A

PROJECT REPORT ON

**“EMPLOYEE MANAGEMENT SYSTEM”**

SUBMITTED BY

**YARDI ADITYA CHANDRASHEKHAR (PRNO: 04416004269) VAGAL DEVESH KIRAN (PRNO: 04416003166)**

TO THE PARTIAL FULFILMENT OF

**BACHELOR OF COMPUTER APPLICATION**

**VIKAS COLLEGE STUDY CENTRE, VIKHROLI (EAST) [MUMBAI] TILAK MAHARASHTRA VIDYAPEETH, PUNE**

**(2018-2019)**

EMPLOYEE MANAGEMENT SYSTEM



TILAK MAHARSHTRA VIDYAPEETH, PUNE

(‘Deemed University’ under section 3 of UGC Act 1956 vide notification NO.F 9- 19/85-U3 dated 24th April 1987 by the Government of India) Vidyapeeth Bhavan, Gultekdi, Pune- 411 037

CERTIFICATE

This is to certify the project titled “EMPLOYEE MANAGEMENT SYSTEM” Is a bonafied work carried out by

**YARDI ADITYA CHANDRASHEKHAR (PRNO: 04416004269) VAGAL DEVESH KIRAN (PRNO: 04416003166)**

By Students of **Bachelor in Computer Application**

Semester 6th

Under Tilak Maharashtra Vidyapeeth in the Year 2019

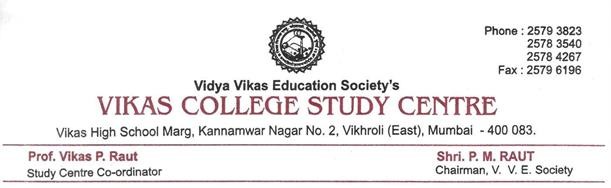
Head of the Department Examiner Internal Examiner External

Date:

Place:

University Seal

EMPLOYEE MANAGEMENT SYSTEM



**GUIDE CERTIFICATE**

This is to certify that the project “EMPLOYEE MANAGEMENT SYSTEM”

Has been satisfactorily completed by

**YARDI ADITYA CHANDRASHEKHAR (PRNO: 04416004269) VAGAL DEVESH KIRAN (PRNO: 04416003166)**

Towards the Partial Fulfillment of the ‘**Bachelor of Computer Application’**,

For the Academic Year (2018-2019) at VIKAS COLLEGE STUDY CENTRE, VIKHROLI (EAST) [MUMBAI]

Tilak Maharashtra Vidyapeeth, Pune (Faculty of Distance Education), And it is Approved.

Project Guide EXAMINER Head of Department

**Mr. HARISHANKAR MISHRA Mr. VIKAS RAUT**

**ACKNOWLEDGEMENT**

EMPLOYEE MANAGEMENT SYSTEM

With immense please we are presenting “EMPLOYEE MANAGEMENT SYSTEM” Project report as a part of the curriculum of ‘Bachelor of Computer Application’. We wish to thank all the people who gave us unending support.

We express our profound thanks to our Head of Department “**Mr. VIKAS RAUT**” and Project Guide and Project In-charge “**Mr. HARISHANKAR MISHRA**” and all those who have indirectly guided and helped us towards development of this project.

**YARDI ADITYA CHANDRASHEKHAR**

**(PRN NO: - 04416004269) VAGAL DEVESH KIRAN (PRN NO: - 04416003166)**

# Project Synopsis

The Report includes a development presentation of an Employee Management System built to administer the employee staff within a small organisation or company. Employees are the backbone of every organisation. Hence it is quintessential for the organisation to keep track of Employee records. With the purpose, this Employee Management System not only helps the employer in monitoring Employees, but also empowers him to ADD new employees, transfer, promote and terminate his employees. This System with the use of KPI(Key Performance Indicators) is capable of quantifying the performance of each and every employee in the organisation. Each employee in database is assigned with a designation in respective department. So, it can be verified if there is any duplication in designations and or employees.

The System is very flexible and easy to use solution, ideal for small-medium scale organisations. It provides various modules for personnel information and performance management so that companies are able to manage their crucial organisation asset – Employees. Combination of these modules into one application assures the perfect platform for re-engineering and aligning Human Resource processes along with the organisational goals.

This system thus brings about an easy way of maintaining the details and evaluating periodic performance of employees working in the organisation.

# Requirement Specifications:

|  |  |
| --- | --- |
| **Name of Component** | **Specification** |
| Operating System | Windows 7 and Above |
| Language Used | C#, SQL |
| Database | SQLServer |
| Development Platform | VisualStudio 2015 Professional, Sql Management Studio 2014 Express |
| Processor | 2.0 Gigahertz with Dual Core or Above |
| Printer | Optional |
| Barcode Scanner | Optional |

**Index**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Topics** | **Pg No.** |
| 1 | Introduction | 1 |
| 2 | Need for New System | 2 - 3 |
| 3 | Feasibility Study | 4 |
| 4 | Technology Review | 5 - 8 |
| 5 | SDLC Life Cycle | 9 |
| 6 | Software Development Model | 10 - 12 |
| 7 | Data Dictionary | 13 - 17 |
| 8 | Gantt Chart | 18 - 19 |
| 9 | UML Diagrams   * 1. E-R Diagrams   2. Use Case Diagrams   3. Class Diagrams   4. Object Diagrams   5. Context Level Diagrams   6. Data Flow Diagrams   7. Functional Decomposition Diagram   8. Sequence Diagram   9. Activity Diagram | 20  21 - 23  24 - 25  26 - 27  28 - 29  30 - 31  32 - 33  34 - 35  36 - 39  40 - 41 |
| 10 | DataStructures Screenshots | 42 - 50 |
| 11 | Project ScreenShots | 51 - 67 |
| 12 | Testing and Validations | 68 - 70 |
| 13 | System Maintenance | 71 - 72 |
| 14 | Limitations and Future Enhancements | 73 |
| 15 | Conclusion | 74 |
| 16 | Bibliography | 75 - 76 |

**Ch.1 Introduction**

* 1. Problem Statement

Comprehending the current Employee Management Systems, they provide limited solutions to the Organisation. Current Softwares either focus on Employee Payrolls, Attendance or Monitor Employee Records. They fail to deliver one stop solution that could address all the key values and execute all functions that are essential for monitoring employee records and administering employee performances. So, any organisation inclined towards automation is either levied upon with burden of adopting multiple softwares that not only becomes cumbersome but also turnouts to be very expensive or else is forced to content with orthodox manual methods. Organisation maintaining those multiple softwares simultaneously makes it difficult to access employee’s relative information at a go.

Also maintenance of Employee Records in the form of hardcopy makes it laborious to reproduce the necessary data as and when required since data filtering consumes time. Exercising Manual practices to map the Employee Salaries and pay on the basis of his pay scale, leaves and performance is challenging.

* 1. Limitations Of Existing Systems:

-Need of extra manual efforts.

-Do not provide one-stop solution.

-Hard to retrieve relative information at a time.

-Complex with regards to user interaction.

-Certain Reports are difficult to generate.

# Ch.2 Need for New System

Proposed System is multi-user, so employee can also participate in the system to mark his attendance, view his salary and resume.

The proposed system provides detailed information and quality insight to employer about the employees along with their personal, qualification, job experience and performance details.

It enhances employer’s monitoring efficiency in Adding Employees, updating their details and enacting their transfers, promotions and appraisals.

It calibrates employee’s Performance by implementing KPIs. Incentives can be offered based on the Employee’s performances. Thus, It will motivate employees for better performance.

Thus proposed system is the right software to be incorporated into automation of Employee Management for helping the organisation needs with respect to skillful and effective Human Resource Management.

* 1. Features of New System

**Real-Time Access to the Employee Data**: In regards to Employee’s Personal/Educational- professional Data.

**Attendance Record**: Maintain the Employee attendance record and track his absenties, late marks that may impact his salary.

**Employee Performance**: Using Key Performance Indicators(KPIs) , real time analysis of Employee Performance.

**Employee Salary**: Dynamic Employee salary manipulation based on employee’s Attendance record and his Performance.

**Employee Transfers, Promotions and Appraisals**: Efficient to take managerial actions with respect to Employee Transfers, Promotions and Appraisals.

**Reporting**: Create and provide reporting as and when required.

**Graphical Analysis**: To provide Graphical analysis on Employee’s Performance.

* 1. Advantages of Proposed System:

1. One-stop Solution for HR Management.
2. Very Easy to operate and Accurate.
3. Savings in Operational Cost.
4. Time Effective and Performance efficient.
5. Graphical Analysis to help Organisation in Decision Making.
6. Foster Employee Culture by motivating Employees to Perform.
7. Secure. Only Administrators are authorised to change Password to access the system. If Employee forgets or wishes to change his password, he needs to request Admin for the same.
8. High Reliability. Data access does not depend on the network solutions. Therefore the database reliability depends only on the existing system. It results in high availability of data.

# Ch.3 Feasibility Study

* 1. Technical Feasibility

The minimum hardware requirement to implement the system is Pentium IV processor with 512 MB RAM. The whole system is being developed on Visual Studio 2015 Professional IDE for GUI (Front End) and uses Microsoft SQL Management Studio 2014 for Database. These are the softwares which are very popular and easily available in market.

* 1. Economical Feasibility

It is cost effective system. No extra efforts are needed to train for using the system. This system is quite beneficial with respect to its cost. The user shall be conversant with computer technology. The system is user friendly and thus easy to use and operate.

* 1. Operational Feasibility

This system will reduce the workload and loss of vital information. It is time saving and will aid the employee administration to be system generated rather than person dependent. Data retrieval and data presenting will be done by the system so the report generation will be automated.

# Ch.4 Technology Review

ASP.Net Framework is chosen for development of Employee Management System with Visual Studio Professional 2015 (IDE) and Visual C# as core Programming language and SQL Server.

* 1. Framework

ASP.Net framework is a revolutionary platform that allows to build the following types of applications −

Windows applications Web applications Web services

The .Net framework applications are multi-platform applications. The framework has been designed in such a way that it can be used from any of the following languages: C#, C++, Visual Basic, Jscript, COBOL, etc. All these languages can access the framework as well as communicate with each other. The .Net framework consists of an enormous library of codes used by the client languages such as C#.

Following two are important components of the .Net framework − Common Language Runtime (CLR)

The .Net Framework Class Library

Other components are: Common Language Specification, Common Type System, Metadata and Assemblies, Windows Forms, ASP.Net and ASP.Net AJAX, ADO.Ne etc.

* + 1. **Common Language Runtime (CLR):** Common Language Runtime(CLR) is the basic and Virtual Machine component of the [.NET Framework](https://www.geeksforgeeks.org/c-net-framework-basic-architecture-component-stack/). It is the run-time environment in the .NET Framework that runs the codes and helps in making the development process easier by providing the various services. Basically, it is responsible for managing the execution of *.NET programs* regardless of any *.NET* programming language. Internally, CLR implements the *VES(Virtual Execution System)* which is defined in the Microsoft’s implementation of the *CLI(Common Language Infrastructure).*

The code that runs under the Common Language Runtime is termed as the Managed Code as CLR provides a managed execution environment for the *.NET* programs by improving the security, including the cross language integration and a rich set of class libraries etc.

* + 1. **Framework Class Library (FCL):** It is the collection of reusable, object- oriented class libraries and methods etc that can be integrated with CLR. Also called the Assemblies. It is just like the header files in C/C++. Installing framework basically is the installation of CLR and FCL into the system.

Following are the commonly used namespaces that contains useful classes and interfaces and defined in Framework Class Library.

Following are the commonly used namespaces that contains useful classes and interfaces and defined in Framework Class Library:

1. System: It includes all common data-types, string values, arrays and methods for data conversion.
2. System.Data, System.Data.SqlClient: These are used to access a database, perform commands on a database and retrieve database.
3. System.Windows.Forms, System.Windows.Forms.Design: These namespaces are used to create Windows-based applications using Windows user interface components.
   1. Integrated Development Environment

For the project, Microsoft Visual Studio is used as a platform for system development. Microsoft Visual Studio is an I[ntegrated Development Environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) from [Microsoft](https://en.wikipedia.org/wiki/Microsoft). It is used to develop [computer programs](https://en.wikipedia.org/wiki/Computer_program), as well as [websites](https://en.wikipedia.org/wiki/Web_site), [web apps](https://en.wikipedia.org/wiki/Web_app), [web services](https://en.wikipedia.org/wiki/Web_service) and [mobile apps](https://en.wikipedia.org/wiki/Mobile_app). Visual Studio uses Microsoft software development platforms such as [Windows](https://en.wikipedia.org/wiki/Windows_API) [API,](https://en.wikipedia.org/wiki/Windows_API) [Windows Forms](https://en.wikipedia.org/wiki/Windows_Forms) etc.

Visual Studio supports 36 different [programming languages](https://en.wikipedia.org/wiki/Programming_language) and allows the code editor and debugger to support (to varying degrees) nearly any programming language, provided a language-specific service exists. Built-in languages include [C](https://en.wikipedia.org/wiki/C_(programming_language))[,C++](https://en.wikipedia.org/wiki/C%2B%2B), [C++/CLI](https://en.wikipedia.org/wiki/C%2B%2B/CLI), [Visual Basic](https://en.wikipedia.org/wiki/Visual_Basic_.NET)

[.NET,](https://en.wikipedia.org/wiki/Visual_Basic_.NET) [C#](https://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [F#,](https://en.wikipedia.org/wiki/F_Sharp_(programming_language))[JavaScript](https://en.wikipedia.org/wiki/JavaScript), [TypeScript,](https://en.wikipedia.org/wiki/TypeScript) [XML,](https://en.wikipedia.org/wiki/XML) [XSLT,](https://en.wikipedia.org/wiki/XSLT) [HTML,](https://en.wikipedia.org/wiki/HTML) and [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets). Support for other languages such as [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), [Ruby,](https://en.wikipedia.org/wiki/Ruby_(programming_language)) [Node.js,](https://en.wikipedia.org/wiki/Node.js) and [M](https://en.wikipedia.org/wiki/MUMPS) among others is available via [plug-ins.](https://en.wikipedia.org/wiki/Plug-in_(computing))

Visual Studio 2015 Professional version has been used for developing EMS. 2015 Pro version of Visual Studio has following added key features:

Syntax Highlighting Visual Studio IntelliSense

Error in Visual Studio 2015 Code Fixes and Refactoring

Visual C# language has been used for the purpose.

* 1. Programming Language

C# is one of the programming languages designed for the Common Language Infrastructure. C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft within its .NET initiative led by Anders Hejlsberg. It is designed for Common Language Infrastructure (CLI), which consists of the executable code and runtime environment that allows use of various high-level languages on different computer platforms and architectures. The language is chosen for the project because of following reasons:

It is a modern, general-purpose programming language. It is a part of .Net Framework.

It is object oriented and component oriented. It is structured and easy to learn.

It can be compiled on a variety of computer platforms.

It provides Automatic Garbage Collection and Standard Library. It supports Properties and Events.

It also features Delegates and Events Management.

* 1. Database

SQL Server Management Studio 2014 Express version is used to design database for the proposed Employee Management System. SQL language and Relational DataBase Management Systems(RDBMS) are used to set the foundation of system’s back end.

RDBMS is the basis for SQL, and for all modern database systems such as MS SQL Server, Oracle, MySQL, and Microsoft Access.

The data in RDBMS is stored in database objects called tables. A table is a collection of related data entries and it consists of columns and rows.

**S**tructured **Q**uery **L**anguage or **SQL** is a standard Database language which is used to create, maintain and retrieve the relational database. It is particularly used to work with structured data where there are relations associated within the data itself.

Programmers embed SQL commands into their application programs to access the data. SQL is a client/server language. Programs use SQL to communicate over a network with database servers that store shared data.

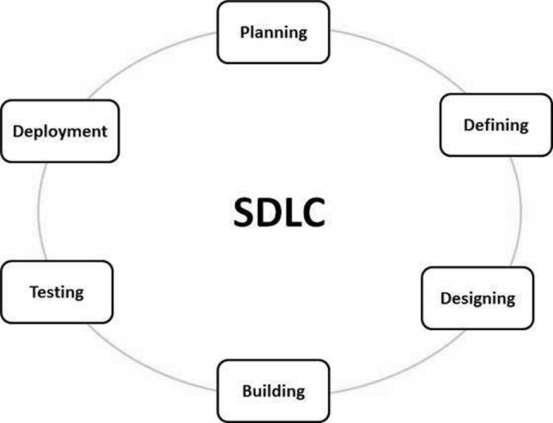
SQL being distributed database language, Distributed Database Management Systems use SQL to distribute data across multiple connected computer systems.

In the proposed system, SQL serves as the link between “front-end” computer systems optimized for user interaction and “back-end” systems specialized for database management.

# Ch.5 SDLC Life Cycle

SDLC is a process followed for a software project, within a software organization. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The Software Development Life Cycle defines a methodology for improving the quality of software and the overall development process.

The following figure is a graphical representation of the various stages of a typical SDLC.



SOFTWARE DEVELOPMENT LIFE CYCLE

# Ch6. SoftwareDevelopment Model

To solve actual problems in an industry, it is imperative to adopt a development strategy that encompasses the process, methods and tool layers. The strategy is incorporated by software engineer and his team and is called Software Development Life Cycle (SDLC). It aims to produce high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

Software Development Process Models are defined and designed and are meant to be followed during the software development process.

Following are the most important and popular SDLC models followed in the industry-

-Waterfall Model

-Iterative Model

-Spiral Model

-RAD Model

-Prototype Model etc.

# Spiral model is chosen for Employee Management System because it provides support for risk management. It is also recommended for complex projects and main advantage of implementing Spiral model is that Change in Requirements at later phase can be incorporated accurately by using this model. In addition to this, it also ensures customer satisfaction as the customer can see the development of the product at the early phase of the software development and thus, they habituated with the system by using it before completion of the total product.

* 1. Structure of Spiral Model:-

Spiral model is a combination of iterative development process model and sequential linear development model i.e. the waterfall model with a very high emphasis on risk analysis. It allows incremental releases of the product or incremental refinement through each iteration around the spiral.

* 1. Spiral Model - Design

The spiral model has four phases. A software project repeatedly passes through these phases in iterations called Spirals.

* + 1. Requirement Gathering:

This phase starts with gathering the business requirements in the baseline spiral.

This phase also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral, the product is deployed in the identified market.

* + 1. Prototyping:

The Prototyping phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design and the final design in the subsequent spirals.

