UNIVERSITY OF RWANDA

COLLEGE OF BUSINESS AND ECONOMICS

SCHOOL OF BUSINESS

BIT DEPARTMENT

LEVEL 2

GROUP 1

COURSE: SYSTEMS ENGINEERING, DATABASE AND JAVA PROGRAMMING

CODE: BIT 2131

PROJECT PROPOSAL

ON

ALMAHA EMPLOYEE DATABASE AND PAYROLL MANAGEMENT SYSTEM

Names: HABAKUBAHO Fidele

Registration number: 221014369

Submitted to: Dr BUGINGO Emmanuel (Lecturer)

Submission date: 20-Jun-2022

TABLE OF CONTENT:

TOPIC: ALMAHA EMPLOYEE DATABASE AND PAYROLL MANAGEMENT SYSTEM

Chapter1: SYSTEM ANALYSES:

1.1 INTRODUCTION

ALMAHA FOR INDASTRIES Co LTD is company located at BUGESERA industrial park which bring its customers product and related services such as refrigerators and accessories, solar water heaters and accessories, gas cookers, freezers and accessories, repair and servicing workshop, detergent and liquid soaps and ceramic products.

After observing how **ALMAHA** FOR INDASTRIES Co LTD works we denoted that it is not better for company like ALMAHA to use manual system while employing their employees then I proposed a ***web based computerized system*** that will handle their problems.

The proposed project "**ALMAHA Employee Database and Payroll Management System**" has been developed to overcome the problems faced in the practicing of manual system. This software is built to eliminate and in some cases reduce the hardships faced by the existing system. More over this system is designed for particular need of the company to carry out its operations in a smooth and effective manner. This web application is reduced as much as possible to avoid errors while entering data. It also provides error message while entering invalid data. It is user-friendly as no formal knowledge is required to use the system. Human resource challenges are faced by every organization which has to be overcome by the organization. Every organization has different employee and payroll management needs. Therefore, I have design exclusive

ALMAHA Employee and payroll Management System that are adapted to the organization’s Managerial Requirements.

1.2 Problem statement

Today **ALMAHA** FOR INDASTRIES Co LTD manual payroll process has become a pain for HR department as well as for the finance team as it need a lot of time, work and effort to make it successful every month. Here we present different problems of using manual payroll system which cost **ALMAHA** FOR INDASTRIES Co LTD heavily:

I. Tracking time and attendance
II. Possibilities of human errors
III. Security issues
IV. Employee productivity
V. Non-compliance

Solution

To provide an automated payroll system for supporting evolving workforce is the solution that bling HR and finance team of **ALMAHA** FOR INDASTRIES Co LTD a smile.

1.4 General Objective

“ALMAHA Employee Database and Payroll Management System” is designed to make the existing manual system automatic with the help of computerized equipment and full-edged computer software, fulfilling their requirements, so that their valuable data and information can be stored for a longer period with easy access and manipulation of the same. The required software is easily available and easy to work with. This web application can maintain and view computerized records without getting redundant entries. The project describes how to manage user data for good performance and provide better services for the client.

Specific objective

I 've proposed this project for **ALMAHA** FOR INDASTRIES Co LTD with the specific purpose of setting up the process that help them in streamline salaries, maintaining their employees' hours and payment, prepare tax payment for each pay checks, and print payroll report for the master budget and expenses budget of **ALMAHA** FOR INDASTRIES Co LTD.

1.4 Fundamental requirements

Available technology to be used:

Languages:

a) HTML: as our system is web based we will need hypertext markup language to develop web based codes.
b) PHP and JavaScript: for assisting HTML developed code.
c) Java programing language: by mean of java IDE it will be used to create interfaces and logical connection to database.

Web server:

a) WAMP server: for solution stack for Microsoft window operating system supporting MySQL database and PHP programing languages.

Development platform:

I. Adobe Dreamweaver: for quickie creation and publish web pages for HTML, CSS and JavaScript.

Available tools to be used:

This project will require tools like: editor which is Dreamweaver for both PHP, WAMP server for MySQL, also it will require operating system which is window 10.

Available hardware to be used:

Processor: intel core i3

RAM: 2GB

Hard disk: 1TB

users of this system:

1) Admin: who is in charge of system maintenance modifying the system, inserting, deleting employee and any department in **ALMAHA** FOR INDASTRIES Co LTD.
2) User: remaining **ALMAHA** FOR INDASTRIES Co LTD employee is the end user of system whose responsibilities is to look for information, using the platform to claim his/her salaries and so on.

1.5Non-functional requirements

The software was designed to fulfill the following non-functional requirements.

- **Performance Requirements:** all functions of the system must be available to the user's every time the system is turned on.

  The calculations performed by the system must comply according to the norms set by the user and should not vary unless changed by the user.

  the system will run on high performance with quick response when the user has got a good working internet connection.
- **Portability:** the system will run with a centralized warehouse of data and accessible over the World Wide Web, this makes it also to run in various number of operating system systems as it will be supported on windows (7,8,10), Mac OS accessible through different browsers including Mozilla Firefox, Google Chrome, Opera Mini, Safari, Microsoft Edge as well as UC Browser.

  The software is designed such that it works even on system having minimum configurations.
- **Security**: a strong password will be required before entering the system and use it.

- **Usability**: the system will be user friendly, menus of the system must be easily navigatable by the users with buttons that are easy to understand.
- **Adaptability:** the finished software will support new user types without needing to be written or recompiled.
- **Reliability:** the system will be available to use if the user has internet. The lower the internet functions, the higher the response time will be.
- **Legal:** the system must be licensed on an individual basis for smaller companies, as well as through multi-license deal for large companies like **ALMAHA** FOR INDASTRIES Co LTD

1.6 FEASIBILITY STUDY

After identifying the scope of the project, the feasibility study is needed to be carried out. It is basically keeping the following points in mind.

Building the software for meeting the scope: This software has met the scope. As there is no data involved in the system, processing on the file, and the behavior of this project is already identified and bundled in quantitative manner. The processing of this software is very simple as it has been designed in php and it has been well divided into several functions according to the need.

Technically feasible: This software is very much technically feasible. This software is very much concerned with specifying equipment and the software will successfully satisfy almost all the admin’s requirements. The technical need for this system may vary considerably but might include: a. The facility to produce output in a given time.

b. Response time under certain conditions.

c. Ability to process data at a particular speed.

Therefore, the basic input/output of data is identified. So, the project can easily be build up and it will also be technically feasible.

State of Art: The project is very much within the state of art since the project is a WINDOWS based; it uses very modern and common technique. Beside it is very much modern and user friendly. It also works as middleware i.e. only in between the user and the file. So, it is completely a state of art project.

Financially Feasible: The project is very much financially feasible. The implementation and development cost of this software under the reach of any college.

Moreover, it requires some training for the use. So, training cost can be neglected and the resources of this software are very much available. It also reduces the labor and extra cost to be paid for labor. So indeed, it is financially feasible.

Resources: As motioned earlier that the resources are easily available and the cost of training is almost negligible. Sometimes situations may arise when it may not be so much easy. For a person completely unaware of using a computer system could result in a training cost or for a very small organization the purchase of a computer, instalment of the system and other charges may lead to a difficult matter.

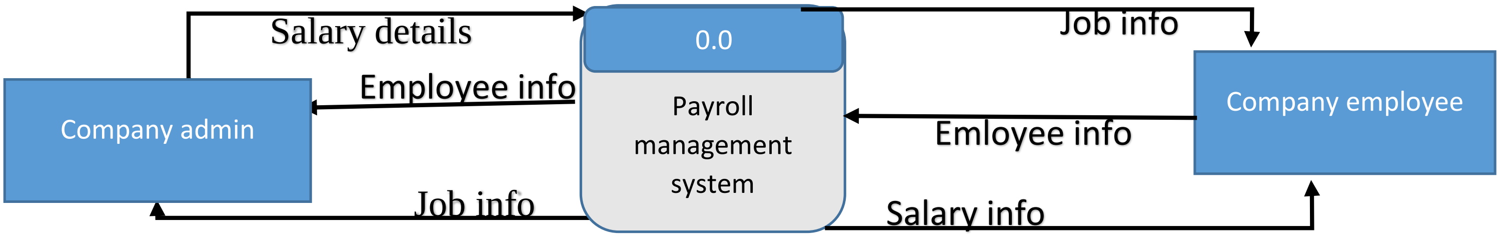
1.7 DATA FLOW DIAGRAM (DFD)

A data-flow diagram is a way of representing a flow of data through a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself

1.7.1 DATA FLOW DIAGRAM LEVEL 0

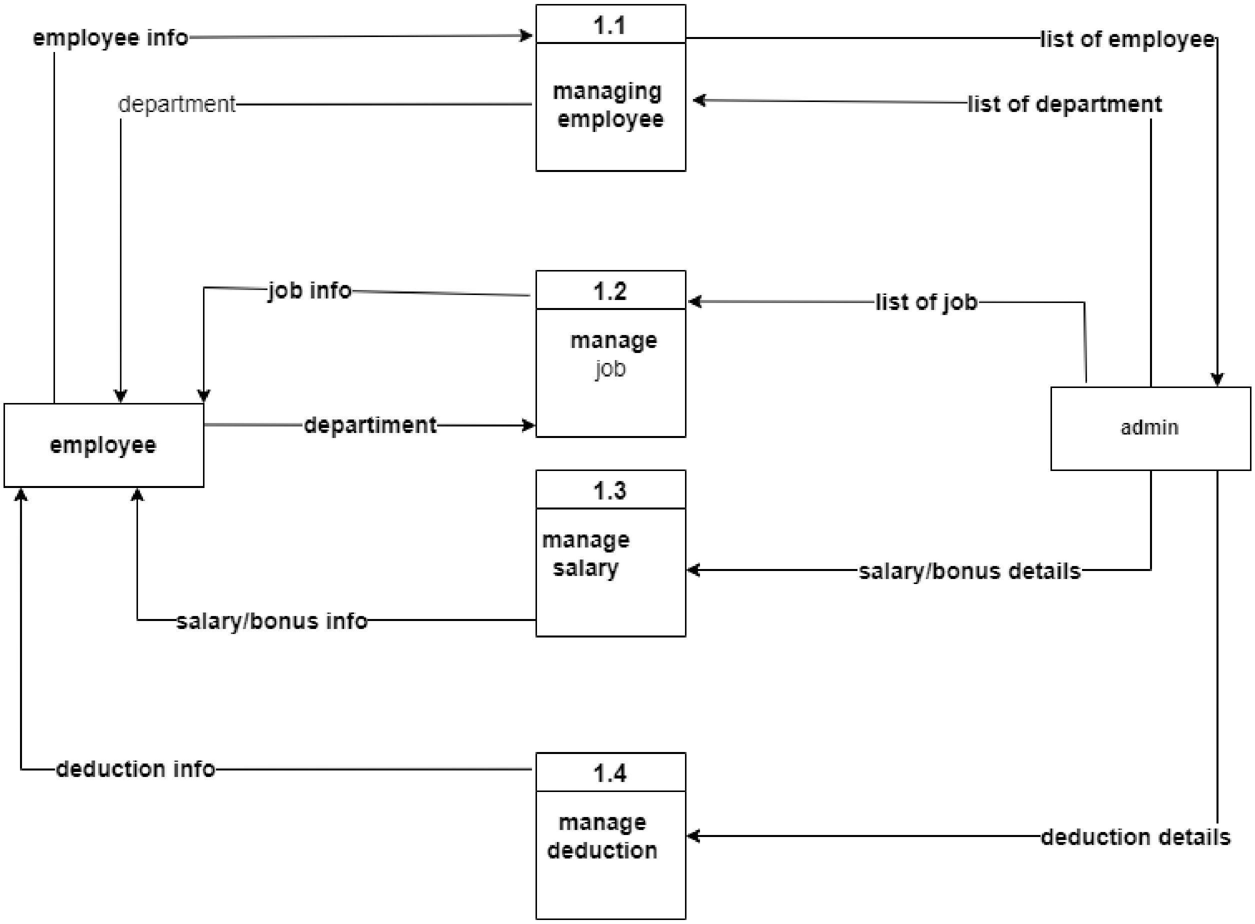
This is the zero level DFD of payroll management system, where we have elaborated the higher level process of payroll. Its basic over view of the whole payroll management system or process being analyzed or modeled. It is designed to be an at-a glance view of the task, pay slip and department showing the system as a single high-level process, with its relationship to external entities of employee, attendance and leave. It should be easily understood by the wide audience, including the employee of ALHAMA, leave and task in zero DFD of this payroll management system, I have described the high level flow of the payroll system

Figure that present level0 of ALHAMA employee database and payroll management system



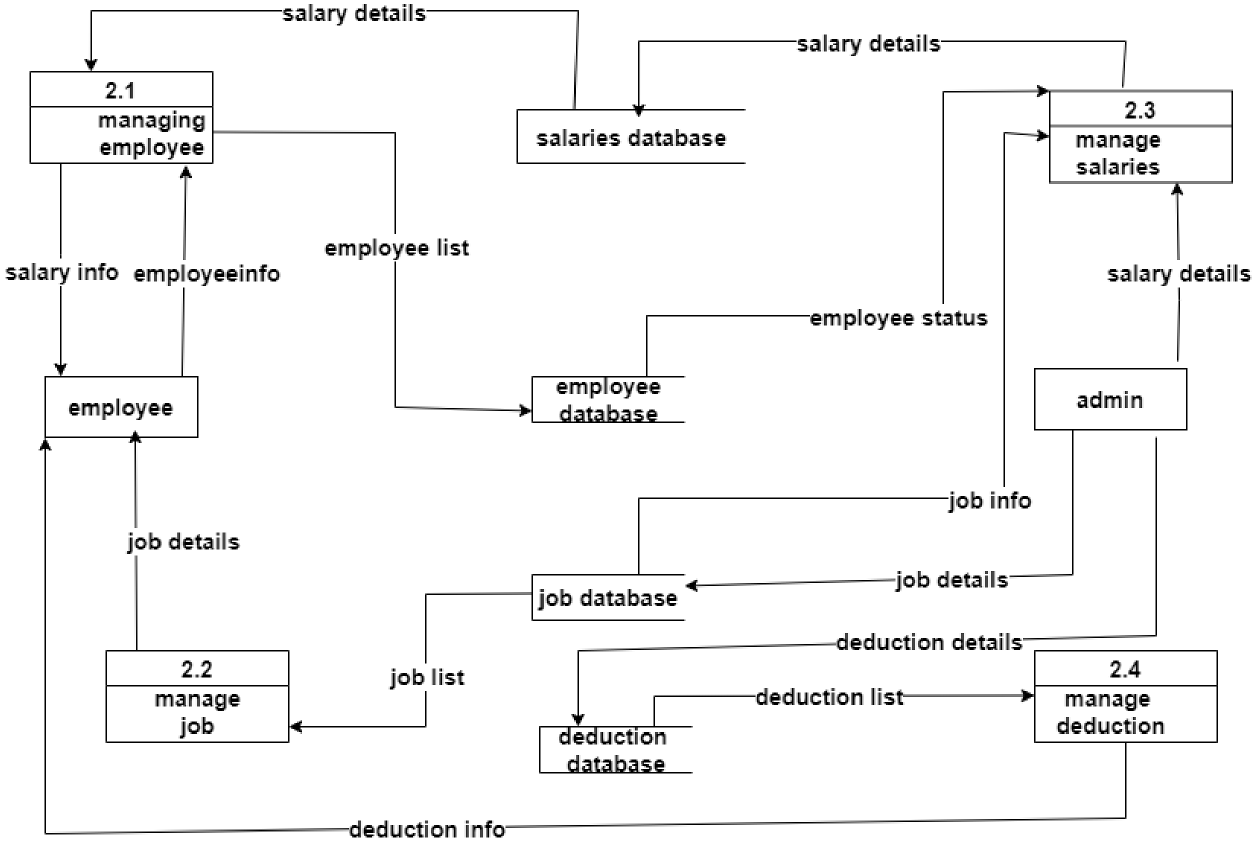
1.7.2 DATA FLOW DIAGRAM LEVEL 1

First level of DFD of proposed project payroll management system shows haw the system is divided into a subsystem each of which deal with one or more of data flows to or from an external agent, and which together provide all of the functionality of the payroll management system as a whole. It also identifies internal data stores of department, pay slip, task, salaries, leave that must be presented in order for the payroll system to do its job, and shows the flow of data between the various parts of employee, leave, pay slip, department, task of the system DFD level of our proposed project provides more detailed breakout of pieces of the first level DFD. Where we highlighted the main functionalities of the payroll.



