**Project Proposal Report On**

**WELINK Employee Management System**



**Submitted by: Submitted to:**

Shovan Raj Joshi Sudeep Bajimaya

22 C Softwarica College of IT

NCC ID - 00175040 and E-Commerce L5DC

**Acknowledgement**

My project is successfully completed with the help and combined efforts of myself and the people I have acknowledged who made contribution. The successful completion of my project can be attributed to the contribution made by the people I hereby acknowledge. To begin with, I would like to mention my teacher Mr. Sudeep Lal Bajimaya and thank him for his support, encouragement for this module and for clearing out my confusion regarding this project. I would also like to thank my friends and seniors for helping and teaching me C# .net framework and helping me out whenever I faced issues and problems. At last, I am very thankful for every person who helped me in one way or another way and thank my family and friends for providing me moral support to help me achieve my goals. I would also want to extend my appreciation to those who could not be mentioned here but here well played their role to help and inspire me during my project.

**Abstract**

WELINK Employee Management System is a standalone application that provides various features and functionality. Employee Management System will enable for an organization to have more simplified process of recording and managing information of employees. The system is created for administrative purpose.

Employee Management System’s sole purpose is to carry out administrative function. The system simplifies the process of maintaining records and managing information of the employees and provide well designed and established database for securely storing detailed information of the employee and information related to it. It will help the company to organize and securely store employee information. Information and details related to employees, projects, assignments and salary is also managed, assigned and stored for future use and to keeps its record.

Overall, Employee Management System will help the company to organize and securely store employee information. This system will help in management of these information and related information of employees.

**Table of Contents**

[**Table of Figures** 5](#_Toc15220228)

[**Chapter 1 Introduction** 6](#_Toc15220229)

[**1.0 Introduction** 6](#_Toc15220230)

[**1.2 Background of the project and Problem Statement** 6](#_Toc15220231)

[**1.3 Overview of the project** 6](#_Toc15220232)

[**1.4 Aim and Objectives** 7](#_Toc15220233)

[**Chapter 2 Analysis** 8](#_Toc15220234)

[**2.1 Introduction to Analysis** 8](#_Toc15220235)

[**2.2 Analysis Methodology** 9](#_Toc15220236)

[**2.3 Feasibility Study** 10](#_Toc15220237)

[**2.4 Requirement Analysis** 11](#_Toc15220238)

[**2.4.1 Functional Requirements.** 12](#_Toc15220239)

[**2.4.2 Non-functional Requirements** 14](#_Toc15220240)

[**2.4.3 MoSCoW Prioritization** 16](#_Toc15220241)

[**2.4.5 Software and Hardware Requirement Specification** 18](#_Toc15220242)

[**2.5 Use Case** 19](#_Toc15220243)

[**2.6 Natural Language Analysis** 21](#_Toc15220244)

[**Initial Class Diagram** 23](#_Toc15220245)

[**Chapter 3 Design** 24](#_Toc15220246)

[**Introduction to Design** 24](#_Toc15220247)

[**3.1 Structural Model** 24](#_Toc15220248)

[**3.1.1 Class Diagram** 25](#_Toc15220249)

[**3.1.2 Flowchart** 27](#_Toc15220250)

[**3.2 Behavioral Model** 28](#_Toc15220251)

[**3.2.1 Activity Diagram** 28](#_Toc15220252)

[**3.2.2 Sequence Diagram** 33](#_Toc15220253)

[**3.3 Data Modeling** 37](#_Toc15220254)

[**3.3.1 Data Dictionary** 37](#_Toc15220255)

[**3.3.2 E.R. Diagram** 39](#_Toc15220256)

[**3.4 Prototype Design** 40](#_Toc15220257)

[**3.5 Architectural Model** 49](#_Toc15220258)

[**Chapter 4 Implementation** 50](#_Toc15220259)

[**dbconnection:-** 50](#_Toc15220260)

[**Users:-** 51](#_Toc15220261)

[**Employee:-** 60](#_Toc15220262)

[**Project:-** 71](#_Toc15220263)

[**assignment:-** 80](#_Toc15220264)

[**salary:-** 88](#_Toc15220265)

[**Chapter 5 Testing** 95](#_Toc15220266)

[**5.1 Testing Introduction** 95](#_Toc15220267)

[**5.2 Unit Testing** 106](#_Toc15220268)

[**Chapter 6 Other Project issues** 110](#_Toc15220269)

[**6.1 Project Limitation** 110](#_Toc15220270)

[**6.2 Future Work** 110](#_Toc15220271)

[**6.3 Other Project Issues** 110](#_Toc15220272)

[**6.4 Risk Management** 111](#_Toc15220273)

[**6.5 Configuration Management** 112](#_Toc15220274)

[**6.6 User Manual** 114](#_Toc15220275)

[**Chapter 7 Conclusion** 124](#_Toc15220276)

[**References** 125](#_Toc15220277)

# **Table of Figures**

[Figure 1 OOD 9](#_Toc15214228)

[Figure 2 Use case 20](#_Toc15214229)

[Figure 3 Initial Class diagram 24](#_Toc15214230)

[Figure 4 Class Diagram 26](#_Toc15214231)

[Figure 5 Flow Chart 28](#_Toc15214232)

[Figure 6 Activity Diagram overview of the system 30](#_Toc15214233)

[Figure 7 Activity Diagram for Employee Mgmt. and Assignment Mgmt. 31](#_Toc15214234)

[Figure 8 Activity Diagram for Employee Mgmt. and Salary Mgmt. 32](#_Toc15214235)

[Figure 9 Activity Diagram for Project Mgmt. and Assignment Mgmt. 33](#_Toc15214236)

[Figure 10 Sequence Diagram for Login 35](#_Toc15214237)

[Figure 11 Sequence Diagram for Employee Mgmt. 35](#_Toc15214238)

[Figure 12 Sequence Diagram for Project Mgmt. 36](#_Toc15214239)

[Figure 13 Sequence Diagram for Assignment Mgmt. 36](#_Toc15214240)

[Figure 14 Sequence Diagram for Salary Mgmt. 37](#_Toc15214241)

[Figure 15 Entity Relationship Diagram 40](#_Toc15214242)

[Figure 16 User Form 1 41](#_Toc15214243)

[Figure 17 User Form 2 41](#_Toc15214244)

[Figure 18 User Form 3 42](#_Toc15214245)

[Figure 19 User Form 4 42](#_Toc15214246)

[Figure 20 User Form 5 43](#_Toc15214247)

[Figure 21 User Form 6 43](#_Toc15214248)

[Figure 22 User Form 7 44](#_Toc15214249)

[Figure 23 User Form 8 44](#_Toc15214250)

[Figure 24 User Form 9 45](#_Toc15214251)

[Figure 25 User Form 10 45](#_Toc15214252)

[Figure 26 User Form 11 46](#_Toc15214253)

[Figure 27 User Form 12 46](#_Toc15214254)

[Figure 28 User Form 13 47](#_Toc15214255)

[Figure 29 User Form 14 47](#_Toc15214256)

[Figure 30 User Form 15 48](#_Toc15214257)

[Figure 31 User Form 16 48](#_Toc15214258)

[Figure 32 User Form 17 49](#_Toc15214259)

[Figure 33 User Form 18 49](#_Toc15214260)

[Figure 34 Unit Test Passed 107](#_Toc15214261)

# **Chapter 1 Introduction**

## **1.0 Introduction**

Employee Management System (EMS) is the project that is to be conducted and developed. The main purpose of this project is to establish proper management system to manage and store Employee information and information relevant and related to employees. Employee Management System will be developed for making management of employee, project, assignment and salary information and details much easier. The application or the system being developed will be developed using Visual Studio C# and .net framework. Employee Management System will include various features and functions for it to be used.

## **1.2 Background of the project and Problem Statement**

Commonly, information of the employee in an organization or company is not electronically managed or has not adopted employee management system and majority of these organization relies on traditional process of record maintenance of employees (Hand-written). Developing Employee Management System will help a company or organization in various ways. It will enable for an organization to have more simplified process of recording and managing information of employees. Therefore, developing an employee management system will be much better approach for managing employee information and also the information will be securely stored.

**Problem Statement:**

* The main purpose for developing Employee management system is to manage employee information and data related to it. It is also serves the purpose of assigning specific projects to specific employees.
* It solves the problem of entering data and is easy to use. The system is well protected.
* It also track information of the existing projects.
* Distribution of salary and storing employee information is much easier and secured.

## **1.3 Overview of the project**

The project's sole purpose is to create an employee management system that will help the company to organize and securely store employee information. Information and details related to employees, projects, assignments and salary is also managed, assigned and stored for future use and to keeps its record. This system will help in management of these information and related information of employees. The system has few limitation such as it is not developed for larger organization and is also not a web-based system. The main aim of the project is to develop an Employee Management System for effective and efficient management of employee information and also manages information related to employees like project, salary and assignment.

The system proposed is a standalone application. Waterfall method is implemented for the development of Employee Management System. The stages/phases of this method are Requirement Specification, Analysis, Design, Implementation, Testing, Deployment and Maintenance. The database contains different tables. OOD and MVC pattern is implemented for coding.

## **1.4 Aim and Objectives**

**Project Aim:-**

* To develop an Employee Management System to make management of employee information easier and maintainable.
* To store important and critical information of employees to assign them to certain projects and manage their salaries.
* To store detailed information of the employees working in the organization in a well-designed database.

**Project Objective:-**

* To create user friendly system which is easy to use and understand.
* To allow admins to register.
* To make use of login feature for the admins.
* To avoid mixing up information of the employees
* To make the data stored secured.
* To make the data entry easier.
* To make the database maintainable.
* To check the system for defects or errors and correcting or fixing them if required.
* To make the system more adaptive to changes and agile.

# **Chapter 2 Analysis**

## **2.1 Introduction to Analysis**

Analysis can be defined as detailed examination of the information or data and also can be evaluated. It can also be said as the process where we divided the elements into smaller components to make a clear understanding of the information or data. It also provides fundamentals for taking better decision and solving problems.

The first stage of SDLC (Software Development Cycle) is analysis. This stage is very important to determine and define the goals and objective of the project. During the analysis stage, possible problems are identified and also solution of the problems. Recommendations are given for the improvement of the project. Costs, benefits, project’s pros and cons are taken into account for proper planning of the project.

Before staring the development works system requirements are analyzed and after this analysis process system requirement specification and its detail is created.

## **2.2 Analysis Methodology**

Object Oriented Design Methodology is the analysis methodology I have chosen to use. Applying this methodology enables to build a more rigid working system which is well-designed.

Single entities called objects are used in Object Oriented Approach. Complex relationships can be represented in Object Oriented models. This also helps us for better analysis and designing. The main aim of this methodology is to make improvement in the quality of system analysis and design.

The benefits of using Object Oriented Design Methodology are:-

* Changes can be made to the system at low and manageable cost.
* Components can be reused.
* Codes can be recycled.
* Integrating components to configure larger system can be simplified.
* Maintainability of software is improved.
* OOD forces extensive planning phase so that the design can better with minimum flaws.

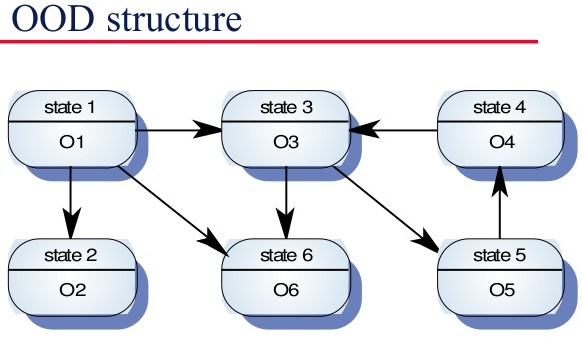


Figure 1 OOD

## **2.3 Feasibility Study**

Initial design phase of the project where elements of knowledge is gathered to see if the project is possible or not. Some of the types of feasibility study are:-

**Technical Feasibility Study**

It is a study done to find out whether the organization have the technological resources and people working on the project have the capabilities to undertake the project. It covers the important aspects of engineering which is required for the project’s design.

**Schedule Feasibility Study**

It is a study done to find out whether the project can be completed in given time and if the time specified for certain objectives are viable.

**Economic Feasibility Study**

It is a study done to find out if the project can be undertaken with the given financial resources and if the project is financially viable. This study also can be called as cost/benefit analysis.

**Cultural Feasibility Study**

It is a study done to find out both the general and local cultural impact. The project that is being developed should be appropriate with the cultural environment. Cultural beliefs and practices should be taken into account while carrying out project development. This will decrease any tension between cultural beliefs and practices of the people.

**Political Feasibility Study**

It is a study done to find out if the project being developed will be effected by political factors. Political factors may also represent legal/ethical viability for the project being developed. So, it important to consider political factors while developing a project.

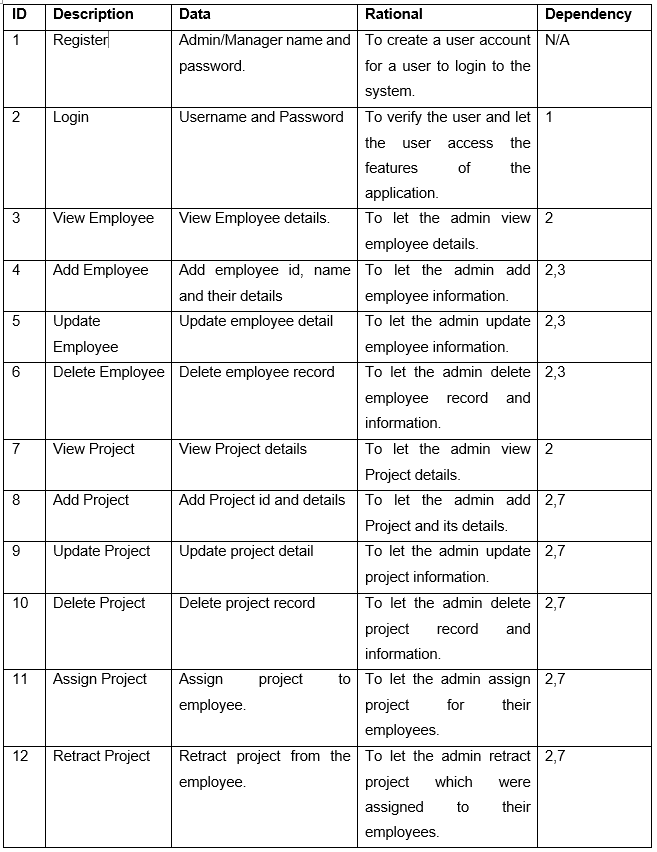
**Safety Feasibility Study**

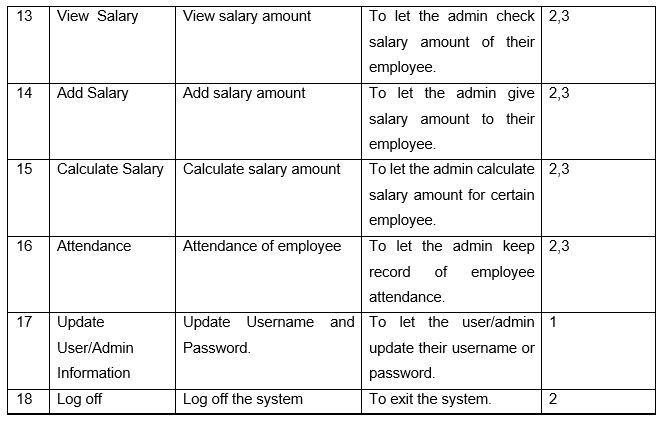
It is a study done to find out whether the project being developed can be developed without major obstruction or having negative impact on the stakeholders and environment.

## **2.4 Requirement Analysis**

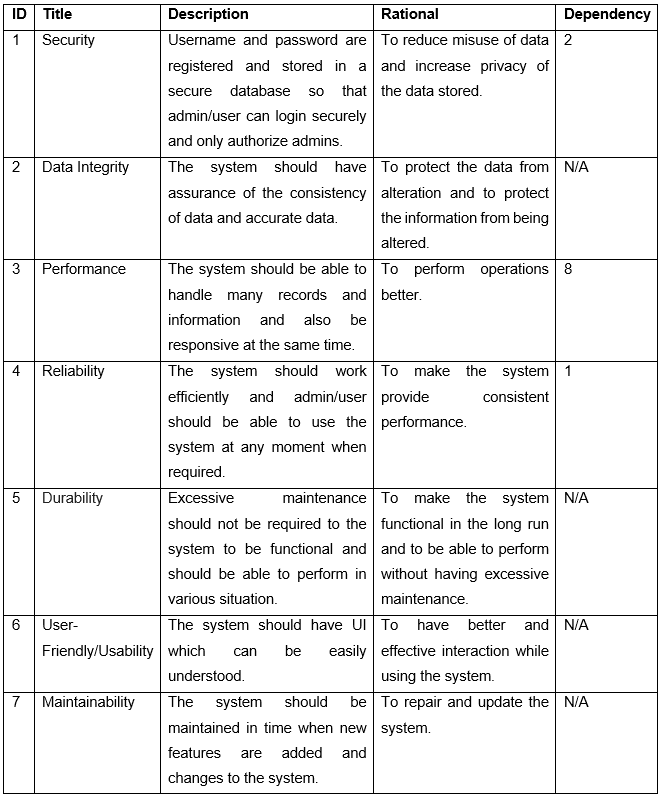
One of the most important part of analysis in project management is requirement analysis. Requirement analysis can be defined as the processes of determining needed and relevant requirements to meet the user expectation and the requirements should be detailed and specified. The two important types of requirements are Functional Requirements and Non-Functional Requirements.

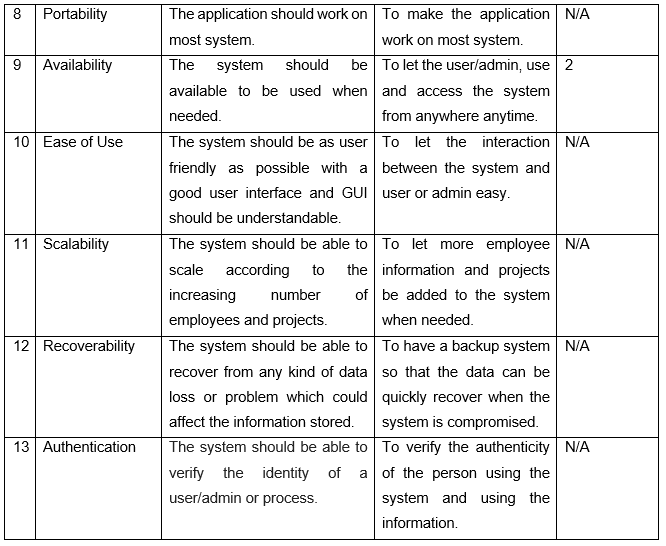
## **2.4.1 Functional Requirements.**





## **2.4.2 Non-functional Requirements**





## **2.4.3 MoSCoW Prioritization**

Moscow prioritization can be defined as prioritization technique which is used for managing, analysis business, management of project and software development to understand the requirements and managing which is important than the other. It can be also explained as the process of meeting necessary and important requirement where both parties (Stakeholders and client) agree on.

The four MoSCoW prioritization categories are:

* Must Have: The requirements that must be in the system or project and also should be fully functional.
* Should Have: The requirements that are just a step below must have. These requirements are not vital but is important. If these requirements are added then they add significant value but it still can function if left out.
* Could Have: The requirements that are not vital for the systems but are nice to have and can be useful but it is not necessary.
* Won’t Have: The requirements that are not important and is not necessary to the system.

Prioritization for Functional Requirements:

|  |  |  |
| --- | --- | --- |
| **ID** | **Functional Requirement** | **MoSCoW** |
| 1 | Register | Must Have |
| 2 | Login | Must Have |
| 3 | View Employee | Must Have |
| 4 | Add Employee | Must Have |
| 5 | Update Employee | Must Have |
| 6 | Delete Employee | Must Have |
| 7 | View Project | Must Have |
| 8 | Add Project | Must Have |
| 9 | Update Project | Must Have |
| 10 | Delete Project | Must Have |
| 11 | Assign Project | Should Have |
| 12 | Retract Project | Should Have |
| 13 | View Salary | Should Have |
| 14 | Add Salary | Must Have |
| 15 | Calculate Salary | Could Have |
| 16 | Attendance | Could Have |
| 17 | Update User/Admin Information | Must Have |
| 18 | Log off | Could Have |

Prioritization for Non-Functional Requirements:

|  |  |  |
| --- | --- | --- |
| **ID** | **Non-Functional Requirement** | **MoSCoW** |
| 1 | Security | Should Have |
| 2 | Data Integrity | Should Have |
| 3 | Performance | Must Have |
| 4 | Reliability | Could Have |
| 5 | Usability | Must Have |
| 6 | User-Friendly | Must Have |
| 7 | Maintainability | Should Have |
| 8 | Portability | Won’t Have |
| 9 | Availability | Won’t Have |
| 10 | Ease of Use | Should Have |
| 11 | Scalability | Could Have |
| 12 | Recoverability | Could Have |
| 13 | Authentication | Could Have |

## **2.4.5 Software and Hardware Requirement Specification**

Software and hardware requirement specification can be described as any exact information about hardware or software products and their requirements. (Inflectra, 2018)

The software and hardware requirement specification recommended for the Employee Management System to operate and function properly are:

|  |  |  |  |
| --- | --- | --- | --- |
| **S.N.** | **Title** | **Minimum** | **Recommended** |
| 1 | Operating Systems | Windows XP/vista | Windows 7, Windows 7, Windows 10(32/64-bit) |
| 2 | Integrated Development Environment (IDE) | Visual Studio 2013 (2.0-4.5.2.NET Framework version) | Visual Studio 2017 (3.5-4.7.NET Framework version) |
| 3 | Relational Database Management System (RDBMS) | SQL Server 2014 | SQL Server 2017 |
| 4 | Processors | i3 1.6 GHz or better | Core i5 or better. |
| 5 | RAM | 4 GB | 8 GB |
| 6 | Storage | 3 GB | 5 GB |

## **2.5 Use Case**

A use case can be defined as a methodology that analyze the system to find out, identify and organize the system and its requirements. Use case is generally made up of possible interactions between users and the system for using the system’s function and features.

