

**DEVELOPMENT OF A TOOL MANAGEMENT SYSTEM FOR A JOB
SHOP**

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY**

BY

BASAR BUDAK

**IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF MECHANICAL ENGINEERING**

SEPTEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Kemal IDER
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. S.Engin KILIÇ
Supervisor

Examining Committee Members

Prof. Dr. Ömer ANLAGAN

Prof. Dr. S. Engin KILIÇ

Prof. Dr. Mustafa I.GÖKLER

Prof. Dr. Can ÇOGUN

Assoc. Prof. Dr. D. Tayyar SEN

ABSTRACT

DEVELOPMENT OF A TOOL MANAGEMENT SYSTEM FOR A JOB SHOP

Budak, Basar

M. Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. S. Engin Kiliç

August 2003, 138 pages

A tool management system for a job shop was developed in this study. The system is based on the rules of the plant of Türk Traktör ve Ziraat Makineleri A.S. in Ankara. An object-oriented methodology was used during the system development. Unified Modeling Language (UML), which is the standard diagramming notation for object-oriented development, is used to model the system. Entity-Relation (E-R) modeling is used during the construction of the database. System was developed using three-tiered Microsoft Windows Distributed Internet Applications (DNA) Architecture. Microsoft Visual Basic 6.0 and Visual InterDev 6.0, which are the members of Microsoft Visual Studio 6.0 were used as programming tools. Internet Information Server was used as web server and Microsoft Transaction Server was used as object broker. Microsoft SQL Server was used as database server.

Software was developed in three modules: gage-fixture, cutting tool-tool holder and tool requirement plan modules. Gage-fixture module tracks the location of measuring equipment, fixture and fixture assemblies; keeps the related data about these tools and manage calibration of the measuring equipments. Cutting tool-tool

holder module tracks the issue and return of cutting tools and tool holders; sharpening of cutting tools; keeps the related data about these tools. Tool requirement plan module tracks the purchase orders and cutting tool receivings from vendors; prepares tool requirement plan and suggests purchase orders according to the requirement calculation.

Keywords: Tool Management, Tool Requirement Planning, Manufacturing Systems Modeling, Distributed Internet Application, DNA for Manufacturing, Manufacturing Execution Systems, Unified Modeling Language

ÖZ

ATELYE TİPİ İŞLETMELER İÇİN TAKIM YÖNETİM SİSTEMİ GELİSTİRİLMESİ

Budak, Basar

Yüksek Lisans, Makine Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. S. Engin Kiliç

Agustos 2003, 138 sayfa

Bu çalışmada küçük ve orta ölçekli işletmeler için bir takım yönetim sistemi geliştirilmiştir. Sistem Ankara Türk Traktör ve Ziraat Makineleri A.S. fabrikasındaki kurallar temel alınarak geliştirilmiştir. Sistemin geliştirilme safhasında nesne yönelimli bir yöntembilim kullanılmıştır. Sistem modellenmesi nesne yönelimli sistem geliştirilmesinin standart semalandırma notasyonu olan UML ile yapılmıştır. Veritabanının oluşturulmasında E-R modelleme kullanılmıştır. Sistem üç katmanlı Microsoft Windows DNA mimarisi üzerine kurulmuştur. Programlama Microsoft Visual Studio 6.0'in üyeleri olan Microsoft Visual Basic 6.0 ve Microsoft Visual InterDev 6.0 ile yapılmıştır. İnternet Information Server ağ sunucusu olarak, Microsoft Transaction Server nesne sunucusu olarak kullanılmıştır. Veritabanı sunucusu olarak Microsoft SQL Server kullanılmıştır.

Yazılım üç modül olarak geliştirilmiştir: ölçme aleti-baglama aparati, kesici takım-takım tutucu ve takım ihtiyaç planlaması modülleri. Ölçme aleti-baglama aparati modülü, ölçme aleti, bağlama aparati ve bağlama aparati assemblelerinin yerlerinin takibi, bu aletlerle ilgili verilerin tutulması ve ölçme aletlerinin

kalibrasyon islemlerinin yönetimi için oluşturulmustur. Kesici takım-takim tutucu modülü, kesici takımların ve takım tutucuların işletmeye dağıtım ve geri toplanması, kesici takımların bileşenmesi ve bu takımlarla ilgili verilerin tutulması için tasarlanmıştır. Takım ihtiyaç planlama modülü, satın alma emirlerinin ve kesici takım alımlarının takibi, takım ihtiyaç planlamasının hazırlanması ve bu plana göre satın alma emirleri önerilmesi işlerini üstlenir.

Anahtar kelimeler: Takım Yönetimi, Takım İhtiyaç Planlama, İmalat Sistem Modellemesi, Dağıtık İnternet Uygulamaları, İmalat için DNA, İmalat Kontrol Sistemleri, Birleşik Modelleme Dili

To My Family

ACKNOWLEDGMENTS

I would like to express my gratefulness and appreciation to my supervisor Prof. Dr. S. Engin Kiliç for his guidance throughout the completion of this thesis. Also thanks go to Prof. Dr. Ömer Anlagan for his encouragement about the study.

I am also indebted to my colleagues Yusuf Basibüyük, Burak Sari, Umut Durak, Saleh Amaitik, Murat Akalp, Erhan Özsüer, Fatih Sari and Boran Alatas in Integrated Manufacturing Technologies Research Group (IMTRG) for their endless support all through this hard work.

I am grateful to Tolga Taskin in Türk Traktör plant who always welcomed me and helped during my thesis study.

Finally, my greatest thanks go to my parents who shaped me with their never ending patience.

TABLE OF CONTENTS

ABSTRACT.....	iii
ÖZ.....	v
ACKNOWLEDGEMENTS	viii
TABLE OF CONTENTS	ix
LIST OF FIGURES	xii
CHAPTER	
1. INTRODUCTION.....	1
1.1 Definition of CIM	1
1.5 Tool Management Concept	1
1.6 Scope	4
2. LITERATURE SURVEY	6
2.1 Tool Management Systems	6
2.1.2 Modern Tool Management Systems.....	6
2.1.2.1 Modern Tool Management Outline	7
2.1.2.2 Effectiveness of Tool Management Systems	8
2.1.2.3 Modern Tool Management Systems in Literature	8
2.2 Basic Characteristics of Object-Oriented Systems	14
2.3 Unified Modeling Language (UML).....	15
2.3.1 UML Use Case Diagrams.....	16
2.3.2 UML Static Structure Diagrams: Class and Object Diagrams ...	17
2.3.3 UML Interaction Diagrams: Sequence and Collaboration Diagrams	21
2.3.4 UML State Diagrams: State and Activity Diagrams	22
2.3.5 UML Implementation Diagrams: Package, Component, and Deployment Diagrams	24
2.4 Entity-Relation Diagrams	26

2.5 Client/Server Architecture and Terms	26
2.5.1 Two-Tier Client/Server Architecture	27
2.5.2 Three-Tier Client/Server Architecture	27
2.6 Microsoft® Windows® Distributed interNet Applications (DNA) Architecture	28
2.6.1 Component Object Model (COM)/Distributed Component Object Model (DCOM)	30
2.6.2 Microsoft Transaction Server (MTS)	32
2.6.3 Microsoft SQL Server	33
2.6.4 Internet Information Server	33
2.6.5 Visual Studio	34
3. SYSTEM DEVELOPMENT	36
3.1 Overview	36
3.2 Requirement Analysis	37
3.3 Analysis and Design Phases	39
3.3.1 E-R Model	45
3.3.2 Authentication Structure	48
3.3.3 Web Site Diagram	50
3.4 System Implementation	52
3.4.1 Data Tier Implementation	53
3.4.2 Business Tier Implementation	54
3.4.3 Client Tier Implementation	59
4. TOOL MANAGEMENT SYSTEM OF IMTRG	62
4.1 Overview	62
4.2 Gage-Fixture Module	63
4.3 Cutting Tool-Tool Holder Module	66
4.4 Tool Requirement Plan Module	70
5. CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK	73
REFERENCES	77
APPENDICES	
A. UML DIAGRAMS	85
A.1 Use Case Diagrams	85

A.2 Activity Diagrams	87
A.3 Sequence Diagrams	89
A.4 Collaboration Diagrams.....	97
A.5 Class Diagram.....	102
A.6 Package Diagram.....	102
A.7 Component Diagrams	102
A.8 Deployment Diagram	103
B. USERS MANUAL	104

LIST OF FIGURES

FIGURES

2.1 The Effect of Tool Control on Inventory and Purchasing [13]	13
2.2 Notation for Use Case Diagram	17
2.3 Notation for Classes	17
2.4 Notation for an Association.....	18
2.5 Notation for Multiplicity of Association.....	18
2.6 Notation for Generalization.....	19
2.7 Notation for Aggregation	19
2.8 Notation for Composition.....	19
2.9 Notation for a Constraint and for Notes	20
2.10 Notation for Objects	20
2.11 Notation for a Sequence Diagram	22
2.12 Notation for Collaboration Diagram	22
2.13 Notation for State Diagrams	23
2.14 Notation for Activity Diagrams	24
2.15 Notation for Package Diagrams	25
2.16 Notation for Component Diagrams	25
2.17 Notation for Deployment Diagrams	26
2.18 Two-Tier Client/Server Architecture [19]	28
2.19 Three-Tier Client/Server Architecture [19]	28
2.20 Windows DNA Architecture	29
2.21 Windows DNA for Manufacturing [21].....	30
2.22 Architecture of Microsoft's Enterprise Development Strategy [19]	32
3.1 Development Road Map	36
3.2 Use Case Diagram of Cutting Tool-Tool Holder Module	38

3.3 Activity Diagram of “Issue Cutting Tool” Use Case	38
3.4 Class Diagram	42
3.5 Sequence Diagram of “Issue Cutting Tool” Use Case	42
3.6 Collaboration Diagram of “Issue Cutting Tool” Use Case	43
3.7 Package Diagram.....	44
3.8 Component Diagram of Client Package	44
3.9 Component Diagram of Business Package	44
3.10 Deployment Diagram	44
3.12 E-R Model of Authentication Structure	48
3.13 Database Diagram of Authentication Structure	49
3.14 Web Site Diagram	52
3.15 Tool Management Database Diagram.....	54
3.16 Design View of CTTHToolMng Module in Visual Basic 6.0 IDE.....	55
3.17 MTS View of Tool Management System.....	59
3.18 Design View of CTTHClientToolMng Project in Visual InterDev	60
3.19 IIS View of Tool Management System.....	61
4.1 C/U/D Fixture Page	63
4.2 Search Fixture Assembly Page	64
4.3 Search Measuring Equipment Page	65
4.4 Search Calibration Information Page	66
4.5 Search Cutting Tool Page	67
4.6 Search Used Cutting Tool Page	68
4.7 Sharpen Cutting Tool Page	69
4.8 Issue Tool Holder Page	70
4.9 Requirement Plan Page	71
4.10 C/U Purchase Order Page	72
A.1 Use Case Diagram of Tool Requirement Plan Module	85
A.2 Use Case Diagram of Cutting Tool-Tool Holder Module	86
A.3 Use Case Diagram of Gage-Fixture Module	86
A.4 Activity Diagram of Cutting Tool Issue	87
A.6 Activity Diagram of Requirement Plan Run	88
A.7 Activity Diagram of Cutting Tool Return	89

A.8 Sequence Diagram of Fixture Assembly Creation.....	89
A.9 Sequence Diagram of Fixture Assembly Deletion.....	90
A.10 Sequence Diagram of Fixture Creation.....	90
A.11 Sequence Diagram of Fixture Deletion.....	90
A.12 Sequence Diagram of Measuring Equipment Creation.....	91
A.13 Sequence Diagram of Measuring Equipment Deletion.....	91
A.14 Sequence Diagram of Cutting Tool Receiving from Vendors	91
A.15 Sequence Diagram of Cutting Tool Issue	92
A.16 Sequence Diagram of Cutting Tool Return.....	92
A.17 Sequence Diagram of Purchase Order Creation.....	93
A.18 Sequence Diagram of Purchase Order Update	93
A.19 Sequence Diagram of Purchase Order Deletion.....	93
A.20 Sequence Diagram of Part Deletion	94
A.21 Sequence Diagram of Measuring Equipment Movement	94
A.22 Sequence Diagram of Fixture Assembly Movement.....	94
A.23 Sequence Diagram of Sending to Sharpening Room	95
A.24 Sequence Diagram of Return from Sharpening Room.....	95
A.25 Sequence Diagram of Sharpening of Cutting Tools	96
A.26 Sequence Diagram of Tool Holder Issue	96
A.27 Sequence Diagram of Tool Holder Return.....	97
A.28 Collaboration Diagram of Fixture Assembly Creation.....	97
A.29 Collaboration Diagram of Fixture Assembly Deletion.....	97
A.30 Collaboration Diagram of Fixture Creation.....	98
A.31 Collaboration Diagram of Fixture Deletion.....	98
A.32 Collaboration Diagram of Measuring Equipment Creation	98
A.33 Collaboration Diagram of Measuring Equipment Deletion	98
A.34 Collaboration Diagram of Cutting Tool Receiving from Vendors	99
A.35 Collaboration Diagram of Cutting Tool Issue	99
A.36 Collaboration Diagram of Cutting Tool Return	99
A.37 Collaboration Diagram of Purchase Order Creation	99
A.38 Collaboration Diagram of Purchase Order Update.....	100
A.39 Collaboration Diagram of Purchase Order Deletion	100

A.40 Collaboration Diagram of Part Deletion.....	100
A.41 Collaboration Diagram of Measuring Equipment Movement	100
A.42 Collaboration Diagram of Fixture Assembly Movement	100
A.43 Collaboration Diagram of Sending to Sharpening Room.....	101
A.44 Collaboration Diagram of Return from Sharpening Room	101
A.45 Collaboration Diagram of Sharpening of Cutting Tool.....	101
A.46 Collaboration Diagram of Tool Holder Issue	101
A.47 Collaboration Diagram of Tool Holder Return	101
A.48 Class Diagram.....	102
A.49 Package Diagram.....	102
A.50 Component Diagram of Business Package	102
A.51 Component Diagram of Client Package	103
A.52 Deployment Diagram	103
B.1 Login Page	104
B.2 Welcome Page	105
B.3 C/U/D Fixture Page	105
B.4 Search Fixture Page	106
B.5 C/U/D Fixture Assembly Page	106
B.6 Move Fixture Assembly Page	107
B.7 Search Fixture Assembly Page	107
B.8 Search Fixture Assembly Movement Page	108
B.9 C/U/D Measuring Equipment Page	110
B.10 Move Measuring Equipment Page	110
B.11 Search Measuring Equipment Page	111
B.12 Search Measuring Equipment Movement Page	111
B.13 Update Calibration Information Page	112
B.14 Create/Update Calibration Log.....	112
B.15 Search Calibration Information Page.....	113
B.16 Calibration Report Page	113
B.18 C/U Cutting Tool Page	117
B.19 Search Cutting Tool Page	117
B.20 Search Used Cutting Tool Page	118

B.21 Issue Cutting Tool Page	118
B.22 Search Cutting Tool Issue Page	119
B.23 Return Cutting Tool Page	119
B.24 Search Cutting Tool Return Page	120
B.25 Send To Sharpening Room Page	120
B.26 Search Sending To Sharpening Page	121
B.27 Return From Sharpening Room Page	121
B.28 Search Return From Sharpening Page	122
B.29 Search Sharpening Room Cutting Tool Page	122
B.30 Sharpen Cutting Tool Page	123
B.31 Search Sharpening Page	123
B.32 C/U Tool Holder Page	124
B.33 Search Tool Holder Page	124
B.34 Issue Tool Holder Page	125
B.35 Search Tool Holder Issue Page	125
B.37 Search Tool Holder Return.....	126
B.38 Run TRP Page	127
B.39 Requirement Plan Page	127
B.40 Single Requirement Plan Page	128
B.41 Suggested Purchases Page	128
B.42 Single Suggested Purchase Page	129
B.43 Warning Page	129
B.44 Calendar Page	130
B.45 C/D Part Page	130
B.46 List Parts Page	131
B.47 C/U/D BOT Page	131
B.48 Search Bill of Tooling Page	132
B.49 C/U/D Schedule Event Page	132
B.50 Search Schedule Page	133
B.51 C/U Purchase Order Page	133
B.52 Search Purchase Order	134
B.53 Receive Cutting Tool From Vendor	134

B.54 Search Receiving From Vendor.....	135
B.55 C/U/D User Page	135

CHAPTER 1

INTRODUCTION

1.1 Definition of CIM

The acronym CIM (Computer Integrated Manufacturing) is used to mean the integration of business, engineering, manufacturing and management information that spans company functions from marketing to product distribution [1]. Integration has become the key word with the effects of expanding global competition and products of increasing variety and lower demand. All the organs of the manufacturing enterprise stated in the definition of CIM should be integrated so that enterprise can be more responsive to the customer demand variations in the global market.

In many companies, CIM initially concentrates on integrating manufacturing as a first priority or goal to be achieved. Other functions are then linked to manufacturing. Thus, manufacturing can be considered as the core of the CIM. The Computer and Automation Systems Association of Society of Manufacturing Engineers defined CIM as the integration of the total manufacturing enterprise through the use of integrated systems and data communications coupled with new managerial philosophies that improve organizational and personnel efficiency.

1.5 Tool Management Concept

Tool management as an idea is easy: For maximum manufacturing productivity, get the right tool to the right machine at the right time to work on

any given part number [2]. Hundreds or even thousands of different tool types may be required in many facilities, and no matter how well jobs are scheduled and machines run, processing operations can only proceed if the right tools are available at the right machines at the right times. The importance of tooling can be seen in the following statistics:

- ?? 16% of scheduled production time is missed because tooling is not available.
- ?? 40-60% of a foreman's time is spent expediting tools and related materials
- ?? 7-10 times more money is spent on tools, jigs, etc. than on capital equipment
- ?? up to 30% of tool inventories consists of lost tooling.

Additional tool-related problems include loss of control of tooling, stockpiling of tools at workstations, increased downtime, employee theft of tools, and weakening in the firm's competitive position [3]. In order to properly organize and control tooling, tool management systems are implemented. A tool management system typically consists of a database for storing tool-related data, a system for identifying and tracking tools (e.g., barcode), and software for executing various tool management functions.

Conceptually, tool management can be classified into tool-level, machine-level, and system-level issues [4]. The tools that are in consideration may be cutting tools, tool holders, tool adapters, fixtures, molds, dies, gages and any recycling tools.

At tool level, tool management issues are determination of tool types, feed rates, and machining speed for a given operation at machine level. At machine level, the issues are equipment selection, tool/part sequencing on a flexible machine, tool placement in a magazine, and tool replacement. At system level, tool management problems arise in master production planning, part-type selection, machine grouping and loading, machine sequencing and process

monitoring, process planning for economic production rates, spares management, and tooling inventory management. The decisions made at one level affect the decisions at other levels, therefore an integrated system is necessary [4].

A number of recent developments in the field of manufacturing systems engineering have influenced the role and function of Tool Management Systems. The first and the foremost of these is the Just In Time Manufacturing (JIT). Where this principle is implemented, tool availability and reliability become critical to the success of the entire corporation, since the non-availability due to breakdown of a tool can directly and immediately endanger the delivery of product to the customer. The key role of a tool management system under JIT is to ensure the availability of all tools when required. A second current development affecting tool management under CIM is Simultaneous Engineering. This can involve much shorter timespans for tool design and development, which means that feedback from the performance of current designs has to be much quicker. Another influential concept is Cellular Manufacturing. The role of a tool management system in a cellular manufacturing system is to control the location and status of the tools which are distributed to the cells in the factory. Total Quality Management places more emphasis on the prevention of mistakes rather than their detection, which gives a tool management system the responsibility of ensuring that the tool performance is inside tolerances. One of the most important concepts affecting Tool Management is Total Productive Maintenance. This philosophy encourages the ownership of facilities and tools by those who use them, delegating more tasks of maintenance and adjustment to the operator. Here, a tool management system is used as a communication tool between the users of tools and those who maintain/issue/design them. Lean Manufacturing increased the essence of the tool management systems by emphasizing importance of eliminating the waste, which can be interpreted to mean waste of time, space, and energy and cost. Thus, a tool management system in a state-of-the-art company has to assist in the minimization of stock levels, maintenance effort, transportation time, and production delay [5].

A tool management system can be a number of things, depending on the size of the company and the number of tools that the company has. It can be a card system, or a barcode system integrated with the badge number control of workers. It can also be a mechanical system for cabinet control. Dispensing units like wending machines can be used. Totally it can be a computerized system that tracks tools; transacts tool issues, returns, rework orders and results, calibrations of gages; suggests purchase orders according to the requirement plan calculated by the software. Computerized tool management systems are available for several years, which have integrated barcode technology to make their use easier and faster. Even an active tool tracking system may be developed that monitors tool movements using embedded miniature battery-powered transmitters, which may be radio, infrared, ultrasonic or some combination of these. The detector stations strategically placed in the shop floor can identify the tools and pass the related tool data to the central computer. Such a system can be effective on the productivity where high tool losses occur [6].

Here the problem is to define the boundaries of the tool management system. Many of the solutions mentioned serve a lot for manufacturing productivity, but the aim is to find the optimum combination for maximum raise in productivity. The specification and objectives of a tool management system in an advanced manufacturing environment need to be tailored to the company's specific needs [5]. Today, computerized tool management systems are used and being developed in some companies also in Turkey. But, if it is more economical, necessary modules of a readily available software can be bought rather than developing it.

1.6 Scope

The scope of this study is to develop a computerized tool management system. The system is planned to have three main modules. The modules are gage-fixtured, cutting tool-tool holder and tool requirement plan. Three-tier application architecture is used. Database was developed using Microsoft SQL

Server. Microsoft Transaction Server (MTS) is used as object broker and transaction coordinator. Microsoft Internet Information Server (IIS) is used as web server. All these services are hosted on NT platform. Microsoft Visual Studio is selected as the development tool suit. Visual Basic is selected for ActiveX development and Visual InterDev is selected for web site development.

CHAPTER 2

LITERATURE SURVEY

2.1 Tool Management Systems

In this section, tool management systems will be discussed. Modern tool management systems in the literature will be presented. Information about traditional tool management systems can be found in the reference [7].

2.1.2 Modern Tool Management Systems

In traditional tool management systems, the tools can be found by indexing tool cards. Unfortunately, only the tool operators know how to index these tool cards, but it is impossible to know the tool conditions and status. If the tools are issued to some workstations, it is hard to track exactly where the tools are and their conditions. Tools disappear very often, it results in buying new tools or making house-made tools. Besides, the work piece designers design with unnecessary geometries that need new tools since they do not have the efficient tool data about the tools that are available in the factory. All these cause the tool inventory to become bigger and bigger. Accordingly the investment in tools is becoming bigger [9].

There are two major directions that studies have been carried out. One direction focuses on tool related decision problems consisting determination of cutting conditions, tool magazine arrangements, tool store dimensioning, tool-sharing concept, tool allocation, tool flow planning, tool replacement. On the other hand, studies on improving the integration of the tool management within

total system are the subjects of the second direction. Rather than the first group, the second one will be discussed in the following paragraphs since the scope of this study covers the second one.

2.1.2.1 Modern Tool Management Outline

According to Gray *et al* [4], tool management broadly requires:

- ?? A design strategy to coordinate tooling inventory, tool tracking, tool handling, and tool loading and unloading;
- ?? A planning strategy to ensure that the appropriate tools are available when needed and are provided in the right quantities;
- ?? A scheduling strategy to account for availability and tool changes;
- ?? A control strategy to coordinate either manual or automatic tool transfers between machines and tool cribs; and
- ?? A tool monitoring strategy to identify and react to unexpected tool wear and breakage.

Rhodes [8] in his paper proposes possible tool management functions as following:

- ?? **Tool Crib Management:** Managing tool crib inventories, order to vendors, components required for an assembly and kits of tools for specific operation.
- ?? **Automatic Tool Identification and Tracking:** Identification of individual tools either by barcodes or by newer technologies like solid-state identification chips.
- ?? **Individual Tool Data Management:** Keeping the track of the individual data of the tools like tool usage, tool offsets etc.
- ?? **Integrated Hierarchical Control:** Integrating medium-term planning and procurement of the enterprise down through kitting to a detailed schedule and tracking condition on machines.
- ?? **Tool Scheduling:** Scheduling of tools according to plant schedule.

2.1.2.2 Effectiveness of Tool Management Systems

Gray *et al* [4] state that to ensure the quality performance of a tool management system, a high level of integration is necessary between tooling capabilities and the other production functions, including process planning, scheduling, part design, and programming. They point out that an effective, well-managed information system is necessary to collect and distribute tooling data across these functions.

O'Brien and Schaefer [5] define effective tool management system as the system that includes all aspects of information management necessary to ensure the continued availability of tools for production and to avoid tools to become a bottleneck to the use of production facilities in the organization. Manufacturing intelligence techniques give various opportunities for enhancing tool management. Intelligent decisions are required mainly for the timing and prediction of maintenance and purchasing.

Kehoe *et al* [10] state that the effectiveness of the tool management system increases with the success of the specifications of the requirements. They point out that importance of effective manufacturing information systems is not met with the established methods for defining the requirements specifications. Inherent complexity of the existing information systems, which are often informal and poorly structured, led to the unsuccessful user-led solutions. Poorly defined user requirements forced many manufacturing companies to select vendor-led solutions.

2.1.2.3 Modern Tool Management Systems in Literature

Several tool management systems can be found in the literature developed by either universities or software companies.

TMS-2000 tool management system, which is developed by Stanley-Vidmar, was bought and implemented by Sherman, one of the 23 manufacturing facilities of Fisher Control. TMS-2000 is designed to run on DEC PDP 11 and VAX computer systems, the software uses Digital Equipment's standard software tools, including the record management system (RMS) and DatatrieveTM for report generation. The modules of the software selected by Fisher were inventory control, bill of tooling, and purchase information. The inventory control module provides control of tooling inventory on the shop floor, in the tool crib, and in any other storage locations. When a tool is issued, whom the tool was issued to and which machine it was allocated to are recorded by TMS-2000. Bill of tooling module issues and tracks packages of tools (kits) for a particular part's manufacturing operation, includes creation, editing, and deletion of bill of tools records, checks the inventory status of each tool or tool assembly required for the operation prior to scheduling, issues packages or kits to the shop floor [2].

An integrated computer tool management system called ICTMS Version 1.0 has been developed at the Laboratory of Process and Production Engineering at Technical University of Denmark. The aim of the study was to integrate tool management system into the CIM cell at the Laboratory and to transfer technology to Danish industries to improve the tooling activities. The detailed objectives are listed as following:

- ?? Tool data is managed by a computer database system in a client server of a network.
- ?? Tool database can be integrated with the production database of the company.
- ?? Tracks tool conditions and physical locations.
- ?? Supports the functions of CIM environment.
- ?? Supports production and process plan to generate time phased tool requirement list.

- ?? Supplies tool usage information (which machine or process the tool can be applied, etc.)
- ?? Supplies tool assembly information, e.g. what elements are needed to assemble a tool and where they are.
- ?? Supports cutting feed data selection for different materials.
- ?? Controls tool inventory to the minimum.
- ?? Supplies tool purchasing principles.
- ?? Calculates tool expenses.
- ?? Records historical tool usage.
- ?? Supports tool flow simulations.

The ICTMS Version 1.0 uses IBM OS/2 Database Manager, which is a relational database system. It runs in the client server of the OS/2 network. IDEFO was used as the modeling methodology. Tool database was integrated to central database of C17M facilities at the laboratory. User interface is developed on DataEase for Windows. Image files are stored in a data field different than the database. Layout of ICTMS is given in Figure 2.5 [9].

The other example was the result of the studies on defining a methodology for the specification of tool management information systems, carried out in Department of Industrial Studies in University of Liverpool [10]. By using the defined methodology LUM-TM, a tool management system was developed in a major British FMS user company. While defining the methodology the borders of a tool management system was drawn as it was consisting basic tool data management, tool inventory control, tool requirements planning, tool specification for process planning, tool preparation & assembly, system integration & security and performance monitoring & management reporting. Nevertheless, in the project carried out in industry, the project included only FMS tooling of the factory. Tooling specification of new and modified tools, specification of geometry, bill of material assembly instruction were also included.

The LUM-TM presents five phases for requirement analysis. They are requirement analysis, which consists of stages -analysis of current system, definition of requirements and evaluation of solutions- requirements solutions, specification of physical system and system implementation plan. Use of data flow diagrams (DFD) and entity-relation (E-R) diagrams are proposed during these phases [10].

Another tool management system is TIMS (Tool Inventory Management System) implemented in a factory of US Army, which is operated by Textron/Lycoming. The system was developed as a project of IREP (Industrial Resources Enhancement Programs). With the aid of several outside computer consultants, an on-line system was developed to control tooling activities such as:

- ?? Entry of new tools into the system.
- ?? Generation of “pick and put” lists.
- ?? Issue of move orders to shop, rework, or vendor (by employee badge number, department number, or vendor code).
- ?? Tool return (including calibration needs).
- ?? Development of user-friendly tool status CRT menus.
- ?? Tool usage (a tool/operation cross reference).
- ?? Tools needed to make a particular part.
- ?? Selection of alternative tooling.
- ?? Identification of tools due for calibration.
- ?? Calibration activity.
- ?? Collection of tool maintenance data.
- ?? Bin location of every tool (including initial assignment and subsequent changes).
- ?? Tool tracking (by employee badge number and department number) [11].

Advanced Tool Management System (ATMS) has been developed by ITC Integrated Systems, Inc. It has three versions using Access, SQL Server or Oracle

database management systems. It has several sections as “Quick Searches”, “Tool Crib Management”, “Gage Control”, “Purchasing Control”, “Reporting”, “Data Management”, and “System Administration”. Several tool cribs can be defined in the factory. In “Quick Searches” section, user can search the tools according to various criteria. “In Tool Crib Management” section, user can issue and return tools, send tools to sharpening, return tools from sharpening, receive tools from vendors, send tools to vendor back, and swap tools between machine tools. In “Gage Control” section, gages can be sent to calibration and returned from calibration. In “Purchasing Control” section, new purchase orders can be created and existing ones can be edited. In “Reporting” section, several standard and customized reports can be viewed. In “Data Management” section, user can define tools, assemblies, - which are assembled tools-, kits, -which are groups tools under an assigned number-, and suppliers of the tools. In “System Administration” section, different users with various roles can be created [12].

Another tool management system developed by ITC Integrated Systems, Inc. is Tool Location Control (TLC) Tool Management System. TLC generates reorder requirements; records historical usage of tools; tracks the tooling issued to the jobs, machines, and employees; maintains primary and secondary locations in the crib; details order, usage, scrap, and rework histories; Allows tooling to be pre-kitted and allows kits within kits to any level; offers transaction and summary cost reporting of tool activity; provides a detailed trail of all tool transactions; allows tools to be reserved for specific jobs; assists the keeping of calibration records for gages; allows reference space for tool, machine, vendor, and user information; provides several standard and custom management reports; is designed to be multi-user for interdepartmental communication. The chart in Figure 2.5 demonstrates the typical tool inventory activity during the first 12 months of installation. The purchasing activity goes way down as the excess inventory is used up in the first 5 months of the installation of TLC. As the inventory begins reaching the correct level the purchasing activity increases again [13].

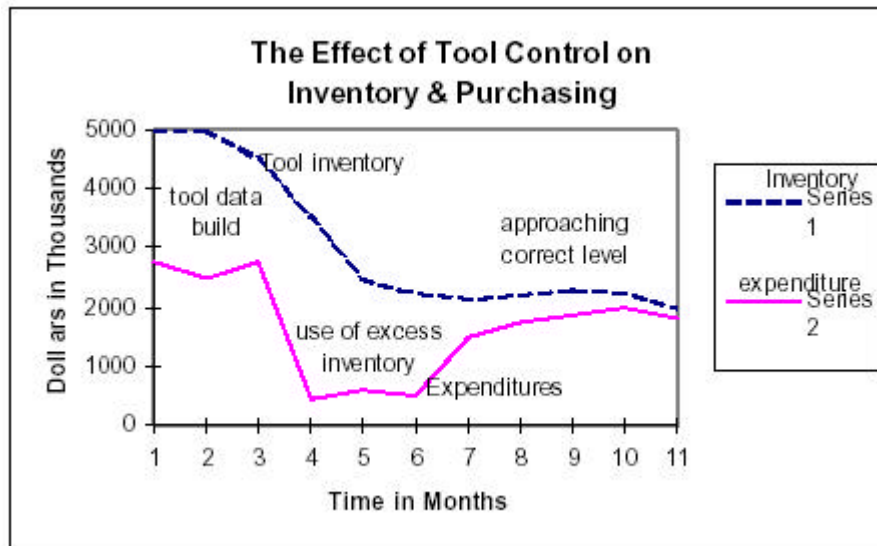


Figure 2.1 The Effect of Tool Control on Inventory and Purchasing [13]

Bell *et al* present a tool management system designed for a flexible machining facility [14]. It consists of four basic modules for tool requirement planning, part scheduling, tool management strategy selection and tool management interrogation. These modules are supported by a centralized integrated database management system. Scheduling module firstly decides the batch size by considering tool magazine capacity and transporter capacity, then sequences the jobs according to the preferred scheduling rule. After scheduling jobs, in the tool requirement plan module, it decides the number of tools required by the related job according to the tool issue strategy adopted. The module calculates the tool requirement for each type of tool, tool inventory, tool life usage, throughput time and machine and transporter utilization. It also traces the tool life during operations. For this purpose, a separate tool-tracing file is allocated to each machine and is updated during operations throughout the manufacturing period. It also tracks how long each tool is kept on which machine, when it is changed, after which job it is changed, if it is worn and if sister tools are required. There are several different strategies applied to the issue of tools; each has its own advantages and disadvantages. The tool management strategy selection module has been designed to select the most convenient tool management strategy and tool issue strategy for the related hardware

configuration. The last module is knowledge-based output analysis module, which has two functions: performance analysis and problem detection. Performance analysis is carried out with respect to seven criteria: manufacturing workstation utilization, central tool store utilization, secondary tool store utilization, primary tool store utilization, tool utilization, transporter utilization, throughput and lead time. Problem detection function supports four main hardware and operational problems: manufacturing cell problem, manufacturing workstation problem, tool store problem and tooling problem.

2.2 Basic Characteristics of Object-Oriented Systems

Objects in the real world can be characterized by two things: data and behavior. For example television has data as brightness, contrast, etc. and behaviors as they can be switched on and off, the channels can be changed and so on. Since software systems are solving real world problems these real world objects can be modeled in a similar way in programming, which is called object-oriented design/programming. Object orientation is a better abstraction of the real world compared to the structured programming.

The data of the object are called “Attributes” and the behaviors of the object are called “Methods”. A “Class” is a template, which defines the attributes and the methods. The class definition is used to create “Instances” of the class, which are the actual objects used within the application. A class explains all the data and behaviors that each object of that type contains. If a registration system for a school is to be built, one of the possible classes would be “student” class. Each student like Erhan Özsüer, Fatih Sari, etc. would be the objects of the student class.

The fundamental concepts in Object Oriented Programming are:

- ?? **Encapsulation and Information Hiding:** Encapsulation is a mechanism that combines the behaviors and data into a single object. The principle of information hiding suggests that only the information required to use an

object should be available outside the object. In short, objects are treated like black boxes. All an object should need to know is the set of methods that other objects can perform and what messages need to be sent to trigger them. If the data or the way of methods, which are executed in an object, is to be changed, the other objects are not affected by these changes.

?? **Inheritance:** Inheritance means that classes can reuse attributes and methods that have been defined in other classes. In many cases, there may be similarities among different classes in an information system. Although each class can be defined separately, it might be simpler to define one general class containing the data and the methods, which are common for some other classes. General classes that have the common data and methods are called “super-classes” and the ones, which inherit these data and methods are called “subclasses”.

?? **Polymorphism:** Polymorphism means that the same message can be interpreted differently by different classes of objects. For example, if “draw yourself” as a message is sent to a square object, a circle object, and a triangle object, the results will be different although the message is the same [15, 16].

2.3 Unified Modeling Language (UML)

UML is an object oriented modeling language used to describe information systems. It provides a common vocabulary of object oriented terms and a set of diagramming techniques that are rich enough to model any systems development project from analysis through implementation. UML is nothing more than a notation and it does not dictate any specific approach to developing information systems.

Systems are modeled to:

- ?? Provide structure for problem solving
- ?? Experiment to explore multiple solutions

- ?? Furnish abstractions
- ?? Reduce time-to-market for business problem solutions
- ?? Decrease development costs
- ?? Manage the risk of mistakes

UML is a graphical language for specifying, visualizing, constructing, and documenting the artifacts of software systems. UML goals are:

- ?? Define an easy-to-learn but semantically rich visual modeling language
- ?? Unify the Booch, OMT, and Object modeling languages
- ?? Include ideas from other modeling languages
- ?? Incorporate industry best practices
- ?? Address contemporary software development issues, which are scale, distribution, concurrency, executability, etc.
- ?? Provide flexibility for applying different processes
- ?? Enable model interchange and define repository interfaces

2.3.1 UML Use Case Diagrams

A use case diagram provides a functional description of a system and its major processes and places a boundary on the problem to be solved. It also provides a graphic description of who will use the system and what kinds of interactions they can expect to have with the system. In use case modeling entities outside the problem area that are going to use the application are called “actors”. An actor may be a person, another software application, or a hardware device. Processes that occur within the application area are called “use cases”. Use case is represented by an oval. The name of the use case is written inside the oval. A rectangle with rounded corners represents the application or the system. A line connecting an actor and a use case represents the interaction between them. Notation of UML use case diagram can be seen in Figure 2.2.

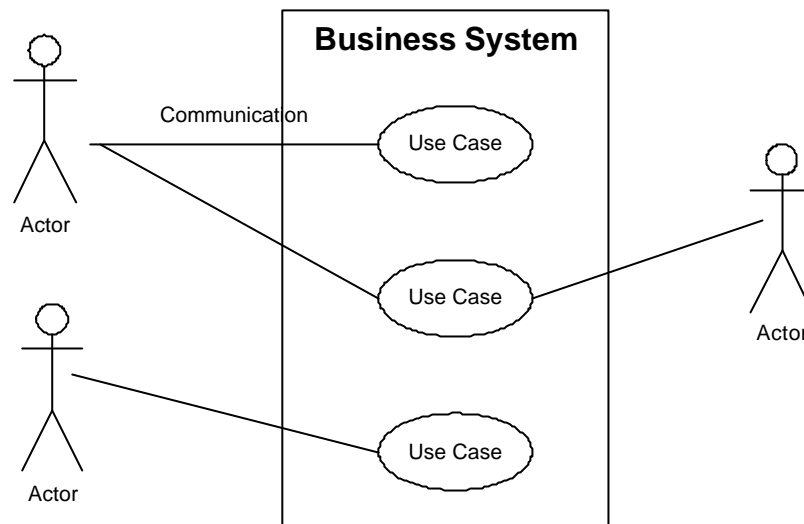


Figure 2.2 Notation for Use Case Diagram

2.3.2 UML Static Structure Diagrams: Class and Object Diagrams

The UML class diagram shows classes and the relationships between classes. The simplest way of representing a class in UML is to draw a rectangle and place the name of the class inside the rectangle. Later this class can be detailed by dividing the rectangle into three horizontal compartments. In the top section the name of the class is indicated, in the second section the attributes are listed, and in the lower section the methods are listed. Notation for classes can be seen in Figure 2.3.

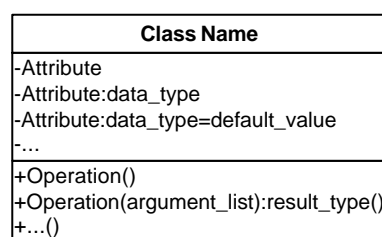


Figure 2.3 Notation for Classes

Any physical or conceptual connection between two or more classes are documented by “association”. The most generic association is represented with a line between two classes. An association is naturally bidirectional. In other words,

an association implies that the objects of one class will send messages to operations associated with the objects of the second class and that the second class's objects will respond when appropriate. The name of the association is written just above or below the line connecting the two classes. But, if the association represents a single obvious relationship, its name can be omitted. The name of the role played by the class is called “role”, and it is written just over or below the association line, next to the class. Notation for association can be seen in Figure 2.4.

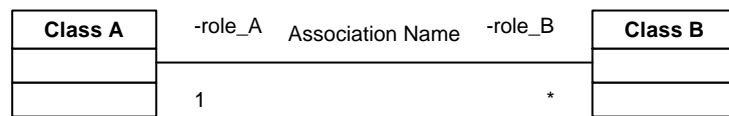


Figure 2.4 Notation for an Association

UML also provides a way to indicate the multiplicity of an association. “Multiplicity” refers to how many objects of one class can relate to each object of another class. Notation for multiplicity of associations can be seen in Figure 2.5.

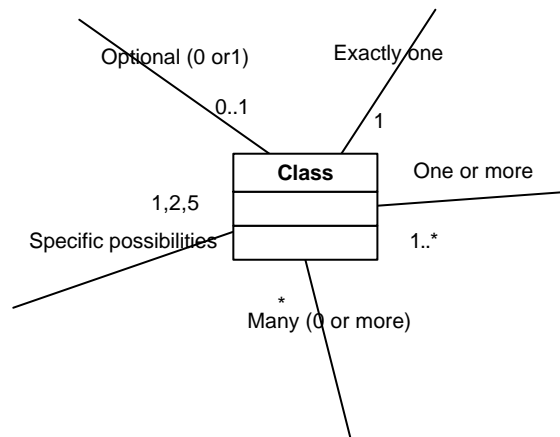


Figure 2.5 Notation for Multiplicity of Association

“Generalization” refers to a relationship when one class shares its properties, methods, relationships, and constraints with another class. UML indicates generalization by means of hollow-headed arrow. The arrow always

points towards the super-class. Notation for generalization can be seen in Figure 2.6.

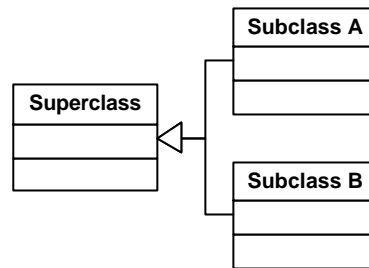


Figure 2.6 Notation for Generalization

An “aggregation” refers to part-whole relationships. It is indicated by placing a small diamond at the end of the association line that runs between part classes and the whole class, pointing at the whole class. Notation for aggregation can be seen in Figure 2.7.

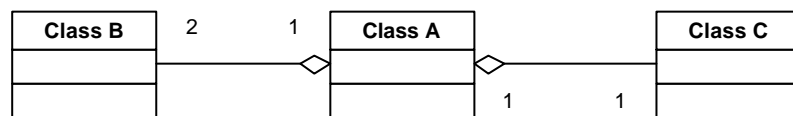


Figure 2.7 Notation for Aggregation

Composition relationships are strong form of aggregation. It is indicated by placing a filled diamond at the end of the association line. Composition indicates that the lifetime of Class B and Class C is dependent upon Class A. Notation for composition can be seen in Figure 2.8.

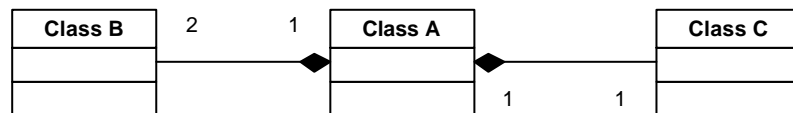


Figure 2.8 Notation for Composition

“Constraints” are functional relationships between elements of a class or object model. A constraint sets limits on the values that one of the elements can take. UML allows write notes on diagrams by putting them in rectangles with a

dog-eared right corner. Notation for constraints and notes can be seen in Figure 2.9.

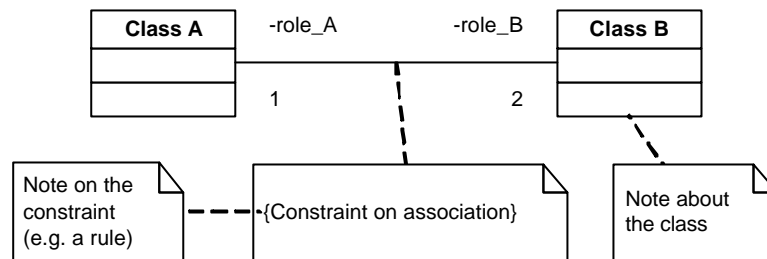


Figure 2.9 Notation for a Constraint and for Notes

In some situations, objects of classes may need to be diagrammed. UML indicates an object as a rectangle like a class. However, the difference between an object and a class can be noticed by the way the name is written inside the rectangle. An object can be named in UML in three ways:

- ?? :Class Name
- ?? object name:
- ?? object name:Class Name

Notation for objects can be seen in Figure 2.10.

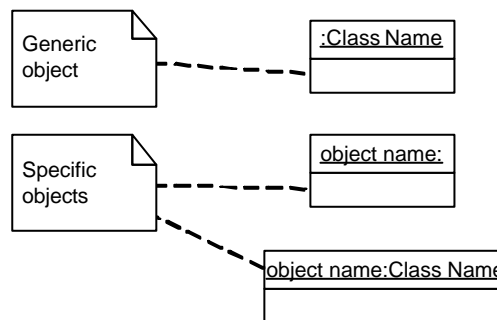


Figure 2.10 Notation for Objects

2.3.3 UML Interaction Diagrams: Sequence and Collaboration Diagrams

A sequence diagram is a graphical way to illustrate a scenario. They show the sequence in which objects pass messages to each other. They can also show how long an object takes to process a message, when objects are created and destroyed, and when objects are synchronous and asynchronous. To create sequence diagram, a set of classes that will be involved in a scenario are identified. Then the objects of these classes are listed in a line along the top of a page. The dotted lines dropped beneath each object represent the lifelines of the objects. The line on the left often represents an actor or an event outside the system boundary. An event arrow connects objects. In effect, the event arrow suggests that a message is moving between these two objects. Events are given names, which can be a phrase or an operation or message name. Constraints are listed within braces. The rectangles that lie below the object rectangles and over the dotted lines indicate when the object is active. The cross symbol at the end of the dotted line beneath an object tells that object is destroyed. Notation for UML sequence diagram can be seen in Figure 2.11.

The collaboration diagram provides a second way of showing the sequence in which events occur. As in an object diagram, objects are shown in rectangles, connected by lines. Lines connecting objects are called links. Links represent the messages flowing between objects. Each link can be labeled with one or more operation names. Each operation name is preceded by a number, which indicates the order of the operation. The arrow after or beside the operation name indicates the direction of the flow. Notation for UML collaboration diagram can be seen in Figure 2.12.

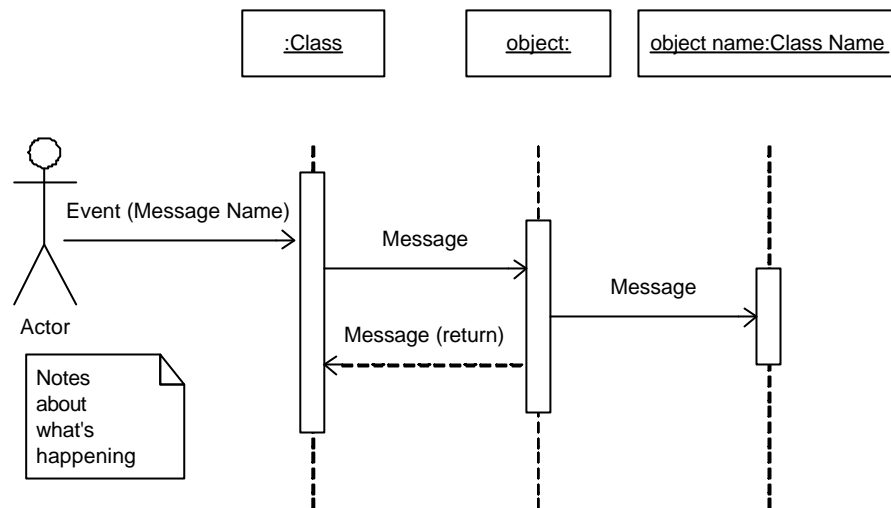


Figure 2.11 Notation for a Sequence Diagram

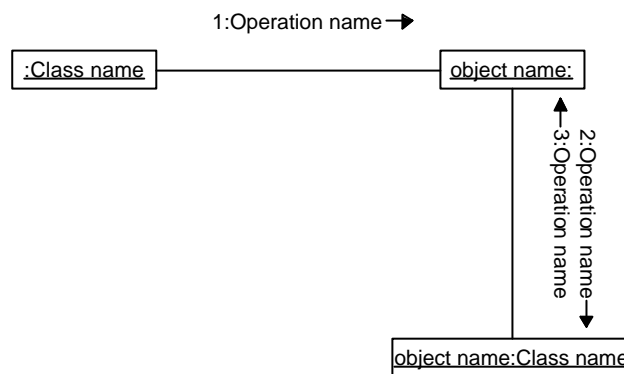


Figure 2.12 Notation for Collaboration Diagram

2.3.4 UML State Diagrams: State and Activity Diagrams

“State” refers to the set of values that describe an object at a specific moment in time. In other words the state of an object is determined by the values associated with its attributes. State diagrams are designed to show how one specific object changes state as it receives and processes messages, which often change the values of attributes (or states) of that object. State diagrams are narrowly focused and good only for analyzing very specific situations. A state diagram focuses on activity within a specific class. They are used only for representing very dynamic classes.

States are represented as rectangles with rounded corners. The name of the state is placed inside the box. Also, this box can be divided into three horizontal compartments for detailed representation. In the top compartment, the name of the state is written. The state variables are listed in the middle compartment and the operations associated with the state are listed in the lower compartment. The arrows connecting the states represent the events. They indicate the new state that results from the transition in a value. An event occurs at an instant in time when a value is changed.

A black dot used to indicate where the system starts in some cases. The first state after the start is called initial state if a starting point is indicated. A black dot inside a circle is used to signify when the state transitions end. Layered state diagrams with one group of states placed within another can be used when things become more complex. Notation for UML state diagram can be seen in figure 2.13.

