SYNOPSIS
ON
EMPLOYEE TRACKING SYSTEM

Submitted in partial fulfillment of the requirements for the award of the degree

of

Bachelor of Technology

In

COMPUTER SCIENCE AND ENGINEERING

Project Guide	Submitted by
Mr. Ravikant	Nilesh Kumar (1212710067)
Assistant Professor	Manpreet Singh (1212710050)
	Manish Jaiswal (1212710049)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

IIMT ENGINEERING COLLEGE, MEERUT-250001

(Affiliated to Dr APJ Abdul Kalam Technical University, Lucknow)

June, 2015

ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend my sincere thanks to all of them.

We are highly indebted to our highly respected and esteemed guide **Mr. Ravikant** for his guidance and constant supervision as well as for providing necessary information regarding the project & also for his support in completing the project.

We also wish to express our gratitude to **Mr. Harshbardhan Barik, HOD, Computer Science and Engineering** for his kind hearted support. We are also grateful to our faculty **Mr. Dheeraj Churamani** for his constant support and guidance.

We would like to express our gratitude towards our parents & members of **IIMT Engineering College, 127, Meerut**, for their kind co-operation and encouragement which help us in completion of this project.

At the end we would like to express our sincere thanks to all our friends and others who helped us directly or indirectly during this project work.

PROJECT ASSOCIATES:-

Nilesh Kumar (1212710067)

Manpreet Singh (1212710050)

Manish Jaiswal (1212710049)

INDEX

Chapter 1

INTRODUCTION

The **Employee Tracking System** is the systematic and the additional method of the employee management for the use of the organization. This is very helpful and highly useful. The system tracks the job of the staff and worker who are doing the job separately and in team for the development of the organization. The word the Employee Tracking have now taken over to the Personnel Management by the term mentions the management of the organizations people. Normally the Human Resource Development refers to the recruiting of candidates, to enhance their skills, using, managing and providing the proper wages according to their job and the need

Here in this **Employee Tracking System Project** the employees when enter in to an organization he starts working on a project or starts doing his task then he gives his detailed time to time work he has done on that particular day to the administrator who in turn prepare the reports of a particular employee and then submits to the project manager, then he analyzes the capability of an employee and he will finally prepare the final reports which include the aspects of an employee are as follows:

- The reports will give the significant amount of time and effort invested by the employees time to time, by helping the management to know about their employee's capability.
- The employee can immediately know his capacity and the working hours weekly once or when the task allotted to him is finished.
- Achievement/target report employee wise.
- Performance indicator for an employee.
- Determining the efficiency and the task completed by the employee

Finally I conclude that this project plays an important role in tracking the timings of the employee and how much time he is working and the entry time and exit time .When a employee cross the gate number of times. And maintain like a small database about the timings of the employee.

Chapter 2

SURVEY

INTRODUCTION

This chapter summarizes the evaluation of the literature relevant to the Employee Tracking System. It examines theories, concepts, approaches, methods and techniques relevant to the project. Similar existing technologies relating to the development the EMS are discussed.

LITERATURE REVIEWS ON TOPICS RELATED TO THE PROJECT

An organization or company with a very large number of employees manages a greater volume of data. This activity can be daunting without a more sophisticated tool to store and retrieve data. The various levels of sophistication can be examined by looking at the evolutionary aspects of HR technology. These aspects can be characterized into four stages of development: Paper-based systems, early personal computer (PC) technology, electronic databases, and Web-based technology.The benefits of automation are becoming widely known to HR and other areas of the business. The focus has shifted to automating as many transactions as possible to achieve effectiveness and efficiencies. The technology of the future will be about speedy access to accurate current information, and reliability to access this information via multiple systems will give organizations a strategic edge. HR is expected to relinquish its role as sole owner of HR information, so that managers and employees can use this information to solve their own problems using Web-based systems. This new system will not necessarily mean reduction in HR staff. The new system will enable HR professionals to focus on transforming information into knowledge that can be used by the organization for decision making; it will be about HR and IT working together to leverage this technology. A recent study by the Hackett Group, a business process advisory firm found that

High-performing organizations spend 25 percent less than their peers on HR because they use technology effectively. The two most popular Web-based HR applications used today are self-service for employees and self-service for managers. These applications have enabled companies to shift responsibility for viewing and updating records onto individual employees and have fundamentally changed the manner in which employees acquire information and relate to their HR departments.

