

**PAYROLL SYSTEM**

**DOCUMENTATION**

**CSCC 20- CCA**

Submittded by

Joseph Chaves

Christian Mabao

Submitted to

Jessie Lagrosas

CSCC 20 Instructor

Submission Date

4th October, 2019

**TABLE OF CONTENTS**

**Introduction - - - - - - - - - - - - - - - - - I**

**REQUIREMENTS ANALYSIS - - - - - - - - - - - - - II**

**USES-CASES ANALYSIS - - - - - - - - - - - - - - III**

**CLASS DIAGRAM - - - - - - - - - - - - - - - -IV**

**SEQUENCE DIAGRAM - - - - - - - - - - - - - - -V**

**API - - - - - - - - - - - - - - - - - - - - - VI**

**APPENDIX - - - - - - - - - - - - - - - - - - vii**

1. **INTRODUCTION**

Applying object oriented programming. Object-oriented programming is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields, and code. Object Oriented Programming course final project was to create a program that applies object oriented paradigm. Through learning the basics of OOP, we students are tasked to create a system/program that applies object oriented programming. Creating payroll system is our choice as buddies and we make its documents.



**PAYROLL SYSTEM**

**REQUIREMENT ANALYSIS**

1. **REQUIREMENTS ANALYSIS**

**Project Name: Payroll System**

This payroll system involves thing that has to do with the payment of employees. This includes keeping track of hours, calculating [wages](https://money.howstuffworks.com/wage.htm), withholding [deductions](https://money.howstuffworks.com/personal-finance/personal-income-taxes/tax-deductions.htm), printing, and viewing wages/attendance. At the end of the month, a payroll department (employer) uses the payroll system to take all of the wage (withholding deductions) information from the current month.

**Functional Requirements:**

* Admin can add employee with its wage rate per day.
* Admin can delete employee.
* Admin can generate monthly payroll for all employees.
* Employee’s daily attendance (Time-in/Time-out).
* Checks employee’s current attendance/payroll.

**Non-Functional Requirement:**

Deductions to employee’s monthly salary:

* Tardiness penalty
* Absent penalty
* Half-day penalty

Additional amount to employee’s monthly salary:

* Overtime
* Awards/Incentives



**PAYROLL SYSTEM**

**USE-CASES ANALYSIS**

1. **USES-CASES ANALYSIS**

**USE CASES:**

1. Time in

* Actor – employee
* Pre-condition - Log in
* Success scenario:
  + After log in, employee need to input time of arrival.
  + Show success message.

**2. Time out**

* Actor – employee
* Pre-condition - Log in
* Success scenario:
  + After log in, employee need to input time of in work.
  + Show success message.

**3. Check Employee Standings**

* Actor – employee
* Pre-condition - Log in
* Success scenario:
  + After log in, employee will select standings in work (feature).
  + Ask employee to select from: attendance record or current wage record.
  + System will show selected record.

**4. Add Employee**

* Actor - admin
* Pre-condition - Log in
* Success scenario:
  + After log in, admin will select add employee (feature).
  + Ask admin to setup employee profile.
  + System will confirm employee added.

**5. Delete Employee**

* Actor – admin
* Pre-condition - Log in
* Success scenario:
  + After log in, admin will select delete employee (feature).
  + Ask admin who to delete.
  + Show warning message.
  + If confirmed delete employee, show success message
* Extension – If admin will not confirmed deletion, exit selected feature (delete employee).

**5. Generate Monthly Payroll**

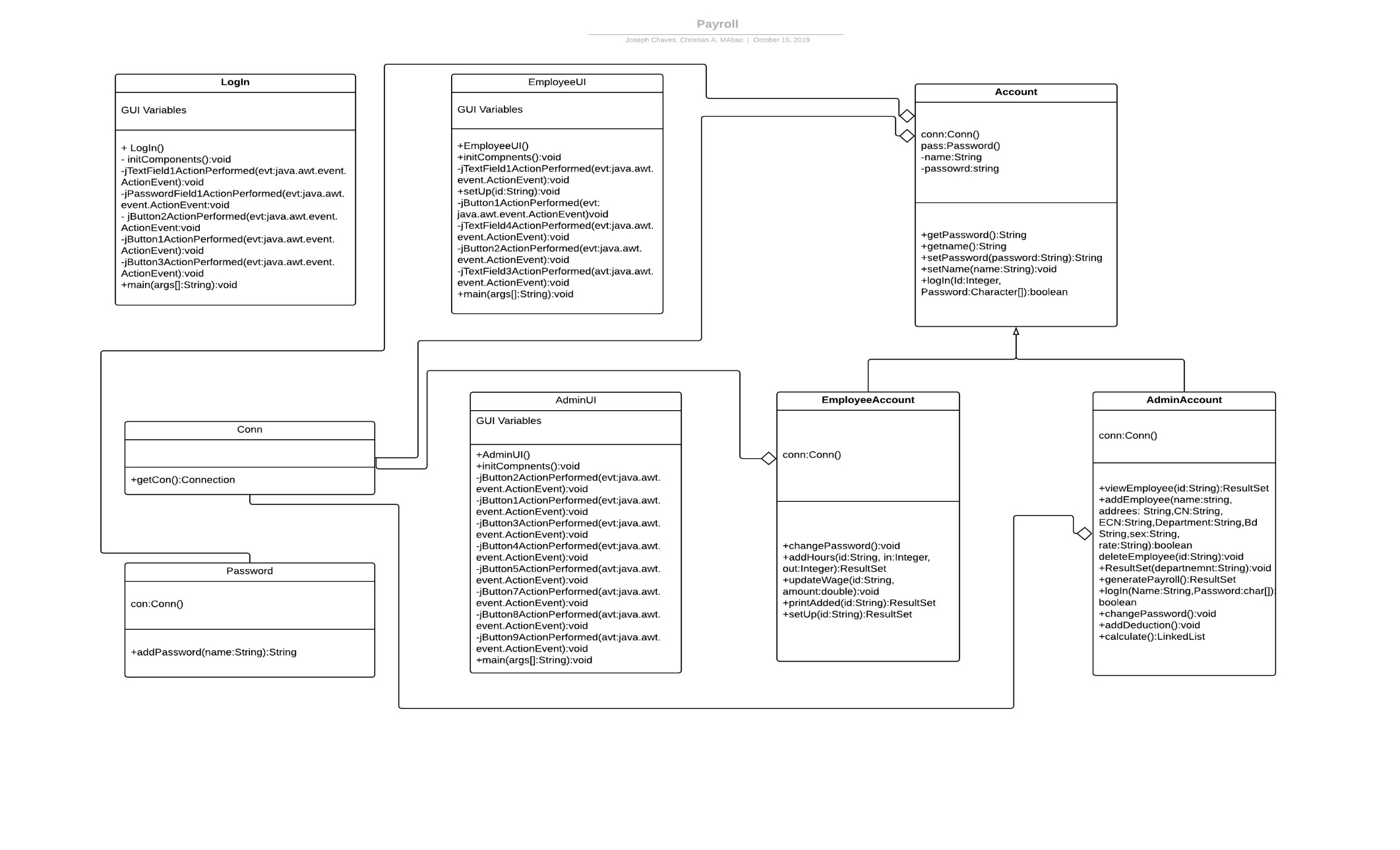
* Actor – admin
* Pre-condition - Log in
* Success scenario:
  + After log in, admin will select generate monthly payroll (feature).
  + Show payroll generated.



