“Store Management System”

Synopsis submitted to the IIMT College of Engineering,Greater Noida in partial fulfillment of the requirements for the award of the Degree of

B.Tech

In

(Information Technology)

By

Richa Gupta *1902160130039* *4th year*

Full time

Under the guidance of

Ms. Laveena Sehgal

IIMT COLLEGE OF ENGINEERING, GR. NOIDA

(Department of Information Technology)

(SESSION:2022-23)

CERTIFICATE

This is to certify that **Richa Gupta 1902160130039** has carried out the Mini project entitled as **"Store Management System"** during his 7th semester under the supervision of **Ms. Laveena Sehgal** in partial fulfillment of the requirements for the award of Bachelor of Technology degree in the department of Information Technology, IIMT College of Engineering.

DECLARATION

We students of 7th semester hereby declare that this project entitled **"Store management system"** has been carried out by us in the Department of Information Technology of IIMT College of Engineering under the guidance of **Ms. Laveena Sehgal.**

Name	**Signature**
Richa Gupta	……

Place **Date:**

IIMT College of Engineering,

Greater Noida. …….

ACKNOWLEDGEMENT

We would like to acknowledge and express my deepest gratitude to **Ms. Laveena Sehgal** for the valuable guidance, sympathy and co-operation for providing necessary facilities and sources during the entire period of this project.

I wish to convey my sincere gratitude to the Head of Department and all the faculties of IT Department who have enlightened me during our studies. The faculties and cooperation received from the technical staff of Department of IT is thankful acknowledged.

1. Richa Gupta (Roll No. 1902160130039)

INDEX

1. Introduction
   1.1 Abstract

2. System Analysis

   2.1 Software Requirement Specification (SRS)
   2.2 Existing system
   2.3 Software & hardware Requirements
   2.4 Feasibility Study
      2.4.1 Technical Feasibility
      2.4.2 Economical Feasibility
      2.4.3 Operational Feasibility
3. System Specification
   3.1 System Environment and Tools

4. System Design
   4.1 Data Flow Diagrams

5. System Implementation
   5.1 Screens and Reports
   5.2 Database Description

6. System Testing

7. Scope of the Project
   Issues Module
   Report Module

8. Conclusion

9. Bibliography

INTRODUCTION

1. Introduction

Stores are required for the following purposes :-

1. Capital works
2. Operation and Maintenances Works
3. Other Commercial activities like hiring equipment etc..,

The ‘Stores Management System’ is targeted to automate the almost all of the processes mentioned above to reduce the clerical labor of the staff working in Stores both technical and as well as Accounts departments using the software Industry’s latest technologies and cost effective tools there by providing the better control to the management by avoiding manual errors etc..,

In this project modules under study are Material Issues module, Reports module.

Material Issues module deals with the Issues functionality of the application. It mainly contains two operations namely Material issues i.e. Issues issued to works based on field requisitions and Inter stores issues i.e. material issues to other stores based on inter store requisitions. For these two operations, we have to issue the gate passes for both types of operations.

Reports module deals with the Reports provided by the application. This module contains various reports namely Monthly SRB Report, Monthly SIB Report, Monthly Section Wise Issues Reports, Priced Ledger, Monthly Stores Abstract, Monthly Work Order wise Details and Monthly Stock Report.

1.1. Abstract

Project Title: Stores Management System

The system creates a web based manufacturing system that enables a manufacturing industry to schedule its manufacturing operations based on the daily update of sales from its dealers. Once the sales figures of items for the past week are entered by the dealers over the internet along with the orders for the next delivery, the schedule for the next week’s production will be drawn up. A report of the required raw materials or parts will be drawn up with the product requirements over the internet & asked to quote their rates.

Once the rates are quoted, the order will be placed with the required delivery schedules. Once the parts the parts are supplied the stocks will be updated. Then a production plan will be drawn up taking the bill of materials into consideration. Once the production plan is approved, the stock will be updated when the material is issued. Once the

finished products are available the delivery schedules will be drawn up based on the orders placed by the Dealers. The stocks with the dealers will also be maintained.

The Benefits of the Stores Management System is

- It is used as an intranet Application.
- Providing High-Security.
- Easy Business Solutions.

SYSTEM ANALYSIS

2. System Analysis

2.1. Software Requirement Specification (SRS)

What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex, it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project us initiated by the client needs. The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

1) **Problem/Requirement Analysis:**
The process is order and more nebuious of the two, deals with understand the problem, the goal and constraints.

2) **Requirement Specification:**
Here, the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

The requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

Role of SRS

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

Purpose

The purpose of this document is to describe all external requirements for mobile task manager. It also describes the interfaces for the system.

2.2. Existing System

The existing system for Stores Management System activities uses open source standard & technologies. It had been developed on WINDOWS 2000 PROFESSIONAL platform with 'POSTGRESQL' database. All the stores of the power distribution company limited are made through the open source standards & technologies. The user interaction is in GUI (Graphical User Interface) mode.

Stores are required for the following purposes.

1. Capital works
2. Operation and Maintenances Works
3. Other Commercial activities like hiring equipment etc.

The 'Stores Management System package' is targeted to automate the almost all of the processes mentioned above to reduce the clerical labour of the staff working in Stores both technical and as well as Accounts departments using the software Industry's latest technologies and cost effective tools there by providing the better control to the management by avoiding manual errors etc..,

2.3. Hardware And Software Requirements

2.3.1. Hardware requirements

SERVER:

Processor	:	Pentium IV
Speed	:	1.7 GHz
Memory Capacity	:	1 GB
Hard Disk Capacity	:	80 GB
Monitor Make	:	HP

Client:

Processor	:	Pentium IV
Speed	:	1.7 GHz
Memory Capacity	:	256 MB
Hard Disk Capacity	:	20 GB
Monitor Make	:	HP

2.3.2. Software Requirements

Operating System	:	Windows 2000 Professional
Web Server	:	Apache Tomcat Web Server
Database	:	Postgresql
Enterprise Application	:	J2EE, JBOSS
Implementation Architecture	:	MVE, 3 Tier using Servlets, JSP
Scripting Languages	:	Java Script
Programming Language	:	Java

2.4. Feasibility Study

The existing system is clearly understood the next step is to conduct the feasibility study, which is a high level capsule version of the entire System Analysis and Design process. The objective is to determine whether the proposed system is feasible. The three tests of feasibility have been carried out:

2.4.1. Technical Feasibility

2.4.2. Economical Feasibility

2.4.3. Operational Feasibility

2.4.1. Technical Feasibility

In technical feasibility study, one has to test whether the proposed system can be developed using existing technology or not. It is planned to implement the proposed system using Windows 2000 Professional, JSP and Apache Tomcat Wed Server. The Organization already possesses Windows 2000 Professional Operating System. It is evident that the necessary hardware and software are available for the development and implementation of the proposed system. Hence the solution is technically feasible.

2.4.2. Economical Feasibility

As part of this, the costs and benefits associated with the proposed system are to be compared and the project is economically feasible only if benefits outweigh costs. The Organization has already its own satellite link, and a host of SUN FIRE 6800 servers. So it need not invest newly for the internet connection and also the organization initiated to use Open Source in project development, hence there is 0 additional cost incurred for the tools that will be used.

2.4.3. Operational Feasibility:

This test of feasibility checks if the system works with least difficulties when it is developed and installed. The technical staff has sufficient knowledge of the tools being used and the users need just to know how to access and run the programs in the Apache Web Sever. Hence it is concluded that the system is operationally feasible.

SYSTEM SPECIFICATION

3. System Specification

3.1. System Environment and Tools

JAVA

Creation Of Java

Java was conceived by James Gosling Patrick Naughton, Chris Wrath, Ed Frank and Mike Sheridan at SUN Micro Systems Incorporation in 1991. It took 18 months to develop the first working version. This language was initially called “oak”, but was renamed “java” in 1995, many more contributed to the design and evolution of the language.

Java Overview:

Java is powerful but lean Object-Oriented programming language. It has generated a lot of excitement because it makes it possible to program for Internet by creating applets, programs that can be embedded in the web page.

The context of an applet is limited by only one’s imagination. For Example, an applet can be an imagination with sound, an interactive game or a ticker tape with constantly updated stock prices. Applets can be just like decorations to liven up the WebPages, or they can be serious applications like word processors or spreadsheet.

But java is more than programming languages for writing applets. It is being used more and more for writing standalone applications as well. It is becoming so popular that many people believe it will become standard language for both general purpose and Internet Programming.

There are many buzzwords associated with java, but because of its spectacular growth in popularity, a new buzzword has appeared ubiquitous. Indeed, all indications are that it will soon be everywhere.

Java builds on the strength of C++. It has taken the best features of C++ and discarded the more problematic and error prone parts. To this lean core, it has added garbage collection (automatic memory management), multi threading (the capacity for one program to do more than one thing at a time) and Security capabilities. The results are that java is simple, elegant, powerful and easy to use.

Java is actually a platform consisting of three components:

1) Java programming language.

2) Java library of classes and interfaces.

3) Java virtual Machine.

The following sections will say more about these components:

Java is Portable:

One of the biggest advantages Java offers is that it is portable. An application written in Java will run on all the major platforms. Any computer with a Java based browser can run the applications or applets written in the Java programming language. A Programmer no longer has to write one program to run on a Macintosh, another program to run on a windows machine, still another to run on a UNIX machine, and so on. In other words, with Java, developers write programs only once. Rather than being compiled in to machine language, which is different for each operating systems and computer architecture, Java code is compiled in to byte codes.

With other languages, the program code is compiled in to a language that the computer can understand; the problem is that other computers with different machine instructions set cannot understand that language. Java code, on the other hand is compiled in to byte codes rather than a machine language. These byte codes go to the Java virtual machine, which executes them directly or translates them in to the language that is understood by the machine running it.

In the summary, these means that with the JDBC API extending Java, a programmer writing Java code can access all the major relational databases on any platform that supports the Java virtual machine.

Java Is Object-Oriented:

Java programming language is object oriented, which makes program design focus on what you are dealing with rather than on how you are going to do something. This makes it more useful for programming in sophisticated

projects because one can break the things down into understandable components. A big benefit is that these components can than be reused.

Object Oriented languages use the paradigm of classes. In simplest terms, a class includes both the data and the functions to operate on the data, all the data members and functionality of its class. Because of this, you can think of a class as being like template, with each object being a specific instance of a particular type of a class.

The class paradigm allows one to encapsulate data so that specific data values are those using the data cannot see function implementation. Encapsulation makes is possible to make the changes in code without breaking other programs that use that code. If for example the implementation of a function is changed, the change is invisible to the programmer who invokes that function, and it does not affect his/her program, except hopefully to improve it.

Java includes inheritance, or the ability to derive new classes from existing classes. The derived class, referred to as the parent class. A subclass can add new data members to those inherited from the parent class. As far as methods are concerned, the subclass can reuse the inherited methods as it is, change them, and/or add its own new methods.

Java Makes It Easy To Write Correct Code:

In addition to being portable and object oriented, java facilitates writing correct code. Programmers spend less time writing java code and a lot less time debugging it. In fact, development time reduces by as much as two thirds.

The following is a list of some of java's features that make it easier to write correct code:

Garbage Collection:

Automatically takes care of allocating and deallocating memory, a huge potential source of errors(Memory Leaks). If an object is no longer being used, then it is automatically removed from memory, or "garbage collected". Programmers don't have to keep track of what has been allocated and deallocated themselves, which makes their job a lot easier, but more importantly it stops memory leaks.

No Pointers:

Eliminates big source errors. By using object references instead of memory pointers, problems with pointer arithmetic are eliminated, and problems with inadvertently accessing the wrong memory address are greatly reduced.

Strong Typing:

Cuts down on runtime errors. Because java enforces strong type checking, many errors are caught when code is compiled. Dynamic binding is possible and often very useful, but static binding with strict type checking is used when possible.

Simplicity:

Makes java easier to learn and use correctly. Java keeps it simple by having just one way to do something instead of having several alternatives, as in some languages. Java also stays lean by not including multiple inheritances, which eliminates the errors and ambiguity that arise when you create a subclass that inherits from two or more classes.

To replace capabilities, multiple inheritances provide, Java lets you add functionality to a class throw the use of interfaces.

Java Includes a Library Of Classes And Interfaces:

Java platform includes an extensive class library so that programmers can use already existing classes, or implement interfaces to augment the capabilities of classes.

Both classes and interfaces contain data members and functions, but there are major differences. In a class, fields may be either variable or constant, and methods are fully implemented. In an interface, fields must be constants, and methods are prototypes with no implementations. The prototypes give the method signature (the return type, the function name, and the number of parameters with the type for each parameter), but the programmer must supply implementations. To use an interface, a programmer defines a class, declares that it implements the interface, and then implements all the methods in that interface as part of the class.

These methods are implemented in a way that is appropriate for the class in which the methods are being used. Interfaces let one add functionality to a class and give a great deal of flexibility in doing it. In other words interfaces provide most of the advantages of multiple inheritance without its advantages.

A Package is a collection of related java classes and interfaces. The following list, though not complete, gives examples of some java packages and what they cover.

Java. lang--the basic classes. This package is so basic that it automatically is +include in any java program. It includes classes dealing with numeric, Strings, objects, runtime, security and threads.

Java.io-- classes that manage reading data from input streams and writing data to the output streams.

Java.util-- miscellaneous utility classes, including generic data structures, bit sets, time, date, the string manipulation, random number generation, system properties, notification and enumeration of data structures.

Java.net--classes for network support.

Java.awt--classes that manage user interface components such as windows, dialog boxes, buttons, checkboxes, lists, menus, scrollbars, and text fields the "AWT" stands Abstract window Toolkit.

Java.awt.image--classes for managing image data, including color models,
dropping, color filtering, setting pixel values, and grabbing snapshots.

Java.applet--The Applet class, which provides the ability to write applets, this package also includes several interfaces that connect an applet to its document and to resources for playing audio.

It is important that a programmer not be able to write subversive code for applications or applets. This is especially true with the Internet being used more and more extensively for services such as electronic commerce and electronic distributions of software and multimedia content.

The Java platform builds in security in four ways.

The way the memory is allocated and laid out: in Java objects location in memory is not determined until the runtime, as opposed to **C** and **C++,** where the compiler marks memory layout decisions. As the result, a programmer cannot look at a class definition and figure out how it might be laid out in memory. Also since, java has no pointers, a programmer cannot forge pointers to memory.

The Way Classes Are Loaded:

The java virtual machine doesn't trust any incoming code and subjects it to what is called Byte code verification. The byte code verifier, part of the virtual machine, checks that

A: The format of incoming code is correct.

B: Incoming code doesn't forge pointers.

C: It doesn't violate access restrictions.

D: It accesses objects as what they are.

The Java byte code loader, another part of the virtual machine, checks whether classes loaded during program execution is local or forms across a network. Imported classes cannot be substituted for built in classes, and built in classes cannot accidentally references classes brought in over a network.

The Way Access Is Restricted For Un trusted Code:

Java security manager allows user to restrict un trusted Java applets so that they cannot access the local network, local files and other resources.

Java Performs Well:

Java performance is better than one might expect. Java's many advantages, such as having built in security and being interpreted as well as compiled, do have a cost attached to them. However, various optimizations have been built in, and the byte code interpreter can run very fast the cost it does not have to do any checking. As a result, java has done quite respectably in performance tests. Its performance numbers for interpreted byte codes are usually more than adequate to run interactive graphical end user applications.

For situation that require unusually high performance, byte codes can be translated on the fly, generated the final machine code for the particular CPU on which the application is running at run time.

High level interpreted scripting languages generally offer great portability and fast prototyping but poor performances. Low level compiled languages like **C** and **C++** offer great performances but require large amounts of time for writing and debugging the code.

Because of problems with areas such memory management, pointers and multiple inheritance. Java offers good performance with the advantages of high level languages but with out the disadvantages of **C** and **C++.** In the world of design trade-offs, you can think of Java providing a very attractive middle ground.