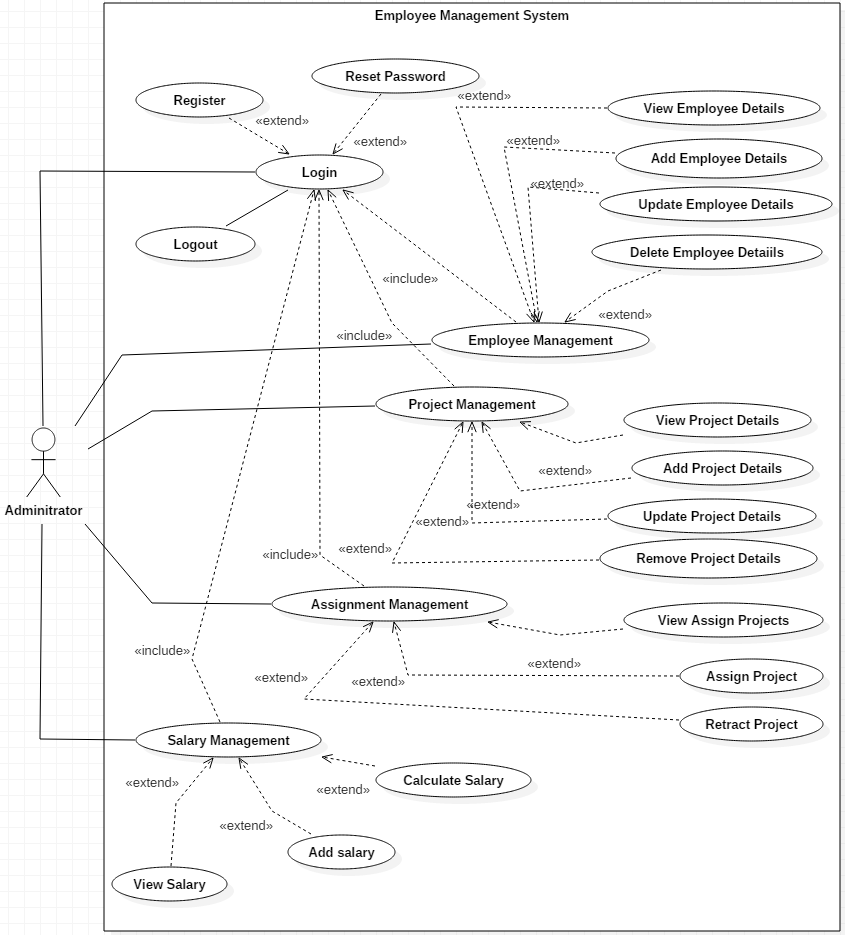


Figure 2 Use case

The use case represent the interaction between the administrator and the function and featured he/she can use. To start off, the administrator should be registered and should login in to use the available functions and features. The functions available are employee management, project management, assignment management and salary management. The features of these functions are similar as employee management, project management allows administrator to view, add, update and delete employee and project respectively. Assignment managements has features to assign project to employees and retract project and salary management allows the administrator to view add and calculate salary. The only user in this system is Administrator and this user only can use the systems function and features once they are logged in. Therefore, the use case represents the interaction between the administrator and employee management system.

## **2.6 Natural Language Analysis**

NLA (Natural Language Analysis) is the process of using a system’s ability to process sentences in natural languages such as English. This process helps to understand the system better. This process involves identifying nouns, verbs and adjectives. (electra0084, 2019)

WELINK consultancy is a small organization. The company is in need of proper and better employee management system. The company follows hardcopy management of employee information and records. The company wants to make a better management system so that the information about employees can be securely stored and management of those information will be more effective. The organization expects the system to have a certain features that will make it effective while making the use of it. The system should consist a feature to add employee information and details such as employee name, gender, age, employee ID, Address etc. They also want the system to have features where projects can be added and for the projects to be assigned for specific employees. Main function of this system is to view, add, update, delete employee information and project information and calculate salary and also assign employee to certain projects. The system should also have some sort of feature for keeping track of salaries that must be provided to the employees and amount that should be provided. The system could have attendance but this feature is not mandatory.

|  |  |  |
| --- | --- | --- |
| **Nouns** | **Verbs** | **Adjectives** |
| Admin | Register | Small |
| User | View | Proper |
| Manager | Add | Better |
| Employee | Update | Effective |
| Project | Delete | Specific |
| Assignment | Detail | Certain |
| Salary | Calculate |  |
| Information | Assign |  |
| System |  |  |
| Attendance |  |  |

**Candidate Class**

|  |  |
| --- | --- |
| **CLASS** | **Function** |
| **Admin** | To let the admin create user details like password and username. |
| **Employee** | For accessing and manipulating employee information. |
| **Project** | For accessing and manipulating project information. |
| **Salary** | For calculating and distributing salary to employee. |
| **Assignment** | For assigning project to employees or to retract assigned project. |

## **Initial Class Diagram**

A class diagram is a static diagram that can be described as the overall system structure which includes classes, attributes and operations and also shows the relationship between classes.

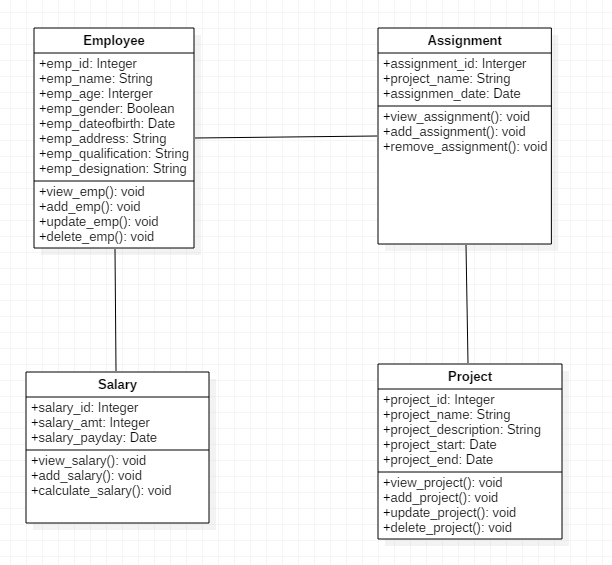


Figure 3 Initial Class diagram

# **Chapter 3 Design**

## **Introduction to Design**

One of the important part while developing a project is Design. The second stage of the software development life cycle (SDLC) is design. The system is designed according to the constructed analysis and requirements. Design contains the structure of a system that is to be built which can be represented by various diagrams. Design stage helps a system to be built without having many complication and the errors can be identified easily. The features and function can be described in detail in design specification. Structural models and Behavioral models are mostly covered in design. The design should also include the design of the database system before implementation.

## **3.1 Structural Model**

The whole framework of the system is represented in structural model. It represents all the classes and objects of the system is represented in structural model and also shows their relationship with each other. Class diagram is one of the type of structural model. Different classes, attributes, operations and the relationship between these objects is represented in class diagram.

## **3.1.1 Class Diagram**

Class diagram can be defined as the unified modeling language that shows the structure of the system which includes all the classes and their attributes, operations and relationship between these objects. The static view of the system is represented here which will be used during the development of the project.

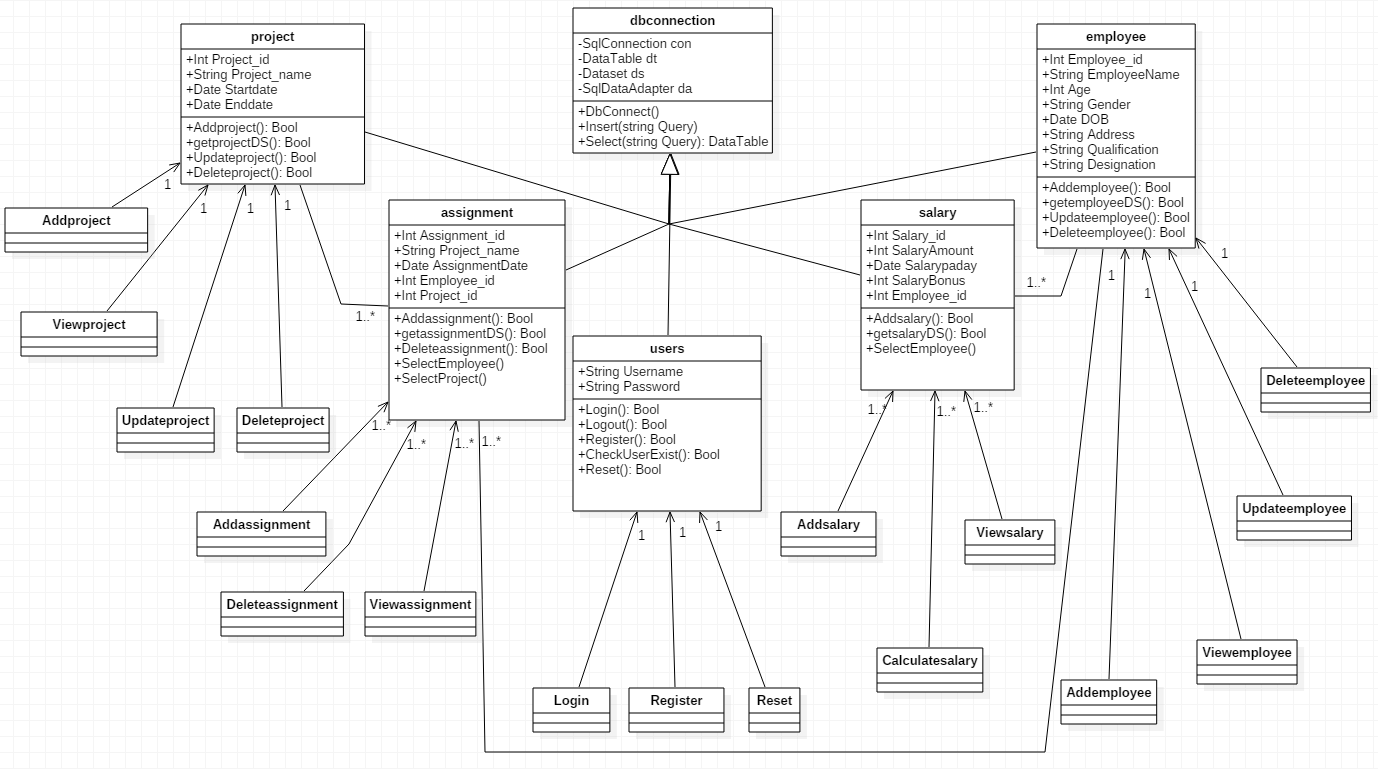


Figure 4 Class Diagram

The above class diagram is for the project (Employee Management System). Model View Controller Pattern is used for this project. The administrator/user will use the four class (Employee, Assignment, Project, and Salary) for their CRUD functions. Login will be used for verifying, authenticating and login of the administrator/user in the system. Register will be used for the registration of the new administrator/user. Add (Employee, Project, Assignment, and Salary) will be used for adding new employee, project, assignment, and salary respectively. View (Employee, Project, Assignment, and Salary) will be used for viewing added and existing employee, project, assignment, and salary respectively. Update (Employee, and Project) will be used for editing and updating employee and project details respectively. Delete (Employee, Project, and Assignment) will be used for deleting their details from the database.

## **3.1.2 Flowchart**

A flow chart can be defined as graphical representation of any activities and action that is involved in a project. The main objective of flow chart is to show the overview of the systems and provides reference point for the people dealing with the project.

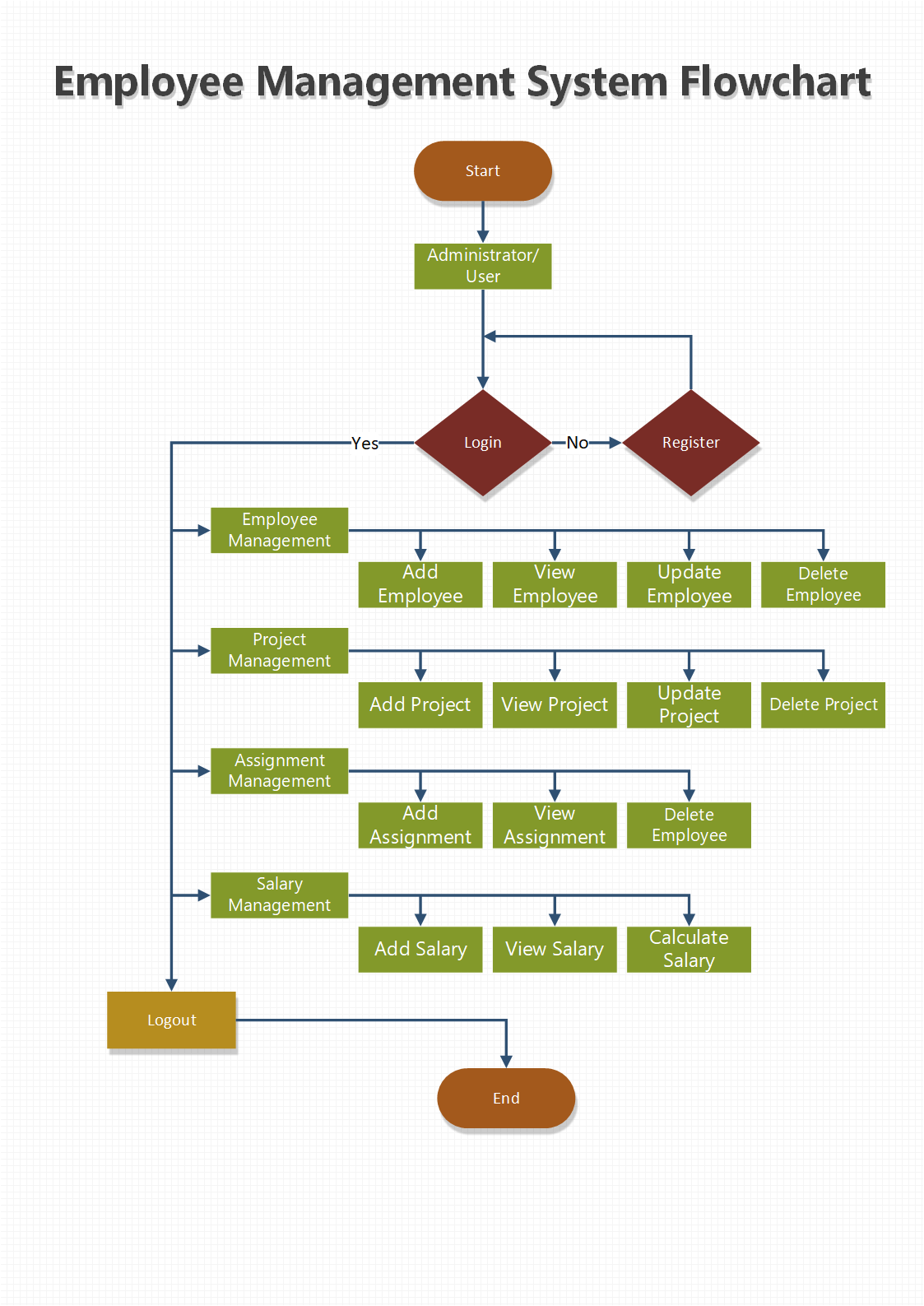


Figure 5 Flow Chart

The flow chart is for the project that is to be developed. Different functions and features like employee management (add, view, update, delete), project management (add, view, update, delete), assignment management (add, view, delete) and salary management (add, view, calculate) can be used by the logged in admin.

## **3.2 Behavioral Model**

Behavioral model can be defined as any graphical representation that shows the internal structure and features of the system. The dynamic sequence of flow of the system is shown. The two types of behavioral model diagram are sequence diagram and activity diagram.

## **3.2.1 Activity Diagram**

An activity diagram can be defined as graphical representation of the system's flow of control and actions/activities. Its main objective is to describe the activities in detail.

**Notations Used**

|  |  |  |
| --- | --- | --- |
| S.N. | Symbol | Name |
| 1 |  | Initial Point |
| 2 |  | Activity |
| 3 |  | Decision Symbol |
| 4 |  | Fork node |
| 5 |  | Join node |
| 6 |  | Activity flow |
| 7 |  | End Point |
| 8 |  | Swinlanes |

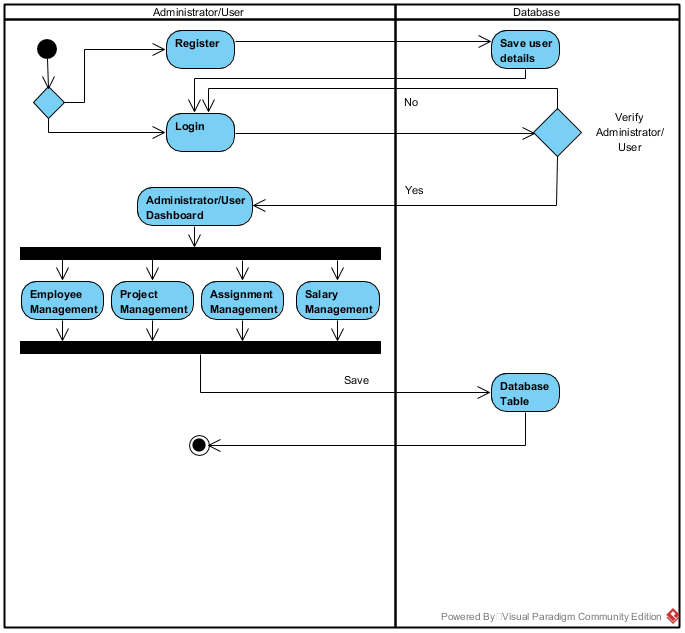


Figure 6 Activity Diagram overview of the system

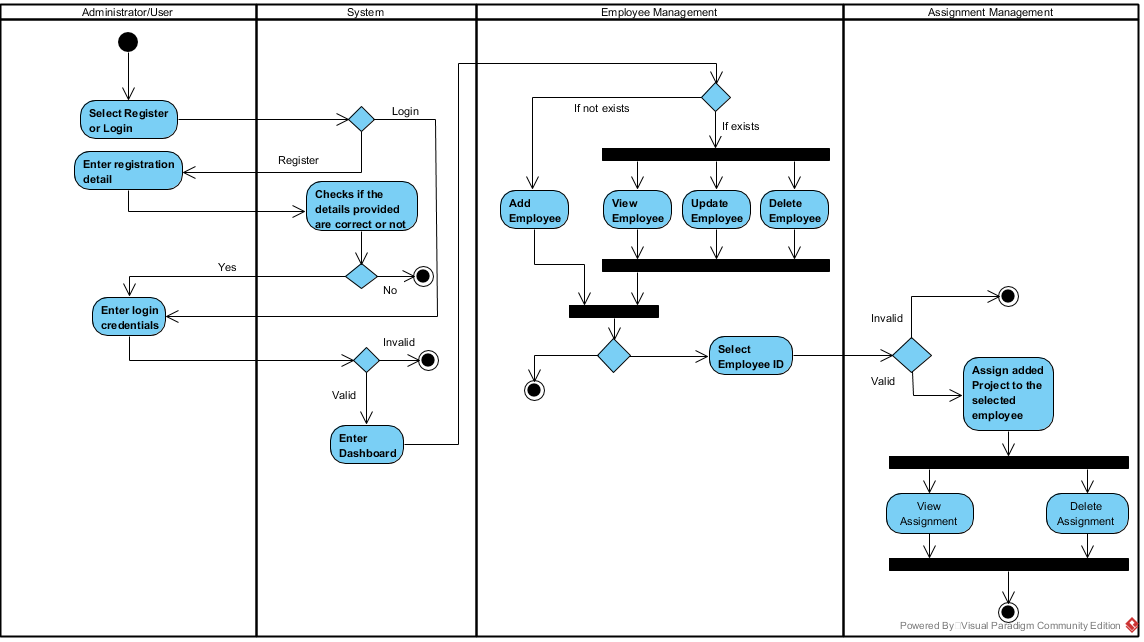


Figure 7 Activity Diagram for Employee Mgmt. and Assignment Mgmt.

