

Employee Management System

¹Yanqi Ma, ¹Wentian Zhang, ²Yanbo Li

¹Information Systems, Northeastern University

²Data Architecture and Management, Northeastern University

{ma.yanqi1, zhang.wenti, li.yanb}@northeastern.edu

Abstract

The project that we are working on focuses on managing telecommuting and employee mobility during the ongoing epidemic.

To achieve this, we have utilized JavaFX and scene builder to design a user-friendly interface for administrators to login, access the home page, add new employees, and manage employee salaries across four pages. Regarding the login screen, the administrator must have the correct username and password to access it. Regarding the home page, we have total employees, total online employees and total inactive employees for comparison, and a bar chart showing how long each employee has been with the system. For the Add Employee page, we have set five buttons to import, add, update, delete and clear to facilitate the function. For the Employee Salary page, we can change the salary by selecting any row and clicking the Update and Clear buttons. Finally we can go back to the home page and see the corresponding changes by the total number of people, the total number of people online and the total number of people inactive.

All the employee information is stored in a MySQL database, which allows for easy retrieval and management of the data. To implement the back-end functionality, we have made use of class definitions, collections, polymorphism, interfaces, and generics in Java, which are essential tools for building robust and scalable software applications. By incorporating these key programming concepts, we have been able to create a reliable and efficient system that can effectively manage employee information and payroll, even in the face of the challenges posed by the ongoing epidemic.

Keywords—Eclipse, JavaFx, SceneBuilder, MySQL, Interface, Class definitions, Collections, Polymorphism, Generics, Covid-19

I. PROBLEM DESCRIPTION

Project Background: The COVID-19 pandemic has had a significant impact on the way that companies operate, with many organizations now allowing their employees to telecommute in order to comply with social distancing guidelines. However, this has led to an increase in employee turnover rates, as well as changes in employee motivation levels due to the remote work environment.

To address this issue, we have developed an employee management system that allows companies to keep their employee information up-to-date in a timely manner. Additionally, our system can adjust salaries for existing employees based on their motivation levels, which helps to keep them engaged and motivated in their work. To provide a

comprehensive overview of the company's workforce, our system displays the total number of employees, the total number of online employees, and the total number of inactive employees on the homepage.

By incorporating these features into our employee management system, we aim to help companies manage their remote workforce more effectively and maintain high levels of employee engagement and productivity in the face of the ongoing pandemic.

II. ANALYSIS (RELATED WORK)

According to the search, there are not too many employee management systems for the high workplace mobility and reduced employee motivation caused by the epidemic because the new crown epidemic is a hot topic in recent years, so our employee management system is time-advanced. Then we determined the theme, UI design and division of labor.

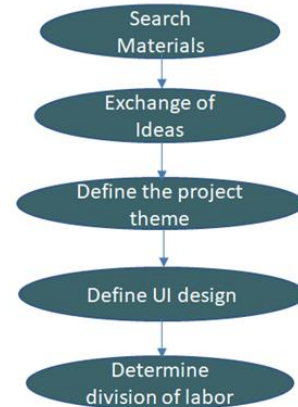


Figure 1. Workflow of Team

Also, We identified the teamwork software. For team communication, we use Whatsapp and Zoom. For version control, we use git and github to collaborate on different code parts concurrently. For Doc file, PPT, we use Google Drive, Google Doc, and Google Slides to synchronize our parts. We used Eclipse, scene builder as our choice of IDE. And MySQL database to store our file

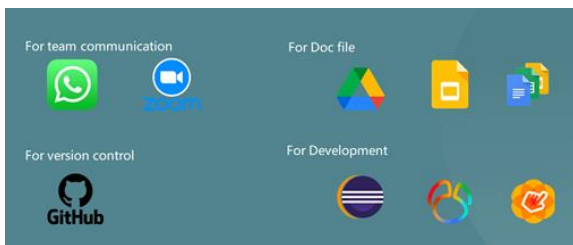


Figure 2. Software of Team

But during the process of the project, we did encounter some difficulties.

