1.0: INTRODUCTION

1.1: Project Overview

"TICKET RESERVATION SYSTEM" is designed to retrieve the data or information in an easy manner and within a short period. It has an increased capacity of providing tickets in terms of time and cost. It provides the accurate information about the Ticket date, Type of information, pooja details. Devotees make poojas in the temple for which they need to make arrangement in the temple like advance booking, advance payment for the pooja. Devotees also book their tickets, pooja timings, pooja dates so the temple authorities make the records. When devotees make donation for the temple, even the donation will be recorded in the books or receipts. Which may consume lots of time, labouress, wages and even stationary expenses. It is a menu-driven interactive package suited for any kind of temple and this project is implemented for 'Tirumala Tirupati Devasthanam'.

This software has been proposed to maintain and use the limited manpower with maximum efficiently and accuracy.

2.0: Literature survey

2.1: Introduction

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next steps is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

We have to analysis the Parallel and distributed systems Survey:

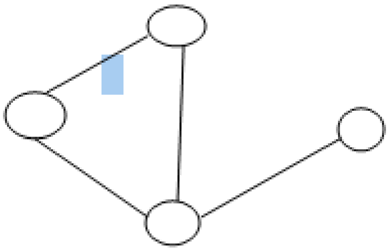
2.2: Parallel and Distributed Computing

1.“A distributed system is a collection of independent computers that appear to the users of the system as a single computer.”
2.“A distributed system consists of a collection of autonomous computers linked to a computer network and equipped with distributed system software.”
3.“A distributed system is a collection of processors that do not to share memory or a clock.”
4.“Distributed system are a term used to define a wide range of computer systems from a weakly-coupled system such as wide area networks, to very strongly coupled systems such as multiprocessor systems.”

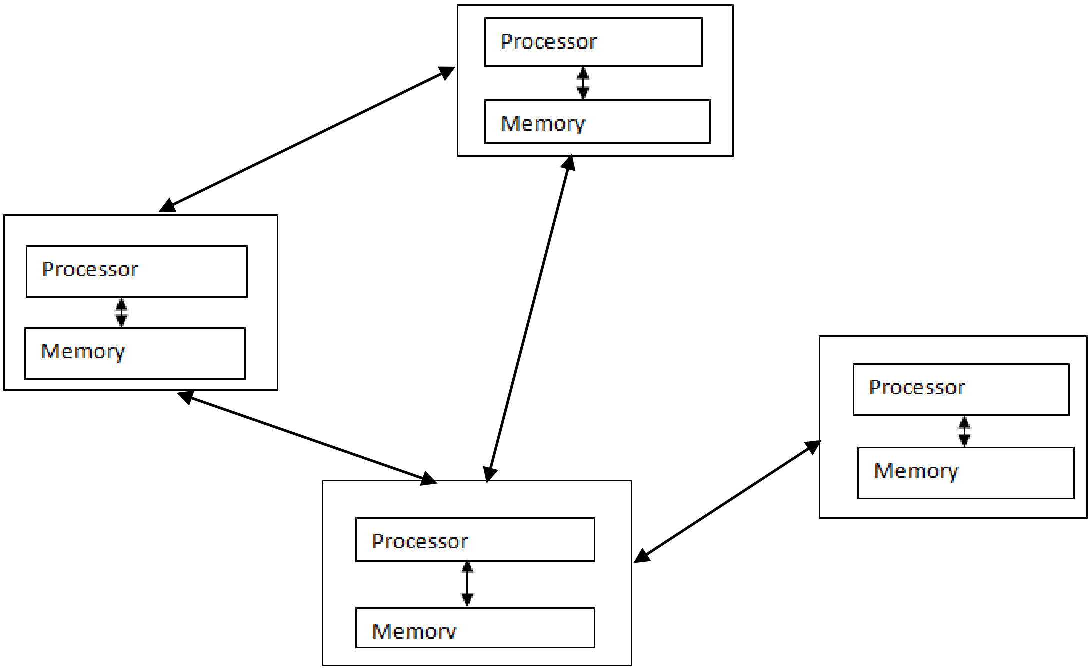
Distributed systems are groups of networked computers, which have the same goal for their work. The terms “concurrent computing”, “parallel computing”, and “distributed computing” have a lot of overlap, and no clear distinction exists between them. The same system may be characterized both as “parallel” and “distributed”; the processors in a typical distributed system run concurrently in parallel. Parallel computing may be seen as a particular tightly-coupled form of distributed computing, and distributed computing may be seen as a loosely-coupled form of parallel computing. Nevertheless, it is possible to roughly classify concurrent systems as “parallel” or “distributed” using the following criteria:

1. In parallel computing , all processors have access to a shared memory. Shared memory can be used to exchange information between processors.
2. In distributed computing, each processor has its own private memory(distributed memory).Information is exchanged by passing messages between the processors.

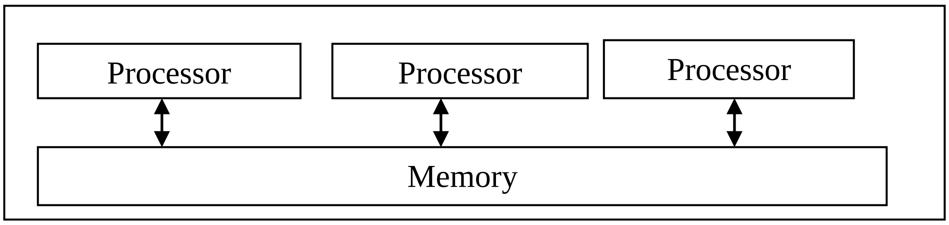
(a)



(b)



(c)



(a)-(b) A distributed system

(c) A parallel system

The figure on the previous page illustrate the difference between distributed and parallel systems. Figure(a) is a schematic view of a typical distributed system; as usual, the system is represented as a network topology in which each node is a computer and line connecting the nodes is a communication link, Figure(b) shows the same distributed system in more

Detail each computer has its own local memory, and information can be exchanged only by passing messages from one node to another by using the available communication links. Figure (c) shows a parallel system in which each processor has a direct access to a shared memory.

The situation is further complicated by the traditional uses of the terms parallel and distributed algorithm that do not quite match the above definitions of parallel and distributed systems; see the section Theoretical foundations below for more detailed discussion. Nevertheless, as a rule of thumb, high-performance parallel computation in a shared-memory multiprocessor uses parallel algorithms while the coordination of a large-scale distributed system uses distributed algorithms.

2.3: Existing system

The study of existing system has thrown a light on various activities involved in a many temples for the ticket booking system finding all requirements needed. The existing system is manual process.

In the reservation all the entries are done manually. Each and every form is filled by applicant and this is processed by the concerned temple . He has to verify all the documents and the report must be send immediately. For this work, they have to maintain lot of entry books. To maintain that record is very important. If any lose there will be reach towards big problems. Large amount of storage cost is required.

Since, it is a manual process there is some possibilities for making wrong entries.

Demerits:

1. No backup facility.
2. Conflicting user environment.
3. Unable to provide instant reports required by the office people.
4. Insecure file handling since paper oriented.
5. Getting up to date information is a difficult task.
6. It needs lot of man power.

2.4: Proposed system

The “TICKET RESERVATION SYSTEM” is used in the way to make drop the drawbacks of existing system. The main objective is to create it on the way to reduce the Separate Group work. So for can be free from most mental tensions. The system proposed will work effectively so that it is working online and it is updated within 24 hours and made user work easy and to lessen the transportation cost.

3.0: FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are:

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

3.1:Technical Feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal .or null changes are required for implementing this system.

3.2:Operational Feasibility

The aspect of study is to check the level of acceptance of the system by the system by the user. This includes the process of training of the user the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so the he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3:Economical Feasibility

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.0:SOFTWARE REQUIREMENT SPECIFICATION

4.1: Hardware Specification

HARDWARE REQUIREMENTS:

Processor	: Intel pentium 4 and later
RAM	: 1 GB
Hard disk	: 40GB
Floppy Drive	: 4.70MB

4.2 Software Specification

SOFTWARE REQUIREMENTS:

Operating system	: Windows8
Front-end	: Visual basic2010
Back-end	: Ms-access
Documentation	: Microsoft Word 2007

4.3: Scope of the project

1. This package is developed to speed up the current manual system. The system maintains all the details of various temples.

2. It stores details about the providing tickets in a easier manner with the help of the proposal system we can retrieve necessary data more easily, accurately and speedily.

3.It is designed to retrieve the data or information in an easy manner and within a short period. It has an increased capacity of providing tickets in terms of time and cost.

4.4: Functional Requirement

Statements of services the system should react to particular inputs and how the system should behave in particular situations.

1. Admin can see the list of poojas and record the details of devotees.
2. And also record the details of donation.
3. Insert the new poojas, delete poojas, update poojas.

4.5:Nonfunctional Requirement

Constraints on the services or functions offered by the system such as timing constraints on the development process, standards, etc. This project provides an option for detecting false data.

1.These define system properties and constraints e.g. reliability, response time and storage requirements. Constraints are I/O device capability, system representations, etc.

