

**DESIGN AND IMPLEMENTATION OF EMPLOYEE
INFORMATION AND PAYROLL SYSTEM**

(USING CALEB UNIVERSITY AS A CASE STUDY)

BY

KOLAWOLE EMMANUEL OLAWALE

013/02/CSC/116

SUBMITTED TO

**DEPARTMENT OF COMPUTER SCIENCE,
MATHEMATICS AND PHYSICS**

COLLEGE OF PURE AND APPLIED SCIENCE

CALEB UNIVERSITY

IMOTA LAGOS, STATE.

**IN PARTIAL FULFILMENT OF THE REQUIREMENT
FOR THE AWARD OF A ABACHELOR OF SCIENCE
(HONOURS) DEGREE IN COMPUTER SCIENCE.**

JULY,2017

CERTIFICATION

This is to certify that the project work titled “Design and Implementation of Employee Information and Payroll System” was prepared and submitted by Kolawole Emmanuel Olawale to the department of computer science and it is accepted by the undersigned, meeting the requirement of the award of the bachelor of science degree (B.Sc) in computer science of the college of Pure And Applied Science in Caleb University, Imota, Lagos State.

Dr. T.M Fagboola (Project Supervisor)

Date

Dr. A.O Oloyede (Head of Department)

Date

DEDICATION

I dedicate this project to first and foremost The Almighty God who without a doubt has protected and guided me in my journey from inception till this day. The life I live remains dedicated to God and so is whatever I do including this work.

ACKNOWLEDGEMENT

I would like to acknowledge the help of God who helped me although my 4 years in Caleb University, Imota Lagos State. He also saw me through the program and gave me a willingness to learn and improve.

I also would like to appreciate my parents Mr and Mrs Kolawole (especially my Mother) who supported me throughout the program and my siblings, Mrs Oyeleke Elizabeth, Miss Kareem Damilola and Miss Kolawole Omotola for their love and encouragment. I also want to appreciate the effort of Mr and Mrs Lofinmakin, they supported me financially and at those times they always kindly provided for my needs. They believed so much in me.

I would like to appreciate my Head of Department, Dr A.O Oloyede, and lecturers most especially Professor Adu, Professor Longe, Dr. Oduroye, Mr Kayode, Mr Akinsanya, Mr Olaoye, Mr Adeyemi, Mr Wasiu, for their love, counsel and fatherly support. I want to specially appreciate my supervisor, Dr T.M Fagboola for making this project a success and all the support he rendered to make sure I finish the project in due time. He always carries me along and was firm but accommodating at the same time. He saw in me professional and helped me learn many new things. As a result of his good work, I was very well prepared and knew to a large extent what was expected of me and also had my expectations set for the program. I would like to appreciate Mr. Ojo, Mr Ayedun, Mr Fowora, Mr Olooto, Mrs Samuel for their support throughout my stay in Caleb University.

Finally, I would like to appreciate my friends in Caleb University most especially, Miss Agbede Justina, Mr Daniel, Miss Tamilore, Mr Micheal, Mr Lekan, Mr Tega, Mr Princewill, Mr Tosin, Mr Victor, Mr Feranmi, Mr Fiyin, Miss Sewa, Miss Yemisi, Miss Antonia, Mr Temi, Miss Emmanuella, Miss Bimbola, Miss Maureen, Miss Simisola for believing in me and for their support, love and care.

ABSTRACT

The function of this system is to calculate the payment of the employee and also contain the information of each employee in the organization. It stores information and records of all employees in a database. The system will be used by the staffs at the human resource department of Caleb University, Imota Lagos State, to calculate and record the data about the employees.

There are several modules in this system such as the employee's payment calculation and registration of new employees and also to update all employees details. The system helps the staffs to check and find information about employees quickly and systematically. The languages used by the researcher for the design and development of the system is JAVA SWING for the front-end and MySQL for the back-end.

The system is designed to prevent problems occurred when using the manual system. The methodology used in the development of this system is the V-Shaped methodology. The structure of V-Shaped methodology consists of the stage of planning, analysis, designing, testing, maintenance and implementation. The scope of the research is focusing more on Caleb University human resource department regarding the payment structure and personal data of the employees. In addition to other existing system, the payroll developed would be able to send pay slips directly to the employees' email address, print out pay slip with the pictures of the employees including their details and also send information of newly registered employees to their email and the school email for reviews and corrections.

TABLE OF CONTENTS

CERTIFICATION	2
DEDICATION.	3
ACKNOWLEDGEMENT	4
ABSTRACT.	6
TABLE OF CONTENTS	7
LIST OF FIGURES	9
LIST OF TABLES.	10
CHAPTER ONE .	11
INTRODUCTION	11
1.1 Background of the Study	11
1.2. Statement of the Problem	11
1.3. Aim and Objectives	12
1.4. Significance of the Study	12
1.5. Scope of the Study	13
1.6. Definition of Terms	13
CHAPTER TWO .	15
LITERATURE REVIEW	15
2.1 Overview of Payroll Systems	15
2.2.1 V Model .	16
2.2.2 The Waterfall Model	17
2.2.3 Prototyping.	18
2.2.4 The Spiral Model	19
2.3 Review of Software Development Tools	19
2.3.1 JAVA	19
2.3.2 PYTHON	21
2.3.3 PHP	22
2.3.4 MySQL	23
2.3.5 ORACLE.	24
2.3.6 SQL SERVER	24
2.4 Related Works.	25
2.4.1 Odyssey Payroll System	25
2.4.2 Infonetics Payroll System	26

2.4.3 OrangeHRM	26
2.4.4 SimpleHRM .	27
2.4.5 Pt. Karya Tama Rimba Cemerlang Payroll Software	28
2.5 Payroll System Development	29
2.5.1 Process Overview and Pre-Development Phase	30
2.5.2 Collecting and Organizing Data	30
2.5.3 Payroll Processing	30
2.5.4 Payroll Tax Filing	31
CHAPTER THREE	32
METHODOLOGY	32
3.1 Design and Implementation Approach	32
3.2 Development Life Cycle of CALPAS	32
3.2.1 Requirement Specification Gathering	32
3.2.2 Requirement Analysis	33
3.2.3 Program Design	34
3.2.4 System Design	40
3.2.5 Implementation/Development	59
3.2.6 Testing.	62
3.2.7 Maintenance .	63
3.3 CALPAS Assessment/Evaluation	64
3.3.1 Questionnaire	64
CHAPTER FOUR.	70
IMPLEMENTATION AND RESULTS	70
4.1 System Requirement Specification	70
4.1.1 Hardware Requirement Specifications:	70
4.1.2 Software Requirement Specifications:	70
4.2 Implementation .	70
4.2.1 Homepage: .	70
4.2.2 Admin Registration	71
4.2.3 Forget Password	72
4.2.4 Dashboard: .	72
4.2.5 New Employee Registration:	73
4.2.6 Update Employee	73

4.2.7 Payroll .	74
4.2.8 Reports .	74
4.3 USER ASSESSEMENT ANALYSIS	74
4.3.1 Adequacy and Coverage	75
4.3.2 Efficiency and Effectiveness	76
4.3.3 Productiveness	76
4.3.4 Elegance and User-friendliness	76
4.3.5 Quality assurance	76
4.3.6 Critical Evaluation	
Error! Bookmark not defined.	
CHAPTER FIVE	77
CONCLUSION AND RECOMMENDATION	77
5.1 Introduction.	77
5.2 Conclusion	77
5.3 Recommendation on Future Improvement	77
REFERENCES	78
APPENDIX	79

LIST OF FIGURES

Figure 2.1: V-Shape model (Nabil and Govardhan (2010))

Figure 2.2: Interface for the Odyssey Payroll System

Figure 2.3: Interface for Infonetics Payroll System: Setup Screen – Payroll by Account

Figure 2.4: Snapshot of OrangeHRM interface

Figure 2.5: Snapshot of SimpleHRM interface

Figure 2.6: Screenshots Pt. Karya Tama Rimba Cemerlang Payroll Software

Figure 3.1: Dataflow Diagram of CALPAS

Figure 3.2: Class Diagram of CALPAS

Figure 3.3: Packages Diagram for CALPAS

Fig 3.4: Use case diagrams for CALPAS

Fig 3.5: Activity Diagram for CALPAS

Figure 3.5: Sequence diagram for the modules of CALPAS

Figure 4.1: Screenshot of CALPAS Homepage

Figure 4.2: Screenshot of CALPAS Admin Registration

Figure 4.3: Screenshot of CALPAS Forgotten Password Page

Fig 4.4: Screenshot of CALPAS Dashboard

Fig 4.5: Screenshot of CALPAS Employee Registration Page

Fig 4.6: Screenshot of CALPAS Employee Update Page

Fig 4.7: Screenshot of CALPAS Employee Payroll Page

LIST OF TABLES

Table 3.1: Admin Registration Table

Table 3.2: Employee Registration Table

Table 3.3: Leave Application Table

Table 3.4: Payroll Table

Table 4.1 User Assessment Analysis

CHAPTER ONE

INTRODUCTION

Over the last few centuries, human beings have experienced two major revolutions: the industrial revolution and the electronic revolution. The former transformed our society from being agriculturally based, whereas the latter transformed our society from being mechanically based to electronically based. In 21st century a lot of manual technics is now being implemented electronically. Payroll is not an exemption. Implementation of this in Caleb University is the goal of this work.

1.1 Background of the Study

Payroll is a company's list of its employees. It is mostly being referred to as the company's records of its employees' salaries and wages, bonuses, and taxes, company's department that calculates and pays these and so on. Payroll design, testing and implementation of Caleb University Payroll System (CALPAS) is the goal of this work. The department uses the manual systems of payroll. As employee increases, there is need to have a computerized payroll system. Therefore, the aim of the project is to develop a computerize payroll software for the university.

1.2. Statement of the Problem

Manual systems of payroll always put pressure on people to be correct in all aspect of their work. It is too easy to accidentally switch details and end up with inconsistency in data entry or in hand written orders. It can be time consuming and expensive. The function of this work is to

design and implement a payroll system owing to the fact that there exist no automated means of managing employees' salaries, records and so on in the university

Other problems caused by the manual payroll system are:

- i. Inconsistence data entry, prone to errors information.
- ii. Cost of training employee on payroll is expensive
- iii. Reduction in sharing information and customer services.
- iv. It is time consuming and costly in reports production.
- v. Data is not secure.
- vi. Duplication of data is possible.

This project builds a computerized payroll system that will reduce and take care of problems associated by manual method of managing and processing employees' salaries, records and so on in the university.

1.3. Aim and Objectives

The aim of this study is to develop a computerized payroll system for Caleb University. The specific objectives are to:

- i. design a computerized payroll system for Caleb University Imota Lagos State.
- ii. implement the designed payroll system in (i)
- iii. evaluate the performance of the implemented payroll system in (ii) qualitatively

1.4. Significance of the Study

This study will aid in reducing errors and fraud associated with the manual processing of employees' salaries and records, increase speed and also enhance the technology edge in Caleb

University, Imota Lagos if successfully implemented. Manual payroll systems in Caleb University will be totally eliminated with CALPAS in place.

1.5. Scope of the Study

The payroll system is developed for Caleb University only and might not fit into other similar systems perfectly and would be called CALPAS (Caleb University Payroll System).

1.6. Definition of Terms

This section gives a definition and explanation of some of the terms used in the project.

- i. Employee Data: This consist of employee's information in the organization. It consists of the employee identification number, employee name, pay rate, bio data, contact information, payment information, personal information, employment information etc.
- ii. Net Pay: This is the total salary amount that would be paid to an employee after all the deductions are subtracted from the gross pay.
- iii. Gross Pay: The gross pay is the amount an employee earns before the deductions are subtracted.
- iv. Payroll: A payroll is a company's list of its employees. It term is mostly refers to as the amount of money that a company pays its employees, the company's records of its employees' salaries and wages, bonuses, and taxes, company's department that calculates and pays these etc.
- v. Payroll System: This is a software which keeps the organization company's list or records of its employees. It is often referring to the total amount of money that a company pays to its employees. It is organizing and arranges all the tasks of employee payment and the filing of employee taxes.

- vi. Pay Slips: a note given to an employee when salary is being paid, detailing the amount of pay given and the tax and insurance deducted.
- vii. Employer: a someone or an organization that employs people in the organization.
- viii. Employee: a someone that is employed for wages or salary, especially at nonexecutive level.
- ix. W-4 tax form: This is a form filled by an employee to specify his or her tax status to the employer. It tells the employer the amount of tax to withhold from employee's paycheck.
- x. Internal Revenue Service (IRS): A states government agency that is responsible for the collection and enforcement of taxes.

CHAPTER TWO

LITERATURE REVIEW

2.1 Overview of Payroll Systems

In organizations, there are several departments and each department has payroll section to manage payroll activities while some have a department that handles majorly the payroll. Irrespective of the department that handles the payroll, each section of the payroll has to perform necessary operations like data collection and preparation, entry, updates monitoring and reporting of data. With a system like this, payroll section would be able to keep a record of employees including their personnel data, pay slips, allowances and deductions etc. Net pay of each employee is calculated by this allowances and deductions mentioned according to the company rules. The individual pay slips are printed out as receipt or mailed to the employee. Pay bands, allowances, deductions, and tax information are updated if there is any amendment in salary structure. The proposed payroll system, CALPAS, will do the afro mentioned.

2.2 Software Development Life Cycle Methodologies

Software development life cycle (SDLC) plays a vital role in developing a new or existing system as it describes the complete requirement of the system. It is used by developers and will be the basis during the testing phase. Common methodologies include waterfall, prototyping, iterative and incremental development, spiral development, rapid application development, extreme programming and various types of agile methodology.

2.2.1 V Model

In Nabil and Govardhan (2010) in the paper that compared the five models of software engineering and defined V-Shaped life cycle as a sequential path of execution of processes. Each phase is completed before the next phase begins. Testing is emphasized in this method more than the waterfall model. The testing procedures are developed early in the life cycle before any coding is done, during each of the phases preceding implementation. Requirements begin the life cycle model just like the waterfall model. Before development is started, a system test plan is created. The test plan focuses on meeting the functionality specified in requirements gathering.

The high-level design phase focuses on system architecture and design. An integration test plan is created in this phase in order to test the pieces of the software systems ability to work together. However, the low-level design phase lies where the actual software components are designed, and unit tests are created in this phase as well. The implementation phase is, again, where all coding takes place. Once coding is complete, the path of execution continues up the right side of the V where the test plans developed earlier are now put to use.

2.2.2.1 Advantages of V Model

- i. Simple and easy to use.
- ii. Each phase has specific deliverables.
- iii. Higher chance of success over the waterfall model due to the early development of test plans during the life cycle.
- iv. Works well for small projects where requirements are easily understood.

2.2.1.2 Disadvantages of V Model

- i. Very rigid like the waterfall model.
- ii. Flexibility and adjusting scope is difficult and expensive.

- iii. Software is developed during the implementation phase, so no early prototypes of the software are produced.
- iv. This Model does not provide a clear path for problems found during testing phases.

