**A PROJECT REPORT**

**ON**

**“Online Payroll System”**

##### Submitted In The Partial Fulfillment For The Award Of Degree

##### Of

## BACHELOR OF ENGINEERING

## **In**

##### COMPUTER SCIENCE AND ENGINEERING

##### From

##### Chhattisgarh Swami Vivekananda Technical University, Bhilai

**(CHHATTISGARH)**

***Submitted by:******Guided by:***

**Deepanjan Kowar** Mr. Dinesh Kumar Bhawnani

3012213024 Department Of CSE



**SESSION 2013 - 2017**

Department of Computer Science and Engineering

**BHILAI INSTITUTE OF TECHNOLOGY**

BHILAI HOUSE, DURG, Chhattisgarh - 491001.

**Bhilai Institute of Technology**

**Bhilai house, Durg (C.G) 491001**

**DECLARATION BY THE CANDIDATE(s)**

I the undersigned solemnly declare that the report of the project work entitled **Online Payroll System**, is based on my own work carried out during the course of my study under the supervision of Mr. Dinesh Kumar Bhawnani.

I assert that the statements made and conclusions drawn are an outcome of the project work. I further declare that to the best of my knowledge and belief that the report does not contain any part of any work, which has been, submitted for the award of any other degree/diploma/certificate in this University/ any other University of India or any other country.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Deepanjan Kowar

Roll No.: 3012213024

Enrollment No.: AN0988

**Bhilai Institute of Technology**

**Bhilai house, Durg (C.G) 491001**

**CERTIFICATE OF PROJECT COMPLETION**

This is to Certify that the report of the project submitted is an outcome of the project work entitled **Online Payroll System** carried out by **Deepanjan Kowar** bearing Roll No. **3012213024,** Enrollment No. **AN0988**; under my guidance and supervision for the award of Degree in Bachelor of Engineering in Computer Science from Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G).

To the best of my knowledge the report

1. Embodies the work of the candidate himself / herself,
2. Has duly been completed,
3. Fulfills the requirement of the Ordinance relating to the BE degree of the University,
4. Is up to the desired standard for the purpose of which is submitted.

**Signature of Coordinator Signature of Coordinator Signature of Guide**

**Prof. Pawan Kumar Patnaik Prof. Kaushal Sinha Prof. Dinesh Kumar Bhawnani**

**Asso. Professor Asst. Professor Asst. Professor**

**Computer Sc. & Engg Computer Sc. & Engg. Computer Sc. & Engg.**

The Project work as mentioned above is hereby being recommended and forwarded for examination and evaluation.

**Dr.(Mrs.) M.V. Padmavati**

**Head of the Department**

**Computer Sc. & Engg**

**Bhilai Institute of Technology**

**Bhilai house, Durg (C.G) 491001**

**CERTIFICATE BY THE EXAMINERS**

This is to Certify that the project the entitled

“**Online Payroll System”**,

Submitted by

**Deepanjan Kowar Enrollment No.: AN0988 Roll No.: 3012213024**

Have been examined by the undersigned as a part of the examination for the award of Bachelor of Engineering degree in Computer Science and Engineering from Chhattisgarh Swami Vivekanand Technical University, Bhilai (C.G)

**(Internal Examiner) (External Examiner) Name: Name:**

**Date: Date:**

**Bhilai Institute of Technology**

**Bhilai house, Durg (C.G) 491001**

###### ACKNOWLEDGEMENT

Working on this project has been a great learning experience for me. There were moments of anxiety, when we could not solve a problem for several days & there were moments when we could solve a problem after struggling for several days. But we have enjoyed every bit of the process and are thankful to all people associated with us during this period.

We convey our sincere thanks to our project guide **Mr.Dinesh Kumar Bhawnani**, for providing us all sorts of facilities. His support & guidance helped us to carry out the project. I owe a great debt of his gratitude for his constant advice, support, cooperation & encouragement throughout the project.

We would also like to express our deep gratitude to **Mrs. Padmavati Metta** for her ever helping and support. We also pay special thanks for her helpful solution & comments enriched by her experience, which improved our ideas for the betterment of project.

It will be our pleasure to acknowledge, utmost co-operation & valuable suggestion from time to time given by other staff members of Department, to whom we owe our entire computer knowledge and also we would like to thank all those persons who have directly or indirectly helped us by providing books & computer peripherals and other necessary amenities which helped in the development of this project which would otherwise have not been possible.

