



*Dwight Look College of*

**ENGINEERING**  
TEXAS A&M UNIVERSITY

# **Team 45: TI Project #3**

## **Bi-Weekly Update 1**

**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 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.



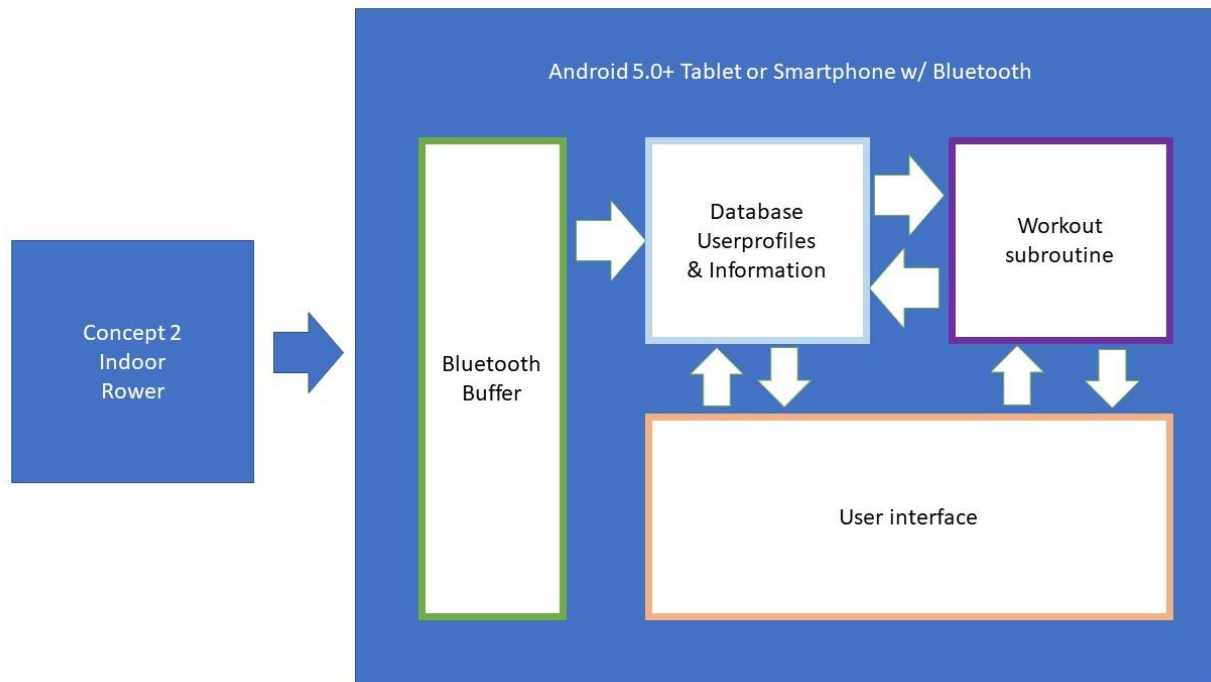
Alyson Garlick

Diego Gumucio

Meredith McKean

Nicholas McNamara

# System Overview



**Data Communication Subsystem:** Collects resistance, cadence, heart rate, and other data from Concept2 machine via Bluetooth

**Database Subsystem:** Manages and stores data for the Smart Rower, handles data transfer to subsystems.

**Workout Subsystem:** Responsible for creating the workout routines and calculating metrics to display through the user interface.

**User Interface Subsystem:** Facilitates user interaction and accessibility of data, ensuring an intuitive experience.



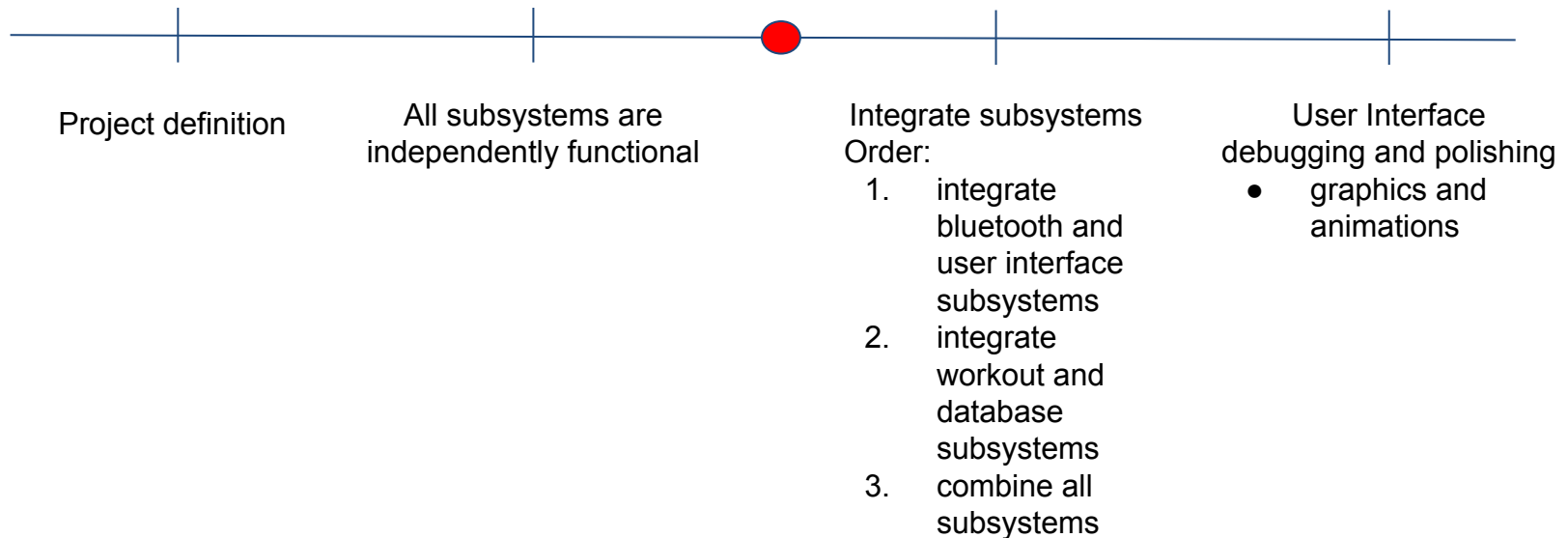
# Major Project Changes for 404

- No major changes since 403
- All 403 accomplishments were met last semester
- The current project plan matches the original plan





# Project Timeline





# Data Communication Subsystem

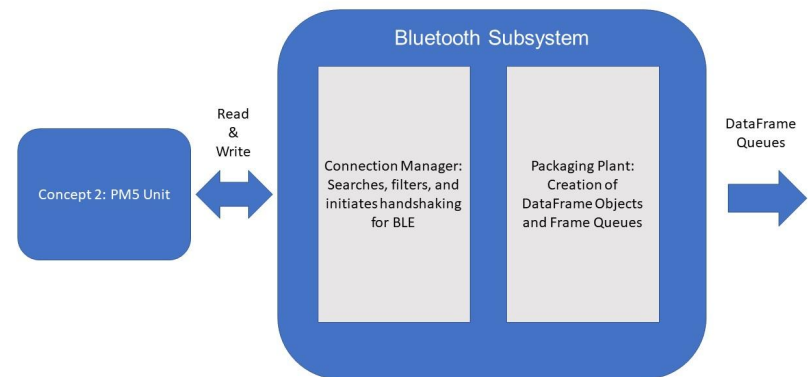
Nicholas McNamara

Accomplishments since 403 3 hrs of effort	Ongoing progress/problems and plans until the next presentation
<ul style="list-style-type: none"><li>Ensured that experimental data types from Kotlin will work in Java</li></ul>	<ul style="list-style-type: none"><li>Discuss and integrate with user interface and database</li><li>Make sure all Kotlin base objects/classes are accessible in Java</li><li>Ensure all GitHub files are in the proper subsections for a fork for integration</li><li>Order Tablet Holder for Concept2 rower</li></ul>

# Data Communication Subsystem

Nicholas McNamara

- Written in Kotlin as opposed to Java
- Uses Bluetooth Low Energy (BLE), therefore is asynchronous
- The connection manager and packaging/queue are intertwined with the android demo driver application
- In order to avoid application crashes a certain calling order is necessary.



```
I The queue full: 10 items
I Popping the most recent dataframe.
I Elapsed Time: 132.51 seconds(s)
I Distance: 512.90 meter(s)
I Drive Length: 109.00 meter(s)
I Drive Time: 60.00 second(s)
I Stroke Recovery time: 1.11 second(s)
I Stroke Distance: 8.10 meter(s)
I Peak Drive Force: 139.30 lbs
I Average Drive Force: 93.50 lbs
I Work Per Stroke: 420.20 Joule(s)
I Stroke Count: 64
I The queue will now clear.
I Characteristic ce060035-43e5-11e4-916c-0800200c9a66 changed | value: Hex: c2 37 00 b0 1
I An item has been placed in the 35 Queue.
```



# Database Subsystem

Meredith McKean

Accomplishments since 403 5 hrs of effort	Ongoing progress/problems and plans until the next presentation
<ul style="list-style-type: none"><li>• Fixed and removed warnings/bugs from original code</li><li>• Made a document explaining all current methods of the local SQLite Database</li><li>• All code is updated on Github along with its documents</li><li>• Current program from 403 is running and working</li></ul>	<ul style="list-style-type: none"><li>• clean github repository</li><li>• began integration with workout subsystem</li></ul>



## Database Subsystem

# Meredith McKean

## “user\_info” Table

COLUMN_1	COLUMN_2	COLUMN_3	COLUMN_4	COLUMN_5	COLUMN_6	COLUMN_7	COLUMN_8	COLUMN_9	COLUMN_10
1	100	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0
2	100	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0
3	100	11.0	12.0	13.0	14.0	15.0	16.0	17.0	18.0

### “dataframe33\_info” Table

C.	COLUMN_TIME_33	COLUMN_INTERVAL	COLUMN_POWER	COLUMN_TOTAL_CAL	COLUMN_SPLIT_PACE	COLUMN_SPLIT_POWER	COLUMN_SPLIT_CAL	COLUMN_LAST_SPLIT_TIME	COLUMN_LAST_SPLIT_DIST
1	1.0	2	3	4	5	6	7	8	9
2	1.0	2	3	4	5	6	7	8	9
3	1.0	2	3	4	5	6	7	8	9

### “dataframe35\_info” Table

C.	COLUMN_TIME_33	COLUMN_INTERVAL	COLUMN_POWER	COLUMN_TOTAL_CAL	COLUMN_SPLIT_FACE	COLUMN_SPLIT_POWER	COLUMN_SPLIT_CAL	COLUMN_LAST_SPLIT_TIME	COLUMN_LAST_SPLIT_DIST
1	1.0	2	3	4	5	6	7	8	9
2	1.0	2	3	4	5	6	7	8	9
3	1.0	2	3	4	5	6	7	8	9

## “history\_info” Table

C...	COLUMN_USER	COLUMN_TIMESTAMP ▲	COLUMN_WORKOUT
1	star	2022-12-04 12:26:48	workout1
2	alice	2022-12-04 12:27:01	workout1
3	alice	2022-12-04 12:27:09	workout1
4	bob	2022-12-04 12:27:22	workout1

```
alice Workouts
2022-12-04 12:27:01
Workout :workout1
2022-12-04 12:27:09
Workout :workout1
```

## “error\_info” Table

C...	COLUMN_USER	COLUMN_TIMESTAMP	COLUMN_ERROR
1	star	2022-12-04 12:26:48	5
2	alice	2022-12-04 12:27:01	5
3	alice	2022-12-04 12:27:09	5
4	bob	2022-12-04 12:27:22	5

```
alice Errors
2022-12-04 12:27:01 Errors: 5
2022-12-04 12:27:09 Errors: 5
```



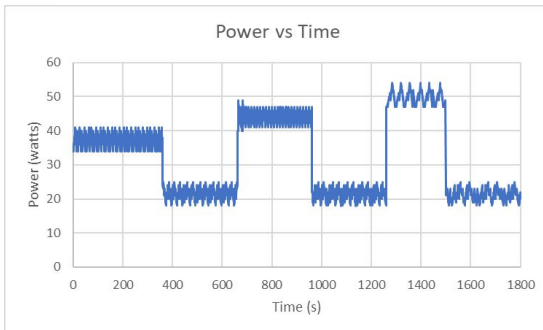
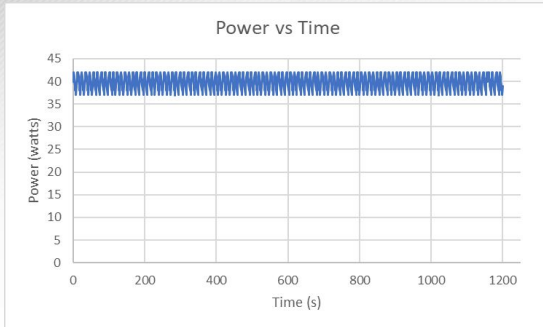
# Workout Subsystem

Alyson Garlick

Accomplishments since 403 3 hrs of effort	Ongoing progress/problems and plans until the next presentation
<ul style="list-style-type: none"><li>• Code from 403 functions fully without error<ul style="list-style-type: none"><li>- FTP, Pace, and Three Interval Workouts</li><li>- Two graphing functions</li></ul></li><li>• Bug testing of functions</li><li>• All code is backed up to GitHub</li></ul>	<ul style="list-style-type: none"><li>• Integrate with the database to change the way the functions are getting raw data</li><li>• Create document outlining how workout subsystem code works for teammate reference.</li><li>• Update GitHub readme file</li></ul>

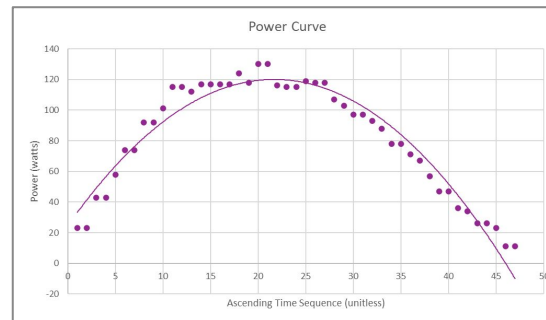
# Workout Subsystem

Alyson Garlick



- Graphing function output examples for pace and Interval 2 workouts

- Graphing function output example for Power Curve graph



Your FTP is 45

Power Zone 1: < 25 watts

Power Zone 2: 25 to 33 watts

Power Zone 3: 34 to 39 watts

Power Zone 4: 40 to 46 watts

Power Zone 5: 47 to 53 watts

Power Zone 6: 54 to 66 watts

Power Zone 7: > 67 watts

- FTP workout output using test data



# User Interface Subsystem

Diego Gumucio

Accomplishments since 403 3 hrs of effort	Ongoing progress/problems and plans until the next presentation
<ul style="list-style-type: none"><li>• Implemented working Chronometer for ongoing workouts</li><li>• Added SQLite library into app's build.gradle file</li></ul>	<ul style="list-style-type: none"><li>• Continuous integration and testing</li><li>• Begin integrating with database</li><li>• Integrate BT and UI subsystems</li><li>• Read from Database subsystem</li></ul>

# User Interface Subsystem

Diego Gumucio

```

37 androidTestImplementation 'androidx.test.ext:junit:1.1.4'
38 androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.0'
39 implementation 'pl.droidsonroids.gif:android-gif-drawable:1.2.22'
40
41 // SQLite library implementation
42 implementation 'com.readystatesoftware.sqliteasset:sqliteassethelper:+'
43
44 }

```

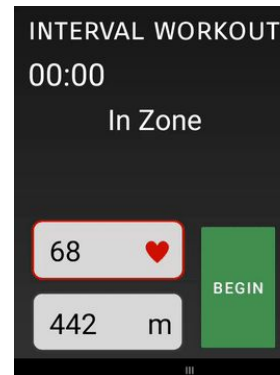
```

33 // Chronometer Functionality
34 chron = (Chronometer) findViewById(R.id.simpleChronometer);
35 btnStartChron = (Button) findViewById(R.id.btnBegin);
36
37 btnStartChron.setOnClickListener(new View.OnClickListener() { // start/
38     @Override
39     public void onClick(View v) {
40         if (!isChronRunning) { // if NOT running
41             chron.setBase(SystemClock.elapsedRealtime()); // start coun
42             chron.start(); // start the chronometer
43             btnStartChron.setText("Stop");
44             isChronRunning = true; // set status to true
45         }
46         else {
47             chron.stop();
48             isChronRunning = false; // set status to false
49             btnStartChron.setText("Start");
50         }
51     }
52 }
53
54 });

```

- Chronometer works
  - Reset and stop feature won't be triggered by the button
  - Implemented using Chronometer widget
  - Will govern the duration of workout

- SQLite library
  - Basic implementation ready
  - Have yet to import Meredith's database and try calling methods





# Parts Ordering Status

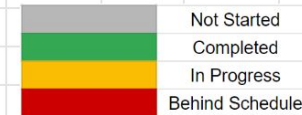
- All parts have been received except the tablet mount
- Tablet mount will arrive within two weeks





# 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
Integrate the Two Systems															3/3/23
Status Update 3															2/22/23
Design Blitz Participation															3/4/23
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															TBA
Final Report															4/29/23





# 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	UNTESTED	Diego & Nick
N/A	Successful Characteristics subscription	Display Dataframe Units on Application	Call properties of Dataframe object for successful connection	UNTESTED	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	UNTESTED	Alyson & Meredith
	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
	User Accounts connection with User Interface	Successfully able to create an account with app	Connect database subsystem with user interface subsystem	UNTESTED	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	UNTESTED	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



*Dwight Look College of*

**ENGINEERING**  
TEXAS A&M UNIVERSITY

**Thank you for listening!**