pkg

entity

AppUsage

- macAddress : String
- app : App
- timestamp : String
- appId : int

+ AppUsage(timestamp : String, macAddress : String, appId : int)
+ AppUsage(timestamp : String, macAddress : String, appId : int, app : App) : void
+ getTimestamp() : timestamp
+ getApp() : App
+ getAppId() : int
+ setTimestamp(timestamp : String) : void
+ setApp(app : App) : void
+ compareTo(o : AppUsage) : int
+ equals(obj : Object) : boolean
+ getDate() : Date
+ getMacAddress() : String
+ hashCode() : int
+ setAppId(appId : int) : void
+ setMacAddress() : void

User

- macAddress : String
- password : String
- gender : String
- name : String
- email : String
- cca : String

+ User(macAddress : String, name : String, password : String : email :String, gender : String, cca : String)
+ getMacAddress() : String
+ getPassword() : String
+ getGender() : String
+ getName() : String
+ getEmail() : String
+ setMacAddress(macAddress : String) : void
+ setName(name : String) : void
+ setPassword(password : String) : void
+ setGender(gender : String) : void
+ setEmailAdd(email : String) : void
+ getCca() : String
+ setCca() : void
+ getSchool() : String
+ getYear() : String

LocationUsage

- location : Location
- timestamp : String
- locationId : int
- macAddress : String

+ LocationUsage(timestamp : String, macAddress : String, locationId : int)
+ LocationUsage(timestamp : String, macAddress : String, location : Location)
+ getLocation() : Location
+ getTimestamp() : Date
+ setLocation(location : Location) : void
+ setTimestamp(timestamp : String) : void
+ getDate() : Date
+ getLocationId() : int
+ getMacAddress() : String
+ setLocationId(locationId : int) : void
+ setMacAddress(macAddress : String) : void

App

- appID : int
- appName : String
- appCategory : String

+ App(appId : int, appName : String, appCategory : String)
+ getAppId() : int
+ getAppName() : String
+ getAppCategory() : String
+ setAppId(appId : int) : void
+ setAppName(appName : String) : void
+ setAppCategory(appCategory : String) : void
+ compareTo(o : App) : int
+ equals(obj : Object) : boolean
+ hashCode() : int

Activeness

- endTime : long
- location : Location
- macAddress : String
- startTime : long

+ Activeness(startTime : long, endTime : long, macAddress : String, location : Location)
+ combine(active : Activeness) : Activeness
+ compareTo(o : Activeness) : Activeness
+ continuation(overlap : Activeness) : boolean
+ equals(obj : Object) : boolean
+ getEndTime() : long
+ getLocation() : Location
+ getMacAddress() : String
+ getStartTime() : long
+ getTime() : long
+ overlap(active : Activeness) : Activeness
+ setEndTime(endTime : long) : void
+ setLocation(location : Location) : void
+ setMacAddress(macAddress : String) : void
+ setStartTime() : void

Location

- locationId : int
- semanticPlace : String
- location : Location

+ Location(locationId : int, semanticPlace : int)
+ getLocationId() : int
+ getSemanticPlace() : String
+ setLocationId(locationId : int) : void
+ setSemanticPlace(semanticPlace : String) : void
+ getName(locationName : String) : void
+ Location(location : Location, semanticPlace : String)
+ compareTo(o : Location) : int
+ equals(obj : Object) : boolean
+ getLocation() : Location
+ hashCode() : int
+ setLocation(location : Location) : void

Admin

- username : String
- password : String

+ Admin(username : String, password : String)
+ getUsername() : String
+ getPassword() : String

Breakdown

- breakdown : ArrayList<HashMap<String,Breakdown>>
- message : String
- type : String

+ Breakdown()
+ Breakdown(message : String)
+ Breakdown(breakdown : ArrayList<HashMap<String,Breakdown>>)
+ addInList(map : HashMap<String,Breakdown>) : void
+ getBreakdown() : ArrayList<HashMap<String,Breakdown>>
+ getMessage() : String
+ getType() : String
+ setBreakdown(breakdown : ArrayList<HashMap<String,Breakdown>>) : void
+ setMessage(message : String) : void
+ setType(type : String) : void
+ toString() : String

1
1
1
1
1
1

pkg

dao

AppDAO

+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, appIdList : HashMap<Integer,String>) : int[]
+ retreveAppbyId(appId : int) : App
+ retrieveByCategory() : TreeMap<String,ArrayList<Integer>>

AppUsageDAO

- appList : TreeMap<String,AppUsage>
- duplicate : TreeMap<String,Integer>

+ AppUsageDAO()
+ add(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int
+ getAppsBySchool(school : String, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getSchoolsByCategory(priKSch : HashMap<String,String>, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getStudentsByCategory(priKMac : HashMap<String,String>, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getUserAppsForSocial(date : String, macAdd : String) : ArrayList<AppUsage>
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, macList : HashMap<String,String>, appIdList : HashMap<Integer,String>) : int[]
+ retrieveByAppCat(startHour : Date, endHour : Date, macAdd : String, appCat : String) : ArrayList<AppUsage>
+ retrieveByUser(macAdd : String, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ retrieveByUserHourly(macAdd:String,startHour:Date,endDate:Date):ArrayList<AppUsage>(macAdd : String, startHour : Date, endHour : Date) : ArrayList<AppUsage>
+ retrieveUserByDemo(startHour : Date, endHour : Date, demoArr : String[]) : ArrayList<User>
+ retrieveUsers(startDate : Date, endDate : Date) : ArrayList<User>
+ retrieveUsers(startDate : Date, endDate : Date, sql : String) : ArrayList<String>

UserDAO

+ UserDAO()
+ add(reader : CsvReader, userMap : TreeMap<Integer,String>, conn : Connection) : int[]
+ checkMacAdd(conn : Connection, macAdd : String) : boolean
+ getCCAs() : ArrayList<String>
+ insert(reader : CsvReader, userMap : TreeMap<Integer,String>, conn : Connection, macList : HashMap<String,String>) : int[]
+ retrieve(username : String, password : String) : User
+ retrieve(username : String) : User
+ retrieveByEmailId(username : String, password : String) : User
+ retrieveByMac(macaddress : String) : User

entity

App

AppUsage

User

LocationUsage

Location

Admin

LocationUsageDAO

- duplicate : TreeMap<String,Integer>
- locationList : TreeMap<String,LocationUsage>

+ add(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int
+ delete(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int[]
+ delete(conn : Connection, macAdd : String, startDate : String, endDate : String, locationId : int, semanticPlace : String) : ArrayLIst<LocationUsage>
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, locationIdList : HashMap<Integer,String>) : int[]
+ retrieve(date : Date, loc : String) : ArrayList<LocationUsage>
+ retrieve(startInLocation : Date, endInLocation : Date, prevLocationId : int, macaddress : String, totalAUList : ArrayList<LocationUsage>) : void
+ retrieveByFloor(date : Date, floor : String) : HashMap<String,LocationUsage>
+ retrieveByUser(macAdd : String, startDate : Date, endDate : Date) : ArrayList<LocationUsage>
+ retrievePeopleExceptUserLocationUsage(date : String, macAddress : String) : ArrayList<LocationUsage>
+ retrieveUserLocationUsage(date : String, macAdd : String) : ArrayList<LocationUsage>

ConnectionManager

- PROPS_FILENAME : String
- dbUser : String
- dbPassword : String
- dbURL : String

+ getConnection() : Connection
+ close(conn : Connection, stmt : Statement, rs : ResultSet) : void
+ close(conn : Connection, stmt : Statement) : void
+ close(conn : Connection) : void
+ initDBDriver() : void
+ readLocalDatabaseProperties() : boolean
+ readOpenshiftDatabaseProperties() : boolean

LocationDAO

+ checkLocationId(conn : Connection, locationId : int) : String
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, locationIdList : HashMap<Integer,String>) : int[]
+ retrieve(floor : String) : ArrayList<String>
+ retrieveAll() : ArrayList<String>
+ retrieveSemPl(semanticPlace : String) : Location

Utility

- categories : ArrayList<String>
- df : SimpleDateFormat
- schoolList : ArrayList<String>
- sdf : SimpleDateFormat

+ checkCategory(chk : String) : boolean
+ checkDate(str : String) : boolean
+ checkEmail(str : String) : boolean
+ checkHexadecimal(str : String) : boolean
+ checkOnlyDate(str : String) : boolean
+ checkPassword(str : String) : boolean
+ checkSchools(schoolString : int) : boolean
+ compareSchools(schools : ArrayList<String>) : ArrayList<String>
+ daysBetweenDates(startDate : Date, endDate : Date) : long
+ formatDate(date : Date) : String
+ formatOnlyDate(date : Date) : String
+ getNextHour(startDate : Date) : Date
+ getSchool(email : String) : String
+ getSchoolList() : ArrayList<String>
+ loadCategories() : void
+ loadSchools() : void
+ parseDate(date : String) : Date
+ parseInt(str : String) : int
+ parseOnlyDate(date : String) : Date
+ parseString(str : String) : String
+ retrieveCategories() : ArrayList<String>
+ retrieveSchools() : ArrayList<String>
+ secondsBetweenDates(startDate : Date, endDate : Date) : long

AdminDAO

+ retrieve(username : String, password : String) : Admin
+ retrieve(username : String) : Admin

InitDAO

+ enableForeignKey(conn : Connection) : void
+ truncateTable(conn : Connection) : void

pkg

dao

AppDAO

+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, appIdList : HashMap<Integer,String>) : int[]
+ retreveAppbyId(appId : int) : App
+ retrieveByCategory() : TreeMap<String,ArrayList<Integer>>

AppUsageDAO

- appList : TreeMap<String,AppUsage>
- duplicate : TreeMap<String,Integer>

+ AppUsageDAO()
+ add(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int
+ getAppsBySchool(school : String, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getSchoolsByCategory(priKSch : HashMap<String,String>, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getStudentsByCategory(priKMac : HashMap<String,String>, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ getUserAppsForSocial(date : String, macAdd : String) : ArrayList<AppUsage>
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, macList : HashMap<String,String>, appIdList : HashMap<Integer,String>) : int[]
+ retrieveByAppCat(startHour : Date, endHour : Date, macAdd : String, appCat : String) : ArrayList<AppUsage>
+ retrieveByUser(macAdd : String, startDate : Date, endDate : Date) : ArrayList<AppUsage>
+ retrieveByUserHourly(macAdd:String,startHour:Date,endDate:Date):ArrayList<AppUsage>(macAdd : String, startHour : Date, endHour : Date) : ArrayList<AppUsage>
+ retrieveUserByDemo(startHour : Date, endHour : Date, demoArr : String[]) : ArrayList<User>
+ retrieveUsers(startDate : Date, endDate : Date) : ArrayList<User>
+ retrieveUsers(startDate : Date, endDate : Date, sql : String) : ArrayList<String>

UserDAO

+ UserDAO()
+ add(reader : CsvReader, userMap : TreeMap<Integer,String>, conn : Connection) : int[]
+ checkMacAdd(conn : Connection, macAdd : String) : boolean
+ getCCAs() : ArrayList<String>
+ insert(reader : CsvReader, userMap : TreeMap<Integer,String>, conn : Connection, macList : HashMap<String,String>) : int[]
+ retrieve(username : String, password : String) : User
+ retrieve(username : String) : User
+ retrieveByEmailId(username : String, password : String) : User
+ retrieveByMac(macaddress : String) : User

LocationUsageDAO

- duplicate : TreeMap<String,Integer>
- locationList : TreeMap<String,LocationUsage>

+ add(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int
+ delete(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection) : int[]
+ delete(conn : Connection, macAdd : String, startDate : String, endDate : String, locationId : int, semanticPlace : String) : ArrayList<LocationUsage>
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, locationIdList : HashMap<Integer,String>) : int[]
+ retrieve(date : Date, loc : String) : ArrayList<LocationUsage>
+ retrieve(startInLocation : Date, endInLocation : Date, prevLocationId : int, macaddress : String, totalAUList : ArrayList<LocationUsage>) : void
+ retrieveByFloor(date : Date, floor : String) : HashMap<String,LocationUsage>
+ retrieveByUser(macAdd : String, startDate : Date, endDate : Date) : ArrayList<LocationUsage>
+ retrievePeopleExceptUserLocationUsage(date : String, macAddress : String) : ArrayList<LocationUsage>
+ retrieveUserLocationUsage(date : String, macAdd : String) : ArrayList<LocationUsage>

LocationDAO

+ checkLocationId(conn : Connection, locationId : int) : String
+ insert(reader : CsvReader, errMap : TreeMap<Integer,String>, conn : Connection, locationIdList : HashMap<Integer,String>) : int[]
+ retrieve(floor : String) : ArrayList<String>
+ retrieveAll() : ArrayList<String>
+ retrieveSemPl(semanticPlace : String) : Location

AdminDAO

+ retrieve(username : String, password : String) : Admin
+ retrieve(username : String) : Admin

InitDAO

+ enableForeignKey(conn : Connection) : void
+ truncateTable(conn : Connection) : void

Utility

- categories : ArrayList<String>
- df : SimpleDateFormat
- schoolList : ArrayList<String>
- sdf : SimpleDateFormat

+ checkCategory(chk : String) : boolean
+ checkDate(str : String) : boolean
+ checkEmail(str : String) : boolean
+ checkHexadecimal(str : String) : boolean
+ checkOnlyDate(str : String) : boolean
+ checkPassword(str : String) : boolean
+ checkSchools(schoolString : int) : boolean
+ compareSchools(schools : ArrayList<String>) : ArrayList<String>
+ daysBetweenDates(startDate : Date, endDate : Date) : long
+ formatDate(date : Date) : String
+ formatOnlyDate(date : Date) : String
+ getNextHour(startDate : Date) : Date
+ getSchool(email : String) : String
+ getSchoolList() : ArrayList<String>
+ loadCategories() : void
+ loadSchools() : void
+ parseDate(date : String) : Date
+ parseInt(str : String) : int
+ parseOnlyDate(date : String) : Date
+ parseString(str : String) : String
+ retrieveCategories() : ArrayList<String>
+ retrieveSchools() : ArrayList<String>
+ secondsBetweenDates(startDate : Date, endDate : Date) : long

controller

AddBatchController

+ addBatch(filePart : Part, userErrMap : TreeMap<Integer,String>, delErrMap : TreeMap<Integer,String>, auErrMap : TreeMap<integer,String>, luerrMap : TreeMap<Integer,String>) : TreeMap<String,Integer>

BootstrapController

+ bootstrap(filePart : Part, userErrMap : TreeMap<Integer,String>, appErrMap : TreeMap<Integer,String>, locErrMap : TreeMap<Integer,String>, auErrMap : TreeMap<Integer,String>, luErrMap : TreeMap<Integer,String>, delErrMap : TreeMap<Integer,String>) : TreeMap<String,Integer>

BasicAppController

+ filterDemo(demo : String, demoType : String, userList : ArrayList<User>) : ArrayList<User>
+ generateAppCategory(startDate : Date, endDate : Date) : TreeMap<String,Double[]>
+ generateDiurnalReport(startDate : Date, demoArr : String[]) : Breakdown
+ generatePercentage(breakdown : Breakdown, total : double) : void
+ generateReport(startDate : Date, endDate : Date, userList : ArrayList<User>, total() : double) : Breakdown
+ generateReportByDemo(startDate : Date, endDate : Date, demoArr : String[]) : Breakdown

DeleteController

+ delete(macAdd : String, startDate : String, endDate : String, startTime : String, endTime : String, locId : String, semanticPl : String, error : ArrayList<String>) : ArrayList<LocationUsage>

HeatmapController

+ generateHeatmap(datetime : Date, floor : String) : TreeMap<String,ArrayList<LocationUsage>>

: SmartphoneOveruseController

+ generateReport(user : User, startDate : Date, endDate : Date) : TreeMap<String,String>

SocialActivenessController

+ generateAwarenessReport(onlyDate : String, macAddress : String, errors : String) : HashMap<String,Breakdown>
+ overlapUser(userList : HashMap<Location,ArrayList<Activeness>>, singleList ArrayList<Activeness> : int, ac Activeness : int, loc Location : int) : void
+ checkMinutes(userList : HashMap<Location,ArrayList<Activeness>>) : void

TopKReportController

+ getTopkApp(topK : int, school : String, strDate : Date, endDate : Date, errors : int) : ArrayList<HashMap<String,String>>
+ getTopkStudents(topK : int, cat : String, strDate : Date, endDate : Date, errors : String) : ArrayList<HashMap<String,String>>
+ getTopkSchool(topK : int, cat : String, strDate : Date, endDate : Date, errors : String) : ArrayList<HashMap<String,String>>