	UserPerformanceMetrics	
	user: UserInfo	
	numberOfDataset: int	
	datasetsCompletenessPercentage: float[]	
	totalNumberOfInstanceLabeled: int	
	totalNumberOfUniqueInstance: int	
	countOfRecurrentInstances: int	
	consistencyPercentage: float	
	times: ArrayList <float></float>	
	avgTime: float	
	stdTime: float	
+	UserPerformanceMetrics()	
+	UserPerformanceMetrics(UserInfo, int)	
	updateDatasetsCompletenessPercentage(ArrayList <instance>, String): void</instance>	
	incrementTotalNumberOfInstanceLabeled(): void	
	incrementTotalNumberOfUniqueInstance(): void	
+	updateConsistencyPercentage(): void	
	addTimes(float): void	
	updateAverageTime(): void	
	updateStdTime(): void	
+	getUser(): UserInfo	
	getCountOfRecurrentInstances(): int	
+	getNumberOfDataset(): int	
+	getDatasetsCompletenessPercentage(): float[]	
	getTotalNumberOfInstanceLabeled(): int	
	getTotalNumberOfUniqueInstance(): int	
	getConsistencyPercentage(): float	
	getTimes(): ArrayList <float></float>	
+	getAvgTime(): float	
+	getStdTime(): float	
+	setUser(UserInfo): void	
+	setNumberOfDataset(int): void	
+	setDatasetsCompletenessPercentage(float[]): void	
+	setTotalNumberOfInstanceLabeled(int): void	
+	setTotalNumberOfUniqueInstance(int): void	
+	setConsistencyPercentage(float): float	
+	setTimes(ArrayList <float>): void</float>	
+	setAvgTime(float): void	
+	setStdTime(float): void	
_	DatasetPerformanceMetrics	
	percentage: float	
	distributionOfFinalInstanceLabels: ArrayList <classlabelandpercentage></classlabelandpercentage>	
	labelAndNumberOfUniqueInstances: ArrayList <labelandnumberofuniqueinstances< th=""><th>nce></th></labelandnumberofuniqueinstances<>	nce>
	numberOfUserAssigned: int	
	assignedUsersAndCompletenessPercentage: ArrayList <userandpercentage></userandpercentage>	
	assignedUsersAndConsistencyPercentage: ArrayList <userandpercentage></userandpercentage>	
+	DatasetPerformanceMetrics(int)	
	updatePercentage(ArrayList <instance>): void</instance>	
	updateDistributionOfFinalInstanceLabels(ArrayList <instance>, ArrayList<classi< th=""><th>_abel>):</th></classi<></instance>	_abel>):
	updateUniqueInstanceForEachLabel(ArrayList <instance>, ArrayList<classlabe< th=""><th></th></classlabe<></instance>	
	updateAssignedUsersAndCompletenessPercentage(UserInfo, String): void	,
	updateAssignedUsersAndConsistencyPercentage(UserInfo): void	
	getPercentage(): float	
	natDistributionOfFinalInstance shele(): Arrayl ist-I learAndParcentage>	

+ undate

+ getCon getTim + getAvg

+ setUse

+ setTota

+ setCon

+ setTim

+ setStd



+ getNumberOfUserAssigned(): int + getAssignedUsersAndCompletenessPercentage(): ArrayList<UserAndPercent - getAssignedUsersAndConsistencyPercentage(): ArrayList<UserAndPercentage>

ClassLabelAndPercentage	
- classLabel: ClassLabel	LabelAndNumberOfUniqueInstance
- percentage: float	- classLabel: ClassLabel
+ ClassLabelAndPercentage(ClassLabel, float)	- count: int
+ ClassLabelAndPercentage()	+ LabelAndNumberOfUniqueInstance(ClassLabel,
+ getClassLabel(): ClassLabe	+ getClassLabel(): ClassLabel
+ getPercentage(): float	+ getCount(): int
+ setClassLabel(ClassLabel): void	
+ setPercentage(float): void	

	UserPerformanceMetricsCreator
- 0	datasetsInfo: ArrayList <datasetinfo></datasetinfo>
- 0	currentUsers: ArrayList <userinfo></userinfo>
+	UserPerformanceMetricsCreator(ArrayList <datasetinfo>, ArrayList<userinfo>)</userinfo></datasetinfo>
+	invoke(): void

SetupDataset	
- config: Config	
- currentDatasetID:int	
- datasetsInfo: ArrayList <datasetinfo></datasetinfo>	
- currentDatasetInfo: DatasetInfo	
+ SetupDataset(Config)	
+ getCurrentDatacetID0: int	

etInfo(): DatasetInfo

+ getDatasetsInfo(): ArrayList<Data

	+ invoke(): SetupData
InstancePerformanceMetricsCreator	
- currentDatasetID: int	
- allUsers: ArrayList <userinfo></userinfo>	
classLabels: ArrayList <classlabel></classlabel>	
-instances: ArrayList <instance></instance>	
+ InstancePerformanceMetricsCreator(int, ArrayList <userinfo>, ArrayList<classlabel>, ArrayL</classlabel></userinfo>	.ist <instance>)</instance>
+ invoke(): void	

+ invoke(): void		
FileReader	ı	_
- reader: FileReader		- cui
		- da

- reader: FileReader	
+ FileReader()	
+ getReader(): FileReader	
+ read(File, Object): Object	

DatasetAndPercentage
- datasetName: String
- percentage: float
+ DatasetAndPercentage(String, float)
→ DatasetAndPercentage()

tName(): String + getPercentage(): float

DatasetCreator
- currentDatasetID: int
- datasetsInfo: ArrayList <datasetinfo></datasetinfo>
- currentDatasetInfo: DatasetInfo
+ DatasetCreator(int, ArrayList <datasetinfo>, DatasetInfo)</datasetinfo>
+ invoke(): Dataset

AssignedUsers
config: Config
currentDatasetInfo: DatasetInfo
logger: Logger
allUsers: ArrayList <userinfo></userinfo>
currentUser: ArrayList <userinfo></userinfo>
- AssignedUsers(Config, DatasetInfo, Logger)

userInfos: ArrayList<UserInfo>

+ readConfig(): Config

datasetID: int

datasetFilePath: String

getDatasetID(): int

getDatasetFilePath(): String

setDatasetName(String): void

setDatasetFilePath(String): void

setDatasetName(String): void imumNumberOfLabelPerIn

setClassLabels(ArrayList<ClassLabel>): void

setInstances(ArrayList<Instance>): void

setDatasetID(int): void

getCurrentUserID(): int

setCurrentUserID(int): void

getUserInfos(): ArrayList<UserInfo>

+ setDatasetInfos(ArrayList<DatasetInfo>): void

+ setAssignUserID(ArrayList<Integer>): void getLabelingMechanismType(): String datasetID: int setLabelingMechanismType(String): void maximumNumberOfLabelsPerInstance: int classLabels: ArrayList<ClassLabel> instances: ArrayList<Instance> datasetPerformanceMetrics: Data getDatasetID(): int erInfo, Instance, ArrayList<ClassLabel>): void getDatasetName(): String getMaximumNumberOfLabelPerInstance(): int getClassLabels(): ArrayList<ClassLabel> getDatasetPerformanceMetrics(): DatasetPerformanceMetrics

UserInfo

userID: int

username: String

+ getUserID(): int

+ getUserType(): String

+ setUserID(int): void

+ setUsername(String): void

consistencyCheckProbability: double

+ getConsistencyCheckProbability(): double

+ getLabeledInstances(): ArrayList<LabeledInstance:

+ setConsistencyCheckProbability(double): void

Label

count: int

+ Label(ClassLabel)

+ getCount(): int

+ incrementCount(): void

ClassLabel

labelText: String

+ getLabelID(): int

setLabelID(int): void

setLabelText(String): void

+ getUserPerformanceMetrics(): UserPerformanceMetrics

+ setUserPerformanceMetrics(UserPerformanceMetrics): void

abeledInstances: ArrayList<LabeledInstance>

userPerformanceMetrics: UserPerformanceMetrics

username: String percentage: float + UserAndPercentage(ClassLabel, int) getUsername(): String

+ getPercentage(): float

InstancePerformanceMetrics allLabelAssignments: ArrayList<Lab totalNumberOfLabelAssignments: int numberOfUniqueLabelAssignments: int numberOfUniqueUsers: int finalLabelAndPercentage: ClassLabelAndPercentage

entropy: float InstancePerformanceMetrics() addNewLabelAssignment(String, String, int, int): void

updateNumberOfUniqueUsers(int): void updateNumberOfUniqueLabelAssignments(int): void
updateFinalLabelAndPercentage(ArrayList<ClassLabel>): void updateLabelsAndPercentage(ArrayList<ClassLabel>): void

updateEntropy(int): void

getTotalNumberOfLabelAssignments(): int

getNumberOfUniqueUsers(): int

getFinalLabelAndPercentage(): ClassLa getClassLabelsAndPercentages(): ArrayList<ClassLabelAndPercentages

getEntropy(): int

setAllLabelAssignments(ArrayList<LabelAssig setTotalNumberOfLabelAssignments(int): void

setNumberOfUniqueUsers(int); void

+ setEntropy(int): void

setClassLabelsAndPercentages(ArrayList<ClassLabelAndPerc

setFinalLabelAndPercentage(ClassLabelAndPercentage)

checkUserLabels(UserInfo): LabeledInstance addLabel(ClassLabel): void removeDuplicates(ArrayList<ClassLabel>): void checkAmountOfLabels(): void getInstance(): String getUserLabels(): ArrayList<LabeledInstance> erformanceMetrics(): InstancePerformar setID(int): void setInstance(String): void setMaxNumberOfLabel(int): void + setInstancePerformanceMetrics(InstancePerformanceMetrics): void Labele labels: ArravList<Label> + LabeledInstance(int, String, UserInfo, LocalDateTime + addLabel(Label): void getWhoLabeled(): UserInfo + getLabels(): ArrayList<Label> + getDate(): LocalDateTime

Instance

id: int instance: String

maxNumberOfLabel; int

serLabels: ArrayList<LabeledInstance:

instancePerformanceMetrics: InstancePerformanceMetrics

updateUserLabels(ArrayList<UserInfo>, ArrayList<ClassLabels>): void

createLabeledInstance(UserInfo): LabeledInstance

updateInstance(LabeledInstance, Label): void addUserLabel(LabeledInstance, Label): void

allLabels: ArravList<ClassLabel>

labelText: String userID: int labelID: int + LabelAssignment(String, String, int, int)

+ setDatasetPerformanceMetrics(DatasetPerformanceMetrics): void

LabelAssignment

- LabelAssignment()

getUsername(): String

getLabelText(): String

+ setLabe(Text(String); void

+ setLabelID(int): void

+ getLabelID(): int

labelAndNumberOfUniqueInstances: ArrayList<LabelAndNumberOfUniqueInstance> assignedUsersAndCompletenessPercentage: ArrayList<UserAndPercentage: assignedUsersAndConsistencyPercentage: ArrayList<UserAndPercentage> + DatasetPerformanceMetrics(int) updatePercentage(ArrayList<Instance>): void undateDistributionOfFinalInstanceLabels(ArrayList<Inc - updateUniqueInstancesForEachLabel(ArrayList<ClassLabel>,ArrayList<Instance>): void - updateAssignedUsersAndConsistencyPercentage (UserInfo): void + getDistributionOfFinalInstanceLabels(); ArravList<ClassLabelAndPercentage> + getLabelAndNumberOfUniqueInstances(); ArrayList<LabelAndNumberOfUniqueInstance + getNumberOfUserAssigned(); int + getAssignedUsersAndCompletenessPercentage(): ArrayList<UserAndPercentage: + getAssignedUsersAndConsistencyPercentage(): ArrayList<UserAndPercentage

LabelingSimulation dataset: Dataset allUsers: ArrayList<UserInfo> instances: ArrayList<Instance> + LabelingSimulation(Dataset, ArrayList<UserInfo>, ArrayList<UserInfo>, ArrayList<Instance>, ArrayList<ClassLabel>) + getNotLabeledInstances(UserInfo): ArravList<Instance> getPreviouslyLabeledInstances(UserInfo): ArrayList<Instance: + getNumberOfTurn(); int startTurn(UserInfo, LabelingMechanism): void getRandomInstance(UserInfo, ArrayList<Instance>, ArrayList<Instance>): Instance + incrementSpecificNumberOfInstanceLabeled(UserInfo, String): void - updateUserPerformanceMetrics(UserInfo, float): void + updateInstancePerformanceMetrics(Instance): void

+ writeUserPerformanceMetrics(UserInfo, UserPerformanceMetrics); void writeInstancePerformanceMetrics(Instance, InstancePerformanceMetrics): void +writeDatasetPerformanceMetrics(DatasetPerformanceMetrics): void

getCurrentUsers(): ArrayList<UserInfo: