**1.1 OBJECTIVE**

Payroll system is the heart of any Human Resource System of an organization. The solution has to take care of calculation of salary as per rules of the company, income tax calculations and various deductions to be done from the salary. It has to generate a pay-slip.

It is understood that we are tired of managing thousands of odd papers, pay slips, payroll reports and salary details and so on. Imagine that we have a payroll processing system that will generate our pay slips within seconds.

We can help others automate their payrolls by developing a customized payroll application that suits your specific requirements.

Main aim of developing an Employee Payroll Management is to provide an easy way not only to automate all functionalities involved managing leaves and payroll for the employees of the organization, but also to provide full functional reports to management of the company with details about usage of leave facility and salaries paid or to be paid to employees.

We are committed to bring the best way of management in the various forms of EPM. We understand that EPM is not just a product to be sold; it is a tool to manage the inner operation of Company related to employee leave and payroll.

**1.2 PROJECT DESCRIPTION**

The Online Payroll System is a web-based solution on PHP platform to ease calculation of salary as per rules of the company, income tax calculations and various deductions to be done from the salary.

All the data such as, employee information, salaries and tax will be stored in database as a track record.

The application works in multiple PC’s installed on multiple Computers by sharing same database by which users of different department which will be located in different places and by keeping domain of application as online.

It will improve the efficiency of the payroll system by allowing us to quickly find the information of an employee detail. It will provide easy and faster access information and an user-friendly environment.

Both the employee and the administrator need to login to access the system. This helps in the authentication process and also ensures that access is given only to the authenticated users. The administrator has the rights to issue salaries to the employees, add or remove employees and many other editing capabilities.

The employee can view pay slips, edit his details and run various queries on the database.

Both the administrator and the employee need to logout once they have accessed the system so that security of the system is not breached.

**1.3 DEFINITIONS AND ABBREVIATIONS**

***PHP*** *is a server-side scripting language designed for web development but also used as a general-purpose programming language.*

**MySQL**, *the most popular Open Source SQL database management system, is developed, distributed, and supported by Oracle Corporation.*

**SRS** - *Software Requirement Specifications*

**EPM** - *Employee Payroll Management*

**2.1 EXISTING AND PROPOSED SYSTEM**

The existing system worked as follows:

The administrators use MS Excel, and maintain their records; however it is not possible for them to share the data from multiple systems in multi user environment. There is lot of duplicate work, and chances of mistake. When the records are changed they need to update each and every excel file. There is no option to find and print previously saved records. There is no security; anybody can access any report and sensitive data, also no reports to summary report. This Online Payroll System is used to overcome the entire problem, which they are facing currently, and making complete atomization of manual system to computerized system.

The problems associated with the existing system are:

* The existing system only provides text-based interface, which is not as user-friendly as Graphical User Interface.
* Since the system is implemented in manual, so the response is very slow.
* The transactions are executed in off-line mode, hence on-line data capture and modification is not possible.
* Off-line reports cannot be generated due to batch mode execution.

Hence, there is a need of reformation of the system with more advantages and flexibility. The Online Payroll System eliminates most of the limitations of the existing software. It has the following objectives:

* **Enhancement**

The main objective of Online Payroll System is to enhance and upgrade the existing system by increasing its efficiency and effectiveness. The software improves the working methods by replacing the existing manual system with the computer-based system.

* **Automation**

The Online Payroll System automates each and every activity of the manual system and increases its throughput. Thus the response time of the system is very less and it works very fast.

* A**ccuracy**

The Online Payroll System provides the user a quick response with very accurate information regarding the users etc.

* **User-Friendly**

The Online Payroll System has a very user-friendly interface. Thus the users will find it very easy to work on it. The system provides accuracy along with a pleasant interface. It also makes the present manual system more interactive, speedy and user friendly.

* **Availability**

The transaction reports of the system can be retrieved as and when required. Thus, there is no delay in the availability of any information, whatever needed, can be captured very quickly and easily.

* **Maintenance Cost**

Reduce the cost of maintenance.

**2.2 FEASIBILITY STUDY**

The proposed Online Payroll System is much more feasible over the exiting traditional system of maintaining the payroll system and the transaction processing system manually or through spreadsheets by the following factors:

* **Attractive Dashboard**

The dashboard of both the administrator and the employee are very attractive so as to make it easy for the users to use it

* **Administrator Module**

The administrator module of the system is interactive and provides features to maintain basic information of his employees such as name, address, date of birth, joining date, social security number etc. An administrator will be assigned a login and password for the system. After authentication he/she will be the only person authorized to add and edit the basic information of the employees. He/she will be able to calculate the salaries of the particular employees on the basis of given parameters.

* **Employee Module**

The employee module of the system is also interactive and provides features to update information of themselves such as address, profile pictures etc. An employee will be assigned an employee no. by the administrator. He/she will be able to view their and run other queries.

* **Payslip Module**

The payslip is the main output of the system, it provides with a printable payslip of a particular month with complete details of basic, deductions etc.

**2.3 METHODOLOGY USED**

The most suitable and appropriate software development methodology I found for the software development process is waterfall software development model, which gives a clear view about the software and helped me to achieve my goal.

Small and medium size software is usually divided into six stages that are interrelated with each other in a top-down approach named as waterfall. The input of one specific stage is the output of previous stage, which initializes the next stage. At every step or stage of the model some additional information is also added up into the input of that stage and generating the results. But that additional information is restricted in scope and previous stages are directly traceable from that stage.

