LocationUpdateDAO

- TBLNAME : String = "location"

+ getLocation(id : String) : LocationUpdate
+ retrieveAllLocations() : ArrayList<String>
+ retrieveAll() : ArrayList<LocationUpdate>
+ retrieveAllValidatedLocationsDetails() : HashMap<String,Integer>
+ retrieveUserUpdatesInWindow(datetime : String) : ArrayList<LocationUpdate>
+ retrieveUserUpdatesInWindow(macAddress : String, datetime : String) : ArrayList<LocationUpdate>
+ retrieveUserUpdatesInWindow(macAddress : String, start : String, end : String) : ArrayList<LocationUpdate>
+ retrieveNearbyUpdates(macAddress : String, startTime : String, endTime : String, locationId : int) : ArrayList<LocationUpdate>
+ insert(path : String) : void
+ deleteDuplicates(timeStamp : String, macAddress : String) : boolean
+ TryUpdate(timeStamp : String, macAddress : String, location : int) : boolean
+ retrieveLocationsInWindow(date : String) : ArrayList<LocationUpdate>
+ deleteTable() : void

AdminDAO

- TBLNAME : String = "admin"

+ getAdmin(id : String) : Admin
+ retrieveAllAdmins() : ArrayList<String>

<<interface>>
Connection

<<interface>>
PreparedStatement

SemanticPlaceDAO

- TBLNAME : String = "semanticplace"

+ getSemanticPlace(id : String) : SemanticPlace
+ retrieveAllSemanticPlaces() : ArrayList<String>
+ deleteTable() : void
+ insert(path : String) : void
+ retrieveAllValidatedLocations() : ArrayList<Integer>
+ retrieveMacsAtSemanticPlace(place : String, datetime : String) : ArrayList<String>
+ retrieveUserVisitsToSemanticPlaces(mac : String, datetime : String) : ArrayList<String[]>

UserDAO

- TBLNAME : String = "users"

+ getUserWithEmail(id : String) : User
+ getUserWithMacAddress(mac : String) : User
+ retrieveUsersInWindow(date : String) : ArrayList<User>
+ retrievePersonsInWindow(date : String) : ArrayList<User>
+ retrieveAllUsers() : ArrayList<User>
+ insert(path : String) : void
+ deleteTable() : void
+ validate(username : String, password : String) : boolean
+ retrievePassword(username : String) : String

CrowdDensityDAO

+ retrieveCrowdDensitiesInWindow(datetime : String) : ArrayList<CrowdDensity>
+ retrieveCrowdDensitiesInWindowOnFloor(datetime : String, floor : String) : ArrayList<CrowdDensity>

AutomaticGroupDetectionDAO

- TBLNAME : String = "location"

+ retrieveUserUpdatesInWindowForAGD(macAddress : String, start : String, end : String) : LinkedHashMap<Integer,ArrayList<Date>>
+ removeUsersWithout12MinWindows(userVisits : HashMap<User,LinkedHashMap<Integer,ArrayList<Date>>>, start : String, end : String) : HashMap<User,LinkedHashMap<Integer,ArrayList<Date>>>
+ updateTrackerForAGD(userVisits : ArrayList<LocationUpdate>) : LinkedHashMap<ArrayList<Date>,Integer>

<<interface>>
ResultSet

ArrayList<CrowdDensity>

ArrayList<Date>

ArrayList<User>

ArrayList<String[]>

ArrayList

LinkedHashMap<Integer,ArrayList<Date>>

LinkedHashMap<ArrayList<Date>,Integer>

HashMap<User,LinkedHashMap<Integer,ArrayList<Date>>>

HashMap<String,Integer>

LinkedHashMap

HashMap

ArrayList<Integer>

ArrayList<String>

LinkedHashMap<Integer,ArrayList<Date>>

ArrayList<LocationUpdate>

HashMap<User,LinkedHashMap<Integer,ArrayList<Date>>>

LinkedHashMap<Integer,ArrayList<Date>>

<<bind>>
<E->CrowdDensity>

<<bind>>
<E->Date>

<<bind>>
<E->User>

<<bind>>
<E->String[]>

<<bind>>
<E->Integer>

<<bind>>
<E->String>

<<bind>>
<E->LocationUpdate>

<<bind>>
<K->Integer,V->ArrayList<Date>>

<<bind>>
<K->ArrayList<Date>,V->Integer>

<<bind>>
<K->String,V->Integer>

<<bind>>
<K->User,V->LinkedHashMap<Integer,ArrayList<Date>>>

<<bind>>
<K->Integer,V->ArrayList<Date>>

<<bind>>
<K->Integer,V->ArrayList<Date>>

<<bind>>
<K->User,V->LinkedHashMap<Integer,ArrayList<Date>>>

- conn
- conn
- conn
- conn
- conn

- stmt
- stmt
- stmt
- stmt
- stmt
- stmt

- results
- results
- results
- results
- results
- results

pkg Entity

User
- – macAddress : String
- – name : String
- – password : String
- – email : String
- – gender : char
- – duration : long

- + getDuration() : long
- + setDuration(duration : long) : void
- + User(macAddress : String, name : String, password : String, email : String, gender : char)
- + User(macAddress : String)
- + getMacAddress() : String
- + getName() : String
- + getPassword() : String
- + getEmail() : String
- + getGender() : char
- + getYear() : String
- + getSchool() : String
- + toString() : String
- + validateMacAddress() : boolean
- + validatePassword() : boolean
- + validateEmail() : boolean
- + validateGender() : boolean
- + getTimeline() : ArrayList<MutablePair<LocationUpdate,Long>>
- + setTimeline(timeline : List<LocationUpdate>) : void

– timeline
{ordered} *

MutablePair<LocationUpdate,Long>

<<bind>>
<K–>LocationUpdate,V–>Long>

MutablePair `K, V`

Date

endTime

– startTime

Group
- – locationId : int
- – macAddressList : ArrayList<String>

- + Group(startTime : Date, endTime : Date, locationId : int, macAddressList : ArrayList<String>)
- + getStartTime() : Date
- + getEndtTime() : Date
- + setStartTime(startTime : Date) : void
- + setEndTime(endTime : Date) : void
- + getLocationId() : int
- + setLocationId(locationId : int) : void
- + getMembers() : ArrayList<String>
- + setMembers(macAddressList : ArrayList<String>) : void
- + hashCode() : int
- + equals(obj : Object) : boolean
- + groupSize() : int

HashMap<String,ArrayList<Integer>>

<<bind>>
<K–>String,V–>ArrayList<Integer>>

HashMap `K, V`

ArrayList<MutablePair<LocationUpdate,Long>>

<<bind>>
<E–>MutablePair<LocationUpdate,Long>>

ArrayList `E`

<<interface>>
List<LocationUpdate>

<<bind>>
<E–>LocationUpdate>

<<interface>>
List `E`

SemanticPlace
- – name : String
- – locationId : int

- + SemanticPlace(locationId : int, name : String)
- + getName() : String
- + getLocationId() : int
- + validName(name : String) : boolean
- + validLocationId(locationId : int) : boolean

CoLocation
- – userMac : String
- – companionMac : String
- – locationId : int
- – duration : long

- + CoLocation(userMac : String, companionMac : String, locationId : int, duration : long)
- + getUserMac() : String
- + getCompanionMac() : String
- + getLocationId() : int
- + getDuration() : long
- + setDuration(duration : long) : void

CrowdDensity
- – name : String
- – numPeople : int
- – density : int = 0

- + CrowdDensity(name : String, num : int)
- + getName() : String
- + getNumPeople() : int
- + getDensity() : int

Admin
- – name : String
- – password : String

- + Admin(name : String, password : String)
- + getName() : String
- + getPassword() : String

LocationUpdate
- – timeStamp : String
- – macAddress : String
- – locationId : int

- + LocationUpdate(timeStamp : String, macAddress : String, locationId : int)
- + getTimeStamp() : String
- + getDate() : Date
- + getMacAddress() : String
- + getLocationId() : int
- + validateTimeStamp() : boolean
- + validateMacAddress() : boolean
- + removeDuplicates(existingLocations : HashMap<String,ArrayList<Integer>>) : boolean
- + validateLocation(validLocations : ArrayList<Integer>) : boolean
- + equals(obj : Object) : boolean

pkg Json

HeatmapJsonServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String
+ dateIsValid(value : String) : boolean

TopKPopJsonServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
+ dateIsValid(value : String) : boolean
# doGet(request : HttpServletRequest, response : HttpServletResponse) : void
# doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

TopKNextPlaceJsonServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
generateUserTrajectory(visits : ArrayList<String[]>) : TreeMap<Date,String>
rankNextPlaces(map : HashMap<String,Integer>) : TreeMap<Integer,ArrayList<String>>
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

HttpServlet

LoginJsonServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
+ authenticatePassword(email : String, pwd : String) : Object
# doGet(request : HttpServletRequest, response : HttpServletResponse) : void
# doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

TopKCompJsonServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
generateUserTrajectory(updates : ArrayList<LocationUpdate>) : TreeMap<String,Integer>
groupByDuration(map : HashMap<String,CoLocation>) : HashMap<String,Long>
getUserCompanions(map : HashMap<String,Long>) : TreeMap<Long,ArrayList<User>>
generateCoLocations(map : HashMap<String,CoLocation>, macAddress : String, nearbyUpdates : ArrayList<LocationUpdate>, locId : int, windowEnd : String) : HashMap<String,CoLocation>
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

GroupDetectionJSONServlet
processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
doGet(request : HttpServletRequest, response : HttpServletResponse) : void

TreeMap<String,Integer>

TreeMap<Date,String>

HashMap<String,CoLocation>

HashMap<String,Long>

<<bind>>
<K->Date,V->String>

TreeMap
<<K, V>>

<<bind>>
<K->String,V->Integer>

<<bind>>
<K->Integer,V->ArrayList<String>>

<<bind>>
<K->String,V->CoLocation>

HashMap
<<K, V>>

<<bind>>
<K->Long,V->ArrayList<User>>

TreeMap<Long,ArrayList<User>>

TreeMap<Integer,ArrayList<String>>

<<bind>>
<K->String,V->Long>

UML Class Diagram

LinkedHashMap

HashMap

LinkedHashMap<Line,Long>

HashMap<Line,Long>

TreeMap<String,ArrayList<User>>

TreeMap<String,ArrayList<User>>

TreeMap<String,ArrayList<User>>

TreeMap<String,ArrayList<User>>

TreeMap<String,ArrayList<User>>

TreeMap<String,ArrayList<User>>

TreeMap

TreeMap<Image,ArrayList<String>>

TreeMap<String,ArrayList>

AdditionsJsonServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # getFileName(url : Port) : String
- # uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
- # error(zipFile : String, destination : String) : boolean
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

BreakdownServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # breakdownByYear(map : TreeMap<String,ArrayList<User>>) : TreeMap<String,ArrayList<User>>
- # breakdownByWeek(map : TreeMap<String,ArrayList<User>>) : TreeMap<String,ArrayList<User>>
- # breakdownByMonth(map : TreeMap<String,ArrayList<User>>) : TreeMap<String,ArrayList<User>>
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

BreakdownJsonServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # yearBreakdown(request : HttpServletRequest, users : ArrayList<User> , countGender : boolean, countEmail : boolean) : void
- # genderBreakdown(request : HttpServletRequest, response : HttpServletResponse, users : ArrayList<User> , countYear : boolean, countEmail : boolean) : void
- # yearMonthBreakdown(request : HttpServletRequest, users : ArrayList<User> , countYear : boolean, countGender : boolean) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

GroupDetectionServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

BootStripServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # getFileName(url : Port) : String
- # uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

TopicCompServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- - sortedSet(map : TreeMap<User,Long>) : LinkedHashMap<User,Long>
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

BootstrapJsonServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # getFileName(url : Port) : String
- # uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
- # unzip(zipFile : String, destination : String) : boolean
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

HttpServlet

BootstrapValidation
- + validLocations(k : ArrayList<User>) : new ArrayList<Integer>()
- + validateList(CHILD_PT_DIRECTORY : String, subsetrow : boolean, tempTermFile : File) : ArrayList[]
- + calculatePathDirectory(CHILD_PT_DIRECTORY : String, tempSubsetPath : File) : ArrayList[]
- + validateCalculate(boolean, boolean) : ArrayList<Integer>
- + validateLocationCHILDPT(FACTORY : String, idSubnIloop : boolean, tempLocationFile : File) : ArrayList[]
- + process(Request : HttpServletRequest, String : Image) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

HeatmapServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

TopicPieServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

TopicNewServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # sortByValue(map : TreeMap<String,Integer>) : TreeMap<Image,ArrayList<String>>
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

LoginServlet
- # processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
- # doGet(request : HttpServletRequest, response : HttpServletResponse) : void
- # doPost(request : HttpServletRequest, response : HttpServletResponse) : void
- + getServletInfo() : String

pkg Utilities

BootstrapJsonServlet

processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
getFileName(part : Part) : String
uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
unzip(zipFile : String, destination : String) : boolean
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

HttpServlet

− bootstrapValidation

BootstrapValidation

− validLocationIds : ArrayList<Integer> = new ArrayList<Integer>()

+ validateUserCSV(OUTPUTDIRECTORY : String, isBootstrap : boolean, tempDemoFile : File) : ArrayList[]
+ validateSemanticPlaceCSV(OUTPUTDIRECTORY : String, tempSemanticFile : File) : ArrayList[]
+ returnValidLocationIds(isBootstrap : boolean) : ArrayList<Integer>
+ validateLocationCSV(OUTPUTDIRECTORY : String, isBootstrap : boolean, tempLocationFile : File) : ArrayList[]
# processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
# doGet(request : HttpServletRequest, response : HttpServletResponse) : void
# doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

BootStrapServlet

processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
getFileName(part : Part) : String
uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
unzip(zipFile : String, destination : String) : boolean
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

− bootstrapValidation   − bootstrapValidation

AdditionalJsonServlet

processRequest(request : HttpServletRequest, response : HttpServletResponse) : void
getFileName(part : Part) : String
uploadFile(filePart : Part, UPLOAD_DIRECTORY : String) : String
unzip(zipFile : String, destination : String) : boolean
doGet(request : HttpServletRequest, response : HttpServletResponse) : void
doPost(request : HttpServletRequest, response : HttpServletResponse) : void
+ getServletInfo() : String

ConnectionManager

− RESOURCES : String = "/Utilities/connection.properties"
− user : String
− password : String
− url : String

− readDatabaseProperties() : void
+ getConnection() : Connection
+ close(conn : Connection, stmt : Statement, rs : ResultSet) : void
− initDBDriver() : void

DAOUtility

+ formatDatetime(datetime : String) : String

Utility

+ percentHalfRoundUp(count : int, total : int) : int