



**Ain Shams University**

**Faculty of Engineering**

**Computer Engineering and Software Systems Program**

---

## **CSE227: Database Systems (1)**

### **Company Management System**

#### **Submitted By:**

|                                     |         |
|-------------------------------------|---------|
| Ahmed Hesham Mohamed Al-Saey        | 16P6000 |
| Ahmed Mamdouh Mohammed Ahmed Mounir | 16P6020 |
| Ahmed Mohamed Ahmed Elqaffas        | 16P6012 |
| Hossam ElDin Khaled Mohamed Ali     | 16P3025 |
| Sherif Ashraf Ahmed Morsy           | 16P9033 |

#### **Submitted to:**

Dr. Gamal Ibrahim Abdel Shafy

## **EXECUTIVE SUMMARY**

This document is intended to be a report for our database course projects. The project is about a company management system. The project is divided into two parts, the designing and implementation of the database, and the development of the actual programs that contains the CRUD operations on the set of tables in this database.

This report focuses on the designing and implementation of the database. It shows the conceptual (although we skipped this phase and made the logical design at once), logical and physical design of the database, project description, the role of each team member, and the team leader. The report also contains test cases and end user guide.

The project description contains all the assumptions that we made during the project execution. We divided ourselves into a stakeholder and designers to simulate what a real project would be like and to come up with functionalities that would make sense.

## TABLE OF CONTENTS

|                                   |     |
|-----------------------------------|-----|
| EXECUIVE SUMMARY.....             | i   |
| LIST OF FIGURES.....              | iii |
| 1.0 INTRODUCTION.....             | 1   |
| 2.0 PROJECT DESCRIPTION.....      | 2   |
| 3.0 ROLES OF TEAM MEMBERS.....    | 2   |
| 4.0 LOGICAL DESIGN .....          | 3   |
| 5.0 QUERIES .....                 | 4   |
| 5.1 Join .....                    | 4   |
| 5.2 View .....                    | 5   |
| 5.3 Inner Select .....            | 5   |
| 6.0 USER GUIDE .....              | 6   |
| 6.1 Adding .....                  | 6   |
| 6.2 View, Modify and Delete ..... | 8   |
| 6.3 Manage Project .....          | 10  |
| 6.3.1 project manager .....       | 11  |
| 6.3.2 project employees .....     | 11  |
| 6.3.3 milestones .....            | 13  |
| 6.4 Generate Reports .....        | 13  |
| APPENDIX .....                    | 15  |

## LIST OF FIGURES

|   |    |
|---|----|
| Figure 1: Entity relationship diagram .....                         | 3  |
| Figure 2: Select add data .....                                     | 6  |
| Figure 3: Choose entity .....                                       | 7  |
| Figure 4: Enter details.....  | 7  |
| Figure 5: Choose Modify/Delete data .....                           | 8  |
| Figure 6: Choose entity to modify .....                             | 9  |
| Figure 7: viewing the search result for department “database” ..... | 9  |
| Figure 8: Modifying the department .....                            | 10 |
| Figure 9: Select manage project.....                                | 10 |
| Figure 10: Assign project manager .....                             | 11 |
| Figure 11: Viewing employees assigned to the project .....          | 12 |
| Figure 12: Assigning employees to a project based on role .....     | 12 |
| Figure 13: Adding a milestone to a certain project .....            | 13 |
| Figure 14: Generating reports .....                                 | 13 |
| Figure 15: Report generation .....                                  | 14 |

# **1.0 INTRODUCTION**

## **1.1 Purpose of The Report**

This report discusses our project for database course. This report is intended for our instructor and TA. However, the content of this report can be understood by anyone who has knowledge of the ER diagram used to describe the logical design of the database, SQL, and the normalization rules for database.

## **1.2 Scope**

Our project is database for company management system. The software is intended to be used by administrators who will manage database entries. The software enables the user to manage company departments, employees, projects, and requests. It also enables the user to manage clients' and suppliers' data. The software is additionally used to generate some statistical information about the company.

## **1.3 Overview**

The report contains project description, ER diagram for the logical design, physical design of the database, SQL script used to make the database tables (DDL), test cases for the software and their results and an end-user guide.

## **2.0 PROJECT DESCRIPTION**

Our project is a company management system. The company is assumed to be a software solutions company, it doesn't develop and sell software for customers but awaits clients who request specific software systems for their companies. The company consists of departments, employees having roles in this department, projects, clients who request these projects, needs (requests) for each department and suppliers who are associated with the company to provide departments needs. For generalization and simplicity, we refer to employees, departments, projects, clients, needs, suppliers as 'entities'.

The project was implemented using java and MySQL. The project is divided into two parts. The first part of our project is the database is required to store all these entities and their attributes, it also stores the employees' roles in their departments and the date when they were assigned to their roles. The second part of our project is the software that contains the CRUD operations on the tables of this database. The software enables the user to add new entities, modify existing entities or remove entities. The software additionally generates some statistical information to the user to help him keep track of company performance.