* + 1. Engineering:

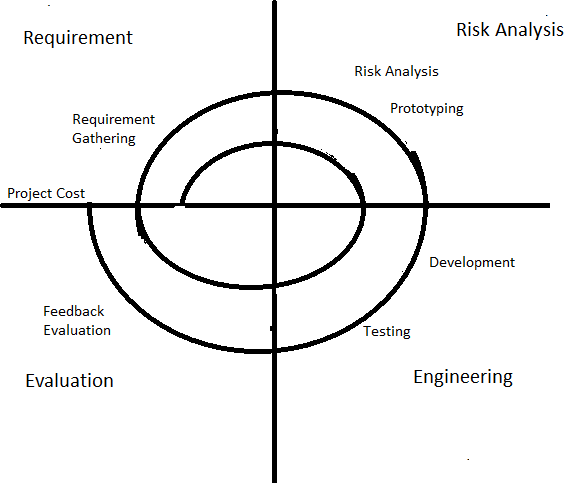
The Engineering phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

* + 1. Evaluation and Risk Analysis:

Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

Spiral Model



* 1. The advantages of the Spiral SDLC Model are as follows -

-Changing requirements can be accommodated.

-Allows extensive use of prototypes.

-Requirements can be captured more accurately.

-Users see the system early.

-Development can be divided into smaller parts and the risky parts can be developed earlier which helps in better risk management.

# Ch.7 Data Dictionary

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **DataType** |
| Username | LOGIN(PK) | varchar(50) |
| Password | LOGIN | varchar(50) |
| Usertype | LOGIN | varchar(50) |
| Question | LOGIN | varchar(50) |
| Answer | LOGIN | varchar(50) |
| EmployeeId | Employee(PK), DepartmentDetails(FK), EmployeeDepartmentDetails(FK) EmployeeJobExperience(FK), EmployeeQualification(FK), EmployeeBankDetails(FK), EmployeePerformance(FK,, EmployeeAppraisa(FK)l, LOGIN(FK),  EmployeeSalary(FK), EmployeeAttendence(FK), Manage Employee(FK), Promote Employee(FK) | varchar(50) |
| EmployeeName | Employee, DepartmentDetails, EmployeeDepartmentDetails EmployeeJobExperience, EmployeeQualification, EmployeeBankDetails, EmployeePerformance, EmployeeAppraisal, LOGIN,  EmployeeSalary, EmployeeAttendence, Manage Employee, Promote Employee | varchar(50) |
| DOB | Employee(NOT NULL) | varchar(50) |
| Age | Employee(NOT NULL) | varchar(50) |
| Gender | Employee | varchar(50) |

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **DataType** |
| MaritalStatus | Employee | varchar(50) |
| Email | Employee(NOT NULL) | varchar(50) |
| EmployeePassword | Employee, LOGIN | varchar(50) |
| ContactNumber | Employee(UNIQUE) | varchar(50) |
| Address | Employee | varchar(150) |
| BloodGroup | Employee(NOT NULL) | varchar(50) |
| Image | Employee(NOT NULL) | image |
| DepartmentName | DepartmentDetails EmployeeDepartmentDetails ManageDepartment EmployeeAppraisal EmployeePerformance EmployeePromotions EmployeeSalary ManageEmployee | varchar(50) |
| DeptId | DepartmentDetails(PK), ManageDepartment(FK), EmployeePromotions(FK) | varchar(50) |
| Designation | EmployeeAppraisal EmployeeDepartmentDetails EmployeePerformance EmployeePromotions EmployeeSalary ManageEmployee EmployeePromotions | varchar(50) |
| Head of Dept | DepartmentDetails ManageDepartment | varchar(100) |
| Date | EmployeeAttendence(date) | varchar(50) |
| TimeIn | EmployeeAttendence | varchar(50) |
| TimeOut | EmployeeAttendence | varchar(50) |
| PresencyRemark | EmployeeAttendence | varchar(50) |
| Firstname | Employee | varchar(50) |
| Lastname | Employee | varchar(50) |

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **DataType** |
| JoiningDate | EmployeeAppraisal(date) | varchar(50) |
| AppraisalSalary | EmployeeAppraisal | numeric(7,4) |
| Appraisalwef | EmployeeAppraisal | varchar(50) |
| AadharNumber | EmployeeBankDetails(PK) | varchar(50) |
| PanNumber | EmployeeBankDetails(UNIQUE) | varchar(50) |
| BankName | EmployeeBankDetails, EmployeeSalary | varchar(50) |
| IFSCCode | EmployeeBankDetails | varchar(50) |
| AccountName | EmployeeBankDetails | varchar(50) |
| AccountNumber | EmployeeBankDetails EmployeeSalary | varchar(50) |
| Shift | EmployeeDepartmentDetails | varchar(50) |
| Status | EmployeeDepartmentDetails ManageEmployee | varchar(50) |
| WorkExperience | EmployeeJobExperience | varchar(50) |
| PastOrganisation | EmployeeJobExperience | varchar(50) |
| PastDepartment | EmployeeJobExperience | varchar(50) |
| PastDesignation | EmployeeJobExperience | varchar(50) |
| PastSalary | EmployeeJobExperience | numeric(7,4) |
| Month | EmployeePerformance EmployeeSalary | varchar(50) |
| TargetSet | EmployeePerformance | varchar(50) |
| Absenties | EmployeePerformance | varchar(50) |
| TargetAchieved | EmployeePerformance | varchar(50) |
| ConnectionsMade | EmployeePerformance | varchar(50) |
| ClientFeedback | EmployeePerformance | varchar(50) |
| ClientRetained | EmployeePerformance | varchar(50) |
| PromotedDepartmentName | EmployeePromotion | varchar(50) |

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **DataType** |
| PromotedDesignation | EmployeePromotion | varchar(50) |
| PromotedSalary | EmployeePromotion | numberic(7,4) |
| SrNo | DepartmentDetails(PK) | int |
| SerNo | EmployeeJobExperience(PK) | int |
| SSCBoardName | EmployeeQualification | varchar(100) |
| SSCYOP | EmployeeQualification | varchar(10) |
| SSCPercentage | EmployeeQualification | numeric(4,2) |
| HSCBoardName | EmployeeQualification | varchar(100) |
| HSCYOP | EmployeeQualification | varchar(10) |
| HSCPercentage | EmployeeQualification | numeric(4,2) |
| UGCourseName | EmployeeQualification | varchar(100) |
| UGUniversity | EmployeeQualification | varchar(100) |
| UGYOP | EmployeeQualification | varchar(10) |
| UGPercentage | EmployeeQualification | numeric(4,2) |
| PGCourseName | EmployeeQualification | varchar(100) |
| PGUniversity | EmployeeQualification | varchar(100) |
| PGYOP | EmployeeQualification | varchar(10) |
| PGPercentage | EmployeeQualification | numeric(4,2) |
| Year | EmployeeSalary | varchar(50) |
| BasicSalary | EmployeeSalary | numeric(7,2) |
| DA | EmployeeSalary | numeric(7,2) |
| HRA | EmployeeSalary | numeric(7,2) |
| MA | EmployeeSalary | numeric(7,2) |
| SP | EmployeeSalary | numeric(7,2) |
| GrossSalary | EmployeeSalary | numeric(7,2) |
| PF | EmployeeSalary | numeric(7,2) |

|  |  |  |
| --- | --- | --- |
| **Field Name** | **Table Name** | **DataType** |
| IT | EmployeeSalary | numeric(7,2) |
| PT | EmployeeSalary | numeric(7,2) |
| ESIC | EmployeeSalary | numeric(7,2) |
| Others | EmployeeSalary | numeric(7,2) |
| Deductions | EmployeeSalary | numeric(7,2) |
| NetPay | EmployeeSalary | numeric(7,2) |

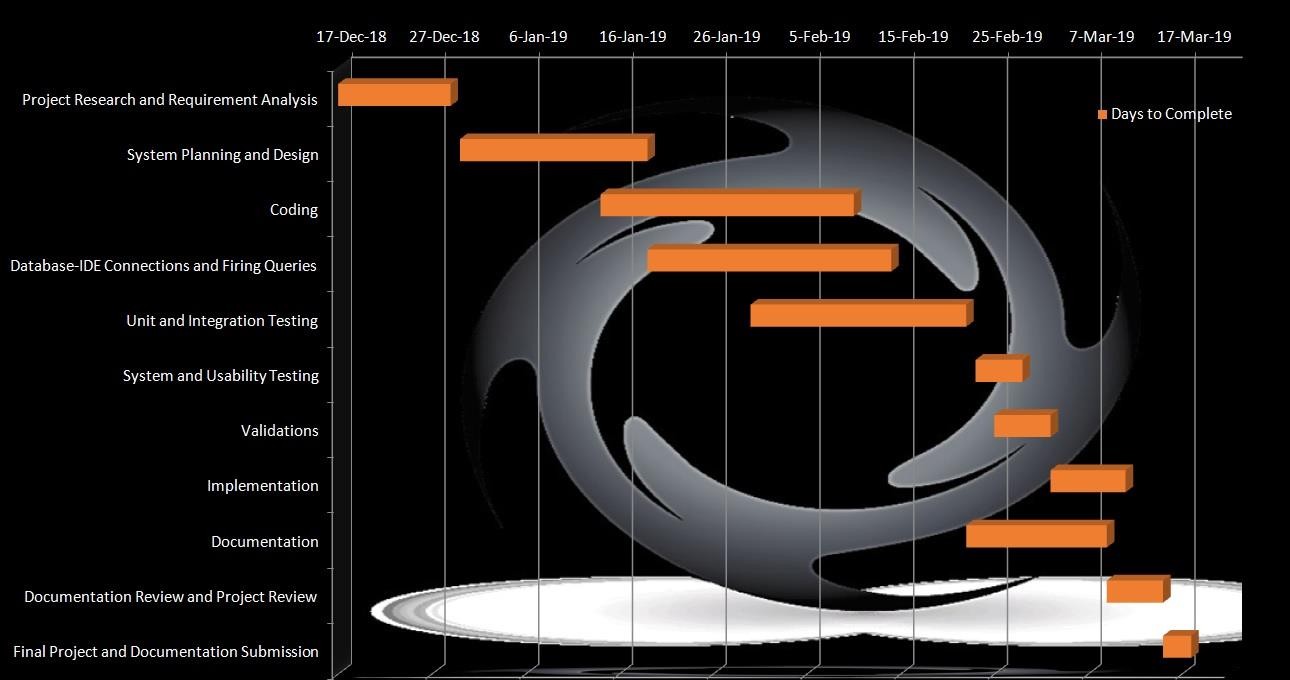
**Ch.8 Gantt Chart**

Gantt chart provides a graphical illustration of a schedule that helps to plan, coordinate, and track specific tasks in a project.

A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (months) and a vertical axis representing the tasks that make up the project: **Employee Management Sytsem**.

It depicts our Project Development Road Map based on task scheduling right from Preliminary Research upto Project Submission. Timelines are indicated that show the time required for the completion of each and every module. It gives a clear idea as to how our project development is phased out on modules to summarise the entire Project Assessment.

# Gantt Chart



This Gantt chart illustrates the Phases in which the “Employee Management System” Project was carried out. It gives a systematic overview as to how each task was scheduled and the time required to complete the task. Thus, it summarises the entire Project Assessment.

# Ch.9 UML DIAGRAMS

The Unified Modelling Language is a standard visual modelling language intended to be used for modelling business and similar processes, analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behaviour of arti-facts of software systems.

UML defines various kinds of diagrams to cover most of the aspects of a system. There are two broad categories of diagrams and they are again divided into subcategories −

# Structural Diagrams:

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

* + 1. Class diagram
    2. Object diagram
    3. Component diagram
    4. Deployment diagram

# Behavioural Diagrams:

Behavioural diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioural diagrams:

1. Use-Case Diagram
2. Sequence Diagram
3. Collaboration Diagram
4. State-Chart Diagram
5. Activity Diagram

# E-R Diagram

The ER or (Entity Relational Model) is a high-level conceptual data model diagram. Entity- Relation model is based on the notion of real-world entities and the relationship between them.

ER modelling helps to analyze data requirements systematically to produce a well-designed database. So, it is ideal to complete ER modelling before implementing your database.

Entity relationship diagram displays the relationships of entity set stored in a database. This model is based on three basic concepts:

1. **Entities:** A real-world thing either living or nonliving that is easily recognizable and unrecognizable.
2. **Attributes:** It is a single-valued property of either an entity-type or a relationship- type. However there exist multi-valued attributes too.
3. **Relationships**: Relationship is nothing but an association among two or more entities.
   * 1. **Cardinalities:**

Cardinality defines the number of entities in one entity set, which can be associated with the number of entities of other set via relationship set.

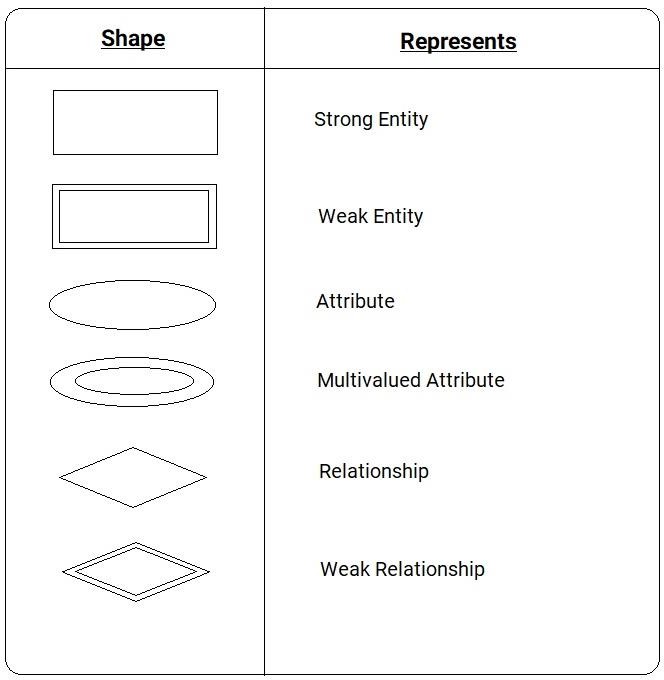
One-to-One − One entity from entity set A can be associated with at most one entity of entity set B and vice versa.

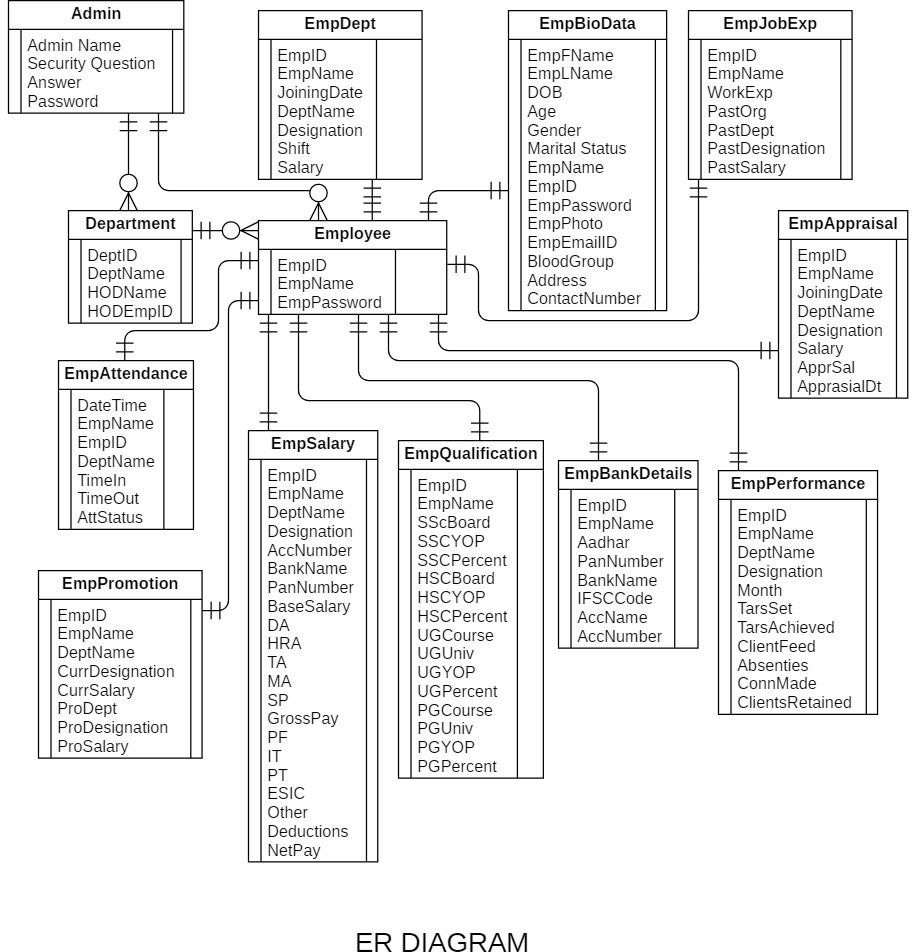
One-to-Many − One entity from entity set A can be associated with more than one entities of entity set B however an entity from entity set B, can be associated with at most one entity.

Many-to-One − More than one entities from entity set A can be associated with at most one entity of entity set B, however an entity from entity set B can be associated with more than one entity from entity set A.

Many-to-Many − One entity from A can be associated with more than one entity from B and vice versa.

# ER DIAGRAM NOTATIONS





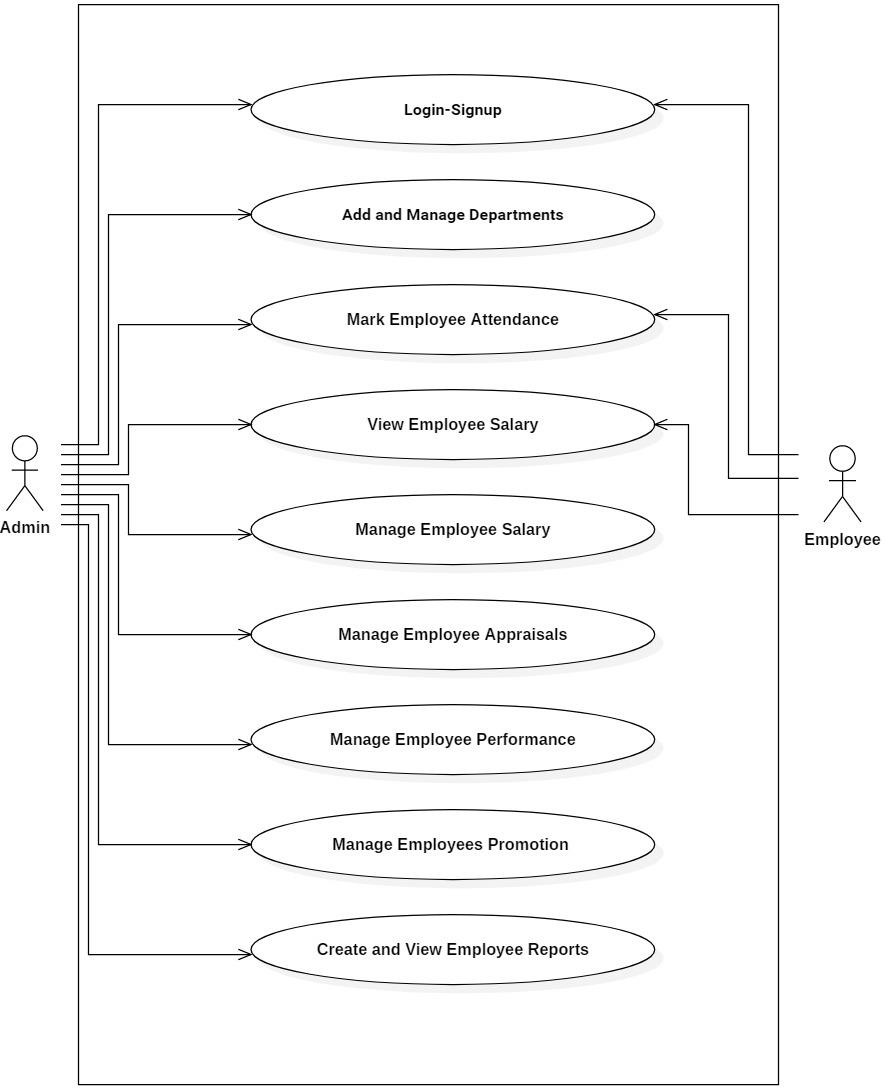
* 1. **Use-Case Diagram**

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as actors.