2. Process requirement may also be specified mandating a particular CASE system, programming language or development method.

3. Non-functional requirements may be more critical than functional requirements. If these are not met, the system is useless.

5.0: SELECTED SOFTWARE

5.1: Features of .Net:

Microsoft Net is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows based applications, and Web solutions. The .Net Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .Net: there are numerous language available to the developer including Managed C++, C#', Visual Basic and Java Script. The .Net framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

".NET" is also the collective name given to various software components built upon the .Net platform. These will be both products (Visual Studio .NET and Windows .NET Server, for instance) and services (like Passport, .NET My Services, and so on).

5.2: Introduction to .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A Hierarchical set of class libraries.

The CLR is described as the "Execution Engine" of .NET. It provides the environment within which programs run. The most important features are:

- Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- Memory management, notably including garbage collection.
- Checking and enforcing security restrictions on the running code.
- Loading and executing programs, with version control and other such features.
- The following features of the .NET framework are also worth description:

5.3: Managed Code

The code that targets .NET, and which contains certain extra information –"metadata" – to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

5.4: Managed data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic .NET and Jscript .NET, whereas others, namely C++, do not. Targeting CLR can, depending on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications – data that doesn't get garbage collected but instead is after by unmanaged code.

5.5: **Common Type System**

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

5.6: Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

5.7: The Class Library

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as Objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object type if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

Other language for which.NET compilers are available include

- **FORTRAN**
- **COBOL**
- **Eiffel**

Fig .Net framework

<table>
<tr><td>ASP.NET
XML WEB SERVICES</td><td>Windows Forms</td></tr>
<tr><td colspan="2">Base Class Libraries</td></tr>
<tr><td colspan="2">Common Language Runtime</td></tr>
<tr><td colspan="2">Operating System</td></tr>
</table>

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR(Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services.

C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, we can use objects, classes, and components created in other CLS –compliant languages in C#.NET. The use of

CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

5.8: Languages Supported By .NET

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), But there are also a number of new additions to the family.

Visual Basic.NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributed and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS –compliant language can use the classes, objects, and components you create in Visual Basic. NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new.NET Framework.

C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J#.NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of other programming languages.

Active State has created Visual Perl and Visual Python, which enable.NET –aware applications to be built in either Perl or Python. Both products can be integrated to the Visual Studio.NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

5.9: Constructor and DE constructor

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

5.10: Garbage collection

Garbage collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use.
In C# .NET, the garbage collector checks for the objects that are nor correctly in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

5.11: Overloading

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, we can use it for constructors and properties in a class.

5.12: Multithreading

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, we can use multithreading to decrease the time taken by an application to respond to user interaction.

5.13: Structured Exception Handling

C# .NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, we need to use Try… finally statements to create exception handlers. Using Try… Catch…Finally statements; we can create robust and effective exception handlers to improve the performance of our application.

5.14: The .NET FRAMEWORK

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet.

Objectives of the .NET FRAMEWORK

1. To provide a consistent object-oriented programming environment whether object codes is stored and executed locally on Internet-distributed, or executed remotely.
2. To provide a code –execution environment to minimize software deployment and guarantees safe execution of code.
3. Eliminates the performance problems.

There are different types of application, such as Windows-based applications and Web-based applications.

5.15: SQL- SERVER

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component now use the term Meta Data Services. The term repository engine within Meta Data Services.

SQL-SERVER database consist of six type of objects, they are

1. TABLE
2. QUARY
3. FORM
4. REPORT
5. MACROR

TABLE:

A database is a collection of data about a specific topic.

VIEWS OF TABLE:

We can work with a table in two types,

1. Design view
2. Datasheet view

Design View

To build or modify the structure of a table we work in the table design view. We can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself we work in tables datasheet view mode.

QUERY:

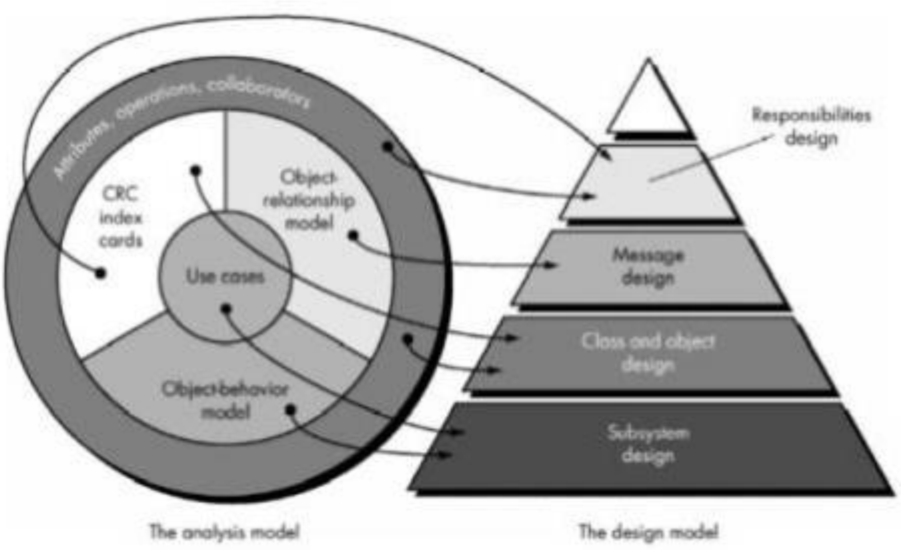
A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynast (if you edit it) or a snapshot (it cannot be edited). Each time we run query, we get latest information in the dynast. Access either displayed the dynast or snapshot for us to view or perform an action on it, such as deleting or updating.

6.0: SYSTEM ANALYSIS

6.1: Introduction

Is the study of sets of interacting entities, including computer systems analysis this field is closely related to requirements analysis or operations research. It is also “an explicit formal inquiry carried out to help someone(referred to as the decision maker) identify a better course of action and make a better decision than he might otherwise have made.

6.2:Analysis Model



6.3:Study of the system

The package entitled "**TICKET RESERVATION SYSTEM**" deals with the computerization of Ticket Booking for both Seva and Rooms. The study of existing manual system paved the way for finding all requirements needed and gave idea of improving the processing activity. The problems in the existing system are noted and eluded from the processing activity.

6.4: Proposed system

The "TICKET RESERVATION SYSTEM" is used in the way to make drop the drawbacks of existing system. The main objective is to create it on the way to reduce the Separate Group work. So for can be free from most mental tensions. The system proposed will work effectively so that it is working online and it is updated within 24 hours and made user work easy and to lessen the transportation cost.

6.5:System Modules

Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus it can be considered to be the most critical stage in giving the user, confidence that the new system will work and be effective.

The implementation stage involves careful planning investigation of the existing System and it's constraints on implementation, designing of methods to achieve changeover and evaluation of changeover methods.

Modules the project

- Login form
- Pooja list form
- Pooja detail form
- Donation form
- Receipt

6.6: Input Design

The input design is the link that types the information system to the world of its users. It consists of developing specification and procedures for data preparation and to input in to the computer for processing. Simply we can say input design is the process of converting them user-generated inputs computer based format.

The important features are as follows:

♦ The input screen is not over crowded, as the user can understand a piece of Information from the screen. Instead more are provided to understand the data entry process.

♦ The input validation is being done at program level to check error and help message are to be provided to effectively enter data into the data entry screen in the software.

♦ As human is prone for errors and mistakes confirmation for critical data entries are being provided.

6.7:Output Design

The entire output screens are informative and interactive in such a way that the user can fulfill his requirements through queries. In this system the main output screens the main output screens are to up-to-date Reports and listing of required information.

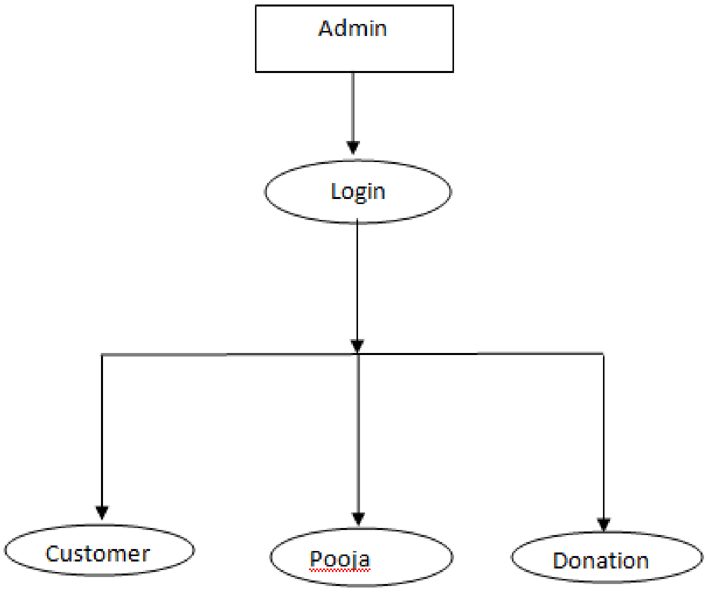
System design is process of planning of a new system that is a complement of the old and replace the existing system with new system. The new design is a solution and the translation of the requirements of the old by the ways of meeting them.