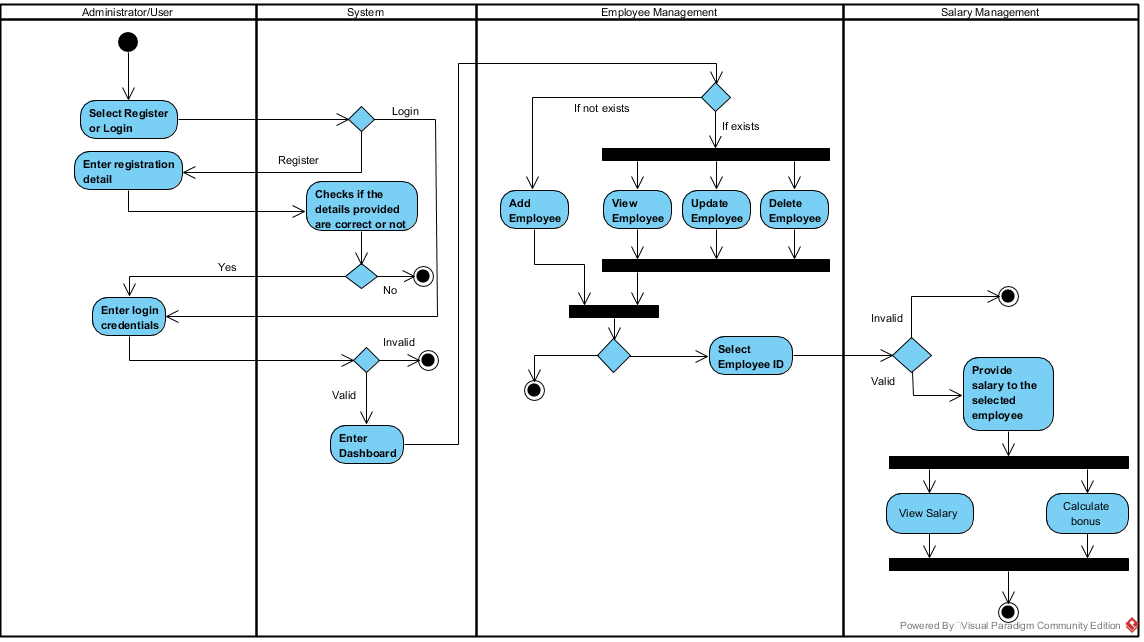


Figure 8 Activity Diagram for Employee Mgmt. and Salary Mgmt.

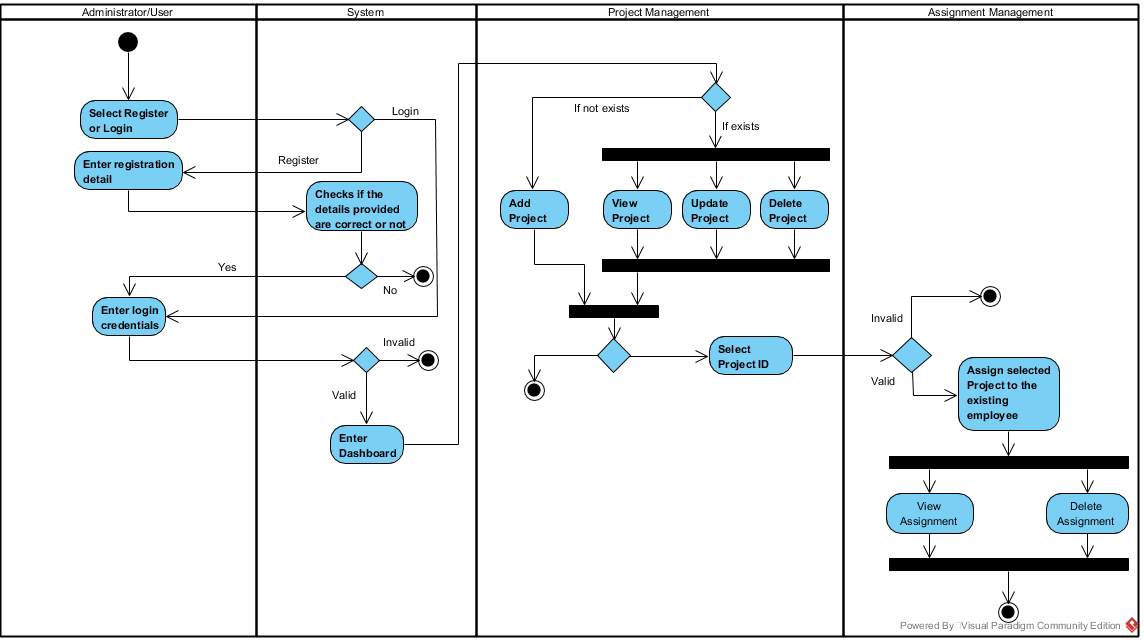


Figure 9 Activity Diagram for Project Mgmt. and Assignment Mgmt.

To begin with, the administrator/user logins to the system. If the administrator/user is not registered than he/she should register in order to login to use the system. When the administrator/user logins, his/her credentials are verified to check if the administrator/user is legit. After the login is successful, he/she is redirected to administrator dashboard where four features are available in the system to be used. The administrator/user can use all the features where he/she will be able to add, view, update and delete employee details in employee management. Administrator/user will be able to add, view, update and delete project details in project management. Administrator/user will be able to assign projects to the employees’ available, view and delete assignment in assignment management. Administrator/user will be able to add, view and calculate salary in salary management. These are all the functions and features that are used by a logged in administrator/user.

## **3.2.2 Sequence Diagram**

Sequence diagram can be defined as operations that are carried out and interactions captured between objects in the content of collaboration. It is one of the most important diagram in design.

**Notations Used**

|  |  |  |
| --- | --- | --- |
| S.N. | Symbol | Name |
| 1 |  | Actor |
| 2 |  | Lifeline |
| 3 |  | Message |
| 4 |  | Self-Message |
| 5 |  | Recursive Message |
| 6 |  | Reply |
| 7 |  | Alt.Combined Fragment |

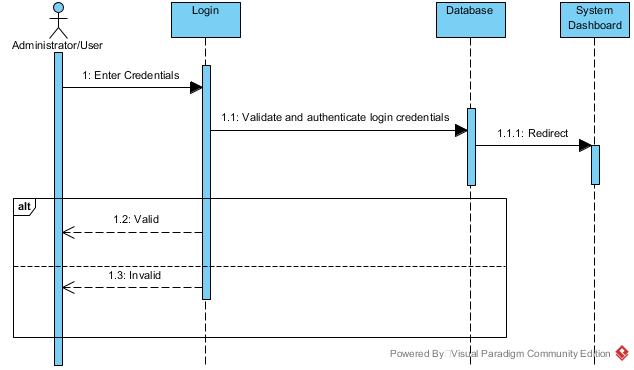


Figure 10 Sequence Diagram for Login

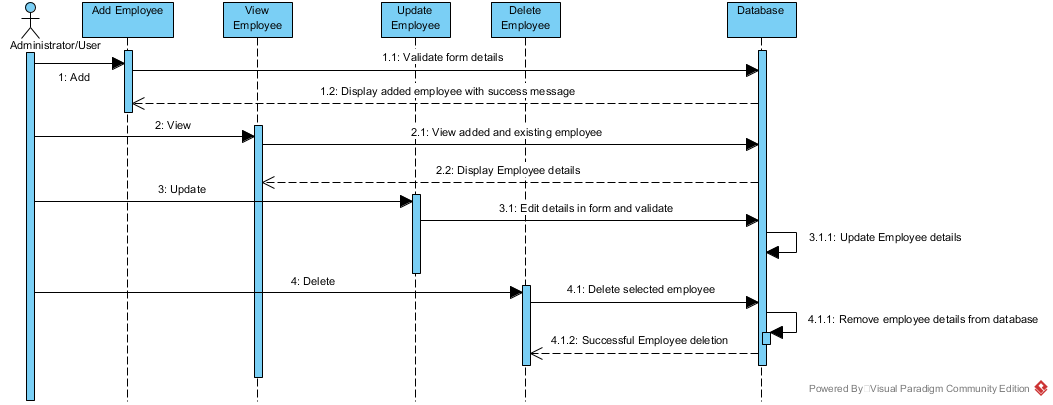


Figure 11 Sequence Diagram for Employee Mgmt.

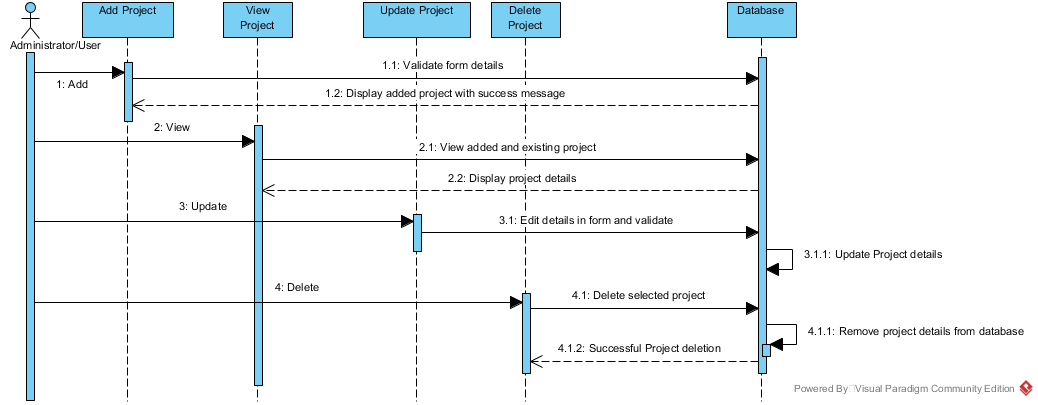


Figure 12 Sequence Diagram for Project Mgmt.

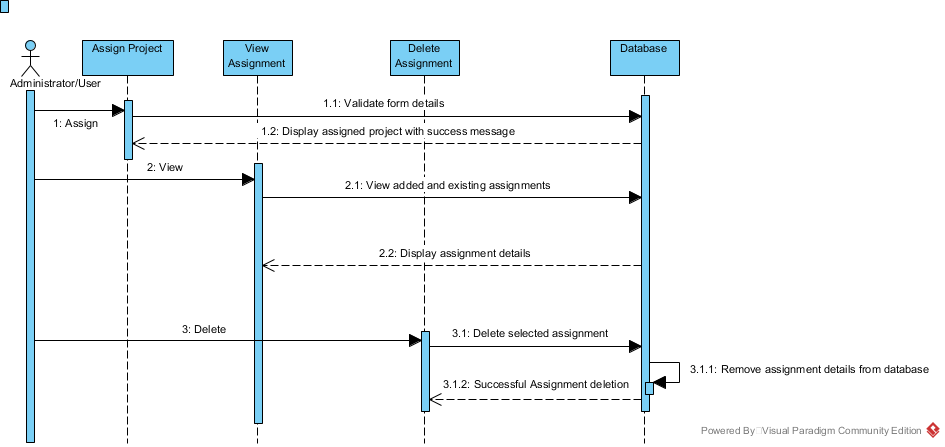


Figure 13 Sequence Diagram for Assignment Mgmt.

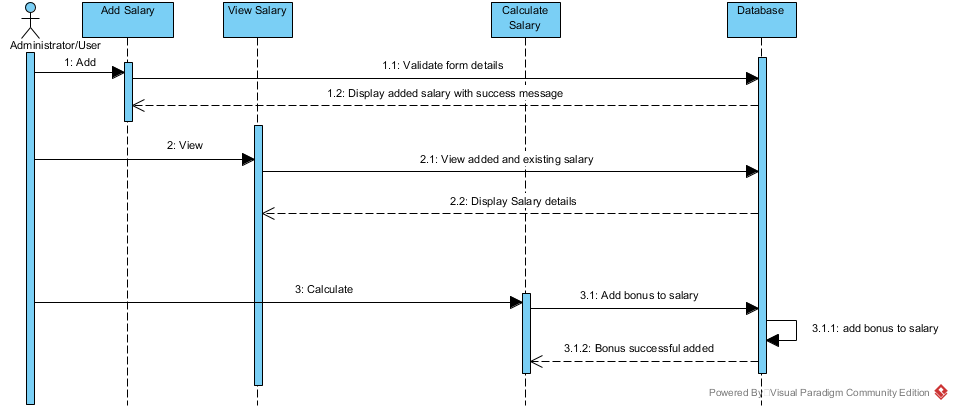


Figure 14 Sequence Diagram for Salary Mgmt.

All the interaction between the administrator/user and the system are represented in the diagrams above. It shows how the features will be used by the administrator/user. The diagram represents the interactions that will happen between the administrator/user and the system. These diagrams shows how the system will react to the user/administrator’s input or action or command to use the functions and features of the system. It shows the sequential process of how the functions are carried out in the system.

## **3.3 Data Modeling**

Database design can be defined as process of collecting the processes that will help the design, implementation and development of a database system.

## **3.3.1 Data Dictionary**

A data dictionary is a set of descriptions about data objects in a data model which are used in a database.

Employee Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Employee\_id | int(10) | Yes | No | No |
| EmployeeName | varchar(55) | No | No | Yes |
| Age | int(10) | No | No | Yes |
| Gender | varchar(55) | No | No | Yes |
| DateOfBirth | date | No | No | Yes |
| Address | varchar(55) | No | No | Yes |
| Qualification | varchar(55) | No | No | Yes |
| Designation | varchar(55) | No | No | Yes |

Administrator/User Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| username | varchar(255) | No | No | Yes |
| password | varchar(255) | No | No | Yes |

Project Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Project\_id | int(10) | Yes | No | No |
| ProjectName | varchar(55) | No | No | Yes |
| ProjectDescription | varchar(155) | No | No | Yes |
| Startdate | date | No | No | Yes |
| Enddate | date | No | No | Yes |

Assignment Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Assignment\_id | int(10) | Yes | No | No |
| Employee\_id | int(10) | No | Yes | No |
| Project\_id | int(10) | No | Yes | No |
| ProjectName | varchar(155) | No | No | Yes |
| AssignmentDate | date | No | No | Yes |

Salary Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Attributes** | **Datatype** | **Primary Key** | **Foreign Key** | **Nullable** |
| Salary\_id | int(10) | Yes | No | No |
| Employee\_id | int(10) | No | Yes | No |
| SalaryAmount | int(10) | No | No | Yes |
| SalaryPayday | date | No | No | Yes |
| SalaryBonus | int(10) | No | No | Yes |

## **3.3.2 E.R. Diagram**

An Entity Relationship Diagram can be defined as a visual representation of a database where tables, entities, entities type and relationships between tables. (techopedia, 2019)

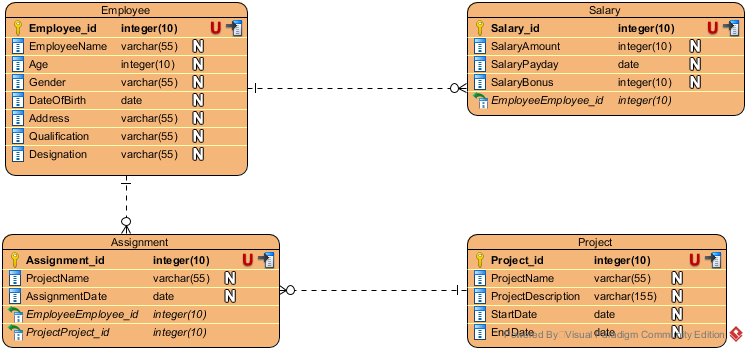


Figure 15 Entity Relationship Diagram

The ER diagram shown above is the ER diagram for the database of the system. There are four main tables in this database system. Employee, project, assignment and salary are the tables’ name. The employee tables shows the detailed information of the employees where employee id is the primary key for this table. The project table shows the detailed information of the projects where project id is the primary key for this table. The assignment table shows the detailed information of the projects assigned to the available employee where assignment id is the primary key and employee id, and Project id are foreign key for this table. The salary table shows the information of the employee's salary where salary id is the primary key and employee id is foreign key for this table.

## **3.4 Prototype Design**

Prototype design can be defined as an interactive development technique where mock-up of the UI in the system is created actively and also allow us to explore solution space for the system.

