Database Definition File

To be held in a master list of all parties

* Guest list object
  + A list of references to users to be held in party data object
* Party Data Object
  + Fields contained in Data Type Party
    - userNameofHost: String
      * String containing the name of the host
    - timeZoneofHost: eNum type
      * Time chosen by the host for the party to be held
    - dateOfParty date java.util.Date:
      * Date the host chose for the party
    - linkRequestTime java.util.Date:
      * Holds the time(date of party-10 minutes ) the host will be asked for the created link to start a party.
    - partyLink String
      * The string held to access the party
    - partyName String
      * The string of the name describing the watch party
    - guestList Guestlist
      * Guestlist that holds the list of all the users attending the party
      * Hold the list of the users
    - private boolean
      * This boolean if sent to true means the party is private and will not appear in a party search.
    - inviteOnly
      * Choice to set the party to invite only. Users may only join via RSVP
  + Store Procedures
    - getTimezoneofHost()
      * Find the time zone of where the party is being hosted
    - setTimezoneofHost()
      * Sets the time zone of where the party is being hosted
    - getUserNameofHost()
      * Finds the name of the host
    - setUserNameofHost()
      * Sets the name of the host for the party
    - getdateOfParty()
      * Find the date of the party that the host chooses
    - setdateOfParty()
      * Set the date of the party that the host chooses
    - getlinkRequestTime()
      * Find the request time of the party is hosting
    - setlinkRequestTime()
      * Set the request time of the party is hosting
    - getPartyLink()
      * Find the link of the hosting party
    - setPartyLink(link String)
      * Set the link of the hosting party
    - getPartyName()
      * Gets the name of the party imputed by the host
    - setPartyName(name String)
      * Sets the party
    - getguestList()
      * Gets the guest list input by the host
    - setguestList(guestList Guestlist()
      * Set the guest list
    - isPrivate()
      * Return the value of the boolean private
    - isinviteOnly()
      * Return the value of the boolean invite only
    - inviteGroup(invitedList GuestList)
      * Invite a list of users to a party
    - invite(invited User)
      * invite a single user to a party
    - addToGuestList(rvsp User)
      * Add a user to a guestlist object
    - invite(invited User)
      * Invite user to the party
    - requestLinkFromHost(userNameofHost String)
      * A private method to request the link from the
    - getHostLanguage
      * Return the value for the hosts native language
    - searchRequest(userInput String)
      * Return a list of the public parties that share the most similarity with the search query
* Movie Data Object

To be held in a master list of all Movies

* + Datafields held in Movie data object
    - MovieName String
      * Holds the title of the movie in the language of the country of origin
    - Synopsis Path
      * Holds a path to a synopsis to be scraped from the web
    - Genre String
      * Holds the genre of the selected movie
    - CastList Path
      * Holds a path to a castlist to be scraped from the web
    - DateOfRelease java.util.Date
      * Holds the original date of release of the film
    - MoviePoster Path
      * It holds the path to the movie image that match with the movie poster
  + Store Procedures
    - getName()
      * Get the title of the movie in the language of the country of origin
    - setName(newName String)
      * Create and store the name of the film
    - getSynopsis()
      * Access the synopsis from the web
    - setSynopsis(path Path)
      * Changes the held path to the synopsis
    - getMoviePoster()
      * Follows the path to the movie image that match with the movie poster
    - setMoviePoster(path Path)
      * sets the path to the movie image that match with the movie poster
    - getCastlist()
      * Follows the path to a castlist to be scraped from the web
    - setCastlist(Path p)
      * Sets the path to a castlist to be scraped from the web
    - getGenre()
      * Returns the genre of the movie
    - setGenre(Genre String)
      * Sets the genre of the movie
    - getDateOfRelease()
      * Returns the date that the movie was released
    - setDateOfRelease (date java.util.Date)
      * Sets the date that the movie was released
* User Data Object
  + To be held in a master list of all User
    - userName: String
      * String containing the name of the host
    - hashedPassword: String
      * String of encrypted password
    - timeOfRegistration: java.util.Date class
      * Time of the registration by user
    - timeZone: eNum type
      * Time chosen by the host for the party to be held
    - birthDate: java.util.Dateclass
      * Date of birth registered by the user
    - isHost: Boolean
      * Holds true if the user is Host, else false
    - hostRating: Double
      * Holds the rating of the given Host
    - partipantRating Rating: Double
      * Holds the current the rating of the given participant
    - partyBeingHostedByUser: Party
      * Holds the party class hosted by given user
    - userHistory: ScehedueledDate Class
      * Holds the user history of the given user
    - languages Spoken String[]
      * Holds the String of languages selected as spoken language by the user
  + **Store Procedures**
    - userNameChange(newUsername String)
      * Changes the User’s name from the stored string to a new string
    - passwordChange(newPassword)
      * Change the password and encrypt it.
    - getTimezone()
      * Access the time zones and your preferred region
    - getBirthDate()
      * Acces the user’s date of birth
    - getUserName()
      * Access the users username
    - gethashedPassword()
      * Get the hash of the encrypted password
    - setTimezone()
      * Get the timezone of the user.
    - setBirthDate()
      * Set the birthdate of the user
    - setUserName()
      * Set the Username of the user
    - sethashedPassword()
      * Set the hashed password
    - newParty(newParty Party)
      * Begin the creation of the party.