SUBMITTED BY:

**Deepanjan Kowar**

ABSTRACT

The purpose of developing this software project is to fully automate salary system for an organization i.e. Bhilai Institute of Technology, Durg. This software is capable of calculating monthly salaries and tax of employees of that organization. On the basis of certain formulas it will generate files as an output such as salary slips.

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

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 and various deductions to be done from the salary. It has to generate a pay-slip.

Since the project is built in PHP, it works in multiple PC’s installed on multiple computers by sharing same database by which users of different department can use it sitting at different locations simultaneously.

|  |  |  |
| --- | --- | --- |
| INDEX | | |
| Contents |  | Page No. |
| **Chapter 1** | **Introduction** | **9** |
| *1.1* | *Purpose* | *10* |
| *1.2* | *Scope* | *10* |
| *1.3* | *Benefits* | *10* |
| *1.4* | *Definitions, Acronyms, Abbreviations* | *10* |
| *1.5* | *Project Overvies* | *11* |
| **Chapter 2** | **Requirement Analysis** | **12** |
| *2.1* | *Methodology* | *13* |
| *2.2* | *The SDLC Waterfall* | *13* |
| *2.3* | *Reasons to Choose Waterfall Model* | 14 |
| *2.4* | *Hardware Interfaces* | 15 |
| *2.5* | *Software Interfaces* | 15 |
| *2.6* | *Functional Requirements* | 15 |
| *2.7* | *Performance Requirements* | 15 |
| *2.8* | *Attributes* | 16 |
| **Chapter 3** | **Design** | **17** |
| *3.1* | *Product Perspective* | *17* |
| *3.2* | *Product Functions* | *17* |
| *3.3* | *Use Case Diagram* | *17* |
| **Chapter 4** | **Implementation** | **20** |
| *4.1* | *Apache* | *20* |
|  |  |  |
|  |  |  |
| *4.2* | *PHP* | *20* |
| *4.3* | *MySQL* | *21* |
| *4.4* | *PHPMyAdmin* | *21* |
| *4.5* | *Software Development Architecture* | *22* |
| *4.6* | *Three-Tier Architecture* | *22* |
| *4.7* | *Implementation of Three-Tier Architecture* | *23* |
| *4.8* | *Snapshots* | *24* |
| **Chapter 5** | **Testing** | **27** |
| *5.1* | *Testing Strategy* | *28* |
| *5.2* | *Unit Testing* | *28* |
| *5.3* | *Integration Testing* | *28* |
| *5.4* | *System Testing* | *28* |
| *5.5* | *Acceptance Testing* | *28* |
| *5.6* | *Black-Box Testing* | *29* |
| *5.7* | *White-Box Testing* | *30* |
| *5.8* | *Performance Testing* | *32* |
| **Chapter 6** | **Conclusion** | **33** |
| *6.1* | *Advantages* | *33* |
| *6.2* | *Limitations* | *34* |
| *6.3* | *Future Enhancements* | *34* |
| *6.4* | *References* | *34* |

Chapter – 1

INTRODUCTION

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.

## 1.1 Purpose

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 Scope

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.

## 1.3 Benefits

* To improve the efficiency.
* Quickly find out information of an employee details.
* To provide easy and faster access information.
* To provide user-friendly environment.

## 1.4 Definitions, Acronyms, 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*

## 1.5 Project Overview

The following subsections provide the complete overview of the software specifications requirements documentation for the product Employee Payroll Management. The entire SRS is documented in view of User and the following sub sections are arranged to give a complete outlook of the software, its perspective, features, system requirements and users know how it is.

Chapter – 2

REQUIREMENT ANALYSIS

Requirements analysis, also called requirements engineering, is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. In software engineering, such requirements are often called functional specifications. Requirements analysis is an important aspect of project management.

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to mold user expectations to fit the requirements.

Requirements analysis is a team effort that demands a combination of hardware, software and human factorsengineering expertise as well as skills in dealing with people.

## 2.1 Methodology

The most suitable and appropriate software development methodology we found for our software development is waterfall software development model, which gave us a clear view about our software and helped us to achieve our goal.

2.2 The SDLC Waterfall

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.

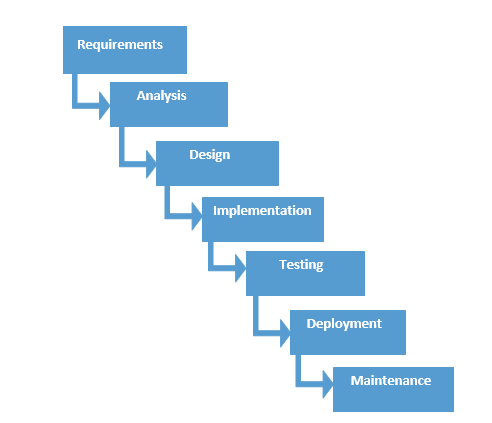
The next stage is the requirement definition stage that takes goals as input from the planning stage and then these goals are defined into a set of one or more requirements. Major functionalities, initial data entities and the operational data areas are defined under this stage.

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.



2.3 Reasons To choose Waterfall Model

As 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.

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 Hardware Interfaces

# Memory minimum of 512MB RAM (1 GB Recommended)

* Hard disk of 40 GB
* Printer

2.5 Software Interfaces

# Operating System - Any

* Browser - Any (Google Chrome Recommended)
* Front End - PHP, HTML, CSS
* Back End - MySQL

2.6 Functional Requirements

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.

2.7 Performance Requirements

* The overall system should be fast and error free.
* It should have built in error checking and correction facilities.
* The system should be able to handle large amount of data comfortably.

2.8 Attributes

# *Reliability:*

In order to ensure reliability, this system is being designed using software that is established to be stable and easy to use.

# *Availability:*

The system is designed to run 24/7 and be readily available to the user.

# *Security:*

The access to the software is given only to valid operators. We need a specific ID and password to get access to the software.

Chapter – 3

DESIGN

3.1 Product Perspective

This software is developed specifically to cater the company employee’s payroll management and is totally self-contained and works efficiently. It provides simple database rather than complex ones for high requirements and it provides good and easy graphical user interface to both new as well as experienced user of the computer.

3.2 Product Functions

a) Employee Module

* Designation
* Department
* Employee Details

b) Search Module

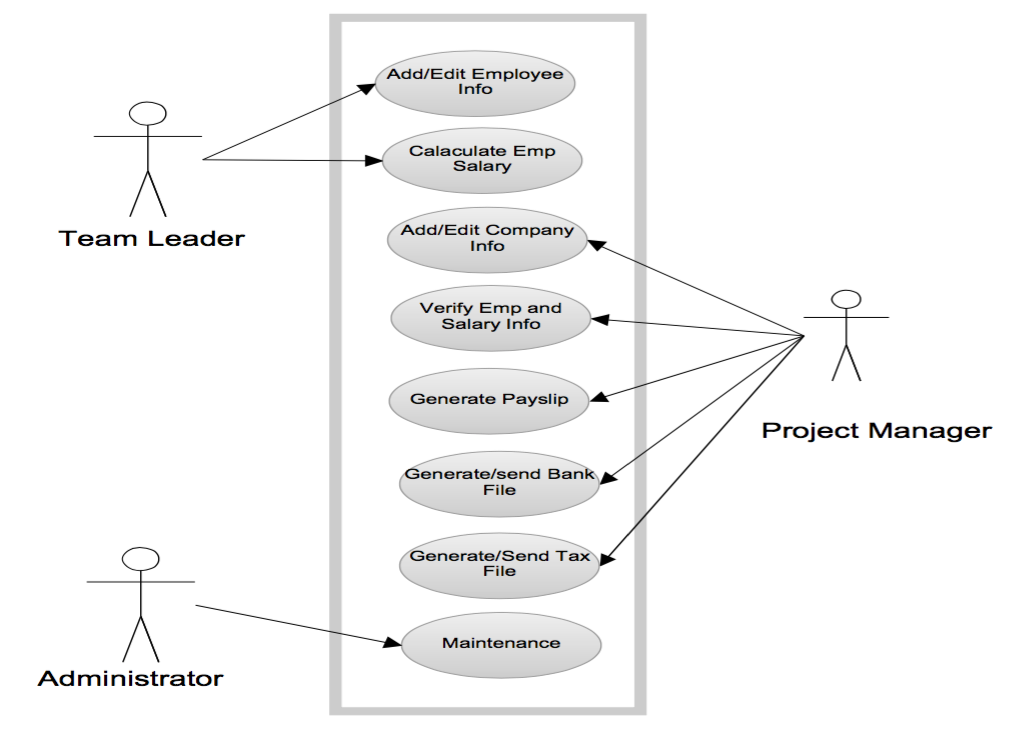
c) Employee Module

d) Salary Module

* Allowance
* Deduction
* Pay Slip

3.3 Use Case 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.



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.

Chapter – 4

IMPLEMENTATION

In this chapter we will describe elements of development phase, technologies, tools that are required in the completion and implementation of our software project.