**PAYROLL SYSTEM**

**CLASS DIAGRAM ANALYSIS**

1. **CLASS DIAGRAM**

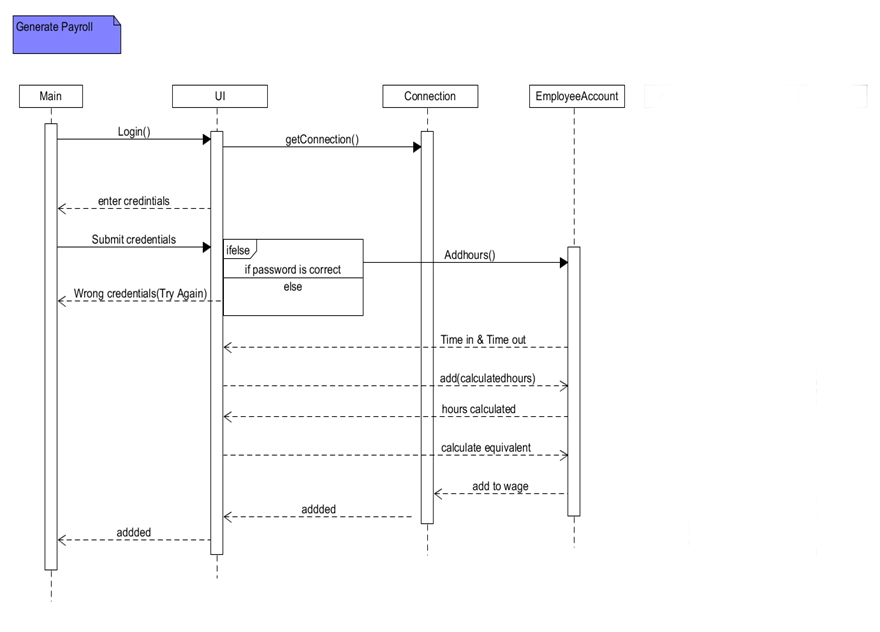
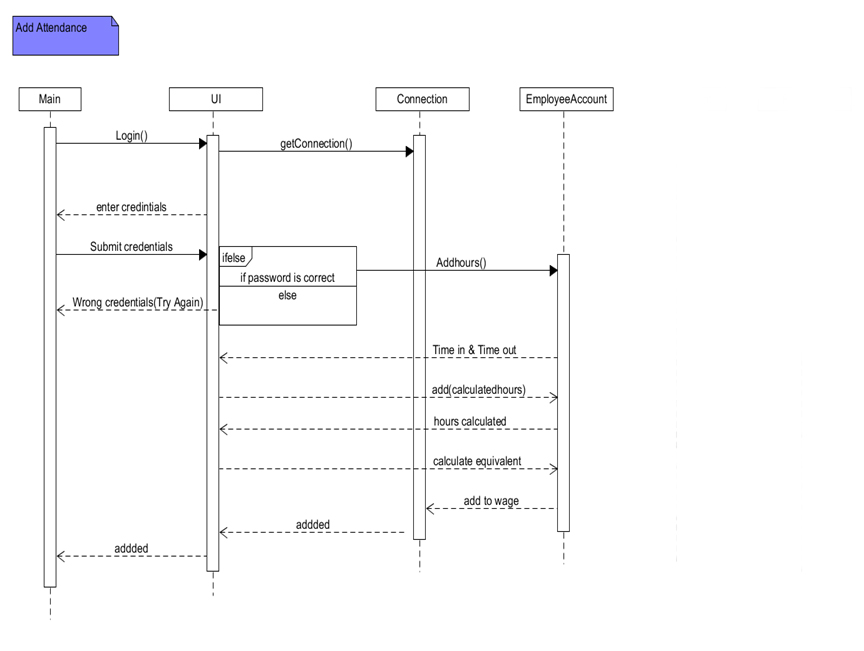
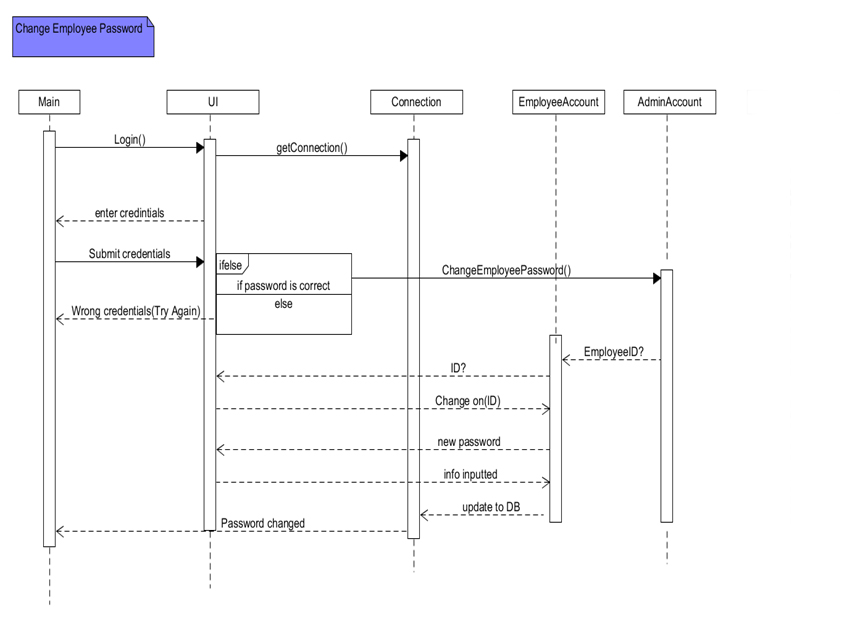
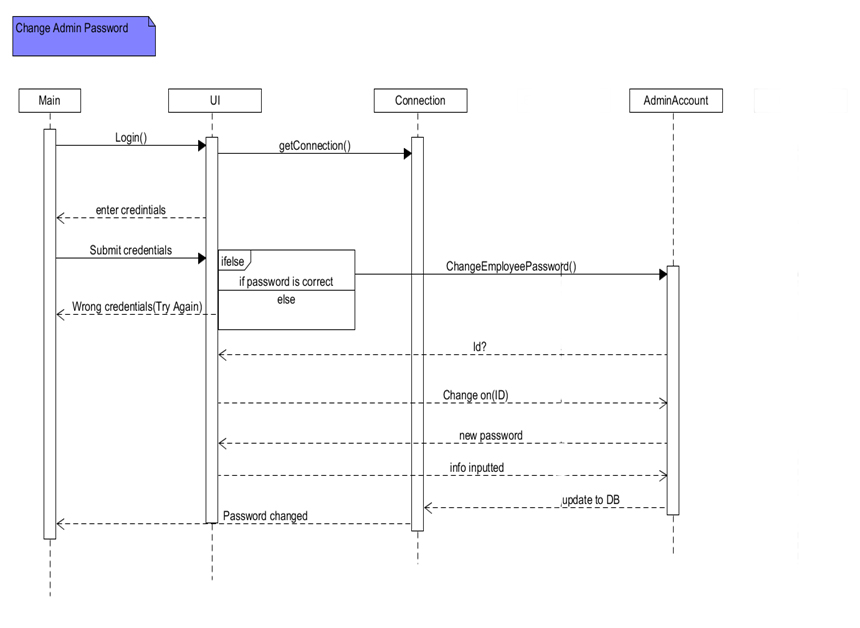
