P-1450

CARITOR ANALYSIS AND REPORTING TOOL

By

BRAHADEESWARAN. N

Reg. No. 71202621010

Of

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

K C T LIBRARY COIMBATORE

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

Of

MASTER OF COMPUTER APPLICATION

June, 2005

BONAFIDE CERTIFICATE

Certified that this project report titled **Caritor Analysis and Reporting Tool** is the bonafide work of **Mr. Brahadeeswaran.N** who carried out the research work under my supervision. Certified further, that to the best part of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

18/5/05

Faculty Guide

Head of the Department

Submitted for the viva-voce examination held on 24.06.2005

Internal Examiner

External Examiner

24/6/05

ABSTRACT

Caritor Analysis and Reporting Tool is about effort tracking and report generation in the organization. The current system tracks the effort spent on each client request thus giving the cumulative effort spent at any point of time. Since, the total effort spent till now on each request is only available, daily effort spent on each request remains unknown. Currently, the organization is forced to generate reports daily and compare it with the previous day's report to track daily effort. All the client information is generated as a single report irrespective of the current status or the age or the severity level of a particular request. Being a stand-alone application system and as the reports generated are editable, it leads to security issues.

Thus, the finding from the existing system is to develop a system, which will track the monthly effort spent; application specific effort spent and the effort spent on each request, dynamically. It has to generate user-friendly reports and have restricted access to solve security issues both to the system and generated reports. Analysis of the generated reports must reflect the trend of the clients' requests, increase/decrease in the number of client requests and the effort put on each of these requests each month.

Caritor Analysis and reporting Tool (CART) is a web-based tool that can be used to track day-wise effort spent by employees of the organization on

various states. Various reports can be generated, depending on the current status of each request, based on application specific effort and request specific effort spent and based on how old each of the current requests are. All generated reports are write-protected Microsoft Excel files. Access to CART is restricted to selected employees of the organization.

ACKNOWLEDGEMENT

I am extremely indebted to our Chairman, **Dr. N. Mahalingam** for providing excellent environment and infrastructure at Kumaraguru College of Technology, Coimbatore. I thank our Correspondent, **Dr. K. Arumugam** and our Principal **Dr. K.K. Padmanabhan** for their moral support.

I convey my sincere regards to our HOD, **Dr. S. Thangasamy**, our Course Coordinator **Prof. A. Muthukumar** and the faculty members for their support. My humble regards to my Project Guide **Ms. V. Geetha** for her valuable guidance, motivation & encouragement in making the project a successful one.

I owe a debt of gratitude to **Mr. Mani Subramanian**, CEO, **Mr. Narendran Chandrasekran**, Project Leader and **Mr. Binu Peter**, Project Manager (Caritor India Pvt. Ltd.) and all other Caritorians for allowing me to use the facilities of the organization and providing me all information and suggestions at all stages of the project, which enabled me to complete it in time. I extend my sincere thanks to my **Parents** and **Friends** who helped me in my

S. NO. TABLE OF CONTENTS PAGE NO.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

1.	BLST	- Baseline Support Tool
2.	CART	- Caritor Analysis and Reporting Tool
3.	DAO	- Data Access Object
4.	DB	- Data Base
5.	ER	- Entity Relationship
6.	FK	- Foreign Key
7.	GB	- Giga Bytes
8.	HHD	- Hard Disk Drive
9.	HTML	- Hyper Text Markup Language
10.	ID	- Identifier/Identification
11.	IDE	- Integrated Development Environment
12.	JSP	- Java Server Page
13.	J2SDK	- Java 2 Servlet Development Kit
14.	PK	- Primary Key
15.	RAM	- Random Access Memory
16.	SERV REQ	- Service Request
17.	SRID	- Service Request Identifier
18.	MB	- Mega Bytes
19.	MER	- Monthly Effort Report
20.	MSR	- Monthly Status Report
21.	MVC	- Model View Controller
22.	URL	- Uniform Resource Locator

CHAPTER 1

INTRODUCTION

1.1 ORGANIZATION PROFILE

Caritor (earlier known as 'IT Solutions') was founded as a U.S. Corporation in the San Francisco Bay Area, California on January 15, 1993 by Mani Subramanian, the Founder-Chairman and CEO. Caritor (incorporated and head quartered in the USA) is a global IT consulting company with operations in the USA, UK, Asia Pacific, and South Africa with Offshore Development Centers in India at Bangalore and Chennai. The revenue for the fiscal year 2004 was US $87 Million with 2250+ employees spread across the globe.

With mature and well-tested delivery processes, Caritor offers solutions to its customers through an onsite-offshore model. The people, processes and technology are certified by various bodies including CMMI Level 5, SEI CMM Level 5, pCMM Level 5, and BS: 7799 (Non-Disclosure of data and information) and ISO 9001 assuring guaranteed value for their customers. Caritor specializes in the following areas:

- IT Consulting, Systems Integration
- Custom Application Development & Maintenance
- Packaged Software Support and Installation
- Software Product Development

1.2 PROBLEM DEFINITION

The existing application in use updates the existing effort spent on the tickets (client requests or split-up of client requests) and does not capture the daily effort spent on those tickets. So, the day-wise effort spent on a ticket by an employee or employees is lost. As a single product category can have many tickets, this system does not provide the over-all effort spent on those products. All request information is generated as a single report and does not provide the status-wise split-up of the service request or the age of requests. The report also includes those requests that were serviced long back which are not required after a certain period of time. The generated report is editable thus questioning the credibility of the report. Also the current system is a stand-alone application, which does not allow remote access for its users.

CHAPTER 2

SYSTEM ANALYSIS

System analysis is the detailed investigation of the component parts of a whole system and their relations in making up the whole. System analysis was conducted to identify the various components of the proposed system and the manner in which these components would communicate with each other. The lists of documents that need to be generated during the analysis phase are:

- Functional Specification Document
- Detailed Design Document
- Unit Test Plan

2.1 EXISTING SYSTEM

The existing tool, Baseline Support Tool (BLST) is being used currently by the organization to maintain the details of the service requests raised by the client/customer. On receipt of every service request, a ticket (client request or split-up of client request) corresponding to that request is created which is further used to track the details of that service request. Later, an employee may be assigned to work on a single service request or more than one employee may work on a single request. This system gives the cumulative effort spent by employees on each request received from the client. All request

the requests or the age of the requests. Effort report that is generated by this system gives the total effort spent on the tickets irrespective of its product type or the level of severity. The Project Manager and Account Manager of any team in the organization can access this tool.

2.2 PROPOSED SYSTEM

Caritor Analysis and Reporting Tool (CART) is a web-application. It is used to track the age and day-wise effort spent on the service requests received from the client. The tool also analyses the status of the client requests and generates various reports like Project Aging Report, Ticket Report, Weekly Status Report and Monthly Status Report.

Project Aging Report gives the client request's age in different time slabs and its product-wise summary report. Weekly Status Report analyses the client requests and generates the report in slabs like the requests that are Completed, In Progress and On Wait for the previous week. Monthly Status Report analyses the client requests and generates the report in slabs like the requests that are Completed, In Progress and On Wait for the previous month along with the statistical report for that month.

CART provides the facility for creating and deleting the details of the client requests in the database. It also provides facility to discard the tracked data from the database at frequent intervals. The effort spent on the client requests are tracked daily and Effort Reports like Monthly Effort Reports and Effort Summary Report are generated. Monthly Effort Reports gives the day-wise effort spent on the client requests and Effort Summary Report gives the

CART provides the user with the facility to compare two reports and generate the Analysis Report. This generated report also includes a graphical representation of the comparison. The files that are to be compared can be uploaded to the server from the client machine itself.

CART generates all the reports as read-only Microsoft Excel files. All the generated reports can be viewed from the client machine. The tool also provides an option for downloading the generated report from the server. Role based access facility of the tool allows the user to access the application based on his role.

2.3 DESIGN ISSUES:

- The tool must provide provision for role-based access
- The tool must provide facility to track daily effort spent on client requests
- Option to generate Effort Summary Report giving the effort spent on requests product-wise
- The tool must provide the provision to generate Effort Reports
- The tool must provide options to upload two reports and generate the Analysis Report for those reports.
- Aging Report, Ticket Report must provide the client requests details based on the age of the requests
- Weekly Status Report and Monthly Status Report must provide the status-wise client requests details
- Monthly Status Report must provide a statistical report too
- The Tool must provide help for users

2.4 USER-INTERFACE REQUIREMENTS:

- The tool provides a very simple user interface to navigate for end-users
- It is designed to suit all sort of user from experts to beginners
- The user interface strikes the right balance between simplicity and sophistication
- New users to the tool can access the help feature to learn about the tool and the use of the system
- User is provided with the option to choose the role if he plays more than one role in the organization
- The tool provides a very simple interface to generate reports
- On selection of a single menu option any report can be generated
- Options to view/download the generated report is provided to the end-users
- All the generated reports are write-protected.

2.5 CONTEXT LEVEL DIAGRAM:

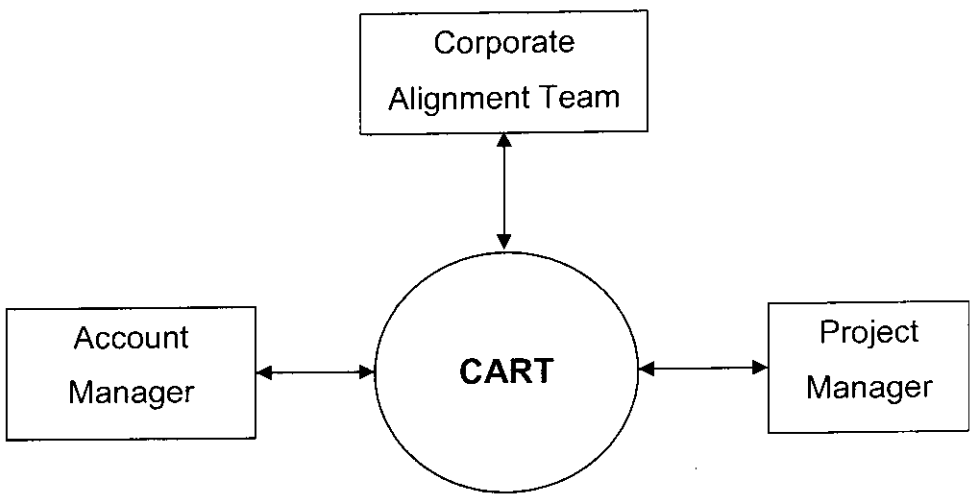


Figure 2.1 – Context Level Diagram

2.6 DATA FLOW DIAGRAMS:

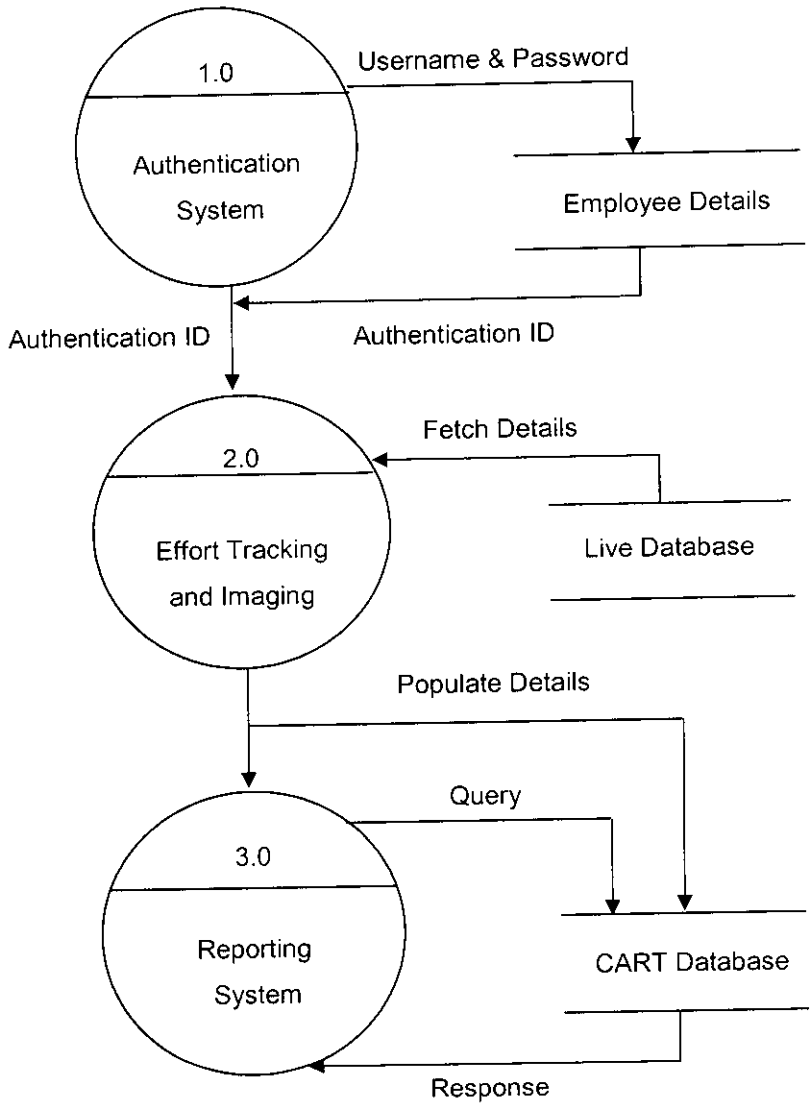


Figure 2.2 – Data Flow Diagram Level 1

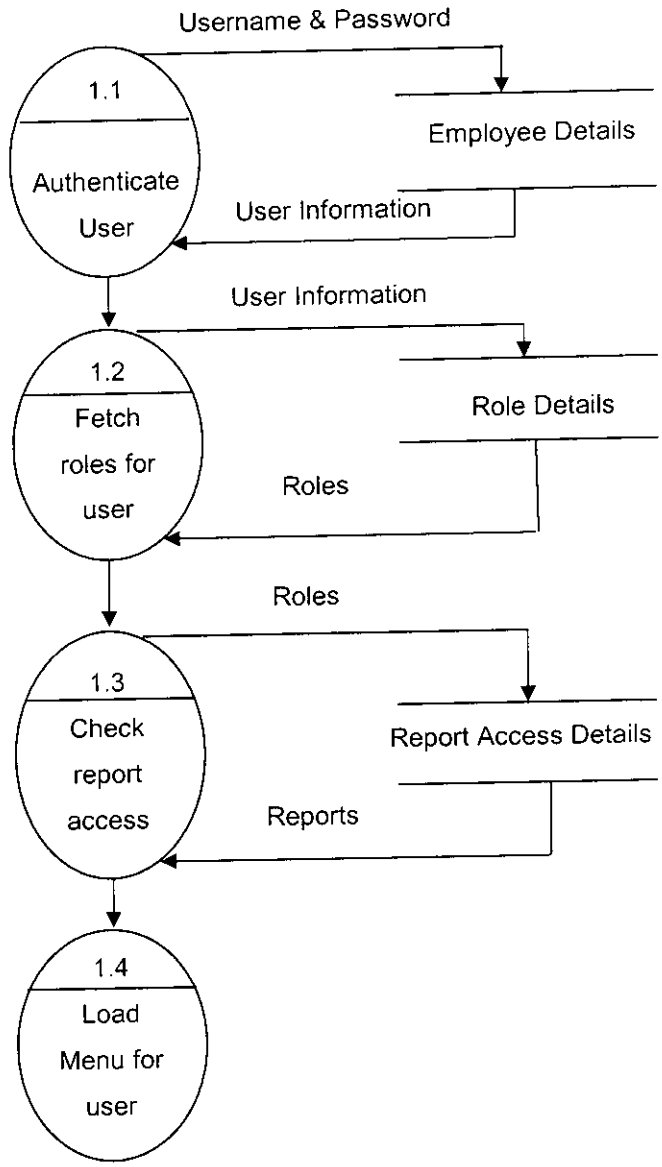


Figure 2.3 – Data Flow Diagram Level 2 (Authentication Process)

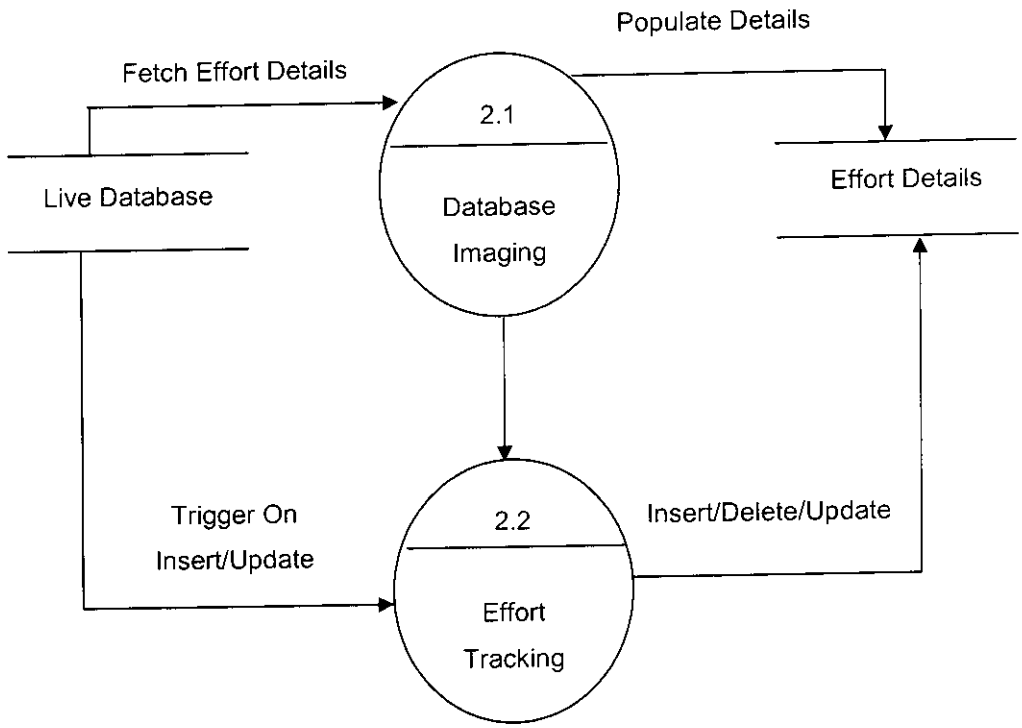
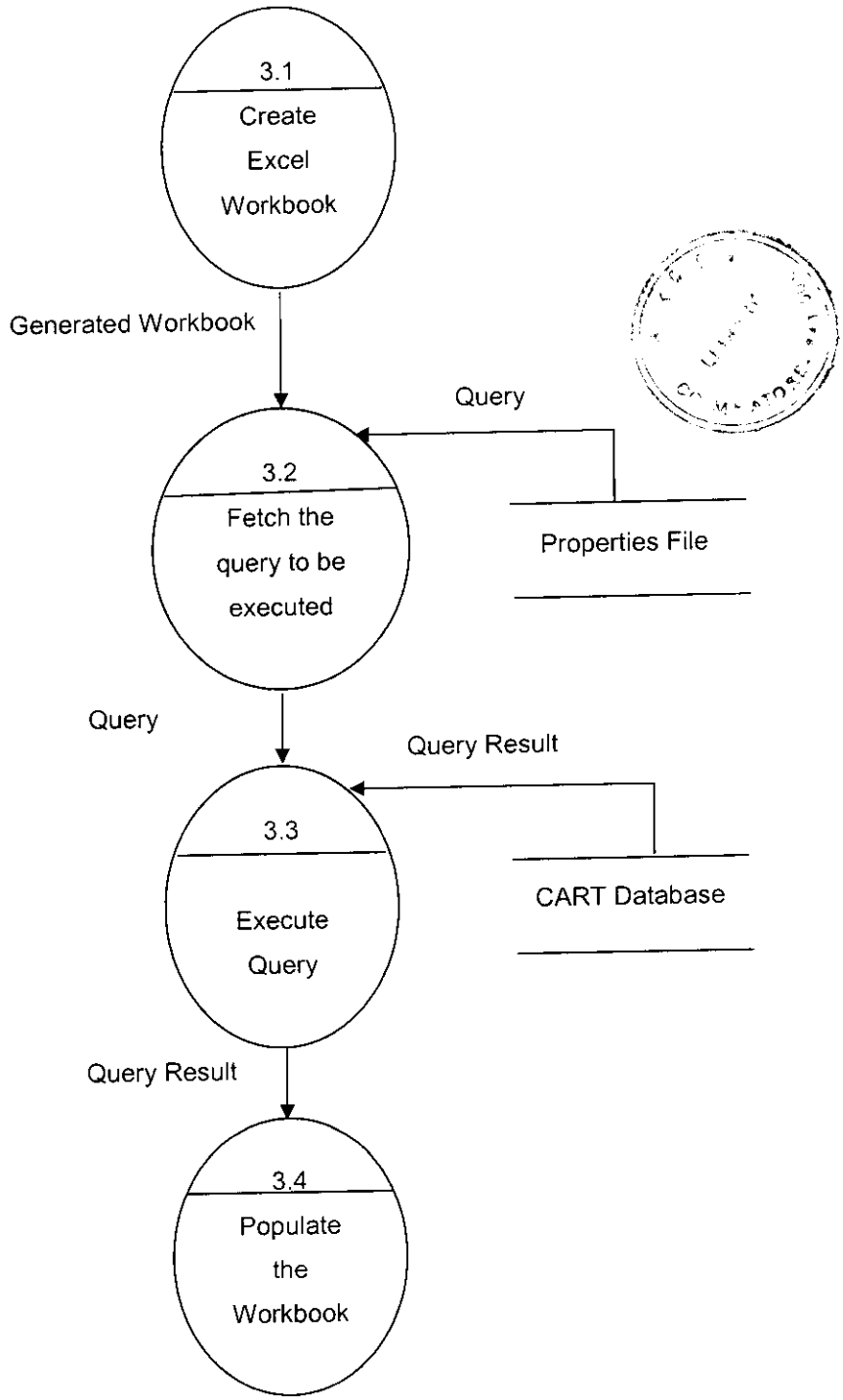


Figure 2.4 – Data Flow Diagram Level 2 (Effort Tracking and Imaging)


3.1
Create
Excel
Workbook
Generated Workbook
Query
3.2
Fetch the
query to be
executed
Properties File
Query
Query Result
3.3
Execute
Query
CART Database
Query Result
3.4
Populate
the
Workbook

CHAPTER 3

SYSTEM DESIGN

3.1 SYSTEM ARCHITECTURE

Caritor Analysis and Reporting Tool (CART) follows Model View Controller (MVC) architecture.

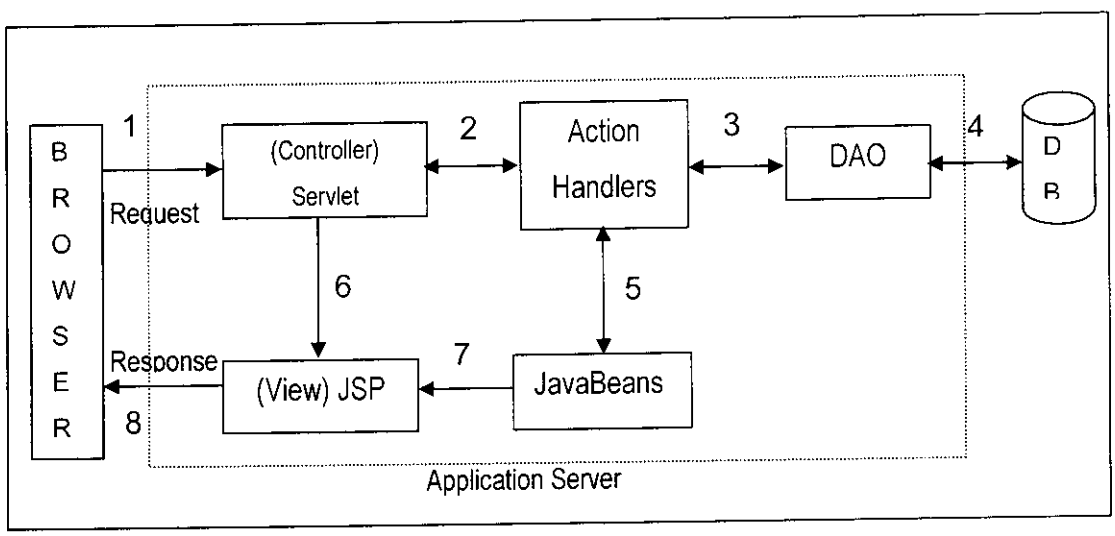


Figure 3.1 – MVC Architecture

Model-View-Controller is an architectural pattern that separates the responsibilities in a web application built using Servlet and JSP technologies.

executing some business logic, and then determining the next view to display can really make for an unattractive JSP page. Application development and maintenance are much easier if the different components of a web application have clear and distinct responsibilities.

The MVC pattern has three key components:

Model : Responsible for the business domain state knowledge
View : Responsible for a presentation view of the business domain
Controller : Responsible for controlling the flow and state of the user input

- The Controller Servlet routes the request between the JSP pages, by performing tasks requested by the source JSP, and passing the results to the destination JSP.
- DAO Classes perform the Database operations. These classes are invoked from the Controller Servlet.
- Java Beans are data holders or value objects. Java Beans are used to store the values retrieved from the Database and pass them to the JSPs through the Controller Servlet.
- Java Server Pages (JSPs) are used to store the HTML framework in the Application Server for displaying the HTML content in the browsers based on client requests.
- Java Script is used to perform client side scripting and validations.
- Style Sheets are used to ensure a uniform look and feel and a single point code change for the systems User Interface font, size, colors etc.
- Oracle JDBC Driver is used to perform Database operations.
- Database Properties file is used to connect to the specific Schema in the Database

- Java Reusable classes encapsulate common methods across the application. This not only has the implicit advantage of reusability and reduced coding time, but also allows the system logic to be configured and fine tuned in just one place rather than doing it across all the Program components. These methods will be called from the DAO class.

3.2 METHODOLOGY

A system development model or methodology is a framework used to structure, plan and control the information development process. The V-Process Methodology is followed in this project. Each Phase in the process represents a state of evolution of software and has associated project deliverables. Each Phase is controlled using an ETVX model (Figure 3.2)

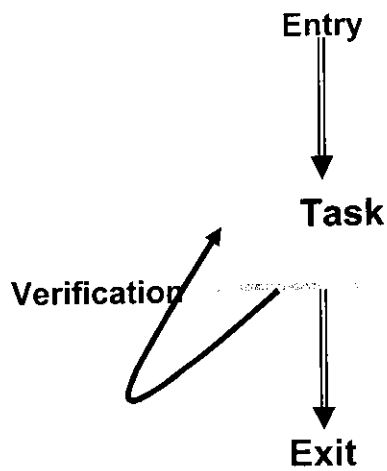


Figure 3.2 – ETVX Model

Entry Criteria	Condition that must be satisfied before beginning the phase
Task	A set of tasks to be carried out in the phase
Verification	A set of verification task to verify the quality of the deliverable
Exit Criteria	A set of conditions that be satisfied before the phase is concluded

Table 3.1 – ETVX Model

3.2.1 V-Process Model

The various phases of the life cycle along with the associated deliverables in each phase are shown below. The corresponding testing process carried is also defined clearly to ensure quality of the software and the deliverables of each phase. Figure 1.2 shows the V-Process Model.

Mandatory activities include:

- Definition Phase
- At least one Design Document in addition to the Analysis
- At least one testing

In the figure,

- The Left of '**V**' is Verification-Here reviews of the deliverables takes place
- The Right of **V**' is Validation-Here Testing of the deliverables takes place

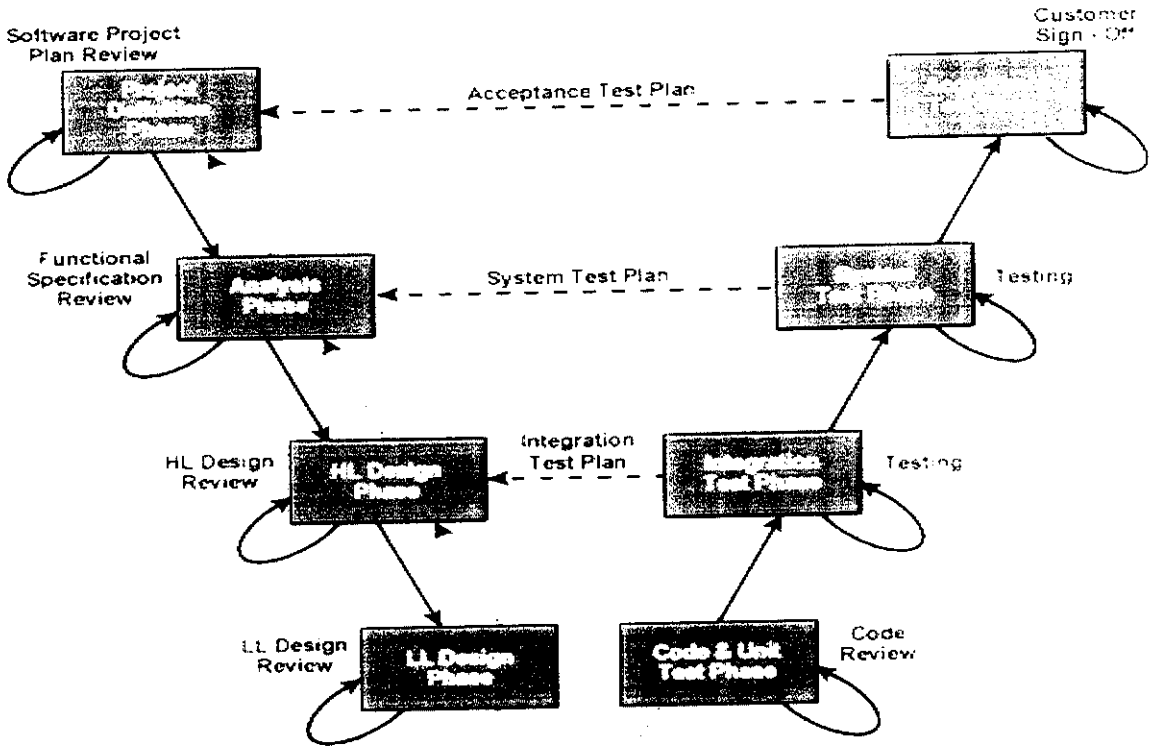


Figure 3.3 – V-Process Life Cycle Model

3.3 PROCEDURAL DESIGN

The modules that will be required for the tool to confirm to the requirements as discovered during the analysis phase are:

- Effort Tracking
- Reporting
- Authentication

3.3.1 Effort Tracking

Effort tracking module tracks the daily effort put on each client request. When an employee enters the effort put for a particular day, a trigger is fired to update the effort table if the new effort entered is not equal to the existing effort. The service request ID, the date and time of update and the effort entered by the employee is tracked in a separate table. This table tracks the day-wise effort spent on various requests for the current month. If a new requests comes in, a new entry is created in the effort table. If the insert/update on the effort table happens on a new month, all the previous month details are removed and tracked day-wise in a separate table. Thus, tracking the historical data.

As all the teams in the organization are not allowed to access the live database, an option for updating the database is provided in the system through Upload File facility. This reads a Microsoft Excel file and updates the database. The file can be uploaded remotely from the client machine itself.

All the requests that were tracked for the last three/six months can be discarded when the user chooses the discard option. An option to delete any request from the effort database is also provided to the user.

3.3.2 Reporting

The user is provided with an option to view/download the reports he has generated. All the generated reports are write-protected Microsoft Excel file.

3.3.2.1 Effort Reports

This module generates two different effort reports such as Effort Summary Report and Monthly Effort Report. Each of these reports differ on the basis in which they are generated.

Effort Summary Report gives the product-wise effort spent on the requests. Each request that the client makes are distinguished based on their severity level. This report gives the cumulative effort spent based on the product and severity.

Monthly Effort Reports give the day-wise effort spent on the client requests. This includes two reports. The first report gives the effort spent on each client request till previous month and the second report gives the effort spent on client requests that was created previous month.

3.3.2.2 Aging Reports

This module generates two different reports namely, Aging Report and Ticket Report. Aging report includes the details of all the requests under different slabs based on the age of each request. The slabs being:

- Requests that are 15 days old
- Requests that are 15-30 days old
- Requests that are 30-60 days old
- Requests that are 60-90 days old
- Requests that are more than 90 days old and

Ticket report is a summary report, which gives

- The count of requests with respect to the slabs
- The count with respect to the product name
- The sum total of all client requests

3.3.2.3 Status Reports

This module generates two different reports, Monthly Status Report and Weekly Status Report.

Monthly Status Report generates two reports viz.

1) Detail Report that reports about
   - The requests that were closed previous month
   - The requests that were in progress previous month and
   - The requests that were in waiting status previous month

2) Summary Report that gives
   - Total number of requests received previous month and
   - Total number of requests serviced previous month

Weekly Status Report is an automated system that generates a report that reports about

- The requests that were closed last week
- The requests that were in progress last week

3.3.2.4 Analysis Report

Analysis of two reports can be done through CART. This module finds the difference between the two uploaded reports and a report along with a chart is generated. Files can be uploaded from the client to the server and the generated Analysis Report can be viewed or downloaded for further reference. If the user uploads two different reports, the user is instructed to upload two similar reports.

3.3.3 Authentication System

This is the basic authentication method that is widely used, and is industry-standard method for collecting user name and password information. Authentication System proceeds as follows:

- The web browser on the client computer displays a login page where users can enter their previously assigned user name and password
- The web browser then attempts to establish a connection using this information
- If the server rejects the information, the browser repeatedly displays the page until the user either enters a valid user name and password or closes the page
- When the web server verifies that the user name and password correspond to a valid user, the connection is established
- If the user plays more than one role in the organization, he is prompted with a combo box listing all his roles for him to choose. Else, he is routed to the home page directly
- Menu options are loaded based on the role the user chooses or

3.4 TABLE DESIGN

Table 3.2 – T_CART_EMPLOYEE

Description: Holds the details of the Employees

Field Name	Data Type	Description
C_Emp_EmployeeID	Number	User name of the employee (PK)
C_Emp_FirstName	Text	First Name of the employee
C_Emp_LastName	Text	Last Name of the employee
C_Emp_Password	Text	Password of the employee

Table 3.3 - T_CART_ROLES

Description: Holds the details of all available Roles

Field Name	Data Type	Description
C_Role_RoleID	Number	Unique Role identifier (PK)
C_Role_Role Name	Text	Name of the role
C_Role_RoleDescription	Text	Short description of the role

Table 3.4 - T_CART_REPORTS

Description: Holds the details of the Reports

Field Name	Data Type	Description
C_ Report_ReportID	Number	Unique Report identifier (PK)
C_ Report_ReportName	Text	Name of the report
C_ Report_ReportDescription	Text	Short description of the report

Table 3.5 - T_CART_EMPROLES

Description: Holds the EmployeeIDs and the Roles they play

Field Name	Data Type	Description
C_ERole_EmployeeID	Number	User name of an employee (PK part of FK)
C_ERole_RoleID	Number	Employee Role (PK part of FK)

Table 3.6 - T_CART_ACCESS
Description: Holds roles and relative reports accessible

Field Name	Data Type	Description
C_Accs_RoleID	Number	Unique role identifier (PK part of FK)
C_Accs_ReportID	Number	Unique report identifier (PK part of FK)

Table 3.7 - T_CART_EFFORTTRACK
Description: Holds the tracked updated effort spent on client requests

Field Name	Data Type	Description
C_EffT_SRID	Number	Client Request identifier (PK)
C_EffT_UpdatedDate	Date	Date when effort is updated (PK)
C_EffT_EffortSpent	Number	Effort spent on the request

Table 3.8 - T_CART_TICKET
Description: Holds the details of client requests

Field Name	Data Type	Description
C_Ticket_SRID	Number	Client Request identifier (PK)
C_Ticket_CustomerID	Text	Customer identifier
C_Ticket_CustomerName	Text	Name of the customer
C_Ticket_ProductName	Text	Name of the product
C_Ticket_ServiceName	Text	Type of the service
C_Ticket_Severity	Text	Severity level of the request
C_Ticket_Description	Text	Description of the service
C_Ticket_Status	Text	Current status of the request
C_Ticket_OwnedBy	Text	Owner of the request
C_Ticket_CreatedDate	Date	Creation date of the request
C_Ticket_ExpectedDate	Date	Expected date of completion
C_Ticket_ResolvedDate	Date	Date when request is resolved
C_Ticket_ClosedDate	Date	Request closure date
C_Ticket_EffortSpent	Number	Total effort spent on the request

Table 3.9 - T_CART_EFFORTMONTH

Description: Holds the tracked day-wise effort spent

Field Name	Data Type	Description
C_EffMnth_SRID	Number	Client request identifier (FK)
C_EffMnth_Month	Text	Month for which effort is tracked (PK)
C_EffMnth_InsertedDate	Date	Last updated date of the ticket
C_EffMnth_Day 01	Number	Effort spent on 01st day of the month
C_EffMnth_Day 02	Number	Effort spent on 02nd day of the month
C_EffMnth_Day 03	Number	Effort spent on 03rd day of the month
C_EffMnth_Day 04	Number	Effort spent on 04th day of the month
C_EffMnth_Day 05	Number	Effort spent on 05th day of the month
C_EffMnth_Day 06	Number	Effort spent on 06th day of the month
C_EffMnth_Day 07	Number	Effort spent on 07th day of the month
C_EffMnth_Day 08	Number	Effort spent on 08th day of the month
C_EffMnth_Day 09	Number	Effort spent on 09th day of the month
C_EffMnth_Day 10	Number	Effort spent on 10th day of the month
C_EffMnth_Day 11	Number	Effort spent on 11th day of the month
C_EffMnth_Day 12	Number	Effort spent on 12th day of the month
C_EffMnth_Day 13	Number	Effort spent on 13th day of the month
C_EffMnth_Day 14	Number	Effort spent on 14th day of the month
C_EffMnth_Day 15	Number	Effort spent on 15th day of the month
C_EffMnth_Day 16	Number	Effort spent on 16th day of the month
C_EffMnth_Day 17	Number	Effort spent on 17th day of the month
C_EffMnth_Day 18	Number	Effort spent on 18th day of the month
C_EffMnth_Day 19	Number	Effort spent on 19th day of the month
C_EffMnth_Day 20	Number	Effort spent on 20th day of the month
C_EffMnth_Day 21	Number	Effort spent on 21st day of the month
C_EffMnth_Day 22	Number	Effort spent on 22nd day of the month
C_EffMnth_Day 23	Number	Effort spent on 23rd day of the month
C_EffMnth_Day 24	Number	Effort spent on 24th day of the month
C_EffMnth_Day 25	Number	Effort spent on 25th day of the month
C_EffMnth_Day 26	Number	Effort spent on 26th day of the month
C_EffMnth_Day 27	Number	Effort spent on 27th day of the month
C_EffMnth_Day 28	Number	Effort spent on 28th day of the month
C_EffMnth_Day 29	Number	Effort spent on 29th day of the month
C_EffMnth_Day 30	Number	Effort spent on 30th day of the month
C_EffMnth_Day 31	Number	Effort spent on 31st day of the month

3.5 ENTITY RELATIONSHIP DIAGRAMS

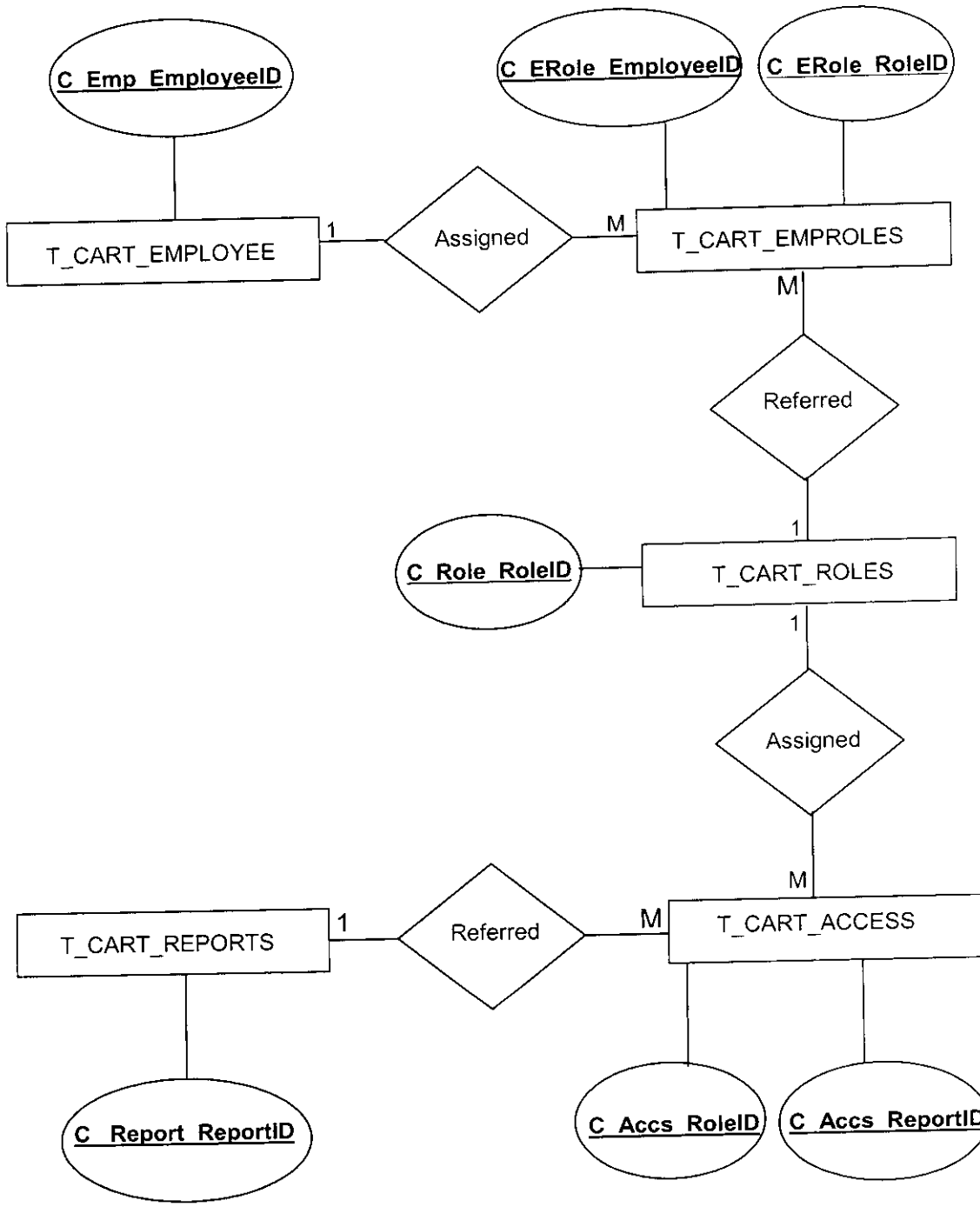


Figure 3.4 – ER Diagram (User Database)

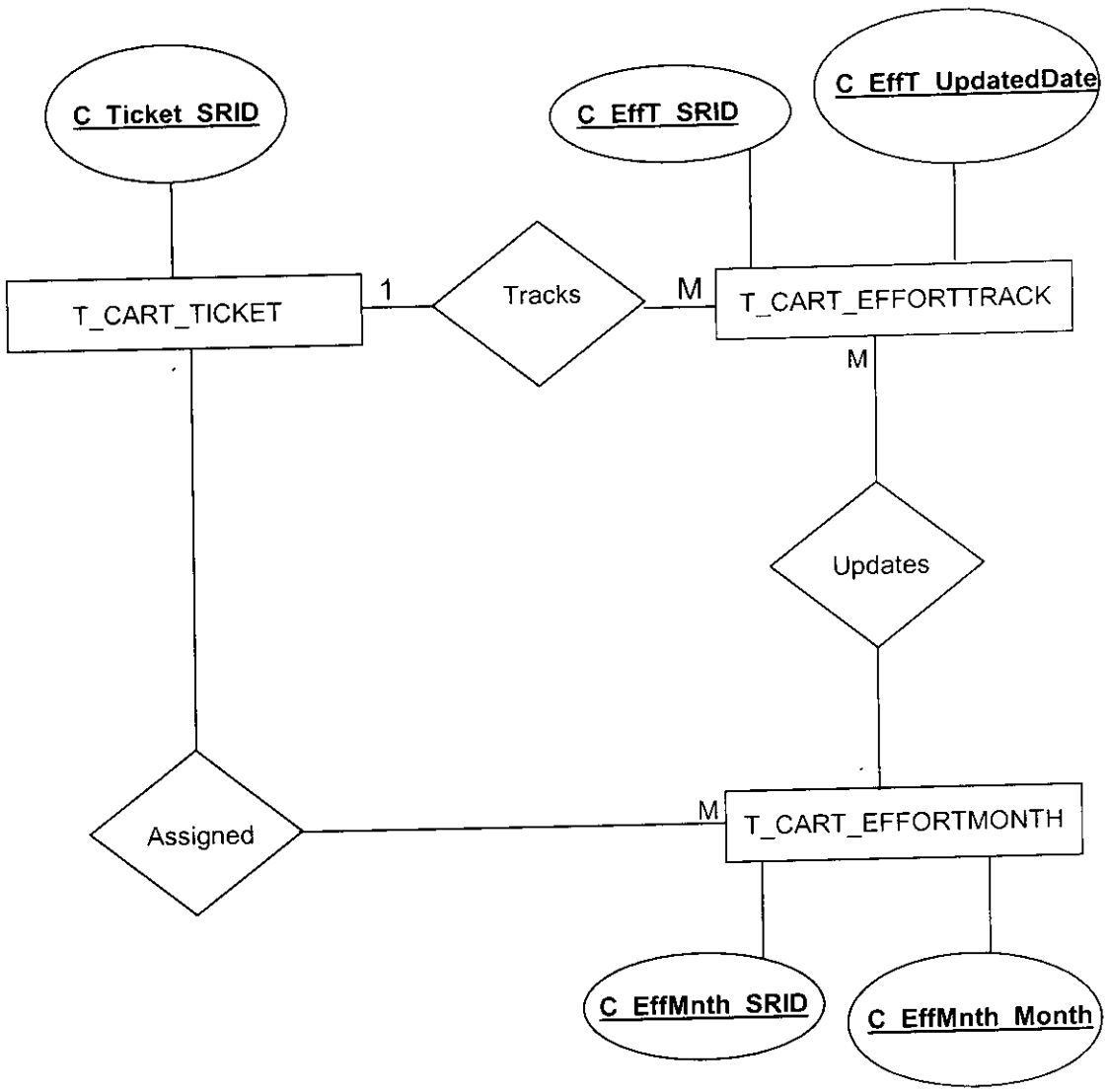


Figure 3.5 – ER Diagram (Effort Database)

3.6 PROCESS FLOW DIAGRAM

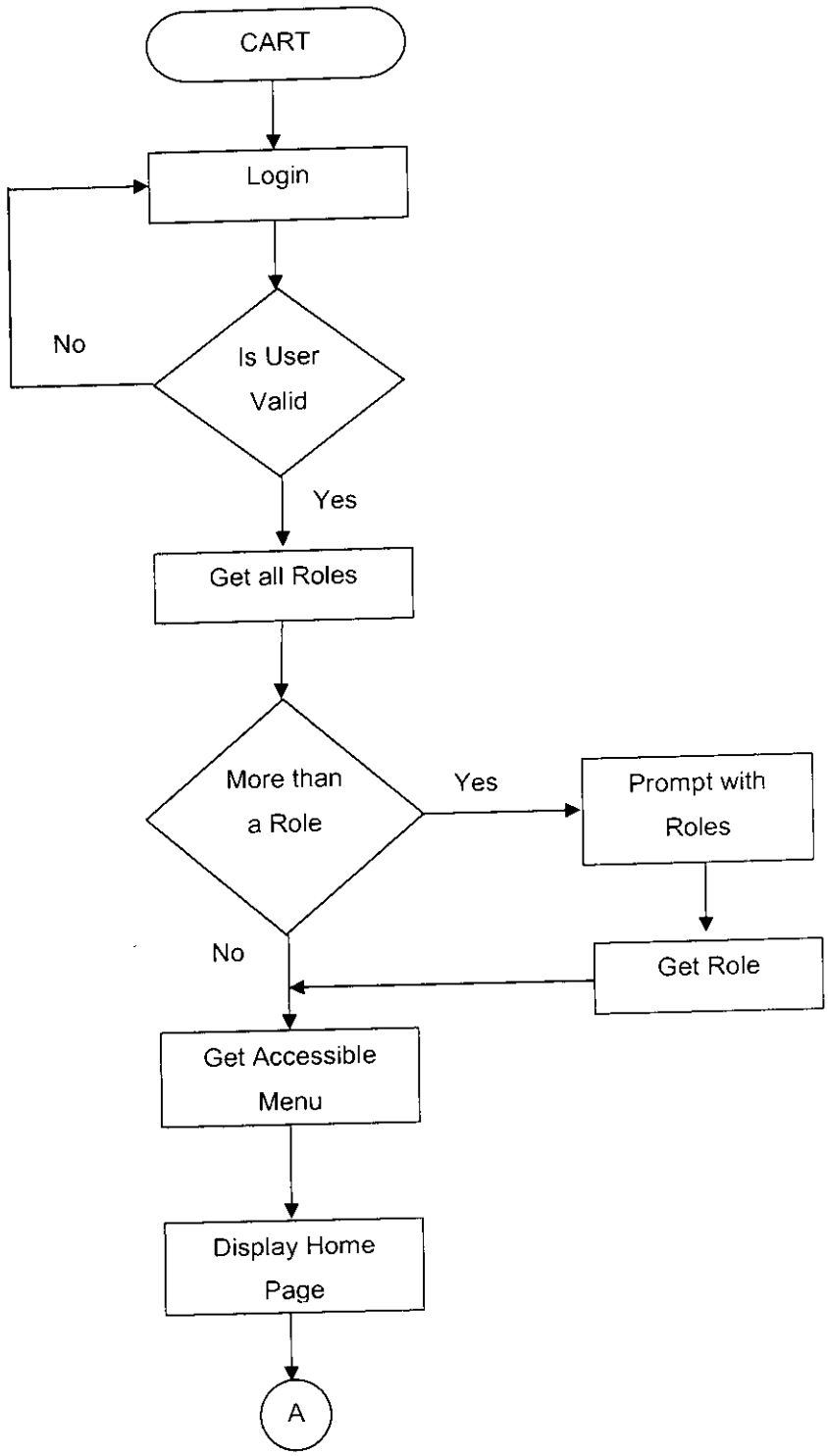


Figure 3.6 (Continued)

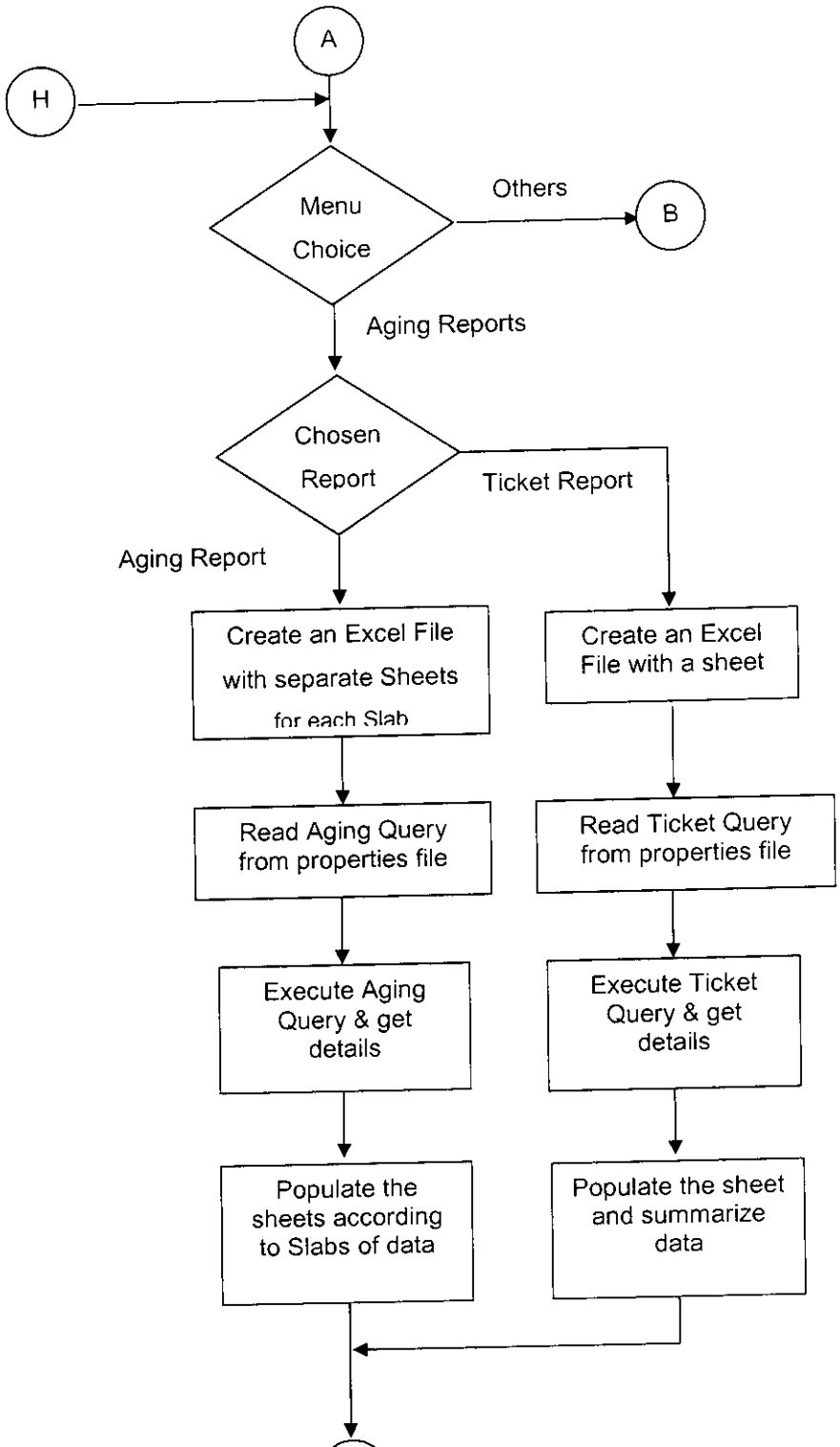


Figure 3.6 (Continued)

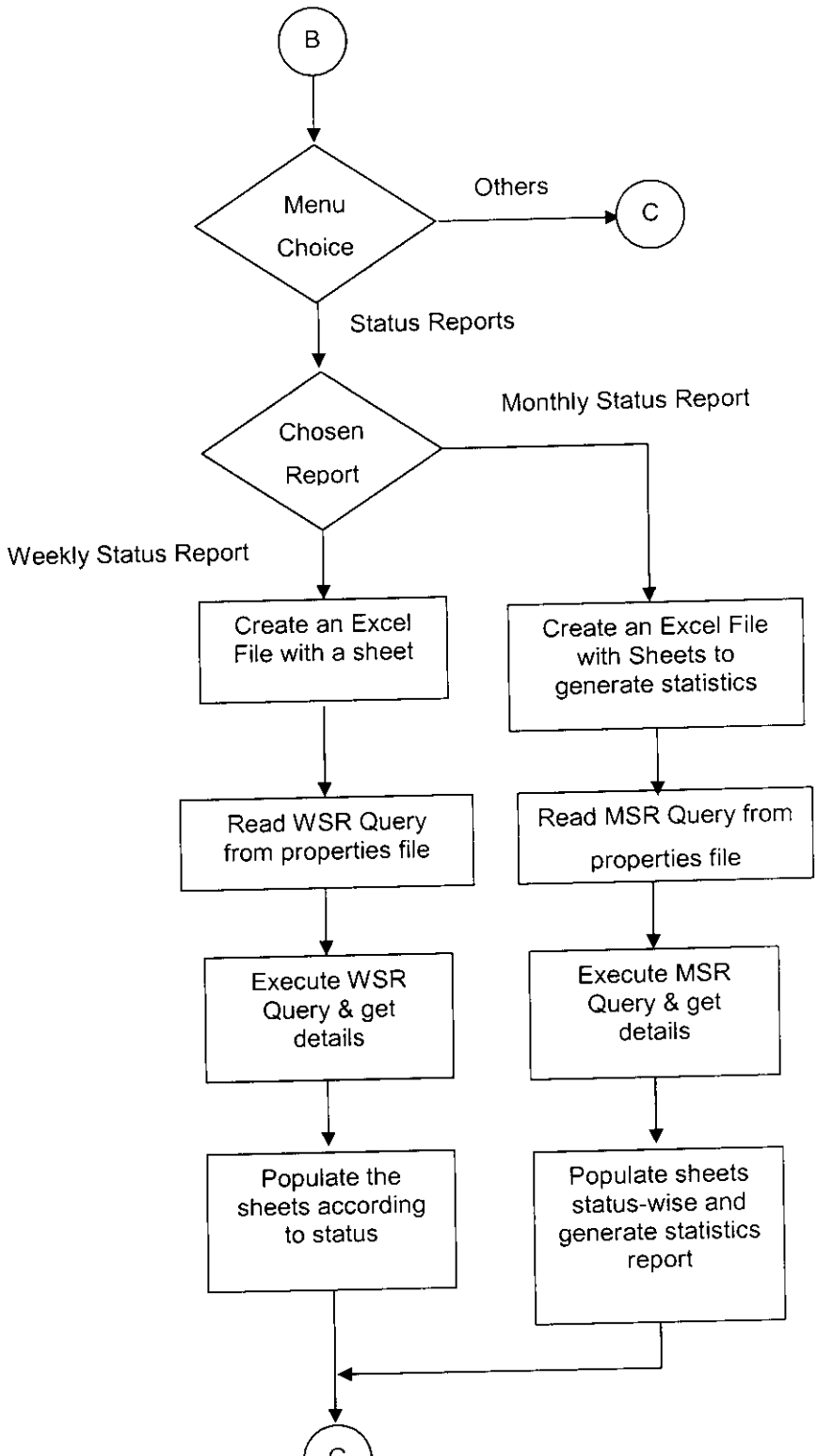


Figure 3.6 (Continued)

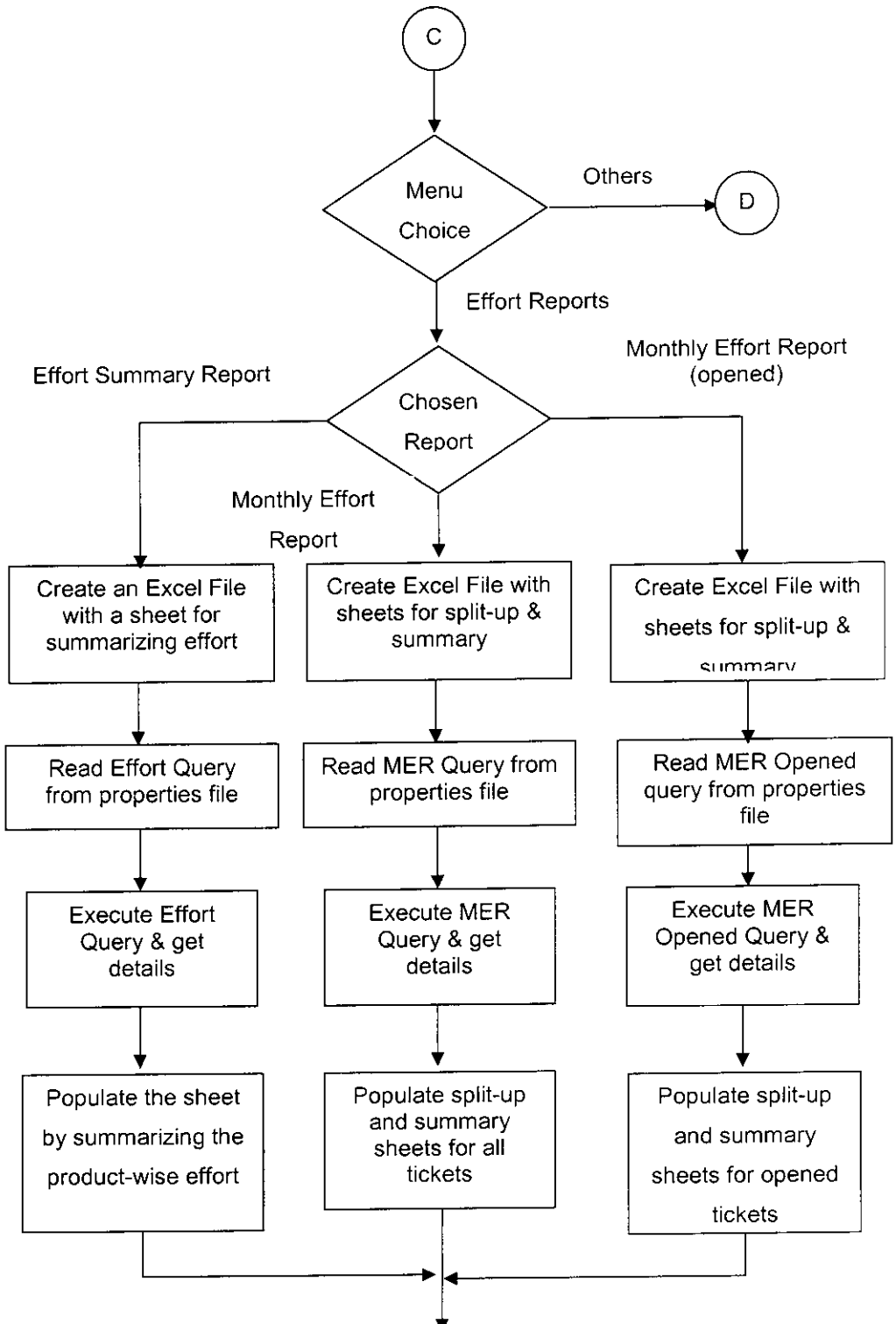


Figure 3.6 (Continued)

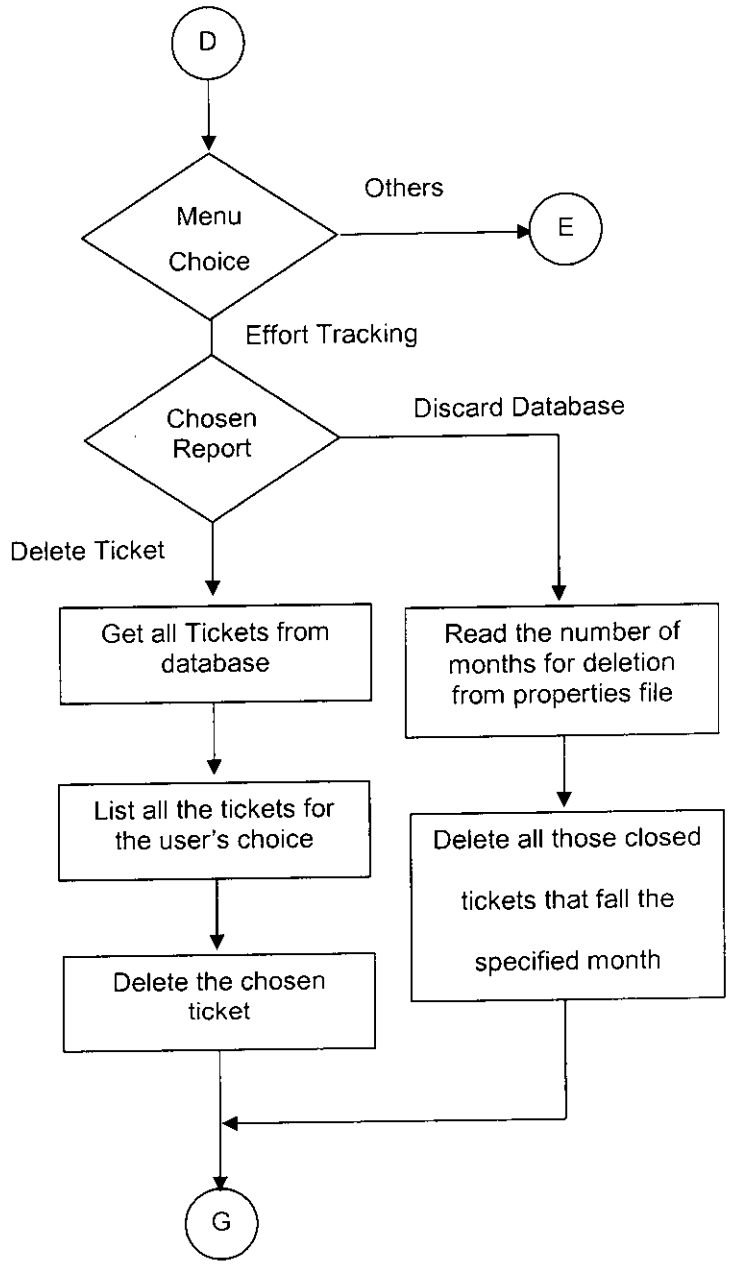


Figure 3.6 (Continued)

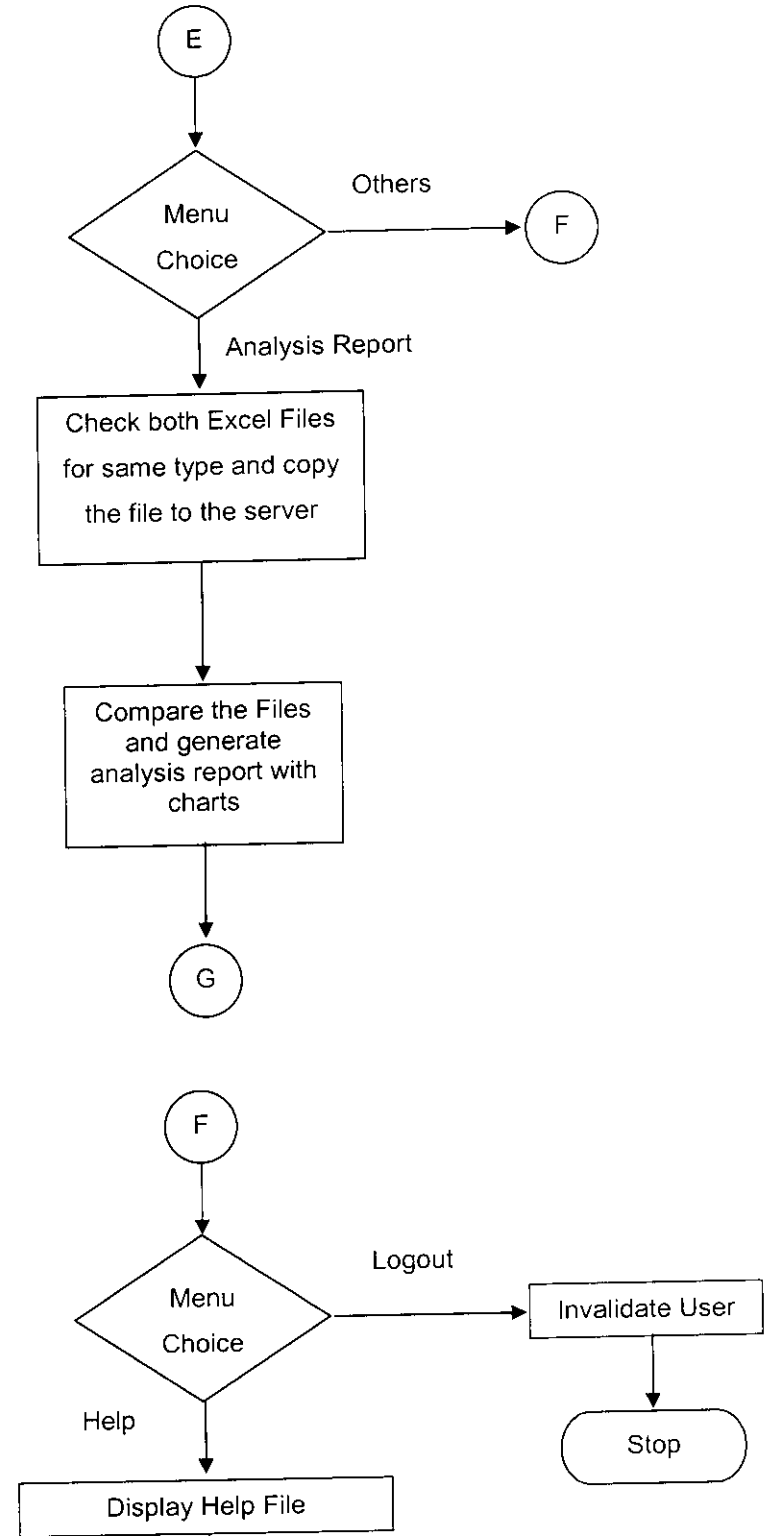
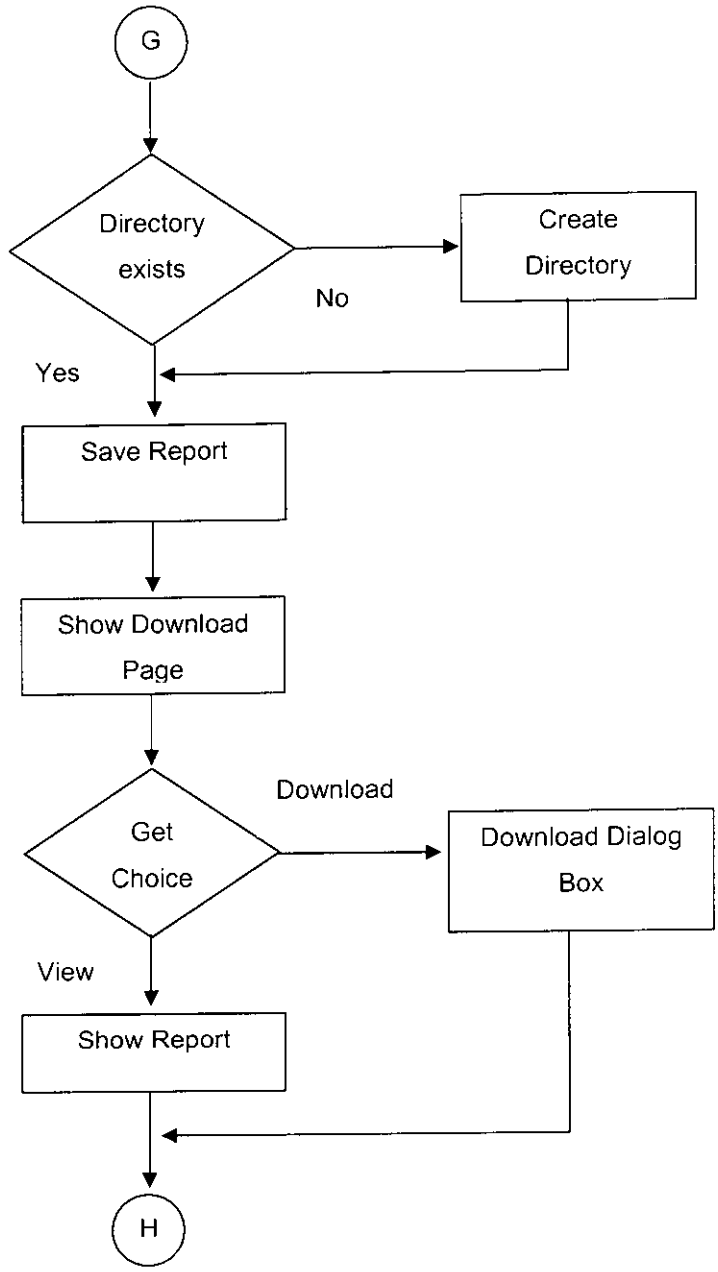


Figure 3.6 (Continued)



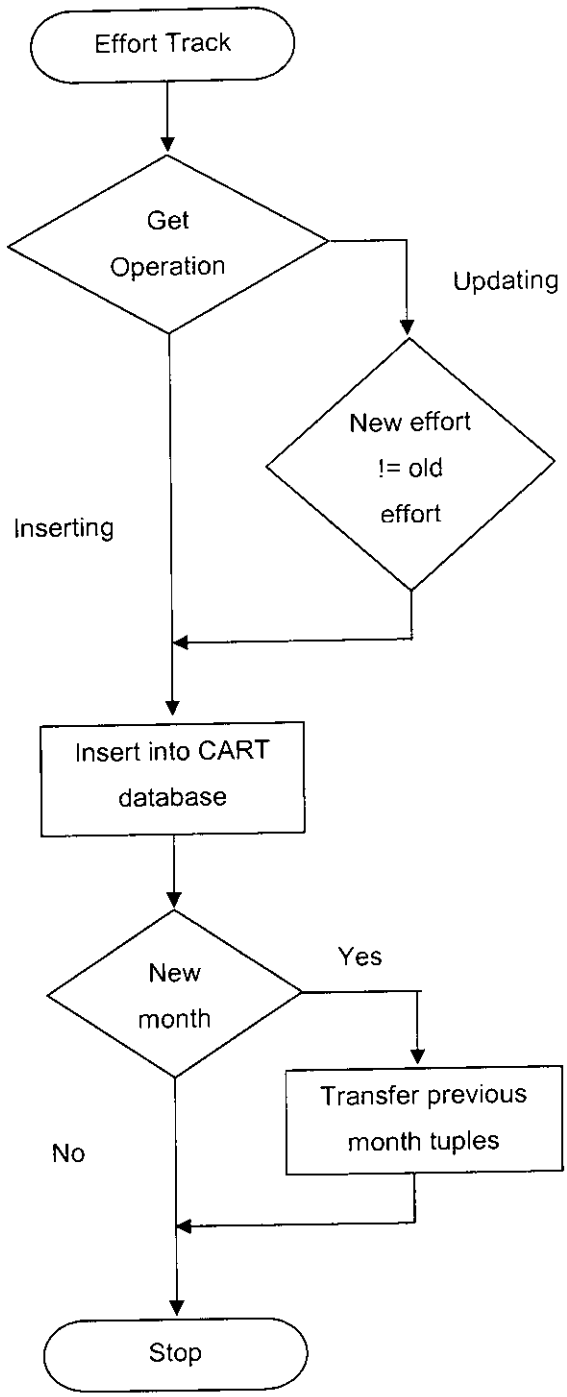


Figure 3.7 – Flow Diagram: Effort Tracking

CHAPTER 4

SYSTEM CONFIGURATION

Caritor Analysis and Reporting Tool have been developed with the following hardware and software configurations:

4.1 HARDWARE CONFIGURATION

Development and Deployment Perspective

Processor	: Intel Pentium IV - 2.8GHz
HDD	: 40 GB
RAM	: 512 MB

4.2 SOFTWARE CONFIGURATION

Development and Deployment Perspective

Client	: Windows 98/2003
Server	: Apache Tomcat 5.0
Tools	: JSP 1.2, Servlet 2.0, HTML 4.0, Java Script 1.1, J2SDK 1.3.0, JExcel 2.0, JUnit Test Suite3.8.1, Log4j
Database	: Oracle 8.1.7.0.0

CHAPTER 5

TESTING

Testing is a process of executing a program/project or an application with the intent of finding errors. It is defined as checking an application to ensure whether it meets the customer requirements / needs. The testing process focuses on the logical internals of the software assuring that all the statements have been tested and also on the functional externals by conducting tests to uncover errors. This process also ensures that defined input will produce actual results that agree with required results. Testing is an important part of the Development Life Cycle. The amount of testing required is related to the size and complexity of the application. The Essence of Software Testing is as follows:

- Preventing injection of defects
- Preventing defects from migrating downstream
- Detecting defects after they are introduced
- Best addressed by improving the way you test and develop
- Understanding test technology is the key to prevention and detection

Following are the tests that were carried out to test the working of CART:

- Unit Testing
- Integration Testing

5.1 UNIT TESTING

Unit testing is performed on a single stand-alone module or unit of code. Unit testing is the phase of testing that tests the basic functionality and structure of the code and carried out by the developer and is white-box oriented. The module interface is tested to ensure that information properly flows into and out of the program unit under test. Boundary conditions are tested to ensure that the module operates properly at boundaries established to limit or restrict processing. All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once. And finally, all error-handling paths are tested. An automated tool called JUnit was used to write the test cases and to execute each one of them. Defects and deviations in Date formats, Special requirements in input conditions (for example Text box where only numeric or alphabets should be entered), selection based on Combo Box's, List Box's, Option buttons, Check Box's would be identified during the Unit Testing phase.

JUnit Plug-in for Eclipse 3.1 is an automated unit testing tool that is used to test the developer's code. The Caritor test tools employed are Inspection and Review Record (IRR) and Defect Tracking System (DTS). The sequence of steps followed is as follows:

- Unit Test Plan was prepared before developing the unit
- UTP was reviewed and approved by the Software Quality Assurance Team
    - Entry Criteria
        - Reviewed Code
        - Approved Unit test plan
    - Tasks

- Create Test Data to test the component / program
- Execute test cases as per Unit Test Plan
- Log Defects in DTS along with Root Cause

o Verification and Validation

- Unit Test Results

o Exit Criteria

Unit Tested Code

5.2 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover errors associated with interfacing. That is, the program is constructed and tested in small segments, which makes it easier to isolate and correct. This testing focuses on the design and construction of the software architecture. Integration testing is black box oriented. The objective is to take unit-tested modules and build a program structure that has been dictated by the design. Both integration and testing takes place in this phase. Units are combined one at a time and tested, till entire software gets integrated.

Since the project-size of CART was small, integration testing was done by the developer itself and reviewed by the project lead. It involved Bottom-Up Integration. It began with the construction and testing of atomic modules. The Caritor test tools employed are Inspection and Review Record (IRR) and Defect Tracking System (DTS).

5.3 SYSTEM TESTING

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system. At this stage the production environment is ready. The purpose of system testing is to validate the system against the functional specification by executing the test cases prepared during the analysis phase. Although each test has a different purpose, all work to verify that system elements have been properly integrated and perform allocated functions.

Since the project-size of CART is small, it will be done by the developers themselves. The System Testing covers the following:

- Functional Requirements
- Non-functional Requirements (Performance, Recovery, Usability)

The Caritor test tools to be employed are Inspection and Review Record (IRR) and Defect Tracking System (DTS). Stress tests and Security tests, which are types of system tests, were carried out successfully for CART.

5.3.1 Stress Testing

Stress testing is designed to confront programs with abnormal situations. Stressing the system may often cause functional failures. It involves testing an application under heavy loads, such as testing of a web site under a range of loads to determine at what point the system's response time degrades

CART was put into stress testing to check the availability of resources at abnormal conditions. This helps in the fine-tuning of the system. The Application should be able to withstand simultaneous access by 50 users without affecting the performance. Generation of reports or Downloading and Uploading of files should not take more than 10 seconds.

5.3.2 Security Testing

Security testing attempts to verify that protection mechanisms built into a system will, in fact, protect it from improper penetration. During Security test, password cracking and unauthorized entry into the software are all taken into consideration. CART was tested to check if a user could directly enter the system by changing the URL. Only valid users of the system must be given access to certain pages. Also as the generated reports are password protected, attempts were made to edit the reports to check its credibility.

5.4 USER ACCEPTANCE TESTING

The user acceptance testing demonstrates to the customer that the predefined acceptance criteria have been met by the system. Acceptance Testing checks if the system delivers what was requested from it. User acceptance testing includes:

- User Acceptance testing is typically used as a mechanism to handover the system, that occurs just before the software is released to the customer. The end-users along with the developers perform the User Acceptance Testing with a certain set of test cases and typical scenarios.

- The customer may write the acceptance test criteria and request that these be executed by the developer, or the developer may produce the acceptance testing criteria which are to be approved by the customer.
- The customer may demand to Run the tests themselves or Witness tests run by the developer or Inspect documented test results.

Since CART has been developed for the internal usage of the organization, the customer is the organization itself. A formal test-drive of the tool was done to check the system against the specified requirements. The Software Quality Assurance team tested CART to identify any issues regarding it. CART has met all the requirements specified and has been successfully deployed in the organization.

5.5 TEST CASES

Table 5.1 – Test Cases

S.No.	Test Condition	Action	Expected Result
1.	Login Checking	Enter Username & Password	Proceed to home page if Username & Password is right
2.	Role Checking	Enter a valid Username & Password who has more than a role.	A combo box populated with all roles in the next page after login page
3.	Illegal Access of JSP Pages	Provide a URL with the page name without logging-in, in the address bar	Routed to Login Page
4.	Invalidate navigation Back option	Try going to previous pages	Retain the same page
5.	Upload File	Upload excel files and non-excel files	Uploads correctly if it's a Excel file or alerted with an error message
6.	Navigation Check	Choose links that route to some other pages	Routed to Chosen page
7.	Menu Check	Perform some operations in the menu	The report corresponding to selected menu option is generated
8.	Download Page	Select a menu operation to generate report	Routed to download page after the generation of the report
9.	View/Download Option	Try viewing and downloading the generated report	For viewing the report is opened & for downloading the browser to choose location is opened
10.	Help links	Click the links in the Help page	The page to the link chosen is routed from

CHAPTER 6

CONCLUSION

Caritor Analysis and Reporting Tool, which has been developed for the internal usage of the organization has met all the specified requirements. Tracking of effort spent by the employees helps in better Human Resource utilization. Partial automation of report generation has resulted in considerable decrease in manual work. CART acts as an aid for overall project tracking of Caritor projects too. Tracking of effort helps the organization in scheduling the future projects and in better resource management. CART also helps the organization in the process of Performance Appraisal.

CART being a web-tool, can be accessed from remote machines thus eliminating the need to install it in each machine for it to be accessed. Restricted access to the tool provides necessary security required. Read-Only reports are generated thus maintaining the credibility of data and restricting illegal alteration of information.

The Aging Reports give the age of all the client requests that is in progress or in wait state, which helps in identifying those requests that are not closed for a long time. Status Reports give the current status of all the client requests. Effort Reports give the monthly effort break-up. These customized

obtaining the current status of all client requests and to track of the progress of the requests.

As CART was developed in a portable application environment like JSP/Servlet's, it gives organization the flexibility to migrate servers and swap tools, as business needs change.

6.1 FUTURE ENHANCEMENTS

The following provides scope for further enhancement of Caritor Analysis and Reporting Tool:

- Provision for sending reports through e-mail
- Automatic generation of statistical report using Microsoft Excel macros
- Provision to switch over user-roles without logging off
- Generation of Analysis Reports without File uploading
- Online Help through Hot Keys

