

A PROJECT
ON
(Database GUI And Data Switching)

BY
(Amruta Pimple-820)
(Tanvi Raut-825)
(Bhagyashri Redekar-827)

Under the guidance of

Internal Guide
Mrs.Sumitra Sadhukhan

External Guide

MCT
MANJARA CHARITABLE TRUST
RAJIV GANDHI INSTITUTE OF TECHNOLOGY
Juhu-Versova Link Road Versova, Andheri (W), Mumbai-53

Department of Computer Engineering

University of Mumbai

Apr-2014

CERTIFICATE

Department of Computer Engineering

This is to certify that

- | | |
|-----------------------|-----|
| 1. AMRUTA PIMPLE | 820 |
| 2. TANVI RAUT | 825 |
| 3. BHAGYASHRI REDEKAR | 827 |

Have satisfactory completed this project entitled

DATABASE GUI AND DATA SWITCHING

Towards the partial fulfillment of the

BACHELOR OF ENGINEERING

IN

COMPUTER ENGINEERING

As laid by University of Mumbai.

GUIDE

Mrs. Sumitra Sadhukhan

H.O.D.

Prof. S. B.Wankhade

Principal

Dr. Udhav Bhosle

Internal Examiner

External Examiner

PROJECT REPORT APPROVAL FOR B. E.

This project report entitled **Database GUI and Database Switching** by **Amruta Pimple, Tanvi Raut and Bhagyashri Redekar** are approved for the degree of **Bachelor of Computer Engineering**.

External Examiners

1.-----

2.-----

Internal Examiner

1.-----

2.-----

HOD

Date:

Place:

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

AMRUTA PIMPLE (B-820)

TANVI RAUT (B-825)

BHAGYASHRI REDEKAR (B-827)

Date:

ACKNOWLEDGEMENT

We wish to express our sincere gratitude to Dr. U. V. Bhosle, Principal and Prof. S. B. Wankhade, H.O.D of Computer Department of RGIT for providing us an opportunity to do our Seminar work on “Database GUI and database switch ”.

This Seminar bears on imprint of many people. We sincerely thank our Seminar guide Mrs.Sumitra for his guidance and encouragement in successful completion of our Seminar work.

We would also like to thank our staff members for their help in carrying out this Seminar work.

Finally, we would like to thank our colleagues and friends who helped us in completing the Seminar successfully.

1. AMRUTA PIMPLE
2. TANVI RAUT
3. BHAGYASHRI REDEKAR

ABSTRACT:

Project: Database GUI and Database Switch

Front End: Microsoft VB.Net

Description:

This Software helps an user to do all database operations through a GUI provided with minimal knowledge of the database. Here user can do all database activities like Creating Table, Inserting Data, Joining two tables etc without knowledge of queries. Also a Training mode is provided where user can learn a query for a particular database activity while performing it. Also user can listen to the query through a speak mode.

No of Modules:

We will cover all the features of Oracle SQL in making the software. i.e.

1. Create Table
2. Alter Table
3. Drop Table
4. View Data (Basic Select Queries)
5. View Table Structure
6. Insert / Update / Delete Data
7. Create Views
8. Drop Views
9. Display Data through Views
10. Joins
11. Summary Queries
12. Sub Queries
13. Security
14. Create User
15. Grant Rights
16. Revoke Rights
17. Drop User

Other feature of the software is that you can switch the database Table, Data etc from One DBMS to other DBMS for eg:

- 1) Oracle
- 2) MySQL
- 3) SQL Server
- 4) Access

Without rewriting the queries. User just has to select the Source DBMS and Table from that DBMS and a switch file is created on a click of a button. Now this switch file can be used in any machine (can be a remote machine) to switch data in any of the DBMS mentioned above.

Table Of Contents

Chapter No	Chapter Name	Page No
1.	Introduction And Motivation	1
	1.1Database GUI	1
	1.2Database Switching	2
	1.3Disadvantages of using query languages	3
2.	Problem Statement	6
3.	Requirement Analysis	8
	3.1Feasibility Study	8
	3.2Application Prototype And GUI Design	10
	3.3Custom Database Design	10
	3.4Custom Application Design	11
	3.5Custom Programming	11
	3.6Testing And Bug Fixing	12
	3.7Deployment	12
	3.8Minimum Hardware Requirement	13
	3.9Minimum Software Requirement	14
4.	Project Design	16
	4.1UML Diagrams	16
	4.2Data Flow Diagrams	21
5.	Implementation Details	24
	5.1Code Of Project	25
	5.2GUI Design	30
6.	Technologies Used	41
	6.1Visual Studio	41

	6.2Windows Form Designer	41
	6.3Microsoft VB.NET 2005	42
	6.4Oracle	44
	6.5My SQL	45
	6.6SQL Server	45
	6.7Microsoft Access	46
	6.8Queries	46
	6.9Tables	47
7.	Test Cases	48
	7.1Login Form	50
	7.2 Main Form	51
	7.3Test Plan For Buttons That Are Common In All Forms	52
	7.4Test Plan For Text Boxes In All Forms	54
	7.5Test Plan For Make Form,Switch To,Speech Mode	55
8.	Project Time Line	56
	8.1Gantt Chart	56
	8.2Timeline Chart	57
9.	Task Distribution	58
	9.1Implementation Plan	58
10.	Conclusion And Future Work	60
11.	References	62
12.	Appendix	64

LIST OF FIGURES

Figure No	Figure Name	Page No
1.2.1	Architecture For Database GUI And Database Switching	02
4.1.1	Use Case Diagram	16
4.1.2	Activity Diagram	17
4.1.3	Sequence Diagram	18
4.1.4	Collaboration Diagram	19
4.1.5	Use Case Diagram for Switching	20
4.2.1	Data Flow Diagram for GUI	21
4.2.2	Data Flow Diagram For Data Switching	23
5.2.1	User Login	30
5.2.2	Existing Tables	31
5.2.3	Alter Table	32
5.2.4	Drop Table	33
5.2.5	Create View	34
5.2.6	Drop View	35
5.2.7	Create User	36
5.2.8	Grant System	37
5.2.9	Revoke System	38
5.2.10	Saving Switched File	39
5.2.11	Switched File Access	40
8.1.1	Gantt Chart	56
8.2.1	Time Line Chart	57

LIST OF TABLES

Table No	Table Name	Page No
3.8.1	Minimum Hardware Requirement	13
3.8.2	Minimum Software Requirement	14
7.1	Login Form	50
7.2	Main Form	51
7.3	Test Plan For Buttons That Are Common In All Forms	52
7.4	Test Plan For Text Boxes In All Forms	54
7.5	Test Plan For Make Form,Switch To,Speech Mode	55
9.1.1	Phase 1 of Project Plan	58
9.1.2	Phase 2 of Project Plan	59

Chapter 1

1. Introduction and Motivation

- Querying through GUI to execute complex as well as simple queries has proved to be an efficient system to people who do not have much knowledge about databases.
- Switching between the databases reduces the work of rewriting the queries for each of the database thus saving time.
- The Graphical User Interface generates the queries using Oracle and My SQL as the database. Similarly other databases can also be used in order to execute the queries using databases such as MySQL, Sqlserver, MS access, etc with the GUI
- This use of databases to generate the queries can be successfully implemented so that the work of writing full queries can be reduced and a non-technical user can also use it without having any knowledge of databases.
- The system at hand can be divided into two inter-related subsystems:

1.1. Database GUI

- This part of the system deals with generating and executing queries and listening to the queries through speech mode using SAPI software.
- There are various processes that are handled by this system are:
 1. User can create a table by clicking on the drop down menus.
 2. User can add data to the tables.
 3. User can read the query to be generated.
 4. User can listen to the query using speech mode.
 5. User can search for a particular entry by specifying a condition.

1.2. Database Switching

- User can select among the four databases such as oracle, MySQL, MS access, Sqlserver from which the file is to be switched and can switch the file to be executed in any of the four databases.
- For e.g. user can select the file to be switched from Oracle and switch this file required by other databases such as MySQL, Sqlserver and MS access.

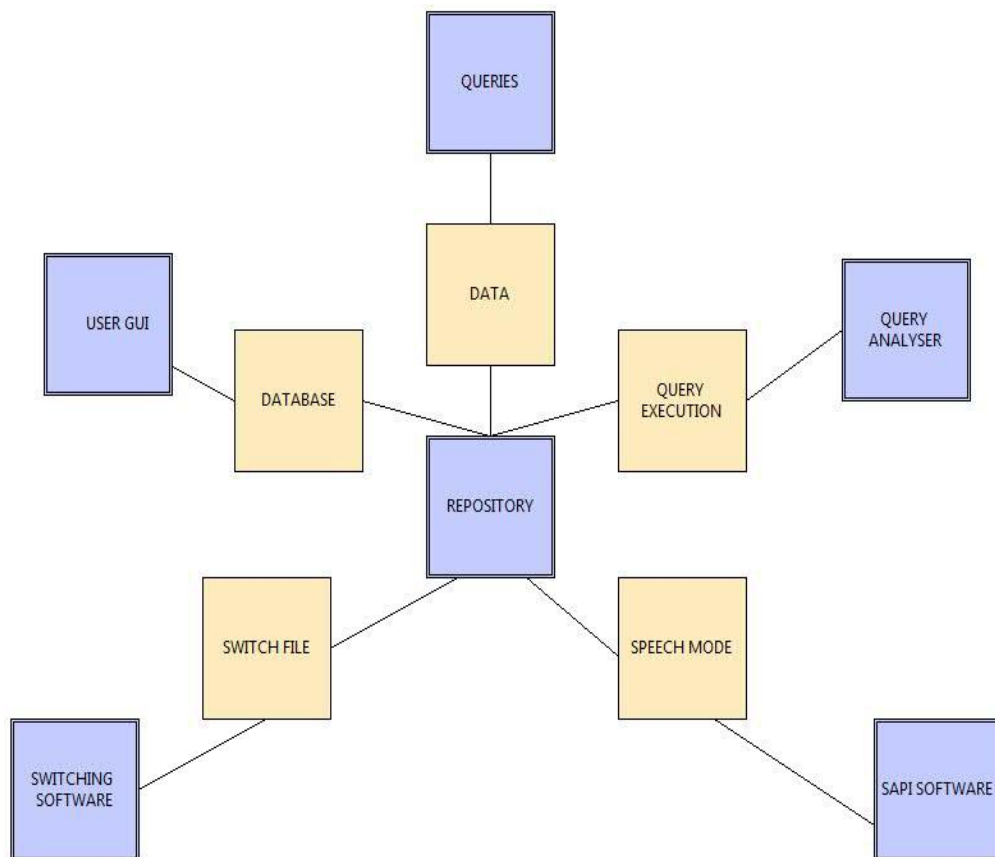


Fig 1.2.1 Architecture for Database GUI and Switching Over Distributed Networks

- The main motivation of our work comes from experiences in using database query languages. Even people with a computer science background (ourselves included) often have difficulty using the so called “high level user friendly languages”.
- Nonexpert users may not have the patience, ability, or desire to learn and use these languages correctly. By non-expert users (as opposed to casual users), we mean non-computer science professionals such as social analysts, statisticians or accountants who have to deal with data regularly.
- The problem becomes much worse in an environment with very large databases that have very large and complex database definitions (schemas). Large statistical databases such as the Census database and energy database are examples of such an environment. We believe that the following factors are the major reasons for the difficulty in using and understanding query languages.

1.3. Disadvantages of using query languages

1. The user has to remember too many things. – The names of the record types and attributes have to be remembered before the user can express a query.

Lower level details such as the format and units of the attributes are only found by exploring the data, or looking the attribute definitions up in a dictionary (sometimes as parts of manuals). On the semantics side, the user has to determine the “meaning” of acronyms used to represent elements (record types and their attributes). All these problems are magnified when the database has a very complex schema (hundreds of record types, thousands of attributes).

2. Semantically poor data models. -- Most high level query languages are based on mathematical concepts such as Predicate Calculus or Algebra or set theory. This sometimes leads to languages with a solid foundation; often, however, users don't relate to mathematical concepts such as range variables,

join clauses, projections or complex aggregate functions. Often the problem comes from inadequate semantics in the underlying data model. An example is the join clause in the relational data model. Join clauses are typically used for establishing relationships that exist implicitly between relations. The burden is put on the user to augment the semantics of the implicit relationships between relations and make them explicit. It is the responsibility of the user to restate these relationships every time the relations are referred to. We believe a more explicit model should be used to support the non-expert user interface to complex data.

3. No feedback during the query process. – The chances of formulating a complex query correctly on the first try (or even the first few trials) ‘are slim. When using current languages, there is the fear and doubt as to whether the query is complete or whether some conditions are missing. Users could benefit from a facility that allows them to build a query in a piecemeal fashion with feedback of partial results available to them at any time. Also, users of large statistical databases (statistical analysts) often do not have a set of fixed queries in mind. Rather their interaction with the databases is exploratory in nature and highly dependent on the intermediate results. Again a facility with feedback of partial results should be helpful in this kind of environment. To simulate this facility with current query systems requires stating separate queries. Query systems should be designed to encourage the experimental, exploratory nature of formulating queries in a piecemeal fashion.

In contrast to a traditional setting where users express queries against the database schema, we assert that the semantics of data can often be understood by viewing the data in the context of the user interface (UI) of the software tool used to enter the data. That is, the users will understand the data in a database by seeing the labels, drop-down menus, tool tips, or other help text that are built into the user interface. Our goal is to allow domain experts with little technical skill to understand and query data. In this, we present our GUI framework and describe how we use forms-based UIs to generate a conceptual

model that represents the information in the user interface. We then describe how we generate a query interface from the conceptual model. We characterize the resulting query language using a subset of the relational algebra. Since most application developers want to craft a physical database to meet desired performance needs, we present here a transformation channel that can be configured by instantiating one or more of our transformation operators. The channel, once configured, automatically transforms queries from our query interface into queries that address the underlying physical database and delivers query results that conform to our query interface. In this, we define and formalize our database transformation operators. The motivations are that first, we demonstrate the feasibility of creating a query interface based directly on the user interface, second, we introduce a general purpose database transformation channel that will likely shorten the application development process and increase the quality of the software by automatically generating software artifacts that are often made manually and are prone to errors, third we switch the queries among different databases without rewriting them.

