**A Project report**

**on**

***Employee Leave Management System***

Submitted in partial fulfillment of the requirements of the award of the   
degree of Bachelor of Computer Application



Guide Submitted by

**Dr.Mukesh Joshi ShubhamPandey**

**SurajDwivedi**

Submitted to

Amrapali Institute of Applied Sciences, Haldwani

**ACKNOWLEDGEMENT**

We take the opportunity to express our gratitude and thanks to all the people who directly or indirectly helped and guided us throughout the project. It is our privilege to express our sincerest regards to our mentor or project coordinator “**Dr.MukeshJoshi** ” for their valuable inputs, able guidance, encouragement, whole-hearted co-operation and constructive criticism throughout the duration of our project. We take this opportunity to thank all our lecturers who have directly or indirectly helped our project.

**CERTIFICATE**

This is to certify that this project entitled “**Employee Leave Management System**” submitted in partial fulfillment of the degree of Bachelor of Computer Applications to the “ **Dr.Mukesh Joshi** ”, Amrapali Group of Institute, done by “ **ShubhamPandey&SurajDwivedi** ” is an authentic work carried out by them at “Amrapali Group of Institute” under my guidance. The matter embodied in this project work has not been submitted earlier for award of any degree or diploma to the best of my knowledge and belief.

**DECLARATION**

We hereby declare that the project work entitled “**Employee Leave Management System**” is being submitted to Faculty of Computer Science and Application is the authentic record of our own project work done under the guidance of **Dr.Mukesh** Joshi FCSA, Amrapali Group of Institutes, Haldwani .

Team: Signature of Team:

ShubhamPandey

SurajDwivedi

**CERTIFICATE**

This is to certify that the project titled “***employee leave management sysytem***” is the bonafide work carried out by “ShubhamPandey&SurajDwivedi ” student of Bachelor of Computer Applications, Amrapali Institute of Applied Sciences, Haldwani affiliated to Kumaun University, Nainital, India, during the academic year 2018-2019, and that the project has not formed on the basis for the award previously of any other degree, diploma, fellowship or any other similar title.

Signature of the Guide:

**CONTENTS**

**INTRODUCTION**

INTRODUCTION TO PROJECT

PURPOSE OF THE PROJECT

SCOPE OF THE PROJECT

OBJECTIVE OF THE PROJECT

DEFINITION, ACRONYMS AND ABBREVIATIONS

**SYSTEM ANALYSIS**

 INTRODUCTION

SYSTEM WORKFLOW

STUDY OF THE SYSTEM

HARDWARE & SOFTWARE REQUIRMENT

PROPOSED SYSTEM

INPUT & OUTPUT

PROCESS MODELS USED WITH JUSTIFICATION

**FEASIBILITY REPORT**

TECHNICAL FEASIBILITY

OPERATIONAL FEASIBILITY

ECONOMIC FEASIBILITY

**SOFTWARE REQUIREMENT SPECIFICATIONS**

FUNCIONAL REQUIREMENTS

PERFORMANCE REQUIREMENTS

**SELECTED SOFTWARE**

INTRODUCTION TO .NET FRAMEWORK

C#.NET

OLedb SERVER

**SYSTEM DESIGN**

ANALYSIS MODEL

NORMALIZATION

E-R DIAGRAM

* LOGICAL E-R DIAGRAM
* CONCEPTUAL E-R DIAGRAM
* PHYSICAL E-R DIAGRAM
* DATABASE RELATIONSHIP CHART

DATA FLOW DIAGRAMS

* 0 Level DFD
* 1 Level DFD

DATA DICTIONARY

**SYSTEM SECURITY**

**CONCLUSION**

**FUTURE SCOPE OF THE PROJECT**

**OUTPUT & INPUT SCREENS**

**INTRODUCTION**

In the existing paper work related to leave management, leaves are maintained using the attendance register for staff. The staff needs to submit their leaves manually to their respective authorities. This increases the paperwork and maintaining the records becomes tedious. Maintaining notices in the records also increases the paper work.

The main objective of the purposed system is to decrease the paper work and help in easier records maintenance by having a particular centralized database system, where leaves and notices are maintained. The purposed system automates the existing system. It decreases the paper work and enables easier record, maintenance. It also reduce changes of data lose. This module intelligently adapts to HR policy of the management and allow the employees and their line managers to manage leaves and replacements for better scheduling of workload.

#### PURPOSE OF THE PROJECT

This project is aimed at developing a Employee Leave Management System, which is of importance to either an organization.. This system can be used to automate the workflow of leave applications and their approvals.

There are features like cancellation of leave, approval of leave, report generators etc in this Tool.

**SCOPE OF THE PROJECT**

The Leave Management software aims at helping the user to address issues from multi-disciplinary angles related to Leave management andservices.

The major benefits ofthis software are –

* 1 .It is a unique software which helps to organize event without any paperwork.
* 2. It has a wide variety of Modules.
* 3. By just few clicks user can check the leave status, leave balance, notices and apply for and grantleave accordingly.

**OBJECTIVE**

* To automate the existing leave management in Organization.
* To decrease the paperwork and enable the process with efficient, reliable record maintenance by using centralized database, thereby reducing chances of data loss.
* To provide for an automated leave management system that intelligently adapts to HR policy for the organization and allows employees and their line managers to manage leave and replacements for better scheduling of work load & processes

**DEFINITION, ACRONYMS AND ABBREVIATIONS**

**ADMIN**: Administrator.

**RAM**: Random access memory.

**ROM**: Read only memory.

**GUI**: Graphics user interface.

**GB**: Giga Byte.

**SRS**: Software Requirement and Specification.

**DFD**: Data Flow Diagram.

**E-RD**: Entity Relationship Diagram.

**OLE DB**:Object Linking and Embedding, Database

**OS**: Operating System.

**HDD:** Hard Disk Drive

**PERT:**Program Evaluation and Review Technique

SYSTEM DESIGN

**ANALYSIS MODEL**

The model that is basically being followed is the WATER FALL MODEL, which states that the phases are organized in a linear order. First of all the feasibility study is done. Once that part is over the requirement analysis and project planning begins. If system exists one and modification and addition of new module is needed, analysis of present system can be used as basic model.

The design starts after the requirement analysis is complete and the coding begins after the design is complete. Once the programming is completed, the testing is done. In this model the sequence of activities performed in a software development project are: -

* Requirement Analysis
* Project Planning
* System design
* Detail design
* Coding
* Unit testing
* System integration & testing

Here the linear ordering of these activities is critical. End of the phase and the output of one phase is the input of other phase. The output of each phase is to be consistent with the overall requirement of the system. Some of the qualities of spiral model are also incorporated like after the people concerned with the project review completion of each of the phase the work done.

### STUDY OF THE SYSTEM

# GUI’S

In the flexibility of the uses the interface has been developed a graphics concept in mind, associated through a interface. The GUI’S at the top level have been categorized as

1. Administrative user interface
2. The operational or generic user interface

The administrative user interface concentrates on the consistent information that is practically, part of the organizational activities and which needs proper authentication for the data collection. The interfaces help the administrations with all the transactional states like Data insertion, Data deletion along with the extensive data search capabilities.

The operational or generic user interface helps the users upon the system in transactions through the existing data and required services. The operational user interface also helps the ordinary users in managing their own information helps the ordinary users in managing their own information in a customized manner as per the assisted flexibilities.

#### HARDWARE& SOFTWARE SPECIFICATIONS

**HARDWARE REQUIREMENTS:**

* 2.8 GHz Processor and Above
* RAM 2 GB and Above
* HDD 20 GB Hard Disk Space and Above

**SOFTWARE REQUIREMENTS:**

* WINDOWS OS (XP / window 7 / window 8 / window 10)
* Visual Studio .Net 2007 Enterprise Edition
* Visual Studio .Net Framework (Minimal for Deployment)
* OLE DB server

**PROPOSED SYSTEM**

To debug the existing system, remove procedures those cause data redundancy, make navigational sequence proper. To provide information about audits on different level and also to reflect the current work status depending on organization/auditor or date. To build strong password mechanism.

**NEED FOR COMPUTERIZATION**

We all know the importance of computerization. The world is moving ahead at lightening speed and every one is running short of time. One always wants to get the information and perform a task he/she/they desire(s) within a short period of time and too with amount of efficiency and accuracy. The application areas for the computerization have been selected on the basis of following factors:

* Minimizing the manual records kept at different locations.
* There will be more data integrity.
* Facilitating desired information display, very quickly, by retrieving inform from users.
* Facilitating various statistical information which helps in decision-making?
* To reduce manual efforts in activities that involved repetitive work.
* Updating and deletion of such a huge amount of data will become easier.

**INPUT AND OUTPUT**

The main inputs, outputs and major functions of the system are as follows

**INPUTS:**

* Admin enters his or her user id and password.
* Employee enter his or her user id and password.
* Employee send request for Leave.
* Employee can check for status for Leave.
* Admin can edit the employee details and so on.

**OUTPUTS:**

* Admin gets his homepage.
* Employee get his homepage.
* Employee leave request data will be stored in database..
* Displays leave Status.
* Admin view employee details.

**PROCESS MODELS USED WITH JUSTIFICATION**

The following commands specify access control identifiers and they are typically used to authorize and authenticate the user (command codes are shown in parentheses).

**USER NAME (USER)**

The user identification is that which is required by the server for access to its file system. This command will normally be the first command transmitted by the user after the control connections are made (some servers may require this).

**PASSWORD (PASS)**

This command must be immediately preceded by the user name command, and, for completes the user's identification for access control. Since password information is quite sensitive, it is desirable in general to "mask" it or suppress type out.

### FEASIBILITY REPORT

 Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:

* Technical Feasibility
* Operation Feasibility
* Economical Feasibility

**Technical Feasibility**

* + The technical issue usually raised during the feasibility stage of the investigation includes the following:
* Does the necessary technology exist to do what is suggested?
* Do the proposed equipments have the technical capacity to hold the data required to use the new system?
* Will the proposed system provide adequate response to inquiries, regardless of the number or location of users?
* Can the system be upgraded if developed?
* Are there technical guarantees of accuracy, reliability, ease of access and data security?

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface. Thus it provides an easy access to the users.

The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified.

**Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. Some of the important issues raised are to test the operational feasibility of a project includes the following: -

* Is there sufficient support for the management from the users?
* Will the system be used and work properly if it is being developed and implemented?
* Will there be any resistance from the user that will undermine the possible application benefits?

This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits.

**Economic Feasibility**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs.

The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies. There is nominal expenditure and economical feasibility for certain.

**SOFTWARE REQUIREMENT SPECIFICATION**

**INTRODUCTION**

**Purpose:** The main purpose for preparing this document is to give a general insight into the analysis and requirements of the existing system or situation and for determining the operating characteristics of the system.

**Scope:** This Document plays a vital role in the development life cycle (SDLC) and it describes the complete requirement of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

**DEVELOPERS RESPONSIBILITIES OVERVIEW:**

The developer is responsible for:

* Developing the system, which meets the SRS and solving all the requirements of the system?
* Demonstrating the system and installing the system at client's location after the acceptance testing is successful.
* Submitting the required user manual describing the system interfaces to work on it and also the documents of the system.
* Conducting any user training that might be needed for using the system.
* Maintaining the system for a period of one year after installation.