The various inputs, the objectives and the output requirements of the system were identified.

7.0: SYSTEM DESIGN

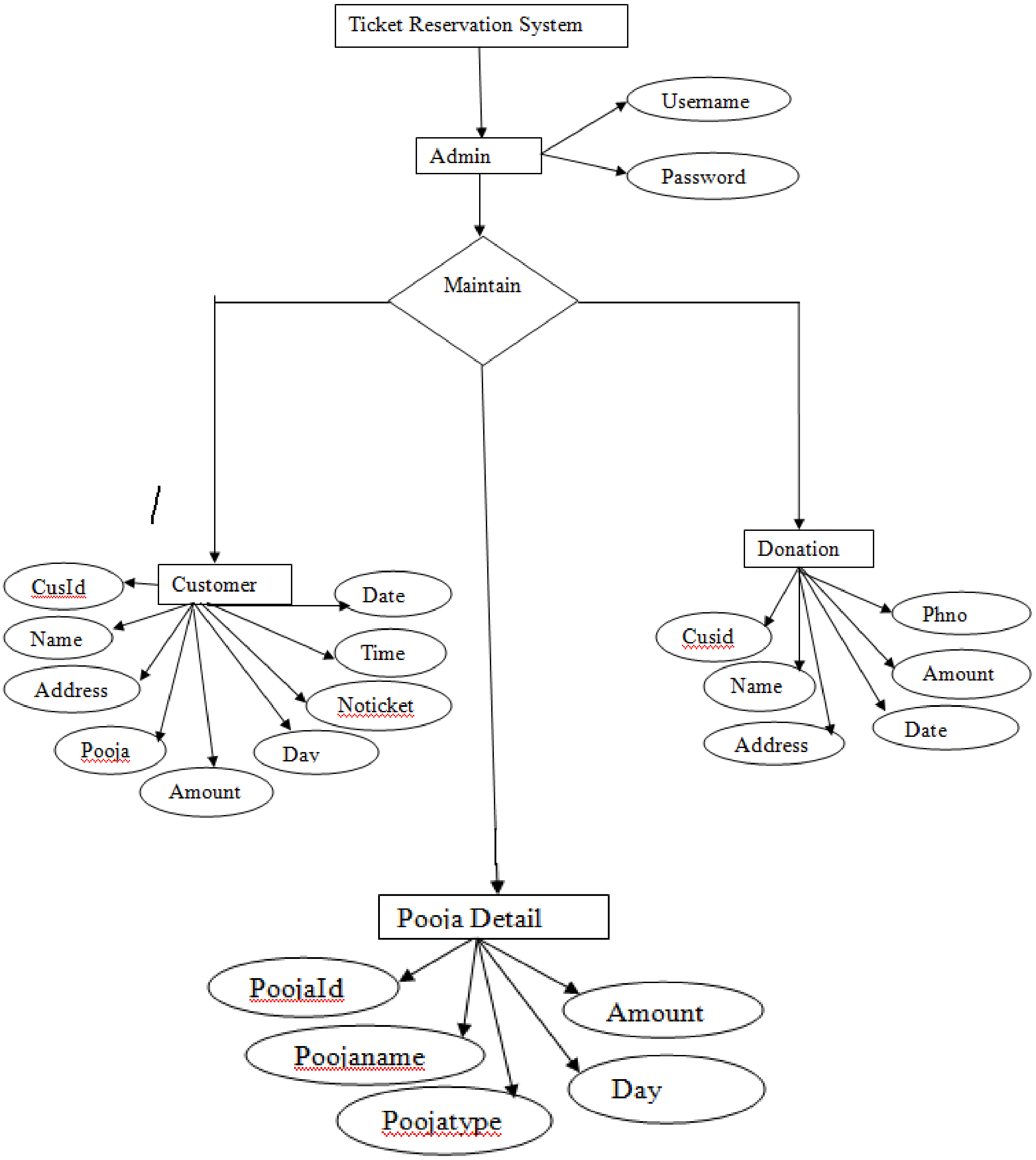
7.1: Data Flow Diagram

A data flow diagram is a graphical representation of the flow of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system.



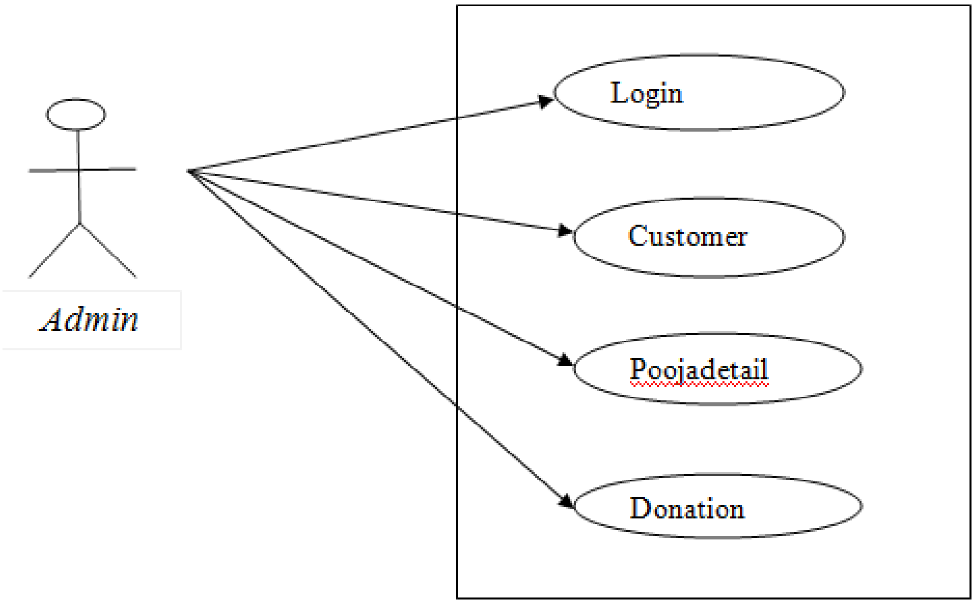
7.2: Entity Relationship Diagram

An ERD is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within that system.



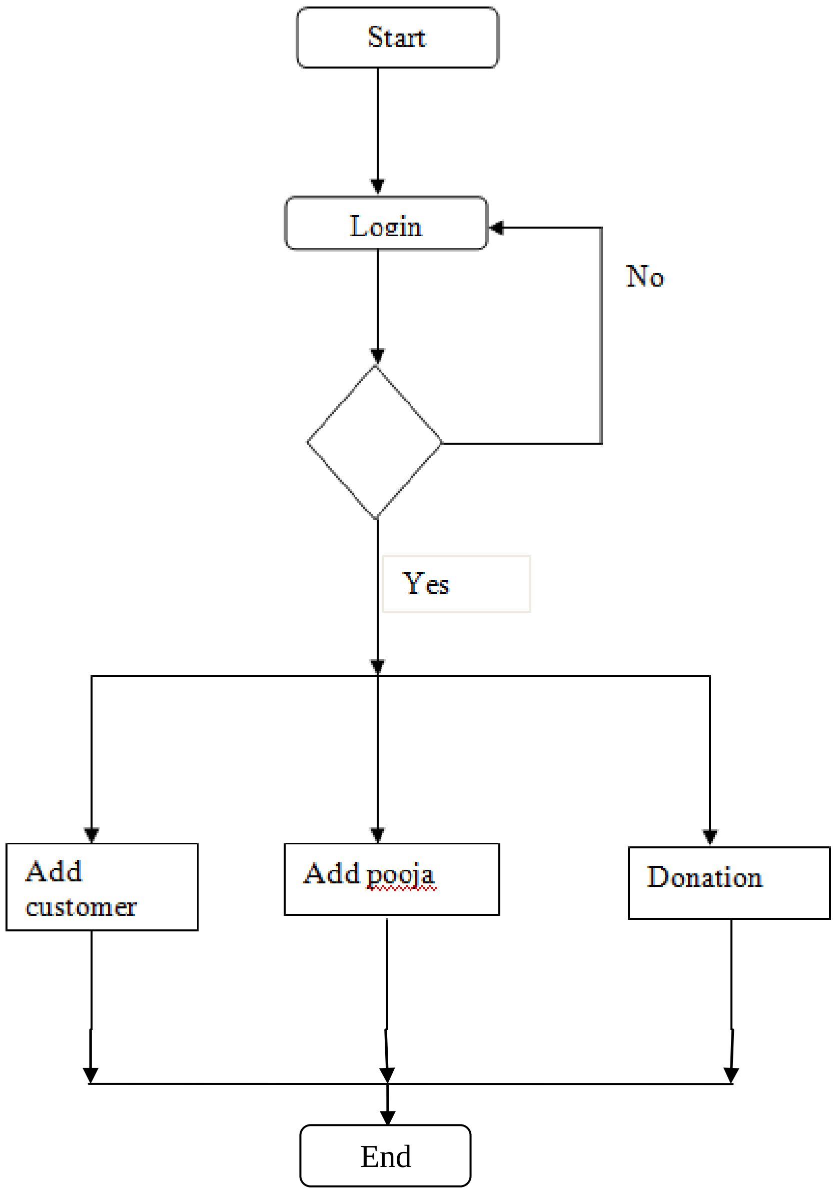
7.3: Use Case Diagram

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.



