**Recruitment Statistics**

Design a Recruitment Statistics app which could be used at each org-level(BU) within a company, planning to hire. The system should be able tocapture statistics on efforts placed so far in the hiring process so that eachBU can easily track and manage. Each BU may be planning to hire candidates across multiple level. Levels could be (L1,L2,L3,L4,...,L9, L10)

**API:Implement atleast 2 api's**

1.API to record candidate and panel involved in an interview and result of the interview(YES\_FOR\_NEXT\_ROUND/REJECT/HIRE). You can assume that there is only one candidate and one interview-panel involved in an interview.  
2.API to track efforts of an interview-panel in interviewing candidates, for given date range. Efforts here means the number of hours spent by a panel in interviews. You can assume that each interview takes 1 hr.   
ex: From 21/05/2019 to 26/06/2019, How many hours was spent by userId123 in interviewing?  
3.API which returns a list of candidates who were hired/rejected for given level, BU and date range   
ex: List the candidates at L3 level who were hired for BU1 business unit from 21/05/2019 to 26/06/2019?

**Expectations**

1.Code quality should be production ready for merge and deployment.  
2.Guidelines have the highest weightage than finishing more api's.  
3.Code should be demo able. Create the sample data in a file, test case or main driver program itself (no external data store). Don't spend time parsing the inputs.  
4.Code should be readable, modular (no monoliths), testable, extensible with proper naming conventions.   
5.Code should handle edge cases properly and fail gracefully.

**Guidelines:**

o Define a detailed object model for entities required by system  
o Make proper use of Inheritance, Abstraction, interfaces, exception handling  
o Have proper commenting in code and should follow best coding practises  
o Use design patterns like Builder, Factory, Visitor etc wherever applicable  
o Justify his/her coding choices i.e. why did he/she choose to define a separate function for a feature or why did he/she not define constructor for initialising class  
o Define Enums, Singleton classes where applicable  
o Separation of concern is addressed  
o Implement unit test cases for key sections of his/her code  
o Use Java 8/7 features like functional interfaces, Auto Closeable resources etc.