APPENDIX 1

SCREEN SHOTS

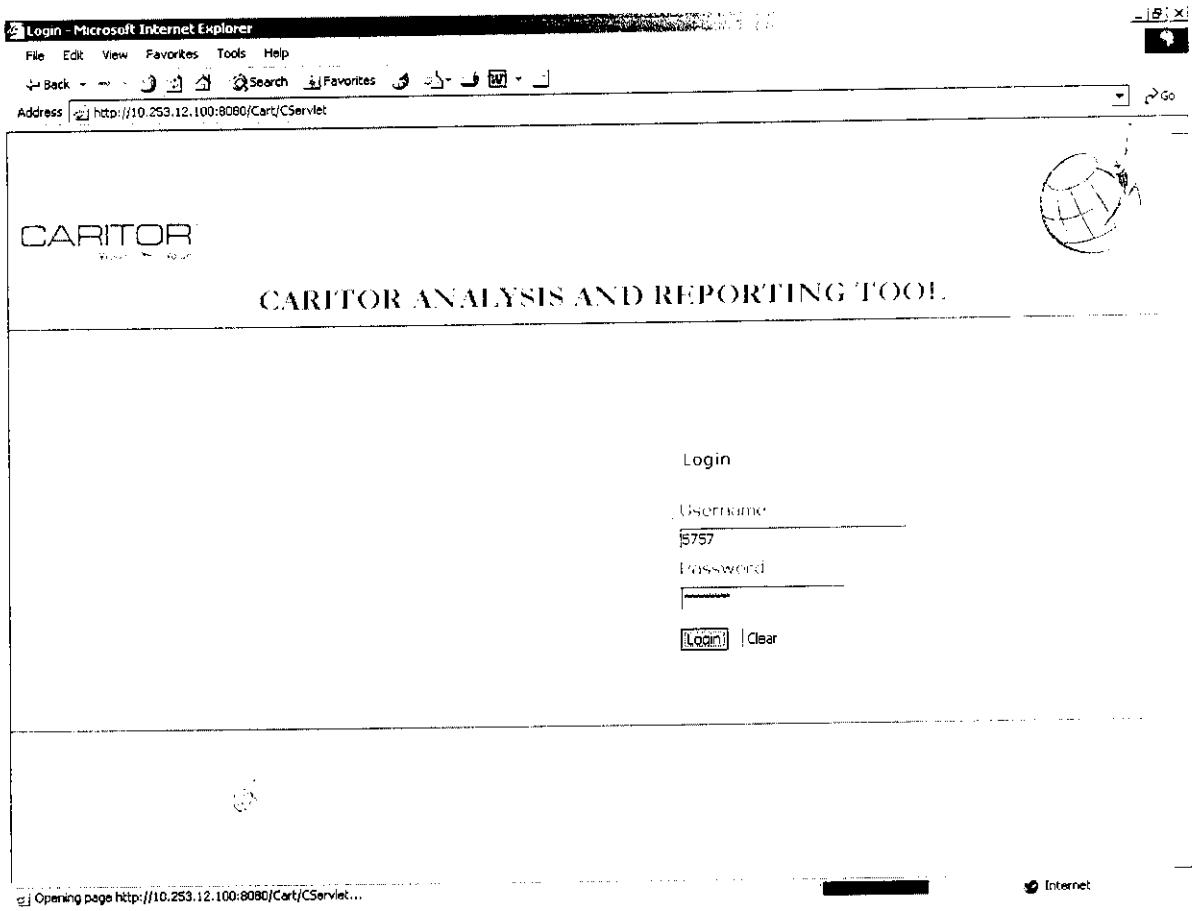


Figure A 1.1 – Login Page

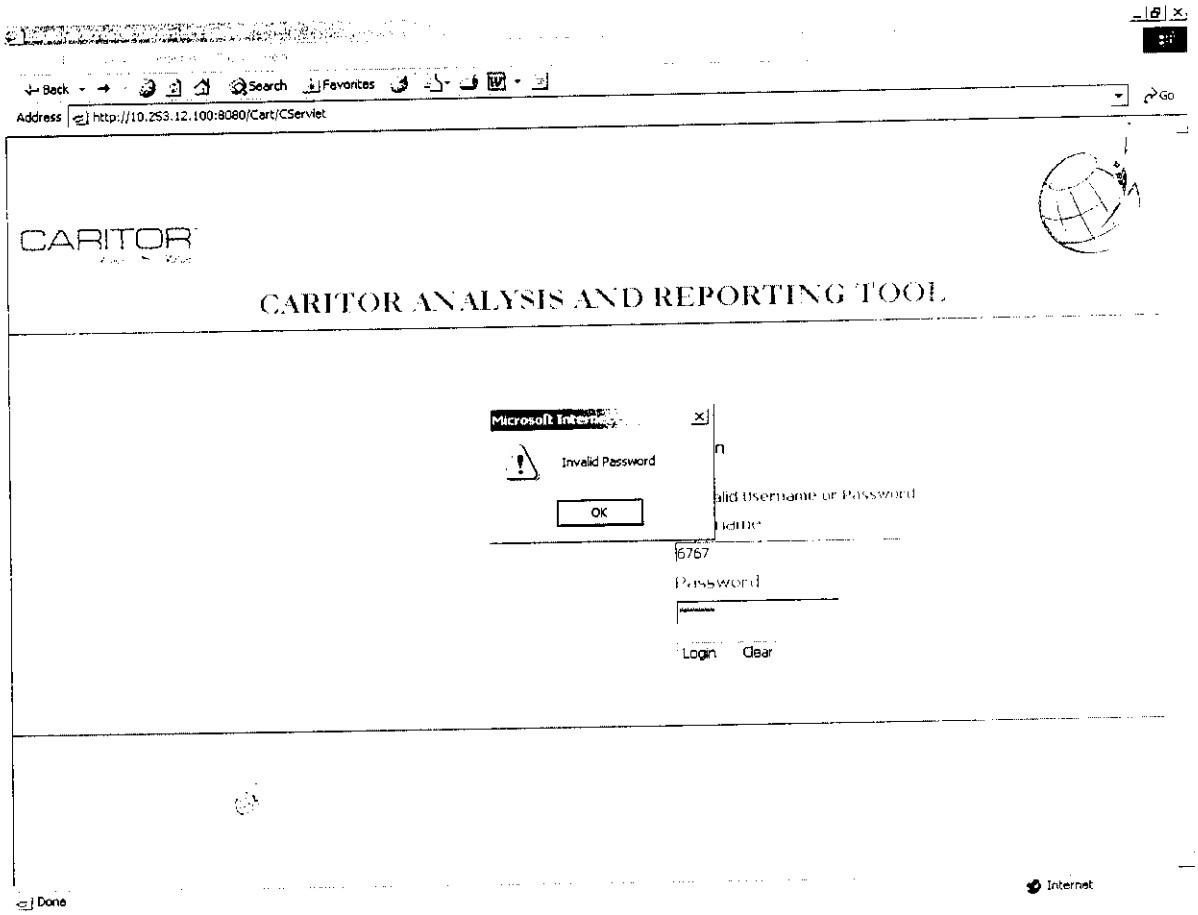


Figure A 1.2 – Invalid Login

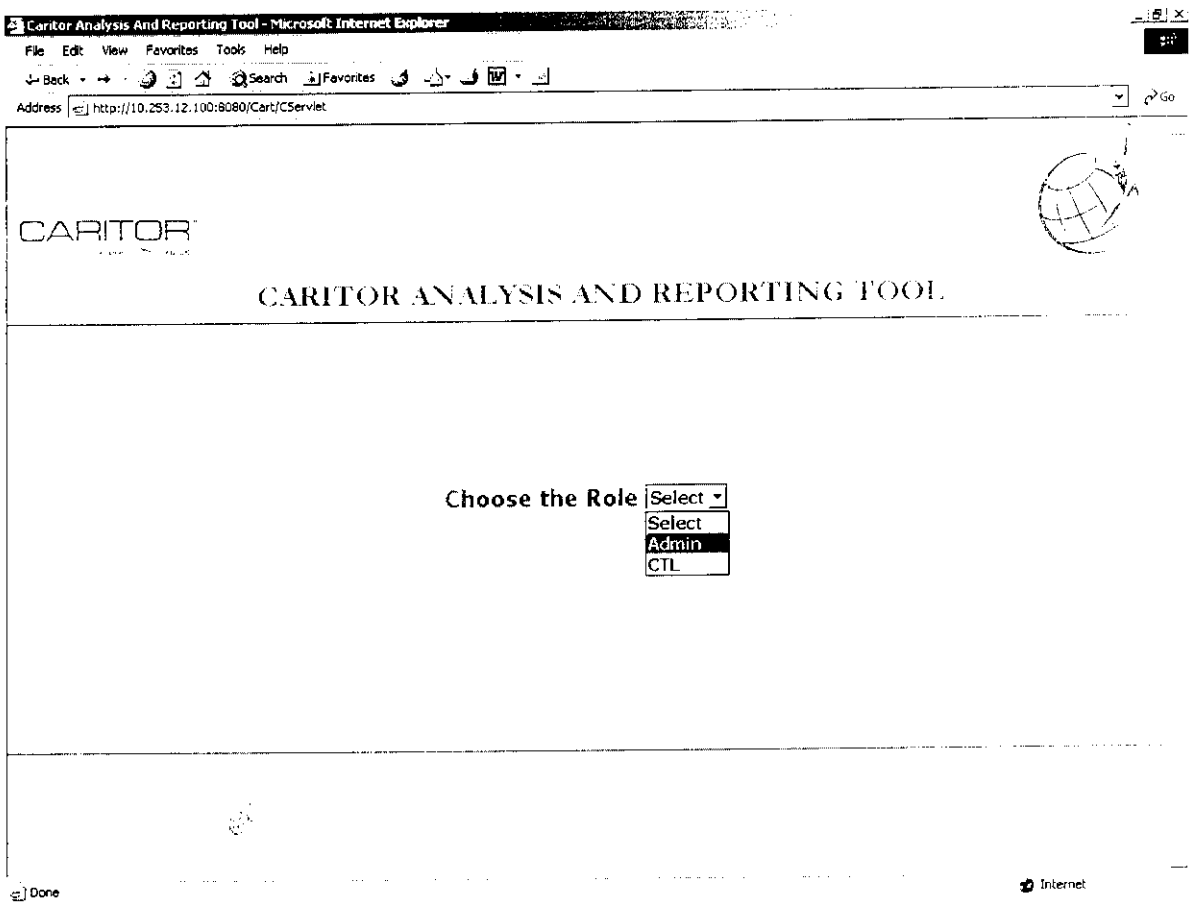


Figure A 1.3 – Role Selection

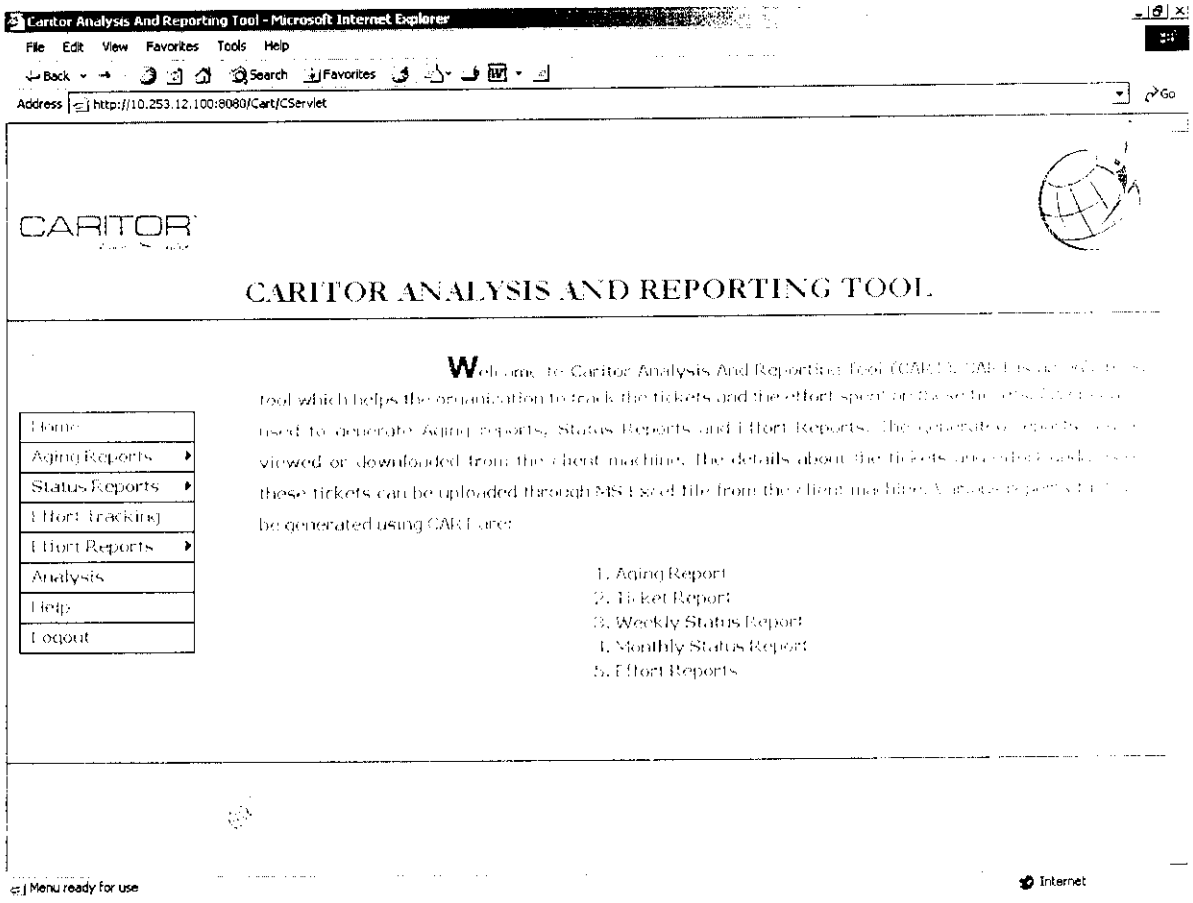


Figure A 1.4 – Home Page

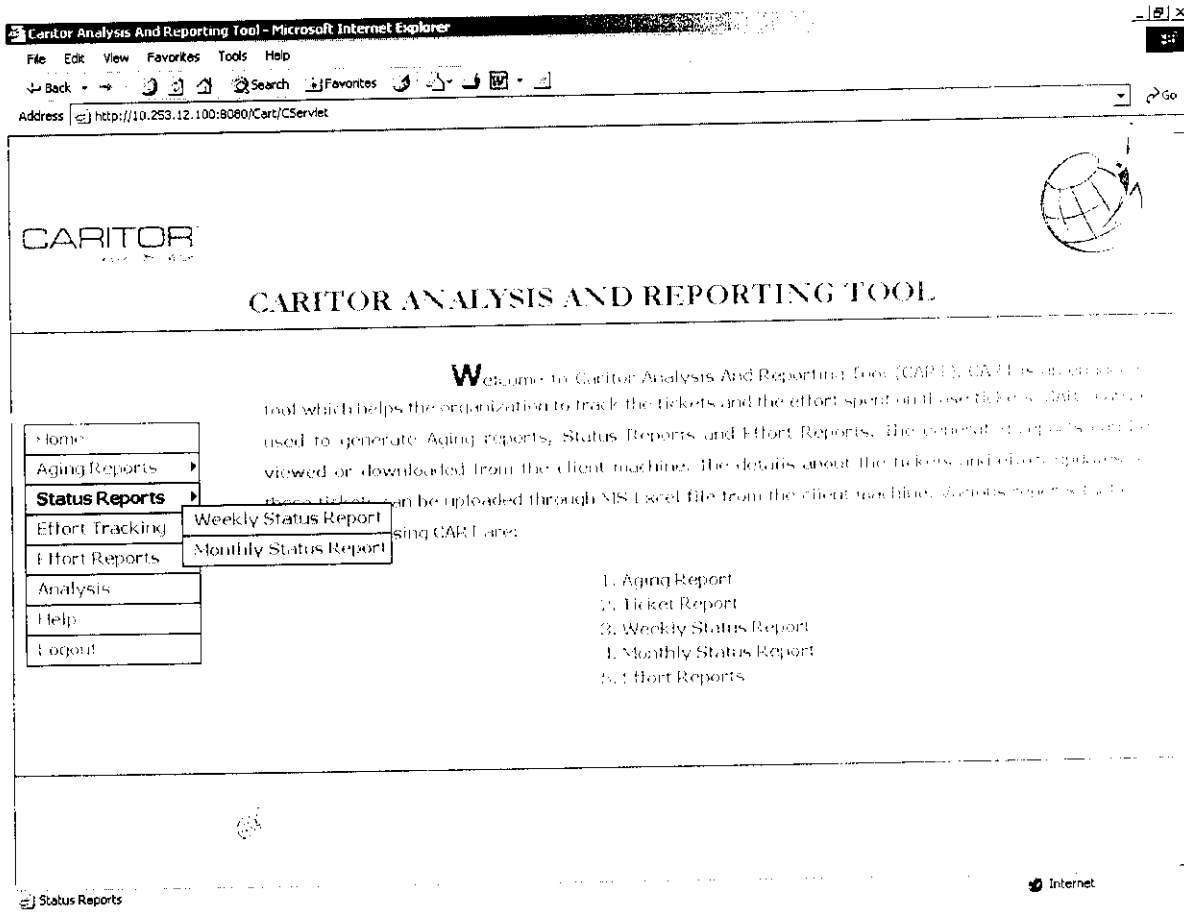


Figure A 1.5 – Report Generation

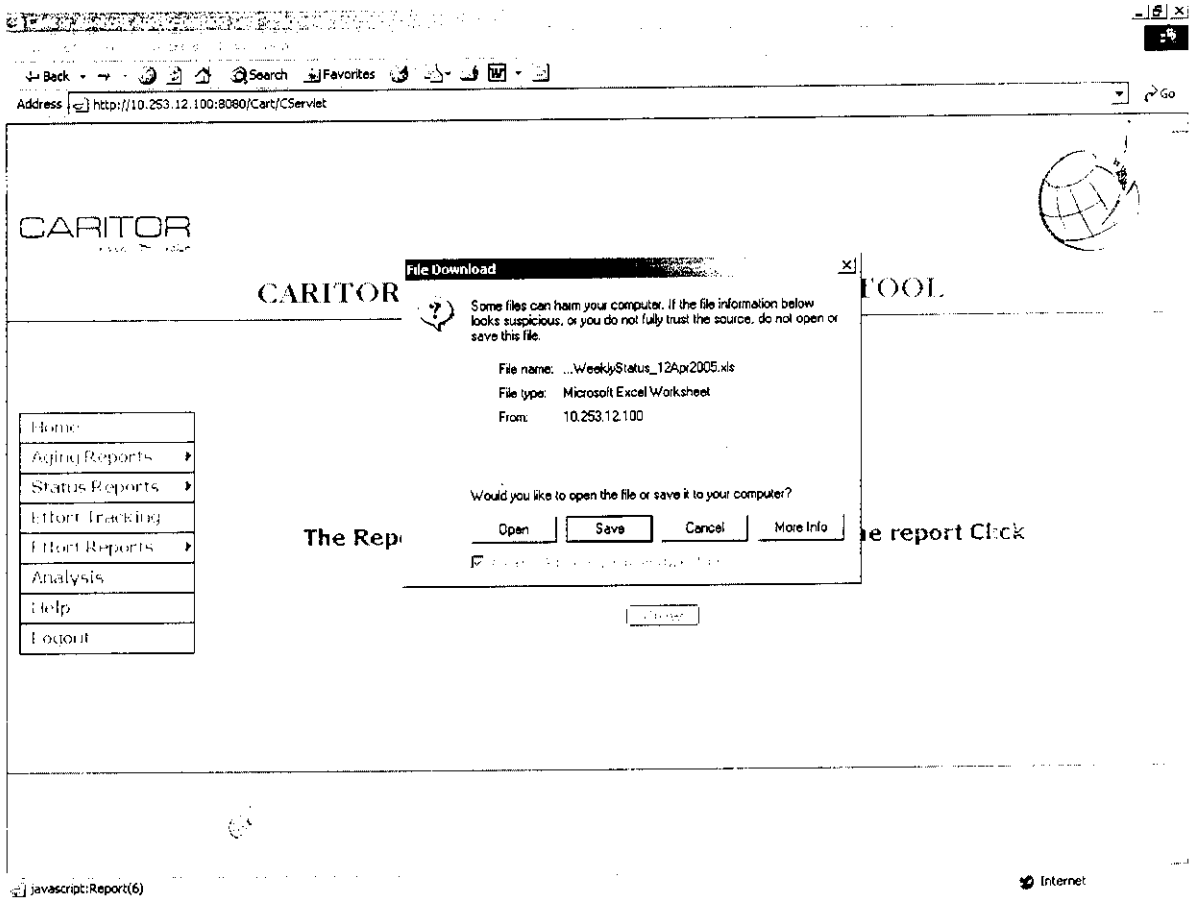


Figure A 1.6 – Download Report

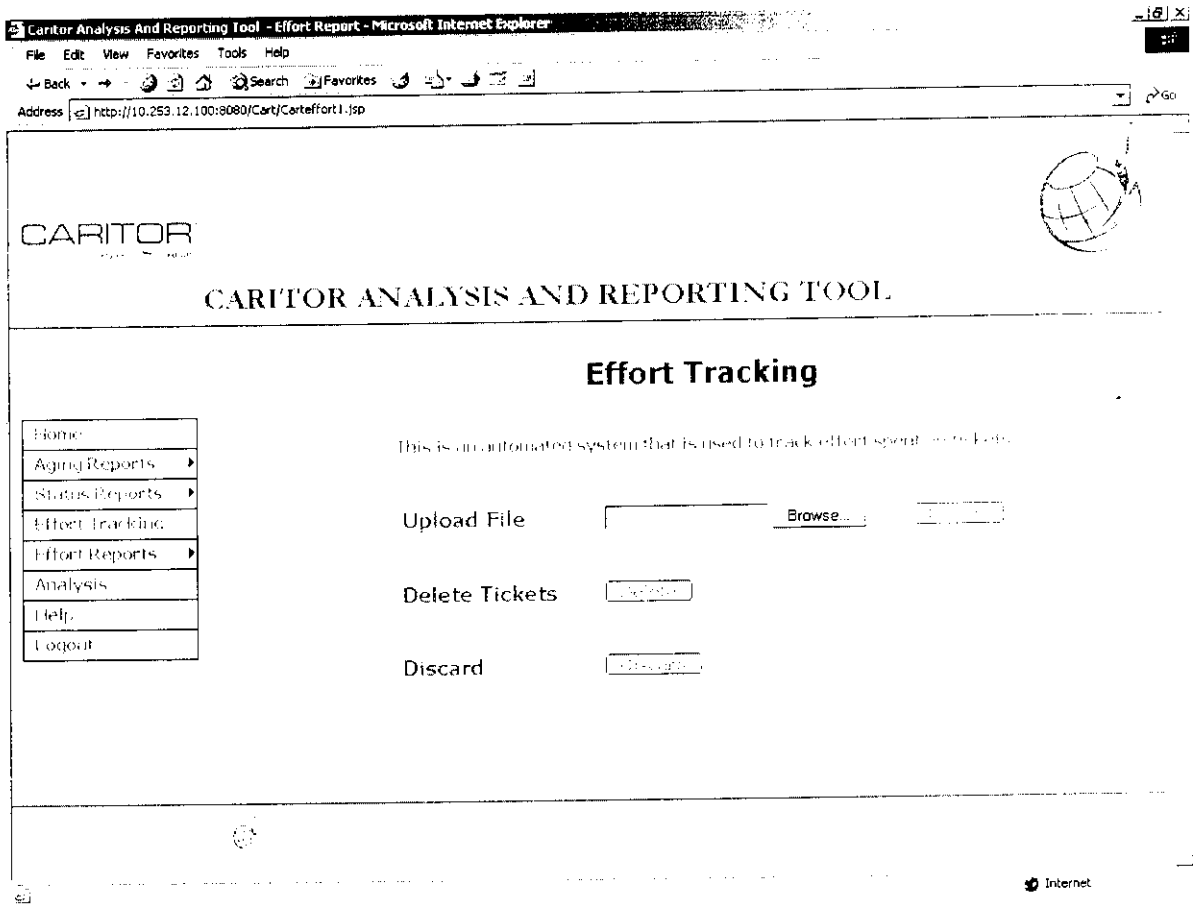


Figure A 1.7– Effort Tracking

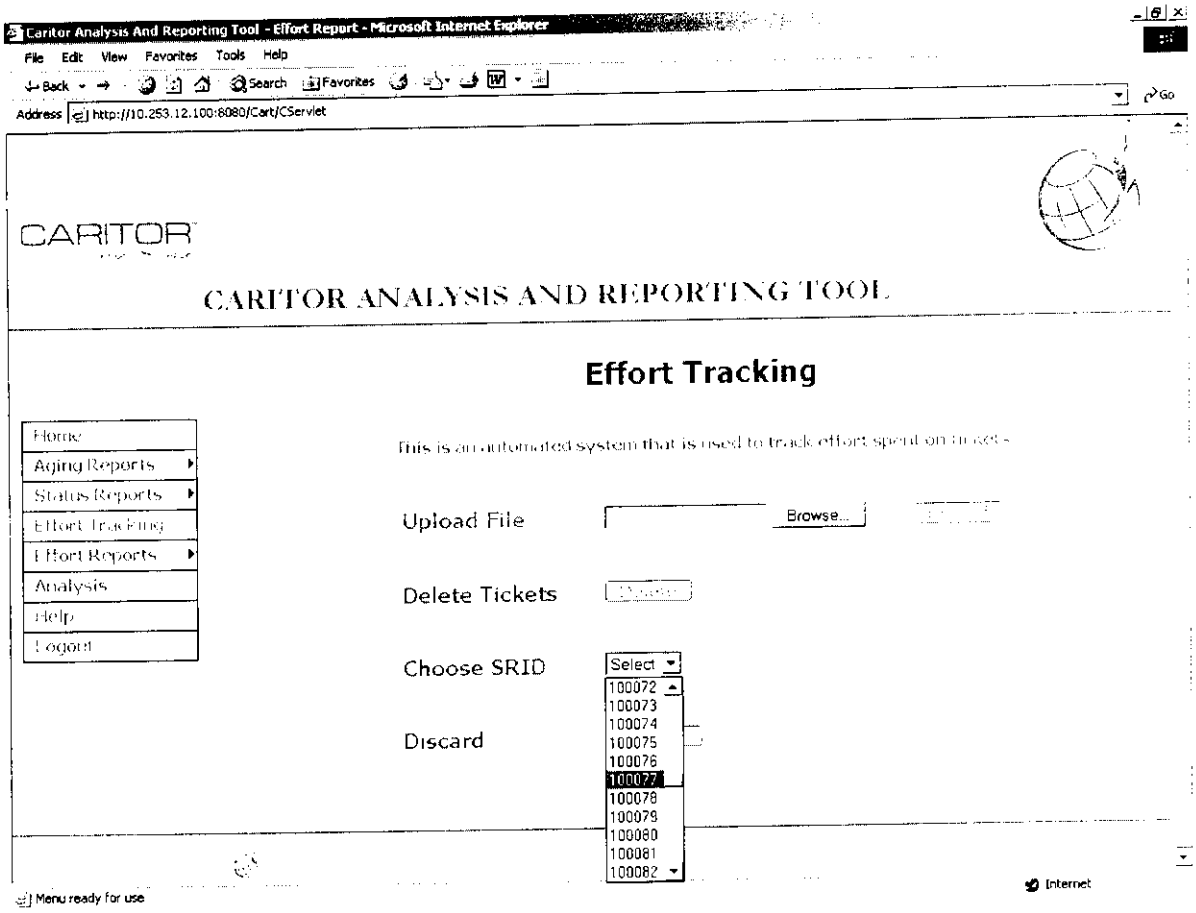


Figure A 1.8 – Listing Client Requests

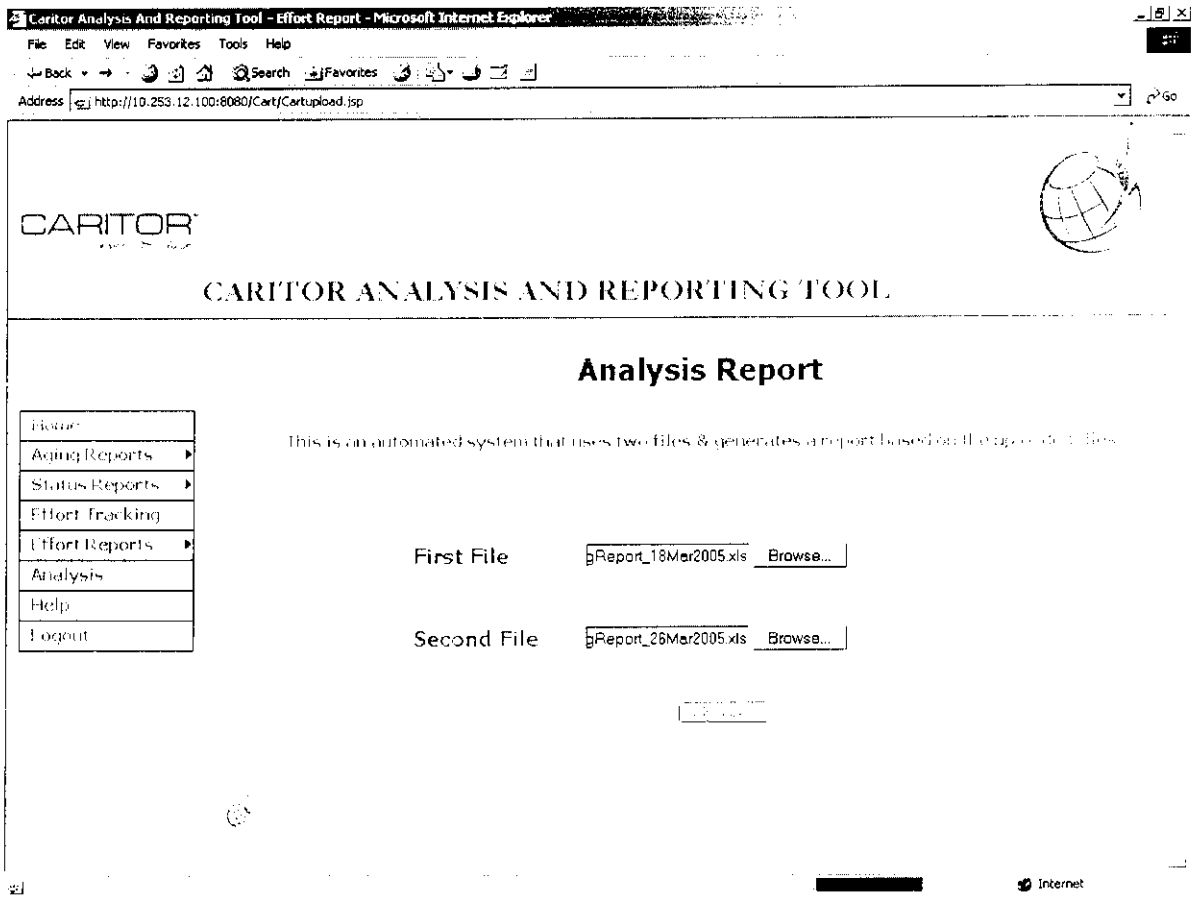


Figure A 1.9 – Analysis Page

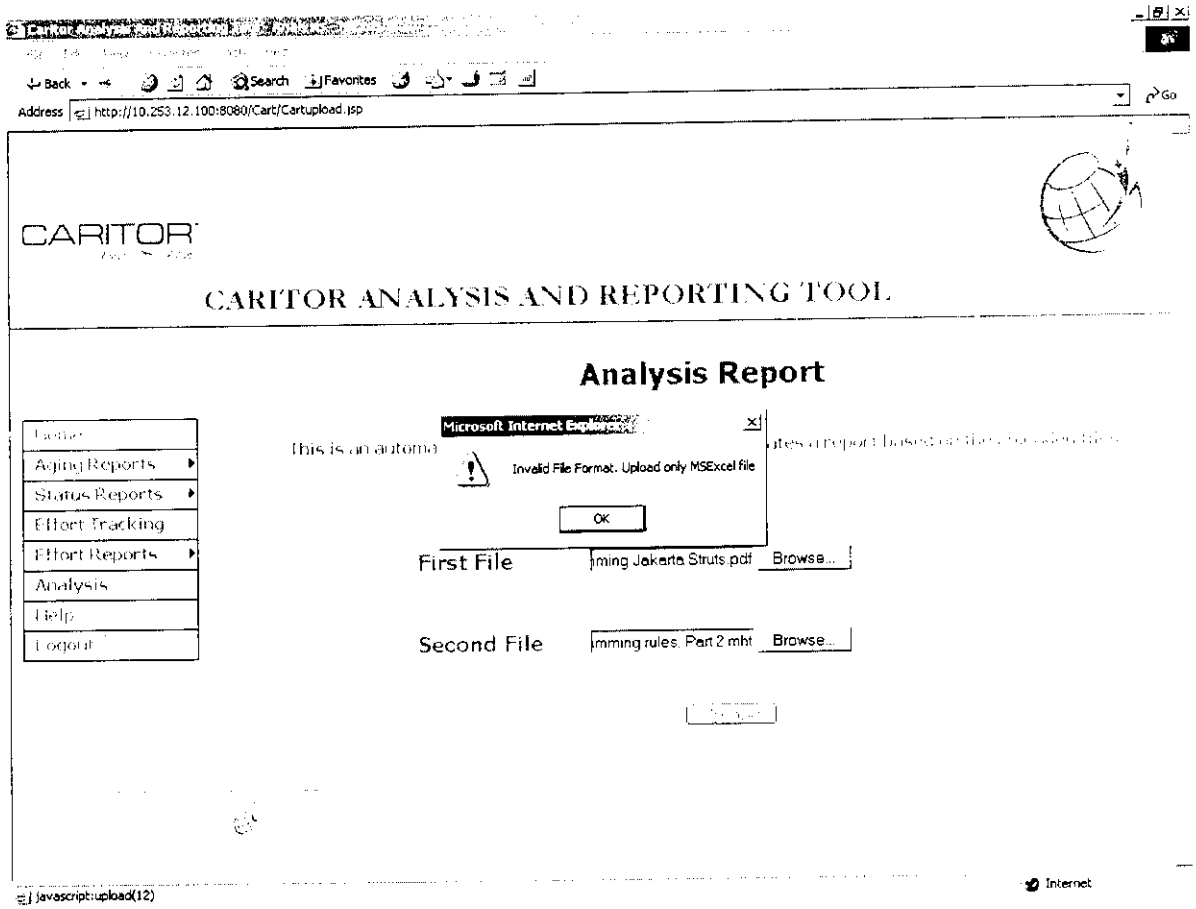
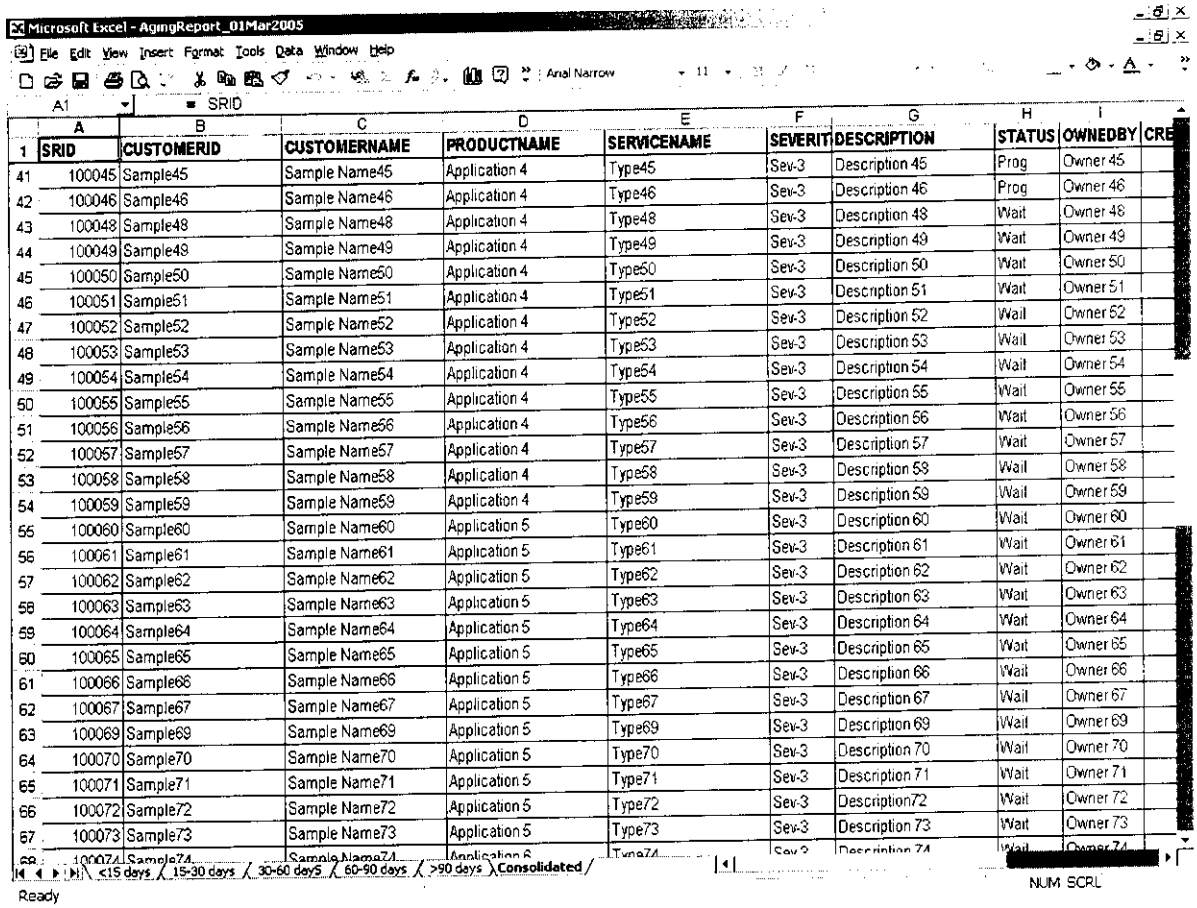


Figure A 1.10 – Invalid File Upload



Microsoft Excel - AgingReport_01Mar2005

File Edit View Insert Format Tools Data Window Help

A1 = SRID

	A	B	C	D	E	F	G	H	I	
1	SRID	CUSTOMERID	CUSTOMERNAME	PRODUCTNAME	SERVICENAME	SEVERIT	DESCRIPTION	STATUS	OWNEDBY	CRE
41	100045	Sample45	Sample Name45	Application 4	Type45	Sev-3	Description 45	Prog	Owner 45	
42	100046	Sample46	Sample Name46	Application 4	Type46	Sev-3	Description 46	Prog	Owner 46	
43	100048	Sample48	Sample Name48	Application 4	Type48	Sev-3	Description 48	Wait	Owner 48	
44	100049	Sample49	Sample Name49	Application 4	Type49	Sev-3	Description 49	Wait	Owner 49	
45	100050	Sample50	Sample Name50	Application 4	Type50	Sev-3	Description 50	Wait	Owner 50	
46	100051	Sample51	Sample Name51	Application 4	Type51	Sev-3	Description 51	Wait	Owner 51	
47	100052	Sample52	Sample Name52	Application 4	Type52	Sev-3	Description 52	Wait	Owner 52	
48	100053	Sample53	Sample Name53	Application 4	Type53	Sev-3	Description 53	Wait	Owner 53	
49	100054	Sample54	Sample Name54	Application 4	Type54	Sev-3	Description 54	Wait	Owner 54	
50	100055	Sample55	Sample Name55	Application 4	Type55	Sev-3	Description 55	Wait	Owner 55	
51	100056	Sample56	Sample Name56	Application 4	Type56	Sev-3	Description 56	Wait	Owner 56	
52	100057	Sample57	Sample Name57	Application 4	Type57	Sev-3	Description 57	Wait	Owner 57	
53	100058	Sample58	Sample Name58	Application 4	Type58	Sev-3	Description 58	Wait	Owner 58	
54	100059	Sample59	Sample Name59	Application 4	Type59	Sev-3	Description 59	Wait	Owner 59	
55	100060	Sample60	Sample Name60	Application 5	Type60	Sev-3	Description 60	Wait	Owner 60	
56	100061	Sample61	Sample Name61	Application 5	Type61	Sev-3	Description 61	Wait	Owner 61	
57	100062	Sample62	Sample Name62	Application 5	Type62	Sev-3	Description 62	Wait	Owner 62	
58	100063	Sample63	Sample Name63	Application 5	Type63	Sev-3	Description 63	Wait	Owner 63	
59	100064	Sample64	Sample Name64	Application 5	Type64	Sev-3	Description 64	Wait	Owner 64	
60	100065	Sample65	Sample Name65	Application 5	Type65	Sev-3	Description 65	Wait	Owner 65	
61	100066	Sample66	Sample Name66	Application 5	Type66	Sev-3	Description 66	Wait	Owner 66	
62	100067	Sample67	Sample Name67	Application 5	Type67	Sev-3	Description 67	Wait	Owner 67	
63	100069	Sample69	Sample Name69	Application 5	Type69	Sev-3	Description 69	Wait	Owner 69	
64	100070	Sample70	Sample Name70	Application 5	Type70	Sev-3	Description 70	Wait	Owner 70	
65	100071	Sample71	Sample Name71	Application 5	Type71	Sev-3	Description 71	Wait	Owner 71	
66	100072	Sample72	Sample Name72	Application 5	Type72	Sev-3	Description72	Wait	Owner 72	
67	100073	Sample73	Sample Name73	Application 5	Type73	Sev-3	Description 73	Wait	Owner 73	
68	100074	Sample74	Sample Name74	Application 5	Type74	Sev-3	Description 74	Wait	Owner 74	

<15 days | 15-30 days | 30-60 days | 60-90 days | >90 days | Consolidated

Ready

NUM SCRL

Figure A 1.11 – Sample Aging Report

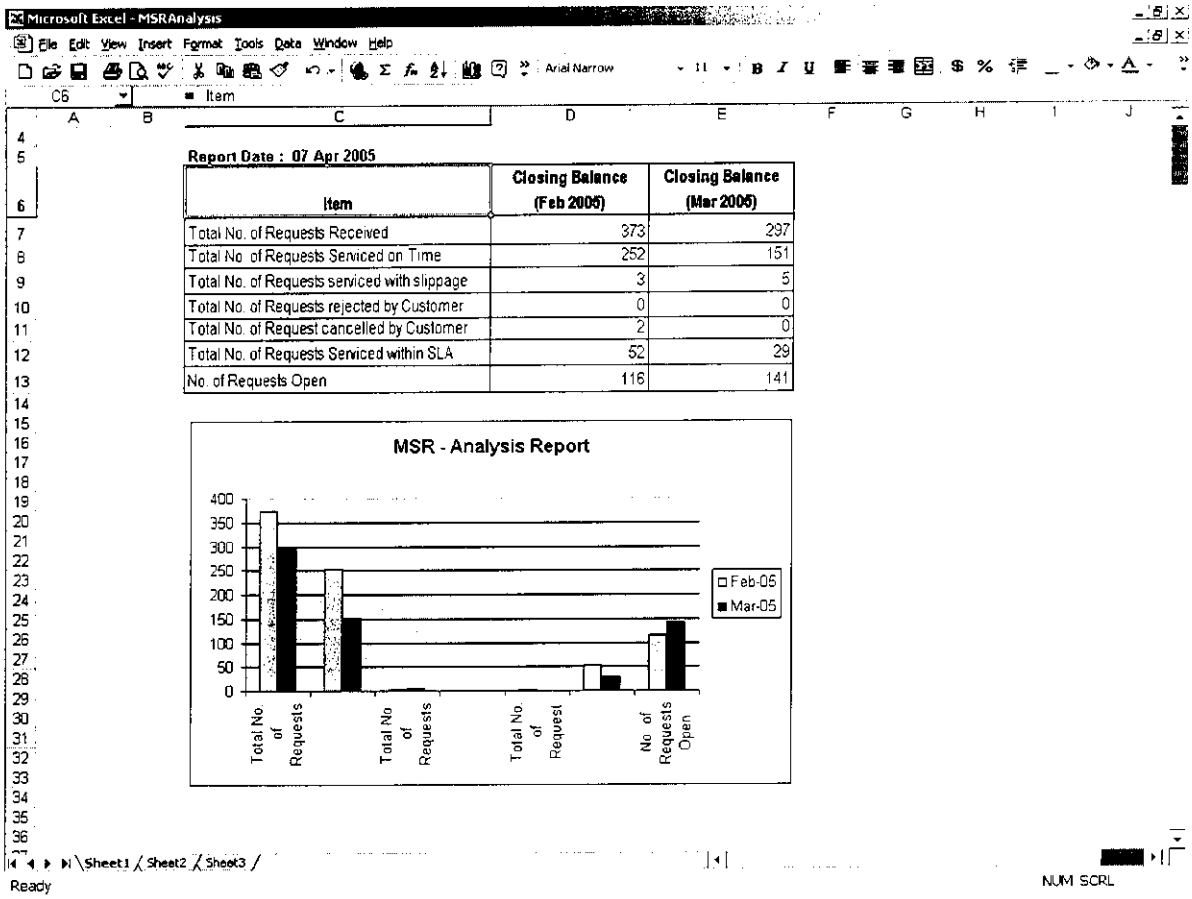


Figure A 1.12 – Sample Analysis Report

REFERENCES

1. Chuck Cavaness (2002) – 'Programming Jakarta Struts', Electronic Version, O'Reilly & Associates, Inc., Chapters 3,4 and 5

2. Danny Goodman (2001) – 'JavaScript Bible' Gold Edition, Hungry Minds, Inc., pp. 409-450,927-950

3. Hans Bergsten (2002) – 'Java Server Pages', O'Reilly & Associates, Inc.

4. Jason Hunter and William Crawford (1998) – 'Java Servlet Programming' First Edition, O'Reilly & Associates, Inc., pp. 107-123

5. Kevin Loney and George Koch (2000) – 'Oracle 8i: The Complete Reference', McGraw-Hill Companies, Inc., pp. 180-205

6. http://www.andykhan.com

7. http://java.sun.com/developer/Books/javaserverpages/Chap12.pdf (JSP Tutorial, Chapter 12)