When the requirements of a system are analyzed, the functionalities are captured in use cases.

UseCase Diagram:



# Class Diagram

Class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

A UML class diagram is made up of:

1. Set of classes
2. Set of relationships between classes

A class notation consists of three parts:

1. Class Name: The name of the class appears in the first partition.
2. Class Attributes:

Attributes are shown in the second partition. The attribute type is shown after the colon.

Attributes map onto member variables (data members) in code.

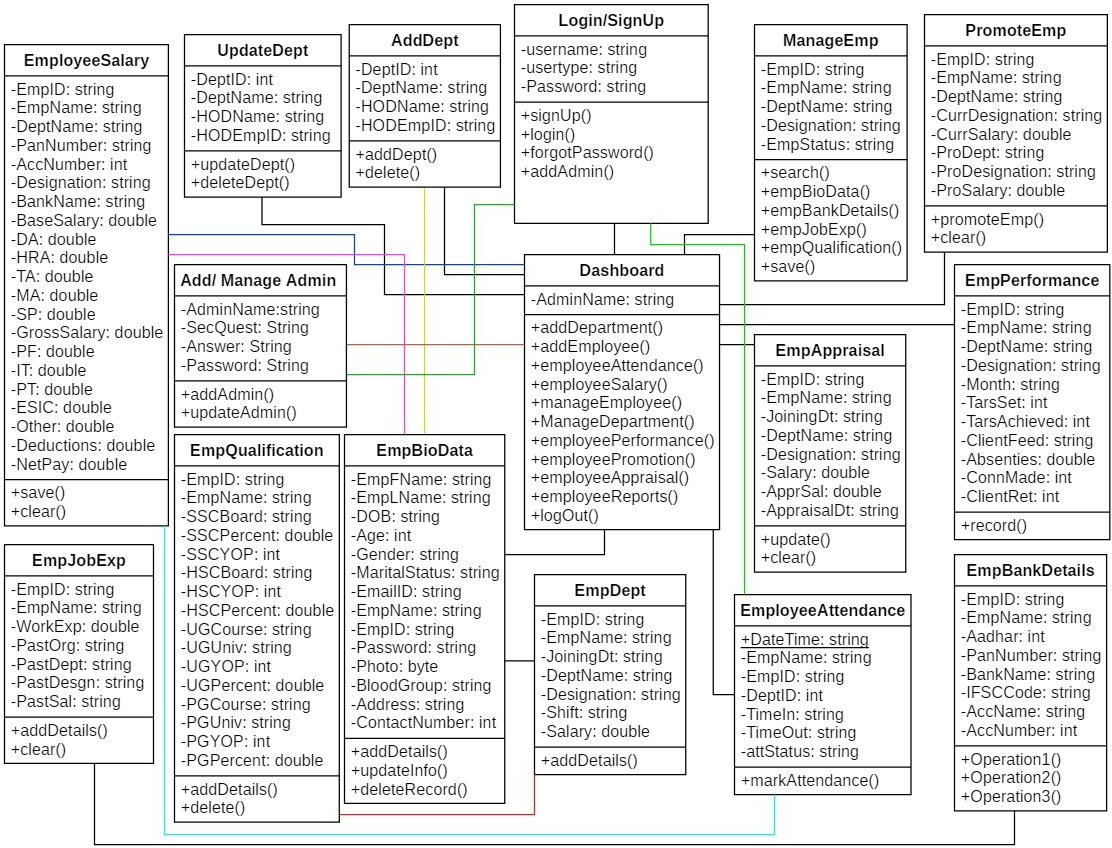
1. Class Operations (Methods)

Operations are shown in the third partition. These are the services a class provides.

The return type of a method is shown after the colon at the end of the method signature.

The return type of method parameters are shown after the colon following the parameter name.

Class Diagram:

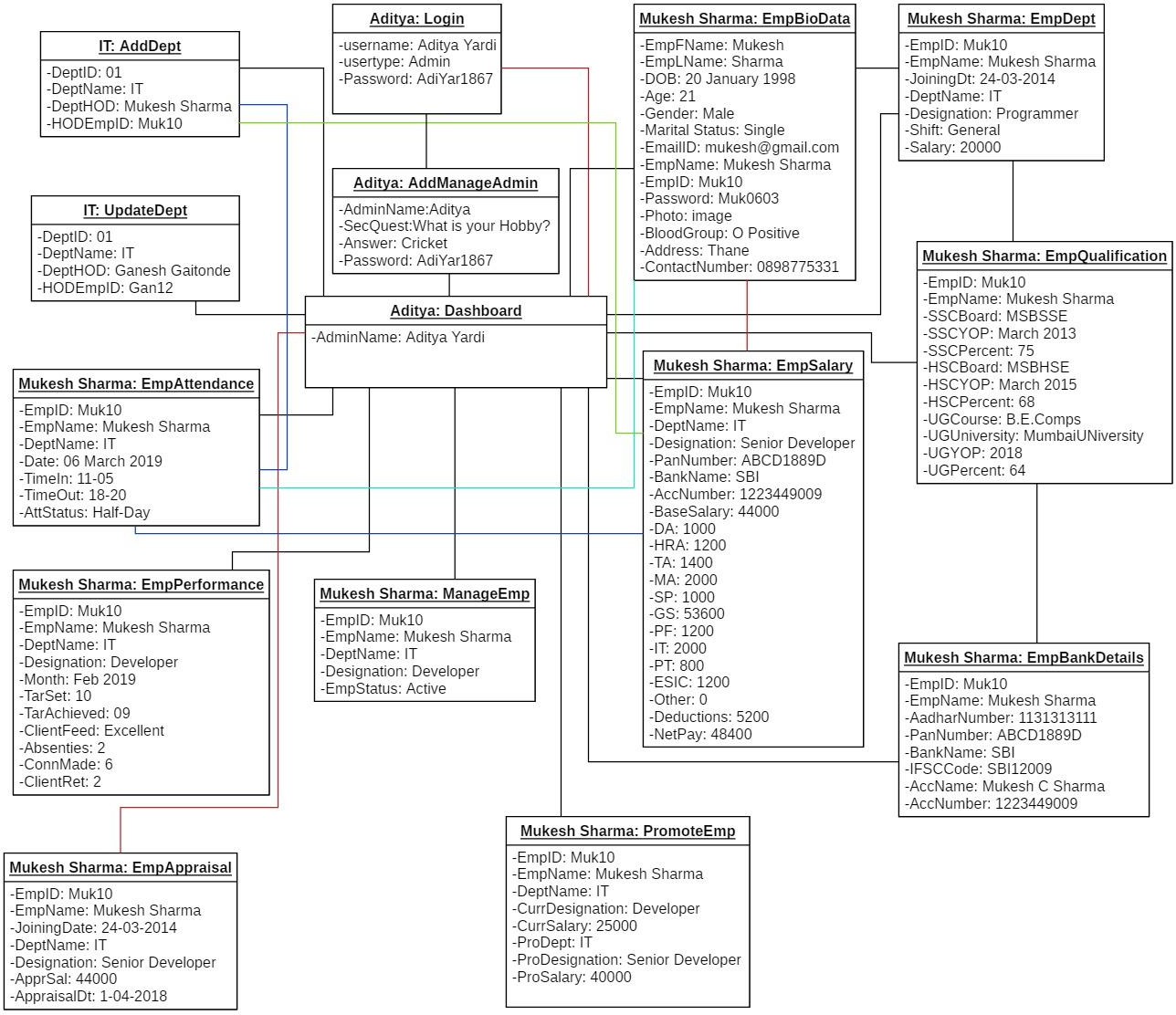


# Object Diagram

Object is an instance of a particular moment in runtime, including objects and data values. A static UML object diagram is an instance of a class diagram; it shows a snapshot of the detailed state of a system at a point in time, thus an object diagram encompasses objects and their relationships at a point in time. It may be considered a special case of a class diagram or a communication diagram.

Object Diagram is used to verify the accuracy and completeness of the class diagram. An object diagram shows this relation between the instantiated classes and the defined class, and the relation between these objects in the system. They are useful to explain smaller portions of your system, when the system class diagram is complex.

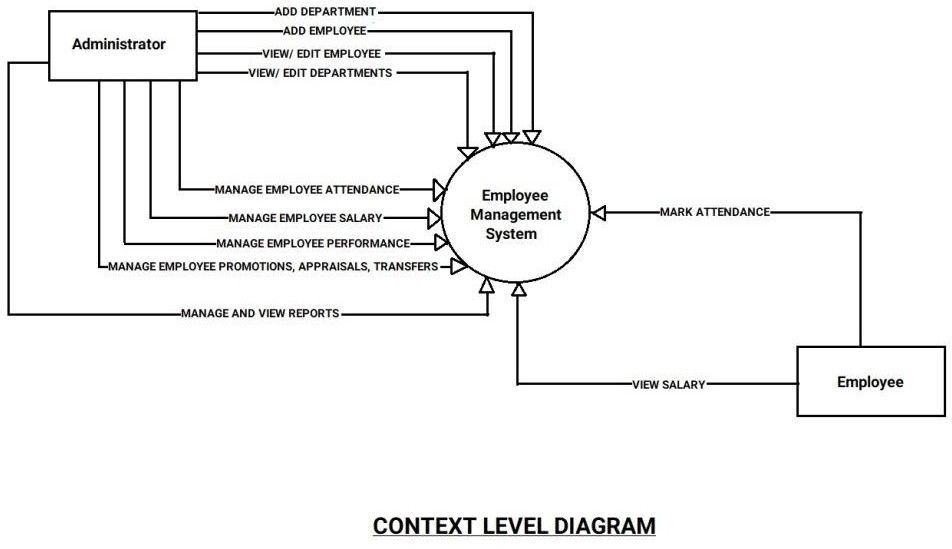
Object Diagram



# Context Level Diagram

A context data flow diagram (DFD), also known as a level 0 DFD, gives a broad overview of an information system and the way it interacts with external entities. Being the highest level in a data flow diagram, it contains only one process representing the entire system.

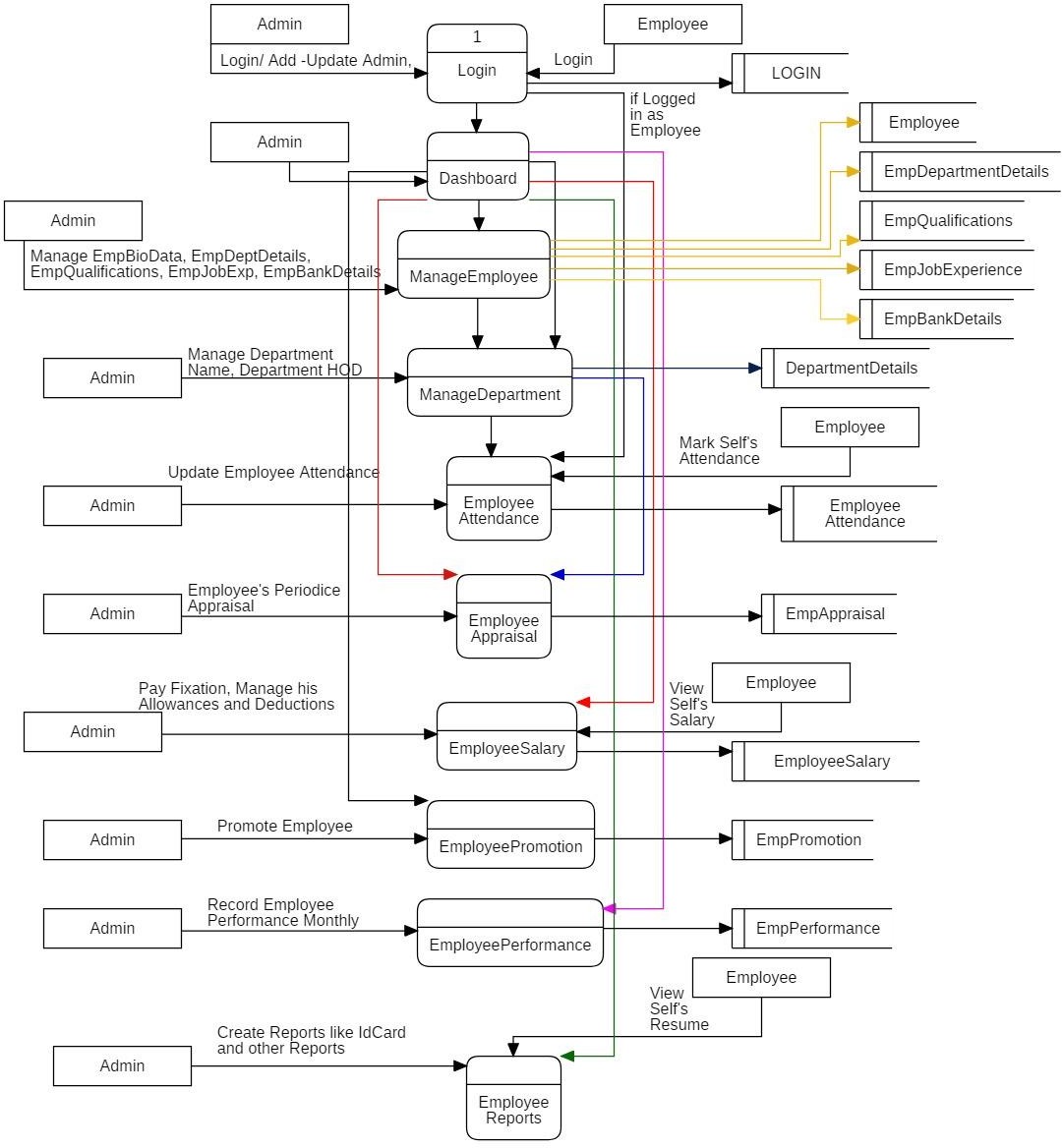
Context Level Diagram describes the overview functionalities required by the external entities; it can be decomposed into a number of sub-level DFDs in hierarchical manner.



# Data Flow Diagram

Data Flow Diagrams (DFD) are used to graphically represent the flow of Data in a business information system. DFD describes the processes that are involved in a system to transfer data from the input to the file storage and reports generation. A Data Flow Diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles, arrows, etc. to show data inputs, outputs, storagepoints and the routes between each destination.

Data Flow Diagram

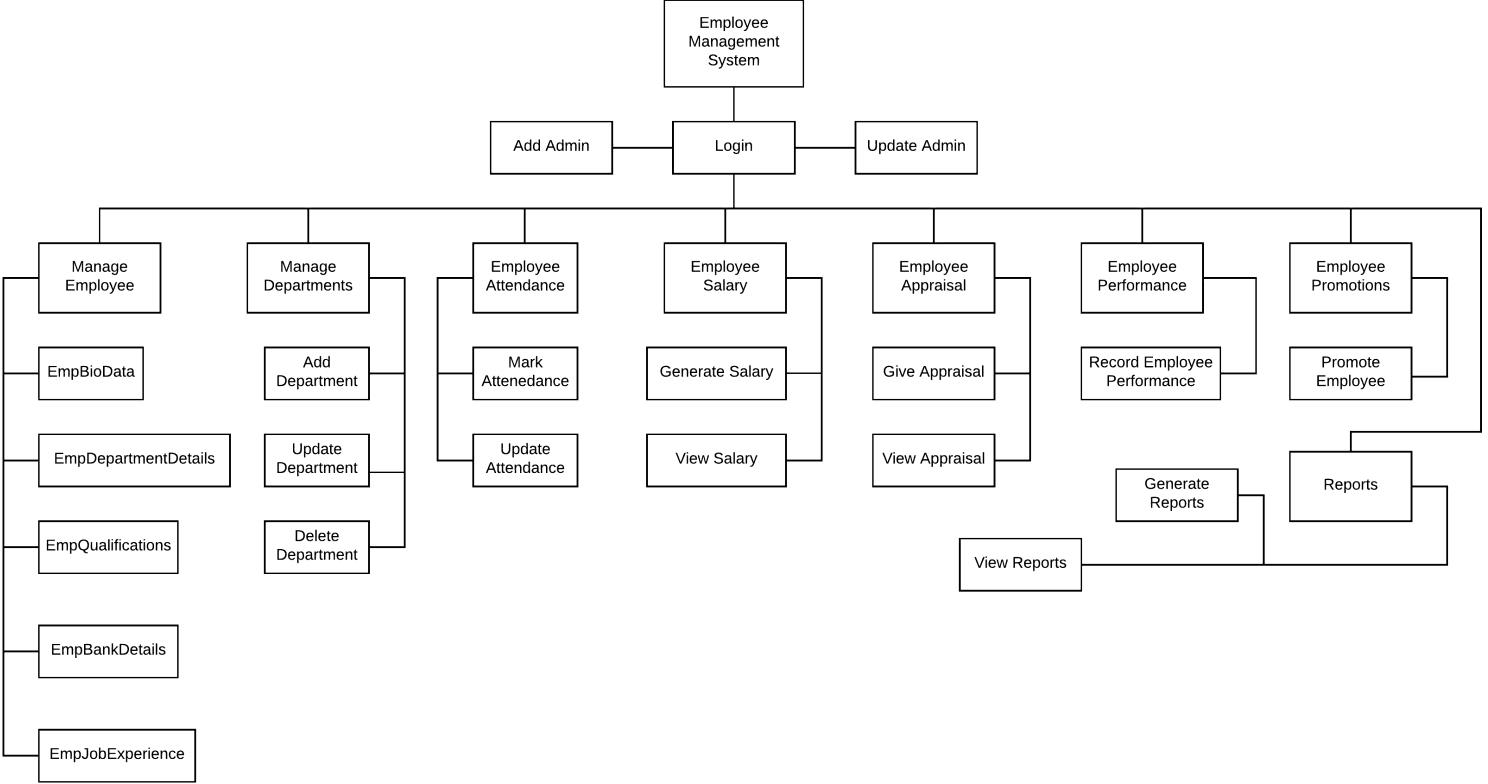


# Functional Decomposition Diagram

FunctionalDecompositionDiagram (FFD corresponds to the various functional relationships as how the original complex business function was developed. It mainly focusses on how the overall functionality is developed and its interaction between various components.