The first one is how to add database. Considering the large number of employees, it was impossible for the administrator to enter each employee's information manually, so we learned to use MySQL for database management by looking up YouTube videos.

```

admin
employee
1 drop table if exists admin;
2 create table admin(
3     id int not null auto_increment,
4     PRIMARY KEY(id),
5     username VARCHAR(100) not null,
6     password VARCHAR(100) not null
7 );
8 insert into admin(username, password)
9 VALUES("admin", "admin123");
10
11
12 drop table if exists employee;
13 create table employee(
14     id int not null auto_increment,
15     PRIMARY KEY(id),
16     employee_id int(100) not null,
17     firstname VARCHAR(100) not null,
18     lastname VARCHAR(100) not null,
19     gender VARCHAR(100) not null,
20     phonenumber VARCHAR(100) not null,
21     position VARCHAR(100) not null,

```

Figure 3. MySQL database

The second is how to make UI design look better and more realistic. Adding jar packages, using CSS files for uniform formatting will realize it.



Figure 4. Jar package

```

6
7
8 .left-form{
9     -fx-background-color:linear-gradient(to bottom right, #74ebd5, #ACB6E5);
10    -fx-border-color:#000;
11    -fx-border-width:.4px 0px .4px .4px;
12 }
13
14 .right-form{
15     -fx-background-color:#fff;
16     -fx-border-color:#000;
17     -fx-border-width:.4px .4px .4px 0px;
18 }
19
20 .close{
21     -fx-background-color:transparent;
22     -fx-cursor:hand;
23 }
24
25 .close:hover{
26     -fx-background-color:#ff3547;
27 }
28
29 .textfield{
30     -fx-background-color:linear-gradient(to bottom, #efefef, #eee);
31     -fx-background-radius:4px;
32     -fx-border-color:#000;
33     -fx-border-width:.4px;
34     -fx-border-radius:4px;
35     -fx-font-size:13px;
36     -fx-font-family:"Arial";
37 }

```

Figure 5. CSS file

III. SYSTEM DESIGN

Our project focuses on admin for this one account case.

For login, we use javafx, scene builder to build fundamental UI design, and then use java code to design controllers like Labels, AnchorPane, Button, TextField, FontAwesomeIcon etc.

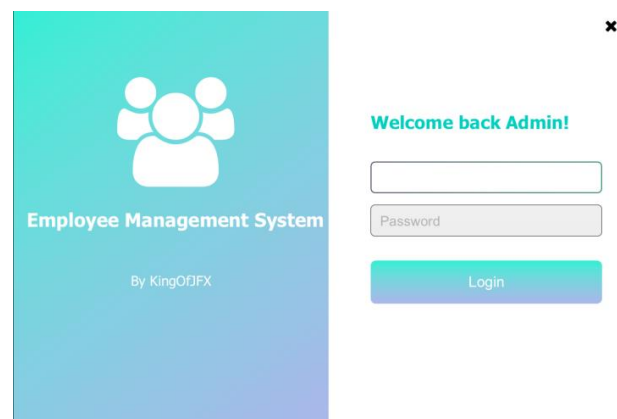


Figure 6. Login Design

After logging in, the administrator can use the Home, Add employee and employee salary buttons to achieve different functions.

For Homepage, we will show the total number of employees, the total number of online employees and the total number of inactive employees respectively through the BarChart.

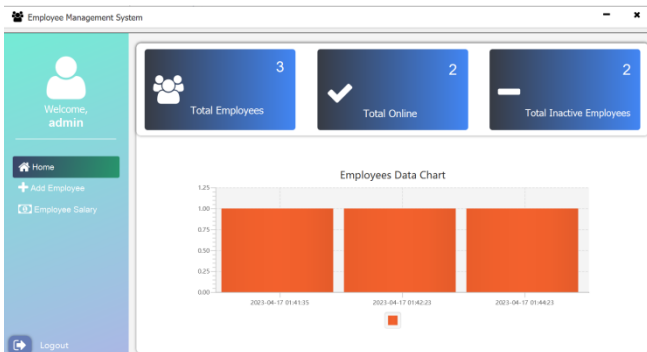


Figure 7. Homepage Design

For the add employee page, we will use add, update, delete and clear buttons to the employee information for the administrator to update and manage the latest employee information.

Figure 8. Add Employee page Design

For the employee salary page, admin can change the employee salary by checking the existing employee information line. If the employee salary is adjusted to be less than the previous salary, it indicates that the employee is less motivated and the number of inactive employees on the main page changes accordingly.