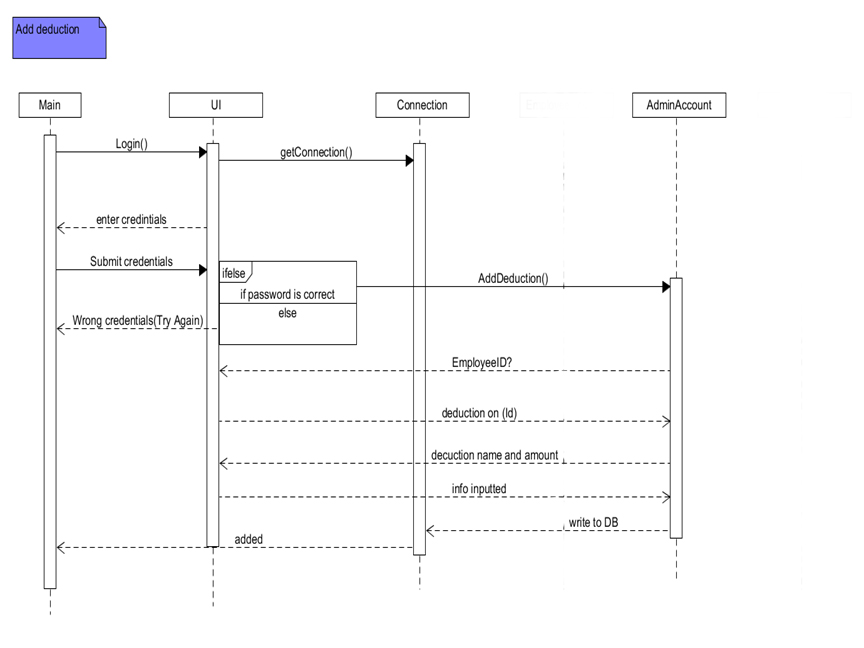
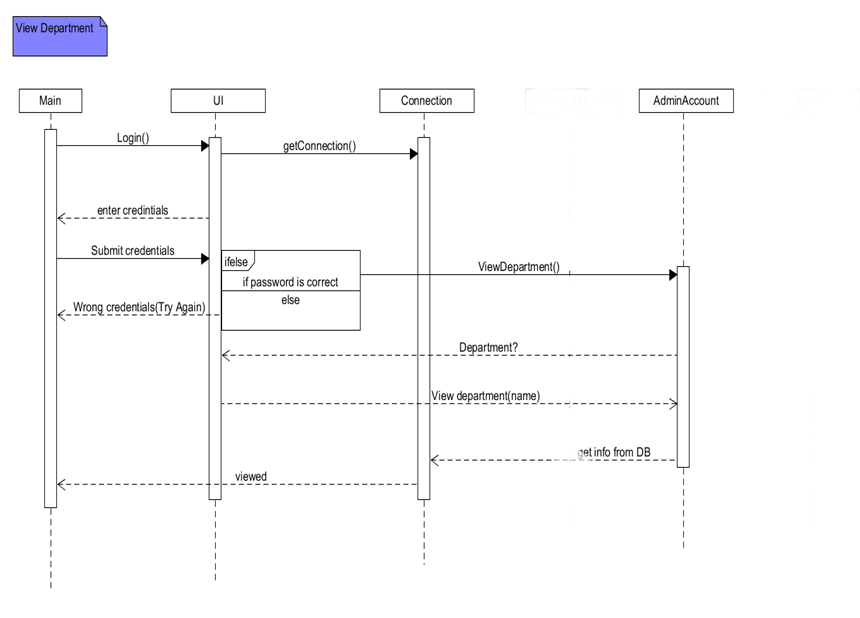
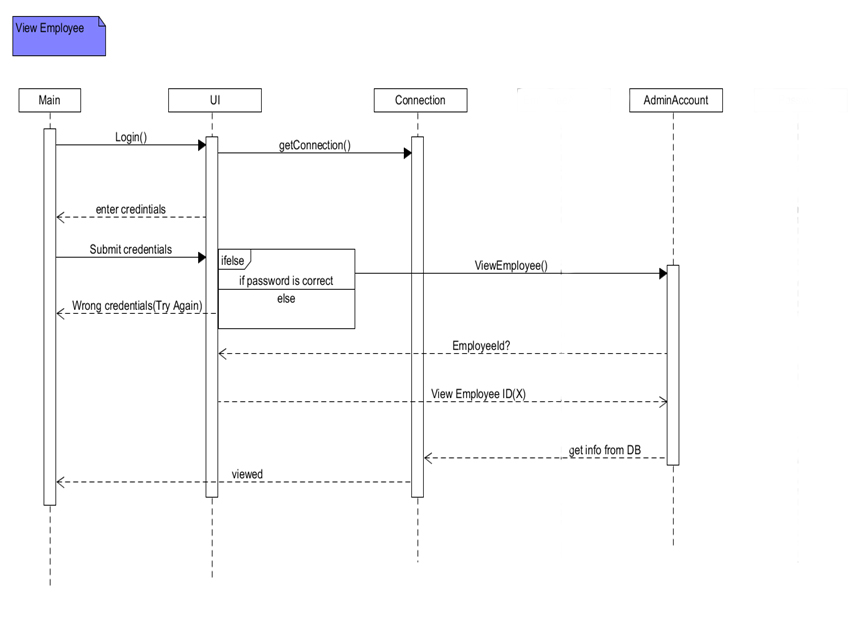
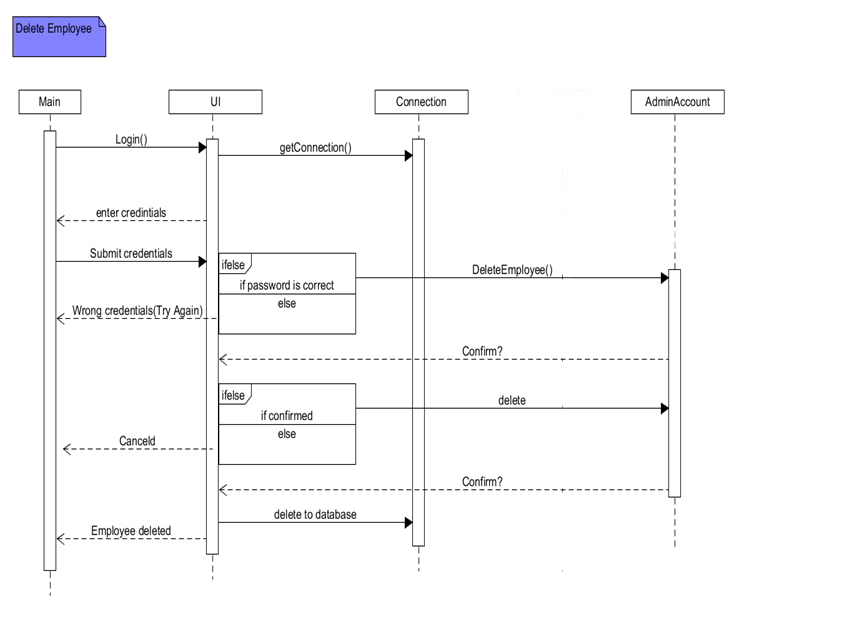
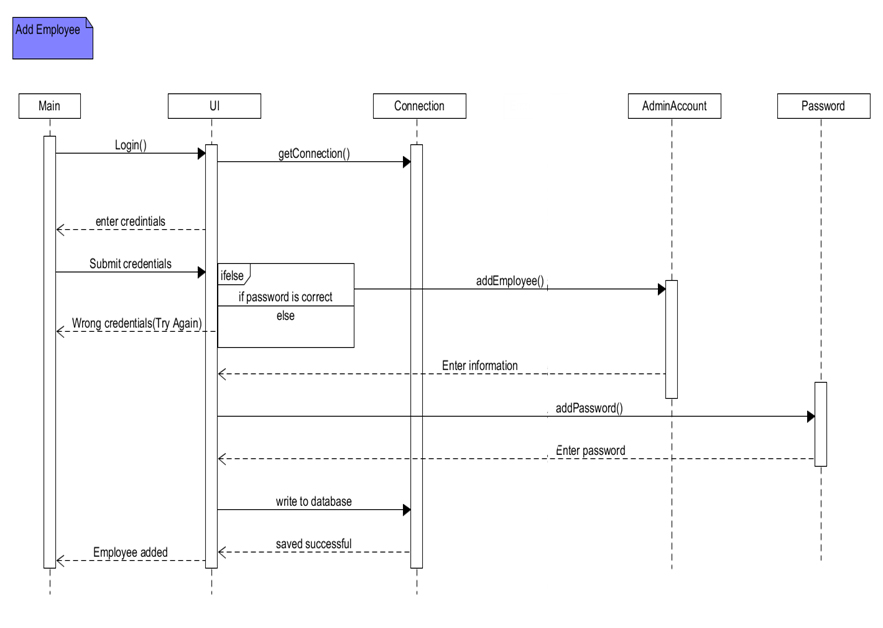
****



**PAYROLL SYSTEM**

**SEQUENCE DIAGRAM**

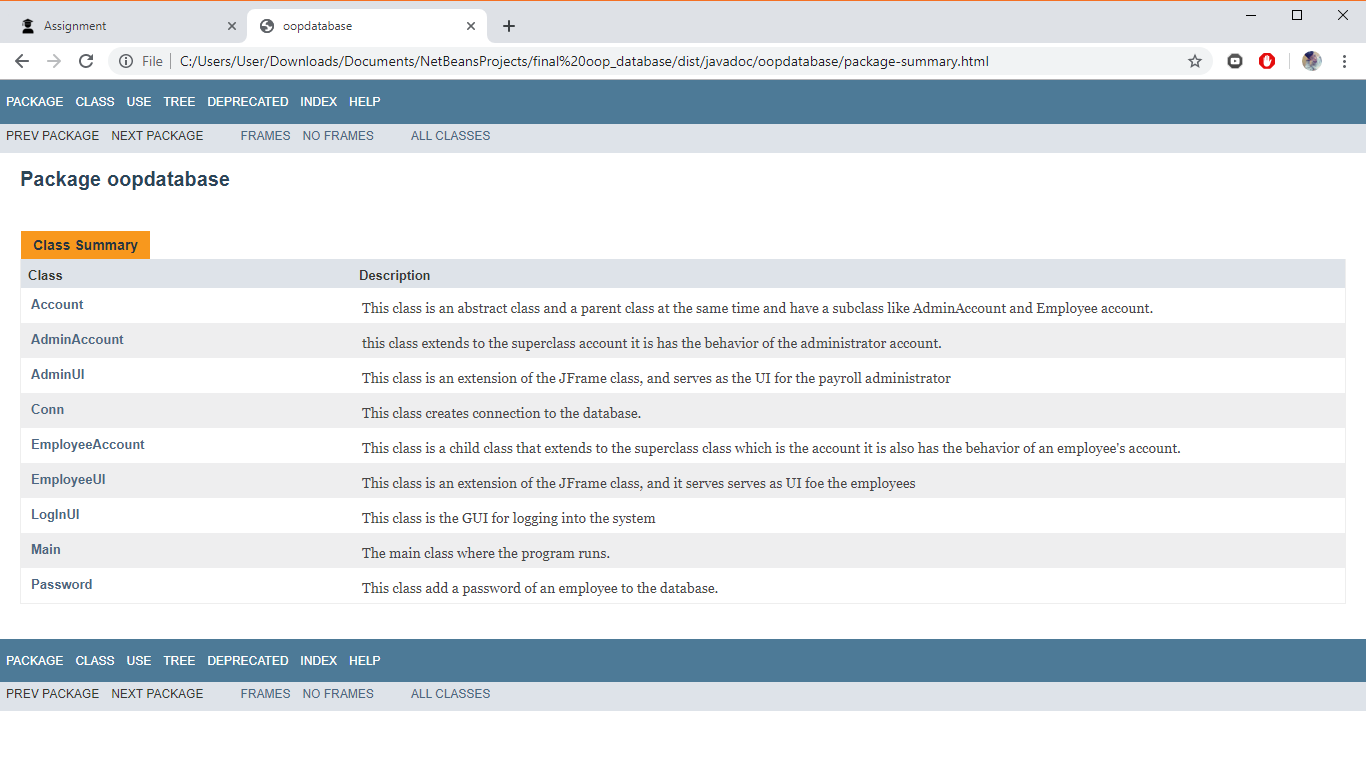
**SEQUENCE DIAGRAM**





**PAYROLL SYSTEM**

**APPLICATION PROGRAMMING INTERFACE**



**NOTE: FULL API IS ATTACHED IN THE ZIP FILE**



PAYROLL SYSTEM

APPENDIX

**MILESTONE 1**

Case Description: Payroll System

Joseph P. Chaves

Christian A. Mabao

General Description:

This payroll system involves thing that has to do with the monthly payment of employees. This includes keeping track of hours, calculating wages, withholding deductions, printing, and viewing wages/attendance. At the end of the month, a payroll department (employer) uses the payroll system to take all of the wages (withholding deductions) information from the current month.

Basic Operations:

In a payroll system it is needed to have a payroll admin and employees to be paid. The admin of the payroll system add employees through getting the employee’s basic information and his/her hourly rate. Admin can also delete employee through manipulating the system by removing data of the specific employee. Employees will check its daily attendance on the system. It can only check time-in and time-out and it is the way to generate the employees’ monthly payroll. Employees can also check their current attendance record and their current payroll.

It is also part of the system’s scope to generate monthly payroll for all the employees, through checking the employees’ worked hours (including overtime) with deduction (taxes, SSS, etc.).

Information Needs:

All data relating to employee, employee’s hourly rate, employee’s attendance, and wage deductions. Employee are responsible for checking their respective daily attendance used to generate worked hours. They can also view their respective tentative bills for the current month. The final wages of all employees are generated from the payroll section of the company by the payroll admin every month. To generate the monthly payroll the following reports are needed:

1. Employee X in Department Y: Employee ID, Employee Name, Employee Address, Employee Phone, Birthdate, Emergency Contact Number, Department, Hourly Rate.

This report is to acquire basic information of the employee to generate his/her wage.

2. Attendance of Employee X: Employee ID, Date, Regular Hours Worked, Non-Regular Hours Worked.

To generate the wage of the said employee, total number of worked hours both in regular and non-regular days, through checking the employee’s attendance.

3. Deductions of Employee X wage: Employee ID, SSS ID, Tax, Others.

After calculating the number of hours worked and multiplying it with the employee’s hourly rate, its wage is deducted by taxes, SSS, and others.

**MILESTONE 2**

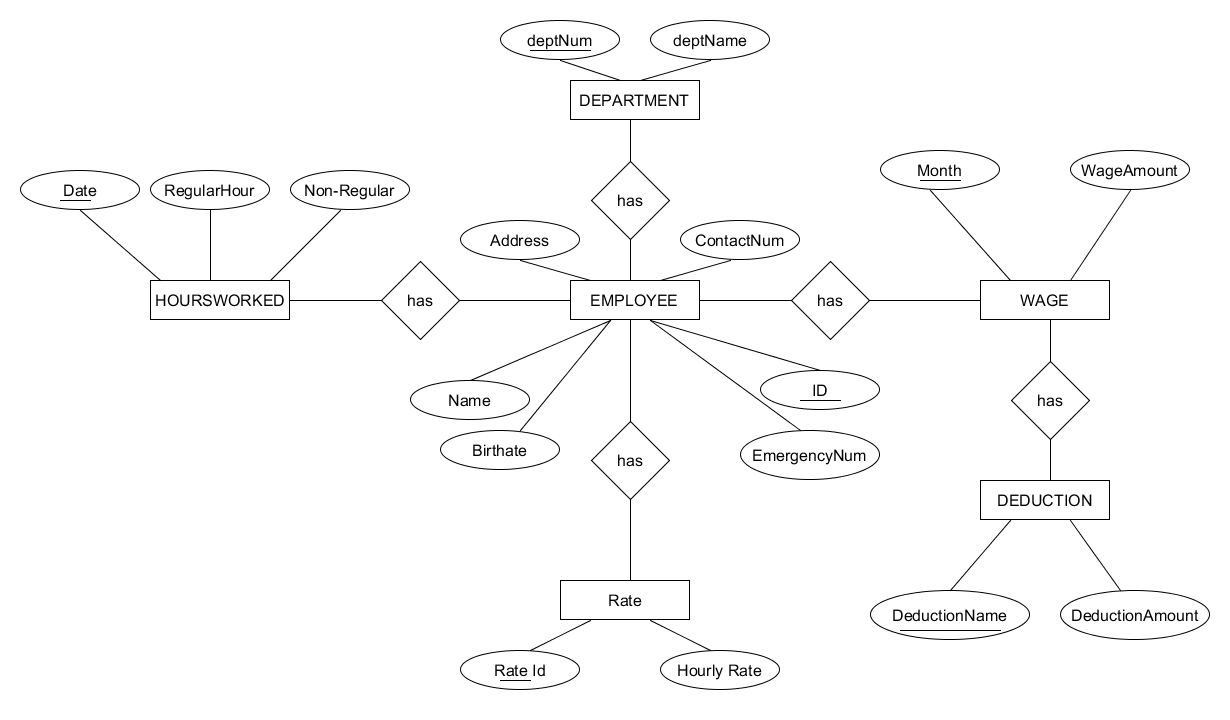
**Payroll System Database:**

**Users’ Requirement Specification**

The **Payroll** database keeps track of a company’s employees, employees’ hours-worked, and deduction of employees’ wages. This information being track is later use to generate the employee’s respective wage every month.

The company is organized into departments. Each department has a unique department number, and name. We keep track of the start **employee’s** **number of hours worked** in every department.

We store each Employee’s ID, Employee Name, Employee Address, Employee Phone, Birthdate, Emergency Contact Number, Department, and Hourly Rate. An employee is assigned to one department. We keep track of the number of hours worked by an employee daily. We want to keep track also the amount of deductions can be subtracted to the employee’s. With these information the database would be able to generate monthly payroll for the company.

**Entity-Relationship Diagram**

**LOGICAL DATA MODEL**

**DEPARTMENT**

|  |  |
| --- | --- |
| **DeptNum** | DeptName |

**EMPLOYEE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Name | Address | ContactNum | EmgergencyNum | Birthdate | **DeptNum** | **RateId** |

**RATE**

|  |  |
| --- | --- |
| **RateID** | HourlyRate |

**HOURS\_WORK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | RegularHours | Non-RegularHours | **EmployeeID** |

**WAGE**

|  |  |  |
| --- | --- | --- |
| **Month** | WageAmount | **EmployeeID** |

**DECUCTIONS**

|  |  |  |
| --- | --- | --- |
| **DeductionName** | Amount | **Month** |

**ADMIN**

|  |  |  |
| --- | --- | --- |
| **AdminID** | Name | Password |

**PASSWORD**

|  |  |
| --- | --- |
| **EmployeeID** | Password |

**Data Dictionary**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| TABLE NAME | ATTRIBUTE NAME | CONTENTS | TYPE | FORMAT | RANGE | REQUIRED | PK  OR  FK | FK REFERENCED TABLE |
| DEPARTMENT | Dept\_Num  Dept\_Name | Department number  Department name | CHAR(5)  CHAR(20) | 99999  Xxxxxxxx | 10000-99999 | Y  Y | PK |  |
| EMPLOYEE | Id  Name  Address  ContactNum  EmergencyNum  Birthdate  Dept Num  Rate Id | Employee id  Employee name  Employee Address  Employee Contact  Employee emergency number  Employee birthdate  Employee department number  Employee rate id | CHAR(5)  CHAR(20)  CHAR(20)  CHAR(10)  CHAR(10)  DATE  CHAR(5)  CHAR(5) | 99999  Xxxxxxxx  Xxxxxxxx  Xxxxxxxx  Xxxxxxxx  dd-mmm-yyy  99999  99999  99999 | 10000-99999  10000-99999 | Y  Y  Y  Y  Y  Y  Y  Y  Y | PK  FK  FK | DEPARTMENT  RATE |
| RATE | Rate\_Id  Hourly\_Rate | Rate id  Hourly Rate | CHAR(5)  DOUBLE | 99999  999.99 | 10000-99999 | Y  Y | PK |  |
| HOURS\_WORK | Date  Regular\_Hours  Non-RegularHours  EmployeeId | Date of working  Regular hours worked  Non-regular hours worked  Employee id | DATE  SMALLINT  SMALLINT  CHAR(5) | dd-mmm-yyy  22222  22222  99999 |  | Y  Y  Y  Y | PK  FK | EMPLOYEE |
| WAGE | Month  WageAmount  ID | Month wage given  Amount of wage  Employee ID | VCHAR(10)  INT  CHAR(5) | Xxxxxxxxx  1000000  99999 |  | Y  Y  Y | PK  FK | EMPLOYEE |
| DEDUCTIONS | DeductionName  Amount  Month | Name of deduction  Deduction amount  Month wage given | CHAR(20)  INT  VCHAR(10) | Xxxxxxx  1000000  Xxxxxxxx |  | Y  Y  Y | PK  FK |  |