4.1 Apache

Apache web server is widely used as HTTP(web) server which is an enhanced version of the previous ones. The reason of its popularity is due to its robustness, rich features, and open source code and platform independence. The main reason to choose the apache is due to its platform independence, which enables our system to run on any environment under any operating system.

4.2 PHP

**History**

PHP is a structured programming language, composition of variables, classes and functions like C++. Rasmus Lerdorf first introduced it in 1994. It had some loose free ware scripts, based on perl and dubbed. At that time PHP was an abbreviation of personal home page. Now days it called hypertext preprocessor. With the release of PHP version 3 in 1998, it appeared as a very popular and strong web-developing tool in the web developing society.

PHP supports a lot of features such as, support to databases of different platforms like ODBC, Oracle, mySQL, images, XML and host of many other technologies. Zend encoder is another powerful feature due to which PHP becomes more popular amongst the professional web developers.

**What is PHP**

PHP is an open source server side scripting language, which needs three things the PHP parser (CGI or Server module), a web server and a web browser to perform work. Another feature in it, which is the reason of its popularity, is platform independence, i.e., PHP can be used for all major operating systems including Linux, many Unix variants, Microsoft windows etc.

It also supports most of the web servers today such as apache, Microsoft IIS, Netscape and many others. PHP supports for procedural, OOP with many code libraries as well as SAX and DOM standards. PHP also generates automatically and saves a variety of output files other than HTML including PDF files, images, XML files and many more. More over a very strong and significant feature of PHP is its support for a wide range of databases. PHP can instantiate java objects to use them as PHP objects.

**Why PHP**

PHP is popular as it is simple and easy to use, even for people with no programming experience can use it for designing and developing web applications. It also provides stability, compatibility (platform independence), working efficiently with a variety of web servers. PHP is an open source; it does not need any license and is available free.

I am using MySQL database, which is perfectly supported by PHP. I chose PHP because we found it to be a flexible, user friendly, platform independent and it supports for a variety of web servers.

4.3 MySQL

MySQL is an open source, multi-threaded, relational database management system created by Micheal Monty in 1995, it can be used by anyone for personal use. There are several reasons of its popularity among other freely available open source database systems such as availability, reliability, scalability, speed, security, and performance. It is providing a strong back end support to web applications.

Moreover MySQL is providing a lot of built-in functions and also a variety of storage engines, which are responsible to handle the queries between the back end storage database and a users SQL statement. That is why I choose it in our project.

4.4 PHPMyAdmin

PHP MyAdmin is an open source tool that is used to administrate MySQL over World Wide Web. It supports most of the MySQL features such as managing databases, tables, fields and setting the privileges to the users, management of stored procedures and triggers. It can import data from SQL and CVS and can export it into various formats like CSV, SQL, PDF, XML etc.

More over it can administrate multiple servers, perform search operations in the database, generate complex quires using query-by-example (QBE) and a wide range of many other features.

As discussed above PHP MyAdmin is a useful tool for the administration of a database and since I am using MySQL in my project so, found phpMyAdmin very useful for my project.

4.5 Software Development Architecture

In software engineering a system can be divided into multiple-layers either physically or logically providing a client-server environment known as n-tier architecture. The purpose of dividing the system is to provide a model for software developers developing flexible, reusable and more secure applications.

We applied 3-tier architecture in the development of our web application due to reliability, re-usability, flexibility and security. Making change in one layer has no effect on other layers; so, less work is required for maintenance and error correction.

4.6 Three-Tier Architecture

A 3-tier architecture comprises of following layers :-

• Presentation/Client Tier

• Middle Tier

• Data Tier

**Presentation/Client Tier**

This layer is on the top of application development hierarchy, which enables a client to put some requests by putting a URL in the web browser, or enter some data into a form. This layer fetches the results according to the request from the other tiers by communicating with them and presents it to the user.

**Middle Tier**

The layer that work as an agent between the data storage and presentation tier is known as middle tier. This layer also performs logical processing of the application and provides all the functions. In web applications, the middle tier consists of a web server, which is responsible to get the request from client tier. It then sends it to data tier after doing appropriate processing on the request generated by the presentation tier, giving back results to the client tier.

**Data Tier**

In 3-tier architecture database server is considered as a third layer of the architecture. At this level data is stored and retrieved according to the requirements and requests sent by the presentation tier through middle tier. This layer is independent of other tiers due to which the performance and scalability of the system increases.