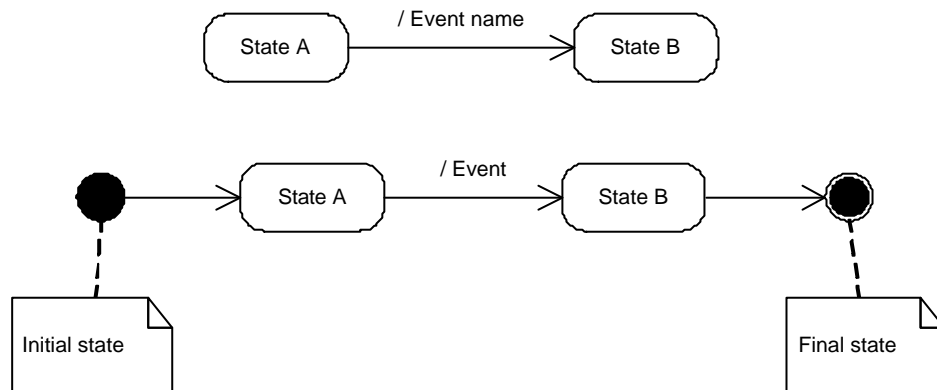


Figure 2.13 Notation for State Diagrams

Activity diagrams are used to model the flow of activities in a procedure. They provide very broad views of business processes or they can be used to break out the activities that occur within a use case. In UML, activities are called states, so activity diagrams are associated with state diagrams. Activity diagrams are just the opposite of the state diagrams. Unlike state diagrams that focus on the events occurring within a single object as it responds to messages, an activity diagram

can be used to model an entire business process or to provide a high level view of what is going inside a use case. Events are referred to as transitions. Diamonds are used to allow the developer to show what happens when a transition can have more than one consequence. Bars are used to show when several streams of events must come together before the next activity can occur. State symbols can be used to show the start and finish points. Notation for UML activity diagram can be seen in Figure 2.14.

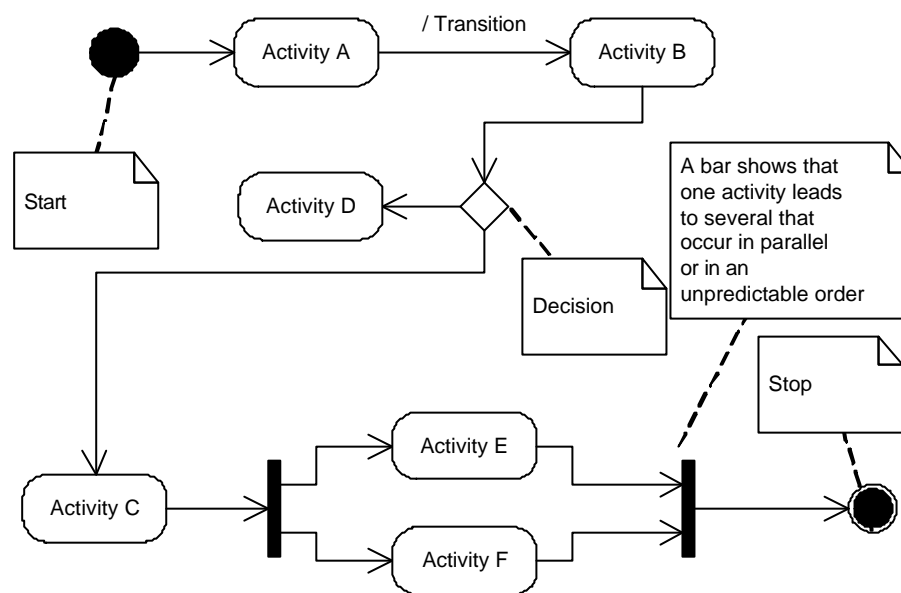


Figure 2.14 Notation for Activity Diagrams

2.3.5 UML Implementation Diagrams: Package, Component, and Deployment Diagrams

Package diagrams allow developers to show how classes can be divided into logical software modules, which are called “packages” in UML. A file folder icon is used to represent a package. Links between packages are indicated as dashed arrows. The arrows can be labeled to show the dependency names. Packages can be nested to any desired depth. The name of the package is written inside package unless the package is a super-package, which has sub-packages. In

such cases, the name of the package is written on the tab of the folder. Notation for UML package diagram can be seen in Figure 2.15.



Figure 2.15 Notation for Package Diagrams

Component diagrams refer to physical components or physical modules of the application. Logical grouping done by package diagrams can be the first approximation of what will eventually turn into a physical grouping. A component is represented by the icon shown in Figure 13.5. The names of the components are written on the icon. Dependencies between components are indicated by dotted arrows. Component diagrams can be used with deployment diagrams to show how physical modules of code are distributed on various hardware platforms. Notation for UML component diagram can be seen in Figure 2.16.



Figure 2.16 Notation for Component Diagrams

Deployment diagrams allow the developer to model the physical platforms and network connections that will be used in the application. In deployment diagrams, each node or processing element in the system is represented by a three dimensional box. Links between the nodes or platforms are shown by solid lines. Components can be placed inside the nodes to show how actual code will be distributed. A class level or an object level configuration, in which the specific instances of the node are referred, can be described. Notation for UML deployment diagram can be seen in Figure 2.17.

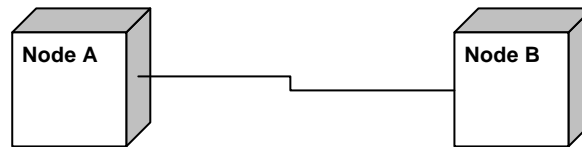


Figure 2.17 Notation for Deployment Diagrams

2.4 Entity-Relation Diagrams

An entity-relation model (or E-R model) is a detailed, logical representation of the data for an organization or for a business area. The E-R model is expressed in terms of entities in the business environment, the relationships among the entities, and the attributes of both the entities and relationships.

An entity is a person, place, object, event, or concept in the user environment about which the organization wishes to maintain data. An entity has its own identity, which distinguishes it from the other entities. These entities are connected to each other by relationships.

E-R diagrams are excellent tools for planning and designing database, especially when they are used with normalization. E-R diagram gives the analyst a clear, high-level view of data. E-R model implies the processes, but it does not clarify the process itself [17, 18].

2.5 Client/Server Architecture and Terms

Client/Server describes physical deployment models where the client computer makes a request to the server computer and the server computer services or responds to the request.

Tiers are a logical concept. The three tiers are generally described as user (first), business (second or middle), and data (third) service tiers. A service is a unit of application logic that implements operations, functions, or transformations

that are applied to object. The concept of tiers emphasizes segmenting applications into the three types of services, and is not about implementing the services nor about the number of physical computers involved in deploying the solution [19].

2.5.1 Two-Tier Client/Server Architecture

Two-tier applications represent a first step in separating database-specific access logic out of the client application and placing it on the server. This is done by implementing services as stored procedures that execute on the server against a database. This makes maintenance, upgrades, and general administration of the code easier because the code only exists on the server and not on every client. Also, the database management system (DBMS) provides the centralized functionality required when supporting many users. Transaction processing and connection handling are the major problems of the architecture. A diagram of two-tier client/server architecture is shown in Figure 2.18 [19].

2.5.2 Three-Tier Client/Server Architecture

Three-tier client/server architecture provides additional flexibility for building applications as they grow in complexity. User services reside on client computers, while business and data services generally reside on server computers that provide centralized, multi user functionality. Its working mechanism is that, the client talks to the middle tier and the middle tier uses a database connection pool to manage the connection with the database. After it completes the request, it passes the data back and releases the connection that client uses. A diagram of three-tier client/server architecture is shown in Figure 2.19 [19].

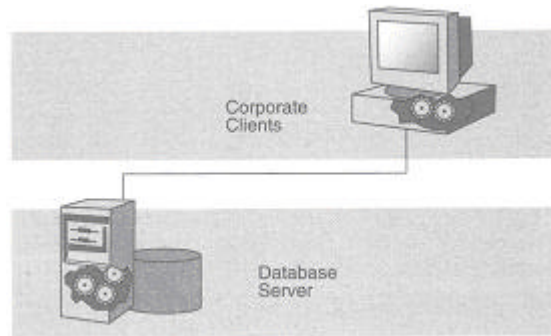


Figure 2.18 Two-Tier Client/Server Architecture [19]

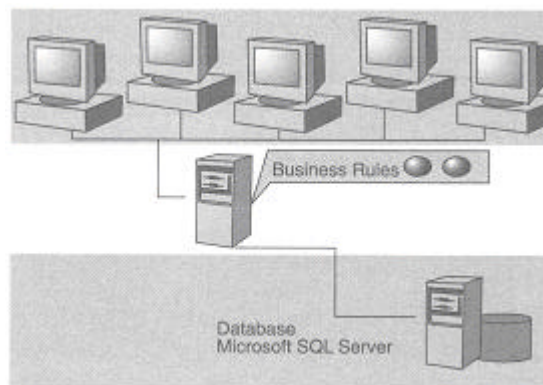


Figure 2.19 Three-Tier Client/Server Architecture [19]

2.6 Microsoft® Windows® Distributed interNet Applications (DNA) Architecture

Windows DNA is the most comprehensive, integrated platform for building and deploying applications for the Business Internet from high traffic e-commerce web sites to enterprise supply chain integration, which provides the fastest, easiest route for the Web application development [20]. Conceptual models and guidelines are designed to:

- ?? Develop reliable, scalable, multi-tier, distributed applications using the Windows DNA platform
- ?? Extend existing data and external applications to support the Internet
- ?? Support a wide range of client devices, maximizing the reach of applications

DNA Architecture is given in Figure 2.20.

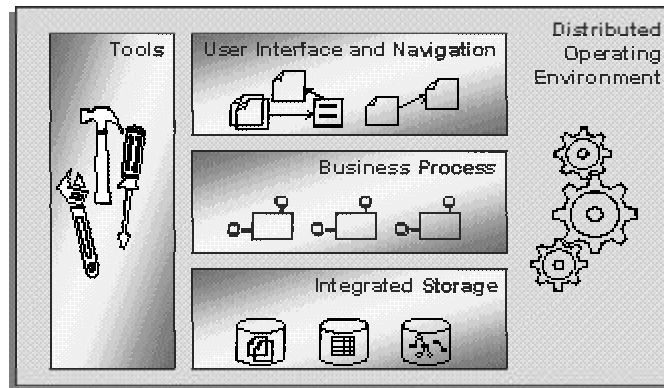


Figure 2.20 Windows DNA Architecture

Windows DNA for Manufacturing, on the other hand is a framework that allows manufacturing software applications to integrate seamlessly with one another. Microsoft announces as it allows companies to build scalable client/server systems that new and existing ERP, -manufacturing execution systems (MES), DCS- and SCADA- based applications, as well as the new breed of PC-based, shop floor control systems. Windows DNA in an integrated manufacturing environment is given in Figure 2.21 [21].

Windows DNA for Manufacturing takes the main features of Windows DNA strategic framework and applies it to manufacturing solutions. These features are common operating system and network, common data access, BackOffice-based tools and services, common application infrastructure, integration with office applications, increased enterprise integration, integrated development environment and tools and methodologies for integration of legacy systems.

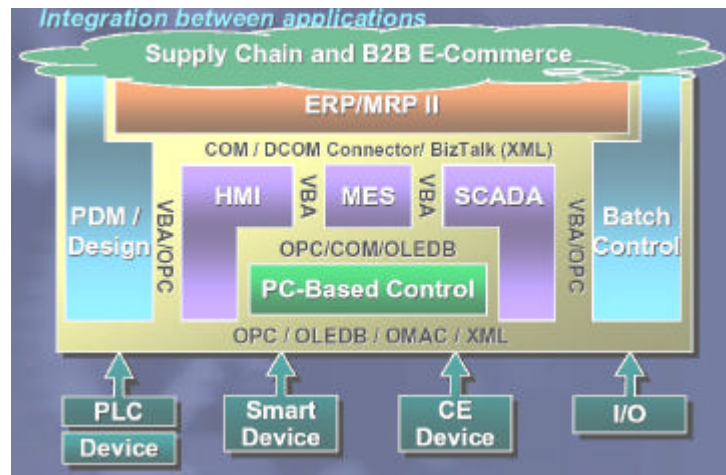


Figure 2.21 Windows DNA for Manufacturing [21]

There are series of products and technologies of Windows DNA for manufacturing like COM/DCOM, COM Services, OLEDB, Microsoft SQL Server, Internet Information Server, Microsoft Internet Explorer, Visual Studio and Exchange Server. COM/DCOM, Microsoft Transaction Server as a COM Service, Microsoft SQL Server, Internet Information Server and Visual Studio will be mentioned in the following sections.

2.6.1 Component Object Model (COM)/Distributed Component Object Model (DCOM)

The Component Object Model (COM) is an object-based programming model designed to promote software interoperability. Internet and traditional client/server application development can be integrated through the COM, which is the way of Microsoft to create components. Component is a discrete piece of software that does some specific predefined work. Through COM, client/server applications take advantage of features such as Dynamic HTML (DHTML); scriptlets; transactions; message queuing; security; and directory, database and data access. COM allows solutions to be assembled from reusable software components rather than being built from scratch, so that applications can be built and tested more efficiently. COM offers five main advantages:

- ?? **Binary compatibility:** COM components can be implemented in a number of different programming languages and used from clients that are written using completely different programming languages.
- ?? **Cross-platform interoperability:** Developer need to write COM components for a specific platform, such as Windows NT or UNIX. A COM component written for a specific platform will not run on another platform, but can communicate with COM objects running on other platforms.
- ?? **Location transparency:** Location transparency means that a component can be called by a client in one way.
- ?? **Code reusability:** COM components contain classes that expose group of methods, known as interfaces, through which clients interact with objects. Since these interfaces are documented, code that implements those interfaces can be reused by many clients.
- ?? **Version control:** Interfaces implemented by classes are static and, once defined, cannot change. New functionality can be added to a COM component by adding a new interface to a class while still supporting the old interfaces. Functionality is not lost when COM components are upgraded.

The Distributed Component Object Model (DCOM) is a COM component that runs on a different computer than the computer that uses it. DCOM enables software components to communicate directly over network boundaries. DCOM enables components in distributed applications to communicate with other components whether they are remote or running on the local computer and access distributed Windows platform services. The illustration in Figure 2.22 shows how the Internet and three-tier client/server applications are combined [19].

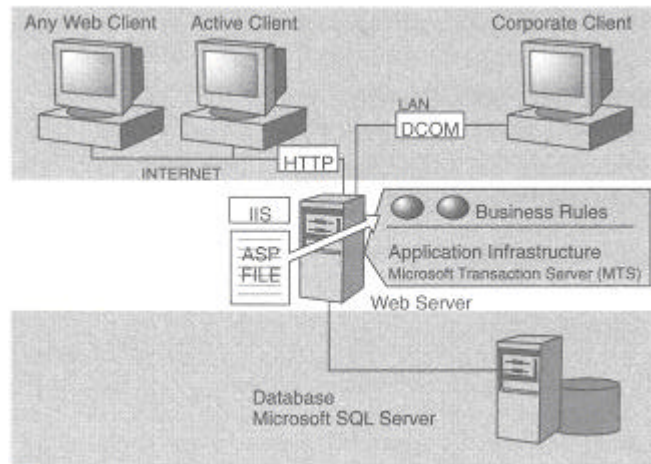


Figure 2.22 Architecture of Microsoft's Enterprise Development Strategy [19]

2.6.2 Microsoft Transaction Server (MTS)

Microsoft Transaction Server (MTS) is a component-based transaction processing system for building, deploying, and administering server applications.

Transactions provide an all-or-nothing simple model of managing work. Either all the objects succeed and all the work is committed, or one or more of the objects fail and none of the work is committed. Based on the outcome, any database tables or files affected by the work will either all be changed, or not changed at all. They will not be left in an inconsistent state. MTS provides this transaction management service. When components are activated, transactions for them are automatically created. Also, cleanup and rollback of a failed transaction are automatically handled by MTS. Objects that participate in transactions must be stateless. State is object data that is kept over more than one method call to the object.

MTS helps conserve server memory by keeping an instance of an object alive only when a client is calling the object. This is known as just-in-time activation, and it allows the server to handle more clients than is possible when the object remains active.

The synchronization issues about the simultaneous calls coming from multiple users in a distributed application environment are solved by MTS through activities, which provide a concurrency model. An activity is the path of execution that occurs from the time a client creates an MTS object until the client releases that object. During the activity the client makes calls to the MTS object. MTS object may create additional MTS objects that are part of the same activity to service these calls [19]. MTS also provides security by allowing define roles.

2.6.3 Microsoft SQL Server

Microsoft SQL server is client/server relational database system. It supports traditional two tier applications and state-of-the-art three tier applications. It is a relational database consisting of multiple tables of data; each of that is further divided into rows (records) and columns (fields). The main features of Microsoft SQL Server are as following:

- ?? It runs on Windows NT, 2000, and 9x environments.
- ?? It is scalable; it can be used for small databases on PC's to terabytes of databases on up to 32CPU machines.
- ?? It has replication ability.
- ?? It gives On-Line Analytical Processing (OLAP) services.
- ?? It has English Query facility, which enables easiest way of constructing queries.
- ?? Data transformation systems enable easy exchange of data.
- ?? It supports distributed transactions.
- ?? It supports ANSI-92 SQL.
- ?? It has centralized management tools for multi server environments [22].

2.6.4 Internet Information Server

Internet Information Server is the web server for Windows NT/2000 Server. Web servers are the softwares that allow clients access to corporate

information and data by facilitating information publishing, application execution and data retrieval.

As Java applets; ActiveX controls and database-access tools were introduced; web servers have become more powerful for publishing dynamic information. They now have powerful capabilities such as connectivity to databases, dynamic page creation, database queries and data encryption. Some of the features of IIS are:

- ?? Web based administration.
- ?? Hypertext Transport Protocol (Http) 1.1 support.
- ?? Content Control.
- ?? Configuration Backup.
- ?? Database Components.
- ?? Per-site Bandwidth Throttling.
- ?? Per-site Performance Monitoring.
- ?? Indexing and Searching.

Microsoft Active Server Page (ASP) is the server side-scripting platform of Microsoft that one can use to create and run dynamic, interactive Web server applications. IIS is the web server with which enables ASP in Microsoft environment.

2.6.5 Visual Studio

Microsoft Visual Studio is a powerful, comprehensive, and integrated suite of developer tools for rapidly building enterprise Web applications on the Windows platform based on the Windows DNA architecture [20]. Visual Studio 6.0 addresses all aspects of Windows DNA application development, including integrated tools for multiple tier application design, user interface development, middle-tier component development and assembly, database programming and design, performance analysis, and team-based development support. Primary

development tools in this product set are Visual Basic, Visual C++, Visual J++, Visual InterDev and Visual FoxPro.

Visual Basic 6.0 provides the most productive development through Rapid Application Development (RAD) functionality and enhanced data connectivity. It can be used in either component development or in combining various components to build the entire application. Visual C++ is the most productive C++ tool for optimized, highest-performance, enterprise-wide development. Visual InterDev is a comprehensive development tool for data-driven web applications. Visual J++ allows Java developers access to the full capabilities of the Windows platform. Visual FoxPro gives the necessary tools to manage data. It can be for organizing tables of information and running queries, creating an integrated relational database management system or programming a fully developed data management application for end users [19].

CHAPTER 3

SYSTEM DEVELOPMENT

3.1 Overview

The method that is used to develop the tool management system will be presented in this section. Object-oriented approach was used during the system development, which is discussed in Chapter 2. UML was used as a standard modeling language of object-oriented system development. UML diagrams used in the phases of development can be seen in Figure 3.1. In the following sections, one example for each UML diagram is examined. Other diagrams can be found in Appendix A.

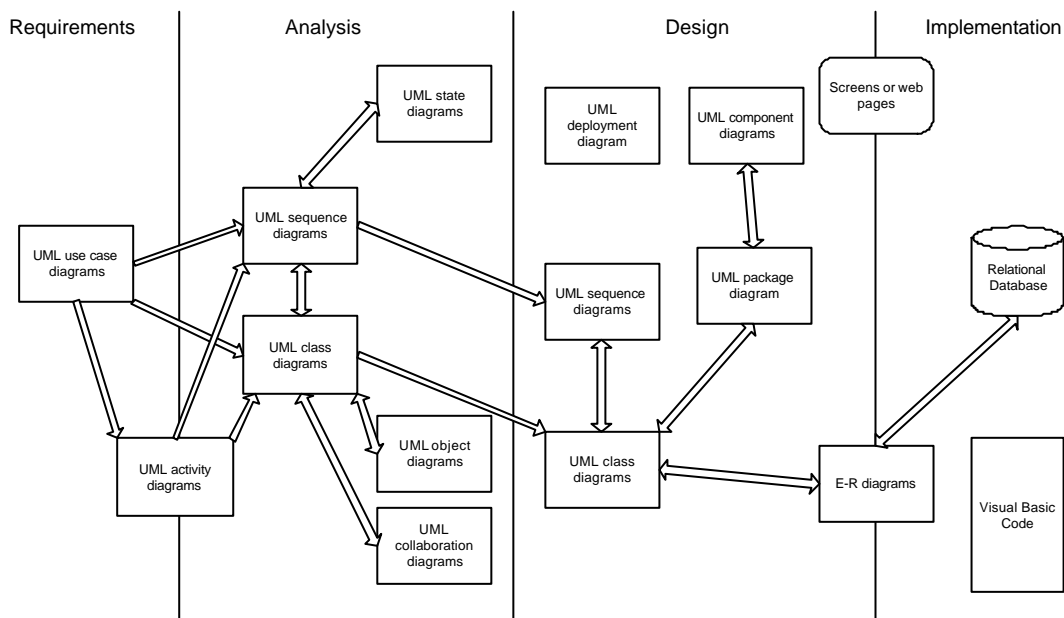


Figure 3.1 Development Road Map

First, the scope and the requirements for the system to be developed are defined. What the system is supposed to do and who will use it is decided. This is the overview of the entire system. During analysis, developer focuses on designing an ideal or logical system that satisfies the requirements without being overly concerned about implementation. During design, the work done is modified and a concrete physical system appears. During the coding phase, the code is written. Then the initial prototype is tested. After a satisfying testing, first iteration in system development ends and the second iteration starts with the analysis phase again. This goes on like that until the entire application is completed.

3.2 Requirement Analysis

Requirement analysis begins with use case analysis. Tool management system was divided into sections and for each section a use case diagram was drawn. Use cases defining the requirements of this section of the system and the actors interacting with them were stated. The interactions were modeled according to the authorization of the actors or users in the system. Use case diagram of Cutting Tool-Tool Holder module can be seen in Figure 3.2. The system is framed by a rectangle and name of the system is written top of the frame, which is “Cutting Tool-Tool Holder Module”. The use cases are: search tool data, modify tool data, issue cutting tool and tool holder, record cutting tool and tool holder return, sharpen used cutting tools. The actors are: Buyer, manufacturing engineer, tool crib employee and sharpening room employee. The interactions of the actors with the use cases are represented with a line between them.

Then the scenarios in the system were modeled using the activity diagrams. Flows of activities in some use cases were modeled to clarify these use cases. Activity diagram is very similar to the workflow diagram. Activity diagram of “Issue Cutting Tool” use case can be seen in Figure 3.3. The issue of a cutting tool starts when a worker comes with tool request. Tool crib employee gets the worker badge number and the tool request. If there exists a used cutting tool for

the tool requested, used one is issued and used cutting tool inventory is updated. If there is no used one, new cutting tool is issued and its inventory is updated.

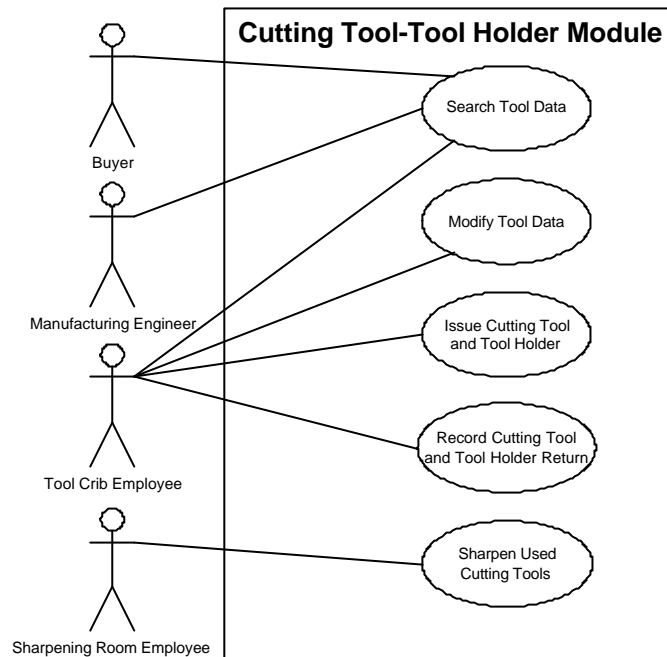


Figure 3.2 Use Case Diagram of Cutting Tool-Tool Holder Module

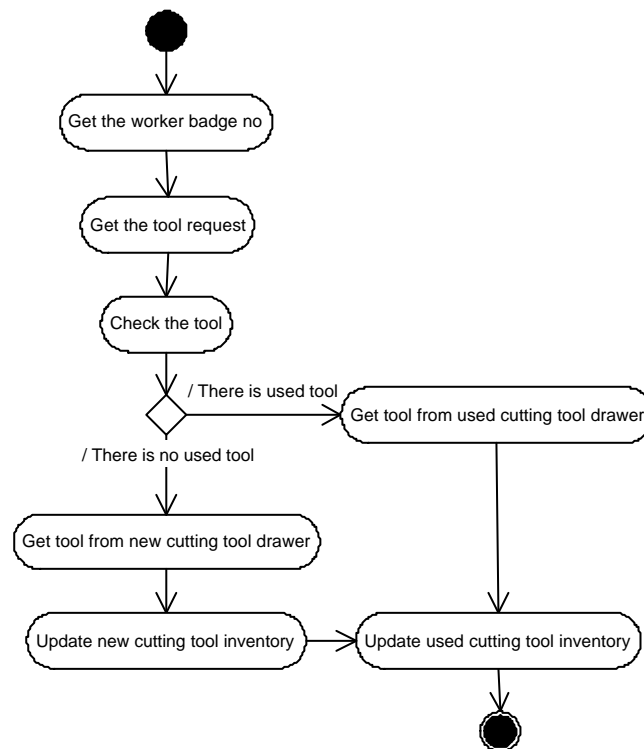


Figure 3.3 Activity Diagram of “Issue Cutting Tool” Use Case

3.3 Analysis and Design Phases

After defining the requirements and scenarios, class and sequence diagrams were drawn. In the class diagram, all the classes are stateless. Only methods of the classes are specified. The classes in the system are: RoleValidator, Calendar, Tool, BOT, Part, Schedule, MeasuringEquipment, CalibrationList, Fixture, FixtureAssembly, Assembly, ToolHolder, ToolHolderLocation, THIssue, THIssueItem, THReturn, THReturnItem, CuttingTool, CuttingToolLocation, RequirementPlan, PurchaseOrder, PurchaseOrderItem, Receiving, ReceivingItem, CTIssue, CTIssueItem, UCT, SendingToSR, SendingToSRItem, ReturnFromSR, ReturnFromSRItem, CTReturn, CTReturnItem, SRCuttingTool, Sharpening, SharpeningItem. BOT is the abbreviation of bill of tooling and UCT is the abbreviation of used cutting tool. CT, TH and SR prefixes are used for cutting tool, tool holder and sharpening room, respectively. The relationships between the classes are:

- ?? Tool is used according to BOT.
- ?? BOT has Part.
- ?? Part has Schedule.
- ?? MeasuringEquipment is a type of Tool.
- ?? MeasuringEquipment has CalibrationList.
- ?? Fixture is a type of Tool.
- ?? FixtureAssembly is a type of Assembly.
- ?? FixtureAssemblyItem is a part of FixtureAssembly
- ?? Fixture has FixtureAssemblyItem.
- ?? ToolHolder is a type of Tool.
- ?? ToolHolder has ToolHolderLocation.
- ?? ToolHolder is issued by THIssueItem.
- ?? THIssueItem is a part of THIssue.
- ?? ToolHolder is returned by THReturnItem.
- ?? THReturnItem is a part of THReturn.

- ?? CuttingTool is a type of Tool.
- ?? CuttingTool has CuttingToolLocation.
- ?? CuttingTool has Requirement.
- ?? CuttingTool is purchased according to PurchaseOrderItem.
- ?? PurchaseOrderItem is a part of PurchaseOrder.
- ?? CuttingTool is received from vendor by ReceivingItem.
- ?? ReceivingItem is a part of Receiving.
- ?? CuttingTool is issued by CTIssueItem.
- ?? CTIssueItem is a part of CTIssue.
- ?? UCT is a type of Tool.
- ?? UCT is sent to sharpening room by SendingToSRItem.
- ?? SendingToSRItem is a part of SendingToSR.
- ?? UCT is returned from sharpening room by ReturnFromSRItem.
- ?? ReturnFromSRItem is a part of ReturnFromSR.
- ?? UCT is returned from shop floor by CTReturnItem.
- ?? CTReturnItem is a part of CTReturn.
- ?? SRCuttingTool is a type of UCT.
- ?? SRCuttingTool is sharpened by SharpeningItem.
- ?? SharpeningItem is a part of Sharpening.

The duty of all of the classes is to communicate with the database. Methods of the classes are defined according to this duty. Data manipulation consists of creating, updating, deleting and retrieving. The method names of the classes start with C, U, D, R. C stands for create, U stands for update, D stands for delete and R stands for retrieve in this context. If data of a single item is to be retrieved, “ROne” prefix is used and if data of a list of items is to be retrieved, “RAIl” prefix is used in retrieving operations. There are some methods, which are not in this context. In RoleValidator class “RCheckLogin” method checks the user name and password of the user and “GetAuthentication” retrieves the authentication data of the user. In Requirement class, “RunRequirement” executes the requirement plan and “GetRequirementStatus” retrieves the status of the

execution. The “+” symbol preceding the names of the methods indicates that these methods are defined as public methods, which means that they can be reached by other classes. Class diagram of the tool management system is in Figure 3.4.

Next step is the sequence diagram to clarify the interactions between the objects of these classes and the sequence of events. It is not a must to define all of the classes and complete the class diagram before going through the sequence diagrams. Each sequence diagram is drawn for a use case. Developer does not have to draw the sequence diagrams for all the use cases. They are drawn to clarify the complex cases only. Issue of a cutting tool starts when the tool crib employee calls the create method (CCTIssue) of cutting tool issue object (:CCTIssue). This object calls the create method (CCTIssueItem) of cutting tool issue item object (:CCTIssueItem). If the condition of the cutting tool is “New”, ROneCuttingTool method of CuttingTool object is called and data of the cutting tool that will be issued is retrieved from the database. The data representing the inventory level of that cutting tool is updated by calling the UCuttingTool method of CuttingTool object. Then, CTIssueItem object calls the ROneUCT method of UCT object and updates the data of UCT by calling the UUCT method. If the condition of the cutting tool is “Used”, then CTIssueItem calls the ROneUCT and UUCT methods of UCT object only. Then, ROneCuttingToolLocation method of CuttingToolLocation object is called by CTIssueItem object. If there are other cutting tools of the same tool type at the same location on the shopfloor, UCuttingToolLocation method is called. If there is no cutting tool of the same tool type at that location, CCuttingToolLocation method is called. After CuttingTool, UCT and CuttingToolLocation objects finishes their work, CTIssueItem object is deactivated. After CTIssueItem is deactivated, CCTIssue object is deactivated and transaction is completed. The sequence diagram of “Issue Cutting Tool” use case can be seen in Figure 3.5.

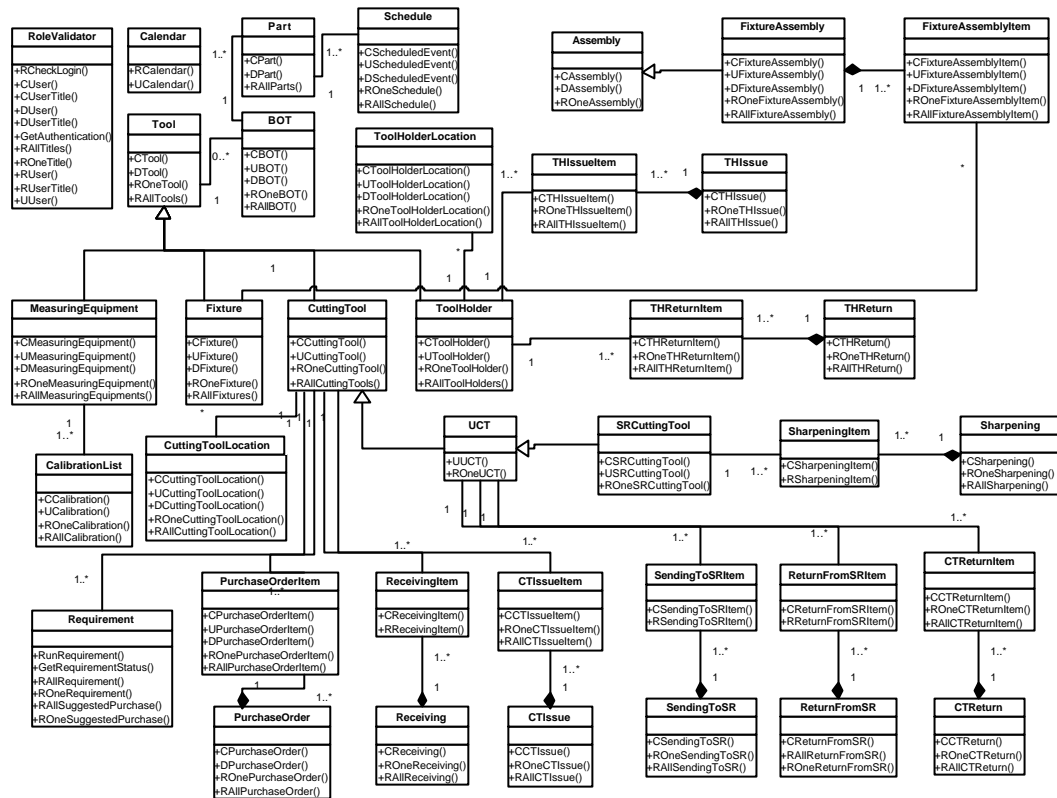


Figure 3.4 Class Diagram

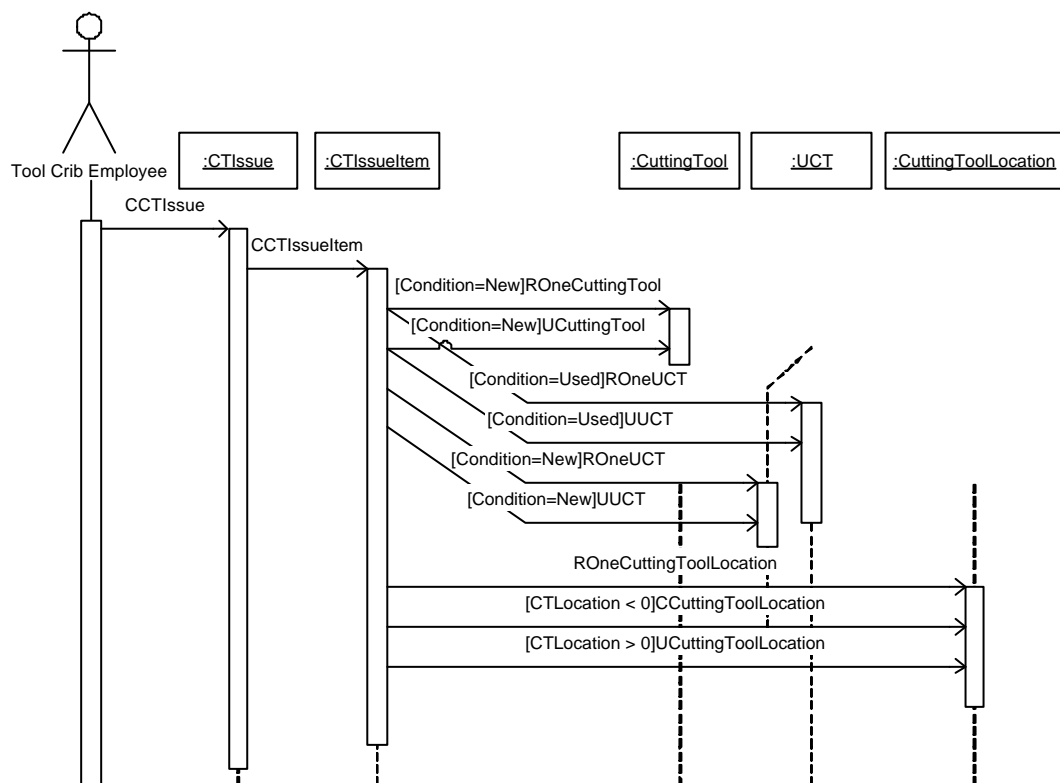


Figure 3.5 Sequence Diagram of "Issue Cutting Tool" Use Case

The collaboration diagram has exactly the same purpose of sequence diagram. They represent the interactions between objects and the sequence of events. The lines connecting the objects show the interactions between them. The numbers preceding the method calls represent the sequence of that event. Since collaboration diagrams can get complex and messy easily, they may be used to narrowly focus a scenario and analyze it in more details. A collaboration diagram can also be used in place of a simple object diagram. The collaboration diagram of “Issue Cutting Tool” use case can be seen in Figure 3.6.

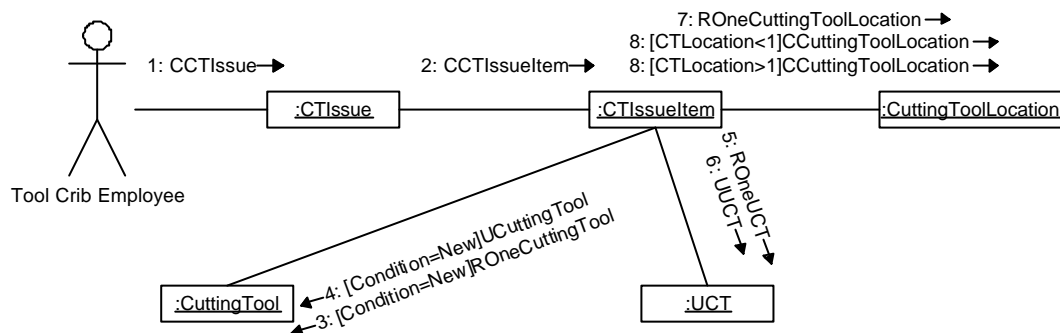


Figure 3.6 Collaboration Diagram of “Issue Cutting Tool” Use Case

Once initial class and sequence diagrams were created in the analysis phase, they were enhanced and expanded in the design phase in parallel. From the expanded class diagrams, E-R model of the data structure was drawn, which leads the design of the database later. Also package diagram was drawn to show the logical groups of classes. Package diagram can be seen in Figure 3.7. Component and deployment diagrams were drawn using the package diagrams to show the physical design of the tool management system. Component diagram in Figure 3.8 shows the projects created by Visual InterDev, which are collected in the client package. `TRPClientToolMng`, `CTTHClientToolMng` and `FGClientToolMng` projects are reached through `LoginClientToolMng`, so they are dependent to it. Component diagram in Figure 3.9 shows the projects created by Visual Basic, which are collected in the business package. None of the classes in these projects or modules are reached or called by the other ones, so none of them are dependent

to each other. Deployment diagram in Figure 3.10 shows the deployment of the system into computers. Business and client packages are deployed into one computer. MTS hosts the business and IIS hosts the client package. Database managed by Microsoft SQL Server 2000 is deployed into another computer. Clients in the intranet of the factory make requests to the business server and business server connects to the data server to manipulate the data. Therefore, client is dependent on the business server and business server is dependent on the data server.

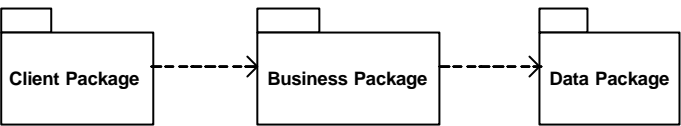


Figure 3.7 Package Diagram

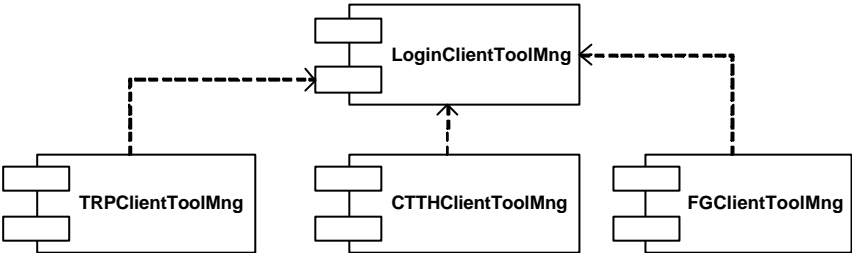


Figure 3.8 Component Diagram of Client Package

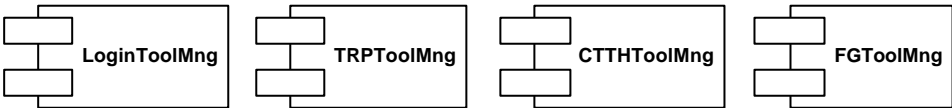


Figure 3.9 Component Diagram of Business Package

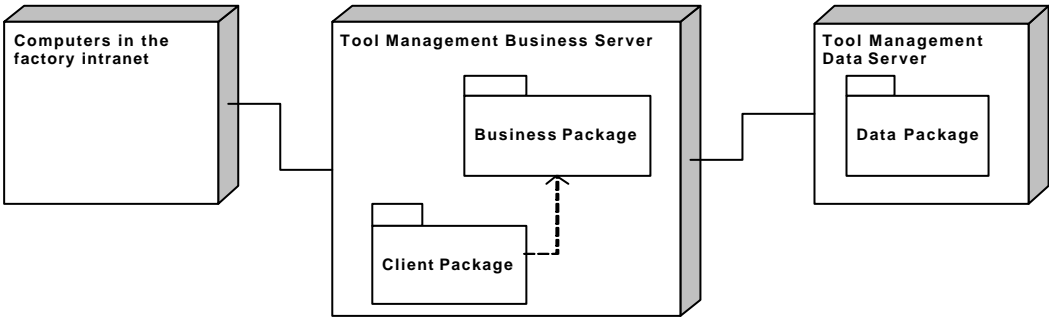


Figure 3.10 Deployment Diagram

3.3.1 E-R Model

E-R model is used to broadly represent the information structure of the system. The entities in the model and the relations between these entities are presented. The entities of the domain can be listed as follows:

Tool: It is any equipment that will be managed by tool management system.

Measuring Equipment: It is a type of tool being used by measurement purposes.

Calibration List: It is a set of record that is kept for each calibration of each measuring equipment.

Fixture: It is a type of tool being used for clamping and locating the workpiece to machine tool.

Assembly: It is an equipment formed by a group of tools.

Fixture Assembly: It is a type of assembly formed by a group of fixtures.

Tool Holder: It is a type of tool in which cutting tools are attached.

Tool Holder Location: It is the record of tool holder locations in the shop floor.

Tool Holder Issue: It is the record of issued tool holders from tool crib to shop floor.

Tool Holder Return: It is the record of returned tool holders from shop floor to tool crib.

Cutting Tool: It is a type of tool being used to cut workpiece.

Cutting Tool Location: It is the record of cutting tool locations in the shop floor.

Purchase Order: It is the record of cutting tools to be purchased.

Receiving: It is the record of cutting tools received from vendors.

Requirement Plan: It is the record of an event on a cutting tool a day.

Suggested Purchase: It is the record of purchase orders of cutting tools suggested by tool management system.

Cutting Tool Issue: It is the record of issued cutting tools from tool crib to shop floor.

UCT: It is the record of used cutting tools.

Sending To Sharpening: It is the record of used cutting tools sent to sharpening room from tool crib.

Return From Sharpening: It is the record of used cutting tools returned from sharpening room to tool crib.

Cutting Tool Return: It is the record of used cutting tools returned from shop floor to tool crib.

SRCutting Tool: It is the record of used cutting tools in the sharpening room.

Sharpening: It is the record of sharpened cutting tools.

Calendar: It is a record of days in the calendar.

TRP Log: It is the record that has the stages of tool requirement planning.

Part: It is the end product of manufacturing processes.

Schedule: It is a record of parts' manufacturing quantities and dates.

The relations between these entities are as follows:

- ?? Tool is used for one and many Parts; Part uses one and many Tool.
- ?? Part is manufactured according to one and many Schedule.
- ?? Measuring Equipment is a type of Tool.
- ?? Measuring Equipment has one and many Calibration List.
- ?? Fixture is a type of Tool.
- ?? Fixture Assembly consists of one and many Fixture.
- ?? Fixture Assembly is a type of Assembly.
- ?? Tool Holder is a type of Tool.
- ?? Tool Holder has zero and many Tool Holder Locations.
- ?? Tool Holder is issued by one and many Tool Holder Issues; Tool Holder Issue issues one and many Tool Holders.
- ?? Tool Holder is returned by one and many Tool Holder Returns; Tool Holder Return returns one and many Tool Holders.

- ?? Cutting Tool is a type of Tool.
- ?? Cutting Tool has zero and many Cutting Tool Locations.
- ?? Cutting Tool has one and many Requirement Plans.
- ?? Cutting Tool is issued by one and many Cutting Tool Issues; Cutting Tool Issue issues one and many Cutting Tools.
- ?? Cutting Tool is purchased by one and many Purchase Orders; Purchase Order purchases one and many Cutting Tools.
- ?? Cutting Tool is purchased by one and many Suggested Purchases; Suggested Purchase purchases one and many Cutting Tools.
- ?? UCT is a type of Tool.
- ?? UCT is sent to sharpening by one and many Sending To Sharpenings; Sending To Sharpening sends to sharpening one and many UCTs.
- ?? UCT is returned from sharpening by one and many Return From Sharpenings; Return From Sharpening returns from sharpening one and many UCTs.
- ?? UCT is returned by one and many Cutting Tool Returns; Cutting Tool Return returns one and many Cutting Tools.
- ?? SRCuttingTool is a type of UCT.
- ?? SRCuttingTool is sharpened by one and many Sharpenings; Sharpening sharpens one and many SRCuttingTools.

The E-R model of tool management system is given in Figure 3.11.

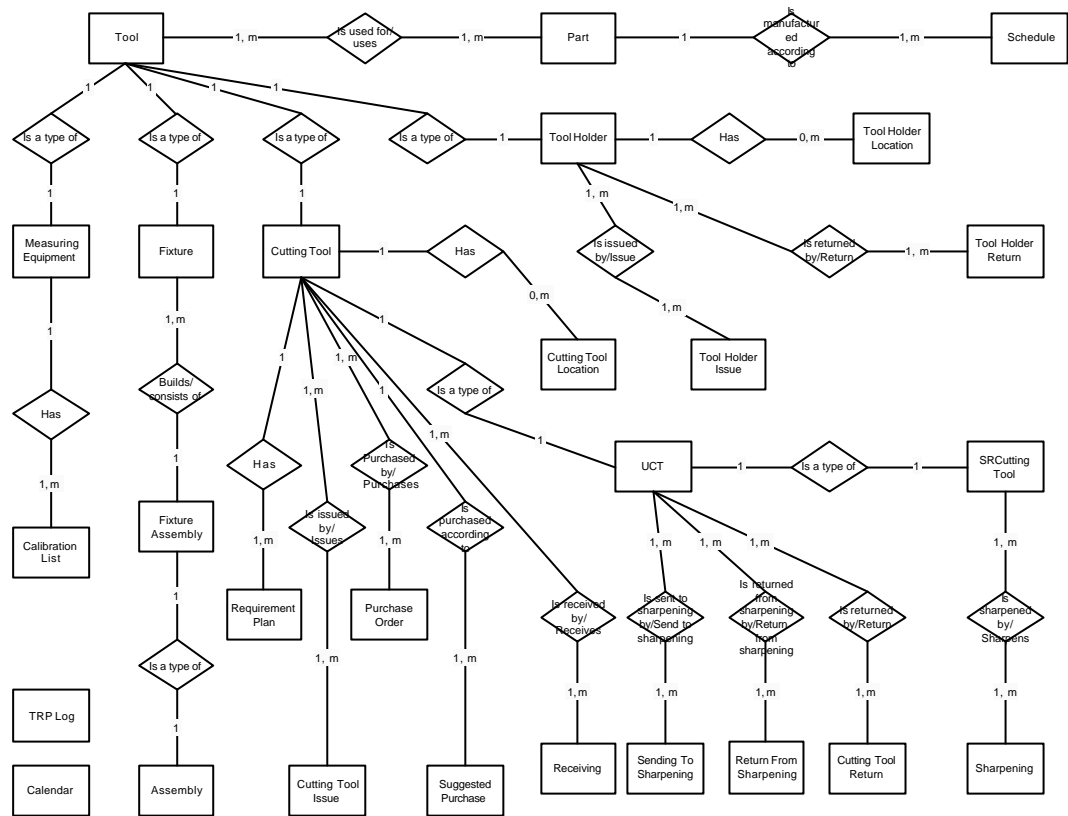


Figure 3.11 E-R Model of Tool Management System

3.3.2 Authentication Structure

The authentication structure of the system is built according to the UML use case models. The actors and their interaction between the use cases define the authorization of the users. The processes that can be done using the system and the titles that are authorized to carry out these processes are defined using the UML use case diagram. Each user is given a title and is authorized according to this title which has a previously drawn authorization border. First, the E-R diagram of the system is drawn. The entities and their relationships between them are modeled. The E-R model of the system authentication is given in Figure 3.12.

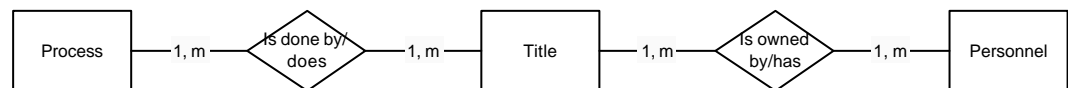


Figure 3.12 E-R Model of Authentication Structure

The entities of the model are as follows:

Process: It is a process or operation done by the users of the tool management system.

Title: It is the title of the users which has previously defined authorization borders.

Personnel: It is a user of the tool management system.

The relationships between these entities are as follows:

- ?? Process is done by one and many Title; Title does one and many Process.
- ?? Title is owned by one and many Personnel; Personnel has one and many Title.