REVIEWS ON SEVERAL SOFTWARE METHODOLOGIES

A software development methodology is a collection of procedures, techniques, tools, and documentation aids which will help the systems developers in their efforts to implement a new information system. There are a number of software development methodology each of which are adopted based on a number of factors relating to the project e.g. Time, cost, incorporation of requirement changes during the development process, system complexity, communication between customers and developers, software criticality, size of the development team. These generic models are not definitive descriptions of software processes. Rather, they are abstractions of the process that can be used to explain different approaches to software development. You can think of them as process frameworks that may be extended and adapted to create more specific software engineering processes. Below are a selected number of models:

The Waterfall Model

The waterfall model is a sequential design process, often used in software development processes. It takes the fundamental process activities of specification, development, validation, and evolution and represents them as separate process phases such as requirements specification, software design, implementation, testing, and so on.

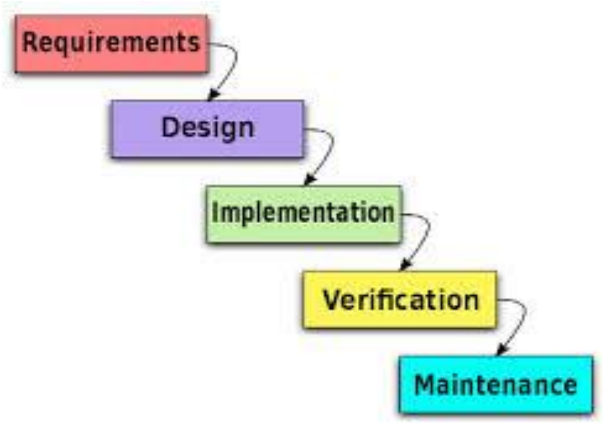


Fig: Waterfall Model

Incremental Model

This approach interleaves the activities of specification, development, and validation. The system is developed as a series of versions (increments), with each version adding functionality to the previous version.

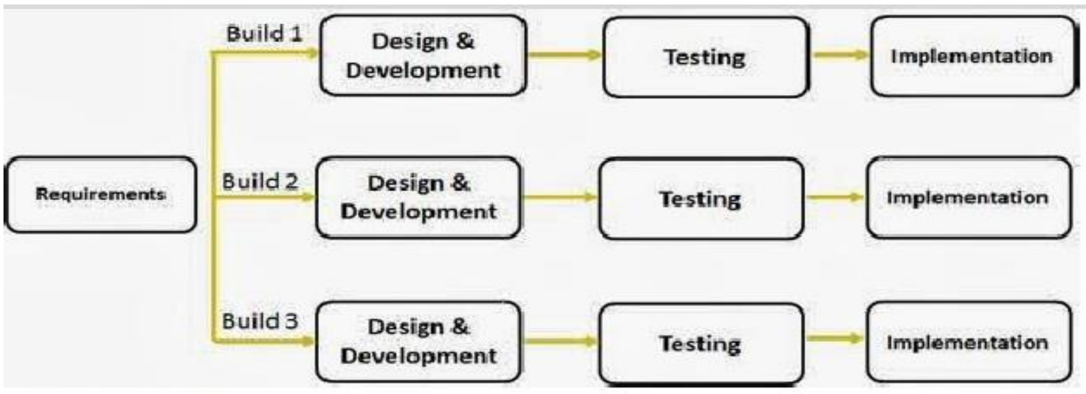


Fig: Incremental Model

Chapter 3

PROBLEM DESCRIPTION

PROBLEM STATEMENT

Manual handling of employee information poses a number of challenges. This is evident in procedures such as leave management where an employee is required to fill in a form which may take several weeks or months to be approved. The use of paper work in handling some of these processes could lead to human error, papers may end up in the wrong hands and not forgetting the fact that this is time consuming. A number of current systems lack employee self-service meaning employees are not able to access and manage their personal information directly without having to go through their HR departments or their managers.

Another challenge is that multi-national companies will have all the employee information stored at the headquarters of the company making it difficult to access the employee information from remote places when needed at short notice. The aforementioned problems can be tackled by designing and implementing a web based HR management system. This system will maintain employee information in a database by fully privacy and authority access. The project is aimed at setting up employee information system about the status of the employee, the educational background and the work experience in order to help monitor the performance and achievements of the employee through a password protected system.

PROJECT BACKGROUND

Employees are the backbone of any company therefore their management plays a major role in deciding the success of an organization. Human Resource Management Software makes it easy for the employer to keep track of all records. This software allows the administrator to edit employees, Add new employees as well as evaluate an employee’s performance. Employees can be managed efficiently without having to retype back their information in the database. You can check to see if there are duplicate positions/employees in the database.

A flexible and easy to use Employee Tracking System solution for small and medium sized companies provides modules for personnel information management thereby organization and companies are able to manage the crucial organization asset.

