# **Chapter 1: Introduction**

Each and every organization either large or small, uses a variety of capital that includes cash, valuables, or best employee to make business work efficiently. Most importantly, employees must be concerned in the company for better welfare by increasing their commitments and level of job satisfaction. But it has been a tough task to handle manual entry of the employee’s performances and payrolls activities that required a lot of labor and space. Besides this, it also decreases the productivity by reducing focus on core areas in business that must be eradicated.

**Employee Payroll System** has been argued to be a solution for above issues in organizations. **EPS** is an operation, designed to maximize the employee’s performance in order to meet their strategic goals and objectives. It focuses on management of employees within companies and emphasizing on policies. Or simply, it is a process of selecting, recruiting employees providing proper orientation and developing skills.

It includes employee’s assessment like performance appraisal, encouragement, maintaining proper relationship with employees, facilitating compensation and benefits and taking care of employee welfare and safety. Finally, it is an entire unified platform of solutions and services to help with all human capital management needs.

## **Background of the System**

In terms of background, the Employee Payroll System was performed manually that consumed lots of time and space that resulted in dissatisfaction of employee and harmed company through various factors. So, the computerized employee payroll system has been developed which is also called as personnel management system that consists of overall activities to ensure the effectiveness utilization of employees toward company.

A company EPS functions focuses on the employee side of management that helps the company to deal effectively with its employee during various phases of employment cycle, including staffing, projects, task-assigned, disciplinary actions and many more. It includes all the activities used to retain and attract employees and ensure their performance meeting company goals. The system efficiency can be increased by including automatic calculations that helps in reducing the manual error with lower cost and effort.

As a result, the web-based employee payroll system is developed.

## **Justification of the System**

In the beginning of employee payroll system, company changes the issues that have been recognized as critical to the success of implementation. The proposed system will be served as a purpose of problem-solving processes that are found in assessing, storing, retrieving, updating and calculating data.

It refers to the anticipated gains to overall functionalities of the company that improves the efficiency of employee’s effort. On the basis of business needs, providing quick allocation of data, reducing redundancy of information and ease the effort needed to calculate net salary is performed. Similarly, on intangible basis the improvement of employee satisfaction and confidence through better understanding of their needs results on transaction improvement and data capturing.

## **Overview of the System**

Employee payroll system is a web-based application software with a primary motive to automate and ease the manual old practice of payroll system. The system encompasses every employee that receives a regular wage or other compensation with project or task assigned including their attendance lists with timings. Different payment methods are calculated and an appropriate paycheck are issued.

Firstly, employees are the key factor of production where productivity is a key to measure company’s potential growth and employee’s quality are responsible for improving productivity. Since, employee is the uncontrollable and unpredictable variable of all companies, its success depends upon their proper management. As the EPS can be handled by multiuser as Manager and employees, Manager does the main task and employees are only allowed to look after their personal information. Manager handles the department, designation, employees, attendance details, leave lists of employees, projects, task assigned, loan sanctioned, assets with logistic support, payroll with slip, notice and sticky note like to-do-list on each dashboard. Similarly, employees are added only by the manager but with respective email address and password, employees are allowed to log into their dashboard with their individual information of attendance, projects, leave and payment lists.

Finally, the complete employee payroll system follows different criteria as security, credibility, compatibility and flexibility that makes it a complete automate system.

## **Features of the System**

Being an integrated system, it focuses on accomplishment of myriad aspects as calculating wages, managing attendance, taxes to planning employee benefits, leaves management, projects and tasks assigned and many more.

The important features required for employee payroll system are as listed below:

1. **Performance evaluation:**

Being a centralized location, managers work for employees, projects assigned, task assigned, field visits, tracking progresses, and setting goals for team.

1. **Employee scheduling:**

It focuses on scheduling shifts and attendance that helps in improving assignment practices and cooperation between departments and designations.

1. **Attendance management and time tracking:**

It includes timesheets to manage attendance which also have module related absence tracking, earned leaved, leaves with wage calculations. For advanced features, biometric integration for bulk attendance data is also managed through import csv file sheet.

1. **Payroll management:**

This feature calculates the wages automatically based upon schedules, attendance, salary grades, benefits, leaves, allowances, loans and withholding taxes. The solutions need to set up manually with rules for completion of payroll process which can be customized along with benefits structure and multiple pay configured for employees.

1. **Expense management:**

The system features a loan ledger to account for all approved loans by manager. This module handles expenses, deductions, advances, loans, reimbursements based upon approved expenses.

1. **Income tax and deductions:**

It processes pre and post-tax deductions that helps to comply with state, federal taxes and security requirements. It automates calculations upon generating payment for particular employee based upon month and year.

1. **Reports and pay slips:**

It is another important feature that helps in generating payroll reports such as salary statements of individual employee, leave summaries, loan summaries and attendance summaries. Likewise, the pay slip can be generated simply through month and year of particular employee that are easily printable in any formats as csv, pdf, excel and quick print.

1. **Individual Dashboard:**

It defines the individual dashboard for each employee where they can easily go through their particular detailed information such as leave data, personal information, leave application, loan application, loan details, payment details, project details, task details and many more. Similarly, the admin dashboard includes overall all functionalities required for a company.

## **Aims of the System**

The purpose of employee payroll system is to ensure that the company is able to achieve success through employee’s proper management. It can be a source of company or organizational capabilities allowing firms to capitalize and learn on new opportunities.

Specifically, employee payroll system aims are as listed below:

1. Develop and enhance the inherent capacities of employee such as their contributions, employability, potential and continuous development opportunities.
2. Creating productive and harmonious relationships that can be maintained between management team and employees.
3. Ensuring that the employees are rewarded and valued for their achievements.
4. Managing a diverse workforce and taking into employment needs, work style and aspirations.
5. Adopting the ethical approach to manage employees based on their concern and transparency.
6. Support the development of firm-specific knowledge and skills resulting organizational learning processes.
7. Implement and develop the policies to balance and adapt the needs of stakeholders and provide for management of diverse workforce.

## **Objectives of the System**

Being pre-determined goals to which organization is directed through individual activities. The objectives of employee payroll system that meets the needs, values, aspirations and dignity of employees concerning for socio-economic problems of the company are as listed below:

1. To utilize and create the motivated workforce to accomplish the basic company goals and maintaining its structure.
2. To secure integration of employees within the company by coordination of their goals with companies.
3. To create opportunities and facilities for employee’s development so as to meet with the organizational growth.
4. To satisfy and identify the employees needs with providing adequate wages, bonuses and security measure for challenging prestige, work, security and status.
5. To appreciate and strengthen the employee assets by providing task with field visits programs continuously.
6. To focus and achieve higher production and effective utilization of available resources in the company leading its goals and objectives.
7. To generate reports and pay-slips at any time and in any formats providing information to top-level management.
8. To prepare detailed records and data of salary of individual or group of employees with their assigned project and task.

## **Problem Statement of the System**

The study arises from the management needs of administration and accounting departments of the company more effectively. With the growing technology, competition between companies and organizations are growing rapidly that must be adaptable, agile, customer-focused and resilient. For the increment of productiveness of the company, the managers must be aware about the trending technology that helps in increasing effectiveness and improving efficiency.

# **Chapter 2: Analysis**

## **2.1 Introduction to Analysis:**

Analysis is the phase of **SDLC** **(Software Development Lifecycle)** where project lifecycle actually begins. Here, the deliverables in **Project Charter** are break-down into detailed business requirements that helps in identifying directions for creating project strategy documents. The main function of analysis is requirements gathering that clearly defines the application process depending upon its complexity. The formal process of analysis phase is developed in four basic steps as **Elicitation, Validation, Specification** and **Verification**.

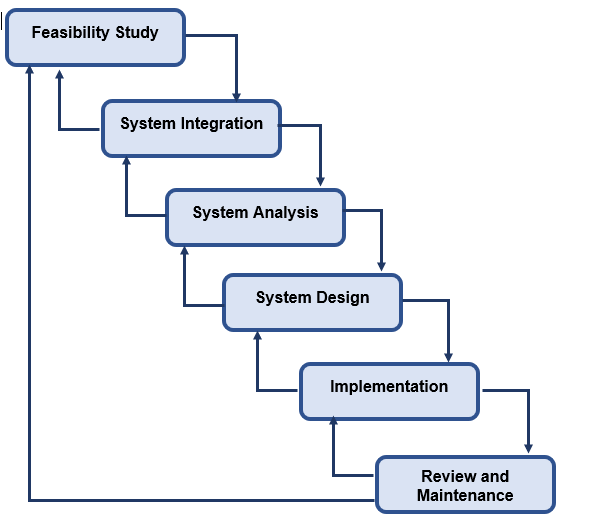
Besides requirements gathering, it helps in many other factors which are as listed below:

1. Analyzing the processes of application that indicates its **feasibility**.
2. Establishing **baseline** and control processes of the application.
3. Focusing upon aims and objectives by defining their detailed **requirements**.
4. Helps in creating **System Requirement Specification (SRS)** that represents functional and non-functional requirements along with hardware and software requirements.
5. Helps in creating technical and designed phase from **Conceptual System Design**.

## **2.2 Analysis Methodology:**

Analysis methodology is a procedure that helps in analyzing performance of system from beginning point and provides guidance to root cause. There are various methodologies for solving various issues in a system or application and some of them are as: **OOA (Object-oriented Analysis), SWOT analysis, Hard approach, Soft approach, Combined approach, Yourdon, People-oriented, Organization-oriented** andmany more.

Among those methodologies, I have preferred **Hard approach** to system analysis. It is also known as **Process Oriented Methodology** that is based upon system engineering where people are treated as passive observers during development process. It is cost-effective workforce where decision making is very quick. It is also a step by step methodology focusing upon its work-flow which can be graphically represented in charts and **DFD (Data Flow Diagram).** It functions for decomposing of system description and their requirements. After preference of hard approach, **SSADM (Structured System Analysis and Design Methodology)** is applied during my system development where six different steps are involved which are as listed below:



**Figure 1: SSADM Life Cycle**

SSADM develops a system by breaking it down into stages, steps, modules and frameworks. The main objectives of SSADM are as listed below:

1. It helps in project management and controlling the issues.
2. It effectively uses the experienced and non-experienced members.
3. The quality of application becomes better.
4. It enables applications supported by computer-based tools.
5. It develops a framework to maintain proper communication between teams.

It is also a waterfall or cascade view of development with six different phases as **feasibility** study, system **integration,** system **analysis,** system **design, implementation** and **maintenance** which are comprehensive model supported by **CASE tools.**

During hard approach to system analysis, there appears several stages and some of them are listed below:

1. The first stage begins with **identifying problem** or **opportunity** of the system.
2. It illustrates the system in detail through **diagram** as **DFD**.
3. It illustrates the system objectives clearly.
4. It evaluates routes that involves **feasibility testing** or **pilot studies**.
5. It evaluates the outcome of the system or application.

### **Data Flow Diagram (DFD / Information analysis and modelling)**

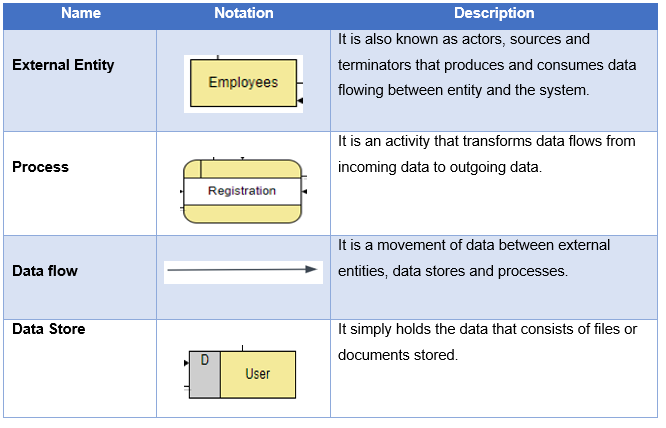
DFD is a visual representation of the information that flows in a system which clearly depicts the requirements manually or automatically. It shows the boundaries and scopes of the system.

**Reason behind choosing Data Flow Diagram**

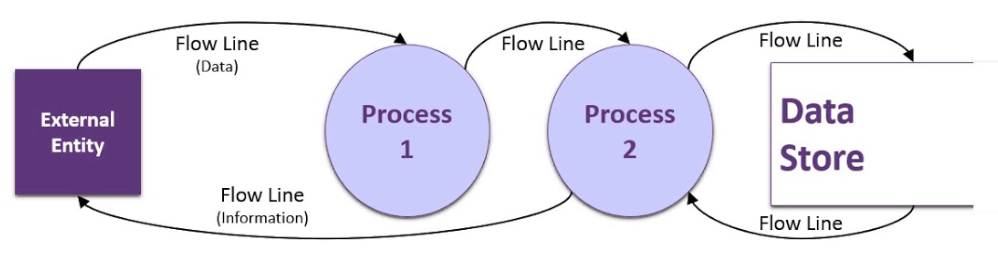
DFD describes the boundaries of the employee payroll system which is beneficial for communicating the system knowledge to the end users. It is a straightforward graphical diagram that is easy to recognize and provides detailed representation of system components. It is easier to understand and supports the logic behind data flow within the system.

**Notations in data flow diagram**

The notations applied in data flow diagram which is also called Gane-Sarson notation are represented below with its description:

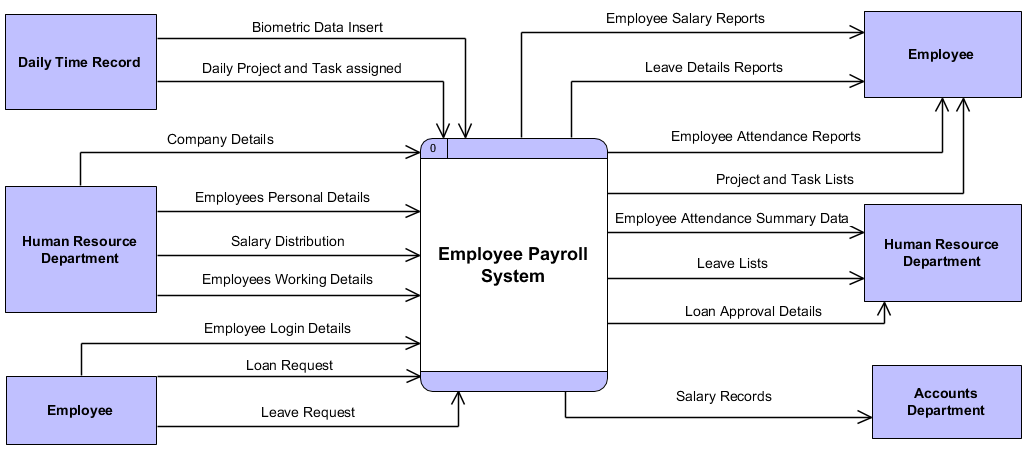
****

**Figure 2: Notation Table of Data Flow Diagram**



**Figure 3: Representation of Data Flow Diagram**

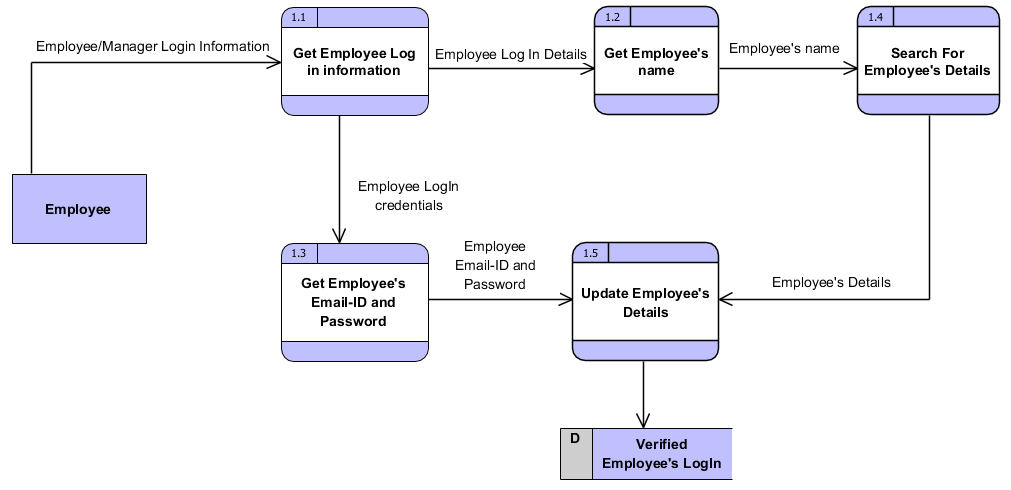
**Data Flow Diagram with Descriptions**

****

**Figure 4: Context Level Diagram of EPS**

**Context level diagram of EPS:**

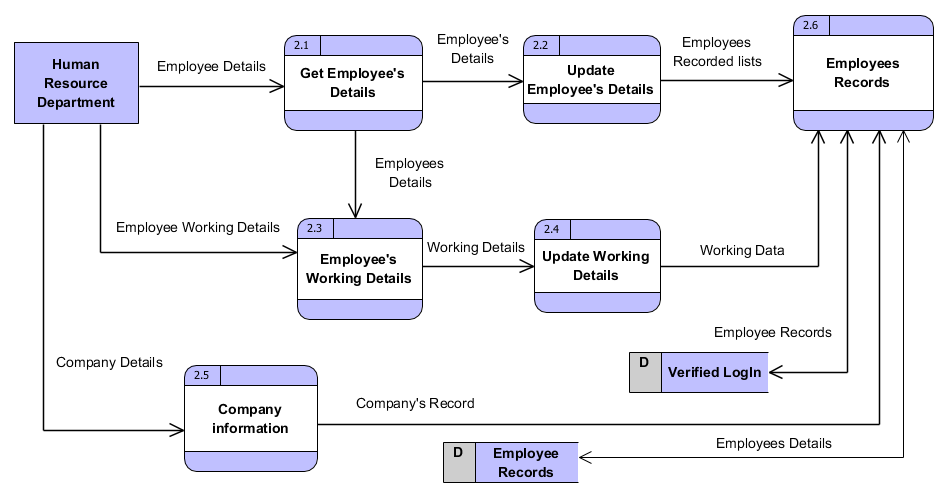
It is a first level of data flow diagram that describes the overview functionalities required by the external entities in hierarchical manner. It contains one process representing the entire system that splits into major processes providing more details. It does not contain any data storage which is also termed as diagram 0.



**Figure 5: Log in Details Data Flow Diagram of EPS**

**Log-In Details Data Flow Diagram:**

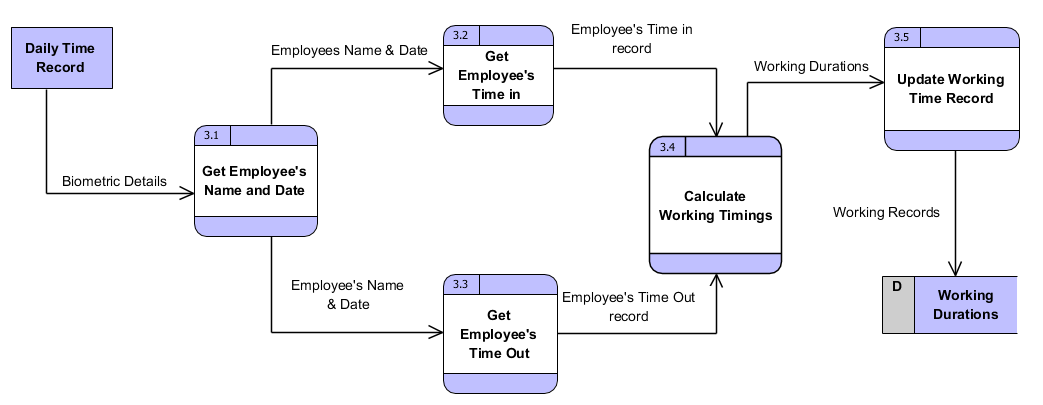
It represents the flow of data during login process where login is verified having complete data storing place and entity including processes.



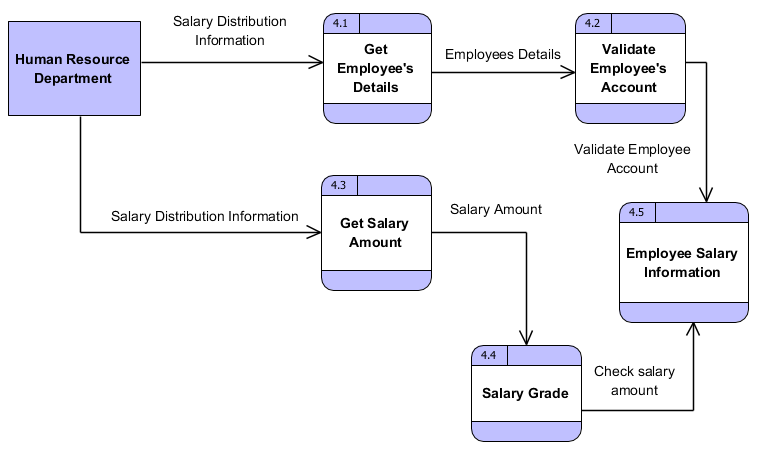
**Figure 6: Manage Employee Details DFD**

**Manage Employee Details Data Flow Diagram:**

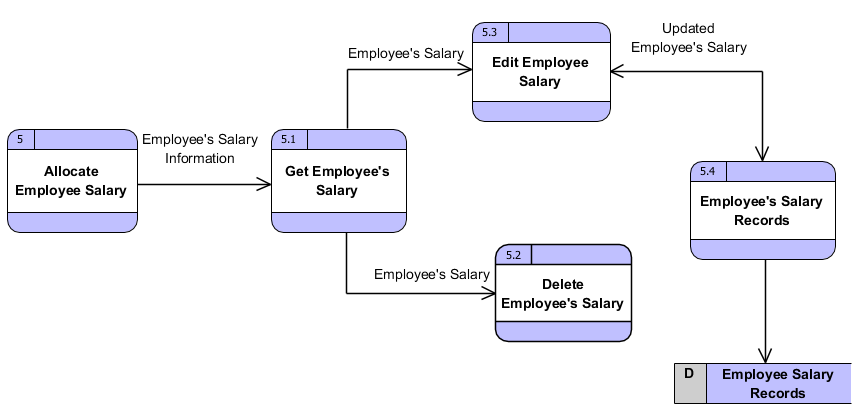
It represents the data flow of employee’s details occurred after successful login verification where data storage are employee records and verified login. It contains several processes as company details, employee details and employee working details.



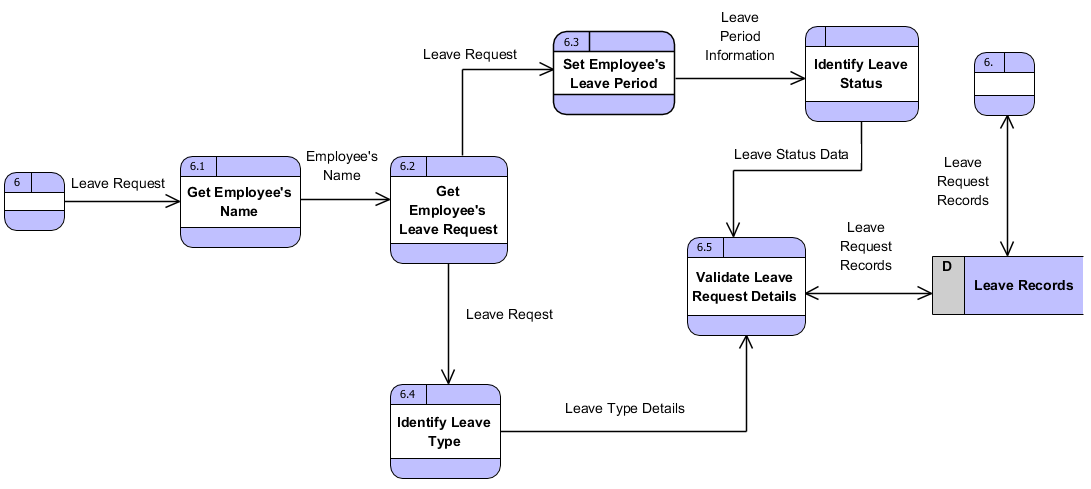
**Figure 7: Calculate Work Duration Data Flow Diagram**



**Figure 8: Allocate Employee Salary Data Flow Diagram**



**Figure 9: Update Employee Salary Data Flow Diagram**

****

**Figure 10: Leave Request Process Data Flow Diagram**

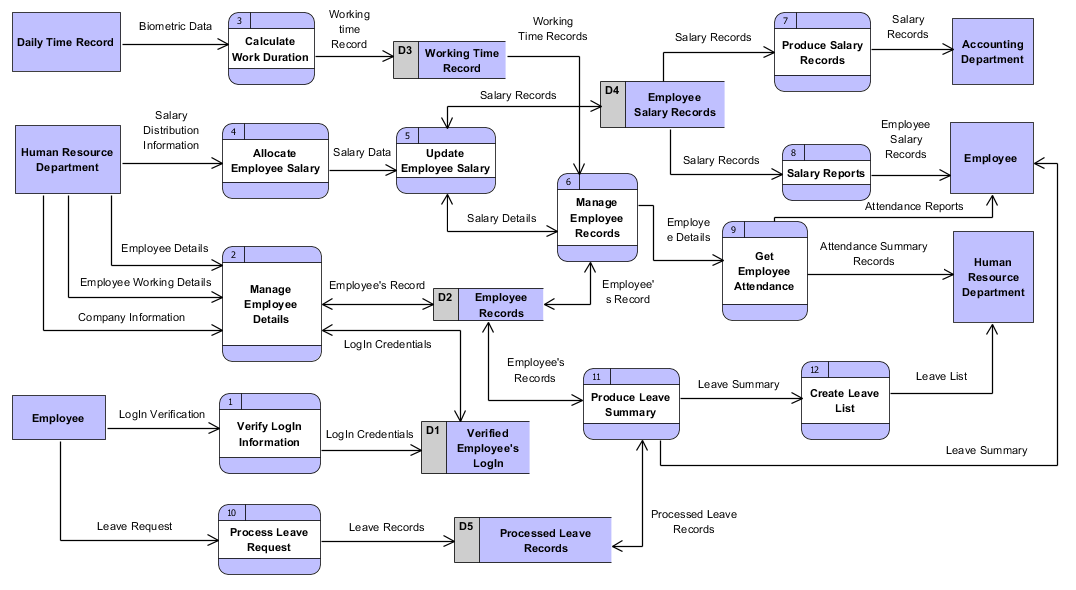


Figure 11: Employee Payroll System Data Flow Diagram

The above diagrams represent the system goals and objectives clearly with processed quantifiable data. It is also unitary view of employee payroll system with proper storage and controlled mechanisms that aims to solve problems occurring on system.

**Reason behind choosing Hard Approach Methodology over other methodologies:**

As I have chosen hard approach to system analysis over other methodologies due to some of its advantages which are as listed below:

1. It is a structured method of system analysis or system engineering which helps to solve overall well-defined problems and focuses upon technical factors foremost.
2. It is a rigid procedure and technique to provide unambiguous solutions to particular problems focusing on computer implementations.
3. For each problem, it provides definite solutions with their achievable goals.
4. It is mostly concerned with system or application dimension.
5. It is mostly suitable to address issues pertaining to structured problems.

Along with advantages, there comes some pitfalls also which are as listed below:

1. It is not concerned with dealing with people and their values, perceptions and interests.
2. It does not address the unstructured issues and it can only be applied for special cases or systems.
3. Communication might create risk.
4. It may be less flexible due to its step by step approach.
5. It is time consuming during any updates and maintenance.

## **Requirements Elicitation Techniques**

**Requirements Elicitation:**

Requirement elicitation is the collected form of requirements of the system from customers, users and stakeholders. It is important to be done for any project to improve its long-term goals and no any huge risk will occur in the future.

**Techniques of requirements elicitation:**

For long term system success, we need a solid elicitation to be performed before development phase. Requirements elicitation helps in growth of business continuously through practicing new techniques which is a key role for project success.



**Figure 12: Requirements Elicitation Techniques**

There are several techniques for requirement elicitation which are as listed below:

1. **CATWOE (C**ustomers**, A**ctors**, T**ransformation, **W**orldview, **O**wners, **E**nvironment**)**
2. **SWOT** analysis **(S**trength, **W**eakness, **O**pportunities, **T**hreat**)**
3. **PEST** analysis **(P**olitical, **E**conomic, **S**ocial, **T**echnological**)**
4. **Interviews**
5. **Stakeholder matrix**
6. **Surveys**
7. **Focus groups**
8. **Brainstorming**
9. **Prototyping**
10. **Questionnaires,** etc.

Among all techniques, I have chosen some of the important techniques which are as listed below:

1. **Brainstorming**

It is an exceptional method that supports diversion sort of thoughtfulness which helps in creating assorted set of opportunities. It emphasizes the issues that offers many fundamental ways out. It helps to get perfect solution of each existing problems through single brainstorming or group brainstorming.

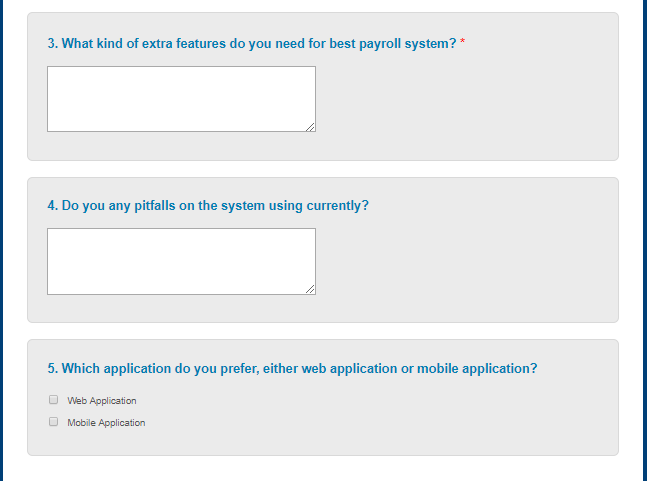
1. **Observations**

It is the most useful technique where it helps picking up useful present information that helps for documentation through diagramming and business procedures prototypes as use cases. Observation eradicates the overall confusions and every task can be performed smoothly. Similarly, for this project **Employee Payroll System (EPS)** to overcome confusions ‘**PaySquare**’ has been observed.

1. **Questionnaires**

It is a series of questions for gathering information which can be performed face-to-face, by telephone or computer. It is quick and cheap method for gaining quantitative and qualitative information.

The sample of employee payroll system questionnaire is as shown below:



**Figure 13: Questionnaires as a requirements elicitation technique**

## **Feasibility Study**

Feasibility study can be applied for each and every system with unlimited resources. It is also an evaluation of system regarding its impact on organization, work ability, effective use of resources and ability to meet user needs. Whenever new system is proposed, it goes through feasibility study before development phase.

We can say that feasibility study and risk analysis are co-related in many ways, such as if project risk becomes greater then feasibility gets reduced. Feasibility study helps to evaluate the success of system where perceived objectivity is the most important factor.

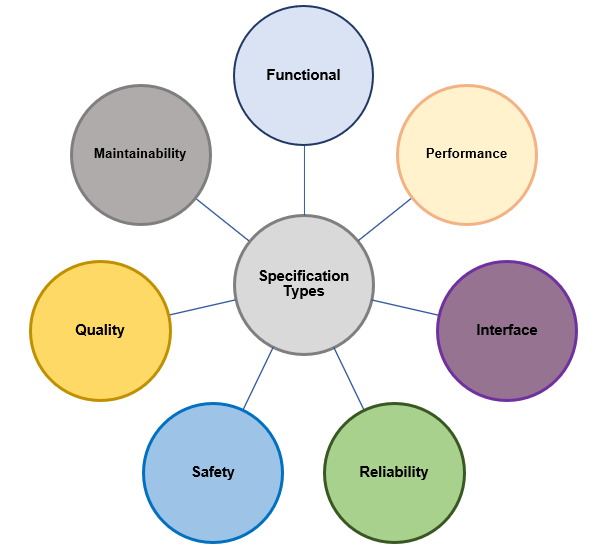


**Figure 14: Feasibility Study Details**

## **2.5 Software Requirement Specification (SRS)**

**SRS (Software Requirement Specification)** which is also known as **requirements documentation** is an output of the requirement phase of software development process. It is a foundation for software engineering activities from analyzed requirements that includes overall system requirements and detailed specifications of the system.

The software requirement specification determines **functions, design constraints, performance** and **quality attributes** of the software which are verified through prescribed methods such as inspection, analysis, test or demonstration. SRS is required to verify and specify the requirements of organization. **Functional, non-functional, assumptions, technical requirements** and many more are documented during SRS.

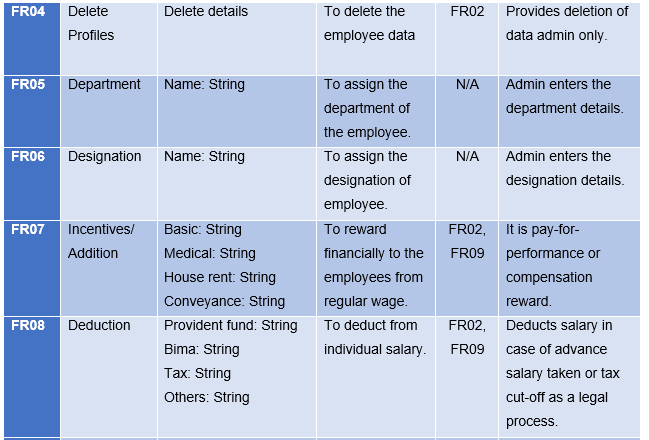


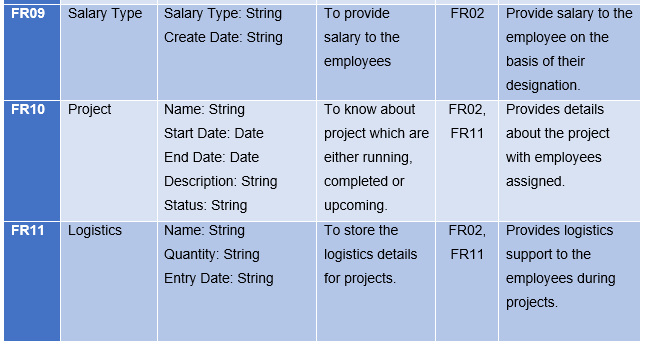
**Figure 15: Software Requirements Specification Types**

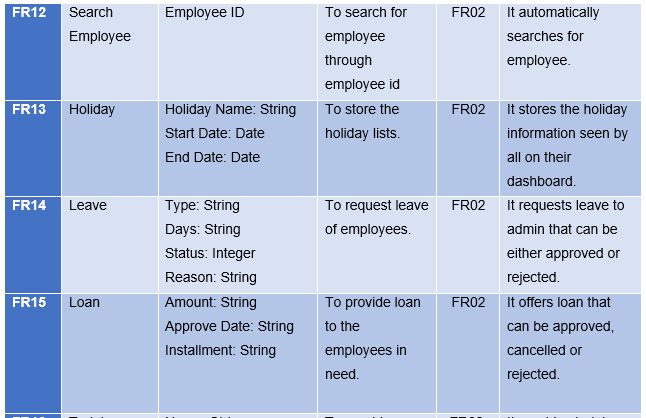
### **2.5.1 Functional Requirements**

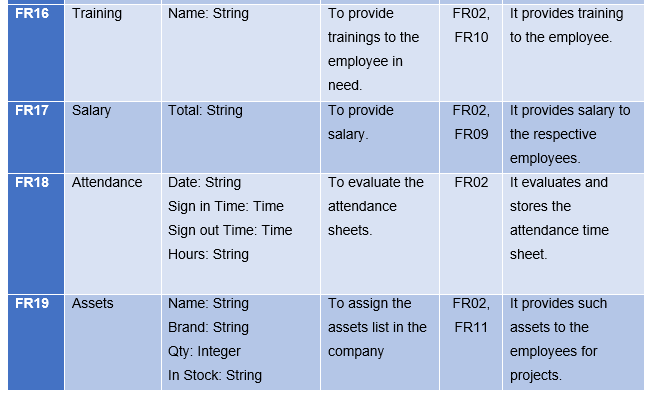
Functional requirements are those requirements that includes operations and activities to be performed by a system which includes data descriptions, operations, work flow or performances, reports or outputs and control access activities for the system or application. With the help of functional requirements, users can easily understand about the functionality of the system.

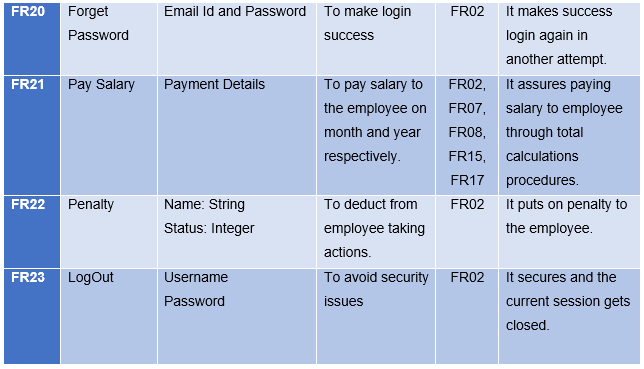
The functional requirements of employee payroll system are being depicted through **function, data, rational, dependencies and remarks** also which is as listed below:









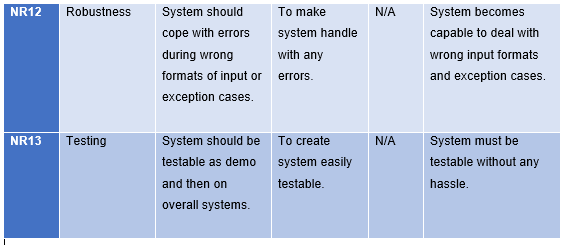
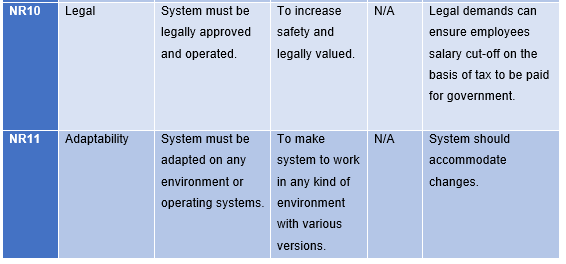
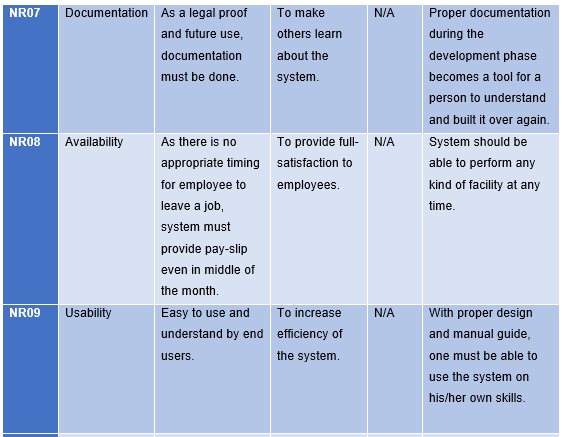
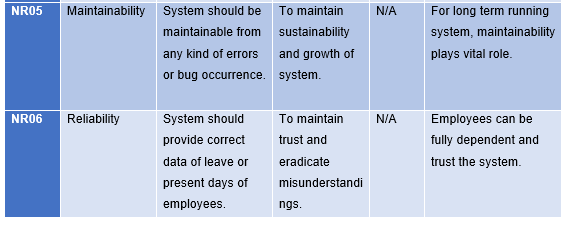
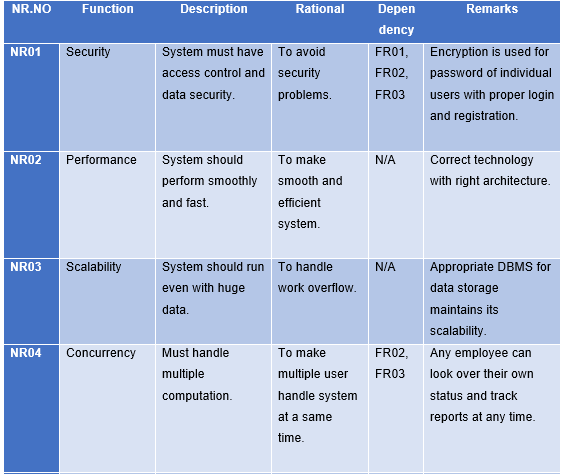


**Figure 16: Functional Requirements of Employee Payroll System**

### **2.5.2 Non-Functional Requirements**

Non-functional requirements are those important characteristics or requirements which are not essential for the system but these are required for better system. It is also known as quality attributes that includes attributes as **security, performance, usability, compatibility** of the system. Among overall non-functional requirements, they must be quantifiable and objective.

The depicted non-functional requirements for EPS are as shown below:



**Figure 17: Non-Functional Requirements of Employee Payroll System**

### **2.5.3 MoSCoW Prioritization**

MoSCoW prioritization is a **prioritization technique** which is also a method or analysis for managing overall requirements that can help stakeholders to know about most significant areas in a system.

The MoSCoW prioritization categories are as,

**M**ust-have initiatives

**S**hould-have initiatives

**C**ould-have initiatives

**W**ill not have (this-time)

**How does MoSCoW prioritization work?**

After proper alignment on objectives and prioritization factors from stakeholders and product team, all must prioritize to it in the form of consensus on percentage that is prioritized on particular technique. One can prioritize such techniques either by 80/20 rule or simply 60/20/20 rule where on 80/20 rule: 80% is provided to Must have and Should have and last 20% to Could have and Would have. Similarly, on 60/20/20 rule: 60% is provided to must have and should have and 20/20 % to could have and would have.

All the techniques are prioritized on the basis of functional departments that shows team to determine required effort into each category which can be modified and made effective.



**Figure 18: MoSCoW Prioritization**

**The prioritization made for mentioned functional and non-functional requirements**

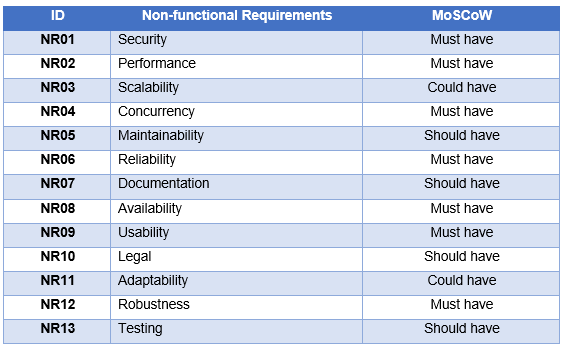
The MoSCoW prioritization made for both functional and non-functional requirements are as listed below:

**Functional Requirements Prioritization:**



**Figure 19: Functional Requirements Prioritization**

**Non-Functional Requirements Prioritization:**



**Figure 20: Non-functional Requirements Prioritization**

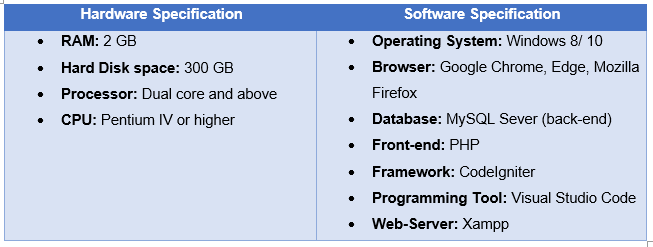
**Reason behind applying MoSCoW prioritization techniques:**

For making a product successful, MoSCoW prioritization technique plays a vital role to rank and classify requirements. As it based upon experts’ opinions of the team, it is quicker and easier to complete. It is a technique defining priorities of the project that are in progress.

Similarly, it has some weakest points also as, it can be subjective and requires the team to have familiarity with the features of employee payroll system.

### **2.5.4 Hardware and Software Specification**

The hardware and software specification of the Employee Payroll System are as listed below:



**Figure 21: Hardware and Software Specifications**

## **2.6 Use Case Diagram**

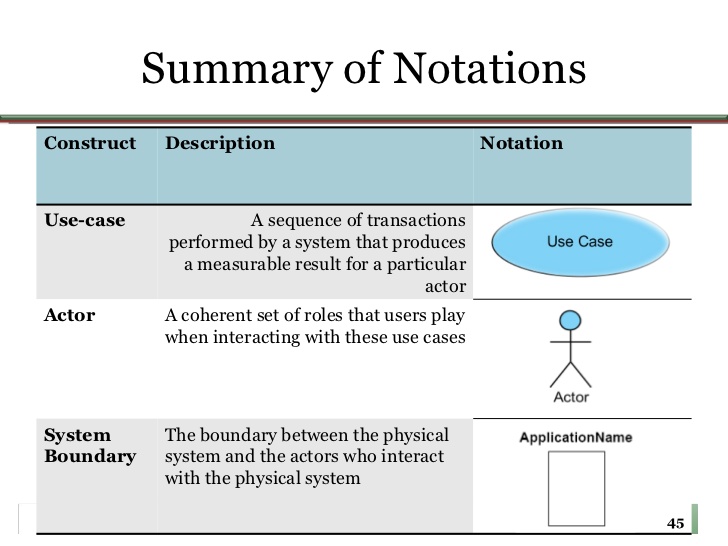
Use case diagram is a behavior diagram that **models the functionalities** using actors and use cases in **UML (Unified Modeling Language)**. The use case defines the functions or task which are done by particular actors. It is also a collection of diagram and text together which are drawn on **Star UML.**

It is also a formal way of representing a business system interaction with environment. It illustrates the activities that needs to be performed by the users in a employee payroll system. It is a **scenario-based technique** in UML containing sequence of actions with particular actor.

**Reason behind choosing Use Case Diagrams?**

Use Cases are the most important factor for any system during development due to following reasons:

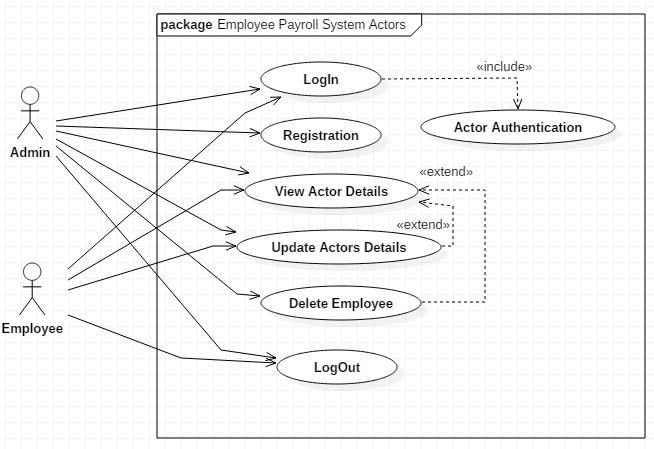
1. Use case helps designer to think about design of the system from different perspectives which is an easy method.
2. It engages users to makes them understand about the system and view their points or needs too.
3. It provides context for requirements of the system.
4. Use Cases represents the understandable features or responsibility by developers.
5. Being a critical tool, it helps in reducing risk of inconsistencies while going through design and implementation.
6. It serves as inputs for the documentation and can carry directly over testing process.



**Figure 22: Use Case Diagram Notations Summary**

Similarly, in employee payroll system project, there are two actors doing overall activities in the system which are **Admin** and **Employee.** The **Use Case diagram** of Employee Payroll System are as listed below:

1. **Use Case Diagram for Log in and Registration**



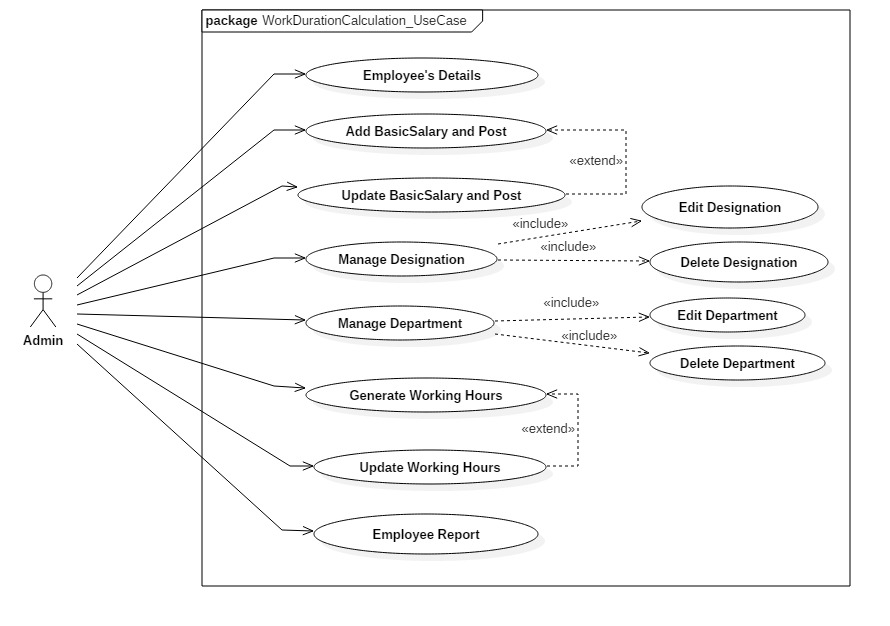
**Figure 23: Actors during registration and log in use case diagram**

**Scenario Description:**

**Actor: Admin and Employee**

The above use case diagram represents the actors as Admin and Employee that works on Employee Payroll System. The diagram shows Login with particular users’ details, along with their CRUD performances and some of them are listed below:

1. Both admin and employee can login to the system that leads to authentication (include) which confirms correct username and password.
2. Incase employees are not registered, sign up form or registration is available to manager only where he/she can input employee’s appropriate personal details in correct format or order.
3. After successful login, actors can go to their respective dashboard where they can verify their information.
4. On the basis of authority, employee can update their own personal information but cannot delete their data themselves.
5. Likewise, Admin can update personal details with deletion of the employees after their leave agreement from the organization.
6. There is also a logout process where their current session gets closed which is an important factor for security.
7. **Manage Employee Use Case Diagram:**

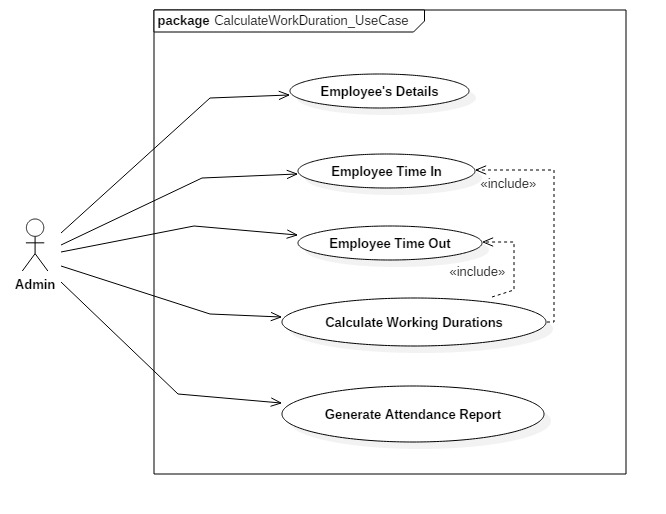
****

**Figure 24: Manage Employee Use Case Diagram**

**Scenario Description: (Actor: Admin)**

It describes the functions performed by admin during managing employee where It does CRUD operations for employee, Working Hours, Department, Designation, Salary and Posts and their reports on the basis of the individual employees.

1. **Calculate Work Duration Use Case Diagram:**

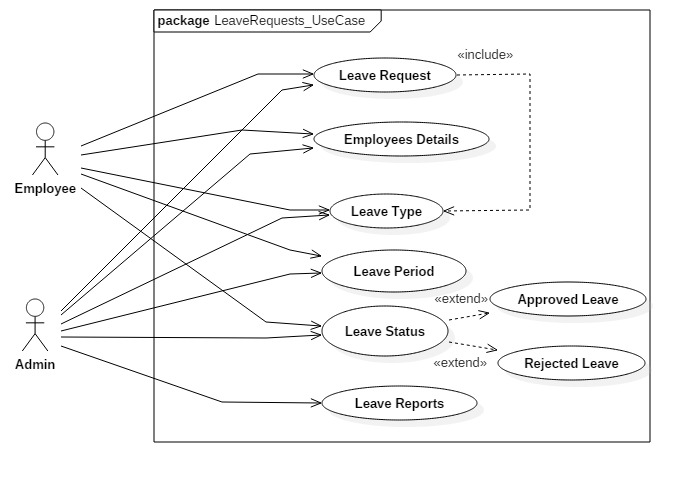
****

**Figure 25: Calculate Work Duration Use Case Diagram**

**Scenario Description: (Actor: Admin)**

It describes the attendance phase of employee payroll system where admin or manager inserts the employee’s working time in and time out one by one. Simply, admin is allowed to add attendance sheets in csv formats as a bulk data. Calculating work duration of employees includes both time-in and time-out with dates and employee-id in a company.

1. **Leave Request Use Case Diagram:**

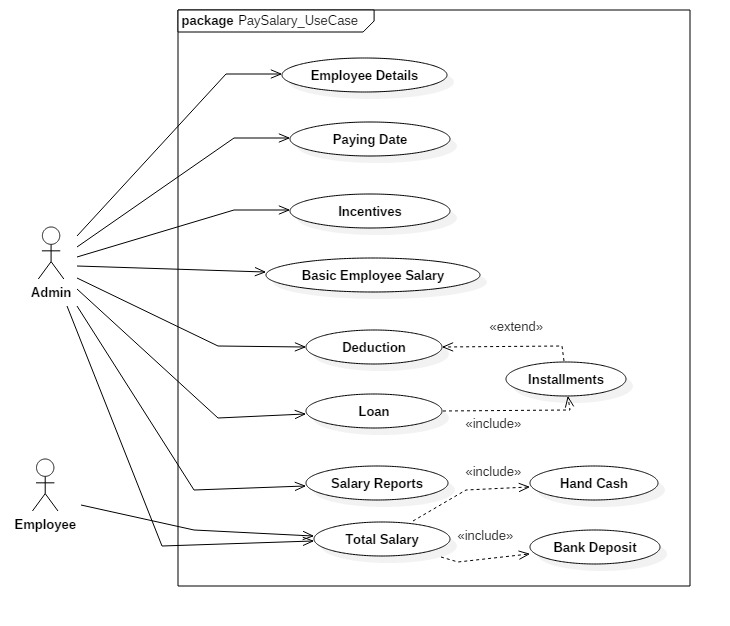


**Figure 26: Leave Request Use Case Diagram**

**Scenario Description: (Actor: Admin and Employee)**

It describes that the leave request will be send by employee to get approved from admin/ manager with full details on number of days, leave type, reason behind leave and many more. The request sent by the employee will be received by manager/ admin where he/she judge the leave application and forward the result that again sends to employee.

1. **Pay Employee Salary Use Case Diagram:**

****

**Figure 27: Pay Employee Salary Use Case Diagram**

**Scenario Description: (Actor: Admin and Employee)**

It describes the payments provided by admin/ manager to the employees after adding incentives and deducting from loans, absent case and many more. The salary will be provided either hand cash or directly to the bank.

1. **Employee Work Flow Use Case Diagram:**



**Figure 28: Employee Work-Flow Use Case Diagram**

**Scenario Description:**

**Actor: Employee**

The use case diagram above shows the work flow and operations that can be performed by employee on employee payroll system. With the help of system, employee can know about their own track report and can fully-trust over system with full satisfaction.

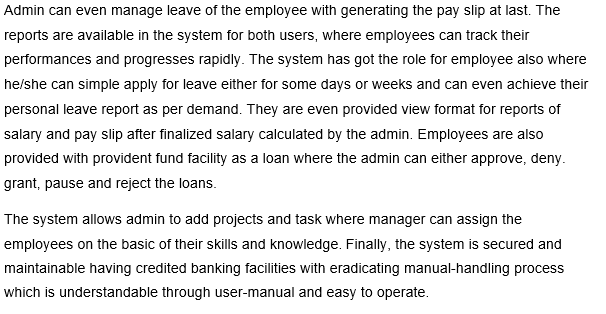
## **2.7 Natural Language Analysis (NLA)**

**Natural Language Analysis (NLA)** is the process where we redefine a **problem statement** through observations and discussions. During analysis, we can identify **nouns** as common, **verbs, attributes** and **relationships**.

As it is not an accurate method, it identifies many **false positives** and **outside the scope** which can be called as **first draft.** Beginning with nouns, it helps to create **core of the class diagram.**

**Problem Statement/Domain (Scenario)**

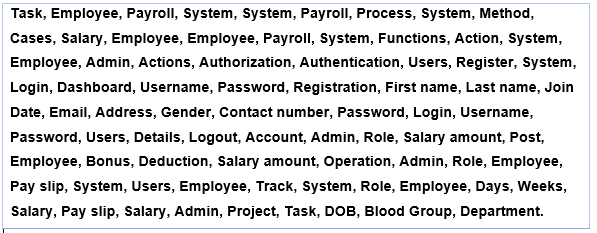




**Figure 29: Problem Domain of EPS**

**Step 1: Finding nouns from scenario for potential classes:**

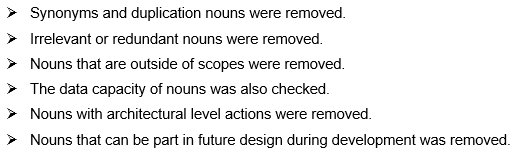
Proper noun provides us **candidate classes** that are reasonably refined from problem statement. Likewise, the nouns from the scenario are as listed below:



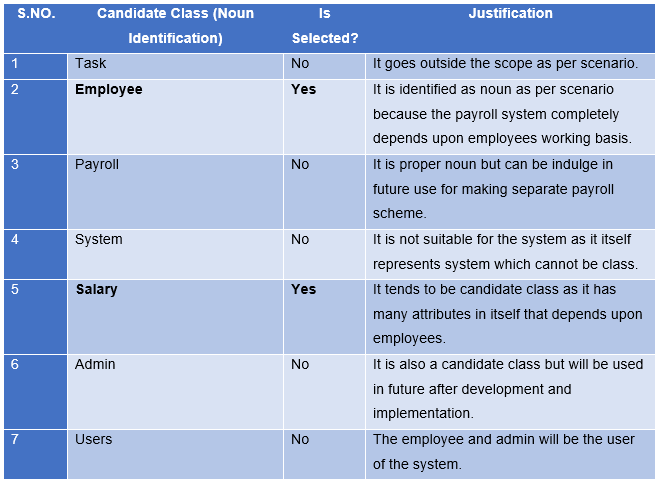
**Figure 30: Nouns for Potential Classes**

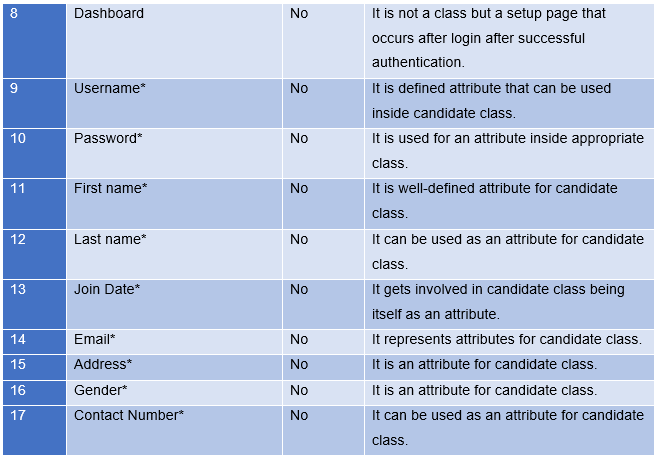
The above listed are nouns that are identified after studying scenario.

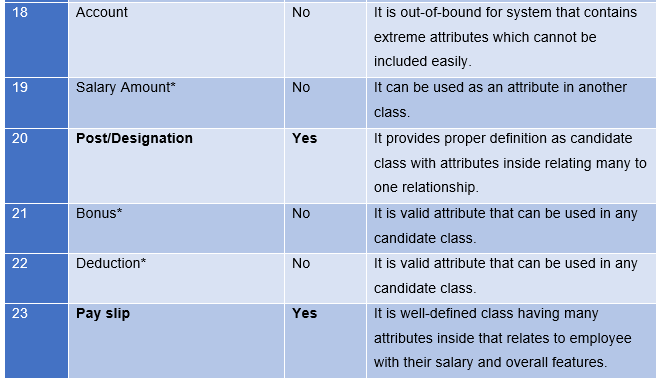
**Filtration must be applied on nouns listed concerning many factors for obtaining final potential classes, which are as listed below:**

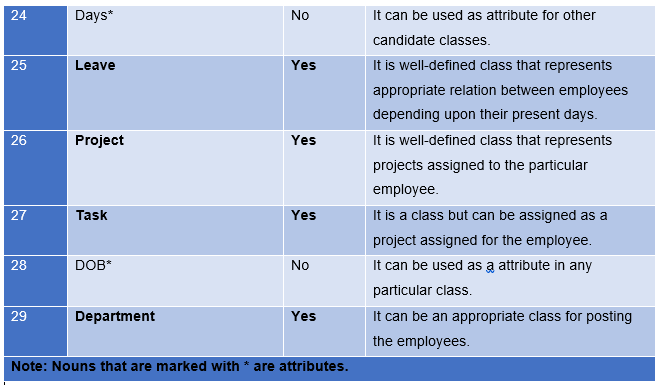


**Step 2: Noun Identification, Selection and Justification for Potential Class**

****

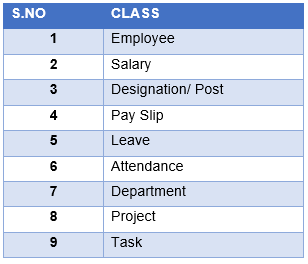
****

****

****

**Figure 31: Nouns Identification**

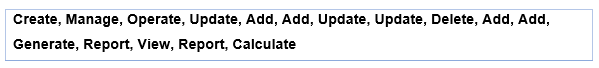
**The list of final candidate class is as listed below:**



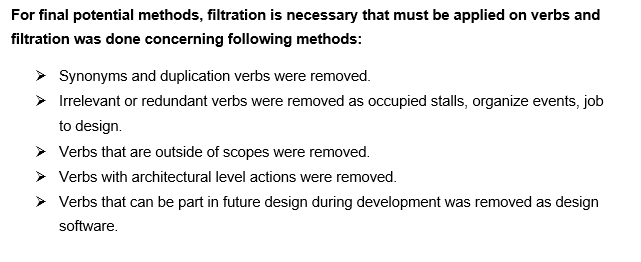
**Figure 32: Final Candidate Class Lists**

**Step 3: Potential methods / Operations as Verbs are found from scenario:**

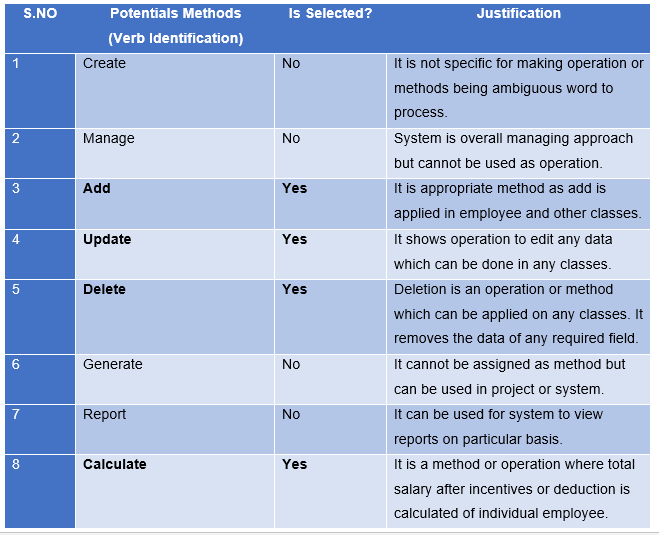
The candidate operations or verbs provided from scenario are as listed below on table:



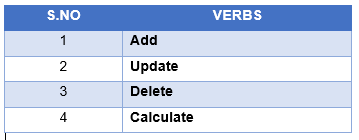
**Figure 33: Potential Methods from scenario**



**Step 4: Verb Identification, Selection and Justification for Potential methods**

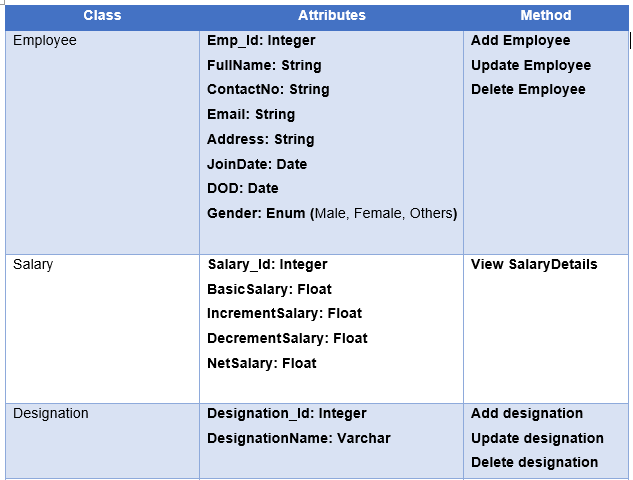
****

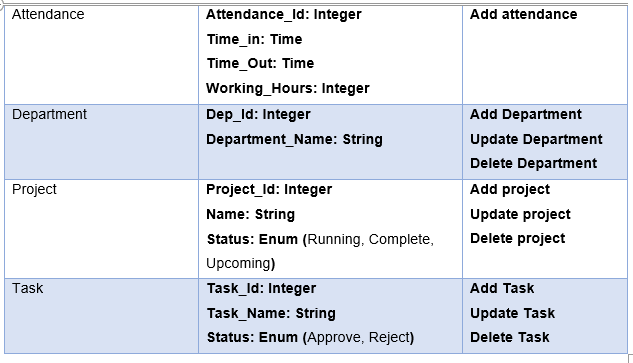
**Figure 34: List of potential methods with specification**

**The list of final candidate methods or operations are as listed below:**

**Figure 35: Final Lists of Methods**

**The final classes and attributes and finalized methods from step 2 and 4 are as listed below:**

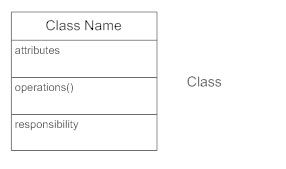




**Figure 36: Lists of final classes, attributes and methods**

**Initial Class Diagram derived from NLA**

Class diagram is a type of **UML (Unified Modeling Language)** that represents the structure of a system by modeling into **classes, attributes, operations and relationships**. It helps to know about the plan and functionalities of a system clearly. Overall, it is a collection of the classes, interfaces, associations, collaborations, and constraints.



**Figure 37: Classes Figure Summary**

**Why do we need class diagram after NLA?**

There are many benefits of class diagram that reflects upon organization and some of them are as listed below:

1. Class diagram helps to illustrate data models for system without any difficulties.
2. It can be easy to understand even by the general people.
3. Specific needs of a system can be easily expressed to the organization.
4. It helps in creating detailed chart with their main functionalities to be implemented carefully.
5. It provides an implementation-independent description involved in system.

**Basic Components of Class Diagram**

There are three section composed in standard class diagram which are as listed below:

1. **Upper Section:**

Upper section contains class name which are classifier or object.

1. **Middle Section:**

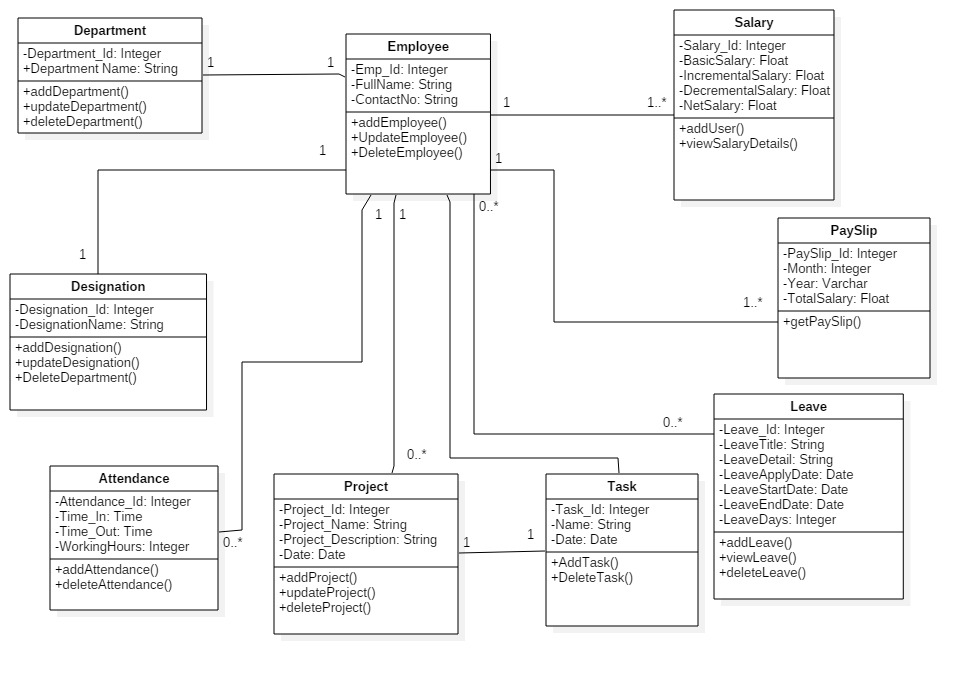
It contains attributes of class that describes qualities of class.

1. **Bottom Section:**

It contains operations or methods of class that describes the interaction with class.

The class diagram derived from NLA are as listed below.

**Initial Class Diagram derived from NLA:**



**Figure 38: Initial Class Diagram Derived from NLA**

# **Chapter 3: Design**

## **3.1 Introduction of Design**

Design phase is a major phase in software development life cycle where **architecture** is established. It breaks down the requirements and leads us from **problem domain** to **solution domain**. In this phase, required level of effort and amount of resources can be easily estimated with application workflows. It represents the **‘how’** phase; where components, interfaces, behaviors, etc. are planned to implement for the development of best application.

It explains in detail about the required specifications, features and operations as a functional requirement that helps in determining needs for the proposed application or system. This phase helps in considering essential components as hardware and software structure with their processing and procedures for accomplishment of the systems objective. It includes **structural** model, **behavioral** model, **database** model, **architectural** model and **UI design** which will be designed using various drawing tools as **Star** **UML, Balsamiq** and **Visual Paradigm**.

## **3.2 Modeling Software**

Modeling software is a modeling language that builds simulations or other models which expresses information in a structured way. It is an abstract model of a system that clary defines the systems perspective. It can be developed from existing system as well as new system which helps in implementation phase in future. It helps the analyst to understand the models and functionalities of system that are used for communication with clients.

## **3.3 Structural Model**

Structural model is both **static** and **dynamic model** that shows a structure of a system in one hand and the executing time period in another hand. It represents the system’s components and relationships which are created from discussion and designing the architecture of system or application. Likewise, I have chosen Class Diagram and Context Diagram among all structural model.

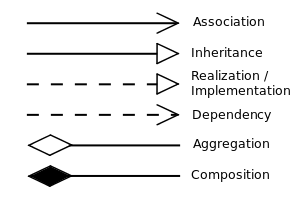
### **Class Diagram**

Class diagram is a dependency among classes in **Unified Modeling Language (UML)** from illustrated relationships and source code. These are used while developing an object-oriented system model to show the classes and **association** between them. It is useful in Object-oriented programming as OOP modeling paradigms. The classes are arranged in groups in a class diagram for sharing characteristics which resembles flowchart that contains methods, attributes and operations. It is also a **blueprint** of an object-oriented modeling.

**Reason behind choosing Class Diagram as a Structural Diagram**

Class diagrams provides insight to the system structure and gives a sense of orientation. Class diagrams are simple, fast and easy to read. It is a foundation point for creating any system. It helps in visualizing the paths between classes which takes various forms of generalization. It is a major part during development of system that illustrates the complete structure of the system. Class diagram displays **attributes, operations** and their logical and statistical **inter-relationships**.

**Notations in class diagram**

****

**Figure 39: Relationship between classes notations**

The notations define the relationships between classes which are of many types such as association, inheritance, realization, dependency, aggregation and composition. Among them, the notations applied on this methodology are shown in the table below:

|  |  |  |
| --- | --- | --- |
| Name | Notation | Description |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Final Class Diagram and Description**

### **Context Diagram**

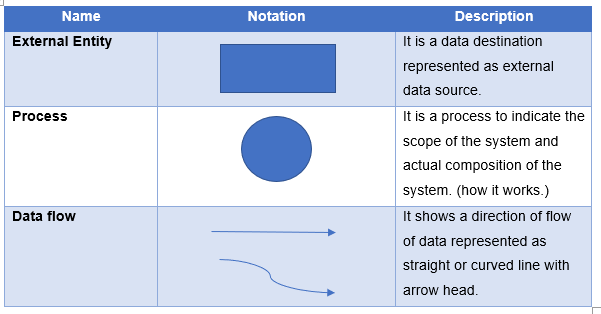
The context diagram is a single high-level process that goes under system consideration showing relationship having external entities as organizational groups, external data stores and many more. It supplies and receives information to and from system boundary which is also called **Context-Level Data-Flow Diagram** or **Level 0-Data Flow Diagram.** It is created for system analysis and design that helps to communicate interactions and data-flow between various business processes.

**Reason behind choosing Context Diagram as a Structural Diagram**

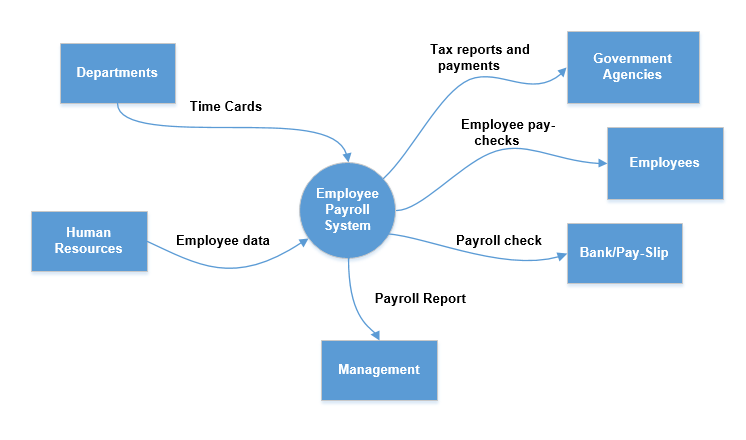
There are many benefits of context diagram that supports during implementation of the system where scopes and boundaries of system are clearly shown. Most important, it is easy to draw and expand with limited notations. For context diagram, there is no need of technical knowledge and understanding.

**Notations in context diagram**

The notations applied in context diagram are represented below with its description.



**Context Diagram and Description**

****

The figure displays overall scenario of the employee payroll system **EPS** that depicts the input and output on the basis of their particular roles. Here, the department figures out the individual employee working basis per month or week or a day from different departments. Human resources department stores the employee’s details and the individual salary will be figured out through the management with overall allowance and deductions. Finally, with tax payments the total salary will be provided to the employee through either pay-slip or directly bank deposit. The management also provides the detailed reports of the employees.

## **3.4 Behavioral Model**

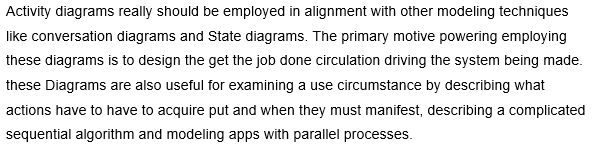
Behavioral model is an internal dynamic aspect of the system supporting business processes describing internal logic without specifying implementation process. The detailed design is fully specified for design and implementation phases. In this model, UML (Unified Modeling Language) diagrams such as sequence diagrams, communication diagrams and behavioral state are used as well as **CRUDE (Create, Read, Update, Delete, Execute)** operations are used.

### **Activity Diagram**

Activity diagram is an UML diagram that describes dynamic aspects of application or system. As it represents flow of activities, it describes an operation of the system and also known as a flowchart.

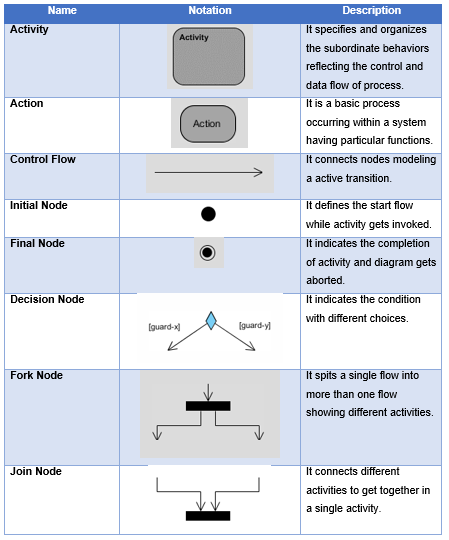
**Reason behind choosing Activity Diagram as a Behavioral Diagram**

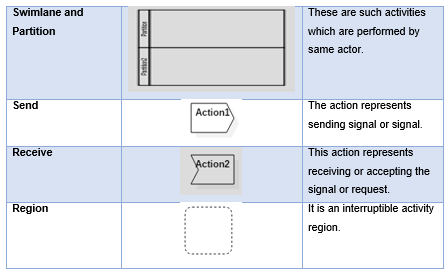
Activity diagram are easily comprehensive for both stakeholders and analyst and it is easy to understand the work-flow of the system. The diagram is very user-friendly and also regarded as important tool for analysts. It allows for multiple conditions and actors using swimlanes within a work-flow.

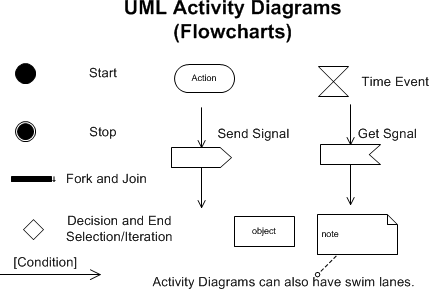


**Notations in activity diagram**

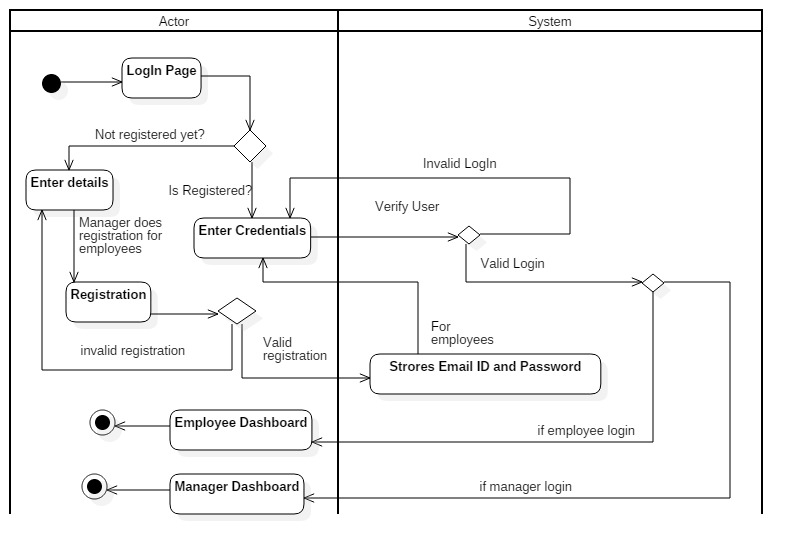
The notations that can be applied for activity diagram are as represented in tabular form below:



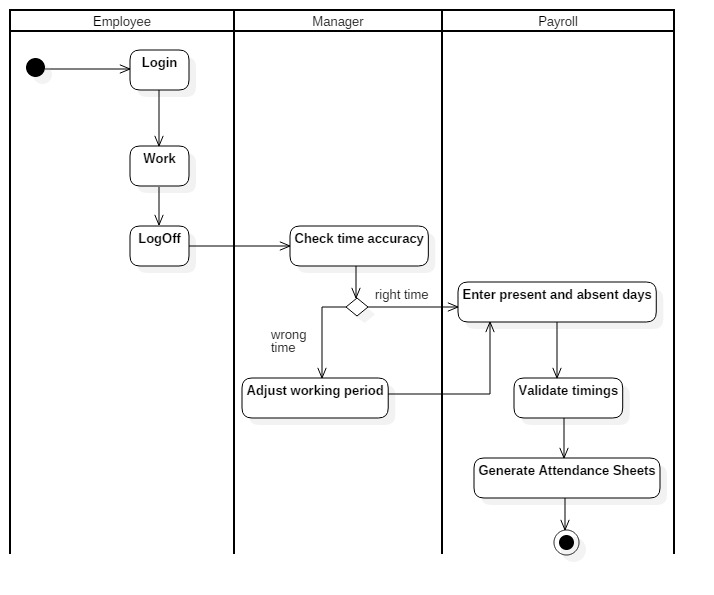




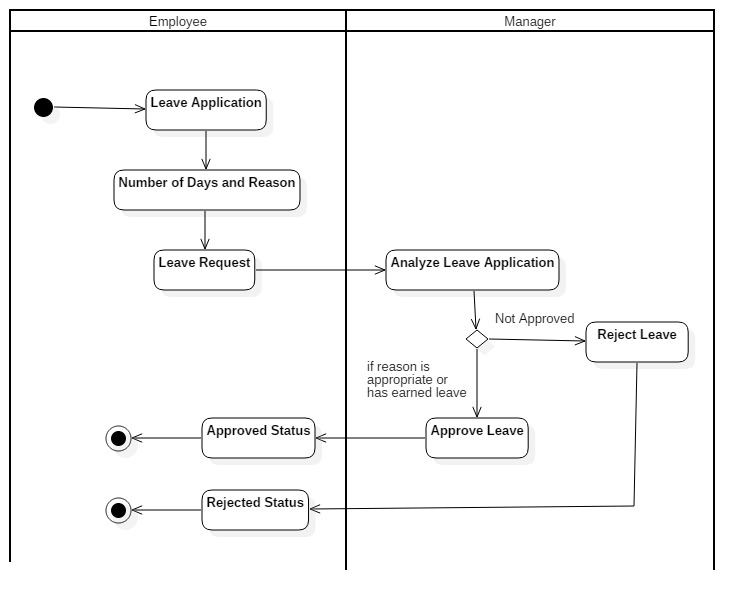
**Activity Diagrams and Description**

****

In this activity diagram, the login and registration process are described in brief where manager registers first with detailed and valid information and then using credentials, he/she can go to respective dashboard. (**Admin dashboard**). Similarly, employee’s details can only be registered by manager where username and password are provided for individual employees. The further employee login can be done through same username and password and get to their respective employee dashboard. The password can be updated after successful login.

****

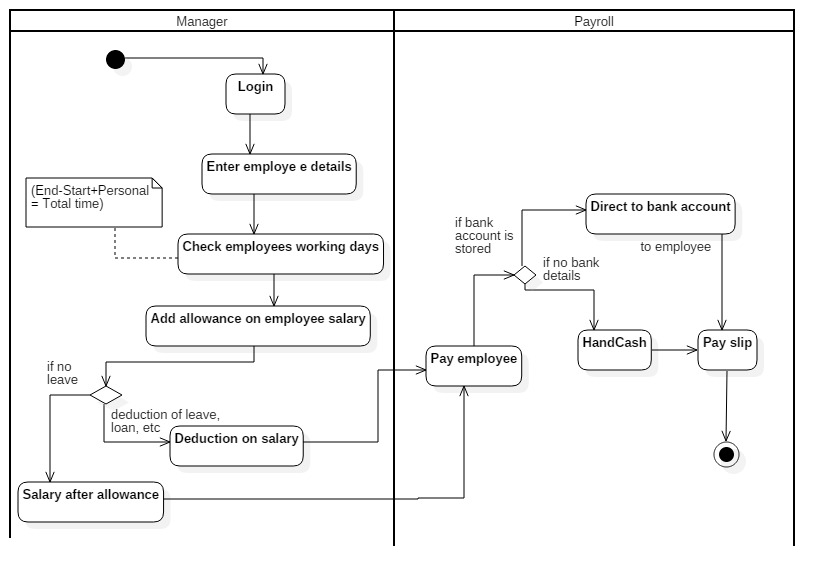
In this activity diagram, there are three swimlanes or partitions as employee, manager and payroll where different operations or functionalities are performed in particular partition. Actions, activity and decisions are applied in this activity diagram with evaluating accurate time period of employees’ working period where over times and leaves are evaluated. Finally, on the basis of timings calculations employees’ salary are paid with pay-slip.



In the above request leave activity diagram, having two swimlanes as employee and manger represents the process of leave request send by the employee to the manager where manager needs to approve or reject the activity diagram on the basis of employee’s working criteria and his/her disciplinary actions.

****

In the above Assign Project activity diagram, employee assigns the project and task to the particular or a team of employee to get involved into project. The tracks of timing and task assigned must be kept updated to manager day to day. The projects are to be assigned on the basis of their skills and knowledge. The note on project assigns the project as a running, completed or upcoming status.

****

In this activity diagram, there are two swimlanes or partitions as manager and payroll where manager sets the employee salary and departments during adding details. The assigned salary gets increased from bonus, medical or many other things. Similarly, the salary gets deducted from work leave or loan taken from organization. Finally, the salary gets calculated and pay-check is provided to the employee either hand cash or directly deposited on bank account. This completes employee payroll system.

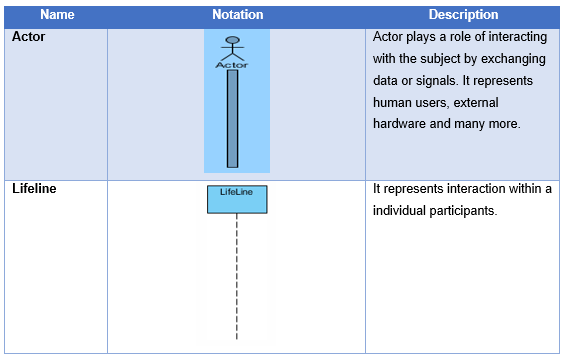
### **Sequence Diagram**

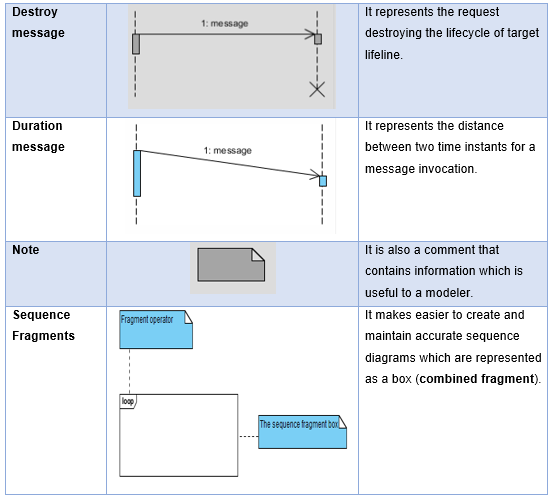
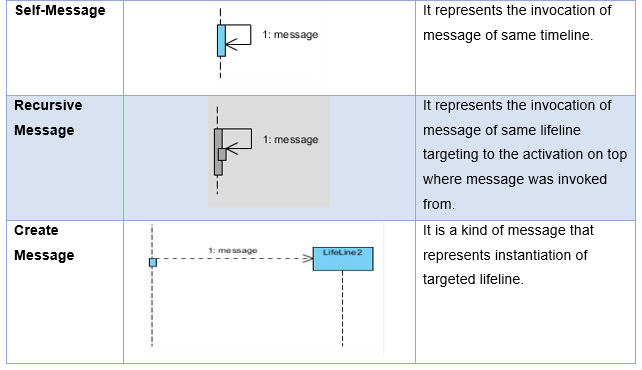
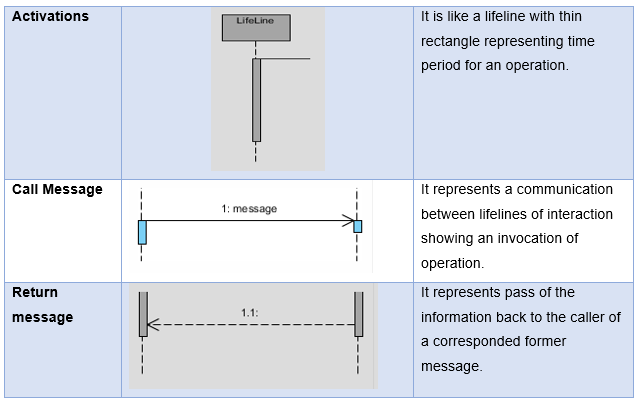
Sequence diagram are the interaction diagrams that briefly describes about the operations performed in the system that are time focused. It exchanges the messages over time which are also known as event diagrams. It is a good way of visualizing and validating runtime scenarios that helps in modeling new system.

**Reason behind choosing Sequence Diagram as a Behavioral Diagram**

As a behavioral diagram, I have chosen sequence diagram due to its high-level interaction between objects within a collaboration realizing use case, operations and paths or flows. Moreover, sequence diagram is easier to read and understand that allows reverse engineering. The UML specification is more sequence diagram centric and are excellent for documentation purpose. The diagram is arranged in a time sequence where messages are exchanged between objects.

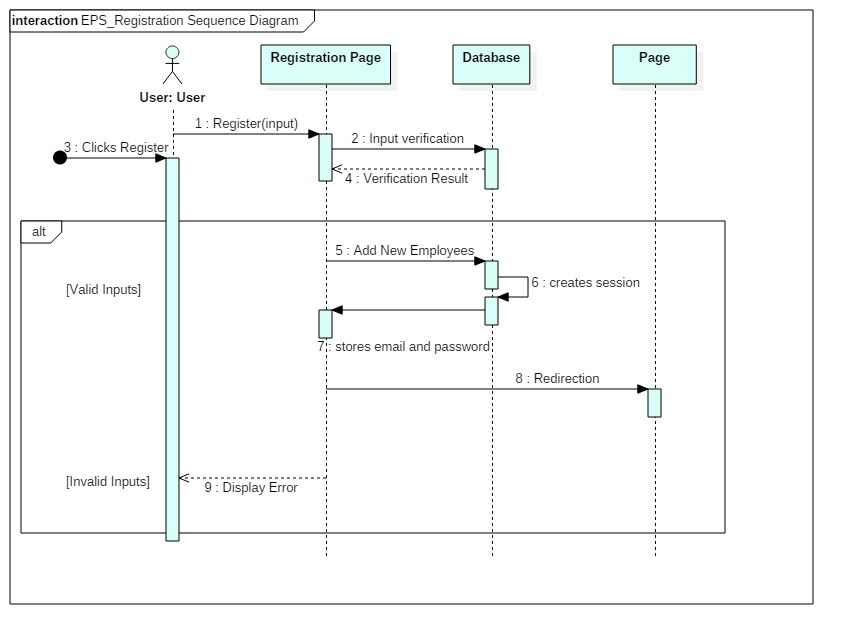
**Notations in sequence diagram**

****

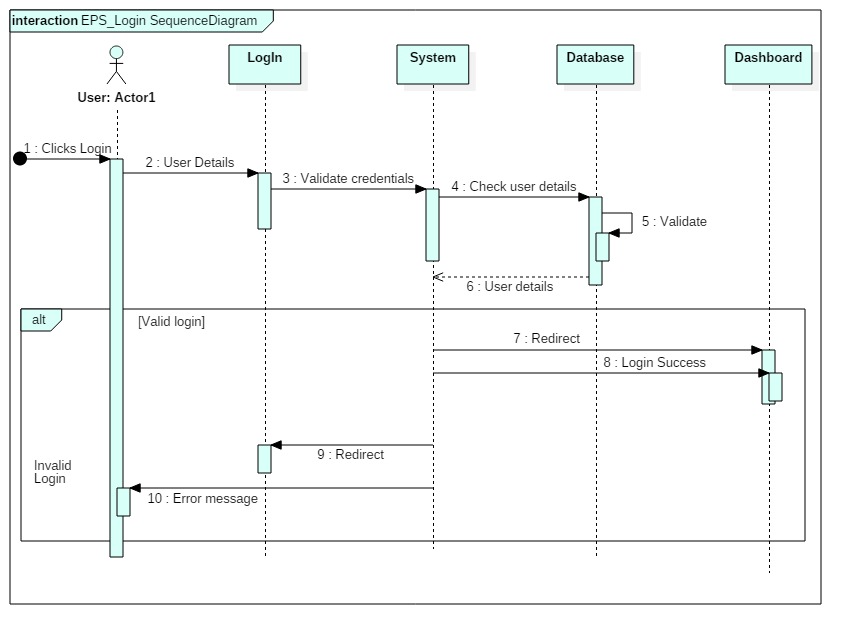
****

The sequence diagram of employee payroll system is illustrated on **Star UML** that is very easy to use and express the functionalities of the system showing each and every operation as a work-flow. It defines the work flow of system performed by either admin or employee.

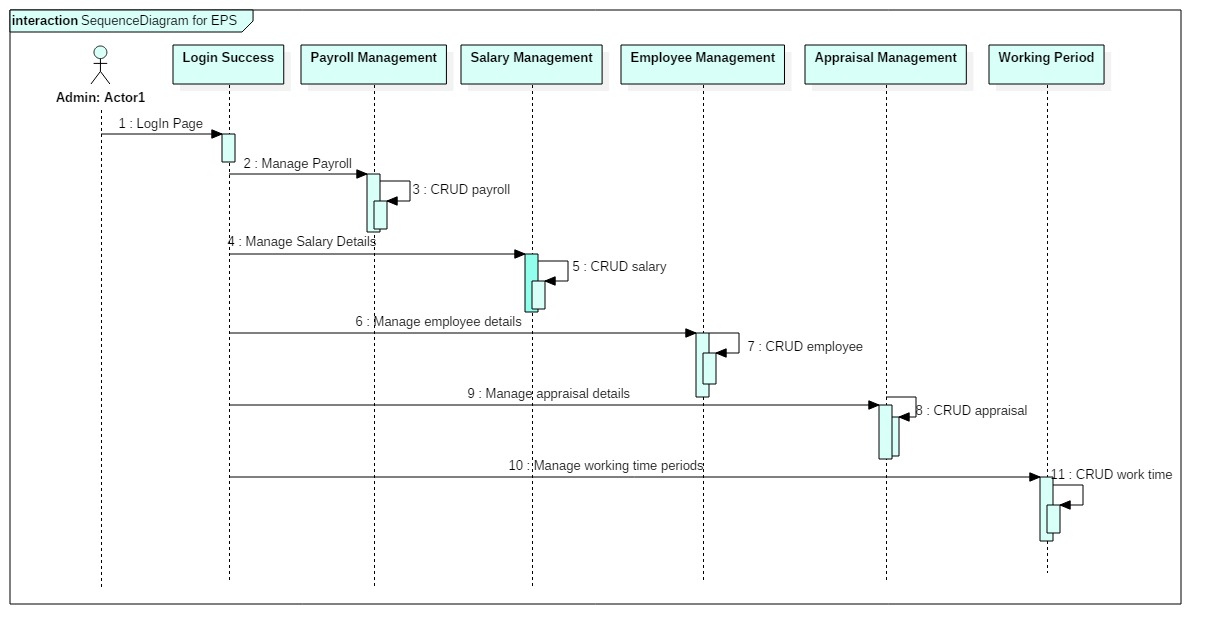
**Sequence diagram and descriptions**

****

In this sequence diagram, the registration process of employee payroll system is represented where valid inputs are only allowed and invalid displays error. During registration, valid details needs to be stored and with final registration success the system provides username and password to the user. Furthermore, the username and password are applied during login the system.

****

In this sequence diagram, the login system of the EPS is described clearly where user enters their respective username and password which will be validated to data stored on database and after correct validation the respective dashboard gets open. In case of incorrect validations, the system shows an error message. This diagram contains actors, lifelines, messages and self-message.

****

In this above sequence diagram, the complete employee payroll system is displayed where actors, lifeline and messages are clearly described which are easy to read. This diagram shows the payroll system where employees, salary, payroll, appraisals and working period are figured out with having CRUD operations. This completes the payroll system.

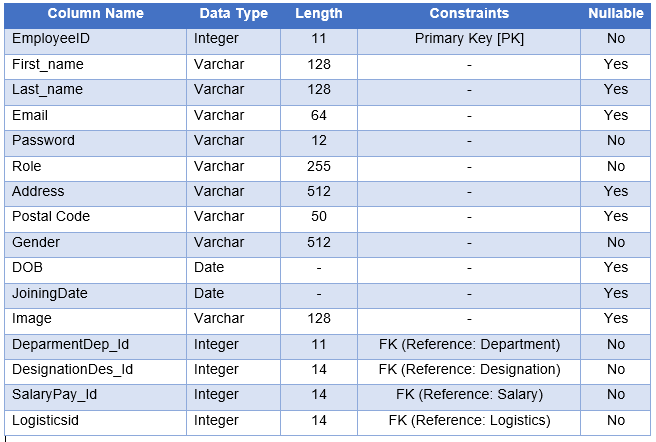
## **3.4 Database Modeling**

A database modeling is the modeling of a database diagram that represents a **logical structure** of database along with their relationships. The structure on the database determines the way of storing and accessing data.

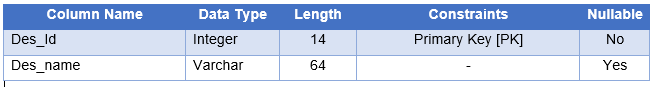
### **3.4.1 Data-Dictionary**

Data-Dictionary is also called data repository or system catalog which is an integral part of database that holds information about the database which is **metadata.** It is important part of DBMS **(Database Management System)** which contains actual database descriptions used by DBMS. The data dictionary is used to retain and handle database and control entrance which consist contents like column name, data type, length, constraints, nullable and many more.

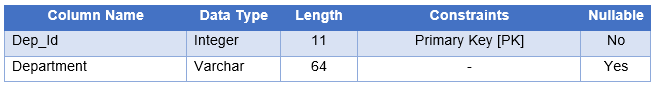
**Table 1: Employee**



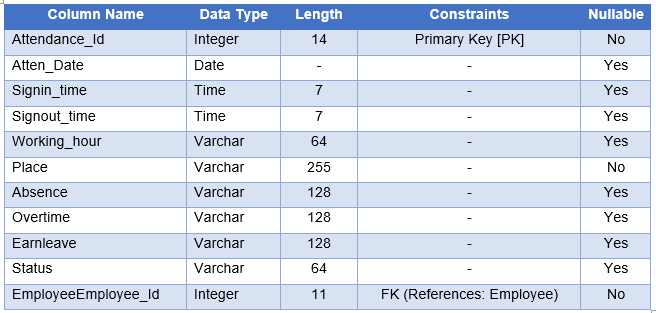
**Table 2: Designation**

****

**Table 3: Department**

****

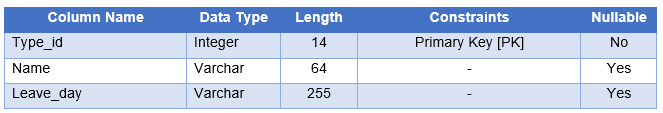
**Table 4: Attendance**

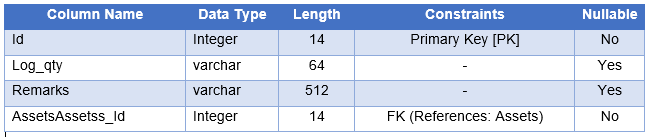
****

**Table 5: Assets**

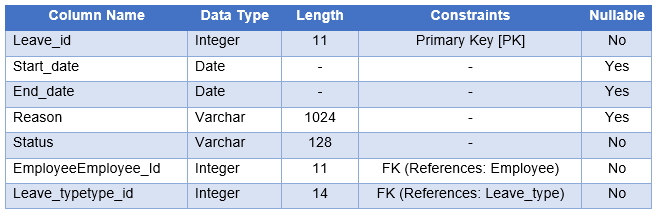
****

**Table 6: Leave Type**

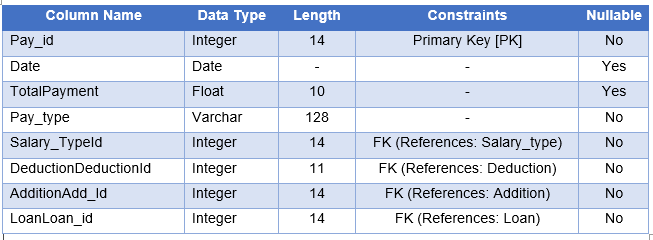
****

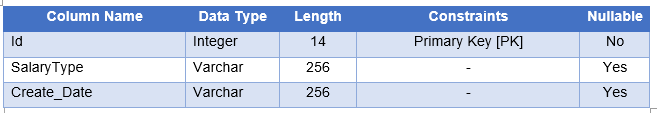
**Table 7: Logistics**

**Table 8: Leave**

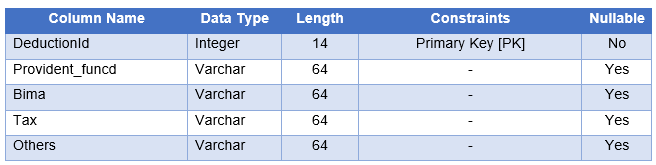
****

**Table 9: Salary**

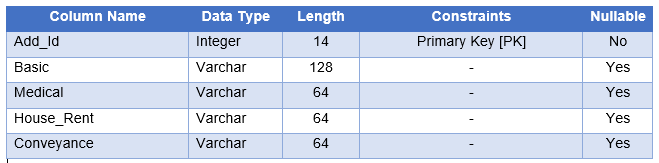
****

**Table 10: Salary\_type**

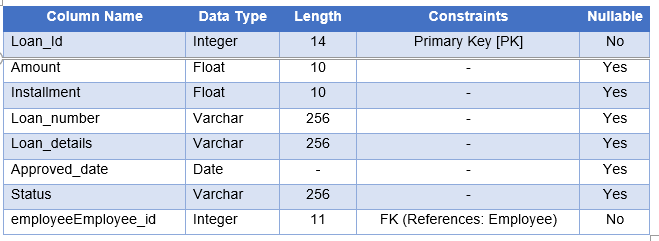
**Table 11: Deduction**

****

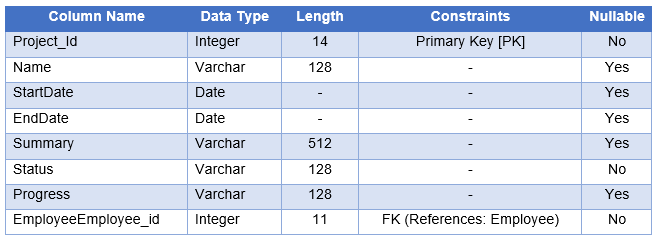
**Table 12: Addition**

****

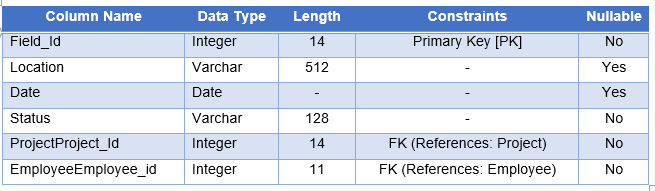
**Table 13: Loan**

****

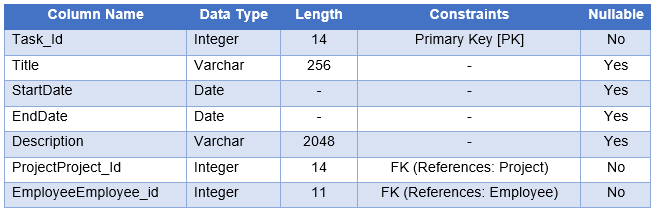
**Table 14: Project**

****

**Table 15: Field\_visit**

****

**Table 16: Task**

****

The above listed tables are the data-dictionary of Employee Payroll System which represents all entities with their column names, data types, lengths, constraints and nullable.

### **3.4.2 ER Diagram [Entity Relationship Diagram]**

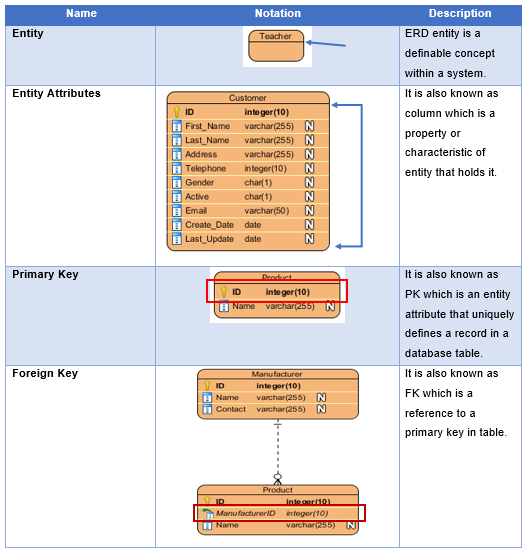
ER diagram, which is also known as entity relationship diagram or ERD is a structural diagram that are used for designing database of the system. ER diagram consists of various symbols and connectors within system scope and inter-relationship among entities.

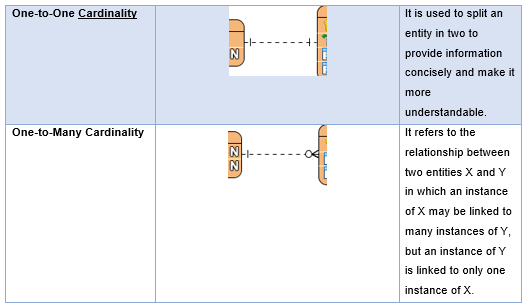
**Reason behind choosing Entity Relationship Diagram as Database modeling**

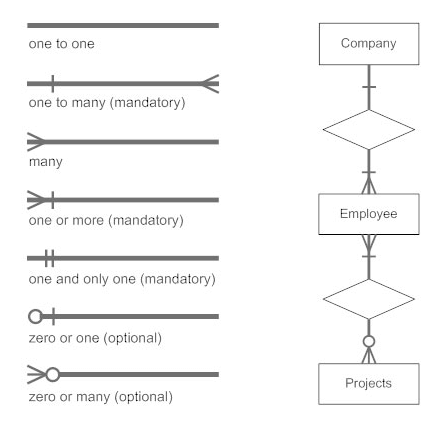
There are many benefits of ER diagram which supports for implementation phase. It offers visual presentation of the system and results in effective flow of communication and information. It is very easy to understand by individuals as the diagram is made in a simple manner. Hence, ER diagram makes the flow of data very efficient and increase the flexibility of the system.

**Notations in Entity Relationship Diagram**

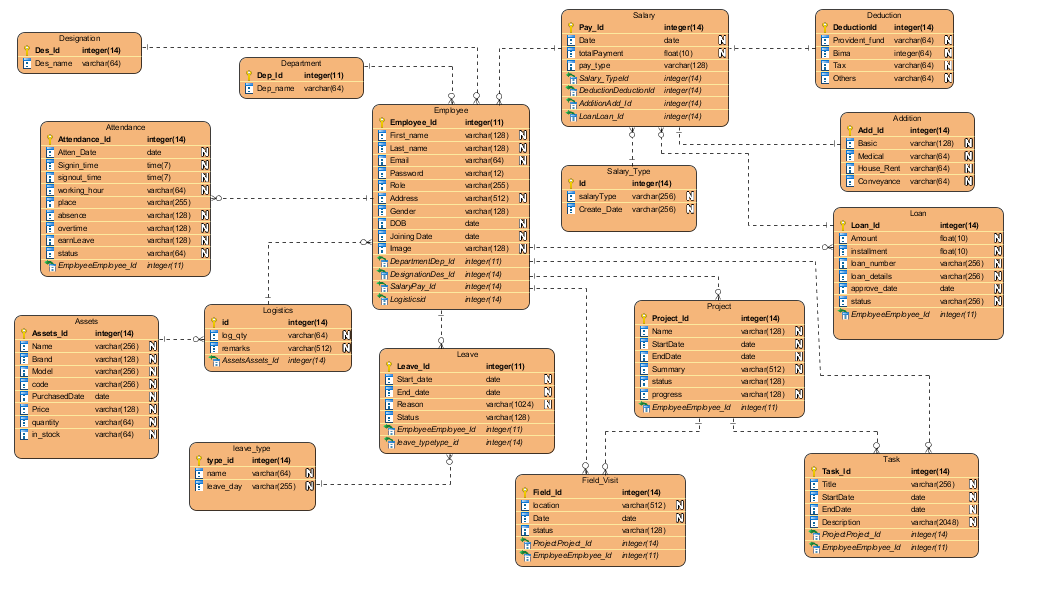
An ER diagram consists of attributes, entities and relationships which are as listed in a tabular format.

****

****

****

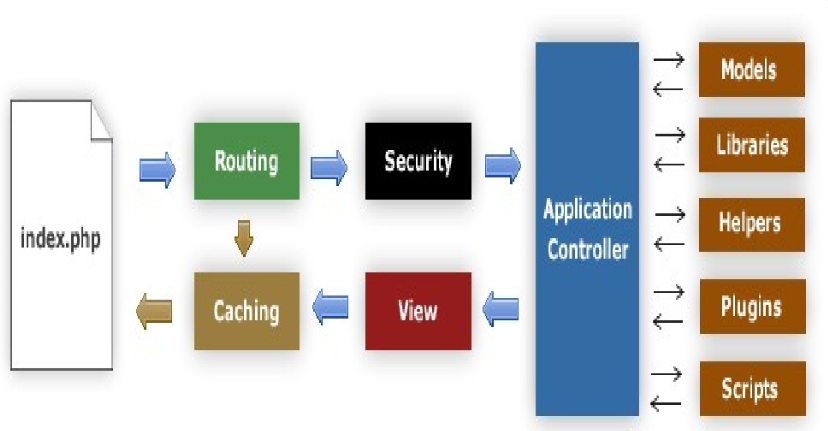
**Entity Relationship Diagram and Description**

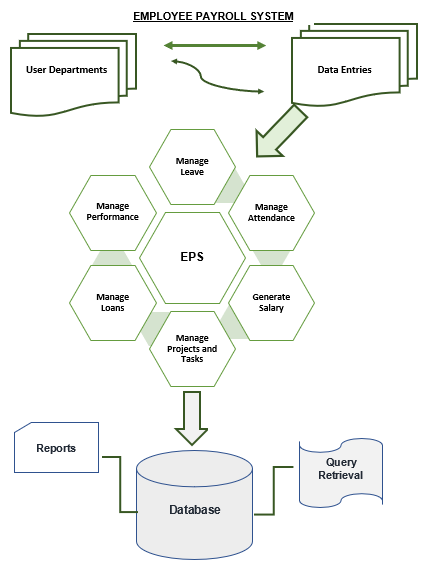
****

The above ER diagram depicts the complete system in an easy way which can be understand by each and every individual. The diagram shows its seven entities in employee payroll system with respective attributes with primary key and foreign key constraints. They are linked by relationship as one-to-many cardinality relationship. The diagram is drawn in **Visual Paradigm** which is itself an easy UML tool.

## **3.5 Working Architecture of Project**

It is also known as architecture modeling where I have decided to go through layered architecture that supports incremental development of system. The layer gets developed continuously and the services are made available to the users timely. As the layer gets added or changed, the additional features are added step wise where localize machine depends upon inner layer. This tends to provide multi-platform for users of an application system. The machine-dependent which is inner layer are tend to be re-implemented to take account of facilities of the database or an operating system.



****

The diagram above represents the working architecture of employee payroll system with database storage where, the framework is represented in a work flow of data.

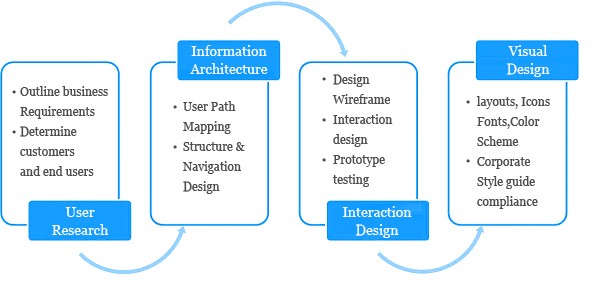
## **3.6 User Interface Design**

User Interface Design is also known as UI Design which is the process of creating **interfaces** in software that focuses mainly upon the styles and positions of images or formats. It is basically a visual element of a particular product or a system that shows the actual presentation and interactivity of the system.

Similarly, I have chosen **Balsamiq** for prototyping purpose or for creating user-interface design.

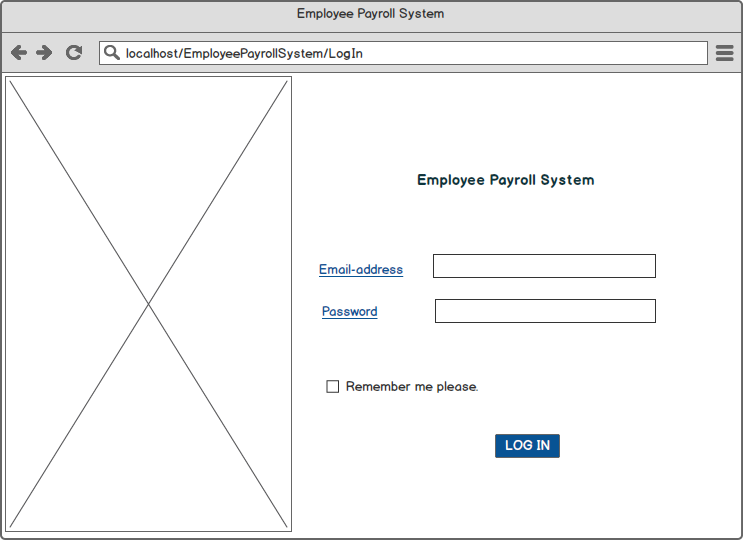
**Reason behind choosing User Interface Design during Design Phase**

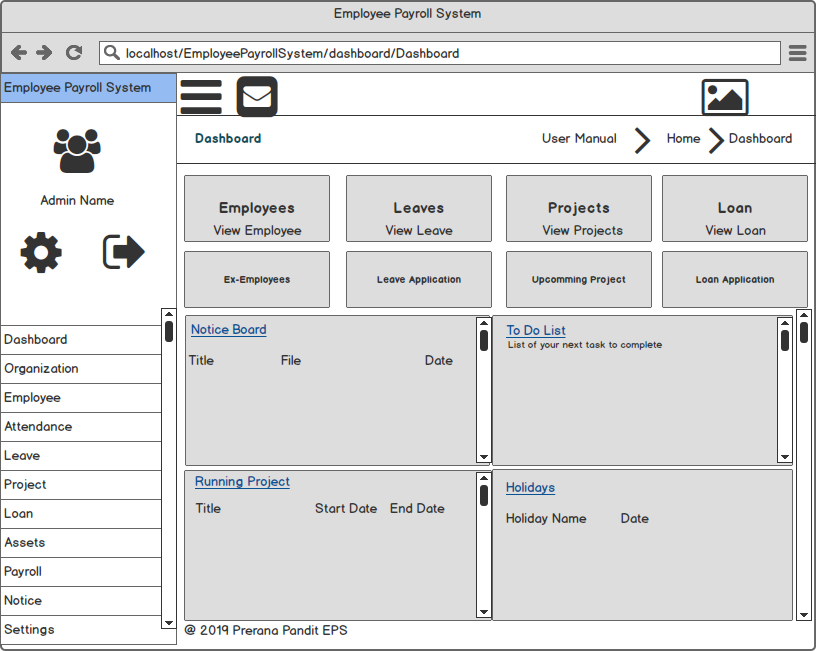
Basically, UI design is overall about experience that combines technical and creative aspects of system. It helps in enhancing sales and optimizing the user experience that helps in reducing initial development costs and further iterations. During User Interface design phase, there will be more interaction with customers or users that builds good communication and better understanding about the system.

