**CHAPTER 1.PROJECT IDEA & NEED**

* The basic idea behind this project is to develop a web based software application which can handle software project tracking with employees’ information like attendance and daily work tracker.
* There will be multiple types of users. One will be from Top Management, who will look after the whole process, will insert master table entries like adding Employee Information, Project/Activity Information and generate reports. Another user will be Employee, who will report their daily work information in the system for their respective project.

**1.1 PROJECT PROFILE**

Project Title : Software Project Management System

Objective : The purpose of the system is to provide web based

Application which can help to manage projects

According to client’s requirements and employees

Information and daily work tracking.

Developed For : Archer Solutions, Ahmedabad

Developed At : Archer Solutions, Ahmedabad

Documentation : Microsoft Office XP.

Tools

Front – End : JSP, Servlet.

Back – End : MySQL.

IDE : Net beans

External Guide : Ankit Virparia

Internal Guide : Prof. M.B. Choudhari

Submitted By : Dhrumit Shah (090130107019)

Pravesh Shah (090130107012)

Submitted To : Government Engineering College Of Gandhinagar

(Gujarat Technological University)

**2 SOFTWARE REQUIREMENT SPECIFICATIONS**

**2.1 INTRODUCTION**

**2.1.1BACKGROUND**

What is Software Project Management System?

* When the plan starts to involve different things happening at different times, some of which are dependent on each other, plus resources required at different times and in different quantities and perhaps working at different rates, the paper plan could start to cover a vast area and be unreadable.
* Nevertheless, the idea that complex plans could be analyzed by a computer tallow someone to control a project is the basis of much of the development in technology that now allows projects of any size and complexity, not only to be planned, but also modeled to answer 'what if?' questions.
* The original programs and computers tended to produce answers long after an event had taken place. Now, there are many project planning and scheduling programs that can provide real time information, as well as linking to risk analysis, time recording, and costing, estimating and other aspects of project controller
* But computer programs are not project management: they are tools for project managers to use. Project management is all that mix of components of control, leadership, teamwork, resource management etc that goes into successful project.

**2.1.2 OVERALL DESCRIPTION**

**Need For System**

Currently, there is no system available to maintain the workflow in the organization. Employees generally use E-mails to forward or submit the status of task given to them. This is not the efficient way, as some times top management are even not aware that what is the final status of particular task and they need to consult employee or its mentor to ask the status orally.

MODULES:

1 Project Management

2 Employee Management

3 Client Management

4 Module Management

5 Notification

6 Personal Message

### 

1.Project Management :

In project management admin add the project by discussing it with client about start date,end date,cost etc. When the project is confirmed by the client admin add the final values.

2 Employee Management:

Here admin manages the employee.Admin fill the details of employee and manages the employees.And these employees assign into different project by the project manager.

3. Client Management

Admin manages the client .Admin fill the details of client and manages the client.

4.Module Management

Here in module management project manager of the project add the modules in that project and assign employees in that module.project manager can track the work of employees.

5. Notification

Here admin,project manager can add notification in particular module of the project and all the employees working in that module get the notification.

6. personal message

Admin,Project manager ,employee and client can send messages to each other.

**2.1.3 ENVIRONMENT CHARACTERISTICS**

**(i) HARDWARE**

|  |  |  |
| --- | --- | --- |
| **Component** | **Minimum** | **Recommended** |
| **Processor** | Pentium-4 | Dual Core |
| **RAM** | 256MB | 512MB |
| **Display** | 800\*600 | 1024 × 768 |
| **Hard disk** | 40 GB | 80 GB |
| **Network** | 56 kilobits per second (Kbps) connection between client computers and server | 56 Kbps or faster connection between client computers and server |

**Table 2.1 Hardware**

**(ii) PERIPHERALS**

No peripherals are required for this system.

**iii) PEOPLE**

There are three types of people interact with the system.

These are:

1) Admin:

* + Manages whole site
  + Can add, edit, delete employees
  + Can add, edit, delete projects/activity
  + Do In/Out of particular employee (exclusive rights)
  + View employee work report
  + Send project work report after specific time

2) Project Manager:

* Who must plan, motivate, organize, and control the employees who do software work.
* Rating calculation of employees’ work

3) Employees’:

* + - Add their daily work reports

4) Customer:

* who specify the requirements for the software to be engineered and other stakeholders who have a peripheral interest in the outcome

**2.2 GOALS OF IMPLEMENTATION**

Project management is the discipline of defining and achieving a set of goals while optimizing the use of allocated resources (time, money, people, space, etc). This includes planning, scheduling and maintaining progress of the activities that comprise the project. Project management is normally reserved for focused, non-repetitive, time-limited activities with some degree of risk and that are beyond the usual scope of program (operational) activities for which the organization is responsible.