## **3.0 ROLES OF TEAM MEMBERS**

As a team, we are all expecting to contribute in every stage in this project; but as we are required to predefine the role of each member, we abstractly divided the work among us.

Ahmed Mohamed (team leader): responsible for the logical and physical design of the tables

Ahmed Mamdouh, Ahmed Hesham, Hossam EIDin Khaled: responsible for implementation of the software

Sherif Ashraf: responsible for writing the document, providing test cases and results, providing end-user guide.

## 4.0 LOGICAL DESIGN

We skipped the conceptual design and made the logical design directly. The physical design (exported database file) as well as the script used to create the database (DDL which is also included in the appendix) are uploaded with this document and the software.

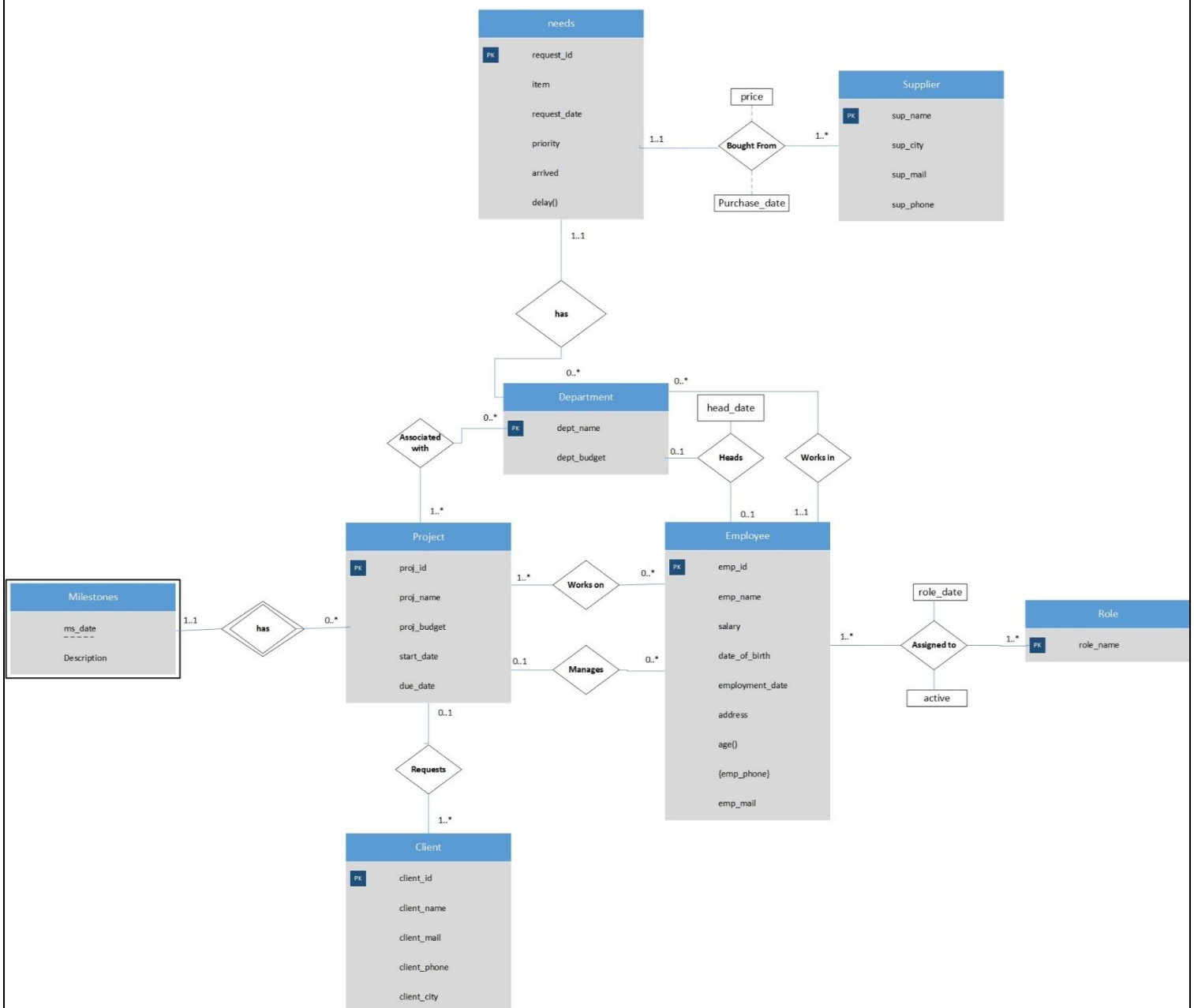


Figure 1: Entity relationship diagram

## 5.0 QUERIES

This section has some of the queries used in the application. The queries chosen features the use of views, join and inner select. The purpose of this section is to show that we applied what we learned from the tutorials and lectures

### 5.1 Join

The following query is used in “dbConnector” class in “searchEmployee” method.

```
select  employee.emp_id,  emp_name,  salary,  birth_date,  employee_mail,
employment_date, dept_name, address, role_name, role_date, active,phone from
employee left join role_emp ON employee.emp_id = role_emp.emp_id left join
emp_phone on employee.emp_id = emp_phone.emp_id where 1 = 1
```

The query joins between the employee table, a table that contains employees' roles that they were assigned to in this company since they were hired, a table that contains the phone numbers of all employees.

The where clause is then concatenated with other conditions based on the user search.



## 5.2 View

The following query is used in the “dbConnector” class in “projectReport” method.

```
create or replace view proj as select * from project natural join proj_dept where 1=1
```

The where clause is also concatenated with other conditions based on user search.

## 5.3 Inner Select

The following query is used in the “dbConnector” class in “projectStat” method.

```
Select MAX (proj_budget) as max_budget, MIN(proj_budget) as min_budget  
, MIN(start_date) as min_sdate, Max(start_date) as max_sdate, MIN(due_date) as  
min_ddate, MAX(due_date) as max_ddate,
```

```
(select proj_name from proj where proj_budget = (select Max(proj_budget) from  
proj) LIMIT 1) as highest,
```

```
(select proj_name from proj where proj_budget = (select MIN (proj_budget) from  
request) LIMIT 1) as lowest,
```

```
(select proj_name from proj where start_date = (select MIN (start_date) from proj)  
LIMIT 1) as earliest,
```

```
(select proj_name from proj where start_date = (select MAX (start_date) from proj)  
order by proj_id desc LIMIT 1) latest,
```

```
(select proj_name from proj where due_date = (select MIN (due_date) from proj)  
LIMIT 1) as nearest, (select proj_name from proj where due_date = (select MAX  
(due_date) from proj) order by proj_id desc LIMIT 1) furthest,  
(Count(distinct(proj_id))) as projects_count from proj
```

The previous query features the use of aggregate functions and the use of subqueries in the select clause

Here is another query to show the use of inner select in where clause

```
update role_emp set active = 0 where 1 = 1 and emp_id in(select emp_id from employee natural join role where 1 = 1)
```

The where clause is also concatenated with other conditions based on user input.

## 6.0 USER GUIDE

To run the program, go to “Company System EXE” folder and run “Company\_Data\_Management\_System.jar”.

### 6.1 Adding

To add data to the database, follow these steps:

1. Select” Add data” from the main menu

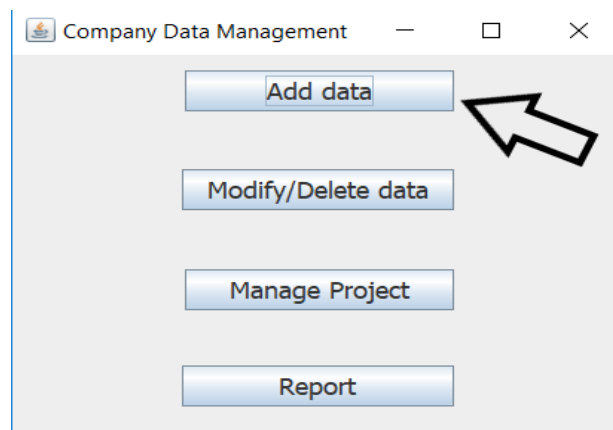


Figure 2: Select add data

2. Select the entity you want to add. In this example, we will choose to add an employee

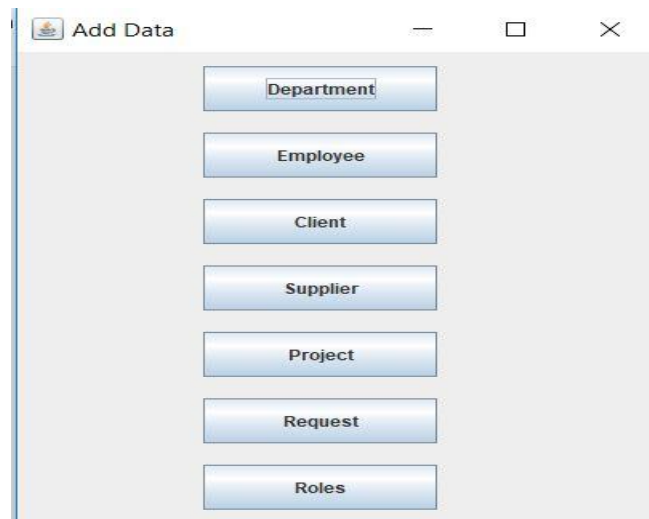


Figure 3: Choose entity

3. Enter the employee data. Click "Add number" button if you want to add another phone number field or submit to add data to the database.

A screenshot of a software window titled "Add Employee". The window contains a form with the following fields and values: "Employee Name:" with "sherif ashraf", "Email:" with "shash@mail.com", "Address:" with "13567 abbas al aqqad", "Birth Date: (Year-Month-Day)" with "2000-11-20", "Department:" with a dropdown menu showing "Database", "Role:" with a dropdown menu showing "junior javaFX developer", "Salary:" with "13000", and "Phone Number:" with two input fields containing "01005809877" and "01096847522". At the bottom of the form, there are two buttons: "Add Number" and "Submit".

Figure 4: Enter details

4. After successful insertion a message should appear saying “Data inserted successfully!”

## 6.2 View, Modify and Delete

This section demonstrates viewing, modifying and deleting data as they are coupled together.

1. Select “Modify/Delete data” from the main menu

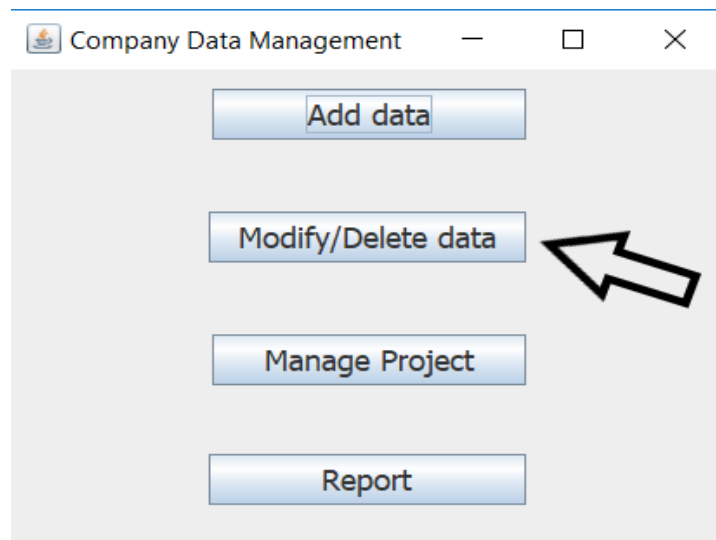


Figure 5: Choose Modify/Delete data

2. Choose an entity. For this example, we will select “Department”

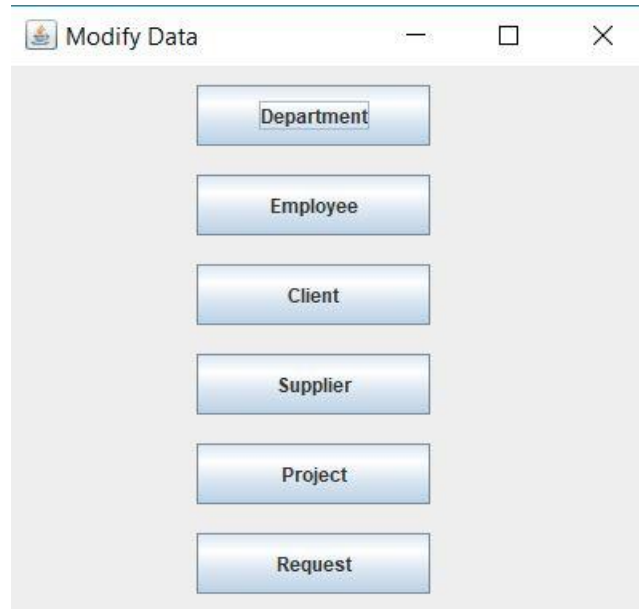


Figure 6: Choose entity to modify

3. First, enter the name of the department you wish to modify or delete. You can click on “view search results” to check the department data first.