After drawing the E-R diagram of the authentication structure, attributes were defined, key attributes were selected, many to many relations were resolved and normalization process was done on the data structure to obtain the third normalized form. The end product was the database structure, given in Figure 3.13. As seen in Figure 3.13, many to many relationships are resolved using the Title_Process and Personnel_Title tables.

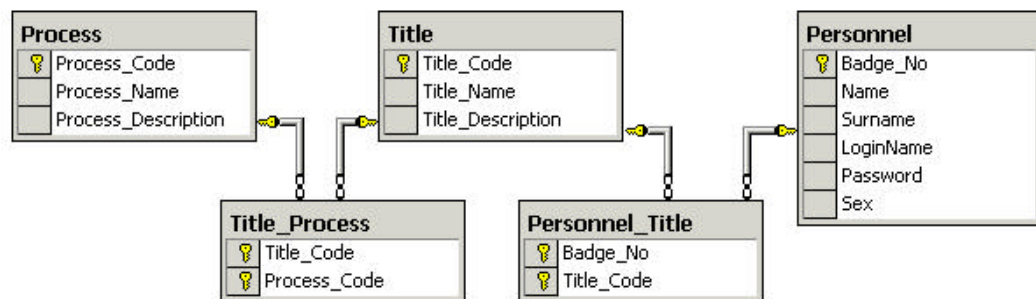


Figure 3.13 Database Diagram of Authentication Structure

The attributes in the tables of this database are as follows:

Process

Process_Code (Primary Key): It is a number that uniquely identifies a process in the tool management system.

Process_Name: It is the name of the process.

Process_Description: It is the description of the process telling what it does.

Title

Title_Code (Primary Key): It is a number that identifies a title of users.

Title_Name: It is the name of the title.

Title_Description: It is the description of the title.

Title_Process

Title_Code (Foreign Key):

Process_Code (Foreign Key):

Personnel

Badge_No (Primary Key): It is a number that uniquely identifies each user of the tool management system.

Name: It is the name of a user.

Surname: It is the surname of a user.

Login_Name: It is the login name of a user which is unique among other users.

Password: It is the password of a user which is needed to login the system.

Sex: It is the sex of the user.

Personnel_Title

Badge_No (Foreign Key):

Title_Code (Foreign Key):

3.3.3 Web Site Diagram

The pages of the system and the navigation between these pages are defined. The module boundaries are shown with dash lines and the navigations between these modules are shown with lines.

System is entered through the login page. As the user supplies his/her user name and password, he/she comes up with the system root upon passing the authentication. System screens are made up of three frames. These frames are created in the “Default” page. The frame in the left-hand side of the screen hosts the options of the tool management system. Also, user can log off from this frame. The frame in the bottom of the screen hosts the modules of the system whose links were created by “Modules” page. User can navigate between the modules and reach options of the modules from this frame. Main frame opens with the “Welcome” page and hosts the interface of the operations done.

The links of the options of the modules are created by “RootFG”, -which links to gage-fixtured module-, “RootCTTH”, -which links to cutting tool-tool holder module-, and “RootTRP”, -which links to tool requirement plan module-, pages.

There are 16 pages in gage-fixtured module, 27 pages in cutting tool-tool holder module and 20 pages in tool requirement plan module. The pages, which have the prefixes “List” and “Single”, can only be reached through the pages they are linked to. All the other pages can be reached directly from the options links of the modules appearing in the left-hand side frame of the screen.

Web site diagram can be seen in Figure 3.14.

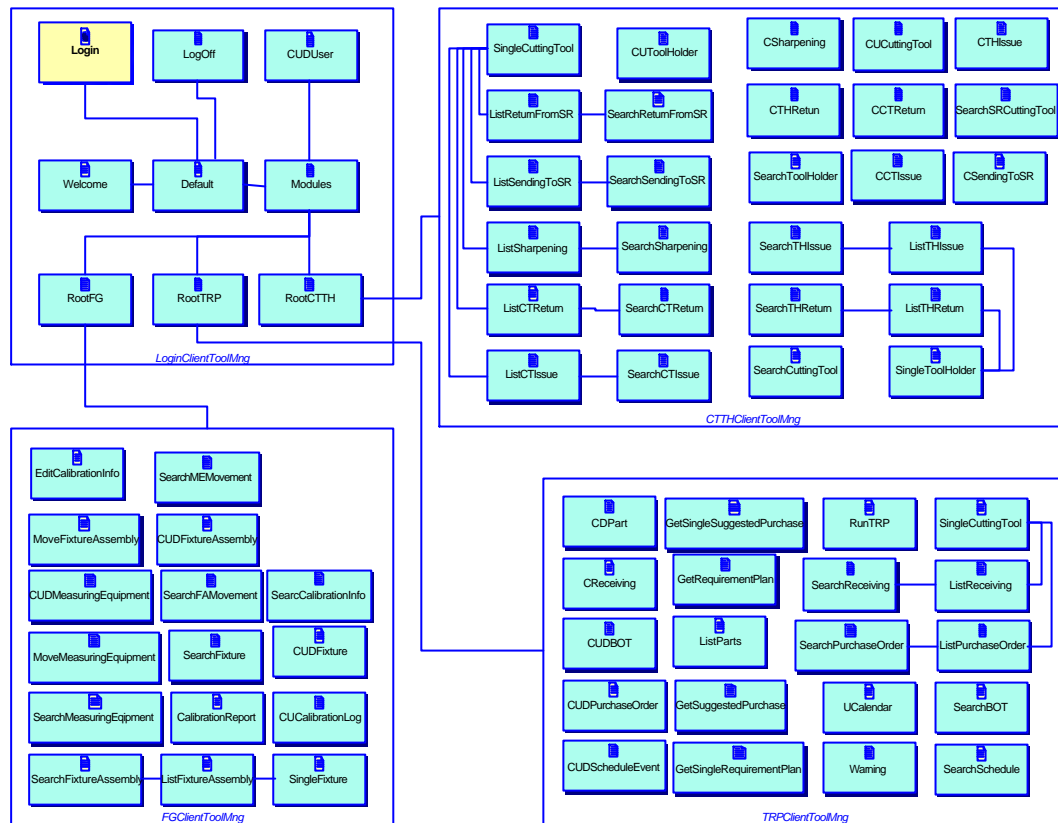


Figure 3.14 Web Site Diagram

3.4 System Implementation

The implementation history of the tool management system will be given in three subsections. The implementation of data tier, business tier and client tier will be presented, respectively.

The physical layout of the system includes one application server, one data server and other computers connected to the network and used as clients. Microsoft Transaction Server (MTS), which is the object broker of the system and Internet Information Server (IIS), which is the web server of the system, are installed on application server. Microsoft SQL Server 2000, which manages the database, is installed on data server. Although other orientations like installing MTS and IIS on separate computers or installing MTS, IIS and Microsoft SQL Server on the same computer, this layout is suggested

3.4.1 Data Tier Implementation

Databases are created using Enterprise Manager of Microsoft SQL Server 2000, which is the database management system. These databases are filled with tables by the help of the E-R diagram developed in the design phase. Columns and their data types are entered. Key attributes of the tables were defined. The relations between these tables were created and the stored procedures are written by using ANSI Structured Query Language (SQL). After the addition of stored procedures into the databases, data tier construction was completed.

Two databases were created for the system and named as ToolManagement and ToolManagementUsers for the tool management system and for the authorization, respectively. There are 40 user-defined tables in ToolManagement database and 5 user-defined tables in ToolManagementUsers database. There are 230 stored procedures in ToolManagement database and 13 stored procedures in ToolManagementUsers database. “spCPart” stored procedure which creates a part in “Part” table is as follows:

```
Create Procedure spCPart
@vPartID as nvarchar(50),
@vPartDesc as nvarchar(150),
As
Insert Into Part (Part_ID, Part_Description)
Values (@vPartID, @vPartDesc)
return
```

@vPartID and @vPartDesc are the parameters of this stored procedure which must be passed by the business tier to the data tier to call this stored procedure. “@” symbol states that the parameter is local. “v” letter is put in front of the name of the parameter just for preventing confusion in the Visual Basic codes. These parameters are inserted into the columns “Part_ID” and

“Part_Description” of the “Part” table. Database diagram of tool management system can be seen in Figure 3.15.

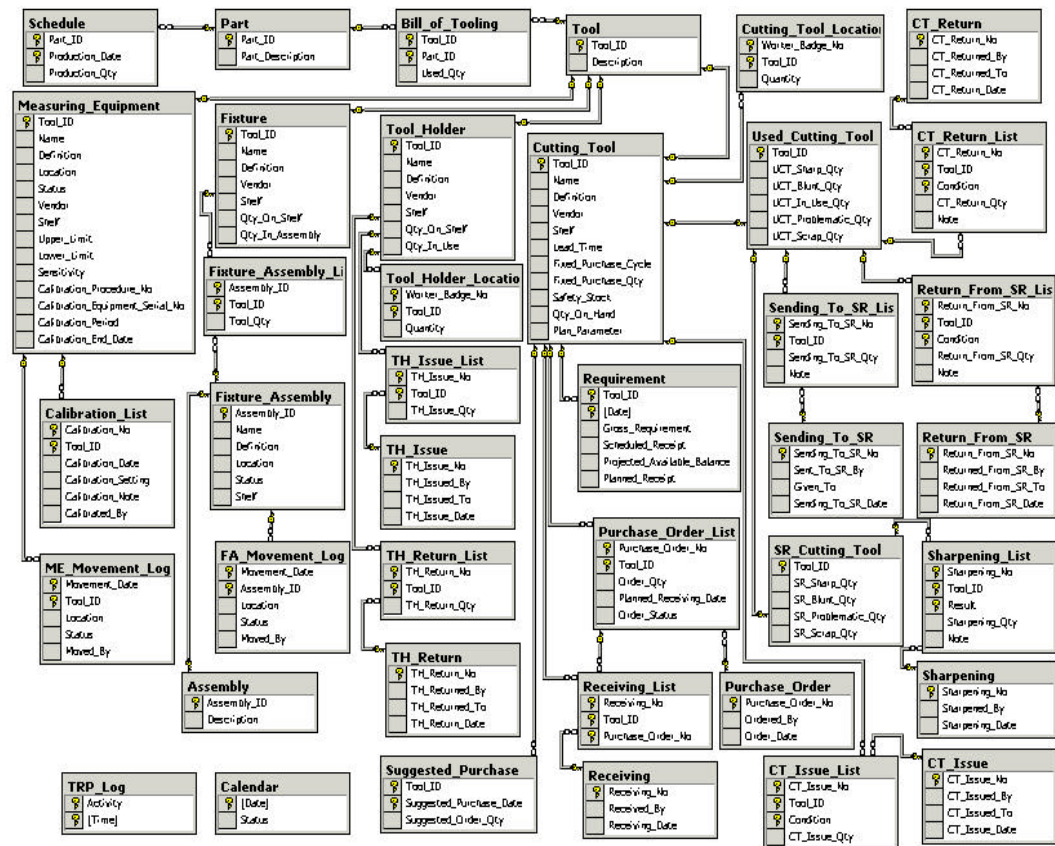


Figure 3.15 Tool Management Database Diagram

3.4.2 Business Tier Implementation

First the business objects were created using Visual Basic 6.0 Integrated Development Environment (IDE). These objects are then deployed into Microsoft Transaction Server.

ActiveX technology was used to develop the business tier. Four different ActiveX DLL (Dynamic Link Library) projects were developed; one for Gage-Fixture (FGToolMng) module, one for Cutting Tool-Tool Holder (CTHToolMng) module, one for Tool Requirement Plan (TRPToolMng)

module, and one for user management (LoginToolMng). TRPToolMng module in the design view of Visual Basic 6.0 IDE can be seen in Figure 3.16.

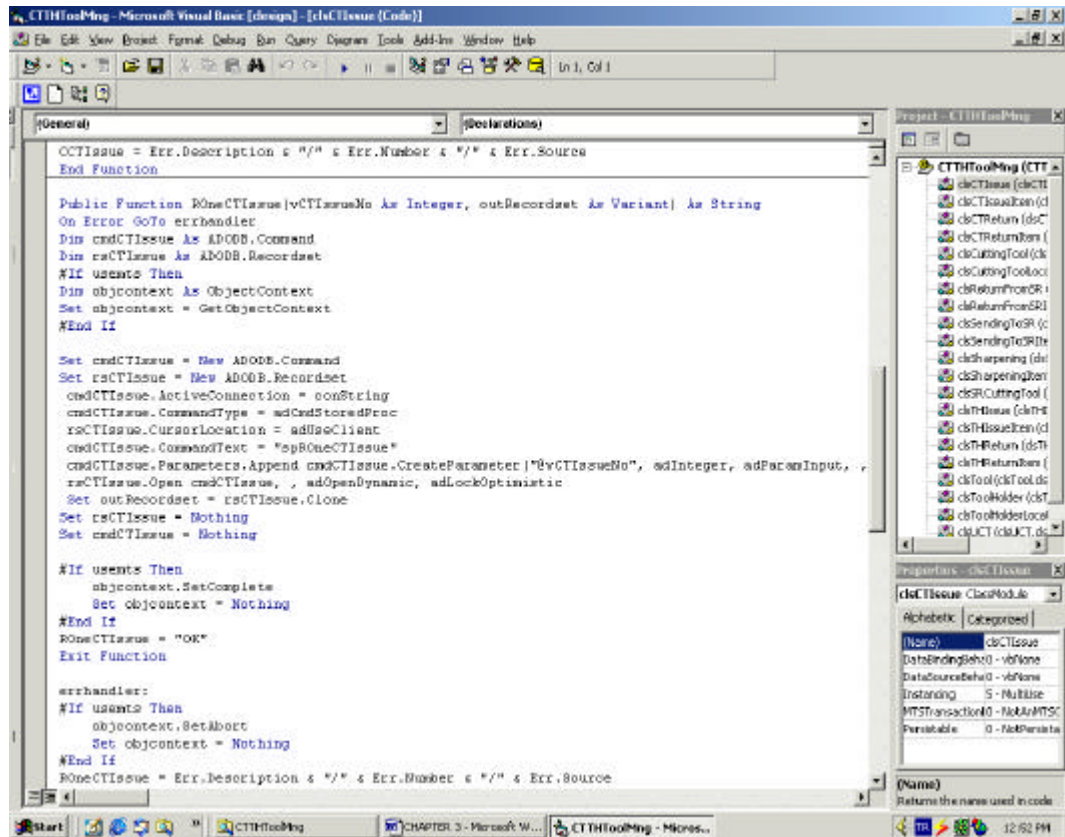


Figure 3.16 Design View of CTHToolMng Module in Visual Basic 6.0 IDE

The classes in the modules are as follows: clsAssembly, clsCalibrationList, clsFAMovementLog, clsFixture, clsFixtureAssembly, clsFixtureAssemblyItem, clsMeasuringEquipment, clsMEMovementLog, clsTool in FGToolMng ActiveX dll; clsCTIssue, clsCTIssueItem, clsCTReturn, clsCTReturnItem, clsCuttingTool, clsCuttingLocation, clsReturnFromSR, clsReturnFromSRItem, clsSendingToSR, clsSendingToSRItem, clsSharpening, clsSharpeningItem, clsSRCuttingTool, clsTHIssue, clsTHIssueItem, clsTHReturn, clsTHReturnItem, clsTool, clsToolHolder, clsToolHolderLocation, clsUCT in CTHToolMng ActiveX dll; clsBOT, clsCalendar, clsCuttingTool, clsPart, clsPurchaseOrder, clsPurchaseOrderItem, clsReceiving, clsReceivingItem, clsRequirement, clsSchedule in TRPToolMng ActiveX dll; clsRoleValidator in LoginToolMng

ActiveX dll projects. The classes are created in ActiveX dll projects. The methods of the classes are defined to be public inside the code. The classes are stateless since all of them are deployed in MTS in which the objects must be stateless. The Visual Basic code of “CPart” function of “clsPart” class can be seen below:

```
Public Function CPart(vPartID As String, vPartDescription As String) As String
```

```
On Error GoTo errhandler
```

```
Dim cmdPart As ADODB.Command
```

```
#If usemts Then
```

```
Dim objcontext As ObjectContext
```

```
Set objcontext = GetObjectContext
```

```
#End If
```

```
Set cmdPart = New ADODB.Command
```

```
cmdPart.ActiveConnection = conString
```

```
cmdPart.CommandType = adCmdStoredProc
```

```
cmdPart.CommandText = "spCPart"
```

```
cmdPart.Parameters.Append cmdPart.CreateParameter("@vPartID",  
adVarChar, adParamInput, 50, vPartID)
```

```
cmdPart.Parameters.Append cmdPart.CreateParameter("@vPartDescription",  
adVarChar, adParamInput, 150, vPartDescription)
```

```
cmdPart.Execute
```

```
Set cmdPart = Nothing
```

```
#If usemts Then
```

```
objcontext.SetComplete
```

```
Set objcontext = Nothing
```

```
#End If
```

CPart = "OK"

Exit Function

errhandler:

#If usemts Then

objcontext.SetAbort

Set objcontext = Nothing

#End If

CPart = Err.Number & "/" & Err.Source & "/" & Err.Description

End Function

“CPart” is a method of “clsPart” class and is defined as public function. The parameters that must be supplied to call this function are defined in the parenthesis following the function name (CPart).

It is very important to have a robust error handling to notify MTS of the status of the component’s work. If a component’s work is not complete then the transaction is failed for the sake of consistency of the database. On Error GoTo syntax is used to report an error to the client. “Err” object, which stores information about any error that occurs, is used to provide the error messages. Err object is an internal object of Visual Basic.

When an object is created, by the client or by another object, it must be aware of the context in which it is being used. Objects need to know that they are running inside the transaction, or in another case they may need to know that certain security requirements are met. MTS provides context by creating an associated context object for each MTS object instance. The context object provides information about the object’s execution environment, such as the identity of the object’s creator, and if the object is in a transaction. The context object also holds security information for the object that can be checked when it creates other MTS objects and collaborates with other context objects in the same

transaction to either commit or abort the transaction. If an object completes its work in the transaction successfully, it indicates its context object that it is complete. If an object fails to complete its work successfully, it indicates its context object that it has to abort the transaction. When all the objects in the transaction are finished running, MTS uses the information recorded in each context object to determine whether or not the transaction should commit. If all objects reported successful completion, then MTS commits the transaction. If one or more objects reported an abort, then MTS rolls back the transaction, undoing all changes made by all objects involved in the transaction. A context object is defined by ObjectContext. GetObjectContext function must be called to get a reference to an ObjectContext object. But, first, a reference to Microsoft Transaction Server Type Library must be set by clicking “References” on the “Project” menu. If the method called completes successfully, it calls the SetComplete method on the ObjectContext object before returning from the method call. If the method called completes unsuccessfully it calls the SetAbort method of the ObjectContext object before returning. If an object involved in a transaction calls SetAbort, the updates roll back even if other objects have called the SetComplete method.

The “Command” object contains a definition of a command that is run against a data source. If you are not using both ADO and DAO in the same application, you do not need to use the prefix “ADODB” before ADO data types. ActiveConnection property of the Command object is used to pass the required connection information, which is defined as “conString” outside the “Cpart” function. To use a stored procedure, CommandType property of the Command object is set to the constant adCmdStoredProc, CommandText property of the Command object is set to the name of the stored procedure (spCPart). Stored procedures may require one or more parameters to be passed to them. For each required parameter, a Parameter object should be created and appended to the Parameters collection of the Command object by using the Append method. Then, the Execute method is invoked. After the parameters are passed to the database, the command variable must be set to “Nothing” not to consume the computer

resources unnecessarily. To retrieve a data from the database, “Recordset“ object is created. An object may need to create another object to complete its work. CreateInstance method of theObjectContext object is used to create a new MTS object and pass context information to that new object.

After the classes of the project modules had been created, they were deployed on Microsoft Transaction Server. The screen shot of Microsoft Transaction Server can be seen in Figure 3.17.

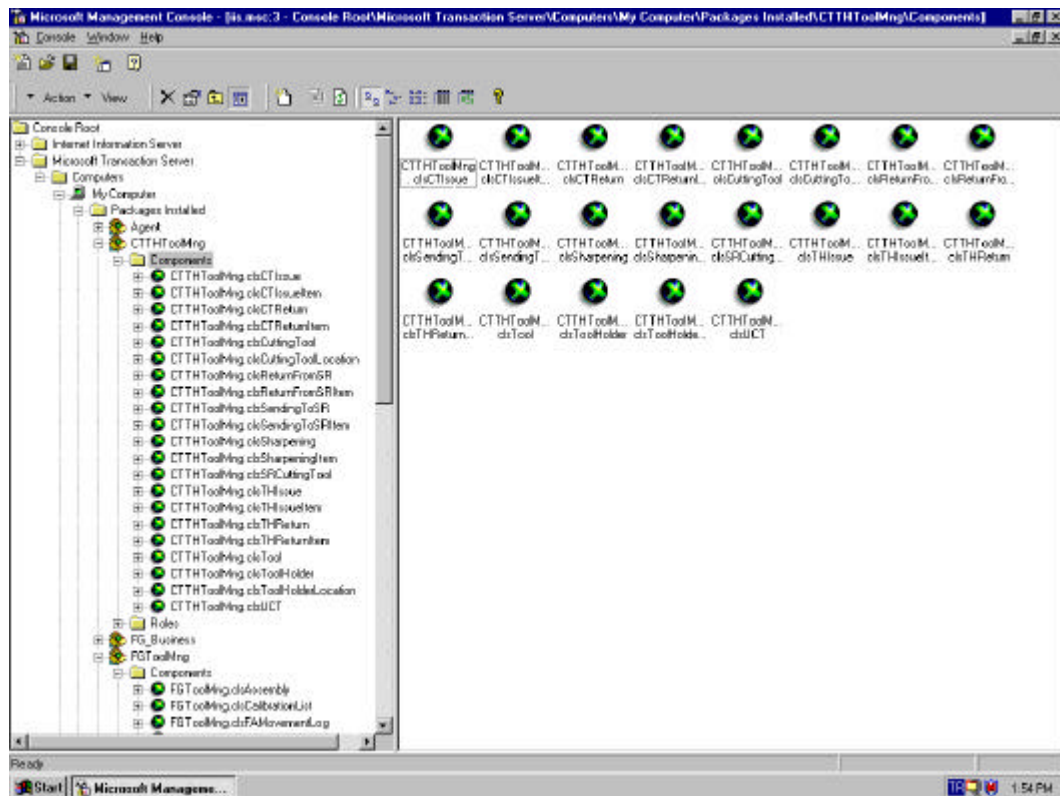


Figure 3.17 MTS View of Tool Management System

3.4.3 Client Tier Implementation

During the client tier implementation, active server pages were created and deployed in Microsoft Internet Information Server. Active server pages were created using Microsoft Visual InterDev. Tool management system consists of 4 Visual InterDev projects. LoginClientToolMng project is created to manage

on Internet Information Server site at the address:
<http://cimlabserver.me.metu.edu.tr/LoginClientToolMng>.

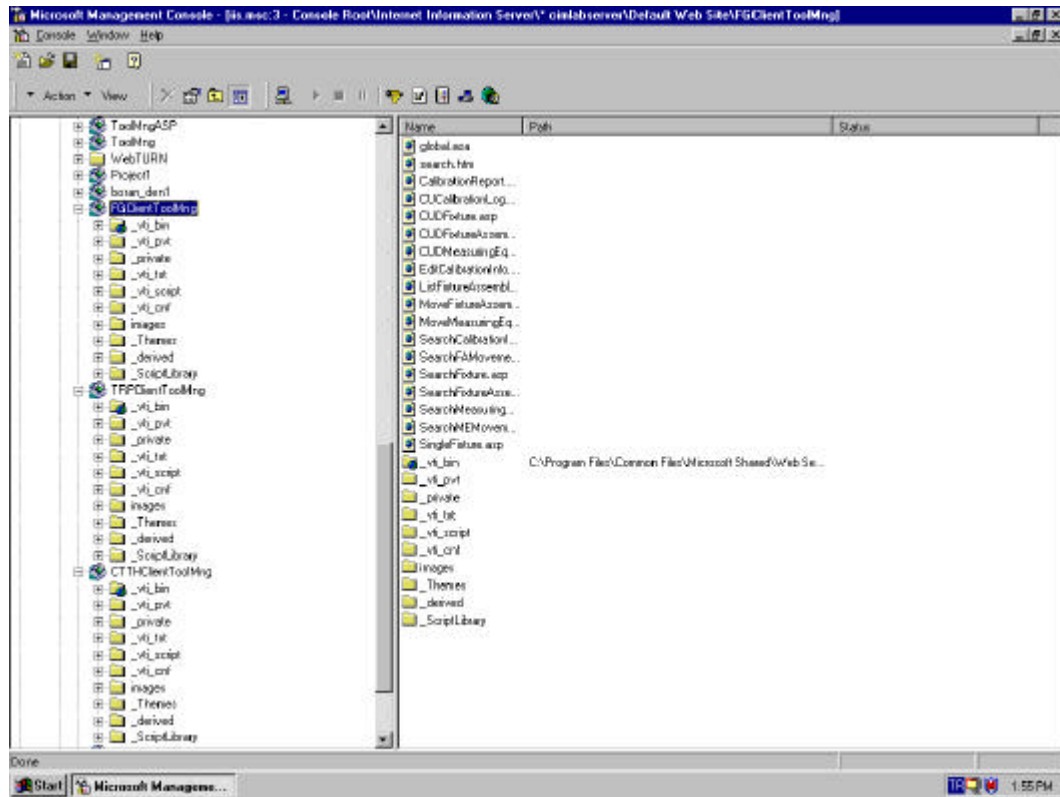


Figure 3.19 IIS View of Tool Management System

CHAPTER 4

TOOL MANAGEMENT SYSTEM OF IMTRG

4.1 Overview

Tool management system developed in the laboratory of IMTRG is an integrated and flexible tool management system. It is developed to carry out the basic tool management functions of a manufacturing company. Integration and flexibility properties of the system come from the software technology used. Windows DNA for Manufacturing is used to develop the system. Software has been developed using the object oriented approach, which makes the future developments easier. System can be extended by keeping the previous developments. The system is partitioned into modules so that necessary parts of the system can be chosen and implemented by different users.

System is partitioned into three modules: gage-fixture, cutting tool-tool holder, and tool requirement plan modules. In gage-fixture module, data of fixtures, fixture assemblies, and measuring equipments are tracked. Calibration data of the measuring equipments are also tracked in this module. Fixture assemblies and measuring equipments have unique tool identification numbers, while fixtures are tracked according to their types. Cutting tool-tool holder module tracks the data of the cutting tools, used cutting tools in the tool crib, cutting tools in the sharpening room, and tool holders. Issue, return, sending to sharpening orders are managed in this module. In tool requirement plan module, tool requirement planning according to the schedule and bill of tooling is calculated. Purchase orders, receivings from vendors, parts to be produced are handled in this module.

4.2 Gage-Fixture Module

Gages and fixture assemblies are tracked; their data and calibration of gages are managed in this module.

Each fixture type is given a tool ID. New fixtures can be added to the database, existing ones can be deleted from the database, and the data can be updated. Fixture data can be reached through the fixture search page. They can be searched according to tool ID, name and vendor. Fixtures are assembled to form a fixture assembly. They can only be moved as a fixture assembly. A screen view of fixture creation/update/deletion is shown in Figure 4.1.

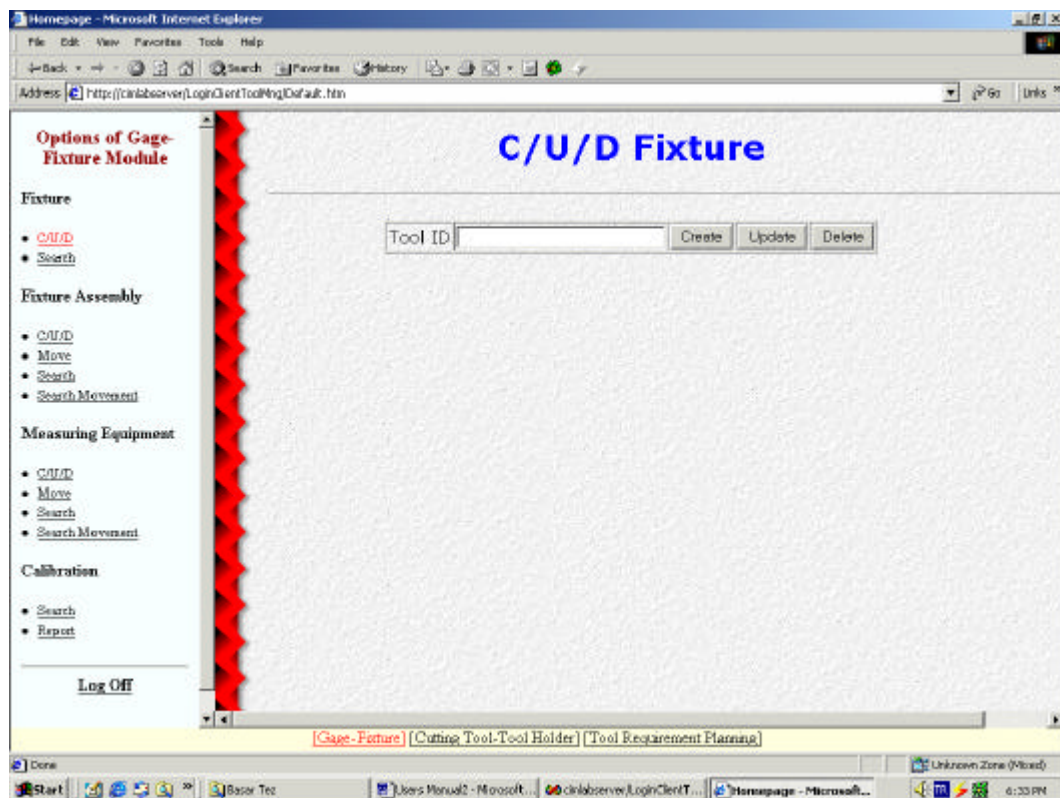


Figure 4.1 C/U/D Fixture Page

Fixture assemblies are tracked according to their unique tool ID's. New fixture assemblies can be created in the database by assembling fixtures. Their

data can be updated, fixtures in the assembly can be disassembled and all the assembly with all parts can be deleted from the database. Fixture assemblies can be moved and these movements are recorded. Fixture assembly data can be reached through the fixture assembly search page. They can be searched according to assembly ID, tool ID, name, location and status. Their movements can also be searched according to assembly ID, movement date, moved by, location and status. Fixture assembly search page is in Figure 4.2.

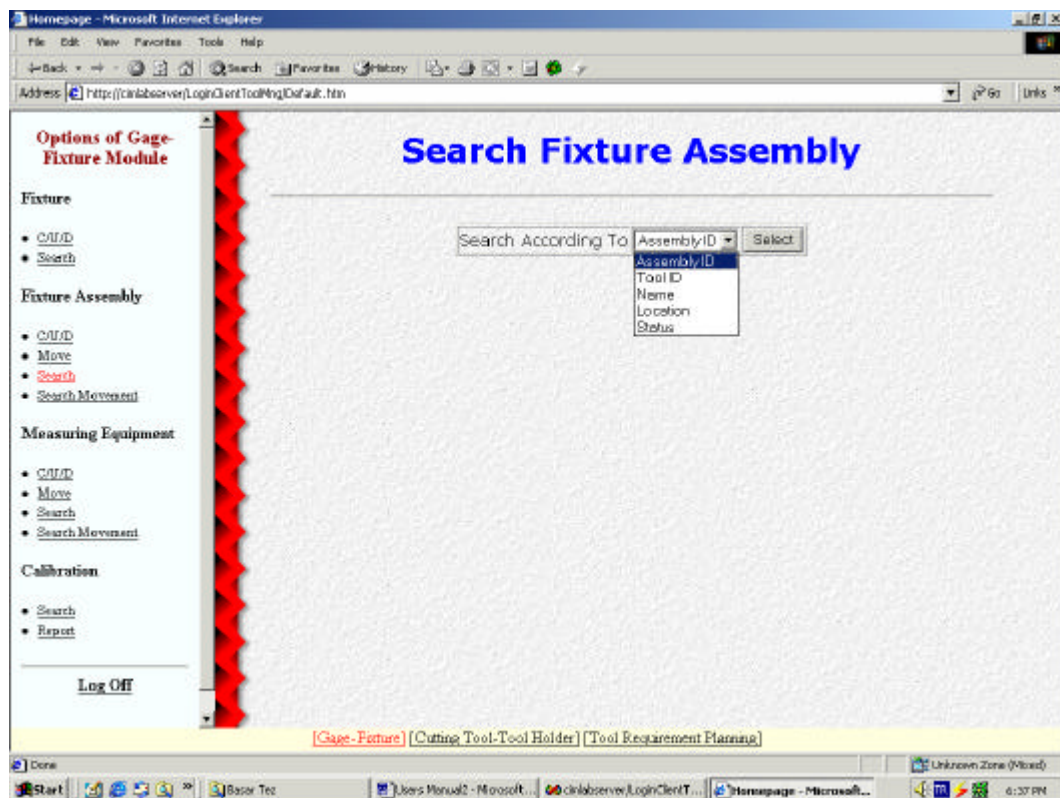


Figure 4.2 Search Fixture Assembly Page

Measuring Equipments are tracked according to their unique tool ID's. Same processes that can be done on a fixture assembly are also applicable on measuring equipments. User can create new measuring equipments, update and delete the existing ones. They can be searched according to tool ID, name, location, status, vendor and sensitivity in the measuring equipment search page. They can be moved and these movements are recorded. Their movements can be

searched according to tool ID, movement date, moved by, location and status. Measuring Equipment search page is in Figure 4.3.

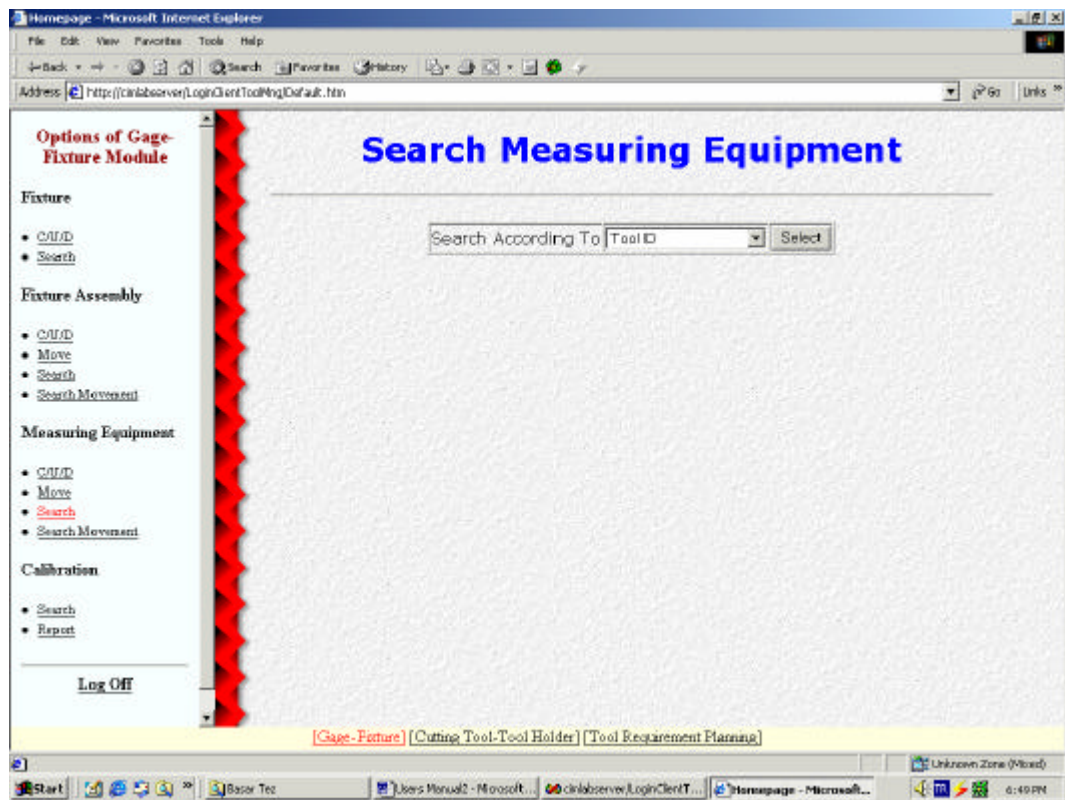


Figure 4.3 Search Measuring Equipment Page

Users can update calibration information of a measuring equipment, create and update calibration log, view the calibration report, get a list of measuring equipments that have one month and one week time to calibration end dates, list of ones whose calibration dates are expired and list of measuring equipments whose calibration end dates are the present day. Calibration information search page is given in Figure 4.4.

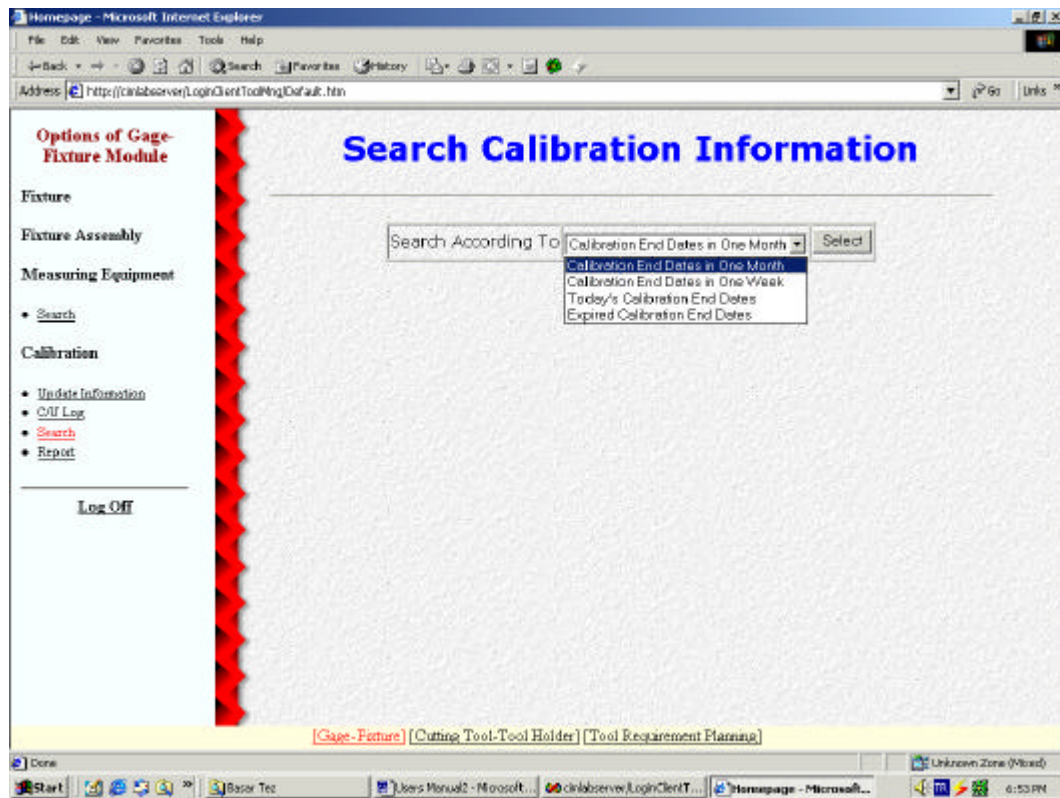


Figure 4.4 Search Calibration Information Page

4.3 Cutting Tool-Tool Holder Module

In this module, cutting tools and tool holders are tracked, their distribution (or issue) to shop floor, return to tool crib, sending to and return from sharpening room orders are managed. Sharpening process is tracked. The conditions of the cutting tools in the tool crib and sharpening room are kept. Used cutting tools are kept in a different database table and new cutting tool issues are not allowed if there is a used one of that tool type.

User can create new cutting tools in the system and update the data of the existing ones. Data of the cutting tools can be reached through cutting tool search page. Cutting tools can be searched according to tool ID, name, vendor, location and inventory under safety stock. Quantity on hand column in the cutting tool data represents brand new cutting tools in the tool crib. Cutting tool search page is in Figure 4.5.

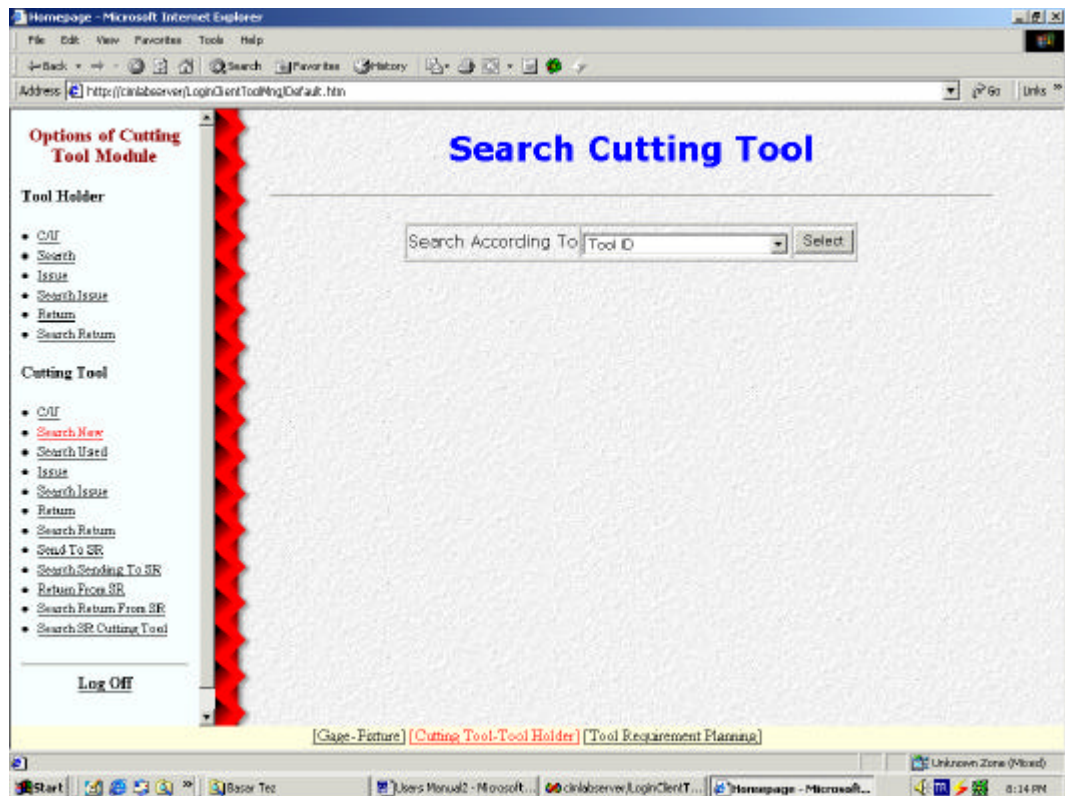


Figure 4.5 Search Cutting Tool Page

Used cutting tools can also be searched in used cutting tool search page. When the tool ID is supplied, the number of the used cutting tool with conditions are retrieved. Used cutting tools can be in sharp, blunt, problematic and, scrap conditions. A cutting tool can not be returned if it is unused. Used cutting tool search page is in Figure 4.6.

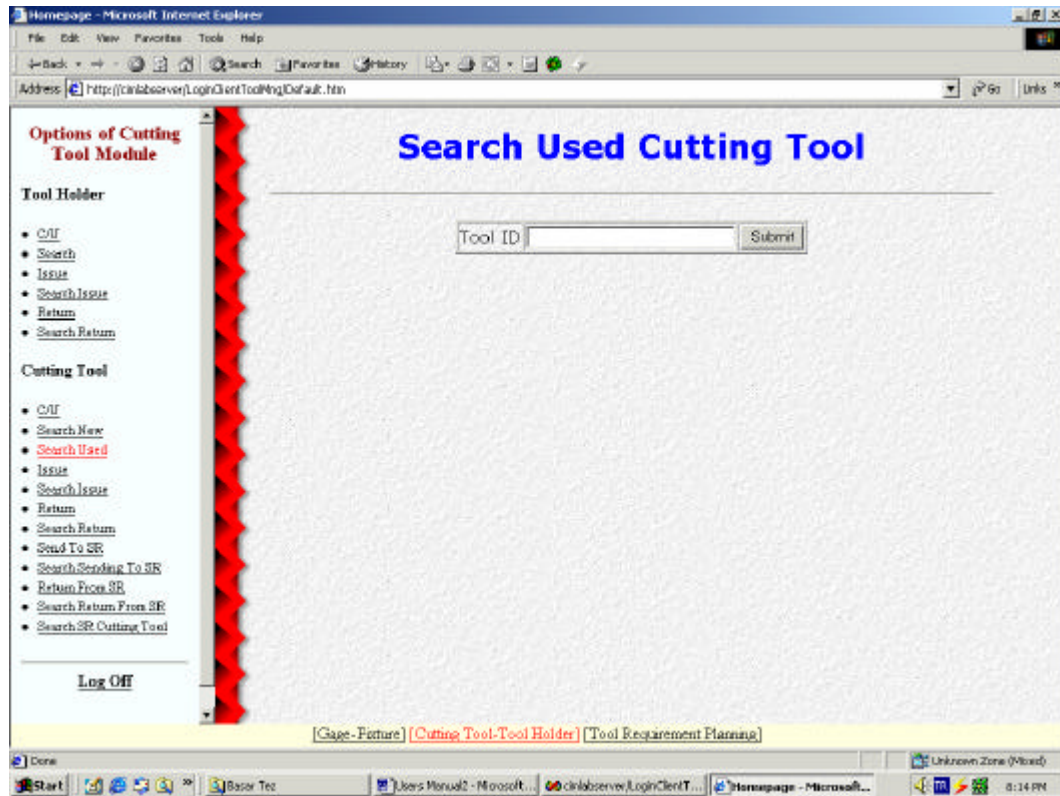


Figure 4.6 Search Used Cutting Tool Page

The sharpening activities carried out in the sharpening room are logged and the inventory of the cutting tools that are sharpened and to be sharpened is managed. Cutting tool issues are done only from the tool crib. All the blunt tools are returned to the tool crib before going to sharpening room and all the tools that have been sharpened are returned to the tool crib even if they are scrap or problematic. The cutting tools in the sharpening room can be searched supplying the tool ID. After the search, number of sharp, blunt, problematic and scrap cutting tools of that tool type can be retrieved. The screen view of the page in which sharpening log is created is in Figure 4.7.

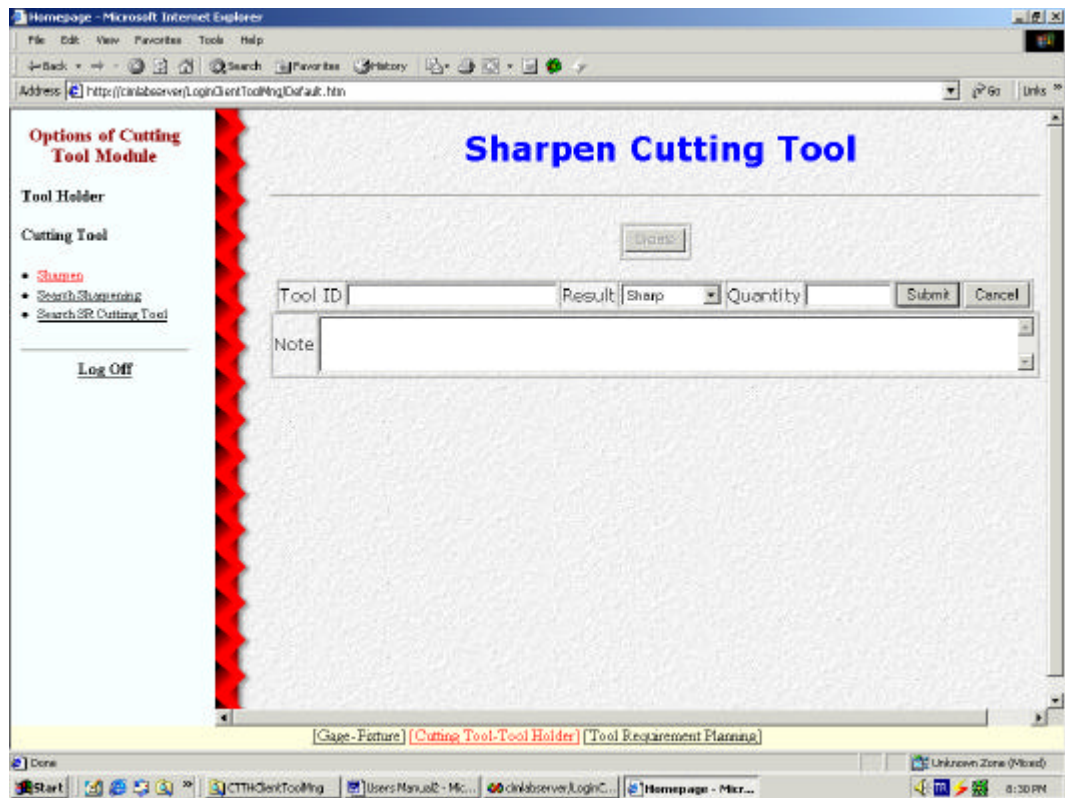


Figure 4.7 Sharpen Cutting Tool Page

Users can create and update tool holders. They can be issued to the shop floor and returned to tool crib in this module. Locations of each tool are tracked. Tool holders can be searched in the tool holder search page. They can be searched according to tool ID, name, vendor and location. Tool holder issue page can be seen in Figure 4.8.