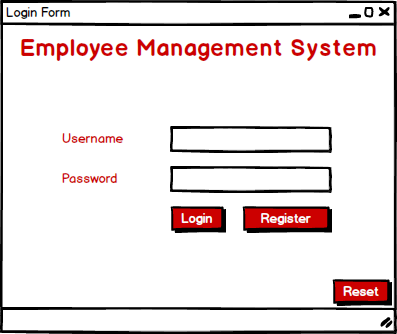


Figure 16 User Form 1

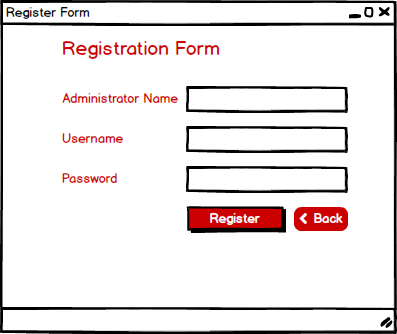


Figure 17 User Form 2

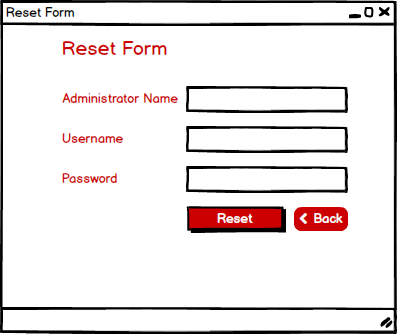


Figure 18 User Form 3

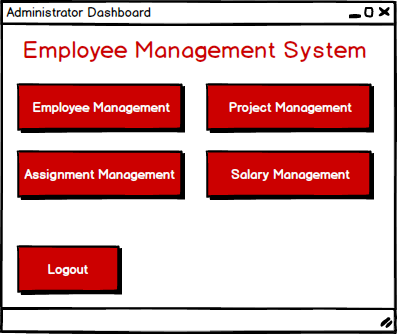


Figure 19 User Form 4



Figure 20 User Form 5

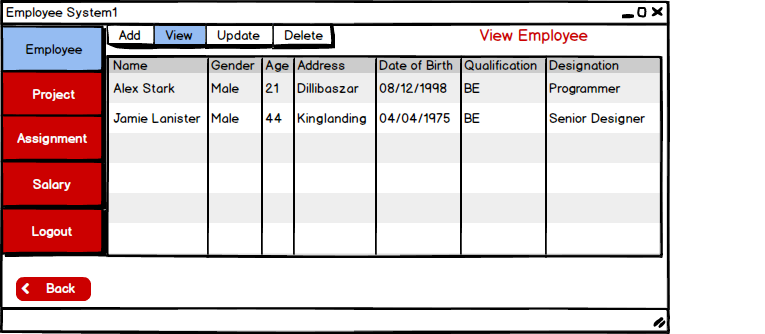


Figure 21 User Form 6

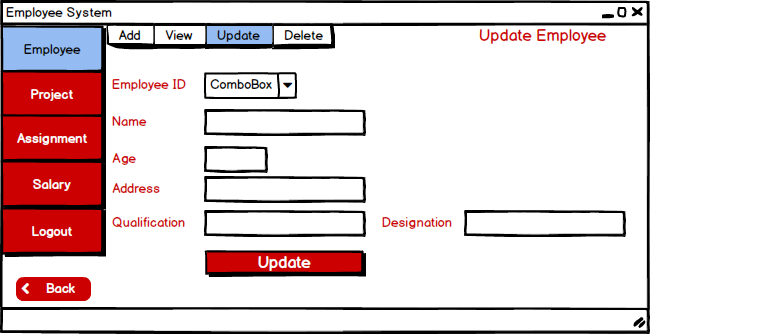


Figure 22 User Form 7

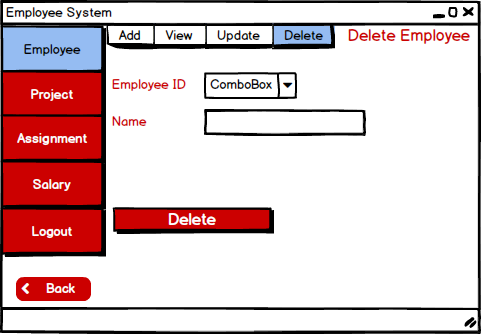
****

Figure 23 User Form 8

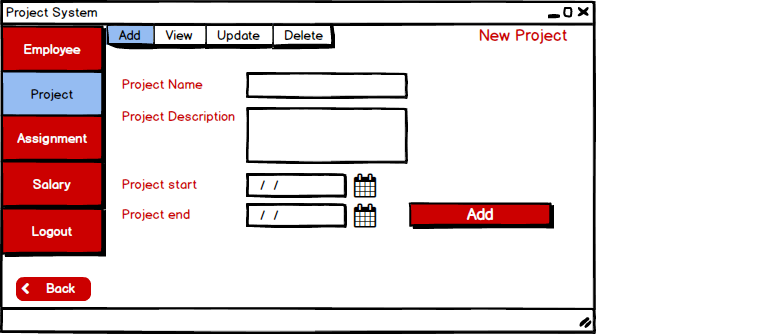


Figure 24 User Form 9

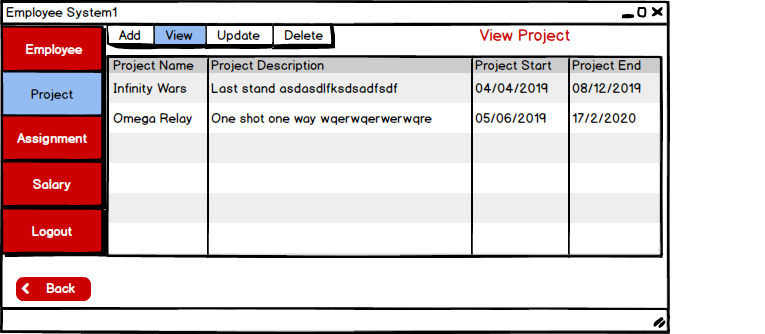
****

Figure 25 User Form 10



Figure 26 User Form 11

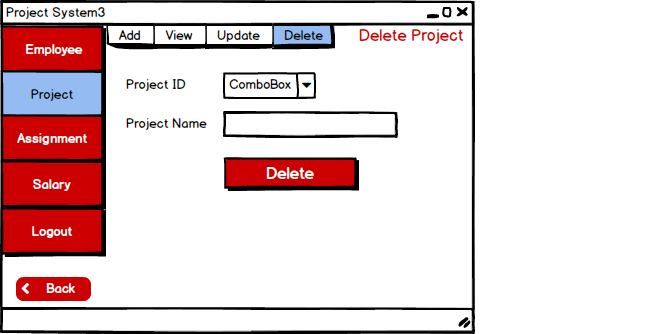
****

Figure 27 User Form 12

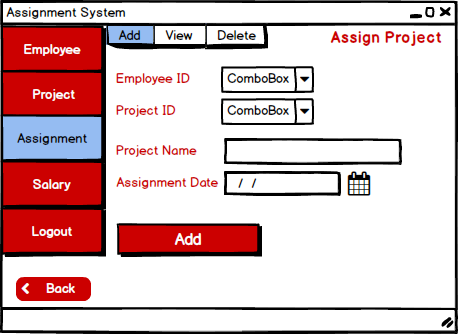


Figure 28 User Form 13

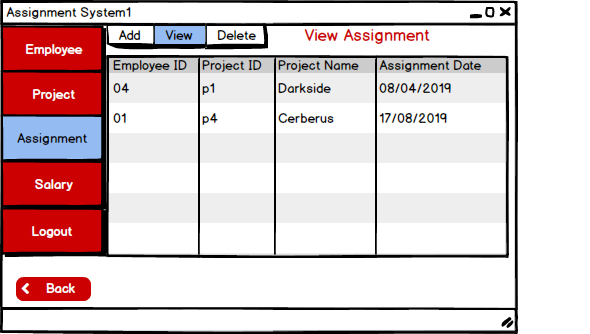


Figure 29 User Form 14

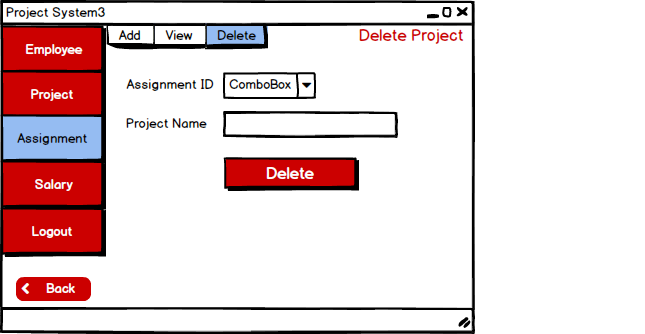


Figure 30 User Form 15

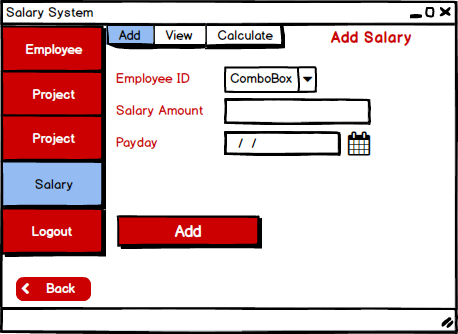


Figure 31 User Form 16

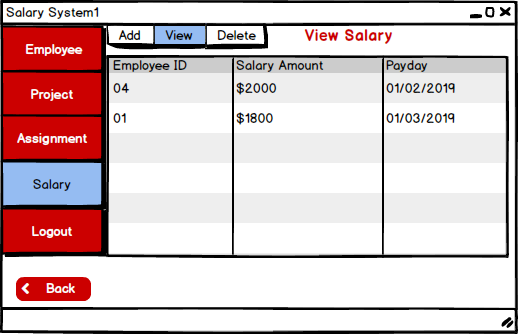


Figure 32 User Form 17

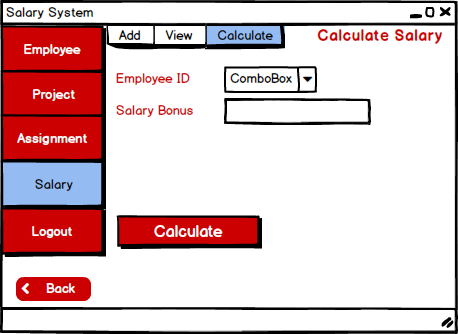


Figure 33 User Form 18

The porotype design for the system is shown above. The software I have used to create these prototype design is Belsamiq Mockup. Every forms that will be in the system is represented and also shows how the system will look and the UI it will have.

## **3.5 Architectural Model**

I have used standalone because the system I am developing is standalone application where the application is installed in a computer and are not dependent in any other servers/computers. The access of this system is limited to a system where the application is installed. Some of the advantages for using standalone application for the system are:-

* There is no need to networking and only requires single system to operate the application.
* Local files can be created and manipulated.
* Does not require internet connection to operate.
* Execution can be done independently and produce output as a UI.
* It can be defined as local service development because it can operate independently.

The system is created for small organizations where small task can be performed to negate simple issue and to make management easier. The system is standalone as it is more focused on providing services for small and compact organization and not for large corporation. The management of the system will be easy and will operate independently. It is easy to deploy and errors can be dealt in short period of time.

# **Chapter 4 Implementation**

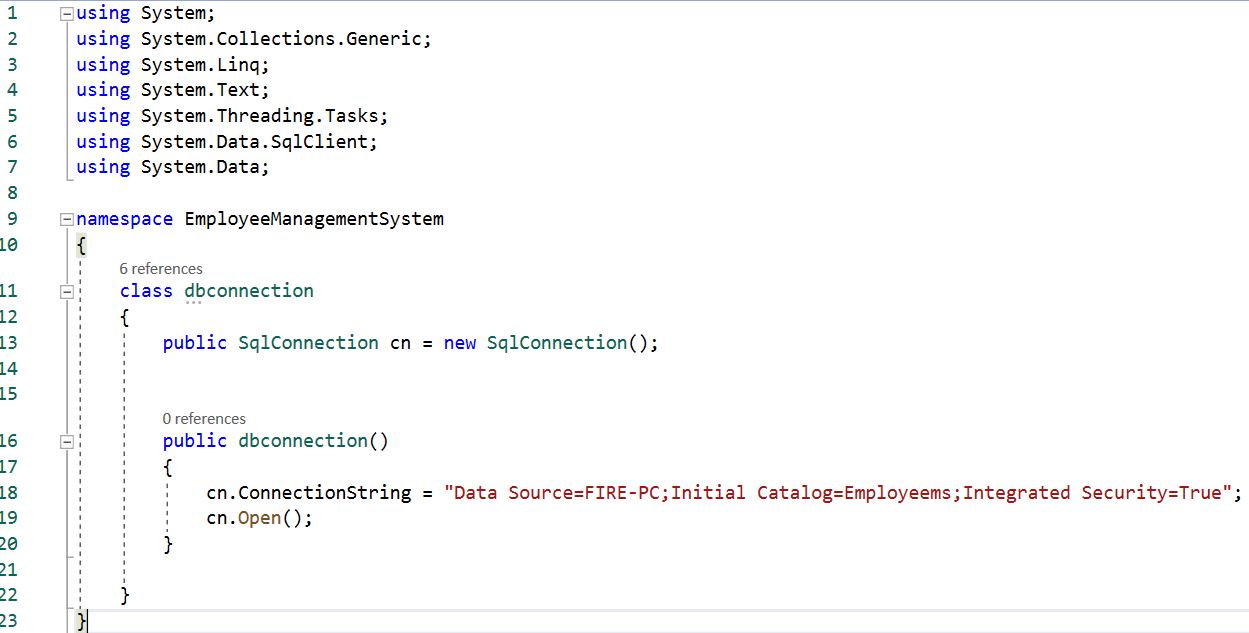
Third phase of the (SDLC) System Development Life Cycle is implementation. The phase is started after the completion of Analysis and Design. After all the analysis and design of the system is completed, implementation and development of the system is started. To convert the design of the system and implementing into code using programming language is the major role of this phase/stage. If the implementation is done correctly and is well-developed then we will face less issues and errors in the next phase/stage (Testing).

Visual Studio c#.net is the programming language used for the development of the Employee Management System. The framework I have used for this system is .net. I have used Model-View-Control (MVC) for the development of this system. The other tools I have used is Microsoft SQL Server as relational database management system for data server.

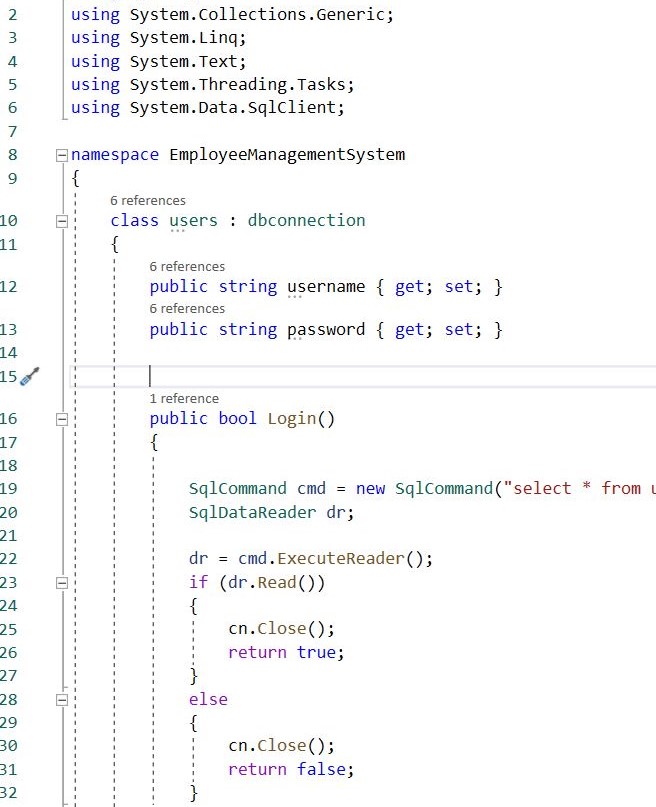
Necessary database and tables for Employee Management System were created. A class (dbconnection) was created to connect the database server so that the details entered were stored in the database and their respective tables.

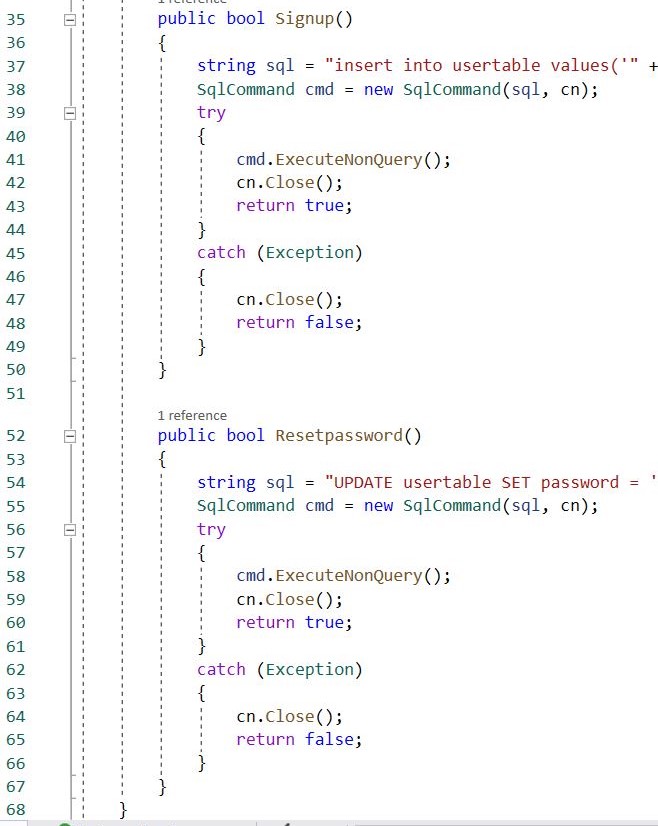
The screenshots of Classes and its respective Form and its code are shown below:-