### FUNCTIONAL REQUIREMENTS

In software engineering, a functional requirement defines a system or its component. It describes the functions a software must perform. A function is nothing but inputs, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform.

Functional software requirements help you to capture the intended behavior of the system. This behaviour may be expressed as functions, services or tasks or which system is required to perform

## Advantages of Functional Requirement

Here, are the pros/advantages of creating a typical functional requirement document-

* Helps you to check whether the application is providing all the functionalities that were mentioned in the functional requirement of that application
* A functional requirement document helps you to define the functionality of a system or one of its subsystems.
* Functional requirements along with requirement analysis help identify missing requirements. They help clearly define the expected system service and behavior.
* Errors caught in the Functional requirement gathering stage are the cheapest to fix.
* Support user goals, tasks, or activities for easy project management.

**NON – FUNCTIONAL REQUIREMENT**

A non-functional requirement defines the quality attribute of a software system. They represent a set of standards used to judge the specific operation of a system. Example, how fast does the website load?

A non-functional requirement is essential to ensure the usability and effectiveness of the entire software system. Failing to meet non-functional requirements can result in systems that fail to satisfy user needs.

Non-functional Requirements allows you to impose constraints or restrictions on the design of the system across the various agile backlogs. Example, the site should load in 3 seconds when the number of simultaneous users are> 10000. Description of non-functional requirements is just as critical as a functional requirement

## Advantages of Non-Functional Requirement

Benefits/pros of Non-functional testing are:

* The nonfunctional requirements ensure the software system follow legal and compliance rules.
* They ensure the reliability, availability, and performance of the software system
* They ensure good user experience and ease of operating the software.
* They help in formulating security policy of the software system.

### Summary:

* A functional requirement defines a system or its component
* A non-functional requirement defines the performance attribute of a software system.
* Functional requirements along with requirement analysis help identify missing requirements
* The advantage of Non-functional requirement is that it helps you to ensure good user experience and ease of operating the software
* Transaction corrections, adjustments, and cancellations, Business Rules, Certification Requirements, Reporting Requirements, Administrative functions, Authorization levels, Audit Tracking, External Interfaces, Historical Data management, Legal or Regulatory Requirements are various types of functional requirements
* Types of Non-functional requirement are Scalability Capacity, Availability, Reliability, Recoverability, Data Integrity, etc.
* Functional Requirement is a verb while Non-Functional Requirement is an attribute

**SELECTED SOFTWARE**

# INTRODUCTION TO .NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and Remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

**C #.NET**

C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and approved by European Computer Manufacturers Association (ECMA) and International Standards Organization (ISO). C# was developed by Anders Hejlsberg and his team during the development of .Net Framework. C# 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 following reasons make C# a widely used professional language:

• It is a modern, general-purpose programming language

• It is object oriented.

• It is component oriented.

• It is easy to learn.

• It is a structured language.

• It produces efficient programs.

• It can be compiled on a variety of computer platforms.

• It is a part of .Net Framework.

# C# OLEDB CONNECTION

The C# **OleDbConnection** instance takes Connection String as argument and pass the value to the Constructor statement. An instance of the C# OleDbConnection class is supported the **OLEDB Data Provider** .

**connetionString = "Provider=Microsoft.Jet.OLEDB.4.0;**

**Data Source=yourdatabasename.mdb;";**

**cnn = new OleDbConnection(connetionString**);

When the connection is established between C# application and the specified Data Source, SQL Commands will execute with the help of the **Connection Object** and retrieve or manipulate data in the database. Once the Database activities is over Connection should be closed and release from the data source resources .

**cnn.Close();**

The Close() method in the OleDbConnection class is used to close the Database Connection. The Close method **Rolls Back** any pending transactions and releases the Connection from the Database connected by the OLEDB Data Provider.

**SOFTWARE DESIGN**

**INTRODUCTION**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage.

During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

**NORMALIZATION**

It is a process of converting a relation to a standard form.  The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity.  To do this we use normal forms or rules for structuring relation.

**Insertion anomaly**: Inability to add data to the database due to absence of other data.

**Deletion anomaly**: Unintended loss of data due to deletion of other data.

**Update anomaly**: Data inconsistency resulting from data redundancy and partial update

**Normal Forms**:  These are the rules for structuring relations that eliminate anomalies.

**FIRST NORMAL FORM**:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation.  By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

**SECOND NORMAL FORM**:

          A relation is said to be in second Normal form is it is in first normal form and it should satisfy any one of the following rules.

1)   Primary key is a not a composite primary key

2)   No non key attributes are present

3)   Every non key attribute is fully functionally dependent on full set of primary key.

**THIRD NORMAL FORM**:

A relation is said to be in third normal form if their exits no transitive dependencies.

**Transitive Dependency**:  If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

          The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

**DATA FLOW DIAGRAM**

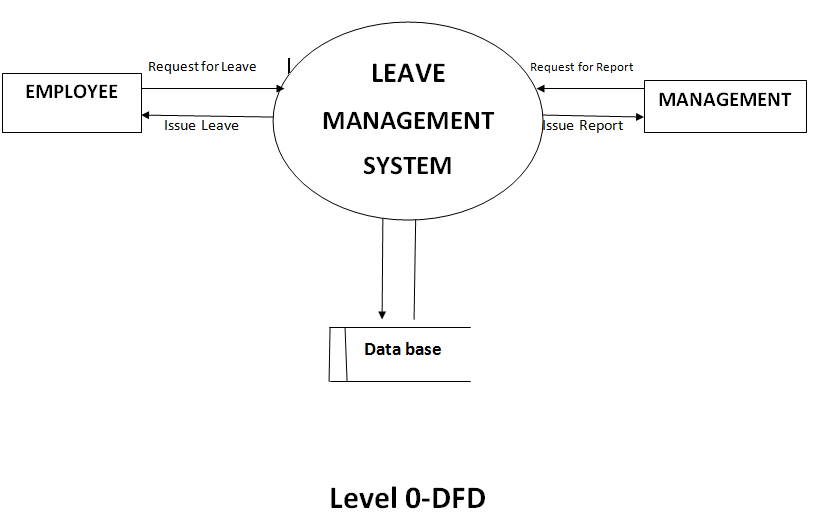
A Data Flow Diagram(DFD)is a diagram that describes the flow of data and the

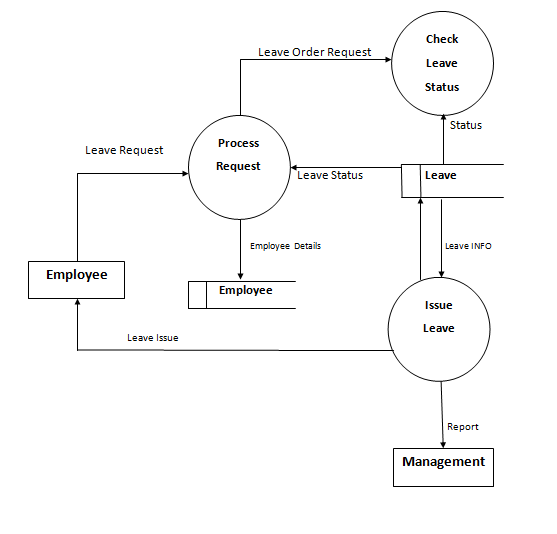
processes that change or transform data throughout a system. It’s a structured analysis and design tool that can be used for flowcharting in place of, or in association with, information oriented and process oriented system flowcharts.

When analysts prepare the Data Flow Diagram, they specify the user needs at a level of detail that virtually determines the information flow into and out of the system and the required data resources. This network is constructed by using a set of symbols that do not imply a physical implementation. The Data Flow Diagram reviews the current physical system, prepares input and output specification, specifies the implementation plan etc.

Four basic symbols are used to construct data flow diagrams. They are symbols that represent data source, data flows, and data transformations and data storage. The points at which data are transformed are represented by enclosed figures, usually circles, which are called nodes.

Data flow diagrams can be used to provide the end user with a physical idea of where the data they input ultimately has an effect upon the structure of the whole system from order to dispatch to report. How any system is developed can be determined through a data flow diagram model. In the course of developing a set of leveled data flow diagrams the analyst/designer is forced to address how the system may be decomposed into component sub-systems, and to identify the transaction data in the data model. Data flow diagrams can be used in both Analysis and Design phase of the SDLC. There are different notations to draw data flow diagrams (Yourdon&Coad and Gane&Sarsen , defining different visual representations for processes, data stores, data flow, and external entities).





***Level 1-DFD***

**ENTITY RELATIONSHIP DIAGRAM**

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database. An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

## Common Entity Relationship Diagram Symbols

An ER diagram is a means of visualizing how the information a system produces is related. There are five main components of an ERD:

* **Entities**, which are represented by rectangles. An entity is an object or concept about

which you want to store information.

.

* A weak entity is an entity that must defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone.



* **Actions**, which are represented by diamond shapes, show how two entities share information in the database



* .In some cases, entities can be self-linked. For example, employees can supervise other employees.



* **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.



* A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values

.

* A derived attribute is based on another attribute. For example, an employee's monthly salary is based on the employee's annual salary

.

* **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.
* **Cardinality** specifies how many instances of an entity relate to one instance of another entity. Ordinality is also closely linked to cardinality. While cardinality specifies the occurrences of a relationship, ordinarily describes the relationship as either mandatory or optional. In other words, cardinality specifies the maximum number of relationships and ordinality specifies the absolute minimum number of relationships.

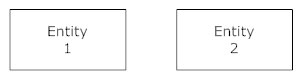
There are many notation styles that express cardinality.