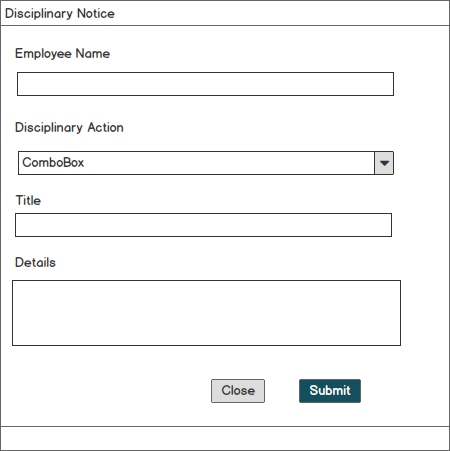
****

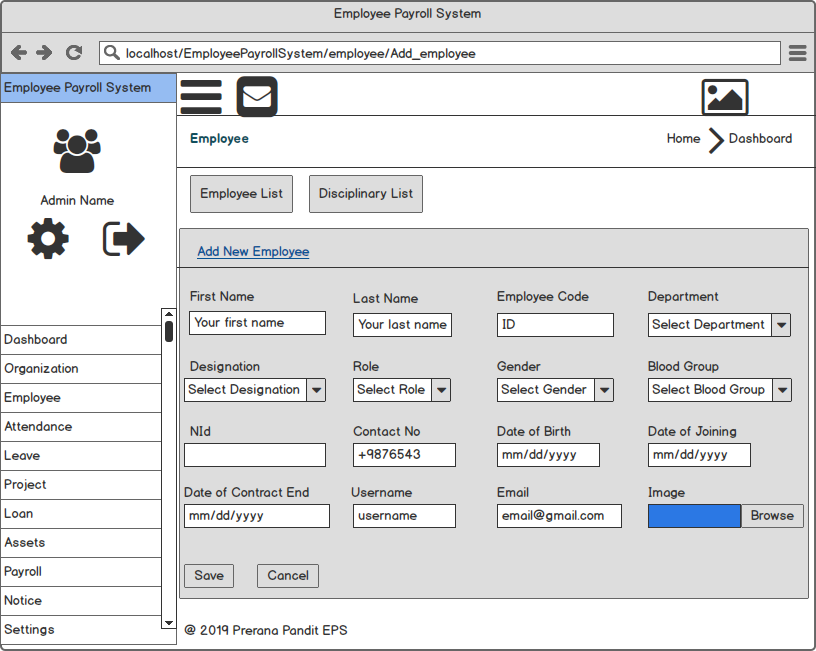
**UI RULES**

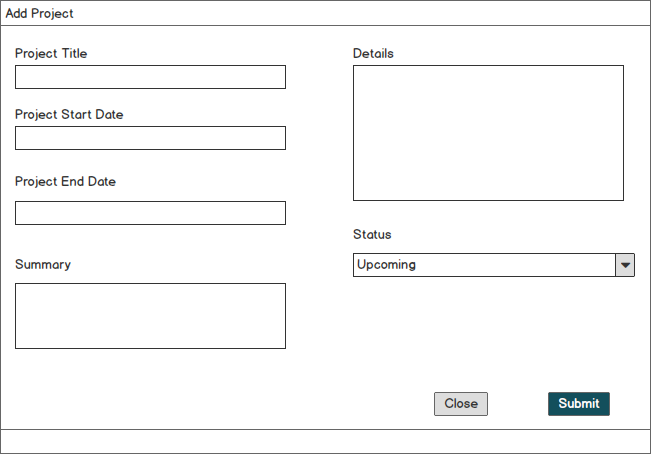
### **Prototypes done on Balsamiq**

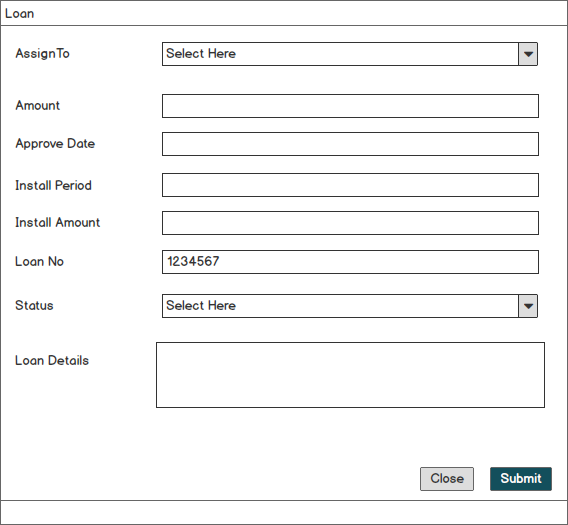
****

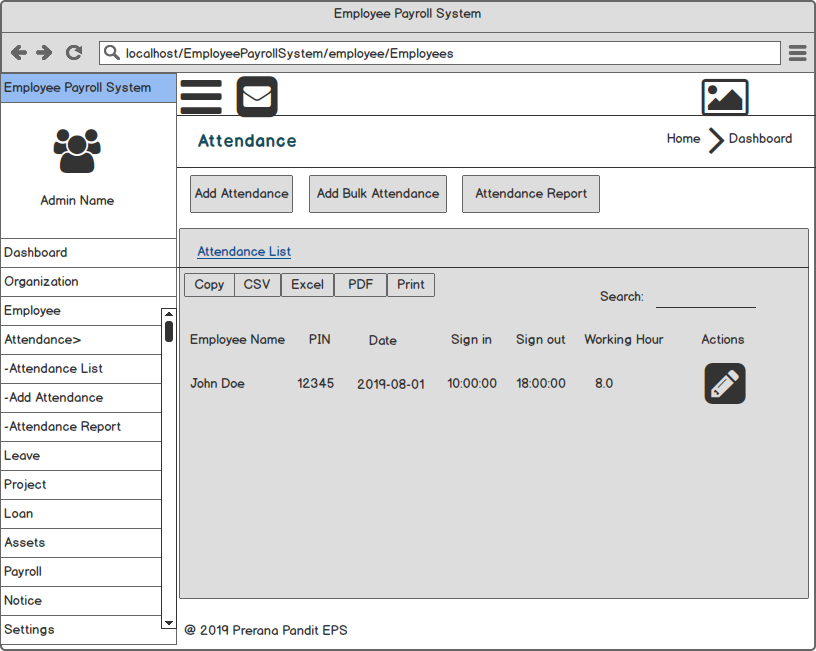
****

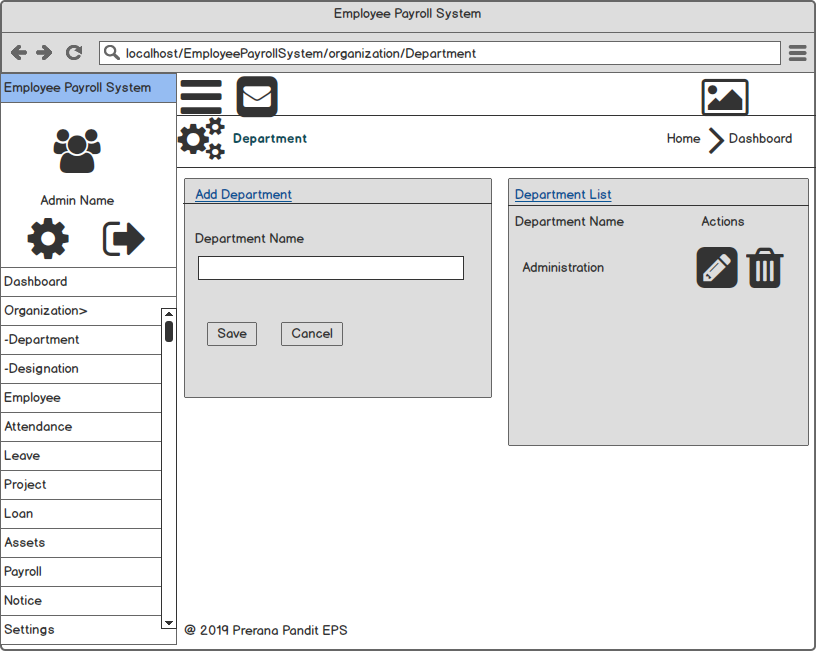
****

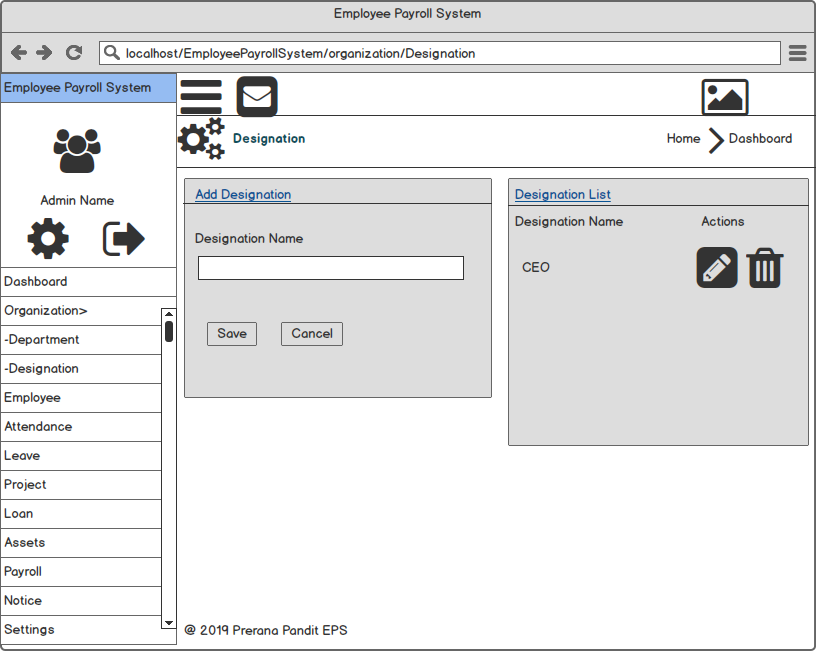
****

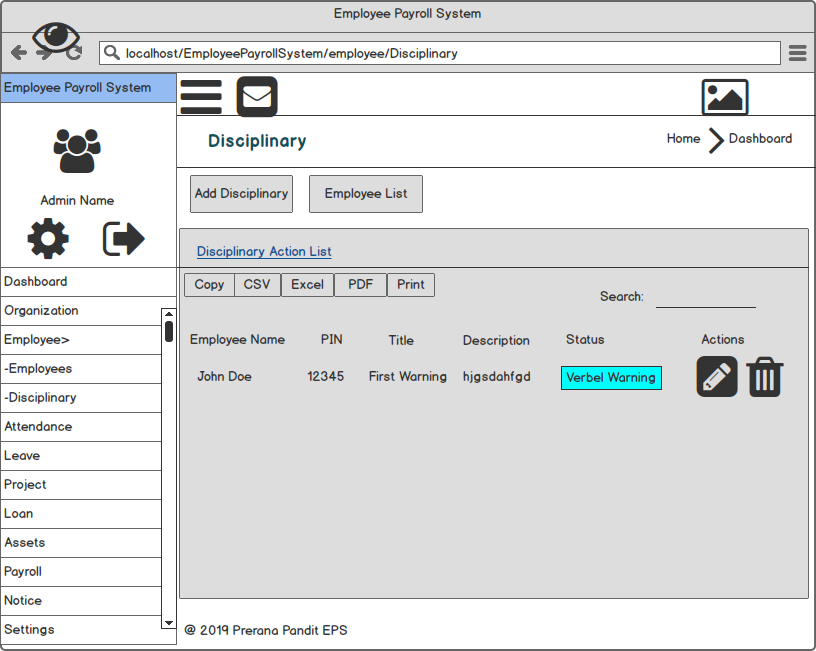
****

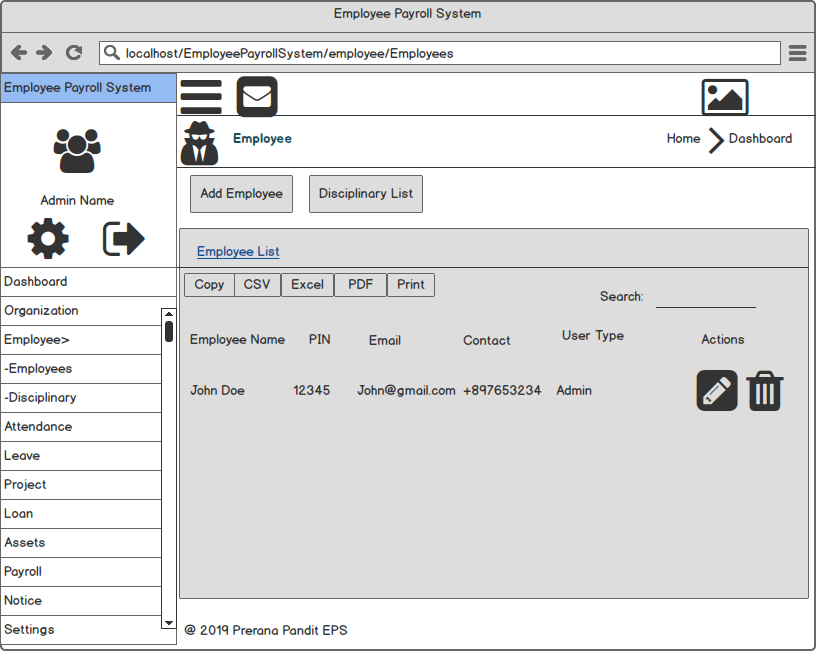
****

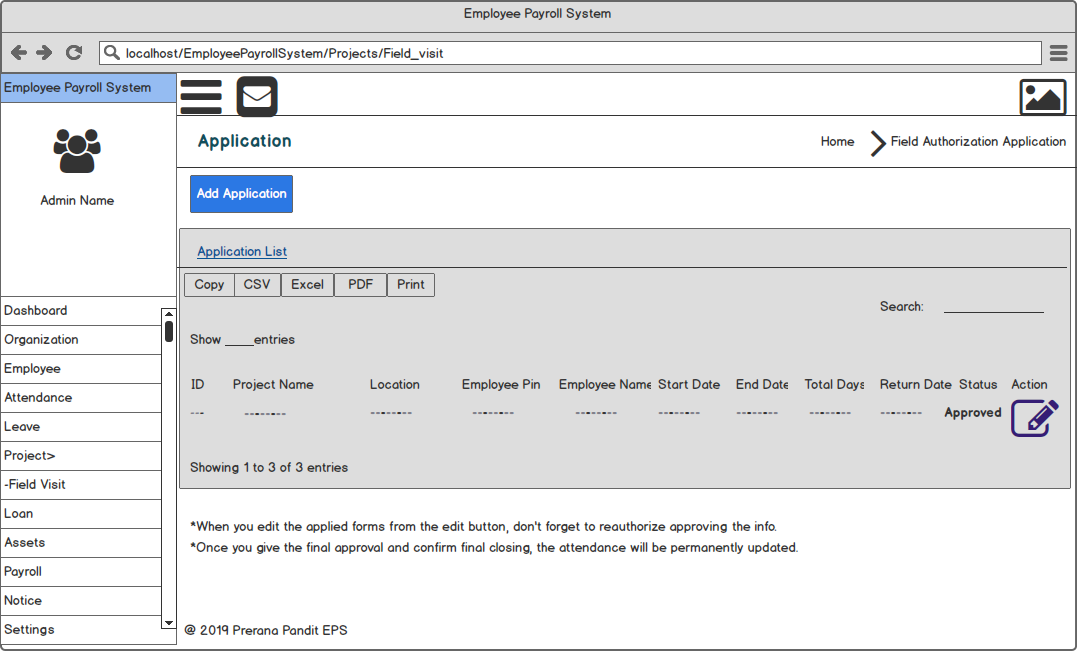
****

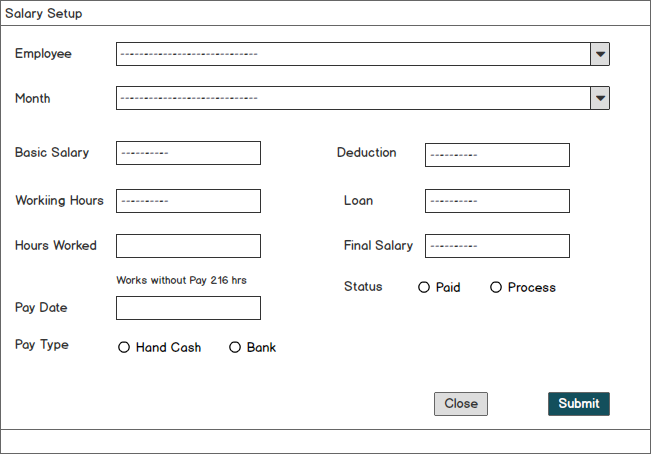
****

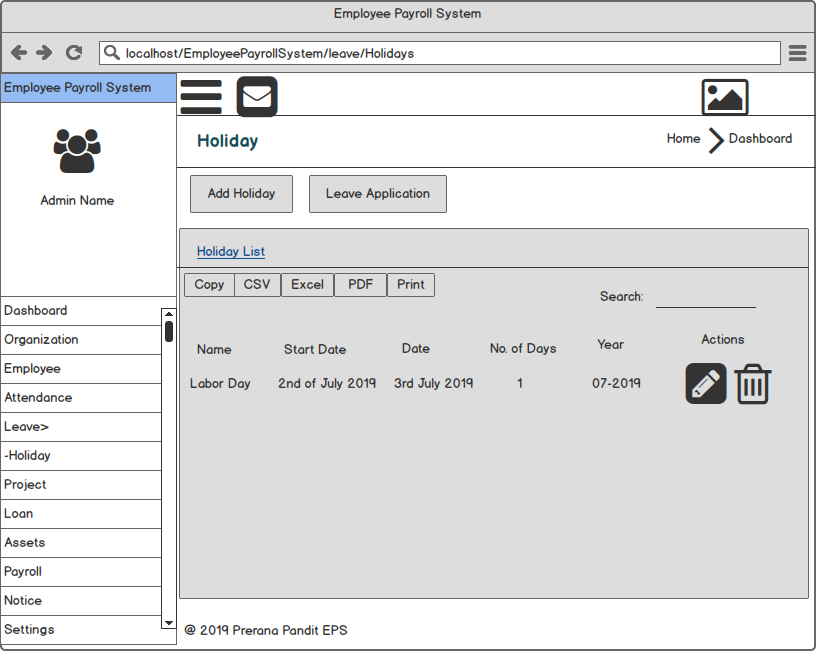
****

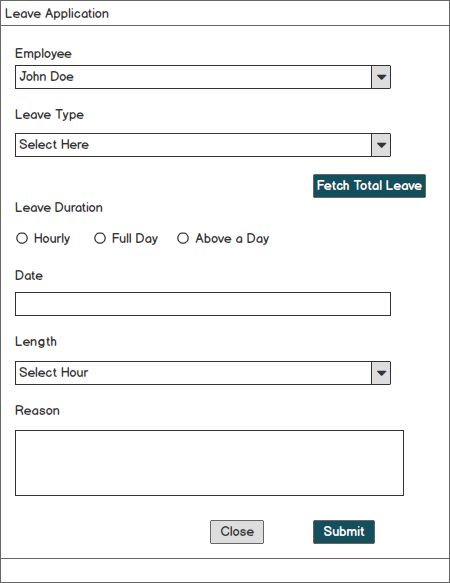
****

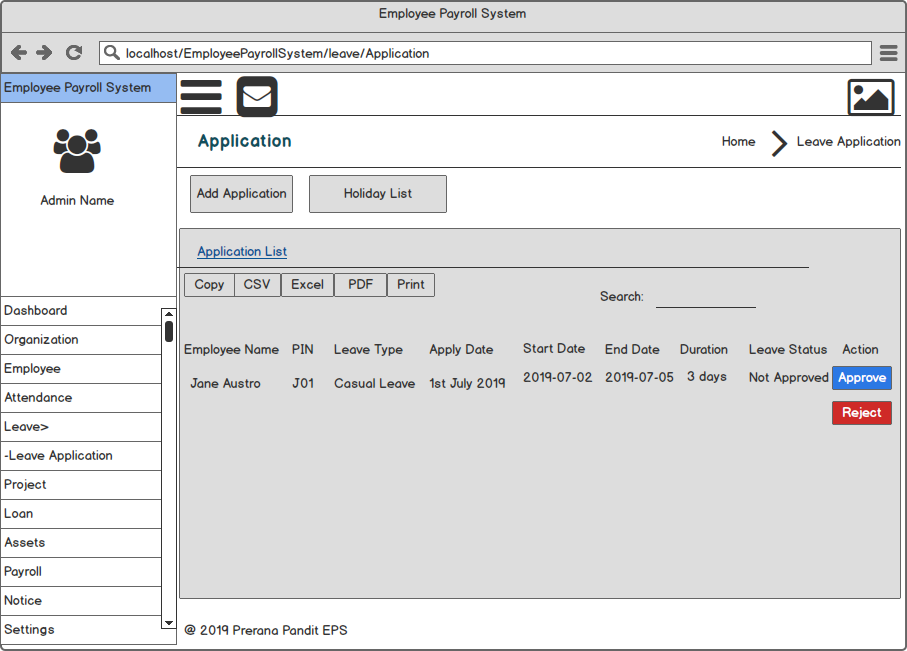
****

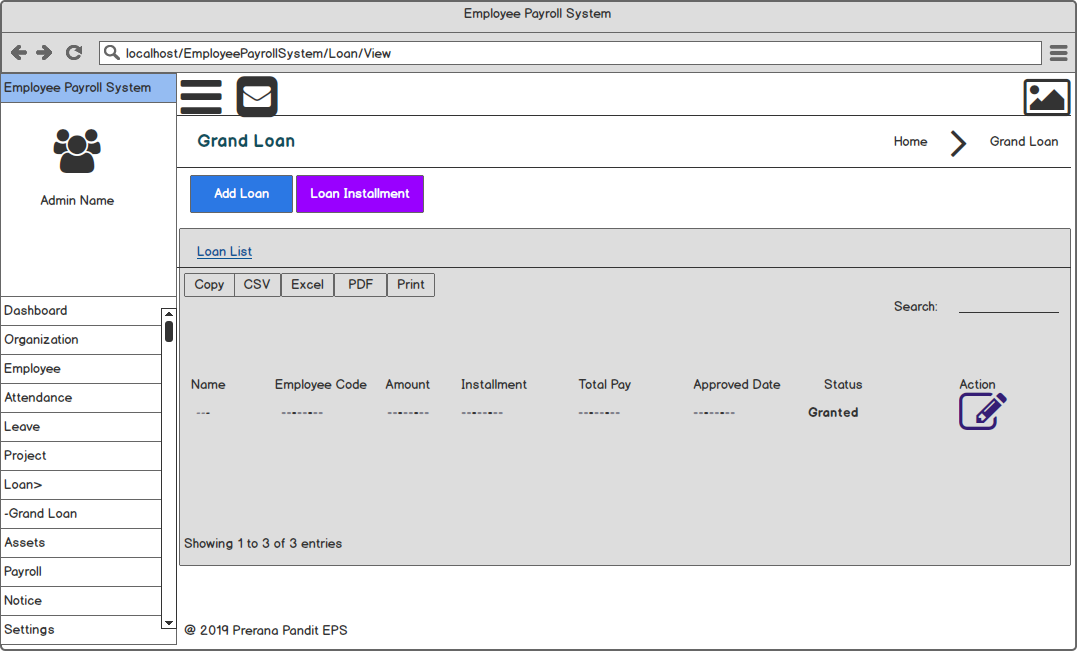
****

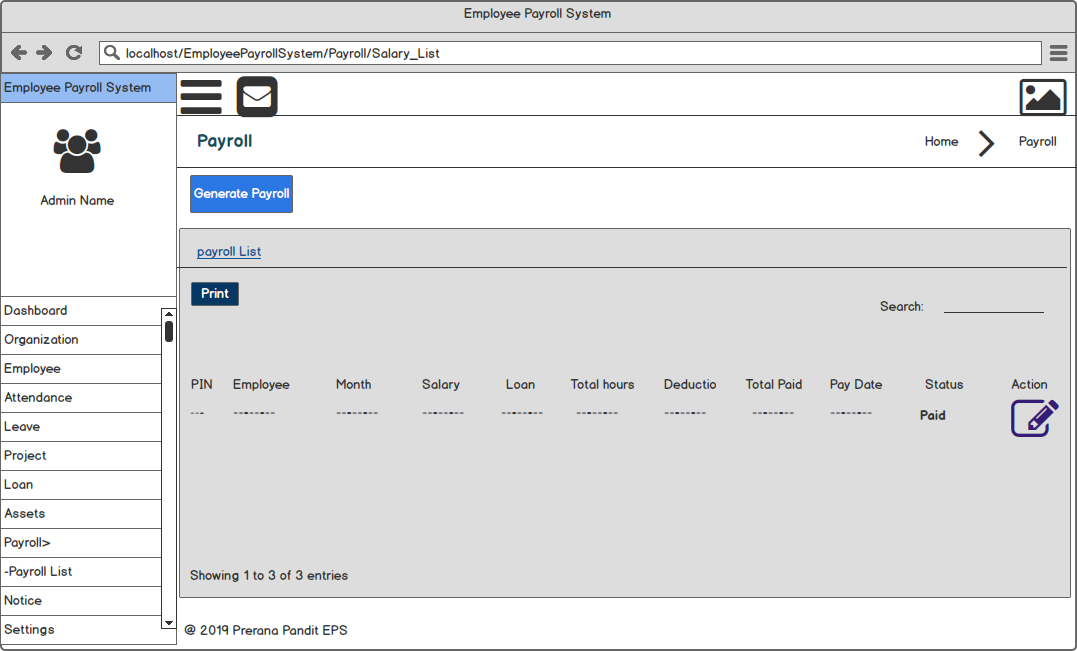
****

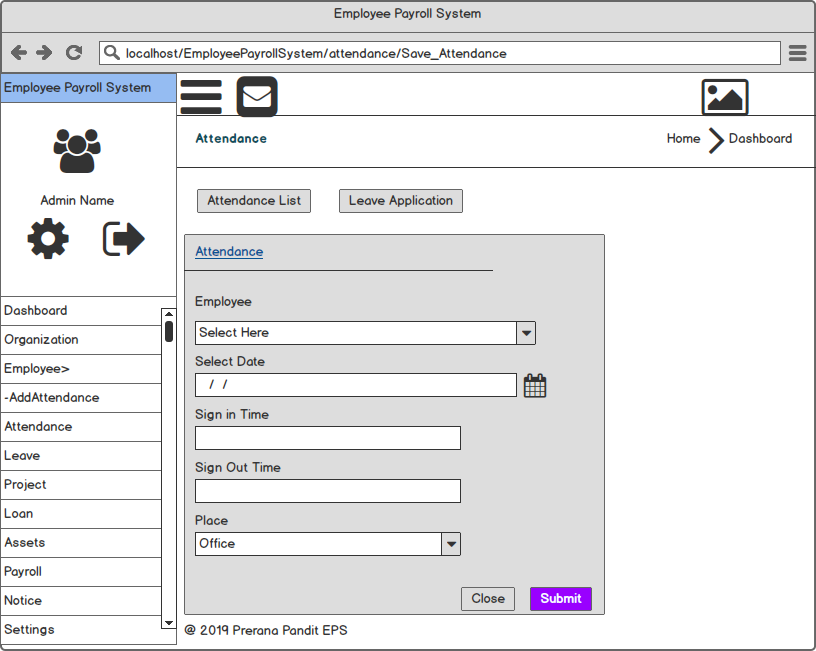
****

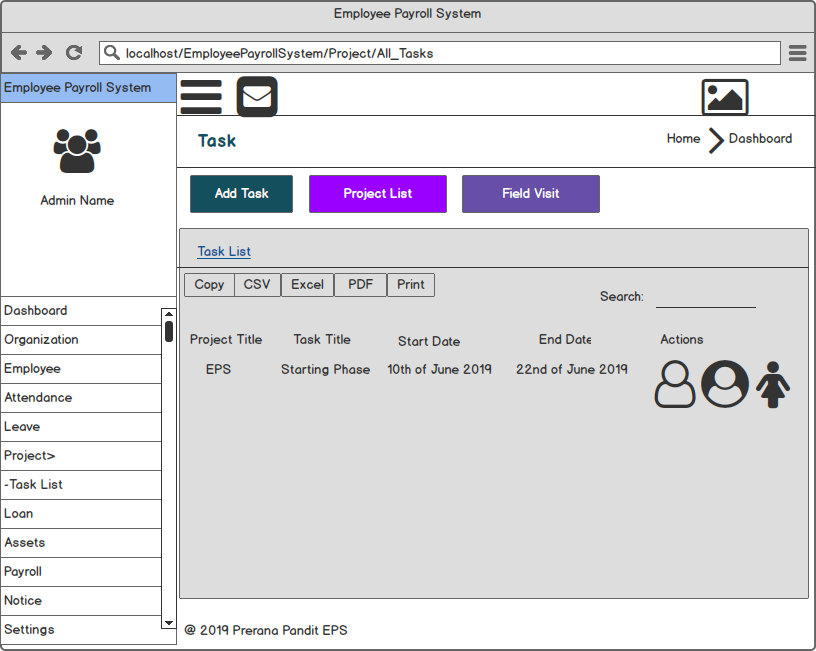
****

****

****

****

****

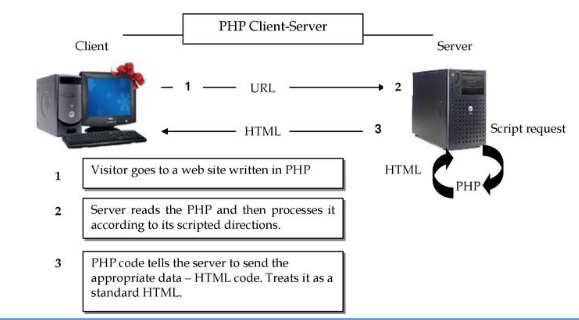
****

# **Chapter 4: Implementation**

After the completion of system design documentation, actual coding begins where work is divided into modules. The implementation is the longest phase where it takes longer time and duration for the completion of coding where developers are the main focus. It is also known as software installation on our machines which requires post-installation configurations. The final output of implementation phase is tested for adaptability and portability where issues are solved.

## **4.1 Programming Language**

The proposed programming language to develop Employee Payroll system is **PHP**. PHP or ‘**Hypertext Preprocessor**’ is an open-source product that is used for file processing, form handling and database access. It is a server-side scripting language which are executed on the server. The database chosen for the system is **MySQL** which is free to use and download. It can run on different platform which has a file extension as .php, .php3, .phtml that are returned to the browser as a plain HTML.



## **4.2 Framework**

As PHP is an applied programming language, the framework utilized for development of the employee payroll system is **CodeIgniter (CI).** It is a toolkit available open source which is an application development framework built by **EllisLab**.

**Reason behind choosing PHP with CodeIgniter framework:**

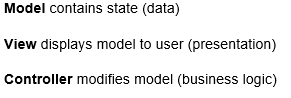
Most importantly, it is free to use and follows **MVC** Model-View-Controller Pattern which separates logical and presentation parts. It is light-weighted as it requires small library that are added upon dynamic request based upon needs. It generates search engine friendly URLs which also appears in full packet libraries that enables web needed tasks as database, form validations, manipulating images, sending email, and many more.

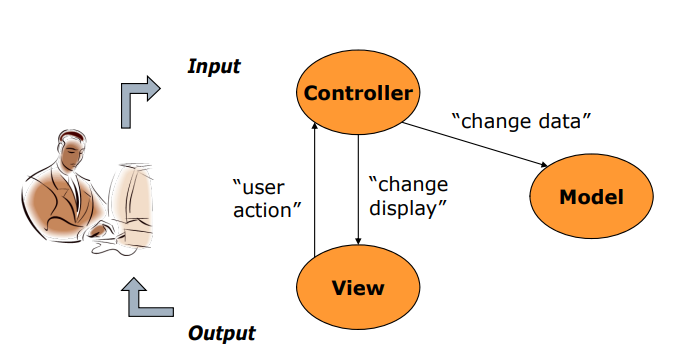
## **4.3 Design Pattern**

Design patterns are the representation pattern that systematically names, explains and motivates a design addressing a recurring design issues in object-oriented system. It explains about the issues and the solution when to be applied with its consequences.



As a result, I have chosen **MVC Design Pattern** which is an architectural pattern that relates to the user interface layer of the system. It is **Model View Controller** design pattern specifying the system consisting a data model, presentation information and control information that separates into various objects.





MVC Pattern

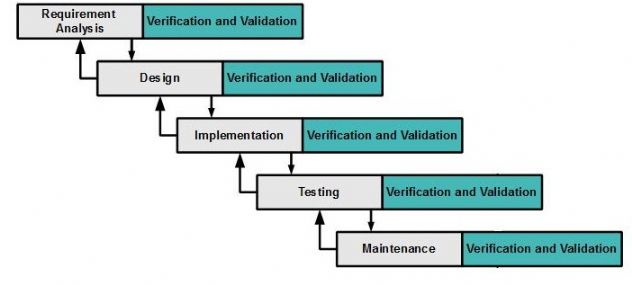
**Reason behind choosing MVC Pattern:**

MVC is a faster development process supporting rapid and parallel development. It has an ability to create multiple views for model where code duplication is limited as it separates data and business logic from view. It supports asynchronous techniques that helps to load application very fast. Any kind of modification does not affect the entire model and returns the data without formatting which indicates the interface uses.

So, the MVC design pattern has been chosen during implementation phase of software development life cycle.

## **4.3 Development Methodology**

The project has been developed using **modified waterfall model** in software development phase that required lesser need for revisions and documentation. Unlike traditional waterfall model, it allows overlapping of phases that shows there may occur many tasks concurrently. This allows overlapping of stages to incorporate changes on the basis of requirements during development process. This model can easily lead back to other phases whenever required as shown in the figure below:



**Requirement Analysis:**

It focuses on gathering of information and requirements required by the user that describes overall functionality of the system which can be done through interviews, questionnaires, focus groups, etc.

**Design:**

It focuses on designing the output of the software which is based upon conceptual framework on the basis of model chosen.

**Implementation:**

It is the phase of execution of plan, model, design, standard, specification, algorithm and realization of an application. It covers overall project requirements and specification with code for the system.

**Testing:**

This phase checks whether the system is working properly or not or whether it meets the requirements or not. This results to changes if required.

**Maintenance:**

It is never ending phase of maintenance in a system during the occurrence of any issues or faults. The upgrades or addition of any level can be performed in this stage with the client’s verification and validation.

**Reason behind choosing modified waterfall model**

The previous phases can be easily revisited during any problem-solving process. It helps in increment of flexibility. It helps in verifying patches and forks updates to next version. Having specific deliverables on each phase, it works well for any kind of projects where requirements are analyzed properly.

## **4.4 Development Tools**

During implementation phase, the development tool that has been applied are as listed below:

**Source Code Editor and Debugger:** Visual Studio Code

**Backup:** Git

**Tested Browser:** Edge, Google Chrome

**Composer:** Apackage manager of PHP

**Web server:** Apache server

**Database server:** MySQL

## **Notes:** The code snippets and user interface designs are kept in **Appendix** section.

# **Chapter 5: Testing**

After the completion of implementation phase, the employee payroll system is deployed to testing environment. The functionality of the entire system is tested one by one which is done for verification to meet overall requirements. The testing phase continues till the software becomes stable, bug-free and work according the system needs.

It is evaluation of the software against requirements gathered from system specifications. It comprises of verification and validation.

## **Necessity of Testing**

Testing is an important part of software development that helps to verify the software requirements. It helps to identify the defects and solve them before software deployment. Effective testing is the demonstration of functions appearing to be working according to behavioral and performance requirements and specifications. It helps to verify the proper interaction or integration of each component of the system. It indicates the software reliability and indication of software quality as a data collection. It is essential to ensure the quality and effective performance of the employee payroll system.

## **Types of Testing**

There are various types of testing based upon two approaches as **functionality** testing and **implementation** testing. Some of the types of testing are Black box testing, Unit testing, Integration testing, Regression testing, Beta testing, Whitebox testing, etc.

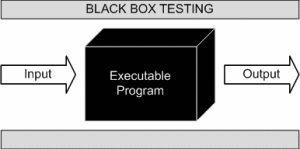
So, among those testing, I have chosen Blackbox testing and Unit testing which are described below.

### **Black Box Testing**

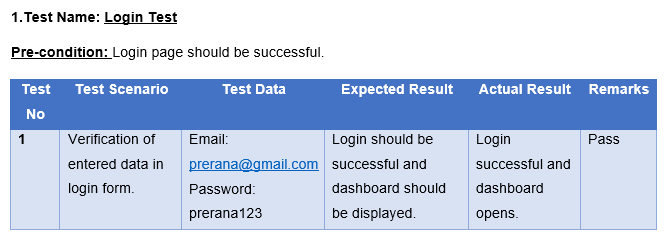
It is a method of testing that examines the functionality of the employee payroll system without peering to its internal workings or structures. It attempts to find the errors in external behavior of the code. It is also known as **behavioral testing**. It does not require knowledge of interior workings. It helps to find the missing functions, performance problems, concurrency and timing errors, usability problems, and many more.

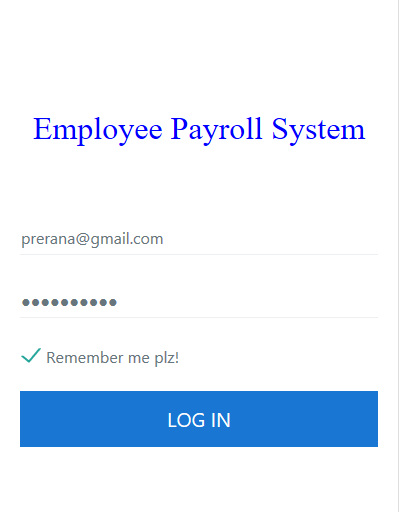
**Reason behind choosing black box testing**

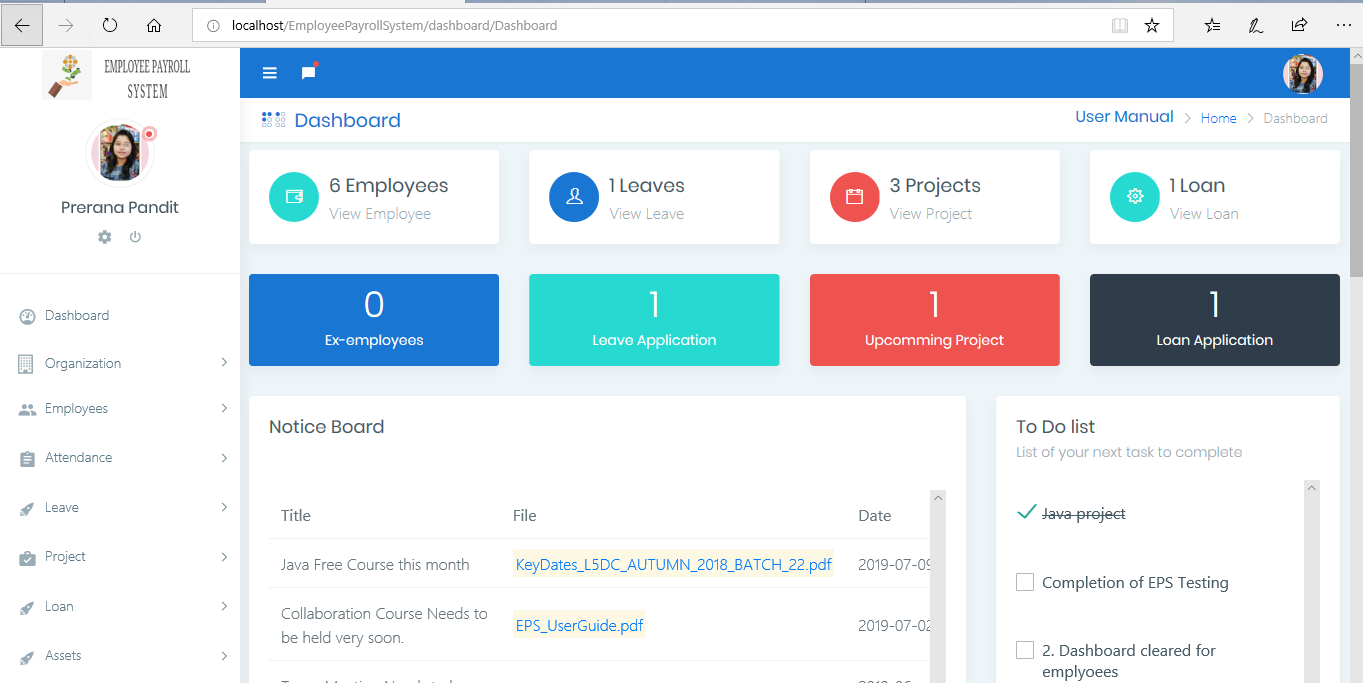
The testing is more effective on larger units of code and requires no knowledge of implementation, including programming languages. The tester and developer are independent of each other for doing black box testing. The tests are done from a user point of view. It helps to expose the ambiguities in the specifications.

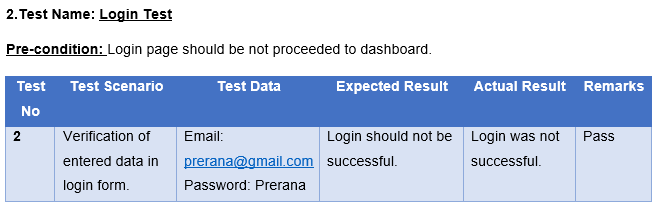


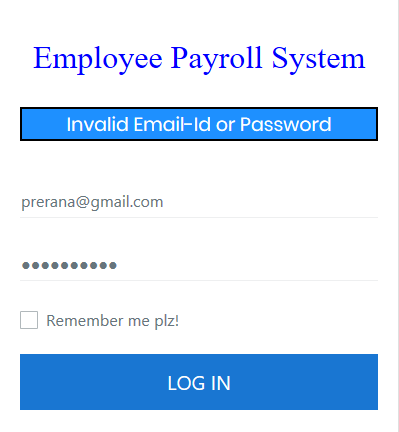
**Black box testing performed:**

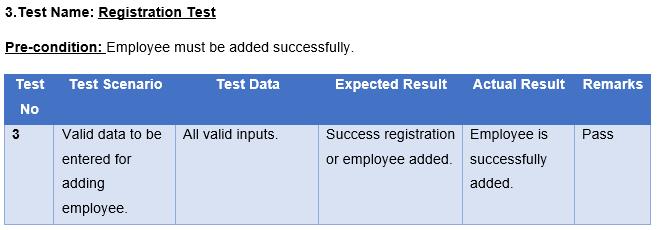


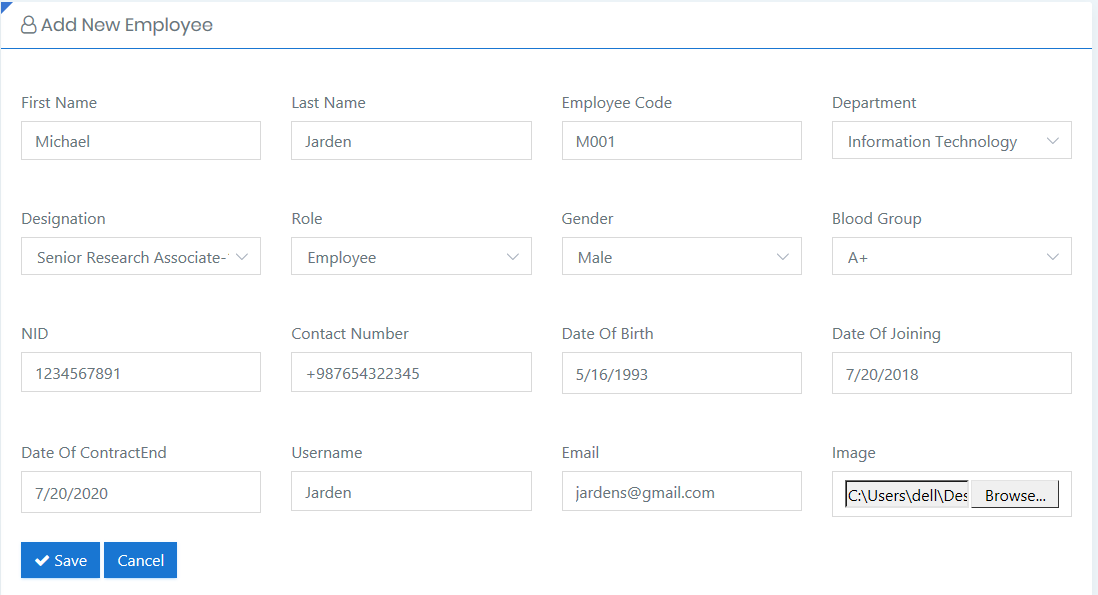


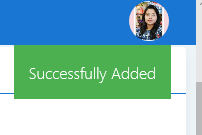


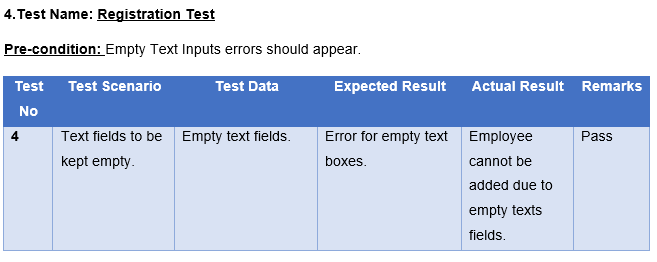


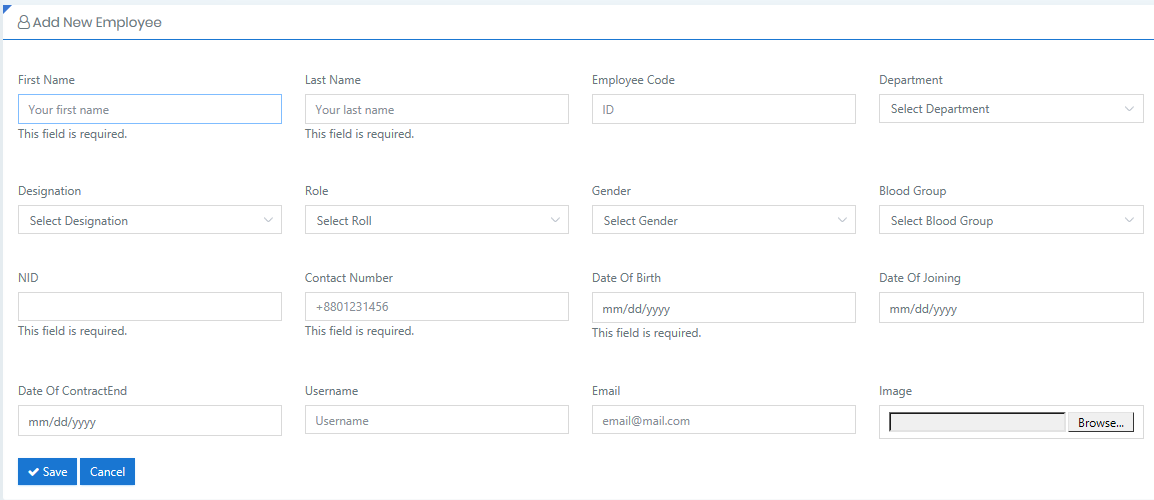


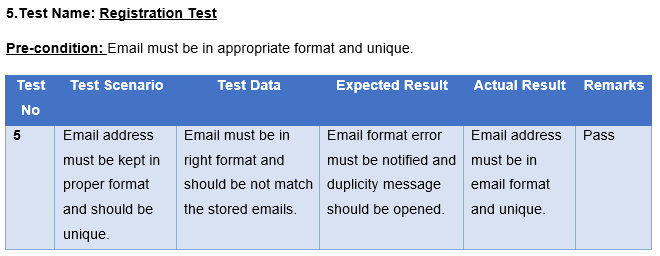
****

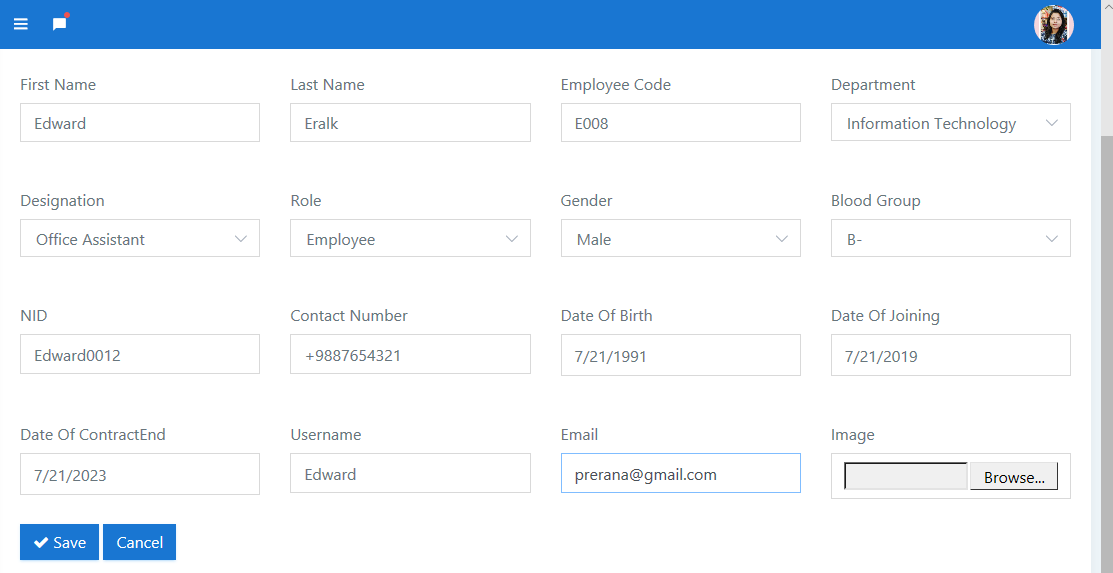
****

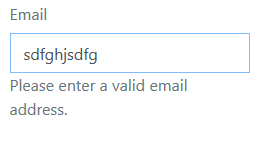
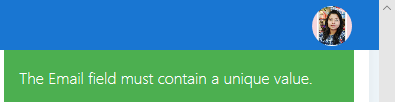
****

