

**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.

i

TABLE OF CONTENTS

EXECUIVE SUMMARY………………………………………………………………………....i

LIST OF FIGURES………………...…………………………………………………………...iii

1. INTRODUCTION......……………………………………………………………………….1
2. PROJECT DESCRIPTION………………………………………………………………...2
3. ROLES OF TEAM MEMBERS…………………………………………………………….2
4. LOGICAL DESIGN ………..………………………………………………………………..3
5. 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

ii

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

iii

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.

1

1. 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 ElDin Khaled: responsible for implementation of the software

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

2

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.

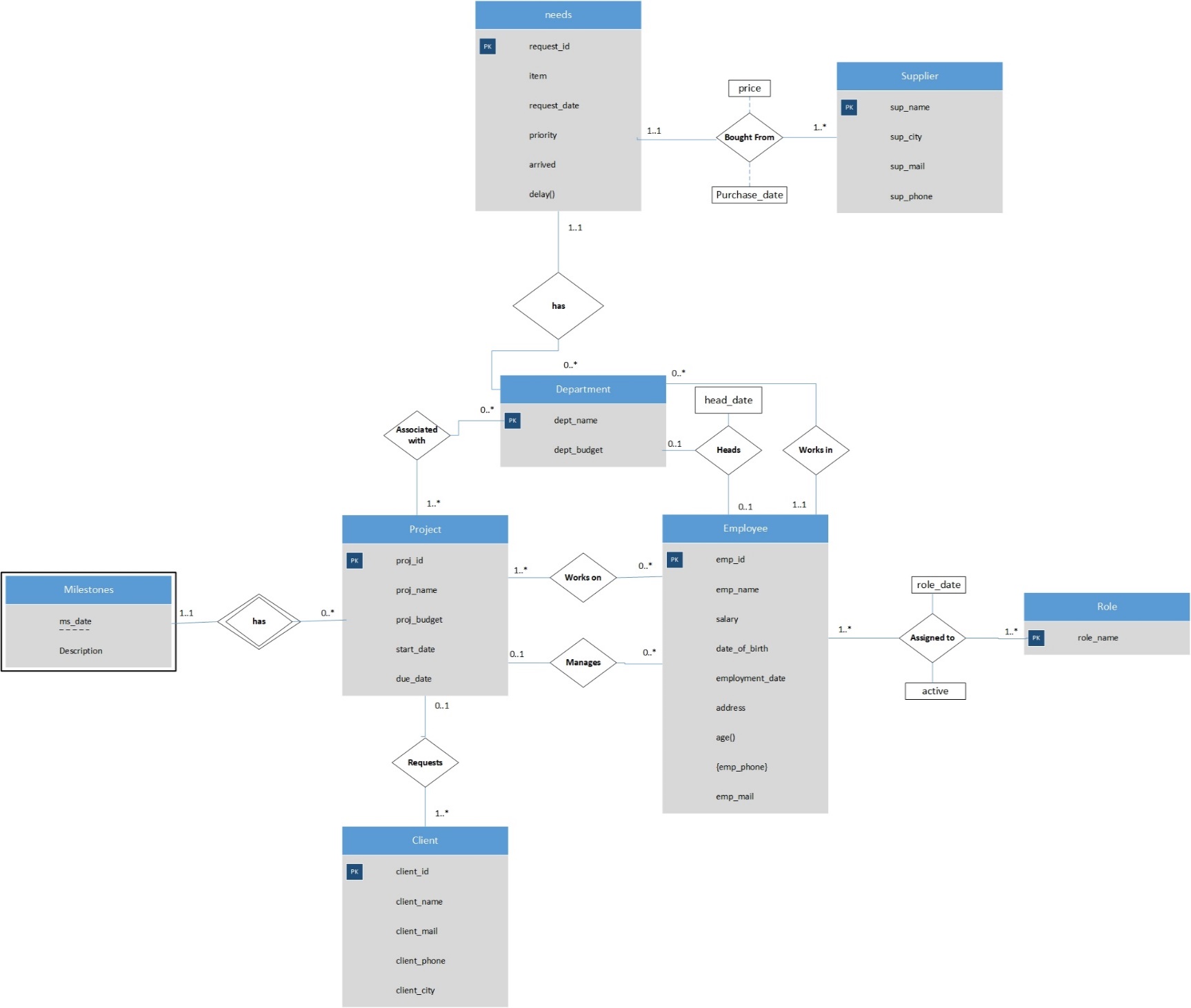


Figure 1: Entity relationship diagram

3

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.