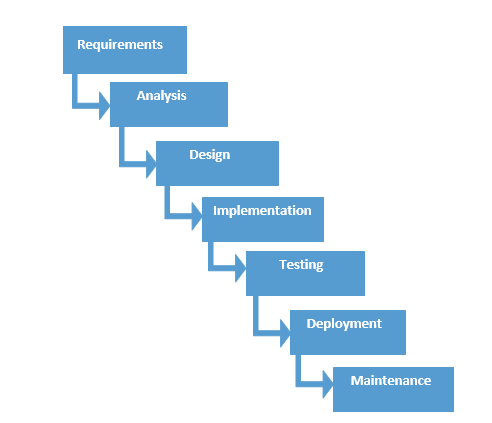
The first stage of the waterfall model is planning stage in which the most critical task is to clearly mention the high-level requirements or goals of the software. It helps to make feasibility and risks associated with the project and also provide the basic project structure.

The next stage is the requirement definition stage that takes goals as input from the planning stage and then these goals are defined into a set of one or more requirements. Major functionalities, initial data entities and the operational data areas are defined under this stage.

The input of design stage is the output of the requirement stage, which is approved requirements documentation. In this stage the design elements are defined with the help of interviews, prototypes and workshops conducted. These design elements consist of functional hierarchy, business process diagrams, pseudo code and entity-relationship diagram. These design elements provide detail description about the software and each element is related to a specific requirement.

The development stage is initialized by the previous design stage. At this stage the code for the design elements of software is written and it provides the functional software components.

Integration and test stage is provided the output of the development stage. At this stage the different software functioning components developed in the previous stage are integrated with each other to provide full fledge software project, which is providing all the high-level requirements. This software is also passed through different test cases to check the validity, correctness, and completeness and hence enables us to achieve our goals regarding our project.

Finally, the installation and acceptance stage arrives. The software is loaded to the server at site of customers and tested with different test cases once again to check the correct working. If all this shows adequate results satisfying the customer then the software is handed over to the customer formally.

**Fig 2.1: Waterfall Model**

The reason to choose this model was that

1. In this model all the phases are in a sequence and are dependent with one another, therefore a phase cannot be started until the previous phase is completed and fully documented. This approach is most appropriate for our project, because all the requirements and goals of our project are very clear.
2. Secondly, it is easy to do work in components and waterfall model is providing this approach. After the completion of all the phases individually, they are integrated together. This model is very economical and risk free due to its sequential approach.

**2.4 TOOLS AND TECHNOLOGIES USED**

The Tools and Technologies which are used in this project are:

* **PHP**

PHP is a [server-side scripting](https://en.wikipedia.org/wiki/Server-side_scripting) language designed for [web development](https://en.wikipedia.org/wiki/Web_development) but also used as a [general-purpose programming language](https://en.wikipedia.org/wiki/General-purpose_programming_language). PHP originally stood for Personal Home Page, but it now stands for the [recursive](https://en.wikipedia.org/wiki/Recursive_acronym) acronym PHP: Hypertext Preprocessor. Fast, flexible and pragmatic, PHP powers everything from blogs to the most popular websites in the world. PHP code may be embedded into [HTML](https://en.wikipedia.org/wiki/HTML) code, or it can be used in combination with various [web template systems](https://en.wikipedia.org/wiki/Web_template_system), web content management system and [web frameworks](https://en.wikipedia.org/wiki/Web_framework). PHP code is usually processed by a PHP [interpreter](https://en.wikipedia.org/wiki/Interpreter_(computing)) implemented as a [module](https://en.wikipedia.org/wiki/Plugin_(computing)) in the web server or as a [Common Gateway Interface](https://en.wikipedia.org/wiki/Common_Gateway_Interface) (CGI) executable. The web server combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a [command-line interface](https://en.wikipedia.org/wiki/Command-line_interface) (CLI) and can be used to implement [standalone](https://en.wikipedia.org/wiki/Computer_software) [graphical applications](https://en.wikipedia.org/wiki/Graphical_user_interface).

* **MySQL**

MySQL is an [open-source](https://en.wikipedia.org/wiki/Open-source) [relational database management system](https://en.wikipedia.org/wiki/Relational_database_management_system) (RDBMS). MySQL is a popular choice of database for use in web applications, and is a central component of the widely used [LAMP](https://en.wikipedia.org/wiki/LAMP_(software_bundle)) open-source web application software stack. LAMP is an acronym for "[Linux](https://en.wikipedia.org/wiki/Linux), [Apache](https://en.wikipedia.org/wiki/Apache_HTTP_Server), MySQL, [Perl](https://en.wikipedia.org/wiki/Perl)/ [PHP](https://en.wikipedia.org/wiki/PHP)/ [Python](https://en.wikipedia.org/wiki/Python_(programming_language))". [Free-software](https://en.wikipedia.org/wiki/Free_software) open-source projects that require a full-featured database management system often use MySQL. MySQL is also used in many high-profile, large-scale [websites](https://en.wikipedia.org/wiki/Website), including [Google](https://en.wikipedia.org/wiki/Google) (though not for searches), [Facebook](https://en.wikipedia.org/wiki/Facebook), [Twitter](https://en.wikipedia.org/wiki/Twitter), [Flickr](https://en.wikipedia.org/wiki/Flickr), and [YouTube](https://en.wikipedia.org/wiki/YouTube).

* **HTML**

HyperText Markup Language, commonly abbreviated as HTML, is the standard [markup language](https://en.wikipedia.org/wiki/Markup_language) used to create [web pages](https://en.wikipedia.org/wiki/Web_page). Along with [CSS](https://en.wikipedia.org/wiki/Cascading_Style_Sheets), and [JavaScript](https://en.wikipedia.org/wiki/JavaScript), HTML is a cornerstone technology used to create web pages, as well as to create user interfaces for mobile and [web applications](https://en.wikipedia.org/wiki/Web_applications). [Web browsers](https://en.wikipedia.org/wiki/Web_browser) can read HTML files and render them into visible or audible web pages. HTML describes the structure of a [website](https://en.wikipedia.org/wiki/Website) [semantically](https://en.wikipedia.org/wiki/Semantic) and, before the advent of Cascading Style Sheets (CSS), included cues for the presentation or appearance of the document (web page), making it a markup language, rather than a [programming language](https://en.wikipedia.org/wiki/Programming_language). HTML can embed [scripts](https://en.wikipedia.org/wiki/Scripting_language) written in languages such as [JavaScript](https://en.wikipedia.org/wiki/JavaScript) which affect the behavior of HTML web pages. HTML markup can also refer the browser to [Cascading Style Sheets](https://en.wikipedia.org/wiki/Cascading_Style_Sheets) (CSS) to define the look and layout of text and other material.

