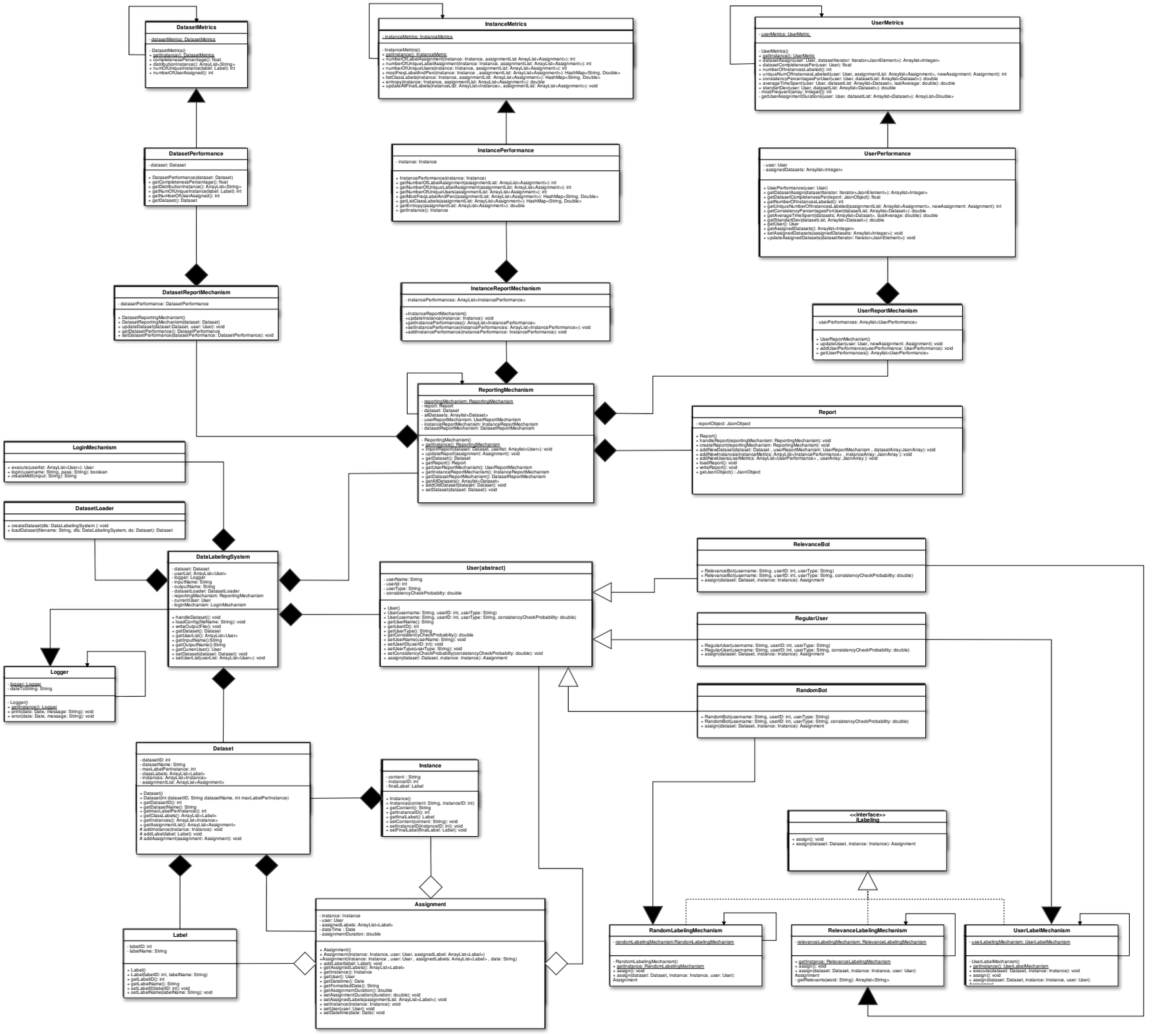
DatasetMetrics
- datasetMetrics: DatasetMetrics
- DatasetMetrics()
+ getInstance(): DatasetMetrics
+ completenessPercentage(): float
+ distributionInstance(): ArrayList<String>
+ numOfUniqueInstance(label: Label): Int
+ numberOfUserAssigned(): int
InstanceMetrics
- instanceMetrics: InstanceMetrics
- InstanceMetrics()
+ getInstance(): InstanceMetric
+ numberOfLabelAssignment(instance: Instance, assignmentList: ArrayList<Assignment>): int
+ numberOfUniqueLabelAssignment(instance: Instance, assignmentList: ArrayList<Assignment>): int
+ numberOfUniqueUsers(instance: Instance, assignmentList: ArrayList<Assignment>): int
+ mostFreqLabelAndPerc(instance: Instance , assignmentList: ArrayList<Assignment>): HashMap<String, Double>
+ listClassLabels(instance: Instance, assignmentList: ArrayList<Assignment>): HashMap<String, Double>
+ entropy(instance: Instance, assignmentList: ArrayList<Assignment>): double
+ updateAllFinalLabels(instanceList: ArrayList<Instance>, asdignmentList: ArrayList<Assignment>): void
UserMetrics
- userMetrics: UserMetric
- UserMetrics()
+ getInstance(): UserMetric
+ datasetAssign(user: User, datasetIterator: Iterator<JsonElement>): Arraylist<Integer>
+ datasetCompletenessPer(user: User): float
+ numberOfInstancesLabeled(): int
+ uniqueNumOfInstancesLabeled(user: User, assignmentList: Arraylist<Assignment>, newAssignment: Assignment): int
+ consistencyPercentagesForUser(user: User, datasetList: Arraylist<Dataset>): double
+ averageTimeSpent(user: User, datasetList: Arraylist<Dataset>, lastAverage: double): double
+ standardDev(user: User, datasetList: Arraylist<Dataset>): double
- modFrequent(array: Integer[]): int
- getUserAssignmentDurations(user: User, dataset List: Arraylist<Dataset>): ArrayList<Double>
DatasetPerformance
- dataset: Dataset
+ DatasetPerformance(dataset: Dataset)
+ getCompletenessPercentage(): float
+ getDistributionInstance(): ArrayList<String>
+ getNumOfUniqueInstance(label: Label): int
+ getNumberOfUserAssigned(): int
+ getDataset(): Dataset
InstancePerformance
- instance: Instance
+ InstancePerformance(instance: Instance)
+ getNumberOfLabelAssignment(assignmentList: ArrayList<Assignment>): int
+ getNumberOfUniqueLabelAssignment(assignmentList: ArrayList<Assignment>): int
+ getNumberOfUniqueUsers(assignmentList: ArrayList<Assignment>): int
+ getMostFreqLabelAndPerc(assignmentList: ArrayList<Assignment>): HashMap<String, Double>
+ getListClassLabels(assignmentList: ArrayList<Assignment>): HashMap<String, Double>
+ getEntropy(assignmentList: ArrayList<Assignment>): double
+ getInstance(): Instance
UserPerformance
- user: User
- assignedDatasets: Arraylist<Integer>
+ UserPerformance(user: User)
+ getDatasetAssign(datasetIterator: Iterator<JsonElement>): Arraylist<Integer>
+ getDatasetCompletenessPer(report: JsonObject): float
+ getNumberOfInstancesLabeled(): int
+ getUniqueNumberOfInstancesLabeled(assignmentLst: Arraylist<Assignment>, newAssignment: Assignment): int
+ getConsistencyPercentagesForUser(datasetLst: Arraylist<Dataset>): double
+ getAverageTimeSpent(datasets: Arraylist<Dataset>, lastAverage: double): double
+ getStandardDev(datasetLst: Arraylist<Dataset>): double
+ getUser(): User
+ getAssignedDatasets(): Arraylist<Integer>
+ setAssignedDatasets(assignedDatasets: Arraylist<Integer>): void
+ updateAssignedDatasets(datasetIterator: Iterator<JsonElement>): void
DatasetReportMechanism
- datasetPerformance: DatasetPerformance
+ DatasetReportingMechanism()
+ DatasetReportingMechanism(dataset: Dataset)
+ updateDataset(dataset Dataset, user: User): void
+ getDatasetPerformance(): DatasetPerformance
+ setDatasetPerformance(DatasetPerformance: DatasetPerformance): void
InstanceReportMechanism
- instancePerformances: ArrayList<InstancePerformance>
+InstanceReportMechanism()
+updateInstance(instance: Instance): void
+getInstancePerformances(): ArrayList<InstancePerformance>
+setInstancePerformance(instancePerformances: ArrayList<InstancePerformance>): void
+addInstancePerformance(instancePerformance: InstancePerformance): void
UserReportMechanism
- userPerformances: Arraylist<UserPerformance>
+ UserReportMechanism()
+ updateUser(user: User, newAssignment: Assignment): void
+ addUserPerformance(userPerformance: UserPerformance): void
+ getUserPerformances(): Arraylist<UserPerformance>
ReportingMechanism
- reportingMechanism: ReportingMechanism
- report: Report
- dataset: Dataset
- allDatasets: Arraylist<Dataset>
- userReportMechanism: UserReportMechanism
- instanceReportMechanism: InstanceReportMechanism
- datasetReportMechanism: DatasetReportMechanism
- ReportingMechanism()
+ getInstance(): ReportingMechanism
+ importReport(dataset: Dataset, usersIst: Arraylist<User>): void
+ updateReport(assignment: Assignment): void
+ getDataset(): Dataset
+ getReport(): Report
+ getUserReportMechanism(): UserReportMechanism
+ getInstanceReportMechanism(): InstanceReportMechanism
+ getDatasetReportMechanism(): DatasetReportMechanism
+ getAllDatasets(): Arraylist<Dataset>
+ addOldDataset(dataset: Dataset): void
+ setDataset(dataset: Dataset): void
Report
- reportObject: JsonObject
+ Report()
+ handleReport(reportingMechanism: ReportingMechanism): void
+ createReport(reportingMechanism: ReportingMechanism): void
+ addNewDataset(dataset: Dataset , userReportMechanism: UserReportMechanism , datasetArray:JsonArray): void
+ addNewInstances(instanceMetrics: ArrayList<InstancePerformance> , instanceArray: JsonArray): void
+ addNewUsers(userMetrics: ArrayList<UserPerformance> , userArray: JsonArray): void
+ loadReport(): void
+ writeReport(): void
+ getJsonObject() : JsonObject
LoginMechanism
+ execute(usersIst: ArrayList<User>): User
+ login(username: String, pass: String): boolean
+ createMd5(input: String): String
DatasetLoader
+ createDataset(dls: DataLabelingSystem): void
+ loadDataset(filename: String, dls: DataLabelingSystem, ds: Dataset): Dataset
DataLabelingSystem
- dataset: Dataset
- userList: ArrayList<User>
- logger: Logger
- input/Name: String
- outputName: String
- datasetLoader: DatasetLoader
- reportingMechanism: ReportingMechanism
- currentUser: User
- loginMechanism: LoginMechanism
+ handleDataset(): void
+ loadConfig(fileName: String): void
+ writeOutputFile(): void
+ getDataset(): Dataset
+ getUserList(): ArrayList<User>
+ getInputName():String
+ getOutputName()! String
+ getCurrentUser(): User
+ setDataset(dataset: Dataset): void
+ setUserList(userList: ArrayList<User>): void
User(abstract)
- userName: String
- userId: int
- userType: String
- consistencyCheckProbability: double
+ User()
+ User(username: String, userID: int, userType: String)
+ User(username: String, userID: int, userType: String, consistencyCheckProbability: double)
+ getUserName(): String
+ getUserID: int
+ getUserType(): String
+ getConsistencyCheckProbability(): double
+ setUserName(userName: String): void
+ setUserID(userID: int): void
+ setUserTypeu(userType: String): void
+ setConsistencyCheckProbability(consistencyCheckProbability: double): void
+ assign(dataset: Dataset, instance: Instance): Assignment
RelevanceBot
+ RelevanceBot(username: String, userID: int, userType: String)
+ RelevanceBot(username: String, userID: int, userType: String, consistencyCheckProbability: double)
+ assign(dataset: Dataset, instance: Instance): Assignment
RegularUser
+ RegularUser(username: String, userID: int, userType: String)
+ RegularUser(username: String, userID: int, userType: String, consistencyCheckProbability: double)
+ assign(dataset: Dataset, instance: Instance): Assignment
RandomBot
+ RandomBot(username: String, userID: int, userType: String)
+ RandomBot(username: String, userID: int, userType: String, consistencyCheckProbability: double)
+ assign(dataset: Dataset, instance: Instance): Assignment
Logger
- logger: Logger
- dateToString: String
- Logger()
+ getInstance(): Logger
+ info(date: Date, message: String): void
+ error(date: Date, message: String): void
Dataset
- datasetID: int
- datasetName: String
- maxLabelPerInstance: int
- classLabels: ArrayList<Label>
- instances: ArrayList<Instance>
- assignmentList: ArrayList<Assignment>
+ Dataset()
+ Dataset(int datasetID, String datasetName, int maxLabelPerInstance)
+ getDatasetID(): int
+ getDatasetName(): String
+ getmaxLabelPerInstance(): int
+ getClassLabels(): ArrayList<Label>
+ getInstances(): ArrayList<Instance>
+ getAssignmentList(): ArrayList<Assignment>
# addInstance(instance: Instance): void
# addLabel(label: Label): void
# addAssignment(assignment: Assignment): void
Instance
- content : String
- instanceID: int
- finalLabel: Label
+ Instance()
+ Instance(content: String, instanceID: int)
+ getContent(): String
+ getInstanceID(): int
+ getfinalLabel(): Label
+ setContent(content: String): void
+ setInstanceID(instanceID: int): void
+ setFinalLabel(finalLabel: Label): void
<<interface>>
ILabeling
+ assign(): void
+ assign(dataset: Dataset, instance: Instance): Assignment
Label
- labelID: int
- labelName: String
+ Label()
+ Label(labelID: int, labelName: String)
+ getLabelID(): int
+ getLabelName(): String
+ setLabelID(labelID: int): void
+ setLabelName(labelName: String): void
Assignment
- instance: Instance
- user: User
- assignedLabels: ArrayList<Label>
- dateTime : Date
- assignmentDuration: double
+ Assignment()
+ Assignment(instance: Instance, user: User, assignedLabel: ArrayList<Label>)
+Assignment(instance: Instance , user: User , assignedLabels: ArrayList<Label> , date: String)
+ addLabel(label: Label): void
+ getAssignedLabels(): ArrayList<Label>
+ getInstance(): Instance
+ getUser(): User
+ getDatetime(): Date
+ getFormatteDate(): String
+ getAssignmentDuration(): double
+ setAssignmentDuration(duration: double): void
+ setAssignedLabels(assignmentLst: ArrayList<Label>): void
+ setInstance(instance: Instance): void
+ setUser(user: User): void
+ setDatetime(date: Date): void
RandomLabelingMechanism
- randomLabelingMechanism:RandomLabelingMechanism
- RandomLabelingMechanism()
+ getInstance: RandomLabelingMechanism
+ assign(): void
+ assign(dataset: Dataset, instance: Instance, user: User): Assignment
RelevanceLabelingMechanism
- relevanceLabelingMechanism: RelevanceLabelingMechanism
+ getInstance: RelevanceLabelingMechanism
+ assign(): void
+ assign(dataset: Dataset, instance: Instance, user: User): Assignment
- getRelevants(word: String): Arraylist<String>
UserLabelMechanism
- userLabelingMechanism: UserLabelMechanism
- UserLabelMechanism()
+ getInstance(): UserLabelMechanism
+ execute(dataset: Dataset, instance: Instance): void
+ assign(): void
+ assign(dataset: Dataset, instance: Instance, user: User): Assignment