**INFORMATION ENGINEERING STYLE  
**

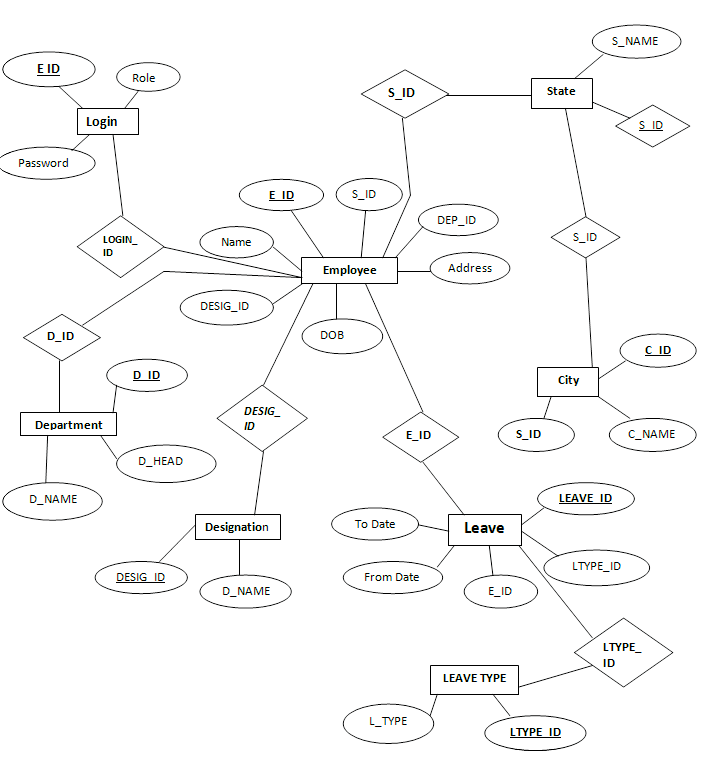
## ER Diagram Uses:-

When documenting a system or process, looking at the system in multiple ways increases the understanding of that system. ERD diagrams are commonly used in conjunction with a data flow diagram to display the contents of a data store. They help us to visualize how data is connected in a general way, and are particularly useful for constructing a relational database.

**Here are some best practice tips for constructing an ERD:**

* Identify the entities. The first step in making an ERD is to identify all of the entities you will use. An entity is nothing more than a rectangle with a description of something that your system stores information about. This could be a customer, a manager, an invoice, a schedule, etc. Draw a rectangle for each entity you can think of on your page. Keep them spaced out a bit.  
  
* Identify relationships. Look at two entities, are they related? If so draw a solid line connecting the two entities.
* Describe the relationship. How are the entities related? Draw an action diamond between the two entities on the line you just added. In the diamond write a brief description of how they are related.
* Add attributes. Any key attributes of entities should be added using oval-shaped symbols.
* Complete the diagram. Continue to connect the entities with lines, and adding diamonds to describe each relationship until all relationships have been described. Each of your entities may not have any relationships, some may have multiple relationships. That is okay.

**LOGICALENTITY RELATIONSHIPDIAGRAM**



**CONCEPTUAL ENTITY RELATIONSHIP DIAGRAM**

***Will***

***Register***

**New Employee**

***Will***

***If\_admin***

***If\_normal***

***Login***

***Can***

**Will Taken**

**Taken**

**Can**

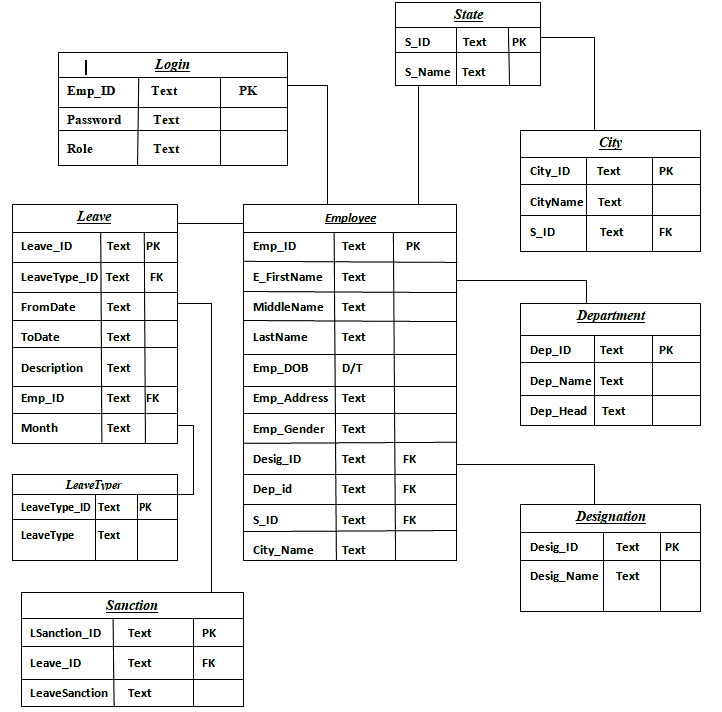
***Delete Employee***

***LeaveTaken***

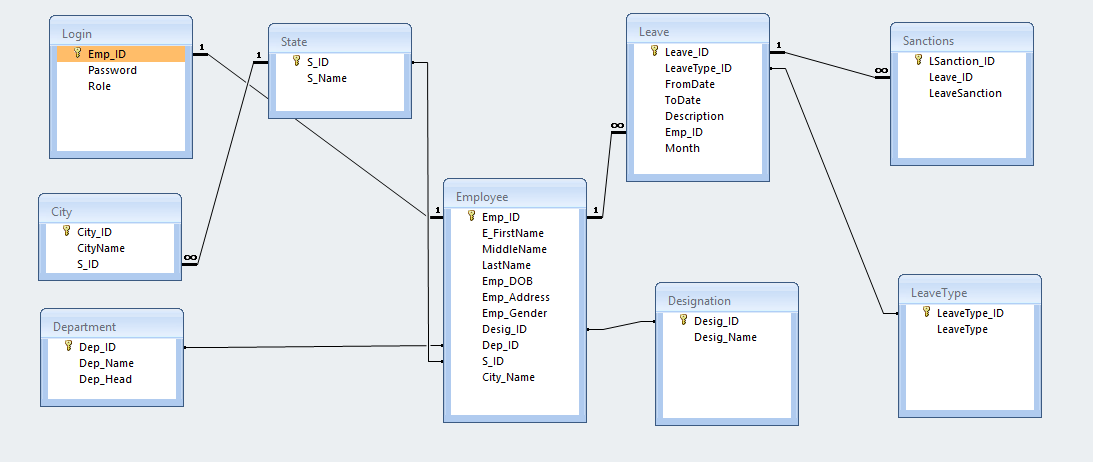
***Sanction Leave***

***Employee***

**PHYSICAL ENTITY RELATIONSHIP DIAGRAM**

****

**DATABASE RELATIONSHIP CHART**

****

**DATA DICTIONARY**

USER INFORMATION

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | Emp\_id | text | 6 |
| 2 | password | text | 20 |
| 3 | Role | text | 20 |