1.7.3 A DFD that represents a decomposed level 1 DFD process is called a ***LEVEL 2 DFD***. There can be a level 2 DFD for each process that appears in the level 1 DFD.

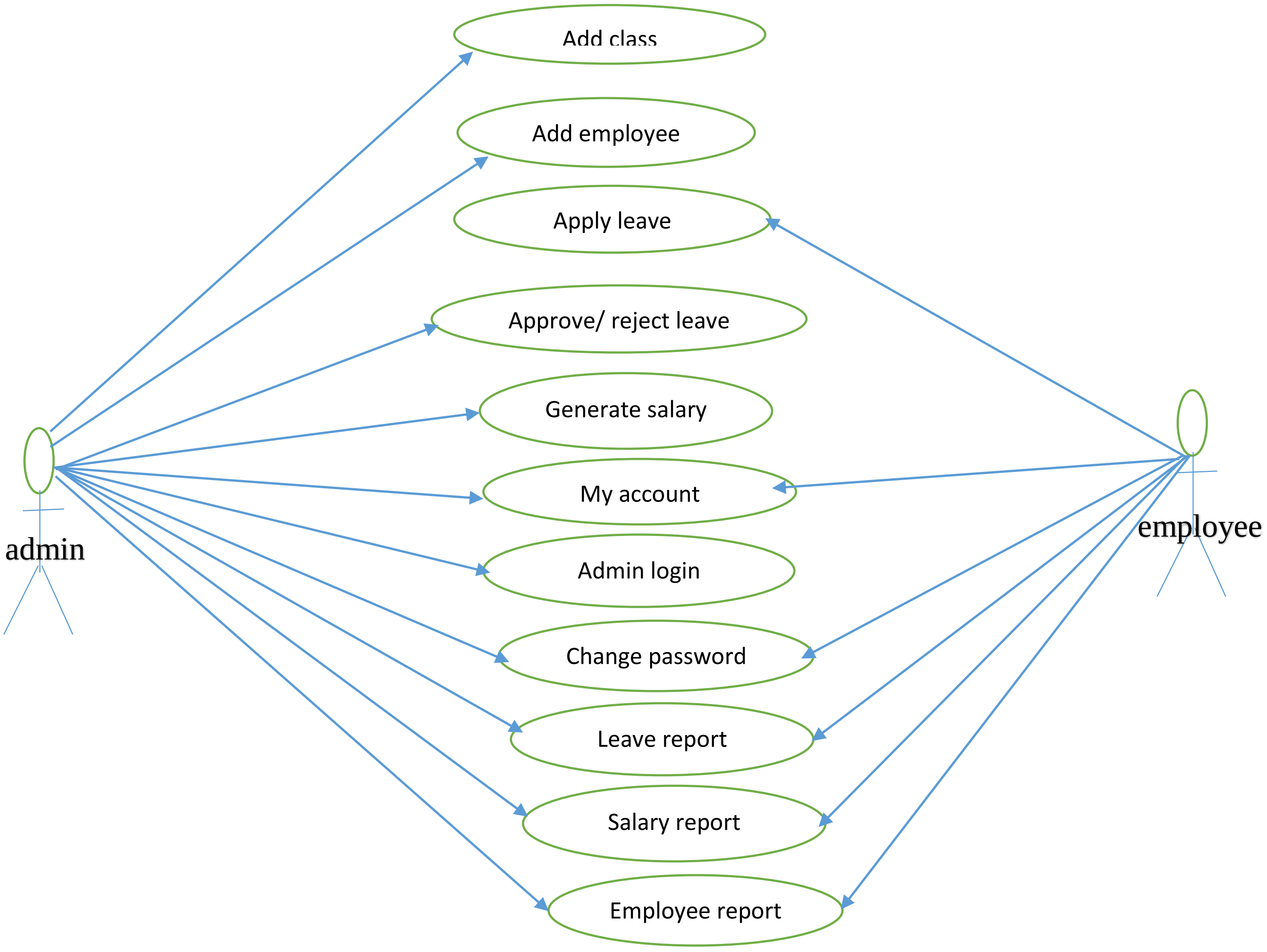
DFD level2 of our proposed project then goes one step deeper into parts of level 1 of payroll. It may require more functionalities of payroll to reach the necessary level of detail about the payroll functioning. First level DFD of payroll system we proposed shows how the system is divided into sub systems. The second level DFD of our project contain more details of department, pay slip, task, salary, leave, attendance, employee



1.8 USE CASE DIAGRAM

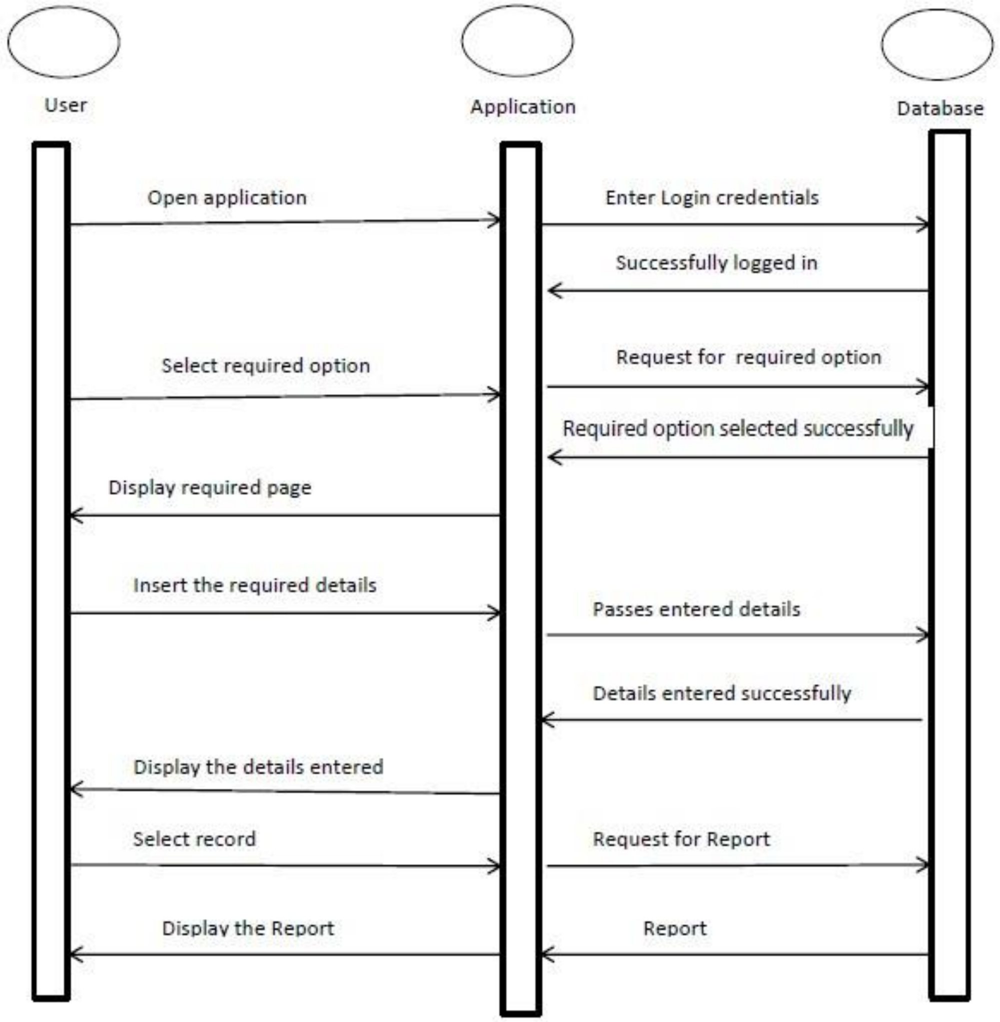
The ***use case diagram*** is usually referred to as behavior diagram used to describe the actions of all user in a ALMAHA Employee Database and Payroll Management System.

Below is figure of user case diagram of ALHAMA Employee Database and Payroll Management System.


Add class
Add employee
Apply leave
Approve/ reject leave
Generate salary
My account
Admin login
Change password
Leave report
Salary report
Employee report
admin
employee

1.8 SIQUENCY DIAGRAM

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.



1.10 Conclusion:

This project is built keeping in mind that it is to be used by only one user that is the admin. It is built for use in small scale organization where the number of employees is limited. According to the requested requirement the admin can add, manipulate, update and delete all employee data in his organization. The admin can add new departments and delete them. The Admin can also add predefined pay grades for the employees. The required records can be easily viewed by the admin anytime time he wants in an instant. The payment of the employee is based on monthly basis. Numerous validations implemented would enable the admin to enter accurate data. The main objective of this framework is to save time, make the system cost effective and management records efficiently.

References:

1. System Engineering and Analysis third edition
   (Benjamin A & Walter j Fabrycky, 2013)
2. Agile System Engineering (Bruce Powel Douglass, 2022)
3. www.incose.org>system-engineering

Chapter2: DATABASE DESIGN

DATA BASE OF THE SYSTEM

2.1INTRODUCTION:

In this chapter we will be describing database of the system tables inside that database and the way those tables were created in, addition on that in this chapter we will show table views from original tables, the relationship between created tables that are in this database.

In database of this system be ready to look at operation used on some table in this database so that you can know what to and what not to do on given entity.

Note that to develop this system database there are some material which were used so that we can get on final output of system as was described in chapter one “system analyses”. Among these the important one is xampp saver MySQL, so now let together navigate this database system.

2.2 SECTION1:

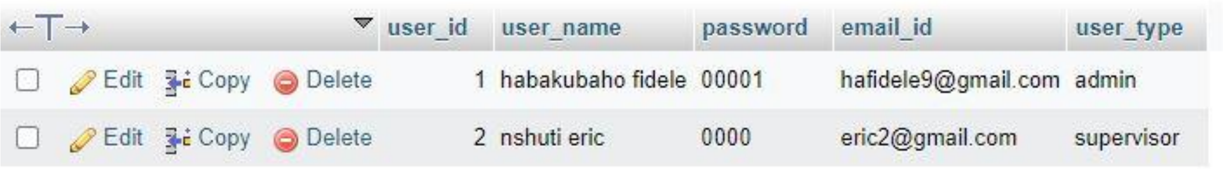
2.2.1 ENTITIES

1. Description of all entities and their corresponding attributes that are in this data base

User table database structure

This is table is table that will be only created by admin and will hold other system users apart from admin ti will give them email and password that they will use to login.

Note: the created user will not have same ability and right in the system.

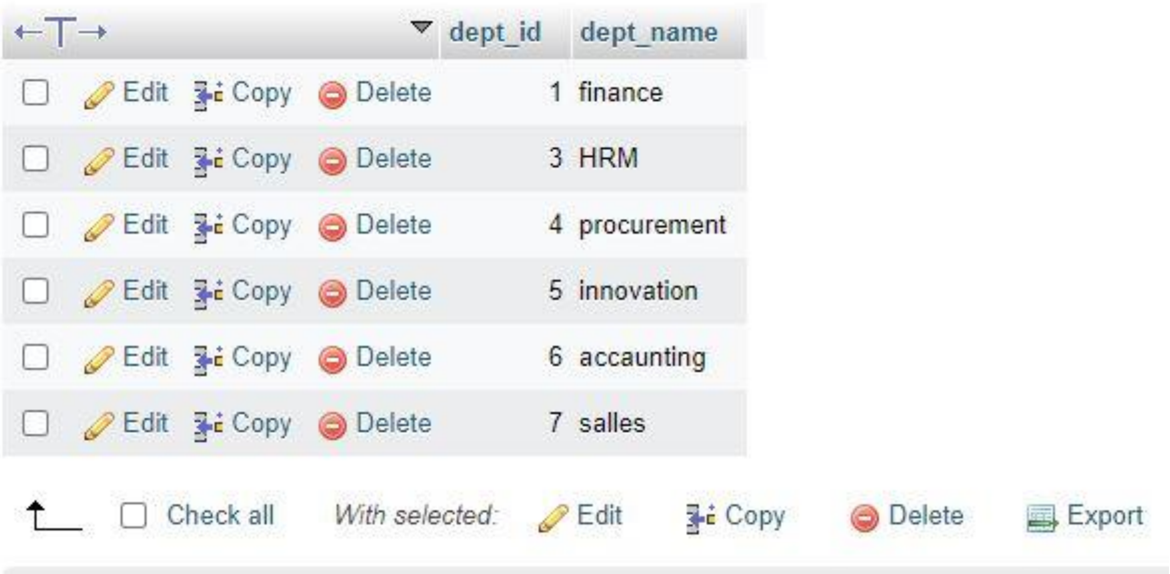


←T→				user_id	user_name	password	email_id	user_type
□	Edit	Copy	Delete	1	habakubaho fidele	00001	hafidele9@gmail.com	admin
□	Edit	Copy	Delete	2	nshuti eric	0000	eric2@gmail.com	supervisor

User_id-primary key	Id for the user
Username(varchar)	Entered name of the user
Password(varchar)	Created password of the user
Email(varchar)	Email of given user
User_type(varchar)	Who is using the system

Department table database structure:

This table to contain all department in the company and will be created by system admin and few allowed user depending on user type.



←T→	dept_id	dept_name
□ Edit Copy Delete	1	finance
□ Edit Copy Delete	3	HRM
□ Edit Copy Delete	4	procurement
□ Edit Copy Delete	5	innovation
□ Edit Copy Delete	6	accaunting
□ Edit Copy Delete	7	salles

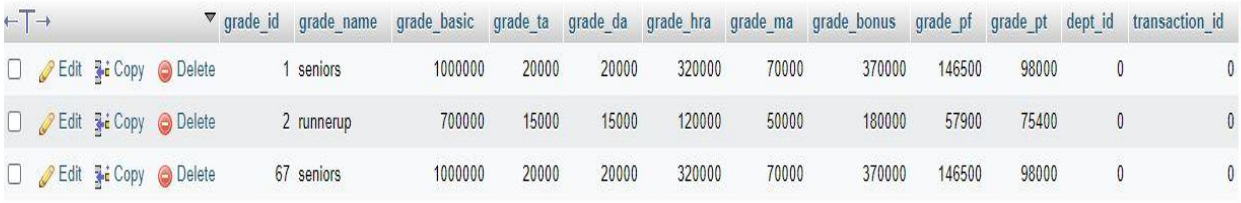
□ Check all *With selected:* Edit Copy Delete Export

Dept_id(int)-primary key	Id of the department
Dept_name(varchar)	Name of the department

Grade table database structure

It is clear that if employee spend long time in job mast have extra reward to thank his/her faithfulness so these reward will be generated according to what grade he/she have and those grade will be generated according to how long you spent in company and what degree you hold.

Under grade we will find some criteria that employee must posess to get some reward and this table will be created by system admin and some allowed user who are in department of HRM.



←T→	grade_id	grade_name	grade_basic	grade_ta	grade_da	grade_hra	grade_ma	grade_bonus	grade_pf	grade_pt	dept_id	transaction_id
□ Edit Copy Delete	1	seniors	1000000	20000	20000	320000	70000	370000	146500	98000	0	0
□ Edit Copy Delete	2	runnerup	700000	15000	15000	120000	50000	180000	57900	75400	0	0
□ Edit Copy Delete	67	seniors	1000000	20000	20000	320000	70000	370000	146500	98000	0	0

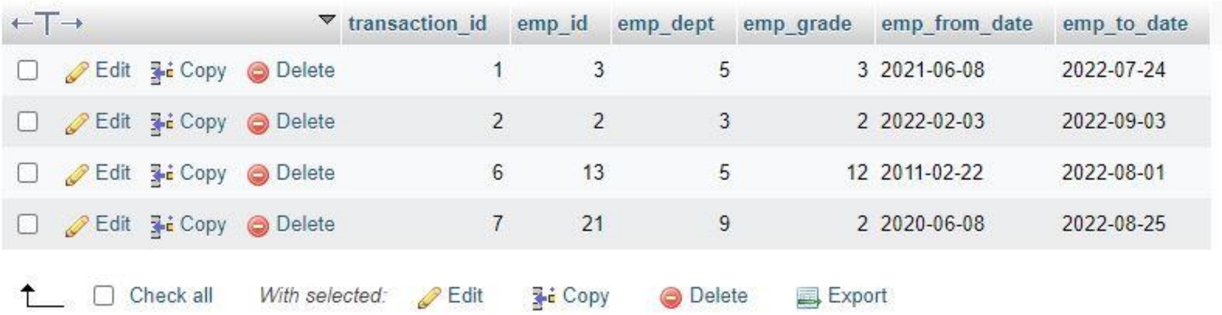
Grade_id(int)-primary key	Id of the pay grade
Grade_name(varchar)	Name of the pay grade
Grade-short_name(varchar)	Short name of the pay grade
Grade_basic(int)	The basic amount
Grade_ta(int)	The amount of travel allowance

Grade_da(int)	The amount of the dearness allowance
Grade_hra(int)	The amount of the house rent allowance
Grade_ma(int)	The amount of medical allowance
Grade_bonus(int)	The amount of bonus received
Grade_pf(int)	The amount of provident fund to be deducted
Grade_pt(int)	Amount of professional tax to be deducted

Employee grade details table database structure

as I have said above the reward and salaries will be given accordance to period in company and degree, so this table will be showing what criteria make one employee to belong in one grade that have been described in above table.

Nothing surprising only system admin and some allowed user who are in charge of HRM are the only ones to create and access that table.



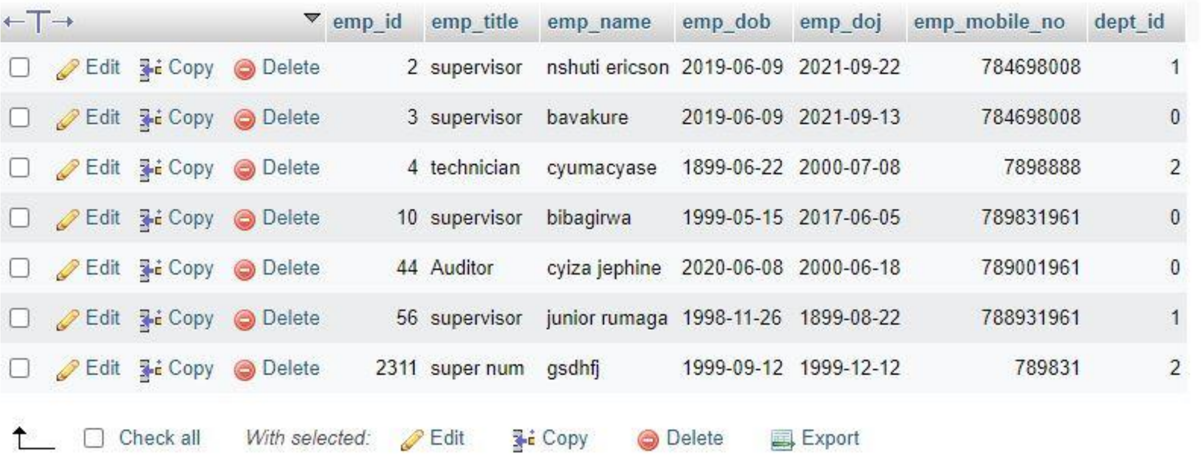
←T→	transaction_id	emp_id	emp_dept	emp_grade	emp_from_date	emp_to_date
☐ Edit Copy Delete	1	3	5	3	2021-06-08	2022-07-24
☐ Edit Copy Delete	2	2	3	2	2022-02-03	2022-09-03
☐ Edit Copy Delete	6	13	5	12	2011-02-22	2022-08-01
☐ Edit Copy Delete	7	21	9	2	2020-06-08	2022-08-25

Transaction_id(int)-primary key	Unique transaction id
Emp_id(int)	Employee id
Emp_dept_id(int)	Department id of employee
Emp_grade_id(int)	Grade id of employee
Emp_from_date(date)	Date of join of employee in department
Emp_to_date(varchar)	Last date of employee in department

Employee table database structure

Hence now the main purpose of the system is to keep track of employee information and company transactions, that why table of employee is much important in this system to hold all information of employees of the company.

At this time, it will hold information that are from employee own registration or HRM and admin registration. Thus this mean in this system employee can register on his/her own or be registered by admin or other system created and allowed user.



←T→				emp_id	emp_title	emp_name	emp_dob	emp_doj	emp_mobile_no	dept_id
□	Edit	Copy	Delete	2	supervisor	nshuti ericson	2019-06-09	2021-09-22	784698008	1
□	Edit	Copy	Delete	3	supervisor	bavakure	2019-06-09	2021-09-13	784698008	0
□	Edit	Copy	Delete	4	technician	cyumacyase	1899-06-22	2000-07-08	7898888	2
□	Edit	Copy	Delete	10	supervisor	bibagirwa	1999-05-15	2017-06-05	789831961	0
□	Edit	Copy	Delete	44	Auditor	cyiza jephine	2020-06-08	2000-06-18	789001961	0
□	Edit	Copy	Delete	56	supervisor	junior rumaga	1998-11-26	1899-08-22	788931961	1
□	Edit	Copy	Delete	2311	super num	gsdhfj	1999-09-12	1999-12-12	789831	2

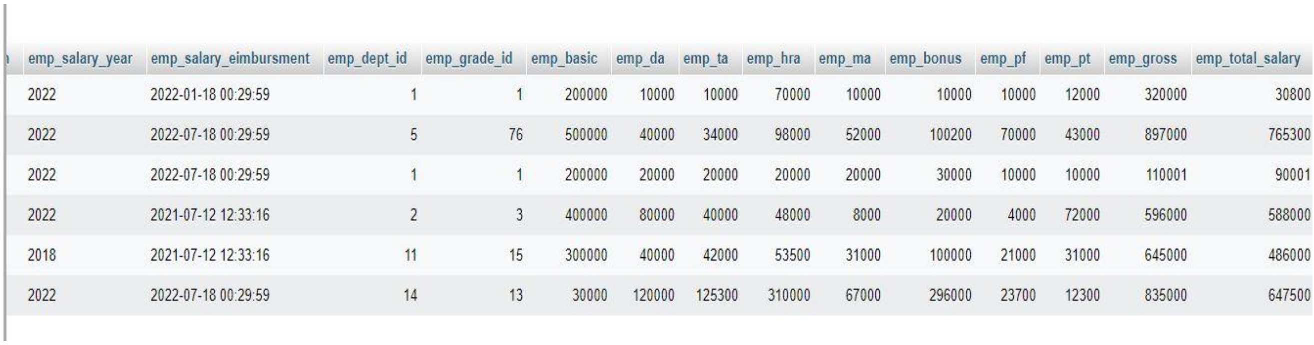
Emp_id(int)-primary key	Id of the employee
Emp_title(varchar)	Title of the employee
Emp_name(varchar)	Name of the employee
Emp_dob(date)	Data of birth of the employee
Emp_doj(date)	Date of join of the employee
Emp_address(varchar)	The address of the employee
Emp_city(varchar)	The city of the employee
Emp_pincod(int)	Pin code of the employee
Emp_mobile_no(int)	Mobile number of the employee
Emp_state(varchar)	State of the employee
Emp_mail_id(varchar)	Mail id of the employee
Emp_pan_no(varchar)	The pan number of the employee
Emp_upload-pan()	The pan card image of the employee

Employee salary details table database structure

Here we go now after keeping track of employee information the next is about what transaction the company make where will we record that? so to cope with that we need to create table for employee salaries.

This table will be in favor of company owner to know how the company is operating by knowing the out flow in term of payment and taxes.

On other hand will be in favor of government in case of auditing because it will show how tax from company is being payed and how much is deducted on every employee so that will be no tax eversion.



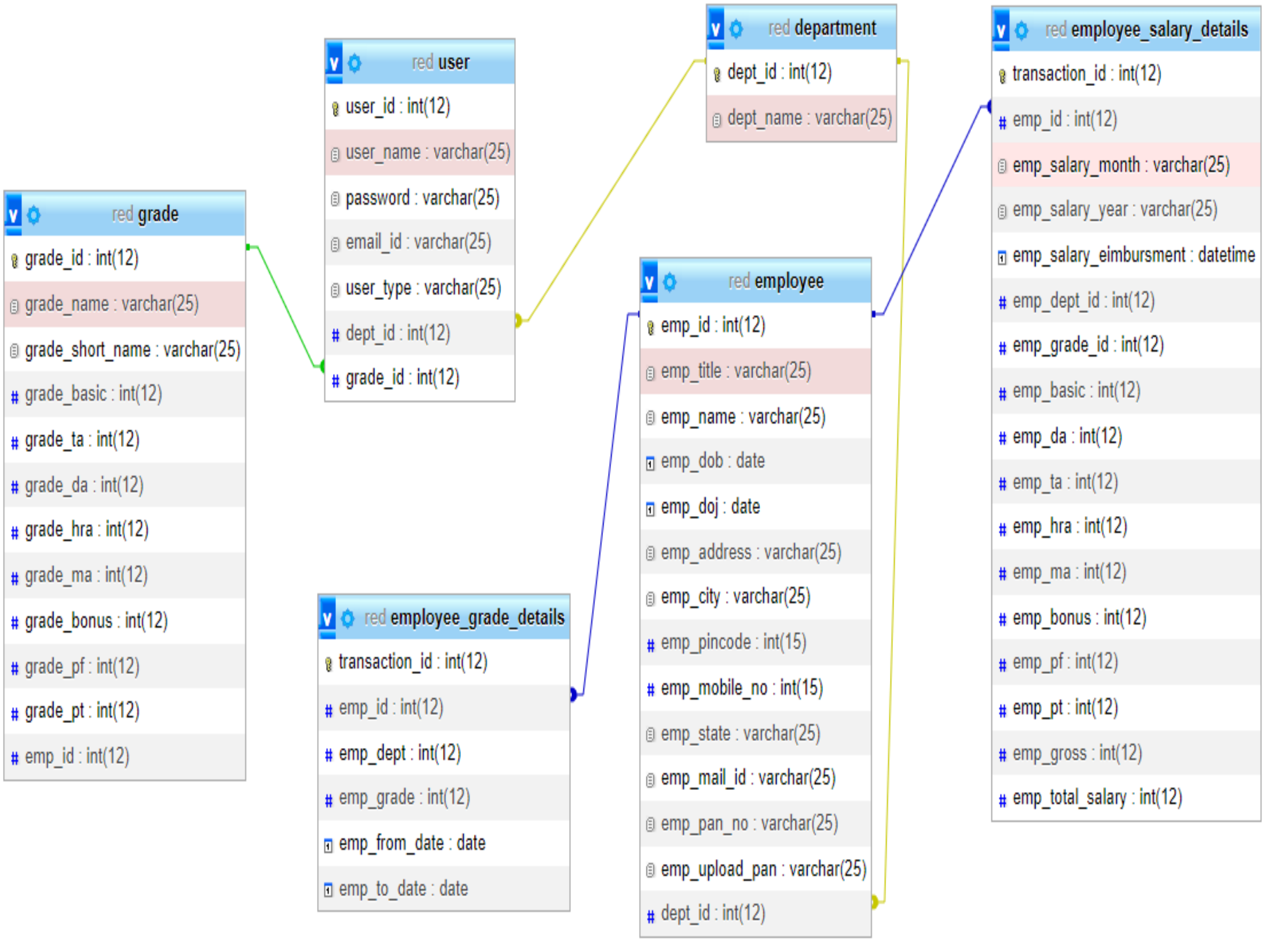
emp_salary_year	emp_salary_eimbursment	emp_dept_id	emp_grade_id	emp_basic	emp_da	emp_ta	emp_hra	emp_ma	emp_bonus	emp_pf	emp_pt	emp_gross	emp_total_salary
2022	2022-01-18 00:29:59	1	1	200000	10000	10000	70000	10000	10000	10000	12000	320000	30800
2022	2022-07-18 00:29:59	5	76	500000	40000	34000	98000	52000	100200	70000	43000	897000	765300
2022	2022-07-18 00:29:59	1	1	200000	20000	20000	20000	20000	30000	10000	10000	110001	90001
2022	2021-07-12 12:33:16	2	3	400000	80000	40000	48000	8000	20000	4000	72000	596000	588000
2018	2021-07-12 12:33:16	11	15	300000	40000	42000	53500	31000	100000	21000	31000	645000	486000
2022	2022-07-18 00:29:59	14	13	30000	120000	125300	310000	67000	296000	23700	12300	835000	647500

Transaction_id(int)-primary key	Unique primary key
Emp_id(int)	Id of the employee
Emp_salary_month(varchar)	Employee salary month
Emp_salary_year(varchar)	Employee salary year
Emp_salary_eimbursment_date(datetime)	The date and time salary was generated
Emp_dept_id(int)	Department id of the employee
Emp_grade_id(int)	Grade id of the employee
Emp_basic(int)	The amount of the basic
Emp_da(int)	The amount of the dearness allowance
Emp_ta(int)	The amount of travel allowance
Emp_hra(int)	The amount of house rent allowance
Emp_ma(int)	The amount of medical allowance
Emp_bonus(int)	The amount bonus
Emp_pf(int)	The amount of provident fund to be deducted
Emp_pt(int)	The amount of professional tax to be deducted

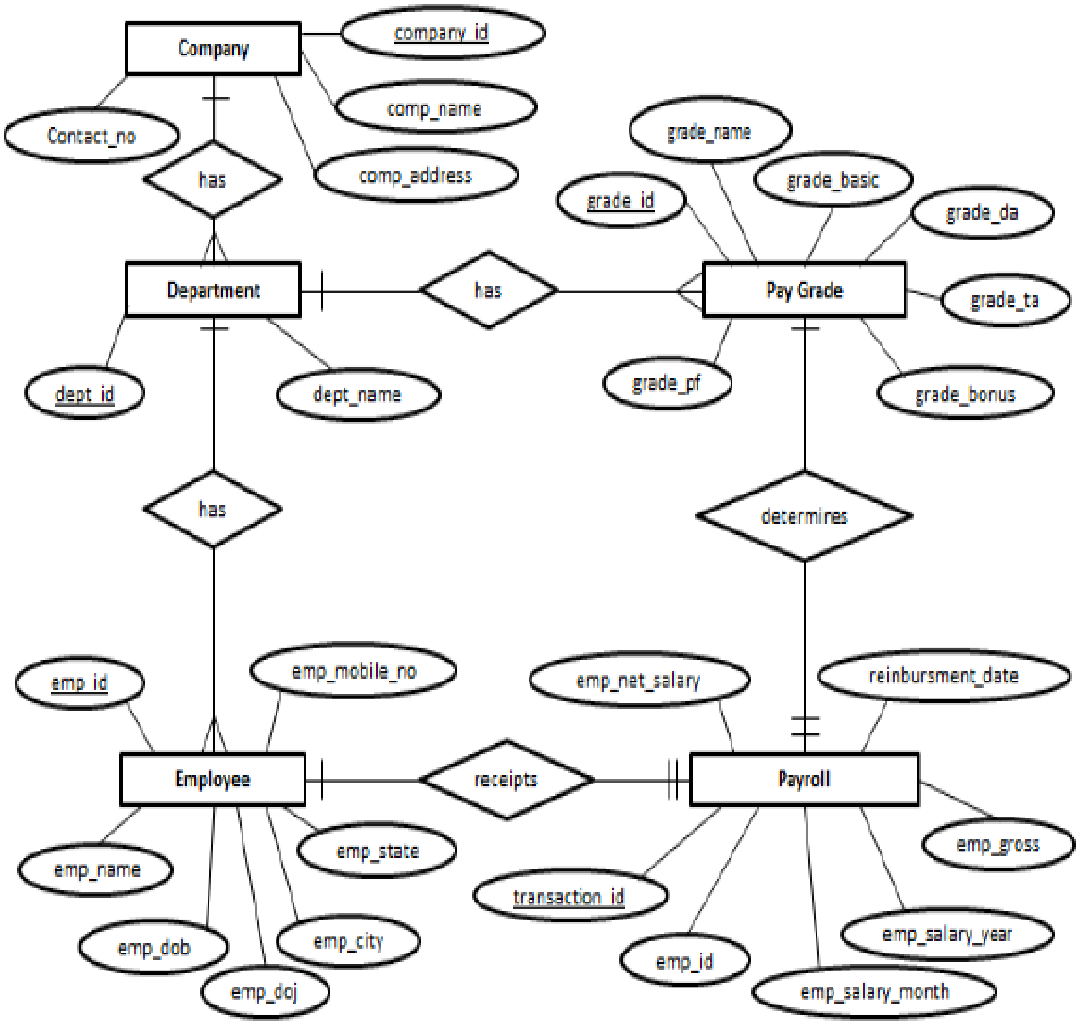
Emp_gross(int)	The gross total received by the employee
Emp_total_salary(int)	The total salary received after deduction

2.2.2 LOGICAL DATABASE MODEL

The figure below describes how table related to one another logically in this system. What to say about this graph is that foreign key and primary one will generate these relationship as shown on this logical database model as well as entity relationship diagram below that.



2.2.3 ENTITY RELETIONSHIP DIAGRAM



2.3 SECTION2:SQL

In this sub unit we will be describing techniques especially SQL queries used to create, delete tables as well as inserting and deleting data in the tables in that database.

1. ***created database is "apms"***
2. ***queries to create tables and their relationships***

```
CREATE TABLE `apms`.`employee` (`emp_id` INT(12) NOT NULL AUTO_INCREMENT , `emp_title` VARCHAR(25) NOT NULL , `emp_name` VARCHAR(25) NOT NULL , `emp_dob` DATE NOT NULL , `emp_doj` DATE NOT NULL , `emp_address` VARCHAR(25) NOT NULL , `emp_city` VARCHAR(25) NOT NULL , `emp_pincode` INT(15) NOT NULL , `emp_mobile_no` INT(15) NOT NULL , `emp_state` VARCHAR(25) NOT NULL , `emp_mail_id` VARCHAR(25) NOT NULL , `emp_pan_no` VARCHAR(25) NOT NULL , `emp_upload_pan` VARCHAR(25) NOT NULL , PRIMARY KEY (`emp_id`)) ENGINE = InnoDB;
```

```
ALTER TABLE `employee` ADD INDEX(`dept_id`);
```

```
CREATE TABLE `apms`.`employee_grade_details` (`transaction_id` INT(12) NOT NULL AUTO_INCREMENT , `emp_id` INT(12) NOT NULL , `emp_dept` INT(12) NOT NULL , `emp_grade` INT(12) NOT NULL , `emp_from_date` DATE NOT NULL , `emp_to_date` DATE NOT NULL , PRIMARY KEY (`transaction_id`)) ENGINE = InnoDB;
```

```
ALTER TABLE `employee_grade_details` ADD INDEX(`emp_id`);
```

```
CREATE TABLE `apms`.`employee_salary_details` (`transaction_id` INT(12) NOT NULL AUTO_INCREMENT , `emp_id` INT(12) NOT NULL , `emp_salary_month` VARCHAR(25) NOT NULL , `emp_salary_year` VARCHAR(25) NOT NULL , `emp_salary_eimbursment` DATETIME NOT NULL , `emp_dept_id` INT(12) NOT NULL , `emp_grade_id` INT(12) NOT NULL , `emp_basic` INT(12) NOT NULL , `emp_da` INT(12) NOT NULL , `emp_ta` INT(12) NOT NULL , `emp_hra` INT(12) NOT NULL , `emp_ma` INT(12) NOT NULL , `emp_bonus` INT(12) NOT NULL , `emp_pf` INT(12) NOT NULL , `emp_pt` INT(12) NOT NULL , `emp_gross` INT(12) NOT NULL , `emp_total_salary` INT(12) NOT NULL , PRIMARY KEY (`transaction_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `amps`.`user` (`user_id` INT(12) NOT NULL AUTO_INCREMENT , `user_name` VARCHAR(25) NOT NULL , `password` VARCHAR(25) NOT NULL , `email_id` VA
RCHAR(25) NOT NULL , `user_type` VARCHAR(25) NOT NULL , PRIMARY KEY (`user_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `apms`.`department` (`dept_id` INT(12) NOT NULL AUTO_INCREMENT , `dept_name` VARCHAR(25) NOT NULL , PRIMARY KEY (`dept_id`)) ENGINE = InnoDB;
```

```
CREATE TABLE `red`.`grade` (`grade_id` INT(12) NOT NULL AUTO_INCREMENT , `grade_name` VARCHAR(25) NOT NULL , `grade_short_name` VARCHAR(25) NOT NULL , `grade_basic` INT(12) NOT NULL , `grade_ta` INT(12) NOT NULL , `grade_da` INT(12) NOT NULL , `grade_hra` INT(12) NOT NULL , `grade_ma` INT(12) NOT NULL , `grade_bonus` INT(12) NOT NULL , `grade_pf` INT(12) NOT NULL , `grade_pt` INT(12) NOT NULL , PRIMARY KEY (`grade_id`)) ENGINE = InnoDB;
```

```
ALTER TABLE `grade` ADD INDEX(`dept_id`);
```

3. Queries to insert data into tables

```
INSERT INTO `user` (`user_id`, `user_name`, `password`, `email_id`, `user_type`) VALUES ('1', 'habakubaho fidele', '0000', 'hafidele9@gmail.com', 'admin'), ('2', 'nshuti eric', '0000', 'eric2@gmail.com', 'supervisor')
```

```
INSERT INTO `grade` (`grade_id`, `grade_name`, `grade_short_name`, `grade_basic`, `grade_ta`, `grade_da`, `grade_hra`, `grade_ma`, `grade_bonus`, `grade_pf`, `grade_pt`) VALUES ('1', 'seniors', 's', '1000000', '20000', '20000', '320000', '70000', '370000', '146500', '98000'), ('2', 'runnerup', 'r', '700000', '15000', '15000', '120000', '50000', '180000', '57900', '75400')
```

```
INSERT INTO `employee_salary_details` (`transaction_id`, `emp_id`, `emp_salary_month`, `emp_salary_year`, `emp_salary_eimbursment`, `emp_dept_id`, `emp_grade_id`, `emp_basic`, `emp_da`, `emp_ta`, `emp_hra`, `emp_ma`, `emp_bonus`, `emp_pf`, `emp_pt`, `emp_gross`, `emp_total_salary`) VALUES ('1', '1', 'may', '2022', '2022-07-18 00:29:59.000000', '12', '1', '30000', '120000', '125300', '310000', '67000', '296000', '23700', '12300', '835000', '647500'), ('2', '34', 'june', '2022', '2022-07-18 00:29:59.000000', '5', '76', '500000', '40000', '34000', '98000', '52000', '100200', '70000', '43000', '897000', '765300')
```

```
INSERT INTO `employee_grade_details` (`transaction_id`, `emp_id`, `emp_dept`, `emp_grade`, `emp_from_date`, `emp_to_date`) VALUES ('1', '3', '5', '3', '2021-06-08', '2022-07-24'), ('7', '21', '9', '2', '2020-06-08', '2022-08-25')
INSERT INTO `employee` (`emp_id`, `emp_title`, `emp_name`, `emp_dob`, `emp_doj`, `emp_address`, `emp_city`, `emp_pincode`, `emp_mobile_no`, `emp_state`, `emp_mail_id`, `emp_pan_no`, `emp_upload_pan`) VALUES ('1', 'admin', 'habakubaho fidele', '5-15-1999', '2017-06-05', 'ntarama-bugesera', 'nyamata', '00000', '0789831961', 'rwanda', 'hafidele9@gmail.com', '00000', ''), ('2', 'supervisor', 'nshuti eric', '2019-06-09', '2021-09-13', 'huye', 'huye', '00000', '0784698008', 'rwanda', 'eric12@gamail.com', '00000', '')
```

```
INSERT INTO `department` (`dept_id`, `dept_name`) VALUES ('1', 'finance'), ('2', 'marketing and research'), ('3', 'HRM'), ('4', 'procurement')
```

4. ***queries to display all information in created tables***

```
SELECT * FROM `user`

SELECT * FROM `grade`

SELECT * FROM `employee_salary_details`

SELECT * FROM `employee_grade_details`

SELECT * FROM `employee`

SELECT * FROM `department`
```

5. ***queries to update two tables in Apms database***

```
UPDATE `employee_salary_details` SET `transaction_id`='98',`emp_id`='21',`emp_salary_month`='SEPT',`emp_salary_year`='1999',`emp_salary_eimbursment`='2021-7-12 12:33:16',`emp_dept_id`='67',`emp_grade_id`='34',`emp_basic`='120000',`emp_da`='50000',`emp_ta`='10000',`emp_hra`='76000',`emp_ma`='33000',`emp_bonus`='500000',`emp_pf`='28000',`emp_pt`='96000',`emp_gross`='895000',`emp_total_salary`='693000' WHERE `emp_id`='1'
```

```
UPDATE `user` SET `user_id`='1',`user_name`='habakubaho fidele',`password`='00001',`email_id`='hafidele9@gmail.com',`user_type`='admin' WHERE `user_type`='admin'
```

2.3 SECTION3

2.3.1 VIEWS, TRIGGERS AND PROCEDURES

1. *view to insert data in given tables*

```
INSERT INTO `transactions`(`user_id`, `user_name`, `password`, `email_id`, `user_type`) VALUES ('33','kadahumeka ally','00123','kada@gmail.com','electrician')
```

```
INSERT INTO `list_of_employee`(`emp_id`, `emp_title`, `emp_name`, `emp_dob`, `emp_doj`, `emp_address`, `emp_city`, `emp_pincode`, `emp_mobile_no`, `emp_state`, `emp_mail_id`, `emp_pan_no`, `emp_upload_pan`) VALUES ('56','supervisor','junior rumaga','1998-11-26','1899-8-22','nyagatare','nyagatare','00000','0788931961','rwanda','[value-11]','rumaga1@gmail.com','')
```

```
INSERT INTO `list_of_departments`(`dept_id`, `dept_name`) VALUES ('5','innovation and creativity')
```

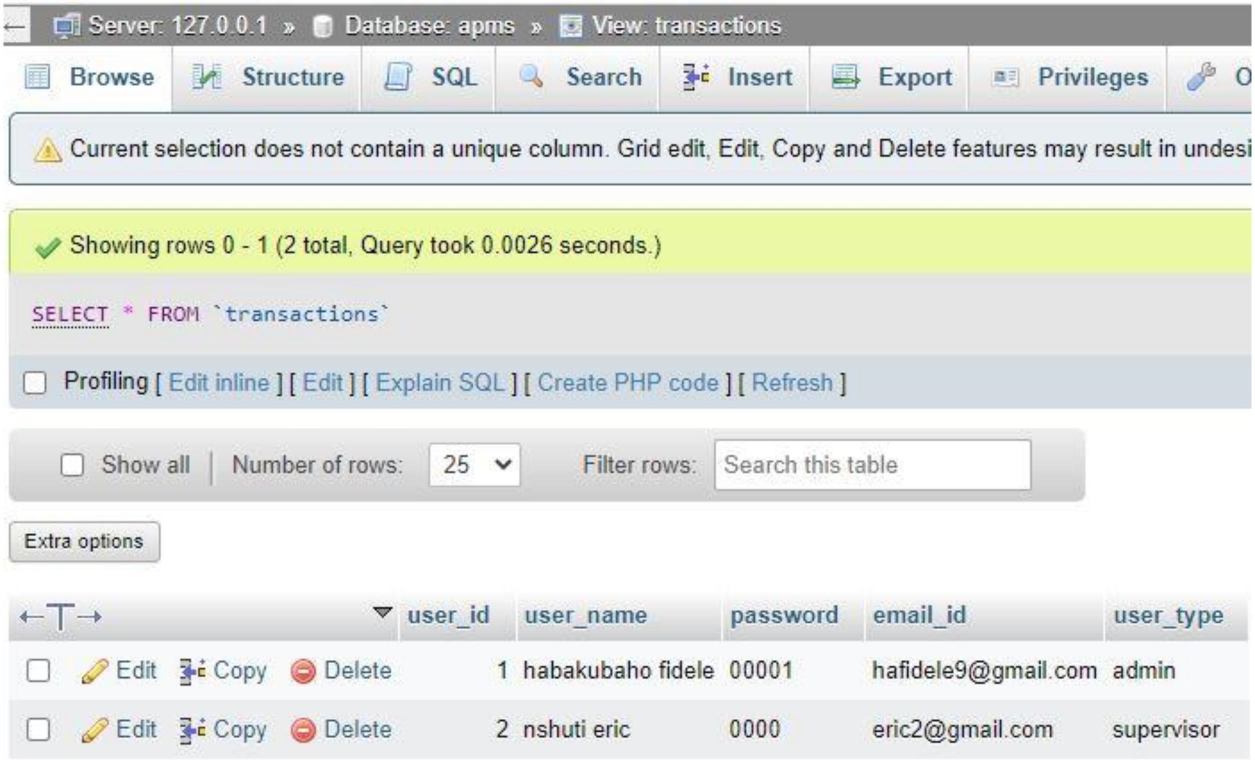
```
INSERT INTO `level`(`grade_id`, `grade_name`, `grade_short_name`, `grade_basic`, `grade_ta`, `grade_da`, `grade_hra`, `grade_ma`, `grade_bonus`, `grade_pf`, `grade_pt`) VALUES ('14','bignners','b','300000','30000','12000','63000','21000','987000','63000','51632')
```

```
INSERT INTO `details_on_salaries`(`transaction_id`, `emp_id`, `emp_salary_month`, `emp_salary_year`, `emp_salary_eimbursment`, `emp_dept_id`, `emp_grade_id`, `emp_basic`, `emp_da`, `emp_ta`, `emp_hra`, `emp_ma`, `emp_bonus`, `emp_pf`, `emp_pt`, `emp_gross`, `emp_total_salary`) VALUES ('11','15','may','2018','2019-1-118:13:37','11','15','300000','40000','42000','53500]','31000','100000','21000','31000','645000','486000')
```

```
INSERT INTO `details_on_grade`(`transaction_id`, `emp_id`, `emp_dept`, `emp_grade`, `emp_from_date`, `emp_to_date`) VALUES ('6','13','5','r','2011-2-22','2022-8-1')
```

2. *view to display all the information in created tables*

```
CREATE VIEW transactions AS SELECT * FROM `user`
```


Server: 127.0.0.1 » Database: apms » View: transactions
Browse
Structure
SQL
Search
Insert
Export
Privileges
O

⚠ Current selection does not contain a unique column. Grid edit, Edit, Copy and Delete features may result in undesi

✓ Showing rows 0 - 1 (2 total, Query took 0.0026 seconds.)

