HRMS

To be developed

**--------------------------Overview-----------------------------------------------------**

HRMS is specifically designed for the HR department of Capgemini which stores employee salary and designation details year wise if employee is either promoted with designation or appraised with salary. Schema definition is provided in the following diagram. This application will be developed by group of 7 people. Application will have 2 types of actors [Admin/Employee] as follows who can perform following operation

**Group Size: 7**

**-----------------------------------Admin1----------------------------------------**

**Admin1: -**

1. Admin- Login with admin role.
2. Admin- Find list of employees joined in last 10 years
3. Admin- Find count of employees joined in last 10 years
4. Admin- Find list of employees joined after 2005.
5. Admin -Find count of employees joined after 2005
6. Admin- Assign department to employee from start to end date.
7. Admin- Assign manager to specific department for specific duration.
8. Admin -Assign employee with title [designation] from start to end date. [Assuming that by default employee will be eligible for promotion after 5 years]
9. Admin- Add new department.
10. Admin- Add new employee
11. Admin- Delete employee once employee leave the organization by empid.
12. Admin- Update the salary for employee when employee is appraised each year depending on rating.

[1: 10000Rs - 2:7000Rs - 3:5000Rs 4:0Rs – 5:0Rs]

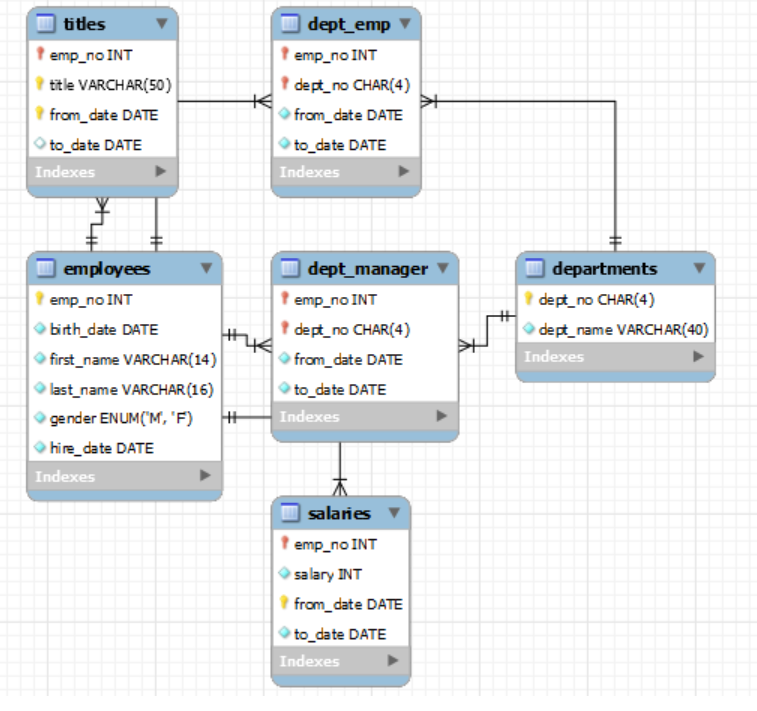
If employee is promoted as well, then update salary by 100000.

1. Admin- Find Employees with 5year/10year/15year/20Year/25Year/30Year experience for getting awarded.
2. Admin- Find count of women employees for sending mail for giving 1000 Rs gift voucher for women’s day celebration
3. Admin- Find list of women employees for sending mail for women’s day invitation so that 1000Rs gift voucher can be given to all .

