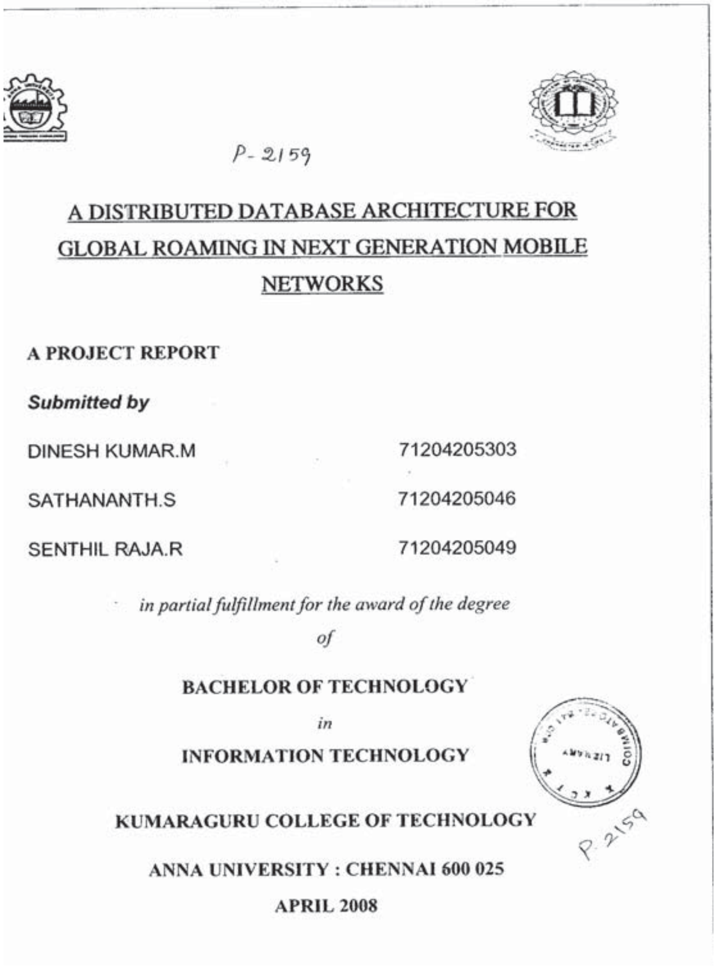


A DISTRIBUTED DATABASE ARCHITECTURE FOR GLOBAL ROAMING IN NEXT GENERATION MOBILE NETWORKS

A PROJECT REPORT

Submitted by

DINESH KUMAR.M	71204205303
SATHANANTH.S	71204205046
SENTHIL RAJA.R	71204205049

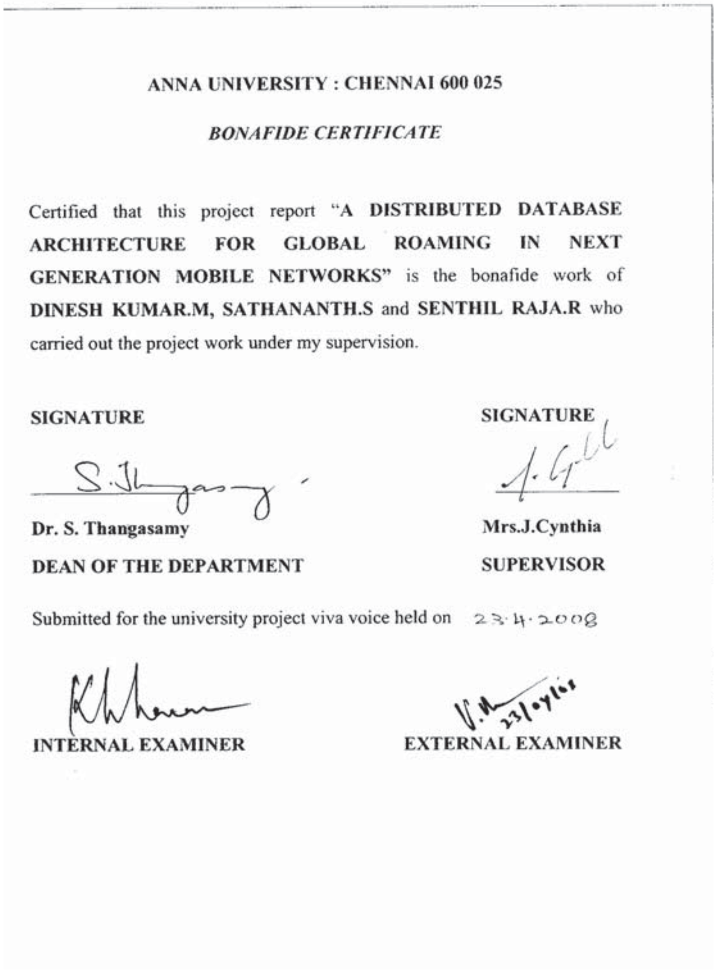
in partial fulfillment for the award of the degree
of

BACHELOR OF TECHNOLOGY
in
INFORMATION TECHNOLOGY

KUMARAGURU COLLEGE OF TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

APRIL 2008



ANNA UNIVERSITY : CHENNAI 600 025

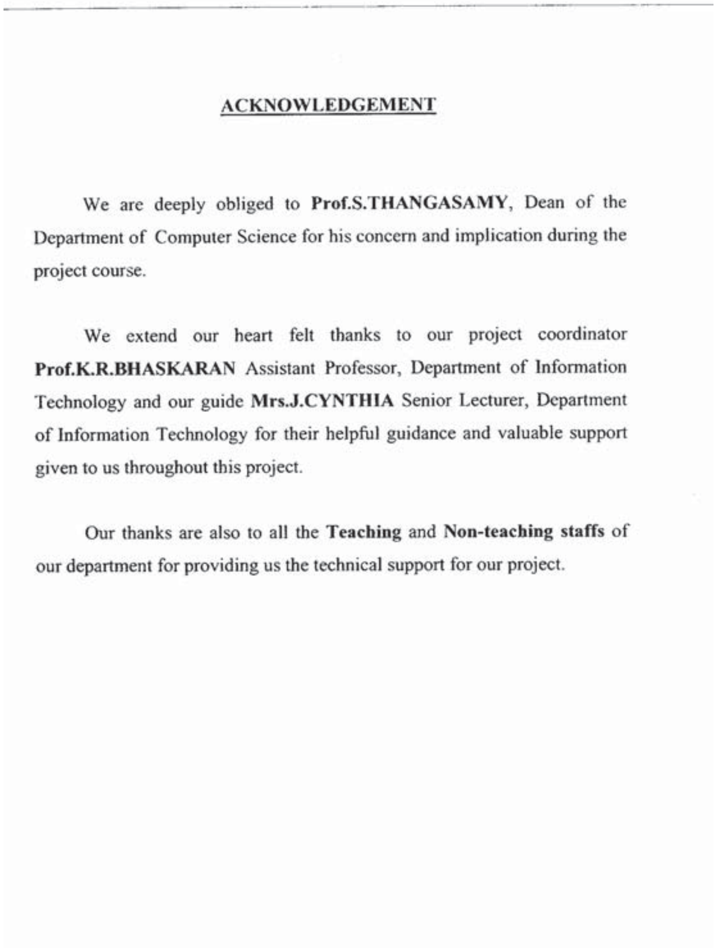
BONAFIDE CERTIFICATE

Certified that this project report **"A DISTRIBUTED DATABASE ARCHITECTURE FOR GLOBAL ROAMING IN NEXT GENERATION MOBILE NETWORKS"** is the bonafide work of **DINESH KUMAR.M, SATHANANTH.S** and **SENTHIL RAJA.R** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
S. Thangasamy	
Dr. S. Thangasamy	**Mrs.J.Cynthia**
DEAN OF THE DEPARTMENT	**SUPERVISOR**

Submitted for the university project viva voice held on 23.4.2008

	23/04/08
INTERNAL EXAMINER	**EXTERNAL EXAMINER**

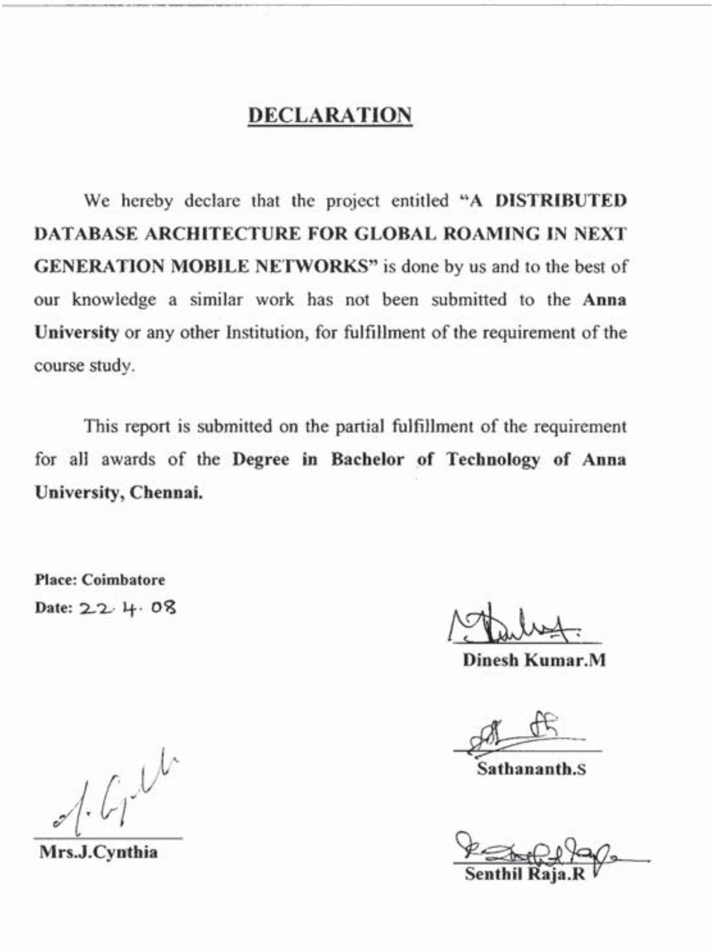


ACKNOWLEDGEMENT

We are deeply obliged to **Prof.S.THANGASAMY**, Dean of the Department of Computer Science for his concern and implication during the project course.

We extend our heart felt thanks to our project coordinator **Prof.K.R.BHASKARAN** Assistant Professor, Department of Information Technology and our guide **Mrs.J.CYNTHIA** Senior Lecturer, Department of Information Technology for their helpful guidance and valuable support given to us throughout this project.

Our thanks are also to all the **Teaching** and **Non-teaching staffs** of our department for providing us the technical support for our project.



DECLARATION

We hereby declare that the project entitled **"A DISTRIBUTED DATABASE ARCHITECTURE FOR GLOBAL ROAMING IN NEXT GENERATION MOBILE NETWORKS"** is done by us and to the best of our knowledge a similar work has not been submitted to the **Anna University** or any other Institution, for fulfillment of the requirement of the course study.

This report is submitted on the partial fulfillment of the requirement for all awards of the **Degree in Bachelor of Technology of Anna University, Chennai.**

Place: Coimbatore
Date: 22.4.08

Dinesh Kumar.M

Sathananth.S

Mrs.J.Cynthia

Senthil Raja.R

ABSTRACT

The next-generation mobile network will support terminal mobility, personal mobility, and service provider portability, making global roaming seamless. A location–Independent Personal Telecommunication number(PTN) scheme is conducive to implementing such a global mobile System. However, the non-geographic PTNs coupled with the anticipated large number of mobile users in future mobile networks may introduce very large centralized database. This necessitates research into the design and performance of high-throughput database technologies used in mobile system to ensure the future system will be able to carry efficiently the anticipated loads.

This paper proposes a scalable, robust, efficient location database architecture based on the location independent PTNs. The proposed multitree database architecture consists of a number of database subsystems, each of which is a three-level tree structure and is connected to the others only through its root. By exploiting the localized nature of calling and mobility patterns, the proposed architecture effectively reduce the database loads as well as the signaling traffic incurred by the location registration and call delivery procedure. In addition, one memory-resident database index, T-tree is proposed for the location database to further improve their throughput.

Location management procedures involve numerous operations in various databases. These databases record the relevant information of a mobile user, trace the user's location by updating the relevant database entries, and map the user's PTN to its current location. Results have revealed that the proposed database architecture for local management can effectively support the anticipated high user density in the future mobile networks.

TABLE OF CONTENTS

LIST OF ABBREVATIONS:

PTN	- Personal Telecommunication Number
PSTN	- Public Switched Telephone Network
MSC	- Mobile Switching Centre
MT	- Mobile Terminal
RA	- Registration Areas
DS	- Database Subsystems
BC	- Billing Centre

1. INTRODUCTION:

1.1 GENERAL:

The next-generation mobile network will be an integrated global system that provides heterogeneous services across network providers, network backbones, and geographical regions . Global roaming is a basic service of the future mobile networks, where terminal mobility, personal mobility, and service provider portability must be supported. A nongeographic personal telecommunication number (PTN) for each mobile user is desirable to implement these types of mobile freedom. With location-independent PTNs, users can access their personalized services regardless of terminal or attachment point to the network; they can move into different service provider's network and continue to receive subscribed services without changing their PTNs. Another advantage of the flat PTN scheme is that it is much more efficient in terms of capacity than the location-dependent numbering scheme where the capacity of the subscriber number (SN) may be exhausted in a highly populated area, whereas the SN's capacity is wasted in a sparsely populated area . However, using the location-independent numbering plan may introduce large centralized databases into a mobile system. To make things worse, each call may require an interrogation to the centralized databases, thus signaling traffic will grow considerably and call setup time may increase dramatically. The large centralized databases may become the bottleneck of the global mobile system, thus necessitating research into the design and performance of high-throughput database technologies as used in mobile networks to meet future demands.

Location management is one of the most important functions to support global roaming. Location management procedures involve numerous operations in number of databases. These databases record the relevant information of a mobile user, trace the user's location by updating the relevant database entries, and map the user's PTN to its current location. When the MT changes its location, all replicated databases need to be updated for the MT, thus incurring a high database update load and signaling traffic, especially for highly mobile users. In summary, each auxiliary strategy outperforms only the others by the public switched telephone network (PSTN), ATM networks, or other networks. The proposed database architecture is motivated by the following.

1) A location-independent PTN provides a basis for global roaming in the next-generation mobile networks where terminal mobility, personal mobility, and service provider portability will be implemented. A mobile subscriber can retain its lifelong PTN regardless of its location and service provider.

2) The multitree database architecture is much more robust than the one-root hierarchical architecture. In the proposed architecture, an MT's profile is stored in one of the root databases according to its current location. Thus, each root database only maintains a small portion of the user profiles in the global mobile system. The crash of one root database will not disrupt the operation of other root databases, and the recovery of failed root database is much easier than in the one-root database architecture where all user profiles need to be recovered once the root is crashed.

