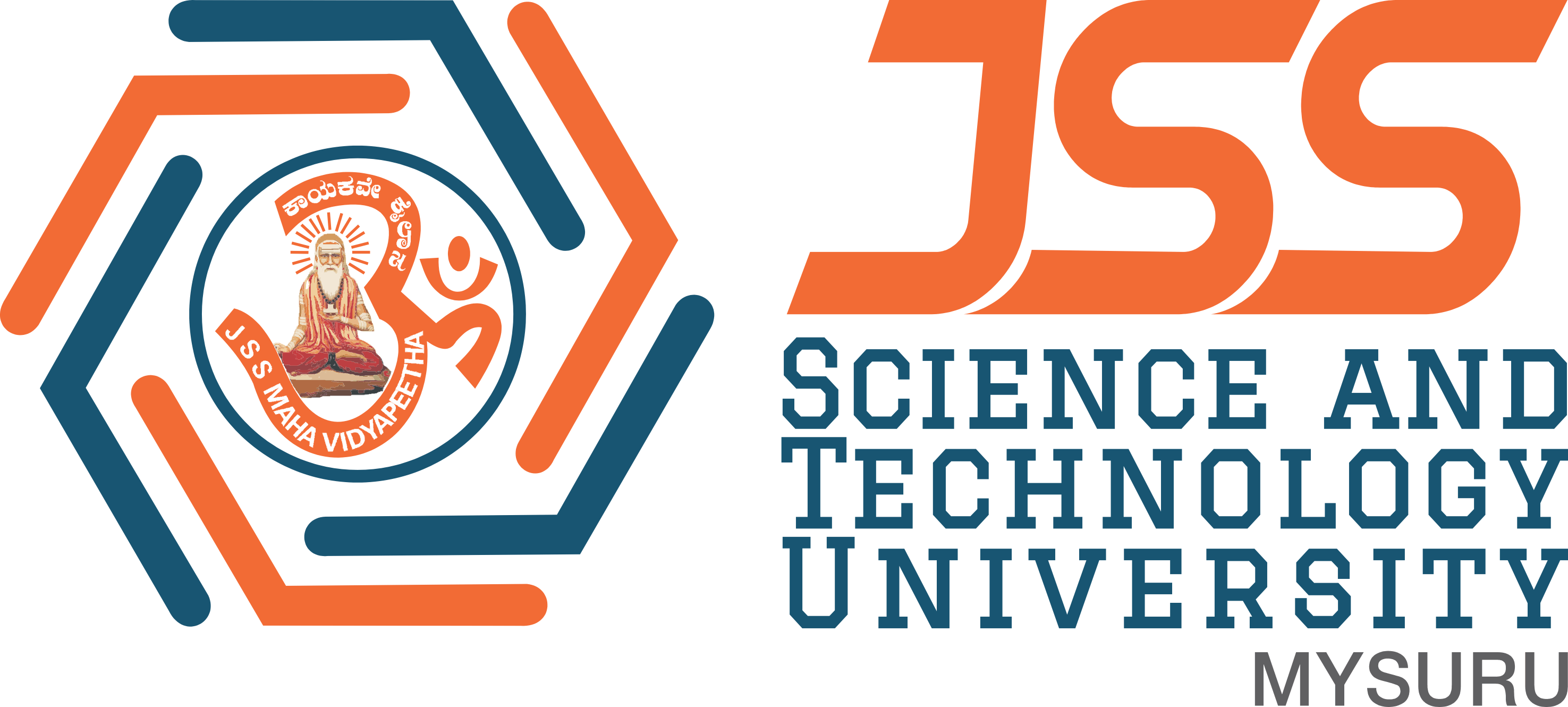
**JSS MAHAVIDYAPEETHA**

**JSS SCIENCE AND TECHNOLOGY UNIVERSITY**

JSS Technical Institutions Campus, Mysuru – 570006



**“EMPLOYEE BACKGROUND VERIFICATION ”**

Mini project report submitted in partial fulfillment of curriculum prescribed for the Database Systems (CS510) course for the award of the degree of