DEPARTMENT

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | Dep\_id | text | 6 |
| 2 | Dep\_Name | text | 20 |
| 3 | Dep\_Head | text | 20 |

STATE

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | S\_ID | text | 6 |
| 2 | S\_Name | text | 20 |

CITY

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | City\_ID | text | 6 |
| 2 | City\_Name | text | 20 |
| 3 | S\_ID | text | 6 |

LEAVE

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | Leave\_ID | text | 6 |
| 2 | LeaveType\_ID | text | 6 |
| 3 | From date | text | 6 |
| 4 | To date | text | 6 |
| 5 | Description | text | 30 |
| 6 | Emp\_ID | text | 6 |
| 7 | Month | text | 2 |

EMPLOYEE REGISTER

|  |  |  |  |
| --- | --- | --- | --- |
| ***#*** | ***Field name*** | ***Data type*** | ***Field size*** |
| 1 | Emp\_ID | text | 6 |
| 2 | Emp\_FirstName | text | 10 |
| 3 | MiddleName | text | 10 |
| 4 | LastName | text | 10 |
| 5 | Emp\_DOB | Date/time | 10 |
| 6 | Emp\_Address | text | 30 |
| 7 | Emp\_Gender | text | 10 |
| 8 | Desig\_ID | text | 20 |
| 9 | Dept\_id | text | 20 |
| 10 | S\_ID | text | 20 |
| 11 | City\_Name | text | 20 |

SYSTEM SECURITY

# INTRODUCTION

The protection of computer based resources that includes hardware, software, data, procedures and people against unauthorized use or natural

Disaster is known as System Security.

System Security can be divided into four related issues:

* Security
* Integrity
* Privacy
* Confidentiality

**SYSTEM SECURITY** refers to the technical innovations and procedures applied to the hardware and operation systems to protect against deliberate or accidental damage from a defined threat.

**DATA SECURITY** is the protection of data from loss, disclosure, modification and destruction.

**SYSTEM INTEGRITY** refers to the power functioning of hardware and programs, appropriate physical security and safety against external threats such as eavesdropping and wiretapping.

**PRIVACY** defines the rights of the user or organizations to determine what information they are willing to share with or accept from others and how the organization can be protected against unwelcome, unfair or excessive dissemination of information about it.

**CONFIDENTIALITY** is a special status given to sensitive information in a database to minimize the possible invasion of privacy. It is an attribute of information that characterizes its need for protection.

### CONCLUSION

 It has been a great pleasure for me to work on this exciting and challenging project. This project proved good for me as it provided practical knowledge of not only programming in C#.NET form-based application and no some extent Windows Application and OleDb Server, but also about all handling procedure related with **“EMPLOYEE LEAVE MANAGEMENT”.**It also provides knowledge about the latest technology used in developing form-based application that will be great demand in future. This will provide better opportunities and guidance in future in developing projects independently.

**BENEFITS:**

The project is identified by the merits of the system offered to the user. The merits of this project are as follows: -

* + This project offers user to enter the data through simple and interactive forms. This is very helpful for the client to enter the desired information through so much simplicity.
  + The user is mainly more concerned about the validity of the data, whatever he is entering. There are checks on every stages of any new creation, data entry or updation so that the user cannot enter the invalid data, which can create problems at later date.
* Sometimes the user finds in the later stages of using project that he needs to update some of the information that he entered earlier. There are options for him by which he can update the records.
* Moreover there is restriction for his that he cannot change the primary data field. This keeps the validity of the data to longer extent.
  + User is provided the option of monitoring the records he entered earlier. He can see the desired records with the variety of options provided by him.
  + From every part of the project the user is provided with the links through framing so that he can go from one option of the project to other as per the requirement. This is bound to be simple and very friendly as per the user is concerned. That is, we can sat that the project is user friendly which is one of the primary concerns of any good project.
  + Data storage and retrieval will become faster and easier to maintain because data is stored in a systematic manner and in a single database.
  + Decision making process would be greatly enhanced because of faster processing of information since data collection from information available on computer takes much less time then manual system.
  + Allocating of sample results becomes much faster because at a time the user can see the records of last years.
  + Easier and faster data transfer through latest technology associated with the computer and communication.  Through these features it will increase the efficiency, accuracy and transparency,

**LIMITATIONS:**

·         The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.

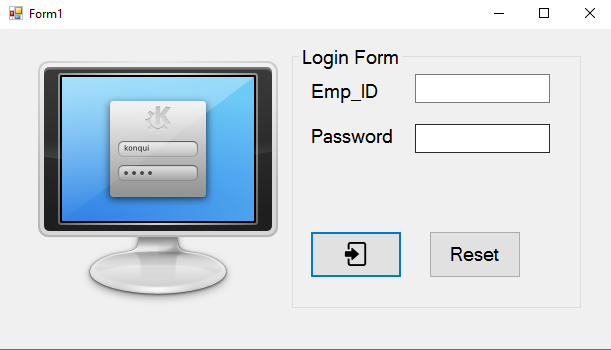
·         Training for simple computer operations is necessary for the   users working on the system.

FUTURE SCOPE OF THE PROJECT

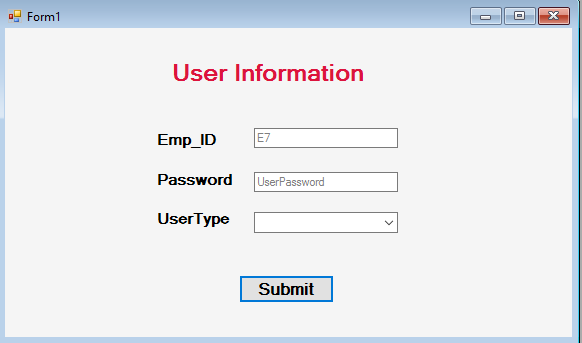
* Moreover, it is just a beginning; further the system may be utilized in various other types of auditing operation viz. Network auditing or similar process/workflow based applications..
* In future my project look useful as today. If user necessities increase with a little effort it can be updated.
* In future my project look useful as today. If user necessities increase with a little effort it can be updated.

OUTPUT AND INPUT SCREEN

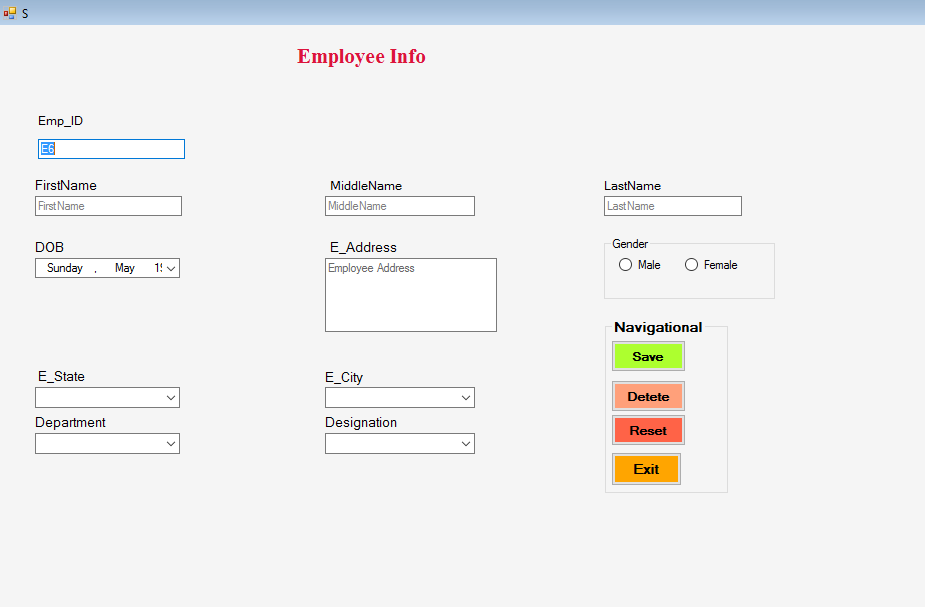
LOGIN



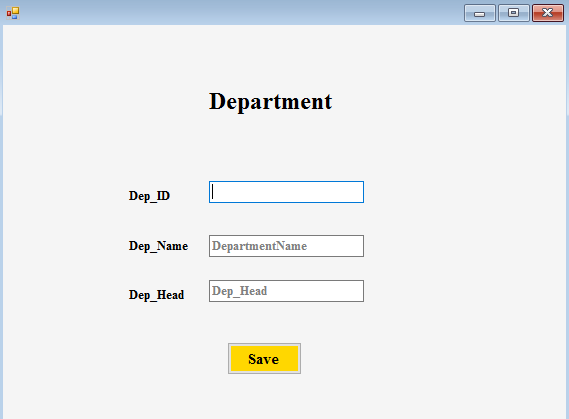
USER INFORMATION



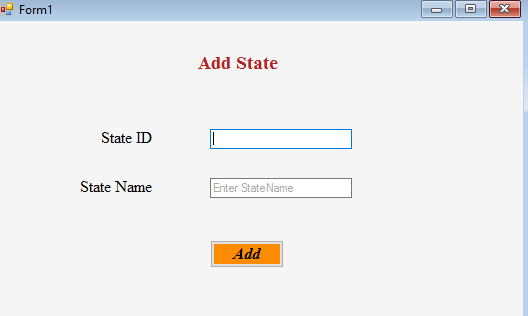
EMPLOYEE INFORMATION



DEPARTMENT



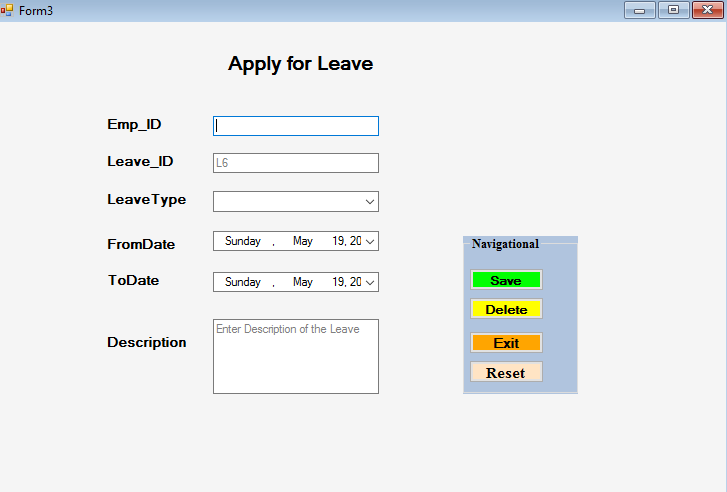
STATE



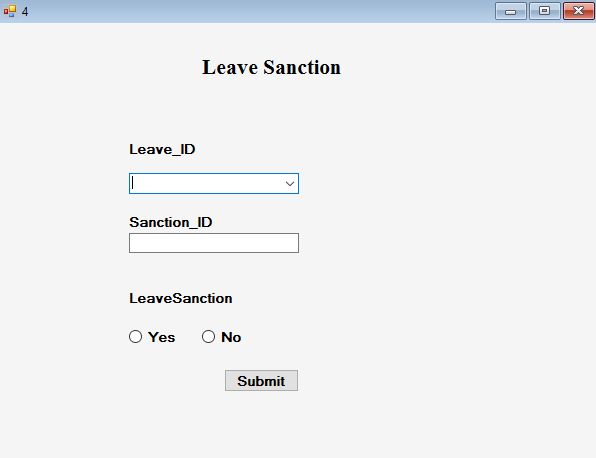
CITY



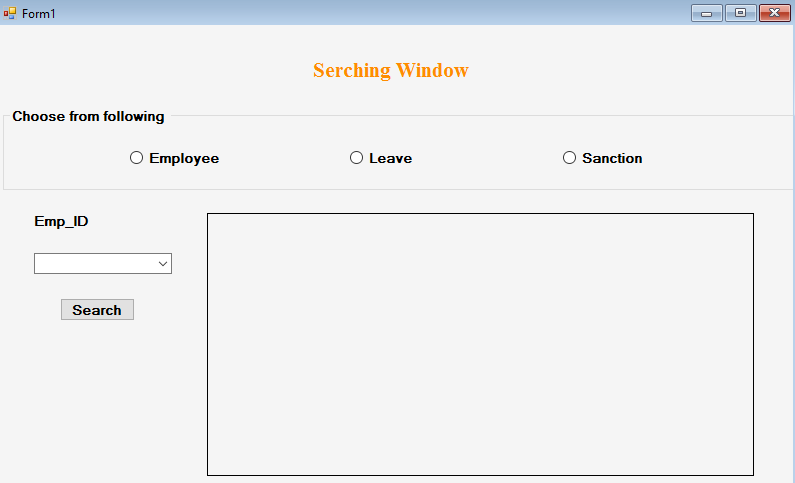
APPLY FOR LEAVE



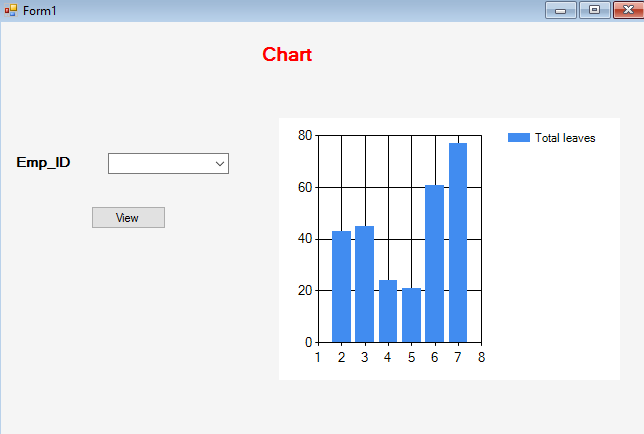
LEAVE SANCTION



SEARCHING WINDOW



CHART



TEST CASES

* 1. LOGIN

TEST CASE ID: T1

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the Employee Leave Management System will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system .  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of Employee Leave Management System  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of Employee Leave Management System 2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. Sign up

TEST CASE ID: T2

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter in dashboard. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the Employee Leave Management System will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get enter to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of Employee Leave Management System 2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of Employee Leave Management System 2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

* 1. RESET PASSWORD

TEST CASE ID: T3

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to Change user id & password | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the Employee Leave Management System will be entered into the system. |
| INPUTS: Username, Password | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of Employee Leave Management System  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of Employee Leave Management System 2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