3) The multitree database architecture is scalable, which is crucial to support continuously increasing number of mobile subscribers in future mobile networks. When the capacity of a root database is saturated, a new DS is readily added. More importantly, the end-to-end delay in location registration and call delivery will not increase due to such an expansion in the mobile network. On the other hand, with the one-root structure, when the capacity of root or a high-level database is saturated, more levels of databases need to be added in order to reduce the burden on the root or high-level databases. This will increase the delays in location registration and call delivery.

4) The proposed multitree database system is easy to expand and maintain in the multioperator environment of a global mobile system. With the multitree architecture, each service provider can have its own DSs and its straight forward for a service provider to expand its service coverage by adding new DSs. It is also easy to operate and manage a DS when the DS is owned by a single service provider.The one-root architecture,however, may not have such advantages.

1.2 PROBLEM DEFINITION:

Many mobile applications contain enormous data and intricate search which are easy to management if those data are preserved into one hard disk or the Disk Array. Therefore, organizing data structure according to device's capacity and system's priority is the way to improve system usability, availability.

In order to enable handling enormous information, we propose a database application architecture, which provides reliable and configurable data storage services to embedded application system with the performance, scalability and reliability. This paper presents a solution for embedded database application architecture.

Maintaining very large databases and retrieving data from the databases is definitely a tedious job. There may not be a proper flow for retrieving data from the databases. Since the database is very large, checking process in the database takes large time. Hence there should be a single database. Also there should be a proper flow for accessing the database.

The transmission time will be high because of many databases. Since data have to be accessed from various databases traffic may be cumbersome to route calls between the mobile users. Moreover the cost have to be considerable. The cost for maintaining such many databases will be of great expense..

1.3 OBJECTIVES:

The next-generation mobile network will be an integrated global system that provides heterogeneous services across network providers, network backbones. Global roaming is a basic service of the future mobile networks, where terminal mobility, personal mobility, and service provider portability must be supported. A nongeographic personal telecommunication number (PTN) for each mobile user is desirable to implement these types of mobile freedom. With location-independent PTNs, users can access their personalized services regardless of terminal or attachment point to the network; they can move into different service provider's network and continue to receive subscribed services without changing their PTNs. Another advantage of the flat PTN scheme is that it is much more efficient in terms of capacity than the location-dependent numbering scheme where the capacity of the subscriber number (SN) may be exhausted in a highly populated area. However, using the location-independent numbering plan may introduce large centralized databases into a mobile system. To make things worse, each call may require an centralized databases, thus signaling traffic will grow considerably and call setup time may increase dramatically. The large centralized databases may become the bottleneck of the global mobile system, thus necessitating research into the design and performance of high-throughput database technologies as used in mobile networks to meet future demands.

2. LITERATURE REVIEW:

2.1 FEASIBILITY STUDY:

From the inception of ideas for a software system, until it is implemented and delivered to customer and even after that the system undergoes gradual developments and evaluations.

The software is said to have life cycle composed of several phases. At the feasibility stage, it is desirable that two or three different configuration will be pursed that satisfy the key technical requirement but which represent different level of ambition and cost.

Feasibility is the determination of whether or not a project is worth doing. A feasibility study is carried out select a best system that mate performance requirements.

The data collected during primary investigation examines system feasibilities that is likelihood that the system will be beneficial to the organization. Four tests for feasibility study are as follows :-

- ➢ Technical Feasibility: This is concerned with specifying equipment and software that ill successfully satisfy the use considerably, but might include
  - ❖ The feasibility to produce output in a given time because system is fast enough to handle multiple users.
  - ❖ Response time under certain circumstances and ability to process a certain volume of transaction of a particular speed.
  - ❖ Feasibility to communicate data to distant location.
- ➢ Economical Feasibility: Economic analysis is the most frequently used technique used for evaluating the effectiveness of a proposed system. More commonly known as cost/benefit analysis the procedure is to determine the benefits and savings that are expected from a proposed system and compared them with cost. Though the cost of installing the system may appear high, it is one time investment. This results in time and cost saving. The system development costs will be significantly low, so the proposed system is economically feasible.The resulting benefits is that automation results in turn around time. The resulting cost/benefit ratio is favorable.
- ➢ Operational Feasibility: It is mainly related to human organizational as social aspects. The points to be considered are :- The system interface is standard, user friendly and provides extensive help. Hence no training is not required. There is a sufficient support for the project from the organization. The proposed system is acceptable to the users. So the proposed system is operationally feasible.
- ➢ Social Feasibility: Social feasibility is determination of whether a proposed project will be acceptable to people or not, So this project is totally Social and Feasible.

2.1.1 CURRENT STATUS OF THE PROBLEM:

Maintaining very large databases and retrieving data from the databases is definitely a tedious job. There may not be a proper flow for retrieving data from the databases. Since the database is very large, checking process in the database takes large time. Hence there should be a single database. Also there should be a proper flow for accessing the database.

The transmission time will be high because of many databases. Since data have to be accessed from various databases traffic may be cumbersome to route calls between the mobile users. Moreover the cost have to be considerable. The cost for maintaining such many databases will be of great expense.

This project provides a solution for embedded database application architecture. The software presents a T-tree database architecture whereas a root table will contain child tables which comes under the root. Since these tables are embedded in a T-Tree database , the database checking is greatly reduced. The transmission time is also very low.

The auxiliary strategies try to exploit the spatial and temporal locality in each user's calling and mobility patterns to reduce the signaling traffic and database loads. Also, this helps to access the data from the entire database very easily. Hence routing call to the mobile become more easy. Therefore the problem of network traffic is well reduced. The database load is reduced. This reduces the network server burden. When the cost is concerned, it might not be very much expensive.

2.1.2 PROPOSED SYSTEM & ADVANTAGES:

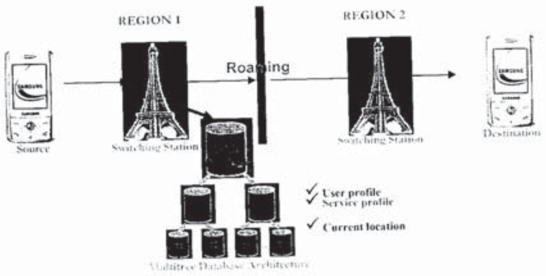
The proposed database system is a multitree structure, consisting of a number of distributed database subsystems, each of which is a three-level tree structure. The proposed database architecture is motivated by the following.

1) A location-independent PTN provides a basis for global roaming in the next-generation mobile networks where terminal mobility, personal mobility, and service provider portability will be implemented.

2) The multitree database architecture is much more robust than the one-root hierarchical architecture. In the proposed architecture

3) The multitree database architecture is scalable, which is crucial to support continuously increasing number of mobile subscribers in future mobile networks

PROPOSED SYSTEM



4) The proposed multitree database system is easy to expand and maintain in the multioperator environment of a global mobile system.

The Advantages of the system are the following:

- Low Cost.
- No Need Of Many Databases.
- Reduce The Database Loading.
- Less Transmission Time.
- Reduce Signaling Traffic.

