

Group E / Iteration 3

Fitness Mobile

Iteration 3 Improvements

1. UI Improvements

- a. Bottom Bar For Navigation
- b. New icon

2. Sleep Tracking

- a. Allows the user to track sleep amounts for each day

3. Google Fit Integration

- a. Pulls Calorie Burned and Step Count information on Google Fit into the app

4. Initial Home Page User Interface

- a. Provides a centralized dashboard for display of all of the information within the app
- b. Initial code only, further work will occur in Iteration 4

Added Databases

```
1 import 'dart:io';
2
3 import 'package:path/path.dart';
4 import 'package:path_provider/path_provider.dart';
5 import 'package:sqflite/sqflite.dart';
6 import '../sleep_log.dart';
7
8 class SleepDatabase {
9   // DB info
10   static final databaseName = "dbsleep.db";
11   static final databaseVersion = 1;
12   static final tableName = "sleep";
13
14   // DB columns
15   static final columnId = 'id';
16   static final columnStartTime = 'start_time';
17   static final columnEndTime = 'end_time';
18
19   // beginning with an '_' means private constructor
20   SleepDatabase._privateConstructor();
21
22   // Singleton -- only one instance
23   static final SleepDatabase instance = SleepDatabase._privateConstructor();
24
25   // Database reference
26   static Database _database;
27
28   // get database function
29   Future<Database> get database async {
30     if (_database != null) return _database;
31     _database = await _initDatabase();
32     return _database;
33   }
34
35   // initialize database helper function for opening and creating database
36   _initDatabase() async {
37     Directory documentsDirectory = await getApplicationDocumentsDirectory();
38     String path = join(documentsDirectory.path, _databaseName);
39     return await openDatabase(path,
40       version: _databaseVersion, onCreate: _onCreate);
41   }
42
43   // helper function to create SQL table
44   Future _onCreate(Database db, int version) async {
45     await db.execute('''
46       CREATE TABLE $tableName (
47         $columnId INTEGER PRIMARY KEY,
48         $columnStartTime INTEGER NOT NULL,
49         $columnEndTime INTEGER NOT NULL,
50       )
51     ''');
52   }
53 }
```

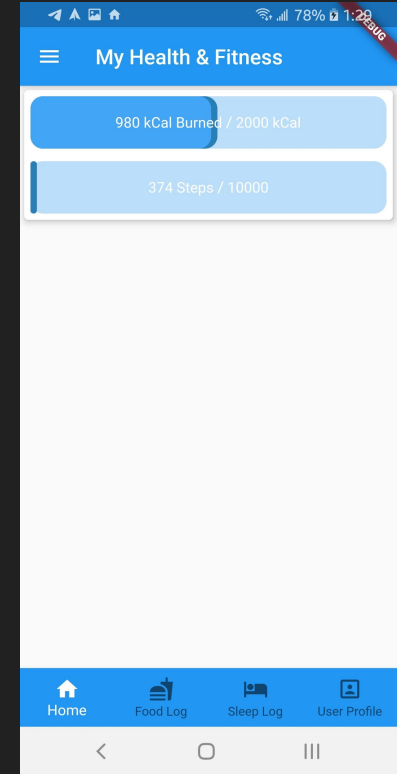
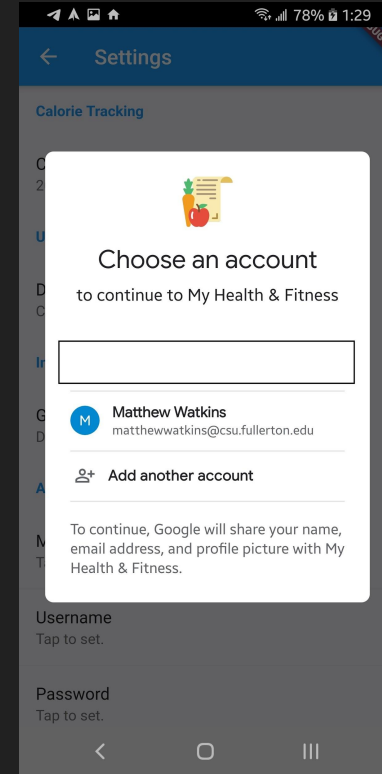
```
1 import 'dart:io';
2
3 import 'package:path/path.dart';
4 import 'package:path_provider/path_provider.dart';
5 import 'package:sqflite/sqflite.dart';
6 import '../sleep_log.dart';
7
8 class UserDatabase {
9   // DB info
10   static final databaseName = "dbuser.db";
11   static final databaseVersion = 1;
12   static final tableName = "user";
13
14   // DB columns
15   static final columnId = 'id';
16   static final columnUsername = 'username';
17   static final columnAge = 'age';
18   static final columnHeight = 'height';
19   static final columnWeight = 'weight';
20   static final columnBMI = 'bmi';
21   static final columnTimestamp = 'timestamp';
22
23   // beginning with an '_' means private constructor
24   UserDatabase._privateConstructor();
25
26   // Singleton -- only one instance
27   static final UserDatabase instance = UserDatabase._privateConstructor();
28
29   // Database reference
30   static Database _database;
31
32   // get database function
33   Future<Database> get database async {
34     if (_database != null) return _database;
35     _database = await _initDatabase();
36     return _database;
37   }
38
39   // initialize database helper function for opening and creating database
40   _initDatabase() async {
41     Directory documentsDirectory = await getApplicationDocumentsDirectory();
42     String path = join(documentsDirectory.path, _databaseName);
43     return await openDatabase(path, version: _databaseVersion, onCreate: _onCreate);
44   }
45
46   // helper function to create SQL table
47   Future _onCreate(Database db, int version) async {
48     await db.execute('''
49       CREATE TABLE $tableName (
50         $columnId INTEGER PRIMARY KEY,
51         $columnBMI REAL NOT NULL,
52         $columnTimestamp INTEGER NOT NULL,
53       )
54     ''');
55   }
56 }
57
```

Interface to database

```
40
41 class SleepLogStorageProvider extends StorageProvider<SleepEvent> {
42     @override
43     void delete(SleepEvent item) async {
44         //FoodDatabase.instance.deleteByTimestamp(item.time);
45     }
46
47     @override
48     void write(SleepEvent item) {
49         //FoodDatabase.instance.insert(item);
50     }
51
52     @override
53     void writeAll(List<SleepEvent> items) {
54
55     }
56
57     @override
58     Future<List<SleepEvent>> read(DateTime date) {
59         Future<List<SleepEvent>> temp;
60         return temp;
61     }
62
63     static SleepLogStorageProvider get instance => SleepLogStorageProvider();
64 }
```

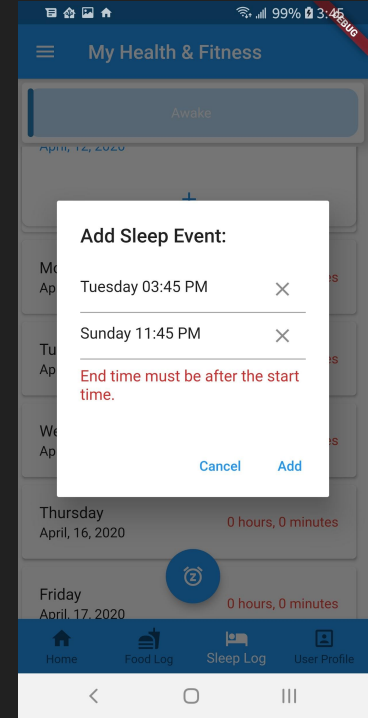
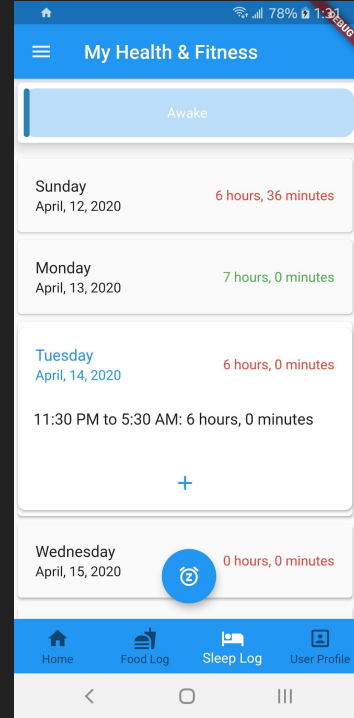
Homepage & Google Fit Integration

- Google Fit integration is functioning
- Users can connect to Google Fit, and their information will automatically be updated in the app.
- Initial homepage code and interface was added to demonstrate integration, further data will be added in the next iteration
- UI was also updated with a bottom bar and a new icon (displayed to the left)



Sleep Tracking

- Users can track their sleep per day
 - Users can add sleep information through a dialog or with a start and stop button
- Sleep amounts are displayed and unhealthy sleep amounts are highlighted in red
- Top bar keeps track of how much sleep has accumulated when using the sleep stopwatch
- Add sleep dialog properly handles error conditions