Figure 9. Employee Salary page Design

```

employeeManagementSystem
├── src
│   ├── employeemanagementsystem
│   │   ├── dashboardController.java
│   │   ├── database.java
│   │   ├── employeeData.java
│   │   ├── EmployeeManagementSystem.java
│   │   ├── FXMLDocumentController.java
│   │   ├── getData.java
│   │   ├── dashboard.fxml
│   │   ├── dashboardDesign.css
│   │   ├── employee.sql
│   │   ├── FXMLDocument.fxml
│   │   └── loginDesign.css
│   ├── JRE System Library [jre]
│   ├── JavaFX
│   ├── Referenced Libraries
│   ├── build
│   ├── dist
│   ├── nbproject
│   └── photos

```

Figure 10. Structure of project

IV. IMPLEMENTATION

A. Admin Login

For the login screen, we store the username and password via MySQL and do null validation, both must be correct before you can login. The entered credentials are then validated against a database or user directory to ensure that they are correct. If the credentials are valid, the user is granted access to the admin panel, which contains the necessary tools and features for managing the website or application.

V. EVALUATION

Present the result of the project. You are suggested to have following content in this section:

- The screenshots of sample run and the explanation

Admin login requires username and password information.

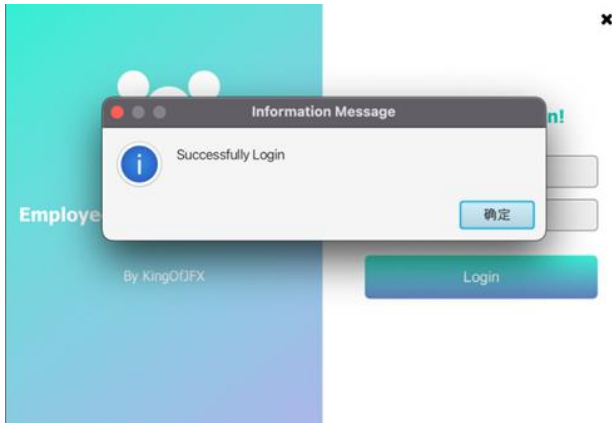


Figure 11. Admin Login

B. Admin dashboard feature

After logging in, enter the admin dashboard interface, admin can click home, add employee, employee salary three button to switch the page, which is used to switchForm function to achieve in the controller. The underlying logic knowledge to different stage of the visitable property for true/false switch.

The other button actions of the three function pages are triggered directly by the scenebuilder. We used JDBC to connect javafx application with MySQL database. JDBC provides a standard way for Java programs to interact with a database by providing a set of classes and interfaces for database connectivity and a driver API layer for interfacing with different database management systems. We change the data in the database by precompiling the sql statements that need to be executed for actions such as add, delete and update.

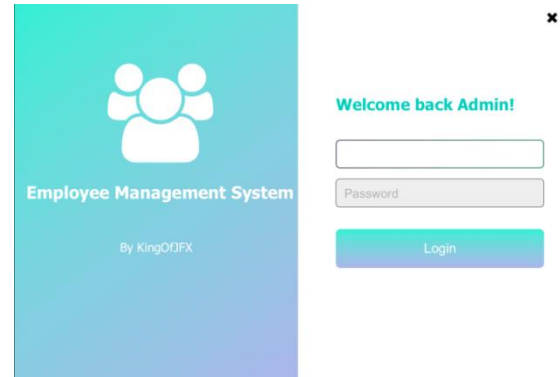


Figure 13. Admin login

When the admin login information has empty values, the system will pop up the pop-up window that please fill all blank fields.

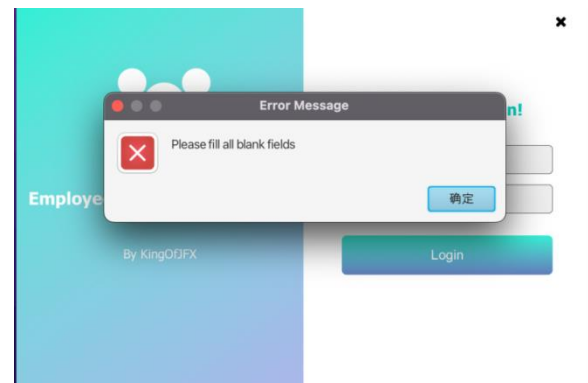


Figure 14. If Admin null

When the admin login information has wrong values, the system will pop up the pop-up window that Wrong Username/Password.