2.2HARDWARE REQUIREMENTS:

Processor	:	P III 800 MHz
Ram	:	256 MB
Hard Disk	:	40 GB

2.3SOFTWARE REQUIREMENTS:

Languages	:	VB.NET,SQL
Operating System	:	Windows XP

2.4 SOFTWARE OVERVIEW

Microsoft Visual Basic . Net used as front end tool. The reason for selecting Visual Basic dot Net as front end tool as follows:

- Visual Basic . Net has flexibility , allowing one or more language to interoperate to provide the solution. This Cross Language Compatibility allows to do project at faster rate.
- Visual Basic . Net has Common Language Runtime , that allows all the component to converge into one intermediate format and then can interact.
- Visual Basic . Net has provide excellent security when your application is executed in the system
- Visual Basic .Net has flexibility, allowing us to configure the working environment to best suit our individual style. We can choose between a single and multiple document interfaces, and we can adjust the size and positioning of the various IDE elements.
- Visual Basic . Net has Intelligence feature that make the coding easy and also Dynamic help provides very less coding time.

- The working environment in Visual Basic .Net is often referred to as Integrated Development Environment because it integrates many different functions such as design, editing, compiling and debugging within a common environment. In most traditional development tools, each of separate program, each with its own interface.
- The Visual Basic .Net language is quite powerful – if we can imagine a programming task and accomplished using Visual Basic .Net.
- After creating a Visual Basic . Net application, if we want to distribute it to others we can freely distribute any application to anyone who uses Microsoft windows. We can distribute our applications on disk, on CDs, across networks, or over an intranet or the internet.
- Toolbars provide quick access to commonly used commands in the programming environment. We click a button on the toolbar once to carry out the action represented by that button. By default, the standard toolbar is displayed when we start Visual Basic. Additional toolbars for editing, form design, and debugging can be toggled on or off from the toolbars command on the view menu.
- Many parts of Visual Basic are context sensitive. Context sensitive means we can get help on these parts directly without having to go through the help menu. For example, to get help on any keyword in the Visual Basic language, place the insertion point on that keyword in the code window and press F1.
- Visual Basic interprets our code as we enter it, catching and highlighting most syntax or spelling errors on the fly. It's almost like having an expert watching over our shoulder as we enter our code.

3. DETAILS OF THE METHODOLOGY EMPLOYED:

This software maintains a centralized database with a set of tables containing the entire details of each and every user. The database is designed through Sql[Structured Query Language]. The software is coded with VB.Net which is one of the high level languages. These two languages alone are used in this software.

The proposed database architecture for location tracking is a multitree structure, where each subsystem is a three-level architecture referred to as a database subsystem (DS) in this paper. Various DSs may represent networks operated possibly by different service providers. All these DSs are interconnected together via a fixed network, such as PSTN or ATM network, and communicate with each other only through their root databases. This architecture can support a multioperator environment which is expected in future mobile networks.

The proposed system is based on these four modules. These modules builds the entire project in a proper systematic manner. In addition to the multitree location database architecture, this paper also proposes the indexing schemes for each type of location databases and analyzes their efficiency and cost in terms of database access time and storage requirement. The location registration and call delivery procedures based on the proposed database structure are also given. Analysis models are developed to study the service response time of each type of databases in the proposed multitree architecture as well as the end-to-end delays incurred by the proposed location registration and call delivery procedures.

The proposed system is based on the following modules. They are as follows:

1. Simulation Screens

Simulation screen refers to the development of the simulation environment of the project containing organized nodes and cells in a map depicting the real world. Simulation screens are developed to study the service response time of each type of databases in the proposed multitree architecture as well as the end-to-end delays incurred by the proposed location registration and call delivery procedures.

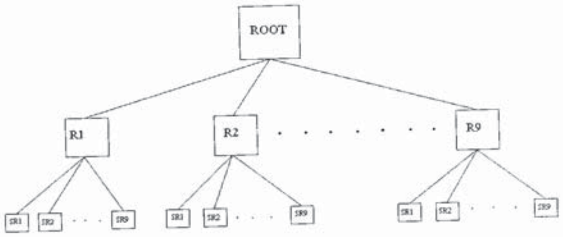
2. PTN Scheme

It is the global numbering scheme for each user instead of subscriber numbering. A personal mobile number is provided to every user according to the region where they reside. A mobile subscriber can retain its lifelong PTN regardless of its location and service provider.

3. T-tree Database Architecture

The proposed database architecture for location tracking is a multitree structure, where each multitree database architecture is for storing home and visitor nodes information and routing tables. The T-tree is fast and contains a number of data items in each node ,thus having good storage utilization.

DATABASE ARCHITECTURE:



4. **Billing Centre**

Billing centre refers to the centre where there will be a billing section which will calculates the cost of each call and the details of the calls done by the user is evaluated and displayed.

4. CONCLUSION:

A distributed multitree database architecture has been proposed for location management in a global mobile system, where the location-independent PTNs are employed to support seamless global roaming. To support the anticipated large number of mobile users in the future mobile system, the efficient database access structure the T-tree were proposed to achieve high database throughput, so that the end-to-end delays in location registration and call delivery can meet the delay requirements in mobile networks.

The proposed database architecture is scalable, robust, and efficient. Compared to the existing two-level location database architecture, the proposed database architecture can support a much higher user density while reducing signaling load significantly. Compared to the one-root tree architecture, the proposed architecture provides better scalability and reliability while supporting a larger user population at a lower signaling cost.

For performance evaluation, analysis model was developed. Numerical results have revealed that the proposed database architecture can effectively handle the anticipated high update and query rates to the location databases in future mobile networks. The proposed database access structures are also suitable for other large centralized databases in mobile networks, such as the authentication center and the equipment identity register.

5. FUTURE ENHANCEMENT:

The project covers only passive subsystems of the organizations. The development can be extended to cover to the other sub systems of the organizations and to have an effective result. The system can be altered according to the needs of the user, which do not affect the present system.

The new system has been developed successfully, since change is a way of life the new requirements in future. The code generated can be further enhanced to simpler forms for the purpose of easier identification. System has been developed in such a way that it will be flexible, so that it can integrate additional procedures in the future to meet any requirements. So the developed and the integrated system will be efficient and flexible.

The coming up modules to be developed in the future in the project is the following:

- Payroll
- Provident Fund

CENTRALISED TABLE:

MobReg	MobNo	Count
region1	91	0
region2	92	1
region3	93	1
region4	94	0
region5	95	0
region6	96	0
region7	97	0
region8	98	1
region9	99	0

SUBREGION TABLE:

Sreg	Mobnos	Mobr	count
subregion981	11	91	0
subregion982	22	91	0
subregion983	33	91	1
subregion984	44	91	0
subregion985	55	91	0
subregion986	66	91	0
subregion987	77	91	0
subregion988	88	91	0
subregion989	99	91	0

DETAILS TABLE:

bileno	names	subreg	mobnotab	mobpic	x	y
5333	bb	33	bb345333	2	150	278

SOURCE CODE:

// MOBILE CREATION

```
Imports System.Data.SqlClient
Public Class mobilecreat
    Dim cnt As New SqlConnection
    Dim da As New SqlDataAdapter
    Dim ds As New DataSet
    Dim countinc As New Integer
    Dim i, dumno As New Integer
    Public areax, areay, cmindex As New Integer
    Dim dum, mobno, mobd As String 'mobile no creater

    Private Sub mobilecreat_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        cnt = New SqlConnection("user
id=sa;password=dinesh;database=mobile;server=rsrd")
        cnt.Open()
        ds.Clear()
        da = New SqlDataAdapter("select * from regno", cnt)
        da.Fill(ds, "regno")
        For i = 0 To ds.Tables(0).Rows.Count - 1
            ComboBox1.Items.Add(ds.Tables("regno").Rows(i).Item(0))
        Next
    End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If TextBox1.Text <> "" Then
            If ComboBox1.SelectedItem <> "" Then
                If ComboBox2.SelectedIndex <> -1 Then
                    If TextBox3.Text <> "" And Len(TextBox3.Text) = 6 Then
                        mobileloc()
                        ds.Clear()
                        da = New SqlDataAdapter("select count from regno", cnt)
                        da.Fill(ds, "regno")
                        countinc =
 (ds.Tables("regno").Rows(ComboBox1.SelectedIndex).Item(2)) + 1
                        ds.Clear()
                        da = New SqlDataAdapter("update regno set count=" +
 CStr(countinc) + "where mobno=" + CStr(91 + ComboBox1.SelectedIndex)
 + "", cnt)
                        da.Fill(ds, "regno")

                        dum = ComboBox1.SelectedItem
                        ds.Clear()
                        da = New SqlDataAdapter("select * from " + dum + "", cnt)
                        da.Fill(ds, dum)
                        countinc =
 (ds.Tables(dum).Rows(ComboBox2.SelectedIndex).Item(3)) + 1
                        ds.Clear()
                da = New SqlDataAdapter("update " + dum + " set count=" +
CStr(countinc) + "where mobnos=" + CStr(11 * (ComboBox2.SelectedIndex
+ 1)) + "", cnt)
                da.Fill(ds, dum)

                dum = ComboBox2.SelectedItem
                mobno = TextBox3.Text
                mobd = "m"
                da = New SqlDataAdapter("insert into " + dum + " values ("
+ mobno + ",'" + TextBox1.Text + "'," + CStr(11 *
(ComboBox2.SelectedIndex + 1)) + ",'" + (TextBox1.Text +
TextBox3.Text) + "','" + CStr(form1.mobselect) + "'," + CStr(areax) + "," +
CStr(areay) + ")", cnt)
                da.Fill(ds, dum)

                dum = CStr(TextBox1.Text + TextBox3.Text)
                da = New SqlDataAdapter("create table " + dum +
"(mobilests varchar(10),callsts varchar(5),callfromto bigint,calldur int,region
varchar(20),calltime varchar(20),calldate datetime)", cnt)
                da.Fill(ds, dum)
                cnt.Close()
                Me.Close()
                form1.regionmap()
            Else
                MsgBox("Enter the 6 digit mobile no", MsgBoxStyle.Critical,
"Mobile number Error")
            End If
```

```
            Else
                MsgBox("Select the Subregion", MsgBoxStyle.Critical,
"SubRegion Error")
            End If
        Else
            MsgBox("Select the region", MsgBoxStyle.Critical, "Region
Error")
        End If
    Else
        MsgBox("Enter the subscriber name", MsgBoxStyle.Critical,
"Subscriber name error")
    End If
End Sub

Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    cnt.Close()
    Me.Close()
    form1.regionmap()
End Sub

Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As
Object, ByVal e As System.EventArgs) Handles
ComboBox1.SelectedIndexChanged
    ComboBox2.Items.Clear()
    mobno = 0
    mobno = (ds.Tables(0).Rows(ComboBox1.SelectedIndex).Item(1))
```

```
    mobd = mobno
    dum = ComboBox1.SelectedItem
    da = New SqlDataAdapter("select * from " + dum + "", cnt)
    da.Fill(ds, dum)
    For i = 0 To ds.Tables(0).Rows.Count - 1
        ComboBox2.Items.Add(ds.Tables(dum).Rows(i).Item(0))
    Next
End Sub

Private Sub ComboBox2_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
ComboBox2.SelectedIndexChanged
    mobno = mobd
    mobno = mobno * 100
    mobno = mobno +
ds.Tables(dum).Rows(ComboBox2.SelectedIndex).Item(1)
    TextBox2.Text = mobno
    cmindex = ComboBox1.SelectedIndex
    da = New SqlDataAdapter("select count from regno", cnt)
    da.Fill(ds, "regno")
    If cmindex = 0 Or cmindex = 3 Or cmindex = 6 Then
        If cmindex = 0 Then
            areay = 68
        ElseIf cmindex = 3 Then
            areay = 175
        ElseIf cmindex = 6 Then
            areay = 278
```

```
        End If
        areax = 46 +
((ds.Tables("regno").Rows(ComboBox1.SelectedIndex).Item(2)) * 15)
    ElseIf cmindex = 1 Or cmindex = 4 Or cmindex = 7 Then
        If cmindex = 1 Then
            areay = 68
        ElseIf cmindex = 4 Then
            areay = 175
        ElseIf cmindex = 7 Then
            areay = 278
        End If
        areax = 150 +
((ds.Tables("regno").Rows(ComboBox1.SelectedIndex).Item(2)) * 15)
    ElseIf cmindex = 2 Or cmindex = 5 Or cmindex = 8 Then
        If cmindex = 2 Then
            areay = 68
        ElseIf cmindex = 5 Then
            areay = 175
        ElseIf cmindex = 8 Then
            areay = 278
        End If
        areax = 250 +
((ds.Tables("regno").Rows(ComboBox1.SelectedIndex).Item(2)) * 15)
    End If
    form1.a(cmindex) =
((ds.Tables("regno").Rows(ComboBox1.SelectedIndex).Item(2)) * 15)
End Sub
```

```
Sub mobileloc()
    If form1.mobselect = 0 Then
        form1.m1.Visible = True
        form1.m1.Left = areax
        form1.m1.Top = areay
        form1.mobselect = 1
        form1.MoveToolStripMenuItem.Visible = True
    ElseIf form1.mobselect = 1 Then
        form1.m2.Visible = True
        form1.m2.Left = areax
        form1.m2.Top = areay
        form1.mobselect = 2
        form1.SelectMobileToolStripMenuItem.Visible = True
    ElseIf form1.mobselect = 2 Then
        form1.m3.Visible = True
        form1.m3.Left = areax
        form1.m3.Top = areay
        form1.mobselect = 3
    ElseIf form1.mobselect = 3 Then
        form1.m4.Visible = True
        form1.m4.Left = areax
        form1.m4.Top = areay
        form1.mobselect = 4
    ElseIf form1.mobselect = 4 Then
        form1.m5.Visible = True
        form1.m5.Left = areax
```

```
        form1.m5.Top = areay
        form1.mobselect = 5
    ElseIf form1.mobselect = 5 Then
        form1.m6.Visible = True
        form1.m6.Left = areax
        form1.m6.Top = areay
        form1.mobselect = 6
    ElseIf form1.mobselect = 6 Then
        form1.m7.Visible = True
        form1.m7.Left = areax
        form1.m7.Top = areay
        form1.mobselect = 7
    ElseIf form1.mobselect = 7 Then
        form1.m8.Visible = True
        form1.m8.Left = areax
        form1.m8.Top = areay
        form1.mobselect = 8
    ElseIf form1.mobselect = 8 Then
        form1.m9.Visible = True
        form1.m9.Left = areax
        form1.m9.Top = areay
        form1.mobselect = 9
    ElseIf form1.mobselect = 9 Then
        form1.m10.Visible = True
        form1.m10.Left = areax
        form1.m10.Top = areay
        form1.mobselect = 10
```

```
    ElseIf form1.mobselect = 10 Then
        form1.m11.Visible = True
        form1.m11.Left = areax
        form1.m11.Top = areay
        form1.mobselect = 11
    ElseIf form1.mobselect = 11 Then
        form1.m12.Visible = True
        form1.m12.Left = areax
        form1.m12.Top = areay
        form1.mobselect = 12
    ElseIf form1.mobselect = 12 Then
        form1.m13.Visible = True
        form1.m13.Left = areax
        form1.m13.Top = areay
        form1.mobselect = 13
    ElseIf form1.mobselect = 13 Then
        form1.m14.Visible = True
        form1.mobselect = 14
    ElseIf form1.mobselect = 14 Then
        form1.m15.Visible = True
        form1.mobselect = 15
    ElseIf form1.mobselect = 15 Then
        form1.m16.Visible = True
        form1.mobselect = 16
    ElseIf form1.mobselect = 16 Then
        form1.m17.Visible = True
        form1.mobselect = 17
```

```
    ElseIf form1.mobselect = 17 Then
        form1.m18.Visible = True
        form1.mobselect = 18
    ElseIf form1.mobselect = 18 Then
        form1.m19.Visible = True
        form1.mobselect = 19
    ElseIf form1.mobselect = 19 Then
        form1.m20.Visible = True
        form1.mobselect = 20
    End If
End Sub


    Private Sub TextBox3_KeyUp(ByVal sender As Object, ByVal e As
System.Windows.Forms.KeyEventArgs) Handles TextBox3.KeyUp
        If Len(TextBox3.Text) = 6 And ComboBox1.SelectedIndex <> -1 And
ComboBox2.SelectedIndex <> -1 Then
            dum = ComboBox2.SelectedItem
            da = New SqlDataAdapter("select * from " + dum + "", cnt)
            da.Fill(ds, dum)
            Label5.Text = "Avaliable"
            For i = 0 To ds.Tables(dum).Rows.Count - 1
                If (TextBox3.Text) = (ds.Tables(dum).Rows(i).Item(0)) Then
                    Label5.Text = "Not avaliable"
                End If
            Next
        ElseIf ComboBox2.SelectedIndex = -1 Then
        Label5.Text = "Sub region Error"
        If ComboBox1.SelectedIndex = -1 Then
          Label5.Text = "Region Error"
        End If
      ElseIf Len(TextBox3.Text) > 0 Then
        Label5.Text = "Checking..."
      End If
    End Sub

End Class



// BILL GENERATION

Imports System.Data.SqlClient
Public Class billfrm
    Dim cnt As New SqlConnection
    Dim dam, dar, das, dan, dan1 As New SqlDataAdapter
    Dim dsm, dsr, dss, dsn, dsn1 As New DataSet
    Dim dum, dum1, dum2, listmob, v As String
    Dim i, j, k, l, chk, callunit, tottime, mi, hr As New Integer

    Private Sub billfrm_Load(ByVal sender As System.Object, ByVal e As
  System.EventArgs) Handles MyBase.Load
        cnt = New SqlConnection("user
  id=sa;password=dinesh;database=mobile;server=rsrd")
```

```
        cnt.Open()
        dam = New SqlDataAdapter("select * from regno", cnt)
        dam.Fill(dsm, "regno")
        For i = 0 To dsm.Tables(0).Rows.Count - 1
          If (dsm.Tables("regno").Rows(i).Item(2)) <> 0 Then
            dum = (dsm.Tables("regno").Rows(i).Item(0))
            dar = New SqlDataAdapter("select * from " + dum + "", cnt)
            dar.Fill(dsr, dum)
            For j = 0 To dsr.Tables(dum).Rows.Count - 1
              If (dsr.Tables(dum).Rows(j).Item(3)) <> 0 Then
                dum1 = (dsr.Tables(dum).Rows(j).Item(0))
                das = New SqlDataAdapter("select * from " + dum1 + "",
cnt)
                das.Fill(dss, dum1)
                For k = 0 To dss.Tables(dum1).Rows.Count - 1
                  listmob = (CStr(dsm.Tables("regno").Rows(i).Item(1)) + ""
+ CStr(dsr.Tables(dum).Rows(j).Item(1)) + "" +
CStr(dss.Tables(dum1).Rows(k).Item(0)))
                  ComboBox1.Items.Add(listmob)
                Next
              End If
            Next
          End If
        Next
        chk = 0
      End Sub
```

```
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        dsm.Clear()
        dam = New SqlDataAdapter("select * from regno", cnt)
        dam.Fill(dsm, "regno")
        For i = 0 To dsm.Tables(0).Rows.Count - 1
          If (dsm.Tables("regno").Rows(i).Item(2)) <> 0 Then
            dum = (dsm.Tables("regno").Rows(i).Item(0))
            dsr.Clear()
            dar = New SqlDataAdapter("select * from " + dum + "", cnt)
            dar.Fill(dsr, dum)
            For j = 0 To dsr.Tables(dum).Rows.Count - 1
              If (dsr.Tables(dum).Rows(j).Item(3)) <> 0 Then
                dum1 = (dsr.Tables(dum).Rows(j).Item(0))
                dss.Clear()
                das = New SqlDataAdapter("select * from " + dum1 + "",
  cnt)
                das.Fill(dss, dum1)
                For k = 0 To dss.Tables(dum1).Rows.Count - 1
                  If (CStr(dsm.Tables("regno").Rows(i).Item(1)) + "" +
  CStr(dsr.Tables(dum).Rows(j).Item(1)) + "" +
  CStr(dss.Tables(dum1).Rows(k).Item(0))) = ComboBox1.SelectedItem And
  chk = 0 Then
                    dum2 = (dss.Tables(dum1).Rows(k).Item(3))
                    dsn.Clear()
                    dan = New SqlDataAdapter("select * from " + dum2 +
  "", cnt)
```

```
                    dan.Fill(dsn, dum2)
                    tottime = 0
                    countc()
                    mi = 0
                    hr = 0
                    While tottime > 59
                      tottime = tottime - 59
                      mi = mi + 1
                    End While
                    While mi > 59
                      mi = mi - 59
                      hr = hr + 1
                    End While
                    th.Text = CStr(CStr(hr) + ":" + CStr(mi) + ":" +
CStr(tottime))
                    sname.Text = CStr(dss.Tables(dum1).Rows(k).Item(1))
                    region1.Text =
CStr(dsm.Tables("regno").Rows(i).Item(0))
                    sregion.Text = CStr(dsr.Tables(dum).Rows(j).Item(0))
                    If RadioButton2.Checked = True And
RadioButton5.Checked = False And RadioButton3.Checked = False And
RadioButton6.Checked = False Then
                      calss("In")
                    ElseIf RadioButton3.Checked = True And
RadioButton6.Checked = False And RadioButton2.Checked = False And
RadioButton5.Checked = False Then
                      calss("Out")
```

```
                ElseIf RadioButton5.Checked = True And
RadioButton2.Checked = False And RadioButton6.Checked = False And
RadioButton3.Checked = False Then
                    regss("home")
                ElseIf RadioButton6.Checked = True And
RadioButton3.Checked = False And RadioButton5.Checked = False And
RadioButton2.Checked = False Then
                    regss("roam")
                ElseIf RadioButton2.Checked = True And
RadioButton5.Checked = True And RadioButton3.Checked = False And
RadioButton6.Checked = False Then
                    calreg("In", "home")
                ElseIf RadioButton3.Checked = True And
RadioButton6.Checked = True And RadioButton2.Checked = False And
RadioButton5.Checked = False Then
                    calreg("Out", "roam")
                ElseIf RadioButton2.Checked = True And
RadioButton6.Checked = True And RadioButton3.Checked = False And
RadioButton5.Checked = False Then
                    calreg("In", "roam")
                ElseIf RadioButton3.Checked = True And
RadioButton5.Checked = True And RadioButton2.Checked = False And
RadioButton6.Checked = False Then
                    calreg("Out", "home")
                Else
                    dsn.Clear()
```

```
                    dan = New SqlDataAdapter("select * from " + dum2
+ "", cnt)
                    dan.Fill(dsn, dum2)
                    callunit = 0
                    countc()
                    DataGridView1.DataSource = dsn.Tables(dum2)
                    cunit.Text = CStr(callunit)
                    dsn1.Clear()
                    dan1 = New SqlDataAdapter("select * from " + dum2
+ " where callsts='Out'and mobilests='home'", cnt)
                    dan1.Fill(dsn1, dum2)
                    callunit = 0
                    tottime = 0
                    countcc()
                    mi = 0
                    If tottime <> 0 Then mi = 1
                    Do While (tottime > 59)
                       tottime = tottime - 59
                       mi = mi + 1
                    Loop
                    lou.Text = mi
                    loc.Text = mi
                    dsn1.Clear()
                    dan1 = New SqlDataAdapter("select * from " + dum2
 + " where callsts='In' and mobilests='roam'", cnt)
                    dan1.Fill(dsn1, dum2)
                    callunit = 0
```

```
                    tottime = 0
                    countcc()
                    mi = 0
                    If tottime <> 0 Then mi = 1
                    Do While tottime > 59
                       tottime = tottime - 59
                       mi = mi + 1
                    Loop
                    riu.Text = mi
                    ric.Text = mi
                    dsn1.Clear()
                    dan1 = New SqlDataAdapter("select * from " + dum2
+ " where callsts='Out'and mobilests='roam'", cnt)
                    dan1.Fill(dsn1, dum2)
                    callunit = 0
                    tottime = 0
                    countcc()
                    mi = 0
                    If tottime <> 0 Then mi = 1
                    Do While tottime > 59
                       tottime = tottime - 59
                       mi = mi + 1
                    Loop
                    rou.Text = mi
                    roc.Text = (mi * 2)
                    cchg.Text = (CInt(loc.Text) + CInt(ric.Text) +
CInt(roc.Text))
```

```
                    End If

                    chk = 1
                  End If
                Next
              End If
            Next
          End If
        Next
        callunit = 0
        chk = 0
    End Sub
    Sub calss(ByVal callstss As String)
        dsn.Clear()
        dan = New SqlDataAdapter("select * from " + dum2 + " where
callsts='" + callstss + "'", cnt)
        dan.Fill(dsn, dum2)
        countc()
        DataGridView1.DataSource = dsn.Tables(dum2)
        cunit.Text = CStr(callunit)
    End Sub
    Sub regss(ByVal regsts As String)
        dsn.Clear()
        dan = New SqlDataAdapter("select * from " + dum2 + " where
mobilests='" + regsts + "'", cnt)
        dan.Fill(dsn, dum2)
        countc()
    DataGridView1.DataSource = dsn.Tables(dum2)
    cunit.Text = CStr(callunit)
  End Sub
  Sub calreg(ByVal callstss As String, ByVal regsts As String)
    dsn.Clear()
    countc()
    dan = New SqlDataAdapter("select * from " + dum2 + " where
callsts='" + callstss + "'and mobilests='" + regsts + "'", cnt)
    dan.Fill(dsn, dum2)
    countc()
    DataGridView1.DataSource = dsn.Tables(dum2)
    cunit.Text = CStr(callunit)
  End Sub
  Sub countc()
    For l = 0 To dsn.Tables(dum2).Rows.Count - 1
      tottime = dsn.Tables(dum2).Rows(l).Item(3) + tottime
      If (dsn.Tables(dum2).Rows(l).Item(3)) < 60 Then
        callunit = callunit + 1
      ElseIf (dsn.Tables(dum2).Rows(l).Item(3)) < 120 Then
        callunit = callunit + 2
      ElseIf (dsn.Tables(dum2).Rows(l).Item(3)) < 180 Then
        callunit = callunit + 3
      End If
    Next
  End Sub
  Sub countcc()
    For l = 0 To dsn1.Tables(dum2).Rows.Count - 1
```