The purpose of the functional decomposition diagram is to show on a single page the capabilities of an organization that are relevant to the consideration of an architecture. By examining the capabilities of an organization from a functional perspective, it is possible to quickly develop models of what the organization does without being dragged into an extended debate on how the organization does it. Once a basic functional decomposition diagram has been developed, it becomes possible to layer heat-maps on top of this diagram to show scope and decisions.

FunctionalDecompositionDiagram



# Sequence Diagram

A sequence diagram simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. Sequence diagrams describe how and in what order the objects in a system function.

These diagrams are widely used by businessmen and software developers to document and understand requirements for new and existing systems.

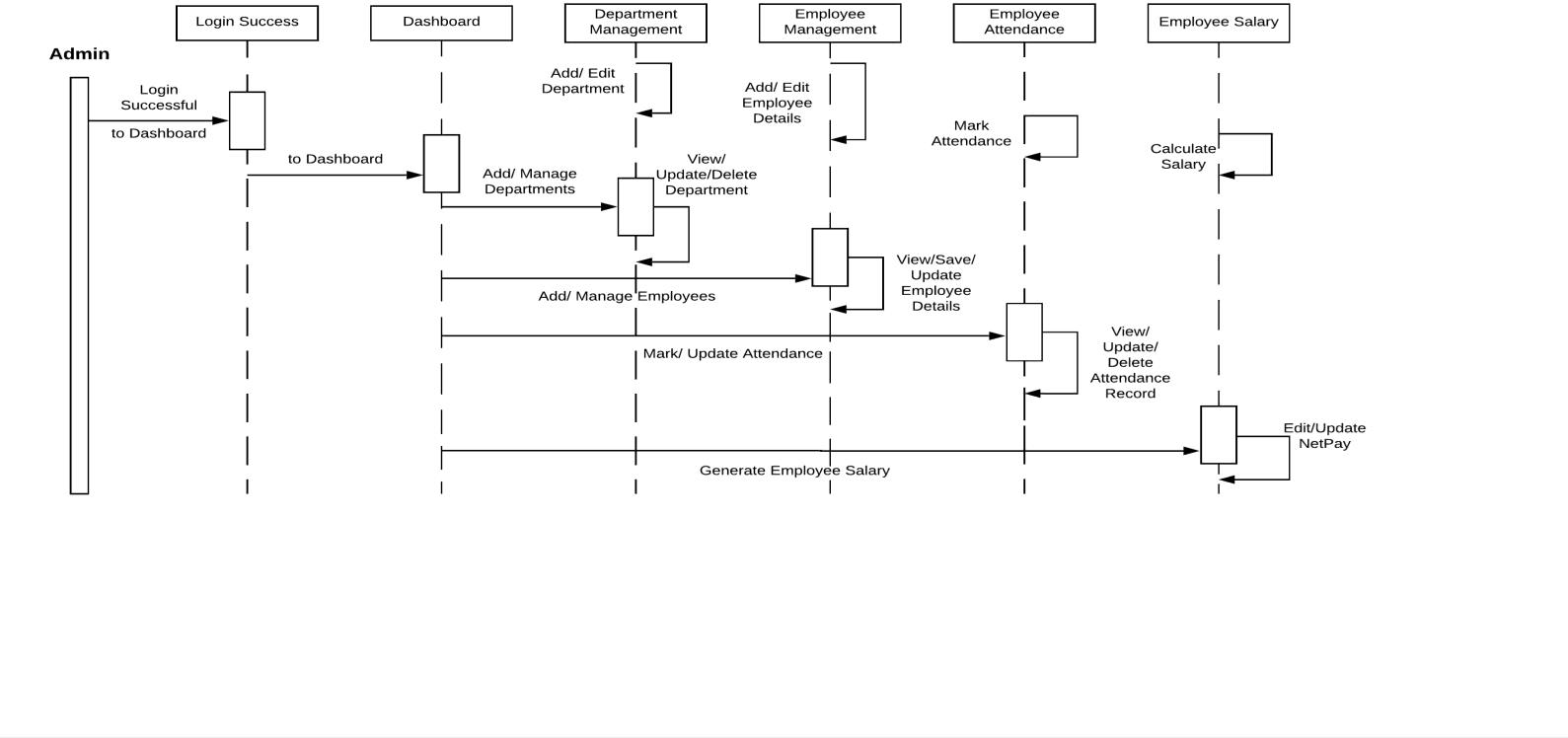
* + 1. Sequence Diagram Notations:

**Actors–**An actor in a UML diagram represents a type of role where it interacts with the system and its objects

**Lifelines –** A lifeline is a named element which depicts an individual participant in a sequence diagram.

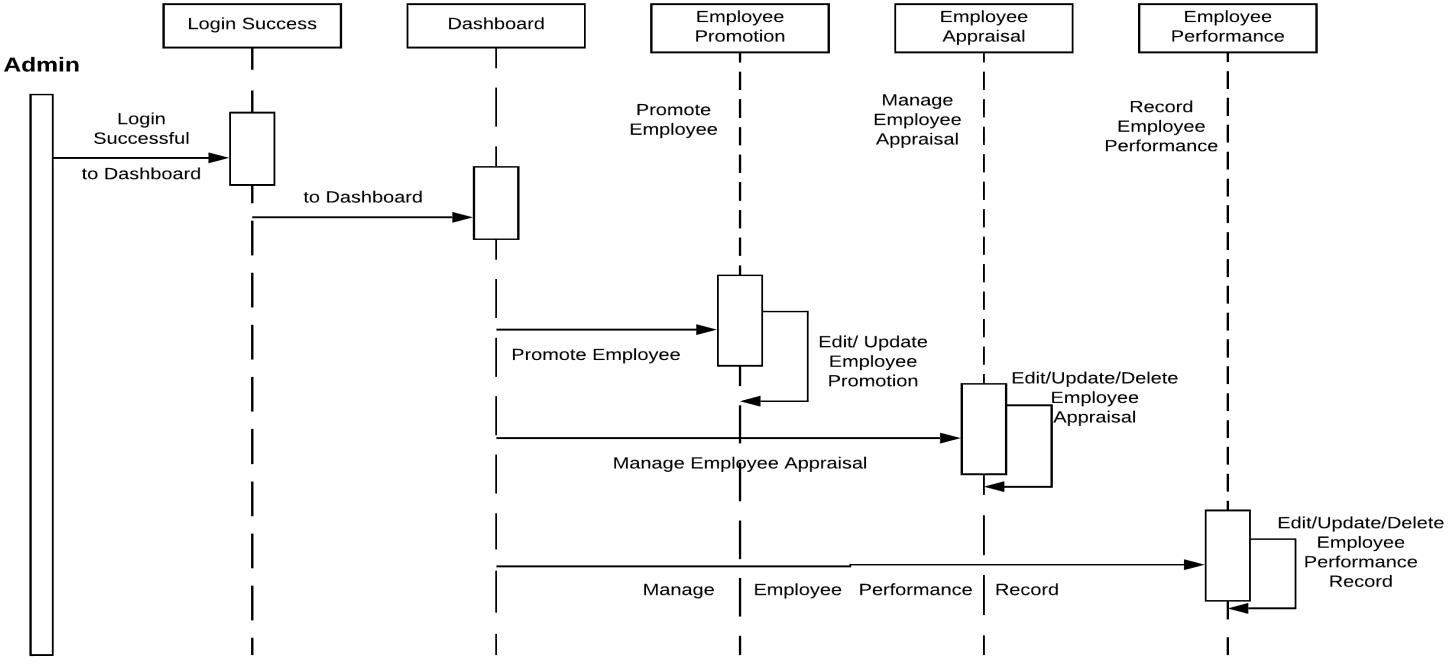
**Messages –** Communication between objects is depicted using messages. The messages appear in a sequential order on the lifeline.

* + 1. Use of Sequence Diagrams:
       1. Used to model and visualise the logic behind a sophisticated function, operation or procedure.
       2. They are also used to show details of UML use case diagrams.
       3. Used to understand the detailed functionality of current or future systems.
       4. Visualise how messages and tasks move between objects or components in a system.

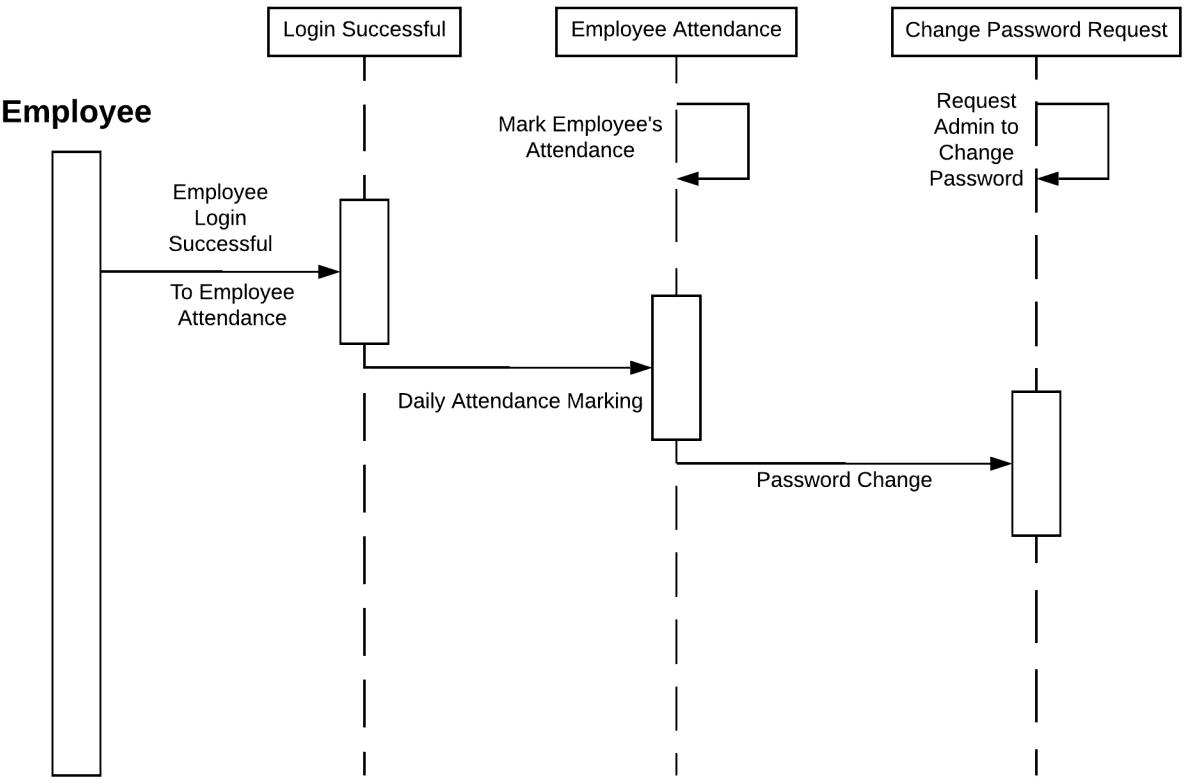


Admin Sequence Diagram (1):

Admin Sequence Diagram (2):



Employee Sequence Diagram:



# Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system.

The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc

Activity is a particular operation of the system. They are used to construct the executable system by using forward and reverse engineering techniques.

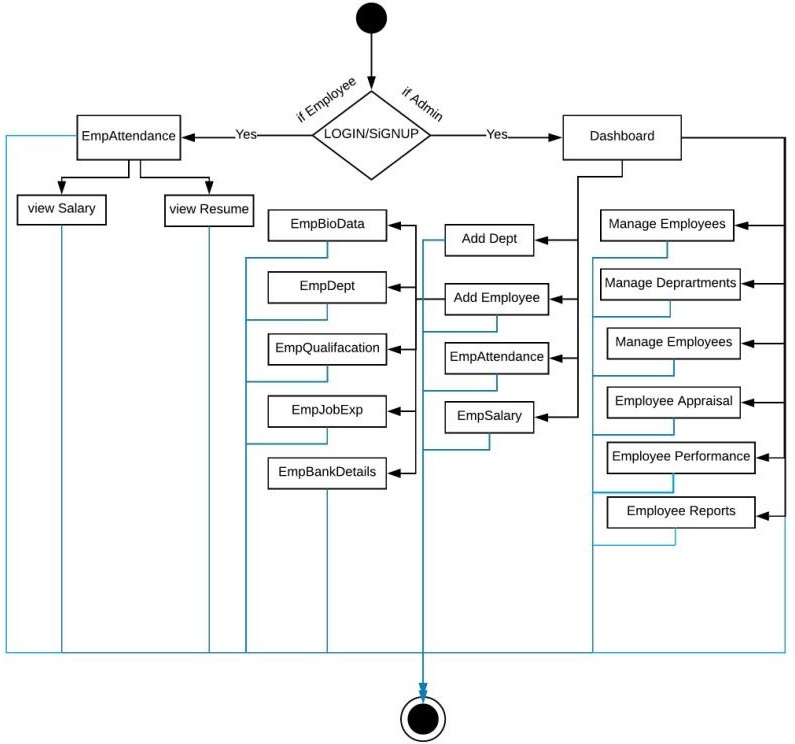
* + 1. The purpose of an activity diagram is:
       1. To Draw the activity flow of a system.
       2. Describe the sequence from one activity to another.
       3. Describe the parallel, branched and concurrent flow of the system.
    2. Activity Diagram consists of following elements −
       1. Activities
       2. Association
       3. Conditions
       4. Constraints

Activity diagram is suitable for modeling the activity flow of the system. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

Activity diagram can be used for −

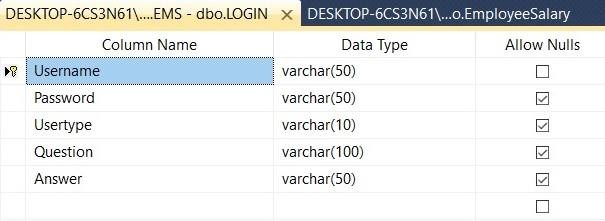
1. Modeling workflow by using activities.
2. Modeling business requirements.
3. High level understanding of the system's functionalities.
4. Investigating business requirements at a later stage.

Activity Diagram

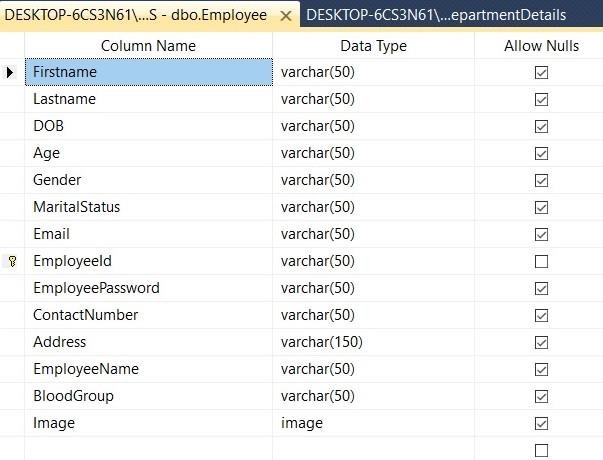


# Ch.10 Sql DataStructures Screen-Shots

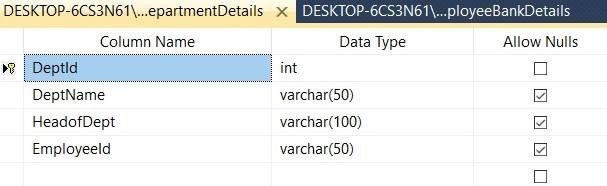
* 1. Login:



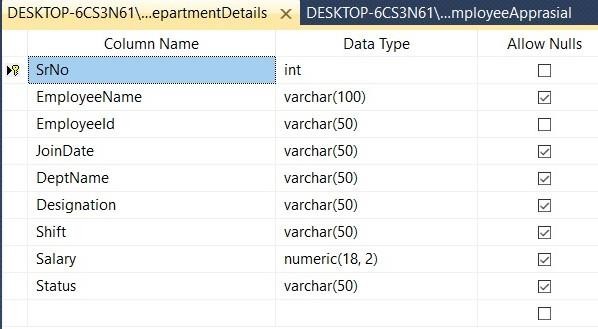
* 1. Employee:



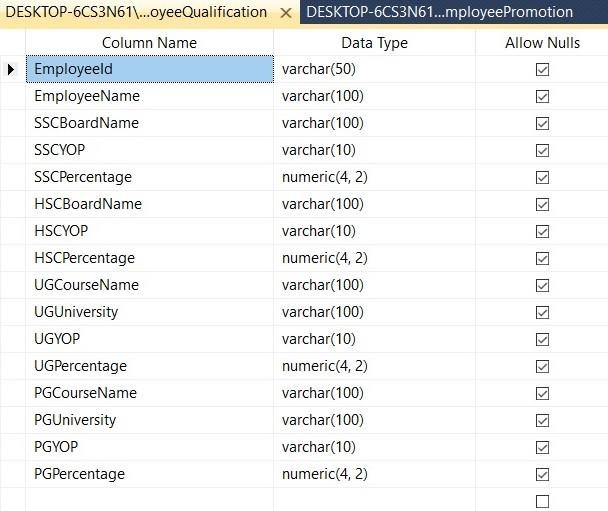
* 1. DepartmentDetails:



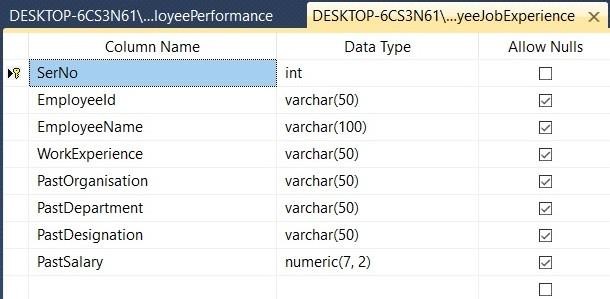
* 1. EmpDepatmentDetails:



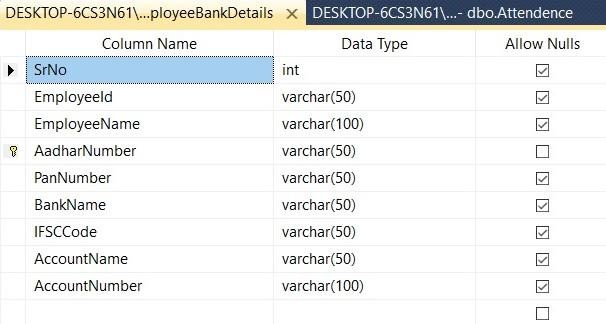
* 1. EmpQualification:



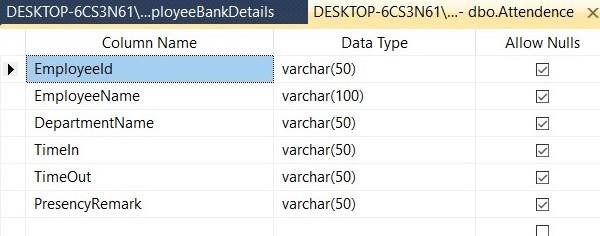
* 1. Employee JobExperience:



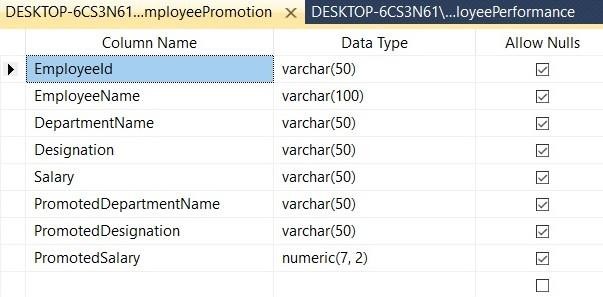
* 1. Employee Bank Details:



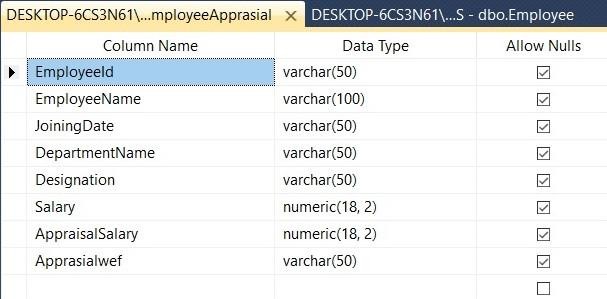
* 1. EmployeeAttendance:



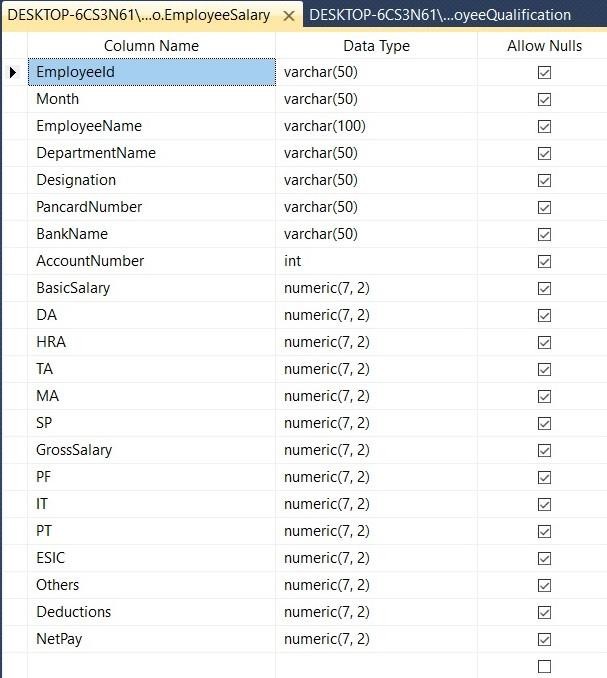
* 1. Employee Promotion:



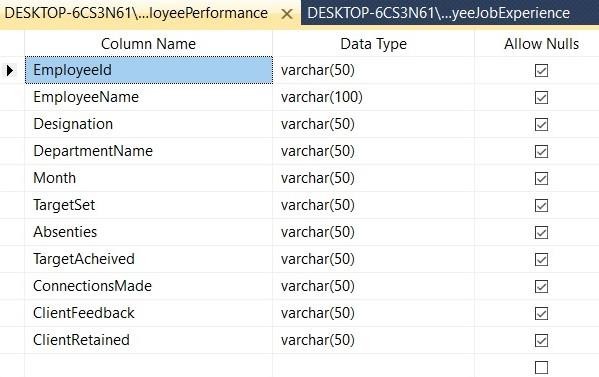
* 1. Employee Appraisal:



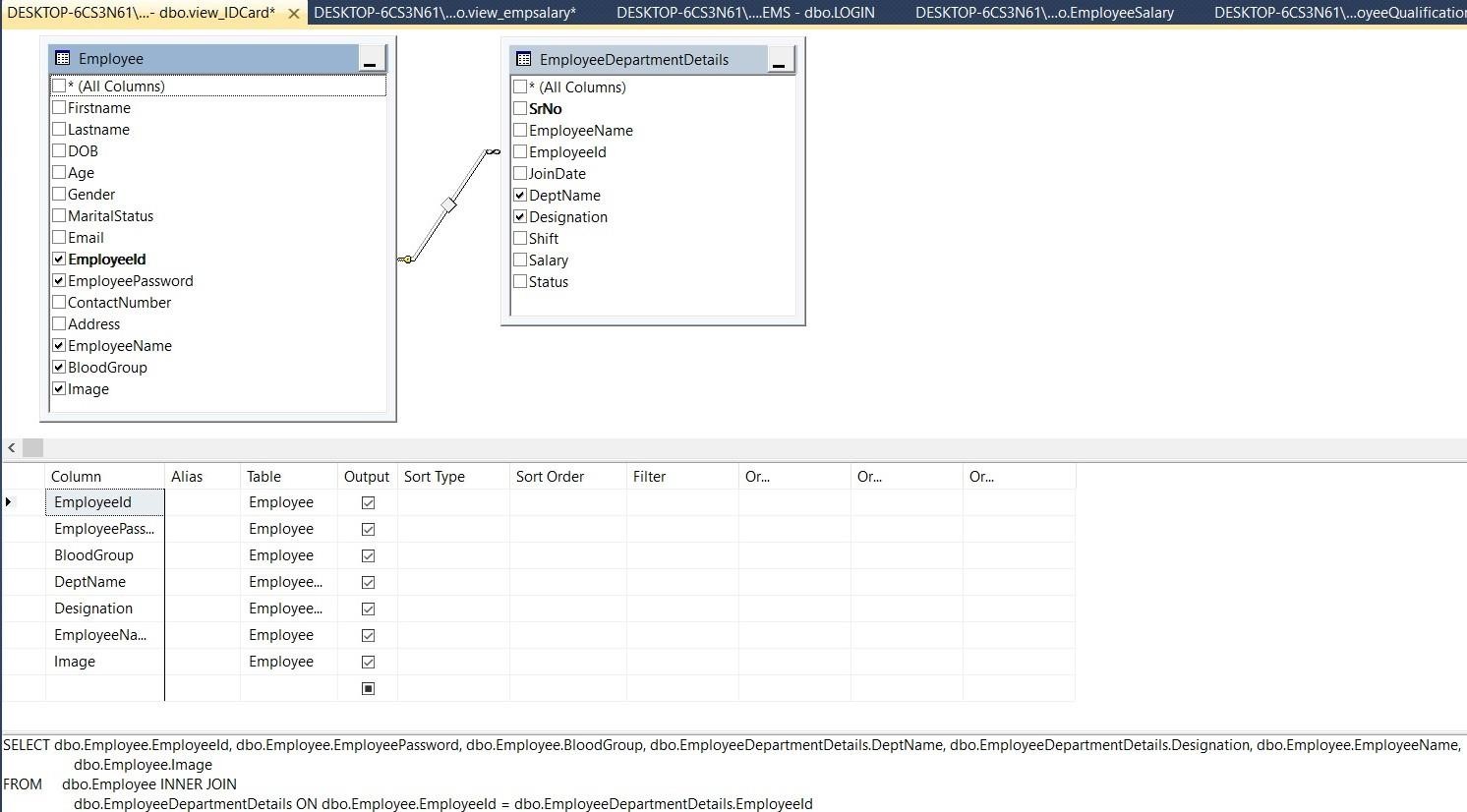
* 1. Employee Salary:



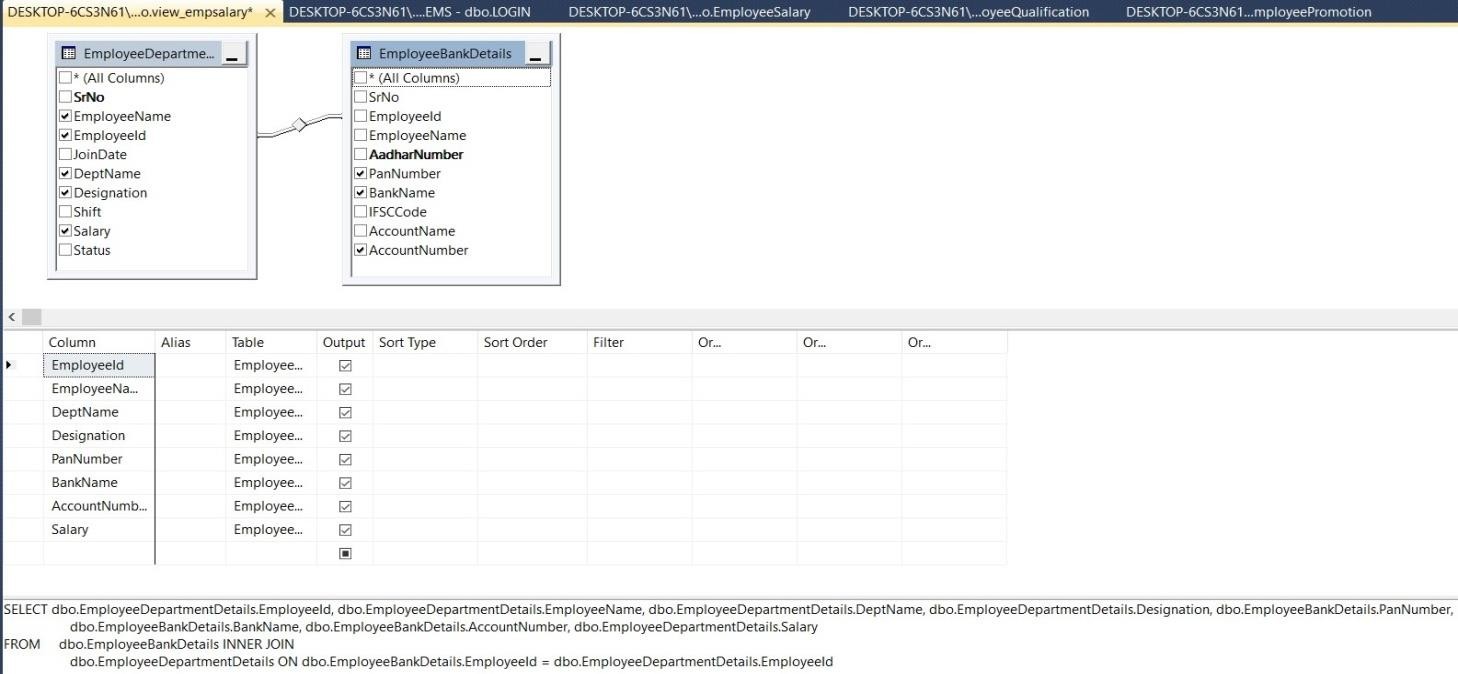
* 1. EmployeePerformance:



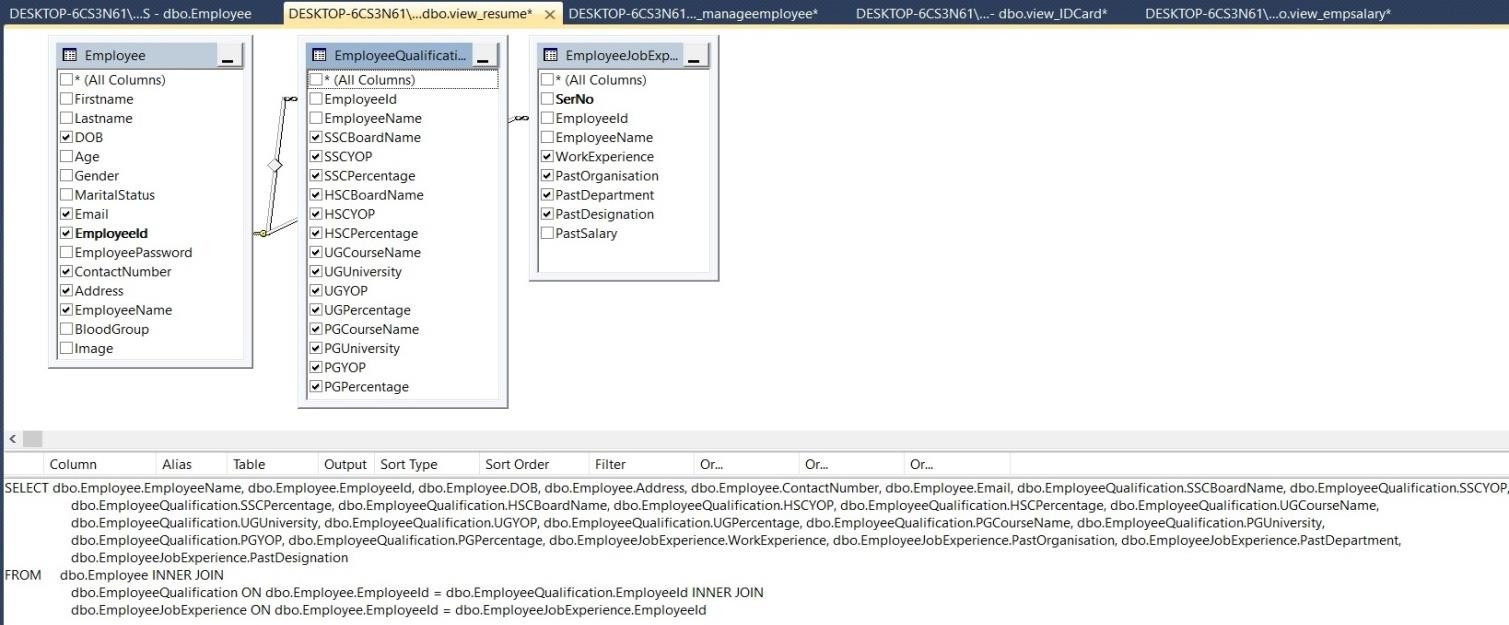
* 1. ID CARD VIEW:



* 1. EmployeeSalaryView:

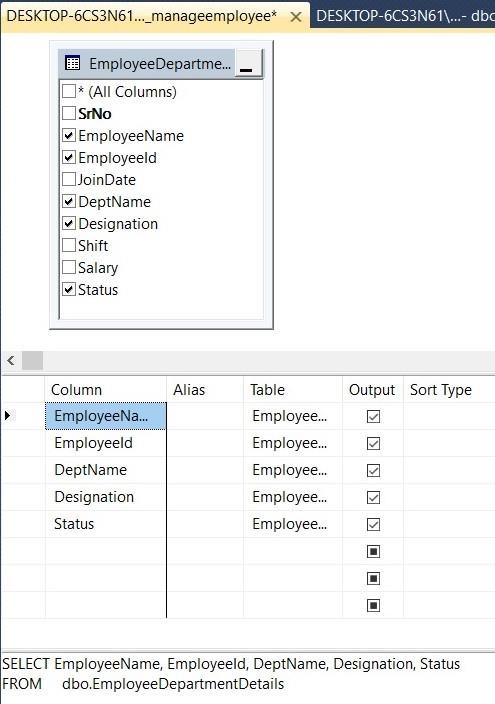


* 1. ResumeView:



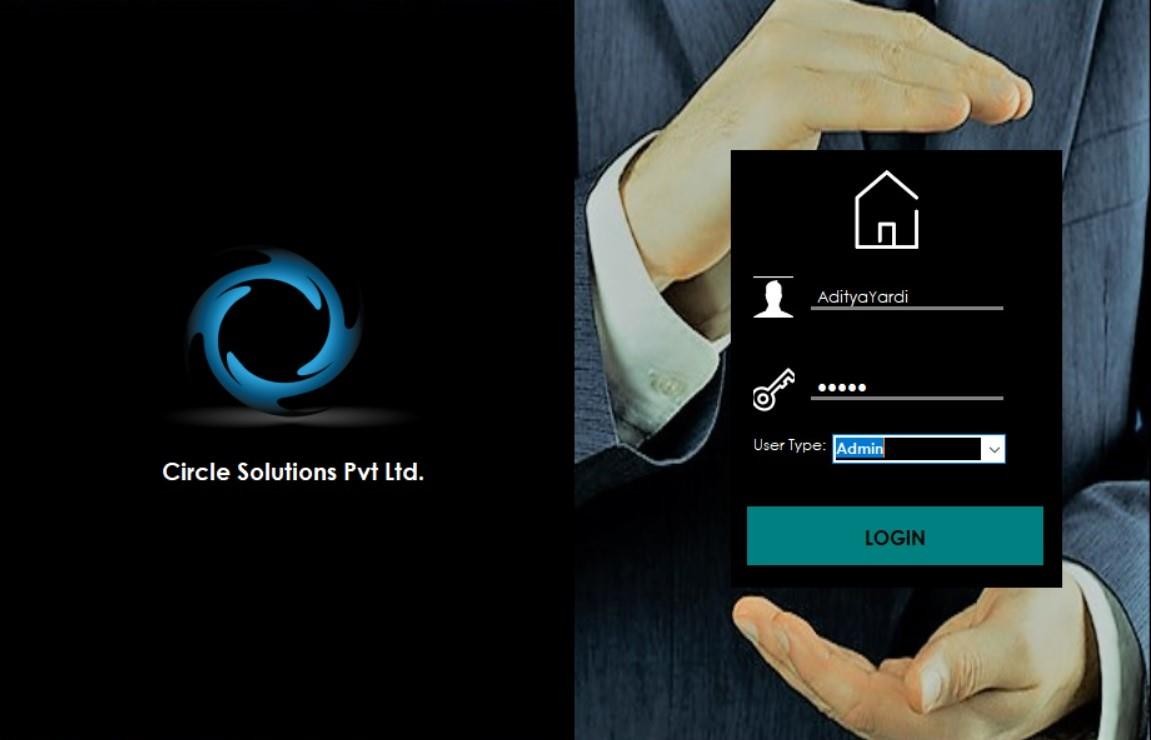
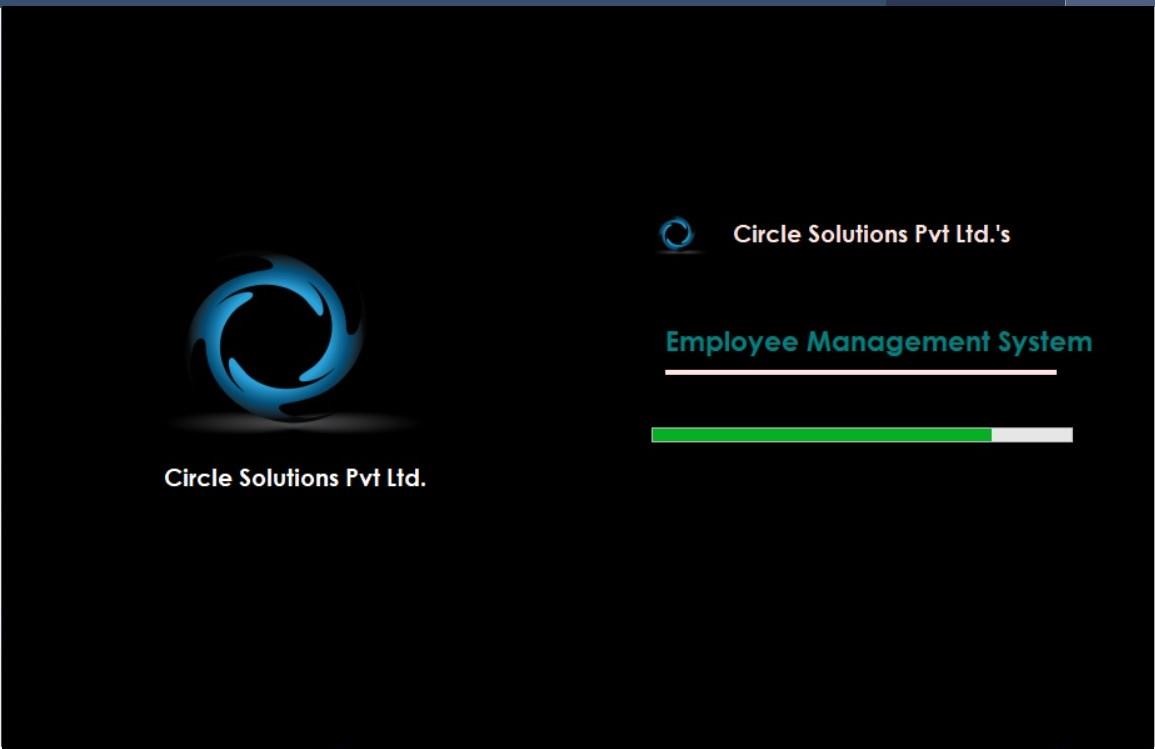
* 1. ManageEmployeeView:

50

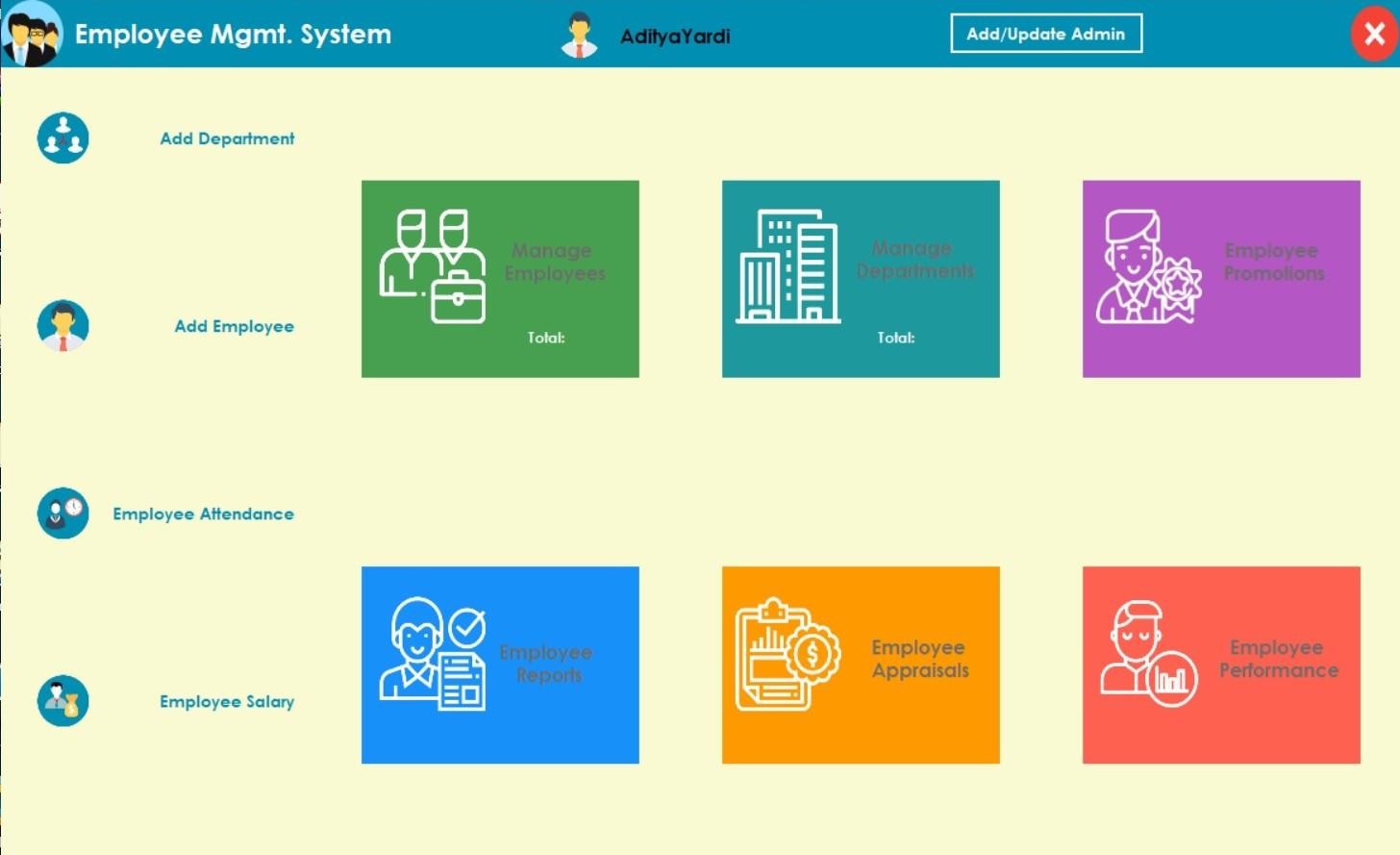


# Ch.11 Project ScreenShots

* 1. Log-In

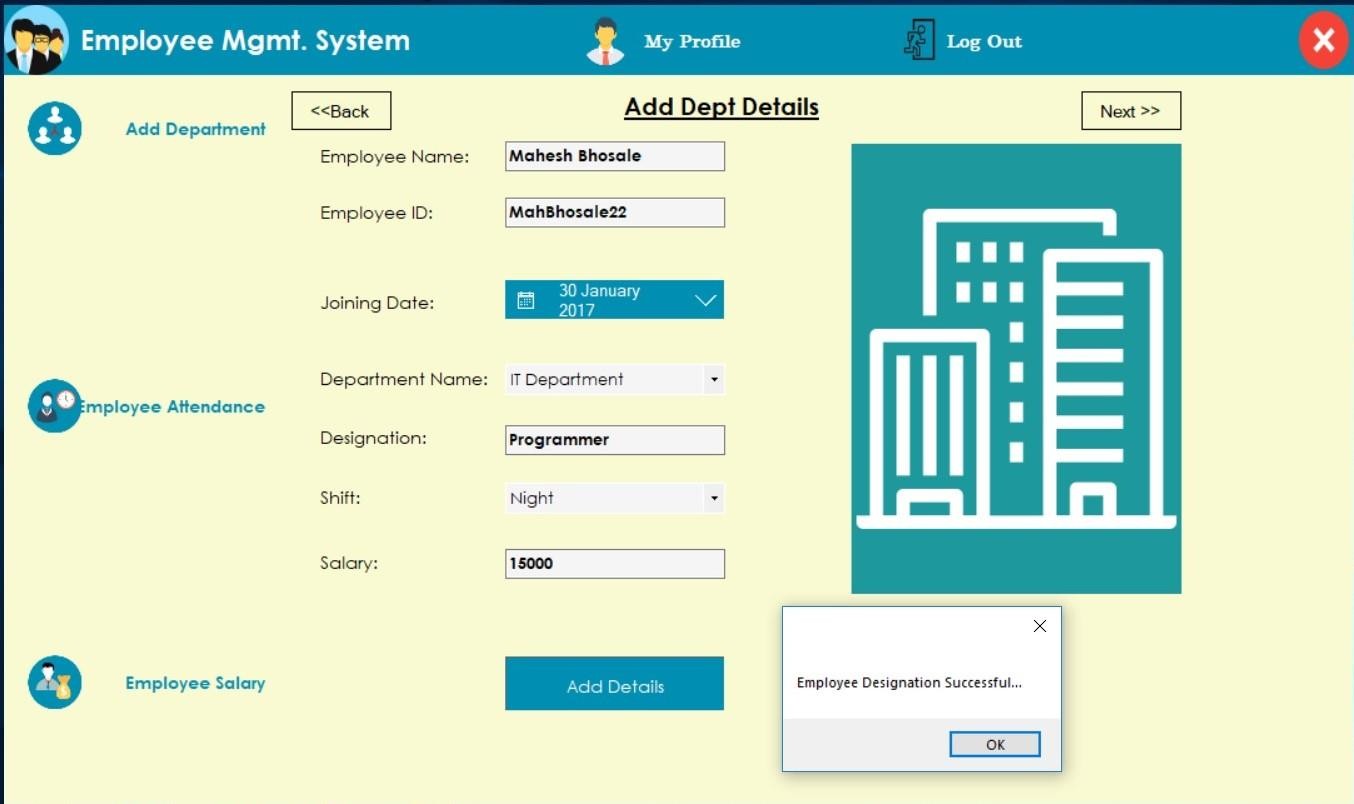


* 1. DashBoard



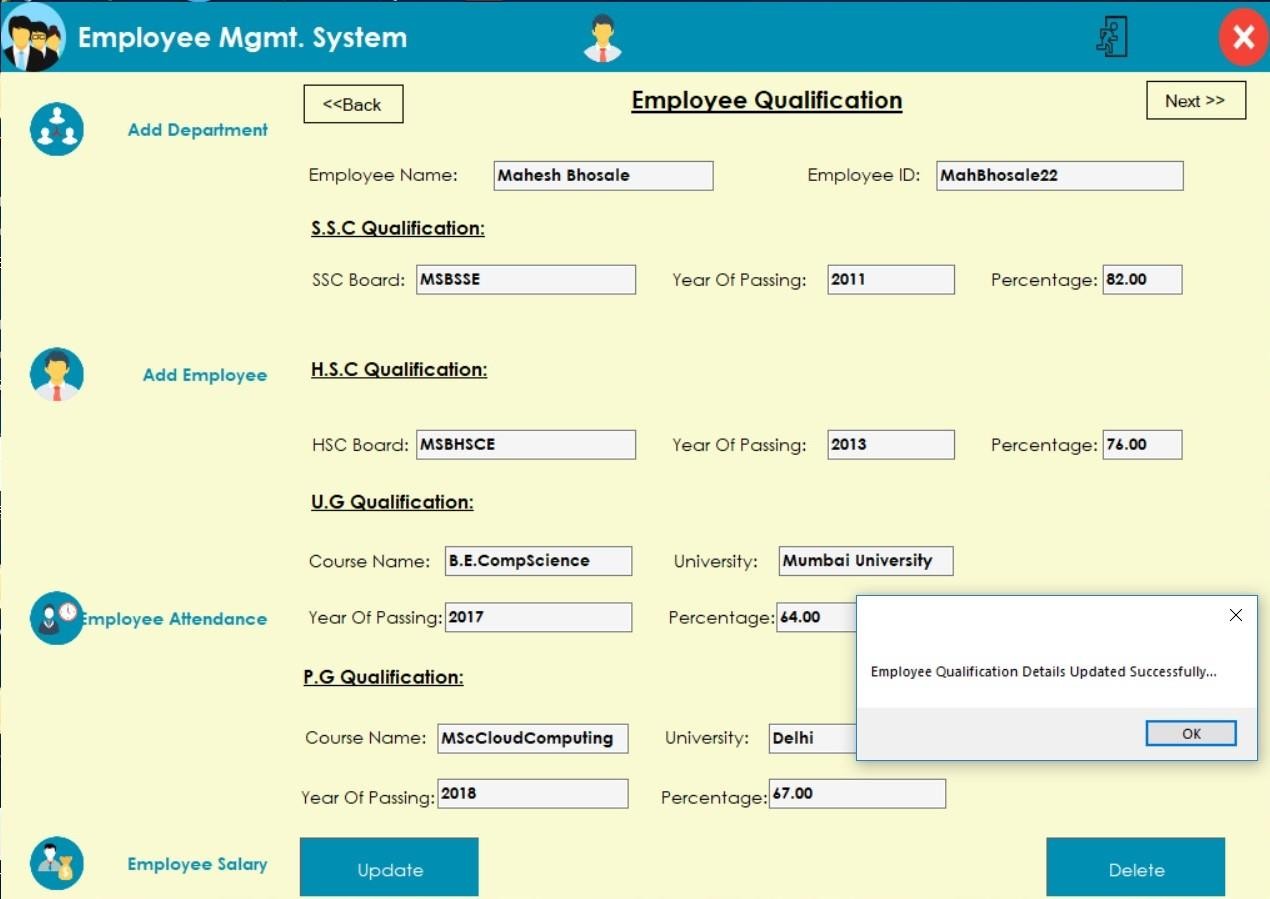
Admin Dashboard: Admin(s) can navigate freely to perform any operation in the System.

* 1. EmpDeptDetails



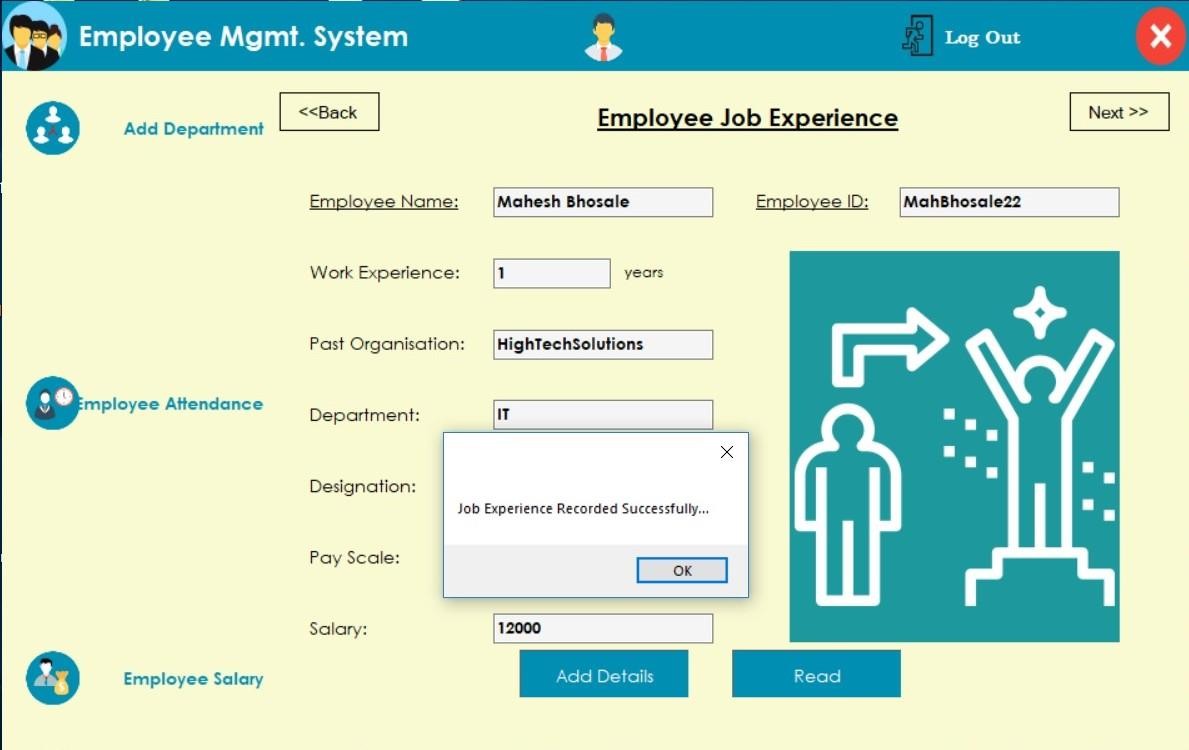
Admin can allocate the Employee to the Department, Assign him a post in department, decide his Shift and fix his salary.

* 1. EmpQualification



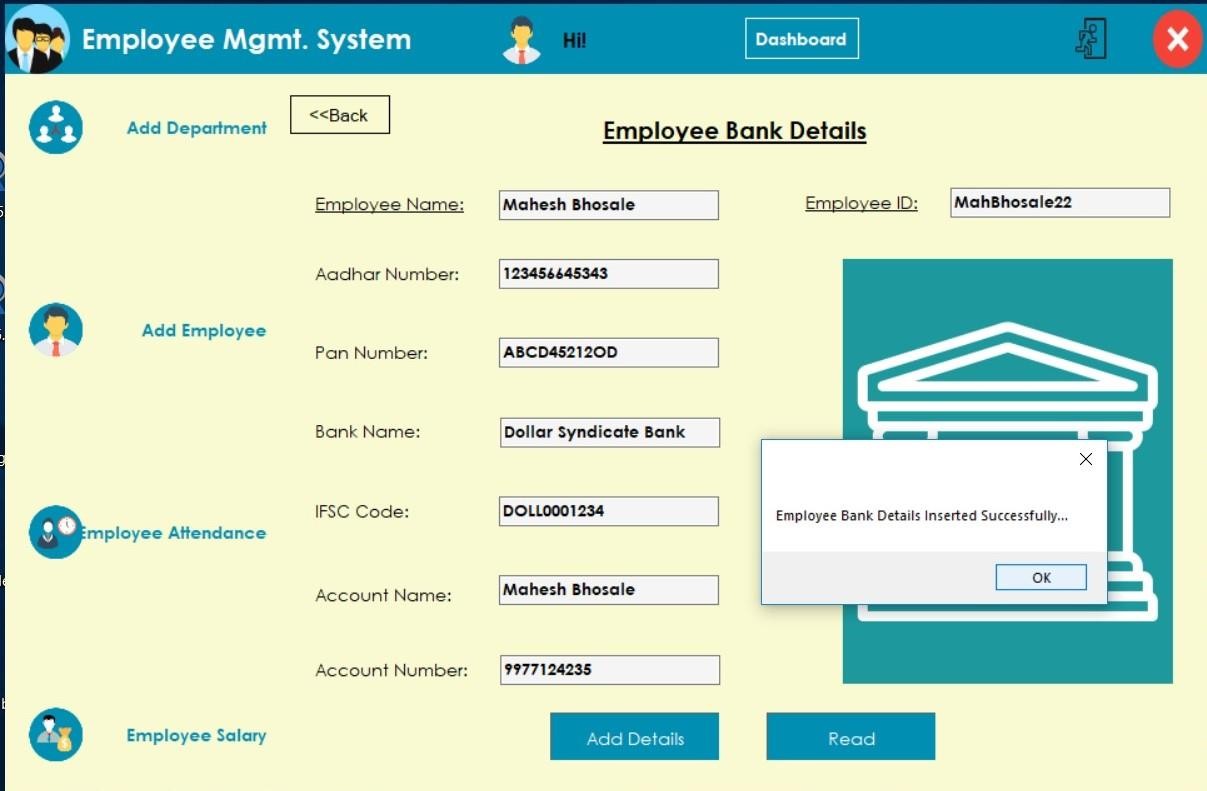
Admin can enter and update the Employee’s necessary Qualification Details as in his SSC, HSC, UnderGraduate Course details etc.