Project management software describes the tools to efficiently coordinate and automate the various project management component processes. Project management software generally offers extensive reporting features, such as day-to-day status updates of project progress, scheduling and dependency trees, and system-generated alerts when schedules slip beyond pre-set tolerances. Most project management tools include web-accessible interfaces so that employees can access features of the software relevant to their needs, and functionality to allow managers to share resource pools without overbooking.

**2.3 FEASIBILITY STUDY**

An important outcome of the preliminary investigation is the determination that the system is feasible or not. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop a project .The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output required to be produced by the system as well as the various constraints on the behavior of the system.

Three types of project feasibility have been considered:

**Economic Feasibility**

The system being developed is economic with respect to user’s point of view. It is cost effective in the sense that has eliminated the paper work & transportation cost completely.

**Technical Feasibility**

The technical requirement for the system is economic and it does not use any other additional Hardware and software.

**Behavioral Feasibility**

The system working is quite easy to use and learn due to its simple but attractive interface. User requires no special training for operating the system.

**Chapter 3: PROJECT MANAGEMENT**

**3.1. PROJECT PLANING AND SCHEDULING**

**3.1.1 PROJECT PLANNING KEY TASKS:**

1. Set goal and scope

2. Select life cycle

3. Set organization team form

4. Start team selection

5. Determine risks

6. Create WBS

7. Identify tasks

8. Estimate size

9. Estimate effort

10. Identify task dependencies

11. Assign resources

12. Schedule work

**3.1.2 PROJECT DEVELOPMENT MODEL**

**Incremental Model**

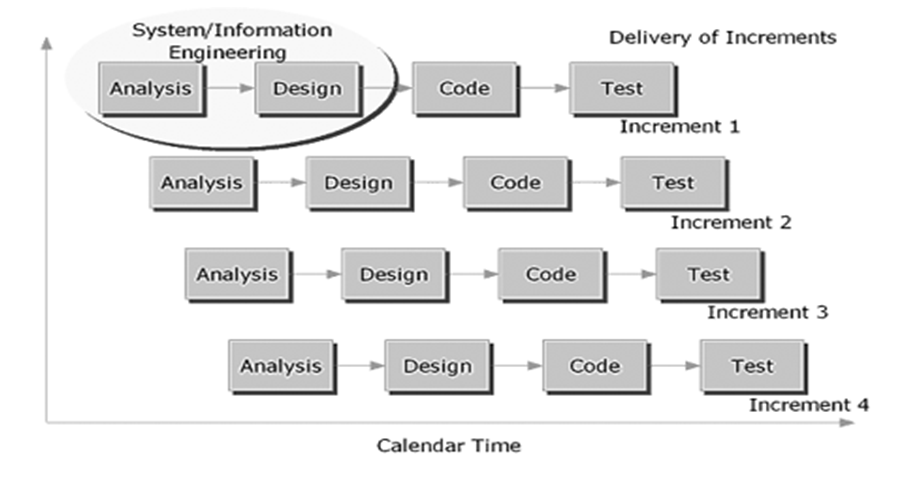
* The incremental model has same phases that are in waterfall model. But iterative in nature. The incremental model includes following phases.

1. Analysis phase
2. Design phase
3. Coding phase
4. Testing phase

* The incremental model delivers series of release to the client.

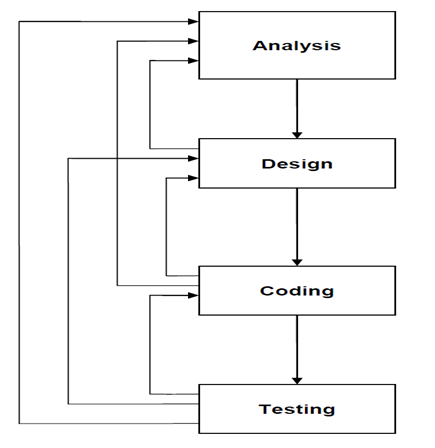
These releases are called increments. More and more functionality is associated with each increment.