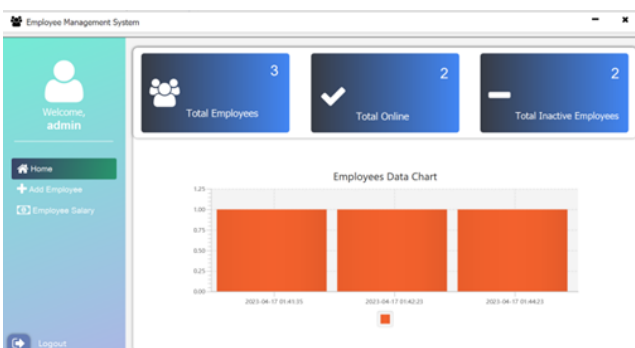


Figure 12. Admin Dashboard

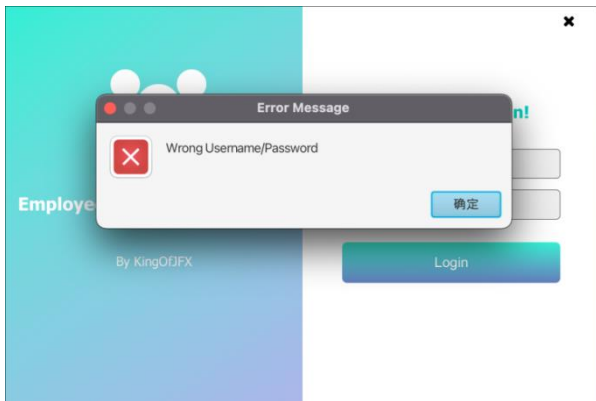


Figure 15. If Admin mismatch

When the admin information is correct, a pop-up window pops up with a successfully login.

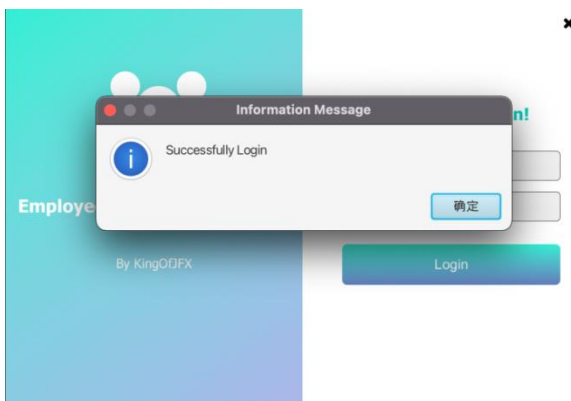


Figure 16. Login successfully

Home page show the total employees, total online and total inactive employees and chart show the every employee join in time. The visible properties of following three pages(homepage, add employee page, employee salary page) will change when click on these three buttons.

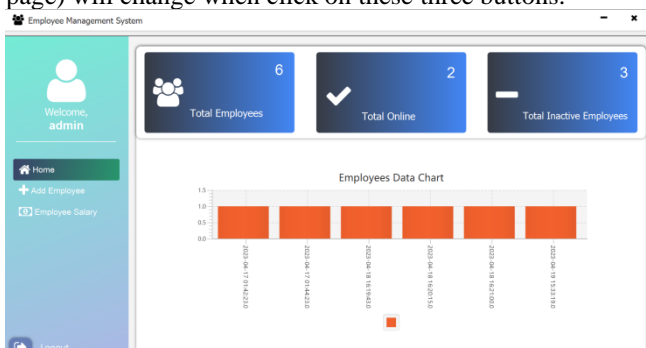


Figure 17. Click home

Add employee includes ID, name, gender, phone and position and photo. The addEmployeeAdd action will be triggered when the “Add” button is clicked. At the same

time, the “insert into” statement of sql will also be executed.

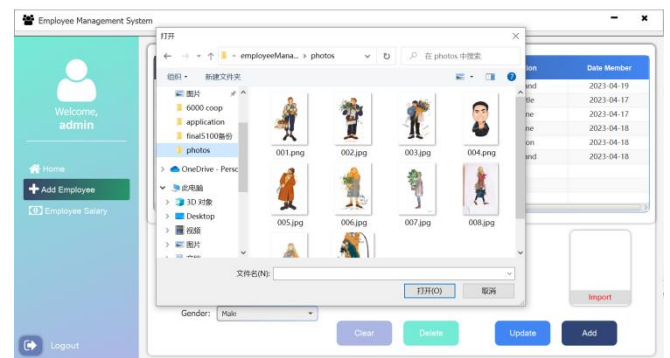


Figure 17. Click add employee

If employee Information has null and the add button is clicked, a pop-up window pops up with a please fill all blank fields. Because the employee table in mysql database required each column to be filled.