* **jQuery**

jQuery is a [cross-platform](https://en.wikipedia.org/wiki/Cross-platform) [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) designed to simplify the [client-side scripting](https://en.wikipedia.org/wiki/Client-side_scripting) of HTML. jQuery is the most popular [JavaScript library](https://en.wikipedia.org/wiki/JavaScript_library) in use today, with installation on 65% of the top 10 million highest-trafficked sites on the Web. jQuery is [free, open-source software](https://en.wikipedia.org/wiki/Free_and_open_source_software) licensed under the [MIT License](https://en.wikipedia.org/wiki/MIT_License). jQuery's syntax is designed to make it easier to navigate a document, select [DOM](https://en.wikipedia.org/wiki/Document_Object_Model) elements, create [animations](https://en.wikipedia.org/wiki/Animation), handle [events](https://en.wikipedia.org/wiki/Event_(computing)), and develop [Ajax](https://en.wikipedia.org/wiki/Ajax_(programming)) applications. jQuery also provides capabilities for developers to create [plug-ins](https://en.wikipedia.org/wiki/Plug-in_(computing)) on top of the JavaScript library. This enables developers to create [abstractions](https://en.wikipedia.org/wiki/Abstraction_(computer_science)) for low-level interaction and animation, advanced effects and high-level, theme-able widgets. The modular approach to the jQuery library allows the creation of powerful [dynamic web pages](https://en.wikipedia.org/wiki/Dynamic_web_page) and Web applications.

* **phpMyAdmin**

phpMyAdmin is a [free and open source](https://en.wikipedia.org/wiki/Free_and_open_source) tool written in [PHP](https://en.wikipedia.org/wiki/PHP) intended to handle the administration of [MySQL](https://en.wikipedia.org/wiki/MySQL) or [MariaDB](https://en.wikipedia.org/wiki/MariaDB) with the use of a [web browser](https://en.wikipedia.org/wiki/Web_browser). It can perform various tasks such as creating, modifying or deleting [databases](https://en.wikipedia.org/wiki/Database), [tables](https://en.wikipedia.org/wiki/Table_(database)), [fields](https://en.wikipedia.org/wiki/Field_(computer_science)) or [rows](https://en.wikipedia.org/wiki/Row_(database)); executing [SQL](https://en.wikipedia.org/wiki/SQL) statements; or managing users and permissions.

**2.5 HARDWARE AND SOFTWARE REQUIREMENTS**

The Hardware Requirement for the project are:

* RAM - 1GB or more
* Hard Disk - 20GB or more
* Processor - Dual Core Processor
* Network connection

The Software Requirement for the project are:

* Front End: PHP
* Back End: Database in MySQL
* Operating System: Any
* Web Technologies: HTML, jQuery
* Browser: Any(Google Chrome Recommended)
* Documentation Tool: Microsoft Word

**2.6 MODULES**

**Masters:**

This module helps the administrator to enter the designation and related description. It also helps to add the department.

**Employee:**

This module helps to add the details of the employee like the personal detail and the employee detail.

**Search:**

This module helps to search the employee details department wise and designation wise.

**Salary:**

This module helps to calculate the salary by adding the allowances and the basic salary and by deductions based on the leaves. It also generates the employee pay slip.

**3.1 USERS**

A system cannot be designed in isolation without the active involvement of the user. There are 2 types of users in Online Payroll System: Administrator and Employee. Both the administrator and the employee need to login to access the system.

The administrator has a different interface as compared to the employee. The administrator has the facility for adding and editing employees, adding and editing payslips, viewing reports, generating payslips, etc.

The employee avails the facility of viewing payslips and running various queries, changing their custom details, etc.

The constant interaction between the administrator and the employee becomes the driving force for the system. These two type of users act as the base to provide automation to the existing system involving the use of human labor or excel sheets.

**3.2 FUNCTIONAL REQUIREMENTS**

The functional requirements of the system include:

* The System aims at providing an efficient interface to the user for managing of payslips and processing salaries, it shall also provide the user varied options for managing the employees through various functions at hand.
* The design is such that the user does not have to manually update the system every time.
* The System calculates the salaries automatically from basic and deductions. Therefore the user never has to worry about manually calculating the estimated usage of the products as the System does it for the user.
* The simple interface of the System has functions like adding a employee or issuing salary, removing or updating the employee details and salaries.

**3.3 NON-FUNCTIONAL REQUIREMENTS**

There are several non-functional requirements such as usability, reliability, performance, supportability, implementation and interfacing related to the system.

**3.3.1. Usability**

* The system must be easy to use by both administrator and employee such that they do not need to read an extensive amount of manuals.
* The system must be quickly accessible by both administrator and employee.
* The system must be intuitive and simple in the way it displays all relevant data and relationships.
* The modules of the system must be easily navigable by the users with buttons that are easy to understand.

**3.3.2. Reliability**

* The System must give accurate salary details to the user.
* The System must successfully add any new employee given by the administrator.
* The system must provide a password-enabled login to the user to avoid any foreign entity changing the data in the system.
* The system should provide the user updates on completion of requested transactions and if the requested transactions fail, it should provide the user the reason for the failure.
* The system should not update the data in any database for any failed transaction.

**3.3.3. Performance**

* The system must not lag, because the users using it don’t have down-time to wait for it to complete an action.
* The system must complete updating the databases, successfully every time the user requests such an action.
* All the functions of the system must be available to the user every time the system is turned on.
* The calculations performed by the system must comply according to the norms set by the user and should not vary unless explicitly changed by the user.