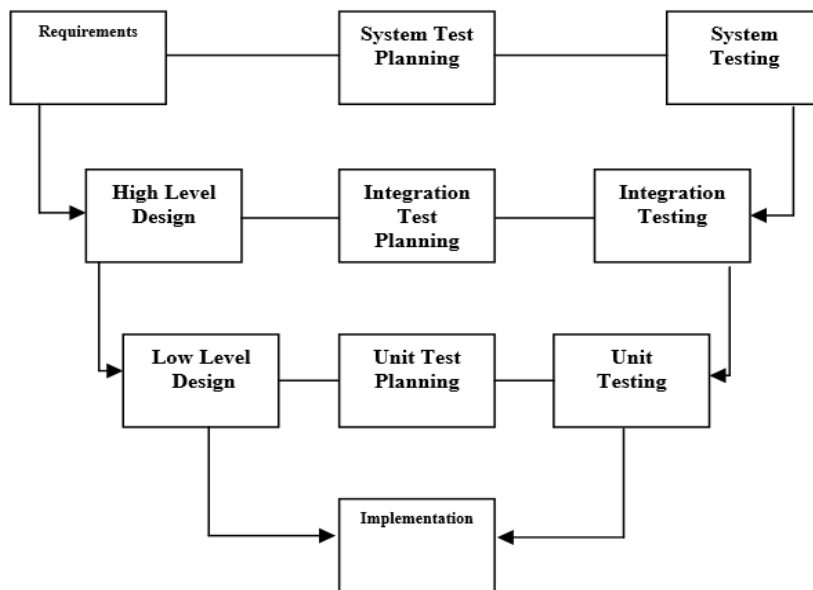


Figure 2.1: V-Shape model (Nabil and Govardhan (2010))

2.2.2 The Waterfall Model

Waterfall model is a sequential (non-iterative) design process, used in software development processes in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of conception, initiation, analysis, design, construction, testing, production/implementation and maintenance.

2.2.2.1 Advantages of Waterfall Model

1. Waterfall model is simple and easy to understand and use.
2. It is easy to manage due to the rigidity of the model
3. Phases are processed and completed one at a time.
4. It works well for smaller projects where requirements are very well understood.

2.2.2.2 Disadvantages of Waterfall Model

1. High amount of risk and uncertainty.
2. Poor model for long and ongoing projects.
3. It is not a good model for complex and object-oriented projects.
4. Once an application is in its testing stage, it is difficult to go back and change something that was not well-thought out in the concept stage.

2.2.3 Prototyping

A prototype is an early sample, model, or release of a product built to test a concept or process or to act as a thing to be replicated or learned from. It is a term used in a variety of contexts, including semantics, design, electronics, and software programming.

2.2.3.1 Advantages of Prototyping Model

1. Users are actively involved in the development.
2. The users get a better understanding of the system being developed.
3. Errors can be detected much earlier.
4. User feedback is available leading to better solutions.
5. Confusing or difficult functions can be identified.

2.2.3.2 Disadvantages of Prototyping Model

1. Leads to implementing and then repairing way of building systems.
2. Incomplete application may cause application not to be used.
3. It will increase the complexity of the system as scope of the system may expand beyond the original plans.

2.2.4 The Spiral Model

The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.

2.2.4.1 Advantages of Spiral Model

1. High amount of risk analysis hence, avoidance of risk is enhanced.
2. Good for large and mission-critical projects.
3. Strong approval and documentation control.
4. Additional functionality can be added at a later date.
5. Software is produced early in the software life cycle.

2.2.4.2 Disadvantages of Spiral Model

1. It is a costly model to use.
2. Risk analysis requires highly specific expertise.
3. Success is highly dependent on the risk analysis phase.
4. Doesn't work well for smaller projects.

2.3 Review of Software Development Tools

The following are various development tools and software that could be used for the system.

2.3.1 JAVA

According to Guillermo et al (2011) in Java in the High Performance Computing Arena: Research, Practice and Experience says that Java has become a leading programming language soon after its release, especially in web-based and distributed computing environments, and it is

an emerging option for High Performance Computing (HPC). The increasing interest in Java for parallel computing is based on its appealing characteristics: built-in networking and multithreading support, object orientation, platform independence, portability, type-safety, security, it has an extensive API and a wide community of developers, and finally, it is the main training language for computer science students. Moreover, performance is no longer an obstacle. The performance gap between Java and native languages such as C and Fortran has been narrowing for the last years, thanks to the Just-in-Time (JIT) compiler of the Java Virtual Machine (JVM) that obtains native performance from Java bytecode. However, the use of Java in HPC is being delayed by the lack of analysis of the existing programming options in this area and thorough and up-to-date evaluations of their performance, as well as the unawareness of the current research projects in Java for HPC, whose solutions are needed in order to boost its adoption.

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. Compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code that can run on any Java virtual machine (JVM) regardless of computer architecture. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

2.3.1.1 Advantages of JAVA

- i. It is simple.
- ii. It allows you to create modular program and reusable code.
- iii. It is easier to move from one computer to another.
- iv. It is secure.

- v. It is capable of performing several tasks simultaneously with a program.

2.3.1.2 Disadvantages of JAVA

- i. It is significantly slower and more memory consuming than natively compiled languages such as C or C++.
- ii. The default looks and feel of GUI applications written in Java using the swing toolkit is very different from native applications.

2.3.2 PYTHON

Python is a widely used high level programming language for general-purpose programming created by Guido Van Rossum and first released in 1991. It is an interpreted language and has a design philosophy which emphasizes code readability and a syntax which allows programmers to express concepts in fewer lines of code which might be used in languages such as C++.

2.3.2.1 Advantages of PYTHON

- i. It is easy to learn for even a novice developer.
- ii. Supports multiple systems and platform.
- iii. It is object-oriented programming driven.
- iv. It opens new avenues for you to create some out-of-the box gadgets.
- v. It makes web programming very flexible.
- vi. It gives rise to quick development by using less code.

2.3.2.2 Disadvantages of PYTHON

- i. It is slow

- ii. It is not a very good language for mobile development.
- iii. It is not good choice for memory intensive tasks.
- iv. It has limitations with database access.
- v. It is near impossible to build a high-graphic 3D game using python
- vi. It is not good for multi-processor/multi-core work.

2.3.3 PHP

PHP (Hypertext Preprocessor) is a widely used open source general-purpose programming language that is especially suited for web development and can be embedded into HTML.

2.3.3.1 Advantages of PHP

- i. It is an open source programming language.
- ii. It is relatively fast since it uses much system resources.
- iii. It is easy to use.
- iv. It is stable and maintained by many developers.
- v. It has powerful library support.
- vi. It is a built-in database connection module.
- vii. It can be run on many platforms.

2.3.3.2 Disadvantages of PHP

- i. Since it is an open sourced, so all people can see the source code if there are bugs in the source code.
- ii. It is not suitable for large applications.

- iii. It is a weak type of programming.

2.3.4 MySQL

MySQL is an open source database that is platform independent and can easily interface with a number of scripting languages; it works best with PHP though. The number of advantages of using MySQL which include the ability to handle stored procedures, triggers, SQL and UserDefined functions. It also offers a high-speed data load utility and support for various drivers (ODBC, JDBC, .NET, PHP). Deploying a MySQL database has proved to be cheap and easy as it doesn't require special hardware or software requirements, it can work well on any web server but most professionals recommend the apache web server. MySQL is an excellent database to use when developing web based applications because its platform independent and can easily interface with a number of scripting languages.

2.3.4.1 Advantages of MYSQL

- i. Redundancy can be avoided
- ii. Data can be shared
- iii. Standards can be enforced
- iv. Security restrictions can be applied
- v. Integrity can be maintained

2.3.4.2 Disadvantages of MYSQL

- i. It's got a few stability issues.
- ii. It suffers from relatively poor performance scaling.
- iii. Development is not community driven and hence has lagged.
- iv. Its functionality tends to be heavily dependent on Addons.

2.3.5 ORACLE

Oracle Database, also referred to as Oracle RDBMS or simply Oracle is an object relational database management system produced and marketed by Oracle Corporation.

Advantages of ORACLE

- i. It has the ability to group several transactions into the same batch for processing sets.
- ii. It offers other methods of improving your database performance.
- iii. It gives you flexibility of choosing to run your database in any operating system.

Disadvantages of ORACLE

- i. It is expensive.
- ii. It is difficult to learn and operate.

2.3.6 SQL SERVER

Microsoft SQL Server is a relational database management system developed by Microsoft. Its primary function is storing and retrieving data as requested by other software applications which may run either on the same computer or on another computer across a network (including the internet).

2.3.6.1 Advantages of SQL SERVER

- i. High speed
- ii. Well defined standards exist
- iii. No coding required.
- iv. Emergence of object oriented database management system.

- i. Difficulty in interfacing.
- ii. More features implemented in proprietary way.

2.4.1 Odyssey Payroll System

```

PET205M1                ***** PERSONNEL / PAYROLL *****                PET205M1
Nov 14,97                - Maintain an Employee's Attendance Days -                12:35 PM

Act Date      Day *Type Hours  Time Types that  Time  Available  Used in
-----
07/01/1997 Tue PERS 2.25 -----
07/02/1997 Wed SICK 1.87 -----
07/03/1997 Thu SICK 7.50 -----
                                Scroll:
                                PERS FLEX 15.00
                                SICK PERS 2.00 2.25
                                WACA SICK 9.37
                                WACA 115.50

SSN: 230680001 Position: 14120 Assignment Start: 07/01/1997 End:
Hours Per Day: 07.50 Reporting Period - Start: 07/01/1997 End: 07/15/1997
Name: DOUGLAS MARC
Add Copy Display Modify Purge

```

25 | Page

2.4.2 Infonetix Payroll System

The payroll system is used to maintain a list of employee details and information and is used to calculate and print payroll checks. In this payroll system, employees' hours are entered and distributed to appropriate department. Moreover, annualized tax tables are then used to automatically compute deductions before the check is printed on bank stock. Next, monthly and quarterly reports are used to prepare tax filings, and provide analytical information. Finally, form W-2's can be generated automatically at year-end.

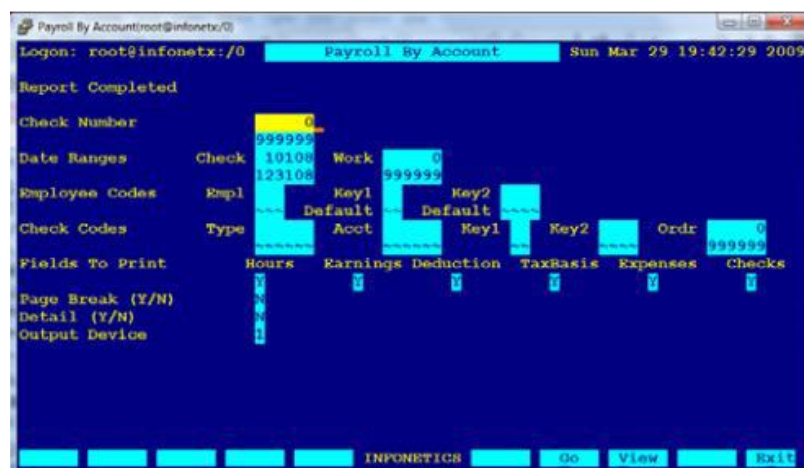


Figure 2.3: Interface for Infonetix Payroll System: Setup Screen – Payroll by Account

2.4.3 OrangeHRM

OrangeHRM is a powerhouse human resources tool that any small or midsize business can benefit from using. With OrangeHRM, you have options: You can download and install the system on your own hardware, or you can purchase a hosted solution. To get prices for the hosted solution, you have to contact them from their Request a Quote page. OrangeHRM's features include: fully modular, addons (benefits, employee self-service, training, budget, job and salary history, and so

on) for purchase, all standard HR functions (employees, leave, benefits, performance and so on), and more. The installation is fairly straight-forward. With a self-extracting Windows installer or full-source installations for Windows, Mac, and Linux, you can get OrangeHRM up and running on nearly every platform.

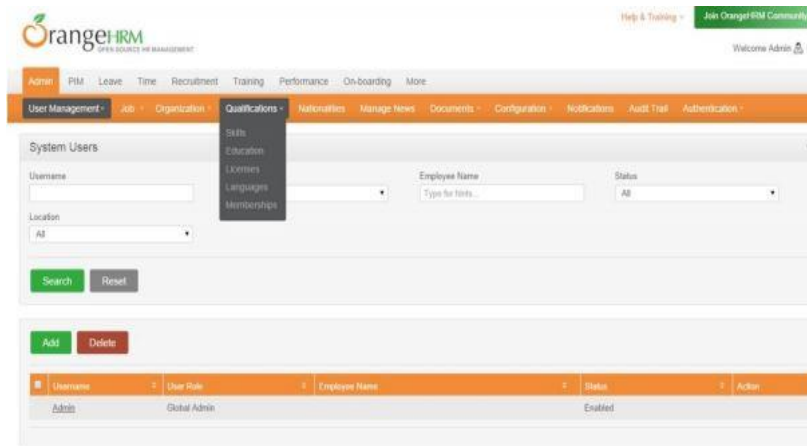


Figure 2.4: Snapshot of OrangeHRM interface

2.4.4 SimpleHRM

SimpleHRM offers an open source version of its professional platform. This version offers time management, and it can be installed on either a Windows Apache MySQL PHP(WAMP) or Linux Apache MySQL PHP(LAMP) server. Once installed, SimpleHRM offers every feature you need to solidify your HRM department: employee information, leave management, travel management, expense management, benefit management, and task reporting. SimpleHRM allows you to assign a Curriculum Vitae (CV) to an employee and define eligibility for rehire. Each major module offers plenty of granular control, and the user interface is well laid out.

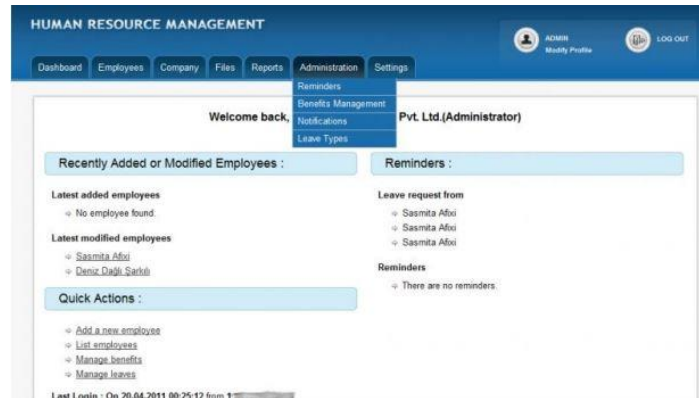


Figure 2.5: Snapshot of SimpleHRM interface

2.4.5 Pt. Karya Tama Rimba Cemerlang Payroll Software

In creating the new payroll system, Agustinus et-al carried out an observation in the company internal system. From the internal data, then the company will be able to design the new payroll system that can be customized based on the needs of the company through a more structured DFD, ERD, and tables. However, the new system is the integration of payroll and administration module. Parts of the payroll system are basic salary, munch money, bonuses, taxation, Jamsostek, THR and cash bon. The administration process comprises the workers' personal data, absenteeism, permission and leave, violation of rules and firing.

Based on the payroll application, the satisfaction level of PT. Karya Tama Rimba Cemerlang as the user is 76%, thus as a conclusion, the application process of the new payroll system has fulfilled the target and has the suitable system. Infact, this system able to support the company with a faster payroll report yet accurate.



Figure 2.6: Screenshots Pt. Karya Tama Rimba Cemerlang Payroll Software (Agustinus et-al, 2011)

2.5 Payroll System Development

Payroll system development refers to the steps involved in setting up and processing payroll. It starts with basic decisions such as how often to pay employees and whether to use a manual or computerized system, and ends when you finish processing the first batch of payroll checks. I am using the computerized system against the manual system used in Caleb University Imota Lagos State. Regardless of the system, the development process is vital to ensuring that each payroll check processed is accurate.

2.5.1 Process Overview and Pre-Development Phase

Payroll system development can be easier to understand by grouping tasks into a series of development phases. These include pre-development activities, data collection, payroll processing, and payroll tax accounting and filing. Pre-development activities include researching Fair Labor Standards and laws for your state, as well as IRS and state tax rules and regulations. In addition, this phase involves reviewing payroll best practices and establishing internal controls that set the stage for developing a secure, efficient system. Final steps involve decisions about how often to pay employees and whether to record time and attendance manually or electronically.

2.5.2 Collecting and Organizing Data

Steps in the data collection phase include data collection policies and practices for both new hires and current employees. Procedures for new hires facilitate the collection and organization of initial paperwork such as W-4, I-9 employment eligibility verification, insurance and retirement plan forms, as well as procedures for submitting new-hire paperwork to the appropriate federal and state agencies. Drafting written procedures current employees must follow when modifying their personal, contact or tax information is also a requirement. The data collection phase ends by establishing procedures for submitting and verifying employee timecard information.

2.5.3 Payroll Processing

Decisions regarding the type of payroll system and procedures for creating, printing and distributing paychecks are the main objectives of the payroll processing phase. For many small businesses, a computerized payroll system is the best choice, as it not only decreases processing time, but also increases accuracy. Options for distributing payroll include printing paychecks or

directly depositing checks into an employee's bank account. While each option has advantages and disadvantages, if your business has 10 or more employees on staff, you may find the direct-deposit option a more cost-effective option.

2.5.4 Payroll Tax Filing

The research you conducted in the pre-development phase will tell you whether the IRS and state tax agencies require you to deposit payroll tax withholdings and payroll tax reports quarterly or annually. State websites and the IRS Employers Tax Guide are also available for answering any additional questions. Filing options include using postal mail or setting up an online account to transmit tax payments electronically. Because strict compliance with payment and report filing due dates is so essential, choosing the online payment option -- which allows you to schedule payments in advance is often a better option.

CHAPTER THREE

METHODOLOGY

3.1 Design and Implementation Approach

After discussing the literature preview in the previous chapter, this chapter will discuss about the project planning phase and methodology use in the project. The sub topics is described and determined by the model to be used. Methodology is the function of selecting the objectives of the project and establishing the policies, procedures and programs necessary for achieving them. It involves planning for schedule development, budget development, project administration, leadership styles and conflict management. It is very important because project planning can eliminate or reduce uncertainty, improve efficiency of the operation, obtain a better understanding of the objective etc. The chapter will end with a questionnaire and analysis of the questionnaire.

3.2 Development Life Cycle of CALPAS

From the case study, every system has its own model to develop the system. Some developers use prototyping model, Spiral Model and so on. CALPAS is designed using V Model because it is simple and easy to use; each phase has specific deliverables, higher chance of success over the waterfall model due to the early development of test plans during the life cycle, works well for small projects where requirements are easily understood and so on.

3.2.1 Requirement Specification Gathering

This describes the needs of the user. The user of CALPAS wants computerized software better than the manual method of calculation that will carry out the following

- i. Have the basic information of the employee name, pay rate, employee number.
- ii. Edit, search and update employee information.

- iii. Apply for leave automatically.
- iv. Have records of the employee pay.
- v. Print out pay-slips.
- vi. Computes net pay
- vii. Record all the payroll data for subsequent processing
- viii. Prepare a payroll ledger or report.

3.2.2 Requirement Analysis

- i. Have The Basic Information of the Employee Name, Pay Rate, Employee Number: CALPAS user wants the system to have basic information of the employee such as the Full name, employee ID, date of birth, level, step, Address, Phone Number, State of Origin, Local Government area, Date Hired, Job Title, P.F.A, PIN, Bank Details and so on for the purpose of documentation and also to have the details of all employees in the database for references and knowing number of employee in the company.
- ii. Edit, Search and Update Employee Information: The user of CALPAS wants the software to be able to search, edit and update the employee's information. This is needed for modification, promotion and correction of wrong data created or inputted.
- iii. Apply for Leave Automatically: CALPAS should be able to create, generate and save leave for employees automatically and saved in a database for purpose of referral and payroll calculation.
- iv. Have Records of the Employee Pay: CALPAS should be able to record, save and print employees pay. Admin should be able to track each payment made to each employees and when the payment was made.

- v. Print out Pay-Slips: The system should be able to print payroll for each and every employee in the company.
- vi. Computes Net Pay: The system should also be able to compute net pay of the employees in the company.
- vii. Record All the Payroll Data for Subsequent Processing: The system should be able to compute and record all the payroll data for subsequent processing of the payroll system.
- viii. Prepare A Payroll Ledger or Report: The System should be able to prepare and also print out general report.

3.2.3 Program Design

The feature of CALPAS is to maintain complete employee information, payroll, leave report, pay slips information, payroll report. The roles involve in this system are the administrator and the super administrator. The role of the administrator is basically to handle the system, register, update and pay the employees and also to maintain all information including operations like modifying, updating the leave report in the system which will be handled by the school bursary administrator while the super administrator is to monitor the flow in the software and what the admin does using the software. The admin is allowed to register and create username and password. Modules defined the features (high level design) implemented in CALPAS. Provisioning defines what is needed or required in the modules. Functionality defines the ability of the modules and alert defines what it shows if the provisioning is correct or wrong.

3.1.4.1 Modules

The types of modules are:

1. Login

2. Administrator Registration
3. Forgotten Password
4. Reset Username and Password
5. Employee Registration
6. Find, Search and Update Employee
7. Payroll
8. Reports

Module 1:

Login: This describes how a user logs into the payroll system and starts when the employee wishes to login to the payroll system.

Provisioning:

Administrator Login:

1. Employee id
2. Password

Functionality:

1. This authenticates the Administrators.
2. If the user of the system is valid, it allows the user to enter into the system and use the system.

Alerts:

1. Alerting when unauthorised users try to access or manipulate the system.

Module 2:

Administrator Registration

The function of this module is for the administrator to register and be able to create his own username and password of his choice that won't be forgotten for the purpose of flexibility.

Provisioning:

1. Administrator ID
2. Email Address
3. Full Name
4. Date of Birth
5. Phone Number
6. Create Username
7. Password
8. Confirm Password

Functionality:

1. Administrator is allowed to register into the payroll using his own employee ID.

Alerts:

1. Alerting when the password entered with the confirm password isn't the same.

Module 3:

Forgotten Password: The function of this module is for employee to retrieve its password in case forgotten. He provides some details and the password is revealed.

Provisioning:

1. Email
2. Phone number
3. Date of Birth

Functionality:

1. It authenticates that the user forgot username and password and valid the user by reset the password in the reset password field.

Alerts:

1. Alerting if one of the details is not correct or corresponds to what is in the database.

Module 4:

Reset Username and Password: This module is where username and password is reset and updated in the database.

Provisioning:

1. Username
2. Password
3. Confirm Password

Functionality:

The function of this module is to update the login details of an administrator i.e the username and password.

Alerts:

1. Alerting when the password entered with the confirm password isn't the same.

Module 5:

Employee Registration: This module is where the admin registers the employee.

Provisioning:

1. Employee ID
2. Full Name of Employee
3. Date of Birth
4. Address
5. Nationality
6. Local Government if Nigerian
7. Email Address
8. Job Title
9. Level
10. Date Hired
11. Bank Branch
12. Account Number

Functionality:

The admin provides the details of the employees including the passport and save it to the database.

Alerts:

Alert if there are empty fields.

Module 6:

Search, Edit and Update Employees: The function of the module is for the administrator to search for an employee, edit the employee information and update the information.

Provisioning:

All that is needed in this module is the employee ID requested.

Functionality:

The basic functionality of this module is to search for the employee, edit employee information and update employee into the database.

Alerts:

Alert if employee's ID doesn't exist.

Module 7:

Payroll: This is the module where payroll is done. Employee is searched and the payment information is provided for payslip printing.

Provisioning:

All that is needed in this module is the employee ID and other payment information to be inputted.

Functionality:

The functionality of this module is basically for payroll management and payslip printing.

Alerts:

Alert if employee inputted is not in the database.

Module 8:

Reports: This module is where reports are being printed and processed.

Functionality:

The functionality of this module is basically for payroll report printing.

Alerts:

Alert if the administrator login is not the super administrator.

3.2.4 System Design

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. It could be seen as the application of systems theory to product development.

3.2.4.1 Data Flow Diagram

A data flow diagram (DFD) is a graphical representation of the “flow” of data through an information system, modelling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated.

First Level Data Flow Diagram for CALPAS System

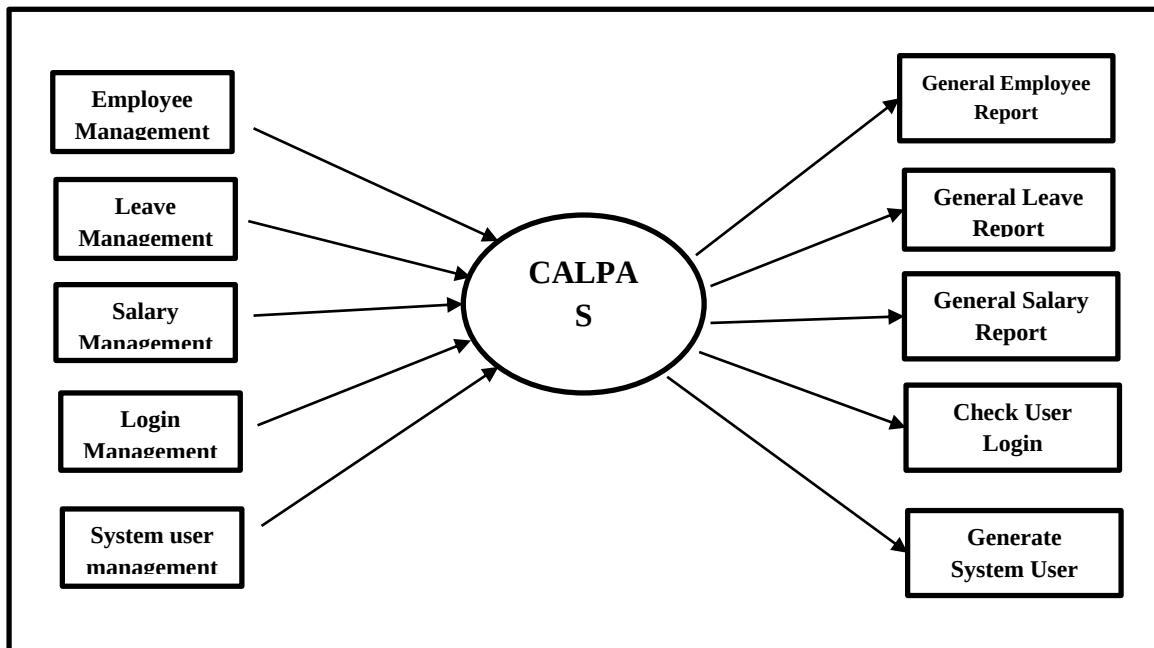


Figure 3.1: Dataflow Diagram of CALPAS

3.2.4.2 UML Diagram

UML is a way of visualizing a software program using a collection of diagrams. The notation has evolved from the work of Grady et-al and the Rational Software Corporation to be used for object-oriented design, but it has since been extended to cover a wider variety of software engineering projects. UML is accepted by the Object Management Group (OMG) as the standard for modelling software development. UML stands for Unified Modelling Language.

Types of UML Diagrams

These diagrams are organized into two distinct groups:

Structural UML Diagrams

- i. Class diagram

- ii. Package diagram
- iii. Object diagram
- iv. Component diagram
- v. Composite structure diagram
- vi. Deployment diagram

Behavioral UML Diagram

- i. Activity diagram
- ii. Sequence diagram
- iii. Use case diagram
- iv. State diagram
- v. Communication diagram
- vi. Interaction overview diagram
- vii. Timing diagram

Class Diagram

A class diagram is a type of structural diagram of UML that describes the structure of a system by showing the system of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects. Class diagrams are a type of structure diagram because they describe what must be present in the system being modelled. They are used to document software architecture. A class diagram is at the heart of UML. It represents the core purposes of UML because it separates the design elements from the coding of the system. UML was set up as a standardized model to describe an object-oriented programming approach. Since classes are the building block of objects, class diagrams are building blocks of UML. The diagramming components in a class diagram can represent the

classes that will actually be programmed, the main objects, or the interaction between class and object.

Basic of Class Diagrams

Class - Diagram Anatomy

The class diagram is composed of three parts:

- i. Upper Section (Name of the class) – This section is always required whether it is the classifier or an object.
- ii. Middle Section (Attributes of the class) – This describes the variables that describes the qualities of the class. This is required when describing the instance of a class.
- iii. Bottom Section (Class operations (methods)) – This is display in list format; each operation takes up its own line. It operations describe how a class can interact with data.

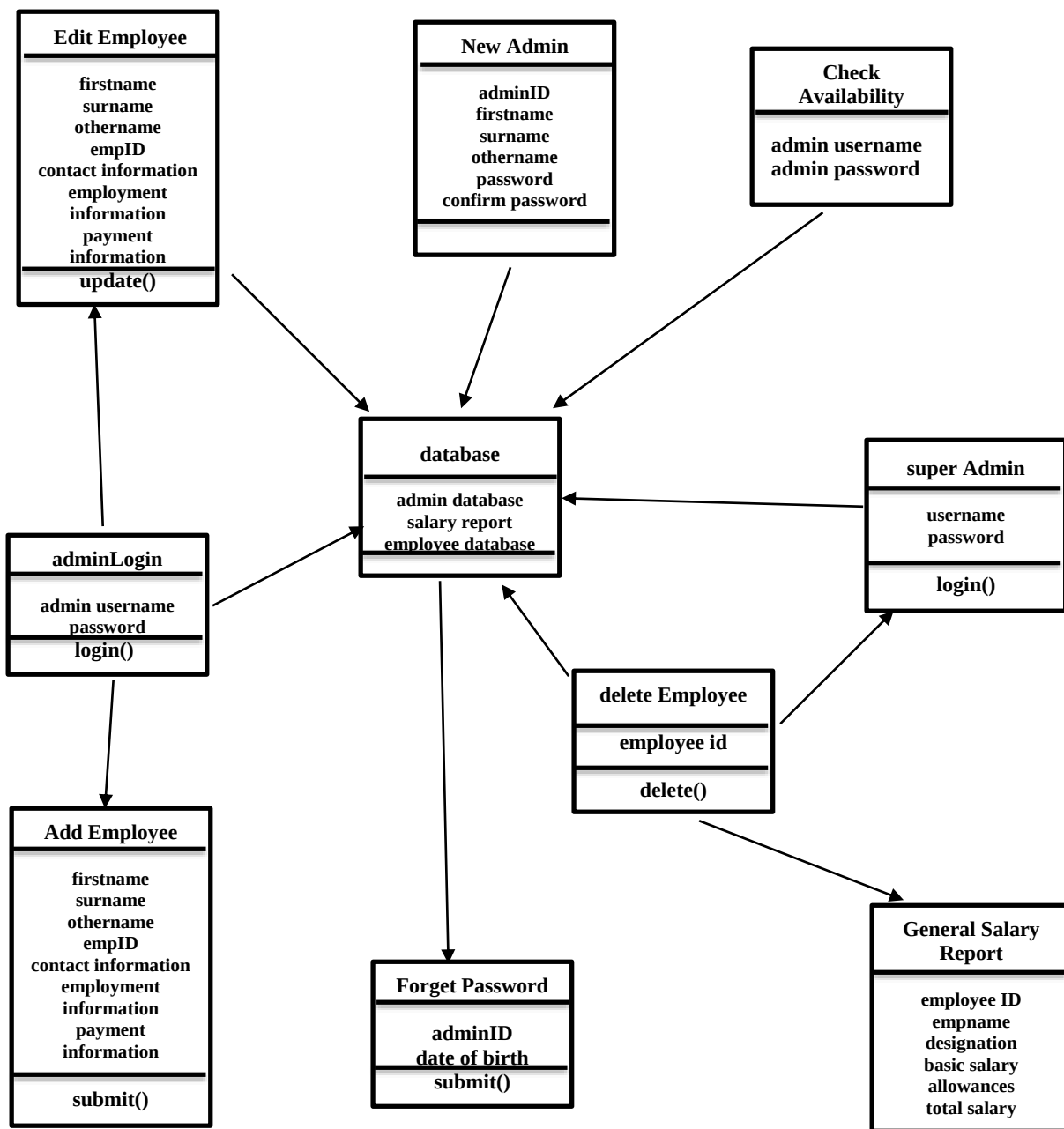


Figure 3.2: Class Diagram of CALPAS

Class Diagram Application

1. Illustrate data models for information systems.
2. Understand the general overview of an application's schematics.

3. Express the needs of a system and disseminate that information throughout the business.
4. Create detailed charts that focus on the programming code needed to implement the described structure.
5. It provides an implementation-independent description of types used in a system and passed between components.

Package Diagram

Package diagram is a structure diagram which shows packages and how they depend on each other. It shows different views of a system, for example, as multi-layered or multi-tiered application. Package, packageable element, dependency, element import, package import and package merge are nodes and edges drawn in a package diagram. Package is a namespace used in grouping elements that are semantically related together. Packageable element is a named element that are owned directly by a package. Examples of packageable elements are type, classifier, class, use case, component, package, constraint, dependency and event.

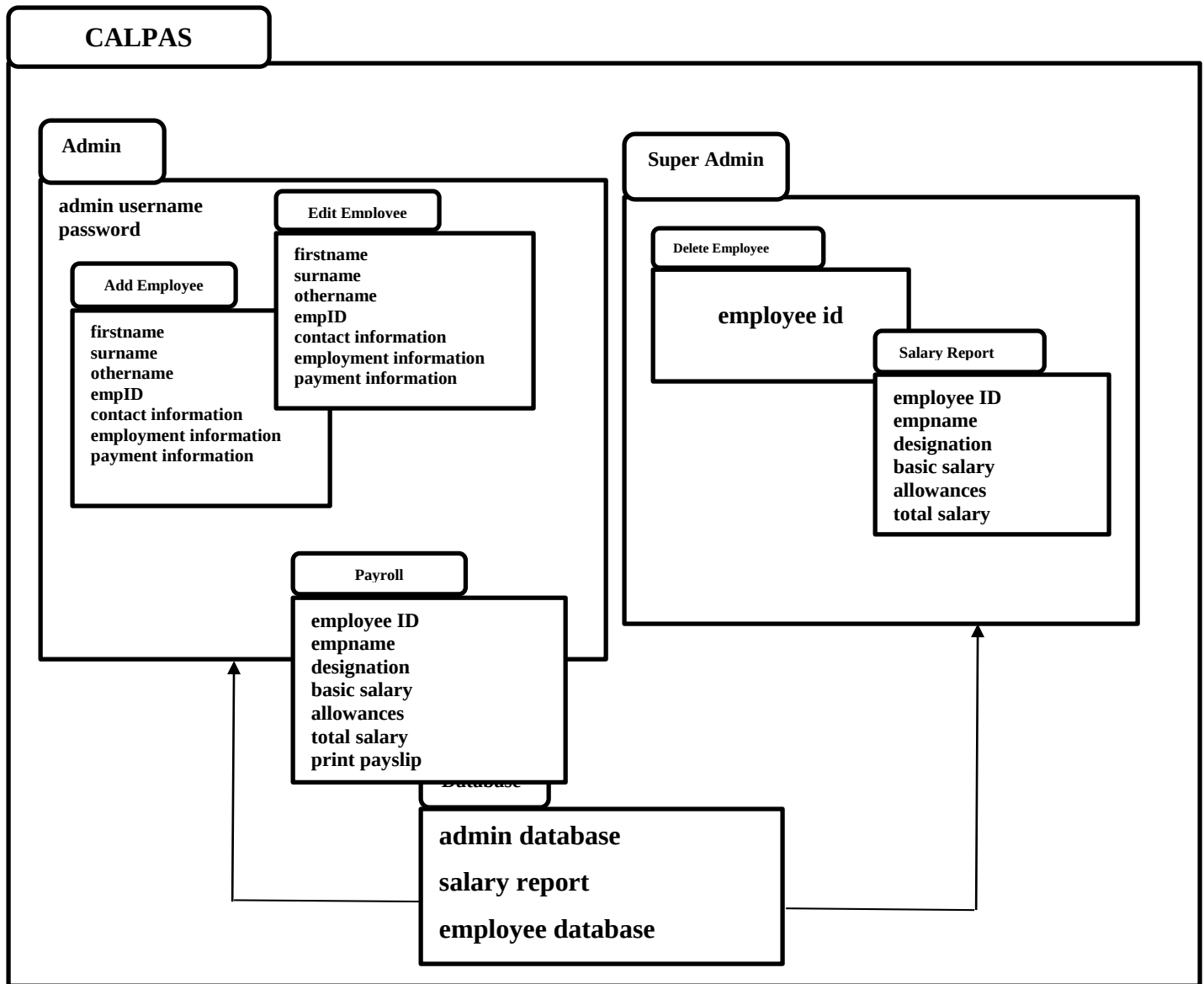


Figure 3.3: Packages Diagram for CALPAS

Use Case Diagram

Use case diagram discusses the functionality of the system using actors and use cases. The actor represents the role of the user of the system. An actor interacts with the system, but has

no control over the use cases. An actor can be someone or something that interacts with or uses the system, provides input to and receives information from the system etc.

The actors identified in CALPAS are:

1. Administrator
2. Super Administrator

A use case can be described as a specific way of using the system from a user's (actor's) perspective. Use cases provide a means of capturing system requirements, communicate with the end users, test the system etc. It is best discovered by examining the actors and defining what the actor will be able to do with the system.

The use cases that are identified in this system are:

- i. Admin Registration
- ii. Admin Login
- iii. Admin Forgotten Password
- iv. Employee Registration
- v. Employee Search, Edit and Update.
- vi. Salary Payroll
- vii. Leave Report
- viii. Salary Report

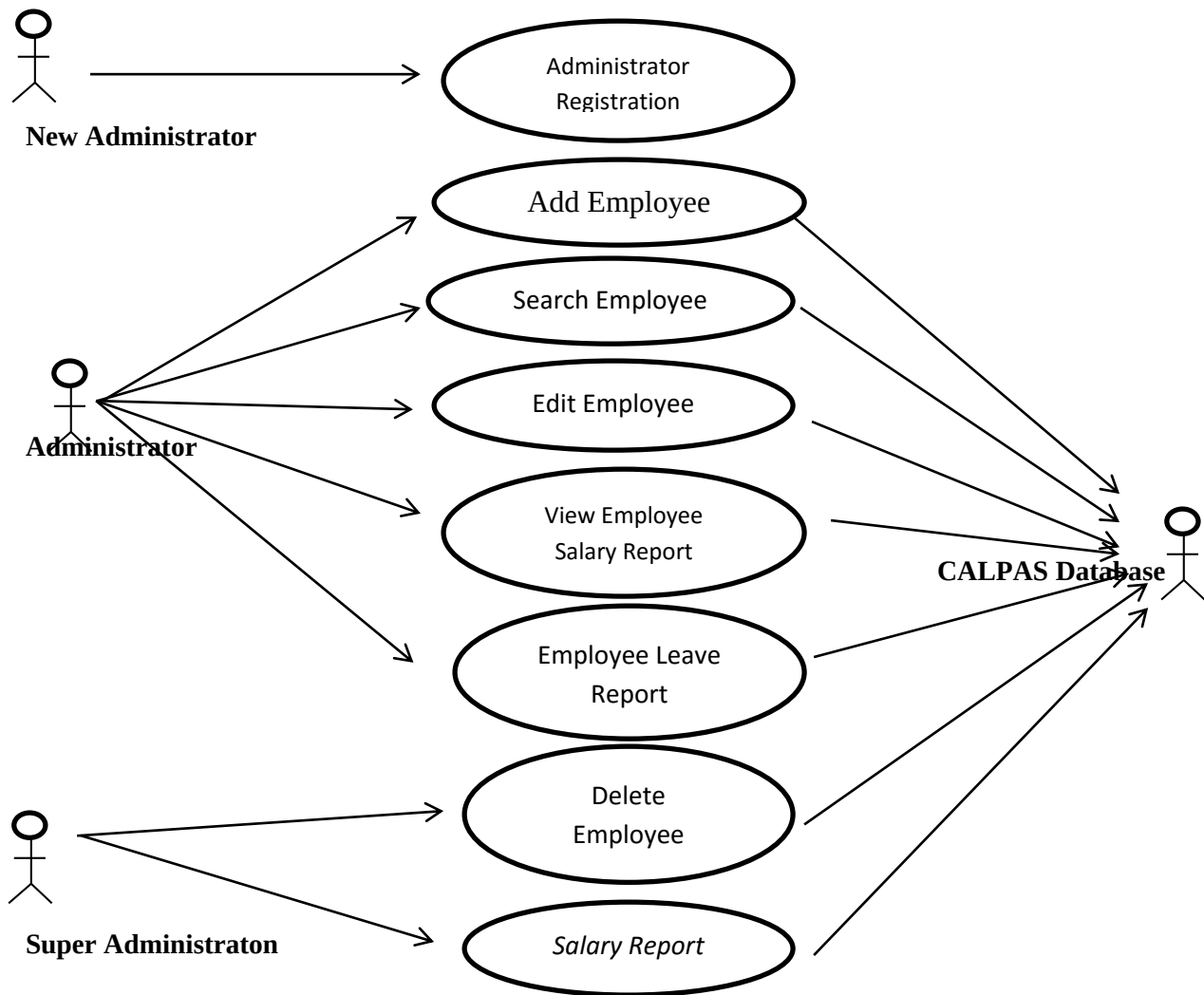


Fig 3.4: Use case diagrams for CALPAS

Activity Diagram

Activity diagrams provide a way of modelling the workflow of a business process, code-specific information such as a class operation. The transitions are implicitly triggered by completion of the actions in the source activities. The main difference between activity diagrams and state charts is activity diagrams are activity centric, while state charts are state centric.

It is used for modelling the sequence of activities in a process, whereas a state chart is better suited the model or discrete stages of an object's lifetime.

Actions on activities can occur at one of four times:

- **On Entry**- task performed when the object enters the state or activity.
- **On Exit**- Task performed when the object exits the state or activity
- **Do**- Task performed while in the state or activity with must continue until exiting the state.
- **On Event**- Task that triggers an action only if a specific event is received.

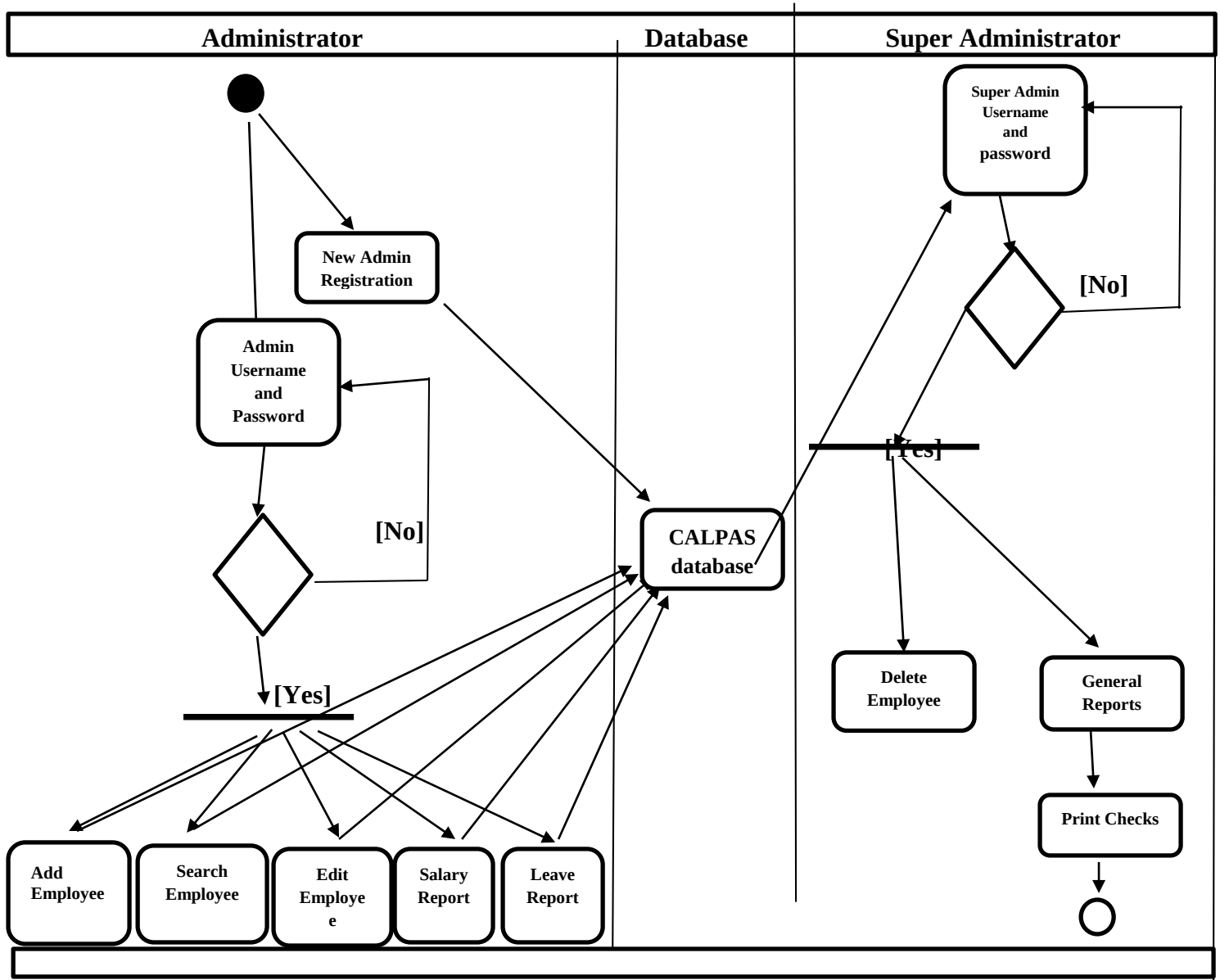


Fig 3.5: Activity Diagram for CALPAS

Sequence Diagram

Sequence diagram is a graphical view of a scenario that shows object interaction in a time based sequence; what happens first, what happens next it establish the roles of objects and help provide essential information to determine class responsibilities and interfaces. Sequence has two dimensions. They are vertical placement which represents time and horizontal placement which represents different objects.

Object:

An object has state, behaviour, and identity. Structure and behaviour of similar objects are defined in their common class. Each object indicates instance of a class.

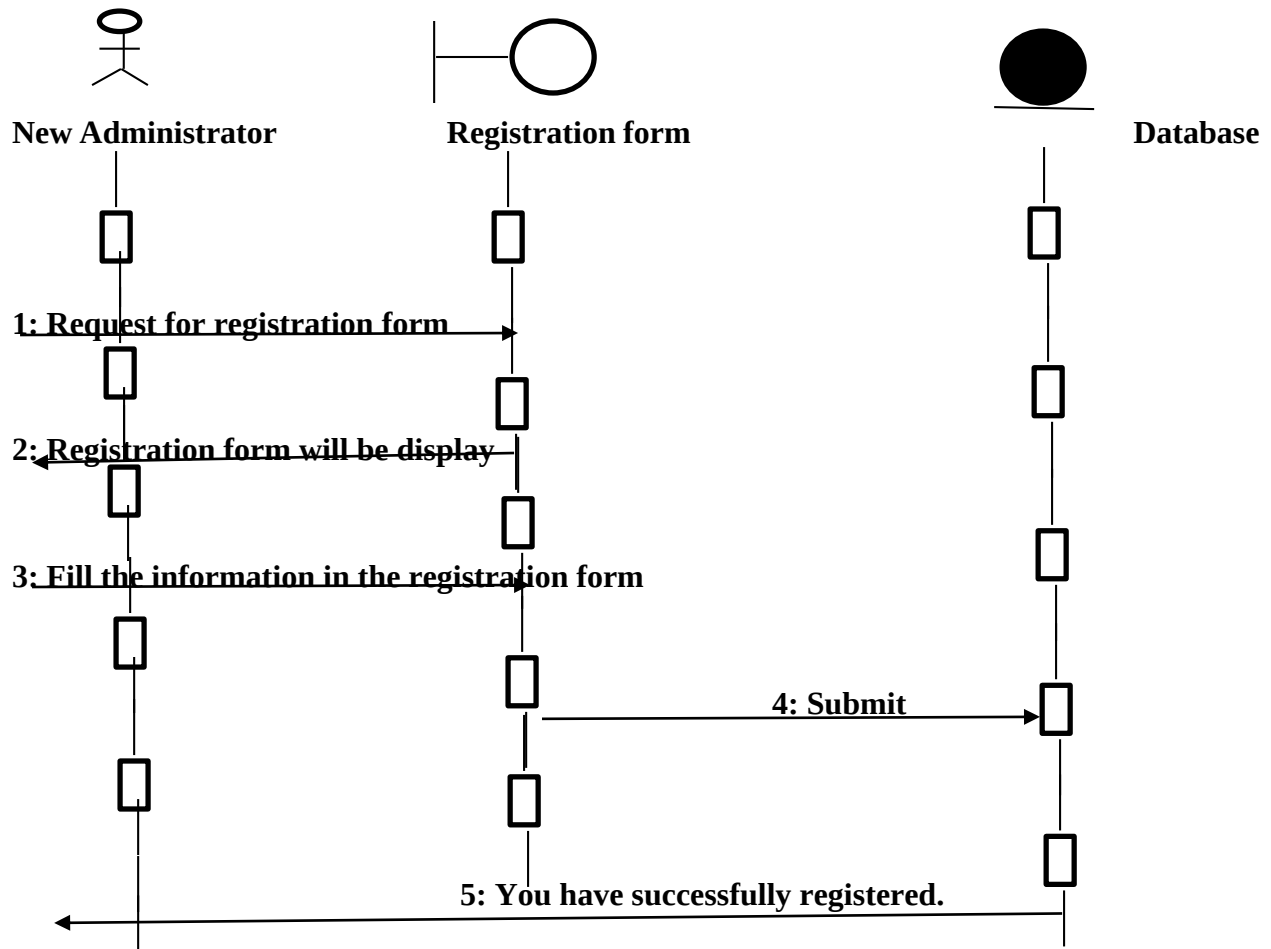
Message:

A message is the communication carried between two objects that trigger an event A message carries information from the source focus of control to the destination focus of control.

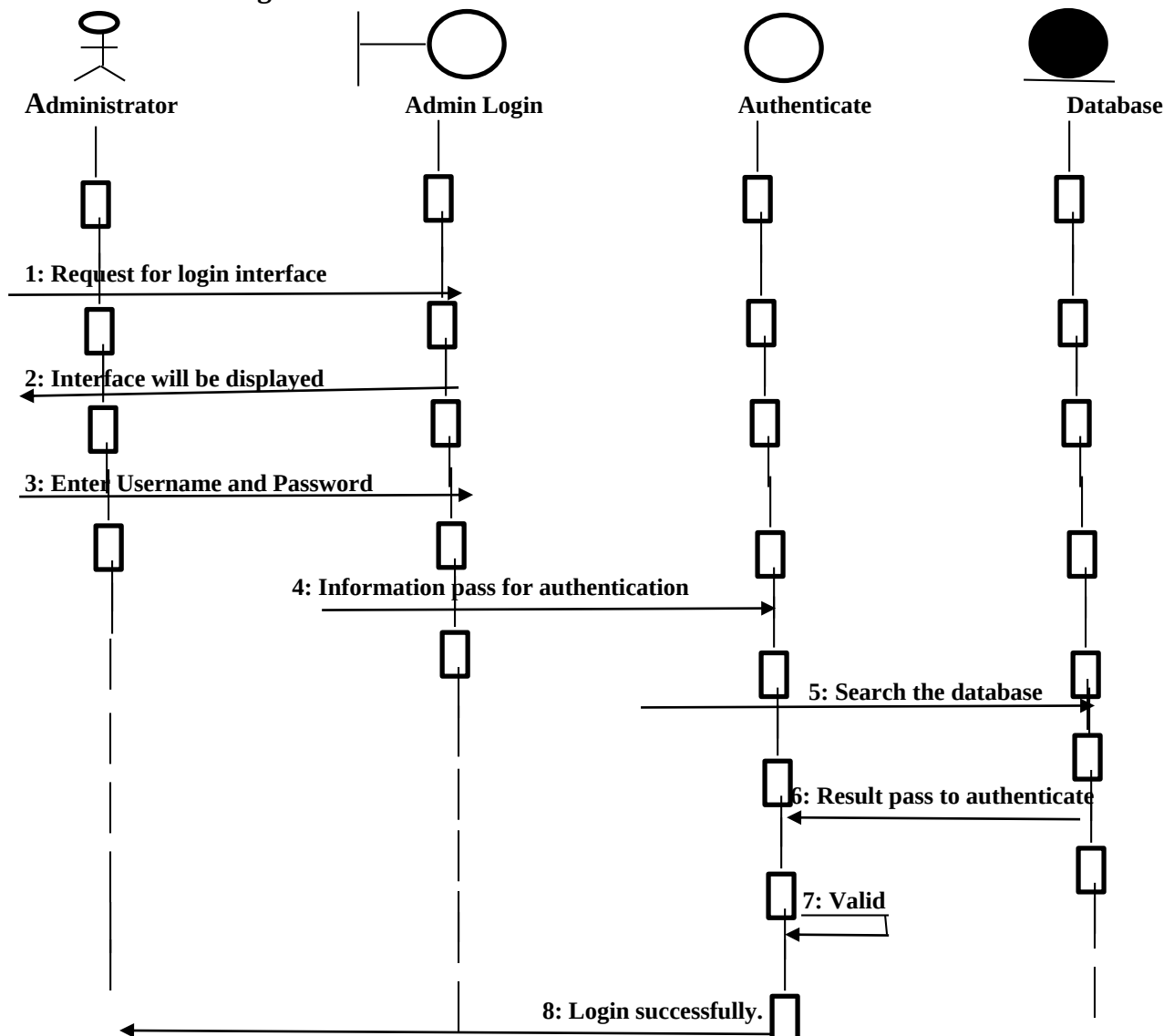
Link:

A link exists between two objects, including class utilities, only if there is a relationship between their corresponding classes. The existence of a relationship between two classes symbolizes a path of communication between instances of the classes: one object may send messages to another.

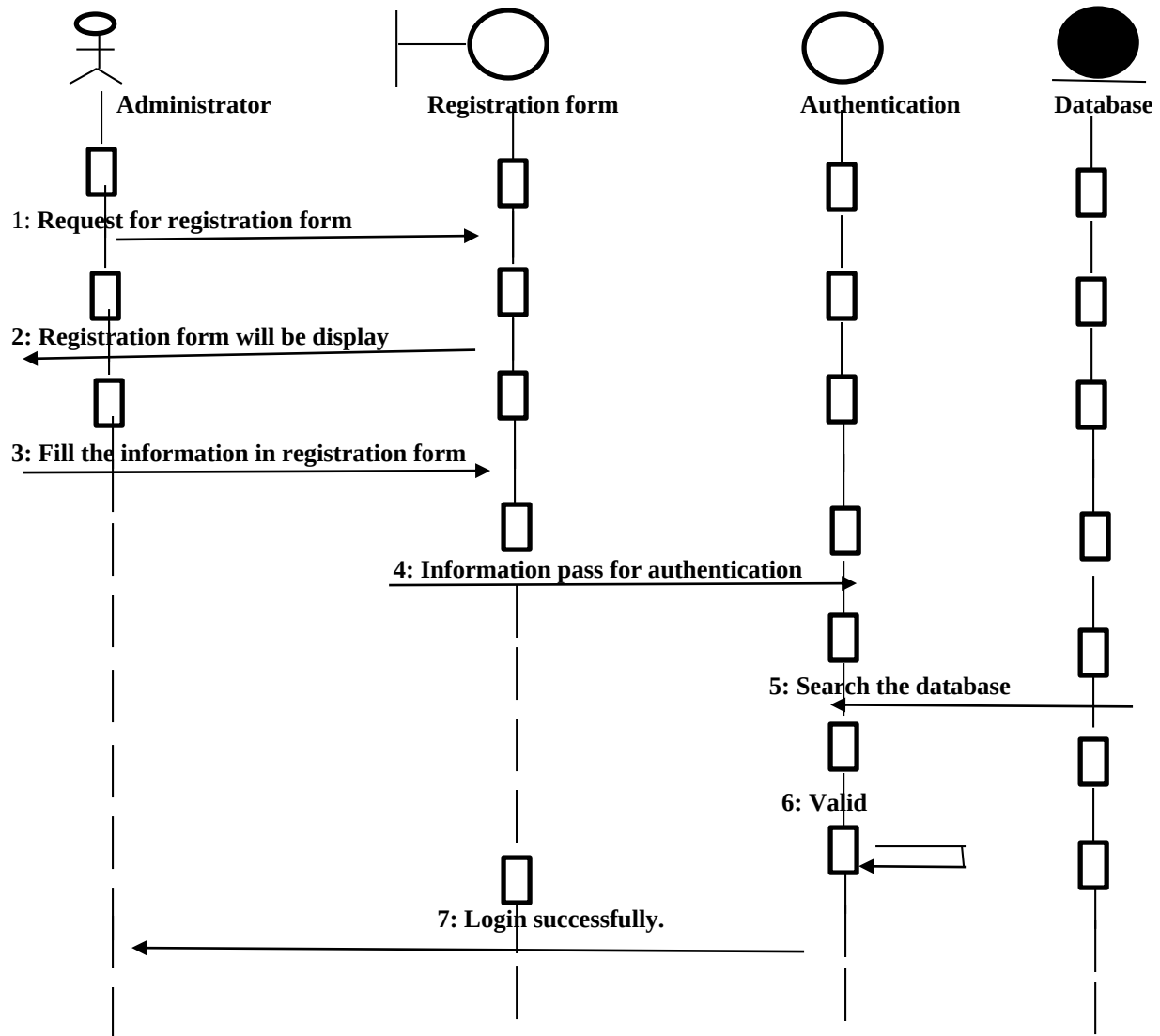
New Administrator:



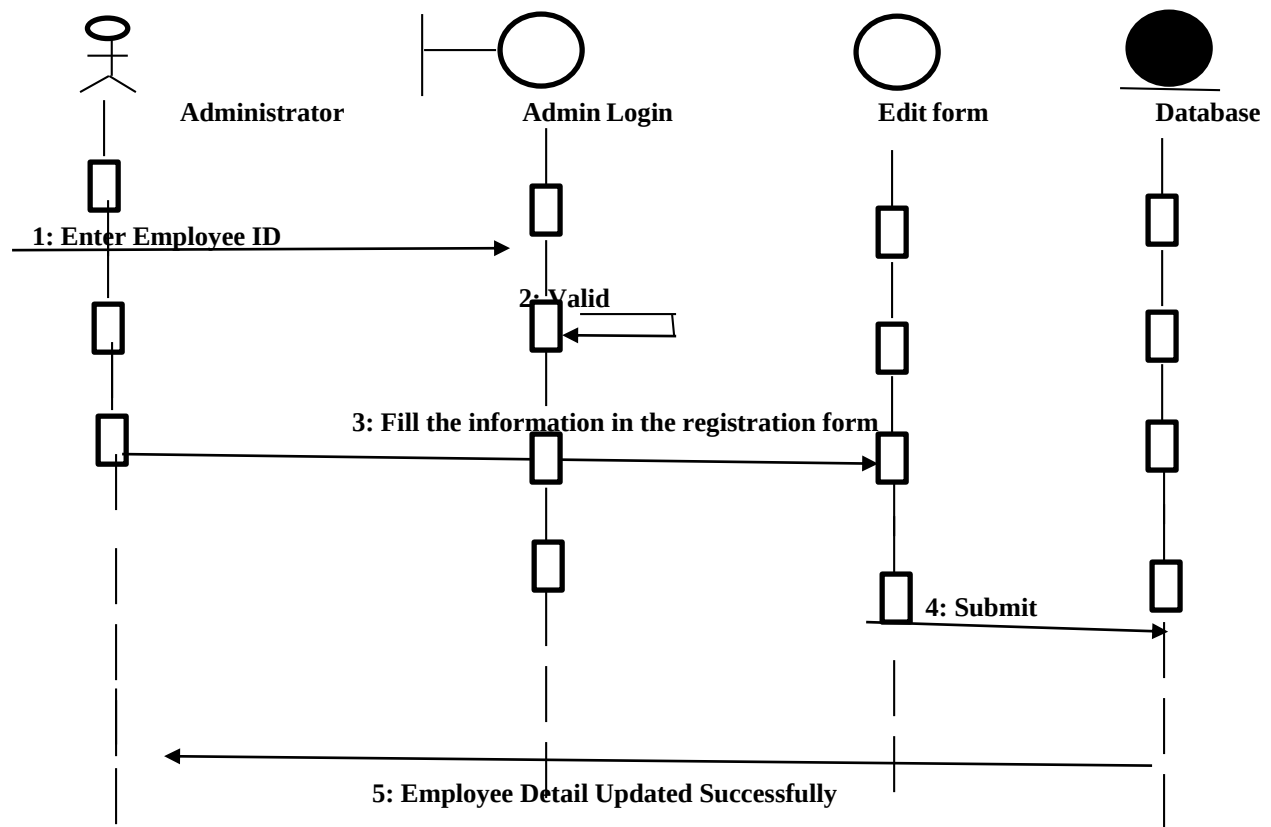
Administration Login:



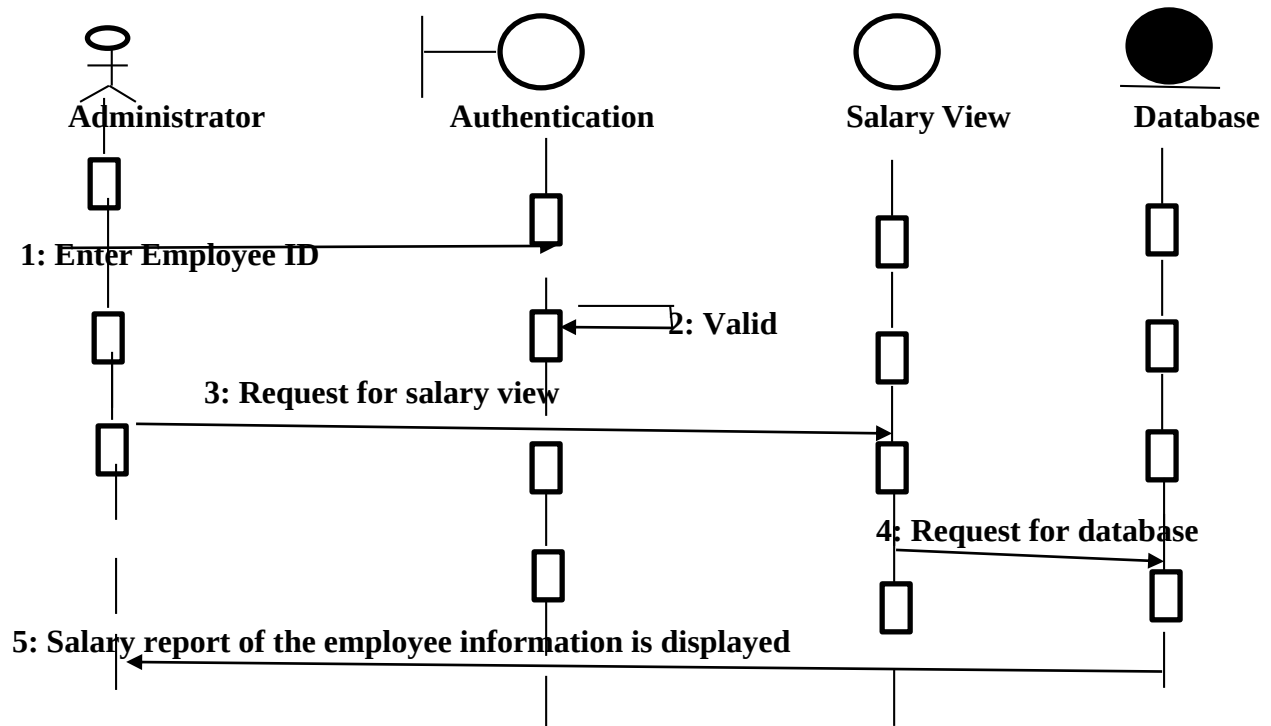
Add Employee



Search and Edit Employee:



View Salary Report:



General Salary Report

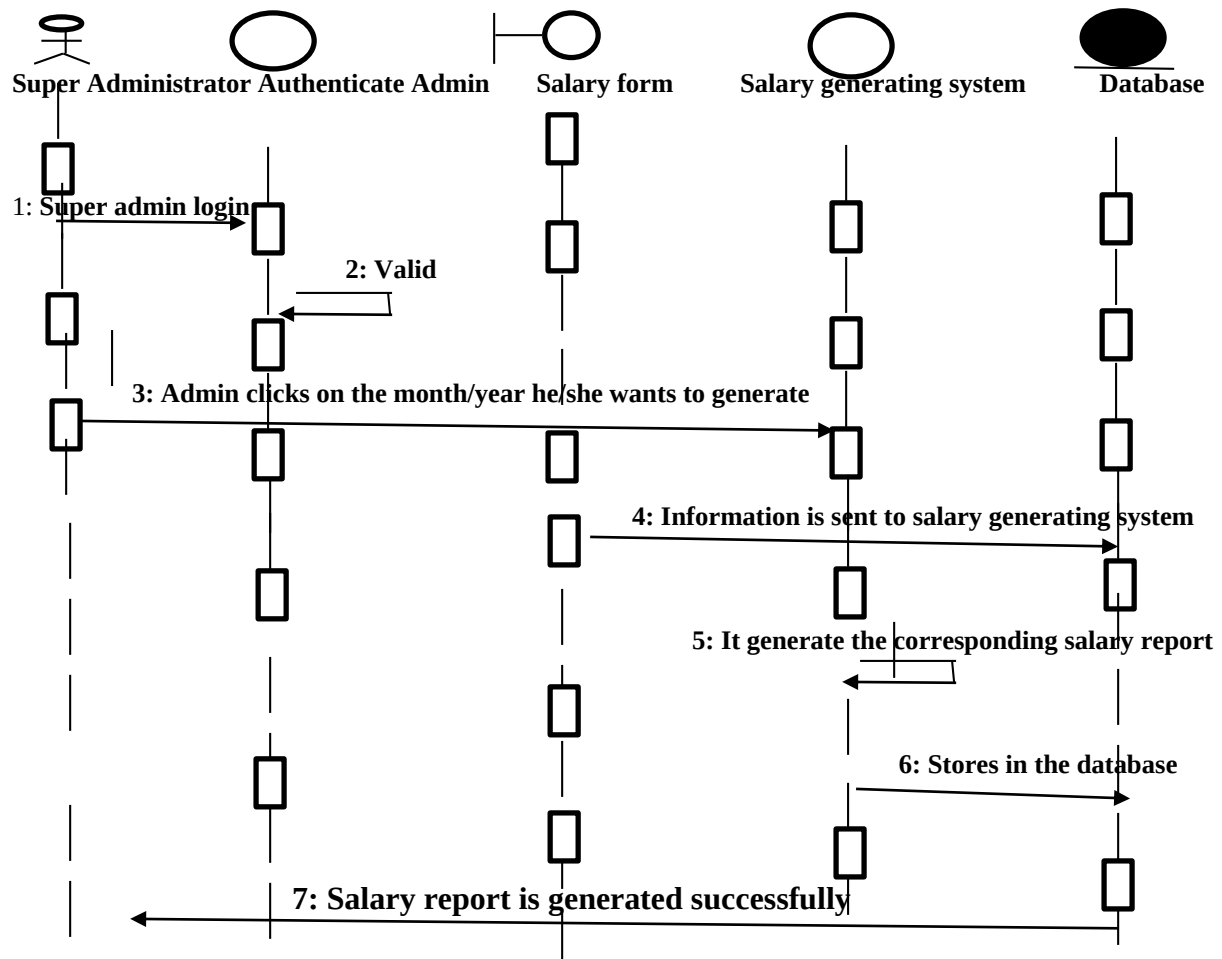


Figure 3.5: Sequence diagram for the modules of CALPAS

3.2.4.2 Table Structure for CALPAS

Figure 3.6 to figure 3.9 describe the tables used in CALPAS Database. It shows the table structure of the software and how they interact with each other.

Administrator Registration Table

Field(s)	Data Type	Emptiness	Size
Employee ID	VARCHAR	NOT NULL	30
First Name	VARCHAR	NULL	30
Last Name	VARCHAR	NULL	30
Other Name	VARCHAR	NULL	30
Date of Birth	VARCHAR	NULL	30
Confirm Date of Birth	VARCHAR	NULL	30
Nationality	VARCHAR	NULL	30
Job Title	VARCHAR	NULL	50
Username	VARCHAR	NULL	30
Password	VARCHAR	NULL	30
Confirm Password	VARCHAR	NULL	30
Registration Date	TIMESTAMP	CURRENT_TIMESTAMP	

Table 3.1: Admin Registration Table

Employee Registration Table

Field(s)	Data Type	Emptiness	Size
Employee ID	VARCHAR	NOT NULL	30
Title	VARCHAR	NULL	150
Surname	VARCHAR	NULL	30
First Name	VARCHAR	NULL	30
Other Name	VARCHAR	NULL	30

DOB	VARCHAR	NULL	150
Address	VARCHAR	NULL	50
Nationality	VARCHAR	NULL	150
If Foreigner	VARCHAR	NULL	30
State of Origin	VARCHAR	NULL	130
Local Government	VARCHAR	NULL	30
Telephone	VARCHAR	NULL	30
Email	VARCHAR	NULL	30
Category	VARCHAR	NULL	150
Job Title	VARCHAR	NULL	150
Grade 1	VARCHAR	NULL	150
Grade 2	VARCHAR	NULL	150
Grade 3	VARCHAR	NULL	30
Employee Type	VARCHAR	NULL	150
Date Hired	VARCHAR	NULL	150
Payment	VARCHAR	NULL	150
Account Number	VARCHAR	NULL	10
Account Type	VARCHAR	NULL	150
Bank	VARCHAR	NULL	150
Pensionable	VARCHAR	NULL	30
Not Pensionable	VARCHAR	NULL	30
PFA	VARCHAR	NULL	30
PIN	VARCHAR	NULL	30

Registration Date	TIMESTAMP	CURRENT_TIMESTAMP	
-------------------	-----------	-------------------	--

Table 3.2: Employee Registration Table

Leave Application Table

Field(s)	Data Type	Emptiness	Size
Employee ID	VARCHAR	NULL	30
Title	VARCHAR	NULL	30
Surname	VARCHAR	NULL	30
First Name	VARCHAR	NULL	30
Other Name	VARCHAR	NULL	30
Leave Type	VARCHAR	NULL	30
Begins	VARCHAR	NULL	30
Ends	VARCHAR	NULL	30
Description	TEXT	NULL	
Approve Date	TIMESTAMP	CURRENT_TIMESTAMP	

Table 3.3: Leave Application Table

Payroll Table

Field(s)	Data Type	Emptiness	Size
Employee ID	VARCHAR	NOT NULL	30
Surname	VARCHAR	NULL	30
First Name	VARCHAR	NULL	30
Designation	VARCHAR	NULL	30

Date of Birth	VARCHAR	NULL	30
Grade	VARCHAR	NULL	30
Basic Salary	VARCHAR	NULL	30
Peculiar Allowance	VARCHAR	NULL	30
Rent	VARCHAR	NULL	30
Other Allowance	VARCHAR	NULL	30
Fuelling of Car and Recharge Card	VARCHAR	NULL	30
Arrears/Underpayment	VARCHAR	NULL	30
Overpayment	VARCHAR	NULL	30
Loan/Advances	VARCHAR	NULL	30
Paye	VARCHAR	NULL	30
Pension	VARCHAR	NULL	30
Other Deductions	VARCHAR	NULL	30
Cooperative	VARCHAR	NULL	30
PayDate	TIMESTAMP	CURRENT_TIMESTAMP	

Table 3.4: Payroll Table

3.2.5 Implementation/Development

CALPAS is designed, developed and implemented using Java Swing technology and MySQL server for the software component. The following describes the development tools, both the software and hardware for CALPAS.

3.2.5.1 JAVA SWING

Java Swing is the new GUI Toolkit, which partly replaces the Java Awt toolkit because Swing uses some fundamental mechanisms of the AWT, like the event handling. Basically, Swing uses the platform-independent AWT features, and the AWT top level windows to get access to the native GUI system. Java Swing is a part of Java Foundation Classes (JFC) that is used to create window-based applications. JFC are set of GUI components which simplify the development of desktop applications. Java Swing is built on the top of AWT (Abstract Windowing Toolkit) API and entirely written in java. Unlike AWT, Java Swing provides platform-independent and lightweight components. The javax.swing package provides classes for java swing API such as JButton, JTextField, JTextArea, JRadioButton, JCheckbox etc. CALPAS front-end was created with the Java Swing.

Commonly used Methods of Components class

The methods of component class are widely used in java swing that are given below

1. **public void add (Component c)-** This is to add a component on another component in the GUI created.
2. **public void setSize (int width, int height) -** This sets size of the component to the assigned width and height.
3. **public void setLayout (LayoutManager m)-** this sets the layout manager for the component.
4. **public void setVisible (Boolean b) –** This set the visibility of the component. It is by default false but can be change to true if you want to set it visible.

FEATURES OF JAVA SWING

Java swing provides many new features; two of its popular features are:

1. Frivolous components
2. Pluggable look and feel

1. **Frivolous Components:** Swing components are frivolous as they are written in java entirely and do not depend on native peers because it is platform specific code resources. Rather it uses simple drawing primitives to render them on the screen. It is not controlled by the underlying operating system especially the look and feel but it is controlled by Swing itself. Also, note that most of the components are lightweight but not all.
2. **Pluggable Look and Feel:** The look and feel of the any application and applets built with Java Swing feature is a pluggable look and feel. It can even switch to different look and feel at runtime. It has the capability to support several look and feels, but at present, it provides support for the Windows and Motif.

3.2.5.2 MySQL

MySQL is a freely available open source Relational Database Management System (RDBMS) that uses Structured Query Language (SQL). SQL is the most popular language for adding, accessing and managing content in a database. It is noted for quick processing, reliability, flexibility and ease. The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB,

now owned by Oracle Corporation. Most free-software-open source projects that require a full-featured database management system often use MySQL.

Hardware Components Needed

- i. **Standard Monitor-** This would be used to display all the data, forms, tables etc.
- ii. **Standard (ergonomic) Keyboard-** A keyboard is needed to key in all necessary data into the system.
- iii. **Memory-** A large memory would be needed since a large storage space is needed to store all the data yearly.
- iv. **CPU-** This will give out all the control signals to all components of the computer.
- v. **ALU-** This is where all the arithmetic and logical comparisons take place.
- vi. **RAM-** The random access memory stores all the temporary program files.
- vii. **ROM-** The read only memory stores permanently files and program.
- viii. **Mouse-** This is needed in order to select and move the cursor.
- ix. **Printer-** This would be needed to print reports and other documents.
- x. **CD ROMS-** This is used to store reports of previous months and years.

3.2.6 Testing

3.2.6.1 Unit Testing

A unit is the smallest part of a software that is tested by a user. Its input is always one or a few and is usually a single output. It is a level of testing where individual units/modules are tested in a software. The purpose of this is to validate that each and every unit/module works well or perform the tasks it is design for. Users will test each unit/modules in CALPAS carefully to ensure it meets the system requirement and specifications.

3.2.6.2 System Testing

According to International Software Testing Qualification Board (ISTQB), system testing is the process of testing an integrated system to verify that it meets specified. It is a level of software testing where complete and integrated software is tested. This is to evaluate compliance with specified requirements. Upon completion, CALPAS would be tested thoroughly.

3.2.7 Maintenance

Maintenance of the system is necessary for longevity of the system. The following 'is useful for the maintenance of CALPAS and should be strictly adhering to.

i. Monthly Maintenance Needs

At least once in a month, maintenance should be done on CALPAS system to fix bugs in coding and configuration and to apply vendor maintenance upgrades. These corrections and upgrades may be done by an in-house or by the developer depending on the contract and organizational structure. As changes are done to the system, employees and managers may need to be trained to adapt to the changes, so training cost and additional staffing needs should be factored in as well.

ii. Quarterly Maintenance Needs

About once every three months the system's security access should be reviewed and fine-tuned by either an in-house or third party professionals. Reviews should be conducted by managers to make sure that the system is still meeting up with the objectives and changes should be made if necessary. About three months after CALPAS implementation, subsequent installation phases should be put into action if needed, but should be repeated.

iii. Biannual Maintenance Needs

At least once in every six months, out-dated reports, functions, and features should be purged from the software to keep information and dashboards well organized and running smoothly. Review for possible upgrades to keep up with technology trends. Theses should be done by an employee that is tech savvy and familiar with both the system in use and the organizational needs.

iv. Annual Maintenance Needs

At least once a year, the system should be reviewed to make sure that all relevant compliance needs are being taken care of. As new legislation is put into effect, these needs may be required in addition to annual maintenance.

3.3 CALPAS Assessment/Evaluation

CALPAS would be using questionnaire as way of analyzing, assessing and evaluating its design. A questionnaire would be created, printed and distributed around the university to be filled. Users will be picked randomly from different departments and field of study to assess, test and work on the software for minutes till they are satisfying and then they will, be handed the questionnaire to fill. After filling the questionnaire, it would be collected and analyzed. The questionnaire should test the efficiency, friendliness, security, flexibility and user interface design of CALPAS.

3.3.1 Questionnaire

QUESTIONNAIRE

USABILITY TESTING OF CALPAS (CALEB UNIVERSITY PAYROLL SYSTEM)

Questionnaire survey has been widely utilized by scholars in many research fields. In a large-scale survey, researchers collect huge amount of information through questionnaires. Its input becomes an important step in analytical process. For convenience and simplicity, most questionnaires are design to have checkboxes so that respondents can answer according to the printed questions.

This questionnaire is to test the flexibility, efficiency, friendliness, security, user interface design, mobile compliance and capability maturity model of the developed **CALPAS (Caleb University Payroll System)**.

Please fill out this questionnaire after testing the software thoroughly to your satisfaction. If you don't have time to stay and complete it, please take it with you, complete it, and return it by tomorrow.

Filling in Your Answers

Please don't write over the check boxes or crosshair (+) symbols.

- **Please keep your marks within the check boxes. Any mark will do:**
- **You should use a pen with dark blue or black ink.**

Respondent Identification

(Please specify)

E-mail: _____

Date: _____

Sex: ☐ MALE ☐ FEMALE

Department: _____

1. Have you used any software payroll before?

YES ☐ NO ☐

2. If yes, can you provide the name of the software? _____.

3. Have you tested **CALPAS (Caleb University Payroll System)**?

YES ☐ NO ☐

S/N	QUESTIONS	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
4.	Do you feel that you successfully completed tested all the modules in the software?					
5.	In relation to other software I have used, I found the CALPAS prototype to be accurate:					

6.	In relation to the Viewer, I found the Application prototype to be timeless:					
7.	The menu items were well organized and functions were easy to find.					
8.	I immediately understood the function of each menu item.					
9.	All of the functions I expected to find in the modules were present.					
10.	The buttons were well organized and easy to find.					
11.	I immediately understood the function of each button.					
12.	All of the functions I expected to find on the button bar were present.					
13.	I found navigating around the Application screen to be very difficult					

14.	My overall impression of CALPAS prototype is very positive.					
15.	How flexible is the software to you?					
16.	Is the software efficient enough?					
17.	Is the software friendly for enough for a user?					
18.	Is the software secured enough?					
19.	Is the user interface design beautiful enough?					

20. Can you recommend the software to a company without doubting?

YES ☐ NO ☐

COMMENTS

Thank you for generously volunteering your time to participate in this usability testing. Your input will be **valuable** in the development of the CALPAS. We hope that you found it to be an interesting and enjoyable experience!

CHAPTER FOUR

IMPLEMENTATION AND RESULTS

4.1 System Requirement Specification

4.1.1 Hardware Requirement Specifications:

- PC with 300 megahertz.
- 128 megabytes (MB) of RAM or higher recommended
- 1.5 gigabytes (GB) of available hard disk space.
- Super VGA (800 X 600) or higher-resolution video adapter and monitor
- CD-ROM or DVD drive.
- Keyboard and Microsoft Mouse or compatible pointing device.

4.1.2 Software Requirement Specifications:

- | | |
|--------------------|----------------------------------|
| • Operating System | Window XP/7/8.1/10 |
| • Web Browser | Internet Explorer/Firefox/Chrome |
| • Languages | JAVA |
| • Technologies | JAVA SWING |
| • Database | MySQL |

4.2 Implementation

4.2.1 Homepage:

This is the starting page of the payroll system. It has the login button, registration button and forgot password link. It also has a link to the school official website.

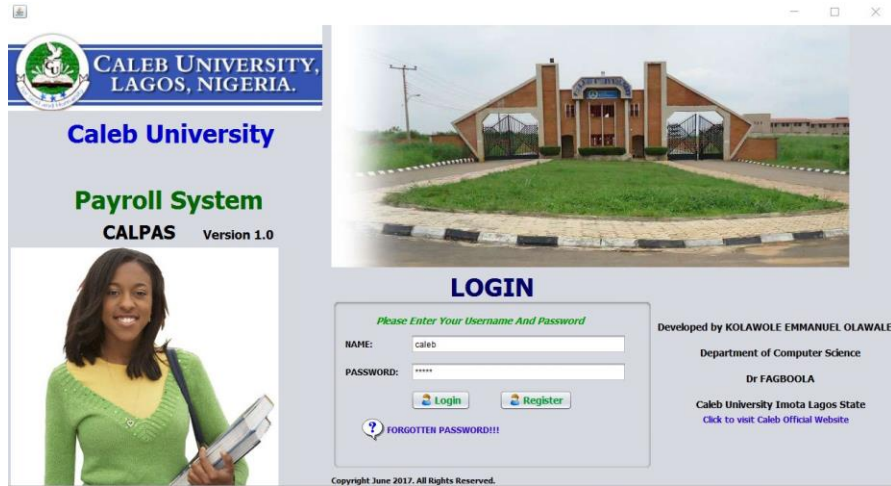


Figure 4.1: Screenshot of CALPAS Homepage

4.2.2 Admin Registration

This is the module where an administrator registers before logging in to the system.

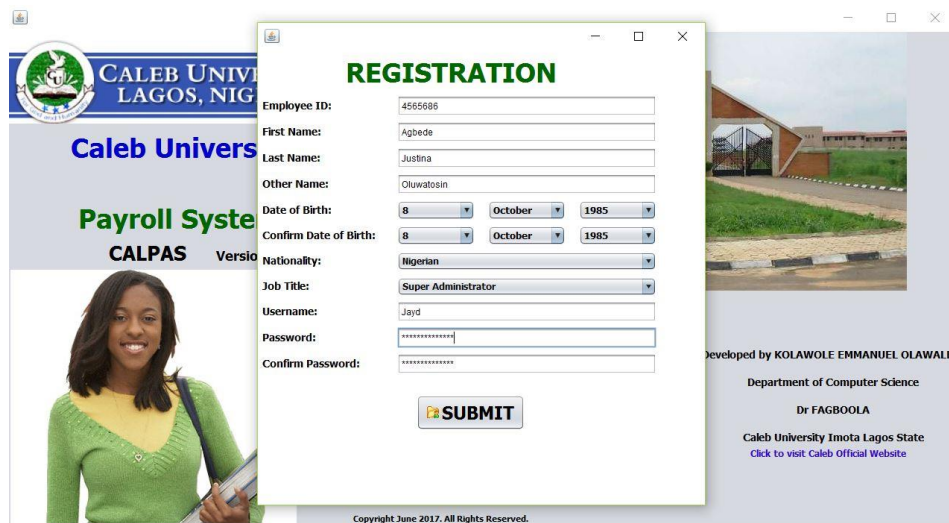


Figure 4.2: Screenshot of CALPAS Admin Registration

4.2.3 Forget Password

This is the module where an administrator is able to retrieve username and password if forgotten at any point in time.

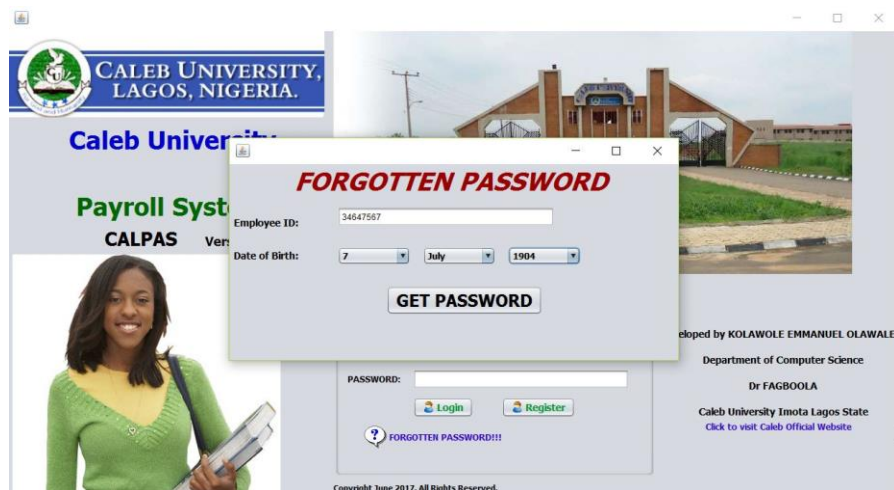


Figure 4.3: Screenshot of CALPAS Forgotten Password Page

4.2.4 Dashboard:

This is where all other modules of the system are located. It contains a menu bar, calculator and a link to the developer official website.

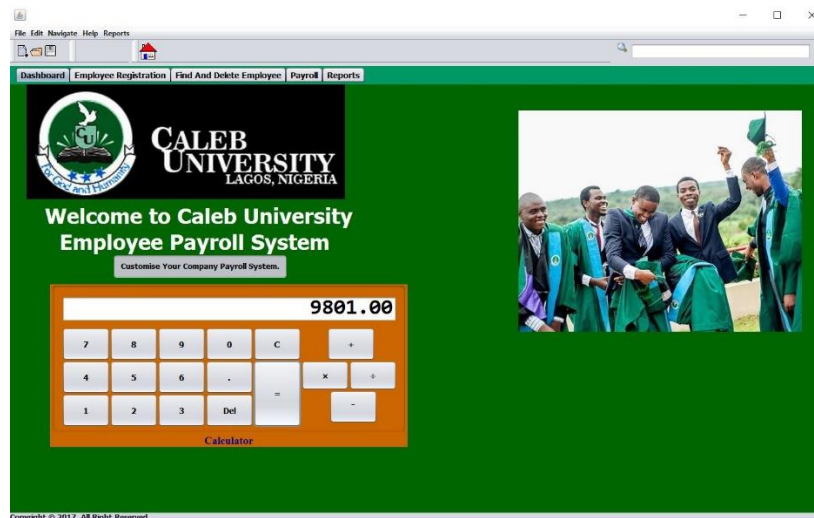
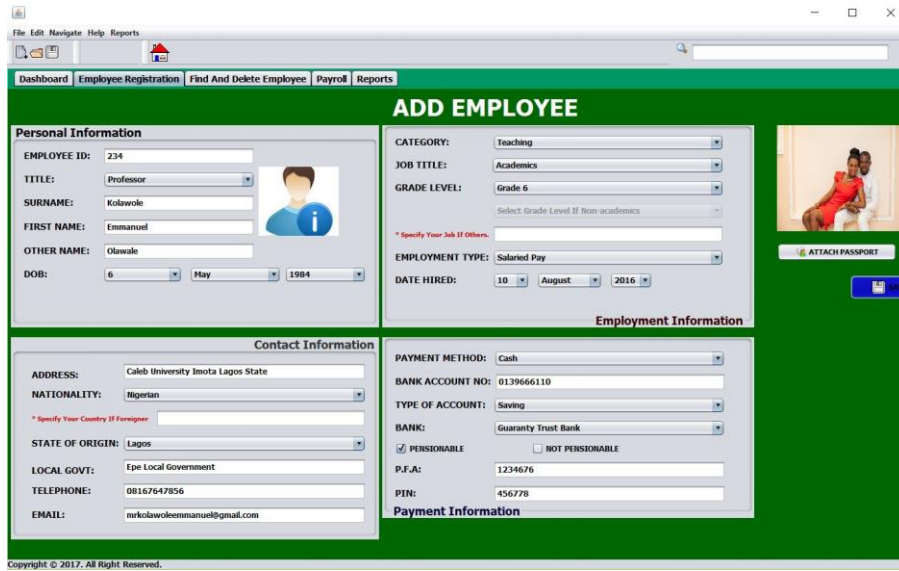


Fig 4.4: Screenshot of CALPAS Dashboard

4.2.5 New Employee Registration:

This is where an admin registers a new employee.

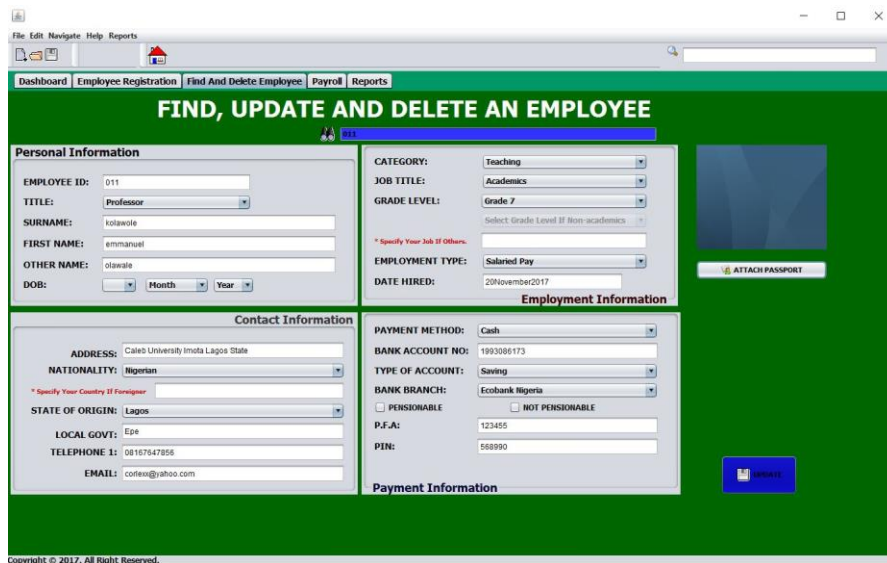


The screenshot shows the 'ADD EMPLOYEE' form in the CALPAS system. The form is divided into several sections: Personal Information, Contact Information, Employment Information, and Payment Information. The Personal Information section includes fields for Employee ID (234), Title (Professor), Surname (Kolawole), First Name (Emmanuel), Other Name (Olawale), and Date of Birth (6 May 1984). The Contact Information section includes Address (Caleb University Imota Lagos State), Nationality (Nigerian), State of Origin (Lagos), Local Govt (Epe Local Government), Telephone (08167647856), and Email (mrkolawoleemmanuel@gmail.com). The Employment Information section includes Category (Teaching), Job Title (Academics), Grade Level (Grade 6), Employment Type (Salaried Pay), and Date Hired (10 August 2016). The Payment Information section includes Payment Method (Cash), Bank Account No (0139666110), Type of Account (Saving), Bank (Guaranty Trust Bank), P.F.A (1234676), and PIN (456778). There is also a section for 'Specify Your Job If Others' and a button to 'ATTACH PASSPORT'.

Fig 4.5: Screenshot of CALPAS Employee Registration Page

4.2.6 Update Employee

This is the module where an employee is search and updates his or her information.



The screenshot shows the 'FIND, UPDATE AND DELETE AN EMPLOYEE' form in the CALPAS system. The form is divided into several sections: Personal Information, Contact Information, Employment Information, and Payment Information. The Personal Information section includes fields for Employee ID (011), Title (Professor), Surname (kolawole), First Name (emmanuel), Other Name (olawale), and Date of Birth (Month Year). The Contact Information section includes Address (Caleb University Imota Lagos State), Nationality (Nigerian), State of Origin (Lagos), Local Govt (Epe), Telephone (08167647856), and Email (corfex@yahoo.com). The Employment Information section includes Category (Teaching), Job Title (Academics), Grade Level (Grade 7), Employment Type (Salaried Pay), and Date Hired (20 November 2017). The Payment Information section includes Payment Method (Cash), Bank Account No (1993086173), Type of Account (Saving), Bank Branch (Ecobank Nigeria), P.F.A (123455), and PIN (568990). There is also a section for 'Specify Your Job If Others' and a button to 'ATTACH PASSPORT'.

Fig 4.6: Screenshot of CALPAS Employee Update Page

4.2.7 Payroll

This is where payroll is being carried out.

The screenshot displays the 'Payroll' tab in the CALPAS system. It shows a form for updating payroll information for Employee ID 011. The form includes fields for various allowances and deductions, a 'SAVE' button, and a summary of the net pay.

EMPLOYEE ID: 011

Employee ID: 011
Basic Salary: 13445
Peculiar Allowance: 334
Rent: 444
Other Allowance: 4456
Fuelling Of Official Car And Recharge Cards: 67899
Arrears/Underpayment: 88900

Overpayment: 546789
Loan/Advances: 435678
PAYE: 6789
Pension: 23776
Rent: 3366778
Other Deductions: 45667787
Cooperative: 34556

TOTAL DEDUCTIONS: [Empty field]
NET PAY: **0.00**

SAVE

Empo...	Surname	First Na...	Design...	Date of ...	Grade L...	Basic S...	Peculiar...	Rent	Other Al...	Fuelling	Arrears/...	Overpay...	LoanId...	PAYE	Pension	Rent	Other D...	Net Pay	Total De...	Net Pay
011	kolawole	emma...	Profes...	28Sep...	Grade 7	13445	334													
011	kolawole	emma...	Profes...	28Sep...	Grade 7	13445	334	444												
011	kolawole	emma...	Profes...	28Sep...	Grade 7	13445	334	444	4456											
011	kolawole	emma...	Profes...	28Sep...	Grade 7	13445	334	444	4456	67899										
011	kolawole	emma...	Profes...	28Sep...	Grade 7	13445	334	444	4456	67899	88900	546789	435678	6789	23776	33667...	45667...	34556		

Copyright © 2017. All Right Reserved.

Fig 4.7: Screenshot of CALPAS Employee Payroll Page

4.2.8 Reports

This is where reports are printed at any point in time.

4.3 USER ASSESSEMENT ANALYSIS

Tabel 4.1 below gives the result of the survey carried out to test the functionality, user friendliness and so on using the questionnaire in 3.2.7. Number 1 represent strongly disagreed, number 2 represent disagreed, number 4 represent agreed and number 5 represents strongly agreed. The survey was carried out in Caleb University with 10 people which includes final year students, the HR manager and lecturers.

Response Questions	1	2	3	4	5
4	-	-	-	-	10
5	-	-	-	3	7
6	-	-	2	3	5
7	-	1	1	4	4
8	-	-	-	2	8
9	-	-	-	-	10
10	-	-	-	1	9
11	-	-	-	-	10
12		2	2	1	5
13	9	-	-	-	1
14	-	-	-	-	10
15	-	-	-	-	10
16	-	-	-	-	10
17	-	-	-	2	8
18	-	-	-	4	6
19	-	-	-	-	10
20	-	-	-	-	10

4.3.1 Adequacy and Coverage

The payroll software can perform the following functions

- i. Store employee information
- ii. Calculate gross and net pay, and determine tax deductions to be made

- iii. Print payslips.
- iv. Provide the user with adequate help by the user manual with the software

4.3.2 Efficiency and Effectiveness

- i. The payroll software calculates the total earnings of the employee and automatically.
- ii. The software does its calculations in a very clear and concise manner. All calculations are guaranteed accuracy.

4.3.3 Productiveness

- i. The payroll software gives the employee the ability to keep track of their earnings by printing their payslip for each month.
- ii. It gives the employers the ability to keep records of how much they pay out as salaries.

4.3.4 Elegance and User-friendliness

- i. The different tasks and functions are outlined in a very simple and clear manner for the users.
- ii. The help file can be used by the users to know how to use the payroll software.
- iii. The interface is very simple and not complicated to allow for easy usage.

4.3.5 Quality assurance

- i. The software was fully tested to ensure it is relatively error free and that it computes results correctly, including deductions such as taxes.
- ii. The software provides a better, time saving and efficient way of keeping track of employees' earnings by speeding up calculations, and reducing paperwork by keeping efficient electronic records.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Introduction

This chapter states the summary of the entire project and also makes recommendation(s) as to what ways and areas that it can be further developed. It also gives the conclusion on the entire design effort that the project represents.

5.2 Conclusion

The main aim of carrying out the project is to design, develop and implement a comprehensive and expansible payroll for Caleb University, Imota, Lagos State called CALPAS. In conclusion, the automated payroll system, CALPAS, is a great improvement over the manual payroll system. It is easier to operate; it saves time, reduces or eradicates fatigue when inputting an employee's identification into the database of the payroll system and improves the method of checking an employee's statistics or report sheet without going through file. It also improves the method of knowing how money is being spent and paid to the employee. It also helps in knowing how many employees are in the company and also their bio data.

5.3 Recommendation on Future Improvement

There always room for improvement. The payroll software created CALPAS can also be improved. This is majorly because we had to create it within a limited time. With more time, the software can be improved to include more security measure and also add different types of users. Also, the software can be made network-enabled and web-enabled.

REFERENCES

Nabil Mohammed Ali Munassar and A. Govardhan (2010). A Comparison Between Five Models of Software Engineering.

Gurlal Singh (2014). Implementation and Development of a Proposed Payroll System.

Poonamdeep Kaur, Dr. Dinesh Grover (2012). Computer Based Payroll System Implementation for E-Governance at Punjab Agricultural University

Asmau Sani Mohammed, Hamman W. Samuel, Malachy Khanoba, Osaetin Evbuoma (2007). Project Report On Payroll System.

Guillermo L. Taboada, Sabela Ramos, Roberto R. Expósito, Juan Touriño, Ramón Doallo (2011). Java in the High Performance Computing Arena: Research, Practice and Experience.

APPENDIX A

```
package ProjectFolder;
import java.awt.Desktop;
import java.io.IOException;
import java.net.URI;
import java.net.URISyntaxException;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import static sun.security.jgss.GSSUtil.login;
    private void initComponents() {
        jPanel1 = new javax.swing.JPanel();
        jLabel13 = new javax.swing.JLabel();
        jLabel8 = new javax.swing.JLabel();
        jLabel9 = new javax.swing.JLabel();
        jPanel3 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        submit = new javax.swing.JButton();
        usernam = new javax.swing.JTextField();
        passwor = new javax.swing.JPasswordField();
        jLabel12 = new javax.swing.JLabel();
        jPanel2 = new javax.swing.JPanel();
        jLabel7 = new javax.swing.JLabel();
        jLabel10 = new javax.swing.JLabel();
        jLabel11 = new javax.swing.JLabel();
        jLabel6 = new javax.swing.JLabel();
        jLabel14 = new javax.swing.JLabel();
        jLabel1 = new javax.swing.JLabel();
        jLabel5 = new javax.swing.JLabel();
        jLabel15 = new javax.swing.JLabel();
        jLabel16 = new javax.swing.JLabel();
        setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
        jLabel13.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/Caleb-University-Entrance.jpg"))); // NOI18N
        javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout(jPanel1);
        jPanel1.setLayout(jPanel1Layout);
        jPanel1Layout.setHorizontalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel1Layout.createSequentialGroup()
                    .addGap(0, 0, Short.MAX_VALUE)
                    .addComponent(jLabel13))
        );
        jPanel1Layout.setVerticalGroup(
            jPanel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel1Layout.createSequentialGroup()
                    .addComponent(jLabel13, javax.swing.GroupLayout.PREFERRED_SIZE, 381,
javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addGap(0, Short.MAX_VALUE))
        );
        jLabel8.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/freshersimglogin.jpg"))); //
NOI18N
        jLabel9.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/CalebUniversityLogo.jpg"))); //
NOI18N
```

```

jPanel3.setBorder(javax.swing.BorderFactory.createTitledBorder(null, "LOGIN",
javax.swing.border.TitledBorder.CENTER, javax.swing.border.TitledBorder.ABOVE_TOP, new
java.awt.Font("Tahoma", 1, 36), new java.awt.Color(0, 0, 102))); // NOI18N
jLabel2.setFont(new java.awt.Font("Tahoma", 3, 14)); // NOI18N
jLabel2.setForeground(new java.awt.Color(0, 153, 0));
jLabel2.setText("Please Enter Your Username And Password");
jLabel3.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel3.setText("NAME:");
jLabel4.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N
jLabel4.setText("PASSWORD:");
submit.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N
submit.setForeground(new java.awt.Color(0, 153, 51));
submit.setIcon(new javax.swing.ImageIcon(getClass().getResource("/images/employee.png")));
submit.setText("Login");
submit.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        submitActionPerformed(evt);    } });
usename.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        usenameActionPerformed(evt);    } });
javax.swing.GroupLayout jPanel3Layout = new javax.swing.GroupLayout(jPanel3);
jPanel3.setLayout(jPanel3Layout);
jPanel3Layout.setHorizontalGroup(
    jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(jPanel3Layout.createSequentialGroup()
            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(jPanel3Layout.createSequentialGroup()
                    .addGroup(jPanel3Layout.createSequentialGroup()
                        .addGroup(jPanel3Layout.createSequentialGroup()
                            .addGroup(jPanel3Layout.createSequentialGroup()
                                .addGap(50, 50, 50)
                                .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 311,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addGroup(jPanel3Layout.createSequentialGroup()
                                .addGroup(jPanel3Layout.createSequentialGroup()
                                    .addContainerGap()
                                    .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.CENTER)
                                        .addComponent(jLabel4)
                                        .addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 47,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                    .addGap(18, 18, 18)
                                .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                                    .addComponent(usename, javax.swing.GroupLayout.DEFAULT_SIZE, 309,
Short.MAX_VALUE)
                                    .addComponent(passwor)))
                                .addContainerGap(25, Short.MAX_VALUE))
                            .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, jPanel3Layout.createSequentialGroup()
                                .addGap(0, 0, Short.MAX_VALUE)
                                .addComponent(submit)
                                .addGap(174, 174, 174))))
                        .addGroup(jPanel3Layout.createSequentialGroup()
                            .linkSize(javax.swing.SwingConstants.HORIZONTAL, new java.awt.Component[] {jLabel3,
jLabel4});
                            jPanel3Layout.setVerticalGroup(
                                jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                    .addGroup(jPanel3Layout.createSequentialGroup()
                                        .addGroup(jPanel3Layout.createSequentialGroup()
                                            .addComponent(jLabel2, javax.swing.GroupLayout.PREFERRED_SIZE, 28,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
                                            .addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

```



```
.addComponent(jLabel3, javax.swing.GroupLayout.PREFERRED_SIZE, 26,  
javax.swing.GroupLayout.PREFERRED_SIZE))  
.addComponent( usernam, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
.addGroup(jPanel3Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)  
.addComponent(jLabel4)  
.addComponent(password, javax.swing.GroupLayout.PREFERRED_SIZE,  
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))  
.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)  
.addComponent(submit)  
.addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE));  
jPanel3Layout.linkSize(javax.swing.SwingConstants.VERTICAL, new java.awt.Component[] {jLabel3,  
jLabel4});  
jLabel12.setFont(new java.awt.Font("Tahoma", 1, 11)); // NOI18N  
jLabel12.setText("Copyright June 2017. All Rights Reserved.");  
jLabel7.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
jLabel7.setText("Department of Computer Science");  
jLabel10.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
jLabel10.setText("Dr FAGBOOLA");  
jLabel11.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
jLabel11.setText("Caleb University Imota Lagos State");  
jLabel6.setFont(new java.awt.Font("Tahoma", 1, 14)); // NOI18N  
jLabel6.setText("Developed by KOLAWOLE EMMANUEL OLAWALE");  
jLabel14.setFont(new java.awt.Font("Tahoma", 1, 12)); // NOI18N  
jLabel14.setForeground(new java.awt.Color(51, 0, 204));  
jLabel14.setText("Click to visit Caleb Official Website");  
jLabel14.addMouseListener(new java.awt.event.MouseAdapter() {  
    public void mouseClicked(java.awt.event.MouseEvent evt) {  
        jLabel14MouseClicked(evt); }  
});  
javax.swing.GroupLayout jPanel2Layout = new javax.swing.GroupLayout(jPanel2);  
jPanel2.setLayout(jPanel2Layout);  
jPanel2Layout.setHorizontalGroup(  
    jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
        .addGroup(jPanel2Layout.createSequentialGroup()  
            .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                .addGroup(jPanel2Layout.createSequentialGroup()  
                    .addGap(0, 0, Short.MAX_VALUE).addComponent(jLabel6)  
                    .addGap(0, 0, Short.MAX_VALUE)  
                )  
                .addGroup(jPanel2Layout.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                    .addGroup(jPanel2Layout.createSequentialGroup().addGap(55, 55, 55)  
                        .addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.CENTER)  
                            .addComponent(jLabel11).addComponent(jLabel10)  
                            .addComponent(jLabel7)))  
                    .addGroup(jPanel2Layout.createSequentialGroup().addGroup(jPanel2Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)  
                        .addGroup(jPanel2Layout.createSequentialGroup().addGap(65, 65, 65)  
                            .addComponent(jLabel14))  
                        .addGroup(jPanel2Layout.createSequentialGroup().addGap(0, 0, Short.MAX_VALUE))  
                            .addComponent(submit)))  
                    .addGroup(jPanel2Layout.createSequentialGroup().addGap(39, 39, 39)  
                        .addComponent(jLabel6)  
                        .addGap(20, 20, 20)  
                        .addComponent(jLabel7)
```

```

        .addGap(20, 20, 20)
        .addComponent(jLabel10)
        .addGap(20, 20, 20)
        .addComponent(jLabel11)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel14)
        .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)));
jLabel1.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
jLabel1.setForeground(new java.awt.Color(0, 0, 204));
jLabel1.setText("Caleb University ");
jLabel5.setFont(new java.awt.Font("Tahoma", 1, 18)); // NOI18N
jLabel5.setText("Version 1.0");
jLabel15.setFont(new java.awt.Font("Tahoma", 1, 27)); // NOI18N
jLabel15.setText("CALPAS");
jLabel16.setFont(new java.awt.Font("Tahoma", 1, 36)); // NOI18N
jLabel16.setForeground(new java.awt.Color(0, 102, 0));
jLabel16.setText("Payroll System");
javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
getContentPane().setLayout(layout);
layout.setHorizontalGroup(
    layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                .addGroup(layout.createSequentialGroup()
                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                        .addGroup(layout.createSequentialGroup()
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(layout.createSequentialGroup()
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addGroup(layout.createSequentialGroup()
                                            .addComponent(jLabel9, javax.swing.GroupLayout.Alignment.TRAILING)
                                            .addGroup(layout.createSequentialGroup()
                                                .createSequentialGroup()
                                                .addContainerGap()
                                                .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 413,
javax.swing.GroupLayout.PREFERRED_SIZE))
                                            .addGroup(layout.createSequentialGroup()
                                                .createSequentialGroup()
                                                .addGap(87, 87, 87)
                                                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                                    .addGroup(layout.createSequentialGroup()
                                                        .createSequentialGroup()
                                                        .addGap(10, 10, 10)
                                                        .addComponent(jLabel16))
                                                        .addComponent(jLabel1))))
                                            .addGap(13, 13, 13))
                                .addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()
                                    .addComponent(jLabel15, javax.swing.GroupLayout.PREFERRED_SIZE, 118,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addGap(26, 26, 26)
                                    .addComponent(jLabel5)
                                    .addGap(83, 83, 83)))
                            .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                .addGroup(layout.createSequentialGroup()
                                    .createSequentialGroup()
                                    .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                                        .addComponent(jLabel12, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                                    .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                                    .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))))
                                .addGroup(layout.createSequentialGroup()
                                    .setVerticalGroup(

```

```

        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(jLabel9)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED_SIZE, 53,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(jLabel16, javax.swing.GroupLayout.PREFERRED_SIZE, 47,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(jLabel15)
        .addComponent(jLabel5))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(jLabel8, javax.swing.GroupLayout.PREFERRED_SIZE, 341,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel1, javax.swing.GroupLayout.PREFERRED_SIZE, 385,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(layout.createSequentialGroup()
        .addComponent(jPanel3, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addGap(0, 0, 0)
        .addComponent(jLabel12, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addComponent(jPanel2, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)))));
    pack();
    setLocationRelativeTo(null); } // </editor-fold>
private void jLabel14MouseClicked(java.awt.event.MouseEvent evt) {
    try { Desktop.getDesktop().browse(new URI("http://www.calebuniversity.edu.ng")); }
    catch (URISyntaxException|IOException ex){ } }
private void submitActionPerformed(java.awt.event.ActionEvent evt) {
    String username = usernam.getText();
    String password = passwor.getText();
    if (usernam.equals("") || password.equals("")){
        JOptionPane.showMessageDialog(rootPane, "EMPTY FIELDS DETECTED") } else { try {
        result = access.Login(username, password);
        catch (SQLException ex) {
            Logger.getLogger(Login.class.getName()).log(Level.SEVERE, null, ex);}
        System.out.println("result: " + result);
        if (result == 0){
            usernam.setText("");
            passwor.setText("");
            JOptionPane.showMessageDialog(rootPane, "INVALID USERNAME AND PASSWORD"); }
        if (result == 1 ){
            usernam.setText("");
            passwor.setText("")
            JOptionPane.showMessageDialog(rootPane, "LOGIN SUCCESSFUL");
            Dashboard dashboard = new Dashboard();
            dashboard.setVisible(true);} } }
    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

```

```

        if ("Nimbus".equals(info.getName())) {
            javax.swing.UIManager.setLookAndFeel(info.getClassName());
            break;}} catch (ClassNotFoundException ex) {
        java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (InstantiationException ex) {
        java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (IllegalAccessException ex) {
        java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex);
    } catch (javax.swing.UnsupportedLookAndFeelException ex) {
        java.util.logging.Logger.getLogger(HomePage.class.getName()).log(java.util.logging.Level.SEVERE, null,
ex); }
    java.awt.EventQueue.invokeLater(new Runnable() {
        public void run() {
            new HomePage().setVisible(true);}); }
private javax.swing.JLabel jLabel1;
private javax.swing.JLabel jLabel10;
private javax.swing.JLabel jLabel11;
private javax.swing.JLabel jLabel12;
private javax.swing.JLabel jLabel13;
private javax.swing.JLabel jLabel14;
private javax.swing.JLabel jLabel15;
private javax.swing.JLabel jLabel16;
private javax.swing.JLabel jLabel2;
private javax.swing.JLabel jLabel3;
private javax.swing.JLabel jLabel4;
private javax.swing.JLabel jLabel5;
private javax.swing.JLabel jLabel6;
private javax.swing.JLabel jLabel7;
private javax.swing.JLabel jLabel8;
private javax.swing.JLabel jLabel9;
private javax.swing.JPanel jPanel1;
private javax.swing.JPanel jPanel2;
private javax.swing.JPanel jPanel3;
private javax.swing.JPasswordField passwor;
private javax.swing.JButton submit;
private javax.swing.JTextField usernam;
// End of variables declaration
}

```