Java Is Robust:

The multi plat formed environment of the web places extraordinary demand on a program, because it must execute reliably in a variety of systems. Thus the ability to create robust programs was given a high priority in the design of Java. To gain reliability, java restricts you in a few key areas to force you to find your mistakes early in program developments. At the same time java frees you from having to worry about many of the most common causes of programming errors. Because Java is strictly typed language, it checks your code at compile time. However, it also checks at runtime. Infact, many hard to track down bugs that often turn up in hard to reproduce runtime situations are simply impossible to create in Java. Knowing that what you have written will behave in a predictable way under diverse conditions is a key feature of java.

To understand how java Robust, consider two main reasons for program failure:

Many management mistakes and mishandled exceptional conditions (run time errors). Memory management can be difficult, tedious task in traditional programming environments. For example some programmers some times forget the memory that has been previously allocated. Or worse, they may free memory that another part of their code is still using.

Java virtually eliminates these problems by managing memory allocates and
Deal locations. Java helps in this area by providing object oriented exception handling. In a well-written java a program should manage program all run time errors.

Java Scales Well:

Java platform is designed to scale well, from portable consumer electronic devices to powerful desktop and server machines. The virtual machine takes a small footprint and Java byte code is optimized to be small and compact. As a result, Java accommodates the need for low storage and for low bandwidth transmission over the Internet. In addition the Java operating system offers a standalone java platform API. This makes java ideal for low cost network computers whose sole purpose is to access the Internet.

Java Is Multithreaded:

Multithreading is simply the ability of a program to do more than one thing at a time. For example an application could be faxing a document at the sane time it is printing another document. Or a program could process new inventory figures while it maintains fever for current prices. Multithreading is particularly important in multi media: a multimedia program might often be running a movie, running a audio track displaying text all at the same time.

Why Java Is Important To The Internet:

The Internet helped catapult java to the forefront of programming and java in turn has had profound effect on the Internet. The reason is simple. Java expands the universe of objects that can move about freely in cyberspace. In a network, there are two broad categorizes of objects transmitted between the server and your personal computer. Passive information and dynamic active programs. For example when you read your email, you are viewing passive data. Even when you download a program, the program's code is still passive data until it is executed. However, there is a second type of object that can be transmitted to your computer: a dynamic, self-executing program. Such a program would be an active agent on the client computer, yet the server would initiate it.

Java Server Pages (JSP)

A JSP page is a text page document that describes how to process request to create a response. JSP is a Java-based technology that simplifies the process of developing dynamic web sites. With JSP, web designers and developers can quickly incorporate dynamic elements into web pages using embedded Java and simple mark-up tags. These tags provide the HTML designer with a way to access data and business logic stored inside Java object.

JSP are txt files with extension '.jsp' which takes place of traditional HTML pages. JSP files contain traditional HTML along with the embedded Code that allows the developer to access data from the Java code running on the server.

JSP is now an integral part of developing web-based application using java. Because of the ability to separate presentation from implementation logic by combing standard mark-up text with scripting element and object-oriented components, JSP provides an excellent front end technology for applications that are the deployed over the web.

Application Model:

JSP pages can be used in combination with servlets, http, html, xml, applets, java beans components and enterprise java beans components, to implement a wide variety of application architecture(s)or models. To facilitate web-based applications, both servlets and JSP are required elements of j2ee specifications. And while both technologies can be used to dynamically generate html to be send to an end-users web browser, only JSP enforces a strong separation between presentation logic involved in displaying data and the business or programming logic used to generate data in first place. When using JSP, changes to presentation of the data do not require any changes to the underlined server code that supplied data in the first place.

The java server pages technology offers the following advantages:

- Write once, run anywhere properties
- High quality tool support.
- Re-use of components and tag libraries.
- Separation of dynamic and static content:
- Support for scripting and actions
- Web access layer for N-tier enterprise application architecture.

Executing JSP:

Adding JSP support:

The most basic requirements for using JSP are a web server implies. Both hardware in the form of a computer accessible over the Internet or a corporate based intranet and s/w in the form of an HTP server running on that hardware. Some of most popular http servers are: apache, Microsoft Internet information sever, web logic etc.

JSP execution start with request for a JSP page, the processing is done based on JSP tag on the page In order to generate contents dynamically the output of the processing is combined with the page static HTML, and presented on the web browser.

Scripting Languages:

The default scripting language for JSP is, naturally, java. Since JSP Pages are compiled into java servlets, this assumption makes the translation of the scripts into servlets code very straight forward. To be acceptable for use with JSP, scripting languages must meet three requirements:

1. It must support the manipulation of Java objects.
2. It must include the ability to catch Java exceptions and specify exception handlers.
3. It must be able to invoke methods on Java objects.

JSP Tags:

JSP provides four major categories of markup tags:

1. Directives, is a set of tags for providing the JSP container with page specific instructions for how the documents containing the directives are to be processed.

2. Scripting elements are used to embedded programming instructions written in the designated scripting language for the pages which are to be executed each time the page is processed for request.

3. Comments are used for adding documentation strings to a JSP page.

4. Action support different behaviors.

<%=%>	:	USED FOR EXPRESSIONS
<%%>	:	USED FOR SCRIPLETS
<%!%>	:	USED FOR DECLARATION
<%@%>	:	USED FOR DIRECTIVES
<%@page%>	:	USED FOR PAGE DIRECTIVES

Expressions:

A JSP expression element is a scripting language expression that is evaluated and that can be converted into a string which is succeed emitted into the out object.

Syntax: <%= expression%>

Scriplet:

Scriplet can contain any code fragment that is valid for scripting language fragment.

Syntax: <%scrip let %>

Declarations:

These are used to declare variables and methods in the scripting language.

A declaration is initialized when the JSP page is initialized and is made available to other declarations and scriplets.

Syntax: <%! Declarations%>

Directives:

These are messages to JSP engine.

Syntax: <%@directives%>

Page Directives:

The page directive defines a number of page dependent attributes and communicates these to the JSP engine.

Syntax: <%@page = directive-attribute-list%>

Introduction To HTML:

The Hypertext Mark Up Languages (HTML) is simple markup language. Used to create hypertext documents that are portable from one platform to another HTML documents are SGML documents with generic semantics that are appropriate for representing information from a wide range of applications. This specification defines HTML version 3.2. HTML 3.2 aims to capture recommended practice as early ’96 and as such replacement for HTML 2.0(RFC 1866).

A set of instruction embedded in a document is called Markup Language. These instructions describe what the document text means and how it should look like in a display. Hyper Text Mark Up Language (HTML) is the language used to encode World Wide Web documents. It is a document layout and hyperlink specification language that defines the syntax and placement of special embedded directions that aren’t displayed by a web browser, but tells it how to display the contents of the documents including text, images and other supported media.

Why to use HTML:

Web site is a collection of pages, publications, and documents that reside on web server. While these pages, publications and documents are formatted in any single format, you should use HTML for home page and all primary pages of the site. This will enable the millions of users to easily access and to take advantage of your web site.

HTML is considered first for formatting any new material you plan to publish on the web. HTML documents are platform independent, meaning that they don't conform to any standard .If they are created properly you can move home page to any server platform or you can access them with any complaint www browser.

Block of HTML:

HTML elements perform a defined task. HTML uses two types of elements.

Empty tags

Container tags

Empty (or open) tags and container tags differ because of what they represent formatting constricts such as line breaks and horizontal rules.

Container tags define a section of text and specify the formatting the container dot all of the selected text. A container tag has both a beginning and an ending.

HTML layout:

An HTML document consists of text, which comprises the content of the document and tags which format the structure and appearance of the document. The structure of an HTML document is impelled, consist of outer.

<HTML> tag enclosing the document header and body:

<HEAD>

<TITLE> The title of the HTML document </TITLE>

</HEAD>

<BODY>

This is where the actual HTML documents Text lies which is displayed in browser

</BODY>

</HTML>

Each document has a head and body delimited by the **<HEAD>** and **<BODY>** tag. The head is where you give your HTML document a title and where you indicate other parameters the browser may use when displaying the document. This includes the text for displaying the text. Tags also references special and indicates the hot spots that link your document to other documents.

HTML Forms:

Creating a from usually involves two independent steps: creating the layout for the form itself and then writing a script program on the server side to process the formation you get back from a form.

To create a form, you use the **<FORM>** tag. Inside the opening and closing form tags are each of the individual form elements plus any other HTML content to create a layout for the form.

The opening tag of the FORM element usually includes the attributes: METHOD and ACTION. The METHOD attributes can be either GET or POST which determines how your form data is sent to the script to process it.

The ACTION attribute is a pointer to the script that processes the form on the server side.
A full URL to a script on your server or somewhere else can indicate by a relative path or the Action. For example, the following <FORM> tag would call a script called form name in cgi-bin director on the server.

www.myserver.com
<form method=post action=http://localhost:8080/webapps/examples/agl.jsp >
</form>

Method Attributes:

The other required attributes for the <form> tag sets the methods by which the browser form's data is sent to the server for processing.

There are two ways:

1. POST method.

2. GET method.

With POST method, the browser sends the data in two steps: the browser first contacts the form-processing server specified in the action attributes, and once contact is made, sends the data to the server in a separate transmission.

The GET method on the other hand contacts the form-processing server and sends the form data in a single transmission step: the browser appends the data to the forms action URL, separated by the question mark (?) character.

About JDBC

The JDBC™ API provides universal data access from the Java™ programming language. Using the JDBC 3.0 API, you can access virtually any data source, from relational databases to spreadsheets and flat files. JDBC technology also provides a common base on which tools and alternate interfaces can be built.

The JDBC 3.0 API is comprised of two packages:

1. The java.sql package

2. The javax.sql package, which adds server-side capabilities

The JDBC-ODBC Bridge allows applications written in the Java™ programming language to use the JDBC™ API with many existing ODBC drivers.

The Bridge is itself a driver based on JDBC technology ("JDBC driver") that is defined in the class " sun.jdbc.odbc.JdbcOdbcDriver " The Bridge defines the JDBC sub protocol Odbc.

Status of the JDBC-ODBC Bridge:

The JDBC-ODBC Bridge should be considered a transitional solution. Sun Microsystems and Merant are working to make the Bridge more reliable and robust, but they do not consider it a supported product. With the development of pure-Java JDBC drivers, the JDBC-ODBC Bridge should become unnecessary.

Why not just use ODBC?

The JDBC-ODBC Bridge offers several advantages over "naked" ODBC.

- The ODBC API is strongly oriented toward C/C++ programmers. The JDBC-ODBC Bridge makes it unnecessary for programmers using the Java programming language to deal with non-Java language concepts.
- The ODBC API is extremely complex and intermingles high-level and low-level functions. The JDBC-ODBC Bridge allows the programmer to rely on the JDBC API, which is relatively simple and easy to learn.
- The JDBC-ODBC Bridge lets a program deal with ODBC through a standardized JDBC interface. That helps a program avoid being tied to ODBC when a better solution comes along.

SYSTEM DESIGN

4. System Design:

4.1. Data Flow Diagrams:

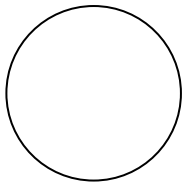
The data flow diagrams from the important modeling tools in the structure system analysis methodologies. The data flow diagrams are on of the most important tools used by system analyst.

Data flow diagram should be the first tool used by the system analyst to model system components. There are three kinds of system components.

1. Process
2. Entity
3. Data flow

1. Process:

Processes show what system does. Each process has one more data inputs or more data outputs. Circles in a DFD represent processes. Each process has a unique name and number. This name and number appear inside the circle that represents the process in a



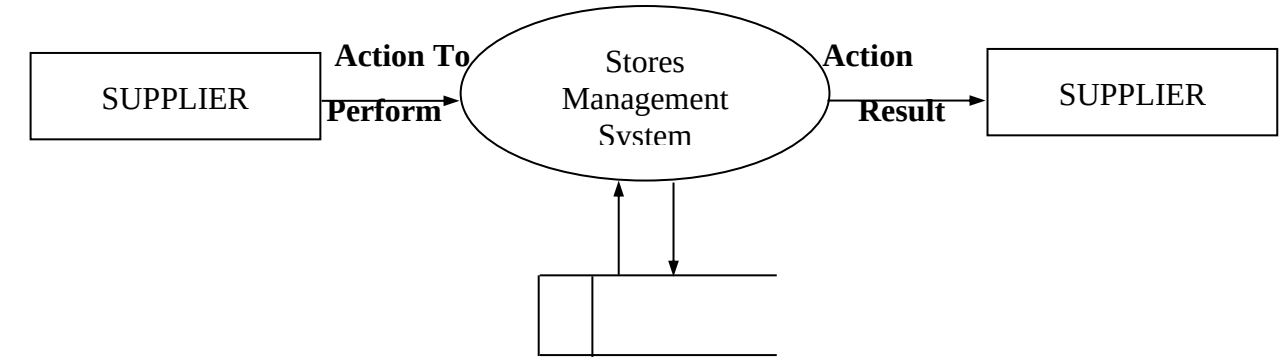
2. External Entities:

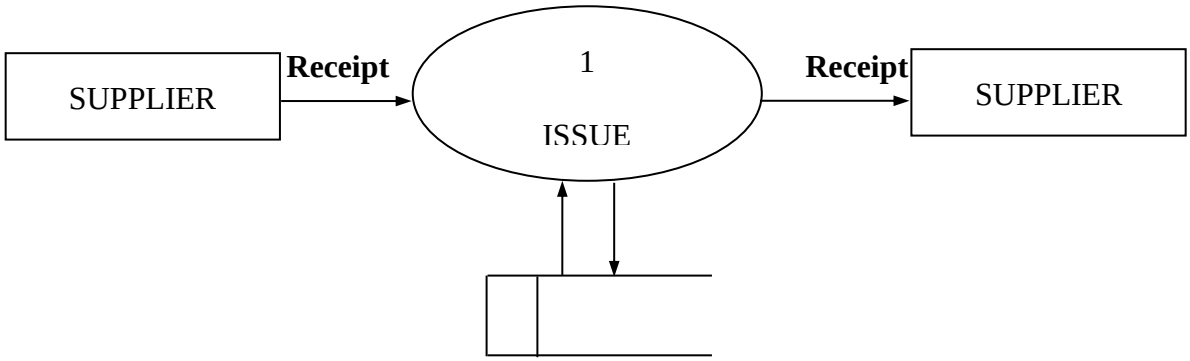
External entities are outside the system they either supply input data into the system or use the system output. They are entities over which the designer has no control. They may be organizational customers or other bodies with which the system interact. External entities may be represented by a square or rectangle. External entities that supply data into a system are sometimes called sources. External entities that use the system data are sometimes called sources. External entities that use the system data are sometimes calls sinks.

3. Data flows:

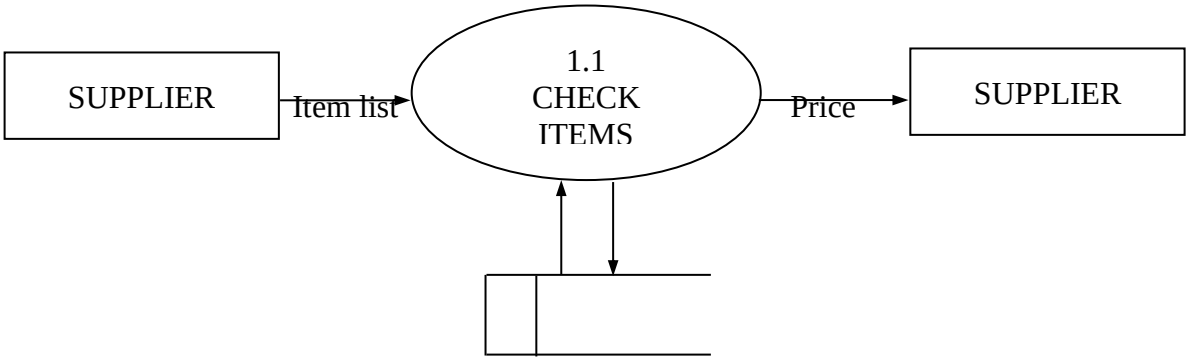
Data flows model the passage of the system and are represented by lines joining the system components. An arrow indicates the direction of the flow and the lines is labeled by the name of the data flow. Flow of data in the system can take place.