The combination of these modules into one application assures the perfect platform for re-engineering and aligning

Human Resource processes along with the organizational goals. This system brings about an easy way of maintaining the details of employees working in any organization. It is simple to understand and can be used by anyone who is not even familiar with simple employees system. It is user friendly and just asks the user to follow step by step operations by giving easy to follow options. It is fast and can perform many operations for a company.

Chapter 4

APPROACH USED

The Object Oriented Approach

Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which are data structures that contain data, in the form of fields, often known as *attributes;* and code, in the form of procedures, often known as *methods.* A distinguishing feature of objects is that an object's procedures can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OO programming, computer programs are designed by making them out of objects that interact with one another. There is significant diversity in object-oriented programming, but most popular languages are class-based, meaning that objects are instances of classes, which typically also determines their type.

The key ideas of the object oriented approach are:

- Objects
- Class
- Data abstraction and encapsulation
- Inheritance
- Polymorphism

Object

Objects - structures that contain both data and procedures

Objects sometimes correspond to things found in the real world. For example, a graphics program may have objects such as "circle," "square," "menu." An online shopping system might have objects such as "shopping cart," "customer," and "product".

Each object is said to be an instance of a particular class (for example, an object with its name field set to "Mary" might be an instance of class Employee). Procedures in object-oriented programming are known as methods; variables are also known as fields, members, attributes, or properties.

Class

In object-oriented programming, a class is an extensible program-code-template for creating objects, providing initial values for state (member variables) and implementations of behaviour (member functions or methods).In many languages, the class name is used as the name for the class (the template itself), the name for the default constructor of the class (a subroutine that creates objects), and as the type of objects generated by instantiating the class; these distinct concepts are easily conflated.

When an object is created by a constructor of the class, the resulting object is called an instance of the class, and the member variables specific to the object are called instance variables, to contrast with the class variables shared across the class.

Data abstraction and encapsulation

The wrapping up of data and function into a single unit (called class) is known as encapsulation. Data **encapsulation** is the most striking feature of class. Abstraction refers to the act of representing essential feature without including the background details or explanation. Data abstraction is the simplest of principles to understand. Data abstraction and encapsulation are closely tied together, because a simple definition of data abstraction is the development of classes, objects, types in terms of their interfaces and functionality, instead of their implementation details. Abstraction denotes a model, a view, or some other focused representation for an actual item. It's the development of a software object to represent an object we can find in the real world. Encapsulation hides the details of that implementation.

If a class disallows calling code from accessing internal object data and forces access through methods only, this is a strong form of abstraction or information hiding known as encapsulation. Some languages (Java, for example) let classes enforce access restrictions explicitly, for example denoting internal data with the private keyword and designating methods intended for use by code outside the class with the public keyword. Methods may also be designed public, private, or intermediate levels such as protected (which typically allows access from other objects of the same class, but not objects of a different class). In other languages (like Python) this is enforced only by convention (for example, naming "private" methods starting with an underscore). This is useful because it prevents the external code from being concerned with the internal workings of an object. This facilitates code refactoring, for example allowing the author of the class to change how objects of that class represent their data internally without changing any external code (as long as "public" method calls work the same way). It also encourages programmers to put all the code that is concerned with a certain set of data in the same class, which organizes it for easy comprehension by other programmers. Encapsulation is often used as a technique for encouraging decoupling.

Inheritance

Inheritance (OOP) is when an object or class is based on another object (prototypal inheritance) or class (class-based inheritance), using the same implementation (inheriting from an object or class) specifying implementation to maintain the same behaviour (realizing an interface; inheriting behaviour). It is a mechanism for code reuse and to allow independent extensions of the original software via public classes and interfaces. The relationships of objects or classes through inheritance give rise to a hierarchy. Inheritance was invented in 1967 for Simula.

Inheritance should not be confused with subtyping. In some languages inheritance and subtyping agree, while in others they differ; in general subtyping establishes an is-a relationship, while inheritance only reuses implementation and establishes a syntactic relationship, not necessarily a semantic relationship (inheritance does not ensure behavioural subtyping). To distinguish these concepts, subtyping is also known as interface inheritance, while inheritance as defined here is known as implementation inheritance or code inheritance.Still, inheritance is a commonly used mechanism for establishing subtype relationships.

Inheritance is contrasted with object composition, where one object contains another object (or objects of one class contain objects of another class); see composition over inheritance. Composition implements a has-a relationship, in contrast to the is-a relationship of subtyping.