**MILESTONE 3**

**LOGICAL DATA MODEL**

**DEPARTMENT**

|  |  |
| --- | --- |
| **DeptNum** | DeptName |

**EMPLOYEE**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **ID** | Name | Address | ContactNum | EmgergencyNum | Birthdate | **DeptNum** | **RateId** |

**RATE**

|  |  |
| --- | --- |
| **RateID** | HourlyRate |

**HOURS\_WORK**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | RegularHours | Non-RegularHours | **EmployeeID** |

**WAGE**

|  |  |  |
| --- | --- | --- |
| **Month** | WageAmount | **EmployeeID** |

**DECUCTIONS**

|  |  |  |
| --- | --- | --- |
| **DeductionName** | Amount | **Month** |

**ADMIN**

|  |  |  |
| --- | --- | --- |
| **AdminID** | Name | Password |

**PASSWORD**

|  |  |
| --- | --- |
| **EmployeeID** | Password |

**Physical Database Design**

department (

DeptNum varchar(5) NOT NULL default '',

DeptName varchar(20) NOT NULL default '',

PRIMARY KEY (DeptNum),

)

employee (

EmployeeId varchar(5) NOT NULL default '',

Name varchar(20) NOT NULL default '',

Address varchar(20) NOT NULL default '',

ContactNum varchar(10) NOT NULL default '',

EmergencyNum varchar(10) NOT NULL default '',

Birthdate Date NOT NULL default '',

DeptNum varchar(5) NOT NULL default '',

RateId varchar(5) NOT NULL default '',

PRIMARY KEY (dnumber,dlocation),

FOREIGN KEY (DeptNum) REFERENCES department(DeptNum),

FOREIGN KEY (RateId) REFERENCES rate(RateId),

)

rate (

RateId varchar(5) NOT NULL default '',

HourlyRate varchar(20) NOT NULL default '',

PRIMARY KEY (dnumber,dlocation),

)

HoursWork (

Date DATE NOT NULL default '',

RegularHours Smallint NOT NULL default '0',

NonRegularHours Smaillint NOT NULL default '0',

PRIMARY KEY (Date),

FOREIGN KEY (EmployeeId) REFERENCES employee(EmployeeId),

)

wage (

Month Smallint NOT NULL default '0',

NonRegularHours Smallint NOT NULL default '0',

EmployeeId varchar(5) NOT NULL default '',

PRIMARY KEY (Month),

FOREIGN KEY (EmployeeId) REFERENCES employee(EmployeeId),

)

password (

EmployeeId varchar(5) NOT NULL default '',

password varchar(20) NOT NULL default '0',

PRIMARY KEY (EmployeeId),

FOREIGN KEY (EmployeeId) REFERENCES employee(EmployeeId),

)