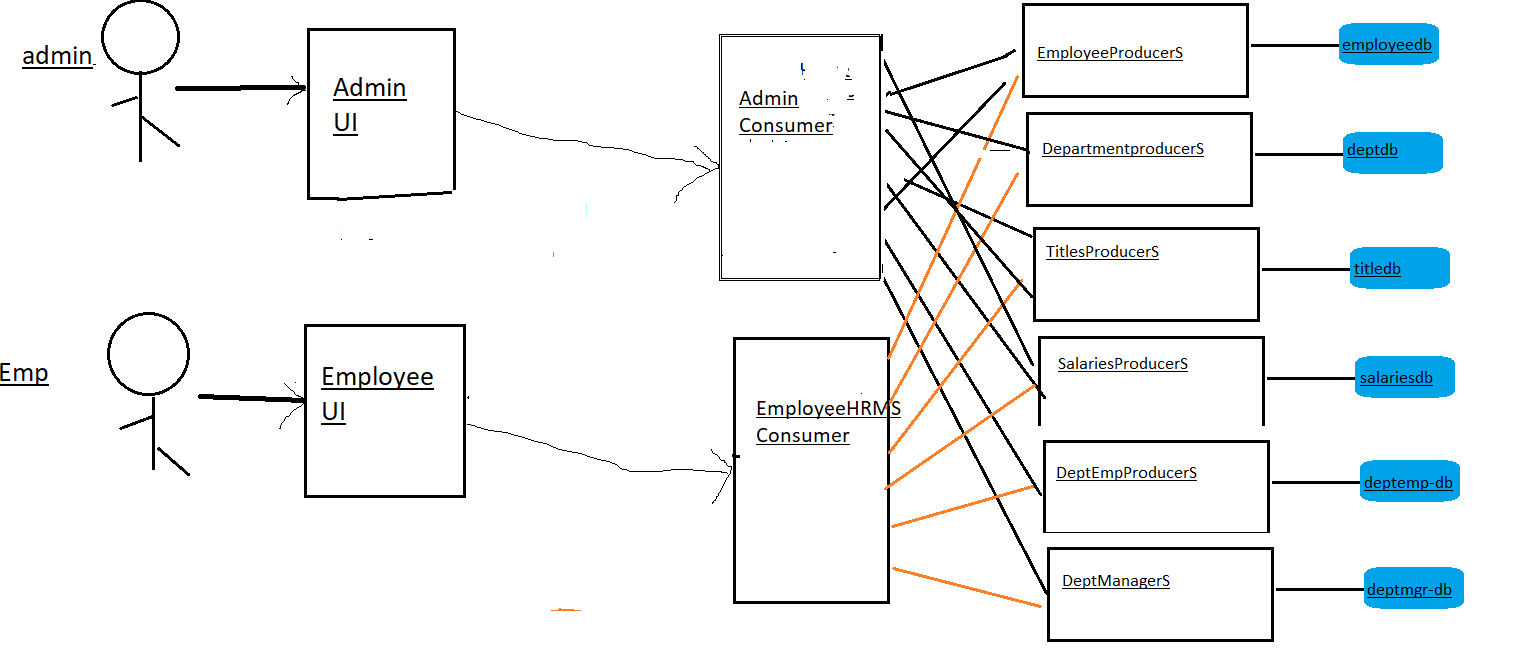
**------------------------------------------Employee-------------------------------**

1. Employee- Login with employee role.
2. Employee - Update last name for him/herself
3. Employee - Get details of employees [id, name, dob, and department name] who are manager.
4. Employee - Get list of employees working in specific department in specific year.
5. Employee Check department wise average/max/min salary
6. Employee - Check designation/title wise average /max/min salary
7. Employee: Find mid age employees. [Employees in 50’s]
8. Employee: Check what all designations are provided in Capgemini.
9. Employee: Print empid, name, hire date of those who have become manager after specific from date.
10. Employee: Check details of those employee who are working as specific title [ ex. “Technique Leader”]

