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

MATERIAL REQUIREMENT PLANNING - A CASE STUDY IN LTD. TOAN BICH

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

Student: NGUYEN NGOC TRAN VY
Thesis advisor: Dr. NGUYEN VAN CHUNG

Ho Chi Minh city, Vietnam

6/2015

MATERIAL REQUIREMENT PLANNING - A CASE STUDY IN LTD. TOAN BICH

By

NGUYEN NGOC TRAN VY

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

6/2015

Signature of Student: ____

Nguyen Ngoc Tran Vy

Certified by : ______________________________

Dr. Nguyen Van Chung

Thesis Advisor

Approved by : ______________________________

Assoc. Prof. Ho Thanh Phong

Head of ISE Department

MATERIAL REQUIREMENT PLANNING - A CASE STUDY IN LTD. TOAN BICH

By

Nguyen Ngoc Tran Vy

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

6/2015

ABSTRACT

Material Requirement Planning (MRP) is a well known tool in manufacturing and been using worldwide but in Vietnamese company this tool is not common use. This thesis is about applying MRP in production in Toan Bich, a furniture company. After a long time working in furniture producing, the company becomes more and more famous in this kind of field, because of this development, the demand also increase. Along with the increase, the company has to face with many problems. A program has been provided in this thesis in order to help company's employees make decision about ordering material so that they can meet customer's demand on time and also save money for inventory at the same time.

Keywords: Material requirement planning, Production planning, Bill of material, Master production planning, Programming

Thesis advisor: Dr. Nguyen Van Chung

ACKNOWLEDGMENTS

I would like to express my deep gratitude to my supervisor, Dr. Nguyen Van Chung for the patience guidance, continuous support, encouragement, motivation and his assistance in all the time of research and writing of this thesis. Thanks to him for given me such a great opportunity to work with. Without his persistent help I could not have finished this thesis.

I also express my appreciation to board of director of Ltd. Toan Bich for allowing me to make the thesis in the company and the company employees for their enthusiasm support. Although this is the busiest time in the year but they have helped me a lot with collecting data for my study.

There is a special thanks to International University for the convenient they have provide to support student for their thesis.

Finally, I would like to express my gratitude to my friends for their cheering and being by my side along the time I work on my thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Introduction

MRP – Material Requirement Planning System is a tool using to manage manufacturing process. It is include production planning, scheduling and inventory controlling system.

Although MRP is an old tool using worldwide, in Vietnam, it is only known in a few years. Because of some objective reasons, MRP is not used commonly in many companies. According to Chamber of Commerce and Industry Vietnam, until June 2006, only 1.1% of business used MRP in manufacturing. The number of company used MRP only reach 7%.

In 1960's, most of the companies use computers for common accounting functions. Joseph Orlicky IBM began using computers to plan and control inventory. After 1972, APICS actively promote the use of MRP and since then has become the model MRP production management primarily in the United States. In 1989, sales of MRP software and implementation services exceeded one billion dollars. By early 2000, SAP's sales exceed $ 3.2 billion alone.

The main focus of MRP is to streamline tasks and orders to meet material requirements by external demand. Two dimensions of MRP are the amount and duration.

MRP tries to determine the appropriate amount of production of all kinds of items from the final product is sold out, the components used to build the final product, to buy inputs such as raw materials rough. It shall determine the time of production that facilitates order to meet the due date. In many MRP systems, time is divided into groups, although some systems use a continuous period. In the kind of bucket system, all the requests are due at the beginning of the bucket. MRP works with both finished products and items last and their constituent parts are called lower-level items. The relationship between the end items and lower commodities is described by BOM, bill of material or product structure.

1.2. Problem Statement

Ltd. Toan Bich (TOBI) was established in 2002 with the main field operated is office furniture. TOBI haven’t applied MRP for manufacturing. Base on the knowledge about the company, it is defined several problem in this company.

TOBI usually have some problem with incoming material. To produce any kind of furniture, workers have to follow some step:

- Cutting wood
- Abrading
- Assembly
- Paste

While producing, it is importance having enough kind of material with enough quantity, otherwise the producing line would be delay this is cause of waste and missing the deadline with customer. On the other hand, lead time of each material is difference and some of them are unstable, this cause many difficulties in making decision for ordering and delivering material. Lacking of material is the reason bottleneck appears in the production line.

Furthermore, many kind of product in Toan Bich share the same kind of handle, wheels, fasteners, est. and all of them share the same fasteners when they are assembled. With a large number of products sharing so many kinds of items, ordering and purchasing becomes a big problem that the company have to deal with every time a new order comes.

1.3. Objective

The objective of this research is to provide to the company a useful tool for ordering material to ensure that the company can order the right part, right quantity, at the right time in order to plan for a complete and accurate load and keep the due date valid.

1.4. Scope and limitations

The time of the thesis is limited, 6 months. There are many of kind of product, in such short time it is impossible to solve all problem of all items. On the other hand, this is the busy moment for all the employees in TOBI because of many orders which cause some trouble in collecting data.

There are more than 1800 products in TOBI. The model focuses on planning purchasing activities and ensures the availability of product for customer delivery for only 6 highline kinds of product that manufacturing in the factory.

1.5. Research Methodology

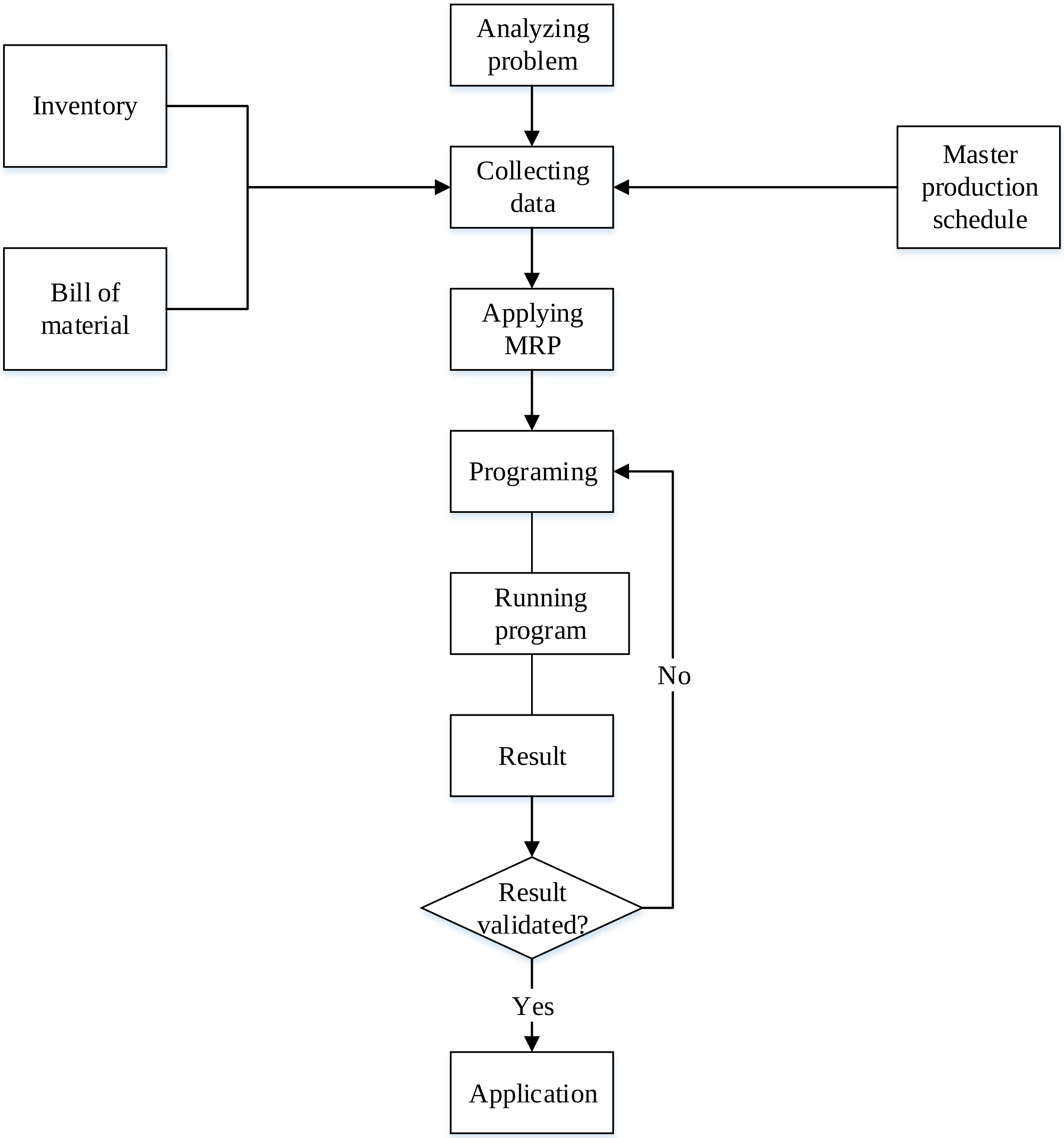


Figure 1: Research flow chart

This research follows these few steps below:

- Analyzing problem: the company have no tool to control their inventory and no guideline for ordering or purchasing material
- Collect data from the company: inventory data, bill of material, master production schedule
- Applying MRP to solve the problem: the aims of this algorithm will be: recommended product schedule, recommended purchasing schedule
- Programming base on the method: because of the scale of data is too large, it is needed to build a program to solve the problem
- Running the program
- Getting result by using the program
- Check the result: if the result is valid, the result will be announce otherwise, the program will be built again
- Announce the result

1.6. Organization of the thesis

This thesis includes 6 chapters:

Chapter 1: Introduction

This chapter is about general knowledge of Material Requirement Planning, the reason this topic is working on and the purpose of this thesis. Research methodology, structure Scope and limitations of the thesis also described.

Chapter 2: About Toan Bich

Some general information about Toan Bich is mention in this chapter.

Chapter 3: Literature Review

This chapter including some related work with this topic and some basic concept that mentioned in the thesis.

Chapter 4: MRP Program

The algorithm of the program when applied two lot sizes is a lot for a lot and multiple lot sizes. The instruction also described in this chapter.

Chapter 5: Program result

The result of the program is show in this chapter.

Chapter 6: Conclusion

Chapter 2 TOAN BICH COMPANY

2.1. About Toan Bich

Established since 2002, Ltd. Toan Bich - with brand TOBI - operating company specialized in office furniture. With extensive experience throughout the years as well as warehouse products rich design, quality and highly advanced applications, brand TOBI has confirmed its position in the furniture industry in particular and on with the consumer market in general.

Staff of experienced personnel:

- Superior design - dynamic personality and creativity - is the foundation to design various products, bringing high practical applicability, contributing to a unique style, a personal brand - Trade Brand TOBI

- Sales Department dedicated technically savvy, enthusiastic advice and insights from the customer want. We are committed to bringing the confidence and satisfaction to customers.

- Officials and employees of construction, monitoring at work and in factory production has experienced, skilled. And the most important is the spirit of responsibility, enthusiasm at work will give customers the quality products and the most beautiful.

<u>The philosophy of the company:</u>

"Each customer is a brick building corporate value". Therefore, all officers and employees of TOBI always try to cultivate more knowledge and experience to offer the most efficient products for our customers.

<u>Mission:</u>

TOBI with their furniture would be a reliable place to adorn perfection beauty availability of Office you. We always strive to become the best choice of customers [12]

2.2. Product

TOBI has experience in produce furniture and has more than 50 types of products.

(Source: tobi.com.vn) [12]

Toan Bich is a well-known furniture company, the company has contracts with many famous companies, for example: NOKIA, JABIL, PANASONIC… company primarily operates in V-ship and Saigon Hi-Tech Park.

Figure 2: Products of Toan Bich

2.3. List of products

Because of the limitation of time and condition, this research only focuses on 6 types of wooden cabinet. The BOM of wooden cabinet THSG03

Table 1: List of product

ID	Name of product
F1	Wooden cabinet THSG03
F2	Wooden cabinet THSG04
F3	Wooden cabinet THSG05
F4	Wooden cabinet THSG06
F5	Wooden cabinet THSG07
F6	Wooden cabinet THSG08

The BOM of wooden cabinet THSG03 is showed below. The BOM of 5 other products is present in Appendix

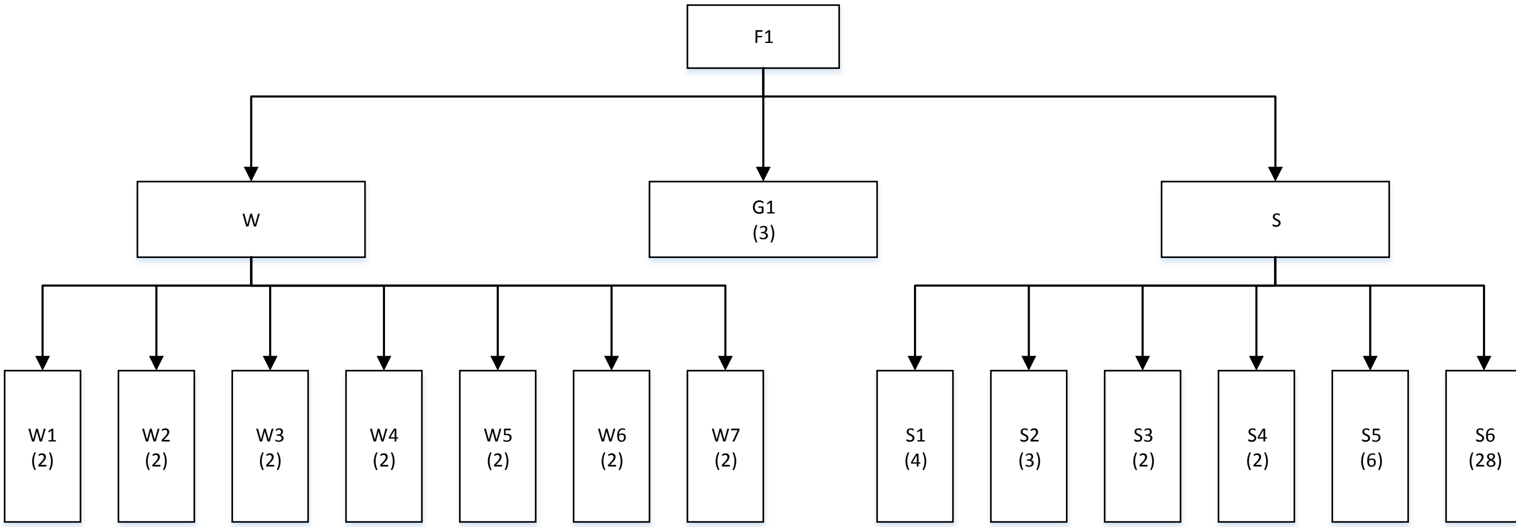


Figure 3: BOM of wooden cabinet THSG03

Table 2: List of material items

ID	PART NAME
G1	Glass part (size 80*30*0.5 cm)
G2	Glass part (size 80*20*0.5 cm)
W1	Wooden part (size 160*25*2 cm)
W2	Wooden part (size 90*25*2 cm)
W3	Wooden part (size 60*25*2 cm)
W4	Wooden part (size 30*25*2 cm)
W5	Wooden part (size 40*30*2 cm)
W6	Wooden part (size 30*20*2 cm)
W7	Wooden part (size 80*25*2 cm)
W8	Wooden part (size 80*25*2 cm)
W9	Wooden part (size 35*12*2 cm)

W10	Wooden part (size 35*25*1 cm)
W11	Wooden part (size 95*25*1 cm)
W12	Wooden part (size 95*25*5 cm)
W13	Wooden part (size 60*25*2 cm)
W14	Wooden part (size 36*20*2 cm)
W15	Wooden part (size 80*35*1.5 cm)
W16	Wooden part (size 100*25*2 cm)
W17	Wooden part (size 60*30*2 cm)
W18	Wooden part (size 135*100*2 cm)
W19	Wooden part (size 135*25*2 cm)
W20	Wooden part (size 75*25*2 cm)
W21	Wooden part (size 23*20*2 cm)
S1	Footstall
S2	Handle type 1
S3	Handle type 2
S4	Lock
S5	Connecting piece
S6	Fastener
S7	Handle type 3
S8	Footstall type 2

Chapter 3 LITERATURE REVIEW

3.1. Related works

Yolanda Masnita and *Tania AnandaMahdani* (2012) [5] have proposed the implementation process factors and results of the conduct of MRP, and to know the relationship between these factors. By using Simple Random Sampling method with alpha = 0.9058 and r-value = 0.444, there are 6 factors are defined, there are: the main management support and formal planning, the accuracy of the data, setting the organization's education or training, special planning or control policies and procedures, characteristics of software, as well as individual characteristics of employees.

According to *Edmund W. Schuster* MRP [8] systems suffer from shortcomings that limit their usefulness for firms in the process industries. They proposed a two-level, spreadsheet-based procedure developed at Welch's, the largest processor of Concord and Niagara grapes, to perform integrated capacity planning. Through a case study of Welch's they explore the attributes of the process industries that permit capacitated material requirements planning (CMRP) in practice. By applying mathematical programming, heuristics, and recursive solution methods they envision large scale CMRP. In the longer term, CMRP will define the next generation of materials planning systems.

Robert C. Leachman and *Jose F. Goncalves* [6] proposed modifications to the basic MRP algorithm so as to accommodate arbitrary (non-integer) lead times in the generation of rate-based production schedules. To do this, they re-interpret the MRP algorithm to express it as a series of manipulations of cumulative production curves, and then extend the algorithm for the case that production schedules of non-integer processes must be rate-based in integer periods. The modifications are computationally practical, and the resulting schedules offer significant reductions in interplant inventory compared to the result of the usual MRP calculations with rounded-up lead times.

According to *R. B. Cooper* and *R. W. Zmud*, 1989, [9] Manufacturing Planing Requirement is useful tool for manufacturing firm because this tool can help manage production and inventory. On the other hand, he also confirmed that this tool "has a significant history of implementation problem". To understand why these problems occur, in their paper, they examine factors effecting MRP implementation from an innovation diffusion perspective. Major contributions of the study include the identification of significant factors that affect the infusion of MRP but which are opposite to those facilitating the adoption of MRP.

In short, there are several things the inability of the industry in the application of MRP, based on initial observation of researchers. This shows the need to do research on using MRP for industrial manufacturing.

3.2. Basic concept

3.2.1. Material requirement planning

3.2.1.1. Definition of MRP

Material Requirements Planning (MRP) is a production planning and inventory control system based on PC. MRP is concerned with both production planning and inventory control. It is a material control system that tries to keep sufficient inventory levels to ensure that the necessary documents are available when needed. MRP is applicable in situations of multiple items with complex bills of materials. MRP is not useful for job shops or for continuous process are closely linked.

The main goal of an MRP system is to simultaneously:

1. Ensure the availability of raw materials, components and products for planning production and delivery to customers,

2. Maintain the lowest possible level of inventory,

3. Plan manufacturing activities, delivery schedules and purchasing activities.

MRP is particularly suited to establish production that needs a lot of components and subassemblies depends on the demand of such commodities facing external demand. The demand for final goods is independent. In contrast, demand for the ingredients used to produce final goods depend on demand for the end position. The distinction between independent demand and depend very important in the classification of inventory items and develop systems to manage the items in each classification needs. MRP system was developed to cope better with the demands dependent items.

Three important inputs of an MRP system is the overall production schedule, the product structure records, and the records inventory status. Without these basic inputs of MRP system cannot operate.

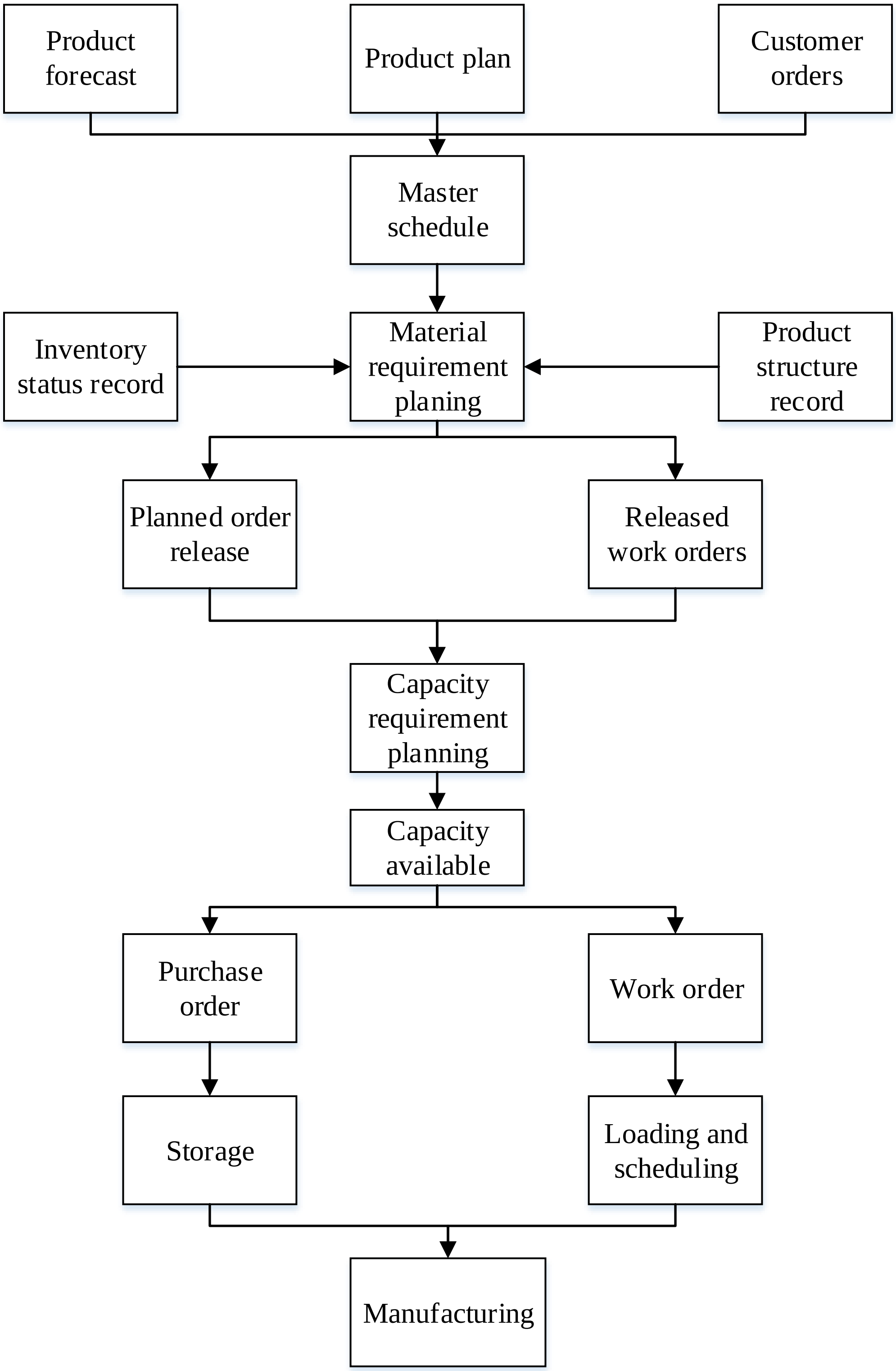


Figure 4: MRP system

(Source: www.productivity.in) [11]

3.2.1.2 MRP algorithm

Let n be the current level. The MRP algorithm has the following step:

Step 1: Determine the gross requirement for all products of the current BOM level per time period (G(t)).

Step 2: Determine the net requirements N(t) for the part of level n at the time period t

$$N(t) = G(t) - S(t) - H(t-1)$$

Where S(t) is the schedule receipt at time period t and H(t-1) is the inventory at the previous time period.

Step 3: Generate the production requirements (order) P(t) for all the product of the current BOM level for the specific time period t according to the lot policy. If N(t) = 0 then P(t) = 0. Otherwise the quantity and the date of the order are determined from the lot policy.

Step 4: calculate the inventory H(t) for the products of level n at time the period t

$$H(t) = S(t) + P(t) + H(t - 1) - G(t)$$

Step 5: If all the time period completed go to step 6. Otherwise continue to the next time period (set t = t + 1) and go to step 1

Step 6: Determine schedule release R(t) of all production orders of step 3 for all products of the current level. If the lead time of the product is L, then R(t – L) = P(t).

Step 7: If the requirements of the products of all levels have been determined, then stop. Otherwise, continue to step 8.

Step 8: Determine requirements of all products of the next BOM level. This is the outcome of the single level explosion of BOM structure of new requirements for already existing products.

Step 9: Continue to the next BOM level, setting n = n + 1 and go to step 1 again

3.2.1.3. MRP objective

MRP is the main theme of "getting the right materials to the right place at the right time." Specific organizational objectives often associated with the design and implementation of MRP can be determined in one of three main dimensions, namely: inventory, priorities and capacity:

Table 3: MRP objectives

Dimension	Objective specifics
Inventory	- Order the right part - Order the right quantity - Order the right time
Priorities	- Order with the right due date - Keep the due date valid
Capacity	- Plan for complete load - Plan for an accurate load - Plan for an adequate time to view future load

The goal of MRP should be identified related to the inputs and outputs associated with it with. Inputs are delineated master production schedule, bill of materials, etc. Therefore, a clear specification of objectives MRP should be combined with a corresponding clearly describes the goal of MRP input and output MRP.

3.2.2. Master production plan (MPS)

A master production schedule (MPS) is a plan for individual items to production in each period, such as manufacturing, payroll, inventory, etc. It is often associated with manufacturing, where plans indicate when and how much of each product will be required. This plan determines the number of critical processes, components, and other resources to optimize production and to identify bottlenecks, and predict demand and finished goods. Since a plant operation control MPS much, accuracy and its ability to survive significant impact profitability. MPS's typically generated by user software tweaks.

Due to restrictions on the software, but especially the intense work required of "master production scheduling", the schedule does not include every aspect of the production, but only important factor has been demonstrated effective control of them, such as forecasts of demand, production costs, storage costs, lead time, working hours, capacity, inventory levels, storage available, and parts supply. The choice of what to models varies between companies and factories. MPS is a statement of what the company expects production and sales (i.e., quantities are produced, the number of employees, date, available to promise, and the balance of the project).

3.2.3. Bill of material (BOM)

A bill of materials is a list of materials, sub-assembly, assembly intermediate, sub-components, parts and the quantity of each required to produce a final product. A BOM can be used for communication between the manufacturing partners, or limited to a single factory. [11]

A BOM can identify the products as they are designed (bill of materials engineering), when they were ordered (sales bill of materials), as they are built (material production bill materials), or when they are maintained (bill of material services or fake bill of material). The different types of BOMs are depends on business needs and uses in which they are intended. In industrial processes, the BOM also called formula, formula, or a list of ingredients. In electronics, the BOM represents the list of ingredients used in printed wiring boards or printed circuit board. When designing the electrical circuit is completed, the Board adopted the list in the PCB layout engineer and owner who would buy the ingredients needed for the design.

3.2.4. Visual studio 2012

Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It is used to develop a computer program for Microsoft Windows, as well as websites, web applications and web services. Visual Studio development platform using Microsoft software such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store and Microsoft Silverlight. It can produce both computer languages and managed code. [13]

Visual Studio includes a code editor supporting IntelliSense as well as improved the code. Integrated debugger works both on the level debugger and debug source code level machine. Other integrated tools include a forms design build GUI applications, web design, design and design class database schema. It accepts plug-ins enhance the functionality at almost all levels, including more support for version control system (like Subversion) and adding a new set of tools like editors and visual design for domain specific language or set of tools for different aspects of the software development process.

Visual Studio support many different programming languages and allows the code editor and debugger to support (to varying degrees), virtually every programming language. These languages include C integrated, C ++ and C ++ / CLI (via Visual C ++), VB.NET (Visual Basic.NET through), C rise (via Visual C #) and F ups (as of Visual Studio 2010 [2]). Support for other languages like J ++ / J ups, Python and Ruby through a separate installation services. It also supports XML / XSLT, HTML / XHTML, JavaScript and CSS.

On 13/9, Vietnam Microsoft officially launched Visual Studio 2012, integrated development solution for building and managing applications. Visual Studio 2012 combined with Microsoft's platform, provide a rich development environment and help build compelling applications to meet modern needs demand ever-changing needs access anywhere and diverse data needs, customers' complex. Modern applications:

- User Guide: Helping build client applications take the center and are accessible from any device
- Social: Integration with the user identifier, connect to colleagues and friends

- User Data: Data integration with instant job, allowing decision-making context.

Visual Studio 2012 helps the application life cycle management of modern applications (Modern Application Lifecycle Management) becomes faster, reduce risk, and continually provide high-quality applications for businesses and consumers. Measures to manage modern application lifecycle Visual Studio 2012 include:

- Management tool and flexible project: help keep the group to the transparent with unified information. This tool includes product backlog management, release planning / closing number, managing and forecasting capacity of the group, access the information by integrating with instruments group (management tools requirements management, development, testing, and feedback), and the ability to display real-time delivery status and risks.
- Tools and processes to help split the group or demolished barriers of functional integration. Integrated user and stakeholder feedback given early and often on product features and orientation, productivity and collaboration of programmers, integration testing to decrease time error detection and patches. Also integrated with operator (DevOps) to accelerate the application development time and reduce MTTR (mean time to repair).

3.4.5. Lot for lot sizing rule

- Order (or produce) only enough quantity required in each period to satisfy gross requirements and to keep safety stock as less as possible.

- “Easy to use, and follow Just-In-Time philosophy of ordering/producing only when required.”
- Lot size can be modified easily for purchase discounts or restrictions, scrap allowances, process constraints, etc.
- Minimizes on-hand inventory, but maximizes number of orders placed (so can be expensive if setup/ordering costs are significant).

3.4.6. Multiple lot sizing rule

Some companies, they have some rules about ordering size, they only sell in lot and in a lot is a certain quantity of product. To buy these kinds of material, the customer have to consider the lot size of the product. For example, when the supplier sells footstall, the lot size is 4, if the requirement is 6, then the number of footstall they have to order is 8. Different kind of product, they have different lot size.

CHAPTER 4 MRP PROGRAM

4.1. Program algorithm

MRP program is built by using Microsoft Visual Studio 2012 (related by Microsoft Corporation) with simple interface design for user who has average knowledge in using computer and information technology. There are two kind of lot size rules are applied: lot for lot and multiple. The database of this program was built base on the data collected in Toan Bich. The source code of this program is show in Appendix.

T	Time
T_{end}	End of time
G	Gross requirement
N	Temporary variables
S	Schedule receipt
POH(T-1)	Project on hand of the last T
PORL	Plan order release
P	Planned order receipt
Lotsize	Lot size
L	Lead time

4.1.1. Multiple

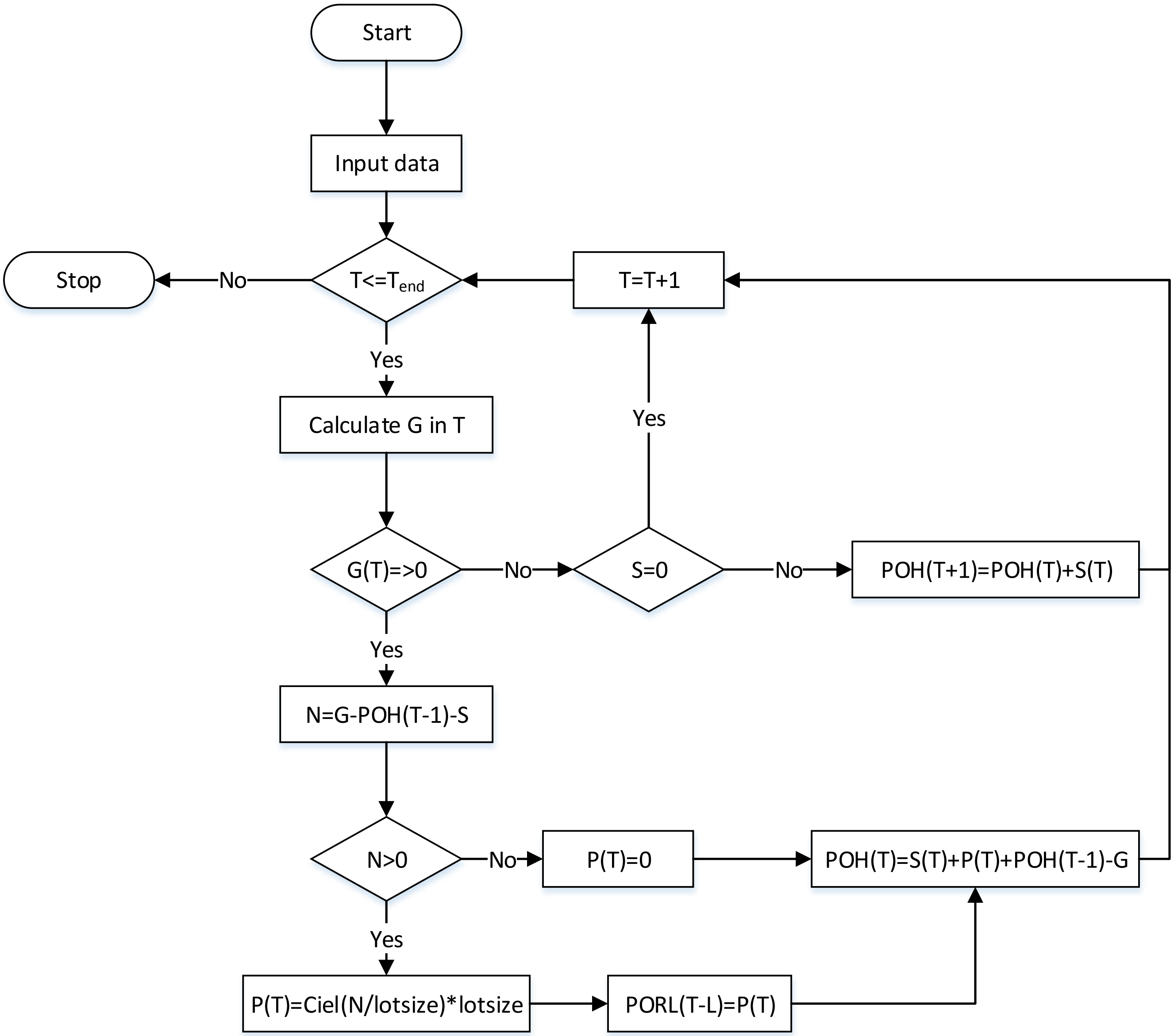


Figure 5: Program algorithm for multiple lot size

4.1.2. Lot for lot

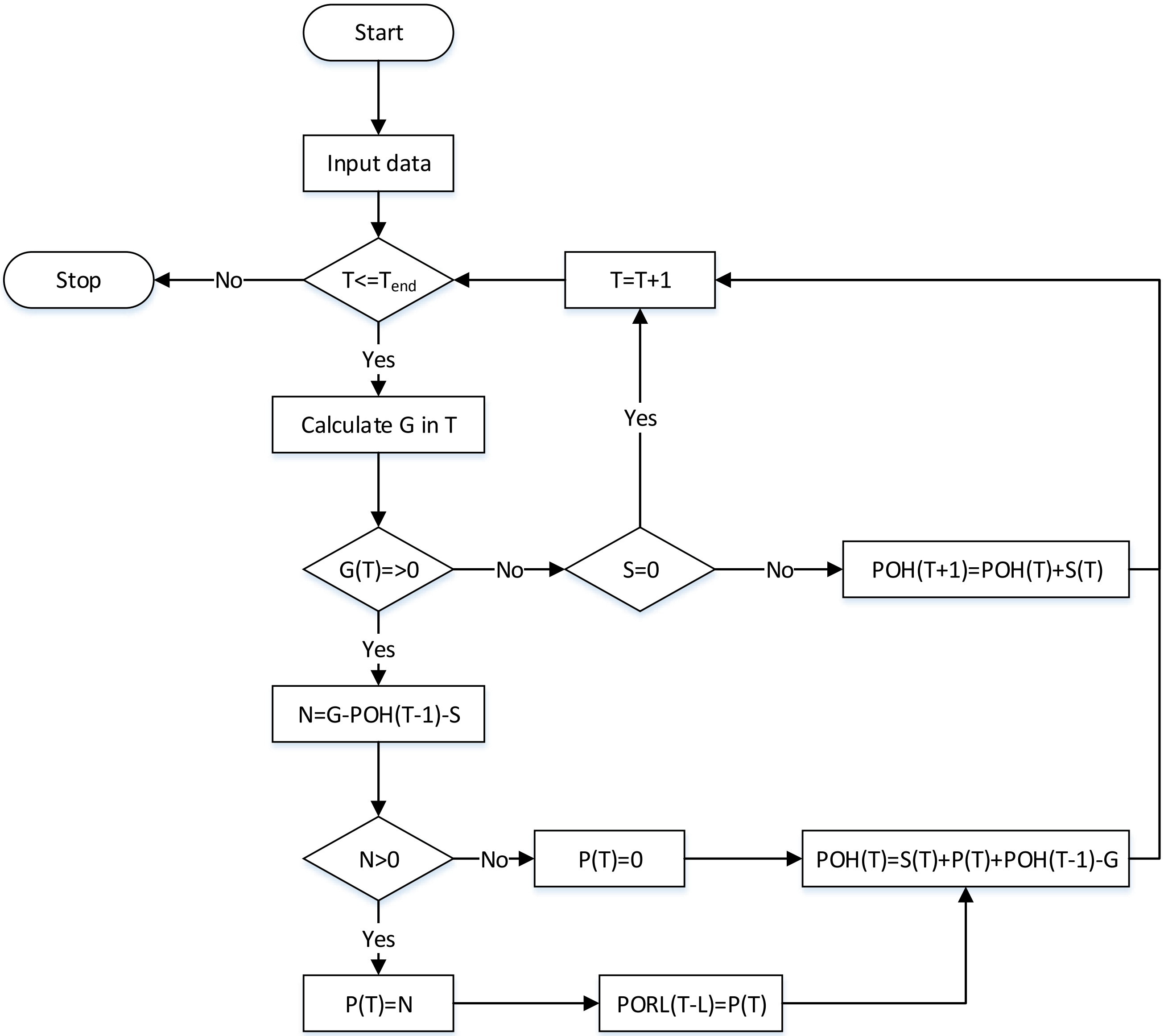


Figure 6: program algorithm for lot for lot

4.2. Data collecting

There are more than 18 kinds of wooden cabinet are being produced in Toan Bich factory. In this study, only 6 kinds of product have highest demand are collected. The data are collected in March 1st, 2015 to March 1st, 2016

Table 4: Demand data

	Product					
Week	F1	F2	F3	F4	F5	F6
9/2015	130	107	101	113	140	147
10						
11						
12						
13	121	130	134	109	130	137
14						
15						
16						
17	115	108	144	117	112	147
18						
19						
20						
21	122	135	116	118	146	125
22						
23						
24						
25	119	111	142	118	135	100
26						
27						
28						

29	119	145	115	144	112	101
30						
31						
32						
33	120	137	144	141	131	122
34						
35						
36						
37	110	142	129	125	125	146
38						
39						
40						
41	134	108	103	115	137	111
42						
43						
44						
1/2016	125	126	108	107	129	145
2						
3						
4						
5	103	128	135	149	131	103
6						
7						
8						
9	146	139	123	108	148	141

Table 5: Lot size, lead time and inventory of each item

ID	LOTSIZE	LEADTIME	SAFETY STOCK	INVENTORY
W5	0	4	40	123
W8	0	4	40	154
W1	0	4	20	121
W2	0	4	40	205
W3	0	4	40	174
W4	0	4	40	126
W6	0	4	40	123
W7	0	4	40	136
G1	0	5	50	50
S1	4	1	20	150
S2	24	1	10	126
S3	12	1	10	108
S4	15	1	10	101
S5	15	1	20	187
S6	6000	2	500	931
W9	0	4	40	136
W10	0	4	40	125
W11	0	4	40	176
W12	0	4	40	104
W13	0	4	40	127

W14	0	4	40	116
W15	0	4	40	199
W16	0	4	40	140
W17	0	4	40	190
W18	0	4	40	144
W19	0	4	40	137
W20	0	4	40	130
W21	0	4	40	147
G2	0	1	0	0
S7	12	1	10	163
S8	4	1	20	165
F2	0	1	0	2
F3	0	1	0	0
F4	0	1	0	1
F5	0	1	0	0
F6	0	1	0	0

4.3. Program Instruction

Program MRP (written in the programming language C #), including major items: items stored information of each item such as name, type (finished product, assembly, material), lot size rule (lot for lot, min, multiple), lot size What is the rule for calculating lot size min or multiple, lead time - hang time set (see the concepts), safety stock- and project inventory on hand inventory; bill of material build relationships trees hold items for one connection from the original material to go through the intermediate stage to the finished product assembly; master production scheduling schedules allow finished product; Sign cave tourism receipts schedule and report schedule export reports for each item ordered is material.

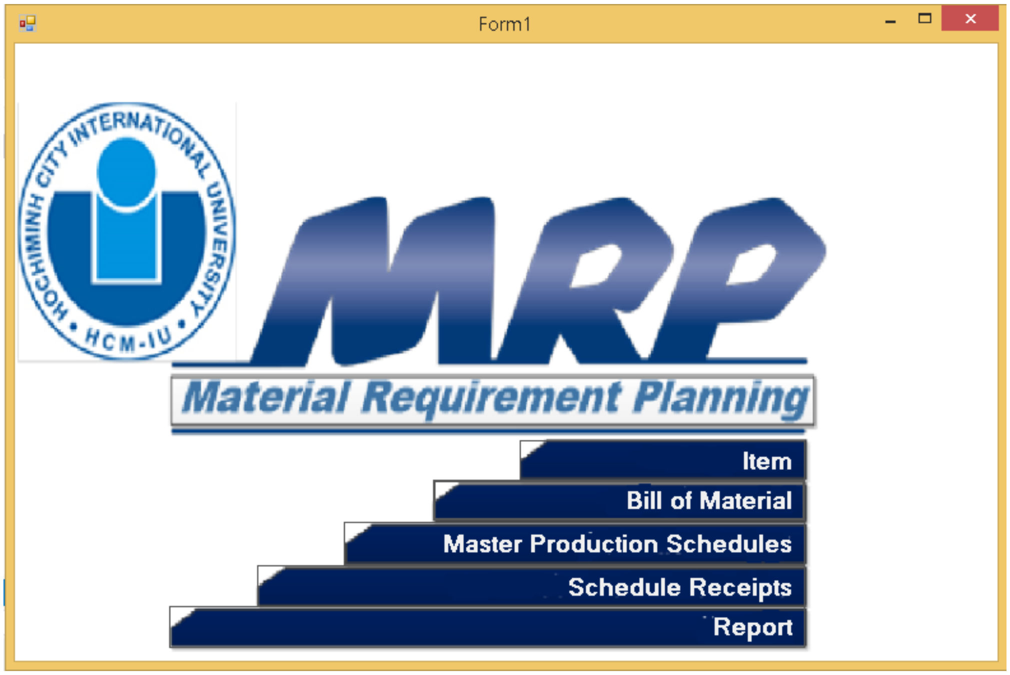
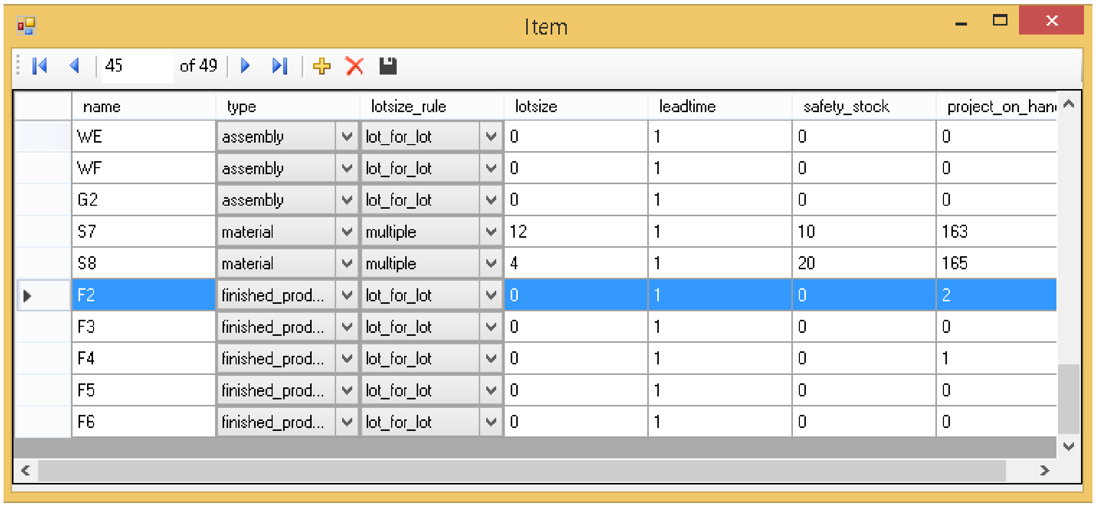

Form1
HOCHIMINH CITY INTERNATIONAL UNIVERSITY • HCM-IU •
MRP
Material Requirement Planning
Item
Bill of Material
Master Production Schedules
Schedule Receipts
Report

Figure 7: The main interface consists of item, bill of material, master production schedules, schedule receipts, report

4.3.1 Item

To start the program, first user must enter information for the item, because only when there is sufficient information item underneath the stage (bill of material, master production schedule receipts) will be performed.



	name	type	lotsize_rule	lotsize	leadtime	safety_stock	project_on_han
	WE	assembly	lot_for_lot	0	1	0	0
	WF	assembly	lot_for_lot	0	1	0	0
	G2	assembly	lot_for_lot	0	1	0	0
	S7	material	multiple	12	1	10	163
	S8	material	multiple	4	1	20	165
▶	F2	finished_prod...	lot_for_lot	0	1	0	2
	F3	finished_prod...	lot_for_lot	0	1	0	0
	F4	finished_prod...	lot_for_lot	0	1	0	1
	F5	finished_prod...	lot_for_lot	0	1	0	0
	F6	finished_prod...	lot_for_lot	0	1	0	0

Figure 8: Data entry window for the item

Item entitled "F2" is selected with the following information:

Type: finished product, lot size rule: lot for lot, lotsize: 0, leadtime: 1, safety stock: 0, project on hand: 2.

⏮ Return to the first item.

◀ Scroll to the item to be entered before (in this case "s8").

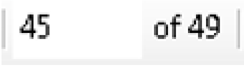
▶ Scroll to the item to be entered later (in this case is "f3").

⏭ Scroll to the last item entered (in this case is "f6").

✕ Remove an item is selected (in this case, the item "f2" in line 45 will be deleted).

💾 Save all data.

✚ Adding an item, after clicking this icon, a new blank line is added at the end of the table.

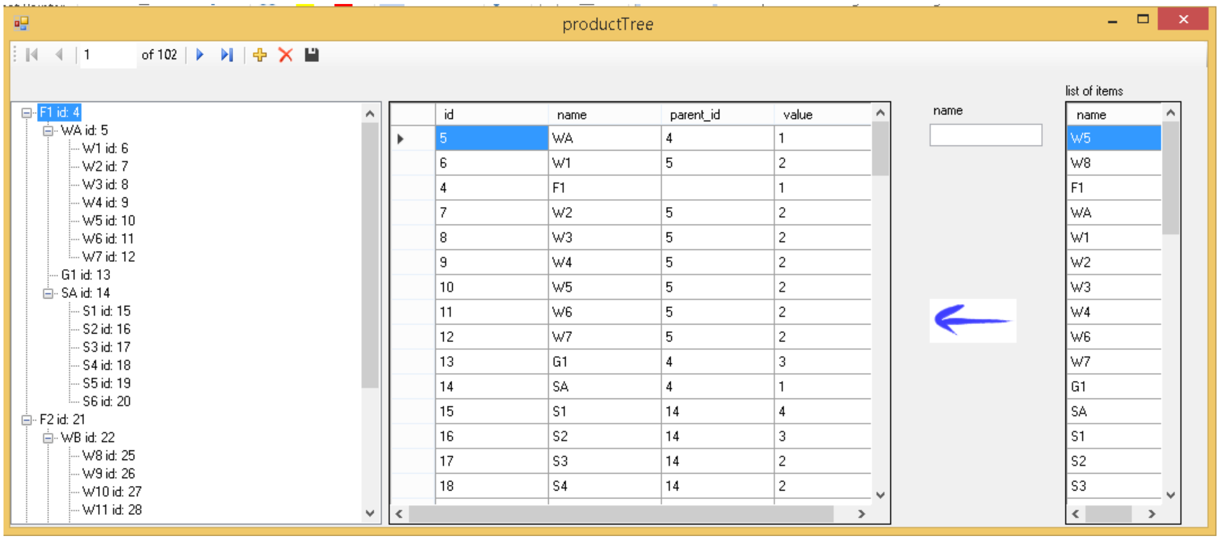

45 of 49

The right number show the total numbers of items have been entered, the left indicates the position selected line item (in this case a total of 49 items, 45 items in the line in the table is selected).

The process started with the new added field name, type, lot size rule, lot size, lead time, safety stock, projects on hand are absolutely no data .

Which name field enter the name of the item required; Required fields selected type of the 3 choices finished product, assembly and material; lot size rule compulsory school with a choice of 3 lot for lot, min and multiple; The remaining cases include lot size, lead time, safety stock, on hand required project data into an integer (type 0 for absence of data).

4.3.2. Bill of material



- F1 id: 4
  - WA id: 5
    - W1 id: 6
    - W2 id: 7
    - W3 id: 8
    - W4 id: 9
    - W5 id: 10
    - W6 id: 11
    - W7 id: 12
  - G1 id: 13
  - SA id: 14
    - S1 id: 15
    - S2 id: 16
    - S3 id: 17
    - S4 id: 18
    - S5 id: 19
    - S6 id: 20
- F2 id: 21
  - WB id: 22
    - W8 id: 25
    - W9 id: 26
    - W10 id: 27
    - W11 id: 28

id	name	parent_id	value
5	WA	4	1
6	W1	5	2
4	F1		1
7	W2	5	2
8	W3	5	2
9	W4	5	2
10	W5	5	2
11	W6	5	2
12	W7	5	2
13	G1	4	3
14	SA	4	1
15	S1	14	4
16	S2	14	3
17	S3	14	2
18	S4	14	2

list of items

name
W5
W8
F1
WA
W1
W2
W3
W4
W6
W7
G1
SA
S1
S2
S3

Figure 9: Enter data for bill of material

The left column is the outermost tree is represented by two components: the item name and id of the item on the tree.

Table representation for BOM data is the center table. In line with id = 5 is selected with item name is "WA", parent ID = 4, value = 1 corresponds to the node tree is our "WA id: 5" with the parent node id = 4, name = "F1 ". Which “id” represented each tree node (without any share node id); parent id parent said present value indicates the number of items. Root node separately (nodes without children) represent the final product has no parent id, in case the item is F1 with id = 4, value = 1, but the parent id not imported.

Rightmost column "list of items" is a list of how the item was entered in Item.

To carry out the new add an item to the tree diagram we conducted press ✚

At this point a new line is added, and simultaneously a new node identifier id on the tree is created, as a parent node id was created earlier (you can choose any value on id column), value is the amount of the merchandise? (Read the concept).

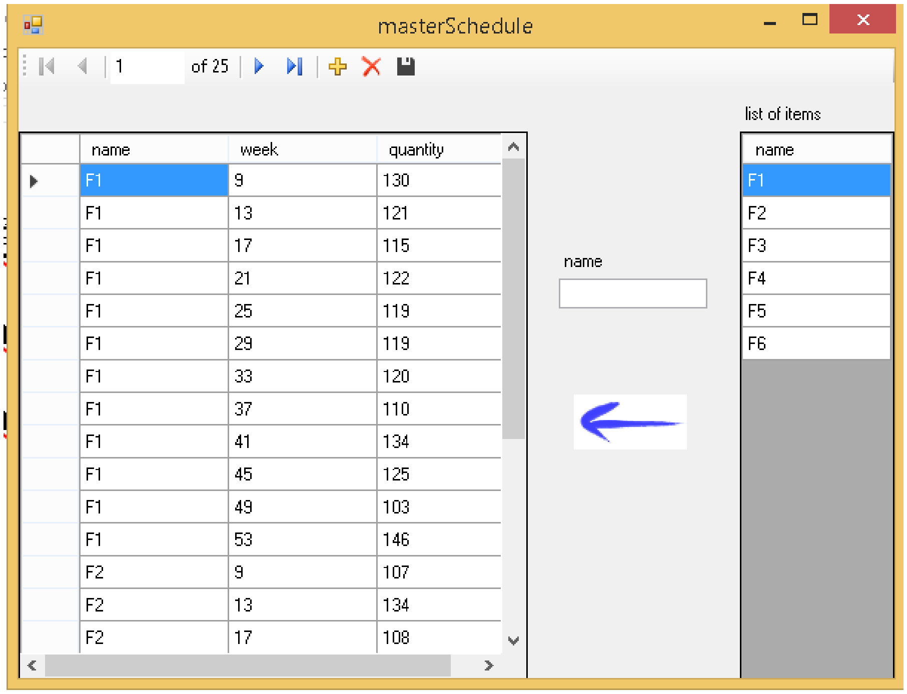
As for the value name, do not be entered directly by right clicking on the column select the corresponding name "list of items".

The "list of items" rightmost display a list of items you can select item name in data entry process to build the tree structure table.

O text box "name" to implement a filter to the column name to extract the items needed for data entry. X also showed comparable item name in the "list of items" the text box "name" contains the value "w5".

Press 💾 to save.

4.3.3. Master production schedule

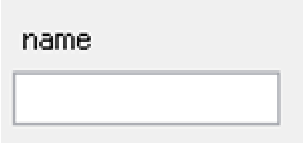


name	week	quantity
F1	9	130
F1	13	121
F1	17	115
F1	21	122
F1	25	119
F1	29	119
F1	33	120
F1	37	110
F1	41	134
F1	45	125
F1	49	103
F1	53	146
F2	9	107
F2	13	134
F2	17	108

name
F1
F2
F3
F4
F5
F6

Figure 10: Production schedule

Column list of items to display the item names were entered in the item type is "finished product". In this case we have (f1, f2, f3, f4, f5, f6).

Input frame “name” 

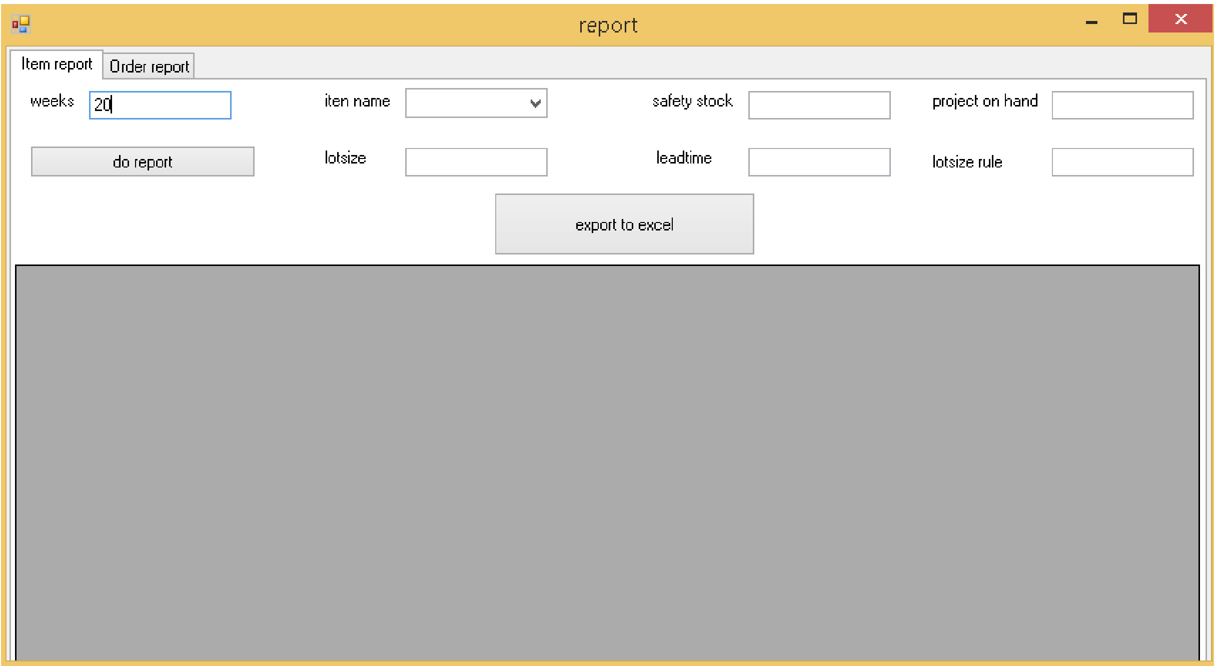
allowing search item for the column name "list of items".

To add a new master schedule click ✚ it is a new line as shown below.

Enter integer value for the week and quantity, with its own name column we noted not be entered directly on the item name, which must click on the corresponding item on the column name list of items (each item will be the program hit a service id number calculation process, so should click on the column item "list of items" to obtain the id).

Ending the process of entering data one click 💾

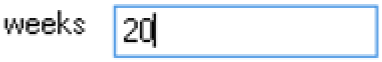
4.3.4. Report



weeks 20

Figure 11: Item report

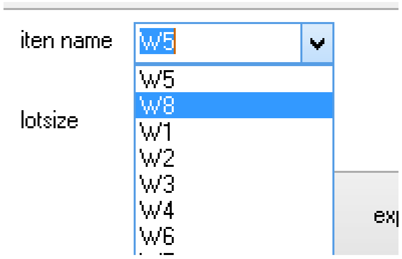
Enter the number of weeks needed to calculate the scheduled week in text input



weeks 20

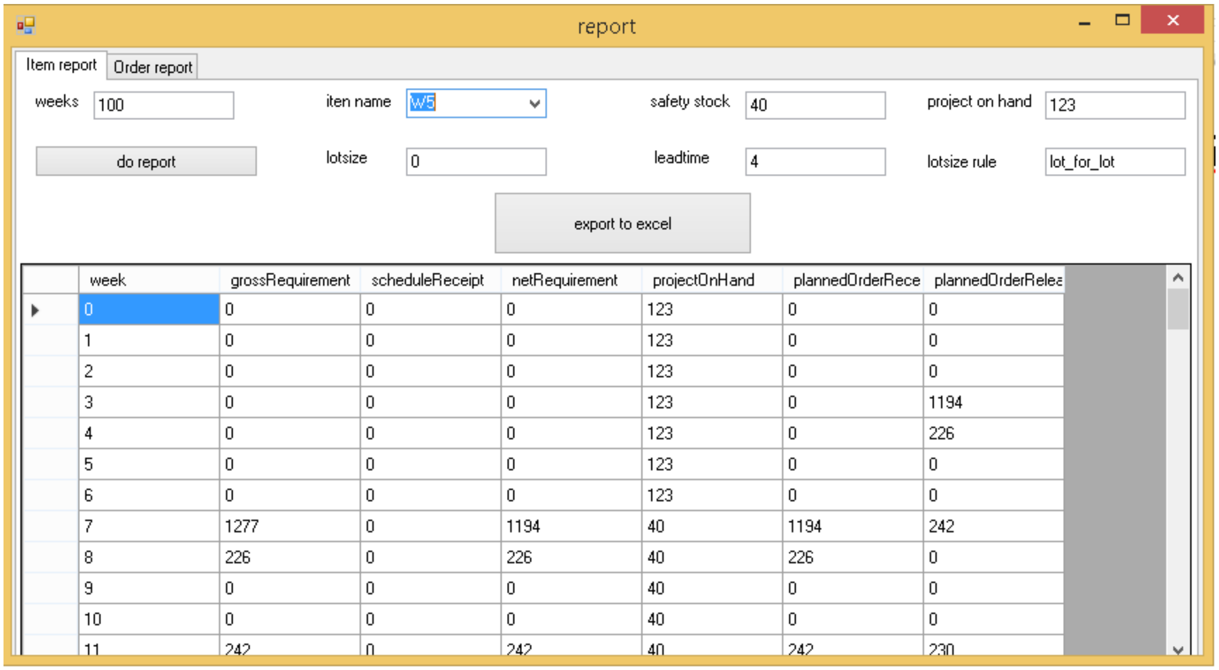
Then select the "do report" perform calculations.

When the calculation is complete you will see combo box "item name" appears to us names of selected items.



iten name W5

When an item is selected from combo box item name, the text box at lot size, safety stock, lead time, projected on hand, lot size rules and schedule will be present for the corresponding item.

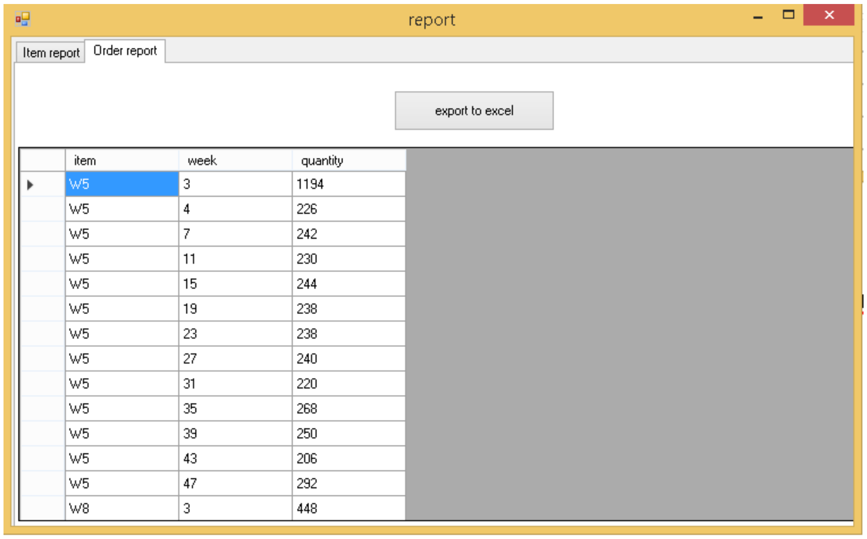


weeks 100
iten name W5
safety stock 40
project on hand 123
lotsize 0
leadtime 4
lotsize rule lot_for_lot

week	grossRequirement	scheduleReceipt	netRequirement	projectOnHand	plannedOrderRece	plannedOrderRelea
0	0	0	0	123	0	0
1	0	0	0	123	0	0
2	0	0	0	123	0	0
3	0	0	0	123	0	1194
4	0	0	0	123	0	226
5	0	0	0	123	0	0
6	0	0	0	123	0	0
7	1277	0	1194	40	1194	242
8	226	0	226	40	226	0
9	0	0	0	40	0	0
10	0	0	0	40	0	0
11	242	0	242	40	242	230

Figure 12: Item report for item "W5"

Click "export to Excel" spreadsheet export schedule to share a folder excel file with executable program file with the name of the item.



item	week	quantity
W5	3	1194
W5	4	226
W5	7	242
W5	11	230
W5	15	244
W5	19	238
W5	23	238
W5	27	240
W5	31	220
W5	35	268
W5	39	250
W5	43	206
W5	47	292
W8	3	448

Figure 13: Order report

Order the panel report results Order (quantity) per week for each specific item.

The "export to excel" save excel table to share the file path running program.

Chapter 5 PROGRAM RESULT

The program will export two kinds of report into Excel: Item report and order report

5.1. Item report

Item report includes information about Gross requirement, schedule receipt, net requirement, project on hand, planed order receipt and planed order release of the chosen item.

Below is the report of item “W5” the item most needed in the production line.

Table 6: Report of item "W5" in 6 month

week	Gross Requirement	Schedule Receipt	Net Requirement	Project On Hand	Planned Order Receipt	Planned Order Release
1	0	0	0	123	0	0
2	0	0	0	123	0	0
3	0	0	0	123	0	1194
4	0	0	0	123	0	226
5	0	0	0	123	0	0
6	0	0	0	123	0	0
7	1277	0	1194	40	1194	1290
8	226	0	226	40	226	218
9	0	0	0	40	0	0
10	0	0	0	40	0	0
11	1290	0	1290	40	1290	1292
12	218	0	218	40	218	234
13	0	0	0	40	0	0
14	0	0	0	40	0	0
15	1292	0	1292	40	1292	1280
16	234	0	234	40	234	236
17	0	0	0	40	0	0

18	0	0	0	40	0	0
19	1280	0	1280	40	1280	1269
20	236	0	236	40	236	236
21	0	0	0	40	0	0
22	0	0	0	40	0	0
23	1269	0	1269	40	1269	1121
24	236	0	236	40	236	288

5.2. Order report

Order report shows user the time and the quantity of each item that should order. Order reports of all items will be showed and export to excel.

Below is the order report of in item W5

Table 7: Order report of item "W5"

item	week	quantity
W5	3	1194
W5	4	226
W5	7	1290
W5	8	218
W5	11	1292
W5	12	234
W5	15	1280
W5	16	236
W5	19	1269
W5	20	236
W5	23	1121
W5	24	288
W5	27	1309
W5	28	282
W5	31	1274

W5	32	250
W5	35	1210
W5	36	230
W5	39	1251
W5	40	214
W5	43	1210
W5	44	298
W5	47	1387
W5	48	216

5.3. Inventory improvement

Table 8: Inventory (6 month) improvement

Item	Inventory before apply MRP	Inventory after apply MRP	Difference
W5	123	40	83
W8	154	40	114
W1	121	40	81
W2	205	40	165
W3	174	40	134
W4	126	40	86
W6	123	40	83
W7	136	40	96
S1	150	22	128
S2	126	33	93
S3	108	21	87
S4	101	20	81
S5	187	32	155
W9	136	40	96

W10	125	40	85
W11	176	40	136
W12	104	40	64
W13	127	40	87
W14	116	40	76
W15	199	40	159
W16	140	40	100
W17	190	40	150
W18	144	40	104
W19	137	40	97
W20	130	40	90
W21	147	40	107
S7	163	12	151
S8	165	4	161
		Total	3049

Applying MRP into manufacturing process make an impact on the inventory, in 6 month, the inventory of the company has decreases an amount of 3049 unit. On the other hand, this program provides a purchasing and delivering schedule so that the company can maintain the inventory as low as possible to save cost.

Chapter 6 CONCLUSION AND RECOMMENDATION

6.1. Conclusion

This study: “MATERIAL REQUIRMENT PLANNING: Case study in Ltd. Toan Bich” describe a scientific tool the help the company to make order decision, meet the deadline on time and control the capacity and reduce the production cost.

This study apply knowledge about production management, inventory management and programming to built this MRP program that can improve the utilization of inventory and reducing cost.

After MRP was apply, it is show that new schedule built by the program reduce the cost for inventory, control delivery and purchasing activities, maintain availability of material for production line.

In a brief case, MRP is a useful tool for controlling inventory and available material; this method also provides a schedule for purchasing and delivering for Toan Bich.

6.2. Recommendation

The inventory of the company is cost a lot of money because they do not have any system to control. To repair this problem, the company should apply some tool in the production line, such as MRP, MRP II, ERP. These tool are famous and have been using worldwide. It is necessary to apply the tool in manufacturing especially for market expansion.

For further study, applying MRP for all products of Toan Bich is recommended. On the other hand, MRP has some weakness such as only be used in short-term, these weakness could be fixed by applying ERP or MRP II.

REFERENCE

1. Lambrecht, M.R., Muckstadt, J. A., and Luyten, R., *Protective Stock in Multi-Stage Production Systems*, International Journal of Production Research, Vol.22, pp.1001-1025 (1984).
2. Mahima Singh, *Study of literature on material requirement plas,* Vol.1, pp1-2 (2014).
3. Karaesmen, F., Liberopoulos, G., and Dallery, Y., *Production/ Inventory Control with Advance Demand In- Formation*, International Series in Operations Research and Man- agementScience, Kluwer Acadamic Publisher, Dordrecht, The Netherland, pp. 243-270 (2003)
4. Karaesmen, F., Buzacott, J. A., and Dalery, Y., *Integrating Advance Order Information in Production Control*, IIE Transaction s, Vol. 34, pp. 649-662 (2002)
5. Yolanda Masnita, Tania Anada Madani, *Factor of MRP Implementation in Manufacturer for Small and Medium-sized Firms*, the 2012 International Conference on Business and Management, 2012.
6. Robert C. Leachman, Jose F. Goncalves, *Rate-Based MRP*.
7. Orlicky, Joseph, 1975. *Material Requirements Planning*, McGraw-Hill, NewYork.
8. Edmund W. Schuster, Stuart J. Allen and Michael P. D'Itri, *Capacitated Materials Requirements Planning and its Application in the Process Industries.*
9. R. B. Cooper, R. W. Zmud, *Material Requirements Planning System Infusion,* OMEGA Int. J of Mgmt Scl., Vol. 17, No. 5, pp. 471-481 (1989).
10. Nahmias, S., *"Production and Operations Analysis with Student CD-Rom."* 4th Ed., McGraw-Hill/Irwin, 2000.
11. Bill of Materials (BoM) Definition. [Online] Available: http://searchmanufacturingerp.techtarget.com/definition/bill-of-materials-BoM
12. About Toan Bich. 2015. [Online] Available: http://tobi.com.vn
13. Microsoft Visual Studio. 2014. [Online] Available: https://vi.wikipedia.org/wiki/Microsoft_Visual_Studio.

14. Just-in-Time Manufacturing. 2015 [Online] Available: https://en.wikipedia.org/wiki/Just-in-Time_Manufacturing

15. Material requirement planing (MRP). [Online] Available: http://www.uoguelph.ca/~dsparlin/mrp.htm

APPENDIX

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Data.SqlServerCe;
using IEIEIU11006.MRPprocess;

namespace IEIEIU11006
{
    public partial class Item : Form
    {
        public Item()
        {
            InitializeComponent();
        }

        private void itemDBBindingNavigatorSaveItem_Click(object sender, 
EventArgs e)
        {
            //var t = itemDBBindingSource.Find("id", 2);
            //var customerList = 
(List<itemData>)this.itemDBBindingSource.DataSource;
            try
            {
                this.Validate();
                this.itemDBBindingSource.EndEdit();
                this.tableAdapterManager.UpdateAll(this.dbDataSet);
            }
            catch (System.Exception ex)
            {
                MessageBox.Show("Update failed");
            }
            //string sFilter = string.Empty;

          //  var h = itemDBBindingSource.Find("id", 2);
            //DataRowView demo = itemDBBindingSource.Position(h);
            //DataRowView temp = 
(DataRowView)itemDBBindingSource.Current;
            //temp["name"] = "aaaaaaa";
           // var ggg = 
((DataRowView)itemDBBindingSource.Current).Row.Item["id"];
           //foreach (DataRowView row in itemDBBindingSource.List) {
```

```
                    /*    if (sFilter != string.Empty){

                            sFilter += " OR ";}

        sFilter += "ClubID = " + row["ClubID"];*/

    //}

           //itemDBBindingSource.ResetItem(0);
           //this.itemDBBindingSource.EndEdit();
           //this.tableAdapterManager.UpdateAll(this.dbDataSet);
        }


        private void Item_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'dbDataSet.itemDB' table. You can move, or remove it, as needed.
            this.itemDBTableAdapter.Fill(this.dbDataSet.itemDB);

        }

        private void toolStripComboBox1_Click(object sender, EventArgs e)
        {

        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IEIEIU11006
{
    public partial class masterSchedule : Form
    {
        public masterSchedule()
        {
            InitializeComponent();
        }

        private void masterDBBindingNavigatorSaveItem_Click(object
sender, EventArgs e)
```

```
        {
            try
            {
                 DataRowView temp =
(DataRowView)this.masterDBBindingSource.Current;

                 if (String.IsNullOrEmpty(temp["id_item"].ToString()) ==
false)
                 {
                    this.Validate();
                    this.masterDBBindingSource.EndEdit();
                    this.tableAdapterManager.UpdateAll(this.dbDataSet);
                 }
                 else
                 {
                     MessageBox.Show("Please choose a name!");
                 }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show("Update failed");
            }

        }

        private void masterSchedule_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'dbDataSet.itemDB' table. You can move, or remove it, as needed.
            this.itemDBTableAdapter.Fill(this.dbDataSet.itemDB);
            // TODO: This line of code loads data into the
'dbDataSet.masterDB' table. You can move, or remove it, as needed.
            this.masterDBTableAdapter.Fill(this.dbDataSet.masterDB);

            //show only master product
            this.itemDBBindingSource.Filter = string.Format("type LIKE
'finished_product'");
        }

        private void itemDBDataGridView_CellClick(object sender,
DataGridViewCellEventArgs e)
        {
            if (this.masterDBBindingSource.Count > 0)
            {
                DataRowView temp1 =
(DataRowView)itemDBBindingSource.Current;
                DataRowView temp2 =
(DataRowView)masterDBBindingSource.Current;
```

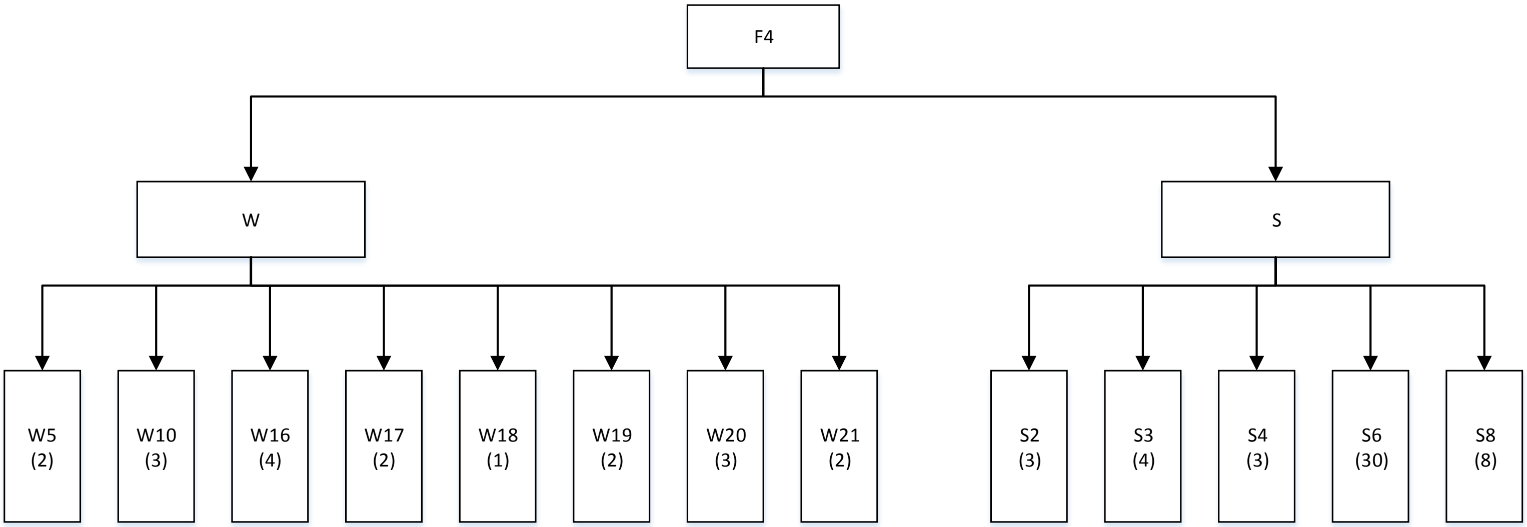
```
                this.textNameSearchMasterSchedule.Text = 
temp1["name"].ToString();
                temp2["name"] = temp1["name"].ToString();
                temp2["id_item"] = int.Parse(temp1["id"].ToString());

this.masterDBBindingSource.ResetItem(this.masterDBBindingSource.Position
);
            }
        }

        private void textNameSearchMasterSchedule_TextChanged(object 
sender, EventArgs e)
        {
            this.itemDBBindingSource.Filter = string.Format("name LIKE 
'" + this.textNameSearchMasterSchedule.Text.ToString() + "%' AND type 
LIKE 'finished_product'");
            this.itemDBDataGridView.Update();
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IEIEIU11006
{
    public partial class MRPform : Form
    {
        public MRPform()
        {
            InitializeComponent();
        }

        private void MRPform_Load(object sender, EventArgs e)
        {
            btnItem.TabStop = false;
            this.btnItem.FlatStyle = FlatStyle.Popup;
            this.btnItem.FlatAppearance.BorderSize = 0;
            this.btnItem.TextAlign = ContentAlignment.MiddleRight;

            this.btn_billOfMaterial.TextAlign = 
ContentAlignment.MiddleRight;
            this.btnMasterProductionSchedule.TextAlign = 
ContentAlignment.MiddleRight;
```

```
            this.btnScheduleReceipt.TextAlign =
ContentAlignment.MiddleRight;
            this.btnReport.TextAlign = ContentAlignment.MiddleRight;
        }

        private void btnItem_Click(object sender, EventArgs e)
        {
            Item itemForm = new Item();
            itemForm.Show();
        }

        private void btn_billOfMaterial_Click(object sender, EventArgs e)
        {
            productTree productTreeForm = new productTree();
            productTreeForm.Show();
        }

        private void btnMasterProductionSchedule_Click(object sender,
EventArgs e)
        {
            masterSchedule masterScheduleForm = new masterSchedule();
            masterScheduleForm.Show();
        }

        private void btnScheduleReceipt_Click(object sender, EventArgs e)
        {
            scheduleReceipt scheduleReceiptForm = new scheduleReceipt();
            scheduleReceiptForm.Show();
        }

        private void btnReport_Click(object sender, EventArgs e)
        {
            report reportForm = new report();
            reportForm.Show();
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IEIEIU11006
{
    public partial class productTree : Form
    {
        public productTree()
        {
            InitializeComponent();
        }

        private void productTree_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the
'dbDataSet.itemDB' table. You can move, or remove it, as needed.
            this.itemDBTableAdapter.Fill(this.dbDataSet.itemDB);
            // TODO: This line of code loads data into the
'dbDataSet.bomDB' table. You can move, or remove it, as needed.
            this.bomDBTableAdapter.Fill(this.dbDataSet.bomDB);

            //load tree
            this.reloadTreeBom();
        }

        private void bomDBBindingNavigatorSaveItem_Click(object sender,
EventArgs e)
        {
            try
            {
                 DataRowView temp =
(DataRowView)bomDBBindingSource.Current;

                 if (String.IsNullOrEmpty(temp["id_item"].ToString()) ==
false)
                 {
                     this.Validate();
                     this.bomDBBindingSource.EndEdit();
                     this.tableAdapterManager.UpdateAll(this.dbDataSet);

                     //reload tree
                     this.reloadTreeBom();
                 }
                 else
                 {
                     MessageBox.Show("Please choose a name!");
                 }
            }
             catch (System.Exception ex)
             {
                 MessageBox.Show("Update failed");
             }

        }

        private void textBox1_TextChanged(object sender, EventArgs e)
        {
            this.itemDBBindingSource.Filter = string.Format("name LIKE 
'"+this.txtSearchName_productTree.Text.ToString()+"%'");
            this.itemDBDataGridView.Update();
        }
        private void itemDBDataGridView_CellClick(object sender, 
DataGridViewCellEventArgs e)
        {
            if (this.bomDBBindingSource.Count > 0)
            {
                DataRowView temp1 = 
(DataRowView)itemDBBindingSource.Current;
                DataRowView temp2 = 
(DataRowView)bomDBBindingSource.Current;

                this.txtSearchName_productTree.Text = 
temp1["name"].ToString();
                temp2["name"] = temp1["name"].ToString();
                temp2["id_item"] = int.Parse(temp1["id"].ToString());
                
this.bomDBBindingSource.ResetItem(this.bomDBBindingSource.Position);
            }
        }

        private void reloadTreeBom()
        {
            this.treeViewBom.Nodes.Clear();

            foreach (DataRowView cr in bomDBBindingSource)
            {
                if (string.IsNullOrEmpty(cr["parent_id"].ToString()) == 
true)
                {
                    TreeNode parentTreeNode = new TreeNode();
                    parentTreeNode.Text = cr["name"].ToString() + " " + 
"id: " + cr["id"].ToString();

                    this.getChildRow(int.Parse(cr["id"].ToString()), 
parentTreeNode);
                    this.treeViewBom.Nodes.Add(parentTreeNode);
                }
            }
        }

        private void getChildRow(int parentID, TreeNode node)
        {
            foreach (DataRowView cr in bomDBBindingSource)
            {
                if (string.IsNullOrEmpty(cr["parent_id"].ToString()) == 
false)
```

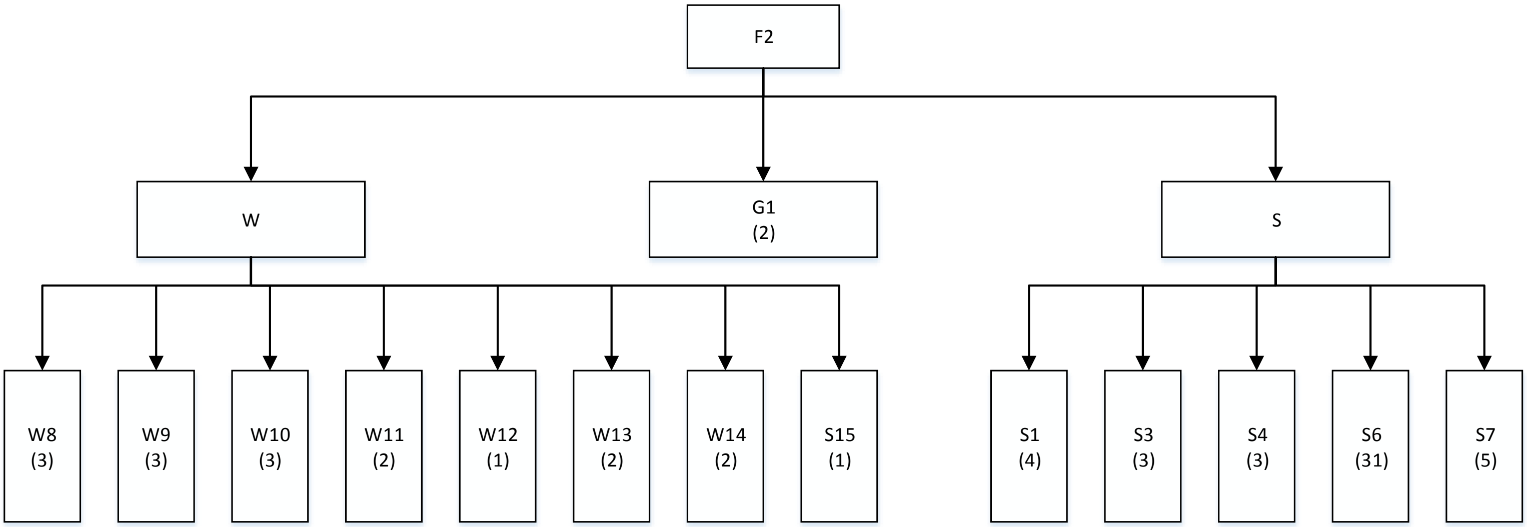
```
                {
                    if (int.Parse(cr["parent_id"].ToString()) ==
parentID)
                    {
                        TreeNode childNode = new TreeNode();
                        childNode.Text = cr["name"].ToString() + " " +
"id: " + cr["id"].ToString();

                        this.getChildRow(int.Parse(cr["id"].ToString()),
childNode);
                        node.Nodes.Add(childNode);
                    }
                }
            }
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using IEIEIU11006.MRPprocess;
using Microsoft.Office.Interop.Excel;
using exportExcel = Microsoft.Office.Interop.Excel.Application;

namespace IEIEIU11006
{
    public partial class report : Form
    {

        private List<itemData> itemList;
        private List<itemData> materialList;
        private List<bomData> bomList;
        private List<masterData> masterList;
        private List<receiptData> receiptList;

        private int duration = 20;

        private Dictionary<int, List<MRPdata>> itemReport = new
Dictionary<int, List<MRPdata>>();
        private List<reportOrderData> OrderReport = new
List<reportOrderData>();

        private void initReportData()
        {
            for (int i = 0; i < this.itemList.Count; ++i)
            {
                int itemID = this.itemList[i].id;

                if (this.itemList[i].type == "material")
                {
                    if (this.itemReport.ContainsKey(itemID) == false)
                    {
                        List<MRPdata> reportData = new List<MRPdata>();
                        for (int j = 0; j < this.duration; ++j)
                        {
                            reportData.Add(new MRPdata());
                            reportData[j].week = j;
                        }

                        this.itemReport.Add(itemID, reportData);
                    }
                }
            }
        }

        private void initScheduleReceipt()
        {
            for (int i = 0; i < this.receiptList.Count; ++i)
            {
                int itemID = this.receiptList[i].id_item;
                int week = this.receiptList[i].week;
                int quantity = this.receiptList[i].quantity;

                if (this.itemReport.ContainsKey(itemID) == true)
                {
                    if (week >= 0 && week < this.duration)
                    {
                        this.itemReport[itemID][week].scheduleReceipt +=
quantity;
                    }
                }
            }
        }

        private void initGrossRequirement(int index, int sLeadtime, int
quantity)
        {
            for (int i = 0; i < this.bomList.Count; ++i)
            {
                int itemID = this.bomList[i].id_item;

                if (this.bomList[i].parent_id == index)
                {
                    float value = this.bomList[i].value;
                    int leadtime = this.getItem(itemID).leadtime;

                    if (this.itemReport.ContainsKey(itemID) == true)
                    {

this.itemReport[itemID][sLeadtime].grossRequirement +=
(int)(Math.Ceiling(value * (double)quantity));
                    }

                    this.initGrossRequirement(this.bomList[i].id,
sLeadtime-leadtime, quantity);
                }
            }
        }

        private void initMasterSchedule()
        {
            for (int i = 0; i < this.masterList.Count; ++i)
            {
                for (int j = 0; j < this.bomList.Count; ++j)
                {
                    if (this.masterList[i].id_item ==
this.bomList[j].id_item)
                    {
                        int index = this.bomList[j].id;
                        int leadtime =
this.getItem(this.masterList[i].id_item).leadtime;
                        int quantity = this.masterList[i].quantity;

                        this.initGrossRequirement(index,
this.masterList[i].week-leadtime, quantity);
                    }
                }
            }
        }

        private itemData getItem(int itemID)
        {
            int index = itemDBBindingSource.Find("id", itemID);
            itemData tmp = new itemData();

            tmp.id = itemList[index].id;
            tmp.name = itemList[index].name;
            tmp.type = itemList[index].type;
            tmp.lotsize_rule = itemList[index].lotsize_rule;
            tmp.lotsize = itemList[index].lotsize;
            tmp.leadtime = itemList[index].leadtime;
            tmp.safety_stock = itemList[index].safety_stock;
            tmp.project_on_hand = itemList[index].project_on_hand;
            return tmp;
        }

        private void getDataFormTable()
        {
            this.itemList = new List<itemData>();
            this.materialList = new List<itemData>();
            this.bomList = new List<bomData>();
            this.masterList = new List<masterData>();
            this.receiptList = new List<receiptData>();

            //item table
            foreach (DataRowView row in this.itemDBBindingSource.List)
            {
                itemData tmp = new itemData();

                tmp.id = int.Parse(row["id"].ToString());
                tmp.name = row["name"].ToString();
                tmp.type = row["type"].ToString();
                tmp.lotsize_rule = row["lotsize_rule"].ToString();
                tmp.lotsize = int.Parse(row["lotsize"].ToString());
                tmp.leadtime = int.Parse(row["leadtime"].ToString());
                tmp.safety_stock = 
int.Parse(row["safety_stock"].ToString());
                tmp.project_on_hand = 
int.Parse(row["project_on_hand"].ToString());

                this.itemList.Add(tmp);
            }

            //get materials
            foreach (DataRowView row in this.itemDBBindingSource.List)
            {
                itemData tmp = new itemData();

                tmp.id = int.Parse(row["id"].ToString());
                tmp.name = row["name"].ToString();
                tmp.type = row["type"].ToString();
                tmp.lotsize_rule = row["lotsize_rule"].ToString();
                tmp.lotsize = int.Parse(row["lotsize"].ToString());
                tmp.leadtime = int.Parse(row["leadtime"].ToString());
                tmp.safety_stock = 
int.Parse(row["safety_stock"].ToString());
                tmp.project_on_hand = 
int.Parse(row["project_on_hand"].ToString());

                if (tmp.type == "material")
                {
                    this.materialList.Add(tmp);
                }
            }

            //bom table
            foreach (DataRowView row in this.bomDBBindingSource.List)
            {
                bomData tmp = new bomData();

                tmp.id = int.Parse(row["id"].ToString());
                tmp.id_item = int.Parse(row["id_item"].ToString());
                tmp.name = row["name"].ToString();
                tmp.value = float.Parse(row["value"].ToString());

                if (string.IsNullOrEmpty(row["parent_id"].ToString()) ==
false)
                {
                    tmp.parent_id =
int.Parse(row["parent_id"].ToString());
                }
                else
                {
                    tmp.parent_id = -1;
                }


                this.bomList.Add(tmp);
            }

            //master table
            foreach (DataRowView row in this.masterDBBindingSource.List)
            {
                masterData tmp = new masterData();

                tmp.id = int.Parse(row["id"].ToString());
                tmp.id_item = int.Parse(row["id_item"].ToString());
                tmp.name = row["name"].ToString();
                tmp.week = int.Parse(row["week"].ToString());
                tmp.quantity = int.Parse(row["quantity"].ToString());

                this.masterList.Add(tmp);
            }

            //receipt table
            foreach (DataRowView row in this.receiptDBBindingSource.List)
            {
                receiptData tmp = new receiptData();

                tmp.id = int.Parse(row["id"].ToString());
                tmp.id_item = int.Parse(row["id_item"].ToString());
                tmp.name = row["name"].ToString();
                tmp.week = int.Parse(row["week"].ToString());
                tmp.quantity = int.Parse(row["quantity"].ToString());

                this.receiptList.Add(tmp);
            }
        }

        public report()
        {
            InitializeComponent();
        }

        private void doL4L(int itemID)
        {
            int safetyStock = this.getItem(itemID).safety_stock;
            int leadTime = this.getItem(itemID).leadtime;
            int projectOnHand = this.getItem(itemID).project_on_hand;

            for (int i = 0; i < itemReport[itemID].Count; ++i)
            {
                itemReport[itemID][i].netRequirement = 
itemReport[itemID][i].grossRequirement
                                                        - 
itemReport[itemID][i].scheduleReceipt
                                                        -(projectOnHand-
safetyStock);

                itemReport[itemID][i].plannedOrderReceipt = 0;

                if (itemReport[itemID][i].netRequirement > 0)
                {
                    itemReport[itemID][i].plannedOrderReceipt = 
itemReport[itemID][i].netRequirement;

                    if (i >= leadTime)
                    {
                        itemReport[itemID][i-
leadTime].plannedOrderRelease = 
itemReport[itemID][i].plannedOrderReceipt;
                    }
                }
                else
                {
                    itemReport[itemID][i].netRequirement = 0;
                }

                projectOnHand = itemReport[itemID][i].scheduleReceipt

+itemReport[itemID][i].plannedOrderReceipt
                                +projectOnHand
                                -itemReport[itemID][i].grossRequirement;
                itemReport[itemID][i].projectOnHand=projectOnHand;
            }
        }

        private void doMULT(int itemID)
        {
            int lostSize = this.getItem(itemID).lotsize;
            int safetyStock = this.getItem(itemID).safety_stock;
            int leadTime = this.getItem(itemID).leadtime;
            int projectOnHand = this.getItem(itemID).project_on_hand;

            for (int i = 0; i < itemReport[itemID].Count; ++i)
            {
                itemReport[itemID][i].netRequirement =
itemReport[itemID][i].grossRequirement
                                                        -
itemReport[itemID][i].scheduleReceipt
                                                        - (projectOnHand
- safetyStock);

                itemReport[itemID][i].plannedOrderReceipt = 0;

                if (itemReport[itemID][i].netRequirement > 0)
                {
                    itemReport[itemID][i].plannedOrderReceipt =
(int)(Math.Ceiling((double)itemReport[itemID][i].netRequirement /
(double)lostSize) * lostSize);

                    if (i >= leadTime)
                    {
                        itemReport[itemID][i -
leadTime].plannedOrderRelease =
itemReport[itemID][i].plannedOrderReceipt;
                    }
                }
                else
                {
                    itemReport[itemID][i].netRequirement = 0;
                }

                projectOnHand = itemReport[itemID][i].scheduleReceipt
                                +
itemReport[itemID][i].plannedOrderReceipt
                                + projectOnHand
                                - itemReport[itemID][i].grossRequirement;

                itemReport[itemID][i].projectOnHand = projectOnHand;
            }
       }
```

```
        private void report_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 
'dbDataSet.receiptDB' table. You can move, or remove it, as needed.
            this.receiptDBTableAdapter.Fill(this.dbDataSet.receiptDB);
            // TODO: This line of code loads data into the 
'dbDataSet.masterDB' table. You can move, or remove it, as needed.
            this.masterDBTableAdapter.Fill(this.dbDataSet.masterDB);
            // TODO: This line of code loads data into the 
'dbDataSet.itemDB' table. You can move, or remove it, as needed.
            this.itemDBTableAdapter.Fill(this.dbDataSet.itemDB);
            // TODO: This line of code loads data into the 
'dbDataSet.bomDB' table. You can move, or remove it, as needed.
            this.bomDBTableAdapter.Fill(this.dbDataSet.bomDB);

            //set weeks default
            this.txtReportWeek.Text = "20";
        }

        private void btnDoReport_Click(object sender, EventArgs e)
        {
            this.duration = 
int.Parse(this.txtReportWeek.Text.ToString());

            //convert table to list
            this.getDataFormTable();
            //create data report
            this.initReportData();
            //create schedule receipt
            this.initScheduleReceipt();
            //create schedule for master product
            this.initMasterSchedule();

            //get data for combobox name
            this.comboBoxItemName.DataSource = this.materialList;
            this.comboBoxItemName.DisplayMember = "name";

            foreach (var pair in itemReport)
            {
                if (this.getItem(pair.Key).lotsize_rule == "lot_for_lot")
                {
                    this.doL4L(pair.Key);
                }
                else
                {
                    this.doMULT(pair.Key);
                }
            }
            this.OrderReport = new List<reportOrderData>();
            foreach (var pair in itemReport)
            {
                List<MRPdata> tmpReport = pair.Value;
                for (int i = 0; i < tmpReport.Count; ++i)
                {
                    if (tmpReport[i].plannedOrderRelease > 0)
                    {
                        reportOrderData tmpOrderReport = new
reportOrderData();

                        tmpOrderReport.week = i;
                        tmpOrderReport.quantity =
tmpReport[i].plannedOrderRelease;
                        tmpOrderReport.item =
this.getItem(pair.Key).name.ToString();

                        this.OrderReport.Add(tmpOrderReport);
                    }
                }
            }

            this.dataGridViewOrderReport.DataSource = this.OrderReport;
        }

        private void exportReport2Excel(DataGridView g, string path,
string file)
        {
            exportExcel obj = new exportExcel();
            obj.Application.Workbooks.Add(Type.Missing);
            obj.Columns.ColumnWidth = 25;

            for (int i = 0; i < g.Columns.Count; ++i)
            {
                obj.Cells[1, i + 1] = g.Columns[i].HeaderText;
            }

            for (int i = 0; i < g.Rows.Count; ++i)
            {
                for (int j = 0; j < g.Columns.Count; ++j)
                {
                    obj.Cells[i + 2, j + 1] =
g.Rows[i].Cells[j].Value.ToString();
                }
            }

            obj.ActiveWorkbook.SaveCopyAs(path + file + ".xlsx");
        }
        private void comboBoxItemName_SelectedIndexChanged(object sender,
EventArgs e)
        {
            itemData tmp = (itemData)this.comboBoxItemName.SelectedItem;

            if (tmp != null)
            {
                this.txtLotsize.Text = tmp.lotsize.ToString();
                this.txtSafetyStock.Text = tmp.safety_stock.ToString();
                this.txtLeadtime.Text = tmp.leadtime.ToString();
                this.txtProjectOnHand.Text =
tmp.project_on_hand.ToString();
                this.txtLotsizeRuleReport.Text =
tmp.lotsize_rule.ToString();

                int tmpID = int.Parse(tmp.id.ToString());

                if (this.itemReport.ContainsKey(tmpID))
                {
                    this.dataGridViewItemReport.DataSource =
itemReport[tmpID];
                }
            }
        }

        private void btmItemReportToExcel_Click(object sender, EventArgs
e)
        {
            itemData tmp = (itemData)this.comboBoxItemName.SelectedItem;

            if (tmp != null)
            {
                string tmpName = tmp.name.ToString();
                int tmpID = int.Parse(tmp.id.ToString());

                if (this.itemReport.ContainsKey(tmpID))
                {
                    string startupPath =
System.IO.Path.GetFullPath(".\\");
                    this.exportReport2Excel(this.dataGridViewItemReport,
startupPath, tmpName);
                }
            }
        }

        private void btnExportOrderReport_Click(object sender, EventArgs
e)
        {
            string tmpName = "order_report";
            string startupPath = System.IO.Path.GetFullPath(".\\");
            this.exportReport2Excel(this.dataGridViewOrderReport,
startupPath, tmpName);
        }
    }
}
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace IEIEIU11006
{
    public partial class scheduleReceipt : Form
    {
        public scheduleReceipt()
        {
            InitializeComponent();
        }

        private void receiptDBBindingNavigatorSaveItem_Click(object
sender, EventArgs e)
        {
            try
            {
                DataRowView temp =
(DataRowView)this.receiptDBBindingSource.Current;

                if (String.IsNullOrEmpty(temp["id_item"].ToString()) ==
false)
                {
                    this.Validate();
                    this.receiptDBBindingSource.EndEdit();
                    this.tableAdapterManager.UpdateAll(this.dbDataSet);
                }
                else
                {
                    MessageBox.Show("Please choose a name!");
                }
            }
            catch (System.Exception ex)
            {
                MessageBox.Show("Update failed");
            }
        }
        private void scheduleReceipt_Load(object sender, EventArgs e)
        {
            // TODO: This line of code loads data into the 
'dbDataSet.itemDB' table. You can move, or remove it, as needed.
            this.itemDBTableAdapter.Fill(this.dbDataSet.itemDB);
            // TODO: This line of code loads data into the 
'dbDataSet.receiptDB' table. You can move, or remove it, as needed.
            this.receiptDBTableAdapter.Fill(this.dbDataSet.receiptDB);

            //show only material item
            this.itemDBBindingSource.Filter = string.Format("type LIKE 
'material'");
            this.itemDBDataGridView.Update();
        }

        private void itemDBDataGridView_CellClick(object sender, 
DataGridViewCellEventArgs e)
        {
            if (this.receiptDBBindingSource.Count > 0)
            {
                DataRowView temp1 = 
(DataRowView)itemDBBindingSource.Current;
                DataRowView temp2 = 
(DataRowView)receiptDBBindingSource.Current;

                this.txtNameSearchScheduleReceipt.Text = 
temp1["name"].ToString();
                temp2["name"] = temp1["name"].ToString();
                temp2["id_item"] = int.Parse(temp1["id"].ToString());

this.receiptDBBindingSource.ResetItem(this.receiptDBBindingSource.Positi
on);
            }
        }

        private void txtNameSearchScheduleReceipt_TextChanged(object 
sender, EventArgs e)
        {
            this.itemDBBindingSource.Filter = string.Format("name LIKE 
'" + this.txtNameSearchScheduleReceipt.Text.ToString() + "%' AND type 
LIKE 'material'");
            this.itemDBDataGridView.Update();
        }
    }
}
```

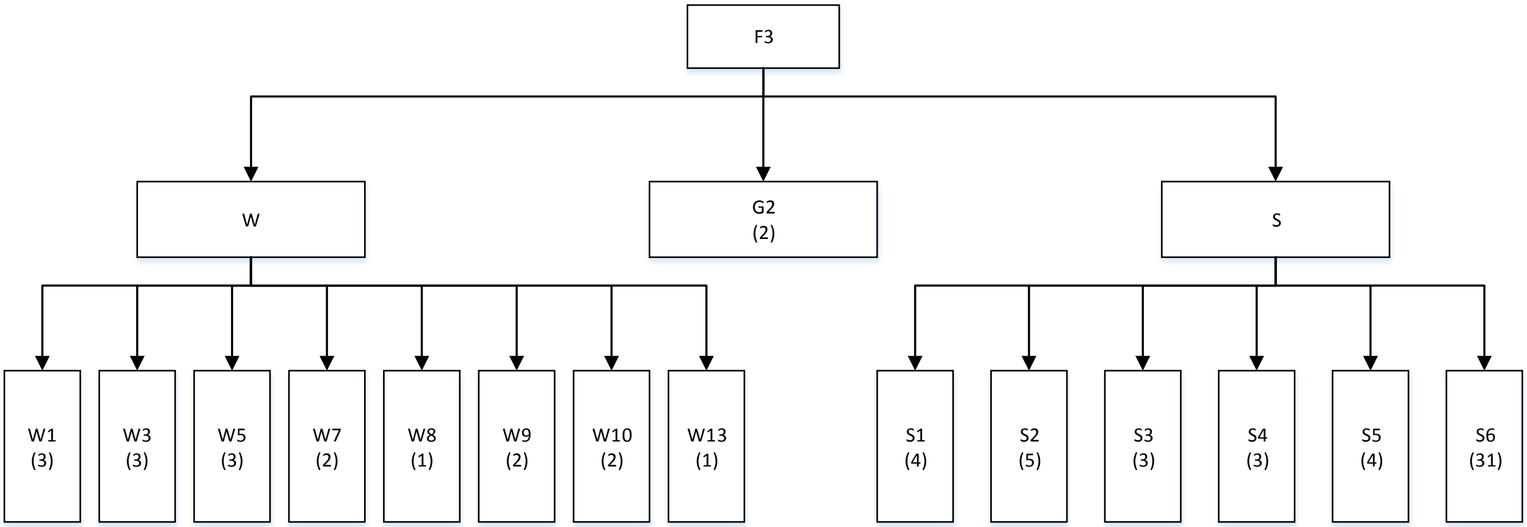
BOM data



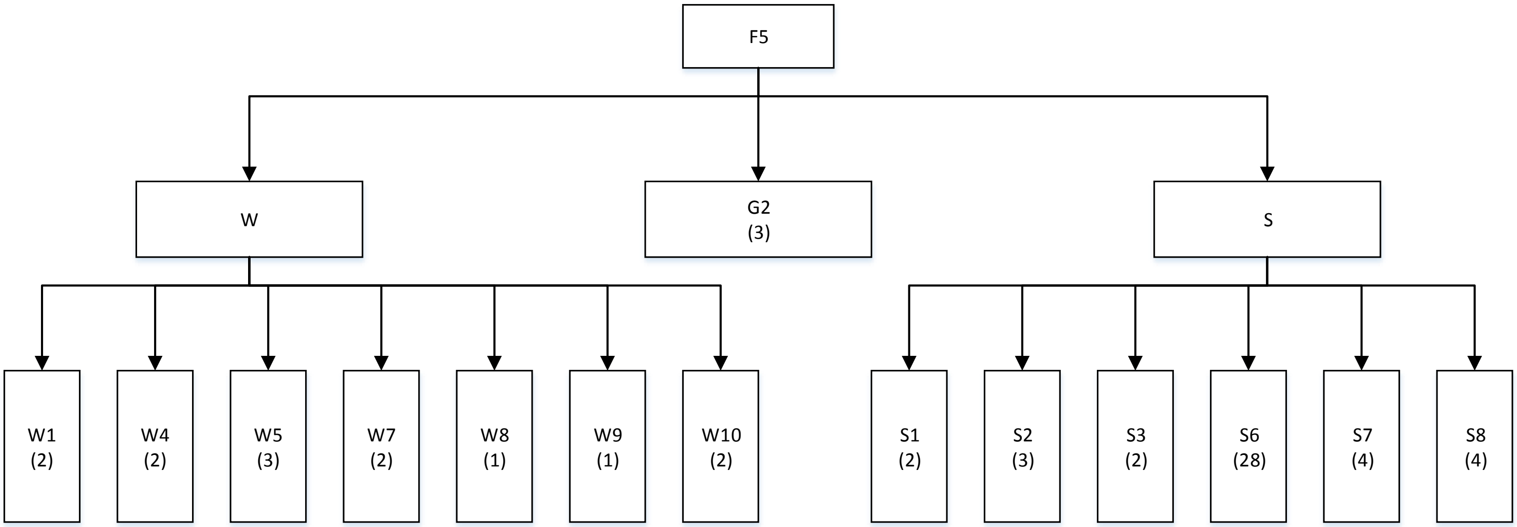
BOM of wooden cabinet THSG06



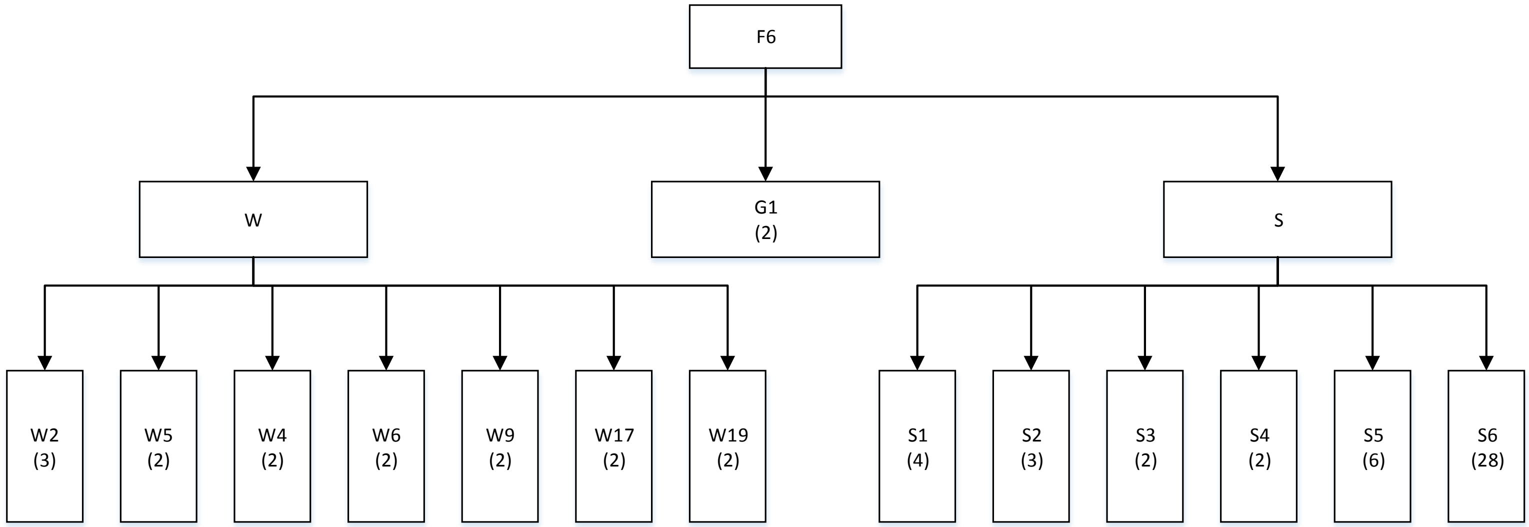
BOM of wooden cabinet THSG06



BOM of wooden cabinet THSG05



BOM of wooden cabinet THSG07



BOM of wooden cabinet THSG08

Order report of all items in 1 year

item	week	quantity	item	week	quantity	item	week	quantity
W5	3	1194	S2	22	1776	W14	35	216
W5	4	226	S2	26	1560	W14	39	252
W5	7	1290	S2	30	1848	W14	43	256
W5	8	218	S2	34	1800	W14	47	278
W5	11	1292	S2	38	1656	W15	7	82
W5	12	234	S2	42	1728	W15	11	108
W5	15	1280	S2	46	1680	W15	15	135
W5	16	236	S2	50	1920	W15	19	111
W5	19	1269	S3	6	1368	W15	23	145
W5	20	236	S3	10	1560	W15	27	137
W5	23	1121	S3	14	1500	W15	31	142
W5	24	288	S3	18	1548	W15	35	108
W5	27	1309	S3	22	1464	W15	39	126
W5	28	282	S3	26	1440	W15	43	128
W5	31	1274	S3	30	1596	W15	47	139
W5	32	250	S3	34	1572	W16	4	352
W5	35	1210	S3	38	1392	W16	8	436
W5	36	230	S3	42	1500	W16	12	468
W5	39	1251	S3	46	1464	W16	16	472
W5	40	214	S3	50	1656	W16	20	472
W5	43	1210	S4	6	1395	W16	24	576
W5	44	298	S4	10	1545	W16	28	564
W5	47	1387	S4	14	1575	W16	32	500
W5	48	216	S4	18	1500	W16	36	460
W8	3	448	S4	22	1395	W16	40	428
W8	7	666	S4	26	1425	W16	44	596
W8	11	580	S4	30	1575	W16	48	432
W8	15	667	S4	34	1620	W17	3	144

W8	19	610	S4	38	1335	W17	4	226
W8	23	662	S4	42	1530	W17	7	256
W8	27	686	S4	46	1410	W17	8	218
W8	31	680	S4	50	1650	W17	11	294
W8	35	564	S5	6	1905	W17	12	234
W8	39	615	S5	10	2025	W17	15	250
W8	43	650	S5	14	2160	W17	16	236
W8	47	688	S5	18	1935	W17	19	200
W1	3	742	S5	22	1890	W17	20	236
W1	7	904	S5	26	1770	W17	23	202
W1	11	886	S5	30	2040	W17	24	288
W1	15	884	S5	34	2040	W17	27	244
W1	19	934	S5	38	1890	W17	28	282
W1	23	807	S5	42	2055	W17	31	292
W1	27	934	S5	46	1770	W17	32	250
W1	31	857	S5	50	2220	W17	35	222
W1	35	851	S6	5	18000	W17	36	230
W1	39	832	S6	9	24000	W17	39	290
W1	43	873	S6	13	18000	W17	40	214
W1	47	957	S6	17	18000	W17	43	206
W2	3	536	S6	21	18000	W17	44	298
W2	7	626	S6	25	18000	W17	47	282
W2	11	671	S6	29	18000	W17	48	216
W2	15	619	S6	33	18000	W18	4	9
W2	19	538	S6	37	18000	W18	8	109
W2	23	541	S6	41	18000	W18	12	117
W2	27	606	S6	45	18000	W18	16	118
W2	31	658	S6	49	18000	W18	20	118
W2	35	601	W9	3	861	W18	24	144
W2	39	685	W9	7	1056	W18	28	141

W2	43	515	W9	11	1018	W18	32	125
W2	47	715	W9	15	1033	W18	36	115
W3	3	429	W9	19	952	W18	40	107
W3	7	644	W9	23	979	W18	44	149
W3	11	662	W9	27	1074	W18	48	108
W3	15	592	W9	31	1101	W19	3	197
W3	19	664	W9	35	889	W19	4	226
W3	23	583	W9	39	1013	W19	7	256
W3	27	672	W9	43	991	W19	8	218
W3	31	607	W9	47	1093	W19	11	294
W3	35	577	W10	3	718	W19	12	234
W3	39	574	W10	4	339	W19	15	250
W3	43	611	W10	7	930	W19	16	236
W3	47	661	W10	8	327	W19	19	200
W4	3	748	W10	11	836	W19	20	236
W4	7	758	W10	12	351	W19	23	202
W4	11	748	W10	15	929	W19	24	288
W4	15	786	W10	16	354	W19	27	244
W4	19	708	W10	19	887	W19	28	282
W4	23	664	W10	20	354	W19	31	292
W4	27	746	W10	23	889	W19	32	250
W4	31	762	W10	24	432	W19	35	222
W4	35	764	W10	27	961	W19	36	230
W4	39	798	W10	28	423	W19	39	290
W4	43	674	W10	31	934	W19	40	214
W4	47	870	W10	32	375	W19	43	206
W6	3	471	W10	35	804	W19	44	298
W6	7	498	W10	36	345	W19	47	282
W6	11	524	W10	39	852	W19	48	216
W6	15	494	W10	40	321	W20	4	249

W6	19	438	W10	43	916	W20	8	327
W6	23	440	W10	44	447	W20	12	351
W6	27	484	W10	47	959	W20	16	354
W6	31	512	W10	48	324	W20	20	354
W6	35	490	W11	3	78	W20	24	432
W6	39	540	W11	7	268	W20	28	423
W6	43	412	W11	11	216	W20	32	375
W6	47	574	W11	15	270	W20	36	345
W7	3	646	W11	19	222	W20	40	321
W7	7	770	W11	23	290	W20	44	447
W7	11	742	W11	27	274	W20	48	324
W7	15	768	W11	31	284	W21	4	119
W7	19	792	W11	35	216	W21	8	218
W7	23	692	W11	39	252	W21	12	234
W7	27	790	W11	43	256	W21	16	236
W7	31	728	W11	47	278	W21	20	236
W7	35	748	W12	3	43	W21	24	288
W7	39	724	W12	7	134	W21	28	282
W7	43	738	W12	11	108	W21	32	250
W7	47	834	W12	15	135	W21	36	230
G1	3	751	W12	19	111	W21	40	214
G1	7	759	W12	23	145	W21	44	298
G1	11	708	W12	27	137	W21	48	216
G1	15	761	W12	31	142	S7	6	948
G1	19	679	W12	35	108	S7	10	1188
G1	23	748	W12	39	126	S7	14	984
G1	27	756	W12	43	128	S7	18	1260
G1	31	760	W12	47	139	S7	22	1104
G1	35	729	W13	3	228	S7	26	1164
G1	39	772	W13	7	402	S7	30	1212

G1	43	668	W13	11	360	S7	34	1212
G1	47	857	W13	15	386	S7	38	1092
S1	6	2092	W13	19	364	S7	42	1140
S1	10	2328	W13	23	405	S7	46	1164
S1	14	2280	W13	27	418	S7	50	1284
S1	18	2284	W13	31	413	S8	6	416
S1	22	2156	W13	35	319	S8	10	520
S1	26	2144	W13	39	360	S8	14	448
S1	30	2356	W13	43	391	S8	18	584
S1	34	2356	W13	47	401	S8	22	540
S1	38	2100	W14	3	138	S8	26	448
S1	42	2272	W14	7	268	S8	30	524
S1	46	2140	W14	11	216	S8	34	500
S1	50	2492	W14	15	270	S8	38	548
S2	6	1656	W14	19	222	S8	42	516
S2	10	1800	W14	23	290	S8	46	524
S2	14	1848	W14	27	274	S8	50	592
S2	18	1752	W14	31	284			