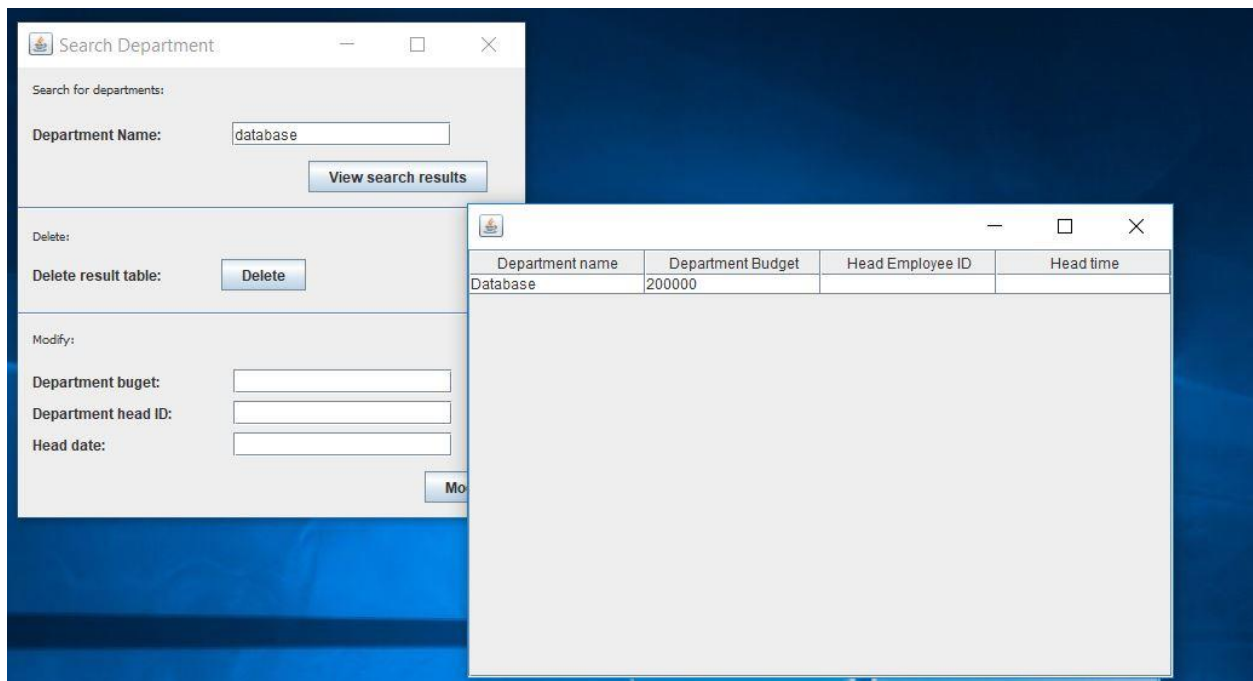
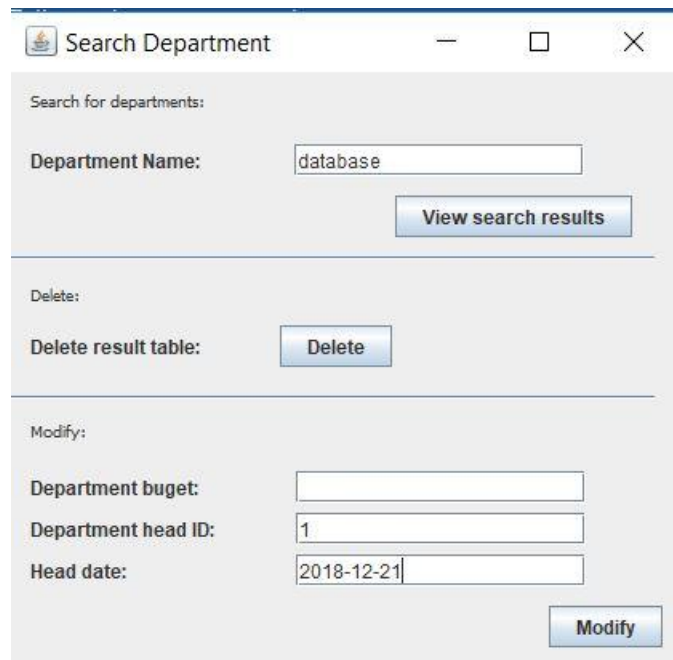


Figure 7: viewing the search result for department” database”

4. To modify the department, enter the new details in the fields at the bottom and click “modify”. A message will appear telling you whether the modification was successful or not.



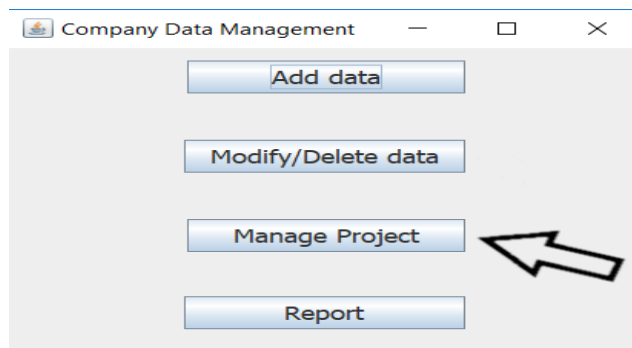
The screenshot shows a window titled "Search Department". It has three main sections: "Search for departments:" with a "Department Name:" field containing "database" and a "View search results" button; "Delete:" with a "Delete result table:" label and a "Delete" button; and "Modify:" with fields for "Department budget:", "Department head ID:" (containing "1"), and "Head date:" (containing "2018-12-21"), along with a "Modify" button.

Figure 8: Modifying the department

5. To delete the department, simply click on the delete button in the middle of the screen

## 6.3 Manage Projects

First, select “Manage project” from the main menu



The screenshot shows a window titled "Company Data Management". It contains four buttons stacked vertically: "Add data", "Modify/Delete data", "Manage Project", and "Report". A large black arrow points to the "Manage Project" button.

Figure 9: Select manage project

### 6.3.1 project manager

To assign a project manager select “Assign project manager”. Enter the project id and employee id and click “assign”.

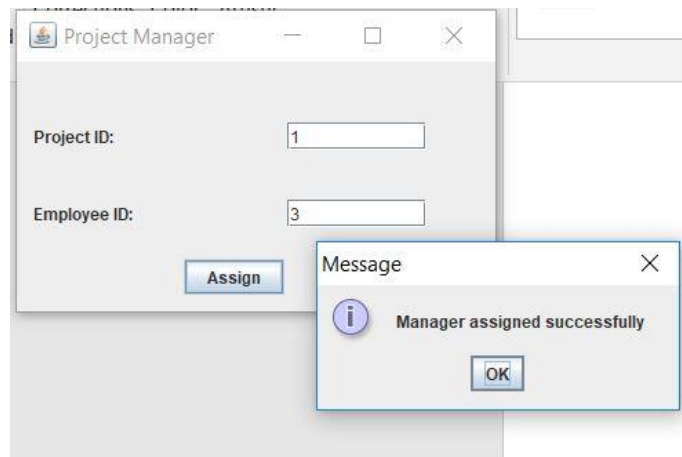


Figure 10: Assign project manager

### 6.3.2 project employees

Select” Project Employees”. Enter the project id, you can click “view employees” to view the assigned employees

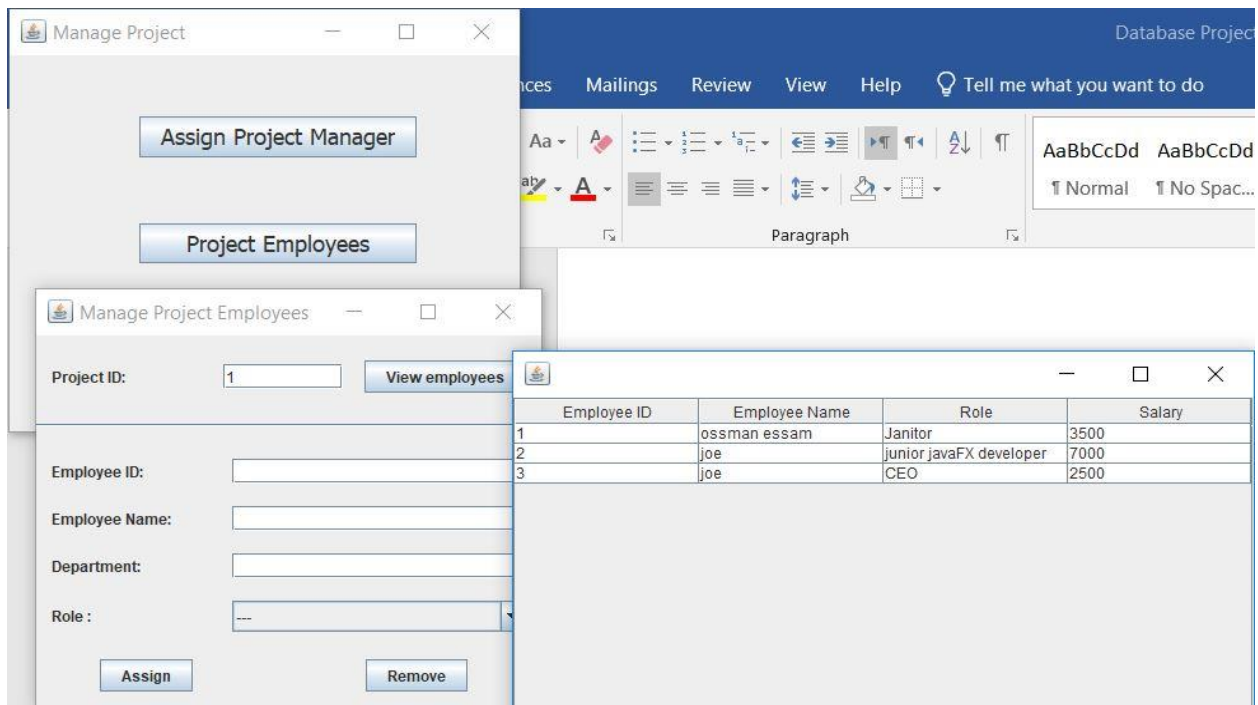


Figure 11: Viewing employees assigned to the project

To assign or remove employees from this project, enter the data you wish to select the employees by then click “Assign” or “Remove”

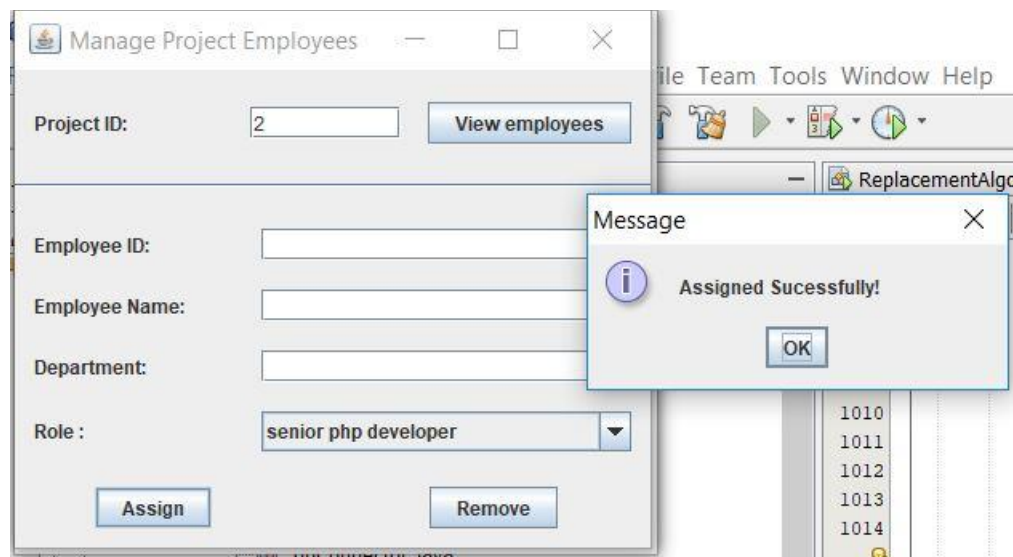


Figure 12: Assigning employees to a project based on role



### 6.3.3 milestones

1. Select “Milestones” from the main menu
2. Enter the id for the project associated with this milestone, milestone date and description
3. Click on “Submit” button