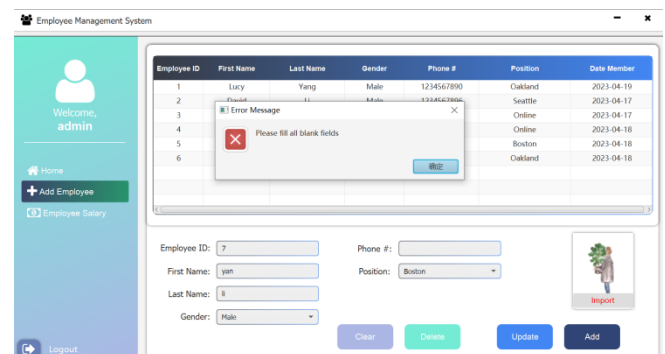


Figure 18. Add employee null

The employee information is filled and the add button is clicked, a pop-up window pops up with a successfully added.

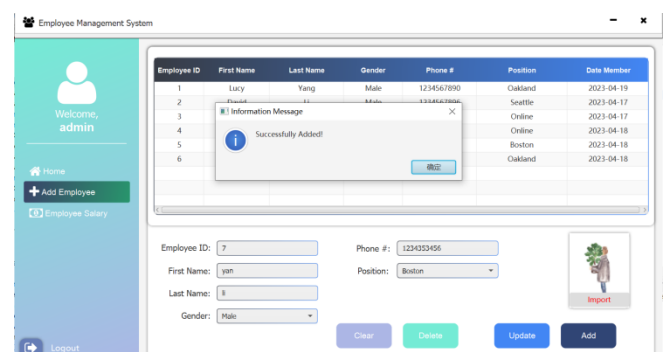


Figure 19. Add employee successfully

Choose the row of employee that you want to update and clicks the update button, a pop-up window pops up with a Are you sure you want to update employee ID: ?. We can

choose the employee's information we want to update by employees' unique id.

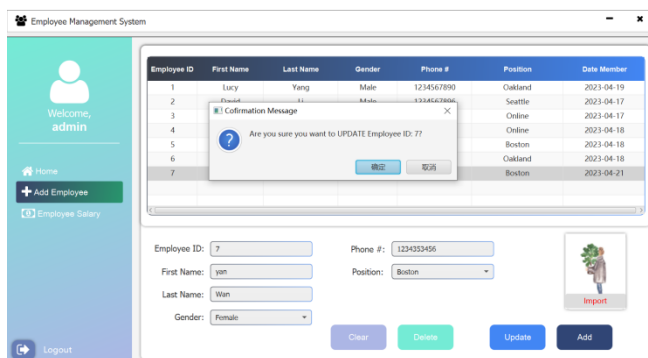


Figure 20. Update employee Confirmation

When you clicks sure, a pop-up window pops up with a Successfully Updated. And the “addEmployeeUpdate” action will be triggered.

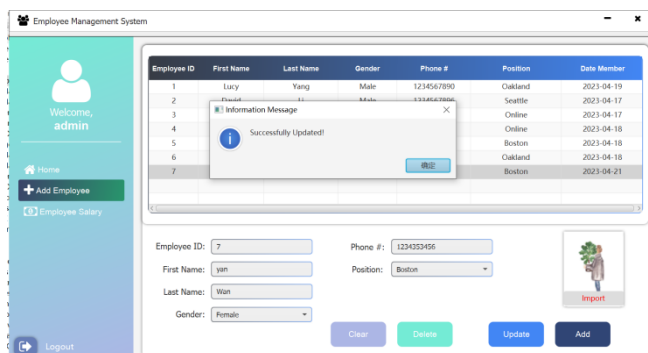


Figure 21. Update employee Successfully

Choose the row of employee that you want to delete and clicks the delete button, a pop-up window pops up with a Are you sure you want to delete employee ID: ?. The “addEmployeeDelete” action will be triggered.

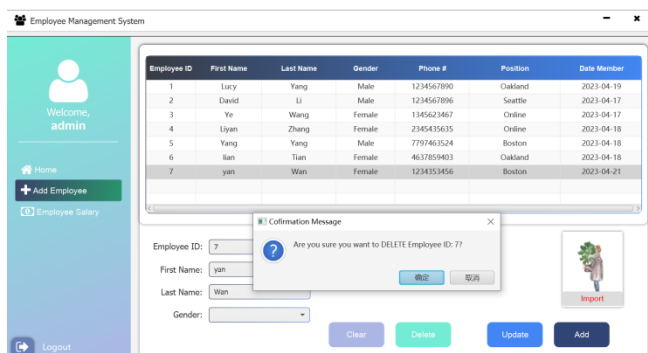


Figure 22. Delete employee Confirmation

When you clicks sure, a pop-up window pops up with a Successfully Deleted. The “where” keyword can be used to

filter out employees that need to be deleted by employee_id.

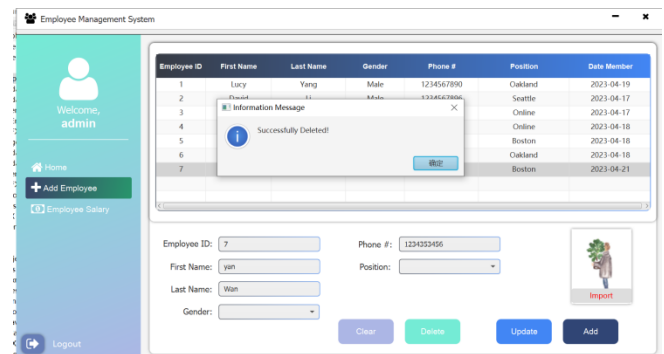


Figure 23. Delete employee successfully

Choose the row of employee that you want to clear, then the textfields will be filled with row information.

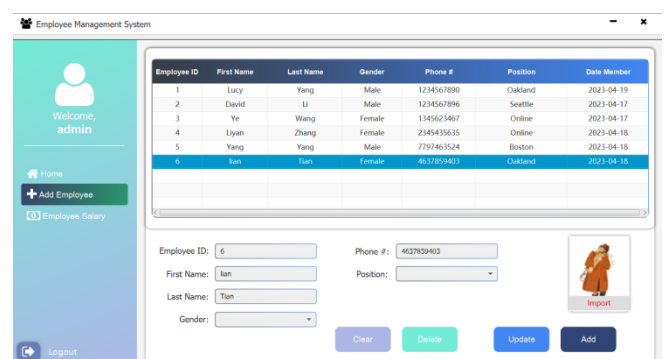


Figure 24. Choose one to clear

Then clicks the clear button. The textfields will be cleared. And the “addEmployeeReset” action will be triggered.

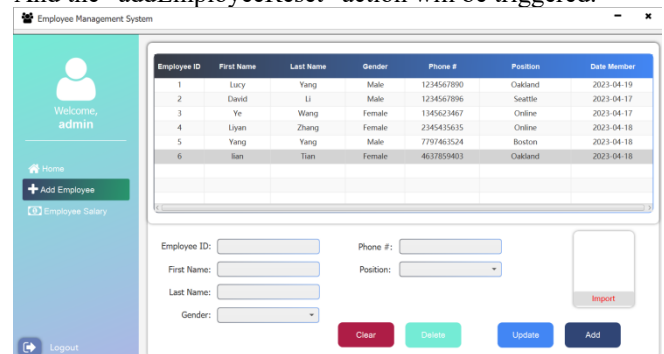


Figure 25. Clear successfully

Click the employee salary, you will see employee information but you can only change salary. Then choose one row you want to change.

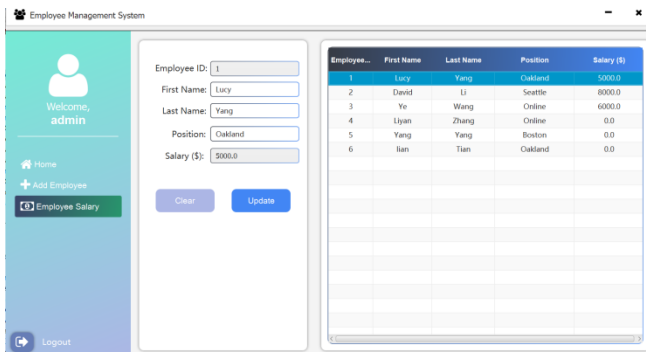


Figure 26. Click employee salary

Change the salary of the row(0 to 8000) and clicks the update button, a pop-up window pops up with a Successfully Updated. And the “salaryUpdate” action will be triggered. At the same time, the precompiled sql statement will be executed.

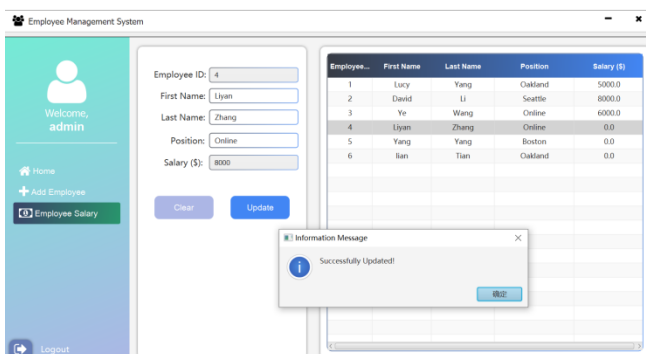


Figure 27. Update employee salary

Also, when you choose one row to clear, you will see the information in all textfields.

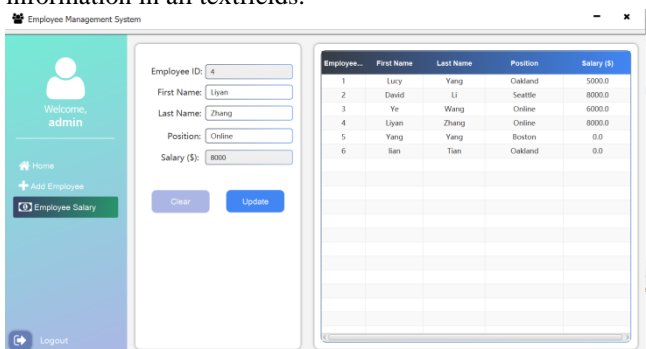


Figure 28. Choose one to clear

Then you clicks clear button, all textfields will be blank.

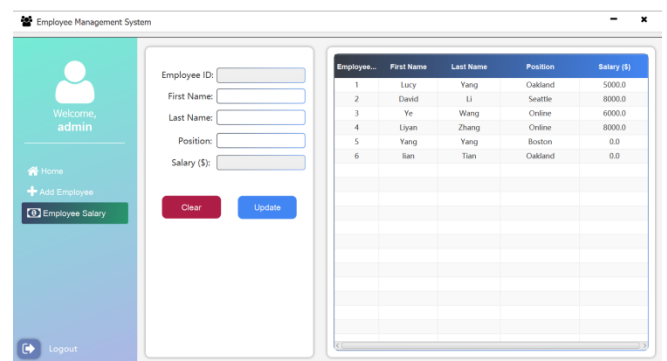


Figure 29. Clear successfully

Because you have changed one employee salary from 0 to 8000, and all the data is selected from the same mysql database, so when you return homepage, the total inactive employees will change from 3 to 2.

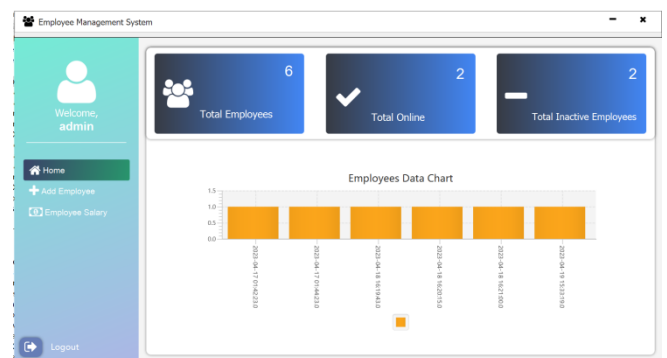


Figure 30. Return home

When you want to log out, click the button at the left of the home page, a pop-up window pops up with a Are you sure you want to logout.