## **dbconnection:-**



## **Users:-**

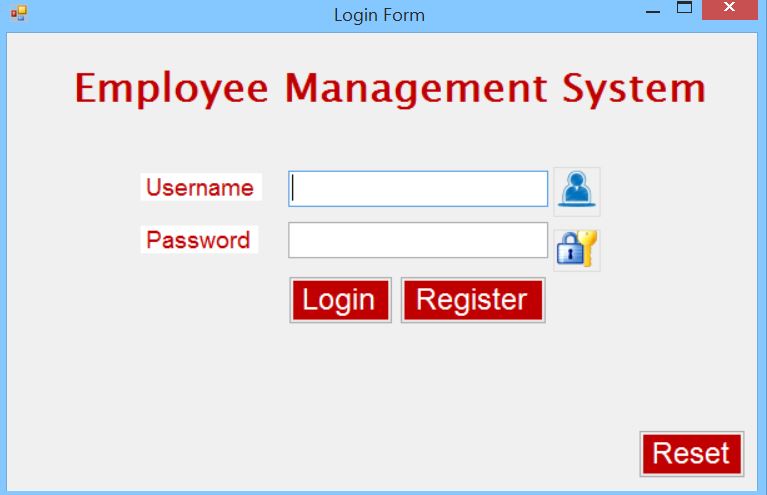




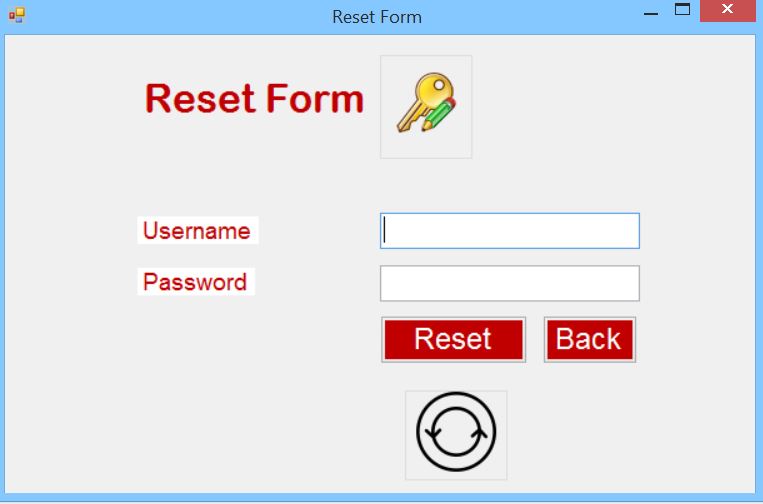
**Forms/Codes**

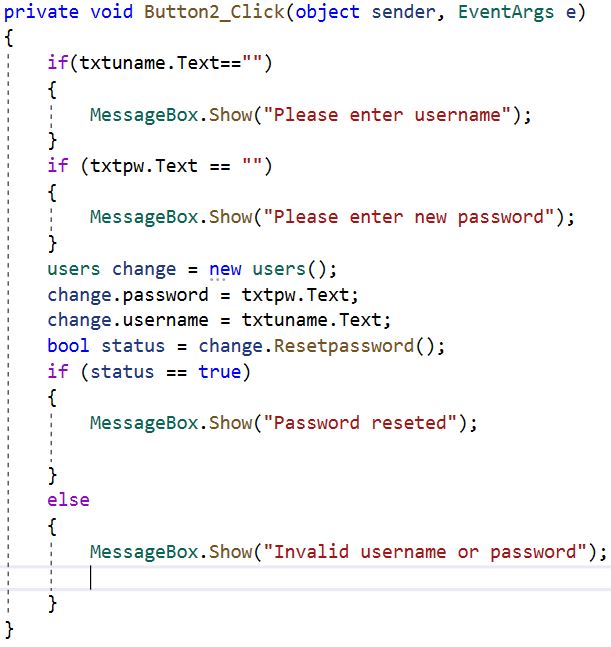


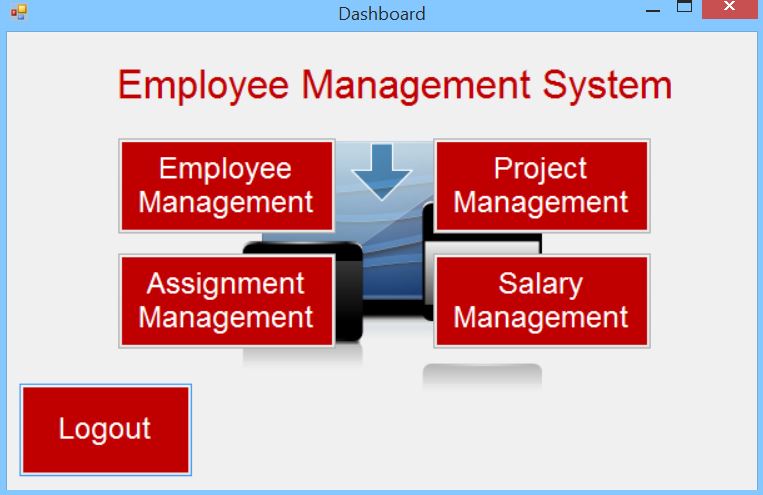




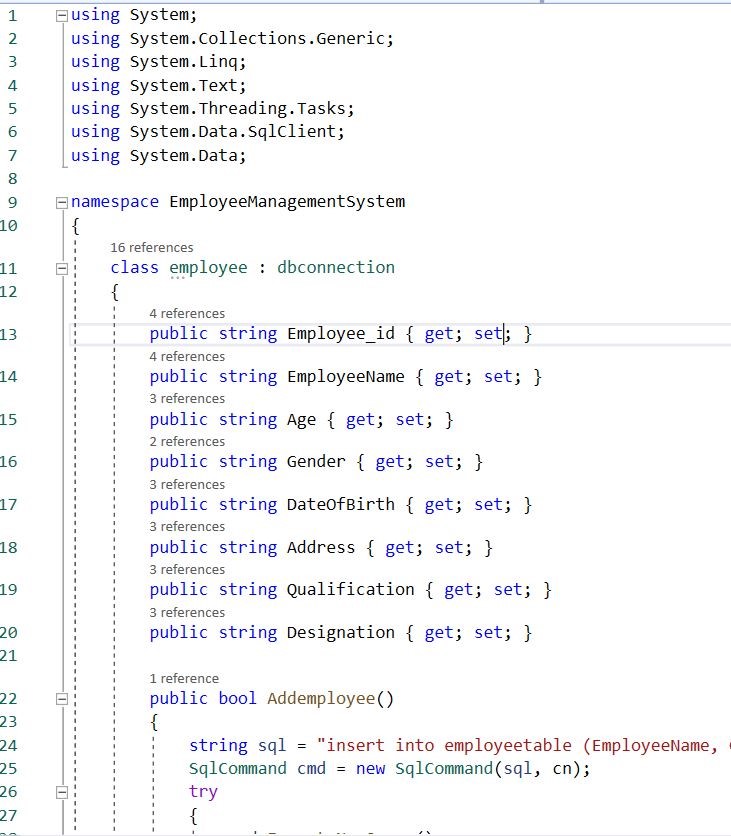


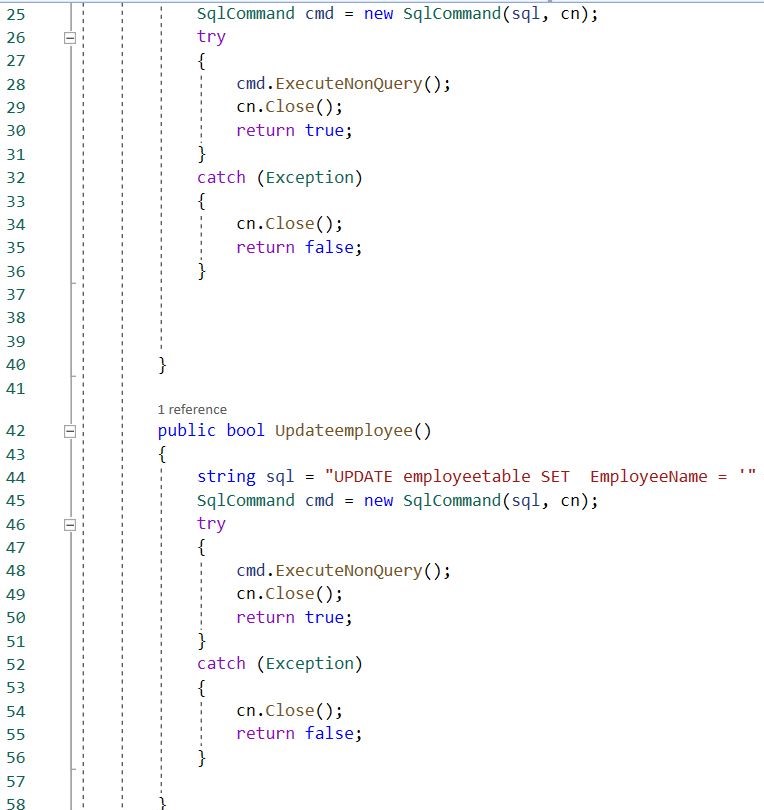






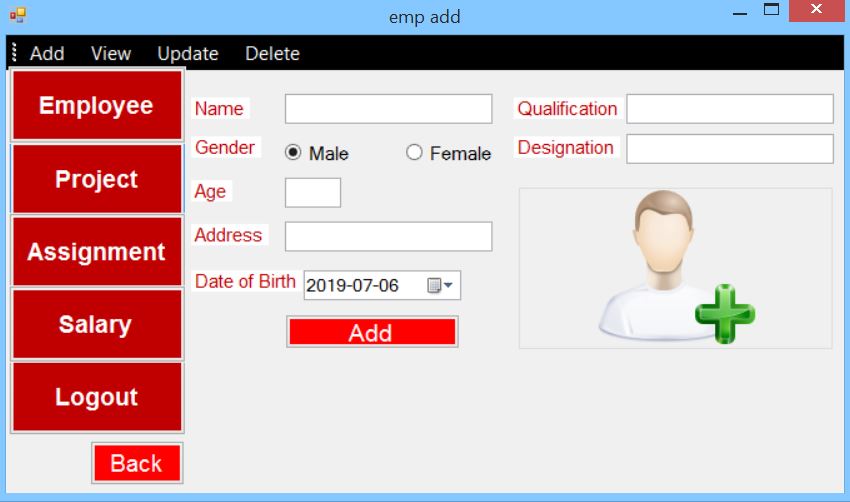
## **Employee:-**

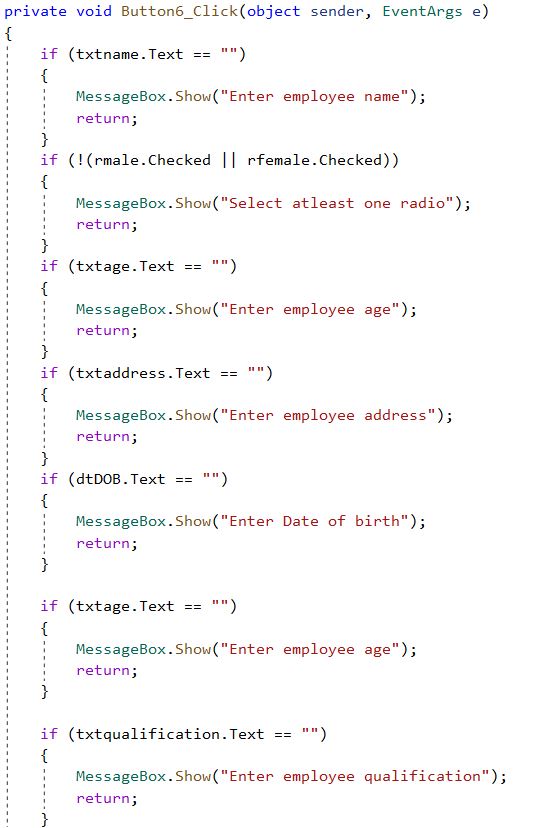


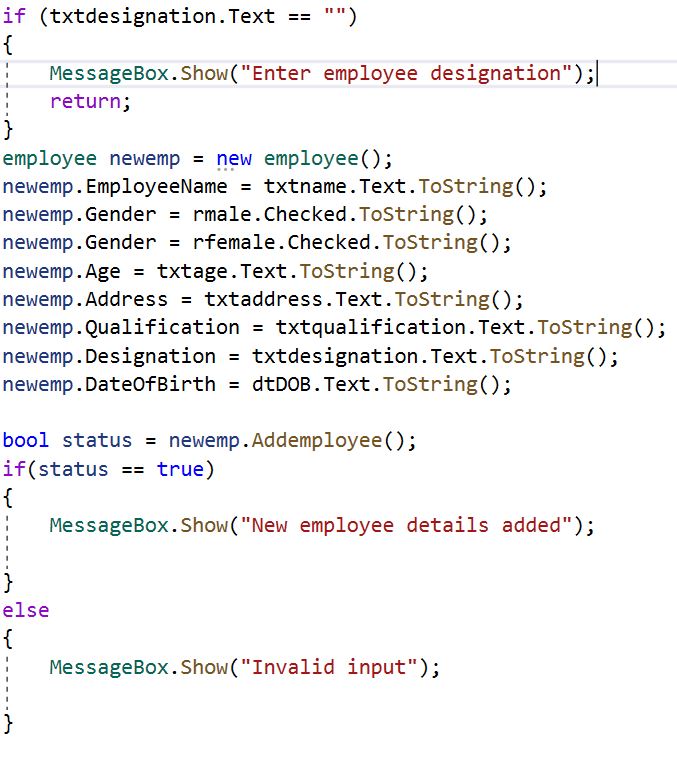


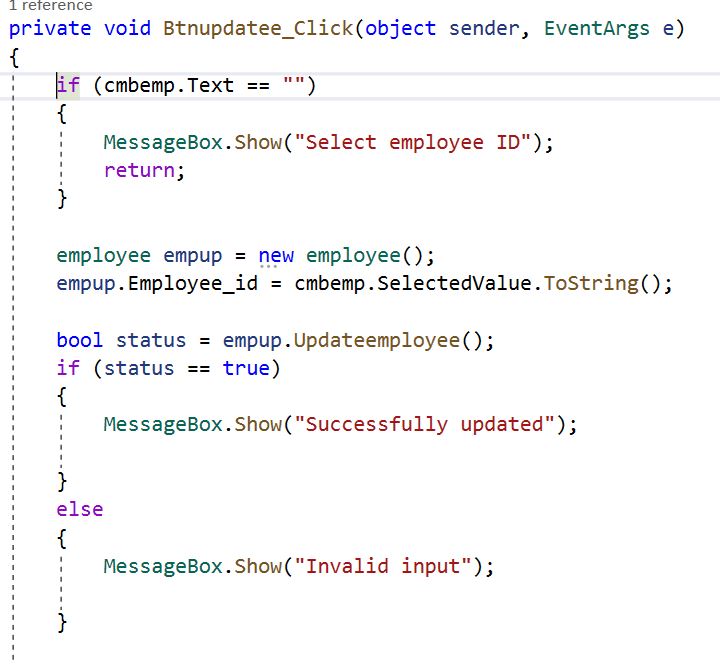
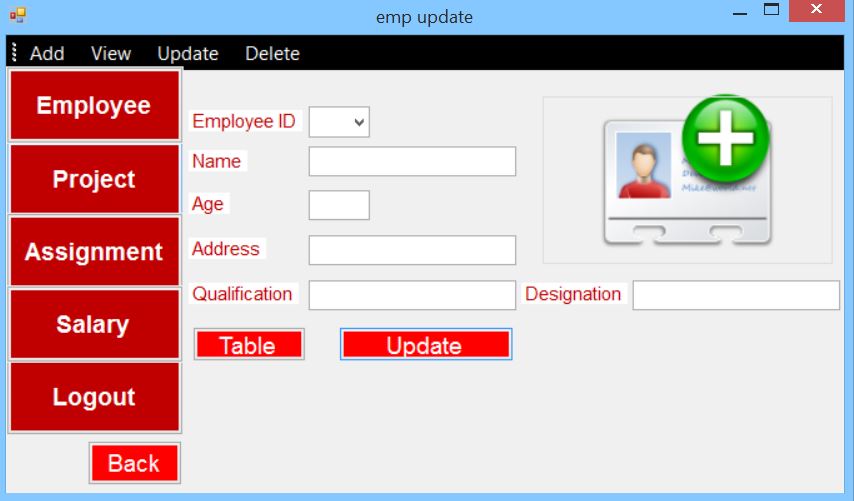


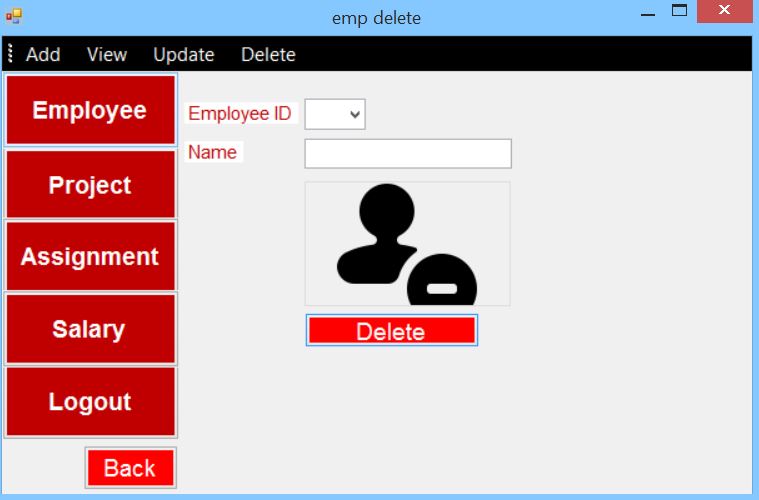
**Forms/Codes**

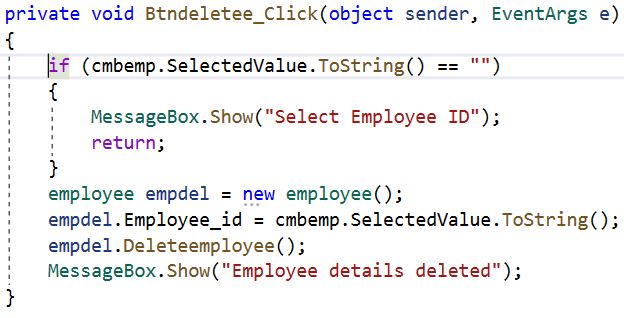


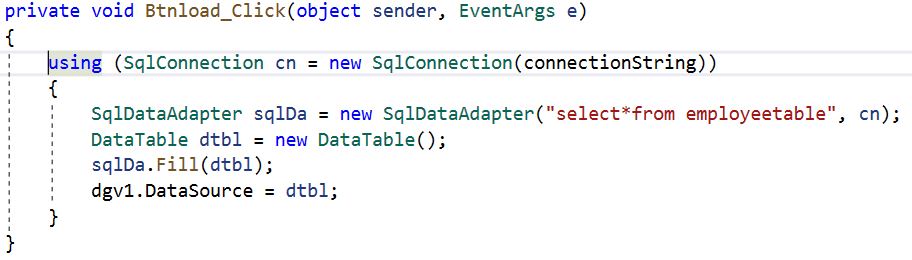
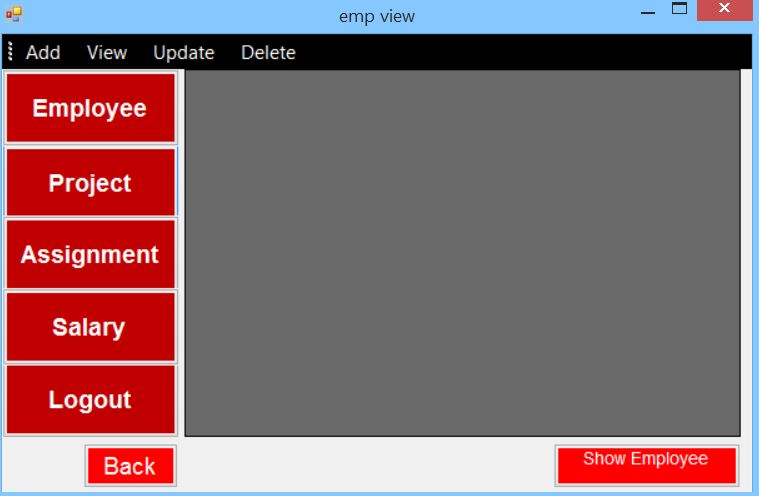




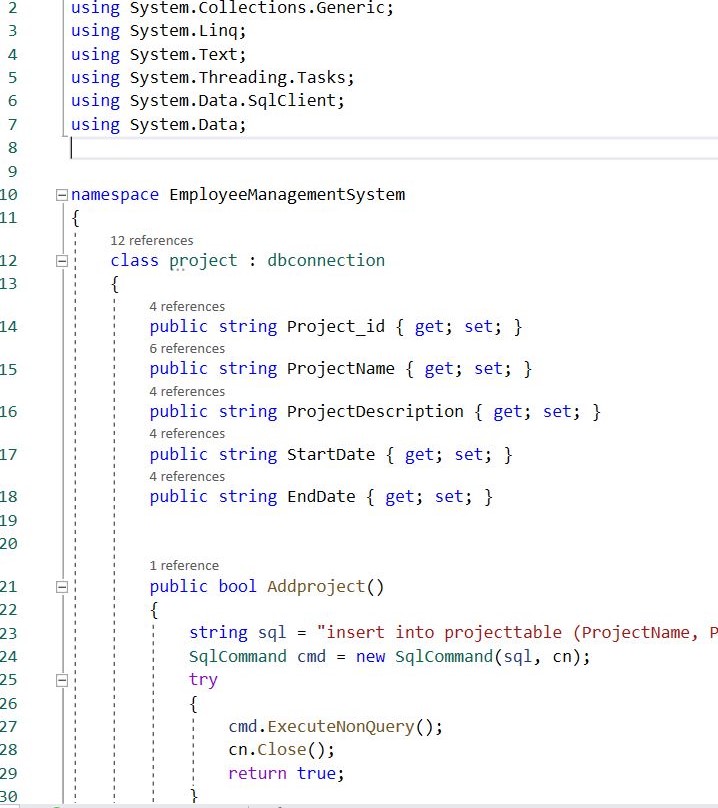








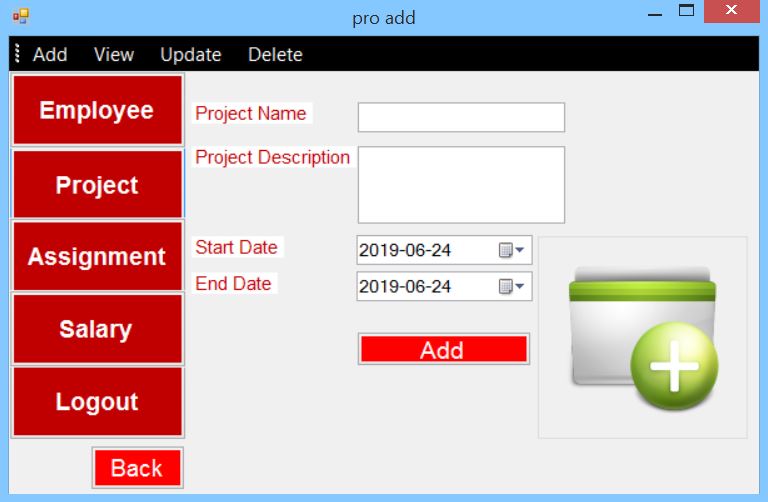
## **Project:-**

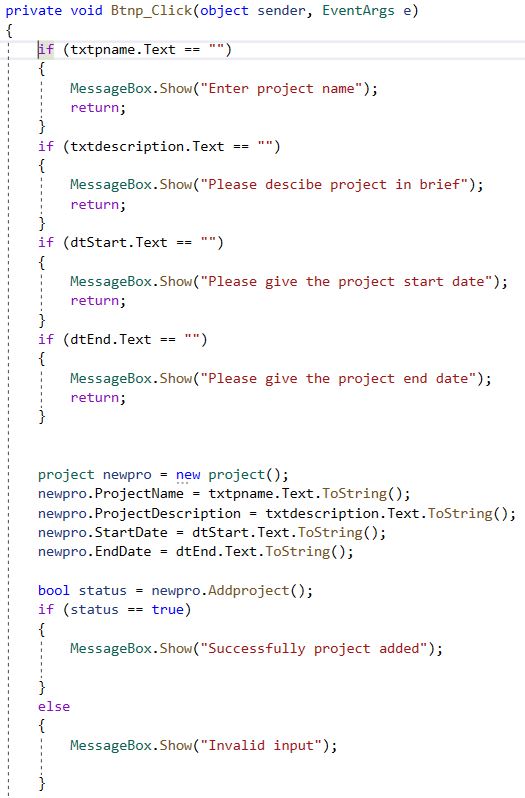


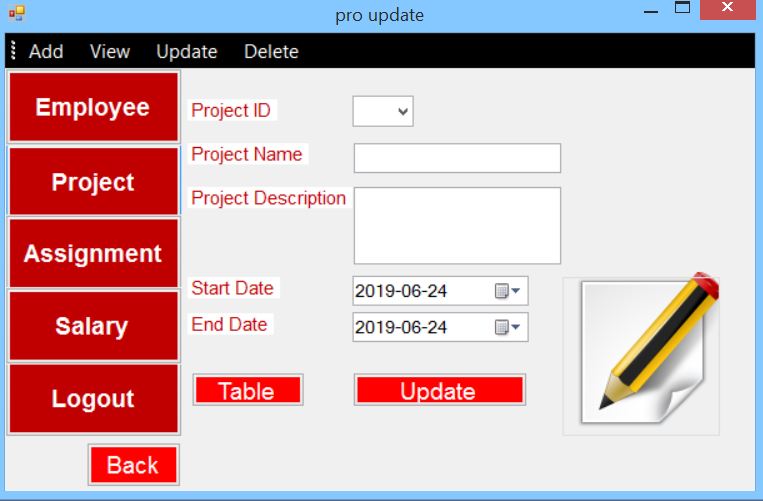


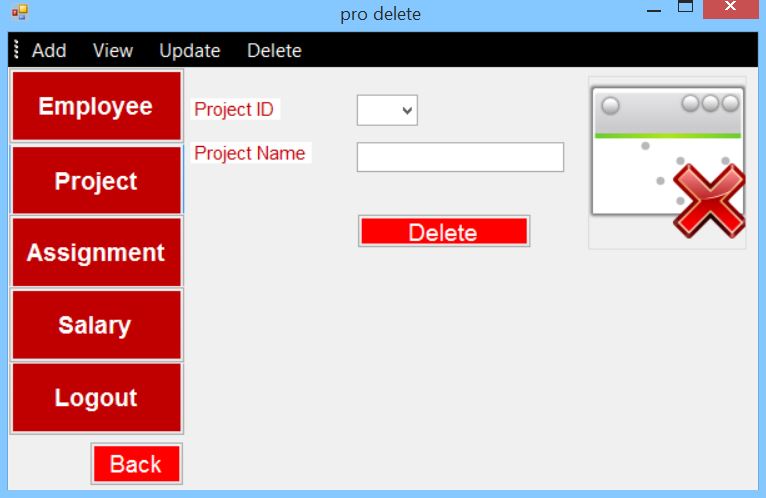


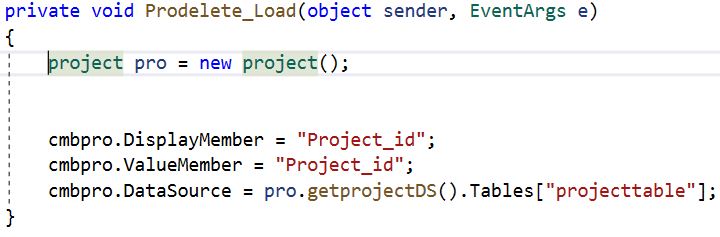
**Forms/Codes**

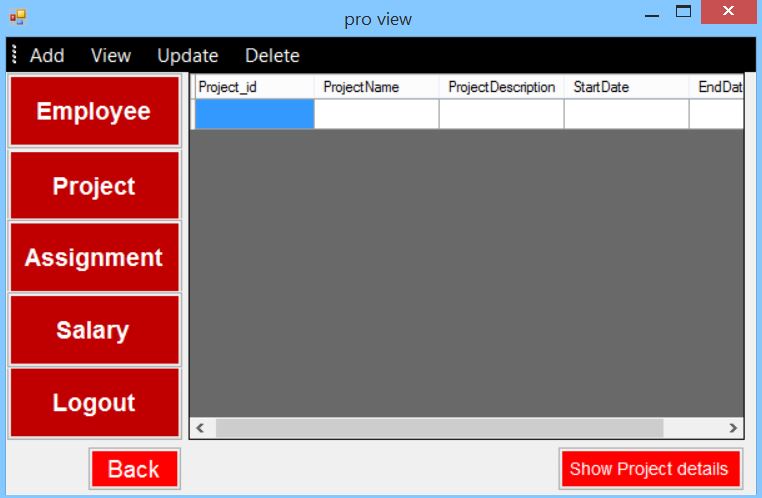






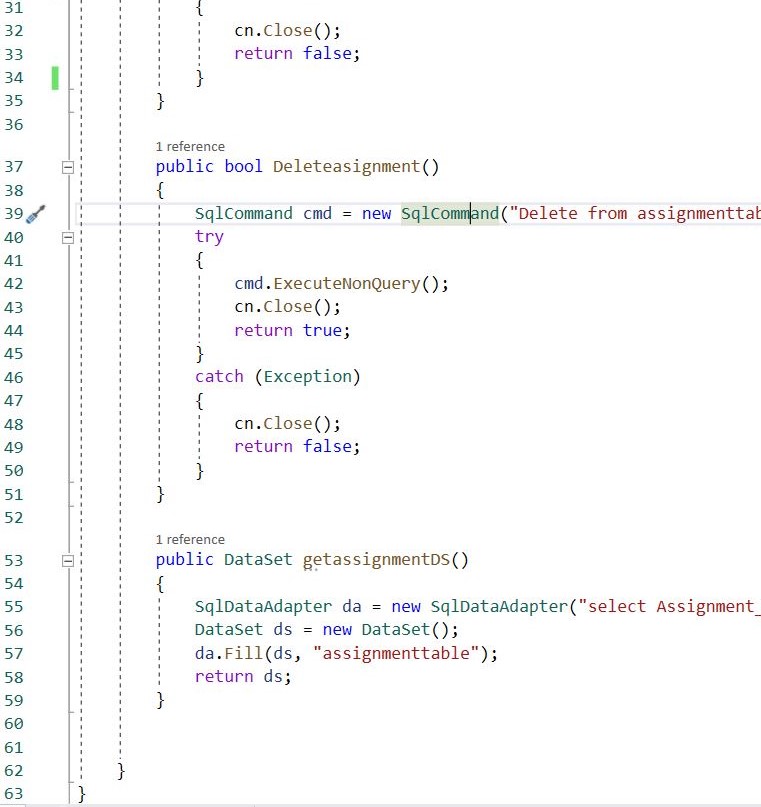




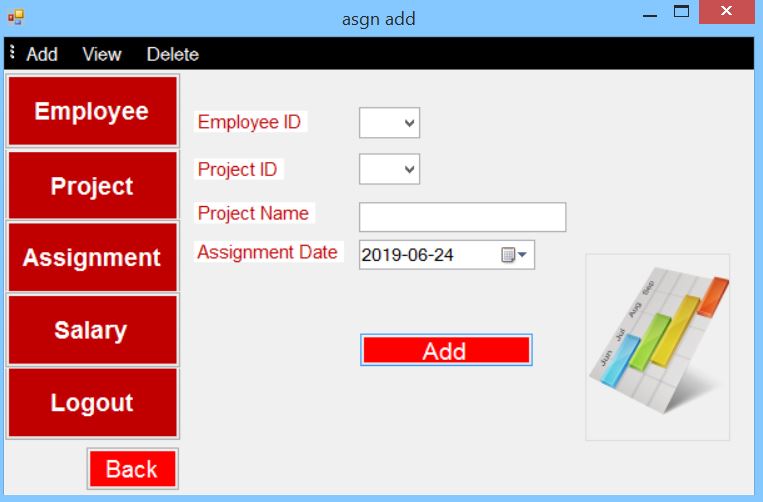


## **assignment:-**



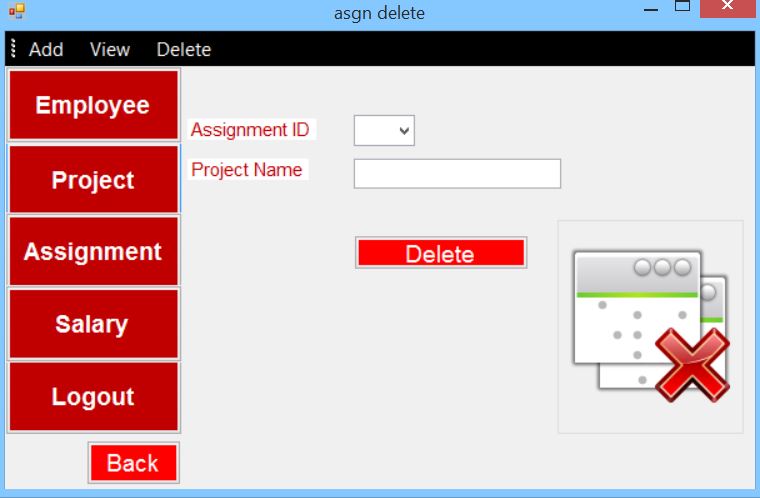


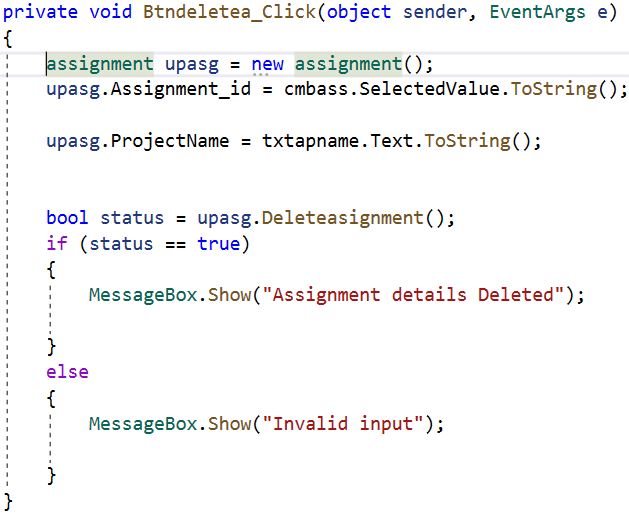
**Forms/Codes**

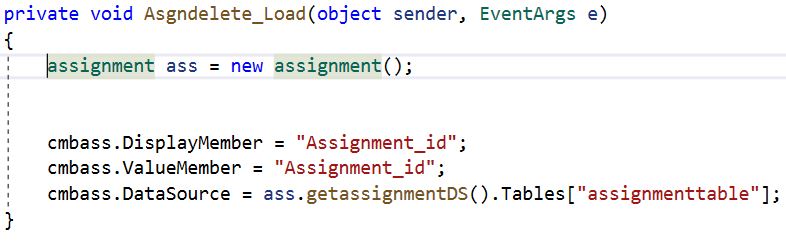


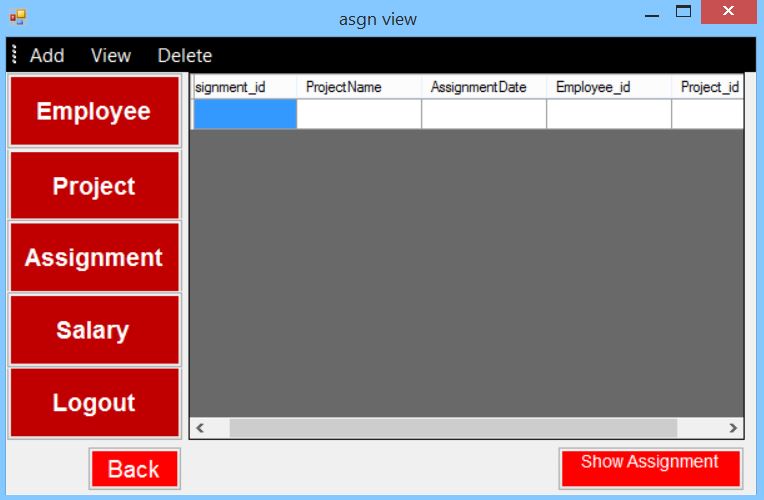


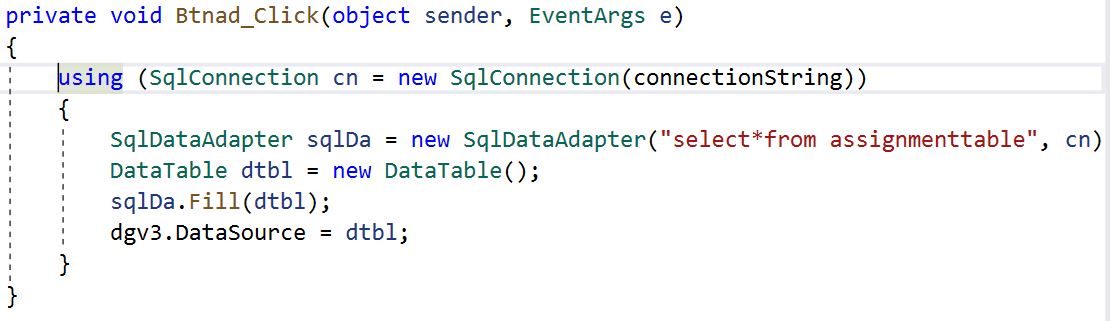




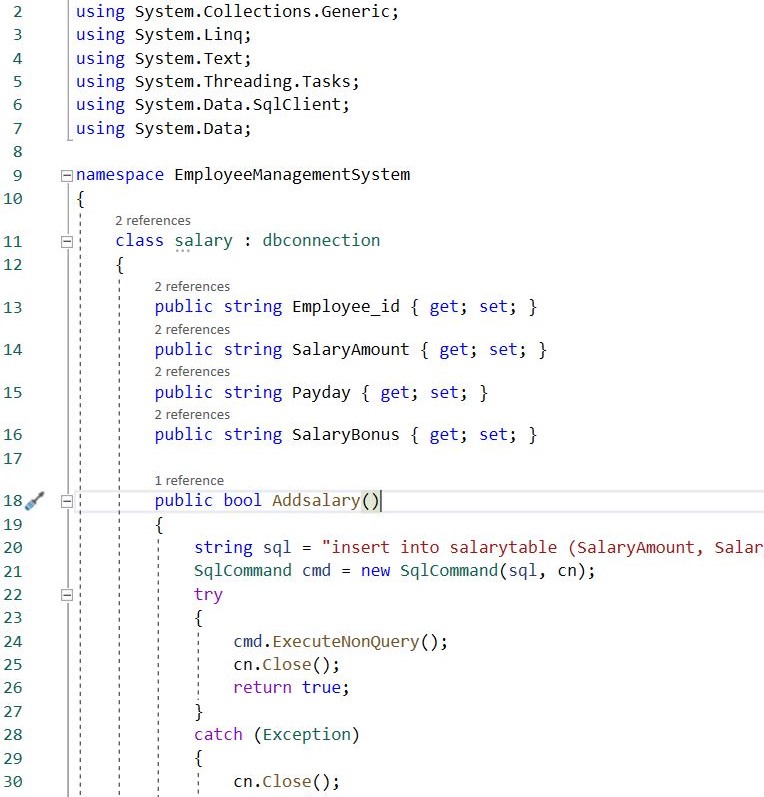


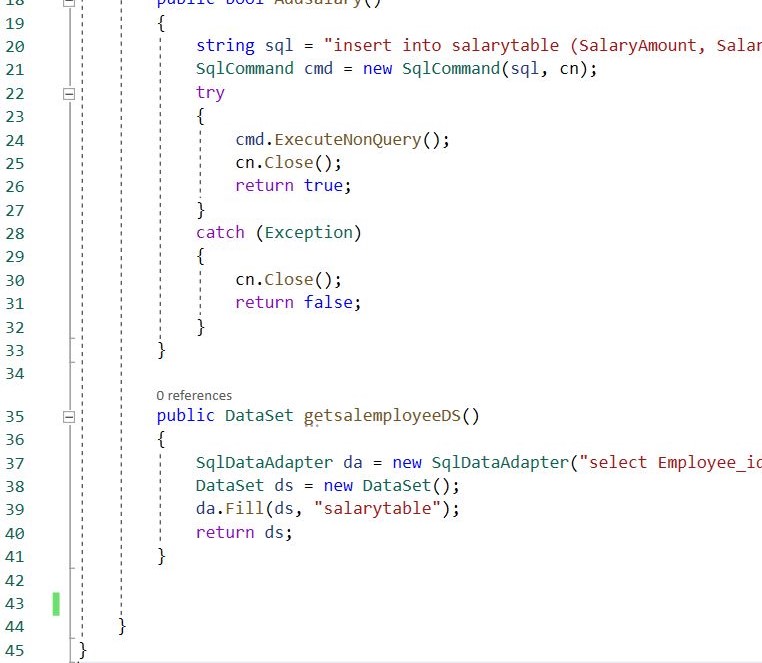




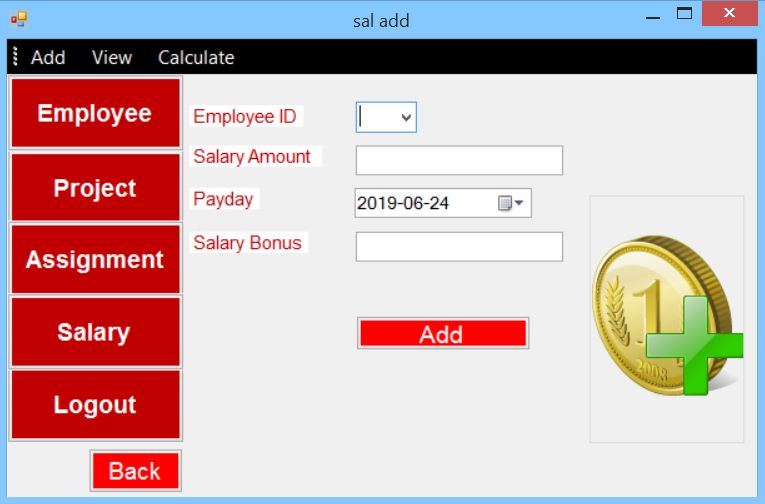


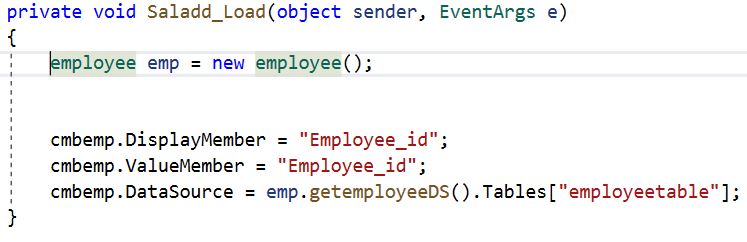
## **salary:-**

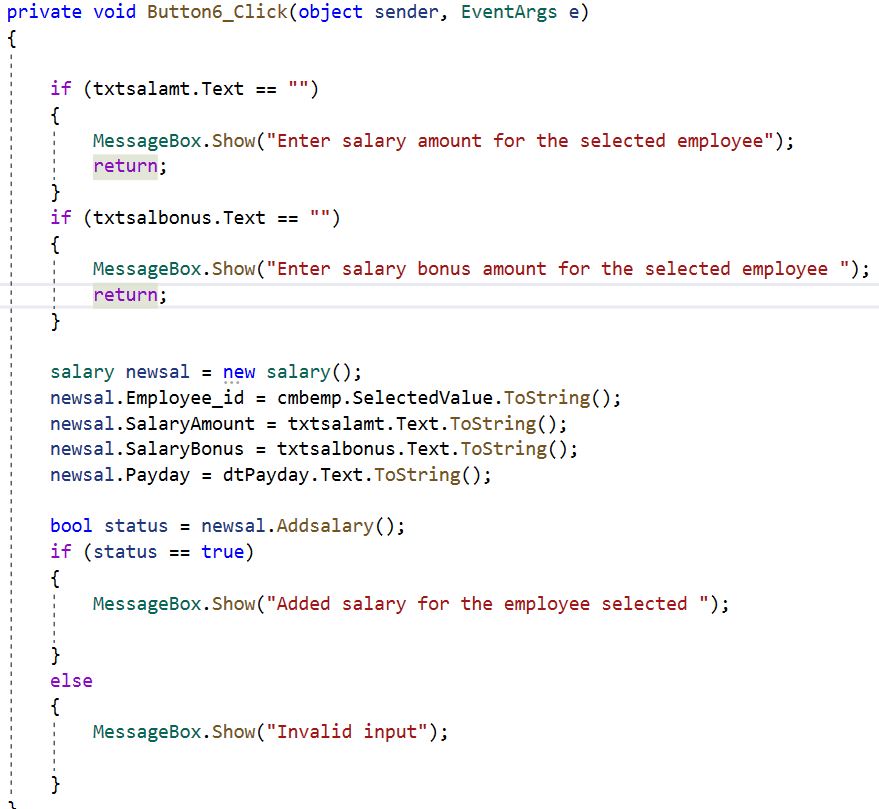




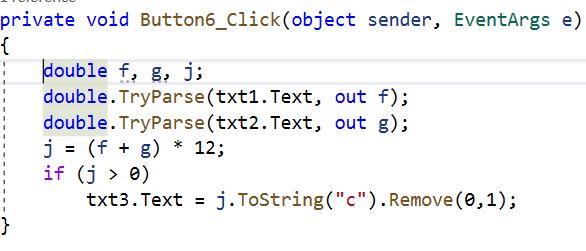
**Forms/Codes**

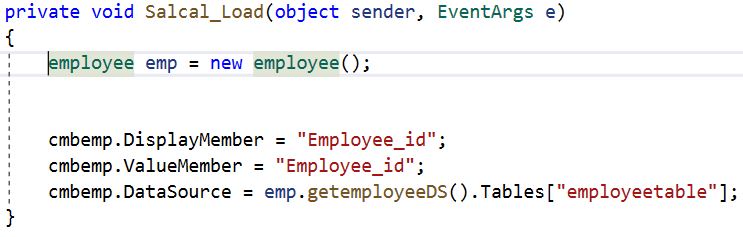
****

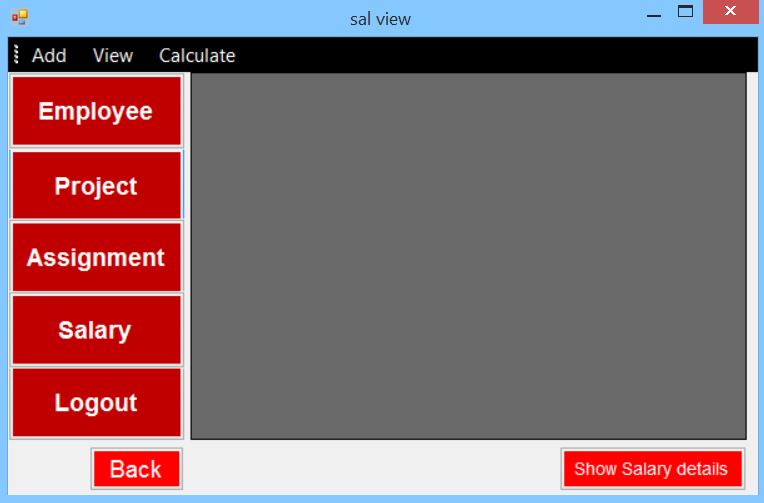
****

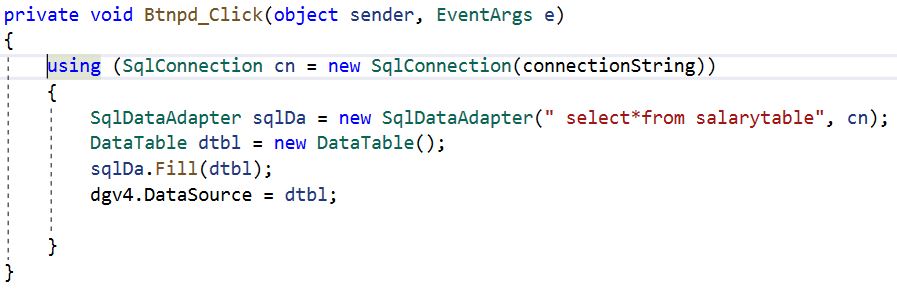
****

****

****

****

****

****

# **Chapter 5 Testing**

## **5.1 Testing Introduction**

The fourth stage/phase of the SDLC is testing. The process to identify and find out bugs and errors in a program is called software/application testing. (tryqa, 2019) Programs and applications can be verified and validated by software testing. One of the important stage/phase of SDLC is testing. Different errors are identified during testing and these errors can be fixed after the testing is successfully done. Actual result/output and expected result/output is also compared and evaluated. Some of the testing used in software testing are:

* Blackbox Testing
* Unit Testing
* Performance Testing
* Regresstion Testing

Blackbox Testing and Unit Testing are the testing I have selected to perform for this project.

5.2 Blackbox Testing

It is a software testing technique where the tester is not concerned with the internal code structure of the software but with the actual and expected result. Both functional and non-functional can be performed but mostly functional is performed in blackbox testing. It is also known as behavioral testing.

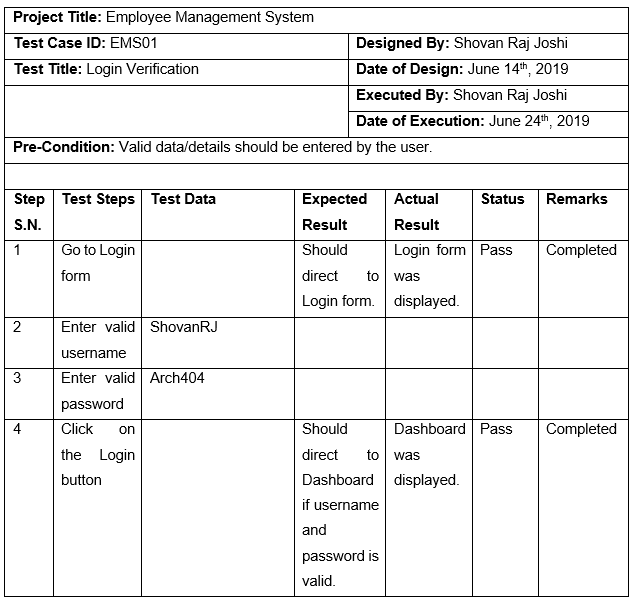
**Test Plan**

To test the features of the employee management system, a test plan has been created. At least (10) ten features will be tested.