**3.3.4. Supportability**

* The software is designed such that it works even on systems having the minimum configuration.
* The system is adaptable even if additional plug-ins or modules are added at a later point.

**3.3.5. Implementation**

* The System User Interface is built on PHP.
* The Programming is done using jQuery and JavaScript.
* The Database is implemented on PHPMyAdmin.

**3.3.6. Interfacing**

* The system must offer an easy and simple way of viewing the current inventory.
* The system must be able to display the relationships between employees, administrator and salaries in an intuitive manner.

**4.1 SYSTEM PERSPECTIVE**

The systems objectives outlined during the feasibility study serve as the basic from which the work of system design is initiated. Much of the activities involved at this stage are of technical nature requiring a certain degree of experience in designing systems, sound knowledge of computer related technology and through understanding of computers available in the market and the various facilities provided by the vendors.

Nevertheless, a system cannot be designed in isolation without the active involvement of the user. The user has a vital role to play at this stage too. As we know that data collected during feasibility study will be utilized systematically during the system design. It should, however be kept in mind that detailed study of the existing system is not necessarily over with the completion of the feasibility study. Depending on the plan of feasibility study, the level of detailed study will vary and the system design stage will also vary in the amount of investigation that still needs to be done. This investigation is generally an urgent activity during the system. Sometimes, but rarely, this investigation may form a separate stage between feasibility study and computer system design.

Designing a new system is a creative process, which calls for logical as well as lateral thinking. The logical approach involves systematic moves towards the end product keeping in mind the capabilities of the personnel and the equipment at each decision making step. Lateral thought implies encompassing of ideas beyond the usual functions and equipment. This is to ensure that no efforts are being made to fit previous solutions into new situations.

The system design process is not a step-by-step adherence of clear procedures and guidelines. Though, certain clear procedures and guidelines have emerged in recent days, but still much of design work depends on knowledge and experience of the designer. When designer starts working on system design, he/she will face different type of problems. Many of these will be due to constraints imposed by the user or limitations of the hardware and software available in the market. Sometimes, it is difficult to enumerate the complexity of the problems and solutions thereof since the variety of likely problems is so great and no solutions are exactly similar. However, following considerations should be kept in mind during the system designing phase:

* **Practicality**

The system must be stable and can be operated by people with average.

* **Efficiency**

This involves accuracy, timeliness and comprehensiveness to the system output.

* **Cost**

It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy all the requirements.

* **Flexibility**

The system should be modifiable depending on the changing needs of the user. Such modifications should not entail extensive reconstructing or recreation of software. It should also be portable to different computer systems.

* **Security**

This is very important aspect of the design and should cover areas of hardware reliability, fall back procedures, physical security of data and provision for detection of fraud and abuse.

The major system design activities include:

* **Database design**

This activity deals with the design of the physical database. A key is to determine how the access paths are to be implemented.

* **Design Process**

The computer system design process is an exercise of specifying how, the system will work. It is an iterative process, which is based on what the system will be do as shown in the feasibility report.

* **Output Design**

The starting point of the design process is the proper knowledge of system requirements, which will normally be converted in terms of output.

* **Input Design**

Once the output requirements have been finalized, the next step is to find out what data need to be made available to the system to produce the desired outputs.

* **File Design**

Once the input data is captured in the system, these may to be preserved either for a short or long period. These data will generally be stored in files in a logical manner. The designer will have to devise the techniques of storing and retrieving data from these files.

* **Procedure Design**

This step involves specifications of how processing will be performed

* **Control Design**

The control design indicates necessary procedures which will ensure correctness of processing, accuracy of data, timely output etc. this will ensure that the system is functioning as per plan.

**4.2 DATABASE DESIGN**

Database named ‘payroll’ is created using PHPMyAdmin. The different tables involved in the database are as follows:

**Table 4.1: admin Table**

|  |  |
| --- | --- |
| **Name** | **Data Type** |
| username | varchar(90) |
| password | varchar(90) |

**Table 4.2: employee Table**

|  |  |
| --- | --- |
| **Name** | **Data Type** |
| id | int(90) |
| empno | int(100) |
| pay | decimal(10,2) |
| dayswork | int(10) |
| otrate | decimal(10,2) |
| othrs | int(10) |
| allow | decimal(10,2) |
| advances | decimal(10,2) |
| insurance | decimal(10,2) |
| time | date |

**Table 4.3: emp\_info Table**

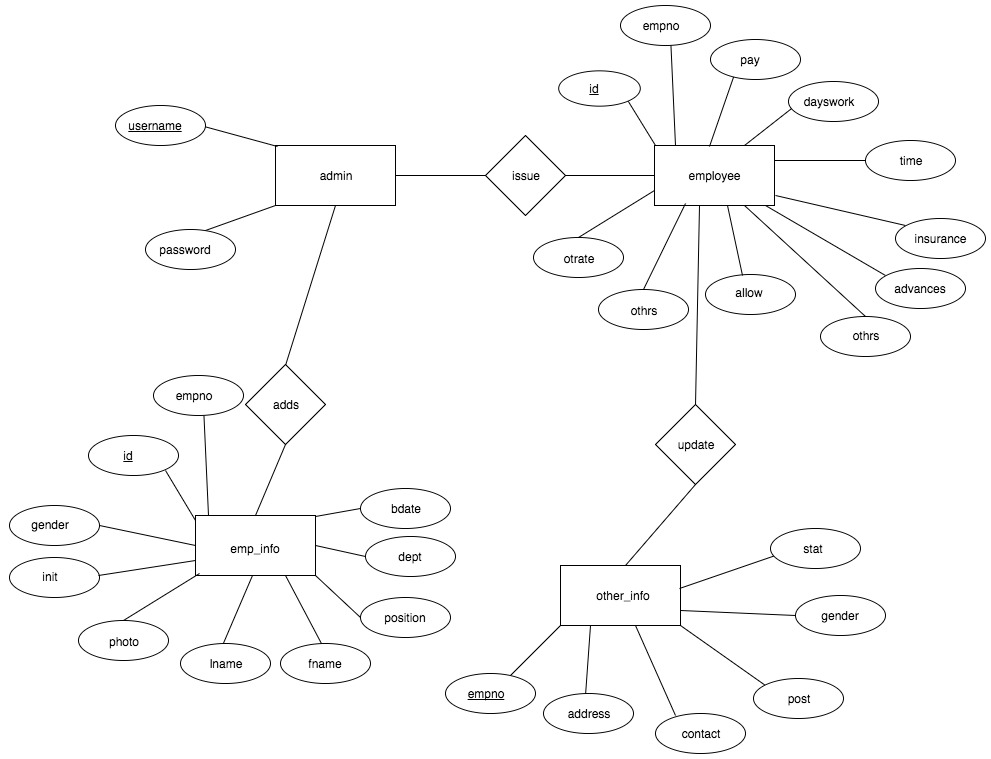
|  |  |
| --- | --- |
| **Name** | **Data Type** |
| id | int(90) |
| empno | int(90) |
| photo | varchar(100) |
| lname | varchar(45) |
| fname | varchar(45) |
| init | varchar(1) |
| gender | varchar(10) |
| bdate | date |
| dept | varchar(15) |
| position | varchar(45) |