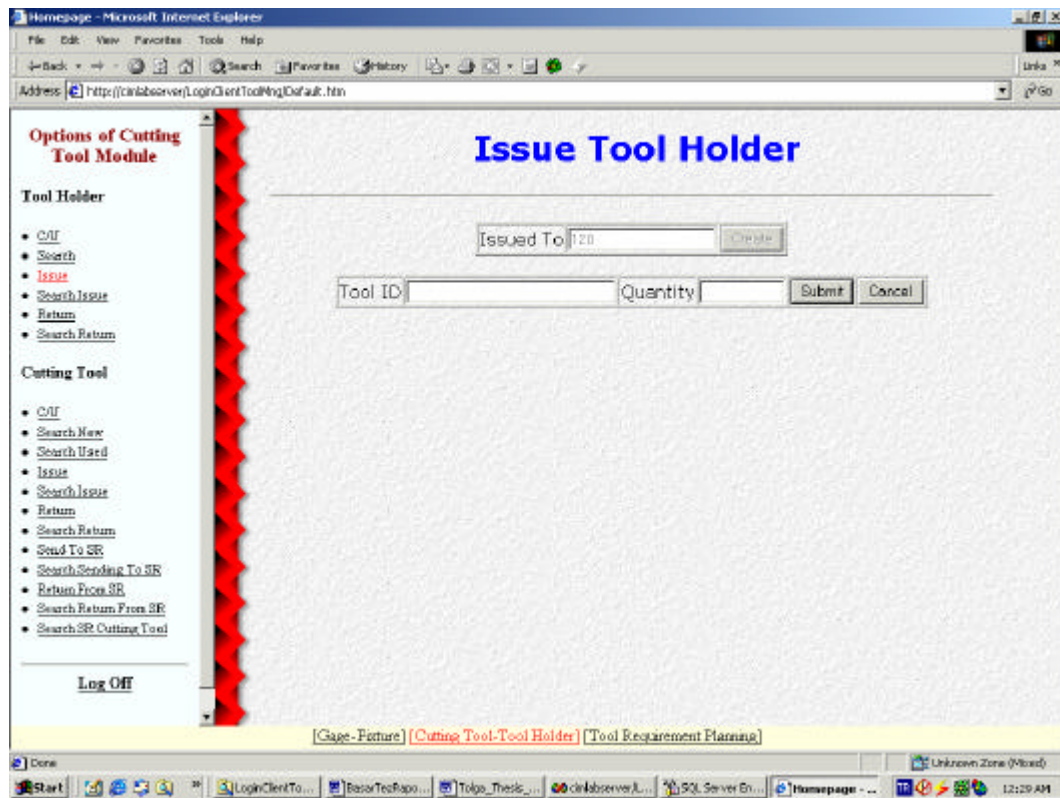


Figure 4.8 Issue Tool Holder Page

4.4 Tool Requirement Plan Module

Tool requirement is calculated and purchase orders are suggested by the computer if it is necessary for a continuity of production. Tools and their numbers required to manufacture each part is entered into the system in bill of tooling creation page. Production schedule of parts is entered into the system in schedule creation page. These data can be searched and updated in the search pages of bill of tooling and schedule. Calendar can be updated changing the status of the days as holiday or workday. Tool requirement is calculated by using the data of calendar, schedule, bill of tooling, purchase orders which are not closed, number of cutting tools on hand, and plan parameters of cutting tools.

There are two plan parameters for a cutting tool: fixed purchase quantity and fixed purchase cycle. If fixed purchase quantity is selected as a plan parameter for a cutting tool, the quantity of the suggested orders is given as the

closest number to the requirement whose multiply is the fixed purchase quantity value of that cutting tool. If fixed purchase cycle is selected, the time period between two successive suggested orders are fixed, which is the fixed purchase cycle value of the cutting tool. Requirement plan of all cutting tools, requirement plan of a single cutting tool, warning list where the cutting tools under the safety stock levels are supplied after the each execution of the requirement plan. Requirement plan page for all the cutting tools can be seen in Figure 4.9.

Options of TRP Module

TRP

- Run TRP
- **Requirement**
- Single Requirement
- Suggested Purchases
- Single Suggested Purchases
- Warning
- U Calendar

Part

- C/D
- List

BOT

- C/D
- Search

Schedule

- C/D
- Search

Purchase Order

Discontinue

Requirement Plan

ct101

Date	Requirement	Scheduled Receipt	Planned Receipt	Total Receipt	Available
8/28/03	0	0	50	50	50
8/29/03	40	0	0	0	10
8/30/03	0	0	50	50	60
9/1/03	40	0	0	0	20
9/2/03	50	0	50	50	20
9/3/03	45	0	25	25	0
9/4/03	50	50	50	100	50
9/5/03	35	0	0	0	15
9/6/03	0	0	25	25	40
9/8/03	40	0	0	0	0
9/9/03	40	0	50	50	10
9/10/03	40	0	50	50	20
9/11/03	40	0	75	75	55
9/12/03	40	0	0	0	15

ct102

Date	Requirement	Scheduled Receipt	Planned Receipt	Total Receipt	Available
8/28/03	0	0	100	100	05

[Gage-Feature] [Cutting Tool-Tool Holder] [Tool Requirement Plan]

Figure 4.9 Requirement Plan Page

Purchase orders and receivings from vendors are recorded in this module. New purchase orders and receivings from vendors can be created, existing purchase orders can be updated. These operations effect the inventory and the next requirement plan. The purchase orders are fixed as planned receipts while the purchase orders suggested by the tool management system are fixed as scheduled receipts in the requirement. Purchase orders can be searched according to purchase order no, purchase order date, planned receiving date, ordered by, tool

ID and status, and tool ID and planned receiving date. Receivings can be searched according to receiving no, receiving date, received by, and tool ID. Purchase order creation and update page can be seen in Figure 4.10.

Options of TRP Module

TRP

- Requirement
- Suggested Purchases

Part

BOT

Schedule

- Search

Purchase Order

- C/U**
- Search

Receiving

[Log Off](#)

C/U Purchase Order

No. Ordered By Order Date

	Tool ID	Qty	Planned Receiving Date	Status
<input type="radio"/>	ct101	50	5/5/03	Not OK
<input type="radio"/>	ct102	50	5/5/03	Not OK
<input type="radio"/>	ct103	40	5/5/03	Not OK
<input type="radio"/>				Not OK

[\[Cage-Parture\]](#)
[\[Cutting Tool-Tool Holder\]](#)
[\[Tool Requirement Planning\]](#)

Figure 4.10 C/U Purchase Order Page

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

An integrated tool management system was developed as a project of IMTRG (Integrated Manufacturing Technologies Research Group) in this study. A tool management system previously developed in this laboratory was used as a basis in the development. The rules of the previous system were changed with more common rules to make the system applicable to more plants. The rules in Türk Traktör plant were adopted for that purpose. Tool crib has the absolute control on the tools. All the tools have to return back to the tool crib before going to sharpening, scrap bin or a new location on the shop floor. New functions were added to the program. Tools can be tracked according to their locations on the shop floor. Used tools can be returned to the tool crib and their data is kept separately. Tool holder tracking was added to the system. System modules have been formed according to the changes made.

Since tool management systems may be different according to the rules of the plants, flexibility and integration capabilities of the software platform was the most important issue of the project. Microsoft Windows DNA was selected as the software platform. “Windows DNA for Manufacturing” vision of Microsoft which implements multi-tier development architecture to manufacturing solutions with its all integration tools and the power of Microsoft on desktop and server platforms were the reasons of choosing DNA platform.

Tool management system has been developed as three modules which are discussed in the previous sections. These modules are gage-fixtured, cutting tool holder and tool requirement plan modules. These three modules basically

cover almost all of the tool management activities of a metal removal based manufacturing firm.

The major benefits of this system can be defined as follows:

- ?? Keeping and distributing static tool properties like name, definition, vendor, etc. in a structured manner: Authorized users can add new tools, update the properties of existing tools. These data can easily be retrieved via internet.
- ?? Rapid access to the locations of measuring equipment, fixture, cutting tool and tool holders on the shop floor: Location of each measuring equipment and fixture assembly can be retrieved easily. By using the search pages of the system, user can track which workers have how many cutting tools and tool holders or which cutting tool and tool holder is being used by which worker.
- ?? Easy calibration management: Measuring equipments which need to be calibrated can be seen. Calibration information of the measuring equipments and calibration reports are recorded.
- ?? Real time monitoring of cutting tool and tool holder inventory: How many cutting tool and tool holders are in use and how many of them are in the tool crib can be seen. Inventory of the cutting tools are managed according to the safety stock levels of cutting tools.
- ?? Purchasing planning based on tool requirement planning: System calculates how many cutting tools will be needed for each day of the production schedule of parts. According to this calculation, suggested purchases are prepared using the purchase plan parameter of the cutting tool. Using the requirement planning function, production is not delayed because of the lack of the cutting tools and unnecessary investment on cutting tools is prevented.
- ?? Online inventory tracking of used cutting tools: Number of the sharp, blunt, problematic, and scrap cutting tools in the tool crib can be retrieved.

- ?? Sharpening room activity tracking: Sharpening activities in the sharpening room and their results are recorded. Number of sharp, blunt, problematic, and scrap cutting tools in the sharpening room can be retrieved.

A generalized tool management system was developed in this work. But, tool management concept is too wide. In this study, no effort is spent on making use of tool management studies that were carried out on decision analysis like tool allocation, tool sharing, tool magazine arrangements, multi purpose scheduling, etc. Reengineering schemes, which exist in the tool management software packages mentioned in the literature survey, like mini crib implementation, tool kits and assemblies, tool data management having technical drawings of tools are not used.

In the future, this tool management program can be further developed in various ways. The following topics are suggested as future work of this thesis:

- ?? Implementation of tool kits and assemblies: Tools are grouped under a kit ID. They are put into kits before the start of the production and issued to the shop floor in these kits. This decreases the possibility of issuing wrong tools and tool waiting time in front of the tool crib. Tool holders, adapters and cutting tools are grouped under an assembly ID and form an assembly. Assembly can be considered as a kind of kit.
- ?? Implementation of mini crib in tool management: Tool inventories of the tool cribs in different locations of the plant are tracked.
- ?? Recording the technical drawings of the tools into the database: Designers can easily access the tool data and drawings and decide if that tool is useful or not.
- ?? Tracking individual tool life using the built-in data chips embedded onto tools: Data chips embedded onto tools can carry data. Remaining tool life of individual cutting tools can be tracked using these data chips. So that same tools can be used for several jobs as long as tool life is available. Tool life sharing between several jobs saves on the number of sister tools, machine downtime, amount of load/unload, tool setup time, tool changing,

kit size and cost. The problem is to measure and record the remaining tool life in a reliable way.

?? Calculation of tool requirement according to scheduled jobs and different tool management and tool issue strategies: After jobs are scheduled among the machine tools, tool requirement for each machine tool is calculated according to the selected tool issue strategy. Tool issue strategies can be grouped as tool-oriented or workpiece-oriented strategies. Workpiece-oriented tool issue strategies are full kitting, differential kitting, resident kitting and single tool strategies. Tool-oriented strategies are full clustering, differential clustering and resident clustering strategies. However, applying these strategies depend on the hardware configuration.

REFERENCES

- [1] Hannam, R., Computer Integrated Manufacturing: from Concepts to Realization, Addison-Wesley Longman Ltd., Harlow, England, 1997
- [2] Gayman, D.J., "Computers in Tool Crib", Manufacturing Engineering, September 1986, pp.41-44
- [3] Anumolu, B., and Shewchuk, J.P., "Design of a Tooling Database Implementation for an Existing Facility", Computers in Industry, Vol.42, 2000, pp.221-229
- [4] Gray, A.E., Siedmann, A., and Steckle, K.E., "A Synthesis of Decision Models for Tool Management in Automated Manufacturing", Management Science, Vol.9, No.5, 1993, pp.549-567
- [5] O'Brien, C., Schafer, H., and Bryant, J.K., "Intelligent Tool Management Systems-A Missing Element in CIM", Proceedings of 2nd International Conference on Computer Integrated Manufacturing, 1993, Vol.2, pp.525-530
- [6] Tap, M., Hewit, J.R., and Meeran, S., "An Active Tool Tracking System for Increased Productivity", International Journal of Production Research, Vol.38, No.16, 2000, pp.3889-3898
- [7] Raisglid, W., Better Tool Cribs, The Industrial Press, New York, 1954
- [8] Rhodes, J.S., "FMS Tool Management Systems", CASA/SME Flexible Manufacturing Systems '86 Conference, 1986, pp.269-282

- [9] Zheng, Y., and Bilberg, A., "Advanced Tool Management Systems in CIM Environment", Proceedings of 2nd International Conference on Computer Integrated Manufacturing, 1993, Vol.2, pp.516-524

- [10] Kehoe, D.F., Little, D., Al-Maliki, I., and Wyatt, T.M., "Method for the Specification of Tool Management Information Systems", Computer Integrated Manufacturing Systems, Vol.9, No.5, 1991, pp.18-25

- [11] Picozzi, D.A., "Computerized Tool Management", Tooling and Production, September 1990, pp.79-82

- [12] "Advanced Tool Management System", White Paper, ITC Integrated Systems, Inc.

- [13] Martin Plute, "Who Controls the Tools in your Company? ", White Paper, ITC Integrated Systems, Inc., 1999

- [14] Ozbayrak, M., De Souza, R.B.R., and Bell, R., "Design of a Tool Management System for a Flexible Machining Facility", Proceedings of the Institution of Mechanical Engineers, 2001, Vol.215, Part B, pp.353-370

- [15] Harmon, P., and Sawyer, B., UML for Visual Basic 6.0 Developers: Using Visual Modeler and Rational Rose 98, Morgan Kaufmann Publishers Inc., USA, 1999

- [16] Dennis, A., Wixom, B., and Tegarden, D., System's Analysis and Design: An Object-Oriented Approach with UML, John Wiley & Sons Inc., USA, 2002

- [17] Hoffer, A.H., George, J.F., and Valacich, J.S., Modern Systems Analysis and Design, Addison-Wesley Longman Inc., USA, 1998

- [18] Kendall, K.E., and Kendall, J.E., Systems Analysis and Design, Prentice-Hall Inc., USA, 1999
- [19] Microsoft Corporation, Enterprise Development Using Microsoft Visual Basic 6.0, Microsoft Press, USA, 1999
- [20] "Windows DNA 2000-The Platform for Building the Business Internet", White Paper, Microsoft, 1999
- [21] "Microsoft Windows DNA for Manufacturing", White Paper, Microsoft, 1999
- [22] Petkovic, D., Microsoft SQL Server 2000: A Beginner's Guide, McGraw-Hill, USA, 2000
- [23] Otey, M., and Conte, P., Microsoft SQL Server 2000: Developer's Guide, McGraw-Hill, USA, 2001
- [24] Microsoft Corporation, Microsoft SQL Server 7.0 Administration Training, Microsoft Press, USA, 2000
- [25] Microsoft Corporation, Microsoft SQL Server 7.0 Data Warehousing Training, Microsoft Press, USA, 2000
- [26] "Designing Efficient Applications for Microsoft SQL Server, White Paper, Microsoft, 1998
- [27] "A Guide to Reviewing and Evaluating Microsoft Transaction Server, White Paper, Microsoft, 1997
- [28] "Microsoft Transaction Server: A General Purpose Infrastructure for Multitier Applications, White Paper, Microsoft, 1996

- [29] Michael Halvorson, Learn Visual Basic 6.0 Now, Microsoft Press, USA, 1999
- [30] Stephen Walther, Active Server Pages Unleashed, Sams.net Publishing, USA, 1998
- [31] Booch, G., Object-Oriented Analysis and Design with Applications, Addison-Wesley Longman Inc., California, USA, 1994
- [32] Yourdon, E., and Argila, C., Case Studies in Object-Oriented Analysis and Design, Prentice-Hall Inc., USA, 1996
- [33] Lee, R.C., and Tepfenhart, W.M., UML and C++: A Practical Guide to Object-Oriented Development, Prentice-Hall Inc., USA, 1997
- [34] Fowler, M., and Scott, K., UML Distilled, Addison-Wesley Longman Inc., USA, 1997
- [35] Quatriani, T., Visual Modeling with Rational Rose and UML, Addison-Wesley Longman Inc., USA, 1998
- [36] Larman, C., Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design, Prentice-Hall Inc., USA, 1998
- [37] Cantor, M.R., Object-Oriented Project Management with UML, John Wiley & Sons Inc., USA, 1998
- [38] Schneider, G., and Winters, J.P., Applying Use Cases: A Practical Guide, Addison-Wesley Longman Inc., USA, 1998

- [39] D'Souza, D.F., and Wills, A.C., Objects, Components and Frameworks with UML, Addison-Wesley Longman Inc., Massachusetts, USA, 1999
- [40] Bennett, S., McRobb, S., and Farmer, R., Object-Oriented System Analysis and Design Using UML, McGraw-Hill, England, 1999
- [41] Priestly, M., Practical Object-Oriented Design with UML, McGraw-Hill, England, 2000
- [42] Marshall, C., Enterprise Modeling with UML: Designing Successful Software through Business Analysis, Addison-Wesley Longman Inc., Massachusetts, USA, 2000
- [43] Stevens, P., and Pooley, R., Using UML: Software Engineering with Objects and Components, Addison-Wesley Longman Inc., Great Britain, 2000
- [44] Martin Plute, "The Basics of Tool Management Systems", Mold Making Technology Magazine, 2002
- [45] Martin, J.M., "Managing Tools Makes the Cell Click", Manufacturing Engineering, April 1989, pp.59-62
- [46] Mason, F., "Getting Control Over Tools is a Trend", American Machinist, May 1991, pp.45-49
- [47] Souza R.D., "Tool-Provisioning Strategies for Flexible Manufacturing Systems", Robotics and Computer Integrated Manufacturing, Vol.13, No.1, 1997, pp.31-39
- [48] Maropoulos, P.G., "Review of Research in Tooling Technology, Process Modeling and Process Modeling: Part1: Tooling and Process Modeling", Computer Integrated Manufacturing Systems, Vol.8, No.1, 1995, pp.5-12

- [49] Cantemessa, M., and Lombardi, F., “Tool Flow Planning in a Flexible Manufacturing Systems”, Computer Integrated Manufacturing Systems, Vol.6, No.2, 1993, pp.93-102

- [50] Tsukada, T.K., and Shin, K.G., “Distributed Tool Sharing in Flexible Manufacturing Systems”, IEEE Transactions on Robotics and Automation, Vol.14, No.3, 1998, pp.379-389

- [51] Akturk, M.S., and Onen, S., “Joint Lot Sizing and Tool Management in a CNC Environment”, Computers in Industry, Vol.40, 1999, pp.61-75

- [52] Atan, T.S., and Pandit, R., “Auxiliary Tool Allocation in Flexible Manufacturing Systems”, European Journal of Operational Research, Vol.89, 1996, pp.642-659

- [53] Koo, P.H., Tanchoco, J.M.A., and Talavage, J.J., “Tool Requirements in Manufacturing Systems under Dynamic Tool Sharing”, Computers & Industrial Engineering, Vol.33, Nos.3-4, 1997, pp.537-540

- [54] Kumar, A., Mohamed, Z., Motwani, J., and Youssef, M., “A Dynamic Tool Requirement Planning Model for Flexible Manufacturing Systems”, The International Journal of Flexible Manufacturing Systems, Vol.9, 1997, pp.307-342

- [55] Choi, Y.C., and Kim, Y.D., “Tool Replacement Policies for a Machining Centre Producing Multiple Types of Products with Distinct Due Dates”, International Journal of Production Research, Vol.39, No.5, 2001, pp.907-921

- [56] Jeang, A., “Reliable Tool Replacement Policy for Quality and Cost”, European Journal of Operational Research, Vol.108, 1998, pp.334-344

- [57] Mohamed, Z.M., and Bernardo, J.J., "Tool Planning Models for Flexible Manufacturing Systems", European Journal of Operational Research, Vol.103, 1997, pp.497-514

- [58] Mursec, B., Cus, F., and Balic, J., "Organization of Tool Supply and Determination of Cutting Conditions", Journal of Materials Processing Technology, Vol.100, 2000, pp.241-249

- [59] Avci, S., and Akturk, M.S., "Tool Magazine Arrangement and Operations Sequencing on CNC Machines", Computers & Operations Research, Vol.23, No.11, 1996, pp.1069-1081

- [60] Gyampah, K.A., Meredith, J.R., and Raturi, A., "A Comparison of Tool Management Strategies and Part Selection Rules for a Flexible Manufacturing System", International Journal of Production Research, Vol.30, No.4, 1992, pp.733-748

- [61] Kashyap, A.S., and Khator, S.K., "Analysis of Tool Sharing in an FMS: A Simulation Study", Computers & Industrial Engineering, Vol.30, No.1, 1996, pp.137-145

- [62] Akturk, M.S., and Avci, S., "Tool Allocation and Machining Conditions Optimization for CNC Machines", European Journal of Operational Research, 1996, Vol.94, pp.335-348

- [63] Zavanella, L., "Guidelines for Tool Store Dimensioning in Flexible Manufacturing Using an Analytical Technique", International Journal of Production Economics, Vol.46-47, 1996, pp.563-573

- [64] Durak, U., Ünver, H.Ö., Anlagan, Ö., and Kiliç, S.E., “Conceptual Design of Gage and Fixture Tracking”, UMTIK 2000 The 9th International Conference on Machine Design and Production, 2000, Ankara, Turkey
- [65] Durak, U., “Tool Management in CIM”, M.Sc. Thesis in Mechanical Engineering, Middle East Technical University, Ankara, Turkey, 2000
- [66] Sari, B., “Development of a Job Shop Planning System: Optimization of Machinig Operations Using Windows DNA Architecture”, M.Sc. Thesis in Mechanical Engineering, Middle East Technical University, Ankara, Turkey, 2001
- [67] Rosen, A., E-Commerce Question and Answer Book: A Survival Guide for Business Managers, AMACOM Books, 1999
- [68] Laudon, K.C., Management Information Systems: Organization and Technology in the Networked Enterprise, Prentice-Hall Inc., USA, 2000