Chapter 2

2. Problem Statement

- Database GUI helps a user to do all database operations through a GUI provided with minimal knowledge of the database. Here user can do all database activities like Creating Table, Inserting Data, Joining two tables etc without knowledge of queries. Also a Training mode is provided where user can learn a query for a particular database activity while performing it. Also user can listen to the query through a speak mode.
- Database Switching includes execution of queries on different machines connected in LAN with different backend without rewriting the queries. User just has to select the Source DBMS and Table from that DBMS and a switch file is created on a click of a button. Now this switch file can be used in any machine (can be a remote machine) to switch data in any of the DBMS.
- Following features of Oracle SQL will be covered in making the software i.e.
 1. Create Table
 2. Alter Table
 3. Drop Table
 4. View Data (Basic Select Queries)
 5. View Table Structure
 6. Insert / Update / Delete Data
 7. Create Views

8. Drop Views

9. Display Data through Views

10. Summary Queries

11. Security

12. Create User

13. Grant Rights

14. Revoke Rights

15. Drop User

- Along with it, user can switch the database Table, Data etc from One DBMS to other DBMS without rewriting the queries. For e.g.:

1) Oracle

2) MySQL

3) SQL Server

4) Access

- User just has to select the Source DBMS and Table from that DBMS and a switch file is created on a click of a button. Now this switch file can be used in any machine (can be a remote machine) to switch data in any of the DBMS mentioned above.

3. Requirement Analysis

3.1 Feasibility Study

- The feasibility study plays a major role in the analysis of the system. The very decision of the system analyst, to design a particular system depends whether the system is feasible or not. Hence, the feasibility study forms the very basic of the system.
- The feasibility study can be categorized into:

3.1.1 Technical Feasibility

- It has been determined that the technology needed for the proposed system is available and that this technology can be integrated into the application.
- Technical evaluation has also evaluated the existing system to find that it cannot be upgraded in keeping with the users needs. Hence, we need to create an entire new system which caters to the specific needs of the user. The end user can be equipped with the pre-mentioned hardware and software requirements.

3.1.2 Operational Feasibility

- There are two aspects of operational feasibility for the system. One aspect is that of technical performance and other is of acceptance.
- It has been determined that the system can provide correct and timely data required for the end user needs. Also it has been determined that the system will be accepted by both users with and without technical knowledge.

3.1.3 Economic Feasibility

- The economic feasibility of the system is mainly concerned with its financial aspects. It determines whether the project is economically feasible. As the hardware and software are already available easily in the market, no further investment is to be made in that direction, the only cost involved is that of implementing the system.
- The front end system is built in Visual Studio 2005 which is freely available. The backend is built in connecting different databases such as Oracle, SQLserver, MS access, MySql server which can be made available with easy efforts and lower cost.
- Also since the application works on middle-level computers it is not necessary to spend a lot of money in obtaining a computer. It was decided that the project was technically feasible because of the following:
 - Necessary technology exists to do what is suggested.
 - The system would be expandable if so decided.
 - The system can be integrated with other networking tools and output of the system can be provided to other tools for high end analysis.
- The purpose of Requirement Analysis is to obtain a thorough and detailed understanding of the project need as defined in Project Origination and captured in the Business Case, and to break it down into discrete requirements, which are then clearly defined, reviewed and agreed upon with the Customer Decision-Makers.
- During Requirements Analysis, the framework for the application is developed, providing the foundation for all future design and development efforts.

3.2 Application Prototype & GUI Design

- Prototyping of the software application along with GUI design helps all stakeholders to get a feel of a working system at the initial stages of the project. A prototype is very useful to crystallize and further refine requirements. This phase helps reduce project risks and introduce certainty into the schedule and cost.
- Prototype allows users to visualize an application that hasn't yet been constructed. This will help developer to get an idea of what the system will look like, and make it easier for users to make design decisions without waiting for the system to be built. Major improvements in communication between users and developers were often seen with the introduction of prototypes. Early views of applications led to fewer changes later and hence reduced overall costs considerably.
- In our project user must be aware of all types of SQL queries. Accordingly user will select the specific operation. User has to fill all the fields and can build the query. Changes will be saved in database after building query.

3.3 Custom Database Design

- Once the requirements and GUI design are signed off by the client, the stage is set for detailed database and application design. Effort goes into ensuring that the database design is logical, not overly complex, and flexible to handle future requirements. We have the skills and the experience to develop scalable database designs optimized for large Internet audiences and high performance.
- Database design completely depends upon the user's operations. Once the user is ready with full flow of data, developer can start with

database design. Developer can decide which fields are required throughout the project.

3.4 Custom Application Design

- Technical architects break up the application into the logical tiers (Presentation, Logic, and Data Access) and specify the class hierarchy for an object oriented design. Typically, this doesn't need to be done from scratch.
- The technologies we use for our projects provide the base framework for this purpose e.g. the Send Framework for PHP based projects, .NET development framework for .NET based applications, and frameworks like Struts for J2EE based development. Developer must select proper front end and back end technology so that software will be suitable to user's operating system.

3.5 Custom Programming

- Backed with a rigorous requirements analysis and technical design effort, the stage is all set for coding to begin. Attention to coding conventions and frequent peer reviews throughout development ensures a high quality code that is well structured, documented, and flexible.
- Once the GUI and database fields of project are clear developer can start with coding. Coding must be simple and flexible. Proper security is provided for coding according to project's requirements

3.6 Testing & Bug-Fixing

- Skilled quality assurance personnel thoroughly and systematically test the application and prepare defect reports. The process of defect-fixing and testing proceeds iteratively until the software is considered stable to be deployed.
- If any errors are present in project, it reduces its efficiency as well as productivity. So errors must be analysed properly and bug-fixing has to be done.

3.7 Deployment

- The software is deployed and put in place with application monitoring and periodic data backups. Typically, we prepare a deployment manual that documents the steps to deploy and rollback releases of the application. This manual will help user if he will get any doubt while using software

3.8 Minimum Hardware Requirement

Requirement	Professional	Standard	Visual Studio Tools for Office
Processor	600 MHz processor Recommended: 1 gigahertz (GHz) processor ¹	Same	Same
RAM	192 MB Recommended: 256 MB ¹	Same	256 MB
Available Hard Disk Space	Without MSDN: <ul style="list-style-type: none"> • 1 GB of available space required on system drive² • 2 GB of available space required on installation drive With MSDN: <ul style="list-style-type: none"> • 1 GB of available space required on system drive • 3.8 GB of available space required on installation drive with a full installation of MSDN • 2.8 GB of available space required on installation drive with a default installation of MSDN 	Same	Same
Operating System	Windows 2000 Service Pack 4, Windows XP Service Pack 2, Windows Server 2003 Service Pack 1, or Windows Vista ^{3,4} For a 64-bit computer, the requirements are as follows: <ul style="list-style-type: none"> • Windows Server 2003 Service Pack 1 x64 editions • Windows XP Professional x64 Edition 	Same	Same
CD-ROM Drive or DVD-ROM Drive	Required ⁵	Required	Required
Video	800 X 600, 256 colors Recommended: 1024 X 768, High Color 16-bit	Same	Same
Mouse	Microsoft mouse or compatible pointing device	Same	Same

Table 3.8.1 Minimum hardware requirements

3.9 Minimum Software Requirement

Application or component	Information
Microsoft Windows Server 2003 (Standard Edition or Enterprise Edition)	For information about installing Microsoft Windows Server 2003 for use with BizTalk Server 2006, see the "BizTalk Server Installation Guide" at http://go.microsoft.com/fwlink/?linkid=46922 . For general information about installing Windows Server 2003, see the Windows Server 2003 Web site at http://go.microsoft.com/fwlink/?LinkId=47391 .
Microsoft Windows Server 2003 Service Pack 1	You can download from http://go.microsoft.com/fwlink/?LinkId=46207 .
Microsoft Internet Information Services 6.0 (if using BAS and Message Repair and New Submission)	For information about installing Internet Information Services 6.0, see the "BizTalk Server Installation Guide" at http://go.microsoft.com/fwlink/?linkid=46922 .
Microsoft Office Excel 2003, and InfoPath (installed on client computers when using BAM and Message Repair and New Submission)	For more information about Microsoft Office Professional Edition, see the Microsoft Office Online Web site at http://go.microsoft.com/fwlink/?LinkId=31630 .
Microsoft Office 2003 Service Pack 2 (installed on client computers when using BAM and Message Repair and New Submission)	You can download Office SP2 from http://go.microsoft.com/fwlink/?LinkId=55034 .
Microsoft Visual Studio 2005 Enterprise or Standard Edition (if in a development environment)	For specific information about installing Microsoft Visual Studio 2005 for use with BizTalk Server 2006, see "Installing Visual Studio 2005" in BizTalk Server 2006 Help. For general information about installing Visual Studio 2005, see the Visual Studio 2005 Web site at http://go.microsoft.com/fwlink/?LinkId=31628 .

Database GUI & Switching over Distributed Networks

Microsoft SQL Server 2000 or Microsoft SQL Server 2005	<p>For specific information about installing SQL Server 2000 or btsSQLServer2005 for use with BizTalk Server 2006, see the "BizTalk Server Installation Guide" at http://go.microsoft.com/fwlink/?linkid=46922.</p> <p>For general information about installing SQL Server 2000 or SQL Server 2005, see the SQL Server Web site at http://go.microsoft.com/fwlink/?LinkId=28031.</p> <p>Note SQL Server 2000 Personal Edition is not compatible with BizTalk Server 2006.</p>
SQL Server Analysis Services (if installing Microsoft btsSQLServer2000 and using BAM aggregations)	You can download SQL Server Analysis Services from http://go.microsoft.com/fwlink?LinkID=22838 .
SQL Server 2000 Service Pack 4 (if installing Microsoft SQL Server 2000)	You can download SQL Server 2000 Service Pack 4 from http://go.microsoft.com/fwlink?LinkID=48932 . The download file for x86 is SQL2000-KB884525-SP4-x86-ENU.EXE.
SQL Server 2000 Analysis Services Service Pack 4 (if installing Microsoft SQL Server 2000)	<p>You can download SQL Server 2000 Analysis Services Service Pack 4 from http://go.microsoft.com/fwlink?LinkID=48932. The download file is SQL2000.AS-KB884525-SP4-x86-ENU.EXE.</p> <p>Note</p> <p>This is a separate service pack download from the SQL Server 2000 Service Pack 4 download.</p>
SQLXML 3.0 with Service Pack 3 (for either Microsoft btsSQLServer2000 or Microsoft SQL Server 2005)	You can download SQLXML 3.0 with Service Pack 3 from http://go.microsoft.com/fwlink?LinkID=47396
SQL Notification Service with Service Pack 1 (if installing Microsoft SQL Server 2000)	You can download SQL Notification Service with Service Pack 1 from http://go.microsoft.com/fwlink?LinkID=47397
Microsoft Windows SharePoint Services 2.0 with Service Pack 2 (if using BAS and Message Repair and New Submission)	<p>For specific information about installing Windows SharePoint Services 2.0 with Service Pack 2 for use with BizTalk Server 2006, see the "BizTalk Server Installation Guide" at http://go.microsoft.com/fwlink/?linkid=46922.</p> <p>You can download Microsoft Windows SharePoint Services 2.0 at http://go.microsoft.com/fwlink/?LinkId=31631. You can download Windows SharePoint Services 2.0 with Service Pack 2 from http://go.microsoft.com/fwlink/?LinkId=47398.</p>
Microsoft BizTalk Server 2006 Enterprise or Standard Edition	<p>See the "BizTalk Server Installation Guide" at http://go.microsoft.com/fwlink/?linkid=46922.</p> <p>For more information about Microsoft BizTalk Server, see the Microsoft BizTalk Server Web site at http://go.microsoft.com/fwlink/?LinkId=12162.</p>

Table 3.8.2 Minimum Software requirements

4. Project Design

4.1 UML Diagrams

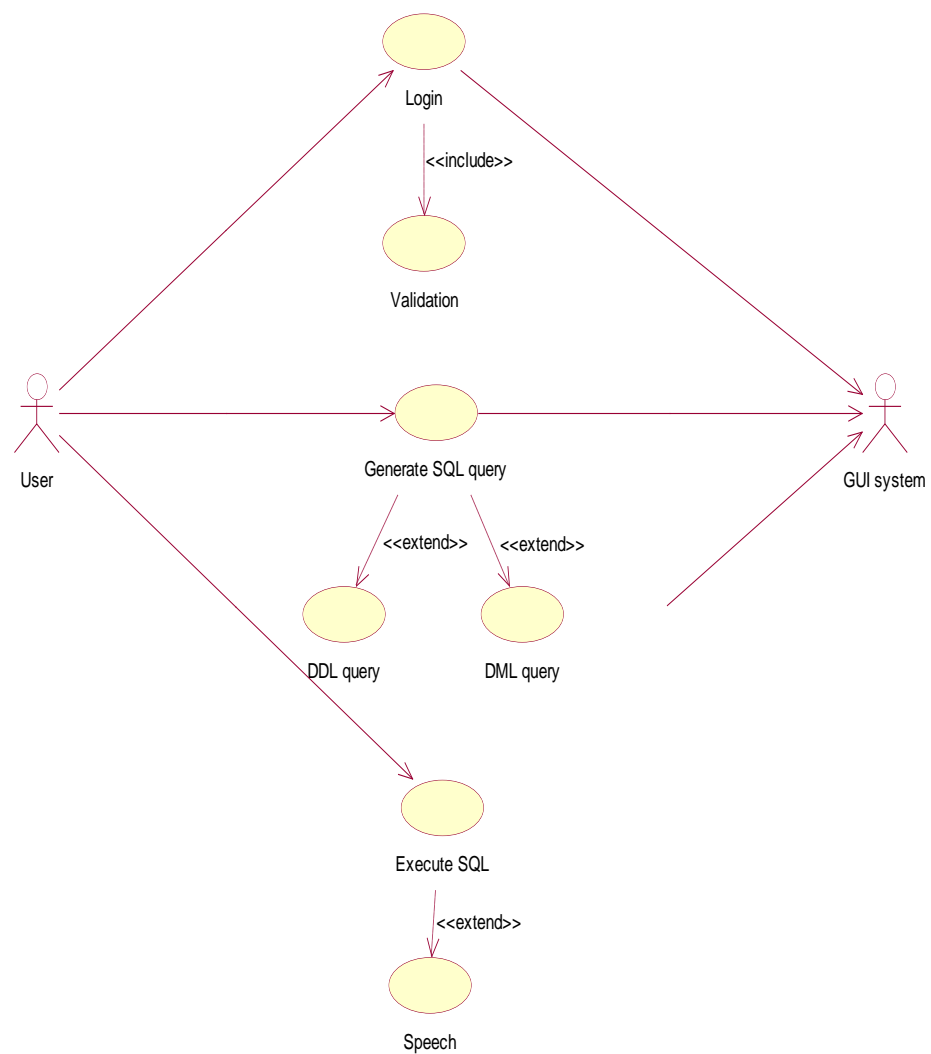


Fig: 4.1.1 Use Case Diagram

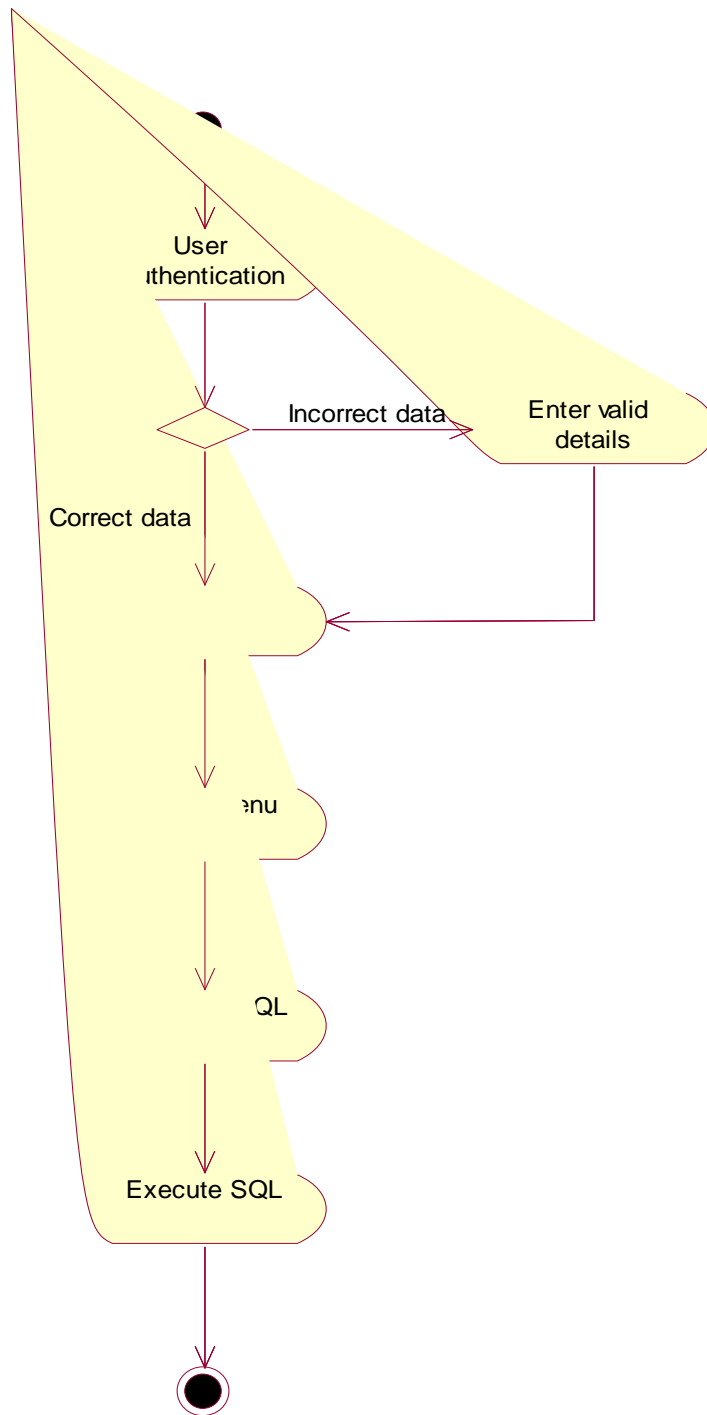


Fig 4.1.2 Activity Diagram

Database GUI & Switching over Distributed Networks

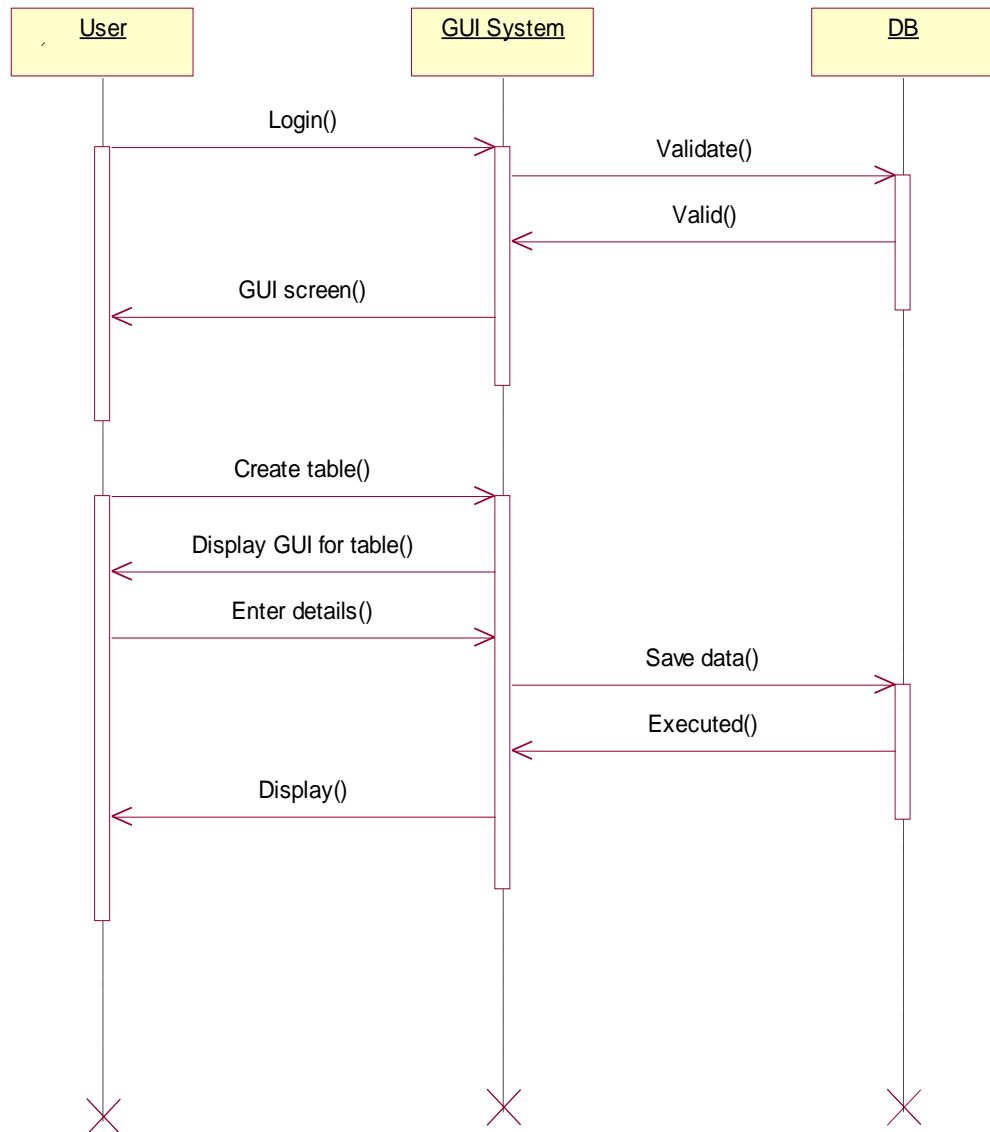


Fig 4.1.3 Sequence Diagram

Database GUI & Switching over Distributed Networks

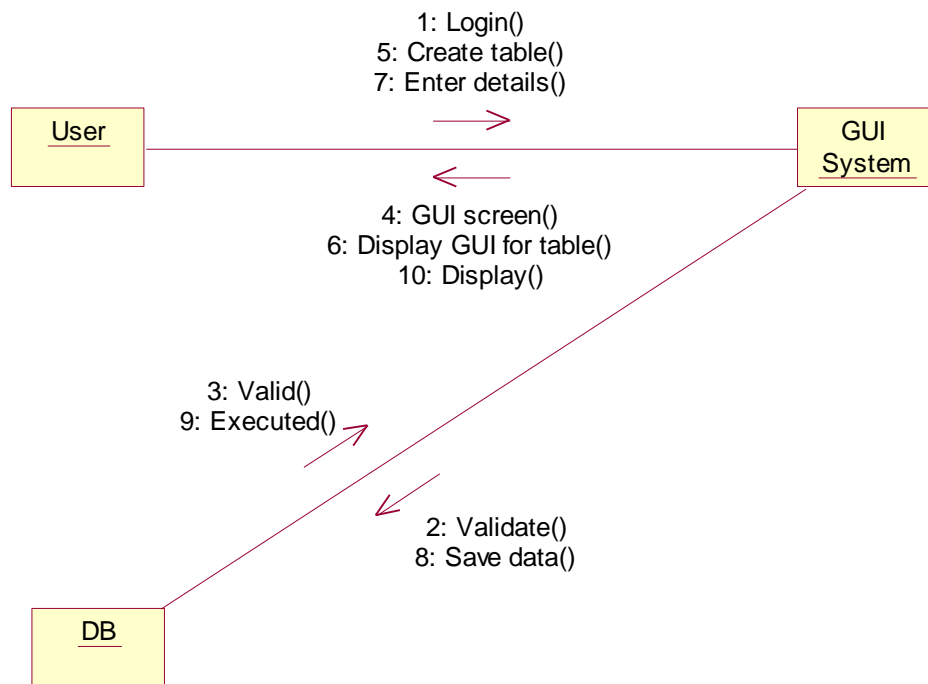


Fig 4.1.4 Collaboration Diagram

Database GUI & Switching over Distributed Networks

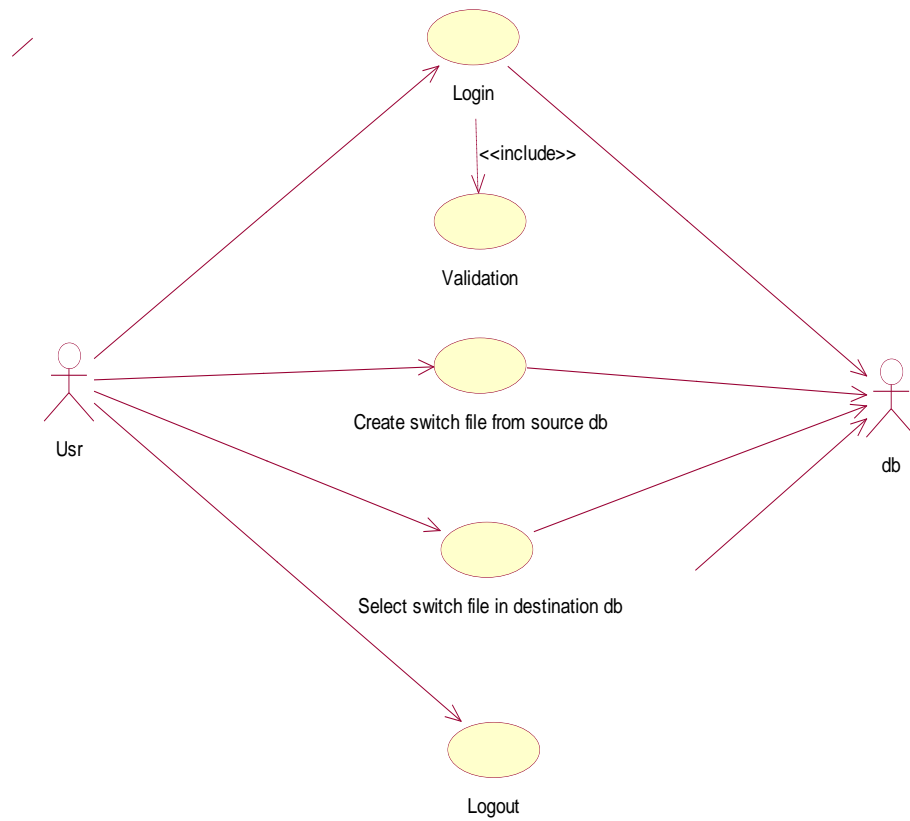
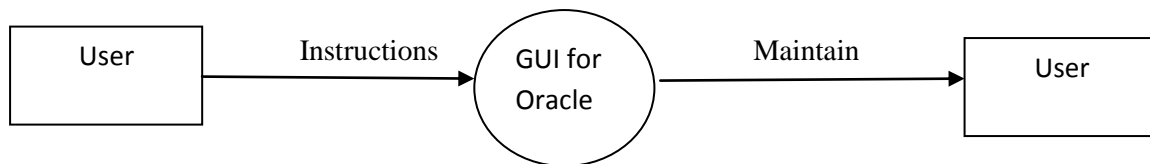


Fig 4.1.5 Use case diagram for switching

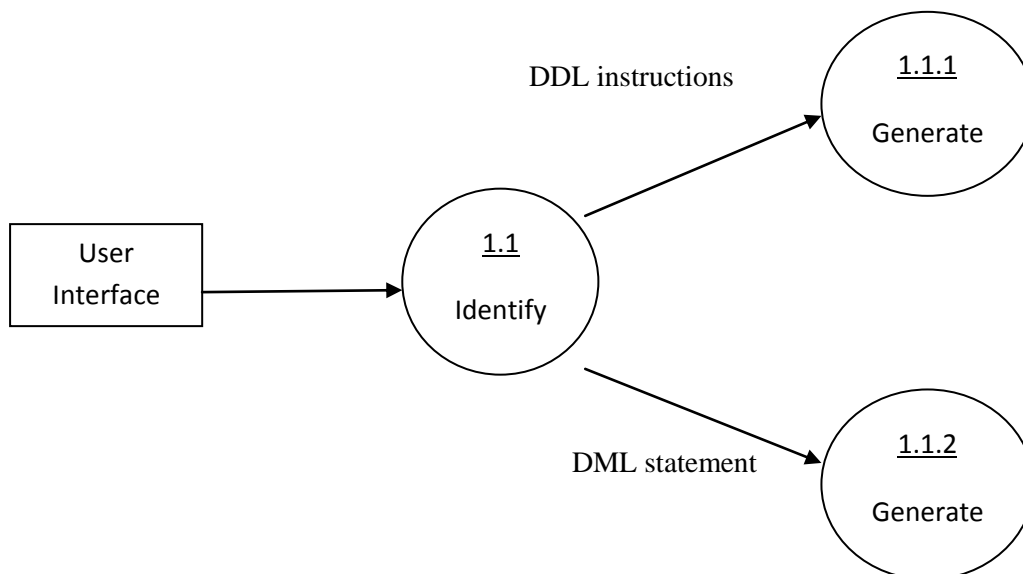
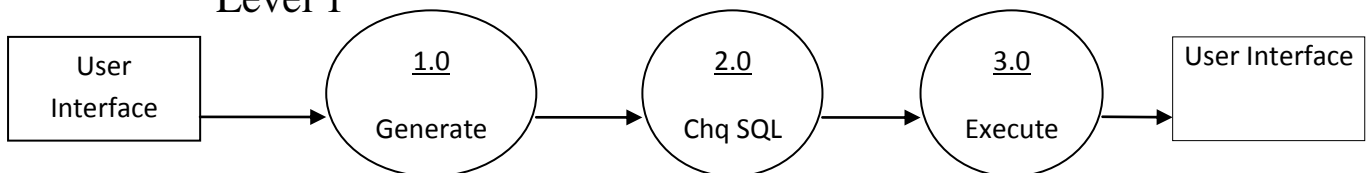
4.2 Data Flow Diagrams

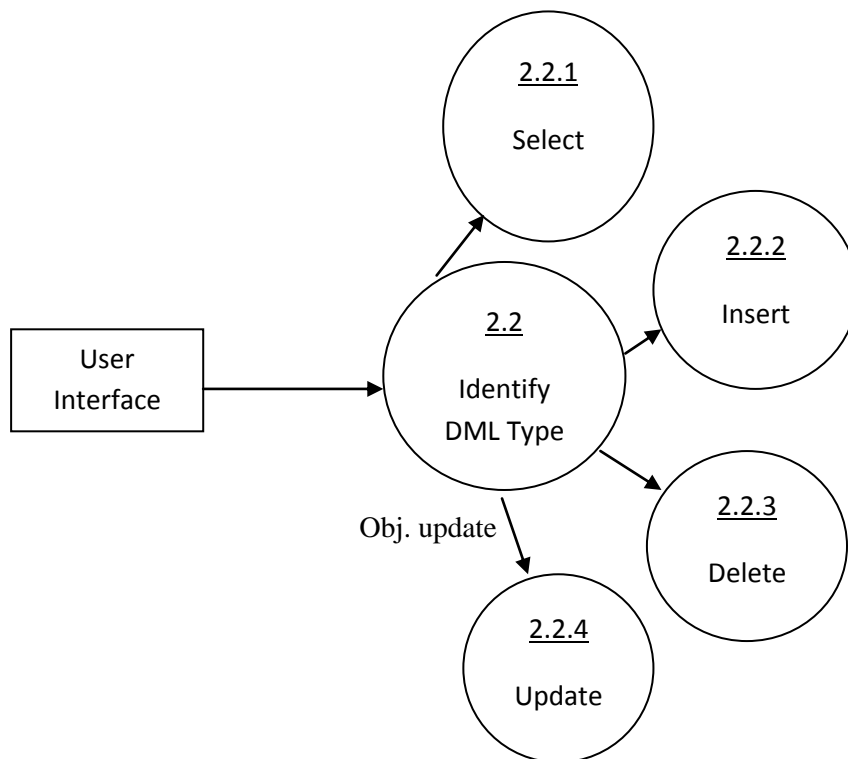
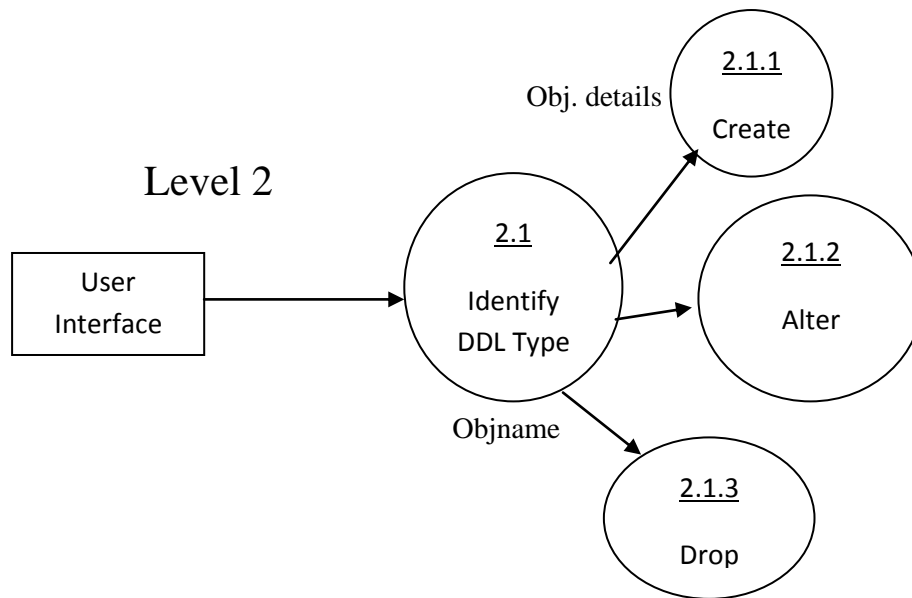
4.2.1 Database GUI

Level 0



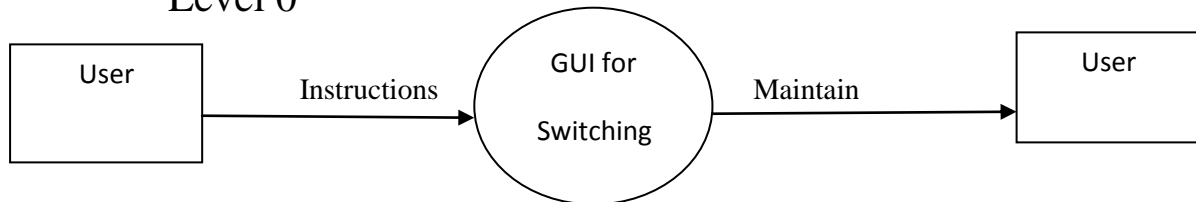
Level 1



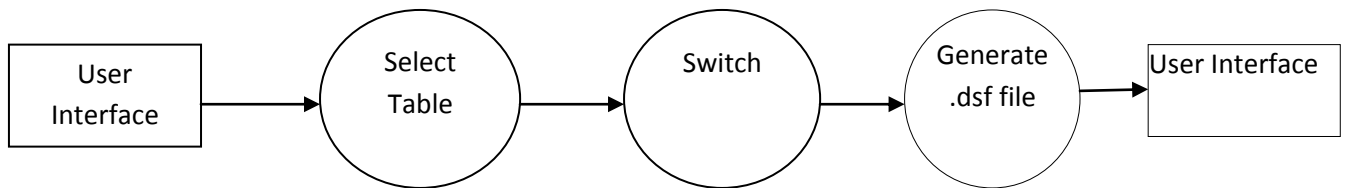


4.2.2 Database Switching

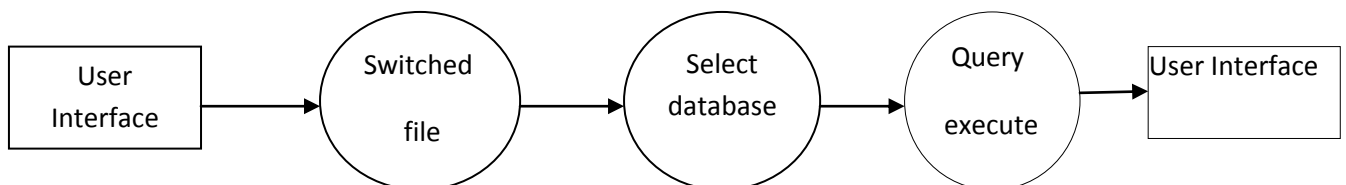
Level 0



Level 1



Level2



Chapter 5

5. Implementation Details

- Database GUI and Switching is an important software application which involves executing the queries on the click of a button.
- First the user uses Visual Basic to execute the queries using Oracle. For this he first has to log in, in order to view or create the tables.
- He can then perform operations such as create table, update table, drop table, delete, insert values, etc.
- The user can also create views for the existing tables.
- A new user can also be created to which system rights can be given. Thus the user can be granted rights and the rights can also be revoked.
- The user can click on the “read” button to listen to the query using speech mode.
- Database Switching involves switching the queries between different databases without rewriting them. Here the user can use the “Make” button which has three databases-Oracle, Mysql, SQLserver. These databases consist of queries that can be switched into other databases like Oracle, Mysql, SQLserver, MS access.
- A “switch” button is provided in order to perform switching. A switch File (.dsf) is made and switched into any of the above databases.

5.1 Code of project

5.1.1 Database GUI

1. Connect

```
Public Class frmConnect
```

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As  
System.EventArgs) Handles Button1.Click
```

```
If TextBox1.Text = "" Then
```

```
MsgBox("Please enter server name", MsgBoxStyle.Critical, "Error")
```

```
Exit Sub
```

```
End If
```

```
Module1.ServerName = TextBox1.Text
```

```
Module1.Username = TextBox2.Text
```

```
Module1.Password = TextBox3.Text
```

```
Me.Close()
```

```
Form1.Connect_Database()
```

```
End Sub
```

```
Private Sub RadioButton2_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs)
```

```
GroupBox1.Enabled = True
```

```
End Sub
```

```
Private Sub RadioButton1_CheckedChanged(ByVal sender As  
System.Object, ByVal e As System.EventArgs)
```

```
GroupBox1.Enabled = False
```

```
End Sub
```

```
End Class
```

6.1.2 Database Switching

```
Imports System.Data.Odbc
```

```
Imports System.Text.RegularExpressions
```

```
Public Class frmSwitchMySQL
```

```
Dim cn As New OdbcConnection
```

```
Dim cmd As New OdbcCommand
```

```
Private Sub frmSwitchMySQL_Load(ByVal sender As System.Object,  
ByVal e As System.EventArgs) Handles MyBase.Load
```

End Sub

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click

Try

If (TextBox1.Text <> "" And TextBox2.Text <> "" And TextBox4.Text <> "") Then

cn.ConnectionString = "DRIVER={MySQL ODBC 3.51 Driver};SERVER=localhost;DATABASE=" + TextBox1.Text + ";UID=" + TextBox2.Text + ";PASSWORD=" + TextBox4.Text + ""

cn.Open()

cn.Close()

GroupBox1.Enabled = True

Else

MsgBox("Please enter Complete details", MsgBoxStyle.Critical, "Error")

End If

Catch ex As Exception

MsgBox(ex.Message)

MsgBox("Error in Connecting to Database", MsgBoxStyle.Critical, "Error")

End Try

End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Try

OpenFileDialog1.ShowDialog()

RichTextBox1.Text = ""

If (OpenFileDialog1.FileName <> "") Then

TextBox3.Text = OpenFileDialog1.FileName

RichTextBox1.LoadFile(TextBox3.Text, RichTextBoxStreamType.PlainText)

Dim s As String

s = ""

Dim s1 As String

Dim dind As Integer

dind = 0

Dim tot As String

Dim rema As String

rema = ""

tot = ""

s1 = ""

tot = RichTextBox1.Text

dind = tot.IndexOf("Data:")

```
If (dind > 0) Then
    s = tot.Substring(0, dind)
    rema = tot.Substring(dind + 5)
    s = s.Trim
    rema = rema.Trim
Else
    s = tot
End If
ListBox1.Items.Clear()
Dim reg As New Regex(",")
For Each s1 In reg.Split(s)
    s1 = s1.Trim()
    If (s1 <> "") Then
        ListBox1.Items.Add(s1)
    End If
Next
If (ListBox1.Items.Count > 0) Then
    Dim i As Integer
    Dim reg2 As New Regex("-")
    Dim str As String
    Dim tabname As String
    ListBox1.SelectedIndex = 0
    str = ListBox1.SelectedItem.ToString()
    s1 = ""
    tabname = ""
    For Each s1 In reg2.Split(str)
        tabname = s1
    Next
    Dim fn As String
    Dim dt As String
    fn = ""
    dt = ""
    Dim fg As Integer
    fg = 0
    Dim qry As String
    qry = "create table " + tabname + "("
    For i = 1 To ListBox1.Items.Count - 1 Step 1
        fn = ""
        dt = ""
        str = ""
        fg = 0
        ListBox1.SelectedIndex = i
        str = ListBox1.SelectedItem.ToString()
        For Each s1 In reg2.Split(str)
            If (fg = 0) Then
```

```

        fn = s1
        fg = 1
    Else
        dt = s1
        fg = 0
    End If
Next
dt = dt.ToUpper
If (dt = "VARCHAR2") Or (dt = "VARCHAR") Then
    dt = "VARCHAR(50)"
ElseIf (dt = "NUMBER") Or (dt = "NUMERIC") Or (dt =
"DECIMAL") Then
    dt = "NUMERIC"
End If
qry = qry + fn + " " + dt + ","
Next
qry = qry.Substring(0, qry.Length - 1)
qry = qry + ")"
MsgBox(qry)
cn.ConnectionString = "DRIVER={MySQL ODBC 3.51
Driver};SERVER=localhost;DATABASE=" + TextBox1.Text + ";UID=" +
TextBox2.Text + ";PASSWORD=" + TextBox4.Text + ""
cn.Open()
cmd.Connection = cn
cmd.CommandText = qry
cmd.ExecuteNonQuery()
cmd.Dispose()
cn.Close()
If (dind > 0) Then
    ListBox1.Items.Clear()
    Dim dr As New Regex("-")
    For Each s1 In dr.Split(rema)
        s1 = Trim(s1)
        If (s1 <> "") Then
            ListBox1.Items.Add(s1)
        End If
    Next
    Dim tl As Integer
    tl = ListBox1.Items.Count
    Dim lp As Integer
    lp = 0

    For lp = 0 To tl - 1 Step 1
        ListBox1.SelectedIndex = lp
        qry = ListBox1.SelectedItem.ToString
    
```

```
        qry = qry.Substring(0, qry.Length - 1)
        cn.Open()
        cmd.Connection = cn
        cmd.CommandText = "insert into " + tabname + " values(" +
qry + ")"
        cmd.ExecuteNonQuery()
        cmd.Dispose()
        cn.Close()
    Next
End If
    MsgBox("Table Switched Successfully",
MsgBoxStyle.Information, "Done")
Else
    MsgBox("Cannot Switch as File is Corrupted",
MsgBoxStyle.Critical, "Error")
End If
End If
Catch ex As Exception
    MsgBox("Error Raised : " + ex.Message.ToString(),
MsgBoxStyle.Critical, "Error")
End Try
End Sub
End Class
```

5.2 GUI Design

1. To start using the SQL commands, first connect to oracle with a correct server name, username and password.

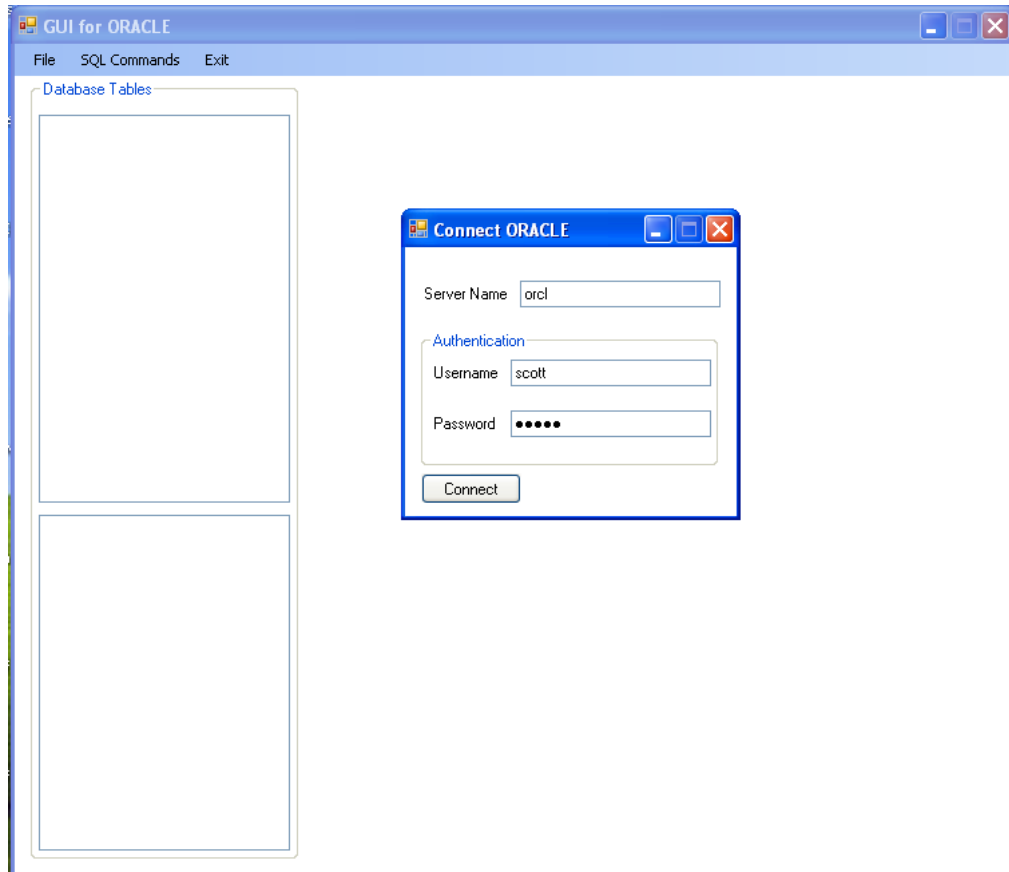


Fig 5.2.1 User login

Database GUI & Switching over Distributed Networks

2. After connecting to oracle, the following list of tables appear that exist already.

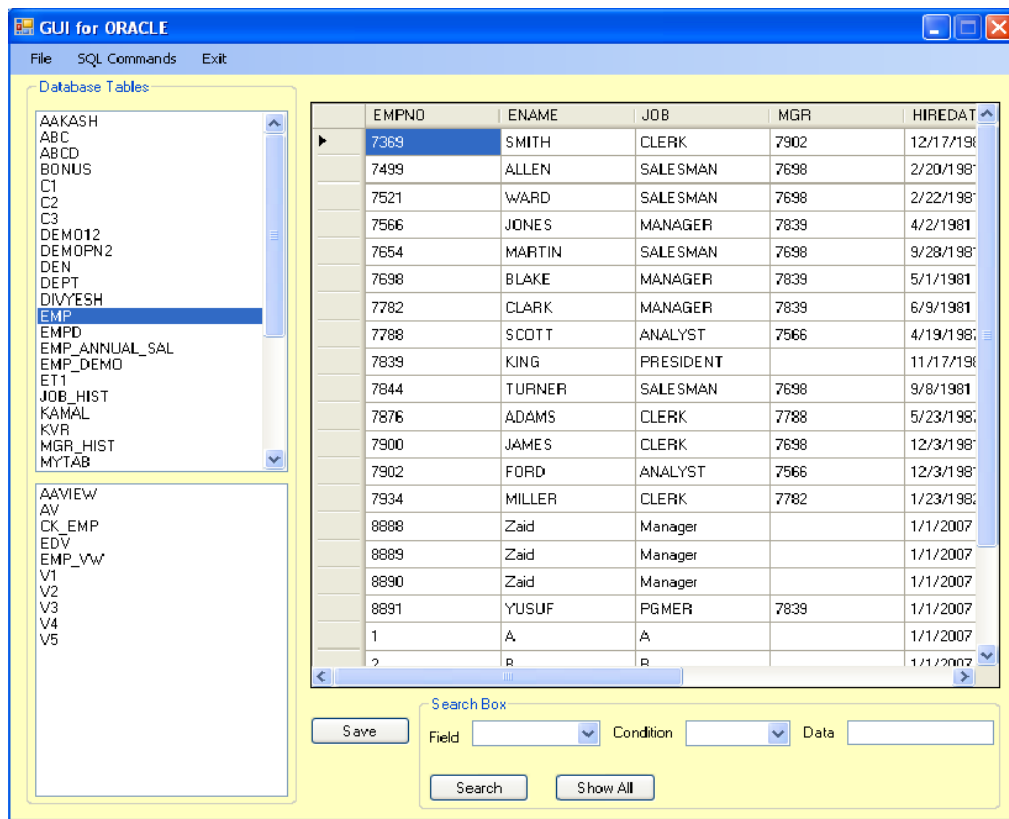


Fig 5.2.2 Existing tables

3. Adding a new field to the existing table.

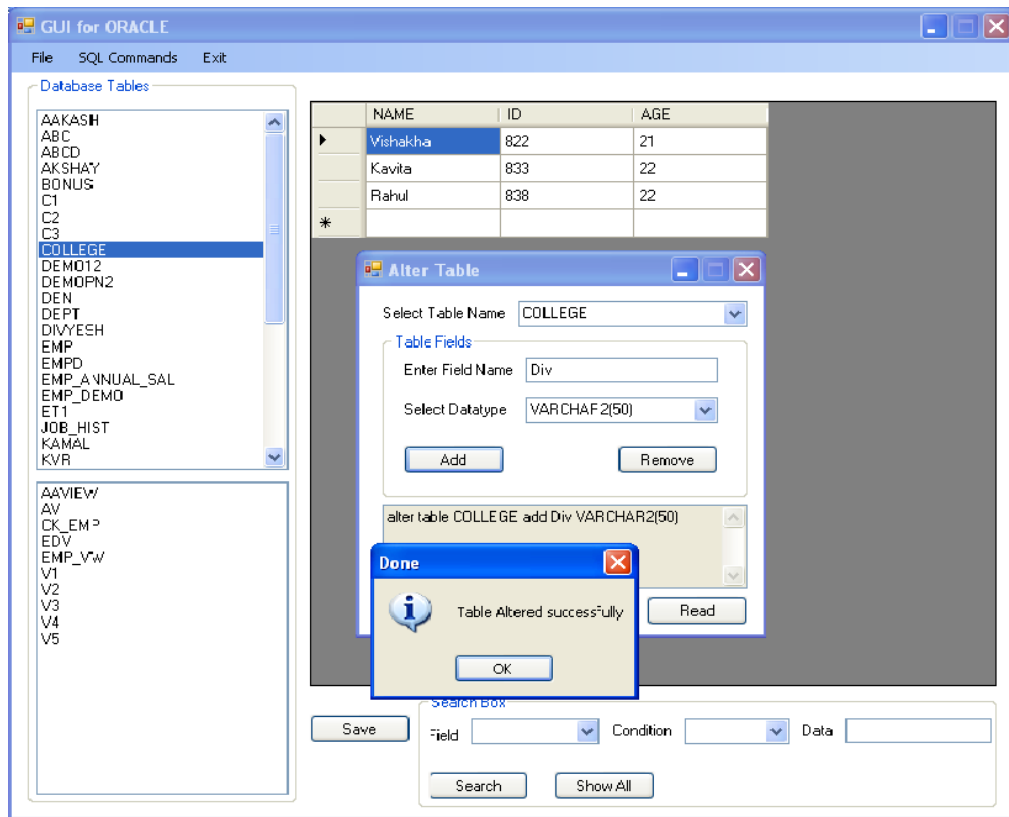


Fig 5.2.3 Alter table

4. Drop table drops the table from the list of database tables.

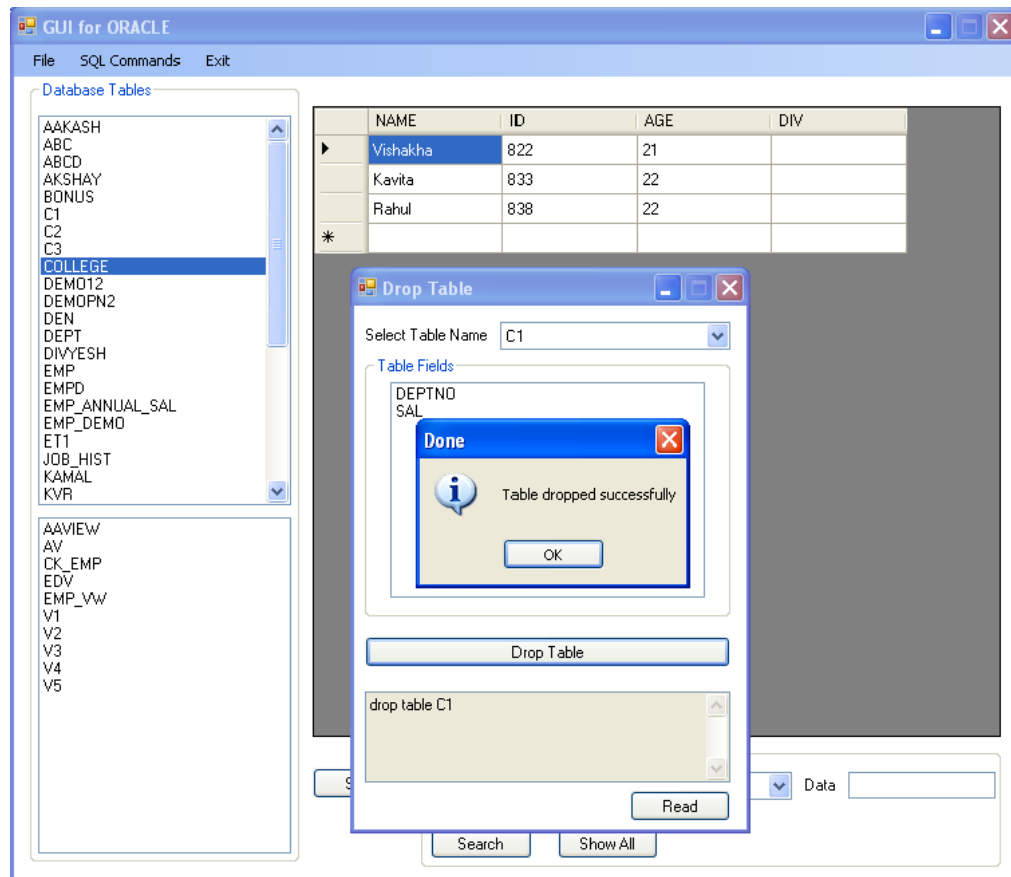


Fig 5.2.4 Drop table

5. In Create View, a view name is to be entered along with the table and the required fields.

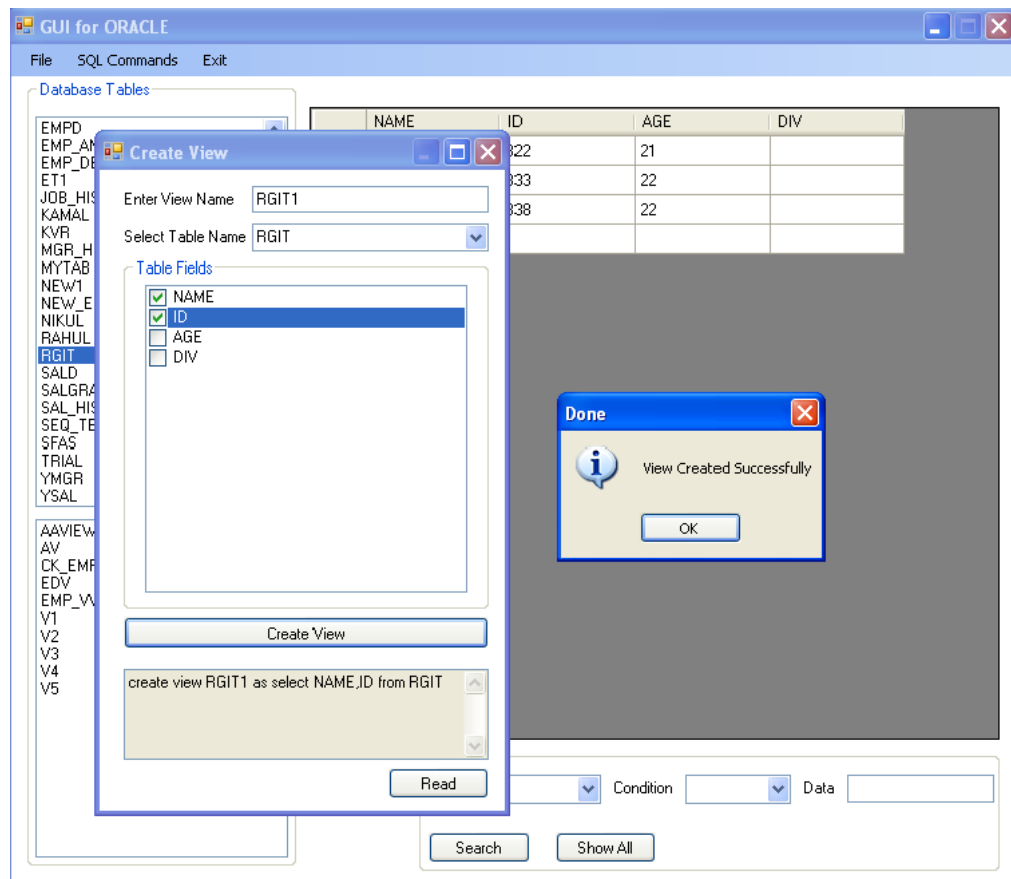


Fig 5.2.5 Create view

6. A view created can also be dropped by just selecting a view name.

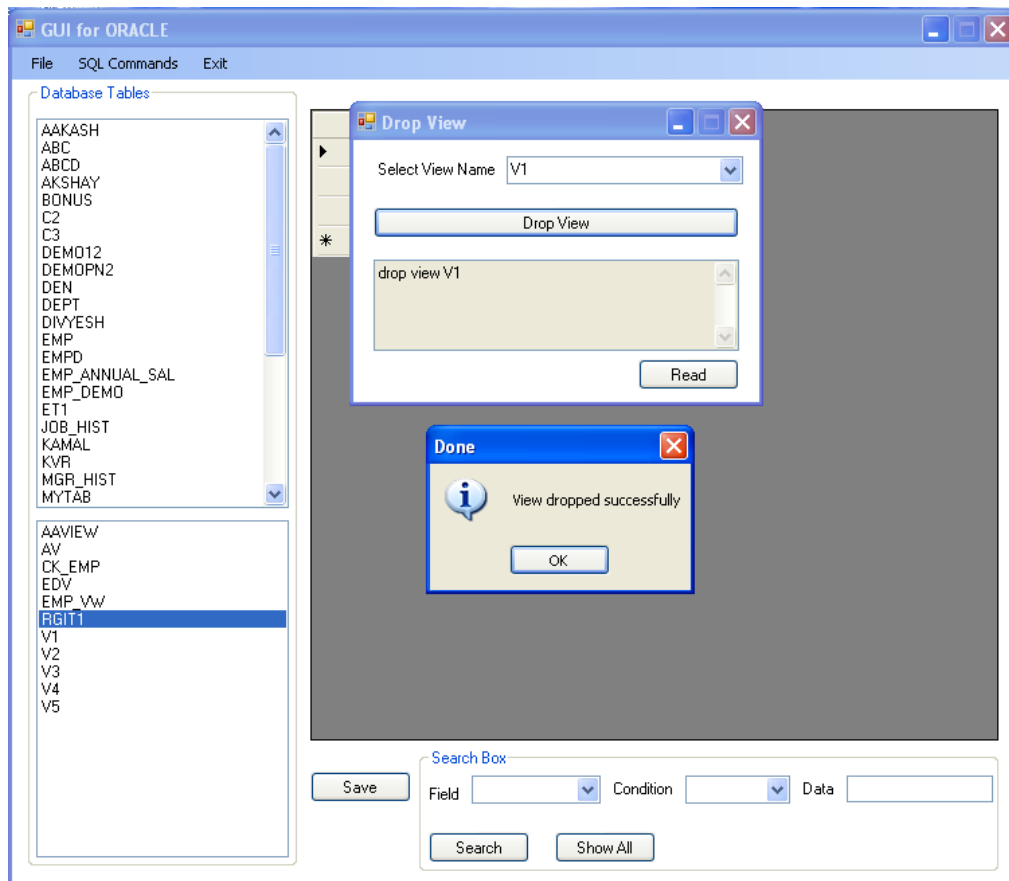


Fig 5.2.6 Drop view

7. A user can be created that requires Username and a password.

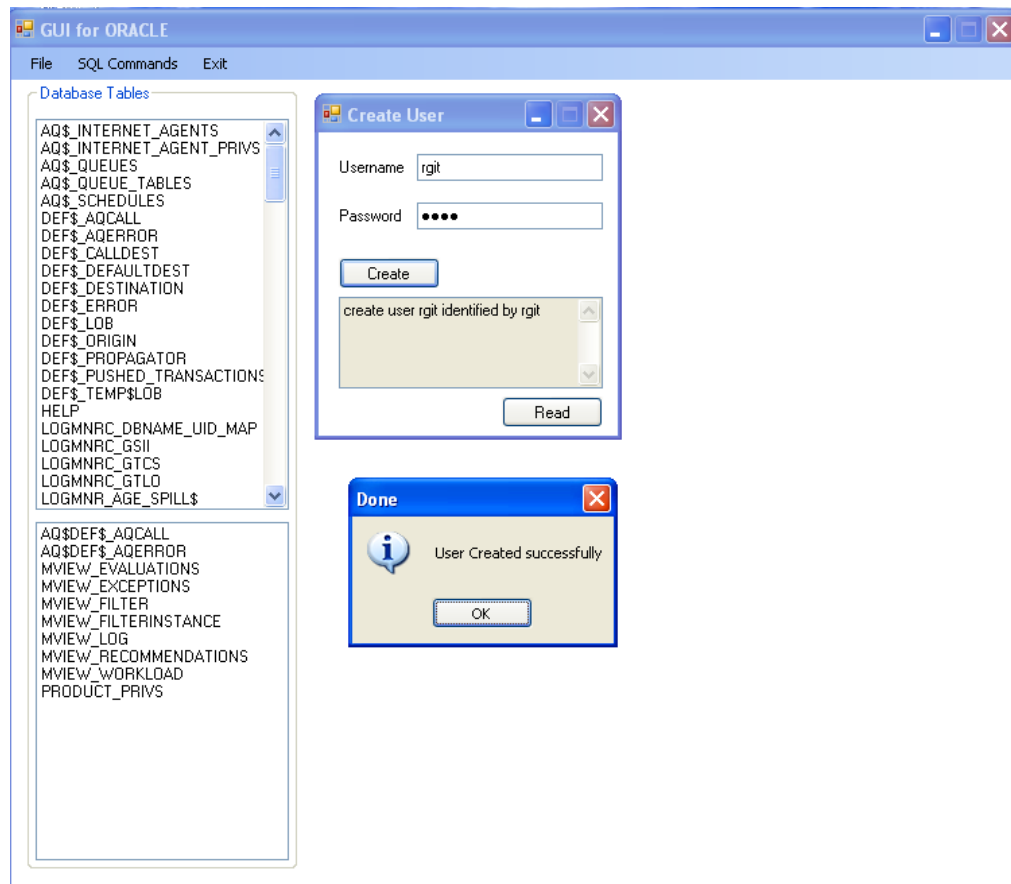


Fig 5.2.7 Create User

8. A user can be given the rights of access using the Grant command.

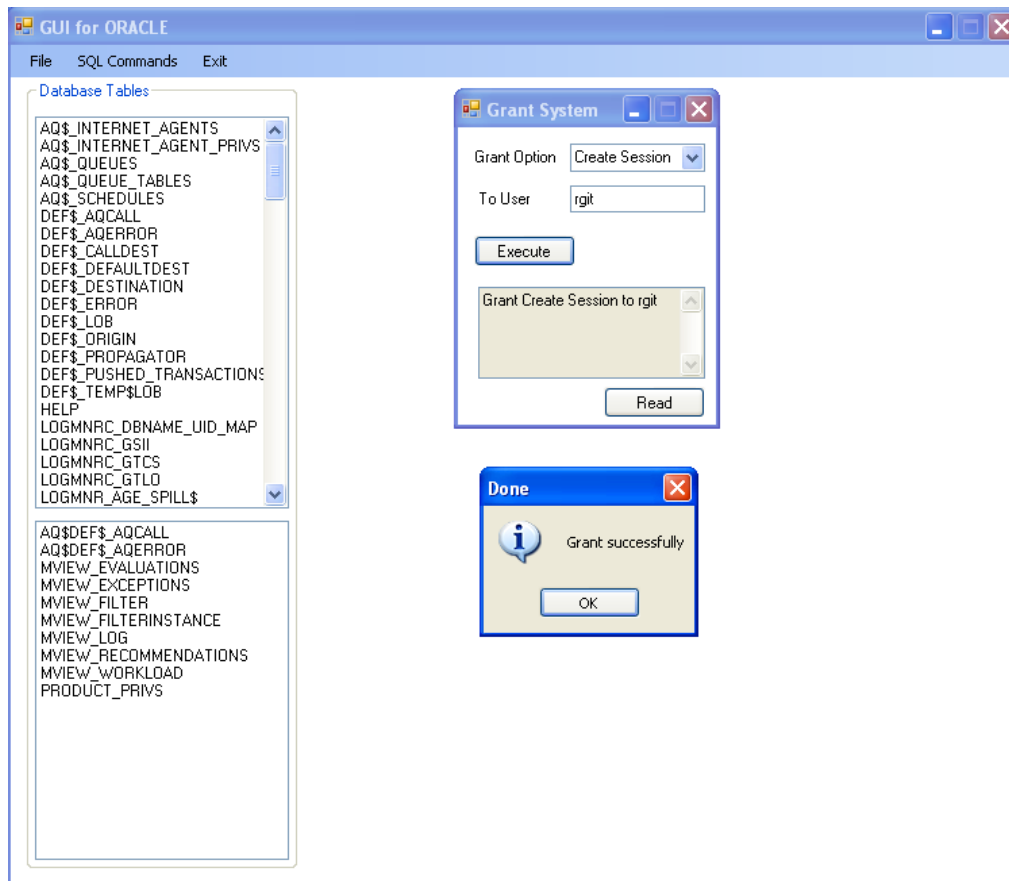


Fig 5.2.8 Grant System

9. The System rights can also be revoked from the user who is denied access.

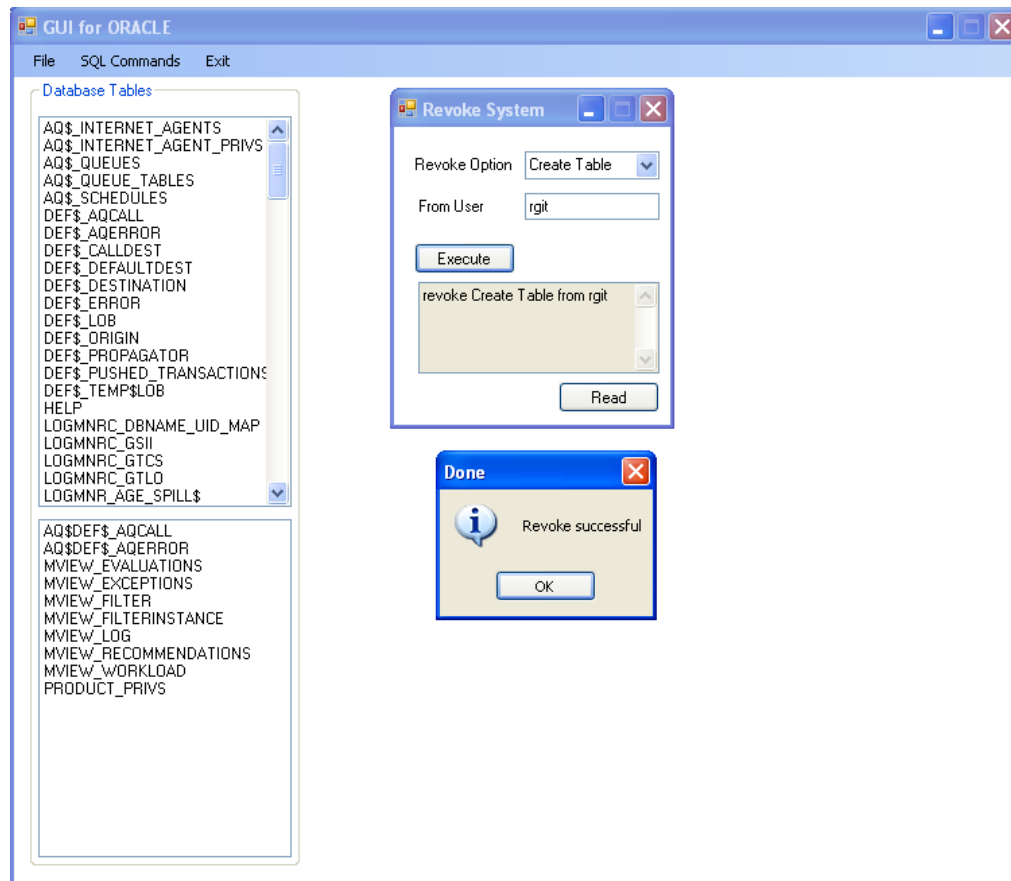


Fig 5.2.9 Revoke System

10. Creating .dsf file from source database

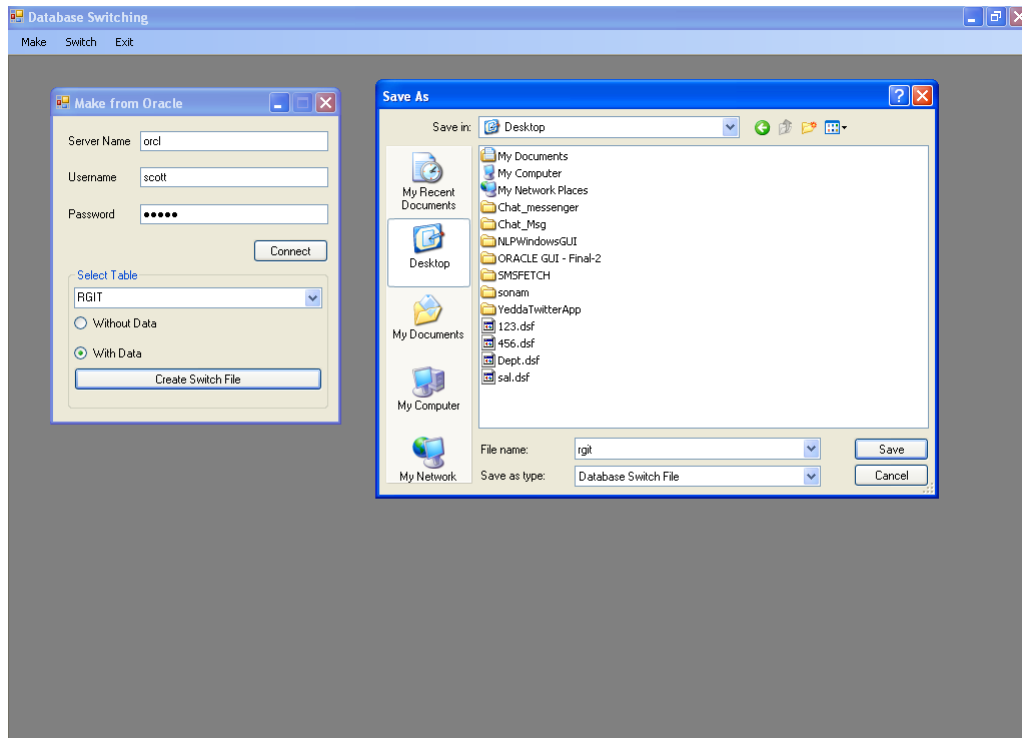


Fig 5.2.10 Saving the Switched File

11. Switching done to the destination database

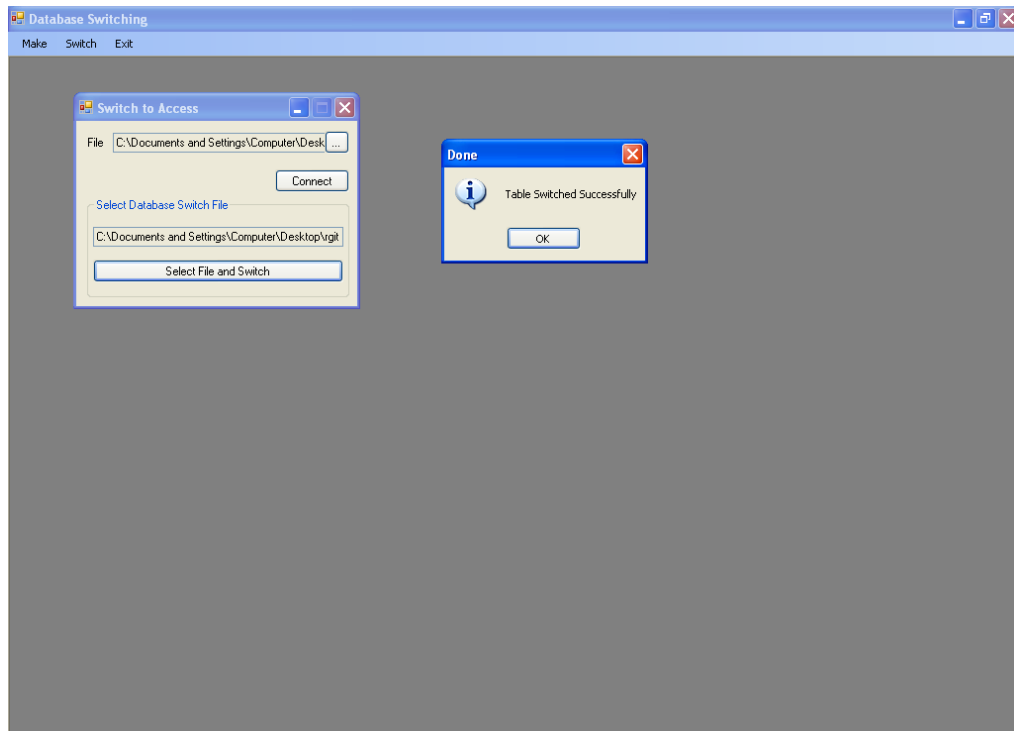


Fig 5.2.11 Switching file to Access

Chapter 6

6. Technologies used

6.1 Visual Studio

- Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web Services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions. In addition, these languages leverage the functionality of the .NET Framework, which provides access to key technologies that simplify the development of ASP Web applications and XML Web Services.
- Windows Forms is for creating Microsoft Windows applications on the .NET Framework. This framework provides a clear, object-oriented, extensible set of classes that enables you to develop rich Windows applications. Additionally, Windows Forms can act as the local user interface in a multi-tier distributed solution.
- Visual Studio includes a host of visual designers to aid in the development of applications. These tools include:

6.2 Windows Forms Designer

- The Windows Forms designer is used to build GUI applications using Windows Forms. Layout can be controlled by housing the controls inside other containers or locking them to the side of the form. Controls that display data (like textbox, list box, grid view, etc.) can be bound to data sources like databases or queries. Data-bound controls can be created by dragging items from the Data Sources

window onto a design surface. The UI is linked with code using an event-driven programming model. The designer generates either C# or VB.NET code for the application.

- Microsoft Visual Studio is a complete set of development tools for building ASP.NET Web applications, XML Web services, desktop applications, and mobile applications. Visual Basic, Visual C++, Visual C#, and Visual J# all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions.
- The Microsoft Visual Studio development system is a suite of development tools designed to aid software developers, whether they are novices or seasoned professionals, face complex challenges and create innovative solutions. Every day, software developers break through tough problems to create software that makes a difference in the lives of others. Visual Studio's role is to improve the process of development to make the work of achieving those breakthroughs easier and more satisfying.

6.3 Microsoft VB.Net 2005

- The .NET Framework is an integral Windows component that supports building and running desktop applications and Web services. It provides a managed execution environment, simplified development and deployment, and support for a wide variety of programming languages.
- Two key components are the common language runtime (CLR), which manages memory, code execution, and other system services, and the .NET Framework class library, which is a collection of reusable types you can use to develop your applications. The .NET Framework also

includes technologies such as ADO.NET, ASP.NET, Windows Presentation Foundation (WPF), Windows Communication Foundation (WCF), and Windows Workflow Foundation (WF).

- Visual Basic .NET (VB.NET), is an object-oriented computer programming language that can be viewed as an evolution of the classic Visual Basic (VB), which is implemented on the .NET Framework. Microsoft currently supplies two major implementations of Visual Basic: Microsoft Visual Studio, which is commercial software and Microsoft Visual Studio Express, which is free of charge. Microsoft implementation of Visual Basic .NET is called "Microsoft Visual Basic".
- Visual Basic 2005 is the name used to refer to the Visual Basic .NET, Microsoft having decided to drop the .NET portion of the title.
- For this release, Microsoft added many features, including:
 1. Edit and Continue
 2. Design-time expression evaluation.
 3. The My pseudo-namespace which provides:
 - a. easy access to certain areas of the .NET Framework that otherwise require significant code to access
 - b. dynamically-generated classes (notably My.Forms)
 4. Improvements to the VB-to-VB.NET converter
 - a. The Using keyword, simplifying the use of objects that require the Dispose pattern to free resources
 - b. Just My Code, which when debugging hides (steps over) boilerplate

Code written by the Visual Studio .NET IDE and system library code

- The above functions (particularly MySql) are intended to reinforce Visual Basic .NET's focus as a rapid application development platform and further differentiate it from C#.
- Visual Basic 2005 introduced features meant to fill in the gaps between itself and other "more powerful" .NET languages, adding:
- .NET 2.0 languages features such as:

generics

Partial classes, a method of defining some parts of a class in one file

and then adding more definitions later; particularly useful for

integrating user code with auto-generated code

Nullable Types

Database GUI has Oracle in backend. For switching there are 4

different types of databases such as Oracle, My SQL, SQL server,

Access.

6.4 Oracle

- Users of the Oracle databases refer to the server-side memory-structure as the SGA (System Global Area). The SGA typically holds cache information such as data-buffers, SQL commands, and user information. In addition to storage, the database consists of online redo logs (or logs), which hold transactional history. Processes can in turn archive the online redo logs into archive logs (offline redo logs),

which provide the basis (if necessary) for data recovery and for some forms of data replication.

6.5 My SQL

- My SQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Many of the world's largest and fastest-growing organizations use MySQL to save time and money powering their high-volume Web sites, critical business systems, and packaged software — including industry leaders such as Yahoo!, Alcatel-Lucent, Google, Nokia, YouTube, Wikipedia, and Booking.com.
- The flagship MySQL offering is MySQL Enterprise, a comprehensive set of production-tested software, proactive monitoring tools, and premium support services available in an affordable annual subscription.
- MySQL is a key part of LAMP (Linux, Apache, MySQL, PHP / Perl / Python), the fast-growing open source enterprise software stack. More and more companies are using LAMP as an alternative to expensive proprietary software stacks because of its lower cost and freedom from platform lock-in.

6.6 SQL Server

- Microsoft SQL Server is a relational database server, developed by Microsoft. It is a software product whose primary function is to store and retrieve data as requested by other software applications, be it those on the same computer or those running on another computer across a network (including the Internet).

- There are at least a dozen different editions of Microsoft SQL Server aimed at different audiences and for different workloads (ranging from small applications that store and retrieve data on the same computer, to millions of users and computers that access huge amounts of data from the Internet at the same time).

6.7 Microsoft Access

- Microsoft Access, is a relational database management system from Microsoft that combines the relational Microsoft Jet Database Engine with a graphical user interface and software-development tools. It is a member of the Microsoft Office suite of applications, included in the Professional and higher editions
- MS Access stores data in its own format based on the Access Jet Database Engine. It can also import or link directly to data stored in other applications and databases.

6.8 Queries

- The most common operation in SQL is the query, which is performed with the declarative `SELECT` statement. `SELECT` retrieves data from one or more tables, or expressions. Standard `SELECT` statements have no persistent effects on the database. Some non-standard implementations of `SELECT` can have persistent effects, such as the `SELECT INTO` syntax that exists in some databases.
- Queries allow the user to describe desired data, leaving the database management system (DBMS) responsible for planning, optimizing, and performing the physical operations necessary to produce that result as it chooses.

- A query includes a list of columns to be included in the final result immediately following the SELECT keyword. An asterisk ("*") can also be used to specify that the query should return all columns of the queried tables. SELECT is the most complex statement in SQL, with optional keywords and clauses.

6.9 Tables

- Tables are the basic structure where data is stored in the database. Given that in most cases, there is no way for the database vendor to know ahead of time what your data storage needs are, chances are that you will need to create tables in the database yourself.
- Many database tools allow you to create tables without writing SQL, but given that tables are the container of all the data, it is important to include the "CREATE TABLE" syntax. Tables are divided into rows and columns. Each row represents one piece of data, and each column can be thought of as representing a component of that piece of data.

Chapter 7

7. Test Cases

- White-box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.
- White-box testing (also known as clear box testing, glass box testing, transparent box testing, and structural testing) is a method of testing **software** that tests internal structures or workings of an application, as opposed to its functionality (i.e. **black-box testing**). In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases.
- The tester chooses inputs to exercise paths through the code and determine the appropriate outputs. While white-box testing can be applied at the **unit**, **integration** and **system** levels of the **software testing** process, it is usually done at the unit level.
- It can test paths within a unit, paths between units during integration, and between subsystems during a system-level test. Though this method of test design can uncover many errors or problems, it might not detect unimplemented parts of the specification or missing requirements.
- In our project, suppose a user forgets to enter a field, he will be prompted immediately saying to complete the necessary details.
- For e.g. while working with the switching system, if the user selects to switch from oracle, he should specify the Server name, Username as well as password.
- If he forgets to enter any of the above details, he will be immediately prompted for error as to complete the above details, and then only he is allowed to proceed further.

- Black-box testing treats the software as a "black box"—without any knowledge of internal implementation. Black-box testing is a method of [software testing](#) that tests the functionality of an application as opposed to its internal structures or workings.
- Specific knowledge of the application's code/internal structure and programming knowledge in general is not required. The tester is only aware of what the software is supposed to do, but not how.
- For e.g. in our project, when he enters a certain input, he gets a certain output i.e. the queries are executed; without being aware of how the output (query) was produced in the first place.
- Test cases are built around specifications and requirements, i.e., what the application is supposed to do. It uses external descriptions of the software, including specifications, requirements, and designs to derive test cases. These tests can be functional or non-functional, though usually functional. The test designer selects valid and invalid inputs and determines the correct output. There is no knowledge of the test object's internal structure.

7.1 Login form:

Sr. No.	Action	Input Data	Expected Result	Actual Result
1.	Pressing button	User name and password	The user is authenticated and the main form appears.	Objective achieved. If user is valid user, then the loading action will take or else an error message is given to the user for entering valid user name or password or both.

Table 7.1 Login form

7.2 Main form:

Sr. No.	Action	Input Data	Expected Result	Actual Result
1.	Selecting menu item from menu bar or from tool bar.		The user is directed to the form according to the menu or tool selected.	Objective achieved. The user is directed to the form depending on the selection of the menu item.

Table 7.2 Main form

7.3 Test plan for buttons that are common in all the forms:

Sr. No	Action	Input Data	Expected Result	Actual Result
1.	Select View option, from Main Menu.		The record related to particular user is displayed.	Objective achieved and user can view the existing records.
2.	After selecting Add option, fill the text boxes and press Save button.	All the text boxes should be filled with valid data.	User can add the new records.	Objective achieved and if not then the error message will appear.
3.	After selecting Delete option, select the appropriate record which you want to delete and then press the delete button.	Record which you want to delete should be selected in drop down menu.	User can delete the existing record which is no longer in use.	Objective achieved and user can delete the existing data.
4.	After selecting clear button, All the data entered in the form gets cleared.	The Text Box gets Cleared after pressing Clear Button Hence no	User can Clear the currently entered data.	Objective achieved and user can clear the existing data.

Database GUI & Switching over Distributed Networks

		need of Inputs.		
5.	After selecting Exit Button from the form we get back to the Exit Screen.	No need of Input Data.	To Exit from the main screen.	Main screen gets Closed and we get back to Exit Screen.
6.	After selecting Exit Project button the form gets exit from screen.	No need of Input Data.	To Exit from the screen.	Project gets Closed.

Table 7.3 Test plan for buttons that are common in all the forms

7.4 Test plan for textboxes in all forms:

Sr. No.	Action	Input Data	Expected Result	Actual Result
1.	Entering data in a text box in which only text is allowed	Text	Data goes into textboxes.	Objective is achieved and data gets entered.
2.	Entering data in a text box in which only text is allowed	Numbers and special characters	Data should not get enters in textboxes.	Objective is achieved and data does not get entered.
3.	Entering data in a text box in which only numbers are allowed	Numbers	Data goes into textboxes.	Objective is achieved and data gets entered.
4.	Entering data in a text box in which only numbers are allowed	Text and special characters	Data should not gets enters in textboxes.	Objective is achieved and data does not get entered.

Table 7.4 Test plan for textboxes in all forms

7.5 Test plan for Make from, Switch to, Speech mode:

Sr. No.	Action	Input Data	Expected Result	Actual Result
1.	Select Make from and Switch to option from main menu, It will generate Switch file for Switching the queries as per option selected.	Data Store in the database.	The switch file is generated in order to execute the queries in different databases.	Objective Achieved.
2.	Listen to the various queries using speech mode	For listening to the queries, at the bottom of every form we have a read button that reads the query using SAPI software.	If correct details for generating the query is selected the query can be listened successfully. Else Fail.	Objective Achieved

Table 7.5 Test plan for Make from, Switch to, Speech mode

Chapter 8

8. Project time line

8.1 Gantt Chart

	Task Name	Duration	Start	Finish	Predecessor	Assignment Owner
1	Database GUI and Switching Over Distributed Networks	177 days?	Tue 8/16/14	Thu 4/12/14		
2	Documentation	155 days?	Mon 8/22/14	Mon 3/19/14		Amruta Pimpale
3	Problem and Scope Definition	2 days?	Tue 8/16/14	Wed 8/17/14		Bhagyashri Redekar
4	Analysis	54 days?	Thu 8/18/14	Wed 10/26/14	3	
5	Requirement Gathering	12 days?	Thu 8/18/14	Fri 9/2/14		Tanvi Raut, Bhagyashri Redekar
6	Feasibility Analysis	10 days?	Mon 9/5/14	Fri 9/16/14	5	Amruta Pimple, Tanvi Raut
7	Identification of Development Tools	7 days?	Sat 9/17/14	Mon 9/26/14	6	Bhagyashri Redekar, Amruta Pimple
8	Study of Database Connectivity	11 days?	Tue 9/27/14	Sun 10/9/14	7	Bhagyashri Redekar
9	Study of Database GUI	8 days?	Mon 10/10/14	Wed 10/19/14	8	Amruta Pimple
10	Study of Database Switching	6 days?	Thu 10/20/14	Wed 10/26/14	9	Tanvi Raut
11	Design	5 days?	Thu 10/27/14	Wed 11/2/14	10	
12	Database GUI and Switching architecture	5 days?	Thu 10/27/14	Wed 11/2/14		Bhagyashri Redekar, Tanvi Raut
13	Implementation	22 days?	Thu 1/12/14	Fri 2/10/14	12	
14	Connectivity	6 days?	Thu 1/12/14	Thu 1/19/14		Bhagyashri Redekar
15	Database GUI	6 days?	Fri 1/20/14	Fri 1/27/14	14	Amruta Pimple
16	Database Switching	10 days?	Mon 1/30/14	Fri 2/10/14	15	Tanvi Raut
17	Testing	33 days?	Wed 2/1/14	Fri 3/16/14		
18	Link Testing	6 days?	Wed 2/1/14	Wed 2/8/14		Amruta Pimple
19	Integration Testing	4 days?	Thu 2/9/14	Tue 2/14/14	18	Amruta Pimple
20	Stress and Load Testing	8 days?	Wed 2/15/14	Fri 2/24/14	19	Tanvi Raut
21	Performance Testing	8 days?	Mon 2/27/14	Wed 3/7/14	20	Bhagyashri Redekar, Tanvi Raut
22	Beta Testing	7 days?	Thu 3/8/14	Fri 3/16/14	21	Bhagyashri Redekar, Tanvi Raut, Amruta Pimple