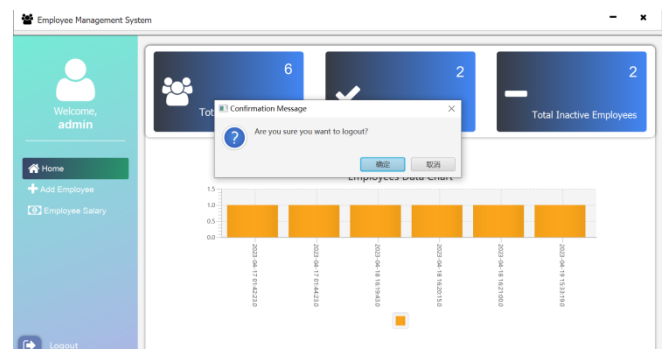


Figure 31. Logout Confirmation

When you clicks sure and run project again, you will see the login again

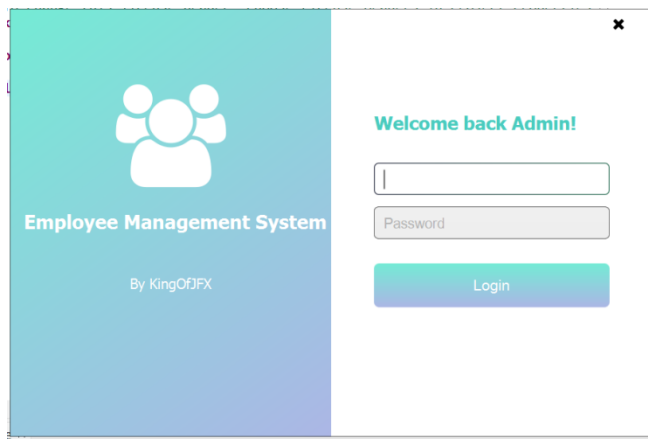


Figure 32. Login again

- Real user feedback

Ruo Zhang(Director of the Human Resources Department of a listed company in China):

When I used the system for employee management during the epidemic, I felt it was very easy and simple, and the page design was very clear and straightforward, but the reason for the payroll drop was not only in the online office, but could also occur in the field office.

It may not be particularly scientific to draw conclusions directly by comparing the number of people working online with the number of inactive employees.

VII. DISCUSSION (REFLECTION)

By comparing the total number of employees, the total number of online workers and the total number of inactive employees on the home page after salary adjustment, it can be concluded that telecommuting does affect the motivation of employees to some extent. Our system includes a feature that allows for the adjustment of employee salaries based on their motivation levels. This feature is designed to help organizations maintain high levels of employee engagement and productivity in the face of the ongoing pandemic, which has led to changes in employee motivation levels due to the remote work environment. By providing a mechanism for adjusting salaries based on employee motivation levels, our system aims to promote a more positive and productive work environment. An employee management system can also help in ensuring keep track of changes in the company's workforce by providing real-time updates to managers.

VIII. CONCLUSIONS AND FUTURE WORK

- What are the advantages or benefits of using your solution?

The employee management system facilitates employee management during special periods such as epidemics, improves employee motivation through salary adjustments, thus improving company efficiency and reducing company personnel costs.

- What are the problems found but not yet explored in the project?

The system only detects unmotivated employees through payroll changes, but there can be multiple reasons for lower motivation and other force majeure reasons such as health, so it may be necessary to develop additional features to improve employee motivation.

- If your team has more time, what do you want to improve?

Maybe add an additional account case, like an employee account, to see the trend of the employee's own salary over the year on the main page or something. Also, considering the realistic situation that it is impossible for admin to remember all employee information, adding a search feature in add employee page will be more convenient.

IX. JOB ASSIGNMENT

- **Yanqi Ma:** Employee delete& clear function completed, Employee Salary assign & update function completed, MySQL Database code, Modify PPT format
- **Wentian Zhang:** Admin Login UI design and function completed, Homepage Interface and function completed , Project Report
- **Yanbo Li:** Employee add & update function completed, MySQL database, PPT design

REFERENCES

- [1] Chen, Z., & Luo, L. (2021). The impact of COVID-19 on remote work: A study of work-life balance and productivity among knowledge workers. *Journal of Business Research*, 133, 511-519.
- [2] Li, X., Liu, J., Chen, H., & Liu, X. (2020). COVID-19 pandemic and remote work: A review of challenges and opportunities. *Journal of Business Research*, 120, 1-