**----------------------------Schema- ER Diagram--------------------------------------------------**



**--------------------------Flow Diagram-------------------------------------------------------**

****

**----------------------------------------------Notes---------------------------------------------------------------**

**Schema Location: [Some share Folder Link] [C:\Users\vaissriv\Downloads\test\_db-master]**

**Step1:**

1. All Rest API request and response format must be in JSON.
2. 200, 201, 400, 404, 401, 500, 505 – use appropriate http status code in your response object.
3. Handle the Rest exceptions globally using Rest exception handler for all endpoints.
4. Sample Error Responses are shown in below table, participants to identify error paths for all the Endpoints.
5. Test your Rest Api using Junit and Mockito.
6. Participants should not alter database schema under any circumstances except wherever changes are mentioned explicitly.
7. All GET and POST requests for customer are secured

**------------------------------------------------------Participants------------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | employees |  |  |
| /api/v1/employees/ | POST | Add new employee object in DB | String: New employee added Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/employees/ | GET | Fetch all employees. | Collection of employees |  |
| /api/v1/employees/lastname/{ln} | GET | Search Employee by Last Name | Collection of employees |  |
| /api/v1/employees/firstname/{fn} | GET | Search employees by First Name | Collection of employees |  |
| /api/v1/employees/lastname/{empno} | PUT | Update Last Name of an employee given empno | Employee | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/employees /firstname/{empno} | PUT | Update First Name of an employee given empno | Employee | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/employees /hiredate/{empno} | PUT | Update hiredate for given empno | Updated Employee |  |
| /api/v1/employees /birthdate/{empno} | PUT | Update birthdate for given empmo | Updated Employee |  |
| /api/v1/employees /gender/{gender} | GET | Search employees by gender | Collection of employees |  |
| /api/v1/employees /empid/{empno} | GET | Search employees by empno | Single Employee Object |  |
| /api/v1/employees /hiredate/{hiredate} | GET | Search employees by hire date | Collection of employees |  |
| /api/v1/employees /birthdate/{birthdate} | GET | Search employees by birth date | Collection of employees |  |

**---------------------------------participant--------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | departments |  |  |
| /api/v1/departments/ | POST | Add new department | String: New department added successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/departments | GET | Fetch all departments | Collection of departments |  |
| /api/v1/departments/{deptno} | GET | Search department by deptno | Single Department Object |  |
| /api/v1/departments/name/{name} | GET | Search department by name | Single Department Object |  |
| /api/v1/departments/{ deptno } | PUT | Update department by department deptno | String: Department updated successfully. | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/departments/name/{name} | PUT | Update department by name | String: Department updated successfully. | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/departments/{ deptno } | DELETE | Delete department by deptno | String: Department Deleted Successfully |  |
| /api/v1/departments/name/{name} | DELETE | Delete department by name | String: Department with given name deleted Successfully |  |

**------------------------------------------------------Participants-----------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | titles |  |  |
| /api/v1/titles/ | POST | Add new titles object in DB | String: New title added Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/titles/ | GET | Fetch all titles objects | Collection of titles |  |
| /api/v1/titles/empno /{empno}/fromdate/{fromdate}/title/{title} | GET | Search titles by from date, empno, title | Title Object |  |
| /api/v1/titles/fromdate/{fromdate} | GET | Fetch all titles objects by from date | Collection of titles |  |
| /api/v1/titles/title/{title} | GET | Fetch all titles objects by title | Collection of titles |  |
| /api/v1/titles/title/{title}/fromdate/{fromdate} | GET | Fetch all titles objects by title and fromdate | Collection of titles |  |
| /api/v1/titles/empno /{empno}/fromdate/{fromdate} | GET | Fetch all titles objects by empno and fromdate | Collection of titles |  |
| /api/v1/titles/empno/{empno}/title/{title} | GET | Fetch all titles objects by empno and title | Collection of titles |  |
| /api/v1/titles/fromdate/{fromdate} | PUT | Update titles details for the given fromdate | Collection of titles | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/titles/title/{title} | PUT | Update titles details for the given title | Collection of titles | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/titles/{empno} | PUT | Update titles details for the given employee no | Collection of titles | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/titles/empno /{empno}/fromdate/{fromdate}/title/{title} | PUT | Update title by empno, fromdate, title | String: title updated Successfully |  |
| /api/v1/titles/empno/{empno}/fromdate/{fromdate}/title/{title} | DELETE | Delete title by empno, from date and title name. | String: title deleted Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/titles/{empno} | DELETE | Delete titles by empno | Collection of titles |  |
| /api/v1/titles/fromdate/{fromdate} | DELETE | Delete titles by fromdate | Collection of titles |  |
| /api/v1/titles/title/{title} | DELETE | Delete titles by title | Collection of titles |  |

**------------------------------------------------------Participants-----------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | salaries |  |  |
| /api/v1/salaries/ | POST | Add new salary object in DB | String: New Salary details added Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/salaries / | GET | Fetch all salary objects | Collection of salaries |  |
| /api/v1/salaries /empno/{empno}/fromdate/{fromdate}/ | GET | Search salaries by from date, empno | Salary Object |  |
| /api/v1/salaries/fromdate/{fromdate} | GET | Fetch all salary objects by from date | Collection of salary |  |
| /api/v1/salaries/empno/{empno } | GET | Fetch all salary objects by empno | Collection of salary |  |
| /api/v1/salaries/salary /{minsalary}//{maxsalary} | GET | Fetch all salary objects by salary | Collection of salary |  |
| /api/v1/salaries/fromdate/{fromdate} | PUT | Update salary details for the given fromdate | Collection of salary | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/salaries/empno/{empno} | PUT | Update salary details for the given empid | Collection of salary | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/salaries /empno/{empno}/fromdate/{fromdate}/ | PUT | Update salary by empno, fromdate, | String: salary updated Successfully |  |
| /api/v1/salaries/empno /{empno}/fromdate/{fromdate} | DELETE | Delete salary by empno and from date | String: salary deleted Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/salaries/{empno} | DELETE | Delete salary by empno | Collection of salary |  |
| /api/v1/salaries/fromdate/{fromdate} | DELETE | Delete salary by fromdate | Collection of salary |  |

**------------------------------------------------------Participants-----------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | Dept\_emp |  |  |
| /api/v1/deptemp/ | POST | Add new deptemp object in DB | String: New employee assigned to department Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptemp / | GET | Fetch all deptemp objects | Collection of deptemp |  |
| /api/v1/deptemp/empno/{empno}/deptno/{deptno} | GET | Search deptemp by empno and deptno | deptemp Object |  |
| /api/v1/deptemp/deptno/{deptno}/fromdate/{fromdate} | GET | Fetch all deptemp objects by , deptno and from date | Collection of deptemp |  |
| /api/v1/deptemp/empno/{empno}/fromdate/{fromdate} | GET | Fetch all deptemp objects by , id and from date | Collection of deptemp |  |
| /api/v1/deptemp/empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | GET | Fetch all deptemp objects by , id ,deptno,and from date | Collection of deptemp |  |
| /api/v1/deptemp/{empno}/{deptno} | PUT | Update deptemp details for the given empno and deptno | String: employee assigned to department updated Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/ deptemp /empno/{empno}/fromdate/{fromdate} | PUT | Update deptemp details for the given empno and fromdate | String: employee with no and fromdate updated Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptemp /deptno/{deptno}/fromdate/{fromdate} | PUT | Update deptemp details for the given deptno and fromdate | Collection of deptemp |  |
| /api/v1/deptemp /empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | PUT | Update deptemp by empno, deptno, fromdate | String: deptemp updated Successfully |  |
| /api/v1/deptemp/empno/{empno}/deptno/{deptno} | DELETE | Delete deptemp by empno and deptno | String: deptemp deleted Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptemp /empno/{empno}/fromdate/{fromdate} | DELETE | Delete deptemp by empno and fromdate | Deptemp obj |  |
| /api/v1/deptemp /deptno/{deptno}/fromdate/{fromdate} | DELETE | Delete deptemp by deptno and fromdate | Collection of deptemp |  |
| /api/v1/deptemp /empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | DELETE | Delete deptemp by empno, deptno and fromdate | Deptemp obj |  |

