**C:\Documents and Settings\Administrator\My Documents\Downloads\logo.jpgBESTLINK COLLEGE OF THE PHILIPPINES**

*1044 Brgy. Sta Monica, Quirino Hi-way, Novaliches, Quezon City*

**SERVICE MANAGEMENT AGENCY: PAYROLL SYSTEM**

A Project Study

Present to the I.T Project Evaluation Committee

Bestlink College of the Philippines

Quezon City, Philippines

In partial fulfillment of the requirements for the Degree

Bachelor of the Science in Information Technology

By;

Baldado,Jeric V.

Cordero,Liezl S.

Coronel,Christian james D.

Pastoril,Regie S.

Tolentino,Jessica L.

March, 2014

**CERTIFICATION**

This Project Study entitled SERVICE MANAGEMENT AGENCY: PAYROLL SYSTEM prepared and submitted by Baldado, Jeric V., Cordero, Liezl S., Coronel, Christian James D., Pastoril, Regie S., Tolentino, Jessica L., In partial fulfillment of the requirements for the degree BACHELOR OF SCIENCE IN INFORMATIONTECHNOLOGY has been examined and recommended for **Oral Examination.**

**Evaluation Committee**

Dr.Rosicar Escober

Chairman

Mr.Rommel Constantino Mr.Rico Bundoc

Member Member

**APPROVAL**

Approval by the Panel on Oral Examination on **13 February 2014** with the grade of .

Dr.Rosicar Escober

Chairman

Mr.Rommel Constantino Mr,Rico Bundoc

Member Member

**CERTIFICATION OF ORIGINALITY**

This is to certify that the research work presented in this Project Study entitled SERVICE MANAGEMENT AGENCY: PAYROLL SYSTEM for the degree Bachelor of Science in Information Technology at the Bestlink College of the Philippines embodies the result of original and scholarly work carried out by the Undersigned. This Project Study does not contain words or ideas for the publish sources nor written work that have been accepted as basis for the award of a degree from any higher education institution, except where proper referencing and acknowledgement were made.

Baldado,Jeric V.

Cordero,Liezl S.

Coronel,Christianjames D.

Pastoril,Regie S.

Tolentino,JessicaL.

Researchers

March 2014

**APPROVAL SHEET**

This Project Proposal entitled

**SERVICE MANAGEMENT AGENCY: PAYROLL SYSTEM**

By:

Baldado,Jeric V.

Cordero,Liezl S.

Coronel ,Christian james D.

Pastoril,Regie S.

Tolentino,Jessica L.

After having been presented

Is hereby approved by

Mr. Rommel Constantino

Project Study Coordinator

March 2014

**ABSTRACT**

Northern Star Investigation and Security Agency Inc. did not have the Computerized Payroll System, they are only using a manual payroll this will make the payroll transaction more difficult and prone to error. The entire employee’s salary has to calculate it as manually which result to a certain problem.

Since that the Northern Star Investigation and Security Agency used manual transaction the company prone a lot of errors such as inaccurate computation of salary due to calculator malfunction, unsecured file storage, time consuming

We as a developer want to make a system that can really help for their payroll system that’s why the group conduct a project study to develop the payroll system that can solve their entire problems that will encountered by their company for their payroll system. And also to assure that all salary record and salary process of employee are secured, including the file storage of record, systematic payroll process

Because Payroll System is basically involves everything that a company uses to managethe records of the employees working in the company. As soon as the company hires an employee, the employer is responsible to pay the corresponding amount of salary correctly. The salary will motivate the employees to work. Payroll System must not fail to fulfill their duties

The payroll system also involves net and gross pay deduction such miscellaneous, bonuses,allowances and deductions.

**ACKNOWLEDGEMENT**

The Proponents would like to acknowledge the people behind the success of this Research;

First of all, the proponents would like to thank our **Almighty God,** for giving us the courage, wisdom, patience and strength on making this project possible;

**Ms. Eunice Balisi,** for her intellectual advices regarding our system;

**Ms. Ma. Aimee Bechiada,** for her long patience in checking our documents.

**Northern Security Agency,** for giving us the information that we need to make this Project Study;

**Our Group mates,** who spend ample of time on doing this Project Study and also for all the sleepless time together;

**Our Parents,** for their unconditional love, understanding and support most especially on financial aspects.

**-RESEARCHERS**

**TABLE OF CONTENTS**

Title page

Certification ……………………………………………………………. i

Certification of Originality……………………………………………………... ii

Approval Sheet …………………………………………………………..... iii  
Abstract ……………………………………………………………. iv  
Acknowledgment ……………………………………………………………. v  
Table of Contents ………………………………………………………….… vi  
List of Acronyms ………………………………………………………….… x

**Chapter 1**

1. Introduction …………………………………………………… 1

1.1 Project Overview …………………………………………… 2

1.2 The Problem Domain ……………………………………………

* + 1. Statement of the Problem  …………………………… 3

1.3 Objectives of the Project  …………………………………… 4

1.4 Significance and Scope of the Project  …………………… 5

1.5 Documentation of Existence and Seriousness of the Problem

1.5.1 Documentation of Current Systems …………………… 6  
1.5.2 Problem/S Identified With the Existing Systems ….…...…. 7

1.5.3 Process Models of Existing Systems

Contributing to the Problem ………………………….… 9

* + 1. Data Models of These Existing Systems ………………... 10

Data (or any form of statistics) that may be relevant to prove

Existing and seriousness of the identified problems ……… 11

1.6 Review of Existing Alternatives …………………………..… 13

**Chapter 2**

2.0 Review of Related Literature - Introduction …………………............ 15

* 1. Related Literature
     1. Foreign Literature ……………………………………..……..... 16
     2. Local ……………………………………………………..…..... 17
  2. Related Studies
     1. Foreign Studies ……………………………………….…….…. 19
     2. Local Studies ………………………………………….…....…… 20
  3. Synthesis And Significance Of The Study …………………....…...... 21

**Chapter 3**

3.1 Software Requirements Specification

* + 1. Introduction
       1. Goals and Objective ………………………….….……… 23
       2. Statement of Scope …………………………….……....... 24
       3. Software Context ……………………………….…...…... 27
       4. Major Constraints ………………………….…………….
    2. Usage Scenario
       1. User Profiles ………………………………….…………. 28
       2. Use-Cases ………………………………….……………
          1. Use-Case Diagram ………………………….... 29
          2. Use-Case Description ……………………...... 30
       3. Special Usage Considerations ……………………............ 33
       4. Activity Diagram …………………………………….....
    3. Data Model Description
       1. Data Objects …………………………………………..… 40
       2. Relationships ……………………………………….….... 43
       3. Complete Data Model ………………………………..… 45
    4. Functional Model Description
       1. Class Diagram ……………………………………........... 46
       2. Software Interface Description
          1. External Machine Interfaces ……………....... 47
          2. External System Interfaces ……………...…...
          3. Human Interface ………………………….….
       3. Reports
          1. Inventory Reports ……………………….…...
          2. Layout of Reports …………………………… 49
          3. Data Dictionary of Reports …………………. 51
    5. Behavioral Model Description
       1. Description for Software Behavior
          1. Events …………………………………….…. 53
          2. States ……………………………………….... 55
       2. State Charts Diagram ……………………………………. 56
    6. Restriction, Limitation, and Constraints …………………….… 59
    7. Validation Criteria ………………………………………..........
       1. Classes of Test …………………………………………...
       2. Expected Software Response ………………………….....
       3. Performance Bound ……………………………………... 60
  1. Software Design Specification
     1. Introduction
        1. Goals and Objectives …………………………………….
        2. Statement of Scope ……………………………………… 61
        3. Software context ……………………………………….... 62
        4. Major Constraints ………………………………………..
     2. Data Design
        1. Internal Data Structure …………………………………...
        2. Global Data Structure ………………………………........ 63
        3. Temporary Data Structure ………………………….........
        4. Database Description ……………………………….…....
     3. Architecture and Component level Design
        1. Architectural Diagram ……………………………….….. 65
        2. Description of Components

3.2.3.2.1 Component Payroll Admin ………………….… 70

3.2.3.2.2 Component Accountant ……………………….. 71

* + - 1. Dynamic behavior for component
         1. Interaction Diagrams ………………………. 74
    1. User Interface Design
       1. Description of the User interface ………………….……. 76
       2. Interface Design Rules ………………………………..… 81
       3. Components available………………………………....… 83
       4. UIDS description ………………………………………..
    2. Restrictions, Limitations, and Constraints ………………………
    3. Testing Issues ………………………………………………….... 84

**Chapter 4-Testing**

Includes Test Scenarios and Evidence ………………………………….. 85

Chapter 5-Evaluation

5.1 Findings………………... …………………………………………… 93

5.2 Analysis ……………………………………………………………... 94

5.3 Lesson Learned …………………………………………………..….. 96

5.4 Recommendations ……………………………………………………

Chapter 6-Conclusion

6.1 Conclusion of the findings …………………………………………... 97

6.2 Future suggested researches …………………………………………

References …………………………………………………………………….… 98

Appendices

Appendix A- Curriculum Vitae ………………………………………… 99

Appendix B- Sign Forms from PEC and adviser (F1-F4) ………………. 105

Appendix C- User Manual (Operations Manual) ……………………….. 114

Appendix D- Code Listing ……………………………………………… 120

**LIST OF ACRONYMS**

SSS – Social Security System (chapter 1, page 2)

BIR – Bureau of Internal Revenue (chapter 1, page 3)

COLA – Cost of Living Allowance (chapter 1, page 7)

OT – Overtime (chapter 1, page 7)

ESIS – Electronic Securities and Information System (chapter 2, page 2)

EPF – Employees Provident Fund (chapter 2, page 2)

DBM – Department of Budget Management (chapter 2, page 3)

NPS – Notice of Payment Schedule (chapter 2, page 4)

DOST – Department of Science and Technology (Chapter, Page 4)

ASTI – Advance Science and Technology Institute (Chapter 2, Page 4)

DOH – Department of Health (chapter 2 page 6)

GSIS – [Government Service Insurance System](http://en.wikipedia.org/wiki/Government_Service_Insurance_System) **-** (chapter 2 page 6)

Y2K – year 2000 (chapter 2 page 6)

CAPELCO – Capiz Electric Cooperative (chapter 2 page 7)

NSSA – Northern Star Security Agency (chapter 3 page 1)

MySQL – My Structured Query Language (chapter 3 page 5)

JDBC – Java Database Connectivity (chapter 3 page 5)

SQL – Structured Query Language (chapter 3 page 5 )

PII – Pentium II, (chapter 3 page 5)

RAM – Random Access Memory, (chapter 3 page 5)

MB – Megabyte, (chapter 3 page 5)

DTR – Daily Time Record (chapter 3 page 15)

GUI – Graphical User Interface (chapter 3 page 37)

DBMS – Database Management System (chapter 3 page 37)

UML – Unified Modeling Language (chapter 3 page 42)

PNB – Philippine national bank

DTR **-** Department of transcript record

**CHAPTER 1**

1. **INTRODUCTION**

Payroll System is basically involves everything that a company uses to managethe records of the employees working in the company. As soon as the company hires an employee, the employer is responsible to pay the corresponding amount of salary correctly. The salary will motivate the employees to work. Payroll System must not fail to fulfill their duties. And also the company must be responsible to make sure that the proper taxes are withheld from each employees paycheck and that those funds are paid to the government agency at the right time.

Changes in technology bring significance and advancement in our lives. This advancement of computer had been a great factor in our daily living. In a company, there are several departments to manage their payroll activities. Each section has to perform necessary personnel operations like data collection and preparation, entry, updates, monitoring and reporting of data. Many of these existing practices and procedures need to consider at this time of changing needs, changing demands of employees and changing technologies. So a company needs an integrated computer based payroll system that would integrate personnel and pay-roll related data collection, processing in individual departments and payroll audit in a more efficient way.

With this computer based payroll system, payroll section would be able to keep a record of employees including their personnel data, pay rate, allowances, deductions, benefits and taxes etc. Net pay of each employee is calculated by using the information about employee’s gross pay which is deducted by all deductions. All these complex calculations have been incorporated herein. Many optional allowances and deductions for the employees are added by specifying the corresponding details. The employees pay slips and deduction vouchers are printed out as a receipt. Daily rate, allowances, deductions and tax information are updated if there is any amendment in salary structure.

* 1. **PROJECT OVERVIEW**

This chapter will briefly discuss the proposed project about payroll system that has to be developed for Northern Star Investigation and Security Agency Inc.

The business and industries are using computers but not using a computer base transaction like payroll system. The Northern Star Investigation and Security Agency Inc is one of a kind which uses a manual payroll instead of a computerize system.

Having a computer base payroll system is very important for a company or industries to make the payroll process easily and effectively. The payroll system will help the Administrator perform his task regarding on payroll accurately, timely and cost efficiently. Therefore developing a payroll system the company is a very useful for them.

The companies have to maintain all employees’ detail and salary information. Using employees ID number, payroll admin can easily locate especially employee’s salary information.

The system is easy in terms of navigation to all features of the program, it provides multilevel password protection, plus built-in backup and restore, tracks payroll records and automatically calculates basic pay, overtime pay, Night deferential, rest day , holiday pay, benefits deductions such as SSS, Philhealth, Pag-ibig, taxes and other deduction, can generate vacation, bonuses such as midyear bonus , Christmas bonus , 13th month pay, allowance, incentives leave with pay, sick and ability to process advance payment and supports pay slip printing and different reports.

* 1. **THE PROBLEM DOMAIN**

Northern Star Investigation and Security Agency Inc. do not have the Computerized Payroll System, they are only using a manual payroll this will make the payroll transaction more difficult and prone to error. The entire employee’s salary has to calculate it as manually which result to a certain problem.

* + 1. **Statement of the Problem**

The purpose of this project is to identify the different problems that the company encountered regarding on their payroll system. The group formulated different questions on how and where does the problem occur? By answering the question the group found the different problem in the company’s payroll system.

* **Inaccurate computation of salary due to calculator malfunction.**

Upon collecting employee’s time record, employee’s salary is calculated using calculator, base on the days that the employee had work. Employees earning will increase by overtime, allowance, holiday pay, incentives and a lot more to be added in their salary. There are also corresponding deductions to be deducted in employee’s salary so if the calculator will not function properly it will lead to inaccurate employee’s salary computation.

* **Unsecured file storage**

Since the company stored their files in a filing cabinet it will lead to a loss of important documents and will be open to anyone or might fall to a wrong person.

* **Time consuming in preparing payroll**

During payroll period it takes a lot of time in verification of time record calculating employees’ salary, deductions, and tax withholding. It also consume a lot of time in finding employees record.

* **Inaccurate tax report generation**

Each year the company needs to submit their employee’s withholding tax to the government office like BIR. There are instance that there are changes in the withholding tax.

* 1. **OBJECTIVES OF THE PROJECT**

The goal of this project is to develop this payroll system that can solve this entire problem encountered by the company for their payroll system. The specific objectives of this project are:

* **To provide an automatic and accurate calculation of salary**

Automate salary calculation particularly in calculating the number if days work, basic pay, overtime, allowances, holiday pay, incentives, gross pay, deduction and any other data. The percentage to make mistake is low since the payrolls systems will be automatically calculated

* **To provide sufficient security of payroll records**.

All payroll information are stored in one location which has a database and control by the payroll controller and the administrator. It required user name and password every time they use that system. This will secure the payroll information and only an authorize person can use the system.

* **To improve time efficiency in updating employee’s record**.

Admin can view and update records of an employee. Detail can be located easily by the use of search engine from the system. This will make the transaction faster.

* **Generate Accurate Tax Reports**.

A payroll system must be able to keep and generate accurate reports on employee income. Accuracy is an important objective of a payroll system, because employee’s income tax should be submitted to the government office fairly and to avoid penalties.

* 1. **SIGNIFICANCE AND SCOPE OF THE PROJECT**

**Scope**

* **Get the list of employees from the Recruitment**

The system need to access the list of employee record from the recruitment system to view the information of employee. It need to be obtained because it is one of the essential information and include it in our main transaction.

* **Get the number of working days from Monitoring**

Payroll systems need get the summary of employee’s time record from monitoring personnel since they are monitoring the employee’s attendance information. The payroll system need that time record in order to compute employee salary.

* **Add employee wage**

The payroll system needs to add employee daily rate.

* **Compute salary deduction such as SSS, Pagibig, Philheath, Tax**

In terms in salary deduction like SSS, Pag-ibig, Philheath and tax are deducted to each employee according to their different range of their income or compensation. There is an official table that needs to follow upon deduction. Different ranges of income has different amount of contribution and withholding tax.

* **Compute employee salary**

Employee’s salary is computed according to the number days they have work. It could increase by overtime, night differential, holiday, allowance bonus. There are also a corresponding amount to be deducted such as SSS, Pag-ibig, Philheath, withholding tax, late and undertime. If the person applies for loan it also is added to his salary deductions.

* **Generate and print reports**

The payroll system can generate reports such as:

* Semi Monthly and monthly payroll reports
* Statutory reports such as SSS, Pag-ibig, Philhealth and withholding tax.
* Allowance report
* Bonus and 13 Moth pay report
* Incentive leave reports
* Loan payment reports
* **Generate and print payslips**

The payroll system can generate and print employee’s payslips to be distributed to the employees. In payslips it states the summary of employee’s compensation such as gross pay, total deduction and net pay.

**Significance**

Developing this project brings significance to the employees and employer. The following are the importance of this project:

* To save time, this will reduce cost and the amount of time spent in calculating employee’s wages.
* Organized filing system. All documents are stored in the system this will eliminate paper works and filing.
* Secure data records in a database which is easy to retrieve.
* Help accuracy and efficiency in computing employee’s salary.
* Systems are much faster and easier compared to the manual processing.
* Easy backup and restore data.
  1. **DOCUMENTATION OF EXISTENCE AND SERIOUSNESS OF THE PROBLEM.**
     1. **Documentation of current systems**

The Northern Star Investigation and Security Agency Inc is using a manual system which contains only paper works for their Payroll. They have encounter different problems regarding on their payroll such as mistaken and redundancy of the employees information and other payroll records such as number of days work, rate per day, overtime, night differential, holidays, allowance , incentives and deduction because they are used manual typing.

The payroll Administrator will get the attendance record of all employees. They type the name of employees in a payroll form. Using microsoft office and calculator they have calculate the numbers of days work. The numbers of days work is multiplied by the rate per day to get the basic pay. The basic pay will be added to the COLA, overtime, OT with night differential, incentives and allowance to sum up the Gross pay. The gross pay will be subtracted to the deduction such as SSS, philhealth, pag-ibig, Tax, etc. to get the Net Pay. Every end of the year or at the month of December employees can get their 13th month pay.

After they calculate the pay for each employee, the payroll administrator will now verify it was calculated correctly, since it is manually calculated so it is prone to error. Then all payroll information will be typed in the payroll form together with their corresponding name. Then they will now distribute the employee’s salaries. As the employee receives the money they will attach their signature in the payroll form.

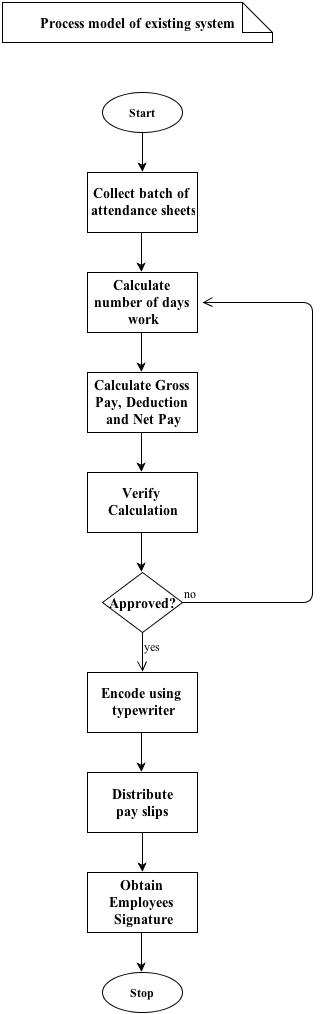
* + 1. **Problem/s identified with the existing systems**

By formulating those questions from statement of the problem there are problem identified with the existing system ofNorthern Star Investigation and Security Agency Inc. since they do not have the Computerized Payroll System, they are only using a manual payroll that make the payroll transaction more difficult. The entire employee’s salary has to calculate it as manually and by only using calculator that is prone to error and it confusing in result would be mistakes in calculating salaries. They have encountered many problems on their Payroll System. The employee’s complain about their pay slip they do not see the accurate amount and what the things that are deducted to their salary.

In addition, the payroll administrator is having difficulties and takes a lot of time Calculating time sheets, wages, taxes, issuing checks, verifying the payroll and recording payroll transactions since there is so many employees in company to be pay. The pay slips for the employee also have to write it as manually or type using type writer. If the payroll process is not systematic, it will make more difficult in searching for particular detail. This problem will make the transaction slow and redundancy of work.

Furthermore, the employee’s record such as cash advance, daily time record, payroll, etc has an insufficient security and back-ups are since it is keep as document in a filing cabinet. This document might be seen or be viewed by an unauthorized person and in case a certain data will be lost it would be hard to retrieve. The entire document has to keep in the secure place to avoid see or steal by other people. All of this problem can be a burden for the payroll administrator before the pay slip can give to employee.

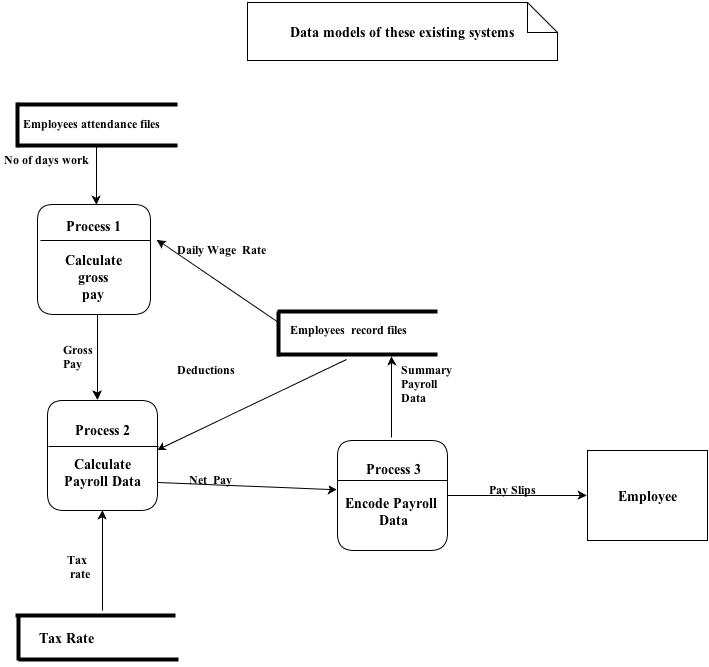
* + 1. **Process models of existing systems contributing to the problem**



**Figure 1.1 Process model of existing system**

*It is the flow on how the company process their payroll*

* + 1. **Data models of these existing systems**

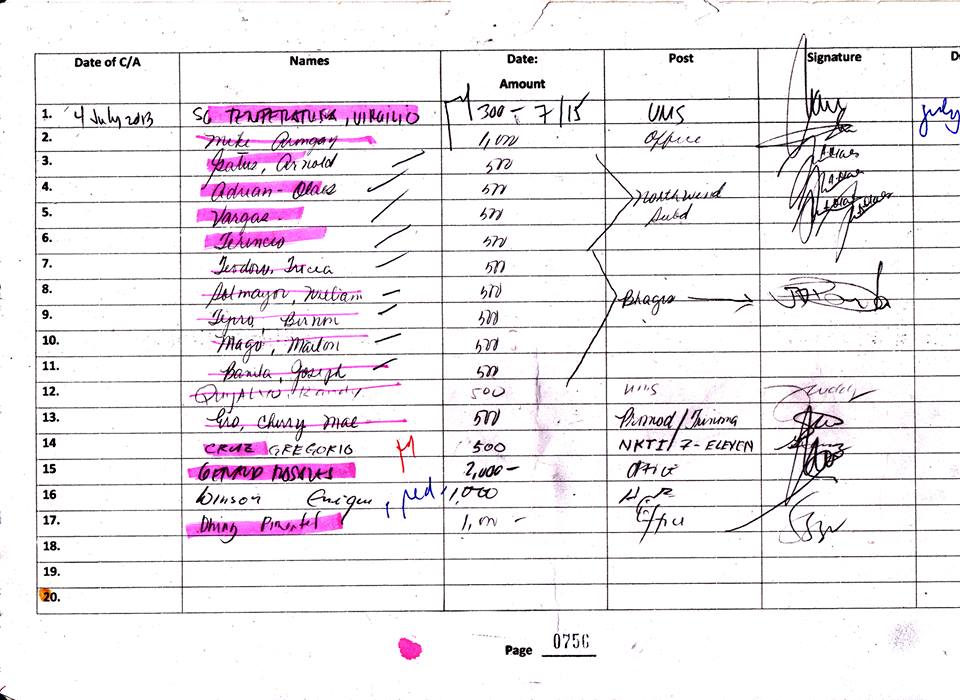
****

**Figure 1.2 data model of their existing system**

*It states on how they calculate employee salary*

* + 1. **Data ( or any form of statistics ) that may be relevant to prove existing and seriousness of the identified problems**

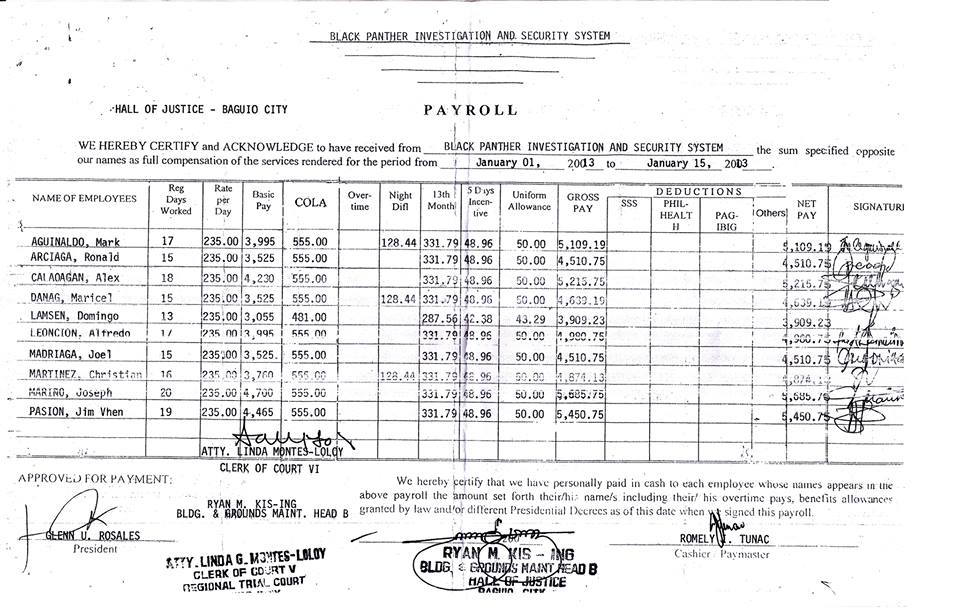
Cash advance form



**Figure 1.3 Data gathered from the company**

*Manual cash advance form*

Payroll record form



**Figure 1.4 Payroll record of the company**

*It state on how the company record their payroll*

* 1. **REVIEW OF EXISTING ALTERNATIVES**

Although the Northern Star Investigation and Security Agency Inc encounter so many problems in their payroll, the payroll administration handle related problems are part of as well. At the end of the working week the payroll administrator uses a batched of timesheets, together with employee record cards from a payroll file, to calculate the pay for each employee. Since they are using manual computation it takes so much time and they can’t avoid having mistake. It is hard for them to compute each employee’s salary. After they have computed all employees’ salary they write it in their payroll form using a typewriter. If an employee is paid an incorrect amount things that issues dealt with by this department. Likewise, tax filing and deduction errors are payroll matters as well.

There are employee’s that complain about their pay slip they did not see the accurate amount and what the things that are deducted to their salary. So in order to solve such problem, the payroll administrator will have to review on particular mistakes and if it was proven that there was really some mistake on that payroll transaction. They will implement the corrections for your incorrect pay it will be processed during the next payroll and will be adjusted accordingly to what has been mistaken on employee’s payroll. If the company owed the employee’s pay, including overtime, or benefit, the payment will be made on the next payroll process date. If you were overpaid, the amount owed will be deducted from the upcoming pay.

Since everything in payroll are bunch of paper works there’s an incident that a document have been lost. The one who lost it will be responsible and will pay to the corresponding amount. It will be deducted to his/her salary.

In order to eliminate and solve the problem regarding on the encountered problems on their manual payroll system we would like to propose a computerized payroll system for Northern Star Investigation and Security Agency Inc.

The develop payroll system will make the work process and the performance become faster for the payroll administrator. The payroll system will help to reduce man hours per job and reduce the risk of lost information. In addition, using the payroll system will also reduce the cost because it no needs many staff to handle payroll transaction. The systems will also make it paperless and store in a secure location and would be easy to restore.