Polymorphism

In object-oriented programming, polymorphism refers to a programming language's ability to process objects differently depending on their data type or class. More specifically, it is the ability to redefine methods for derived classes. For example, given a base class shape, polymorphism enables the programmer to define different area methods for any number of derived classes, such as circles, rectangles and triangles. No matter what shape an object is, applying the area method to it will return the correct results. Polymorphism is considered to be a requirement of any true object-oriented programming language (OOPL).

There are several fundamentally different kinds of polymorphism:

Ad hoc polymorphism: when a function denotes different and potentially heterogeneous implementations depending on a limited range of individually specified types and combinations. Ad hoc polymorphism is supported in many languages using function overloading.

Parametric polymorphism: when code is written without mention of any specific type and thus can be used transparently with any number of new types. In the object-oriented programming community, this is often known as generics or generic programming. In the functional programming community, this is often simply called polymorphism.

Subtyping (also called subtype polymorphism or inclusion polymorphism): when a name denotes instances of many different classes related by some common superclass. In the object-oriented programming community, this is often simply referred to as polymorphism.

Chapter 5

SYSTEM REQUIREMENT'S

Hardware Requirement:-

- At least 128/256 MB of RAM
- Disk Drive 250 MB (required for installation)
- Disk Space – 500 MB (required for running)

Software Requirements:-

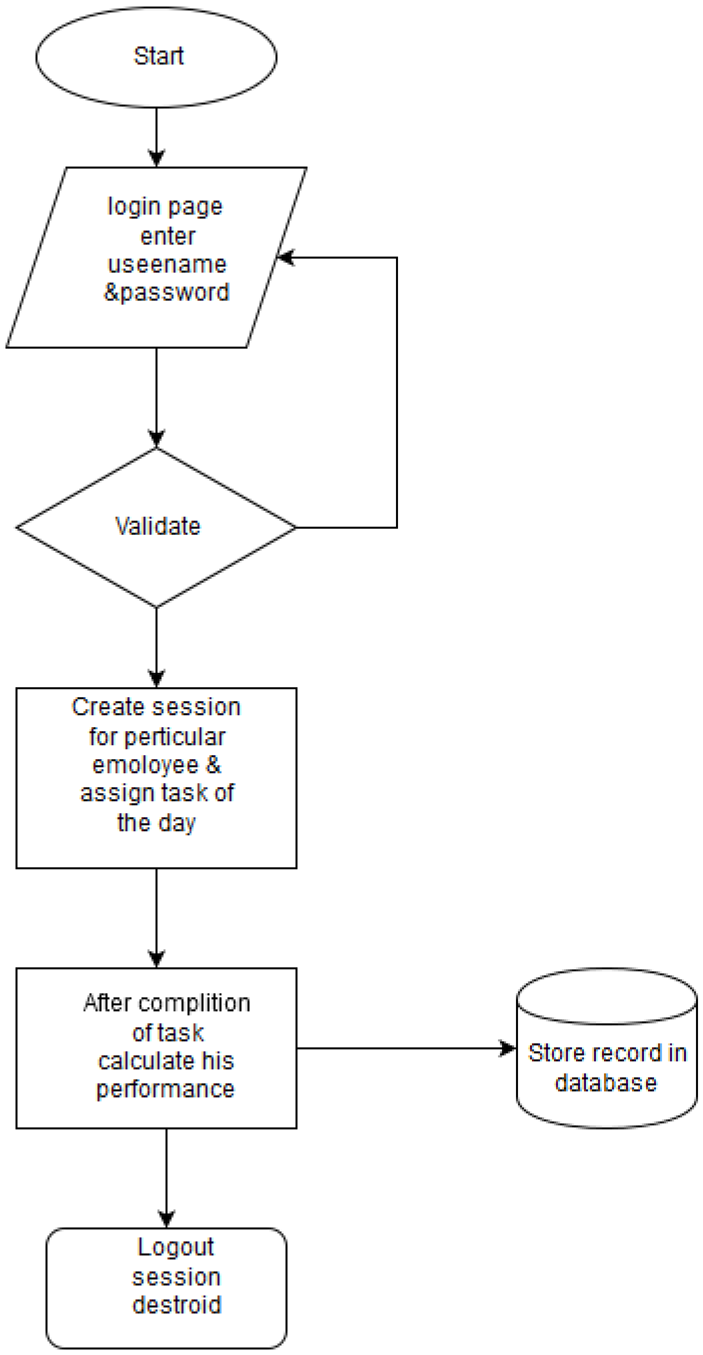
- Any Version of Microsoft Windows (Higher than 98)/ Linux (All version)
- NetBeans 8.1
- Oracle 10g