7.4: Flow Chart

A flowchart is a type of diagram that represents an algorithm, workflow or process, showing the steps as boxes of various kinds, and their order by connecting them with arrows



8.0:IMPLEMENTATION

Implementation is the status of project when theoretical design turned into a working system.

Login Form:

```
Imports System.Data.OleDb
Public Class Form1
    Dim cn As New OleDbConnection

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        If TextBox1.Text = "" And TextBox2.Text = "" Then
            MsgBox("please enter Username and Password.", MsgBoxStyle.Information,
"login")
        Else
            Try
                Dim p1 As New OleDbParameter
                p1.ParameterName = "p1"
                p1.Value = TextBox1.Text
                Dim cmd As New OleDbCommand
                cmd.CommandText = "select * from login where username=@p1"
                cmd.Connection = cn
                cmd.Parameters.Add(p1)
                Dim dr As OleDbDataReader
                dr = cmd.ExecuteReader
                Dim str As String = dr.Read
                Dim epwd As String = dr(1)
                Dim pwd As String = TextBox2.Text
                If (str) Then
                    Dim f As Integer = StrComp(pwd, epwd)
                    If f <> 0 Then
                        MsgBox("Invalid user")
                    Else
                        MsgBox("login successfully")
                        Form11.MenuStrip1.Enabled = True
                        Me.Close()
                    End If


                End If
            Catch
                MsgBox("Error")
            End Try
        End If

    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        cn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\HP\Documents\Database2.accdb"
        cn.Open()
    End Sub
End Class
```

Pooja list:

```
Public Class Form4

    Private Sub Form4_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        ListBox1.Items.Add("Daily pujas")
        ListBox1.Items.Add("Thomala seva")
        ListBox1.Items.Add("Unjal seva")
        ListBox1.Items.Add("Sahasra Deepaalankarana seva")
        ListBox1.Items.Add("Ekanta seva")
        ListBox2.Items.Add("Weekly sevas")
        ListBox2.Items.Add("Visesha pooja(monday)")
        ListBox2.Items.Add("Ashtadala pada padmaaradhana seva(Tuesday)")
        ListBox2.Items.Add("Sahasra kalasabhishekam seva(wednesday)")
        ListBox2.Items.Add("Tiruppavada  seva(thursday)")
        ListBox2.Items.Add("Poolangi seva(friday)")
        ListBox2.Items.Add("Abhishekam(saturday and sunday)")
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Me.Hide()
        Form5.Show()
    End Sub
End Class
```

Customer detail:

```
Imports System.Data.OleDb
Public Class Form5
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myConnection As OleDbConnection = New OleDbConnection


    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        Dim p1 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = TextBox8.Text
        Dim cmd As New OleDbCommand
        cmd.CommandText = "delete from customer where cus_id=@p1"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.ExecuteNonQuery()
        MsgBox("Deleted successfully")

    End Sub

    Private Sub Form5_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        provider = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
        datafile = "C:\Users\HP\Documents\Database2.accdb"
        connstring = provider & datafile
        myConnection.ConnectionString = connstring
        myConnection.Open()

        Dim cmd As New OleDbCommand
        cmd.CommandText = "select puja_name from puja_mst"
        cmd.Connection = myConnection

        Dim dr As OleDbDataReader
        dr = cmd.ExecuteReader
        While dr.Read
            ComboBox1.Items.Add(dr(0))
        End While
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim p1, p2, p3, p4, p5, p6, p7, p8, p9 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = TextBox8.Text
        p2.ParameterName = "p2"
        p2.Value = TextBox1.Text
        p3.ParameterName = "p3"
        p3.Value = TextBox2.Text
        p4.ParameterName = "p4"
        p4.Value = ComboBox1.Text
        p5.ParameterName = "p5"
        p5.Value = TextBox7.Text
        p6.ParameterName = "p6"
        p6.Value = TextBox6.Text
        p7.ParameterName = "p7"
        p7.Value = TextBox4.Text
        p8.ParameterName = "p8"
        p8.Value = DateTimePicker1.Text
        p9.ParameterName = "p9"
        p9.Value = TextBox5.Text
        Dim cmd As New OleDbCommand
        cmd.CommandText = "insert into customer
values(@p1,@p2,@p3,@p4,@p5,@p6,@p7,@p8,@p9)"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.Parameters.Add(p2)
        cmd.Parameters.Add(p3)
        cmd.Parameters.Add(p4)
        cmd.Parameters.Add(p5)
        cmd.Parameters.Add(p6)
        cmd.Parameters.Add(p7)
        cmd.Parameters.Add(p8)
        cmd.Parameters.Add(p9)
        cmd.ExecuteNonQuery()
        MsgBox("Record added successfully")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Me.Hide()
        Form6.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Me.Hide()
        End
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        Dim p4 As New OleDbParameter
        p4.ParameterName = "p4"
        p4.Value = ComboBox1.SelectedItem
        Dim cmd As New OleDbCommand
        cmd.CommandText = "select  amount,day from puja_mst where puja_name=@p4"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p4)
        Dim dr As OleDbDataReader
        dr = cmd.ExecuteReader
        While dr.Read
            TextBox3.Text = dr(0)
            TextBox6.Text = dr(1)
        End While
        dr.Close()
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        TextBox7.Text = Val(TextBox3.Text) * Val(TextBox4.Text)
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
        TextBox8.Clear()
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox6.Clear()
        TextBox4.Clear()
        TextBox7.Clear()
        TextBox5.Clear()

    End Sub

    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
        Form9.Show()
    End Sub
End Class
```

Pooja:

```
Imports System.Data.OleDb
Public Class Form6
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myConnection As OleDbConnection = New OleDbConnection


    Private Sub Form6_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        provider = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
        datafile = "C:\Users\HP\Documents\Database2.accdb"
        connstring = provider & datafile
        myConnection.ConnectionString = connstring
        myConnection.Open()

        Dim cmd As New OleDbCommand
        cmd.CommandText = "select puja_name from puja_mst"
        cmd.Connection = myConnection

        Dim dr As OleDbDataReader
        dr = cmd.ExecuteReader
        While dr.Read
            ComboBox1.Items.Add(dr(0))
        End While
        dr.Close()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim p1, p2, p3, p4, p5 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = txtid.Text()
        p2.ParameterName = "p2"
        p2.Value = txtpnm.Text()
        p3.ParameterName = "p3"
        p3.Value = txtptyp.Text()
        p4.ParameterName = "p4"
        p4.Value = txtday.Text()
        p5.ParameterName = "p5"
        p5.Value = txtamt.Text()
        Dim cmd As New OleDbCommand
        cmd.CommandText = "insert into puja_mst values(@p1,@p2,@p3,@p4,@p5)"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.Parameters.Add(p2)
        cmd.Parameters.Add(p3)
        cmd.Parameters.Add(p4)
        cmd.Parameters.Add(p5)
        cmd.ExecuteNonQuery()
        MsgBox("Record added successfully")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim p1 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = txtid.Text
        Dim cmd As New OleDbCommand
        cmd.CommandText = "delete from puja_mst where puja_id=@p1"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.ExecuteNonQuery()
        MsgBox("Deleted successfully")
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        txtid.Clear()
        txtpnm.Clear()
        txtptyp.Clear()
        txtday.Clear()
        txtamt.Clear()
    End Sub

    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ComboBox1.SelectedIndexChanged
        Dim p2 As New OleDbParameter
        p2.ParameterName = "p2"
        p2.Value = ComboBox1.SelectedItem
        Dim cmd As New OleDbCommand
        cmd.CommandText = "select * from puja_mst where puja_name=@p2"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p2)
        Dim dr As OleDbDataReader
        dr = cmd.ExecuteReader
        While dr.Read
            txtid.Text = dr(0)
            txtpnm.Text = dr(1)
            txtptyp.Text = dr(2)
            txtday.Text = dr(3)
            txtamt.Text = dr(4)
        End While
        dr.Close()
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Me.Hide()
        Form7.Show()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click

        txtptyp.Text = txtptyp.Text
        txtday.Text = txtday.Text
        txtamt.Text = txtamt.Text
        Dim cmd As New OleDbCommand

        cmd.CommandText = "update  puja_mst set  puja_type='" & txtptyp.Text & "'
where puja_name='" & txtpnm.Text & "'"

        cmd.CommandText = "update  puja_mst set  day='" & txtday.Text & "' where
puja_name='" & txtpnm.Text & "'"

        cmd.CommandText = "update  puja_mst set  amount='" & txtamt.Text & "' where
puja_name='" & txtpnm.Text & "'"
        cmd.Connection = myConnection
        cmd.ExecuteNonQuery()
        MsgBox("updated successfully")
    End Sub
End Class
```