admin (

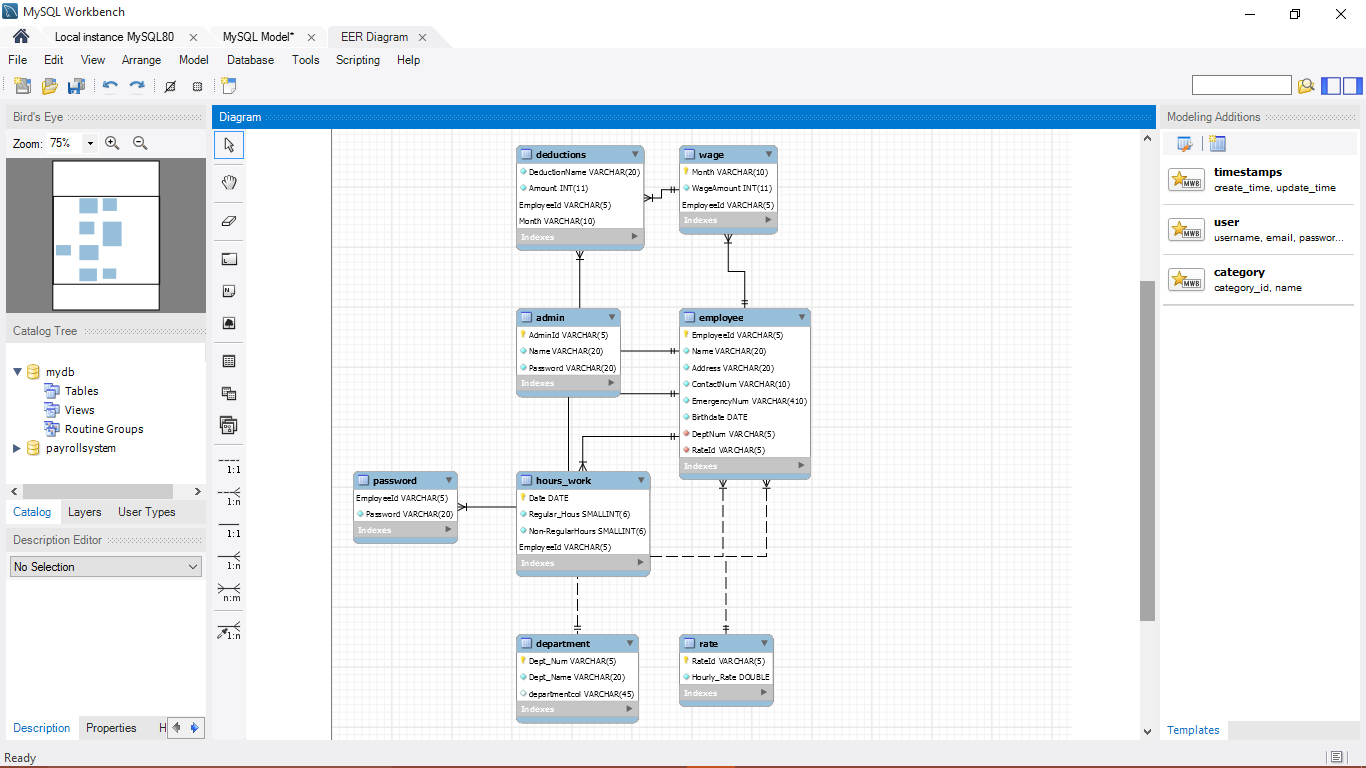
AdminId varchar(5) NOT NULL default '',

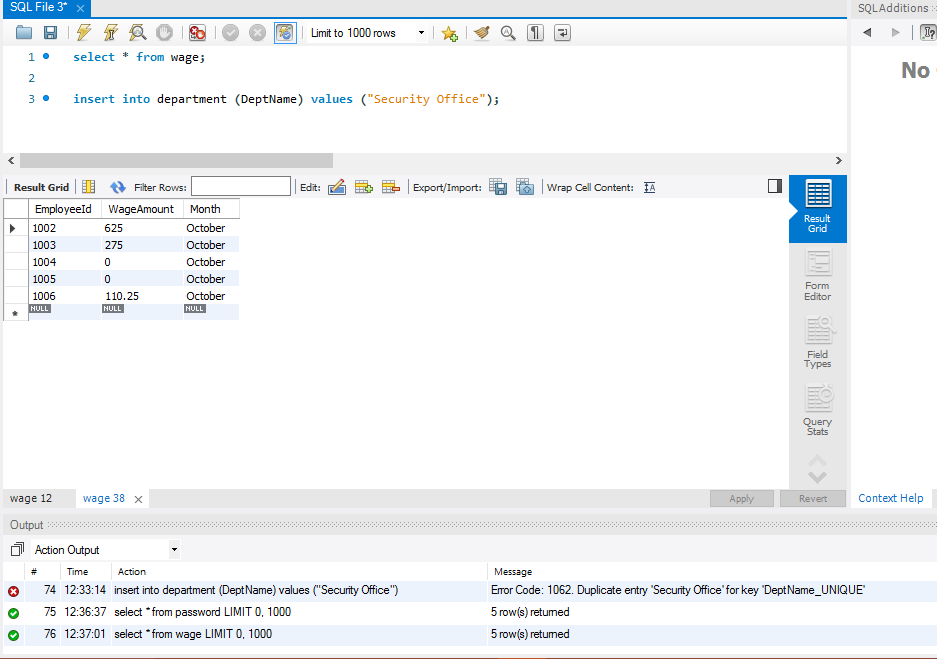
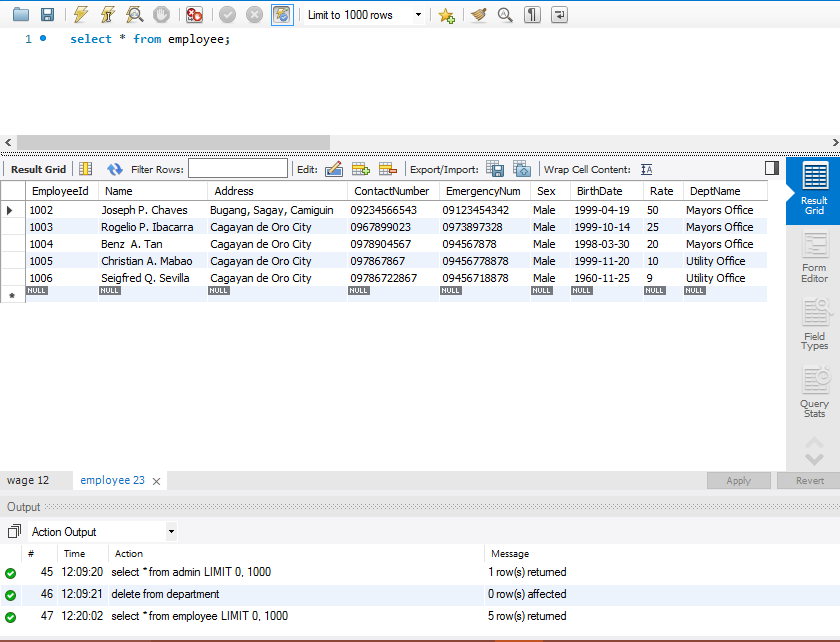
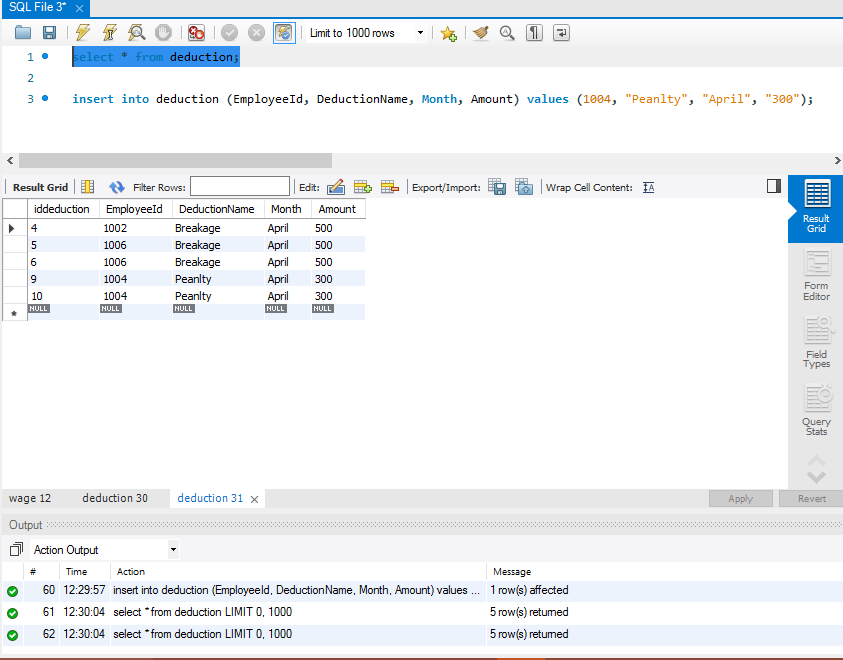
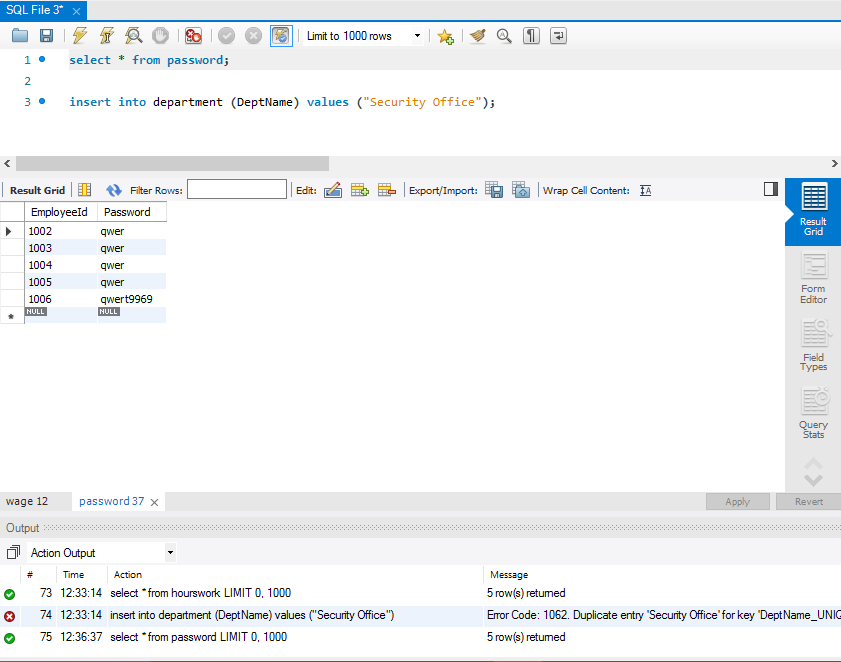
Name varchar(20) NOT NULL default '0',