About Us / Help Center

ss-mobile > lib > pages > user_page.dart > ...

```
}

Widget _buildUserPage() {
  return Center(
    child: Column(
      children: <Widget>[
        EditUserInfo(),
        FlatButton(
          child: Text("About"),
          onPressed: () {
            Navigator.push(context, MaterialPageRoute(
              builder: (BuildContext context) {
                return Scaffold(body: about_us());
              },
            ));
          },
        ),
        FlatButton(
          child: Text("Help Center"),
          onPressed: () {
            Navigator.push(context, MaterialPageRoute(
              builder: (BuildContext context) {
                return Scaffold(body: help_center());
              },
            ));
          },
        ),
      ],
    ),
  );
}
```

To start off, I had to link the about us and help center to the rest of the application.

So I made a Widget, which was a flat button to redirect to the new U/I page. This was done in the user profile.

About Us / Help Center

fitness-mobile > lib > pages > about_us.dart > about_us > build

```
1 import 'package:flutter/material.dart';
2
3 class about_us extends StatelessWidget {
4   @override
5   Widget build(BuildContext context) {
6     return Scaffold(
7
8     backgroundColor: Colors.white,
9     appBar: AppBar(
10       brightness: Brightness.light,
11       backgroundColor: Colors.blue,
12       elevation: 0,
13       title: Text("Meet The Team", style: TextStyle(color: Colors.white,
14     ),
15     body: SafeArea(
16       child: SingleChildScrollView(
17         child: Padding(
18           padding: EdgeInsets.all(20.0),
19
20           child: Column(
21             crossAxisAlignment: CrossAxisAlignment.start,
22             children: <Widget> [
23               SizedBox(height: 10,),
24               Text("Licenses, Agreements, and Attributes", style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold,color: Colors.black)),
25               SizedBox(height: 20,),
26               Text("""name: csuf_fitness
27 description: Allows users to track their health and fitness.
28
29 The following defines the version and build number for your application.
30 A version number is three numbers separated by dots, like 1.2.43
31 followed by an optional build number separated by a +.
32 Both the version and the builder number may be overridden in flutter
33 build by specifying --build-name and --build-number, respectively.
34 In Android, build-name is used as versionName while build-number used as versionCode.
```

{double fontSize}

The size of glyphs (in logical pixels) to use when painting the text.

During painting, the [fontSize] is multiplied by the current `textScaleFactor` to let users make it easier to read text by increasing its size.

[getParagraphStyle] will default to 14 logical pixels if the font size isn't specified here.

The about us is consisted of an app bar at the top. The app bar is used to let the user know what page they are on at all times. After I established that, I had to take the context of the .yaml and add it under the licenses and attributes section. In order to add multiple line text, I had to use the multiple quotes. For example,

"""" Dependencies and licenses """"

Also, had to use a scroll down widget to not overload the U/I page with pixels.