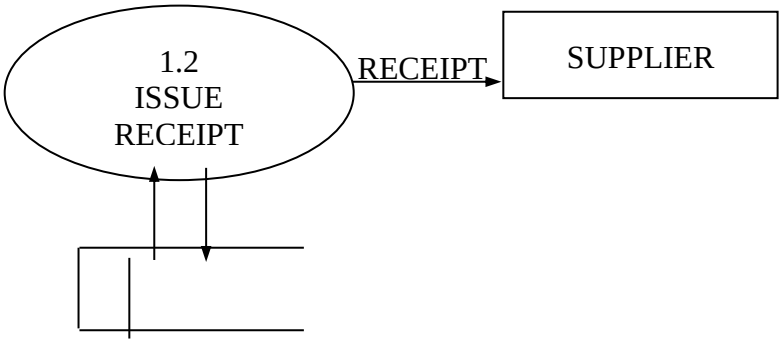
CONTEXT LEVEL



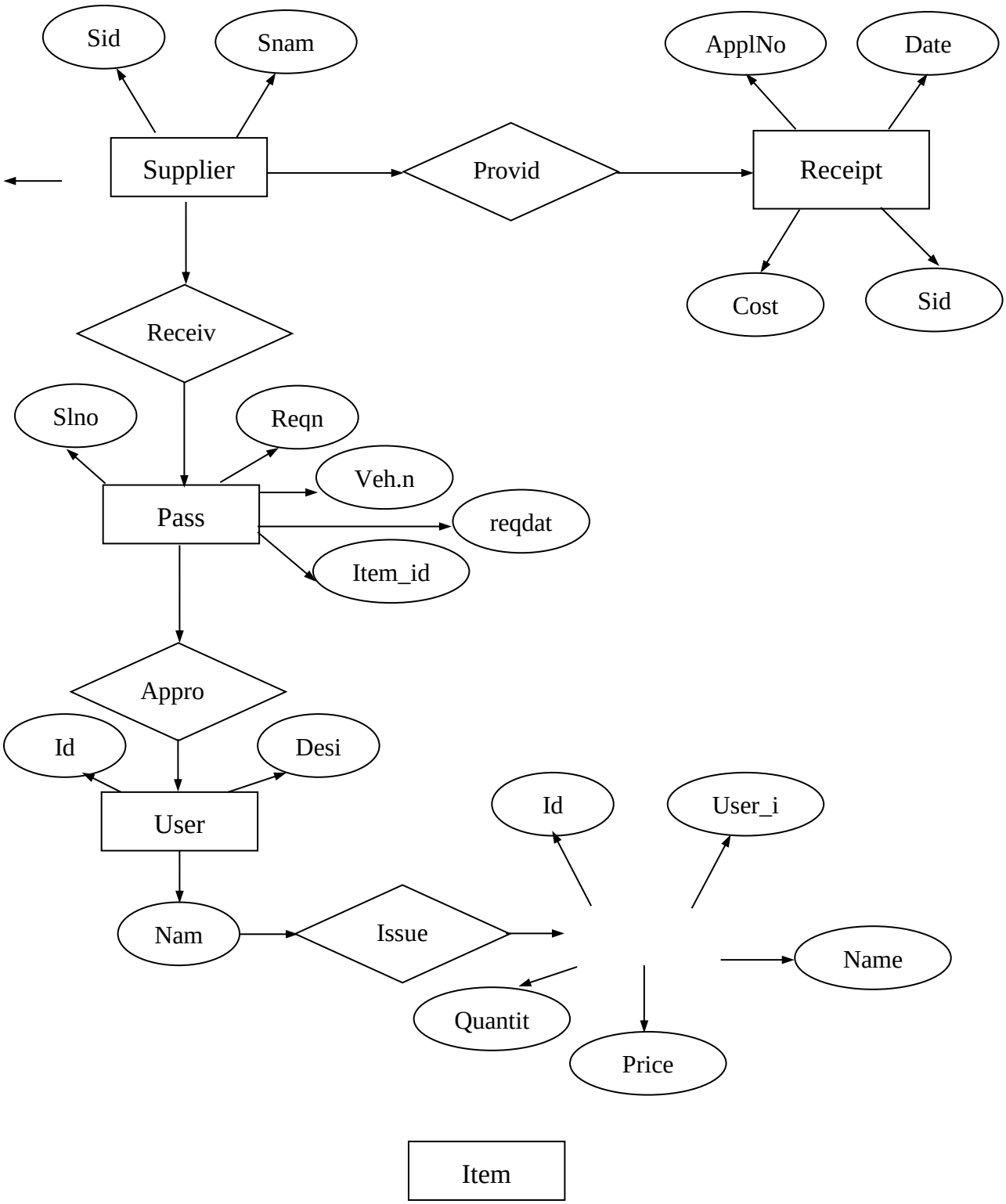


LEVEL 02


SUPPLIER
Item list
1.1
CHECK
ITEMS
Price
SUPPLIER


1.2
ISSUE
RECEIPT
RECEIPT
SUPPLIER

Issues Module

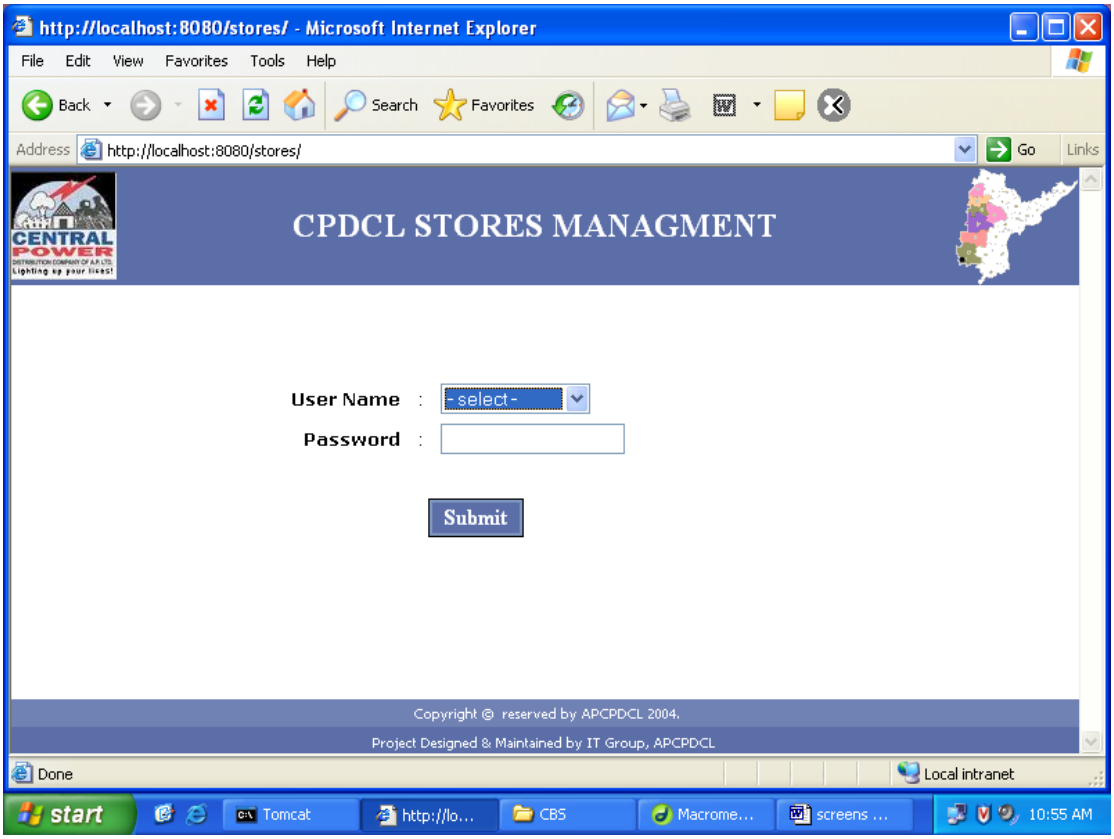


SYSTEM IMPLEMENTATION

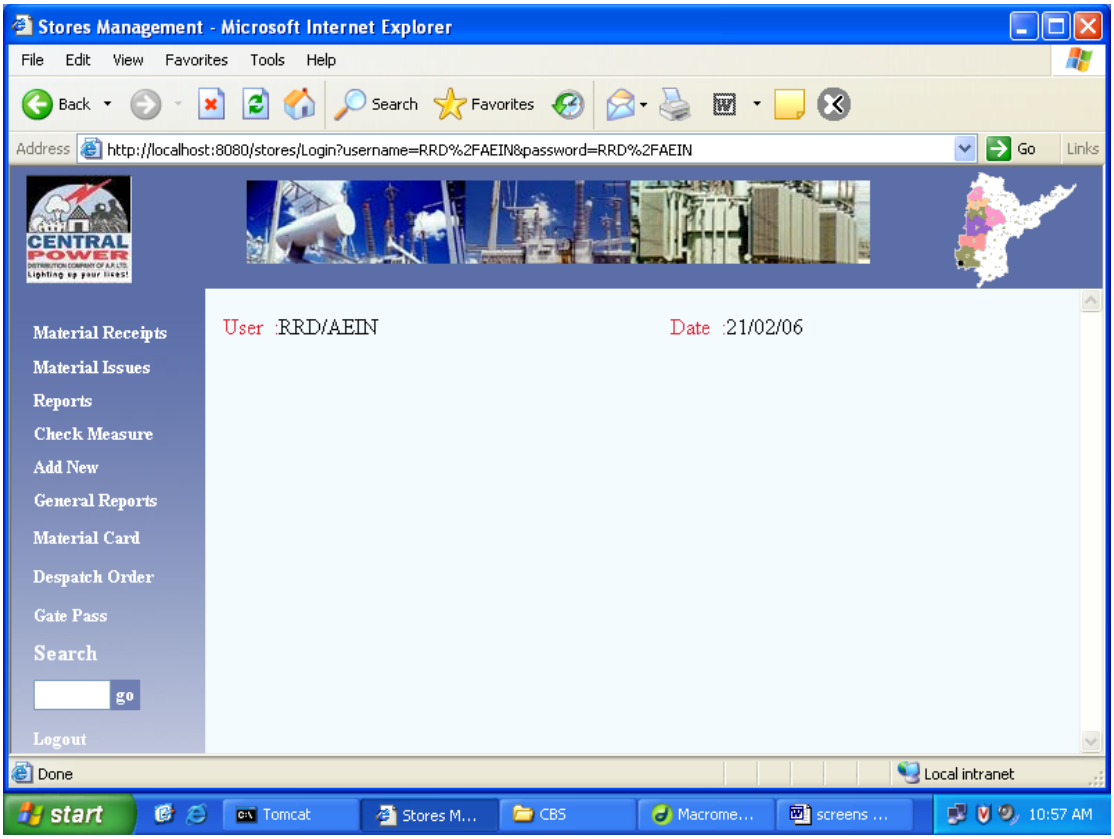
5. System Implementation:

5.1. Screens & Reports:

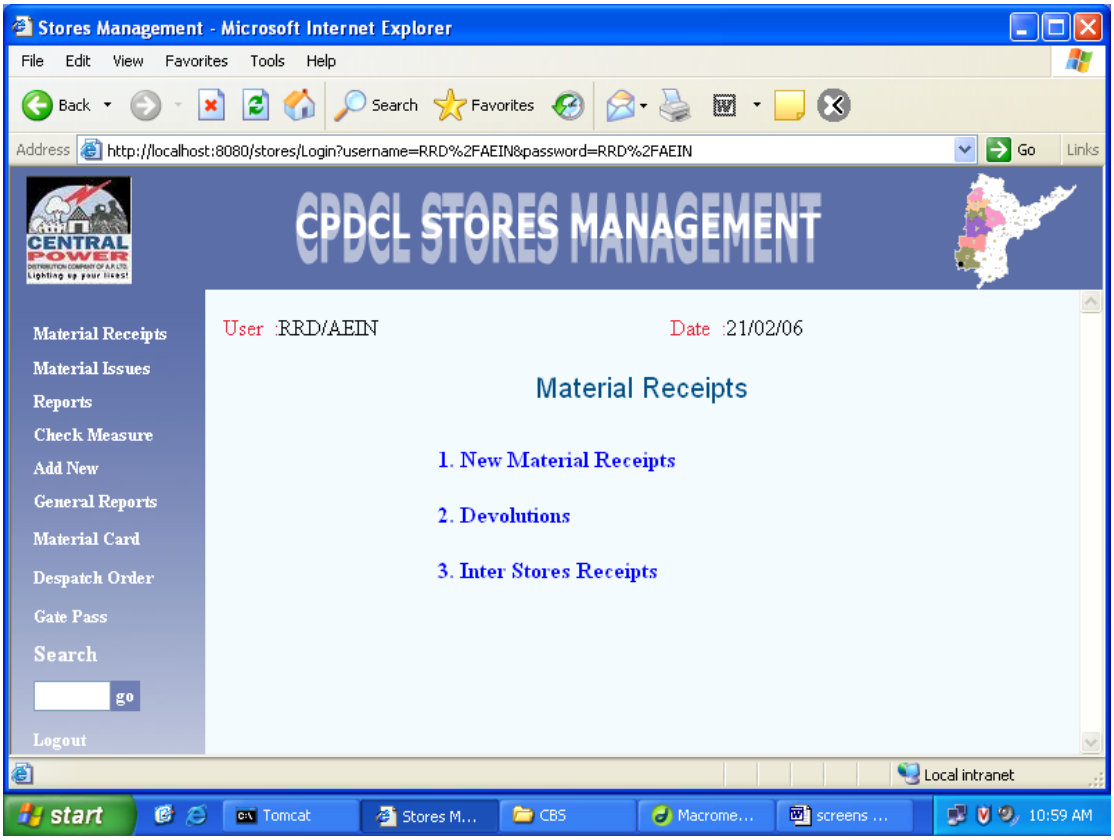
1. Login page



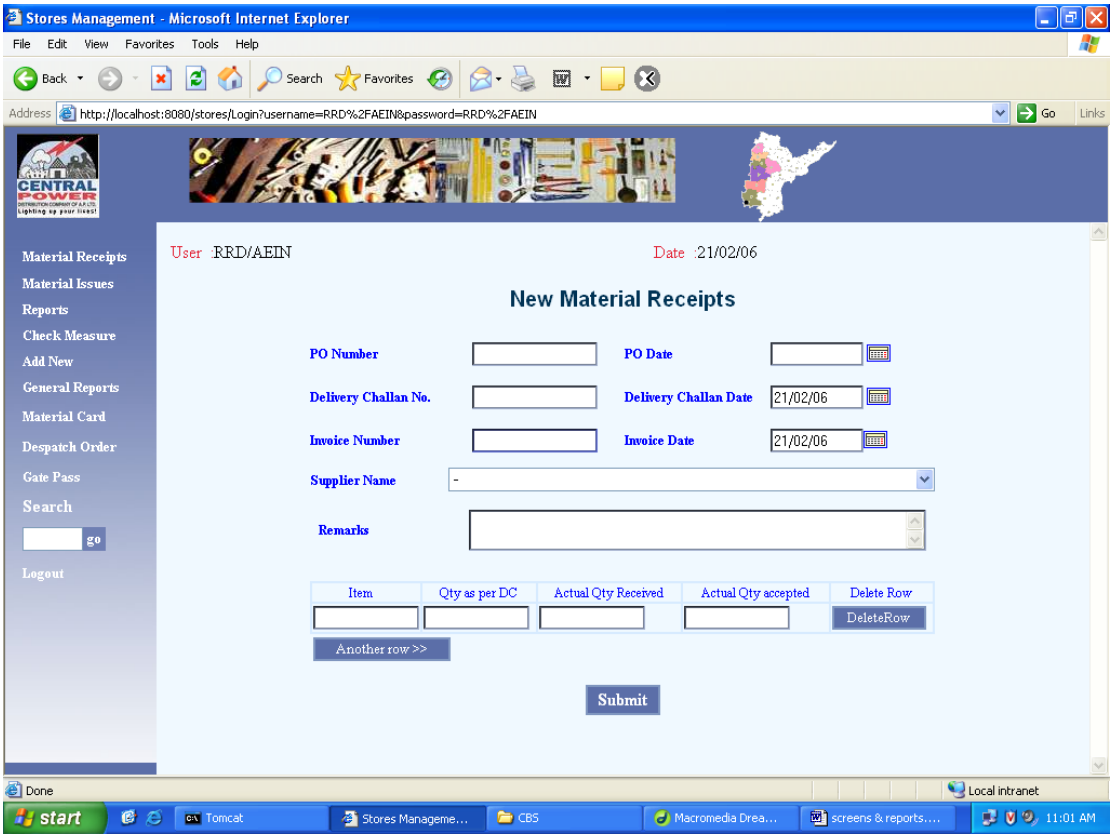
2. Main page


Stores Management - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/stores/Login?username=RRD%2FAEIN&password=RRD%2FAEIN
Go Links
CENTRAL POWER
User :RRD/AEIN
Date :21/02/06
Material Receipts
Material Issues
Reports
Check Measure
Add New
General Reports
Material Card
Despatch Order
Gate Pass
Search
go
Logout
Done
Local intranet
start
Tomcat
Stores M...
CBS
Macrome...
screens ...
10:57 AM

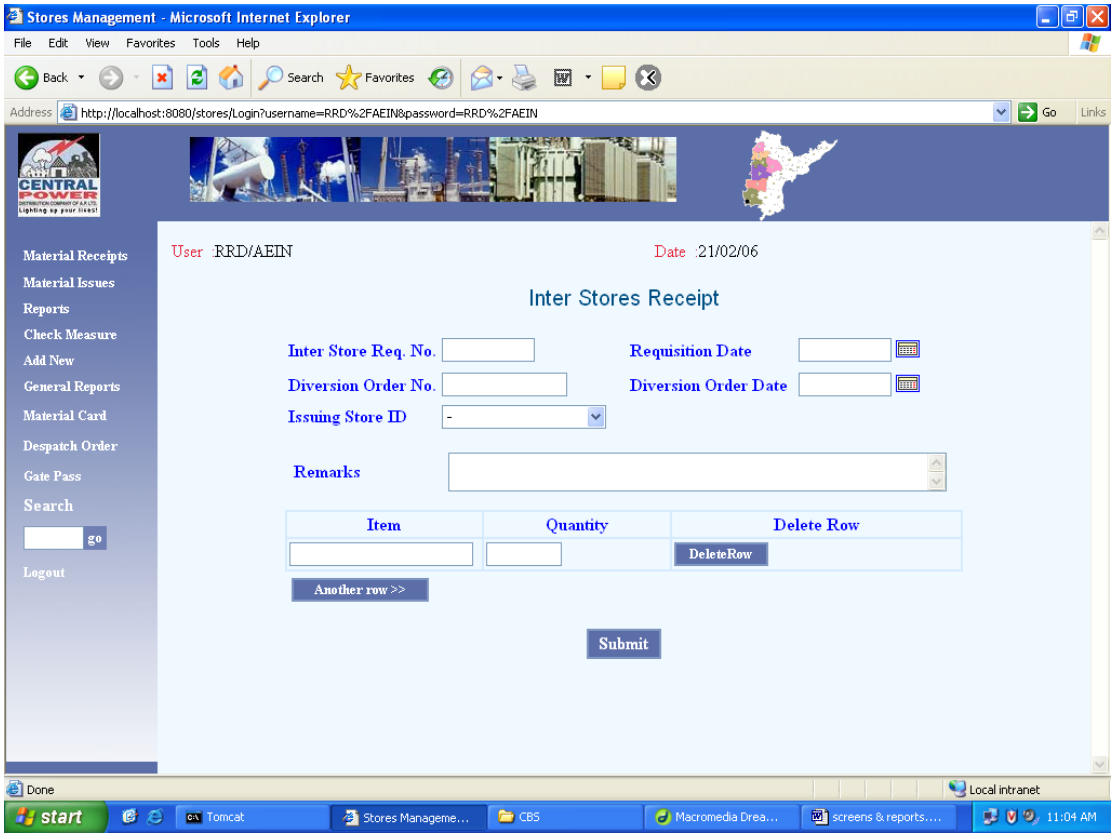
3. Material Receipts Module



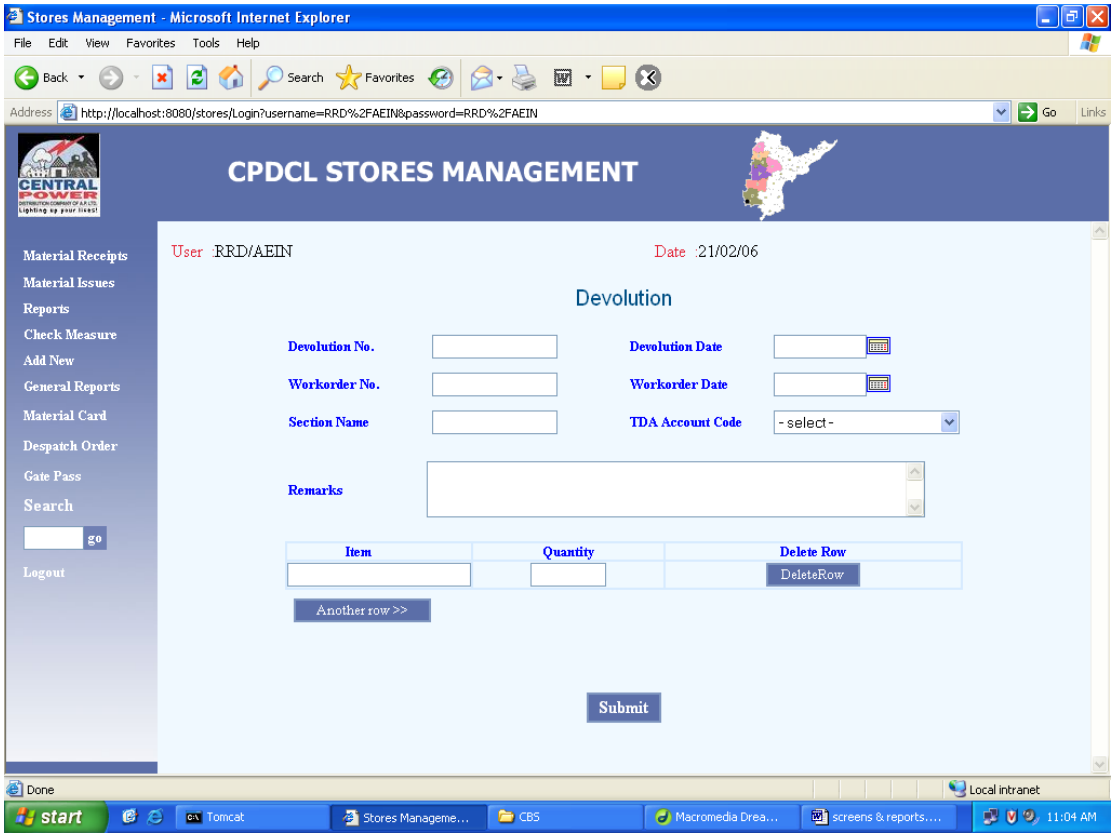
New Material Receipts



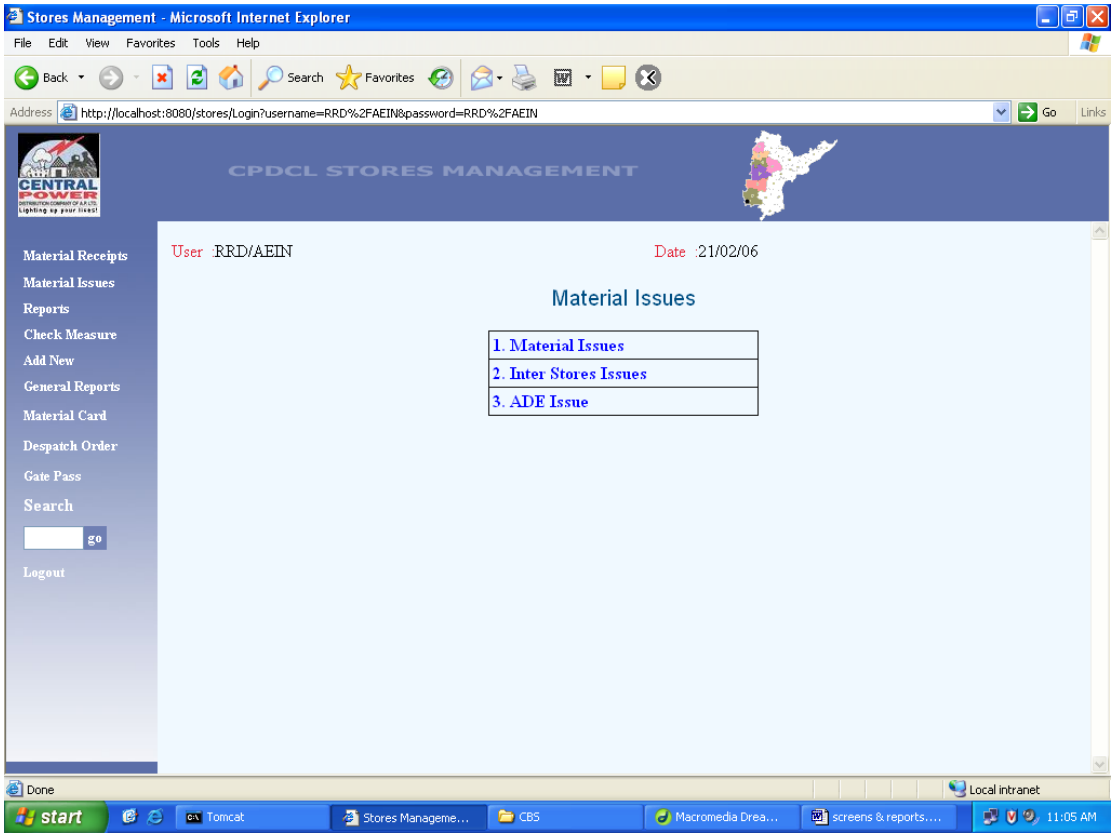
Inter Stores Receipts



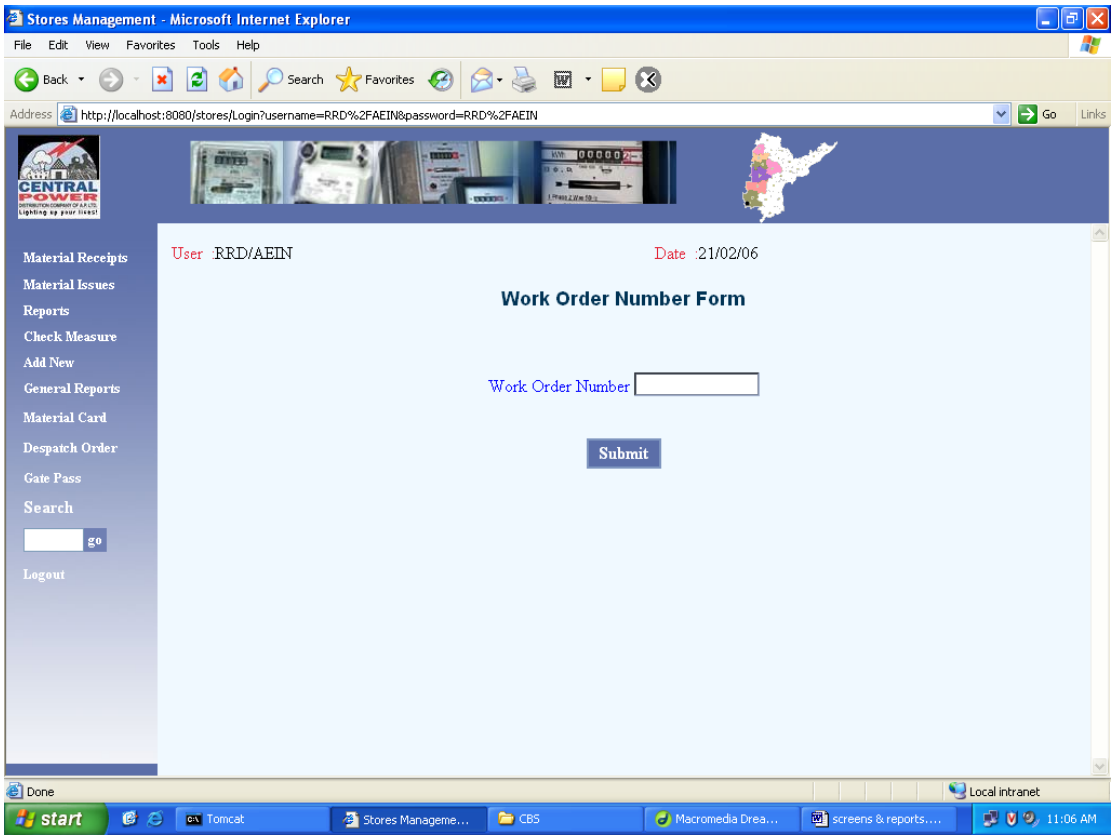
Devolutions

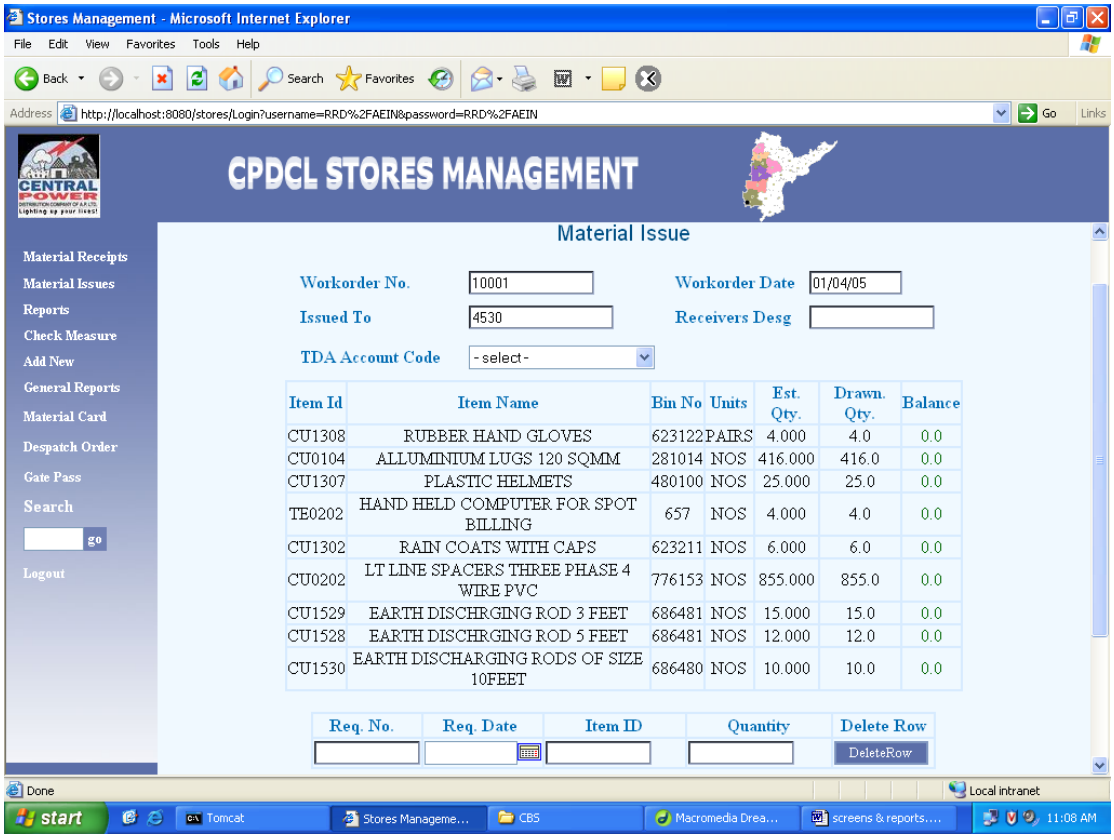


4 Material Issues Module

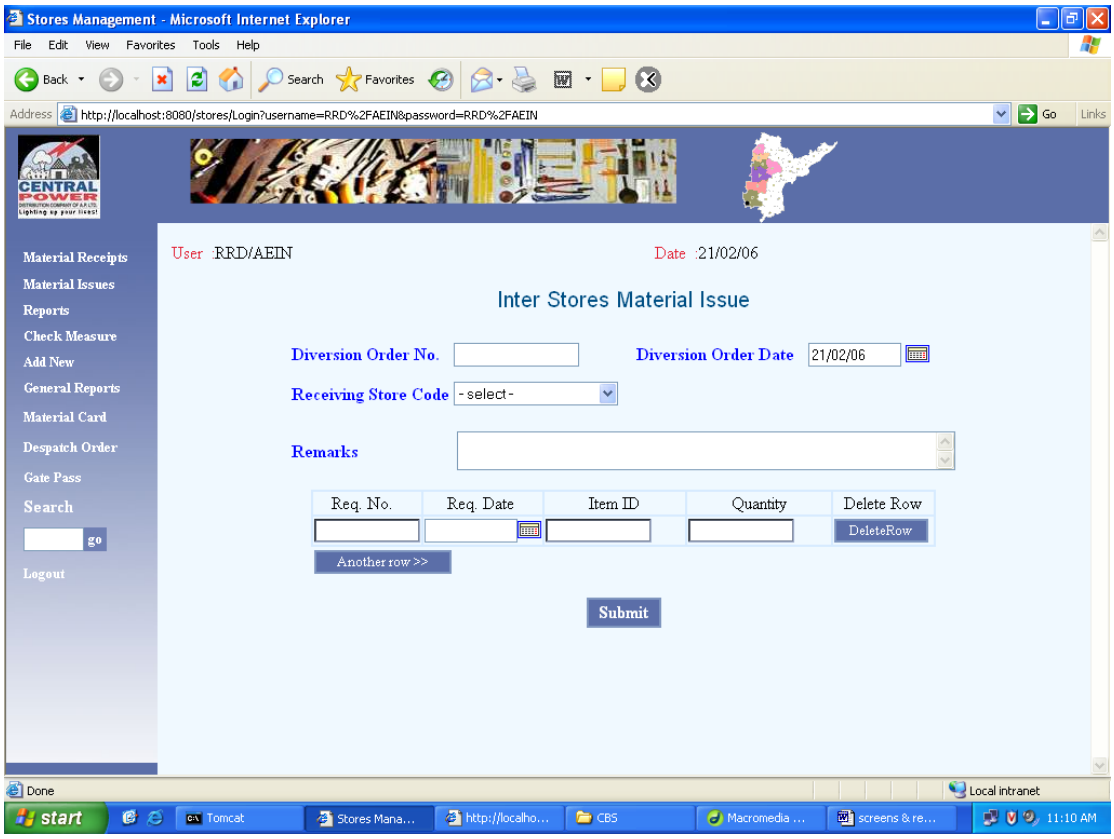


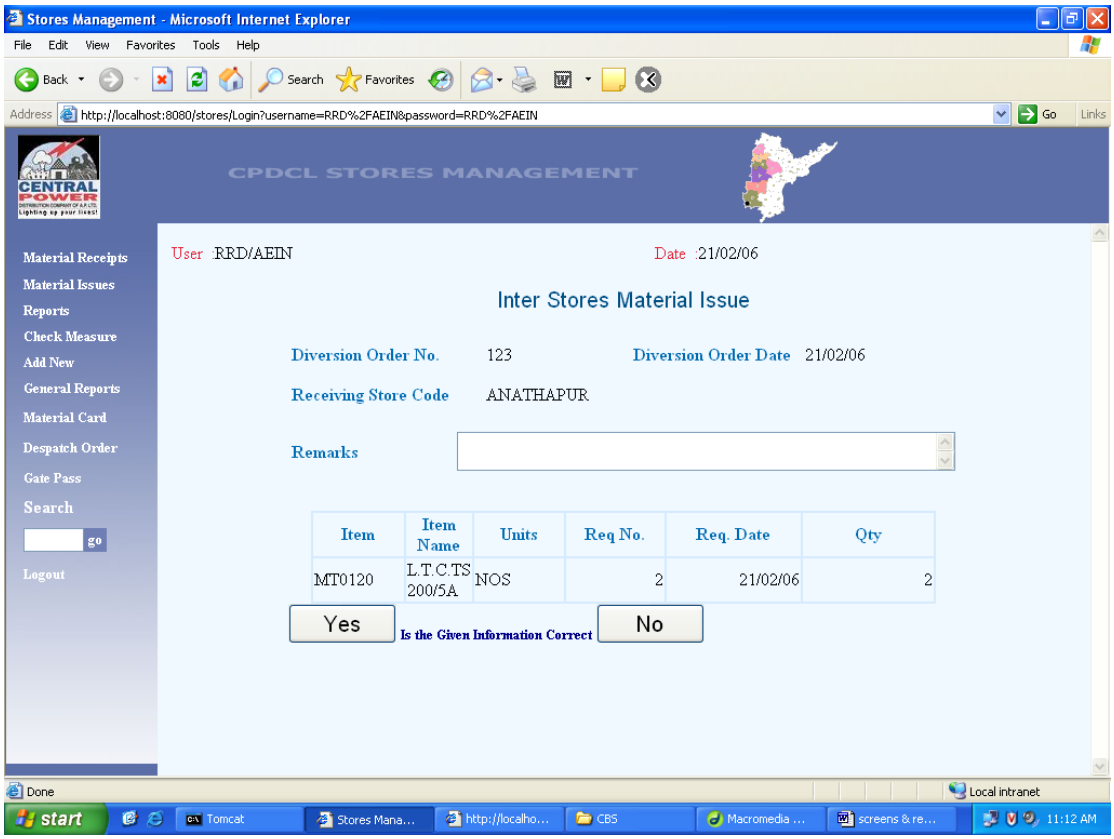
Material Issue



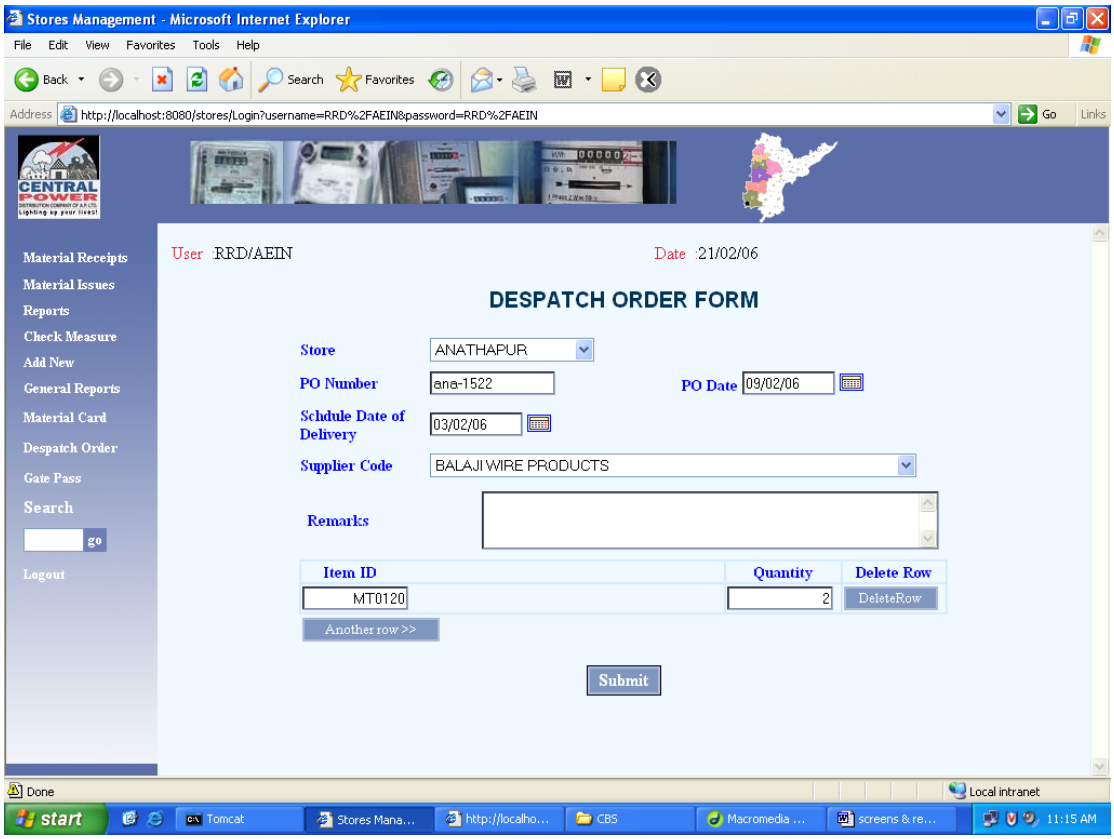


Inter Stores Issues

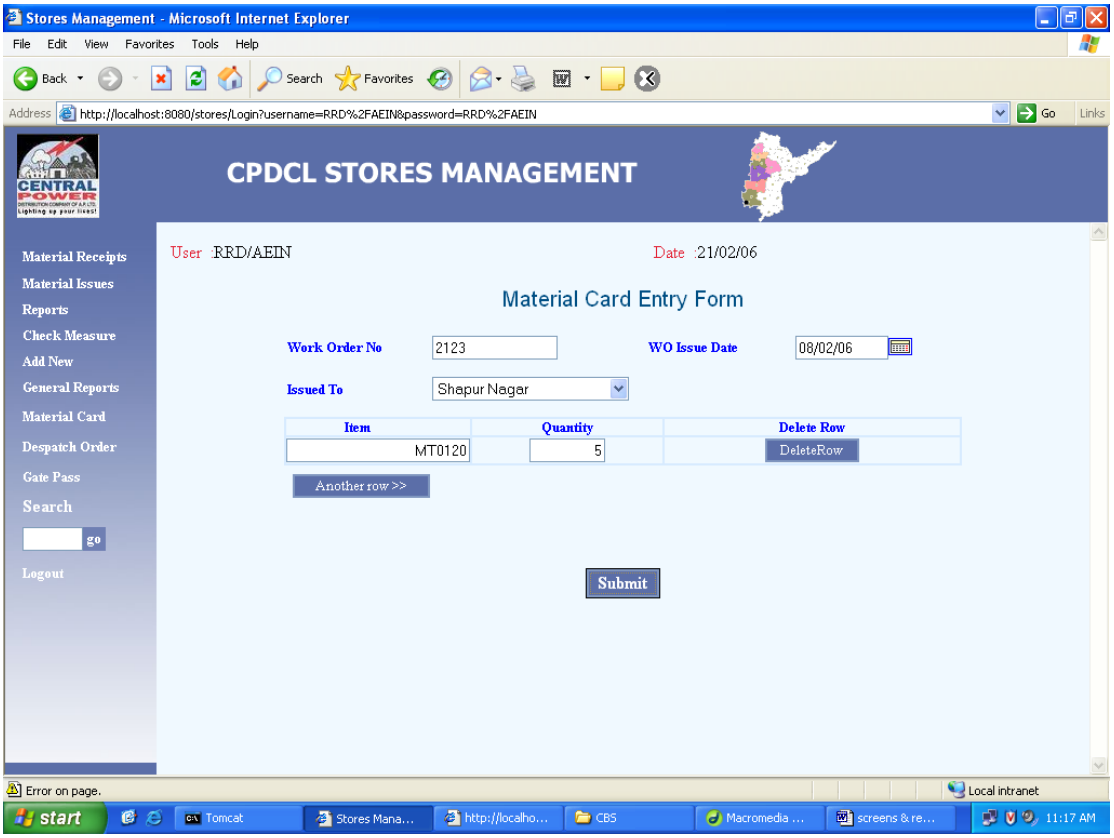




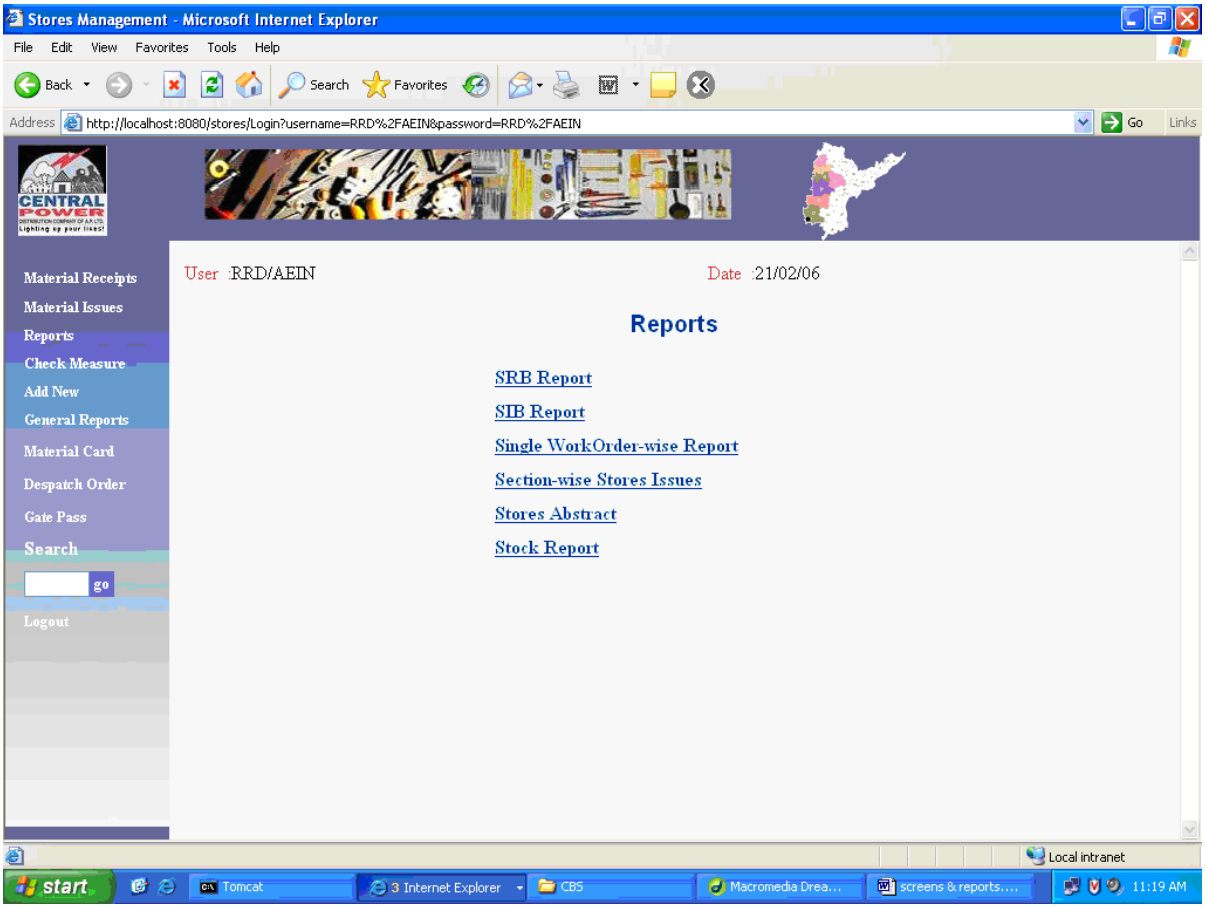
5. Dispatch Order



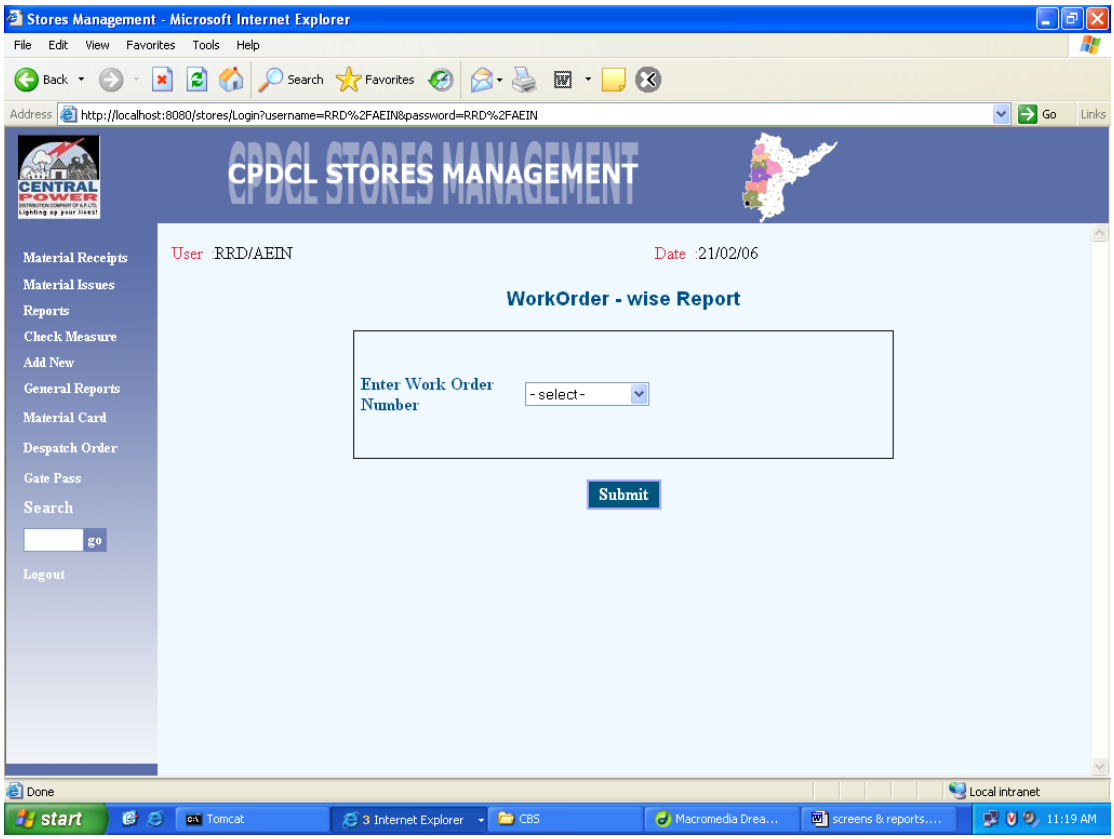
6. Material card

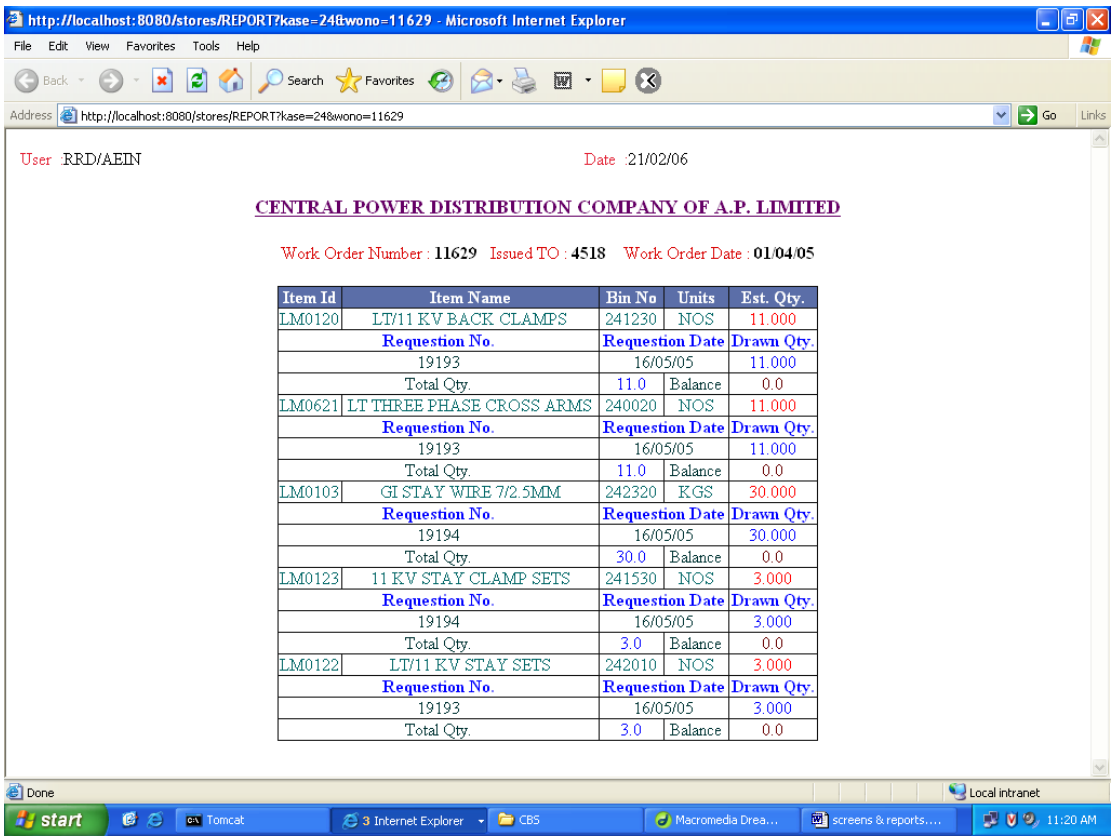


7. Reports