**Table 4.4: other\_info Table**

|  |  |
| --- | --- |
| **Name** | **Data Type** |
| empno | int(90) |
| address | varchar(90) |
| contact | int(50) |
| post | varchar(90) |
| gender | varchar(90) |
| stat | date |

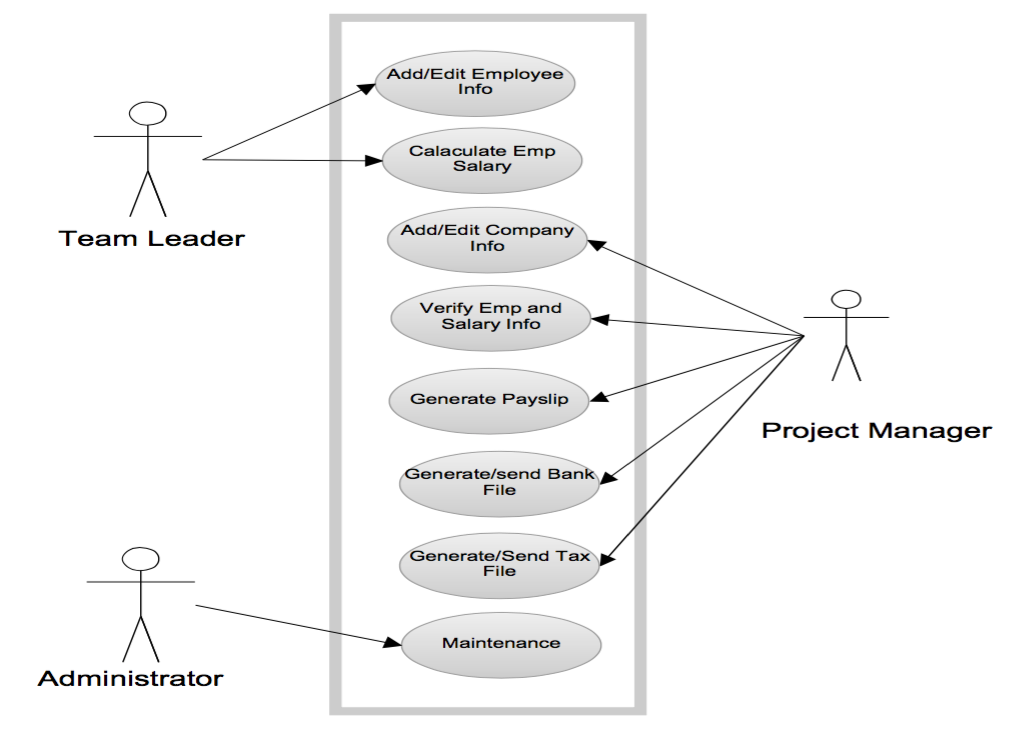
**4.3 ER DIAGRAM**

The ER Diagram related to above database is as follows:



**4.4 USECASE DIAGRAM**

In this section we will discuss the use case of payroll management system. Uses cases show how the users interact with the system. There are three actors in our system, team leaders, project managers and administrator. Each one has its own responsibilities and level of access to the system. The figure is showing a general overview of use cases and its general description.



**Fig 4.1: Usecase Diagram**

***Team Leader***

Team leader is responsible to maintain basic information of his team members such as name, address, date of birth, joining date, social security number etc. A team leader will be assigned a login and password for the system. After authentication he/she will be the only person authorized to add and edit the basic information of the employees. He/she will be able to calculate the salaries of the particular employees on the basis of given parameters.

***Project Manager***

Project manager is the main actor in our system. After passing through the authentication process the he has the access to the system and is authorized to do the following actions: -

1. Addition of necessary information, updates and news etc.
2. Verification of employee data and salary calculation entered by the team leaders.
3. To generate salary slips through the system.
4. To calculate tax payable by employees and organization on the basis of formula provided by the tax office.
5. To generate tax files and send them to the tax office.

***System Administrator***

The last actor is system administrator. Maintenance is one of his major responsibilities. He checks the security logs, handles the security issues, checks the application logs, handles the error massages of the application, handles the database errors and web server problems etc.

