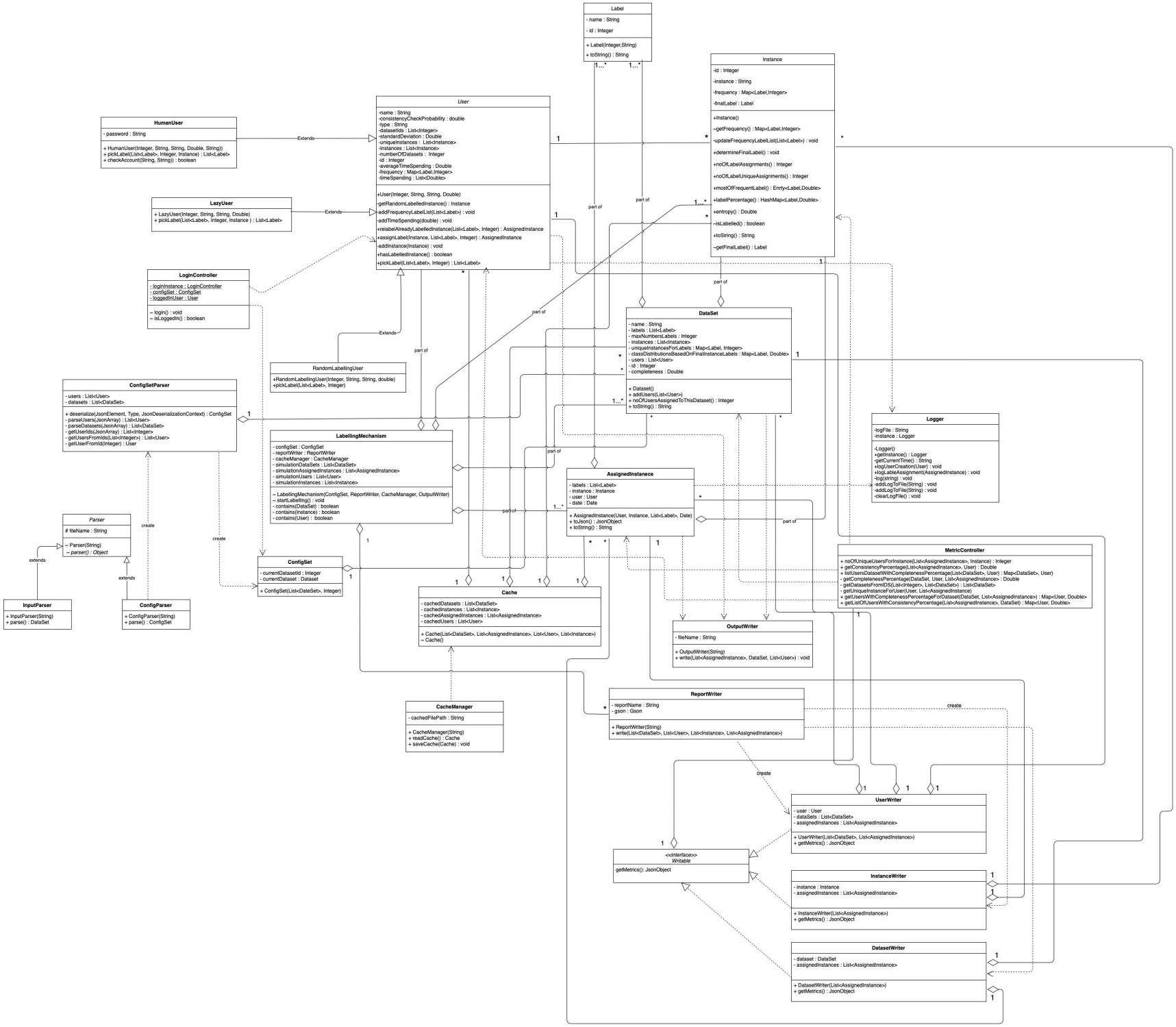
Label
- name : String
- id : Integer
+ Label(Integer,String)
+ toString() : String
Instance
-id : Integer
-instance : String
-frequency : Map<Label,Integer>
-finalLabel : Label
+Instance()
~getFrequency() : Map<Label,Integer>
-updateFrequencyLabelList(List<Label>) : void
+determineFinalLabel() : void
+noOfLabelAssignments() : Integer
+noOfLabelUniqueAssignments() : Integer
+mostOfFrequentLabel() : Entry<Label,Double>
+labelPercentage() : HashMap<Label,Double>
+entropy() : Double
~isLabelled() : boolean
+toString() : String
~getFinalLabel() : Label
User
-name : String
-consistencyCheckProbability : double
-type : String
-datasetIds : List<Integer>
-standardDeviation : Double
-uniqueInstances : List<Instance>
-instances : List<Instance>
-numberOfDatasets : Integer
-id : Integer
-averageTimeSpending : Double
-frequency : Map<Label,Integer>
-timeSpending : List<Double>
+User(Integer, String, String, Double)
-getRandomLabelledInstance() : Instance
-addFrequencyLabelList(List<Label>) : void
-addTimeSpending(double) : void
+relabelAlreadyLabelledInstance(List<Label>, Integer) : AssignedInstance
+assignLabel(Instance, List<Label>, Integer) : AssignedInstance
-addInstance(Instance) : void
+hasLabelledInstance() : boolean
+pickLabel(List<Label>, Integer) : List<Label>
HumanUser
- password : String
+ HumanUser(Integer, String, String, Double, String))
+ pickLabel(List<Label>, Integer, Instance) : List<Label>
+ checkAccount(String, String)) : boolean
Extends
LazyUser
+ LazyUser(Integer, String, String, Double)
+ pickLabel(List<Label>, Integer, Instance) : List<Label>
Extends
LoginController
- loginInstance : LoginController
- configSet : ConfigSet
- loggedInUser : User
~ login() : void
~ isLoggedIn() : boolean
RandomLabellingUser
+RandomLabellingUser(Integer, String, String, double)
+pickLabel(List<Label>, Integer)
Extends
DataSet
- name : String
- labels : List<Label>
- maxNumbersLabels : Integer
- instances : List<Instance>
- uniqueInstancesForLabels : Map<Label, Integer>
- classDistributionsBasedOnFinalInstanceLabels : Map<Label, Double>
- users : List<User>
- id : Integer
- completeness : Double
+ Dataset()
+ addUsers(List<User>)
+ noOfUsersAssignedToThisDataset() : Integer
+ toString() : String
ConfigSetParser
- users : List<User>
- datasets : List<DataSet>
+ deserialize(JsonElement, Type, JsonDeserializationContext) : ConfigSet
- parseUsers(JsonArray) : List<User>
- parseDatasets(JsonArray) : List<DataSet>
- getUserIds(JsonArray) : List<Integer>
- getUsersFromIds(List<Integer>) : List<User>
- getUserFromId(Integer) : User
LabellingMechanism
- configSet : ConfigSet
- reportWriter : ReportWriter
- cacheManager : CacheManager
- simulationDataSets : List<DataSet>
- simulationAssignedInstances : List<AssignedInstance>
- simulationUsers : List<User>
- simulationInstances : List<Instance>
~ LabellingMechanism(ConfigSet, ReportWriter, CacheManager, OutputWriter)
~ startLabelling() : void
- contains(DataSet) : boolean
- contains(Instance) : boolean
- contains(User) : boolean
AssignedInstance
- labels : List<Label>
- instance : Instance
- user : User
- date : Date
+ AssignedInstance(User, Instance, List<Label>, Date)
+ toJson() : JsonObject
+ toString() : String
Logger
-logFile : String
-instance : Logger
-Logger()
+getInstance() : Logger
-getCurrentTime() : String
+logUserCreation(User) : void
+logLableAssignment(AssignedInstance) : void
-log(string) : void
-addLogToFile(String) : void
-addLogToFile(String) : void
-clearLogFile() : void
Parser
fileName : String
~ Parser(String)
~ parser() : Object
extends
extends
create
create
InputParser
+ InputParser(String)
+ parse() : DataSet
ConfigParser
+ ConfigParser(String)
+ parse() : ConfigSet
ConfigSet
- currentDatasetId : Integer
- currentDataset : Dataset
+ ConfigSet(List<DataSet>, Integer)
Cache
- cachedDatasets : List<DataSet>
- cachedInstances : List<Instance>
- cachedAssignedInstances : List<AssignedInstance>
- cachedUsers : List<User>
+ Cache(List<DataSet>, List<AssignedInstance>, List<User>, List<Instance>)
~ Cache()
MetricController
+ noOfUniqueUsersForInstance(List<AssignedInstance>, Instance) : Integer
+ getConsistencyPercentage(List<AssignedInstance>, User) : Double
+ listUsersDatasetWithCompletenessPercentage(List<DataSet>, User) : Map<DataSet>, User)
- getCompletenessPercentage(DataSet, User, List<AssignedInstance> : Double
- getDatasetsFromIDS(List<Integer>, List<DataSet>) : List<DataSet>
- getUniqueInstanceForUser(User, List<AssignedInstance)
+ getUsersWithCompletenessPercentageForDataset(DataSet, List<AssignedInstance>) : Map<User, Double>
+ getListOfUsersWithConsistencyPercentage(List<AssignedInstance>, DataSet) : Map<User, Double>
OutputWriter
- fileName : String
+ OutputWriter(String)
+ write(List<AssignedInstance>, DataSet, List<User>) : void
CacheManager
- cachedFilePath : String
+ CacheManager(String)
+ readCache() : Cache
+ saveCache(Cache) : void
ReportWriter
- reportName : String
- gson : Gson
+ ReportWriter(String)
+ write(List<DataSet>, List<User>, List<Instance>, List<AssignedInstance>)
create
UserWriter
- user : User
- dataSets : List<DataSet>
- assignedInstances : List<AssignedInstance>
+ UserWriter(List<DataSet>, List<AssignedInstance>)
+ getMetrics() : JsonObject
<<interface>>
Writable
getMetrics(): JsonObject
InstanceWriter
- instance : Instance
- assignedInstances : List<AssignedInstance>
+ InstanceWriter(List<AssignedInstance>)
+ getMetrics() : JsonObject
DatasetWriter
- dataset : DataSet
- assignedInstances : List<AssignedInstance>
+ DatasetWriter(List<AssignedInstance>)
+ getMetrics() : JsonObject
part of