The system will be an advantage for the company since it is computerized one of its feature is to have an automatic calculation of salary including deduction and allowance. This will make more easy for the administrator pay the employee’s salary with error free because it calculated by the system. But of course the exact calculated amount of salary will depends on the input number of working days and the input amount of deduction and etc. By the help of the system the administrator will avoid incorrect payment for the employee. The Pay slip will also be generated automatically by the system. There is no need for the administrator to write pay slip manually or type using type writer.

The administrator may also check each employee’s payroll information in a fastest and easy way by the use of a search box. He/she can locate the information of an employee in just minutes. In terms of losing important data, the system will be a great advantage because it is capable of restoring and back up the data to avoid serious problem cause by the lost payroll data. By the use of payroll system the payroll administrator can change the detail of employee if it is wrong, then the correction will be done immediately.

**CHAPTER 2**

**REVIEW OF RELATED LITERATURE**

**2.0 INTRODUCTION**

The group conduct a research about the payroll system study just become our bases to make a project study.

According to Wikipedia “payroll is the sum of all [financial](http://en.wikipedia.org/wiki/Financial) records of [salaries](http://en.wikipedia.org/wiki/Salary) for an employee, wages, bonuses and [deductions](http://en.wikipedia.org/wiki/Tax_deduction). In accounting, payroll refers to the amount paid to employees for services they provided during a certain period of time”.

According to Genesis Trias “A payroll system is an important application that should be used by businesses of any size. These systems are a guaranteed way to make sure your payroll is done on time and in compliance with all taxing authorities”.

According to the blogger Subbarao (2011) “Using Payroll software is cost worthy as you need to pay a single time for the installation and then need not pay on a regular basis for accountants. Someone with simple knowledge of a computer will be able to operate this system.”

We have got a lot of information at the internet through searching. We also have found different existing payroll system and there features that we can use as a basis for the development of our computerize payroll system. In addition there are also books available and free to read on Google Book about the payroll management.

* 1. **RELATED LITERATURE**
     1. **Foreign Literature**

In a [company](http://en.wikipedia.org/wiki/Company), payroll is the sum of all [financial](http://en.wikipedia.org/wiki/Financial) records of [salaries](http://en.wikipedia.org/wiki/Salary) for an employee, wages, bonuses and [deductions](http://en.wikipedia.org/wiki/Tax_deduction). In accounting, payroll refers to the amount paid to employees for services they provided during a certain period of time. Payroll plays a major role in a company for several reasons.

From an accounting perspective, payroll is crucial because payroll and payroll taxes considerably affect the net income of most companies and they are subject to laws and regulations (e.g. in the US payroll is subject to federal and state regulations). From an ethics in business viewpoint payroll is a critical department as employees are responsive to payroll errors and irregularities: good employee morale requires payroll to be paid timely and accurately. The primary mission of the payroll department is to ensure that all employees are paid accurately and timely with the correct withholdings and deductions, and to ensure the withholdings and deductions are remitted in a timely manner. This includes salary payments, tax withholdings, and deductions from [paycheck](http://en.wikipedia.org/wiki/Paycheck). ([http://en.wikipedia.org/wiki/Payroll. Retrieve August 11](http://en.wikipedia.org/wiki/Payroll.%20%20Retrieve%20August%2011), 2013)

According to Subbarao “Payroll is a very vital department in any organization. This is a department which will highly be affecting the net income of the organization. One needs to concentrate on the payroll calculation. There are various factors of tax deduction involved in it. One needs to be extra careful while calculating the taxes and ESIS and EPF. You may consider going for applying software for doing the service. There is software available on the internet that can give an end to end solution for your problem related to payroll.”

The software once set with the deductions and calculations can produce the Payroll in seconds for each employee. The software is to assist you with the job hence it needs to be programmed with the systems of your organizations. Using software is also cost worthy as you need to pay a single time for the installation and then need not pay on a regular basis for accountants. Someone with simple knowledge of a computer will be able to operate this system. Before buying any of the software, try to check out the software feed back with the existing clients and make online surveys of a number of software available to see which one is for your system.

([http://bestpayrollsystem.blogspot.com/2011/01/payroll-system-and-its-importance-in.html. Retrieve August 10](http://bestpayrollsystem.blogspot.com/2011/01/payroll-system-and-its-importance-in.html.%20Retrieve%20August%2010), 2013)

According to Genesis Trias “ payroll system is an important application that should be used by businesses of any size. These systems are a guaranteed way to make sure your payroll is done on time and in compliance with all taxing authorities. Tax laws and rates have continually changed in recent years, new tax credits, expiration of old credits and rate changes can all affect the outcome of your payroll.”

These systems also can help the business owner calculate the amount of taxes they owe the government at the end of each quarter. Failure to submit the correct taxes each quarter is the number one reason that business owners find themselves in trouble with the Internal Revenue Service.

Payroll systems are very simple to use. They can easily be incorporated into any business applications being used by the company. The use of this type of system in your company will help ensure that you remain up-to-date with the current tax laws and your obligations to the different taxing authorities. (<http://ezinearticles.com/?What-Is-a-Payroll-System?&id=5681334>. Retrieve Aug. 10, 2013)

* + 1. **Local Literature**

## According to Adrienne Valdez Philippine govt to use central payroll system Department of Budget and Management (DBM) Secretary, Florencio Abad, recently announced a central payroll system, included in the national payment portal to be launched next year.

“By next year, the National Treasury will be the one to remit premiums to the Government Service Insurance Systems and withholding taxes for the Bureau of Internal Revenue. This is to prevent problems of non-remittance of premiums and taxes,” Abad said in an official statement.

As Abad disclosed the program during his presentation of the National Action Plan on Anti-Corruption and Good Governance, he further explained the government’s plan to utilize information technology to be more accountable to the people.

According to Abad, an integrated financial information management system aimed to improve public finance management is set to be in place by 2016.

Future Gov Asia. By Adrienne Valdez , 2 October 2011

## According to [Clarice Africa](http://www.futuregov.asia/users/34/) the Philippines modernizes civil service payroll system “Switching to electronic means for the management of payroll processes is hardly a new concept. Its part for the course in the private sector, where companies are able to generate significant cost savings by abolishing check payments and the manual processing of monthly salaries and withholding fees. Similarly, the NPS will relieve the government’s human resource units from the burden of handling cash and manually keeping track of all payroll activities,” the Budget Secretary said.

Future Gov Asia. by [Clarice Africa](http://www.futuregov.asia/users/34/) | 31 July 2013

QuickPayroll is the brainchild of Wikonec Inc. located in DOST-PEZA Open Technology Business Incubator inside the University of the Philippines Diliman. The company is duly registered under the Security and Exchange Commission and operates in association with the Department of Science and Technology – Advance Science and Technology Institute (DOST-ASTI).

QuickPayroll solution can easily run and compute payroll for different business needs. Whether you need computation for irregular employees, commissions, deductions and loans or automatic computation with current time sheet, computation for leaves, absences and overtime; our system can do it for you quickly. **(**<https://www.quickpayroll.ph/Home/About>. Retrieve Aug. 10, 2013)

* 1. **RELATED STUDIES** 
     1. **Foreign Studies**

According to Bragg “A payroll system that is just as popular as outsourcing is the in-house computerized system. Payroll software is very inexpensive, as it is now bundled with accounting software that cost just a few hundred dollars. More comprehensive system for use with large numbers of employee’s are much more expensive, but are cost effective solution for large entities.”

There are benefits that can gain when using a payroll software, it is not more expensive and the company can lessen their expenses since using a software can minimize the labor cost of the company.

Essencial Of Payroll System Management and Accounting. Steven M. Bragg . Publish Jhon And Wiley & Sons Inc. 2003, Hoboken New Jersey. Page 10

The Federal Personnel Payroll System is a modern, mainframe base, portable, integrated, on-line, and real-time personnel and payroll system. The system provides personnel and payroll support to numerous agencies, servicing accounts. Te system is customer support driven, creating and generating the full life cycle of personnel transaction, enabling agencies to maintain records electronically. It handles all current regulations including specialized pay, garnishments, and special appointment programs.

It is highly sophisticated and comprehensive database management system providing and extensive array of human recourse and payroll automation support within its core, including requirements prescribe by clients agencies.

All data is defined in a single database which will eliminates redundancy and multiple update routines. The single database concept also provides for ease of maintenance and ensures data integrity.

* + 1. **Local Studies**

The City Accounting Department of Cagayan de Oro City Hall Computerized Payroll System has introduced an upgraded computerized payroll system for permanent employees in a bid to improve its services and maximize resources. "This project, which was conceived two years ago, basically aims to improve our services particularly in the processing of payrolls for permanent employees," City Accountant Wilma Polley-Rugay told payroll in-charge of the different departments and offices at City Hall during a briefing and orientation on the concept of the one-month payroll system held Friday last week at the City Council session hall in Cagayan de Oro City. With the added features of the new computerized payroll system, Rugay said the preparation, processing and payment of payroll system to permanent employees would be hastened and fast-tracked.

Adopting of the one-month payroll system is also beneficial, not only to the City Accounting Department, but also to the payroll-in-charge as it would save time, energy and resources. “Employees who handle payroll preparation and processing can now attend to other office needs and concerns as the new payroll system will lessen their workload," she added.

Under the new payroll system, preparation and processing of payroll, which reflects the accrued, amount payable every 15th and 30th day of the month, done only once. Unlike the time-consuming old payroll system, preparation and processing of payroll is done twice a month covering the first half and second half of the month. ([www.sunstar.com.ph](http://www.sunstar.com.ph) . Retrieve Aug. 10, 2013)

The Computerized Payroll System is a window based program especially designed to facilitate and simplify the monthly preparation of general payroll and related reports such as Standard computerized payroll system for use in all DOH offices, Allows faster and more accurate computation of monthly gross income, deductions and net salary, Faster and less-resource-consuming generation of General Payroll and other payroll related reports, Security and integrity of payroll data and information. The Features of computerized payroll system of DOH are Graphical User Interface, Interactive and menu-driven program, Systematic maintenance and retrieval of employee records, Flexible as it provides options to include additional fields for other compensation and deductions unique to an office or unit, Automatic computation of monthly net income, GSIS, PAG-IBIG, Withholding Tax and other deductions, Y2K compliant, Fast and easy generation of the General Payroll and all other payroll related reports which include Monthly reports like GSIS and PAG-IBIG remittances, Denominations Report and Leave Credits Report and annual Year-end Tax Reports, Password security to ensure the integrity of data. General payroll reports such as Payroll Summary sheet, Pay slip, Leave Credits Report, Withholding Tax Statement (W-2), BIR Remittance Reports, GSIS Remittance Reports, PAG-IBIG Remittance Reports, Other User-defined Reports (PERA, ACA, PNB, etc..). (www2.doh.gov.ph. Retrieve Aug. 10, 2013)

It covers the process of preparing the Pay slip of each employee, the regular and job order employee, keeping of records safety and computing the exact wages of each employee. Including the stored report list of employee such as Monthly Salary, Basic salary, overtime, Gross payment, Net pay, and Deduction such as withholding tax, SSS, and Medicare, SSS loan, Advances, Multi-purpose loan, PAG-IBIG Contribution, CAPELCO loan, Cash Advance, Bond, Energy Sales, Motorcycle, Canteen, under time, Cigna, WELCA, union loan, Lot/housing loan, health Care and AR.

The proposed Payroll System for Capiz Electric Cooperative is expected to generate Pay Slip of Employees, Report list of employees, Remittances for respective agencies, Payroll register of regular and job-order employees, and Payroll form for every pay period. (<http://gracepioquidgara.blogspot.com/2009/10/chapter-2.html>. Retrieve Aug. 10, 2013)

* 1. **SYNTHESIS AND SIGNIFICANCE OF THE STUDY**

Through the help of internet we gather different literature and studies about the Payroll System. It helps us to understand more about the importance and the process of conducting a payroll system. Especially on how to calculate salary every month and what are the deduction included.

We have learned from Payroll System different features of computerize system. According to these features the salary, deduction benefits are computed automatically and be able to generate pay slip. It is also important to generate tax report to be submitted to the government office. For the Federal Personnel Payroll System it’s important to have a database to maintain and organize all the records of the company.

The literature review may also from real user. The developer can get information by doing an interview to the companies or industries. This is very important research because by doing interview we can determine the requirements and the needs of the user and can give idea on how to improve the develop system.

It is very important to gather such information so it can help the developer to understand and can give some idea in order to achieve their goal or objectives. These will serves as guide and background in developing a system because without collecting or gathering information we cannot get an idea on how to develop the system and what are the requirements needed.

**CHAPTER 3**

**ANALYSIS AND DESIGN**

* 1. **SOFTWARE REQUIREMENTS SPECIFICATION DOCUMENT**
     1. **INTRODUCTION**
        1. **Goals And Objective**

The purpose of this document is to define the requirements for Northern Star Security Agency (NSSA) Computerized Payroll System. These requirements form boundaries and constraints for the project’s design process as well as providing a reference for the development of the final product to achieve the goal and meet the outcome of the project. The document also provides an informative context and illustrates by the used of some modeling diagram to give much better understanding about the software.

The software is a window base and can be integrated to a different subsystem in a certain company. The user interface is design by the used of java language and have a centralize database connected to the network to share data information.

Northern Star Security Agency (NSSA) Payroll software is intended to use toward paying employees what they earn within a reasonable time frame and accurately tracking paycheck amounts as well as tax and benefit withholdings. Payroll systems should be thorough without being overly complex and cumbersome, and they should be designed thoughtfully enough to work well with a minimum of oversight. Having a well-designed payroll system is in the best interest of employees, business owners and human resources staff we just want to innovate the daily routine or traditional in terms of their payroll system in Northern Star Security Agency (NSSA).This system most important motivators for an employee in their company that pays its employees the correct wage on time engenders loyalty from them because they know they can count on their firm to deliver as promised.

* + - 1. **Statement of Scope**

This section contains the description of the software functionality and detailed requirements of the project.

The software allows handling of employee’s payroll information. The system has two types of user the Payroll Administrator and an Accountant. Admin and accountant are prompt to provide their corresponding user name and password, while the employee type their ID number to view their payroll status.

The payroll admin controls the entire system while the accountant is only allowing process the payroll of the employee, accountant also can view and print personal payroll detail but not allowed to make any changes on payroll record. Every end of the week the administrator will collect the attendance of employees from the Personnel Management Info and Assignment Monitoring and import the numbers of days work to the payroll system.

Every 15th and last day of month the employee’s salary is auto calculated by system such as gross pay, deduction and net pay. The administrator will make necessary adjustments if there are changes on their pay such as overtime pay, night differential, holiday pay, COLA(Cost of Living Allowance), allowance, incentives, leave with pay, sick pay, bonus, deduction such as late, SSS contribution, Pag-ibig, Philhealth, Withholding Tax and etc. Every year employees 13 moth pay is also calculated.

The exact amount of SSS, philhealth, pag-ibig, and withholding tax will be deducted by using their corresponding table of contribution and with holding tax table from BIR. All deduction are deducted every pay day period.

As the salaries are calculated the admin or accountant will be able to print pay slip or pay stubs for each employee. The various payroll reports may also generate and print by the administrator and accountant using the software. All payroll information is stored in a database and maintain by the administrator.

The system is also capable of showing payroll schedule notification only if the administrator will set a payroll schedule date and time for payday and tax fillings date.

The table will show the various categories of user requirements. Each requirements indicate priority whether it is high, medium or low to determine which should be given priority upon implementation.

User requirements for Northern Star Security Agency (NSSA) Payroll software

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement No. | Priority | Reference | Description |
| **Access privilege** | | | |
| R1 | High | Customer | There should be two levels of access one for Administrator, and one for accountant. |
| R2 | High | Customer | Administrator allowed making any changes on the payroll. |
| R3 | High | Customer | Administrators are allowed to use reminders. |
| R4 | Med | Customer | Administrator allowed printing pay slips. |
| R5 | High | Customer | Accountant allowed to process the payroll |
| R6 | Med | Customer | Accountant also allowed printing pay slips. |
| **Security** | | | |
| R7 | High | Customer | Administrator and accountant should be log on using their unique username and password. |
| R8 | High | Customer | The password will be assigned to each user |
| R9 | High | Customer | Password must be indicated alpha numeric combination |
| R10 | Med | Customer | Employees use their Id number to log on. |
| R11 | Med | Code works | System will shut down after 3 unsuccessful attempts, until password is reset. |
| **Reminders** |  |  |  |
| R12 | Med | Customer | Administrator will set a pay date payroll notification. |
| R13 | Med | Code works | Date will be selected from the calendar. |
| R14 | Med | Code works | User is allowed to enter 8-16 characters. |
| R15 | Med | Code works | Reminders will pop-up when the admin will log-on. |
| **Payroll Information** | | | |
| R16 | High | Customer | Admin can add, edit and print employee salary rate record |
| R17 | High | Customer | Accountant can manage the salary process and print employee salary rate record |
| R18 | High | Customer | Employee’s information shall contain: name, age, address, contact, gender, number, status and position |
| **User Interface** | | | |
| R19 | High | Customer | The system shall have user name and password interface for administrator and accountant |
| R20 | Med | Customer | The background of all system shall be red |
| R21 | Med | Customer | The system responds to all request within a 20 seconds. |

* + - 1. **Software Context**

This payroll software is a very efficient one, which maintains database of payroll records. And this software is less expensive and also reduces complexity. In this software we use Java and MySQL and JDBC for the database connectivity. The database search is very fast such that the system responds to all requests within 20 seconds.

The Agency payroll system makes it easy for the company to process payroll and meets your organization’s needs. In addition, the Payroll System allows you to create unlimited user defined data fields and user configured screens, which means a fully customized system to work the way that you work. This comprehensive payroll system addresses the unique payroll requirements of the Service management agency. In addition, it includes a robust general ledger interface and balancing tools. You gain peace of mind knowing that you are diminishing the chance of human errorassociated with manual systems and ensuring tax and regulatory compliance.

It Handle all of your end-to-end payroll needswith the and easy to use, easy to access Payroll System. Build efficiencies with your payroll process by decreasing administrative tasks and data entry. From paychecks to W2 forms, Process checks at your facility and sends to the corporate office for approval. Or process and approve payroll at the corporate office and cut checks at your facility.

* + - 1. **Major Constraints**

The software is to run on a system, with SQL Therefore the systems should be properly connected to the server to access database. The Systems should be good enough to perform all the actions fast for example it is recommended that system should be no older than PII .The RAM should be not less than 128 MB.

* + 1. **USAGE SCENARIO**
       1. **User Profiles**

Administrator the administrator is responsible for entering employee’s payroll information

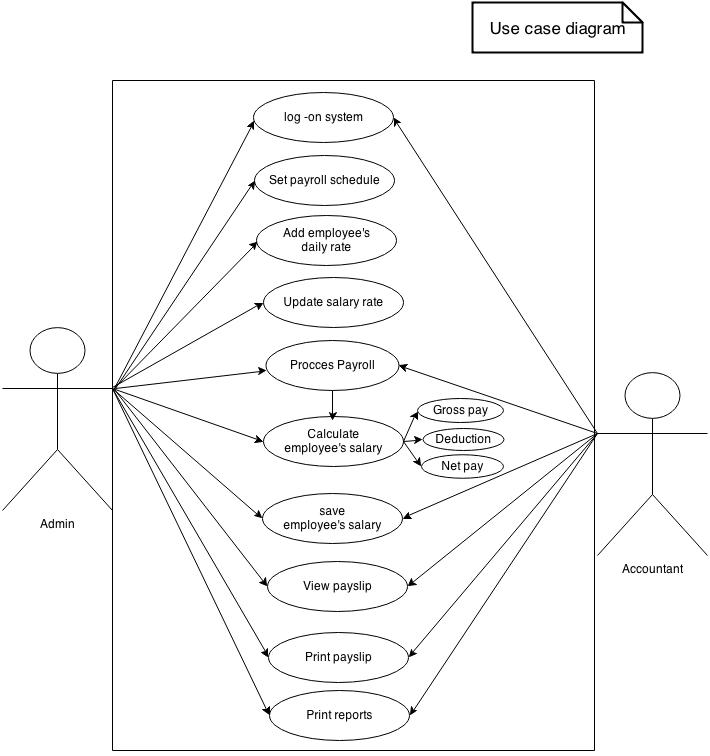
Accountant Accountant process employees payroll, encode

* + - 1. **Use-Cases**

The following use cases are typical interactions between the external environment and the internal software system.

1. Log on to system
2. Get employee record
3. Add employees salary rate
4. Compute Salary
5. Update employee salary rate
6. View payroll information
7. Print pay slips and reports
8. Sign out
   * + - 1. **Use-Case Diagram**

Use Case diagrams show three actors the various activities the users can perform on the system. The System is something that performs a function. They model the dynamic aspects of the system. It provides a user’s perspective of the system.

****

**Figure 3.1: Use-Case Diagram of Northern Star Security Agency (NSSA) Payroll System**

* + - * 1. **Use-Case Description**

|  |  |
| --- | --- |
| Use Case | Log on to System |
| Primary Actor | Payroll Admin |
| Goal In Context | To gain access to the system |
| Preconditions | Payroll Administrator should enter valid user name and password |
| Trigger | The payroll administrator must access to the system to perform his task. |
| Scenario | 1. Enter user name and password. 2. System will send the user name and password to the database and verify if it’s authorized. 3. The payroll administrator is given access to the system to perform his task. |
| Exceptions | User name and password is incorrect. |

|  |  |
| --- | --- |
| Use Case | Get employees record |
| Primary Actor | Payroll Admin |
| Goal In Context | Retrieve Employees Information |
| Preconditions | Employee’s record must exist. |
| Trigger | Admin need to view their employee’s personal information |
| Scenario | 1. Payroll admin must log-on to system 2. From home select employee record 3. Choose employees record 4. The system prompt for employees ID number 5. System request record from the database. 6. Employee’s information is displayed. |
| Exceptions | The account does not exist. |

|  |  |
| --- | --- |
| Use Case | Add employee salary rate |
| Primary Actor | Payroll Admin |
| Goal In Context | To add employees salary rate |
| Preconditions | Newly hired employee |
| Trigger | Payroll admin need to set employees daily rate. |
| Scenario | 1. The Admin must logon to the system. 2. Click employee salary rate 3. The system prompts to search employees ID. 4. The system sends information to database. 5. Employee’s basic information is being displayed. 6. The payroll admin must enter employee’s salary rate. 7. The system will store the information to database. |
| Exceptions | Record already exist |

|  |  |
| --- | --- |
| Use Case | Calculate Salary |
| Primary Actor | Accountant |
| Goal In Context | To calculate employees salary |
| Preconditions | Employee must be working in a company. |
| Trigger | Accountant need to compute employee salary every semi- monthly |
| Scenario | 1. Accountant must logon to the system 2. Select payroll compute from salary compute icon 3. The system prompt for employees ID 4. System will send information to database. 5. Information will be displayed. 6. Just click the compute and then save. 7. The system will store the information to database. |
| Exceptions | An employee does not work. |

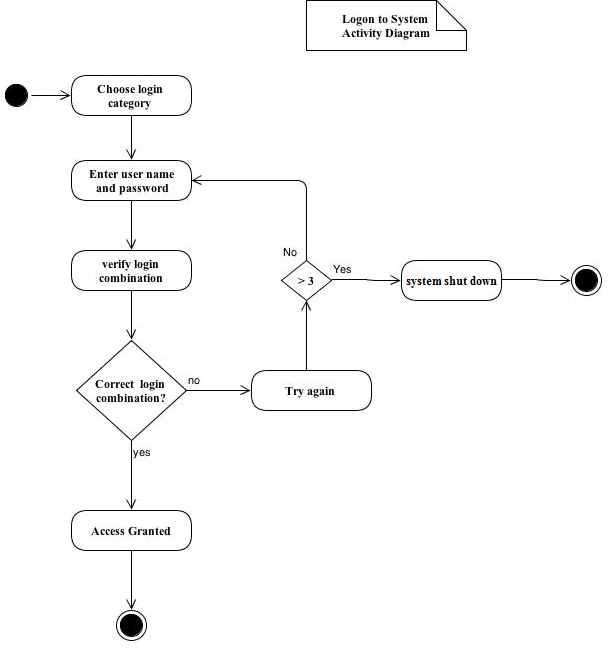
|  |  |
| --- | --- |
| Use Case | Update employee salary rate |
| Primary Actor | Payroll Admin |
| Goal In Context | To update employees salary |
| Preconditions | Employees must have an existing record |
| Trigger | Payroll admin needs to edit employees current salary rate |
| Scenario | * + - 1. The Admin must logon to the system.       2. Choose employee salary rate.       3. Search employees ID       4. System send information to database       5. System will display information.       6. Change salary rate.       7. Click update       8. System will store new information |
| Exceptions | No existing record |

|  |  |
| --- | --- |
| Use Case | View payroll information |
| Primary Actor | Payroll admin or accountant |
| Goal In Context | Retrieve payroll information. |
| Preconditions | There must be a record exist |
| Trigger | User need to view employee’s payroll records. |
| Scenario | 1. Payroll admin or accountant must logon to system 2. Select reports and choose payroll records 3. Choose payroll records 4. Search employees by ID. 5. The system will send information to database. 6. If record exists, system will display information. |
| Exceptions | Record does not exist |

|  |  |
| --- | --- |
| Use Case | Print payslip and reports |
| Primary Actor | Payroll Admin or accountant |
| Goal In Context | To print |
| Preconditions |  |
| Trigger | Print Monthly, semi-monthly or yearend reports |
| Scenario | * + - 1. User must log onto the system       2. Select reports on home       3. Select the category to be printed such as payslip, 13 month pay, allowance, bonus, statutory deductions, and payroll reports.       4. Press print       5. Records will be printed. |
| Exceptions |  |

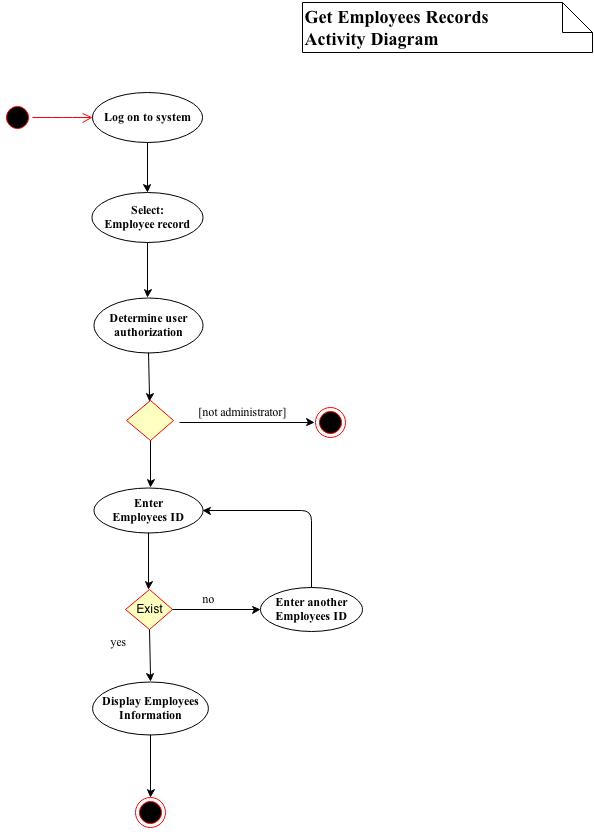
* + - 1. **Special Usage Considerations**
* Salary rate can be change according to the employee’s position.
* Statutory deduction rates can be change according to the changes issued by the government
  + - 1. **Activity Diagram**

The following activity diagrams show the actions that occur during particular use-case. Figure 3.2 shows the steps taken as an employee logs on to the computer system. Access is only granted if the correct user ID / password combination is entered. If entered incorrect user name more than three times system will automatically shut down.

****

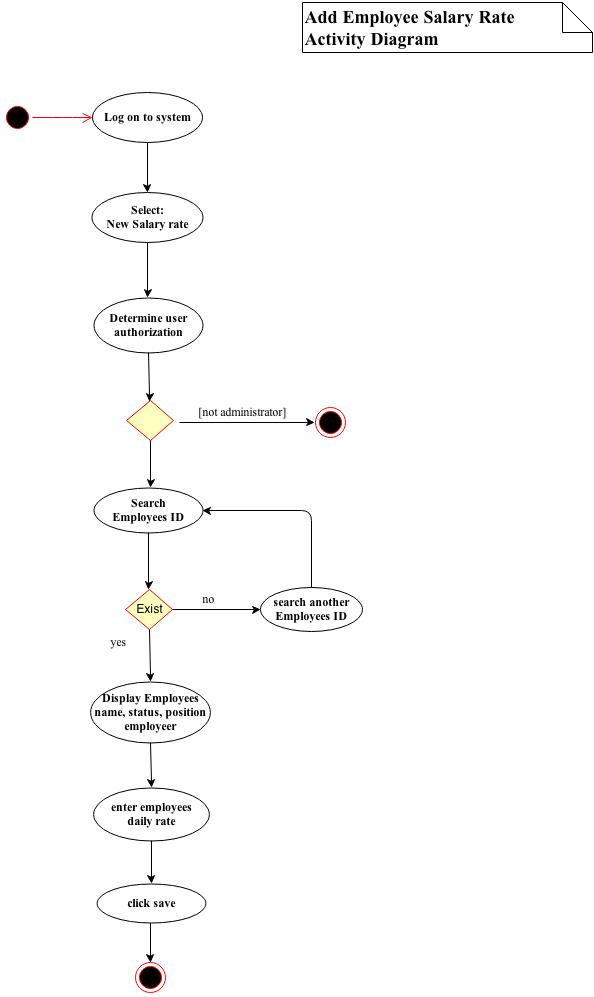
**Figure 3.2: Activity Diagram for System login**

In Figure 3.3 shows the activity involved in viewing employee’s personal information. The user needs the employees ID to search the employee’s information.