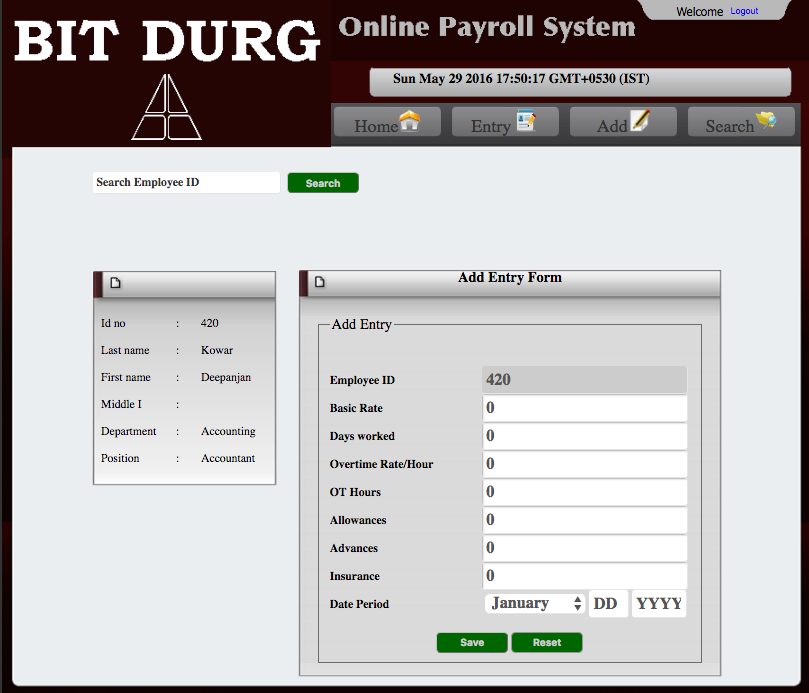
**5.1 SCREENSHOTS**

****

**Fig 5.1: Login Screen**

****

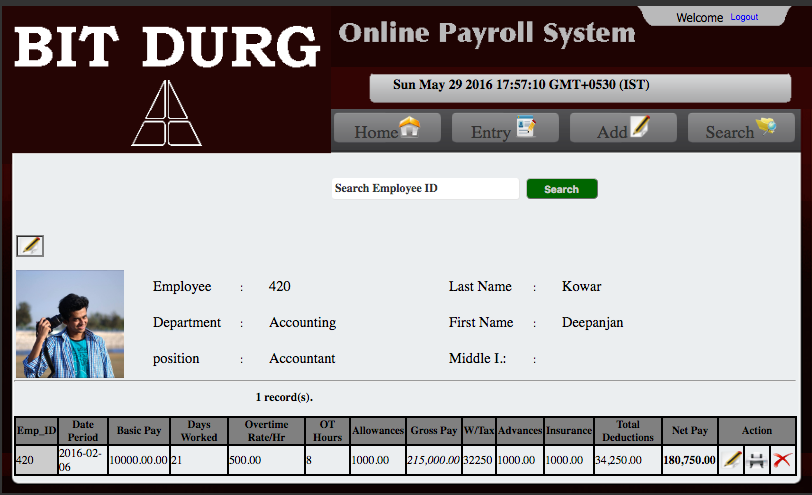
**Fig 5.2: Admin Home Screen**

****

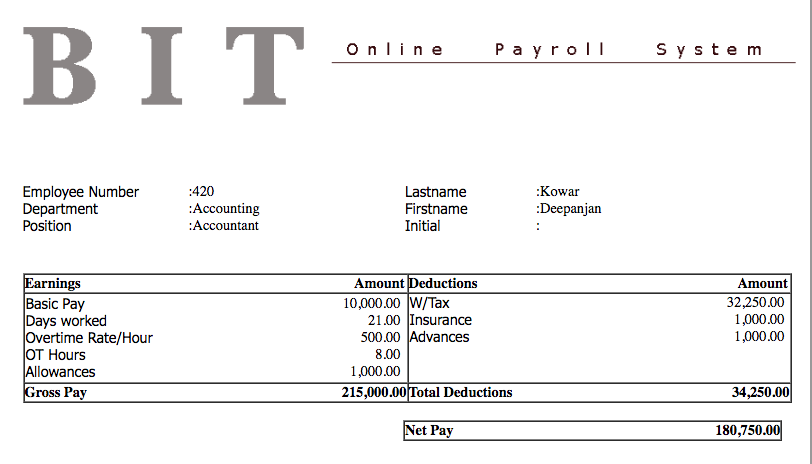
**Fig 5.3: Add Salary Screen**



**Fig 5.4: Add Employee Screen**



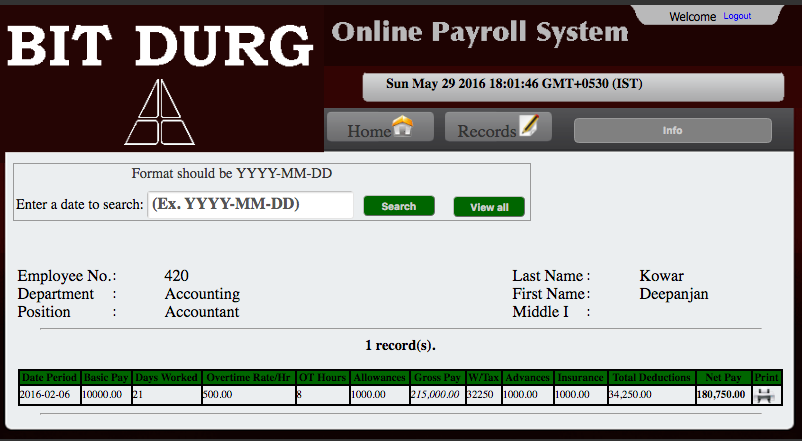
**Fig 5.5: Search Screen**



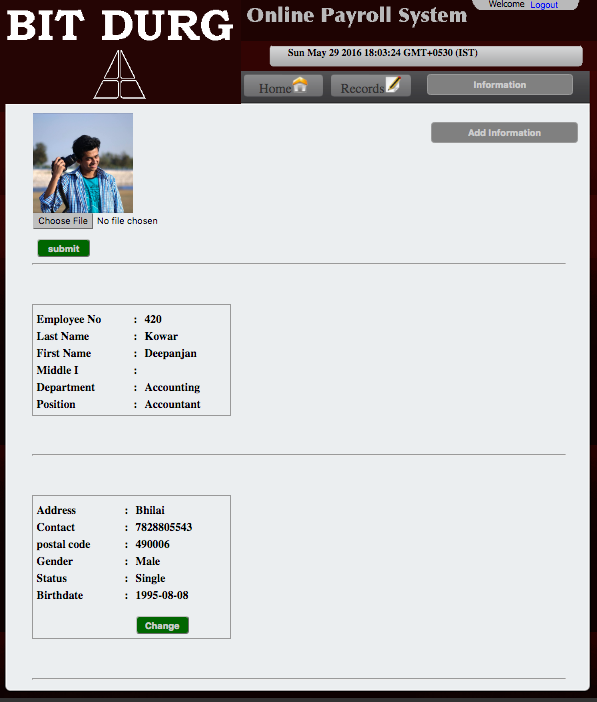
**Fig 5.6: Generated Pay Slip**



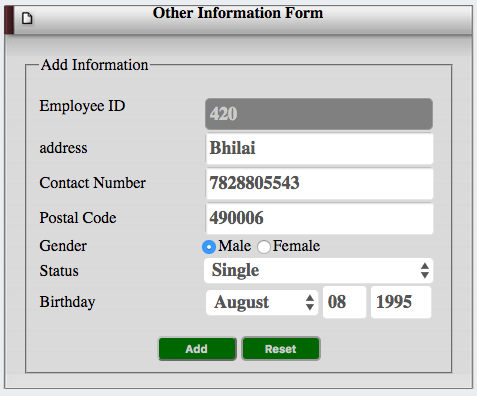
**Fig 5.7: Employee Login Screen**



**Fig 5.8: Records Screen**



**Fig 5.9: Employee Info Screen**



**Fig 5.10: Employee Info Edit Screen**

**6.1 SOFTWARE TESTING**

Software testing is a critical element of software quality assurance and represent the ultimate review of specification, design, coding. The purpose of product testing is to verify and validate the various work products viz. units, integrated unit, final product to ensure that they meet their requirements.

Software testing is an investigation conducted to provide stakeholders with information about the quality of the product or service under test. Software testing can also provide an objective, independent view of the [software](https://en.wikipedia.org/wiki/Software) to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding [software bugs](https://en.wikipedia.org/wiki/Software_bug) (errors or other defects).

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

* meets the requirements that guided its design and development,
* responds correctly to all kinds of inputs,
* performs its functions within an acceptable time,
* is sufficiently usable,
* can be installed and run in its intended [environments](https://en.wikipedia.org/wiki/Operating_environment) and
* achieves the general result its stakeholders desire.

In Online Payroll System following testing has been used.

6.2 UNIT TESTING

Unit Testing is essential for the verification of the code produced during the coding phase and the goal is test for the internal logic of the module/program. In the generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested, they are rightly connected or not.

6.3 INTEGRATION TESTING

All the tested modules are combined into subsystems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

6.4 SYSTEM TESTING

It is mainly used if the software meets its requirements. The reference document for this process is the requirement document.

6.5 ACCEPTANCE TESTING

It is performed with realistic data of the client to demonstrate that the software is working satisfactorily. In the generic code project testing is done to check whether the creation of tables and respected data entry is working successfully or not.

6.6 BLACK BOX TESTING

The black-box approach is a testing method in which test data are derived from the specified functional requirements without regard to the final program structure. It is also termed data-driven, input/output driven , or requirements-based testing. Because only the functionality of the software module is of concern, black-box testing also mainly refers to functional testing -- a testing method emphasized on executing the functions and examination of their input and output data. The tester treats the software under test as a black box -- only the inputs, outputs and specification are visible, and the functionality is determined by observing the outputs to corresponding inputs. In testing, various inputs are exercised and the outputs are compared against specification to validate the correctness. All test cases are derived from the specification. No implementation details of the code are considered.