APPENDIX A

UML DIAGRAMS

A.1 Use Case Diagrams

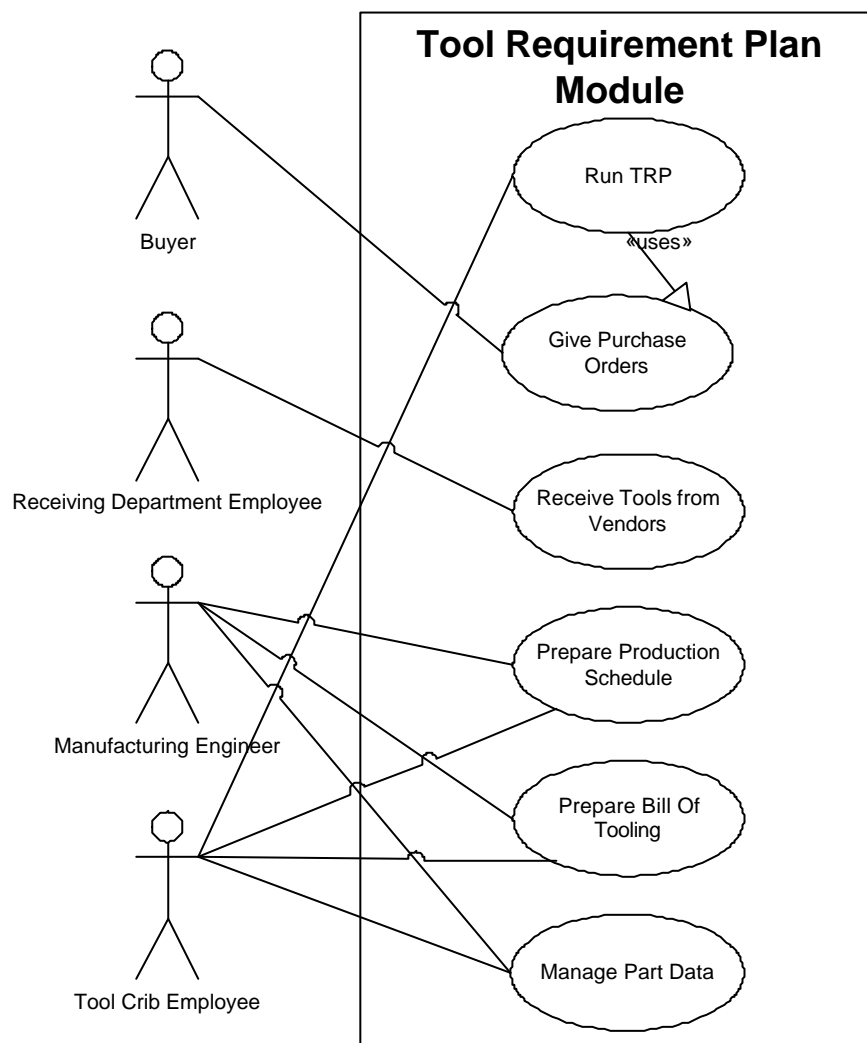


Figure A.1 Use Case Diagram of Tool Requirement Plan Module

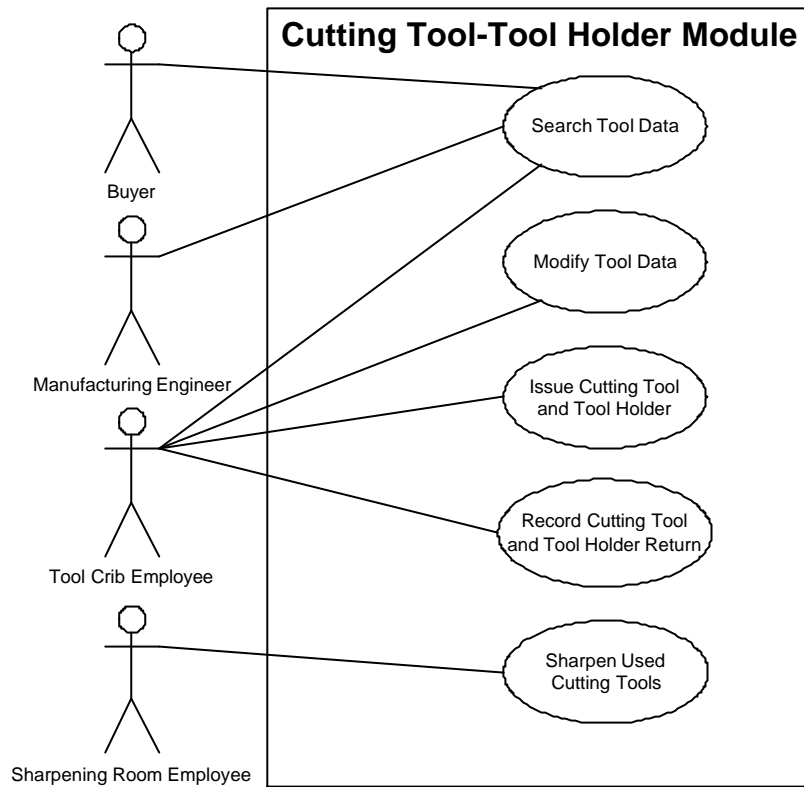


Figure A.2 Use Case Diagram of Cutting Tool-Tool Holder Module

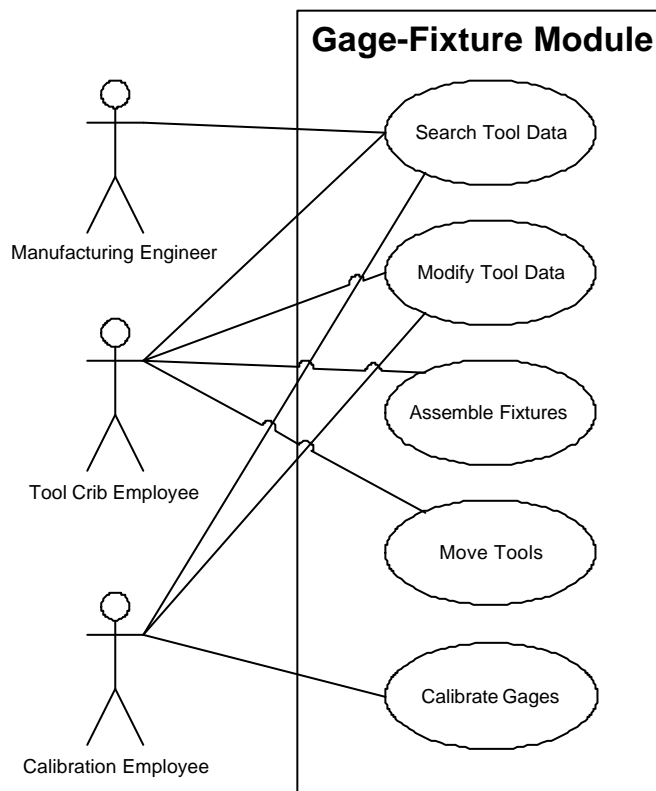


Figure A.3 Use Case Diagram of Gage-Fixture Module

A.2 Activity Diagrams

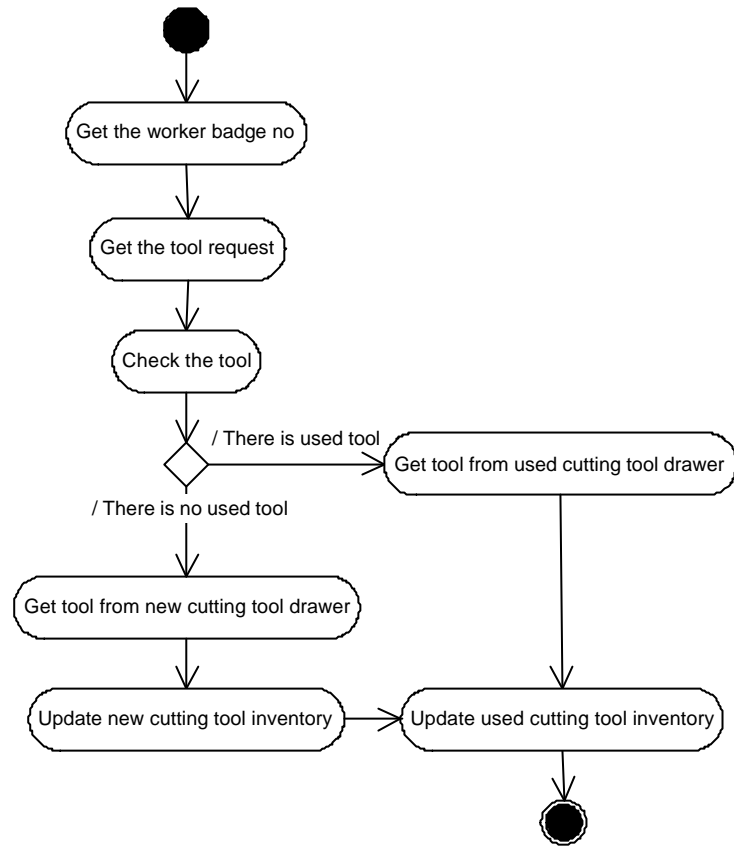


Figure A.4 Activity Diagram of Cutting Tool Issue

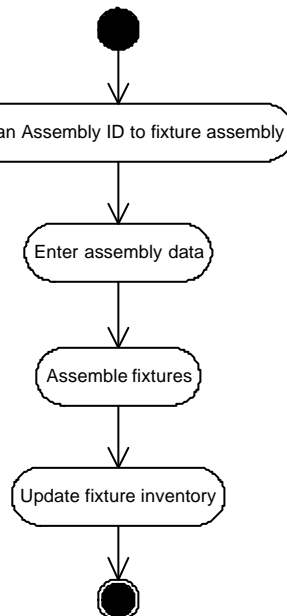


Figure A.5 Activity Diagram of Fixture Assembly Creation

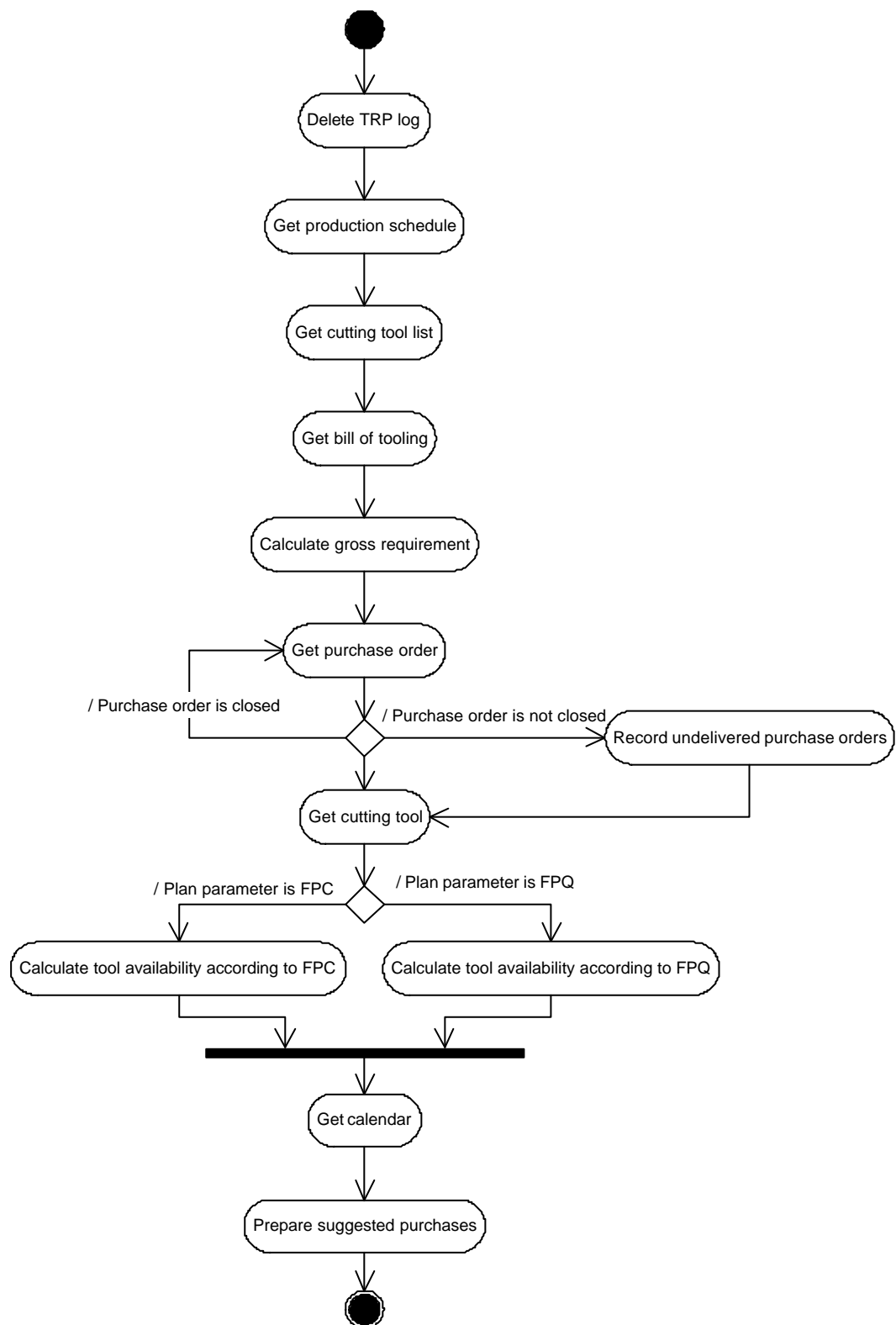


Figure A.6 Activity Diagram of Requirement Plan Run

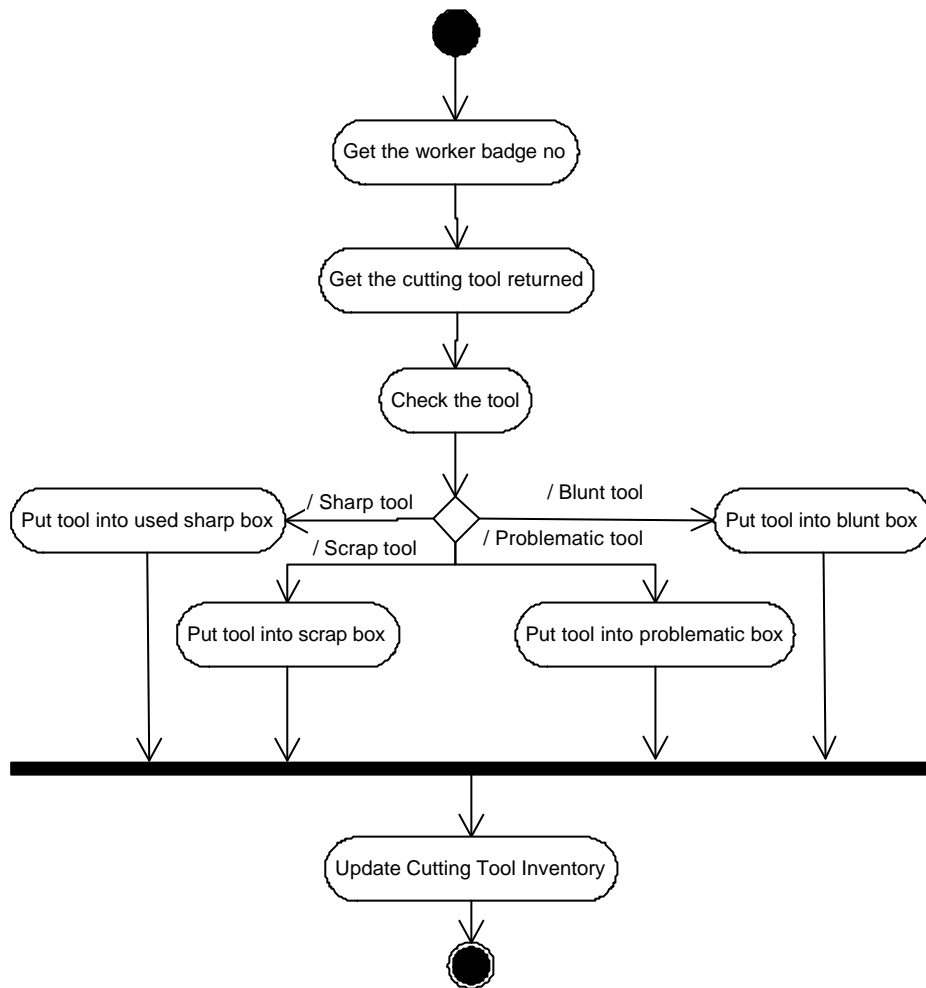


Figure A.7 Activity Diagram of Cutting Tool Return

A.3 Sequence Diagrams

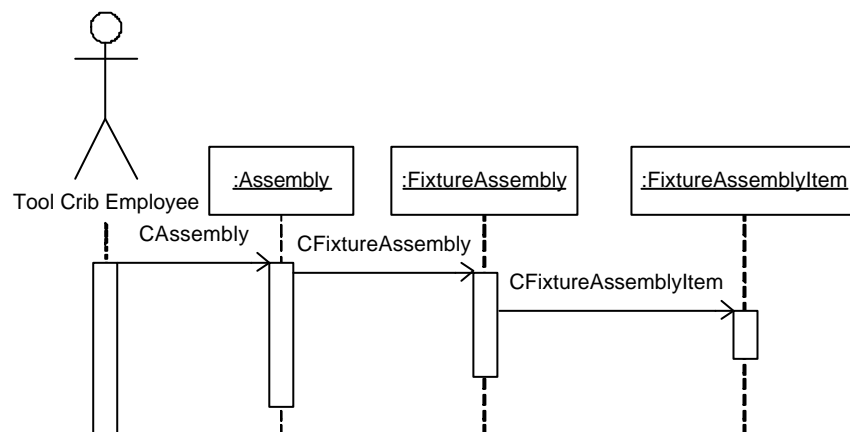


Figure A.8 Sequence Diagram of Fixture Assembly Creation

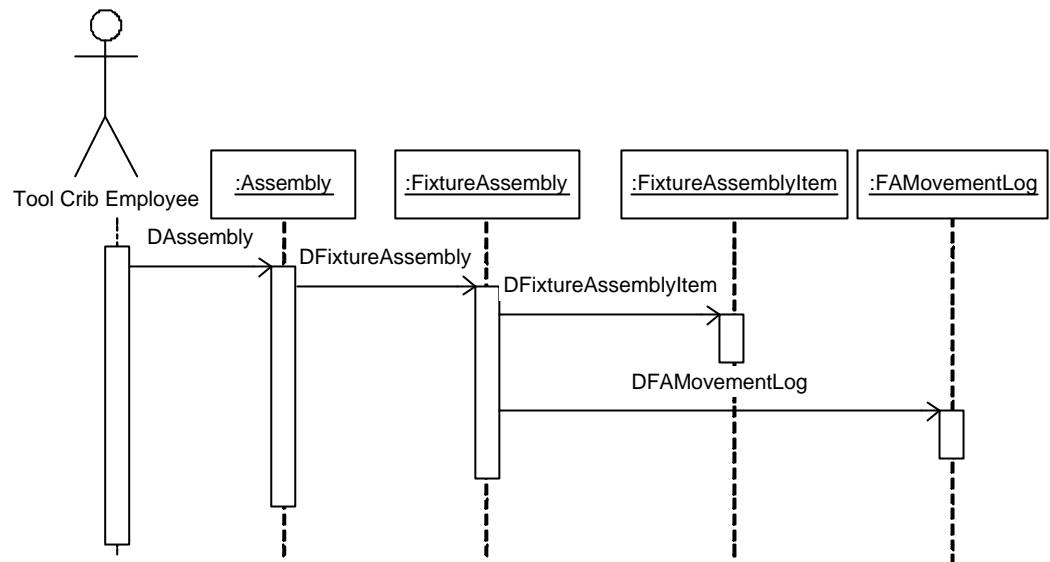


Figure A.9 Sequence Diagram of Fixture Assembly Deletion

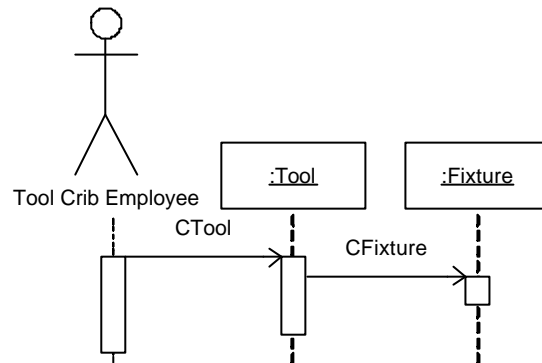


Figure A.10 Sequence Diagram of Fixture Creation

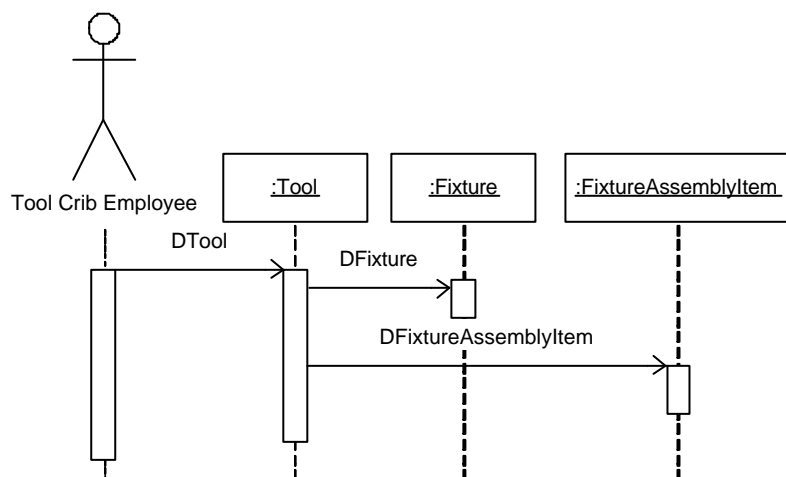


Figure A.11 Sequence Diagram of Fixture Deletion

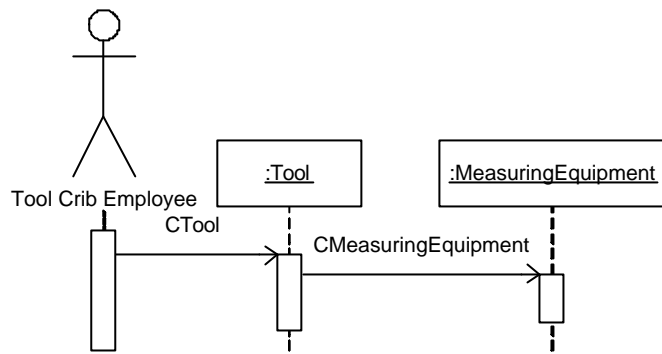


Figure A.12 Sequence Diagram of Measuring Equipment Creation

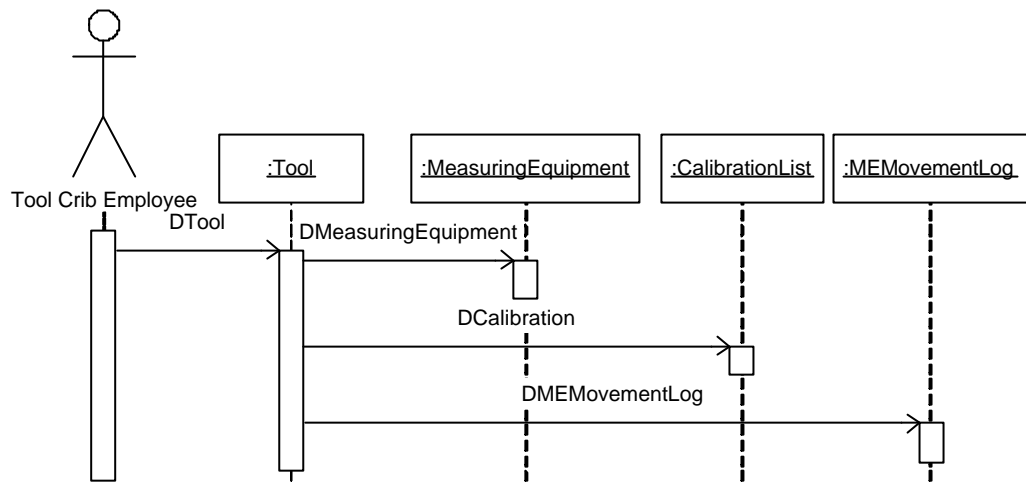


Figure A.13 Sequence Diagram of Measuring Equipment Deletion

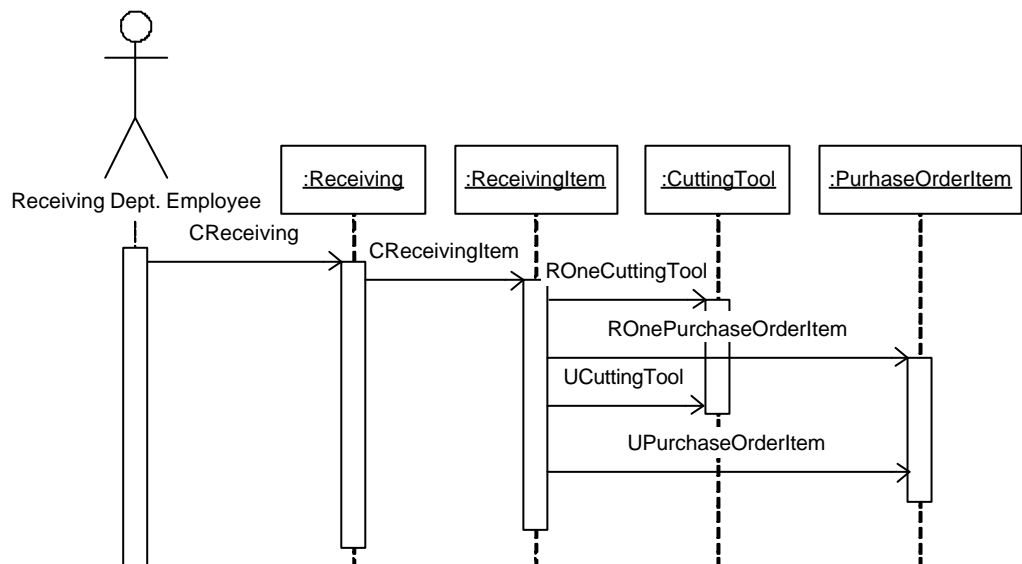


Figure A.14 Sequence Diagram of Cutting Tool Receiving from Vendors

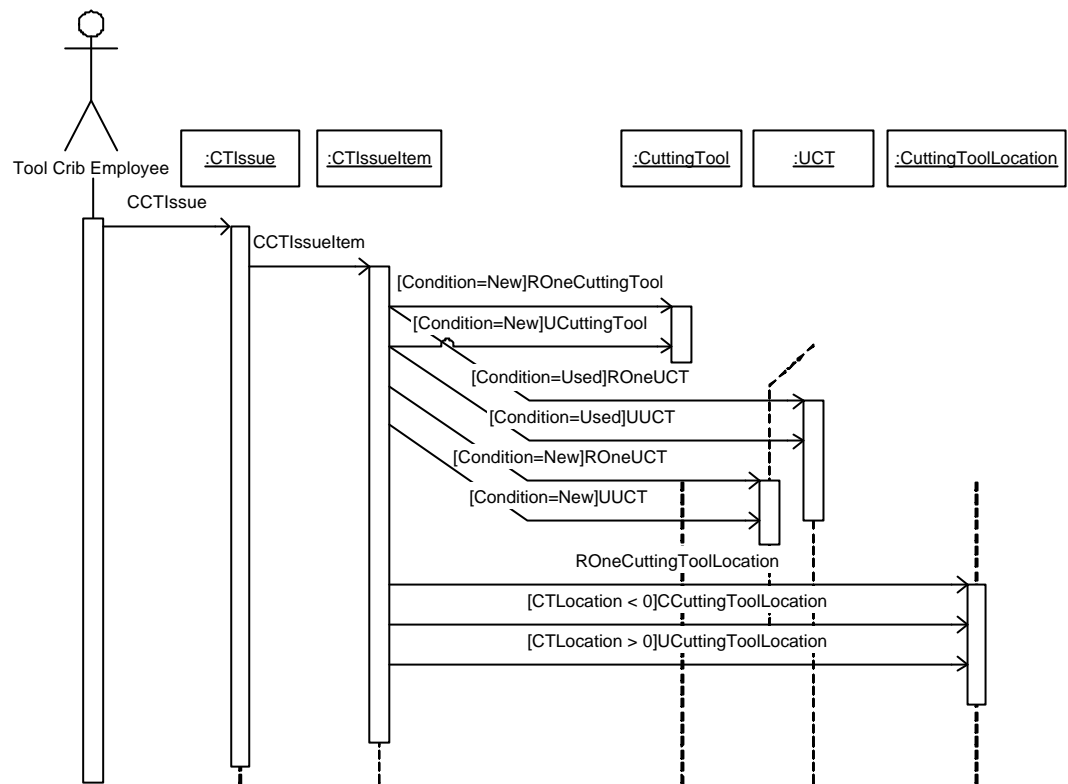


Figure A.15 Sequence Diagram of Cutting Tool Issue

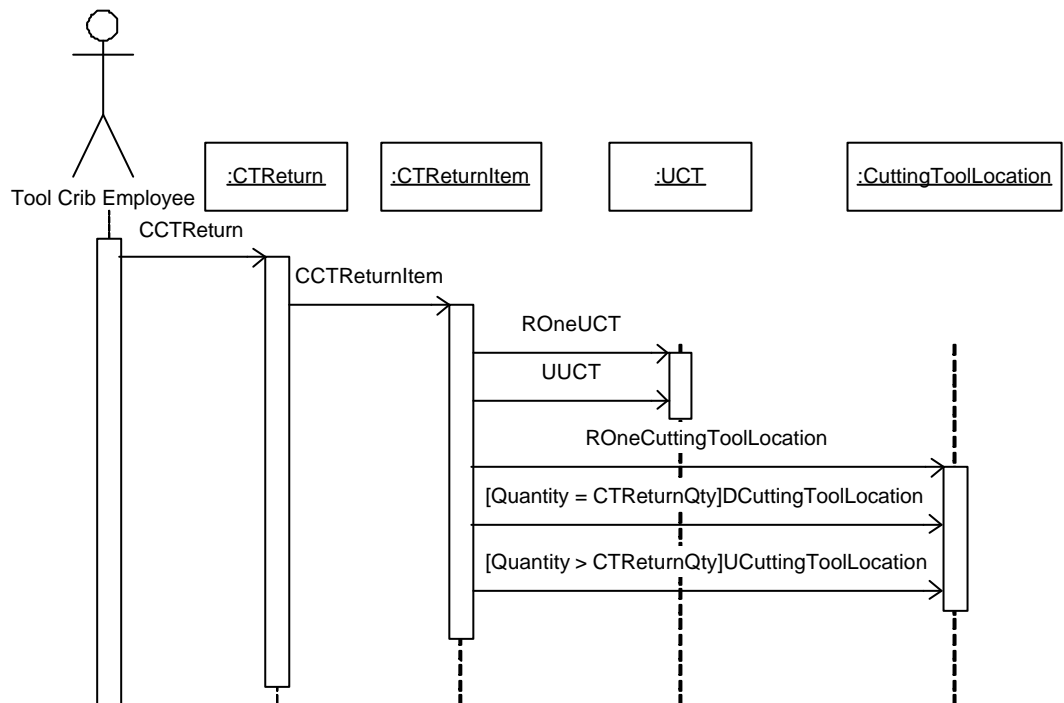


Figure A.16 Sequence Diagram of Cutting Tool Return

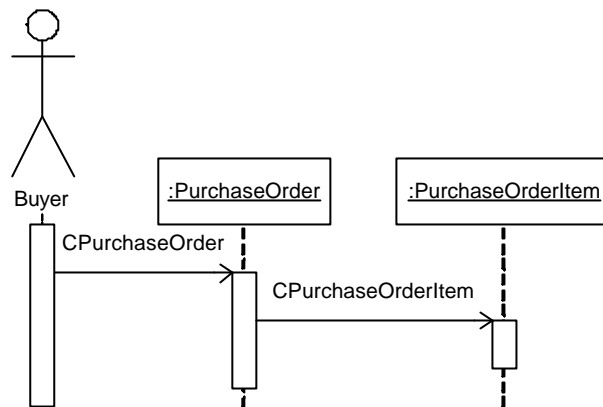


Figure A.17 Sequence Diagram of Purchase Order Creation

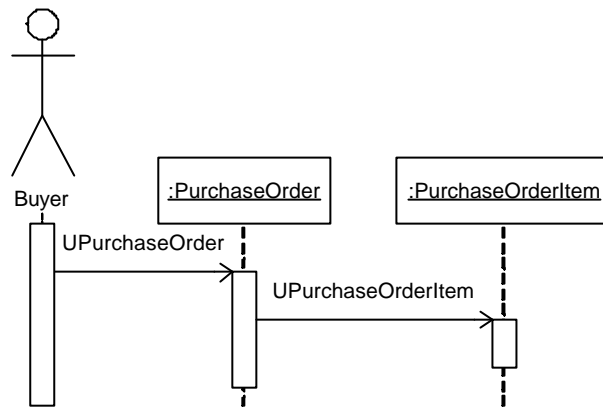


Figure A.18 Sequence Diagram of Purchase Order Update

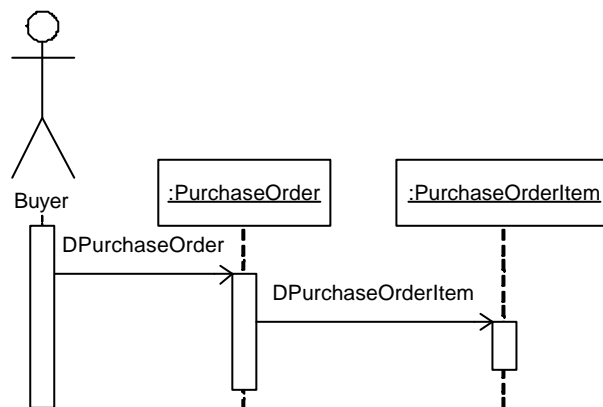


Figure A.19 Sequence Diagram of Purchase Order Deletion

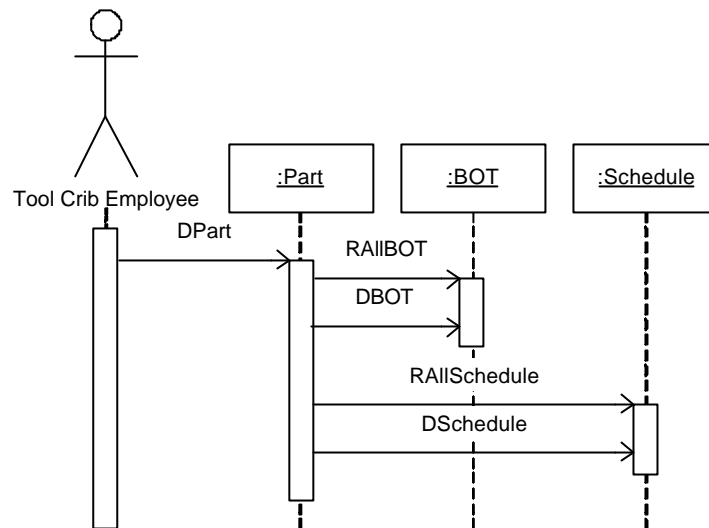


Figure A.20 Sequence Diagram of Part Deletion

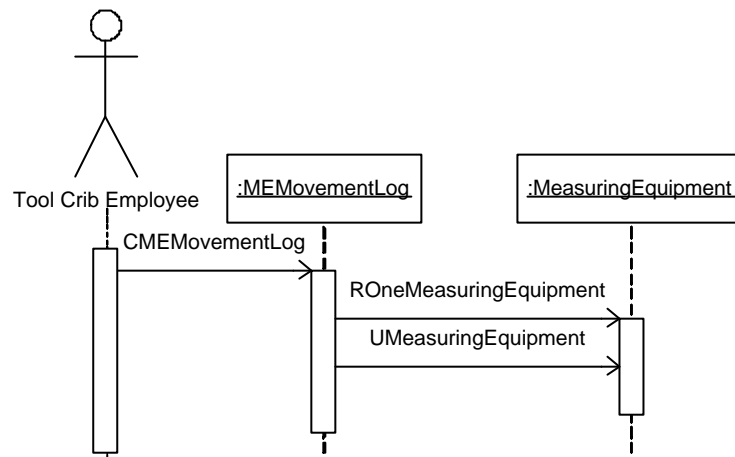


Figure A.21 Sequence Diagram of Measuring Equipment Movement

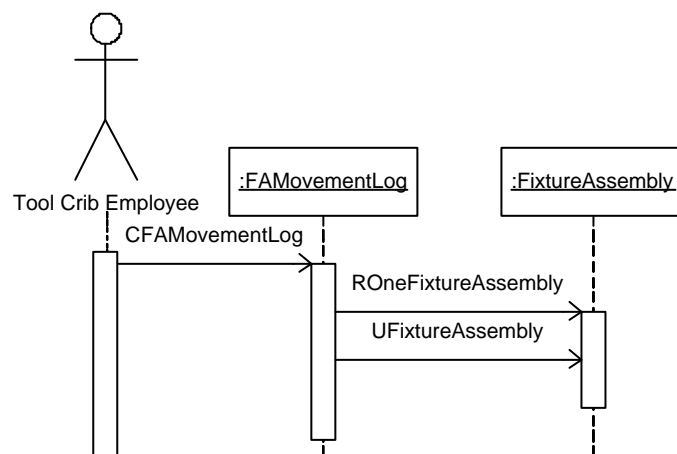


Figure A.22 Sequence Diagram of Fixture Assembly Movement

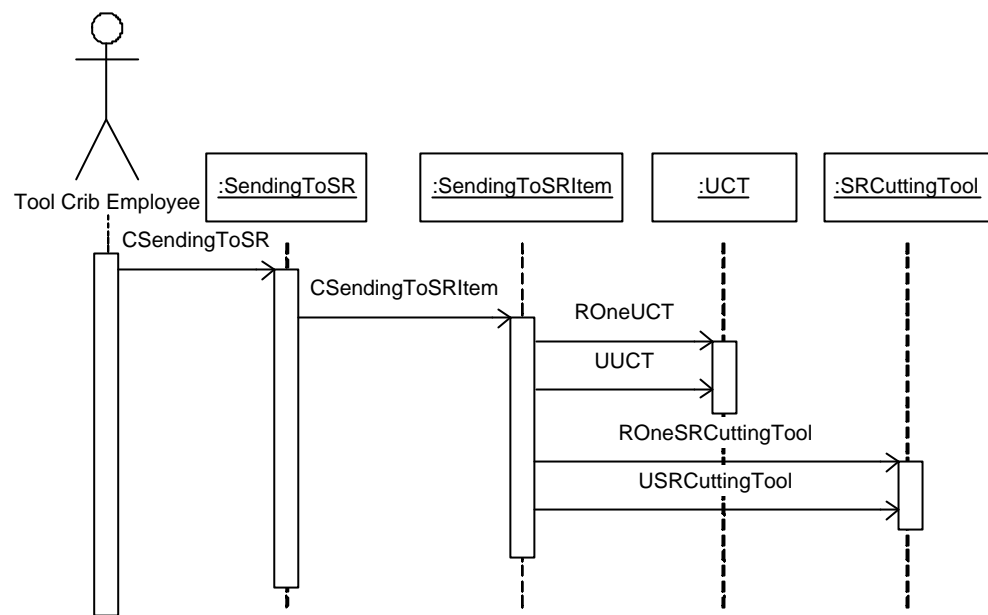


Figure A.23 Sequence Diagram of Sending to Sharpening Room

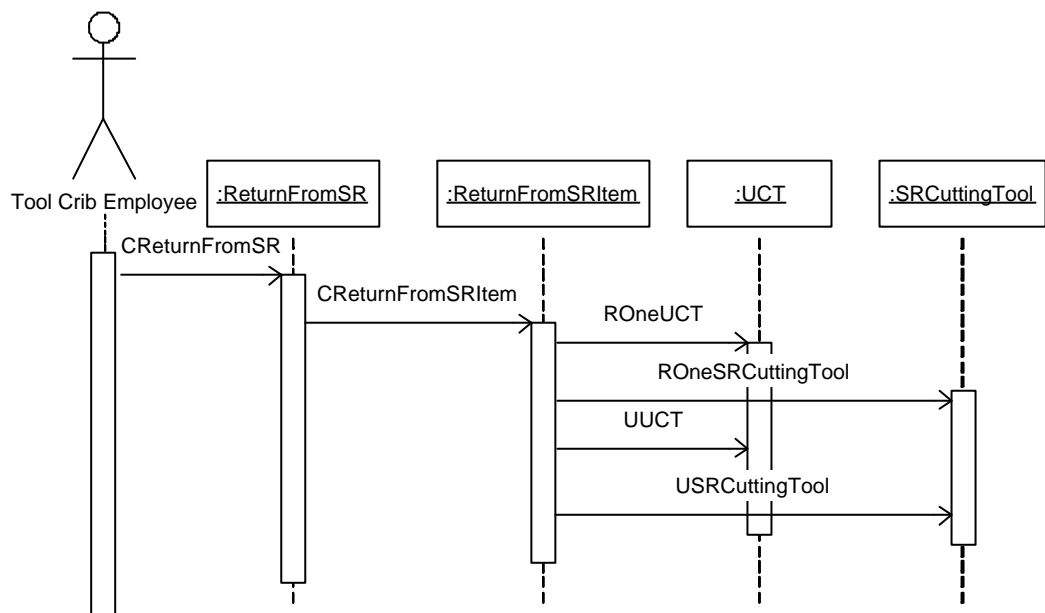


Figure A.24 Sequence Diagram of Return from Sharpening Room

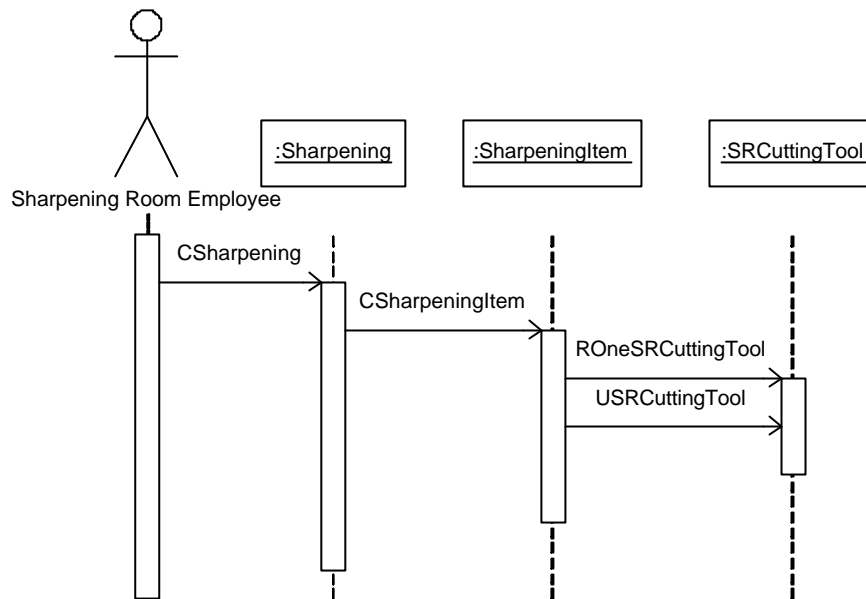


Figure A.25 Sequence Diagram of Sharpening of Cutting Tools

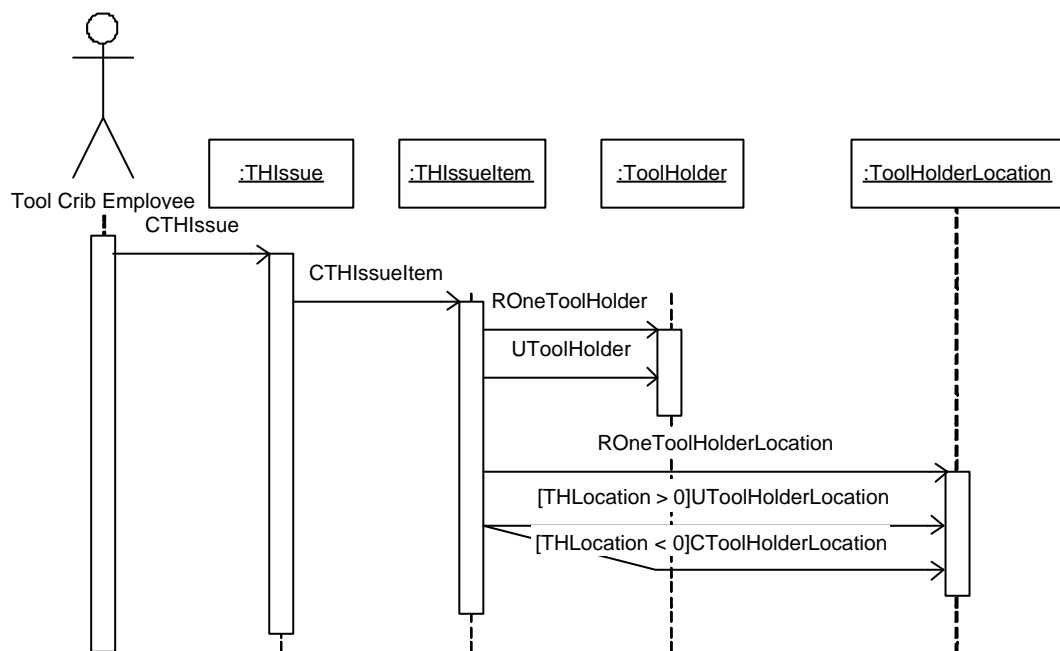


Figure A.26 Sequence Diagram of Tool Holder Issue

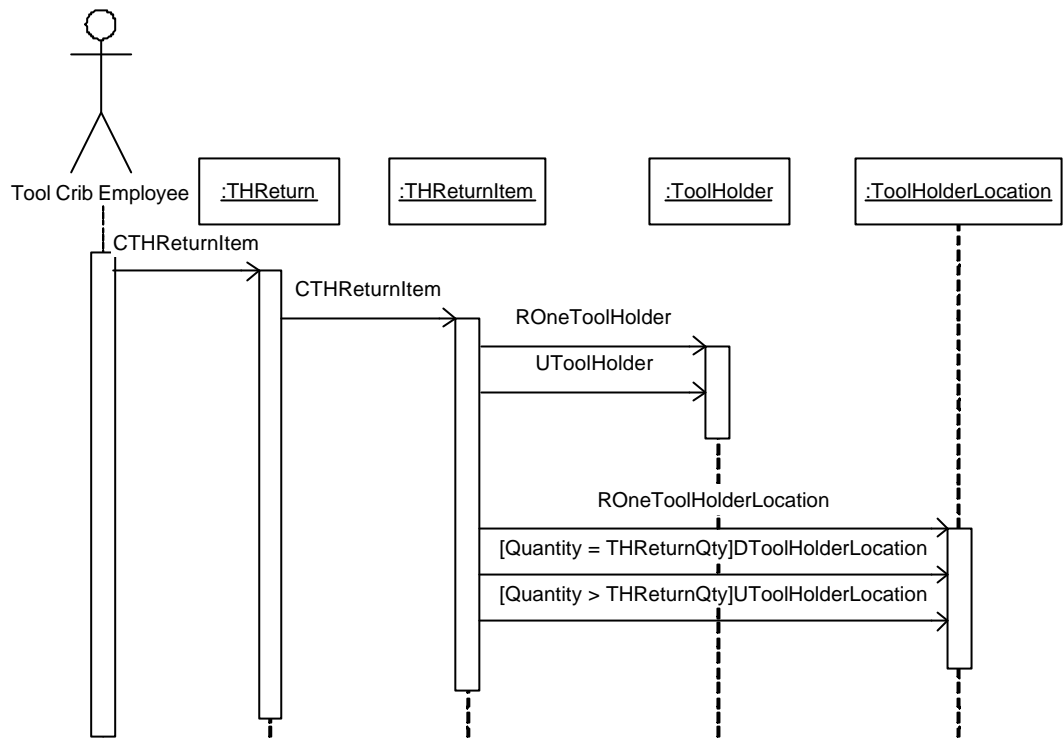


Figure A.27 Sequence Diagram of Tool Holder Return

A.4 Collaboration Diagrams

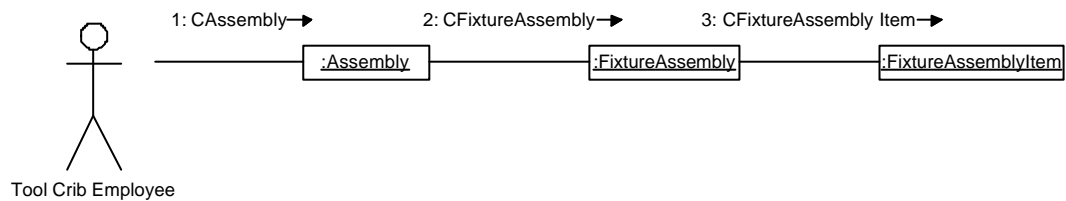


Figure A.28 Collaboration Diagram of Fixture Assembly Creation

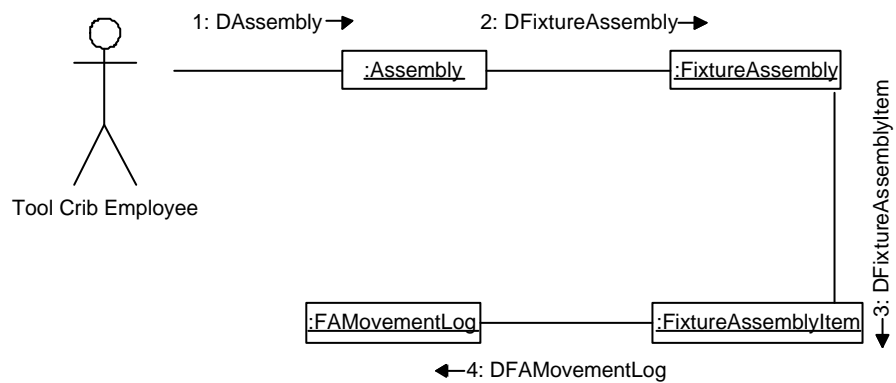


Figure A.29 Collaboration Diagram of Fixture Assembly Deletion

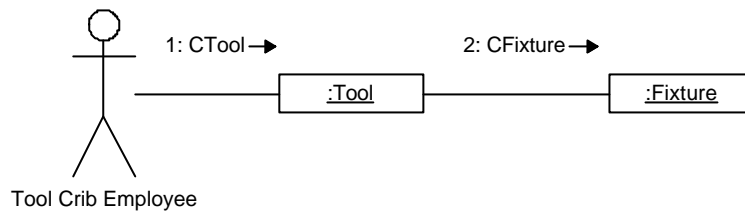


Figure A.30 Collaboration Diagram of Fixture Creation

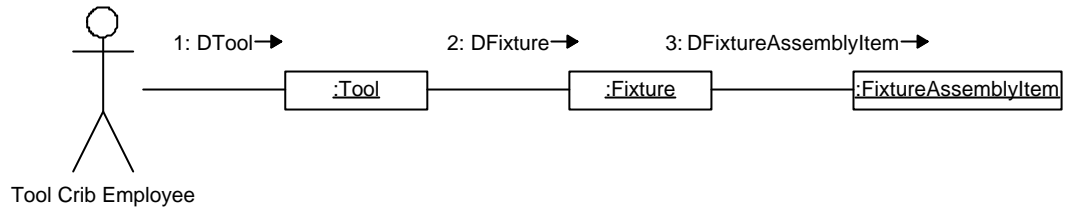


Figure A.31 Collaboration Diagram of Fixture Deletion

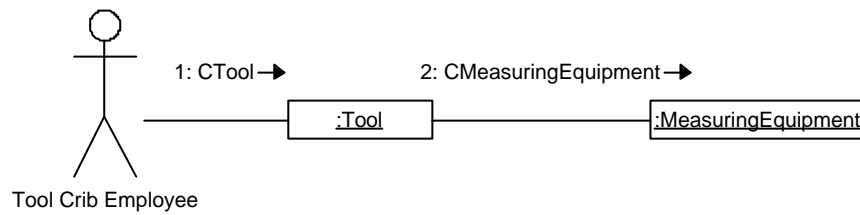


Figure A.32 Collaboration Diagram of Measuring Equipment Creation

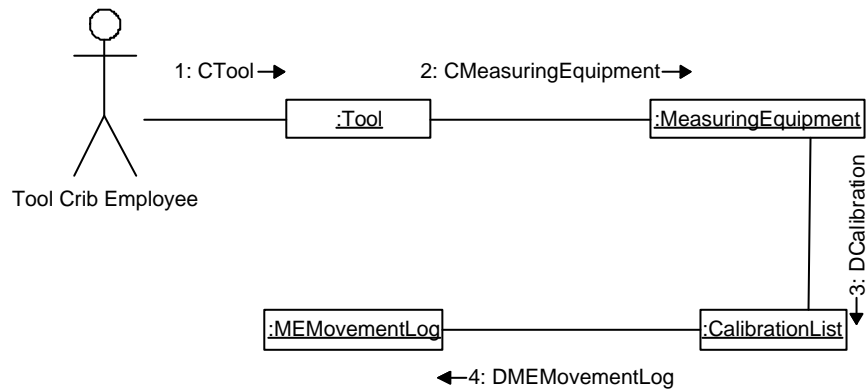


Figure A.33 Collaboration Diagram of Measuring Equipment Deletion

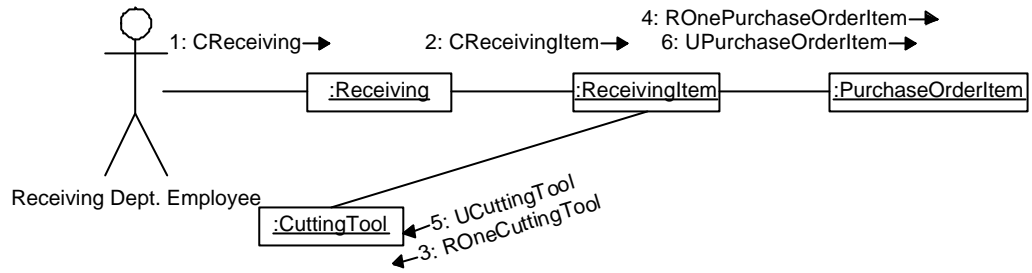


Figure A.34 Collaboration Diagram of Cutting Tool Receiving from Vendors

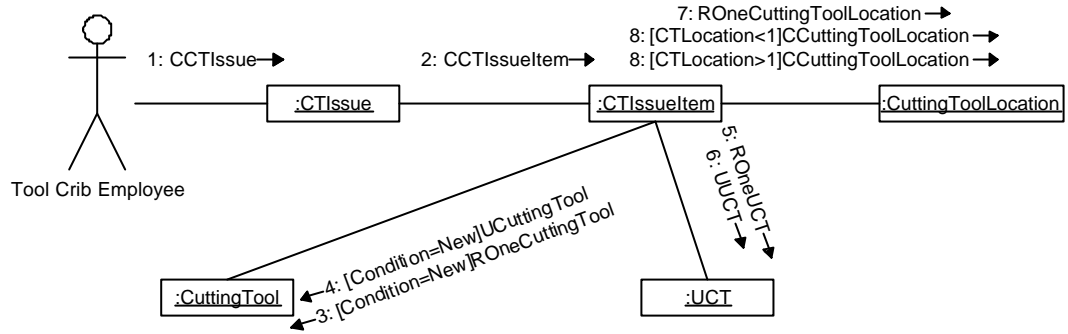


Figure A.35 Collaboration Diagram of Cutting Tool Issue

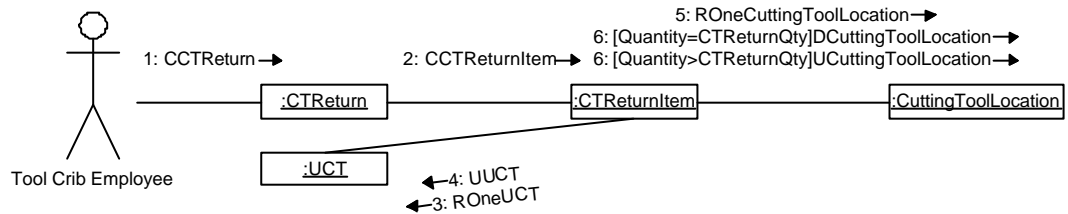


Figure A.36 Collaboration Diagram of Cutting Tool Return

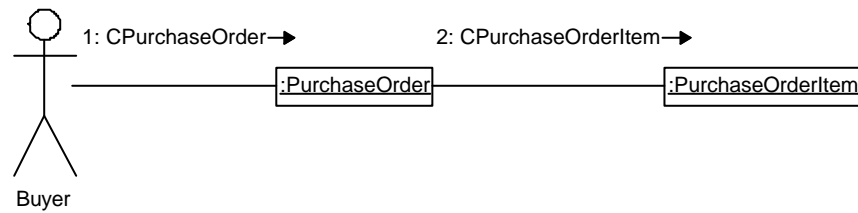


Figure A.37 Collaboration Diagram of Purchase Order Creation

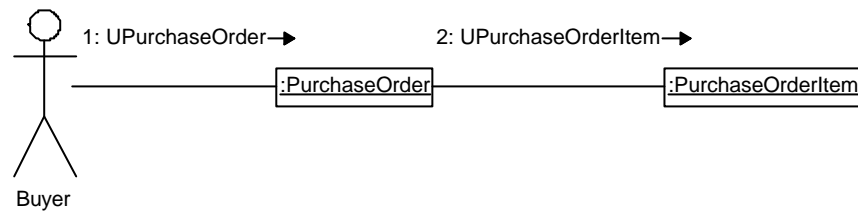


Figure A.38 Collaboration Diagram of Purchase Order Update

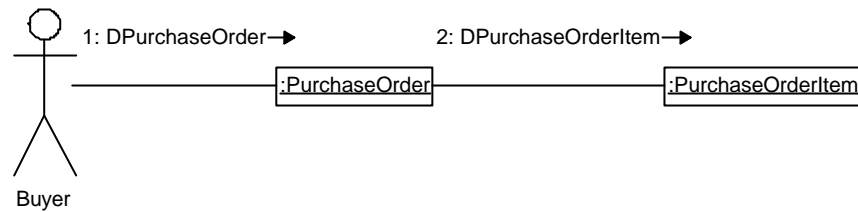


Figure A.39 Collaboration Diagram of Purchase Order Deletion

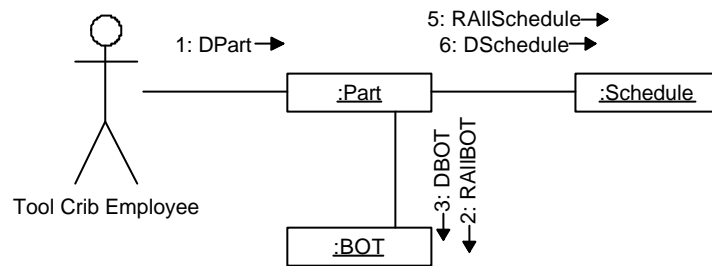


Figure A.40 Collaboration Diagram of Part Deletion

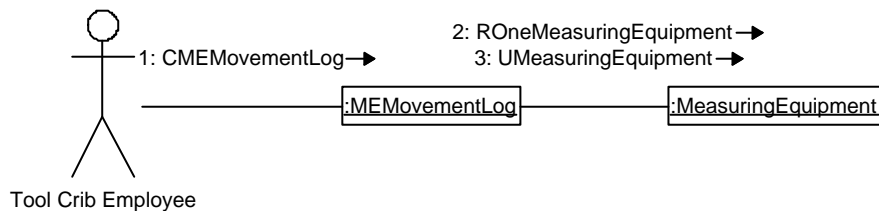


Figure A.41 Collaboration Diagram of Measuring Equipment Movement

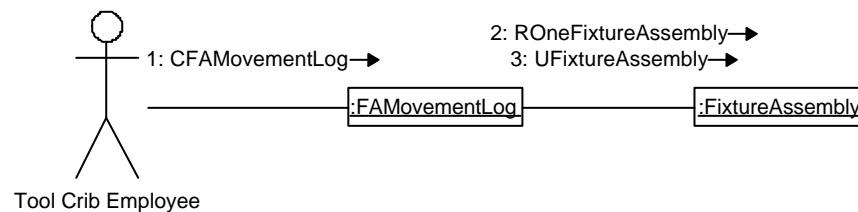


Figure A.42 Collaboration Diagram of Fixture Assembly Movement

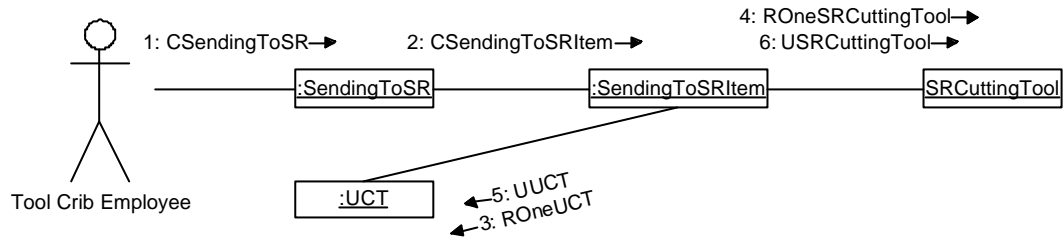


Figure A.43 Collaboration Diagram of Sending to Sharpening Room

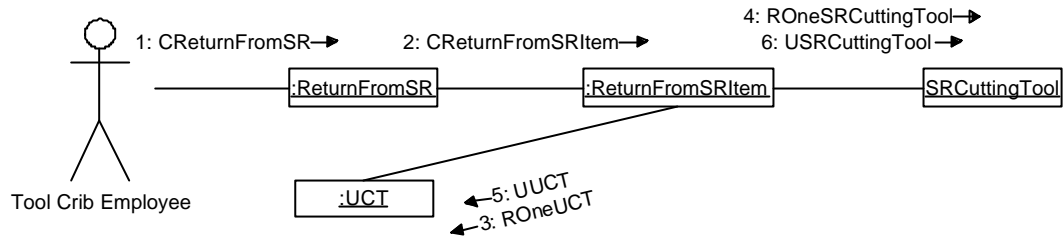


Figure A.44 Collaboration Diagram of Return from Sharpening Room

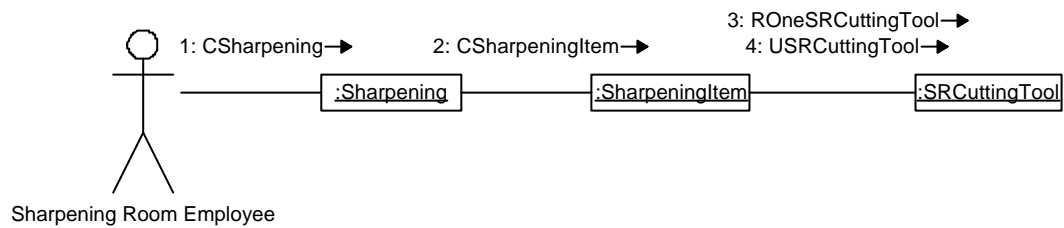


Figure A.45 Collaboration Diagram of Sharpening of Cutting Tool

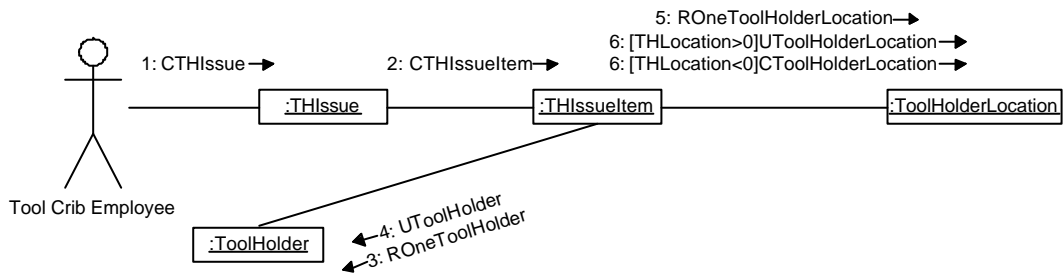


Figure A.46 Collaboration Diagram of Tool Holder Issue

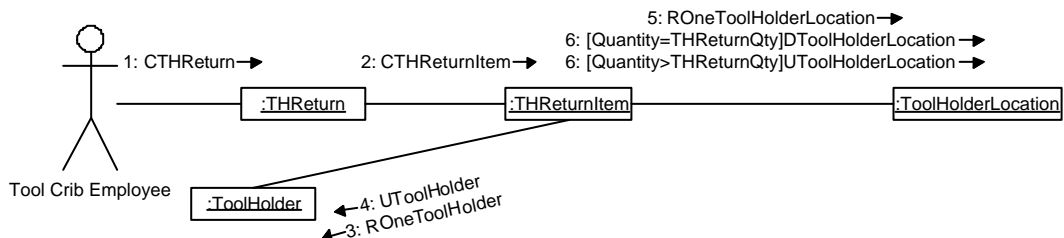


Figure A.47 Collaboration Diagram of Tool Holder Return

A.5 Class Diagram

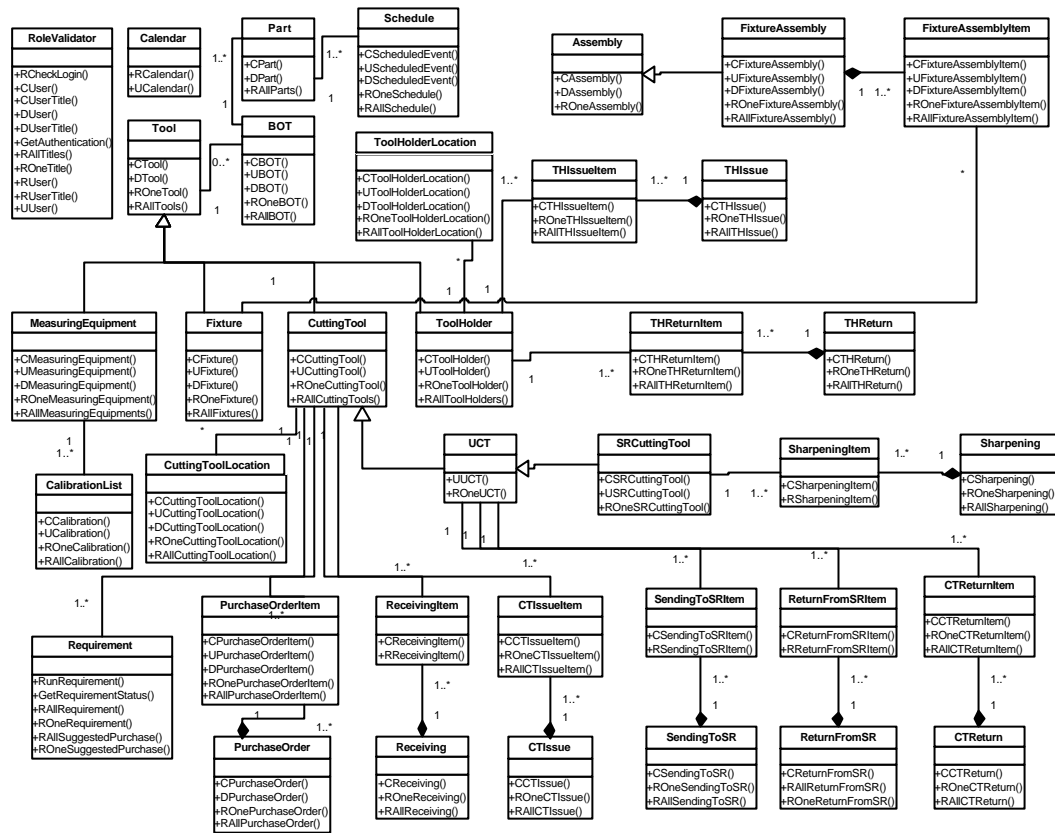


Figure A.48 Class Diagram

A.6 Package Diagram

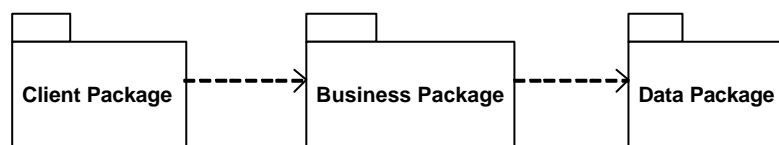


Figure A.49 Package Diagram

A.7 Component Diagrams

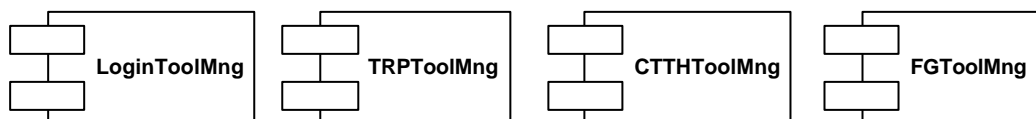


Figure A.50 Component Diagram of Business Package

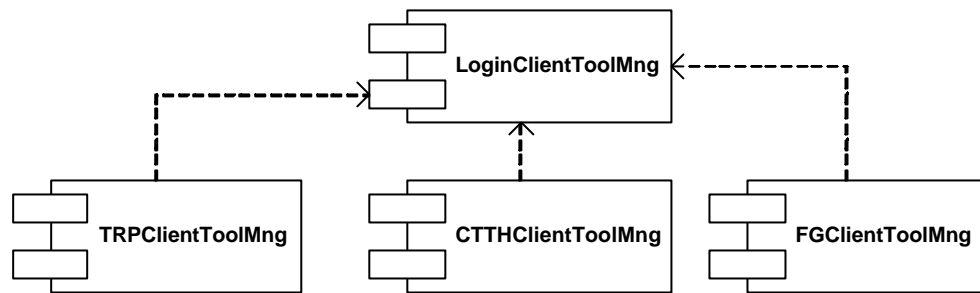


Figure A.51 Component Diagram of Client Package

A.8 Deployment Diagram

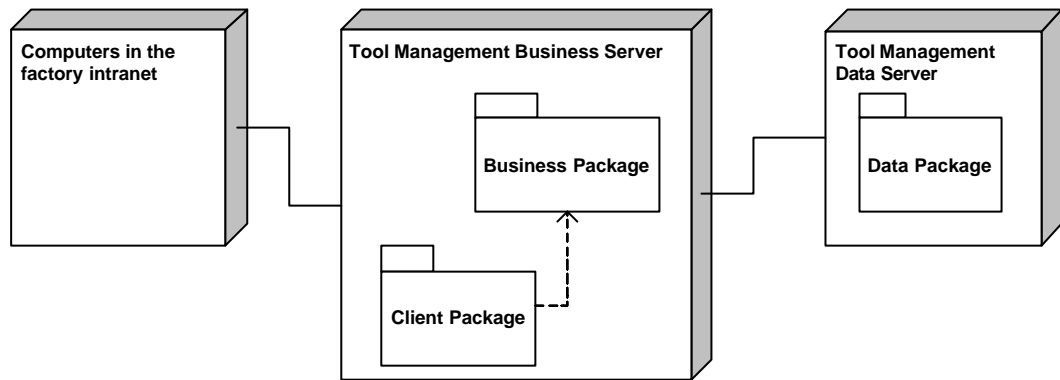


Figure A.52 Deployment Diagram

APPENDIX B

USERS MANUAL

The users manual of the tool management system will be supplied in this section. A brief explanation about how to use the system is given under the screen shots of the system interface. All users access the system through “Login” page, given in Figure B.1. User presses the “Submit” button after entering the user name and password. The next page is the “Welcome” page given in Figure B.2. User can navigate between modules using the links below. In each module, there are options in the left-hand side of the screen changing according to the authorization of the user.

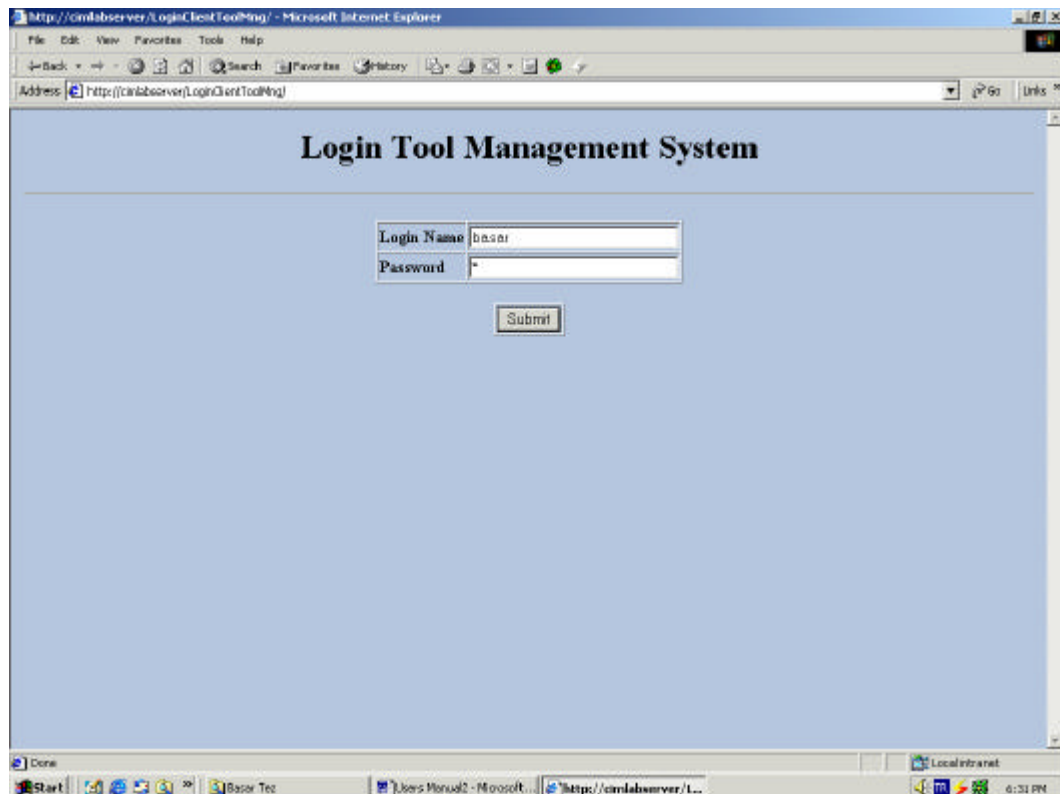


Figure B.1 Login Page



Figure B.2 Welcome Page

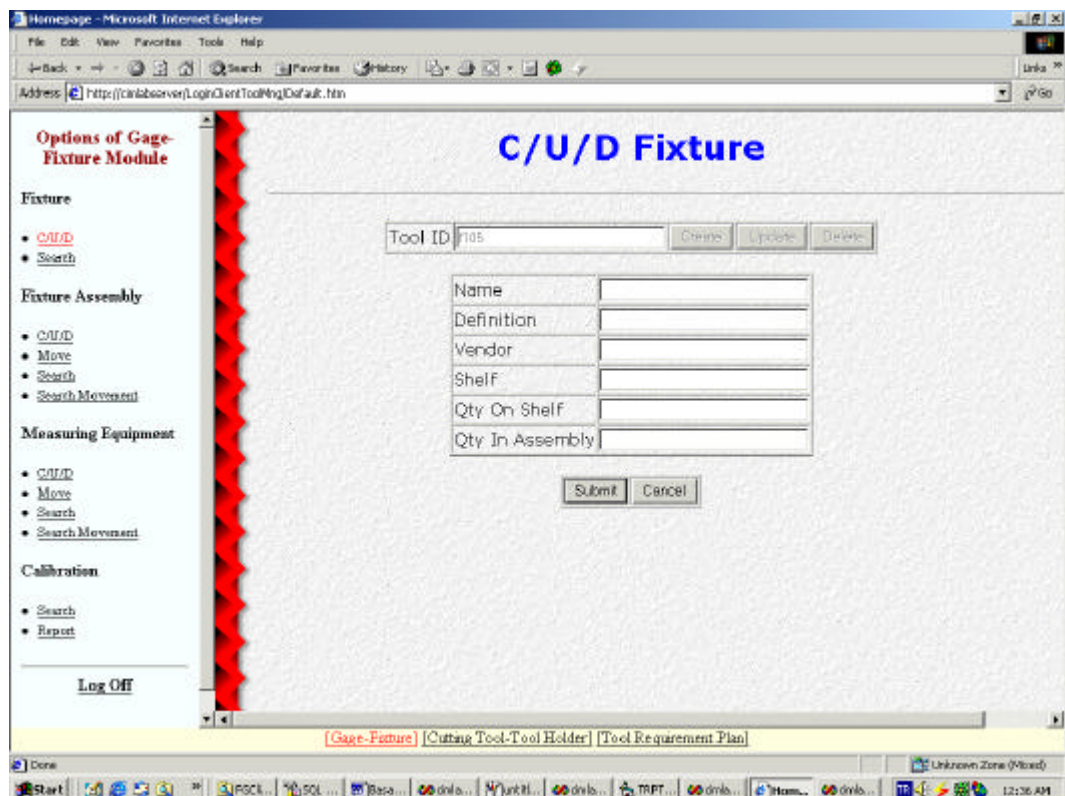


Figure B.3 C/U/D Fixture Page

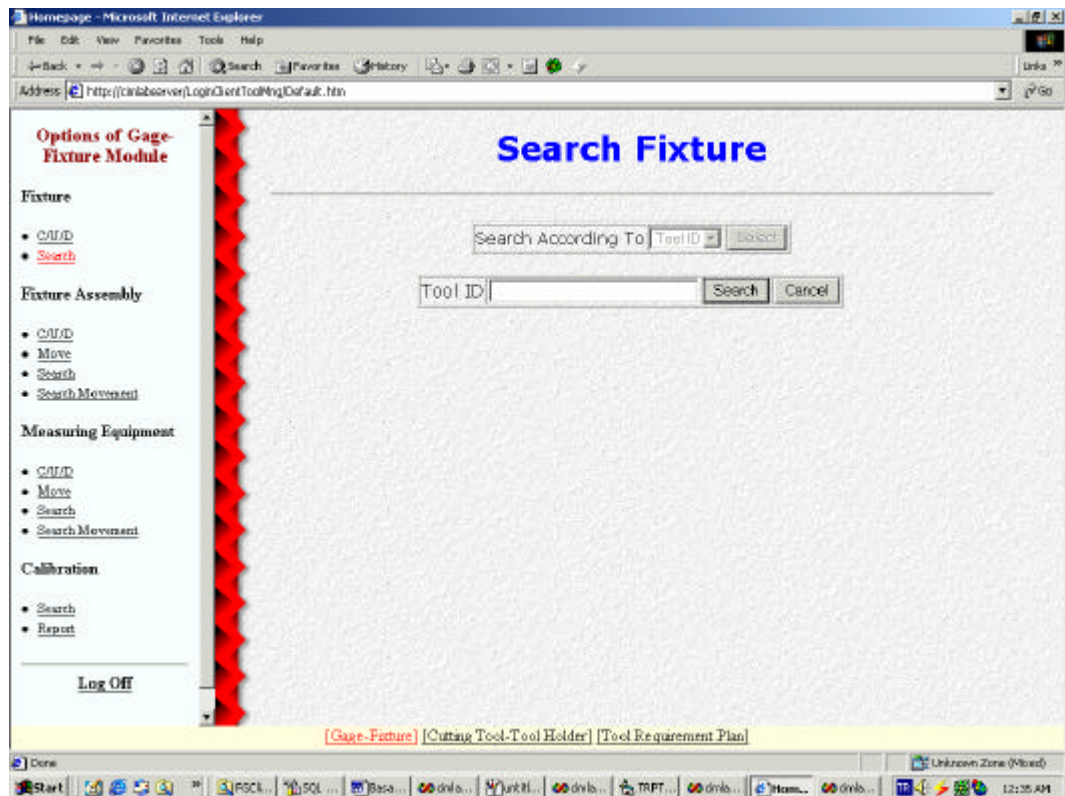


Figure B.4 Search Fixture Page

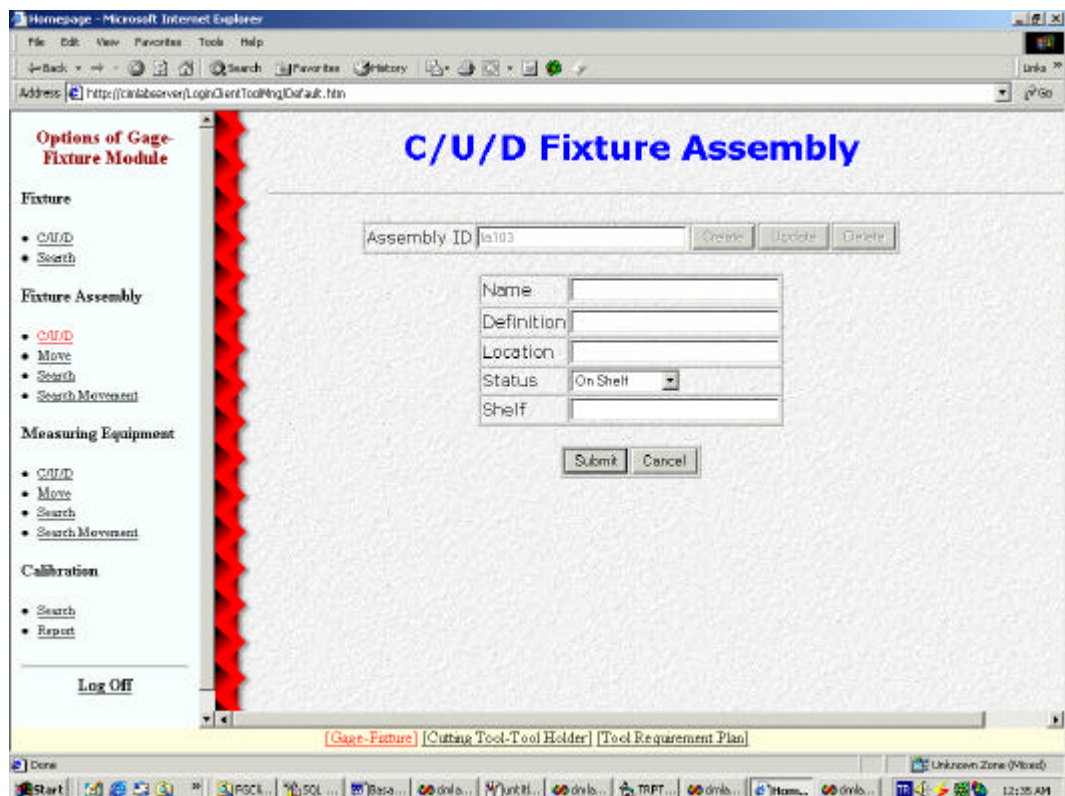


Figure B.5 C/U/D Fixture Assembly Page

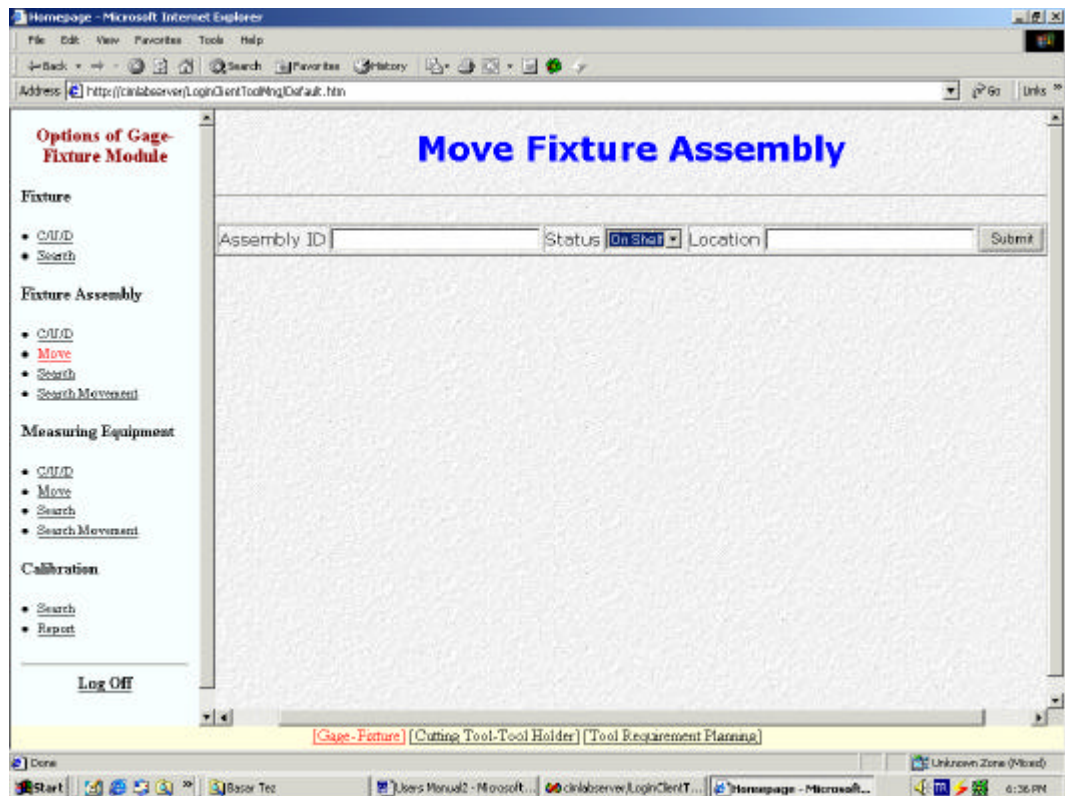


Figure B.6 Move Fixture Assembly Page

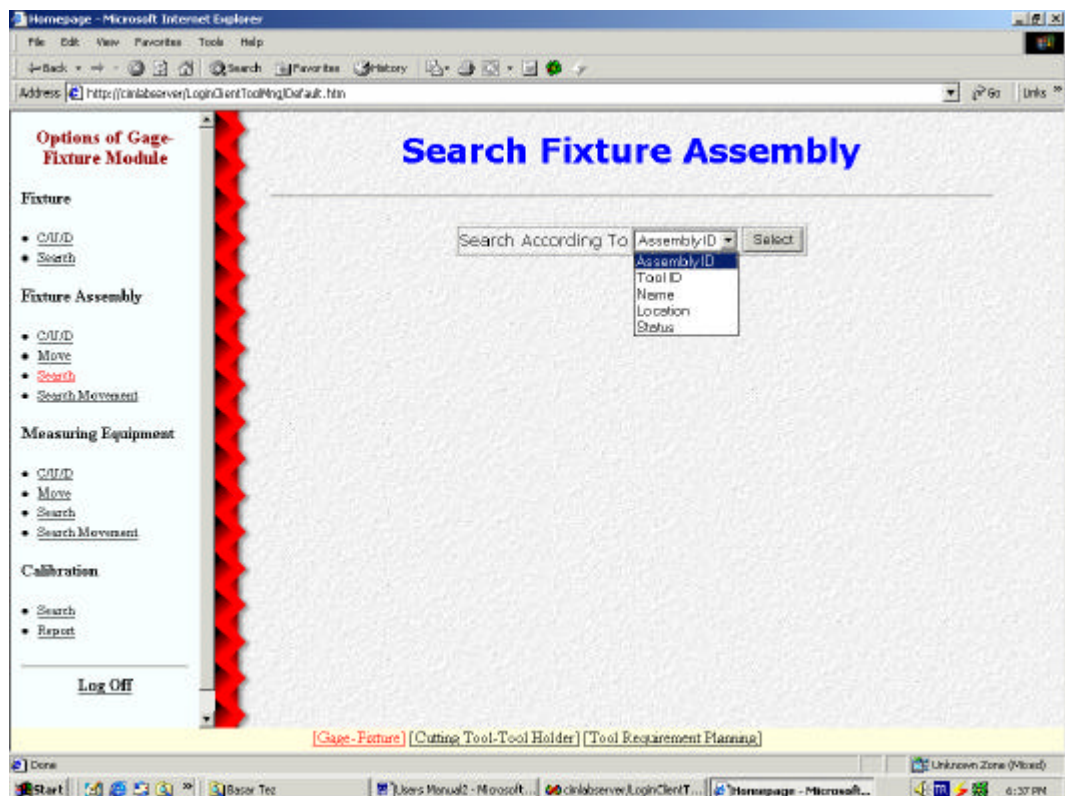


Figure B.7 Search Fixture Assembly Page

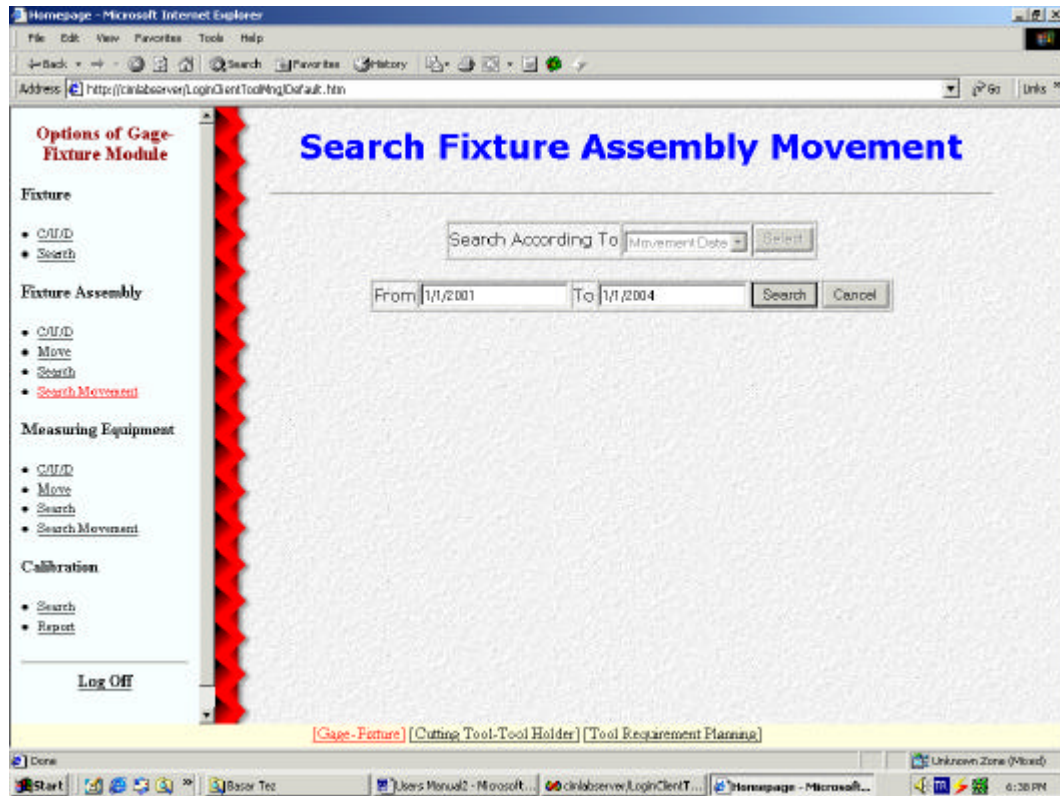


Figure B.8 Search Fixture Assembly Movement Page

Gage-Fixture module includes the pages from Figure B.3 to B.8. “C/U/D Fixture” given in Figure B.3 is used to create, update and delete fixtures. Tool ID is entered into the “Tool ID” field. If one of “Create” or “Update” buttons is clicked, the fields of data of the fixture is appeared. Data can be entered or updated in these fields. If “Delete” button is clicked, the data fields of the fixture is appeared but no change is allowed. The create, update, or delete operation is completed by clicking the “Submit” button, or can be canceled by clicking the “Cancel” button. In “Search Fixture” page, given in Figure B.4 , Fixtures can be searched according to their tool ID, name, or vendor. After clicking the “Select” button, the parameter according to which the search operation is executed can be entered into the textbox. Operation is executed by clicking the “Search”, or canceled by clicking the “Cancel” button. “C/U/D Fixture Assembly” page given in Figure B.5 is used to create, update, and delete fixture assemblies. Assembly ID of the assembly that will be created, updated or deleted is entered into the “Assembly ID” field. If one of the “Create” or “Update” buttons is clicked, the

data fields of the assembly is appeared first. After clicking “Submit” button, the tool ID’s and the quantities of the fixtures that will be assembled are entered into the “Tool ID” and “Tool Qty” fields. “Assemble” button is used to assemble the fixture, “Dismantle” button is used to dismantle the fixture in the assembly, “Update_Item” is used to update the quantity of a fixture in the assembly. “Cancel” buttons are used to cancel the operation. In “Move Fixture Assembly” page, given in Figure B.6, fixture assembly is moved. User supplies the assembly ID of the assembly, chooses the status as “On Shelf” or “In Use”. After the status is chosen, new location of the fixture assembly is entered. “Submit” button records the movement. In “Search Fixture Assembly” page, given in Figure B.7, Fixture assemblies can be searched according to assembly ID, tool ID of the fixtures in the fixture assembly, name, location, and status. Smilar to the “Fixture Search” page, the parameter is written into the appearing textbox or chosen from the listbox after clicking the “Select” button. The list of the fixtures in the assembly can be reached through the link written as the assembly ID of the fixture assembly. In “List Fixture Assembly” page, fixture data can be reached through the link given as the tool ID of the fixture in the fixture assembly. In “Search Fixture Assembly Movement” page, given in Figure B.8, fixture assemblies’ movements can be searched according to assembly ID, movement date, location, status and moved by. After the search parameter is entered, “Search” button is clicked to get the search result, or “Cancel” button is clicked to cancel the operation.

Measuring equipment operations are similar to fixture and fixture assembly operations. Some attributes are different only. Attributes of the measuring equipment are different than fixtures. Besides “On Shelf”, “In Use”, and “Not Available”, “In Calibration” is added to the status list. Also in “Search Measuring Equipment” page, search can be done according to range and sensitivity additionally. The screenshots of measuring equipment pages can be seen in Figure B.10 to Figure B.12.

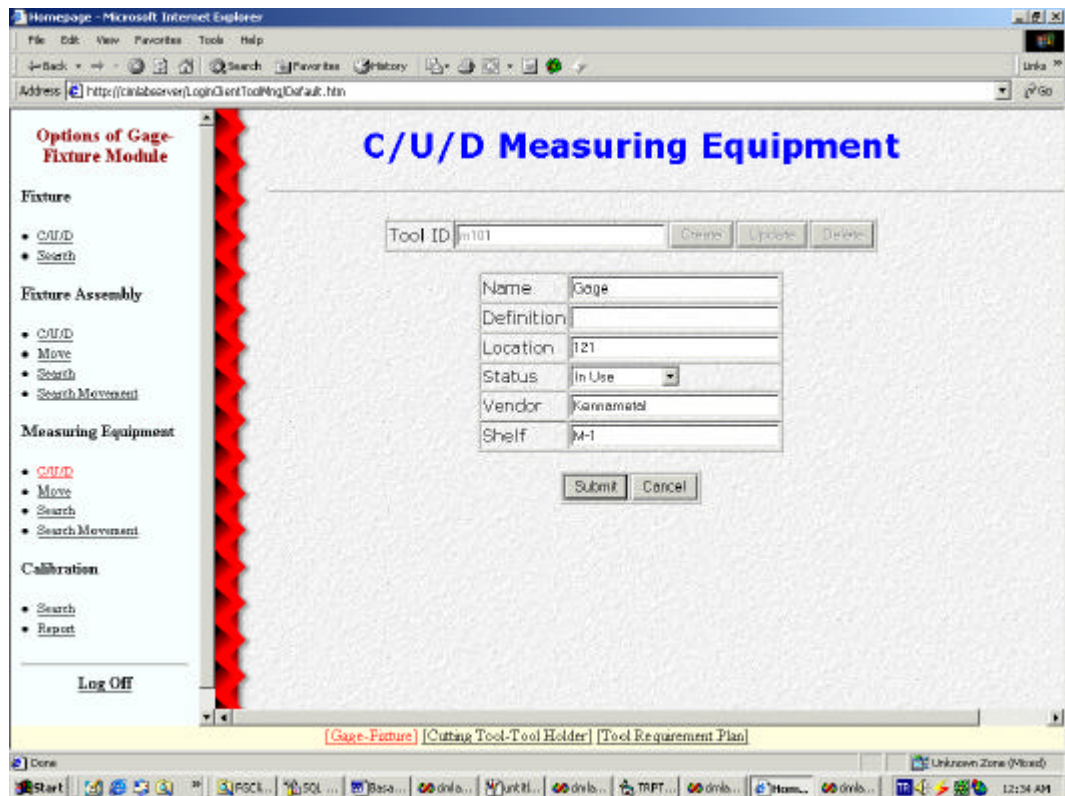


Figure B.9 C/U/D Measuring Equipment Page

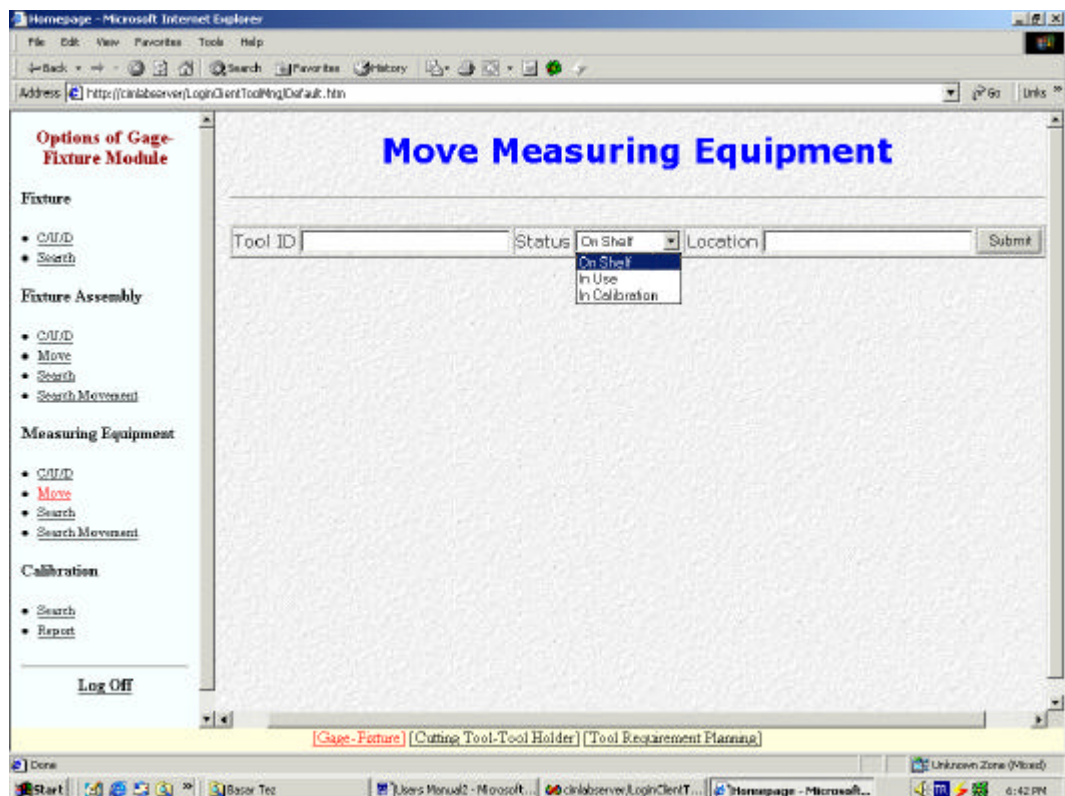


Figure B.10 Move Measuring Equipment Page

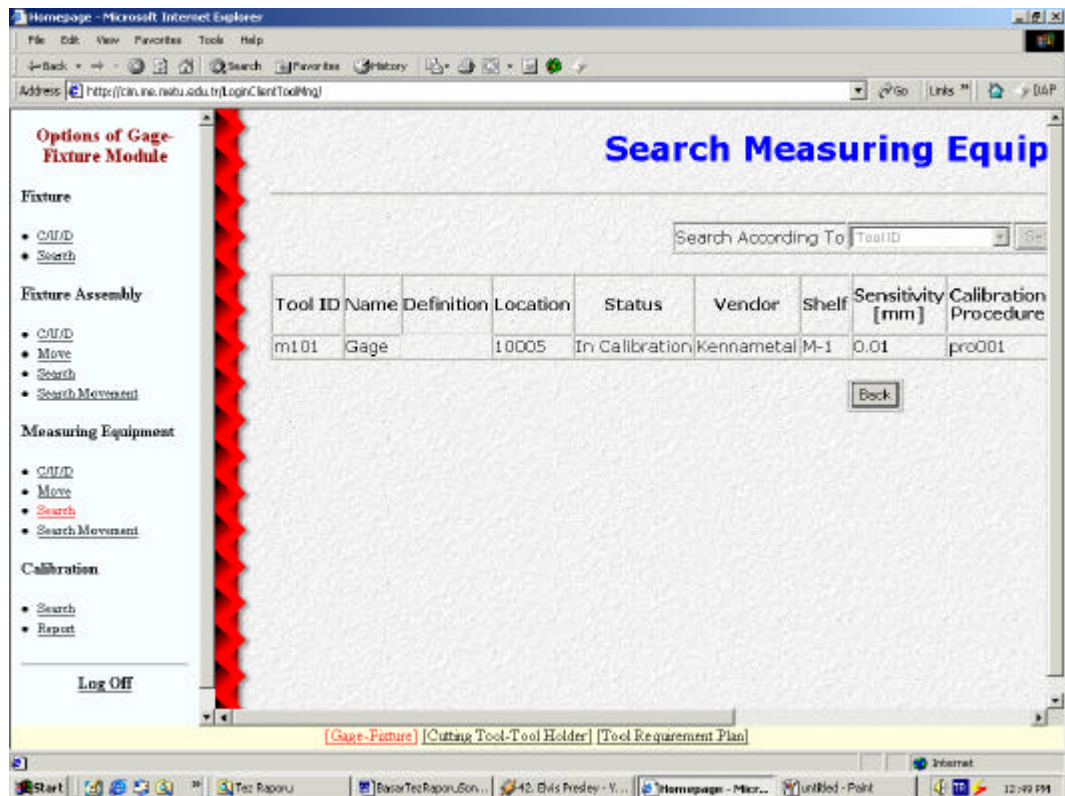


Figure B.11 Search Measuring Equipment Page

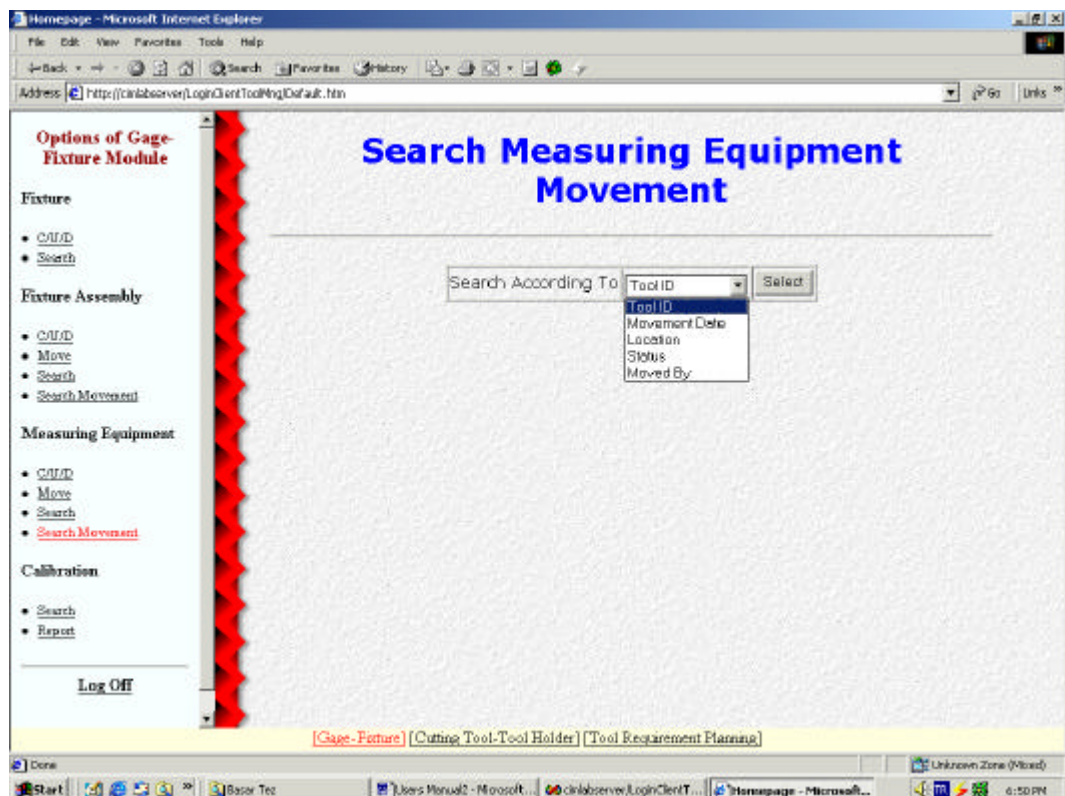


Figure B.12 Search Measuring Equipment Movement Page

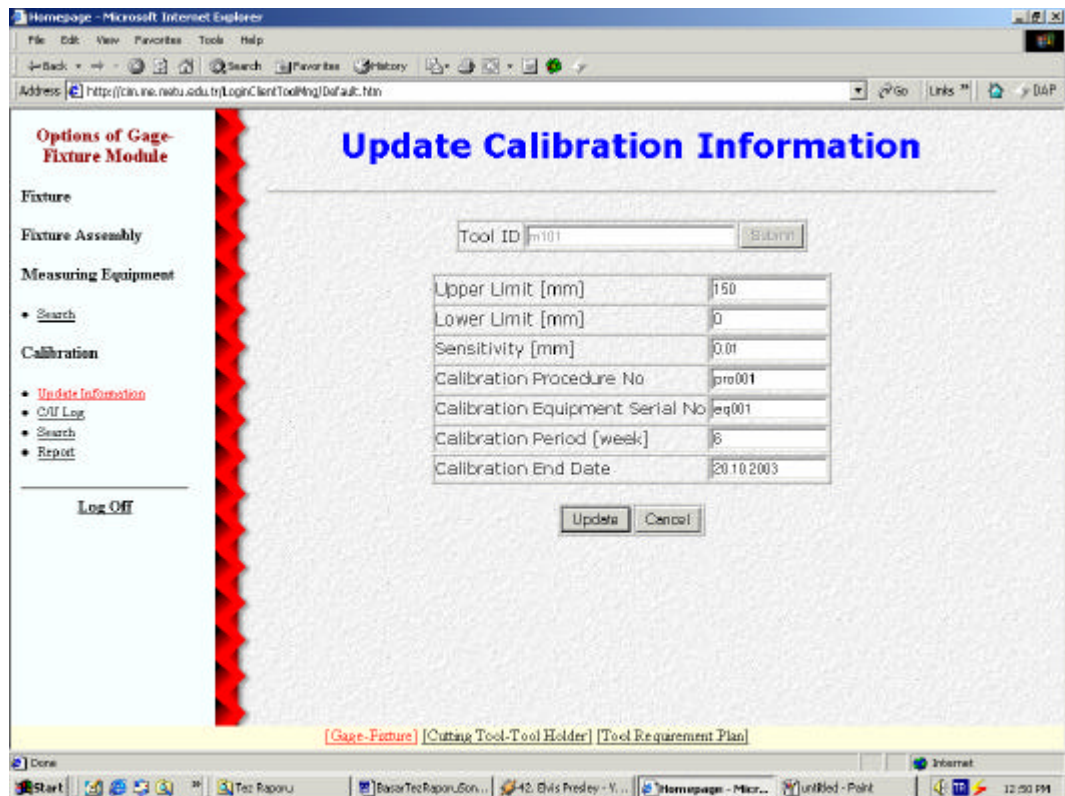


Figure B.13 Update Calibration Information Page

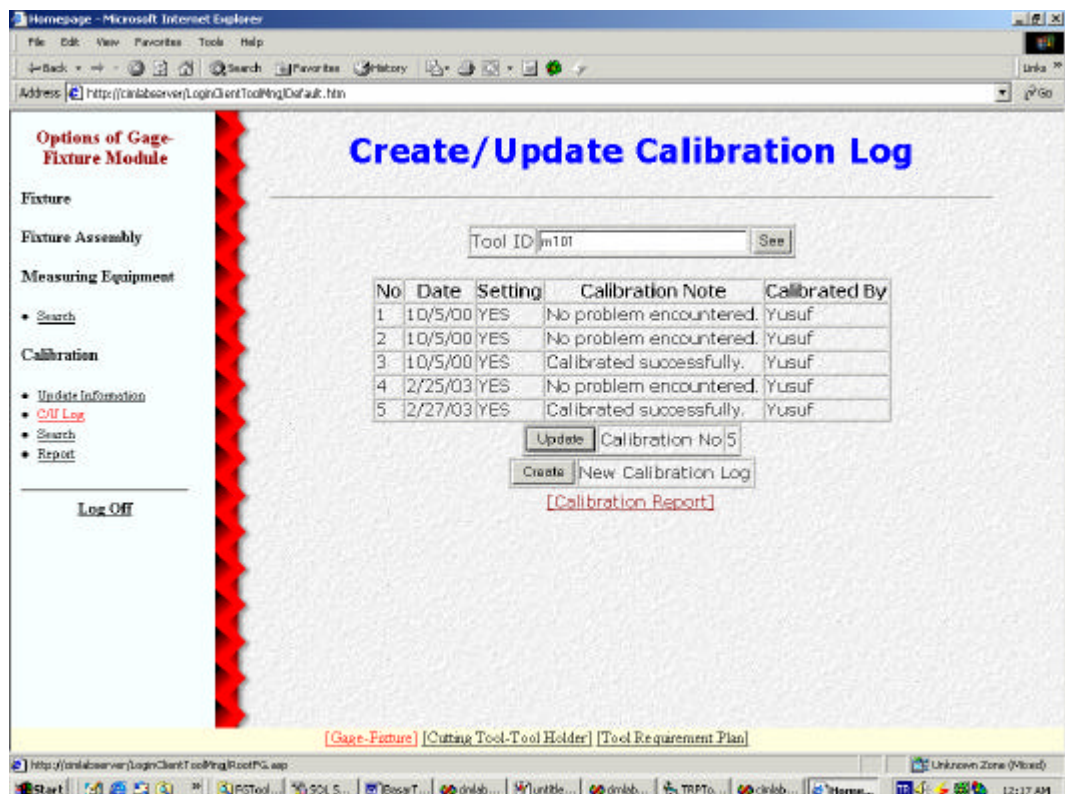


Figure B.14 Create/Update Calibration Log

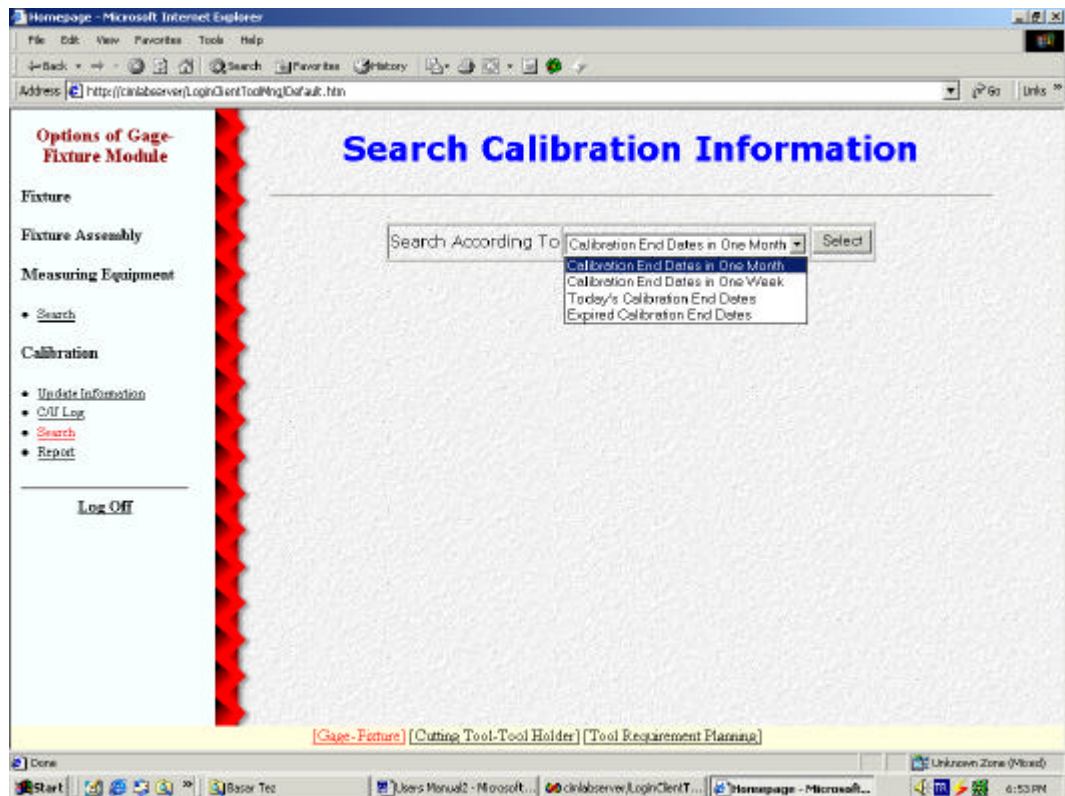


Figure B.15 Search Calibration Information Page

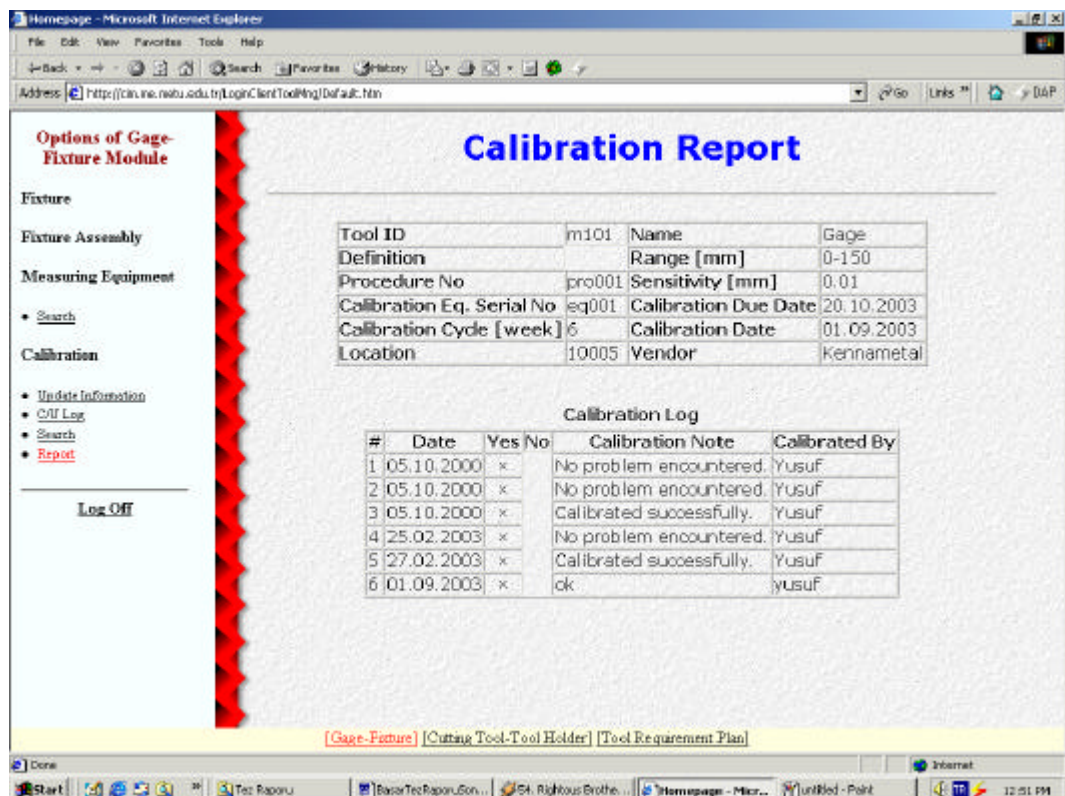


Figure B.16 Calibration Report Page

In the calibration section, calibration attributes of a measuring equipment can be changed in “Update Calibration Information” page, given in Figure B.13. In this page, “Submit” button is clicked after the tool ID is supplied. The data about calibration information can be entered into the fields and recorded by clicking the “Update” button. In the “Create/Update Calibration Log” page, given in Figure B.14, the list of previous calibrations appears after clicking “See” button. “Update” button updates the record of the last calibration. “Create” button creates a new calibration log. The “Calibration Report” link reaches the calibration attributes and calibration log of the measuring equipment. In the “Search Calibration Information” page, given in Figure B.15, user can list the tools which have one month to calibration end date, one week to calibration end date, whose calibration end date is the present day, and list the tools whose calibration end date is passed. “Select” button lists the tools according to the search criterion selected. “Calibration Report” page, given in Figure B.16, allows the user to reach the calibration report of the measuring equipment whose tool ID is entered into the “Tool ID” field.

Cutting tool-tool holder module includes the pages from Figure B.18 to Figure B.37. In “C/U Cutting Tool ” page, given in Figure B.18, tool ID of the cutting tool that will be created or updated is entered into the “Tool ID” field. “Create” button creates a new cutting tool in the database and attributes of it is entered. “Update” button updates a cutting tool data in the database. “Submit” button records the changes into the database. “Cancel” button cancels the operation. In “Search Cutting Tool” page, given Figure B.19, cutting tools in the inventory can be searched according to tool ID, name, vendor, location and the inventory under safety stock level. Similar to other search pages, parameters are entered into the field that appears after clicking the “Select” button. Locations of the cutting tool on the shop floor with its attributes appear after clicking the “Submit” button. “Cancel” button cancels the operation. In “Search Used Cutting Tool” page, given in Figure B.20, is used to search the used cutting tools’ conditions and quantities. After entering the tool ID of the cutting tool that will be searched, “Submit” button is clicked. Numbers of sharp, blunt, scrap, problematic,

and used cutting tools in use are retrieved. In “Search Sharpening Room Cutting Tool” page, given in Figure B.29, cutting tools in the sharpening room can be searched. Number of sharpened, blunt, problematic and scrap tools can be retrieved by entering the tool ID of the cutting tool and clicking the “Submit” button.

In “Issue Cutting Tool” page, given in Figure B.21, cutting tools are issued to the shop floor. Worker badge no of the worker to whom the cutting tools will be issued is entered into the “Issued To” field and “Create” button is clicked. Then tool ID, condition, and the quantity of the cutting tool is entered into the appropriate fields. If there are used tools, system does not allow to issue new tools. “Submit” button records the issue list item. “Cancel” button finishes that issue operation. In “Search Cutting Tool Issue” page, given in Figure B.22, cutting tool issues can be searched according to issued to, issued by, issue date, issue no, tool ID, and condition. After choosing the search criterion and clicking the “Select” button, the criterion parameter is entered into the field appearing. “Submit” button brings the search result. “Cancel” button cancels the search. Issue number link in the first column of the search result table accesses to the issue list. Cutting tool link in the issue list page accesses to the cutting tool attributes. In “Return Cutting Tool” page, given in Figure B.23, user can record the cutting tool returns from the shop floor. Worker badge number of the worker returning the cutting tools is entered into the “Returned By” field and “Create” button is clicked. Tool ID, condition, and quantity is entered into the appropriate fields. Note about the cutting tool can also be added if desired. “Submit” button records the return into the database and “Cancel” button cancels the operation. “Search Cutting Tool Return” page, given in Figure B.24 is similar to the “Search Cutting Tool Issue” page. Cutting tool returns can be searched according to returned by, returned to, return date, return no, condition, and tool ID. Return number link in the first column of the search result table accesses to the return list. Cutting tool link in the return list page accesses to the cutting tool attributes. In “Send to Sharpening Room” page, given in Figure B.25, user can send the cutting tools that need to be sharpened to the sharpening room. Name of the sharpening room

employee is entered into “Given To” field. After clicking “Create” button, tool ID and the quantity of the cutting tool is entered. Note also can be added. Only blunt tools can be sent to the sharpening room. “Submit” button records the sending operation. In “Search Sending To Sharpening” page, given in Figure B.26, sending to sharpening room records can be searched according to sent by, given to, sending date, sending no, and tool ID similar to the previous search pages. Sending number link in the first column of the search result table accesses to the sending list. Cutting tool link in the sending list page accesses to the cutting tool attributes. In “Return From Sharpening Room” page, given in Figure B.27, name of the sharpening room employee is entered into “Returned By” field, and “Create” button is clicked. Tool ID, condition, and quantity of the cutting tool is entered. Note can be added also. Returning cutting tool condition can be “Sharp”, “Problematic”, or “Scrap”. In “Search Return From Sharpening” page, given in Figure B.28, returned cutting tool records can be searched according to returned to, returned by, return date, return no, condition, and tool ID. Return number link in the first column of the search result table accesses to the return list. Cutting tool link in the return list page accesses to the cutting tool attributes. In “Sharpen Cutting Tool” page, given in Figure B.30, user can sharpen cutting tools by clicking the “Create” button first. Then, tool ID, condition, and quantity of the cutting tool sharpened is entered. Condition can be “Sharp”, “Problematic”, or “Scrap”. Note can also be added. “Submit” button records the sharpening operation into the database. In “Search Sharpening” page, given in Figure B.31, sharpening activities can be searched according to sharpened by, sharpening date, sharpening no, result and tool ID. Result can be sharp, problematic or scrap.

Tool holder options are used in a way similar to the cutting tool options. They don’t have conditions and do not have critical stock level since tool holders are not perishable tools. Also data fields of tool holders are different than the cutting tools. Search options are same except tool holders can not be searched according to condition. Tool holder options are given in figures from B.32 to B.37.

