Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 45: TI Project #3
Bi-Weekly Update 5

Alyson Garlick, Diego Gumucio, Meredith McKean, Nicholas McNamara

Sponsor: Matthew Krebs

TA: Pranav Dhulipala

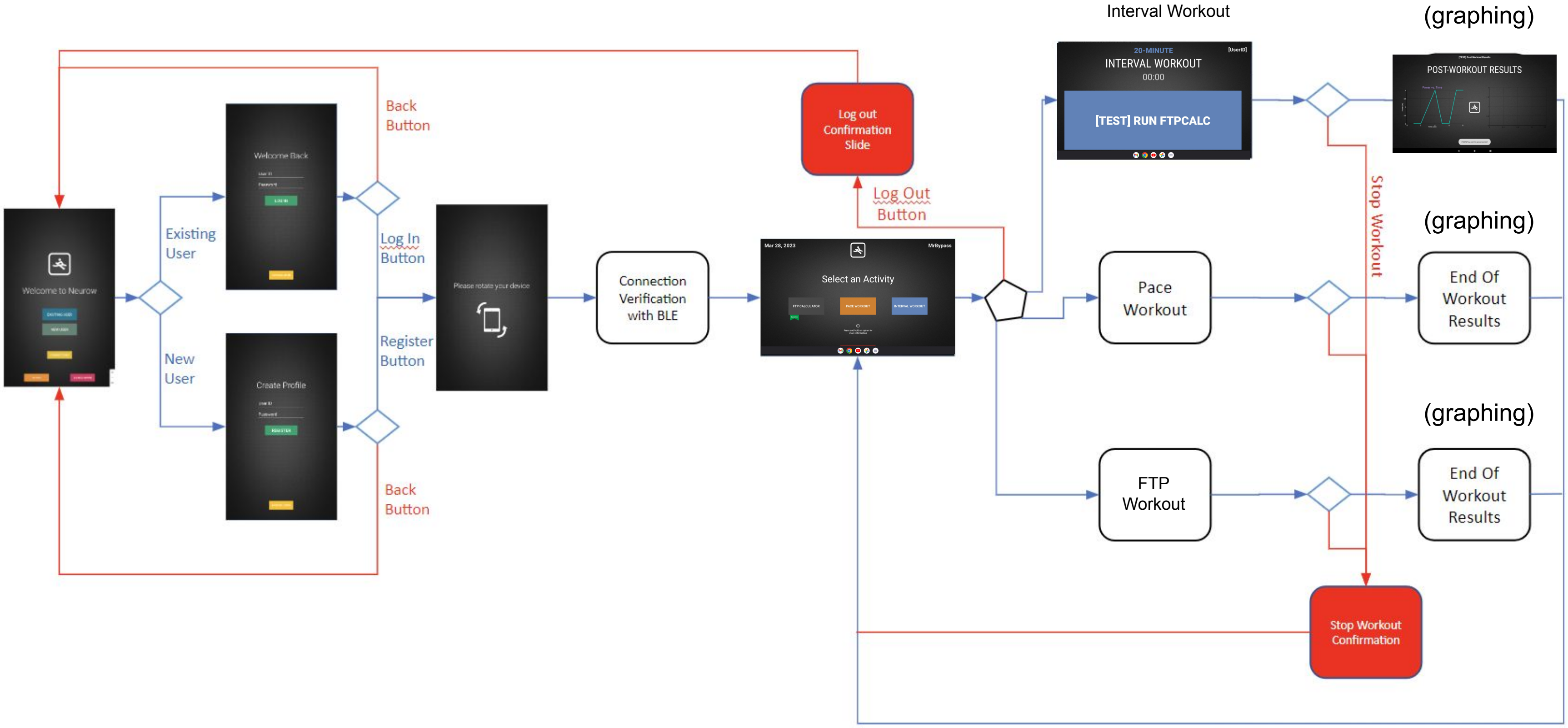
Project Summary

- A problem of older less expensive rower machines is that they lack features of higher end models and those from other types of equipment. These features keep the user engaged and focused on improving their overall fitness via Functional Threshold Power. FTP is a metric to maximize performance gains in the shortest span of time.

- To provide this experience, our subsystems will integrate on an Android application to connect to a Concept2 rower machine and provide:
  - A measurement of a user's FTP
  - Workouts consisting of seven different power zones
  - Recommendations based on their respective FTP
  - Storage for user's history local to the device.

Application Flow Overview

Key	
Blue	Main Path
Red	Reverse Path Precautions



Project Timeline

Project Definition (completed 9/9)	Subsystem Designs and Testing (completed 12/2)	Integration of Database & Workouts and UI & Bluetooth (to complete by 3/1)	Integration (to complete by 3/5)	Front End Validation (to complete by 3/24)	Back End Validation (to complete by 4/12)	Demo and Report (to complete by 4/29)

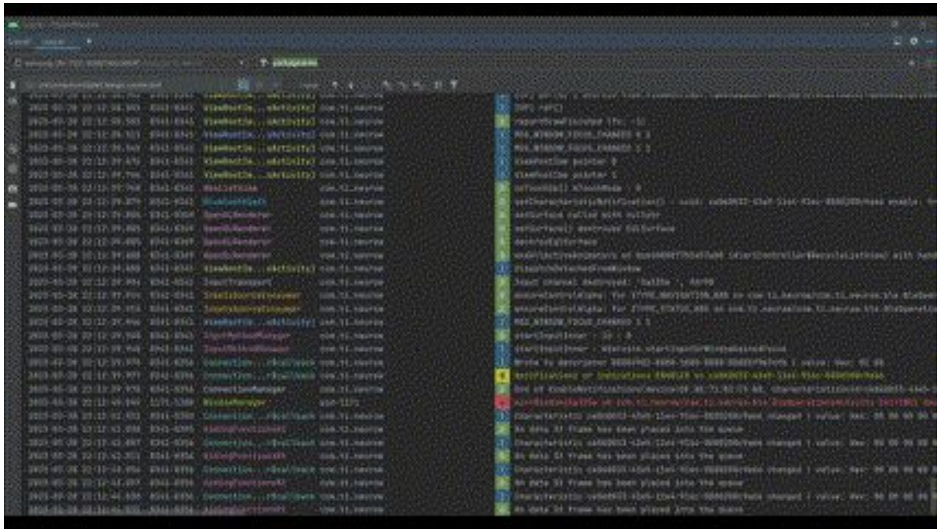
Database + Bluetooth

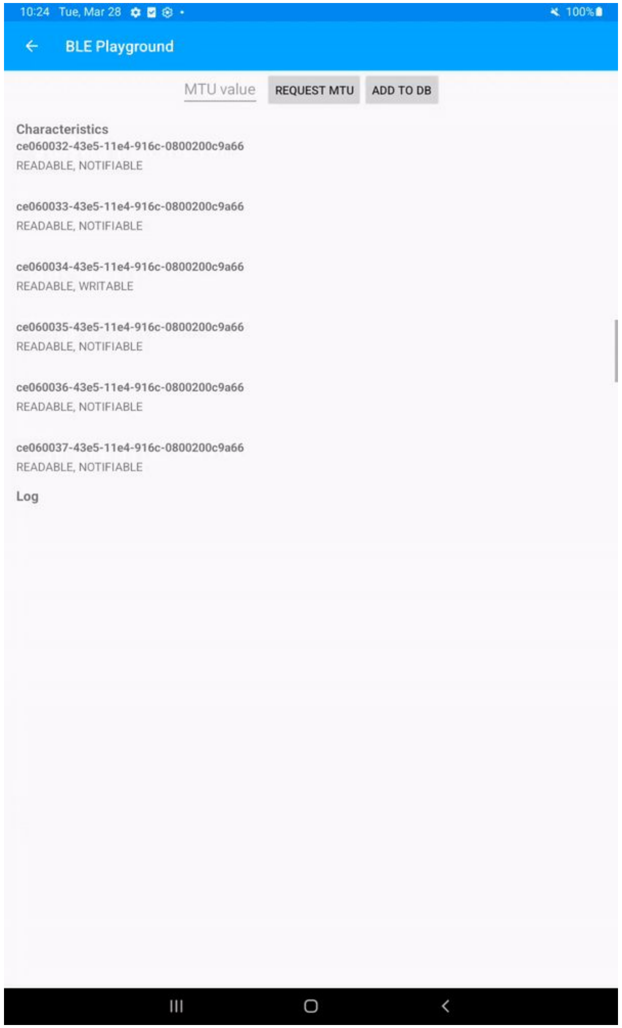
Nick McNamara and Meredith McKean

Accomplishments since last update 25 hrs of effort each	Ongoing progress/problems and plans until the next presentation
● Validated exit of BLE driver application ● Database and Bluetooth communication ● Database now holds real data	● Creating method to add bluetooth data to database in a time range ● Data cleaning ● Database Storage Validation

Database + Bluetooth

- BLE Connection runs in background w/o the need of an activity





10:24 Tue, Mar 28 100%

BLE Playground

MTU value | REQUEST MTU | ADD TO DB

Characteristics

ce060032-43e5-11e4-916c-0800200c9a66
READABLE, NOTIFIABLE

ce060033-43e5-11e4-916c-0800200c9a66
READABLE, NOTIFIABLE

ce060034-43e5-11e4-916c-0800200c9a66
READABLE, WRITABLE

ce060035-43e5-11e4-916c-0800200c9a66
READABLE, NOTIFIABLE

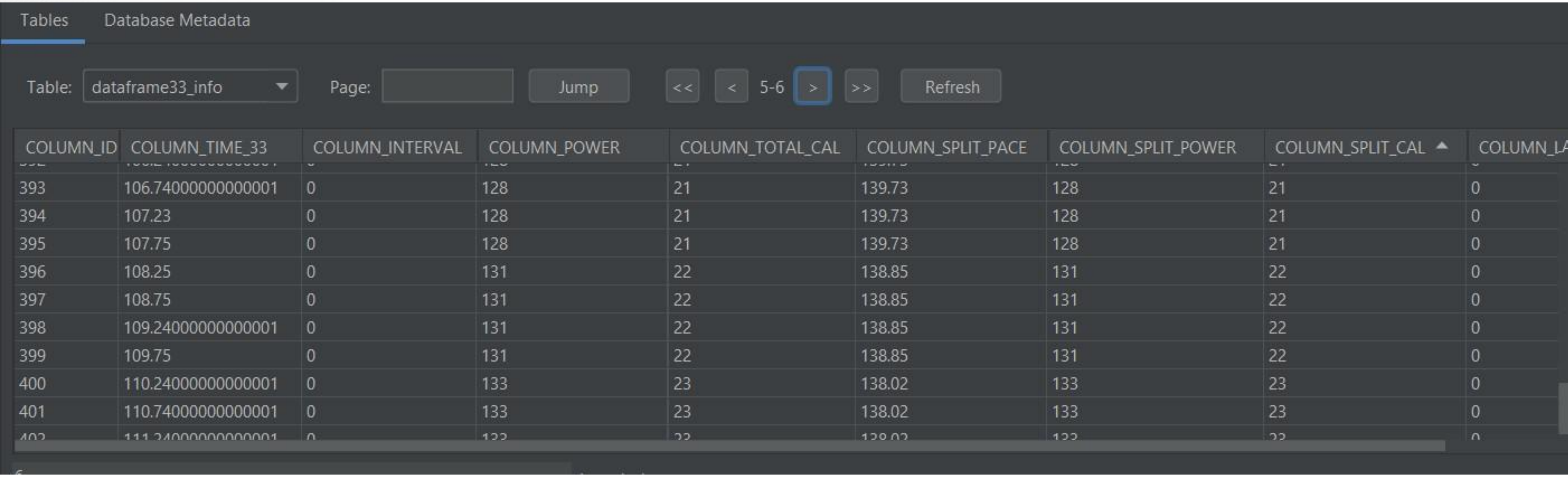
ce060036-43e5-11e4-916c-0800200c9a66
READABLE, NOTIFIABLE

ce060037-43e5-11e4-916c-0800200c9a66
READABLE, NOTIFIABLE

Log

Database + Bluetooth

Dataframe33s stored in the SQLite



COLUMN_ID	COLUMN_TIME_33	COLUMN_INTERVAL	COLUMN_POWER	COLUMN_TOTAL_CAL	COLUMN_SPLIT_PACE	COLUMN_SPLIT_POWER	COLUMN_SPLIT_CAL ▲	COLUMN_LA
393	106.74000000000001	0	128	21	139.73	128	21	0
394	107.23	0	128	21	139.73	128	21	0
395	107.75	0	128	21	139.73	128	21	0
396	108.25	0	131	22	138.85	131	22	0
397	108.75	0	131	22	138.85	131	22	0
398	109.24000000000001	0	131	22	138.85	131	22	0
399	109.75	0	131	22	138.85	131	22	0
400	110.24000000000001	0	133	23	138.02	133	23	0
401	110.74000000000001	0	133	23	138.02	133	23	0
402	111.24000000000001	0	133	23	138.02	133	23	0

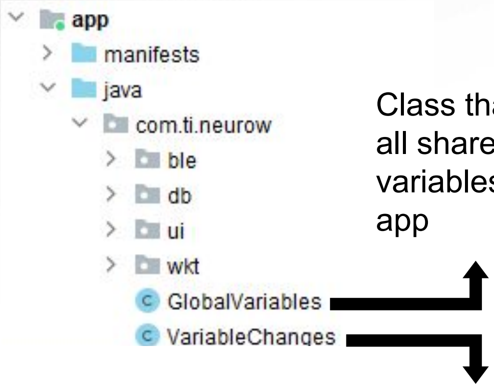
UI + Workout

Alyson Garlick and Diego Gumucio

Accomplishments since last update 30 hrs of effort each	Ongoing progress/problems and plans until the next presentation
● Created GlobalVariables class ● Created VariableChanges class to listen for changes in variables ● Integrated UI into workout methods to print feedback to screen ● Implemented graphing into UI to display Power vs Time data from workout methods ● Integrated UI into suggestion and prediction methods	● Finish implementing graphing technique for Power of Pull graph ● Establish remaining shared variables and listeners ● Determine whether workout engine/UI is best to call certain methods

UI + Workout

Alyson Garlick and Diego Gumucio

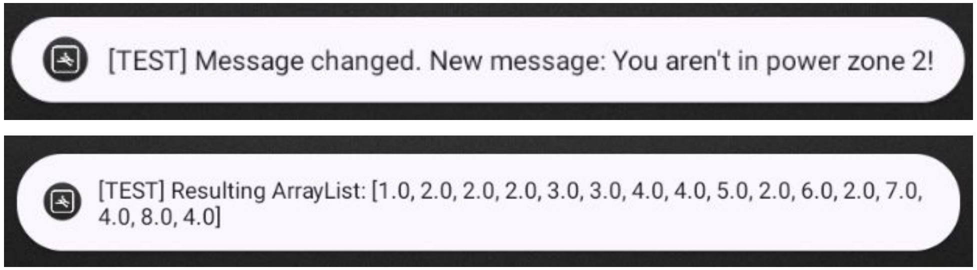


```
app
  manifests
  java
    com.ti.neurow
      ble
      db
      ui
      wkt
      GlobalVariables
      VariableChanges
```

Class that holds all shared variables in the app

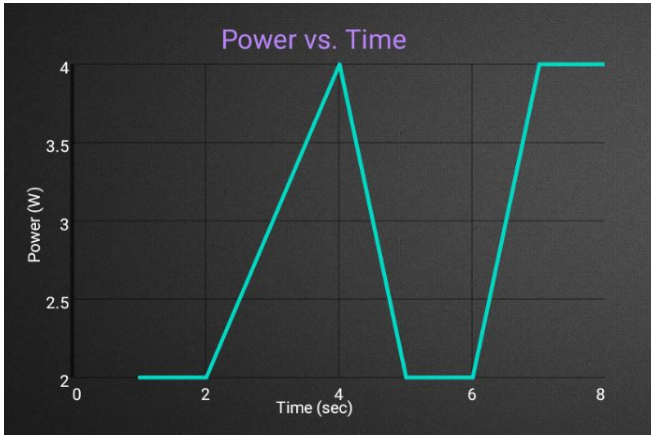
Class that detects changes in certain variables through a listener

Test Toasts (UI print-equivalent)

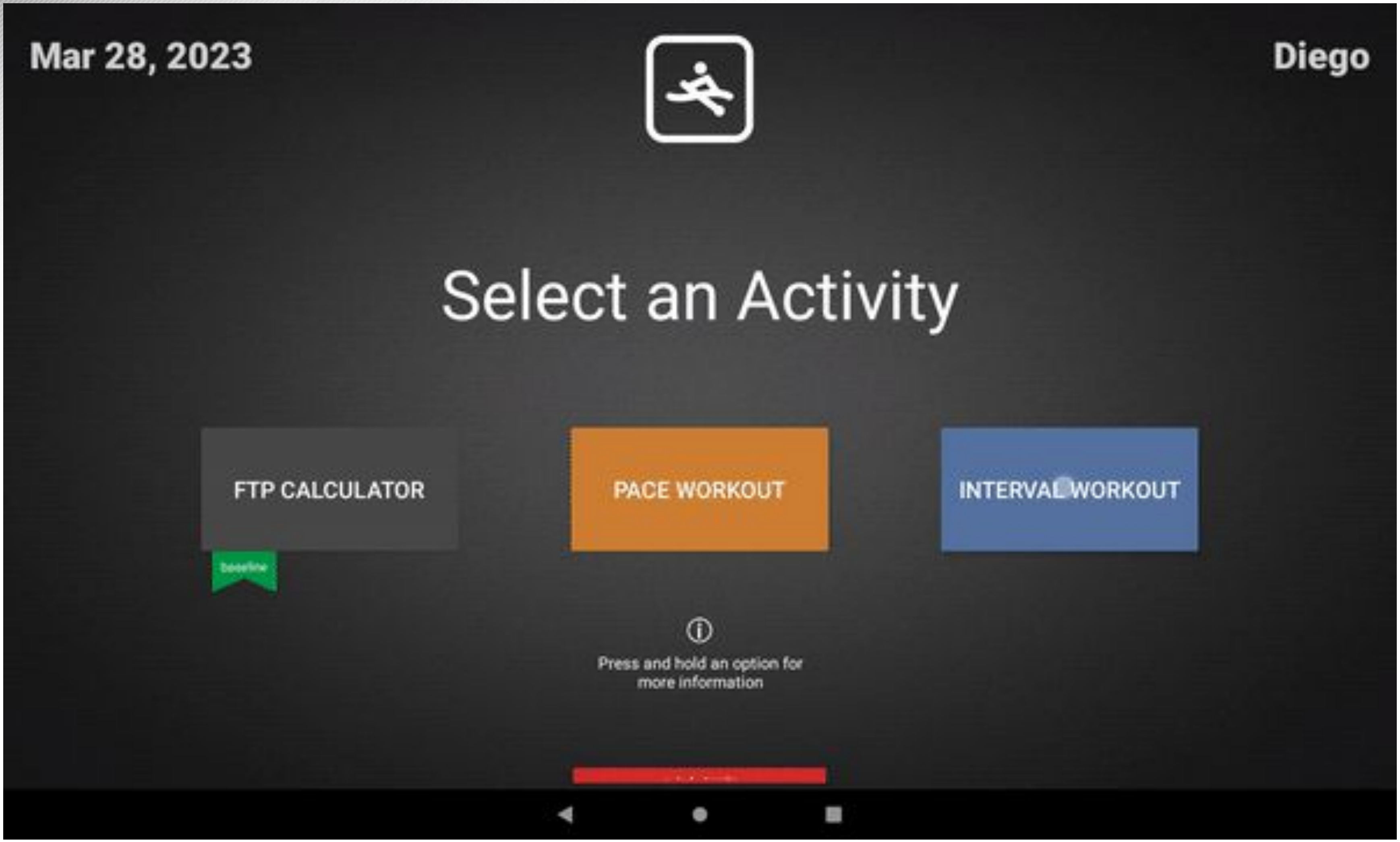


[TEST] Message changed. New message: You aren't in power zone 2!

[TEST] Resulting ArrayList: [1.0, 2.0, 2.0, 2.0, 3.0, 3.0, 4.0, 4.0, 5.0, 2.0, 6.0, 2.0, 7.0, 4.0, 8.0, 4.0]



- Dynamic X,Y bounds
- Zoomable chart interface



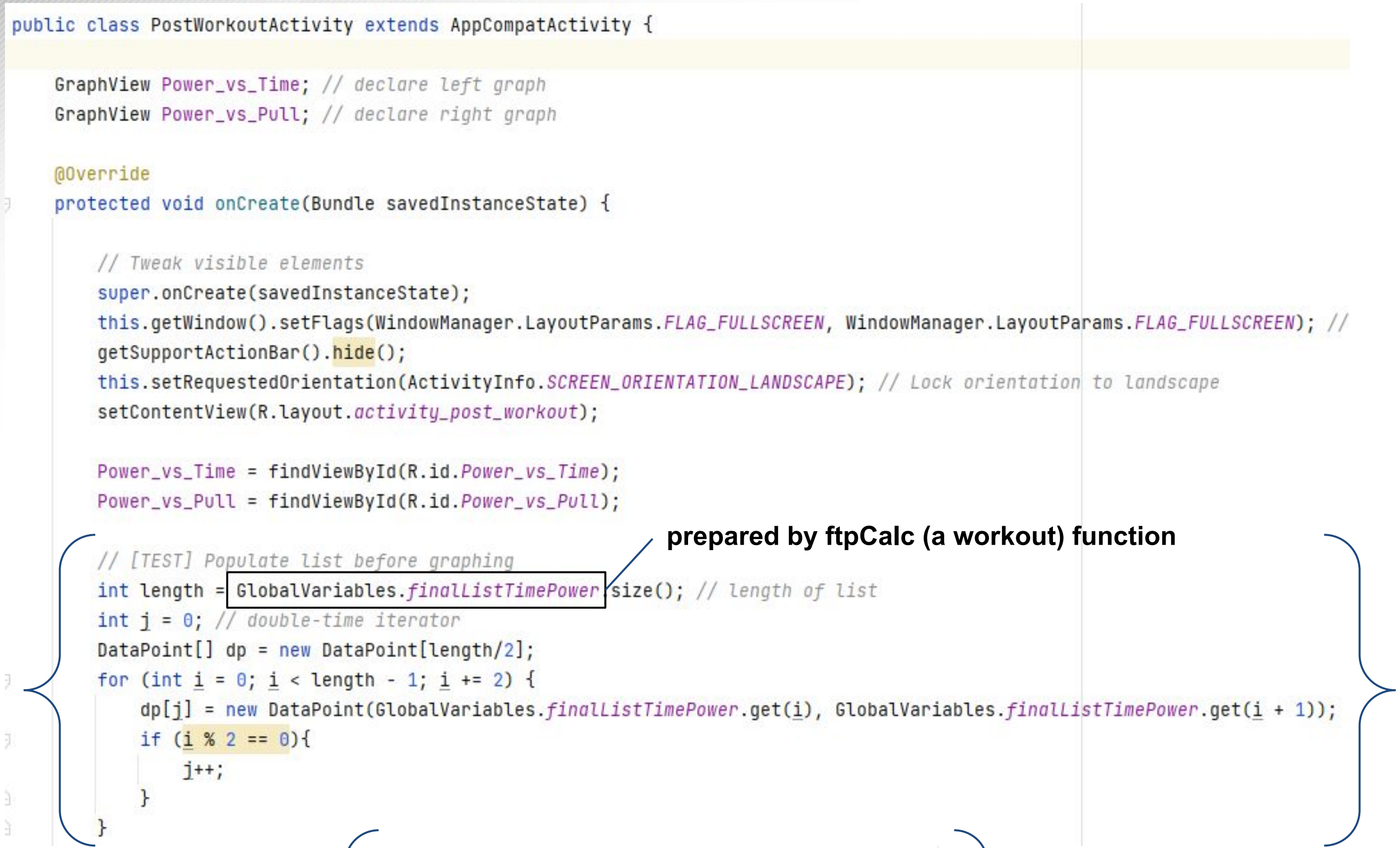
Mar 28, 2023

Diego

Select an Activity

Press and hold an option for
more information

Post Workout Activity Screen

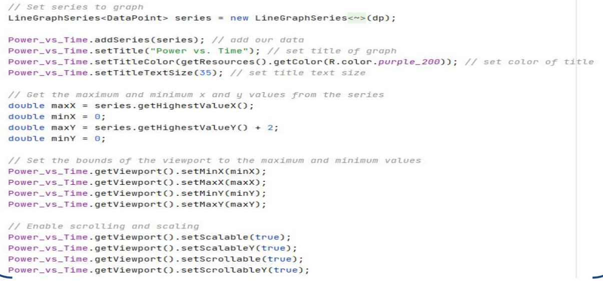


```
public class PostWorkoutActivity extends AppCompatActivity {

    GraphView Power_vs_Time; // declare left graph
    GraphView Power_vs_Pull; // declare right graph

    @Override
    protected void onCreate(Bundle savedInstanceState) {

        // Tweak visible elements
        super.onCreate(savedInstanceState);
        this.getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN); //
        getSupportActionBar().hide();
        this.setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_LANDSCAPE); // Lock orientation to landscape
        setContentView(R.layout.activity_post_workout);

        Power_vs_Time = findViewById(R.id.Power_vs_Time);
        Power_vs_Pull = findViewById(R.id.Power_vs_Pull);

        // [TEST] Populate list before graphing
        int length = GlobalVariables.finalListTimePower.size(); // length of list
        int j = 0; // double-time iterator
        DataPoint[] dp = new DataPoint[length/2];
        for (int i = 0; i < length - 1; i += 2) {
            dp[j] = new DataPoint(GlobalVariables.finalListTimePower.get(i), GlobalVariables.finalListTimePower.get(i + 1));
            if (i % 2 == 0){
                j++;
            }
        }
```

prepared by ftpCalc (a workout) function

Code to populate list to graph

...



```
// Set series to graph
LineGraphSeries<DataPoint> series = new LineGraphSeries<~>(dp);

Power_vs_Time.addSeries(series); // add our data
Power_vs_Time.setTitle("Power vs. Time"); // set title of graph
Power_vs_Time.setTitleColor(getResources().getColor(R.color.purple_200)); // set color of title
Power_vs_Time.setTitleTextSize(35); // set title text size

// Get the maximum and minimum x and y values from the series
double maxX = series.getHighestValueX();
double minX = 0;
double maxY = series.getHighestValueY() + 2;
double minY = 0;

// Set the bounds of the viewport to the maximum and minimum values
Power_vs_Time.getViewport().setMinX(minX);
Power_vs_Time.getViewport().setMaxX(maxX);
Power_vs_Time.getViewport().setMinY(minY);
Power_vs_Time.getViewport().setMaxY(maxY);

// Enable scrolling and scaling
Power_vs_Time.getViewport().setScalable(true);
Power_vs_Time.getViewport().setScalableY(true);
Power_vs_Time.getViewport().setScrollable(true);
Power_vs_Time.getViewport().setScrollableY(true);
```

Code to graph

Integrated System

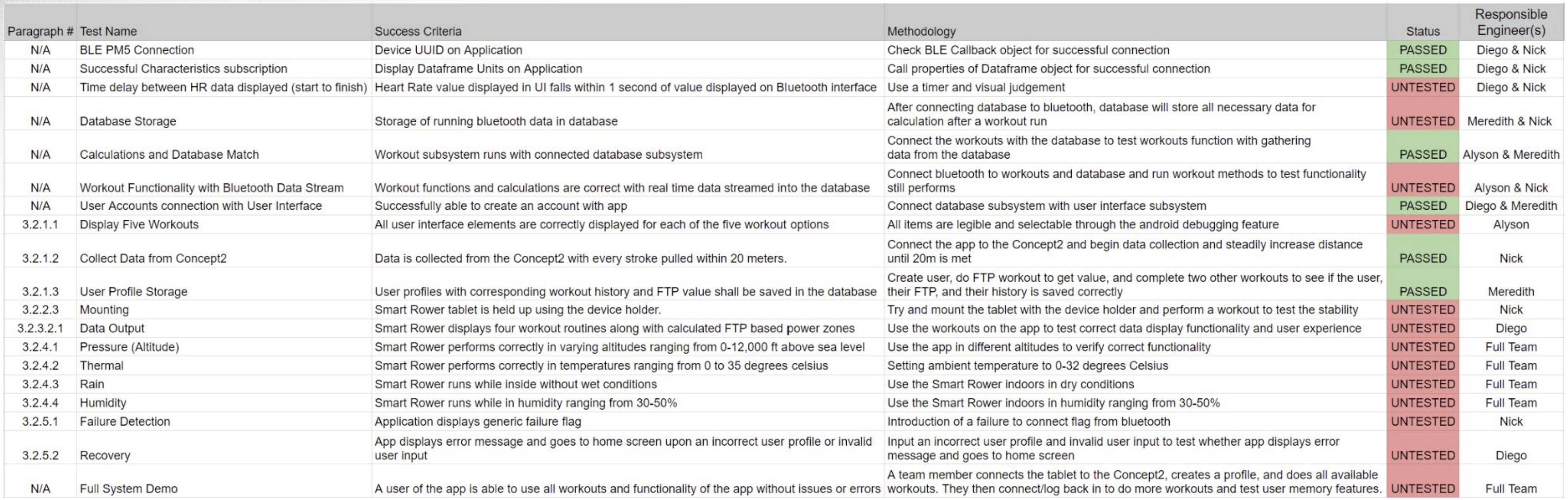
Accomplishments since last update	Ongoing progress/problems and plans until the next presentation
● UI connected login/register activity with database ● Bluetooth raw data able to populate database ● Workout methods integrated with UI and database and returning usable results	● Populate database as data is streamed in ● Validate workouts with real time data streamed in

Execution Plan

	1/23/23	1/30/22	2/6/23	2/13/23	2/20/23	2/27/23	3/6/23	3/13/23	3/20/23	3/27/23	4/3/23	4/10/23	4/17/23	4/24/23	Date
Status Update 1	🟩														1/25/23
Github Reorganization	🟩	🟩													2/3/23
Bluetooth/User Interface Integration	🟪	🟪	🟪												2/10/23
Database/Workout Intergration	🟪	🟪	🟪												2/10/23
Status Update 2			🟩												2/8/23
Validation of Initial Integration				🟪											2/17/23
Dry Run Presentation with Sponsor				🟩											2/13/23
Create New Github of Kotlin App				🟩											2/19/23
Reacheive Functionality of Subsystems in Kotlin				🟩	🟩										2/22/23
Status Update 3					🟩										2/22/23
Workout and Database Integration					🟩	🟩									3/1/23
UI and BLE Integration					🟩	🟩									3/1/23
All Subsytems Integrate						🟩									3/5/23
Design Blitz Participation						🟩									3/5/23
Status Update 4							🟩								3/8/23
Integration Debug							🟥	🟥	🟥						3/24/23
Front End Validation							🟥	🟥	🟥						3/24/23
User Interface Polishing									🟨	🟨					3/31/23
Database Encryption									⬜	⬜					3/31/23
Status Update 5										🟨					3/29/23
Back End Validation										⬜	⬜	⬜			4/12/23
Corner Case Validation										⬜	⬜	⬜			4/12/23
Final Presentation												⬜			4/12/23
Demo and Poster Preperation													⬜	⬜	4/26/23
Final System Demo														⬜	4/26/23
Project Showcase														⬜	4/28/23
Final Report														⬜	4/29/23

⬜	Not Started
🟩	Completed
🟨	In Progress
🟥	Behind Schedule
🟪	N/A

Validation Plan



Paragraph #	Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
N/A	BLE PM5 Connection	Device UUID on Application	Check BLE Callback object for successful connection	PASSED	Diego & Nick
N/A	Successful Characteristics subscription	Display Dataframe Units on Application	Call properties of Dataframe object for successful connection	PASSED	Diego & Nick
N/A	Time delay between HR data displayed (start to finish)	Heart Rate value displayed in UI falls within 1 second of value displayed on Bluetooth interface	Use a timer and visual judgement	UNTESTED	Diego & Nick
N/A	Database Storage	Storage of running bluetooth data in database	After connecting database to bluetooth, database will store all necessary data for calculation after a workout run	UNTESTED	Meredith & Nick
N/A	Calculations and Database Match	Workout subsystem runs with connected database subsystem	Connect the workouts with the database to test workouts function with gathering data from the database	PASSED	Alyson & Meredith
N/A	Workout Functionality with Bluetooth Data Stream	Workout functions and calculations are correct with real time data streamed into the database	Connect bluetooth to workouts and database and run workout methods to test functionality still performs	UNTESTED	Alyson & Nick
N/A	User Accounts connection with User Interface	Successfully able to create an account with app	Connect database subsystem with user interface subsystem	PASSED	Diego & Meredith
3.2.1.1	Display Five Workouts	All user interface elements are correctly displayed for each of the five workout options	All items are legible and selectable through the android debugging feature	UNTESTED	Alyson
3.2.1.2	Collect Data from Concept2	Data is collected from the Concept2 with every stroke pulled within 20 meters.	Connect the app to the Concept2 and begin data collection and steadily increase distance until 20m is met	PASSED	Nick
3.2.1.3	User Profile Storage	User profiles with corresponding workout history and FTP value shall be saved in the database	Create user, do FTP workout to get value, and complete two other workouts to see if the user, their FTP, and their history is saved correctly	PASSED	Meredith
3.2.2.3	Mounting	Smart Rower tablet is held up using the device holder.	Try and mount the tablet with the device holder and perform a workout to test the stability	UNTESTED	Nick
3.2.3.2.1	Data Output	Smart Rower displays four workout routines along with calculated FTP based power zones	Use the workouts on the app to test correct data display functionality and user experience	UNTESTED	Diego
3.2.4.1	Pressure (Altitude)	Smart Rower performs correctly in varying altitudes ranging from 0-12,000 ft above sea level	Use the app in different altitudes to verify correct functionality	UNTESTED	Full Team
3.2.4.2	Thermal	Smart Rower performs correctly in temperatures ranging from 0 to 35 degrees celsius	Setting ambient temperature to 0-32 degrees Celsius	UNTESTED	Full Team
3.2.4.3	Rain	Smart Rower runs while inside without wet conditions	Use the Smart Rower indoors in dry conditions	UNTESTED	Full Team
3.2.4.4	Humidity	Smart Rower runs while in humidity ranging from 30-50%	Use the Smart Rower indoors in humidity ranging from 30-50%	UNTESTED	Full Team
3.2.5.1	Failure Detection	Application displays generic failure flag	Introduction of a failure to connect flag from bluetooth	UNTESTED	Nick
3.2.5.2	Recovery	App displays error message and goes to home screen upon an incorrect user profile or invalid user input	Input an incorrect user profile and invalid user input to test whether app displays error message and goes to home screen	UNTESTED	Diego
N/A	Full System Demo	A user of the app is able to use all workouts and functionality of the app without issues or errors	A team member connects the tablet to the Concept2, creates a profile, and does all available workouts. They then connect/log back in to do more workouts and test user memory features.	UNTESTED	Full Team

Thank you for listening!