It is obvious that the more we have covered in the input space, the more problems we will find and therefore we will be more confident about the quality of the software. Ideally we would be tempted to exhaustively test the input space. But as stated above, exhaustively testing the combinations of valid inputs will be impossible for most of the programs, let alone considering invalid inputs, timing, sequence, and resource variables. Combinatorial explosion is the major roadblock in functional testing. To make things worse, we can never be sure whether the specification is either correct or complete. Due to limitations of the language used in the specifications (usually natural language), ambiguity is often inevitable. Even if we use some type of formal or restricted language, we may still fail to write down all the possible cases in the specification. Sometimes, the specification itself becomes an intractable problem: it is not possible to specify precisely every situation that can be encountered using limited words. And people can seldom specify clearly what they want -- they usually can tell whether a prototype is, or is not, what they want after they have been finished. Specification problems contributes approximately 30 percent of all bugs in software.

The research in black-box testing mainly focuses on how to maximize the effectiveness of testing with minimum cost, usually the number of test cases. It is not possible to exhaust the input space, but it is possible to exhaustively test a subset of the input space. Partitioning is one of the common techniques. If we have partitioned the input space and assume all the input values in a partition is equivalent, then we only need to test one representative value in each partition to sufficiently cover the whole input space. Domain testing partitions the input domain into regions, and consider the input values in each domain an equivalent class. Domains can be exhaustively tested and covered by selecting a representative value(s) in each domain. Boundary values are of special interest. Experience shows that test cases that explore boundary conditions have a higher payoff than test cases that do not. Boundary value analysis requires one or more boundary values selected as representative test cases. The difficulties with domain testing are that incorrect domain definitions in the specification can not be efficiently discovered.

Good partitioning requires knowledge of the software structure. A good testing plan will not only contain black-box testing, but also white-box approaches, and combinations of the two.

6.7 WHITE BOX TESTING

Contrary to black-box testing, software is viewed as a white-box, or glass-box in white-box testing, as the structure and flow of the software under test are visible to the tester. Testing plans are made according to the details of the software implementation, such as programming language, logic, and styles. Test cases are derived from the program structure. White-box testing is also called glass-box testing, logic-driven testing or design-based testing .

There are many techniques available in white-box testing, because the problem of intractability is eased by specific knowledge and attention on the structure of the software under test. The intention of exhausting some aspect of the software is still strong in white-box testing, and some degree of exhaustion can be achieved, such as executing each line of code at least once (statement coverage), traverse every branch statements (branch coverage), or cover all the possible combinations of true and false condition predicates (Multiple condition coverage).

Control-flow testing, loop testing, and data-flow testing, all maps the corresponding flow structure of the software into a directed graph. Test cases are carefully selected based on the criterion that all the nodes or paths are covered or traversed at least once. By doing so we may discover unnecessary "dead" code -- code that is of no use, or never get executed at all, which can not be discovered by functional testing.

In mutation testing, the original program code is perturbed and many mutated programs are created, each contains one fault. Each faulty version of the program is called a mutant. Test data are selected based on the effectiveness of failing the mutants. The more mutants a test case can kill, the better the test case is considered. The problem with mutation testing is that it is too computationally expensive to use. The boundary between black-box approach and white-box approach is not clear-cut. Many testing strategies mentioned above, may not be safely classified into black-box testing or white-box testing. It is also true for transaction-flow testing, syntax testing, finite-state testing, and many other testing strategies not discussed in this text. One reason is that all the above techniques will need some knowledge of the specification of the software under test. Another reason is that the idea of specification itself is broad -- it may contain any requirement including the structure, programming language, and programming style as part of the specification content.

We may be reluctant to consider random testing as a testing technique. The test case selection is simple and straightforward: they are randomly chosen. Study in indicates that random testing is more cost effective for many programs. Some very subtle errors can be discovered with low cost. And it is also not inferior in coverage than other carefully designed testing techniques. One can also obtain reliability estimate using random testing results based on operational profiles. Effectively combining random testing with other testing techniques may yield more powerful and cost-effective testing strategies.

6.8 PERFORMANCE TESTING

Not all software systems have specifications on performance explicitly. But every system will have implicit performance requirements. The software should not take infinite time or infinite resource to execute. "Performance bugs" sometimes are used to refer to those design problems in software that cause the system performance to degrade.

Performance has always been a great concern and a driving force of computer evolution. Performance evaluation of a software system usually includes: resource usage, throughput, stimulus-response time and queue lengths detailing the average or maximum number of tasks waiting to be serviced by selected resources. Typical resources that need to be considered include network bandwidth requirements, CPU cycles, disk space, disk access operations, and memory usage . The goal of performance testing can be performance bottleneck identification, performance comparison and evaluation, etc. The typical method of doing performance testing is using a benchmark -- a program, workload or trace designed to be representative of the typical system usage.

**7.1 CONCLUSION**

The conclusions drawn from online payroll system are :

1. Speed

With an online payroll service, you can pull up information or perform calculations quickly. According to PCMag.com, once you set up the online system, it takes just a few minutes to initiate a payroll run. Your main task is to simply enter the number of hours worked per week for each employee and the system will do the rest.

1. Accuracy

If you attempt to perform payroll tasks manually, you run the risk of human error when making calculations. An online payroll system does the calculating for you, so you can keep mistakes to a minimum. You just need to be certain that you have set up your payroll correctly initially and that you enter the correct hours worked.

1. Reduced Labor Costs

Online payroll eliminates the need to hire an outside payroll service or to assign the task to one of your employees, which can help reduce expenses. By performing the weekly updates yourself, you are cutting back on labor costs while taking little time from your other functions. As a business owner, any way of reducing costs can have a big impact on your bottom line.

1. Convenience

An online payroll system allows you to manage your payroll at your convenience. You can enter information at the last minute before you need to meet your payroll if necessary. You also have information at your disposal whenever your need it and you won’t have to wait for a third party to respond to any questions.

1. Keeping Up With Regulations

According to the Top ten Reviews website, a major complaint of business owners is that keeping up with changing government regulations makes managing payroll difficult. With an online payroll service, your system is updated as any changes occur, so you won’t need to worry about violating any tax rules. You won’t have to depend on a third party to keep you up-to-date.

**8.1 FUTURE ENHANCEMENTS**

Everything that is made has some or the other things to be added to make it better than revolutions. The project “Online Payroll System” has been tried to develop a robust and fault free system, still enough flexibility has been provided for further enhancements and modifications. As mentioned earlier that the designed forms are typically reflections of the developer, so it is strongly believed that the enhancement to be done with the project are acceptable for its betterment. But at the same time it should be mentioned that since one cannot claim himself/herself as a master of technology there is always some scope of technical modifications in the project that may lead to find code redundancy & storage space minimization.

The only disadvantage is that it is expensive to install the first time but looking into the future scope the amount paid can be neglected to some extent.

As the increase in the retail market, many high-tech applications and equipment are used where this software is very good option to use. It can further be enhanced to combine with other systems like, attendance, leave, etc. moreover a transaction gateway may be added.

**BIBLIOGRAPHY**

**Books:**

1. Kevin Yank, “PHP & MySQL: Novice To Ninja”, SitePoint Private Limited,3th Edition, Published in: 2009
2. Larry Ullman, “PHP & MySQL for Dynamic Web Sites: Visual Quick Pro Guide”, Peachpit Press , 4nd Edition, Published in: 2011
3. Vikram Vasvani, “The Complete Reference MySQL”, McGraw-Hill Companies, 4th Edition, Published in: 2004
4. David Flanagan, “jQuery Pocket Reference”, O'Reilly Media, 1st Edition, Published in: 2010

**Websites:**

1. [php.net](http://www.java2s.com/Code/Java/CatalogJava.htm)
2. <http://stackoverflow.com/>
3. <https://en.wikipedia.org/>
4. <http://programmers.stackexchange.com/>
5. <https://jquery.com/>