* 1. EmpJobExperience



Admin can Enter Employee’s Job Experience Details. I.e.mainly concerned about past organisation he used to work with, department he was associated with,designation he held and Slary he used to get.

* 1. EmpBankDetails



Admin can Add BankDetails and Update Bank Details of an Employee

* 1. EmpAdminAttendence:



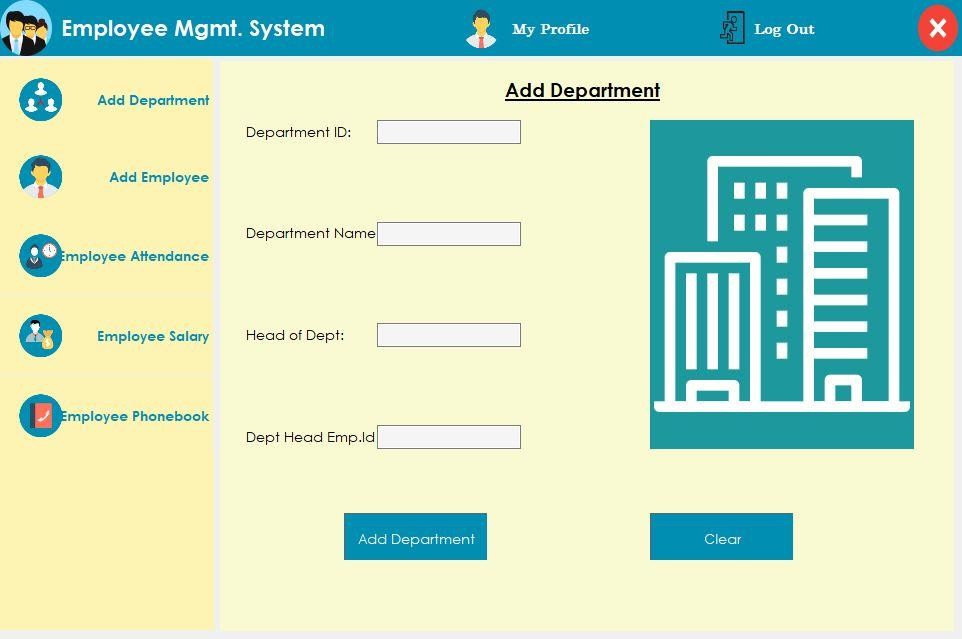
Admin can Record Employee’s Attendence if he fails to or admin can even correct the wrong inputs fed by the Employee.

1.8 EmployeeSalary:



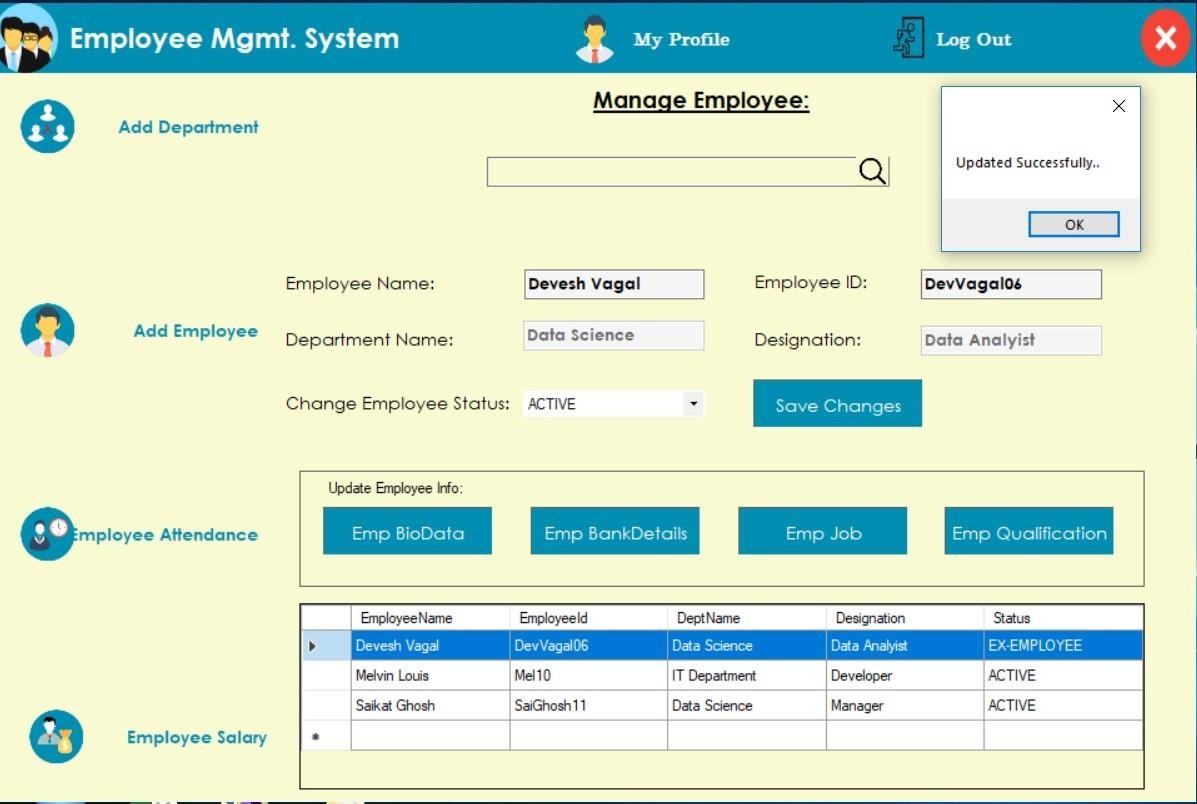
Admin can fix a monthly pay to a particular pay by adding allowances to his basic salary and manage additional deductions; if any. Employee also can view his salary.

* 1. Add Department



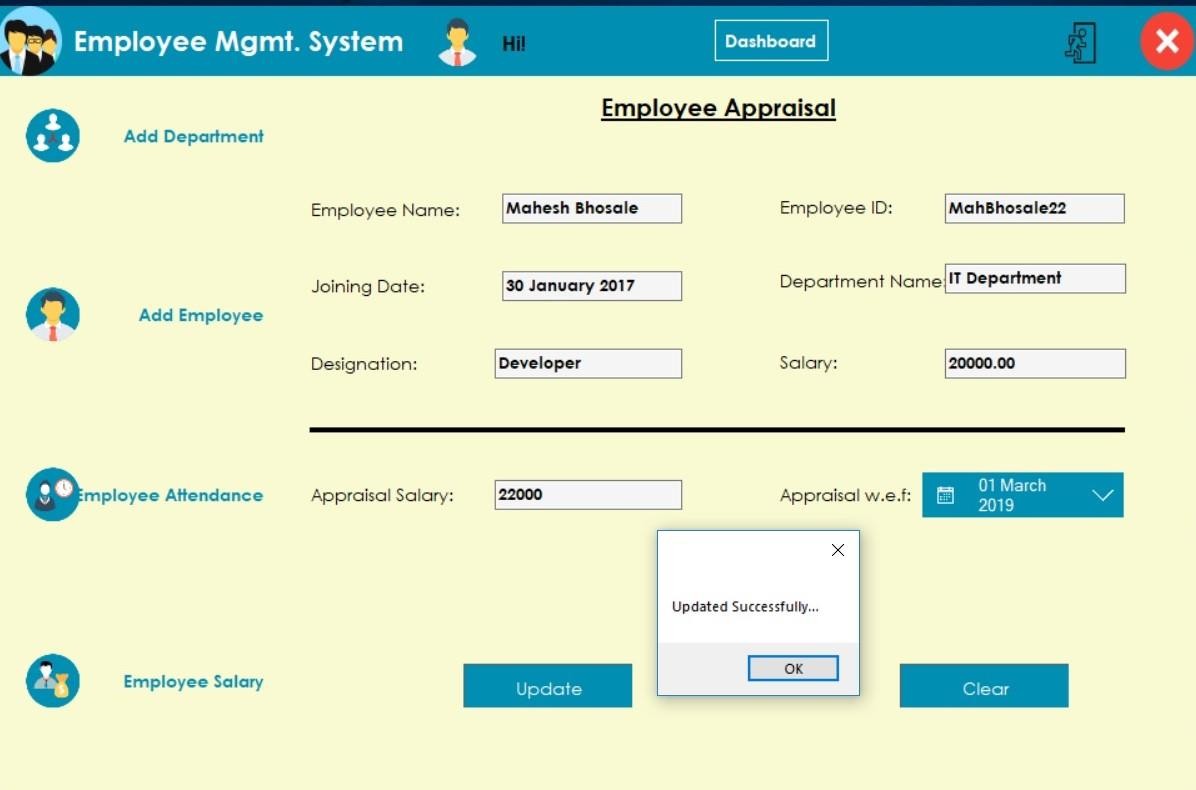
Admin can add Departments and Assign a new Head of Department

* 1. Manage Employee



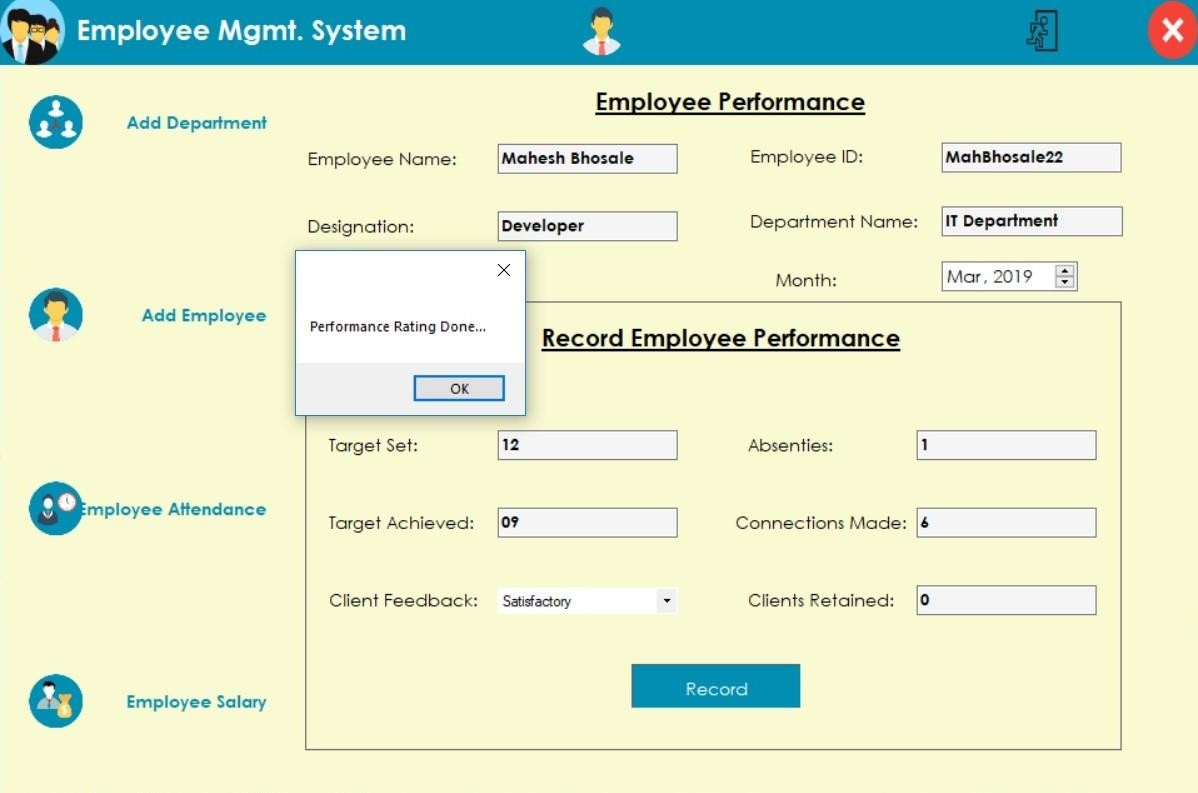
Admin can manage Particular Employee’s Record, change his Working Status to Active/ ExEmployee/ Resignation Period. Admin can also update Employee’s BioData/BankDetails/Qualification etc.

* 1. Employee Appraisal



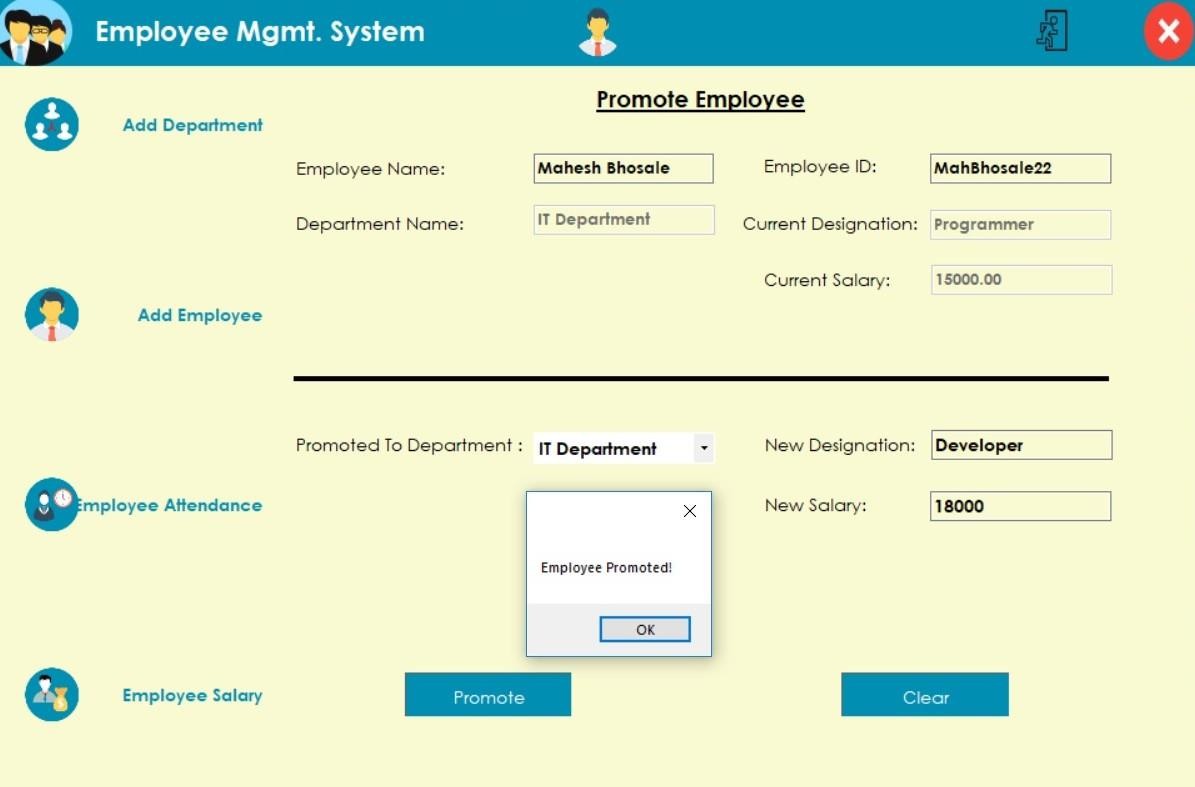
Admin can Update Employee’s Pay periodically or on Employee merits. He can decide the Date from which the revised pay would be effective.

* 1. Employee Performance



Admin can record Employee’s Performance on monthly basis.

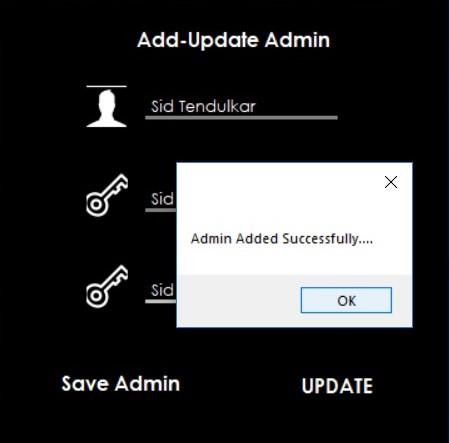
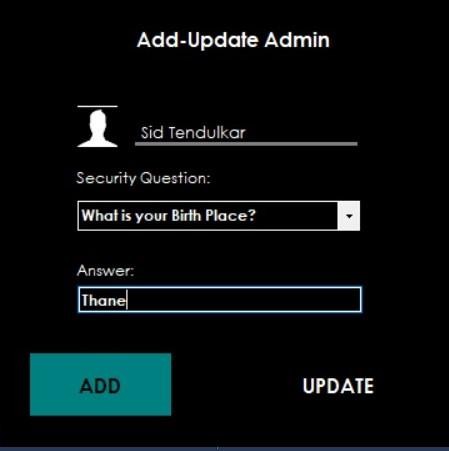
* 1. Employee Promotions



Admin can make decisions regarding Employee Promotions based on his punctuality and Performance.

