

## Assignment # 5

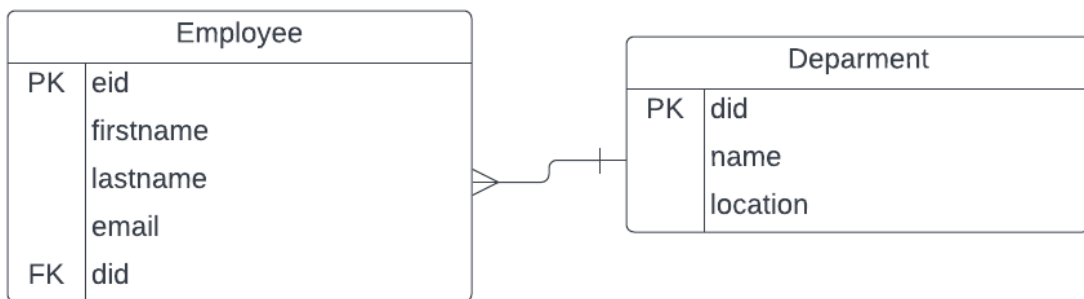
**Due Date Tuesday, April 19, 2022, @11: 59 PM**

In this Assignment, you will modify **the employee management app (emApp)** app's backend that is currently using file-based repository to read and write data with mongo database.

### Preparation

1. Sync cmps350-lab repo to get the Assignment files
2. Open the **assignment5/emApp** project
3. The project contains the complete implementation of assignment 4
4. Run **npm install** to install all the necessary packages.

### Implementing the Database



In this part you should implement the entity schemas and the database repository for the **emApp**. The database repository should implement the same functionality as the file-based repository provided in the base solution, with four additional methods of department.

Note : You can not use any of the array function such as **filter, map, reduce** etc... to filter or aggregate values. You should use the **mongoose model methods** to get, delete, update, aggregate data. Not using mongoose functions means you will get 0 for that section.

Method	Description
getEmployees()	Return all employees from the database
getEmployee(eid)	Return a single employee that has the same employee id(eid)
addEmployee(newEmployee)	Adds new employee to the database
updateEmployee(updatedEmployee)	Updates existing employee
deleteEmployee(eid)	Deletes specific employee from the database
deleteAllEmployees()	Remove all the employees saved inside the database

<p>getDepartmentStatistics()</p> <p>[you must <b>use aggregate function</b> to implement this. Using array functions such as <b>reduce, map</b> will not be accepted]</p>	<p>Returns an object containing the total number of employees in the department</p> <p>Example object should look like</p> <pre> {   "did": 103,   "name": "HR",   "location": "C08",   "numberOfEmployees": 1 }, {   "did": 104,   "name": "Health and safety",   "location": "C06",   "numberOfEmployees": 3 }, </pre>
<p>getDepartmentEmployees()</p> <p>[you must <b>use mongoose function</b> to implement this. Using array functions such as <b>reduce, map</b> will not be accepted]</p>	<p>Returns the all the departments and their employees.</p> <p>Example object should look like</p> <pre> {   "did": 104,   "name": "Health and safety",   "location": "C06",   "employees": [     {       "eid": 1649454019810,       "firstName": "Mohamed",       "lastName": "Abdou",       "email": "abdou@gmail.com",       "did": 104     },     {       "eid": 1649715536980,       "firstName": "Abdulahi",       "lastName": "Mohamed",       "email": "abdala.baaaji@gmail.com",       "did": "104"     },     {       "eid": 3444445,       "firstName": "Ali",       "lastName": "Essa",       "email": "abdala.baaaji@gmail.com",       "did": "104"     }   ] }, </pre>

- Test your API's using **Postman and the Client side application**

After you complete the Assignment, fill in the ***TestingDoc-Grading-Sheet.docx*** and save it inside the ***Assignment5*** folder. Push your work to the Github repository.