4

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 were 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

5

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

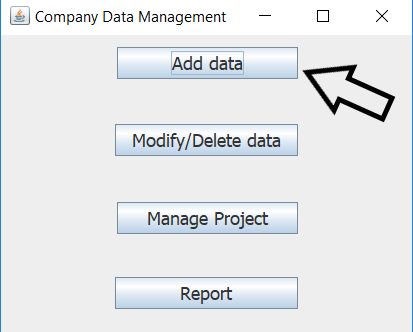


Figure 2: Select add data

6

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

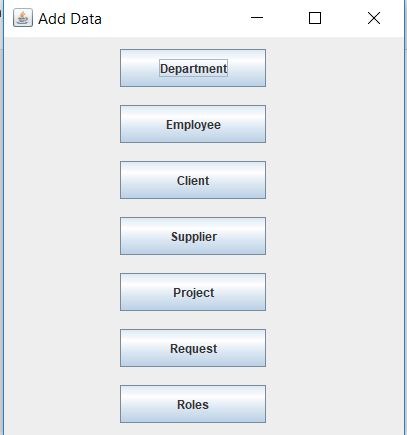


Figure 3: Choose entity

1. 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.

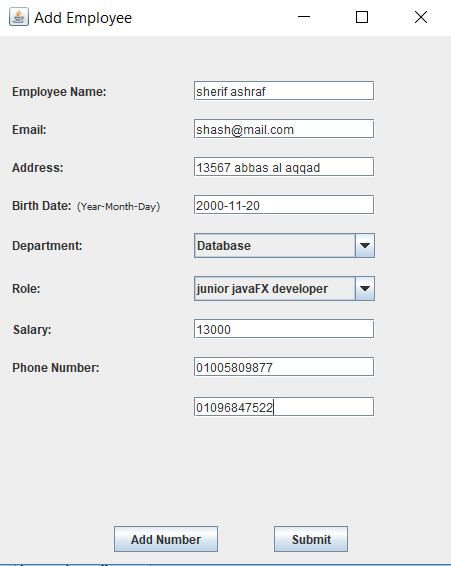


Figure 4: Enter details

7

1. 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

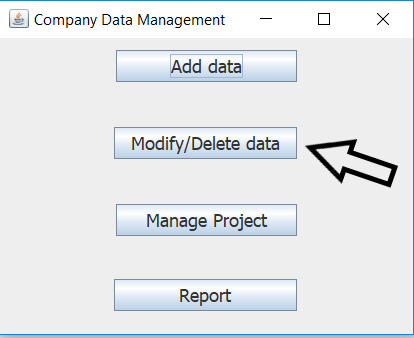


Figure 5: Choose Modify/Delete data

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

8

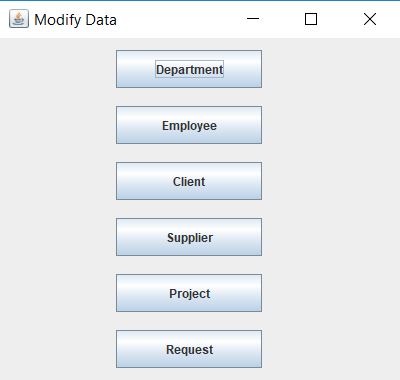


Figure 6: Choose entity to modify

1. 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.

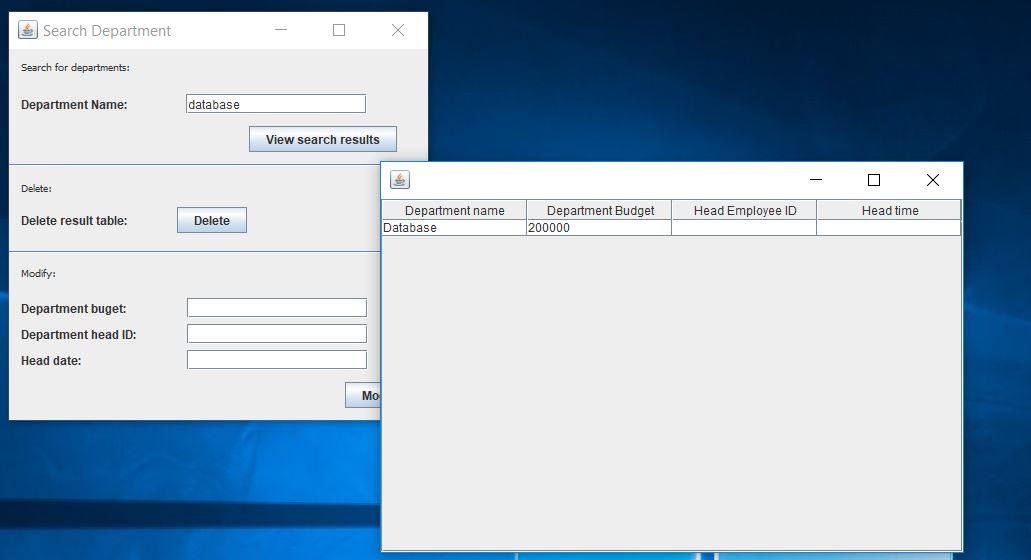


Figure 7: viewing the search result for department” database”

9

1. 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.