4.7 Implementation of Three-Tier Architecture

**User Interface**

It is easy to use and easy to understand user interface. User interacts with the web server through interface and performs different tasks like employee registration or salary calculation etc. Web browser is required to run the user interface.

**Web Server**

In our project apache is used as web server. Web server is used for secure communication between user interface and database server. Web server receives the request from client and sends them to the database. Similarly, it takes data from the database against the client’s request and forwards it to the client.

**Database Server**

MySql server is used in this project to handle database operations. The client does not directly interact with database; rather it goes through web server in order to keep the database secure from an unauthorized access. It is also called a third tier in the three-tier architecture as implemented in this project.

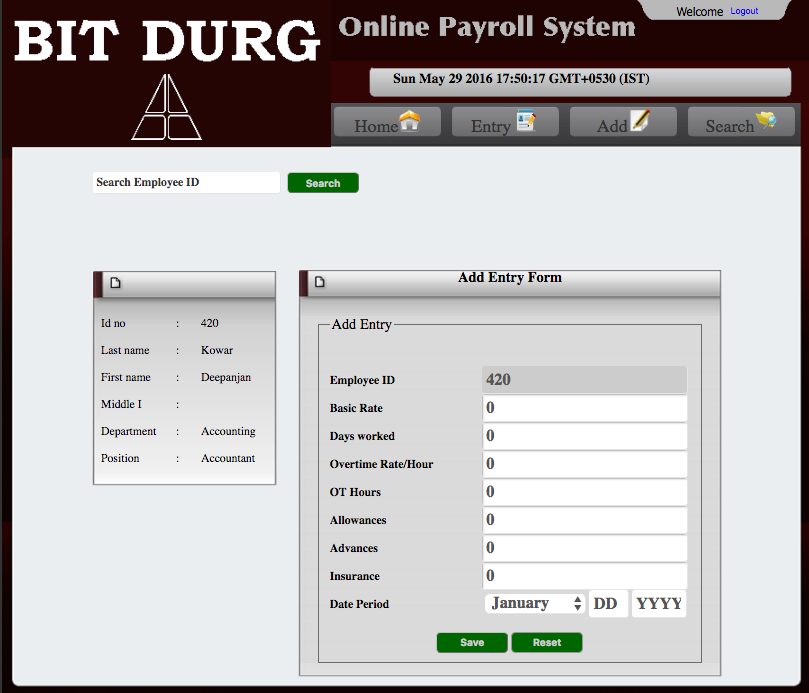
4.8 Snapshots



Login Screen



Admin Home Screen

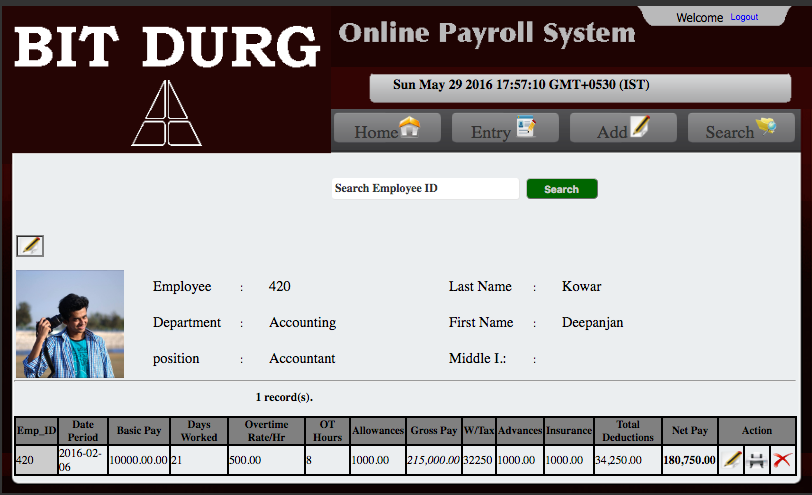


Entry Form

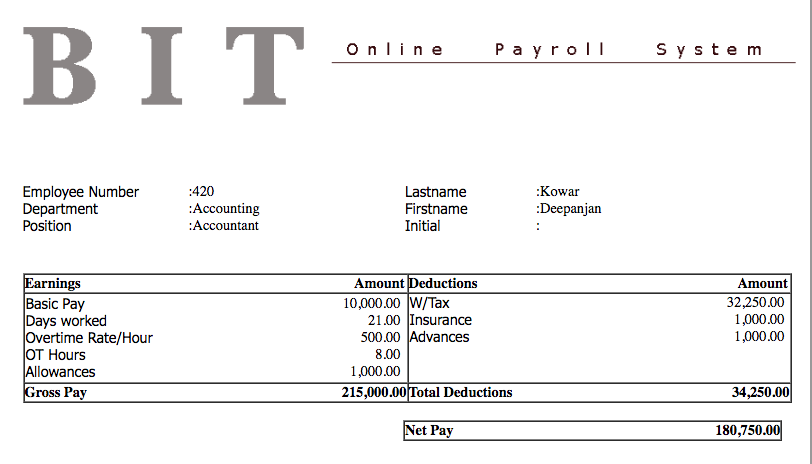


New Employee Add

Screen



Search Screen

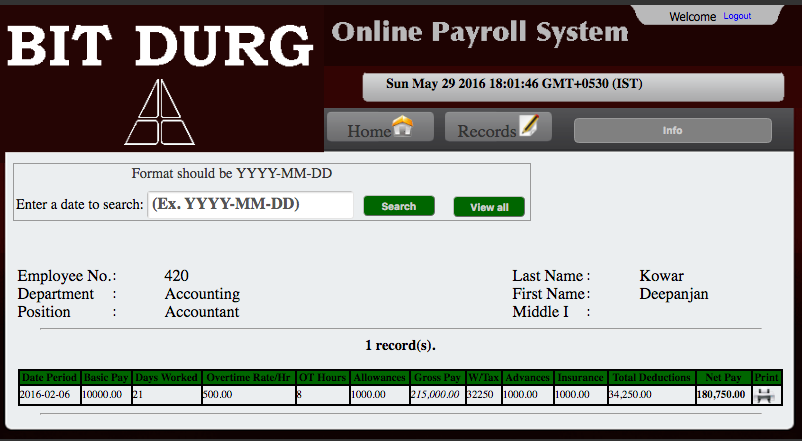


Generated Pay Slip

Employee Login Screen

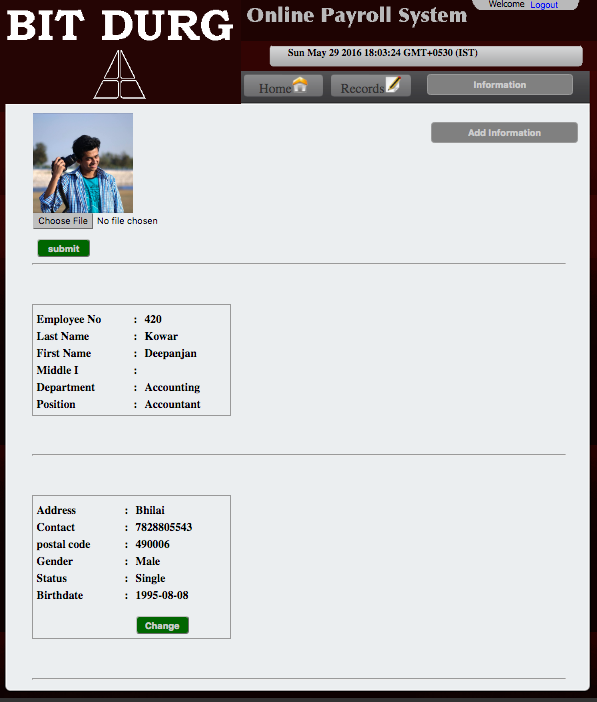


Employee Records



Employee Information

Screen



Chapter – 5

TESTING

Software testing is any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Although crucial to software quality and widely deployed by programmers and testers, software testing still remains an art, due to limited understanding of the principles of software. The difficulty in software testing stems from the complexity of software: we can not completely test a program with moderate complexity. Testing is more than just debugging. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. Testing can be used as a generic metric as well. Correctness testing and reliability testing are two major areas of testing. Software testing is a trade-off between budget, time and quality.

5.1 Testing Strategy

Software testing strategy includes the following:

* Test activities are determined and test data selected.
* The test is conducted and test results are compared with the expected results

There are various types of testing

5.2 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.

5.3 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.

5.4 System Testing

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

5.5 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.

5.6 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.

5.7 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.

5.8 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.

Chapter - 6

CONCLUSION

6.1 Advantages

### 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.

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.

## 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.

## 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.

## 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.

6.2 Limitation

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.

6.3 *FUTURE ENHANCEMENTS*

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

6.4 References

[**www.w3schools.com**](http://www.w3schools.com)

[**www.php.net**](http://www.php.net)

[**www.pear.php.net**](http://www.pear.php.net)