JobComparisonAnalysis

-/rankedJobs: List<JobDetails>
-/settings:ComparisonSettings

+ JobComparisonAnalysis(jobsToCompare:
List<JobDetails>, settings:ComparisonSettings)
+calculateJobRankings()

OfferComparisonApp

-currentJob: JobDetails
-jobOffers: List<JobDetails>
-settings: ComparisionSettings

+compareJobOffers(jobsToCompare: List<JobDetails>):
JobComparisonAnalysis
+createNewJob(isCurrent: boolean): JobDetails
+recalculateJobScores()
+editJobDetails(jobId: UUID)
+saveJob(job: JobDetails)
+cancel(): String

Generates

Edit/Enter

1

0..*

0..*

1

JobDetails

-/id: UUID
-title: String
-company: String
-location: Location
-annualSalary: float
-annualBonus: float
-relocStipend: float
-retirementBenefitsPct: Integer
-restrictedStockUnit: Integer
-isCurrentJob: boolean

+calculateAYS(): Double
+calculateAYB(): Double
+calculateScore(): Double

calculateScore = AYS + AYB + RS +
(RPB * AYS / 100) + (RSUA / 4)

score is derived from JobDetails attributes:
AYS = yearly salary adjusted for cost of living
AYB = yearly bonus adjusted for cost of living
RBP = retirement benefits percentage
RS = relocation stipend
RSUA = restricted stock unit award

0..*

1

Adjusts

Leverages

0..*

1

1

1

ComparisonSettings

-YearlySalaryWeight : Int = 7
-YearlyBonusWeight: Int = 7
-RetirementBenefitsWeight: Int = 7
-RelocationStipedWeight: Int = 7
-RestrictedStockAwardWeight: Int = 7

Location

-city: String
-state: State
-costOfLiving: float

+Location(city: String, state: State,
costOfLiving: float)

<<enumeration>>
State

value: String

0..*

1