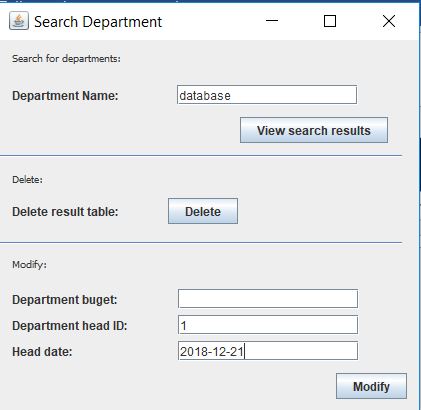


Figure 8: Modifying the department

1. 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

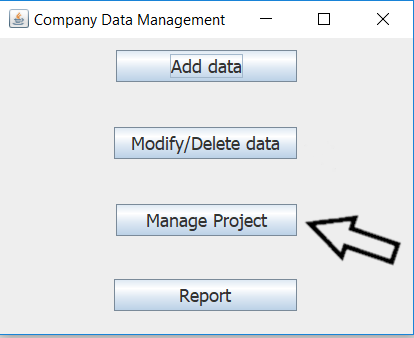


Figure 9: Select manage project

10

6.3.1 project manager

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

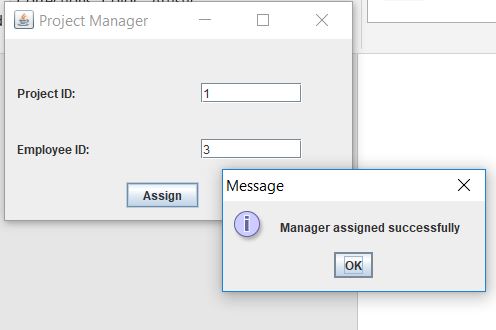


Figure 10: Assign project manager

6.3.2 project employees

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

11

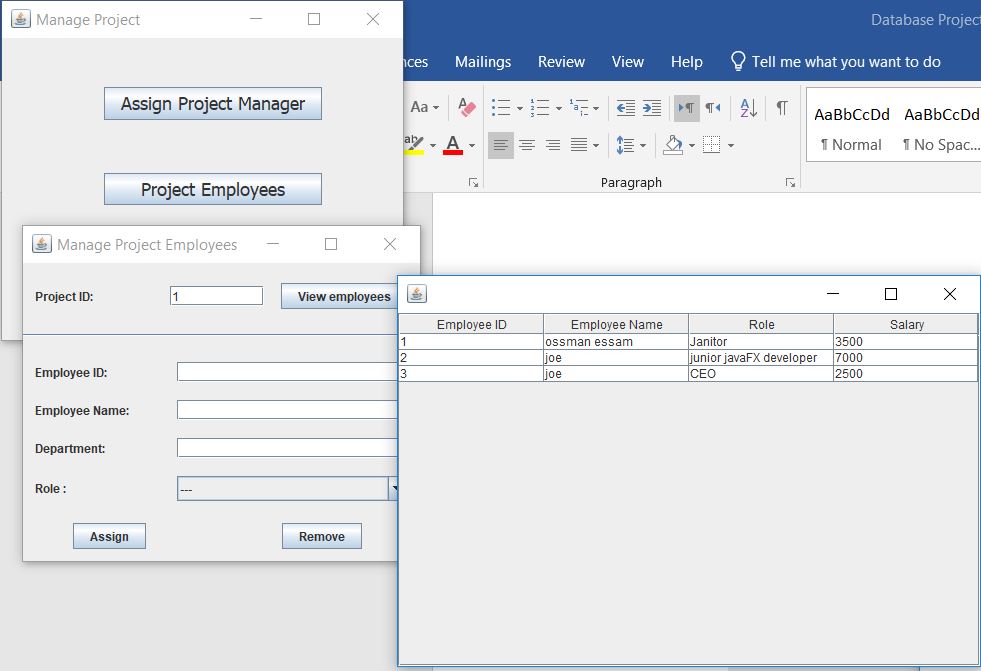


Figure 11: Viewing employees assigned to the project

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

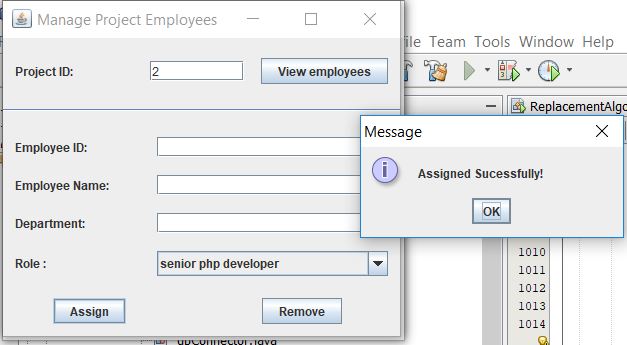


Figure 12: Assigning employees to a project based on role

12

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

