



**CONTENTS**

**S.No Topics Page No.**

1. **Introduction 3**
2. **Description of the Project 4-5**
3. **Algorithm 5-7**
4. **Project Code 8-10**
5. **Result 11**
6. **Conclusion 12**

**CHAPTER 1**

# INTRODUCTION

1.1 Introduction of University Employee Management System

A university employee management system is a computer-based software solution designed to streamline and automate the process of managing employees within a university or educational institution. This system helps to manage employee records, track attendance, manage payroll, and automate the process of scheduling and managing employee tasks.

1.2 Features

A. Centralized database:

The system provides a centralized database that allows administrators to manage employee records and access them easily.B. Attendance tracking:

The system allows administrators to track employee attendance easily.C. Payroll management:

The system helps to manage payroll and make the payment process more efficient.D. Task scheduling:

The system automates the process of scheduling and managing employee tasks.E. Reporting and analytics:

The system includes reporting and analytics features that provide valuable insights into employee performance, attendance, and other important metrics.

**CHAPTER 2**

**DESCRIPITION OF THE PROJECT**

2.1 Employee Data Management

The UEMS allows the HR department to manage employee data such as personal information, job details, qualifications, and other relevant information. It provides a central repository for employee records, allowing HR to access and update employee data in real-time. This module also provides advanced search and reporting features for easy retrieval of employee information.

2.2 Payroll Management

The payroll module of the UEMS automates the process of calculating employee salaries, deductions, and taxes. It streamlines the payroll process, making it easy for HR to manage employee pay and generate payslips. The system also allows for direct deposit, reducing the time and effort required to distribute paychecks.

2.3 Attendance Management

The attendance module of the UEMS tracks employee attendance, including clock-in and clock-out times. It provides real-time attendance data, allowing HR to monitor employee attendance and track any attendance-related issues. The system also generates reports to help HR analyze attendance patterns and identify areas for improvement.

2.4 Leave Management

The UEMS streamlines the process of applying for and approving employee leave requests. The system allows employees to submit leave requests and HR to approve or reject them. It also tracks employee leave balances and generates reports to help HR monitor leave trends and manage employee absences.

2.5 Performance Management

The UEMS provides tools for managing employee performance, including setting goals, conducting performance evaluations, and tracking performance metrics. The system allows HR to manage the entire performance evaluation process, from setting goals to generating performance reports. It also provides dashboards for employees and managers to track performance metrics and progress towards goals.

2.6 Training and Development

The UEMS provides tools for managing employee training and development programs. It allows HR to create training programs, assign courses to employees, and track progress towards completion. The system also provides reports on training completion rates, allowing HR to identify areas where additional training may be needed.

**CHAPTER 3**

**ALGORITHM**

STEP 1 - Initialize the system:

* Set up a database to store employee information, such as names, roles, contact information, and other relevant data.
* Create user interfaces for different stakeholders, such as administrators, HR personnel, and employees, to interact with the system.

STEP 2 - Employee registration:

* Employees can register in the system by providing their personal information, such as name, contact information, and employment details, and creating a unique username and password.
* Validate and store the employee information in the database.

STEP 3 - Employee authentication:

* Employees can log in to the system using their registered username and password.
* Verify the credentials against the stored information in the database to authenticate the employee.

STEP 4 - Employee profile management:

* + Employees can update their personal information, such as contact details, bank account information, and emergency contact information.
  + Validate and store the updated information in the database.

STEP 5 - Role-based access control:

* + Different roles, such as administrators, HR personnel, and employees, have different levels of access to the system.
  + Implement role-based access control to restrict or grant access to different features and functions of the system based on the user's role.

STEP 6 - Leave management:

* + Employees can apply for leaves through the system, specifying the type of leave, start and end dates, and reason.
  + HR personnel can review and approve/reject the leave applications based on company policies and available leave balances.
  + Update the leave balances and notify the employees about the status of their leave applications.

STEP 7 - Attendance management:

* + Employees can mark their attendance through the system, either manually or using a biometric or card-based system.
  + HR personnel can review and manage attendance records, generate attendance reports, and track attendance trends.

STEP 8 - Payroll management:

* + HR personnel can manage employee salaries, deductions, and other payroll-related information.
  + Generate monthly salary slips, tax forms, and other payroll-related reports.
  + Update the payroll information based on employee promotions, bonuses, or other changes.

STEP 9 - Performance management:

* + HR personnel can set up performance goals, conduct performance appraisals, and track performance metrics for employees.
  + Employees can view their performance goals, self-assessments, and feedback from supervisors.
  + Generate performance reports and use them for employee evaluations and promotions.

STEP 10 - Employee offboarding:

* + HR personnel can manage employee terminations, retirements, or resignations.
  + Update the employee's status in the system, deactivate their account, and collect necessary exit clearance forms.

STEP 11 - Error handling and reporting:

* + Implement error handling mechanisms to handle exceptions, data validation errors, and other system errors gracefully.
  + Log system activities and generate reports for auditing and monitoring purposes.

STEP 12 - Security measures:

* + Implement data encryption, access controls, and other security measures to protect employee information and ensure data privacy.
  + Regularly update and patch the system to protect against security vulnerabilities.

STEP 13 - User support:

* Provide user support through documentation, helpdesk, and training materials to assist users in navigating and using the system effectively.

**CHAPTER 4**

**CODE**

#include <gtk/gtk.h>

#include <stdio.h>

#include <string.h>

#define MAX\_EMPLOYEES 10

struct Employee {

int id;

char name[100];

int age;

double salary;

};

GtkWidget \*window, \*grid, \*idLabel, \*nameLabel, \*ageLabel, \*salaryLabel, \*idEntry, \*nameEntry, \*ageEntry, \*salaryEntry, \*addButton;

struct Employee employees[MAX\_EMPLOYEES] = {0};

int numEmployees = 0;

void addEmployee(GtkWidget \*widget, gpointer data) {

if (numEmployees >= MAX\_EMPLOYEES) {

GtkWidget \*dialog = gtk\_message\_dialog\_new(GTK\_WINDOW(window), GTK\_DIALOG\_DESTROY\_WITH\_PARENT, GTK\_MESSAGE\_ERROR, GTK\_BUTTONS\_CLOSE, "Maximum number of employees reached!");

gtk\_dialog\_run(GTK\_DIALOG(dialog));

gtk\_widget\_destroy(dialog);

return;

}

const char \*idString = gtk\_entry\_get\_text(GTK\_ENTRY(idEntry));

const char \*nameString = gtk\_entry\_get\_text(GTK\_ENTRY(nameEntry));

const char \*ageString = gtk\_entry\_get\_text(GTK\_ENTRY(ageEntry));

const char \*salaryString = gtk\_entry\_get\_text(GTK\_ENTRY(salaryEntry));

if (strlen(idString) == 0 || strlen(nameString) == 0 || strlen(ageString) == 0 || strlen(salaryString) == 0) {

GtkWidget \*dialog = gtk\_message\_dialog\_new(GTK\_WINDOW(window), GTK\_DIALOG\_DESTROY\_WITH\_PARENT, GTK\_MESSAGE\_ERROR, GTK\_BUTTONS\_CLOSE, "Please fill in all fields!");

gtk\_dialog\_run(GTK\_DIALOG(dialog));

gtk\_widget\_destroy(dialog);

return;

}

int id = atoi(idString);

int age = atoi(ageString);

double salary = atof(salaryString);

strcpy(employees[numEmployees].name, nameString);

employees[numEmployees].id = id;

employees[numEmployees].age = age;

employees[numEmployees].salary = salary;

numEmployees++;

gtk\_entry\_set\_text(GTK\_ENTRY(idEntry), "");

gtk\_entry\_set\_text(GTK\_ENTRY(nameEntry), "");

gtk\_entry\_set\_text(GTK\_ENTRY(ageEntry), "");

gtk\_entry\_set\_text(GTK\_ENTRY(salaryEntry), "");

GtkWidget \*dialog = gtk\_message\_dialog\_new(GTK\_WINDOW(window), GTK\_DIALOG\_DESTROY\_WITH\_PARENT, GTK\_MESSAGE\_INFO, GTK\_BUTTONS\_CLOSE, "Employee added successfully!");

gtk\_dialog\_run(GTK\_DIALOG(dialog));

gtk\_widget\_destroy(dialog);

// Write employees to CSV file

FILE \*fp = fopen("emp.csv", "a");

if (fp == NULL) {

printf("Error opening file!\n");

return;

}

if (numEmployees == 1) {

fprintf(fp, "ID,Name,Age,Salary\n");

}

fprintf(fp, "%d,%s,%d,%f\n", employees[numEmployees-1].id, employees[numEmployees-1].name, employees[numEmployees-1].age, employees[numEmployees-1].salary);

fclose(fp);

}

const char \*getEntryText(GtkWidget \*entry) {

const char \*text = gtk\_entry\_get\_text(GTK\_ENTRY(entry));

if (strlen(text) == 0) {

GtkWidget \*dialog = gtk\_message\_dialog\_new(GTK\_WINDOW(window), GTK\_DIALOG\_DESTROY\_WITH\_PARENT, GTK\_MESSAGE\_ERROR, GTK\_BUTTONS\_CLOSE, "Please fill in all fields!");

gtk\_dialog\_run(GTK\_DIALOG(dialog));

gtk\_widget\_destroy(dialog);

return NULL;

}

return text;

}

int main(int argc, char \*argv[]) {

GtkWidget \*idLabel, \*nameLabel, \*ageLabel, \*salaryLabel, \*addButton;

gtk\_init(&argc, &argv);

// Create the main window

window = gtk\_window\_new(GTK\_WINDOW\_TOPLEVEL);

gtk\_window\_set\_title(GTK\_WINDOW(window), "Employee Management System");

g\_signal\_connect(window, "destroy", G\_CALLBACK(gtk\_main\_quit), NULL);

// Create the grid to hold the input fields and button

grid = gtk\_grid\_new();

gtk\_container\_add(GTK\_CONTAINER(window), grid);

// Add the ID field

idLabel = gtk\_label\_new("ID:");

gtk\_grid\_attach(GTK\_GRID(grid), idLabel, 0, 0, 1, 1);

idEntry = gtk\_entry\_new();

gtk\_grid\_attach(GTK\_GRID(grid), idEntry, 1, 0, 1, 1);

// Add the name field

nameLabel = gtk\_label\_new("Name:");

gtk\_grid\_attach(GTK\_GRID(grid), nameLabel, 0, 1, 1, 1);

nameEntry = gtk\_entry\_new();

gtk\_grid\_attach(GTK\_GRID(grid), nameEntry, 1, 1, 1, 1);

// Add the age field

ageLabel = gtk\_label\_new("Age:");

gtk\_grid\_attach(GTK\_GRID(grid), ageLabel, 0, 2, 1, 1);

ageEntry = gtk\_entry\_new();

gtk\_grid\_attach(GTK\_GRID(grid), ageEntry, 1, 2, 1, 1);

// Add the salary field

salaryLabel = gtk\_label\_new("Salary:");

gtk\_grid\_attach(GTK\_GRID(grid), salaryLabel, 0, 3, 1, 1);

salaryEntry = gtk\_entry\_new();

gtk\_grid\_attach(GTK\_GRID(grid), salaryEntry, 1, 3, 1, 1);

// Add the "Add" button

addButton = gtk\_button\_new\_with\_label("Add Employee");

g\_signal\_connect(addButton, "clicked", G\_CALLBACK(addEmployee), NULL);

gtk\_grid\_attach(GTK\_GRID(grid), addButton, 0, 4, 2, 1);

// Show the window and start the main loop

gtk\_widget\_show\_all(window);

gtk\_main();

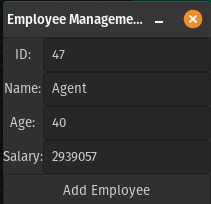
return 0;

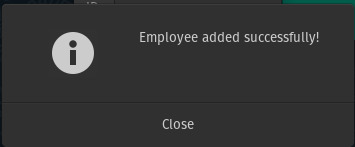
}

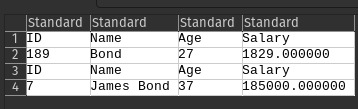
**CHAPTER 5**

**RESULT**

EMPLOYEE ADDED







**CHAPTER 6**

**CONCLUSION**

Overall, a university employee management system can be a valuable tool for any educational institution looking to improve the management of their employees and streamline their operations. The system can help to increase productivity, improve employee satisfaction, and reduce costs for the institution. Investing in a well-designed and well-implemented employee management system can help universities or any organization streamline their HR processes, save time and resources, and ultimately enhance the employee experience. The University of Employee Management System is a comprehensive HR solution designed to streamline employee management within the University. It provides a centralized platform for managing employee data, payroll, attendance, leave, performance, and training. The system improves efficiency and accuracy, allowing HR to focus on strategic initiatives that support the University's goals.