Donation:

```
Imports System.Data.OleDb
Public Class Form7
    Dim provider As String
    Dim datafile As String
    Dim connstring As String
    Dim myConnection As OleDbConnection = New OleDbConnection

    Private Sub Form7_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        provider = "Provider=Microsoft.ACE.OLEDB.12.0;Data Source="
        datafile = "C:\Users\HP\Documents\Database2.accdb"
        connstring = provider & datafile
        myConnection.ConnectionString = connstring
        myConnection.Open()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Dim p1, p2, p3, p4, p5, p6 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = TextBox1.Text
        p2.ParameterName = "p2"
        p2.Value = TextBox2.Text
        p3.ParameterName = "p3"
        p3.Value = TextBox3.Text
        p4.ParameterName = "p4"
        p4.Value = DateTimePicker1.Text
        p5.ParameterName = "p5"
        p5.Value = TextBox5.Text
        p6.ParameterName = "p6"
        p6.Value = TextBox4.Text

        Dim cmd As New OleDbCommand
        cmd.CommandText = "insert into hundi values(@p1,@p2,@p3,@p4,@p5,@p6)"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.Parameters.Add(p2)
        cmd.Parameters.Add(p3)
        cmd.Parameters.Add(p4)
        cmd.Parameters.Add(p5)
        cmd.Parameters.Add(p6)
        cmd.ExecuteNonQuery()
        MsgBox("Record added successfully")
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Dim p1 As New OleDbParameter
        p1.ParameterName = "p1"
        p1.Value = TextBox1.Text
        Dim cmd As New OleDbCommand
        cmd.CommandText = "delete from hundi where cus_id=@p1"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p1)
        cmd.ExecuteNonQuery()
        MsgBox("Deleted successfully")

    End Sub

    Private Sub TextBox2_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox2.TextChanged
    End Sub

    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
        TextBox1.Clear()
        TextBox2.Clear()
        TextBox3.Clear()
        TextBox5.Clear()
        TextBox4.Clear()
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
        Me.Hide()
        End
    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
        Dim p2 As New OleDbParameter
        p2.ParameterName = "p2"
        p2.Value = TextBox2.Text
        Dim cmd As New OleDbCommand
        cmd.CommandText = "select*from hundi where cus_name=@p2"
        cmd.Connection = myConnection
        cmd.Parameters.Add(p2)
        Dim dr As OleDbDataReader
        dr = cmd.ExecuteReader
        While dr.Read
            TextBox1.Text = dr(0)
            TextBox3.Text = dr(2)
            DateTimePicker1.Text = dr(3)
            TextBox5.Text = dr(4)
            TextBox4.Text = dr(5)
        End While
        dr.Close()
    End Sub

    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
        Form8.Show()
    End Sub
End Class
```

Receipt:

```
Imports CrystalDecisions.CrystalReports.Engine
Imports System.Data.OleDb

Public Class Form8

    Dim cn As New OleDbConnection
    Dim rdoc As New ReportDocument

    Private Sub Form8_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        cn = New OleDbConnection
        cn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\HP\Documents\Database2.accdb"
        cn.Open()
        Dim i As Integer = Integer.Parse(Form7.TextBox1.Text)
        Dim da As New OleDbDataAdapter("select * from hundi where cus_id=" & i, cn)
        Dim ds As New DataSet
        da.Fill(ds, "hundi")
        rdoc.Load("..\..\CrystalReport5.rpt")
        rdoc.SetDataSource(ds)
        CrystalReportViewer1.ReportSource = rdoc
        cn.Close()
    End Sub
End Class
```

```
Imports CrystalDecisions.CrystalReports.Engine
Imports System.Data.OleDb

Public Class Form9
    Dim cn As New OleDbConnection
    Dim rdoc As New ReportDocument
    Private Sub Form9_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        cn = New OleDbConnection
        cn.ConnectionString = "Provider=Microsoft.ACE.OLEDB.12.0;Data
Source=C:\Users\HP\Documents\Database2.accdb"
        cn.Open()
        Dim i As Integer = Integer.Parse(Form5.TextBox8.Text)
        Dim da As New OleDbDataAdapter("select * from customer where cus_id=" & i, cn)
        Dim ds As New DataSet
        da.Fill(ds, "customer")
        rdoc.Load("..\..\CrystalReport4.rpt")
        rdoc.SetDataSource(ds)
        CrystalReportViewer1.ReportSource = rdoc
        cn.Close()

    End Sub

    Private Sub CrystalReportViewer1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CrystalReportViewer1.Load

    End Sub
End Class
```

MDI form:

```
Public Class Form11

    Private Sub Form11_Load(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles MyBase.Load
        MenuStrip1.Enabled = False
        Form1.ShowDialog()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles ExitToolStripMenuItem.Click
        End
    End Sub

    Private Sub PujaToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles PujaToolStripMenuItem.Click
        Form4.ShowDialog()
    End Sub

    Private Sub CustomerToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e 
As System.EventArgs) Handles CustomerToolStripMenuItem.Click
        Form6.ShowDialog()
    End Sub

    Private Sub DonationToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e 
As System.EventArgs) Handles DonationToolStripMenuItem.Click
        Form7.ShowDialog()
    End Sub

    Private Sub BillToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles BillToolStripMenuItem.Click
        Form8.ShowDialog()
        Form9.ShowDialog()
    End Sub

    Private Sub UserToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As 
System.EventArgs) Handles UserToolStripMenuItem.Click
        Form5.ShowDialog()
    End Sub
End Class
```

9.0: SYSTEM TESTING

9.1: Introduction

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product it is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

9.2: Unit Testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

9.3: Integration Testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

9.4: Functional Test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input :Identified classes of valid input must be accepted.

Invalid Input :Identified classes of invalid input must be rejected.

Functions :Identified functions must be exercised.

Output : Identified classes of application outputs must be exercised.

System/Procedures : Interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Businesses process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and effective value of current tests is determined.

9.5: System test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

9.6: Unit testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

9.7: Strategy and Approach

Field testing will be performed manually and functional tests will be written in detail.
Test objectives

- All field entries must work properly.
- Page must be activated from the indentified link.
- The entry screen, messages and responses must not be delayed.

Features to the tested

- Verify that the entries are of the correct format.
- No duplicate entries should be allowed.
- All links should entries take the user to the correct page.

9.8: Integration Testing

Software integration testing is the incremental int5egration testing of two or more integrated software component on a single platform to produce failures caused by interface defects.

The task of the integration testing test is to check that components or software applications, e.g. components in a software system or-one step up-software applications at the company level-interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.9: Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

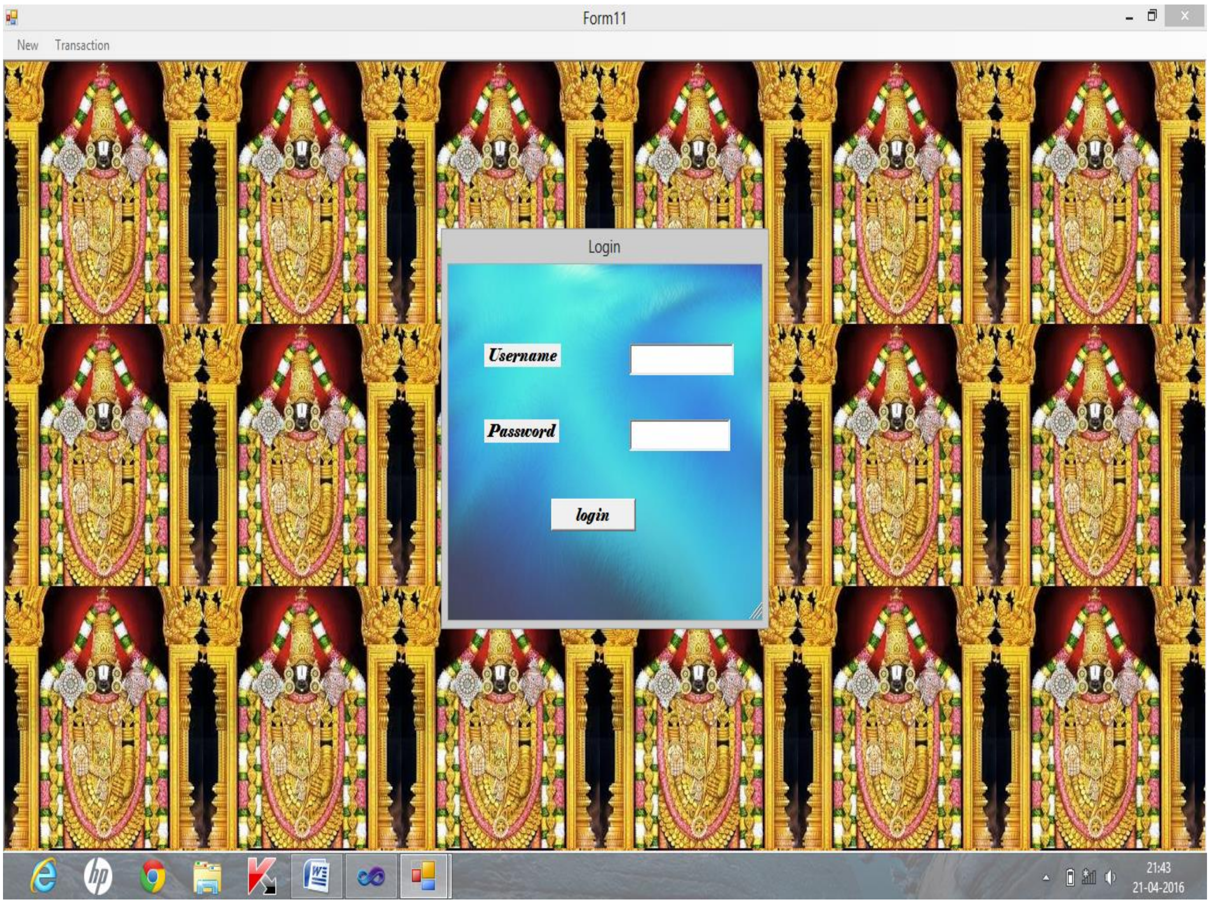
Test Results: All the test cases mentioned above passed successfully. No defects encountered.