* The first increment is called core product. In this release the basic requirements are implemented and then in subsequent increments new requirements are added.
* The word processing software package can be considered as an example of incremental model. In the second increment, more sophisticated document producing and processing facilities, file management functionalities are given. In the next increment spelling and grammar checking facilities can be given. Thus in incremental model progressive functionalities are obtained with each release.



**Figure 3.1 Incremental Model**

**3.1.3 PROJECT PLAN**



**Figure 3.2 Project Plan**

 Analysis

After deciding the project we did the analysis of the project. We did the analysis such that no. of user required in the project, no. of modules required in the project, and various features required in project.

 Design

The designing of the project is done according to the analysis and the features required. We started with our project designing according to the module. We completed our one module and start for the other.

 Coding

As soon as the basic designing of the form was completed we began with the coding of that form again we analyse some basic features we again do the coding as per our features we want to add.

 Testing

At the end of each and every phase is on coding the individual objects described in the object design document.

**3.1.4 SCHEDULE REPRESENTATION**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Index** | **Task Name** | **Task Description** | **Deadline**  Week number  from the start  of the project | **Note** |
| 1 | Requirements Specification | Complete specification of the system (with appropriate assumptions). | 1 | Attempt should be made to add some more relevant functionality other than those that are listed in this document. |
| 2 | Technology familiarization | Understanding of the technology needed to implement the project. | 5-6 | The presentation should be from the point of view of being able to apply it to the project, rather than from a theoretical perspective. |
| 3 | Database creation | A database of at least 100 entries of employees of all grades should be created. | 3-4 | It is important to finalize on the database at this stage itself so that development and testing can proceed with the actual database itself. |
| 4 | High-level and Detailed Design | Listing down all possible scenarios like different types of tutorials. Then coming up with flow-charts or pseudo code to handle the scenario. | 7-9 | The scenarios should map to the requirement specification as well as suggestions on Students part would be invited with concrete reasoning. |
| 5 | Implementation of the front-end of the system | Implementation of the main screen giving the login, screen that follows the login giving Various options, screens for each of the options. | 6-7 | During this milestone period, it would be a good idea for the team (or one person from the team) to start working on a test-plan for the entire system. This test-plan can be updated as and when new scenarios come to mind. |
| 6 | Integrating the front-end with the database | The front-end developed in the earlier milestone will now be able to update the employee database. In short the system should be ready for integration testing. | 4-5 |  |
| 7 | Integration Testing | The system should be thoroughly tested by running all the test cases written for the system. | 3-4 | Another 2 weeks should be there to handle any issues found during testing of the system. After that, the final demo can be arranged. |
| 8 | Final Review | Issues found during the previous milestone are fixed and the system is ready for the final review. | 2-3 | During the final review of the project, it should be checked that all the requirements specified during milestone number 1 are fulfilled (or appropriate reasons given for not fulfilling the same) |

**Table 3.1 Schedule Representation**

**3.2 RISK MANAGEMENT:**

**3.2.1 RISK IDENTIFICATION**

To be able to manage project risks, you must first understand what constitutes, a risk. All uncertain occurrences are not risks. Only those occurrences that have an adverse impaction the progress of a project are risks to the project. Risk is not a bad thing. Risk is bad only when it results in loss for an organization. Unless there is a potential for loss, there is no risk. Moreover, loss can be interpreted as either a bad outcome or a lost opportunity. The tendency of most project managers is to jump at the statement this is a risk.

However, the desired reaction is to pre-empt all possible outcome and plan for them. Project risks can be broadly categorized into development process risks and product risks.

**Development Process Risks**

The risks encountered during product development are categorized as development process risks. These comprise developer errors, natural disasters, disgruntled employees, and poor management objectives. Developer errors could be attributed to poor training due to budgetary constraints and

Inadequate skills and software tools. Ergonomic problems, environment problems, and interruptions or distractions at office also account for developer risks. Other risks in this category include problems in personnel acquisition and retention. Similarly, natural disasters such as flood, cyclone, fire, storm, and snowfall are also risks to a project. Disgruntled employees can also become a risk to an organization. For example, a sacked employee can use password snuffers to gain unauthorized access. A dismissed person can flood the system with senseless messages. A disgruntled employee can also try to sabotage the project work by destroying files and programs. A poorly defined management objective is another development process risk. If the language in the management objective is ambiguous and not stated clearly, the risk

management program will not function properly. Narrowly focused and changing objectives that are not updated can also be counted as risks. Lack of contingency plans, incomplete cost estimates, and unrealistic schedules are also potential risks in a project. Similarly, unrealistic performance standards are also potential risks to the development process. Other possible risks include contractual risks, technological risks, and inadequate documentation of other concurrent projects.