#### BACHELOR OF ENGINEERING

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*by*

# **Naveen J Chethana S**

**(01JST16CS057) (01JST16CS024)**

# **Srihari Charith Harshitha M.L**

**(01JST16CS105) (01JST16CS068)**

*Under the Guidance of*

#### Dr. Trisiladevi C. Nagavi

Assistant Professor,

Dept.of CS & E,

JSS STU Mysore

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

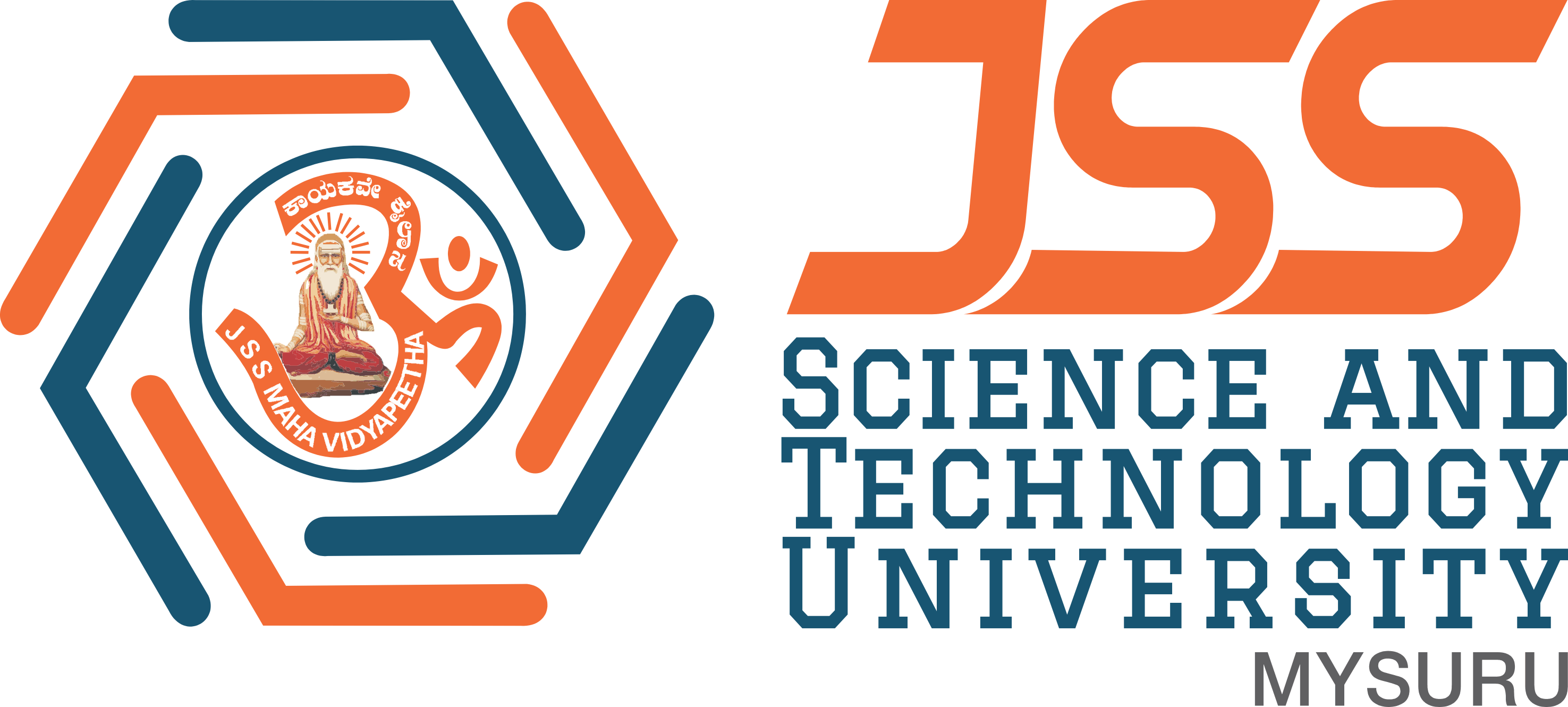
**November 2018**

|  |  |  |
| --- | --- | --- |
|  |  |  |

**JSS MAHAVIDYAPEETHA**

**JSS SCIENCE AND TECHNOLOGY UNIVERSITY**

JSS Technical Institutions Campus, Mysuru – 570006



**CERTIFICATE**

This is to certify that the work entitled “Employee Background Verification” is a bonafied work carried out by Naveen J,Srihari Charith,Harshitha M.L and Chethana.S in partial fulfillment of the award of the degree of Bachelor of Engineering in Computer Science and Engineering of JSS Science and Technology, Mysuru during the year 2018. It is certified that all corrections / suggestions indicated during CIE have been incorporated in the report. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed for the Database Systems (CS510) course.

## Course In Charge and Guide Lab In Charge and Guide

**Dr. Trisiladevi C. Nagavi Dr. Manimala S.**

Assistant Professor Assistant Professor

Dept. of CS & E Dept. of CS & E

JSS STU Mysore JSS STU Mysore

**Place:** Mysore **Date : 15.11.2018**

**CONTENTS:**

**1: INTRODUCTION**

Any company will conduct background verification of employees after conducting interview. Utilizing background checks in hiring procedures can protect the company from future litigation charges related to hiring negligence, safeguard the companies’ assets and public relations, and make current employees feel safe and secure in their work environments. For these reasons, in combination with the fact that in today’s fast paced and globalized world it can be difficult to verify the details of a CV, background checks are an important part of many companies’ hiring procedures

* 1. **Scope and objective**

