**LMS (Leave Management System) Case Study:**

We need to build a leave management application. This is an application where employees

\* can view their personal details

\* can apply for leave

\* can check the history of their leave applications

**In Database:**

[DBname : lms , username: root, password: Password123]

Create two tables: Employee and LeaveDetails.

**EMPLOYEE TABLE DESCRIPTION**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** |
| EMP\_ID | INT(10) | NO | PRIMARY KEY |
| EMP\_NAME | VARCHAR(40) | NO |  |
| EMP\_MGR\_ID | INT(10) | YES |  |
| EMP\_LEV\_BALANCE | INT(10) | YES |  |
| EMP\_DEP\_NAME | VARCHAR (20) | YES |  |

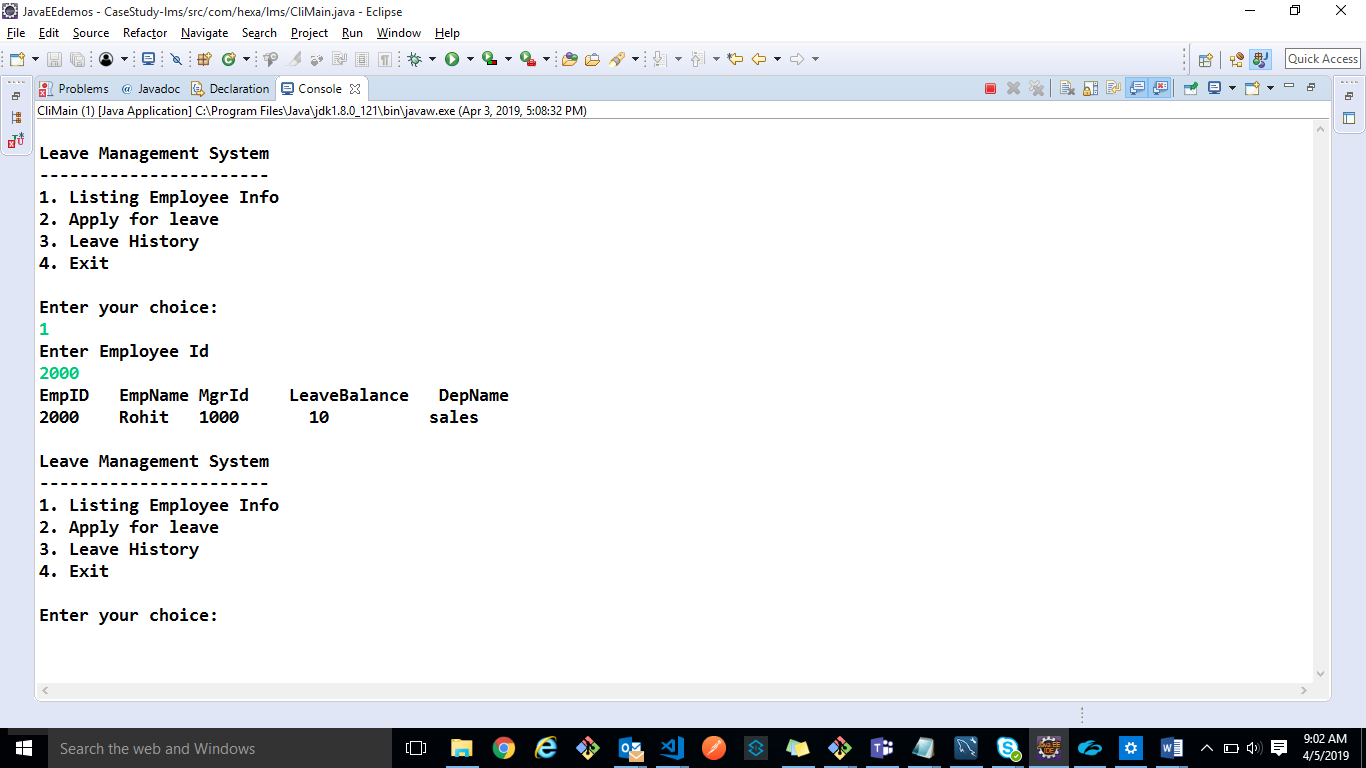
**LEAVEDETAILS TABLE DESCRIPTION:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** |
| LEV\_ID | INT(10) | NO | PRIMARY KEY |  |
| LEV\_DATE\_APPLIED | DATETIME | NO |  | CURRENT\_TIMESTAMP |
| LEV\_START\_DATE | DATE | NO |  |  |
| LEV\_END\_DATE | DATE | NO |  |  |
| LEV\_NO\_OF\_DAYS | INT(10) | NO |  |  |
| LEV\_REASON | VARCHAR(100) | YES |  |  |
| EMP\_ID | INT(10) | NO | FOREIGN KEY |  |