**Product Risks**

Product risks crop up in the form of changing requirements during product development. Incomplete and unclear requirements are a risk to the product during development. Similarly, problems in meeting design specifications can also be categorized as risk to product development. Risks could arise if the project deliverables or objectives are not clearly defined or if

technical data is missing. The possibility of several alternatives at any given time during the project is also a cause of concern. If errors are not recognized during the design phase, they could turn into risks for the project. Similarly, risks could arise due to the size and complexity of the product or while achieving client acceptance of the product.

**3.2.2 RISK MANAGEMENT PROCESS**

Project Risk Management includes the processes concerned with conducting risk management planning, identification, analysis, responses, and control on a project. The objectives of Risk Management are to increase the probability and impacts of positive events and decrease the probability and impacts of events adverse to project objectives. The process of risk management begins during the analysis phase of software development life cycle. However, the actual process of managing risks continues throughout the product development phase. Risk management is a dynamic process because it deals with the activities that are yet to happen. Risk management has a two- fold agenda. First, deciding actions for preventing risks from happening, and second, deciding actions for tackling risks that materialize. Therefore, risk management is all about pre-empting a risk, coming up with a plan for resolving the risk, and finally

executing the plan.

Risk management process consists of three steps:

1. Risk identification

2. Risk analysis

3. Risk Migration

1. Risk identification:

In this step, the project manager gathers information about the potential risks in the project. The project manager plans the strategies for avoiding risks or controlling them. The project team conducts brainstorming sessions and discussions among team members about the requirements document. They discuss the available technology, manpower, prevailing environment, and all project-related factors. The project manager picks up the thread from these and creates a risk log. After the risk log is prepared, the project manager calls a meeting within the team and technical experts to discuss the risk log and

the mitigation plans. An effective way of identifying' risks is using a questionnaire.

2. Risk analysis:

After identifying the risks, the project manager needs to analyze the risks. Uncertainty and loss are the two characteristics of risk. The uncertainty factor in risk means that the unknown event mayor may not happen. While analyzing risks, the project manager needs to quantify the level of uncertainty and the degree of loss. Based on this, the project manager plans schedules and costs. During analysis, information on risk is converted into information on decision-making. Analysis provides the basis for the project manager to work on the "right" risks.

3. Risk migration:

Risk mitigation is the best possible approach adopted by the project manager to avoid risks from occurring. The probability of the risk occurring and the potential impact of the risk can be mitigated by dealing with the problem early in the project. Essentially, risk mitigation involves three possibilities and the project manager needs to adopt a risk mitigation strategy aimed at them.

The three possibilities include:

· Risk avoidance

· Risk monitoring

· Contingency planning

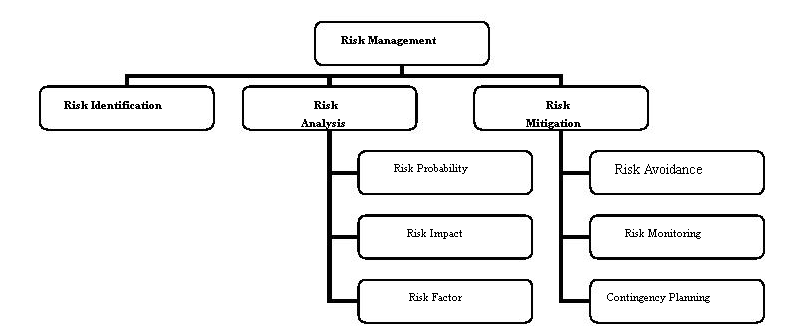


Figure 3.3 Risk Management Process

**Chapter 4: SYSTEM MODELLING**

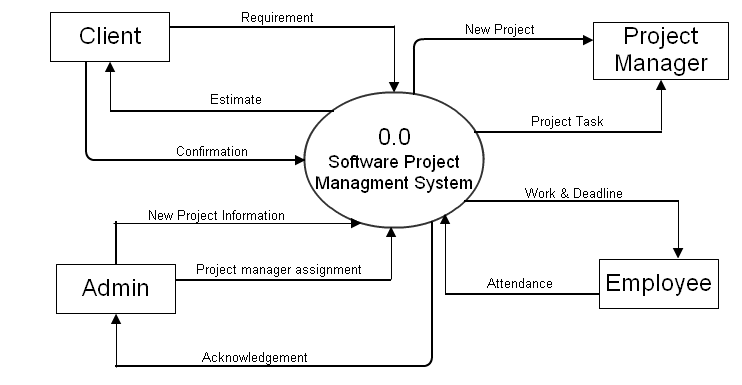
**4.1 DATA FLOW DIAGRAMS**

# Context Level Diagram:

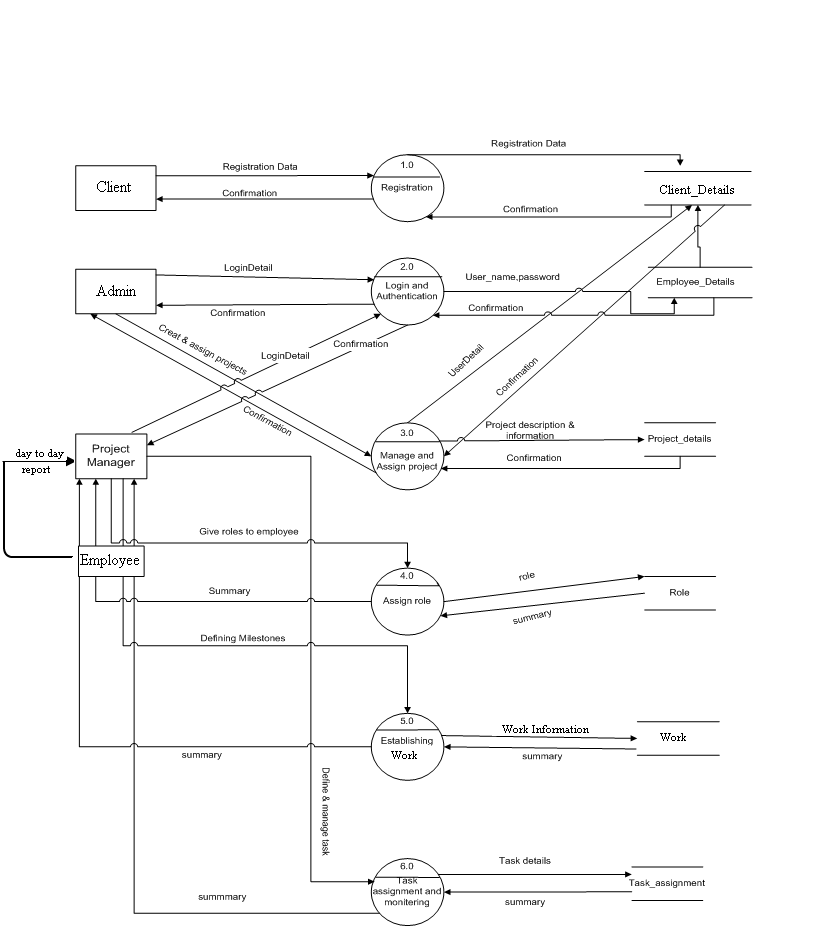
* A Context diagram giving system overview is used as the first step in developing a DFD.
* The Context diagram is expanded into a series of DFD, each describing a specific function.
* This method of top analysis and breaking down DFD to give more and more details is known as leveling.

# Data Flow Diagram:

* Data flow diagrams are used to depict the flow and transformation of data in an information process system.
* A Rectangle to depict an external Entity,
* A Circle to represent a Process,
* Two Parallel lines to represent a file or Data Storage or Repository of data,
* Lines with Arrows to depict Data Flows,
* The Direction of Arrow shows the direction of the flow of Data.
* DFD are used for representing logical processing of data.
* External Entity may be source of input data or destination of results.
* A Data flow can enter or leave a data store.
  + 1. **CONTEXT LEVEL DIAGRAM**