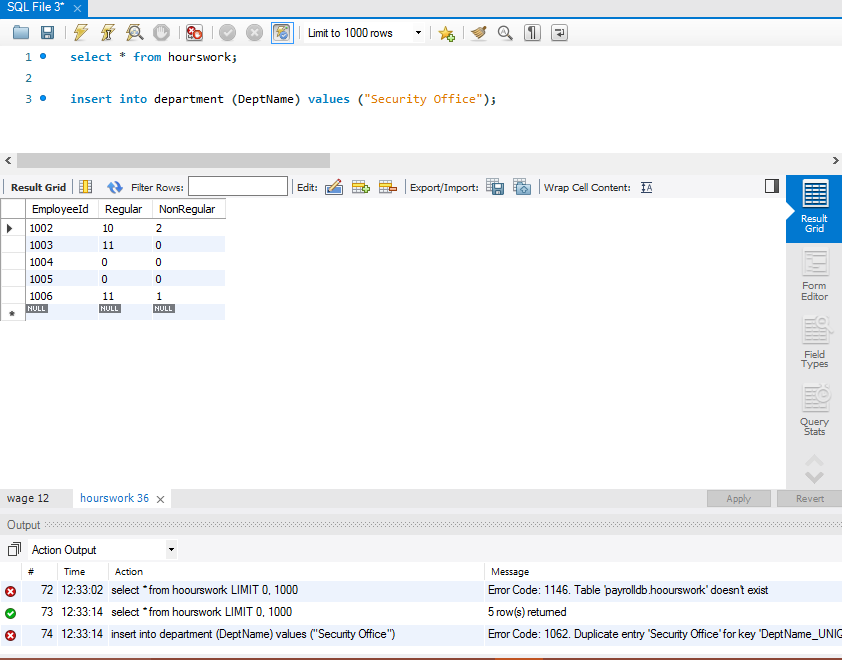
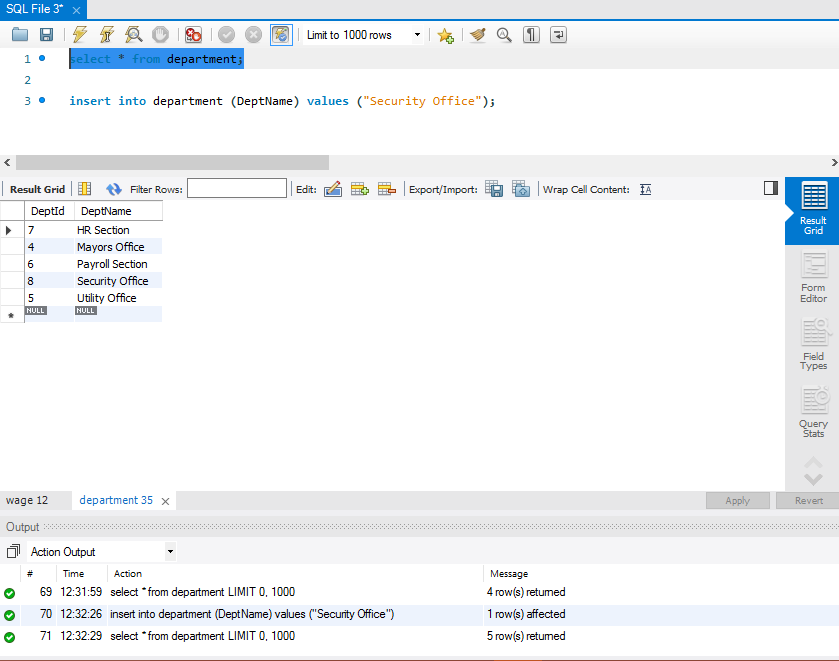
password varchar(20) NOT NULL default '0',

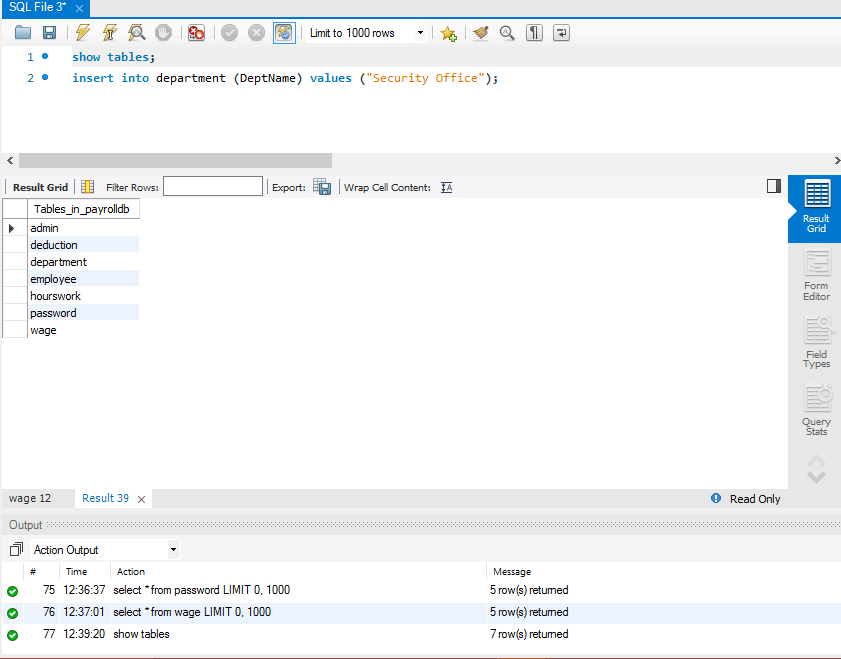
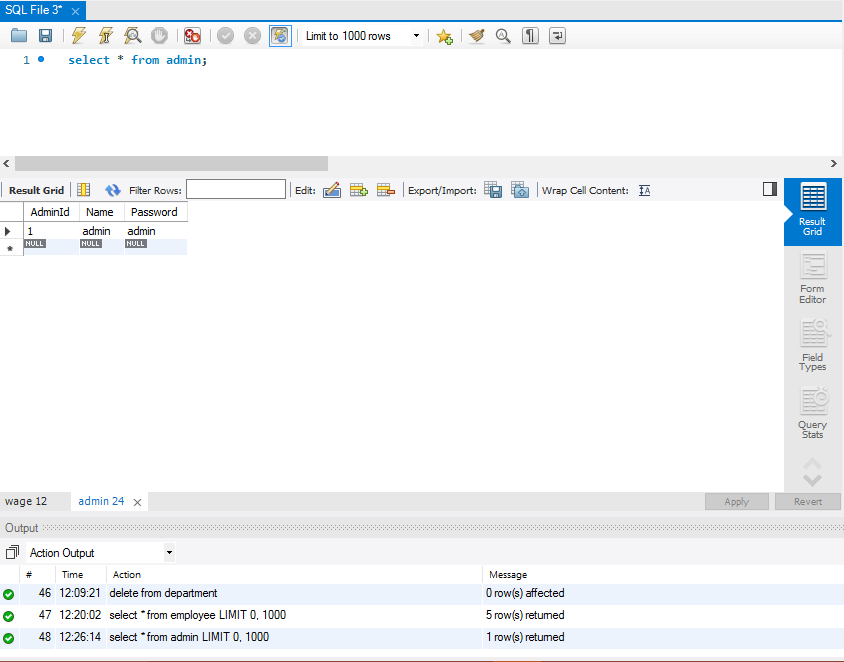
PRIMARY KEY (AdminId) )

**ERD (Workbench)**

****

**Table (Workbench)**

****

****