9.10: TEST CASES

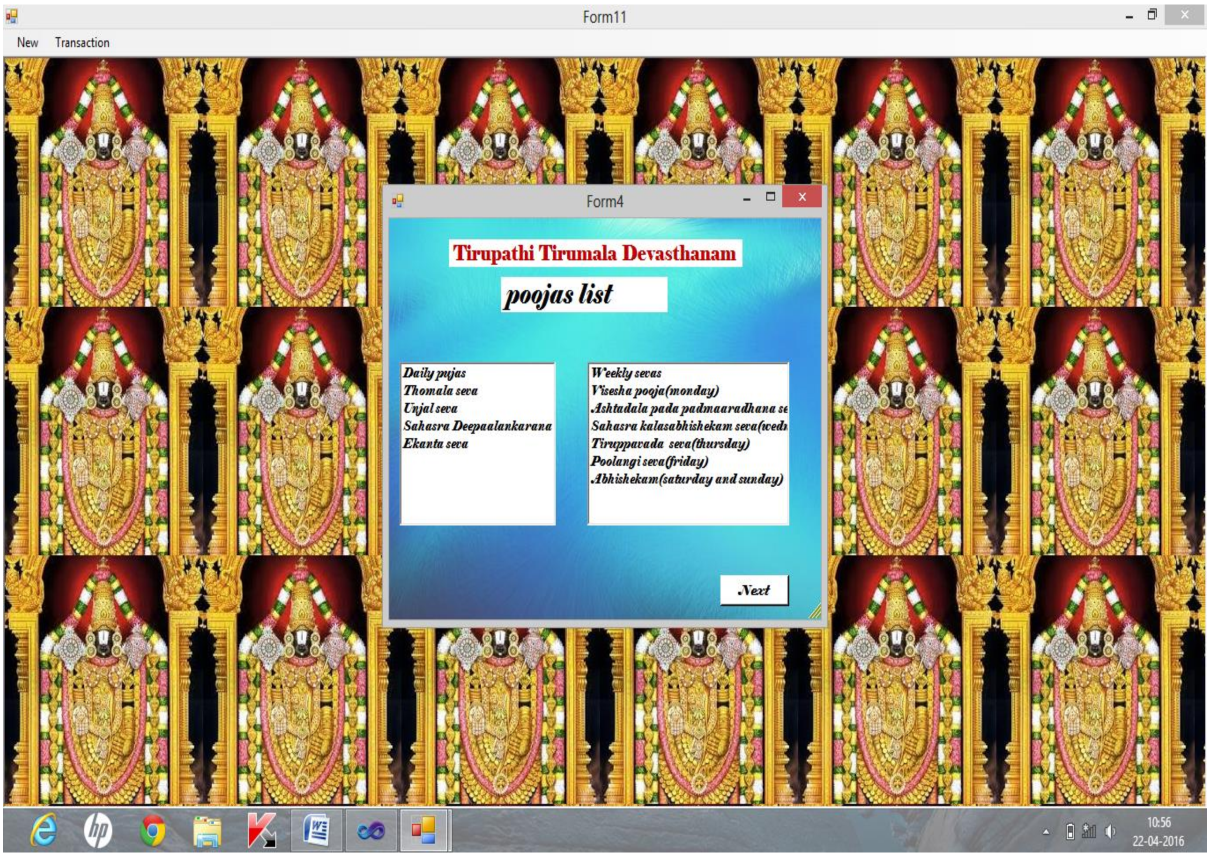
Test id	Test case	Expected Result	Actual Result	Final Result
1	Login	Should display main page	Displaying the main page	Pass
2	Pooja list	Should display pooja list	Displaying the pooja list form	Pass
3	Insert the detail of Customer	Inserted data will be updated in particular table	Data will be inserted	Pass
4	Update and Insert the pooj	Inserted and updated data will be in particular table	data will be Updated	Pass
5	Donation	Should display the inserted data	Displaying inserted data	Pass
6	Receipt	Should display the receipt of particular customer	Displaying the particular customer receipt	Pass
7	Logout	Should be exit	Exit from the page	Pass

10.0: SNAPSHOT

Login:



Pooja list:



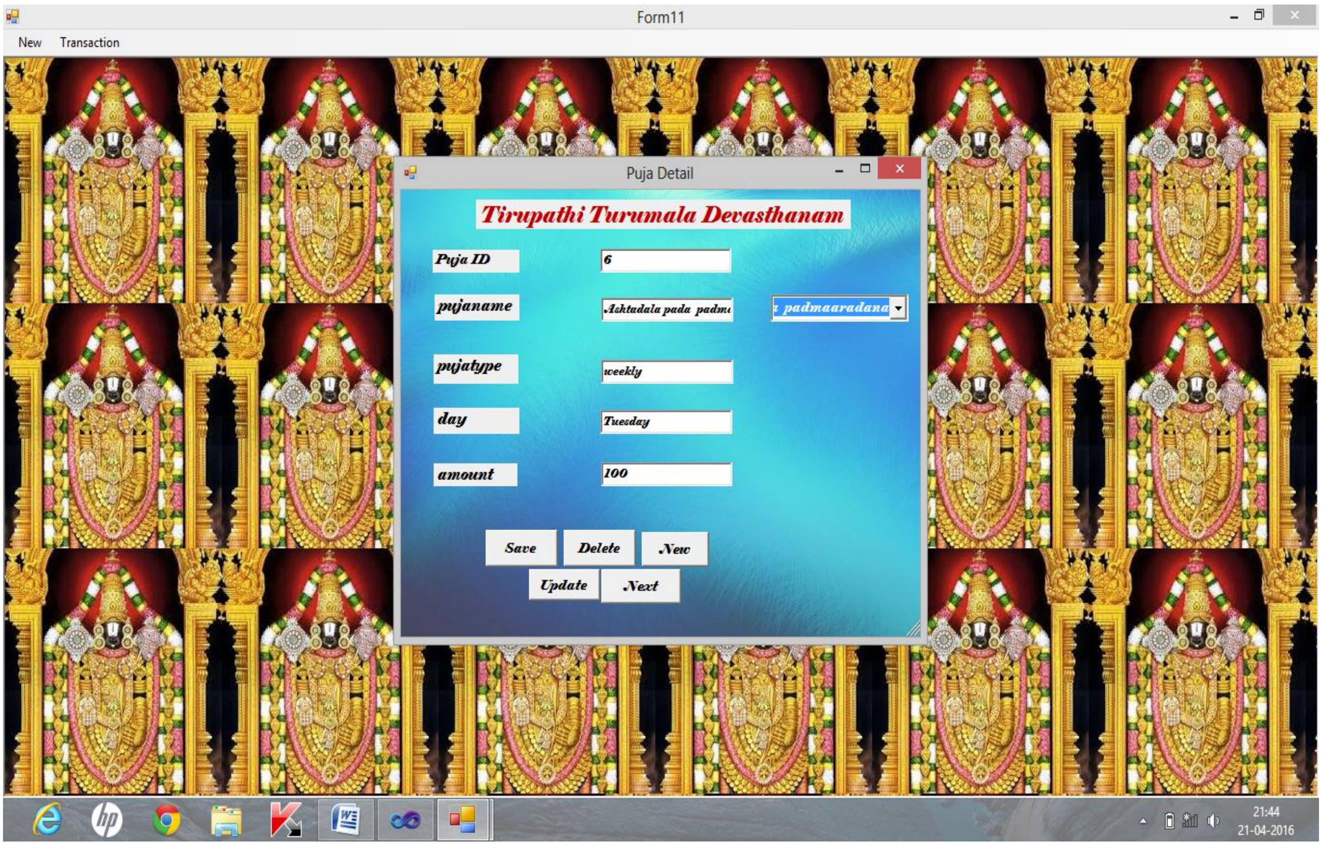
Tirupathi Tirumala Devasthanam

poojas list

Daily pujas
Thomala seva
Unjal seva
Sahasra Deepaalankarana
Ekanta seva

Weekly sevas
Visesha pooja(monday)
Ashtadala pada padmaaradhana se
Sahasra kalasabhishekam seva(wedn
Tiruppavada seva(thursday)
Poolangi seva(friday)
Abhishekam(saturday and sunday)