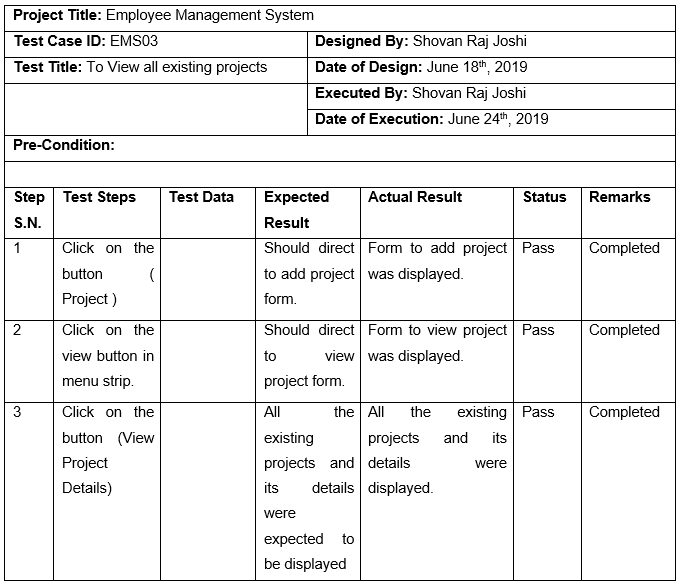
Features tested are:

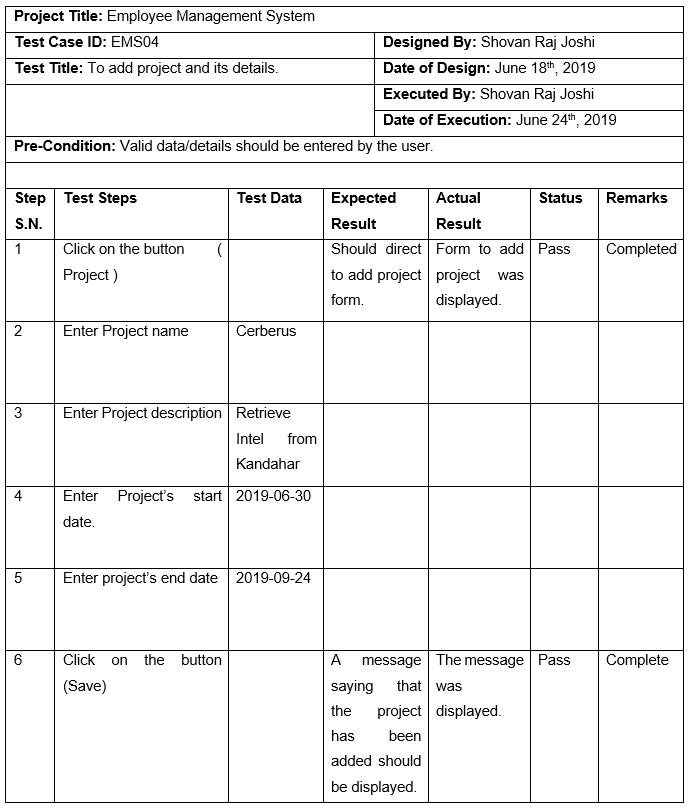
* Register/Login
* CRUD Functions

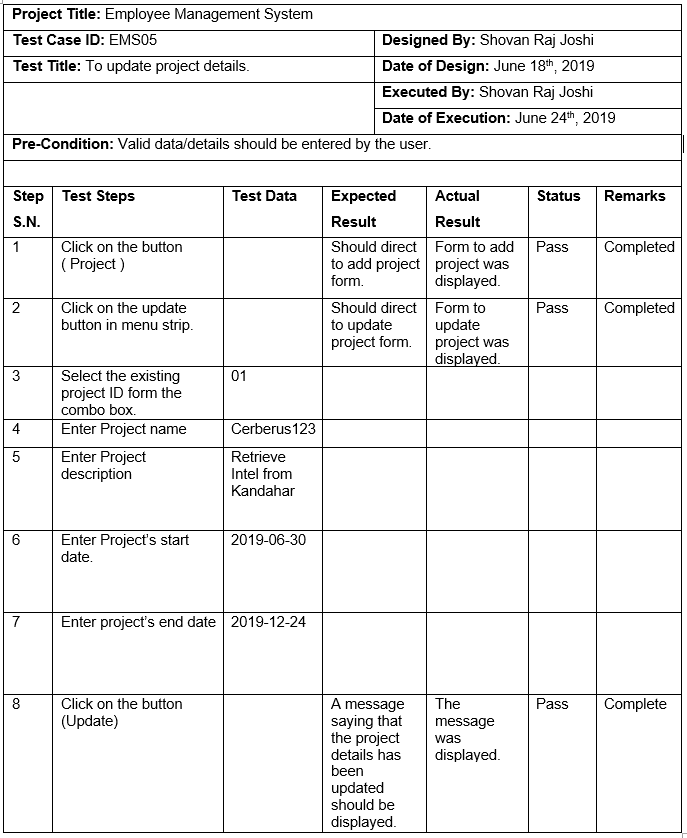
**Test Cases for User/Administrator:**

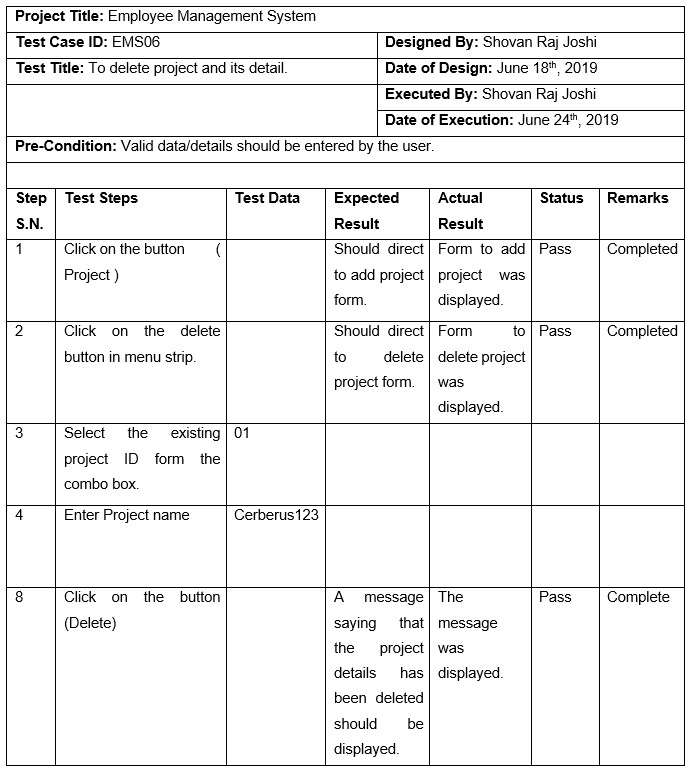


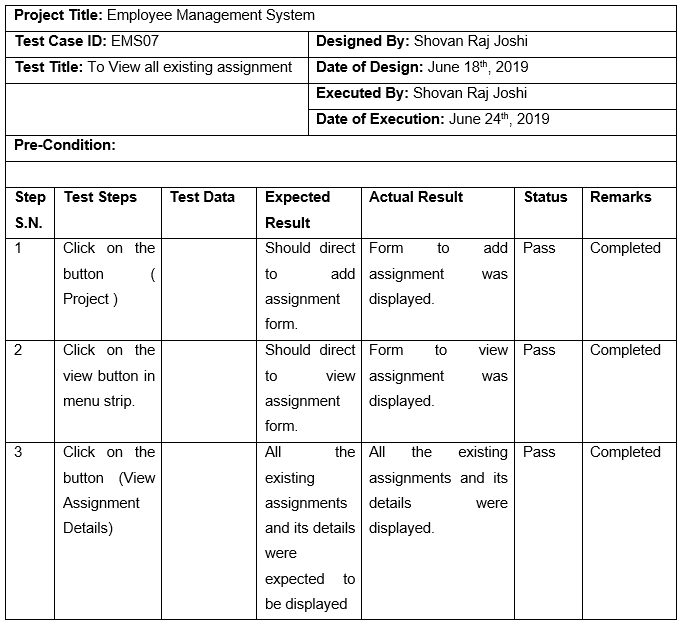


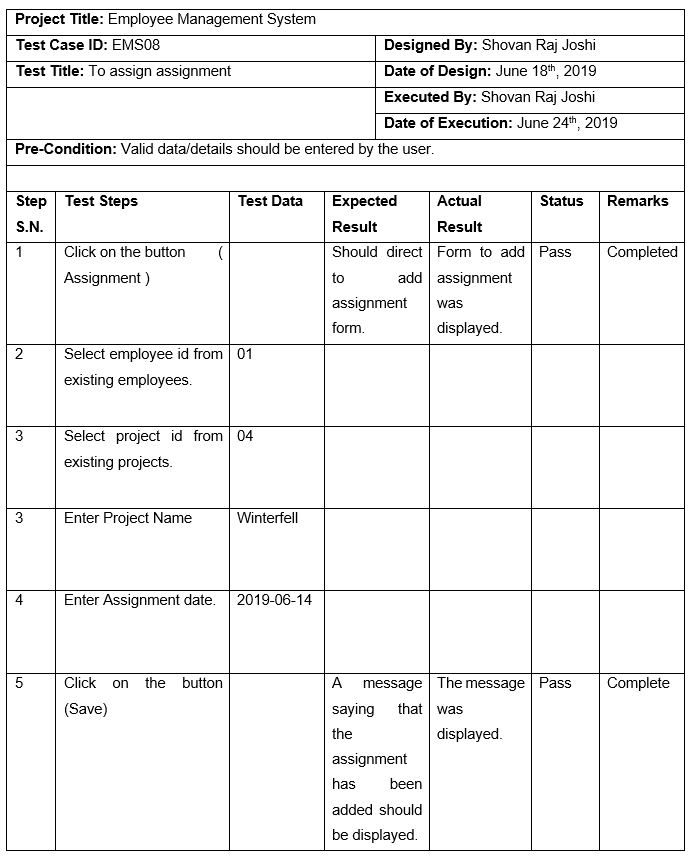


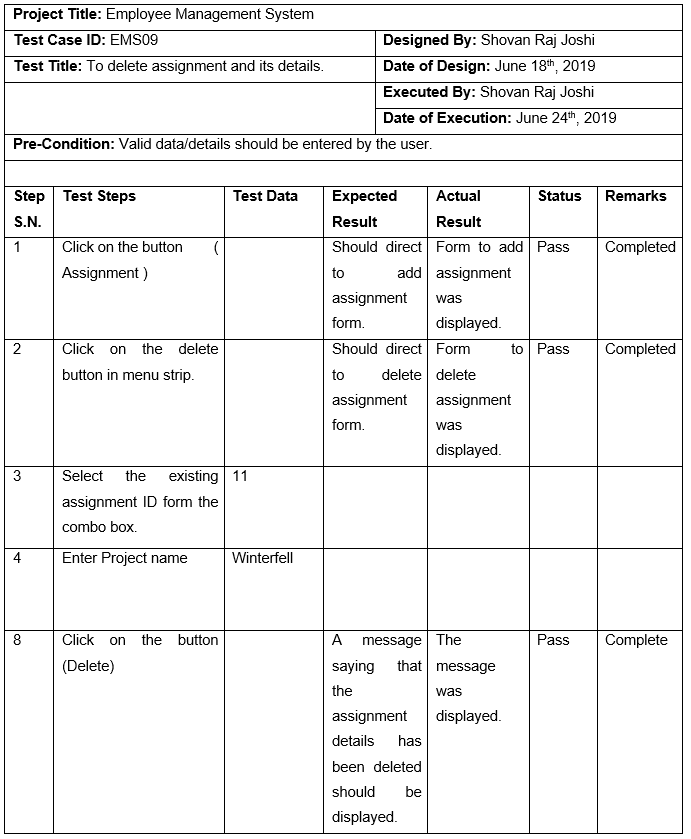


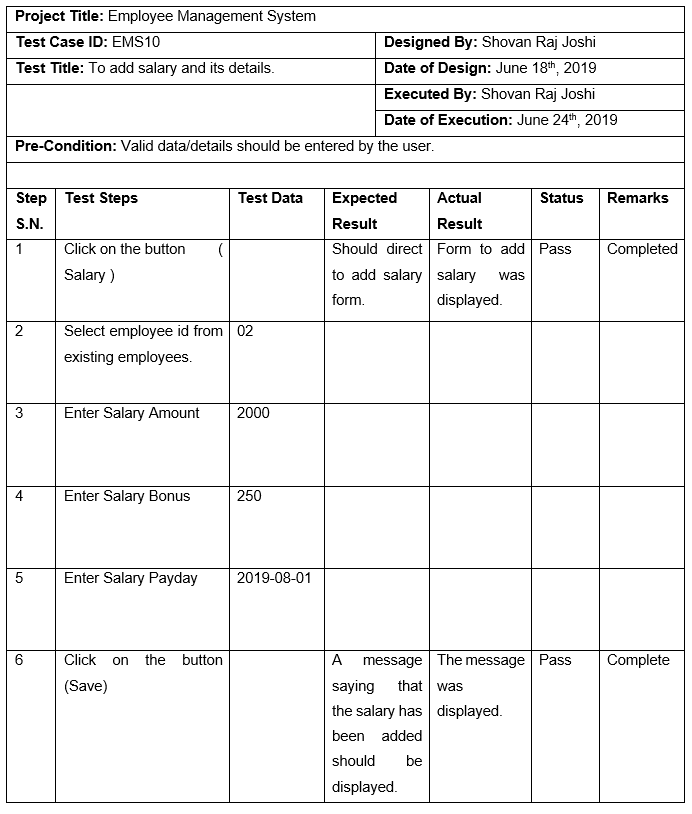












## **5.2 Unit Testing**

Individual units or components of the software are tested in unit testing. The main purpose of this testing is to test the software’s each unit is performed according to the design and validate them. Unit testing is very important is helps to evaluate the software. All the testing was passed and was successful.

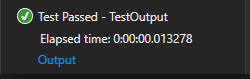
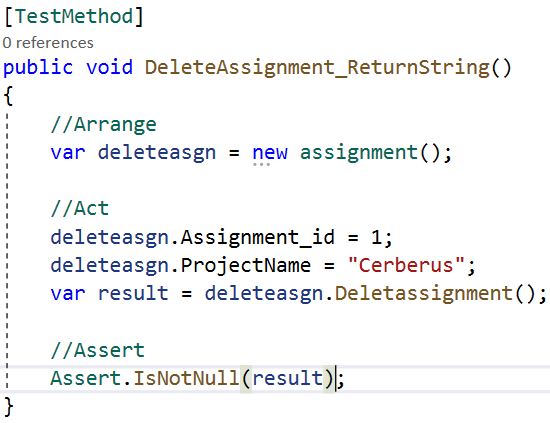


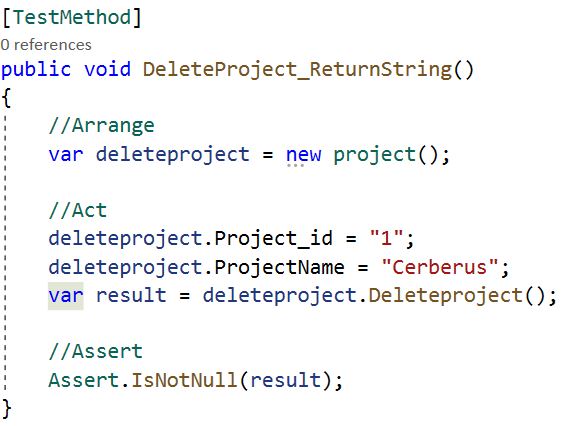
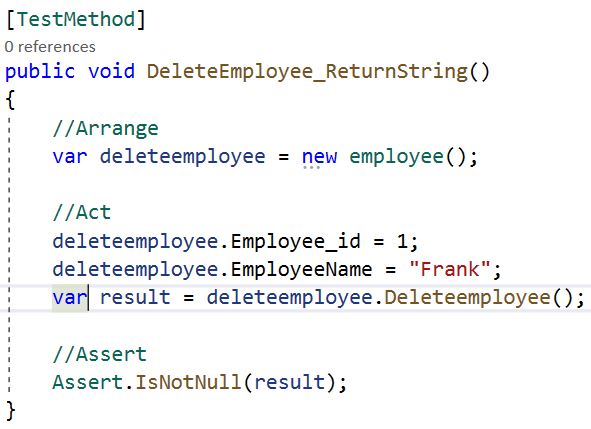
Figure 34 Unit Test Passed

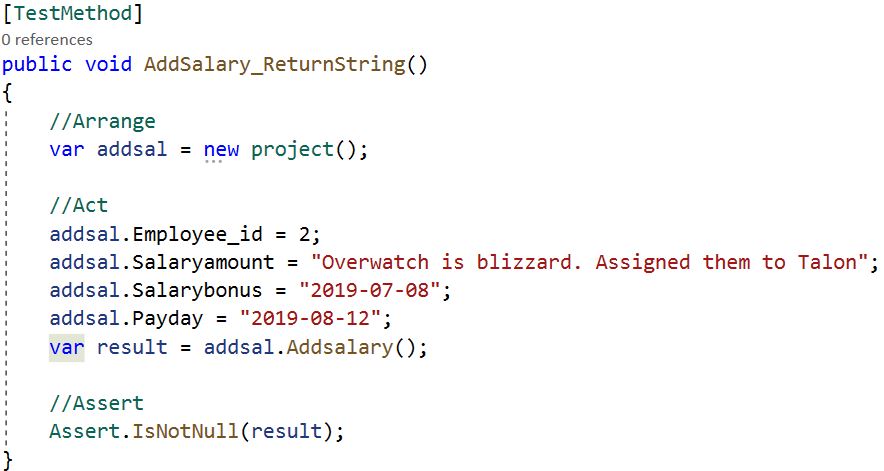
Register Unit testing. Login Unit testing.



Delete Assignment Unit testing

Delete Project Unit testing. Delete Employee Unit testing



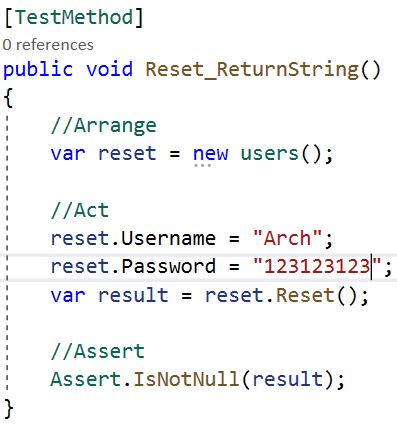
Add Salary Unit testing.



Add Project Unit testing.



Update Project Unit testing.

Reset Pasword Unit testing. Add Assignment Unit testing.

# **Chapter 6 Other Project issues**

## **6.1 Project Limitation**

Some of the limitations for this project are:-

* **Security**

As this is a standalone application, the employee management system does not have the highest amount of security. The security should be focused because the system contains important and vital information of the company’s project, employees and their salary.

* **Compatibility**

The system may not be compatible with all the computer and some computer may not meet the requirements to run the system effectively. The system may also not be compatible with old windows. It is also not compatible with mobile.

* **Backup**

As it is not a web based or cloud based system. The data is stored only in the computer where the employee management system is available. So to avoid the risk of data loss back up should be done regularly or constantly so that it can be recovered easily.

## **6.2 Future Work**

The project was developed successfully however few requirements and features have not been added. Employee management system can contain many more functionality by adding other features. The features that can be added in the employee management system in the future are:-

* Add attendance management.
* To search employee, project and salary details through name.
* To view which employees took leave.
* Implement better security process.
* Add department management
* To view the employees who got bonus.

## **6.3 Other Project Issues**

During the development of this system the project will face different issues and problems. The issue and problems faced during the development of this project should be addressed in order to avoid and prevent these issues while working on future projects. The few issues that needed to be dealt were:-

* Some requirements needed to be removed or changed.
* Time taken for implementation/coding for employee management system was difficult to manage.
* Lack of experience on developing a fully functioning application/system.
* Lack of knowledge on C# .net framework.

## **6.4 Risk Management**

There is possibility of risks in every project. Risk management is the process that identifying the risk and sorting or solving those threats and risks from causing further damage to the project.

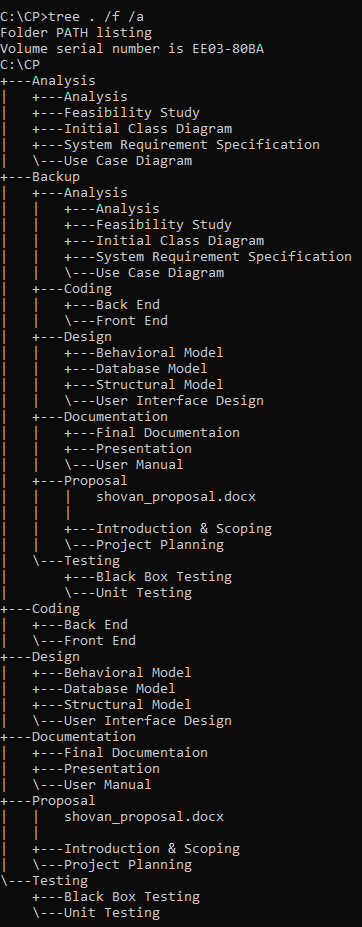


**Impact**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.N.** | **Risk** | **Consequences** | **Likelihood** | **Impact** | **Action** |
| 1 | Time Management | 4 | 2 | 8 | Different tasks are properly scheduled and time is managed accordingly. |
| 2 | Compatibility | 2 | 2 | 4 | Test are done in various windows and computer having different specs. |
| 3 | Requirements changed or removed | 4 | 3 | 12 | Analyzing and reviewing requirements again. |
| 4 | Infeasible Design | 5 | 1 | 5 | Re-designing the project according to new requirements. |

## **6.5 Configuration Management**

My files are broken-down according to Work Breakdown Structure. Files are kept in their respective folders. All the folders and files are also kept in a backup folder. Also, files and folders are constantly update whenever there is change made.



## **6.6 User Manual**

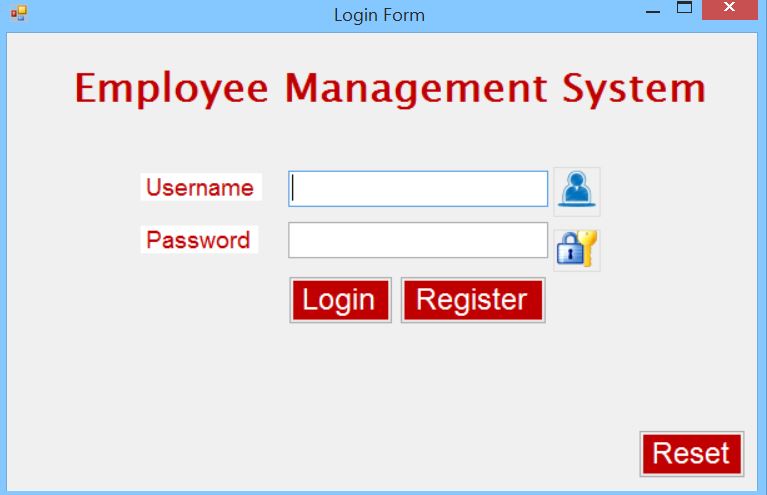
User Guide document that is provided by the suppliers of the system which help the user guides them how to make the best use of the system and its function. Necessary steps and instruction to use the system is also provided.



