasing order also needs for Return on Equity and competitive ability of the company. The main reason is that the Sale Department was mainly based on experiences without helping of specialized information technology tool. This thesis presents a design of MRP system for Garmex Saigon Company to improve its material requirement planning. The system utilizes several tools of inventory management and scheduling for calculation of required materials, quantity and time to release orders.

Keywords: material requirement planning, MRP, inventory management, scheduling, quantity, time for releasing order.

Thesis advisor: Dr. Nguyen Van Chung

ACKNOWLEDGMENTS

Foremost, I wish to express my sincere thanks to Dr Nguyen Van Chung, my supervisor, for continuous encouragement, motivation, patience guide and assistance in the period of writing this thesis. I am extreamly thankful and indebted to him for sharing expertise, and valuable guidance to me.

I take this opportunity to express gratitude to all of the Department faculty members for their help and support. I also thank my parents for the unceasin encouragement, support and attention. I am also grateful to my friend who supported me throught this venture.

I also place on record, my sense of gratitude to on and all, who directly or indirectly, have lent their hand in this venture.

TABLE OF CONTENTS

Contents

LIST OF TABLES

LIST OF FIGURES

Chapter 1 INTRODUCTION

This chapter introduces the brief description of the research's background and rationale; next, research problems are specified. Finally, the scope and limitations are discussed.

1.1 Introduction

Material Requirement Planning (MRP) is a control system, which calculates the material requirement and supply schedule to satisfy the demand of all products and parts used for manufacturing process.

MRP first introduced by Alfred Sloan in 1921. Over the time, it is developed and improved by many people, for instance Joseph Orlicky (1964), Oliver Wight (1983). MRP became an important part of manufacturing and were used by many companies in the world. By 1989, 1.2 billion dollars of MRP II software were sold to American industry.

In Vietnam, beside the international companies which applied MRP for a long time, many manufacturers in Vietnam still operate traditionally without controlling material, leading to unsatisfied customer demands, increased costs and incompatible with other companies. Therefore, MRP system is now necessary for these traditional companies expanded profit through controlling material and occupied many opportunities to meet customer satisfactions.

Information Technology plays an important role of designing and controlling Material Requirement System. Based on information analysis of sale orders, production orders, current inventory and forecasts, MRP programs determine the optimum inventory

level, ensuring enough material requirements and scheduling the purchasing and delivery activities. To be more specific, the main objectives of MRP is to eliminate stock out and excess stock, to equalize the need date and due date. It also breaks the products into subassemblies, then plans for those parts to come into warehouse when the production need.

MRP system uses four kinds of input information to determine the above questions:

- Master Production Schedule, which informs when products will be manufactured.
- Bill of Materials, which lists precisely material required to produce a finish good.
- Production cycle time and amount of material used for each cycle time.
- Supplier lead times

In this writing, I will mainly focus on applying MRP in Garmex Saigon by using methodology of material requirement planning and information technology.

1.2 Problem Statement

Garmex Saigon Cooperation is the third party manufacturing high quality garment products for many International Brands. Company mainly imports raw materials from other countries, which leads to several problems.

The first issue is Tariff Tax. The imported material used for producing garment products, which is exported to other countries, has 0% tax, while the unused imported material has to suffer 12% of tariff tax and 10% of value-added tax. Moreover, the

unused raw material just has very low price in the domestic market compare with the real buying value from the international suppliers.

Secondly, the quantity of order also raises high concerns of the company. The great amount of order forces company spending a large budget for the inventory and purchasing costs with low Return on Equity opportunity. However, they can suffer stock out, which affects directly on manufacturing activities, and deal with low order quantity that can reject by suppliers.

Lastly, the matter of lead time, purchasing time and low level of stocking also needs to be solved to improve the competitive ability of company.

The main reason for these issues is uncontrollable material planning due to lack of supported information system. Based on the virtual needs of solving problem for material planning, delivery schedule and purchasing activities, Material Requirement Planning System, which applied just in time theory, is the best solution for Garmex Saigon.

1.3 Objectives

The purpose of this study is design an MRP System which can satisfy two main objectives:

- Ensure materials are available for production and products are available for delivery to customers.
- Maintain the lowest possible material and product levels in store.

1.4 Scopes and limitations

1.4.1 Limitation:

Due to the time limitation and data securities of the Garmex Saigon company, the data collection used for this thesis is just 3 main styles of clothing, including 24 kinds of products which is manufacturing in An Phu factory, over 248 products (120 products in production line and 128 kinds which is discountinued operations) of the company. The collected data is taken in six months, from January to June of 2015.

The planning products including 3 clothing styles, including Uvablo, Micro Fiber Bunt Orange, and Woman Blazer.

The data is collected from An Phu Plant, one of four factories of the company. The lead time may be various between different materials and different suppliers. However, each supplier ensures a constant lead time for their product.

1.4.2 Scope

Developing a case study is applying MRP system for a Garmex Saigon Company to solve the inventory and purchasing problem.

Garment is one of the important industries of Vietnam, which contributes a large portion in the GDP. Expanding the MRP application system in the garment industry not only improves the efficiency, but also the competitive ability with other foreign companies.

Chapter 2 LITERATURE REVIEW

This chapter will briefly review seceral related researches of Material Requirement Planning.; next, basic concepts are specified, namely material requirement planning, bill of materials, master production schedule, item master file, management information system, programing, Visual Studio and MySQL.

2.1 Literature Review

Material Requirement Planning (MRP) is a popular method for scheduling and production planning. Though there have been several researches addressing MRP implementation, evaluating the impacts of technology and its characteristics.

Masnita and Mahdani (2012) [10] specified six components to control the execution procedure of MRP, to be specific "the principle administration backing and formal arranging", "the precision of the information", "setting the association's instruction or preparing", "extraordinary arranging or control approaches and techniques", "normal for programming", and "individual attributes of representatives". The examination likewise brings up the commitment components influencing on MRP application in Small-Medium Enterprise, such that consumer loyalty, arranging and administration control, expanded proficiency, information and skills, hierarchical atmosphere, lastly expanded creation limit and expense gauges.

The outcome in view of the various relapse examination demonstrates the positive relationship between controlling elements and commitment elements. The creators proposed that doing these elements is essential for contending and enhancing corporate benefits of the organization.

E. Shevtchenko (2004) [6] examined expert and cons of two MRP merchants: "Scala Business arrangement" and " Monitor Industriukveckling AB". The study goes for investigating hypothetical MRP base and programming proficiency examination of "asset arranging administration" issues, with a specific end goal to discover the first of each answer for expanding benefit at assembling venture. In addition, the paper demonstrated the advantage of actualizing MRP programming, contrasted with the occasional paper work. It likewise exhibits the contrast between the accomplishing arranging result and the genuine result, then proposes the conceivable enhancements in the "Screen" and " Scala" programming.

The arrangement examination primarily concentrates on: utilizing of work focus stacking reproduction potential outcomes for the improvement of present and future assembling limit; pre-count for the future items; rating of subcontractors; EOQ estimation; conveyance execution investigation; streamlining of stock; production network administration. The noteworthy estimation demonstrates the pattern of development for the ERP (Enterprise Resource Planning) programming mix for little and medium size ventures, yet the greater organization will allude APS (Advanced Planning and Scheduling) frameworks, which upgrades existing assembling abilities , while helping oversees valuable and ongoing data with development booking and quality control usefulness.

H.Sadeghi, et al. (2014) [2] accepted that segments lead-times are probabilistic and a MRP with occasional request amount (POQ) strategy is connected for segments arranging. They utilized Monte-Carlo reproduction to produce a few situations with a specific end goal to locate the suitable answer for ward interest and vulnerability lead-

time issues. The fundamental point is minimizing the aggregate expense, including parts holding expense, last item accumulating expense, last item holding cost and set up expense.

Arbitrary lead-time may comes about either surplus or lack of stock. Questionable lead-time can be planed as the aggregate of the anticipated and security lead-time. The percentile of likelihood appropriations of genuine lead-time is utilized to define the exchange off between over stocking and stock out while minimizing the aggregate expense.

2.2 Basic Concept

2.2.1 Material Requirement Planning (MRP)

Material Requirement Planning (MRP) is designed to help production managers placing dependent demand items and scheduling based on computer management system. Component of finished products, which inventory quantity is depend on the production level of final product, is called dependent demand items, like subassemblies, raw material or components parts. For instance, in garment factory, dependent demand items include shell, zippers, button, and labels.

In the 1940s to 1950s, the first Material Requirement Planning systems exploded the information from bill of materials by computer mainframe for planning purchasing items of a certain product. MRP then was expand for production manager to receive feedback, change and update inputs when needed. Manufacturing Resource Planning or MRP II incorporated human resources, finance, marketing, engineering and accounting into planning procedure. ERP (Enterprise Resource Planning), which is linked multifunctional areas through entire business enterprise, is expanded from MRP concepts.

MRP calculates backward from a finished goods plan to develop order requirements for raw materials. Starting with the finished products, MRP evaluates schedule requirements for assemble items, part of components and raw materials, which is used to produce final goods. Material Requirement Planning system is designed to answer three main questions: What component is needed? When is it needed? And how much is needed?

In MRP, inventory requirements is broken down into planning period in order to complete production in a punctual time while keeping minimum inventory level to reduce carrying costs. MRP evaluates capacity requirement and allocates production time for production manager. However, implementing this system is costly and time consuming, which is not always good for small businesses. Moreover, input information needs to accurate and update to get the most precise output information. Therefore, companies have to maintain accurate updating bills of materials, records of inventory, part quantities when MRP is applied.

2.2.2 Bill of Material (BOM)

A bill of materials is a product structure which records components, such as raw material, supplies and subassembly, used to produce a finished good. A bill of materials not only includes quantity requirement to make a finished good, but also estimate percentage of waste during production procedure. Bill of materials also can be evaluated product cost and order quantity from vendors.

Bills of materials is used to design process, manufacturing , and it is one of the most important feature for calculating material requirement planning and enterprise resource planning systems.

2.2.3 Master Production Schedule (MPS)

A Master Production Schedule (MPS) is a plan for production finished goods in the short-range of planning horizon. MPS sets number of end products need to be completed in a period of time.

The Master Production Schedule gives information to plan and control production, purchasing, and planning manufacturing operation. It also used for business planning and forecasting to inform operation detail through MPS. The aim of Master Production Schedule is to decrease carrying costs and raise stability of plans. It also ensure availability of materials, safety stock and low level of stock. Therefore, storage costs are reduced, typically for high value materials.

The master plan of these materials will drive production requirement and specific materials for Material Requirement Planning module. This plan influences whole production process. The raw materials planning is the result of finished goods and assemblies planning. MRP can be unstable when the level of finished goods is changed frequently.

2.2.4 Item Master File

Master Data Management (MDM) is a thorough system for empowering a venture to connection the greater part of its basic information to one document, called an master record, that gives a typical perspective. At the point when legitimately done, expert information administration streamlines information sharing among work force and offices. Furthermore, ace information administration can encourage figuring in different framework architectures, stages and applications.

At its center Master Data Mangagement (MDM) can be seen as an "order for particular quality change" characterized by the strategies and techniques put set up by an information administration association.

In business, Master Data Mangagement (MDM) embodies the procedures, administration, strategies, principles and instruments that reliably characterize and deal with the discriminating information of an association to give a solitary perspective.

Procedures generally found in expert information administration incorporate source ID, information gathering, information change, standardization, principle organization, lapse location and adjustment, information combination, information stockpiling, information conveyance, information characterization, scientific classification administrations, thing expert creation, composition mapping, item codification, information enhancement and information administration.

2.2.5 Management Information System (MIS)

MIS represents the most critical -- and often, most proprietary and difficult to upgrade -- applications used internally by enterprises.

A sorted out way to deal with the investigation of the data needs of an association's administration at each level in making operational, strategic, and key choices. Its goal is to plan and execute techniques, procedures, and schedules that give suitably itemized reports in a precise, steady, and opportune way.

In an administration data framework, present day, electronic frameworks consistently assemble significant information, both from inside and outside an association. This information is then handled, coordinated, and put away in a brought together database (or information distribution center) where it is always upgraded and

made accessible to all who have the power to get to it, in a shape that suits their motivation.

The MIS division was initially the entire of data innovation. From the 1960s to the mid 1980s, professionals and business colleges alluded to MIS as opposed to IT. In the good 'ol days, endeavor processing's principle part was to help the CEO and CFO with data frameworks administration for a couple key run-the-business errands, for example, request passage, bookkeeping and planning. No endeavor applications existed; software engineers carefully composed code to do these capacities, for the most part on a centralized computer. These frameworks were business-discriminating, implying that a business would come up short in the event that it needed to about-face to manual bookkeeping. On the off chance that MIS fizzled, the business was in risk. The CFO administered MIS, guaranteeing that the engineers and directors conveyed what bookkeeping required.

Today, Management Information Systems is utilized extensively as a part of different settings and incorporates however is not restricted to: choice emotionally supportive networks, asset and individuals administration applications, venture administration, and database recovery applications. In spite of the fact that the limits have get to be fluffy throughout the years, regularly MIS still covers frameworks that are discriminating to the organization's capacity to survive, including bookkeeping and request passage. Upper administration ought not dismiss this. In numerous organizations, MIS handles legacy programming and equipment, coded by developers since a long time ago resigned, who left no documentation for the frameworks. The undertaking updates or modernizes these frameworks just precisely, and with high energy about the dangers

included. Along these lines, MIS, and the individuals who bolster it and know its eccentricities, remains a crucial if under-commended piece of big business IT.

2.2.6 Programming

People speak with one another through dialect, making words and sound examples. Programming dialect like that, which is a situated of words and images that permits developers or clients can converse with the PC. Like English, Spanish or Chinese and the other voice dialect, programming dialects are additionally called linguistic structure rules (grammar) to guarantee that dialect is connected accurately determination. It is a situated of directions are organized in a foreordained request to control the execution of PC control, activities important to meet a foreordained objective of the human as recovering information, inquiry, illuminate baffles, ... The order can be composed in a wide range of programming dialects.

There are many distinctive programming dialects, every dialect has its own sentence structure. A few dialects are being created for utilization on particular sorts of PCs, different dialects then - because of the accomplishment of it - has turn into the standard and are connected on most PCs. Programming dialects can be partitioned into 3 fundamental sorts: machine dialect, low level computing construct and high - level dialect.

Machine dialect (machine code) is a dialect stage processor. The system is composed in a wide range of different dialects at last are changed over into machine dialect before the project is executed. Due to the machine dialect direction set relies on upon the sort of processor machine dialect will be distinctive on the PC that utilization diverse processor. The upside of composing projects in machine dialect developers can

control the PC straightforwardly and accomplish precisely what they need to do. Hence, the machine dialect projects are elegantly composed exceptionally successful system (quick execution speed, little size). The disservice of the machine dialect system would typically take a ton of time to compose, extremely hard to peruse, screen for issues. Moreover, on the grounds that the project is composed in the script relies on upon the processors, the system just keeps running on PCs with the same processor just. Machine dialect is otherwise called low-level dialect.

Constructing agent was produced to help developers infectious project mandates more. Comparative constructing agent dialect utilizing machine yet reminiscent images speak to the machine code. Another component is that the regular dialect permits the location structure (typical tending to), ie an area in PC memory that can be referenced by a name or images, for example, TOTAL rather utilize its genuine location (utilizing twofold numbers) in machine dialect. The get together program incorporates macroeconomic markers (full scale guideline) can produce machine code orders. The get together program was exchanged to machine code by an extraordinary system called constructing agent (constructing agent). Albeit moderately simple to utilize low level computing construct than constructing agent machine code yet is still viewed as a low level dialect in light of the fact that it is still close with each of PC outline.

The upheaval of programming languages started with the improvement of abnormal state dialect in the late 1950s and 1960. The abnormal state dialect closer to the thought of dialect that most everybody knows, it incorporates things, verbs, scientific images, contacts and rationale operations. These elements can be facilitated, connected together constitute a type of inquiry. The "v" is called suggestions of the system (program

proclamation). Another vital point of interest is the abnormal state basic dialect does not rely on upon the PC, implying that projects written in abnormal state dialect that can keep running on distinctive sorts of PCs (utilizing diverse processors).

2.2.7 Visual Studio

Microsoft Visual Studio is an incorporated advancement environment (IDE) from Microsoft. It is utilized to build up a PC program for Microsoft Windows, and additionally sites, web applications and web administrations. Visual Studio improvement stage utilizing Microsoft programming, for example, Windows API, Windows Forms , Windows Presentation Foundation , Windows Store and Microsoft Silverlight . It can deliver both scripting languages and oversaw code.

Visual Studio incorporates a code editorial manager supporting IntelliSense and in addition enhancing the code. Incorporated debugger works both on the level debugger and investigate source code level machine. Other coordinated devices incorporate a structures outline assemble GUI applications, web plan, plan and configuration class database pattern. It acknowledges modules upgrade the usefulness at all levels, including more backing for adaptation control framework (like Subversion) and including another arrangement of devices like editors and visual outline for area particular dialect or set of instruments for diverse parts of the product improvement process .

Visual Studio support various programming dialects and permits the code editorial manager and debugger to backing (to differing degrees), practically every programming dialect . The coordinated dialect incorporate C , C ++ and C ++/CLI (by means of Visual C ++) , VB.NET (Visual Basic.NET through) , C rise (by means of Visual C #) and F ups (as of Visual Studio 2010) . Support for different dialects like J

++/J ups, Python and Ruby through a different establishment administrations. It likewise underpins XML/XSLT, HTML/XHTML, JavaScript and CSS. Microsoft offers a rendition of " Express " is the free form of Visual Studio .

Visual Studio does not bolster any programming dialect or instrument arrangements significantly, rather it permits fitting capacities encoded as a VSPackage. At the point when introduced, these capacities are accessible as an administration. IDE gives three administrations: SVsSolution give the capacity to rundown activities and arrangements; SVsUIShell give windows and UI and SVsShell. Additionally, IDE is likewise in charge of organizing and empowering correspondence between administrations. All editors, planners, venture sorts and different instruments agree VSPackages. Visual Studio utilizes COM to get to VSPackages. Visual Studio SDK additionally incorporates the Managed Package Framework (MPF) is a situated of oversaw wrapper around the COM-interfaces permit the bundle to be composed in any dialect. Nonetheless, MPF does not give all the usefulness portrayed in Visual Studio COM interfaces. The administration can be devoured to make different bundles, to add usefulness to Visual Studio IDE.

Visual Studio elements foundation arrangement (likewise called incremental gathering). As code is being composed, Visual Studio aggregates it out of sight to give input about language structure and assemblage mistakes, stamped with a wavy red underline. Foundation compiler does not make executable code, on the grounds that it obliges a compiler other than to use to make executable code. Arrange the first stage was presented with Microsoft Visual Basic yet has now been reached out to all dialects.

Visual Studio has a debugger works both as a source-level debugger and a machine-level debugger. It lives up to expectations with both oversaw code and also machine dialect and can be utilized to troubleshoot applications written in dialects bolstered by Visual Studio. Furthermore, it can likewise connect to running procedures and screen and troubleshoot those procedures. On the off chance that the source code for the operation is accessible, it will show the code as it is being run. On the off chance that the source code is not accessible, it can demonstrate the dismantling. The Visual Studio debugger can likewise make memory dumps and in addition load them later for investigating. The multi-strung projects are likewise bolstered. Debugger can be arranged to be propelled when an application is running outside the Visual Studio environment crashes.

2.2.8 MySQL

MySQL is an information administration framework for nothing, incorporated utilization with apache, PHP. Key considers group improvement ought to mysql open source has a great deal of backing from engineers who adoration open source. Mysql additionally have the same access and the same code with SQL dialect. Be that as it may, no overall Mysql questions as SQL premium. Mysql basically just meets straightforward recovery in the operation of the site yet most can take care of the issue in PHP.

MySQL is a fast databases, security and convenience, portability, takes a shot at numerous working frameworks give an expansive framework extremely dependable utility capacities. With rate and high security, MySQL is appropriate for applications that get to databases on the web. MySQL is totally free so you can download from the landing page MySQL.

MySQL is utilized for supporting PHP, Perl, and different dialects, it makes the capacity of data on the site pages written in PHP or Perl. MySQL is one of the exceptionally essential illustrations System Database Administration social inquiry dialect uses organized (SQL)

2.3 MRP System

2.3.1 Cost

2.3.1.1 Inventory Holding Cost

Inventory holding costs are the costs of maintaining and storing items over a period of time. Particularly, the holding cost can be described as per cent of item value stored in warehouse during a year. They commonly depend on the field of business. In Garmex Saigon, holding cost occupies a large value of inventory on hand, generally around 30 per cents. In this MRP program, the carrying cost (or Ch in short) is calculated and divided into two type, one for high value items like shells and another for low value items.

2.3.1.2 Ordering Cost

Ordering cost in MRP system are the incremental costs of goods order procedure from a vendor. For instance, costs of an order can be considered as the costs of placing a purchase order, costs of inspection after receiving batches, preparing a requisition costs, processing the invoice of vendor and remitting the supplier payment.

2.3.1.3 Purchasing Cost

The price that a company pays for an item is called purchase price. The price is given by the supplier and it can be fluctuated depend on quantity of order. In case of

vendors offer a discount price of an item when the order reaches a certain quantity, the total order quantity need to be recalculated to achieve the lowest unit price.

The purchasing cost is the cost of buying items in an order, expressed in formula below:

Purchasing Cost = Purchasing Price * Total Order Quantity

2.3.1.4 Total Cost

Total Cost is calculated in order to propose an estimated cost for planner. Base on this total cost, planner can evaluate price of item which has to maintain both criteria, competitive price and maximize profit.

Generally, the Total Cost in the system is considered as the sum of holding cost, ordering cost and purchasing cost of order items in a time interval.

Total Cost = Total Ordering Cost + Total Holding Cost + Total Purchasing Cost

2.3.2 Lead Time

The time between the process initiation and its completion is lead time. This paper will focus on the manufacturing lead time, which is the time required to make or buy an item. Lead time is used to calculate planned order releases date based on the delivery date of orders .There are two main methods for this calculation: Fixed Lead Time and Variable Lead Time.

2.3.2.1 Calculating Fixed Lead Time (Buy Item Lead Time)

This method is used in the case lead time set in an item thought to be fixed (such as purchasing item), and the delivery order date is used for the Planned Release Date calculation, as the formula below:

Planned Release Date = Delivery Date - (Lead Time + Safety Lead Time)

= Fixed Lead Time

2.3.2.2 Calculating Variable Lead Time (Make Item Lead Time)

According to the lot size, the lead time is changed in this method. Time for waiting, set up, post-processing and transfer set for an item is supposed to be fixed and the variable lead time can be figured out with the cycle time (actual operating time per piece) by the following formula:

Planned Release Date = Delivery Date - [(Waiting + Setup + Post-Processing + Transfer) + Order Requirements x Actual Operating Time per 1 piece]

= Fixed Lead Time + Variable Lead Time

Where Fixed Lead Time = Delivery Date - (Waiting + Setup + Post-Processing + Transfer)

Variable Lead Time = Order Requirements x Actual Operating Time per piece

2.3.2.3 Total Lead Time

In general, the total lead time is the sum of the fixed lead time and the variable lead time. The MRP System uses the total lead time to obtain order release dates from order due dates.

2.3.3 Lot-Sizing in MRP

This MRP System provides 7 Lot-Sizing Rules, including Multiple Order Quantity, Minimum Order Quantity, Lot For Lot, Economic Order Quantity (EOQ), Periodic Order Quantity (POQ), and Discount Order Quantity and Least Total Cost.

2.3.3.1 Multiple Order Quantity

Multiple order quantity allows vendors to state minimum order multiple. The quantity of the order is depending on the number of item in a vendor package type, such as a pack of polybags, a carton or a pallet.

2.3.3.2 Minimum Order Quantity

Minimum order quantity is the minimum size of an order which is decided by a vendor. For instance, Garmex factory needs 300 zippers for manufacturing; however, the zipper supplier requires that an order has to place at least 500 zippers at once. In order word, that vendor are not interested in a customer with an order lower than 500.

2.3.3.3 Lot for Lot

Replenishment orders are planned as required.

2.3.3.4 Economic Order Quantity

In general, Economic Order Quantity model is applied for constant uniform demand. In reality, demands are not stable, leading to the modification of EOQ procedure base on lumpy and discrete demand.

The order can excess the needs, leading left over inventory from week to week. For example, there are 41 inventory items excess from week 7 put into week 8 when new order is released.

When the requirement exceed sum of EOQ lot size and inventory at the previous period, the order quantity needs to increase to meet the requirements.

2.3.3.5 Periodic Order Quantity

Periodic Order Quantity application fulfills requirements in the fixed number of time period, which is determined as Economic Order Quantity over mean demand rate.

Lot size in Periodic Order Quantity method can be varied and the time intervals between replenishment orders are fixed, leading the combination of orders during low demand period.

2.3.3.6 Discount Model – Least Unit Cost

Least Unit Cost Rule is particularly suitable for calculating the purchasing lot-size in case of discount quantity exists.

a/ Discount Quantity Problem

This problem happens when the suppliers propose the discount prices in a specific order quantity. In this case, Least Unit Cost is used to find the optimum quantity for the purchase order which satisfies lowest unit cost.

b/ Least Unit Cost

We follow the below steps to calculate lot size base on Least Unit Cost:

-Accumulate the requirements period by period until the order quantity reaches the required quantity of discount price.

-Also accumulate the order quantity requirement exactly equal to the quantity of discount.

-Base on calculated the least unit cost, determine the optimal quantity to place order.

Unit cost = (Ordering cost + Carrying cost + purchase price) divided by

order quantity

Chapter 3 METHODOLOGY

This chapter introduces methodology of the research; next, some basic knowledge related to MRP system is specified. Finally, testing method is introduced to verify MRP system.

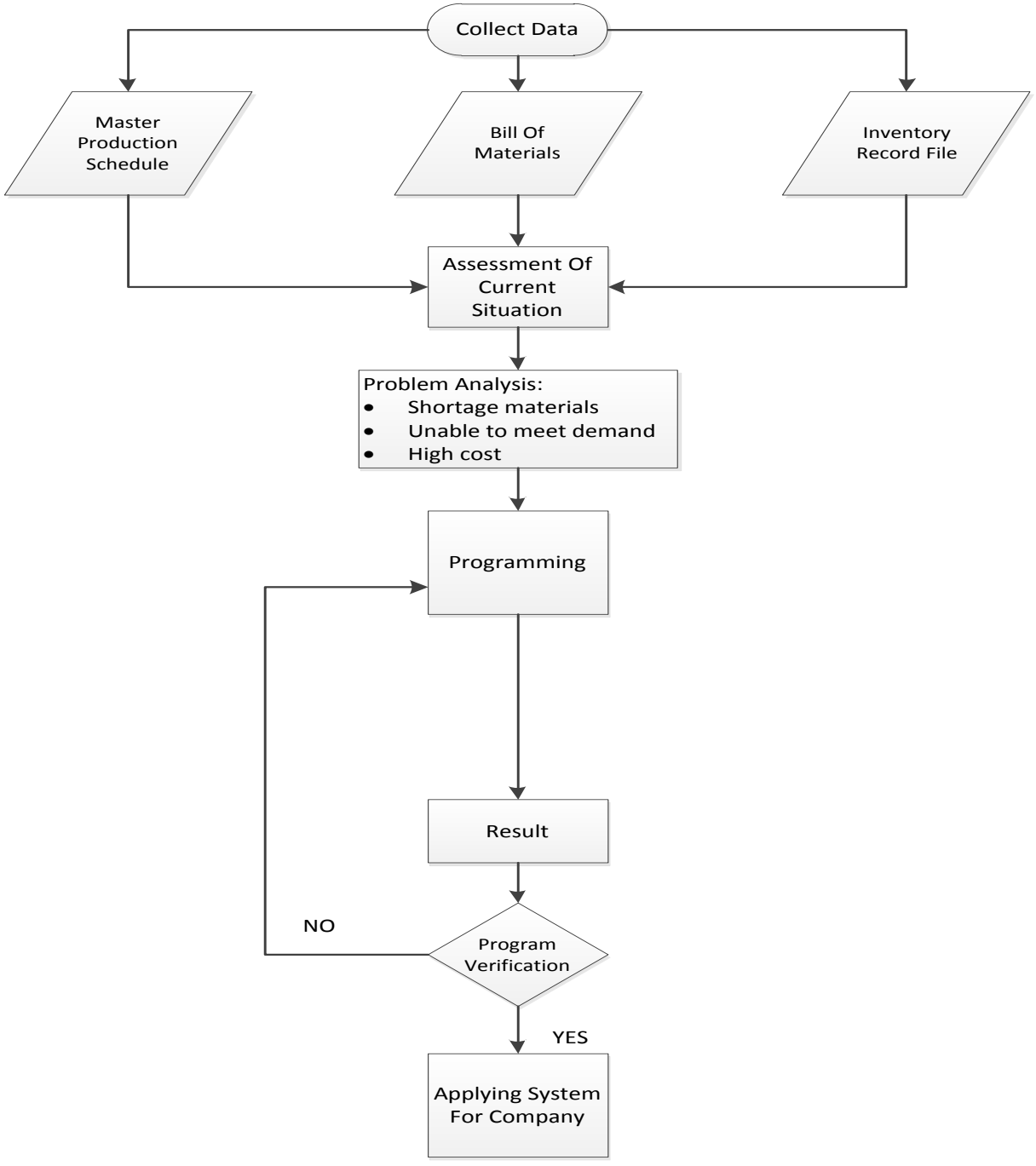


Figure1: Methodology Flow Chart of MRP System Programming

The methodology flow chart (Figure 1) expresses the method of designing Material Requirement System for Garmex Saigon company, following the procedure:

1/ Data collection is the main base of the MRP program, including Master Production Schedule, Bill Of Material, Inventory Record File.

- Master Production Schedule, which informs when products will be manufactured.
- Bill of Materials, which list precisely material required to produce a finish good.
- Inventory Record File, including supplier lead time and inventory on hand status

2/ Assessment of current situation of the studied company

3/ Problem analysis: While many types of materials for production shortages frequently, there are several kinds of component over excess allowance level of inventory. The company is also unable to meet customer demands. Material requirement planning system is a chosen solution to solve material control problem at Garmex Saigon.

4/ Programing: The process of writing a MRP System program, namely inserting database, coding and debugging, designing interface.

5/ Result

6/ Program Verification: this is a very important step in order to check the reliability of the program.

7/ Apply the MRP system for the Garmex Saigon company if the grogram passes the verification procedure.

3.1 Testing

Comparison test is applied to evaluate the reliability of MRP program. To test each type of lot-sizing rule, three products are used in three times based on the bills of material structure and data of demand. The test is solved by hand, and then the results will be compared to the output of MRP program to check the algorithm authenticity.

Chapter 4 INTRODUCTION TO GARMEX SAIGON

This chapter introduces the brief information of Garmex Saigon Company; next, manufacturing process are specified, which are then directed down to list of products. Finally, the data of demand is discussed, coming up with the final part which is bills of material.

4.1 Saigon Garmex Company

Established in 1976 - one year after reunification, JSC Saigon Garment Manufacturing Trade (Garmex Saigon js) started as a state-possessed undertaking . At first the organization called Union of Garment Factory Ho Chi Minh City overseeing unit deals with a sizable number of fare article of clothing processing plants in Ho Chi Minh City. In 2004, Saigon js Garmex equitized. In 2006, Garmex js Saigon Securities recorded on the Ho Chi Minh City (HoSE) with the securities code is GMC . Saigon js Garmex headquartered at 252 Nguyen Van Luong Street, Ward 17 , Go Vap District , Ho Chi Minh City .

In 2006, Garmex js Saigon Securities recorded on the Ho Chi Minh City (HoSE) with the securities code is GMC . Saigon js Garmex headquartered at 252 Nguyen Van Luong Street, Ward 17, Go Vap District , Ho Chi Minh City

At present, Garmex Saigon js 2 backup undertakings including Nhon Garment Factory and A Phu Garment Factory are situated in Ho Chi Minh City .

To grow creation limit, contributing Garmex Saigon js construct Tan My Garment Company on a region of 50,000 m2 , situated in the modern complex Hac Dich , Ba Ria Vung Tau .

Keep on planning for business opportunities from exchange assentions in the middle of Vietnam and different nations in the district, Garmex Saigon has collaborated with Blue Exchange (the design store chain prevalent in Vietnam) foundation Saigon Garment Co., Ltd. Green in 2012, resulting to the development of two units appended - Blue Saigon LLC is headquartered in Los Angeles, California (USA) and Ha Lam Garment Factory movement in Quang Nam. Blue Saigon LLC (USA) are attempting to recognize a line from America to generation in Ha Lam Garment Factory, the usage of methodologies ... direct deals to US clients by method of ODM (configuration completed available to be purchased to clients)

The principle fare markets are the EU js Garmex Saigon, the US and Japan with substantial clients as Decathlon 3 (France), New Wave (Sweden) and Columbia (United States)

Notwithstanding the creation of pieces of clothing fares are characterized as center capabilities are fundamentally productive , Garmex Saigon js are searching for chances to add to the venture of area possessed , including Central Project Service focal point of Applied Technology High Tech Medical (coordinated effort with Medical University Hospital ., Ho Chi Minh City) is situated at 213 Hong Bang Street, District 5 , Ho Chi Minh.

With beginning sanction capital of 22 billion , after over 10 years of operation , up to 2013 , Garmex Saigon has raised its contract cashflow to more than 106 billion and value to 212 billion. Aggregate recorded shares is 8,868,571 by 2,551 individual shareholders in the nation, 46 local lawful shareholders , 60 shareholders and remote people and lawful elements 8 outside shareholders holding.

In the setting of the current serious rivalry, Garmex Saigon js has effectively enhanced working strategies and administration of generation , applying LEAN framework (incline fabricating)/JIT to enhance profitability , quality and effectiveness business. Likewise, staff arrangements taken focused WORKERS makes representatives feel secure in life and work , empowering them drew in and dynamic commitment to the supportable advancement of js Garmex Saigon .

4.2 Production

Garmex Saigon mainly focuses on garment manufacturing, which includes several processes from receiving orders to dispatching the finished garments to customers. The flow chart below illustrates a transformation process from raw materials to the garment products.

Manufacturing processes in Garmex Saigon are divided in to three categories:

- Pre-Production Processes - includes sampling, purchasing raw materials, Approvals, Pre-Production meeting etc.
- Production processes - cutting, sewing etc.
- Post production processes - thread trimming, pressing, checking, folding, packing, shipment inspection etc.

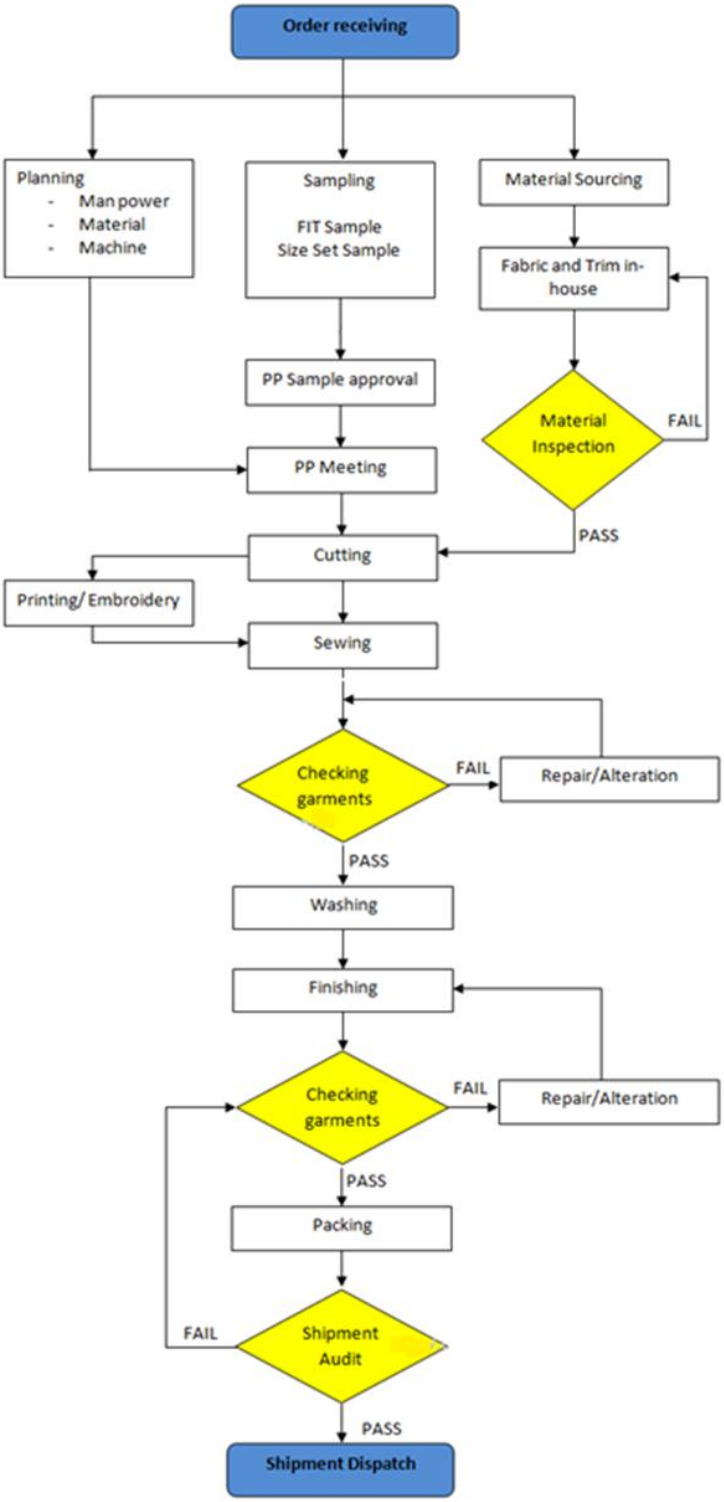


Figure 2 Manufacturing Process

[13]

4.3 List of products

Table 1: List of products

Style	WG6756		
No	ID	Color	Size
1	WG6756-1	Purple Lotus	XXS
2	WG6756-2	Purple Lotus	XS
3	WG6756-3	Purple Lotus	M
4	WG6756-4	Purple Lotus	L
5	WG6756-5	Bright Rose	XXS
6	WG6756-6	Bright Rose	XS
7	WG6756-7	Bright Rose	M
8	WG6756-8	Bright Rose	L
Style	**VL6114**		
No	ID	Color	Size
1	VL6114-1	Black	S
2	VL6114-2	Black	M
3	VL6114-3	Black	L
4	VL6114-4	Charcoal Heather	S
5	VL6114-5	Charcoal Heather	M
6	VL6114-6	Charcoal Heather	L
7	VL6114-7	Columbia Navy	S
8	VL6114-8	Columbia Navy	M
Style	**XM6138**		
No	ID	Color	Size
1	XM6138-1	Black	S
2	XM6138-2	Black	M
3	XM6138-3	Black	L
4	XM6138-4	Charcoal Heather	S
5	XM6138-5	Charcoal Heather	M
6	XM6138-6	Charcoal Heather	L
7	XM6138-7	Columbia Navy	M
8	XM6138-8	Columbia Navy	L

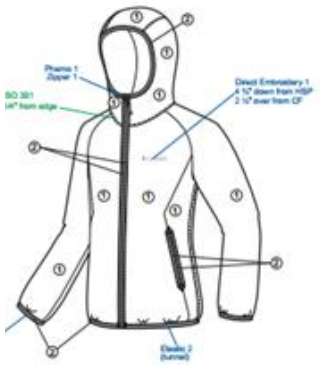


Figure 3: Product XM6138

4.4 Data of demand

Table 2: Master Production Schedule

Style	WG6756						
No	ID	Date	Quantity	Date	Quantity	Date	Quantity
1	WG6756-1	2/3/2015	1758	27/4/2015	1658	22/6/2015	1773
2	WG6756-2	9/3/2015	1271	4/5/2015	1553	29/6/2015	1414
3	WG6756-3	16/3/2015	502	11/5/2015	1426	6/7/2015	1492
4	WG6756-4	23/3/2015	1136	18/5/2015	570	13/7/2015	877
5	WG6756-5	30/3/2015	957	25/5/2015	1767	20/7/2015	1800
6	WG6756-6	6/4/2015	1242	1/6/2015	1258	27/7/2015	1416
7	WG6756-7	13/4/2015	1034	8/6/2015	1102		
8	WG6756-8	20/4/2015	493	15/6/2015	1407		
Style	VL6114						
No	ID	Date	Quantity	Date	Quantity	Date	Quantity
1	VL6114-1	4/3/2015	750	29/4/2015	591	24/6/2015	1499
2	VL6114-2	11/3/2015	1183	6/5/2015	750	1/7/2015	911
3	VL6114-3	18/3/2015	1594	13/5/2015	416	8/7/2015	630
4	VL6114-4	25/3/2015	1175	20/5/2015	792	15/7/2015	458
5	VL6114-5	1/4/2015	1473	27/5/2015	1274	22/7/2015	555
6	VL6114-6	8/4/2015	894	3/6/2015	1010	27/7/2015	1245
7	VL6114-7	15/4/2015	1446	10/6/2015	1752	29/7/2015	1775
8	VL6114-8	22/4/2015	538	17/6/2015	1639		
Style	XM6138						
No	ID	Date	Quantity	Date	Quantity	Date	Quantity
1	XM6138-1	6/3/2015	1758	1/5/2015	1658	26/6/2015	1110
2	XM6138-2	16/3/2015	1271	8/5/2015	1553	3/7/2015	1138
3	XM6138-3	20/3/2015	502	15/5/2015	1426	10/7/2015	1180
4	XM6138-4	27/3/2015	1136	22/5/2015	570	17/7/2015	1164
5	XM6138-5	3/4/2015	957	29/5/2015	1767	24/7/2015	561
6	XM6138-6	10/4/2015	1242	5/6/2015	1258	31/7/2015	1377
7	XM6138-7	17/4/2015	1034	12/6/2015	1102		
8	XM6138-8	24/4/2015	493	19/6/2015	1407		

4.5 Bills of materials:

Table 3: Similarties and differences in BOM

MATERIAL	STYLE																							
	WG6756								VL6114								XM6131							
	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8	1	2	3	4	5	6	7	8
003299	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■								
004476A	■	■	■	■													■	■	■	■	■	■		
004476B					■	■	■	■															■	■
005038A	■				■																			
005038B		■				■																		
005038C			■				■																	
005038D				■				■																
005274	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
005542A																	■	■	■	■	■	■		
005542B																							■	■
006492									■	■	■	■	■	■	■	■								
006564									■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
007016												■	■	■						■	■	■		
013061A	■				■				■			■			■		■			■				
013061B		■				■				■			■			■		■			■		■	
013061C			■				■				■			■					■			■		■
013061D				■				■																
013891A	■	■	■	■																				
013891B					■	■	■	■																
015370	■	■	■	■	■	■	■	■									■	■	■	■	■	■	■	■
029906																	■	■	■	■	■	■	■	■
029909	■	■	■	■	■	■	■	■																
029913									■	■	■	■	■	■	■	■								
053817A									■	■	■	■	■	■	■	■								
055257	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■	■
058818A	■	■	■	■					■	■	■						■	■	■				■	
058818B					■	■	■	■							■	■								■
059197																	■	■	■	■	■	■	■	■
059199A	■				■				■			■			■									
059199B		■				■				■			■			■								
059199C			■				■				■			■										
059199D				■				■																
059200																								
059200A																	■			■				

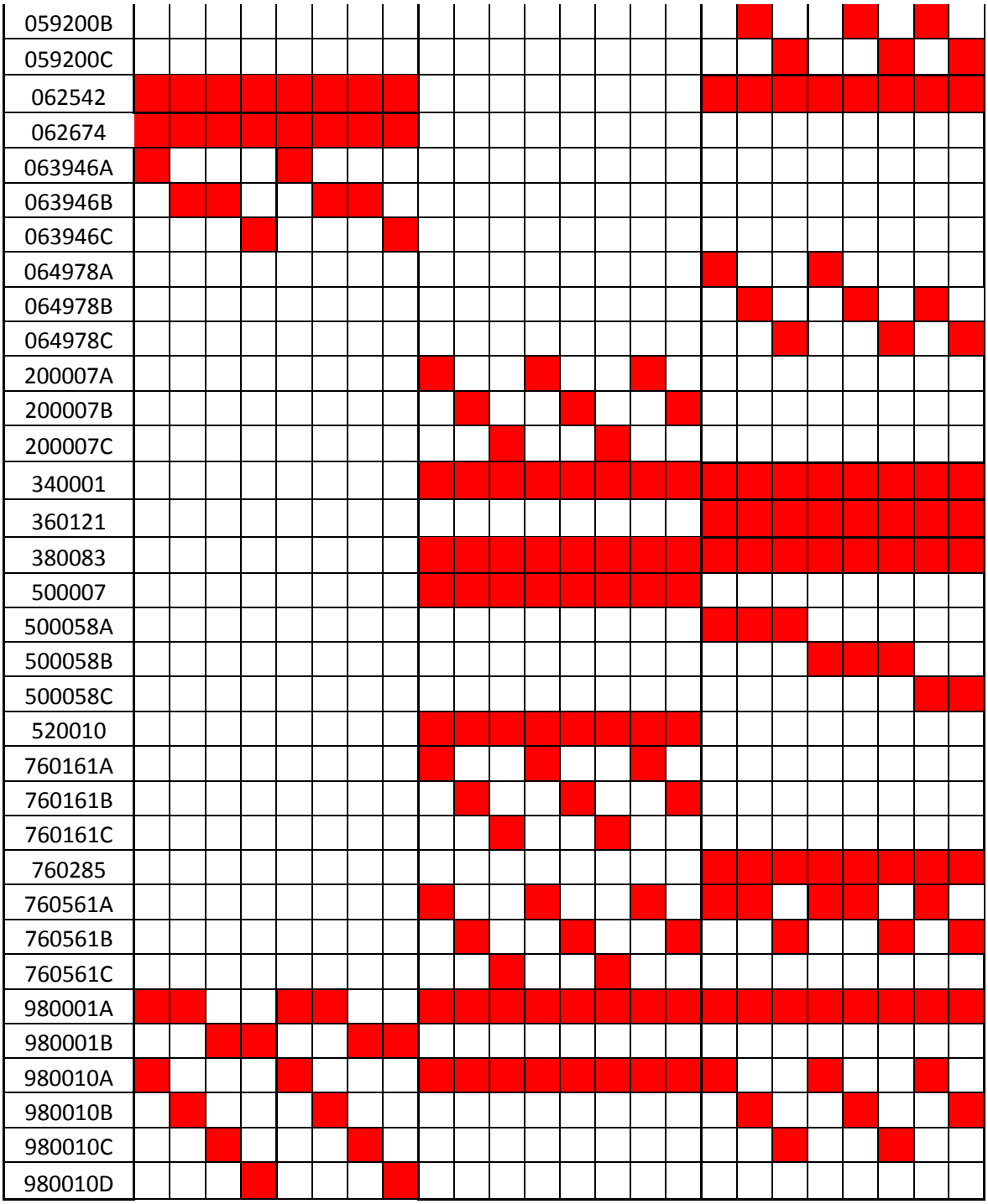


Table 3 expresses requirement materials used for 24 types of product. There are particular materials of each type, also several common materials between types.

In each style of product, the shape of bills of materials is quit similar. However, the components in each BOM are different and depended on the color, shell material and size of product. For instance, the BOM of XM6138 below express one style of product with the same black color but different sizes, the components 98001, 013061,059200.

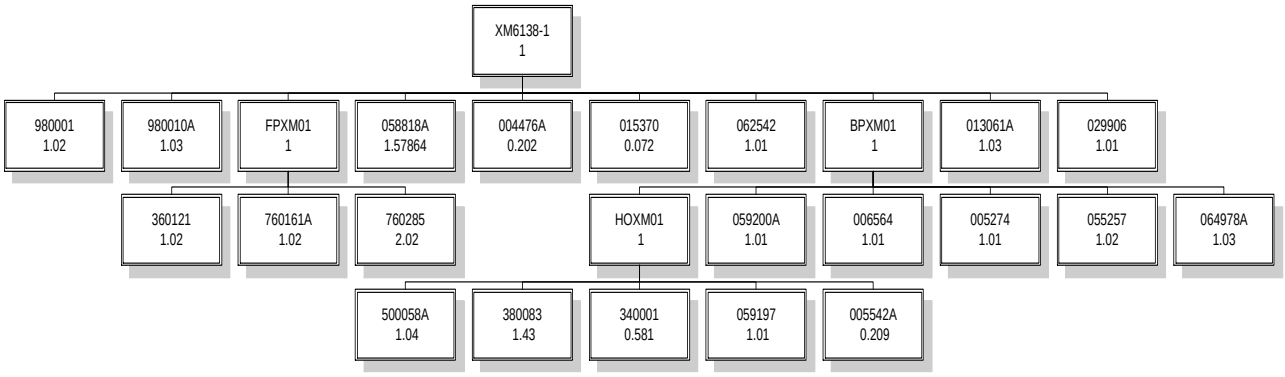


Figure 4 : BOM of XM6138-1 – Black – S

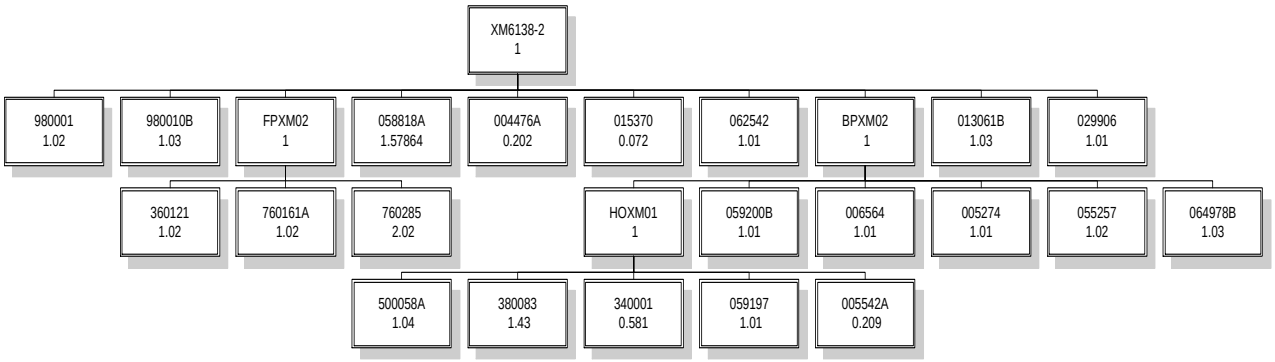


Figure 5 : BOM of XM6138-2 - Black - M

Other bills of materials, including WG6756-1, WG6756-2, WG6756-3, WG6756-4, WG6756-5, WG6756-6, WG6756-7, WG6756-8, VL6114-1, VL6114-2, VL6114-3, VL6114-4, VL6114-5, VL6114-6, VL6114-7, VL6114-8, XM6131-3, XM6131-4, XM6131-5, XM6131-6, XM6131-7, XM6131-8, are showed in Appendix.

Chapter 5 MRP SYSTEM

This chapter introduces the program algorithm of MRP, which is used in this research; next, program instruction is specified. Finally, the results are discussed.

5.1 Program Algorithm

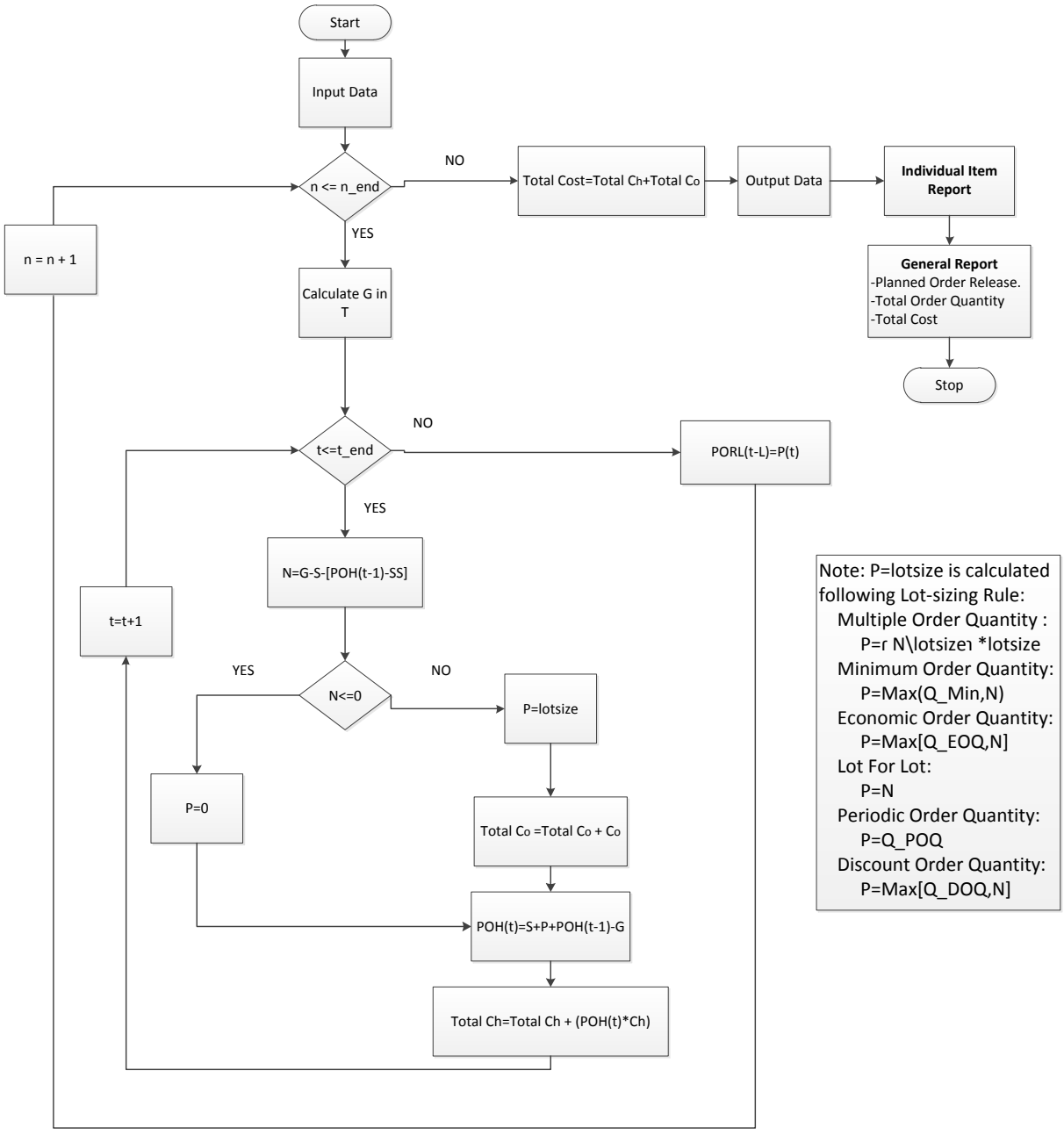


Figure 6: Program Algorithm of MRP

The MRP algorithm follows 11 steps below: [12]

Let n is the current level.

Step 1: Base on the existing information of requirement structure, MRP determines the gross requirement for all type of products in the current bills of material level over period of time.

Step 2: Determine the net requirement N(t) for items of level n at time t by formula:

$$N(t) = G(t) – S(t) – [POH(t – 1)\text{-}SS]$$

Where G(t) are the gross requirement at the current time t, S(t) are the scheduled receipts of the current time, POH(t-1) is the existing inventory of the previous time period and SS is safety stock requirement. If N(t)<=0, set N(t) = 0.

Step 3: According to the lot size rules for each typical item, the production requirement quantity P(t) is generated for all items at current bill of material level in the specific time t. If N(t) = 0 then P(t) = 0. Otherwise the order quantity is calculated follow the lot size policy.

Step 4: Calculate total ordering cost by calculate cumulative ordering cost Co at time t:

$$\text{Total Co} = \text{Total Co} + Co$$

Step 5: Calculate current stocks POH(t) for the items of level n at time t by formula:

$$POH(t) = S(t) + P(t) + POH(t – 1) – G(t)$$

Step 8: Calculate total holding cost (Ch) by calculate cumulative holding cost:

$$\text{Total Ch} = \text{Total Ch} + [POH(t)*Ch]$$

Step 9: Continue go to the next period of time (set t = t + 1) and go to step 1 until all the time period are finished (t=t_end), then go to step 10.

Step 10: Determine planned order releases PORL(t) of all production requirement in step 3 of items at current level. With lead time of item is L, then

$$PORL(t - L) = P(t)$$

Step 11: Stop when all the requirement of the items are determined in all level (n = n_end). Otherwise, continue to step 12.

Step 12: Determine the requirement of all items in the next bills of material level.

Step 13: Go to the next bills of material level, set n = n + 1, then go back to step 1.

Table 4: Program Variable

n	Level n
n_end	Lowest Level
G	Gross Requirement
T	Total time period
t	Time t
t_end	End of time
N	Net requirement
S	Schedule receipt
POH	Project on hand
SS	Safety stock
P	Planned oder receipt
lotsize	lot size
Co	Ordering Cost
Ch	Holding Cost
L	Lead time
PORL	Planned order release
Q_Min	Minimum order quantity
Q_EOQ	Economic order quantity
Q_DOQ	Discount order quantity

5.2 Program Instruction

5.4.1 Main Screen:

Figure 7: MRP main screen

Step	Menu Option	Function
1	Production Structure	Click on this button to open Product Structure Screen.
2	Master Production Schedules	Click on this button to Master Production Schedules Screen.
3	Schedule Receipts	Click on this button to Schedule Receipts Screen.
4	Reports	Click on this button to Reports.

5.4.2 Product Structure Screen:

5.4.1.1 Item Description:

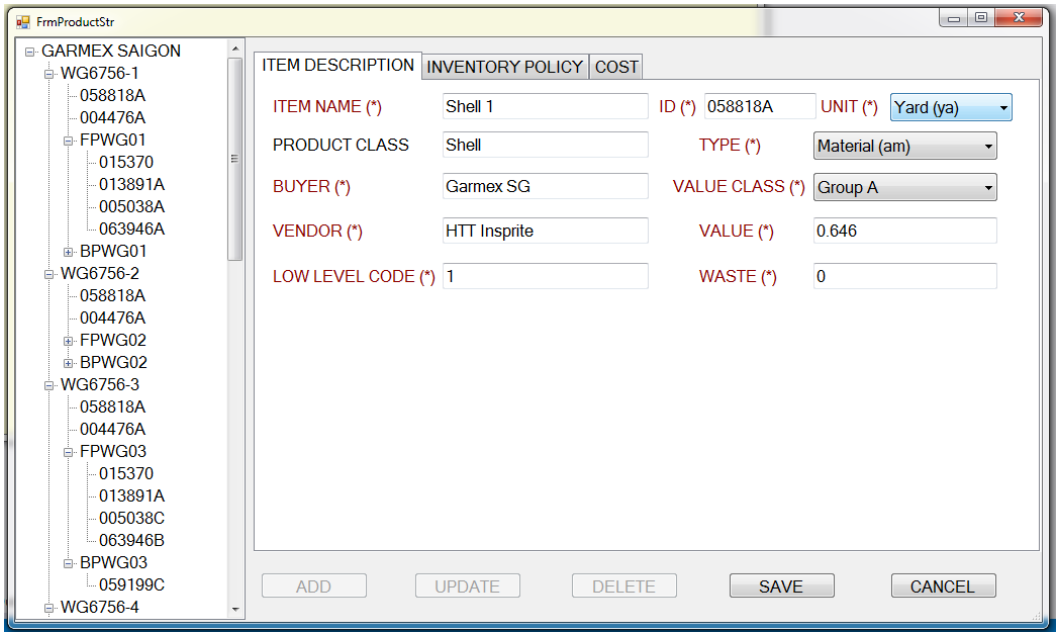


Figure 8: Item Description box

Add Item		
Step	Action	
1	Click on the upper level item on the Tree Box	
2	Click on the "Add" button.	
3	Choose Item Description Box	Fill in the blank " Item Name" to insert name of item
4		Fill in the blank " ID" to insert identification of item
5		Fill in the blank "Product Class" to insert class of item
6		Fill in the blank " Buyer" to insert Buyer Name
7		Fill in the blank "Vendor" to insert name of supplier
8		Fill in the blank "Low Level Code" to insert level of item
9		Click on button "Unit" to choose unit of item
10		Click on button "Type" to choose type of item
11		Click on button "Value Class" to choose unit of value class of item
12		Fill in the blank "Value" to insert value of item
13		Fill in the blank "Waste" to insert percentage of waste.

5.4.1.2 Inventory Policy:

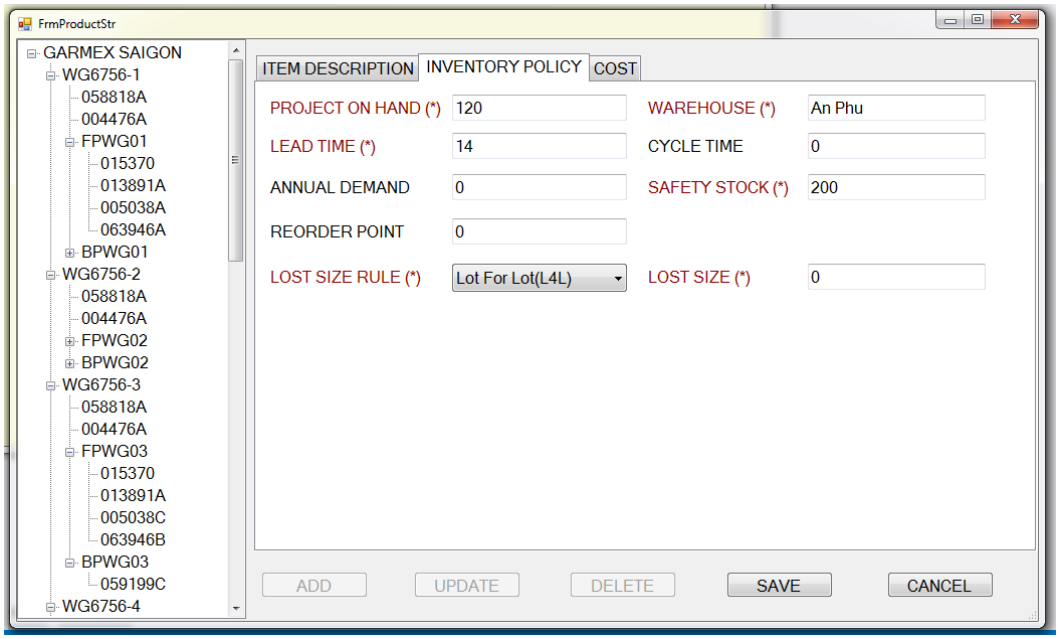


Figure 9 Inventory Policy box

Add Item		
Step	Action	
14	Chose Inventory Policy	Fill in the blank "Project on hand" to insert inventory on hand.
15		Fill in the blank "Warehouse" to insert name of warehouse.
16		Fill in the blank "Lead Time" to insert lead time of item.
17		Fill in the blank "Cycle Time" to insert Cycle time of finished product or assembly item.
18		Fill in the blank "Annual Demand" to insert annual demand of item.
19		Fill in the blank "Reorder Point" to insert reorder point of inventory.
20		Fill in the blank "Safety Stock" to insert safety stock.
21		Fill in the blank "Lot Size" to insert lot size of item.
22		Click on button "Lot Size Rule" to choose Lot-sizing Rule used for calculation

5.4.1.3 Cost:

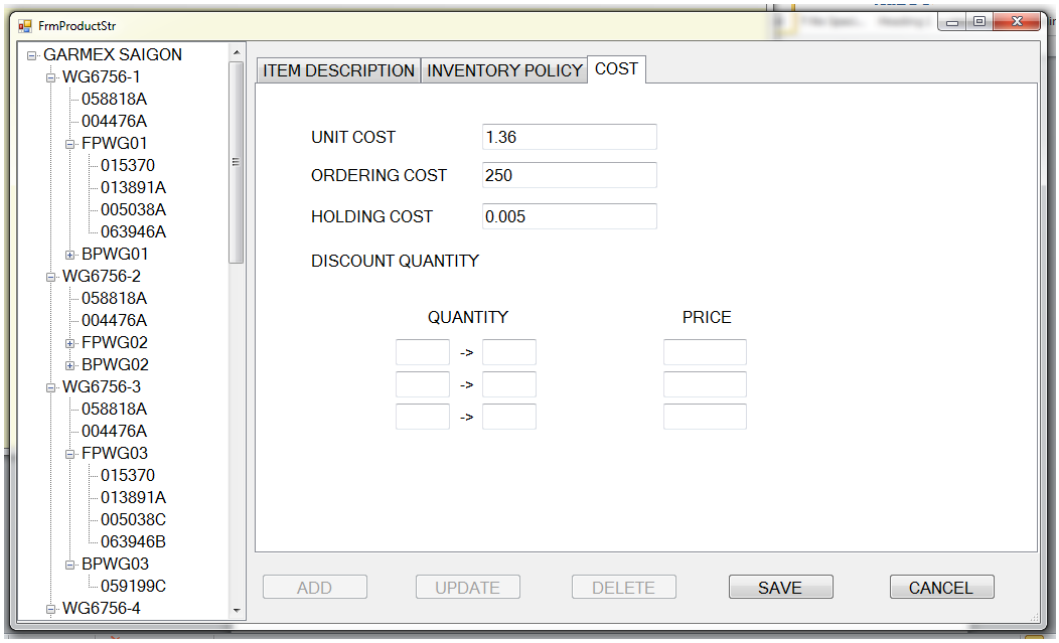


Figure 10 Cost box

Add Item		
Step	Action	
	Choose Cost	Without Discount Quantity
23		Fill in the blank "Unit Cost" to insert Unit Cost of item.
24		Fill in the blank "Ordering Cost" to insert ordering cost of item.
25		Fill in the blank "Holding Cost" to insert holding cost of item.
		Discount Quantity is offered
23		Fill in the blank "Ordering Cost" to insert ordering cost of item.
24		Fill in the blank "Holding Cost" to insert holding cost of item.
25		Fill in the blank "Quantity" to insert range of quantity received discounts.
26		Fill in the blank "Price" to insert price corresponding with the range of discount quantity
27	Click on "Save" button to save the item information.	

Update Item	
Step	Action
1	Click on the item on the Tree Box
2	Click on "Update" button to update information of item
3	Update necessary information
4	Click on "Save" button to save the item information.

Delete Item	
Step	Action
1	Click on the item on the Tree Box
2	Click on "Delete" button to delete item information

5.4.3 Master Production Screen:

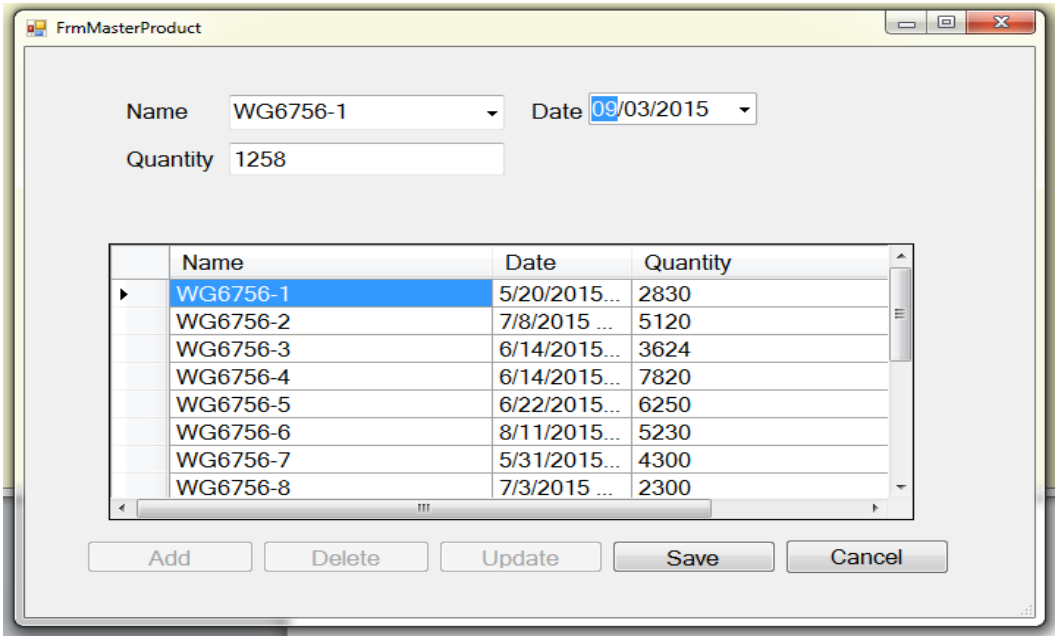


Figure 11 Master Production Screen

Add Master Production Information	
Step	Action
1	Click on "Name" , choose item ID to add information of master production.
2	Click on "Date", choose date of finished product
3	Fill in the blank "Quantity" to insert quantity finished product.
4	Click on "Save" button to save the item information.

Update Master Production Information	
Step	Action
1	Click on Item ID which is needed to update showed in Information Box below
2	Click on "Update" button and update information.
4	Click on "Save" button to save the item information.

Delete Master Production Information	
Step	Action
1	Click on the item on the Information box
2	Click on "Delete" button to delete item information

5.4.4 Schedule Receipts Screen:

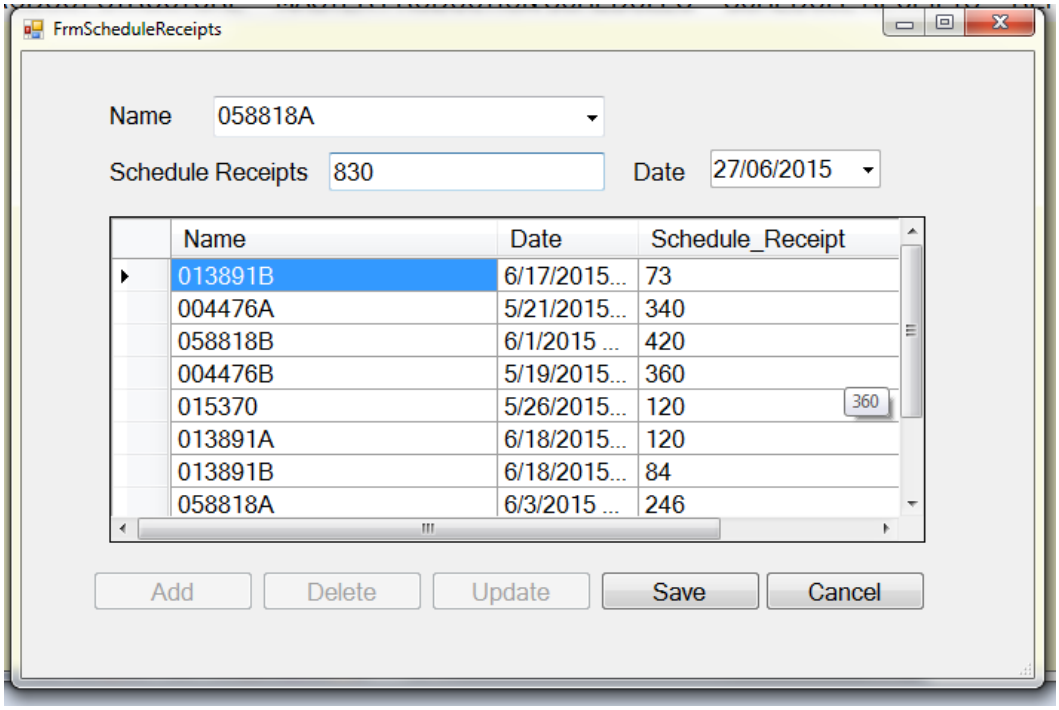


Figure 12 Schedule Receipts Screen

Add schedule receipts	
Step	Action
1	Click on "Name" , choose item ID to add information of schedule receipts
2	Click on "Date", choose date receipts item
3	Fill in the blank "Quantity" to insert quantity of item received
4	Click on "Save" button to save the item information.

Updated schedule receipts	
Step	Action
1	Click on Item ID which is needed to update showed in Information Box below
2	Click on "Update" button and update information.
4	Click on "Save" button to save the item information.

Delete schedule receipts	
Step	Action
1	Click on the item on the Information box
2	Click on "Delete" button to delete item information

5.4.5 Report Screen:

5.4.5.1 Item Report:

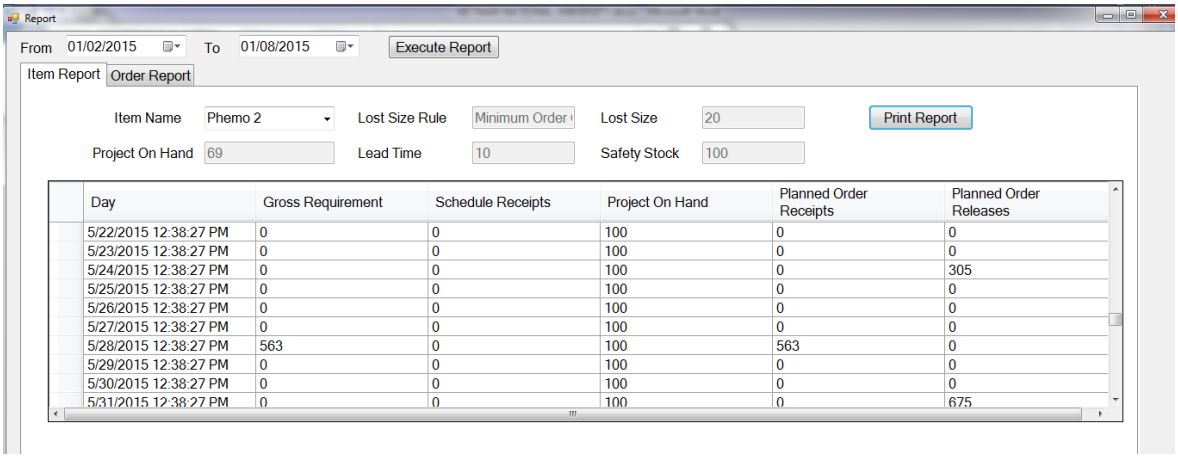


Figure 13 Item Report

Print Report	
Step	Action
1	Click on "From" , choose starting date of calculation
2	Click on "To" , choose end date of calculation
3	Click "Execute Report" to calculate MRP.
	Item Report
4	Click on "Item Report"
5	Click "Item Name" to choose name of item.
6	Click on "Print Report" to show item report

5.4.5.2 Order Report:

a/

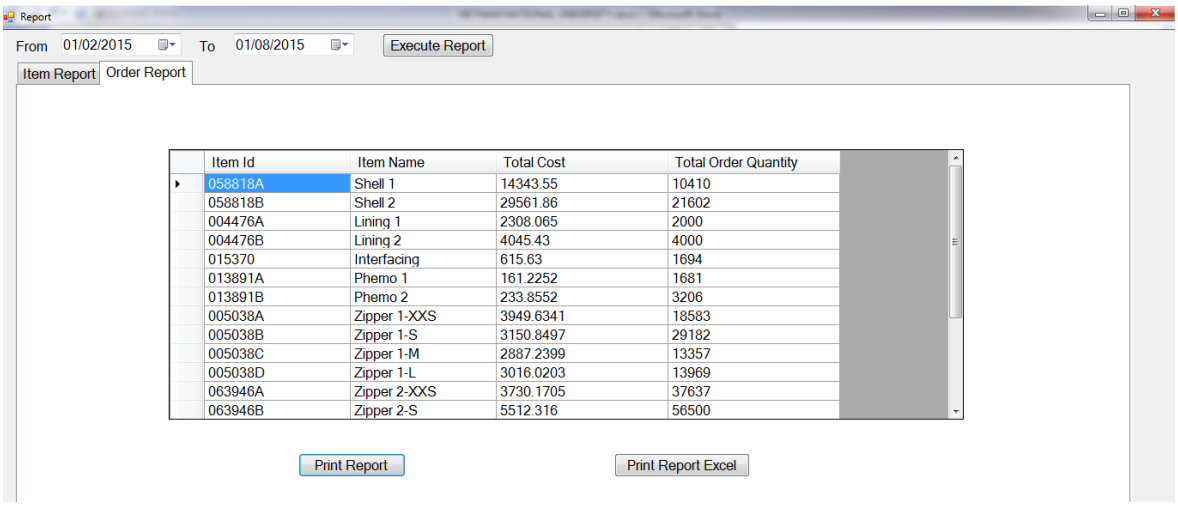


Figure 14 Order Report

	Order Report
7	Click on "Order Report"
8	Click on "Print report" to print total calculation report
9	Click on "Print Excel Report" to print general report in excel file

b/

Table 5 General Report

General Report	Date : 7/6/2015			
ItemID	**ItemName**	**Quantity**	**Unit**	**ReleasedDate**
058818A	058818A	571	Yard (ya)	3/3/2015
013061B	013061B	1178	Piece (pc)	3/3/2015
29909	29909	1232	Piece (pc)	3/3/2015
62542	62542	1185	Piece (pc)	3/3/2015
980001A	980001A	1280	Piece (pc)	3/3/2015
980010B	980010B	1309	Piece (pc)	3/3/2015
200007B	200007B	3	Yard (ya)	3/3/2015
005038C	005038C	435	Piece (pc)	3/5/2015
063946B	063946B	1015	Piece (pc)	3/5/2015
059199C	059199C	415	Piece (pc)	3/5/2015
3299	3299	505	Piece (pc)	3/5/2015
5274	5274	505	Piece (pc)	3/5/2015

5.3 Result

Table 6 General Report

General Report	Date : 7/6/2015			
ItemID	ItemName	Quantity	Unit	ReleasedDate
058818A	058818A	571	Yard (ya)	3/3/2015
013061B	013061B	1178	Piece (pc)	3/3/2015
29909	29909	1232	Piece (pc)	3/3/2015
62542	62542	1185	Piece (pc)	3/3/2015
980001A	980001A	1280	Piece (pc)	3/3/2015
980010B	980010B	1309	Piece (pc)	3/3/2015
200007B	200007B	3	Yard (ya)	3/3/2015
005038C	005038C	435	Piece (pc)	3/5/2015
063946B	063946B	1015	Piece (pc)	3/5/2015
059199C	059199C	415	Piece (pc)	3/5/2015
3299	3299	505	Piece (pc)	3/5/2015
5274	5274	505	Piece (pc)	3/5/2015
55257	55257	513	Piece (pc)	3/5/2015
62674	62674	505	Piece (pc)	3/5/2015
55257	55257	1626	Piece (pc)	3/7/2015
980001A	980001A	1297	Piece (pc)	3/8/2015
058818A	058818A	327	Yard (ya)	3/10/2015
013061C	013061C	428	Piece (pc)	3/10/2015
013061D	013061D	491	Piece (pc)	3/10/2015
29909	29909	505	Piece (pc)	3/10/2015
62542	62542	505	Piece (pc)	3/10/2015
980001A	980001A	1626	Piece (pc)	3/10/2015
980001B	980001B	452	Piece (pc)	3/10/2015
980010C	980010C	452	Piece (pc)	3/10/2015
200007C	200007C	139	Yard (ya)	3/10/2015
013891A	013891A	63	Yard (ya)	3/12/2015
005038D	005038D	907	Piece (pc)	3/12/2015
063946C	063946C	2158	Piece (pc)	3/12/2015
059199D	059199D	1055	Piece (pc)	3/12/2015
3299	3299	1142	Piece (pc)	3/12/2015
5274	5274	1142	Piece (pc)	3/12/2015
55257	55257	1159	Piece (pc)	3/12/2015
62674	62674	1142	Piece (pc)	3/12/2015
980001A	980001A	513	Piece (pc)	3/12/2015
55257	55257	1199	Piece (pc)	3/14/2015

058818A	058818A	739	Yard (ya)	3/17/2015
004476A	004476A	80	Yard (ya)	3/17/2015
013061A	013061A	1079	Piece (pc)	3/17/2015
29909	29909	1142	Piece (pc)	3/17/2015
62542	62542	1142	Piece (pc)	3/17/2015
980001A	980001A	1199	Piece (pc)	3/17/2015
980001B	980001B	1144	Piece (pc)	3/17/2015
980010D	980010D	1034	Piece (pc)	3/17/2015
7016	7016	1721	Yard (ya)	3/17/2015
200007A	200007A	194	Yard (ya)	3/17/2015
6492	6492	15	Yard (ya)	3/17/2015
15370	15370	11	Yard (ya)	3/19/2015
013891B	013891B	30	Yard (ya)	3/19/2015
005038A	005038A	767	Piece (pc)	3/19/2015
063946A	063946A	1802	Piece (pc)	3/19/2015
059199A	059199A	804	Piece (pc)	3/19/2015
3299	3299	842	Piece (pc)	3/19/2015
5274	5274	882	Piece (pc)	3/19/2015
55257	55257	887	Piece (pc)	3/19/2015
62674	62674	842	Piece (pc)	3/19/2015
980001A	980001A	1159	Piece (pc)	3/19/2015
55257	55257	1503	Piece (pc)	3/21/2015
058818B	058818B	695	Yard (ya)	3/24/2015
013061A	013061A	902	Piece (pc)	3/24/2015
013061B	013061B	1438	Piece (pc)	3/24/2015
29909	29909	812	Piece (pc)	3/24/2015
62542	62542	842	Piece (pc)	3/24/2015
980001A	980001A	934	Piece (pc)	3/24/2015
980010A	980010A	772	Piece (pc)	3/24/2015
7016	7016	2278	Yard (ya)	3/24/2015
200007B	200007B	269	Yard (ya)	3/24/2015
6492	6492	68	Yard (ya)	3/24/2015
15370	15370	50	Yard (ya)	3/26/2015
013891B	013891B	137	Yard (ya)	3/26/2015
005038B	005038B	1215	Piece (pc)	3/26/2015
063946B	063946B	2469	Piece (pc)	3/26/2015
059199B	059199B	1205	Piece (pc)	3/26/2015
3299	3299	1249	Piece (pc)	3/26/2015
5274	5274	1249	Piece (pc)	3/26/2015
55257	55257	1267	Piece (pc)	3/26/2015
62674	62674	1249	Piece (pc)	3/26/2015

Table 5 above shows general report to inform the quantity requirement and order date of specific items which is planned by MRP, base on the Production Master Schedule. This is just a part of report, full report will be showed in Appendix.

Table 7: Inventory Improvement (six months period)

Material ID	Inventory		Difference (pc)
	Without MRP	Applied MRP	
360121	15368	13226	2142
760285	62030	51934	10096
003299	67520	76630	-9110
004476A	14586	9846	4740
004476B	13202	14052	-850
005038A	18025	14364	3661
005038B	13021	12058	963
005038C	5210	6010	-800
005038D	7830	6630	1200
005274	72560	74133	-1573
005542A	9320	4520	4800
005542B	1502	843	659
006492	1263	1107	156
007016	35468	37471	-2003
013061A	28920	28562	358
013061B	58320	46222	12098
013061C	38760	23725	15035
013061D	2843	4707	-1864
013891A	20520	15430	5090
013891B	16805	12476	4329
015370	52120	23020	29100
029906	32025	25921	6104
029909	62030	40994	21036
029913	22692	24691	-1999
053817A	19350	15364	3986
055257	86250	81301	4949
058818A	92356	76466	15890
058818B	31206	29348	1858
059197	24350	25921	-1571
059199A	39620	35372	4248

059199B	14603	13105	1498
059199C	15861	16257	-396
059199D	8630	6623	2007
059200	39650	25921	13729
059200A	8260	7477	783
059200B	12302	9486	2816
059200C	9250	8983	267
062542	82600	74133	8467
062674	41250	40994	256
063946A	26351	23561	2790
063946B	36245	29638	6607
063946C	9658	10875	-1217
064978A	7825	7654	171
064978B	10042	9711	331
064978C	13504	9195	4309
200007A	1890	1914	-24
200007B	1670	1556	114
200007C	3206	964	2242
340001	28320	24877	3443
380083	82650	56559	26091
500007	63250	30703	32547
500058A	12560	11724	836
500058B	13134	10142	2992
500058C	8600	4080	4520
520010	29200	30703	-1503
760161A	33502	27801	5701
760161B	2563 0	19190	6440
760161C	34500	14515	19985
760561A	26125	21177	4948
760561B	24320	17216	7104
760561C	13746	10671	3075
980001A	132640	81301	51339
980001B	13806	8803	5003
980010A	120350	106926	13424
980010B	24050	18301	5749
980010C	13600	10006	3594
980010D	8630	4707	3923
		Total	376686

The table above shows the improvement of inventory after 6 months applied MRP. The difference is calculated as the inventory before using MRP minus the inventory after applying MRP. As we can see, the negative numbers in Difference column express the shortage of inventory components, which is eliminated when MRP is used. The result also points out that application of MRP decreased 376686 units in inventory while satisfying the material requirements for production.

Moreover, MRP program also plans delivery schedules and order quantity, which support planners to control the lowest inventory level, reduce cost and maintain availability of materials for production.

Chapter 6 CONCLUSION

This study case applied Material Requirement Planning for Garmex Saigon Company in order to help company simultaneous meet requirements of customer and reduce the production cost as much as possible through material plans.

The results after using MRP proved that MRP is a effective tools for planner when it not only generates exact order schedule for materials, but also controls time of delivery and order activities, eliminates shortage of inventory and ensures material available when production needs.

By consequences, the Material Requirement Planning system is the most effective solution for planning problem in order to apply for whole Garmex Saigon Company.

REFERENCES

[1] Carol Ptak and Chad Smith, *Orlicky's Material Requirement Planning 3rd Edition*, 2011.

[2] H.Sadeghi, A.Makui and M.Heydari, "A simulation method for Material Requirement Planning supply dependent demand and uncertainty lead- time", *African Journal of Business Management*,28th February 2014.

[3] Gerhard Plenert, "Focusing material requirement planning (MRP) towards performance", *European Journal of Operational Research*, 1st September 1998.

[4] DavoodGharakhani, "Optimization of material requirement planning by Goal programming model", *Asian Journal of Management Research*, 2011.

[5] R.B.Cooper and R.W.Zmud, "Material Requirements Planning System Infusion", *OMEGA Int J of MgmentSci*, Vol 17,No 5,pp 477-481,1989.

[6] Eduard Shevtchenko,"MRP Systems Research", in *4th International DAAAM Conference*, 29-30th April 2004.

[7] Patty W. Cheng, "Effective Use of MRP- Type Computer Systems to Support Manufacturing", 19th March 1997.

[8] Edmund W. Schuster, Stuart J.Allen and Michael P.D'Itri, "Capacitated Materials Requirements Planning and its Application in the Process industries".

[9] K.E. Kurbel, "Enterprise Resource Planning and Supply Chain Management", Progress in IS , Heidelberg 2013.

[10] Yolanda Masnita and T.A. Mahdani, “ Factors of MRP Implementation in Manufacturer for Small and Medium- Sized Firms”, in The 2012 International Conference on Business and Mangagement, 6-7th September 2012.

[11] Lutfu Sabansua, “Information technologies and material requirement planning in supply chain management as a basic for a new model”, Bulgarian Journal of Science and Education Policy (BJSEP), Volume 4, Number 2, 2010

[12] D. B. S. Nikoletseas, "An efficient implementation of Material Requirements Planning by using cache memory," 1994.

[13] Microsoft Visual Studio. 2014.[online] available: www.microsoft.com/Visual_studio.

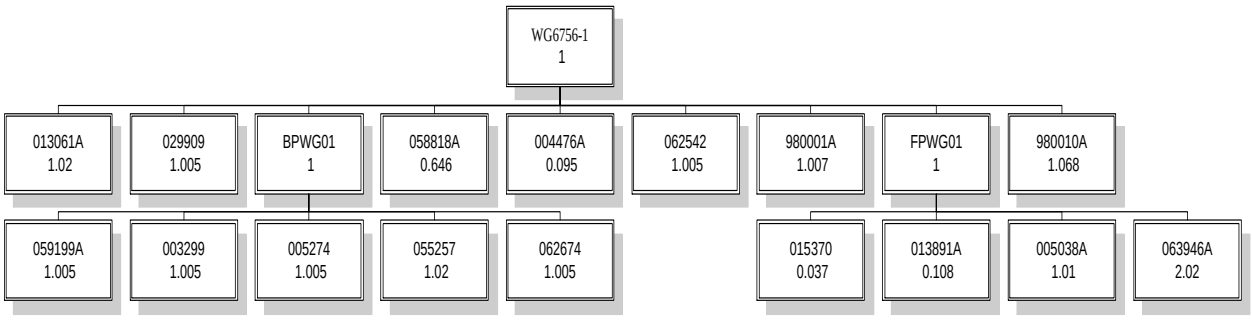
[14] Garmex Saigon Jst, 2015. [Online] available: www.garmexsaigon-gmc.com

[15] MySQL 5.0 Reference manual.2013[Online] available: http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html

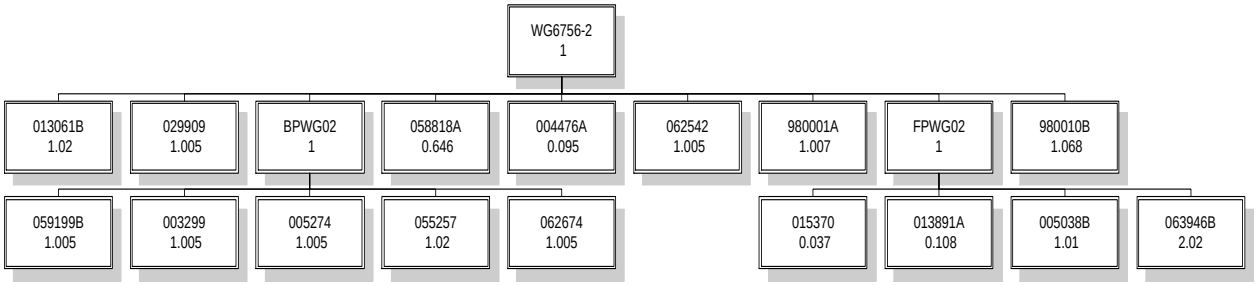
Appendix

Bills of materials

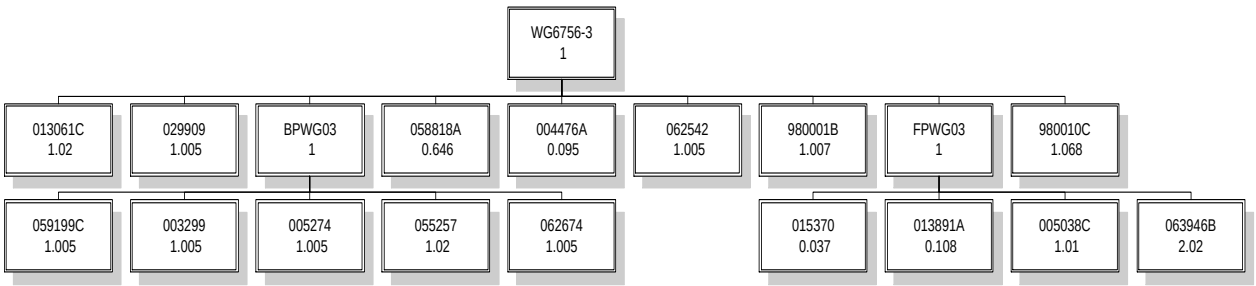
Style: WG6756



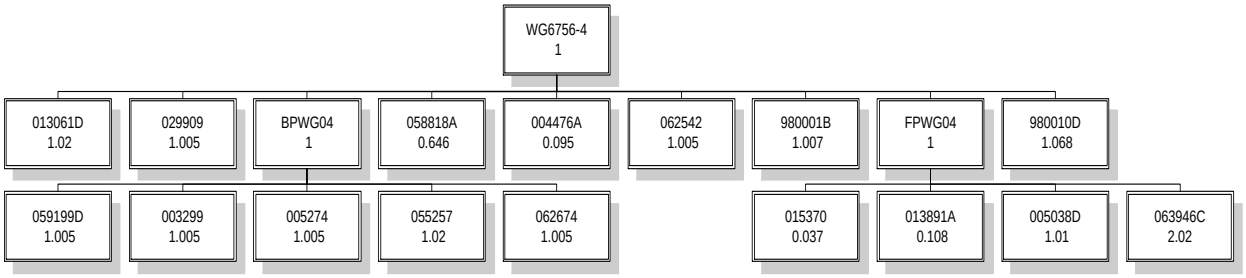
BOM of WG6756-1 – Purple Lotus - XXS



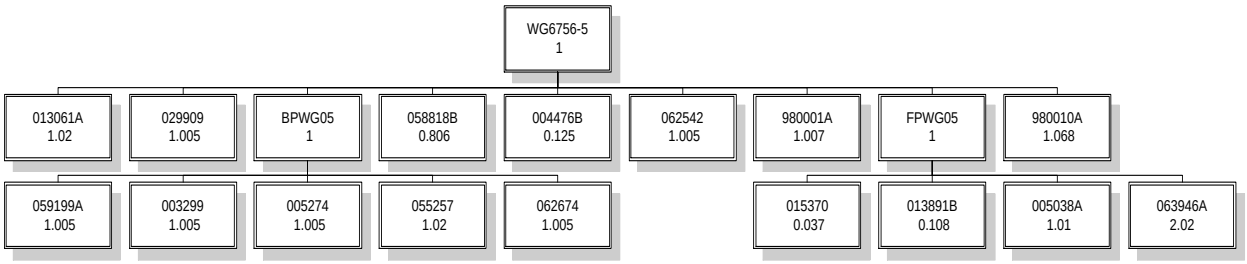
BOM of WG6756-2 – Purple Lotus - XS



BOM of WG6756-3 – Purple Lotus - M

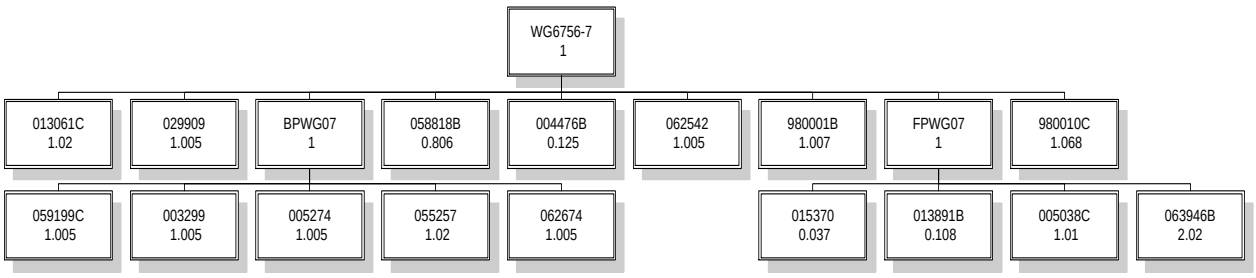


BOM of WG6756-4 – Purple Lotus - L

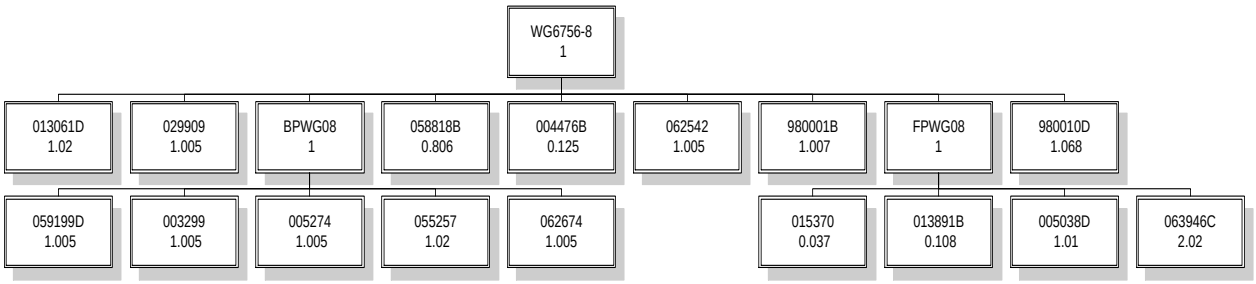


BOM of WG6756-5 – Bright Rose - XXS

BOM of WG6756-6 – Bright Rose - XS

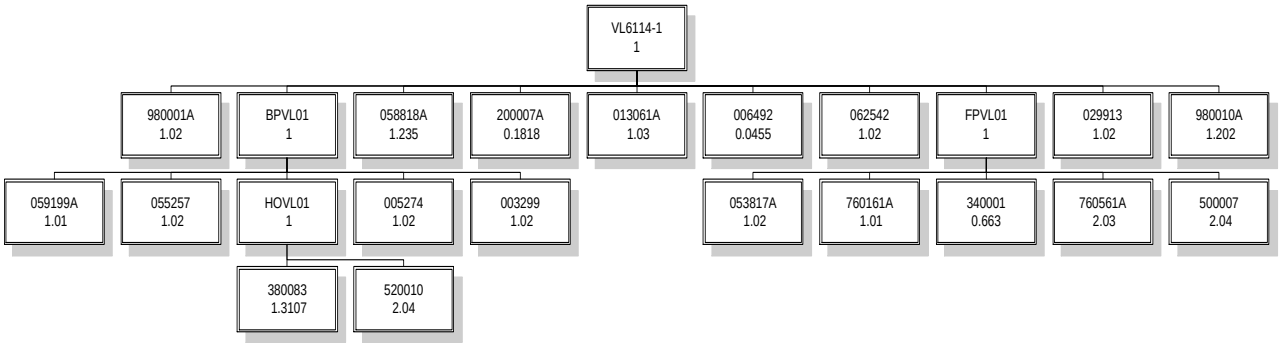


BOM of WG6756-7 – Bright Rose - M

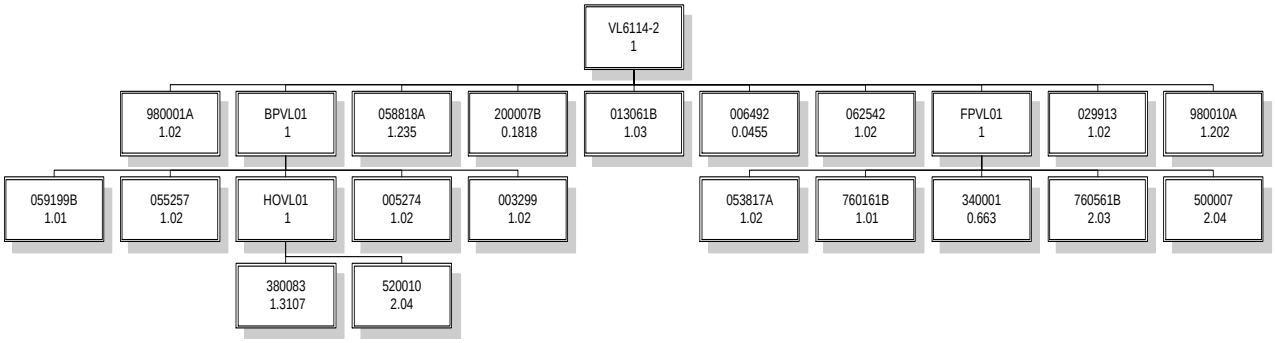


BOM of WG6756-8 – Bright Rose - L

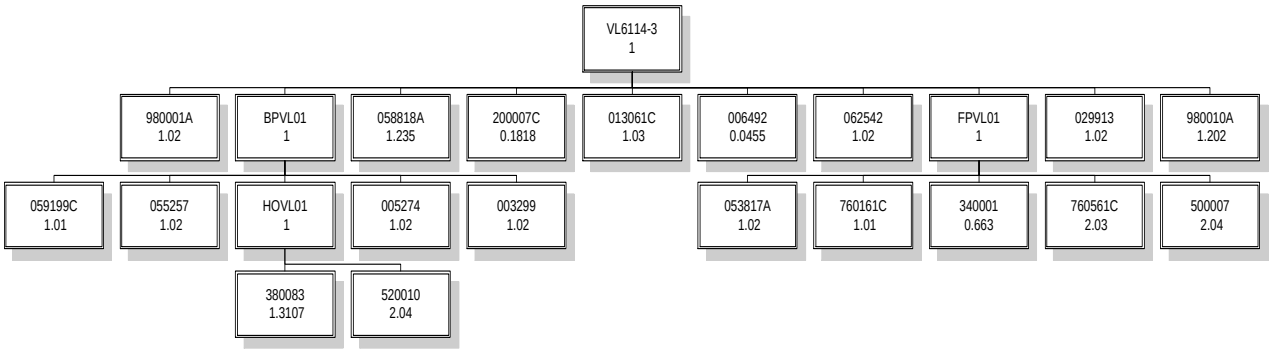
Style: VL6114



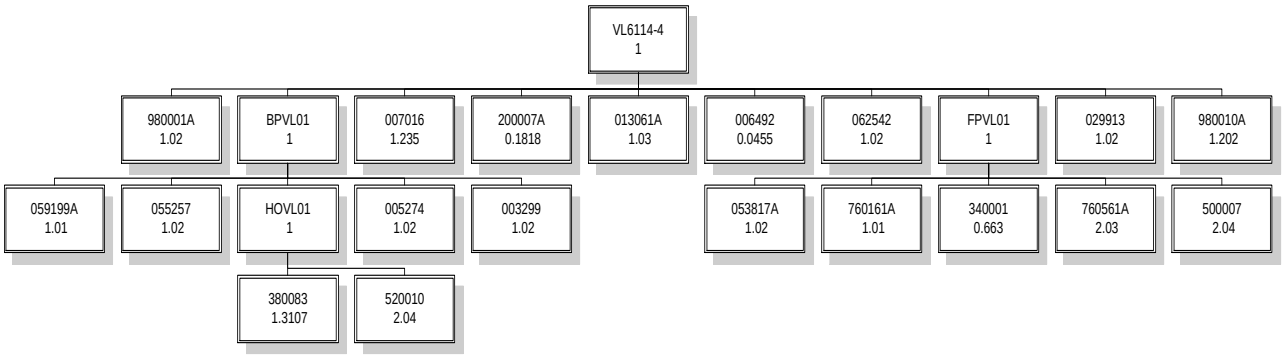
BOM of VL6114-1 – Black - S



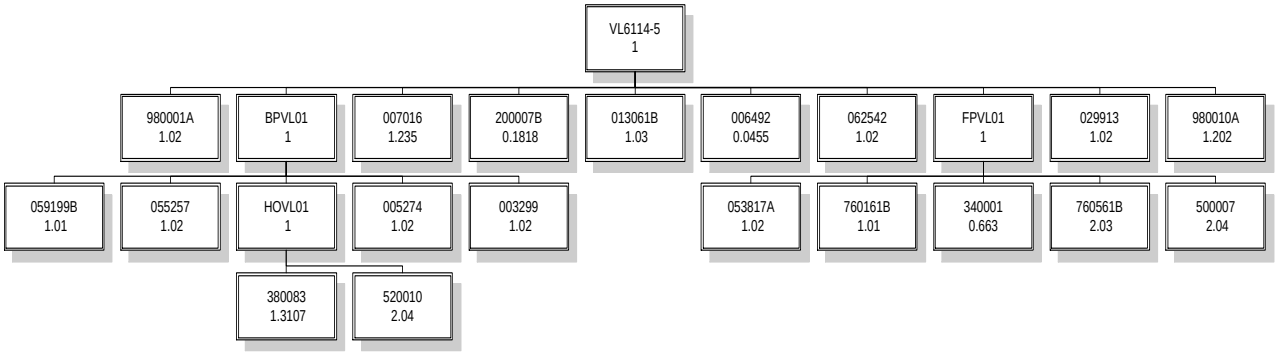
BOM of VL6114-2 – Black - M



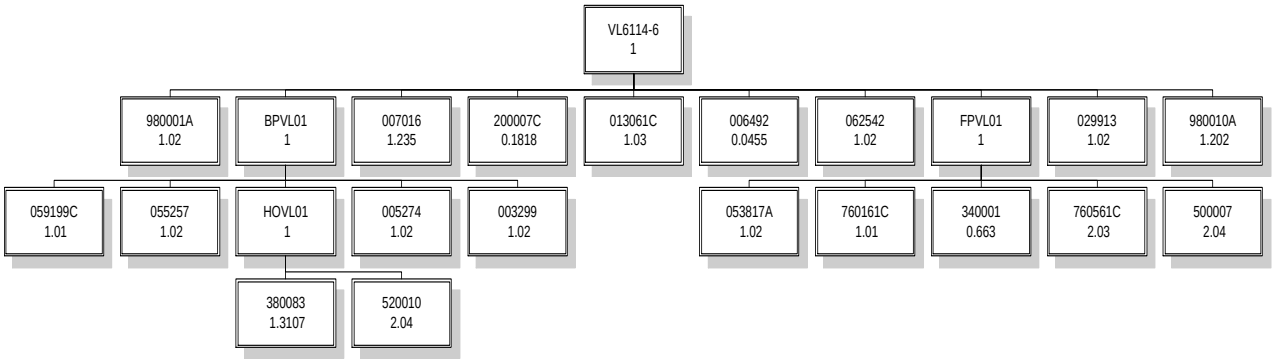
BOM of VL6114-3 – Black - L



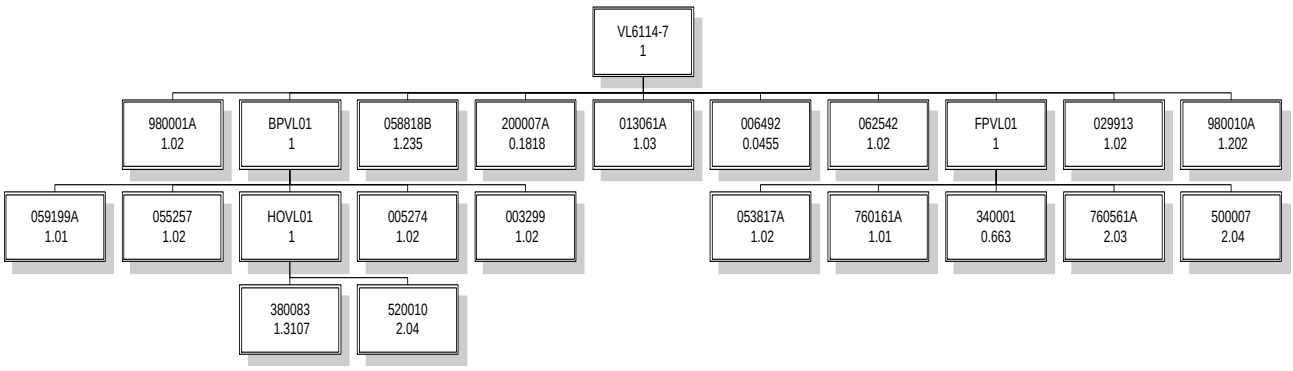
BOM of VL6114-4 –Charcoal Heather- S



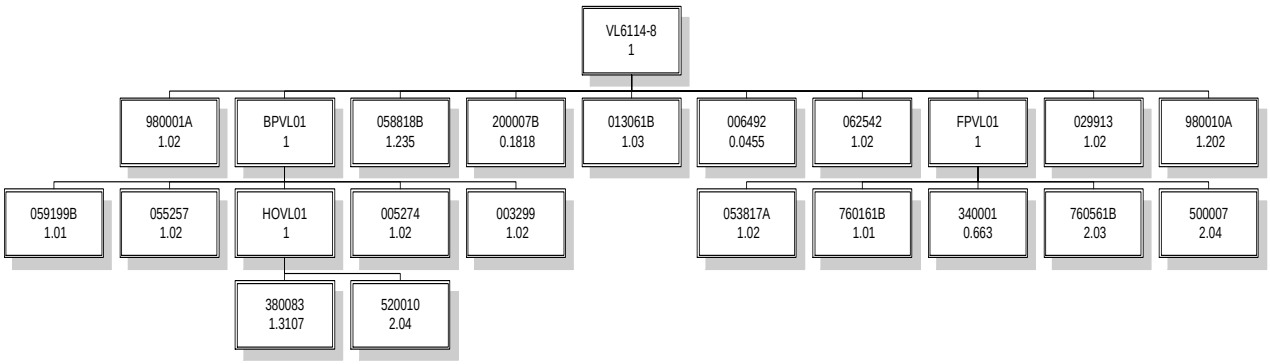
BOM of VL6114-5 –Charcoal Heather- M



BOM of VL6114-6 –Charcoal Heather- L

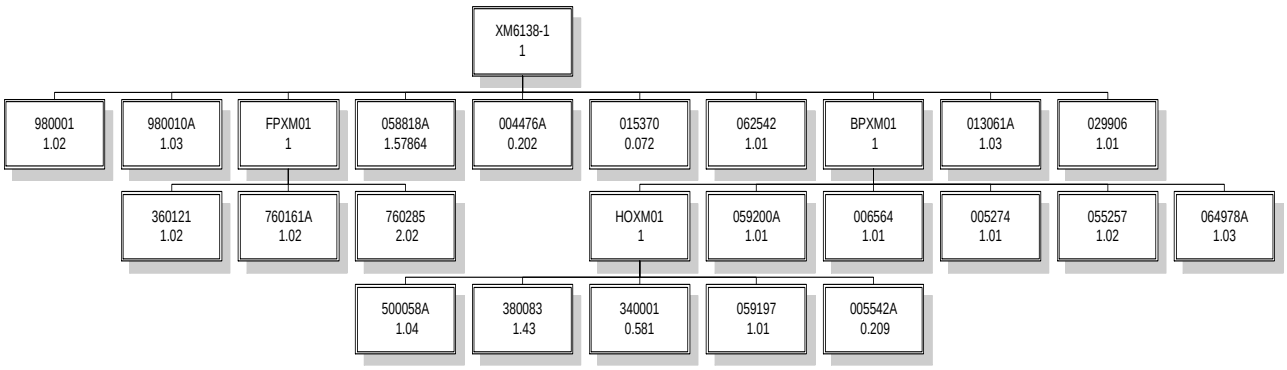


BOM of VL6114-7 –Columbia Navy- S

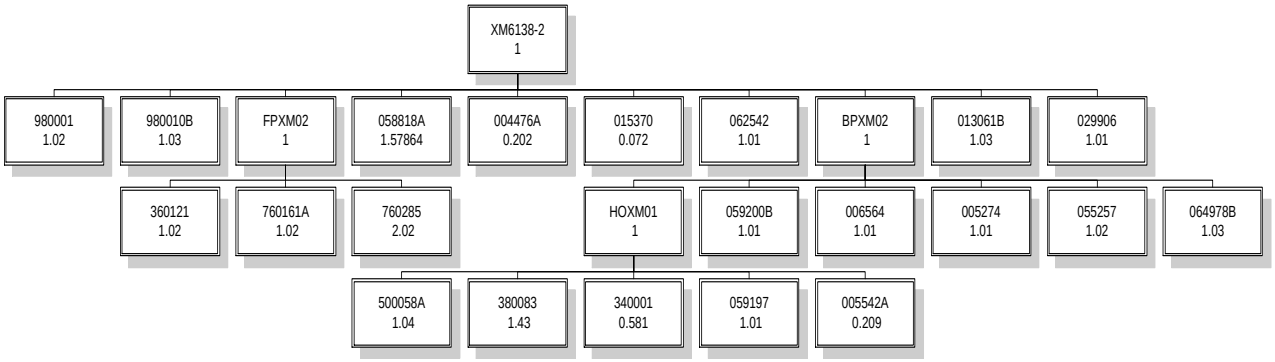


BOM of VL6114-8 –Columbia Navy- M

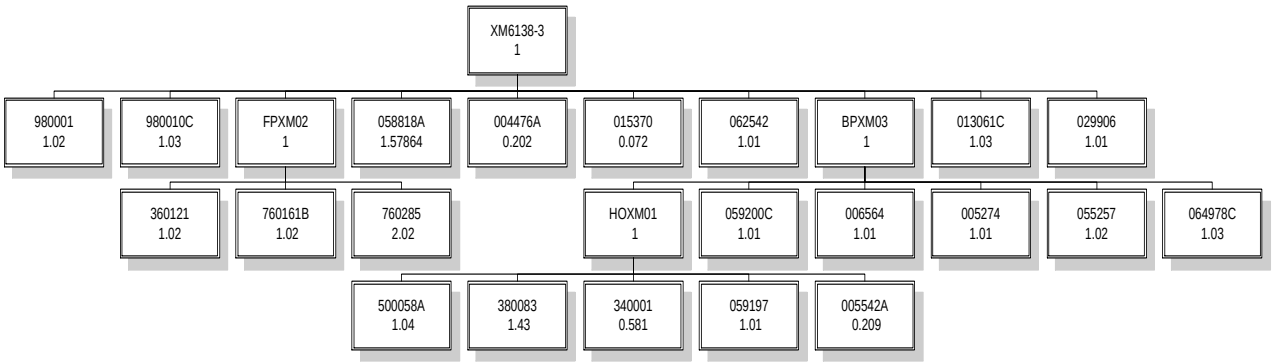
Style: XM6138



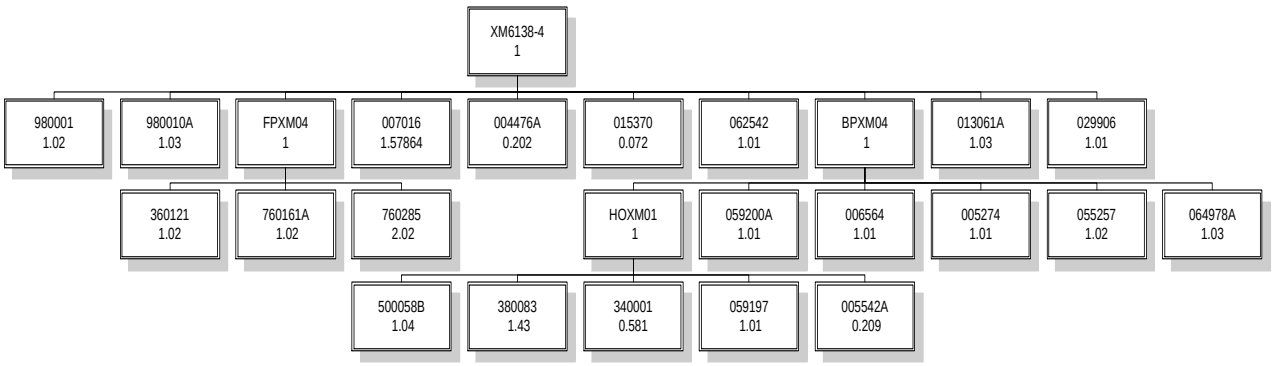
BOM of XM6138-1 – Black - S



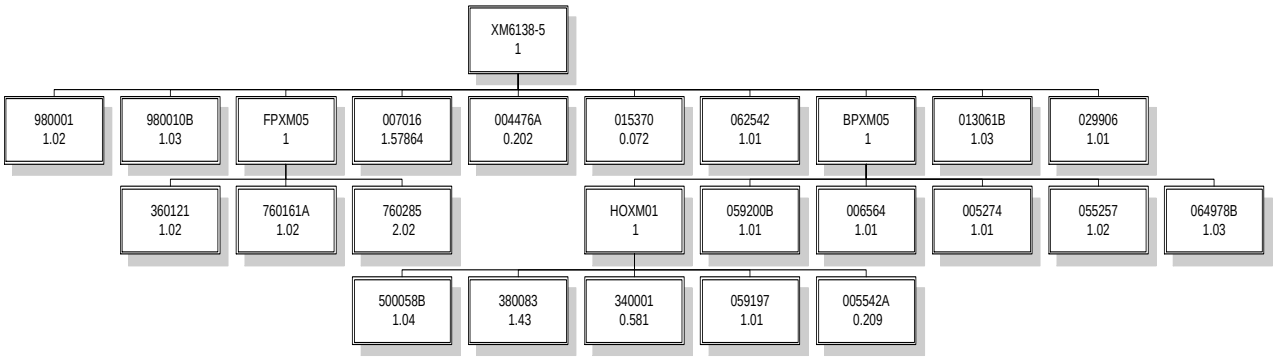
BOM of XM6138-2 – Black - M



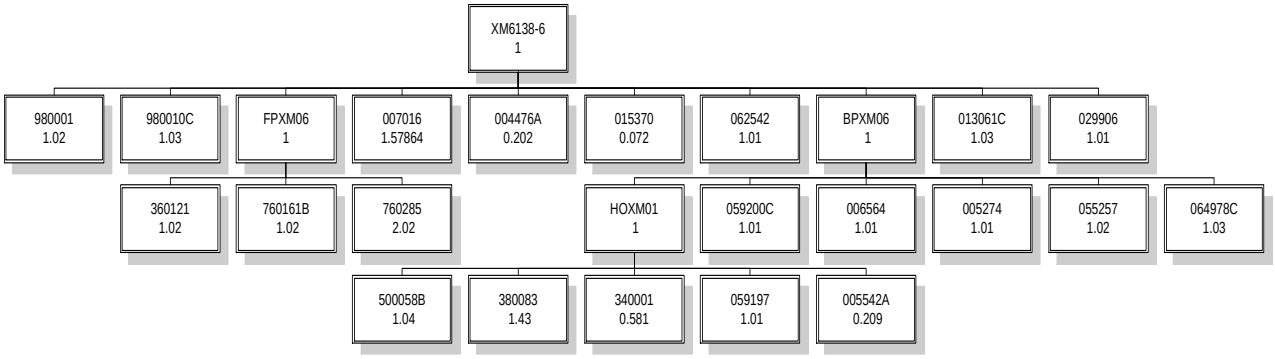
BOM of XM6138-3 – Black - L



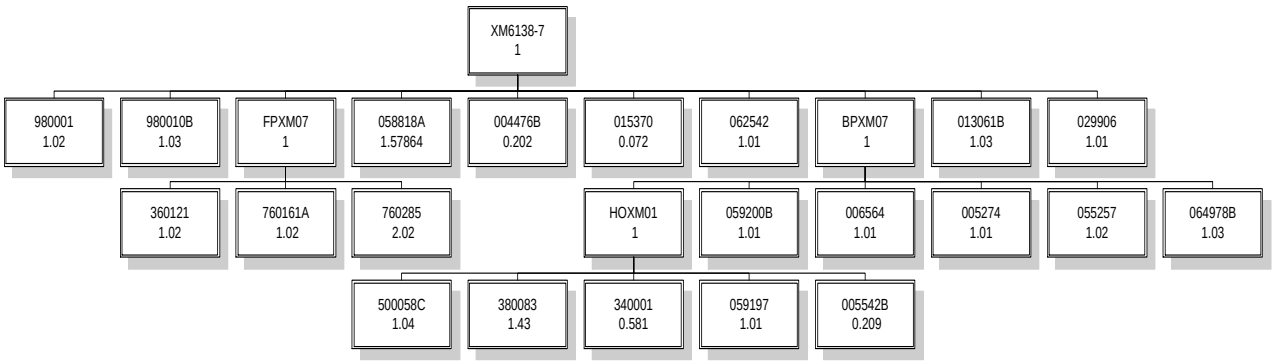
BOM of XM6138-4 – Charcoal Heather - S



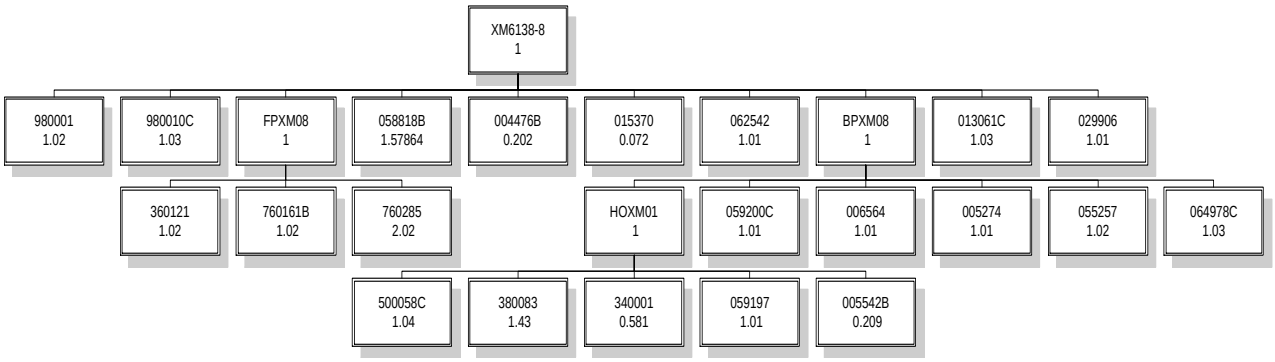
BOM of XM6138-5 – Charcoal Heather - M



BOM of XM6138-6 – Charcoal Heather - L



BOM of XM6138-7 – Columbia Navy - M



BOM of XM6138-8 – Columbia Navy - L

MRP Report:

General Report	Date : 7/6/2015			
ItemID	ItemName	Quantity	Unit	ReleasedDate
058818A	058818A	571	Yard (ya)	3/3/2015
013061B	013061B	1178	Piece (pc)	3/3/2015
29909	29909	1232	Piece (pc)	3/3/2015
62542	62542	1185	Piece (pc)	3/3/2015
980001A	980001A	1280	Piece (pc)	3/3/2015
980010B	980010B	1309	Piece (pc)	3/3/2015
200007B	200007B	3	Yard (ya)	3/3/2015
005038C	005038C	435	Piece (pc)	3/5/2015
063946B	063946B	1015	Piece (pc)	3/5/2015
059199C	059199C	415	Piece (pc)	3/5/2015
3299	3299	505	Piece (pc)	3/5/2015
5274	5274	505	Piece (pc)	3/5/2015
55257	55257	513	Piece (pc)	3/5/2015
62674	62674	505	Piece (pc)	3/5/2015
55257	55257	1626	Piece (pc)	3/7/2015
980001A	980001A	1297	Piece (pc)	3/8/2015
058818A	058818A	327	Yard (ya)	3/10/2015
013061C	013061C	428	Piece (pc)	3/10/2015
013061D	013061D	491	Piece (pc)	3/10/2015
29909	29909	505	Piece (pc)	3/10/2015
62542	62542	505	Piece (pc)	3/10/2015
980001A	980001A	1626	Piece (pc)	3/10/2015
980001B	980001B	452	Piece (pc)	3/10/2015
980010C	980010C	452	Piece (pc)	3/10/2015
200007C	200007C	139	Yard (ya)	3/10/2015
013891A	013891A	63	Yard (ya)	3/12/2015
005038D	005038D	907	Piece (pc)	3/12/2015
063946C	063946C	2158	Piece (pc)	3/12/2015
059199D	059199D	1055	Piece (pc)	3/12/2015
3299	3299	1142	Piece (pc)	3/12/2015
5274	5274	1142	Piece (pc)	3/12/2015
55257	55257	1159	Piece (pc)	3/12/2015
62674	62674	1142	Piece (pc)	3/12/2015
980001A	980001A	513	Piece (pc)	3/12/2015
55257	55257	1199	Piece (pc)	3/14/2015
058818A	058818A	739	Yard (ya)	3/17/2015

004476A	004476A	80	Yard (ya)	3/17/2015
013061A	013061A	1079	Piece (pc)	3/17/2015
29909	29909	1142	Piece (pc)	3/17/2015
62542	62542	1142	Piece (pc)	3/17/2015
980001A	980001A	1199	Piece (pc)	3/17/2015
980001B	980001B	1144	Piece (pc)	3/17/2015
980010D	980010D	1034	Piece (pc)	3/17/2015
7016	7016	1721	Yard (ya)	3/17/2015
200007A	200007A	194	Yard (ya)	3/17/2015
6492	6492	15	Yard (ya)	3/17/2015
15370	15370	11	Yard (ya)	3/19/2015
013891B	013891B	30	Yard (ya)	3/19/2015
005038A	005038A	767	Piece (pc)	3/19/2015
063946A	063946A	1802	Piece (pc)	3/19/2015
059199A	059199A	804	Piece (pc)	3/19/2015
3299	3299	842	Piece (pc)	3/19/2015
5274	5274	882	Piece (pc)	3/19/2015
55257	55257	887	Piece (pc)	3/19/2015
62674	62674	842	Piece (pc)	3/19/2015
980001A	980001A	1159	Piece (pc)	3/19/2015
55257	55257	1503	Piece (pc)	3/21/2015
058818B	058818B	695	Yard (ya)	3/24/2015
013061A	013061A	902	Piece (pc)	3/24/2015
013061B	013061B	1438	Piece (pc)	3/24/2015
29909	29909	812	Piece (pc)	3/24/2015
62542	62542	842	Piece (pc)	3/24/2015
980001A	980001A	934	Piece (pc)	3/24/2015
980010A	980010A	772	Piece (pc)	3/24/2015
7016	7016	2278	Yard (ya)	3/24/2015
200007B	200007B	269	Yard (ya)	3/24/2015
6492	6492	68	Yard (ya)	3/24/2015
15370	15370	50	Yard (ya)	3/26/2015
013891B	013891B	137	Yard (ya)	3/26/2015
005038B	005038B	1215	Piece (pc)	3/26/2015
063946B	063946B	2469	Piece (pc)	3/26/2015
059199B	059199B	1205	Piece (pc)	3/26/2015
3299	3299	1249	Piece (pc)	3/26/2015
5274	5274	1249	Piece (pc)	3/26/2015
55257	55257	1267	Piece (pc)	3/26/2015
62674	62674	1249	Piece (pc)	3/26/2015
980001A	980001A	977	Piece (pc)	3/26/2015

55257	55257	912	Piece (pc)	3/28/2015
058818B	058818B	1007	Yard (ya)	3/31/2015
004476B	004476B	128	Yard (ya)	3/31/2015
013061B	013061B	1267	Piece (pc)	3/31/2015
013061C	013061C	862	Piece (pc)	3/31/2015
29909	29909	1249	Piece (pc)	3/31/2015
62542	62542	1249	Piece (pc)	3/31/2015
980001A	980001A	1251	Piece (pc)	3/31/2015
980010B	980010B	1177	Piece (pc)	3/31/2015
7016	7016	1413	Yard (ya)	3/31/2015
200007C	200007C	103	Yard (ya)	3/31/2015
15370	15370	42	Yard (ya)	4/2/2015
013891B	013891B	114	Yard (ya)	4/2/2015
005038C	005038C	995	Piece (pc)	4/2/2015
063946B	063946B	2089	Piece (pc)	4/2/2015
059199C	059199C	965	Piece (pc)	4/2/2015
3299	3299	1040	Piece (pc)	4/2/2015
5274	5274	1040	Piece (pc)	4/2/2015
55257	55257	1055	Piece (pc)	4/2/2015
62674	62674	1040	Piece (pc)	4/2/2015
980001A	980001A	1267	Piece (pc)	4/2/2015
55257	55257	1475	Piece (pc)	4/4/2015
058818B	058818B	838	Yard (ya)	4/7/2015
004476B	004476B	135	Yard (ya)	4/7/2015
013061A	013061A	1475	Piece (pc)	4/7/2015
013061C	013061C	1055	Piece (pc)	4/7/2015
29909	29909	1040	Piece (pc)	4/7/2015
62542	62542	1040	Piece (pc)	4/7/2015
980001A	980001A	1475	Piece (pc)	4/7/2015
980001B	980001B	1012	Piece (pc)	4/7/2015
980010C	980010C	1035	Piece (pc)	4/7/2015
200007A	200007A	234	Yard (ya)	4/7/2015
6492	6492	29	Yard (ya)	4/7/2015
15370	15370	20	Yard (ya)	4/9/2015
005038D	005038D	498	Piece (pc)	4/9/2015
063946C	063946C	996	Piece (pc)	4/9/2015
059199D	059199D	438	Piece (pc)	4/9/2015
3299	3299	496	Piece (pc)	4/9/2015
5274	5274	496	Piece (pc)	4/9/2015
55257	55257	503	Piece (pc)	4/9/2015
62674	62674	496	Piece (pc)	4/9/2015

980001A	980001A	1055	Piece (pc)	4/9/2015
55257	55257	549	Piece (pc)	4/11/2015
058818A	058818A	350	Yard (ya)	4/14/2015
058818B	058818B	400	Yard (ya)	4/14/2015
004476B	004476B	65	Yard (ya)	4/14/2015
013061B	013061B	549	Piece (pc)	4/14/2015
013061D	013061D	383	Piece (pc)	4/14/2015
29909	29909	496	Piece (pc)	4/14/2015
62542	62542	496	Piece (pc)	4/14/2015
980001A	980001A	549	Piece (pc)	4/14/2015
980001B	980001B	497	Piece (pc)	4/14/2015
980010D	980010D	437	Piece (pc)	4/14/2015
200007B	200007B	58	Yard (ya)	4/14/2015
6492	6492	25	Yard (ya)	4/14/2015
15370	15370	67	Yard (ya)	4/16/2015
013891A	013891A	183	Yard (ya)	4/16/2015
005038A	005038A	1675	Piece (pc)	4/16/2015
063946A	063946A	3350	Piece (pc)	4/16/2015
059199A	059199A	1675	Piece (pc)	4/16/2015
3299	3299	1667	Piece (pc)	4/16/2015
5274	5274	1667	Piece (pc)	4/16/2015
55257	55257	1692	Piece (pc)	4/16/2015
62674	62674	1667	Piece (pc)	4/16/2015
980001A	980001A	503	Piece (pc)	4/16/2015
55257	55257	603	Piece (pc)	4/18/2015
058818A	058818A	1078	Yard (ya)	4/21/2015
004476A	004476A	166	Yard (ya)	4/21/2015
013061A	013061A	1692	Piece (pc)	4/21/2015
29909	29909	1667	Piece (pc)	4/21/2015
62542	62542	1667	Piece (pc)	4/21/2015
980001A	980001A	1670	Piece (pc)	4/21/2015
980010A	980010A	1771	Piece (pc)	4/21/2015
200007A	200007A	108	Yard (ya)	4/21/2015
6492	6492	28	Yard (ya)	4/21/2015
15370	15370	63	Yard (ya)	4/23/2015
013891A	013891A	171	Yard (ya)	4/23/2015
005038B	005038B	1569	Piece (pc)	4/23/2015
063946B	063946B	3138	Piece (pc)	4/23/2015
059199B	059199B	1569	Piece (pc)	4/23/2015
3299	3299	1561	Piece (pc)	4/23/2015
5274	5274	1561	Piece (pc)	4/23/2015

55257	55257	1585	Piece (pc)	4/23/2015
62674	62674	1561	Piece (pc)	4/23/2015
980001A	980001A	1692	Piece (pc)	4/23/2015
55257	55257	765	Piece (pc)	4/25/2015
058818A	058818A	1010	Yard (ya)	4/28/2015
004476A	004476A	156	Yard (ya)	4/28/2015
013061B	013061B	1585	Piece (pc)	4/28/2015
29909	29909	1561	Piece (pc)	4/28/2015
62542	62542	1561	Piece (pc)	4/28/2015
980001A	980001A	1564	Piece (pc)	4/28/2015
980010B	980010B	1659	Piece (pc)	4/28/2015
200007B	200007B	137	Yard (ya)	4/28/2015
6492	6492	35	Yard (ya)	4/28/2015
15370	15370	58	Yard (ya)	4/30/2015
013891A	013891A	157	Yard (ya)	4/30/2015
005038C	005038C	1441	Piece (pc)	4/30/2015
063946B	063946B	2881	Piece (pc)	4/30/2015
059199C	059199C	1441	Piece (pc)	4/30/2015
3299	3299	1434	Piece (pc)	4/30/2015
5274	5274	1434	Piece (pc)	4/30/2015
55257	55257	1455	Piece (pc)	4/30/2015
62674	62674	1434	Piece (pc)	4/30/2015
980001A	980001A	1585	Piece (pc)	4/30/2015
55257	55257	425	Piece (pc)	5/2/2015
058818A	058818A	927	Yard (ya)	5/5/2015
004476A	004476A	143	Yard (ya)	5/5/2015
013061C	013061C	1455	Piece (pc)	5/5/2015
013061D	013061D	1455	Piece (pc)	5/5/2015
29909	29909	1434	Piece (pc)	5/5/2015
62542	62542	1434	Piece (pc)	5/5/2015
980001A	980001A	425	Piece (pc)	5/5/2015
980001B	980001B	1436	Piece (pc)	5/5/2015
980010C	980010C	1523	Piece (pc)	5/5/2015
200007C	200007C	76	Yard (ya)	5/5/2015
6492	6492	20	Yard (ya)	5/5/2015
15370	15370	23	Yard (ya)	5/7/2015
013891A	013891A	63	Yard (ya)	5/7/2015
005038D	005038D	576	Piece (pc)	5/7/2015
063946C	063946C	1152	Piece (pc)	5/7/2015
059199D	059199D	576	Piece (pc)	5/7/2015
3299	3299	573	Piece (pc)	5/7/2015

5274	5274	573	Piece (pc)	5/7/2015
55257	55257	582	Piece (pc)	5/7/2015
62674	62674	573	Piece (pc)	5/7/2015
980001A	980001A	1455	Piece (pc)	5/7/2015
55257	55257	808	Piece (pc)	5/9/2015
058818A	058818A	371	Yard (ya)	5/12/2015
004476A	004476A	57	Yard (ya)	5/12/2015
013061A	013061A	808	Piece (pc)	5/12/2015
29909	29909	573	Piece (pc)	5/12/2015
62542	62542	573	Piece (pc)	5/12/2015
980001A	980001A	808	Piece (pc)	5/12/2015
980001B	980001B	574	Piece (pc)	5/12/2015
980010D	980010D	609	Piece (pc)	5/12/2015
7016	7016	1252	Yard (ya)	5/12/2015
200007A	200007A	145	Yard (ya)	5/12/2015
6492	6492	37	Yard (ya)	5/12/2015
15370	15370	71	Yard (ya)	5/14/2015
013891B	013891B	195	Yard (ya)	5/14/2015
005038A	005038A	1785	Piece (pc)	5/14/2015
063946A	063946A	3570	Piece (pc)	5/14/2015
059199A	059199A	1785	Piece (pc)	5/14/2015
3299	3299	1776	Piece (pc)	5/14/2015
5274	5274	1776	Piece (pc)	5/14/2015
55257	55257	1803	Piece (pc)	5/14/2015
62674	62674	1776	Piece (pc)	5/14/2015
980001A	980001A	582	Piece (pc)	5/14/2015
55257	55257	1300	Piece (pc)	5/16/2015
058818B	058818B	1432	Yard (ya)	5/19/2015
004476B	004476B	230	Yard (ya)	5/19/2015
013061A	013061A	1803	Piece (pc)	5/19/2015
013061B	013061B	1300	Piece (pc)	5/19/2015
29909	29909	1776	Piece (pc)	5/19/2015
62542	62542	1776	Piece (pc)	5/19/2015
980001A	980001A	1780	Piece (pc)	5/19/2015
980010A	980010A	1888	Piece (pc)	5/19/2015
7016	7016	2013	Yard (ya)	5/19/2015
200007B	200007B	232	Yard (ya)	5/19/2015
6492	6492	59	Yard (ya)	5/19/2015
15370	15370	51	Yard (ya)	5/21/2015
013891B	013891B	139	Yard (ya)	5/21/2015
005038B	005038B	1271	Piece (pc)	5/21/2015

063946B	063946B	2542	Piece (pc)	5/21/2015
059199B	059199B	1271	Piece (pc)	5/21/2015
3299	3299	1265	Piece (pc)	5/21/2015
5274	5274	1265	Piece (pc)	5/21/2015
55257	55257	1284	Piece (pc)	5/21/2015
62674	62674	1265	Piece (pc)	5/21/2015
980001A	980001A	1803	Piece (pc)	5/21/2015
55257	55257	1031	Piece (pc)	5/23/2015
058818B	058818B	1019	Yard (ya)	5/26/2015
004476B	004476B	164	Yard (ya)	5/26/2015
013061B	013061B	1284	Piece (pc)	5/26/2015
013061C	013061C	1031	Piece (pc)	5/26/2015
29909	29909	1265	Piece (pc)	5/26/2015
62542	62542	1265	Piece (pc)	5/26/2015
980001A	980001A	1267	Piece (pc)	5/26/2015
980010B	980010B	1344	Piece (pc)	5/26/2015
7016	7016	1596	Yard (ya)	5/26/2015
200007C	200007C	184	Yard (ya)	5/26/2015
6492	6492	47	Yard (ya)	5/26/2015
15370	15370	45	Yard (ya)	5/28/2015
013891B	013891B	122	Yard (ya)	5/28/2015
005038C	005038C	1114	Piece (pc)	5/28/2015
063946B	063946B	2227	Piece (pc)	5/28/2015
059199C	059199C	1114	Piece (pc)	5/28/2015
3299	3299	1108	Piece (pc)	5/28/2015
5274	5274	1108	Piece (pc)	5/28/2015
55257	55257	1125	Piece (pc)	5/28/2015
62674	62674	1108	Piece (pc)	5/28/2015
980001A	980001A	1284	Piece (pc)	5/28/2015
55257	55257	1788	Piece (pc)	5/30/2015
058818B	058818B	893	Yard (ya)	6/2/2015
004476B	004476B	144	Yard (ya)	6/2/2015
013061A	013061A	1788	Piece (pc)	6/2/2015
013061C	013061C	1125	Piece (pc)	6/2/2015
29909	29909	1108	Piece (pc)	6/2/2015
62542	62542	1108	Piece (pc)	6/2/2015
980001A	980001A	1788	Piece (pc)	6/2/2015
980001B	980001B	1110	Piece (pc)	6/2/2015
980010C	980010C	1177	Piece (pc)	6/2/2015
200007A	200007A	319	Yard (ya)	6/2/2015
6492	6492	81	Yard (ya)	6/2/2015

15370	15370	57	Yard (ya)	6/4/2015
005038D	005038D	1422	Piece (pc)	6/4/2015
063946C	063946C	2843	Piece (pc)	6/4/2015
059199D	059199D	1422	Piece (pc)	6/4/2015
3299	3299	1415	Piece (pc)	6/4/2015
5274	5274	1415	Piece (pc)	6/4/2015
55257	55257	1436	Piece (pc)	6/4/2015
62674	62674	1415	Piece (pc)	6/4/2015
980001A	980001A	1125	Piece (pc)	6/4/2015
55257	55257	1672	Piece (pc)	6/6/2015
058818A	058818A	1066	Yard (ya)	6/9/2015
058818B	058818B	1140	Yard (ya)	6/9/2015
004476B	004476B	183	Yard (ya)	6/9/2015
013061B	013061B	1672	Piece (pc)	6/9/2015
013061D	013061D	1436	Piece (pc)	6/9/2015
29909	29909	1415	Piece (pc)	6/9/2015
62542	62542	1415	Piece (pc)	6/9/2015
980001A	980001A	1672	Piece (pc)	6/9/2015
980001B	980001B	1417	Piece (pc)	6/9/2015
980010D	980010D	1503	Piece (pc)	6/9/2015
200007B	200007B	299	Yard (ya)	6/9/2015
6492	6492	76	Yard (ya)	6/9/2015
15370	15370	71	Yard (ya)	6/11/2015
013891A	013891A	196	Yard (ya)	6/11/2015
005038A	005038A	1791	Piece (pc)	6/11/2015
063946A	063946A	3582	Piece (pc)	6/11/2015
059199A	059199A	1791	Piece (pc)	6/11/2015
3299	3299	1782	Piece (pc)	6/11/2015
5274	5274	1782	Piece (pc)	6/11/2015
55257	55257	1809	Piece (pc)	6/11/2015
62674	62674	1782	Piece (pc)	6/11/2015
980001A	980001A	1436	Piece (pc)	6/11/2015
55257	55257	1529	Piece (pc)	6/13/2015
058818A	058818A	1153	Yard (ya)	6/16/2015
004476A	004476A	178	Yard (ya)	6/16/2015
013061A	013061A	1809	Piece (pc)	6/16/2015
29909	29909	1782	Piece (pc)	6/16/2015
62542	62542	1782	Piece (pc)	6/16/2015
980001A	980001A	1786	Piece (pc)	6/16/2015
980010A	980010A	1894	Piece (pc)	6/16/2015
200007A	200007A	273	Yard (ya)	6/16/2015

6492	6492	69	Yard (ya)	6/16/2015
15370	15370	57	Yard (ya)	6/18/2015
013891A	013891A	156	Yard (ya)	6/18/2015
005038B	005038B	1429	Piece (pc)	6/18/2015
063946B	063946B	2857	Piece (pc)	6/18/2015
059199B	059199B	1429	Piece (pc)	6/18/2015
3299	3299	1422	Piece (pc)	6/18/2015
5274	5274	1422	Piece (pc)	6/18/2015
55257	55257	1443	Piece (pc)	6/18/2015
62674	62674	1422	Piece (pc)	6/18/2015
980001A	980001A	1133	Piece (pc)	6/18/2015
55257	55257	930	Piece (pc)	6/20/2015
058818A	058818A	920	Yard (ya)	6/23/2015
004476A	004476A	142	Yard (ya)	6/23/2015
013061B	013061B	1443	Piece (pc)	6/23/2015
29909	29909	1422	Piece (pc)	6/23/2015
62542	62542	1422	Piece (pc)	6/23/2015
980001A	980001A	1424	Piece (pc)	6/23/2015
980010B	980010B	1511	Piece (pc)	6/23/2015
200007B	200007B	166	Yard (ya)	6/23/2015
6492	6492	42	Yard (ya)	6/23/2015
15370	15370	60	Yard (ya)	6/25/2015
013891A	013891A	165	Yard (ya)	6/25/2015
005038C	005038C	1507	Piece (pc)	6/25/2015
063946B	063946B	3014	Piece (pc)	6/25/2015
059199C	059199C	1507	Piece (pc)	6/25/2015
3299	3299	1500	Piece (pc)	6/25/2015
5274	5274	1500	Piece (pc)	6/25/2015
55257	55257	1522	Piece (pc)	6/25/2015
62674	62674	1500	Piece (pc)	6/25/2015
980001A	980001A	1161	Piece (pc)	6/25/2015
55257	55257	643	Piece (pc)	6/27/2015
058818A	058818A	970	Yard (ya)	6/30/2015
004476A	004476A	150	Yard (ya)	6/30/2015
013061C	013061C	1522	Piece (pc)	6/30/2015
013061D	013061D	1522	Piece (pc)	6/30/2015
29909	29909	1500	Piece (pc)	6/30/2015
62542	62542	1500	Piece (pc)	6/30/2015
980001A	980001A	643	Piece (pc)	6/30/2015
980001B	980001B	1503	Piece (pc)	6/30/2015
980010C	980010C	1594	Piece (pc)	6/30/2015

200007C	200007C	115	Yard (ya)	6/30/2015
6492	6492	29	Yard (ya)	6/30/2015
15370	15370	36	Yard (ya)	7/2/2015
013891A	013891A	97	Yard (ya)	7/2/2015
005038D	005038D	886	Piece (pc)	7/2/2015
063946C	063946C	1772	Piece (pc)	7/2/2015
059199D	059199D	886	Piece (pc)	7/2/2015
3299	3299	882	Piece (pc)	7/2/2015
5274	5274	882	Piece (pc)	7/2/2015
55257	55257	895	Piece (pc)	7/2/2015
62674	62674	882	Piece (pc)	7/2/2015
980001A	980001A	1204	Piece (pc)	7/2/2015
55257	55257	468	Piece (pc)	7/4/2015
058818A	058818A	571	Yard (ya)	7/7/2015
004476A	004476A	88	Yard (ya)	7/7/2015
013061A	013061A	468	Piece (pc)	7/7/2015
29909	29909	882	Piece (pc)	7/7/2015
62542	62542	882	Piece (pc)	7/7/2015
980001A	980001A	468	Piece (pc)	7/7/2015
980001B	980001B	884	Piece (pc)	7/7/2015
980010D	980010D	937	Piece (pc)	7/7/2015
7016	7016	724	Yard (ya)	7/7/2015
200007A	200007A	84	Yard (ya)	7/7/2015
6492	6492	22	Yard (ya)	7/7/2015
15370	15370	72	Yard (ya)	7/9/2015
013891B	013891B	198	Yard (ya)	7/9/2015
005038A	005038A	1818	Piece (pc)	7/9/2015
063946A	063946A	3636	Piece (pc)	7/9/2015
059199A	059199A	1818	Piece (pc)	7/9/2015
3299	3299	1809	Piece (pc)	7/9/2015
5274	5274	1809	Piece (pc)	7/9/2015
55257	55257	1836	Piece (pc)	7/9/2015
62674	62674	1809	Piece (pc)	7/9/2015
980001A	980001A	1188	Piece (pc)	7/9/2015
55257	55257	567	Piece (pc)	7/11/2015
058818B	058818B	1458	Yard (ya)	7/14/2015
004476B	004476B	234	Yard (ya)	7/14/2015
013061A	013061A	1836	Piece (pc)	7/14/2015
013061B	013061B	567	Piece (pc)	7/14/2015
29909	29909	1809	Piece (pc)	7/14/2015
62542	62542	1809	Piece (pc)	7/14/2015

980001A	980001A	1813	Piece (pc)	7/14/2015
980010A	980010A	1923	Piece (pc)	7/14/2015
7016	7016	877	Yard (ya)	7/14/2015
200007B	200007B	102	Yard (ya)	7/14/2015
6492	6492	26	Yard (ya)	7/14/2015
15370	15370	57	Yard (ya)	7/16/2015
013891B	013891B	156	Yard (ya)	7/16/2015
005038B	005038B	1431	Piece (pc)	7/16/2015
063946B	063946B	2861	Piece (pc)	7/16/2015
059199B	059199B	1431	Piece (pc)	7/16/2015
3299	3299	1424	Piece (pc)	7/16/2015
5274	5274	1424	Piece (pc)	7/16/2015
55257	55257	1445	Piece (pc)	7/16/2015
62674	62674	1424	Piece (pc)	7/16/2015
980001A	980001A	573	Piece (pc)	7/16/2015
55257	55257	1811	Piece (pc)	7/18/2015
013061C	013061C	1270	Piece (pc)	7/19/2015
62542	62542	1270	Piece (pc)	7/19/2015
980001A	980001A	1270	Piece (pc)	7/19/2015
7016	7016	1968	Yard (ya)	7/19/2015
200007C	200007C	227	Yard (ya)	7/19/2015
6492	6492	58	Yard (ya)	7/19/2015
058818B	058818B	1147	Yard (ya)	7/21/2015
004476B	004476B	185	Yard (ya)	7/21/2015
013061A	013061A	1811	Piece (pc)	7/21/2015
013061B	013061B	1445	Piece (pc)	7/21/2015
29909	29909	1424	Piece (pc)	7/21/2015
62542	62542	1424	Piece (pc)	7/21/2015
980001A	980001A	1426	Piece (pc)	7/21/2015
980010B	980010B	1513	Piece (pc)	7/21/2015
200007A	200007A	324	Yard (ya)	7/21/2015
6492	6492	82	Yard (ya)	7/21/2015
980001A	980001A	1405	Piece (pc)	7/23/2015

Example Lot-Sizing in MRP

Table 8: Weekly net requirement schedule example 1:

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	Calculation depends on Lot-Sizing Rule											
Net requirement			10	10	15	20	70	180	250	270	230	40	0	10

Ordering Cost (Co) = $ 300 per order

Inventory holding cost (Ch) =$ 2 per unit

Unit cost = $50 per unit

Lead time = 1 week

Total net requirement = 1105 (from week 3 to week 14)

Average weekly demands (d) = $\frac{1105}{12} = 92.1$

1/Multiple Order Quantity

Table 9 Multiple Order Quantity example with lot size = 20 units

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	10	0	5	5	15	15	5	15	5	5	5	15
Net requirement			10		15	15	65	165	235	265	215	35		5
Planned order receipts			20		20	20	80	180	240	280	220	40		20
Planned order releases		20		20	20	80	180	240	280	220	40		20	

Total ordering cost = 300*10 = $ 3000

Total inventory holding cost = (10+5+5+15+15+5+15+5+5+5+15)*2 = $200

Purchasing Cost = (20+20+20+80+180+240+280+220+40+20)*50 = $56000

Total cost = $56000+$3000+$200= $59200

2/ Minimum Order Quantity

Table 10: Minimum Order Quantity example with minimum order is 50 units

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	40	30	15	45	25	0	0	0		10	10	0
Net requirement			10			5	25	155	250	270	230	40		
Planned order receipts			50			50	50	155	250	270	230	50		
Planned order releases		50			50	50	155	250	270	230	50			

Total ordering cost = 300*8 = $ 2400

Total inventory holding cost = (40+30+15+45+25+10+10)*2 = $350

Purchasing Cost = (50+50+50+155+250+270+230+50)*50 = $55250

Total cost = $55250+$2400+$350= $58000

3/ Lot for Lot

Table 11 Lot for Lot method solved for example 1

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	0	0	0	0	0	0	0	0	0	0	0	0
Net requirement			10	10	15	20	70	180	250	270	230	40	0	10
Planned order receipts			10	10	15	20	70	180	250	270	230	40	0	10
Planned order releases		10	10	15	20	70	180	250	270	230	40	0	10	

Total ordering cost = 300*11 = $ 3300

Total inventory holding cost = 0

Purchasing Cost = (10+10+15+20+70+180+250+270+230+40+10)*50 = $55250

Total cost = $55250+$2400+0 = $58550

4/ Economic Order Quantity

In requirement planning, EOQ total cost equation cannot be used and lot size formula is as below:

$$EOQ = \sqrt{\frac{2dC_o}{C_h}} = \sqrt{\frac{2 * 92.1 * 300}{2}} = 166\ units$$

Problem example 3.1 is solved by EOQ method representing in the table below:

Table 12: EOQ Method used for example 1

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	156	146	131	111	41	27	0	0	0	126	126	116
Net requirement			10	10	15	20	70	180	250	270	230	40	0	10
Planned order receipts			166					166	223	270	230	166		
Planned order releases		166					166	223	270	230	166			

Total ordering cost = 300*6 = $ 1800

Total inventory holding cost = (156+146+131+111+41+27+126+126+116)*2= $1960

Purchasing Cost = (166+166+223+270+230+166)*50= $ 61050

Total cost = $61050+$1800+$1960= $ 64810

5/ Periodic Order Quantity

The time between orders of the example 5.1 is $\frac{EOQ}{d} = \frac{166}{92.1} = 1.8 \approx 2\ weeks$

Table 13: Periodic Order Quantity example

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	10	20	10	15	20	70	180	250	270	230	40	0	10
Schedule reciept	60													
Project on hand (25)	20	10	10	0	20	0	180	0	270	0	40	0	0	0
Net requirement			10	10	15	20	70	180	250	270	230	40	0	10
Planned order receipts			20		35		250		520		270			10
Planned order releases		20		35		250		520		270			10	

Total ordering cost = 300*6 = $ 1800

Total inventory holding cost = (10+20+180+270+40)*2= $ 1040

Purchasing Cost = (20+35+250+520+270+10)*50= $ 55250

Total cost = $55250+$1800+$1040= $ 58090

6/ Discount Model – Least Unit Cost

Least Unit Cost Rule is particularly suitable for calculating the purchasing lot-size in case of discount quantity exists.

a/ Discount Quantity Problem

This problem happens when the suppliers propose the discount prices in a specific order quantity. In this case, Least Unit Cost is used to find the optimum quantity for the purchase order which satisfies lowest unit cost.

Table 14: Example purchase discount problem

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross Requirement	65	50	90	100	124	100	50	50	100	125	125	100	50	100
Schedule Receipts	70													
Project On Hand (55)	60	10	Calculation depends on lot-sizing rule.											
Net Requirements			80	100	125	100	50	50	100	125	125	100	50	100

Order cost = $ 100

Inventory carrying cost = $ 2/period/unit

Base price = $ 500/unit

Discount price = $ 450/unit

Discount quantity = 350 units

All unit discount schedule

Table 15: Discount Quantity

Quantity (unit)	Price (USD)
0->349	500
350->+∞	450

a/ Least Unit Cost

Table 16: Least Unit Cost calculation example

Trial periods combined	Trial Lot size (Cumulative Net Requirement)	Ordering Cost	Holding Cost	Purchasing Cost	Cumulative cost	Cost per unit
3	80	100	0	80*500=40000	100+0+40000=40100	501.25
3,4	180	100	200	180*500=90000	100+200+90000=90300	501.67
3,4,5	304	100	696	304*500=152000	100+696+152000=152796	502.62
3,4,5,6	**404**	100	1296	404*450=181800	100+1296+181800=183196	453.45
3,4,5,5*	**350**	100	972	350*450=157500	100+972+157500=158572	**453.06**
5**	54	100	0	54*500=27000	100+0+27000=27100	501.85
5**,7	104	100	100	104*500=52000	100+100+52000=52200	501.92
5**,7,8	154	100	300	154*500=77000	100+300+77000=77400	502.60
5**,7,8,9	254	100	900	254*500=127000	100+900+127000=128000	503.94
5**,7,8,9,10	379	100	1900	379*450=170550	100+1900+170550=172550	455.28
5,7,8,9,9***	**350**	100	1610	350*450=157500	100+1610+157500=159210	**454.89**
9**	29	100	0	29*500=14500	100+0+14500=14600	503.45
9**,11	154	100	250	154*500=77000	100+250+77000=77350	502.27
9**,11,12	254	100	650	254*500=127000	100+650+127000=127750	502.95
9**,11,12,13	304	100	950	304*500=152000	100+950+152000=153050	503.45
9**,11,12,13,14	404	100	1750	404*450=181800	100+1750+181800=183650	454.58
9,11,12,13,13***	**350**	100	1318	350*450=157500	100+1318+157500=158918	**454.05**
13**	**54**	100	0	54*500=27000	100+0+27000=27100	**501.85**

Note that 5*, 5** is a part of period 6 and total quantity requirement of period 5* and 5** is the quantity requirement of period 6, similar to 9* and 9**, 13* and 13**. After determining the lot size requirement, the process is repeated in the remaining periods. The table below expresses result of calculation:

Table 17: Discount Quantity model using Least Unit Cost

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Gross requirement	65	50	90	100	124	100	50	50	100	125	125	100	50	100
Schedule reciept	70													
Project on hand (55)	60	10	270	170	46	296	246	196	96	321	196	96	46	0
Net requirement			80			54				29				54
Planned order receipts			350			350				350				54
Planned order releases		350			350				350				54	

Total Cost = 158572 +159210 +158918 +27100 = $ 503800

Source Code

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataAccessLayer;
using QLSX.DataTransferObject;
using System.Data;
namespace QLSX.BusinessLogicLayer
{
    class BLMasterProduct
    {
        public static DAMasterProduct Select_ALL()
        {
            return new DAMasterProduct();
        }

        public static DTOMasterProduct Select_Current_Row(object CurrentRow)
        {
            DataRow dr = ((DataRowView)CurrentRow).Row;
            DTOMasterProduct dt = new DTOMasterProduct();
            dt.Id_Name = dr["Id_Name"].ToString();
            dt.Day = DateTime.Parse(dr["Day"].ToString());
            dt.Quantity = int.Parse(dr["Quantity"].ToString());
            dt.Id = int.Parse(dr["Id"].ToString());
            return dt;
        }
        public void Them(DTOMasterProduct dt, object source)
        {
            DAMasterProduct DAms = (DAMasterProduct)source;
            DAms.Them(dt);
        }
        public void Xoa(DTOMasterProduct dt, object source)
        {
            DAMasterProduct DAms = (DAMasterProduct)source;
            DAms.Xoa(dt);
        }
        public void Sua(DTOMasterProduct dt, object source)
        {
            DAMasterProduct DAms = (DAMasterProduct)source;
            DAms.Sua(dt);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataAccessLayer;
using QLSX.DataTransferObject;

namespace QLSX.BusinessLogicLayer
{
    class BLProductStructure
    {
```

```
        public static DAProductStructure Select_All()
        {
            return new DAProductStructure();
        }
        public static DAProductStructure Select_by_Id(int id)
        {
            return new DAProductStructure(id);
        }
        public static DAProductStructure Select_Schedule_Receipt()
        {
            return new
DAProductStructure(DAProductStructure.Select_Schedule_Receipt());
        }
        public static DAProductStructure Select_Master_Production_Schedule()
        {
            return new
DAProductStructure(DAProductStructure.Select_Master_Production_Schedule());
        }
        public static DAProductStructure Select_By_Id_Name(string Id_Name)
        {
            return new
DAProductStructure(DAProductStructure.Select_By_Id_Name(Id_Name));
        }
        public void Them(DTOProductStruct tg, object source)
        {
            DAProductStructure DAps = (DAProductStructure)source;
            DAps.Them(tg);
        }
        public void Sua(DTOProductStruct tg, object source)
        {
            DAProductStructure DAps = (DAProductStructure)source;
            DAps.Sua(tg);

        }
        public void Xoa(DTOProductStruct tg, object source)
        {
            DAProductStructure DAps = (DAProductStructure)source;
            DAps.Xoa(tg);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataAccessLayer;
using QLSX.DataTransferObject;
using System.Data;

namespace QLSX.BusinessLogicLayer
{
    class BLScheduleReceipt
    {
        public static DAScheduleReceipt Select_ALL()
        {
            return new DAScheduleReceipt();
        }
```

```
        public static DTOScheduleReceipt Select_Current_Row(object CurrentRow)
        {
            DataRow dr = ((DataRowView)CurrentRow).Row;
            DTOScheduleReceipt dt = new DTOScheduleReceipt();
            dt.Id_Name = dr["Id_Name"].ToString();
            dt.Day = DateTime.Parse(dr["Day"].ToString());
            dt.Schedule_Receipt = int.Parse(dr["Schedule_Receipt"].ToString());
            dt.Id = int.Parse(dr["Id"].ToString());
            return dt;
        }
        public void Them(DTOScheduleReceipt dt, object source)
        {
            DAScheduleReceipt DAms = (DAScheduleReceipt)source;
            DAms.Them(dt);
        }
        public void Xoa(DTOScheduleReceipt dt, object source)
        {
            DAScheduleReceipt DAms = (DAScheduleReceipt)source;
            DAms.Xoa(dt);
        }
        public void Sua(DTOScheduleReceipt dt, object source)
        {
            DAScheduleReceipt DAms = (DAScheduleReceipt)source;
            DAms.Sua(dt);
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;
using System.Data.SqlClient;
using MySql.Data.MySqlClient;
using System.Windows.Forms;

namespace QLSX.DataAccessLayer
{
    class AppManagement : DataTable
    {
        private MySqlConnection _ketnoi;
        private MySqlDataAdapter _bodocghi;
        private String _duongdan;
        private String _tenbang;
        private String _chuoisql;
        public String Chuoisql
        {
            get { return _chuoisql; }
            set { _chuoisql = value; }
        }
        public string DuongDan
        {
            get { return _duongdan; }
            set { _duongdan = value; }
        }
        public string Tenbang
```

```
        {
            get { return _tenbang; }
            set { _tenbang = value; }
        }
        public MySqlDataAdapter Bodocghi
        {
            get { return _bodocghi; }
            set { _bodocghi = value; }
        }
        public MySqlConnection Ketnoi
        {
            get { return _ketnoi; }
            set { _ketnoi = value; }
        }
        public int So_dong
        {
            get { return this.DefaultView.Count; }
        }
        public AppManagement()
            : base()
        {
        }
        public AppManagement(string pTenBang)
            : base(pTenBang)
        {
            Tenbang = pTenBang;
            DocBang();
        }
        public AppManagement(string pTenBang, string pChuoiSQL)
            : base(pTenBang)
        {
            Tenbang = pTenBang;
            Chuoisql = pChuoiSQL;
            DocBang();
        }
        private void TaoKetNoi()
        {
            if (Ketnoi == null)
                Ketnoi = new MySqlConnection(cThuVien.connectString);
        }
        public string GetLastID(string nameTable, string nameSelectColumn)
        {
            TaoKetNoi();
            string tmp = "SELECT " + nameSelectColumn + " FROM " + nameTable + "
ORDER BY " + nameSelectColumn + " DESC LIMIT 1";


            MySqlCommand sql = new MySqlCommand(tmp, Ketnoi);

            Ketnoi.Open();
            string str = (sql.ExecuteScalar()).ToString();
            Ketnoi.Close();
            return str;

        }

        private void DocBang()
        {
```

```
            if (Chuoisql == null) Chuoisql = "select * from " + Tenbang;
            TaoKetNoi();
            Bodocghi = new MySqlDataAdapter(Chuoisql, Ketnoi);
            Bodocghi.FillSchema(this, SchemaType.Mapped);
            Bodocghi.Fill(this);
            Bodocghi.SelectCommand.CommandText = "select * from " + Tenbang;
            MySqlCommandBuilder TaoBoLenh = new MySqlCommandBuilder(Bodocghi);

            Bodocghi.RowUpdated += new
MySqlRowUpdatedEventHandler(Bodocghi_RowUpdated);
        }

        void Bodocghi_RowUpdated(object sender, MySqlRowUpdatedEventArgs e)
        {
            if (this.PrimaryKey[0].AutoIncrement)
            {
                if (e.Status == UpdateStatus.Continue && e.StatementType ==
StatementType.Insert)
                {
                    this.PrimaryKey[0].ReadOnly = false;
                    MySqlCommand LaySoTuDong = new MySqlCommand("select
@@identity", Ketnoi);
                    e.Row[0] = LaySoTuDong.ExecuteScalar();
                    e.Row.AcceptChanges();
                }
            }
        }
        public void Ghi()
        {
            Bodocghi.Update(this);
        }
        public void Khong_ghi()
        {
            this.RejectChanges();
        }

    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataTransferObject;
using System.Data;

namespace QLSX.DataAccessLayer
{
    class DAMasterProduct:AppManagement
    {
        public DAMasterProduct()
            : base("master_product")
        {
        }

        public DAMasterProduct(String sql)
            : base("master_product", sql)
        {
```

```
        }


        public void Them(DTOMasterProduct dt)
        {
            DataRow dr = this.NewRow();
            dr["Id_Name"] = dt.Id_Name;
            dr["Day"] = dt.Day;
            dr["Quantity"] = dt.Quantity;
            this.Rows.Add(dr);
            this.Ghi();
        }

        public void Sua(DTOMasterProduct dt)
        {
            DataRow dr = this.Rows.Find(dt.Id);
            dr["Id_Name"] = dt.Id_Name;
            dr["Day"] = dt.Day;
            dr["Quantity"] = dt.Quantity;
            this.Ghi();
        }
        public void Xoa(DTOMasterProduct dt)
        {
            DataRow dr = this.Rows.Find(dt.Id);
            dr.Delete();
            this.Ghi();
        }
    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Data;
using System.Data.SqlClient;
using System.Threading.Tasks;
using QLSX.DataTransferObject;

namespace QLSX.DataAccessLayer
{
    class DAProductStructure:AppManagement
    {
        public DAProductStructure()
            : base("product_structure")
        {
        }

        public DAProductStructure(String sql)
            : base("product_structure", sql)
        {
        }

        public DAProductStructure(int Id)
            : base("product_structure", "Select * from product_structure where Id
= " + Id)
        {
        }
```

```
        public static string Select_By_Id_Name(string Id_Name)
        {
            string str = string.Format("Select * from product_structure where
Id_Name = '{0}'", Id_Name);
            return str;
        }

        public static string Select_Schedule_Receipt()
        {
            string str = "select distinct id_name,item_name,unit from
product_structure where type = 2";
            return str;
        }

        public static string Select_Master_Production_Schedule()
        {
            string str = "select * from product_structure where level = 1";
            return str;
        }

        public void Them(DTOProductStruct dt)
        {
            DataRow dr = this.NewRow();
            dr["Id"] = dt.Id;
            dr["Annualdemand"] = dt.Annualdemand;
            dr["Buyer"] = dt.Buyer;
            dr["Cycletime"] = dt.Cycletime;
            dr["Discount_Quantity"] = dt.Discount_Quantity;
            dr["Holdingcost"] = dt.Holdingcost;
            dr["Id_Name"] = dt.Id_Name;
            dr["Item_Name"] = dt.Item_Name;
            dr["Leadtime"] = dt.Leadtime;
            dr["Lost_Size"] = dt.Lost_Size;
            dr["Lost_Size_Rule"] = dt.Lost_Size_Rule;
            dr["Low_Level_Code"] = dt.Low_Level_Code;
            dr["Orderingcost"] = dt.Orderingcost;
            dr["Product_Class"] = dt.Product_Class;
            dr["Project_On_Hand"] = dt.Project_On_Hand;
            dr["Reorderpoint"] = dt.Reorderpoint;
            dr["Safetystock"] = dt.Safetystock;
            dr["Type"] = dt.Type;
            dr["Unit"] = dt.Unit;
            dr["Unitcost"] = dt.Unitcost;
            dr["Value"] = dt.Value;
            dr["Value_Class"] = dt.Value_Class;
            dr["Vendor"] = dt.Vendor;
            dr["Warehouse"] = dt.Warehouse;
            dr["Waste"] = dt.Waste;
            dr["Level"] = dt.Level;
            dr["Parent_Id"] = dt.Parent_Id;
            this.Rows.Add(dr);
            this.Ghi();

        }
        public void Sua(DTOProductStruct dt)
        {
            DataRow dr = this.Rows.Find(dt.Id); ;
            dr["Annualdemand"] = dt.Annualdemand;
```

```
            dr["Buyer"] = dt.Buyer;
            dr["Cycletime"] = dt.Cycletime;
            dr["Discount_Quantity"] = dt.Discount_Quantity;
            dr["Holdingcost"] = dt.Holdingcost;
            dr["Id_Name"] = dt.Id_Name;
            dr["Item_Name"] = dt.Item_Name;
            dr["Leadtime"] = dt.Leadtime;
            dr["Lost_Size"] = dt.Lost_Size;
            dr["Lost_Size_Rule"] = dt.Lost_Size_Rule;
            dr["Low_Level_Code"] = dt.Low_Level_Code;
            dr["Orderingcost"] = dt.Orderingcost;
            dr["Product_Class"] = dt.Product_Class;
            dr["Project_On_Hand"] = dt.Project_On_Hand;
            dr["Reorderpoint"] = dt.Reorderpoint;
            dr["Safetystock"] = dt.Safetystock;
            dr["Type"] = dt.Type;
            dr["Unit"] = dt.Unit;
            dr["Unitcost"] = dt.Unitcost;
            dr["Value"] = dt.Value;
            dr["Value_Class"] = dt.Value_Class;
            dr["Vendor"] = dt.Vendor;
            dr["Warehouse"] = dt.Warehouse;
            dr["Waste"] = dt.Waste;
            dr["Level"] = dt.Level;
            dr["Parent_Id"] = dt.Parent_Id;
            this.Ghi();

        }
        public void Xoa(DTOProductStruct dt)
        {
            DataRow dr = this.Rows.Find(dt.Id);
            dr.Delete();
            this.Ghi();
        }
        public static string XuLy()
        {
            AppManagement app = new AppManagement();
            return app.GetLastID("product_structure", "Id");
        }

    }
}
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataTransferObject;
using System.Data;

namespace QLSX.DataAccessLayer
{
    class DAScheduleReceipt : AppManagement
    {
        public DAScheduleReceipt()
            : base("schedule_receipts")
        {
        }
```

```
        public DAScheduleReceipt(String sql)
            : base("schedule_receipts", sql)
        {
        }


        public void Them(DTOScheduleReceipt dt)
        {
            DataRow dr = this.NewRow();
            dr["Id_Name"] = dt.Id_Name;
            dr["Day"] = dt.Day;
            dr["Schedule_Receipt"] = dt.Schedule_Receipt;
            this.Rows.Add(dr);
            this.Ghi();
        }

        public void Sua(DTOScheduleReceipt dt)
        {
            DataRow dr = this.Rows.Find(dt.Id);
            dr["Id_Name"] = dt.Id_Name;
            dr["Day"] = dt.Day;
            dr["Schedule_Receipt"] = dt.Schedule_Receipt;
            this.Ghi();
        }
        public void Xoa(DTOScheduleReceipt dt)
        {
            DataRow dr = this.Rows.Find(dt.Id);
            dr.Delete();
            this.Ghi();
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using QLSX.DataTransferObject;

namespace QLSX.DataTransferObject
{
    class doReport
    {
        protected string methodName = "";
        protected IDictionary<string, dataItem> requiredItems = new
Dictionary<string, dataItem>();

        protected double demand = 0;
        protected int eoq = 0;

        protected int multiple(int a, int b)
        {
            return ((int)Math.Ceiling((double)a / (double)b)) * b;
        }

        protected delegate int mathCalculate(int a, int b);
```

```
        protected void calculateEOQ(dataItem rItem, dataItem oItem)
        {
            //calculate for demand
            int safetyStock = oItem.safetyStock;
            int leadTime = oItem.leadTime;
            int projectOnHand = oItem.projectOnHand;
            int netRequirement = 0;

            double holdingCost = oItem.holdingCost;
            double orderingCost = oItem.orderringCost;

            int startDay = int.MaxValue;
            int endDay = 0;

            for (int iSchedule = 0; iSchedule < rItem.scheduleItem.Count;
++iSchedule)
            {
                int grossRequirement =
oItem.scheduleItem[iSchedule].getGrossRequirement();
                int scheduleReceipt =
oItem.scheduleItem[iSchedule].getScheduleReceipt();

                if (netRequirement <= 0)
                {
                    netRequirement = grossRequirement - scheduleReceipt -
(projectOnHand - safetyStock);
                }
                else
                {
                    netRequirement += grossRequirement;
                    startDay = Math.Min(startDay, iSchedule);
                    endDay = Math.Max(endDay, iSchedule);
                }
            }

            if (endDay >= startDay)
            {
                this.demand /= (double)(endDay - startDay + 1);
                this.eoq = (int)Math.Ceiling(Math.Sqrt((2.0 * this.demand *
holdingCost) / orderingCost));
            }
        }

        protected void generalMethod(dataItem rItem, dataItem oItem, int minOrder,
mathCalculate mathMethod)
        {
            double holdingCostTotal = 0;
            double orderringCostTotal = 0;

            double holdingCost = oItem.holdingCost;
            double orderingCost = oItem.orderringCost;
            double unitCost = oItem.unitCost;

            int safetyStock = oItem.safetyStock;
            int leadTime = oItem.leadTime;
            int projectOnHand = oItem.projectOnHand;
```

```
            for (int iSchedule = 0; iSchedule < rItem.scheduleItem.Count;
++iSchedule)
            {
                int grossRequirement =
oItem.scheduleItem[iSchedule].getGrossRequirement();
                int scheduleReceipt =
oItem.scheduleItem[iSchedule].getScheduleReceipt();
                int netRequirement = grossRequirement - scheduleReceipt -
(projectOnHand - safetyStock);
                int plannedOrderReceipt = 0;


rItem.scheduleItem[iSchedule].setGrossRequirement(grossRequirement);

                if (netRequirement > 0)
                {
                    plannedOrderReceipt = mathMethod(netRequirement, minOrder);
                    orderringCostTotal += orderingCost + (unitCost *
(double)plannedOrderReceipt);


rItem.scheduleItem[iSchedule].setNetRequirement(netRequirement);

rItem.scheduleItem[iSchedule].setPlannedOrderReceipt(plannedOrderReceipt);
                    rItem.sumQuantity += plannedOrderReceipt;

                    if (iSchedule >= leadTime)
                    {
                        rItem.scheduleItem[iSchedule -
leadTime].setPlannedOrderRelease(plannedOrderReceipt);
                    }
                }

                projectOnHand = scheduleReceipt + plannedOrderReceipt +
projectOnHand - grossRequirement;
                holdingCostTotal += (double)projectOnHand * holdingCost;

                rItem.scheduleItem[iSchedule].setProjectOnHand(projectOnHand);
            }

            rItem.cost = holdingCostTotal + orderringCostTotal;
        }

        //public method
        public doReport() { }
        public string getMethodName() { return this.methodName; }

        public IDictionary<string, dataItem> getItemSchedule() { return
this.requiredItems; }
        public dataItem getItemSchedule(string idName) { return
this.requiredItems[idName]; }
    }

    class L4L : doReport
    {
        public L4L(IDictionary<string, dataItem> otherItems)
        {
```

```
        this.methodName = "Lot_For_Lot";
        this.requiredItems = new Dictionary<string, dataItem>(otherItems);

        for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
        {
            var rItem = requiredItems.ElementAt(iItem).Value;
            var oItem = otherItems.ElementAt(iItem).Value;
            int minOrder = 0;

            this.generalMethod(rItem, oItem, minOrder, Math.Max);
        }
    }
}

class MULT : doReport
{
    public MULT(IDictionary<string, dataItem> otherItems)
    {

        this.methodName = "Multiple_Order_Quantity";
        this.requiredItems = new Dictionary<string, dataItem>(otherItems);

        for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
        {
            var rItem = requiredItems.ElementAt(iItem).Value;
            var oItem = otherItems.ElementAt(iItem).Value;
            int minOrder = oItem.lostSize;

            this.generalMethod(rItem, oItem, minOrder, multiple);
        }
    }
}

class MIN : doReport
{
    public MIN(IDictionary<string, dataItem> otherItems)
    {

        this.methodName = "Minimum_Order_Quantity";
        this.requiredItems = new Dictionary<string, dataItem>(otherItems);

        for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
        {
            var rItem = requiredItems.ElementAt(iItem).Value;
            var oItem = otherItems.ElementAt(iItem).Value;
            int minOrder = oItem.lostSize;

            this.generalMethod(rItem, oItem, minOrder, Math.Max);
        }
    }
}

class EOQ : doReport
{
    public EOQ(IDictionary<string, dataItem> otherItems)
    {

        this.methodName = "Ecomomic_Order_Quantity";
```

```
            this.requiredItems = new Dictionary<string, dataItem>(otherItems);

            for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
            {
                var rItem = requiredItems.ElementAt(iItem).Value;
                var oItem = otherItems.ElementAt(iItem).Value;

                this.calculateEOQ(rItem, oItem);
                int minOrder = this.eoq;

                this.generalMethod(rItem, oItem, minOrder, Math.Max);
            }
        }
    }

    class POQ : doReport
    {
        private int poq = 0;

        public POQ(IDictionary<string, dataItem> otherItems)
        {

            this.methodName = "Periodic_Order_Quantity";
            this.requiredItems = new Dictionary<string, dataItem>(otherItems);

            for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
            {
                var rItem = requiredItems.ElementAt(iItem).Value;
                var oItem = otherItems.ElementAt(iItem).Value;

                double holdingCostTotal = 0;
                double orderringCostTotal = 0;

                double holdingCost = oItem.holdingCost;
                double orderingCost = oItem.orderringCost;
                double unitCost = oItem.unitCost;

                int safetyStock = oItem.safetyStock;
                int leadTime = oItem.leadTime;
                int projectOnHand = oItem.projectOnHand;
                //int lostSize = oItem.Value.lostSize;

                this.calculateEOQ(rItem, oItem);
                poq = (this.demand > 0) ? (int)Math.Ceiling((double)this.eoq /
this.demand) : 1;

                for (int iSchedule = 0; iSchedule < rItem.scheduleItem.Count;
++iSchedule)
                {
                    int grossRequirement =
oItem.scheduleItem[iSchedule].getGrossRequirement();
                    int scheduleReceipt =
oItem.scheduleItem[iSchedule].getScheduleReceipt();
                    int netRequirement = grossRequirement - scheduleReceipt -
(projectOnHand - safetyStock);
                    int plannedOrderReceipt = 0;
```

```
rItem.scheduleItem[iSchedule].setGrossRequirement(grossRequirement);

                    int tmpRequire = netRequirement;
                    int tmpConculateRequire = 0;

                    if (iSchedule >= leadTime && netRequirement >  0)
                    {
                        for (int iPerio = iSchedule + 1; iPerio <
rItem.scheduleItem.Count && (iPerio - iSchedule) < poq; ++iPerio)
                        {
                            tmpConculateRequire -=
oItem.scheduleItem[iPerio].getScheduleReceipt();
                            tmpConculateRequire +=
oItem.scheduleItem[iPerio].getGrossRequirement();

                            if (tmpConculateRequire > 0)
                            {
                                tmpRequire += tmpConculateRequire;
                                tmpConculateRequire = 0;
                            }
                        }
                    }
                    else
                    {
                        tmpRequire = 0;
                    }

                    if (tmpRequire > 0)
                    {
                        plannedOrderReceipt = tmpRequire;
                        orderringCostTotal += orderingCost + (unitCost *
(double)plannedOrderReceipt);

                        if (netRequirement > 0)
                        {

rItem.scheduleItem[iSchedule].setNetRequirement(netRequirement);
                        }


rItem.scheduleItem[iSchedule].setPlannedOrderReceipt(plannedOrderReceipt);
                        rItem.sumQuantity += plannedOrderReceipt;

                        if (iSchedule >= leadTime)
                        {
                            rItem.scheduleItem[iSchedule -
leadTime].setPlannedOrderRelease(plannedOrderReceipt);
                        }
                    }

                    projectOnHand = scheduleReceipt + plannedOrderReceipt +
projectOnHand - grossRequirement;
                    holdingCostTotal += (double)projectOnHand * holdingCost;

                    rItem.scheduleItem[iSchedule].setProjectOnHand(projectOnHand);
                }
```

```
                rItem.cost = holdingCostTotal + orderringCostTotal;
            }
        }
    }

    class DQ : doReport
    {
        private int dqData = 0;

        private void getDataForDQ(string dq, double[] storeDQ)
        {
            string[] tmp1 = dq.Split('/');
            int offset = 0;

            for (int i = 0; i < tmp1.Length; ++i)
            {
                for (int j = 0; j < tmp1[j].Split('-').Length; ++j)
                {
                    if (string.IsNullOrEmpty(tmp1[i].Split('-')[j]) == false)
                    {
                        storeDQ[offset] = double.Parse(tmp1[i].Split('-')[j]);
                    }
                    else
                    {
                        storeDQ[offset] = 0.0;
                    }

                    ++offset;
                }
            }
        }

        private void calculateDQ(dataItem item)
        {
            double[] storeDQ = new double[9] { 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0 };

            this.getDataForDQ(item.dqData, storeDQ);
            this.dqData = this.eoq;

            double holdingCost = item.holdingCost;
            double orderingCost = item.orderringCost;

            double tc = double.MaxValue;
            double tmpTC = 0.0;

            int tmpDQ = this.eoq;

            for (int i = 0; i < 3; ++i)
            {
                int f0 = i * 3 + 0;
                int f1 = i * 3 + 1;
                int f2 = i * 3 + 2;

                if (storeDQ[f0] <= eoq)
                {
                    if (eoq <= storeDQ[f1])
                    {
```

```
                        tmpTC = (orderingCost * demand) / (double)eoq +
(holdingCost * (double)eoq) / 2 + demand * storeDQ[f2];
                    }
                    else
                    {
                        tmpTC = (orderingCost * demand) / storeDQ[f0] +
(holdingCost * storeDQ[f0]) / 2 + demand * storeDQ[f2];
                        tmpDQ = (int)storeDQ[f0];
                    }

                    if (tmpTC < tc)
                    {
                        tc = tmpTC;
                        this.dqData = tmpDQ;
                    }
                }
            }
        }

        public DQ(IDictionary<string, dataItem> otherItems)
        {
            this.methodName = "Discount_Quantity";
            this.requiredItems = new Dictionary<string, dataItem>(otherItems);

            for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
            {
                var rItem = requiredItems.ElementAt(iItem).Value;
                var oItem = otherItems.ElementAt(iItem).Value;

                this.calculateDQ(oItem);
                int minOrder = this.dqData;

                this.generalMethod(rItem, oItem, minOrder, Math.Max);
            }
        }
    }

    class LTC : doReport
    {
        public LTC(IDictionary<string, dataItem> otherItems)
        {

            this.methodName = "Least_Total_Cost";
            this.requiredItems = new Dictionary<string, dataItem>(otherItems);

            for (int iItem = this.requiredItems.Count - 1; iItem >= 0; --iItem)
            {
                var rItem = requiredItems.ElementAt(iItem).Value;
                var oItem = otherItems.ElementAt(iItem).Value;

                double holdingCost = oItem.holdingCost;
                double orderingCost = oItem.orderringCost;
                double unitCost = oItem.unitCost;

                rItem.cost = 0;

                int safetyStock = oItem.safetyStock;
                int leadTime = oItem.leadTime;
```

```
                int projectOnHand = oItem.projectOnHand;

                int[] dis = new int[rItem.scheduleItem.Count];
                int dis_run = 0;
                int dis_flag = 0;

                for (int i = 0; i < dis.Length; ++i)
                {
                    dis[i] = 0;
                }

                    //using all project on hand
                for (int iSchedule = 0; iSchedule < rItem.scheduleItem.Count;
++iSchedule)
                {
                    int grossRequirement =
oItem.scheduleItem[iSchedule].getGrossRequirement();
                    int scheduleReceipt =
oItem.scheduleItem[iSchedule].getScheduleReceipt();
                    int netRequirement = grossRequirement - scheduleReceipt -
(projectOnHand - safetyStock);


rItem.scheduleItem[iSchedule].setGrossRequirement(grossRequirement);

                    dis_run += dis_flag;
                    dis[iSchedule] = dis_run;

                    if (netRequirement > 0)
                    {
                        projectOnHand = Math.Min(projectOnHand, safetyStock);

rItem.scheduleItem[iSchedule].setNetRequirement(netRequirement);

                        dis_run = 0;
                        dis_flag = 1;
                    }
                    else
                    {
                        projectOnHand = scheduleReceipt + projectOnHand -
grossRequirement;
                    }

                    rItem.cost += (double)(projectOnHand) * holdingCost;
                    rItem.scheduleItem[iSchedule].setProjectOnHand(projectOnHand);
                }

                //order
                //pt1: [Y*unitCost + orderingCost] + (h + k)*(unitCost) +
orderingCost
                //pt2: [Y*unitCost + orderingCost] + (h + k)*(unitCost) + (h +
k)*holdingCost
                //pt3: [Y*unitCost + orderingCost] + (h + k)*(unitCost) +
orderingCost + k*holdingCost

                double nbCost = 0;
                int nbMaterial = 0;
```

```
                for (int iSchedule = rItem.scheduleItem.Count - 1; iSchedule >= 0;
--iSchedule)
                {
                    int netRequirement =
rItem.scheduleItem[iSchedule].getNetRequirement();

                    nbCost += holdingCost * (double)nbMaterial;
                    nbMaterial += netRequirement;
                    nbCost += unitCost * (double)netRequirement;

                    double tmp1 = holdingCost * (double)(dis[iSchedule] *
nbMaterial);
                    double tmp2 = nbCost - unitCost * (double)nbMaterial;
                    double comp = tmp1 + tmp2 - orderingCost;

                    if (comp <= 0)
                    {
                        rItem.scheduleItem[iSchedule].setNetRequirement(0);

rItem.scheduleItem[iSchedule].setPlannedOrderReceipt(nbMaterial);
                        rItem.scheduleItem[iSchedule -
leadTime].setPlannedOrderRelease(nbMaterial);
                        rItem.sumQuantity += nbMaterial;
                        rItem.cost += nbCost + orderingCost;

                        nbMaterial = 0;
                        nbCost = 0;
                    }
                    else
                    {
                        projectOnHand =
rItem.scheduleItem[iSchedule].getProjectOnHand();

rItem.scheduleItem[iSchedule].setProjectOnHand(projectOnHand + nbMaterial);

rItem.scheduleItem[iSchedule].setNetRequirement(netRequirement);
                    }
                }
            }
        }
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data;

namespace QLSX.DataTransferObject
{
    class dataMRP
    {
        private int grossRequirement = 0;
        private int scheduleReceipt = 0;
        private int projectOnHand = 0;
        private int netRequirement = 0;
```

```
        private int plannedOrderReceipt = 0;
        private int plannedOrderRelease = 0;
        DateTime date;

        public dataMRP() { }

        public dataMRP(dataMRP other)
        {
            this.grossRequirement = other.grossRequirement;
            this.scheduleReceipt = other.scheduleReceipt;
            this.projectOnHand = other.projectOnHand;
            this.netRequirement = other.netRequirement;
            this.plannedOrderReceipt = other.plannedOrderReceipt;
            this.plannedOrderRelease = other.plannedOrderRelease;
        }

        public int getGrossRequirement () {return this.grossRequirement;}
        public int getScheduleReceipt() { return this.scheduleReceipt; }
        public int getProjectOnHand() { return this.projectOnHand; }
        public int getNetRequirement() { return this.netRequirement; }
        public int getPlannedOrderReceipt() { return this.plannedOrderReceipt; }
        public int getPlannedOrderRelease() { return this.plannedOrderRelease; }
        public DateTime getDate() { return this.date; }

        public void setGrossRequirement(int otherGrossRequirement) {
this.grossRequirement = otherGrossRequirement; }
        public void setScheduleReceipt(int otherScheduleReceipt) {
this.scheduleReceipt = otherScheduleReceipt; }
        public void setProjectOnHand(int otherProjectOnHand) { this.projectOnHand
= otherProjectOnHand; }
        public void setNetRequirement(int otherNetRequirement) {
this.netRequirement = otherNetRequirement; }
        public void setPlannedOrderReceipt(int otherPlannedOrderReceipt) {
this.plannedOrderReceipt = otherPlannedOrderReceipt; }
        public void setPlannedOrderRelease(int otherPlannedOrderRelease) {
this.plannedOrderRelease = otherPlannedOrderRelease; }
        public void setDate(DateTime otherDate) { this.date = otherDate; }
    }

    class dataItem
    {
        public List<dataMRP> scheduleItem = new List<dataMRP>();
        //using for multiple and minimum method
        public int lostSize = 0;
        public int safetyStock = 0;
        public int leadTime = 0;
        public int projectOnHand = 0;
        public int sumQuantity = 0;
        public double holdingCost = 0;
        public double orderringCost = 0;
        public double unitCost = 0;
        public string dqData = "";

        public double cost = 0;

        //constructor
        public dataItem(dataItem other)
        {
```

```
            this.lostSize = other.lostSize;
            this.safetyStock = other.safetyStock;
            this.leadTime = other.leadTime;
            this.projectOnHand = other.projectOnHand;
            this.holdingCost = other.holdingCost;
            this.orderringCost = other.orderringCost;
            this.unitCost = other.unitCost;
            this.dqData = other.dqData;

            for (int i = 0; i < other.scheduleItem.Count; ++i)
            {
                this.scheduleItem.Add(new dataMRP(other.scheduleItem[i]));
            }
        }

        public dataItem(DataRow idtPS, int duration)
        {
            int lostSizeItem            =
int.Parse(idtPS["Lost_Size"].ToString());
            int safetyStockItem         =
int.Parse(idtPS["SafetyStock"].ToString());
            int leadTimeItem            = int.Parse(idtPS["Leadtime"].ToString());
            int projectOnHandItem       =
int.Parse(idtPS["Project_On_Hand"].ToString());
            double holdingCostItem      =
double.Parse(idtPS["HoldingCost"].ToString());
            double orderringCostItem    =
double.Parse(idtPS["OrderingCost"].ToString());
            double unitCostItem         =
double.Parse(idtPS["Unitcost"].ToString());
            string dqDataItem = idtPS["Discount_Quantity"].ToString();

            this.lostSize = lostSizeItem;
            this.safetyStock = safetyStockItem;
            this.leadTime = leadTimeItem;
            this.projectOnHand = projectOnHandItem;
            this.holdingCost = holdingCostItem;
            this.orderringCost = holdingCostItem;
            this.unitCost = unitCostItem;
            this.dqData = dqDataItem;

            //this.scheduleItem.AddRange(Enumerable.Repeat(new dataMRP(),
duration));
            for (int i = 0; i < duration; ++i)
            {
                this.scheduleItem.Add(new dataMRP());
            }
        }

        public dataItem(DataRow idtPS, int duration, DateTime date)
        {
            int lostSizeItem = int.Parse(idtPS["Lost_Size"].ToString());
            int safetyStockItem = int.Parse(idtPS["SafetyStock"].ToString());
            int leadTimeItem = int.Parse(idtPS["Leadtime"].ToString());
            int projectOnHandItem =
int.Parse(idtPS["Project_On_Hand"].ToString());
            double holdingCostItem =
double.Parse(idtPS["HoldingCost"].ToString());
```

```
            double orderringCostItem =
int.Parse(idtPS["OrderingCost"].ToString());
            double unitCostItem = double.Parse(idtPS["Unitcost"].ToString());
            string dqDataItem = idtPS["Discount_Quantity"].ToString();

            this.lostSize = lostSizeItem;
            this.safetyStock = safetyStockItem;
            this.leadTime = leadTimeItem;
            this.projectOnHand = projectOnHandItem;
            this.holdingCost = holdingCostItem;
            this.orderringCost = holdingCostItem;
            this.unitCost = unitCostItem;
            this.dqData = dqDataItem;

            //this.scheduleItem.AddRange(Enumerable.Repeat(new dataMRP(),
duration));
            for (int i = 0; i < duration; ++i)
            {
                this.scheduleItem.Add(new dataMRP());
                this.scheduleItem[i].setDate(date.AddDays(i));
            }
        }
    }

    class dataTree
    {
        private int parentID = 0;
        private int leadTime = 0;
        private int masterItem = 0;
        private int sumLeadTime = 0;
        private int type = 0;
        private bool flag = false;
        private double value;
        private string idName = "";

        public int getParentID() { return this.parentID; }
        public int getLeadTime() { return this.leadTime; }
        public int getSumLeadTime() { return this.sumLeadTime; }
        public int getType() { return this.type; }
        public bool getFlag() { return this.flag; }
        public double getValue() { return this.value; }
        public int getMasterItem() { return this.masterItem; }
        public string getIdName() { return this.idName; }

        public void setParentID(int otherParentID) { this.parentID =
otherParentID; }
        public void setLeadTime(int otherLeadTime) { this.leadTime =
otherLeadTime; }
        public void setSumLeadTime(int otherSumLeadTime) { this.sumLeadTime =
otherSumLeadTime; }
        public void setType(int otherType) { this.type = otherType; }
        public void setFlag(bool otherFlag) { this.flag = otherFlag; }
        public void setValue(double otherValue) { this.value = otherValue; }
        public void setMasterItem(int otherMasterItem) { this.masterItem =
otherMasterItem; }
        public void setIdName(string otherIdName) { this.idName = otherIdName; }
    }
```

```
    class requiredMRP
    {
        private int rootID = 1;
        private IDictionary<string, dataItem> requiredItems = new
Dictionary<string, dataItem>();
        //IDictionary<id in database, datatree>
        //save data from database
        private IDictionary<int, dataTree> reStructureTree = new Dictionary<int,
dataTree>();
        //
        private IDictionary<int,List<dataTree>> materialTree = new Dictionary<int,
List<dataTree>>();

        //private method
        private void createDataStructure(DataTable dtPS, int duration, DateTime
frompicker)
        {
            for (int i = dtPS.Rows.Count - 1; i >= 0; --i)
            {
                string idName   = dtPS.Rows[i]["Id_Name"].ToString();
                int type        = int.Parse(dtPS.Rows[i]["Type"].ToString());
                int id          = int.Parse(dtPS.Rows[i]["Id"].ToString());
                int parentID    = int.Parse(dtPS.Rows[i]["Parent_Id"].ToString());
                int leadTime    = int.Parse(dtPS.Rows[i]["Leadtime"].ToString());
                double value    = double.Parse(dtPS.Rows[i]["Value"].ToString());

                //restructure product tree
                reStructureTree.Add(id, new dataTree());

                reStructureTree[id].setFlag(false);
                reStructureTree[id].setParentID(parentID);
                reStructureTree[id].setType(type);
                reStructureTree[id].setValue(value);
                reStructureTree[id].setLeadTime(leadTime);
                reStructureTree[id].setIdName(idName);

                //create schedule structure for "additional item"
                if (this.requiredItems.ContainsKey(idName) == false && type == 2)
                {
                    this.requiredItems.Add(idName, new dataItem(dtPS.Rows[i],
duration, frompicker));
                }
            }
        }
        private void createScheduleReceipts(DataTable dtSR, int duration, DateTime
frompicker)
        {
            //create schedule receipts
            for (var i = dtSR.Rows.Count - 1; i >= 0; --i)
            {
                string idName = dtSR.Rows[i]["Id_Name"].ToString();

                if (this.requiredItems.ContainsKey(idName))
                {
                    int offset =
Convert.ToInt32((DateTime.Parse(dtSR.Rows[i]["Day"].ToString()) -
frompicker).TotalDays);
```

```
                    int receipt =
int.Parse(dtSR.Rows[i]["Schedule_Receipt"].ToString());

                    if (offset >= 0 && offset < duration)
                    {

this.requiredItems[idName].scheduleItem[offset].setScheduleReceipt(receipt);
                    }
                }
            }
        }
        private void createMaterialTree(int id)
        {
            if (this.reStructureTree.ContainsKey(id) == true)
            {
                //if was not checked
                if (this.reStructureTree[id].getFlag() == false)
                {
                    int parentID = this.reStructureTree[id].getParentID();

                    //the parent item is available
                    if (this.reStructureTree.ContainsKey(parentID) == true)
                    {
                        this.createMaterialTree(parentID);
                        int sumLeadTime = reStructureTree[parentID].getLeadTime()
+ reStructureTree[parentID].getSumLeadTime();
                        this.reStructureTree[id].setSumLeadTime(sumLeadTime);

                        if (parentID == this.rootID)
                        {
                            this.reStructureTree[id].setMasterItem(id);
                        }
                        else
                        {

this.reStructureTree[id].setMasterItem(this.reStructureTree[parentID].getMasterIte
m());
                        }
                    }

                    this.reStructureTree[id].setFlag(true);
                }
            }
        }

        //public

        public IDictionary<string, dataItem> getScheduleItems() {return
this.requiredItems;}

        public IDictionary<string, dataItem> getScheduleItems(string itemName) {

            IDictionary<string, dataItem> tmp = new Dictionary<string,
dataItem>();

            if(this.requiredItems.ContainsKey(itemName) == true) {

                tmp.Add(itemName, this.requiredItems[itemName]);
```

```
            }

            return tmp;
        }

        //constructor
        public requiredMRP(DataTable dtMS, DataTable dtSR, DataTable dtPS,
DateTime frompicker, DateTime topicker)
        {
            int duration = Convert.ToInt32((topicker - frompicker).TotalDays);

            this.createDataStructure(dtPS, duration, frompicker);
            //create schedule receipts
            this.createScheduleReceipts(dtSR, duration, frompicker);

            this.reStructureTree[this.rootID].setFlag(true);

            for (int i = dtPS.Rows.Count - 1; i >= 0; --i)
            {
                int idNode = int.Parse(dtPS.Rows[i]["Id"].ToString());

                if (this.reStructureTree.ContainsKey(idNode) == true)
                {
                    this.createMaterialTree(idNode);

                    if (idNode != this.rootID)
                    {
                        int masterItem =
this.reStructureTree[idNode].getMasterItem();
                        int typeItem = this.reStructureTree[idNode].getType();

                        if (masterItem != idNode && typeItem == 2)
                        {
                            if (this.materialTree.ContainsKey(masterItem) ==
false)
                            {
                                this.materialTree.Add(masterItem, new
List<dataTree>());
                            }


this.materialTree[masterItem].Add(this.reStructureTree[idNode]);
                        }
                    }
                }
            }

            for (int iMaster = dtMS.Rows.Count - 1; iMaster >= 0; --iMaster)
            {
                string idName = dtMS.Rows[iMaster]["Id_Name"].ToString();
                //int idNode =
int.Parse(dtMS.Rows[iMaster]["Id_Node"].ToString());
                int idNode = frmReport.get_Id(idName);
                int day =
Convert.ToInt32((DateTime.Parse(dtMS.Rows[iMaster]["Day"].ToString()) -
frompicker).TotalDays);
                int quantity =
int.Parse(dtMS.Rows[iMaster]["Quantity"].ToString());
```

```
                for (int iMaterial = this.materialTree[idNode].Count - 1;
iMaterial >= 0; --iMaterial)
                {
                    int dayPlanOrder = day -
this.materialTree[idNode][iMaterial].getSumLeadTime();
                    string idMaterialName =
this.materialTree[idNode][iMaterial].getIdName();

                    if (dayPlanOrder >= 0)
                    {
                        int grossRequirement =
(int)Math.Ceiling(this.materialTree[idNode][iMaterial].getValue() *
(double)quantity);

                        if (dayPlanOrder < duration && dayPlanOrder >= 0)
                        {

this.requiredItems[idMaterialName].scheduleItem[dayPlanOrder].setGrossRequirement(
grossRequirement);
                        }
                    }
                }
            }
        }

        public requiredMRP(IDictionary<string, dataItem> otherItems)
        {
            for (int iItem = 0; iItem < otherItems.Count; ++iItem)
            {
                dataItem ndata = new dataItem(otherItems.ElementAt(iItem).Value);
                string name = otherItems.ElementAt(iItem).Key;
            }
        }
    }
}
```