Pooja detail:

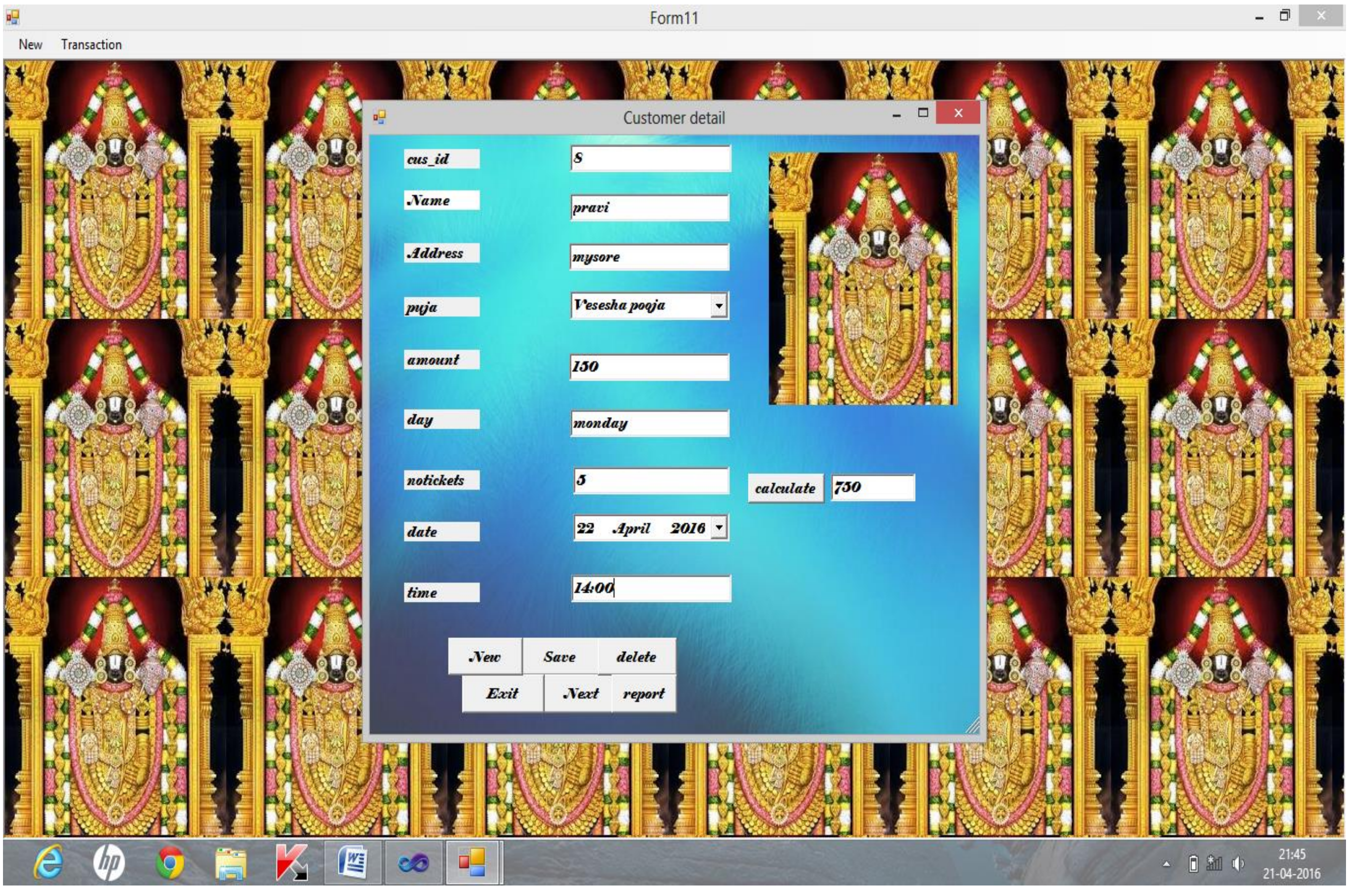


Tirupathi Turumala Devasthanam

Puja ID	6
pujaname	Ashtadala pada padm
pujatype	weekly
day	Tuesday
amount	100

a padmaaradana

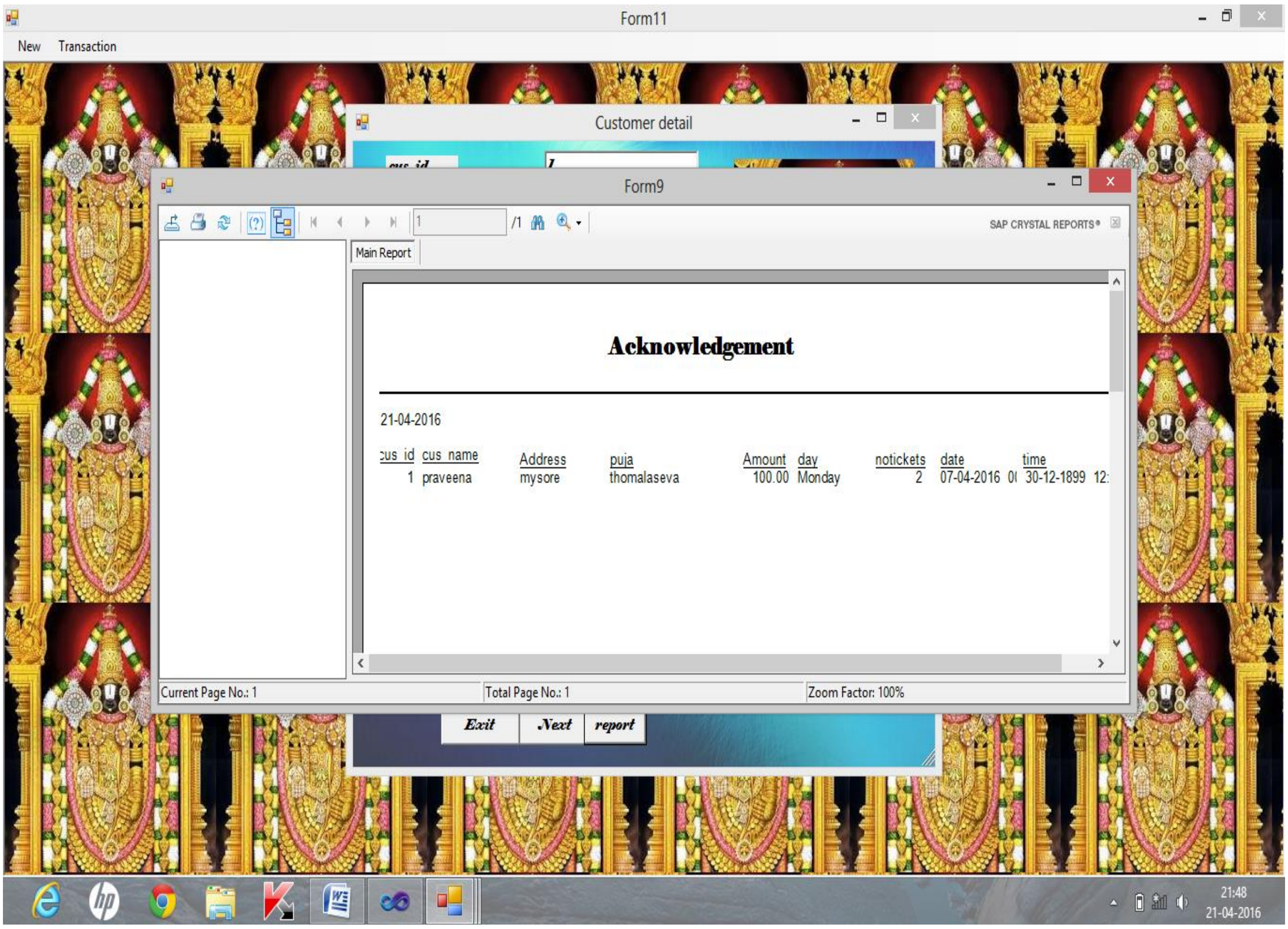
Customer detail:



cus_id	8
Name	pravi
Address	mysore
puja	Vesesha pooja
amount	150
day	monday
notickets	5
date	22 April 2016
time	14:00

calculate 750

Receipt:

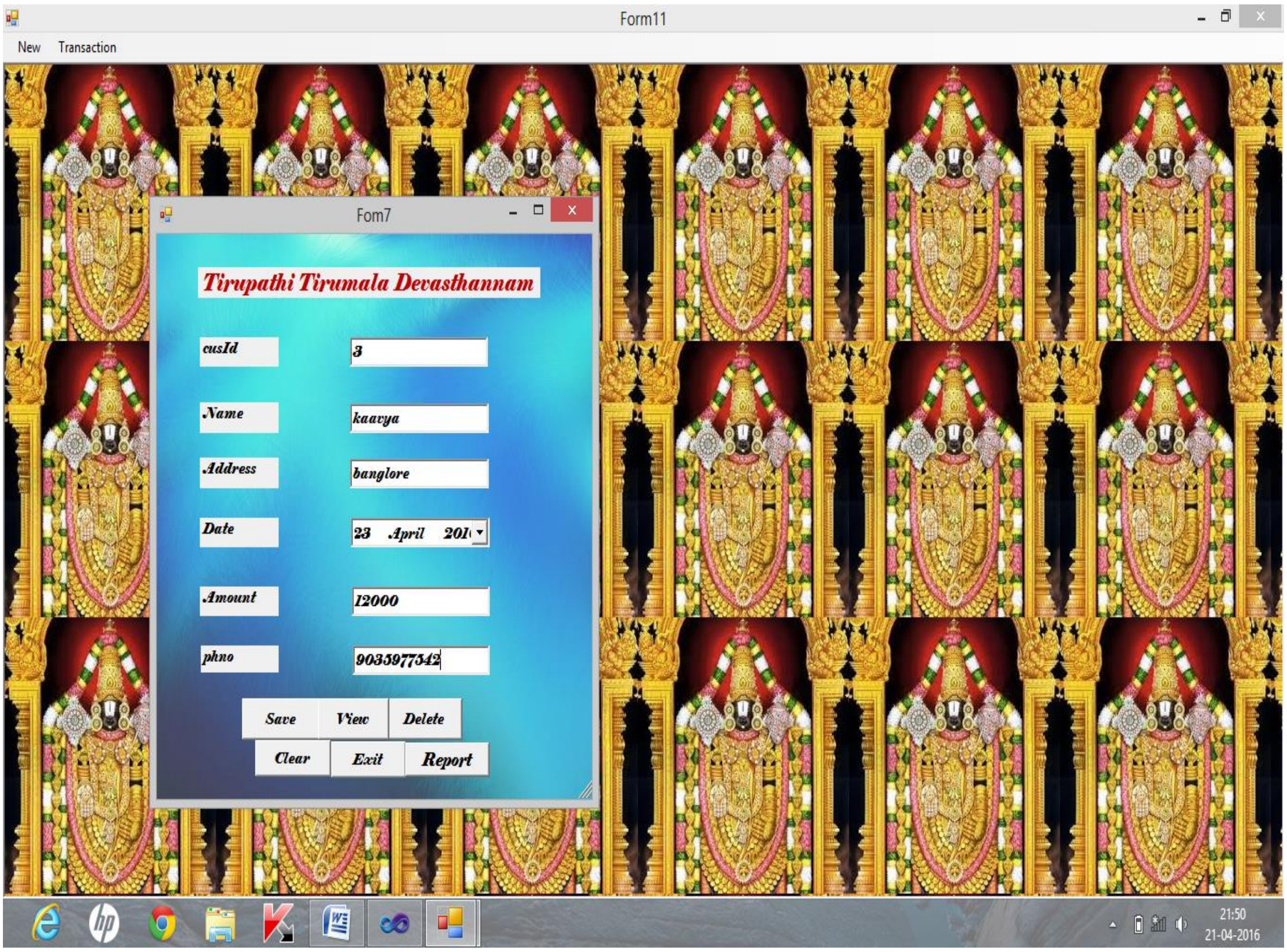


Acknowledgement

21-04-2016

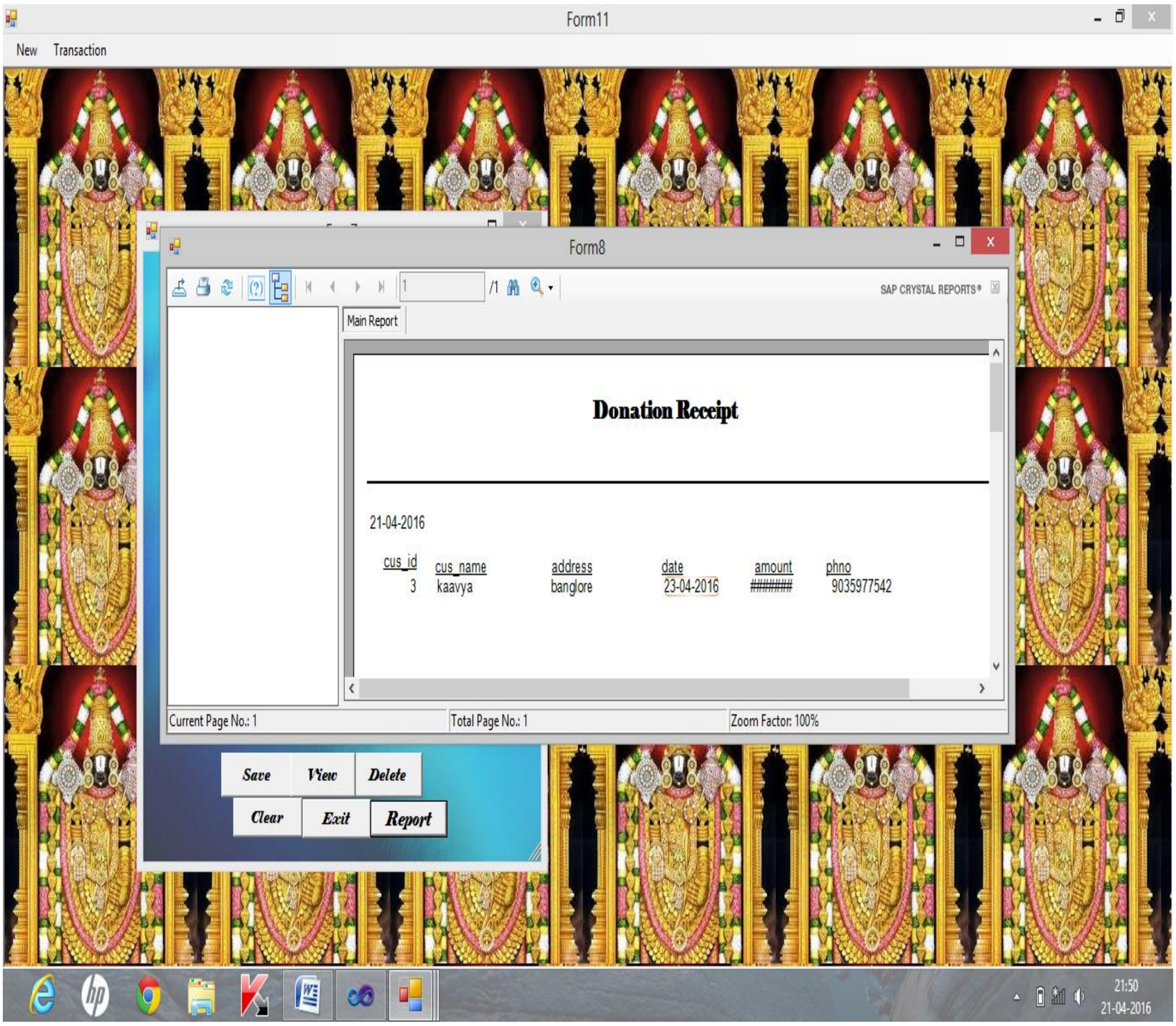
:us id	cus name	Address	puja	Amount	day	notickets	date	time
1	praveena	mysore	thomalaseva	100.00	Monday	2	07-04-2016 0(	30-12-1899 12:

Donation:



cusId	3
Name	kaavya
Address	banglore
Date	23 April 201
Amount	12000
phno	9035977542

Receipt:



Donation Receipt

21-04-2016

cus_id	cus_name	address	date	amount	phno
3	kaavya	banglore	23-04-2016	#######	9035977542

Current Page No.: 1

Total Page No.: 1

Zoom Factor: 100%

21:50
21-04-2016

11.0: Future Enhancement

In future the following features can be added
1.You can add or update new poojas.
2.Able to add many forms.
3.Developing similar system for Ticket Reservation.

CONCLUSION

Our project is namely used for maintaining the management for the **“TICKET RESERVATION SYSTEM”.** The project reduces the manual work and time. The security for storing the result will be accurate. Our project is mainly used for maintaining the ticket booking for pooja and donation details. The gathering of input is very easy and cuts down the accumulation of relevant data. The possible input screens are designed in the model of forms requesting a specific detail at a time. The developed system is flexible and changes can be made easily. Speed and accuracy is enhanced. Needed reports can be printed well.

REFERENCES

References Made From:

[1] Steven Holzner " visual basic.Net programming Black book" in 2005 Edition
[2] Murach's "beginning visual basic.NET"
[3] "The Ultimate VB.NET and ASP.NET code Book" Karl Moore
[5] "Visual Basic.NET TIPS & TECHNIQUES" Kris Jamsa

Sites Referred:

- www.vbtutor.net
- www.dotnetspider.com
- www.codeproject.com
- www.tutorialspoint.com
- www.techotopia.com
- https://www.safaribooksonline.com
- https://msdn.microsoft.com