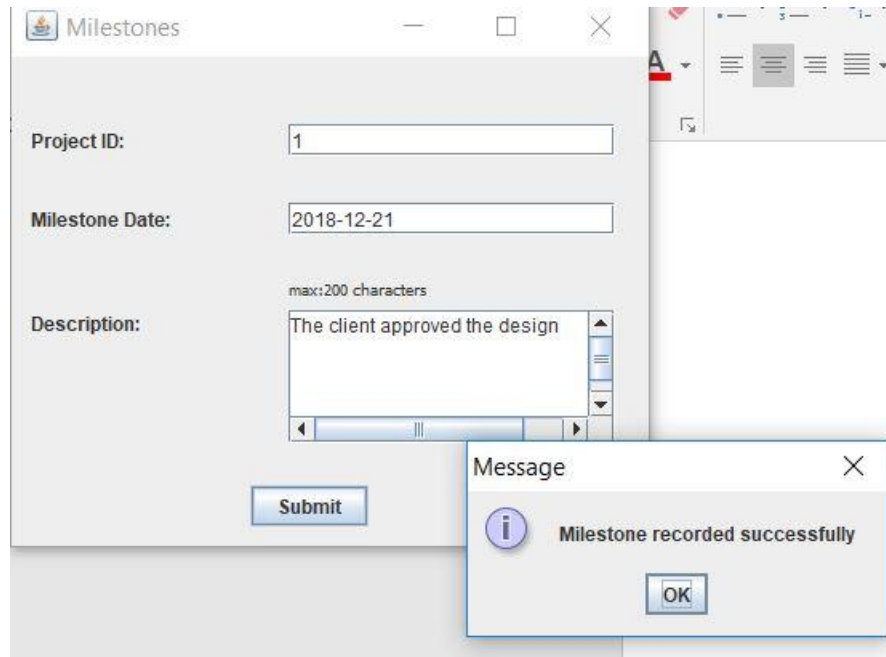


Figure 13: Adding a milestone to a certain project

## 6.4 Generate Reports

1. Select “Reports” from the main menu

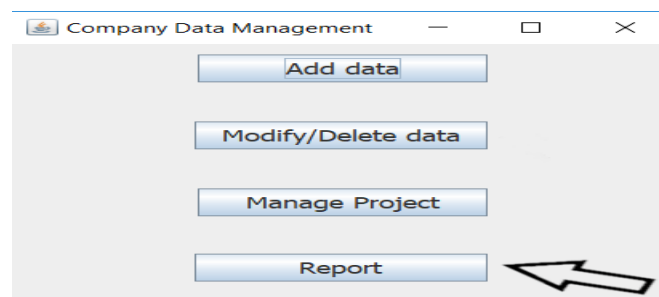


Figure 14: Generating reports

2. Select the entity you wish to generate a report about. In this example, we chose “Request”
3. Enter the details of the requests you want to include in the report and click “submit”.

The screenshot displays three overlapping windows from a software application:

- Request Report**: A search form on the left with fields for Request ID, Item Name, Priority (dropdown), Arrived (radio buttons: Yes, No, Any), Department Name (text: java), Supplier Name, Price (text: 500, radio buttons: Less, More), and Purchase Date (text, radio buttons: Before, After). A Submit button is at the bottom.
- Requests Table**: A central table showing search results.
 

| Request... | Item na... | Request... | Priority | Arrived | Depart... | Supplier | Price | Purchas... | Delay L... |
|------------|------------|------------|----------|---------|-----------|----------|-------|------------|------------|
| 2          | Air Con... | 2018-12... | Low      | true    | Java      | soheil   | 10100 | 2018-12... | 8          |
| 3          | HDMI       | 2018-12... | low      | false   | Java      | soheil   | 12000 | 2018-12... | 3          |
| 4          | laptops    | 2018-10... |          | false   | Java      | soheil   | 5000  | 2018-11... | 16         |
| 5          | mouse      | 2018-12... |          | false   | Java      | soheil   | 10000 | 2018-12... | 1          |
- Requests Stats**: A statistics panel on the right showing:
  - Number of Requests: 4
  - Average Price: 9275
  - Most Expensive Item: HDMI, It's Price: 12000
  - Cheapest Item: laptops, It's Price: 5000
  - Oldest Purchased Item: laptops, Date: 2018-11-01
  - Newest Purchased Item: mouse, Date: 2018-12-15

Figure 15: Report generation

As shown in the figure, a table containing the list of requests searched for is generated as well as a form containing the statistics.

Note that the date format entered in any text field should be YYYY-MM-DD.

Even a number less than ten should be written in two digits

Ex: “2018-05-07” not “2018-5-7”

## APPENDIX

Here is the data definition language

```
CREATE TABLE department(
```

```
    dept_name      varchar(50),
```

```
    dept_budget    int unsigned,
```

```
    head           int UNIQUE,
```

```
    head_date      date,
```

```
    PRIMARY KEY(dept_name)
```

```
);
```

```
CREATE TABLE employee(
```

```
    emp_id         int AUTO_INCREMENT,
```

```
    emp_name       varchar(50) NOT NULL,
```

```
    salary         int unsigned,
```

```
    birth_date     date NOT NULL,
```

```
    employee_mail  varchar(50) UNIQUE NOT NULL,
```

```
    employment_date date NOT NULL,
```

```
    dept_name      varchar(50) NOT NULL,
```

```
    address        varchar(50),
```

```
    PRIMARY KEY(emp_id),
```

```
    FOREIGN KEY (dept_name) REFERENCES department(dept_name) ON DELETE  
    CASCADE ON UPDATE CASCADE
```

```
);
```

```
ALTER TABLE department
  ADD FOREIGN KEY (head)
  REFERENCES employee(emp_id) ON DELETE SET NULL ON UPDATE CASCADE;
```

```
CREATE TABLE client(
  client_id      int AUTO_INCREMENT,
  client_name    varchar(50) NOT NULL,
  client_mail    varchar(50) UNIQUE NOT NULL,
  client_phone   varchar(11) UNIQUE,
  client_city    varchar(20),

  PRIMARY KEY(client_id)
);
```

```
CREATE TABLE project(
  proj_id       int AUTO_INCREMENT,
  proj_name     varchar(50) NOT NULL,
  proj_budget   int unsigned,
  start_date    date NOT NULL,
  due_date      date,
  client_id     int,
  manager       int,
  PRIMARY KEY(proj_id),
  FOREIGN KEY(client_id) REFERENCES client(client_id) ON DELETE SET NULL
  ON UPDATE CASCADE,
  FOREIGN KEY(manager) REFERENCES employee(emp_id) ON DELETE SET
  NULL ON UPDATE CASCADE
);
```

```
CREATE TABLE supplier(  
    sup_name      varchar(50),  
    sup_city      varchar(20),  
    sup_mail      varchar(50) UNIQUE NOT NULL,  
    sup_phone     varchar(11) UNIQUE,  
    PRIMARY KEY(sup_name)  
);
```

```
CREATE TABLE needs(  
    request_id    int AUTO_INCREMENT,  
    item          varchar(100) NOT NULL,  
    request_date  date,  
    priority      varchar(10),  
    arrived       boolean,  
    dept_name     varchar(50) NOT NULL,  
    sup_name      varchar(50),  
    price         int unsigned,  
    purchase_date date,  
    PRIMARY KEY(request_id),  
    FOREIGN KEY(dept_name) REFERENCES department(dept_name) ON DELETE  
CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(sup_name) REFERENCES supplier(sup_name) ON DELETE SET  
NULL ON UPDATE CASCADE  
);
```

```
CREATE TABLE role(  
    role_name    varchar(50),  
    PRIMARY KEY(role_name)  
);
```

```
CREATE TABLE proj_dept(  
    proj_id      int,  
    dept_name    varchar(50),  
    PRIMARY KEY(proj_id,dept_name),  
    FOREIGN KEY(proj_id) REFERENCES project(proj_id) ON DELETE CASCADE  
ON UPDATE CASCADE,  
    FOREIGN KEY(dept_name) REFERENCES department(dept_name) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE proj_emp(  
    proj_id      int,  
    emp_id      int,  
    PRIMARY KEY(proj_id,emp_id),  
    FOREIGN KEY(proj_id) REFERENCES project(proj_id) ON DELETE CASCADE  
ON UPDATE CASCADE,  
    FOREIGN KEY(emp_id) REFERENCES employee(emp_id) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```

CREATE TABLE role_emp(
    emp_id          int,
    role_name       varchar(50),
    role_date       date,
    active          boolean NOT NULL,
    PRIMARY KEY(emp_id,role_name,role_date),
    FOREIGN KEY(role_name) REFERENCES role(role_name) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(emp_id) REFERENCES employee(emp_id) ON DELETE
CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE emp_phone(

    emp_id          int,
    phone           varchar(11) UNIQUE,
    PRIMARY KEY(emp_id,phone),
    FOREIGN KEY(emp_id) REFERENCES employee(emp_id) ON DELETE
CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE milestone(
    proj_id         int,
    ms_date         date,
    description      varchar(200),
    PRIMARY KEY(proj_id,ms_date),
    FOREIGN KEY(proj_id) REFERENCES project(proj_id) ON DELETE CASCADE
ON UPDATE CASCADE
);

```