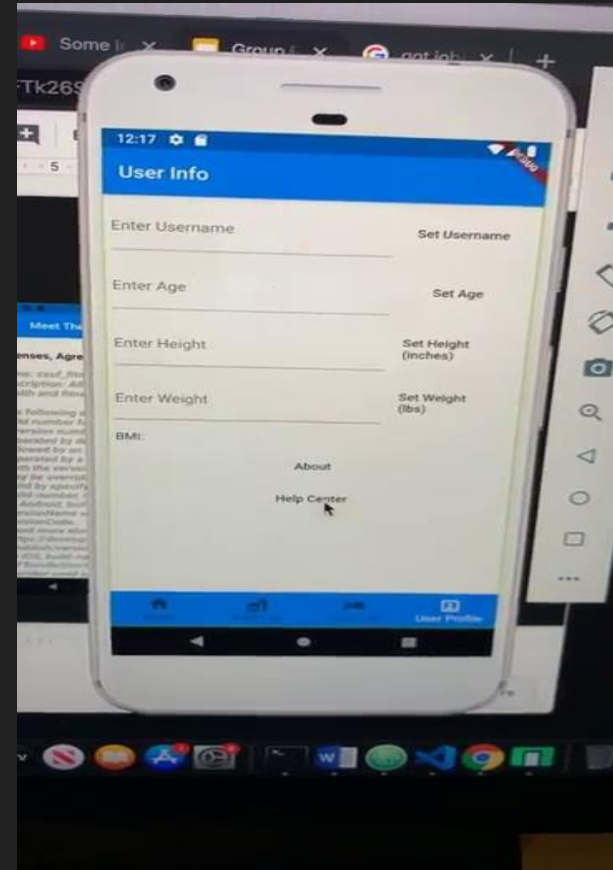
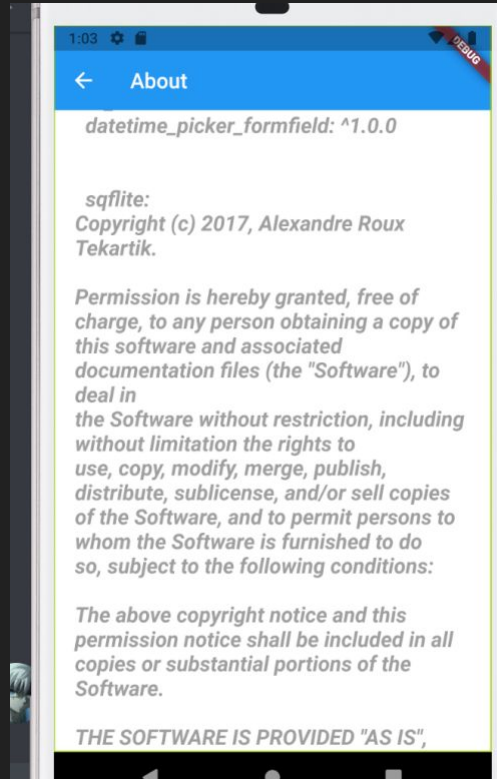
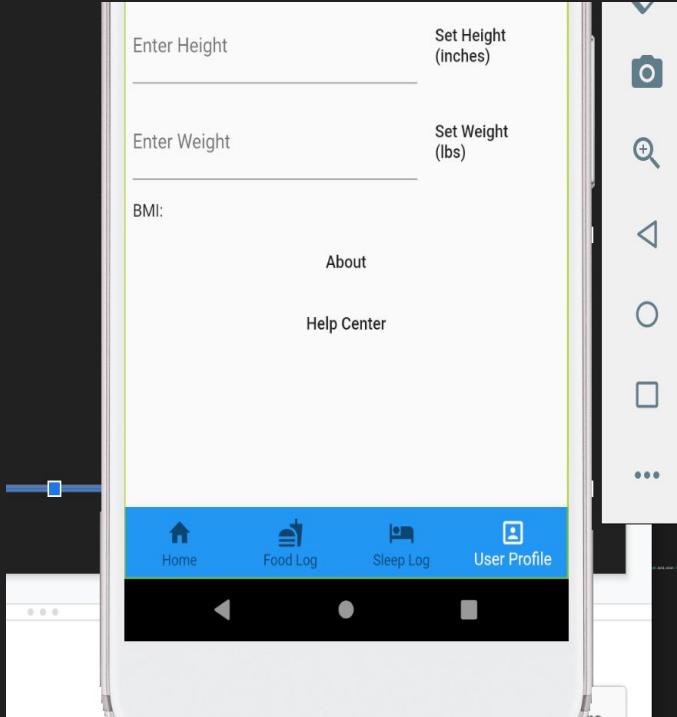
About Us / Help Center

fitness-mobile > lib > pages > help_center.dart > help_center

```
1  import 'package:flutter/material.dart';
2
3  class help_center extends StatelessWidget{
4    Tween<double> _scaleTween = Tween<double>(begin: 1, end:2 );
5    @override
6    Widget build(BuildContext context) {
7      return MaterialApp(
8        title: 'Welcome To The Help Center ',
9        home: Scaffold (
10          body: Center (
11            child: TweenAnimationBuilder(
12              tween: _scaleTween,
13              duration: Duration (seconds: 1),
14              builder: (context, scale, child) {
15                return Transform.scale(scale: scale, child: child);
16              },
17
18              child: Text(
19                'Welcome To The Help Center ',
20                style: TextStyle(fontSize: 9, fontWeight: FontWeight.bold,color: Colors.black,height: 0),
21              )
22            ),
23          ),
24        ),
25      ),
26    ),
27    );
28  }
29
30
```

After I finished the about us page, I moved on to the help center. The design of this page is a little different than the about us. The goal was to somehow mimic some of the artwork that goes into Twitters UI design. For that, an animation was implemented. To bring a more of a welcoming interaction with the user after being redirected to the help center.

About Us / Help Center



Iteration 4 Goals

- Weight tracking
- UI refinements
- Update the homepage with more user information
- Suggestions for daily calorie counts based on user profile