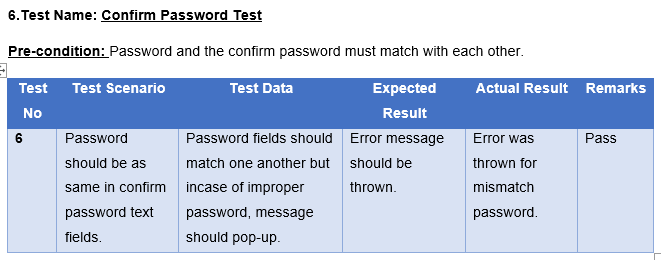


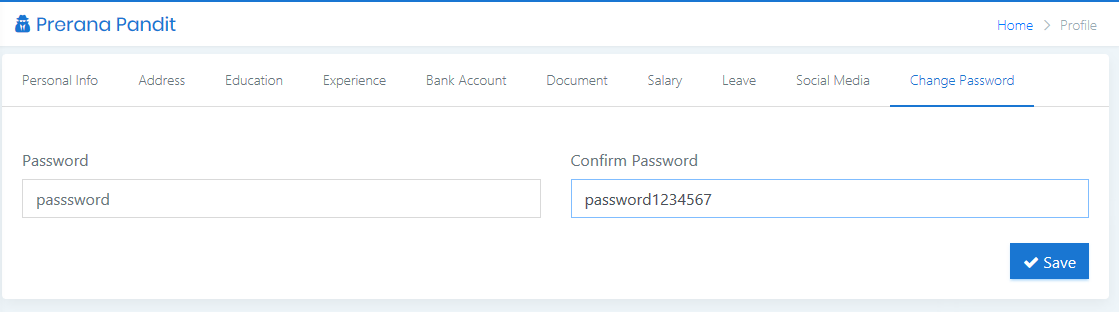


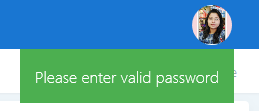


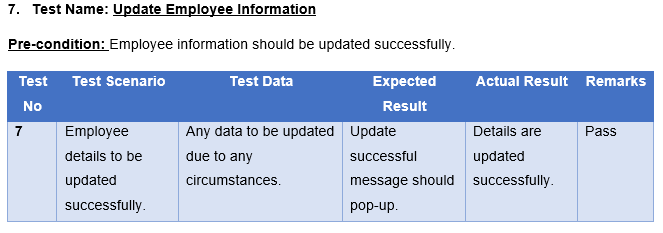


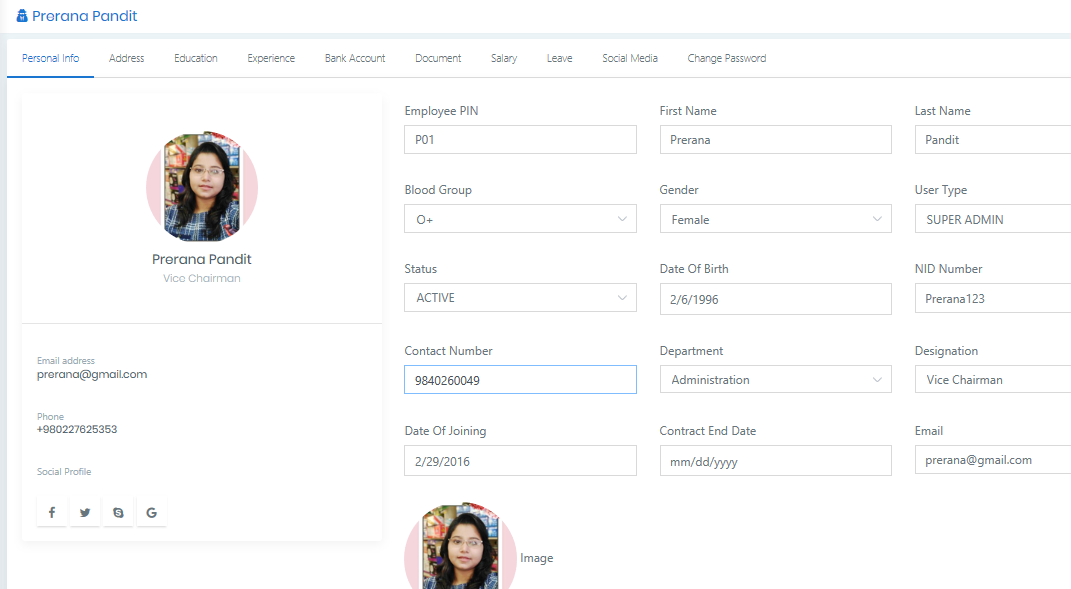




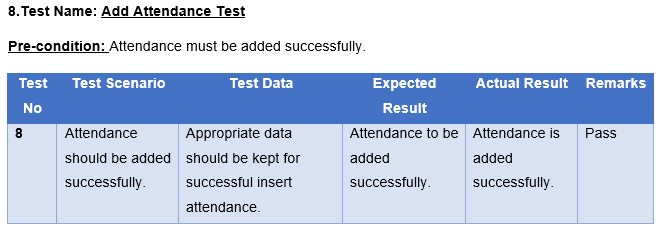


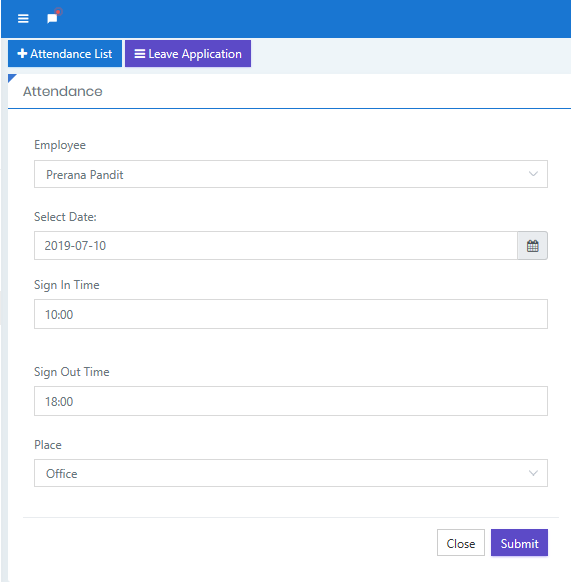


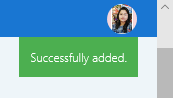


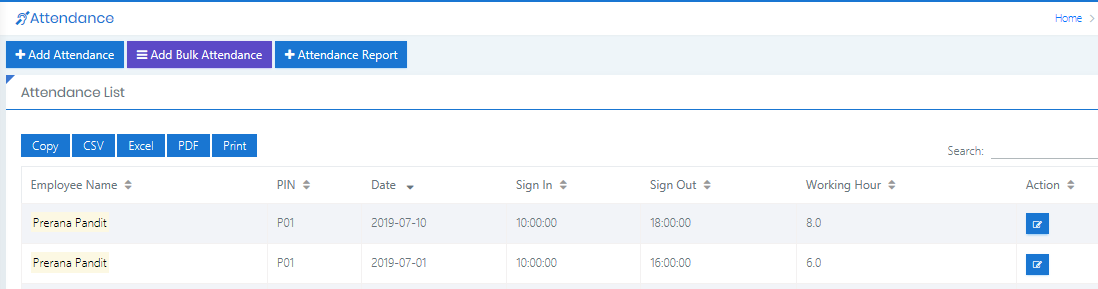


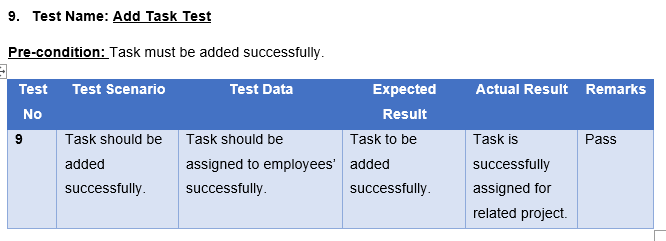


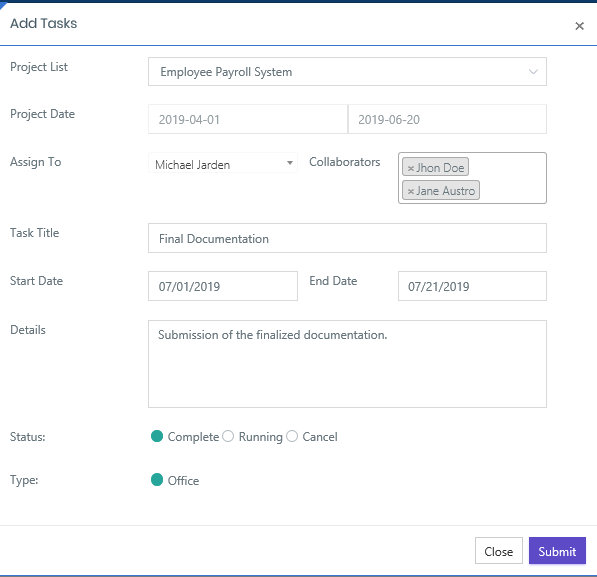


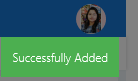


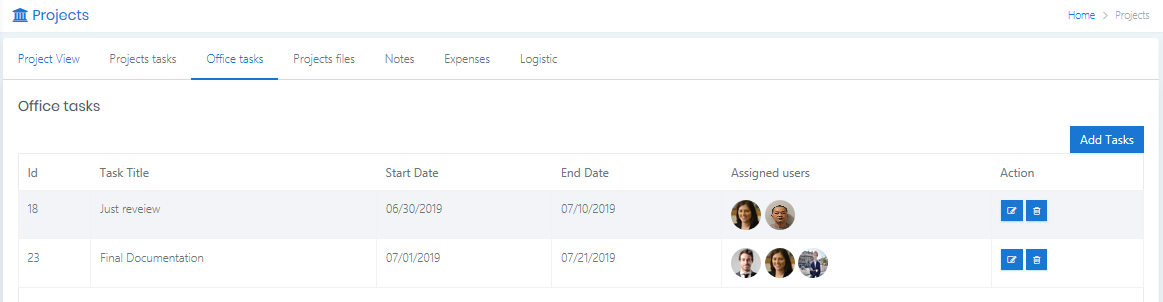


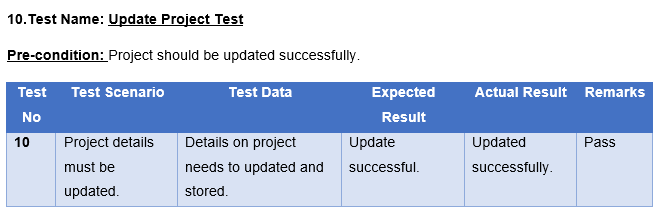


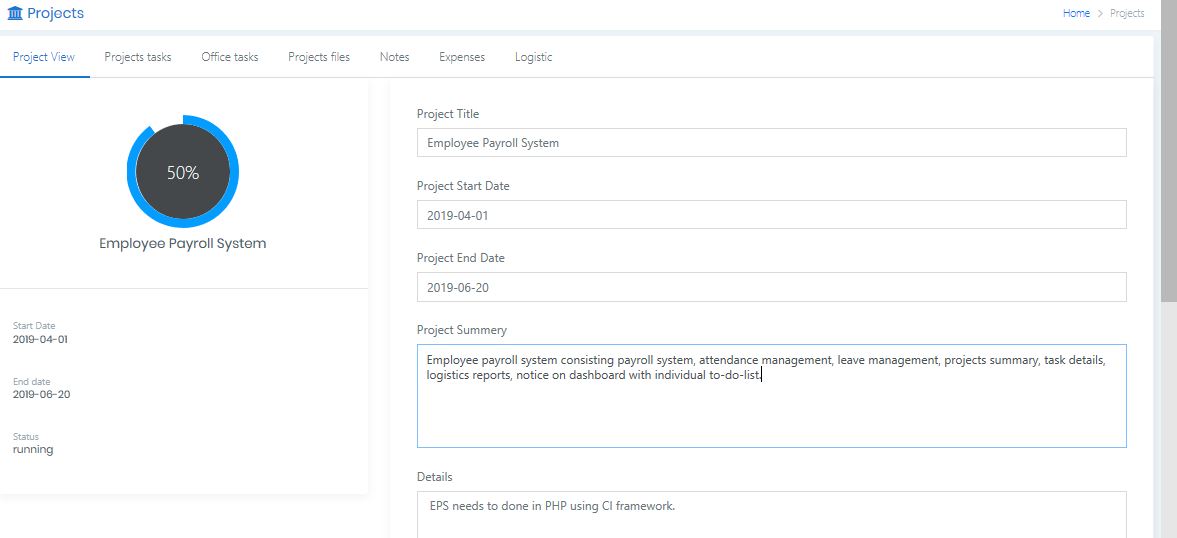




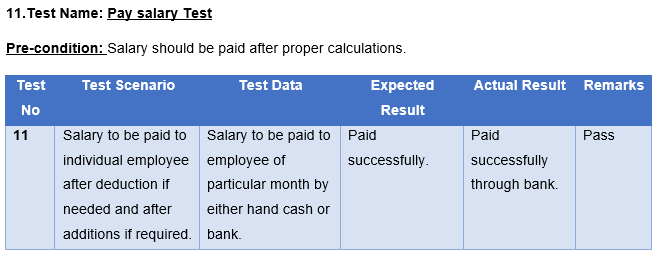


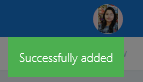
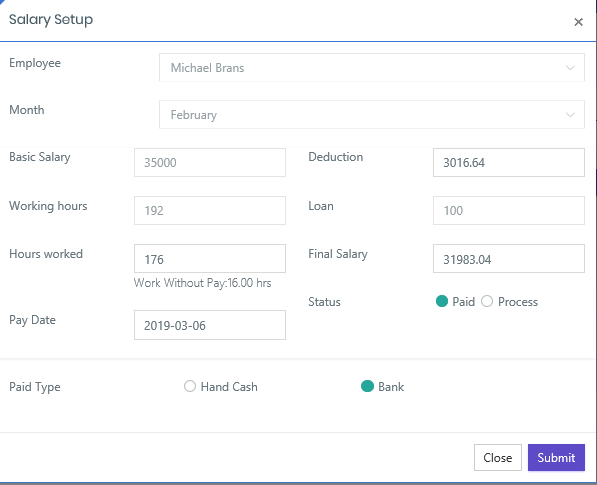


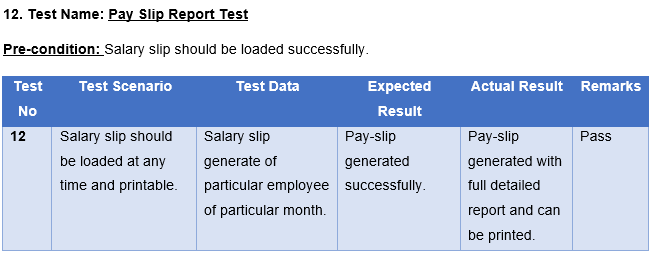


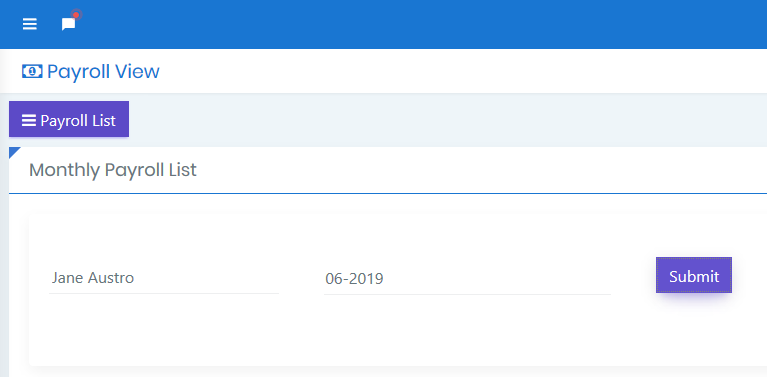


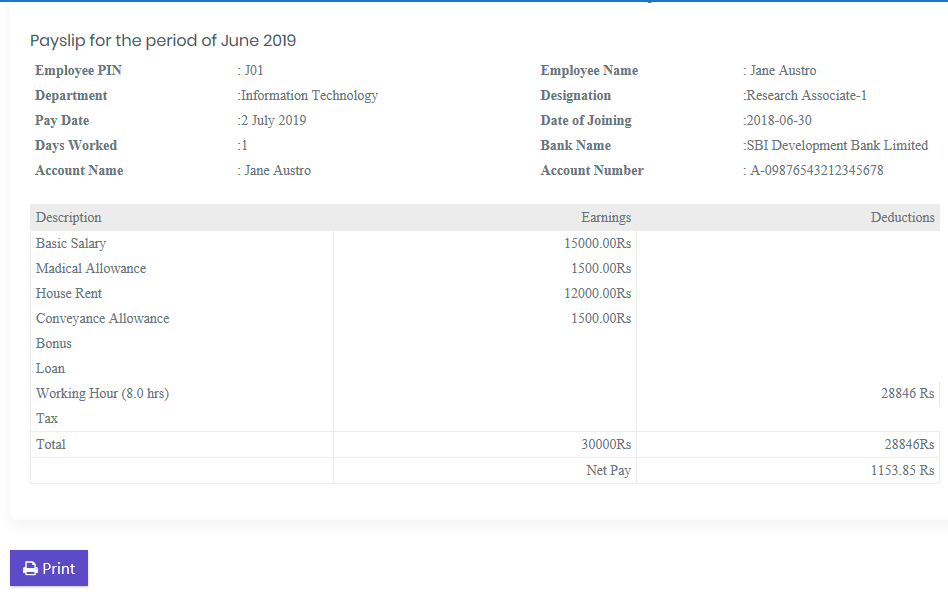


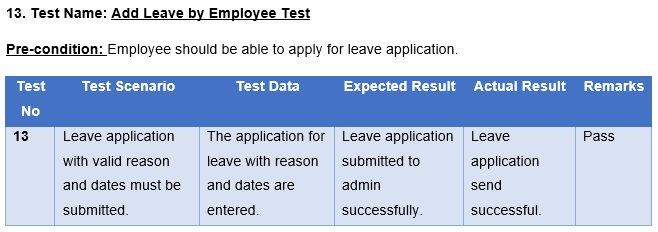
****

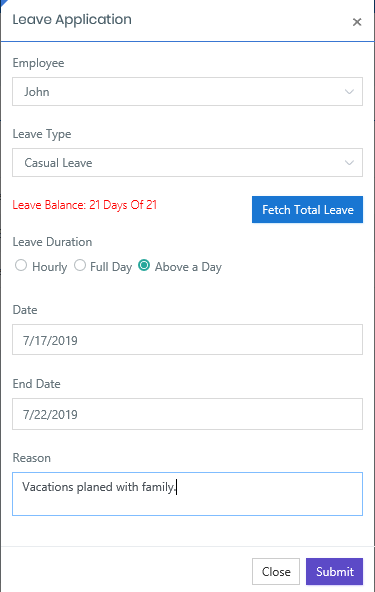
****

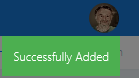
****

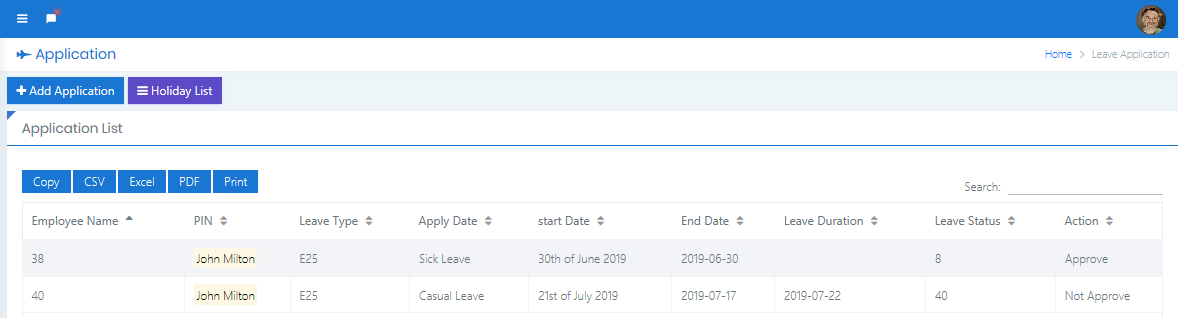
****

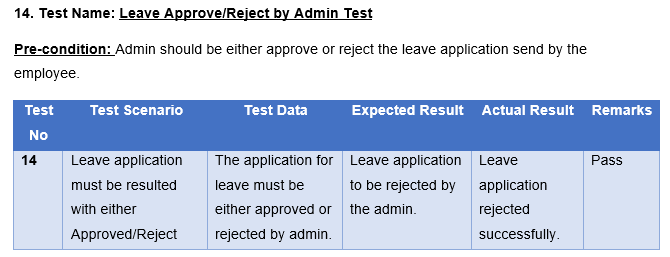
****

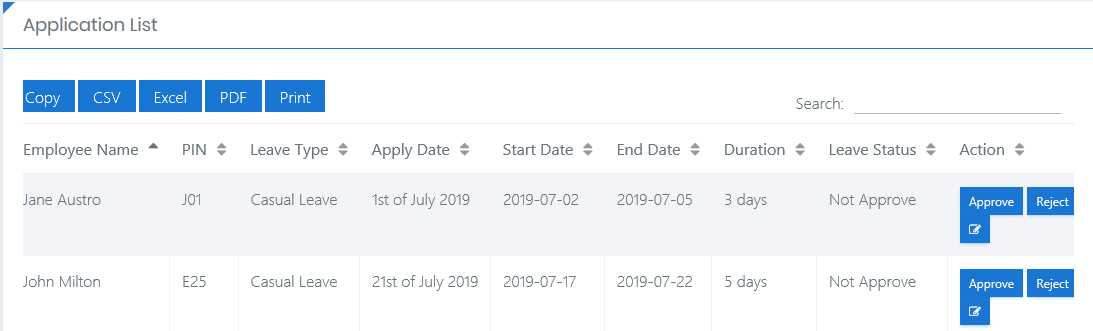
****

****

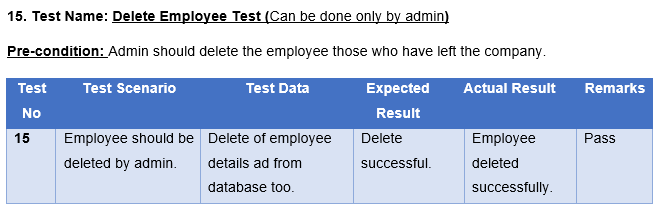
****

****

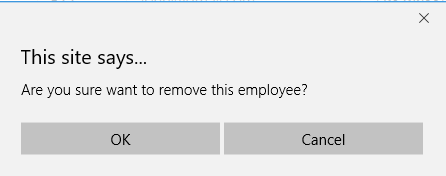


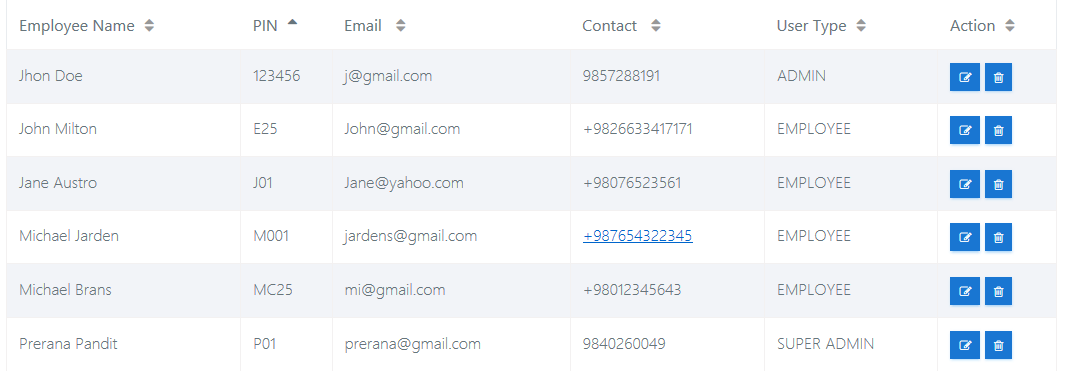




****

****

****

****

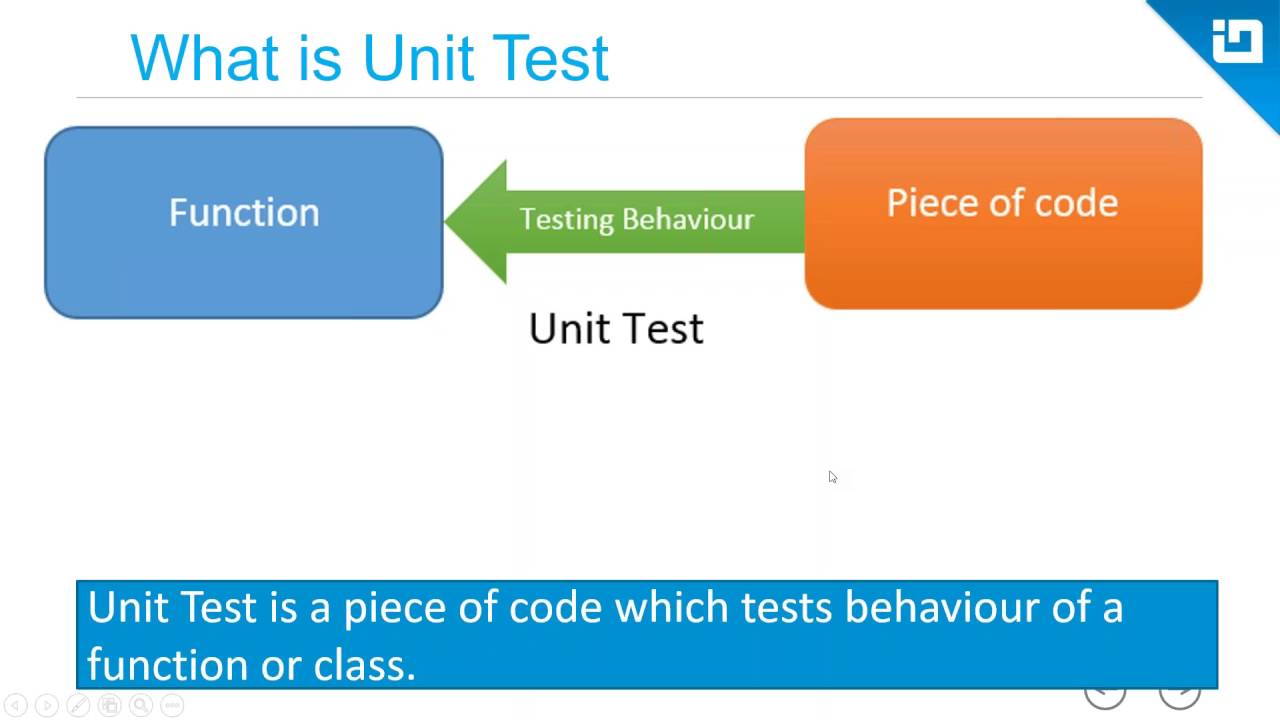
### **Unit Testing**

It is a method of testing that is used to validate the individual units of source code whether they are working properly. It is a testable unit as a function, procedure, etc. it verifies the individual part of the employee payroll system. Its objective is to isolate the section of the codes and verify the correctness of it. It cannot be performed by the non-technical persons. So, the unit testing is performed in employee payroll system using ‘**Test Unit’** library.

**Reason behind choosing unit testing**

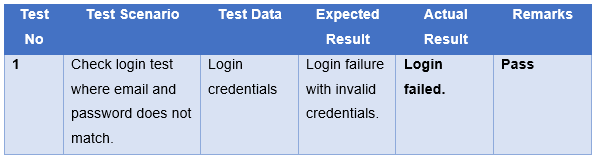
The testing fixes bugs early in the software development life cycle that prevents to spend more cost later. It enables the developer to change code quickly which helps them to understand the code base. It even helps to serve as a code documentation. It can be helpful for code re-use purpose that makes migrating the codes and testing new project.

Most importantly, unit testing makes codes more agile and improves the quality of code. It ensures readability, maintainability, and helps for refactoring purpose.



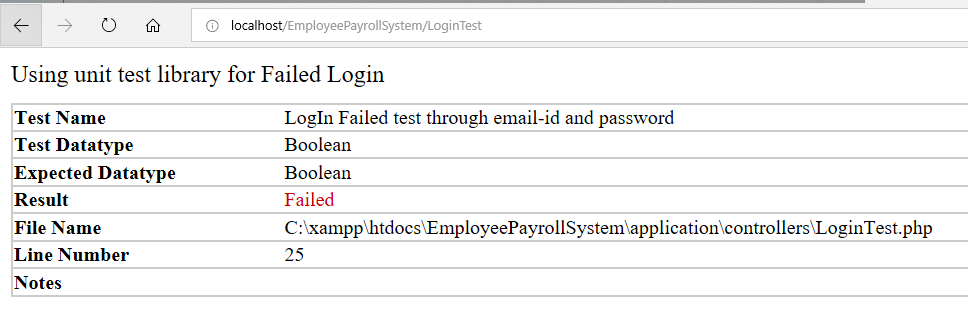
**Unit testing performed:**

1. **Test Name: Invalid Login Test**

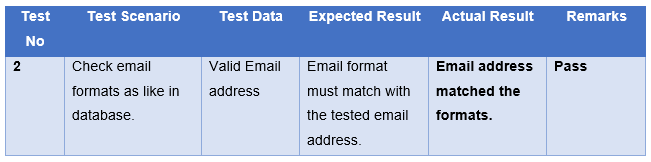
****

****

**Test Output:**

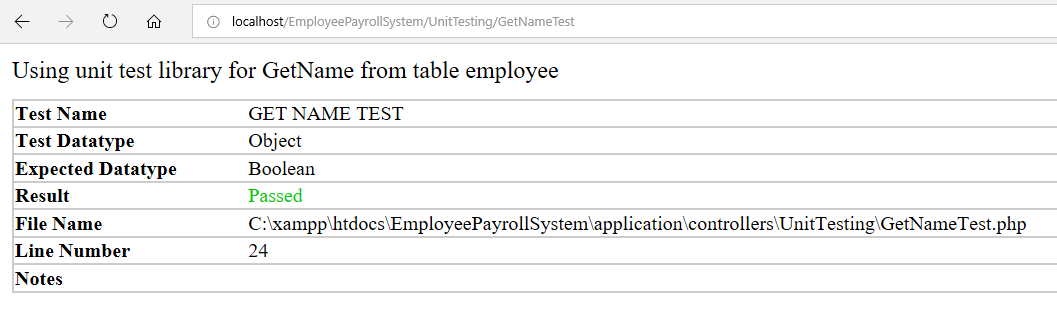
****

1. **Test Name: Check Email Test**

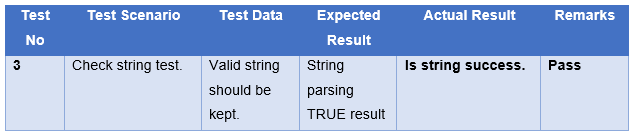
****

****

**Test Output:**

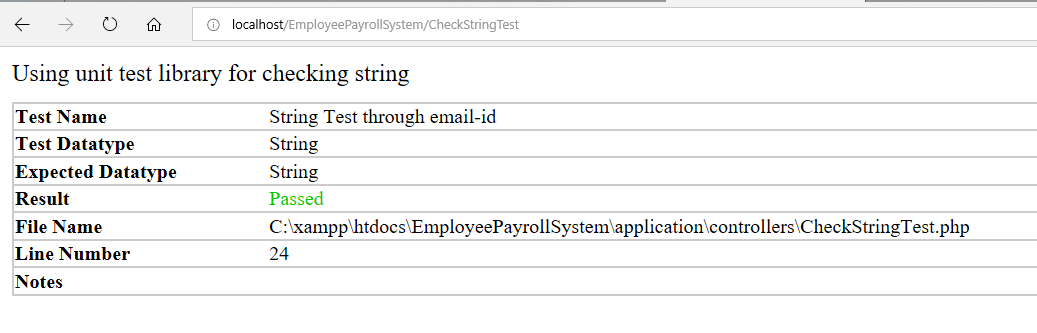
****

1. **Test Name: Check String Test**

****

****

**Test Output:**

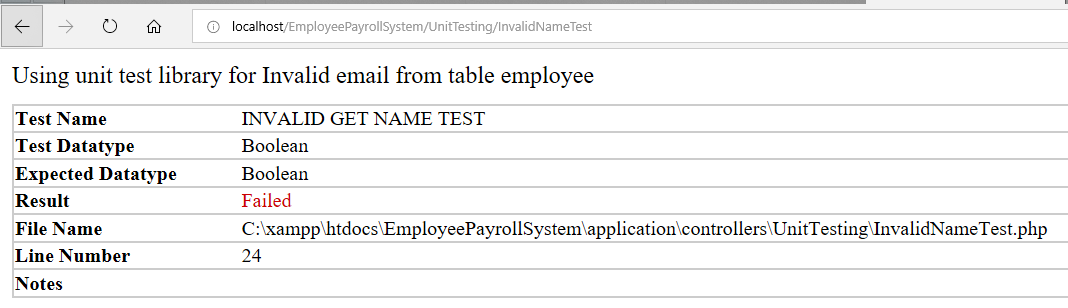
****

1. **Test Name: Invalid Name Test**

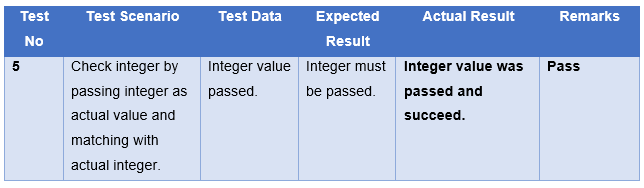
****

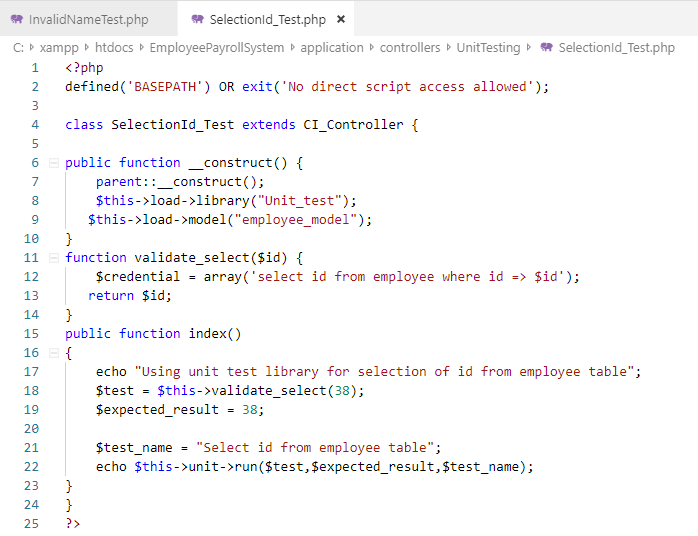
****

**Test Output:**

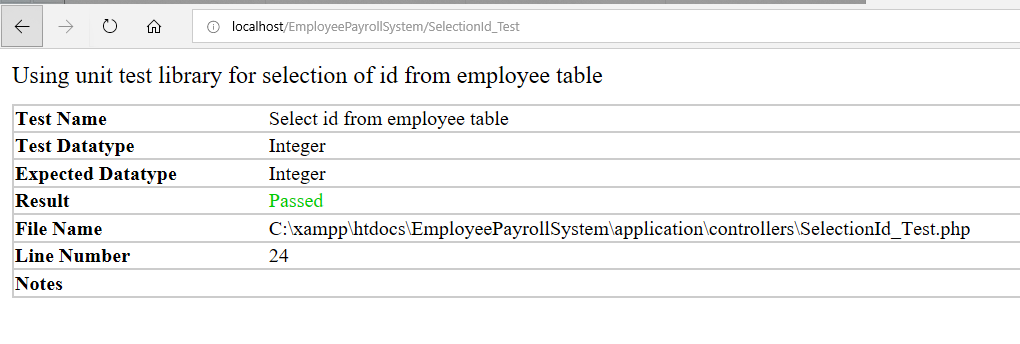
****

1. **Test Name: Check ID Test**

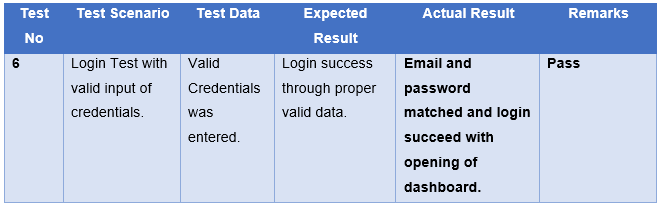
****

****

**Test Output:**

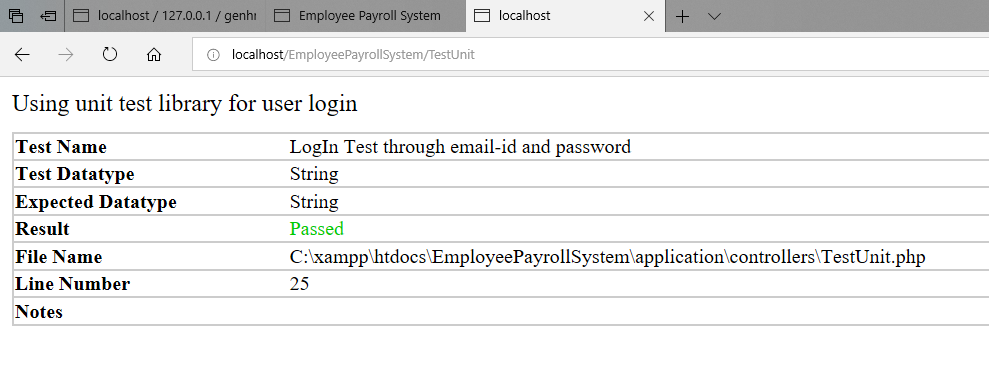
****

1. **Test Name: Log in Success Test**

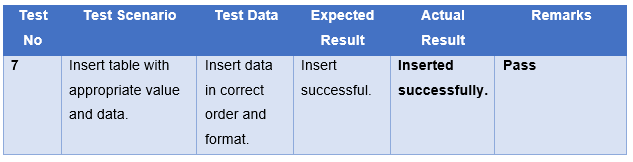
****

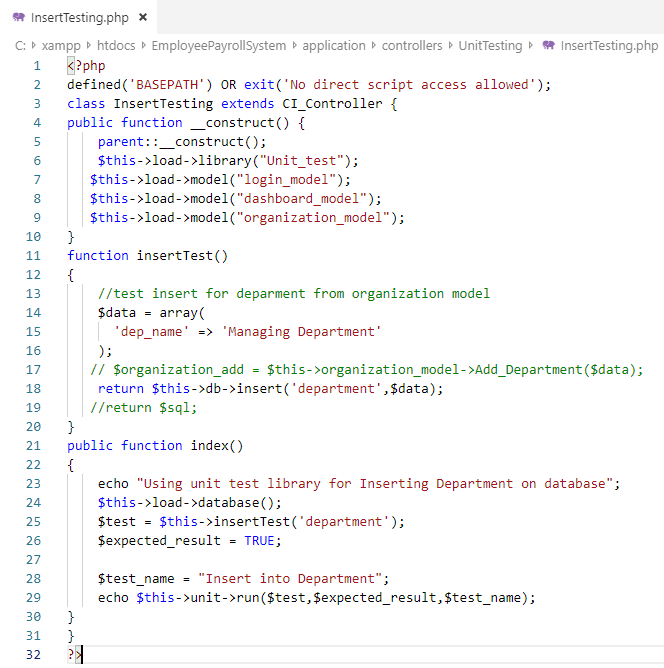
****

**Test Output:**

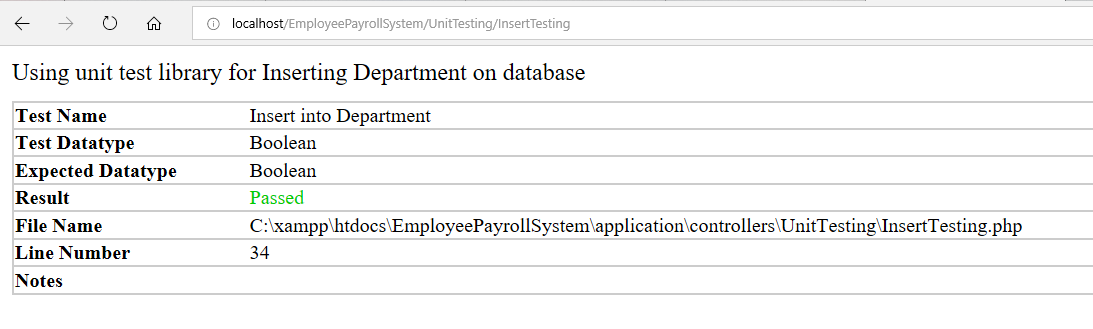
****

1. **Test Name: Insert Test**

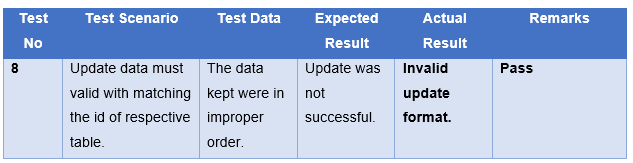
****

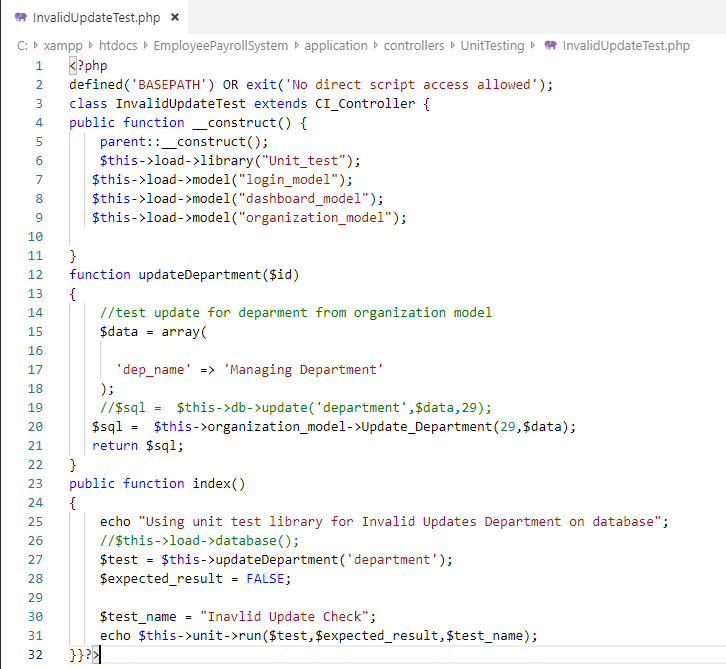
****

**Test output:**

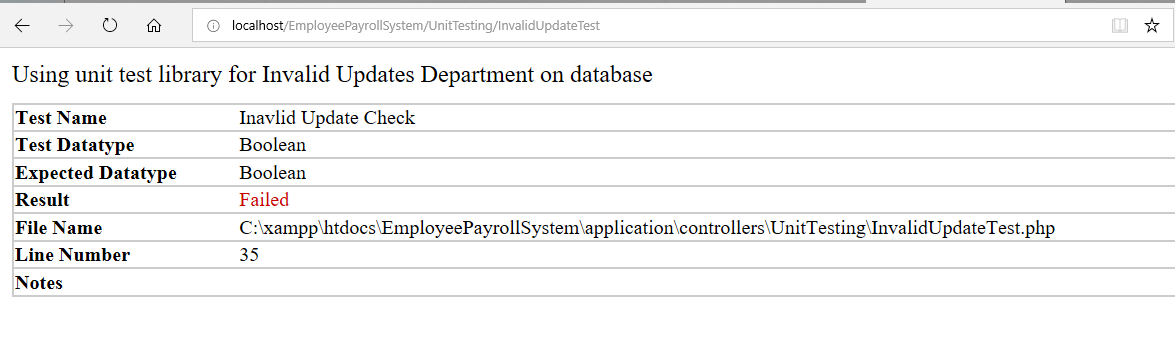
****

1. **Test Name: Invalid Update Test**

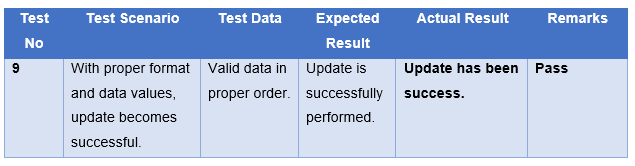
****

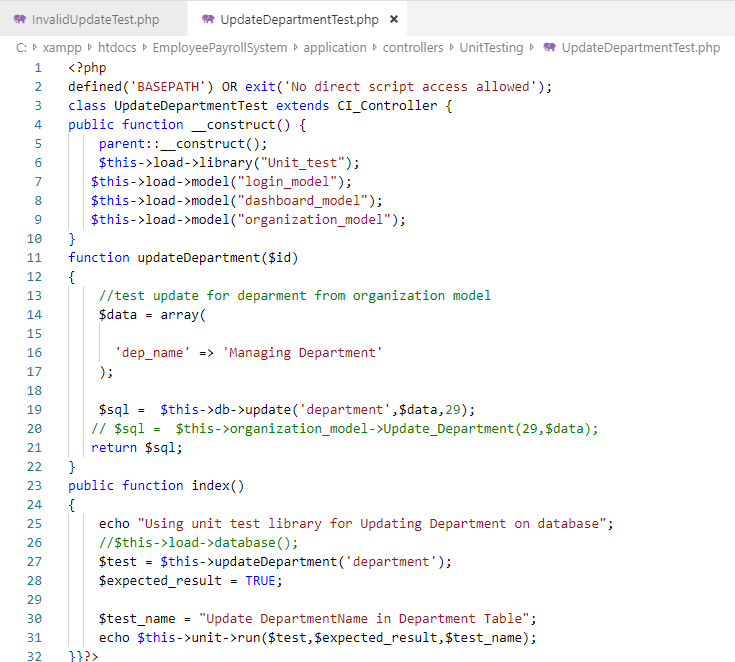
****

**Test Output:**

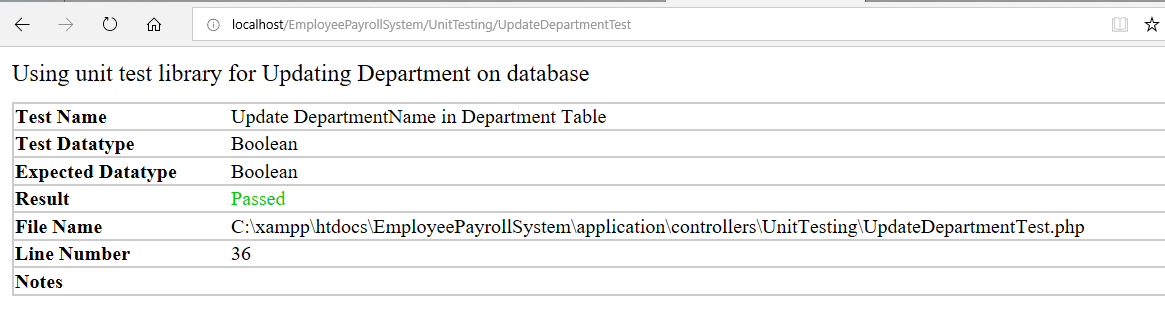
****

1. **Test Name: Update Success Test**

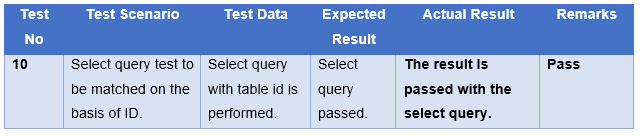
****

****

**Test Output:**

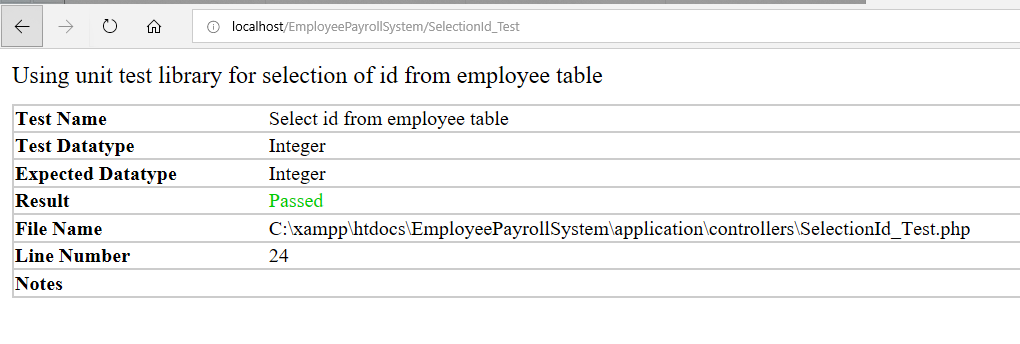
****

1. **Test Name: Select Success Test**

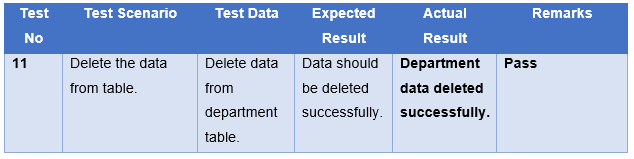
****

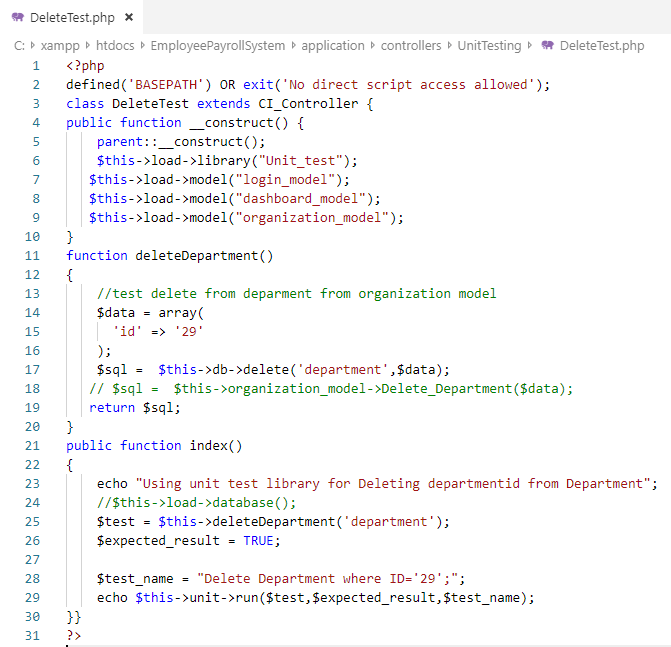
****

**Test Output:**

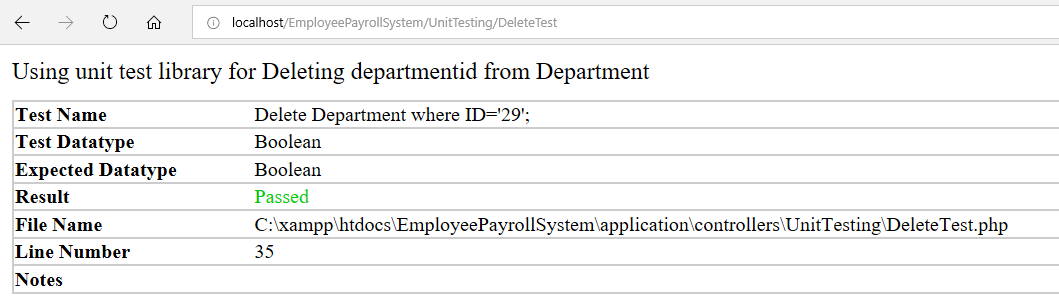
****

1. **Test Name: Delete Success Test**

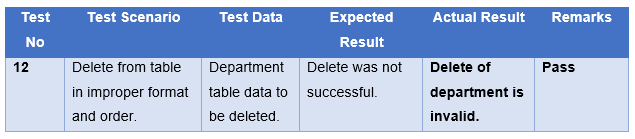
****

****

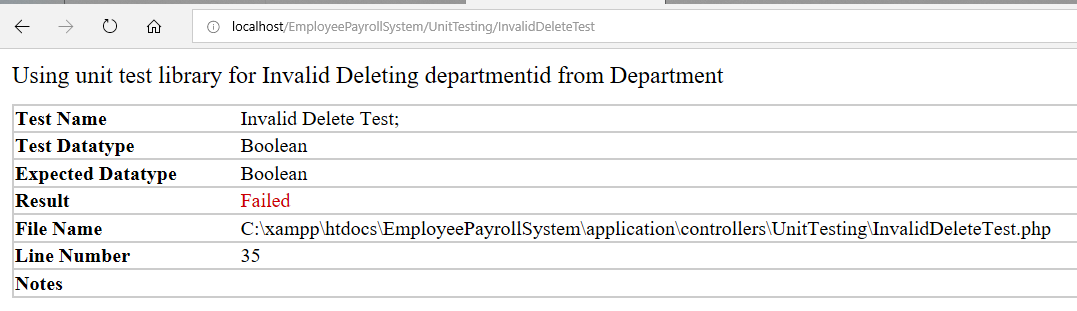
**Test Output:**

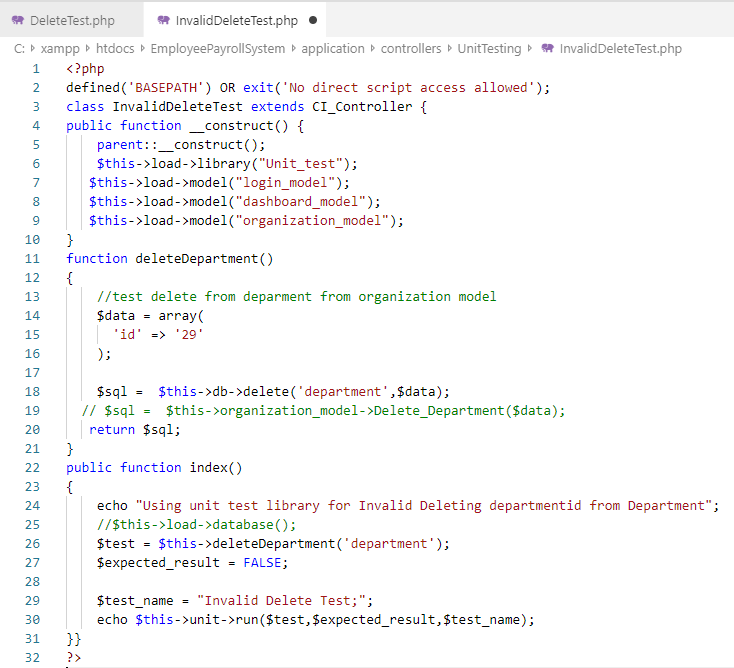
****

1. **Test Name: Invalid Delete Test**

****

**Test Output:**

****

****

# **Chapter 6: Other Project Issues**

Project issues are the events occurring and affecting the quality, schedule and cost of a project. The project issues are either called a risk that could not be mitigated. It has an negative consequences for a project and while developing a system, there appears many issues that needs to be controlled and solved. The major goal of the employee payroll system is to eradicate the issues by controlling the upcoming or ongoing risk occurring in the system.

## **Issues during the Project**

There were many issues faced during the development of the employee payroll system and those issues were handled properly and proper solutions were carried out practically. Some of the issues related are as listed below:

1. **Validation issues:**

During implementation phase, the major issues that was concerned was validation problem. The validation format did not support completely to the browser where the project was not able to lead ahead.

**Solution:** The format for validation purpose was changed throughout the project where the library of validation did not support.

1. **CodeIgniter environment did not match:**

The program suddenly stopped and did not worked resulting upon environment of CodeIgniter mismatch. The day before project was in running stage but after some certain changes in coding, the project showed an error of environment mismatch.

**Solution:** The problem was though solved after limited operations. The Xampp file was only re-installed where it disturbed upon CI environment which solved the error and project kept on running.

1. **Employees bulk attendance insert issue:**

The issue appeared during inserting the attendance in a **CSV** format in a bulk where the data was not inserted. The error of php encounter message was thrown continuously which stopped insert operation.

**Solution:** The bulk insert supporting CSV format library was in different format that the coding structure on controller file which disturbed the operations. Then, the proper structure was made in proper format and order that helped in bulk insert.

1. **Calculation of final salary issue:**

The major issue faced during the project development was the calculation part that became tough and took a long time to solve it. The appropriate calculation was not resulted as an output during generating the payment of employee at a particular time.

**Solution:** The calculation format was learned first from actual running payroll system then was implemented on this developing employee payroll system. This helped a lot and the issue was vanished.

1. **Approval on leave application:**

It was the issue that was to be consulted and eradicated completely. Being employees, they have a right for leave apply but the project was facing a problem regarding the leave application to be forwarded to admin, which was to be resulted as either approved or rejected outcome.

**Solution:**  I took a help from **stack overflow** where I got different solution regarding the issue. Finally, implementing one of the solutions worked very well and the problem was solved where leave application can easily be approved or rejected by the admin/ manager.

## **Evaluation of own work**

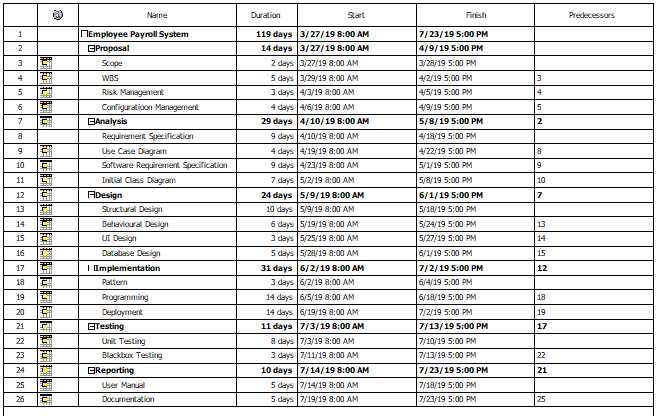
While developing the system, the work-evaluation is performed at first where we calculate the timing that will be consumed on particular phase or task. We usually assign the particular time for related phase or stage that must be accomplished as evaluated. The evaluation is done after proper analysis of each and every aspects of the project as a result to meet the requirements.

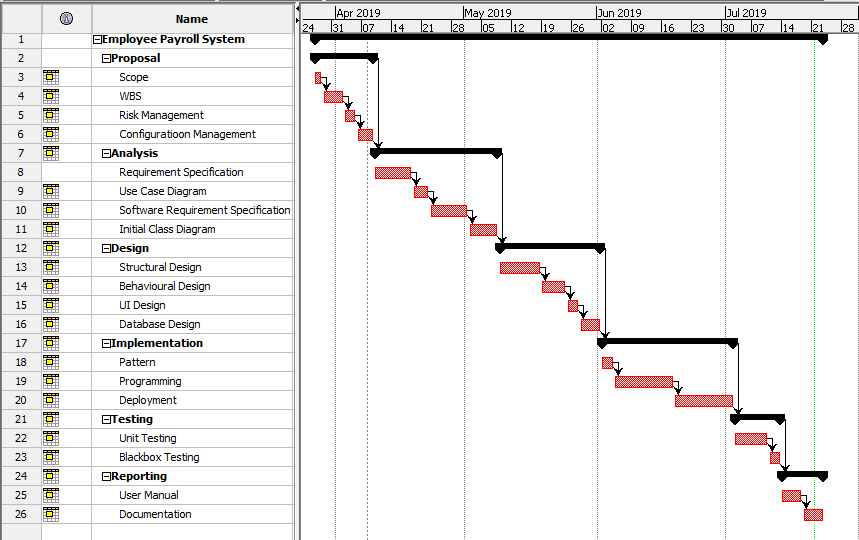
Similarly, evaluation of work has been done during development of employee payroll system which are shown on below phrases.

### **Scheduling**

The chart helps us to assess the timings of overall project for completion that determines resources and plans along with dependencies between each task. It simply represents visual view of each scheduled tasks with specific days and dates that represents timings of any task can be completed within certain time interval. The scheduling is represented through **Gantt Chart** drawn on **Project Libre.**

The scheduling was already covered up during proposal phase of employee payroll system but the updated scheduling that shows the duration of the task completion are as listed below.



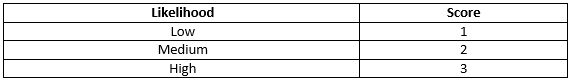


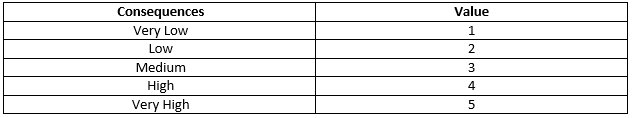
On the basis of planning and recommendations, the project was completed passing through several phases where some of them are wisely chosen and some of them are left out for future purpose. The above **Gantt Chart** represents of proper timing taken for development of employee payroll system as a whole where some of the schedules were overlapped and some of them went as directed.

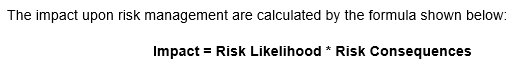
### **Risk Management**

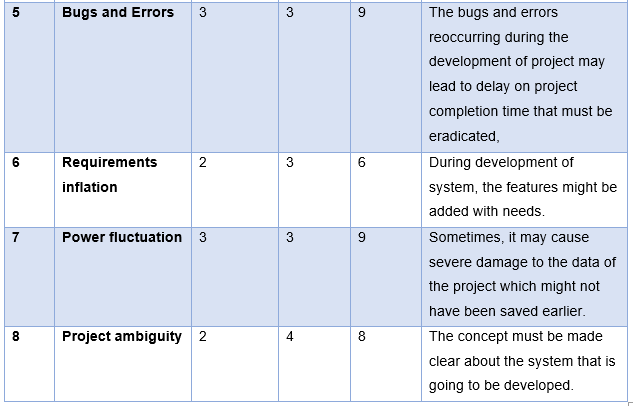
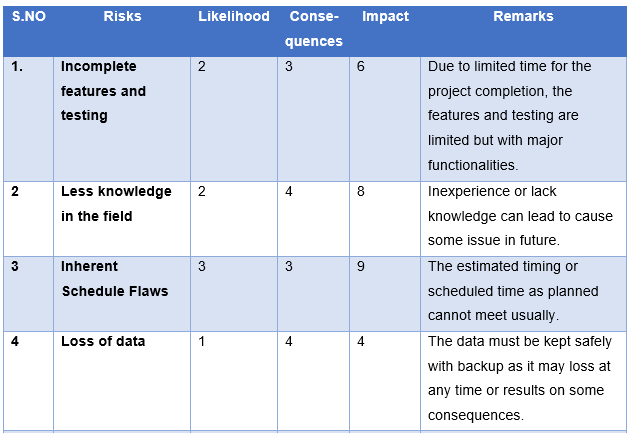
As we have already discussed about risk or issue that makes a negative impact upon system and the process where such risks are identified and migrating them are known as risk management.

It can also be known as vulnerability assessment that focuses on prevention of raised problems. The three sections in risk management are risk likelihood, risk consequences and impact upon risk occurrence.









### **Configuration Management**

Configuration management is the system engineering process that helps in improvement of change-impact analysis and reducing the outages. Providing audit and compliance support in operating accounting, configuration management focuses on establishment of a system performance, cost, requirements, design and its physical attributes acting upon rules and regulations. Controlling the process of modification of source code and development of documentation can be easily represented from working directory that consists of steps with respect to sub-directories.

Backup is an important aspect for any system which is also included during configuration management. For further backup, distributed version control ‘GitHub’ is being used where project is fully controlled with overall coding and proposals.

The account named in GitHub is <https://github.com/Preranapandit> where project is handled securely.

The tree structure of the directory is as listed below: