# **2.1 Introduction to Analysis**

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

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

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

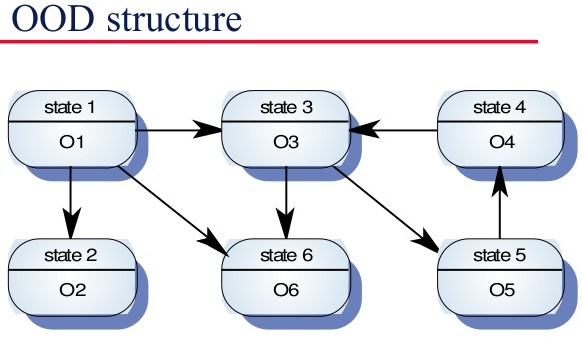
# **2.2 Analysis Methodology**

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

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

The benefits of using Object Oriented Design Methodology are:-

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



# **2.3 Feasibility Study**

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

**Technical Feasibility Study**

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

**Schedule Feasibility Study**

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

**Economic Feasibility Study**

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

**Cultural Feasibility Study**

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

**Political Feasibility Study**

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

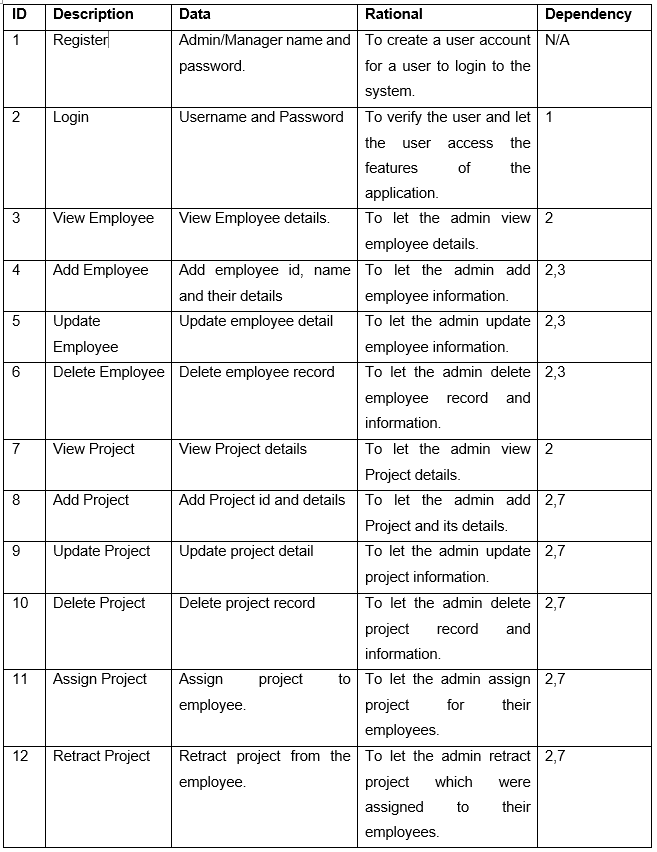
**Safety Feasibility Study**

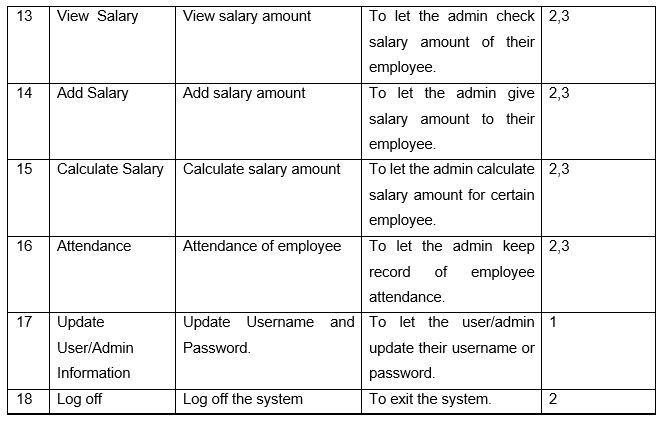
It is a study done to find out whether the project being developed can be developed without major obstruction or having negative impact on the stakeholders and environment.

# **2.4 Requirement Analysis**

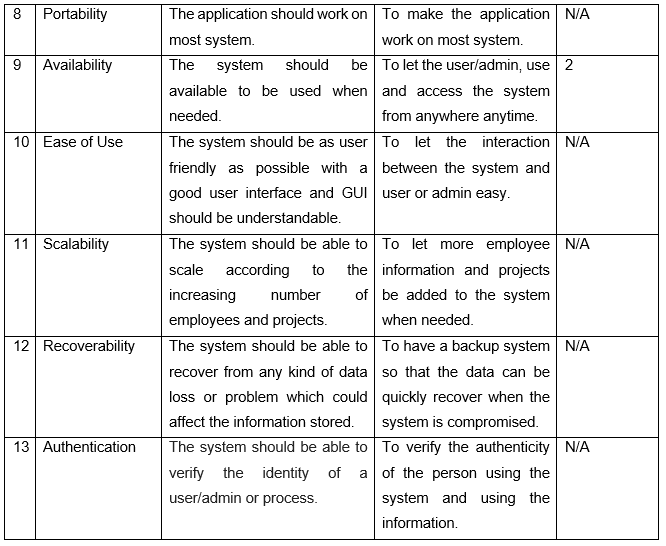
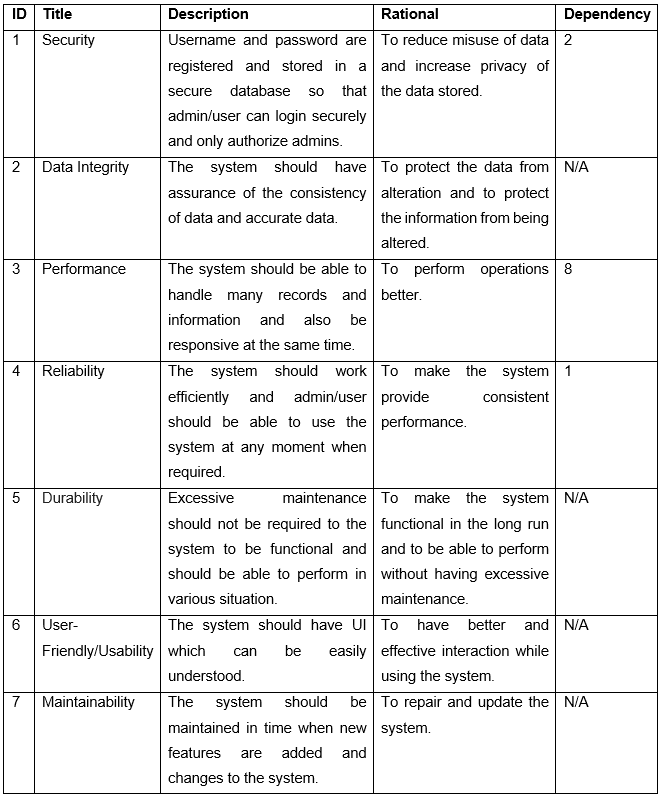
One of the most important part of analysis in project management is requirement analysis. Requirement analysis can be defined as the processes of determining needed and relevant requirements to meet the user expectation and the requirements should be detailed and specified. The two important types of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

# **2.4.1 Functional Requirements.**





# **2.4.2 Non-functional Requirements**



# **2.4.3 MoSCoW Prioritization**

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

The four MoSCoW prioritization categories are:

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

Prioritization for Functional Requirements:

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

Prioritization for Non-Functional Requirements:

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

# **2.4.5 Software and Hardware Requirement Specification**

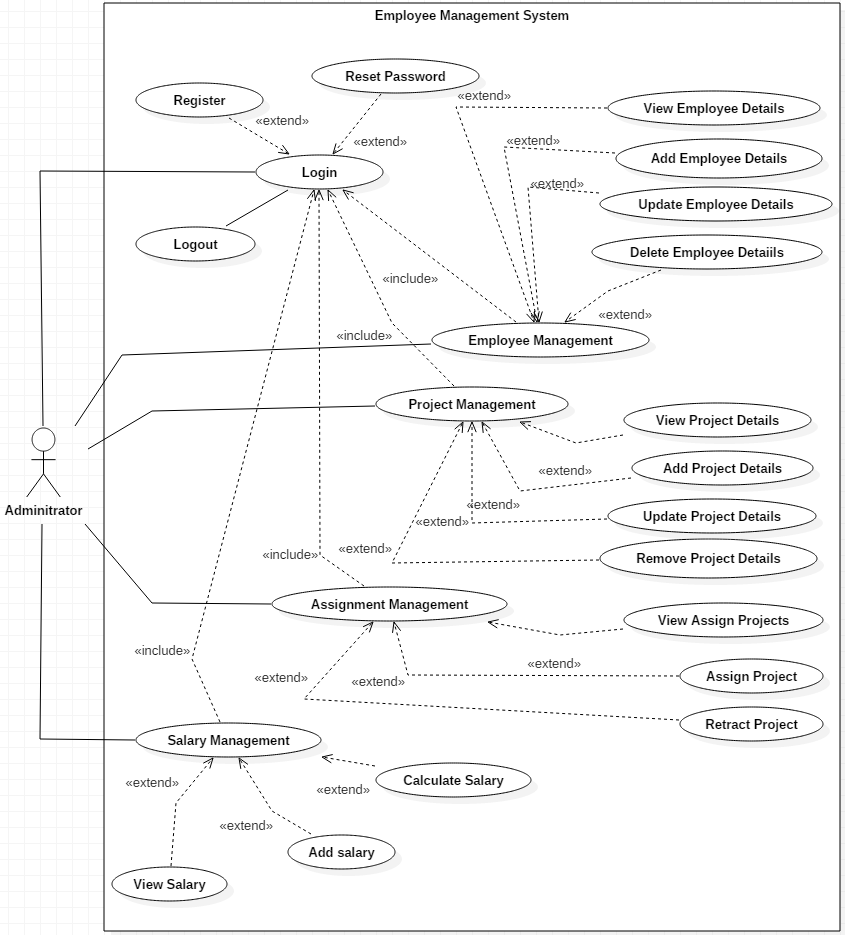
Software and hardware requirement specification can be described as any precise information about hardware or software products and their requirements.

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

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

# **2.5 Use Case**

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



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

# **2.6 Natural Language Analysis**

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

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

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

**Candidate Class**

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

# **Initial Class Diagram**

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