**------------------------------------------------------Participants-----------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | Dept\_manager |  |  |
| /api/v1/deptmanager/ | POST | Add new deptmanager object in DB | String: New employee assigned as manager Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptmanager / | GET | Fetch all deptmanager objects | Collection of deptmanager |  |
| /api/v1/deptmanager /empno/{empno}/deptno/{deptno} | GET | Search deptmanager by empno and deptno | deptmanager Object |  |
| /api/v1/deptmanager /deptno/{deptno}/fromdate/{fromdate} | GET | Fetch all deptmanager objects by , deptno and from date | Collection of deptmanager |  |
| /api/v1/deptmanager /empno/{empno}/fromdate/{fromdate} | GET | Fetch all deptmanager objects by , id and from date | Collection of deptmanager |  |
| /api/v1/deptmanager /empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | GET | Fetch all deptmanager objects by , id ,deptno,and from date | Collection of deptmanager |  |
| /api/v1/deptmanager /{empno}/{deptno} | PUT | Update deptmanager details for the given empno and deptno | String: employee assigned to department updated Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptmanager /empno/{empno}/fromdate/{fromdate} | PUT | Update deptmanager details for the given empno and fromdate | String: employee with no and fromdate updated Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptmanager /deptno/{deptno}/fromdate/{fromdate} | PUT | Update deptmanager details for the given deptno and fromdate | Collection of deptmanager |  |
| /api/v1/deptmanager /empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | PUT | Update deptmanager by empno, deptno, fromdate | String: deptmanager updated Successfully |  |
| /api/v1/deptmanager /empno/{empno}/deptno/{deptno} | DELETE | Delete deptmanager by empno and deptno | String: deptmanager deleted Successfully | {"timeStamp":"2023-06-08", "message": "Validation failed …”} |
| /api/v1/deptmanager /empno/{empno}/fromdate/{fromdate} | DELETE | Delete deptmanager by empno and fromdate | deptmanager obj |  |
| /api/v1/deptmanager /deptno/{deptno}/fromdate/{fromdate} | DELETE | Delete deptmanager by deptno and fromdate | Collection of deptmanager |  |
| /api/v1/deptmanager /empno/{empno}/deptno/{deptno}/fromdate/{fromdate} | DELETE | Delete deptmanager by empno, deptno and fromdate | deptmanager obj |  |

**------------------------------------------------------Participants------------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | Adminhrmsconsumer |  |  |
| /api/v1/adminhrmsconsumer/user/signup | POST | Signup new user with admin role in the DB | String: New Admin User registered Successfully |  |
| /api/v1/adminhrmsconsumer/user/login | POST | Login existing user with admin role in the DB | String: Admin User logged in Successfully |  |
| /api/v1/adminhrmsconsumer/employees/{noofyear} | GET | Get list of employees joined in last 10 years.[Get empno , name and hiredate ] | Collection of Employee DTO |  |
| /api/v1/adminhrmsconsumer/employees/count/{noofyear } | GET | Find count of employees joined in last 10 years | integer |  |
| /api/v1/adminhrmsconsumer/employees/{year} | GET | Find list of employees joined after 2005. | Collection of Employee DTO |  |
| /api/v1/adminpayrollconsumer/employees/count/{year} | GET | Find count of employees joined after 2005 | integer |  |
| /api/v1/adminhrmsconsumer/assigndept | POST | Assign department to employee from fromdate to end date. | String: Employee is assigned to the department successfully. |  |
| /api/v1/adminpayrollconsumer/assignmgr | POST | Assign manager to specific department from fromdate to end date. | String: Employee is assigned as manager for department successfully. |  |
| /api/v1/adminhrmslconsumer/assigntitle | POST | Assign employee with title [designation] from start to end date. [Assuming that by default employee will be eligible for promotion after 5 years] | String: Employee is assigned with title from fromdate to enddate |  |
|  | | | | |
| **Endpoints** | **HTTP Verb/Method** | **Description** | **Success Response** | **Error Response** |
| **Adminhrmsconsumer** | | | | |
| /api/v1/adminhrmsconsumer/department | POST | Add new department | String: Department is added successfully |  |
| /api/v1/adminhrmsconsumer/employee | POST | Add new employee | String: Employee is added successfully |  |
| /api/v1/ adminhrmsconsumer/employee/{empno} | DELETE | Delete employee once employee leave the organization by empno | String: Employee is deleted successfully |  |
| /api/v1/ adminhrmsconsumer /salaries/{empno} | PUT | Update the salary for employee when employee is appraised each year depending on rating.  [1: 10000Rs - 2:7000Rs - 3:5000Rs 4:0Rs – 5:0Rs]  If employee is promoted as well, then update salary by 100000. | String: Employee year wise salary details updated successfully. |  |
| /api/v1/ adminhrmsconsumer /employee/experience/{noofyear} | GET | Find List of employees with 5year/10year/15year/20Year/25Year/30Year experience for getting awarded. | Collections of Employees |  |
| /api/v1/adminhrmsconsumer/employee/gender/{gender}/count | GET | Find count of women employees for sending mail for giving 1000 Rs gift voucher for women’s day celebration | Integer |  |
| /api/v1/adminhrmsconsumer/employee/gender/{gender} | GET | Find list of women employees for sending mail for women’s day invitation so that 1000Rs gift voucher can be given to all . | Collection Of Employees |  |

**------------------------------------------------------Participants----------------------------------------------------------**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Endpoints | HTTP Verb/Method | Description | Success Response | Error Response |
|  |  | employeehrmsconsumer |  |  |
| /api/v1/employeehrmsconsumer/user/signup | POST | Signup new user with employee role in the DB | String: New Employee User registered Successfully |  |
| /api/v1/ employeehrmsconsumer/user/login | POST | Login existing user with employee role in the DB | String: Employee User logged in Successfully |  |
| /api/v1/employeehrmsconsumer /manager | GET | Get details of employees [id, name, dob, and department name] who are manager. | Collection of Manager DTO |  |
| /api/v1/ employeehrmsconsumer /employees/department/{deptno}/fromdate/{fromdate} | GET | Get list of employees working in specific department in specific year. | Collection of EmployeeDTO  [With empId,name,fromdate, todate,deptname. |  |
| /api/v1/ employeehrmsconsumer /department/ | GET | Check department wise average/max/min salary | Collection of DepartmentReport  Dto[Deptno,dept,name,Average,Max/Min] |  |
| //api/v1/employeehrmsconsumer /designation/details | GET | Check designation/title wise average /max/min salary | Collection of DesignationReport  Dto[designation,name,Average,Max/Min] |  |
| /api/v1/employeehrmsconsumer /midageemp | GET | Find mid age employees. [Employees in 50’s] | Collection of Employee |  |
| /api/v1/ employeehrmsconsumer/designations | GET | Check what all designations are provided in Capgemini. | Collection of strings |  |
| /api/v1/employeehrmsconsumer /manager/{fromdate} | GET | Print empid, name, hire date of those who have become manager after specific from date. | Collection |  |
| /api/v1/ employeehrmsconsumer /employees/titles/{title} | GET | Check details of those employee who are working as specific title [ ex. “Technique Leader”] | Collections.  [Get empno,name,fromdate,todate,designation |  |
| /api/v1/employeehrmsconsumer /employee/{empno} | PUT | Update last name for the given empno employee. |  |  |

**------------------------ 2 Participants will work on security-------------------------------**