# **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**.

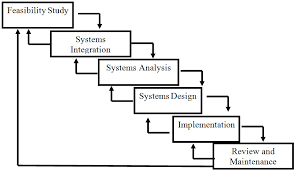
Besides requirements gathering, it helps in many other factors which are as listed below:

* Analyzing the processes of application that indicates its **feasibility**.
* Establishing **baseline** and control processes of the application.
* Focusing upon aims and objectives by defining their detailed **requirements**.
* Helps in creating **System Requirement Specification (SRS)** that represents functional and non-functional requirements along with hardware and software requirements.
* 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:

****

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

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

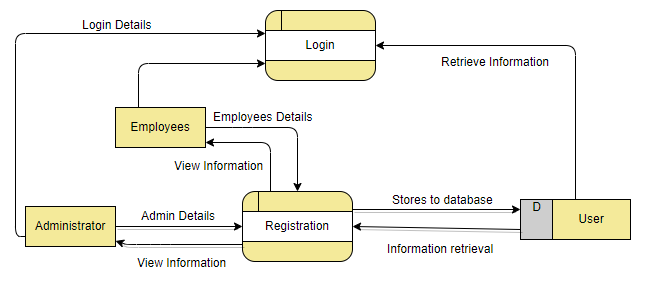
It is also a waterfall or cascade view of development with six different phases as **feasibility** study, system **integration,** system **analysis,** system **design, implementation** and **maintenance** which are comprehensive model supported by **CASE tools.**

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

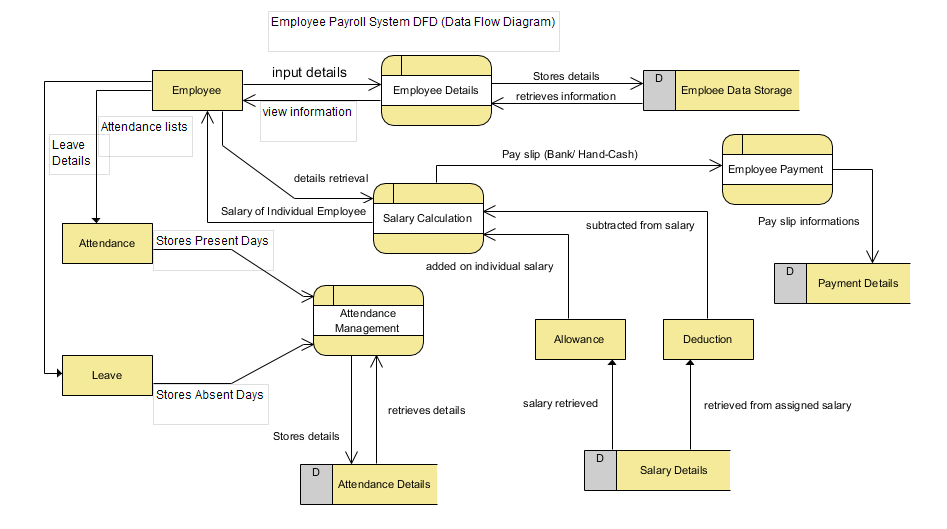
1. The first stage begins with **identifying problem** or **opportunity** of the system.
2. It illustrates the system in detail through **diagram** as **DFD**.
3. It illustrates the system objectives clearly.
4. It evaluates routes that involves **feasibility testing** or **pilot studies**.
5. It evaluates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



DFD (login and registration)

****

The above diagrams represents 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:

* 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.
* It is a rigid procedure and technique to provide unambiguous solutions to particular problems focusing on computer implementations.
* For each problem, it provides definite solutions with their achievable goals.
* It is mostly concerned with system or application dimension.
* It is mostly suitable to address issues pertaining to structured problems.

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

* It is not concerned with dealing with people and their values, perceptions and interests.
* It does not address the unstructured issues and it can only be applied for special cases or systems.

**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.



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

* **CATWOE (C**ustomers**, A**ctors**, T**ransformation, **W**orldview, **O**wners, **E**nvironment**)**
* **SWOT** analysis **(S**trength, **W**eakness, **O**pportunities, **T**hreat**)**
* **PEST** analysis **(P**olitical, **E**conomic, **S**ocial, **T**echnological**)**
* **Interviews**
* **Stakeholder matrix**
* **Surveys**
* **Focus groups**
* **Brainstorming**
* **Prototyping**
* **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.

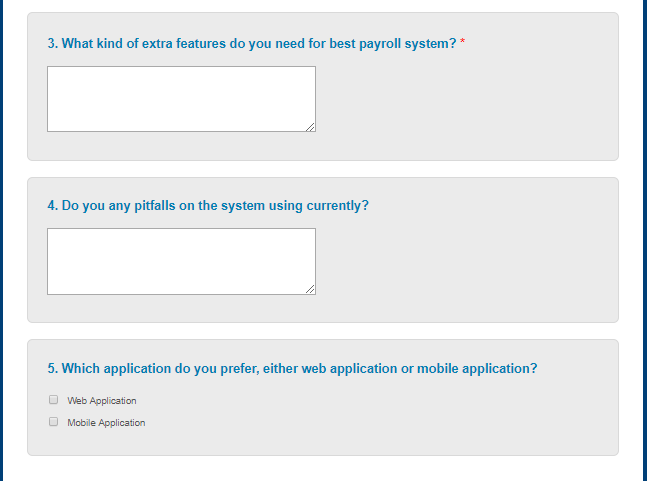
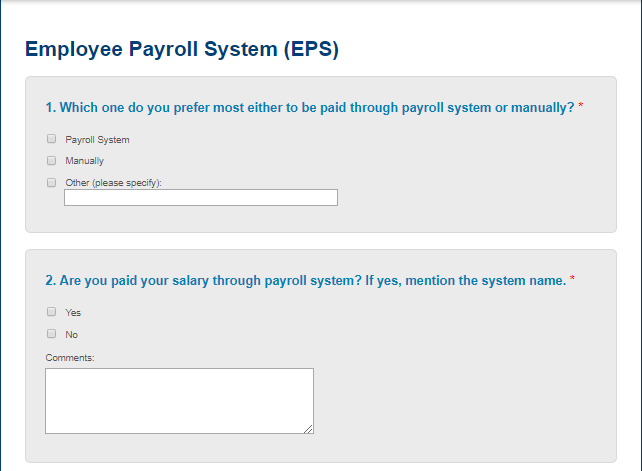
1. **Observations**

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:



## **2.3 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. Likewise, there are several types of feasibility studies which are listed below in the table.

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Types of feasibility study | Questions answered by study | Relation with project |
| 1 | **Operational Feasibility** | How a project plan satisfies overall requirements in analysis phase of system development? | As it provides advantage to employee and administrator both, it works with feasibility after installation. |
| 2 | **Economic Feasibility (Cost/benefit analysis)** | Does the cost, viability of the project provides positive economic benefit to organization? | The development cost is evaluated weighted against ultimate income and is economically justified for development. |
| 3 | **Technical Feasibility** | Does the technical resources meet capacity of organization? | The skill is growing through the technology of internet and the system day by day. |
| 4 | **Scheduling Feasibility** | Can the project be completed on scheduled time? | As development time effects machine time and cost, Gantt chart and work breakdown structure (WBS) is planned. |
| 5 | **Marketing Feasibility** | Does this market requires Employee payroll system for individual organization? | Through requirements elicitation techniques, marketing strategy has been performed carefully through questionnaires and many more for successful marketing. |

## **2.4 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.

**2.4.1 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 is being depicted through **function, data, rational, dependencies and remarks** also which is as listed below:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| FR.NO | Function | Data | Rational | Dependencies | Remarks |
| FR01 | Registration / Signup | FirstName : String  LastName : String  Age : Int  Email : String  Gender : String  Join Date : Date  Password : String | To create new employee / user. | N/A | New employee/ user will be added. |
| FR02 | Login / SignIn | Username : String  Password : String | To authorize registered users or employees only. | FR01 | Leads to employee’s/user dashboard |
| FR03 | Forget Password | Email / Username | To edit password from profile. | FR02 | Provides settings for new password. |
| FR04 | Edit Profiles | Edit details | To edit the employees or user details. | FR02 | Provides update/ modify for information of either employee or user. |
| FR05 | Delete Profiles | Delete details | To delete the employee data | FR02 | Provides deletion of data either by employee or admin itself. |
| FR06 | Basic Salary and Post recommend | Name : String  Designation : String  Department : String  Join Date : Date  Salary Type : String  Salary Amount : Float  Bank AC : String | To assign the designation/department of the employee with their respective salary and working hours. | FR02 | Admin provides suitable working hours with salary on the basis of employees post. |
| FR07 | Update Salary and Post | Salary entry details | To update/modify the designation and salary of each employee. | FR06 | Easy modification can be done if one’s post gets increased. |
| FR08 | Incentives | Employee details,  Incentives Information with percentage and amount | To reward financially to the employees from regular wage. | FR02, FR06 | It is pay-for-performance or compensation reward. |
| FR09 | Bonus | Employee details, Bonus Type  Bonus Amount | To reward employee with extra bonus on the basis of extra working hours. | FR02, FRO6 | Provides bonus to particular employee from their working hours. |
| FR10 | Deduction | Employee details,  Advance Payment,  Loan,  Professional Tax | To deduct from individual salary | FR02, FR06 | Deducts salary in case of advance salary taken or tax cut-off as a legal process. |
| FR11 | Leave approval | Employee details,  Leave type,  Leave Days,  Balance Leave | To provide leave for employee. | FR02 | Provides leave to respective employee for some reasons. |
| FR12 | Leave Status | Employee details,  Monthly Leave days | To maintain leave status of individual employee. | FR02, FR11 | Maintains leave status and can be used for reports as well as to deduct on salary. |
| FR13 | Search Employee | Employee ID | To search for employee through employee id | FR02 | It automatically search for employee on the basis of employee ID. |
| FR14 | Pay Slip | Employee Details  Basic Salary  Incentives Amount  Deduction Amount  Total Salary | To calculate correct salary of an employee. | FR02, FR06, FR08, FR09, FR10, FR11 | It automates salary of employee with pay-slip or it directly sends amount to their respective bank account. |
| FR15 | LogOut | Username  Password | To avoid security issues | FR02 | It secures and the current session gets closed. |

**2.4.2 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:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| NR.NO | Function | Description | Rational | Dependency | Remarks |
| NR01 | Security | System must have access control and data security. | To avoid security problems. | FR01, FR02, FR03 | Encryption is used for password of individual users with proper login and registration. |
| NR02 | Performance | System should perform smoothly and fast. | To make smooth and efficient system. | N/A | Correct technology with right architecture. |
| NR03 | Scalability | System should run even with huge data. | To handle work overflow. | N/A | Appropriate DBMS for data storage maintains its scalability. |
| NR04 | Concurrency | Must handle multiple computation. | To make multiple user handle system at a same time. | FR02, FR03 | Any employee can look over their own status and track reports at any time. |
| NR05 | Maintainability | System should be maintainable from any kind of errors or bug occurrence. | To maintain sustainability and growth of system. | N/A | For long term running system, maintainability plays vital role. |
| NR06 | Reliability | System should provide correct data of leave or present days of employees. | To maintain trust and eradicate misunderstandings. | N/A | Employees can be fully dependent and trust the system. |
| NR07 | Documentation | As a legal proof and future use, documentation must be done. | To make others learn about the system. | N/A | Proper documentation during the development phase becomes an tool for a person to understand and built it over again. |
| NR08 | Availability | As there is no appropriate timing for employee to leave a job, system must provide pay-slip even in middle of the month. | To provide full-satisfaction to employees. | N/A | System should be able to perform any kind of facility at any time. |
| NR09 | Usability | Easy to use and understand by end users. | To increase efficiency of the system. | N/A | With proper design and manual guide, one must be able to use the system on his/her own skills. |
| NR10 | Legal | System must be legally approved and operated. | To increase safety and legally valued. | N/A | Legal demands can ensure employees salary cut-off on the basis of tax to be paid for government. |
| NR11 | Adaptability | System must be adapted on any environment or operating systems. | To make system to work in any kind of environment with various versions. | N/A | System should accommodate changes. |
| NR12 | Robustness | System should cope with errors during wrong formats of input or exception cases. | To make system handle with any errors. | N/A | System becomes capable to deal with wrong input formats and exception cases. |
| NR13 | Testing | System should be testable as demo and then on overall systems. | To create system easily testable. | N/A | System must be testable without any hassle. |

**2.4.3 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.

**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:

|  |  |  |
| --- | --- | --- |
| ID | Functional Requirements | MoSCoW |
| FR01 | Registration | Must Have |
| FR02 | Login | Must Have |
| FR03 | Forget Password | Could Have |
| FR04 | Edit Profiles | Must Have |
| FR05 | Delete Profiles | Should Have |
| FR06 | Basic Salary and Post Recommend | Should Have |
| FR07 | Update Salary and Post | Must Have |
| FR08 | Incentives | Could Have |
| FR09 | Bonus | Should Have |
| FR10 | Deduction | Must Have |
| FR11 | Leave Approval | Could Have |
| FR12 | Leave Status | Should Have |
| FR13 | Search Employee | Could Have |
| FR14 | Pay Slip | Must Have |
| FR15 | LogOut | Must Have |

|  |  |  |
| --- | --- | --- |
| ID | Non-functional Requirements | MoSCoW |
| NR01 | Security | Must have |
| NR02 | Performance | Must have |
| NR03 | Scalability | Could have |
| NR04 | Concurrency | Must have |
| NR05 | Maintainability | Should have |
| NR06 | Reliability | Must have |
| NR07 | Documentation | Should have |
| NR08 | Availability | Must have |
| NR09 | Usability | Must have |
| NR10 | Legal | Should have |
| NR11 | Adaptability | Could have |
| NR12 | Robustness | Must have |
| NR13 | Testing | Should have |

**2.4.4 Hardware and Software Specification**

The hardware and software specification of the system are as listed below:

|  |  |
| --- | --- |
| Hardware Specification | Software Specification |
| * RAM : 4 GB * Hard Disk space : 300 GB * Processor : Dual core and above | * **Operating System : Windows 8, 10, Linux** * **Browser : Google Chrome, Edge, Firefox** * **Database : MySQL** * **Java : Oracle JDK 1.8, JRE 1.8** * **Web server : Apache Tomcat** |

## **2.5 Use Case Diagram**

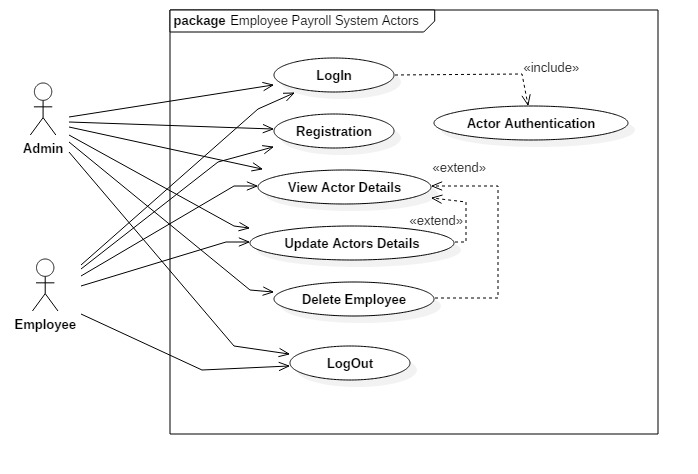
Use case diagram is a behavior diagram that **models the functionalities** using actors and use cases in **UML (Unified Modeling Language)**. The use case defines the functions or task which are done by particular actors. It is also a collection of diagram and text together.

**Why do we need Use Case Diagrams?**

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

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

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:

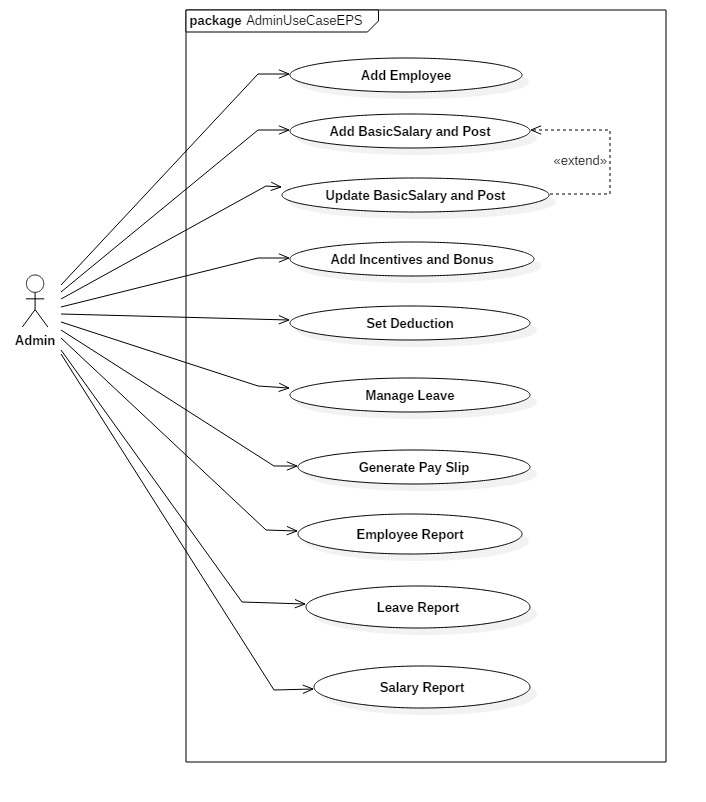


**Scenario Description:**

**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:

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

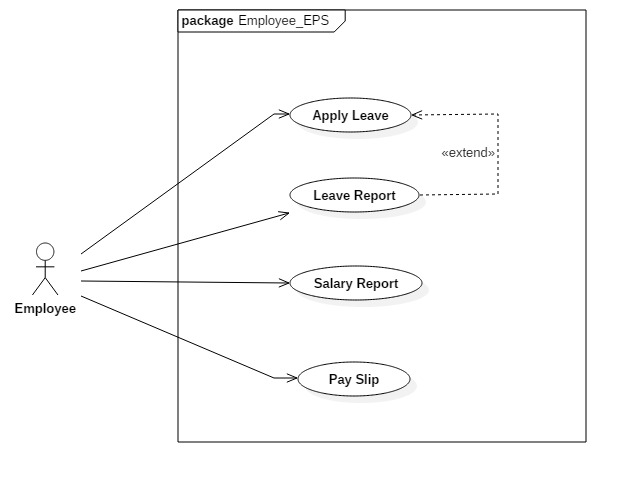


**Scenario Description:**

**Actor: Admin**

The use case diagram shown above represents the operations that are authorized to admin. Admin is the most important actor of Employee Payroll System that plays vital role performing overall functions and doing overall calculations along with tracked reports. The operations done by Admin are as listed below:

* Admin can add employees with their full detailed information with the help of system.
* Admin can assign the provided post and valid salary to respective employees.
* Admin has an authority to change employees post and salary in case any promotion done to the employees.
* As per employee working performance or extra working hours, admin can easily add incentives and bonuses to the salary of the respective employee.
* The leave period of employee are managed by admin though the system and gets easy for deduction on salary on the basis of their leave period.
* The pay slip during payment time of the employee is being handle by admin where appropriate salary without any errors is automatically calculated on respected date and time period.
* Reports like employee report, leave report and salary report can be visible to admin where he/she can track the employees’ performances and progresses.



**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. Similary, the operations or functionalities that can be performed by the employees are as listed below:

* Employees are able to get leave or mention their leave period though the help of system.
* Employees can look after their leave reports wherever they want which is extend of a leave applied by themselves.
* They can look after their salary report in any confusions which helps in building trust between employee and the organization.
* Employees are able to get pay slip after generated by admin. The salary can either be taken through hand-cash or simply they can allow admin to send their amount to their bank account.

## **2.6 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)**

Our task is to create a new Employee Payroll System, as the old system has become outdated and cannot manage overall payroll process. Moreover, the system was using manual method in some cases to provide salary to the employee. Therefore, new employee payroll system has been customized with complete functions and actions to be done by each one.

The new system can be operated by both Employee and Admin where they perform various actions with proper authorization and authentication. Both of the users are allowed to register upon the system and login through dashboard where username and password are kept aside for security purpose. During registration, First name, Last name, Join Date, Email, Address, Gender, Contact Number, Password are kept where login requires username and password. Both users can update their details and even logout to make their account secure. Likewise, admin plays many role as Add employee, Add basic salary where salary amount and post are assigned for employee. Update salary and post with adding bonus and incentives with proper deduction on salary amount are performed as an operations. Admin can even manage leave of the employee with generating the pay slip at last. The reports are available in the system for both users, where employees can track their performances and progresses rapidly. The system has got the role for employee also where he/she can simple apply for leave either for some days or weeks and can even achieve their personal leave report as per demand. They are even provided view format for reports of salary and pay slip after finalized salary calculated by the admin.

Finally, the system is secured and maintainable having credited banking facilities with eradicating manual-handling process which is understandable through user-manual and easy to operate.

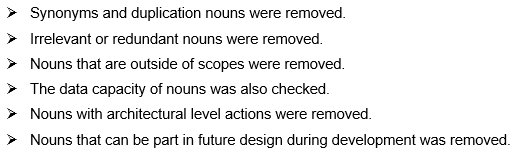
**Step 1: Finding nouns from scenario for potential classes:**

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

|  |
| --- |
| Task, Employee, Payroll, System, System, Payroll, Process, System, Method, Cases, Salary, Employee, Employee, Payroll, System, Functions, Action, System, Employee, Admin, Actions, Authorization, Authentication, Users, Register, System, Login, Dashboard, Username, Password, Registration, First name, Last name, Join Date, Email, Address, Gender, Contact number, Password, Login, Username, Password, Users, Details, Logout, Account, Admin, Role, Salary amount, Post, Employee, Bonus, Deduction, Salary amount, Operation, Admin, Role, Employee, Pay slip, System, Users, Employee, Track, System, Role, Employee, Days, Weeks, Salary, Pay slip, Salary, Admin |

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**

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO. | Candidate Class (Noun Identification) | Is Selected? | Justification |
| 1 | Task | No | It goes outside the scope as per scenario. |
| 2 | **Employee** | **Yes** | It is identified as noun as per scenario because the payroll system completely depends upon employees working basis. |
| 3 | Payroll | No | It is proper noun but can be indulge in future use for making separate payroll scheme. |
| 4 | System | No | It is not suitable for the system as it itself represents system which cannot be class. |
| 5 | **Salary** | **Yes** | It tends to be candidate class as it has many attributes in itself that depends upon employees. |
| 6 | Admin | No | It is also a candidate class but will be used in future after development and implementation. |
| 7 | Users | No | The employee and admin will be the user of the system. |
| 8 | Dashboard | No | It is not a class but a setup page that occurs after login after successful authentication. |
| 9 | Username\* | No | It is defined attribute that can be used inside candidate class. |
| 10 | Password\* | No | It is used for an attribute inside appropriate class. |
| 11 | First name\* | No | It is well-defined attribute for candidate class. |
| 12 | Last name\* | No | It can be used as an attribute for candidate class. |
| 13 | Join Date\* | No | It gets involved in candidate class being itself as an attribute. |
| 14 | Email\* | No | It represents attributes for candidate class. |
| 15 | Address\* | No | It is an attribute for candidate class. |
| 16 | Gender\* | No | It is an attribute for candidate class. |
| 17 | Contact Number\* | No | It can be used as an attribute for candidate class. |
| 18 | Account | No | It is out-of-bound for system that contains extreme attributes which cannot be included easily. |
| 19 | Salary Amount\* | No | It can be used as an attribute in another class. |
| 20 | **Post/Designation** | **Yes** | It provides proper definition as candidate class with attributes inside relating many to one relationship. |
| 21 | Bonus\* | No | It is valid attribute that can be used in any candidate class. |
| 22 | Deduction\* | No | It is valid attribute that can be used in any candidate class. |
| 23 | **Pay slip** | **Yes** | It is well-defined class having many attributes inside that relates to employee with their salary and overall features. |
| 24 | Days\* | No | It can be used as attribute for other candidate classes. |
| 25 | **Leave** | **Yes** | It is well-defined class that represents appropriate relation between employees depending upon their present days. |
| Note: Nouns that are marked with \* are attributes. | | | |

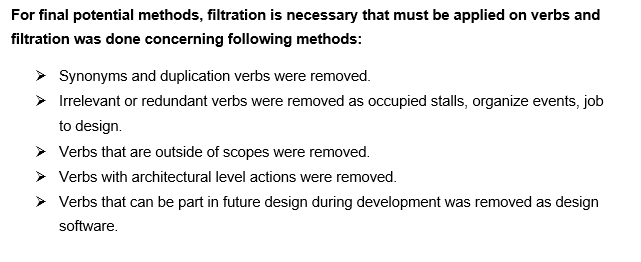
**The list of final candidate class are as listed below:**

|  |  |
| --- | --- |
| S.NO | CLASS |
| 1 | Employee |
| 2 | Salary |
| 3 | Designation/ Post |
| 4 | Pay Slip |
| 5 | Leave |

**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:

|  |
| --- |
| Create, Manage, Operate, Update, Add, Add, Update, Update, Delete, Add, Add, Generate, Report, View, Report, Calculate |



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

|  |  |  |  |
| --- | --- | --- | --- |
| S.NO | Potentials Methods  (Verb Identification) | Is Selected? | Justification |
| 1 | Create | No | It is not specific for making operation or methods being ambiguous word to process. |
| 2 | Manage | No | System is overall managing approach but cannot be used as operation. |
| 3 | **Add** | **Yes** | It is appropriate method as add is applied in employee and other classes. |
| 4 | **Update** | **Yes** | It shows operation to edit any data which can be done in any classes. |
| 5 | **Delete** | **Yes** | Deletion is an operation or method which can be applied on any classes. It removes the data of any required field. |
| 6 | Generate | No | It cannot be assigned as method but can be used in project or system. |
| 7 | Report | No | It can be used for system to view reports on particular basis. |
| 8 | **Calculate** | **Yes** | It is an method or operation where total salary after incentives or deduction is calculated of individual employee. |

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

|  |  |
| --- | --- |
| S.NO | VERBS |
| 1 | **Add** |
| 2 | **Update** |
| 3 | **Delete** |
| 4 | **Calculate** |

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

|  |  |  |
| --- | --- | --- |
| Class | Attributes | Method |
| Employee | **Emp\_Id: Integer**  **FullName: String**  **ContactNo: String**  **Email: String**  **Address: String**  **JoinDate: Date** | **Add Employee**  **Update Employee**  **Delete Employee** |
| Salary | **Salary\_Id : Integer**  **BasicSalary : Float**  **IncrementSalary : Float**  **DecrementSalary : Float**  **NetSalary : Float** | **View SalaryDetails** |
| Designation | **Designation\_Id : Integer**  **DesignationName : Varchar**  **NetSalary : Float** | **Add designation**  **Update designation** |
| Pay\_Slip | **PaySlip\_Id : Integer**  **Month : Integer**  **Year : Varchar**  **Net Salary : Float** |  |
| Leave | **Leave\_Id : Integer**  **LeaveTitle : Varchar**  **LeaveDetail : Varchar**  **LeaveApplyDate : Date**  **LeaveStartDate : Date**  **LeaveEndDate : Date**  **LeaveDays : Integer** | **Add leave**  **Cancel leave**  **View leave** |

**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.

**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:

* Class diagram helps to illustrate data models for system without any difficulties.
* It can be easy to understand even by the general people.
* Specific needs of a system can be easily expressed to the organization.
* It helps in creating detailed chart with their main functionalities to be implemented carefully.
* 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:

