**Task 1 Assessment.**

* For task 1 the first initial approach was to count the number of reports each employee has under them.

John Lennon

/ \

Paul McCartney Ringo Starr

/ \

Pete Best George Harrison

In above example the distinct employee reports is 4.

But this leads to an issue where one employee could be reporting to multiple employees and hence might be counted twice.

John Lennon

/ \

Paul McCartney Ringo Starr

/ / \

Pete Best Pete Best George Harrison

Here *Pete Best* reports to both *Paul & Ringo* and if we count each report *Pete* will be counted twice and distinct employee will come out to be 5.

To remedy this, I have taken HashSet of employeeID of each reporting employee and hence eliminating the multiple count of the employees.

* The second issue that could arise is if we have a cycle, example if data has *John Lennon* as direct report of *George Harrison* which would lead to cycle and we will be locked in infinite loop. Here I have taken assumption that no cycles exist in the database.

**Task 2 Assessment.**

The task 2 requires constructing Compensation create and read REST endpoints.

I have taken the effectiveDate in Compensation type as String for this challenge, but a better way will be taking a Calendar object and put appropriate compensation effective date.

Have created CompensationService and its Implementation for reading and creating employee Compensations.

**Task 1:**

Post Employee:

Graphical user interface, text, application, email

Description automatically generated

Get Created employee

Graphical user interface, text, application, email

Description automatically generated

Update department

Graphical user interface, text, application, email

Description automatically generated

NumberOfReports

John Lennon : 4

Graphical user interface, text, application, email

Description automatically generated

Ringo Starr : 2

Graphical user interface, text, application, email

Description automatically generated

**Task 2:**

Create Compensation

Graphical user interface, text, application, email

Description automatically generated

Read Compensation

Graphical user interface, text, application, email

Description automatically generated