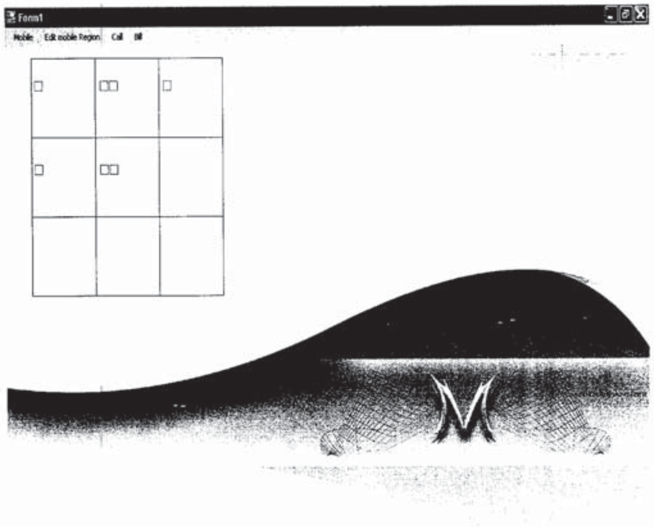
```
      tottime = dsn1.Tables(dum2).Rows(l).Item(3) + tottime
      If (dsn1.Tables(dum2).Rows(l).Item(3)) < 60 Then
        callunit = callunit + 1
      End If
    Next
  End Sub

  Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
    Me.Close()
  End Sub
End Class
```