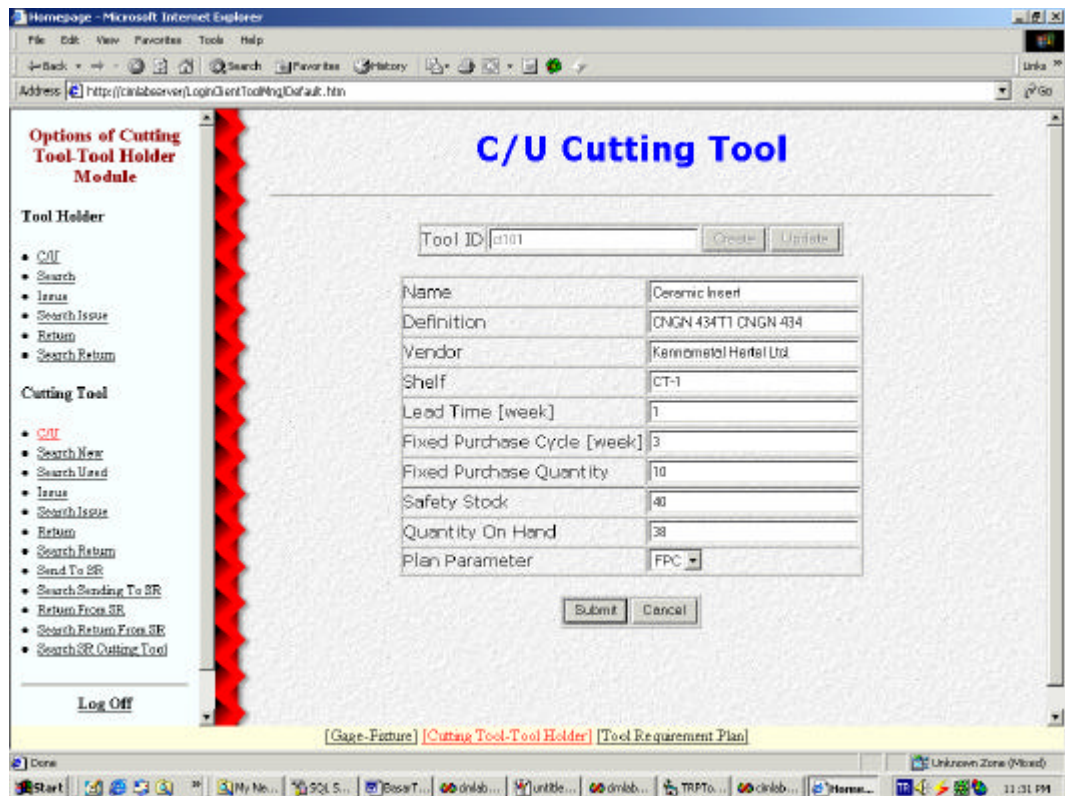


Figure B.18 C/U Cutting Tool Page

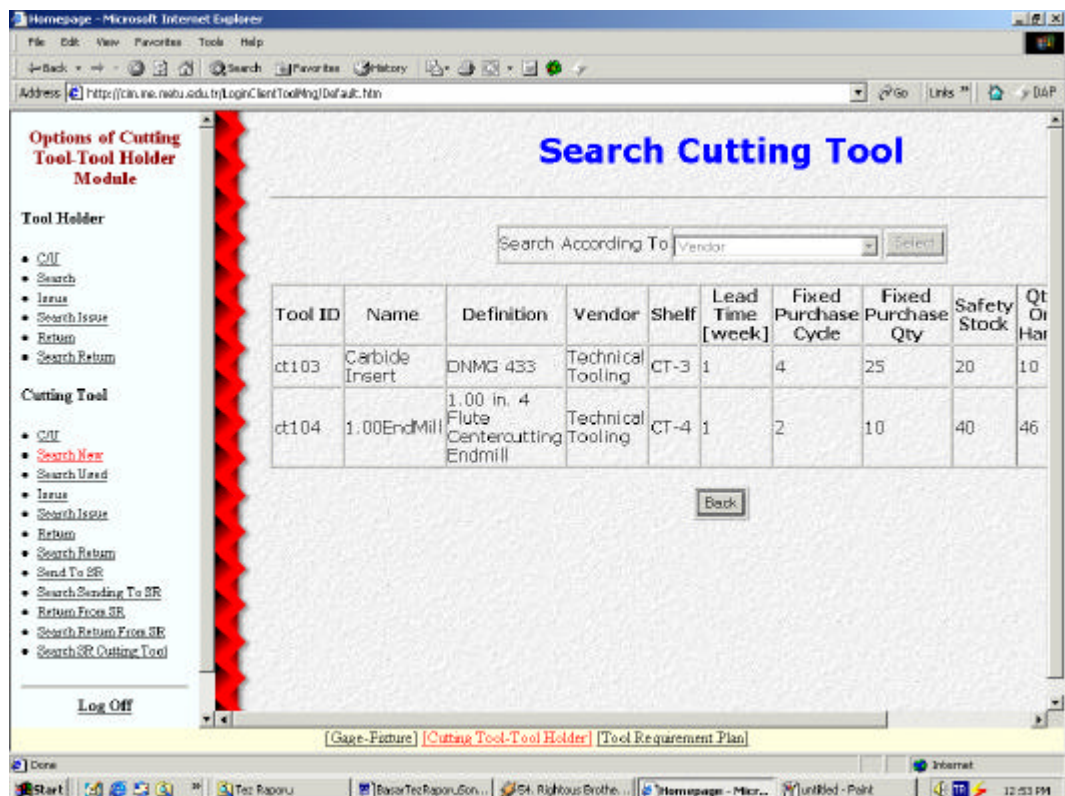


Figure B.19 Search Cutting Tool Page

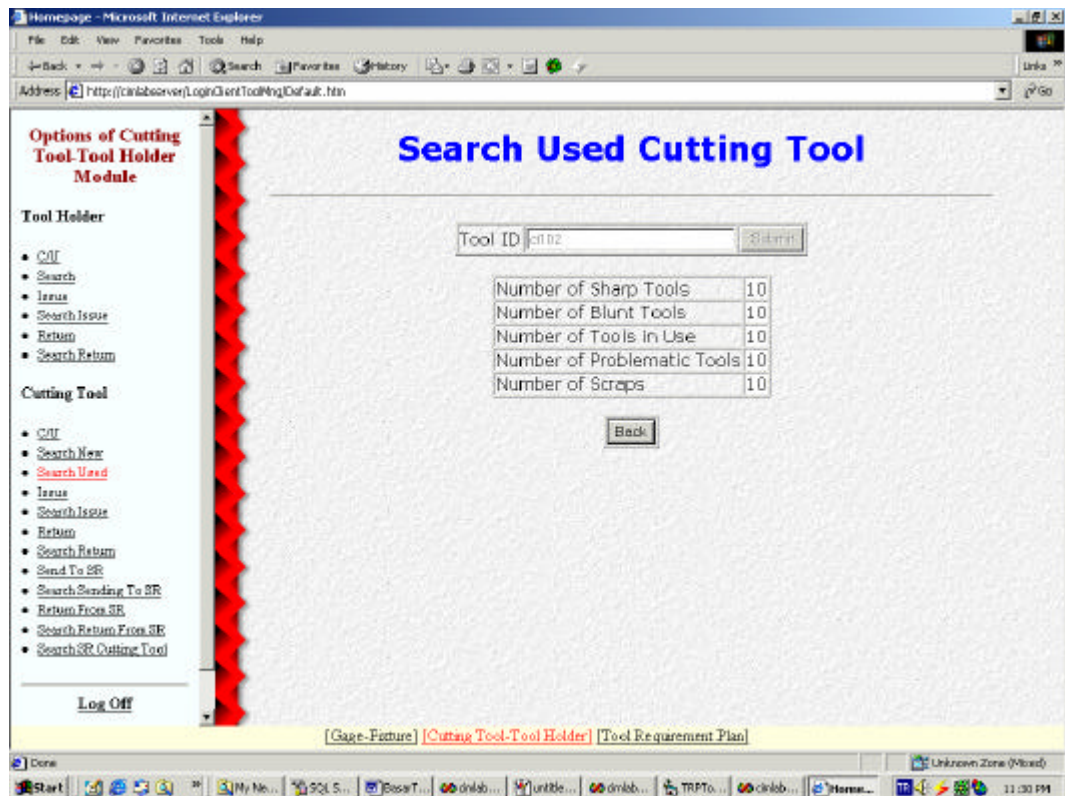


Figure B.20 Search Used Cutting Tool Page

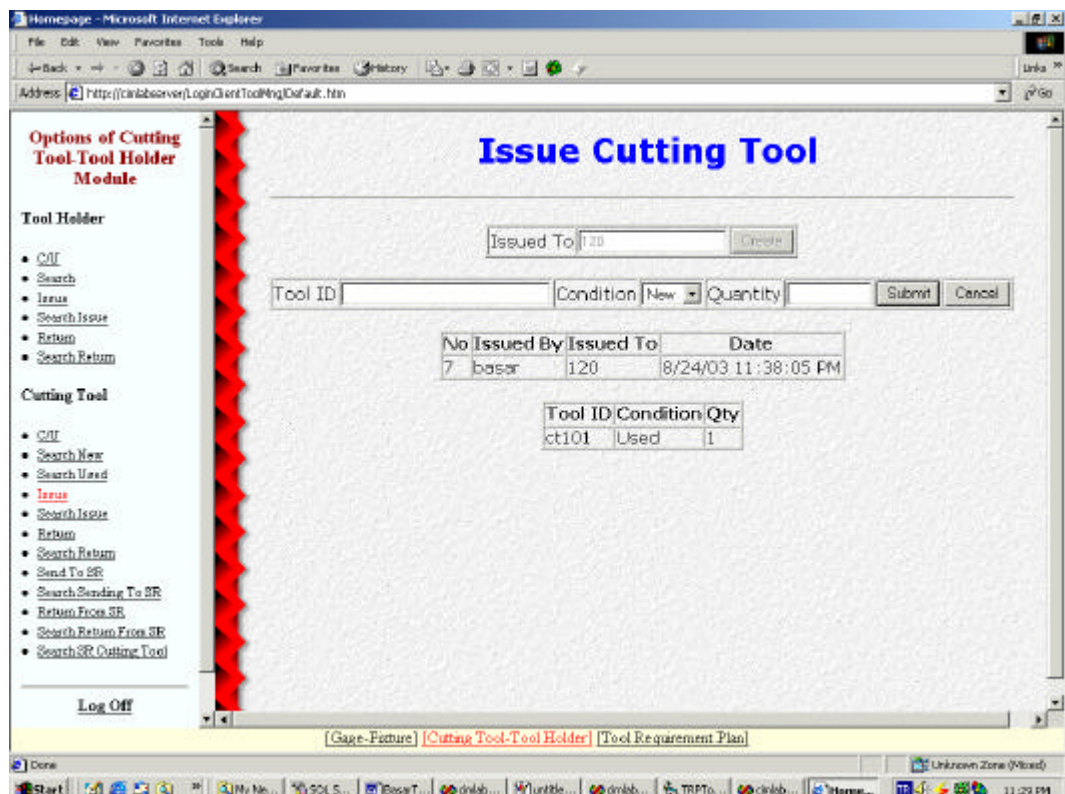


Figure B.21 Issue Cutting Tool Page

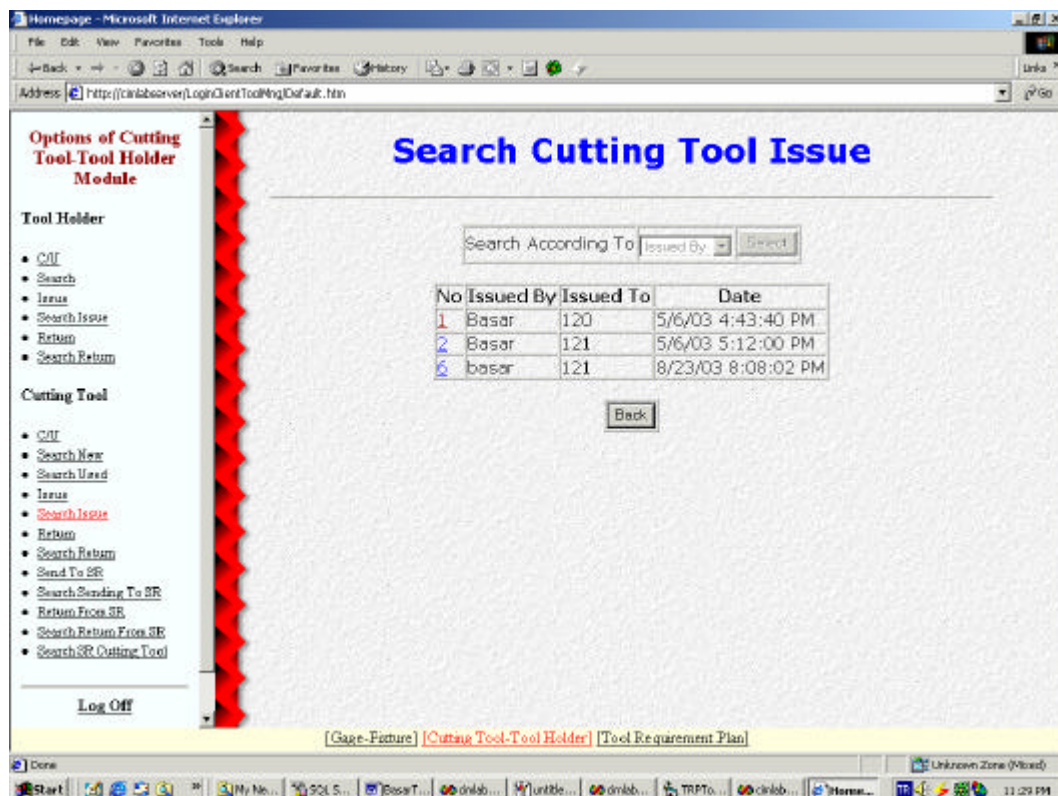


Figure B.22 Search Cutting Tool Issue Page

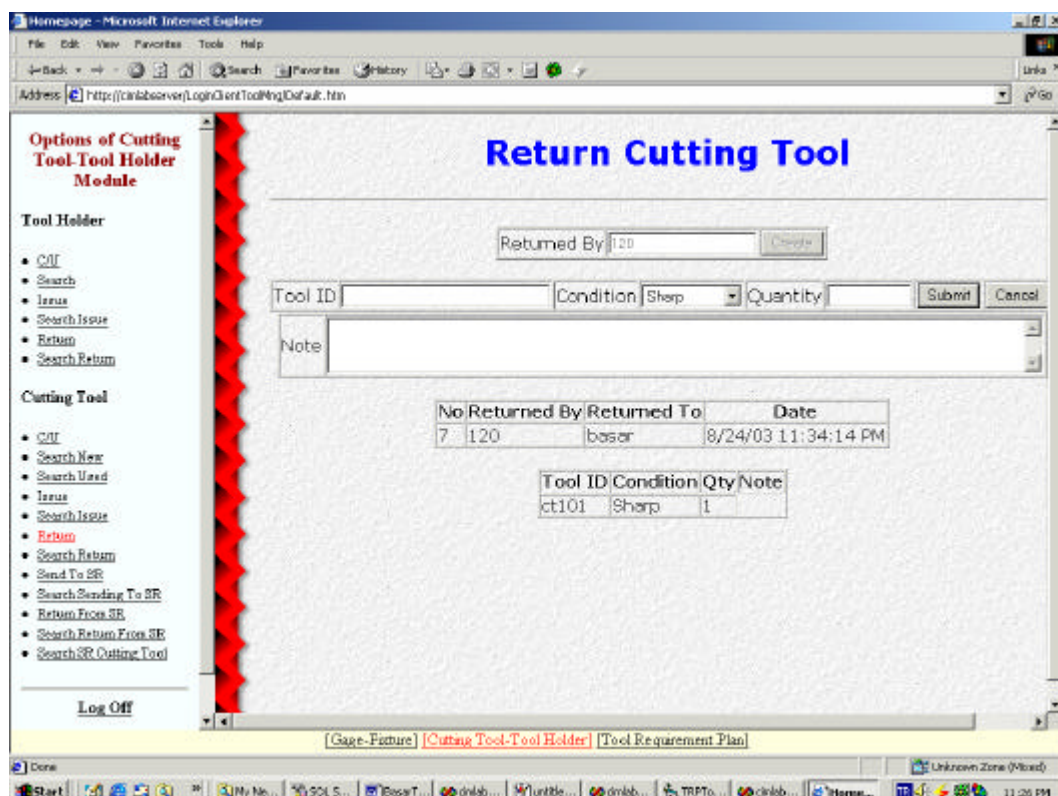


Figure B.23 Return Cutting Tool Page

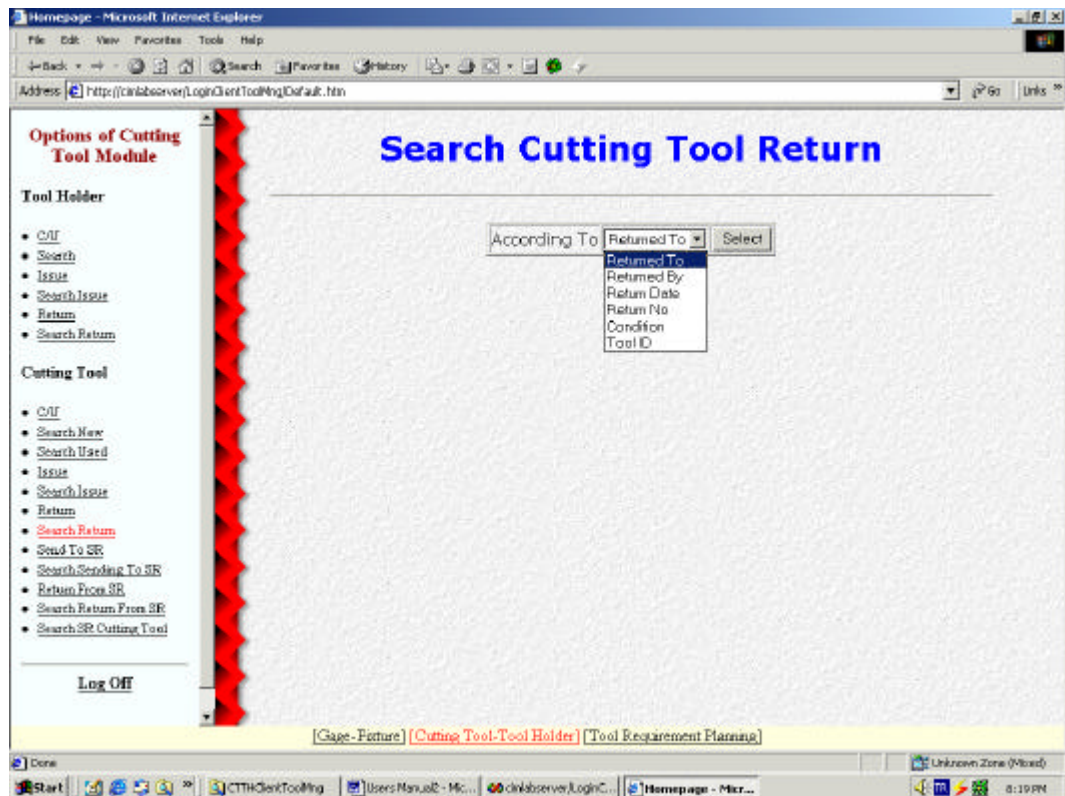


Figure B.24 Search Cutting Tool Return Page

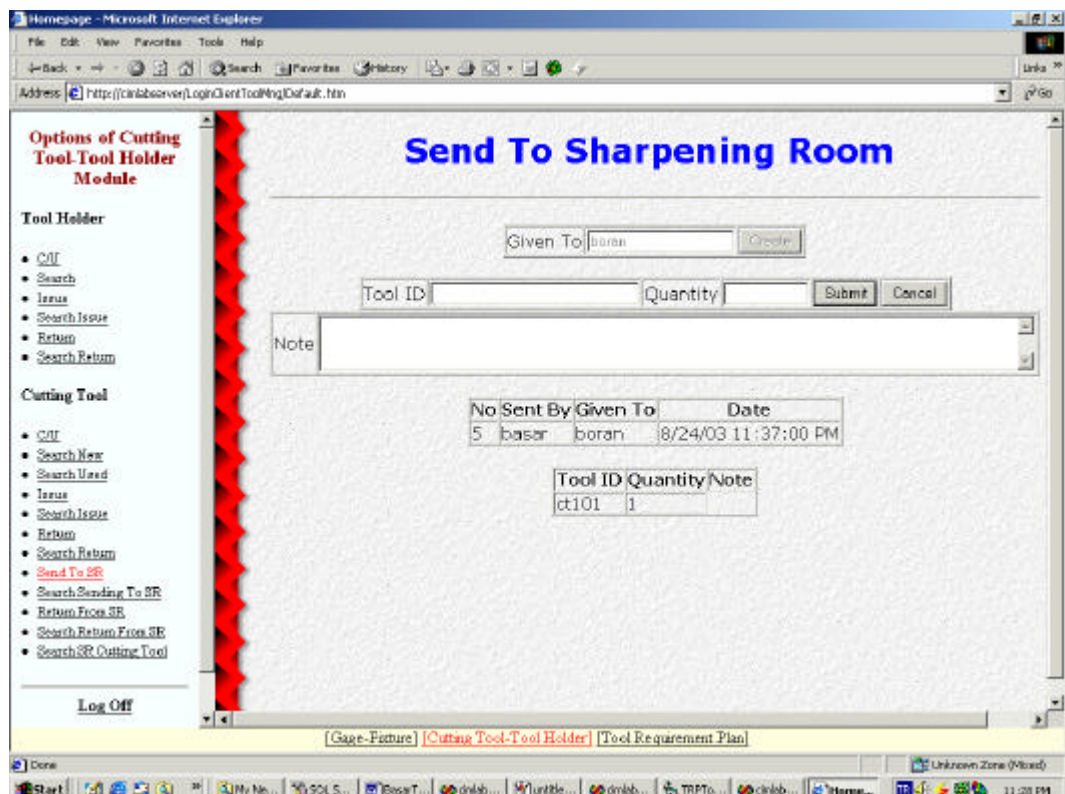


Figure B.25 Send To Sharpening Room Page

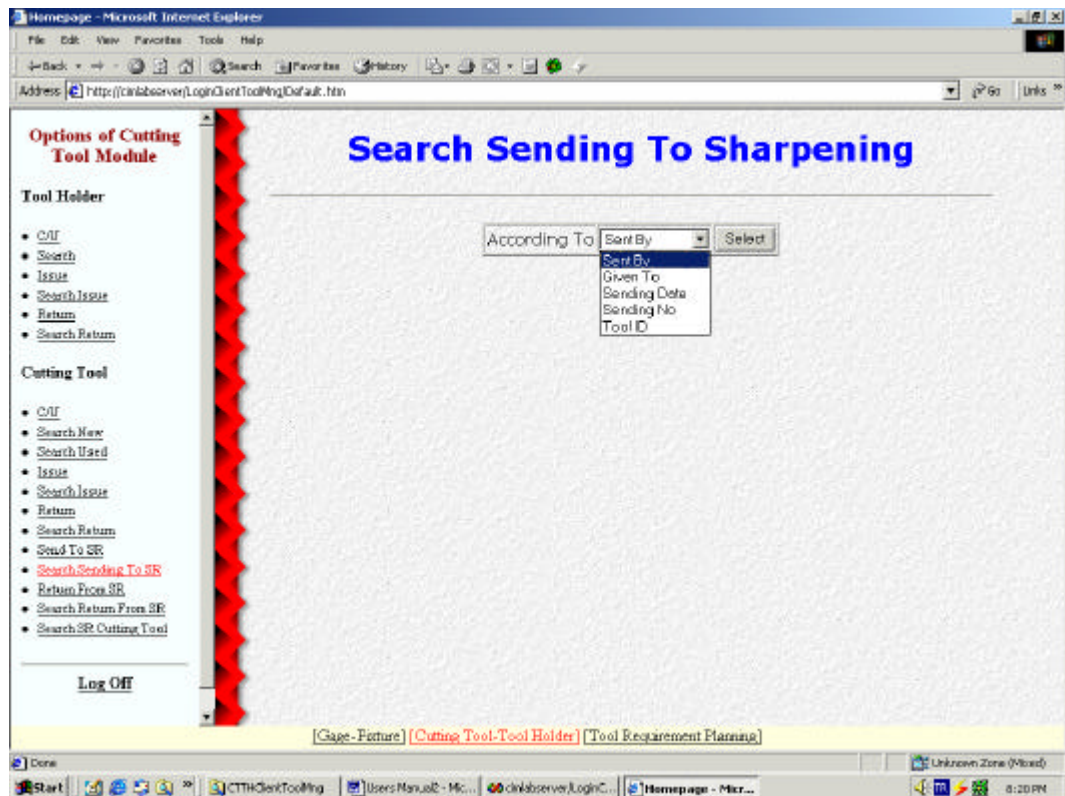


Figure B.26 Search Sending To Sharpening Page

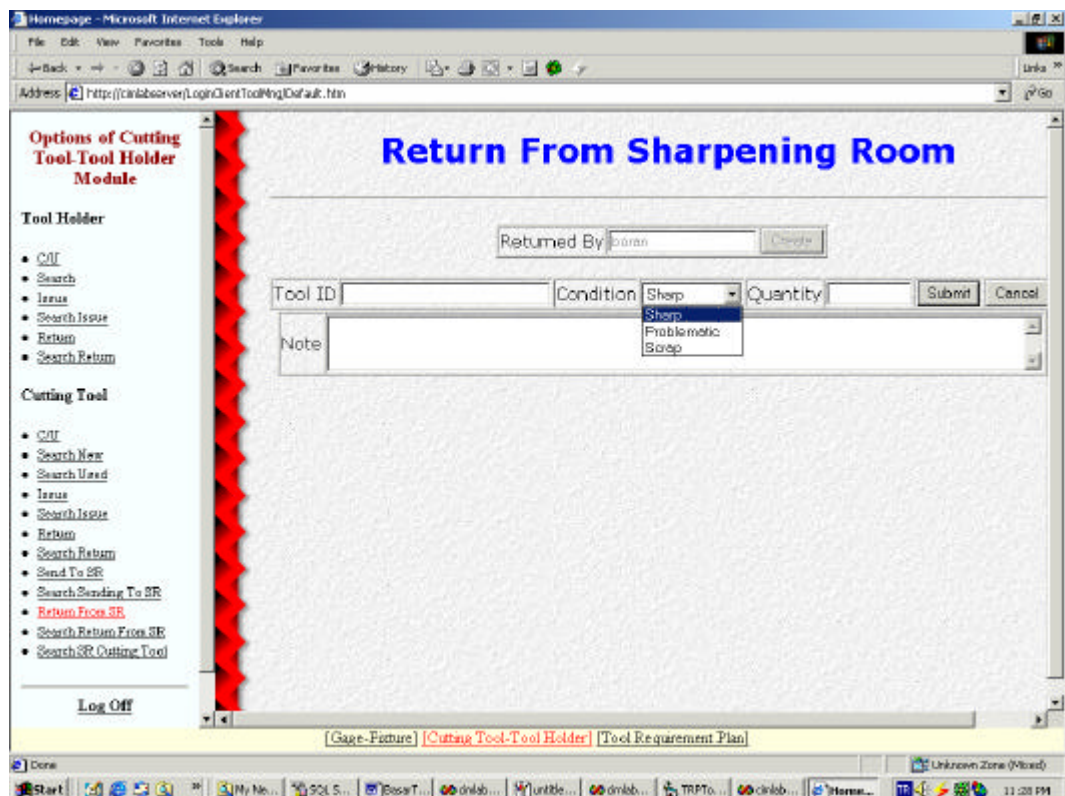


Figure B.27 Return From Sharpening Room Page

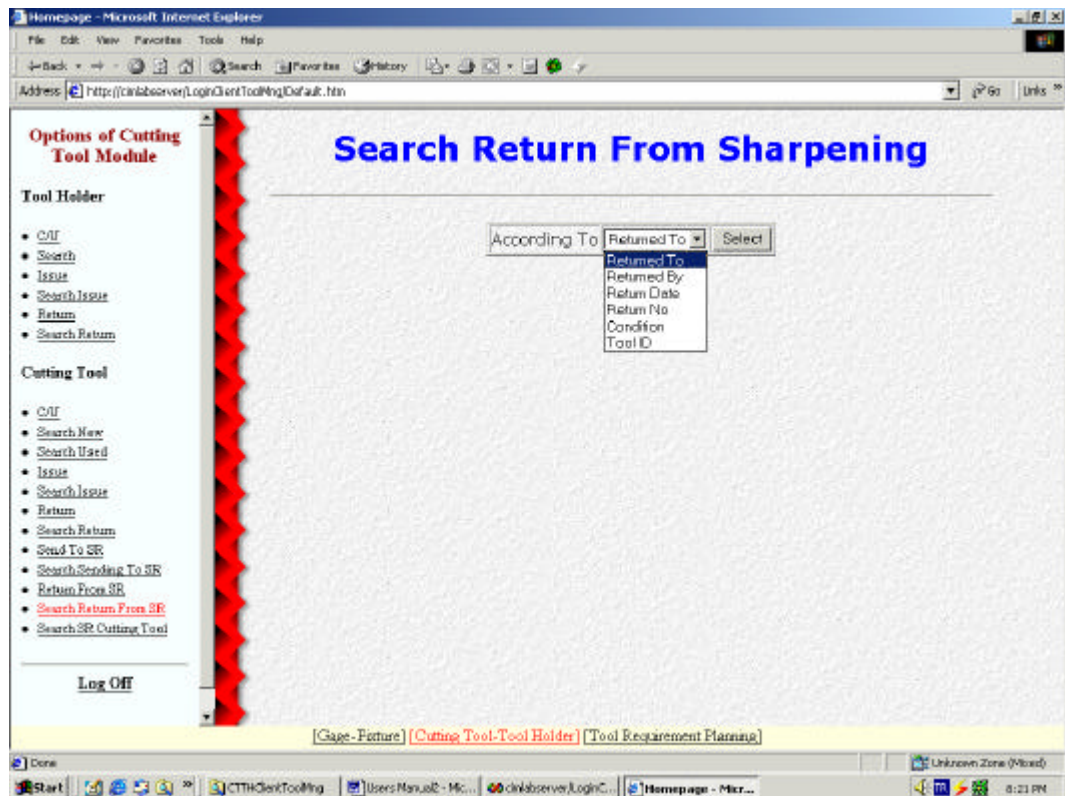


Figure B.28 Search Return From Sharpening Page

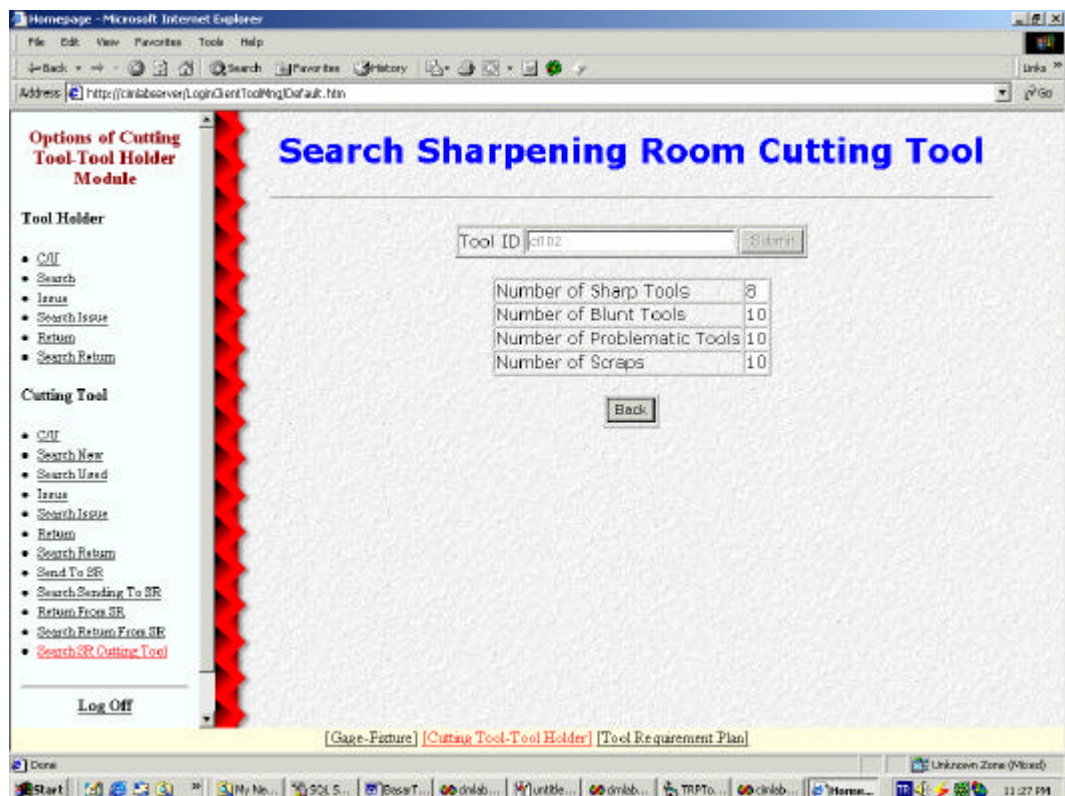


Figure B.29 Search Sharpening Room Cutting Tool Page

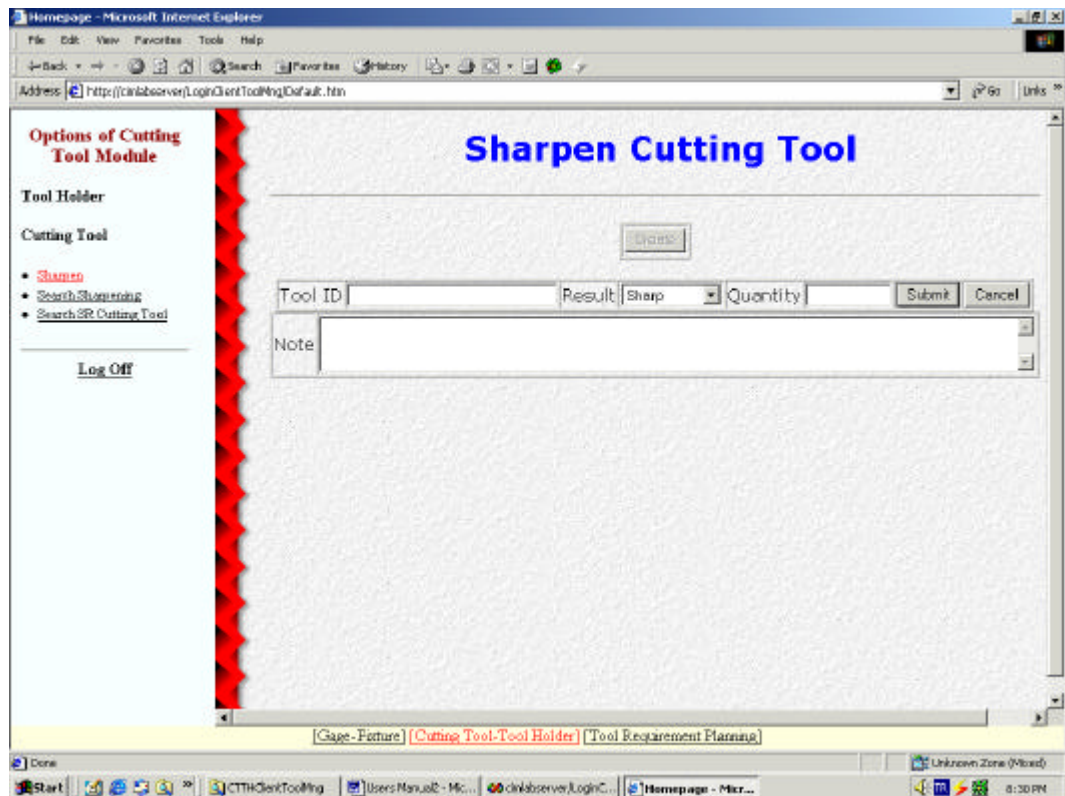


Figure B.30 Sharpen Cutting Tool Page

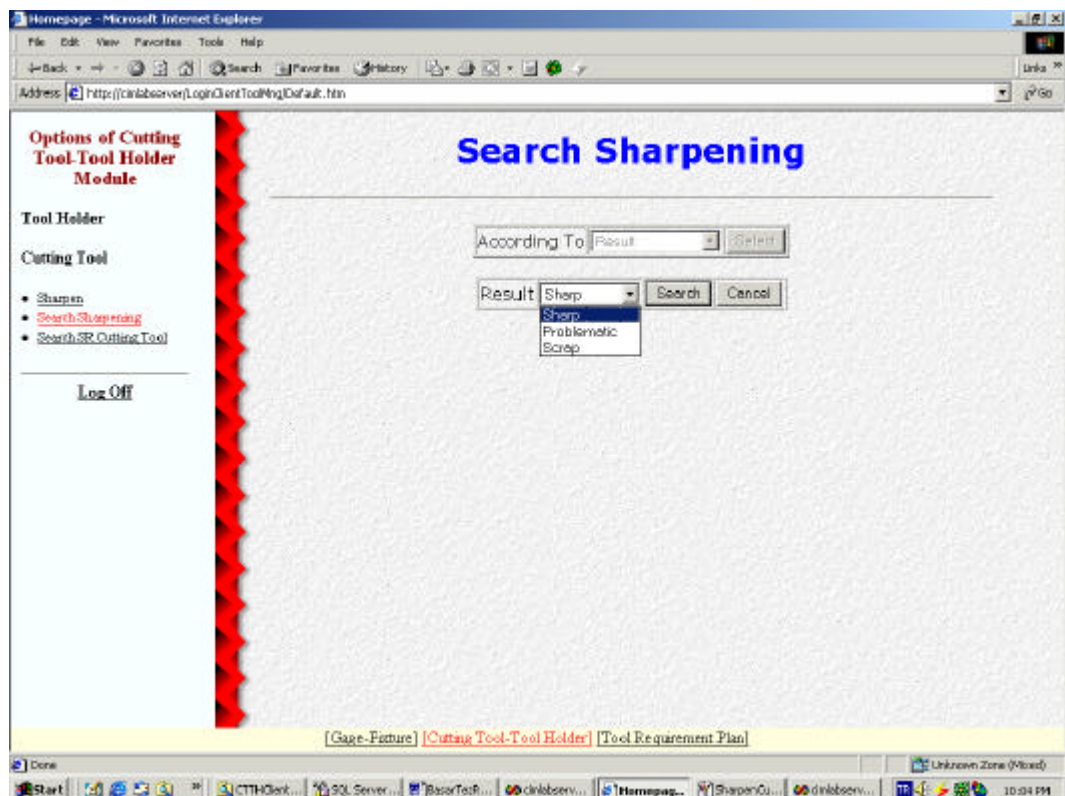


Figure B.31 Search Sharpening Page

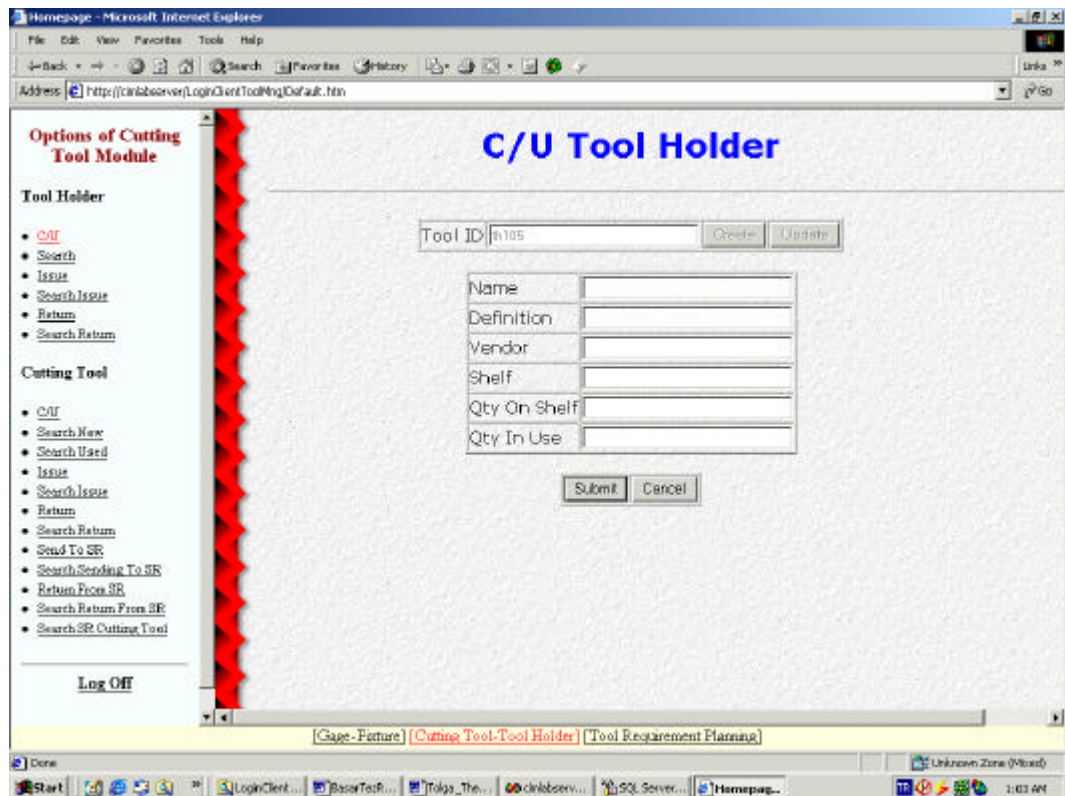


Figure B.32 C/U Tool Holder Page

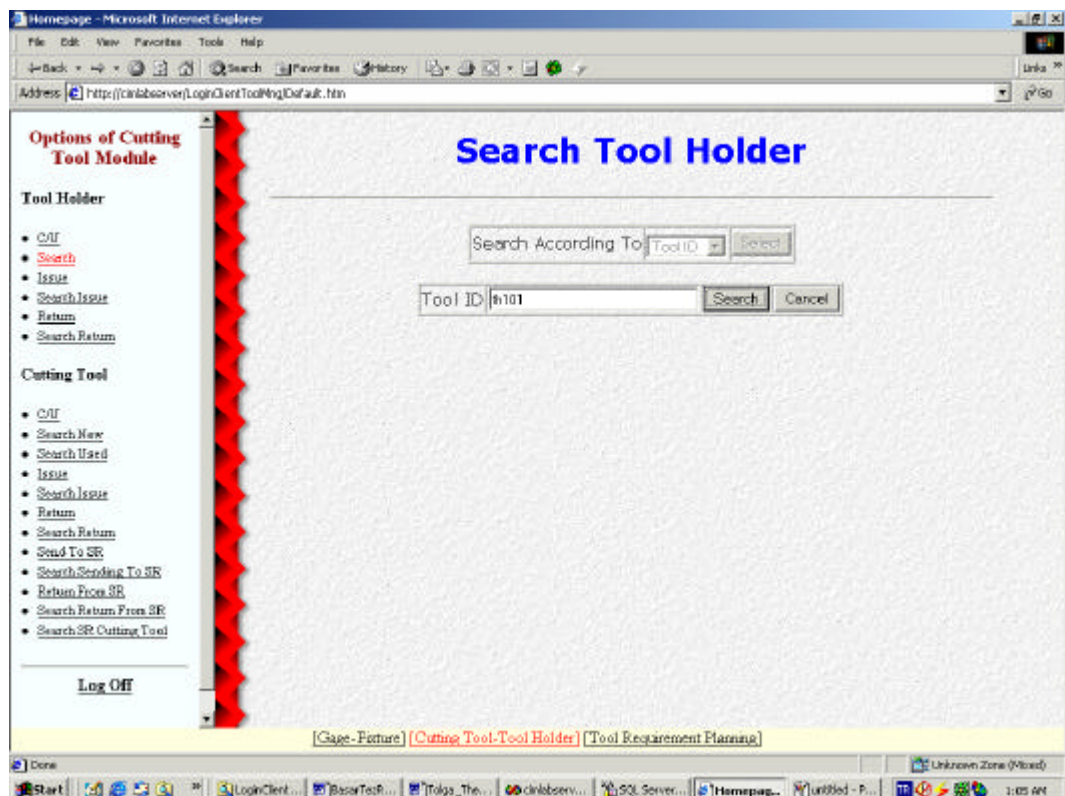


Figure B.33 Search Tool Holder Page

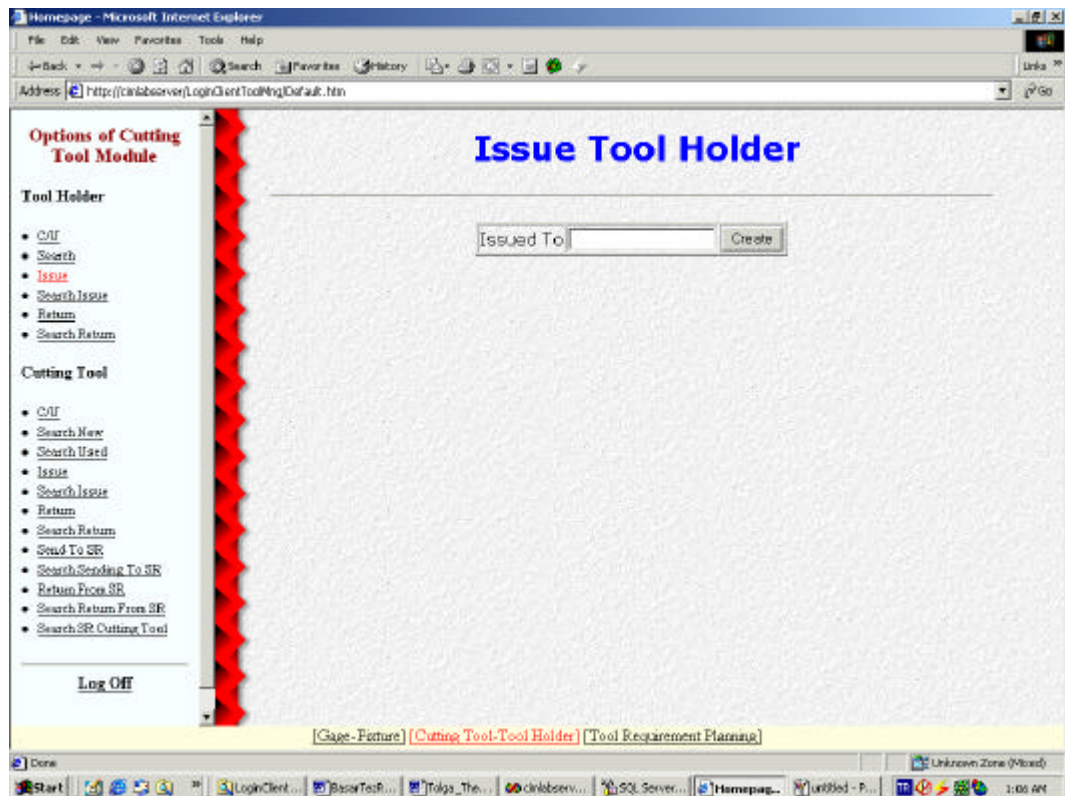


Figure B.34 Issue Tool Holder Page

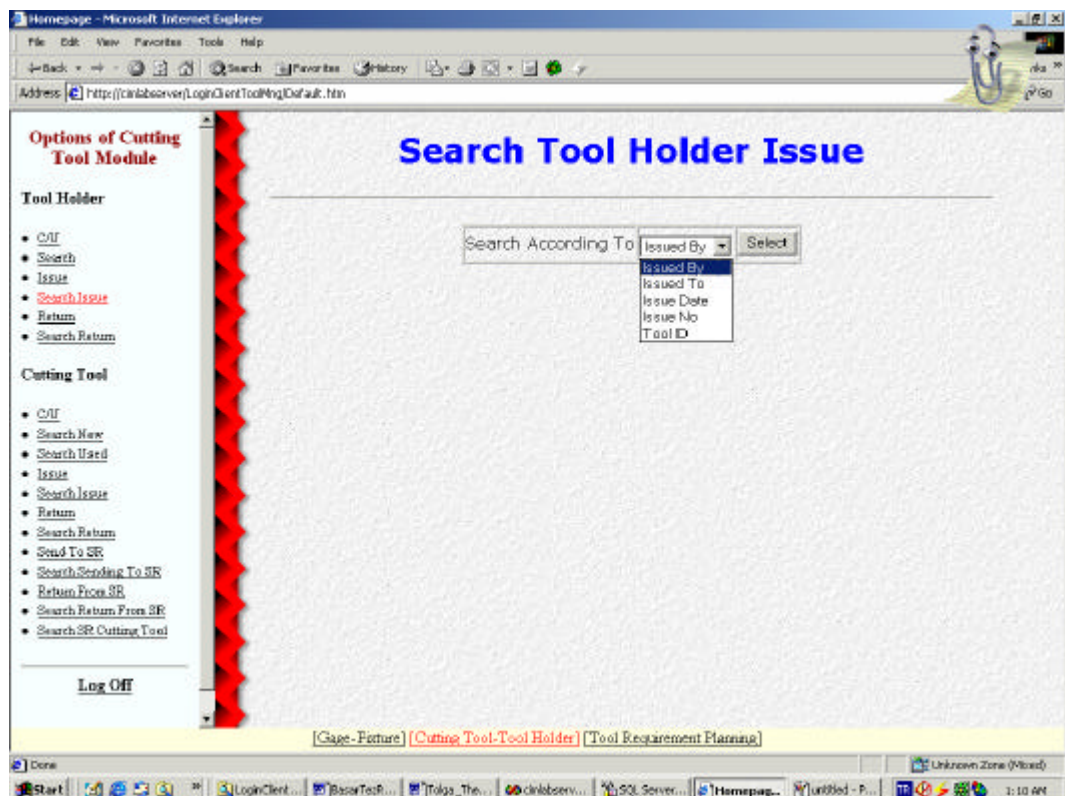


Figure B.35 Search Tool Holder Issue Page

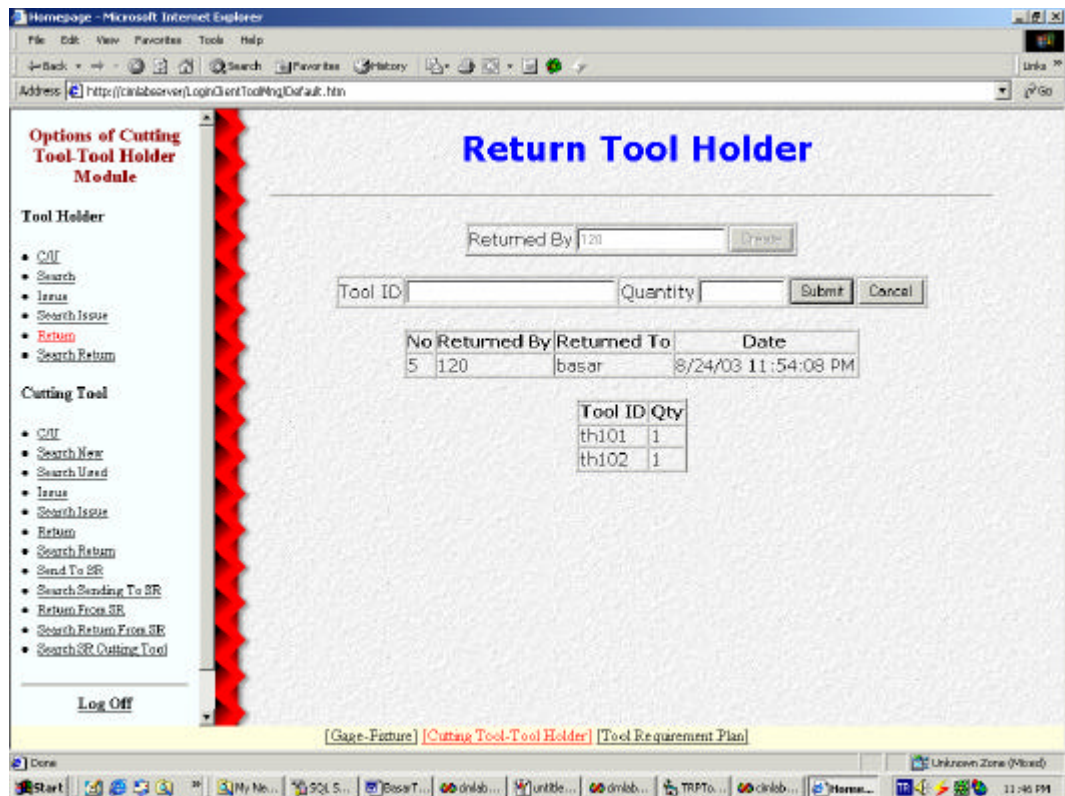


Figure B.36 Return Tool Holder Page

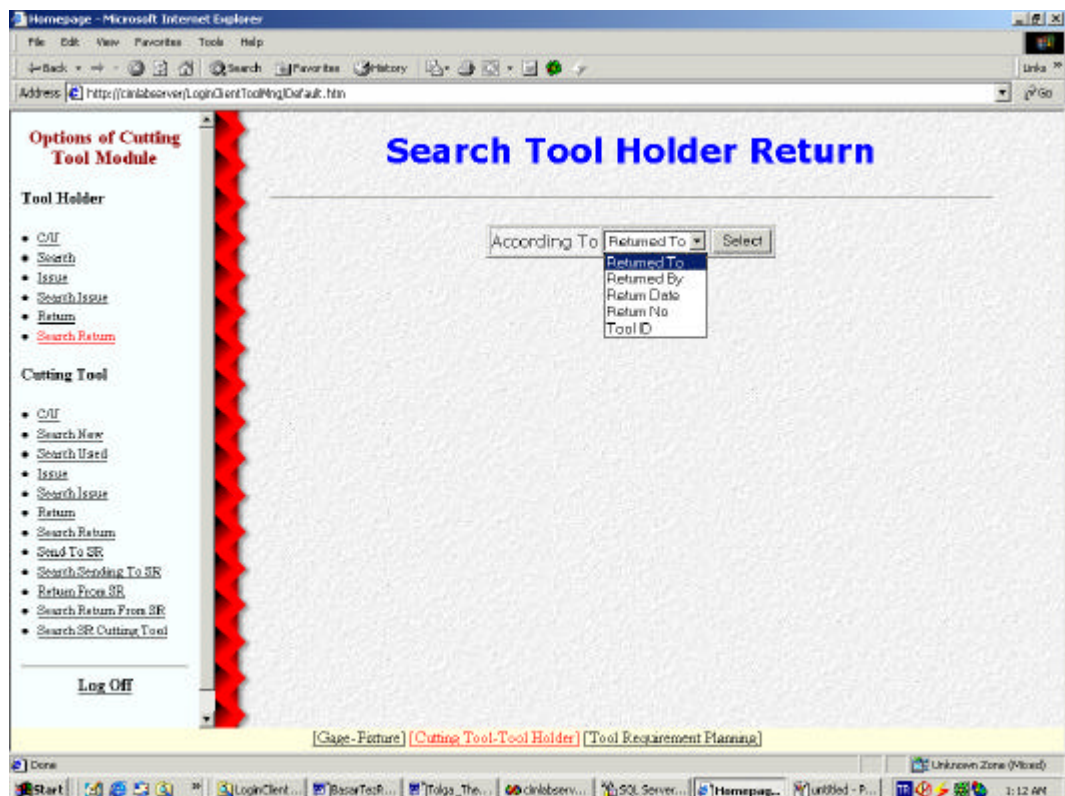


Figure B.37 Search Tool Holder Return

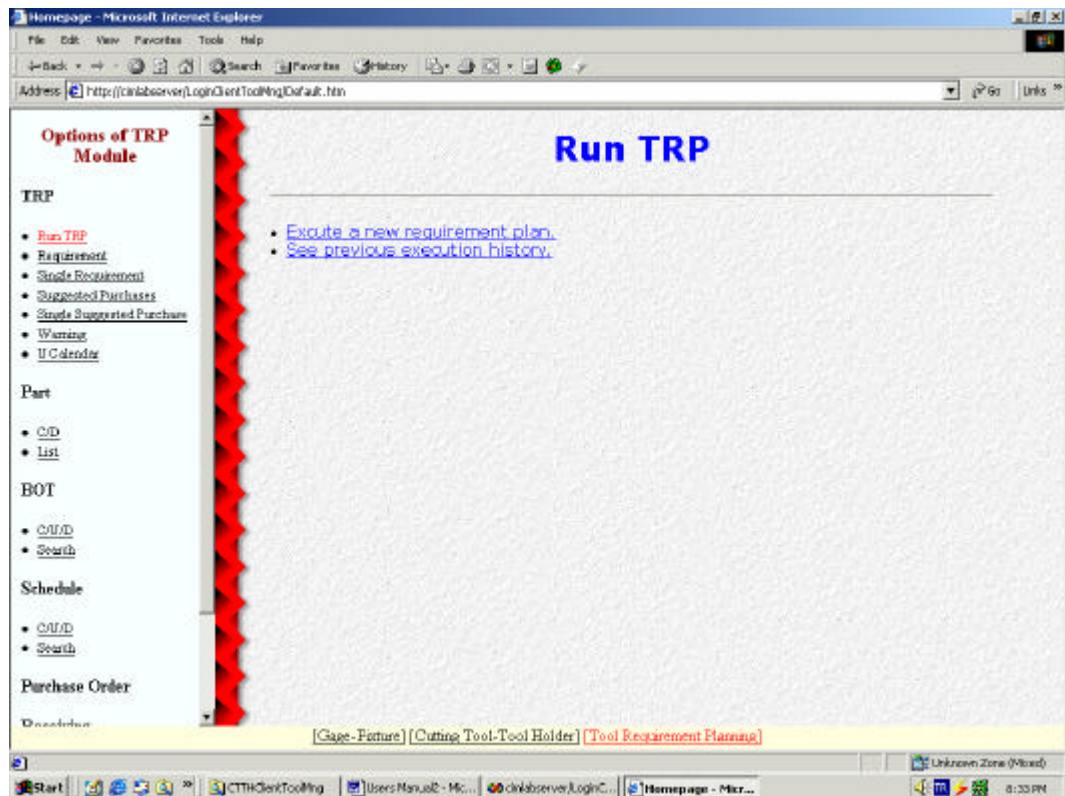


Figure B.38 Run TRP Page

Date	Requirement	Scheduled Receipt	Planned Receipt	Total Receipt	Available
8/28/03	0	0	50	50	50
8/29/03	40	0	0	0	10
8/30/03	0	0	50	50	60
9/1/03	40	0	0	0	20
9/2/03	50	0	50	50	20
9/3/03	45	0	25	25	0
9/4/03	50	50	50	100	50
9/5/03	35	0	0	0	15
9/6/03	0	0	25	25	40
9/8/03	40	0	0	0	0
9/9/03	40	0	50	50	10
9/10/03	40	0	50	50	20
9/11/03	40	0	75	75	55
9/12/03	40	0	0	0	15