**Project Description:**

1. **Can view their personal details**

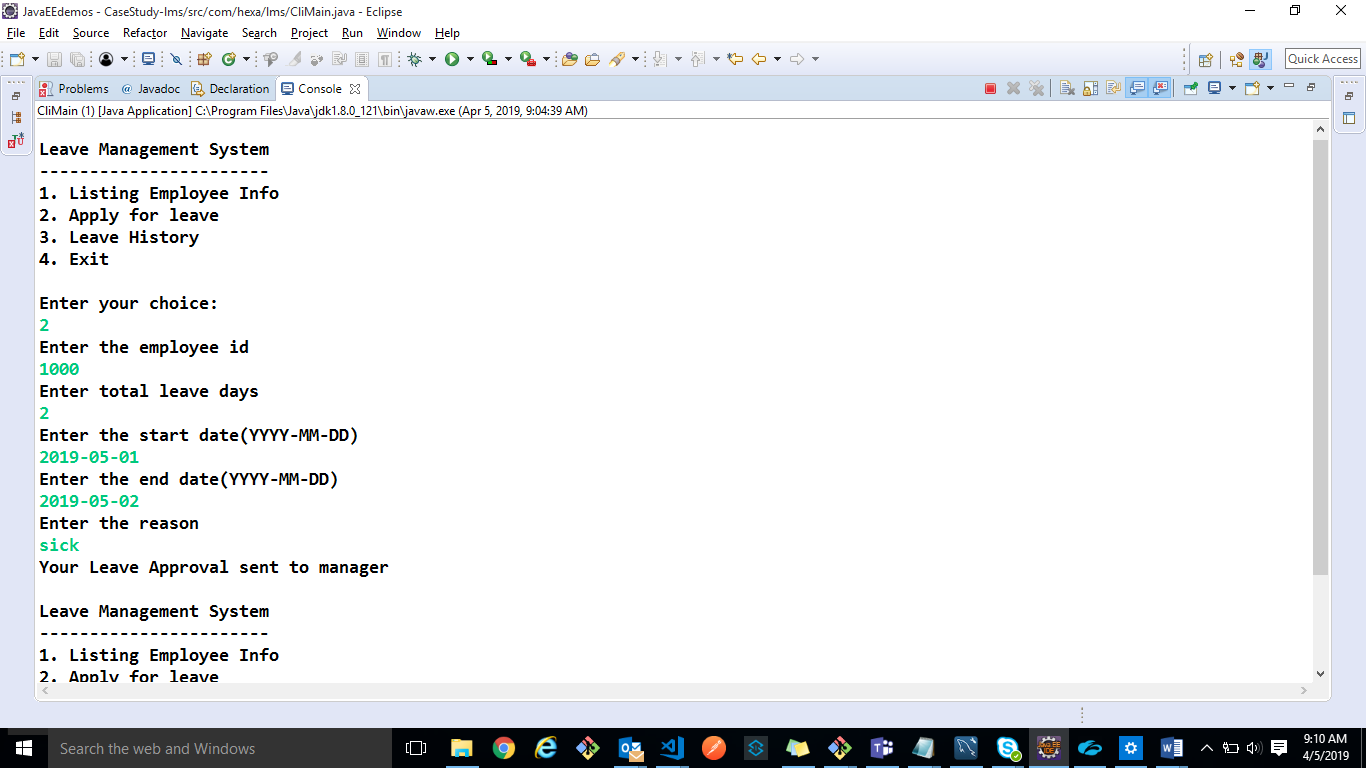
- Get empid from the user and display their employeeid, employeename, managerid, leavebalance and departmentname.



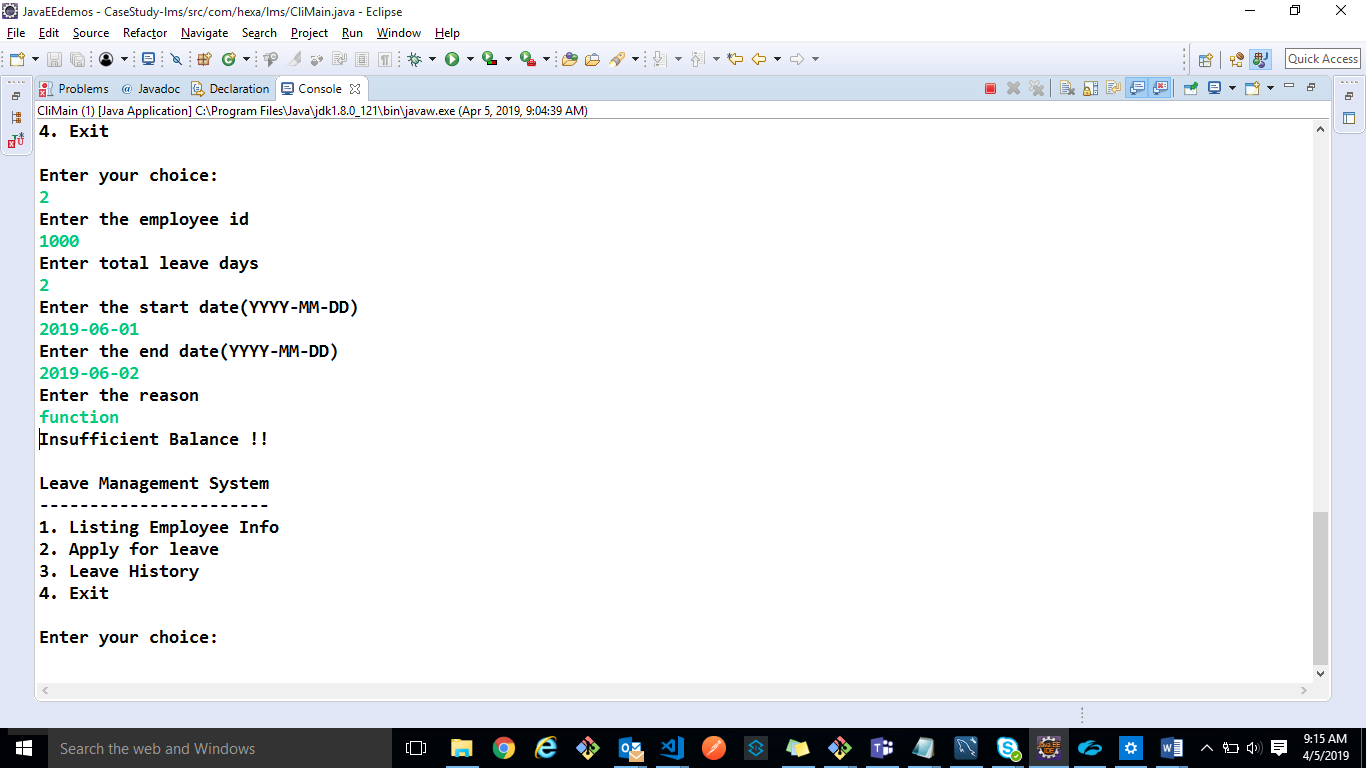
2. **Can apply for leave**

- Get employeeid, total leave days, startdate, enddate, reason for leave from the user and based on their available leave balance, deduct it in the employees table, leave application need to be added in leave details table and show the message as “**your Leave Approval sent to manager”** otherwise show “**insufficient leave balance”** message.

For example, Check the below output for Employee 1000 who applies leave for 2 days.

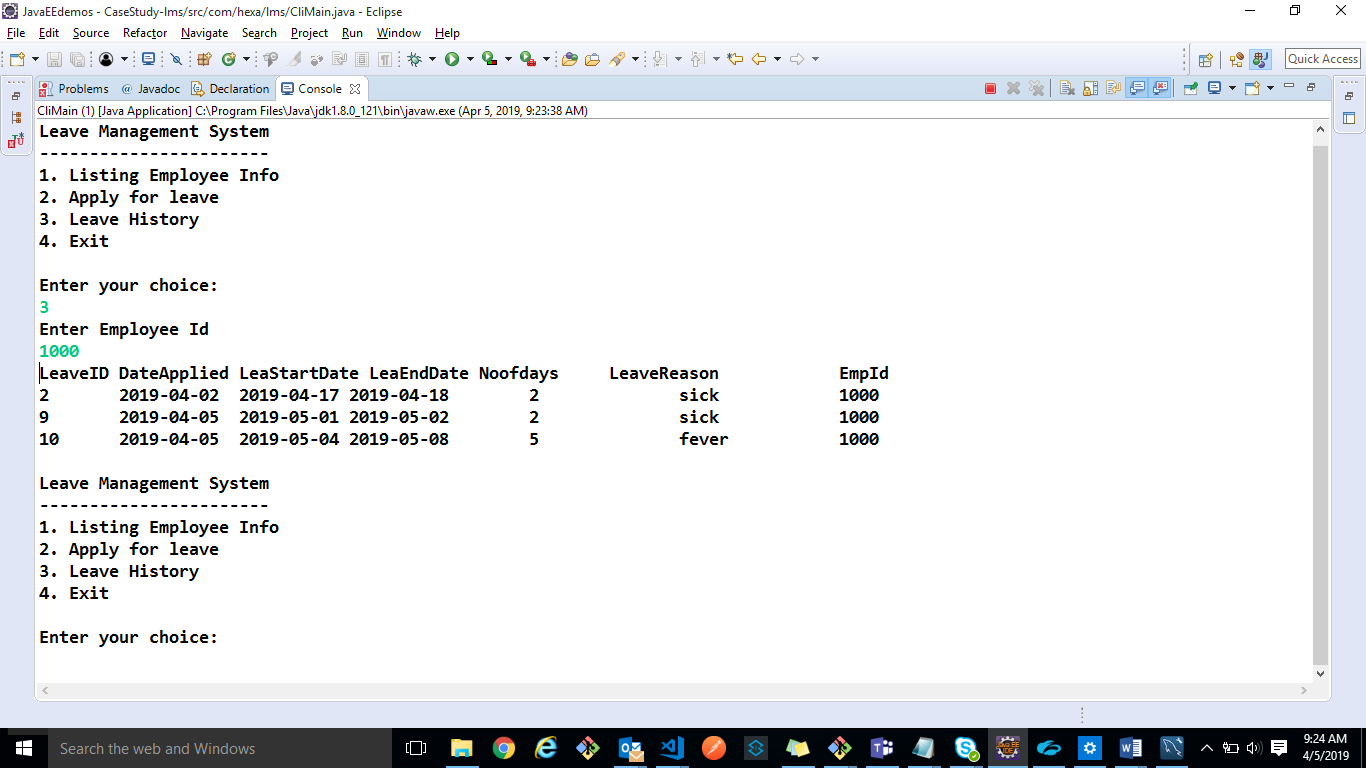


* Check the below output Employee 1000, if the leave balance is 0 day.



3**. Can check their history of their leave applications**

- Get the employee id from the user and list the employee’s leave history details (Leaveid, dateapplied, leavestartdate, leaveenddate, noofdays, leavereason, empid).



**How to start?**

Given Leave Management System project, create two java classes Employee and LeaveDetails.

**In Employee class:**

1. Add the given attributes:

* empId
* empName
* empMgrId
* empLevBalance
* empDepName

1. add the constructors, getters and setters for that attributes.
2. Then create two methods:
3. **Method name: listById**

**Parameter: empId**

**Return type: Employee**

**Description:** The method **listById** return the employee personal details (employeeid, employeename, managerid, leavebalance and departname) based on empId.

1. **Method name: applyLeave**

**Parameters: levStartDate,levEndDate,levNoOfDays,**

**levReason,levEmpId**

**Return type: String**

**Description:** The method **applyLeave** need to check the condition, that the given no of leave days is greater than and equal to their available leave balance or not. If that condition satisfies deduct the leave balance in employee table, add the leave application details in leave details table and return the message as “**your Leave Approval sent to manager”.** If not return“**insufficient leave balance”.**

**In LeaveDetails class:**

1. Add the given attributes:

* levId
* levDateApplied
* levStartDate
* levEndDate
* levNoOfDays
* levReason
* levEmpId

1. add the constructors, getters and setters for that attributes.
2. Then create a method:

**Method name: listById**

**Parameter: empId**

**Return type: List<LeaveDetails>**

**Description:** The method **listById** return the list of employee’s leave history details (Leaveid, dateapplied, leavestartdate, leaveenddate, noofdays, leavereason, empid) based on empId.

**DbConnection and CliMain class description:**

In the **LeaveManagementSystem** project has **DbConnection** and **CliMain** class.

* To make connection with database, call **getConnect**() method from **DbConnection** class.
* The main method is defined in **CliMain** class.

And the class has three empty methods,

1. **listEmployeeDetail()** [for displaying employee personal details]

* Get required inputs from user, call the **listById** method from **employee** **class**, pass the required parameters and print the return output**.**

1. **applyForLev()** [ for applying leave]

* Get the required inputs from user, call the **applyLeave** method from **employee class,** pass the required parameters and print the return output**.**

1. **leaveHistory()** [for displaying employee leave history data]

* Get the required inputs from user, call the **listById** from **leave details class,** pass the required parameters and print the return output.