Demo

Description:

I will be demoing a test app that will demonstrate all methods and functionality of the local SQlight database through Android Studio. The Classes and Tables I will demo includes:

-user class
-dataframe 33 class (data from Bluetooth)
-dataframe 35 class (data from Bluetooth)

-user table
-dataframe 33 table
-dataframe 35 table
-history table
-error table

-Password is visible in table for denomination of functionality. Once verified Password will be hidden with stars by changing the storage type in the table.

BTN Login:

checks and sees if account exists and checks to see if user matches password

BTN Create Account:

creates a new account with new user and password

if new username matches existing username, account will not be created

A new row in the “user_info” table will be added with the following default values:

```
//Test Data
int FTP = 1;
int pz_1 = 1;
int pz_2 = 2;
int pz_3 = 3;
int pz_4 = 4;
int pz_5 = 5;
int pz_6 = 6;
int pz_7 = 7;
```

BTN Delete Account:

Delete an account with an existing username and matching password

BTN Class Testing:

Enters default values shown below into classes and displays toast commands displaying the functionality of all go getters within class.

User	dataframe33	dataframe35

`int FTP = 1;`
`int pz_1 = 1;`
`int pz_2 = 2;`
`int pz_3 = 3;`
`int pz_4 = 4;`
`int pz_5 = 5;`
`int pz_6 = 6;`
`int pz_7 = 7;`
	`double time_33 = 1.0;`
	`int interval = 2;`
	`int power = 3;`
	`int total_cal = 4;`
	`double split_pace = 5.0;`
	`int split_power = 6;`
	`double split_cal = 7.0;`
	`double last_split_time = 8.0;`
	`double last_split_dist = 9.0;`
		`double time_35 = 10.0;`
		`double dist = 11.0;`
		`double drive_len = 12.0;`
		`double drive_time = 13.0;`
		`double stroke_rec_time = 14.0;`
		`double stroke_dist = 15.0;`
		`double peak_drive_force = 16.0;`
		`double avg_drive_force = 17.0;`
		`double work_per_stroke = 18.0;`
		`int stroke_count = 19;`

After toast commands displays, *a row of data will be added to the "dataframe33_info" and "dataframe35_info" tables!*

BTN Testing Go Getters:

Test receiving data from all necessary tables: “user_info”, “dataframe33”, “dataframe35”

Note: “history_info” and “error_info” tables are for user viewing only. No calculation necessary.

Display all attributes defined by

Table	Defined by
user_info	username: user input
dataframe33	Time33: 1
dataframe35	Time35: 10

Results should be:

user_info	dataframe33_info	dataframe35_info

int FTP = 1;
int pz_1 = 1;
int pz_2 = 2;
int pz_3 = 3;
int pz_4 = 4;
int pz_5 = 5;
int pz_6 = 6;
int pz_7 = 7;
	double time_33 = 1.0;
	int interval = 2;
	int power = 3;
	int total_cal = 4;
	double split_pace = 5.0;
	int split_power = 6;
	double split_cal = 7.0;
	double last_split_time = 8.0;
	double last_split_dist = 9.0;
		double time_35 = 10.0;
		double dist = 11.0;
		double drive_len = 12.0;
		double drive_time = 13.0;
		double stroke_rec_time = 14.0;
		double stroke_dist = 15.0;
		double peak_drive_force = 16.0;
		double avg_drive_force = 17.0;
		double work_per_stroke = 18.0;
		int stroke_count = 19;

Updated FTP:

```
db.updateuserFTP(et_username.getText().toString(), FTP: 8, pz_1: 9, pz_2: 10, pz_3: 11, pz_4: 12, pz_5: 13, pz_6: 14, pz_7: 15)
```

BTN Update Password:

If username exists, password entered will be updated within the “user_info” table

BTN Update FTP:

If username exists, FTP and all power zones will be updated in the “user_info” table for that user account

FTP updated values:

```
db.updateuserFTP(et_username.getText().toString(), FTP: 8, pz_1: 9, pz_2: 10, pz_3: 11, pz_4: 12, pz_5: 13, pz_6: 14, pz_7: 15)
```

BTN Delete Real Time Tables:

Delete all data in “dataframe33_info” and “dataframe35_info” tables

BTN Insert Hist/Error Tables:

With username entered, **workout 1** will be entered to the history table and **5** will be entered to the error table.

BTN View Hist/Error Tables:

With username entered, display all data in “history_info” table and display all data in “error_info” table

Show data in all tables!!

Storage verification will take place next semester with integration of Bluetooth subsystem

(281 terabytes)

Email verification is still being worked on(was not mentioned until after final presentation)

-what I’m following: https://www.geeksforgeeks.org/how-to-add-user-registration-with-email-verification-in-android/

-encryption and description of incoming data will take place next semester with the integration of Bluetooth subsystem