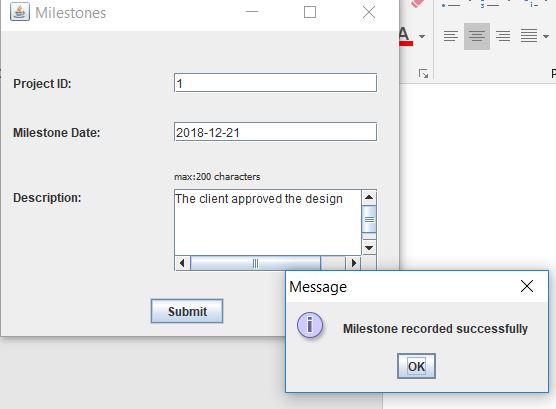


Figure 13: Adding a milestone to a certain project

6.4 Generate Reports

1. Select “Reports” from the main menu

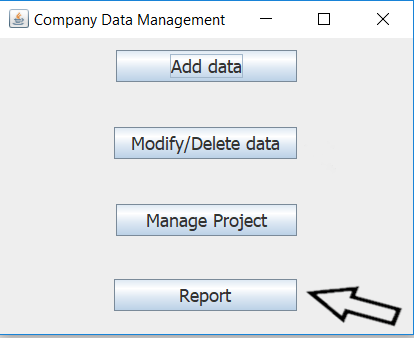


Figure 14: Generating reports

13

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

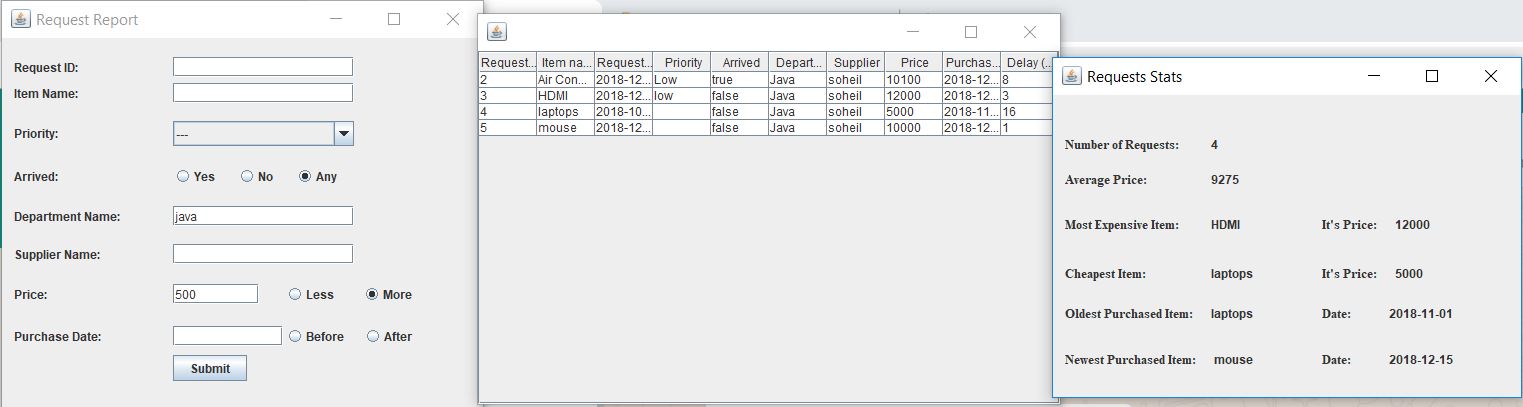


Figure 15: Report generation

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”

14

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

);

15

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

);

16

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

);

17

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

);

18

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

);

19