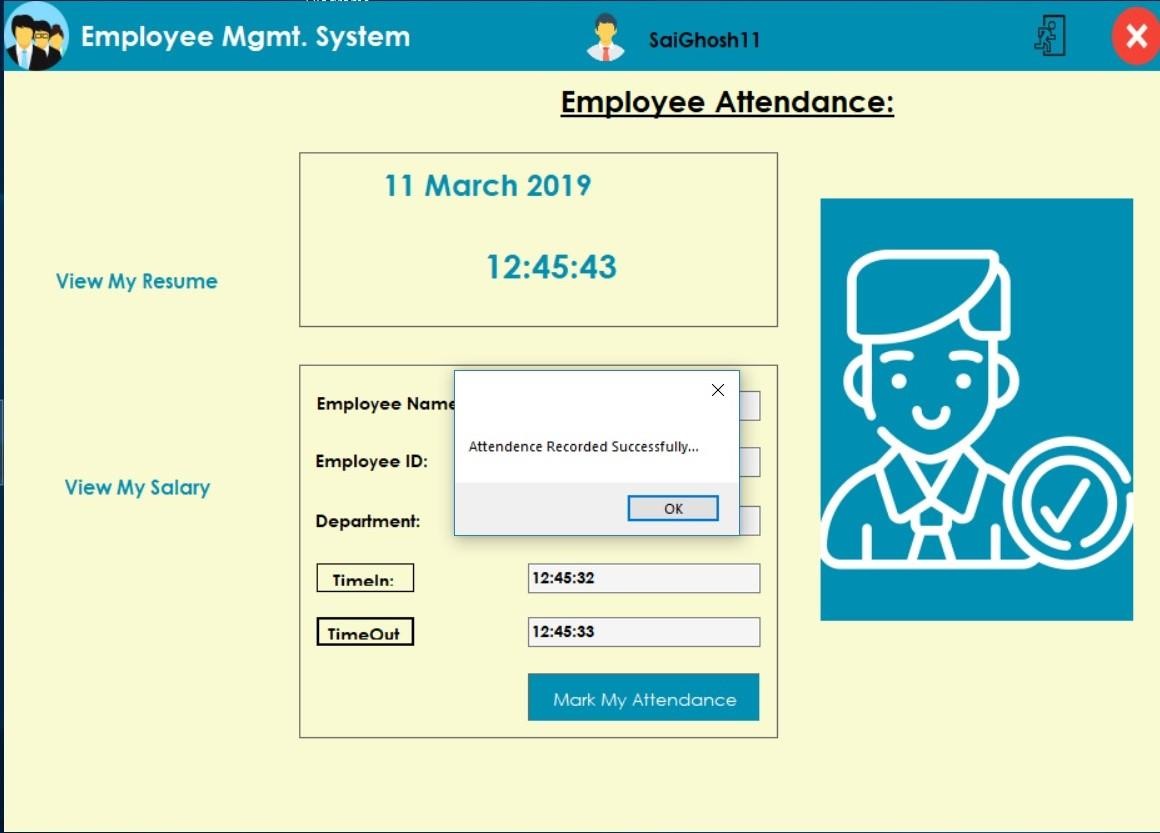
# Project ScreenShots

* 1. Add/Update Admin



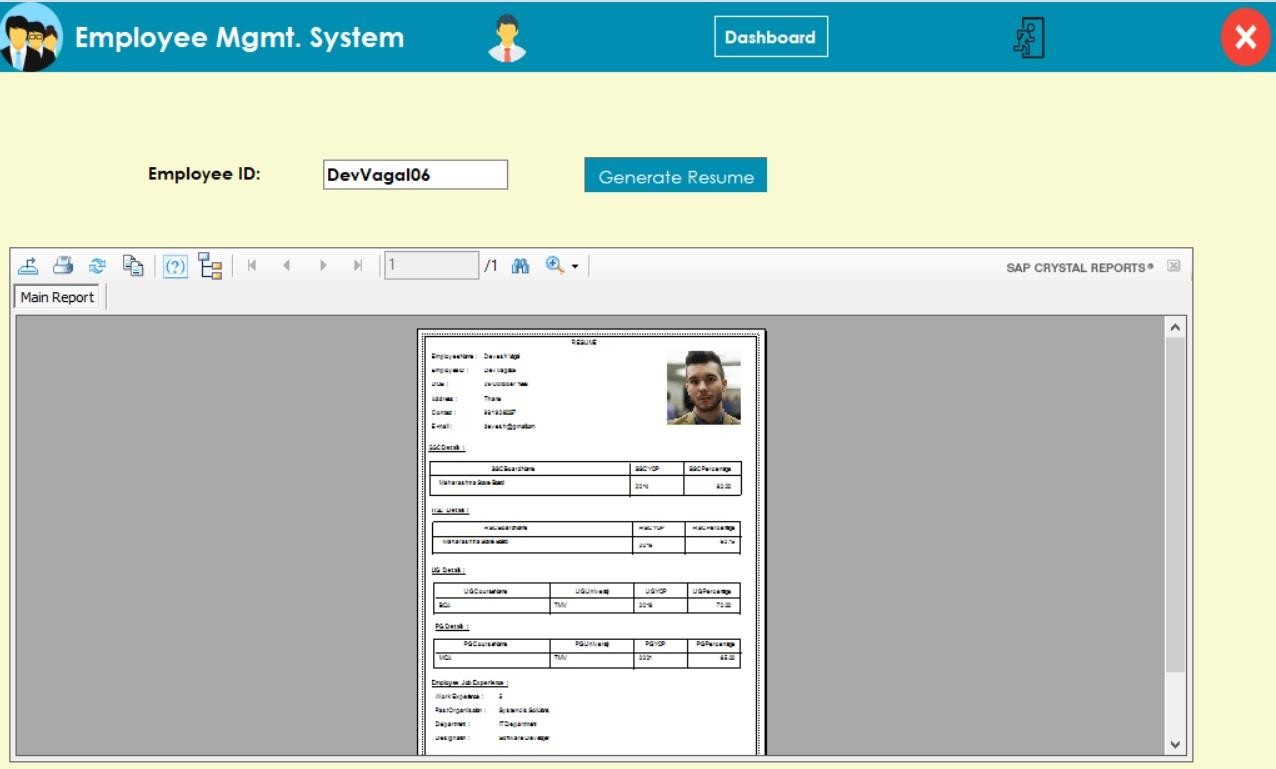
Admin can Update the existing Admin and can also Add another Admin

* 1. Employee Attendence



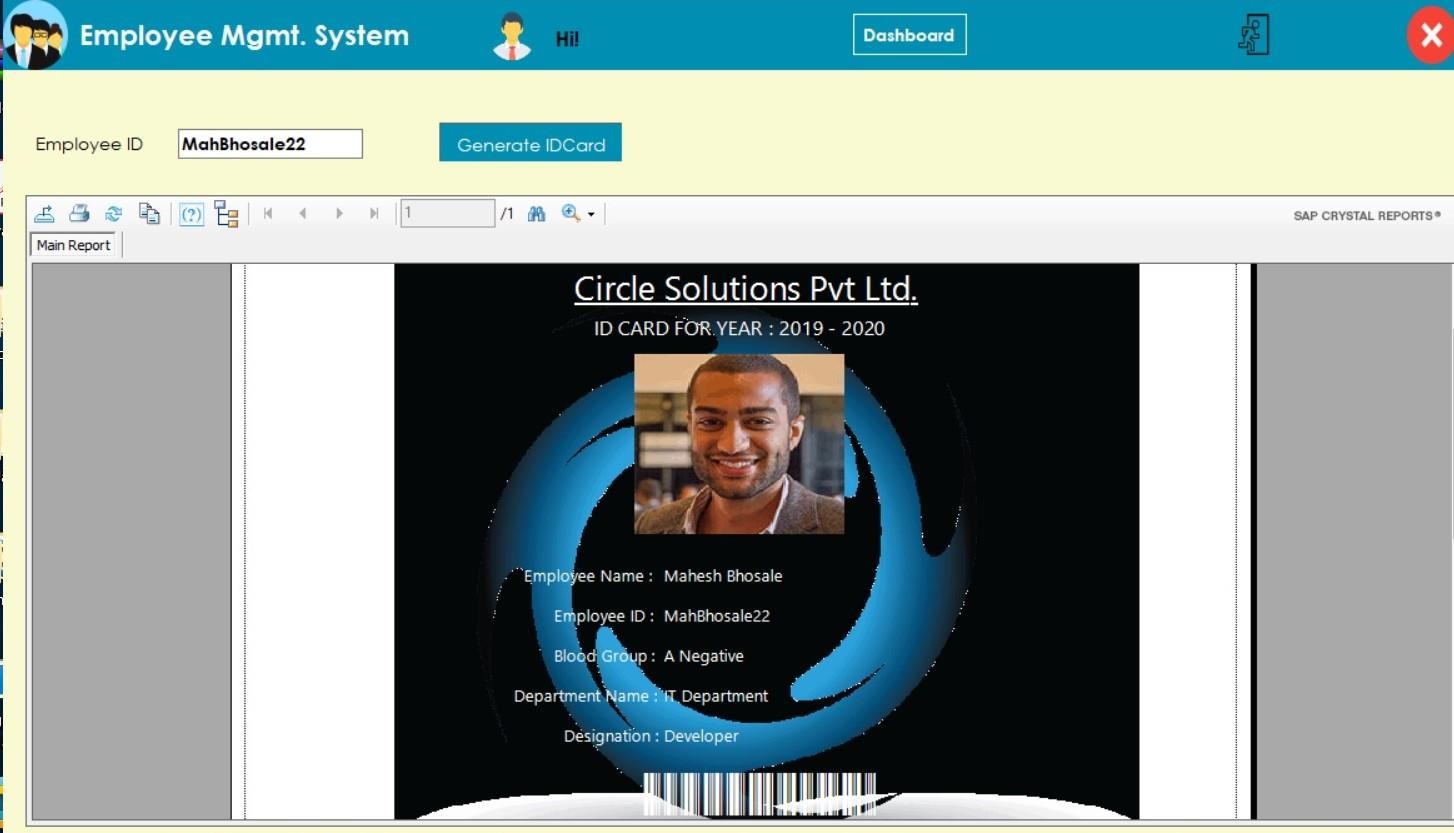
Employee can record his attendance for the System Date according to System Time.

* 1. EmployeeResumeReport



Admin can view every Employee’s Resume Report whereas Employee can view only self’s Resume Report.

* 1. EmployeeIDCard



Admin can generate Employee IDCard for his employees with Unique Barcode. It can be used for marking Employee’s Attendance.

# Ch. 12 Testing and Validations

12.1 Why Is Testing Important?

This is the most important part of the software life cycle. It provides better quality of software to end users; therefore, those end users won't come across software issues. Testing of any software is very important for validating functionality of the software. Testing will provide the following information: It finds issues during early phases, which can be fixed before finalization. It assures stability and reliability of software in different conditions. It helps to provide issue- free software for delivery. Any application must be tested with different methodologies. If the application is not tested properly, then some faulty application will be delivered to customers. Delivering such quality of application will reduce credibility, and the customers will be not delighted with application. Testing is usually conducted by development and quality assurance teams. This testing validates the functionality of the application.

For this **Employee Management System**, Validation Testing is performed. Various validation checks are implemented to ensure efficient and apt functioning of the System.

Validations are executed with the use of ERROR-PROVIDERS tool in Visual Studio. Main Validation Checks that are performed are:

1. RequiredField Validation: It ensures that the the field on which this particular error- provider has been assigned, should not be left void.
2. RangeValidation: It ensures that the number of characters that a textbox can take should be within the specified Range.
3. Custom Validation: This Validation can be customized to check on the desired input. It is executed for E-mailID and ContactNumber (so that textbox accepts only digits).
4. CompareValidation: This validation is used to compare and check whether the user has entered the authentic Password.

Below are the ScreenShots that depict the implementation of Validations in our System:

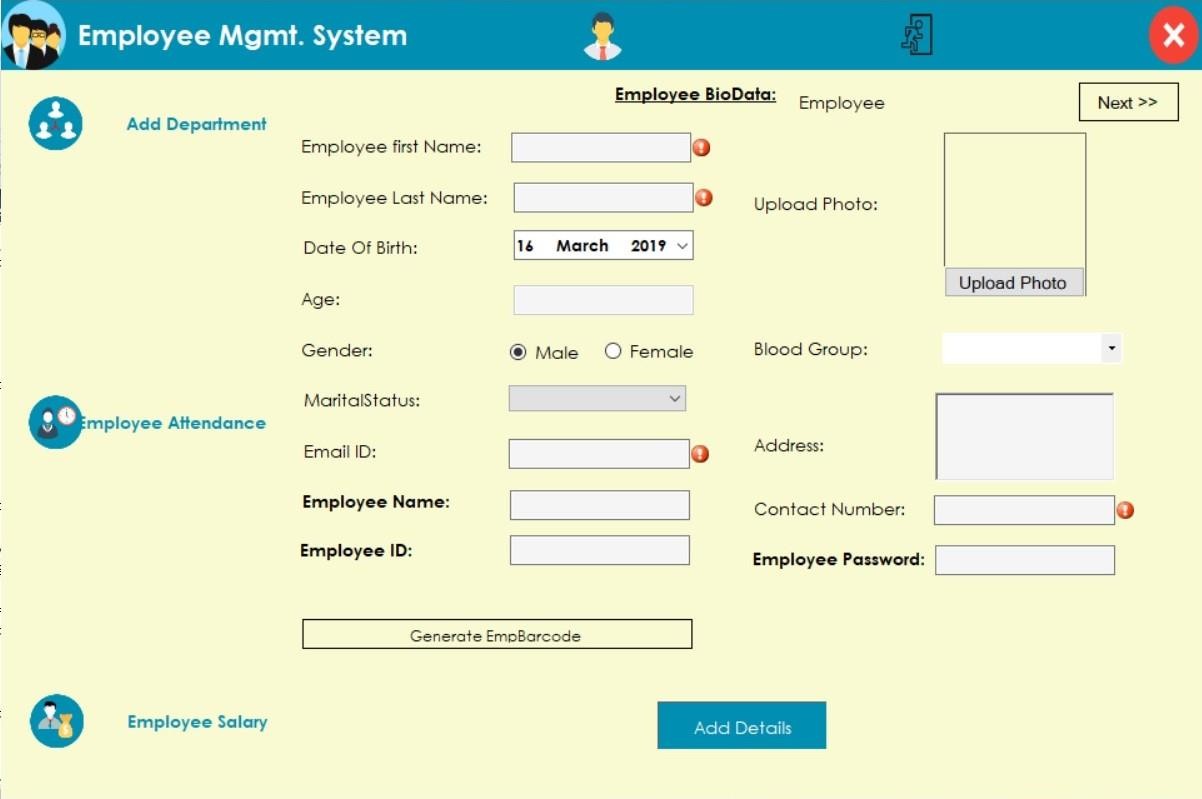
Validation ScreenShot No. 1:



RequiredField Validation is performed on UserName Field and USerType. They cannot be left blank.

.

Validation ScreenShot No. 2:



RequiredFieldValidations are performed on: Employee First Name, Employee Last Name, EmailID and Contact Number

Custom Validation is executed on EmailID whereas Range Validation alongwith Custom Validation is checked for ContactNumber such that it accepts 10 digit Number Only.

# Ch. 13 System Maintenance

System Maintenance stands for all the modifications and updations done after the delivery of software product. The software requirements vary according to client needs. Hence the software must be customizable to be able to meet Client Needs.

* 1. The Need for Modification in our System can be of the following form:
     1. **Market Conditions** - Policies, which changes over the time, such as taxation and newly introduced constraints like, how to maintain bookkeeping, may trigger need for modification.
     2. **Client Requirements** - Over the time, customer may ask for new features or functions in the software.
     3. **Host Modifications** - If any of the hardware and/or platform (such as operating system) of the target host changes, software changes are needed to keep adaptability.
     4. **Organization Changes** - If there is any business level change at client end, such as reduction of organization strength, acquiring another company, organization venturing into new business, need to modify in the original software may arise.

In Employee Management System lifetime, type of maintenance may vary based on its nature. It may either be just a routine maintenance task as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature. Following are some types of maintenance based on their characteristics:

* + 1. **Corrective Maintenance** - It includes modifications and updations done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
    2. **Adaptive Maintenance** - It includes modifications and updations applied to keep the software product up-to date and tuned to the ever changing world of technology and business environment.
    3. **Perfective Maintenance** - It includes modifications and updates done in order to keep the software usable over long period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.
    4. **Preventive Maintenance** - It includes modifications and updations to prevent future problems of the software. It aims to attend problems, which are not significant at this moment but may cause serious issues in future.

# Ch.14 Limitations and Future Enhancements

* 1. Limitations:

The Employee Management System makes use of Crystal Reports which are not portable. For the clients, the plug-in needs to be installed on their machine with Visual Studio and Sql Management Studio compatible versions. System design is hard to be customized for Client Needs. Since EMS is Desktop application, remote access is not possible.

* 1. Future Enhancements:
     + Project Assigning and Task Completion Modules to be implemented in Current System to help Employer / Manager to supervise and monitor Employee’s Performance. It will increase efficiency of Employee and thus benefit the organisation.
     + Concept of Centralization to be implemented so that Head Of Department will be awarded with Admin Privileges to Record Staff’s Attendance, calculate monthly salary of his staff and register their Performances.
     + Email like Communication system to be implemented to ensure swift and smooth internal vertical and lateral official conversations to help Employees and Admins to interact within Departments.
     + Data Warehousing and Data Visualization to be implemented to help Admin/ Employer to view Reports Graphically.

# Ch. 15 Conclusion

This Project “Employee Management Sytsem” is prepared to meet the requirements of Small Organisations to manage their Employee Base, their HR Requisites and effective control, therebymake it convenient for Employee Administration and eventually the Organisation. This software will help Organisation improve its WorkForce Managemant Efficiencies. This System in particular also offers solution to increase Employee Engagement and thus will positively impact Employee Retention. With tools such as KPIs will be able to provide a modern technique to track and Evaluate Employee Performances.

Thus, this System will by far be a complete One-Stop Offline Solution for Small Organisations to aid them in Employee Administration effectively.

# Ch. 16 Bibliography

For Icons: htt[ps://www.flati](http://www.flaticon.com/)c[on.com](http://www.flaticon.com/)

htt[ps://www.iconfinder.com](http://www.iconfinder.com/)

For Sql Queries: htt[ps://www.w3schools.com/sql/default.asp](http://www.w3schools.com/sql/default.asp)

For Synopsis Reference: https://nevonprojects.com

https://final-year-projects.in

For Coding Reference: C#: The Complete Reference Book. Author: Herbert Schildt

https://www.youtube.com(Vetrivel D. UI Academy, Fox Learn) https://stackoverflow.com

https://github.com htt[ps://www.quor](http://www.quora.com/)a[.com](http://www.quora.com/) https:/[/www.c](http://www.c-sharpcorner.com/)-[sharpcorner.com](http://www.c-sharpcorner.com/)

For Performance Metrics: htt[ps://www.ana](http://www.analyticsinhr.com/blog/employee-performance-)l[yticsinhr.com/blog/employee-performance-](http://www.analyticsinhr.com/blog/employee-performance-)

metrics

htt[ps://www.ere.net/](http://www.ere.net/7-easy-ways-to-measure-employee-)7[-easy-ways-to-measure-employee-](http://www.ere.net/7-easy-ways-to-measure-employee-) performance

Shared Libraries(.dll) used: Bunifu-UI,

Nugget Packages

SAP Crystal Reports in Visual Studio Color Theme reference: https://[www.color-hex.com](http://www.color-hex.com/)

htt[ps://www.w3schools.com/](http://www.w3schools.com/colors/colors_picker.asp)c[olors/colors\_picker.asp](http://www.w3schools.com/colors/colors_picker.asp)

Font for Barcode: [www.idautomation.com](http://www.idautomation.com/)

For UML Diagrams: staruml and lucidcharts:

https://staruml.io htt[ps://www.lucidchart.com](http://www.lucidchart.com/)