Stores Management - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/stores/Login?username=RRD%2FAEIN&password=RRD%2FAEIN
Go Links
CENTRAL POWER
Material Receipts
Material Issues
Reports
Check Measure
Add New
General Reports
Material Card
Despatch Order
Gate Pass
Search
go
Logout
User :RRD/AEIN
Date :21/02/06
Reports
SRB Report
SIB Report
Single WorkOrder-wise Report
Section-wise Stores Issues
Stores Abstract
Stock Report
Local intranet
start
Tomcat
3 Internet Explorer
CBS
Macromedia Drea...
screens & reports....
11:19 AM

Work Order wise Report





User :RRD/AEIN

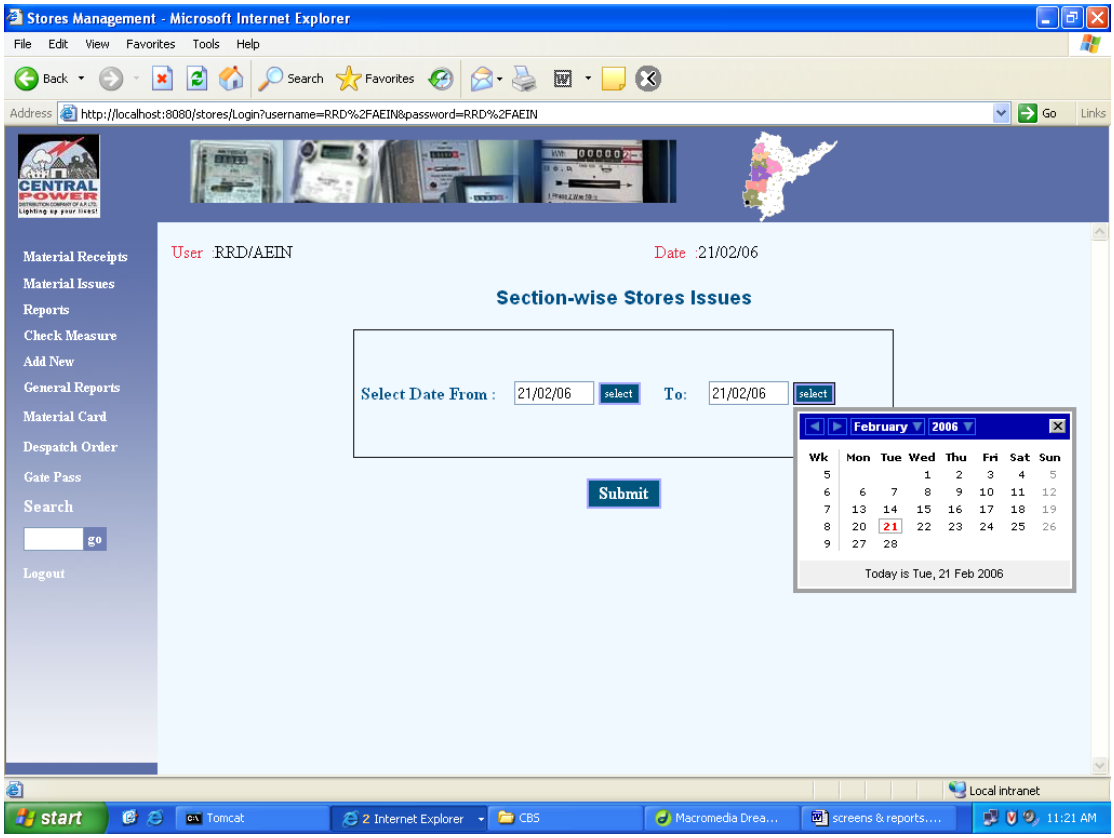
Date :21/02/06

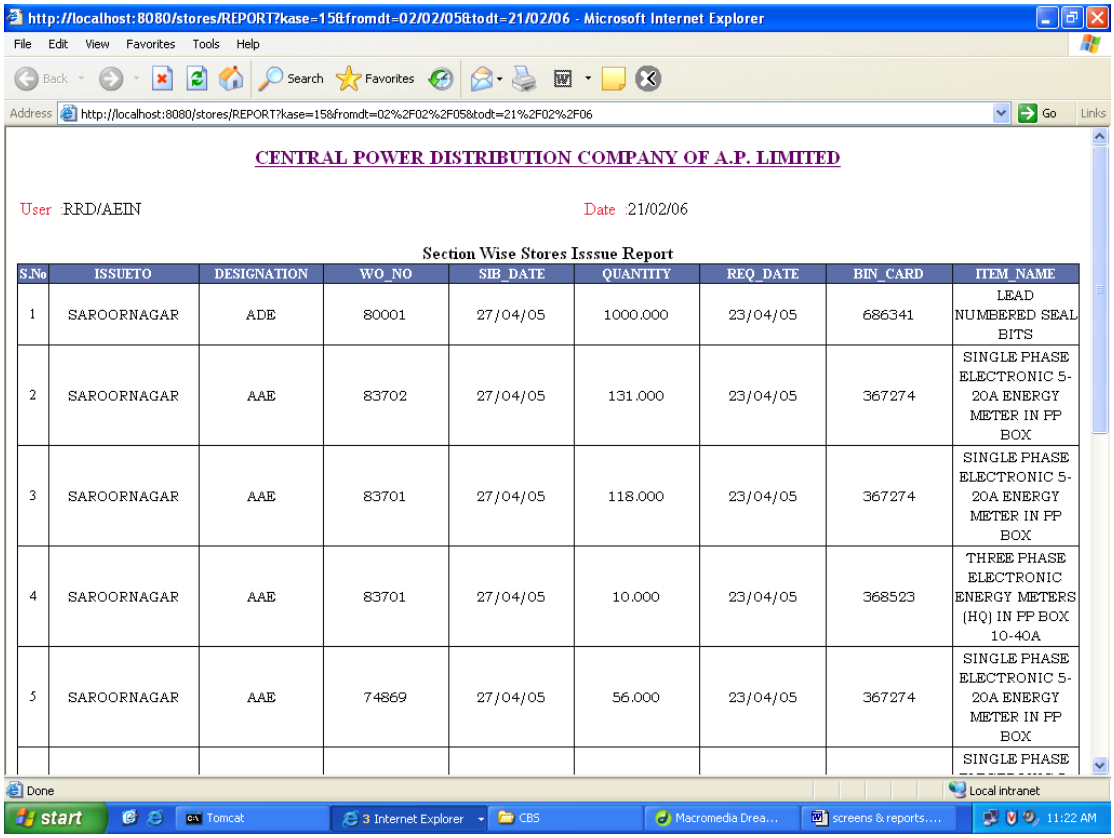
CENTRAL POWER DISTRIBUTION COMPANY OF A.P. LIMITED

Work Order Number : **11629** Issued TO : **4518** Work Order Date : **01/04/05**

Item Id	Item Name	Bin No	Units	Est. Qty.
LM0120	LT/11 KV BACK CLAMPS	241230	NOS	11.000
Requestion No.		**Requestion Date**		**Drawn Qty.**
19193		16/05/05		11.000
Total Qty.		11.0	Balance	0.0
LM0621	LT THREE PHASE CROSS ARMS	240020	NOS	11.000
Requestion No.		**Requestion Date**		**Drawn Qty.**
19193		16/05/05		11.000
Total Qty.		11.0	Balance	0.0
LM0103	GI STAY WIRE 7/2.5MM	242320	KGS	30.000
Requestion No.		**Requestion Date**		**Drawn Qty.**
19194		16/05/05		30.000
Total Qty.		30.0	Balance	0.0
LM0123	11 KV STAY CLAMP SETS	241530	NOS	3.000
Requestion No.		**Requestion Date**		**Drawn Qty.**
19194		16/05/05		3.000
Total Qty.		3.0	Balance	0.0
LM0122	LT/11 KV STAY SETS	242010	NOS	3.000
Requestion No.		**Requestion Date**		**Drawn Qty.**
19193		16/05/05		3.000
Total Qty.		3.0	Balance	0.0