4 .LEAVE SANCTION

TEST CASE ID: T4

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id, password & other inputs. | EXECUTION HISTORY: |
| PRECONITION: User must visit before this use case begin | RESULT: After log in the Employee Leave Management System will be entered into the system. |
| INPUTS: Leave\_id Sanction\_id | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of Employee Leave Management System  2.Enter User Id  3.Enter Password  4.      Click Submit | [email=790.bhattsapna@gmail.com](mailto:email=790.bhattsapna@gmail.com)  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of Employee Leave Management System 2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |

5 SEARCHING WINDOW

TEST CASE ID: T5

|  |  |
| --- | --- |
| SECTION-1  (BEFORE EXECUTION) | SECTION-2  (AFTER EXECUTION) |
| PURPOSE: Use to provide way to enter through user id & password to give assignment to the students. | EXECUTION HISTORY: |
| PRECONITION: User must visit the before this use case begin | RESULT: After log in the Employee Leave Management System will be entered into the system. |
| INPUTS: Check Employee or Check Leave or Check sanction And Emp\_ID . | IF FAILS, ANY POSSIBLE REASON: Either with invalid password or username. |
| EXPECTED OUTPUTS: User get login to the system with valid id and password. | ANY OTHER OBSERVATION: No |
| POST CONDITONS: If the use case was successful, the actor is logged into the system.  If not the system state unchanged. | ANY SUGGESTION: By entering valid username and password you can log in to the system. |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TU01 | Check User Login with valid Data | 1. Go to application of Employee Leave Management System .  2.Enter User Id  3.Enter Password  4.      Click Submit | email=790.bhattsapna@gmail.com  Password = 1234 | User should Login into application | As Expected | Pass |
| TU02 | Check User Login with invalid Data | 1. Go to application of Employee Leave Management System .  2. Enter User Id  3.Enter Password  4.Click Submit | email = [riya@gmail.com](mailto:riya@gmail.com)  Password = 12345 | User should not Login into application | As Expected | Pass |