23	Documentation Auditing	11 days?	Mon 3/19/14	Mon 4/2/14	22	Bhagyashri Redekar, Amruta Pimple
24	Deployment	19 days?	Mon 3/19/14	Thu 4/12/14	22	Bhagyashri Redekar, Amruta Pimple, Tanvi Raut

Fig 8.1.1 Gantt chart

8.2 Timeline Chart

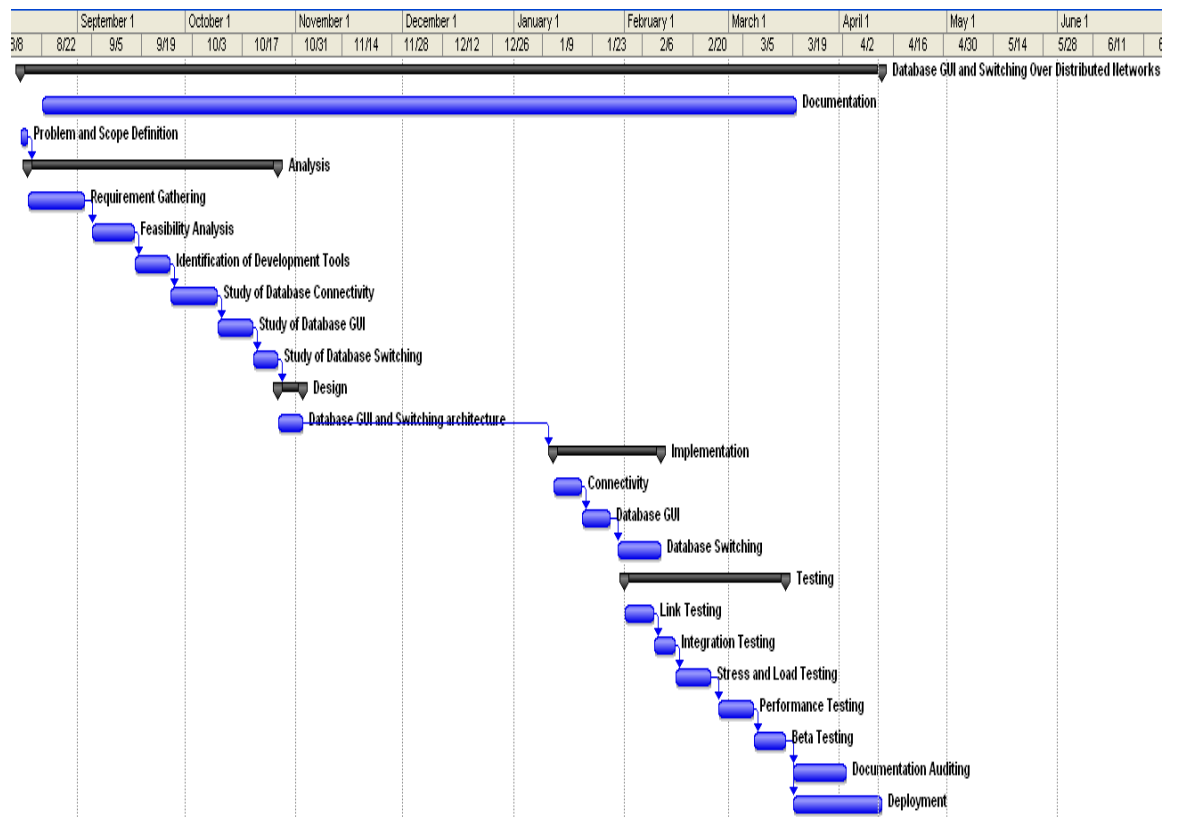


Fig 8.2.1 Timeline Chart

Chapter 9

9. Task Distribution

9.1 Implementation Plan

The following tables give the project plan for the Phase 1 & 2 of our project:

Activity	Description	Effort in person weeks	Deliverable
Phase 1			
P1-01	Requirement Analysis	2 weeks	Requirement Gathering
P1-02	Existing System Study & Literature	3 weeks	Existing System Study & Literature
P1-03	Technology Selection	2 weeks	.NET
P1-04	Modular Specifications	2 weeks	Module Description
P1-05	Design & Modeling	4 weeks	Analysis Report
	Total	13 weeks	

TABLE 9.1.1 Phase 1 of the Project Plan

Database GUI & Switching over Distributed Networks

Activity	Description	Effort in person weeks	Deliverable
Phase 2			
P2-01	Detailed Design	2 weeks	LLD / DLD Document
P2-02	UI and user interactions design	Included in above	UI document
P2-03	Coding & Implementation	12 weeks	Code Release
P2-04	Testing & Bug fixing	2 weeks	Test Report
P2-05	Performance Evaluation	4 weeks	Analysis Report
P2-06	Release	Included in above	System Release
	Total	20 weeks	Deployment efforts are extra

TABLE 9.1.2 Phase 2 of the Project Plan

Chapter 10

10. Conclusion & future work

- In contrast to a traditional setting where users express queries against the database schema, we assert that the semantics of data can often be understood by viewing the data in the context of the user interface (UI) of the software tool used to enter the data. That is, the users will understand the data in a database by seeing the labels, drop-down menus, tool tips, or other help text that are built into the user interface.
- Querying through GUI to execute complex as well as simple queries has proved to be an efficient system to people who do not have much knowledge about databases.
- Switching between the databases reduces the work of rewriting the queries for each of the database thus saving time.
- The Graphical User Interface generates the queries using Oracle as the database. Similarly other databases can also be used for future work in order to execute the queries using databases such as MySQL, Sqlserver, MS access, etc with the GUI
- This use of databases to generate the queries can be successfully implemented so that the work of writing full queries can be reduced and a non-technical user can also use it without any knowledge of databases.
- The system at hand can be divided into two inter-related subsystems:

1. Database GUI

This part of the system deals with generating and executing queries and

Listening to the queries through speech mode using SAPI software.

- There are various processes that are handled by this system are:

1. User can create a table by clicking on d drop down menus.
 2. User can add data to the tables.
 3. User can read the query to be generated.
 4. User can listen to the query using speech mode.
 5. User can search for a particular entry by specifying a condition.
2. Database Switching

User can select among the four databases such as oracle, MySQL, MS access, Sqlserver from which the file is to be switched and can Switch the file to be executed in any of the four databases.

For e.g. user can select the file to be switched from Oracle and switch This file required by other databases such as MySQL, Sqlserver and MS access.

Future Work

Database GUI and switching is based on GUI using Oracle to execute the queries of the user. The proposed System can be enhanced by using the different databases along with the GUI. That is the system can be expanded to execute the queries using Sqlserver, MS access and Sqlserver as well. Thus the database of the system can be modified without much modification in the background working of the core system.

Chapter 11

11. References

❖ Authored Books:-

- Data Engineering (ICDE), 2011 IEEE 27th International Conference
- Industrial and Information Systems, 2007. ICIIS 2007. International Conference
- Oracle 9i Complete Reference By Kevin Loney & Gorge Koch
- Complete Reference-VB.NET By Jeffrey R. Shaprio
- SQL Server Book Online By Microsoft Corporation

❖ Paper In Journal:-

- <http://www.ijcsit.com/docs/Volume%203/vol3Issue3/ijcsit2012030306.pdf>

❖ Websites:-

- <http://www.oracle.com>
- <http://www.mendeley.com/research/querying-through-a-user-interface>
- <http://www.cio.ny.gov/pmmp/guidebook2/SystemReq.pdf>
- <http://www.slideshare.net/mksaad/structure-vs-object-analysis>
- http://www.w3schools.com/sql/sql_intro.asp
- www.ieee.org/publications_standards/publications/periodicals/index

12. Appendix

GUI	Graphical User Interface
SAPI	Speech Application Programming Interface
DML	Data Manipulation Language
DDL	Data Definition Language
SQL	Structured Query Language

➤ Data Manipulation Language(DML):

It is a family of computer languages used by computer programs or database users to retrieve, inserts, deletes and update data in a database. The DML for oracle is SQL. These commands manipulate data within existing database objects, examples are SELECT, UPDATE, INSERT, DELETE statements.

➤ Data Definition Language(DDL):

A DDL is syntax similar to a computer programming language for defining data structures especially database schemas. Examples are CREATE, DROP, ALTER statements.

➤ Graphical User Interface(GUI):

A GUI is a human interface (i.e. a way for humans to interact with computers) that uses windows, icons and menus which can be manipulated by

a mouse (and often to a limited extent by a keyboard as well). GUI stands in sharp contrast to a Command Line Interface (CLI), which uses only text, and are accessed solely by a keyboard.

➤ Oracle:

Oracle is a relational database management system (RDBMS) developed and copyrighted by the Oracle Corporation. It is popular as a back end for client-server databases. It supports SQL and runs on personal computers, minis and main frames.

➤ Structured Query Language:

SQL is a standard interactive and programming language for getting information from and updating a database. Although SQL is both an ANSI and ISO standard, many database products support SQL with proprietary extensions to the standard language. Queries take the form a command language that lets you select, insert, update and find out location of data and so forth