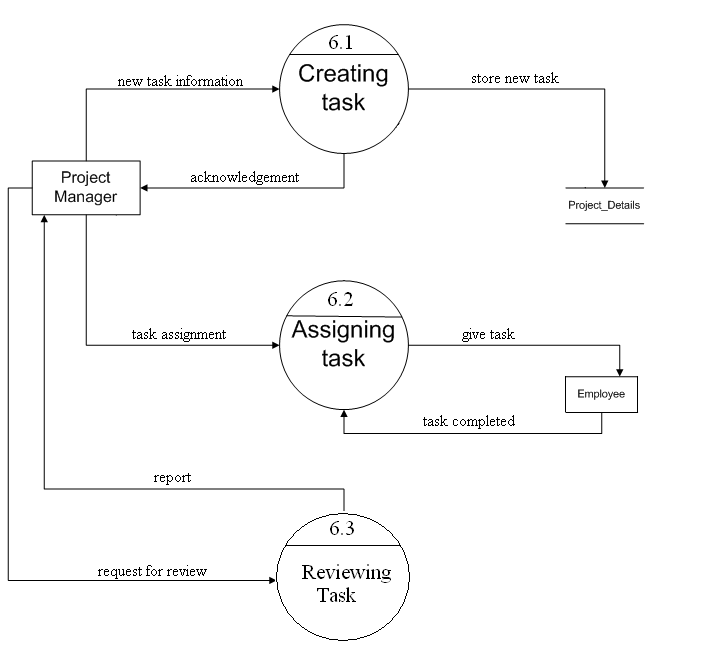


**Figure 4.1 Level-0 DFD**

**4.1.2 FIRST LEVEL DIAGRAM**

**Figure 4.2 Level-1 DFD**

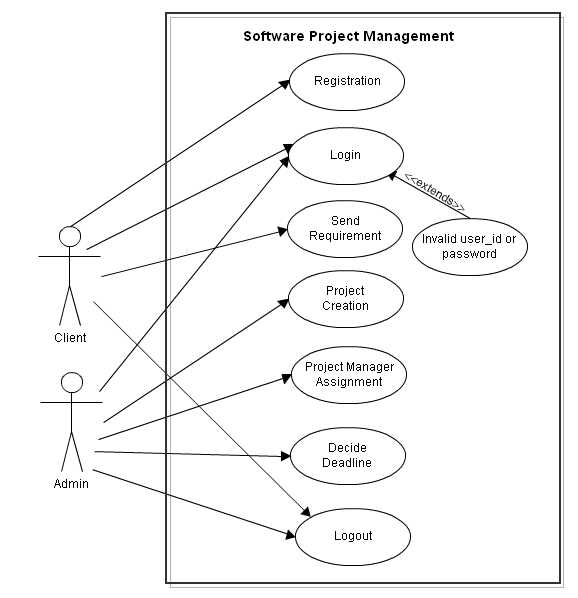
**4.1.3 SECOND LEVEL DIAGRAM**

****

**Figure4.3 Level-2 DFD**

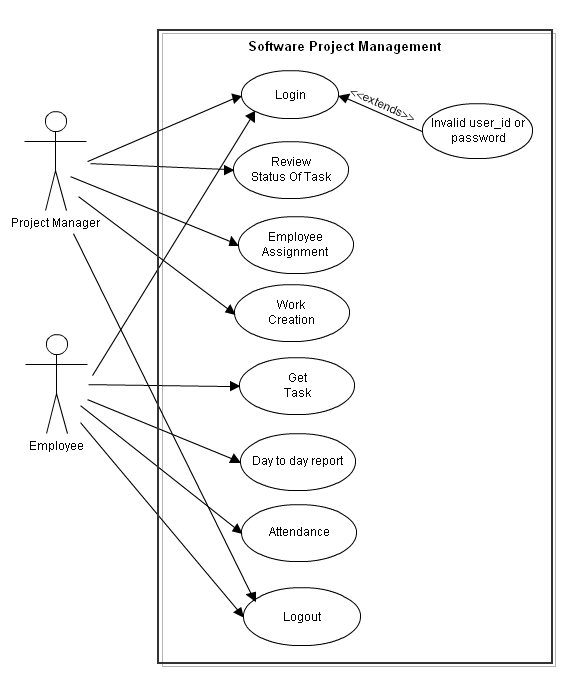
**4.2 USE CASE DIAGRAMS**

**4.2.1 USECASE DIAGRAM- CLIENT,ADMIN**

****

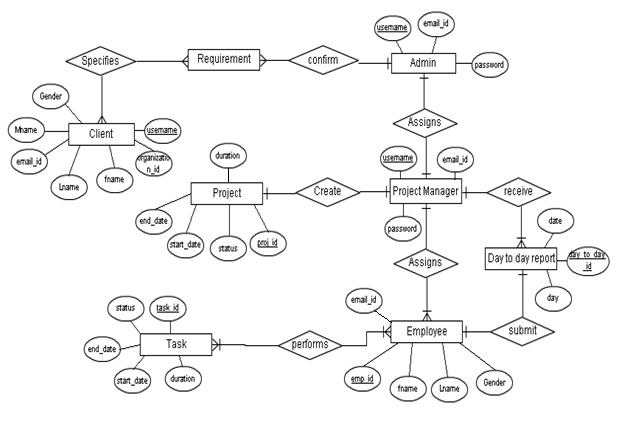
**Figure4.4 Use case Diagram--client,admin**

**4.2.2 USECASE DIAGRAM- PROJECT MANAGER, EMPLOYEE**

****

**Figure 4.5 Use case Diagram-Project Manager, Employee**

**4.3 ER DIAGRAM**

****

**Figure 4.6 ER Diagram**

**Chapter 5: DATA MODELLING AND DESIGN**

# What is Data-Dictionary?

A data dictionary is a catalog a repository of the elements in a system. AS the name suggested, these elements center around data and the way they are structured to meet user requirements and organization needs. In a data dictionary you will find a list of all the elements composing the data flowing through a system. The major elements are data flow, data stores and processes. The data dictionary stores details and descriptions of these elements.

If analysts want to know how many characters are in a data item, by what other names it is referenced in the system, of where it is used in the system, they should be able to fine the answers in a property developed data dictionary.

# Why Data-Dictionary used?

The Data-Dictionary is used for the following purpose...

* Storage of the personal Information
* Storage of user files and directories
* To manage details in large system
* To communicate the features of the system
* To document the featured of the system
* To Facilitate analysis of the details in order to evaluate characteristics and determine where system should be made.
* To locate errors and omissions in the System

**5.1 DATA DICTIONARY**

**ProjectDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **constraint** | **Description** |
| project\_id | int | 200 | Primary key | Project id |
| project\_name | varchar | 300 | Not null | Project Name |
| project\_manager\_id | int | 200 | Foreign Key | Project Manager id |
| Client\_id | int | 200 | Foreign Key | Client id |
| description | varchar | 1000 | Not null | description |
| Expected\_start\_date | Date |  | Not null | Expected start date |
| Final\_start\_date | Date |  | Not null | Final Start date |
| Expected\_end\_date | Date |  | Not null | Expected end Dare of project |
| Final\_end\_date | Date |  | Not null | Final end date |
| Expected\_cost | varchar | 300 | Not null | Expected cost |
| Final\_cost | varchar | 300 | Not null | Final cost |
| Payment\_mode | varchar | 100 | Not Null | Payment Mode |
| Project\_status | varchar | 100 | Not Null | Project Status |

**Table 5.1 ProjectDetail**

**ClientDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraints** | **Description** |
| Client\_id | Auto incremented | 100 | Primary key | Identification number of client |
| Client\_name | varchar | 300 | Not Null | Name of client |
| Company\_name | Varchar | 300 | Not null | Name of company |
| Company\_address | Varchar | 300 | Not null | Address of company |
| Description | Varchar | 1000 | Not null | Details about company |
| Domain | Varchar | 300 | Not null | Work domain |
| email | Varchar | 100 | Not null | Email Id |
| Contact\_no. | Number | 100 | Not null | Contact No |

**Table 5.2 ClientDetail**

**AdminDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field name** | **Data type** | **Size** | **Constraint** | **Description** |
| Admin\_id | Auto incremented | 200 | Primary key | Id of admin |
| Admin\_name | Varchar | 200 | Not null | Name of admin |
| email | Varchar | 200 | Not null | email |
| password | Varchar | 200 | Not null | password |
| User\_type | varchar | 200 | Not null | User type |

**Table 5.3 AdminDetail**

**EmployeeDeatail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraint | Description |
| Employee\_id | int | 100 | Primary Key | Id of employee |
| Employee\_name | varchar | 300 | Not Null | Nmae of employee |
| gender | varchar | 100 | Not Null | gender |
| address | varchar | 300 | Not Null | address |
| email | varchar | 100 | Not Null | Email id |
| Contact\_no | varchar | 200 | Not Null | Contact No |
| Employee\_type | varchar | 100 | Not Null | Employee type |

**Table 5.4 EmployeeDetail**

EmployeeModule

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Size | Constraint | Description |
| Module\_number | varchar | 1000 | Foreign Key | Module Number |
| Employee\_id | int | 200 | Foreign Key | Id of employee |
| Employee\_work\_status | varchar | 300 | Not Null | Work status |

**Table 5.5 EmployeeModule**

**Login**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| Login\_id | Auto Incremented` | 100 | Primary key | Id of login |
| email | Varchar | 200 | Foreign Key | Email |
| password | Varchar | 200 | Foreign key | Password |
| name | Varchar | 200 | Foreign key | Name |
| User\_type | Varchar | 200 | Foreign key | Type of user |

**Table 5.6 Login**

**MessageDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| Message\_id | Auto incremented | 100 | Primary key | Id of Meesage |
| Sender\_id | int | 100 | Foreign key | Id of sender |
| subject | varchar | 200 | Not Null | Subject of message |
| message | Varchar | 200 | Not null | Message |
| Send\_date | Date |  | Not null | Send date |
| Read\_flag | Varchar | 200 | Not null | Read flag |

**Table 5.7 MessageDetail**

**ModuleDetail**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| **Module\_id** | **int** | **100** | **Primary Key** | **Id of module** |
| **Module\_number** | **varchar** | **300** | **Foreign Key** | **Module number** |
| **Module\_name** | **varchar** | **300** | **Not Null** | **Module Name** |
| **Project\_Id** | **int** | **100** | **Foreign Key** | **Project id** |
| **Module\_start\_date** | **Date** |  | **Not Null** | **Start Date** |
| **Module\_end\_date** | **Date** |  | **Not Null** | **End Date** |
| **Module\_status** | **varchar** | **300** | **Not Null** | **Module Status** |

**Table 5.8 ModuleDetail**

**Notification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Size** | **Constraints** | **Description** |
| Notification\_id | int | 100 | Primary Key | Id of notification |
| Login\_id | int | 100 | Foreign Key | Id of login |
| Module\_number | varchar | 300 | Foreign Key | Module Number |
| description | varchar | 300 | Not Null | description |
| Priority | varchar | 300 | Not Null | priority |
| Expiry\_date | Date |  | Not Null | Expiry date |

**Table 5.9 Notification**

**CHAPTER: 6 LIMITATIONS AND**

**FUTERE ENHANCEMENT**

**Limitation:**

* The size of the database increases day-by-day, increasing the load on the database back up and data maintenance activity.
* Training for simple computer operations is necessary for the users working on the system.

**Future Enhancement**

* This System being web-based and an undertaking of Cyber Security Division, needs to be thoroughly tested to find out any security gaps.
* A console for the data center may be made available to allow the personnel to monitor on the sites which were cleared for hosting during a particular period.

The time spent on the project as well as each task can be measured in terms of hours spent on it. Further each task is represented in terms of percentage by which we can know exact status of the task.

The efficiency of the employee can be calculated from the number of tasks, the amount of time spent on those task and errors that occurred in them.

**CHAPTER: 7 CONCLUSION**

* At the end of project management system we can say that due the system the wastage of time and effort put into the work can be minimized.
* The workflow will be automated and there will be a facility of complete information and data storage of assumptions and analysis made to drive out the final project management system.
* **Summary:**

1. We first learned the key requirements of our project and analyze those requirements.
2. We had then collected the existing system problems and try to divide our project into different modules but with less associated problems.
3. We had then design our system according to user requirements.
4. We had prepared different milestones and deliverables.
5. Just now we are doing the implementation of the system.

**CHAPTER: 8 BIBLIOGRAPHY AND REFERENCES**

References:

1 Pressman R.S., Software Engineering: A Practitioner's Approach, TMH.

2 Oriented Modeling and Design with UML second edition by michae Blaha

and James Rambaug.

3 System Analysis and Design- Alias M. Awad

Web-sites:

* [www.wikipedia.org](http://www.wikipedia.org)
* [www.google.com](http://www.google.com)