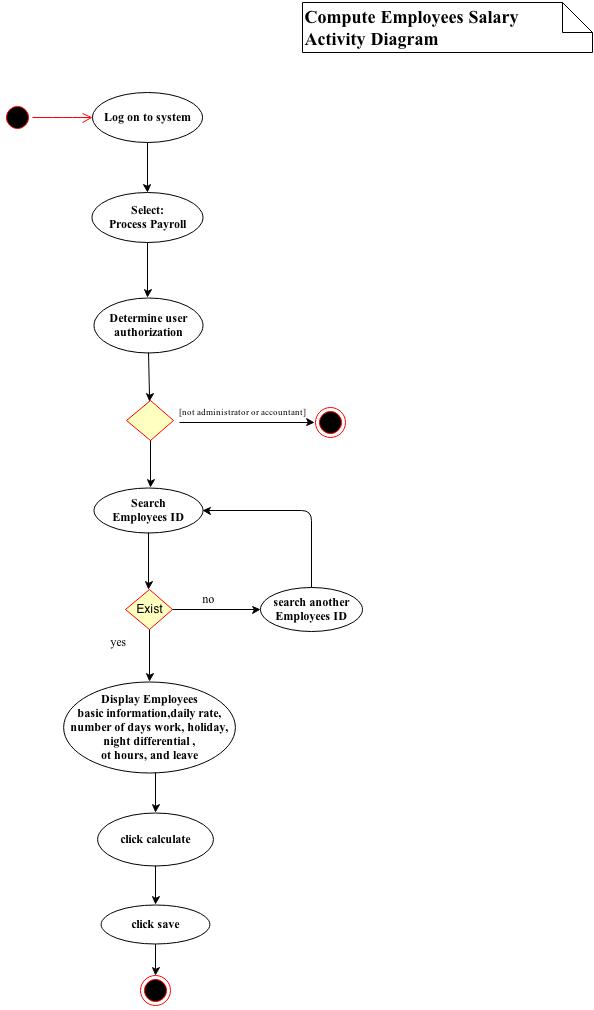
**Figure 3.3: Activity Diagram for Employees Personal Information**

In this Figure 3.4 shows the activity involved in adding employee’s salary rate. The administrator needs the newly hired employees ID to get employees basic information and modify the amount for daily rate.



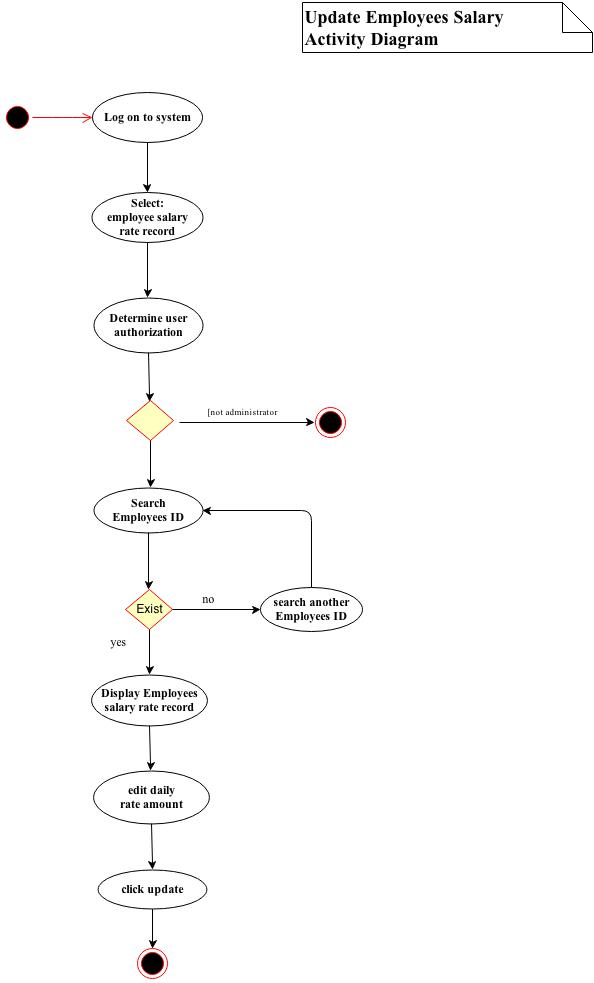
**Figure 3.4: Activity Diagram for Employees salary rate**

As the administrator logon to the system this figure 3.5 is about computation of employees salary. Select employee by ID. If ID exists it will display the employees basic information such as name, position, status, employer and daily rate. Then it will also display the employees DTR summary. The accountant needs to modify the allowance, COLA, bonus to get the gross pay and deduct all deduction to get the net pay.

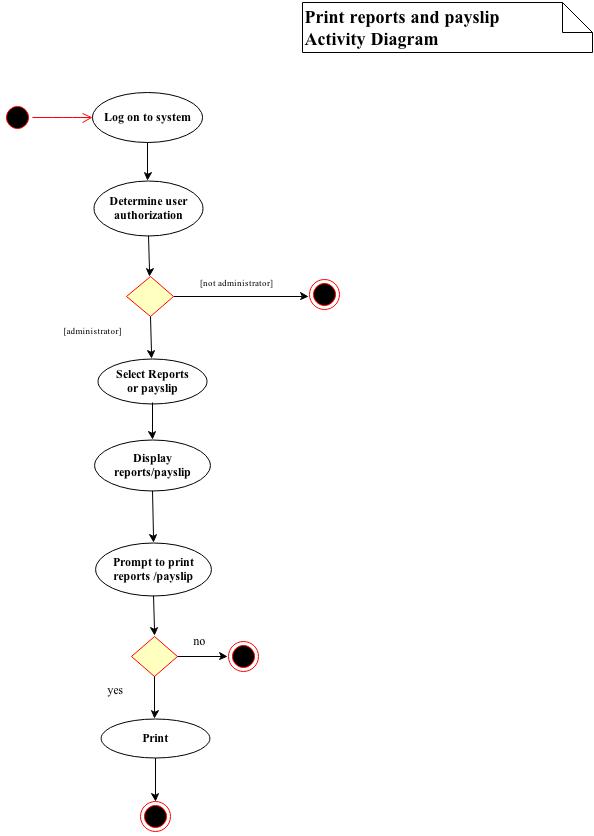
****

**Figure 3.5: Activity Diagram for Salary Calculation**

In this Figure 3.6, the admin can update employee’s salary rate by searching employees ID or name , then the admin can edit the daily rate amount and update the record.

 **Figure 3.6: Activity Diagram for updating employee salary rate record**

Printing payslip and reports is another activity that is allowed for the payroll administrator and accounting staff to perform in the system. In this figure 3.7 the employees needs a payroll records to print the reports and payslip.

****

**Figure 3.7: Activity Diagram for printing payslip and reports**

* + 1. **DATA MODEL DESCRIPTION** 
       1. **Data Objects**

## 13 Month Pay Data Object

|  |  |
| --- | --- |
| **13mp\_id** | Unique identification for 13 month pay record |
| Empid | Employees identification number |
| Name | Employees name |
| Amount | 13 month pay amount |
| date\_pay | Date of 13 month pay release |

## Allowance Data Object

|  |  |  |  |
| --- | --- | --- | --- |
| **allowance\_id** | Unique identification for allowance record |  |  |
| Empid | Employees identification number |  |  |
| Name | Employees name |  |  |
| allowance\_type | Type of allowance |  |  |
| allowance\_amount | Amount of allowance to be given |  |  |
| date\_paid | Date that allowance release |  |  |

## Bonus Data Object

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| **bon\_id** | Unique identification for bonus record |  |  |
| Empid | Employees identification number |  |  |
| Name | Employees name |  |  |
| bon\_amount | Amount of bonus to be given to employee |  |  |
| date\_pay | Date of bonus release |  |  |

## Pagibig Collection Data Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
| *pi\_colect\_no* | | Unique identification number for pagibig collection record |  |  |
| Empid | | Employees identification number |  |  |
| Name | | Employees name |  |  |
| ee\_contri | | Amount of employees contribution |  |  |
| er\_contri | | Amount of employer contribution |  |  |
| Total | | Total amount of employee and employer contribution |  |  |
| Date | | Date deducted |  |  |
|  |

## philhealth contribution Data Object

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *ph\_colec\_no* | Unique identification number for philhealth collection record |  |  |
| Empid | Employees identification number |  |  |
| Name | Employees name |  |  |
| ee\_share | Amount of employee share |  |  |
| er\_share | Amount of employer share |  |  |
| Total | Total amount of employee and employer contribution share |  |  |
| Date | Date deducted |  |  |

## SSS Contribution Data Object

|  |  |  |  |
| --- | --- | --- | --- |
| *sss\_colec\_no* | Unique identification number for SSS collection record |  |  |
| Empid | Employees identification number |  |  |
| Name | Employees name |  |  |
| ee\_cotri | Amount of employees contribution |  |  |
| er\_contri | Amount of employer contribution |  |  |
| Date | Date deducted |  |  |

## wtax Collection Data Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
| *wtax\_colec\_no* | | Unique identification number for tax record |  |  |
| Empid | | Employees identification number |  |  |
| Name | | Employees name |  |  |
| Tax | | Amount of withholding tax deducted from gross pay |  |  |
| Date | | Date deducted |  |  |
|  |

## Deduction Data Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
| *deduc\_id* | | Unique identification number for deduction record |  |  |
| emp\_id | | Employees identification number |  |  |
| Late | | Employees amount deducted for being late |  |  |
| Undertime | | Amount deducted on employees gross pay for undertime |  |  |
| Loan | | Amount deducted for loan payment |  |  |
| Sss | | Amount deducted for sss contribution |  |  |
| Philhealth | | Amount deducted for philhealth contribution |  |  |
| Pagibig | | Amount deducted for pagibig contribution |  |  |
| Tax | | Amount deducted for tax contribution |  |  |
| Totaldeduc | | Total of all deduction |  |  |
|  |

## payroll\_loan Data Object

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *loan\_id* | Unique identification number for loan record | |  |
| emp\_id | Employees identification number |  |  |
| date\_loan | Employees name |  |  |
| loan\_amount | Amount loan |  |  |
| start\_deduc | Date to start the loan payment | |  |
| loan\_status | Payment status |  |  |
| Amortization | Amount to be paid every payday. |  | |

## payroll\_login Data Object

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | |  |  |  |
| *userId* | | | | | | | Unique identification number for login |  |  |
| User | | | | | | | Unique username |  |  |
| Pass | | | | | | | Password |  |  |
| Name | | | | | | | Name of user |  |  |
| Des | | | | | | | Designation of user |  |  |
|  | |  |  |  |  |
|  |

## Pay compute Data Object

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
| *payrol\_id* | | Unique identification form payroll records |  |  |
| Empid | | Employees identification number |  |  |
| date\_fr | | Date of payroll period that employees start to work |  |  |
| date\_to | | Date of payroll period end |  |  |
| pay\_date | | Payroll date |  |  |
| no\_day\_work | | Number of days that an employee had work |  |  |
| daily-rate | | Employees daily rate |  |  |
| basic\_pay | | Employees basic pay |  |  |
| Ot | | Employees overtime pay |  |  |
| Otnd | | Employees night differential pay |  |  |
| duty\_on\_dayoff | | Duty on day off pay |  |  |
| sp\_holiday | | Special holiday pay |  |  |
| l\_holiday | | Legal holiday pay |  |  |
| total\_alowance | | Total of allowance an employee recieve |  |  |
| total\_bunus | | Total of all bonus an employee |  |  |
| 13mp | | 13 month pay |  |  |
| leave\_pay | | Employees leave pay |  |  |
| Deduction | | Total of all deduction |  |  |
| gross\_pay | | Employees total compensation |  |  |
| net\_pay | | Employees net salary |  |  |
|  |

## salary\_rate Data Object

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| *rate\_id* | Unique identification form employees salary rate records |  |  |
| Empid | Identification number of employee |  |  |
| Name | Employees name |  |  |
| Status | Employees civil status |  |  |
| Position | Employees position |  |  |
| Employeer | Person where the employee is working |  |  |
| daily\_rate | Employees daily rate |  |  |
| effective\_date | Date |  |  |

* + - 1. **Relationships**

To compute employee’s salaries there must be one accountant representative that will be log on to the system. One administrator may add many employees to the payroll records so the relationships exist are one-to-many relationship. If the system is not having a payroll records the administrator cannot update while the accountant can’t compute payroll.

As an employee has been hired in a company he/she is given only one daily rate. The relation between employee and rate is one-to-many relationships. Since an employee cannot have many daily rate, wage rate, same as the relationship between employee and the salary. Since each employee can get only one salary every pay day.

The payroll could process many employees salary in a month, if there is no employee there is no payroll. The relationship between payroll and employee is one to many.

Before payroll can be process each employee could get many allowances in a month like COLA, transportation and meal allowance. That means one-to-many relationships exist.

Each employee can also get many bonuses in a month therefore one-to-many relationships exist.

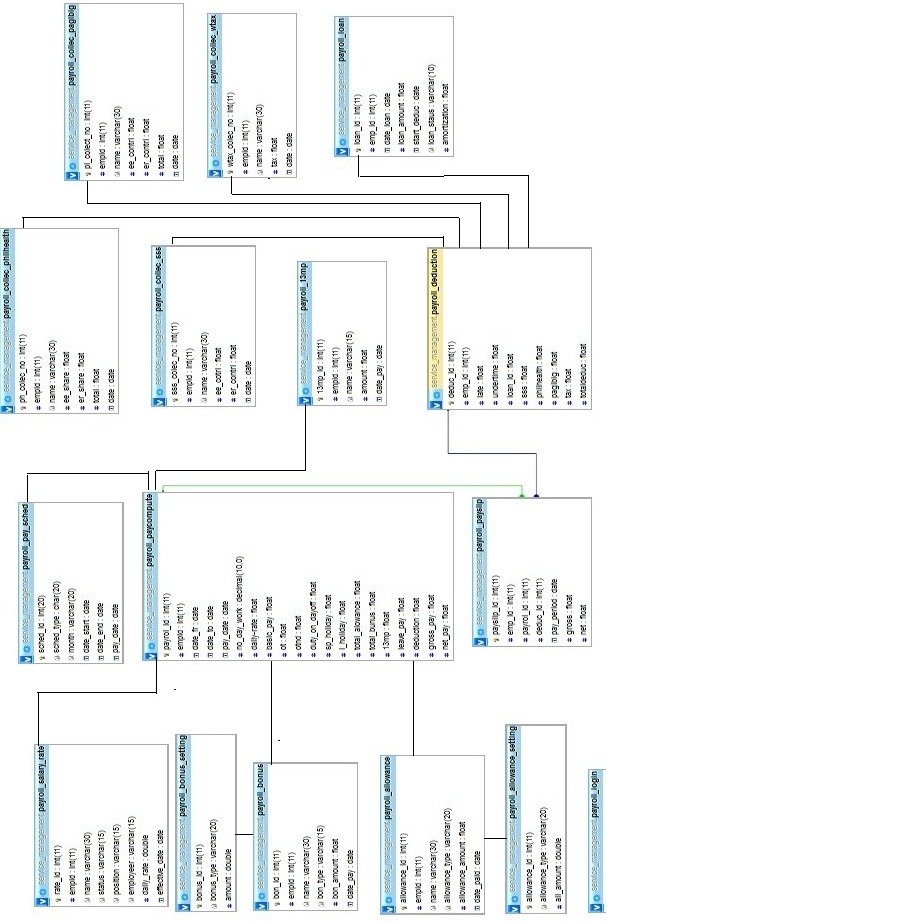
Every pay day on last day of month employee salary could have many deductions such as sss, tax, philhealth, pag-ibig and loan. Therefore the relationship between employee and deduction is one-to -many.

In a month there are several holiday and that is categorize as special and legal Holiday. So an employee could have many holiday pay in a month. There is a corresponding pay if an employee will work or may not work on a holiday still. Therefore one-to-many relationships exist.

In terms of setting a notification only the administrator can set payroll schedule. Admin may be able to set many schedule. The relationship between administrator and reminders would be one-to-many relationships.

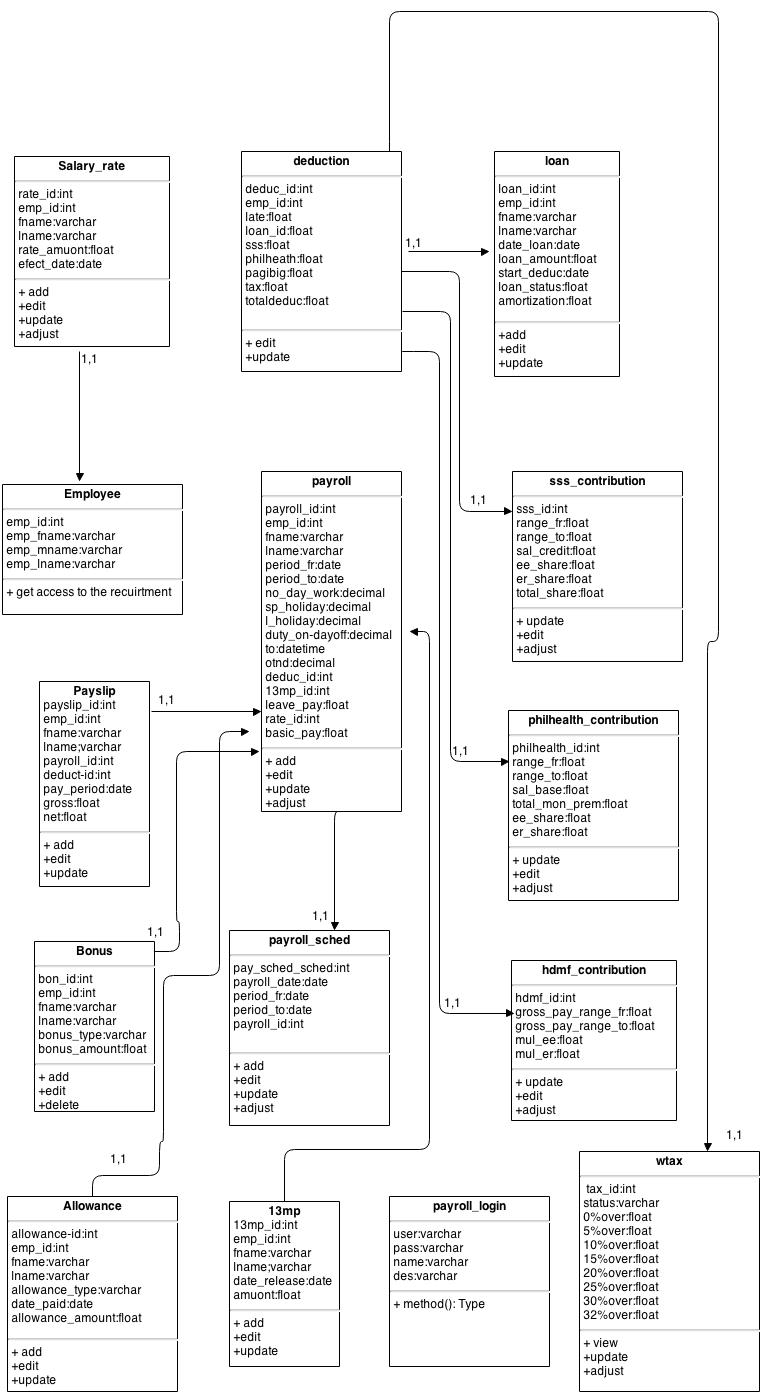
* + - 1. **Complete Data Model**

The relationship between data objects describe in this section.

****

**Figure 3.8 Relationship Diagram of Northern Star Security Agency Payroll System**

* + 1. **FUNCTIONAL MODEL DESCRIPTION** 
       1. **Class Diagram**

****

**Figure 3.9: Relationship Diagram of Northern Star Security Agency Payroll System**

* + - 1. **Software Interface Description** 
         1. **External Machine Interfaces**

The software will be capable of printing pay slips and reports on a local or network printer.

* + - * 1. **External System Interfaces**

The NSSA payroll system will be communicate only through land base connection

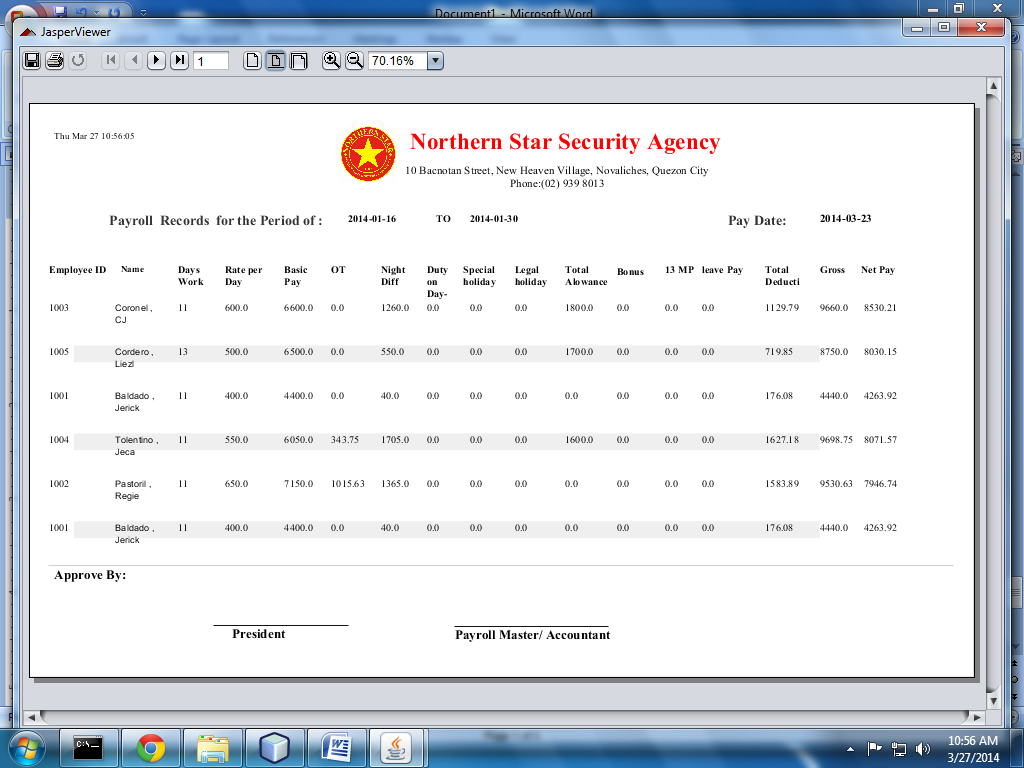
* + - * 1. **Human Interface**

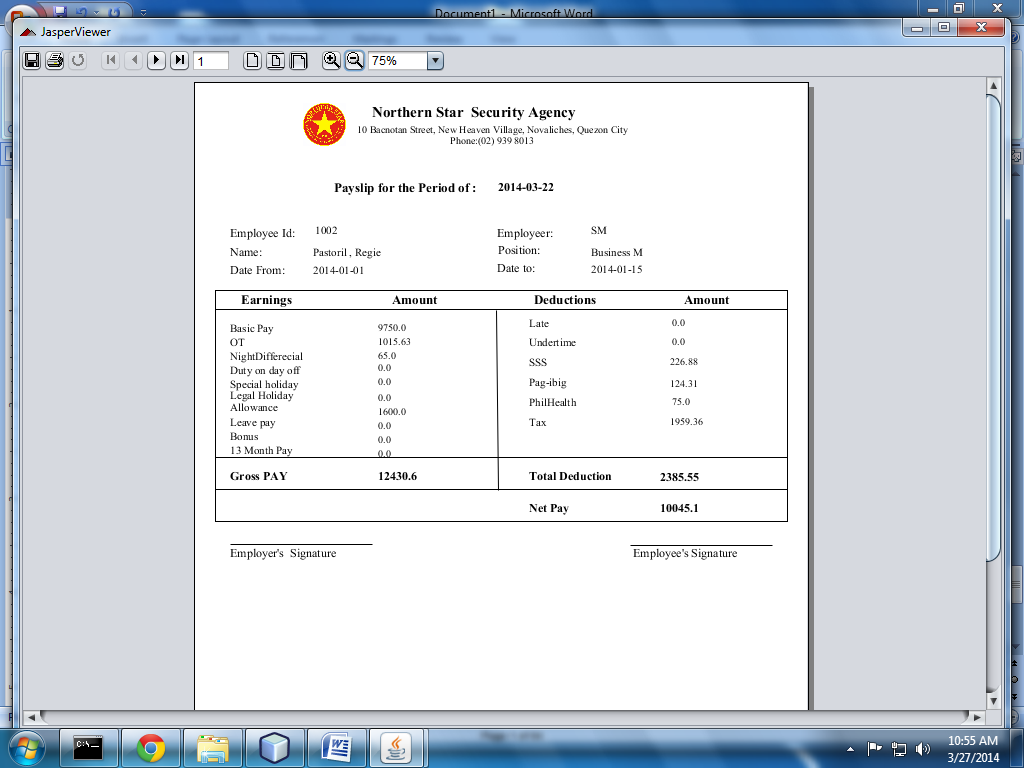
The Web pages shall permit complete navigation using the keyboard alone, in addition to using mouse and keyboard combinations.

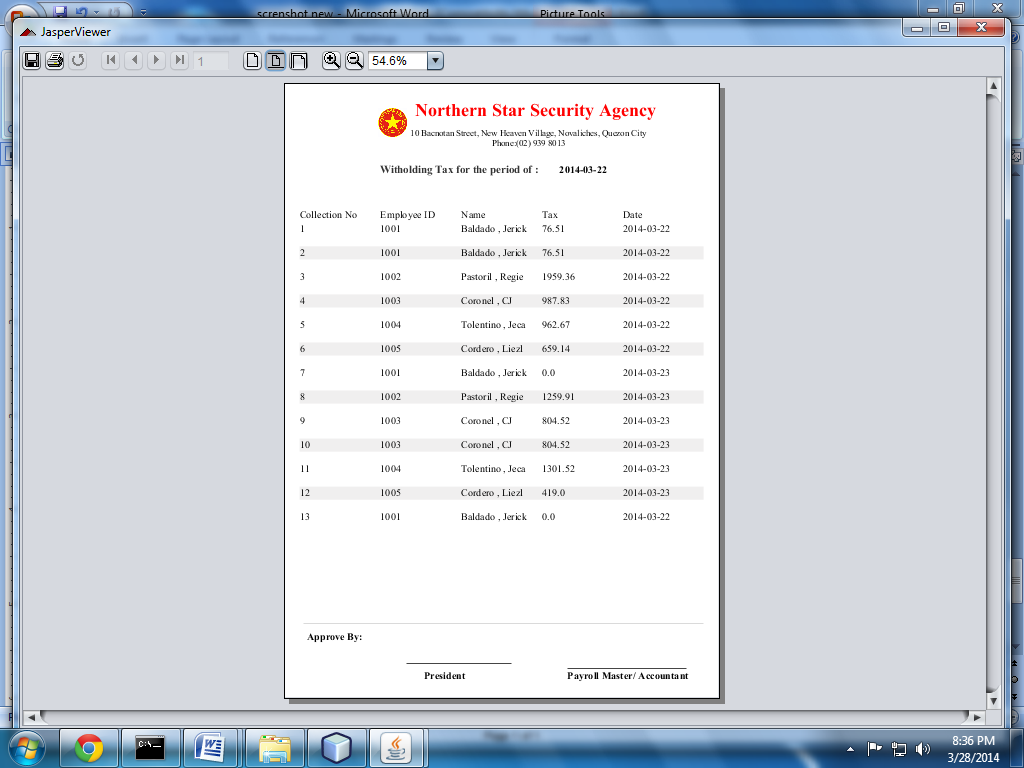
* + - 1. **Reports** 
         1. **Inventory Reports**

|  |  |  |
| --- | --- | --- |
| **Form** | **Fields** | **Description** |
| Payroll Records | Employee ID  Name  Payroll date from,  Payroll date to,  Pay date,  Number of day work,  Special holiday,  Legal holiday,  Total allowance  Bonus  13th mp,  Leave pay,  Total deduction,  Gross pay  net pay | This contain the detail of employees salary, including the gross pay, total deduction, and netpay |
| Withholding tax | Employee ID  Name  Tax  Date | This table calculates the equivalent amount of tax base in salary of the employee |
| Pag-ibig | Employee ID  Name  Employees contribution Employeer contribution  Total  Date | This table holds the information about the contribution of the employee and the share of the employer including the total share of the employee |
| Philhealth | Employee ID  Name  Employee share,  Employeer share  Total  date | This table holds the information about the contribution of each employee. The record of the monthly salary base |
| SSS | Employee ID  Name  Employees contribution Employeer contribution  Total  Date | This table holds the information about the contribution of each employee. The record of the monthly salary base |
| Bonus | Employee ID  Name  Bonus amount  Date pay | This table holds the information of the bonus of each employee including the date and also the amount that will be received of the particular employee. |
| Allowance | Employee ID  Name  Allowance Type  allowance amount  date paid | The purpose of this table is to keep the record of the employee for the there allowances the type and date that will be pay and also the amount. |

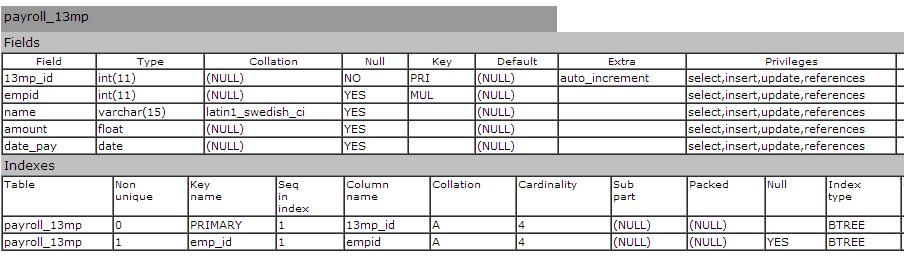
* + - * 1. **Layout Of Reports**

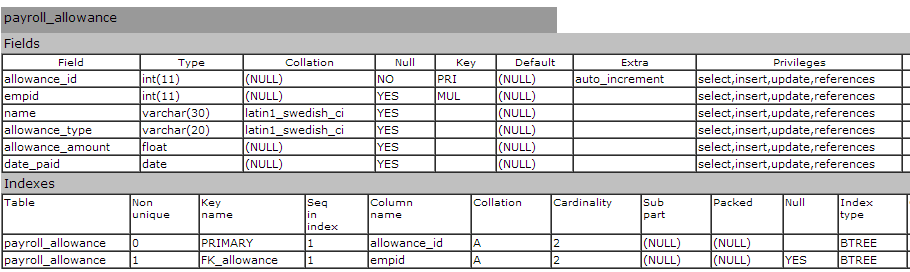
****

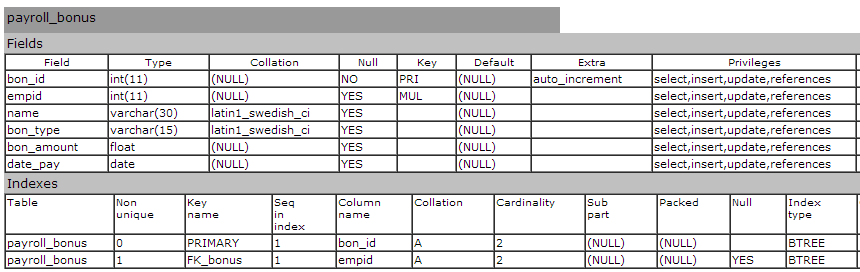
****

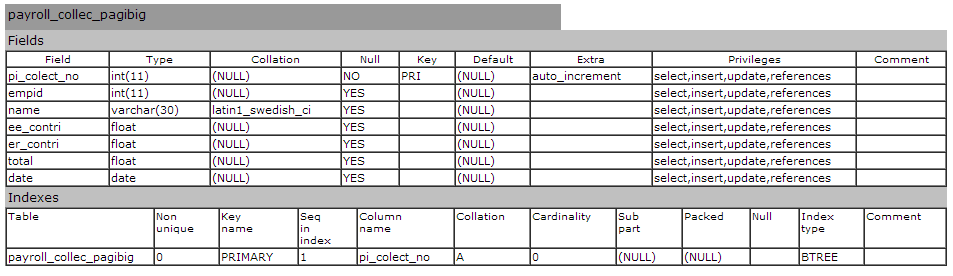
****

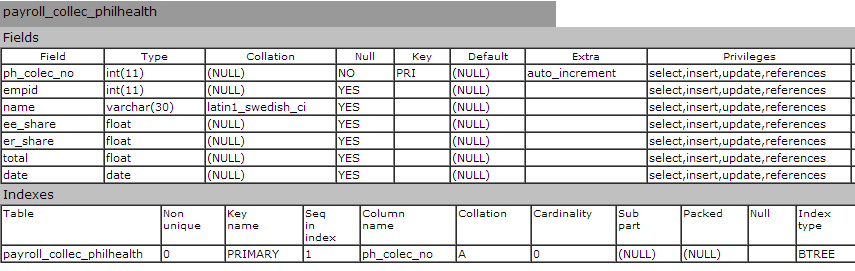
* + - * 1. **Data Dictionary Of Reports**

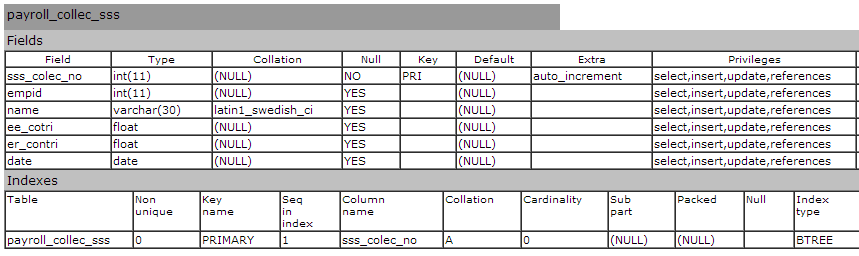
****

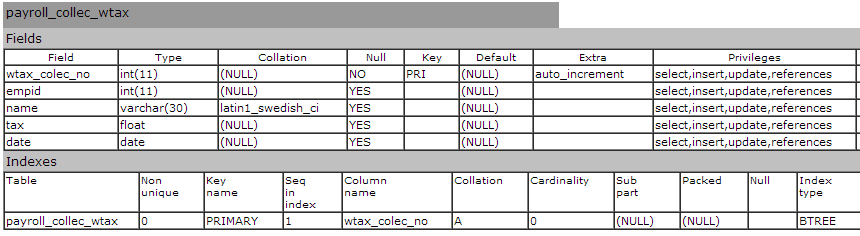
****

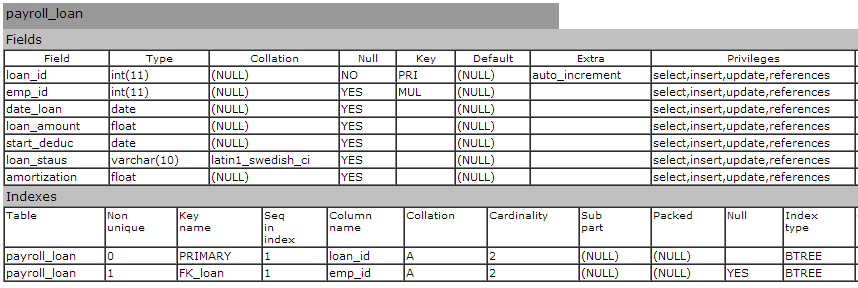
****

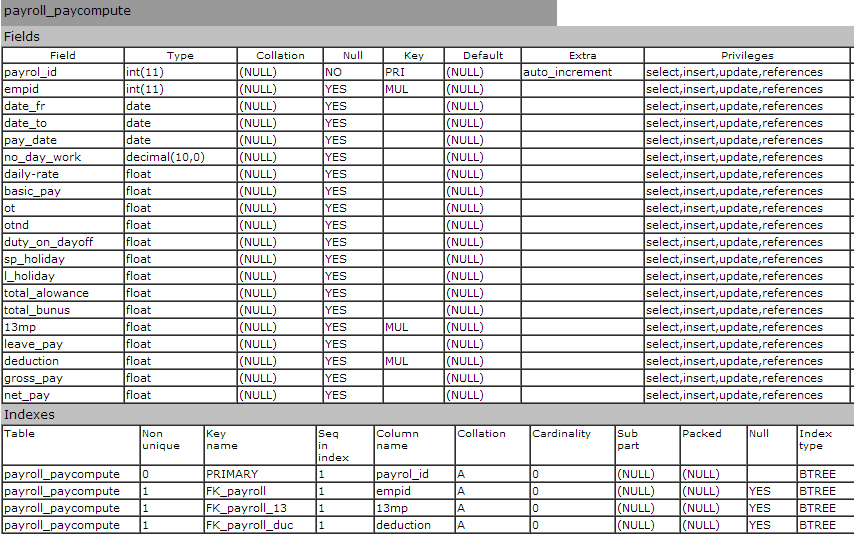
****

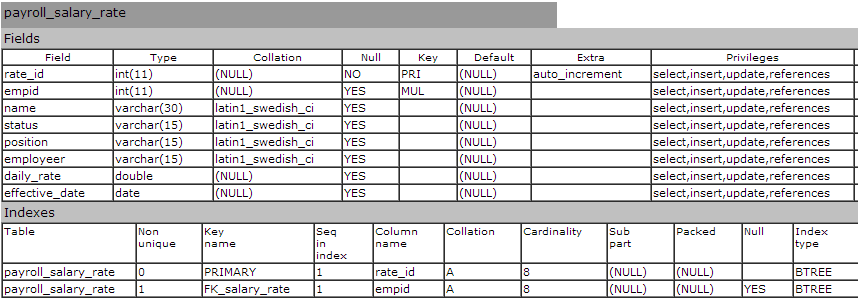
****

****

****

****

****

****

* + 1. **BEHAVIORAL MODEL DESCRIPTION**
       1. **Description For Software Behavior** 
          1. **Events**

**Payroll Login class event**

Choose user, admin or accountant

Login to system

Add employee

View records

Print reports and pay slips

Logout to system

**Payroll class events**

Payday period

Calculate employee’s salary

Add payment to database

Generate pay slip and report

**Admin class events**

Login to system  
monitor the system

Add records  
Edit records

View salary reports  
print pay slip

Logout to system  
  
**Accountant class events**Login to system  
generate the employee payroll  
save and view salary reports  
print pay slip  
Logout to system

**Salary rate class events**

Starting year

Employee salary is recorded

Daily rate is assign to an employee

Edit employee daily rate.

**Allowance class events**

Employee gets an allowance

Allowance amount has been change

**Bonus class events**

Employee gets a bonus

Date of bonus issued

**Deduction class events**

Deduction is added

Deducted from employees gross pay

Deductions remove.

**Loan class events**

Loan status of employee

* + - * 1. **States**

**Payroll Login Description**

Login admin and accountant has logon to system

Added admin add an employee to payroll records

Editing admin edit employee payroll records

Viewing admin view employee payroll records

Printing admin and accountant print reports and employees pay slips

Logout admin and accountant has log out to system

**Payroll state Description**

Payday period days in which an employee work is paid

Calculating employee’s salary is calculated such as gross pay, deduction and net pay

Added the calculated payment to of an employee is being saved.

Generating note given to an employee when they have been paid, detailing the amount of pay given .

**Employee’s state Description**

Working Employee has start working and has been added to payroll accounts

Login logon to system using their identification number

Viewing employee can be able to view his/her salary reports

Logout logout to system

Inactive in able the employee info

**Salary rate state Description**

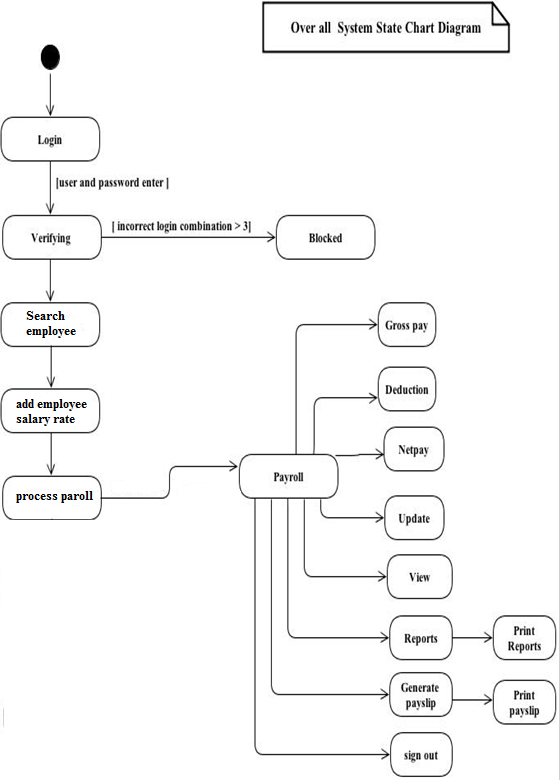
Starting the date an employee start to work

Recoding employee’s salary has been recorded

Assigning Daily rate is assign to an employee

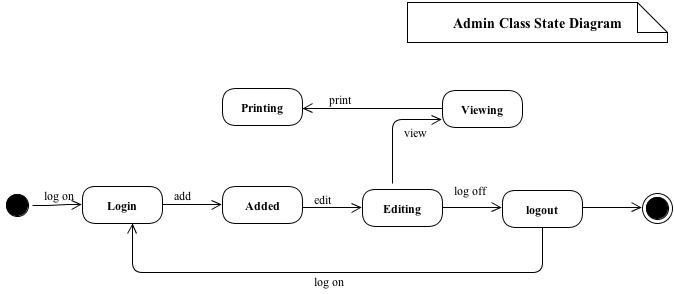
Editing employee daily rate can be change

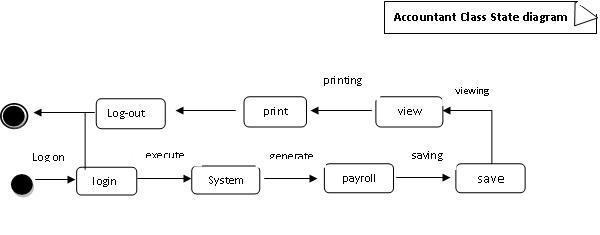
* + - 1. **State Charts Diagram**

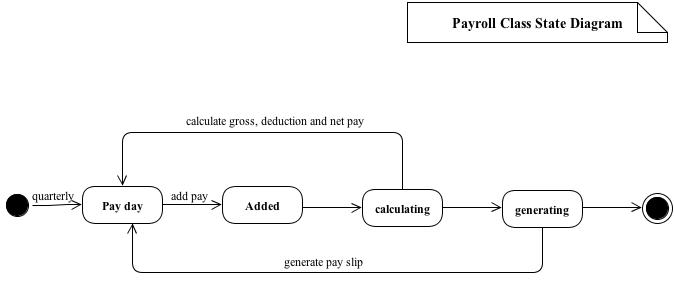
****

process

**Figure 3.10: State Diagram of Northern Star Security Agency Payroll System**

****



****

**Figure 3.11: Class State Diagram of Northern Star Security Agency Payroll System**

* + 1. **Restriction, Limitation, And Constraints**
* Only the payroll administrator can update the information.
* Accountant should only allowed for payroll process
* The software is using SQL, therefore the systems should be properly connected to the server to access database.
  + 1. **Validation Criteria**

To ensure the validation criteria of the software, some test are conducted if the system responds to according to their function and user meet its expectation. Therefore the user must be involved in some phases of test procedure.

* + - 1. **Classes Of Test**
* Every sub system of the software must undergo a unit testing, and these include following:

1. Log in
2. Reminders
3. Viewing and printing reports
4. Viewing and printing pay slips
5. Salary computation

* Black box testing which is also called the functional testing will be focus on the output of selected input of the software system. This type of testing is based entirely on the software requirements and specifications.
* Acceptance testing is basically done by the user or customer. Acceptance test represents some expected result from the system. Customers are responsible for verifying the correctness of the acceptance tests.
  + - 1. **Expected Software Response**
* The software must response to its appropriate function. If the user enters incorrect it must then it must display an error message.
* The software must not be capable of deleting an employee information, it must use either active or inactive.
  + - 1. **Performance Bound**
* The user cannot enter the system if he/she’s user and password wrong.
  1. **SOFTWARE DESIGN SPECIFICATION**
     1. **INTRODUCTION**
        1. **Goals And Objective**

Developing the software for the payroll processing is an attempt to make things simpler, easier, accurate and efficient. There are many goals for which the software is developed such as:

1. Efficient
2. Reduces complexity
3. Minimize cost
4. Timeliness
5. Preventing loss of data
6. Compete with the other payroll system

The software has a capability to store unlimited number of employees. It developed for the service management agencies payroll that can maintain a database of employee’s personal information, employee’s semi-monthly or monthly salaries and other payroll information. Its can perform operation such as auto computation of employees salary, generation of payslips, reports, updating records etc. It reduces complexity as there is no need of paper and pencil work. Furthermore performing calculation in efficiently and not taking more of the bookkeeper's time than necessary so it minimizes cost for the agency. It can track employee’s payroll information in an efficient way. It process payroll in a timeliness manner since all employee are entitle to be paid on designated payday.

The software is also integrated with different subsystem such as: assignment and monitoring, contract and management system, general ledger, and communication management.

* + - 1. **Statement of Scope**

The software is capable of storing employee’s payroll records. Before the software process the payroll, employee’s working day’s data are collected from the Assignment and monitoring system. Employees are identified using their unique identification number.

**Major inputs:**

Employee’s number of days work

Salary rate

Allowance

Bonus

Deduction

Overtime

Loan

Tax rate

Employees pay period

**Processing Functionalities**

|  |  |
| --- | --- |
| Function | Priority |
| Employees information entry | Essential |
| Viewing employees information | Essential |
| Update employees payroll records | Essential |
| Payroll computation | Essential |
| View reports | Essential |
| Print reports | Essential |
| Cash advance balance tracking | Essential |
| Leave with pay tracking | Essential |
| Generate pay slips | Essential |
| Payroll period alert | Essential |

**Outputs**

Employee Record reports, Employee Pay Summary Record or Pay Period, Employee Salary Distribution Summary, Vacation Negative Balances, Vacation Sick Leave Balances, Statutory reports, Summary reports, payroll reports, payslips and etc.

* + - 1. **Software Context**

Developing this payroll software will process payroll in a timeliness manner, it is efficient and reduce cost since it does not need lot of payroll master in computing employee’s salary. This software is develop using Java language, mySql for database and connector/J for the connection.

* + - 1. **Major Constraints**

The software is to run on a system, with SQL Therefore the systems should be properly connected to the server to access database. The Systems should be good enough to perform all the actions fast for example it is recommended that system should be no older than PII .The RAM should be not less than 128 MB.

* + 1. **DATA DESIGN**
       1. **Internal Data Structure**

The component JTextField of the GUI is use in adding and updating data in the database table. When add or update button is pressed the information in those text fields is directly sent to the database for add or update. For accessing or viewing any table the data is fetched directly from the DBMS to the java database object is displayed directly on the screen. For report printing temporary local variables have been defined in the procedures with meaningful names to store different values and then print the report. After finishing those variables are discarded.

* + - 1. **Global Data Structure**

The only kind of data available globally to the whole system is the database itself stored in sqlYog. After the connector/J connection, any data stored can be fetched, updated, or deleted. For more information regarding the database stored refer.

* + - 1. **Temporary Data Structure**

The Payroll Software application does not use any intermediate temporary files of its own. As data is being stored in a DBM system all the intermediate data it being stored by the Database until the command “commit” is being executed in the SQL query by the application. Hence no intermediate files or data structures are used.

* + - 1. **Database Description**

**Table Name:** Payroll\_13th month pay

**Attributes:** 13thmp\_ID, emp\_ID, name, amount, date pay  
  
 **Description:** This table holds the record of the 13 month pay of each employee. The date that will be released the 13 month pay of the employee. The Primary key of this table is the Payslip\_ID.

**Table Name:** Payroll\_Allowance

**Attributes:** allowance\_ID, emp\_ID, name, allowance\_type , allowance amount , date\_paid

**Description:** The purpose of this table is to keep the record of the employee for the there allowances the type and date that will be pay and also the amount.Primary of this table is Allowance\_ID.

**Table Name:** Payroll\_Bonus

**Attributes:** bon\_id, empid, name, bonus\_type, bonus\_amount, date\_pay

**Description:** This table holds the information of the bonus of each employee including the date type and also the amount that will be receive of the particular employee. Bon\_ID is the Primary key of this table.

**Table Name:** payroll\_collec\_pag-ibig

**Attributes:** pi\_collect\_no, empid, name, ee\_contributes, er\_contributes, total, date

**Description:** This table holds the information about the contribution of the employee and the share of the employer including the total share of the employee

**Table Name:** payroll\_collec\_philhealth

**Attribute:** ph\_colec\_no, empid, name, ee\_share, er\_share, total, date

**Description:** This table holds the information about the contribution of each employee. The record of the monthly salary base

**Table Name:** payroll\_collec\_sss

**Attributes:** sss\_colec\_no, empid, name, ee\_contri, er\_contri, date

**Description:** This table holds the information about the contribution of each employee. The record of the monthly salary base

**Table Name:** payroll\_collec\_wtax

**Attribute:** wtax\_collec\_no, empid, name, tax, date

**Description:** this table calculates the equivalent amount of tax base in salary of the employee

**Table Name:** payroll\_loan

**Attributes:** loan\_id, emp\_id, date\_loan, loan\_amount, start\_Deduc, loan\_staus, amortization

**Description:**. This table holds the information about the debt. of each employee. The record of the monthly salary base

**Table Name:** payroll\_paycompute

**Attributes:** payroll\_id, empid, date\_fr, date\_to, pay\_date, no\_day\_work, sp\_holiday, l\_holiday, total\_allowance, total\_bonus, 13th mp, leave\_pay, deduction, gross\_pay, net\_pay

**Description:** this table generates of all employee salary including 13th month pay, bonus gross pay and net pay

**Table Name:** payroll\_salary\_Rate

**Attributes:** rate\_id, empid, name, status, position, employeer, daily\_rate, effective\_Date

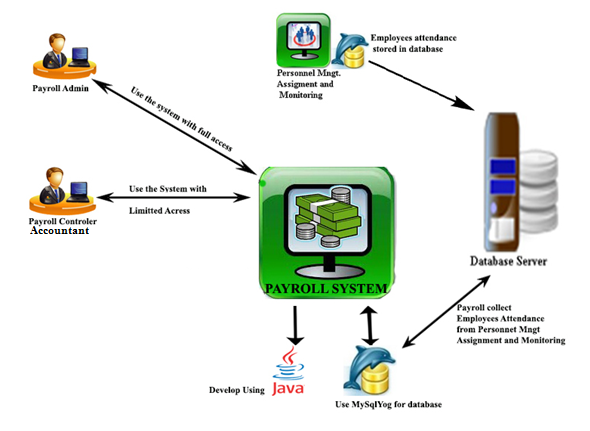
**Description:** this table holds the record the salary rate of employee including of each position

**3.2.1 Architectural and Component Level Design**

**3.2.1.1 Architectural Diagram**

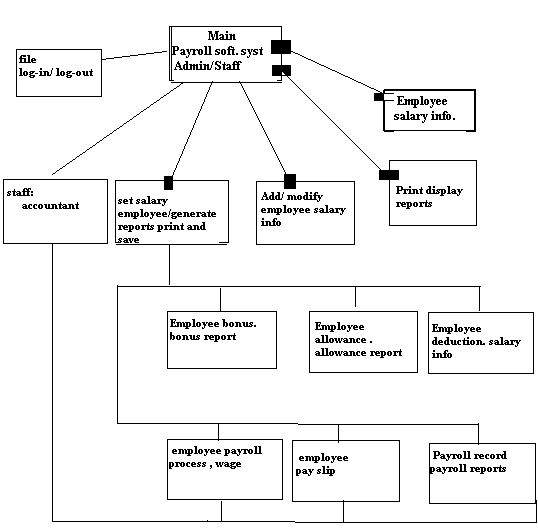
The architectural context diagram for the software is shown below. The payroll system is developed using Java Language for the front end and mySqlYog for the database. Before processing the payroll, employees attendance are collected from the Personnel Mngt. Assignment and Monitoring database.

The software has a various users that are allowed to access the system, such as Payroll admin, payroll controller and employees. The admin manage the payroll system given a full access to the system. Payroll controller can use the software but only limited access, it only allows to add, view but not to delete payroll data. Employees can access the system only to view their payslips and other payroll records.



**Figure 3.12 Architectural Diagram**

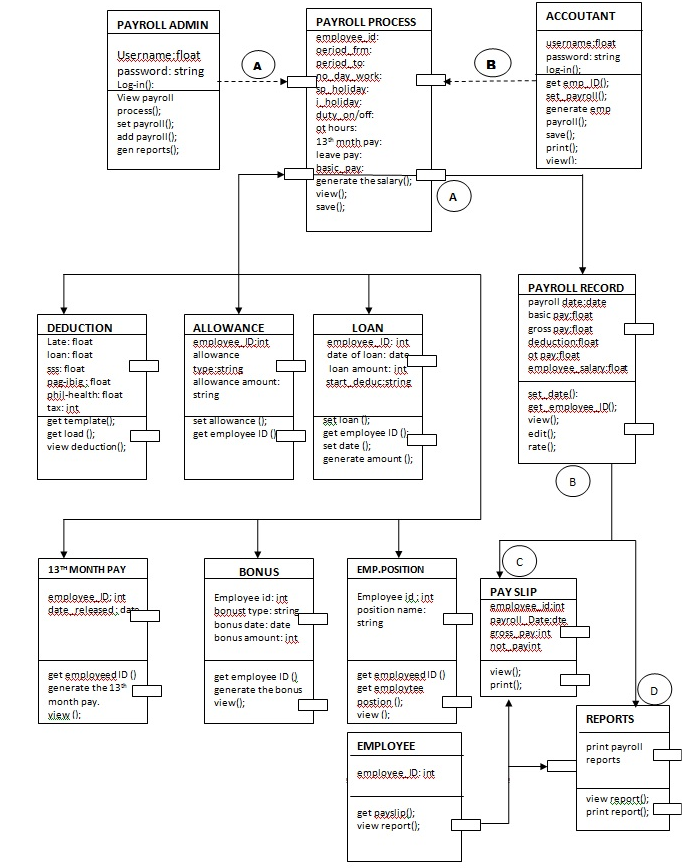
There are various types of architectures to represent the context. Basically the architectures can be divided as system architectures and application architectures. The system architecture described the architecture context diagram. Application architecture represent application architecture such as call and return, pipe and filter, object oriented architecture etc. the system case calls and return architecture and object oriented architecture are shown. However the main emphasis is on object oriented architecture. First call and return architecture is drawn. There is a main which calls the controller function and this function further call the application functions. These are functions return and required for data. The call un return architecture for the service management payroll system software is shown below (using UML component diagram):



**Figure 3.13 Architectural Context Diagram**

The groups going to describe the object-oriented architecture, since this architecture is derived from analysis payroll admin. First is refers to analysis payroll admin in requirement specification. From those classes, the design classes are made and component level designing id done. Now each class has a component. And each some interface, which described in the component level designing, can see in architecture drawn on next page.

The architecture describe the details about the classes, the group’s going to use the variables and their types. The types of relationship is also shown for e.g payroll admin access the payroll system , account access only payroll process , payslip , view and print reports of the salary of employee through payroll system. Our architecture diagram gives more details than the call and return architecture. the only difference on call and return architecture and object oriented is that call and return architecture considers represent object-oriented view when using object oriented programming, the object-oriented architecture is follow on next page:



**Figure 3.14 Object oriented architecture**

**3.2.1.1.1 Description and Components**

**3.2.1.1.1.1 Component Payroll Admin**

**3.2.1.1.1.2 Processing narrative of component Payroll Admin**

The payroll admin the person who responsible for all tax activities , including compliance with local, state, and federal regulations and filing local, state, and federal payroll tax returns. May supervise 1 or more subordinates in the calculation and reporting of tax liabilities, in responding to questions and special requests from employees and regulatory agencies, in researching laws and regulations regarding taxes and payroll withholding procedures, and in preparing budgets, projections, and special reports for senior management

**3.2.1.1.1.3 Interface description of the component Payroll Admin**

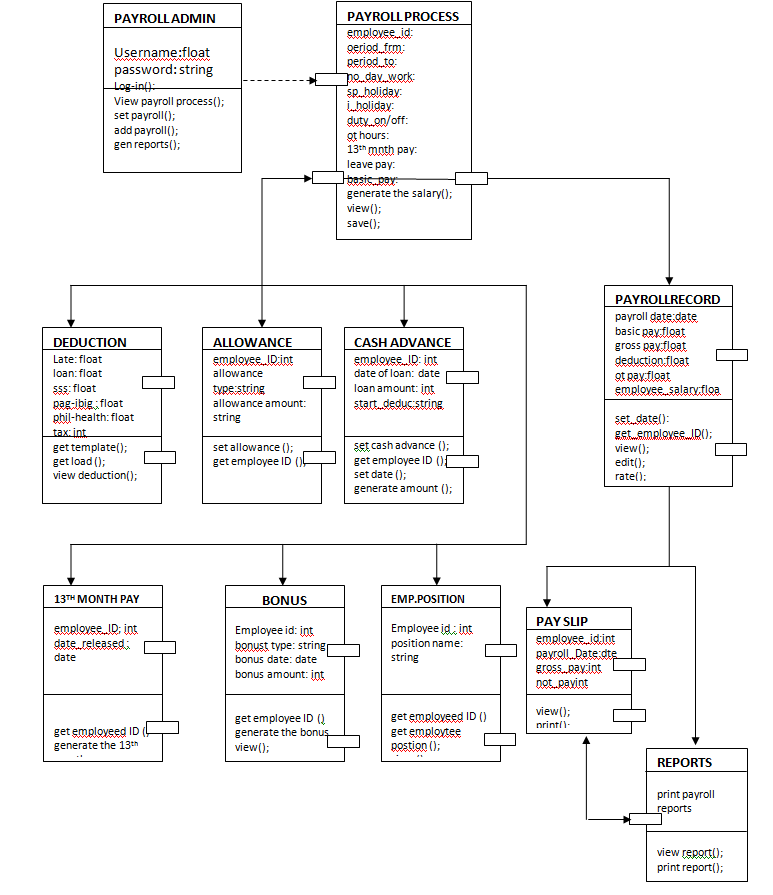
A log-in screen will display for the payroll administrator . He/she can log onto the system which contain Username and Password .after logging on , the main menu for the payroll admin will display and all data and records of employee has in there through searching names.

**3.2.1.1.1.4 Algorithmic description of Payroll Admin**

The Payroll Admin log onto the system the payroll admin select view payroll information.The system requests the employees’ payroll information from the server.

The system looks for computed payroll of the employee. Payroll admin can select the report he wants to view. The system requests the report from the server. The payroll admin is given the option either close or print the report. He/she checks the report. Payroll admin can print reports that allow you to verify the payroll before you print paychecks and generate the direct deposit file. He/she can also make adjustments, if necessary.

**3.2.1.1.1.5 Design class hierarchy of Payroll Admin**

****

**3.2.1.1.2.1 Component Accountant**

**3.2.1.1.2.1 Processing narrative of component accountant**

The component of accountant contains the class accountant. It contains the attributes which are processing payroll, payroll records, reports and payslip. Accountant generates the salary or amount of the employee , accountant also responsible for view and print the reports and payslip of the employee.

**3.2.1.1.2.3 Interface description of the component accountant**

The component of accountant has an interface with log in system. It has indicated username and password. Accountant can only authorized the system if he/she has an account of

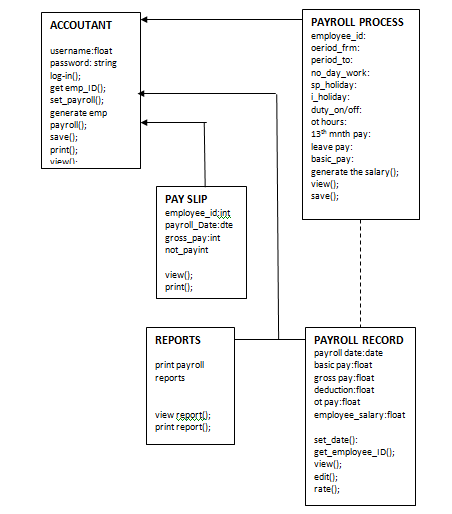
the system. The attributes of accountant is only for wage of the employee, sending reports to the admin and giving payslip to the employee.

**3.2.1.1.3.3 Algorithmic description of component accountant**

Start();

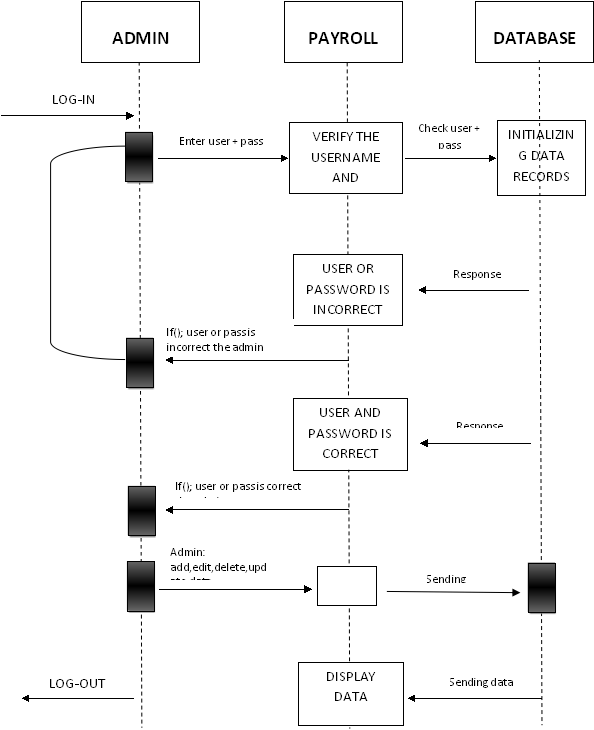
Accountant log-in with his username and password  
generating the wage of the employee  
saving record to database  
print payslip  
end();

**3.2.1.1.2.4 Design Class Hierarchy**

****

**3.2.1.1 Dynamic Behavior for Component**

**3.2.1.1.1 Interaction Diagram**



Sequence diagram where the use case Component of the payroll admin the first event is login. The admin need to enter its user name and password. The payroll system will verify the user and password through database if user or password is not correct the admin cannot access the system he/she will re-entered or sign-up again. If the user or pass is correct the admin shall allow to add, edit and update. the data and records .that will be display on payroll system. And employee record.

**ACCOUTANT**

**PAYROLL SYSTEM**

**DATABASE**

LOG IN

**VERIFY USER +PASSWORD**

Enter user+pass

Execute query

Initiali-zing data record

**USERNAME AND PASSWORD IS NOT CORRECT**

Response

If () user name or password isn’t correct , accountant can’t enter to the system

**USERNAME/PASSWORD CORRECT**

If() user name or password are both correct, the accountant can now access the payroll system

Accountant () generate employee wage, report, print and payslip

Print payslip

LOG OUT

PRINTING

START

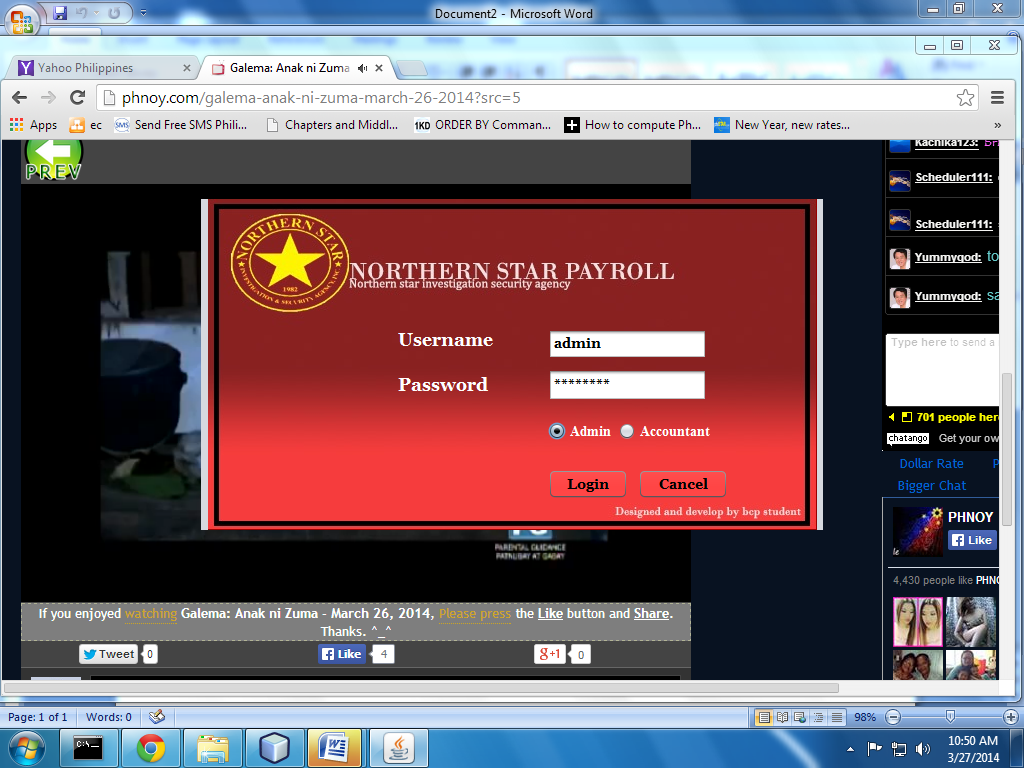
3x attempt

Sequence diagram where the use case component of the accountant , First event is login, accountant need to enter the username and password, system will verify the input data and send it to database if the username or password is in correct, accountant can’t enter to payroll system. Accountant can only attempt at least 3x of login, if accountant username and password correct. Accountant can now access the payroll system, the scope of accountant is only, generating the salary of employee including save payroll records , view and reports of payslip.

* + 1. **User Interface Design**
       1. **Description of the User Interface** 
          1. **Screen Images**

The GUI for the payroll system is very user friendly and does not require any special training just to learn how to use the software. Below are images and description of the GUI.

The first image is the login screen. The user should have a user name and a password and choose a designation if admin or accountant in order to login.

  
After entering username and password you must click the login button and the program will verifies the information entered. Upon verification the program will allow you to login or display a message that username and password are incorrect.

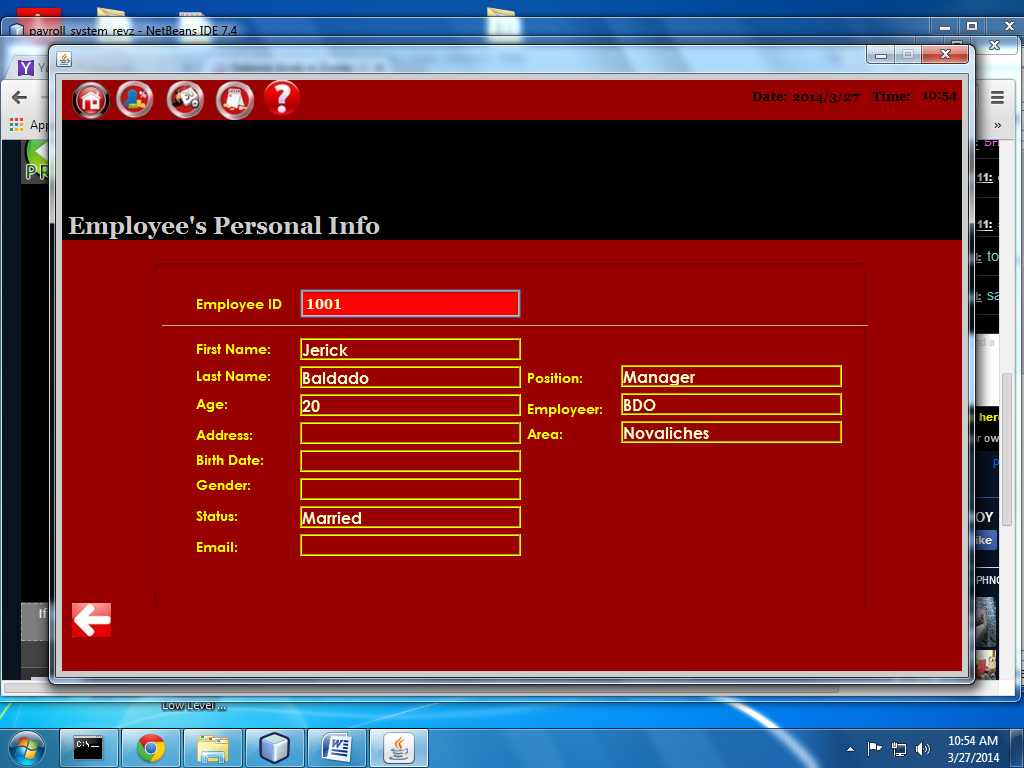
The program will also check the whether the person login is an admin, payroll controller or employee. Each of them has a different level of access.

The next image is the home page interface. After the login is granted the Admin will be directed to the home page.



At the home page the admin can choose different option such as employee record, Employee salary rate , payroll compute and reports.

When employees record is click the user will be directed to the employees information form. As shown in the image below.



The image below is the employee’s record. Records are displayed through a table. There are also different categories to search an employee record.

Image below is the employee wage. The admin can add, edit , search and print employees record.



Next is the payroll processing page. The payroll administrator can compute employee’s salary in this page and save it on the database. The interface is easy to navigate.



At the home page when the reports button is click different reports are such as payroll records, statutory deductions, allowance, 13 month pay and bonus report. It shows in image below.





If the user choose the statutory deduction the image below will show.



* + - 1. **Interface Design Rules**

To improve the usability of an application it is important to have a well designed interface. These “Eight Golden Rules of Interface Design" are a guide to good interaction design.  
  
 1**) Offer informative feedback.**

For every operator action, there should be some system feedback. For frequent and minor actions, the response can be modest, while for infrequent and major actions, the response should be more substantial.

2**) Design dialog to yield closure.**

Sequences of actions should be organized into groups with a beginning, middle, and end. The informative feedback at the completion of a group of actions gives the operators the satisfaction of accomplishment, a sense of relief, the signal to drop contingency plans and options from their minds, and an indication that the way is clear to prepare for the next group of actions.

3) **Strive for consistency.**

Consistent sequences of actions should be required in similar situations; identical terminology should be used in prompts, menus, and help screens; and consistent commands should be employed throughout.

4**) Enable frequent users to use shortcuts.**

As the frequency of use increases, so do the user's desires to reduce the number of interactions and to increase the pace of interaction. Abbreviations function keys, hidden commands, and macro facilities are very helpful to an expert user.  
  
 5**) Permit easy reversal of actions.**

This feature relieves anxiety, since the user knows that errors can be undone; it thus encourages exploration of unfamiliar options. The units of reversibility may be a single action, a data entry, or a complete group of actions.

6**) Offer simple error handling.**

As much as possible, design the system so the user cannot make a serious error. If an error is made, the system should be able to detect the error and offer simple, comprehensible mechanisms for handling the error.

**7) Reduce short-term memory load.**

The limitation of human information processing in short-term memory requires that displays be kept simple, multiple page displays be consolidated, window-motion frequency be reduced, and sufficient training time be allotted for codes, mnemonics, and sequences of actions.

**8.)Support internal locus of control.**

Experienced operators strongly desire the sense that they are in charge of the system and that the system responds to their actions. Design the system to make users the initiators of actions rather than the responders.

* + - 1. **Components Available**

Java swing library have many components available but for now only JFrame, JTextLabel, JTextField, and JButton are being used in the GUI. Among them only JButton have been added to the mouse action listener. Rests have no action listener attached to them. More components may be added in future if necessary.

* + - 1. **UIDS Description**

No user interface development system has been used in the development of the childcare system.

* + 1. **Restrictions, Limitation, and Constraints**

a.Only the manager can update or delete the information

b. Every employee should have emp\_ID to access the information.

c. The functions and data members of the component are called only when the user requires to manipulate the data of the component

d. The software is to run on a system, with SQL Therefore the systems should be properly connected to the server to access database.

e. The Software should be designed in such a way that the processing speed of the operations should be high

* + 1. **Testing Issues**

Test strategy and preliminary test case specification are presented in this section. The various tests to be conducted to validate the software developed are login test, Child information report testing, Parent information report test, Employee information report test, employee pay check test, child billing report test.  
  
 There are two types of testing techniques. Black Box and White (Clear Box).  
 In Black Box we consider the domain of inputs to the system and without considering details in the software. In white box, we consider each and every path, branch for the testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Classes of testing** | **Description** | **Expected  software response** | **Performance bounds** | **Components to be identified** |
| Log-in | This test class is defined to validate whether the user is able to login with his ID if admin user/pass | Accountant /admin able to be log-in w/ his, if admin username/pass | The user should not enter the system if username/pass is not correct | records |
| Allowance | This test class is defined to validate the allowances of the employee | Test will view/print the allowance |  | Employee,reports |
| Bonus | This test class is defined to validate the Bonus of the employee | Test will view/print the Bonus |  | Employee,reports |
| 13th month pay | This test class is defined to validate the 13th monthly pay of each employee base on there word of days | Test will view/print  the 13th month pay |  | Employee,reports |
| Loan | This test class is defined to deduction, loan of employee. | Test will view/print  the loan |  | Employee,reports |
| Salary rate | This test class is defined to validate , the salary rate of the employee per day. | Test will edit the salary rate |  | Employee,reports,bonus,allowance,loan,bonus etc. |
| Process payroll | This test class is defined to generate the salary of the employee including net pay and gross pay | Test will save view and print the salary |  | Employee records |

**CHAPTER 4**

**4.0 TESTING**

**Introduction**

In developing a payroll system for service management, errors can be encountered at any stage during development. There are different techniques for detecting and eliminating errors that originate in that phase. However, no technique is perfect, and it is expected that some of the errors of the earlier phases will finally manifest themselves in the code. The code developed during the coding activity is likely to have some requirement errors and design errors, in addition to errors introduced during the coding activity. Behavior can be observed, testing is the phase where the errors remaining from all the previous phases must be detected. Hence, testing performs a very critical role for quality assurance and for ensuring the reliability of system. During testing, the program to be tested is executed with a set of test cases, and the output of the program for the test cases is evaluated to determine if the program is performing as expected. Due to its approach, dynamic testing can only ascertain the presence of errors in the program; the exact nature of the errors is not usually decided by testing. Testing forms the first step in determining the errors in a program. Clearly, the success of testing in revealing errors in programs depends critically on the test cases.

Generally, parts of the program are tested before testing the entire program. The reason for testing parts separately is that if a test case detects an error in a large program and also to determine or to identify what are those sub system connected to our system, it will be extremely difficult to pinpoint the source of the error. That is, if a whole system does not work, determining which module has errors can be difficult. Hence, it is clear that for a large system, we should first test different parts of the system independently, before testing the entire system.

## *Integration*

Integration testing in the software testing model comes before system testing and after the unit testing has been done.

The way that integration testing works is by, getting the individual modules that have been through the unit testing phase and integrating each module into a group, the integration testing phase will make sure when the modules are being integrated together that any problems, for example errors or bugs, caused due to the integration of the modules are eliminated. Integration testing does not deal with the integration of the whole system but deals with the integration of a process in the system.

In the integration testing stage there are three things that are created, to ensure that the integration of the modules is successful and that it runs successfully as well, a test plan, test cases and test data is produced to effectively test that the integration is successful. Test data is normally used by test cases but I have mentioned each type below;

**Integration test plan**

When producing a test plan it must include the following information to be effective,

* A strategy to use when testing the integrated modules and how the tests will be conducted.
* To know what are the responsibility of each integrated in our system
* Get data from other integration
* Responsibilities e.g (personnel management etc)
* Testing and pass condition
* Risk involved
* Data base record

Above are just some of the most important information that is in the test plan but as a test plan is simply planning there is more information and points that can be included in the test plan by a company. Most test plans are approved and worked on with the other sub system

### Integration Test Data

Test data is simply data that is used in order to test the actual program or the integrated modules. Test data would normally be used in a test case as this would be used to check the inputs and expected outputs.

I have included an example of what test data is in the test case example that was shown in the test case section above.

### Different types of Integration Testing The group need to get data of employee through other integrated in our system. Below is a list of the different integrations testing types:

**Data base testing**

This is where the system get the data base information from the other integrated system. It was implemented by employee information. Since our system is about payroll, we need to integrate from the other system. And to get there employee info through client info database, contract management database, employee info and assignment monitoring database.

**Client info Data base testing**

The client info need to get the number of employees inside of agency and basic info about employees including employee id, last name, first name etc. That where going to implement in database table. After the client info make a database table from there system. The personnel management recruitment will be integrated on client info to get employees list and basic info.

**Personnel management recruitment data base testing**

The personnel management recruitment database was include list of employees and basic info, personnel management recruitment will be integrated on employee information and assignment monitoring to gave the list of employee and basic info from personnel management recruitment

**Employee information and assignment monitoring data base testing**

The employee info And assignment monitoring database was included list of employees basic info, position, dtr, time in, time out, list of day work, etc that will be included of their database table. That the payroll system will be integrated from their and to get all database table info

**Different types of integrated system and scenarios information**

Upon developing the payroll system for service management agency the group also conduct a research for those sub systems connected in our work.

**Client info and contract management**

Specification scenario for client info and contract management

The system need to conduct scenarios comes from the client and agency just to confirm The agreement of each individual, which part on payroll transaction

Scenario of contract

After the client get info from the company. The agreement of each individual will agree the employees will be signed a contract in agency that will dispatched to the client  
**Test scenario for client info and contract management**

The client of service management agency is need to get an information of their agency. First client need to prepare a term of references and get calls/writes/advertise that possible on services providers. Then the service management agency will response and give write letter of client intent which attached in portfolio, after the agency submitted. The agency evaluates their service providers, then the client will chose, Client will select service providers, after to select the client the management agency will prepare a contract from the client. The agency will submitted the contract to client, the client will evaluate the contract, if the client not agree with the contract of agency. client will provide the list of amendments, the client will past it on agency, after agency verify the amendments of the client. The agency will edit the contract by wants of clients, then the agency evaluate it again, if the client agree with the contract of agency, the agency will get approve comes from the client, then the agency will execute the contract.

**Personnel management recruitment**

Specification scenario for personnel management recruitment

The system need to conduct scenario from the personnel management recruitment. To identify the contracts and personnel needs, list of applicants, list of qualified personnel per client

**Scenario for contracts and personnel needs**

The system need to identify what are the agreement between the client and the agency. Including of their terms and condition about on giving salary

**Scenario for list of applicants**

The system need to get all data records of applicant from personnel management recruitment. To identify what kind of position their

**Scenario for list of qualified personnel per client**

The system need to get all qualified personnel per client in personnel management recruitment. To identify the name and position of each qualified personnel

**Test Scenario for personnel management recruitment**

The agency will identifies the need for applicant that based on contracts. After identify agency evaluates existing and available roster of personnel. After the evaluation the agency will advertise job vacancies/seeks referrals. If the applicant is apply. The agency will conduct of some interview and tests. If applicant is pas in the interviews and test, the agency prepares the list of qualified personnel. And send the list of qualified to the client. The client will review all resume and may conduct further testing. After the testing , client will identify the personnel and submit list to dispatched firm. After the client submit, the list of personnel the agency will receives the list. Then give to the employee. If the employee is less than the required personnel the employee said NO employee will need to stop. If employee said YES the employee will undergo to the agency.

**Employee info and assignment monitoring**  
  
Specification scenario for personnel management - employee info and assignment monitoring

The system need to conduct scenario from the personnel management – employee info and assignment monitoring. To identify list of client and assigned employees, list of employees and contract dates, list of employees with legal cases, list of unassigned employees with active contracts list of employees with multiple assignments.

**Scenario to identify the list of client and assigned employees**

The personnel management need to identify all list of client and assigned employee. Comes from agency records

**Scenario for list of employees and contract dates**

The personnel management will get the list of employees and contract dates to the record of agency and clients including their name position and contract dates  
  
**Scenario for list of employees with legal cases**

The employee should have verification about on their legal cases

Scenario for list of unassigned employees with active contracts list of employees with multiple assignments

The personnel management must need to identify those employees who unassigned by their dispatched area with active contracts the personnel management also need to identify the employees which have multiple assignments and give all data to payroll management

**Test scenario for personnel management**

The employee will undergo to the agency. To submit the review and sign contract, then the agency will sent to the client. If the employee does not want the contract the client will do another request for employee. After make another request the client give a new request to the agency. The agency will reassign employee to the client, if the agency accept the employee. The client will send to the agency. And the agency will maintain the details of assignment. If the employee said YES the commits a gross misconduct and give to case ducket. If the employee said NO contract is end.

The agency should identify the assignments and prepare of employee contract and give to the employee. To reviews and sign the contract.

**General ledger**

**Specification scenario for general ledger**

The system need to conduct scenario from the general ledger to identify the incomes statement, balance sheet, cash flow statement and trial balances

**Scenario for incomes statement**

General ledger is need to calculate all incomes of the agency and give each statement

**Test scenario for general ledger**  
  
GL process flow is a five step process from recording the transactions in the system to finally running the reports containing financial data out of the system. The input for GL Process Flow is the raw accounting data and the output is the accounting reports that can be used to provide various levels of financial information.  
  
Step1. Create Journal or Import Journal from Sub‐Ledger

Step2: Review Journals

Step3: Approve Journals

Step4: Journals Posting

Step5: Run Financial Reports

General ledger is the final form of transactions, general ledger generate all reports of the agency which also include payroll reports.

**CHAPTER 5**

**EVALUATION**

**5.1 FINDINGS**

Login scenario, when enter the username and password then click the login button. The user and password that has been entered should match the data record in database to gain access to the system. After the message box displayed that access granted the system home page will be displayed.

Employee salary wage scenario, this will be able to add, edit, delete and view records of employee salary. The first stage is to add the employee’s salary rate. In order to add employee salary the admin needs to enter employees employee number, first name, last name, salary type, salary amount, then click the save button. Employee salary data should be recorded at the database upon save button click. When saving record successful a message box will show notifying that the record has been successfully added. The next stage is to view the employee’s salary record. The record is displayed in a table and when the data at the table is click the information will be populated at the textbox. Next stage is to edit the employee salary records. The admin can edit the existing record and when the update button is click the record in the database will be updated. If update successful the message box will display successful. Next stage is about deletion of record, the user needs to select the record that will be deleted, and click the delete a button

The next employees allowance scenario, at the home page, employees tab an allowance button should be click to show the allowance page. This will be add, update, adjust the information about the allowance of the one employee.

Bonus scenario it defined bonus type of the employee and get the bonus to exact date and amount of bonus .This scenario will be able to add, update and view records of each employee.

13th month pay scenario each employee will get 13th month pay not less than 1/12 of the basic salary of the employee. The amount of 13th month pay will release before 24th December.

The next is loan scenario will able to add, edit and view the record of the employee to know the loan status of each employee. admin can edit the existing record and add the information about the employee.

Payroll schedule scenario, the user will set the date of the payroll schedule in order to compute salary before the day of payoff.

The next is salary rate scenario of employee this will be able to add and update about the salary rate of employee. This updating the rate of each employee if their something changes to the wages of the employee

Payroll process scenario, the admin will compute the wages or the salary of an employee. Base on the days of work ,different deduction, contribution that will be need to be successful of this transaction. There’s a lots of the information will need to do this scenario .this scenario the admin can navigate the save and compute of the salary of each employee.

**5.2 Analysis**

Purpose of the research is to analyze, the design, and recommended the payroll of accounting information system that will support the internal control to solve the problem. Research methods used are book studies, field studies, and design studies. Fields studies done by survey, research and interview. The expected result are to give a review about the payroll of accounting information system in the ongoing business process of company and to solve all the weakness in the payroll system, so the company can use the integrated information system in counting of the payroll. There are two major object

**PAYROLL ADMIN** The payroll admin the person who responsible for all tax activities , including compliance with local, state, and federal regulations and filing local, state, and federal payroll tax returns. May supervise 1 or more subordinates in the calculation and reporting of tax liabilities, in responding to questions and special requests from employees and regulatory agencies, in researching laws and regulations regarding taxes and payroll withholding procedures, and in preparing budgets, projections, and special reports for senior management

The Payroll Admin log onto the system

The payroll admin select view payroll information

The system requests the employees’ payroll information from the server

The system looks for computed payroll of the employee

Payroll admin can select the report he wants to view

The system requests the report from the server.

The payroll admin is given the option either close or print the report. He/she checks the report

Payroll admin can print reports that allow you to verify the payroll before you print paychecks and generate the direct deposit file. He/she can also make adjustments, if

necessary.

**EMPLOYEE** refers to all people working in the organization.

Employee works part time or under time under a contract of employment .whether it’s oral or written.

When the employee is log on onto the system, he/she can do different task and options.

The employee can selects Account Settings under utility menu. The system will requests the record of the signed in employee from the server and will display the account information. When the account information is display on the screen, the employee can edit the proper fields. The employee can selects Save or Cancel. When the employee selects the Save button, the system will send the updated record to the server for storage. When the employee selects the cancel button, he/she will go back to the main menu. The employee’s ID number, the date, and the nature of the change are logged The employee receives confirmation that the information was saved this two objects generate payroll system.

**5.3 Lesson Learned**

The Payroll System generates the sum of all financial records, salary for employee, wage, bonuses and deductions. payroll refers to the amount paid to employees for services they provided during a certain period of time. Payroll plays a mar ably affect the net income of most companies and they are subject to laws and regulations. We need to conduct a lot of studies and research on how to make a payroll system.

**Payroll system** is not easy than you think because making a payroll system is too complicated. It consist a lot of calculations (e.g financial records of employee including wage bonuses deduction and taxes)  
  
the group conduct a lot of research just to fulfill all wants of the system, moreover we learned that how the system works in the business and how is it important to the company.

**5.4 Recommendations**

The group trying to make the system different again the other system, but we didn’t achieve to apply other features in cause of lack of knowledge about coding. The group trying to make a user history from log-in to log-out including username, date and hour, the purpose of this new feature is to make the system more secured specially our system holds payroll system, talking about the salary of the employee. We made this note just to recommend all up-coming 4th year students here at beslink college of the Philippines who will be assigning also on payroll system. To continue our work, to be come much better on your project study.

**CHAPTER 6**

**CONCLUSION**

As the conclusion, the Payroll System can be used by small and medium scale of industries. It is useful for any kind of firm or industries and also those people who want to make a business someday

To make a Payroll System make sure that you will have an administrator because the administrator is important actor. His the one who responsible to observe the happening of the system, administrator also responsible for innovation of the system

Therefore by develop the system all of this problem will be and the administrator no need to calculate the salary as manually because the all salary will automatic calculate by the system. Moreover the payroll system also has many advance features for user. for example ,user can search, save , edit and update the details and print out the pay slip and reports. this is the advantage of using the payroll system, it can decrease the burden and all the transaction will be more systematic.

**REFERENCE**

Essencial Of Payroll System Management and Accounting. Steven M. Bragg . Publish Jhon And Wiley & Sons Inc. 2003, Hoboken New Jersey. Page 10

Future Gov Asia. By Adrienne Valdez , 2 October 2011

Future Gov Asia. by [Clarice Africa](http://www.futuregov.asia/users/34/) | 31 July 2013

[http://en.wikipedia.org/wiki/Payroll. Retrieve August 11](http://en.wikipedia.org/wiki/Payroll.%20%20Retrieve%20August%2011), 2013

[http://bestpayrollsystem.blogspot.com/2011/01/payroll-system-and-its-importance-in.html. Retrieve August 10](http://bestpayrollsystem.blogspot.com/2011/01/payroll-system-and-its-importance-in.html.%20Retrieve%20August%2010), 2013

<http://ezinearticles.com/?What-Is-a-Payroll-System?&id=5681334>. Retrieve Aug. 10, 2013

<https://www.quickpayroll.ph/Home/About>. Retrieve Aug. 10, 2013

[www.sunstar.com.ph](http://www.sunstar.com.ph) . Retrieve Aug. 10, 2013

www2.doh.gov.ph. Retrieve Aug. 10, 2013

<http://gracepioquidgara.blogspot.com/2009/10/chapter-2.html>. Retrieve Aug. 10, 2013

<http://www.hostitwise.com/software_design/nenios_care_center_srs_usage_scenario.html>

<http://www.1keydata.com/sql/sqlorderby.html>

<https://support.payrollhero.com/hc/en-us/articles/200845508-How-to-compute-Philippine-BIR-taxes->

http://www.philstar.com/business/2014/01/07/1275831/new-year-new-rates

**APPENDIX A- CURRICULUM VITAE**

**BALDADO, JERIC V.**  
Blk 37 Lot 10 Lily St. Monica Park Nova Q.C  
Contact No: 09489928749  
E-Add: [jeric.pogi@yahoo.com](https://www.facebook.com/jerik.neutron?fref=ts)

**POSITION:     Project Manager**  
**PERSONAL DATA**

Gender: Male

Age: 20 years old

Date of Birth: July 12 1993

Place of Birth: General Hospital

Civil Status: Single

Religion: Roman Catholic

Skills: Could operate effectively with Microsoft office stools, namely words, excel, power point and access. Can program different programming languages such as, visual basic, PHP, and Java.

**EDUCATIONAL BACKGROUND:**

**Tertiary:** Bestlink College of the Philippines

Quirino Hi-way, Novaliches Quezon City

        Bachelor of Science in Information Technology

2010-2014

**Secondary:** Maligaya High School

Quezon City   
(2007-2010)

**Elementary:** Maligay Elem. School  
Quezon City   
(2000-2006)

**CORDERO LIEZL S.**

Deparo Caloocan City

Contact No: 09129437151

E-Add : [cyma0413@yahoo.com](https://www.facebook.com/kish.mea?fref=ts)

**POSITION: Document Specialist**

**PERSONAL DATA:**

Gender: Female

Age: 22 Years old

Date of Place: Naval, Biliran

Civil Status: Single

Religion: Catholic

Skills: Trouble Shooting, Could operate effectively with Microsoft office stools, namely words, excel, power point and access. Can program different programming languages such as, visual basic, PHP, and Java.

**EDUCATIONAL BACKGROUND:**

**Tertiary:**  Bestlink College of the Philippines

Novaliches Quezon City

Bachelor of Science in Information Technology

2010-2014

**Secondary:**  Naval School Of Fisheries

Caraycaray Naval, Biliran

2003-2007

**Elementary**  San Pablo Elementary School

San Pablo Naval, Biliran

1997-2003

**CORONEL CHRISTIAN JAMES D.**

23 B Coronel Camp. Bag-Bag Nova Q.C

Contact No :  09484751268  
E-Add: dave.cornel01@yahoo.com

**POSITION: Lead Programmer**

**PERSONAL DATA:**

Gender: Male

Age: 19

Date of Birth: May 1 1994

Place of Birth: Q.C

Civil Status: Single

Religion: Roman Catholic

Skills: Technician, Could operate effectively with Microsoft office stools, namely words, excel, power point and access. Can program different programming languages such as, visual basic, PHP, and Java.

**EDUCATIONAL BACKGROUND:**

**Tertiary:**  Bestlink College of the Philippines

Novaliches Quezon City

Bachelor of Science in Information Technology

2010-2014

        **Secondary:** Tandang Sora High School  
                                              Tandang Sora Q.C  
                                              2006-2010

        **Elementary:** Goodwill Elem. School  
                                              Bag-bag Q, C  
                                              2000-2006

**PASTORIL REGIE S**  
A5-Sitio Cabuyao Sauyo Road Nova Q, C  
Contact No: 09475252727  
E-Add: [regiepastoril101@yahoo.com](https://www.facebook.com/pastorilregie)

**POSITION: Business Analyst**

**PERSONAL DATA**

Gender: Male  
Age: 19 years old

Date of Birth: May 15 1994

Place of Birth: Palanas City

Civil Status: Single

Religion: Roman Catholic

Skills: Technician, Could operate effectively with Microsoft office stools, namely words, excel, power point and access. Can program different programming languages such as , visual basic , PHP, Java.

**EDUCATIONAL BACKGROUND**

**Tertiary:** Bestlink College of the Philippines

Novaliches Quezon City

Bachelor of Science in Information Technology

2010-2014

**Secondary:** Sauyo High School  
Nia Village Q.C   
2006-2010

**Elementary:**  Sauyo Elem. School  
Nia Village Q.C   
2000-2006

**TOLENTINO JESSICA L.**   
Add: Blk #1 Carreon Village Nova Q.C  
Contact No: 09264159040  
E-Add: [jessicatolentino10@yahoo.com](https://www.facebook.com/jecca.tolentino.7?fref=ts)

**POSITION: System Analyst**

**PERSONAL DATA:**

Gender: Female

Age: 20 years old

Date of Place: Feb 08 1993

Civil Status: Single

Religion: Roman Catholic

Skills: Could operate effectively with Microsoft office stools, namely words, excel, power point and access. Can program different programming languages such as, visual basic, PHP, and Java.

**EDUCATIONAL BACKGROUND:**

**Tertiary:** Bestlink College of the Philippines

Novaliches Quezon City

Bachelor of Science in Information Technology

2010 – 2014

**Secondary:**  Sanbartolome High School

Sinforosa St. Quirino Hi-Way

(2006-2010)

**Elementary:**  Sanbartolome Elem. School

Sanbartolome Nova Q.C   
(2000-2006)

**APPENDIX B – PROJECT IDENTIFICATION FORM**

**PROJECT IDENTIFICATION FORM (F1)**

Name of Group/Individual: 1. Baldado, Jeric .

2. Cordero, Liezl S.

3. Coronel, Christian James

4. Pastoril, Regie S.

5. Tolentino, Jessica L.

BSIT Registration No.: BSIT 4204

Index No.: 08A

Email Address: 1. [jeric.pogi@yahoo.com](mailto:jeric.pogi@yahoo.com)

2. cyma0413@yahoo.com

3. [christianjamescoronel@yahoo.com](mailto:christianjamescoronel@yahoo.com)

4. [regiepastoril@yahoo.com](mailto:regiepastoril@yahoo.com)

5. jecatolentino@ymail.com

Title of the Project: Service Management Agency (Payroll System)

Name of Advisers: Ms. Eunice Balisi

**PROJECT DESCRIPTION FOR (F2)**

Index No.: BSIT4104-08A

Title of the Project: Service Management Agency (Payroll System)

Web based project or not? : No

Name of advisers: Ms. Eunice Balisi

1. **Statement of Objectives**

(a)Goals of objectives of your proposed project with measurable outcomes

## Generate Accurate Reports.

A payroll system must be able to keep and generate accurate reports on employee income. Accuracy is an important objective of a payroll system, because employees should be paid fairly for the work they have done, and a company's financial well depends on not overpaying employees for the time they have worked.

* **Generate Paychecks and Pay Stubs on time.**

Generate paychecks and pay stubs on time is an important objective of payroll system. Employees should be paid on designated paydays. Failure to pay employees on time can lead to problems.

* **To provide user friendly and modernized system.**

The payroll system is user friendly in the various activities such as record updating, maintenance, and searching.

* **To improve time efficiency.**

User can retrieve and view the records of the employee’s detail by the use of search engine from the system and generate the various reports. This will make the transaction faster.

(b.) Itemized list of deliverables of the system with respect to the functionality of the system (not project study, CD, etc.) – user and management reports:

* User Documentation
* Installation Guide
* Standalone Payroll
* Pay slip
* Performance Reports
* Salary Reports

(c.) Scope

The System is designed for the Payroll System for Company. It includes the features that can Add Employees record, Edit Employees information and Delete Employees record, print / save. Conduct generation of reports and Pay Slip of each employee casual or regular. Updating Employees information as well as the Weekly Salary, Monthly Salaries, Cash advance, the rate per day, overtime, night differential, late , absent, pay during day off, special and legal holiday pay, Gross payment, Net pay, and Deduction such as withholding tax and SSS contribution, Phil Health contributions, Pag-Ibig contributions.

(d.) Limitation

The payroll System does not covers the bank deposit of employee salary , time-in and time-out of employees, deployment of agencies, promotion of employee, termination of employee, salary deduction of applicant , percentage of agency that came from applicant. It also does not support online program or online transaction.

(e.) Reasons for choosing the proposed project?

The reason why the group choose the propose system because, payroll plays an important role in a company or an organization. It deals with money and relationship between the employee and employer. Sooner or later the each one of the group will be able to work, it either become an employee or be employer. So at this early stage the by developing the system the group would be able to understand and know the process of the payroll.

(f.) Your strategy for getting started:

* Conduct a research about the payroll system.
* Create an interview questioner.
* Find a company and conduct interview to get information on their existing payroll system and their process.

1. **Required Development and Testing Environment**
2. Hardware Requirements – (processor, RAM, storage capacity etc.), Software requirements – (OS, language versions, libraries etc.), Special Requirements (if any)

* Hardware requirements
  + Processor - Pentium 300 MHz or faster
  + RAM - At least 128 megabytes (MB) or higher
  + Disk Drives - At least 2 gigabytes (GB) of available space on the hard disk or higher
  + Video Display – VGA or better video display.
  + Keyboard / mouse
* Software requirements
  + Windows xp / 7
  + Java language
  + SQLyog

1. How do you intend to obtain access for these facilities? Indicate the location, frequency of access to use these facilities and its approximate durations on monthly/weekly/daily basis.

Our system is accessible only for accounting and administrator office. We assured that our system can take several years through help of maintenance and weekly check-up

1. How do you intend to demonstrate your project work? Indicate access to software, hardware, data required for the demonstration.

In order for us to demonstrate our project work we need the following equipment:

* Hardware: Desktop ,Laptop ,LAN Connection, Router,
* Software: Java Language, Sqlyog , Internet

1. Will the final system be tested/evaluated by real users:

We conduct a brief survey to different employees of the company who are  
 knowledgeable or unknowledgeable about the system. We let them use the system for testing. This will test whether the system is a “user-friendly” and easy to use. The user will also evaluate if the system performance, cost, reliability and compatibility of the system.

1. State the way of generating test data/test cases:

We generate test cases to insure if the system is functioning well and if there are no errors or bugs. The user interface, the function and the performance should be tested to know if it their designated function respond correctly.

1. State the way of determining the acceptability/success of final system:

To determine the success of the system we need to have a post implementation maintenance and review to remove bugs and other error in our developed software before implementing.

The hardware and software to be used by the programmer for developing the system should meet the expected result in the development of a project, which would satisfy all the basic needs for which the project has been created.

1. The procedure to evaluate the performance of the system:

The procedure to evaluate the performance of our system is we need first to investigate the specific physical and performance characteristics for each hardware component to be acquired. Both hardware and software must consider these factors of performance, cost, availability, reliability, compatibility, technology, and of course the connectivity in local area network another important evaluation factor. It must also Efficient, flexible in handling operation, if it provides security control procedures for errors, malfunctions and improper use, does software document include helpful user instructions

After we checked the hardware and software, we’re going to demonstrate our system to the personnel in-charge or person who only authorized in our system like accountant and administrator office.

1. **The Work Plan**

Use a project management tool such as Microsoft Project. Include each schedule in a separate sheet. A schedule should fit into a single A4 page.

1. The project schedule (using a gantt chart):
2. The production schedule for the project study (using a gantt chart)
3. **Additional Information** (only for clarifications)

**Group/Individual Signature Date: / /**

**APPENDIX D – ADVISER CONSENT FORM (/F3)**

Name of Group/Individual: Baldado, Jeric .

Cordero, Liezl S.

Coronel, Christian James

Pastoril, Regie S.

Tolentino, Jessica L.

BSIT Registration No.: BSIT – 4104 Index No.: 4104 – 08A

Title of Project: Service Management System ( Payroll System )

|  |  |  |  |
| --- | --- | --- | --- |
|  | Adviser 1 (IT Related) | Adviser 2 (optional-client) | |
| Name | Ms. Eunice Balisi |  | |
| Designation | IT Intructor |  | |
| Work Places | BCP |  | |
| Academic/Professional  Qualifications/  Membership | BS Computer Science |  | |
| Email Address | nicebalisi@yahoo.com |  | |
| Telephone | 09088100869 |  | |
| **Adviser’s Signature** | I hereby confirm that I have undertaken to supervise the project mentioned above and I do certify that I am not a member of the Project Examination Board (PEC) of BESTLINK | | |
| ...................................................  DATE: | | ………………………………….  DATE: |

**LOG FORM (F4)**

No: 08-A

Date: …………………………………… Time: From: ……………………

To: …………………………

Work carried out/achievements since the last meeting with the adviser:

Work in arrears:

Problems encountered:

Work to be carried out/major decisions taken

Next meeting: Date: …………………….. From: …………………… To: …………

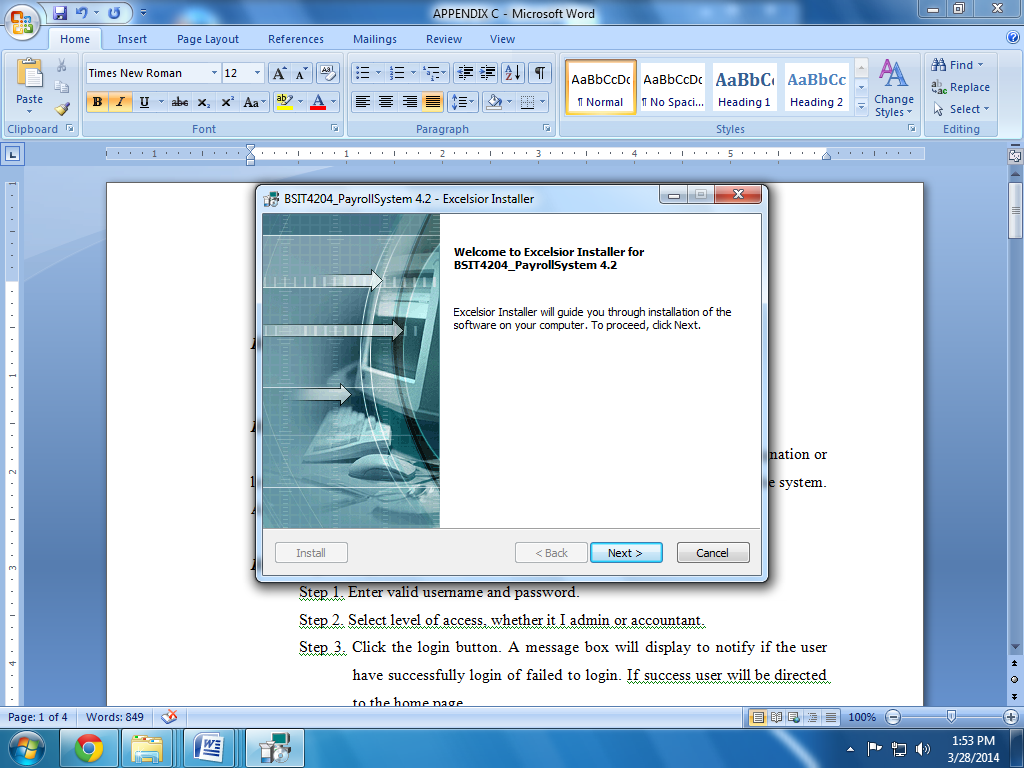
…………………………………… ………………………………………

Signature of group/individual Signature of adviser

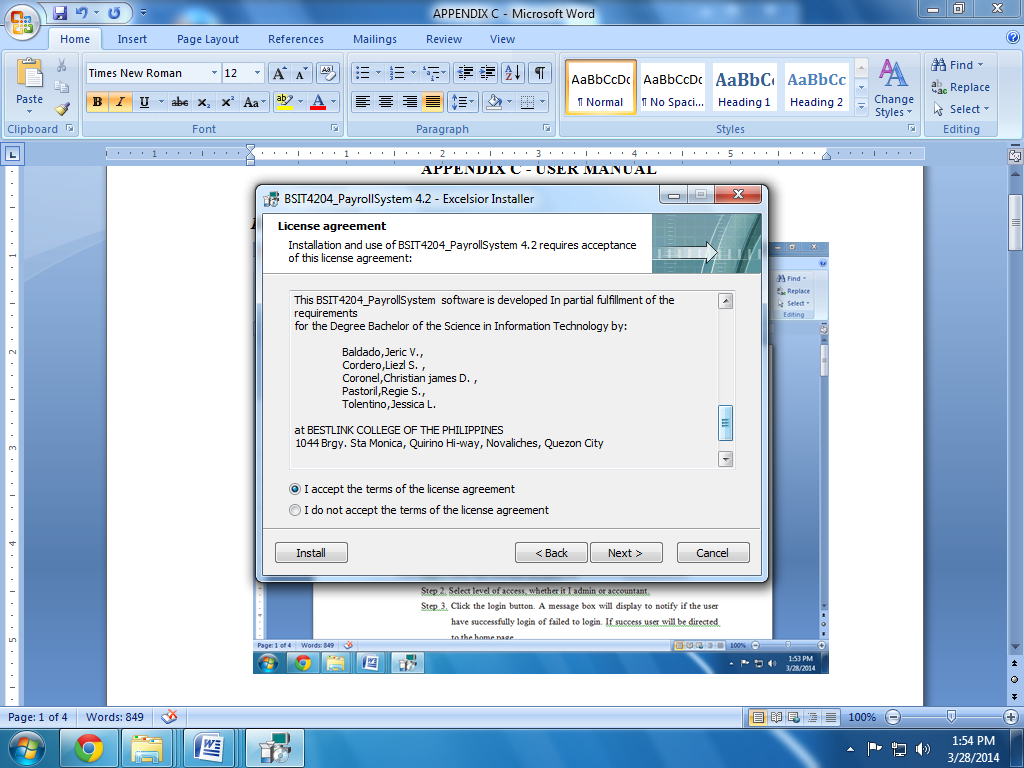
**APPENDIX C - USER MANUAL**

***How to install the system?***

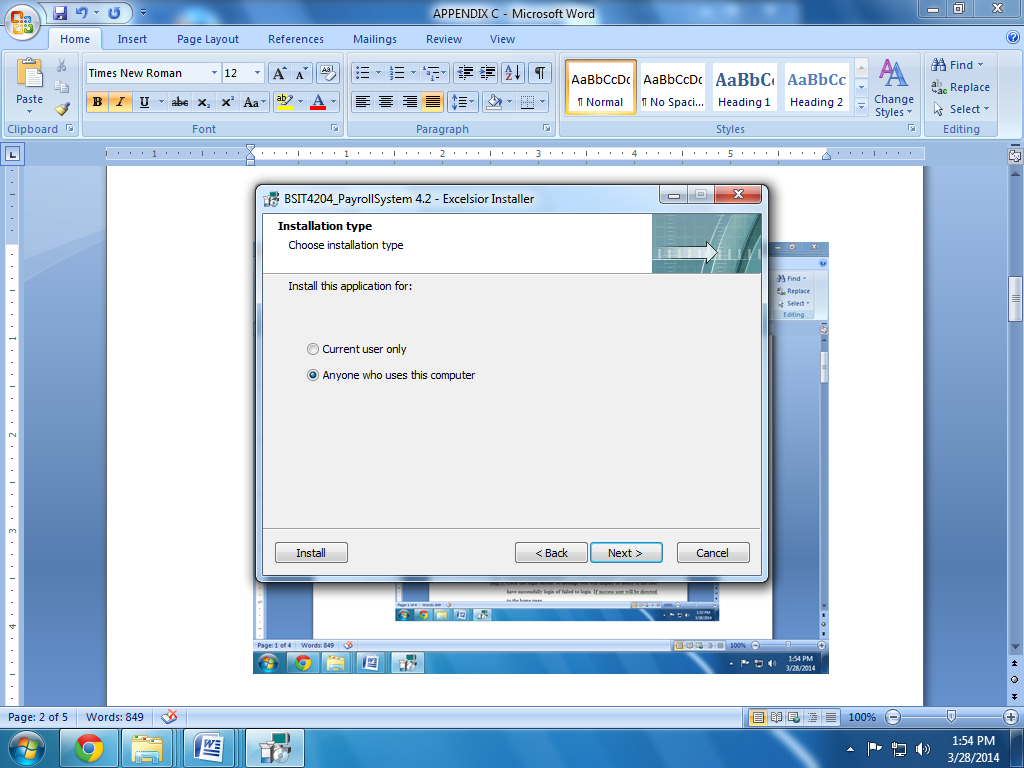
Just click the BSIT4204\_PayrollSystem-setup. Read the instructions and click next button

******

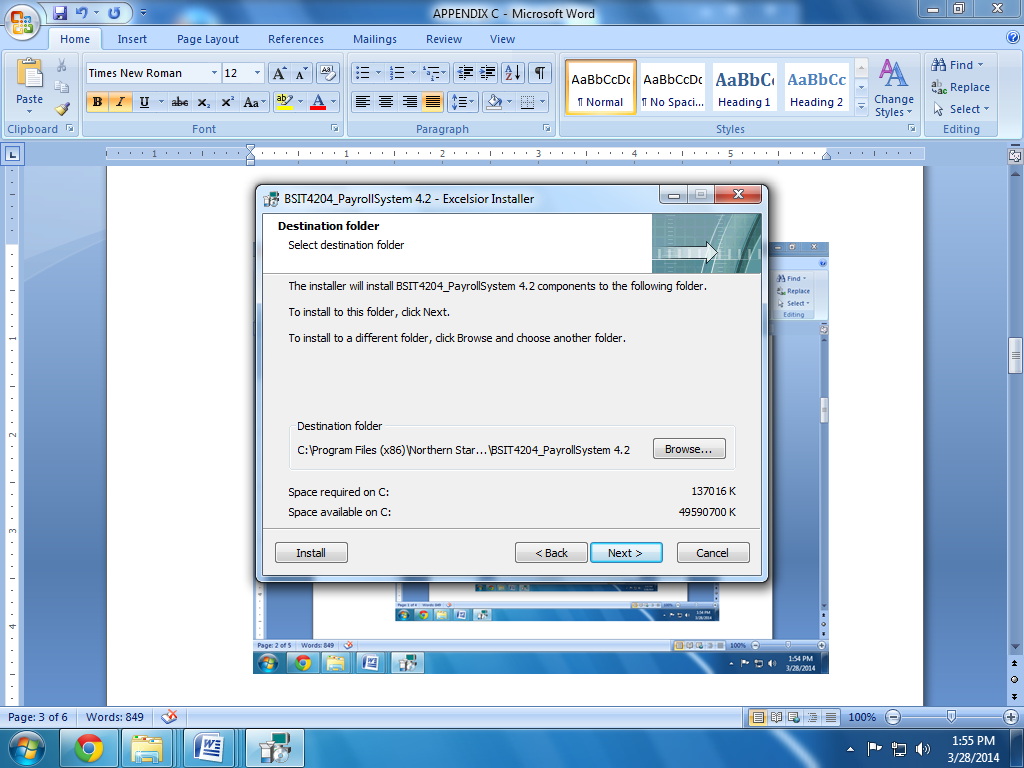
Accept License agreement, and then click next

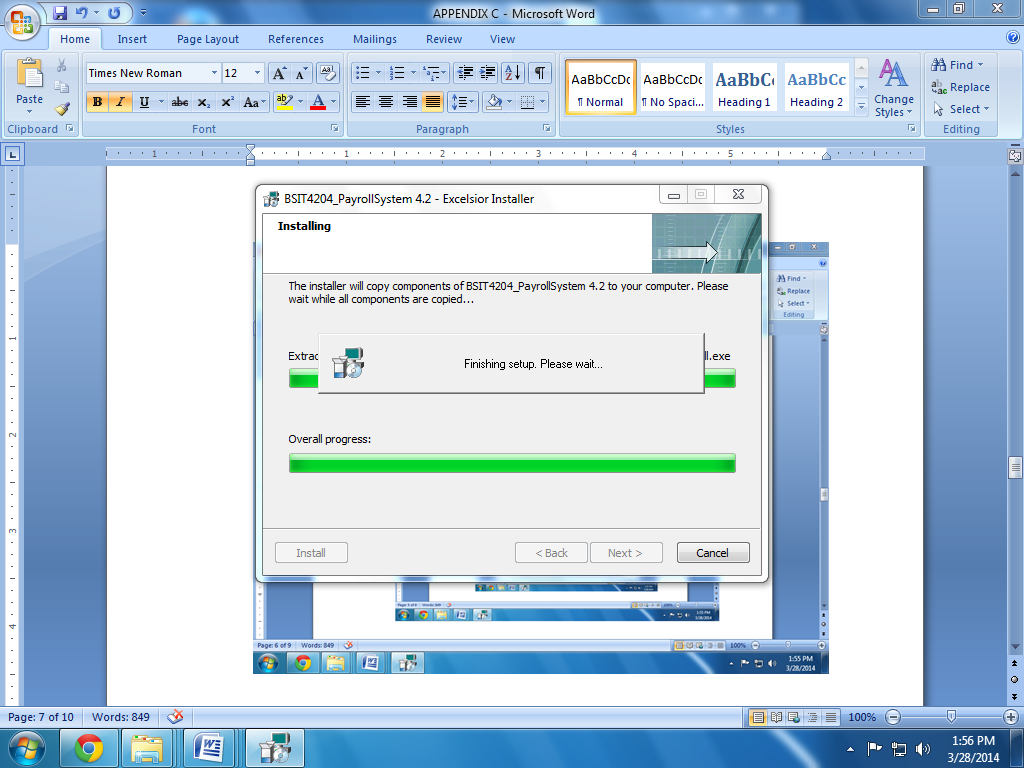
******

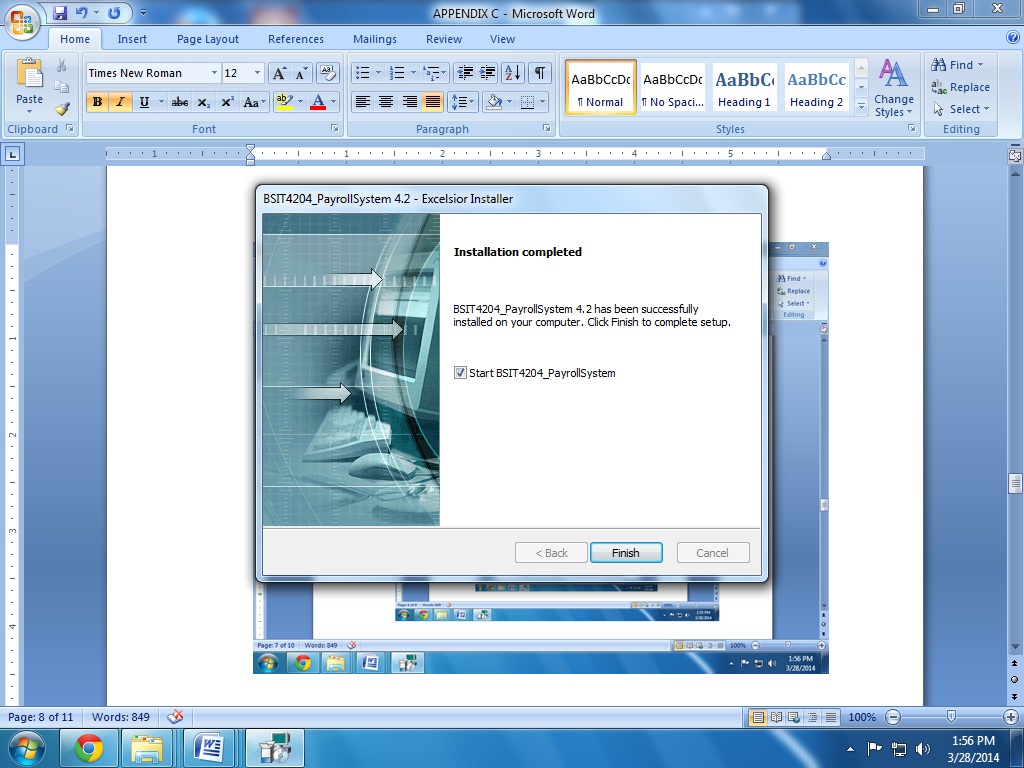
Choose installation type then click next

******

You can modify the installation folder by clicking browse, else just click next, next until it start the installation and click finish if it’s done.

******

******

******

***How to use the system?***

User must have a valid username and password, depending on their designation or level of access. The admin is the only one that has a full control or access to the system. Accountant has a limited access to the system.

***How to login the system?***

Step 1. Enter valid username and password.

Step 2. Select level of access, whether it I admin or accountant.

Step 3. Click the login button. A message box will display to notify if the user have successfully login of failed to login. If success user will be directed to the home page.

*Note:* three consecutive login failed system will shut down.

***How to add employee salary rate?***

Step 1. Login to system.

Step 2. At the home page select EMPLOYEE’S SALARY RATE button. User will be directed to the employee’s salary rate window.

Step 3. Click the add button to enable the text field.

Step 4. Enter employees ID to retrieve the employee’s information such as status, position, employer, and area.

Step 5. If information has been retrieve, determine if employee has dependents. If yes then choose one of the option given, else do nothing.

Step 6. Enter the amount of employee’s salary rate per day.

Step 7. Click SAVE button to save the data. A successful message will displayed.

***How to update employee’s salary rate?***

Step 1. At the salary rate window. Click the EDIT button.

Step 2. Search for existing employees record by name or by ID.

Step 3. Click employee’s information on the table below. As soon as the employees information has been retrieve to the text filed.

Step 4. Make changes to the employee’s daily rate amount or even the employee’s dependents if necessary.

Step 5. Click the UPDATE button. A successful message will display.

***How to compute employee’s salary***

Step 1. At the home window, choose payroll compute button. The user will be directed at the payroll computation window.

Step 2. Select payroll period day, whether it is the 15th or 30th day of month. At the 15th day there are no statutory deductions to be deducted. It will only be deducted on 30th day of month.

Step 3. Search for employees by ID. If employee’s record has been retrieve.

Step 4. Click the compute button. Then review if computed data are correct.

Step 5. If correct, click the save button. If not, click reprocess.

**Payroll Computations**

|  |  |
| --- | --- |
| **Descriptions** | **Computations** |
| Total No. of regular days worked | Total No. of regular days worked \* rate = Basic Pay |
| Regular Overtime (hours) | (Rate/8) \* 1.25 (constant) = OT Pay |
| Overtime with Night Differential | (Rate/8) \* 1. 35 (constant) = OTWND Pay |
| Duty on Day off | Rate \* 1. 3 = DOD Pay |
| Legal Holidays (If emp work)  (work with overtime)  (if emp did not work)  (work on rest day)  (workonrest day w/ OT) | Daily rate + COLA) x 200%  Hourly rate of the basic daily wage x 200% x 130% x number of hours worked  Daily rate + Cost of Living Allowance) x 100%  Daily rate + COLA) x 200%] + (30% [Daily rate x 200%)  Hourly rate of the basic daily wage x 200% x 130% x 130% x number of hours worked |
| Special Holidays  (WORK)  (excess of eight hours (overtime work)  (work on rest day)  (rest day w/ OT | If the employee did not work, the “no work, no pay” principle shall apply, unless collective bargaining agreement (CBA) granting payment on a special day.  (Daily rate x 130%) + COLA)  Hourly rate of the basic daily wage x 130% x 130% x number of hours worked  (Daily rate x 150%) + COLA  Hourly rate of the basic daily wage x 150% x 130% x number of hours worked) |
| Gross Pay | Basic Pay + OT Pay + OTWND Pay + DOD Pay + Legal Holiday Pay + Special Holiday Pay = Gross Pay |
| Lates | (Rate/8 hours) / 60 minutes = Late |
| Total Deductions | Late + under time + Loan + SSS + Pagibig + PhiHealth + Tax |
| Net Pay | Gross Pay – Total Deductions = Net Pay |
| 13th Month Pay Computation | Daily rate \* 26 regular working days of an employee per month / 313, regular working days of an employee per year = result \* total number of days worked by the employee = 13th Month Pay. |

***How to print reports?***

Step 1. At the home window select reports. Different reports button will be displayed.

Step 2. Choose one of whether payroll records, bonus, 13 month pay, allowance reports etc.

Step 3. User will be directed to the corresponding window.

Step 4. Click print.

***How to print payroll reports?***

Step 1. At the reports options choose payroll records. Record form will be displayed.

Step 2. Search records by payroll period date from and date to, or payroll pay date

Step 3. Click Print records.

***How to print payslips?***

Step 1. At the reports options choose payroll records. Record form will be displayed.

Step 2. Search employees by ID, or by name and then Click Print payslip

**APPENDIX D – CODE LISTING**

**Connection**

package pk\_payroll;

import java.sql.\*;

import javax.swing.\*;

public class dbConn {

static Connection ConnectDb() {

throw new UnsupportedOperationException("Not supported yet.");

}

Connection conn = null;

public static Connection conectDB(){

try{

Class.forName("com.mysql.jdbc.Driver");

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/service\_management", "root", "");

return conn;

}catch (Exception e){

JOptionPane.showMessageDialog(null, e);

return null;

}

}

}

**Login form**

package pk\_payroll;

import java.awt.Toolkit;

import java.awt.event.KeyEvent;

import java.awt.event.WindowEvent;

import java.sql.\*;

import javax.swing.\*;

public class login extends javax.swing.JFrame {

private int attempt = 3;

Connection con=null;

ResultSet rs = null;

PreparedStatement pst=null;

public login() {

initComponents();

con=dbConn. conectDB();

}

private void cmd\_cancelActionPerformed(java.awt.event.ActionEvent evt) {

if (evt.getSource()==cmd\_cancel)

{

System.exit(0);

}

}

private void cmd\_loginMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

if (rb\_admin.isSelected())

{

String sql ="select \* from payroll\_login where user=? and pass=?";

try{

cmd\_login.setOpaque(false);

pst=con.prepareStatement(sql);

pst.setString(1, txt\_user.getText());

pst.setString(2, txt\_pass.getText());

rs=pst.executeQuery();

String username = txt\_user.getText();

String password = txt\_pass.getText();

if(username.isEmpty() && !password.isEmpty()){

JOptionPane.showMessageDialog(null, "Please Fill Username Field");

}

else if(password.isEmpty() && !username.isEmpty()){

JOptionPane.showMessageDialog(null, "Please Fill Password Field");

}

else if(username.isEmpty() && password.isEmpty()){

JOptionPane.showMessageDialog(null,"Please Fill Username and Password ");

}

else if(rs.next()){

pst = con.prepareStatement(sql);

pst.setString(1, txt\_user.getText());

pst.setString(2, txt\_pass.getText());

pst.execute();

JOptionPane.showMessageDialog(null,"Login Successfully");

rs.close();

PAYROLL\_HOME s =new PAYROLL\_HOME();

s.setVisible(true);

hide();

try {

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

else{

attempt--;

if(attempt==0)

{ JOptionPane.showMessageDialog(null,"No more attempts left.\nProgram will automatically terminated.");

System.exit(0);

}

JOptionPane.showMessageDialog(null,"Login Failed");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

else if ( rb\_acount.isSelected())

{

String sql ="select \* from login2 where user=? and pass=?";

try{

cmd\_login.setOpaque(false);

pst=con.prepareStatement(sql);

pst.setString(1, txt\_user.getText());

pst.setString(2, txt\_pass.getText());

rs=pst.executeQuery();

String username = txt\_user.getText();

String password = txt\_pass.getText();

if(username.isEmpty() && !password.isEmpty()){

JOptionPane.showMessageDialog(null, "Please Fill Username Field");

}

else if(password.isEmpty() && !username.isEmpty()){

JOptionPane.showMessageDialog(null, "Please Fill Password Field");

}

else if(username.isEmpty() && password.isEmpty()){

JOptionPane.showMessageDialog(null,"Please Fill Username and Password ");

}

else if(rs.next()){

pst = con.prepareStatement(sql);

pst.setString(1, txt\_user.getText());

pst.setString(2, txt\_pass.getText());

pst.execute();

JOptionPane.showMessageDialog(null,"Login Successfully");

rs.close();

PAYROLL\_HOME1 s =new PAYROLL\_HOME1();

s.setVisible(true);

hide();

try {

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

else{

attempt--;

if(attempt==0)

{ JOptionPane.showMessageDialog(null,"No more attempts left.\nProgram will automatically terminated.");

System.exit(0);

}

JOptionPane.showMessageDialog(null,"Login Failed");

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}

else

{

JOptionPane.showMessageDialog(null, " Please Select Access level");

}

}

private void txt\_passKeyTyped(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

char c = evt.getKeyChar();

if (!Character.isLetterOrDigit(c))

evt.consume();

if(evt.getSource().equals(txt\_pass)){

if(txt\_pass.getText().length() > 14){

evt.setKeyChar('\n');

}

}

}

public static void main(String args[]) {

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new login().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.ButtonGroup buttonGroup1;

private javax.swing.JButton cmd\_cancel;

private javax.swing.JButton cmd\_login;

private javax.swing.JLabel jLabel1;

private javax.swing.JLabel jLabel2;

private javax.swing.JLabel jLabel6;

private javax.swing.JPanel jPanel1;

private javax.swing.JPanel jPanel3;

private javax.swing.JPanel jPanel4;

private javax.swing.JPanel jPanel5;

private javax.swing.JPanel jPanel6;

private javax.swing.JRadioButton rb\_acount;

private javax.swing.JRadioButton rb\_admin;

private javax.swing.JPasswordField txt\_pass;

private javax.swing.JTextField txt\_user;

// End of variables declaration

}

**Employee Information form**

package pk\_payroll;

import java.awt.event.KeyEvent;

import java.sql.\*;

import java.util.Calendar;

import java.util.GregorianCalendar;

import javax.swing.\*;import net.proteanit.sql.DbUtils;

public class emp\_info extends javax.swing.JFrame {

Connection con=null;

ResultSet rs = null;

PreparedStatement pst=null;

public emp\_info() {

initComponents();

con=dbConn. conectDB();

CurrrentDate();}

private void clearObj() {

lbl\_pos.setText("");

lbl\_er.setText("");

lbl\_area.setText("");

}

private void txt\_idKeyPressed(java.awt.event.KeyEvent evt) {

if(evt.getKeyCode()==KeyEvent.VK\_BACK\_SPACE||evt.getKeyCode()==KeyEvent.VK\_BACK\_SPACE)

lbl\_pos.setText("");

lbl\_er.setText("");

lbl\_area.setText("");

// TODO add your handling code here:

}

public void CurrrentDate() {

Calendar cal = new GregorianCalendar();

int month = cal.get(Calendar.MONTH);

int year = cal.get(Calendar.YEAR);

int day = cal.get(Calendar.DAY\_OF\_MONTH);

txt\_date.setText( year +"/" +(month + 1) + "/" + day + " ");

int second = cal.get(Calendar.SECOND);

int minute = cal.get(Calendar.MINUTE);

int hour = cal.get(Calendar.HOUR);

txt\_times.setText( hour + ":" + (minute) );

}

private void txt\_idKeyReleased(java.awt.event.KeyEvent evt) {

String sql = "select \* from assignments where empid= ? ";

try {

pst = con.prepareStatement (sql);

pst.setString(1, txt\_id.getText());

rs = pst.executeQuery();

if (rs.next()) {

String a1 = rs.getString("Position");

lbl\_pos.setText(a1);

String b2 = rs.getString("Company");

lbl\_er.setText(b2);

String c3 = rs.getString("Address");

lbl\_area.setText(c3); }

}catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}finally {

try {

rs.close();

pst.close();

} catch (Exception e) {

}

}

}

try {

String sql = "select \* from empinfo where ID\_number = ? ";

pst = con.prepareStatement (sql);

pst.setString(1, txt\_id.getText());

rs = pst.executeQuery();

if (rs.next()) {

String add1 = rs.getString("FirstName");

lbl\_fname.setText(add1);

String add2 = rs.getString("LastName");

lbl\_lname.setText(add2);

String add3 = rs.getString("Age");

lbl\_age.setText(add3);

String add4 = rs.getString("Address");

lbl\_add.setText(add4);

String add5 = rs.getString("Birthdate");

lbl\_bd.setText(add5);

String add6 = rs.getString("Gender");

lbl\_gen.setText(add6);

String add7 = rs.getString("Email");

lbl\_email.setText(add7);

String add8 = rs.getString("Status");

lbl\_stat.setText(add8);

}

}catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

private void txt\_idKeyTyped(java.awt.event.KeyEvent evt) {

if(txt\_id.getText().length()>=20)

{

txt\_id.setText(txt\_id.getText().substring(0, 20));

}

char c = evt.getKeyChar();

if (!Character.isDigit(c))

evt.consume();

// TODO add your handling code here:

if(evt.getSource().equals(txt\_id)){

if(txt\_id.getText().length() > 8){

evt.setKeyChar('\n');

}

}

}

**Salary record form**

package pk\_payroll;

import java.awt.Cursor;

import java.awt.event.KeyEvent;

import static java.lang.Thread.sleep;

import java.sql.\*;

import java.util.Calendar;

import java.util.GregorianCalendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import net.proteanit.sql.DbUtils;

import net.sf.jasperreports.engine.design.\*;

import net.sf.jasperreports.view.JasperViewer;

import net.sf.jasperreports.engine.\*;

import javax.swing.\*;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

public class wage\_rec extends javax.swing.JFrame {

Connection con=null;

ResultSet rs = null;

PreparedStatement pst=null;

public wage\_rec() {

initComponents();

con=dbConn. conectDB();

TableUpdate();

CurrrentDate();

}

private void TableUpdate(){

sql = "select \* from payroll\_salary\_rate";

try{

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

tbl\_wage.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

public void CurrrentDate() {

Thread clock=new Thread(){

public void run(){

for(;;){

Calendar cal = new GregorianCalendar();

int month = cal.get(Calendar.MONTH);

int year = cal.get(Calendar.YEAR);

int day = cal.get(Calendar.DAY\_OF\_MONTH);

txt\_date.setText( year +"/" +(month + 1) + "/" + day + " ");

int second = cal.get(Calendar.SECOND);

int minute = cal.get(Calendar.MINUTE);

int hour = cal.get(Calendar.HOUR);

txt\_times.setText( hour + ":" + (minute) + ":" + second);

try{

sleep(1000);

}catch(InterruptedException ex){

Logger.getLogger(wage\_rec.class.getName()).log(Level.SEVERE,null,ex);

}finally {

try {

rs.close();

pst.close();

} catch (Exception e) {

}

}

}

}

};

clock.start();

}

private void tbl\_wageMouseClicked(java.awt.event.MouseEvent evt) {

lbl\_lname.setVisible(false);

lbl\_save.setEnabled(false);

lbl\_update.setEnabled(true);

lbl\_edit.setEnabled(false);

lbl\_add.setEnabled(false);

try {

int row = tbl\_wage.getSelectedRow();

String tbl\_click = (tbl\_wage.getModel().getValueAt(row, 0).toString());

sql= "SELECT \* FROM payroll\_salary\_rate where rate\_id ='"+tbl\_click+"' ";

pst=con.prepareStatement(sql);

rs=pst.executeQuery();

if (rs.next());

{

String adid = rs.getString("empid");

txt\_id.setText(adid);

String adFN = rs.getString("name");

lbl\_fname.setText(adFN);

String adLN = rs.getString("status");

lbl\_status.setText(adLN);

String adp = rs.getString("position");

lbl\_pos.setText(adp);

String ader = rs.getString("employeer");

lbl\_ER.setText(ader);

String adRA = rs.getString("daily\_rate");

txt\_drate.setText(adRA);

}

}

catch( Exception e){

}

}

private void txtid1KeyReleased(java.awt.event.KeyEvent evt) {

if (cb\_search.getSelectedIndex()==1)

{

String sql = " select \* from payroll\_salary\_rate where empid like ?";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid1.getText() + "%");

rs = pst.executeQuery();

tbl\_wage.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

if (cb\_search.getSelectedIndex()==2)

{

String sql = " select \* from payroll\_salary\_rate where name like ?";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid1.getText() + "%");

rs = pst.executeQuery();

tbl\_wage.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

// TODO add your handling code here:

}

private void txt\_idKeyReleased(java.awt.event.KeyEvent evt) {

String pos = lbl\_pos.getText();

String name = lbl\_fname.getText();

if(!name.isEmpty() && pos.isEmpty())

{

JOptionPane.showMessageDialog(null, "Employee has not assign, please search another Employee ");

}

else{

{

String sql = "select \* from assignments where empid= ? ";

try {

pst = con.prepareStatement (sql);

pst.setString(1, txt\_id.getText());

rs = pst.executeQuery();

if (rs.next()) {

String a1 = rs.getString("Position");

lbl\_pos.setText(a1);

String b2 = rs.getString("Company");

lbl\_ER.setText(b2);

}

}catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}finally {

try {

rs.close();

pst.close();

} catch (Exception e) {

}

}

}

try {

String sql = "select \* from empinfo where ID\_number = ? ";

pst = con.prepareStatement (sql);

pst.setString(1, txt\_id.getText());

rs = pst.executeQuery();

if (rs.next()) {

String add1 = rs.getString("FirstName");

lbl\_fname.setText(add1);

String add2 = rs.getString("LastName");

lbl\_lname.setText(add2);

String add7 = rs.getString("Status");

lbl\_status.setText(add7);

}

}catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

// TODO add your handling code here:

}

private void txt\_idKeyPressed(java.awt.event.KeyEvent evt) {

if(evt.getKeyCode()==KeyEvent.VK\_BACK\_SPACE||evt.getKeyCode()==KeyEvent.VK\_BACK\_SPACE)

lbl\_fname.setText("");

lbl\_lname.setText("");

lbl\_status.setText("");

lbl\_pos.setText("");

lbl\_ER.setText("");

private void lbl\_updateMousePressed(java.awt.event.MouseEvent evt) {

String id1 = txt\_id.getText();

String name = lbl\_fname.getText();

if(id1.isEmpty())

{

JOptionPane.showMessageDialog(null, "Please click data on the table");

}

else{

sv = JOptionPane.showConfirmDialog(null,"Do you want to Update Employee Salary?","title", 1);

if (sv == 0)

{

sv = JOptionPane.showConfirmDialog(null,"Are you sure you want to Update Employee Salary?","title", 0);

if (sv == 0)

{

try{

String id= txt\_id.getText();

String f=lbl\_fname.getText();

String rt= lbl\_status.getText();

String dt= lbl\_pos.getText();

String p= lbl\_ER.getText();

String st= txt\_drate.getText();

String ed = txt\_date.getText();

sql = "UPDATE payroll\_salary\_rate set empid='"+id+"', name='"+f+"', status='"+rt+"',position='"+dt+"' ,employeer='"+p+"' , daily\_rate='"+st+"' ,effective\_date='"+ed+"' WHERE empid='"+id+"' " ;

pst = con.prepareStatement(sql);

pst.execute();

TableUpdate();

JOptionPane.showMessageDialog(null, "Employee Salary has been Successfully Updated");

TableUpdate();

clearObj();

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

}

}

// TODO add your handling code here:

}

private void lbl\_printMouseClicked(java.awt.event.MouseEvent evt) {

try{

JasperDesign jasperDesign = JRXmlLoader.load("report2.jrxml");

String sql = "select \* from payroll\_salary\_rate where empid ='" + txt\_id.getText() + "'";

JRDesignQuery newQuery = new JRDesignQuery();

newQuery.setText(sql);

jasperDesign.setQuery(newQuery);

JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, con);

JasperViewer.viewReport(jasperPrint, false);

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

// TODO add your handling code here:

}

private void lbl\_printallMouseClicked(java.awt.event.MouseEvent evt) {

try{

JasperDesign jasperDesign = JRXmlLoader.load("report2.jrxml");

String sql = "select \* from payroll\_salary\_rate";

JRDesignQuery newQuery = new JRDesignQuery();

newQuery.setText(sql);

jasperDesign.setQuery(newQuery);

JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, con);

JasperViewer.viewReport(jasperPrint, false);

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

// TODO add your handling code here:

}

private void lbl\_saveMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

String id = txt\_id.getText();

String name = lbl\_fname.getText();

String rate = txt\_drate.getText();

String s= rb\_s1.getText();

if(id.isEmpty())

{

JOptionPane.showMessageDialog(null, "There is no data to be save, please search employee's ID");

}

else if(name.isEmpty() && !id.isEmpty())

{

JOptionPane.showMessageDialog(null, "Employee ID does not exist, Please Search another ID");

}

else if(!id.isEmpty() && !name.isEmpty() && rate.isEmpty()){

JOptionPane.showMessageDialog(null, "Please Fill Employees daily rate");

}

else{

sv = JOptionPane.showConfirmDialog(null,"Do you want to ADD Employee Salary?","title", 0);

if (sv == 0)

{

sql = "INSERT into payroll\_salary\_rate (empid, name, status,position,employeer, daily\_rate,effective\_date) values(?, ?, ?,?,?,?,?) " ;

try{

pst = con.prepareStatement(sql);

pst.setString(1, txt\_id.getText());

pst.setString(2, lbl\_lname.getText()+" , "+lbl\_fname.getText());

if (rb\_s1.isSelected() || rb\_s2.isSelected()|| rb\_s3.isSelected()|| rb\_s4.isSelected()){

pst.setString(3, dep);

}

else if(rb\_none.isSelected()){

pst.setString(3, lbl\_status.getText());

}

else {

pst.setString(3, lbl\_status.getText());

}

pst.setString(4, lbl\_pos.getText());

pst.setString(5, lbl\_ER.getText());

pst.setString(6, txt\_drate.getText());

pst.setString(7, txt\_date.getText());

pst.execute();

JOptionPane.showMessageDialog(null, "Employee Salary has been Successfully added");

TableUpdate();

clearObj();

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

}

}

**Payroll Compute Form**

package pk\_payroll;

import java.awt.event.KeyEvent;

import static java.lang.Thread.sleep;

import java.sql.\*;

import java.util.Calendar;

import java.util.GregorianCalendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.\*;

import net.proteanit.sql.DbUtils;

/\*\*

\*

\* @author O7

\*/

public class payroll extends javax.swing.JFrame {

Connection con=null;

ResultSet rs = null;

PreparedStatement pst=null;

String sql , tmp;

int sv;

public payroll() {

initComponents();

con=dbConn. conectDB();

allfilCombo();

CurrrentDate();

}

private void clearObj() {

// TODO Auto-generated method stub

txt\_NDW.setText("");

txt\_BP.setText("0.00");

txt\_OT.setText("0.00");

dt\_fr.setText("");

dt\_to.setText("");

txt\_id.setText("");

txt\_name.setText("");

txt\_rate.setText("");

txt\_status.setSelectedItem("");

txt\_ndPay.setText("0.00");

txt\_shPay.setText("0.00");

txt\_lhPay.setText("0.00");

txt\_othours.setText("0");

txt\_nd.setText("0");

txt\_sh.setText("0");

txt\_lh.setText("0");

txt\_position.setText("");

txt\_er.setText("0.00");

txt\_trans.setText("0.00");

txt\_meal.setText("0.00");

txt\_cola.setText("0.00");

txt\_others.setText("0.00");

lbl\_tAlow.setText("0.00");

txt\_bon.setText("0.00");

txt\_13.setText("0.00");

txt\_late.setText("0.00");

txt\_Lhr.setText("0");

txt\_UT.setText("0.00");

txt\_uthr.setText("0");

txt\_loan.setText("0.00");

txt\_ph.setText("0.00");

txt\_pgibg.setText("0.00");

txt\_sss.setText("0.00");

txt\_tax.setText("0.00");

txt\_ILpay.setText("0.00");

txt\_SLpay.setText("0.00");

txt\_ILday.setText("0");

txt\_SLday.setText("0");

lbl\_total\_leave.setText("0.00");

txt\_grosPay.setText("0.00");

txt\_net.setText("0.00");

jLabel3.setText("0.00");

}

/\*\*

\* Creates new form payroll

\*/

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

private void allfilCombo()

{

try

{

sql="SELECT \* FROM payroll\_allowance\_setting";

pst=con.prepareStatement(sql);

rs=pst.executeQuery();

while (rs.next()){

String id=rs.getString("allowance\_type");

cb\_allowance.addItem(id);

}

}

catch(Exception e)

{

JOptionPane.showMessageDialog(null,e);

}

}

public void CurrrentDate() {

Thread clock=new Thread(){

public void run(){

for(;;){

Calendar cal = new GregorianCalendar();

int month = cal.get(Calendar.MONTH);

int year = cal.get(Calendar.YEAR);

int day = cal.get(Calendar.DAY\_OF\_MONTH);

txt\_date.setText( year +"/" +(month + 1) + "/" + day + " ");

int second = cal.get(Calendar.SECOND);

int minute = cal.get(Calendar.MINUTE);

int hour = cal.get(Calendar.HOUR);

txt\_times.setText( hour + ":" + (minute) + ":" + second);

try{

sleep(1000);

}catch(InterruptedException ex){

Logger.getLogger(wage\_rec.class.getName()).log(Level.SEVERE,null,ex);

}finally {

try {

rs.close();

pst.close();

} catch (Exception e) {

}

}

}

}

};

clock.start();

}

private void lbl\_calcMousePressed(java.awt.event.MouseEvent evt) {

if (pay\_period.getSelectedIndex()==1 )

{

Double r,dw, bp, otrate ,othr, otpay ,ndpay,nd, sh,shpay, lh, lhpay, tal, per,mid,otrB,Tbon,

il,ilpay,sl, slpay, gros ,sss ,tax ;

/\*\*basic pay\*\*/

dw=Double.parseDouble(txt\_NDW.getText());

r=Double.parseDouble(txt\_rate.getText());

bp=r\*dw;

txt\_BP.setText(""+bp);

/\*\*OT pay\*\*/

othr=Double.parseDouble(txt\_othours.getText());

otrate=(r/8)\*1.25;

otpay=othr\*otrate;

txt\_OT.setText((String.format("%.02f",otpay)));

/\*\*Night Diff pay\*\*/

nd=Double.parseDouble(txt\_nd.getText());

ndpay=(r\*.10)+r\*nd;

txt\_ndPay.setText((String.format("%.02f",ndpay)));

/\*\*Legal pay\*\*/

lh=Double.parseDouble(txt\_lh.getText());

if (lh>1)

{

lhpay=(r+15)\*2.0\*lh;

txt\_lhPay.setText((String.format("%.02f",lhpay)));

}

/\*\*Special pay\*\*/

sh=Double.parseDouble(txt\_sh.getText());

if (sh>1)

{

shpay=(r\*1.3)+15\*Double.parseDouble(txt\_sh.getText());;

txt\_shPay.setText((String.format("%.02f",shpay)));

}

/\*\*total allowance pay\*\*/

tal=Double.parseDouble(txt\_trans.getText())+Double.parseDouble(txt\_meal.getText())+Double.parseDouble(txt\_cola.getText())+Double.parseDouble(txt\_others.getText());

lbl\_tAlow.setText((String.format("%.02f",tal)));

/\*\*total BUNOS pay\*\*/

Tbon=Double.parseDouble(txt\_bon.getText());

txt\_bon.setText((String.format("%.02f",Tbon)));

/\*\*incentive leave pay\*\*/

il=Double.parseDouble(txt\_ILday.getText());

ilpay=il\*r;

txt\_ILpay.setText((String.format("%.02f",ilpay)));

/\*\*sick leave pay\*\*/

sl=Double.parseDouble(txt\_SLday.getText());

slpay=sl\*r;

txt\_SLpay.setText((String.format("%.02f",slpay)));

/\*\*leave pay total \*\*/

Double lTotal=Double.parseDouble(txt\_SLday.getText())+ Double.parseDouble(txt\_ILday.getText());

lbl\_total\_leave.setText((String.format("%.02f",lTotal)));

/\*\*late \*\*/

Double late =(r/8)\*Double.parseDouble(txt\_Lhr.getText());

txt\_late.setText((String.format("%.02f",late)));

/\*\*Udertime \*\*/

Double undertime =(r/8)\*Double.parseDouble(txt\_uthr.getText());

txt\_UT.setText((String.format("%.02f",undertime)));

/\*\* Duty on Day Off \*\*/

Double dodPay =(r\*1.35) \*Double.parseDouble(txt\_duty.getText());

txt\_dod.setText((String.format("%.02f",dodPay)));

/\*\*GROSS PAY\*\*/

bp=Double.parseDouble(txt\_BP.getText());

otpay=Double.parseDouble(txt\_OT.getText());

ndpay=Double.parseDouble(txt\_ndPay.getText());

lhpay=Double.parseDouble(txt\_lhPay.getText());

shpay=Double.parseDouble(txt\_shPay.getText());

ilpay=Double.parseDouble(txt\_ILpay.getText());

slpay=Double.parseDouble(txt\_SLpay.getText());

gros= bp + otpay + ndpay + lhpay + shpay + lTotal +tal+Tbon+dodPay;

txt\_grosPay.setText((String.format("%.02f",gros)));

Double totalDeduc = Double.parseDouble(txt\_late.getText())+Double.parseDouble(txt\_UT.getText());

jLabel3.setText((String.format("%.02f",totalDeduc)));

Double net=gros-totalDeduc;

txt\_net.setText((String.format("%.02f", net)) );

}

else if ( pay\_period.getSelectedIndex()==2)

{

Double r,dw, bp, otrate ,othr, otpay ,ndpay,nd, sh,shpay, lh, lhpay, tal,Tbon,

il,ilpay,sl, slpay, gros ,sss ,tax ;

/\*\*basic pay\*\*/

dw=Double.parseDouble(txt\_NDW.getText());

r=Double.parseDouble(txt\_rate.getText());

bp=r\*dw;

txt\_BP.setText(""+bp);

/\*\*OT pay\*\*/

othr=Double.parseDouble(txt\_othours.getText());

otrate=(r/8)\*1.25;

otpay=othr\*otrate;

txt\_OT.setText((String.format("%.02f",otpay)));

/\*\*Night Diff pay\*\*/

nd=Double.parseDouble(txt\_nd.getText());

ndpay=(r\*.10)+r\*nd;

txt\_ndPay.setText((String.format("%.02f",ndpay)));

/\*\*Legal pay\*\*/

lh=Double.parseDouble(txt\_lh.getText());

if (lh>1)

{

lhpay=(r+15)\*2.0\*lh;

txt\_lhPay.setText((String.format("%.02f",lhpay)));

}

/\*\*Special pay\*\*/

sh=Double.parseDouble(txt\_sh.getText());

if (sh>1)

{

shpay=(r\*1.3)+15\*Double.parseDouble(txt\_sh.getText());;

txt\_shPay.setText((String.format("%.02f",shpay)));

}

/\*\*total allowance pay\*\*/

tal=Double.parseDouble(txt\_trans.getText())+Double.parseDouble(txt\_meal.getText())+Double.parseDouble(txt\_cola.getText())+Double.parseDouble(txt\_others.getText());

lbl\_tAlow.setText((String.format("%.02f",tal)));

/\*\*total BUNOS pay\*\*/

Tbon=Double.parseDouble(txt\_bon.getText());

txt\_bon.setText((String.format("%.02f",Tbon)));

/\*\*incentive leave pay\*\*/

il=Double.parseDouble(txt\_ILday.getText());

ilpay=il\*r;

txt\_ILpay.setText((String.format("%.02f",ilpay)));

/\*\*sick leave pay\*\*/

sl=Double.parseDouble(txt\_SLday.getText());

slpay=sl\*r;

txt\_SLpay.setText((String.format("%.02f",slpay)));

/\*\*leave pay total \*\*/

Double lTotal=Double.parseDouble(txt\_SLday.getText())+ Double.parseDouble(txt\_ILday.getText());

lbl\_total\_leave.setText((String.format("%.02f",lTotal)));

/\*\*late \*\*/

Double late =(r/8)\*Double.parseDouble(txt\_Lhr.getText());

txt\_late.setText((String.format("%.02f",late)));

/\*\*Udertime \*\*/

Double undertime =(r/8)\*Double.parseDouble(txt\_uthr.getText());

txt\_UT.setText((String.format("%.02f",undertime)));

/\*\* Duty on Day Off \*\*/

Double dodPay =(r\*1.35) \*Double.parseDouble(txt\_duty.getText());

txt\_dod.setText((String.format("%.02f",dodPay)));

/\*\*GROSS PAY\*\*/

bp=Double.parseDouble(txt\_BP.getText());

otpay=Double.parseDouble(txt\_OT.getText());

ndpay=Double.parseDouble(txt\_ndPay.getText());

lhpay=Double.parseDouble(txt\_lhPay.getText());

shpay=Double.parseDouble(txt\_shPay.getText());

ilpay=Double.parseDouble(txt\_ILpay.getText());

slpay=Double.parseDouble(txt\_SLpay.getText());

gros= bp + otpay + ndpay + lhpay + shpay + lTotal +tal+Tbon+dodPay;

txt\_grosPay.setText((String.format("%.02f",gros)));

/\*\* sss contribution \*\*/

Double sser, pher,pier;

if (gros>=1000 && gros<=1249.99)

{

sss= 1000\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 1000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=1250 && gros<=1749.99)

{

sss= 1500\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 1500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=1750 && gros<=2249.99)

{

sss= 2000\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 2000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=2250 && gros<=2749.99)

{

sss= 2500\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 2500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=2750 && gros<=3249.99)

{

sss= 3000\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 3000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=3250 && gros<=3749.99)

{

sss= 3500\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 3500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=3750 && gros<=4249.99)

{

sss= 4000\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 4000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=4250 && gros<=4749.99)

{

sss= 4500\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 4500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=4750 && gros<=5249.99)

{

sss= 5000\*0.0363;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 5000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=5250 && gros<=5749.99)

{

sss= 5500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 5500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=5750 && gros<=6249.99)

{

sss= 6000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 6000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=6250 && gros<=6749.99)

{

sss= 6500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 6500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=6750 && gros<=7249.99)

{

sss= 7000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 7000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=7250 && gros<=7749.99)

{

sss= 7500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 7500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=7750 && gros<=8249.99)

{

sss= 8000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 8000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=8250 && gros<=8749.99)

{

sss= 8500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 8500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=8750 && gros<=9249.99)

{

sss= 9000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 9000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=9250 && gros<=9749.99)

{

sss= 9500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 9500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=9750 && gros<=10249.99)

{

sss= 10000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 10000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=10250 && gros<=10749.99)

{

sss= 10500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 10500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=10750 && gros<=11249.99)

{

sss= 11000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 11000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=11250 && gros<=11749.99)

{

sss= 11500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 11500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=11750 && gros<=12249.99)

{

sss= 12000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 12000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=12250 && gros<=12749.99)

{

sss= 12500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 12500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=12750 && gros<=13249.99)

{

sss= 13000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 13000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=13250 && gros<=13749.99)

{

sss= 13500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 13500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=13750 && gros<=14249.99)

{

sss= 14000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 14000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>=14250 && gros<=14749.99)

{

sss= 14500\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 14500\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

else if (gros>14750 )

{

sss= 15000\*0.0363 ;

txt\_sss.setText((String.format("%.02f",sss)));

sser= 15000\*0.0737;

ers.setText((String.format("%.02f",sser)));

}

/\*\* PhilHealth collection \*\*/

Double ph;

if (gros <8999.99)

{

ph= 8000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 8000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=9000 && gros<=9999.99)

{

ph= 9000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 9000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=10000 && gros<=10999.99)

{

ph= 10000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 10000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=11000 && gros<=11999.99)

{

ph= 11000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 11000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=12000 && gros<=12999.99)

{

ph= 12000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 12000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=13000 && gros<=13999.99)

{

ph= 13000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 13000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=14000 && gros<=14999.99)

{

ph= 14000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 14000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=15000 && gros<=15999.99)

{

ph= 15000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 15000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=16000 && gros<=16999.99)

{

ph= 16000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 16000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=17000 && gros<=17999.99)

{

ph= 17000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 17000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=18000 && gros<=18999.99)

{

ph= 18000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 18000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=19000 && gros<=19999.99)

{

ph= 19000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 19000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=20000 && gros<=20999.99)

{

ph= 20000\*0.0125;

txt\_ph.setText((String.format("%.02f",ph)));

pher=20000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=21000 && gros<=21999.99)

{

ph= 21000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 21000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=22000 && gros<=22999.99)

{

ph= 22000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 22000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=23000 && gros<=23999.99)

{

ph= 23000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 23000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=24000 && gros<=24999.99)

{

ph= 24000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 24000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=25000 && gros<=25999.99)

{

ph= 25000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 25000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=26000 && gros<=26999.99)

{

ph= 26000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 26000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=27000 && gros<=27999.99)

{

ph= 27000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 27000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=28000 && gros<=28999.99)

{

ph= 28000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 28000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=29000 && gros<=29999.99)

{

ph= 29000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 29000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=30000 && gros<=30999.99)

{

ph= 300000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 30000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=31000 && gros<=31999.99)

{

ph= 31000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 31000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=32000 && gros<=32999.99)

{

ph= 32000\*0.0125;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 32000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=33000 && gros<=33999.99)

{

ph= 33000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 33000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>=34000 && gros<=34999.99)

{

ph= 34000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 34000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

else if (gros>35000 )

{

ph= 35000\*0.0125 ;

txt\_ph.setText((String.format("%.02f",ph)));

pher= 35000\*0.0125 ;

erph.setText((String.format("%.02f",pher)));

}

/\*\* Pagibig collection \*\*/

Double pi;

if (gros<1499)

{

pi= gros\*0.01 ;

txt\_pgibg.setText((String.format("%.02f",pi)));

pier= gros\*0.02 ;

erp.setText((String.format("%.02f",pier)));

}

else if (gros>1500 )

{

pi= gros\*0.02 ;

txt\_pgibg.setText((String.format("%.02f",pi)));

pier= gros\*0.02 ;

erp.setText((String.format("%.02f",pier)));

}

/\*\* withholding tax \*\*/

Double a=0.00, b=41.67,c=208.33,d=708.33,e=1875.00,f=4166.00,g=10416.67;

Double taxaIncome= gros-Double.parseDouble(txt\_sss.getText())-Double.parseDouble(txt\_ph.getText())-Double.parseDouble(txt\_pgibg.getText());

if (txt\_status.getSelectedIndex()==1 || txt\_status.getSelectedIndex()==2 )

{

if (gros>4167 && gros<4999.99)

{

tax=(taxaIncome-4167)\*0.05+a;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>5000 && gros<6666.99)

{

tax=(taxaIncome-5000)\*0.10+b;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>6667 && gros<9999.99)

{

tax=(taxaIncome-6667)\*0.15+c;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>10000 && gros<15832.99)

{

tax=(taxaIncome-10000)\*0.20+d;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>15833 && gros<24999.99)

{

tax=(taxaIncome-15833)\*0.25+e;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>25000 && gros<45832.99)

{

tax=(taxaIncome-25000)\*0.30+f;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>45833)

{

tax=(taxaIncome-45833)\*0.32+g;

txt\_tax.setText((String.format("%.02f",tax)));

}

}

if (txt\_status.getSelectedIndex()==3)

{

if (gros>6250 && gros<7082.99)

{

tax=(taxaIncome-45833)\*0.05+a;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>7083 && gros<8749.99)

{

tax=(taxaIncome-7083)\*0.10+b;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>8750 && gros<12082.99)

{

tax=(taxaIncome-8750)\*0.15+c;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>12083 && gros<17916.99)

{

tax=(taxaIncome-12083)\*0.20+d;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>17917 && gros<27083.99)

{

tax=(taxaIncome-17917)\*0.25+e;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>27083 && gros<47916.99)

{

tax=(taxaIncome-27083)\*0.30+f;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>47917)

{

tax=(taxaIncome-23958)\*0.32+g;

txt\_tax.setText((String.format("%.02f",tax)));

}

}

if (txt\_status.getSelectedIndex()==4)

{

if (gros>8333 && gros<9166.99)

{

tax=(taxaIncome-8333)\*0.05+a;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>9167 && gros<10832.99)

{

tax=(taxaIncome-9167)\*0.10+b;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>10833 && gros<14166.99)

{

tax=(taxaIncome-10833)\*0.15+c;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>14167 && gros<29999.99)

{

tax=(taxaIncome-14167)\*0.20+d;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>20000 && gros<29166.99)

{

tax=(taxaIncome-20000)\*0.25+e;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>29167 && gros<49999.99)

{

tax=(taxaIncome-50000)\*0.30+f;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>50000)

{

tax=(taxaIncome-50000)\*0.32+g;

txt\_tax.setText((String.format("%.02f",tax)));

}

}

if (txt\_status.getSelectedIndex()==5)

{

if (gros>10417 && gros<11249.99)

{

tax=(taxaIncome-10417 )\*0.05+a;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>11250 && gros<12916.99)

{

tax=(taxaIncome-11250)\*0.10+b;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>12917 && gros<16249.99)

{

tax=(taxaIncome-12917)\*0.15+c;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>16250 && gros<22082.99)

{

tax=(taxaIncome-16250)\*0.20+d;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>22083 && gros<31249.99)

{

tax=(taxaIncome-22083)\*0.25+e;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>31250 && gros<52082.99)

{

tax=(taxaIncome-31250)\*0.30+f;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>52083)

{

tax=(taxaIncome-52083)\*0.32+g;

txt\_tax.setText((String.format("%.02f",tax)));

}

}

if (txt\_status.getSelectedIndex()==6)

{

if (gros>12500 && gros<13332.99)

{

tax=(taxaIncome-12500)\*0.05+a;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>13333 && gros<14999.99)

{

tax=(taxaIncome-13333)\*0.10+b;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>15000&& gros<18332.99)

{

tax=(taxaIncome-15000)\*0.15+c;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>18333&& gros<24166.99)

{

tax=(taxaIncome-18333)\*0.20+d;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>24167 && gros<33332.99)

{

tax=(taxaIncome-24167)\*0.25+e;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>33333 && gros<52166.99)

{

tax=(taxaIncome-33333)\*0.30+f;

txt\_tax.setText((String.format("%.02f",tax)));

}

else if (gros>54167)

{

tax=(taxaIncome-26042)\*0.32+g;

txt\_tax.setText((String.format("%.02f",tax)));

}

}

Double totalDeduc = Double.parseDouble(txt\_late.getText())+Double.parseDouble(txt\_UT.getText())+

Double.parseDouble(txt\_sss.getText())+Double.parseDouble(txt\_ph.getText())+

Double.parseDouble(txt\_pgibg.getText())+ Double.parseDouble(txt\_tax.getText());

jLabel3.setText((String.format("%.02f",totalDeduc)));

Double net=gros-totalDeduc;

txt\_net.setText((String.format("%.02f", net)) );

}

// TODO add your handling code here:

}

private void lbal\_saveMousePressed(java.awt.event.MouseEvent evt) {

sv = JOptionPane.showConfirmDialog(null,"Do you want to ADD Employee Salary?","title", 0);

if (sv == 0)

{

try{

String sql = "INSERT into payroll\_paycompute (empid, date\_fr,date\_to, pay\_date, no\_day\_work,daily\_rate, basic\_pay, ot, otnd,"

+ " duty\_on\_dayoff,sp\_holiday, l\_holiday, total\_alowance, total\_bunus,13mp,leave\_pay,"

+ "deduction, gross\_pay, net\_pay ) values(?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?) " ;

pst = con.prepareStatement(sql);

pst.setString(1, txt\_id.getText());

pst.setString(2, dt\_fr.getText());

pst.setString(3, dt\_to.getText());

pst.setString(4, txt\_date.getText());

pst.setString(5, txt\_NDW.getText());

pst.setString(6, txt\_rate.getText());

pst.setString(7, txt\_BP.getText());

pst.setString(8, txt\_OT.getText());

pst.setString(9, txt\_ndPay.getText());

pst.setString(10, txt\_dod.getText());

pst.setString(11, txt\_shPay.getText());

pst.setString(12, txt\_lhPay.getText());

pst.setString(13, lbl\_tAlow.getText());

pst.setString(14, txt\_bon.getText());

pst.setString(15, txt\_13.getText());

pst.setString(16, lbl\_total\_leave.getText());

pst.setString(17, jLabel3.getText());

pst.setString(18, txt\_grosPay.getText());

pst.setString(19, txt\_net.getText());

pst.execute();

Double sstotal = Double.parseDouble(txt\_sss.getText()) + Double.parseDouble(ers.getText());

String SSS = "INSERT into payroll\_collec\_sss (empid,name , ee\_cotri,er\_contri, totalContribution, date ) values(?,?,?,?,?,?) " ;

pst = con.prepareStatement(SSS);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_sss.getText());

pst.setString(4, ers.getText());

pst.setString(5, sstotal.toString());

pst.setString(6, txt\_date.getText());

pst.execute();

Double pitotal = Double.parseDouble(txt\_pgibg.getText()) + Double.parseDouble(erp.getText());

String pagibig = "INSERT into payroll\_collec\_pagibig (empid,name , ee\_contri, er\_contri, total,date ) Values(?,?,?,?,?,?) " ;

pst = con.prepareStatement(pagibig);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_pgibg.getText());

pst.setString(4, erp.getText());

pst.setString(5, pitotal.toString());

pst.setString(6, txt\_date.getText());

pst.execute();

Double phtotal = Double.parseDouble(txt\_ph.getText()) + Double.parseDouble(erph.getText());

String ph = "INSERT into payroll\_collec\_philhealth (empid,name , ee\_share, er\_share,total,date ) Values(?,?,?,?,?,?) " ;

pst = con.prepareStatement(ph);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_ph.getText());

pst.setString(4, erp.getText());

pst.setString(5, phtotal.toString());

pst.setString(6, txt\_date.getText());

pst.execute();

String tax = "INSERT into payroll\_collec\_wtax (empid,name , tax,date ) Values(?,?,?,?) " ;

pst = con.prepareStatement(tax);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_tax.getText());

pst.setString(4, txt\_date.getText());

pst.execute();

String deduc = "INSERT into payroll\_deduction (emp\_id,late , undertime,loan, sss,philhealth, pagibig, tax, totaldeduc,date ) Values(?,?,?,?,?,?,?,?,?,?) " ;

pst = con.prepareStatement(deduc);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_late.getText());

pst.setString(3, txt\_UT.getText());

pst.setString(4, txt\_loan.getText());

pst.setString(5, txt\_sss.getText());

pst.setString(6,txt\_ph.getText());

pst.setString(7, txt\_pgibg.getText());

pst.setString(8, txt\_tax.getText());

pst.setString(9, jLabel3.getText());

pst.setString(10, txt\_date.getText());

pst.execute();

Double amount = Double.parseDouble(txt\_trans.getText());

if (amount >1)

{

String alow = "INSERT into payroll\_allowance (empid,name , allowance\_type,allowance\_amount,date\_paid ) Values(?,?,?,?,?) " ;

pst = con.prepareStatement(alow);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, lbl\_trans.getText());

pst.setString(4, txt\_trans.getText());

pst.setString(5, txt\_date.getText());

pst.execute();

}

Double amount1 = Double.parseDouble(txt\_meal.getText());

if (amount1 >1)

{

String alow1 = "INSERT into payroll\_allowance (empid,name , allowance\_type,allowance\_amount,date\_paid ) Values(?,?,?,?,?) " ;

pst = con.prepareStatement(alow1);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, lbl\_meal.getText());

pst.setString(4, txt\_meal.getText());

pst.setString(5, txt\_date.getText());

pst.execute();

}

Double amount2 = Double.parseDouble(txt\_cola.getText());

if (amount2 >1)

{

String alow3 = "INSERT into payroll\_allowance (empid,name , allowance\_type,allowance\_amount,date\_paid ) Values(?,?,?,?,?) " ;

pst = con.prepareStatement(alow3);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, lbl\_cola.getText());

pst.setString(4, txt\_cola.getText());

pst.setString(5, txt\_date.getText());

pst.execute();

}

Double amount3 = Double.parseDouble(txt\_others.getText());

if (amount3 >1)

{

String alow4 = "INSERT into payroll\_allowance (empid,name , allowance\_type,allowance\_amount,date\_paid ) Values(?,?,?,?,?) " ;

pst = con.prepareStatement(alow4);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, lbl\_other.getText());

pst.setString(4, txt\_others.getText());

pst.setString(5, txt\_date.getText());

pst.execute();

}

Double Bamount = Double.parseDouble(txt\_bon.getText());

if (Bamount >1)

{

String bonus = "INSERT into payroll\_bonus (empid,name ,bon\_amount,date\_pay ) Values(?,?,?,?) " ;

pst = con.prepareStatement(bonus);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_bon.getText());

pst.setString(4, txt\_date.getText());

pst.execute();

}

Double mp = Double.parseDouble(txt\_13.getText());

if (mp >1)

{

String m13 = "INSERT into payroll\_13mp (empid,name ,amount,date\_pay ) Values(?,?,?,?) " ;

pst = con.prepareStatement(m13);

pst.setString(1, txt\_id.getText());

pst.setString(2, txt\_name.getText());

pst.setString(3, txt\_13.getText());

pst.setString(4, txt\_date.getText());

pst.execute();

}

JOptionPane.showMessageDialog(null, "Employee Salary has been Successfully added");

clearObj();

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

// TODO add your handling code here:

}

**Payroll Record Form**

package pk\_payroll;

import static java.lang.Thread.sleep;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.Calendar;

import java.util.GregorianCalendar;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

import javax.swing.JTextField;

import net.proteanit.sql.DbUtils;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JRDesignQuery;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

import net.sf.jasperreports.view.JasperViewer;

public class pay\_rec extends javax.swing.JFrame {

Connection con=null;

ResultSet rs = null;

PreparedStatement pst=null;

String sql , tmp;

int sv;

public pay\_rec() {

initComponents();

con=dbConn. conectDB();

TableUpdate();

CurrrentDate();

}

private void TableUpdate(){

sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid ORDER BY payroll\_paycompute.pay\_date DESC";

try{

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

public void CurrrentDate() {

Thread clock=new Thread(){

public void run(){

for(;;){

Calendar cal = new GregorianCalendar();

int month = cal.get(Calendar.MONTH);

int year = cal.get(Calendar.YEAR);

int day = cal.get(Calendar.DAY\_OF\_MONTH);

txt\_date.setText( (month + 1) + "/" + day + "/" + year + " ");

int second = cal.get(Calendar.SECOND);

int minute = cal.get(Calendar.MINUTE);

int hour = cal.get(Calendar.HOUR);

txt\_times.setText( hour + ":" + (minute) + ":" + second);

try{

sleep(1000);

}catch(InterruptedException ex){

Logger.getLogger(wage\_rec.class.getName()).log(Level.SEVERE,null,ex);

}finally {

try {

rs.close();

pst.close();

} catch (Exception e) {

}

}

}

}

};

clock.start();

}

private void txtidKeyReleased(java.awt.event.KeyEvent evt) {

// TODO add your handling code here:

if (sc.getSelectedIndex()==1)

{

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " where payroll\_salary\_rate.empid like ? order by payroll\_paycompute.pay\_date DESC";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid.getText() + "%");

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

else if (sc.getSelectedIndex()==2)

{

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " where payroll\_salary\_rate.name like ? order by payroll\_paycompute.pay\_date DESC";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid.getText() + "%");

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

else if (sc.getSelectedIndex()==3)

{

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " where payroll\_paycompute.date\_fr like ? order by payroll\_paycompute.date\_fr DESC";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid.getText() + "%");

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

else if (sc.getSelectedIndex()==4)

{

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " where payroll\_paycompute.date\_to like ? order by payroll\_paycompute.date\_to DESC";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid.getText() + "%");

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

else if (sc.getSelectedIndex()==5)

{

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " where payroll\_paycompute.pay\_date like ? order by payroll\_paycompute.pay\_date DESC";

try{

pst = con.prepareStatement(sql);

pst.setString(1, txtid.getText() + "%");

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}

catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

}

private void jButton1MouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

try{

JasperDesign jasperDesign = JRXmlLoader.load("report5PS.jrxml");

String sql = "select payroll\_paycompute.empid,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay,\n" +

"\n" +

" payroll\_salary\_rate.name,\n" +

" payroll\_salary\_rate.status,\n" +

" payroll\_salary\_rate.position,\n" +

" payroll\_salary\_rate.employeer,\n" +

" payroll\_salary\_rate.effective\_date,\n" +

"\n" +

" payroll\_collec\_sss.ee\_cotri,\n" +

"\n" +

" payroll\_collec\_philhealth.ee\_share,\n" +

"\n" +

" payroll\_collec\_pagibig.ee\_contri,\n" +

" payroll\_collec\_wtax.tax,\n" +

"\n" +

"\n" +

" payroll\_deduction.late,\n" +

" payroll\_deduction.undertime,\n" +

" payroll\_deduction.loan from payroll\_paycompute INNER JOIN payroll\_salary\_rate ON payroll\_paycompute.empid = payroll\_salary\_rate.empid\n" +

" INNER JOIN payroll\_collec\_wtax ON payroll\_paycompute.empid = payroll\_collec\_wtax.empid\n" +

" INNER JOIN payroll\_collec\_sss payroll\_collec\_sss ON payroll\_paycompute.empid = payroll\_collec\_sss.empid\n" +

" INNER JOIN payroll\_collec\_philhealth ON payroll\_paycompute.empid = payroll\_collec\_philhealth.empid\n" +

" INNER JOIN payroll\_collec\_pagibig ON payroll\_paycompute.empid = payroll\_collec\_pagibig.`empid`\n" +

" INNER JOIN payroll\_deduction ON payroll\_paycompute.empid = payroll\_deduction.emp\_id where payroll\_paycompute.empid ='" + txtid.getText() + "'";

JRDesignQuery newQuery = new JRDesignQuery();

newQuery.setText(sql);

jasperDesign.setQuery(newQuery);

JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, con);

JasperViewer.viewReport(jasperPrint, false);

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

}

private void jButton2MouseClicked(java.awt.event.MouseEvent evt) {

String dp = ((JTextField) datepay.getDateEditor().getUiComponent()).getText();

try{

JasperDesign jasperDesign = JRXmlLoader.load("report6.jrxml");

String sql = " SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid "

+ " ORDER BY payroll\_paycompute.pay\_date DESC ";

JRDesignQuery newQuery = new JRDesignQuery();

newQuery.setText(sql);

jasperDesign.setQuery(newQuery);

JasperReport jasperReport = JasperCompileManager.compileReport(jasperDesign);

JasperPrint jasperPrint = JasperFillManager.fillReport(jasperReport, null, con);

JasperViewer.viewReport(jasperPrint, false);

} catch (Exception e){

JOptionPane.showMessageDialog(null, e);

}

// TODO add your handling code here:

}

private void searchActionPerformed(java.awt.event.ActionEvent evt) {

String dp = ((JTextField) datepay.getDateEditor().getUiComponent()).getText();

String sql = "SELECT payroll\_salary\_rate.empid,payroll\_salary\_rate.name,\n" +

" payroll\_paycompute.date\_fr,\n" +

" payroll\_paycompute.date\_to,\n" +

" payroll\_paycompute.pay\_date,\n" +

" payroll\_paycompute.no\_day\_work,\n" +

" payroll\_paycompute.daily\_rate,\n" +

" payroll\_paycompute.basic\_pay,\n" +

" payroll\_paycompute.ot,\n" +

" payroll\_paycompute.otnd,\n" +

" payroll\_paycompute.duty\_on\_dayoff,\n" +

" payroll\_paycompute.sp\_holiday,\n" +

" payroll\_paycompute.l\_holiday,\n" +

" payroll\_paycompute.total\_alowance,\n" +

" payroll\_paycompute.total\_bunus,\n" +

" payroll\_paycompute.13mp,\n" +

" payroll\_paycompute.leave\_pay,\n" +

" payroll\_paycompute.deduction,\n" +

" payroll\_paycompute.gross\_pay,\n" +

" payroll\_paycompute.net\_pay\n" +

" from payroll\_salary\_rate INNER JOIN payroll\_paycompute ON payroll\_salary\_rate.empid = payroll\_paycompute.empid wherepayroll\_paycompute.pay\_date '" + dp + "' ";

try{

pst = con.prepareStatement(sql);

rs = pst.executeQuery();

tbl\_payrec.setModel(DbUtils.resultSetToTableModel(rs));

}catch(Exception e)

{

JOptionPane.showMessageDialog(null, e);

}

}