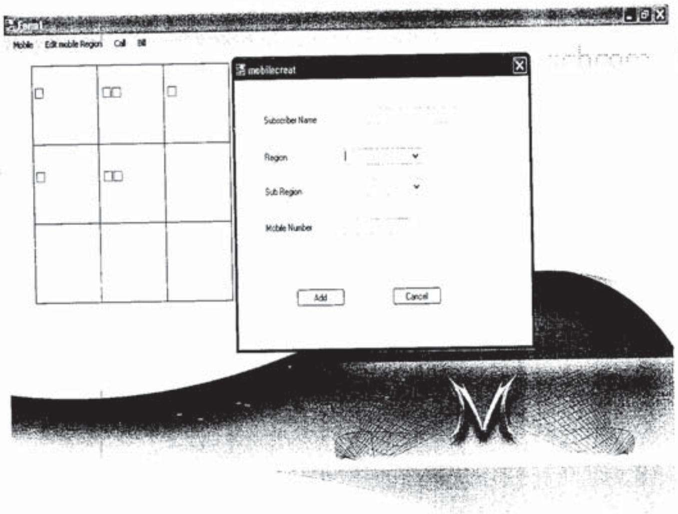
7. APPENDICES:

SCREEN SHOTS:

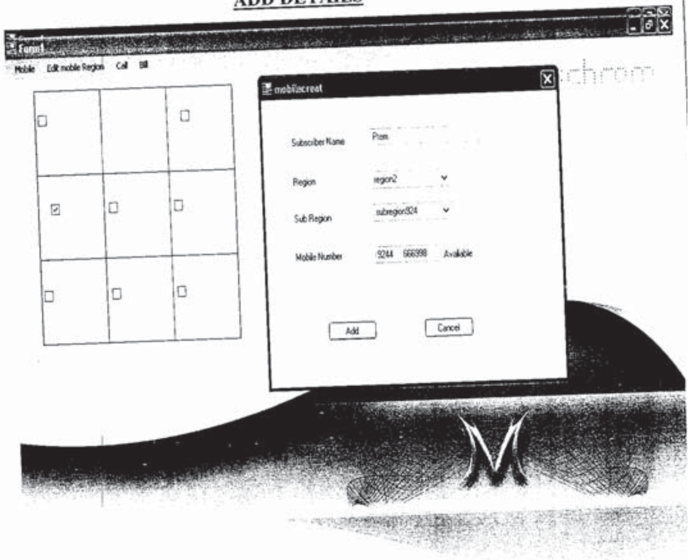
SIMULATION SCREEN



PTN SCHEME



ADD DETAILS



□		□
☑	□	□
□	□	□

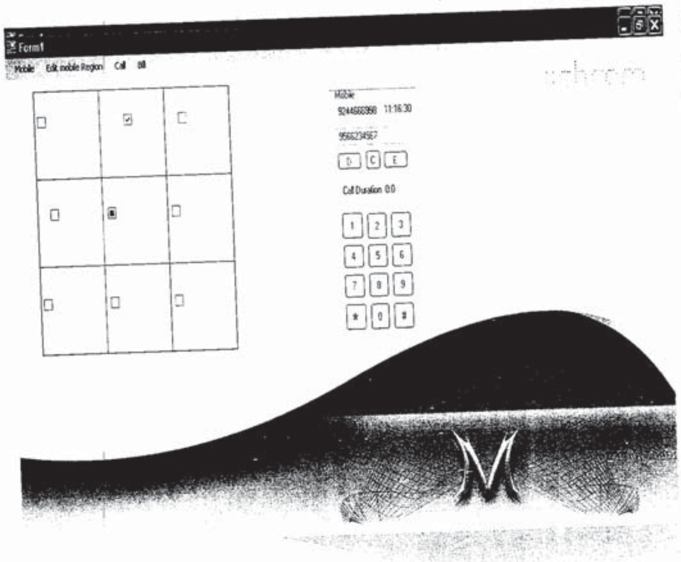
Subscriber Name Prem

Region region2

Sub Region subregion924

Mobile Number 9244 666998 Available

Add Cancel



□	☑	□
□	■	□
□	□	□

Mobile

9244666998 11:16:30

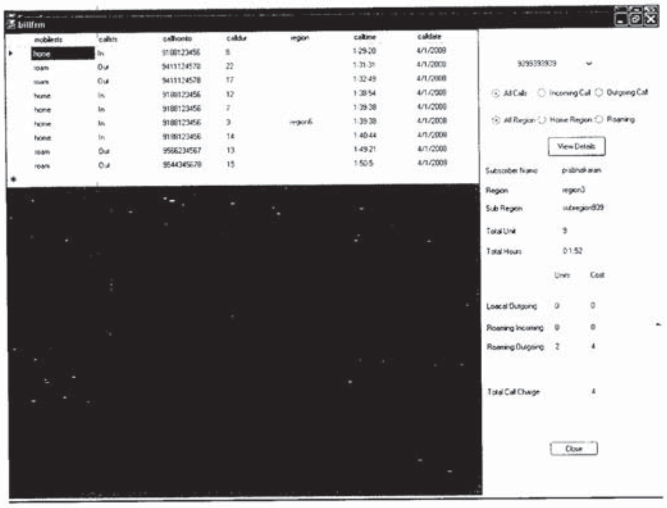
9566234567

D C E

Call Duration 0:0

1	2	3
4	5	6
7	8	9
*	0	#

BILLING CENTRE



mobilests	callsts	callnoto	calldur	region	calltime	calldate
home	In	9188123456	6		1:29:20	4/1/2008
roam	Out	9411124578	22		1:31:31	4/1/2008
roam	Out	9411124578	17		1:32:49	4/1/2008
home	In	9188123456	12		1:38:54	4/1/2008
home	In	9188123456	7		1:39:38	4/1/2008
home	In	9188123456	3	region6	1:39:38	4/1/2008
home	In	9188123456	14		1:40:44	4/1/2008
roam	Out	9566234567	13		1:49:21	4/1/2008
roam	Out	9544345678	15		1:50:5	4/1/2008

9299393939

● All Calls ○ Incoming Call ○ Outgoing Call

● All Region ○ Home Region ○ Roaming

View Details

Subscriber Name prabhakaran

Region region3

Sub Region subregion939

Total Unit 9

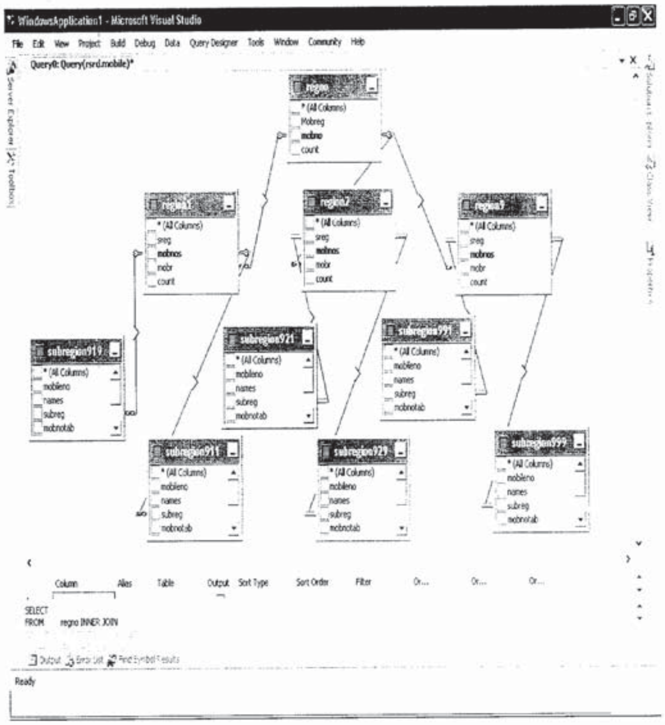
Total Hours 01:52

	Unit	Cost
Loacal Outgoing	0	0
Roaming Incoming	0	0
Roaming Outgoing	2	4

Total Call Charge 4

Close

RELATIONSHIPS:



```
SELECT
FROM     regno INNER JOIN
```

REFERENCES

[1] I. F. Akyildiz, J. Mcnair, J. S. M. Ho, H. Uzunalioglu, and W. Wang. "Mobility management in next-generation wireless systems," *Proc. IEEE.* vol. 87, pp. 1347–1384, Aug. 2006.

[2] I.-R. Chen, T.-M. Chen, and C. Lee, "Agent-based forwarding strategies for reducing location management cost in mobile networks," *ACM/Baltzer J. Mobile Netw. Applicat.*, vol. 6, no. 2, pp. 105–115, 2005.

[3] R. Jain, Y.-B. Lin, C. Lo, and S. Mohan, "A caching strategy to reduce network impacts of PCS," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 1434–1444, Oct. 2002.

[4] R. Jain, S. Rajagopalan, and L. F. Chang, "Phone number portability for PCS systems with ATM backbones using distributed dynamic hashing," *IEEE J. Select. Areas Commun.*, vol. 15, pp. 96–105, Jan. 2003.

[5] L. Kleinrock, *Queueing Systems: Vol. I—Theory*. New York: Wiley, 2000.

[6] T. J. Lehman and M. J. Carey, "A study of index structures for main memory database management systems," in *Proc. 12th Int. Conf. Very Large Data Bases*, Aug. 1999, pp. 294–303.