Section wise Stores Issues





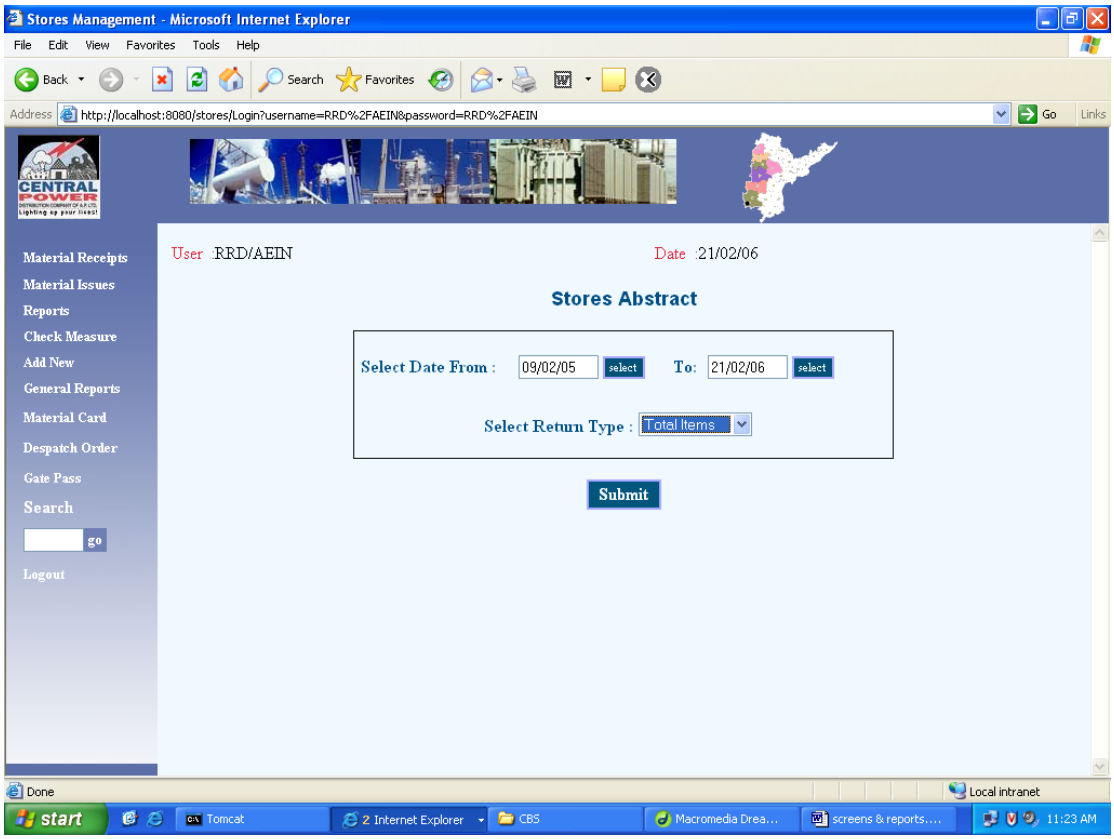
CENTRAL POWER DISTRIBUTION COMPANY OF A.P. LIMITED

User :RRD/AEIN Date :21/02/06

Section Wise Stores Isssue Report

S.No	ISSUETO	DESIGNATION	WO_NO	SIB_DATE	QUANTITY	REQ_DATE	BIN_CARD	ITEM_NAME
1	SAROORNAGAR	ADE	80001	27/04/05	1000.000	23/04/05	686341	LEAD NUMBERED SEAL BITS
2	SAROORNAGAR	AAE	83702	27/04/05	131.000	23/04/05	367274	SINGLE PHASE ELECTRONIC 5-20A ENERGY METER IN PP BOX
3	SAROORNAGAR	AAE	83701	27/04/05	118.000	23/04/05	367274	SINGLE PHASE ELECTRONIC 5-20A ENERGY METER IN PP BOX
4	SAROORNAGAR	AAE	83701	27/04/05	10.000	23/04/05	368523	THREE PHASE ELECTRONIC ENERGY METERS (HQ) IN PP BOX 10-40A
5	SAROORNAGAR	AAE	74869	27/04/05	56.000	23/04/05	367274	SINGLE PHASE ELECTRONIC 5-20A ENERGY METER IN PP BOX
								SINGLE PHASE

Stores abstract



CENTRAL POWER DISTRIBUTION COMPANY OF A.P. LIMITED

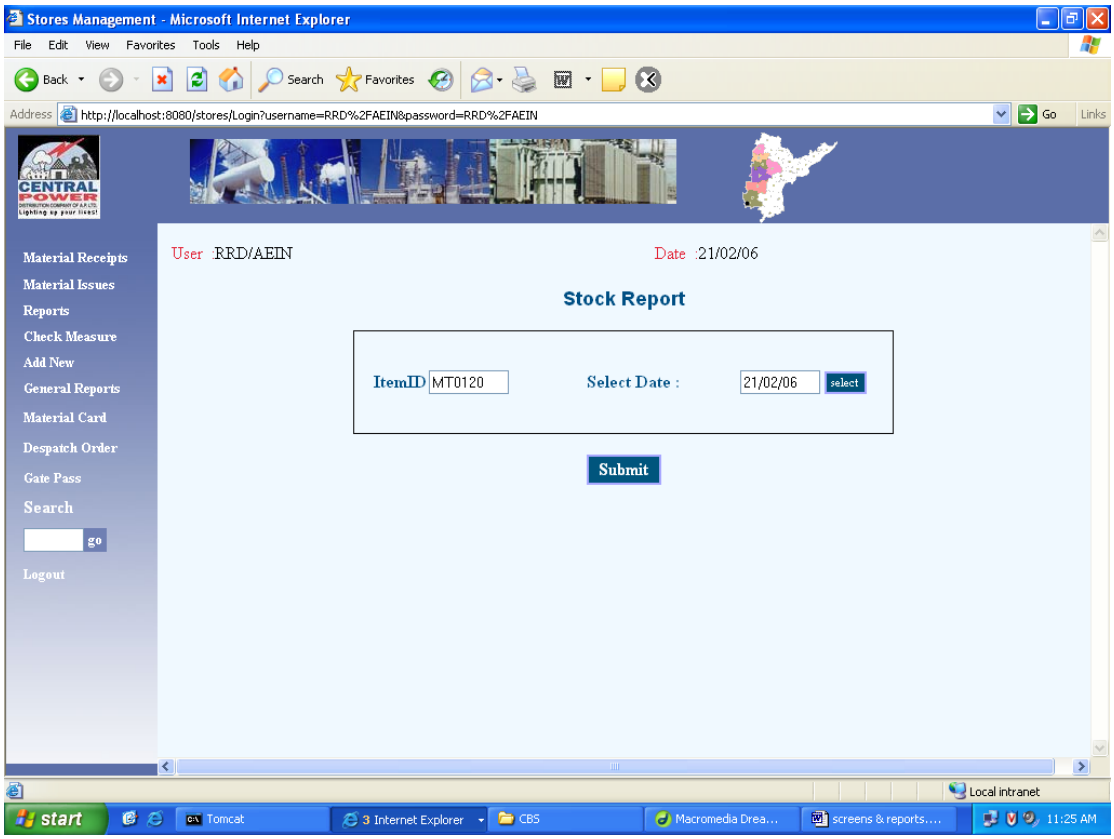
User :RRD/AEIN Date :21/02/06

Strores Abstract Report

javaException...java.sql.SQLException: [Microsoft][ODBC driver for Oracle][Oracle]ORA-01858: a non-numeric character was found where a numeric was expected

ITEM NAME	Bin	UNITS	Opening Stock	RECEIPTS			ISSUES			Closing stock
				Receipts	Devolutions	Inter	Section	inter	without mcard	
SILICAGEL	134	KGS								

Stock Report



CENTRAL POWER DISTRIBUTION COMPANY OF A.P. LIMITED

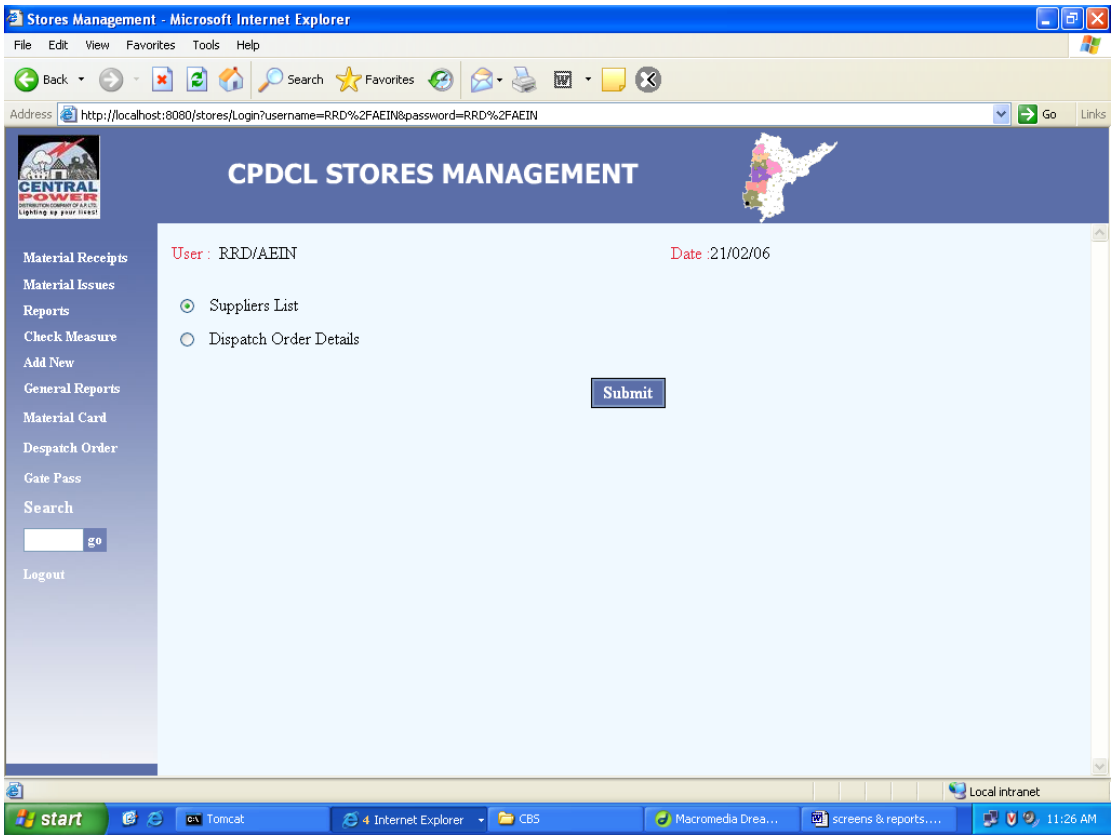
User :RRD/AEIN Date :21/02/06

Stock Report

Sl.No.	ITEM-ID	ITEM NAME	Bin Card	Units	Stock
1	MT0120	L.T.C.TS 200/5A	465	NOS	2862.000

8. General Reports

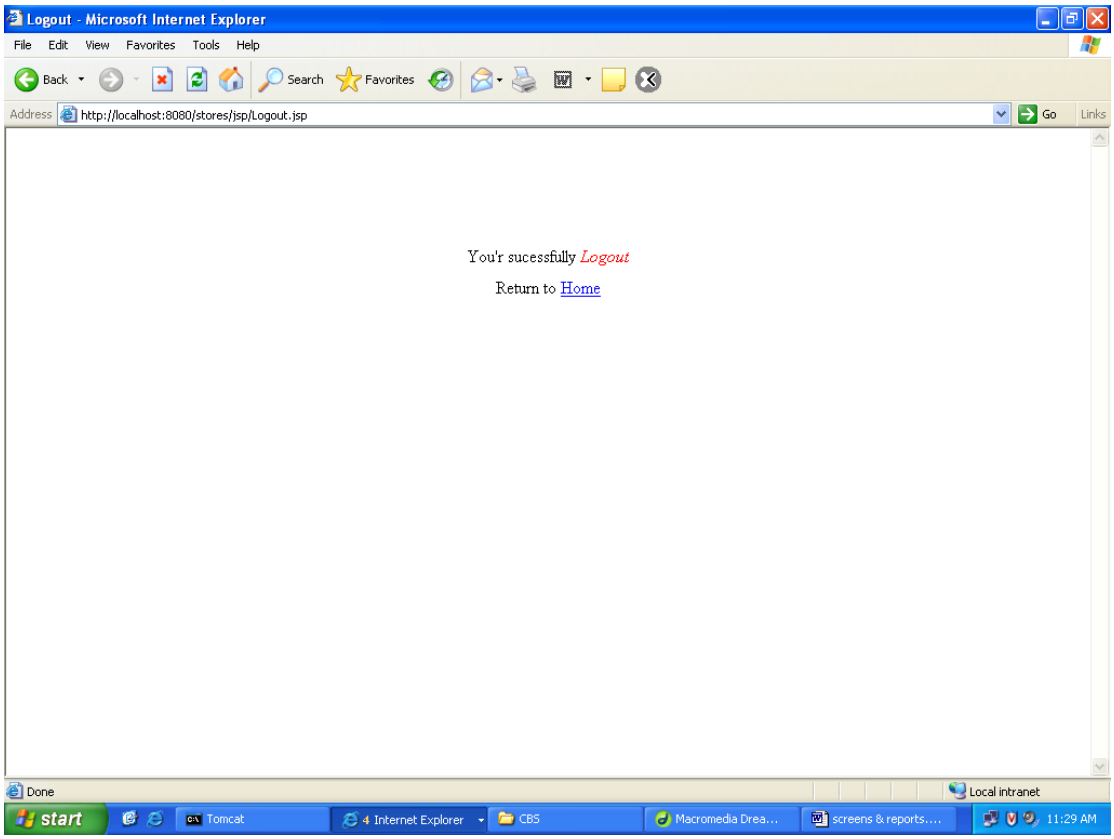
a. Suppliers List



Stores Management - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8080/stores/Login?username=RRD%2FAEIN&password=RRD%2FAEIN
Go Links
CENTRAL POWER
Material Receipts
Material Issues
Reports
Check Measure
Add New
General Reports
Material Card
Despatch Order
Gate Pass
Search
go
Logout
CENTRAL POWER DISTRIBUTION COMPANY OF A.P. LIMITED
User :RRD/AEIN
Date :21/02/06
Supplier Details List
S.No SUPPLIER_ID SUPPLIER_NAME CONTACT_PERSON ADDRESS STREET CITY PHONE FAX
1 C1 VERSETEK SWITCHGEARS (P)LTD null null null HYDERABAD null null
2 C2 INDIAN RAYON AND INDUSTRIES null null null HYDERABAD null null
3 C3 A BOND STRANDS (P) LTD null null null HYDERABAD null null
4 C4 A.P.E.E.C null null null HYDERABAD null null
5 C5 ACCORD ENTERPRISES null null null HYDERABAD null null
6 C6 ACCURATE TRANSFORMERS LTD null null null HYDERABAD null null
7 C7 ACHYUTA ELECTRICALS LTD null null null HYDERABAD null null
ADITYA
Done
Local intranet
start
Tomcat
4 Internet Explorer
CBS
Macromedia Drea...
screens & reports....
11:27 AM

b. Dispatch Order List

9. Logout page



5.2. Database Description:

User

Field	Data type	Constraint
Userid	Varchar2 (50)	Primary key
Password	Varchar2 (50)	
Fname	Varchar2 (50)	
Lname	Varchar2 (50)	
Initials	Varchar2 (50)	
Designation	Varchar2 (20)	
Job	Varchar2 (20)	
Store_code	Number (10)	Foreign key
Timestamp	Varchar2 (100)	
Macaddress	Varchar2 (100)	

Designation

Field	Data type	Constraint
Designation	Number (10)	Primary key
Short_name	Varchar2(20)	
Full_name	Varchar2(50)	

Organization

Field	Data type	Constraint
Organisationid	Varchar2 (10)	Primary key
Org_short_name	Varchar2 (50)	
Org_full_name	Varchar2 (200)	
Address	Varchar2 (100)	
Street	Varchar2 (100)	
City	Varchar2 (100)	

Phone	Varchar2 (50)	
url	Varchar2 (50)	

Log table

Field	Data type	Constraint
Userid	Varchar2(50)	Foreign key
Appname	Varchar2(50)	
Operation	Varchar2(100)	
Date	Date	

Issues

Field	Data type	Constraint
Sib_slno	Varchar2 (20)	Primary key
Sib_number	Varchar2 (20)	
Date	Date	
Receiving_section	Number (10)	
Receivers_name	Varchar2 (150)	
Receivers_designation	Varchar2 (50)	
Tda	Varchar2 (20)	
Work_order_number	Varchar2(100)	
Remarks	Varchar2 (100)	
Store_id	Varchar2(20)	
Userid	Varchar2(50)	Foreign key
Timestamp	Timestamp	
Macadderess	Varchar2(50)	
Reqno	Varchar2(100)	
Reqdate	Date	
Work_order_date	Date	
Scheme	Number(10)	

Issues-items

Field	Data type	Constraint
Sib_slno	Varchar2 (20)	Primary key
Item_id	Varchar2 (20)	
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (20)	Foreign key
Timestamp	Timestamp	
Macadress	Varchar2 (20)	
Reqno	Varchar2(20)	
Reqdate	Date	
Flag	Varchar2(20)	

Issues_corr

Field	Data type	Constraint
Sib_slno	Varchar2 (20)	Primary key
Sib_number	Varchar2 (20)	
Date	Date	
Receiving_section	Number (10)	
Recievies_name	Varchar2 (150)	
Receivers_designation	Varchar2(50)	
Tda	Varchar2 (20)	
Work_order_number	Varchar2(100)	
Remarks	Varchar2 (1000)	
Store_id	Varchar2(20)	
Userid	Varchar2 (50)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2 (50)	
Reqno	Varchar2(100)	
Reqdate	Date	
Work_order_date	Date	

Scheme	Number (10)	
E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macadress	Varchar2(50)	

Issue_item_corr

Field	Data type	Constraint
Sib_slno	Varchar2 (20)	Primary key
Item_id	Varchar2 (20)	
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (50)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2 (50)	
E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macaddress	Varchar2(50)	

Issues_item_corr

Field	Data type	Constraint
Sib_slno	Varchar2 (20)	Primary key
Item_id	Varchar2 (20)	
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (50)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2 (50)	
E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macaddress	Varchar2(50)	

Getpass

Field	Data type	Constraint
Gpslno	Varchar2 (20)	Primary key
Sib_slno	Varchar2 (20)	
Reqno	Varchar2 (100)	
Reqdate	Date	
Item_id	Varchar2 (50)	
Pqty	Double	
Vehicle_no	Varchar2 (50)	
Remarks	Varchar2(500)	
Userid	Varchar2(50)	Foreign key
Timestamp	Varchar2(50)	
Macadress	Varhcar2(50)	

Getpass_diversion

Field	Data type	Constraint
Gpslno	Varchar2 (20)	Primary key
Sib_slno	Varchar2 (20)	
Reqno	Varchar2 (100)	
Reqdate	Date	
Item_id	Varchar2 (50)	
Pqty	Double	
Vehicle_no	Varchar2 (50)	
Remarks	Varchar2(500)	
Userid	Varchar2(50)	Foreign key
Timestamp	Varchar2(50)	
Macadress	Varhcar2(50)	

Sis-Item

Field	Data type	Constraint
Item_id	Varchar2 (20)	Primary key
Item_name	Varchar2 (200)	
Min_stock	Number (10)	
Max_stock	Number (10)	
Units	Varchar2 (50)	
Bin_card	Varchar2 (100)	
Qty	Double	
Ledger_folio	Varchar2(300)	
Major_head	Varchar2(10)	
Submajor_head	Varchar2(10)	
Minor_head	Varhcar2(10)	
Storage	Varchar2 (10)	

Sis-item_new

Field	Data type	Constraint
Item_id	Varchar2 (20)	Primary key
Item_name	Varchar2 (200)	
Min_stock	Number (10)	
Max_stock	Number (10)	
Units	Varchar2 (50)	
Bin_card	Varchar2 (100)	
Qty	Double	
Ledger_folio	Varchar2(300)	
Major_head	Varchar2(10)	
Submajor_head	Varchar2(10)	
Minor_head	Varhcar2(10)	
Storage	Varchar2 (10)	

Sis-item_old

Field	Data type	Constraint
Item_id	Varchar2 (20)	Primary key
Item_name	Varchar2 (200)	
Min_stock	Number (10)	
Max_stock	Number (10)	
Units	Varchar2 (50)	
Bin_card	Varchar2 (100)	
Qty	Double	
Ledger_folio	Varchar2(300)	
Major_head	Varchar2(10)	
Submajor_head	Varchar2(10)	
Minor_head	Varhcar2(10)	
Storage	Varchar2 (10)	

Sis-item_new1

Field	Data type	Constraint
Item_id_new	Varchar2 (20)	Primary key
Item_id_old	Varchar2 (20)	Foreign key
Ledger folio	Varchar2 (15)	
Units	Varchar2 (100)	
Name	Varchar2 (20)	
Stork	Number (15,5)	
Price	Number (20,2)	
Value	Number (25,2)	
Account	Varchar2(20)	
Mhead	Varchar2(5)	
Smhead	Varhcar2(5)	

Sis_diversion_items

Field	Data type	Constraint
Isd_slno	Varchar2 (20)	Primary key
Item_id	Varchar2 (20)	Foreign key
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (20)	
Timestamp	Timestamp	
Macaddress	Varchar2(50)	
Reqno	Varchar2(50)	
Reqdate	Date	
Flag	Varchar2(10)	

Sis_diversion_modifications

Field	Data type	Constraint
Srb_modno	Varchar2 (20)	Primary key
Isd_slno	Varchar2 (20)	Foreign key
Date	Date	
Order_number	Varchar2 (100)	
Storied	Varchar2 (20)	
Receiving_storeid	Varchar2(20)	
Remarks	Varchar2(1000)	
Srb_number	Varchar2(20)	
Userid	Varchar2(20)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2(50)	
Requisition_no	Varchar2(50)	
Requisition_date	Date	
Diversion_order_date	Date	
Flag	Varchar2(10)	

Sis_item_types

Field	Data type	Constraint
Type_id	Varchar2 (20)	
Type_name	Varchar2 (20)	Foreign key

Sis_diversion_items_corr

Field	Data type	Constraint
Isd_slno	Varchar2 (20)	
Item_id	Varchar2 (20)	Foreign key
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (50)	
Timestamp	Timestamp	
Macaddress	Varchar2(50)	
E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macaddress	Varchar2(50)	

Sis_diversion_corr

Field	Data type	Constraint
Isd_slno	Varchar2 (20)	
Date	Date	
Order_number	Varchar2 (100)	
Storied	Varchar2 (20)	
Receiving_storied	Varchar2 (20)	
Remarks	Varchar2(1000)	
Srb_number	Varchar2(20)	
Userid	Varchar2(50)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2(50)	

E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macaddress	Varchar2(50)	
Requisition_no	Varchar2(50)	
Requisition_date	Date	
Division_order_date	Date	

Si_devolution_item2_corr

Field	Data type	Constraint
Dev_slno	Varchar2 (20)	Primary key
Item_id	Varchar2(20)	Foreign key
Quantity	Number (10)	
Price	Number (10)	
Userid	Varchar2 (50)	Foreign key
Timestamp	Timestamp	
Macaddress	Varchar2(50)	
E_userid	Varchar2(50)	
E_timestamp	Timestamp	
E_macaddress	Varchar2(50)	

Form 13

Field	Data type	Constraint
Slno	Varchar2 (20)	Primary key
Srb_slno	Varchar2(20)	
Status	Varchar 2(1)	
Timestamp	Timestamp	

Updates

Field	Data type	Constraint
Storied	Varchar2 (20)	Primary key
Client_time	Timestamp	
Server_time	Varchar2(20)	
Hwaddress	Varchar2(20)	
Userid	Varchar2(50)	Foreign key

Tda

Field	Data type	Constraint
Tda_slno	Varchar2 (20)	Primary key
Branched	Varchar2(20)	
Branch_name	Varchar2(20)	
User_id	Varchar2(50)	Foreign key
Timestamp	Varchar2(50)	
Macaddress	Varchar2(50)	

Pricing

Field	Data type	Constraint
Ledger_folio	Varchar2 (20)	
Stock	Varchar2(20)	
Value	Number (20,2)	

Sis_price_list

Field	Data type	Constraint
Pl_slno	Number (10)	
Branched	Varchar2(20)	Foreign key
Date	Date	
Item_id	Varchar2(20)	Foreign key

Price	Number (20,2)	
Stock	Number (20,5)	
Table_name	Varchar2(50)	
Row_id	Varchar2(20)	

SYSTEM TESTING

6. System Testing:

The development of Software system involves a series of production activities. There is a chance of errors to occur at any stage. Because of human inability to perform and communicate with perfection, a Quality Assurance Activity accompanies software development.

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

The increasing visibility of software as a system element and the costs associated with software failure are motivating forces for well planned, thorough testing.

For testing the system we followed the strategies given below.

Testing Techniques:

Different types of testing are

- Boundary Condition Testing
- Integration Testing
- Black Box Testing
- Validation Testing
- User Acceptance Testing

During the implementation for the system each module of the system is tested separately to uncover errors within its boundaries. User interface is used as a guide in this process. The validations have been done for all the inputs using Java Script.

For example to check whether the work allotted among the database correctly without exceeding the schemes which are not validated thoroughly and the internal database has to check the reflections in the database.

Boundary conditions Test:

Boundary conditions as in case of generating sequences were tested to ensure that the module operates properly at boundaries establish to limit or restrict processing also it is able to handle incorrect out of the boundary values properly.

Integration Test:

The objective of Integration Test is to take the until tested modules and build a program structure that has been defined in the design. We have done top down integration, which is constructing and testing small segments where errors are easier to isolate, and correct. The integration process was performed in three steps:

- The main control was used as test driver.
- Test was conducted as each module was integrated.
- Regression testing to ensure that new errors have not been introduced due to the corrections.

Block Box Testing:

It focuses on functional requirements of the software. Block box testing attempts to find errors in the following categories.

Incorrect or missing function
Interface error
Errors in external device access
Performance error
Initialization and termination errors

The above testing was successfully carried out for the developed system.

Validation Testing:

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests namely validation tests are performed. Validation succeeds when the software functions in the manner that can be easily accepted by the customer.

After validation test has been conducted, one of the possible condition is satisfied. The functions or performance characteristics confirmed to specifications are acceptable. The deviation form specifications are uncovered and a note of what is lacking is made. The developed system has been tested satisfactorily to ensure its performance is satisfactory and it is working efficiently.

User Acceptance Testing:

User acceptance of a system is a key factor for the success of any system. The system under consideration was tested for user acceptance constantly, by keeping the users informed of the progress and incorporating changes suggested, at the development time itself.

Test Case Report:

Here we specify all the test cases that are used for system testing. The different conditions that need to be tested along with the test case used for testing those conditions and the expected outputs are given. The goal is to test the different functional requirements. Test cases have been selected for both valid and invalid inputs.

S.No	Test case	Condition	Expected Output
1	Get Systems	Input Domain name	Print list of all system in current domains & response time
2	Get User	Input Domain name	System id, user id, port no, domain name
3	Get Processes details	Select process	Output the details of processes
4	Get modules details	Select process & select thread opt	Details of modules
5	Get thread details	Select process & select thread opt	Details of threads
6	Stop the processes	System id, user id, password	Process close
7	Stop the system	System id	System close

Testing Analysis:

S.No	Testing object	Expected value	Simulated value	Explanation	Remarks
1	User name & Password	AEIND GUEST	AEIND GUEST	Equal of expected and simulated values	Pass
2	User name & Password	AEIND GUEST	AEIND GUEST	Unequal of expected and simulated value	Fail
3	Change password	GUESS	GUEST (Old password)	Equal of these two passwords	Pass
4	Start time and end time	11:03:05	11:03:05	Equal of these times	Pass
5	Start date and end date	02/01/2006	02/01/2006	Equal of these dates	Pass

SCOPE OF THE PROJECT

7. Scope Of The Project:

Stores Management System :

The proposed system is ‘Stores Management System ’. This system is GUI based system and is user friendly. Stores Management System is accessible through the internet. Stores are required for the following purposes.

1. Capital works
2. Operation and Maintenances Works
3. Other Commercial activities like hiring equipment etc..,

The ‘Stores Management System package’ is targeted to automate the almost all of the processes mentioned above to reduce the clerical labour of the staff working in Stores both technical and as well as Accounts departments using the software industry’s latest technologies and cost effective tools there by providing the better control to the management by avoiding manual errors etc..,

This application discusses the initial screens like login, logout and main menu of the application. The application can be accessed by typing the URL in the address bar of the web browser like Internet Explorer. The URL is of the format http://localhost:8080/contextname on which it will invoke login screen.

The stores operations as per the stores manual have been described as two different processors owing to the complexity and tediousness of the operations by effective division of labour as physical receipts and issue has been the responsibility of Accounts section.

To solve the problem using computer systems for which the complexity and tediousness is not an issue, the pricing is calculated at the time of receiving the materials itself and stock price will be fixed then and there and fetching that price while issuing the material.

The software package is described in detailed below. The package provides three basic functionalities

1. Material Receipts Module
2. Material Issues Module

3. Different Reports serving the Quantity ledgers, Priced ledgers and material indexing and ledger folio indexing etc.

In this project Issues module and Reports module have been handled.

7.1. Issues Module:

The Issues Module deals with the issues functionality of the application. In this Module, there are different operations performed under issues function.

1. Material Issues
2. Inter Stores Issues
3. Gate Pass Generation

1. Material Issues:

Material Issues to works based on field requisitions.

In this New Material Issues, it has two parts. The top portion is called Header area and bottom portion is called detailed area. Enter the data in all fields including detail area that is item details. After entering the data, press Submit button. If we try to submit with out filling any field, appropriate error message indicating the same will be thrown. Facilities are there to enter multiple items for a requisition by pressing the 'Add_Item' button in the detail area. Provision is also there to enter multiple requisitions in an Issue by pressing the 'Another Requisition' button in the bottom of the screen. On pressing the Submit button a screen will be displayed which gives a message 'Is Given Information Correct'. If you press 'Yes' button a Print screen is shown & if you press 'No' button then again it will enter into material issues form.

After printing this page, these transactions will be received by another user.

2. Inter Stores Issues:

Material is issued to other stores based on Inter store Requisitions.

In this Inter Stores Issues, for processing ‘Inter Stores Issues’ truncation we have to select the ‘Inter Stores Issues’ link of the Issues Menu. The processing will be same as explained in the above with slight modification in header area.

3. Gate Pass Generation:

By selecting the ‘Material Issues Gate Pass’ link in the Issues main menu page you will be taken to selection of gate pass page. To issues a gate pass for the Issue we have to select edit link of that Issue.

By checking the required material you want, send in a particular vehicle by selecting the check boxes against those materials and by providing the vehicle no in the text field provided and submitting the page by pressing the ‘Submit’ button.

If you press the Issue button then gate pass process will be generated & printable gate pass will be displayed. If you press the cancel button the gate pass generation will be canceled.

7.2. Reports module:

In this module, we will discuss about various Reports provided by the Application. There are different reports forms in this module.

There are

1. Monthly SRB Report
2. Monthly SIB Report
3. Single Work Order Wise Report
4. Section Wise Stores Issues
5. Monthly Stores Abstract
6. Stock Report

1. Monthly SRB Report:

This Report will display the Monthly Receipts i.e. all New Material Reports. Devolutions from works and Receipts from other stores will be available. Selection of monthly SRB Report will take you to the following selection screen to enable you to specify the year, starting month and ending month for which the details are required.

2. **Monthly SIB report:**

This Report displays the details like slno, sibno, requisition no…etc of the material issues i.e. both issues to the field, issues to the other stores.

3. Single Work Order Wise Report:

This report will display the details of the transaction i.e., devolutions and Issues grouped on work orders no and accounts code (TDA code).

4. Section Wise Stores Issues:

This report will display the issues details, for the month /month's specified initial selection screen of the report, grouped on DA wise i.e. Accounts code wise.

5. Monthly Stores Abstract:

This report will also provide the same details of the items as that of Monthly Pricing Ledger except abstract values of quantities and values of all receipts and all issues for each material with in the specified dates and opening balances and closing balances.

6. Stock Report:

This Report will display the stock balances i.e., closing balances of all materials at the end of the selected month or present day's closing balance if the selected month is current month.

CONCLUSION

8. Conclusion:

We have been given the problem of automating the material of stores in **APCPDCL (Andhra Pradesh Central Power Distribution Company Limited)** at **APTRANSCO**. Earlier the materials of stores have only been automated. In our project **"STORES MANAGEMENT SYSTEM "** we have automated the stores of the central power distribution company limited.

In Issues module, we have tracked the information regarding the issues of the receipts for the material that have been stored in the stores of the central power distribution company limited at **CPDCL**. This information can be easily made available to all the stores throughout the AP through intranet/ Internet.

As a result of this automation, manual workload is reduced and data retrieval becomes easy. This project can be helpful for centralization of information regarding the stores of **APCPDCL** at **APTRANSCO**.

BIBLIOGRAPHY

9. Bibliography

1. Database Management System

   -C.J.Data

2.JAVA2 The complete Reference

   -Patrick Naughton

3. Pure Java Script

   -R.Allen Wyke & Charlton Ting

1. Hyper Text Markup Language
2. java.sun.com/products/**jsp/**
3. **www.javascript.**com
4. Jakarta.**apache.**org/**tomcat**
5. java.sun.com/**j2ee**
6. www.w3schools.com