Chapter 6

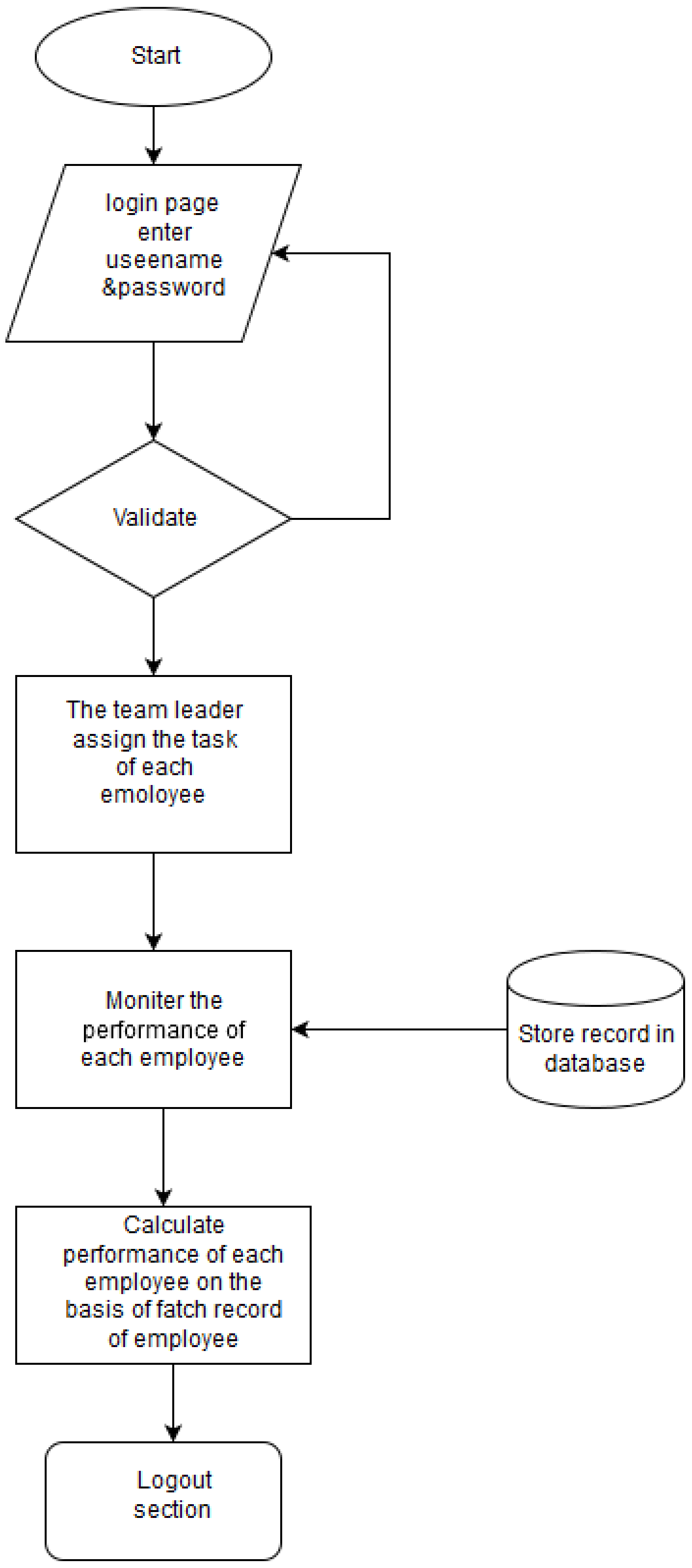
FLOW CHART

In this project **Employee tracking system** there is two view employee view and team leader view.

1. Employee view


Start
login page enter useename &password
Validate
Create session for perticular emoloyee & assign task of the day
After complition of task calculate his performance
Store record in database
Logout session destroid

2. Team leader view


Start
login page enter useename &password
Validate
The team leader assign the task of each emoloyee
Moniter the performance of each employee
Store record in database
Calculate performance of each employee on the basis of fatch record of employee
Logout section

Chapter 7

FUTURE WORK

SCOPE OF FUTURE APPLICATIONS

1. This GUI Application can be further modify to a large scale organization.
2. Each module of the application can be further modified & extended. Such as recruitment, planning, people management, payroll, appraisal, **employee tracking policies**.
3. For a business application it can be further modified to a web application.

Chapter 8

BIBLIOGRAPHY

References:-

1. **https://docs.oracle.com**
2. **Software engineering: A practitioner's approach/Roger S.Pressman.-5th edition.**
3. **https://www.draw.io/**
4. **https://en.wikipedia.org**