GENERAL INFORMATION

Query the database by performing:

DBQueries db = DBQueries.getInstance()

db.*function*

ResultSet (rs): A set of rows containing the output of the SQL query. There is a pointer that originally points to just before the first row in the set, so you must call rs.next() in order to access the first row of the data.

Useful operations with ResultSet rs:

if/while (rs.next()) — returns true if that next row exists, false otherwise. By accessing the information this way, you already handle calling rs.next() to view the first row of data

rs.getInt(int i) — returns the integer stored at that row in column i (columns begin at 1). There are similar functions for getting other value types

rs.getString(String s) — returns the string stored at that row in the column labeled s. There are similar functions for getting other value types

MISC FUNCTIONS

[*helper*] **void nullEmail(String email)** throws IllegalArgumentException if email is null

[*helper*] **void nullPassword(String p)** throws IllegalArgumentException if p is null

[*helper*] **void nullGroup(String group_id)** throws IllegalArgumentException if group_id is null

[*helper*] **void validDueDate(String date)** throws an IllegalArgumentException if date is null, isn't 10 characters, or if characters 4 and 6 aren't '-"

[*helper*] **void nullGroupName(String name)** throws an IllegalArgumentException if name is null or >20 characters

[*helper*] **String generateCode(int length)** creates a random *length*-character string using the English alphabet and 0-9

USER FUNCTIONS

int login(String email, String password) handles login functionality. Returns 0 if email doesn't exist in the database, 1 if password was incorrect, 2 if successful

[*helper*] **boolean userExists(String email)** returns whether the email exists in the database

boolean enteredCorrectPassword(String email, String oldPassword) returns whether the given password matches the user's password

boolean signUp(String email, String password) handles signing a new user up to the app, returns whether it was successful

String forgotPassword(String email) generates, stores, & returns a reset code for the user in the database, and sets the expiration date to 1 week from the current day.

int allowReset(String email, String code) returns 2 if the user should be given permission to update his/her password, 1 if the code already expired, 0 if the code is incorrect

boolean resetPassword(String user, String password) handles resetting the user's password with the new one, returns whether the update was successful

boolean setNickname(String user, String nickname) handles setting the user's nickname, returns whether the update was successful

String getNickname(String user) returns the user's nickname, null if it was never set

boolean setIcon(String user, String icon) handles setting the user's icon, returns whether the update was successful

String getIcon(String user) returns the user's icon, null if it hasn't been set

ResultSet userGroups(String email) returns all the groups a user is in, null if none

ResultSet userBills(String email) returns all bill info stored for each bill assigned to the user, null if none

ResultSet userBillsInGroup(String email, String group_id) returns all bill info stored for each bill assigned to the user in the given group, null if none

boolean deleteAccount(String email) handles safe deletion of a user's account, and returns whether it was successful

BILL FUNCTIONS

[*helper*] **boolean billExists(String bill_id)** returns whether bill_id is being used in the database

boolean addBill(String user, String group, String name, double amt, String date, String desc) handles functionality for adding a bill for a given user_id and group_id, returns whether the operation was successful

boolean deleteBill(String user_id, String group_id, String bill_id) handles deleting a bill, returns whether the operation was successful

boolean changeDueDate(String bill_id, String due_date) handles changing the due date for the specified bill, returns whether it was successful

boolean changeAmount(String bill_id, double amt) handles changing the bill's amount value for the specified bill, returns whether it was successful

GROUP FUNCTIONS

[*helper*] **boolean groupExists(String group_id)** returns whether the group_id is being used in the database

[*helper*] **boolean codeInUse(String code)** checks and returns whether that group invite code is being used in the database already

String getInviteCode(String group_id) generates, stores, and returns an invite code for the group. The code expires after one week. GENERATE A NEW CODE FOR EACH INVITED USER

String getCodeGroup(String code) returns the group_id associated with that invite code, null if the code doesn't exist in the database

ArrayList<String> groupMembers(String group_id) returns an array list of all the email addresses of the users of a given group. The list is empty if the group doesn't exist.

String groupName(String group_id) returns the name associated with the given group_id, an empty string if the group_id doesn't exist in the database

[*helper*] **ResultSet groupBills(String group_id)** returns all bill info stored for each bill in a group, null if none

boolean createGroup(String user_id, String group_name) handles creating a group for the user with the given name, returns whether it was successful

int addUserToGroup(String user_id, String code) returns 0 if the code was wrong, 1 if the code had expired, 2 if the user doesn't exist in the database, 3 if the addition was successful

boolean leaveGroup(String user_id, String group_id) handles a safe removal of a user from a group, returns whether it was successful

boolean deleteGroup(String group_id) handles safe deletion of a group, returns whether it was successful

int groupParticipation(String group_id) returns the number of users in a group, -1 if the group doesn't exist