COMPONENT INFORMATION SYSTEM

PROJECT WORK DONE AT

ISRO SATELLITE CENTRE,
BANGALORE

PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the award of the degree of
M.Sc Applied Science (Computer Technology)
Of Bharathiar University, Coimbatore.

Submitted By

N.ARUN
(Reg. No. 9937Q0001)

Guided By

EXTERNAL GUIDE

Ms. Bhuvaneswari Chandrasekar,
Engineer – SC,
ICG Division,
ISAC. Bangalore – 560 037.

INTERNAL GUIDE

Dr.S.Thangaswamy, Ph.D.,
Head of the Department,
Dept. of Comp.Sci and Engg.,
Kumaraguru College of Tech,
Coimbatore – 641 006.

Department of Computer Science and Engineering
KUMARAGURU COLLEGE OF TECHNOLOGY
Coimbatore – 641 006.
May 2001.

CERTIFICATE

CERTIFICATE

This is to certify that the project work entitled

"COMPONENT INFORMATION SYSTEM"

Done by

N.ARUN
(REG. NO. 9937Q0001)

Submitted in partial fulfillment of the requirements for the award of the degree of **M.Sc Applied Science (Computer Technology) of Bharathiar University.**

Professor and Head 23/4/'01

Internal Guide 23/4/'01

Submitted to University Examination held on 27/4/2001

Internal Examiner 23/4/'01

External Examiner 27/4/2001

भारत सरकार
अन्तरिक्ष विभाग
इसरो उपग्रह केन्द्र
पोस्ट बॉक्स नं : 1795 हवाई पत्तन मार्ग,
विमानपुरा डाक घर, बेंगलूर - 560 017. भारत.
दूरभाष : 5266251, ...
......................................(D)
फैक्स : ....

सत्यमेव जयते

GOVERNMENT OF INDIA
DEPARTMENT OF SPACE
ISRO SATELLITE CENTRE
Post Box No. 1795, Airport Road,
Vimanapura Post, Bangalore - 560 017, India.
Telephone : 5266251, ...
....(D)
Fax : ....

CERTIFICATE

This is to certify that the Project Report entitled "**Component Information System**" is a bonafide record of the work done by **N.Arun (Reg.No. 9937Q0001)**, doing final year **M.Sc. Applied Science Computer Technology in Bharathiar University, Coimbatore**, under my guidance and supervision in Components Division, at **ISRO Satellite Centre, Bangalore**, from January 2001 to April 2001 towards the partial fulfillment of the requirements for the award of the Degree in **Master of Science in Applied Science Computer Technology of Bharathiar University, Coimbatore**. The work has been completed successfully to my satisfaction.

Bhuvaneswari
(Ms. Bhuvaneswari Chandrasekar)

Ms. Bhuvaneswari Chandrasekar
Engineer – SC
ICG division
ISAC. Bangalore.

DECLARATION

DECLARATION

I hereby declare that the project entitled

"COMPONENT INFORMATION SYSTEM"

Submitted to **Bharathiar University** as the project work of **M.Sc Applied Science Computer Technology** Degree, is a record of original work done by me under the supervision and guidance of **Ms. Bhuavaneswari Chandrasekar, Engineer – SC, ICG Division, ISAC. Bangalore** and **Dr.S.Thangaswamy, Ph.D.,Head of the Department of Computer Science and Engneering, Kumaraguru College of Technology, Coimbatore** and this project work has not found the basis for the award of any Degree/Diploma/ Associate ship/Fellowship or similar title to any candidate of any university.

Signature of the student

Place: COIMBATORE

Date: 23/04/2001

Countersigned by

S. Thangaswamy 23/4/'01

(Internal Guide)

Bhuvaneswari

(External Guide)

DEDICATED TO
MY BELOVED PARENTS

CONTENTS

CONTENTS

ACKNOWLEDGEMENT

ACKNOWLEDGEMENT

I would like to thank the following people who helped me in accomplishing the task of completing this project and writing this report.

I am grateful to **Mr. Arun Batra,** Group Director CIG, for giving me this opportunity to work for this project.

I thank to **Mr.B.K.S.V.L.Varaprasad** who as a guide gave me excellent guidance, valuable suggestions and guided me through out the project.

I thankful to **Mrs.BhuvaneswariChandrasekar** who gave me ideas in developing this project, guided me through out the project and put it to successful comple

tion.

I thankful to **Mrs. Sheela Iyer**, **Mr. Ananth Ramu** and **Mr.Sriram** for giving me this opportunity to do project work at ISAC.

Its my great privilege and pleasure of mine to express my deep sense of gratitude to esteemed **Dr. K.K.Padmanaban B.Sc., (Engg)., M.Tech., Ph.D.,** Principal, K.C.T college allowing me to take on this project.

I express my deep sense of gratitude to **HOD & Prof Dr.S.Thangaswamy, B.E., (Hons)., Ph.D.,** for guiding me with his valuable and timely support and extended me through out the project, with his valuable guidance comments and through out my M.Sc Computer Technology course.

I reciprocate the kindness shown to me by my friends and family members, staff members of my department of Computer Science.

SYNOPSIS

SYNOPSIS

Indian Space Research Organization is a Central Government Organization involved in construction of Satellites and development of such space related technologies.

Component Information System is an inventory control. This is a major technique that effectively controls the costs as well make sure the availability of materials.

The application provides the following benefits.

- Reduces the time for choosing the appropriate component for a specified space vehicle.
- Reduces the Inventory while ensuring the availability of stocks.
- Reduces the cycle time for purchasing of items.
- Improves the vendor relationships by increasing the transparency.

Broadly the application provides secured access to the users of ISAC, which aims to achieve the above-mentioned benefits.

Briefly,

The Satellite system is divided into various levels yielding various sub-systems, packages and component level. According to this each satellite requires some component and proper selection of component play major role in this project. Availability is very important to maintain the project schedule. A need was felt to computerize the inventory of components from procurement to fabrication.

The Component Information System is mainly inventory, which is controlled by three modules. The modules, which are as follows:

1. **Component Deposit 1**: This module is mainly to deposit the component, which are as new to the store. These components are bought from vendor who manufacture for ISAC purpose.

2. **Component WithDrawal**: This module is mainly for with drawl of component, which is placed in the store. The withdrawal of component is done through request.

3. **Component Deposit 2**: This module mainly explains about the component life period, i.e., the components, which as soon as get life period over are again been checked in the component division and reentered in to the store.

To build this project the software used for the database design is Oracle and for interface design and data accessing, Visual Basic is used.

INTRODUCTION

I. INTRODUCTION

1.1 PROJECT OVERVIEW

A Component Information System is a system that aids management in making and carrying out, controlling decisions and performing its job in an effective manner, by providing timely, accurate and meaning full data or information. The systems that monitor and retrieve the data from transactions and operations with in the firm filters and selects data present them as information.

Objectives:

- The software provides an easy interface for the organization to edit its component details and upload the same.
- The software also provides a user-friendly interface for the organization to change or add new users and password.
- An interface to edit the details of a component and uploads the same.
- An interface to ensure strict security constraints, which allows the super user to provide access to various modules of the package to various users.
- Vendor information's are stored and processed for every given component and helps the user to identify the suitable vendor.
- Developing an error free interface to produce the components information.
- Developing an efficient Database system to handle the data about all the above detail.

1.2 ORGANISATION PROFILE

The Indian Space Research Organization (ISRO) is the prime agency charged with the responsibility laid down by the space commission and the department of space for executing the research and development programs and the schemes of the department of space.

From its beginning in 1972 as the Indian Scientific Satellite Project (IISP) for the realization of Aryabatta, the ISRO Satellite Center has today grown in to the nations premier center for research and development in satellite technology. ISAC has so far successfully planned and executed several satellite projects of ISAC, namely Aryabatta, Bhaskara I and II. The Satellite of Rohini Series, Apple, Sross-I, II, C, C2, IRS-1A and 1B, INSAT 2A and 2B. ISAC situated in Bangalore carries out all the functional, R and D tasks and infrastructure facility with an overall staff strength of 2500 members.

ISRO Satellite Center has thus contributed majestically to India's technological advancement and taken our country's name to the list of the few countries, which have been able to build and launch satellite launch vehicles.

The satellite launch craft program of ISAC is aimed at achieving capability to launch from with in India, scientific, technological and operational satellites for communication and Remote Sensing applications. The program started with SLV-3 launched in July 18 1980 followed by ASLV, launched successfully in 1992. The organization proved its capability to launch the remote sensing satellites using PSLV.GSLV, incorporating a cryogenic stage will provide indigenous capability to launch INSAT class for communication satellites. All the launches are conducted from ISRO SHAR center at sriharikota.

ISAC is now aiming to launch GSAT-I and INSAT series of communication satellites in to geo-stationary orbits. Once in orbit this type of satellites will serve various purposes as:

1. Domestic long distance applications.
2. Meteorological earth observations and date entry.
3. Radio and television program distributions.
4. Search and rescue missions.

ISRO Satellite Center has set up facilities for electronics and mechanical fabrication and environmental facilities including a Large Space Simulation Chamber (LSSC). Facilities and expertise for integration ground check out and testing of spacecraft and pre-launch support also forms a part of the center's activities.

INDIGENISATION AND COMPONENTS GROUP

ISRO Satellite Center is organized into Mechanical systems area, Electronics area, and Control and Mission area. The center also has independent groups like Indigenisation and Components Group, Program Planning and Evaluation Group, Systems reliability group etc.

The Indigenisation and components Group (ICG) at ISAC has the prime responsibility of performing Indigenisation of components together with screening of components used at various parts of satellite, rocket etc. The group therefore plays the important part of checking and evaluating the performance level of the components and passing the components to be used for flight.

The components therefore have been broadly classified into

1. ICs
2. Microwave and passive components.
3. Discrete devices.

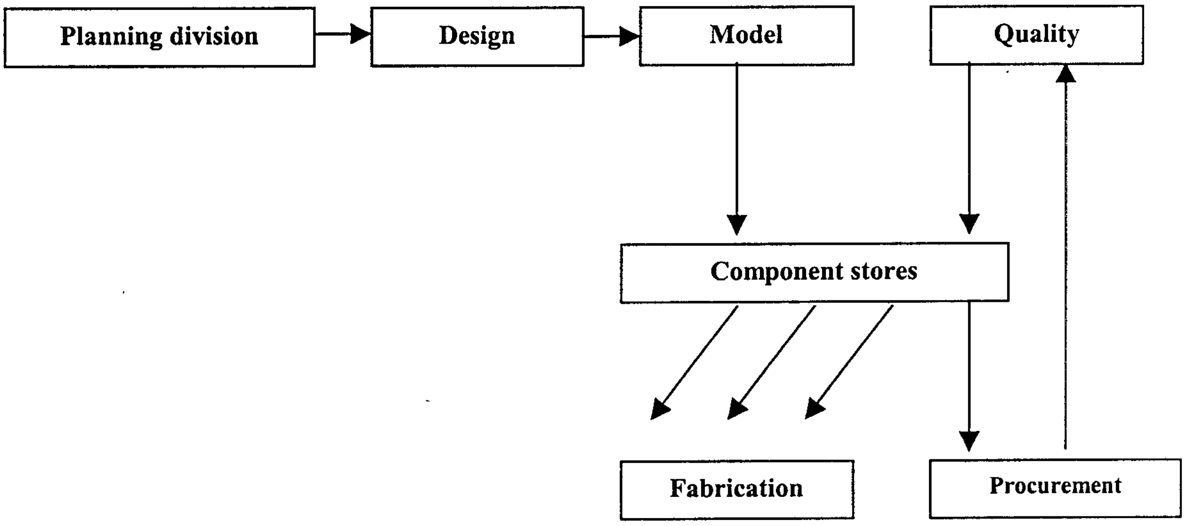
By screening we mean the sequence of non-destructive tests performed to eliminate infant mortality of the components. That is, tests are performed to components to check for manufacturing and work man-ship errors and then passed to be used by the satellites.

These components thus screened and evaluated and then sent along with the reports to bonded stores from where they are actually issued to the fabrication facility for fabrication.

SYSTEM STUDY AND ANALYSIS

II. SYSTEM STUDY AND ANALYSIS

This section describes the flow of general flow of information in the system. It describes the various problems the software must solve, that are the perspective of the project, functions of the project and the user characteristics. Information content represents the information domain that needs to be processed by the system. The information flow represents the actual flow of information into the system.



The general process in ISAC is as follows:

The process designing or building a satellite is broken into a various levels, the first being that of various sub systems of the satellite (main system). These subsystems are further divided into packages and finally come to the components. The components division of ISAC involves validation of component for flight usage. Such components that are procured, fabricated and supplied for the demand made.

Once a space ship is planned (the planning division) a design for the vehicle is done (design). Based on the design a model should be prepared. To prepare the model and following packages, they are developed and based on availability of components in the stores if not available then procurement is made, the component is tested for its quality and the stores is updated.

To make sure of the high reliability of these components a technique called the PPL (standards for Preferred Parts List) is used.

The list provides complete description of various components that are to be purchased based on various parameters of the components with their previous history of usage. This history helps to identify the effectiveness of the components.

2.1 EXISTING SYSTEM – LIMITATIONS

The existing Component Information System is basically built with FoxPro and running in Intranet. The limitations are as follows:

- This existing system has not been satisfied because the past package is not available with all relational concepts.
- The existing system was not very Graphical User Interface and it has to be redesigned with latest technologies.
- Recent technologies are better in handling the table design that FoxPro, which shows it, has certain limitations.
- There is no flexibility in questioning of data in FoxPro.
- Client / Server architecture of present technologies becomes to be better than FoxPro.

Therefore it necessary to user a GUI interfaces package as Visual Basic and Relational Database Management as Oracle.

2.2 PROPOSED SYSTEM

The proposed system is to take care of all the shortcomings of the existing system and is to be developed with the latest technologies and methodologies. It is to meet the software requirement specification and with good interface communications. The modules that are developed will be integrated with the Component Information System. After development the system will put to various testing procedures.

The testing procedures would ensure that the new system is error free from bugs. All the functional part of the system would be tested for defects. This would optimize the system in the usage of memory.

The processing schemes take the concept of time complexity by optimizing the loop structures and memory allocations to dynamic variables. This would ensure optimized usage of resources making the system more fast and reliable. The coding structure and the standard would ensure that there are no memory leaks and that the allocated memory is returned to the free store.

The proposed system is to have tool tip help to the menus, tool bar, item etc. This would ensure an improved way to access the functions of the systems. High lighting facility in the data display screens would ensure quick viewing of data. The controls would be reused in various display formats.

Thus the proposed system is to take advantage of all the new technologies ensured a smooth functioning.

2.3 REQUIREMENTS ON NEW SYSTEM

2.3.1 PURPOSE:

- Minimize the effort of ISAC users by using the GUI based interface packages to make it user friendly.
- Minimize the effort by the ISAC in searching for the appropriate components for design and development of Satellite, i.e. procurement procedures are simplified.
- Provide an easy access to the information about the inventory of components available, history – the success proportion, failure of specific components in their various areas of usage from various branches\offices of ISAC situated around India.
- Reduces the effort of the planning department to choose appropriate components for the specific spacecraft.
- Reduces the effort put forth by the user in checking the order and placing the order.
- Providing access to all above state with proper security.
- Minimize the effort of selecting the component and vendor by preparing parts lists.

2.3.2 SCOPE:

- The software provides an easy user-friendly interface to the organization to view through the organizations inventory holdings, supplier information, component information etc.

- It also monitors inventory and provides queries and enhanced reports to the top management of the organization for taking futuristic decisions.

2.3.3 GENERAL DESCRIPTION

2.3.3.1 PRODUCT PERSPECTIVE:

- Component related data are available as per the users.
- The new components and its information like parameters of components etc are register in stores.
- Retrieval of component stock information transaction etc are provided for authorized user.
- Information about all vendors of various components, their location and contact details are maintained.
- Information about the staff members who can provide specific information about components for the benefit of suppliers and ISAC users also.
- Providing access to various modules of the project to various types of users based on their requirement and their importance with respect to the system.
- Provision to add new users of various types to the system and allowing the existing users to modify the user name password etc.

The perspective of the projects is to provide the user with an effective database to store the components parameter, stock transaction related data.

2.3.3.2 PRODUCT FUNCTION:

The following points would briefly explain the functions of the proposed system.

- A user friendly interface for Data entry of components and their parameters.
- Providing a status report of all listed components on various parameters.
- Provide an interface for access of information through Intranet.
- Provide reports on failure history of the components.
- Providing the availability stock details.
- Provision for security.
- Providing of clarification information.

2.3.3.3 GENERAL CONSTRAINTS:

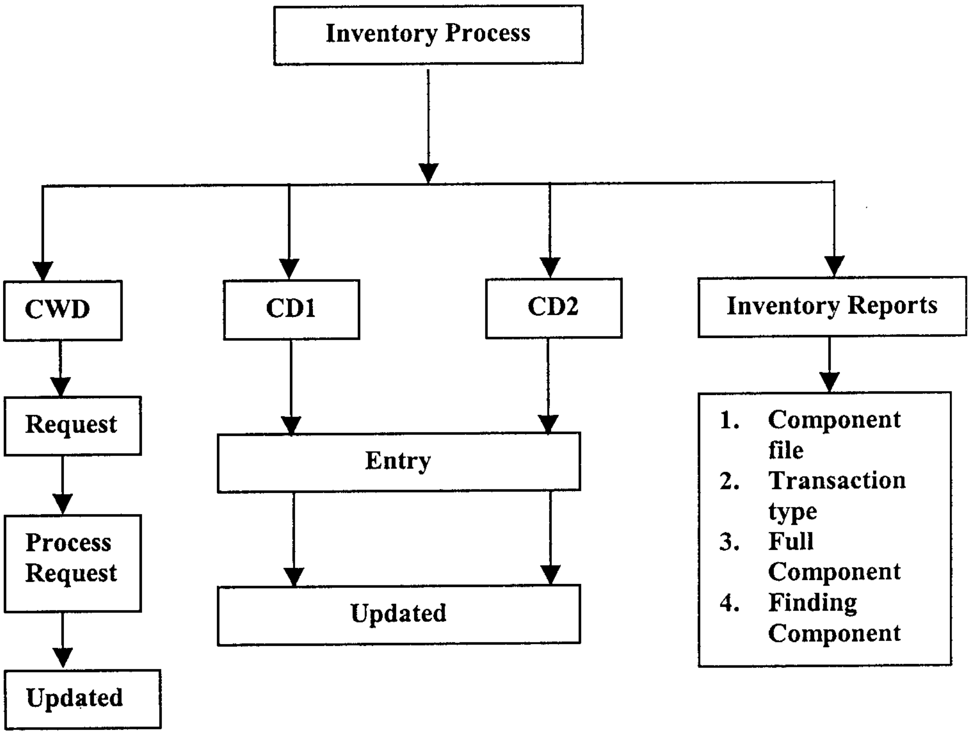
The constraints regarding components information system is

- The software should be loaded in the Intranet then only every other department inside ISAC can use the system.
- The user should not forget the password in which it is been given to him.
- Training has to be given to the ISAC users.
- Generally the time gap occurs for adding, deleting, viewing and modifying a data in to the database.
- The password that is given to each person must be confidential.
- Whenever new components are being added into the store it must be checked.
- Administrator is the super user to perform any operation in the package.

2.3.4 SPECIFIC REQUIREMENTS

2.3.4.1 FUNCTIONAL REQUIREMENTS

Mainly by using all the inputs the information processing is done. When these inputs are being provided then the transaction table like history, card and segmentation table gets updated or deleted or viewed. The diagram that illustrates the Component Information System process:



CWD – Component With Drawl

CD1 – Component Deposit 1

CD2 – Component Deposit 2

Integrated Components Database System is component management system, which manages all transactions of components between stores and different user of

ISAC. The main objective is to minimize the manual work that is carried out so far and as far as possible also make retrieval of date more accurate and quicker. Also maintain accurate and precise information about the components, their parameters, and the values of their parameters and the associated history of the components.

Functionality's are as follows:

1. The Inventory
2. Component with drawl
3. Component deposit 1
4. Component deposit 2
7. Security
8. User and Password
9. Storcs

The Inventory System:

This module is the very important module in the whole of ICSDBS since this completely maintains the component related data and stock details. That is all details of the master database; preferred parts list, history or failure analysis database and the technical analysis databases are stored accessed, retrieved and manipulated.

Component With Drawl:

This system deals with issue of components as per the requests made by the ISRO staff members. The requests are of:

1. **Request-I**: The staff person who places a general request for the components from the stores, which has its part number and quantity.

2. **Request-II:** The staff person who places a specific requests for the components from the stores where in they are not particular about part number, quantity with extra details.

According to the request the super user prepares a preview of the components and displays it to the staff. If satisfies he will ask the storekeeper to issue it.

Component Deposit 1:

The system deals with receipt of components as per the components, which are being purchased from the supplier. The actual process, which deals, is that when an ISAC organization wanted for a component they places an order for that component to the supplier. Supplier will deliver this ordered component to the organization, where in the organization checks for the components by giving it to component division. If the component meets all the required stands they will by the components and provide a receipt on it.

Component Deposit 2:

This option is mainly used for the already available components, which has to be validated for flight usage by components division and to be returned to stores.

Security Module:

There are various levels of security in ICSDBS. This is because of the fact that various people in various sections of ISAC use this package. Therefore security at various stages of ICSDBS is incorporated. For instance a user belonging to a particular department of ISRO can use only those module like only viewing of components available in the master. Access to any other module in the system is denied.

Each user can provide with their own user id and password, which protects accessing limitations. Only the request provider can delete his request but other user can view the request.

Only the administrator can determine and decide what user can access and what they cannot.

User and Password Module:

This module is to be used only by the super user who by this module can add new users, delete the existing user and can access their passwords.

Stores Module:

This module allows the user to view the stocks available in the stores, orders that are placed, orders that are pending etc. This module takes the input of components from the components issue module, components receipt module and components return notification module.

2.3.4.2 LIST OF INPUTS:

The input of Component Information System process is divided in to 3 stages.

Stage – I: The input to stage-I represents about the store receipt voucher. The inputs are entered for the new components, which enter in to the store.

Reference number, Date of the item entered, Item number, Yearweek of the item, Expiry date, Quantity which enters in to the store, Manufacturer number, Type of the receipt, Entered by which user with date, updated by the user with date, status of the receipt.

Stage – II: The input of stage-II represents about stores issue voucher. The input of this stage is a kind of request to the store in search of the component.

Stock number, Request number, Type of request, Component part, Requested quantity, Status of the request, Requested by which user, Type of the component, Part number of the component.

After request, the storekeeper will process it. For that the inputs are

Request number, date, Request link, Yearweek of the component, Expiry date, Quantity of component, Request component class, Request card number, Request date.

After this processed the query has to be updated for that the inputs are

Issued yearweek, Issued quality id, Manufacturer, Expiry date, Stock issue number with date.

Stage – III: The input of stage-III represents about the stores return note, where the components are put back in the stores after validation by component division for the already procured component.

Stock number, Store issue - number, date, quantity, part , returned - from, stock number, card number, part number, yearweek, date, expiry date, manufacturer, quantity, stores return - number, date, changed by with date, class number, quality number, part number, component number.

For all the 3 stage the 3 table like card, history, detail has to be updated. Inputs are Stock number, type number, part number, date on entered, card number, part type, quality of the item, yearweek, expiry date, manufacturer, quantity, changed by, changed date, satellite number, destination code.

2.3.5 PERFORMANCE REQUIREMENTS

2.3.5.1 SECURITY

There are various levels of security in ICSDBS . This is because of the fact that various people in various sections of ISAC use this package. Therefore security at various stages of ICSDBS is incorporated. For instance a user belonging to a particular department of ISAC can use only those module like only viewing of components available in the master. Access to any other module in the system is denied. Only the storekeeper can determine and decide what user can access and what they cannot.

Users of this security system can be classified into following types.

1. The super user or the administrator.
2. The registered ISRO user.
3. The guest Users.
4. The registered suppliers.

2.3.5.2 AVAILABILITY

Availability is the probability that a program is operating according to requirements at a given point of time. The availability is an indirect measure of the maintainability of software.

The Component Information Systems consist several screen formats, which contains tools to enter data. If the data entered is appropriate or data is missing the system indicates possibility of an error. This accounts for the maintainability of a system; thus enhancing availability.

2.3.5.3 CAPACITY

Capacity measures number of system a software can access. The Component Information System works through Intranet. The package loaded for Intranet is window NT. Window NT can hold several systems. Hence this capacity of window NT indicates the capacity of Component Information System.

2.3.5.4 RESPONSE TIME

Response time is the time with in which a system identifies the instruction of the user and responds to it.

The response time of the system is made efficient and quick so that the requests are accepted according to their validity and the result can be delivered to the user with an immediate effect.

2.3.6 DESIGN CONSTRAINTS

2.3.6.1 STANDARD COMPONENTS

The standard component which is used to built the Component Information System is a powerful front end tool as Microsoft Visual Basic Studio 6.0 version and an relation database management system as Oracle 6.0

2.3.6.2 HARDWARE LIMITATIONS

Since Visual Basic and Oracle technology is used to develop the user interface and database design with respectively the hardware limitations required is

- A Pentium processor of 60mhz speed. (To make sure that application does not take too long to run even a lessen configurations would run the application)
- A Random access memory of 16 MB.
- A hard disk space minimal enough to load the application
- A mouse and keyboard.
- A windows at least Microsoft Windows 95/Windows NT 3.51

2.3.6.3 USER INTERFACE AND SCREEN FORMATS

It is required to maintain certain GUI standards during development of this system. The following are the generalized forms.

1. Label Box: Label displays a text that the user cannot modify or interact with.
2. Check Box: Check Box displays a True or False or Yes or No option.
3. Text Box: Text Box is a control used to display messages and enters text.
4. List Box: List Box displays a list of items from which a user can select one.
5. Combo Box: Combo Box contains a Text Box and a List Box. This allows the user to select an item from the dropdown List Box, or type in a selection in the Text Box.
6. Hscroll Bar or Vscroll Bar: Allows controlling the user to select a value with in the specified range of values.
7. Timer control executes the timer events at specified intervals of time.

8. Picture Box displays icons or bitmaps and metafiles.
9. Command Button carries out the specified action when the user chooses it.
10. Menu which mainly used to handle the different forms to be loaded at run time and also for selection of values form the menu item as an input to a form
11. Data grid control displays and enables data manipulation of a series of rows and columns representing records and fields from a record set object.

2.3.6.4 HARDWARE INTERFACES AND SOFTWARE INTERFACES

Hardware Interfaces:

In this division we mention about the new hardware requirements if any needed to run the proposed system. The current working environment is as follows.

ISRO satellite center has a highly sophisticated computer facility. The IBM computer center is equipped with several servers and workstations. These workstations are of three major types. Sun Solaris workstations with about 16 machines operating on SUN Solaris operating system, which is similar to that of Linux operating system. The HP workstation with same number and about 32 IBM machines, which can run on AIX(Advanced Interactive Executive) operating system. All of these provide a GUI environment facilitating a powerful environment for software development.

This scenario as explained above makes it very clear that no hardware interface is required in specific.

Software Interfaces:

The project of Component Information System is an existing system which is basically built with FoxPro so the Database are created with in these and data's are also stored in this package. The interface is being developed in such a way that the data, which are stored in FoxPro, are being imported in the Oracle.

This software interfaces are developed using BCP (Bulk Copy) call which specifies that all the data in FoxPro are being converted into text format and retrieved in the Oracle and by using Bulk out procedure all the data are posted into the Oracle.

2.3.7 OTHER REQUIREMENTS

2.3.7.1 OPERATION REQUIRED BY THE USER

There will be a storekeeper for component information system, created with a userid and password. This user can modify the activities at any time. The user is only capable for loading the components in to store entry with fixing a name, number of stocks and from which manufacturer. Other operations are being performed as by the program.

2.3.7.2 SITE ADAPTATION REQUIREMENTS

The form will be running under Windows NT operating system. There is no system / installation specific data to be entered for the operation component information system.

2.3.7.3 USER CHARACTERISTICS

Users of this system can be broadly classified into following types.

1. The super user or the administrator.
2. The Authorized user.

3. The guest Users.
4. The trainee suppliers.

The Super user or the Administrator:

This user is the one who has access to all parts of the system. He controls all the other users from accessing the software. This user can create a new user and provide him with access to parts of the system, which the new user is eligible to use. He also has the access to all other users password can stop or change the access of any other user.

The Authorized users:

This category of user is an ISAC registered user who registers and gets access to the various functions of the system through the super user. He is allowed to modify his user name password etc. This user gets the user code as his employee number without which his account may not be created.

The Guest users:

These can be ISAC employees or visitors form any other organization. They are allowed only to view the functioning of the software. No addition, deletion or modification functions of any category are allowed to these users. The administrator can create other users.

The Trainee suppliers:

These are the users from ISAC who comes to do the project work. These users are provided with username and password, which means to say that he cannot change the name and password and visibility of components can only be done.

PROGRAMMING ENVIRONMENT

III. PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION

Processor	: Pentium ® II
Processor Speed	: 350 MHz
Hard Disk	: 4.00 GB
CD – ROM	: 32 X SCSI CD – ROM Drive
DISPLAY	: 17 " Color Monitor
RAM	: 64 MB, SDRAM

3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED – REASONS FOR THE CHOICE

1. VISUAL BASIC

Visual Basic is an ideal programming language for developing sophisticated professional applications for Microsoft Windows. It makes use of Graphical User Interface for creating robust and powerful applications. The Graphical User Interface as the name suggests, uses illustrations for text, which enable users to interact with an application. This feature makes it easier to comprehend things in a quicker and easier way.

The Integrated Development Environment:

One of the most significant changes in Visual Basic 6.0 is the integrated development environment (IDE). IDE is a term commonly used in the programming world to describe the interface and environment that we use to create our applications. It is called integrated because we can access virtually all of the development tolls that we need from one screen called an interface. The IDE is also commonly referred to as the design environment, or the program.

Tool Box:

Toolbox contains a set of controls that are used to place on a form at design time thereby creating the user interface area. Additional controls can be included in the toolbox by using the components menu item on the project item on the project menu.

1. Label Box: Label displays a text that the user cannot modify or interact with.
2. Check Box: Check Box displays a True or False or Yes or No option.
3. Text Box: Text Box is a control used to display messages and enters text.
4. List Box: List Box displays a list of items from which a user can select one.
5. Combo Box: Combo Box contains a Text Box and a List Box. This allows the user to select an item from the dropdown List Box, or type in a selection in the Text Box.
6. Hscroll Bar or Vscroll Bar: Allows controlling the user to select a value with in the specified range of values.
7. Timer control executes the timer events at specified intervals of time.
8. Picture Box displays icons or bitmaps and metafiles.
9. Command Button carries out the specified action when the user chooses it.
10. Menu which mainly used to handle the different forms to be loaded at run time and also for selection of values form the menu item as an input to a form

11. Data grid control displays and enables data manipulation of a series of rows and columns representing records and fields from a record set object.

Form serves as a window that can be customized and controls, graphics and pictures can also be added to it.

Event Driven Programming:

Visual Basic programs are built around events. Events are various things that can happen in a program. This will become clearer when studied in contrast to procedural programming. In procedural languages, an application written is executed by checking for the program logically through the program statements, one after another. In event driven applications, the program statements are executed only when a particular event calls a specific part of the code that is assigned to the event.

For example textbox controls support various events such as click, change, MouseMove and many more, listed in the properties drop-down list.

- The code entered in the change event fires when there is a change in the contents of the textbox.
- The click event fires when the textbox control is clicked.
- The mousemove event fires when the move is moved over the text box.

Data Access using RDO:

Remote Data Objects (RDO) is specifically designed to access remote ODBC relational data sources, and makes it easier to use ODBC without complex application code. RDO is a primary means of accessing SQL Server, Oracle, or any relational database that is exposed with an ODBC driver.

The general characteristics of RDO are:

- Simplicity (when compared to the ODBC API).
- High performance against remote ODBC data sources.
- Programmatic control of cursors.
- Complex cursors, including batch.
- Ability to return multiple result sets from a single query.
- Synchronous, asynchronous, or event-driven query execution.
- Reusable, property-changeable objects.
- Ability to expose underlying ODBC handles (for those ODBC functions that are not handled by RDO).
- Excellent error trapping.

Compared to the older Data Access Objects (DAO) technology, RDO is a smaller, faster, more sophisticated alternative. RDO is especially capable of building and executing queries against stored procedures and handling all types of result sets, including those generated by multiple results set procedures, those returning output arguments and return status, and those requiring complex input parameters.

If you have an RDO application that works well today, there's no reason to change it. But if you need to extend your application to access other kinds of data, you should consider reengineering to use ADO.

If you are developing a new application, you should consider using ADO instead.

Data Access using ADO:

ActiveX Data Objects (ADO) is designed to be an easy-to-use application-level interface to any OLE DB data provider, including relational and non-relational databases, e-mail and file systems, text and graphics, and custom business objects, as well as existing ODBC data sources. Virtually all of the data available throughout the enterprise is available using the ADO data access technology.

ADO is easy to use, language-independent, implemented with a small footprint, uses minimal network traffic, and has few layers between the client application and the data source — all to provide lightweight, high-performance data access.

The general characteristics of ADO are:

- Ease of use.
- High performance.
- Programmatic control of cursors.
- Complex cursor types, including batch and server- and client-side cursors.
- Ability to return multiple result sets from a single query.
- Synchronous, asynchronous, or event-driven query execution.
- Reusable, property-changeable objects.
- Advanced recordset cache management.
- Flexibility — it works with existing database technologies and all OLE DB providers.
- Excellent error trapping.

The simple semantics of ADO and universal application mean minimal developer training, rapid application development, and inexpensive maintenance.

ADO is Microsoft's premier data access technology. The ADO data access technology and their partners OLE DB comprise the recommended solution for all data access. If you are developing a new application, you should definitely use ADO.

If you're considering migration to ADO, you have to decide if characteristics and benefits of ADO are enough to justify the cost of converting existing software. Older code written in RDO and DAO will not automatically convert to ADO code. However, whatever solutions you previously developed using other data access strategies can definitely be implemented using ADO. In the long run, you should convert to ADO.

Data Access Using Open Database Connectivity (ODBC):

Open Database Connectivity (ODBC) provides a universal database connectivity applications programming interface (API) that enables applications to access data in a wide range of proprietary databases. Based on the X/Open SQL Access Group's Call Level Interface (CLI) specification, ODBC is an open, vendor-neutral way to uniformly access data stored in different formats and database engines.

ODBC is the most widely used interface to relational data. It's also quite fast, but you pay for the fast access with complex application code.

The general characteristics of ODBC are:

➢ Highly efficient performance.

- Coding difficulty.
- Reasonable memory requirements.
- Compatibility with existing database technologies.
- Portability across many operating system platforms.
- A connection model that allows for different networks, security systems, and database options.

As the standard interface to relational data, your application can access a lot of data using ODBC. But ODBC does require that your data look like a relational database, so it's not always the best way to expose data. If you don't have a relational database, it can be very difficult to write an ODBC driver to expose your data because you basically have to write a relational engine on top of the existing data structure.

A number of factors influence choosing the ODBC approach. These include a requirement for high performance, more granular control over the interface, and a small footprint.

The ODBC API is considerably harder to code than the object-based interfaces, but provides a finer degree of control over the data source. Unlike other data access technologies (such as ADO, RDO, or ODBCDirect), the ODBC API has not been made "bullet proof." While it's fairly easy to create ODBC errors during development, the ODBC API provides excellent error handling with detailed error messages. In general, developing, debugging, and supporting an ODBC API application requires a lot of knowledge, experience, and many lines of code. As a general rule, developers prefer to access data by using a simpler, higher-level object interface such as ADO.

ODBC is not suitable for non-relational data such as ISAM (Indexed Sequential Access Method) data because it has no interfaces for seeking records, setting ranges, or browsing indexes. ODBC simply was not designed to access ISAM data. While you can use the Microsoft Jet ODBC driver to handle ISAM and native Microsoft Jet engine data, what is really happening is that the Microsoft Jet database engine converts the ISAM data to relational data and then provides limited ISAM functionality. Performance in this situation is slow due to the extra layer imposed by the Microsoft Jet Engine.

If your application requires very fast access to existing ODBC data, and you're willing to write many lines of complex code (or you already have a lot of ODBC code available for reuse), ODBC is a good choice.

Microsoft Data Report Designer:

The Microsoft Data Report designer is a versatile data report generator that features the ability to created-banded hierarchical reports. Used in conjunction with a data source such as the Data Environment designer, you can create reports from several different relational tables. In addition to creating printable reports, you can also export the report to HTML or text files.

Possible Uses:

- Automatically create reports that are exported in HTML format for instant distribution on the Internet.
- Create reports that show the sums of transactions occurring on a daily basis.

Data Report Designer Features:

The Data Report designer has several features:

1. **Drag-and-Drop Functionality for Fields**—Drag fields from the Microsoft Data Environment designer to the Data Report designer. When you do this, Visual Basic automatically creates a text box control on the data report and sets the DataMember and DataField properties of the dropped field. You can also drag a Command object from the Data Environment designer to the Data Report designer. In that case, for each of the fields contained by the Command object, a text box control will be created on the data report; the DataMember and DataField property for each text box will be set to the appropriate values.
2. **Toolbox Controls**—The Data Report designer features its own set of controls. When a Data Report designer is added to a project, the controls are automatically created on a new Toolbox tab named DataReport. Most of the controls are functionally identical to Visual Basic intrinsic controls, and include a Label, Shape, Image, TextBox, and Line control. The sixth control, the Function control, automatically generates one of four kinds of information: Sum, Average, Minimum, or Maximum. For more information about the Function control, see "Adding a Function Control to the Data Report."
3. **Print Preview**—Preview the report by using the Show method. The data report is then generated and displayed in its own window.
4. **Print Reports**—Print a report programmatically by calling the PrintReport method. When the data report is in preview mode, users can also print by clicking the printer icon on the toolbar.

5. **File Export**—Export the data report information using the ExportReport method. Formats for export include HTML and text.
6. **Export Templates**—You can create a collection of file templates to be used with the ExportReport method. This is useful for exporting reports in a variety of formats, each tailored to the report type.
7. **Asynchronous Operation**—The DataReport object's PrintReport and ExportReport methods are asynchronous operations. Using the ProcessingTimeout event, you can monitor the state of these operations and cancel any that are taking too long.

2. ORACLE

Relational Database System:

Dr. E.F.Codd first introduced the Relational Database Model in 1970. The relational model allows data to be represented in a simple row-column format. Each data field is considered as a column and each record is considered as a row of a table. Codd's rule is designed in such a way that when the database is ready for use it encapsulates the relational theory to its full potential. The twelve rules are listed below.

- ➢ The information rule.
- ➢ The rule of guaranteed access.
- ➢ The systematic treatment of null values.
- ➢ The database description rule.
- ➢ Comprehensive data sub language.

- The view updating rule.
- The insert and update rule.
- The physical independence rule.
- The logical data independence rule.
- The integrity independence rule.
- The distribution rule.
- The non-subversion rule.

Overview:

Oracle 8 is an object relational database management system (ORDBMS). It offers capabilities of both relational and object-oriented database systems. In general Objects can be defined as reusable software codes which are location independent and perform a specific task on any application environment with little or no change to the code.

Oracle products are based on a concept known as the client/server technology. This concept involves segregating the processing of an application between two systems. One performs all activities related to the database (server) and other performs activities that help the user to interact with the application (client).

A client or front-end database application also interacts with the database by requesting and receiving information from the database server. It acts as an interface between the user and the database. Further, it also checks for validation against the data entered by the user.

The database server or back end is used to manage the database tables optimally among multiple clients who concurrently request the server for the same data. It also enforces data integrity across all client applications and controls database access and other security.

Tools of Oracle:

The tools provided by oracle are:

- SQL * Plus - SQL * Plus is a Structured Query Language supported by Oracle. Through SQL * Plus we can store retrieve, edit, enter and run SQL commands and PL/SQL blocks. Using this we can perform calculations, list column definitions for any table and can also format query results in the form of a report.
- PL / SQL - PL/SQL is an extension of SQL. PL/SQL block can contain any number of SQL statements integrated with flow of control statements. Thus PL/SQL combines the data manipulating power of SQL with data processing power of procedural languages.

SQL:

SQL was invented and developed by IBM in early 1970's. SL stands for Structured Query Language. IBM was able to demonstrate how to control relational databases using SQL. The SQL implemented by ORACLE CORPORATION is 100% complaint with ANSI/ISO standard SQL data language. Oracle database language is SQL, which is used for storing and retrieving information in Oracle. A table is a primary database object of SQL that is used to store data.

SQL supports the following categories of commands:

- Data Definition Language
- Data Manipulation Language
- Transaction Control Language
- Data Control Language

The following are the benefits of SQL:

- Non-procedural language, because more than one record can be accessed rather than one record at a time.
- It is the common language for all relational databases. In other words it is portable and it requires very few modifications so that it can work on other databases.
- Very simple commands for querying, inserting, deleting and modifying data and objects.

Data Definition Language

The Data Definition Language is used to create an object, alter the structure of an object and also to drop the object created. A table is a unit of storage that holds data in the form of rows and columns. The DDL used for table definition can be classified into the following four:

- Create table command
- Alter table command
- Drop table command
- Truncate table command

Data Manipulation Language:

Data Manipulation commands are the most frequently using SQL commands. They are used to query and manipulate existing objects like tables. The DML commands are as follows:

- Insert command – Once the table is created the insert command is used to add one or more rows to a table.
- Select command – Request for information stored in a table can be done through the select command.
- Update command – Sometimes changes to the database become imminent. To reflect some changes in the existing records in a table the update command is used. With the update command we can update a single row or multiple rows or specific rows.
- Delete command – After inserting rows in the table can also delete them if required.

Transaction Control Language:

A transaction is a logical unit of work. All changes made to the database can be referred to as a transaction. Transaction changes can be made permanent to a database only if they are committed. A transaction begins with an executable SQL statement and ends explicitly with either rollback or commits statements and implicitly, i.e., automatically, when a DDL statement is used.

- Commit – This commit is used to end a transaction. Only with the help of the commit command, transaction changes can be made permanent to the database.

- Savepoint – savepoints are like markers to divide a very lengthy transaction to smaller ones. They are used to identify a point in a transaction to which we can later rollback. Thus savepoint is used in conjunction with rollback, to rollback portions of the current transactions.
- Rollback – a rollback command is used to undo the work done in the current transaction. We can either rollback entire transaction or rollback to a savepoint.

Data Control Language:

Data Control Language provides user with privilege commands. The owner of databases objects, say tables. Has the sole authority over them.

- Grant privileges – Allows them to perform operations with in their privileges.
- Revoke privileges – To withdraw the privilege that has been granted to a user.

SQL Set Operators:

Set operators combine the results of two queries in to a single one. The following set operators aid SQL in joining queries.

- Union – The union operator returns all distinct rows selected by both queries.
- Union all – The union all operator returns all rows selected by either query including duplicated.
- Intersect – Intersect operator returns only rows that are common to both the queries.
- Minus - Minus operator returns all distinct rows selected only by the first query and not by the second.

Views:

A view is a virtual table with no data, but can be used likely any other table. It is like a window through which you can view the data of another table, which is called the base table. This window is called a view and is given a name, and can be operated with some restrictions.

Packages:

A package is a database object, which is an encapsulation of related PL/SQL types, subprograms, cursors, exceptions, variables and constants. It consists of two parts, a specification and a body. In the package specification we can declare types, variables, constants, exceptions, cursors and subprograms. A package body implements cursors, subprograms defined in the package specification.

Exception:

Exceptions are designed for run time error handling, rather than compile time error handling. Exceptions and exception handlers are the method by which the program reacts and deals with run time errors. When an error occurs, an exception is raised. When an exception occurs, the control is passed to the exception handler, which is separate section in the program.

Stored procedures:

A stored procedure is collections of PL/SQL statements that reside on the Oracle server and can be executed by any user who has been granted execute permissions. A stored procedure has number of features that give it good advantages over a batch.

A stored procedure can:

- Be called from other procedures. This feature helps in increasing modularity.
- Return values, which could be an indication of success or failure.
- Produce different ouputs depending on information provided to it at the time of executing it. This information is passed to it in the form of parameters. This feature of parameter passing makes procedures more flexible.
- Reports back or return more than one value to the calling procedure or batch in the form of return parameters.

Triggers:

Triggers are special kind of stored procedures that are fired automatically when an insert, delete or an update takes place in a specified table. Unlike stored procedures, which must be explicitly invoked and executed, triggers are automatically invoked. Triggers can help maintain referential integrity in a database by maintaining consistency among logically related data in different tables. Since each trigger is specific to one or more of the data modification operations – update, insert or delete, there can be a maximum of three triggers per table. Each trigger applies to one table only. Triggers cannot be explicitly called and they cannot take parameters.

Constraints:

maintaining security and integrity of a database is the most important factor in judging the success of a system. This integrity can be applied to different degrees of severity. An integrity constraint is a mechanism is used to prevent invalid data entry into the table.

Following are the various types of integrity constraints:

- Domain integrity constraints – These constraints set a range, and any violations that take place will prevent the user from performing the manipulation that caused the breach. There are two types
  1. Not Null constraints.
  2. Check constraints.
- Entity integrity constraints
  1. Unique constraints – Unique constraints allows only unique values to be stored in the column. This constraint rejects duplicate of records when the unique key constraint is used.
  2. Primary key constraints – The primary key constraint avoids duplication of values. Its need is best felt when a relation has to be set between tables.
- Referential integrity constraints – This enforces relationship between tables. It designates a column or combination of columns as a foreign key. The foreign key establishes a relationship with a specified primary or unique key in another table, called the referenced key.

SYSTEM DESIGN & DEVELOPMENT

IV. SYSTEM DESIGN AND DEVELOPMENT

Design is a creative process. Although methods and guidelines are helpful, judgement and flair are still required to design a software system. A software design is a model of real-world system that has many participating entities and relationships. The design is used in a number of different ways. It acts as a basis for detailed implementation; it serves as a communication medium between the designers of subsystems; it provides information to the system maintainers about the original intentions of the system designers and so on.

The design process involves developing several models of the system at different level of abstraction. As a design is decomposed, errors and omissions in earlier stages are discovered. These feed back to allow earlier design models to be improved. There are two important design strategies and are summarized as follows:

Functional design:

The system is designed from a functional viewpoint, starting with a high level and progressively refining this in to a more detailed design. The System State is centralized and shared between the functions operating on that state.

Object oriented design:

The system is viewed as a collection of objects rather than as functions. Object oriented design is based on the idea of information hiding, here, the system state is decentralized and each object maintains its own state of information.

Objects have a set of attributed defining their state and operations acts on these attributes.

Data flow Diagrams:

Data flow diagrams (DFDs) are commonly used in problem in understanding the system and can be effectively used for partitioning during analysis. DFD shows the flow of data through a system- manual or automated including the process, datastore, and delays in the system. DFDs are thus central tool and the basis from which other components are developed. The transformation of data from input to output through may be described logically and independent of physical components. It shows the movement of data through different transformations in the system.

Notations:

Rectangle unclosed:

Show an external input or output, from or to the user respectively. User selection and reports generated as shown with a small circle.

Oval or Circle:

This symbol represents a transformation or a process. It is named with a small description about its function.

Rectangle:

It presents a data store. It can be a file, a part of the memory, arrays e.t.c. Secondary storage can also be represented.

Arrows:

These unidirectional named arrows represent the data flow. The names indicate the data.

The DFDs development for the project is as follows.

4.1 INPUT DESIGN

The input of Component Information System process is divided in to 3 stages.

Stage – I: The input to stage-I represents about the store receipt voucher. The inputs are entered for the new components, which enter in to the store.

Reference number, Date of the item entered, Item number, Yearweek of the item, Expiry date, Quantity which enters in to the store, Manufacturer number, Type of the receipt, Entered by which user with date, updated by the user with date, status of the receipt.

Stage – II: The input of stage-II represents about stores issue voucher. The input of this stage is a kind of request to the store in search of the component.

Stock number, Request number, Type of request, Component part, Requested quantity, Status of the request, Requested by which user, Type of the component, Part number of the component.

After request, the storekeeper will process it. For that the inputs are

Request number, date, Request link, Yearweek of the component, Expiry date, Quantity of component, Request component class, Request card number, Request date.

After this processed the query has to be updated for that the inputs are

Issued yearweek, Issued quality id, Manufacturer, Expiry date, Stock issue number with date.

Stage – III: The input of stage-III represents about the stores return note, which means some components may return from the user itself or it may return from the stores itself because of expiry date. The inputs are:

Stock number, Store issue - number, date, quantity, part, returned - from, stock number, card number, part number, yearweek, date, expiry date, manufacturer, quantity, stores return - number, date, changed by with date, class number, quality number, part number, component number.

For all 3 stages the 3 tables like card, history, segmentation has to be entered. Inputs are Stock number, type number, part number, date on entered, card number, part type, quality of the item, yearweek, expiry date, manufacturer, quantity, changed by, changed date, satellite number, destination code.

4.2 DATABASE DESIGN

1. MASTER TABLE:

1. COMPONENT TABLE

Field Name	Data type	Length	Description
CTYPE	VARCHAR	15	Component type
CID	VARCHAR	3	Component id
TID	VARCHAR	2	Type id

2. SATELLITE TABLE

Field Name	Data type	Length	Description
SNAME	VARCHAR	15	Satellite name
PROJECTCMP	VARCHAR	8	Project component
SACODE	NUMBER	2	Satellite code

3. PROJECT TABLE

Field Name	Data type	Length	Description
PNAME	VARCHAR	15	Project name
PCODE	NUMBER	2	Project code

4. STOCK TABLE

Field Name	Data type	Length	Description
SNAME	VARCHAR	15	Stock name
SID	NUMBER	1	Stock id

5. LOGIN TABLE

Field Name	Data type	Length	Description
STAFF	NUMBER	5	Staff number
IN	DATE	Default	Login date
OUT	DATE	Default	Logout date
STATUS	NUMBER	1	Status of the login or logout
NODE	VARCHAR	5	Node to which the user connected

6. STAFF TABLE

Field Name	Data type	Length	Description
NO	NUMBER	5	Staff number
PWD	VARCHAR	8	Password
SCODE	NUMBER	4	Staff code
VALID	DATE	Default	Valid up to
ENTRY	DATE	Default	Entered on which date

7. CLASS TABLE

Field Name	Data type	Length	Description
CLNAME	VARCHAR	15	Class name
CLID	NUMBER	1	Class id

8. QUALITY TABLE

Field Name	Data type	Length	Description
QNAME	VARCHAR	15	Quality name
QID	VARCHAR	2	Quality id
QCODE	VARCHAR	5	Quality code
CLID	NUMBER	1	Class id

9. UNIQUE TABLE

Field Name	Data type	Length	Description
TID	VARCHAR	2	Type id
PART	VARCHAR	30	Basic part item
PID	VARCHAR	5	Part id

10. VENDOR TABLE

Field Name	Data type	Length	Description
VNAME	VARCHAR	30	Vendor name
VTYPE	NUMBER	1	Vendor type
ADDRESS	VARCHAR	40	Vendor address
COUNTRY	VARCHAR	15	Vendor from which country
FAX	VARCHAR	15	Vendor fax number
WEB	VARCHAR	20	Vendor web address
VID	NUMBER	4	Vendor id

2.TRANSACTION TABLE:

1. CARD TABLE:

Field Name	Data type	Length	Description
SID	NUMBER	1	Stock id
TID	VARCHAR	2	Type id
PART	VARCHAR	30	Part
PID	VARCHAR	5	Part id
CLID	NUMBER	1	Class id
LOC	VARCHAR	15	Location
SDATE	DATE	Default	Date on which the component is added
QUANTITY	NUMBER	7	Quantity of component

2. HISTORY TABLE:

Field Name	Data type	Length	Description
SID	NUMBER	1	Stock id
TID	VARCHAR	2	Type id
PID	VARCHAR	5	Part id
CNO	NUMBER	4	Card number
QID	VARCHAR	2	Quality id
YEARWEEK	VARCHAR	6	Year and week of component entered
FINALDATE	DATE	Default	Expiry date
MCODE	NUMBER	4	Manufacturer code

QUANTITY	NUMBER	7	Quantity
BALANCE QTY	NUMBER	7	Balance quantity
CHANGEDBY	NUMBER	5	Changed by
CHANGEDDATE	DATE	Default	Changed date
PROCESSTYPE	NUMBER	1	Processed by whom
PROCESSNO	NUMBER	1	Processed number
PROCESSDATE	DATE	Default	Processed date
CHANGEDTYPE	NUMBER	1	Changed type
DCODE	NUMBER	1	Destination code
SACODE	NUMBER	2	Satellite code

3. SEGMENTATION TABLE:

Field Name	Data type	Length	Description
TID	VARCHAR	2	Type id
PID	VARCHAR	5	Part id
CID	NUMBER	1	Component id
CNO	NUMBER	4	Card number
PART	VARCHAR	30	Part id
QID	VARCHAR	2	Quality id
YEARWEEK	VARCHAR	6	Year and week of component will expire
FINALDATE	DATE	Default	Expiry date
MCODE	NUMBER	4	Manufacturer code
QUANTITY	NUMBER	7	Quantity

4. REQUEST TABLE

Field Name	Data type	Length	Description
SID	NUMBER	1	Stock id
RNO	NUMBER	6	Request number
RDATE	DATE	Default	Request date
RTYPE	NUMBER	1	Request type
RFROM	NUMBER	1	Request from
PINFO	VARCHAR	15	Project information
RPART	VARCHAR	30	Request part
RQTY	NUMBER	6	Request quantity
RSTATUS	NUMBER	1	Request status
RBY	NUMBER	5	Requested by
RLINK	NUMBER	6	Request link
SNO	NUMBER	4	Component WithDrawal number
SDATE	DATE		Component WithDrawal date
IQTY	NUMBER	6	Issued quantity
CID	VARCHAR	2	Component id
PID	VARCHAR	5	Part id

5. PROCESS TABLE:

Field Name	Data type	Length	Description
ALINK	NUMBER	1	Advised link
ADC	VARCHAR	4	Advised dc
AEDATE	DATE	Default	Advised expiry date
AQTY	NUMBER	6	Advised qty
ANO	NUMBER	6	Advised number
AQID	VARCHAR	2	Advised quality id
ACLID	NUMBER	1	Advised class id
ACNO	NUMBER	4	Advised card number
ADATE	DATE	Default	Advised date
CWDNO	NUMBER	4	Component WithDrawal number
CWDDATE	DATE	Default	Component WithDrawal date
IDC	VARCHAR	4	Issued dc
IEDATE	DATE	Default	Issued expiry date
IMFR	NUMBER	4	Issued manufacturer
IQID	VARCHAR	2	Issued quality id
CHANGED BY	NUMBER	1	Changed by

6. CD2 TABLE

Field Name	Data type	Length	Description
SID	NUMBER	1	Stock id
SNO	NUMBER	4	Component WithDrawal number
SDATE	DATE	Default	Component WithDrawal date
SQTY	NUMBER	6	Component WithDrawal quantity
RFROM	NUMBER	1	Referred number
SBY	NUMBER	1	Screened by
RESTOCK	NUMBER	1	Returned stock
RENO	NUMBER	4	Returned number
REDC	VARCHAR	4	Returned yearweek
REDATE	DATE	Default	Returned expiry date
REMFR	NUMBER	4	Return manufacturer
REQTY	NUMBER	6	Return quantity
REFDATE	DATE	Default	Return referenced data
CDNO	NUMBER	4	Component deposit 2 number
CDDATE	DATE	Default	Component deposit 2date
CHANGEDBY	NUMBER	5	Changed by
CHANGEDDATE	DATE	Default	Changed date
CLID	NUMBER	1	Class id
QID	VARCHAR	2	Quality id
PID	VARCHAR	5	Part id
CID	VARCHAR	2	Component id

7. CD1 TABLE

Field Name	Data type	Length	Description
SREF	NUMBER	6	Stores reference number
SID	NUMBER	1	Stock id
PART	VARCHAR	30	Part
DC	VARCHAR	4	Yearweek
EDATE	DATE	Default	Expiry date
QTY	NUMBER	6	Quantity
MCODE	NUMBER	3	Manufacturer code
CDTYPE	NUMBER	1	Component deposit 1 type
CDSTATUS	NUMBER	1	Component deposit 1 status
POBY	NUMBER	6	Posted by
PODATE	DATE	Default	Posted date
CHANGEDBY	NUMBER	6	Changed by
CHANGEDDATE	DATE	Default	Changed date

4.3 OUTPUT DESIGN

Output design is also very important since the accuracy and ease of understand in the output is important. The output should be in a suitable format so that the user is fully satisfied with the report. The output should be able to convey a clear message about the status of the system.

Reports generated for Component Information System are as follows:

1. All Component wise Report.
2. Component WithDrawal Report.
3. Daily Transaction Report.
4. Transaction wise Report.
5. History Report.

All Component wise Report: This report consists of total components available for each project in the store with their Vendor name, total number of components, and their corresponding valid date.

Component WithDrawal Report: This report consists of details of components been requested by the user for their purpose.

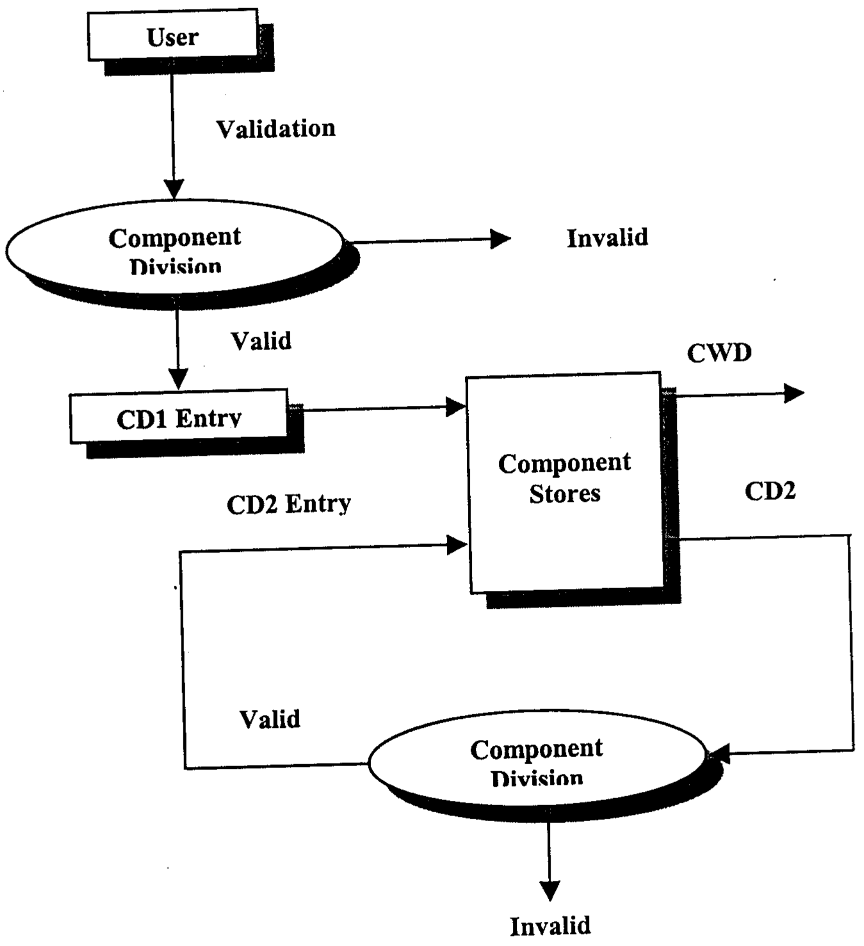
Daily Transaction Report: This report records the specific transactions of issuing, withdrawal and deposit of components in the store daily.

Transaction wise Report: This report records the transactions of issuing, withdrawal and deposit of components in the store for a time period.

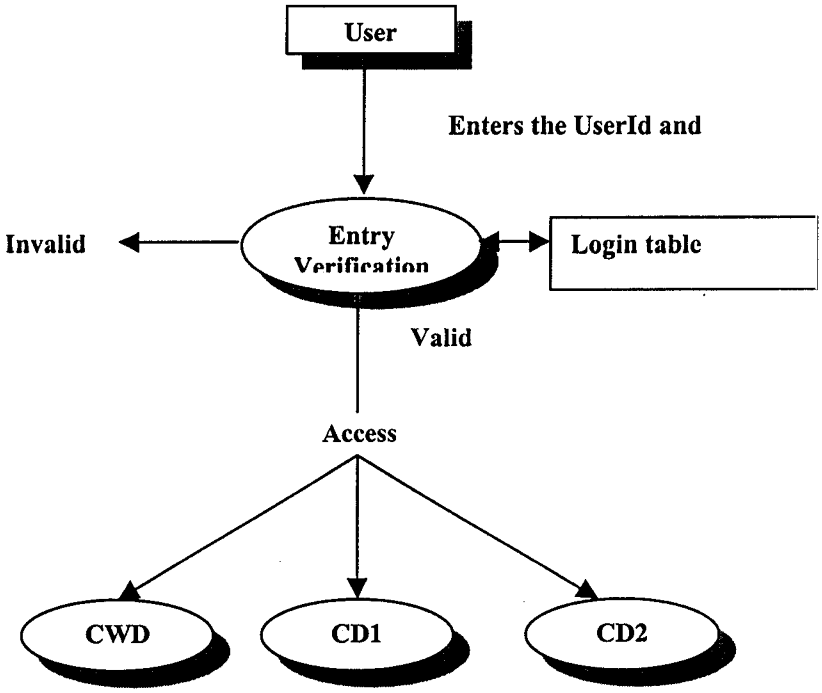
History Report: This report records all transactions made for inventory like issuing, depositing and withdrawing of various components from the store.

4.4 PROCESS DESIGN

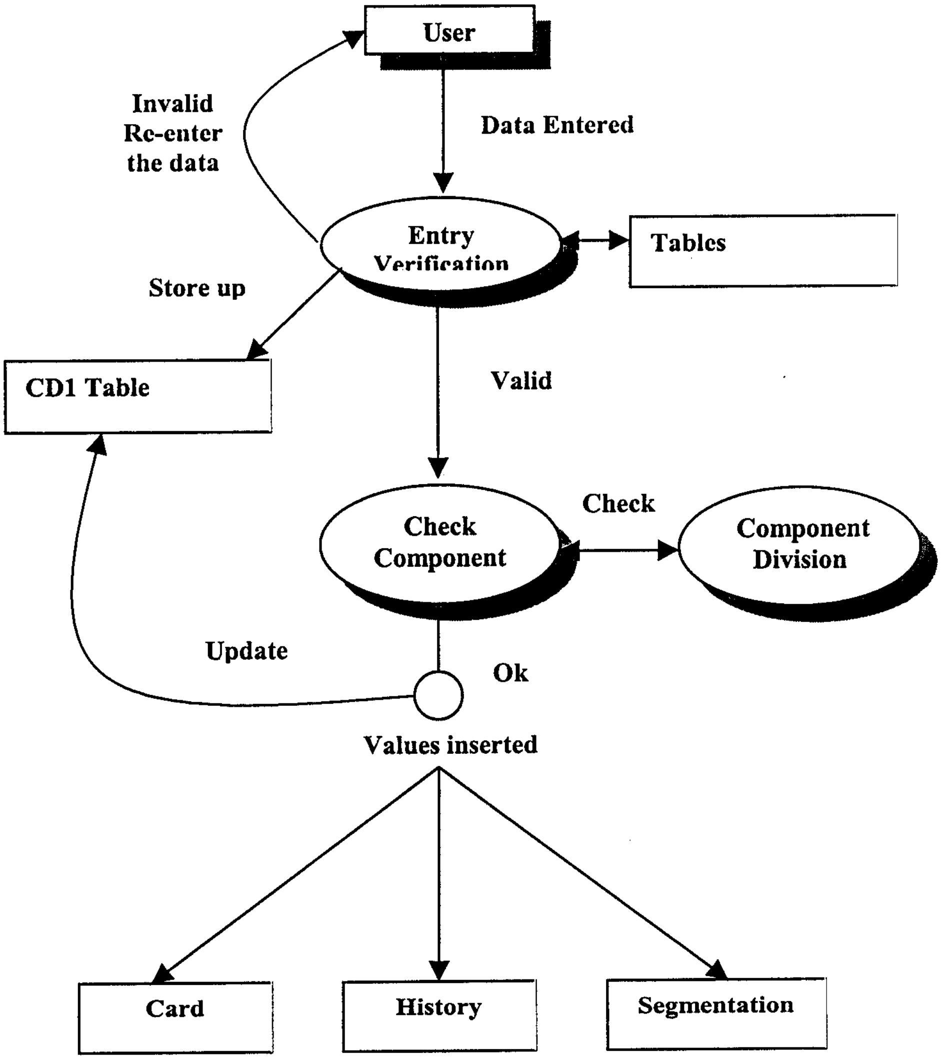
Data Flow Diagram: General Diagram



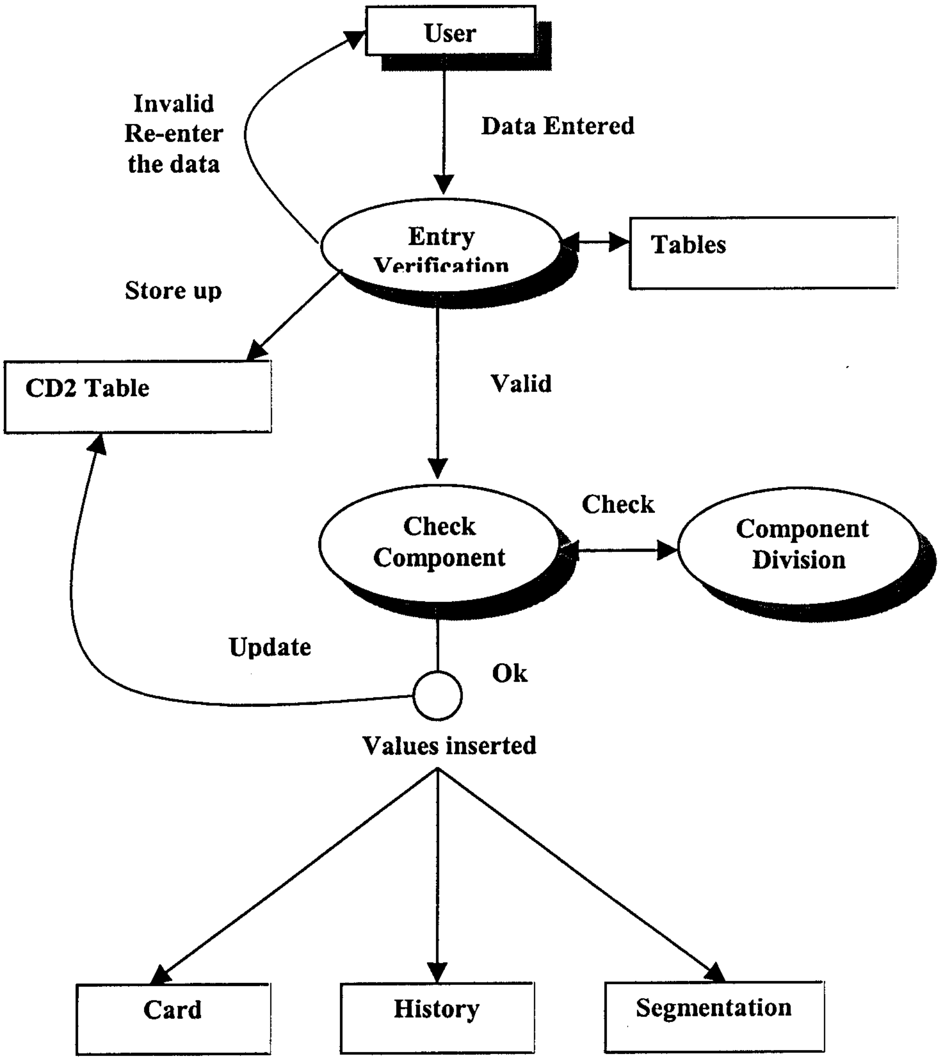
Data Flow Diagram: User Password



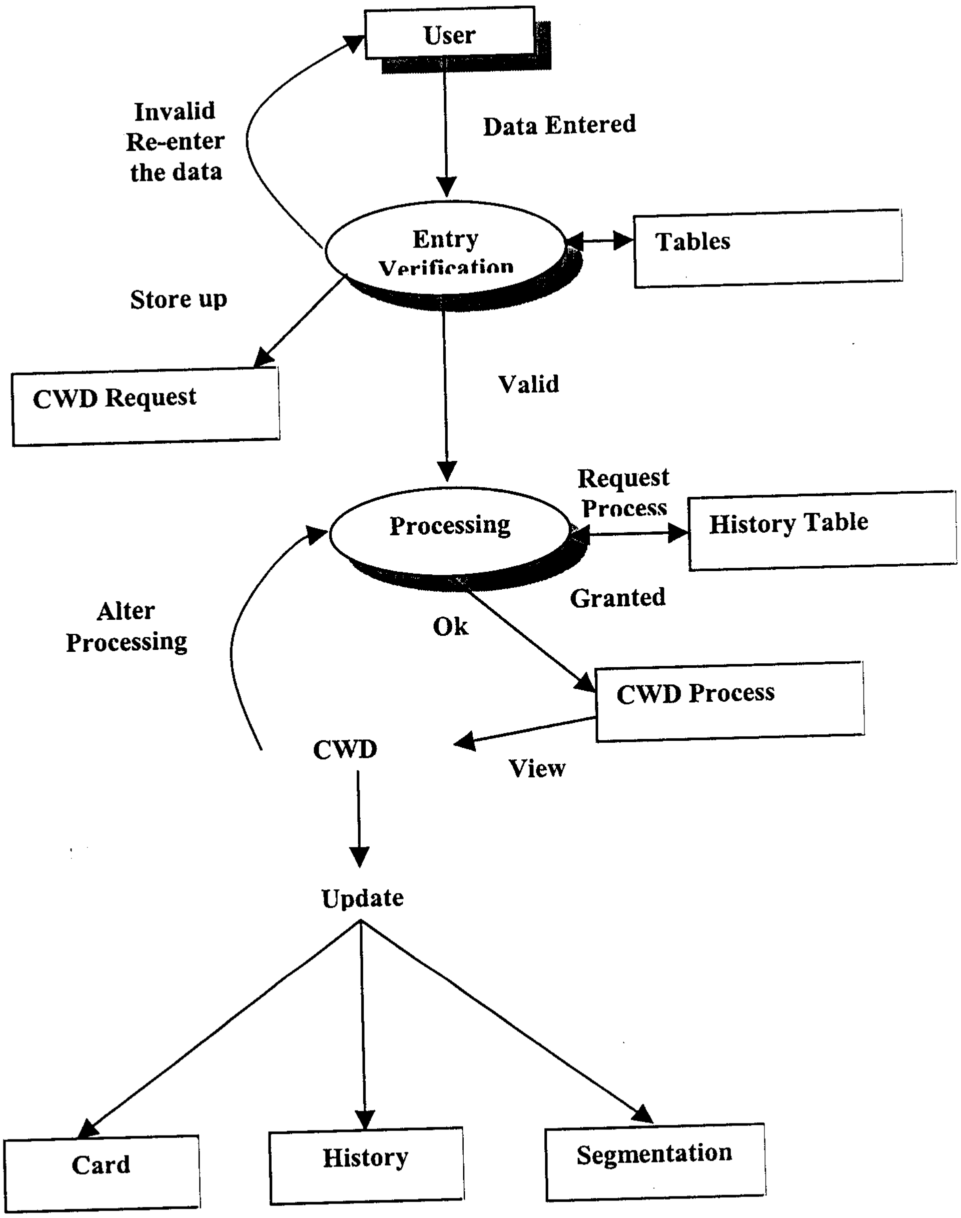
Data Flow Diagram: Component Deposit 1 (CD1)



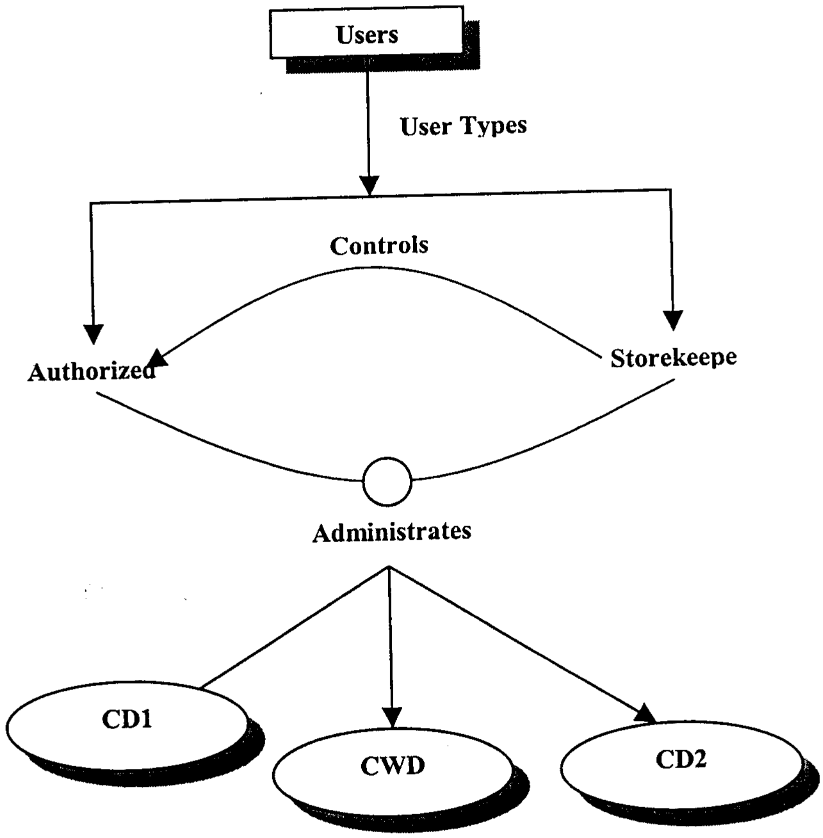
Data Flow Diagram: Component Deposit 2 (CD2)



Data Flow Diagram: Component With Drawal (CWD)



Data Flow Diagram: Types of Users



SYSTEM IMPLEMENTATION & TESTING

V. SYSTEM IMPLEMENTATION AND TESTING

5.1 SYSTEM IMPLEMENTATION

Implementation is the stage of the project when the theoretical design is turned into a working system. At this stage the main workload and the major impact on existing practices shifts to the user department. If the implementation stage is carefully planned and controlled, it can cause chaos.

It involves careful planning, investigation of the current system and constraints and also it should satisfy the careful space to be worked on the system. The project has to be noted on which platform it will work on the system. The project has to be noted on which platform it will work and how for it can occupy i.e. the capacity on working stage.

According to the Component Information System is basically an Intranet package and is to be utilized by the ISAC. The project, which is been developed with a Window NT operating system. So it is a platform dependent so each system, which loads the project, should be a Windows NT workstation with version 4.0. The software, which is used to build, is Visual Basic.

The program developed with this visual basic is made an exe and is implemented on Intranet, so that the system, which is connected with this Windows NT, can work on this project.

According to the ODBC connectivity it is mainly depended on the server because the server, which is important and handles entire data's in a database a tables. The backend system is built in Oracle and which is been implemented with its own database and data's are been feed in to it.

Actually implementation starts with server of all the tables implemented in Oracle and making an OBBC connectivity and implementing in Visual Basic in a Windows NT platform the project does the function.

5.2 SYSTEM TESTING

5.2.1 TESTING PROCESS

Except for small software, system should not be tested as a single, monolithic unit. Large systems are built out of sub-systems, which are built out of sub-systems, which are composed of procedures and functions. The testing process should therefore proceed in stages where testing is carried out incrementally in conjunction with system implementation.

There are 5 test stages and as defects are discovered at any stage, they require program modifications to correct them and this may require other stages in the testing process to be repeated. The process therefore is an interactive one with information being fed back from later stages to earlier parts of the process. The stages in the testing process are:

5.2.1.1 UNIT TESTING

Individual components are tested to ensure that they operate correctly. Each component is tested independently without other system components. With respect to this project the functions that compute various values are tested and results are verified.

5.2.1.2 MODULE TESTING

A module is a collection of dependent components such as an object classes an abstract datatype or some looser collection of procedures and functions. A module encapsulated related components so it can be tested with other system components. Group of computational function for module and report presentation functions from another module. These modules are tested in this stage.

5.2.1.3 SUB-SYSTEM TESTING

This phase involves testing collection of modules, which have been integrated into sub-system. Sub-system may be independently designed and implemented. The most common problems that arise in large software system are sub-system interface miss-matches. The sub-system process should therefore concentrate on the detection of interface errors by rigorously excessing these interfaces. Both the modules data processing and report preparation of Component Information System are treated a sub-system and tested in this stage.

5.2.1.4 SYSTEM TESTING

The sub-system is integrated to make up the entire system. The testing process is concerned with finding errors, which result from unanticipated interactions between sub-system and system components. It is also concerned with validating that the system meets its functional requirements. After integration of the above sub-system with the whole of Component Information System, the entire system is tested for errors.

5.2.1.5 ACCEPTANCE TESTING

This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system procedure rather than simulated data. Acceptance testing may reveal errors and omissions in the system requirements definition because the real data exercises the system in different way from the test data. Acceptance testing may also reveal requirement problems where the system's facilities do not really meet the users need for the system performance is unacceptable. Test and reveal data were provided to the system and checked for errors.

5.2.2 DEFECT TESTING

Defect testing is intended to exercise a system so those latent defects are exposed before the system is delivered. This contrasts with validation testing which is intended to demonstrate that a system meets its requirements. Validation testing requires system to perform correctly using given acceptance test cases. A successful defect test is a test, which causes the system to perform incorrectly and hence express the defect. It demonstrates the presence, not the absence of program faults. Various values with in the limit and exceeding the limit were provided repeatedly to individual components of processing and presentation modules. These brought out the defects in the system and were corrected.

5.2.2.1 BLACK BOX TESTING

It relies on the specification of the system or component, which is being tested to derive test cases. The system is black box whose behavior can only be determined by studying its inputs and the related outputs. This is also called as functional testing because mathematical functions can be specified using only their inputs and outputs.

Following black box methods were applied to the modules to test some functions.

- Usage of only one function of entire project. This proved that the program works.
- Usage of several functions through out the project helps us to produce a correct output.

5.2.2.2 STRUCTURAL TESTING

This is a complementary approach to black box testing and is sometimes called structural, white box or glass box testing. The tester can analyse the code and use knowledge about the structure of a component to derive test data. The advantage of structural testing is that an analysis of the code can be used to find how many test cases are needed to guarantee a given level of test coverage. A dynamic analyser can then be used to measure the extent of this coverage and help with test case design.

Path testing:

Path testing is a white box testing strategy whose objective is to exercise every independent execution path through the component. If every independent path is executed than all the statements in the program must have been executed at least once. Further all condition statements are tested for both true and false cases and unwanted looping statements. This helped to improve the program efficiently with respect to time complexity and memory usage.

5.2.2.3 INTERFACE TESTING

Interface testing takes place when modules or sub-systems are integrated to create larger systems. Each module or sub-system has a defined interface, which is called by other program components. The objective of this testing is to detect faults which may have been introduced into the system because of interface errors or invalid assumptions about interfaces.

There are different types of interface between program components and consequently different types of interface error can occur.

Parameter interfaces: These are interfaces where data or sometimes function references are passed from one component to another.

Shared memory interfaces: These are interfaces where block of memory is shared between sub-systems and retrieved from there by other sub-system.

Procedural interfaces: These are interfaces where one sub-system encapsulates a set of procedures, which can be called by other sub-system.

Message passing interfaces: These are interfaces where one sub-system requests a service from another sub-system by passing a message to it. A return message includes the results of executing the service.

CONCLUSION & SCOPE FOR FUTURE DEVELOPMENT

VI. CONCLUSION AND SCOPE FOR THE FUTURE DEVELOPMENT

This project was done keeping in mind the fact that it should follow all the steps of the software engineering process and covers the complete software development in life cycle. The user interface provided by this project is widely accepted by the users in general. This project has left the scope of further enhancements wide open as it was developed in an environment that supports Intranet widely that in use of the Visual Basic programming language and the effective RDBMS like Oracle which would support a large size of data storage ensuring security and accuracy. A complete documentation that is provided makes the changes and enhancements that are to be done very easy and provides the vitality of documentation. The preference of Visual Basic to any other GUI is also justified since any further enhancement of this project would mean that more of platform independent extensions that are fore seen with this project.

According to the scope for the future, even the system meets all the user requirements, new features can added and some possible improvements can be incorporated in GUI based data acquisition and Component Information System for the better user interface.

Some possible improvements are:

The project is been developed in Intranet, which is of platform dependent.

Project can be made platform independent by using the latest technologies can enhance this project.

BIBLIOGRAPHY

VII. BIBLIOGRAPHY

1. Lee

   System Analysis And Design

   Galgotia Book Source, 1980.

2. Pressman .S. Roger

   Software Engineering – A Practitioners Approach

   Fourth Edition, McGraw – Hill Book Company, 1990.

3. Microsoft

   Microsoft Visual Basic 6.0 programmers guide.

4. Steven Hoizner

   Visual Basic 6.0 Programming

   Comdex Publishing Company Limited 1st edition.

5. Scott Urman

   Oracle 8 PL/SQL – Programming

   Tata McGraw – Hill Publishers edition 1999.

6. George Koch, Kevin Loney

   Oracle Complete Reference

   Tata McGraw – Hill Publishers 3rd edition.

7. Bipin C.Desai

   Database Management System.

   Galgotia Publishers Company Limited 1998.

Web Sites:

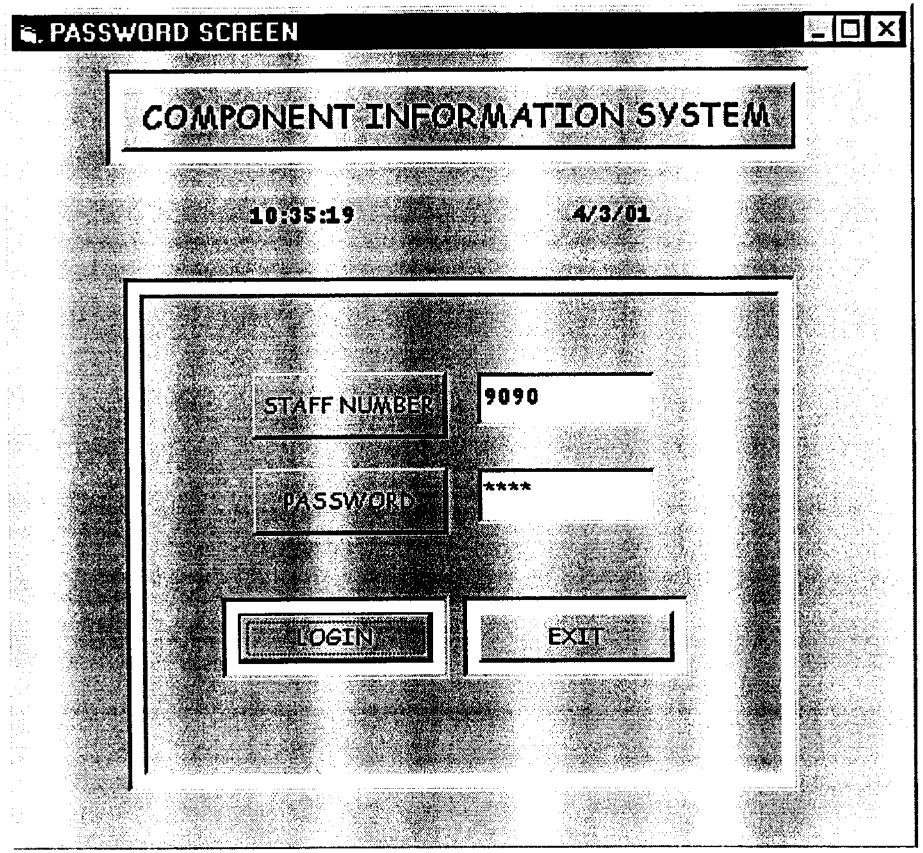
www.codeguru.com

www.microsoft.com/msdn

Other References:

MSDN Library October 99 edition – Microsoft corporation.

APPENDICES



PASSWORD SCREEN

COMPONENT INFORMATION SYSTEM

10:35:19

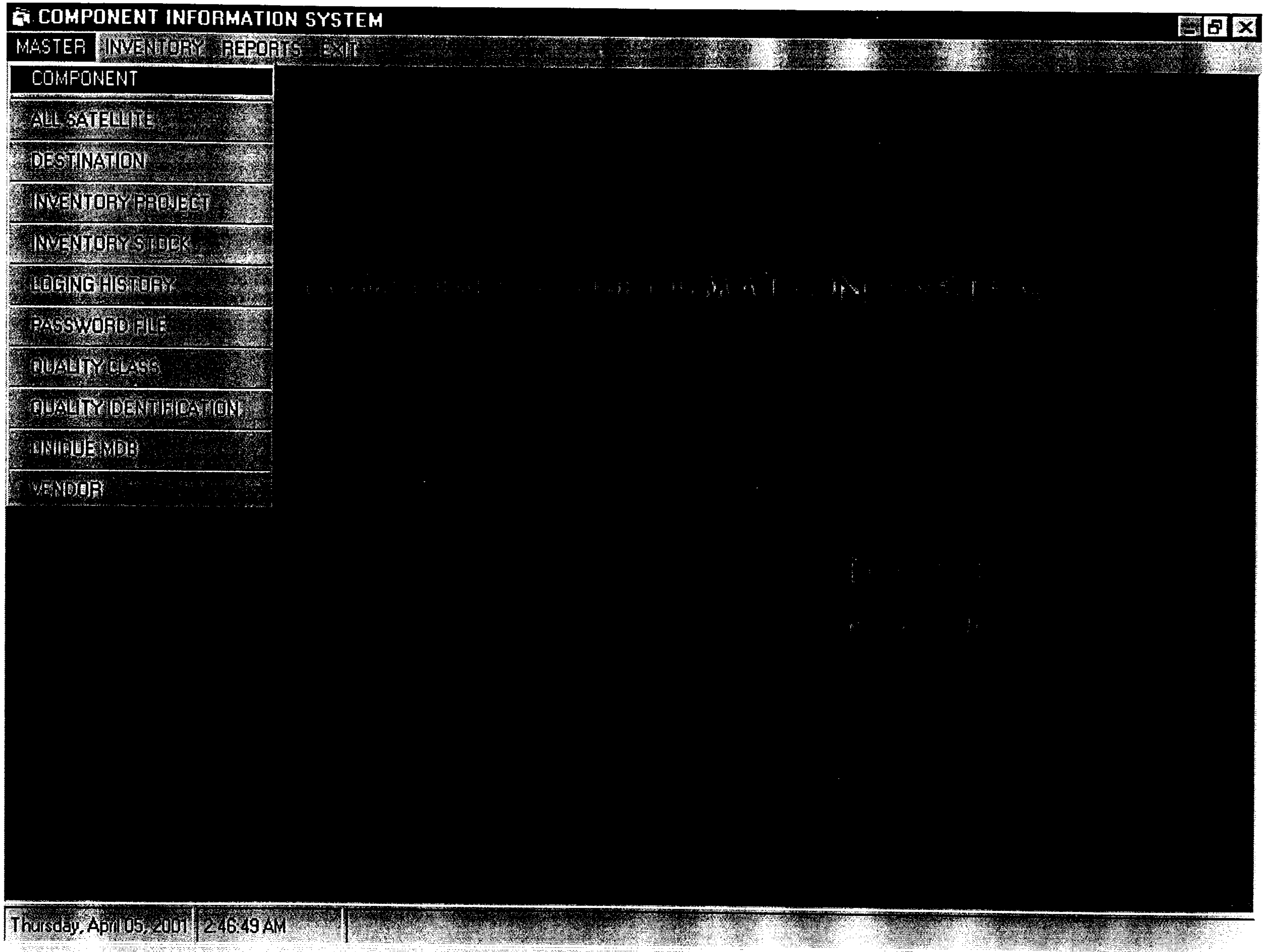
4/3/01

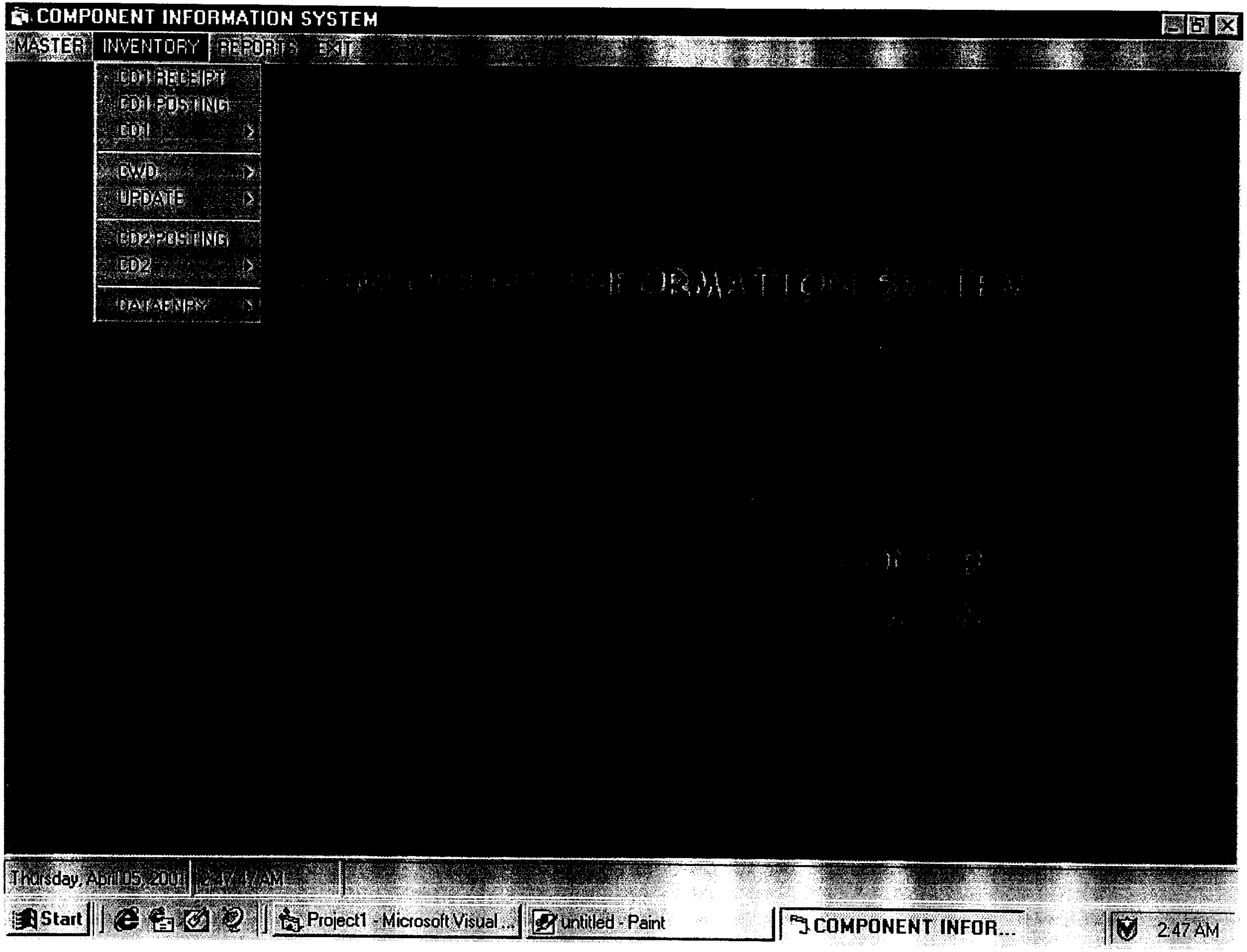
STAFF NUMBER 9090

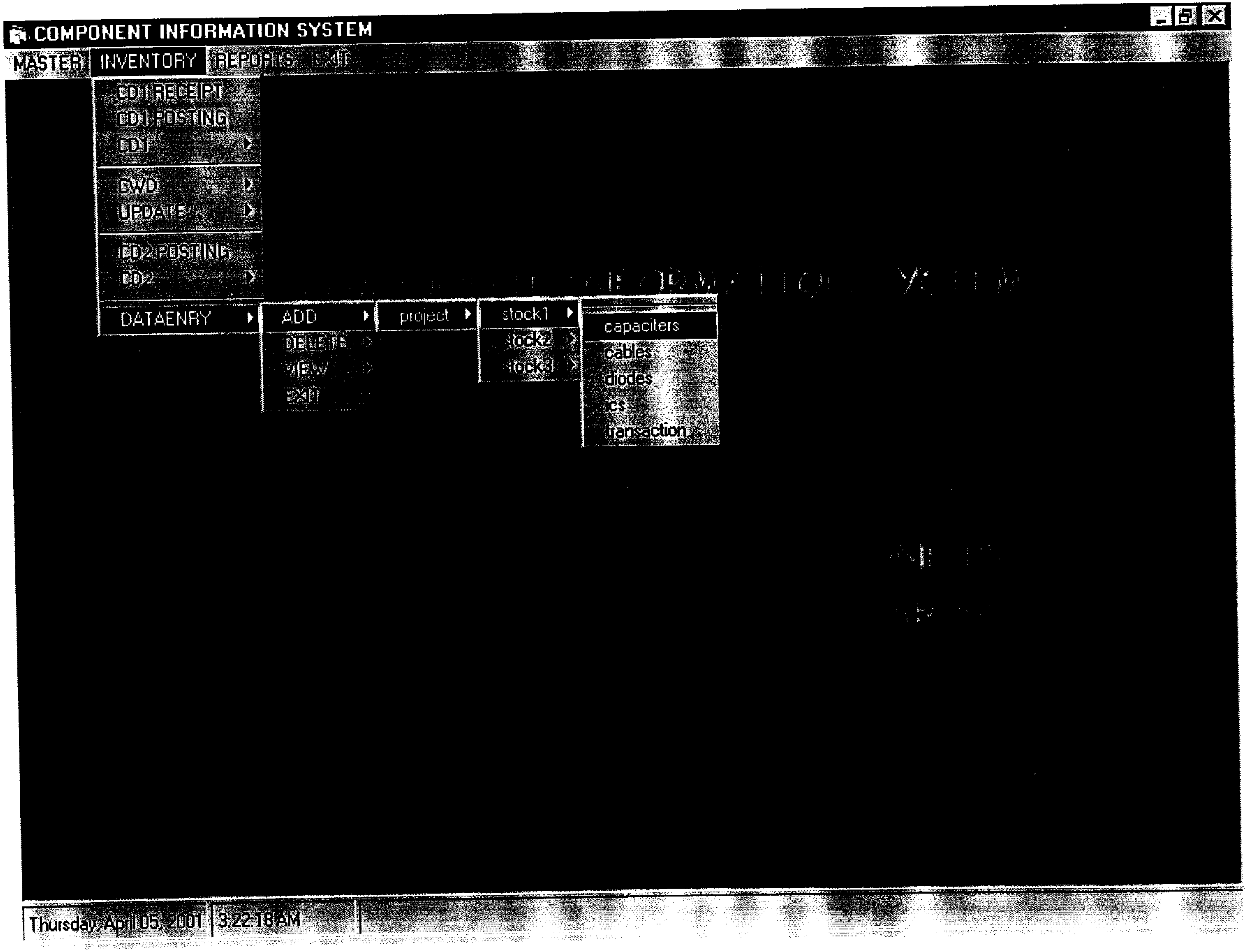
PASSWORD ****

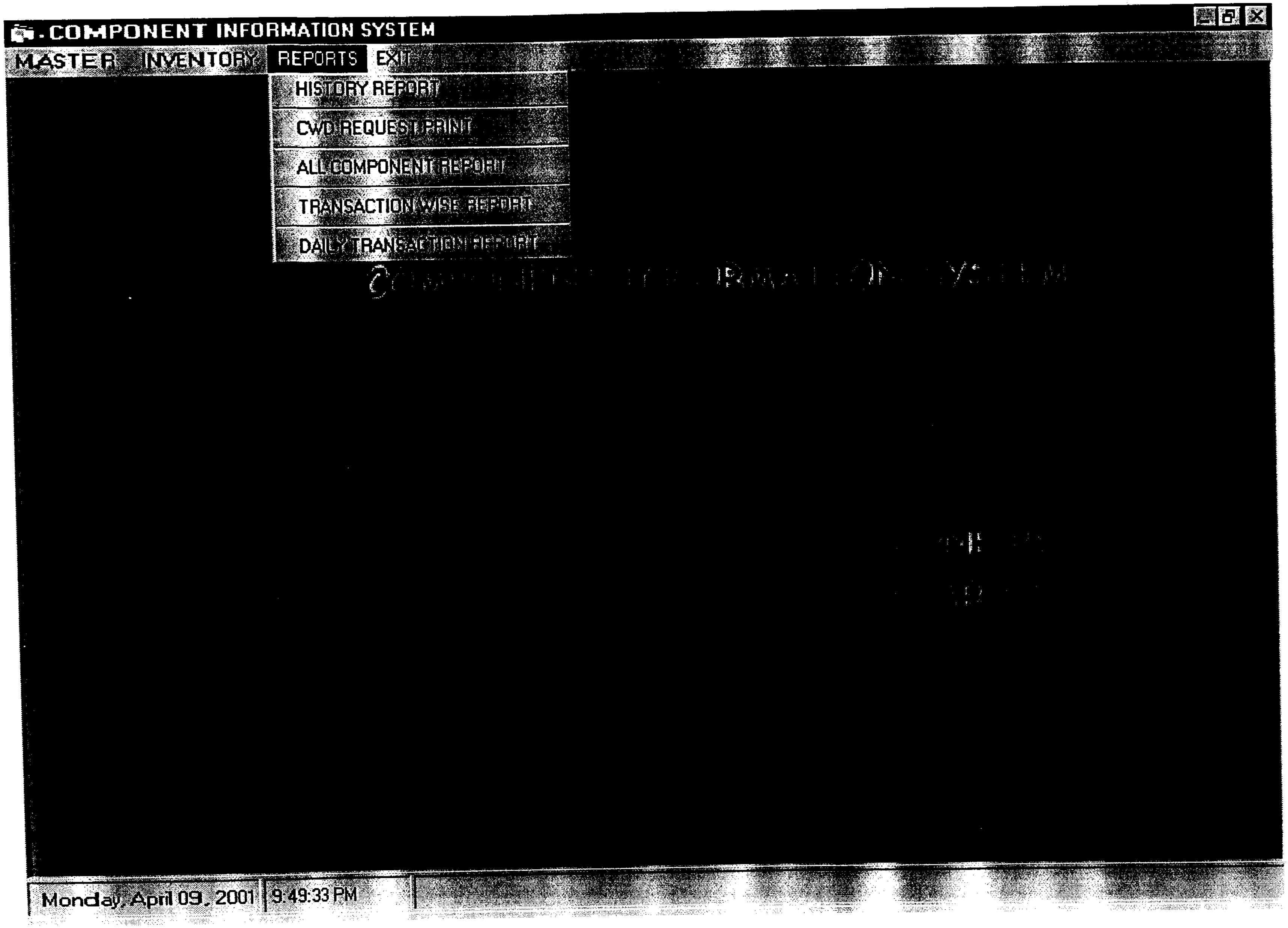
COMPONENT INFORMATION SYSTEM
MASTER
INVENTORY
REPORTS
EXIT
COMPONENT INFORMATION SYSTEM
Start
Project1 - Microsoft Visual ...
untitled - Paint
COMPONENT INFOR...
2:48 AM

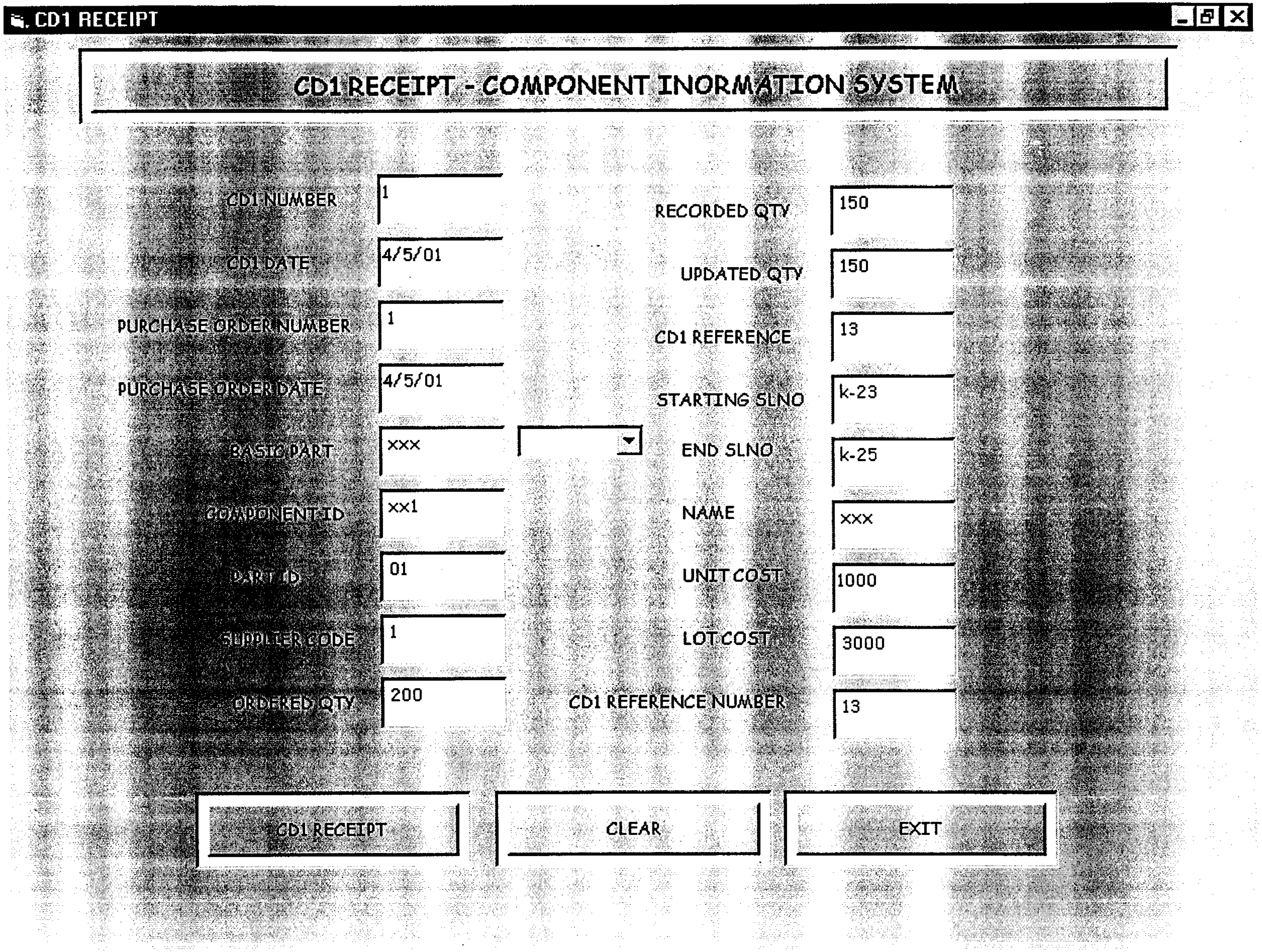
Thursday, April 05, 2001 2:48:06 AM


COMPONENT INFORMATION SYSTEM
MASTER
INVENTORY
REPORTS
EXIT
COMPONENT
ALL SATELLITE
DESTINATION
INVENTORY PROJECT
INVENTORY STOCK
LOGING HISTORY
PASSWORD FILE
QUALITY CLASS
QUALITY IDENTIFICATION
UNIQUE MDB
VENDOR
Thursday, April 05, 2001
2:46:49 AM


COMPONENT INFORMATION SYSTEM
MASTER
INVENTORY
REPORTS
EXIT
CD1 RECEIPT
CD1 POSTING
CD1
CWD
UPDATE
CD2 POSTING
CD2
DATAENRY
Thursday, April 05, 2001
2:47:47 AM
Start
Project1 - Microsoft Visual ...
untitled - Paint
COMPONENT INFOR...
2:47 AM

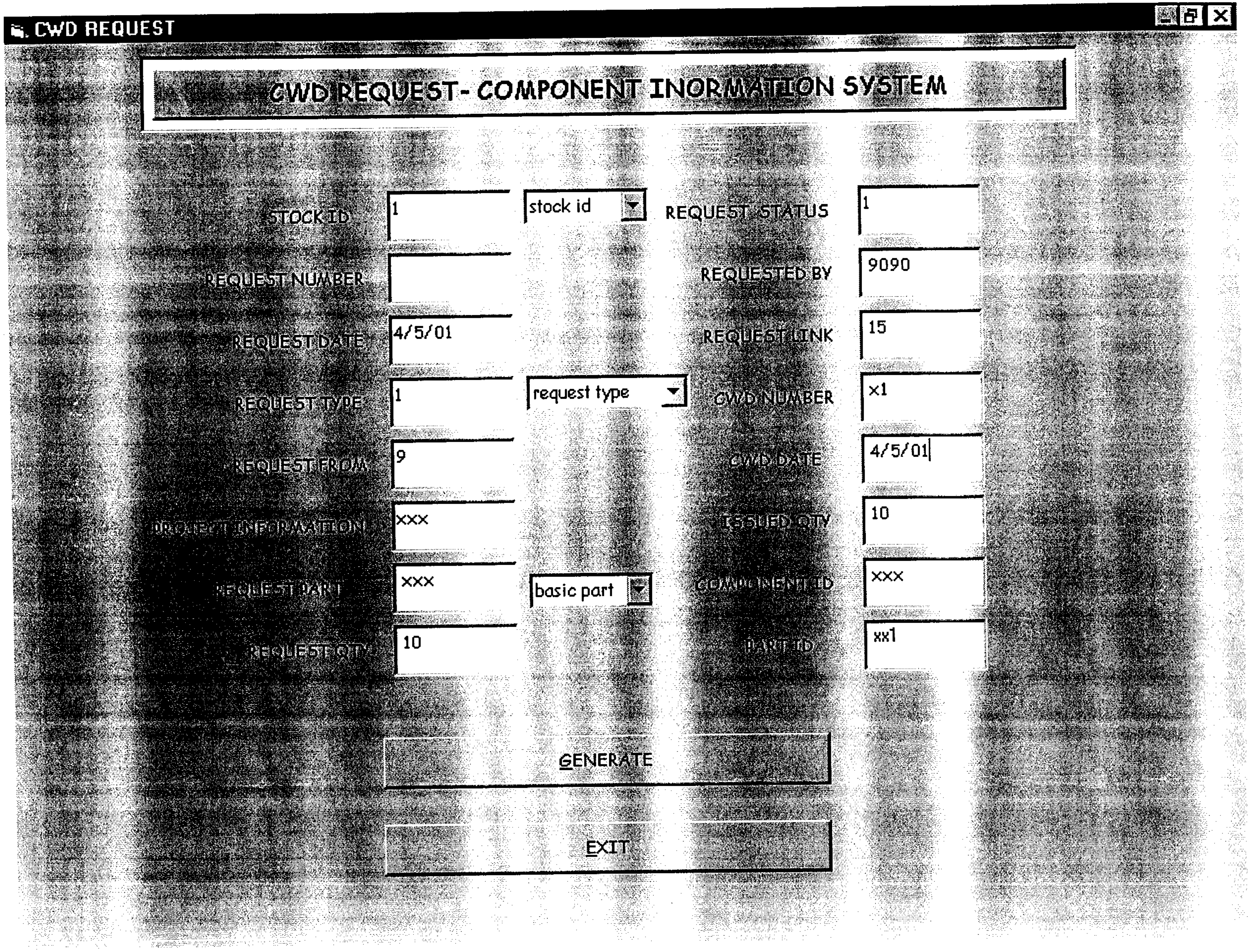

COMPONENT INFORMATION SYSTEM
MASTER
INVENTORY
REPORTS
EXIT
CD1 RECEIPT
CD1 POSTING
CD1
CWD
UPDATE
CD2 POSTING
CD2
DATAENRY
ADD
DELETE
VIEW
EXIT
project
stock1
stock2
stock3
capaciters
cables
diodes
ics
transaction
Thursday, April 05, 2001
3:22:18 AM


COMPONENT INFORMATION SYSTEM
MASTER
INVENTORY
REPORTS
EXIT
HISTORY REPORT
CWD REQUEST PRINT
ALL COMPONENT REPORT
TRANSACTION WISE REPORT
DAILY TRANSACTION REPORT
Monday, April 09, 2001
9:49:33 PM



CD1 RECEIPT - COMPONENT INORMATION SYSTEM

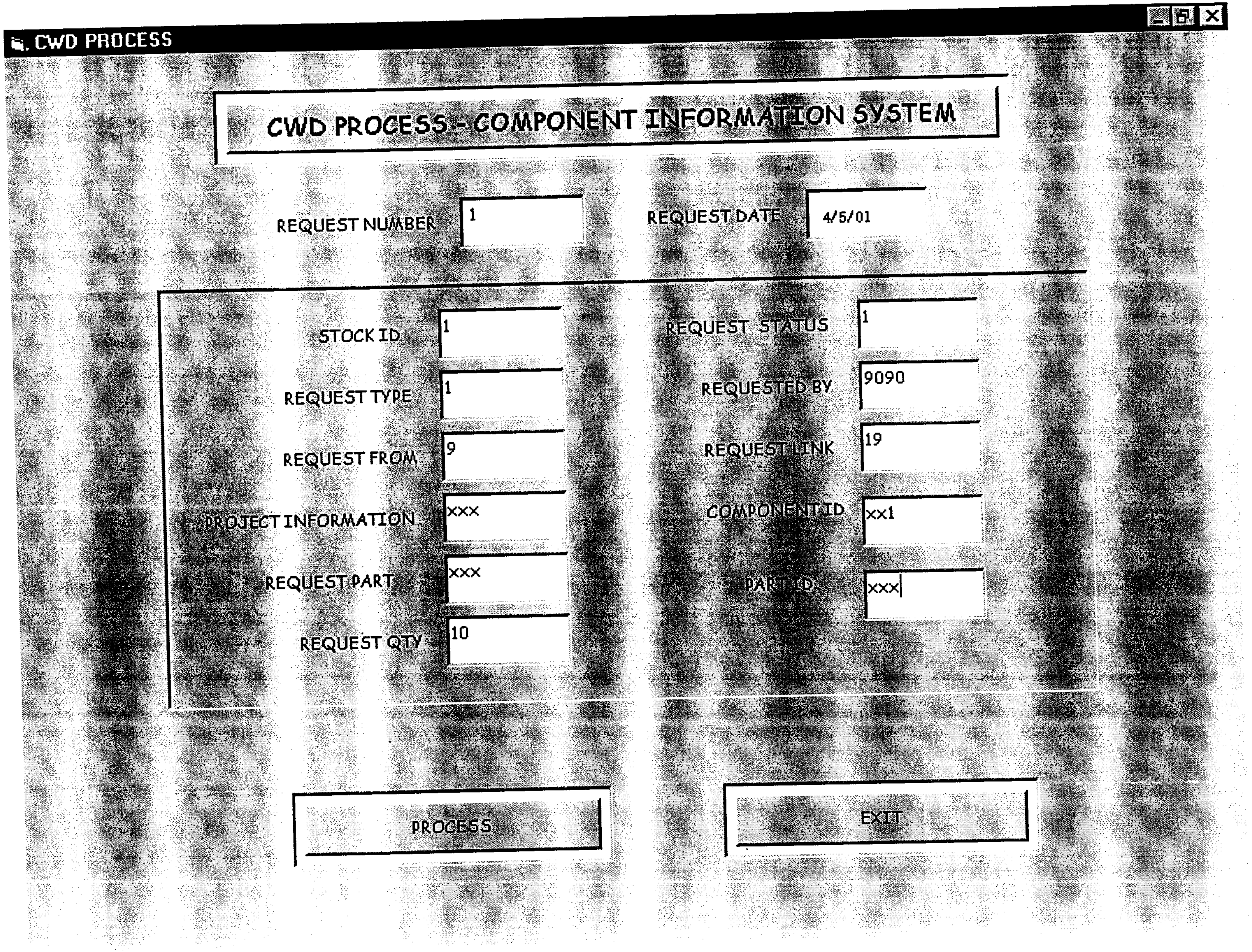
CD1 NUMBER	1	RECORDED QTY	150
CD1 DATE	4/5/01	UPDATED QTY	150
PURCHASE ORDER NUMBER	1	CD1 REFERENCE	13
PURCHASE ORDER DATE	4/5/01	STARTING SLNO	k-23
BASIC PART	xxx	END SLNO	k-25
COMPONENT ID	xx1	NAME	xxx
PART ID	01	UNIT COST	1000
SUPPLIER CODE	1	LOT COST	3000
ORDERED QTY	200	CD1 REFERENCE NUMBER	13



CWD REQUEST

CWD REQUEST- COMPONENT INORMATION SYSTEM

STOCK ID	1	stock id ▼	REQUEST STATUS	1
REQUEST NUMBER			REQUESTED BY	9090
REQUEST DATE	4/5/01		REQUEST LINK	15
REQUEST TYPE	1	request type ▼	CWD NUMBER	x1
REQUEST FROM	9		CWD DATE	4/5/01
PROJECT INFORMATION	xxx		ISSUED QTY	10
REQUEST PART	xxx	basic part ▼	COMPONENT ID	xxx
REQUEST QTY	10		PART ID	xx1



CWD PROCESS - COMPONENT INFORMATION SYSTEM

REQUEST NUMBER 1

REQUEST DATE 4/5/01

STOCK ID 1

REQUEST TYPE 1

REQUEST FROM 9

PROJECT INFORMATION XXX

REQUEST PART XXX

REQUEST QTY 10

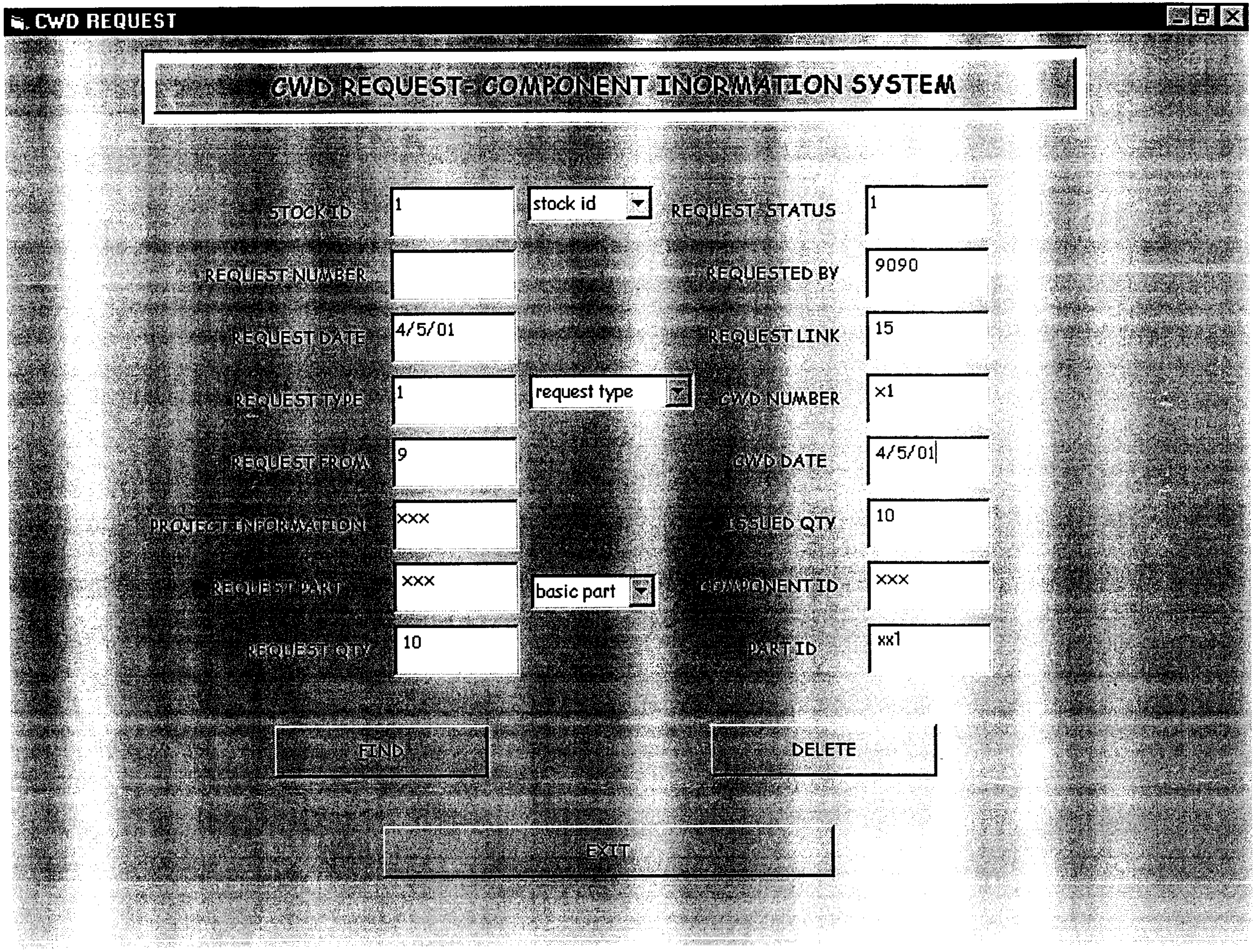
REQUEST STATUS 1

REQUESTED BY 9090

REQUEST LINK 19

COMPONENT ID xx1

PART ID xxx

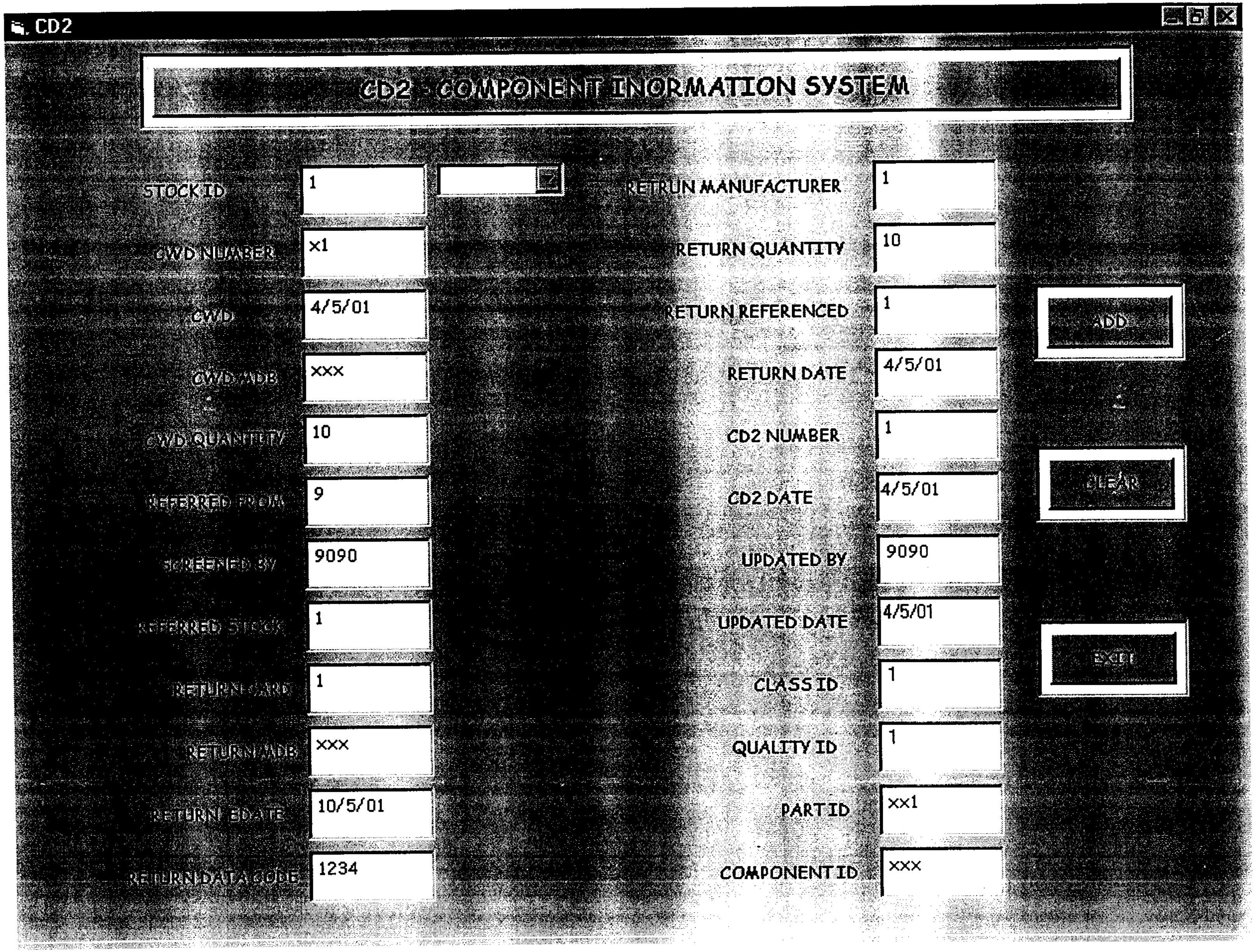


CWD REQUEST

CWD REQUEST- COMPONENT INORMATION SYSTEM

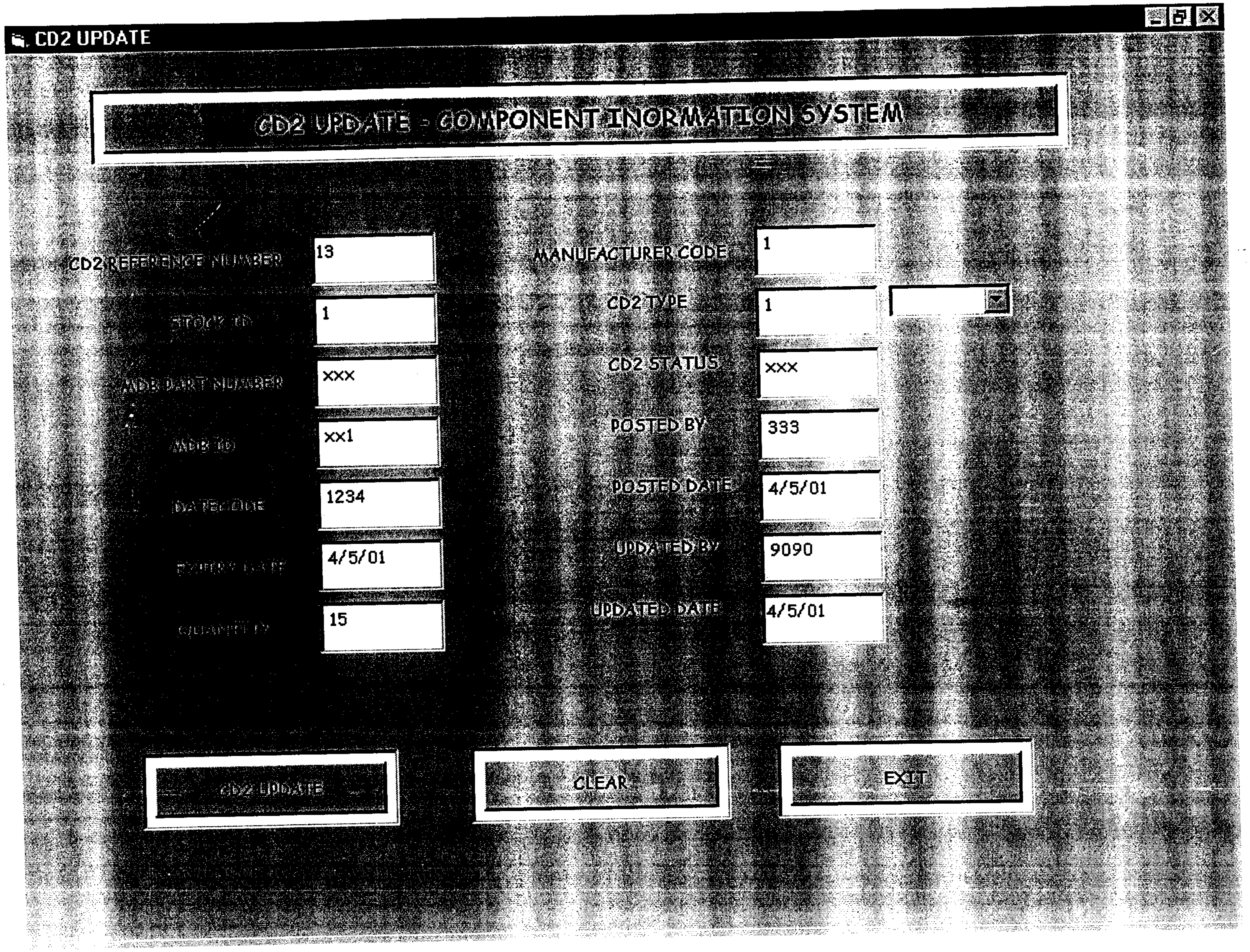
STOCK ID: 1 | stock id ▾
REQUEST NUMBER:
REQUEST DATE: 4/5/01
REQUEST TYPE: 1 | request type ▾
REQUEST FROM: 9
PROJECT INFORMATION: xxx
REQUEST PART: xxx | basic part ▾
REQUEST QTY: 10

REQUEST STATUS: 1
REQUESTED BY: 9090
REQUEST LINK: 15
CWD NUMBER: x1
CWD DATE: 4/5/01
ISSUED QTY: 10
COMPONENT ID: xxx
PART ID: xx1



CD2 - COMPONENT INORMATION SYSTEM

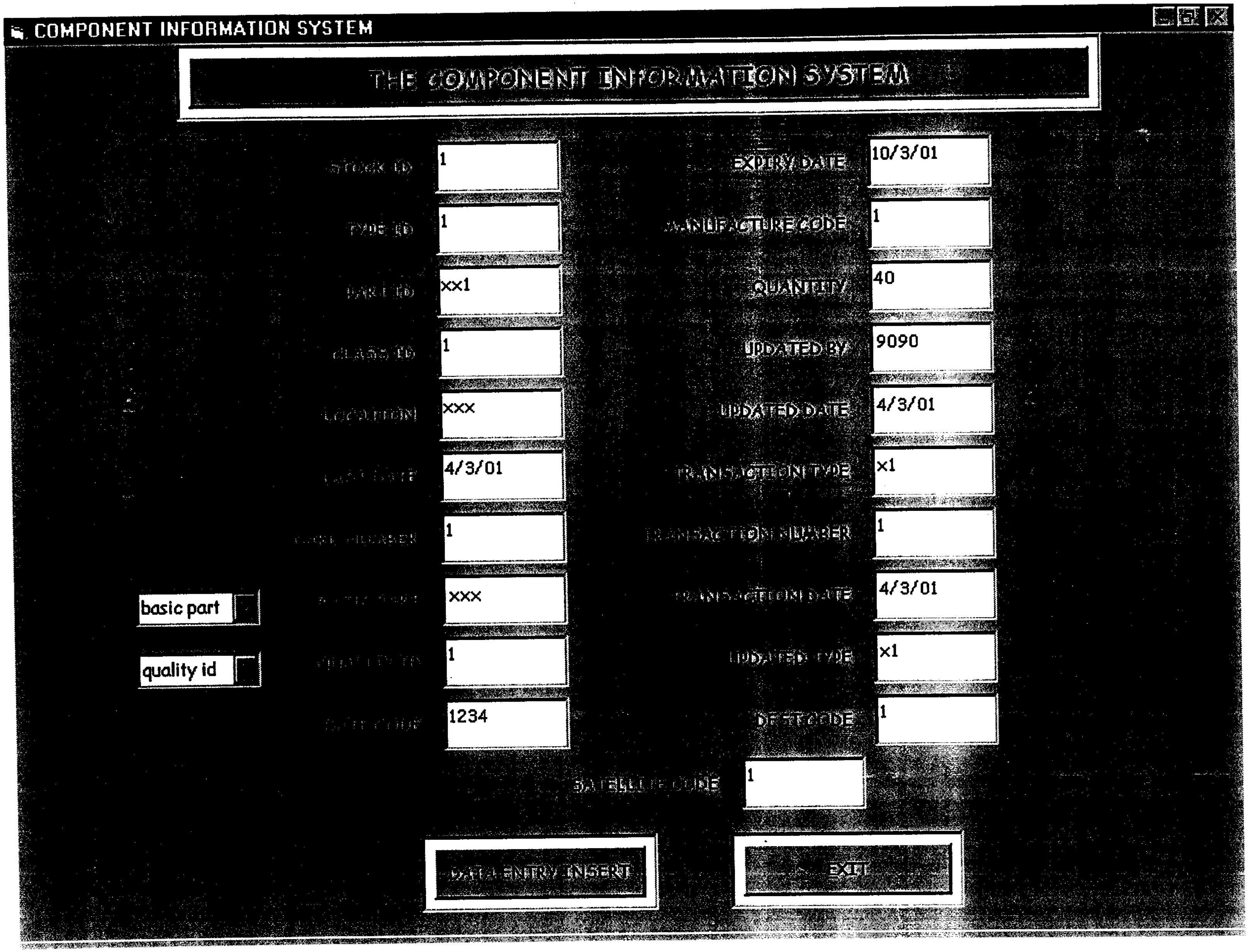
STOCK ID	1	RETRUN MANUFACTURER	1
CWD NUMBER	x1	RETURN QUANTITY	10
CWD	4/5/01	RETURN REFERENCED	1
CWD ADB	xxx	RETURN DATE	4/5/01
CWD QUANTITY	10	CD2 NUMBER	1
REFERRED FROM	9	CD2 DATE	4/5/01
SCREENED BY	9090	UPDATED BY	9090
REFERRED STOCK	1	UPDATED DATE	4/5/01
RETURN CARD	1	CLASS ID	1
RETURN ADB	xxx	QUALITY ID	1
RETURN EDATE	10/5/01	PART ID	xx1
RETURN DATA CODE	1234	COMPONENT ID	xxx



CD2 UPDATE

CD2 UPDATE - COMPONENT INORMATION SYSTEM

CD2 REFERENCE NUMBER	13	MANUFACTURER CODE	1
STOCK ID	1	CD2 TYPE	1
MDB PART NUMBER	xxx	CD2 STATUS	xxx
MDB ID	xx1	POSTED BY	333
DATE CODE	1234	POSTED DATE	4/5/01
EXPIRY DATE	4/5/01	UPDATED BY	9090
QUANTITY	15	UPDATED DATE	4/5/01



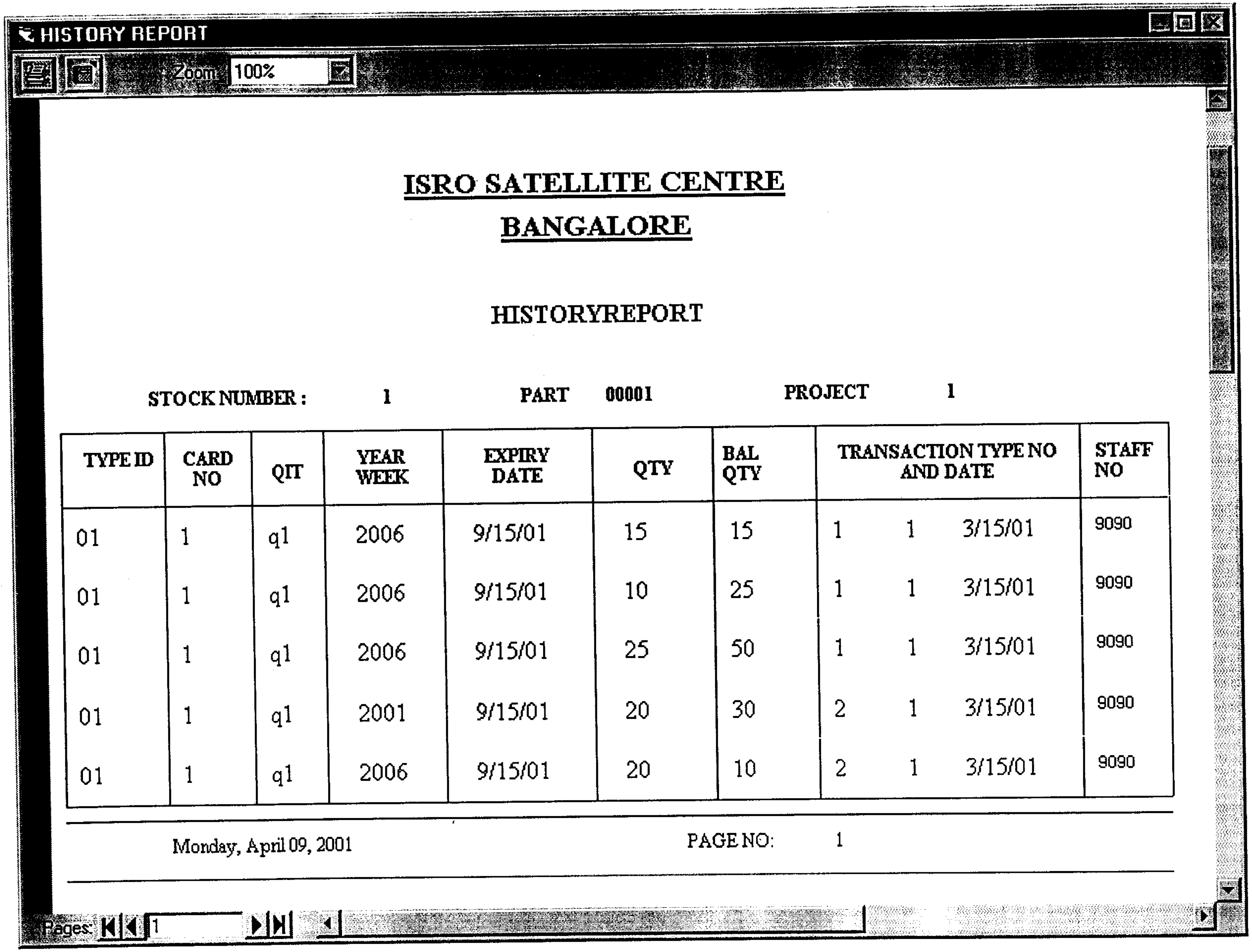
THE COMPONENT INFORMATION SYSTEM

STOCK ID	1	EXPIRY DATE	10/3/01
TYPE ID	1	MANUFACTURE CODE	1
PART ID	xx1	QUANTITY	40
CLASS ID	1	UPDATED BY	9090
LOCATION	xxx	UPDATED DATE	4/3/01
LAST DATE	4/3/01	TRANSACTION TYPE	x1
[illegible] NUMBER	1	TRANSACTION NUMBER	1
[illegible] PART	xxx	TRANSACTION DATE	4/3/01
[illegible] ID	1	UPDATED TYPE	x1
[illegible] CODE	1234	DEFECT CODE	1