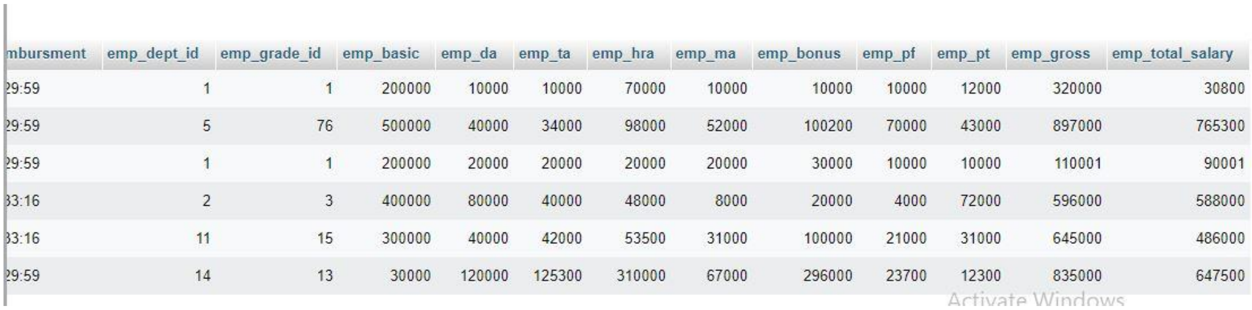
```
SELECT * FROM `transactions`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25 Filter rows: Search this table
Extra options

←T→	user_id	user_name	password	email_id	user_type
□ Edit Copy Delete	1	habakubaho fidele	00001	hafidele9@gmail.com	admin
□ Edit Copy Delete	2	nshuti eric	0000	eric2@gmail.com	supervisor

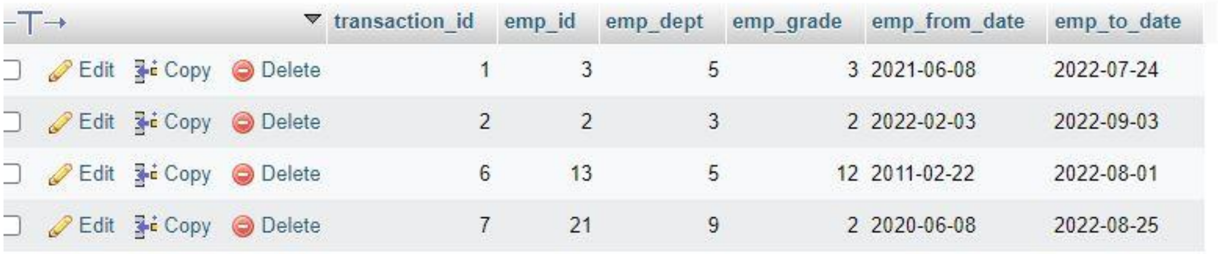
CREATE VIEW level AS SELECT * FROM `grade`

CREATE VIEW details_on_salaries AS SELECT * FROM `employee_salary_details`



mbursment	emp_dept_id	emp_grade_id	emp_basic	emp_da	emp_ta	emp_hra	emp_ma	emp_bonus	emp_pf	emp_pt	emp_gross	emp_total_salary
29:59	1	1	200000	10000	10000	70000	10000	10000	10000	12000	320000	30800
29:59	5	76	500000	40000	34000	98000	52000	100200	70000	43000	897000	765300
29:59	1	1	200000	20000	20000	20000	20000	30000	10000	10000	110001	90001
33:16	2	3	400000	80000	40000	48000	8000	20000	4000	72000	596000	588000
33:16	11	15	300000	40000	42000	53500	31000	100000	21000	31000	645000	486000
29:59	14	13	30000	120000	125300	310000	67000	296000	23700	12300	835000	647500

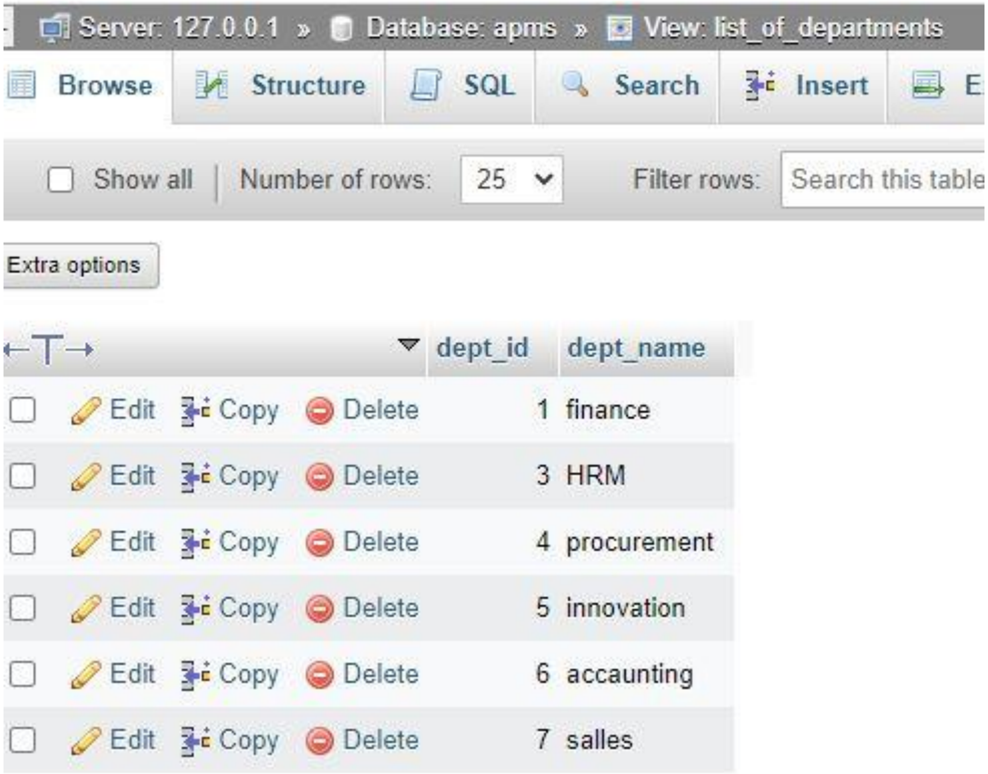
CREATE VIEW details_on_grade AS SELECT * FROM `employee_grade_details`



-T→	transaction_id	emp_id	emp_dept	emp_grade	emp_from_date	emp_to_date
□ Edit Copy Delete	1	3	5	3	2021-06-08	2022-07-24
□ Edit Copy Delete	2	2	3	2	2022-02-03	2022-09-03
□ Edit Copy Delete	6	13	5	12	2011-02-22	2022-08-01
□ Edit Copy Delete	7	21	9	2	2020-06-08	2022-08-25

CREATE VIEW list_of_employee AS SELECT * FROM `employee`

CREATE VIEW list_of_departments AS SELECT * FROM `department`


Server: 127.0.0.1 » Database: apms » View: list_of_departments
Browse
Structure
SQL
Search
Insert
E
Show all
Number of rows: 25
Filter rows: Search this table
Extra options

←T→	dept_id	dept_name
□ Edit Copy Delete	1	finance
□ Edit Copy Delete	3	HRM
□ Edit Copy Delete	4	procurement
□ Edit Copy Delete	5	innovation
□ Edit Copy Delete	6	accaunting
□ Edit Copy Delete	7	salles

3. **<u>view to update information in any two tables</u>**

UPDATE `transactions` SET `user_id`='3',`user_name`='kadahumeka henry',`password`='00111',`email_id`='kadahumeka2@gmail',`user_type`='techinician' WHERE `user_id`='33'

UPDATE `list_of_departments` SET `dept_id`='2',`dept_name`='sales marketing and research' WHERE`dept_id`='2'

4. **<u>view to delete data in any two tables</u>**

DELETE FROM `level` WHERE grade_id=14

DELETE FROM `list_of_departments` WHERE dept_id=2

5.<u>view to consider sub-queries</u>

CREATE VIEW details_on_grade AS SELECT * FROM `employee_grade_details(SELECT* level` WHERE grade_id=14)

Section4

1.creating stored procedure to insert data in tables

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE insertdepartment(IN dept_idparam int(12),
dept_nameParam varchar(25))

BEGIN

   insert into department values (dept_idParam,dept_nameParam);

END $$

DELIMITER ;

call insertdepartment(44,'Auditing')
```

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE insertemployee(IN emp_id_idparam
int(12),emp_titleparam varchar(25),emp_nameParam varchar(25),emp_dobparam
DATE,emp_dojparam DATE,emp_addressparam VARCHAR(25),emp_cityparam
VARCHAR(25),emp_pincodeparam INT(15),emp_mobile_noparam
INT(15),emp_stateparam VARCHAR(25),emp_mail_idparam
VARCHAR(25),emp_pan_noparam VARCHAR(25),emp_upload_panparam
VARCHAR(25))

BEGIN

   insert into employee values (emp_id_idparam,emp_titleparam,emp_nameParam
,emp_dobparam ,emp_dojparam,emp_addressparam,emp_cityparam ,emp_pincodeparam
,emp_mobile_noparam ,emp_stateparam ,emp_mail_idparam ,emp_pan_noparam
,emp_upload_panparam);

END $$

DELIMITER ;

call insertemployee(44,'Auditor','cyiza jeph','2020-06-08','2000-06-18','ntarama-bugesera',
'nyamata', '00000', '0789001961', 'rwanda', 'cyiza9@gmail.com', '00000', '')



DELIMITER $$

CREATE OR REPLACE PROCEDURE insertemployee_grade_details(IN
transaction_idparam INT(12),emp_idparam INT(12),emp_deptparam INT(12),
emp_gradeparam INT(12),emp_from_dateparam DATE,emp_to_dateparam DATE)

BEGIN

   insert into employee_grade_details values (transaction_idparam, emp_idparam,
emp_deptparam,emp_gradeparam,emp_from_dateparam,emp_to_dateparam);

END $$

DELIMITER ;

call insertemployee_grade_details(12, 33, '5', '3', '2021-06-08', '2022-07-24')



DELIMITER $$

CREATE OR REPLACE PROCEDURE insertemployee_salary_details(IN
transaction_idparam INT(12) ,emp_idparam INT(12) , emp_salary_monthparam
VARCHAR(25) , emp_salary_yearparam VARCHAR(25) ,
emp_salary_eimbursmentparam DATETIME,emp_dept_idparam
INT(12),emp_grade_idparam INT(12),emp_basicparam INT(12),emp_daparam
INT(12),emp_taparam INT(12),emp_hraparam INT(12),emp_maparam
INT(12),emp_bonusparam INT(12),emp_pfparam INT(12),emp_ptparam
INT(12),emp_grossparam INT(12),emp_total_salaryparam INT(12))

BEGIN
```

```
    insert into employee_salary_details values
(transaction_idparam,emp_idparam,emp_salary_monthparam,
emp_salary_yearparam,emp_salary_eimbursmentparam,emp_dept_idparam,emp_grade_i
dparam,emp_basicparam,emp_daparam,emp_taparam,emp_hraparam,emp_maparam,em
p_bonusparam,emp_pfparam,emp_ptparam,emp_grossparam,emp_total_salaryparam);

END $$

DELIMITER ;

call insertemployee_salary_details(27, 31, 'may', '2022', '2022-07-18 00:29:59.000000', 14,
13, 30000, 120000, 125300, 310000, 67000, 296000, 23700, 12300, 835000, 647500)
```

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE insertgrade(IN grade_idparam int(12),
grade_nameparam varchar(25), grade_short_nameparam varchar(25), grade_basicparam
int(12),grade_taparam int(12),grade_daparam int(12), grade_hraparam
int(12),grade_maparam int(12),grade_bonusparam int(12), grade_pfparam int(12),
grade_ptparam int(25))

BEGIN

    insert into grade values (grade_idparam, grade_nameparam, grade_short_nameparam,
grade_basicparam,grade_taparam ,grade_daparam, grade_hraparam,grade_maparam
,grade_bonusparam, grade_pfparam, grade_ptparam);
```

```
END $$

DELIMITER ;

call insertgrade(67, 'seniors', 's', 1000000, 20000, 20000, 320000, 70000, 370000, 146500,
98000)


DELIMITER $$

CREATE OR REPLACE PROCEDURE insertuser(IN user_idparam
int(12),user_nameparam varchar(25),passwordparam varchar(25),email_idparam
varchar(25),user_typeparam varchar(25))

BEGIN

   insert into user values (user_idparam,user_nameparam,passwordparam,email_idparam
,user_typeparam);

END $$

DELIMITER ;

call insertuser(12, 'ndungutse john', '0000', 'ndungu@gmail.com', 'driver')
```

3 <u>stored procedures to display all information in tables</u>

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE getdepartmentDetails()

BEGIN

   select * from department;

END $$

DELIMITER ;
DELIMITER $$
CREATE OR REPLACE PROCEDURE getemployeeDetails()
BEGIN
   select * from employee;
END $$
DELIMITER ;

DELIMITER $$
CREATE OR REPLACE PROCEDURE getgrade_of_employeeDetails()
BEGIN
   select * from employee_salary_details;
END $$
DELIMITER




DELIMITER $$
CREATE OR REPLACE PROCEDURE getemp_salaryDetails()
BEGIN
   select * from employee_salary_details;
END $$
DELIMITER ;

DELIMITER $$
```

```
CREATE OR REPLACE PROCEDURE getgradeDetails()
BEGIN
   select * from grade;
END $$
DELIMITER ;

DELIMITER $$
CREATE OR REPLACE PROCEDURE getuserDetails()
BEGIN
   select * from user;
END $$
DELIMITER ;
```

4 stored procedure to update information in any two tables

```
DELIMITER //
CREATE OR REPLACE PROCEDURE updateuser(IN user_idParam int(12))
BEGIN
   update user set password=00022 where user_id=user_idParam;
END //
DELIMITER ;
DELIMITER //
CREATE OR REPLACE PROCEDURE updategrade(IN grade_idParam int(12))
BEGIN
   update grade set grade_da=20000 where grade_id= grade_idParam;
END //
DELIMITER ;
```

5 stored procedure to delete information in any two tables

```
DELIMITER $$
CREATE OR REPLACE PROCEDURE deleteuser(IN user_idParam int(12))
BEGIN
delete from user where user_id=12;
END $$
DELIMITER ;
```

```
DELIMITER $$
CREATE OR REPLACE PROCEDURE deletegrade(IN grade_idParam int(12))
BEGIN
delete from grade where grade_id=67;
END $$
DELIMITER ;
```

6 stored procedure view that consider sub-queries

```
DELIMITER $$

CREATE OR REPLACE PROCEDURE VIEW insertdepartment(IN dept_idparam
int(12), dept_nameParam varchar(25))

BEGIN

 insert into department values (SELECT dept_idParam,dept_nameParam FROM
department);

END $$

DELIMITER ;

call insertdepartment(44,'Auditing')
```

section5

1. **create insert triggers for any two tables**

```
DELIMITER $$

CREATE TRIGGER after_user_insert

AFTER INSERT

ON user FOR EACH ROW

BEGIN

INSERT INTO user(user_idId,user_name,password,email_id,user_type)

VALUES(77,'mbogo ally','00333','mbogo@gmail.com','worker');

END$$

DELIMITER ;


DELIMITER $$

CREATE TRIGGER after_grade_insert

AFTER INSERT

ON grade FOR EACH ROW

BEGIN

 INSERT INTO grade(grade_id,grade_name,grade_short_name,grade_basic,grade_ta,

          grade_da,grade_hra,grade_ma,grade_bonus,grade_pf,grade_pt)
```

```
    VALUES(6,'sinior','s',500000,70000,20000,360000,120000,300000,135000,780000);
END$$
DELIMITER ;
```

2. trigger to update any two tables

```
DELIMITER $$
CREATE TRIGGER after_user_update
AFTER UPDATE
ON user FOR EACH ROW
BEGIN
UPDATE user SET grade_name='mbogamizi ally'WHERE user_id=77;
END$$
DELIMITER ;
```

```
DELIMITER $$
CREATE TRIGGER after_grade_update
AFTER UPDATE
ON grade FOR EACH ROW
BEGIN
UPDATE user SET grade_name='bignner'WHERE grade_id=6;
END$$
```

```
DELIMITER ;
```

3. **after delete trigger in any two tables**

```
DELIMITER $$
CREATE TRIGGER after_user_delete
AFTER DELETE
ON user FOR EACH ROW
BEGIN
DELETE user WHERE grade_id=6;
END$$
DELIMITER ;
```

Section6

1. **allowing created user all privileges**

```
CREATE USER 'habakubaho_fidele'@'%hafidele' IDENTIFIED VIA mysql_native_password
USING '***';GRANT ALL PRIVILEGES ON *.* TO 'habakubaho_fidele'@'%hafidele'
REQUIRE NONE WITH GRANT OPTION MAX_QUERIES_PER_HOUR 0
MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0
MAX_USER_CONNECTIONS 0;GRANT ALL PRIVILEGES ON `apms`.* TO
'habakubaho_fidele'@'%hafidele';
```

2. **allowing created user insert, delete and update privileges**

```
CREATE USER 'habakubaho_fidele_semi'@'%hafidele_semi' IDENTIFIED VIA
mysql_native_password USING '***';GRANT INSERT, UPDATE, DELETE ON *.* TO
'habakubaho_fidele_semi'@'%hafidele_semi' REQUIRE NONE WITH
MAX_QUERIES_PER_HOUR 0 MAX_CONNECTIONS_PER_HOUR 0
MAX_UPDATES_PER_HOUR 0 MAX_USER_CONNECTIONS 0;GRANT ALL
PRIVILEGES ON `apms`.* TO 'habakubaho_fidele_semi'@'%hafidele_semi';
```

3. **revoke insert privilege to above user**

```
REVOKE ALL PRIVILEGES ON *.* FROM 'habakubaho_fidele_semi'@'%hafidele_semi';
REVOKE GRANT OPTION ON *.* FROM 'habakubaho_fidele_semi'@'%hafidele_semi';
GRANT INSERT ON *.* TO
'habakubaho_fidele_semi'@'%hafidele_semi' REQUIRE NONE WITH MAX_QUERIES_PER_
HOUR 0 MAX_CONNECTIONS_PER_HOUR 0 MAX_UPDATES_PER_HOUR 0 MAX_US
ER_CONNECTIONS 0;
```

2.4 Conclusion:

By concluding on this database we have achieved more on this system using MySQL and the system in terms of keeping data is working well with different operation we have settled on it in some tables like revoking some activities, triggers are created in this database and relationship of much of these tables in this database are there to link them one by one. So this will lead us to creating the link between data store which is database and the final user of the system which will be focusing more on user interface and connectivity.

Reference:

1. **Six-Step Relational Database Design Second Edition (Fidel A Captain, 2016)**
2. **www.erbessd-instruments.com**

Chapter2: JAVA PROGRAMMING

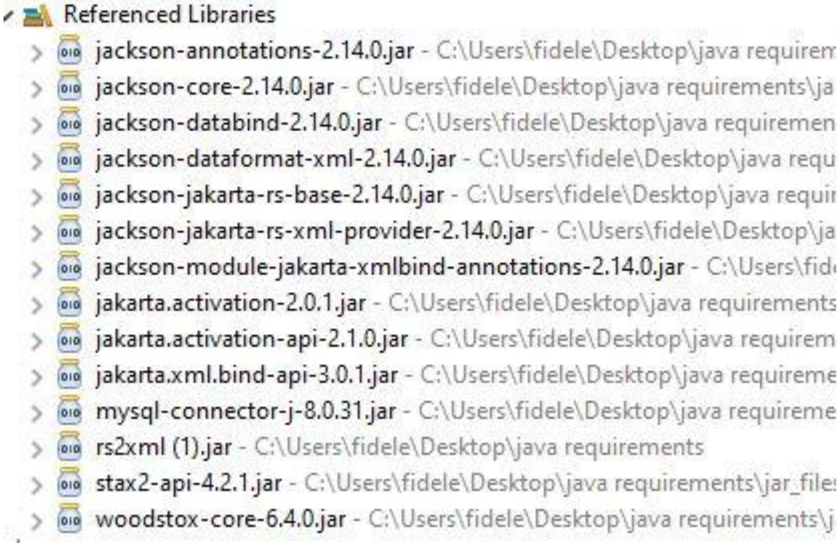
3.1 introduction

In this chapter I will be describing how powerful general-purpose **programming** language was used to create the analyzed system. Under this chapter I will undergo full detail of how everything will function together with database that have been describe above and how it cope with full analyzed system.

3.2 Tools used to develop this system in java programming:

Eclipse IDE: an integrated development environment used in computer programming. It contains a base workspace and an extensible plug-in system for customizing the environment. It is the second-most-popular IDE for Java development, and, until 2016, was the most popular.

JAR stands for Java Archive. It's a file format based on the popular ZIP file format and is used for aggregating many files into one. Although JAR can be used as a general archiving tool, the primary motivation for its development was so that Java applets and their requisite components.



```
Referenced Libraries
  jackson-annotations-2.14.0.jar - C:\Users\fidele\Desktop\java requirem
  jackson-core-2.14.0.jar - C:\Users\fidele\Desktop\java requirements\ja
  jackson-databind-2.14.0.jar - C:\Users\fidele\Desktop\java requiremen
  jackson-dataformat-xml-2.14.0.jar - C:\Users\fidele\Desktop\java requ
  jackson-jakarta-rs-base-2.14.0.jar - C:\Users\fidele\Desktop\java requi
  jackson-jakarta-rs-xml-provider-2.14.0.jar - C:\Users\fidele\Desktop\ja
  jackson-module-jakarta-xmlbind-annotations-2.14.0.jar - C:\Users\fid
  jakarta.activation-2.0.1.jar - C:\Users\fidele\Desktop\java requirements
  jakarta.activation-api-2.1.0.jar - C:\Users\fidele\Desktop\java requirem
  jakarta.xml.bind-api-3.0.1.jar - C:\Users\fidele\Desktop\java requireme
  mysql-connector-j-8.0.31.jar - C:\Users\fidele\Desktop\java requireme
  rs2xml (1).jar - C:\Users\fidele\Desktop\java requirements
  stax2-api-4.2.1.jar - C:\Users\fidele\Desktop\java requirements\jar_file
  woodstox-core-6.4.0.jar - C:\Users\fidele\Desktop\java requirements\j
```

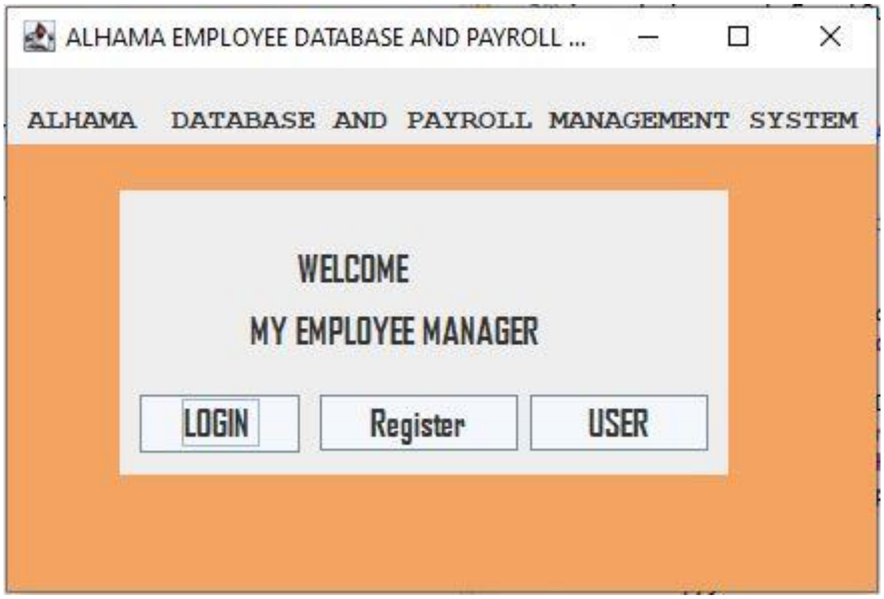
MySQL Connectors. **MySQL** provides standards-based drivers for JDBC, ODBC, and .Net enabling developers to build database applications in their language

3.3 Forms description

So let look together how the system will function one by one from the beginning up to the end.

1) Welcome page(APDMS):

this the first page of the system where system will choose destination depending on the function he/she possess in the company.


ALHAMA EMPLOYEE DATABASE AND PAYROLL ...
ALHAMA DATABASE AND PAYROLL MANAGEMENT SYSTEM
WELCOME
MY EMPLOYEE MANAGER
LOGIN
Register
USER

From this form 3 button login, Register and user are there with different direction and function in the system but all having same goal.

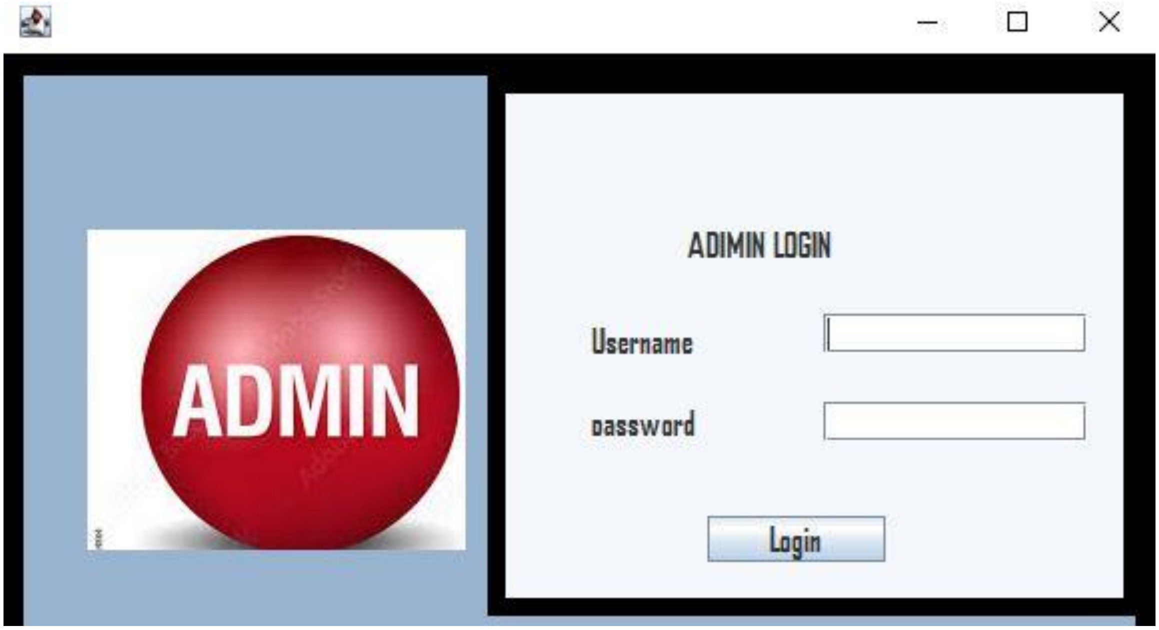
Login: this button is dedicated for system admin to login so that he/she can do more in the system, like adding employee or department and many more.

Register: individual employee can register on his/her own so this register button will direct this employee to form to fill his/her information.

User: button dedicated to other system user to login and perform some tasks that they are allowed to do so depending on the task they hold in the company.

2) ADMIN LOGIN:

As we have described in above page login form will direct as on this admin login page.

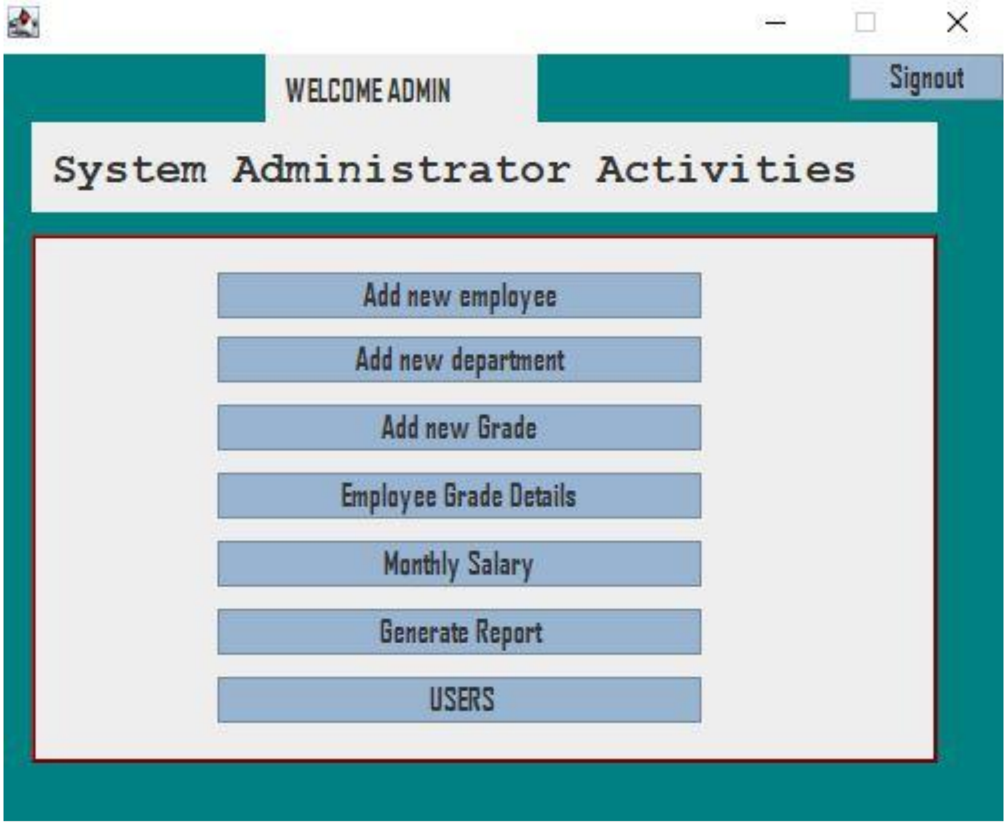


For admin to continue his/her tasks he will need to login using created and allowed username and password otherwise no access will be given.

3) Administrator home page:

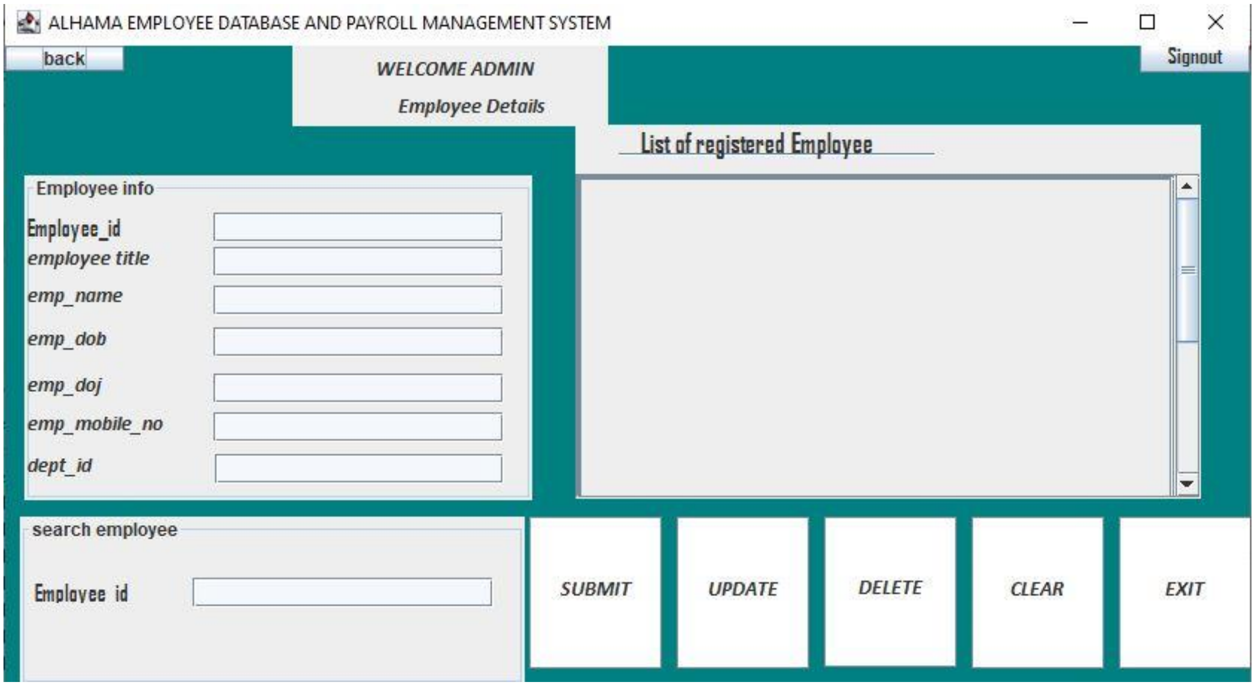
With this page we will be informed what tasks can system administrator do in this system like adding new employee for those employee who van not access the system or who do not have knowledge to use the system.

Adding new department, adding new system users or generating new grade as well as salaries of the employee will be the part to focus on this admin home page.



4) Employee information form:

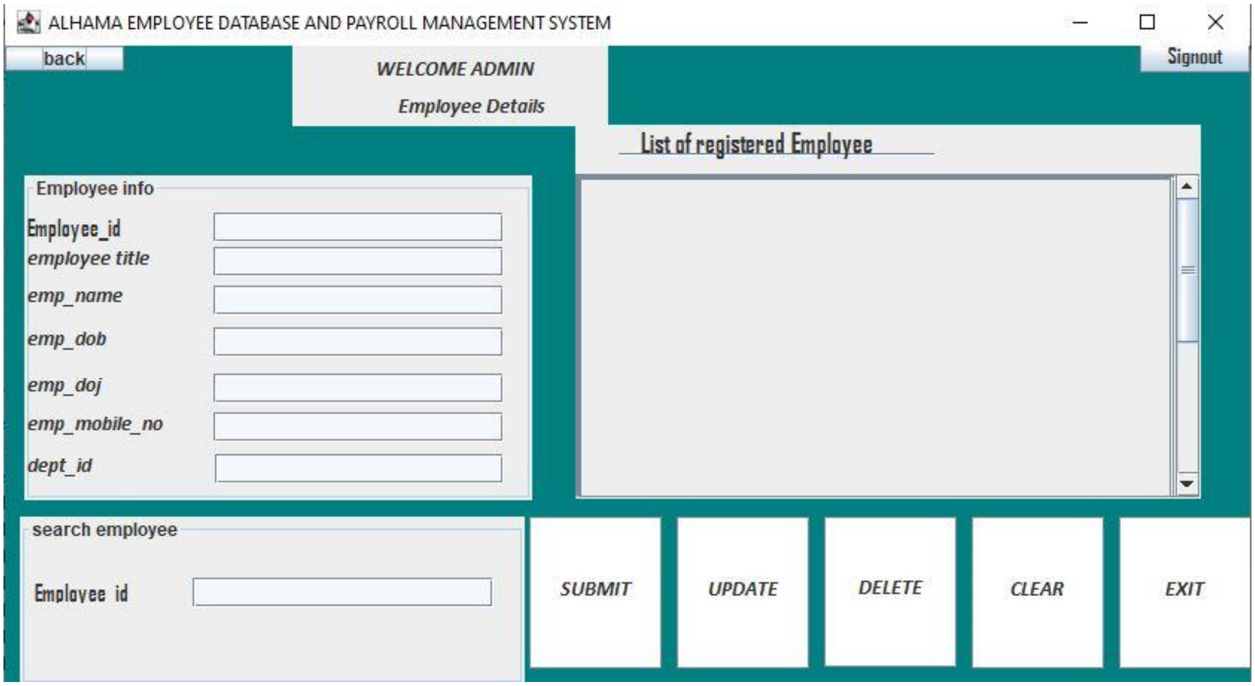
Under this page the system admin will add, delete, or even update employee table in the system using generated information addition on that admin can view list of all registered employee whether those who are registered by admin, other system allowed users or these registered by employee on their own.



5) Department page:

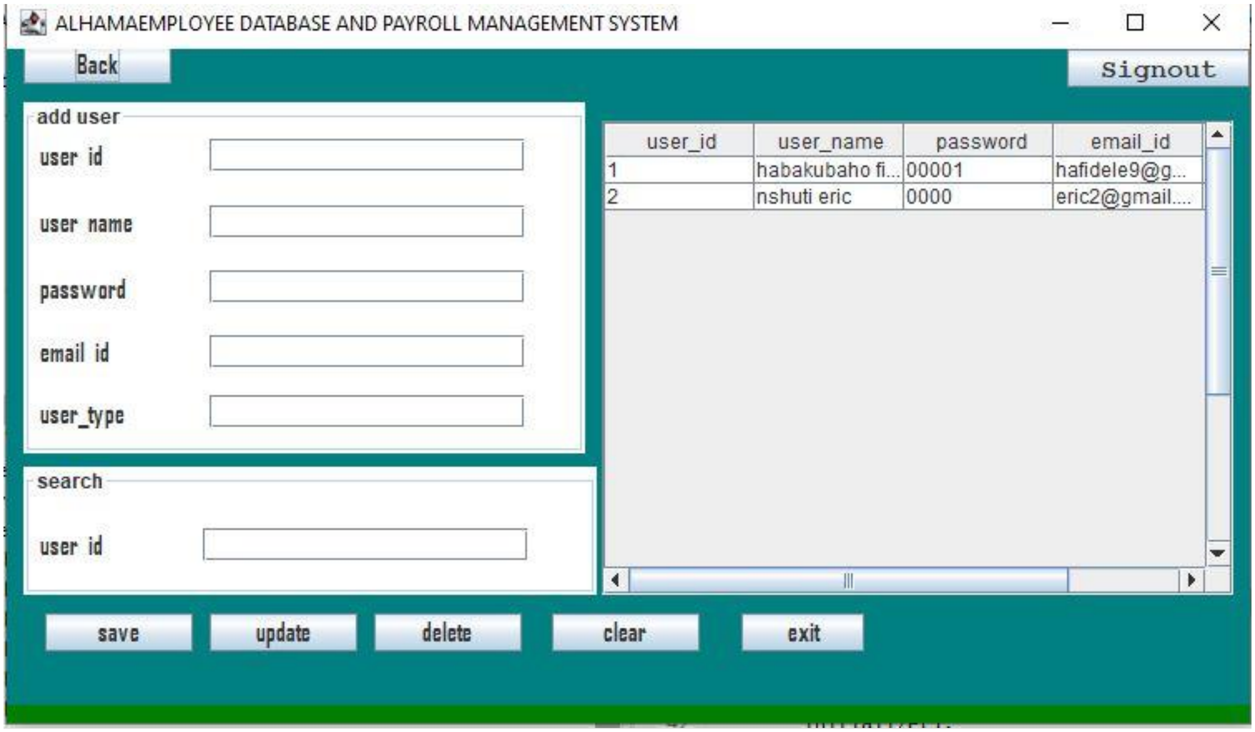
As company expand or need to operate in transparent manner may be needed to add the new department, this will be done by admin and other few allowed user who are in charge of coordination.

This page will generate what criteria that a department must possess to be called a department in the company.



6) New system user page:

We spent much time talking about new allowed user under this page we are going to know who create these users, what information to possess and what everyone will do after.



user_id	user_name	password	email_id
1	habakubaho fi...	00001	hafidele9@g...
2	nshuti eric	0000	eric2@gmail....

Under this system administrator will create new system user who can access the system by logging in using email_id and password that have been given, and they will access system with limited access depending on what function he/she possess in that company.

Under this page only admin can add or modify the information of the user in in this system so that the functioning of the system will continue.

Note that: the information entered from this page will go directly to be stored in database table called user.

7) Grade page:

This is a page where admin will generate which grade is allowed in company depending on criteria that make that grade to exist.

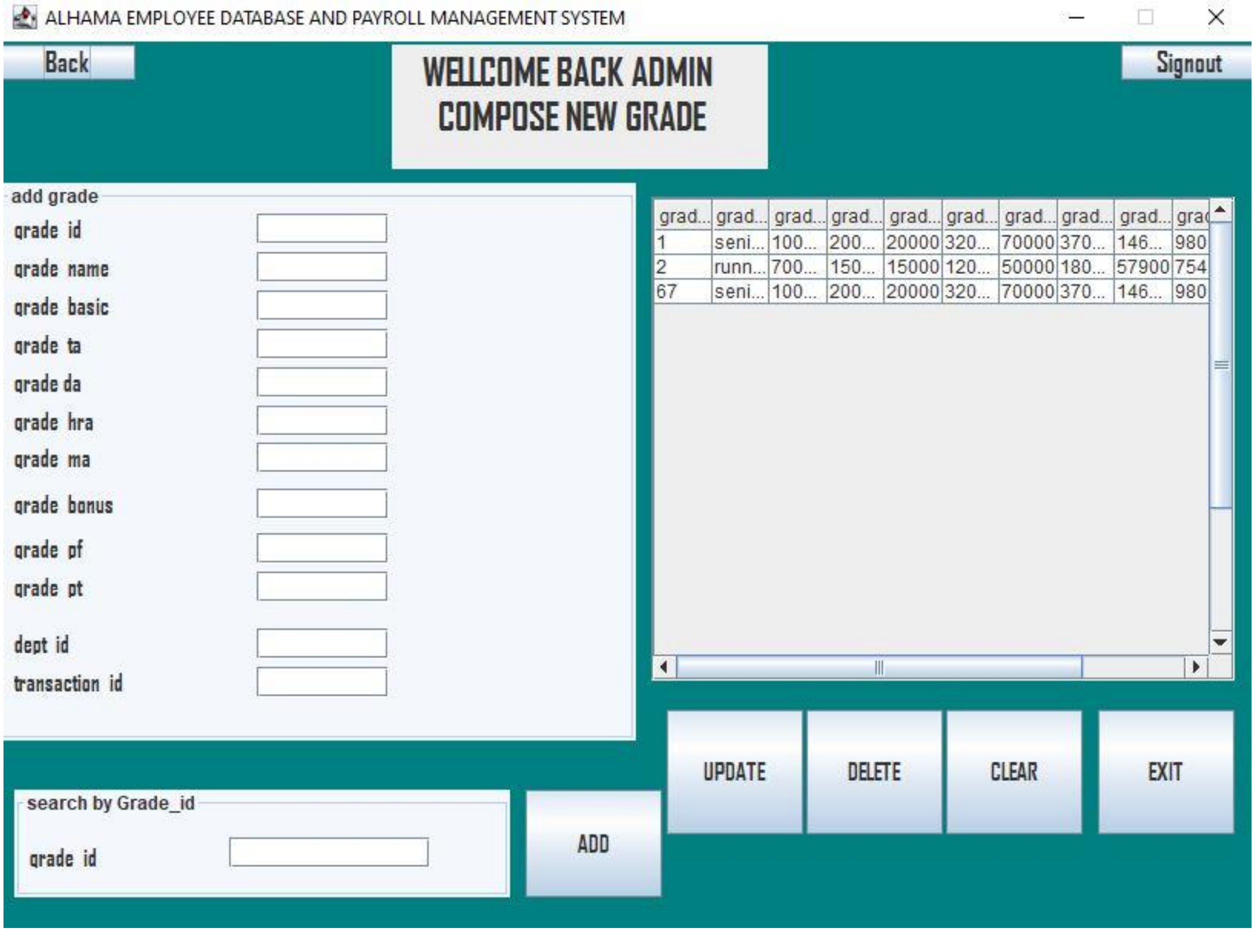
This grade page will show what to gain or to be deducted from employee depending on his/her grade.

This is important part of the company system because it will be focusing on company out follow as well as employee income from the company but holding together that contributing in country development is very important so they have to pay tax.

All allowances like dearness allowance, medical allowance, transport allowance and many more will be given to employee depending on grade he/she hold in the company.

Addition on that is that even deduction like tax and other will be deducted depending on that grade.

So this means that every grade has the certain percentage to use while deducting or giving bonus to employee basic salaries.



grad...	grad...	grad...	grad...	grad...	grad...	grad...	grad...	grad...	grad
1	seni...	100...	200...	20000	320...	70000	370...	146...	980
2	runn...	700...	150...	15000	120...	50000	180...	57900	754
67	seni...	100...	200...	20000	320...	70000	370...	146...	980

After admin have inserted new grade can load table from database called grade to view inserted or update information about grade.

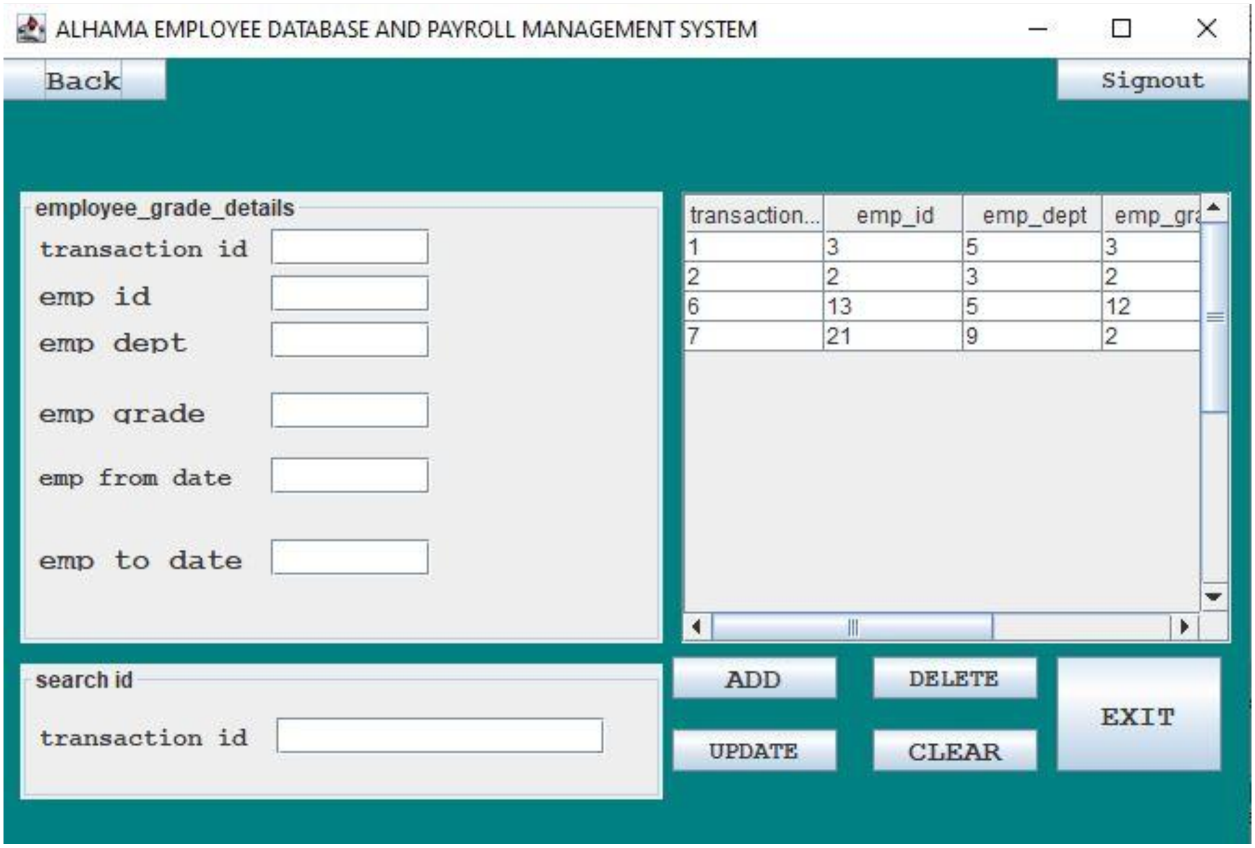
8) Employee grade details:

Every employee must have grade! But how everyone will be given that grade?

With this page we will be informed what criteria that employee must possess to belong in one grade.

To be in any given grade you need to be long company running worker thus this means that you must have been working so long or you are higher graded to belong in grade which make you get more reward but even get deducted more especially in terms of tax.

So at this time the difference between joining and current working time will generate the grade you are in.



transaction...	emp_id	emp_dept	emp_gra
1	3	5	3
2	2	3	2
6	13	5	12
7	21	9	2

9) Employee salary details page:

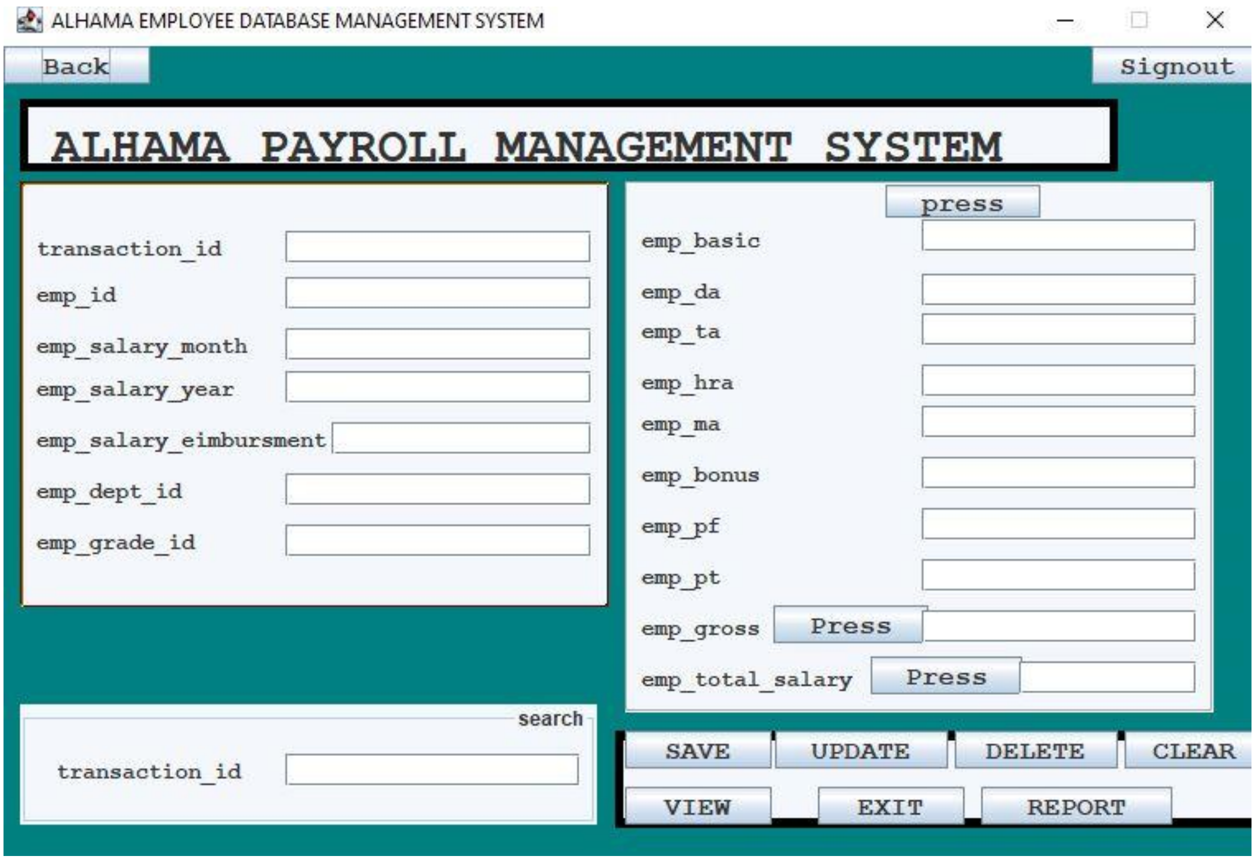
An important page for both company, employee and for the government as well.

It is important because it will generate what to get from the company to employee what expenses company spend in terms of paying employee as well as how much they will pay as tax.

At this page every out flow will be generated by the employee grade, thus this means what they need is to insert which grade employee hold and then press button to show all extra reward employee deserve and what deduction must face as well as what to get in his/her pocket after deduction.

This will follow strong logic of having certain percent on basic salary of the employee.

This page can be used by the financier to give report to company owner/shareholders so that they can audit that.

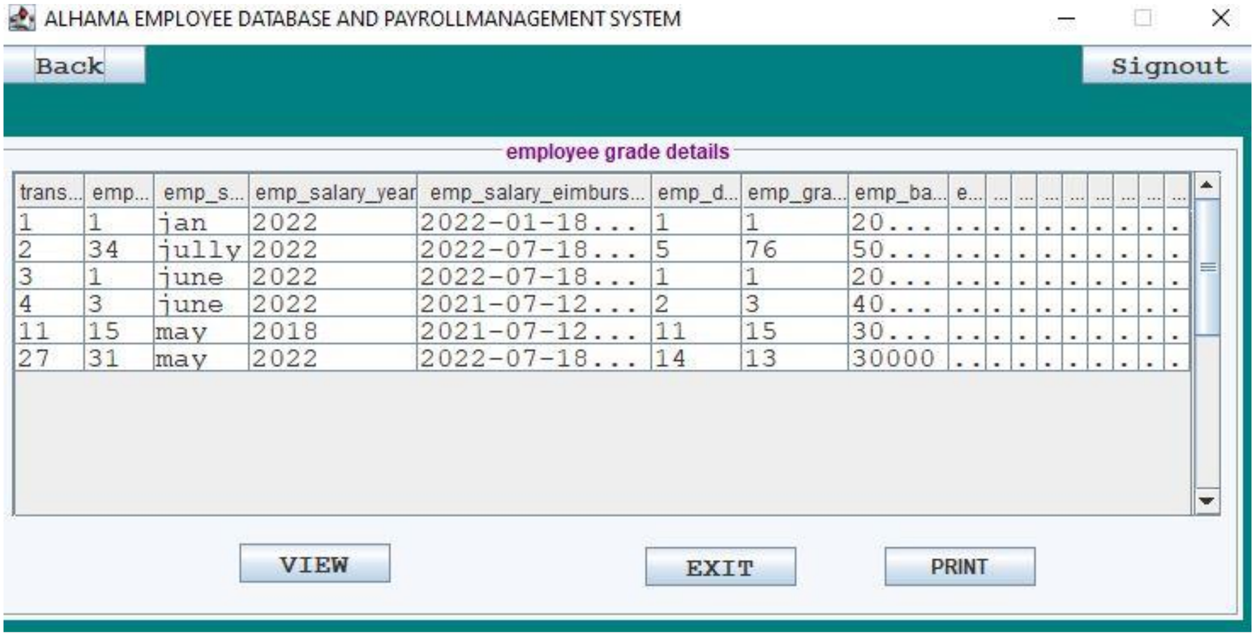


System admin can add more details of employee salaries and modify them as he/she need to.

Note that: the view button from this page will lead to checking list of details that are recorded in database.

10) View from employee salary details:

This is a page that show every individual employee salaries details in table that can be printed to be the evidence to different states like salary claiming, and can be presented to government or company owners.



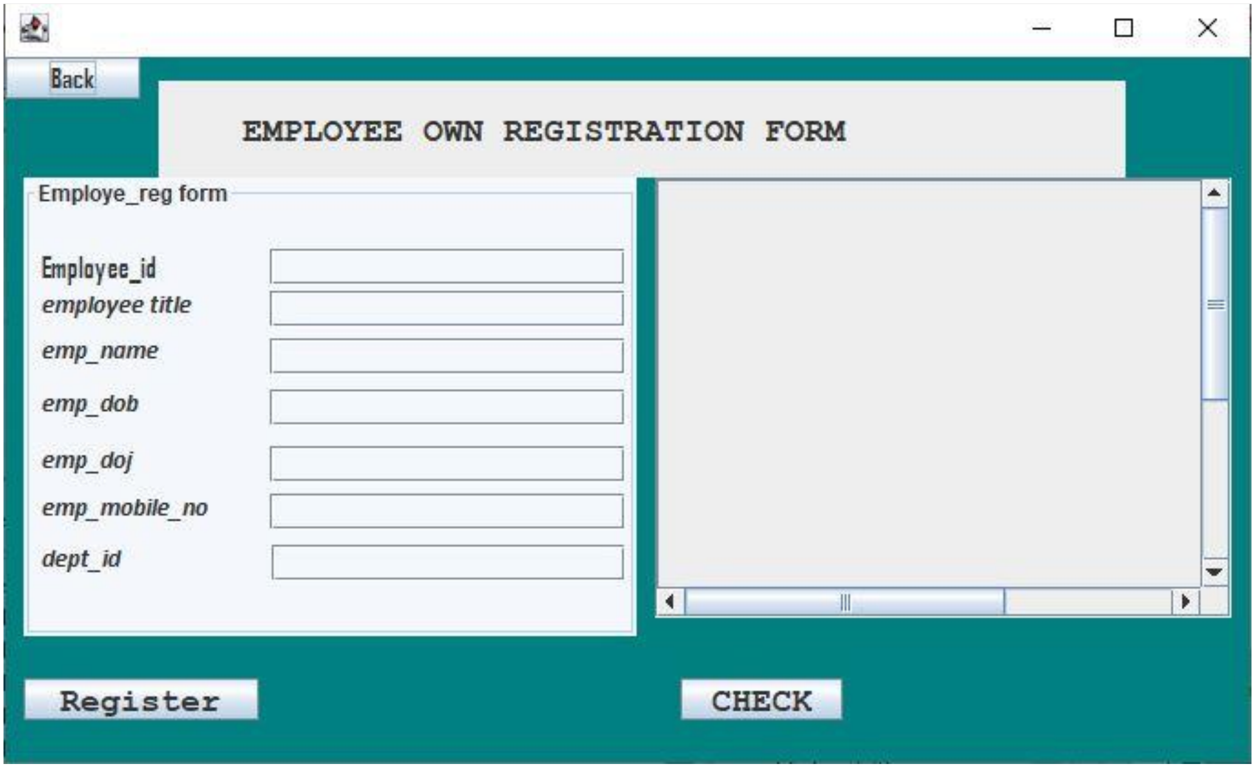
employee grade details

trans...	emp...	emp_s...	emp_salary_year	emp_salary_eimburs...	emp_d...	emp_gra...	emp_ba...	e...	...	...	...	...	...	...	...	...
1	1	jan	2022	2022-01-18...	1	1	20...	...	.	.	.	.	.	.	.	.
2	34	jully	2022	2022-07-18...	5	76	50...	...	.	.	.	.	.	.	.	.
3	1	june	2022	2022-07-18...	1	1	20...	...	.	.	.	.	.	.	.	.
4	3	june	2022	2021-07-12...	2	3	40...	...	.	.	.	.	.	.	.	.
11	15	may	2018	2021-07-12...	11	15	30...	...	.	.	.	.	.	.	.	.
27	31	may	2022	2022-07-18...	14	13	30000	...	.	.	.	.	.	.	.	.

This page will be accessed by every system user because it will be needed by the employee in different department in case of claiming so they do not all need to go to admin or financier directly they will need to approach closer system user to check for them.

11)Employee own register page:

Under this page employee will be allowed to register and view if he/she is registered well, without waiting for system user/admin to come and register for him/her. The only difference here is that employee cannot modify anything if he or she need to modify he/she will contact admin or system user to do it.

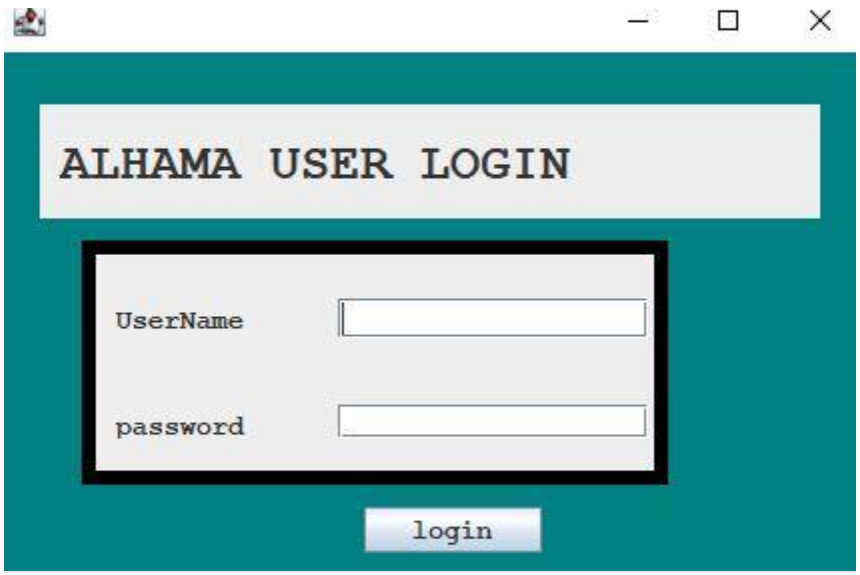


This is page that will be accessed while register button from the welcome page is pressed.

12)User login page:

This the page on which new created user will use given username and password that are recorded in database called in table called users to login so that he/she can access the system features he/she is allowed to.

Every user must have access on system depending his/her type, thus this mean that if you are HRM you have to have access referring to employee recruitment and hiring not procurement. User type from user table is the key element to give user on system access.



3.4 Special buttons:

Back button:

this is button where pressing on it bring you to the previous page you were on.

Sign out button:

At this button you will get out of the system so that you will be required to enter login creditantial to enter again.

Print button:

Button used to direct you to print the information from the system.

3.5 Conclusion:

By concluding this chapter concerns with java programming especially in my developed system, we can say that I have final product that I was expecting to have it, the manipulation of data is going well the design is there with special appearance, but there much to go on and that need to be improved will be gained from external view apart from system developer.

Reference:

1. Java Performance: The Definite Guide First Edition (Scott Oaks, 2017)
2. www.eclipse.org>windowbuilider
3. www.vogella.com>article
4. www.formdev.com>ides>eclipse