Date	Requirement	Scheduled Receipt	Planned Receipt	Total Receipt	Available
8/28/03	0	0	100	100	05

Figure B.39 Requirement Plan Page

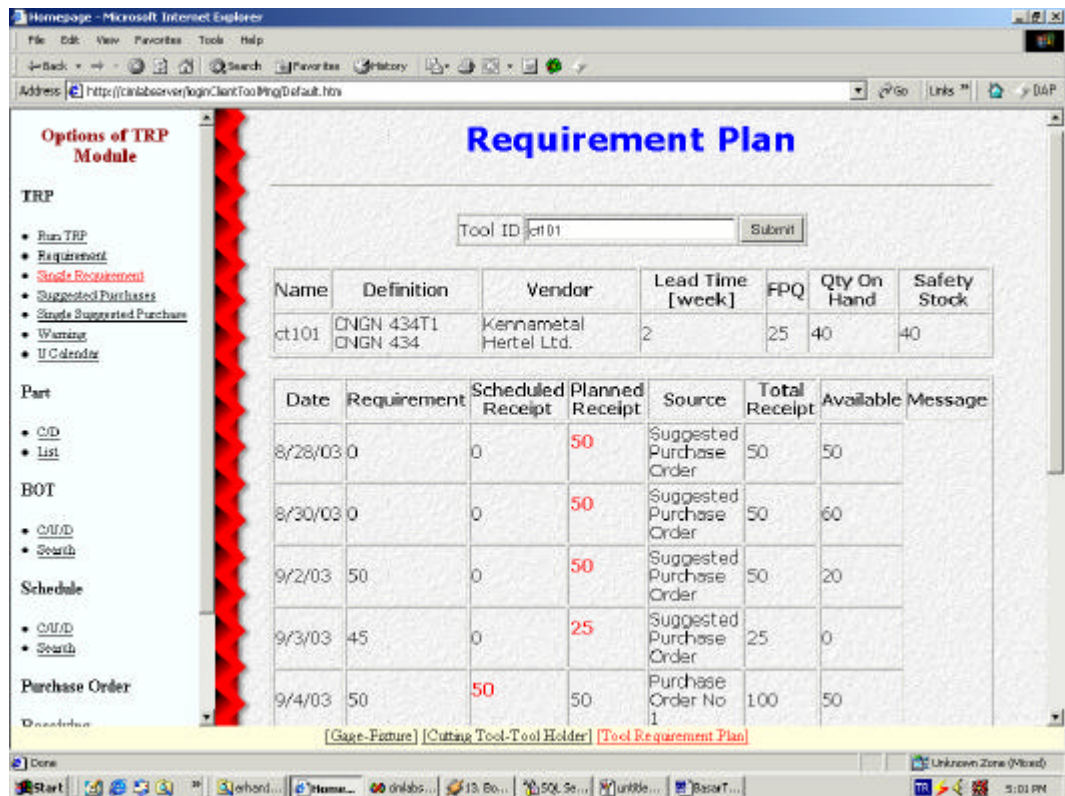


Figure B.40 Single Requirement Plan Page

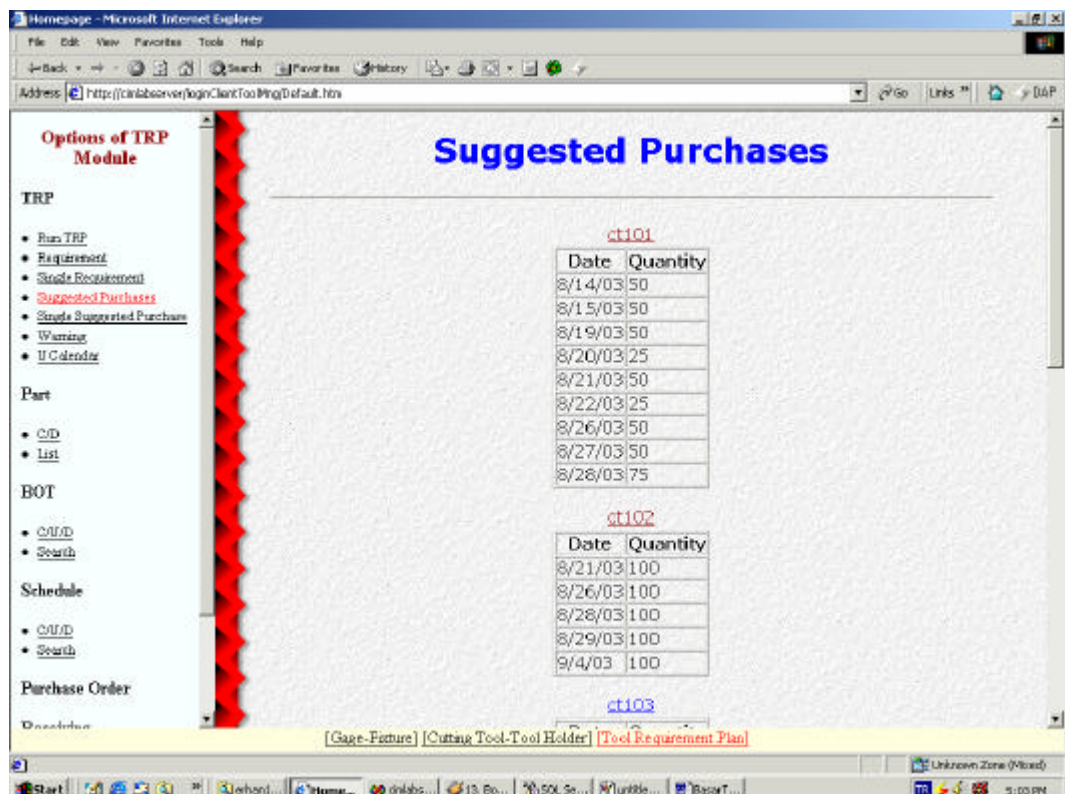


Figure B.41 Suggested Purchases Page

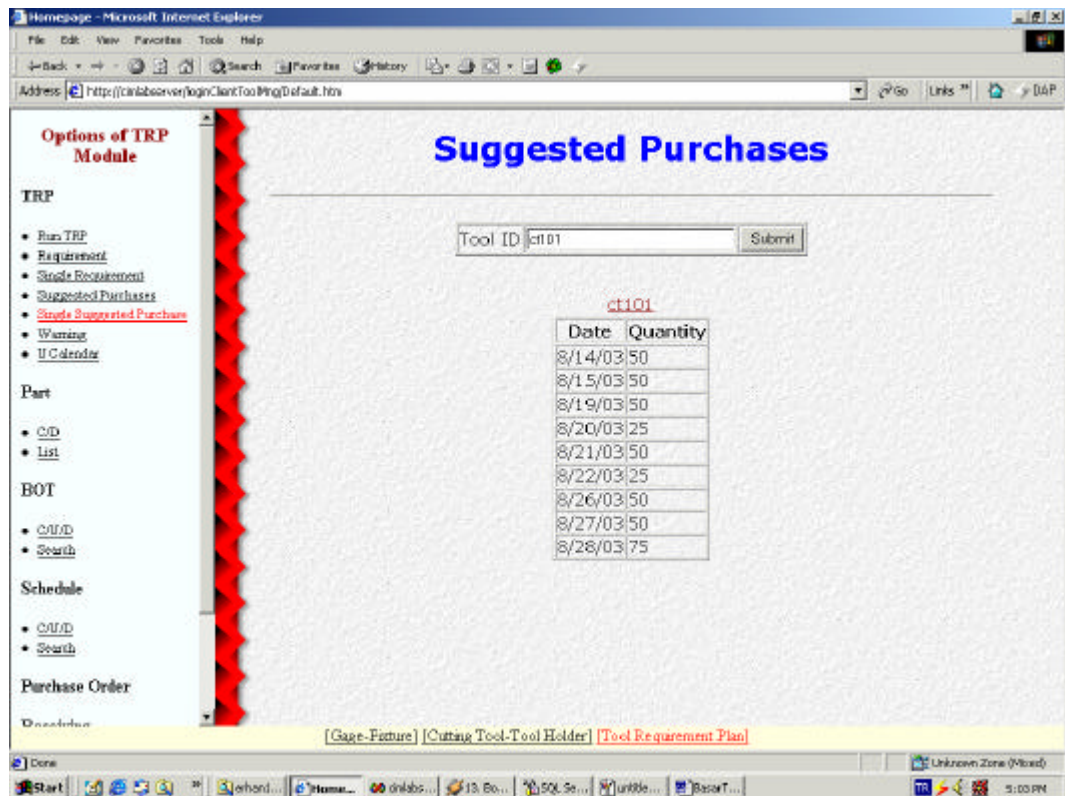


Figure B.42 Single Suggested Purchase Page

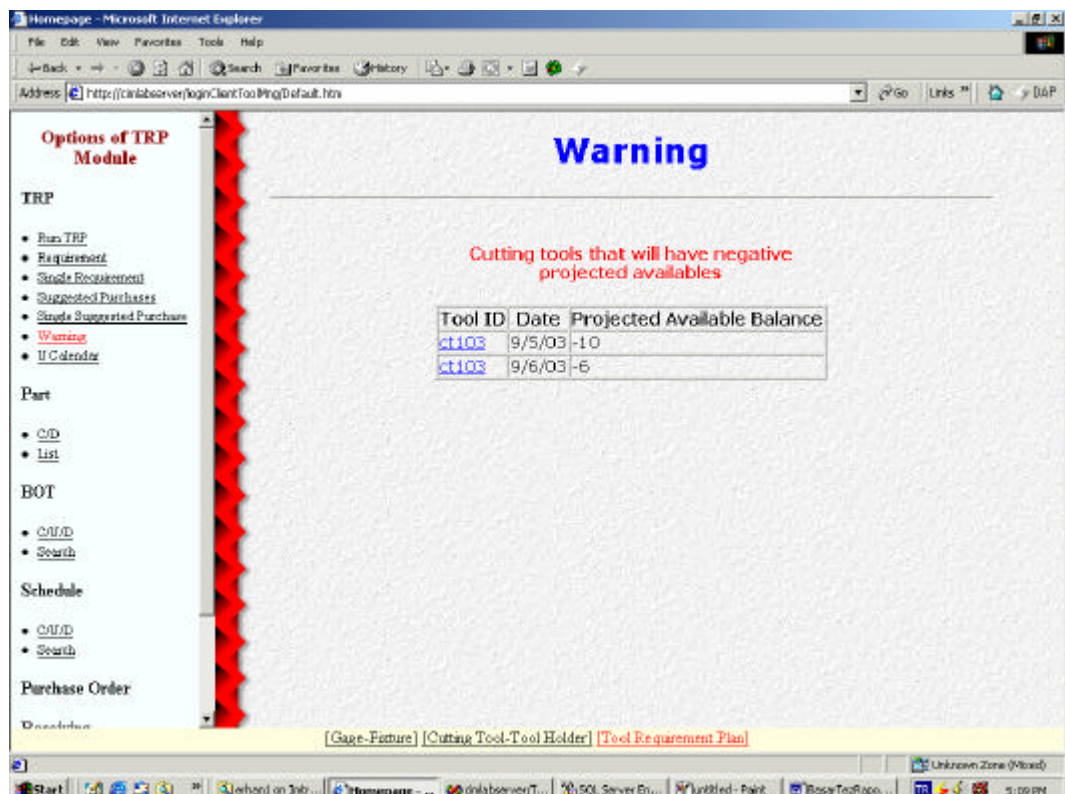


Figure B.43 Warning Page

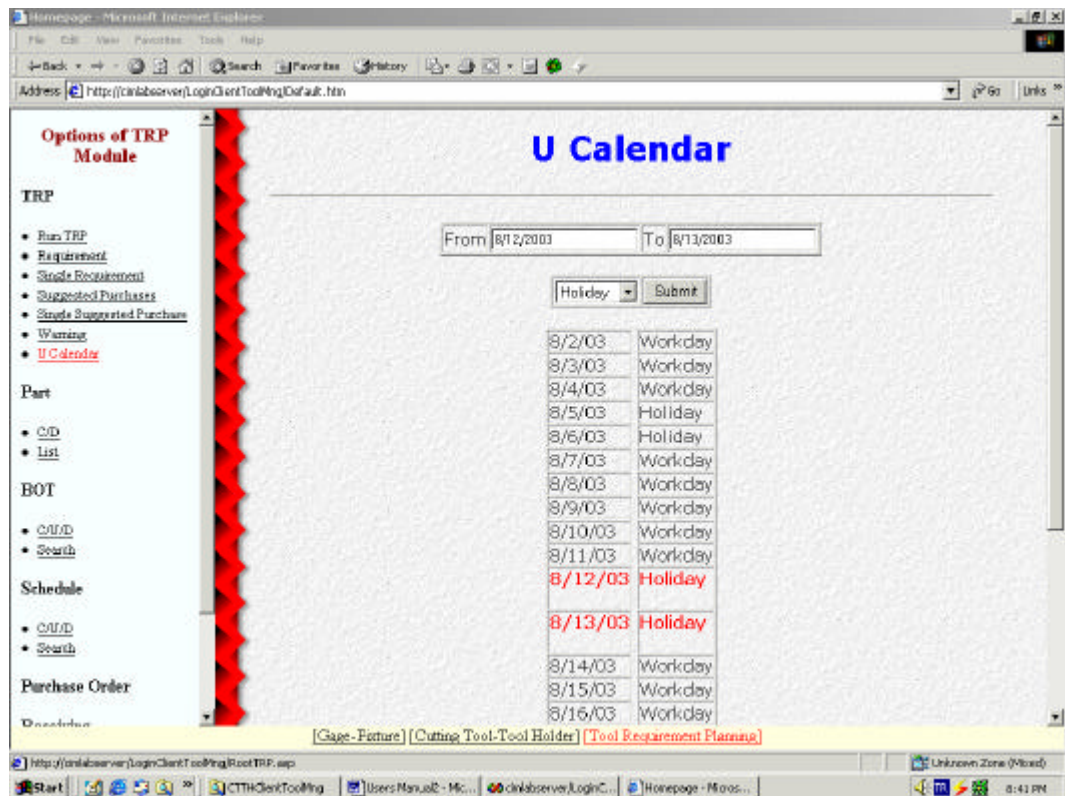


Figure B.44 Calendar Page

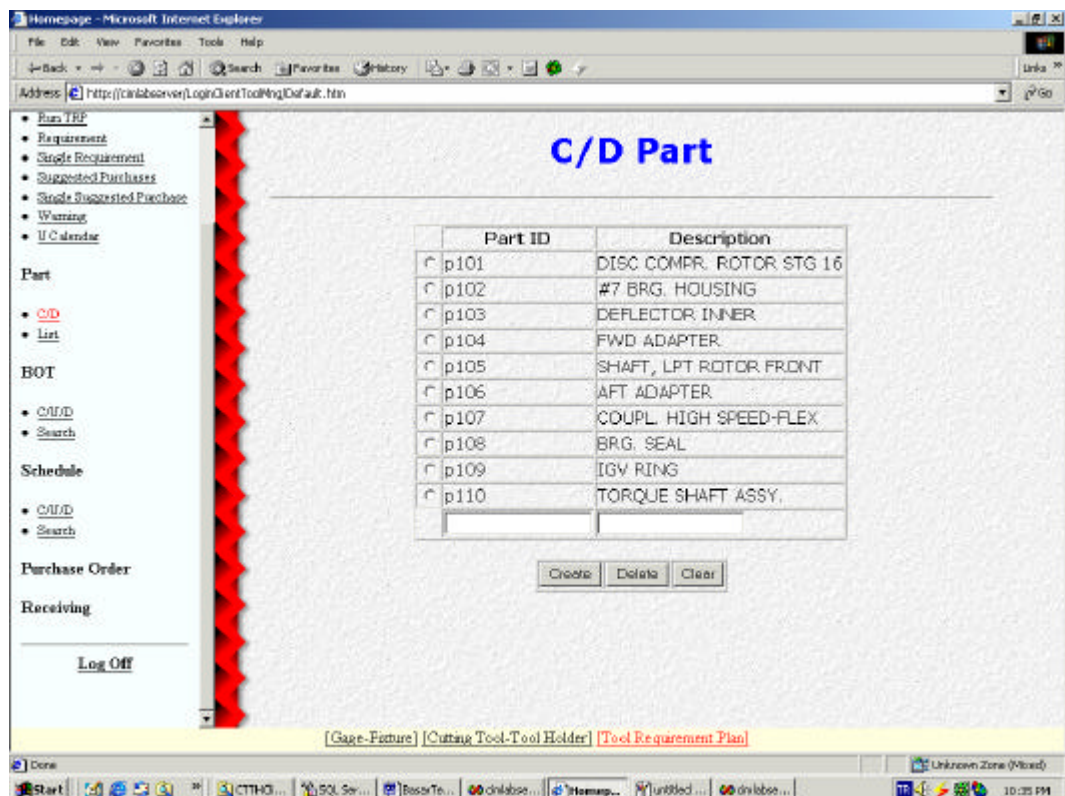


Figure B.45 C/D Part Page

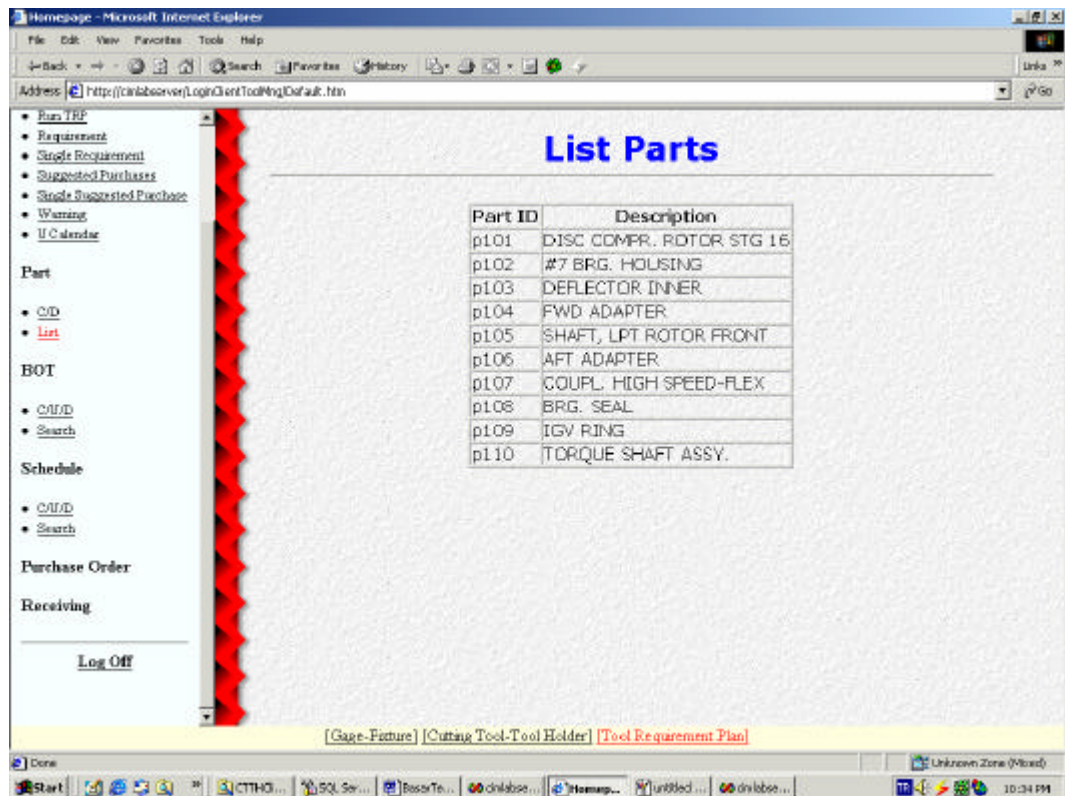


Figure B.46 List Parts Page

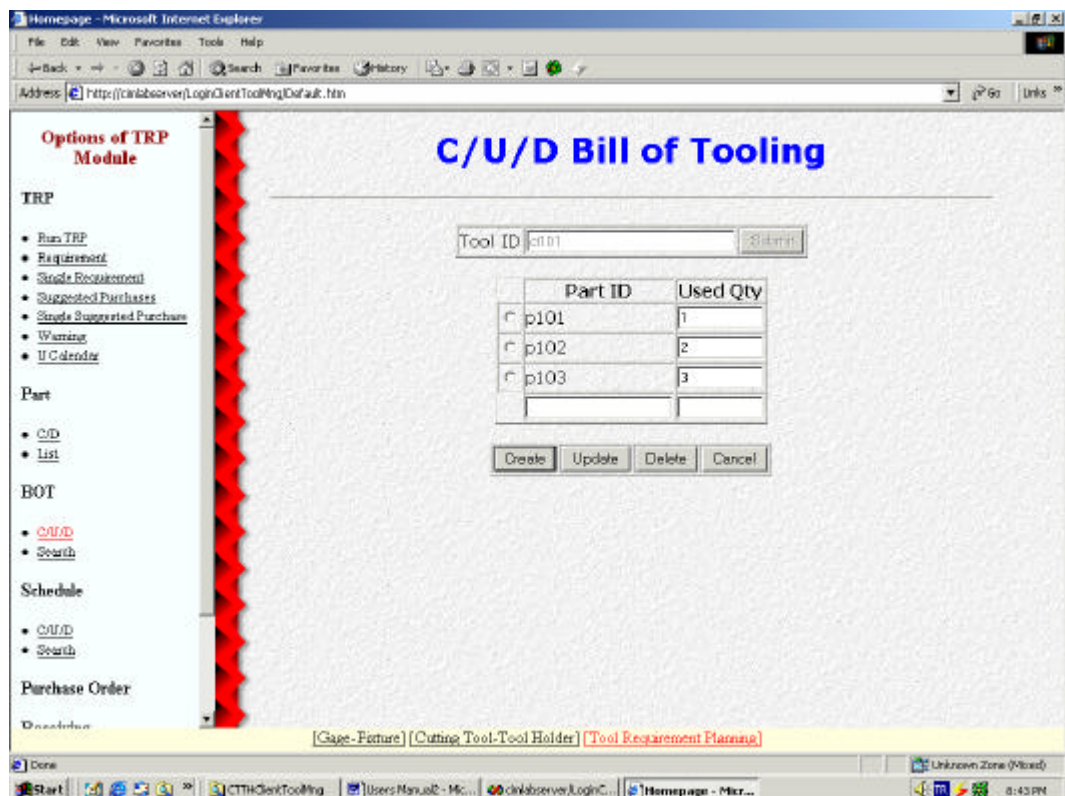


Figure B.47 C/U/D BOT Page

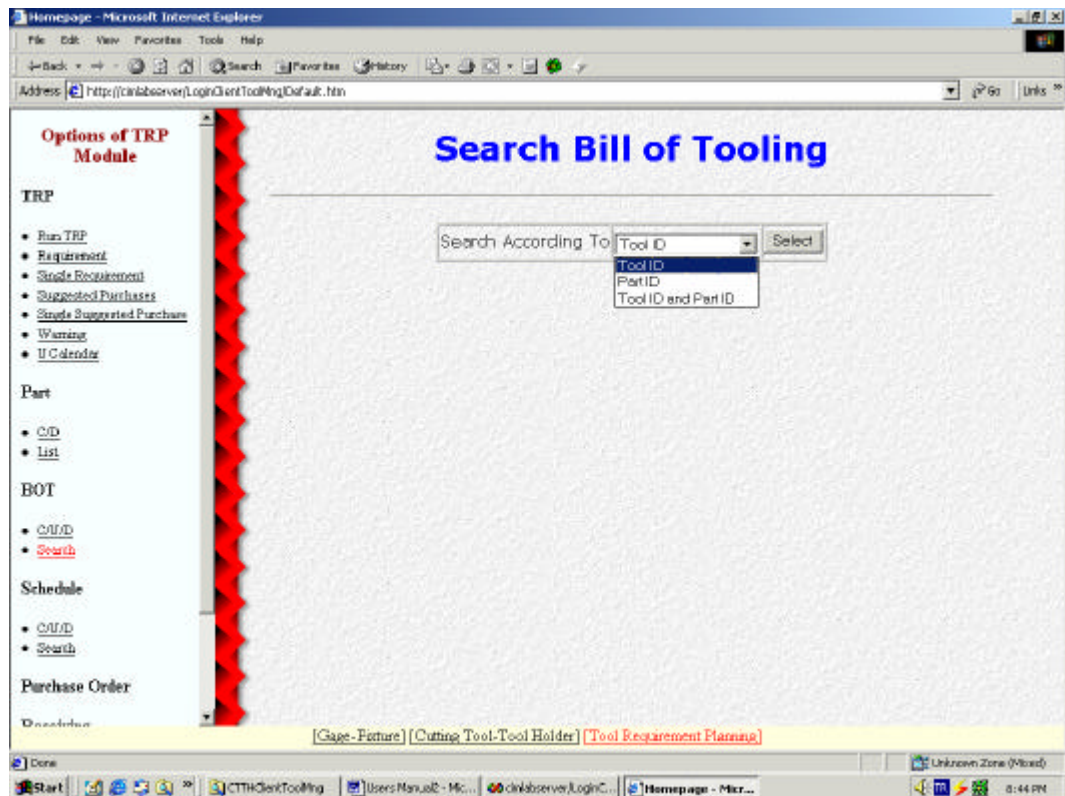


Figure B.48 Search Bill of Tooling Page

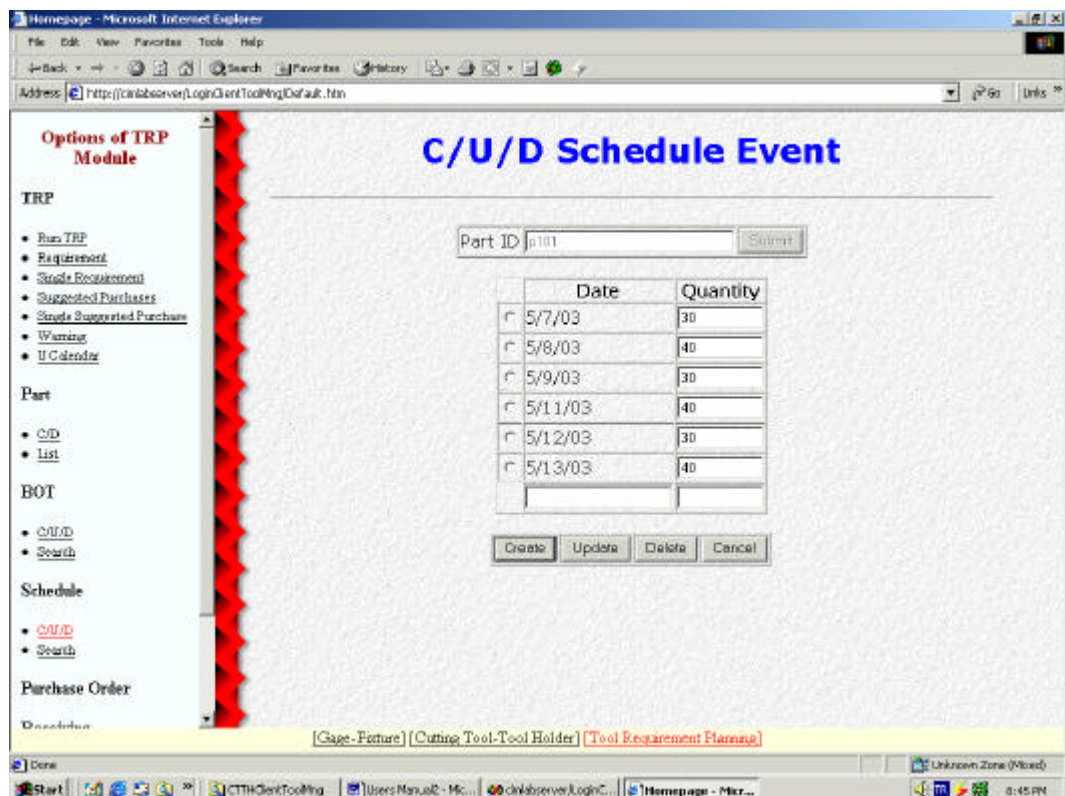


Figure B.49 C/U/D Schedule Event Page

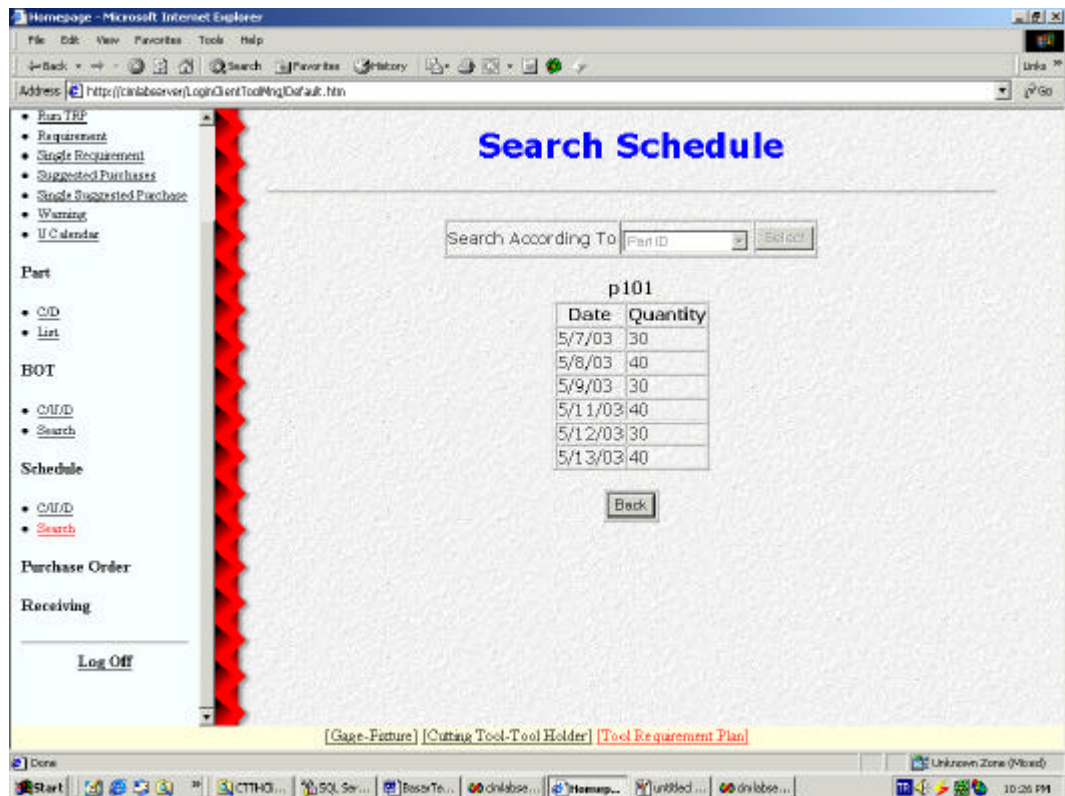


Figure B.50 Search Schedule Page

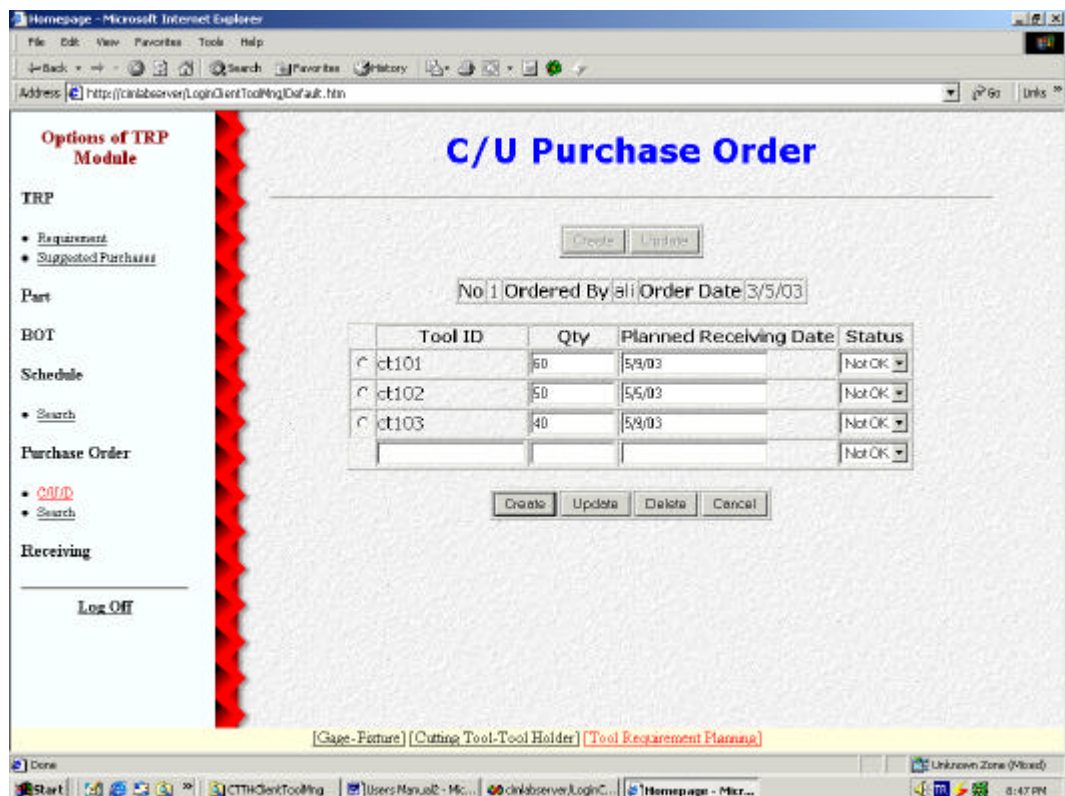


Figure B.51 C/U Purchase Order Page

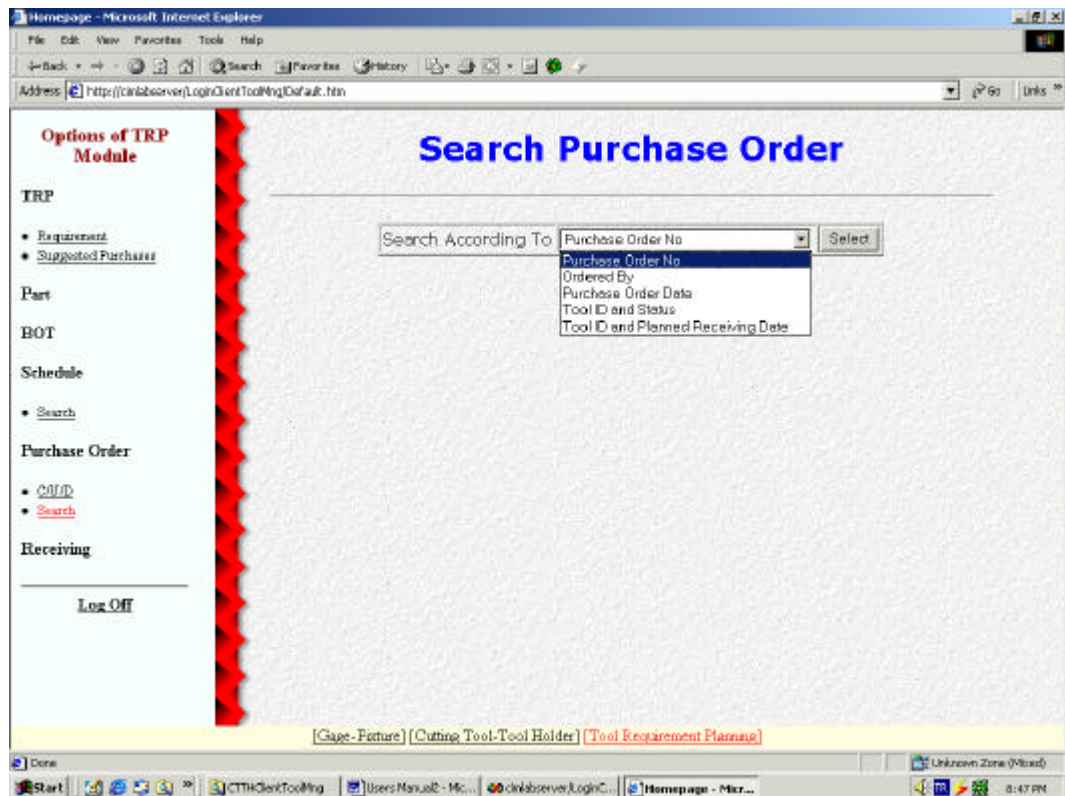


Figure B.52 Search Purchase Order

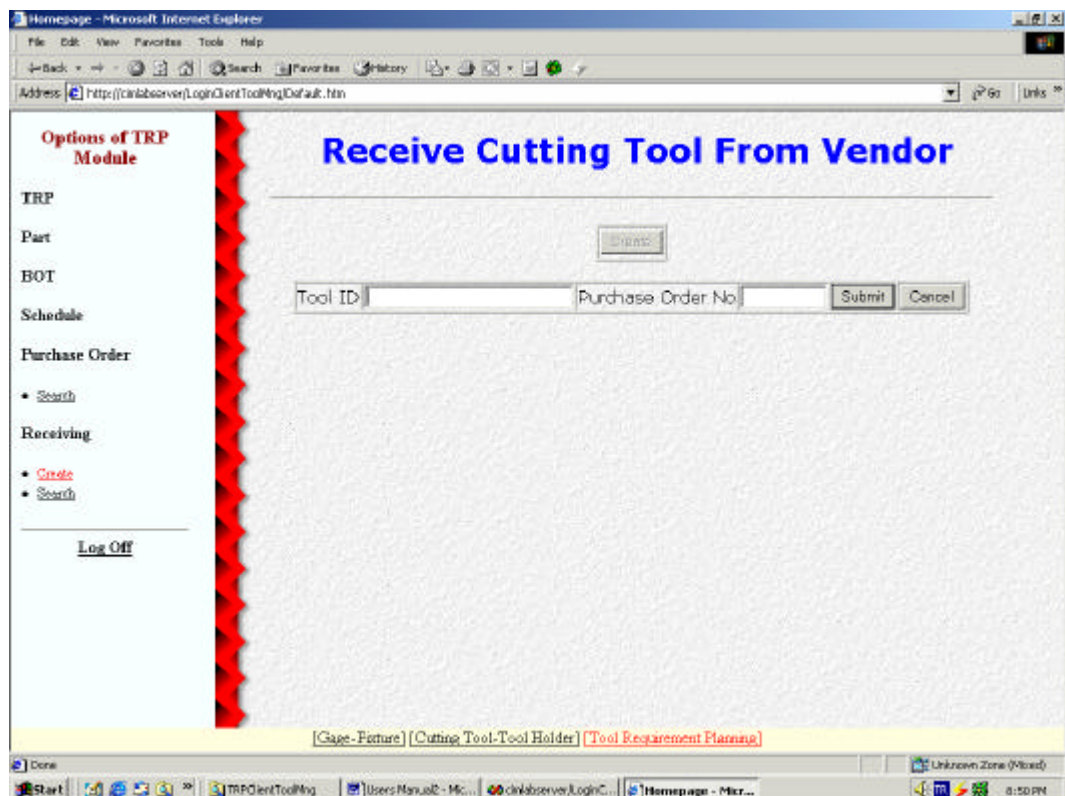


Figure B.53 Receive Cutting Tool From Vendor

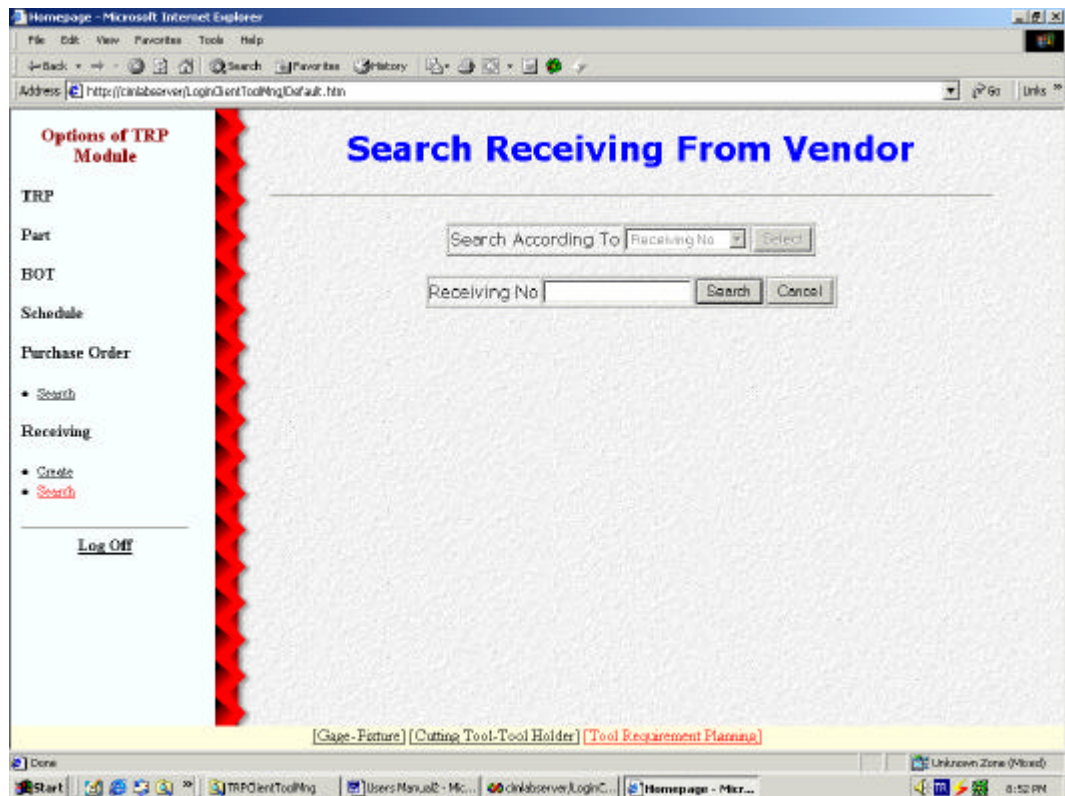


Figure B.54 Search Receiving From Vendor

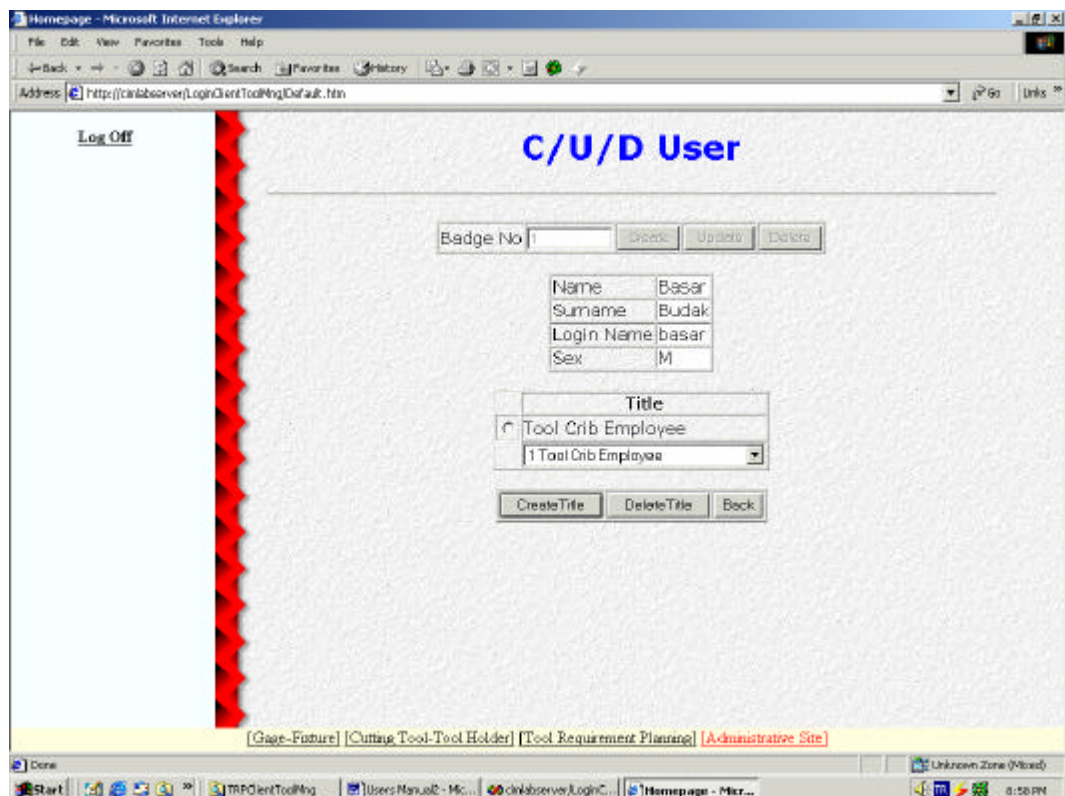


Figure B.55 C/U/D User Page

Tool requirement plan module includes the pages from Figure B.37 to Figure B.54. In “Run TRP” page, given in Figure B.38, user can execute a new requirement plan or see the previous execution history clicking the appropriate links in the page. “Requirement Plan” page retrieves all the requirement data as given in Figure B.39. In “Single Requirement Plan” page, given in Figure B.40, user can get the requirement plan of a single cutting tool whose tool ID is entered into the “Tool ID” field after clicking the “Submit” button. “Suggested Purchases” page, given in Figure B. 41, lists all the suggested purchases of cutting tools. In “Single Suggested Purchase” page, given in Figure B.42, user can get the suggested purchase of a cutting tool whose tool ID is supplied. Cutting tool attributes can be retrieved through the link written as the name of the cutting tool both in the pages above. “Warning” page, given in Figure B.43, lists the tools and the dates when these tools’ projected available balances will be negative. “U Calendar” page, given in Figure B.44, changes the status of the days between the dates entered into “From” and “To” fields after pressing “Submit” button.

“C/D Part” page, given in Figure B.45, is used to create or delete parts. Part ID and the description of the part is entered into the textboxes and “Create” button is pressed to create it in the database. The part that will be deleted is selected using the radio button and “Delete” button is clicked to delete it. “List Part” page, given in Figure B.46, is used to list all the parts.

“C/U/D BOT” page, given in Figure B.47, is used to create, update, or delete the bill of tooling items. Tool ID of the cutting tool is supplied and “Submit” button is clicked. Parts that use this tool is listed. New parts can be added, used tool quantity for each part can be updated or the selected part can be deleted by clicking the “Create”, “Update” and “Delete” buttons, respectively. “Cancel” button returns to the first step. In “Search BOT” page, given in Figure B.48, bill of tooling can be searched according to tool ID, part ID or both part ID and tool ID. Search criterion is selected and criteria parameter is supplied. Clicking the “Submit” button shows the search result.

In “C/U/D Schedule Event” page, given in Figure B.49, part ID is supplied and the production schedule of that part can be seen clicking the “Submit” button. New production dates can be added, quantities of the existing ones can be updated, or the whole item can be deleted by clicking the “Create”, “Update” and “Delete” buttons. “Cancel” button returns to the first step. “Search Schedule” page, given in Figure B.50, is used to search the schedule according to part ID, date or both part ID and date. After selecting the search criterion clicking the “Select” button, criterion parameter is entered and search is started by clicking the “Submit” button.

In “C/U Purchase Order” page, given in Figure B.51, purchase orders can be created or updated by clicking the “Create” and “Update” buttons, respectively. Tool ID, order quantity and planned receiving date is supplied to fill the purchase order list. To update a purchase order, purchase order number is supplied and “Submit” button is clicked. Tool ID, quantity, planned receiving date, and status of the purchase order item can be updated by clicking “Update” button or the whole purchase order item can be deleted by clicking the “Delete” button. New purchase order items can be added by clicking the “Create” button after supplying the attributes given above. Status can be “OK” or “Not OK” mentioning if the purchase order item is received from the vendor or not, respectively. “Search Purchase Order” page, given in Figure B.52, is used to search the purchase orders according to purchase order no, ordered by, purchase order date, tool ID and status, and tool ID and planned receiving date. After selecting the criterion “Select” button is clicked, criterion parameter is entered and “Submit” button is clicked. Purchase order items can be retrieved by clicking the link written as the purchase order number. Cutting tool attributes can be retrieved by clicking the link written as the tool ID in the purchase order list page.

In “Receive Cutting Tool From Vendor” page, given in Figure B.53, cutting tools received from vendors are recorded. Tool ID and the purchase order number is entered and then “Submit” button is clicked to create a receiving. After

a receiving is created, the status of the purchase order item is changed to “OK”. “Search Receiving From Vendor” page, given in Figure B.54, is used to search the receivings according to receiving no, received by, receiving date, or tool ID. After selecting the search criterion by clicking the “Select” button and entering the criterion parameter, “Submit” button is clicked to start search. Receiving items can be seen through the link written as receiving number and cutting tool attributes can be seen through the link written as tool ID in the receiving list page.

When the system administrator logs in the system, “Administrative Site” link appears in the lower frame. Users of the system are managed in the “C/U/D User” page, given in Figure B.55, in the administrative site. New users can be added clicking the “Create” button. Existing users can be updated or deleted clicking the “Update” or “Delete” buttons. For these operations, badge number of the user must be entered into the “Badge No” field. First the personal information of the user is created or updated. After clicking the “Submit” button, title of the user is created or deleted in the next page. According to the title chosen, user’s accessibility to various functions of the tool management system differs.