How to register to use the system:

* Enter a username and password.
* Click on Register button.

After registering, a message will be displayed saying that the registration was successful.

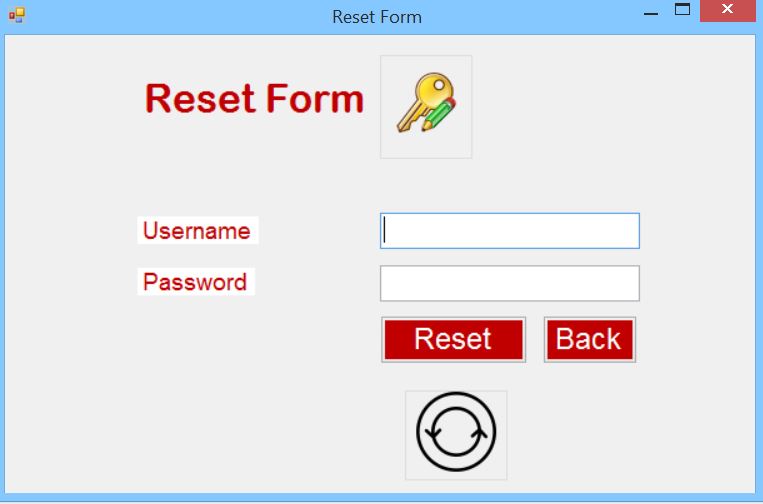


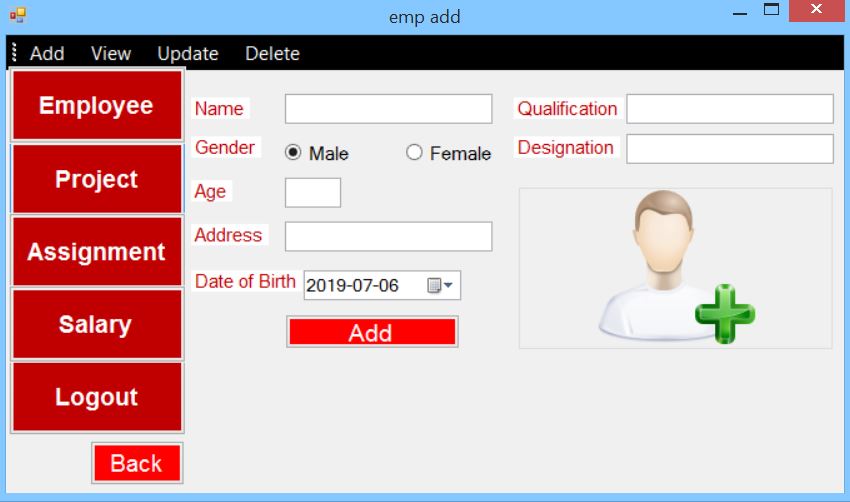
How to login to access the system:

* Enter your username and password.
* Click on login button.

After entering valid details you will be able to gain access to use the employee management system and will redirect you to its dashboard.

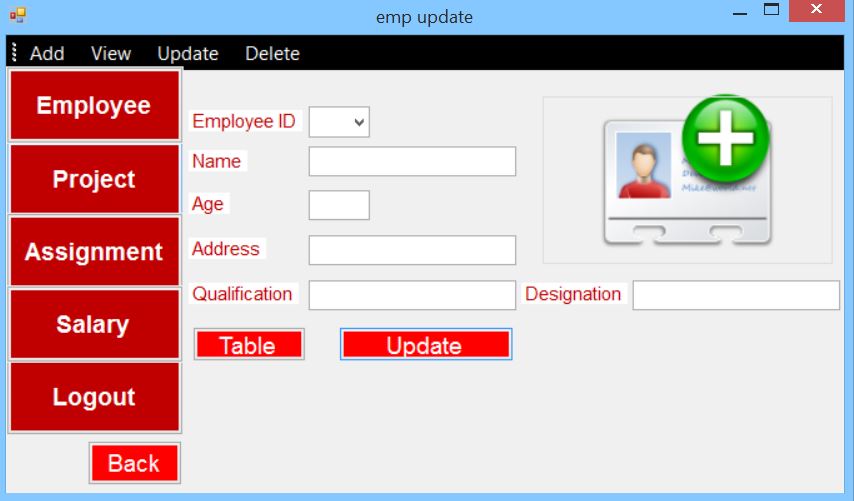
User/administrator who are not registered should go to registration form and register in order to use the system.





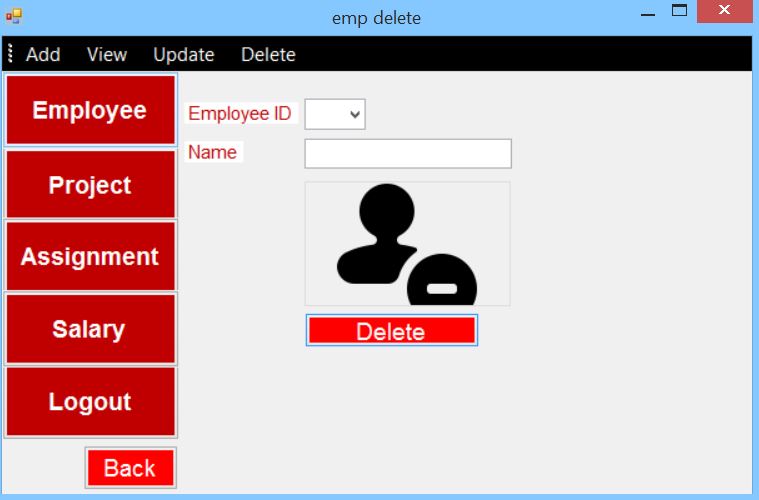
To add employee:

* Fill out the forms accordingly to the information and labels given
* All required details should be entered for it to be added.
* Click Add button.



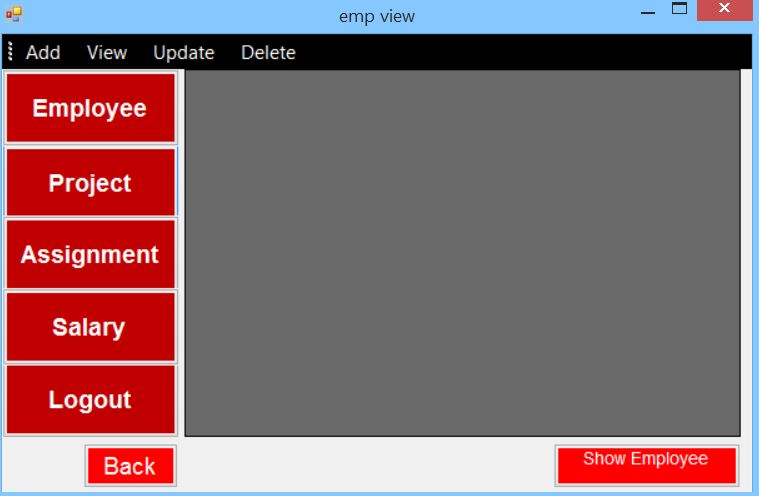
To update employee:

* Select an existing employee ID and fill out the forms accordingly to the information and labels given. You also can open table for reference to add employee details that doesn’t need change.
* All required details should be entered for it to be updated.
* Click Update button.



To delete employee:

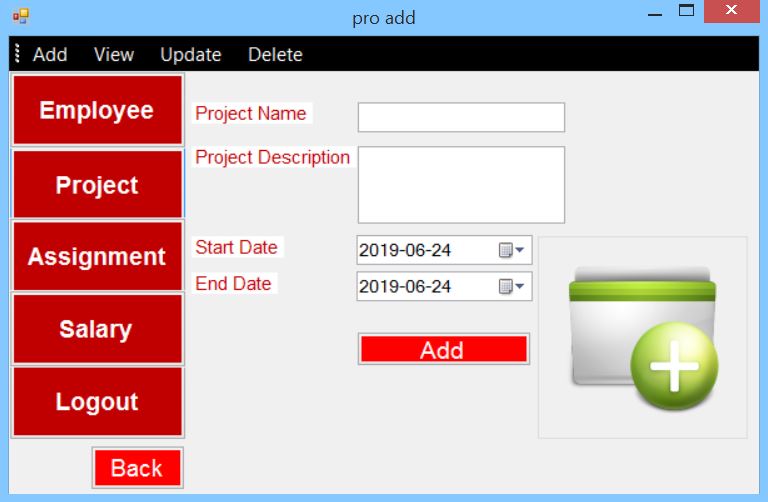
* Select an existing employee ID and the employee name of that employee ID.
* All required details should be entered for it to be deleted.
* Click Delete button.



To view employee:

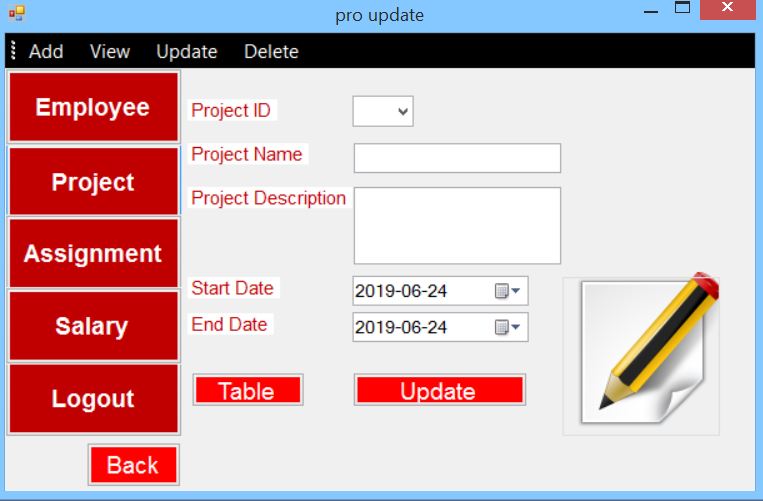
* Click Show Employee button.

All the existing employees’ information and details are displayed.



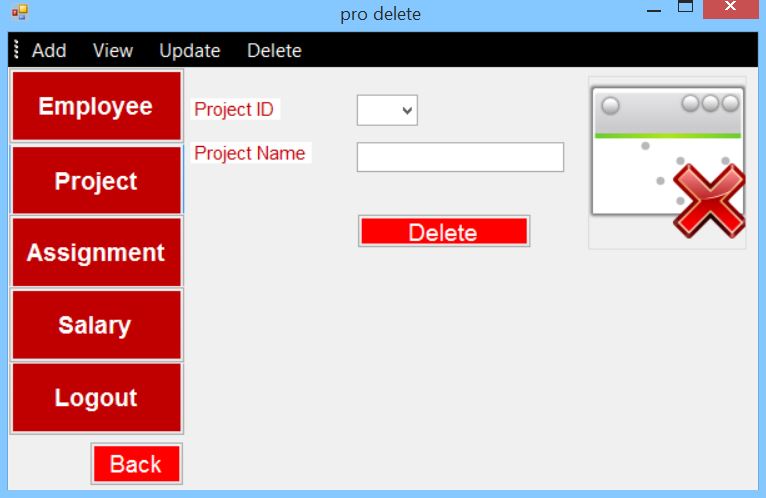
To add project:

* Fill out the forms accordingly to the information and labels given
* All required details should be entered for it to be added.
* Click Add button.



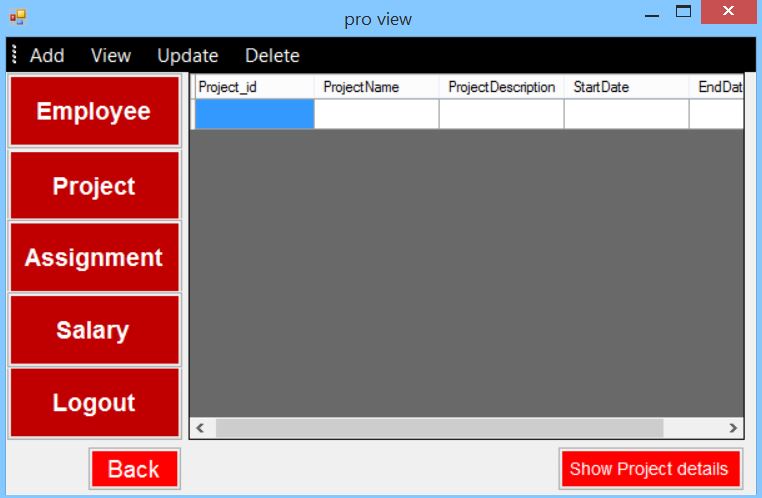
To update project:

* Select an existing project ID and fill out the forms accordingly to the information and labels given. You also can open table for reference to add project details that doesn’t need change.
* All required details should be entered for it to be updated.
* Click Update button.



To delete employee:

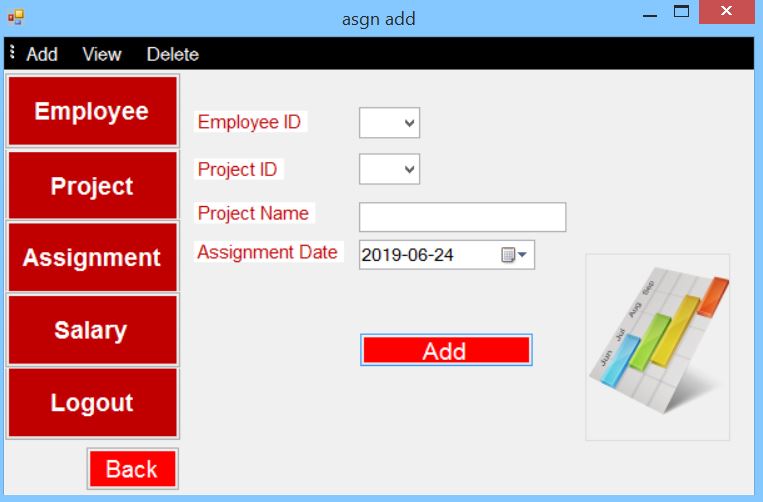
* Select an existing project ID and the project name of that project ID.
* All required details should be entered for it to be deleted.
* Click Delete button.



To view project:

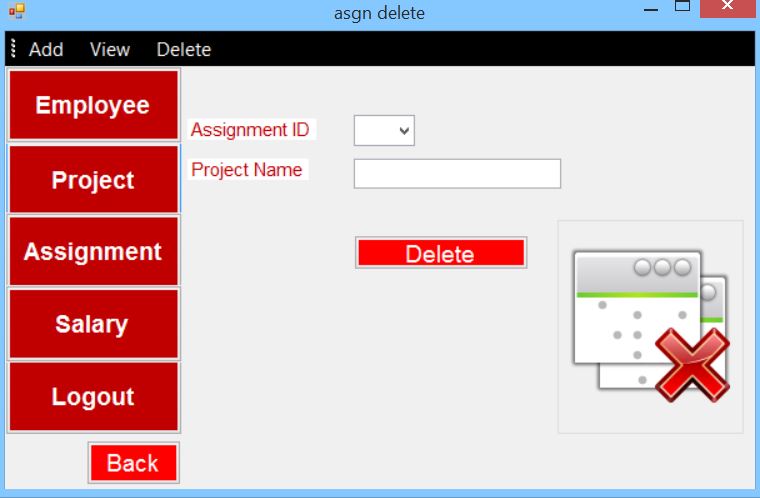
* Click Show Project details button.

All the existing project information and details are displayed.



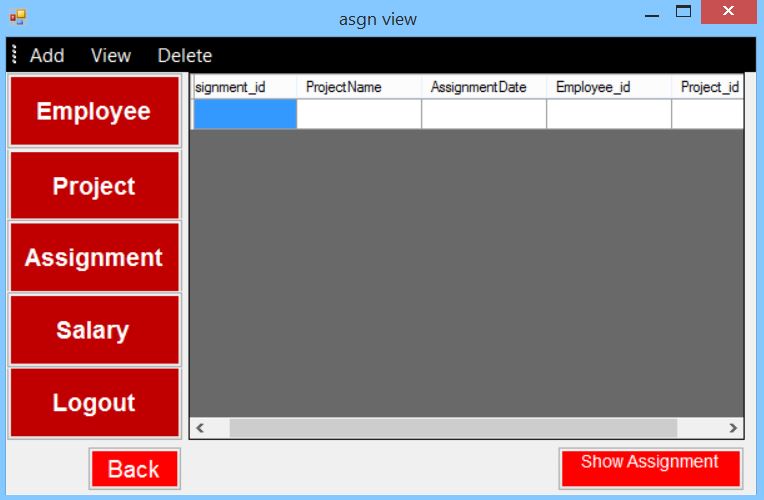
To add assignment:

* Select existing employee ID and project ID and fill out the forms accordingly to the information and labels given
* All required details should be entered for it to be added.
* Click Add button.



To delete assignment:

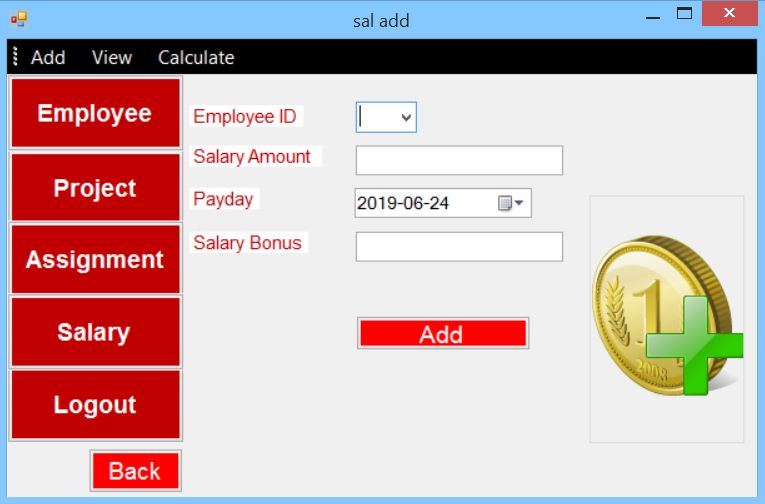
* Select existing assignment ID and the project name of that assignment ID.
* All required details should be entered for it to be deleted.
* Click Delete button.



To view assignment:

* Click Show Assignment button.

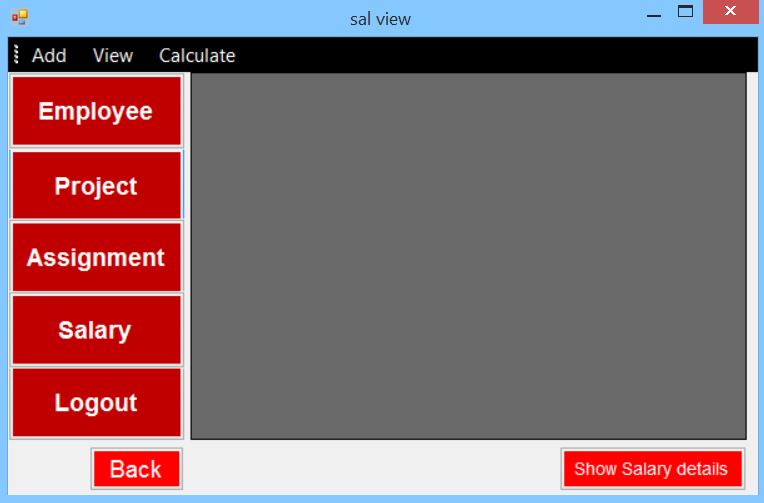
All the existing assignment information and details are displayed.

****

To add salary:

* Select existing employee id and fill out the forms accordingly to the information and labels given
* All required details should be entered for it to be added.
* Click Add button.

****

****

To view employee’s salary:

* Click Show Salary details button.

All the existing employees’ salary details are displayed.

# **Chapter 7 Conclusion**

At last, the project Employee Management System was successfully completed after number of research and countless hours dedicated to its studies. Various issues and problems faced were sorted out in one way or another with the help of friends and teachers during the development of this employee management system.

All the tasks given were done successfully. Analysis, design, implementation and testing were done successfully. At first, the requirements analysis was done and gathered all the requirements for the project. Project’s aim, features and objectives were specified. All of these were again analyzed. MoSCoW prioritization, risk management and configuration management were also performed for the project. After analysis was done, the design specification was don’t for the project. Various diagrams like behavioral diagram, static diagram, Entity Relationship diagram etc. were drawn and designed successfully for the employee management system. All these helped and contributed for the overall design of the project. After the design phase was completed, implementation of code was done using C# .net framework and MS-SQL was used for database. After the completion of implementation, both blackbox testing and unit testing was done to identify and sort errors if there were any. This way testing both of the testing were done. User guide was also created for the employee management system so that the users would not face issue while using it and go through if they faced any problem. This would increase efficiency of the project.

All the necessary steps were done and completed successfully for the development of Employee Management System. Therefore, Employee Management System was successfully developed.

# **References**

electra0084, 2019. *coursehero.* [Online]   
Available at: https://www.coursehero.com/file/pdfhi7/NLA-Process-Natural-Language-Analysis-NLA-is-the-process-of-identifying-verbs/  
[Accessed 3 May 2019].

Inflectra, 2018. *Inflectra.* [Online]   
Available at: https://www.inflectra.com/ideas/topic/requirements-definition.aspx  
[Accessed 22 May 2019].

techopedia, 2019. *techopedia.* [Online]   
Available at: https://www.techopedia.com/definition/1200/entity-relationship-diagram-erd  
[Accessed 14 May 2019].

tryqa, 2019. *tryqa.* [Online]   
Available at: http://tryqa.com/what-is-software-testing/  
[Accessed 2 July 2019].