Overall Desktop Efficiency Tracking System

By

L.Gopi

Register Number: 71206621015

Of

KUMARAGURU COLLEGE OF TECHNOLOGY

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 APPLICATIONS

ANNA UNIVERSITY
CHENNAI 600 025

June 2009

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE - 641006

BONAFIDE CERTIFICATE

Certified that this project report titled **"Overall Desktop Efficiency Tracking System"** is the bonafide work of **"Mr. L.Gopi"** (Register Number: **71206621015**) who carried out the research under my supervision. Certified further, that to the best 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.

Supervisor **Head of the Department**

Submitted to Project and Viva Examination held on 06-03-09

Internal Examiner **External Examiner**

Cognizant Technology Solutions India Private Ltd.
Mountain Mist Campus, Unit -1, STPI - IT Park,
No.363, KGISL Campus, Thudiyalur Road,
Coimbatore - 641035
Phone : (+91-0422) 3923000
Fax : (+91-0422) 3923060

www.cognizant.com

Cognizant

To whomsoever it may Concern

This is to certify that **Mr. Gopi L** a student of **Kumaraguru College of Technology** had undergone a project titled **Overall Desktop Efficiency Tracking System (ODETS)** at **Cognizant Technology Solutions India Pvt.Ltd** under the guidance of **Mr. Sreekanth Kurella** (Assistant Manager Projects).

The duration of the project was from 01/11/08 and 01/05/09.

We wish him all the success for his future endeavors

For Cognizant Technology Solutions India Pvt. Ltd.

Sunil Kumar C
Human Resources

May 10, 2009

ABSTRACT

The project "**Overall Desktop Efficiency Tracking System**" is developed for the Cognizant Technology Solutions India Private Limited The main objective/aim of the project is to track the desktop efficiency in the company. Due to this project, the project manager will be able to better gauge the productivity of his team members.

The project enables us to track the Desktop Usage Efficiency of every employee of an organization. The application's prominence is monitoring the time consumption of the employee in each application, he is working upon.

Application tracking involves, tracking the application process and the window title of the active window application. Also it stores the process and window title of the active application along with the date, duration, start time and end time of the active application.

The application tracking was done based on the API Functions and Hash table technique in the C# .Net. This technique enables us to track and store the active window application details along with the User details in the Hash table.

This project allows the Administrator to view and generate various Desktop efficiency reports of each Employee in the Company.

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without mentioning the names of people who made it possible, whose constant guidance and encouragement crowns all efforts with success.

I convey my earnest thanks to **Professor R. Annamalai,** Vice Principal, Kumaraguru College of Technology, Coimbatore for the successful completion of the project work.

I am very gladly taking this opportunity to express a special word of thanks to **Mr. S. Thangasamy**, Dean, Department of Computer Applications and to **Dr. M. Gururajan,** Head, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore for encouraging me to do this work.

I would like to express my sincere thanks and deep sense of gratitude to my Coordinator **Mr. M. MuthuKumar,** Assistant Professor, Department of MCA, and guide **Mr. M. Manikantan,** Senior Lecturer, Department of MCA, Kumaraguru College of Technology as with out his best guidance it would not have been possible for me to successfully complete this project who also gave his innovative ideas at crucial times and tremendous encouragement.

It is my pleasure to express my profound gratitude to **Cognizant Technology Solutions India Private Limited**, Coimbatore for admitting into this project. I am thankful to **Mr. Sreekanth Kurella,** Assistant Project Manager of Cognizant Technology Solutions India Private Limited, for his excellent guidance, timely suggestions and constant support in all my endeavors.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Acronyms	Full Form
ODETS	Overall Desktop Efficiency Tracking
ERP	Enterprise Resource Planning
ASP	Active Server Pages
CLR	Common Language Runtime
CTS	Common Type System
MSIL	Microsoft Intermediate Language
SQL	Structured Query Language

CHAPTER 1

1. INTRODUCTION

1.1. Company Profile

Cognizant Technology Solutions is a provider of custom information technology (IT) consulting and technology services, as well as outsourcing services for Global 2000 companies located in North America, Europe and Asia. The Company's principal services include technology strategy consulting; complex systems development; enterprise software package implementation and maintenance; data warehousing and business intelligence; application testing; application maintenance; infrastructure management, and vertically-oriented business process outsourcing (V-BPO). The Company operates in four business segments: Financial Services, Healthcare, Manufacturing/Retail/Logistics and Other, which includes communications, media and information services and high technology.

Financial Services

During the year ended December 31, 2008, the Company's Financial Services business segment represented approximately 45.6% of its total revenues. This business segment provides services to its customers operating in the industries, such as capital markets, banking and insurance. Cognizant focuses on the needs of broker/dealers, asset management firms, depositories, clearing organizations and exchanges. The Company focuses on retail and commercial banks, and diversified financial enterprises. Cognizant assists these clients in such areas as consumer lending, cards and payments, wholesale banking, risk management, investment management, corporate services and retail banking. It assists with the needs of property and casualty insurers, life insurers, reinsurance firms and insurance brokers. The Company focuses on such areas as business acquisition, policy administration, claims processing, management reporting, regulatory compliance and reinsurance.

Healthcare

During 2008, Cognizant's Healthcare business segment represented approximately 24.4% of its total revenues. This business segment provides services to its customers operating in industries, including healthcare and life sciences. The Company's healthcare service teams focus on the industry solutions, such as broker compensation, sales and underwriting systems, provider management, plan sponsor administration, electronic enrollment, membership, billing, claims processing, medical management and pharmacy benefit management. Some of Cognizant's life sciences solutions include Prescriber behavior analysis and insight, longitudinal prescription data management systems, sales force compensation systems, sales data and claims data management systems, clinical trial solutions, data mining and business intelligence solutions, e-business and data portals, and enterprise resource planning (ERP) implementation, upgrade and maintenance services.

Manufacturing / Retail / Logistics

During 2008, the Company's Manufacturing, Logistics and Retail business segment represented approximately 15.8% of its total revenues. This business segment services customers in industry groups, such as manufacturing and logistics, and retail. Some of its manufacturing and logistics solutions include supply chain management, warehouse and yard management, waste management, transportation management, optimization, portals and ERP solutions. Cognizant serves a range of retailers and distributors, including supermarkets, specialty premium retailers and mass-merchandise discounters.

1.2. Outline of Project

ODETS requires the following,

- Active Window Application Tracking
- Check for Admin User.

Active Window Application Tracking

ODETS in the Cognizant Technology Solutions normally keeps track of the Employee Access Card information and the login-logoff details of the desktop to generate Desktop Efficiency report of each Employee.

Active Window Application Tracking involves in tracking the Application usage details of Individual User in the Company. Based on this, the ODETS will generate Desktop Efficiency report in more accuracy for individual Employee.

Parameters that play a crucial role in Active Window Application Tracking are

- Name of the Active Application process
- Name of the Active Window title
- The desktop Active User Name
- The Active User Domain Name
- The Start Time of the active application
- The End Time of the application
- The Total usage duration of the active application
- Date of the active application

Check for Admin User

The ODETS requires checking for the Administrator of the desktop because the application is visible only for the Admin User. The application run as the back ground process and is not visible to the limited account user.

The ODETS imports the System class file in the C# DOT Net to verify the system Admin User. If the user is not an admin, it hides the application from the user.

CHAPTER 2

2. SYSTEM CONFIGURATION

The minimum requirements for the system to run effectively are:

2.1. Hardware Specifications

Processor	:	Intel Pentium IV 2.4 GHz
Ram	:	256 MB
Hard disk	:	40 GB hard disk
Monitor	:	SVGA or VGA
Keyboard	:	105 keys
Pointing device	:	Optical mouse

2.2. Software Specifications

Operating system	:	Windows XP
Front End	:	C# DOT Net
Back End	:	SQL Server 2000

2.3. About the Software

2.3.1. The .NET Framework

. The .NET Framework is an integral Windows component that supports building and running the next generation of applications and XML Web services. The .NET Framework is a large set of class libraries that can be used for many programming languages, like Microsoft's C#, Visual Basic, Managed C++ and more. So the first thing we notice here is that the .NET Framework class libraries can be used for more than one programming language.

Besides the Class Libraries, the Framework provides a Common Language Runtime (CLR) that manages the execution of any .NET application written in it using a .NET programming language. Simply put, the .NET Framework consists of Class Library that provides the common system services and functions that you will use and extend in your applications, and an Execution Environment that manages .NET Applications. The functionality provided by the Class Library will help you to develop Windows applications, Web applications, distributed applications and even let you integrate XML and XML Web services into your applications.

. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that promotes safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

.NET Framework to develop the following types of applications and services:

- Console applications. See Building Console Applications.
- Windows GUI applications (Windows Forms). See Windows Forms.

- Windows Presentation Foundation (WPF) applications. See Introduction to Windows Presentation Foundation.
- ASP.NET applications. See Creating ASP.NET Web Pages.
- Web services. See Creating Web Services in Managed Code.
- Windows services. See Introduction to Windows Service Applications.

Advantage of .NET

- simple and faster system development
- rich object model
- enhanced built in functionality
- Many different ways to communicate with the outside world
- Integration of different languages into one platform
- Easy deployment and execution
- Wide range of scalability
- Interoperability with existing applications
- Simple and easy to build sophisticated development tools
- Fewer bugs
- Potentially better performance

C# .NET

C# .Net is a new programming language from Microsoft designed specifically to target the .NET Framework. Microsoft's .NET Framework is a runtime environment and class library that dramatically simplifies the development and deployment of modern, component-based applications.

C# supports concepts such as inheritance, encapsulation, polymorphism, and interface-based programming. C# supports common C, C++, and Java language constructs such as classes, structures, interfaces, and enums, as well as more novel constructs such as delegates, which provide a type-safe equivalent to C/C++ function pointers, and custom attributes, which allow annotation of code elements with additional information.

In addition, C# incorporates features from C++ such as operator overloading, user-defined conversions, true rectangular arrays, and pass-by-reference semantics that are currently missing from Java.

Unlike most programming languages, C# has no runtime library of its own. Instead, C# relies on the vast class library in the .NET Framework for all its needs, including console I/O, network and file handling, collection data structures, and many other facilities.

Compilation of c#

When we write an Application in C# the compiler doesn't generate a binary file (like traditional compilers) but rather it generates a file (for now you can call this file a .NET Module) that contains Microsoft Intermediate Language (MSIL) code.

The Common Language Runtime (which is the execution environment for the .NET Framework Applications) has a compiler called Just-In-Time, or the JIT Compiler, which will compile the MSIL Instructions into native CPU Instructions upon the first execution of your application methods on a one-by-one basis. I think this needs a little more explanation.

Common Type System

Microsoft has designed a Common Type System (or CTS) that defines most of the data types that have been used by most popular programming languages. The Common Type System doesn't define the syntax that programming languages use, nor does it defines the keywords; it just defines the Common Data Types for all the languages that will be managed by the .NET CLR Environment.

Assemblies

Every time you compile C# code the file produced (.dll file or .exe file) is called an Assembly. The Assembly is a self-describing component; when you compile the source code files the compiler emits information about the types contained in the application, along with other information that is needed by the runtime (such as what files are needed to run the application).

The Assembly is a logical group of one or more files, so you can have one assembly represented as MyAssembly.dll file, and you can have another assembly that consists of more than one file (or module).

2.3.2. SQL Server 2000

Database

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, SQL Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

Relational Database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the tables and enables you to define relationships between the tables.

Foreign Key

When a field is one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

Data Abstraction

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributed and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

Features of SQL Server

- Enterprise wide Data Sharing
- Portability
- Open Systems
- Distributed Data Sharing
- Unmatched Performance

CHAPTER 3

3. SYSTEM ANALYSIS

3.1. Problem in existing system

- There is no any application usage details of the Employee

- They were generating desktop efficiency reports based on the Employee Access Card information and the login-logoff details of the desktop.

- The existing ODETS will produce the efficiency reports with less accuracy.

- Management cannot be able to forecast the performance of the Project Team.

3.2. Solution of these problems

- Tracking the Employee access card information.

- Employee desktop login-logoff details.

- Tracking Application usage details of the Employee.

- Hiding the Application from normal user.

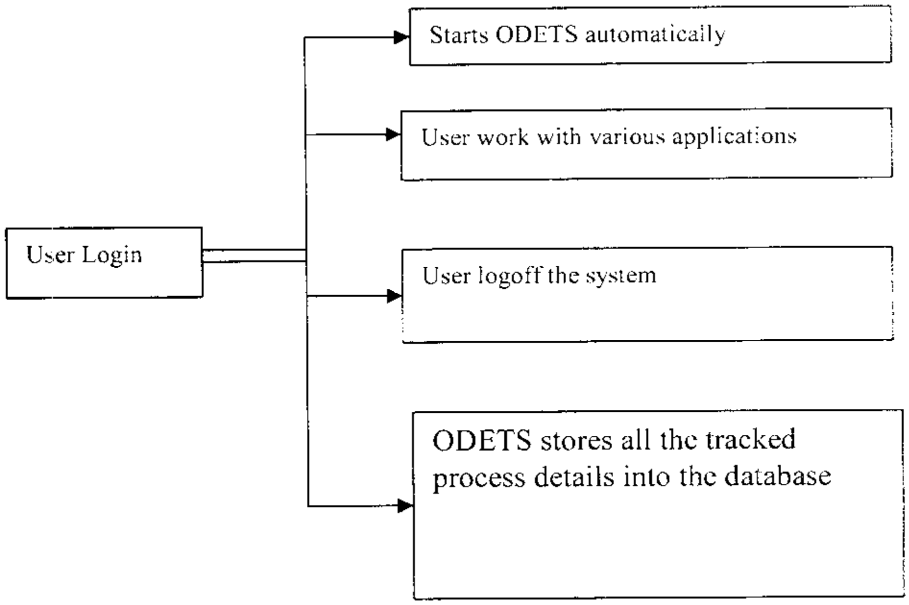
The adding of the functionality enables the application to generate desktop efficiency report for individual employee in more accuracy. Project Team Members in the company will be able to know the time spent by them in various applications on a daily basis. Project Manager in the company will be able to better gauge the productivity of his team members.

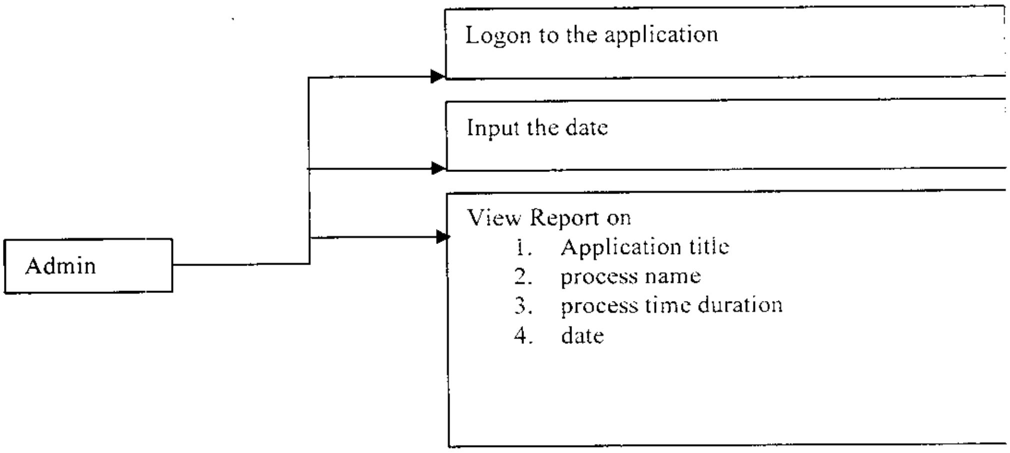
CHAPTER 4

4. SYSTEM DESIGN

4.1. Information Flow

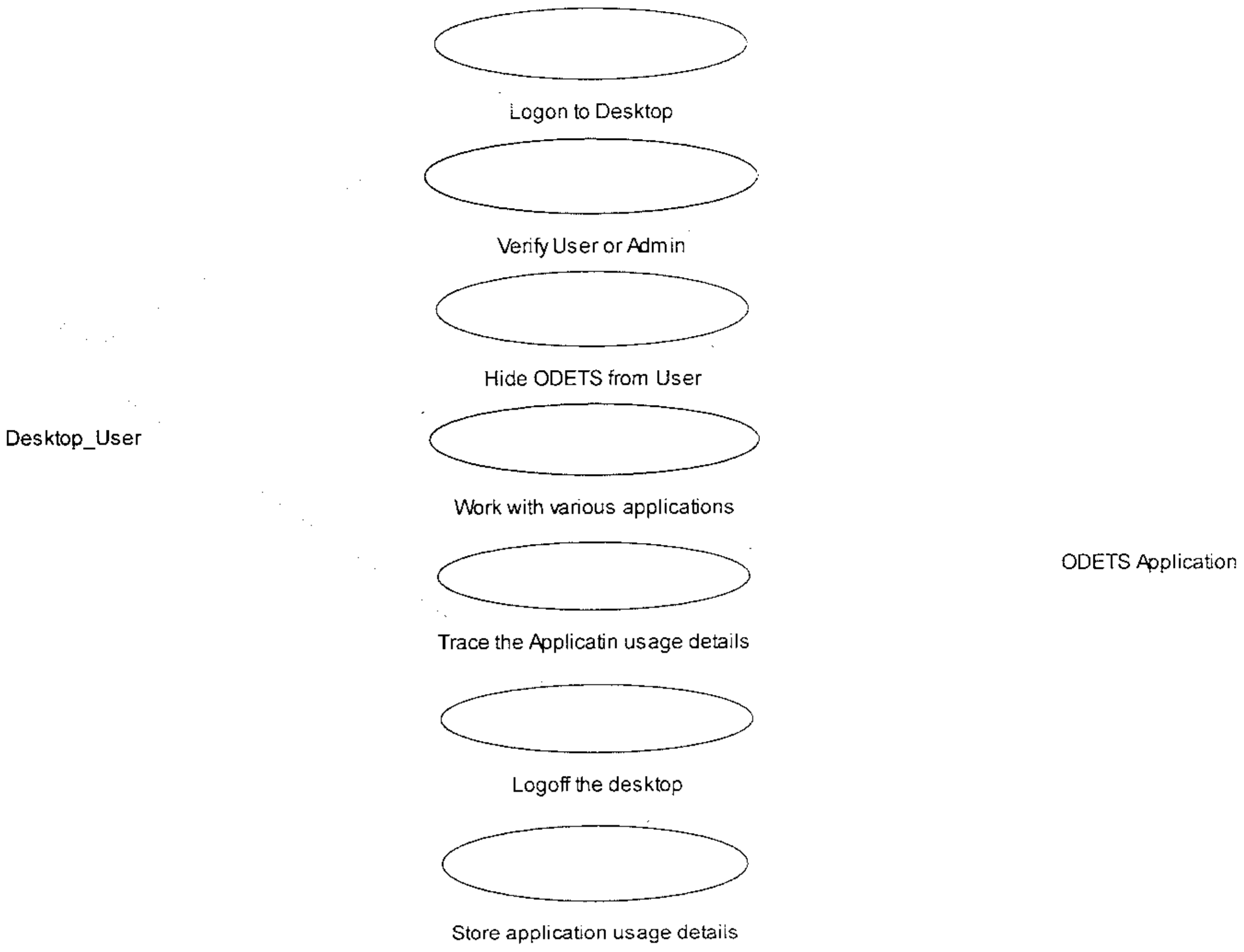
Data flow diagrams are provided to understand the working of the system.


User Login
Starts ODETS automatically
User work with various applications
User logoff the system
ODETS stores all the tracked process details into the database

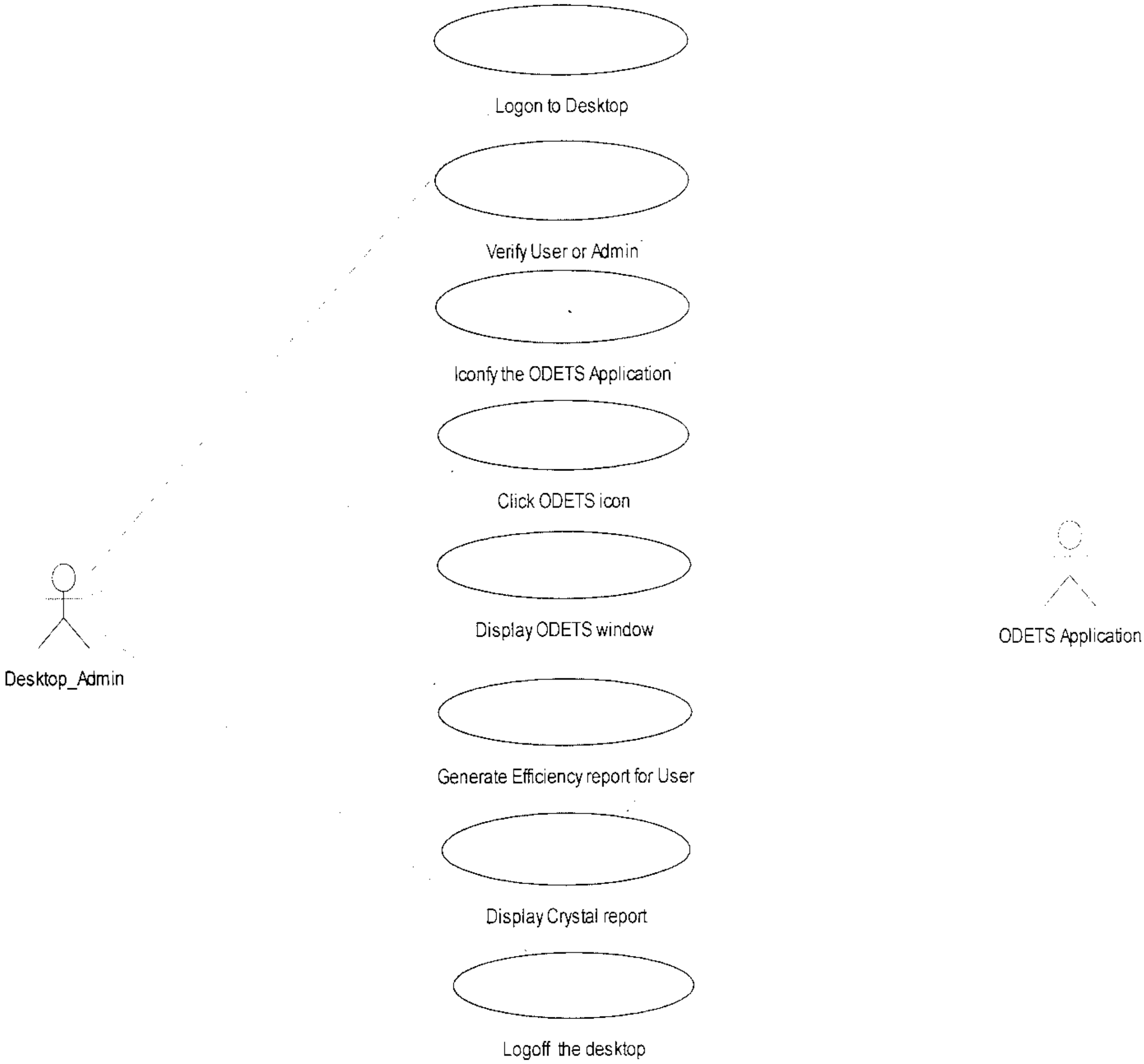


4.2. Usecase Diagram

Desktop_User:

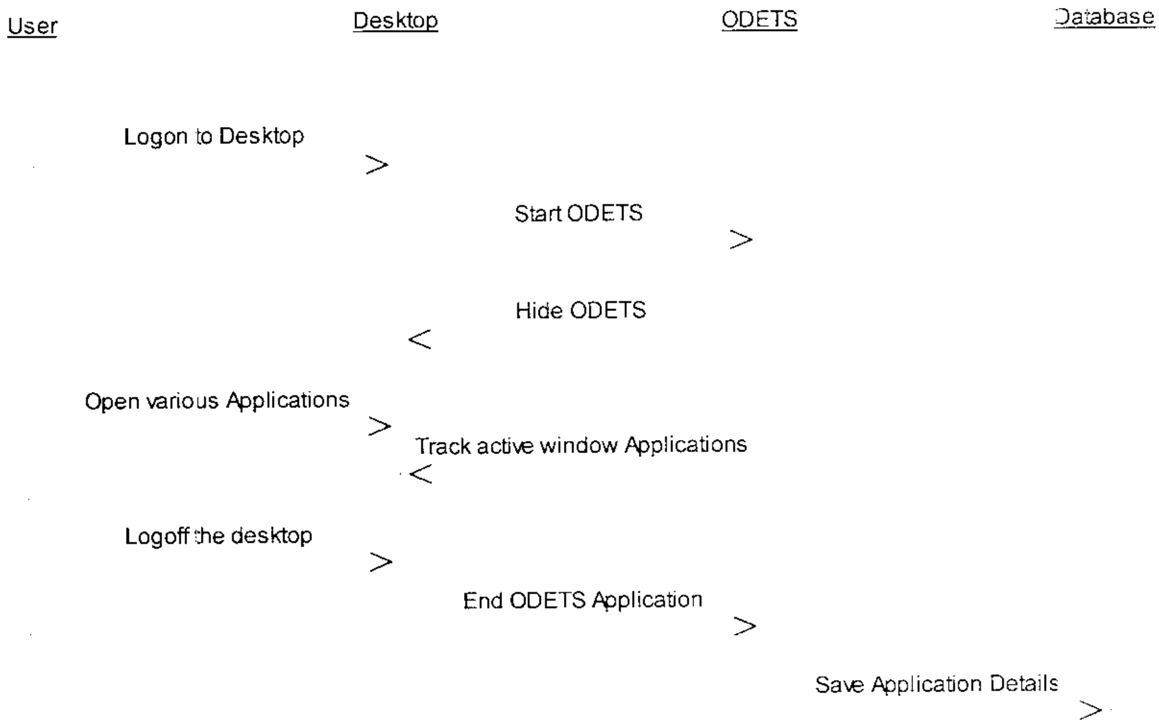


Desktop_Administrator:

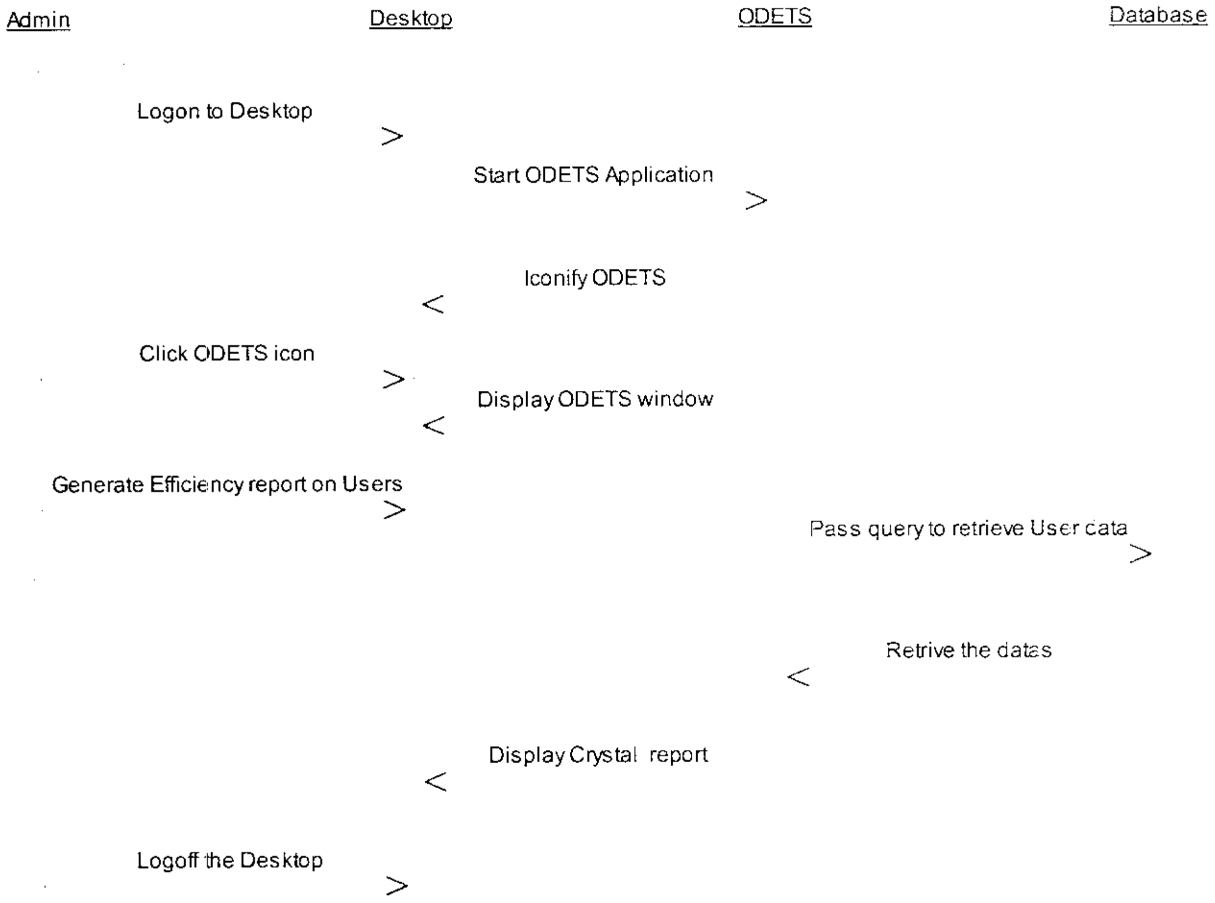


Sequence Diagram

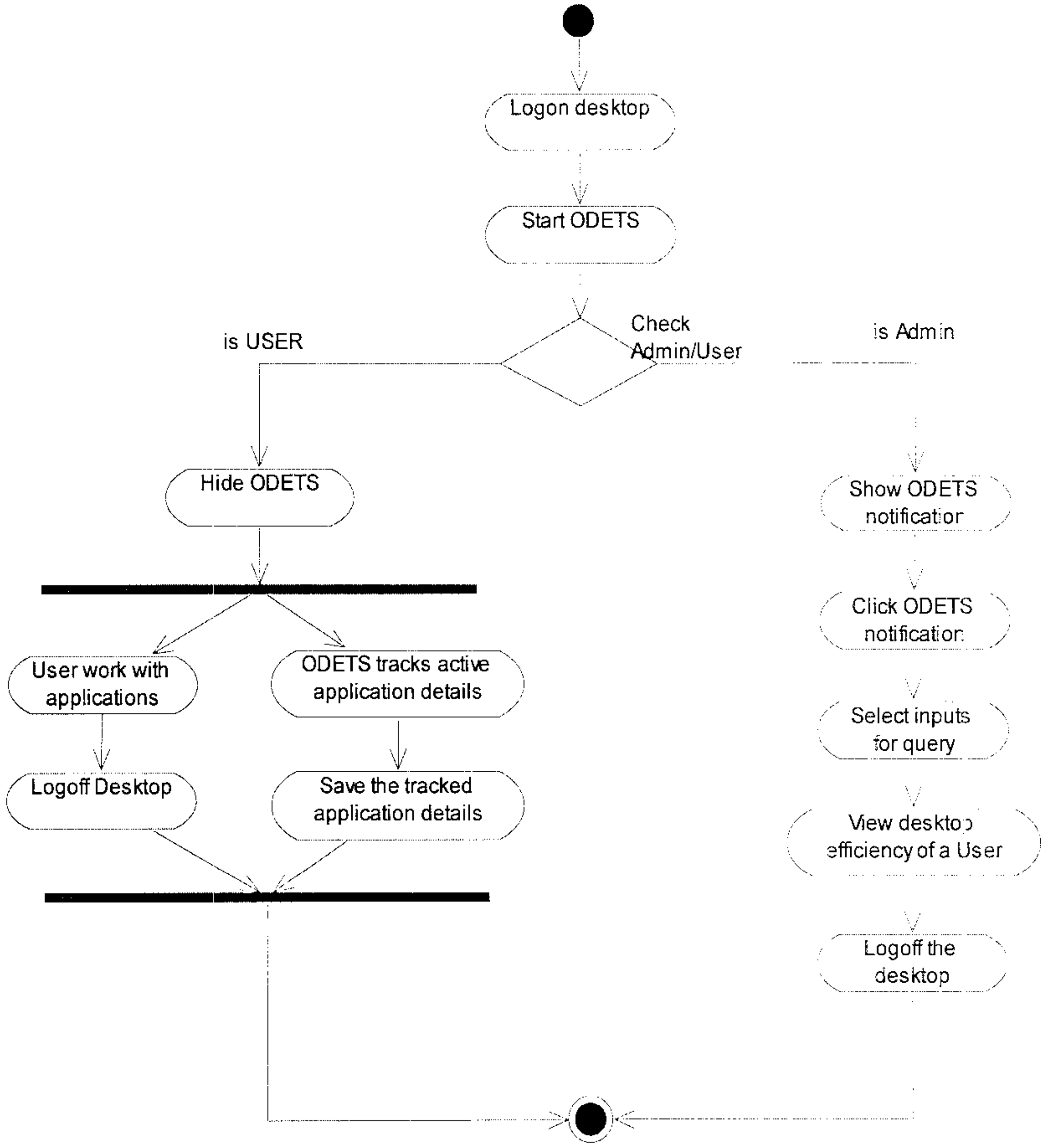
Desktop_User:



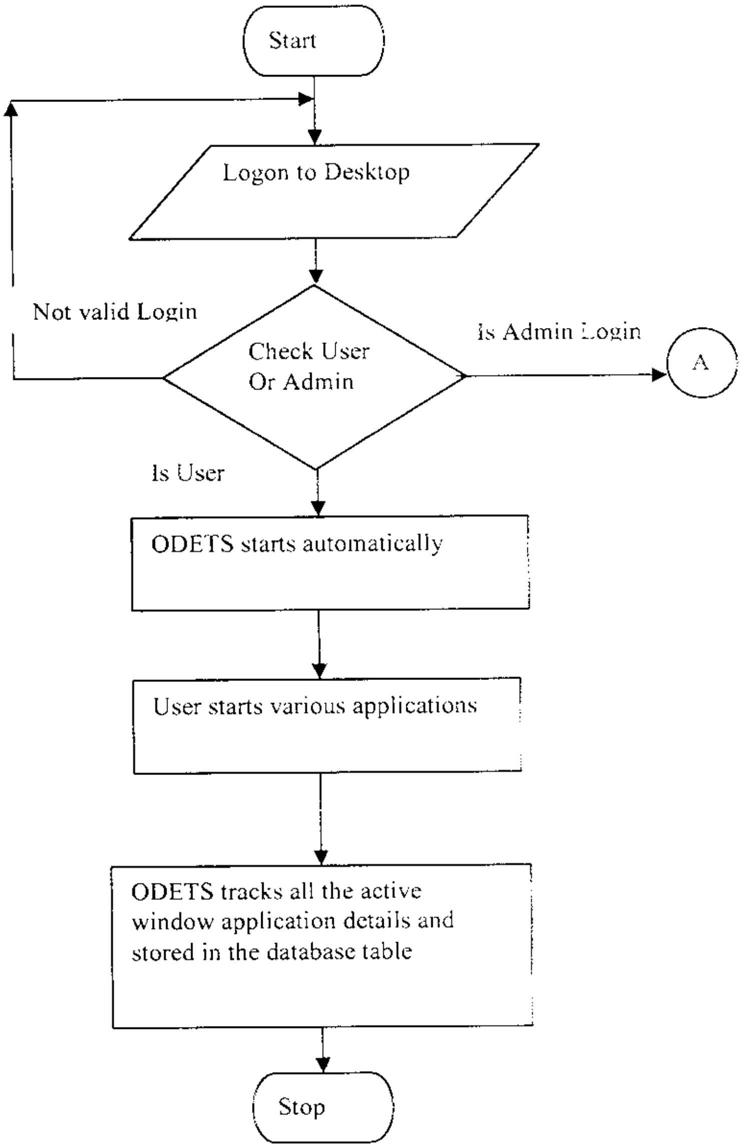
Desktop_Administrator:

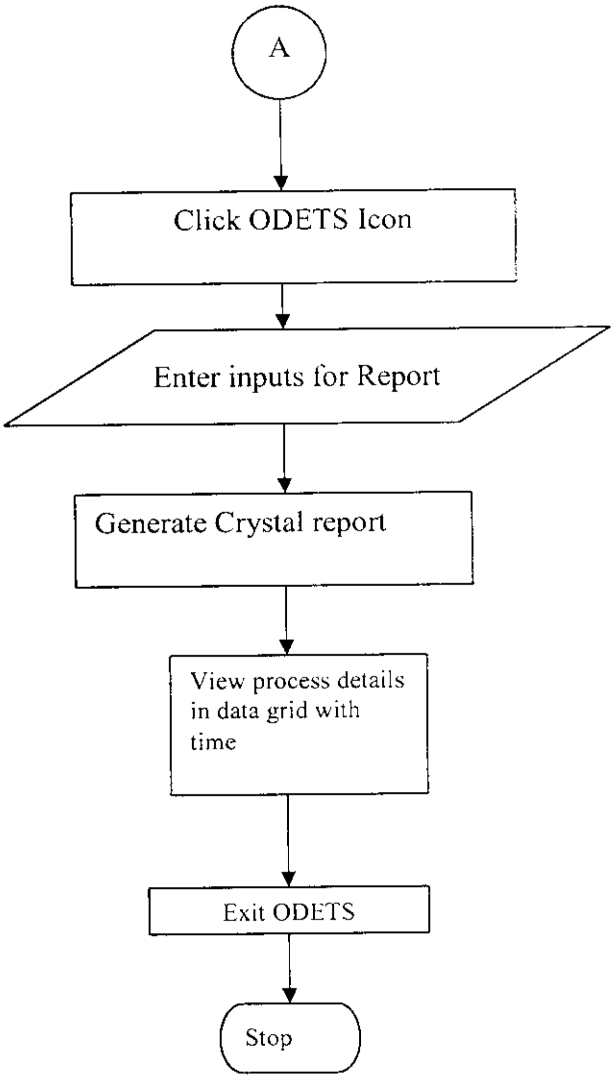


Activity Diagram



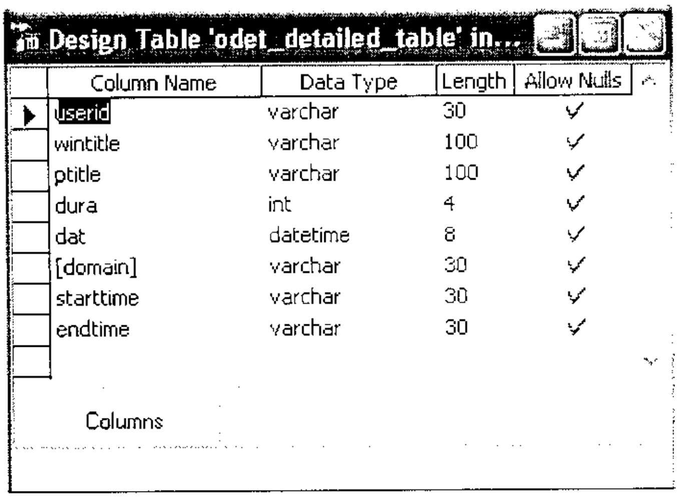
4.3. System Flow Diagram


Start
Logon to Desktop
Not valid Login
Check User
Or Admin
Is Admin Login
A
Is User
ODETS starts automatically
User starts various applications
ODETS tracks all the active
window application details and
stored in the database table
Stop


A
Click ODETS Icon
Enter inputs for Report
Generate Crystal report
View process details in data grid with time
Exit ODETS
Stop

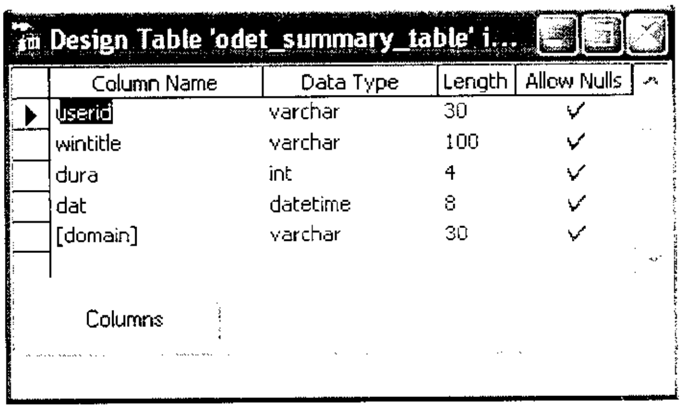
4.4. Table Structurs

Odet_detailed_table


Design Table 'odet_detailed_table' in...

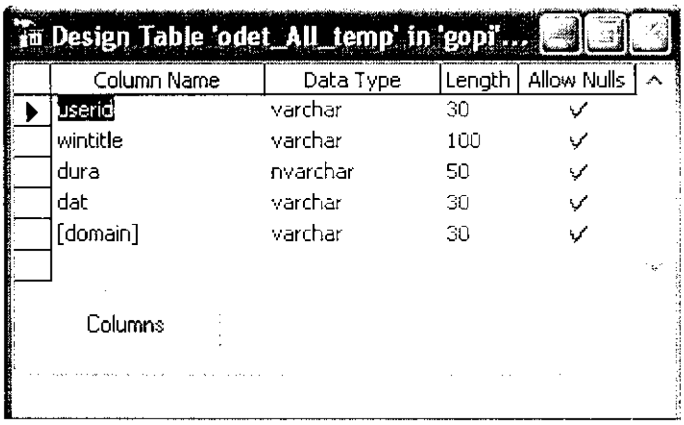
Column Name	Data Type	Length	Allow Nulls
userid	varchar	30	✓
wintitle	varchar	100	✓
ptitle	varchar	100	✓
dura	int	4	✓
dat	datetime	8	✓
[domain]	varchar	30	✓
starttime	varchar	30	✓
endtime	varchar	30	✓

Odet_summary_table


Design Table 'odet_summary_table' i...

Column Name	Data Type	Length	Allow Nulls
userid	varchar	30	✓
wintitle	varchar	100	✓
dura	int	4	✓
dat	datetime	8	✓
[domain]	varchar	30	✓

Odet_All_temp



Design Table 'odet_All_temp' in 'gopi'...

Column Name	Data Type	Length	Allow Nulls
userid	varchar	30	✓
wintitle	varchar	100	✓
dura	nvarchar	50	✓
dat	varchar	30	✓
[domain]	varchar	30	✓

CHAPTER 5

5. SYSTEM DEVELOPMENT

5.1. Introduction

System development is a series of operations performed to manipulate data to produce output from computer system. This aim at translating the design of the system produced during the design phase into code in user programming language. A modular approach is used for the development of the software.

The development phase for the project was created from the specifications created during the design phase. A principal activity of the development phase is coding and testing the computer program that make up the computer program component of the overall system. Other important activities include implementation, planning, equipment acquisition and system testing. The development phase concludes with the report and review.

5.2. Module description

- Odets_Detailed
- Odets_Summary
- Odets_All

5.2.1. Odets_Detailed

The Odets_Detialed is the detailed table that contains all the details of the Active window applications that are stored by the ODETS applications. It contains the details such as Username, domain name, Application name, Window title, date, start time and end time of the application.

5.2.2. Odets_Summary

The Odets_Summary contains the summarized details of the Active application details that are stored by the ODETS Application.

CHAPTER 6

6. SYSTEM IMPLEMENTATION

Implementation is the state in the System where the theoretical design is turned into a working system. The system can be implemented only after through testing is done and if found to work according to the specification. The most crucial stage in achieving a new successful system relies in giving confidence for the users on the new system that will work efficiently and effectively.

It involves careful planning, investigation of the current system and to constraints on implementation, design of methods to achieve the changeover, an evaluation of changeover methods apart from planning. System Analysis and design efforts will be more complex system being used for writing program code.

Program Code Preparation

One of the important development activities is the code of programming. The system Usecases, Sequence, Activity and other channels are converted to modular programs; they have to be complied, tested and debugged.

CHAPTER 7

7. SYSTEM TESTING

7.1. Introduction on Testing:

Software testing is a critical element of software quality assurance and represents the ultimate reviews of specification, design and coding testing represents interesting anomaly for the software. During earlier definition and development phases, it was attempted to build software from an abstract concept to tangible implementation.

Te testing phase involves the testing of developed system using various test data. Preparation of the test data plays vital role in the system testing. After preparing the test data the system under study was tested using those data. While testing the system, errors were found and corrected by using the following testing steps and corrections are also noted for future use. Thus, a series of testing is performed for the proposed system was ready for the implementation.

7.2. Unit Testing:

Unit testing focuses verification efforts even in the smallest design in each module. This is also known as "Unit Testing". Using the test plans, prepared in design phase of the system development as a guide, important control paths are tested to uncover error with in the boundary of the proposed project. In this testing each module is found to be working satisfactory, as regard to expected output from the proposed project.

7.3. Integration Testing:

Data can be lost across an interface, one module can have an adverse effect on the other sub-functions, when combined may not produce the desired functions. Integrated testing is the systematic testing to uncover the errors within the interface.

This testing is done with simple data and developed systems has run successfully with this simple data .In the proposed project, each module will run successfully and produce valid outputs in the integration testing.

7.4. Validation Testing:

Software validation is achieved through a series of black box testing that demonstrate conformity with requirements. A test plans out lines the classes of testes to be conducted and test procedure defines specific test cases that will be used to demonstrate conformity with requirements.

At the culmination of the black box testing, software is completely assembled as a package. Interfacing errors have been uncovered and correct final series of software Validation test begins. Validation test can is defined with simple definition that validation succeeds when the software functions in a manner that can be reasonably accepted by the user. In proposed project, validation testing will finally occur and it produces required outputs to the user.

7.5. Security Testing:

If the users have to enters the system we have to specify the user name and password. When the user enters the user name and the password, checking it with the database. If it matches, then only the user is allowed to access the page. Otherwise he is denied accesses and there by provides a strong security. In the proposed project, the username is **Admin** and password is **Admin$**.

7.6. White Box Testing:

White box testing some times called glass box testing, is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineering can derive test cases.

- Guarantee that all independent paths with in a module have been exercised at least once,
- Exercise all logical decisions on their true and false sides,
- Execute all loops at their boundaries and with in their operational bounds, and Execute internal data structure to ensure there validity.

7.7. Black Box Testing:

Black box testing also called behavior testing focuses on the functional requirements of the software. That is, black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

Black box testing attempts to find errors in the following categories:

- In correct or missing functions,
- Interface errors,
- Errors in data structure or external data base access.
- Behavior or performance errors and

CHAPTER 8

8. CONCLUSION

This project **Overall Desktop Efficiency Tracking System** was prepared with guidance and discussion with personal involved in this project and technical staff.

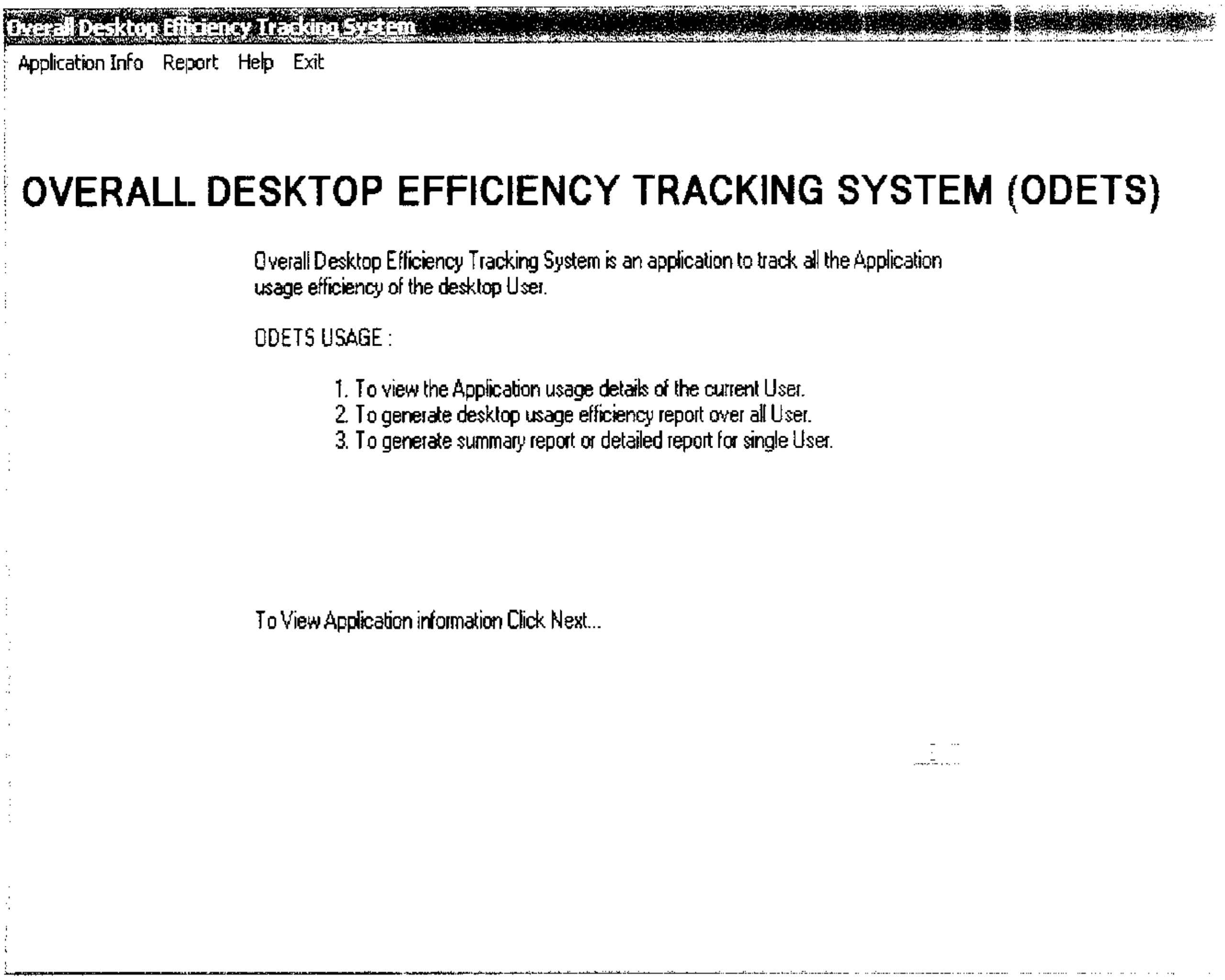
This project has developed with maximum care. It has been developed with an eye on expansion and flexibility at every stage of the module. This is, developed to meet almost all the requirements of the user. This will add accuracy to the existing system. This is more advantageous over the existing system as it assigns 100% accuracy in generating the Efficiency reports among the various desktop Users in the company. It is accurate and very fast and produces various kinds of detailed reports.

Further enhancements can be made at any later point of time. Reports can be represented in all-necessary perspectives. Added options can be included in designing reports. This project is developed in a user friendly manner in GUI software. The user can perform the operations such as addition, deletion, and modification of the database very easily but in a specified manner.

CHAPTER 9

9. APPENDICES

Welcome Screen



OVERALL DESKTOP EFFICIENCY TRACKING SYSTEM (ODETS)

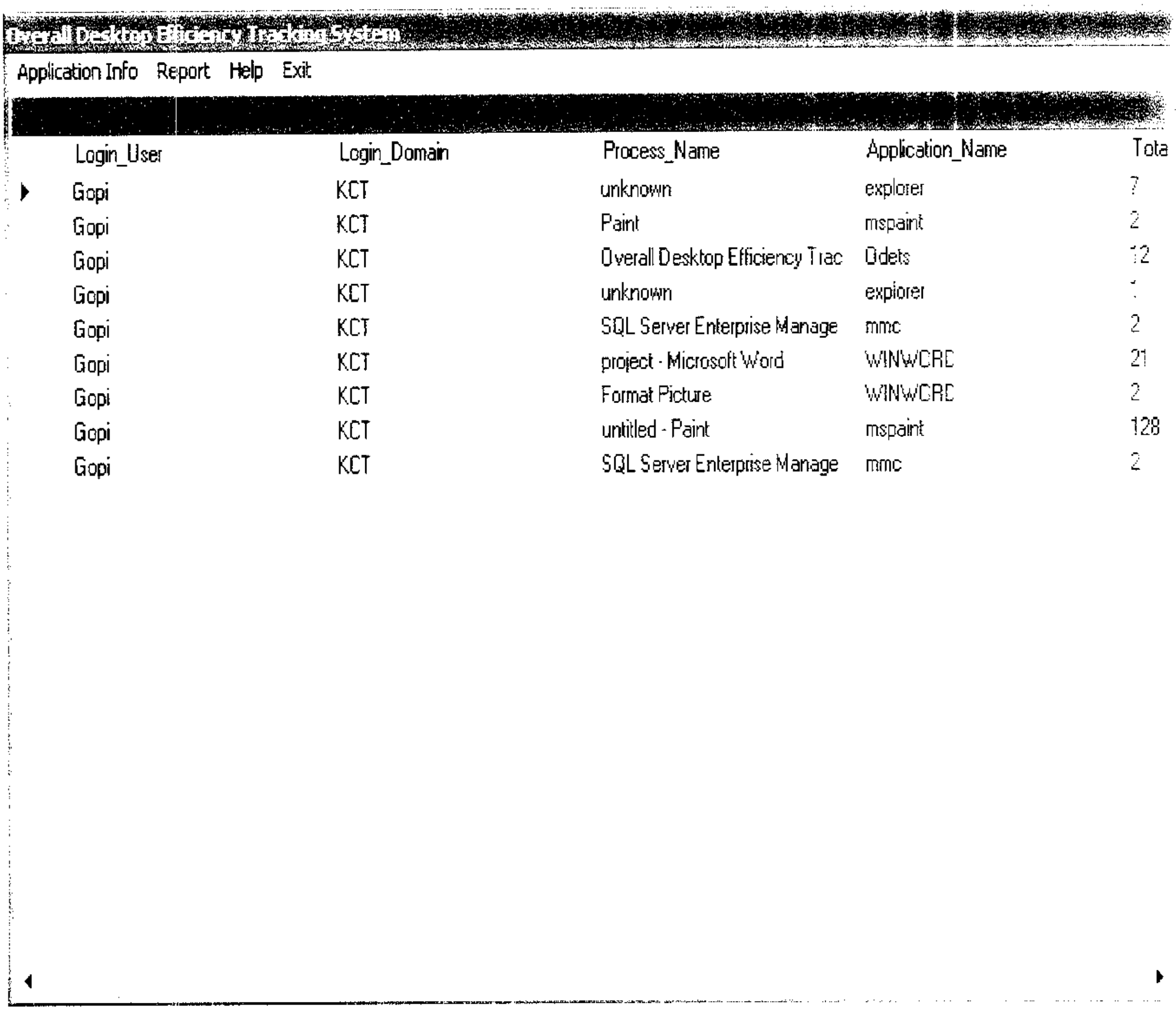
Overall Desktop Efficiency Tracking System is an application to track all the Application usage efficiency of the desktop User.

ODETS USAGE :

1. To view the Application usage details of the current User.
2. To generate desktop usage efficiency report over all User.
3. To generate summary report or detailed report for single User.

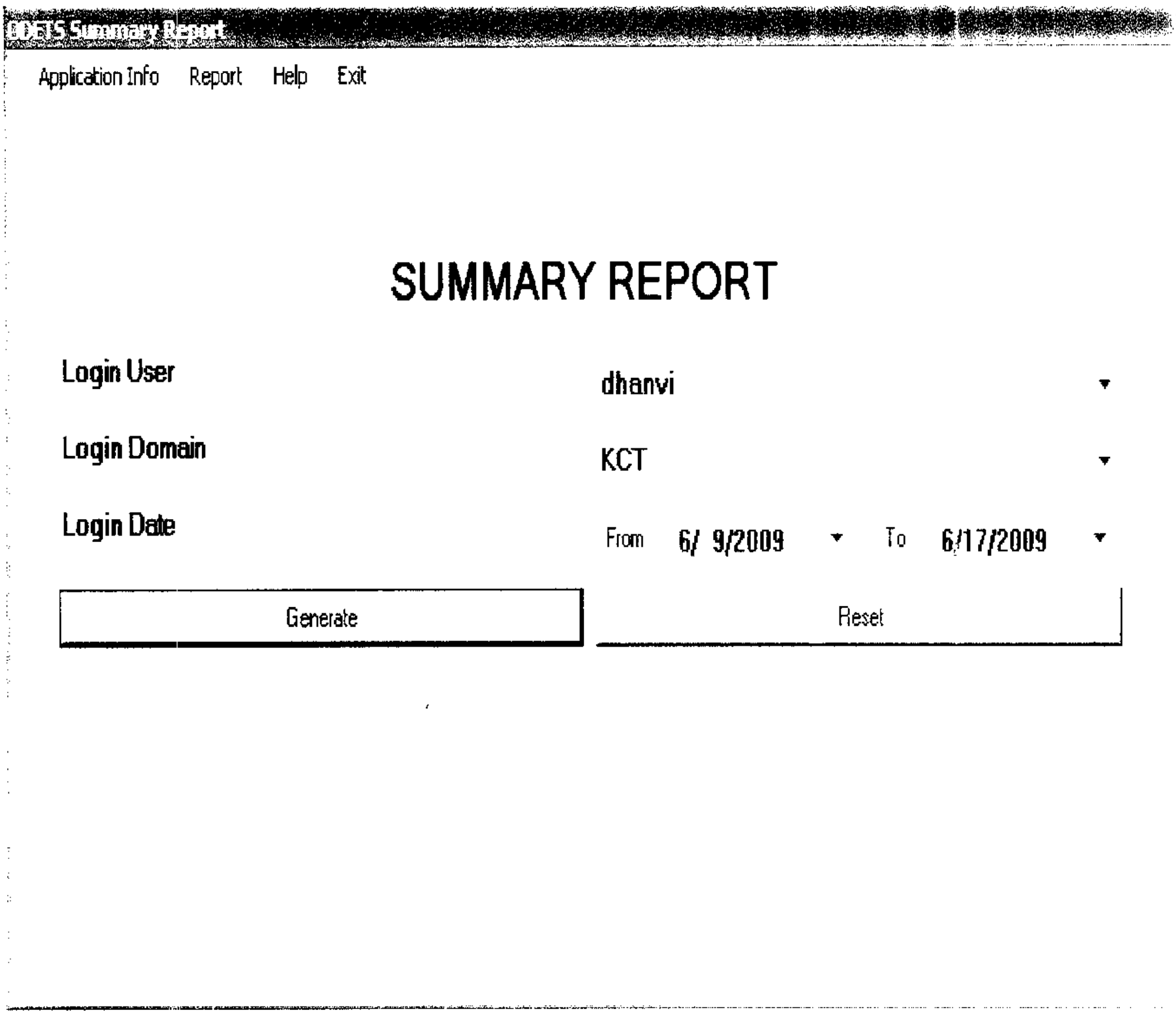
To View Application information Click Next...

ODETS Screen



Login_User	Login_Domain	Process_Name	Application_Name	Tota
Gopi	KCT	unknown	explorer	7
Gopi	KCT	Paint	mspaint	2
Gopi	KCT	Overall Desktop Efficiency Trac	Odets	12
Gopi	KCT	unknown	explorer	1
Gopi	KCT	SQL Server Enterprise Manage	mmc	2
Gopi	KCT	project - Microsoft Word	WINWORD	21
Gopi	KCT	Format Picture	WINWORD	2
Gopi	KCT	untitled - Paint	mspaint	128
Gopi	KCT	SQL Server Enterprise Manage	mmc	2

Summary Report Screen



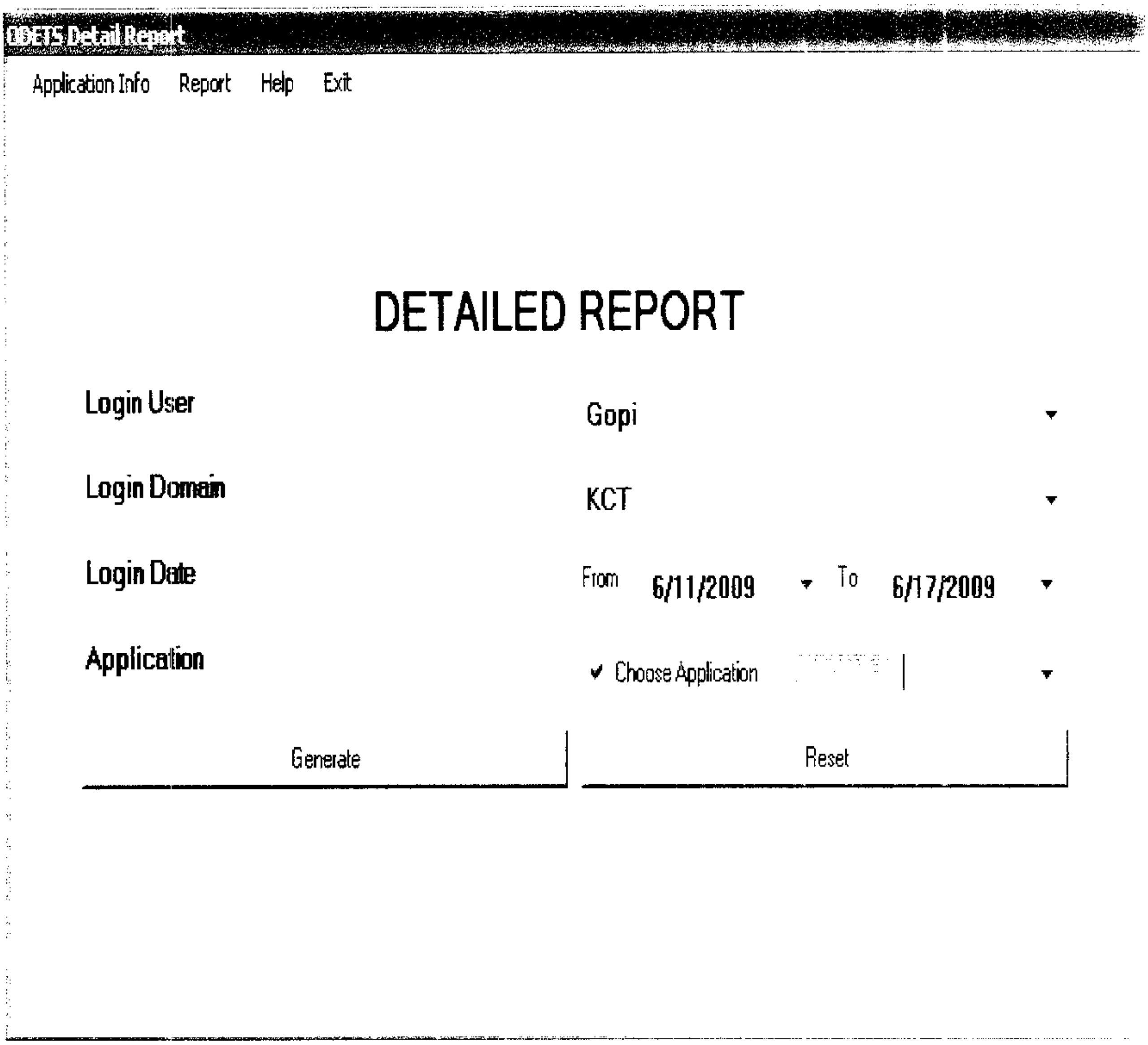
SUMMARY REPORT

Login User: dhanvi

Login Domain: KCT

Login Date: From 6/ 9/2009 To 6/17/2009

Detailed Report Screen



DETAILED REPORT

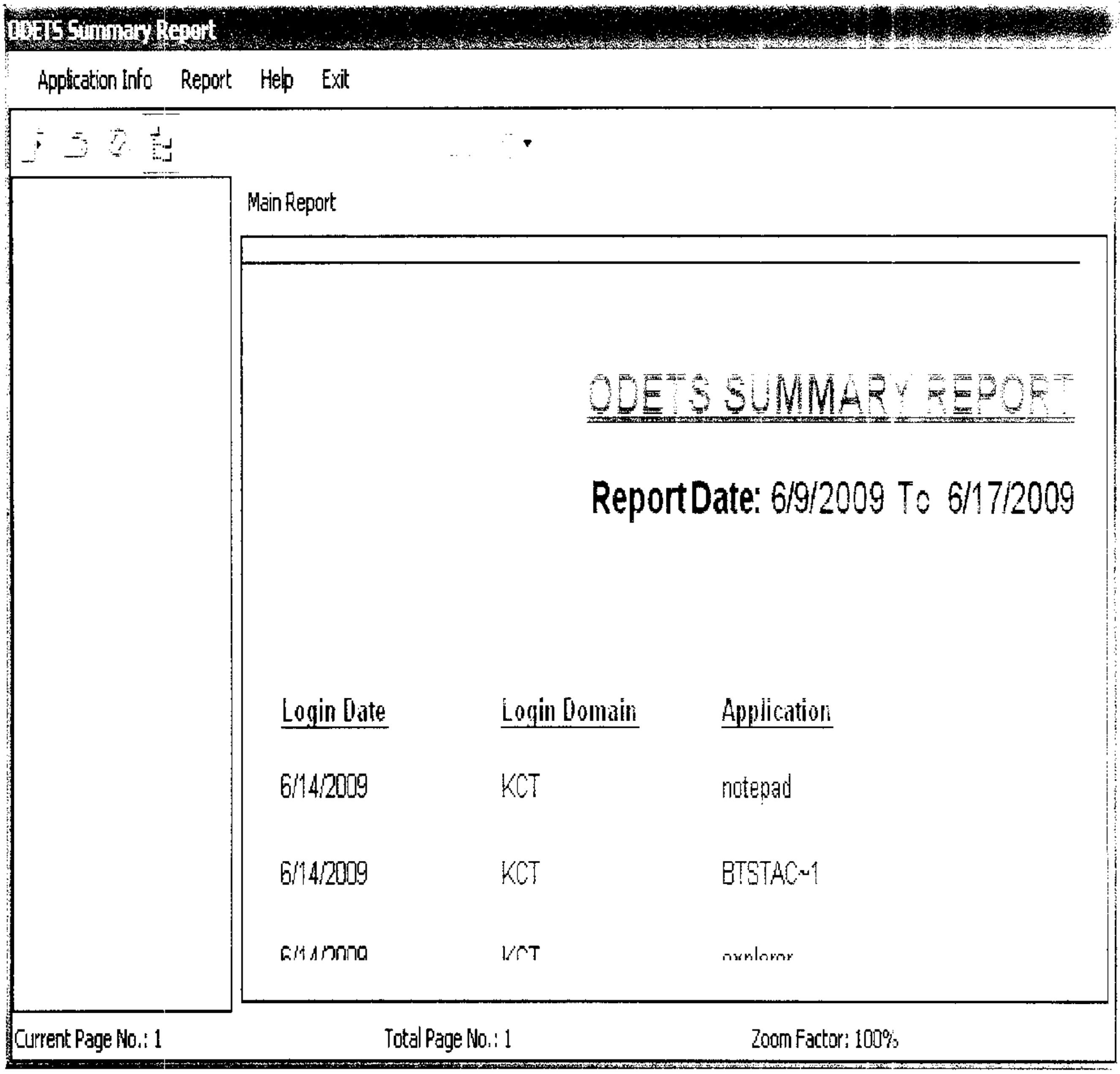
Login User: Gopi

Login Domain: KCT

Login Date: From 6/11/2009 To 6/17/2009

Application: ✓ Choose Application

ODETS Crystal Report Screen



ODETS SUMMARY REPORT

Report Date: 6/9/2009 To 6/17/2009

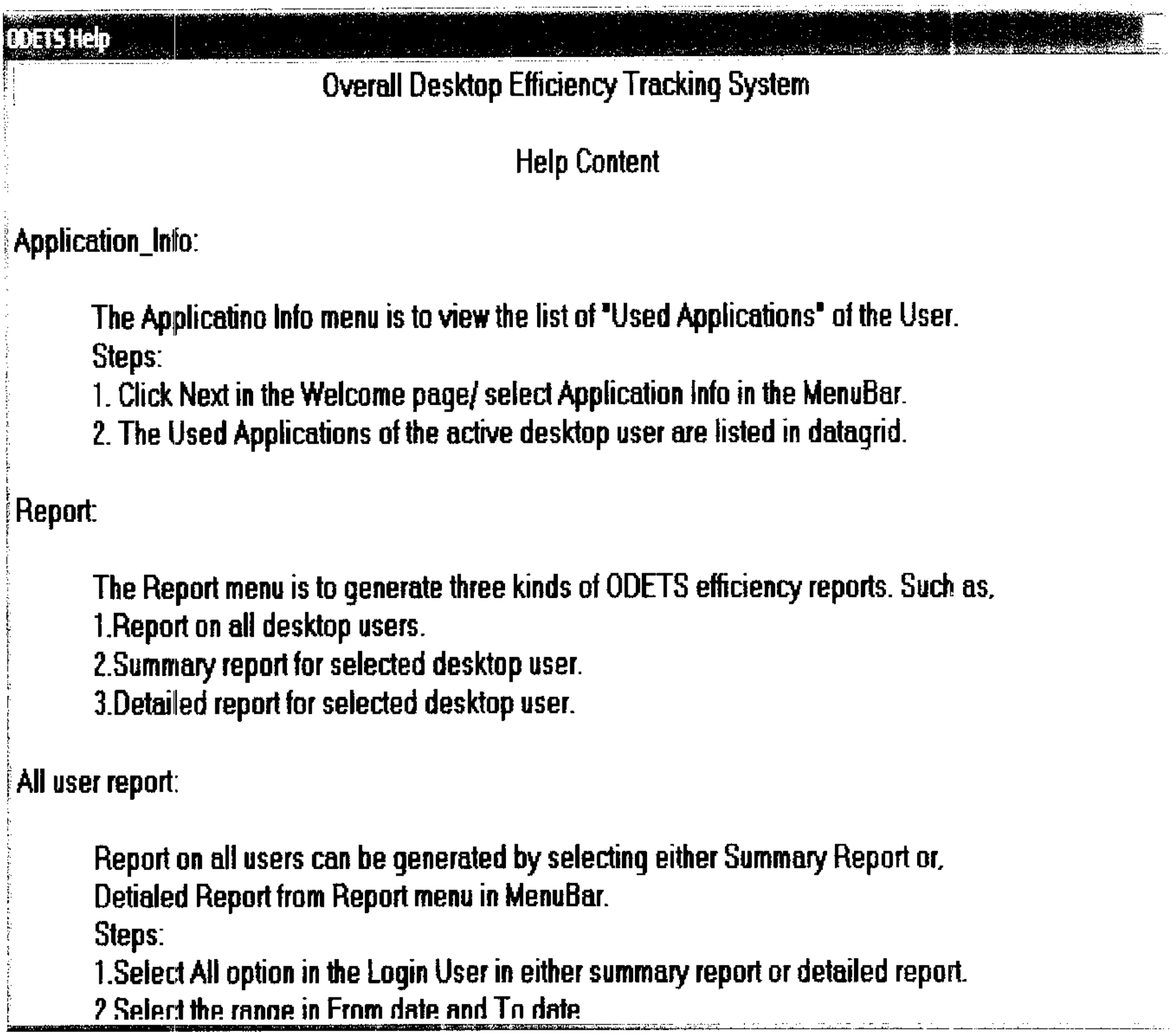
Login Date	Login Domain	Application
6/14/2009	KCT	notepad
6/14/2009	KCT	BTSTAC~1
6/14/2009	KCT	explorer

Current Page No.: 1

Total Page No.: 1

Zoom Factor: 100%

ODETS Help Screen



Overall Desktop Efficiency Tracking System

Help Content

Application_Info:

The Applicatino Info menu is to view the list of "Used Applications" of the User.
Steps:
1. Click Next in the Welcome page/ select Application Info in the MenuBar.
2. The Used Applications of the active desktop user are listed in datagrid.

Report:

The Report menu is to generate three kinds of ODETS efficiency reports. Such as,
1.Report on all desktop users.
2.Summary report for selected desktop user.
3.Detailed report for selected desktop user.

All user report:

Report on all users can be generated by selecting either Summary Report or,
Detialed Report from Report menu in MenuBar.
Steps:
1.Select All option in the Login User in either summary report or detailed report.
2.Select the range in From date and To date

CHAPTER 10

10. REFERENCES

BOOKS

1. **PROFESSIONAL C# 2005** by Christian Nagel, Bill Evjen Published by Wiley-India, 2005.

2. **Dissecting a C# Application: Inside SharpDevelop:** Christian Holm, Mike Kruger, and Bernhard Spuida Published by Wrox Press, 2003.

3. **Programming Microsoft SQL Server 2000** by Stephen Forte Published by Microsoft Press, 2004.

WEB SITES

1. www.csharpkey.com
2. www.dotnetcoding.com