The objective of this project is to conduct pre-employment background screening and conducting background check throughout individuals’ employment and reduce frauds. Utilizing background checks in hiring procedures can protect the company from future litigation charges related to hiring negligence, safeguard the company’s assets and public relations, and make current employees feel safe and secure in their work environments. For these reasons, in combination with the fact that in today’s fast paced and globalized world, it can be difficult to verify the details of a CV, thus background checks are an important part of many companies hiring procedures. Dangers of hiring and on-boarding employees with fudged education certificates, tampered salary slips can be reduced.

* 1. FEATURES OF THE PROJECT

System Requirements

*Software requirements*

* Frame work: DOTNET
* IDE: Visual Studio 2010
* Front end: ASP.NET 4.0
* Programming Language : C#.NET

*Hardware Requirements*

* RAM : 1GB+
* Processor: Pentium 4+
* Processor Speed: 2ghz+
* Hard Disk: 20GB+

Microsoft's core implementation of Dot-NET includes:

* C# (a new programming language)
* the Common Language Runtime (for support of other programming languages)
* a collection of components that provide support for networking, security, and other "base" services commonly needed in distributed applications

C#(pronounced as see sharp) is a multi-paradigm programming language encompassing strong typing, imperative, declarative, functional,generic, object-oriented (class-based), and component-oriented programming disciplines. It was developed by Microsoft within its .NET initiative and later approved as a standard by Ecma (ECMA-334) and ISO (ISO/IEC 23270:2006). C# is one of the programming languages designed for the Common Language Infrastructure

2.SYSTEM DESIGN

The purpose of the design phase is to plan a solution of the problem specified by the requirements document. This phase is the first step in moving from the problem domain to the solution domain. In other words, starting with what is needed; design takes us toward how to satisfy the needs. The design of a system is perhaps the most critical factor affecting the quality of the software; it has a major impact on the later phases particularly testing and maintenance.

The design activity often results in three separate outputs –

* Architecture design.
* High level design.
* Detailed design.

**Architecture Design:**

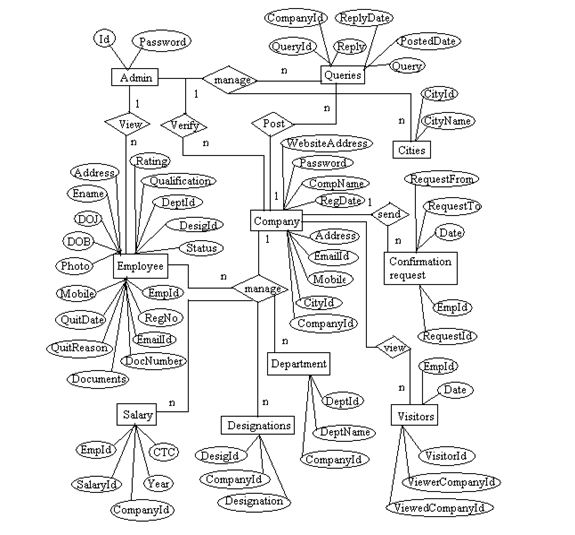
Architecture focuses on looking at a system as a combination of many different components, and how they interact with each other to produce the desired result. The focus is on identifying components or subsystems and how they connect. In other words, the focus is on what major components are needed.

**High Level Design:**

In high level design identifies the modules that should be built for developing the system and the specifications of these modules. At the end of system design all major data structures, file format, output formats, etc., are also fixed. The focus is on identifying the modules. In other words, the attention is on what modules are needed.

**Detailed Design:** In the detailed design the internal logic of each of the modules is specified. The focus is on designing the logic for each of the modules. In other words how modules can be implemented in software is the issue.

**2.1 ER DIAGRAM- HIGH LEVEL DATA MODELING**



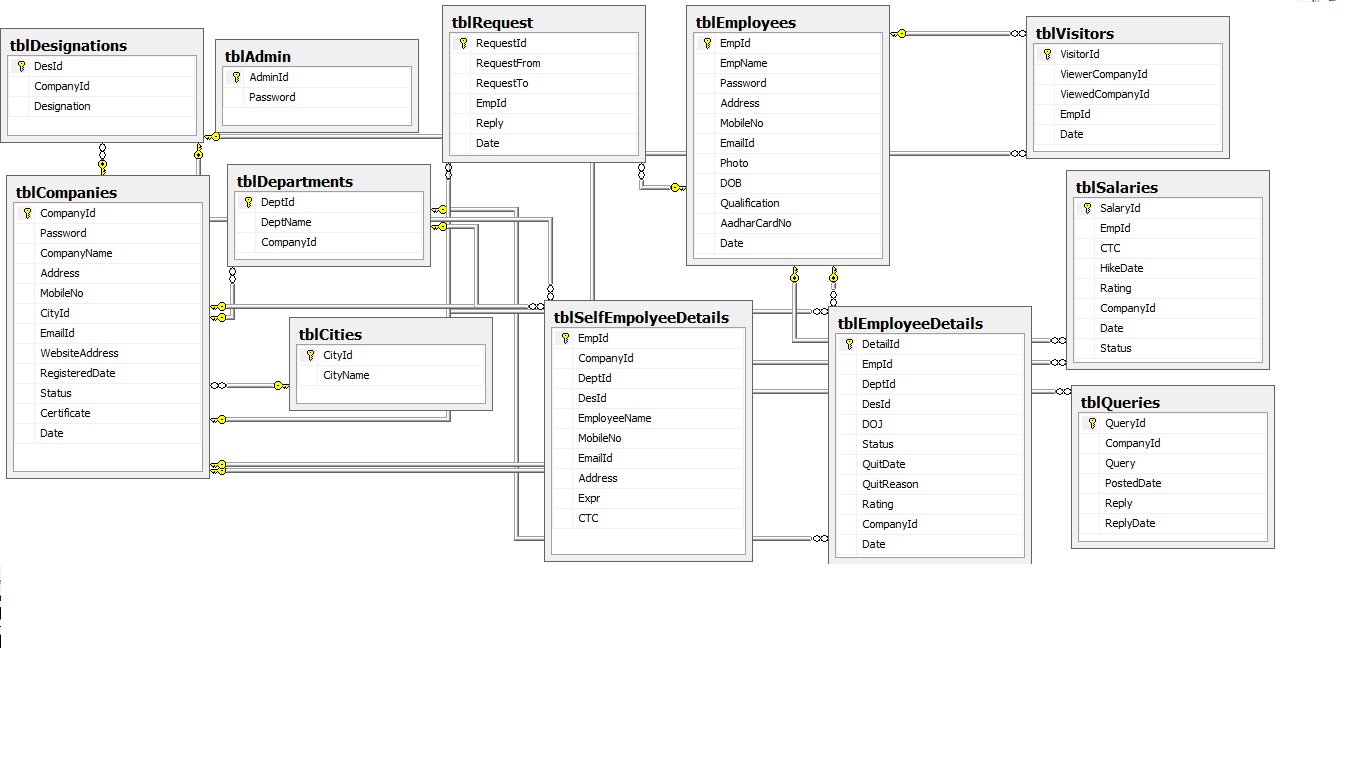
2.2 SCHEMA DIAGRAM – CONCEPTUAL DATA MODELING

A conceptual data model is a summary-level data model that is most often used on strategic data projects.  It typically describes an entire enterprise.  Due to its highly abstract nature, it may be referred to as a conceptual model.

**Common characteristics of a conceptual data model:**

* Enterprise-wide coverage of the business concepts.  Think Customer, Product, Store, Location, Asset.
* Designed and developed primarily for a business audience
* Contains around 20-50 entities (or concepts) with no or extremely limited number of attributes described. Sometimes architects try to limit it to printing on one page.
* Contains relationships between entities, but may or may not include cardinality and nullability.
* Entities will have definitions.
* Designed and developed to be independent of DBMS, data storage locations or technologies.  In fact, it would address digital and non-digital concepts. This means it would model paper records and artifacts as well as database artifacts.

2.3 STATE DIAGRAM



3.

4.SYSTEM IMPLEMENTATION

This web application is implemented using object oriented programming language. Object oriented programming is an approach that provides a way of modularizing programs by creating partitioned memory area for both data and functions that can be used as templates for creating copies of such modules on demand.

Features of Object Oriented paradigm:

* Emphasis is on data rather than procedure.
* Programs are divided into what are known as objects.
* Data structures are designed such that they characterize the objects.
* Methods that operate on the data of an object are tied together in the data structure.
* Objects may communicate with each other through methods.
* New data and methods can be easily added whenever necessary.
* Follows bottom-up approach in program design.
* Data is hidden and cannot be accessed by external functions.

**4.1 INTRODUCTION TO SQL SERVER**

Microsoft SQL Server 2005 is a database platform for large-scale online transaction processing (OLTP), data warehousing, and e-commerce applications; it is also a business intelligence platform for data integration, analysis, and reporting solutions.

SQL Server 2005 introduces "studios" to help you with development and management tasks: SQL Server Management Studio and Business Intelligence Development Studio. In Management Studio, you develop and manage SQL Server Database Engine and notification solutions, manage deployed Analysis Services solutions, manage and run Integration Services packages, and manage report servers and Reporting Services reports and report models.

In BI Development Studio, you develop business intelligence solutions using Analysis Services projects to develop cubes, dimensions, and mining structures; Reporting Services projects to create reports; the Report Model project to define models for reports; and Integration Services projects to create packages.

In the studios, SQL Server 2005 provides the graphical tools you need to design, develop, deploy, and administer relational databases, analytic objects, data transformation packages, replication topologies, reporting servers and reports, and notification servers. Additionally, SQL Server 2005 includes command prompt utilities to perform administrative tasks from the command prompt. To quickly get to important high-level topics for tools and utilities, go to Tools and Utilities Documentation Map.

The Database Engine is the core service for storing, processing, and securing data. The Database Engine provides controlled access and rapid transaction processing to meet the requirements of the most demanding data consuming applications within your enterprise.

Use the Database Engine to create relational databases for online transaction processing or online analytical processing data. This includes creating tables for storing data, and database objects such as indexes, views, and stored procedures for viewing, managing, and securing data. You can use SQL Server Management Studio to manage the database objects, and SQL Server Profiler for capturing server events.

4.2 RELATIONAL ALGEBRAIC QUERIES

4.3 QUERIES DESIGNED USING SQL COMMANDS

1.CREATE

a. CREATE TABLE [dbo].[16BIT](

[EmpId] [int] NULL,

[16bit] [varchar](50) NULL

) ON [PRIMARY]

b.CREATE TABLE [dbo].[ADMIN](

[AdminId] [varchar](150) NULL,

[AdminPassword] [varchar](150) NULL

) ON [PRIMARY]

2.INSERT

a. INSERT INTO [database].[dbo].[EXPERIENCE]

([EmpId]

,[EmpName]

,[Domain]

,[Projects]

,[NoOfYrs]

,[Description]

,[CurrentProject])

VALUES

(<EmpId, int,>

,<EmpName, varchar(50),>

,<Domain, varchar(50),>

,<Projects, varchar(350),>

,<NoOfYrs, int,>

,<Description, varchar(350),>)

b. INSERT INTO [database].[dbo].[COMPANY]

([CmpnyName]

,[NoOfEmp]

,[Address]

,[Website]

,[Mail]

,[Password])

VALUES

(<CmpnyName, varchar(50),>

,<NoOfEmp, int,>

,<Address, varchar(50),>

,<Website, varchar(50),>

,<Mail, varchar(50),>

,<Password, varchar(50),>)

3.SELECT

a.SELECT [EmpID]

,[CmpnyId]

,[EmpName]

,[Email]

,[Password]

,[Address]

,[Status]

,[Adhaar]

FROM [database].[dbo].[EMPLOYEE]

b. SELECT [EmpId]

,[EmpName]

,[Domain]

,[Projects]

,[NoOfYrs]

,[Description]

,[CurrentProject]

FROM [database].[dbo].[EXPERIENCE]

4.ALTER

a.ALTER view [dbo].[viewemp]

as

SELECT EMPLOYEE.\*

FROM EMPLOYEE;

5.UPDATE

a.UPDATE [database].[dbo].[COMPANY]

SET [CmpnyName] = <CmpnyName, varchar(50),>

,[NoOfEmp] = <NoOfEmp, int,>

,[Address] = <Address, varchar(50),>

,[Website] = <Website, varchar(50),>

,[Mail] = <Mail, varchar(50),>

,[Password] = <Password, varchar(50),>

WHERE <Search Conditions,,>

b. UPDATE [database].[dbo].[16BIT]

SET [EmpId] = <EmpId, int,>

,[16bit] = <16bit, varchar(50),>

WHERE <Search Conditions,,>

6. DELETE

a. DELETE FROM [database].[dbo].[THIRDPARTY]

WHERE <Search Conditions,,>

b. DELETE FROM [database].[dbo].[CATEGORY]

WHERE <Search Conditions>

7. GROUP BY

a. SELECT c.CmpnyName as [Company Name],count(e.EmpID)

from COMPANY c,EMPLOYEE e

where c.CmpnyId=e.CmpnyId

GROUP BY c.CmpnyName

HAVING count(e.EmpID)>0;

b. SELECT COMPANY.CmpnyName as CompanyName, MAX(EMPLOYEE.Salary) AS [maximum salary]

FROM COMPANY INNER JOIN

EMPLOYEE ON COMPANY.CmpnyId = EMPLOYEE.CmpnyId

GROUP BY COMPANY.CmpnyName

8.NESTED

a.SELECT EmpID,EmpName, Email, Password, Address, Adhaar

from EMPLOYEE e

where EXISTS

(select \* from EXPERIENCE E where

e.EmpID=E.EmpID )

and e.CmpnyId=@cid;

b.SELECT \*

from GUEST

where GuestId in

(select GuestId from GUEST

where bit=0);

9.CORRELATED

a. SELECT COMPANY.CmpnyName as CompanyName, MAX(EMPLOYEE.Salary) AS [maximum salary]

FROM COMPANY INNER JOIN

EMPLOYEE ON COMPANY.CmpnyId = EMPLOYEE.CmpnyId

GROUP BY COMPANY.CmpnyName

10. HAVING CLAUSE

SELECT c.CmpnyName as [Company Name],count(e.EmpID)

from COMPANY c,EMPLOYEE e

where c.CmpnyId=e.CmpnyId

group by c.CmpnyName

having count(e.EmpID)>0;

11.VIEWS

CREATE VIEW [dbo].[vEmployeeWithCompany]

AS

SELECT dbo.COMPANY.CmpnyName, dbo.EMPLOYEE.EmpID, dbo.EMPLOYEE.CmpnyId, dbo.EMPLOYEE.EmpName, dbo.EMPLOYEE.Email

FROM dbo.COMPANY INNER JOIN

dbo.EMPLOYEE ON dbo.COMPANY.CmpnyId = dbo.EMPLOYEE.CmpnyId

12. LIKE

SELECT \* from COMPANY

where CmpnyId like '1%'

13.BETWEEN

SELECT Address, CmpnyId, CmpnyName, Mail, NoOfEmp, Password, Website FROM COMPANY WHERE (CmpnyName = @cname and( CmpnyId between 1000 and 2000));

14.EXISTS

SELECT EmpId, EmpName, Domain, Projects, NoOfYrs, Description, CurrentProject

FROM EXPERIENCE

where exists

(select \* from experience where NoOfYrs>0)

15.UNION

SELECT CatId, Category, Description, Requirements, Salary, Vacancy FROM CATEGORY WHERE (CompanyID =@CompanyID and CatId<=15) union SELECT CatId, Category, Description, Requirements, Salary, Vacancy FROM CATEGORY WHERE (CompanyID =@CompanyID and CatId>=1); GetCatById(cid)

5. SYSTEM TESTING AND RESULTS

System Testing is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements.

System Testing is a level of the software testing process where a complete, integrated system/software is tested. The purpose of this test is to evaluate the system’s compliance with the specified requirements

* Finding defects which may get created by the programmer while developing the software.
* To prevent defects.
* To make sure that the end result meets the business and user requirements.
* To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
* To gain the confidence of the customers by providing them a quality product.
* All functional requirements are satisfied.
* All performance requirements are achieved.
* Other requirements like transportability, compatibility, error recovery etc. are satisfied.
* Acceptance criteria specified by the user is met.

6. CONCLUSION

The project which we have developed is very useful for all the company. Once registered to this website, company can perform background check of employees very easily. We save time and money for background check. Whenever any employee joins, the details of his CV are verified in this website based on his unique identity number..

For a fresher, if we enter his unique identity number to the website, it will accept the number. If experienced, when we enter the unique identity number, the website already shows the previous employment details. Thus, we are verifying each employee. By constant use of our website, we are successful in achieving the accurate result.

7.REFERENCES

REFERRED WEBSITES

http://www.wikipedia.org

http://www.ieeexplore.ieee.org

http://download-llnw.oracle.com/javase/tutorial/index.html

REFERRED BOOKS

* “Programming in C#, 3E” by Balguruswamy Tata McGraw-Hill Education, 2010
* “Begin ASP.NET 2.0 with visual C#.NET”, Wrox, By Chris Ullman.
* Software Engineering, Ian Summerville, Sixth Edition, Pearson Education Ltd, 2001
* Ian Somerville, Software Engineering, Pearson Education, Sixth Edition
* ASP.net complete reference - Pearson Publication