basic part

quality id

SATELLITE CODE 1



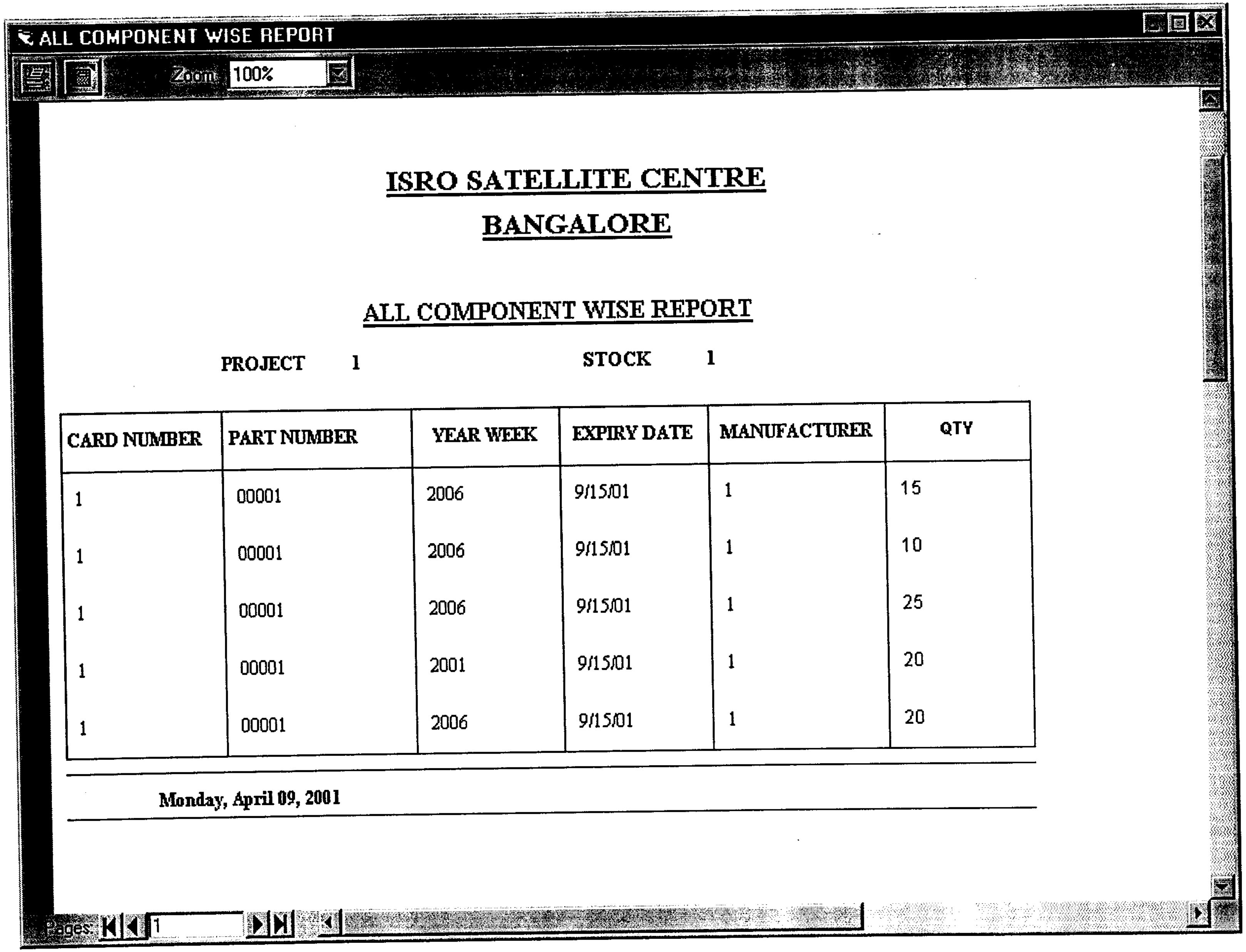
ISRO SATELLITE CENTRE
BANGALORE

HISTORYREPORT

STOCK NUMBER : 1 PART 00001 PROJECT 1

TYPE ID	CARD NO	QIT	YEAR WEEK	EXPIRY DATE	QTY	BAL QTY	TRANSACTION TYPE NO AND DATE	STAFF NO
01	1	q1	2006	9/15/01	15	15	1 1 3/15/01	9090
01	1	q1	2006	9/15/01	10	25	1 1 3/15/01	9090
01	1	q1	2006	9/15/01	25	50	1 1 3/15/01	9090
01	1	q1	2001	9/15/01	20	30	2 1 3/15/01	9090
01	1	q1	2006	9/15/01	20	10	2 1 3/15/01	9090

Monday, April 09, 2001 PAGE NO: 1



ISRO SATELLITE CENTRE

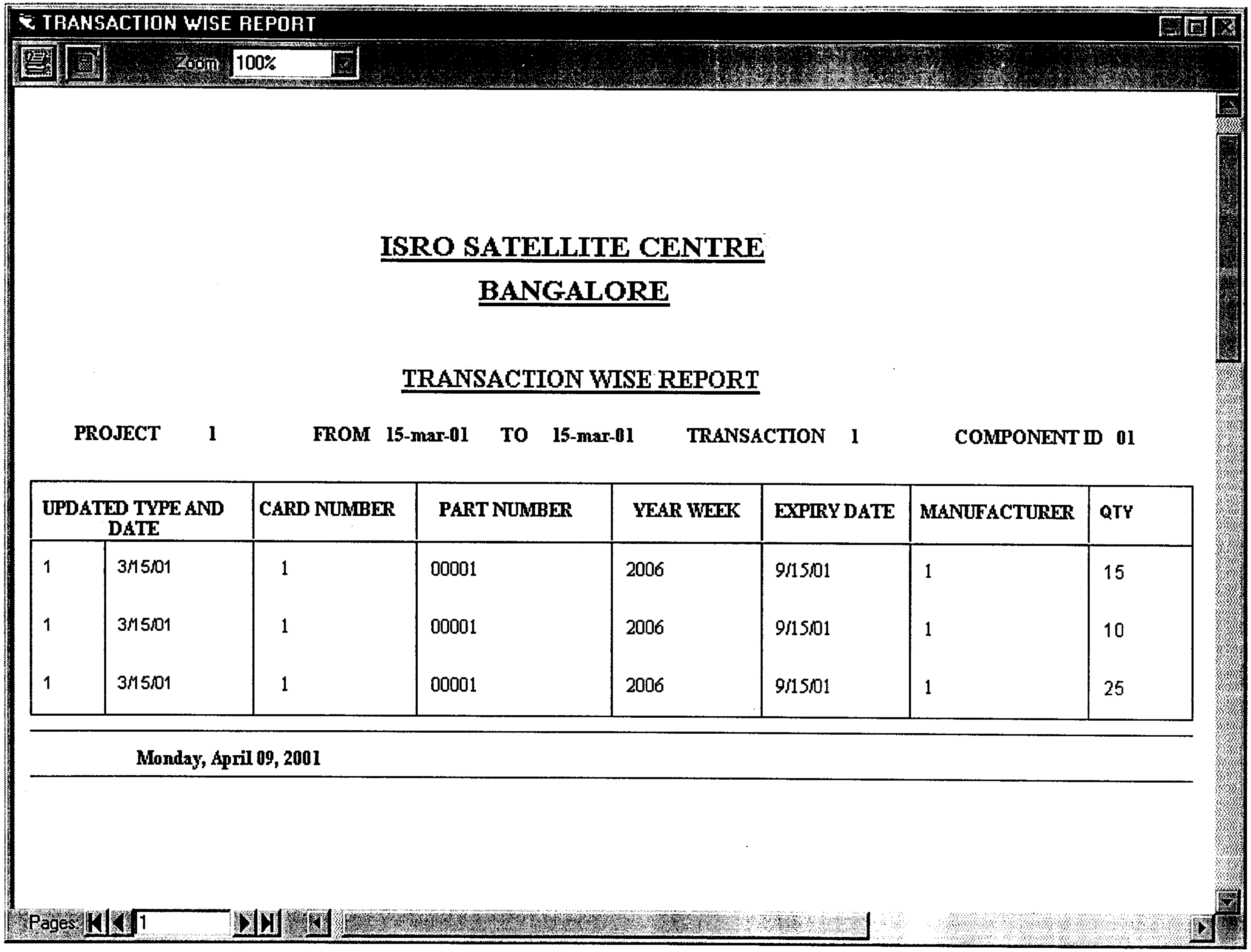
BANGALORE

ALL COMPONENT WISE REPORT

PROJECT 1 STOCK 1

CARD NUMBER	PART NUMBER	YEAR WEEK	EXPIRY DATE	MANUFACTURER	QTY
1	00001	2006	9/15/01	1	15
1	00001	2006	9/15/01	1	10
1	00001	2006	9/15/01	1	25
1	00001	2001	9/15/01	1	20
1	00001	2006	9/15/01	1	20

Monday, April 09, 2001



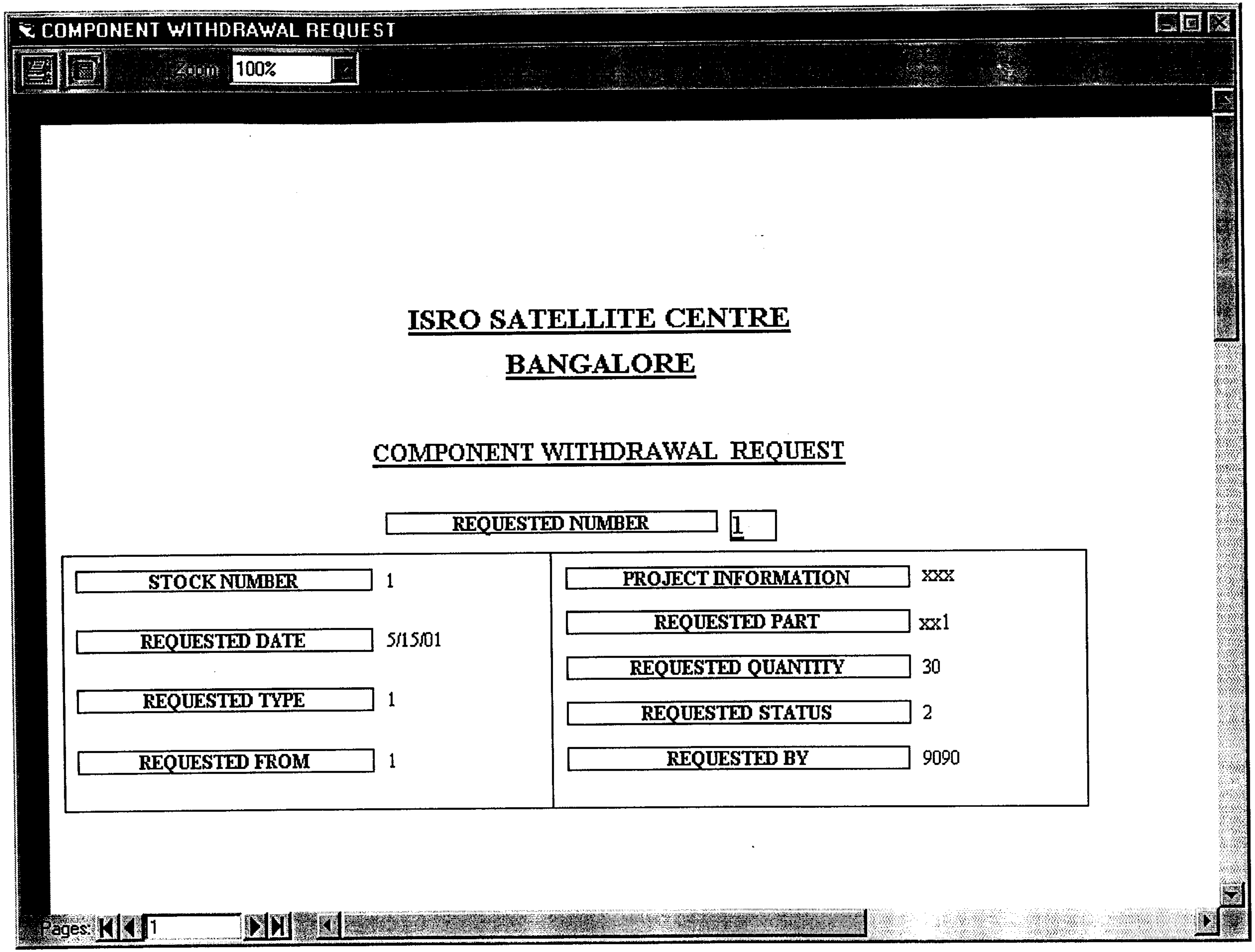
ISRO SATELLITE CENTRE
BANGALORE

TRANSACTION WISE REPORT

PROJECT 1 FROM 15-mar-01 TO 15-mar-01 TRANSACTION 1 COMPONENT ID 01

UPDATED TYPE AND DATE		CARD NUMBER	PART NUMBER	YEAR WEEK	EXPIRY DATE	MANUFACTURER	QTY
1	3/15/01	1	00001	2006	9/15/01	1	15
1	3/15/01	1	00001	2006	9/15/01	1	10
1	3/15/01	1	00001	2006	9/15/01	1	25

Monday, April 09, 2001



COMPONENT WITHDRAWAL REQUEST

Zoom 100%

ISRO SATELLITE CENTRE

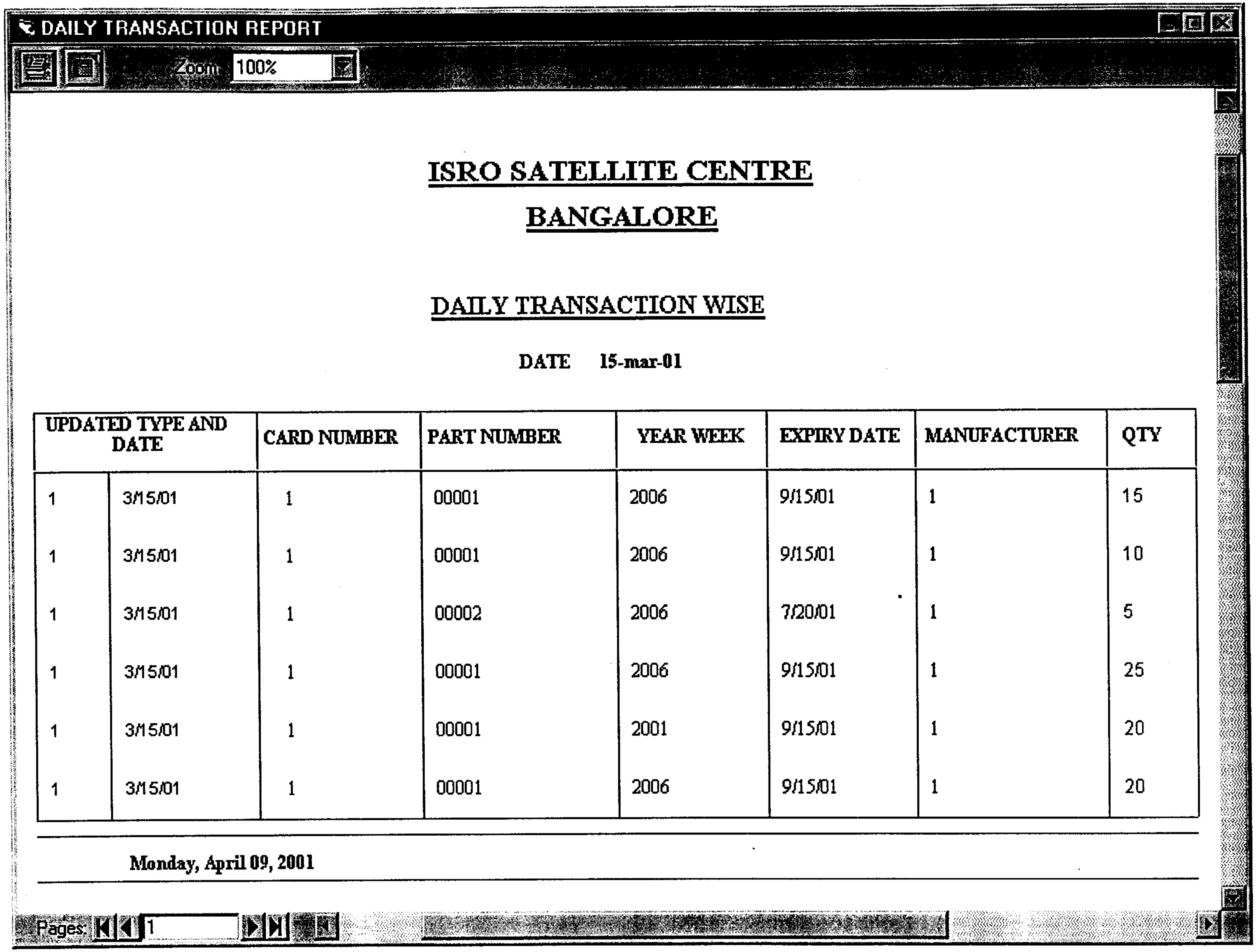
BANGALORE

COMPONENT WITHDRAWAL REQUEST

REQUESTED NUMBER 1

STOCK NUMBER	1	PROJECT INFORMATION	xxx
		REQUESTED PART	xx1
REQUESTED DATE	5/15/01		
		REQUESTED QUANTITY	30
REQUESTED TYPE	1		
		REQUESTED STATUS	2
REQUESTED FROM	1	REQUESTED BY	9090

Pages: 1



ISRO SATELLITE CENTRE

BANGALORE

DAILY TRANSACTION WISE

DATE 15-mar-01

UPDATED TYPE AND DATE		CARD NUMBER	PART NUMBER	YEAR WEEK	EXPIRY DATE	MANUFACTURER	QTY
1	3/15/01	1	00001	2006	9/15/01	1	15
1	3/15/01	1	00001	2006	9/15/01	1	10
1	3/15/01	1	00002	2006	7/20/01	1	5
1	3/15/01	1	00001	2006	9/15/01	1	25
1	3/15/01	1	00001	2001	9/15/01	1	20
1	3/15/01	1	00001	2006	9/15/01	1	20

Monday, April 09, 2001